Developer Guide

AWS WAF, AWS Firewall Manager, AWS Shield Advanced, and AWS Shield network security director

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS WAF, AWS Firewall Manager, AWS Shield Advanced, and AWS Shield network security director: Developer Guide

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Services or capabilities described in AWS documentation might vary by Region. To see the differences applicable to the AWS European Sovereign Cloud Region, see the AWS European Sovereign Cloud User Guide.

Table of Contents

..... xiii

What are AWS WAF, Shield Advanced, AWS Shield network security director and Firewall Manager? 1

AWS WAF 1

Shield Advanced 3

AWS Shield network security director 3

AWS Firewall Manager 4

Setting up your account 5

Sign up for an AWS account 5

Download tools 5

AWS WAF 7

Getting started with AWS WAF 8

Set up AWS WAF (new console) 9

Set up AWS WAF (standard console) 13

How AWS WAF works 19

Resources that you can protect with AWS WAF 21

Working with the updated console experience 23

Understanding the new dashboards 24

Configuring protection 26

Creating a protection pack (web ACL) 28

Editing a protection pack (web ACL) 38

Managing rule group behavior 44

Associating or disassociating protection with an AWS resource 47

Using protection packs (web ACLs) with rules and rule groups 51

Setting the protection pack (web ACL) default action 58

Body inspection considerations 60

Configuring CAPTCHA, challenge, and tokens 62

Viewing web traffic metrics 62

Deleting a protection pack (web ACL) 63

AWS WAF rules 64

Using rule actions 66

Using rule statements 70

Using match rule statements 101

Using logical rule statements 126

Using rate-based rule statements ... 134
Using rule group rule statements ... 155
AWS WAF rule groups ... 158
Using managed rule groups ... 159
Managing your own rule groups ... 381
AWS Marketplace rule groups ... 387
Recognizing rule groups from other services ... 390
Web ACL capacity units (WCUs) ... 391
Determining the WCUs for a rule group, protection pack (web ACL), or web ACL ... 393
Oversize web request components ... 393
Blocking oversized components ... 396
Supported regular expression syntax ... 397
IP sets and regex pattern sets ... 397
Creating and managing an IP set ... 398
Creating and managing a regex pattern set ... 400
Customized web requests and responses ... 402
Inserting custom request headers ... 404
Dynamic label interpolation ... 407
Sending custom responses ... 412
Supported response status codes ... 415
Web request labeling ... 417
How labeling works ... 419
Label syntax and naming requirements ... 421
Rules that add labels ... 424
Rules that match labels ... 425
Intelligent threat mitigation ... 430
Mitigation options ... 431
Best practices ... 442
Tokens in intelligent threat mitigation ... 445
AWS WAF Fraud Control account creation fraud prevention (ACFP) ... 459
AWS WAF Fraud Control account takeover prevention (ATP) ... 482
AWS WAF Bot Control ... 502
Distributed Denial of Service (DDoS) prevention ... 541
Client application integrations ... 553
CAPTCHA and Challenge ... 593
AI traffic monetization ... 605

How AI traffic monetization works ... 606
Getting started with AI traffic monetization ... 608
Pricing configuration ... 611
Payment networks and settlement ... 614
Revenue analytics ... 615
Communicating license terms to AI agents ... 617
Data protection and logging for web traffic ... 618
Logging ... 619
Data protection ... 664
Testing and tuning your protections ... 691
Testing and tuning high-level steps ... 692
Preparing for testing ... 693
Monitoring and tuning your AWS WAF protections ... 696
Enabling your protections in production ... 713
Using AWS WAF with Amazon CloudFront ... 715
How AWS WAF works with different distribution types ... 716
Using AWS WAF with CloudFront Pricing Plans ... 718
AI traffic monetization ... 719
Use cases ... 720
Security in your use of the AWS WAF service ... 724
Protecting your data ... 725
Using IAM with AWS WAF ... 726
Logging and monitoring ... 796
Validating compliance ... 797
Building for resilience ... 798
Infrastructure security ... 798
AWS WAF quotas ... 798
Migrating your AWS WAF Classic resources to AWS WAF ... 805
Why migrate to AWS WAF? ... 806
Migration caveats ... 807
How the migration works ... 809
Migrating a protection pack (web ACL) ... 810
AWS WAF Classic ... 817
Setting up AWS WAF Classic ... 818
Sign up for an AWS account ... 5
Download tools ... 819

How AWS WAF Classic works ... 820
AWS WAF Classic pricing ... 824
Getting started with AWS WAF Classic ... 825
Step 1: Set up AWS WAF Classic ... 826
Step 2: Create a Web ACL ... 826
Step 3: Create an IP match condition ... 827
Step 4: Create a geo match condition ... 828
Step 5: Create a string match condition ... 829
Step 5A: Create a regex condition (optional) ... 831
Step 6: Create a SQL injection match condition ... 833
Step 7: (Optional) create additional conditions ... 834
Step 8: Create a rule and add conditions ... 835
Step 9: Add the rule to a Web ACL ... 837
Step 10: Clean up your resources ... 838
Creating and configuring a Web Access Control List (Web ACL) ... 840
Working with conditions ... 843
Working with rules ... 890
Working with web ACLs ... 902
Working with AWS WAF Classic rule groups for use with AWS Firewall Manager ... 919
Creating an AWS WAF Classic rule group ... 920
Adding and deleting rules from an AWS WAF Classic rule group ... 922
Getting started with AWS Firewall Manager to enable AWS WAF Classic rules ... 924
Step 1: Complete the prerequisites ... 925
Step 2: Create rules ... 925
Step 3: Create a rule group ... 926
Step 4: Create and apply an AWS Firewall Manager AWS WAF Classic policy ... 928
Tutorial: Creating an AWS Firewall Manager policy with hierarchical rules ... 930
Step 1: Designate a Firewall Manager administrator account ... 931
Step 2: Create a rule group using the Firewall Manager administrator account ... 931
Step 3: Create a Firewall Manager policy and attach the common rule group ... 932
Step 4: Add account-specific rules ... 932
Conclusion ... 932
Logging Web ACL traffic information ... 933
Listing IP addresses blocked by rate-based rules ... 940
How AWS WAF Classic works with Amazon CloudFront features ... 941
Using AWS WAF Classic with CloudFront custom error pages ... 942

Using AWS WAF Classic with CloudFront for applications running on your own HTTP server ... 942
Choosing the HTTP methods that CloudFront responds to ... 943
Security ... 944
Data protection ... 945
Identity and access management ... 947
Logging and monitoring ... 969
Compliance validation ... 970
Resilience ... 971
Infrastructure security ... 971
AWS WAF Classic quotas ... 972
AWS Shield ... 977
How Shield and Shield Advanced work ... 978
AWS Shield Standard overview ... 980
AWS Shield Advanced overview ... 980
Resources that Shield Advanced protects ... 982
Shield Advanced capabilities and options ... 982
Deciding whether to subscribe to AWS Shield Advanced ... 985
Examples of DDoS attacks ... 987
How Shield detects events ... 988
How Shield mitigates events ... 993
Building DDoS resilient architectures ... 1001
DDoS resiliency architecture for web applications ... 1002
DDoS resiliency architecture for TCP and UDP applications ... 1004
Combining Shield Advanced with other AWS services ... 1006
Setting up AWS Shield Advanced ... 1007
Subscribing to Shield Advanced ... 1008
Adding and configuring resource protections ... 1010
Setting up SRT support ... 1016
Creating a DDoS dashboard ... 1018
SRT support ... 1019
Granting access for the SRT ... 1020
Setting up proactive engagement ... 1023
Contacting the SRT ... 1024
Setting up custom mitigations with the SRT ... 1025
Resource protections ... 1026

List of protected resources ... 1027
Protecting Amazon EC2 instances and Network Load Balancers ... 1028
Protecting the application layer (layer 7) ... 1029
Health-based detection using health checks ... 1047
Adding protection to a resource ... 1057
Editing protections ... 1058
Creating alarms and notifications ... 1060
Removing protection from a resource ... 1061
Protection groups ... 1062
Tracking protection changes ... 1065
Visibility into DDoS events ... 1065
Global and account activity ... 1067
Events ... 1070
Event visibility across accounts ... 1080
Attack flow logs ... 1082
Enable publishing flow logs to Amazon S3 ... 1083
To enable flow log delivery ... 1085
Flow log files ... 1086
Flow log record syntax ... 1086
Responding to DDoS events ... 1087
Contacting support for an application layer attack ... 1088
Manually mitigating an application layer attack ... 1090
Requesting a credit after an attack ... 1091
Security in your use of the Shield service ... 1093
Protecting your data ... 1094
Using IAM with Shield ... 1095
Logging and monitoring ... 1121
Validating compliance ... 1122
Building for resilience ... 1122
Infrastructure security ... 1122
AWS Shield Advanced quotas ... 1123
AWS Shield network security director (preview) ... 1124
AWS Shield network security director pricing ... 1124
Use cases ... 1124
Key concepts ... 1125
Findings reference ... 1126

Setting up your account 1129
Prerequisites 1129
Understanding AWS Organizations integration 1130
Choosing a delegated administrator 1130
IAM requirements for the delegated administrator 1131
Setup checklist 1133
Enabling AWS Shield network security director 1134
Enabling AWS Shield network security director and delegating a service administrator ... 1134
Enabling AWS Shield network security director for member accounts with delegated administrator 1136
Exploring resources and findings 1137
To identify which accounts have findings 1137
Analyze network security with Amazon Q Developer 1140
Example questions 1141
Data protection considerations 1141
AWS Shield network security director quotas 1142
Troubleshooting AWS Shield network security director 1142
Unsupported Cross-Account Shared Resources 1142
Availability of Resources, Findings, and Suppression 1143
Performance Considerations 1143
Additional Resources 1144
Security 1144
Identity and Access Management 1145
Identity-based policy examples 1146
Using service-linked roles 1150
Logging AWS Shield network security director API calls with AWS CloudTrail 1158
network security director information in CloudTrail 1158
network security director API operations logged by CloudTrail 1158
Understanding network security director log file entries 1159
Monitoring CloudTrail logs with Amazon CloudWatch 1160
Best practices for CloudTrail with network security director 1160
AWS Firewall Manager 1161
AWS Firewall Manager prerequisites 1162
Joining and configuring AWS Organizations for using Firewall Manager 1162
Creating an AWS Firewall Manager default administrator account 1163
Enabling AWS Config for using Firewall Manager 1164

Subscribing in the AWS Marketplace and configuring third-party settings for Firewall Manager third-party policies ... 1166
Enabling resource sharing for Network Firewall and DNS Firewall policies with AWS RAM ... 1167
Using AWS Firewall Manager in Regions that are disabled by default ... 1167
Using Firewall Manager administrators ... 1168
Creating a Firewall Manager administrator account ... 1170
Updating a Firewall Manager administrator account ... 1172
Revoking a Firewall Manager administrator account ... 1172
Changing the default administrator account ... 1173
Disqualifying changes to an administrator account ... 1174
Setting up AWS Firewall Manager policies ... 1175
Setting up AWS WAF policies ... 1176
Setting up AWS Shield Advanced policies ... 1179
Setting up Amazon VPC security group policies ... 1185
Setting up Amazon VPC network ACL policies ... 1188
Setting up AWS Network Firewall policies ... 1191
Setting up DNS Firewall policies ... 1195
Setting up Palo Alto Networks Cloud NGFW policies ... 1197
Setting up Fortigate CNF policies ... 1201
Using AWS Firewall Manager policies ... 1205
General settings ... 1206
Creating a policy ... 1206
Deleting a policy ... 1244
Using the policy scope ... 1245
AWS WAF policies ... 1247
AWS Shield Advanced policies ... 1261
Security group policies ... 1267
Network ACL policies ... 1280
Network Firewall policies ... 1288
DNS Firewall policies ... 1299
Palo Alto Networks Cloud NGFW policies ... 1302
Fortigate CNF policies ... 1302
Resource sharing for Network Firewall and DNS Firewall policies ... 1303
Using managed lists ... 1303
Managed list versioning ... 1304

Using managed lists ... 1305
Creating a custom managed list ... 1305
Viewing a managed list ... 1307
Deleting a custom managed list ... 1308
Grouping your resources ... 1310
Considerations when working with resource sets in Firewall Manager ... 1310
Creating resource sets ... 1311
Deleting a resource set ... 1312
Viewing compliance for a policy ... 1312
Firewall Manager integration with Security Hub CSPM ... 1317
AWS WAF policy findings ... 1318
AWS Shield Advanced policy findings ... 1319
Security group common policy findings ... 1320
Security group content audit policy findings ... 1321
Security group usage audit policy findings ... 1321
DNS Firewall policy findings ... 1322
AWS Config findings ... 1322
Security in your use of the Firewall Manager service ... 1323
Data protection ... 1324
Identity and Access Management ... 1325
Logging and monitoring ... 1354
Compliance validation ... 1355
Resilience ... 1355
Infrastructure security ... 1355
AWS Firewall Manager quotas ... 1356
Soft quotas ... 1356
Hard quotas ... 1359
Migrating AWS WAF Classic Web ACLs in Firewall Manager ... 1361
Migrating Web ACLs in AWS WAF Classic Policies ... 1361
Migrating Web ACLs in Shield Advanced Policies ... 1362
Monitoring ... 1364
Monitoring tools ... 1365
Automated monitoring tools ... 1365
Manual tools ... 1366
Monitoring with CloudWatch ... 1367
Viewing metrics and dimensions ... 1368

AWS WAF metrics and dimensions 1369
AWS Shield Advanced metrics 1386
AWS Firewall Manager notifications 1392
Logging API calls with AWS CloudTrail 1392
AWS WAF information in AWS CloudTrail 1393
AWS Shield Advanced information in CloudTrail 1403
AWS Firewall Manager information in CloudTrail 1406
Using the AWS WAF and AWS Shield Advanced API 1409
Using the AWS SDKs 1409
Making HTTPS requests to AWS WAF or Shield Advanced 1409
Request URI 1409
HTTP headers 1409
HTTP request body 1411
HTTP responses 1412
Error responses 1413
Authenticating requests 1413
Related information 1416
FAQs on supported protocols 1418
What are the AWS JSON and Smithy RPC v2 CBOR protocols, and how do they differ? 1418
How do I get started with the Smithy RPC v2 CBOR protocol? 1418
What are the risks of enabling the Smithy RPC v2 CBOR protocol? 1419
What if my solution does not support Smithy RPC v2 CBOR? 1419
What AWS Regions support the Smithy RPC v2 CBOR protocol? 1419
What latency improvements can I expect when upgrading to SDK versions that support Smithy RPC v2 CBOR? 1419
Will the AWS JSON protocol be deprecated? 1419
Where can I find more information about Smithy RPC v2 CBOR? 1420
Document history 1421
Updates before 2018 1479

Introducing a new console experience for AWS WAF

You can now use the updated experience to access AWS WAF functionality anywhere in the console. For more details, see Working with the console.

What are AWS WAF, AWS Shield Advanced, AWS Shield network security director and AWS Firewall Manager?

You can use AWS WAF, AWS Shield, and AWS Firewall Manager together to create a comprehensive security solution. AWS WAF is a web application firewall that you can use to monitor web requests that your end users send to your applications and to control access to your content. Shield Advanced provides protection against distributed denial of service (DDoS) attacks for AWS resources, at the network and transport layers (layer 3 and 4) and the application layer (layer 7). AWS Firewall Manager provides management of protections like AWS WAF and Shield Advanced across accounts and resources, even as new resources are added.

Topics

- What is AWS WAF?
- What is AWS Shield Advanced?
- What is AWS Shield network security director?
- What is AWS Firewall Manager?

What is AWS WAF?

AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to your protected web application resources. You can protect the following resource types:

- Amazon CloudFront distribution
- Amazon API Gateway REST API
- Application Load Balancer
- AWS AppSync GraphQL API
- Amazon Cognito user pool
- AWS App Runner service
- Amazon Bedrock AgentCore Gateway
- AWS Verified Access instance
- AWS Amplify

AWS WAF lets you control access to your content. Based on conditions that you specify, such as the IP addresses that requests originate from or the values of query strings, your protected resource responds to requests either with the requested content, with an HTTP 403 status code (Forbidden), or with a custom response.

At the simplest level, AWS WAF lets you choose one of the following behaviors:

- **Allow all requests except the ones that you specify** – This is useful when you want Amazon CloudFront, Amazon API Gateway, Application Load Balancer, AWS AppSync, Amazon Cognito, AWS App Runner, Amazon Bedrock AgentCore Gateway, or AWS Verified Access to serve content for a public website, but you also want to block requests from attackers.
- **Block all requests except the ones that you specify** – This is useful when you want to serve content for a restricted website whose users are readily identifiable by properties in web requests, such as the IP addresses that they use to browse to the website.
- **Count requests that match your criteria** – You can use the Count action to track your web traffic without modifying how you handle it. You can use this for general monitoring and also to test your new web request handling rules. When you want to allow or block requests based on new properties in the web requests, you can first configure AWS WAF to count the requests that match those properties. This lets you confirm your new configuration settings before you switch your rules to allow or block matching requests.
- **Run CAPTCHA or challenge checks against requests that match your criteria** – You can implement CAPTCHA and silent challenge controls against requests to help reduce bot traffic to your protected resources.

Using AWS WAF has several benefits:

- Additional protection against web attacks using criteria that you specify. You can define criteria using characteristics of web requests such as the following:
  - IP addresses that requests originate from.
  - Country that requests originate from.
  - Values in request headers.
  - Strings that appear in requests, either specific strings or strings that match regular expression (regex) patterns.
  - Length of requests.
  - Presence of SQL code that is likely to be malicious (known as *SQL injection*).

  - Presence of a script that is likely to be malicious (known as *cross-site scripting*).
- Rules that can allow, block, or count web requests that meet the specified criteria. Alternatively, rules can block or count web requests that not only meet the specified criteria, but also exceed a specified number of requests in a minute or in five minutes.
- Rules that you can reuse for multiple web applications.
- Managed rule groups from AWS and AWS Marketplace sellers.
- Real-time metrics and sampled web requests.
- Automated administration using the AWS WAF API.

If you want granular control over the protections that you add to your resources, AWS WAF alone might be the right choice. For more information about AWS WAF, see AWS WAF.

What is AWS Shield Advanced?

You can use AWS WAF web access control lists (web ACLs) to help minimize the effects of a Distributed Denial of Service (DDoS) attack. For additional protection against DDoS attacks, AWS also provides AWS Shield Standard and AWS Shield Advanced. AWS Shield Standard is automatically included at no extra cost beyond what you already pay for AWS WAF and your other AWS services.

Shield Advanced provides expanded DDoS attack protection for your Amazon EC2 instances, Elastic Load Balancing load balancers, CloudFront distributions, Route 53 hosted zones, and AWS Global Accelerator standard accelerators. Shield Advanced incurs additional charges. Shield Advanced options and features include automatic application layer DDoS mitigation, advanced event visibility, and dedicated support from the Shield Response Team (SRT). If you own high visibility websites or are otherwise prone to frequent DDoS attacks, consider purchasing the additional protections that Shield Advanced provides. For additional information, see AWS Shield Advanced capabilities and options and Deciding whether to subscribe to AWS Shield Advanced and apply additional protections.

What is AWS Shield network security director?

AWS Shield network security director helps secure your AWS environment by discovering your compute, networking, and network security resources across your account. network security director evaluates each resource's security configuration by analyzing network topology and

security configurations against AWS best practices and threat intelligence. To help you strengthen your security, network security director rates its findings from low to critical severity and shares specific remediation steps, which you can explore using natural language queries through Amazon Q Developer.

For more information about AWS Shield network security director, see AWS Shield network security director (preview).

What is AWS Firewall Manager?

AWS Firewall Manager simplifies your administration and maintenance tasks across multiple accounts and resources for a variety of protections, including AWS WAF, AWS Shield Advanced, Amazon VPC security groups and network ACLs, AWS Network Firewall, and Amazon Route 53 Resolver DNS Firewall. With Firewall Manager, you set up your protections just once and the service automatically applies them across your accounts and resources, even as you add new accounts and resources.

For more information about Firewall Manager, see AWS Firewall Manager.

Setting up your account to use the services

This topic describes preliminary steps, such as creating an account, to prepare you to use AWS WAF, AWS Firewall Manager, and AWS Shield Advanced. You aren't charged for these preliminary items. You are charged only for AWS services that you use.

Topics

- Sign up for an AWS account
- Download tools

Sign up for an AWS account

To get started with AWS, you need an AWS account. For information about creating an AWS account, see Getting started with an AWS account in the *AWS Account Management Reference Guide*.

Download tools

The AWS Management Console includes a console for AWS WAF, AWS Shield Advanced, and AWS Firewall Manager, but if you want to access the services programmatically, see the following:

- The API guides document the operations that the services support and provide links to the related SDK and CLI documentation:
  - AWS WAF API Reference
  - AWS Shield Advanced API Reference
  - AWS Firewall Manager API Reference
- To call an API without having to handle low-level details like assembling raw HTTP requests, you can use an AWS SDK. The AWS SDKs provide functions and data types that encapsulate the functionality of AWS services. To download an AWS SDK and access installation instructions, see the applicable page:
  - Java
  - JavaScript
  - .NET
  - Node.js

- PHP
- Python
- Ruby

  For a complete list of AWS SDKs, see Tools for Amazon Web Services.

- You can use the AWS Command Line Interface (AWS CLI) to control multiple AWS services from the command line. You can also automate your commands using scripts. For more information, see AWS Command Line Interface.
- AWS Tools for Windows PowerShell supports these AWS services. For more information, see AWS Tools for PowerShell Cmdlet Reference.

AWS WAF

AWS WAF is a web application firewall that lets you monitor the HTTP(S) requests that are forwarded to your protected web application resources. You can protect the following resource types:

- Amazon CloudFront distribution
- Amazon API Gateway REST API
- Application Load Balancer
- AWS AppSync GraphQL API
- Amazon Cognito user pool
- AWS App Runner service
- Amazon Bedrock AgentCore Gateway
- AWS Verified Access instance
- AWS Amplify

AWS WAF lets you control access to your content. Based on criteria that you specify, such as the IP addresses that requests originate from or the values of query strings, the service associated with your protected resource responds to requests either with the requested content, with an HTTP 403 status code (Forbidden), or with a custom response.

> **Note**
>
> You can also use AWS WAF to protect your applications that are hosted in Amazon Elastic Container Service (Amazon ECS) containers. Amazon ECS is a highly scalable, fast container management service that makes it easy to run, stop, and manage Docker containers on a cluster. To use this option, you configure Amazon ECS to use an Application Load Balancer that is enabled for AWS WAF to route and protect HTTP(S) layer 7 traffic across the tasks in your service. For more information, see Service Load Balancing in the *Amazon Elastic Container Service Developer Guide*.

Topics

- Get started with AWS WAF

- How AWS WAF works
- Configuring protection in AWS WAF
- AWS WAF rules
- AWS WAF rule groups
- Web ACL capacity units (WCUs) in AWS WAF
- Oversize web request components in AWS WAF
- Supported regular expression syntax in AWS WAF
- IP sets and regex pattern sets in AWS WAF
- Customized web requests and responses in AWS WAF
- Web request labeling in AWS WAF
- Intelligent threat mitigation in AWS WAF
- AI traffic monetization
- Data protection and logging for AWS WAF protection pack (web ACL) traffic
- Testing and tuning your AWS WAF protections
- Using AWS WAF with Amazon CloudFront
- Security in your use of the AWS WAF service
- AWS WAF quotas
- Migrating your AWS WAF Classic resources to AWS WAF

Get started with AWS WAF

Getting started with AWS WAF depends on which console experience you use. Both experiences provide access to the same core AWS WAF functionality, but they differ in how you configure and manage your web application protections.

AWS WAF offers two options for using the console:

The **new console** aims to simplify web ACL configuration process required by standard console workflows. You can use guided workflows to simplify the web ACL creation and management process through a protection pack. A protection pack makes it easier to use and manage web ACLs in the console, but is not functionally different from a web ACL. In addition to the improved protection configuration process, the new console offers enhanced visibility into your protections

through security dashboards, making it easier to monitor your security posture within the AWS WAF console.

The **standard AWS WAF console** provides a traditional approach to configuring web application firewall protections using web ACLs. It offers granular control over individual rules and rule groups and is familiar to existing AWS WAF users. With this console, you have detailed control over your protection configurations, allowing for precise customization of your security settings.

> **Tip**
>
> Choose the console experience that best fits your needs. If you're new to AWS WAF or want to begin configuring protections based on AWS recommendations, we recommend starting with the new console experience. However, the standard experience is always available to open from the navigation pane in the console.

The following sections provide getting started guidance for both console experiences. Review each approach and select the one that best aligns with your security requirements and operational preferences:

Topics

- Getting started with AWS WAF using the new console experience
- Getting started with AWS WAF using the standard console experience

Getting started with AWS WAF using the new console experience

This section guides you through setting up AWS WAF using the new new console experience, which provides simplified configuration workflows and enhanced security management capabilities.

Access the new console experience

To access the new AWS WAF console experience:

Sign in to the new AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2-pro.

- In the navigation pane, locate and select **Try the new experience**.

> **Note**
>
> You can switch between console experiences at any time using the link in the navigation pane.

Get started with a protection pack (web ACL)

This tutorial shows you how to create and configure a protection pack (web ACL) to protect your applications. Protection packs (Web ACLs) provide pre-configured security rules tailored to specific workload types.

In this tutorial, you'll learn how to:

- Create a protection pack (web ACL)
- Configure application-specific protection settings
- Add AWS resources to protect
- Choose and customize rules
- Configure logging and monitoring

> **Note**
>
> AWS typically bills you less than US $0.25 per day for the resources that you create during this tutorial. When you're finished, we recommend that you delete the resources to prevent incurring unnecessary charges.

Step 1: Set up AWS WAF

If you haven't already followed the general setup steps in Setting up your account to use the services, do that now.

Step 2: Create a protection pack (web ACL)

In this step, you'll create a protection pack (web ACL) and configure its basic settings to match your application type.

1. Sign in to the new AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2-pro.
2. In the navigation pane, choose **Resources & protection packs (web ACLs)**.
3. On the **Resources & protection packs (web ACLs)** page, choose **Add protection pack (web ACL)**.
4. Under **Tell us about your app**, for **App category**, select one or more app categories that best describe your application.
5. For **Traffic source**, choose the type of traffic your application handles:

   - **API** - For API-only applications
   - **Web** - For web-only applications
   - **Both API and Web** - For applications that handle both types of traffic

Step 3: Add resources to protect

Now you'll specify which AWS resources to protect with your protection pack (web ACL).

1. Under **Resources to protect**, choose **Add resources**.
2. Choose the category of AWS resource to associate with this protection pack (web ACL):

   - Amazon CloudFront distributions
   - Regional resources

   For more information about resource types, see Associating protection with an AWS resource.

Step 4: Choose initial protections

In this step, you'll select the rules for your protection pack (web ACL). For first-time users, we recommend choosing the **Recommended** option.

AWS WAF generates **Recommended** for you based on your selections in the **Tell us about your app** section. These packs implement security best practices for your application type.

- Choose **Next** to continue with the protection pack (web ACL) setup.

> **Note**
>
> If you're interested in creating custom rules or using the **You build it** option, we recommend first gaining experience with the pre-configured options. For more information about creating custom protection packs (web ACLs) and rules, see Creating a protection pack (web ACL) in AWS WAF.

Step 5: Customize protection pack (web ACL) settings

Now you'll configure additional settings like default actions, rate limits, and logging.

1. Under **Name and description**, enter a name for your protection pack (web ACL). Optionally, enter a description.

   > **Note**
   >
   > You can't change the name after you create the protection pack (web ACL).

2. Under **Customize protection pack (web ACL)**, configure the following settings:

   a. Under **Default rule actions**, choose the default action for requests that don't match any rules. For more information, see Customized web requests and responses in AWS WAF.

   b. Under **Rule configuration**, customize these settings:

   - **Default rate limits** - Set limits to protect against DDoS attacks
   - **IP Addresses** - Configure IP allow/block lists
   - **Country specific origins** - Manage access by country

   c. For **Logging destination**, configure where you want to store logs. For more information, see AWS WAF logging destinations.

3. Review your settings and choose **Add protection pack (web ACL)**.

Step 6: Clean up your resources

You've now successfully completed the tutorial. To prevent your account from accruing additional AWS WAF charges, you should either delete the protection pack (web ACL) you created or modify it to match your production needs.

To delete your protection pack (web ACL)

1. In the navigation pane, choose **Resources & protection packs (web ACLs)**.
2. Select the protection pack (web ACL) you created.
3. Choose the trash icon, then confirm the deletion by typing "delete".

> **Note**
>
> If you plan to use this protection pack (web ACL) in production, instead of deleting it, you should review and adjust the protection settings to match your application's security requirements.

Getting started with AWS WAF using the standard console experience

The AWS WAF console guides you through the process of configuring AWS WAF to block or allow web requests based on criteria that you specify, such as the IP addresses that the requests originate from or values in the requests. In this step, you create a protection pack (web ACL). For more information about AWS WAF protection packs (web ACLs), see Configuring protection in AWS WAF.

This tutorial shows how to use AWS WAF to perform the following tasks:

- Set up AWS WAF.
- Create a web access control list (web ACL) using the wizard in the AWS WAF console.

To create a web ACL

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. From the AWS WAF home page, choose **Create web ACL**.
3. For **Name**, enter the name that you want to use to identify this web ACL.

   > **Note**
   >
   > You can't change the name after you create the web ACL.

4. (Optional) For **Description - optional**, enter a longer description for the web ACL if you want to.

5. For **CloudWatch metric name**, change the default name if applicable. Follow the guidance on the console for valid characters. The name can't contain special characters, white space, or metric names reserved for AWS WAF, including "All" and "Default_Action."

   > **Note**
   >
   > You can't change the CloudWatch metric name after you create the web ACL.

6. For **Resource type**, choose **CloudFront distributions**. The **Region** automatically populates to **Global (CloudFront)** for CloudFront distributions.
7. (Optional) For **Associated AWS resources - optional**, choose **Add AWS resources**. In the dialog box, choose the resources that you want to associate, and then choose **Add**. AWS WAF returns you to the **Describe web ACL and associated AWS resources** page.
8. Choose **Next**.

> **Note**
>
> AWS typically bills you less than US $0.25 per day for the resources that you create during this tutorial. When you're finished with the tutorial, we recommend that you delete the resources to prevent incurring unnecessary charges.

Step 1: Set up AWS WAF

If you haven't already followed the general setup steps in Setting up your account to use the services, do that now.

Step 2: Create a web ACL

The AWS WAF console guides you through the process of configuring AWS WAF to block or allow web requests based on criteria that you specify, such as the IP addresses that the requests originate from or values in the requests. In this step, you create a web ACL. For more information about AWS WAF web ACLs, see Configuring protection in AWS WAF.

To create a web ACL

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.

2. From the AWS WAF home page, choose **Create web ACL**.
3. For **Name**, enter the name that you want to use to identify this web ACL.

   > **Note**
   >
   > You can't change the name after you create the web ACL.

4. (Optional) For **Description - optional**, enter a longer description for the web ACL if you want to.
5. For **CloudWatch metric name**, change the default name if applicable. Follow the guidance on the console for valid characters. The name can't contain special characters, white space, or metric names reserved for AWS WAF, including "All" and "Default_Action."

   > **Note**
   >
   > You can't change the CloudWatch metric name after you create the web ACL.

6. For **Resource type**, choose **CloudFront distributions**. The **Region** automatically populates to **Global (CloudFront)** for CloudFront distributions.
7. (Optional) For **Associated AWS resources - optional**, choose **Add AWS resources**. In the dialog box, choose the resources that you want to associate, and then choose **Add**. AWS WAF returns you to the **Describe web ACL and associated AWS resources** page.
8. Choose **Next**.

Step 3: Add a string match rule

In this step, you create a rule with a string match statement and indicate what to do with matching requests. A string match rule statement identifies strings that you want AWS WAF to search for in a request. Usually, a string consists of printable ASCII characters, but you can specify any character from hexadecimal 0x00 to 0xFF (decimal 0 to 255). In addition to specifying the string to search for, you specify the web request component that you want to search, such as a header, a query string, or the request body.

This statement type operates on a web request component, and requires the following request component settings:

- **Request component** – The part of the web request to inspect, for example, a query string or the body.

  > **Warning**
  >
  > If you inspect the request components **Body**, **JSON body**, **Headers**, or **Cookies**, read about the limitations on how much content AWS WAF can inspect at Oversize web request components in AWS WAF.

  For information about web request components, see Adjusting rule statement settings in AWS WAF.

- **Optional text transformations** – Transformations that you want AWS WAF to perform on the request component before inspecting it. For example, you could transform to lowercase or normalize white space. If you specify more than one transformation, AWS WAF processes them in the order listed. For information, see Using text transformations in AWS WAF.

For additional information about AWS WAF rules, see AWS WAF rules.

To create a string match rule statement

1. On the **Add rules and rule groups** page, choose **Add rules**, **Add my own rules and rule groups**, **Rule builder**, then **Rule visual editor**.

   > **Note**
   >
   > The console provides the **Rule visual editor** and also a **Rule JSON editor**. The JSON editor makes it easy for you to copy configurations between web ACLs and is required for more complex rule sets, like those with multiple levels of nesting.
   > This procedure uses the **Rule visual editor**.

2. For **Name**, enter the name that you want to use to identify this rule.
3. For **Type** choose **Regular rule**.
4. For **If a request** choose **matches the statement**.

   The other options are for the logical rule statement types. You can use them to combine or negate the results of other rule statements.

5. On **Statement**, for **Inspect**, open the dropdown and choose the web request component that you want AWS WAF to inspect. For this example, choose **Single header**.

   When you choose **Single header**, you also specify which header you want AWS WAF to inspect. Enter **`User-Agent`**. This value isn't case sensitive.

6. For **Match type**, choose where the specified string must appear in the `User-Agent` header.

   For this example, choose **Exactly matches string**. This indicates that AWS WAF inspects the user-agent header in each web request for a string that is identical to the string that you specify.

7. For **String to match**, specify a string that you want AWS WAF to search for. The maximum length of **String to match** is 200 characters. If you want to specify a base64-encoded value, you can specify up to 200 characters before encoding.

   For this example, enter **MyAgent**. AWS WAF will inspect the `User-Agent` header in web requests for the value `MyAgent`.

8. Leave **Text transformation** set to **None**.

9. For **Action**, select the action that you want the rule to take when it matches a web request. For this example, choose **Count** and leave the other choices as they are. The count action creates metrics for web requests that match the rule, but doesn't affect whether the request is allowed or blocked. For more information about action choices, see Using rule actions in AWS WAF and Setting rule priority.

10. Choose **Add rule**.

Step 4: Add a AWS Managed Rules rule group

AWS Managed Rules offers a set of managed rule groups for your use, most of which are free of charge to AWS WAF customers. For more information about rule groups, see AWS WAF rule groups. We'll add an AWS Managed Rules rule group to this web ACL.

To add an AWS Managed Rules rule group

1. On the **Add rules and rule groups** page, choose **Add rules**, and then choose **Add managed rule groups**.

2. On the **Add managed rule groups** page, expand the listing for the **AWS managed rule groups**. (You'll also see listings offered for AWS Marketplace sellers. You can subscribe to their offerings and then use them in the same way as for AWS Managed Rules rule groups.)

3. For the rule group that you want to add, do the following:

   a. In the **Action** column, turn on the **Add to web ACL** toggle.

   b. Select **Edit** and, in the rule group's **Rules** listing, open the **Override all rule actions** dropdown and select **Count**. This sets the action for all rules in the rule group to count only. This allows you to see how all of the rules in the rule group behave with your web requests before you put any of them to use.

   c. Choose **Save rule**.

4. In the **Add managed rule groups** page, choose **Add rules**. This returns you to the **Add rules and rule groups** page.

Step 5: Finish your web ACL configuration

When you're done adding rules and rule groups to your web ACL configuration, finish up by managing the priority of the rules in the web ACL and configuring settings like metrics, tagging, and logging.

To finish your web ACL configuration

1. On the **Add rules and rule groups** page, choose **Next**.
2. On the **Set rule priority** page, you can see the processing order for the rules and rule groups in the web ACL. AWS WAF processes them starting from the top of the list. You can change the processing order by moving the rules up or down. To do this, select one in the list and choose **Move up** or **Move down**. For more information about rule priority, see Setting rule priority.
3. Choose **Next**.
4. On the **Configure metrics** page, for **Amazon CloudWatch metrics**, you can see the planned metrics for your rules and rule groups and you can see the web request sampling options. For information about viewing sampled requests, see Viewing a sample of web requests. For information about Amazon CloudWatch metrics, see Monitoring with Amazon CloudWatch.

   You can access summaries of the web traffic metrics on the web ACL's page in the AWS WAF console, under the **Traffic overview** tab. The console dashboards provide near real-time summaries of the web ACL's Amazon CloudWatch metrics. For more information, see Traffic overview dashboards for protection packs (web ACLs).
5. Choose **Next**.
6. On the **Review and create web ACL** page, review your settings, then choose **Create web ACL**.

The wizard returns you to the **web ACL** page, where your new web ACL is listed.

Step 6: Clean up your resources

You've now successfully completed the tutorial. To prevent your account from accruing additional AWS WAF charges, clean up the AWS WAF objects that you created. Alternatively, you can change the configuration to match the web requests that you really want to manage using AWS WAF.

> **Note**
>
> AWS typically bills you less than US $0.25 per day for the resources that you create during this tutorial. When you're finished, we recommend that you delete the resources to prevent incurring unnecessary charges.

To delete the objects that AWS WAF charges for

1. In the **web ACL** page, select your web ACL from the list and choose **Edit**.
2. On the **Associated AWS resources** tab, for each associated resource, select the radio button next to the resource name and then choose **Disassociate**. This disassociates the web ACL from your AWS resources.
3. In each of the following screens, choose **Next** until you return to the **web ACL** page.

   In the **web ACL** page, select your web ACL from the list and choose **Delete**.

Rules and rule statements don't exist outside of rule group and web ACL definitions. If you delete a web ACL, this deletes all individual rules that you've defined in the web ACL. When you remove a rule group from a web ACL, you just remove the reference to it.

How AWS WAF works

You use AWS WAF to control how your protected resources respond to HTTP(S) web requests. You do this by defining a web access control list (web ACL) and then associating it with one or more web application resources that you want to protect. The associated resources forward incoming requests to AWS WAF for inspection by the web ACL.

The **new console** simplifies the web ACL configuration process. It introduces protection packs to streamline setup while maintaining full control over your security rules.

Protection packs are the new location for web ACLs and simplify web ACL management in the console, but they don't change the underlying web ACL functionality. When using the standard console or the API, you'll still work directly with web ACLs.

In your protection pack (web ACL), you create rules to define traffic patterns to look for in requests and to specify the actions to take on matching requests. The action choices include the following:

- Allow the requests to go to the protected resource for processing and response.
- Block the requests.
- Count the requests.
- Run CAPTCHA or challenge checks against requests to verify human users and standard browser use.

AWS WAF components

The following are the central components of AWS WAF:

- **web ACLs** – You use a web access control list (web ACL) to protect a set of AWS resources. You create a web ACL and define its protection strategy by adding rules. Rules define criteria for inspecting web requests and they specify the action to take on requests that match their criteria. You also set a default action for the web ACL that indicates whether to block or allow through any requests that the rules haven't already blocked or allowed. For more information about web ACLs, see Configuring protection in AWS WAF.

  A web ACL is an AWS WAF resource.
- **Protection pack (Web ACL)s** – In the new console, protection packs are the new location for your web ACLs. During setup, you provide information about your apps and resources. AWS WAF reccomends a protection pack tailored to your scenario, and then creates a web ACL that contains rules, rule groups, and actions defined by the protection pack (web ACL) you choose. For more information about protection packs (web ACLs), see Configuring protection in AWS WAF.

  A protection pack (web ACL) is an AWS WAF resource.
- **Rules** – Each rule contains a statement that defines the inspection criteria, and an action to take if a web request meets the criteria. When a web request meets the criteria, that's a match. You can configure rules to block matching requests, allow them through, count them, or run bot controls against them that use CAPTCHA puzzles or silent client browser challenges. For more information about rules, see AWS WAF rules.

A rule is not an AWS WAF resource. It only exists in the context of a protection pack (web ACL) or rule group.

- **Rule groups** – You can define rules directly inside a protection pack (web ACL) or in reusable rule groups. AWS Managed Rules and AWS Marketplace sellers provide managed rule groups for your use. You can also define your own rule groups. For more information about rule groups, see AWS WAF rule groups.

  A rule group is an AWS WAF resource.

- **web ACL capacity units (WCUs)** – AWS WAF uses WCUs to calculate and control the operating resources that are required to run your rules, rule groups, protection packs (web ACLs), or web ACLs.

  A WCU is not an AWS WAF resource. It only exists in the context of a protection pack (web ACL), rule, or rule group.

AI traffic monetization – AWS WAF enables content and API providers to charge AI bots and agents for access to protected resources. By configuring monetization on your web ACL, you define pricing per request. Payments are verified and settled at the edge using an open payment protocol, with funds delivered directly to your configured wallet. For more information, see AI traffic monetization.

Resources that you can protect with AWS WAF

You can use an AWS WAF protection pack (web ACL) to protect global or regional resource types. You do this by associating the protection pack (web ACL) with the resources that you want to protect. The protection pack (web ACL) and any AWS WAF resources that it uses must be located in the Region where the associated resource is located. For Amazon CloudFront distributions, this is set to US East (N. Virginia).

Amazon CloudFront distributions

You can associate an AWS WAF protection pack (web ACL) with a CloudFront distribution using the AWS WAF console or APIs. You can also associate a protection pack (web ACL) with a CloudFront distribution when you create or update the distribution itself. To configure an association in AWS CloudFormation, you must use the CloudFront distribution configuration. For information about Amazon CloudFront, see Using AWS WAF to Control Access to Your Content in the *Amazon CloudFront Developer Guide*.

AWS WAF is available globally for CloudFront distributions, but you must use the Region US East (N. Virginia) to create your protection pack (web ACL) and any resources used in the protection pack (web ACL), such as rule groups, IP sets, and regex pattern sets. Some interfaces offer a region choice of "Global (CloudFront)". Choosing this is identical to choosing Region US East (N. Virginia) or "us-east-1".

Regional resources

You can protect regional resources in all Regions where AWS WAF is available. You can see the list at AWS WAF endpoints and quotas in the *Amazon Web Services General Reference*.

You can use AWS WAF to protect the following regional resource types:

- Amazon API Gateway REST API
- Application Load Balancer
- AWS AppSync GraphQL API
- Amazon Cognito user pool
- AWS App Runner service
- Amazon Bedrock AgentCore Gateway
- AWS Verified Access instance
- AWS Amplify

You can only associate a protection pack (web ACL) to an Application Load Balancer that's within AWS Regions. For example, you cannot associate a protection pack (web ACL) to an Application Load Balancer that's on AWS Outposts.

You must create any protection pack (web ACL) that you want to associate with an Amplify app in the Global CloudFront Region. You might already have a Regional protection pack (web ACL) in your AWS account, but they are not compatible with Amplify.

The protection pack (web ACL) and any other AWS WAF resources that it uses must be located in the same Region as the protected resources. When monitoring and managing web requests for a protected regional resource, AWS WAF keeps all data in the same Region as the protected resource.

Restrictions on multiple resource associations

You can associate a single protection pack (web ACL) with one or more AWS resources, with the following restrictions:

- You can associate each AWS resource with only one protection pack (web ACL). The relationship between protection pack (web ACL) and AWS resources is one-to-many.
- You can associate a protection pack (web ACL) with one or more CloudFront distributions. You cannot associate a protection pack (web ACL) that you have associated with a CloudFront distribution with any other AWS resource type.

Working with the updated console experience

AWS WAF offers 2 options for using the console:

The **new console** aims to simplify web ACL configuration process required by standard console workflows. You can use guided workflows to simplify the web ACL creation and management process through a protection pack (web ACL). A protection pack (web ACL) makes it easier to use and manage web ACLs in the console, but is not functionally different from a web ACL. In addition to the improved protection configuration process, the new console offers enhanced visibility into your protections through security dashboards, making it easier to monitor your security posture within the AWS WAF console.

The **standard AWS WAF console** provides a traditional approach to configuring web application firewall protections using web ACLs. It offers granular control over individual rules and rule groups and is familiar to existing AWS WAF users. With this console, you have detailed control over your protection configurations, allowing for precise customization of your security settings.

> **Tip**
>
> Choose the console experience that best fits your needs. If you're new to AWS WAF or want to begin configuring protections based on AWS recommendations, we recommend starting with the new console experience. However, the standard experience is always available to open from the navigation pane in the console.

Feature parity between the new and standard console experience

The new console experience maintains complete feature parity with the existing console while introducing new capabilities:

- All existing AWS WAF functionality remains available

- Enhanced visibility through unified dashboards
- Simplified configuration workflows
- New protection pack (web ACL) templates

> **Important**
>
> The new console experience uses the same WAFv2 APIs as the existing console. This means that protection packs created in the new console are implemented as standard WAFv2 web ACLs at the API level.

Key differences

Comparison of Console Experiences

Feature	Previous AWS WAF console experience	Updated console experience
Configuration process	Multi-page workflow	Single-page interface
Rule configuration	Individual rule creation	Option for pre-configured protection packs
Monitoring	Separate dashboards	Unified visibility including AI Traffic Analysis

Understanding the new dashboards

Dashboards available through new provide unified visibility into your security posture through these visualizations:

Traffic insight recommendations – AWS Threat Intelligence monitors your previous 2 weeks of allowed traffic, analyzes vulnerabilities, and provides the following:

- Traffic-based rule suggestions
- Application-specific security recommendations
- Protection optimization guidance

Summary – Shows request counts for all traffic during a specified time range. You can use the following criteria to filter traffic data:

- **Rule** – Filter by the individual rules in the protection pack.
- **Actions** – Show counts for specific actions taken on traffic like Allow, Block, Captcha, and Challenge.
- **Traffic type** – Only show counts for specific traffic types like anti-DDoS or bots.
- **Time range** – Choose from a selection of predefined time ranges, or set a custom range.
- **Local or UTC time** – You can set your preferred time format.

AI Traffic Analysis – Provides comprehensive visibility into AI bot and agent activity:

- **Bot identification** – Bot names, organizations, and verification status.
- **Intent analysis** – Purpose and behavior patterns of AI agents.
- **Access patterns** – Most frequently accessed URLs and endpoints.
- **Temporal trends** – Activity patterns by time of day and historical trends (0-14 days).
- **Traffic characteristics** – Volume, distribution, and anomaly detection for AI traffic.

Protection activity – Visualizes your protection rules and how their order contributes to terminating actions.

- **Traffic flow through your rules** – Show the traffic flow through your rules. Switch from **Sequential rules view** to **Non-sequential rules view** to see how rule order affects outcomes.
- **Rule actions and their outcomes** – Shows the terminating actions that a rule took on traffic in the specified time period.

Action totals – A chart that visualizes the total number of actions taken on requests during a specified time range. Use the **Overlay last 3 hours** option to compare the current time range with the previous 3 hour time window. You can filter data by:

- **Allow action**
- **Total actions**
- **Captcha actions**
- **Challenge actions**
- **Block actions**

All rules – A chart that visualizes metrics for all rules in the protection pack.

- Use the **Overlay last 3 hours** option to compare the current time range with the previous 3 hour time window.

Overview Dashboard – Provides a comprehensive, graphical view of your security status, including the following:

- **Traffic characteristics** – See an overview of traffic by origin, attack types, or by the device type of the clients that sent requests.
- **Rule characteristics** – A breakdown of attacks by the 10 most common rules and termnating actions.
- **Bots** – Visualize bot activity, detection, categories, and bot-related signal labels.
- **Anti-DDoS** – An overview of detected and mitigated layer 7 DDoS activity.

Configuring protection in AWS WAF

This page explains what protection packs (web ACLs) are and how they work.

A protection pack (web ACL) gives you fine-grained control over all of the HTTP(S) web requests that your protected resource responds to. You can protect Amazon CloudFront, Amazon API Gateway, Application Load Balancer, AWS AppSync, Amazon Cognito, AWS App Runner, AWS Amplify, Amazon CloudWatch, Amazon Bedrock AgentCore Gateway, and AWS Verified Access resources.

You can use criteria like the following to allow or block requests:

- IP address origin of the request
- Country of origin of the request
- String match or regular expression (regex) match in a part of the request
- Size of a particular part of the request
- Detection of malicious SQL code or scripting

You can also test for any combination of these conditions. You can block or count web requests that not only meet the specified conditions, but also exceed a specified number of requests in a single minute. You can combine conditions using logical operators. You can also run CAPTCHA puzzles and silent client session challenges against requests.

You provide your matching criteria and the action to take on matches in AWS WAF rule statements. You can define rule statements directly inside your protection pack (web ACL) and in reusable rule groups that you use in your protection pack (web ACL). For a full list of the options, see Using rule statements in AWS WAF and Using rule actions in AWS WAF.

When you create a protection pack (web ACL), you specify the types of resources that you want to use it with. For information, see Creating a protection pack (web ACL) in AWS WAF. After you define a protection pack (web ACL), you can associate it with your resources to begin providing protection for them. For more information, see Associating or disassociating protection with an AWS resource.

Note

On some occasions, AWS WAF might encounter an internal error that delays the response to associated AWS resources about whether to allow or block a request. On those occasions, CloudFront typically allows the request or serves the content, while the Regional services typically deny the request and don't serve the content.

Production traffic risk

Before you deploy changes in your protection pack (web ACL) for production traffic, test and tune them in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune your updated rules in count mode with your production traffic before enabling them. For guidance, see Testing and tuning your AWS WAF protections.

Note

Using more than 1,500 WCUs in a protection pack (web ACL) incurs costs beyond the basic protection pack (web ACL) price. For more information, see Web ACL capacity units (WCUs) in AWS WAF and AWS WAF Pricing.

Temporary inconsistencies during updates

When you create or change a protection pack (web ACL) or other AWS WAF resources, the changes take a small amount of time to propagate to all areas where the resources are stored. The propagation time can be from a few seconds to a number of minutes.

The following are examples of the temporary inconsistencies that you might notice during change propagation:

- After you create a protection pack (web ACL), if you try to associate it with a resource, you might get an exception indicating that the protection pack (web ACL) is unavailable.
- After you add a rule group to a protection pack (web ACL), the new rule group rules might be in effect in one area where the protection pack (web ACL) is used and not in another.
- After you change a rule action setting, you might see the old action in some places and the new action in others.
- After you add an IP address to an IP set that is in use in a blocking rule, the new address might be blocked in one area while still allowed in another.

Topics

- Creating a protection pack (web ACL) in AWS WAF
- Editing a protection pack (web ACL) in AWS WAF
- Managing rule group behavior
- Associating or disassociating protection with an AWS resource
- Using protection packs (web ACLs) with rules and rule groups in AWS WAF
- Setting the protection pack (web ACL) default action in AWS WAF
- Considerations for managing body inspection in AWS WAF
- Configuring CAPTCHA, challenge, and tokens in AWS WAF
- Viewing web traffic metrics in AWS WAF
- Deleting a protection pack (web ACL)

Creating a protection pack (web ACL) in AWS WAF

Using the new console

This section provides procedures for creating protection packs (web ACLs) through the new AWS console.

To create a new protection pack (web ACL), use the protection pack (web ACL) creation wizard following the procedure on this page.

> ⚠ **Production traffic risk**
>
> Before you deploy changes in your protection pack (web ACL) for production traffic, test and tune them in a staging or testing environment until you are comfortable with the

potential impact to your traffic. Then test and tune your updated rules in count mode with your production traffic before enabling them. For guidance, see Testing and tuning your AWS WAF protections.

Note

Using more than 1,500 WCUs in a protection pack (web ACL) incurs costs beyond the basic protection pack (web ACL) price. For more information, see Web ACL capacity units (WCUs) in AWS WAF and AWS WAF Pricing.

1. Sign in to the new AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2-pro.
2. In the navigation pane, choose **Resources & protection packs (web ACLs)**.
3. On the **Resources & protection packs (web ACLs)** page, choose **Add protection pack (web ACL)**.
4. Under **Tell us about your app**, for **App category**, select one or more app categories.
5. For **Traffic source**, choose the type of traffic the application engages with; **API**, **Web**, or **Both API and Web**.
6. Under **Resources to protect,** choose **Add resources**.
7. Choose the category of AWS resource that you want to associate with this protection pack (web ACL), either Amazon CloudFront distributions or Regional resources. For more information, see Associating or disassociating protection with an AWS resource.
8. Under **Choose initial protections,** select your preferred protection level: **Recommended**, **Essentials**, or **You build it**.
9. (Optional) If you choose **You build it**, build your rules.

    a. (Optional) If you want to add your own rule, on the **Add rules** page, choose **Custom rule** and then choose **Next**.

        i. Choose the rule type.

        ii. For **Action**, select the action you want the rule to take when it matches a web request. For information on your choices, see Using rule actions in AWS WAF and Using protection packs (web ACLs) with rules and rule groups in AWS WAF.

If you are using the **CAPTCHA** or **Challenge** action, adjust the **Immunity time** configuration as needed for the rule. If you don't specify the setting, the rule inherits it from the protection pack (web ACL). To modify the protection pack (web ACL) immunity time settings, edit the protection pack (web ACL) after you create it. For more information about immunity times, see Setting timestamp expiration and token immunity times in AWS WAF.

> **Note**
>
> You are charged additional fees when you use the CAPTCHA or Challenge rule action in one of your rules or as a rule action override in a rule group. For more information, see AWS WAF Pricing.

If you want to customize the request or response, choose the options for that and fill in the details of your customization. For more information, see Customized web requests and responses in AWS WAF.

If you want to have your rule add labels to matching web requests, choose the options for that and fill in your label details. For more information, see Web request labeling in AWS WAF.

iii. For **Name**, enter the name that you want to use to identify this rule. Don't use names that start with `AWS`, `Shield`, `PreFM`, or `PostFM`. These strings are either reserved or could cause confusion with rule groups that are managed for you by other services.

iv. Enter your rule definition, according to your needs. You can combine rules inside logical `AND` and `OR` rule statements. The wizard guides you through the options for each rule, according to context. For information about your rules options, see AWS WAF rules.

v. Choose **Create rule**.

> **Note**
>
> If you add more than one rule to a protection pack (web ACL), AWS WAF evaluates the rules in the order that they're listed for the protection pack

(web ACL). For more information, see Using protection packs (web ACLs) with rules and rule groups in AWS WAF.

b. (Optional) If you want to add managed rule groups, on the **Add rules** page, choose **AWS-managed rule group** or **AWS Marketplace rule group** and then choose **Next**. Do the following for each managed rule group that you want to add:

   i. On the **Add rules** page, expand the listing for AWS managed rule groups or for the AWS Marketplace seller.

   ii. Choose the version of the rule group.

   iii. To customize how your protection pack (web ACL) uses the rule group, choose **Edit**. The following are common customization settings:

   - Reduce the scope of the web requests that the rule group inspects by adding a scope-down statement in the **Inspection** section. For information about this option, see Using scope-down statements in AWS WAF.
   - Override the rule actions for some or all rules in **Rule overrides**. If you don't define an override action for a rule, the evaluation uses the rule action that's defined inside the rule group. For information about this option, see Overriding rule group actions in AWS WAF.
   - Some managed rule groups require you to provide additional configuration. See the documentation from your managed rule group provider. For information specific to the AWS Managed Rules rule groups, see AWS Managed Rules for AWS WAF.

   iv. Choose **Next**.

c. (Optional) If you want to add your own rule group, on the **Add rules** page, choose **Custom rule group** and then choose **Next**. Do the following for each rule group that you want to add:

   i. For **Name**, enter the name that you want to use for the rule group rule in this protection pack (web ACL). Don't use names that start with `AWS`, `Shield`, `PreFM`, or `PostFM`. These strings are either reserved or could cause confusion with rule groups that are managed for you by other services. See Recognizing rule groups provided by other services.

   ii. Choose your rule group from the list.

iii. (Optional) Under **Rule configuration**, choose a **Rule override**. You can override the rule actions to any valid action setting, the same as you can do for managed rule groups.

iv. (Optional) Under **Add labels**, choose **Add label** and then enter any labels you want to add to requests that match the rule. Rules that are evaluated later in the same protection pack (web ACL) can reference the labels this rule adds.

v. Choose **Create rule**.

10. Under **Name and description**, enter a name for your protection pack (web ACL). Optionally, enter a description.

> **Note**
>
> You can't change the name after you create the protection pack (web ACL).

11. (Optional) Under **Customize protection pack (web ACL)**, configure default rule actions, configurations, and logging destination:

    a. (Optional) Under **Default rule actions**, choose the default action for the protection pack (web ACL). This is the action that AWS WAF takes on a request when the rules in the protection pack (web ACL) don't explicitly take an action. For more information, see Customized web requests and responses in AWS WAF.

    b. (Optional) Under Rule configuration, customize settings for rules in the protection pack (web ACL):

    - **Default rate limits** - Set rate limits to block Denial of Service (DoS) attacts that can affect availability, compromise security, or consume excessive resources. This rule rate blocks requests per IP address that exceed the allowed rate for your application. For more information, see Using rate-based rule statements in AWS WAF
    - **IP Addresses** - Enter IP addresses to block or allow. This setting overrides other rules.
    - **Country specific origins** - Block requests from specified countries or Count all traffic.

    c. For **Logging destination**, configure the logging destination type and the place to store logs. For more information, see AWS WAF logging destinations.

12. (Optional) If you want to monetize AI bot traffic, expand **Monetization configuration** and configure your payment networks, wallet addresses, and base price. You can skip this step and add a MonetizationConfig later.

13. Review your settings and choose **Add protection pack (web ACL)**.

Using the standard console

This section provides procedures for creating web ACLs through the AWS console.

To create a new web ACL, use the web ACL creation wizard following the procedure on this page.

> **Production traffic risk**
>
> Before you deploy changes in your web ACL for production traffic, test and tune them in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune your updated rules in count mode with your production traffic before enabling them. For guidance, see Testing and tuning your AWS WAF protections.

> **Note**
>
> Using more than 1,500 WCUs in a protection pack (web ACL) incurs costs beyond the basic protection pack (web ACL) price. For more information, see Web ACL capacity units (WCUs) in AWS WAF and AWS WAF Pricing.

To create a web ACL

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. Choose **web ACLs** in the navigation pane, and then choose **Create web ACL**.
3. For **Name**, enter the name that you want to use to identify this web ACL.

   > **Note**
   >
   > You can't change the name after you create the web ACL.

4. (Optional) For **Description - optional**, enter a longer description for the web ACL if you want to.

5. For **CloudWatch metric name**, change the default name if applicable. Follow the guidance on the console for valid characters. The name can't contain special characters, white space, or metric names reserved for AWS WAF, including "All" and "Default_Action."

   > **Note**
   >
   > You can't change the CloudWatch metric name after you create the web ACL.

6. Under **Resource type**, choose the category of AWS resource that you want to associate with this web ACL, either Amazon CloudFront distributions or Regional resources. For more information, see Associating or disassociating protection with an AWS resource.
7. For **Region**, if you've chosen a Regional resource type, choose the Region where you want AWS WAF to store the web ACL.

   You only need to choose this option for Regional resource types. For CloudFront distributions, the Region is hard-coded to the US East (N. Virginia) Region, `us-east-1`, for Global (CloudFront) applications.
8. (CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway) For **Web request inspection size limit - optional**, if you want to specify a different body inspection size limit, select the limit. Inspecting body sizes over the default of 16 KB can incur additional costs. For information about this option, see Considerations for managing body inspection in AWS WAF.
9. (Optional) For **Associated AWS resources - optional**, if you want to specify your resources now, choose **Add AWS resources**. In the dialog box, choose the resources that you want to associate, and then choose **Add**. AWS WAF returns you to the **Describe web ACL and associated AWS resources** page.

   > **Note**
   >
   > When you choose to associate an Application Load Balancer with your web ACL, **Resource-level DDoS protection** is enabled. For more information, see AWS WAF Distributed Denial of Service (DDoS) prevention.

10. (Optional) If you want to monetize AI bot traffic, expand **Monetization configuration** and configure your payment networks, wallet addresses, and base price. You can skip this step and add a MonetizationConfig later.
11. Choose **Next**.

12. (Optional) If you want to add managed rule groups, on the **Add rules and rule groups** page, choose **Add rules**, and then choose **Add managed rule groups**. Do the following for each managed rule group that you want to add:

    a. On the **Add managed rule groups** page, expand the listing for AWS managed rule groups or for the AWS Marketplace seller of your choice.

    b. For the rule group that you want to add, in the **Action** column, turn on the **Add to web ACL** toggle.

    To customize how your web ACL uses the rule group, choose **Edit**. The following are common customization settings:

    - Override the rule actions for some or all rules. If you don't define an override action for a rule, the evaluation uses the rule action that's defined inside the rule group. For information about this option, see Overriding rule group actions in AWS WAF.
    - Reduce the scope of the web requests that the rule group inspects by adding a scope-down statement. For information about this option, see Using scope-down statements in AWS WAF.
    - Some managed rule groups require you to provide additional configuration. See the documentation from your managed rule group provider. For information specific to the AWS Managed Rules rule groups, see AWS Managed Rules for AWS WAF.

    When you're finished with your settings, choose **Save rule**.

    Choose **Add rules** to finish adding managed rules and return to the **Add rules and rule groups** page.

    > **Note**
    >
    > If you add more than one rule to a web ACL, AWS WAF evaluates the rules in the order that they're listed for the web ACL. For more information, see Using protection packs (web ACLs) with rules and rule groups in AWS WAF.

13. (Optional) If you want to add your own rule group, on the **Add rules and rule groups** page, choose **Add rules**, and then choose **Add my own rules and rule groups**. Do the following for each rule group that you want to add:

a. On the **Add my own rules and rule groups** page, choose **Rule group**.

b. For **Name**, enter the name that you want to use for the rule group rule in this web ACL. Don't use names that start with `AWS`, `Shield`, `PreFM`, or `PostFM`. These strings are either reserved or could cause confusion with rule groups that are managed for you by other services. See Recognizing rule groups provided by other services.

c. Choose your rule group from the list.

> **Note**
>
> If you want to override the rule actions for a rule group of your own, first save it to the web ACL, and then edit the web ACL and the rule group reference statement in the web ACL's rule listing. You can override the rule actions to any valid action setting, the same as you can do for managed rule groups.

d. Choose **Add rule**.

14. (Optional) If you want to add your own rule, on the **Add rules and rule groups** page, choose **Add rules**, **Add my own rules and rule groups**, **Rule builder**, then **Rule visual editor**.

> **Note**
>
> The console **Rule visual editor** supports one level of nesting. For example, you can use a single logical `AND` or `OR` statement and nest one level of other statements inside it, but you can't nest logical statements within logical statements. To manage more complex rule statements, use the **Rule JSON editor**. For information about all options for rules, see AWS WAF rules.
> This procedure covers the **Rule visual editor**.

a. For **Name**, enter the name that you want to use to identify this rule. Don't use names that start with `AWS`, `Shield`, `PreFM`, or `PostFM`. These strings are either reserved or could cause confusion with rule groups that are managed for you by other services.

b. Enter your rule definition, according to your needs. You can combine rules inside logical `AND` and `OR` rule statements. The wizard guides you through the options for each rule, according to context. For information about your rules options, see AWS WAF rules.

c. For **Action**, select the action you want the rule to take when it matches a web request. For information on your choices, see Using rule actions in AWS WAF and Using protection packs (web ACLs) with rules and rule groups in AWS WAF.

If you are using the **CAPTCHA** or **Challenge** action, adjust the **Immunity time** configuration as needed for the rule. If you don't specify the setting, the rule inherits it from the web ACL. To modify the web ACL immunity time settings, edit the web ACL after you create it. For more information about immunity times, see Setting timestamp expiration and token immunity times in AWS WAF.

> **Note**
>
> You are charged additional fees when you use the CAPTCHA or Challenge rule action in one of your rules or as a rule action override in a rule group. For more information, see AWS WAF Pricing.

If you want to customize the request or response, choose the options for that and fill in the details of your customization. For more information, see Customized web requests and responses in AWS WAF.

If you want to have your rule add labels to matching web requests, choose the options for that and fill in your label details. For more information, see Web request labeling in AWS WAF.

d. Choose **Add rule**.

15. Choose the default action for the web ACL, either Block or Allow. This is the action that AWS WAF takes on a request when the rules in the web ACL don't explicitly allow or block it. For more information, see Setting the protection pack (web ACL) default action in AWS WAF.

    If you want to customize the default action, choose the options for that and fill in the details of your customization. For more information, see Customized web requests and responses in AWS WAF.

16. You can define a **Token domain list** to enable token sharing between protected applications. Tokens are used by the CAPTCHA and Challenge actions and by the application integration SDKs that you implement when you use the AWS Managed Rules

rule groups for AWS WAF Fraud Control account creation fraud prevention (ACFP), AWS WAF Fraud Control account takeover prevention (ATP), and AWS WAF Bot Control.

Public suffixes aren't allowed. For example, you can't use `gov.au` or `co.uk` as a token domain.

By default, AWS WAF accepts tokens only for the domain of the protected resource. If you add token domains in this list, AWS WAF accepts tokens for all domains in the list and for the domain of the associated resource. For more information, see AWS WAF protection pack (web ACL) token domain list configuration.

17. Choose **Next**.
18. In the **Set rule priority** page, select and move your rules and rule groups to the order that you want AWS WAF to process them. AWS WAF processes rules starting from the top of the list. When you save the web ACL AWS WAF assigns numeric priority settings to the rules, in the order that you have them listed. For more information, see Setting rule priority.
19. Choose **Next**.
20. In the **Configure metrics** page, review the options and apply any updates that you need. You can combine metrics from multiple sources by providing the same **CloudWatch metric name** for them.
21. Choose **Next**.
22. In the **Review and create web ACL** page, check over your definitions. If you want to change any area, choose **Edit** for the area. This returns you to the page in the web ACL wizard. Make any changes, then choose **Next** through the pages until you come back to the **Review and create web ACL** page.
23. Choose **Create web ACL**. Your new web ACL is listed in the **web ACLs** page.

Editing a protection pack (web ACL) in AWS WAF

Using the new console

This section provides procedures for editing protection packs (web ACLs) through the AWS console.

To add or remove rules from a protection pack (web ACL) or change configuration settings, access the protection pack (web ACL) using the procedure on this page. While updating a

protection pack (web ACL), AWS WAF provides continuous coverage to the resources that you have associated with the protection pack (web ACL).

Production traffic risk

Before you deploy changes in your protection pack (web ACL) for production traffic, test and tune them in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune your updated rules in count mode with your production traffic before enabling them. For guidance, see Testing and tuning your AWS WAF protections.

Note

Using more than 1,500 WCUs in a protection pack (web ACL) incurs costs beyond the basic protection pack (web ACL) price. For more information, see Web ACL capacity units (WCUs) in AWS WAF and AWS WAF Pricing.

To edit a protection pack (web ACL)

1. Sign in to the new AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2-pro.
2. In the navigation pane, choose **Resources & protection packs (web ACLs)**.
3. Choose the protection pack (web ACL) that you want to edit. The console makes the main protection pack (web ACL) card editable, and also opens a side pane with details you can edit.
4. Edit the protection pack (web ACL) as needed.

   The following lists the editable protection pack (web ACL) configuration components.

   This section provides procedures for editing web ACLs through the AWS console.

   To add or remove rules from a web ACL or change configuration settings, access the web ACL using the procedure on this page. While updating a web ACL, AWS WAF provides continuous coverage to the resources that you have associated with the web ACL.

Production traffic risk

Before you deploy changes in your web ACL for production traffic, test and tune them in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune your updated rules in count mode with your production traffic before enabling them. For guidance, see Testing and tuning your AWS WAF protections.

Note

Using more than 1,500 WCUs in a protection pack (web ACL) incurs costs beyond the basic protection pack (web ACL) price. For more information, see Web ACL capacity units (WCUs) in AWS WAF and AWS WAF Pricing.

Temporary inconsistencies during updates

When you create or change a protection pack (web ACL) or other AWS WAF resources, the changes take a small amount of time to propagate to all areas where the resources are stored. The propagation time can be from a few seconds to a number of minutes.

The following are examples of the temporary inconsistencies that you might notice during change propagation:

- After you create a protection pack (web ACL), if you try to associate it with a resource, you might get an exception indicating that the protection pack (web ACL) is unavailable.
- After you add a rule group to a protection pack (web ACL), the new rule group rules might be in effect in one area where the protection pack (web ACL) is used and not in another.
- After you change a rule action setting, you might see the old action in some places and the new action in others.
- After you add an IP address to an IP set that is in use in a blocking rule, the new address might be blocked in one area while still allowed in another.

Using the standard console

This section provides procedures for editing web ACLs through the AWS console.

To add or remove rules from a web ACL or change configuration settings, access the web ACL using the procedure on this page. While updating a web ACL, AWS WAF provides continuous coverage to the resources that you have associated with the web ACL.

Production traffic risk

Before you deploy changes in your web ACL for production traffic, test and tune them in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune your updated rules in count mode with your production traffic before enabling them. For guidance, see Testing and tuning your AWS WAF protections.

Note

Using more than 1,500 WCUs in a protection pack (web ACL) incurs costs beyond the basic protection pack (web ACL) price. For more information, see Web ACL capacity units (WCUs) in AWS WAF and AWS WAF Pricing.

To edit a web ACL

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **web ACLs**.
3. Choose the name of the web ACL that you want to edit. The console takes you to the web ACL's description.
4. Edit the web ACL as needed. Select the tabs for the configuration areas that you're interested in and edit the mutable settings. For each setting that you edit, when you choose **Save** and return to the web ACL's description page, the console saves your changes to the web ACL.

   The following lists the tabs that contain web ACL configuration components.

   - **Rules** tab
     - **Rules defined in the web ACL** – You can edit and manage the rules that you have defined in the web ACL, similar to how you did during web ACL creation.

> **Note**
>
> Don't change the names of any rules that you didn't add by hand to your web ACL. If you are using other services to manage rules for you, changing their names could remove or lessen their ability to provide the intended protections. AWS Shield Advanced and AWS Firewall Manager both can create rules in your web ACL. For information, see Recognizing rule groups provided by other services.

> **Note**
>
> If you change the name of a rule and you want the rule's metric name to reflect the change, you must update the metric name as well. AWS WAF doesn't automatically update the metric name for a rule when you change the rule name. You can change the metric name when you edit the rule in the console, by using the rule JSON editor. You can also change both names through the APIs and in any JSON listing that you use to define your protection pack (web ACL) or rule group.

For information about rules and rule group settings, see AWS WAF rules and AWS WAF rule groups.

- **web ACL rule capacity units used** – The current capacity usage for your web ACL. This is view only.
- **Default web ACL action for requests that don't match any rules**– For information about this setting, see Setting the protection pack (web ACL) default action in AWS WAF.
- **web ACL CAPTCHA and challenge configurations** – These immunity times determine how long a CAPTCHA or challenge token remains valid after it's acquired. You can only modify this setting here, after you create the web ACL. For information about these settings, see Setting timestamp expiration and token immunity times in AWS WAF.
- **Token domain list** – AWS WAF accepts tokens for all domains in the list and for the domain of the associated resource. For more information, see AWS WAF protection pack (web ACL) token domain list configuration.

- **Associated AWS resources** tab
  - **Web request inspection size limit** – Included only for web ACLs that protect CloudFront distributions. The body inspection size limit determines how much of the body component is forwarded to AWS WAF for inspection. For more information about this setting, see Considerations for managing body inspection in AWS WAF.
  - **Associated AWS resources** – The list of resources that the web ACL is currently associated with and protecting. You can locate resources that are within the same Region as the web ACL and associate them to the web ACL. For more information, see Associating or disassociating protection with an AWS resource.
- **Custom response bodies** tab
  - Custom response bodies that are available for use by your web ACL rules that have the action set to Block. For more information, see Sending custom responses for Block actions.
- **Logging and metrics** tab
  - **Logging** – Logging for the traffic that the web ACL evaluates. For information, see Logging AWS WAF protection pack (web ACL) traffic.
  - **Security Lake integration** – The status of any data collection that you've configured for the web ACL in Amazon Security Lake. For information, see Collecting data from AWS services in the *Amazon Security Lake user guide*.
  - **Sampled requests** – Information about the rules that match web requests. For information about viewing sampled requests, see Viewing a sample of web requests.
  - **Data protection settings** – You can configure web traffic data redaction and filtering for all data that's available for the web ACL and for just the data that the AWS WAF sends to the configured web ACL logging destination. For information about data protection, see Data protection and logging for AWS WAF protection pack (web ACL) traffic.
  - **CloudWatch metrics** – Metrics for the rules in your web ACL. For information about Amazon CloudWatch metrics, see Monitoring with Amazon CloudWatch.

Temporary inconsistencies during updates

When you create or change a protection pack (web ACL) or other AWS WAF resources, the changes take a small amount of time to propagate to all areas where the resources are stored. The propagation time can be from a few seconds to a number of minutes.

The following are examples of the temporary inconsistencies that you might notice during change propagation:

- After you create a protection pack (web ACL), if you try to associate it with a resource, you might get an exception indicating that the protection pack (web ACL) is unavailable.
- After you add a rule group to a protection pack (web ACL), the new rule group rules might be in effect in one area where the protection pack (web ACL) is used and not in another.
- After you change a rule action setting, you might see the old action in some places and the new action in others.
- After you add an IP address to an IP set that is in use in a blocking rule, the new address might be blocked in one area while still allowed in another.

Managing rule group behavior

This section describes your options for modifying how you use a rule group in your protection pack (web ACL). This information applies to all rule group types. After you add a rule group to a protection pack (web ACL), you can override the actions of the individual rules in the rule group to Count or to any other valid rule action setting. You can also override the rule group's resulting action to Count, which has no effect on how the rules are evaluated inside the rule group.

For information about these options, see Overriding rule group actions in AWS WAF.

Overriding rule actions in a rule group

For each rule group in a protection pack (web ACL), you can override the contained rule's actions for some or all of the rules.

The most common use case for this is overriding the rule actions to Count to test new or updated rules. If you have metrics enabled, you receive metrics for each rule that you override. For more information about testing, see Testing and tuning your AWS WAF protections.

You can make these changes when you're adding a managed rule group to the protection pack (web ACL), and you can make them to any type of rule group when you edit the protection pack (web ACL). These instructions are for a rule group that has already been added to the protection pack (web ACL). See additional information about this option at Rule group rule action overrides.

Using the new console

To override rule actions in a rule group

1. Choose the protection pack (web ACL) that you want to edit. The console makes the main protection pack (web ACL) card editable, and also opens a side panel with details you can edit.
2. In the protection pack (web ACL) card, choose the **Edit** link next to **Rules** to open the **Manage rules** panel.
3. In the **Manage rules** section for the rule group, choose the managed rule to open its action settings.
   - **Override rule group** – Changes the rule group action to Count mode but keeps all individual rule actions unchanged.
   - **Override all rule actions** – Applies a rule action to all rules, overriding their current state.
   - **Single rule override** – Applies a rule action to an individual rule.
4. When you are finished making your changes, choose **Save rule**.

Using the standard console

To override rule actions in a rule group

1. Edit the web ACL.
2. In the web ACL page **Rules** tab, select the rule group, then choose **Edit**.
3. In the **Rules** section for the rule group, manage the action settings as needed.
   - **All rules** – To set an override action for all rules in the rule group, open the **Override all rule actions** dropdown and select the override action. To remove the overrides for all rules, select **Remove all overrides**.
   - **Single rule** – To set an override action for a single rule, open the rule's dropdown and select the override action. To remove an override for a rule, open the rule's dropdown and select **Remove override**.
4. When you are finished making your changes, choose **Save rule**. The rule action and override action settings are listed in the rule group page.

The following example JSON listing shows a rule group declaration inside a protection pack (web ACL) that overrides the rule actions to Count for the rules `CategoryVerifiedSearchEngine` and `CategoryVerifiedSocialMedia`. In the JSON, you override all rule actions by providing a `RuleActionOverrides` entry for each individual rule.

```
{
    "Name": "AWS-AWSBotControl-Example",
   "Priority": 5,
   "Statement": {
    "ManagedRuleGroupStatement": {
        "VendorName": "AWS",
        "Name": "AWSManagedRulesBotControlRuleSet",
        "RuleActionOverrides": [
          {
            "ActionToUse": {
              "Count": {}
            },
            "Name": "CategoryVerifiedSearchEngine"
          },
          {
            "ActionToUse": {
              "Count": {}
            },
            "Name": "CategoryVerifiedSocialMedia"
          }
        ],
        "ExcludedRules": []
    },
   "VisibilityConfig": {
       "SampledRequestsEnabled": true,
       "CloudWatchMetricsEnabled": true,
       "MetricName": "AWS-AWSBotControl-Example"
   }
}
```

Overriding a rule group's evaluation result to Count

You can override the action that results from a rule group evaluation, without altering how the rules in the rule group are configured or evaluated. This option is not commonly used. If any rule in the rule group results in a match, this override sets the resulting action from the rule group to Count.

> **Note**
>
> This is an uncommon use case. Most action overrides are done at the rule level, inside the rule group, as described in Overriding rule actions in a rule group.

You can override the rule group's resulting action in the protection pack (web ACL) when you add or edit the rule group. In the console, open the rule group's **Override rule group action - optional** pane and enable the override. In the JSON set `OverrideAction` in the rule group statement, as shown in the following example listing:

```
{
   "Name": "AWS-AWSBotControl-Example",
   "Priority": 5,
   "Statement": {
    "ManagedRuleGroupStatement": {
     "VendorName": "AWS",
     "Name": "AWSManagedRulesBotControlRuleSet"
     }
   },
    "OverrideAction": {
       "Count": {}
    },
   "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "AWS-AWSBotControl-Example"
   }
}
```

Associating or disassociating protection with an AWS resource

You can use AWS WAF to create the following associations between protection packs (web ACLs) and your resources:

- Associate a regional protection pack (web ACL) with any of the regional resources listed below. For this option, the protection pack (web ACL) must be in the same region as your resource.
  - Amazon API Gateway REST API
  - Application Load Balancer
  - AWS AppSync GraphQL API

- Amazon Cognito user pool
- AWS App Runner service
- Amazon Bedrock AgentCore Gateway
- AWS Verified Access instance
- AWS Amplify

- Associate a global protection pack (web ACL) with a Amazon CloudFront distribution. The global protection pack (web ACL) will have a hard-coded Region of US East (N. Virginia) Region.

You can also associate a protection pack (web ACL) with a CloudFront distribution when you create or update the distribution itself. For information, see Using AWS WAF to Control Access to Your Content in the *Amazon CloudFront Developer Guide*.

Restrictions on multiple associations

You can associate a single protection pack (web ACL) with one or more AWS resources, according to the following restrictions:

- You can associate each AWS resource with only one protection pack (web ACL). The relationship between protection pack (web ACL) and AWS resources is one-to-many.
- You can associate a protection pack (web ACL) with one or more CloudFront distributions. You cannot associate a protection pack (web ACL) that you have associated with a CloudFront distribution with any other AWS resource type.

Additional restrictions

The following additional restrictions apply to protection pack (web ACL) associations:

- You can only associate a protection pack (web ACL) to an Application Load Balancer within AWS Regions. For example, you cannot associate a protection pack (web ACL) to an Application Load Balancer that is on AWS Outposts.
- You can't associate an Amazon Cognito user pool with a protection pack (web ACL) that uses the AWS WAF Fraud Control account creation fraud prevention (ACFP) managed rule group `AWSManagedRulesACFPRuleSet` or the AWS WAF Fraud Control account takeover prevention (ATP) managed rule group `AWSManagedRulesATPRuleSet`. For information about account creation fraud prevention, see AWS WAF Fraud Control account creation fraud prevention

(ACFP). For information about account takeover prevention, see AWS WAF Fraud Control account takeover prevention (ATP).

> **Production traffic risk**
>
> Before you deploy your protection pack (web ACL) for production traffic, test and tune it in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune your rules in count mode with your production traffic before enabling them. For guidance, see Testing and tuning your AWS WAF protections.

Associating protection with an AWS resource

Using the new console

1. Choose the protection pack (web ACL) that you want to edit. The console makes the main protection pack (web ACL) card editable, and also opens a side panel with details you can edit.
2. In the protection pack (web ACL) card, choose the **Edit** link next to **Resources** to open the **Manage resources** panel.
3. In the **Manage resources** section for the rule group, choose **Add regional resources** or **Add global resources**.
4. Choose resources and then choose **Add**.

Using the standard console

To associate a web ACL with an AWS resource, perform the following procedure.

To associate a web ACL with an AWS resource

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **web ACLs**.
3. Choose the name of the web ACL that you want to associate with a resource. The console takes you to the web ACL's description, where you can edit it.
4. On the **Associated AWS resources** tab, choose **Add AWS resources**.

5. When prompted, choose the resource type, select the radio button next to the resource that you want to associate, and then choose **Add**.

Disassociating a protection from an AWS resource

Using the new console

1. Choose the protection pack (web ACL) that you want to edit. The console makes the main protection pack (web ACL) card editable, and also opens a side panel with details you can edit.
2. In the protection pack (web ACL) card, choose the **Edit** link next to **Resources** to open the **Manage resources** panel.
3. In the **Manage resources** section for the rule group, choose the resource you want to disassociate, and then choose **Disassociate**.

> **Note**
>
> You must disassociate one resource at a time. Do not choose multiple resources.

4. In the confirmation page, type "disassociate", and then choose **Disassociate**.

Using the standard console

To dissociate a web ACL from an AWS resource, perform the following procedure.

To disassociate a web ACL from an AWS resource

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **web ACLs**.
3. Choose the name of the web ACL that you want to disassociate from your resource. The console takes you to the web ACL's description, where you can edit it.
4. On the **Associated AWS resources** tab, select the resource that you want to disassociate this web ACL from.

> **Note**
>
> You must disassociate one resource at a time. Do not choose multiple resources.

> **Note**
>
> When you choose to associate an Application Load Balancer with your webACL, **Resource-level DDoS protection** is enabled. For more information, see AWS WAF Distributed Denial of Service (DDoS) prevention.

5. Choose **Disassociate**. The console opens a confirmation dialogue. Confirm your choice to disassociate the web ACL from the AWS resource.

Using protection packs (web ACLs) with rules and rule groups in AWS WAF

This section introduces how protection packs (web ACLs) and web ACLs work with rules and rule groups.

The way a protection pack (web ACL) handles a web request depends on the following:

- The numeric priority settings of the rules in the protection pack (web ACL) and inside rule groups
- The action settings on the rules and protection pack (web ACL)
- Any overrides that you place on the rules in the rule groups that you add

For a list of the rule action settings, see Using rule actions in AWS WAF.

You can customize request and response handling in your rule action settings and default protection pack (web ACL) action settings. For information, see Customized web requests and responses in AWS WAF.

Topics

- Setting rule priority
- How AWS WAF handles rule and rule group actions

- Overriding rule group actions in AWS WAF

Setting rule priority

This section explains how AWS WAF uses numeric priority settings to set the evaluation order for rules.

In a protection pack (web ACL) and inside any rule group, you determine the evaluation order of the rules using numeric priority settings. You must give each rule in a protection pack (web ACL) a unique priority setting within that protection pack (web ACL), and you must give each rule in a rule group a unique priority setting within that rule group.

> **Note**
>
> When you manage rule groups, protection packs (web ACLs) through the console, AWS WAF assigns unique numeric priority settings for you based on the order of the rules in the list. AWS WAF assigns the lowest numeric priority to the rule at the top of the list, and the highest numeric priority to the rule at the bottom.

When AWS WAF evaluates any rule group, protection pack (web ACL) against a web request, it evaluates the rules from the lowest numeric priority setting on up until it either finds a match that terminates the evaluation or exhausts all of the rules.

For example, say you have the following rules and rule groups in your protection pack (web ACL), prioritized as shown:

- Rule1 – priority 0
- RuleGroupA – priority 100
  - RuleA1 – priority 10,000
  - RuleA2 – priority 20,000
- Rule2 – priority 200
- RuleGroupB – priority 300
  - RuleB1 – priority 0
  - RuleB2 – priority 1

AWS WAF would evaluate the rules for this protection pack (web ACL) in the following order:

- Rule1
- RuleGroupA RuleA1
- RuleGroupA RuleA2
- Rule2
- RuleGroupB RuleB1
- RuleGroupB RuleB2

How AWS WAF handles rule and rule group actions

This section explains how AWS WAF uses rules and rule groups to handle actions.

When you configure your rules and rule groups, you choose how you want AWS WAF to handle matching web requests:

- **Allow and Block are terminating actions** – Allow and Block actions stop all other processing of the protection pack (web ACL) on the matching web request. If a rule in a protection pack (web ACL) finds a match for a request and the rule action is Allow or Block, that match determines the final disposition of the web request for the protection pack (web ACL). AWS WAF doesn't process any other rules in the protection pack (web ACL) that come after the matching one. This is true for rules that you add directly to the protection pack (web ACL) and rules that are inside an added rule group. With the Block action, the protected resource doesn't receive or process the web request.
- **Count is a non-terminating action** – When a rule with a Count action matches a request, AWS WAF counts the request, then continues processing the rules that follow in the protection pack (web ACL) rule set.
- **CAPTCHA and Challenge can be non-terminating or terminating actions** – When a rule with one of these actions matches a request, AWS WAF checks its token status. If the request has a valid token, AWS WAF treats the match similar to a Count match, and then continues processing the rules that follow in the protection pack (web ACL) rule set. If the request doesn't have a valid token, AWS WAF terminates the evaluation and sends the client a CAPTCHA puzzle or silent background client session challenge to solve.

If the rule evaluation doesn't result in any terminating action, then AWS WAF applies the protection pack (web ACL) default action to the request. For information, see Setting the protection pack (web ACL) default action in AWS WAF.

In your protection pack (web ACL), you can override the action settings for rules inside a rule group and you can override the action that's returned by a rule group. For information, see Overriding rule group actions in AWS WAF.

Interaction between actions and priority settings

The actions that AWS WAF applies to a web request are affected by the numeric priority settings of the rules in the protection pack (web ACL). For example, say that your protection pack (web ACL) has a rule with Allow action and a numeric priority of 50 and another rule with Count action and a numeric priority of 100. AWS WAF evaluates the rules in a protection pack (web ACL) in the order of their priority, starting from the lowest setting, so it will evaluate the allow rule before the count rule. A web request that matches both rules will match the allow rule first. Since Allow is a terminating action, AWS WAF will stop the evaluation at this match and won't evaluate the request against the count rule.

- If you only want to include requests that don't match the allow rule in your count rule metrics, then the priority settings of the rules would work.
- On the other hand, if you want count metrics from the count rule even for requests that match the allow rule, you'd need to give the count rule a lower numeric priority setting than the allow rule, so that it runs first.

For more information about priority settings, see Setting rule priority.

Overriding rule group actions in AWS WAF

This section explains how to override rule group actions.

When you add a rule group to your protection pack (web ACL), you can override the actions it takes on matching web requests. Overriding the actions for a rule group inside your protection pack (web ACL) configuration doesn't alter the rule group itself. It only alters how AWS WAF uses the rule group in the context of the protection pack (web ACL).

Rule group rule action overrides

You can override the actions of the rules inside a rule group to any valid rule action. When you do this, matching requests are handled exactly as if the configured rule's action were the override setting.

> **Note**
>
> Rule actions can be terminating or non-terminating. A terminating action stops the protection pack (web ACL) evaluation of the request and either lets it continue to your protected application or blocks it.

Here are the rule action options:

- **Allow** – AWS WAF allows the request to be forwarded to the protected AWS resource for processing and response. This is a terminating action. In rules that you define, you can insert custom headers into the request before forwarding it to the protected resource.
- **Block** – AWS WAF blocks the request. This is a terminating action. By default, your protected AWS resource responds with an HTTP `403 (Forbidden)` status code. In rules that you define, you can customize the response. When AWS WAF blocks a request, the Block action settings determine the response that the protected resource sends back to the client.
- **Count** – AWS WAF counts the request but does not determine whether to allow it or block it. This is a non-terminating action. AWS WAF continues processing the remaining rules in the protection pack (web ACL). In rules that you define, you can insert custom headers into the request and you can add labels that other rules can match against.
- **CAPTCHA and Challenge** – AWS WAF uses CAPTCHA puzzles and silent challenges to verify that the request is not coming from a bot, and AWS WAF uses tokens to track recent successful client responses.

  CAPTCHA puzzles and silent challenges can only run when browsers are accessing HTTPS endpoints. Browser clients must be running in secure contexts in order to acquire tokens.

  > **Note**
  >
  > You are charged additional fees when you use the CAPTCHA or Challenge rule action in one of your rules or as a rule action override in a rule group. For more information, see AWS WAF Pricing.

  These rule actions can be terminating or non-terminating, depending on the state of the token in the request:

- **Non-terminating for valid, unexpired token** – If the token is valid and unexpired according to the configured CAPTCHA or challenge immunity time, AWS WAF handles the request similar to the Count action. AWS WAF continues to inspect the web request based on the remaining rules in the protection pack (web ACL). Similar to the Count configuration, in rules that you define, you can optionally configure these actions with custom headers to insert into the request, and you can add labels that other rules can match against.
- **Terminating with blocked request for invalid or expired token** – If the token is invalid or the indicated timestamp is expired, AWS WAF terminates the inspection of the web request and blocks the request, similar to the Block action. AWS WAF then responds to the client with a custom response code. For CAPTCHA, if the request contents indicate that the client browser can handle it, AWS WAF sends a CAPTCHA puzzle in a JavaScript interstitial, which is designed to distinguish human clients from bots. For the Challenge action, AWS WAF sends a JavaScript interstitial with a silent challenge that is designed to distinguish normal browsers from sessions that are being run by bots.

For additional information, see CAPTCHA and Challenge in AWS WAF.

For information about how to use this option, see Overriding rule actions in a rule group.

Overriding the rule action to Count

The most common use case for rule action overrides is overriding some or all of the rule actions to Count, to test and monitor a rule group's behavior before putting it into production.

You can also use this to troubleshoot a rule group that's generating false positives. False positives occur when a rule group blocks traffic that you aren't expecting it to block. If you identify a rule within a rule group that would block requests that you want to allow through, you can keep the count action override on that rule, to exclude it from acting on your requests.

For more information about using the rule action override in testing, see Testing and tuning your AWS WAF protections.

JSON listing: `RuleActionOverrides` replaces `ExcludedRules`

If you set rule group rule actions to Count in your protection pack (web ACL) configuration before October 27, 2022, AWS WAF saved your overrides in the protection pack (web ACL) JSON as `ExcludedRules`. Now, the JSON setting for overriding a rule to Count is in the `RuleActionOverrides` settings.

We recommend that you update all of your `ExcludedRules` settings in your JSON listings to `RuleActionOverrides` settings with the action set to Count. The API accepts either setting, but you'll get consistency in your JSON listings, between your console work and your API work, if you only use the new `RuleActionOverrides` setting.

> **Note**
>
> In the AWS WAF console, the protection pack (web ACL) **Sampled requests** tab doesn't show samples for rules with the old setting. For more information, see Viewing a sample of web requests.

When you use the AWS WAF console to edit the existing rule group settings, the console automatically converts any `ExcludedRules` settings in the JSON to `RuleActionOverrides` settings, with the override action set to Count.

- Current setting example:

```
       "ManagedRuleGroupStatement": {
          "VendorName": "AWS",
          "Name": "AWSManagedRulesAdminProtectionRuleSet",
          "RuleActionOverrides": [
            {
              "Name": "AdminProtection_URIPATH",
              "ActionToUse": {
                "Count": {}
              }
            }
          ]
```

- Old setting example:

```
OLD SETTING
       "ManagedRuleGroupStatement": {
          "VendorName": "AWS",
          "Name": "AWSManagedRulesAdminProtectionRuleSet",
          "ExcludedRules": [
            {
              "Name": "AdminProtection_URIPATH"
            }
          ]
```

```
OLD SETTING
```

Rule group return action override to Count

You can override the action that the rule group returns, setting it to Count.

> **Note**
>
> This is not a good option for testing the rules in a rule group, because it doesn't alter how AWS WAF evaluates the rule group itself. It only affects how AWS WAF handles results that are returned to the protection pack (web ACL) from the rule group evaluation. If you want to test the rules in a rule group, use the option described in the preceding section, Rule group rule action overrides.

When you override the rule group action to Count, AWS WAF processes the rule group evaluation normally.

If no rules in the rule group match or if all matching rules have a Count action, then this override has no effect on the processing of the rule group or the protection pack (web ACL).

The first rule in the rule group that matches a web request and that has a terminating rule action causes AWS WAF to stop evaluating the rule group and return the terminating action result to the protection pack (web ACL) evaluation level. At this point, in the protection pack (web ACL) evaluation, this override takes effect. AWS WAF overrides the terminating action so that the result of the rule group evaluation is only a Count action. AWS WAF then continues processing the rest of the rules in the protection pack (web ACL).

For information about how to use this option, see Overriding a rule group's evaluation result to Count.

Setting the protection pack (web ACL) default action in AWS WAF

This section explains how protection pack (web ACL) default actions work.

When you create and configure a protection pack (web ACL), you must set the protection pack (web ACL) default action. AWS WAF applies this action to any web request that makes it through all of the protection pack (web ACL)'s rule evaluations without having a terminating action applied to it.

A terminating action stops the protection pack (web ACL) evaluation of the request and either lets it continue to your protected application or blocks it. For information about rule actions, see Using rule actions in AWS WAF.

The protection pack (web ACL) default action must determine the final disposition of the web request, so it's a terminating action:

- **Allow** – If you want to allow most users to access your website, but you want to block access to attackers whose requests originate from specified IP addresses, or whose requests appear to contain malicious SQL code or specified values, choose Allow for the default action. Then, when you add rules to your protection pack (web ACL), add rules that identify and block the specific requests that you want to block. With this action, you can insert custom headers into the request before forwarding it to the protected resource.
- **Block** – If you want to prevent most users from accessing your website, but you want to allow access to users whose requests originate from specified IP addresses, or whose requests contain specified values, choose Block for the default action. Then when you add rules to your protection pack (web ACL), add rules that identify and allow the specific requests that you want to allow in. By default, for the Block action, the AWS resource responds with an HTTP `403 (Forbidden)` status code, but you can customize the response.

For information about customizing requests and responses, see Customized web requests and responses in AWS WAF.

Your configuration of your own rules and rule groups depends in part on whether you want to allow or block most web requests. For example, if you want to *allow* most requests, you would set the protection pack (web ACL) default action to Allow, and then add rules that identify web requests that you want to *block*, such as the following:

- Requests that originate from IP addresses that are making an unreasonable number of requests
- Requests that originate from countries that either you don't do business in or are the frequent source of attacks
- Requests that include fake values in the `User-agent` header
- Requests that appear to include malicious SQL code

Managed rule group rules usually use the Block action, but not all do. For examples, some rules used for Bot Control use the CAPTCHA and Challenge action settings. For information about managed rule groups, see Using managed rule groups in AWS WAF.

Considerations for managing body inspection in AWS WAF

The body inspection size limit is the maximum request body size that AWS WAF can inspect. When a web request body is larger than the limit, the underlying host service only forwards the contents that are within the limit to AWS WAF for inspection.

- For Application Load Balancer and AWS AppSync, the limit is fixed at 8 KB (8,192 bytes).
- For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway, the default limit is 16 KB (16,384 bytes), and you can increase the limit for any of the resource types by increments of 16 KB, up to 64 KB. The setting options are 16 KB, 32 KB, 48 KB, and 64 KB.

Important

AWS WAF does not support request body inspection rules for gRPC traffic. If you enabled these rules on the protection pack (web ACL) for a CloudFront distribution or Application Load Balancer, any request that uses gRPC will ignore the request body inspection rules. All other AWS WAF rules will still apply. For more information, see Enable AWS WAF for distributions in the *Amazon CloudFront Developer Guide*.

Oversize body handling

If your web traffic includes bodies that are larger than the limit, your configured oversize handling will apply. For information about the options for oversize handling, see Oversize web request components in AWS WAF.

Pricing considerations for increasing the limit setting

AWS WAF charges a base rate for inspecting traffic that's within the default limit for the resource type.

For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway resources, if you increase the limit setting, the traffic that AWS WAF can inspect includes body sizes up to your new limit. You're charged extra only for the inspection of requests that have body sizes larger than the default 16 KB. For more information about pricing, see AWS WAF Pricing.

Options for modifying the body inspection size limit

You can configure the body inspection size limit for CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, or Amazon Bedrock AgentCore Gateway resources.

When you create or edit a protection pack (web ACL), you can modify the body inspection size limits in the resource association configuration. For the API, see the protection pack (web ACL)'s association configuration at AssociationConfig. For the console, see the configuration on the page where you specify the protection pack (web ACL)'s associated resources. For guidance on the console configuration, see Viewing web traffic metrics in AWS WAF.

AWS WAF inspection controls for HTTP/2 traffic on Application Load Balancer

You can configure how AWS WAF inspects HTTP/2 request bodies on your (Application Load Balancer) when routing to HTTP/2 targets. Adjusting the inspection timing allows you to balance security coverage with your application's communication patterns.

To modify this setting, edit your target group attributes for the HTTP/2 target. You can find this setting in the Application Load Balancer target group console under **WAF HTTP/2 traffic inspection behavior**. For more information, see ALB documentation.

The attributes for **WAF HTTP/2 traffic inspection behavior** are:

Inspect immediately (default)

AWS WAF inspects each HTTP/2 request immediately using the available request data.

Choose this option for bidirectional streaming applications where clients expect server responses before completing their request transmission. This prevents timeouts in applications that require real-time bidirectional communication.

Inspect after sufficient data

AWS WAF inspection begins only after receiving sufficient HTTP/2 data frames from the client, ensuring complete request inspection for enhanced security.

Choose this option for standard request-response applications where clients send all request data before expecting a server response. This is the recommended setting for most use cases.

Configuring CAPTCHA, challenge, and tokens in AWS WAF

You can configure options in your protection pack (web ACL) for the rules that use the CAPTCHA or Challenge rule actions and for the application integration SDKs that manage silent client challenges for AWS WAF managed protections.

These features mitigate bot activity by challenging end users with CAPTCHA puzzles and by presenting client sessions with silent challenges. When the client responds successfully, AWS WAF provides a token for them to use in their web request, timestamped with the last successful puzzle and challenge responses. For more information, see Intelligent threat mitigation in AWS WAF.

In your protection pack (web ACL) configuration, you can configure how AWS WAF manages these tokens:

- **CAPTCHA and challenge immunity times** – These specify how long a CAPTCHA or challenge timestamp remains valid. The protection pack (web ACL) settings are inherited by all rules that don't have their own immunity time settings configured and also by the application integration SDKs. For more information, see Setting timestamp expiration and token immunity times in AWS WAF.
- **Token domains** – By default, AWS WAF accepts tokens only for the domain of the resource that the protection pack (web ACL) is associated with. If you configure a token domain list, AWS WAF accepts tokens for all domains in the list and for the domain of the associated resource. For more information, see AWS WAF protection pack (web ACL) token domain list configuration.

Viewing web traffic metrics in AWS WAF

This section explains how to access summaries of web traffic metrics.

For any protection pack (web ACL) that you're using, you can access summaries of the web traffic metrics on the protection pack (web ACL)'s page in the AWS WAF console, under the **Traffic overview** tab. The console dashboards provide near real-time summaries of the Amazon CloudWatch metrics that AWS WAF collects when it evaluates your application web traffic. For more information about the dashboards, see Traffic overview dashboards for protection packs (web ACLs). For additional information about monitoring your protection pack (web ACL)'s traffic, see Monitoring and tuning your AWS WAF protections.

Deleting a protection pack (web ACL)

This section provides procedures for deleting protection packs (web ACLs) through the AWS console.

> **Important**
>
> Deleting a protection pack (web ACL) is permanent and can't be undone.

To delete a protection pack (web ACL), you first disassociate all AWS resources from the protection pack (web ACL). Perform the following procedure.

Using the new console

1. Sign in to the new AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2-pro.
2. In the navigation pane, choose **Resources & protection packs (web ACLs)**.
3. In the protection pack (web ACL) card, choose the **Edit** link next to **Resources** to open the **Manage resources** panel.
4. In the **Manage resources** section for the rule group, choose the resource you want to disassociate, and then choose **Disassociate**.

   > **Note**
   >
   > You must disassociate one resource at a time. Do not choose multiple resources.

5. In the confirmation page, type "disassociate", and then choose **Disassociate**. Repeat to disassociate each resource in the protection pack (web ACL).
6. Choose the protection pack (web ACL) that you want to delete. The console makes the main protection pack (web ACL) card editable, and also opens a side panel with details you can edit.
7. In the details panel, choose the trash can icon.
8. In the confirmation page, type "delete" and then choose **Delete**.

Using the standard console

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **web ACLs**.
3. Select the name of the web ACL that you want to delete. The console takes you to the web ACL's description, where you can edit it.

   > **Note**
   >
   > If you don't see the web ACL that you want to delete, make sure the Region selection inside the web ACLs section is correct. Any web ACLs that protect Amazon CloudFront distributions are in **Global (CloudFront)**.

4. On the **Associated AWS resources** tab, for each associated resource, select the radio button next to the resource name and then choose **Disassociate**. This disassociates the protection pack (web ACL) from your AWS resources.
5. In the navigation pane, choose **web ACLs**.
6. Select the radio button next to the web ACL that you are deleting, and then choose **Delete**.

AWS WAF rules

This section explains what an AWS WAF rule is and how it works.

An AWS WAF rule defines how to inspect HTTP(S) web requests and the action to take on a request when it matches the inspection criteria. You define rules only in the context of a rule group or protection pack (web ACL).

Rules don't exist in AWS WAF on their own. They aren't AWS resources, and they don't have Amazon Resource Names (ARNs). You can access a rule by name in the rule group or protection pack (web ACL) where it's defined. You can manage rules and copy them to other protection packs (web ACLs) by using the JSON view of the rule group or protection pack (web ACL) that contains the rule. You can also manage them through the AWS WAF console rule builder, which is available for protection packs (web ACLs) and rule groups.

Rule name

Each rule requires a name. Avoid names that start with `AWS` and names that are used for rule groups or rules that are managed for you by other services. See Recognizing rule groups provided by other services.

> **Note**
>
> If you change the name of a rule and you want the rule's metric name to reflect the change, you must update the metric name as well. AWS WAF doesn't automatically update the metric name for a rule when you change the rule name. You can change the metric name when you edit the rule in the console, by using the rule JSON editor. You can also change both names through the APIs and in any JSON listing that you use to define your protection pack (web ACL) or rule group.

Rule statement

Each rule also requires a rule statement that defines how the rule inspects web requests. The rule statement might contain other, nested statements at any depth, depending on the rule and statement type. Some rule statements take sets of criteria. For example, you can specify up to 10,000 IP addresses or IP address ranges for an IP set match rule.

You can define rules that inspect for criteria like the following:

- Scripts that are likely to be malicious. Attackers embed scripts that can exploit vulnerabilities in web applications. This is known as cross-site scripting (XSS).
- IP addresses or address ranges that requests originate from.
- Country or geographical location that requests originate from.
- Length of a specified part of the request, such as the query string.
- SQL code that is likely to be malicious. Attackers try to extract data from your database by embedding malicious SQL code in a web request. This is known as SQL injection.
- Strings that appear in the request, for example, values that appear in the `User-Agent` header or text strings that appear in the query string. You can also use regular expressions (regex) to specify these strings.
- Labels that prior rules in the protection pack (web ACL) have added to the request.

In addition to statements with web request inspection criteria, like the ones in the preceding list, AWS WAF supports logical statements for AND, OR, and NOT that you use to combine statements in a rule.

For example, based on recent requests that you've seen from an attacker, you might create a rule with a logical AND statement that combines the following nested statements:

- The requests come from 192.0.2.44.
- They contain the value BadBot in the User-Agent header.
- They appear to include SQL-like code in the query string.

In this case, the web request needs to match all of the statements to result in a match for the top-level AND.

Topics

- Using rule actions in AWS WAF
- Using rule statements in AWS WAF
- Using match rule statements in AWS WAF
- Using logical rule statements in AWS WAF
- Using rate-based rule statements in AWS WAF
- Using rule group rule statements in AWS WAF

Using rule actions in AWS WAF

This section explains how rule actions work.

The rule action tells AWS WAF what to do with a web request when it matches the criteria defined in the rule. You can optionally add custom behavior to each rule action.

> **Note**
>
> Rule actions can be terminating or non-terminating. A terminating action stops the protection pack (web ACL) evaluation of the request and either lets it continue to your protected application or blocks it.

Here are the rule action options:

- **Allow** – AWS WAF allows the request to be forwarded to the protected AWS resource for processing and response. This is a terminating action. In rules that you define, you can insert custom headers into the request before forwarding it to the protected resource.
- **Block** – AWS WAF blocks the request. This is a terminating action. By default, your protected AWS resource responds with an HTTP `403 (Forbidden)` status code. In rules that you define, you can customize the response. When AWS WAF blocks a request, the Block action settings determine the response that the protected resource sends back to the client.
- **Count** – AWS WAF counts the request but does not determine whether to allow it or block it. This is a non-terminating action. AWS WAF continues processing the remaining rules in the protection pack (web ACL). In rules that you define, you can insert custom headers into the request and you can add labels that other rules can match against.
- **CAPTCHA and Challenge** – AWS WAF uses CAPTCHA puzzles and silent challenges to verify that the request is not coming from a bot, and AWS WAF uses tokens to track recent successful client responses.

  CAPTCHA puzzles and silent challenges can only run when browsers are accessing HTTPS endpoints. Browser clients must be running in secure contexts in order to acquire tokens.

  > **Note**
  >
  > You are charged additional fees when you use the CAPTCHA or Challenge rule action in one of your rules or as a rule action override in a rule group. For more information, see AWS WAF Pricing.

  These rule actions can be terminating or non-terminating, depending on the state of the token in the request:

  - **Non-terminating for valid, unexpired token** – If the token is valid and unexpired according to the configured CAPTCHA or challenge immunity time, AWS WAF handles the request similar to the Count action. AWS WAF continues to inspect the web request based on the remaining rules in the protection pack (web ACL). Similar to the Count configuration, in rules that you define, you can optionally configure these actions with custom headers to insert into the request, and you can add labels that other rules can match against.

- **Terminating with blocked request for invalid or expired token** – If the token is invalid or the indicated timestamp is expired, AWS WAF terminates the inspection of the web request and blocks the request, similar to the Block action. AWS WAF then responds to the client with a custom response code. For CAPTCHA, if the request contents indicate that the client browser can handle it, AWS WAF sends a CAPTCHA puzzle in a JavaScript interstitial, which is designed to distinguish human clients from bots. For the Challenge action, AWS WAF sends a JavaScript interstitial with a silent challenge that is designed to distinguish normal browsers from sessions that are being run by bots.

For additional information, see CAPTCHA and Challenge in AWS WAF.

For information about customizing requests and responses, see Customized web requests and responses in AWS WAF.

For information about adding labels to matching requests, see Web request labeling in AWS WAF.

For information about how protection pack (web ACL) and rule settings interact, see Using protection packs (web ACLs) with rules and rule groups in AWS WAF.

Monetize action

The Monetize action returns an HTTP 402 Payment Required response to the requesting client. The response contains machine-readable x402 payment instructions that the client uses to complete payment and gain access to the resource. Use the Monetize action to charge AI bots and agents for content access. This action is available only for web ACLs associated with Amazon CloudFront distributions and requires a MonetizationConfig on the web ACL.

How the Monetize action works

When a rule with the Monetize action matches a request:

1. If the request does not include a valid `payment-signature` header, AWS WAF issues a Payment Required Challenge (HTTP 402 response with payment instructions).
2. The client signs a payment authorization and resubmits the request with the `payment-signature` header.
3. AWS WAF verifies the payment authorization.
4. On successful verification, the request proceeds to origin.

5. After origin returns a 2xx response, the payment is settled on-chain via third-party facilitation services.
6. Content is served to the client with settlement confirmation.

> **Important**
>
> To implement your AI traffic monetization policies, we use multiple detection techniques such as behavioral signals and risk-based systems to inspect and categorize inbound traffic. While these methods are designed to provide high-confidence classification, they are probabilistic and might not correctly identify or categorize all bot traffic in all cases. We continuously test and update our analysis methods to increase accuracy. We recommend using Test mode to validate that your policies produce the expected results before enabling live monetization.

Monetize action parameters

Parameter	Required	Description
`PriceMultiplier`	No	An integer multiplier (1–100) applied to the base price defined in the web ACL's MonetizationConfig. The effective price for the request is the base price multiplied by this value. Specify as a string. Default: `"1"`.

Monetize action behavior

- **Terminates evaluation** – Like Block, the Monetize action terminates rule evaluation for the matching request.
- **CloudFront only** – Supported only for web ACLs protecting Amazon CloudFront distributions.
- **Metrics** – Emits a `MonetizeRequests` metric in CloudWatch, separate from `BlockedRequests` and `AllowedRequests`.

- **Logging** – AWS WAF logs include monetization-specific fields when triggered.

Example: Monetize rule in JSON

```
{
  "Name": "MonetizeUnverifiedBots",
  "Priority": 5,
  "Statement": {
    "LabelMatchStatement": {
      "Scope": "LABEL",
      "Key": "awswaf:managed:aws:bot-control:bot:unverified"
    }
  },
  "Action": {
    "Monetize": {
      "PriceMultiplier": "2"
    }
  },
  "VisibilityConfig": {
    "CloudWatchMetricsEnabled": true,
    "MetricName": "MonetizeUnknownBotsMetric",
    "SampledRequestsEnabled": true
  }
}
```

Limitations

- Available only with Amazon CloudFront (global scope).
- Requires a MonetizationConfig on the web ACL.
- Rate-based rules cannot use Monetize as their action.

Using rule statements in AWS WAF

This section explains how rule statements work.

Rule statements are the part of a rule that tells AWS WAF how to inspect a web request. When AWS WAF finds the inspection criteria in a web request, we say that the web request matches the statement. Every rule statement specifies what to look for and how, according to the statement type.

Every rule in AWS WAF has a single top-level rule statement, which can contain other statements. Rule statements can be very simple. For example, you could have a statement that provides a set of originating countries to inspect your web requests for or you could have a rule statement in a protection pack (web ACL) that just references a rule group. Rule statements can also be very complex. For example, you could have a statement that combines many other statements with logical AND, OR, and NOT statements.

For most rules, you can add custom AWS WAF labeling to matching requests. The rules in the AWS Managed Rules rule groups add labels to matching requests. The labels that a rule adds provide information about the request to rules that are evaluated later in the protection pack (web ACL) and also in AWS WAF logs and metrics. For information about labeling, see Web request labeling in AWS WAF and Label match rule statement.

Nesting rule statements

AWS WAF supports nesting for many rule statements, but not for all. For example, you can't nest a rule group statement inside of another statement. You need to use nesting for some scenarios, such as scope-down statements and logical statements. The rule statement lists and rule details that follow describe the nesting capabilities and requirements for each category and rule.

The visual editor for rules in the console supports only one level of nesting for rule statements. For example, you can nest many types of statements inside a logical AND or OR rule, but you can't nest another AND or OR rule, because that requires a second level of nesting. To implement multiple levels of nesting, provide the rule definition in JSON, either through the JSON rule editor in the console or through the APIs.

Topics

- Adjusting rule statement settings in AWS WAF
- Using scope-down statements in AWS WAF
- Referencing reusable entities in AWS WAF

Adjusting rule statement settings in AWS WAF

This section describes the settings that you can specify in rule statements that inspect a component of the web request. For information on usage, see the individual rule statements at Using match rule statements in AWS WAF.

A subset of these web request components can also be used in rate-based rules, as custom request aggregation keys. For information, see Aggregating rate-based rules in AWS WAF.

For the request component settings, you specify the component type itself, and any additional options, depending on the component type. For example, when you inspect a component type that contains text, you can apply text transformations to it before inspecting it.

> **Note**
>
> Unless otherwise noted, if a web request doesn't have the request component that's specified in the rule statement, AWS WAF evaluates the request as not matching the rule criteria.

Contents

- Request components in AWS WAF
  - HTTP method
  - Single header
  - All headers
  - Header order
  - Cookies
  - URI fragment
  - URI path
  - JA3 fingerprint
  - JA4 fingerprint
  - Query string
  - Single query parameter
  - All query parameters
  - Body
  - JSON body
- Using forwarded IP addresses in AWS WAF
- Inspecting HTTP/2 pseudo headers in AWS WAF
- Using text transformations in AWS WAF
- Using pre-parse text transformations in AWS WAF

Request components in AWS WAF

This section describes the components of the web request that you can specify for inspection. You specify the request component for match rule statements that look for patterns inside the web request. These types of statements include string match, regex match, size constraint, and SQL injection attack statements. For information on how to use these request component settings, see the individual rule statements at Using match rule statements in AWS WAF

Unless otherwise noted, if a web request doesn't have the request component that's specified in the rule statement, AWS WAF evaluates the request as not matching the rule criteria.

> **Note**
>
> You specify a single request component for each rule statement that requires it. To inspect more than one component of a request, create a rule statement for each component.

The AWS WAF console and API documentation provide guidance for the request component settings in the following locations:

- **Rule builder** on the console – In the **Statement** settings for a regular rule type, choose the component that you want to inspect in the **Inspect** dialogue under **Request components**.
- **API statement contents** – `FieldToMatch`

The rest of this section describes the options for the part of the web request to inspect.

Topics

- HTTP method
- Single header
- All headers
- Header order
- Cookies
- URI fragment
- URI path
- JA3 fingerprint

- JA4 fingerprint
- Query string
- Single query parameter
- All query parameters
- Body
- JSON body

HTTP method

Inspects the HTTP method for the request. The HTTP method indicates the type of operation that the web request is asking your protected resource to perform, such as `POST` or `GET`.

Single header

Inspects a single named header in the request.

For this option, you specify the header name, for example, `User-Agent` or `Referer`. The string match for the name is not case sensitive and is performed after trimming leading and trailing spaces from both the request header and the rule.

All headers

Inspects all of the request headers, including cookies. You can apply a filter to inspect a subset of all headers.

For this option, you provide the following specifications:

- **Match patterns** – The filter to use to obtain a subset of headers for inspection. AWS WAF looks for these patterns in the headers keys.

  The match patterns setting can be one of the following:

  - **All** – Match all keys. Evaluate the rule inspection criteria for all headers.
  - **Excluded headers** – Inspect only the headers whose keys don't match any of the strings that you specify here. The string match for a key is not case sensitive. The matching is performed after trimming the leading and trailing spaces from request header and the match rule.
  - **Included headers** – Inspect only the headers that have a key that matches one of the strings that you specify here. The string match for a key is not case sensitive. The matching is

performed after trimming the leading and trailing spaces from request header and the match rule.

- **Match scope** – The parts of the headers that AWS WAF should inspect with the rule inspection criteria. You can specify **Keys**, **Values**, or **All** to inspect both keys and values for a match.

  All does not require a match to be found in the keys and a match to be found in the values. It requires a match to be found in the keys or the values or both. To require a match in the keys and in the values, use a logical AND statement to combine two match rules, one that inspects the keys and another that inspects the values.

- **Oversize handling** – How AWS WAF should handle requests that have header data that is larger than AWS WAF can inspect. AWS WAF can inspect at most the first 8 KB (8,192 bytes) of the request headers and at most the first 200 headers. The content is available for inspection by AWS WAF up to the first limit reached. You can choose to continue the inspection, or to skip inspection and mark the request as matching or not matching the rule. For more information about handling oversize content, see Oversize web request components in AWS WAF.

Header order

Inspect a string containing the list of the request's header names, ordered as they appear in the web request that AWS WAF receives for inspection. AWS WAF generates the string and then uses that as the field to match component in its inspection. AWS WAF separates the header names in the string with colons and with no added spaces, for example `host:user-agent:accept:authorization:referer`.

For this option, you provide the following specifications:

- **Oversize handling** – How AWS WAF should handle requests that have header data that is more numerous or larger than AWS WAF can inspect. AWS WAF can inspect at most the first 8 KB (8,192 bytes) of the request headers and at most the first 200 headers. The content is available for inspection by AWS WAF up to the first limit reached. You can choose to continue inspecting the headers that are available, or to skip inspection and mark the request as matching or not matching the rule. For more information about handling oversize content, see Oversize web request components in AWS WAF.

Cookies

Inspects all of the request cookies. You can apply a filter to inspect a subset of all cookies.

For this option, you provide the following specifications:

- **Match patterns** – The filter to use to obtain a subset of cookies for inspection. AWS WAF looks for these patterns in the cookie keys.

  The match patterns setting can be one of the following:

  - **All** – Match all keys. Evaluate the rule inspection criteria for all cookies.
  - **Excluded cookies** – Inspect only the cookies whose keys don't match any of the strings that you specify here. The string match for a key is case sensitive and must be exact.
  - **Included cookies** – Inspect only the cookies that have a key that matches one of the strings that you specify here. The string match for a key is case sensitive and must be exact.
- **Match scope** – The parts of the cookies that AWS WAF should inspect with the rule inspection criteria. You can specify **Keys**, **Values**, or **All** for both keys and values.

  All does not require a match to be found in the keys and a match to be found in the values. It requires a match to be found in the keys or the values or both. To require a match in the keys and in the values, use a logical AND statement to combine two match rules, one that inspects the keys and another that inspects the values.
- **Oversize handling** – How AWS WAF should handle requests that have cookie data that is larger than AWS WAF can inspect. AWS WAF can inspect at most the first 8 KB (8,192 bytes) of the request cookies and at most the first 200 cookies. The content is available for inspection by AWS WAF up to the first limit reached. You can choose to continue the inspection, or to skip inspection and mark the request as matching or not matching the rule. For more information about handling oversize content, see Oversize web request components in AWS WAF.

URI fragment

> **Note**
>
> Uri Fragment inspection is available only for CloudFront distributions and Application Load Balancers.

Inspects the part of a URL that follows the "#" symbol, providing additional information about the resource, for example, #section2. For information, see Uniform Resource Identifier (URI): Generic Syntax.

If you don't use a text transformation with this option, AWS WAF doesn't normalize the URI fragment and inspects it exactly as it receives it from the client in the request. For information about text transformations, see Using text transformations in AWS WAF.

Rule statement requirements

You must provide a fallback behavior for this rule statement. The fallback behavior is the match status that you want AWS WAF to assign to the web request if URI is missing the fragment or associated service is not Application Load Balancer or CloudFront. If you choose to match, AWS WAF treats the request as matching the rule statement and applies the rule action to the request. If you choose to not match, AWS WAF treats the request as not matching the rule statement.

URI path

Inspects the part of a URL that identifies a resource, for example, `/images/daily-ad.jpg`. For information, see Uniform Resource Identifier (URI): Generic Syntax.

If you don't use a text transformation with this option, AWS WAF doesn't normalize the URI and inspects it exactly as it receives it from the client in the request. For information about text transformations, see Using text transformations in AWS WAF.

JA3 fingerprint

Inspects the request's JA3 fingerprint.

> **Note**
>
> JA3 fingerprint inspection is available only for Amazon CloudFront distributions and Application Load Balancers.

The JA3 fingerprint is a 32-character hash derived from the TLS Client Hello of an incoming request. This fingerprint serves as a unique identifier for the client's TLS configuration. AWS WAF calculates and logs this fingerprint for each request that has enough TLS Client Hello information for the calculation. Almost all web requests include this information.

How to get the JA3 fingerprint for a client

You can obtain the JA3 fingerprint for a client's requests from the protection pack (web ACL) logs. If AWS WAF is able to calculate the fingerprint, it includes it in the logs. For information about the logging fields, see Log fields for protection pack (web ACL) traffic.

Rule statement requirements

You can inspect the JA3 fingerprint in a string match, regex match, regex pattern set match, or size constraint statement. You can't inspect it in an SQL injection or cross-site scripting match statement.

To match a single TLS configuration, provide the JA3 fingerprint string from the logs in a string match statement that's set to exactly match the string that you provide. For information about the string match statement, see String match rule statement.

To match a group of TLS configurations with a single pattern, use a regex match statement or a regex pattern set match statement. For information about these statements, see Regex match rule statement and Regex pattern set match rule statement.

You must provide a fallback behavior for this rule statement. The fallback behavior is the match status that you want AWS WAF to assign to the web request if AWS WAF is unable to calculate the JA3 fingerprint. If you choose to match, AWS WAF treats the request as matching the rule statement and applies the rule action to the request. If you choose to not match, AWS WAF treats the request as not matching the rule statement.

To use this match option, you must log your protection pack (web ACL) traffic. For information, see Logging AWS WAF protection pack (web ACL) traffic.

JA4 fingerprint

Inspects the request's JA4 fingerprint.

> **Note**
>
> JA4 fingerprint inspection is available only for Amazon CloudFront distributions and Application Load Balancers.

The JA4 fingerprint is a 36-character hash derived from the TLS Client Hello of an incoming request. This fingerprint serves as a unique identifier for the client's TLS configuration. JA4 fingerprinting is an extension of the JA3 fingerprinting that can result in fewer unique fingerprints for some browsers. AWS WAF calculates and logs this fingerprint for each request that has enough TLS Client Hello information for the calculation. Almost all web requests include this information.

How to get the JA4 fingerprint for a client

You can obtain the JA4 fingerprint for a client's requests from the protection pack (web ACL) logs. If AWS WAF is able to calculate the fingerprint, it includes it in the logs. For information about the logging fields, see Log fields for protection pack (web ACL) traffic.

Rule statement requirements

You can inspect the JA4 fingerprint in a string match, regex match, regex pattern set match, or size constraint statement. You can't inspect it in an SQL injection or cross-site scripting match statement.

To match a single TLS configuration, provide the JA4 fingerprint string from the logs in a string match statement that's set to exactly match the string that you provide. For information about the string match statement, see String match rule statement.

To match a group of TLS configurations with a single pattern, use a regex match statement or a regex pattern set match statement. For information about these statements, see Regex match rule statement and Regex pattern set match rule statement.

You must provide a fallback behavior for this rule statement. The fallback behavior is the match status that you want AWS WAF to assign to the web request if AWS WAF is unable to calculate the JA4 fingerprint. If you choose to match, AWS WAF treats the request as matching the rule statement and applies the rule action to the request. If you choose to not match, AWS WAF treats the request as not matching the rule statement.

To use this match option, you must log your protection pack (web ACL) traffic. For information, see Logging AWS WAF protection pack (web ACL) traffic.

Query string

Inspects the part of the URL that appears after a ? character, if any.

> **Note**
>
> For cross-site scripting match statements, we recommend that you choose **All query parameters** instead of **Query string**. Choosing **All query parameters** adds 10 WCUs to the base cost.

Single query parameter

Inspects a single query parameter that you have defined as part of the query string. AWS WAF inspects the value of the parameter that you specify.

For this option, you also specify a **Query argument**. For example, if the URL is `www.xyz.com?UserName=abc&SalesRegion=seattle`, you can specify `UserName` or `SalesRegion` for the query argument. The maximum length for the name of the argument is 30 characters. The name is not case sensitive, so if you specify `UserName`, AWS WAF matches all variations of `UserName`, including `username` and `UsERName`.

If the query string contains more than one instance of the query argument that you've specified, AWS WAF inspects all the values for a match, using OR logic. For example, in the URL `www.xyz.com?SalesRegion=boston&SalesRegion=seattle`, AWS WAF evaluates the name that you've specified against `boston` and `seattle`. If either is a match, the inspection is a match.

All query parameters

Inspects all query parameters in the request. This is similar to the single query parameter component choice, but AWS WAF inspects the values of all arguments within the query string. For example, if the URL is `www.xyz.com?UserName=abc&SalesRegion=seattle`, AWS WAF triggers a match if either the value of `UserName` or `SalesRegion` match the inspection criteria.

Choosing this option adds 10 WCUs to the base cost.

Body

Inspects the request body, evaluated as plain text. You can also evaluate the body as JSON using the JSON content type.

The request body is the part of the request that immediately follows the request headers. It contains any additional data that is needed for the web request, for example, data from a form.

- In the console, you select this under the **Request option** choice **Body**, by selecting the **Content type** choice **Plain text**.
- In the API, in the rule's `FieldToMatch` specification, you specify `Body` to inspect the request body as plain text.

For Application Load Balancer and AWS AppSync, AWS WAF can inspect the first 8 KB of the body of a request. For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and

Amazon Bedrock AgentCore Gateway, by default, AWS WAF can inspect the first 16 KB, and you can increase the limit up to 64 KB in your protection pack (web ACL) configuration. For more information, see Considerations for managing body inspection in AWS WAF.

You must specify oversize handling for this component type. Oversize handling defines how AWS WAF handles requests that have body data that is larger than AWS WAF can inspect. You can choose to continue the inspection, or to skip inspection and mark the request as matching or not matching the rule. For more information about handling oversize content, see Oversize web request components in AWS WAF.

You can also evaluate the body as parsed JSON. For information about this, see the section that follows.

JSON body

Inspects the request body, evaluated as JSON. You can also evaluate the body as plain text.

The request body is the part of the request that immediately follows the request headers. It contains any additional data that is needed for the web request, for example, data from a form.

- In the console, you select this under the **Request option** choice **Body**, by selecting the **Content type** choice **JSON**.
- In the API, in the rule's `FieldToMatch` specification, you specify `JsonBody`.

For Application Load Balancer and AWS AppSync, AWS WAF can inspect the first 8 KB of the body of a request. For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway, by default, AWS WAF can inspect the first 16 KB, and you can increase the limit up to 64 KB in your protection pack (web ACL) configuration. For more information, see Considerations for managing body inspection in AWS WAF.

You must specify oversize handling for this component type. Oversize handling defines how AWS WAF handles requests that have body data that is larger than AWS WAF can inspect. You can choose to continue the inspection, or to skip inspection and mark the request as matching or not matching the rule. For more information about handling oversize content, see Oversize web request components in AWS WAF.

Choosing this option doubles the match statement's base cost WCUs. For example, if the match statement base cost is 5 WCUs without JSON parsing, using JSON parsing doubles the cost to 10 WCUs.

For this option, you provide additional specifications, as described in the following section.

How AWS WAF handles JSON body inspection

When AWS WAF inspects the web request body as JSON, it performs steps to parse the body and extract the JSON elements for inspection. AWS WAF performs these steps in accordance with your configuration choices.

The following lists the steps that AWS WAF performs.

1. **Parse the body contents** – AWS WAF parses the contents of the web request body in order to extract the JSON elements for inspection. AWS WAF does its best to parse the entire contents of the body, but parsing can fail for a variety of error states in the contents. Examples include invalid characters, duplicate keys, truncation, and content whose root node isn't an object or an array.

   The option **Body parsing fallback behavior** determines what AWS WAF does if it fails to completely parse the JSON body:

   - **None (default behavior)** - AWS WAF evaluates the content only up to the point where it encountered a parsing error.
   - **Evaluate as string** - Inspect the body as plain text. AWS WAF applies the text transformations and inspection criteria that you defined for the JSON inspection to the body text string.
   - **Match** - Treat the web request as matching the rule statement. AWS WAF applies the rule action to the request.
   - **No match** - Treat the web request as not matching the rule statement.

   > **Note**
   >
   > This fallback behavior only triggers when AWS WAF encounters an error while parsing the JSON string.

   #### Parsing doesn't fully validate the JSON

   AWS WAF parsing doesn't fully validate the input JSON string, so parsing can succeed even for invalid JSON.

   For example, AWS WAF parses the following invalid JSON without error:

   - Missing comma: `{"key1":"value1""key2":"value2"}`

- Missing colon: `{"key1":"value1","key2""value2"}`
- Extra colons: `{"key1"::"value1","key2""value2"}`

For cases such as these where the parsing succeeds but the result isn't completely valid JSON, the outcome of the subsequent steps in the evaluation can vary. The extraction might miss some elements, or the rule evaluation might have unexpected results. We recommend that you validate the JSON that you receive in your application and handle invalid JSON as needed.

2. **Extract the JSON elements** – AWS WAF identifies the subset of JSON elements to inspect according to your settings:

- The option **JSON match scope** specifies the types of elements in the JSON that AWS WAF should inspect.

  You can specify **Keys**, **Values**, or **All** for both keys and values.

  All does not require a match to be found in the keys and a match to be found in the values. It requires a match to be found in the keys or the values or both. To require a match in the keys and in the values, use a logical `AND` statement to combine two match rules, one that inspects the keys and another that inspects the values.

- The option **Content to inspect** specifies how to filter the element set to the subset that you want AWS WAF to inspect.

  You must specify one of the following:

  - **Full JSON content** - Evaluate all elements.
  - **Only included elements** - Evaluate only elements whose paths match the JSON Pointer criteria that you provide. Don't use this option to indicate *all* paths in the JSON. Instead, use **Full JSON content**.

    For information about JSON Pointer syntax, see the Internet Engineering Task Force (IETF) documentation JavaScript Object Notation (JSON) Pointer.

    For example, in the console, you can provide the following:

    ```
    /dogs/0/name
    /dogs/1/name
    ```

    In the API or CLI, you can provide the following:

```
"IncludedPaths": ["/dogs/0/name", "/dogs/1/name"]
```

For example, say that the **Content to inspect** setting is **Only included elements**, and the included elements setting is `/a/b`.

For the following example JSON body:

```
{
  "a":{
    "c":"d",
    "b":{
      "e":{
        "f":"g"
      }
    }
  }
}
```

The element sets that AWS WAF would inspect for each **JSON match scope** setting are listed below. Note that the key `b`, which is part of the included elements path, isn't evaluated.

- **All**: `e`, `f`, and `g`.
- **Keys**: `e` and `f`.
- **Values**: `g`.

3. **Inspect the JSON element set** – AWS WAF applies any text transformations that you've specified to the extracted JSON elements and then matches the resulting element set against the rule statement's match criteria. This is the same transformation and evaluation behavior as for other web request components. If any of the extracted JSON elements match, the web request is a match for the rule.

Using forwarded IP addresses in AWS WAF

This section applies to rule statements that use the IP address of a web request. By default, AWS WAF uses the IP address from the web request origin. However, if a web request goes through one or more proxies or load balancers, the web request origin will contain the address of the last proxy, and not the originating address of the client. In this case, the originating client address is usually forwarded in another HTTP header. This header is typically `X-Forwarded-For` (XFF), but it can be a different one.

Rule statements that use IP addresses

The rule statements that use IP addresses are the following:

- IP set match - Inspects the IP address for a match with the addresses that are defined in an IP set.
- Geographic match - Uses the IP address to determine country and region of origin and matches the country of origin against a list of countries.
- ASN match - Uses the IP address to determine the Autonomous System Number (ASN) and matches the ASN against a list of ASNs.
- Using rate-based rule statements - Can aggregate requests by their IP addresses to ensure that no individual IP address sends requests at too high a rate. You can use IP address aggregation by itself or in combination with other aggregation keys.

You can instruct AWS WAF to use a forwarded IP address for any of these rule statements, either from the `X-Forwarded-For` header or from another HTTP header, instead of using the web request's origin. For details on how to provide the specifications, see the guidance for the individual rule statement types.

> **Note**
>
> If a header is missing, AWS WAF evaluates any statement that uses that header as "No match." If you use a NOT statement with a "No match" result, AWS WAF converts the evaluation to "Match." Missing headers don't trigger fallback behavior - only invalid header values do.

Fallback behavior

When you use the forwarded IP address, you indicate the match status for AWS WAF to assign to the web request if the request doesn't have a valid IP address in the specified position:

- **MATCH** - Treat the web request as matching the rule statement. AWS WAF applies the rule action to the request.
- **NO MATCH** - Treat the web request as not matching the rule statement.

IP addresses used in AWS WAF Bot Control

The Bot Control managed rule group verifies bots using the IP addresses from AWS WAF. If you use Bot Control and you have verified bots that route through a proxy or load balancer, you need to explicitly allow them using a custom rule. For example, you can configure a custom IP set match rule that uses forwarded IP addresses to detect and allow your verified bots. You can use the rule to customize your bot management in a number of ways. For information and examples, see AWS WAF Bot Control.

Detecting client IP behind supported CDNs

For requests that arrive at AWS WAF through a third-party CDN, the Bot Control managed rule group and the Anti-DDoS managed rule group automatically recognize traffic that originates from the IP ranges of Amazon CloudFront, Akamai, Cloudflare, and Fastly. When the rule groups detect a request from one of these CDNs, they transparently use the originating client IP from the upstream CDN's standard client IP header (for example, the `Fastly-Client-IP` header for Fastly). You don't need to configure forwarded IP addresses on these rule groups for these CDNs to work correctly. For other rule statements that use IP addresses, such as your custom IP set match, geo match, ASN match, and rate-based rules, you still need to specify a forwarded IP configuration if you want them to evaluate the upstream client IP instead of the CDN's IP.

General considerations for using forwarded IP addresses

Before you use a forwarded IP address, note the following general caveats:

- A header can be modified by proxies along the way, and the proxies might handle the header in different ways.
- Attackers might alter the contents of the header in an attempt to bypass AWS WAF inspections.
- The IP address inside the header can be malformed or invalid.
- The header that you specify might not be present at all in a request.

Considerations for using forwarded IP addresses with AWS WAF

The following list describes requirements and caveats for using forwarded IP addresses in AWS WAF:

- For any single rule, you can specify one header for the forwarded IP address. The header specification is case insensitive.
- For rate-based rule statements, any nested scoping statements do not inherit the forwarded IP configuration. Specify the configuration for each statement that uses a forwarded IP address.

- For geo match, ASN match, and rate-based rules, AWS WAF uses the first address in the header. For example, if a header contains `10.1.1.1, 127.0.0.0, 10.10.10.10` AWS WAF uses `10.1.1.1`
- For IP set match, you indicate whether to match against the first, last, or any address in the header. If you specify any, AWS WAF inspects all addresses in the header for a match, up to 10 addresses. If the header contains more than 10 addresses, AWS WAF inspects the last 10.
- Headers that contain multiple addresses must use a comma separator between the addresses. If a request uses a separator other than a comma, AWS WAF considers the IP addresses in the header malformed.
- If the IP addresses inside the header are malformed or invalid, AWS WAF designates the web request as matching the rule or not matching, according to the fallback behavior that you specify in the forwarded IP configuration.
- If the header that you specify isn't present in a request, AWS WAF doesn't apply the rule to the request at all. This means that AWS WAF doesn't apply the rule action and doesn't apply the fallback behavior.
- A rule statement that uses a forwarded IP header for the IP address won't use the IP address that's reported by the web request origin.

Best practices for using forwarded IP addresses with AWS WAF

When you use forwarded IP addresses, use the following best practices:

- Carefully consider all possible states of your request headers before enabling forwarded IP configuration. You might need to use more than one rule to get the behavior you want.
- To inspect multiple forwarded IP headers or to inspect the web request origin and a forwarded IP header, use one rule for each IP address source.
- To block web requests that have an invalid header, set the rule action to block and set the fallback behavior for the forwarded IP configuration to match.

Example JSON for forwarded IP addresses

The following geo match statement matches only if the `X-Forwarded-For` header contains an IP whose country of origin is US:

```
{
  "Name": "XFFTestGeo",
```

```
  "Priority": 0,
  "Action": {
    "Block": {}
  },
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "XFFTestGeo"
  },
  "Statement": {
    "GeoMatchStatement": {
      "CountryCodes": [
        "US"
      ],
      "ForwardedIPConfig": {
        "HeaderName": "x-forwarded-for",
        "FallbackBehavior": "MATCH"
      }
    }
  }
}
```

The following rate-based rule aggregates requests based on the first IP in the `X-Forwarded-For` header. The rule counts only requests that match the nested geo match statement, and it only blocks requests that match the geo match statement. The nested geo match statement also uses the `X-Forwarded-For` header to determine whether the IP address indicates a country of origin of US. If it does, or if the header is present but malformed, the geo match statement returns a match.

```
{
  "Name": "XFFTestRateGeo",
  "Priority": 0,
  "Action": {
    "Block": {}
  },
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "XFFTestRateGeo"
  },
  "Statement": {
    "RateBasedStatement": {
```

```
      "Limit": "100",
      "AggregateKeyType": "FORWARDED_IP",
      "ScopeDownStatement": {
        "GeoMatchStatement": {
          "CountryCodes": [
            "US"
          ],
          "ForwardedIPConfig": {
            "HeaderName": "x-forwarded-for",
            "FallbackBehavior": "MATCH"
          }
        }
      },
      "ForwardedIPConfig": {
        "HeaderName": "x-forwarded-for",
        "FallbackBehavior": "MATCH"
      }
    }
  }
}
```

Inspecting HTTP/2 pseudo headers in AWS WAF

This section explains how you can use AWS WAF to inspect HTTP/2 pseudo headers.

Protected AWS resources that support HTTP/2 traffic do not forward HTTP/2 pseudo headers to AWS WAF for inspection, but they provide contents of pseudo headers in web request components that AWS WAF inspects.

You can use AWS WAF to inspect only the pseudo headers that are listed in the following table.

HTTP/2 pseudo header contents mapped to web request components

HTTP/2 pseudo header	Web request component to inspect	Documentation
`:method`	HTTP method	HTTP method
`:authority`	`Host` header	Single header All headers

HTTP/2 pseudo header	Web request component to inspect	Documentation
`:path` URI path	URI path	URI path
`:path` query	Query string	Query string Single query parameter All query parameters

Using text transformations in AWS WAF

This section explains how to provide transformations for AWS WAF to apply before inspecting the request.

In statements that look for patterns or set constraints, you can provide transformations for AWS WAF to apply before inspecting the request. A transformation reformats a web request to eliminate some of the unusual formatting that attackers use in an effort to bypass AWS WAF.

When you use this with the JSON body request component selection, AWS WAF applies your transformations after parsing and extracting the elements to inspect from the JSON. For more information, see JSON body.

If you provide more than one transformation, you also set the order for AWS WAF to apply them.

WCUs – Each text transformation is 10 WCUs.

The AWS WAF console and API documentation also provide guidance for these settings in the following locations:

- **Rule builder** on the console – **Text transformation**. This option is available when you use request components.
- **API statement contents** – `TextTransformations`

Options for text transformations

Each transformation listing shows the console and API specifications followed by the description.

Base64 decode – `BASE64_DECODE`

AWS WAF decodes a Base64-encoded string.

Base64 decode extension – `BASE64_DECODE_EXT`

AWS WAF decodes a Base64-encoded string, but uses a forgiving implementation that ignores characters that aren't valid.

Command line – `CMD_LINE`

This option mitigates situations where attackers might be injecting an operating system command-line command and are using unusual formatting to disguise some or all of the command.

Use this option to perform the following transformations:

- Delete the following characters: `\ " ' ^`
- Delete spaces before the following characters: `/ (`
- Replace the following characters with a space: `, ;`
- Replace multiple spaces with one space
- Convert uppercase letters, `A-Z`, to lowercase, `a-z`

Command line Unix – `CMD_LINE_UNIX`

This option mitigates situations where attackers might be injecting a Unix or Linux operating system command-line command and are using unusual formatting to disguise some or all of the command.

Use this option to perform the following transformations:

- Delete the following characters: `\ " '`
- Replace the following characters with a space: tab `\t` (ASCII 9), newline `\n` (ASCII 10), carriage return `\r` (ASCII 13), vertical tab `\v` (ASCII 11), and formfeed `\f` (ASCII 12)
- Replace multiple spaces with one space
- Remove leading and trailing spaces
- Convert uppercase letters, `A-Z`, to lowercase, `a-z`

Command line Windows – `CMD_LINE_WIN`

This option mitigates situations where attackers might be injecting a Windows operating system command-line or PowerShell command and are using unusual formatting to disguise some or all of the command.

Use this option to perform the following transformations:

- Delete the following characters: `" ' ^ `
- Delete caret line-continuation sequences, a caret followed by a newline `\n` or by a carriage return and newline `\r\n`, as a single unit
- Replace the following characters with a space: tab `\t` (ASCII 9), newline `\n` (ASCII 10), carriage return `\r` (ASCII 13), vertical tab `\v` (ASCII 11), and formfeed `\f` (ASCII 12)
- Replace multiple spaces with one space
- Remove leading and trailing spaces
- Convert uppercase letters, `A-Z`, to lowercase, `a-z`
- Replace consecutive backslashes with a single backslash. Single backslashes are preserved, so Windows paths such as `c:\windows\system32\cmd.exe` and UNC paths remain intact.

Compress whitespace – `COMPRESS_WHITE_SPACE`

AWS WAF compresses white space by replacing multiple spaces with one space and replacing the following characters with a space character (ASCII 32):

- Formfeed (ASCII 12)
- Tab (ASCII 9)
- Newline (ASCII 10)
- Carriage return (ASCII 13)
- Vertical tab (ASCII 11)
- Non-breaking space (ASCII 160)

CSS decode – `CSS_DECODE`

AWS WAF decodes characters that were encoded using CSS 2.x escape rules `syndata.html#characters`. This function uses up to two bytes in the decoding process, so it can help to uncover ASCII characters that were encoded using CSS encoding that wouldn't typically be encoded. It's also useful in countering evasion, which is a combination of a backslash and non-hexadecimal characters. For example, `ja\vascript` for `javascript`.

Escape sequences decode – `ESCAPE_SEQ_DECODE`

AWS WAF decodes the following ANSI C escape sequences: `\a`, `\b`, `\f`, `\n`, `\r`, `\t`, `\v`, `\\`, `\?`, `\'`, `\"`, `\xHH` (hexadecimal), `\0000` (octal). Encodings that aren't valid remain in the output.

Hex decode – `HEX_DECODE`

AWS WAF decodes a string of hexadecimal characters into a binary.

HTML entity decode – HTML_ENTITY_DECODE

AWS WAF replaces characters that are represented in hexadecimal format &#xhhhh; or decimal format &#nnnn; with the corresponding characters.

AWS WAF replaces the following HTML-encoded characters with unencoded characters. This list uses lowercase HTML encoding, but the handling is case insensitive, for example &QuOt; and " are treated the same.

HTML-encoded character	replaced with...
"	"
&	&
<	<
>	>
or	non-breaking space, decimal 160
\n, decimal 10
	\t, decimal 9
{ or {	{
|, |, or		|
} or }	}
!	!
#	#
$	$
&percent; or %	%
'	\
(	(

HTML-encoded character	replaced with...
)	)
* or *	*
+	+
,	,
.	.
/	/
:	:
;	;
=	=
?	?
˜ or ˜	~
−	-
[or [	[
\	\\
] or]	]
&hat;	^
_ or &underbar;	_
` or `	`

JS decode – `JS_DECODE`

AWS WAF decodes JavaScript escape sequences. If a `\uHHHH` code is in the full-width ASCII code range of `FF01-FF5E`, then the higher byte is used to detect and adjust the lower byte. If not, only the lower byte is used and the higher byte is zeroed, causing a possible loss of information.

JS decode extended – `JS_DECODE_EXT`

AWS WAF decodes JavaScript escape sequences like `JS_DECODE`, but uses a forgiving implementation so that Microsoft Windows file paths aren't altered. `JS_DECODE_EXT` differs from `JS_DECODE` as follows:

- Decode the escapes `\\`, `\/`, `\'`, and `\"` by removing the backslash and keeping the subsequent character.
- Preserve the single-character escapes `\a`, `\b`, `\f`, `\n`, `\r`, `\t`, and `\v` by keeping both the backslash and the subsequent character.
- Preserve any other unrecognized `\C` escape sequence (for example `\d` or `\w`) by keeping both the backslash and the subsequent character, so that Windows file paths such as `\default` and `\windows` aren't altered.

Lowercase – `LOWERCASE`

AWS WAF converts uppercase letters (A-Z) to lowercase (a-z).

MD5 – `MD5`

AWS WAF calculates an MD5 hash from the data in the input. The computed hash is in a raw binary form.

None – `NONE`

AWS WAF inspects the web request as received, without any text transformations.

Normalize path – `NORMALIZE_PATH`

AWS WAF normalizes the input string by removing multiple slashes, directory self-references, and directory back-references that are not at the beginning of the input.

Normalize path Windows – `NORMALIZE_PATH_WIN`

AWS WAF converts backslash characters to forward slashes and then processes the resulting string using the `NORMALIZE_PATH` transformation.

Remove comments characters – `REMOVE_COMMENTS_CHAR`

AWS WAF removes common comment characters from the input: /*, */, --, and #.

Remove nulls – `REMOVE_NULLS`

AWS WAF removes all `NULL` bytes from the input.

Remove whitespace – `REMOVE_WHITESPACE`

AWS WAF removes all whitespace characters from the input.

Replace comments – `REPLACE_COMMENTS`

AWS WAF replaces each occurrence of a C-style comment (/* ... */) with a single space. It doesn't compress multiple consecutive occurrences. It replaces unterminated comments with a space (ASCII 0x20). It doesn't change a standalone termination of a comment (*/).

Replace nulls – `REPLACE_NULLS`

AWS WAF replaces each `NULL` byte in the input with the space character (ASCII 0x20).

SHA256 – `SHA256`

AWS WAF calculates a SHA-256 hash from the data in the input. The computed hash is in a raw binary form.

SQL hex decode – `SQL_HEX_DECODE`

AWS WAF decodes SQL hex data. For example, AWS WAF decodes (`0x414243`) to (`ABC`).

Trim – `TRIM`

AWS WAF removes whitespace from both the left and right sides of the input.

Trim left – `TRIM_LEFT`

AWS WAF removes whitespace from the left side of the input.

Trim right – `TRIM_RIGHT`

AWS WAF removes whitespace from the right side of the input.

Uppercase – `UPPERCASE`

AWS WAF converts all lowercase letters (a-z) to uppercase (A-Z).

URL decode – `URL_DECODE`

AWS WAF decodes a URL-encoded value.

URL decode Unicode – `URL_DECODE_UNI`

Like `URL_DECODE`, but with support for Microsoft-specific `%u` encoding. If the code is in the full-width ASCII code range of `FF01-FF5E`, the higher byte is used to detect and adjust the lower byte. Otherwise, only the lower byte is used and the higher byte is zeroed.

UTF8 to Unicode – `UTF8_TO_UNICODE`

AWS WAF converts all UTF-8 character sequences to Unicode. This helps normalize input and it minimizes false-positives and false-negatives for non-English languages.

Using pre-parse text transformations in AWS WAF

Pre-parse text transformations normalize the raw query string before AWS WAF parses it into individual query arguments. These transformations address parser differential attacks and HTTP Parameter Pollution (HPP), where attackers encode or manipulate query string delimiters to bypass inspection rules.

You can use pre-parse text transformations only when the request component is `SingleQueryArgument` or `AllQueryArguments`. AWS WAF applies pre-parse text transformations before standard text transformations. You can specify up to 10 pre-parse text transformations per rule statement. Each pre-parse text transformation adds 10 WCUs to the rule statement cost.

In the API, you configure pre-parse text transformations using the optional `PreParseTextTransformations` setting in your rule statement.

Note

If you don't need any pre-parse normalization, don't specify a pre-parse text transformation. The default behavior is `NONE`, which passes the raw query string to the parser without modification.

The following pre-parse text transformations are available.

None – `NONE`

AWS WAF inspects the web request as received, without any pre-parse text transformations.

URL decode – `URL_DECODE`

AWS WAF URL-decodes the raw query string before parsing. Decodes %HH sequences (for example, %26 becomes & and %3D becomes =). This might expose new query parameter boundaries that were previously encoded.

URL decode Unicode – `URL_DECODE_UNI`

Like `URL_DECODE`, but also decodes Unicode-encoded sequences (%uHHHH). If the code is in the full-width ASCII code range of `FF01-FF5E`, the higher byte is used to detect and adjust the lower byte. Otherwise, only the lower byte is used and the higher byte is zeroed.

Combine duplicate query arguments by comma – `COMBINE_DUPLICATE_QUERY_ARGS_BY_COMMA`

AWS WAF merges all values for duplicate query parameter keys into a single comma-separated (0x2C) string at the first occurrence position, then removes the later duplicates. For example, `a=1&b=2&a=3` becomes `a=1,3&b=2`. Use this transformation when your backend combines duplicate query parameters into arrays. This is common in .NET, API Gateway, and Express.js. AWS WAF inspects the combined value as a single unit.

Replace semicolons with ampersands – `REPLACE_SEMICOLONS_WITH_AMPERSANDS`

AWS WAF replaces all semicolons (0x3B) in the raw query string with ampersands (0x26) before parsing. Some web frameworks and services, including AWS API Gateway REST APIs, treat semicolons as query parameter delimiters equivalent to ampersands. This transformation normalizes the query string. AWS WAF then splits on semicolons the same way your backend does, ensuring consistent parameter parsing.

Using scope-down statements in AWS WAF

This section explains what a scope-down statement and how it works.

A scope-down statement is a nestable rule statement that you add inside a managed rule group statement or a rate-based statement to narrow the set of requests that the containing rule evaluates. The containing rule only evaluates the requests that first match the scope-down statement.

- **Managed rule group statement** – If you add a scope-down statement to a managed rule group statement, AWS WAF evaluates any request that doesn't match the scope-down statement as not matching the rule group. Only those requests that match the scope-down statement are

evaluated against the rule group. For managed rule groups with pricing that's based on the number of requests evaluated, scope-down statements can help contain costs.

For more information about managed rule group statements, see Using managed rule group statements in AWS WAF.

- **Rate-based rule statement** – A rate-based rule statement without a scope-down statement rate limits all requests that the rule evaluates. If you want to only control the rate for a specific category of requests, add a scope-down statement to the rate-based rule. For example, to only track and control the rate of requests from a specific geographical area, you can specify that geographical area in a geographic match statement and add it to your rate-based rule as the scope-down statement.

  For more information about rate-based rule statements, see Using rate-based rule statements in AWS WAF.

You can use any nestable rule in a scope-down statement. For the available statements, see Using match rule statements in AWS WAF and Using logical rule statements in AWS WAF. The WCUs for a scope-down statement are the WCUs required for the rule statement that you define in it. There's no additional cost for the use of a scope-down statement.

You can configure a scope-down statement in the same way as you do when you use the statement in a regular rule. For example, you can apply text transformations to a web request component that you're inspecting and you can specify a forwarded IP address to use as the IP address. These configurations apply only to the scope-down statement and are not inherited by the containing managed rule group or rate-based rule statement.

For example, if you apply text transformations to a query string in your scope-down statement, the scope-down statement inspects the query string after applying the transformations. If the request matches the scope-down statement criteria, AWS WAF then passes the web request to the containing rule in its original state, without the scope-down statement's transformations. The rule that contains the scope-down statement might apply text transformations of its own, but it doesn't inherit any from the scope-down statement.

You can't use a scope-down statement to specify any request inspection configuration for the containing rule statement. You can't use a scope-down statement as a web request preprocessor for the containing rule statement. The only role of a scope-down statement is to determine which requests are passed to the containing rule statement for inspection.

Referencing reusable entities in AWS WAF

This section explains how reusable entities work in AWS WAF.

Some rules use entities that are reusable and that are managed outside of your web ACLs, either by you, AWS, or an AWS Marketplace seller. When the reusable entity is updated, AWS WAF propagates the update to your rule. For example, if you use an AWS Managed Rules rule group in a protection pack (web ACL), when AWS updates the rule group, AWS propagates the change to your web ACL, to update its behavior. If you use an IP set statement in a rule, when you update the set, AWS WAF propagates the change to all rules that reference it, so any protection packs (web ACLs) that use those rules are kept up-to-date with your changes.

The following are the reusable entities that you can use in a rule statement.

- **IP sets** – You create and manage your own IP sets. On the console, you can access these from the navigation pane. For information about managing IP sets, see IP sets and regex pattern sets in AWS WAF.
- **Regex match sets** – You create and manage your own regex match sets. On the console, you can access these from the navigation pane. For information about managing regex pattern sets, see IP sets and regex pattern sets in AWS WAF.
- **AWS Managed Rules rule groups** – AWS manages these rule groups. On the console, these are available for your use when you add a managed rule group to your protection pack (web ACL). For more information about these, see AWS Managed Rules rule groups list.
- **AWS Marketplace managed rule groups** – AWS Marketplace sellers manage these rule groups and you can subscribe to them to use them. To manage your subscriptions, on the navigation pane of the console, choose **AWS Marketplace**. The AWS Marketplace managed rule groups are listed when you add a managed rule group to your protection pack (web ACL). For rule groups that you haven't yet subscribed to, you can find a link to AWS Marketplace on that page as well. For more information about AWS Marketplace seller managed rule groups, see AWS Marketplace rule groups.
- **Your own rule groups** – You manage your own rule groups, usually when you need some behavior that isn't available through the managed rule groups. On the console, you can access these from the navigation pane. For more information, see Managing your own rule groups.

Deleting a referenced set or rule group

When you delete a referenced entity, AWS WAF checks to see if it's currently being used in a protection pack (web ACL). If AWS WAF finds that it's in use, it warns you. AWS WAF is almost always able to determine if an entity is being referenced by a protection pack (web ACL). However, in rare cases, it might not be able to do so. If you need to be sure that the entity that you want to delete isn't in use, check for it in your web ACLs before deleting it.

Using match rule statements in AWS WAF

This section explains what a match statement is and how it works.

Match statements compare the web request or its origin against criteria that you provide. For many statements of this type, AWS WAF compares a specific component of the request for matching content.

Match statements are nestable. You can nest any of these statements inside logical rule statements and you can use them in scope-down statements. For information about logical rule statements, see Using logical rule statements in AWS WAF. For information about scope-down statements, see Using scope-down statements in AWS WAF.

This table describes the regular match statements that you can add to a rule and provides some guidelines for calculating protection pack (web ACL) capacity units (WCU) usage for each. For information about WCUs, see Web ACL capacity units (WCUs) in AWS WAF.

Match Statement	Description	WCUs
Geographic match	Inspects the request's country of origin and applies labels for the country and region of origin.	1
ASN match	Inspects the request against an ASN associated with IP addresses and address ranges.	1
IP set match	Inspects the request against a set of IP addresses and address ranges.	1 for most cases. If you configure the statement to use a header with forwarded

Match Statement	Description	WCUs
		IP addresses and specify a position in the header of Any, then increase the WCUs by 4.
Label match rule statement	Inspects the request for labels that have been added by other rules in the same protection pack (web ACL).	1
Regex match rule statement	Compares a regex pattern against a specified request component.	3, as a base cost. If you use the request component **All query parameters**, add 10 WCUs. If you use the request component **JSON body**, double the base cost WCUs. For each **Text transformation** that you apply, add 10 WCUs.
Regex pattern set	Compares regex patterns against a specified request component.	25 per pattern set, as a base cost. If you use the request component **All query parameters**, add 10 WCUs. If you use the request component **JSON body**, double the base cost WCUs. For each **Text transformation** that you apply, add 10 WCUs.

Match Statement	Description	WCUs
Size constraint	Checks size constraints against a specified request component.	1, as a base cost. If you use the request component **All query parameters**, add 10 WCUs. If you use the request component **JSON body**, double the base cost WCUs. For each **Text transformation** that you apply, add 10 WCUs.
SQLi attack	Inspects for malicious SQL code in a specified request component.	20, as a base cost. If you use the request component **All query parameters**, add 10 WCUs. If you use the request component **JSON body**, double the base cost WCUs. For each **Text transformation** that you apply, add 10 WCUs.
String match	Compares a string to a specified request component.	The base cost depends on the type of string match and is between 1 and 10. If you use the request component **All query parameters**, add 10 WCUs. If you use the request component **JSON body**, double the base cost WCUs. For each **Text transformation** that you apply, add 10 WCUs.

Match Statement	Description	WCUs
XSS scripting attack	Inspects for cross-site scripting attacks in a specified request component.	40, as a base cost. If you use the request component **All query parameters**, add 10 WCUs. If you use the request component **JSON body**, double the base cost WCUs. For each **Text transformation** that you apply, add 10 WCUs.

Geographic match rule statement

This section explains what a geographic match statement is and how it works.

Use geographic or geo match statements to manage web requests based on country and region of origin. A geo match statement add labels to web requests that indicate the country of origin and the region of origin. It adds these labels regardless of whether the statement criteria is a match for the request. A geo match statement also performs matching against the request's country of origin.

How to use the geo match statement

You can use the geo match statement for country or region matching, as follows:

- **Country** — You can use a geo match rule by itself to manage requests based solely on their country of origin. The rule statement matches against country codes. You can also follow a geo match rule with a label match rule that matches on the country of origin label.

  > **Note**
  >
  > To filter traffic from Hong Kong, use the ISO 3166-1 alpha-2 country code HK in your geo match statement.

- **Region** — Use a geo match rule followed by a label match rule to manage requests based on their region of origin. You can't use a geo match rule alone to match against region codes.

For information about using label match rules, see Label match rule statement and Web request labeling in AWS WAF.

How the geo match statement works

With the geo match statement, AWS WAF manages each web request as follows:

1. **Determines the request's country and region codes** — AWS WAF determines the country and region of a request based on its IP address. By default, AWS WAF uses the IP address of the web request's origin. You can instruct AWS WAF to use an IP address from an alternate request header, like `X-Forwarded-For`, by enabling forwarded IP configuration in the rule statement settings.

   AWS WAF determines the location of requests using MaxMind GeoIP databases. MaxMind reports very high accuracy of their data at the country level, although accuracy varies according to factors such as country and type of IP. For more information about MaxMind, see MaxMind IP Geolocation. If you think any of the GeoIP data is incorrect, you can submit a correction request to Maxmind at MaxMind Correct GeoIP2 Data.

   AWS WAF uses the alpha-2 country and region codes from the International Organization for Standardization (ISO) 3166 standard. You can find the codes at the following locations:

   - At the ISO website, you can search the country codes at ISO Online Browsing Platform (OBP).
   - On Wikipedia, the country codes are listed at ISO 3166-2.

     The region codes for a country are listed at the URL `https://en.wikipedia.org/wiki/ISO_3166-2:`*`<ISO country code>`*. For example, the regions for the United States are at ISO 3166-2:US and for Ukraine they're at ISO 3166-2:UA.
2. **Determines the country label and region label to add to the request** — The labels indicate whether the geo match statement uses the origin IP or a forwarded IP configuration.

   - **Origin IP**

     The country label is `awswaf:clientip:geo:country:`*`<ISO country code>`*. Example for the United States: `awswaf:clientip:geo:country:US`.

     The region label is `awswaf:clientip:geo:region:`*`<ISO country code>`*`-`*`<ISO region code>`*. Example for Oregon in the United States: `awswaf:clientip:geo:region:US-OR`.
   - **Forwarded IP**

The country label is `awswaf:forwardedip:geo:country:<ISO country code>`. Example for the United States: `awswaf:forwardedip:geo:country:US`.

The region label is `awswaf:forwardedip:geo:region:<ISO country code>-<ISO region code>`. Example for Oregon in the United States: `awswaf:forwardedip:geo:region:US-OR`.

If the country or region code isn't available for a request's specified IP address, AWS WAF uses XX in the labels, in the place of the value. For example, the following label is for a client IP whose country code isn't available: `awswaf:clientip:geo:country:XX` and the following is for a forwarded IP whose country is the United States, but whose region code isn't available: `awswaf:forwardedip:geo:region:US-XX`.

3. **Evaluates the request's country code against the rule criteria**

The geo match statement adds country and region labels to all requests that it inspects, regardless of whether it finds a match.

> **Note**
>
> AWS WAF adds any labels at the end of a rule's web request evaluation. Because of this, any label matching that you use against the labels from a geo match statement must be defined in a separate rule from the rule that contains the geo match statement.

If you want to inspect only region values, you can write a geo match rule with Count action and with a single country code match, followed by a label match rule for the region labels. You are required to supply a country code for the geo match rule to evaluate, even for this approach. You can reduce logging and count metrics by specifying a country that's very unlikely to be a source of traffic to your site.

CloudFront distributions and the CloudFront geo restriction feature

For CloudFront distributions, if you use the CloudFront geo restriction feature, be aware that the feature doesn't forward blocked requests to AWS WAF. It does forward allowed requests to AWS WAF. If you want to block requests based on the geography plus other criteria that you can specify in AWS WAF, use the AWS WAF geo match statement and don't use the CloudFront geo restriction feature.

Rule statement characteristics

Nestable – You can nest this statement type.

WCUs – 1 WCU.

Settings – This statement uses the following settings:

- **Country codes** – An array of country codes to compare for a geo match. These must be two-character country codes from the alpha-2 country ISO codes of the ISO 3166 international standard, for example, `["US","CN"]`.
- **(Optional) Forwarded IP configuration** – By default, AWS WAF uses the IP address in the web request origin to determine country of origin. Alternatively, you can configure the rule to use a forwarded IP in an HTTP header like `X-Forwarded-For` instead. AWS WAF uses the first IP address in the header. With this configuration, you also specify a fallback behavior to apply to a web request with a malformed IP address in the header. The fallback behavior sets the matching result for the request, to match or no match. For more information, see Using forwarded IP addresses.

Where to find this rule statement

- **Rule builder** on the console – For **Request option**, choose **Originates from a country in**.
- **API** – GeoMatchStatement

Examples

You can use the geo match statement to manage requests from specific countries or regions. For example, if you want to block requests from certain countries, but still allow requests from a specific set of IP addresses in those countries, you could create a rule with the action set to Block and the following nested statements, shown in pseudocode:

- AND statement
  - Geo match statement listing the countries that you want to block
  - NOT statement
    - IP set statement that specifies the IP addresses that you want to allow through

Or, if you want to block some regions in certain countries, but still allow requests from other regions in those countries, you could first define a geo match rule with the action set to Count. Then, define a label match rule that matches against the added geo match labels and handles the requests as you need.

The following pseudo code describes an example of this approach:

1. Geo match statement listing the countries with regions that you want to block, but with the action set to Count. This labels every web request regardless of match status, and it also gives you count metrics for the countries of interest.
2. AND statement with Block action
   - Label match statement that specifies the labels for the countries that you want to block
   - NOT statement
     - Label match statement that specifies the labels of the regions in those countries that you want to allow through

The following JSON listing shows an implementation of the two rules described in the prior pseudocode. These rules block all traffic from the United States except for traffic from Oregon and Washington. The geo match statement adds country and region labels to all requests that it inspects. The label match rule runs after the geo match rule, so it can match against the country and region labels that the geo match rule has just added. The geo match statement uses a forwarded IP address, so the label matching also specifies forwarded IP labels.

```
{
   "Name": "geoMatchForLabels",
   "Priority": 10,
   "Statement": {
     "GeoMatchStatement": {
       "CountryCodes": [
         "US"
       ],
       "ForwardedIPConfig": {
           "HeaderName": "X-Forwarded-For",
           "FallbackBehavior": "MATCH"
       }
     }
   },
   "Action": {
     "Count": {}
```

```
    },
    "VisibilityConfig": {
      "SampledRequestsEnabled": true,
      "CloudWatchMetricsEnabled": true,
      "MetricName": "geoMatchForLabels"
    }
},
{
    "Name": "blockUSButNotOROrWA",
    "Priority": 11,
    "Statement": {
      "AndStatement": {
        "Statements": [
          {
            "LabelMatchStatement": {
              "Scope": "LABEL",
              "Key": "awswaf:forwardedip:geo:country:US"
            }
          },
          {
            "NotStatement": {
              "Statement": {
                 "OrStatement": {
                   "Statements": [
                     {
                        "LabelMatchStatement": {
                          "Scope": "LABEL",
                          "Key": "awswaf:forwardedip:geo:region:US-OR"
                        }
                     },
                     {
                        "LabelMatchStatement": {
                          "Scope": "LABEL",
                          "Key": "awswaf:forwardedip:geo:region:US-WA"
                        }
                     }
                   ]
                 }
              }
            }
          }
        ]
      }
    },
```

```
    "Action": {
      "Block": {}
    },
    "VisibilityConfig": {
      "SampledRequestsEnabled": true,
      "CloudWatchMetricsEnabled": true,
      "MetricName": "blockUSButNotORORorWA"
    }
}
```

As another example, you can combine geo matching with rate-based rules to prioritize resources for users in a particular country or region. You create a different rate-based statement for each geo match or label match statement that you use to differentiate your users. Set a higher rate limit for users in the preferred country or region and set a lower rate limit for other users.

The following JSON listing shows a geo match rule followed by rate-based rules that limit the rate of traffic from the United States. The rules allow traffic from Oregon to come in at a higher rate than traffic from anywhere else in the country.

```
{
  "Name": "geoMatchForLabels",
  "Priority": 190,
  "Statement": {
    "GeoMatchStatement": {
      "CountryCodes": [
        "US"
      ]
    }
  },
  "Action": {
    "Count": {}
  },
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "geoMatchForLabels"
  }
},
{
  "Name": "rateLimitOregon",
  "Priority": 195,
  "Statement": {
    "RateBasedStatement": {
```

```
      "Limit": 3000,
      "AggregateKeyType": "IP",
      "ScopeDownStatement": {
        "LabelMatchStatement": {
          "Scope": "LABEL",
          "Key": "awswaf:clientip:geo:region:US-OR"
        }
      }
    }
  },
  "Action": {
    "Block": {}
  },
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "rateLimitOregon"
  }
},
{
  "Name": "rateLimitUSNotOR",
  "Priority": 200,
  "Statement": {
    "RateBasedStatement": {
      "Limit": 100,
      "AggregateKeyType": "IP",
      "ScopeDownStatement": {
        "AndStatement": {
          "Statements": [
            {
              "LabelMatchStatement": {
                "Scope": "LABEL",
                "Key": "awswaf:clientip:geo:country:US"
              }
            },
            {
              "NotStatement": {
                "Statement": {
                  "LabelMatchStatement": {
                    "Scope": "LABEL",
                    "Key": "awswaf:clientip:geo:region:US-OR"
                  }
                }
              }
```

```
            }
          ]
        }
      }
    }
  },
  "Action": {
    "Block": {}
  },
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "rateLimitUSNotOR"
  }
}
```

IP set match rule statement

This section explains what an IP set match statement is and how it works.

The IP set match statement inspects the IP address of a web request against a set of IP addresses and address ranges. Use this to allow or block web requests based on the IP addresses that the requests originate from. By default, AWS WAF uses the IP address from the web request origin, but you can configure the rule to use an HTTP header like `X-Forwarded-For` instead.

AWS WAF supports all IPv4 and IPv6 CIDR ranges except for `/0`. For more information about CIDR notation, see the Wikipedia entry Classless Inter-Domain Routing. An IP set can hold up to 10,000 IP addresses or IP address ranges to check.

> **Note**
>
> Each IP set match rule references an IP set, which you create and maintain independent of your rules. You can use a single IP set in multiple rules, and when you update the referenced set, AWS WAF automatically updates all rules that reference it.
> For information about creating and managing an IP set, see Creating and managing an IP set in AWS WAF.

When you add or update the rules in your rule group or protection pack (web ACL), choose the option **IP set** and select the name of the IP set that you want to use.

Rule statement characteristics

Nestable – You can nest this statement type.

WCUs – 1 WCU for most. If you configure the statement to use forwarded IP addresses and specify a position of ANY, increase the WCU usage by 4.

This statement uses the following settings:

- **IP set specification** – Choose the IP set that you want to use from the list or create a new one.
- **(Optional) Forwarded IP configuration** – An alternate forwarded IP header name to use in place of the request origin. You specify whether to match against the first, last, or any address in the header. You also specify a fallback behavior to apply to a web request with a malformed IP address in the specified header. The fallback behavior sets the matching result for the request, to match or no match. For more information, see Using forwarded IP addresses.

Where to find this rule statement

Where to find this rule statement

- **Rule builder** on the console – For **Request option**, choose **Originates from an IP address in**.
- **Add my own rules and rule groups** page on the console – Choose the **IP set** option.
- **API** – IPSetReferenceStatement

Autonomous System Number (ASN) match rule statement

An ASN match rule statement in AWS WAF allows you to inspect web traffic based on the Autonomous System Number (ASN) associated with the request's IP address. ASNs are unique identifiers assigned to large internet networks managed by organizations such as internet service providers, enterprises, universities, or government agencies. By using ASN match statements, you can allow or block traffic from specific network organizations without having to manage individual IP addresses. This approach offers a more stable and efficient way to control access compared to IP-based rules, as ASNs change less frequently than IP ranges.

ASN matching is particularly useful for scenarios such as blocking traffic from known problematic networks or allowing access only from trusted partner networks. The ASN match statement provides flexibility in determining the client IP address through optional forwarded IP

configuration, making it compatible with various network setups including those using content delivery networks (CDNs) or reverse proxies.

> **Note**
>
> ASN matching supplements, but doesn't replace, standard authentication and authorization controls. We recommend that you implement authentication and authorization mechanisms, such as IAM, to verify the identity of all requests in your applications.

How the ASN match statement works

AWS WAF determines the ASN of a request based on its IP address. By default, AWS WAF uses the IP address of the web request's origin. You can configure AWS WAF to use an IP address from an alternate request header, like `X-Forwarded-For`, by enabling forwarded IP configuration in the rule statement settings.

The ASN match statement compares the request's ASN against the list of ASNs specified in the rule. If the ASN matches one in the list, the statement evaluates to true, and the associated rule action is applied.

Handling unmapped ASNs

If AWS WAF cannot determine an ASN for a valid IP address, it assigns ASN 0. You can include ASN 0 in your rule to handle these cases explicitly.

Fallback Behavior for Invalid IP Addresses

When you configure the ASN match statement to use forwarded IP addresses, you can specify a fallback behavior of *Match* or *No match* for requests with invalid or missing IP addresses in the designated header.

Rule statement characteristics

Nestable – You can nest this statement type.

WCUs – 1 WCU

This statement uses the following settings:

- **ASN list** – An array of ASN numbers to compare for an ASN match. Valid values range from 0 to 4294967295. You can specify up to 100 ASNs for each rule.

- **(Optional) Forwarded IP configuration** – By default, AWS WAF uses the IP address in the web request origin to determine the ASN. Alternatively, you can configure the rule to use a forwarded IP in an HTTP header like `X-Forwarded-For` instead. You specify whether to use the first, last, or any address in the header. With this configuration, you also specify a fallback behavior to apply to a web request with a malformed IP address in the header. The fallback behavior sets the matching result for the request, to match or no match. For more information, see Using forwarded IP addresses.

Where to find this rule statement

- **Rule builder** on the console – For **Request option**, choose **Originates from ASN in**.
- **API** – AsnMatchStatement

Examples

This example blocks requests originating from two specific ASNs derived from an `X-Forwarded-For` header. If the IP address in the header is malformed, the configured fallback behavior is `NO_MATCH`.

```
{
  "Action": {
    "Block": {}
  },
  "Name": "AsnMatchStatementRule",
  "Priority": 1,
  "Statement": {
    "AsnMatchStatement": {
      "AsnList": [64496, 64500]
    },
    "ForwardedIPConfig": {
      "FallbackBehavior": "NO_MATCH",
      "HeaderName": "X-Forwarded-For"
    }
  },
  "VisibilityConfig": {
    "CloudWatchMetricsEnabled": true,
    "MetricName": "AsnMatchRuleMetrics",
    "SampledRequestsEnabled": true
  }
},
```

```
"VisibilityConfig": {
  "CloudWatchMetricsEnabled": true,
  "MetricName": "WebAclMetrics",
  "SampledRequestsEnabled": true
}
}
```

Label match rule statement

This section explains what a label match statement is and how it works.

The label match statement inspects the labels that are on the web request against a string specification. The labels that are available to a rule for inspection are those that have already been added to the web request by other rules in the same protection pack (web ACL) evaluation.

Labels don't persist outside of the protection pack (web ACL) evaluation, but you can access label metrics in CloudWatch and you can see summaries of label information for any protection pack (web ACL) in the AWS WAF console. For more information, see Label metrics and dimensions and Monitoring and tuning your AWS WAF protections. You can also see labels in the logs. For information, see Log fields for protection pack (web ACL) traffic.

> **Note**
>
> A label match statement can only see labels from rules that are evaluated earlier in the protection pack (web ACL). For information about how AWS WAF evaluates the rules and rule groups in a protection pack (web ACL), see Setting rule priority.

For more information about adding and matching labels, see Web request labeling in AWS WAF.

Rule statement characteristics

Nestable – You can nest this statement type.

WCUs – 1 WCU

This statement uses the following settings:

- **Match scope** – Set this to **Label** to match against the label name and, optionally, the preceding namespaces and prefix. Set this to **Namespace** to match against some or all of the namespace specifications and, optionally, the preceding prefix.

- **Key** – The string that you want to match against. If you specify a namespace match scope, this should only specify namespaces and optionally the prefix, with an ending colon. If you specify a label match scope, this must include the label name and can optionally include preceding namespaces and prefix.

For more information about these settings, see AWS WAF rules that match labels and AWS WAF label match examples.

Where to find this rule statement

- **Rule builder** on the console – For **Request option**, choose **Has label**.
- **API** – LabelMatchStatement

Regex match rule statement

This section explains what a regex match statement is and how it works.

A regex match statement instructs AWS WAF to match a request component against a single regular expression (regex). A web request matches the statement if the request component matches the regex that you specify.

This statement type is a good alternative to the Regex pattern set match rule statement for situations where you want to combine your matching criteria using mathematical logic. For example, if you want a request component to match against some regex patterns and to not match against others, you can combine the regex match statements using the AND rule statement and the NOT rule statement.

AWS WAF supports the pattern syntax used by the PCRE library `libpcre` with some exceptions. The library is documented at PCRE - Perl Compatible Regular Expressions. For information about AWS WAF support, see Supported regular expression syntax in AWS WAF.

Rule statement characteristics

Nestable – You can nest this statement type.

WCUs – 3 WCUs, as a base cost. If you use the request component **All query parameters**, add 10 WCUs. If you use the request component **JSON body**, double the base cost WCUs. For each **Text transformation** that you apply, add 10 WCUs.

This statement type operates on a web request component, and requires the following request component settings:

- **Request component** – The part of the web request to inspect, for example, a query string or the body.

  > **Warning**
  >
  > If you inspect the request components **Body**, **JSON body**, **Headers**, or **Cookies**, read about the limitations on how much content AWS WAF can inspect at Oversize web request components in AWS WAF.

  For information about web request components, see Adjusting rule statement settings in AWS WAF.
- **Optional text transformations** – Transformations that you want AWS WAF to perform on the request component before inspecting it. For example, you could transform to lowercase or normalize white space. If you specify more than one transformation, AWS WAF processes them in the order listed. For information, see Using text transformations in AWS WAF.

Where to find this rule statement

- **Rule builder** on the console – For **Match type**, choose **Matches regular expression**.
- **API** – RegexMatchStatement

Regex pattern set match rule statement

This section explains what a regex pattern set match statement is and how it works.

The regex pattern set match inspects the part of the web request that you specify for the regular expression patterns that you've specified inside a regex pattern set.

AWS WAF supports the pattern syntax used by the PCRE library `libpcre` with some exceptions. The library is documented at PCRE - Perl Compatible Regular Expressions. For information about AWS WAF support, see Supported regular expression syntax in AWS WAF.

Note

Each regex pattern set match rule references a regex pattern set, which you create and maintain independent of your rules. You can use a single regex pattern set in multiple rules, and when you update the referenced set, AWS WAF automatically updates all rules that reference it.
For information about creating and managing a regex pattern set, see Creating and managing a regex pattern set in AWS WAF.

A regex pattern set match statement instructs AWS WAF to search for any of the patterns in the set inside the request component that you choose. A web request will match the pattern set rule statement if the request component matches any of the patterns in the set.

If you want to combine your regex pattern matches using logic, for example to match against some regular expressions and not match against others, consider using Regex match rule statement.

Rule statement characteristics

Nestable – You can nest this statement type.

WCUs – 25 WCUs, as a base cost. If you use the request component **All query parameters**, add 10 WCUs. If you use the request component **JSON body**, double the base cost WCUs. For each **Text transformation** that you apply, add 10 WCUs.

This statement type operates on a web request component, and requires the following request component settings:

- **Request component** – The part of the web request to inspect, for example, a query string or the body.

  Warning

  If you inspect the request components **Body**, **JSON body**, **Headers**, or **Cookies**, read about the limitations on how much content AWS WAF can inspect at Oversize web request components in AWS WAF.

  For information about web request components, see Adjusting rule statement settings in AWS WAF.

- **Optional text transformations** – Transformations that you want AWS WAF to perform on the request component before inspecting it. For example, you could transform to lowercase or normalize white space. If you specify more than one transformation, AWS WAF processes them in the order listed. For information, see Using text transformations in AWS WAF.

This statement requires the following settings:

- Regex pattern set specification – Choose the regex pattern set that you want to use from the list or create a new one.

Where to find this rule statement

- **Rule builder** on the console – For **Match type**, choose **String match condition** > **Matches pattern from regular expression set**.
- **API** – RegexPatternSetReferenceStatement

Size constraint rule statement

This section explains what a size constraint statement is and how it works.

A size constraint statement compares the number of bytes that AWS WAF receives for a web request component to a number that you provide, and matches according to your comparison criteria.

The comparison criteria is an operator such as greater than (>) or less than (<). For example, you can match on requests that have a query string with a size that's greater than 100 bytes.

If you inspect the URI path, any `/` in the path counts as one character. For example, the URI path `/logo.jpg` is nine characters long.

> **Note**
>
> This statement only inspects the size of the web request component. It doesn't inspect the contents of the component.

Rule statement characteristics

Nestable – You can nest this statement type.

WCUs – 1 WCU, as a base cost. If you use the request component **All query parameters**, add 10 WCUs. If you use the request component **JSON body**, double the base cost WCUs. For each **Text transformation** that you apply, add 10 WCUs.

This statement type operates on a web request component, and requires the following request component settings:

- **Request component** – The part of the web request to inspect, for example, a query string or the body. For information about web request components, see Adjusting rule statement settings in AWS WAF.

  A size constraint statement inspects only the size of the component after any transformations have been applied. It does not inspect the contents of the component.
- **Optional text transformations** – Transformations that you want AWS WAF to perform on the request component before inspecting its size. For example, you could compress white space or decode HTML entities. If you specify more than one transformation, AWS WAF processes them in the order listed. For information, see Using text transformations in AWS WAF.

Additionally, this statement requires the following settings:

- **Size match condition** – This indicates the numerical comparison operator to use to compare the size that you provide with the request component that you've chosen. Choose the operator from the list.
- **Size** – The size setting, in bytes, to use in the comparison.

Where to find this rule statement

- **Rule builder** on the console – For **Match type**, under **Size match condition**, choose the condition that you want to use.
- **API** – SizeConstraintStatement

SQL injection attack rule statement

This section explains what a SQL injection rule statement is and how it works.

An SQL injection rule statement inspects for malicious SQL code. Attackers insert malicious SQL code into web requests in order to do things like modify your database or extract data from it.

Rule statement characteristics

Nestable – You can nest this statement type.

WCUs – The base cost depends on the sensitivity level setting for the rule statement: Low costs 20 and High costs 30.

If you use the request component **All query parameters**, add 10 WCUs. If you use the request component **JSON body**, double the base cost WCUs. For each **Text transformation** that you apply, add 10 WCUs.

This statement type operates on a web request component, and requires the following request component settings:

- **Request component** – The part of the web request to inspect, for example, a query string or the body.

  > **Warning**
  >
  > If you inspect the request components **Body**, **JSON body**, **Headers**, or **Cookies**, read about the limitations on how much content AWS WAF can inspect at Oversize web request components in AWS WAF.

  For information about web request components, see Adjusting rule statement settings in AWS WAF.
- **Optional text transformations** – Transformations that you want AWS WAF to perform on the request component before inspecting it. For example, you could transform to lowercase or normalize white space. If you specify more than one transformation, AWS WAF processes them in the order listed. For information, see Using text transformations in AWS WAF.

Additionally, this statement requires the following setting:

- **Sensitivity level** – This setting tunes the sensitivity of the SQL injection match criteria. The options are LOW and HIGH. The default setting is LOW.

  The HIGH setting detects more SQL injection attacks, and is the recommended setting. Due to the higher sensitivity, this setting generates more false positives, especially if your web requests typically contain unusual strings. During your protection pack (web ACL) testing and tuning, you

might need to do more work to mitigate false positives. For information, see Testing and tuning your AWS WAF protections.

The lower setting provides less stringent SQL injection detection, which also results in fewer false positives. LOW can be a better choice for resources that have other protections against SQL injection attacks or that have a low tolerance for false positives.

Where to find this rule statement

- **Rule builder** on the console – For **Match type**, choose **Attack match condition** > **Contains SQL injection attacks**.
- **API** – SqliMatchStatement

String match rule statement

This section explains what a string match statement is and how it works.

A string match statement indicates the string that you want AWS WAF to search for in a request, where in the request to search, and how. For example, you can look for a specific string at the start of any query string in the request or as an exact match for the request's `User-agent` header. Usually, the string consists of printable ASCII characters, but you can use any character from hexadecimal 0x00 to 0xFF (decimal 0 to 255).

Rule statement characteristics

Nestable – You can nest this statement type.

WCUs – The base cost depends on the type of match that you use.

- **Exactly matches string** – 2
- **Starts with string** – 2
- **Ends with string** – 2
- **Contains string** – 10
- **Contains word** – 10

If you use the request component **All query parameters**, add 10 WCUs. If you use the request component **JSON body**, double the base cost WCUs. For each **Text transformation** that you apply, add 10 WCUs.

This statement type operates on a web request component, and requires the following request component settings:

- **Request component** – The part of the web request to inspect, for example, a query string or the body.

  > **Warning**
  >
  > If you inspect the request components **Body**, **JSON body**, **Headers**, or **Cookies**, read about the limitations on how much content AWS WAF can inspect at Oversize web request components in AWS WAF.

  For information about web request components, see Adjusting rule statement settings in AWS WAF.
- **Optional text transformations** – Transformations that you want AWS WAF to perform on the request component before inspecting it. For example, you could transform to lowercase or normalize white space. If you specify more than one transformation, AWS WAF processes them in the order listed. For information, see Using text transformations in AWS WAF.

Additionally, this statement requires the following settings:

- **String to match** – This is the string that you want AWS WAF to compare to the specified request component. Usually, the string consists of printable ASCII characters, but you can use any character from hexadecimal 0x00 to 0xFF (decimal 0 to 255).
- **String match condition** – This indicates the search type that you want AWS WAF to perform.
  - **Exactly matches string** – The string and the value of the request component are identical.
  - **Starts with string** – The string appears at the beginning of the request component.
  - **Ends with string** – The string appears at the end of the request component.
  - **Contains string** – The string appears anywhere in the request component.
  - **Contains word** – The string that you specify must appear in the request component.

    For this option, the string that you specify must contain only alphanumeric characters or underscore (A-Z, a-z, 0-9, or _).

    One of the following must be true for the request to match:

- The string exactly matches the value of the request component, such as the value of a header.
- The string is at the beginning of the request component and is followed by a character other than an alphanumeric character or underscore (_), for example, `BadBot;`.
- The string is at the end of the request component and is preceded by a character other than an alphanumeric character or underscore (_), for example, `;BadBot`.
- The string is in the middle of the request component and is preceded and followed by characters other than alphanumeric characters or underscore (_), for example, `-BadBot;`.

Where to find this rule statement

- **Rule builder** on the console – For **Match type**, choose **String match condition**, and then fill in the strings that you want to match against.
- **API** – ByteMatchStatement

Cross-site scripting attack rule statement

This section explains what an XSS (cross-site scripting) attack statement is and how it works.

An XSS attack statement inspects for malicious scripts in a web request component. In an XSS attack, the attacker uses vulnerabilities in a benign website as a vehicle to inject malicious client-site scripts into other legitimate web browsers.

Rule statement characteristics

Nestable – You can nest this statement type.

WCUs – 40 WCUs, as a base cost. If you use the request component **All query parameters**, add 10 WCUs. If you use the request component **JSON body**, double the base cost WCUs. For each **Text transformation** that you apply, add 10 WCUs.

This statement type operates on a web request component, and requires the following request component settings:

- **Request component** – The part of the web request to inspect, for example, a query string or the body.

> **Warning**
>
> If you inspect the request components **Body**, **JSON body**, **Headers**, or **Cookies**, read about the limitations on how much content AWS WAF can inspect at Oversize web request components in AWS WAF.

For information about web request components, see Adjusting rule statement settings in AWS WAF.

- **Optional text transformations** – Transformations that you want AWS WAF to perform on the request component before inspecting it. For example, you could transform to lowercase or normalize white space. If you specify more than one transformation, AWS WAF processes them in the order listed. For information, see Using text transformations in AWS WAF.

Where to find this rule statement

- **Rule builder** on the console – For **Match type**, choose **Attack match condition** > **Contains XSS injection attacks**.
- **API** – XssMatchStatement

Using logical rule statements in AWS WAF

This section explains what a logical rule statement is and how it works.

Use logical rules statements to combine other statements or negate their results. Every logical rule statement takes at least one nested statement.

To logically combine or negate rule statement results, you nest the statements under logical rule statements.

Logical rules statements are nestable. You can nest them inside other logical rule statements and use them in scope-down statements. For information about scope-down statements, see Using scope-down statements in AWS WAF.

> **Note**
>
> The visual editor on the console supports one level of rule statement nesting, which works for many needs. To nest more levels, edit the JSON representation of the rule on the console or use the APIs.

This table describes the logical rule statements and provides guidelines for calculating protection pack (web ACL) capacity units (WCU) usage for each. For information about WCUs, see Web ACL capacity units (WCUs) in AWS WAF.

Logical Statement	Description	WCUs
AND logic	Combines nested statements with AND logic.	Based on nested statements
NOT logic	Negates the results of a nested statement.	Based on nested statement
OR logic	Combines nested statements with OR logic.	Based on nested statements

AND rule statement

The AND rule statement combines nested statements with a logical AND operation, so all nested statements must match for the AND statement to match. This requires at least two nested statements.

Rule statement characteristics

Nestable – You can nest this statement type.

WCUs – Depends on the nested statements.

Where to find this rule statement

- **Rule builder** on the console – For **If a request**, choose **matches all the statements (AND)**, and then fill in the nested statements.
- **API** – AndStatement

Examples

The following listing shows the use of AND and NOT logical rule statements to eliminate false positives from the matches for an SQL injection attack statement. For this example, suppose we can write a single byte match statement to match the requests that are resulting in false positives.

The AND statement matches for requests that do not match the byte match statement and that do match the SQL injection attack statement.

```
{
      "Name": "SQLiExcludeFalsePositives",
      "Priority": 0,
      "Statement": {
        "AndStatement": {
          "Statements": [
            {
              "NotStatement": {
                "Statement": {
                  "ByteMatchStatement": {
                    "SearchString": "string identifying a false positive",
                    "FieldToMatch": {
                      "Body": {
                        "OversizeHandling": "MATCH"
                      }
                    },
                    "TextTransformations": [
                      {
                        "Priority": 0,
                        "Type": "NONE"
                      }
                    ],
                    "PositionalConstraint": "CONTAINS"
                  }
                }
              }
            },
            {
              "SqliMatchStatement": {
                "FieldToMatch": {
                  "Body": {
                    "OversizeHandling": "MATCH"
                  }
                },
```

```
                "TextTransformations": [
                  {
                    "Priority": 0,
                    "Type": "NONE"
                  }
                ]
              }
            }
          ]
        }
      },
      "Action": {
        "Block": {}
      },
      "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "SQLiExcludeFalsePositives"
      }
    }
```

Using the console rule visual editor, you can nest a non-logical statement or a NOT statement under an OR or AND statement. The nesting of the NOT statement is shown in the prior example.

Using the console rule visual editor, you can nest most nestable statements under a logical rule statement, such as the one shown in the prior example. You can't use the visual editor to nest OR or AND statements. To configure this type of nesting, you need to provide your rule statement in JSON. For example, the following JSON rule listing includes an OR statement nested inside an AND statement.

```
{
  "Name": "match_rule",
  "Priority": 0,
  "Statement": {
    "AndStatement": {
      "Statements": [
        {
          "LabelMatchStatement": {
            "Scope": "LABEL",
            "Key": "awswaf:managed:aws:bot-control:bot:category:monitoring"
          }
        },
        {
```

```
          "NotStatement": {
            "Statement": {
              "LabelMatchStatement": {
                "Scope": "LABEL",
                "Key": "awswaf:managed:aws:bot-control:bot:name:pingdom"
              }
            }
          }
        },
        {
          "OrStatement": {
            "Statements": [
              {
                "GeoMatchStatement": {
                  "CountryCodes": [
                    "JM",
                    "JP"
                  ]
                }
              },
              {
                "ByteMatchStatement": {
                  "SearchString": "JCountryString",
                  "FieldToMatch": {
                    "Body": {}
                  },
                  "TextTransformations": [
                    {
                      "Priority": 0,
                      "Type": "NONE"
                    }
                  ],
                  "PositionalConstraint": "CONTAINS"
                }
              }
            ]
          }
        }
      ]
    }
  },
  "Action": {
    "Block": {}
  },
```

```
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "match_rule"
  }
}
```

NOT rule statement

The NOT rule statement logically negates the results of a single nested statement, so the nested statements must not match for the NOT statement to match, and vice versa. This requires one nested statement.

For example, if you want to block requests that don't originate in a specific country, create a NOT statement with action set to block, and nest a geographic match statement that specifies the country.

Rule statement characteristics

Nestable – You can nest this statement type.

WCUs – Depends on the nested statement.

Where to find this rule statement

- **Rule builder** on the console – For **If a request**, choose **doesn't match the statement (NOT)**, and then fill in the nested statement.
- **API** – NotStatement

OR rule statement

The OR rule statement combines nested statements with OR logic, so one of the nested statements must match for the OR statement to match. This requires at least two nested statements.

For example, if you want to block requests that come from a specific country or that contain a specific query string, you could create an OR statement and nest in it a geo match statement for the country and a string match statement for the query string.

If instead you want to block requests that *don't* come from a specific country or that contain a specific query string, you would modify the previous OR statement to nest the geo match

statement one level lower, inside a NOT statement. This level of nesting requires you to use the JSON formatting, because the console supports only one level of nesting.

Rule statement characteristics

Nestable – You can nest this statement type.

WCUs – Depends on the nested statements.

Where to find this rule statement

- **Rule builder** on the console – For **If a request**, choose **matches at least one of the statements (OR)**, and then fill in the nested statements.
- **API** – OrStatement

Examples

The following listing shows the use of OR to combine two other statements. The OR statement is a match if either of the nested statements match.

```
{
  "Name": "neitherOfTwo",
  "Priority": 1,
  "Action": {
    "Block": {}
  },
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "neitherOfTwo"
  },
  "Statement": {
    "OrStatement": {
      "Statements": [
        {
          "GeoMatchStatement": {
            "CountryCodes": [
              "CA"
            ]
          }
        },
        {
```

```
          "IPSetReferenceStatement": {
            "ARN": "arn:aws:wafv2:us-east-1:111111111111:regional/ipset/test-ip-
set-22222222/33333333-4444-5555-6666-777777777777"
          }
        }
      ]
    }
  }
}
```

Using the console rule visual editor, you can nest most nestable statements under a logical rule statement, but you can't use the visual editor to nest OR or AND statements. To configure this type of nesting, you need to provide your rule statement in JSON. For example, the following JSON rule listing includes an OR statement nested inside an AND statement.

```
{
  "Name": "match_rule",
  "Priority": 0,
  "Statement": {
    "AndStatement": {
      "Statements": [
        {
          "LabelMatchStatement": {
            "Scope": "LABEL",
            "Key": "awswaf:managed:aws:bot-control:bot:category:monitoring"
          }
        },
        {
          "NotStatement": {
            "Statement": {
              "LabelMatchStatement": {
                "Scope": "LABEL",
                "Key": "awswaf:managed:aws:bot-control:bot:name:pingdom"
              }
            }
          }
        },
        {
          "OrStatement": {
            "Statements": [
              {
                "GeoMatchStatement": {
                  "CountryCodes": [
```

```
                    "JM",
                    "JP"
                  ]
                }
              },
              {
                "ByteMatchStatement": {
                  "SearchString": "JCountryString",
                  "FieldToMatch": {
                    "Body": {}
                  },
                  "TextTransformations": [
                    {
                      "Priority": 0,
                      "Type": "NONE"
                    }
                  ],
                  "PositionalConstraint": "CONTAINS"
                }
              }
            ]
          }
        }
      ]
    }
  },
  "Action": {
    "Block": {}
  },
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "match_rule"
  }
}
```

Using rate-based rule statements in AWS WAF

This section explains what a rate-based rule statement is and how it works.

A rate-based rule counts incoming requests and rate limits requests when they are coming at too fast a rate. The rule aggregates requests according to your criteria, and counts and rate limits the aggregate groupings, based on the rule's evaluation window, request limit, and action settings.

> **Note**
>
> You can also rate limit web requests using the targeted protection level of the Bot Control AWS Managed Rules rule group. Using this managed rule group incurs additional fees. For more information, see Options for rate limiting in rate-based rules and targeted Bot Control rules.

AWS WAF tracks and manages web requests separately for each instance of a rate-based rule that you use. For example, if you provide the same rate-based rule settings in two web ACLs, each of the two rule statements represents a separate instance of the rate-based rule and each gets its own tracking and management by AWS WAF. If you define a rate-based rule inside a rule group, and then use that rule group in multiple places, each use creates a separate instance of the rate-based rule that gets its own tracking and management by AWS WAF.

Not nestable – You can't nest this statement type inside other statements. You can include it directly in a protection pack (web ACL) or rule group.

Scope-down statement – This rule type can take a scope-down statement, to narrow the scope of the requests that the rule tracks and rate limits. The scope-down statement can be optional or required, depending on your other rule configuration settings. The details are covered in this section. For general information about scope-down statements, see Using scope-down statements in AWS WAF.

WCUs – 2, as a base cost. For each custom aggregation key that you specify, add 30 WCUs. If you use a scope-down statement in the rule, calculate and add the WCUs for that.

Where to find this rule statement

- **Rule builder** in your protection pack (web ACL), on the console – Under **Rule**, for **Type**, choose **Rate-based rule**.
- **API** – RateBasedStatement

Topics

- Rate-based rule high-level settings in AWS WAF
- Rate-based rule caveats in AWS WAF
- Aggregating rate-based rules in AWS WAF

- Rate-based rule aggregation instances and counts
- Applying rate limiting to requests in AWS WAF
- Rate-based rule examples in AWS WAF
- Listing IP addresses that are being rate limited by rate-based rules

Rate-based rule high-level settings in AWS WAF

A rate-based rule statement uses the following high level settings:

- **Evaluation window** – The amount of time, in seconds, that AWS WAF should include in its request counts, looking back from the current time. For example, for a setting of 120, when AWS WAF checks the rate, it counts the requests for the 2 minutes immediately preceding the current time. Valid settings are 60 (1 minute), 120 (2 minutes), 300 (5 minutes), and 600 (10 minutes), and 300 (5 minutes) is the default.

  This setting doesn't determine how often AWS WAF checks the rate, but how far back it looks each time it checks. AWS WAF checks the rate frequently, with timing that's independent of the evaluation window setting.
- **Rate limit** – The maximum number of requests matching your criteria that AWS WAF should just track for the specified evaluation window. The lowest limit setting allowed is 10. When this limit is breached, AWS WAF applies the rule action setting to additional requests matching your criteria.

  AWS WAF applies rate limiting near the limit that you set, but does not guarantee an exact limit match. For more information, see Rate-based rule caveats.
- **Request aggregation** – The aggregation criteria to use on the web requests that the rate-based rule counts and rate limits. The rate limit that you set applies to each aggregation instance. For details, see Aggregating rate-based rules and Aggregation instances and counts.
- **Action** – The action to take on requests that the rule rate limits. You can use any rule action except Allow. This is set at the rule level as usual, but has some restrictions and behaviors that are specific to rate-based rules. For general information about rule actions, see Using rule actions in AWS WAF. For information specific to rate limiting, see Applying rate limiting to requests in AWS WAF in this section.
- **Scope of inspection and rate limiting** – You can narrow the scope of the requests that the rate-based statement tracks and rate limits by adding a scope-down statement. If you specify a scope-down statement, the rule only aggregates, counts, and rate limits requests that match the

scope-down statement. If you choose the request aggregation option **Count all**, then the scope-down statement is required. For more information about scope-down statements, see Using scope-down statements.

- **(Optional) Forwarded IP configuration** – This is only used if you specify **IP address in header** in your request aggregation, either alone or as part of the custom keys settings. AWS WAF retrieves the first IP address in the specified header and uses that as the aggregation value. A common header for this purpose is `X-Forwarded-For`, but you can specify any header. For more information, see Using forwarded IP addresses.

Rate-based rule caveats in AWS WAF

This section lists the caveats for using rate-based rules.

AWS WAF rate limiting is designed to control high request rates and protect your application's availability in the most efficient and effective way possible. It's not intended for precise request-rate limiting.

- AWS WAF estimates the current request rate using an algorithm that gives more importance to more recent requests. Because of this, AWS WAF will apply rate limiting near the limit that you set, but does not guarantee an exact limit match.
- Each time that AWS WAF estimates the rate of requests, AWS WAF looks back at the number of requests that came in during the configured evaluation window. Due to this and other factors such as propagation delays, it's possible for requests to be coming in at too high a rate for up to several minutes before AWS WAF detects and rate limits them. Similarly. the request rate can be below the limit for a period of time before AWS WAF detects the decrease and discontinues the rate limiting action. Usually, this delay is below 30 seconds.
- If you change any of the rate limit settings in a rule that's in use, the change resets the rule's rate limiting counts. This can pause the rule's rate limiting activities for up to a minute. The rate limit settings are the evaluation window, rate limit, request aggregation settings, forwarded IP configuration, and scope of inspection.

Aggregating rate-based rules in AWS WAF

This section explains your options for aggregating requests.

By default, a rate-based rule aggregates and rate limits requests based on the request IP address. You can configure the rule to use various other aggregation keys and key combinations. For

example, you can aggregate based on a forwarded IP address, on the HTTP method, or on a query argument. You can also specify aggregation key combinations, such as IP address and HTTP method, or the values of two different cookies.

> **Note**
>
> All of the request components that you specify in the aggregation key must be present in a web request for the request to be evaluated or rate limited by the rule.

You can configure your rate-based rule with the following aggregation options.

- **Source IP address** – Aggregate using only the IP address from the web request origin.

  The source IP address might not contain the address of the originating client. If a web request goes through one or more proxies or load balancers, this will contain the address of the last proxy.
- **IP address in header** – Aggregate using only a client address in an HTTP header. This is also referred to as a forwarded IP address.

  With this configuration, you also specify a fallback behavior to apply to a web request with a malformed IP address in the header. The fallback behavior sets the matching result for the request, to match or no match. For no match, the rate-based rule doesn't count or rate limit the request. For match, the rate-based rule groups the request together with other requests that have a malformed IP address in the specified header.

  Use caution with this option, because headers can be handled inconsistently by proxies and they can also be modified to bypass inspection. For additional information and best practices, see Using forwarded IP addresses in AWS WAF.
- **ASN** – Aggregate using an Autonomous System Number (ASN) associated with the source IP address as an aggregate key. This might not be the address of the originating client. If a web request goes through one or more proxies or load balancers, this contains the address of the last proxy.

  If AWS WAF can’t derive an ASN from the IP address, it counts the ASN as ASN 0. If you don't want to apply rate limiting to unmapped ASNs, you can create a scope-down rule that excludes requests with ASN 0.

- **ASN in header** – Aggregate using an ASN associated with a client IP address in an HTTP header. This is also referred to as a forwarded IP address. With this configuration, you also specify a fallback behavior to apply to a web request with an invalid or malformed IP address. The fallback behavior sets the matching result for the request, to match or no match. If you set the fallback behavior to match in the forwarded IP configuration, AWS WAF treats the invalid IP address as a matching value. This allows AWS WAF to continue evaluating any remaining parts of your rate-based rule's composite key. For no match, the rate-based rule doesn't count or rate limit the request.

  Use caution with this option, as headers can be handled inconsistently by proxies and they can be modified to bypass inspection. For additional information and best practices, see Using forwarded IP addresses in AWS WAF.
- **Count all** – Count and rate limit all requests that match the rule's scope-down statement. This option requires a scope-down statement. This is typically used to rate limit a specific set of requests, such as all requests with a specific label or all requests from a specific geographic area.
- **Custom keys** – Aggregate using one or more custom aggregation keys. To combine either of the IP address options with other aggregation keys, define them here under custom keys.

  Custom aggregation keys are a subset of the web request component options described at Request components in AWS WAF.

  The key options are the following. Except where noted, you can use an option multiple times, for example, two headers or three label namespaces.

  - **Label namespace** – Use a label namespace as an aggregation key. Each distinct fully qualified label name that has the specified label namespace contributes to the aggregation instance. If you use just one label namespace as your custom key, then each label name fully defines an aggregation instance.

    The rate-based rule uses only labels that have been added to the request by rules that are evaluated beforehand in the protection pack (web ACL).

    For information about label namespaces and names, see Label syntax and naming requirements in AWS WAF.
  - **Header** – Use a named header as an aggregation key. Each distinct value in the header contributes to the aggregation instance.

    Header takes an optional text transformation. See Using text transformations in AWS WAF.

- **Cookie** – Use a named cookie as an aggregation key. Each distinct value in the cookie contributes to the aggregation instance.

  Cookie takes an optional text transformation. See Using text transformations in AWS WAF.

- **Query argument** – Use a single query argument in the request as an aggregate key. Each distinct value for the named query argument contributes to the aggregation instance.

  Query argument takes an optional text transformation. See Using text transformations in AWS WAF.

- **Query string** – Use the entire query string in the request as an aggregate key. Each distinct query string contributes to the aggregation instance. You can use this key type once.

  Query string takes an optional text transformation. See Using text transformations in AWS WAF.

- **URI path** – Use the URI path in the request as an aggregate key. Each distinct URI path contributes to the aggregation instance. You can use this key type once.

  URI path takes an optional text transformation. See Using text transformations in AWS WAF.

- **JA3 fingerprint** – Use the JA3 fingerprint in the request as an aggregate key. Each distinct JA3 fingerprint contributes to the aggregation instance. You can use this key type once.

- **JA4 fingerprint** – Use the JA4 fingerprint in the request as an aggregate key. Each distinct JA4 fingerprint contributes to the aggregation instance. You can use this key type once.

- **HTTP method** – Use the request's HTTP method as an aggregate key. Each distinct HTTP method contributes to the aggregation instance. You can use this key type once.

- **IP address** – Aggregate using the IP address from the web request origin in combination with other keys.

  This might not contain the address of the originating client. If a web request goes through one or more proxies or load balancers, this will contain the address of the last proxy.

- **IP address in header** – Aggregate using the client address in an HTTP header in combination with other keys. This is also referred to as a forwarded IP address.

  Use caution with this option, as headers can be handled inconsistently by proxies and they can be modified to bypass inspection. For additional information and best practices, see Using forwarded IP addresses in AWS WAF.

Rate-based rule aggregation instances and counts

This section explains how a rate-based rule evaluates web requests.

When a rate-based rule evaluates web requests using your aggregation criteria, each unique set of values that the rule finds for the specified aggregation keys defines a unique *aggregation instance*.

- **Multiple keys** – If you've defined multiple custom keys, the value for each key contributes to the aggregation instance definition. Each unique combination of values defines an aggregation instance.
- **Single key** – If you've chosen a single key, either in the custom keys or by selecting one of the singleton IP address choices, then each unique value for the key defines an aggregation instance.
- **Count all - no keys** – If you've selected the aggregation option **Count all**, then all requests that the rule evaluates belong to a single aggregation instance for the rule. This choice requires a scope-down statement.

A rate-based rule counts web requests separately for each aggregation instance that it identifies.

For example, assume a rate-based rule evaluates web requests with the following IP address and HTTP method values:

- IP address 10.1.1.1, HTTP method POST
- IP address 10.1.1.1, HTTP method GET
- IP address 127.0.0.0, HTTP method POST
- IP address 10.1.1.1, HTTP method GET

The rule creates different aggregation instances according to your aggregation criteria.

- If the aggregation criteria is just the IP address, then each individual IP address is an aggregation instance, and AWS WAF counts requests separately for each. The aggregation instances and request counts for our example would be the following:
  - IP address 10.1.1.1: count 3
  - IP address 127.0.0.0: count 1
- If the aggregation criteria is HTTP method, then each individual HTTP method is an aggregation instance. The aggregation instances and request counts for our example would be the following:

  - HTTP method POST: count 2
  - HTTP method GET: count 2
- If the aggregation criteria is IP address and HTTP method, then each IP address and each HTTP method would contribute to the combined aggregation instance. The aggregation instances and request counts for our example would be the following:
  - IP address 10.1.1.1, HTTP method POST: count 1
  - IP address 10.1.1.1, HTTP method GET: count 2
  - IP address 127.0.0.0, HTTP method POST: count 1

Applying rate limiting to requests in AWS WAF

This section explains how rate limiting behavior works for rate-based rules.

The criteria that AWS WAF uses to rate limit requests for a rate-based rule is the same criteria that AWS WAF uses to aggregate requests for the rule. If you define a scope-down statement for the rule, AWS WAF only aggregates, counts, and rate limits requests that match the scope-down statement.

The match criteria that causes a rate-based rule to apply its rule action settings to a specific web request are as follows:

- The web request matches the rule's scope-down statement, if one is defined.
- The web request belongs to an aggregation instance whose request count is currently over the rule's limit.

How AWS WAF applies the rule action

When a rate-based rule applies rate limiting to a request, it applies the rule action and, if you've defined any custom handling or labeling in your action specification, the rule applies those. This request handling is the same as the way a match rule applies its action settings to matching web requests. A rate-based rule only applies labels or performs other actions on requests that it is actively rate limiting.

You can use any rule action except Allow. For general information about rule actions, see Using rule actions in AWS WAF.

The following list describes how rate limiting works for each of the actions.

- **Block** – AWS WAF blocks the request and applies any custom blocking behavior that you've defined.
- **Count** – AWS WAF counts the request, applies any custom headers or labels that you've defined, and continues the protection pack (web ACL) evaluation of the request.

  This action doesn't limit the rate of requests. It just counts the requests that are over the limit.
- **CAPTCHA or Challenge** – AWS WAF handles the request either like a Block or like a Count, depending on the state of the request's token.

  This action doesn't limit the rate of requests that have valid tokens. It limits the rate of requests that are over the limit and are also missing valid tokens.

  - If the request doesn't have a valid, unexpired token, the action blocks the request and sends the CAPTCHA puzzle or the browser challenge back to the client.

    If the end user or client browser responds successfully, the client receives a valid token and it automatically resends the original request. If rate limiting for the aggregation instance is still in effect, this new request with the valid, unexpired token will have the action applied to it as described in the next bullet point.
  - If the request has a valid, unexpired token, the CAPTCHA or Challenge action verifies the token and takes no action on the request, similar to the Count action. The rate-based rule returns the request evaluation back to the protection pack (web ACL) without taking any terminating action, and the protection pack (web ACL) continues its evaluation of the request.

  For additional information, see CAPTCHA and Challenge in AWS WAF.

If you rate limit only the IP address or forwarded IP address

When you configure the rule to rate limit only IP address for forwarded IP address, you can retrieve the list of IP addresses that the rule is currently rate limiting. If you're using a scope-down statement, the requests that are rate limited are only those in the IP list that match the scope-down statement. For information about retrieving the IP address list, see Listing IP addresses that are being rate limited by rate-based rules.

Rate-based rule examples in AWS WAF

This section describes example configurations for a variety of common rate-based rules use cases.

Each example provides a description of the use case and then shows the solution in JSON listings for the custom configured rules.

> **Note**
>
> The JSON listings shown in these examples were created in the console by configuring the rule and then editing it using the **Rule JSON editor**.

Topics

- Rate limit the requests to a login page
- Rate limit the requests to a login page from any IP address, user agent pair
- Rate limit the requests that are missing a specific header
- Rate limit the requests with specific labels
- Rate limit the requests for labels that have a specified label namespace
- Rate limit the requests with specific ASNs

Rate limit the requests to a login page

To limit the number of requests to the login page on your website without affecting traffic to the rest of your site, you could create a rate-based rule with a scope-down statement that matches requests to your login page and with the request aggregation set to **Count all**.

The rate-based rule will count all requests for the login page in a single aggregation instance and apply the rule action to all requests matching the scope-down statement when the requests exceed the limit.

The following JSON listing shows an example of this rule configuration. The count all aggregation option is listed in the JSON as the setting `CONSTANT`. This example matches login pages that start with `/login`.

```
{
  "Name": "test-rbr",
  "Priority": 0,
  "Action": {
    "Block": {}
  },
  "VisibilityConfig": {
```

```
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "test-rbr"
  },
  "Statement": {
    "RateBasedStatement": {
      "Limit": 1000,
      "EvaluationWindowSec": 300,
      "AggregateKeyType": "CONSTANT",
      "ScopeDownStatement": {
        "ByteMatchStatement": {
          "FieldToMatch": {
            "UriPath": {}
          },
          "PositionalConstraint": "STARTS_WITH",
          "SearchString": "/login",
          "TextTransformations": [
            {
              "Type": "NONE",
              "Priority": 0
            }
          ]
        }
      }
    }
  }
}
```

Rate limit the requests to a login page from any IP address, user agent pair

To limit the number of requests to the login page on your website for IP address, user agent pairs that exceed your limit, set the request aggregation to **Custom keys** and provide the aggregation criteria.

The following JSON listing shows an example of this rule configuration. In this example, we've set the limit to 100 requests in any five minute period per IP address, user agent pair.

```
{
  "Name": "test-rbr",
  "Priority": 0,
  "Action": {
    "Block": {}
  },
```

```
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "test-rbr"
  },
  "Statement": {
    "RateBasedStatement": {
      "Limit": 100,
      "EvaluationWindowSec": 300,
      "AggregateKeyType": "CUSTOM_KEYS",
      "CustomKeys": [
        {
          "Header": {
            "Name": "User-Agent",
            "TextTransformations": [
              {
                "Priority": 0,
                "Type": "NONE"
              }
            ]
          }
        },
        {
          "IP": {}
        }
      ],
      "ScopeDownStatement": {
        "ByteMatchStatement": {
          "FieldToMatch": {
            "UriPath": {}
          },
          "PositionalConstraint": "STARTS_WITH",
          "SearchString": "/login",
          "TextTransformations": [
            {
              "Type": "NONE",
              "Priority": 0
            }
          ]
        }
      }
    }
  }
```

```
}
```

Rate limit the requests that are missing a specific header

To limit the number of requests that are missing a specific header, you can use the **Count all** aggregation option with a scope-down statement. Configure the scope-down statement with a logical `NOT` statement containing a statement that returns true only if the header exists and has a value.

The following JSON listing shows an example of this rule configuration.

```
{
  "Name": "test-rbr",
  "Priority": 0,
  "Action": {
    "Block": {}
  },
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "test-rbr"
  },
  "Statement": {
    "RateBasedStatement": {
      "Limit": 1000,
      "AggregateKeyType": "CONSTANT",
      "EvaluationWindowSec": 300,
      "ScopeDownStatement": {
        "NotStatement": {
          "Statement": {
            "SizeConstraintStatement": {
              "FieldToMatch": {
                "SingleHeader": {
                  "Name": "user-agent"
                }
              },
              "ComparisonOperator": "GT",
              "Size": 0,
              "TextTransformations": [
                {
                  "Type": "NONE",
                  "Priority": 0
                }
```

```
                ]
              }
            }
          }
        }
      }
    }
  }
}
```

Rate limit the requests with specific labels

To limit the number of requests of various categories, you can combine rate limiting with any rule or rule group that add labels to requests. To do this, you configure your protection pack (web ACL) as follows:

- Add the rules or rule groups that add labels, and configure them so that they don't block or allow the requests that you want to rate limit. If you use managed rule groups, you might need to override some rule group rule actions to Count to achieve this behavior.
- Add a rate-based rule to your protection pack (web ACL) with a priority number setting that is higher than the labeling rules and rule groups. AWS WAF evaluates rules in numeric order, starting from the lowest, so your rate-based rule will run after the labeling rules. Configure your rate limiting on the labels using a combination of label matching in the rule's scope-down statement and label aggregation.

The following example uses the Amazon IP reputation list AWS Managed Rules rule group. The rule group rule `AWSManagedIPDDoSList` detects and labels requests whose IPs are known to be actively engaging in DDoS activities. The rule's action is configured to Count in the rule group definition. For more information about the rule group, see the section called “Amazon IP reputation list”.

The following protection pack (web ACL) JSON listing uses the IP reputation rule group followed by a label-matching rate-based rule. The rate-based rule uses a scope-down statement to filter for requests that have been marked by the rule group rule. The rate-based rule statement aggregates and rate limits the filtered requests by their IP addresses.

```
{
  "Name": "test-web-acl",
  "Id": ...
  "ARN": ...
```

```
  "DefaultAction": {
    "Allow": {}
  },
  "Description": "",
  "Rules": [
    {
      "Name": "AWS-AWSManagedRulesAmazonIpReputationList",
      "Priority": 0,
      "Statement": {
        "ManagedRuleGroupStatement": {
          "VendorName": "AWS",
          "Name": "AWSManagedRulesAmazonIpReputationList"
        }
      },
      "OverrideAction": {
        "None": {}
      },
      "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "AWS-AWSManagedRulesAmazonIpReputationList"
      }
    },
    {
      "Name": "test-rbr",
      "Priority": 1,
      "Statement": {
        "RateBasedStatement": {
          "Limit": 100,
          "EvaluationWindowSec": 300,
          "AggregateKeyType": "IP",
          "ScopeDownStatement": {
            "LabelMatchStatement": {
              "Scope": "LABEL",
              "Key": "awswaf:managed:aws:amazon-ip-list:AWSManagedIPDDoSList"
            }
          }
        }
      },
      "Action": {
        "Block": {}
      },
      "VisibilityConfig": {
        "SampledRequestsEnabled": true,
```

```
        "CloudWatchMetricsEnabled": true,
        "MetricName": "test-rbr"
      }
    }
  ],
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "test-web-acl"
  },
  "Capacity": 28,
  "ManagedByFirewallManager": false,
  "RetrofittedByFirewallManager": false,
  "LabelNamespace": "awswaf:0000000000:webacl:test-web-acl:"
}
```

Rate limit the requests for labels that have a specified label namespace

> **Note**
>
> The common level rules in the Bot Control managed rule group add labels for bots of various categories, but they only block requests from unverified bots. For information about these rules, see Bot Control rules listing.

If you use the Bot Control managed rule group, you can add rate limiting for requests from individual verified bots. To do this, you add a rate-based rule that runs after the Bot Control rule group and aggregates requests by their bot name labels. You specify the **Label namespace** aggregation key and set the namespace key to `awswaf:managed:aws:bot-control:bot:name:`. Each unique label with the specified namespace will define an aggregation instance. For example, the labels `awswaf:managed:aws:bot-control:bot:name:axios` and `awswaf:managed:aws:bot-control:bot:name:curl` each define an aggregation instance.

The following protection pack (web ACL) JSON listing shows this configuration. The rule in this example limits requests for any single bot aggregation instance to 1,000 in a two minute period.

```
{
  "Name": "test-web-acl",
  "Id": ...
  "ARN": ...
  "DefaultAction": {
```

```
    "Allow": {}
  },
  "Description": "",
  "Rules": [
    {
      "Name": "AWS-AWSManagedRulesBotControlRuleSet",
      "Priority": 0,
      "Statement": {
        "ManagedRuleGroupStatement": {
          "VendorName": "AWS",
          "Name": "AWSManagedRulesBotControlRuleSet",
          "ManagedRuleGroupConfigs": [
            {
              "AWSManagedRulesBotControlRuleSet": {
                "InspectionLevel": "COMMON"
              }
            }
          ]
        }
      },
      "OverrideAction": {
        "None": {}
      },
      "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "AWS-AWSManagedRulesBotControlRuleSet"
      }
    },
    {
      "Name": "test-rbr",
      "Priority": 1,
      "Statement": {
        "RateBasedStatement": {
          "Limit": 1000,
          "EvaluationWindowSec": 120,
          "AggregateKeyType": "CUSTOM_KEYS",
          "CustomKeys": [
            {
              "LabelNamespace": {
                "Namespace": "awswaf:managed:aws:bot-control:bot:name:"
              }
            }
          ]
```

```
        }
      },
      "Action": {
        "Block": {}
      },
      "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "test-rbr"
      }
    }
  ],
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "test-web-acl"
  },
  "Capacity": 82,
  "ManagedByFirewallManager": false,
  "RetrofittedByFirewallManager": false,
  "LabelNamespace": "awswaf:0000000000:webacl:test-web-acl:"
}
```

Rate limit the requests with specific ASNs

To limit the number of requests from specific Autonomous System Numbers (ASNs) based on the IP address of the requests, set the request aggregation to *Custom keys* and provide the aggregation criteria.

The following JSON shows an example of a rule aggregating ASNs derived from forwarded IP addresses found in the `X-Forwarded-For` header. If AWS WAF can't derive an ASN because the IP address is malformed, the fallback behavior is set to `MATCH`.

```
{
    "Name": "test-rbr",
    "Priority": 0,
    "Statement": {
        "RateBasedStatement": {
            "AggregateKeyType": "CUSTOM_KEYS",
            "CustomKeys": [
                {
                    "ASN": {}
                },
```

```
                {
                    "ForwardedIP": {}
                }
            ],
            "EvaluationWindowSec": 300,
            "ForwardedIPConfig": {
                "FallbackBehavior": "MATCH",
                "HeaderName": "X-Forwarded-For"
            },
            "Limit": 2000
        }
    },
    "VisibilityConfig": {
        "CloudWatchMetricsEnabled": true,
        "MetricName": "test-rbr",
        "SampledRequestsEnabled": true
    }
}
```

Listing IP addresses that are being rate limited by rate-based rules

This section explains how to access the list of IP addresses that are currently rate-limited by a rate-based rule by using the CLI, the API, or any of the SDKs.

If your rate-based rule only aggregates on IP address or forwarded IP address, you can retrieve the list of IP addresses that the rule is currently rate limiting. AWS WAF stores these IP addresses in the rule's managed keys list.

> **Note**
>
> This option is only available if you aggregate on only the IP address or only an IP address in a header. If you use the custom keys request aggregation, you can't retrieve a list of rate limited IP addresses, even if you use one of the IP address specifications in your custom keys.

A rate-based rule applies its rule action to requests from the rule's managed keys list that match the rule's scope-down statement. When a rule has no scope-down statement, it applies the action to all requests from the IP addresses that are in the list. The rule action is Block by default, but it can be any valid rule action except for Allow. The maximum number of IP addresses that AWS

WAF can rate limit using a single rate-based rule instance is 10,000. If more than 10,000 addresses exceed the rate limit, AWS WAF limits those with the highest rates.

You can access a rate-based rule's managed keys list using the CLI, the API, or any of the SDKs. This topic covers access using the CLI and APIs. The console doesn't provide access to the list at this time.

For the AWS WAF API, the command is GetRateBasedStatementManagedKeys.

For the AWS WAF CLI, the command is get-rate-based-statement-managed-keys.

The following shows the syntax for retrieving the list of rate limited IP addresses for a rate-based rule that's being used in a protection pack (web ACL) on an Amazon CloudFront distribution.

```
aws wafv2 get-rate-based-statement-managed-keys --scope=CLOUDFRONT --region=us-east-1
 --web-acl-name=WebACLName --web-acl-id=WebACLId --rule-name=RuleName
```

The following shows the syntax for a regional application, an Amazon API Gateway REST API, an Application Load Balancer, an AWS AppSync GraphQL API, an Amazon Cognito user pool, an AWS App Runner service, AWS Amplify, an Amazon Bedrock AgentCore Gateway, or an AWS Verified Access instance.

```
aws wafv2 get-rate-based-statement-managed-keys --scope=REGIONAL --region=region --web-
acl-name=WebACLName --web-acl-id=WebACLId --rule-name=RuleName
```

AWS WAF monitors web requests and manages keys independently for each unique combination of protection pack (web ACL), optional rule group, and rate-based rule. For example, if you define a rate-based rule inside a rule group, and then use the rule group in a protection pack (web ACL), AWS WAF monitors web requests and manages keys for that protection pack (web ACL), rule group reference statement, and rate-based rule instance. If you use the same rule group in a second protection pack (web ACL), AWS WAF monitors web requests and manages keys for this second usage completely independent of your first.

For a rate-based rule that you've defined inside a rule group, you need to provide the name of the rule group reference statement in your request, in addition to the protection pack (web ACL) name and the name of the rate-based rule inside the rule group. The following shows the syntax for a regional application where the rate-based rule is defined inside a rule group, and the rule group is used in a protection pack (web ACL).

```
aws wafv2 get-rate-based-statement-managed-keys --scope=REGIONAL --region=region --web-acl-name=WebACLName --web-acl-id=WebACLId --rule-group-rule-name=RuleGroupRuleName --rule-name=RuleName
```

Using rule group rule statements in AWS WAF

> **Note**
>
> Rule group rule statements are not nestable.

This section describes the rule group rule statements that you can use in your protection pack (web ACL). Rule group protection pack (web ACL) capacity units (WCUs) are set by the rule group owner at the time of creation. For information about WCUs, see Web ACL capacity units (WCUs) in AWS WAF.

Rule group statement	Description	WCUs
Using managed rule group statements	Runs the rules that are defined in the specified managed rule group. You can narrow the scope of requests that the rule group evaluates by adding a scope-down statement. You can't nest a managed rule group statement inside any other statement type.	Defined by the rule group, plus any additional WCUs for a scope-down statement.
Using rule group statements	Runs the rules that are defined in a rule group that you manage. You can't add a scope-down statement to a rule group	You define the WCU limit for the rule group when you create it.

Rule group statement	Description	WCUs
	reference statement for your own rule group. You can't nest a rule group statement inside any other statement type	

Using managed rule group statements in AWS WAF

This section explains how managed rule group rule statements work.

The managed rule group rule statement adds a reference in your protection pack (web ACL) rules list to a managed rule group. You don't see this option under your rule statements on the console, but when you work with the JSON format of your web ACL, any managed rule groups that you've added show up under the protection pack (web ACL) rules as this type.

A managed rule group is either an AWS Managed Rules rule group, most of which are free for AWS WAF customers, or an AWS Marketplace managed rule group. You automatically subscribe to the paid AWS Managed Rules rule groups when you add them to your protection pack (web ACL). You can subscribe to AWS Marketplace managed rule groups through AWS Marketplace. For more information, see Using managed rule groups in AWS WAF.

When you add a rule group to a protection pack (web ACL), you can override the actions of rules in the group to Count or to another rule action. For more information, see Overriding rule group actions in AWS WAF.

You can narrow the scope of the requests that AWS WAF evaluates with the rule group. To do this, you add a scope-down statement inside the rule group statement. For information about scope-down statements, see Using scope-down statements in AWS WAF. This can help you manage how the rule group affects your traffic and can help you contain costs associated with traffic volume when you use the rule group. For information and examples for using scope-down statements with the AWS WAF Bot Control managed rule group, see AWS WAF Bot Control.

Rule statement characteristics

Not nestable – You can't nest this statement type inside other statements, and you can't include it in a rule group. You can include it directly in a protection pack (web ACL).

(Optional) Scope-down statement – This rule type takes an optional scope-down statement, to narrow the scope of the requests that the rule group evaluates. For more information, see Using scope-down statements in AWS WAF.

WCUs – Set for the rule group at creation.

Where to find this rule statement

- **Console** – During the process of creating a protection pack (web ACL), on the **Add rules and rule groups** page, choose **Add managed rule groups**, and then find and select the rule group that you want to use.
- **API** – ManagedRuleGroupStatement

Using rule group statements in AWS WAF

This section explains how rule group rule statements work.

The rule group rule statement adds a reference to your protection pack (web ACL) rules list to a rule group that you manage. You don't see this option under your rule statements on the console, but when you work with the JSON format of your protection pack (web ACL), any of your own rule groups that you've added show up under the protection pack (web ACL) rules as this type. For information about using your own rule groups, see Managing your own rule groups.

When you add a rule group to a protection pack (web ACL), you can override the actions of rules in the group to Count or to another rule action. For more information, see Overriding rule group actions in AWS WAF.

Rule statement characteristics

Not nestable – You can't nest this statement type inside other statements, and you can't include it in a rule group. You can include it directly in a protection pack (web ACL).

WCUs – Set for the rule group at creation.

Where to find this rule statement

- **Console** – During the process of creating a protection pack (web ACL), on the **Add rules and rule groups** page, choose **Add my own rules and rule groups**, **Rule group**, and then add the rule group that you want to use.

- **API** – RuleGroupReferenceStatement

AWS WAF rule groups

This section explains what a rule group is and how it works.

A rule group is a reusable set of rules that you can add to a protection pack (web ACL). For more information about protection packs (web ACLs), see Configuring protection in AWS WAF.

Rule groups fall into the following main categories:

- Your own rule groups, which you create and maintain.
- Managed rule groups that AWS Managed Rules teams create and maintain for you.
- Managed rule groups that AWS Marketplace sellers create and maintain for you.
- Rule groups that are owned and managed by other services like AWS Firewall Manager and Shield Advanced.

Differences between rule groups and protection packs (web ACLs)

Rule groups and protection packs (web ACLs) both contain rules, which are defined in the same manner in both places. Rule groups differ from protection packs (web ACLs) in the following ways:

- Rule groups can't contain rule group reference statements.
- You can reuse a single rule group in multiple protection packs (web ACLs) by adding a rule group reference statement to each protection pack (web ACL). You can't reuse a protection pack (web ACL).
- Rule groups don't have default actions. In a protection pack (web ACL), you set a default action for each rule or rule group that you include. Each individual rule inside a rule group or protection pack (web ACL) has an action defined.
- You don't directly associate a rule group with an AWS resource. To protect resources using a rule group, you use the rule group in a protection pack (web ACL).
- The system defines a maximum capacity of 5,000 protection pack (web ACL) capacity units (WCUs) for each protection pack (web ACL). Each rule group has a WCU setting that must be set at creation. You can use this setting to calculate the additional capacity requirements that using a rule group would add to your protection pack (web ACL). For more information about WCUs, see Web ACL capacity units (WCUs) in AWS WAF.

For information about rules, see AWS WAF rules.

This section provides guidance for creating and managing your own rule groups, describes the managed rule groups that are available to you, and provides guidance for using managed rule groups.

Topics

- Using managed rule groups in AWS WAF
- Managing your own rule groups
- AWS Marketplace rule groups
- Recognizing rule groups provided by other services

Using managed rule groups in AWS WAF

This section explains what managed rule groups are and how they work.

Managed rule groups are collections of predefined, ready-to-use rules that AWS and AWS Marketplace sellers write and maintain for you. Basic AWS WAF pricing applies to your use of any managed rule group. For AWS WAF pricing information, see AWS WAF Pricing.

- *The AWS Managed Rules rule groups for AWS WAF Bot Control, AWS WAF Fraud Control account takeover prevention (ATP), and AWS WAF Fraud Control account creation fraud prevention (ACFP)* are available for additional fees, beyond the basic AWS WAF charges. For pricing details, see AWS WAF Pricing.
- *All other AWS Managed Rules rule groups* are available to AWS WAF customers at no additional cost.
- *AWS Marketplace rule groups* are available by subscription through AWS Marketplace. Each of these rule groups is owned and managed by the AWS Marketplace seller. For pricing information to use an AWS Marketplace rule group, contact the AWS Marketplace seller.

Some managed rule groups are designed to help protect specific types of web applications like WordPress, Joomla, or PHP. Others offer broad protection against known threats or common web application vulnerabilities, including some of the ones listed in the OWASP Top 10. If you're subject to regulatory compliance like PCI or HIPAA, you might be able to use managed rule groups to satisfy web application firewall requirements.

Automatic updates

Keeping up to date on the constantly changing threat landscape can be time consuming and expensive. Managed rule groups can save you time when you implement and use AWS WAF. Many AWS and AWS Marketplace sellers automatically update managed rule groups and provide new versions of rule groups when new vulnerabilities and threats emerge.

In some cases, AWS is notified of new vulnerabilities before public disclosure, due to its participation in a number of private disclosure communities. In those cases, AWS can update the AWS Managed Rules rule groups and deploy them for you even before a new threat is widely known.

Restricted access to rules in a managed rule group

Each managed rule group provides a comprehensive description of the types of attacks and vulnerabilities that it's designed to protect against. To protect the intellectual property of the rule group providers, you can't view all of the details for the individual rules within a rule group. This restriction also helps to keep malicious users from designing threats that specifically circumvent published rules.

Topics

- Using versioned managed rule groups in AWS WAF
- Working with managed rule groups
- AWS Managed Rules for AWS WAF

Using versioned managed rule groups in AWS WAF

This section explains how versioning is handled for managed rule groups.

Many managed rule group providers use versioning to update a rule group's options and capabilities. Usually, a specific version of a managed rule group is static. Occasionally, a provider might need to update some or all of the static versions of a managed rule group, for example, to respond to an emerging security threat.

When you use a versioned managed rule group in your protection pack (web ACL), you can select the default version and let the provider manage which static version you use, or you can select a specific static version.

Can't find the version you want?

If you don't see a version in a rule group's version listing, the version is probably scheduled for expiration or already expired. After a version is scheduled for expiration, AWS WAF no longer lets you to choose it for the rule group.

SNS notifications for AWS Managed Rules rule groups

The AWS Managed Rules rule groups all provide versioning and SNS update notifications except for the IP reputation rule groups. The AWS Managed Rules rule groups that provide notifications all use the same SNS topic Amazon Resource Name (ARN). To sign up for SNS notifications, see Getting notified of new versions and updates.

Topics

- Version life cycle for managed rule groups
- Version expiration for managed rule groups
- Best practices for handling managed rule group versions

Version life cycle for managed rule groups

Providers handle the following life cycle stages of a managed rule group static version:

- **Release and updates** – A managed rule group provider announces upcoming and new static versions of their managed rule groups through notifications to an Amazon Simple Notification Service (Amazon SNS) topic. Providers might also use the topic to communicate other important information about their rule groups, such as urgent required updates.

  You can subscribe to the rule group's topic and configure how you want to receive notifications. For more information see Getting notified of new versions and updates.
- **Expiration scheduling** – A managed rule group provider schedules older versions of a rule group for expiration. A version that's scheduled to expire cannot be added to your protection pack (web ACL) rules. After expiration is scheduled for a version, AWS WAF tracks the expiration with a countdown metric in Amazon CloudWatch.
- **Version expiration** – If you have a protection pack (web ACL) configured to use an expired version of a managed rule group, then during protection pack (web ACL) evaluation, AWS WAF uses the rule group's default version. Additionally, AWS WAF blocks any updates to the protection pack (web ACL) that don't either remove the rule group or change its version to an unexpired one.

If you use AWS Marketplace managed rule groups, ask the provider for any additional information about version life cycles.

Version expiration for managed rule groups

This section explains how version expiration works for a versioned managed rule group.

If you use a specific version of a rule group, make sure that you don't keep using a version past its expiration date. You can monitor version expiration through the rule group's SNS notifications and through Amazon CloudWatch metrics.

If a version that you're using in a protection pack (web ACL) is expired, AWS WAF blocks any updates to the protection pack (web ACL) that don't include moving the rule group to an unexpired version. You can update the rule group to an available version or remove it from your protection pack (web ACL).

Expiration handling for a managed rule group depends on the rule group provider. For AWS Managed Rules rule groups, an expired version is automatically changed to the rule group's default version. For AWS Marketplace rule groups, ask the provider how they handle expiration.

When the provider creates a new version of the rule group, it sets the version's forecasted lifetime. While the version isn't scheduled to expire, the Amazon CloudWatch metric value is set to the forecasted lifetime setting, and in CloudWatch, you'll see a flat value for the metric. After the provider schedules the metric to expire, the metric value diminishes each day until it reaches zero on the day of expiration. For information about monitoring expiration, see Tracking version expiration.

Best practices for handling managed rule group versions

Follow this best practice guidance for handling versioning when you use a versioned managed rule group.

When you use a managed rule group in your protection pack (web ACL), you can choose to use a specific, static version of the rule group, or you can choose to use the default version:

- **Default version** – AWS WAF always sets the default version to the static version that's currently recommended by the provider. When the provider updates their recommended static version, AWS WAF automatically updates the default version setting for the rule group in your protection pack (web ACL).

  When you use the default version of a managed rule group, do the following as best practice:

- **Subscribe to notifications** – Subscribe to notifications for changes to the rule group and keep an eye on those. Most providers send advanced notification of new static versions and of default version changes. These let you check the effects of a new static version before AWS switches the default version to it. For more information see Getting notified of new versions and updates.
- **Review the effects of static version settings and make adjustments as needed before your default is set to it** – Before your default is set to a new static version, review the effects of the static version on the monitoring and management of your web requests. The new static version might have new rules to review. Look for false positives or other unexpected behavior, in case you need to modify how you use the rule group. You can set rules to count, for example, to stop them from blocking traffic while you figure out how you want to handle the new behavior. For more information, see Testing and tuning your AWS WAF protections.

- **Static version** – If you choose to use a static version, you must manually update the version setting when you're ready to adopt a new version of the rule group.

  When you use a static version of a managed rule group, do the following as best practice:

  - **Keep your version up to date** – Keep your managed rule group as close as you can to the latest version. When a new version is released, test it, adjust settings as needed, and implement it in a timely manner. For information about testing, see Testing and tuning your AWS WAF protections.
  - **Subscribe to notifications** – Subscribe to notifications for changes to the rule group, so you know when your provider releases new static versions. Most providers give advanced notification of version changes. Additionally, your provider might need to update the static version that you're using to close a security loophole or for other urgent reasons. You'll know what's happening if you're subscribed to the provider's notifications. For more information, see Getting notified of new versions and updates.
  - **Avoid version expiration** – Don't allow a static version to expire while you're using it. Provider handling of expired versions can vary and might include forcing an upgrade to an available version or other changes that can have unexpected consequences. Track the AWS WAF expiry metric and set an alarm that gives you a sufficient number of days to successfully upgrade to a supported version. For more information, see Tracking version expiration.

Working with managed rule groups

This section provides guidance for accessing and managing your managed rule groups.

When you add a managed rule group to your protection pack (web ACL), you can choose the same configuration options as you can your own rule groups, plus additional settings.

Through the console, you access managed rule group information during the process of adding and editing the rules in your protection packs (web ACLs). Through the APIs and the command line interface (CLI), you can directly request managed rule group information.

When you use a managed rule group in your protection pack (web ACL), you can edit the following settings:

- **Version** – This is available only if the rule group is versioned. For more information, see Using versioned managed rule groups in AWS WAF.
- **Override rule actions** – You can override the actions for rules in the rule group to any action. Setting them to Count is useful for testing a rule group before using it to manage your web requests. For more information, see Rule group rule action overrides.
- **Scope-down statement** – You can add a scope-down statement, to filter out web requests that you don't want to evaluate with the rule group. For more information, see Using scope-down statements in AWS WAF.
- **Override rule group action** – You can override the action that results from the rule group evaluation, and set it to Count only. This option isn't commonly used. It doesn't alter how AWS WAF evaluates the rules in the rule group. For more information, see Rule group return action override to Count.

To edit the managed rule group settings in your protection pack (web ACL)

- **Console**
  - (Option) When you add the managed rules group to your protection pack (web ACL), you can choose **Edit** to view and edit the settings.
  - (Option) After you've added the managed rule group into your protection pack (web ACL), from the **protection packs (web ACLs)** page, choose the protection pack (web ACL) you just created. This takes you to the protection pack (web ACL) edit page.
    - Choose **Rules**.
    - Select the rule group, then choose **Edit** to view and edit the settings.
- **APIs and CLI** – Outside of the console, you can manage the managed rule group settings when you create and update the protection pack (web ACL).

Retrieving the list of managed rule groups

You can retrieve the list of managed rule groups that are available for you to use in your protection packs (web ACLs). The list includes the following:

- All AWS Managed Rules rule groups.
- The AWS Marketplace rule groups that you have subscribed to.

> **Note**
>
> For information about subscribing to AWS Marketplace rule groups, see AWS Marketplace rule groups.

When you retrieve the list of managed rule groups, the list you get back depends on the interface that you're using:

- **Console** – Through the console, you can see all managed rule groups, including the AWS Marketplace rule groups that you haven't subscribed to yet. For the ones that you haven't subscribed to yet, the interface provides links that you can follow to subscribe.
- **APIs and CLI** – Outside of the console, your request returns only the rule groups that are available for you to use.

To retrieve the list of managed rule groups

- **Console** – During the process of creating a web ACL, on the **Add rules and rule groups** page, choose **Add managed rule groups**. At the top level, the provider names are listed. Expand each provider listing to see the list of managed rule groups. For versioned rule groups, the information shown at this level is for the default version. When you add a managed rule group to your protection pack (web ACL), the console lists it based on the naming scheme `<Vendor Name>-<Managed Rule Group Name>`.
- **API** –
  - `ListAvailableManagedRuleGroups`
- **CLI** –
  - `aws wafv2 list-available-managed-rule-groups --scope=<CLOUDFRONT|REGIONAL>`

Retrieving the rules in a managed rule group

You can retrieve a list of the rules in a managed rule group. The API and CLI calls return the rules specifications that you can reference in the JSON model or through AWS CloudFormation.

To retrieve the list of rules in a managed rule group

- **Console**
  - (Option) When you add the managed rules group to your protection pack (web ACL), you can choose **Edit** to view the rules.
  - (Option) After you've added the managed rule group into your protection pack (web ACL), from the **protection packs (web ACLs)** page, choose the protection pack (web ACL) you just created. This takes you to the protection pack (web ACL) edit page.
    - Choose **Rules**.
    - Select the rule group you want to see a rules list for, then choose **Edit**. AWS WAF shows the list of rules in the rule group.
- **API** – `DescribeManagedRuleGroup`
- **CLI** – `aws wafv2 describe-managed-rule-group --scope=<CLOUDFRONT|REGIONAL> --vendor-name <vendor> --name <managedrule_name>`

Retrieving the available versions for a managed rule group

The available versions of a managed rule group are versions that haven't yet been scheduled to expire. The list indicates which version is the current default version for the rule group.

To retrieve a list of the available versions of a managed rule group

- **Console**
  - (Option) When you add the managed rule group to your protection pack (web ACL), choose **Edit** to see the rule group's information. Expand the **Version** dropdown to see the list of available versions.
  - (Option) After you've added the managed rule group into your protection pack (web ACL), choose **Edit** on the protection pack (web ACL), and then select and edit the rule group rule. Expand the **Version** dropdown to see the list of available versions.
- **API** –
  - `ListAvailableManagedRuleGroupVersions`

- **CLI** –
  - `aws wafv2 list-available-managed-rule-group-versions --scope=<CLOUDFRONT|REGIONAL> --vendor-name <vendor> --name <managedrule_name>`

Adding a managed rule group to a protection pack (web ACL) through the console

This section explains how to add a managed rule group to a protection pack (web ACL) through the console. This guidance applies to all AWS Managed Rules rule groups and to the AWS Marketplace rule groups that you're subscribed to.

> **Production traffic risk**
>
> Before you deploy changes in your protection pack (web ACL) for production traffic, test and tune them in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune your updated rules in count mode with your production traffic before enabling them. For guidance, see Testing and tuning your AWS WAF protections.

> **Note**
>
> Using more than 1,500 WCUs in a protection pack (web ACL) incurs costs beyond the basic protection pack (web ACL) price. For more information, see Web ACL capacity units (WCUs) in AWS WAF and AWS WAF Pricing.

To add a managed rule group to a protection pack (web ACL) through the console

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. Choose **protection packs (web ACLs)** in the navigation pane.
3. In the **protection packs (web ACLs)** page, from the list of protection packs (web ACLs), select the one that you want to add the rule group to. This takes you to the page for the single protection pack (web ACL).
4. In your protection pack (web ACL)'s page, choose the **Rules** tab.

5. In the **Rules** pane, choose **Add rules**, then choose **Add managed rule groups**.
6. In the **Add managed rule groups** page, expand the selection for your rule group vendor, to see the list of available rule groups.
7. For each rule group that you want to add, choose **Add to protection pack (web ACL)**. If you want to change the protection pack (web ACL)'s configuration for the rule group, choose **Edit**, make your changes, and then choose **Save rule**. For information about the options, see the versioning guidance at Using versioned managed rule groups in AWS WAF and the guidance for using a managed rule group in a protection pack (web ACL) at Using managed rule group statements in AWS WAF.
8. At the bottom of the **Add managed rule groups** page, choose **Add rules**.
9. In the **Set rule priority** page, adjust the order that the rules run as needed, then choose **Save**. For more information, see Setting rule priority.

In your protection pack (web ACL)'s page, the managed rule groups that you've added are listed under the **Rules** tab.

Test and tune any changes to your AWS WAF protections before you use them for production traffic. For information, see Testing and tuning your AWS WAF protections.

Temporary inconsistencies during updates

When you create or change a protection pack (web ACL) or other AWS WAF resources, the changes take a small amount of time to propagate to all areas where the resources are stored. The propagation time can be from a few seconds to a number of minutes.

The following are examples of the temporary inconsistencies that you might notice during change propagation:

- After you create a protection pack (web ACL), if you try to associate it with a resource, you might get an exception indicating that the protection pack (web ACL) is unavailable.
- After you add a rule group to a protection pack (web ACL), the new rule group rules might be in effect in one area where the protection pack (web ACL) is used and not in another.
- After you change a rule action setting, you might see the old action in some places and the new action in others.
- After you add an IP address to an IP set that is in use in a blocking rule, the new address might be blocked in one area while still allowed in another.

Getting notified of new versions and updates to a managed rule group

This section explains how to receive Amazon SNS notifications of new versions and updates.

A managed rule group provider uses SNS notifications to announce rule group changes, like upcoming new versions and urgent security updates.

How to subscribe to SNS notifications

To subscribe to notifications for a rule group, you create an Amazon SNS subscription for the rule group's Amazon SNS topic ARN in the US East (N. Virginia) Region eusc-de-east-1.

For information about how to subscribe, see the Amazon Simple Notification Service Developer Guide.

> **Note**
>
> Create your subscription for the SNS topic only in the eusc-de-east-1 Region.

The versioned AWS Managed Rules rule groups all use the same SNS topic Amazon Resource Name (ARN). For more information about AWS Managed Rules rule group notifications, see Deployment notifications.

Where to find the Amazon SNS topic ARN for a managed rule group

AWS Managed Rules rule groups use a single SNS topic ARN, so you can retrieve the topic ARN from one of the rule groups and subscribe to it to get notifications for all of the AWS Managed Rules rule groups that provide SNS notifications.

- **Console**
  - (Option) When you add the managed rule group to your protection pack (web ACL), choose **Edit** to see the rule group's information, which includes the rule group's Amazon SNS topic ARN.
  - (Option) After you've added the managed rule group into your protection pack (web ACL), choose **Edit** on the protection pack (web ACL), and then select and edit the rule group rule to see the rule group's Amazon SNS topic ARN.
- **API** – `DescribeManagedRuleGroup`
- **CLI** – `aws wafv2 describe-managed-rule-group --scope=<CLOUDFRONT|REGIONAL> --vendor-name <vendor> --name <managedrule_name>`

For general information about Amazon SNS notification formats and how to filter the notifications that you receive, see Parsing message formats and Amazon SNS subscription filter policies in the Amazon Simple Notification Service Developer Guide.

Tracking a rule group's version expiration

This section explains how to monitor expiration scheduling for a managed rule group through Amazon CloudWatch.

If you use a specific version of a rule group, make sure that you don't keep using a version past its expiration date.

> **Tip**
>
> Sign up for Amazon SNS notifications for managed rule groups, and keep current with managed rule group versions. You'll benefit from the most up-to-date protections from the rule group and stay ahead of expiration. For information, see Getting notified of new versions and updates.

To monitor expiration scheduling for a managed rule group through Amazon CloudWatch

1. In CloudWatch, locate the expiry metrics from AWS WAF for your managed rule group. The metrics have the following metric names and dimensions:
   - Metric name: DaysToExpiry
   - Metric dimensions: Region, ManagedRuleGroup, Vendor, and Version

   If you have a managed rule group in your protection pack (web ACL) that's evaluating traffic, you will get a metric for it. The metric isn't available for rule groups that you don't use.
2. Set an alarm on the metrics that you're interested in, so that you're notified in time to switch to a newer version of the rule group.

For information about using Amazon CloudWatch metrics and configuring alarms, see the Amazon CloudWatch User Guide.

Example managed rule group configurations in JSON and YAML

This section provides example managed rule group configurations.

The API and CLI calls return a list of all rules in the managed rule group that you can reference in the JSON model or through AWS CloudFormation.

JSON

You can reference and modify managed rule groups within a rule statement using JSON. The following listing shows the AWS Managed Rules rule group, `AWSManagedRulesCommonRuleSet`, in JSON format. The RuleActionOverrides specification lists a rule whose action has been overridden to Count.

```
{
    "Name": "AWS-AWSManagedRulesCommonRuleSet",
    "Priority": 0,
    "Statement": {
      "ManagedRuleGroupStatement": {
        "VendorName": "AWS",
        "Name": "AWSManagedRulesCommonRuleSet",
        "RuleActionOverrides": [

          {

            "ActionToUse": {

              "Count": {}

            },

            "Name": "NoUserAgent_HEADER"

          }

        ],
        "ExcludedRules": []
      }
    },
    "OverrideAction": {
      "None": {}
    },
    "VisibilityConfig": {
      "SampledRequestsEnabled": true,
      "CloudWatchMetricsEnabled": true,
      "MetricName": "AWS-AWSManagedRulesCommonRuleSet"
    }
```

```
}
```

YAML

You can reference and modify managed rule groups within a rule statement using the CloudFormation YAML template. The following listing shows the AWS Managed Rules rule group, `AWSManagedRulesCommonRuleSet`, in CloudFormation template. The RuleActionOverrides specification lists a rule whose action has been overridden to Count.

```
Name: AWS-AWSManagedRulesCommonRuleSet
Priority: 0
Statement:
  ManagedRuleGroupStatement:
    VendorName: AWS
    Name: AWSManagedRulesCommonRuleSet
    RuleActionOverrides:
    - ActionToUse:
        Count: {}
      Name: NoUserAgent_HEADER
    ExcludedRules: []
OverrideAction:
  None: {}
VisibilityConfig:
  SampledRequestsEnabled: true
  CloudWatchMetricsEnabled: true
  MetricName: AWS-AWSManagedRulesCommonRuleSet
```

AWS Managed Rules for AWS WAF

This section explains what AWS Managed Rules for AWS WAF is.

AWS Managed Rules for AWS WAF is a managed service that provides protection against application vulnerabilities or other unwanted traffic. You have the option of selecting one or more rule groups from AWS Managed Rules for each web ACL, up to the maximum protection pack (web ACL) capacity unit (WCU) limit.

Mitigating false positives and testing rule group changes

Before using any managed rule group in production, test it in a non-production environment according to the guidance at Testing and tuning your AWS WAF protections. Follow the testing and tuning guidance when you add a rule group to your protection pack (web ACL), to test a new version of a rule group, and whenever a rule group isn't handling your web traffic as you need it to.

Shared security responsibilities

AWS Managed Rules are designed to protect you from common web threats. When used in accordance with the documentation, AWS Managed Rules rule groups add another layer of security for your applications. However, AWS Managed Rules rule groups aren't intended as a replacement for your security responsibilities, which are determined by the AWS resources that you select. Refer to the Shared Responsibility Model to ensure that your resources in AWS are properly protected.

> **Important**
>
> AWS Managed Rules are designed to protect you from common web threats. When used in accordance with the documentation, AWS Managed Rules rule groups add another layer of security for your applications. However, AWS Managed Rules rule groups aren't intended as a replacement for your security responsibilities, which are determined by the AWS resources that you select. Refer to the Shared Responsibility Model to ensure that your resources in AWS are properly protected.

AWS Managed Rules rule groups list

This section provides a list of available AWS Managed Rules rule groups.

This section describes the most recent versions of the AWS Managed Rules rule groups. You see these on the console when you add a managed rule group to your protection pack (web ACL). Through the API, you can retrieve this list along with the AWS Marketplace rule groups that you're subscribed to by calling `ListAvailableManagedRuleGroups`.

> **Note**
>
> For information about retrieving an AWS Managed Rules rule group's versions, see Retrieving the available versions for a managed rule group.

All AWS Managed Rules rule groups support labeling, and the rule listings in this section include label specifications. You can retrieve the labels for a managed rule group through the API by calling `DescribeManagedRuleGroup`. The labels are listed in the AvailableLabels property in the response. For information about labeling, see Web request labeling in AWS WAF.

Test and tune any changes to your AWS WAF protections before you use them for production traffic. For information, see Testing and tuning your AWS WAF protections.

AWS Managed Rules rule groups

- Baseline rule groups
  - Core rule set (CRS) managed rule group
  - Admin protection managed rule group
  - Known bad inputs managed rule group
- Use-case specific rule groups
  - SQL database managed rule group
  - Linux operating system managed rule group
  - POSIX operating system managed rule group
  - Windows operating system managed rule group
  - PHP application managed rule group
  - WordPress application managed rule group
- IP reputation rule groups
  - Amazon IP reputation list managed rule group
  - Anonymous IP list managed rule group
- AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group
  - Considerations for using this rule group
  - Labels added by this rule group
    - Token labels
    - ACFP labels
  - Account creation fraud prevention rules listing
- AWS WAF Fraud Control account takeover prevention (ATP) rule group
  - Considerations for using this rule group
  - Labels added by this rule group
    - Token labels
    - ATP labels
  - Account takeover prevention rules listing
- AWS WAF Bot Control rule group
  - Protection levels
  - Considerations for using this rule group

- Labels added by this rule group
  - Token labels
  - Bot Control labels
- Bot Control rules listing
- AWS WAF Distributed Denial of Service (DDoS) prevention rule group
  - Considerations for using this rule group
  - Labels added by this rule group
    - Token labels
    - Anti-DDoS labels
  - Anti-DDoS rules listing

Baseline rule groups

Baseline managed rule groups provide general protection against a wide variety of common threats. Choose one or more of these rule groups to establish baseline protection for your resources.

Core rule set (CRS) managed rule group

VendorName: `AWS`, Name: `AWSManagedRulesCommonRuleSet`, WCU: 700

> **Note**
>
> This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup. The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
> If you need more information than you find here, contact the AWS Support Center.

The core rule set (CRS) rule group contains rules that are generally applicable to web applications. This provides protection against exploitation of a wide range of vulnerabilities, including some of the high risk and commonly occurring vulnerabilities described in OWASP publications such as OWASP Top 10. Consider using this rule group for any AWS WAF use case.

This managed rule group adds labels to the web requests that it evaluates, which are available to rules that run after this rule group in your protection pack (web ACL). AWS WAF also records the labels to Amazon CloudWatch metrics. For general information about labels and label metrics, see Web request labeling and Label metrics and dimensions.

Rule name	Description and label
`NoUserAgent_HEADER`	Inspects for requests that are missing the HTTP `User-Agent` header. Rule action: Block Label: `awswaf:managed:aws:core-rule-set:NoUserAgent_Header`
`UserAgent_BadBots_HEADER`	Inspects for common `User-Agent` header values that indicate that the request is a bad bot. Example patterns include `nessus`, and `nmap`. For bot management, see also AWS WAF Bot Control rule group. Rule action: Block Label: `awswaf:managed:aws:core-rule-set:BadBots_Header`
`SizeRestrictions_QUERYSTRING`	Inspects for URI query strings that are over 2,048 bytes. Rule action: Block Label: `awswaf:managed:aws:core-rule-set:SizeRestrictions_QueryString`
`SizeRestrictions_Cookie_HEADER`	Inspects for cookie headers that are over 10,240 bytes.

Rule name	Description and label
	Rule action: Block Label: `awswaf:managed:aws:core-rule-set:SizeRestrictions_Cookie_Header`
`SizeRestrictions_BODY`	Inspects for request bodies that are over 8 KB (8,192 bytes). Rule action: Block Label: `awswaf:managed:aws:core-rule-set:SizeRestrictions_Body`
`SizeRestrictions_URIPATH`	Inspects for URI paths that are over 1,024 bytes. Rule action: Block Label: `awswaf:managed:aws:core-rule-set:SizeRestrictions_URIPath`

Rule name	Description and label
`EC2MetaDataSSRF_BODY`	Inspects for attempts to exfiltrate Amazon EC2 metadata from the request body. **Warning** This rule only inspects the request body up to the body size limit for the protection pack (web ACL) and resource type. For Application Load Balancer and AWS AppSync, the limit is fixed at 8 KB. For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway, the default limit is 16 KB and you can increase the limit up to 64 KB in your protection pack (web ACL) configuration. This rule uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:core-rule-set:EC2MetaDataSSRF_Body`

Rule name	Description and label
EC2MetaDataSSRF_COOKIE	Inspects for attempts to exfiltrate Amazon EC2 metadata from the request cookie. Rule action: Block Label: awswaf:managed:aws:core-rule-set:EC2MetaDataSSRF_Cookie
EC2MetaDataSSRF_URIPATH	Inspects for attempts to exfiltrate Amazon EC2 metadata from the request URI path. Rule action: Block Label: awswaf:managed:aws:core-rule-set:EC2MetaDataSSRF_URIPath
EC2MetaDataSSRF_QUERYARGUMENTS	Inspects for attempts to exfiltrate Amazon EC2 metadata from the request query arguments. Rule action: Block Label: awswaf:managed:aws:core-rule-set:EC2MetaDataSSRF_QueryArguments
GenericLFI_QUERYARGUMENTS	Inspects for the presence of Local File Inclusion (LFI) exploits in the query arguments. Examples include path traversal attempts using techniques like ../../. Rule action: Block Label: awswaf:managed:aws:core-rule-set:GenericLFI_QueryArguments

Rule name	Description and label
`GenericLFI_URIPATH`	Inspects for the presence of Local File Inclusion (LFI) exploits in the URI path. Examples include path traversal attempts using techniques like `../../`. Rule action: Block Label: `awswaf:managed:aws:core-rule-set:GenericLFI_URIPath`

Rule name	Description and label
`GenericLFI_BODY`	Inspects for the presence of Local File Inclusion (LFI) exploits in the request body. Examples include path traversal attempts using techniques like `../../`. **Warning** This rule only inspects the request body up to the body size limit for the protection pack (web ACL) and resource type. For Application Load Balancer and AWS AppSync, the limit is fixed at 8 KB. For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway, the default limit is 16 KB and you can increase the limit up to 64 KB in your protection pack (web ACL) configuration. This rule uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:core-rule-set:GenericLFI_Body`

Rule name	Description and label
`RestrictedExtensions_URIPATH`	Inspects for requests whose URI paths contain system file extensions that are unsafe to read or run. Example patterns include extensions like `.log` and `.ini`. Rule action: Block Label: `awswaf:managed:aws:core-rule-set:RestrictedExtensions_URIPath`
`RestrictedExtensions_QUERYARGUMENTS`	Inspects for requests whose query arguments contain system file extensions that are un safe to read or run. Example patterns include extensions like `.log` and `.ini`. Rule action: Block Label: `awswaf:managed:aws:core-rule-set:RestrictedExtensions_QueryArguments`
`GenericRFI_QUERYARGUMENTS`	Inspects the values of all query parameter s for attempts to exploit RFI (Remote File Inclusion) in web applications by embedding URLs that contain IPv4 addresses. Examples include patterns like `http://`, `https://`, `ftp://`, `ftps://`, and `file://`, with an IPv4 host header in the exploit attempt. Rule action: Block Label: `awswaf:managed:aws:core-rule-set:GenericRFI_QueryArguments`

Rule name	Description and label
`GenericRFI_BODY`	Inspects the request body for attempts to exploit RFI (Remote File Inclusion) in web applications by embedding URLs that contain IPv4 addresses. Examples include patterns like `http://`, `https://`, `ftp://`, `ftps://`, and `file://`, with an IPv4 host header in the exploit attempt. **Warning** This rule only inspects the request body up to the body size limit for the protection pack (web ACL) and resource type. For Application Load Balancer and AWS AppSync, the limit is fixed at 8 KB. For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway, the default limit is 16 KB and you can increase the limit up to 64 KB in your protection pack (web ACL) configura tion. This rule uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:core-rul e-set:GenericRFI_Body`

Rule name	Description and label
`GenericRFI_URIPATH`	Inspects the URI path for attempts to exploit RFI (Remote File Inclusion) in web applications by embedding URLs that contain IPv4 addresses. Examples include patterns like `http://`, `https://`, `ftp://`, `ftps://`, and `file://`, with an IPv4 host header in the exploit attempt. Rule action: Block Label: `awswaf:managed:aws:core-rule-set:GenericRFI_URIPath`
`CrossSiteScripting_COOKIE`	Inspects the values of cookie headers for common cross-site scripting (XSS) patterns using the built-in AWS WAF Cross-site scripting attack rule statement. Example patterns include scripts like `<script>alert("hello")</script>` . **Note** The rule match details in the AWS WAF logs is not populated for version 2.0 of this rule group. Rule action: Block Label: `awswaf:managed:aws:core-rule-set:CrossSiteScripting_Cookie`

Rule name	Description and label
`CrossSiteScripting_QUERYARGUMENTS`	Inspects the values of query arguments for common cross-site scripting (XSS) patterns using the built-in AWS WAF Cross-site scripting attack rule statement. Example patterns include scripts like `<script>alert("hello")</script>` . **Note** The rule match details in the AWS WAF logs is not populated for version 2.0 of this rule group. Rule action: Block Label: `awswaf:managed:aws:core-rule-set:CrossSiteScripting_QueryArguments`

Rule name	Description and label
`CrossSiteScripting_BODY`	Inspects the request body for common cross-site scripting (XSS) patterns using the built-in AWS WAF Cross-site scripting attack rule statement. Example patterns include scripts like `<script>alert("hello")</script>` . **Note** The rule match details in the AWS WAF logs is not populated for version 2.0 of this rule group. **Warning** This rule only inspects the request body up to the body size limit for the protection pack (web ACL) and resource type. For Application Load Balancer and AWS AppSync, the limit is fixed at 8 KB. For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway, the default limit is 16 KB and you can increase the limit up to 64 KB in your protection pack (web ACL) configuration. This rule uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF.

Rule name	Description and label
	Rule action: Block Label: `awswaf:managed:aws:core-rule-set:CrossSiteScripting_Body`
`CrossSiteScripting_URIPATH`	Inspects the value of the URI path for common cross-site scripting (XSS) patterns using the built-in AWS WAF Cross-site scripting attack rule statement. Example patterns include scripts like `<script>alert("hello")</script>` . **Note** The rule match details in the AWS WAF logs is not populated for version 2.0 of this rule group. Rule action: Block Label: `awswaf:managed:aws:core-rule-set:CrossSiteScripting_URIPath`

Admin protection managed rule group

VendorName: `AWS`, Name: `AWSManagedRulesAdminProtectionRuleSet`, WCU: 100

Note

This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup.

The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
If you need more information than you find here, contact the AWS Support Center.

The Admin protection rule group contains rules that allow you to block external access to exposed administrative pages. This might be useful if you run third-party software or want to reduce the risk of a malicious actor gaining administrative access to your application.

This managed rule group adds labels to the web requests that it evaluates, which are available to rules that run after this rule group in your protection pack (web ACL). AWS WAF also records the labels to Amazon CloudWatch metrics. For general information about labels and label metrics, see Web request labeling and Label metrics and dimensions.

Rule name	Description and label
`AdminProtection_URIPATH`	Inspects for URI paths that are generally reserved for administration of a web server or application. Example patterns include `sqlmanager` . Rule action: Block Label: `awswaf:managed:aws:admin-protection:AdminProtection_URIPath`

Known bad inputs managed rule group

VendorName: `AWS`, Name: `AWSManagedRulesKnownBadInputsRuleSet`, WCU: 200

Note

This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup.

The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
If you need more information than you find here, contact the AWS Support Center.

The Known bad inputs rule group contains rules to block request patterns that are known to be invalid and are associated with exploitation or discovery of vulnerabilities. This can help reduce the risk of a malicious actor discovering a vulnerable application.

This managed rule group adds labels to the web requests that it evaluates, which are available to rules that run after this rule group in your protection pack (web ACL). AWS WAF also records the labels to Amazon CloudWatch metrics. For general information about labels and label metrics, see Web request labeling and Label metrics and dimensions.

Rule name	Description and label
`JavaDeserializationRCE_HEADER`	Inspects the keys and values of HTTP request headers for patterns indicating Java deserialization Remote Command Execution (RCE) attempts, such as the Spring Core and Cloud Function RCE vulnerabilities (CVE-2022-22963, CVE-2022-22965). Example patterns include `(java.lang.Runtime).getRuntime().exec("whoami")` . **Warning** This rule only inspects the first 8 KB of the request headers or the first 200 headers, whichever limit is reached first, and it uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF.

Rule name	Description and label
	Rule action: Block Label: `awswaf:managed:aws:known-bad-inputs:JavaDeserializationRCE_Header`

Rule name	Description and label
`JavaDeserializationRCE_BODY`	Inspects the request body for patterns indicating Java deserialization Remote Command Execution (RCE) attempts, such as the Spring Core and Cloud Function RCE vulnerabilities (CVE-2022-22963, CVE-2022-22965). Example patterns include `(java.lang.Runtime).getRuntime().exec("whoami")` . **Warning** This rule only inspects the request body up to the body size limit for the protection pack (web ACL) and resource type. For Application Load Balancer and AWS AppSync, the limit is fixed at 8 KB. For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway, the default limit is 16 KB and you can increase the limit up to 64 KB in your protection pack (web ACL) configuration. This rule uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:known-bad-inputs:JavaDeserializationRCE_Body`

Rule name	Description and label
`JavaDeserializationRCE_URIPATH`	Inspects the request URI for patterns indicating Java deserialization Remote Command Execution (RCE) attempts, such as the Spring Core and Cloud Function RCE vulnerabilities (CVE-2022-22963, CVE-2022-22965). Example patterns include `(java.lang.Runtime).getRuntime().exec("whoami")`. Rule action: Block Label: `awswaf:managed:aws:known-bad-inputs:JavaDeserializationRCE_URIPath`
`JavaDeserializationRCE_QUERYSTRING`	Inspects the request query string for patterns indicating Java deserialization Remote Command Execution (RCE) attempts, such as the Spring Core and Cloud Function RCE vulnerabilities (CVE-2022-22963, CVE-2022-22965). Example patterns include `(java.lang.Runtime).getRuntime().exec("whoami")`. Rule action: Block Label: `awswaf:managed:aws:known-bad-inputs:JavaDeserializationRCE_QueryString`

Rule name	Description and label
`Host_localhost_HEADER`	Inspects the host header in the request for patterns indicating localhost. Example patterns include `localhost`. Rule action: Block Label: `awswaf:managed:aws:known-bad-inputs:Host_Localhost_Header`
`PROPFIND_METHOD`	Inspects the HTTP method in the request for `PROPFIND`, which is a method similar to `HEAD`, but with the extra intention to exfiltrate XML objects. Rule action: Block Label: `awswaf:managed:aws:known-bad-inputs:Propfind_Method`
`ExploitablePaths_URIPATH`	Inspects the URI path for attempts to access exploitable web application paths. Example patterns include paths like `web-inf`. Rule action: Block Label: `awswaf:managed:aws:known-bad-inputs:ExploitablePaths_URIPath`

Rule name	Description and label
`Log4JRCE_HEADER`	Inspects the keys and values of request headers for the presence of the Log4j vulnerability (CVE-2021-44228, CVE-2021-45046, CVE-2021-45105) and protects against Remote Code Execution (RCE) attempts. Example patterns include `${jndi:ldap://example.com/}` . **Warning** This rule only inspects the first 8 KB of the request headers or the first 200 headers, whichever limit is reached first, and it uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:known-bad-inputs:Log4JRCE_Header`

Rule name	Description and label
`Log4JRCE_QUERYSTRING`	Inspects the query string for the presence of the Log4j vulnerability (CVE-2021-44228, CVE-2021-45046, CVE-2021-45105) and protects against Remote Code Execution (RCE) attempts. Example patterns include `${jndi:ldap://example.com/}` . Rule action: Block Label: `awswaf:managed:aws:known-bad-inputs:Log4JRCE_QueryString`

Rule name	Description and label
`Log4JRCE_BODY`	Inspects the body for the presence of the Log4j vulnerability (CVE-2021-44228, CVE-2021-45046, CVE-2021-45105) and protects against Remote Code Execution (RCE) attempts. Example patterns include `${jndi:ldap://example.com/}` . **Warning** This rule only inspects the request body up to the body size limit for the protection pack (web ACL) and resource type. For Application Load Balancer and AWS AppSync, the limit is fixed at 8 KB. For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway, the default limit is 16 KB and you can increase the limit up to 64 KB in your protection pack (web ACL) configuration. This rule uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:known-bad-inputs:Log4JRCE_Body`

Rule name	Description and label
`Log4JRCE_URIPATH`	Inspects the URI path for the presence of the Log4j vulnerability (CVE-2021-44228, CVE-2021-45046, CVE-2021-45105) and protects against Remote Code Execution (RCE) attempts. Example patterns include `${jndi:ldap://example.com/}` . Rule action: Block Label: `awswaf:managed:aws:known-bad-inputs:Log4JRCE_URIPath`

Rule name	Description and label
`ReactJSRCE_BODY`	Inspects the request body for patterns indicating presence of CVE-2025-55182. **Warning** This rule only inspects the request body up to the body size limit for the protection pack (web ACL) and resource type. For Application Load Balancer and AWS AppSync, the limit is fixed at 8 KB. For CloudFront, API Gateway, Amazon Cognito, App Runner, and AWS Verified Access, the default limit is 16 KB and you can increase the limit up to 64 KB in your protection pack (web ACL) configuration. This rule uses the `CONTINUE` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:known-bad-inputs:ReactJSRCE_Body`

Use-case specific rule groups

Use-case specific rule groups provide incremental protection for many diverse AWS WAF use cases. Choose the rule groups that apply to your application.

SQL database managed rule group

VendorName: `AWS`, Name: `AWSManagedRulesSQLiRuleSet`, WCU: 200

> **Note**
>
> This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup. The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
> If you need more information than you find here, contact the AWS Support Center.

The SQL database rule group contains rules to block request patterns associated with exploitation of SQL databases, like SQL injection attacks. This can help prevent remote injection of unauthorized queries. Evaluate this rule group for use if your application interfaces with an SQL database.

This managed rule group adds labels to the web requests that it evaluates, which are available to rules that run after this rule group in your protection pack (web ACL). AWS WAF also records the labels to Amazon CloudWatch metrics. For general information about labels and label metrics, see Web request labeling and Label metrics and dimensions.

Rule name	Description and label
`SQLi_QUERYARGUMENTS`	Uses the built-in AWS WAF SQL injection attack rule statement, with sensitivity level set to Low, to inspect the values of all query parameters for patterns that match malicious SQL code. Rule action: Block Label: `awswaf:managed:aws:sql-database:SQLi_QueryArguments`

Rule name	Description and label
`SQLiExtendedPatterns_QUERYARGUMENTS`	Inspects the values of all query parameters for patterns that match malicious SQL code. The patterns this rule inspects for aren't covered by the rule `SQLi_QUERYARGUMENTS`. Rule action: Block Label: `awswaf:managed:aws:sql-database:SQLiExtendedPatterns_QueryArguments`

Rule name	Description and label
`SQLi_BODY`	Uses the built-in AWS WAF SQL injection attack rule statement, with sensitivity level set to Low, to inspect the request body for patterns that match malicious SQL code. **Warning** This rule only inspects the request body up to the body size limit for the protection pack (web ACL) and resource type. For Application Load Balancer and AWS AppSync, the limit is fixed at 8 KB. For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway, the default limit is 16 KB and you can increase the limit up to 64 KB in your protection pack (web ACL) configuration. This rule uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:sql-database:SQLi_Body`

Rule name	Description and label
`SQLiExtendedPatterns_BODY`	Inspects the request body for patterns that match malicious SQL code. The patterns this rule inspects for aren't covered by the rule `SQLi_BODY`. **Warning** This rule only inspects the request body up to the body size limit for the protection pack (web ACL) and resource type. For Application Load Balancer and AWS AppSync, the limit is fixed at 8 KB. For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway, the default limit is 16 KB and you can increase the limit up to 64 KB in your protection pack (web ACL) configuration. This rule uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:sql-database:SQLiExtendedPatterns_Body`

Rule name	Description and label
`SQLiExtendedPatterns_HEADER`	Inspects the request headers for patterns that match malicious SQL code. Rule action: Block Label: `awswaf:managed:aws:sql-database:SQLiExtendedPatterns_Header`
`SQLiExtendedPatterns_URIPATH`	Inspects the request path for patterns that match malicious SQL code. The patterns this rule inspects for aren't covered by the rule `SQLi_URIPATH` . Rule action: Block Label: `awswaf:managed:aws:sql-database:SQLiExtendedPatterns_UriPath`
`SQLi_COOKIE`	Uses the built-in AWS WAF SQL injection attack rule statement, with sensitivity level set to Low, to inspect the request cookie headers for patterns that match malicious SQL code. Rule action: Block Label: `awswaf:managed:aws:sql-database:SQLi_Cookie`

Rule name	Description and label
`SQLi_URIPATH`	Uses the built-in AWS WAF SQL injection attack rule statement, with sensitivity level set to Low, to inspect the request cookie headers for patterns that match malicious SQL code. Rule action: Block Label: `awswaf:managed:aws:sql-database:SQLi_URIPath`

Linux operating system managed rule group

VendorName: `AWS`, Name: `AWSManagedRulesLinuxRuleSet`, WCU: 200

> **Note**
>
> This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup. The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
> If you need more information than you find here, contact the AWS Support Center.

The Linux operating system rule group contains rules that block request patterns associated with the exploitation of vulnerabilities specific to Linux, including Linux-specific Local File Inclusion (LFI) attacks. This can help prevent attacks that expose file contents or run code for which the attacker should not have had access. You should evaluate this rule group if any part of your application runs on Linux. You should use this rule group in conjunction with the POSIX operating system rule group.

This managed rule group adds labels to the web requests that it evaluates, which are available to rules that run after this rule group in your protection pack (web ACL). AWS WAF also records the

labels to Amazon CloudWatch metrics. For general information about labels and label metrics, see Web request labeling and Label metrics and dimensions.

Rule name	Description and label
`LFI_URIPATH`	Inspects the request path for attempts to exploit Local File Inclusion (LFI) vulnerabilities in web applications. Example patterns include files like `/proc/version`, which could provide operating system information to attackers. Rule action: Block Label: `awswaf:managed:aws:linux-os:LFI_URIPath`
`LFI_QUERYSTRING`	Inspects the values of querystring for attempts to exploit Local File Inclusion (LFI) vulnerabilities in web applications. Example patterns include files like `/proc/version`, which could provide operating system information to attackers. Rule action: Block Label: `awswaf:managed:aws:linux-os:LFI_QueryString`
`LFI_HEADER`	Inspects request headers for attempts to exploit Local File Inclusion (LFI) vulnerabilities in web applications. Example patterns include files like `/proc/version`, which could provide operating system information to attackers.

Rule name	Description and label
	Warning This rule only inspects the first 8 KB of the request headers or the first 200 headers, whichever limit is reached first, and it uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:linux-os:LFI_Header`

POSIX operating system managed rule group

VendorName: `AWS`, Name: `AWSManagedRulesUnixRuleSet`, WCU: 100

> **Note**
>
> This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup. The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
> If you need more information than you find here, contact the AWS Support Center.

The POSIX operating system rule group contains rules that block request patterns associated with the exploitation of vulnerabilities specific to POSIX and POSIX-like operating systems, including Local File Inclusion (LFI) attacks. This can help prevent attacks that expose file contents or run code for which the attacker should not have had access. You should evaluate this rule group if any part

of your application runs on a POSIX or POSIX-like operating system, including Linux, AIX, HP-UX, macOS, Solaris, FreeBSD, and OpenBSD.

This managed rule group adds labels to the web requests that it evaluates, which are available to rules that run after this rule group in your protection pack (web ACL). AWS WAF also records the labels to Amazon CloudWatch metrics. For general information about labels and label metrics, see Web request labeling and Label metrics and dimensions.

Rule name	Description and label
`UNIXShellCommandsVariables_QUERYSTRING`	Inspects the values of the query string for attempts to exploit command injection, LFI, and path traversal vulnerabilities in web applications that run on Unix systems. Examples include patterns like `echo $HOME` and `echo $PATH`. Rule action: Block Label: `awswaf:managed:aws:posix-os:UNIXShellCommandsVariables_QueryString`
`UNIXShellCommandsVariables_BODY`	Inspects the request body for attempts to exploit command injection, LFI, and path traversal vulnerabilities in web applications that run on Unix systems. Examples include patterns like `echo $HOME` and `echo $PATH`. **Warning** This rule only inspects the request body up to the body size limit for the protection pack (web ACL) and resource type. For Application Load Balancer and AWS AppSync, the

Rule name	Description and label
	limit is fixed at 8 KB. For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway, the default limit is 16 KB and you can increase the limit up to 64 KB in your protection pack (web ACL) configuration. This rule uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:posix-os:UNIXShellCommandsVariables_Body`

Rule name	Description and label
`UNIXShellCommandsVariables_HEADER`	Inspects all request headers for attempts to exploit command injection, LFI, and path traversal vulnerabilities in web applications that run on Unix systems. Examples include patterns like `echo $HOME` and `echo $PATH`. **Warning** This rule only inspects the first 8 KB of the request headers or the first 200 headers, whichever limit is reached first, and it uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:posix-os:UNIXShellCommandsVariables_Header`

Windows operating system managed rule group

VendorName: `AWS`, Name: `AWSManagedRulesWindowsRuleSet`, WCU: 200

Note

This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup.

The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
If you need more information than you find here, contact the AWS Support Center.

The Windows operating system rule group contains rules that block request patterns associated with the exploitation of vulnerabilities specific to Windows, like remote execution of PowerShell commands. This can help prevent exploitation of vulnerabilities that permit an attacker to run unauthorized commands or run malicious code. Evaluate this rule group if any part of your application runs on a Windows operating system.

This managed rule group adds labels to the web requests that it evaluates, which are available to rules that run after this rule group in your protection pack (web ACL). AWS WAF also records the labels to Amazon CloudWatch metrics. For general information about labels and label metrics, see Web request labeling and Label metrics and dimensions.

Rule name	Description and label
`WindowsShellCommands_HEADER`	Inspects all request headers and blocks WindowsShell command injection attempts in web applications. The match patterns represent WindowsShell commands. Example patterns include `||nslookup` and `;cmd`. Rule action: Block Label: `awswaf:managed:aws:windows-os:WindowsShellCommands_Header`
`WindowsShellCommands_QUERYARGUMENTS`	Inspects the values of all query parameters for WindowsShell command injection attempts in web applications. The match patterns represent WindowsShell commands. Example patterns include `||nslookup` and `;cmd`.

Rule name	Description and label
	Rule action: Block Label: `awswaf:managed:aws:windows-os:WindowsShellCommands_QueryArguments`
`WindowsShellCommands_QUERYSTRING`	Inspects the query string and blocks WindowsShell command injection attempts in web applications. The match patterns represent WindowsShell commands. Example patterns include `||nslookup` and `;cmd`. Rule action: Block Label: `awswaf:managed:aws:windows-os:WindowsShellCommands_QueryString`
`WindowsShellCommands_URIPATH`	Inspects the request path and blocks WindowsShell command injection attempts in web applications. The match patterns represent WindowsShell commands. Example patterns include `||nslookup` and `;cmd`. Rule action: Block Label: `awswaf:managed:aws:windows-os:WindowsShellCommands_UriPath`

Rule name	Description and label
`WindowsShellCommands_BODY`	Inspects the request body for WindowsShell command injection attempts in web applications. The match patterns represent WindowsShell commands. Example patterns include `||nslookup` and `;cmd`. **Warning** This rule only inspects the request body up to the body size limit for the protection pack (web ACL) and resource type. For Application Load Balancer and AWS AppSync, the limit is fixed at 8 KB. For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway, the default limit is 16 KB and you can increase the limit up to 64 KB in your protection pack (web ACL) configuration. This rule uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:windows-os:WindowsShellCommands_Body`

Rule name	Description and label
`PowerShellCommands_COOKIE`	Inspects the request cookie headers for PowerShell command injection attempts in web applications. The match patterns represent PowerShell commands. For example, `Invoke-Expression` . Rule action: Block Label: `awswaf:managed:aws:windows-os:PowerShellCommands_Cookie`
`PowerShellCommands_QUERYARGUMENTS`	Inspects the values of all query parameters for PowerShell command injection attempts in web applications. The match patterns represent PowerShell commands. For example, `Invoke-Expression` . Rule action: Block Label: `awswaf:managed:aws:windows-os:PowerShellCommands_QueryArguments`

Rule name	Description and label
`PowerShellCommands_BODY`	Inspects the request body for PowerShell command injection attempts in web applications. The match patterns represent PowerShell commands. For example, `Invoke-Expression`. **Warning** This rule only inspects the request body up to the body size limit for the protection pack (web ACL) and resource type. For Application Load Balancer and AWS AppSync, the limit is fixed at 8 KB. For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway, the default limit is 16 KB and you can increase the limit up to 64 KB in your protection pack (web ACL) configuration. This rule uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:windows-os:PowerShellCommands_Body`

PHP application managed rule group

VendorName: `AWS`, Name: `AWSManagedRulesPHPRuleSet`, WCU: 100

> **Note**
>
> This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup. The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
> If you need more information than you find here, contact the AWS Support Center.

The PHP application rule group contains rules that block request patterns associated with the exploitation of vulnerabilities specific to the use of the PHP programming language, including injection of unsafe PHP functions. This can help prevent exploitation of vulnerabilities that permit an attacker to remotely run code or commands for which they are not authorized. Evaluate this rule group if PHP is installed on any server with which your application interfaces.

This managed rule group adds labels to the web requests that it evaluates, which are available to rules that run after this rule group in your protection pack (web ACL). AWS WAF also records the labels to Amazon CloudWatch metrics. For general information about labels and label metrics, see Web request labeling and Label metrics and dimensions.

Rule name	Description and label
`PHPHighRiskMethodsVariables_HEADER`	Inspects all headers for PHP script code injection attempts. Example patterns include functions like `fsockopen` and the `$_GET` superglobal variable. **Warning** This rule only inspects the first 8 KB of the request headers or the first 200 headers, whichever limit is reached

Rule name	Description and label
	first, and it uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:php-app:PHPHighRiskMethodsVariables_Header`
`PHPHighRiskMethodsVariables_QUERYSTRING`	Inspects everything after the first ? in the request URL, looking for PHP script code injection attempts. Example patterns include functions like `fsockopen` and the `$_GET` superglobal variable. Rule action: Block Label: `awswaf:managed:aws:php-app:PHPHighRiskMethodsVariables_QueryString`

Rule name	Description and label
`PHPHighRiskMethodsVariables_BODY`	Inspects the values of the request body for PHP script code injection attempts. Example patterns include functions like `fsockopen` and the `$_GET` superglobal variable. **Warning** This rule only inspects the request body up to the body size limit for the protection pack (web ACL) and resource type. For Application Load Balancer and AWS AppSync, the limit is fixed at 8 KB. For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway, the default limit is 16 KB and you can increase the limit up to 64 KB in your protection pack (web ACL) configuration. This rule uses the `Continue` option for oversize content handling. For more information, see Oversize web request components in AWS WAF. Rule action: Block Label: `awswaf:managed:aws:php-app:PHPHighRiskMethodsVariables_Body`

Rule name	Description and label
PHPHighRiskMethodsVariables_URIPATH	Inspects the request path for PHP script code injection attempts. Example patterns include functions like fsockopen and the $_GET superglobal variable. Rule action: Block Label: awswaf:managed:aws:php-app:PHPHighRiskMethodsVariables_URIPath

WordPress application managed rule group

VendorName: AWS, Name: AWSManagedRulesWordPressRuleSet, WCU: 100

> **Note**
>
> This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup. The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
> If you need more information than you find here, contact the AWS Support Center.

The WordPress application rule group contains rules that block request patterns associated with the exploitation of vulnerabilities specific to WordPress sites. You should evaluate this rule group if you are running WordPress. This rule group should be used in conjunction with the SQL database and PHP application rule groups.

This managed rule group adds labels to the web requests that it evaluates, which are available to rules that run after this rule group in your protection pack (web ACL). AWS WAF also records the labels to Amazon CloudWatch metrics. For general information about labels and label metrics, see Web request labeling and Label metrics and dimensions.

Rule name	Description and label
`WordPressExploitableCommands_QUERYSTRING`	Inspects the request query string for high risk WordPress commands that can be exploited in vulnerable installations or plugins. Examples patterns include commands like `do-reset-wordpress` . Rule action: Block Label: `awswaf:managed:aws:wordpress-app:WordPressExploitableCommands_QUERYSTRING`
`WordPressExploitablePaths_URIPATH`	Inspects the request URI path for WordPress files like `xmlrpc.php` , which are known to have easily exploitable vulnerabilities. Rule action: Block Label: `awswaf:managed:aws:wordpress-app:WordPressExploitablePaths_URIPATH`

IP reputation rule groups

IP reputation rule groups block requests based on their source IP address.

> **Note**
>
> These rules use the source IP address from the web request origin. If you have traffic that goes through one or more proxies or load balancers, the web request origin will contain the address of the last proxy, and not the originating address of the client.

Choose one or more of these rule groups if you want to reduce your exposure to bot traffic or exploitation attempts, or if you are enforcing geographic restrictions on your content. For bot management, see also AWS WAF Bot Control rule group.

The rule groups in this category don't provide versioning or SNS update notifications.

Amazon IP reputation list managed rule group

VendorName: `AWS`, Name: `AWSManagedRulesAmazonIpReputationList`, WCU: 25

> **Note**
>
> The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
> If you need more information than you find here, contact the AWS Support Center.

The Amazon IP reputation list rule group contains rules that are based on Amazon internal threat intelligence. This is useful if you would like to block IP addresses typically associated with bots or other threats. Blocking these IP addresses can help mitigate bots and reduce the risk of a malicious actor discovering a vulnerable application.

This managed rule group adds labels to the web requests that it evaluates, which are available to rules that run after this rule group in your protection pack (web ACL). AWS WAF also records the labels to Amazon CloudWatch metrics. For general information about labels and label metrics, see Web request labeling and Label metrics and dimensions.

Rule name	Description and label
`AWSManagedIPReputationList`	Inspects for IP addresses that have been identified as actively engaging in malicious activities. AWS WAF collects the IP address list from various sources, including MadPot, a threat intelligence tool that Amazon uses to protect customers from cybercrime. For more information about MadPot, see https://www.aboutamazon.com/news/aws/amazon-madpot-stops-cybersecurity-crime.

Rule name	Description and label
	Rule action: Block Label: `awswaf:managed:aws:amazon-ip-list:AWSManagedIPReputationList`
`AWSManagedReconnaissanceList`	Inspects for connections from IP addresses that are performing reconnaissance against AWS resources. Rule action: Block Label: `awswaf:managed:aws:amazon-ip-list:AWSManagedReconnaissanceList`
`AWSManagedIPDDoSList`	Inspects for IP addresses that have been identified as actively engaging in DDoS activities. Rule action: Count Label: `awswaf:managed:aws:amazon-ip-list:AWSManagedIPDDoSList`

Anonymous IP list managed rule group

VendorName: `AWS`, Name: `AWSManagedRulesAnonymousIpList`, WCU: 50

> **Note**
>
> The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
> If you need more information than you find here, contact the AWS Support Center.

The Anonymous IP list rule group contains rules to block requests from services that permit the obfuscation of viewer identity. These include requests from VPNs, proxies, Tor nodes, and web hosting providers. This rule group is useful if you want to filter out viewers that might be trying to hide their identity from your application. Blocking the IP addresses of these services can help mitigate bots and evasion of geographic restrictions.

This managed rule group adds labels to the web requests that it evaluates, which are available to rules that run after this rule group in your protection pack (web ACL). AWS WAF also records the labels to Amazon CloudWatch metrics. For general information about labels and label metrics, see Web request labeling and Label metrics and dimensions.

Rule name	Description and label
`AnonymousIPList`	Inspects for a list of IP addresses of sources known to anonymize client information, like TOR nodes, temporary proxies, and other masking services. Rule action: Block Label: `awswaf:managed:aws:anonymous-ip-list:AnonymousIPList`
`HostingProviderIPList`	Inspects for a list of IP addresses from web hosting and cloud providers, which are less likely to source end-user traffic. The IP list does not include AWS IP addresses. Rule action: Block Label: `awswaf:managed:aws:anonymous-ip-list:HostingProviderIPList`

AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group

This section explains what the AWS WAF Fraud Control account creation fraud prevention (ACFP) managed rule group does.

VendorName: AWS, Name: AWSManagedRulesACFPRuleSet, WCU: 50

> **Note**
>
> This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup. The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
> If you need more information than you find here, contact the AWS Support Center.

The AWS WAF Fraud Control account creation fraud prevention (ACFP) managed rule group labels and manages requests that might be part of fraudulent account creation attempts. The rule group does this by inspecting account creation requests that clients send to your application's registration and account creation endpoints.

The ACFP rule group inspects account creation attempts in various ways, to give you visibility and control over potentially malicious interactions. The rule group uses request tokens to gather information about the client browser and about the level of human interactivity in the creation of the account creation request. The rule group detects and manages bulk account creation attempts by aggregating requests by IP address and client session, and aggregating by the provided account information such as the physical address and phone number. Additionally, the rule group detects and blocks the creation of new accounts using credentials that have been compromised, which helps protect the security posture of your application and of your new users.

Considerations for using this rule group

This rule group requires custom configuration, which includes the specification of your application's account registration and account creation paths. Except where noted, the rules in this rule group inspect all requests that your clients send to these two endpoints. To configure and implement this rule group, see the guidance at AWS WAF Fraud Control account creation fraud prevention (ACFP).

> **Note**
>
> You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

This rule group is part of the intelligent threat mitigation protections in AWS WAF. For information, see Intelligent threat mitigation in AWS WAF.

To keep your costs down and to be sure you're managing your web traffic as you want, use this rule group in accordance with the guidance at Best practices for intelligent threat mitigation in AWS WAF.

This rule group isn't available for use with Amazon Cognito user pools. You can't associate a protection pack (web ACL) that uses this rule group with a user pool, and you can't add this rule group to a protection pack (web ACL) that's already associated with a user pool.

Labels added by this rule group

This managed rule group adds labels to the web requests that it evaluates, which are available to rules that run after this rule group in your protection pack (web ACL). AWS WAF also records the labels to Amazon CloudWatch metrics. For general information about labels and label metrics, see Web request labeling and Label metrics and dimensions.

Token labels

This rule group uses AWS WAF token management to inspect and label web requests according to the status of their AWS WAF tokens. AWS WAF uses tokens for client session tracking and verification.

For information about tokens and token management, see Token use in AWS WAF intelligent threat mitigation.

For information about the label components described here, see Label syntax and naming requirements in AWS WAF.

Client session label

The label `awswaf:managed:token:id:`*`identifier`* contains a unique identifier that AWS WAF token management uses to identify the client session. The identifier can change if the client acquires a new token, for example after discarding the token it was using.

> **Note**
>
> AWS WAF doesn't report Amazon CloudWatch metrics for this label.

Browser fingerprint label

The label `awswaf:managed:token:fingerprint:`*`fingerprint-identifier`* contains a robust browser fingerprint identifier that AWS WAF token management computes from various client browser signals. This identifier stays the same across multiple token acquisition attempts. The fingerprint identifier is not unique to a single client.

> **Note**
>
> AWS WAF doesn't report Amazon CloudWatch metrics for this label.

Token status labels: Label namespace prefixes

Token status labels report on the status of the token and of the challenge and CAPTCHA information that it contains.

Each token status label begins with one of the following namespace prefixes:

- `awswaf:managed:token:` – Used to report the general status of the token and to report on the status of the token's challenge information.
- `awswaf:managed:captcha:` – Used to report on the status of the token's CAPTCHA information.

Token status labels: Label names

Following the prefix, the rest of the label provides detailed token status information:

- `accepted` – The request token is present and contains the following:
  - A valid challenge or CAPTCHA solution.
  - An unexpired challenge or CAPTCHA timestamp.
  - A domain specification that's valid for the protection pack (web ACL).

  Example: The label `awswaf:managed:token:accepted` indicates that the web requests's token has a valid challenge solution, an unexpired challenge timestamp, and a valid domain.
- `rejected` – The request token is present but doesn't meet the acceptance criteria.

  Along with the rejected label, token management adds a custom label namespace and name to indicate the reason.
  - `rejected:not_solved` – The token is missing the challenge or CAPTCHA solution.

- `rejected:expired` – The token's challenge or CAPTCHA timestamp has expired, according to your protection pack (web ACL)'s configured token immunity times.
- `rejected:domain_mismatch` – The token's domain isn't a match for your protection pack (web ACL)'s token domain configuration.
- `rejected:invalid` – AWS WAF couldn't read the indicated token.

  Example: The labels `awswaf:managed:captcha:rejected` and `awswaf:managed:captcha:rejected:expired` together indicate that the request didn't have a valid CAPTCHA solve because the CAPTCHA timestamp in the token has exceeded the CAPTCHA token immunity time that's configured in the protection pack (web ACL).

- `absent` – The request doesn't have the token or the token manager couldn't read it.

  Example: The label `awswaf:managed:captcha:absent` indicates that the request doesn't have the token.

ACFP labels

This rule group generates labels with the namespace prefix `awswaf:managed:aws:acfp:` followed by the custom namespace and label name. The rule group might add more than one label to a request.

You can retrieve all labels for a rule group through the API by calling `DescribeManagedRuleGroup`. The labels are listed in the `AvailableLabels` property in the response.

Account creation fraud prevention rules listing

This section lists the ACFP rules in `AWSManagedRulesACFPRuleSet` and the labels that the rule group's rules add to web requests.

All of the rules in this rule group require a web request token, except for the first two `UnsupportedCognitoIDP` and `AllRequests`. For a description of the information that the token provides, see AWS WAF token characteristics.

Except where noted, the rules in this rule group inspect all requests that your clients send to the account registration and account creation page paths that you provide in the rule group configuration. For information about configuring this rule group, see AWS WAF Fraud Control account creation fraud prevention (ACFP).

Note

This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup. The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
If you need more information than you find here, contact the AWS Support Center.

Rule name	Description and label
`UnsupportedCognitoIDP`	Inspects for web traffic going to an Amazon Cognito user pool. ACFP isn't available for use with Amazon Cognito user pools, and this rule helps to ensure that the other ACFP rule group rules are not used to evaluate user pool traffic. Rule action: Block Labels: `awswaf:managed:aws:acfp:unsupported:cognito_idp` and `awswaf:managed:aws:acfp:UnsupportedCognitoIDP`
`AllRequests`	Applies the rule action to requests that access the registration page path. You configure the registration page path when you configure the rule group. By default, this rule applies the Challenge to requests. By applying this action, the rule ensures that the client acquires a challenge token before any requests are evaluated by the rest of the rules in the rule group.

Rule name	Description and label
	Ensure that your end users load the registration page path before they submit an account creation request. Tokens are added to requests by the client application integration SDKs and by the rule actions CAPTCHA and Challenge. For the most efficient token acquisition, we highly recommend that you use the application integration SDKs. For more information, see Client application integrations in AWS WAF. Rule action: Challenge Labels: None

Rule name	Description and label
`RiskScoreHigh`	Inspects for account creation requests with IP addresses or other factors that are considered to be highly suspicious. This evaluation is usually based on multiple contributing factors, which you can see in `risk_score` labels that the rule group adds to the request. Rule action: Block Labels: `awswaf:managed:aws:acfp:risk_score:high` and `awswaf:managed:aws:acfp:RiskScoreHigh` The rule might also apply `medium` or `low` risk score labels to the request. If AWS WAF doesn't succeed at evaluating the risk score for the web request, the rule adds the label `awswaf:managed:aws:acfp:risk_score:evaluation_failed` Additionally, the rule adds labels with the namespace `awswaf:managed:aws:acfp:risk_score:contributor:` that include risk score evaluation status and results for specific risk score contributors, such as IP reputation and stolen credentials evaluations.

Rule name	Description and label
`SignalCredentialCompromised`	Searches the stolen credential database for the credentials that were submitted in the account creation request. This rule ensures that new clients initialize their accounts with positive security posture. **Note** You can add a custom blocking response, to describe the problem to your end user and tell them how to proceed. For information, see ACFP example: Custom response for compromised credentials. Rule action: Block Labels: `awswaf:managed:aws:acfp:signal:credential_compromised` and `awswaf:managed:aws:acfp:SignalCredentialCompromised` The rule group applies the following related label, but takes no action on it, because not all requests in account creation will have credentials: `awswaf:managed:aws:acfp:signal:missing_credential`

Rule name	Description and label
`SignalClientHumanInteractivityAbsentLow`	Inspects the account creation request's token for data that indicates abnormal human interactivity with the application. Human interactivity is detected through interactions such as mouse movements and key presses. If the page has an HTML form, human interactivity includes interactions with the form. **Note** This rule only inspects requests to the account creation path and is only evaluated if you've implemented the application integration SDKs. The SDK implementations passively capture human interactivity and stores the information in the request token. For more information, see AWS WAF token characteristics and Client application integrations in AWS WAF. Rule action: CAPTCHA Labels: None. The rule determines a match based on varying factors, so there is no individual label that applies for every possible match scenario. The rule group can apply one or more of the following labels to requests:

Rule name	Description and label
	awswaf:managed:aws:acfp:signal:client:human_interactivity: *low|medium|high* awswaf:managed:aws:acfp:SignalClientHumanInteractivityAbsent *Low|Medium|High* awswaf:managed:aws:acfp:signal:client:human_interactivity:insufficient_data awswaf:managed:aws:acfp:signal:form_detected .
AutomatedBrowser	Inspects for indicators that the client browser might be automated. Rule action: Block Labels: awswaf:managed:aws:acfp:signal:automated_browser and awswaf:managed:aws:acfp:AutomatedBrowser
BrowserInconsistency	Inspects the request's token for inconsistent browser interrogation data. For more information, see AWS WAF token characteristics. Rule action: CAPTCHA Labels: awswaf:managed:aws:acfp:signal:browser_inconsistency and awswaf:managed:aws:acfp:BrowserInconsistency

Rule name	Description and label
`VolumetricIpHigh`	Inspects for high volumes of account creation requests sent from individual IP addresses. A high volume is more than 20 requests in a 10 minute window. **Note** The thresholds that this rule applies can vary slightly due to latency. For the high volume, a few requests might make it through beyond the limit before the rule action is applied. Rule action: CAPTCHA Labels: `awswaf:managed:aws:acfp:aggregate:volumetric:ip:creation:high` and `awswaf:managed:aws:acfp:VolumetricIpHigh` The rule applies the following labels to requests with medium volumes (more than 15 requests per 10 minute window) and low volumes (more than 10 requests per 10 minute window), but takes no action on them: `awswaf:managed:aws:acfp:aggregate:volumetric:ip:creation:medium` and `awswaf:managed:aws:acfp:aggregate:volumetric:ip:creation:low` .

Rule name	Description and label
`VolumetricSessionHigh`	Inspects for high volumes of account creation requests sent from individual client sessions. A high volume is more than 10 requests in a 30 minute window. **Note** The thresholds that this rule applies can vary slightly due to latency. A few requests might make it through beyond the limit before the rule action is applied. Rule action: Block Labels: `awswaf:managed:aws:acfp:aggregate:volumetric:session:creation:high` and `awswaf:managed:aws:acfp:VolumetricSessionHigh` The rule group applies the following labels to requests with medium volumes (more than 5 requests per 30 minute window) and low volumes (more than 1 request per 30 minute window), but takes no action on them: `awswaf:managed:aws:acfp:aggregate:volumetric:session:creation:medium` and `awswaf:managed:aws:acfp:aggregate:volumetric:session:creation:low` .

Rule name	Description and label
`AttributeUsernameTraversalHigh`	Inspects for a high rate of account creation requests from a single client session that use different usernames. The threshold for a high evaluation is more than 10 requests in 30 minutes. **Note** The thresholds that this rule applies can vary slightly due to latency. A few requests might make it through beyond the limit before the rule action is applied. Rule action: Block Labels: `awswaf:managed:aws:acfp:aggregate:attribute:username_traversal:creation:high` and `awswaf:managed:aws:acfp:AttributeUsernameTraversalHigh` The rule group applies the following labels to requests with medium volumes (more than 5 requests per 30 minute window) and low volumes (more than 1 request per 30 minute window) of username traversal requests, but takes no action on them: `awswaf:managed:aws:acfp:aggregate:attribute:username_traversal:creation:medium` and `awswaf:managed:aws:acfp:aggregate:at`

Rule name	Description and label
	`tribute:username_traversal: creation:low` .

Rule name	Description and label
`VolumetricPhoneNumberHigh`	Inspects for high volumes of account creation requests that use the same phone number. The threshold for a high evaluation is more than 10 requests in 30 minutes. **Note** The thresholds that this rule applies can vary slightly due to latency. A few requests might make it through beyond the limit before the rule action is applied. Rule action: Block Labels: `awswaf:managed:aws:acfp:aggregate:volumetric:phone_number:high` and `awswaf:managed:aws:acfp:VolumetricPhoneNumberHigh` The rule group applies the following labels to requests with medium volumes (more than 5 requests per 30 minute window) and low volumes (more than 1 request per 30 minute window), but takes no action on them: `awswaf:managed:aws:acfp:aggregate:volumetric:phone_number:medium` and `awswaf:managed:aws:acfp:aggregate:volumetric:phone_number:low`.

Rule name	Description and label
`VolumetricAddressHigh`	Inspects for high volumes of account creation requests that use the same physical address. The threshold for a high evaluation is more than 100 requests per 30 minute window. **Note** The thresholds that this rule applies can vary slightly due to latency. A few requests might make it through beyond the limit before the rule action is applied. Rule action: Block Labels: `awswaf:managed:aws:acfp:aggregate:volumetric:address:high` and `awswaf:managed:aws:acfp:VolumetricAddressHigh`

Rule name	Description and label
`VolumetricAddressLow`	Inspects for low and medium volumes of account creation requests that use the same physical address. The threshold for a medium evaluation is more than 50 requests per 30 minute window, and for a low evaluation is more than 10 requests per 30 minute window. The rule applies the action for either medium or low volumes. **Note** The thresholds that this rule applies can vary slightly due to latency. A few requests might make it through beyond the limit before the rule action is applied. Rule action: CAPTCHA Labels: `awswaf:managed:aws:acfp:aggregate:volumetric:address:` *`low|medium`* and `awswaf:managed:aws:acfp:VolumetricAddress` *`Low|Medium`*

Rule name	Description and label
`VolumetricIPSuccessfulResponse`	Inspects for a high volume of successful account creation requests for a single IP add ress. This rule aggregates success responses from the protected resource to account creation requests. The threshold for a high evaluation is more than 10 requests per 10 minute window. This rule helps protect against bulk account creation attempts. It has a lower threshold than the rule `VolumetricIpHigh` , which counts just the requests. If you've configured the rule group to inspect the response body or JSON components, AWS WAF can inspect the first 65,536 bytes (64 KB) of these component types for success or failure indicators. This rule applies the rule action and labeling to new web requests from an IP address, based on the success and failure responses from the protected resource to recent login attempts from the same IP address. You define how to count successes and failures when you configure the rule group. **Note** AWS WAF only evaluates this rule in protection packs (web ACLs) that protect Amazon CloudFront distributions. AWS WAF doesn't inspect

Rule name	Description and label
	responses for web requests that clients send over HTTP/3 (QUIC). **Note** The thresholds that this rule applies can vary slightly due to latency. It's possible for the client to send more successful account creation attempts than are allowed before the rule starts matching on subsequent attempts. Rule action: Block Labels: `awswaf:managed:aws:acfp:aggregate:volumetric:ip:successful_creation_response:high` and `awswaf:managed:aws:acfp:VolumetricIPSuccessfulResponse` The rule group also applies the following related labels to requests, without any associated action. All counts are for a 10-minute window. `awswaf:managed:aws:acfp:aggregate:volumetric:ip:successful_creation_response:medium` for more than 5 successful requests, `awswaf:managed:aws:acfp:aggregate:volumetric:ip:successful_creation_resp`

Rule name	Description and label
	onse:low for more than 1 successful request, awswaf:managed:aws:acfp:aggregate:volumetric:ip:failed_creation_response:high for more than 10 failed requests, awswaf:managed:aws:acfp:aggregate:volumetric:ip:failed_creation_response:medium for more than 5 failed requests, and awswaf:managed:aws:acfp:aggregate:volumetric:ip:failed_creation_response:low for more than 1 failed request.

Rule name	Description and label
`VolumetricSessionSuccessful Response`	Inspects for a low volume of success responses from the protected resource to account creation requests that are being sent from a single client session. This helps to protect against bulk account creation attempts. The threshold for a low evaluation is more than 1 request per 30 minute window. This helps protect against bulk account creation attempts. This rule uses a lower threshold than the rule `Volumetri cSessionHigh` , which tracks only the requests. If you've configured the rule group to inspect the response body or JSON components, AWS WAF can inspect the first 65,536 bytes (64 KB) of these component types for success or failure indicators. This rule applies the rule action and labeling to new web requests from a client session, based on the success and failure responses from the protected resource to recent login attempts from the same client session. You define how to count successes and failures when you configure the rule group. **Note** AWS WAF only evaluates this rule in protection packs (web ACLs) that protect Amazon CloudFront distributions. AWS WAF doesn't inspect

Rule name	Description and label
	responses for web requests that clients send over HTTP/3 (QUIC).
	Note The thresholds that this rule applies can vary slightly due to latency. It's possible for the client to send more failed account creation attempts than are allowed before the rule starts matching on subsequent attempts.
	Rule action: Block Labels: `awswaf:managed:aws:acfp:aggregate:volumetric:session:successful_creation_response:low` and `awswaf:managed:aws:acfp:VolumetricSessionSuccessfulResponse` The rule group also applies the following related labels to requests. All counts are for a 30-minute window. `awswaf:managed:aws:acfp:aggregate:volumetric:session:successful_creation_response:high` for more than 10 successful requests, `awswaf:managed:aws:acfp:aggregate:volumetric:session:successful_creation_response:medium` for more than 5 successful requests, `awswaf:ma`

Rule name	Description and label
	`naged:aws:acfp:aggregate:volumetric:session:failed_creation_response:high` for more than 10 failed requests, `awswaf:managed:aws:acfp:aggregate:volumetric:session:failed_creation_response:medium` for more than 5 failed requests, and `awswaf:managed:aws:acfp:aggregate:volumetric:session:failed_creation_response:low` for more than 1 failed request.
`VolumetricSessionTokenReuseIp`	Inspects account creation requests for the use of a single token among more than 5 distinct IP addresses. **Note** The thresholds that this rule applies can vary slightly due to latency. A few requests might make it through beyond the limit before the rule action is applied. Rule action: Block Labels: `awswaf:managed:aws:acfp:aggregate:volumetric:session:creation:token_reuse:ip` and `awswaf:managed:aws:acfp:VolumetricSessionTokenReuseIp`

AWS WAF Fraud Control account takeover prevention (ATP) rule group

This section explains what the AWS WAF Fraud Control account takeover prevention (ATP) managed rule group does.

VendorName: `AWS`, Name: `AWSManagedRulesATPRuleSet`, WCU: 50

> **Note**
>
> This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup. The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
> If you need more information than you find here, contact the AWS Support Center.

The AWS WAF Fraud Control account takeover prevention (ATP) managed rule group labels and manages requests that might be part of malicious account takeover attempts. The rule group does this by inspecting login attempts that clients send to your application's login endpoint.

- **Request inspection** – ATP gives you visibility and control over anomalous login attempts and login attempts that use stolen credentials, to prevent account takeovers that might lead to fraudulent activity. ATP checks email and password combinations against its stolen credential database, which is updated regularly as new leaked credentials are found on the dark web. ATP aggregates data by IP address and client session, to detect and block clients that send too many requests of a suspicious nature.
- **Response inspection** – For CloudFront distributions, in addition to inspecting incoming login requests, the ATP rule group inspects your application's responses to login attempts, to track success and failure rates. Using this information, ATP can temporarily block client sessions or IP addresses that have too many login failures. AWS WAF performs response inspection asynchronously, so this doesn't increase latency in your web traffic. AWS WAF doesn't inspect responses for web requests that clients send over HTTP/3 (QUIC).

Considerations for using this rule group

This rule group requires specific configuration. To configure and implement this rule group, see the guidance at AWS WAF Fraud Control account takeover prevention (ATP).

This rule group is part of the intelligent threat mitigation protections in AWS WAF. For information, see Intelligent threat mitigation in AWS WAF.

> **Note**
>
> You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

To keep your costs down and to be sure you're managing your web traffic as you want, use this rule group in accordance with the guidance at Best practices for intelligent threat mitigation in AWS WAF.

This rule group isn't available for use with Amazon Cognito user pools. You can't associate a protection pack (web ACL) that uses this rule group with a user pool, and you can't add this rule group to a protection pack (web ACL) that's already associated with a user pool.

Labels added by this rule group

This managed rule group adds labels to the web requests that it evaluates, which are available to rules that run after this rule group in your protection pack (web ACL). AWS WAF also records the labels to Amazon CloudWatch metrics. For general information about labels and label metrics, see Web request labeling and Label metrics and dimensions.

Token labels

This rule group uses AWS WAF token management to inspect and label web requests according to the status of their AWS WAF tokens. AWS WAF uses tokens for client session tracking and verification.

For information about tokens and token management, see Token use in AWS WAF intelligent threat mitigation.

For information about the label components described here, see Label syntax and naming requirements in AWS WAF.

Client session label

The label `awswaf:managed:token:id:`*`identifier`* contains a unique identifier that AWS WAF token management uses to identify the client session. The identifier can change if the client acquires a new token, for example after discarding the token it was using.

> **Note**
>
> AWS WAF doesn't report Amazon CloudWatch metrics for this label.

Browser fingerprint label

The label `awswaf:managed:token:fingerprint:`*`fingerprint-identifier`* contains a robust browser fingerprint identifier that AWS WAF token management computes from various client browser signals. This identifier stays the same across multiple token acquisition attempts. The fingerprint identifier is not unique to a single client.

> **Note**
>
> AWS WAF doesn't report Amazon CloudWatch metrics for this label.

Token status labels: Label namespace prefixes

Token status labels report on the status of the token and of the challenge and CAPTCHA information that it contains.

Each token status label begins with one of the following namespace prefixes:

- `awswaf:managed:token:` – Used to report the general status of the token and to report on the status of the token's challenge information.
- `awswaf:managed:captcha:` – Used to report on the status of the token's CAPTCHA information.

Token status labels: Label names

Following the prefix, the rest of the label provides detailed token status information:

- `accepted` – The request token is present and contains the following:
  - A valid challenge or CAPTCHA solution.
  - An unexpired challenge or CAPTCHA timestamp.
  - A domain specification that's valid for the protection pack (web ACL).

  Example: The label `awswaf:managed:token:accepted` indicates that the web requests's token has a valid challenge solution, an unexpired challenge timestamp, and a valid domain.
- `rejected` – The request token is present but doesn't meet the acceptance criteria.

  Along with the rejected label, token management adds a custom label namespace and name to indicate the reason.
  - `rejected:not_solved` – The token is missing the challenge or CAPTCHA solution.
  - `rejected:expired` – The token's challenge or CAPTCHA timestamp has expired, according to your protection pack (web ACL)'s configured token immunity times.
  - `rejected:domain_mismatch` – The token's domain isn't a match for your protection pack (web ACL)'s token domain configuration.
  - `rejected:invalid` – AWS WAF couldn't read the indicated token.

  Example: The labels `awswaf:managed:captcha:rejected` and `awswaf:managed:captcha:rejected:expired` together indicate that the request didn't have a valid CAPTCHA solve because the CAPTCHA timestamp in the token has exceeded the CAPTCHA token immunity time that's configured in the protection pack (web ACL).
- `absent` – The request doesn't have the token or the token manager couldn't read it.

  Example: The label `awswaf:managed:captcha:absent` indicates that the request doesn't have the token.

ATP labels

The ATP managed rule group generates labels with the namespace prefix `awswaf:managed:aws:atp:` followed by the custom namespace and label name.

The rule group might add any of the following labels in addition to the labels that are noted in the rules listing:

- `awswaf:managed:aws:atp:signal:credential_compromised` – Indicates that the credentials that were submitted in the request are in the stolen credential database.

- `awswaf:managed:aws:atp:aggregate:attribute:suspicious_tls_fingerprint` – Available only for protected Amazon CloudFront distributions. Indicates that a client session has sent multiple requests that used a suspicious TLS fingerprint.
- `awswaf:managed:aws:atp:aggregate:volumetric:session:token_reuse:ip` – Indicates the use of a single token among more than 5 distinct IP addresses. The thresholds that this rule applies can vary slightly due to latency. A few requests might make it through beyond the limit before the label is applied.

You can retrieve all labels for a rule group through the API by calling `DescribeManagedRuleGroup`. The labels are listed in the `AvailableLabels` property in the response.

Account takeover prevention rules listing

This section lists the ATP rules in `AWSManagedRulesATPRuleSet` and the labels that the rule group's rules add to web requests.

> **Note**
>
> This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup. The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
> If you need more information than you find here, contact the AWS Support Center.

Rule name	Description and label
`UnsupportedCognitoIDP`	Inspects for web traffic going to an Amazon Cognito user pool. ATP isn't available for use with Amazon Cognito user pools, and this rule helps to ensure that the other ATP rule group rules are not used to evaluate user pool traffic.

Rule name	Description and label
	Rule action: Block Labels: `awswaf:managed:aws:atp:unsupported:cognito_idp` and `awswaf:managed:aws:atp:UnsupportedCognitoIDP`

Rule name	Description and label
`VolumetricIpHigh`	Inspects for high volumes of requests sent from individual IP addresses. A high volume is more than 20 requests in a 10 minute window. **Note** The thresholds that this rule applies can vary slightly due to latency. For the high volume, a few requests might make it through beyond the limit before the rule action is applied. Rule action: Block Labels: `awswaf:managed:aws:atp:aggregate:volumetric:ip:high` and `awswaf:managed:aws:atp:VolumetricIpHigh` The rule group applies the following labels to requests with medium volumes (more than 15 requests per 10 minute window) and low volumes (more than 10 requests per 10 minute window), but takes no action on them: `awswaf:managed:aws:atp:aggregate:volumetric:ip:medium` and `awswaf:managed:aws:atp:aggregate:volumetric:ip:low`.

Rule name	Description and label
`VolumetricSession`	Inspects for high volumes of requests sent from individual client sessions. The threshold is more than 20 requests per 30 minute window. This inspection only applies when the web request has a token. Tokens are added to requests by the application integration SDKs and by the rule actions CAPTCHA and Challenge. For more information, see Token use in AWS WAF intelligent threat mitigation. **Note** The thresholds that this rule applies can vary slightly due to latency. A few requests might make it through beyond the limit before the rule action is applied. Rule action: Block Labels: `awswaf:managed:aws:atp:aggregate:volumetric:session` and `awswaf:managed:aws:atp:VolumetricSession`

Rule name	Description and label
`AttributeCompromisedCredentials`	Inspects for multiple requests from the same client session that use stolen credentials. Rule action: Block Labels: `awswaf:managed:aws:atp:aggregate:attribute:compromised_credentials` and `awswaf:managed:aws:atp:AttributeCompromisedCredentials`
`AttributeUsernameTraversal`	Inspects for multiple requests from the same client session that use username traversal. Rule action: Block Labels: `awswaf:managed:aws:atp:aggregate:attribute:username_traversal` and `awswaf:managed:aws:atp:AttributeUsernameTraversal`
`AttributePasswordTraversal`	Inspects for multiple requests with the same username that use password traversal. Rule action: Block Labels: `awswaf:managed:aws:atp:aggregate:attribute:password_traversal` and `awswaf:managed:aws:atp:AttributePasswordTraversal`

Rule name	Description and label
`AttributeLongSession`	Inspects for multiple requests from the same client session that use long lasting sessions. The threshold is more than 6 hours of traffic that has at least one login request every 30 minutes. This inspection only applies when the web request has a token. Tokens are added to requests by the application integration SDKs and by the rule actions CAPTCHA and Challenge. For more information, see Token use in AWS WAF intelligent threat mitigation. Rule action: Block Labels: `awswaf:managed:aws:atp:aggregate:attribute:long_session` and `awswaf:managed:aws:atp:AttributeLongSession`
`TokenRejected`	Inspects for requests with tokens that are rejected by AWS WAF token management. This inspection only applies when the web request has a token. Tokens are added to requests by the application integration SDKs and by the rule actions CAPTCHA and Challenge. For more information, see Token use in AWS WAF intelligent threat mitigation. Rule action: Block Labels: None. To check for token rejected, use a label match rule to match on the label: `awswaf:managed:token:rejected` .

Rule name	Description and label
`SignalMissingCredential`	Inspects for requests with credentials that are missing the username or password. Rule action: Block Labels: `awswaf:managed:aws:atp:signal:missing_credential` and `awswaf:managed:aws:atp:SignalMissingCredential`

Rule name	Description and label
`VolumetricIpFailedLoginResponseHigh`	Inspects for IP addresses that have recently been the source of too high a rate of failed login attempts. A high volume is more than 10 failed login requests from an IP address in a 10 minute window. If you've configured the rule group to inspect the response body or JSON components, AWS WAF can inspect the first 65,536 bytes (64 KB) of these component types for success or failure indicators. This rule applies the rule action and labeling to new web requests from an IP address, based on the success and failure responses from the protected resource to recent login attempts from the same IP address. You define how to count successes and failures when you configure the rule group. **Note** AWS WAF only evaluates this rule in protection packs (web ACLs) that protect Amazon CloudFront dis tributions. AWS WAF doesn't inspect responses for web requests that clients send over HTTP/3 (QUIC). **Note** The thresholds that this rule applies can vary slightly due to latency. It's

Rule name	Description and label
	possible for the client to send more failed login attempts than are allowed before the rule starts matching on subsequent attempts. Rule action: Block Labels: `awswaf:managed:aws:atp:aggregate:volumetric:ip:failed_login_response:high` and `awswaf:managed:aws:atp:VolumetricIpFailedLoginResponseHigh` The rule group also applies the following related labels to requests, without any associated action. All counts are for a 10-minute window. `awswaf:managed:aws:atp:aggregate:volumetric:ip:failed_login_response:medium` for more than 5 failed requests, `awswaf:managed:aws:atp:aggregate:volumetric:ip:failed_login_response:low` for more than 1 failed request, `awswaf:managed:aws:atp:aggregate:volumetric:ip:successful_login_response:high` for more than 10 successful requests, `awswaf:managed:aws:atp:aggregate:volumetric:ip:successful_login_response:medium` for more than 5 successful requests, and `awswaf:managed:aws:atp:aggregate:volumetric:ip:successful_login`

Rule name	Description and label
	`_response:low` for more than 1 successful request.

Rule name	Description and label
`VolumetricSessionFailedLoginResponseHigh`	Inspects for client sessions that have recently been the source of too high a rate of failed login attempts. A high volume is more than 10 failed login requests from a client session in a 30 minute window. If you've configured the rule group to inspect the response body or JSON components, AWS WAF can inspect the first 65,536 bytes (64 KB) of these component types for success or failure indicators. This rule applies the rule action and labeling to new web requests from a client session, based on the success and failure responses from the protected resource to recent login attempts from the same client session. You define how to count successes and failures when you configure the rule group. **Note** AWS WAF only evaluates this rule in protection packs (web ACLs) that protect Amazon CloudFront distributions. AWS WAF doesn't inspect responses for web requests that clients send over HTTP/3 (QUIC). **Note** The thresholds that this rule applies can vary slightly due to latency. It's

Rule name	Description and label
	possible for the client to send more failed login attempts than are allowed before the rule starts matching on subsequent attempts. This inspection only applies when the web request has a token. Tokens are added to requests by the application integration SDKs and by the rule actions CAPTCHA and Challenge. For more information, see Token use in AWS WAF intelligent threat mitigation. Rule action: Block Labels: `awswaf:managed:aws:atp:aggregate:volumetric:session:failed_login_response:high` and `awswaf:managed:aws:atp:VolumetricSessionFailedLoginResponseHigh` The rule group also applies the following related labels to requests, without any associated action. All counts are for a 30-minute window. `awswaf:managed:aws:atp:aggregate:volumetric:session:failed_login_response:medium` for more than 5 failed requests, `awswaf:managed:aws:atp:aggregate:volumetric:session:failed_login_response:low` for more than 1 failed request, `awswaf:managed:aws:atp:aggregate:volumetric:session:successful_`

Rule name	Description and label
	`login_response:high` for more than 10 successful requests, `awswaf:managed:aws:atp:aggregate:volumetric:session:successful_login_response:medium` for more than 5 successful requests, and `awswaf:managed:aws:atp:aggregate:volumetric:session:successful_login_response:low` for more than 1 successful request.

AWS WAF Bot Control rule group

This section explains what the Bot Control managed rule group does.

VendorName: `AWS`, Name: `AWSManagedRulesBotControlRuleSet`, WCU: 50

> **Note**
>
> This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup. The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
> If you need to request a new bot classification for Bot Control or require additional information not covered here, contact the AWS Support Center.

The Bot Control managed rule group provides rules that manage requests from bots. Bots can consume excess resources, skew business metrics, cause downtime, and perform malicious activities.

Protection levels

The Bot Control managed rule group provides two levels of protection that you can choose from:

- **Common** – Detects a variety of self-identifying bots, such as web scraping frameworks, search engines, and automated browsers. Bot Control protections at this level identify common bots using traditional bot detection techniques, such as static request data analysis. The rules label traffic from these bots and block the ones that they cannot verify.
- **Targeted** – Includes the common-level protections and adds targeted detection for sophisticated bots that do not self identify. Targeted protections mitigate bot activity using a combination of rate limiting and CAPTCHA and background browser challenges.
  - **`TGT_`** – Rules that provide targeted protection have names that begin with `TGT_`. All targeted protections use detection techniques such as browser interrogation, fingerprinting, and behavior heuristics to identify bad bot traffic.
  - **`TGT_ML_`** – Targeted protection rules that use machine learning have names that begin with `TGT_ML_`. These rules use automated, machine-learning analysis of website traffic statistics to detect anomalous behavior indicative of distributed, coordinated bot activity. AWS WAF analyzes statistics about your website traffic such as timestamps, browser characteristics, and previous URL visited, to improve the Bot Control machine learning model. Machine learning capabilities are enabled by default, but you can disable them in your rule group configuration. When machine learning is disabled, AWS WAF does not evaluate these rules.

The targeted protection level and the AWS WAF rate-based rule statement both provide rate limiting. For a comparison of the two options, see Options for rate limiting in rate-based rules and targeted Bot Control rules.

Considerations for using this rule group

This rule group is part of the intelligent threat mitigation protections in AWS WAF. For information, see Intelligent threat mitigation in AWS WAF.

> **Note**
>
> You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

To keep your costs down and to be sure you're managing your web traffic as you want, use this rule group in accordance with the guidance at Best practices for intelligent threat mitigation in AWS WAF.

We periodically update our machine learning (ML) models for the targeted protection level ML-based rules, to improve bot predictions. The ML-based rules have names that start with `TGT_ML_`. If you notice a sudden and substantial change in the bot predictions made by these rules, contact us through your account manager or open a case at AWS Support Center.

Labels added by this rule group

This managed rule group adds labels to the web requests that it evaluates, which are available to rules that run after this rule group in your protection pack (web ACL). AWS WAF also records the labels to Amazon CloudWatch metrics. For general information about labels and label metrics, see Web request labeling and Label metrics and dimensions.

Token labels

This rule group uses AWS WAF token management to inspect and label web requests according to the status of their AWS WAF tokens. AWS WAF uses tokens for client session tracking and verification.

For information about tokens and token management, see Token use in AWS WAF intelligent threat mitigation.

For information about the label components described here, see Label syntax and naming requirements in AWS WAF.

Client session label

The label `awswaf:managed:token:id:`*`identifier`* contains a unique identifier that AWS WAF token management uses to identify the client session. The identifier can change if the client acquires a new token, for example after discarding the token it was using.

> **Note**
>
> AWS WAF doesn't report Amazon CloudWatch metrics for this label.

Browser fingerprint label

The label `awswaf:managed:token:fingerprint:`*`fingerprint-identifier`* contains a robust browser fingerprint identifier that AWS WAF token management computes from various client browser signals. This identifier stays the same across multiple token acquisition attempts. The fingerprint identifier is not unique to a single client.

> **Note**
>
> AWS WAF doesn't report Amazon CloudWatch metrics for this label.

Token status labels: Label namespace prefixes

Token status labels report on the status of the token and of the challenge and CAPTCHA information that it contains.

Each token status label begins with one of the following namespace prefixes:

- `awswaf:managed:token:` – Used to report the general status of the token and to report on the status of the token's challenge information.
- `awswaf:managed:captcha:` – Used to report on the status of the token's CAPTCHA information.

Token status labels: Label names

Following the prefix, the rest of the label provides detailed token status information:

- `accepted` – The request token is present and contains the following:
  - A valid challenge or CAPTCHA solution.
  - An unexpired challenge or CAPTCHA timestamp.
  - A domain specification that's valid for the protection pack (web ACL).

  Example: The label `awswaf:managed:token:accepted` indicates that the web requests's token has a valid challenge solution, an unexpired challenge timestamp, and a valid domain.
- `rejected` – The request token is present but doesn't meet the acceptance criteria.

  Along with the rejected label, token management adds a custom label namespace and name to indicate the reason.
  - `rejected:not_solved` – The token is missing the challenge or CAPTCHA solution.
  - `rejected:expired` – The token's challenge or CAPTCHA timestamp has expired, according to your protection pack (web ACL)'s configured token immunity times.
  - `rejected:domain_mismatch` – The token's domain isn't a match for your protection pack (web ACL)'s token domain configuration.

- `rejected:invalid` – AWS WAF couldn't read the indicated token.

  Example: The labels `awswaf:managed:captcha:rejected` and `awswaf:managed:captcha:rejected:expired` together indicate that the request didn't have a valid CAPTCHA solve because the CAPTCHA timestamp in the token has exceeded the CAPTCHA token immunity time that's configured in the protection pack (web ACL).

- `absent` – The request doesn't have the token or the token manager couldn't read it.

  Example: The label `awswaf:managed:captcha:absent` indicates that the request doesn't have the token.

Bot Control labels

The Bot Control managed rule group generates labels with the namespace prefix `awswaf:managed:aws:bot-control:` followed by the custom namespace and label name. The rule group might add more than one label to a request.

Each label reflects the Bot Control rule findings:

- `awswaf:managed:aws:bot-control:bot:` – Information about the bot associated with the request.
  - `awswaf:managed:aws:bot-control:bot:name:`*`<name>`* – The bot name, if one is available, for example, the custom namespaces `bot:name:slurp`, `bot:name:googlebot`, and `bot:name:pocket_parser`.
  - `awswaf:managed:aws:bot-control:bot:name:`*`<rfc_name>`* – Identifies the specific bot using the RFC product token from the WBA signature. This is used to create granular custom rules for specific bots. For example, allow `GoogleBot` but rate-limit other crawlers.
  - `awswaf:managed:aws:bot-control:bot:category:`*`<category>`* – The category of bot, as defined by AWS WAF, for example, `bot:category:search_engine` and `bot:category:content_fetcher`.
  - `awswaf:managed:aws:bot-control:bot:account:`*`<hash>`* –For bots using Amazon Bedrock Agent Core only. This label contains an opaque hash uniquely identifying the AWS account that owns the agent. Use this label to create custom rules that allow, block, or rate-limit bots from specific AWS accounts without exposing account IDs in logs.

- `awswaf:managed:aws:bot-control:bot:web_bot_auth:<status>` – Applied when Web Bot Authentication (WBA) validation is performed on a request. The status suffix indicates the verification outcome:
  - `web_bot_auth:verified` – Signature successfully validated against public key directory
  - `web_bot_auth:invalid` – Signature present but cryptograpic validation failed
  - `web_bot_auth:expired` – Signature used an expired cryptograpic key
  - `web_bot_auth:unknown_bot` – Key ID not found in the key directory

  > **Note**
  >
  > When the `web_bot_auth:verified` label is present, the `Category<category_name>` and `TGT_TokenAbsent` rules do not match, allowing verified WBA hosts to proceed.

- `awswaf:managed:aws:bot-control:bot:organization:<organization>` – The bot's publisher, for example, `bot:organization:google`.
- `awswaf:managed:aws:bot-control:bot:verified` – Used to indicate a bot that identifies itself and that Bot Control has been able to verify. This is used for common desirable bots, and can be useful when combined with category labels like `bot:category:search_engine` or name labels like `bot:name:googlebot`.

  > **Note**
  >
  > Bot Control uses the IP address from the web request origin to help determine whether a bot is verified. You can't configure it to use the AWS WAF forwarded IP configuration, to inspect a different IP address source. If you have verified bots that route through a proxy or load balancer, you can add a rule that runs before the Bot Control rule group to help with this. Configure your new rule to use the forwarded IP address and explicitly allow requests from the verified bots. For information about using forwarded IP addresses, see Using forwarded IP addresses in AWS WAF.

- `awswaf:managed:aws:bot-control:bot:vendor:<vendor_name>` – Identifies the vendor or operator of a verified bot. Currently available only for Agentcore. Use to create custom rules that allow or block specific bot vendors regardless of individual bot names.

  - `awswaf:managed:aws:bot-control:bot:user_triggered:verified` – Used to indicate a bot that is similar to a verified bot, but that might be directly invoked by end users. This category of bot is treated by the Bot Control rules like an unverified bot.
  - `awswaf:managed:aws:bot-control:bot:developer_platform:verified` – Used to indicate a bot that is similar to a verified bot, but that is used by developer platforms for scripting, for example Google Apps Script. This category of bot is treated by the Bot Control rules like an unverified bot.
  - `awswaf:managed:aws:bot-control:bot:unverified` – Used to indicate a bot that identifies itself, so it can be named and categorized, but that doesn't publish information that can be used to independently verify its identify. These types of bot signatures can be falsified, and so are treated as unverified.
- `awswaf:managed:aws:bot-control:targeted:<additional-details>` – Used for labels that are specific to the Bot Control targeted protections.
- `awswaf:managed:aws:bot-control:signal:<signal-details>` and `awswaf:managed:aws:bot-control:targeted:signal:<signal-details>` – Used to provide additional information about the request in some situations.

  The following are examples of signal labels. This is not an exhaustive list:

  - `awswaf:managed:aws:bot-control:signal:cloud_service_provider:<CSP>` – Indicates a cloud service provider (CSP) for the request. Examples of CSPs include `aws` for Amazon Web Services infrastructure, `gcp` for Google Cloud Platform (GCP) infrastructure, `azure` for Microsoft Azure cloud services, and `oracle` for Oracle Cloud services.
  - `awswaf:managed:aws:bot-control:targeted:signal:browser_automation_extension` – Indicates the detection of a browser extension that assists in automation, such as Selenium IDE.

    This label is added whenever a user has this type of extension installed, even if they're not actively using it. If you implement a label match rule for this, be aware of this possibility of false positives in your rule logic and action settings. For example, you might use a CAPTCHA action instead of Block or you might combine this label match with other label matches, to increase your confidence that automation is in use.
  - `awswaf:managed:aws:bot-control:signal:automated_browser` – Indicates that the request contains indicators that the client browser might be automated.

- `awswaf:managed:aws:bot-control:targeted:signal:automated_browser` – Indicates that the request's AWS WAF token contains indicators that the client browser might be automated.

You can retrieve all labels for a rule group through the API by calling `DescribeManagedRuleGroup`. The labels are listed in the `AvailableLabels` property in the response.

The Bot Control managed rule group applies labels to a set of verifiable bots that are commonly allowed. The rule group doesn't block these verified bots. If you want, you can block them, or a subset of them by writing a custom rule that uses the labels applied by the Bot Control managed rule group. For more information about this and examples, see AWS WAF Bot Control.

Bot Control rules listing

This section lists the Bot Control rules.

> **Note**
>
> This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup. The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
> If you need to request a new bot classification for Bot Control or require additional information not covered here, contact the AWS Support Center.

Rule name	Description
`CategoryAdvertising`	Inspects for bots that are used for advertising purposes. For example, you might use third-party advertising services that need to programmatically access your website.

Rule name	Description
	Rule action, applied only to unverified bots: Block Labels: `awswaf:managed:aws:bot-control:bot:category:advertising` and `awswaf:managed:aws:bot-control:CategoryAdvertising` For verified bots, the rule group does not match this rule and takes no action, but it adds the bot name and category labeling plus the label `awswaf:managed:aws:bot-control:bot:verified`.
`CategoryArchiver`	Inspects for bots that are used for archiving purposes. These bots crawl the web and capture content for the purposes of creating archives. Rule action, applied only to unverified bots: Block Labels: `awswaf:managed:aws:bot-control:bot:category:archiver` and `awswaf:managed:aws:bot-control:CategoryArchiver` For verified bots, the rule group does not match this rule and takes no action, but it adds the bot name and category labeling plus the label `awswaf:managed:aws:bot-control:bot:verified`.

Rule name	Description
`CategoryContentFetcher`	Inspects for bots that visit the application's website on behalf of a user, to fetch content like RSS feeds or to verify or validate your content. Rule action, applied only to unverified bots: Block Labels: `awswaf:managed:aws:bot-control:bot:category:content_fetcher` and `awswaf:managed:aws:bot-control:CategoryContentFetcher` For verified bots, the rule group does not match this rule and takes no action, but it adds the bot name and category labeling plus the label `awswaf:managed:aws:bot-control:bot:verified`.

Rule name	Description
`CategoryEmailClient`	Inspects for bots that check links within emails that point to the application's website. This can include bots run by businesses and email providers, to verify links in emails and flag suspicious emails. Rule action, applied only to unverified bots: Block Labels: `awswaf:managed:aws:bot-control:bot:category:email_client` and `awswaf:managed:aws:bot-control:CategoryEmailClient` For verified bots, the rule group does not match this rule and takes no action, but it adds the bot name and category labeling plus the label `awswaf:managed:aws:bot-control:bot:verified`.

Rule name	Description
`CategoryHttpLibrary`	Inspects for requests that are generated by bots from the HTTP libraries of various programming languages. These may include API requests that you choose to allow or monitor. Rule action, applied only to unverified bots: Block Labels: `awswaf:managed:aws:bot-control:bot:category:http_library` and `awswaf:managed:aws:bot-control:CategoryHttpLibrary` For verified bots, the rule group does not match this rule and takes no action, but it adds the bot name and category labeling plus the label `awswaf:managed:aws:bot-control:bot:verified`.
`CategoryLinkChecker`	Inspects for bots that check for broken links. Rule action, applied only to unverified bots: Block Labels: `awswaf:managed:aws:bot-control:bot:category:link_checker` and `awswaf:managed:aws:bot-control:CategoryLinkChecker` For verified bots, the rule group does not match this rule and takes no action, but it adds the bot name and category labeling plus the label `awswaf:managed:aws:bot-control:bot:verified`.

Rule name	Description
`CategoryMiscellaneous`	Inspects for miscellaneous bots that don't match other categories. Rule action, applied only to unverified bots: Block Labels: `awswaf:managed:aws:bot-control:bot:category:miscellaneous` and `awswaf:managed:aws:bot-control:CategoryMiscellaneous` For verified bots, the rule group does not match this rule and takes no action, but it adds the bot name and category labeling plus the label `awswaf:managed:aws:bot-control:bot:verified`.

Rule name	Description
`CategoryMonitoring`	Inspects for bots that are used for monitoring purposes. For example, you might use bot monitoring services that periodically ping your application website to monitor things like performance and uptime. Rule action, applied only to unverified bots: Block Labels: `awswaf:managed:aws:bot-control:bot:category:monitoring` and `awswaf:managed:aws:bot-control:CategoryMonitoring` For verified bots, the rule group does not match this rule and takes no action, but it adds the bot name and category labeling plus the label `awswaf:managed:aws:bot-control:bot:verified`.

Rule name	Description
`CategoryPagePreview`	Inspects for bots that generate page previews and link previews when content is shared on messaging platforms, social media, or collaboration tools. Rule action, applied only to unverified bots: Block Labels: `awswaf:managed:aws:bot-control:bot:category:page_preview` and `awswaf:managed:aws:bot-control:CategoryPagePreview` For verified bots, the rule group does not match this rule and takes no action, but it adds the bot name and category labeling plus the label `awswaf:managed:aws:bot-control:bot:verified`.

Rule name	Description
`CategoryScrapingFramework`	Inspects for bots from web scraping frameworks, which are used to automate crawling and extracting content from websites. Rule action, applied only to unverified bots: Block Labels: `awswaf:managed:aws:bot-control:bot:category:scraping_framework` and `awswaf:managed:aws:bot-control:CategoryScrapingFramework` For verified bots, the rule group does not match this rule and takes no action, but it adds the bot name and category labeling plus the label `awswaf:managed:aws:bot-control:bot:verified`.

Rule name	Description
`CategorySearchEngine`	Inspects for search engine bots, which crawl websites to index content and make the information available for search engine results. Rule action, applied only to unverified bots: Block Labels: `awswaf:managed:aws:bot-control:bot:category:search_engine` and `awswaf:managed:aws:bot-control:CategorySearchEngine` For verified bots, the rule group does not match this rule and takes no action, but it adds the bot name and category labeling plus the label `awswaf:managed:aws:bot-control:bot:verified`.

Rule name	Description
`CategorySecurity`	Inspects for bots that scan web applications for vulnerabilities or that perform security audits. For example, you might use a third-party security vendor that scans, monitors, or audits your web application's security. Rule action, applied only to unverified bots: Block Labels: `awswaf:managed:aws:bot-control:bot:category:security` and `awswaf:managed:aws:bot-control:CategorySecurity` For verified bots, the rule group does not match this rule and takes no action, but it adds the bot name and category labeling plus the label `awswaf:managed:aws:bot-control:bot:verified`.

Rule name	Description
`CategorySeo`	Inspects for bots that are used for search engine optimization. For example, you might use search engine tools that crawl your site to help you improve your search engine rankings. Rule action, applied only to unverified bots: Block Labels: `awswaf:managed:aws:bot-control:bot:category:seo` and `awswaf:managed:aws:bot-control:CategorySeo` For verified bots, the rule group does not match this rule and takes no action, but it adds the bot name and category labeling plus the label `awswaf:managed:aws:bot-control:bot:verified`.

Rule name	Description
`CategorySocialMedia`	Inspects for bots that are used by social media platforms to provide content summaries when users share your content. Rule action, applied only to unverified bots: Block Labels: `awswaf:managed:aws:bot-control:bot:category:social_media` and `awswaf:managed:aws:bot-control:CategorySocialMedia` For verified bots, the rule group does not match this rule and takes no action, but it adds the bot name and category labeling plus the label `awswaf:managed:aws:bot-control:bot:verified`.

Rule name	Description
`CategoryWebhooks`	Inspects for bots that deliver automated notifications and data updates from one application to another through HTTP callbacks . Rule action, applied only to unverified bots: Block Labels: `awswaf:managed:aws:bot-control:bot:category:webhooks` and `awswaf:managed:aws:bot-control:CategoryWebhooks` For verified bots, the rule group does not match this rule and takes no action, but it adds the bot name and category labeling plus the label `awswaf:managed:aws:bot-control:bot:verified` .

Rule name	Description
`CategoryAI`	Inspects for artificial intelligence (AI) bots. **Note** This rule applies the action to all matches, regardless of whether the bots are verified or unverified. Rule action: Block Labels: `awswaf:managed:aws:bot-control:bot:category:ai` and `awswaf:managed:aws:bot-control:CategoryAI` For verified bots, the rule group matches this rule and takes an action. It additionally adds the bot name and category labeling, the rule labeling, plus the label `awswaf:managed:aws:bot-control:bot:verified` .

Rule name	Description
`SignalAutomatedBrowser`	Inspects requests that are not from verified bots for indicators that the client browser might be automated. Automated browsers can be used for testing or scraping. For example, you might use these types of browsers to monitor or verify your application website. Rule action: Block Labels: `awswaf:managed:aws:bot-control:signal:automated_browser` and `awswaf:managed:aws:bot-control:SignalAutomatedBrowser` For verified bots, the rule group does not match this rule and does not apply any signal or rule labels.
`SignalKnownBotDataCenter`	Inspects requests that are not from verified bots for indicators of data centers that are typically used by bots. Rule action: Block Labels: `awswaf:managed:aws:bot-control:signal:known_bot_data_center` and `awswaf:managed:aws:bot-control:SignalKnownBotDataCenter` For verified bots, the rule group does not match this rule and does not apply any signal or rule labels.

Rule name	Description
`SignalNonBrowserUserAgent`	Inspects requests that are not from verified bots for user agent strings that don't seem to be from a web browser. This category can include API requests. Rule action: Block Labels: `awswaf:managed:aws:bot-control:signal:non_browser_user_agent` and `awswaf:managed:aws:bot-control:SignalNonBrowserUserAgent` For verified bots, the rule group does not match this rule and does not apply any signal or rule labels.

Rule name	Description
`TGT_VolumetricIpTokenAbsent`	Inspects requests that are not from verified bots with 5 or more requests from a single client in the last 5 minutes that don't include a valid challenge token. For information about tokens, see Token use in AWS WAF intelligent threat mitigation. **Note** It's possible for this rule to match on a request that has a token if requests from the same client have recently been missing tokens. The threshold that this rule applies can vary slightly due to latency. This rule handles missing tokens differently from the token labeling: `awswaf:managed:token:absent`. The token labeling labels individual requests that don't have a token. This rule maintains a count of requests that are missing their token for each client IP, and it matches against clients that go over the limit. Rule action: Challenge Labels: `awswaf:managed:aws:bot-control:targeted:aggregate:volumetric:ip:token_absent` and `awswaf:managed:aws:bot-control:TGT_VolumetricIpTokenAbsent`

Rule name	Description
`TGT_TokenAbsent`	Inspects requests that are not from verified bots that don't include a valid challenge token. For information about tokens, see Token use in AWS WAF intelligent threat mitigation. Rule action: Count Labels: `awswaf:managed:aws:bot-control:TGT_TokenAbsent`

Rule name	Description
`TGT_VolumetricSession`	Inspects for an abnormally high number of requests that are not from verified bots that come from a single client session in a 5 minute window. The evaluation is based on a comparison to standard volumetric baselines that AWS WAF maintains using historic traffic patterns. This inspection only applies when the web request has a token. Tokens are added to requests by the application integration SDKs and by the rule actions CAPTCHA and Challenge. For more information, see Token use in AWS WAF intelligent threat mitigation. **Note** This rule can take 5 minutes to go into effect after you enable it. Bot Control identifies anomalous behavior in your web traffic by comparing the current traffic to traffic baselines that AWS WAF computes. Rule action: CAPTCHA Labels: `awswaf:managed:aws:bot-control:targeted:aggregate:volumetric:session:high` and `awswaf:managed:aws:bot-control:TGT_VolumetricSession` The rule group applies the following labels to medium volume and lower volume

Rule name	Description
	requests that are above a minimum threshold. For these levels, the rule takes no action, regardless of whether the client is verified: `awswaf:managed:aws:bot-control:targeted:aggregate:volumetric:session:medium` and `awswaf:managed:aws:bot-control:targeted:aggregate:volumetric:session:low`.

Rule name	Description
`TGT_VolumetricSessionMaximum`	Inspects for an abnormally high number of requests that are not from verified bots that come from a single client session in a 5 minute window. The evaluation is based on a comparison to standard volumetric baselines that AWS WAF maintains using historic traffic patterns. This rule indicates the maximum confidence in the assessment. This inspection only applies when the web request has a token. Tokens are added to requests by the application integration SDKs and by the rule actions CAPTCHA and Challenge. For more information, see Token use in AWS WAF intelligent threat mitigation. **Note** This rule can take 5 minutes to go into effect after you enable it. Bot Control identifies anomalous behavior in your web traffic by comparing the current traffic to traffic baselines that AWS WAF computes. Rule action: Block Labels: `awswaf:managed:aws:bot-control:targeted:aggregate:volumetric:session:maximum` and `awswaf:managed:aws:bot-cont`

Rule name	Description
	`rol:TGT_VolumetricSessionMa ximum`
`TGT_SignalAutomatedBrowser`	Inspects the tokens of requests that are not from verified bots for indicators that the client browser might be automated. For more i nformation, see AWS WAF token character istics. This inspection only applies when the web request has a token. Tokens are added to requests by the application integration SDKs and by the rule actions CAPTCHA and Challenge. For more information, see Token use in AWS WAF intelligent threat mitigation. Rule action: CAPTCHA Labels: `awswaf:managed:aws:bot-control:targeted:signal:automated_browser` and `awswaf:managed:aws:bot-control:TGT_SignalAutomatedBrowser`

Rule name	Description
`TGT_SignalBrowserAutomationExtension`	Inspects requests that are not from verified bots that indicate the presence of a browser extension that assists in automation, such as Selenium IDE. This rule matches whenever a user has this type of extension installed, even if they're not actively using it. This inspection only applies when the web request has a token. Tokens are added to requests by the application integration SDKs and by the rule actions CAPTCHA and Challenge. For more information, see Token use in AWS WAF intelligent threat mitigation. Rule action: CAPTCHA Labels: `awswaf:managed:aws:bot-control:targeted:signal:browser_automation_extension` and `awswaf:managed:aws:bot-control:TGT_SignalBrowserAutomationExtension`

Rule name	Description
`TGT_SignalBrowserInconsistency`	Inspects requests that are not from verified bots for inconsistent browser interrogation data. For more information, see AWS WAF token characteristics. This inspection only applies when the web request has a token. Tokens are added to requests by the application integration SDKs and by the rule actions CAPTCHA and Challenge. For more information, see Token use in AWS WAF intelligent threat mitigation. Rule action: CAPTCHA Labels: `awswaf:managed:aws:bot-control:targeted:signal:browser_inconsistency` and `awswaf:managed:aws:bot-control:TGT_SignalBrowserInconsistency`

Rule name	Description
`TGT_ML_CoordinatedActivityLow` , `TGT_ML_CoordinatedActivityMedium` , `TGT_ML_CoordinatedActivityHigh`	Inspects requests that are not from verified bots for anomalous behavior that's consistent with distributed, coordinated bot activity. The rule levels indicate the level of confidence that a group of requests are participants in a coordinated attack. **Note** These rules only run if the rule group is configured to use machine learning (ML). For information about configuring this choice, see Adding the AWS WAF Bot Control managed rule group to your web ACL. **Note** The thresholds that these rules apply can vary slightly due to latency. A few requests might make it through beyond the limit before the rule action is applied. AWS WAF performs this inspection through machine learning analysis of website traffic statistics. AWS WAF analyzes web traffic every few minutes and optimizes the analysis for the detection of low intensity, long-duration bots that are distributed across many IP addresses.

Rule name	Description
	These rules might match on a very small number of requests before determining that a coordinated attack is not underway. So if you see just a match or two, the results might be false positives. If you see a lot of matches coming out of these rules however, then you're probably experiencing a coordinated attack. **Note** These rules can take up to 24 hours to go into effect after you enable the Bot Control targeted rules with the ML option. Bot Control identifies anomalous behavior in your web traffic by comparing the current traffic to traffic baselines that AWS WAF has computed. AWS WAF only computes the baselines while you're using the Bot Control targeted rules with the ML option, and it can take up to 24 hours to establish meaningful baselines. We periodically update our machine learning models for these rules, to improve bot predictions. If you notice a sudden and substantial change in the bot predictions that these rules make, contact your account manager or open a case at AWS Support Center. Rule actions:

Rule name	Description
	• Low: Challenge • Medium: CAPTCHA • High: CAPTCHA Labels: `awswaf:managed:aws:bot-control:targeted:aggregate:coordinated_activity:` *`low|medium|high`* and `awswaf:managed:aws:bot-control:TGT_ML_CoordinatedActivity` *`Low|Medium|High`*

Rule name	Description
`TGT_TokenReuseIpLow` , `TGT_TokenReuseIpMedium` , `TGT_TokenReuseIpHigh`	Inspects requests that are not from verified bots for the use of a single token among multiple IPs in the last 5 minutes. Each level has a limit on the number of distinct IPs: • Low: more than 2 • Medium: more than 5 • High: more than 8 **Note** The thresholds that these rules apply can vary slightly due to latency. A few requests might make it through beyond the limit before the rule action is applied. Rule actions: • Low: Count • Medium: CAPTCHA • High: Block Labels: `awswaf:managed:aws:bot-control:targeted:aggregate:volumetric:session:token_reuse:ip:` *`low|medium|high`* and `awswaf:managed:aws:bot-control:TGT_TokenReuseIp` *`Low|Medium|High`*

Rule name	Description
`TGT_TokenReuseCountryLow` , `TGT_TokenReuseCountryMedium` , `TGT_TokenReuseCountryHigh`	Inspects requests that are not from verified bots for the use of a single token across multiple countries in the last 5 minutes. Each level has a limit on the number of distinct countries: • Low: more than 1 • Medium: more than 2 • High: more than 3 **Note** The thresholds that these rules apply can vary slightly due to latency. A few requests might make it through beyond the limit before the rule action is applied. Rule actions: • Low: Count • Medium: CAPTCHA • High: Block Labels: `awswaf:managed:aws:bot-control:targeted:aggregate:volumetric:session:token_reuse:country:` *`low|medium|high`* and `awswaf:managed:aws:bot-control:TGT_TokenReuseCountry` *`Low|Medium|High`*

Rule name	Description
`TGT_TokenReuseAsnLow` , `TGT_TokenReuseAsnMedium` , `TGT_TokenReuseAsnHigh`	Inspects requests that are not from verified bots for the use of a single token across multiple networking autonomous system numbers (ASNs) in the last 5 minutes. Each level has a limit on the number of distinct ASNs: • Low: more than 1 • Medium: more than 2 • High: more than 3 **Note** The thresholds that these rules apply can vary slightly due to latency. A few requests might make it through beyond the limit before the rule action is applied. Rule actions: • Low: Count • Medium: CAPTCHA • High: Block Labels: `awswaf:managed:aws:bot-control:targeted:aggregate:volumetric:session:token_reuse:asn:` *`low|medium|high`* and `awswaf:managed:aws:bot-control:TGT_TokenReuseAsn` *`Low|Medium|High`*

AWS WAF Distributed Denial of Service (DDoS) prevention rule group

This section describes the AWS WAF managed rule group for the protection against Distributed Denial of Service (DDoS) attacks.

VendorName: `AWS`, Name: `AWSManagedRulesAntiDDoSRuleSet`, WCU: 50

> **Note**
>
> This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup. The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
> If you need more information than you find here, contact the AWS Support Center.

The Anti-DDoS managed rule group provides rules that detect and manage requests that are participating or likely to be participating in DDoS attacks. Additionally, the rule group labels all requests that it evaluates during a probable event.

Considerations for using this rule group

This rule group provides soft and hard mitigations for web requests coming to resources that are under DDoS attack. To detect different threat levels, you can tune the sensitivity of both mitigation types to high, medium, or low suspicion levels.

- **Soft mitigation** – The rule group can send silent browser challenges in response to requests that can handle the challenge interstitial. For information about the requirements for running the challenge, see CAPTCHA and Challenge action behavior.
- **Hard mitigation** – The rule group can block requests altogether.

For more information about how the rule group works and how to configure it, see Advanced Anti-DDoS protection using the AWS WAF Anti-DDoS managed rule group.

> **Note**
>
> You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

This rule group is part of the intelligent threat mitigation protections in AWS WAF. For information, see Intelligent threat mitigation in AWS WAF.

To minimize costs and optimize traffic management, use this rule group in accordance with best practice guidelines. See, Best practices for intelligent threat mitigation in AWS WAF.

Labels added by this rule group

This managed rule group adds labels to the web requests that it evaluates, which are available to rules that run after this rule group in your protection pack (web ACL). AWS WAF also records the labels to Amazon CloudWatch metrics. For general information about labels and label metrics, see Web request labeling and Label metrics and dimensions.

Token labels

This rule group uses AWS WAF token management to inspect and label web requests according to the status of their AWS WAF tokens. AWS WAF uses tokens for client session tracking and verification.

For information about tokens and token management, see Token use in AWS WAF intelligent threat mitigation.

For information about the label components described here, see Label syntax and naming requirements in AWS WAF.

Client session label

The label `awswaf:managed:token:id:`*`identifier`* contains a unique identifier that AWS WAF token management uses to identify the client session. The identifier can change if the client acquires a new token, for example after discarding the token it was using.

> **Note**
>
> AWS WAF doesn't report Amazon CloudWatch metrics for this label.

Browser fingerprint label

The label `awswaf:managed:token:fingerprint:`*`fingerprint-identifier`* contains a robust browser fingerprint identifier that AWS WAF token management computes from various client browser signals. This identifier stays the same across multiple token acquisition attempts. The fingerprint identifier is not unique to a single client.

> **Note**
>
> AWS WAF doesn't report Amazon CloudWatch metrics for this label.

Token status labels: Label namespace prefixes

Token status labels report on the status of the token and of the challenge and CAPTCHA information that it contains.

Each token status label begins with one of the following namespace prefixes:

- `awswaf:managed:token:` – Used to report the general status of the token and to report on the status of the token's challenge information.
- `awswaf:managed:captcha:` – Used to report on the status of the token's CAPTCHA information.

Token status labels: Label names

Following the prefix, the rest of the label provides detailed token status information:

- `accepted` – The request token is present and contains the following:
  - A valid challenge or CAPTCHA solution.
  - An unexpired challenge or CAPTCHA timestamp.
  - A domain specification that's valid for the protection pack (web ACL).

  Example: The label `awswaf:managed:token:accepted` indicates that the web requests's token has a valid challenge solution, an unexpired challenge timestamp, and a valid domain.
- `rejected` – The request token is present but doesn't meet the acceptance criteria.

  Along with the rejected label, token management adds a custom label namespace and name to indicate the reason.

- `rejected:not_solved` – The token is missing the challenge or CAPTCHA solution.
- `rejected:expired` – The token's challenge or CAPTCHA timestamp has expired, according to your protection pack (web ACL)'s configured token immunity times.
- `rejected:domain_mismatch` – The token's domain isn't a match for your protection pack (web ACL)'s token domain configuration.
- `rejected:invalid` – AWS WAF couldn't read the indicated token.

  Example: The labels `awswaf:managed:captcha:rejected` and `awswaf:managed:captcha:rejected:expired` together indicate that the request didn't have a valid CAPTCHA solve because the CAPTCHA timestamp in the token has exceeded the CAPTCHA token immunity time that's configured in the protection pack (web ACL).

- `absent` – The request doesn't have the token or the token manager couldn't read it.

  Example: The label `awswaf:managed:captcha:absent` indicates that the request doesn't have the token.

Anti-DDoS labels

The Anti-DDoS managed rule group generates labels with the namespace prefix `awswaf:managed:aws:anti-ddos:` followed by any custom namespace and the label name. Each label reflects some aspect of the Anti-DDoS findings.

The rule group might add more than one of the following labels to a request, in addition to the labels that are added by individual rules.

- `awswaf:managed:aws:anti-ddos:event-detected` – Indicates that the request is going to a protected resource for which the managed rule group detects a DDoS event. The managed rule group detects events when the traffic to the resource has a significant deviation from the resource's traffic baseline.

  The rule group adds this label to every request that goes to the resource while it's in this state, so legitimate traffic and attack traffic get this label.

- `awswaf:managed:aws:anti-ddos:ddos-request` – Indicates that the request is coming from a source suspected of participating in an event.

  In addition to the general label, the rule group adds the following labels that indicate the level of confidence.

`awswaf:managed:aws:anti-ddos:low-suspicion-ddos-request` – Indicates a probable DDoS attack request.

`awswaf:managed:aws:anti-ddos:medium-suspicion-ddos-request` – Indicates a very likely DDoS attack request.

`awswaf:managed:aws:anti-ddos:high-suspicion-ddos-request` – Indicates a highly likely DDoS attack request.

- `awswaf:managed:aws:anti-ddos:challengeable-request` – Indicates that the request URI is capable of handling the Challenge action. The managed rule group applies this to any request whose URI isn't exempted. URIs are exempted if they match the rule group's exempt URI regular expressions.

  The rule group matches each exempt URI regular expression against the request's host followed by its normalized URI path (the scheme and query string aren't included), so a pattern anchored to the start of the path such as `^/api/` doesn't match. The path is normalized first, so `/api/` is evaluated as `/api`. Only `GET` requests are challengeable; the rule group never applies this label to requests that use other methods.

  For information about the requirements for requests that can take a silent browser challenge, see CAPTCHA and Challenge action behavior.

You can retrieve all labels for a rule group through the API by calling `DescribeManagedRuleGroup`. The labels are listed in the `AvailableLabels` property in the response.

The Anti-DDoS managed rule group applies labels to requests, but doesn't always act on them. The request management depends on the confidence with which the rule group determines participation in an attack. If you want, you can manage requests that the rule group labels by adding a label matching rule that runs after the rule group. For more information about this and examples, see AWS WAF Distributed Denial of Service (DDoS) prevention.

Anti-DDoS rules listing

This section lists the Anti-DDoS rules.

Note

This documentation covers the most recent static version release of this managed rule group. We report version changes in the changelog log at AWS Managed Rules changelog. For information about other versions, use the API command DescribeManagedRuleGroup. The information that we publish for the rules in the AWS Managed Rules rule groups is intended to provide you with what you need to use the rules without giving bad actors what they need to circumvent the rules.
If you need more information than you find here, contact the AWS Support Center.

Rule name	Description
`ChallengeAllDuringEvent`	Matches requests that have the label `awswaf:managed:aws:anti-ddos:challengeable-request` for any protected resource that is currently under attack. Rule action: Challenge You can only override this rule action to Allow or Count. The use of Allow is not recommended. For any rule action setting, the rule only matches requests that have the `challengeable-request` label. The configuration of this rule affects the evaluation of the next rule, `ChallengeDDoSRequests`. AWS WAF only evaluates that rule when the action for this rule has override set to Count, in the web ACL's configuration of the managed rule group. If your workload is vulnerable to unexpected request volume changes, we recommend challenging all challengable requests, by keeping the default action setting of

Rule name	Description
	Challenge. For less sensitive applicati ons, you can set the action for this rule to Count and then tune the sensitivity of your Challenge responses with the rule `ChallengeDDoSRequests` . Labels: `awswaf:managed:aws:anti-ddos:ChallengeAllDuringEvent`
`ChallengeDDoSRequests`	Matches requests for a protected resource that meet or exceed the rule group's configured challenge sensitivity setting, during times that the resource is under attack. Rule action: Challenge You can only override this rule action to Allow or Count. The use of Allow is not recommend ed. In any case, the rule only matches requests that have the `challengeable-request` label. AWS WAF only evaluates this rule if you override the action to Count in the prior rule, `ChallengeAllDuringEvent` . Labels: `awswaf:managed:aws:anti-ddos:ChallengeDDoSRequests`

Rule name	Description
`DDoSRequests`	Matches requests for a protected resource that meet or exceed the rule group's configured block sensitivity setting, during times that the resource is under attack. Rule action: Block Labels: `awswaf:managed:aws:anti-ddos:DDoSRequests`

Deployments for versioned AWS Managed Rules rule groups

This section introduces how AWS deploys updates to AWS Managed Rules rule groups.

AWS deploys changes to its versioned AWS Managed Rules rule groups in three standard deployments: release candidate, static version, and default version. Additionally, AWS might sometimes need to release an exception deployment or roll back a default version deployment.

> **Note**
>
> This section applies only to AWS Managed Rules rule groups that are versioned. The only rule groups that aren't versioned are the IP reputation rule groups.

Topics

- Notifications for AWS Managed Rules rule groups deployments
- Overview of the standard deployments for AWS Managed Rules
- Typical version states for AWS Managed Rules
- Release candidate deployments for AWS Managed Rules
- Static version deployments for AWS Managed Rules
- Default version deployments for AWS Managed Rules
- Exception deployments for AWS Managed Rules
- Default deployment rollbacks for AWS Managed Rules

Notifications for AWS Managed Rules rule groups deployments

This section explains how Amazon SNS notifications work with AWS Managed Rules rule groups.

The versioned AWS Managed Rules rule groups all provide SNS update notifications for deployments and they all use the same SNS topic Amazon Resource Name (ARN). The only rule groups that aren't versioned are the IP reputation rule groups.

For deployments that affect your protections, such as changes to the default version, AWS provides SNS notifications to inform you of planned deployments and to let you know when a deployment is starting. For deployments that don't affect your protections, such as release candidate and static version deployments, AWS might notify you after the deployment has started or even after it's completed. At the completion of the deployment of a new static version, AWS updates this guide, in the changelog at AWS Managed Rules changelog and in the document history page at Document history.

To receive all updates that AWS provides for the AWS Managed Rules rule groups, subscribe to the RSS feed from any HTML page of this guide, and subscribe to the SNS topic for the AWS Managed Rules rule groups. For information about subscribing to the SNS notifications, see Getting notified of new versions and updates to a managed rule group.

Contents of the SNS notifications

The fields in the Amazon SNS notifications always include the Subject, Message, and MessageAttributes. Additional fields depend on the type of message and which managed rule group the notification is for. The following shows an example notification listing for `AWSManagedRulesCommonRuleSet`.

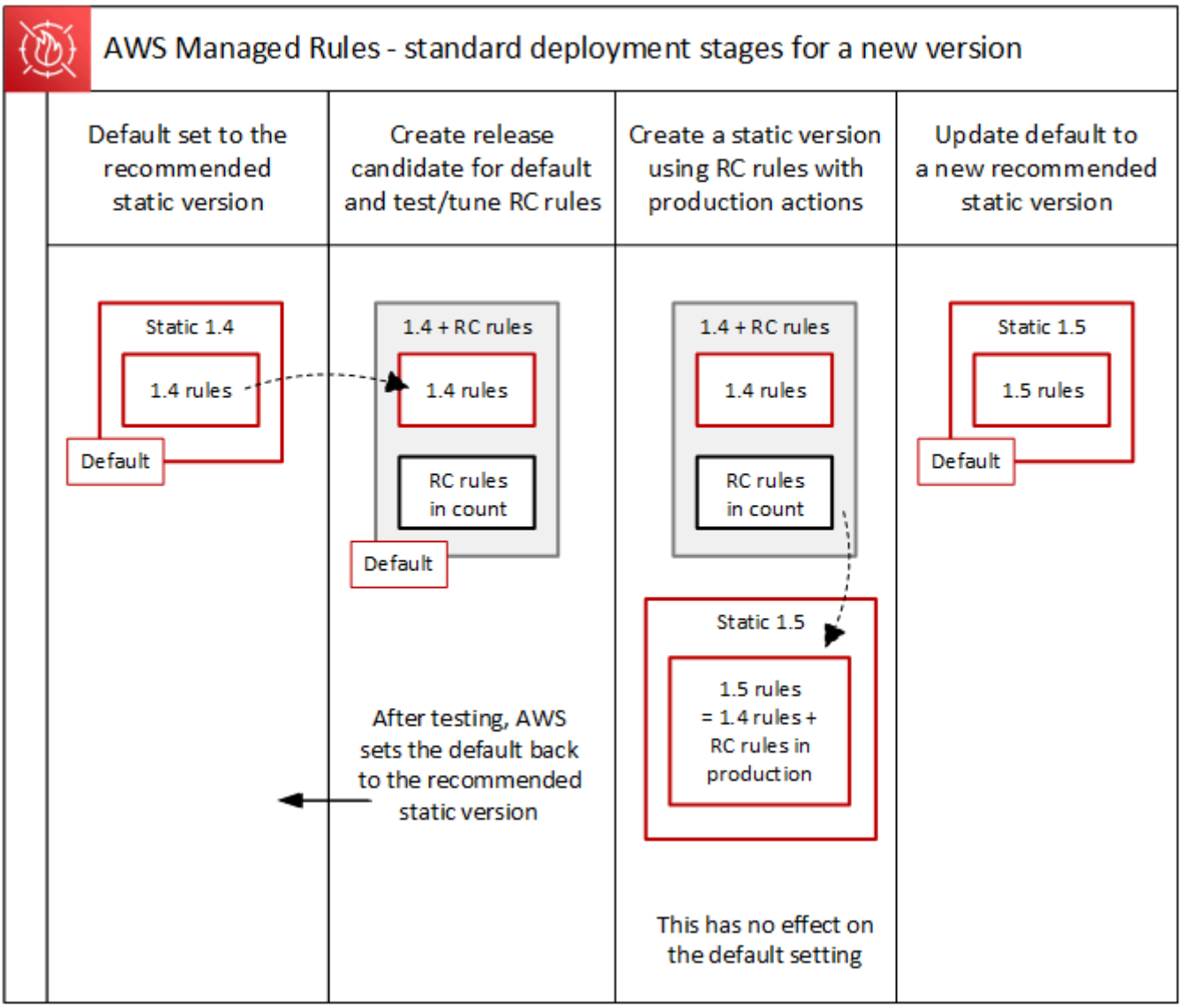
```
{
    "Type": "Notification",
    "MessageId": "4286b830-a463-5e61-bd15-e1ae72303868",
    "TopicArn": "arn:aws:sns:us-west-2:123456789012:MyTopic",
    "Subject": "New version available for rule group AWSManagedRulesCommonRuleSet",
    "Message": "Welcome to AWSManagedRulesCommonRuleSet version 1.5! We've updated
 the regex specification in this version to improve protection coverage, adding
 protections against insecure deserialization. For details about this change, see
 http://updatedPublicDocs.html. Look for more exciting updates in the future! ",
    "Timestamp": "2021-08-24T11:12:19.810Z",
    "SignatureVersion": "1",
    "Signature": "EXAMPLEHXgJm...",
    "SigningCertURL": "https://sns.us-west-2.amazonaws.com/SimpleNotificationService-
f3ecfb7224c7233fe7bb5f59f96de52f.pem",
```

```
    "SubscribeURL": "https://sns.us-west-2.amazonaws.com/?
Action=ConfirmSubscription&TopicArn=arn:aws:sns:us-
west-2:123456789012:MyTopic&Token=2336412f37...",
    "MessageAttributes": {
        "major_version": {
            "Type": "String",
            "Value": "v1"
        },
        "managed_rule_group": {
            "Type": "String",
            "Value": "AWSManagedRulesCommonRuleSet"
        }
    }
}
```

Overview of the standard deployments for AWS Managed Rules

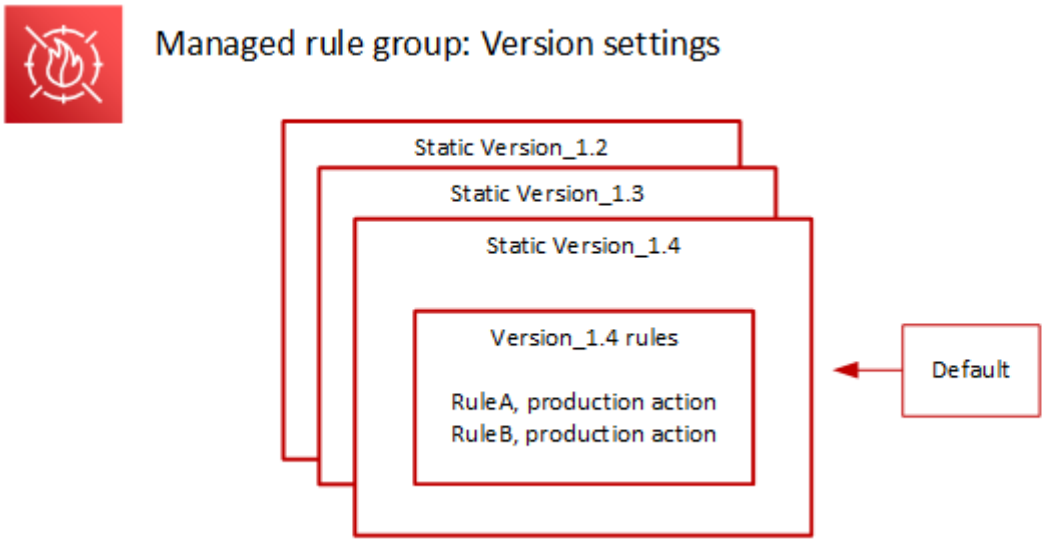
AWS rolls out new AWS Managed Rules functionality using three standard deployment stages: release candidate, static version, and default version.

The following diagram depicts these standard deployments. Each is described in more detail in the sections that follow.



Typical version states for AWS Managed Rules

Normally, a versioned managed rule group has a number of unexpired static versions, and the default version points to the static version that AWS recommends. The following figure shows an example of the typical set of static versions and default version setting.



The production action for most rules in a static version is Block, but it might be set to something different. For detailed information about rule action settings, see the rule listings for each rule group at AWS Managed Rules rule groups list.

Release candidate deployments for AWS Managed Rules

This section explains how a temporary release candidate deployment works.

When AWS has a candidate set of rule changes for a managed rule group, it tests them in a temporary release candidate deployment. AWS evaluates the candidate rules in count mode against production traffic, and performs final tuning activities, including mitigating false positives. AWS tests release candidate rules in this way for all customers who use the default version of the rule group. Release candidate deployments don't apply to customers who use a static version of the rule group.

If you use the default version, a release candidate deployment won't alter how your web traffic is managed by the rule group. You might notice the following while the candidate rules are being tested:

- Default version name change from `Default (using Version_X.Y)` to `Default (using Version_X.Y_PLUS_RC_COUNT)`.
- Additional count metrics in Amazon CloudWatch with `RC_COUNT` in their names. These are generated by the release candidate rules.

AWS tests a release candidate for about a week, then removes it and resets the default version to the current recommended static version.

AWS performs the following steps for a release candidate deployment:

1. **Create the release candidate** – AWS adds a release candidate based on the current recommended static version, which is the version that the default is pointing to.

   The name of the release candidate is the static version name appended with `_PLUS_RC_COUNT`. For example, if the current recommended static version is `Version_2.1`, then the release candidate would be named `Version_2.1_PLUS_RC_COUNT`.

   The release candidate contains the following rules:

   - Rules copied exactly from the current recommended static version, with no changes to rule configurations.
   - Candidate new rules with rule action set to Count and with names that end with `_RC_COUNT`.

     Most candidate rules provide proposed improvements to rules that exist already in the rule group. The name for each of these rules is the existing rule's name appended with `_RC_COUNT`.
2. **Set the default version to the release candidate and test** – AWS sets the default version to point to the new release candidate, to perform testing against your production traffic. Testing usually takes about a week.

   You'll see the default version's name change from the one that indicates only the static version, such as `Default (using Version_1.4)`, to one that indicates the static version plus the release candidate rules, such as `Default (using Version_1.4_PLUS_RC_COUNT)`. This naming scheme lets you identify which static version you're using to manage your web traffic.

   The following diagram shows the state of the example rule group versions at this point.
   At the top of the figure are three stacked static versions, with Version_1.4 on the top. Separate from the static versions stack is the version Version_1.4_PLUS_RC_COUNT. This version contains the rules from Version_1.4 and it also contains two release candidate rules, RuleB_RC_COUNT and RuleZ_RC_COUNT, both with count action. The default version indicator points to Version_1.4_PLUS_RC_COUNT.

   The release candidate rules are always configured with Count action, so they don't alter how the rule group manages web traffic.

The release candidate rules generate Amazon CloudWatch count metrics that AWS uses to verify behavior and to identify false positives. AWS makes adjustments as needed, to tune the behavior of the release candidate count rules.

The release candidate version isn't a static version, and it's not available for you to choose from the list of static rule group versions. You can only see the name of the release candidate version in the default version specification.

3. **Return the default version to the recommended static version** – After testing the release candidate rules, AWS sets the default version back to the current recommended static version. The default version name setting drops the `_PLUS_RC_COUNT` ending, and the rule group stops generating CloudWatch count metrics for the release candidate rules. This is a silent change, and is not the same as a deployment of a default version rollback.

   The following diagram shows the state of the example rule group versions after the testing of the release candidate is complete.
   This is the typical version states figure again. Three static versions Version_1.2, Version_1.3, and Version_1.4 are stacked with Version_1.4 on the top. Version_1.4 has two rules, RuleA and RuleB, both with production action. A default version indicator points to Version_1.4.

Timing and notifications

AWS deploys release candidate versions on an as-needed basis, to test improvements to a rule group.

- **SNS** – AWS sends an SNS notification at the start of the deployment. The notification indicates the estimated time that the release candidate will be tested. When testing is complete, AWS silently returns the default to the static version setting, without a second notification.
- **Change log** – AWS doesn't update the change log or other parts of this guide for this type of deployment.

Static version deployments for AWS Managed Rules

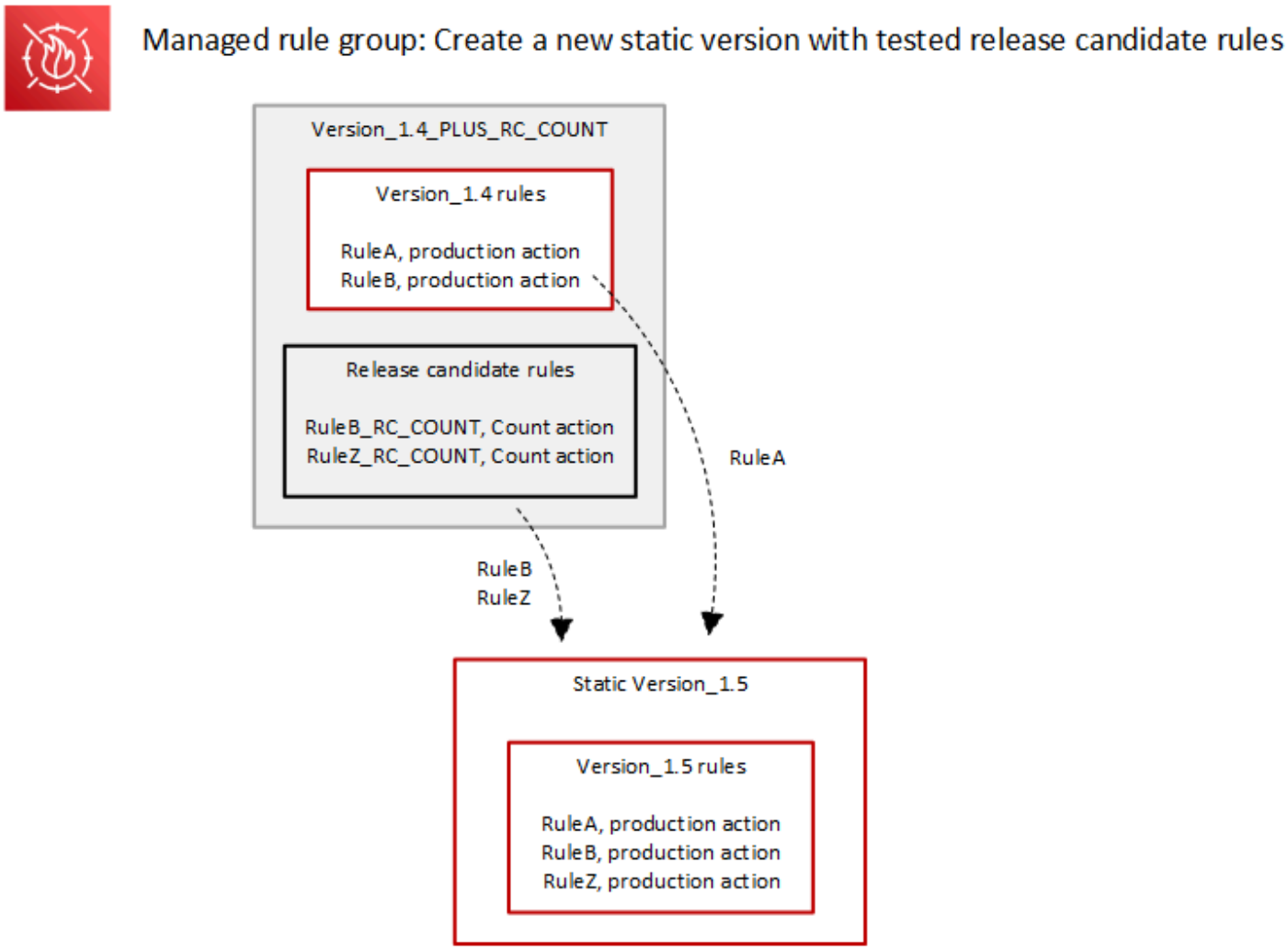
When AWS determines that a release candidate provides valuable changes to the rule group, AWS deploys a new static version for the rule group based on the release candidate. This deployment doesn't change the default version of the rule group.

The new static version contains the following rules from the release candidate:

- Rules from the prior static version that don't have a replacement candidate among the release candidate rules.
- Release candidate rules, with the following changes:
  - AWS changes the rule name by removing the release candidate suffix `_RC_COUNT`.
  - AWS changes the rule actions from Count to their production rule actions.

  For release candidate rules that are replacements of prior existing rules, this replaces the functionality of the prior rules in the new static version.

The following diagram depicts the creation of the new static version from the release candidate.



After deployment, the new static version is available for you to test and to use in your protections if you want to. You can review new and updated rule actions and descriptions in the rule group's rule listings at AWS Managed Rules rule groups list.

A static version is immutable after deployment, and only changes when AWS expires it. For information about version life cycles, see Using versioned managed rule groups in AWS WAF.

Timing and notifications

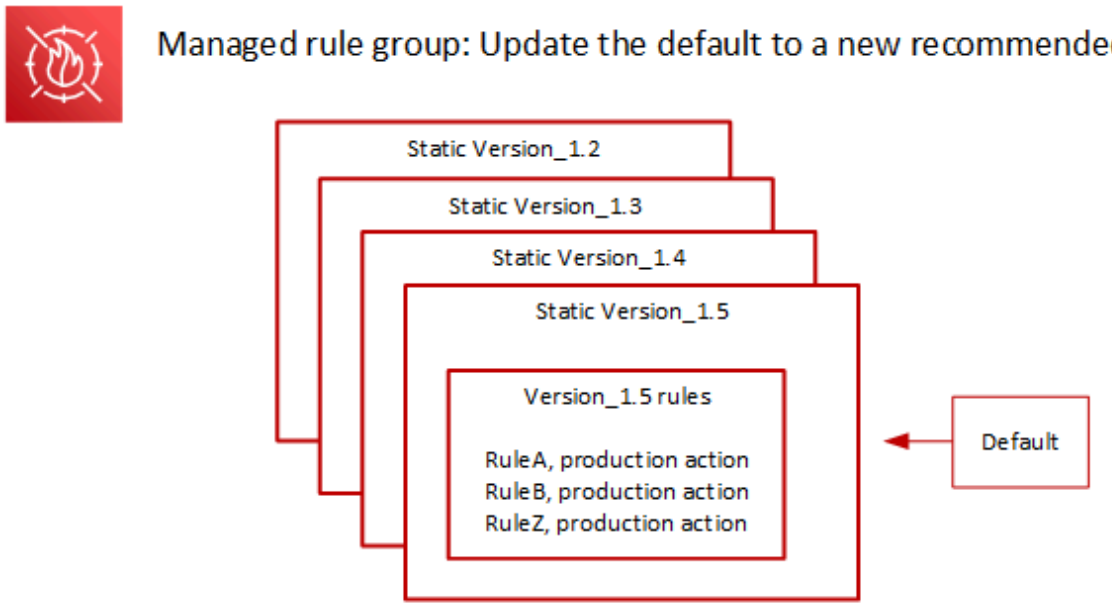
AWS deploys a new static version as needed, in order to deploy improvements to rule group functionality. The deployment of a static version doesn't impact the default version setting.

- **SNS** – AWS sends an SNS notification when the deployment completes.
- **Change log** – After the deployment is complete everywhere that AWS WAF is available, AWS updates the rule group definition in this guide as needed, and then announces the release in the AWS Managed Rules rule group change log and in the documentation history page.

Default version deployments for AWS Managed Rules

When AWS determines that a new static version provides improved protections for the rule group compared to the current default, AWS updates the default version to the new static version. AWS might release multiple static versions before promoting one to the rule group's default version.

The following diagram shows the state of the example rule group versions after AWS moves the default version setting to the new static version.



Before deploying this change to the default version, AWS provides notifications so that you can test and prepare for the upcoming changes. If you use the default version, you can take no action

and remain on it through the update. If instead you want to delay switching to the new version, before the planned start of the default version deployment, you can explicitly configure your rule group to use the static version that the default is set to.

Timing and notifications

AWS updates the default version when it recommends a different static version for the rule group than the one that's currently in use.

- **SNS** – AWS sends an SNS notification at least one week prior to the targeted deployment day and then another on the deployment day, at the start of the deployment. Each notification includes the rule group name, the static version that the default version is being updated to, the deployment date, and the scheduled timing of the deployment for each AWS Region where the update is being performed.
- **Change log** – AWS doesn't update the change log or other parts of this guide for this type of deployment.

Exception deployments for AWS Managed Rules

AWS might bypass the standard deployment stages in order to quickly deploy updates that address critical security risks. An exception deployment might involve any of the standard deployment types, and it might be rolled out quickly across the AWS Regions.

AWS provides as much advance notification as possible for exception deployments.

Timing and notifications

AWS performs exception deployments only when required.

- **SNS** – AWS sends an SNS notification as far ahead of the targeted deployment day as possible and then another one at the start of the deployment. Each notification includes the rule group name, the change that's being made, and the deployment date.
- **Change log** – If the deployment is for a static version, after the deployment is complete everywhere that AWS WAF is available, AWS updates the rule group definition in this guide as needed, and then announces the release in the AWS Managed Rules rule group change log and in the documentation history page.

Default deployment rollbacks for AWS Managed Rules

Under certain conditions, AWS might roll back the default version to its prior setting. A rollback usually takes less than ten minutes for all AWS Regions.

AWS performs a rollback only to mitigate a significant issue in a static version, such as an unacceptably high level of false positives.

After the rollback of the default version setting, AWS expedites both the expiration of the static version that has the issue and the release of a new static version to address the issue.

Timing and notifications

AWS performs default version rollbacks only when required.

- **SNS** – AWS sends a single SNS notification at the time of the rollback. The notification includes the rule group name, the version that the default version is being set to, and the deployment date. This deployment type is very quick, so the notification doesn't provide timing information for Regions.
- **Change log** – AWS doesn't update the change log or other parts of this guide for this type of deployment.

AWS Managed Rules changelog

This section lists changes to the AWS Managed Rules for AWS WAF since their release in November, 2019.

> **Note**
>
> This changelog reports changes to the rules and rule groups in AWS Managed Rules for AWS WAF.
> For the IP reputation rule groups, this changelog reports changes to the rules and rule group, and it reports significant changes to the sources of the IP address lists that the rules use. It does not report changes to the IP address lists themselves, due to the dynamic nature of those lists. If you have questions about the IP address lists, contact your account manager or open a case at AWS Support Center.

Rule group and rules	Description	Date
SQL database managed rule group This release updates the following rules: • SQLi_BODY • SQLiExtendedPatterns_QUERYARGUMENTS • SQLiExtendedPatterns_BODY • SQLiExtendedPatterns_HEADER • SQLiExtendedPatterns_URIPATH	Released static version 2.5 of this rule group. Updated the SQLi_BODY rule and all SQLiExtendedPatterns rules to reduce false positives and improve detection.	2026-09-16
Core rule set (CRS) managed rule group This release updates the following rules: • CrossSiteScripting_BODY • CrossSiteScripting_COOKIE • CrossSiteScripting_QUERYARGUMENTS •	Released static version 1.23 of this rule group. Improved detection signatures for the cross site scripting rules.	2026-08-28

Rule group and rules	Description	Date
CrossSiteScripting_URIPATH		
SQL database managed rule group This release updates the following rules: • SQLi_COOKIE • SQLi_URIPATH • SQLi_QUERYARGUMENTS • SQLi_BODY • SQLiExtendedPatterns_QUERYARGUMENTS • SQLiExtendedPatterns_BODY • SQLiExtendedPatterns_HEADER • SQLiExtendedPatterns_URIPATH	Released static version 2.4 of this rule group. This release updates all rules to improve SQL injection detection logic.	2026-08-12

Rule group and rules	Description	Date
AWS WAF Bot Control rule group	Released static version 6.1 of this rule group. Added new bot detection signatures across multiple categories, including Advertising, AI, Content Fetcher, Scraping Framework, Search Engine, Security, SEO, and Social Media.	2026-07-24
Core rule set (CRS) managed rule group • `GenericLFI_QUERYARGUMENTS` • `GenericLFI_URIPATH` • `GenericLFI_BODY`	Released static version 1.22 of this rule group. Improved detection signatures for the generic LFI rules.	2026-07-20
Known bad inputs managed rule group • `ReactJSRCE_BODY`	Released static version 1.26 of this rule group. Added the `REMOVE_NULLS` text transformation to the listed rule.	2026-07-15

Rule group and rules	Description	Date
Windows operating system managed rule group • WindowsShellCommands_HEADER • WindowsShellCommands_URIPATH • WindowsShellCommands_BODY • WindowsShellCommands_QUERYARGUMENTS • WindowsShellCommands_QUERYSTRING	Released static version 2.5 of this rule group. Removed WindowsShellCommands_QUERYARGUMENTS and replaced it with WindowsShellCommands_QUERYSTRING . Updated WindowsShellCommands rules to improve detection logic.	2026-06-01

Rule group and rules	Description	Date
AWS WAF Bot Control rule group	Released static version 6.0 of this rule group. Added new bot detection signatures across multiple categories, including Advertising, Content Fetcher, and Search Engine. Expanded Web Bot Auth support to commercial partition regional AWS WAF integrations. Changed the behavior of Web Bot Auth verified bots to be treated as standard verified bots for all categories, not only the AI category. This static version does not change the default version behavior.	2026-05-22

Rule group and rules	Description	Date
SQL database managed rule group • SQLi_COOKIE • SQLi_URIPATH • SQLi_QUERYARGUMENTS • SQLi_BODY • SQLiExtendedPatterns_QUERYARGUMENTS • SQLiExtendedPatterns_BODY • SQLiExtendedPatterns_HEADER • SQLiExtendedPatterns_URIPATH	Released static version 2.3 of this rule group. Added new rules SQLiExtendedPatterns_HEADER and SQLiExtendedPatterns_URIPATH . Added REPLACE_COMMENTS text transformations to all rules. Added JS_DECODE text transformations to SQLi_COOKIE , SQLi_URIPATH , and to all the new rules. Updated all the rules to improve detection logic. Version_2.3 is the successor to Version_1.3, not Version_2.0. These are separate lineages with different detection trade-offs. Customers relying on Version_2.0 detection behavior should pin that version explicitly.	2026-04-29

Rule group and rules	Description	Date
Windows operating system managed rule group • WindowsShellCommands_HEADER • WindowsShellCommands_COOKIE • WindowsShellCommands_URIPATH • WindowsShellCommands_BODY • WindowsShellCommands_QUERYARGUMENTS	Released static version 2.4 of this rule group. Removed WindowsShellCommands_COOKIE and replaced it with WindowsShellCommands_HEADER which matches on all headers. Added new rule WindowsShellCommands_URIPATH. Added JSON body inspection to the WindowsShellCommands_BODY. Added double URL_DECODE_UNI and JS_DECODE text transformations to the WindowsShellCommands rules. Updated WindowsShellCommands rules to reduce false positives.	2026-04-28
Core rule set (CRS) managed rule group • RestrictedExtensions_QUERYARGUMENTS • RestrictedExtensions_URIPATH	Released static version 1.21 of this rule group. Improved detection signatures for these rules.	2026-04-06

Rule group and rules	Description	Date
PHP application managed rule group • PHPHighRiskMethodsVariables_BODY • PHPHighRiskMethodsVariables_QUERYSTRING • PHPHighRiskMethodsVariables_HEADER • PHPHighRiskMethodsVariables_URIPATH	Released static version 2.2 of this rule group. Improved detections and added PHPHighRiskMethodsVariables_URIPATH rule.	2026-03-24

Rule group and rules	Description	Date
AWS WAF Bot Control rule group New rules: • `CategoryPagePreview` • `CategoryWebhooks`	Released static version 5.0 of this rule group. Added 400+ new bots across multiple categories, including two new bot categories with their respective rules: Page Preview and Webhooks. **Key Improvements** Improved accuracy of bot detection signals and generic bot pattern matching, resulting in more precise traffic classification. This update changes how the managed rule group prioritizes bot detection . Specific unverified bot patterns are now evaluated before generic patterns and detection signals. This means that requests are more likely to be classified based on their most specific characteristics rather than generic indicators. **What this means for your tr affic:** Generic bot pattern will now match less frequently. These patterns only apply when no	2026-02-25

Rule group and rules	Description	Date
	more specific bot rule has already identified the traffic. This reduces over-classification and ensures that requests are labeled with the most accurate bot identification available. Detection signals such as indicators that a request originates from a cloud service provider, known bot data center, or uses a non-browser user agent, are now applied after bot identification rules. This ensures that specific bot classifications take precedence over generic traffic signals. **Impact:** You may see fewer labels for generic bot patterns in your traffic logs, as requests are now more accurately classified by specific bot rules. This provides clearer insight into the actual nature of your automated traffic and reduces noise from overly broad pattern matching. Unverified bot classifications will be more prominent and accurate, helping you better understand and manage	

Rule group and rules	Description	Date
	automated requests to your applications. **Note:** This version includes the `awswaf:managed:aws:bot-control:bot:web_bot_auth` labels and rules updates from Version_4.0, but the `Web Bot Auth` functionality is still only available in CloudFront.	
POSIX operating system managed rule group	Released static version 3.2 of this rule group. Improved detection signatures for all the rules.	2026-01-15
Known bad inputs managed rule group • `ReactJSRCE_BODY`	Released static version 1.25 of this rule group. Updated the `ReactJSRCE_BODY` to improve detection.	2025-12-08
POSIX operating system managed rule group	Released static version 3.1 of this rule group. Improved detection signatures for all the rules.	2025-12-08

Rule group and rules	Description	Date
Known bad inputs managed rule group • `ReactJSRCE_BODY`	Released static version 1.24 of this rule group. Updated the `ReactJSRCE_BODY` to improve detection.	2025-12-04

Rule group and rules	Description	Date

AWS WAF Bot Control rule group
 	
New labels:
 	
• awswaf:managed:aws:bot-control:bot:web_bot_auth<status>
  • verified – Successful verification
  • invalid – Failed verification attempt
  • expired –Expired key usage
  • unknown_bot – Unrecognized key usage
• awswaf:managed:aws:bot-control:bot:vendor<vendor_name> – Bot vendor or operator
• awswaf:managed:aws:bot-control:bot:name<rfc_name> – Bot name (RFC token from WBA)
• awswaf:managed:aws:bot-control:bot:a
	Deployed new static version AWSManagedRulesBotControlRuleSet Version_4.0 with Web Bot Authentication (WBA) support for cryptographic bot verification. This version must be explicitly selected and does not automatically update existing deployments using the default version.
	
	New capabilities:
	
	• Web Bot Authentication (WBA) cryptographic verification for AI bots and agents
	• Automatic allowlisting of verified WBA bots in Category AI rule
	• Enhanced bot identification with vendor, name, and account labels
	• Support for Amazon Bedrock Agent Core bot detection
	
	Rule updates:
		2025-11-20

Rule group and rules	Description	Date

ccount<hash> – AWS account identifier. (Amazon Bedrock Agent Core agents only)
 	
Scope: CloudFront
	• Category AI (Action: Block)
	  • Update to not match requests with web_bot_auth:verified label.
	  • Verified WBA bots are automatically allowed
	• TGT_TokenAbsent (Action: Count)
	  • Updated to not match requests with web_bot_auth:verified label
	
	Important
	Version_4.0 is a static version only – it does not change the default version behavior. To use WBA features, explicitly select Version _4.0 when configuring your web ACL.	

Rule group and rules	Description	Date
AWS WAF Bot Control rule group New verified bot labels: Advertising: • `awswaf:managed:aws:bot-control:bot:name:meta-externalads` – Meta advertising bot • `awswaf:managed:aws:bot-control:bot:organization:facebook` – Meta organization AI: • `awswaf:managed:aws:bot-control:bot:name:meta-webindexer` – Meta web indexer • `awswaf:managed:aws:bot-control:bot:name:perplexity-user` – Perplexity user bot • `awswaf:managed:aws:bot-control:bot:name:perplexitybot` – Perplexity bot •	Key improvements: • Expanded verified bot detection across multiple categories (Advertising, AI, Content Fetcher and Social Media) • Added organization-level labels for Meta, Facebook bots • Enhanced AI bot detection including Perplexity, Byt espider, and DuckDuckGo bots • Added TikTok social media bot detection **Important** Bot category rules in Bot Control trigger only on unverified bots, except for CategoryAI which also triggers on verified bots. Version_3.3 is a static version only – it doesn't change	2025-11-17

Rule group and rules	Description	Date
`awswaf:managed:aws:bot-control:bot:name:bytespider` – Bytespider bot • `awswaf:managed:aws:bot-control:bot:name:duckassistbot` – DuckDuckGo assistant bot • `awswaf:managed:aws:bot-control:bot:organization:facebook` – Meta organization Content Fetcher: • `awswaf:managed:aws:bot-control:bot:name:meta-externalfetcher` – Meta content fetcher • `awswaf:managed:aws:bot-control:bot:organization:facebook-externalfetcher` – Meta organization Social Media: • `awswaf:managed:aws:bot-control:bot:n`	the default version behavior.	

Rule group and rules	Description	Date
`ame:tiktok` – TikTok bot		
Core rule set (CRS) managed rule group • `EC2MetaDataSSRF_BODY` • `EC2MetaDataSSRF_COOKIE` • `EC2MetaDataSSRF_URIPATH` • `EC2MetaDataSSRF_QUERYARGUMENTS`	Released static version 1.20 of this rule group. Improved detection signatures for the Server Side Request Forgery (SSRF) rules.	2025-10-02
Core rule set (CRS) managed rule group • `CrossSiteScripting_BODY` • `CrossSiteScripting_COOKIE` • `CrossSiteScripting_QUERYARGUMENTS` • `CrossSiteScripting_URIPATH`	Released static version 1.19 of this rule group. Improved detection signatures for the cross site scripting rules.	2025-08-14

Rule group and rules	Description	Date
Core rule set (CRS) managed rule group • CrossSiteScripting_BODY • CrossSiteScripting_COOKIE • CrossSiteScripting_QUERYARGUMENTS • CrossSiteScripting_URIPATH	Released static version 1.18 of this rule group. Improved detection signatures for the cross site scripting rules.	2025-06-18

Rule group and rules	Description	Date

AWS WAF Bot Control rule group
 	
New labels:
• Search engine bots:
  • `awswaf:managed:aws:bot-control:bot:name:evensi`
  • `awswaf:managed:aws:bot-control:bot:name:yisouspider`
• AI bots:
  • `awswaf:managed:aws:bot-control:bot:name:searchbot`
  • `awswaf:managed:aws:bot-control:bot:name:nova_act`
• Organization labels for AI bots:
  • `awswaf:managed:aws:bot-control:bot:o`
	Released static version 3.2 of this rule group.
	
	Added the listed new labels.
		2025-05-29

Rule group and rules	Description	Date
rganizati on:openai • awswaf:ma naged:aws:bot- control:bot:o rganizati on:marfeel • awswaf:ma naged:aws:bot- control:bot:o rganizati on:facebook • awswaf:ma naged:aws:bot- control:bot:o rganizati on:metaex ternalagent • awswaf:ma naged:aws:bot- control:bot:o rganizati on:amazon • Content fetcher bots: • awswaf:ma naged:aws:bot- control:bot:n ame:googl		

Rule group and rules	Description	Date
`e_cloud_vertex_bot` • Cloud service provider signals: • `awswaf:managed:aws:bot-control:signal:cloud_service_provider:alibaba`		
Core rule set (CRS) managed rule group • `CrossSiteScripting_BODY` • `CrossSiteScripting_COOKIE` • `CrossSiteScripting_QUERYARGUMENTS` • `CrossSiteScripting_URIPATH`	Released static version 1.17 of this rule group. Improved detection signatures for the cross site scripting rules.	2025-03-03

Rule group and rules	Description	Date
SQL database managed rule group • SQLi_COOKIE • SQLi_URIPATH • SQLi_QUERYARGUMENTS • SQLi_BODY	Released static version 1.3 of this rule group. Added double URL_DECODE_UNI text transformation to the listed rules.	2025-01-24
Linux operating system managed rule group • LFI_HEADER • LFI_URIPATH • LFI_QUERYSTRING	Released static version 2.6 of this rule group. Added signatures to improve detection.	2025-01-24
AWS WAF Bot Control rule group New bot name label in the Bot Control labels: awswaf:managed:aws:bot-control:bot::name:nytimes	Released static version 3.1 of this rule group. Added the New York Times label to the list of bot name labels.	2024-11-07

Rule group and rules	Description	Date
Core rule set (CRS) managed rule group • `CrossSiteScripting_BODY` • `CrossSiteScripting_COOKIE` • `CrossSiteScripting_QUERYARGUMENTS` • `CrossSiteScripting_URIPATH`	Released static version 1.16 of this rule group. Improved detection signatures for the cross site scripting rules.	2024-10-16

Rule group and rules	Description	Date
AWS WAF Bot Control rule group New rules: • `TGT_TokenAbsent` • `TGT_VolumetricSessionMaximum` • `TGT_SignalBrowserAutomationExtension` • `TGT_ML_CoordinatedActivityLow` , `TGT_ML_CoordinatedActivityMedium` , and `TGT_ML_CoordinatedActivityHigh` • `TGT_TokenReuseIpLow` , `TGT_TokenReuseIpMedium` , and `TGT_TokenReuseIpHigh` • `TGT_TokenReuseAsnLow` , `TGT_TokenReuseAsnMedium` , and `TGT_TokenReuseAsnHigh` • `TGT_TokenReuseCountryLow` , `TGT_TokenReuseCountryMedium` , and `TGT_TokenReuseCountryHigh` Deleted rules:	Released static versions 2.0 and 3.0 of this rule group. Version 2.0 is the same as version 3.0, but with rule actions for all new rules set to Count. This guide documents the latest version of each rule group. Added the listed new rules. Updated labeling so that all rules apply a label with the pattern `awswaf:managed:aws:bot-control:` *`<RuleName>`* . Added cloud service provider labels to the Bot Control signal labels. Added new bot name labels that are inspected for by bot category rules.	2024-09-13

Rule group and rules	Description	Date

• `TGT_TokenReuseIp` . Replaced with the corresponding new low, medium, and high rules.
 	
New labels:
• HTTP Library bots:
  • `awswaf:managed:aws:bot-control:bot:name:fasthttp`
• AI bots:
  • `awswaf:managed:aws:bot-control:bot:name:bedrockbot`
  • `awswaf:managed:aws:bot-control:bot:name:claudebot`
  • `awswaf:managed:aws:bot-control:bot:name:anthropic`
  • `awswaf:managed:aws:bot-control:bot:name:metaexternalagent`
  • `awswaf:managed:aws:bot-`		

Rule group and rules	Description	Date
control:bot:name:bytespider • awswaf:managed:aws:bot-control:bot:name:omgili • awswaf:managed:aws:bot-control:bot:name:diffbot • awswaf:managed:aws:bot-control:bot:name:perplexitybot • awswaf:managed:aws:bot-control:bot:name:timpibot • awswaf:managed:aws:bot-control:bot:name:cohere • Search engine bots: • awswaf:managed:aws:bot-control:bot:name:naver • Advertising bots: • awswaf:managed:aws:bot-control:bot:name:naver_ads		

Rule group and rules	Description	Date

• Social media bots:
  • `awswaf:managed:aws:bot-control:bot:name:snapchat`
• Content fetcher bots:
  • `awswaf:managed:aws:bot-control:bot:name:naver_preview`
  • `awswaf:managed:aws:bot-control:bot:name:censys`
  • `awswaf:managed:aws:bot-control:bot:name:imessage_preview`
  • `awswaf:managed:aws:bot-control:bot:name:imagesift`
• Cloud service provider signals:
  • `awswaf:managed:aws:bot-control:signal:cloud_service_provider:aws`
  • `awswaf:managed:aws:bot-`		

Rule group and rules	Description	Date
control:signal:cloud_service_provider:azure • awswaf:managed:aws:bot-control:signal:cloud_service_provider:gcp • awswaf:managed:aws:bot-control:signal:cloud_service_provider:oracle • awswaf:managed:aws:bot-control:signal:cloud_service_provider:digital_ocean • awswaf:managed:aws:bot-control:signal:cloud_service_provider:akamai • awswaf:managed:aws:bot-control:signal:cloud_s		

Rule group and rules	Description	Date
`ervice_pr ovider:cloudflare` • `awswaf:ma naged:aws:bot-control:signa l:cloud_s ervice_pr ovider:ibm_cloud` Additional labeling in existing rules.		
AWS WAF Fraud Control account takeover prevention (ATP) rule group All rules	Released static version 1.1 of this rule group. Updated labeling so that all rules apply a label with the pattern `awswaf:managed:aws :atp: <RuleName>` .	2024-09-13
AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group All rules	Released static version 1.1 of this rule group. Updated labeling so that all rules apply a label with the pattern `awswaf:managed:aws :acfp: <RuleName>` .	2024-09-13
Linux operating system managed rule group All rules	Released static version 2.5 of this rule group. Added signatures to improve detection.	2024-09-02

Rule group and rules	Description	Date
Core rule set (CRS) managed rule group • GenericLFI_QUERYARGUMENTS • GenericLFI_URIPATH • GenericLFI_BODY	Released static version 1.15 of this rule group. Improved detection signatures for the generic LFI rules.	2024-08-30
Windows operating system managed rule group • WindowsShellCommands_QUERYARGUMENTS • WindowsShellCommands_BODY • WindowsShellCommands_COOKIE	Released static version 2.3 of this rule group. Adjusted detection signatures in the listed rules to reduce false positives.	2024-08-28
WordPress application managed rule group • WordPressExploitableCommands_QUERYSTRING	Released static version 1.3 of this rule group. Added the JS_DECODE text transformation to the listed rule.	2024-07-15
Linux operating system managed rule group • LFI_QUERYSTRING	Released static version 2.4 of this rule group. Added the JS_DECODE text transformation to the listed rule.	2024-07-12

Rule group and rules	Description	Date
Core rule set (CRS) managed rule group • EC2MetaDataSSRF_BODY • EC2MetaDataSSRF_QUERYARGUMENTS • GenericLFI_QUERYARGUMENTS • GenericLFI_BODY • RestrictedExtensions_QUERYARGUMENTS • GenericRFI_QUERYARGUMENTS • GenericRFI_BODY • CrossSiteScripting_QUERYARGUMENTS • CrossSiteScripting_BODY • CrossSiteScripting_COOKIE • CrossSiteScripting_URIPATH	Released static version 1.14 of this rule group. Added the JS_DECODE text transformation to the listed rules.	2024-07-09

Rule group and rules	Description	Date
PHP application managed rule group • PHPHighRiskMethodsVariables_BODY • PHPHighRiskMethodsVariables_QUERYSTRING	Released static version 2.1 of this rule group. Added the JS_DECODE text transformation to the listed rules.	2024-07-03
Windows operating system managed rule group • WindowsShellCommands_QUERYARGUMENTS • WindowsShellCommands_BODY • PowerShellCommands_QUERYARGUMENTS • PowerShellCommands_BODY	Released static version 2.2 of this rule group. Added the JS_DECODE text transformation to the listed rules.	2024-07-03
Linux operating system managed rule group All rules	Released static version 2.3 of this rule group. Added signatures to improve detection.	2024-06-06

Rule group and rules	Description	Date
AWS WAF Bot Control rule group AWS WAF Fraud Control account takeover prevention (ATP) rule group AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group	The bot and fraud rule groups are now versioned. If you're using any of these rule groups, this update doesn't change how they handle your web traffic. This update sets the current rule group version to static version 1.0 and sets the default version to point to it. For more information about versioned managed rules, see the following: • Using versioned managed rule groups in AWS WAF • Deployments for versioned AWS Managed Rules rule groups • Getting notified of new versions and updates to a managed rule group	2024-05-29

Rule group and rules	Description	Date
POSIX operating system managed rule group • UNIXShellCommandsVariables_QUERYARGUMENTS • UNIXShellCommandsVariables_QUERYSTRING • UNIXShellCommandsVariables_HEADER • UNIXShellCommandsVariables_BODY	Released static version 3.0 of this rule group. Removed UNIXShellCommandsVariables_QUERYARGUMENTS and replaced it with UNIXShellCommandsVariables_QUERYSTRING. If you have rules that match on the label for UNIXShellCommandsVariables_QUERYARGUMENTS, when you use this version, switch them to match on the label for UNIXShellCommandsVariables_QUERYSTRING. The new label is awswaf:managed:aws:posix-os:UNIXShellCommandsVariables_QueryString. Added the rule UNIXShellCommandsVariables_HEADER, which matches on all headers. Updated all the rules in the managed rule group with improved detection logic. Corrected the documented capitalization of the label	2024-05-28

Rule group and rules	Description	Date
	for `UNIXShellCommandsVariables_BODY` .	
Core rule set (CRS) managed rule group • `CrossSiteScripting*`	Released static version 1.12 of this rule group. Added signatures to all of the cross site scripting rules to improve detection and reduce false positives.	2024-05-21
SQL database managed rule group • `SQLi_BODY` • `SQLi_QUERYARGUMENTS` • `SQLiExtendedPatterns_QUERYARGUMENTS`	Released static version 1.2 of this rule group. Added the `JS_DECODE` text transformation to the listed rules.	2024-05-14
Known bad inputs managed rule group • `JavaDeserializationRCE_BODY` • `JavaDeserializationRCE_QUERYSTRING` • `Log4JRCE_QUERYSTRING` • `Log4JRCE_BODY` • `Log4JRCE_HEADER`	Released static version 1.22 of this rule group. Added the `JS_DECODE` text transformation to the listed rules.	2024-05-08
POSIX operating system managed rule group	Released static version 2.2 of this rule group. Added the `JS_DECODE` text transformation to both rules.	2024-05-08

Rule group and rules	Description	Date
Windows operating system managed rule group • `PowerShellCommands_BODY`	Released static version 2.1 of this rule group. Added signatures to `PowerShellCommands_BODY` to improve detection.	2024-05-03
Amazon IP reputation list managed rule group • `AWSManagedIPReputationList`	Updated the sources of the IP reputation list, to improve identification of addresses that are actively engaging in malicious activities and to reduce false positives. This update doesn't involve a new version because this rule group isn't versioned.	2024-03-13
Known bad inputs managed rule group	Released static version 1.21 of this rule group. Added signatures to improve detection and reduce false positives.	2023-12-16

Rule group and rules	Description	Date
Known bad inputs managed rule group • `ExploitablePaths_URIPATH`	Released static version 1.20 of this rule group. Updated the `ExploitablePaths_URIPATH` rule to add detection for requests that match the Atlassian Confluence CVE-2023-22518 Improper Authorization vulnerability. This vulnerability affects all versions of Confluence Data Center and Server. For more information, see NIST: National Vulnerability Database: CVE-2023-22518 Detail.	2023-12-14
Core rule set (CRS) managed rule group • `CrossSiteScripting*`	Released static version 1.11 of this rule group. Added signatures to all of the cross site scripting rules to improve detection and reduce false positives.	2023-12-06
AWS WAF Bot Control rule group • New label: `awswaf:managed:aws:bot-control:targeted:aggregate:coordinated_activity:low`	Added the coordinated activity low label to the rule group's targeted protection level labels. This label isn't associated with any rule. This labeling is in addition to the medium and high level rules and labels.	2023-12-05

Rule group and rules	Description	Date
Bot Control labels • Label: `awswaf:managed:aws:bot-control:targeted:signal:browser_automation_extension`	Added a signal label to the rule group that indicates the detection of a browser extension that assists in automation. This label isn't specific to an individual rule.	2023-11-14
Core rule set (CRS) managed rule group • `EC2MetaDataSSRF_QUERYARGUMENTS`	Released static version 1.10 of this rule group. Updated one rule to improve detection and reduce false positives.	2023-11-02
Core rule set (CRS) managed rule group • `EC2MetaDataSSRF_BODY` • `EC2MetaDataSSRF_COOKIE` • `EC2MetaDataSSRF_URIPATH` • `EC2MetaDataSSRF_QUERYARGUMENTS`	Released static version 1.9 of this rule group. Updated rules to improve detection and reduce false positives.	2023-10-30

Rule group and rules	Description	Date
POSIX operating system managed rule group • `UNIXShellCommandsVariables_QUERYARGUMENTS`	Released static version 2.1 of this rule group. Updated the query arguments rule to improve detection.	2023-10-12
Core rule set (CRS) managed rule group • `GenericLFI_QUERYARGUMENTS` • `GenericLFI_URIPATH` • `RestrictedExtensions_URIPATH` • `RestrictedExtensions_QUERYARGUMENTS`	Released static version 1.8 of this rule group. Updated rules to improve detection.	2023-10-11

Rule group and rules	Description	Date
Known bad inputs managed rule group • `ExploitablePaths_URIPATH`	Exception deployment: released static version 1.19 of this rule group. Updated the default version to use version 1.19. Updated the `ExploitablePaths_URIPATH` rule to add detection for requests matching the Atlassian Confluence CVE-2023-22515 Privilege Escalation Vulnerability. This vulnerability affects some versions of Atlassian Confluence. For more information, see NIST: National Vulnerability Database: CVE-2023-22515 Detail and Atlassian Support: FAQ for CVE-2023-22515. For information about this deployment type, see Exception deployments for AWS Managed Rules.	2023-10-04

Rule group and rules	Description	Date
Known bad inputs managed rule group • `Host_localhost_HEADER` • `Log4J*` • `JavaDeserialization*`	Exception deployment: released static version 1.18 of this rule group. This is a quick rollout of this static version to accommodate the creation and rollout of version 1.19. Updated the `Host_localhost_HEADER` rule and all Log4J and Java deserialization rules for improved detection. For information about this deployment type, see Exception deployments for AWS Managed Rules.	2023-10-04

Rule group and rules	Description	Date
AWS WAF Bot Control rule group • `TGT_TokenReuseIp` • `TGT_ML_CoordinatedActivityMedium` • `TGT_ML_CoordinatedActivityHigh`	Added rules to the rule group with Count action. The token reuse IP rule detects and counts token sharing across IP addresses. The coordinated activity rules use automated, machine-learning (ML) analysis of website traffic to detect bot-related activity. In your rule group configuration, you can opt out of the use of ML. With this release, customers who are currently using the targeted protection level are opted in to the use of ML. Opting out disables the coordinated activity rules.	2023-09-06
AWS WAF Bot Control rule group • `CategoryAI`	Added the rule `CategoryAI` to the rule group.	2023-08-30

Rule group and rules	Description	Date
Core rule set (CRS) managed rule group • `RestrictedExtensions_URIPATH` • `RestrictedExtensions_QUERYARGUMENTS` • `EC2MetaDataSSRF_COOKIE` • `EC2MetaDataSSRF_QUERYARGUMENTS` • `EC2MetaDataSSRF_BODY` • `EC2MetaDataSSRF_URIPATH`	Released static version 1.7 of this rule group. Updated restricted extensions and EC2 metadata SSRF rules to improve detection and reduce false positives.	2023-07-26
AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group All rules in new rule group	Added the rule group `AWSManagedRulesACFPRuleSet` .	2023-06-13
Linux operating system managed rule group • `LFI_HEADER` • `LFI_URIPATH` • `LFI_QUERYSTRING`	Released static version 2.2 of this rule group. Added signatures to improve detection.	2023-05-22

Rule group and rules	Description	Date
Core rule set (CRS) managed rule group • `RestrictedExtensions_URIPATH` • `RestrictedExtensions_QUERYARGUMENTS` • `CrossSiteScripting_COOKIE` • `CrossSiteScripting_QUERYARGUMENTS` • `CrossSiteScripting_BODY` • `CrossSiteScripting_URIPATH`	Released static version 1.6 of this rule group. Updated cross-site scripting (XSS) and restricted extension rules to improve detection and reduce false positives.	2023-04-28

Rule group and rules	Description	Date
PHP application managed rule group • Updated PHPHighRiskMethodsVariables_BODY • Removed PHPHighRiskMethodsVariables_QUERYARGUMENTS • Added PHPHighRiskMethodsVariables_QUERYSTRING • Added PHPHighRiskMethodsVariables_HEADER	Released static version 2.0 of this rule group. Added signatures to improve detection in all rules. Replaced the rule PHPHighRiskMethodsVariables_QUERYARGUMENTS with PHPHighRiskMethodsVariables_QUERYSTRING, which inspects the entire query string instead of just the query arguments. Added the rule PHPHighRiskMethodsVariables_HEADER, to expand coverage to include all headers. Updated the following labels to align with standard AWS Managed Rules labeling: • Old name: PHPHighRiskMethodsVariables_BODY New name: PHPHighRiskMethodsVariables_Body • Old name: PHPHighRiskMethodsVariables_QUERYARGUMENTS New name: PHPHighRi	2023-02-27

Rule group and rules	Description	Date
	skMethodsVariables_QueryString	
AWS WAF Fraud Control account takeover prevention (ATP) rule group • VolumetricIpFailedLoginResponseHigh • VolumetricSessionFailedLoginResponseHigh	Added login response inspection rules for use with protected Amazon CloudFront distributions. These rules can block new login attempts from IP addresses and client sessions that have recently been the source of too many failed login attempts.	2023-02-15
Core rule set (CRS) managed rule group • NoUserAgent_HEADER • CrossSiteScripting_COOKIE • CrossSiteScripting_QUERYARGUMENTS • CrossSiteScripting_BODY • CrossSiteScripting_URIPATH	Released static version 1.5 of this rule group. Updated Cross Site Scripting (XSS) filters to improve detection.	2023-01-25

Rule group and rules	Description	Date
Linux operating system managed rule group • LFI_COOKIE - removed • LFI_HEADER - added • LFI_URIPATH • LFI_QUERYSTRING	Released static version 2.1 of this rule group. Removed the rule LFI_COOKIE and its label awswaf:managed:aws:linux-os:LFI_Cookie , and replaced them with the new rule LFI_HEADER and its label awswaf:managed:aws:linux-os:LFI_Header . This change expands inspection to multiple headers. Added text transformations and signatures to all rules to improve detection.	2022-12-15
Core rule set (CRS) managed rule group • NoUserAgent_HEADER • CrossSiteScripting_COOKIE • CrossSiteScripting_QUERYARGUMENTS • CrossSiteScripting_BODY • CrossSiteScripting_URIPATH	Released static version 1.4 of this rule group. Added a text transformation to NoUserAgent_HEADER to remove all null bytes. Updated the filters in the cross-site scripting rules to improve detection.	2022-12-05

Rule group and rules	Description	Date
Known bad inputs managed rule group • `JavaDeserializationRCE_BODY` • `JavaDeserializationRCE_URIPATH` • `JavaDeserializationRCE_HEADER` • `JavaDeserializationRCE_QUERYSTRING` • `Host_localhost_HEADER`	Released static version 1.17 of this rule group. Updated the Java deserialization rules to add detection for requests matching Apache CVE-2022-42889, a remote code execution (RCE) vulnerability in Apache Commons Text versions prior to 1.10.0. For more information, see NIST: National Vulnerability Database: CVE-2022-42889 Detail and CVE-2022-42889: Apache Commons Text prior to 1.10.0 allows RCE when applied to untrusted input due to insecure interpolation defaults. Improved detection in `Host_localhost_HEADER` .	2022-10-20

Rule group and rules	Description	Date
Known bad inputs managed rule group • Log4JRCE_HEADER • Log4JRCE_QUERYSTRING • Log4JRCE_URIPATH • Log4JRCE_BODY	Released static version 1.16 of this rule group. Removed false positives that AWS identified in version 1.15.	2022-10-05
POSIX operating system managed rule group PHP application managed rule group WordPress application managed rule group	Corrected the documented label names.	2022-09-19
IP reputation rule groups • AWSManagedIPDDoSList	This change doesn't alter how the rule group handles web traffic. Added a new rule with Count action to inspect for IP addresses that are actively engaging in DDoS activities, according to Amazon threat intelligence.	2022-08-30

Rule group and rules	Description	Date
Known bad inputs managed rule group • Log4JRCE • Log4JRCE_HEADER • Log4JRCE_QUERYSTRING • Log4JRCE_URIPATH • Log4JRCE_BODY • JavaDeserializationRCE_HEADER • JavaDeserializationRCE_BODY • JavaDeserializationRCE_URIPATH • JavaDeserializationRCE_QUERYSTRING • Host_localhost_HEADER • PROPFIND_METHOD	Released static version 1.15 of this rule group. Removed Log4JRCE and replaced it with Log4JRCE_HEADER , Log4JRCE_QUERYSTRING , Log4JRCE_URI , and Log4JRCE_BODY , for more granular monitoring and management of false positives. Added signatures for improved detection and blocking to PROPFIND_METHOD and to all JavaDeserializationRCE* and Log4JRCE* rules. Updated labels to correct capitalization in Host_localhost_HEADER and in all JavaDeserializationRCE* rules. Corrected the description of JavaDeserializationRCE_HEADER .	2022-08-22

Rule group and rules	Description	Date
AWS WAF Fraud Control account takeover prevention (ATP) rule group • `UnsupportedCognitoIDP`	Added a rule to prevent the use of the account takeover prevention managed rule group for Amazon Cognito user pool web traffic.	2022-08-11
Core rule set (CRS) managed rule group	AWS has scheduled expiration for versions `Version_1.2` and `Version_2.0` of the rule group. The versions will expire on September 9, 2022. For information about version expiration, see Using versioned managed rule groups in AWS WAF.	2022-06-09
Core rule set (CRS) managed rule group • `GenericLFI_URIPATH` `GenericRFI_URIPATH`	Released version 1.3 of this rule group. This release updates the match signatures in the rules `GenericLFI_URIPATH` and `GenericRFI_URIPATH`, to improve detection.	2022-05-24
AWS WAF Bot Control rule group • `CategoryEmailClient`	Added the rule `CategoryEmailClient` to the rule group.	2022-04-06

Rule group and rules	Description	Date
Known bad inputs managed rule group • JavaDeserializationRCE_HEADER • JavaDeserializationRCE_BODY • JavaDeserializationRCE_URI • JavaDeserializationRCE_QUERYSTRING	Released version 1.14 of this rule group. The four JavaDeserializtion RCE rules are moved to Block mode.	2022-03-31
Known bad inputs managed rule group • JavaDeserializationRCE_HEADER_RC_COUNT • JavaDeserializationRCE_BODY_RC_COUNT • JavaDeserializationRCE_URI_RC_COUNT • JavaDeserializationRCE_QUERYSTRING_RC_COUNT	Released version 1.13 of this rule group. Updated the text transformation for Spring Core and Cloud Function RCE vulnerabilities. These rules are in count mode to gather metrics and evaluate matched patterns. The label can be used to block requests in a custom rule. A subsequent version will be deployed with these rules in block mode.	2022-03-31

Rule group and rules	Description	Date
Known bad inputs managed rule group • JavaDeserializationRCE_HEADER_RC_COUNT • JavaDeserializationRCE_BODY_RC_COUNT • JavaDeserializationRCE_URI_RC_COUNT • JavaDeserializationRCE_QUERYSTRING_RC_COUNT • Log4JRCE_HEADER • Log4JRCE_QUERYSTRING • Log4JRCE_URI • Log4JRCE_BODY • Log4JRCE	Released version 1.12 of this rule group. Added signatures for Spring Core and Cloud Function RCE vulnerabilities. These rules are in count mode to gather metrics and evaluate matched patterns. The label can be used to block requests in a custom rule. A subsequent version will be deployed with these rules in block mode. Removed the rules Log4JRCE_HEADER , Log4JRCE_QUERYSTRING , Log4JRCE_URI , and Log4JRCE_BODY and replaced them with the rule Log4JRCE.	2022-03-30
IP reputation rule groups • AWSManagedReconnaissanceList	Updated the AWSManagedReconnaissanceList rule to change the action from count to block.	2022-02-15

Rule group and rules	Description	Date
AWS WAF Fraud Control account takeover prevention (ATP) rule group All rules in new rule group	Added the rule group `AWSManagedRulesATPRuleSet` .	2022-02-11
Known bad inputs managed rule group • `Log4JRCE` • `Log4JRCE_HEADER` • `Log4JRCE_QUERYSTRING` • `Log4JRCE_URI` • `Log4JRCE_BODY`	Released version 1.9 of this rule group. Removed the rule `Log4JRCE` and replaced it with the rules `Log4JRCE_HEADER` , `Log4JRCE_QUERYSTRING` , `Log4JRCE_URI` , and `Log4JRCE_BODY` , for flexibility in the use of this functionality. Added signatures to improve detection and blocking.	2022-01-28
Core rule set (CRS) • `CrossSiteScripting_URIPATH` • `CrossSiteScripting_BODY` • `CrossSiteScripting_QUERYARGUMENTS` • `CrossSiteScripting_COOKIE`	Released version 2.0 of this rule group. For these rules, tuned detection signatures to reduce false positives. Replaced the `URL_DECODE` text transformation with the double `URL_DECODE_UNI` text transformation. Added the `HTML_ENTITY_DECODE` text transformation.	2022-01-10

Rule group and rules	Description	Date
Core rule set (CRS) • RestrictedExtensions_URIPATH • RestrictedExtensions_QUERYARGUMENTS	As part of the release of version 2.0 of this rule group, added the URL_DECODE_UNI text transformation. Removed the URL_DECODE text transformation from RestrictedExtensions_URIPATH .	2022-01-10
SQL database • SQLi_BODY • SQLi_QUERYARGUMENTS • SQLi_COOKIE • SQLi_URIPATH • SQLiExtendedPatterns_BODY • SQLiExtendedPatterns_QUERYARGUMENTS	Released version 2.0 of this rule group. Replaced the URL_DECODE text transformation with the double URL_DECODE_UNI text transformation and added the COMPRESS_WHITE_SPACE text transformation. Added more detection signatures to SQLiExtendedPatterns_QUERYARGUMENTS . Added JSON inspection to SQLi_BODY . Added the rule SQLiExtendedPatterns_BODY . Removed the rule SQLi_URIPATH .	2022-01-10

Rule group and rules	Description	Date
Known bad inputs • Log4JRCE	Released version 1.8 of the rule Log4JRCE to improve header inspection and matching criteria.	2021-12-17
Known bad inputs • Log4JRCE	Released version 1.4 of the rule Log4JRCE to tune the matching criteria and to inspect additional headers. Released version 1.5 to tune the matching criteria.	2021-12-11
Known bad inputs • Log4JRCE • BadAuthToken_COOKIE_AUTHORIZATION	Added the rule Log4JRCE version 1.2 in response to the recently disclosed security issue within Log4j. For information see CVE-2021-44228. This rule inspects common URI paths, query strings, the first 8KB of the request body, and common headers. The rule uses double URL_DECODE_UNI text transformations. Released version 1.3 of Log4JRCE to tune the matching criteria and to inspect additional headers. Removed the rule BadAuthToken_COOKIE_AUTHORIZATION .	2021-12-10

The following table lists changes prior to December, 2021.

Rule group and rules	Description	Date	
Amazon IP reputation list	`AWSManagedReconnaissanceList`	Added the `AWSManagedReconnaissanceList` rule in monitoring/count mode. This rule contains IP addresses that are performing reconnaissance against AWS resources.	2021-11-23

Windows operating system	`WindowsShellCommands`
	
	`PowerShellCommands`
		Added three new rules for WindowsShell commands: `WindowsShellCommands_COOKIE`, `WindowsShellCommands_QUERYARGUMENTS`, and `WindowsShellCommands_BODY`.
		
		Added a new PowerShell rule: `PowerShellCommands_COOKIE`.
		
		Restructured the `PowerShellComands` rules
			2021-11-23

Rule group and rules	Description	Date	
		naming by removing the string _Set1 and _Set2. Added more comprehensive detection signatures to `PowerShellRules`. Added `URL_DECODE_UNI` text transformation to all Windows operating system rules.	

Linux operating system	`LFI_URIPATH`
	
	`LFI_QUERYSTRING`
	
	`LFI_BODY`
	
	`LFI_COOKIE`
		Replaced double `URL_DECODE` text transformation with double `URL_DECODE_UNI`.
		
		Added `NORMALIZE_PATH_WIN` as a second text transformation.
		
		Replaced the `LFI_BODY` rule with the `LFI_COOKIE` rule.
		
		Added more comprehensive detection signatures for all `LFI` rules.
			2021-11-23

Rule group and rules	Description	Date	
Core rule set (CRS)	`SizeRestrictions_BODY`	Reduced the size limit to block web requests with body payloads larger than 8 KB. Previously, the limit was 10 KB.	2021-10-27

Core rule set (CRS)	`EC2MetaDataSSRF_BODY`
	`EC2MetaDataSSRF_COOKIE`
	`EC2MetaDataSSRF_URIPATH`
	`EC2MetaDataSSRF_QUERYARGUMENTS`
		Added more detection signatures. Added double unicode URL decode to improve blocking.	2021-10-27

Core rule set (CRS)	`GenericLFI_QUERYARGUMENTS`
	`GenericLFI_URIPATH`
	`RestrictedExtensions_URIPATH`
	`RestrictedExtensions_QUERYARGUMENTS`
		Added double unicode URL decode to improve blocking.	2021-10-27

Rule group and rules	Description	Date	

Core rule set (CRS)	`GenericRFI_QUERYARGUMENTS`
	`GenericRFI_BODY`
	`GenericRFI_URIPATH`
		Updated the rule signatures to reduce false positives, based on customer feedback. Added double unicode URL decode to improve blocking.	2021-10-27

All	All rules	Added support for AWS WAF labels to all rules that didn't already support labeling.	2021-10-25
Amazon IP reputation list	`AWSManagedIPReputationList_xxxx`	Restructured the IP reputation list, removed suffixes from rule name, and added support for AWS WAF labels.	2021-05-04
Anonymous IP list	`AnonymousIPList` `HostingProviderList`	Added support for AWS WAF labels.	2021-05-04
Bot Control	All	Added the Bot Control rule set.	2021-04-01
Core rule set (CRS)	`GenericRFI_QUERYARGUMENTS`	Added double URL decode.	2021-03-03

Rule group and rules	Description	Date	
Core rule set (CRS)	`RestrictedExtensions_URIPATH`	Improved the configuration of the rules and added an extra URL decode.	2021-03-03
Admin protection	`AdminProtection_URIPATH`	Added double URL decode.	2021-03-03
Known bad inputs	`ExploitablePaths_URIPATH`	Improved the configuration of the rules and added an extra URL decode.	2021-03-03
Linux operating system	`LFI_QUERYARGUMENTS`	Improved the configuration of the rules and added an extra URL decode.	2021-03-03
Windows operating system	All	Improved the configuration of the rules.	2020-09-23

PHP application	`PHPHighRiskMethodsVariables_QUERYARGUMENTS`
	
	`PHPHighRiskMethodsVariables_BODY`
		Changed the text transformation from HTML decode to URL decode, to improve blocking.	2020-09-16

Rule group and rules	Description	Date	

POSIX operating system	`UNIXShellCommandsVariables_QUERYARGUMENTS`
	
	`UNIXShellCommandsVariables_BODY`
		Changed the text transformation from HTML decode to URL decode, to improve blocking.	2020-09-16

Core rule set	`GenericLFI_QUERYARGUMENTS`
	
	`GenericLFI_URIPATH`
	
	GenericLFI_BODY
		Changed the text transformation from HTML decode to URL decode, to improve blocking.	2020-08-07

Linux operating system	`LFI_URIPATH`
	
	`LFI_QUERYARGUMENTS`
	
	`LFI_BODY`
		Changed the text transformation from HTML entity decode to URL decode, to improve detection and blocking.	2020-05-19

Anonymous IP List	All	New rule group in IP reputation rule groups to block requests from services that permit the obfuscation of viewer identity, to help mitigate bots and evasion of geographic restrictions.	2020-03-06

Rule group and rules	Description	Date	
WordPress application	`WordPressExploitableCommands_QUERYSTRING`	New rule that checks for exploitable commands in the query string.	2020-03-03

Core rule set (CRS)	`SizeRestrictions_QUERYSTRING`
	
	`SizeRestrictions_Cookie_HEADER`
	
	`SizeRestrictions_BODY`
	
	`SizeRestrictions_URIPATH`
		Adjusted the size value constraints for improved accuracy.	2020-03-03

SQL database	`SQLi_URIPATH`	The rules now check the message URI.	2020-01-23

SQL database	`SQLi_BODY`
	
	`SQLi_QUERYARGUMENTS`
	
	`SQLi_COOKIE`
		Updated text transformations.	2019-12-20

Rule group and rules	Description	Date	

Core rule set (CRS)	`CrossSite Scripting _URIPATH`
	
	`CrossSite Scripting_BODY`
	
	`CrossSite Scripting _QUERYARG UMENTS`
	
	`CrossSite Scripting _COOKIE`
		Updated text transformations.	2019-12-20

Managing your own rule groups

You can create your own rule group to reuse collections of rules that you either don't find in the managed rule group offerings or that you prefer to handle on your own.

Rule groups that you create hold rules just like a protection pack (web ACL) does, and you add rules to a rule group in the same way as you do to a protection pack (web ACL). When you create your own rule group, you must set an immutable maximum capacity for it.

Topics

- Creating a rule group
- Editing a rule group
- Using your rule group in a protection pack (web ACL)
- Deleting a rule group
- Sharing a rule group

Creating a rule group

To create a new rule group, follow the procedure on this page.

To create a rule group

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **Rule groups**, and then **Create rule group**.
3. Enter a name and description for the rule group. You'll use these to identify the rule set to manage it and use it.

   Don't use names that start with `AWS`, `Shield`, `PreFM`, or `PostFM`. These strings are either reserved or could cause confusion with rule groups that are managed for you by other services. See Recognizing rule groups provided by other services.

   > **Note**
   >
   > You can't change the name after you create the rule group.

4. For **Region**, choose the Region where you want to store the rule group. To use a rule group in protection packs (web ACLs) that protect Amazon CloudFront distributions, you must use the global setting. You can use the global setting for regional applications, too.
5. Choose **Next**.
6. Add rules to the rule group using the **Rule builder** wizard, the same as you do in protection pack (web ACL) management. The only difference is that you can't add a rule group to another rule group.
7. For **Capacity**, set the maximum for the rule group's use of protection pack (web ACL) capacity units (WCUs). This is an immutable setting. For information about WCUs, see Web ACL capacity units (WCUs) in AWS WAF.

   As you add rules to the rule group, the **Add rules and set capacity** pane displays the minimum required capacity, which is based on the rules that you've already added. You can use this and your future plans for the rule group to help estimate the capacity that the rule group will require.
8. Review the settings for the rule group, and choose **Create**.

Editing a rule group

To add or remove rules from a rule group or change configuration settings, access the rule group using the procedure on this page.

Production traffic risk

If you change a rule group that you're currently using in a protection pack (web ACL), those changes will affect your protection pack (web ACL) behavior wherever it's being used. Be sure to test and tune all changes in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune your updated rules in count mode with your production traffic before enabling them. For guidance, see Testing and tuning your AWS WAF protections.

To edit a rule group

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **Rule groups**.
3. Choose the name of the rule group that you want to edit. The console takes you to the rule group's page.

   Note

   If you don't see the rule group that you want to edit, check the Region selection inside the **Rule groups** section. For rule groups used to protect Amazon CloudFront distributions, use the **Global (CloudFront)** setting.

4. Edit the rule group as needed. You can edit the rule group's mutable properties, similar to how you did during creation. The console saves your changes as you go.

   Note

   If you change the name of a rule and you want the rule's metric name to reflect the change, you must update the metric name as well. AWS WAF doesn't automatically update the metric name for a rule when you change the rule name. You can change the metric name when you edit the rule in the console, by using the rule JSON editor. You can also change both names through the APIs and in any JSON listing that you use to define your protection pack (web ACL) or rule group.

Temporary inconsistencies during updates

When you create or change a protection pack (web ACL) or other AWS WAF resources, the changes take a small amount of time to propagate to all areas where the resources are stored. The propagation time can be from a few seconds to a number of minutes.

The following are examples of the temporary inconsistencies that you might notice during change propagation:

- After you create a protection pack (web ACL), if you try to associate it with a resource, you might get an exception indicating that the protection pack (web ACL) is unavailable.
- After you add a rule group to a protection pack (web ACL), the new rule group rules might be in effect in one area where the protection pack (web ACL) is used and not in another.
- After you change a rule action setting, you might see the old action in some places and the new action in others.
- After you add an IP address to an IP set that is in use in a blocking rule, the new address might be blocked in one area while still allowed in another.

Using your rule group in a protection pack (web ACL)

To use a rule group in a protection pack (web ACL), you add it to the protection pack (web ACL) in a rule group reference statement.

Production traffic risk

Before you deploy changes in your protection pack (web ACL) for production traffic, test and tune them in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune your updated rules in count mode with your production traffic before enabling them. For guidance, see Testing and tuning your AWS WAF protections.

Note

Using more than 1,500 WCUs in a protection pack (web ACL) incurs costs beyond the basic protection pack (web ACL) price. For more information, see Web ACL capacity units (WCUs) in AWS WAF and AWS WAF Pricing.

To use a rule group

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **Rule groups**.
3. Choose the name of the rule group that you want to use.
4. Choose **Add rules**, and then choose **Add my own rules and rule groups**.
5. Choose **Rule group** and select your rule group from the list.

In your protection pack (web ACL), you can alter the behavior of a rule group and its rules by setting the individual rule actions to Count or any other action. This can help you do things like test a rule group, identify false positives from rules in a rule group, and customize how a managed rule group handles your requests. For more information, see Overriding rule group actions in AWS WAF.

If your rule group contains a rate-based statement, each protection pack (web ACL) where you use the rule group has its own separate rate tracking and management for the rate-based rule, independent of any other protection pack (web ACL) where you use the rule group. For more information, see Using rate-based rule statements in AWS WAF.

Temporary inconsistencies during updates

When you create or change a protection pack (web ACL) or other AWS WAF resources, the changes take a small amount of time to propagate to all areas where the resources are stored. The propagation time can be from a few seconds to a number of minutes.

The following are examples of the temporary inconsistencies that you might notice during change propagation:

- After you create a protection pack (web ACL), if you try to associate it with a resource, you might get an exception indicating that the protection pack (web ACL) is unavailable.
- After you add a rule group to a protection pack (web ACL), the new rule group rules might be in effect in one area where the protection pack (web ACL) is used and not in another.
- After you change a rule action setting, you might see the old action in some places and the new action in others.
- After you add an IP address to an IP set that is in use in a blocking rule, the new address might be blocked in one area while still allowed in another.

Deleting a rule group

Follow the guidance in this section to delete a rule group.

Deleting referenced sets and rule groups

When you delete an entity that you can use in a protection pack (web ACL), like an IP set, regex pattern set, or rule group, AWS WAF checks to see if the entity is currently being used in a protection pack (web ACL). If it finds that it is in use, AWS WAF warns you. AWS WAF is almost always able to determine if an entity is being referenced by a protection pack (web ACL). However, in rare cases it might not be able to do so. If you need to be sure that nothing is currently using the entity, check for it in your protection packs (web ACLs) before deleting it. If the entity is a referenced set, also check that no rule groups are using it.

To delete a rule group

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **Rule groups**.
3. Choose the rule group that you want to delete, and then choose **Delete**.

> **Note**
>
> If you don't see the rule group that you want to delete, check the Region selection inside the **Rule groups** section. For rule groups used to protect Amazon CloudFront distributions, use the **Global (CloudFront)** setting.

Sharing a rule group

You can share a rule group with other acccounts, for use by those accounts.

Sharing a rule group

You can share with one or more specific accounts, and you can share with all accounts in an organization.

To share a rule group, you use the AWS WAF API to create a policy for the rule group sharing that you want. For more information, see PutPermissionPolicy in the *AWS WAF API Reference*.

Using a rule group that's been shared with you

If a rule group has been shared with your account, you can access it through the API and you can reference it when you create or update your protection packs (web ACLs) through the API. For more information, see GetRuleGroup, CreateWebACL, and UpdateWebACL in the *AWS WAF API Reference*. Rule groups that are shared with you don't appear in your AWS WAF console rule groups listing.

AWS Marketplace rule groups

This section explains how to use AWS Marketplace rule groups.

AWS Marketplace rule groups are available by subscription through the AWS Marketplace console at AWS Marketplace. After you subscribe to an AWS Marketplace rule group, you can use it in AWS WAF. To use an AWS Marketplace rule group in an AWS Firewall Manager AWS WAF policy, each account in your organization must subscribe to it.

You can subscribe to different types of rule groups through AWS Marketplace:

- AWS WAF partner-managed rule groups
- Client-side protections

Test and tune any changes to your AWS WAF protections before you use them for production traffic. For information, see Testing and tuning your AWS WAF protections.

AWS Marketplace Rule Group Pricing

AWS Marketplace rule groups are available with no long-term contracts, and no minimum commitments. When you subscribe to a rule group, you're charged a monthly fee (prorated hourly) and ongoing request fees based on volume. However, you're only charged the subscription fee when you add the subscribed rule group to a web ACL and begin using it. For more information, see AWS WAF Pricing and the description for each AWS Marketplace rule group at AWS Marketplace.

Have questions about an AWS Marketplace rule group?

For questions about a rule group that's managed by an AWS Marketplace seller and to request changes in functionality, contact the provider's customer support team. To find contact information, see the provider's listing at AWS Marketplace.

The AWS Marketplace rule group provider determines how to manage the rule group, for example how to update the rule group and whether the rule group is versioned. The provider also

determines the details of the rule group, including the rules, rule actions, and any labels that the rules add to matching web requests.

Subscribing to AWS Marketplace rule groups

You can subscribe to and unsubscribe from AWS Marketplace rule groups on the AWS WAF console.

> **Important**
>
> To use an AWS Marketplace rule group in an AWS Firewall Manager policy, each account in your organization must first subscribe to that rule group.

To subscribe to an AWS Marketplace rule group

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **Add-on protections**.
3. In the **AWS Marketplace** section, choose the name of a rule group to view the details and pricing information.

   > **Tip**
   >
   > Use the filters to quickly sort for the rules you're most interested in. For example, you can use the **Category** filter to view client-side protections only.

4. To subscribe to an AWS Marketplace rule group:

   a. Navigate to a rule group, then choose **Subscribe via Marketplace**.

   b. In the Marketplace page that opens, choose **View purchase options**, then choose **Subscribe**.

   > **Note**
   >
   > If you decide not to subscribe to the rule group, simply close the pop-up.

After you're subscribed to an AWS Marketplace rule group, you use it in your protection packs (web ACLs) as you do other managed rule groups. For information, see Creating a protection pack (web ACL) in AWS WAF.

When adding a rule group to a protection pack (web ACL), you can override the actions of rules in the rule group and of the rule group result. For more information, see Overriding rule group actions in AWS WAF.

Unsubscribing from AWS Marketplace rule groups

You can unsubscribe from AWS Marketplace rule groups on the AWS Marketplace console.

> **Important**
>
> To stop the subscription charges for an AWS Marketplace rule group, you must remove it from all protection packs (web ACLs) in AWS WAF and in any Firewall Manager AWS WAF policies, in addition to unsubscribing from it. If you unsubscribe from an AWS Marketplace rule group but don't remove it from your protection packs (web ACLs), you will continue to be charged for the subscription.

To unsubscribe from an AWS Marketplace rule group

1. Remove the rule group from all protection packs (web ACLs). For more information, see Editing a protection pack (web ACL) in AWS WAF.
2. Open the AWS console at https://console.aws.amazon.com/marketplace.

   The **Manage subscriptions page appears**.
3. Open the **Delivery method** list and choose **SaaS**.
4. Under **Agreement**, open the **Actions list** and choose **Cancel subscription** next to the name of the rule group that you want to unsubscribe from.
5. In the **Cancel subscription** dialog box, enter **`confirm`**, then choose **Yes, cancel subscription**.

Troubleshooting AWS Marketplace rule groups

If you find that an AWS Marketplace rule group is blocking legitimate traffic, you can troubleshoot the problem by performing the following steps.

To troubleshoot an AWS Marketplace rule group

1. Override the actions to count for the rules that are blocking legitimate traffic. You can identify which rules are blocking specific requests using either the AWS WAF sampled requests or AWS WAF logs. You can identify the rules by looking at the `ruleGroupId` field in the log or the `RuleWithinRuleGroup` in the sampled request. You can identify the rule in the pattern `<Seller Name>#<RuleGroup Name>#<Rule Name>`.
2. If setting specific rules to only count requests doesn't solve the problem, you can override all of the rule actions or change the action for the AWS Marketplace rule group itself from **No override** to **Override to count**. This allows the web request to pass through, regardless of the individual rule actions within the rule group.
3. After overriding either the individual rule action or the entire AWS Marketplace rule group action, contact the rule group provider's customer support team to further troubleshoot the issue. For contact information, see the rule group listing on the product listing pages on AWS Marketplace.

Contacting AWS support

For problems with AWS WAF or a rule group that is managed by AWS, contact AWS Support. For problems with a rule group that is managed by an AWS Marketplace seller, contact the provider's customer support team. To find contact information, see the provider's listing on AWS Marketplace.

Recognizing rule groups provided by other services

If you or an administrator in your organization uses AWS Firewall Manager or AWS Shield Advanced to manage resource protections using AWS WAF, you might see rule group reference statements added to protection packs (web ACLs) in your account.

The names of these rule groups begin with the following strings:

- **`ShieldMitigationRuleGroup`** – These rule groups are managed by AWS Shield Advanced and used to provide automatic application layer DDoS mitigation to protected application layer (layer 7) resources.

  When you enable automatic application layer DDoS mitigation for a protected resource, Shield Advanced adds one of these rule groups to the protection pack (web ACL) that you have associated with the resource. Shield Advanced assigns the rule group reference statement a priority setting of 10,000,000, so that it runs after the rules that you have configured in the

protection pack (web ACL). For more information about these rule groups, see Automating application layer DDoS mitigation with Shield Advanced.

> **Warning**
>
> Don't try to manually manage this rule group in your protection pack (web ACL). In particular, don't manually delete the `ShieldMitigationRuleGroup` rule group reference statement from your protection pack (web ACL). Doing this could have unintended consequences for all resources that are associated with the protection pack (web ACL). Instead, use Shield Advanced to disable automatic mitigation for the resources that are associated with the protection pack (web ACL). Shield Advanced will remove the rule group for you when it's not needed for automatic mitigation.

- **`PREFMManaged` and `POSTFMManaged`** – These rule groups are managed by AWS Firewall Manager based on Firewall Manager AWS WAF policy configurations. Firewall Manager provides these rule groups inside protection packs (web ACLs) that Firewall Manager manages.

  Firewall Manager creates protection packs (web ACLs) for you with names that begin with `FMManagedWebACLV2`. You can configure Firewall Manager to retrofit your existing protection packs (web ACLs) as well. For these, the protection pack (web ACL) name is the one that you specified when you created it. In either case, Firewall Manager will add these rule groups to the protection pack (web ACL). For more information, see Using AWS WAF policies with Firewall Manager.

Web ACL capacity units (WCUs) in AWS WAF

This section explains what web ACL capacity units (WCUs) are and how they work.

AWS WAF uses WCUs to calculate and control the operating resources that are required to run your rules, rule groups, and web ACLs. AWS WAF enforces WCU limits when you configure your rule groups and web ACLs. WCUs don't affect how AWS WAF inspects web traffic.

AWS WAF manages capacity for rules, rule groups, and web ACLs.

Rule WCUs

AWS WAF calculates rule capacity when you create or update a rule. AWS WAF calculates capacity differently for each rule type, to reflect each rule's relative cost. Simple rules that cost little to run use fewer WCUs than more complex rules that use more processing power. For example, a size

constraint rule statement uses fewer WCUs than a statement that inspects requests using a regex pattern set.

Rule capacity requirements generally start at a base cost for the rule type and increase with complexity, for example, when you add text transformations before inspection or if you inspect the JSON body. For information about rule capacity requirements, see the listings for the rule statements at Using rule statements in AWS WAF.

Rule group WCUs

The WCU requirements for a rule group are determined by the rules that you define inside the rule group. The maximum capacity for a rule group is 5,000 WCUs.

Each rule group has an immutable capacity setting, which the owner assigns at creation. This is true for managed rule groups and rule groups that you create through AWS WAF. When you modify a rule group, your changes must keep the rule group's WCUs within its capacity. This ensures that protection packs (web ACLs) or web ACLs that are using the rule group remain within their capacity requirements.

The WCUs that are in use in a rule group is the sum of the WCUs for the rules minus any processing optimizations that AWS WAF is able to obtain by combining the behavior of the rules. For example, if you define two rules to examine the same web request component, and the rules each apply a particular transformation to the component before inspecting it, AWS WAF might be able to charge you just once for applying the transformation. The WCU cost to use a rule group in a protection pack (web ACL) is always the fixed WCU setting that you defined at the rule group creation.

When you create a rule group, take care to set the capacity high enough to accommodate the rules that you'll want to use throughout the rule group's lifetime.

Protection pack or web ACL WCUs

The WCU requirements for a protection pack (web ACL) are determined by the rules and rule groups that you use inside the protection pack (web ACL).

- The cost of using a rule group in a protection pack (web ACL) is the rule group's capacity setting.
- The cost of using a rule is the rule's calculated WCUs minus any processing optimizations that AWS WAF is able to obtain from the protection pack (web ACL)'s combination of rules. For example, if you define two rules to examine the same web request component, and the rules each apply a particular transformation to the component before inspecting it, AWS WAF might be able to charge you just once for applying the transformation.

The basic price for a protection pack (web ACL) includes up to 1,500 WCUs. Using more than 1,500 WCUs incurs additional fees, according to a tiered pricing model. AWS WAF automatically adjusts your protection pack (web ACL) pricing as your protection pack (web ACL) WCU usage changes. For pricing details, see AWS WAF Pricing.

The maximum capacity for a protection pack (web ACL) is 5,000 WCUs.

Determining the WCUs for a rule group, protection pack (web ACL), or web ACL

As noted in prior sections, the total WCUs used in a rule group, protection pack (web ACL), or web ACL will be equal to *or less than* the sum of the WCUs for all of the rules that are defined in the rule group, protection pack (web ACL), or web ACL.

In the AWS WAF console, you can see the capacity consumed when you add rules to your protection pack (web ACL), web ACL, or rule group. The console displays the current capacity units used as you add the rules.

Through the API, you can check the maximum capacity requirements for the rules that you want to use in a protection pack (web ACL), web ACL, or rule group. To do this, provide the JSON listing of the rules to the check capacity call. For more information, see CheckCapacity in the *AWS WAFV2 API Reference*.

Oversize web request components in AWS WAF

This section explains how to manage the size limits on inspecting the web request body, headers, and cookies in AWS WAF.

AWS WAF doesn't support inspecting very large contents for the web request components body, headers, or cookies. The underlying host service has count and size limits on what it forwards to AWS WAF for inspection. For example, the host service doesn't send more than 200 headers to AWS WAF, so for a web request with 205 headers, AWS WAF can't inspect the last 5 headers.

When AWS WAF allows a web request to proceed to your protected resource, the entire web request is sent, including any contents that are outside of the count and size limits that AWS WAF was able to inspect.

Component inspection size limits

The component inspection size limits are as follows:

- **`Body` and `JSON Body`** – For Application Load Balancer and AWS AppSync, AWS WAF can inspect the first 8 KB of the body of a request. For CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway, by default, AWS WAF can inspect the first 16 KB, and you can increase the limit up to 64 KB in your protection pack (web ACL) configuration. For more information, see Considerations for managing body inspection in AWS WAF.
- **`Headers`** – AWS WAF can inspect at most the first 8 KB (8,192 bytes) of the request headers and at most the first 200 headers. The content is available for inspection by AWS WAF up to the first limit reached.

  These limits apply when you inspect all headers in a request. When you inspect a single header, AWS WAF can inspect the full contents of that header without these size or count restrictions.
- **`Cookies`** – AWS WAF can inspect at most the first 8 KB (8,192 bytes) of the request cookies and at most the first 200 cookies. The content is available for inspection by AWS WAF up to the first limit reached.

Oversize handling options for your rule statements

When you write a rule statement that inspects one of these request component types, you specify how to handle oversize components. Oversize handling tells AWS WAF what to do with a web request when the request component that the rule inspects is over the size limits.

The options for handling oversize components are as follows:

- **Continue** – Inspect the request component normally according to the rule inspection criteria. AWS WAF will inspect the request component contents that are within the size limits.
- **Match** – Treat the web request as matching the rule statement. AWS WAF applies the rule action to the request without evaluating it against the rule's inspection criteria.
- **No match** – Treat the web request as not matching the rule statement without evaluating it against the rule's inspection criteria. AWS WAF continues its inspection of the web request using the rest of the rules in the protection pack (web ACL) like it would do for any non-matching rule.

In the AWS WAF console, you're required to choose one of these handling options. Outside the console, the default option is Continue.

If you use the Match option in a rule that has its action set to Block, the rule will block a request whose inspected component is oversize. With any other configuration, the final disposition of the

request depends on various factors, such as the configuration of the other rules in your protection pack (web ACL) and the protection pack (web ACL)'s default action setting.

Oversize handling in rule groups that you don't own

Component size and count limitations apply to all rules that you use in your protection pack (web ACL). This includes any rules that you use but don't manage, in managed rule groups and in rule groups that are shared with you by another account.

When you use a rule group that you don't manage, the rule group might have a rule that inspects a limited request component but that doesn't handle oversized contents the way you need them to be handled. For information about how AWS Managed Rules manage oversize components, see AWS Managed Rules rule groups list. For information about other rule groups, ask your rule group provider.

Guidelines for managing oversize components in your protection pack (web ACL)

The way you handle oversize components in your protection pack (web ACL) can depend on a number of factors such as the expected size of your request component contents, your protection pack (web ACL)'s default request handling, and how other rules in your protection pack (web ACL) match and handle requests.

The general guidelines for managing oversized web request components are as follows:

- If you need to allow some requests with oversize component contents, if possible, add rules to explicitly allow only those requests. Prioritize those rules so that they run before any other rules in the protection pack (web ACL) that inspect the same component types. With this approach, you won't be able to use AWS WAF to inspect the entire contents of the oversize components that you allow to pass to your protected resource.
- For all other requests, you can prevent any additional bytes from passing through by blocking requests that go over the limit:
  - **Your rules and rule groups** – In your rules that inspect components with size limits, configure oversize handling so that you block requests that go over the limit. For example, if your rule blocks requests with specific header contents, set the oversize handling to match on requests that have oversize header content. Alternately, if your protection pack (web ACL) blocks requests by default and your rule allows specific header contents, then configure your rule's oversize handling to not match on any request that has oversize header content.
  - **Rule groups that you don't manage** – To prevent rule groups that you don't manage from allowing oversize request components, you can add a separate rule that inspects the request

component type and blocks requests that go over the limits. Prioritize the rule in your protection pack (web ACL) so that it runs before the rule groups. For example, you can block requests with oversize body content before any of your body inspection rules run in the protection pack (web ACL). The following procedure describes how to add this type of rule.

Blocking oversized web request components

You can add a rule in your protection pack (web ACL) that blocks requests with oversized components.

To add a rule that blocks oversized contents

1. When you create or edit your protection pack (web ACL), in the rules settings, choose **Add rules**, **Add my own rules and rule groups**, **Rule builder**, then **Rule visual editor**. For guidance on creating or editing a protection pack (web ACL), see Viewing web traffic metrics in AWS WAF.
2. Enter a name for your rule, and leave the **Type** setting at **Regular rule**.
3. Change the following match settings from their defaults:
   a. On **Statement**, for **Inspect**, open the dropdown and choose the web request component that you need, either **Body**, **Headers**, or **Cookies**.
   b. For **Match type**, choose **Size greater than**.
   c. For **Size**, type a number that's at least the minimum size for the component type. For headers and cookies, type 8192. In Application Load Balancer or AWS AppSync protection packs (web ACLs), for bodies, type 8192. For bodies in CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, or Amazon Bedrock AgentCore Gateway protection packs (web ACLs), if you're using the default body size limit, type 16384. Otherwise, type the body size limit that you've defined for your protection pack (web ACL).
   d. For **Oversize handling**, select **Match**.
4. For **Action**, select **Block**.
5. Choose **Add rule**.
6. After you add the rule, on the **Set rule priority** page, move it above any rules or rule groups in your protection pack (web ACL) that inspect the same component type. This gives the new rule a lower numeric priority setting, which causes AWS WAF to evaluate it first. For more information, see Setting rule priority.

Supported regular expression syntax in AWS WAF

AWS WAF supports the regular expression pattern syntax used by the PCRE library `libpcre`. The library is documented at PCRE - Perl Compatible Regular Expressions.

AWS WAF doesn't support all constructs of the library. For example, it supports some zero-width assertions, but not all. We do not have comprehensive list of the constructs that are supported. However, if you provide a regex pattern that isn't valid or use unsupported constructs, the AWS WAF API reports a failure.

AWS WAF does not support the following PCRE patterns:

- Backreferences and capturing subexpressions
- Subroutine references and recursive patterns
- Conditional patterns
- Backtracking control verbs
- The \C single-byte directive
- The \R newline match directive
- The \K start of match reset directive
- Callouts and embedded code
- Atomic grouping and possessive quantifiers

IP sets and regex pattern sets in AWS WAF

This section introduces the topics of IP sets and regex pattern sets.

AWS WAF stores some more complex information in sets that you use by referencing them in your rules. Each of these sets has a name and is assigned an Amazon Resource Name (ARN) at creation. You can manage these sets from inside your rule statements and you can access and manage them on their own, through the console navigation pane.

You can use a managed set in a rule group or protection pack (web ACL).

- To use an IP set, see IP set match rule statement.
- To use a regex pattern set see Regex pattern set match rule statement.

Temporary inconsistencies during updates

When you create or change a protection pack (web ACL) or other AWS WAF resources, the changes take a small amount of time to propagate to all areas where the resources are stored. The propagation time can be from a few seconds to a number of minutes.

The following are examples of the temporary inconsistencies that you might notice during change propagation:

- After you create a protection pack (web ACL), if you try to associate it with a resource, you might get an exception indicating that the protection pack (web ACL) is unavailable.
- After you add a rule group to a protection pack (web ACL), the new rule group rules might be in effect in one area where the protection pack (web ACL) is used and not in another.
- After you change a rule action setting, you might see the old action in some places and the new action in others.
- After you add an IP address to an IP set that is in use in a blocking rule, the new address might be blocked in one area while still allowed in another.

Topics

- Creating and managing an IP set in AWS WAF
- Creating and managing a regex pattern set in AWS WAF

Creating and managing an IP set in AWS WAF

An IP set provides a collection of IP addresses and IP address ranges that you want to use together in a rule statement. IP sets are AWS resources.

To use an IP set in a protection pack (web ACL) or rule group, you first create an AWS resource, `IPSet` with your address specifications. Then you reference the set when you add an IP set rule statement to a protection pack (web ACL) or rule group.

Creating an IP set

Follow the procedure in this section to create a new IP set.

> **Note**
>
> In addition to the procedure in this section, you have the option to add a new IP set when you add an IP match rule to your protection pack (web ACL) or rule group. Choosing that option requires you to provide the same settings as those required by this procedure.

To create an IP set

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **IP sets** and then **Create IP set**.
3. Enter a name and description for the IP set. You'll use these to identify the set when you want to use it.

   > **Note**
   >
   > You can't change the name after you create the IP set.

4. For **Region**, choose Global (CloudFront) or choose the Region where you want to store the IP set. You can use regional IP sets only in protection packs (web ACLs) that protect regional resources. To use an IP set in protection packs (web ACLs) that protect Amazon CloudFront distributions, you must use Global (CloudFront).
5. For **IP version**, select the version you want to use.
6. In the **IP addresses** text box, enter one IP address or IP address range per line, in CIDR notation. AWS WAF supports all IPv4 and IPv6 CIDR ranges except for `/0`. For more information about CIDR notation, see the Wikipedia article Classless Inter-Domain Routing.

   Here are some examples:

   - To specify the IPv4 address 192.0.2.44, type **192.0.2.44/32**.
   - To specify the IPv6 address 2620:0:2d0:200:0:0:0:0, type **2620:0:2d0:200:0:0:0:0/128**.
   - To specify the range of IPv4 addresses from 192.0.2.0 to 192.0.2.255, type **192.0.2.0/24**.
   - To specify the range of IPv6 addresses from 2620:0:2d0:200:0:0:0:0 to 2620:0:2d0:200:ffff:ffff:ffff:ffff, enter **2620:0:2d0:200::/64**.

7. Review the settings for the IP set, and choose **Create IP set**.

Deleting an IP set

Follow the guidance in this section to delete a referenced set.

Deleting referenced sets and rule groups

When you delete an entity that you can use in a protection pack (web ACL), like an IP set, regex pattern set, or rule group, AWS WAF checks to see if the entity is currently being used in a protection pack (web ACL). If it finds that it is in use, AWS WAF warns you. AWS WAF is almost always able to determine if an entity is being referenced by a protection pack (web ACL). However, in rare cases it might not be able to do so. If you need to be sure that nothing is currently using the entity, check for it in your protection packs (web ACLs) before deleting it. If the entity is a referenced set, also check that no rule groups are using it.

To delete an IP set

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **IP sets**.
3. Select the IP set that you want to delete and choose **Delete**.

Creating and managing a regex pattern set in AWS WAF

A regex pattern set provides a collection of regular expressions that you want to use together in a rule statement. Regex pattern sets are AWS resources.

To use a regex pattern set in a protection pack (web ACL) or rule group, you first create an AWS resource, `RegexPatternSet` with your regex pattern specifications. Then you reference the set when you add a regex pattern set rule statement to a protection pack (web ACL) or rule group. A regex pattern set must contain at least one regex pattern.

If your regex pattern set contains more than one regex pattern, when it's used in a rule, the pattern matching is combined with `OR` logic. That is, a web request will match the pattern set rule statement if the request component matches any of the patterns in the set.

AWS WAF supports the pattern syntax used by the PCRE library `libpcre` with some exceptions. The library is documented at PCRE - Perl Compatible Regular Expressions. For information about AWS WAF support, see Supported regular expression syntax in AWS WAF.

Creating a regex pattern set

Follow the procedure in this section to create a new regex pattern set.

To create a regex pattern set

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **Regex pattern sets** and then **Create regex pattern set**.
3. Enter a name and description for the regex pattern set. You'll use these to identify it when you want to use the set.

   > **Note**
   >
   > You can't change the name after you create the regex pattern set.

4. For **Region**, choose Global (CloudFront) or choose the Region where you want to store the regex pattern set. You can use regional regex pattern sets only in protection packs (web ACLs) that protect regional resources. To use a regex pattern set in protection packs (web ACLs) that protect Amazon CloudFront distributions, you must use Global (CloudFront).
5. In the **Regular expressions** text box, enter one regex pattern per line.

   For example, the regular expression `I[a@]mAB[a@]dRequest` matches the following strings: `IamABadRequest`, `IamAB@dRequest`, `I@mABadRequest`, and `I@mAB@dRequest`.

   AWS WAF supports the pattern syntax used by the PCRE library `libpcre` with some exceptions. The library is documented at PCRE - Perl Compatible Regular Expressions. For information about AWS WAF support, see Supported regular expression syntax in AWS WAF.
6. Review the settings for the regex pattern set, and choose **Create regex pattern set**.

Deleting a regex pattern set

Follow the guidance in this section to delete a referenced set.

Deleting referenced sets and rule groups

When you delete an entity that you can use in a protection pack (web ACL), like an IP set, regex pattern set, or rule group, AWS WAF checks to see if the entity is currently being used in a

protection pack (web ACL). If it finds that it is in use, AWS WAF warns you. AWS WAF is almost always able to determine if an entity is being referenced by a protection pack (web ACL). However, in rare cases it might not be able to do so. If you need to be sure that nothing is currently using the entity, check for it in your protection packs (web ACLs) before deleting it. If the entity is a referenced set, also check that no rule groups are using it.

To delete a regex pattern set

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **Regex pattern sets**.
3. Select the regex pattern set that you want to delete and choose **Delete**.

Customized web requests and responses in AWS WAF

This section explains how to add custom web request and response handling behavior to your AWS WAF rule actions and default protection pack (web ACL) actions. Your custom settings apply whenever the action they're attached to applies.

You can customize web requests and responses in the following ways:

- With Allow, Count, CAPTCHA, and Challenge actions, you can insert custom headers into the web request. When AWS WAF forwards the web request to the protected resource, the request contains the entire original request plus the custom headers that you've inserted. For the CAPTCHA and Challenge actions, AWS WAF only applies the customization if the request passes the CAPTCHA or challenge token inspection.

- With Block actions, you can define a complete custom response, with response code, headers, and body. The protected resource responds to the request using the custom response provided by AWS WAF. Your custom response replaces the default Block action response of `403 (Forbidden)`.

Action settings that you can customize

You can specify a custom request or response when you define the following action settings:

- Rule action. For information, see Using rule actions in AWS WAF.

- Default action for a protection pack (web ACL). For information, see Setting the protection pack (web ACL) default action in AWS WAF.

Action settings that you cannot customize

You *cannot* specify custom request handling in the override action for a rule group that you use in a protection pack (web ACL). See Using protection packs (web ACLs) with rules and rule groups in AWS WAF. Also see Using managed rule group statements in AWS WAF and Using rule group statements in AWS WAF.

Temporary inconsistencies during updates

When you create or change a protection pack (web ACL) or other AWS WAF resources, the changes take a small amount of time to propagate to all areas where the resources are stored. The propagation time can be from a few seconds to a number of minutes.

The following are examples of the temporary inconsistencies that you might notice during change propagation:

- After you create a protection pack (web ACL), if you try to associate it with a resource, you might get an exception indicating that the protection pack (web ACL) is unavailable.
- After you add a rule group to a protection pack (web ACL), the new rule group rules might be in effect in one area where the protection pack (web ACL) is used and not in another.
- After you change a rule action setting, you might see the old action in some places and the new action in others.
- After you add an IP address to an IP set that is in use in a blocking rule, the new address might be blocked in one area while still allowed in another.

Limits on your use of custom requests and responses

AWS WAF defines maximum settings for your use of custom requests and responses. For example, a maximum number of request headers per protection pack (web ACL) or rule group, and a maximum number of custom headers for a single custom response definition. For information, see AWS WAF quotas.

Topics

- Inserting custom request headers for non-blocking actions
- Dynamic label interpolation

- Sending custom responses for Block actions
- Supported status codes for custom responses

Inserting custom request headers for non-blocking actions

This section explains how to instruct AWS WAF to insert custom headers into the original HTTP request when a rule action doesn't block the request. With this option, you only add to the request. You can't modify or replace any part of the original request. Use cases for custom header insertion include signaling a downstream application to process the request differently based on the inserted headers, and flagging the request for analysis.

> **Important**
>
> This option applies to the rule actions Allow, Count, CAPTCHA, and Challenge and to protection pack (web ACL) default actions that are set to Allow. For more information about rule actions, see Using rule actions in AWS WAF. For more information about default protection pack (web ACL) actions, see Setting the protection pack (web ACL) default action in AWS WAF.

Considerations when using custom request header names

Prefixes added to request headers

AWS WAF prefixes all request headers that it inserts with `x-amzn-waf-`, to avoid confusion with the headers that are already in the request. For example, if you specify the header name `sample`, AWS WAF inserts the header `x-amzn-waf-sample`.

> **Important**
>
> As a security practice, you can add a string match rule that blocks requests where the header already starts with `x-amzn-waf-`. This blocks requests from non-AWS WAF sources that mimic the `x-amzn-waf-` prefix string that is inserted by AWS WAF when processing custom request headers.

The following example shows a string match rule configured to block traffic where the `x-amzn-waf-` prefix was not inserted by AWS WAF:

```
    "Rules": [
        {
          "Name": "CustomHeader",
          "Priority": 0,
          "Statement": {
            "ByteMatchStatement": {
              "SearchString": "x-amzn-waf-",
              "FieldToMatch": {
                "Headers": {
                  "MatchPattern": {
                    "All": {}
                  },
                  "MatchScope": "KEY",
                  "OversizeHandling": "MATCH"
                }
              },
              "TextTransformations": [
                {
                  "Priority": 0,
                  "Type": "NONE"
                }
              ],
              "PositionalConstraint": "STARTS_WITH"
            }
          },
          "Action": {
            "Block": {}
          },
          "VisibilityConfig": {
            "SampledRequestsEnabled": true,
            "CloudWatchMetricsEnabled": true,
            "MetricName": "CustomHeader"
          }
        }
      ]
```

For information on using string match rules, see String match rule statement.

Headers with the same name

If the request already has a header with the same name that AWS WAF is inserting, AWS WAF overwrites the header. So, if you define headers in multiple rules with identical names, the last rule

to inspect the request and find a match would have its header added, and any previous rules would not.

Using custom headers with non-terminating rule actions

Unlike the Allow action, the Count action doesn't stop AWS WAF from processing the web request using the rest of the rules in the protection pack (web ACL). Similarly, when CAPTCHA and Challenge determine that the request token is valid, these actions don't stop AWS WAF from processing the web request. So, if you insert custom headers using a rule with one of these actions, subsequent rules might also insert custom headers. For more information about rule action behavior, see Using rule actions in AWS WAF.

For example, suppose you have the following rules, prioritized in the order shown:

1. RuleA with a Count action and a customized header named `RuleAHeader`.
2. RuleB with an Allow action and a customized header named `RuleBHeader`.

If a request matches both RuleA and RuleB, AWS WAF inserts the headers `x-amzn-waf-RuleAHeader` and `x-amzn-waf-RuleBHeader`, and then forwards the request to the protected resource.

AWS WAF inserts custom headers into a web request when it finishes inspecting the request. So if you use custom request handling with a rule that has the action set to Count, the custom headers that you add are not inspected by subsequent rules.

Custom request handling example

You define custom request handling for a rule's action or for a protection pack (web ACL)'s default action. The following listing shows the JSON for custom handling added to the default action for a protection pack (web ACL).

```
{
 "Name": "SampleWebACL",
 "Scope": "REGIONAL",
 "DefaultAction": {
  "Allow": {
   "CustomRequestHandling": {
    "InsertHeaders": [
     {
```

```
      "Name": "fruit",
      "Value": "watermelon"
     },
     {
      "Name": "pie",
      "Value": "apple"
     }
    ]
   }
  }
 },
 "Description": "Sample protection pack (web ACL) with custom request handling
 configured for default action.",
 "Rules": [],
 "VisibilityConfig": {
  "SampledRequestsEnabled": true,
  "CloudWatchMetricsEnabled": true,
  "MetricName": "SampleWebACL"
 }
}
```

Dynamic label interpolation

Dynamic label interpolation lets you embed label values directly in custom request headers, custom response headers, and custom response bodies using `${namespace:}` syntax. AWS WAF resolves each placeholder at evaluation time against the labels attached to the request, so you don't have to write a separate rule for each label value.

Interpolation works within the existing AWS WAF API with no new fields or configuration steps. You use the placeholder syntax in your existing string values. Existing static header values continue to work unchanged. Interpolation only activates when a value contains `${namespace:}` clauses.

Where interpolation is supported

You can use `${namespace:}` interpolation in the following locations:

- **Custom request headers** – Insert resolved label values into headers forwarded to your origin. Use the `${namespace:}` syntax in the header value field.
- **Custom response bodies** – Embed label values and synthetic labels in block pages, challenge pages, and other custom responses. Use the `${namespace:}` syntax in the response body content field.

- **Custom response headers** – Insert label values into response headers, for example in a `Location` header for redirects. Use the `${namespace:}` syntax in the response header value field.

Interpolation syntax and resolution

To use interpolation, include `${namespace:}` clauses in your header values or custom response body content. The trailing colon is significant. It tells AWS WAF to resolve all labels within that namespace rather than match a single label literally.

AWS WAF resolves each clause at evaluation time using the following rules:

1. **Single label match** – The clause resolves to the label's terminal value. For example, if the request has the label `awswaf:managed:aws:bot-control:bot:category:scraping`, the clause `${awswaf:managed:aws:bot-control:bot:category:}` resolves to `scraping`.
2. **Multiple labels** – When multiple labels match the namespace, values are returned as a comma-separated list with the namespace prefix stripped. For example, `scraping,advertising`.
3. **No match** – The clause resolves to an empty string.

> **Note**
>
> A single string value supports up to 10 `${namespace:}` placeholders. If a value contains more than 10 placeholders, AWS WAF resolves the first 10 and leaves any additional placeholders in the output as literal `${namespace:}` text. This limit applies for each string value, not for each request. You can use up to 10 placeholders in each of multiple headers without issue.

> **Important**
>
> Custom labels use a short name in `ruleLabels`. For example, `app:tier:enterprise`. However, AWS WAF automatically prefixes the short name with the protection pack (web ACL) context, producing a fully-qualified label. For example, `awswaf:`*`ACCOUNT_ID`*`:webacl:`*`WEBACL_NAME`*`:app:tier:enterprise`. The label match statement works with the short namespace, but interpolation references must always use the fully-qualified namespace.

Synthetic labels

Interpolation also supports synthetic labels. These are built-in values that AWS WAF resolves from the request context rather than from the label store. You can combine synthetic labels with namespace-based labels in the same value string.

Synthetic label	Description
`${awswaf:request_id:}`	The unique AWS WAF request identifier.
`${awswaf:ip:}`	The client IP address.
`${awswaf:ja3:}`	The JA3 TLS fingerprint.
`${awswaf:ja4:}`	The JA4 TLS fingerprint.

Dynamic label interpolation examples

Application signaling with dynamic headers

The following rule forwards several Bot Control signal namespaces to your origin as separate headers. One rule covers the whole namespace, so you don't need to update the rule when the managed rule group adds new labels.

```
{
  "Name": "forward-waf-signals",
  "Statement": {
    "LabelMatchStatement": {
      "Scope": "NAMESPACE",
      "Key": "awswaf:managed:aws:bot-control:bot:category:"
    }
  },
  "RuleAction": {
    "Count": {
      "CustomRequestHandling": {
        "InsertHeaders": [
          { "Name": "bot-category",
            "Value": "${awswaf:managed:aws:bot-control:bot:category:}" },
          { "Name": "bot-name",
            "Value": "${awswaf:managed:aws:bot-control:bot:name:}" },
```

```
          { "Name": "bot-signals",
            "Value": "${awswaf:managed:aws:bot-control:signal:}" },
          { "Name": "client-ip",
            "Value": "${awswaf:ip:}" }
        ]
      }
    }
  }
}
```

The `bot-signals` header (output: `x-amzn-waf-bot-signals`) demonstrates multi-value resolution. The `signal:` namespace may contain multiple labels, for example `non_browser_user_agent,automated_browser`, which resolve to a comma-separated list. The `client-ip` header (output: `x-amzn-waf-client-ip`) uses a synthetic label.

Custom block page with debug information

Synthetic labels let you build block pages that include request-specific context. Embed the client's IP address and the AWS WAF request ID directly in the response body to give users something to share when they report a false positive.

```
{
  "CustomResponseBodies": {
    "BlockPage": {
      "Content": "Your request was blocked.\n\nIP: ${awswaf:ip:}\nRequest ID:
 ${awswaf:request_id:}\n\nIf you believe this is an error, contact support with the
 Request ID above.",
      "ContentType": "TEXT_PLAIN"
    }
  }
}
```

Custom label interpolation

Interpolation works with any label namespace, including custom labels that you define in your own rules. The following example classifies requests by API key and forwards the tier value to the origin.

Rule 1: Classify tier based on API key

This rule matches requests with an enterprise API key and applies a custom label.

```
{
```

```
{
  "Name": "classify-tier",
  "Priority": 100,
  "Statement": {
    "ByteMatchStatement": {
      "SearchString": "pk_enterprise_",
      "FieldToMatch": { "SingleHeader": { "Name": "x-api-key" } },
      "PositionalConstraint": "STARTS_WITH",
      "TextTransformations": [{ "Priority": 0, "Type": "NONE" }]
    }
  },
  "RuleLabels": [{ "Name": "app:tier:enterprise" }],
  "Action": { "Count": {} }
}

}
```

Rule 2: Forward the resolved tier value

This rule matches any label in the `app:tier` namespace and forwards the resolved value as a header.

```
{
{
  "Name": "forward-tier",
  "Priority": 200,
  "Statement": {
    "LabelMatchStatement": {
      "Scope": "NAMESPACE",
      "Key": "app:tier:"
    }
  },
  "Action": {
    "Count": {
      "CustomRequestHandling": {
        "InsertHeaders": [{
          "Name": "customer-tier",
          "Value": "${awswaf:ACCOUNT_ID:webacl:WEBACL_NAME:app:tier:}"
        }]
      }
    }
  }
}
```

The first rule applies the label `app:tier:enterprise`. The second rule matches any label in the `app:tier` namespace and forwards the resolved value as the `customer-tier` header. The output header name is `x-amzn-waf-customer-tier`. You can add more classification rules for other tiers, such as `app:tier:standard` or `app:tier:trial`. The forwarding rule picks them up without any changes.

For a deployable sample that demonstrates all of these patterns, see the AWS WAF Dynamic Label Interpolation sample on GitHub.

Sending custom responses for Block actions

This section explains how to instruct AWS WAF to send a custom HTTP response back to the client for rule actions or protection pack (web ACL) default actions that are set to Block. For more information about rule actions, see Using rule actions in AWS WAF. For more information about default protection pack (web ACL) actions, see Setting the protection pack (web ACL) default action in AWS WAF.

When you define custom response handling for a Block action, you define the status code, headers, and response body. For a list of status codes that you can use with AWS WAF, see the section that follows, Supported status codes for custom responses.

Use cases

The use cases for custom responses include the following:

- Sending a non-default status code back to the client.
- Sending custom response headers back to the client. You can specify any header name except for `content-type`.
- Sending a static error page back to the client.
- Redirecting the client to a different URL. To do this, you specify one of the `3xx` redirection status codes, like `301 (Moved Permanently)` or `302 (Found)`, and then specify a new header named `Location` with the new URL.

Interaction with responses that you define in your protected resource

Custom responses that you specify for the AWS WAF Block action take precedence over any response specifications that you define in your protected resource.

The host service for the AWS resource that you protect with AWS WAF might permit custom response handling for web requests. Examples include the following:

- With Amazon CloudFront, you can customize the error page based on status code. For information, see Generating custom error responses in the *Amazon CloudFront Developer Guide*.
- With Amazon API Gateway you can define the response and status code for your gateway. For information, see Gateway responses in API Gateway in the *Amazon API Gateway Developer Guide*.

You can't combine AWS WAF custom response settings with custom response settings in the protected AWS resource. The response specification for any individual web request comes either completely from AWS WAF or completely from the protected resource.

For web requests that AWS WAF blocks, the following shows the order of precedence.

1. **AWS WAF custom response** – If the AWS WAF Block action has a custom response enabled, the protected resource sends the configured custom response back to the client. Any response settings that you might have defined in the protected resource itself have no effect.
2. **Custom response defined in the protected resource** – Otherwise, if the protected resource has custom response settings specified, the protected resource uses those settings to respond to the client.
3. **AWS WAF default Block response** – Otherwise, the protected resource responds to the client with the AWS WAF default Block response `403 (Forbidden)`.

For web requests that AWS WAF allows, your configuration of the protected resource determines the response that it sends back to the client. You can't configure response settings in AWS WAF for allowed requests. The only customization that you can configure in AWS WAF for allowed requests is the insertion of custom headers into the original request, before forwarding the request to the protected resource. This option is described in the preceding section, Inserting custom request headers for non-blocking actions.

Custom response headers

You can specify any header name except for `content-type`.

Custom response bodies

You define the body of a custom response within the context of the protection pack (web ACL) or rule group where you want to use it. After you've defined a custom response body, you can use it

by reference anywhere else in the protection pack (web ACL) or rule group where you created it. In the individual Block action settings, you reference the custom body that you want to use and you define the status code and header of the custom response.

When you create a custom response in the console, you can choose from response bodies that you've already defined or you can create a new body. Outside of the console, you define your custom response bodies at the protection pack (web ACL) or rule group level, and then reference them from the action settings within the protection pack (web ACL) or rule group. This is shown in the example JSON in the following section.

Custom response example

The following example lists the JSON for a rule group with custom response settings. The custom response body is defined for the entire rule group, then referenced by key in the rule action.

```
{
 "ARN": "test_rulegroup_arn",
 "Capacity": 1,

 "CustomResponseBodies": {
  "CustomResponseBodyKey1": {
   "Content": "This is a plain text response body.",
   "ContentType": "TEXT_PLAIN"
  }
 },

 "Description": "This is a test rule group.",
 "Id": "test_rulegroup_id",
 "Name": "TestRuleGroup",

 "Rules": [
  {
   "Action": {
    "Block": {
     "CustomResponse": {
      "CustomResponseBodyKey": "CustomResponseBodyKey1",
      "ResponseCode": 404,
      "ResponseHeaders": [
       {
        "Name": "BlockActionHeader1Name",
        "Value": "BlockActionHeader1Value"
       }
```

```
      ]
     }
    }
   },
   "Name": "GeoMatchRule",
   "Priority": 1,
   "Statement": {
    "GeoMatchStatement": {
     "CountryCodes": [
      "US"
     ]
    }
   },
   "VisibilityConfig": {
    "CloudWatchMetricsEnabled": true,
    "MetricName": "TestRuleGroupReferenceMetric",
    "SampledRequestsEnabled": true
   }
  }
 ],
 "VisibilityConfig": {
  "CloudWatchMetricsEnabled": true,
  "MetricName": "TestRuleGroupMetric",
  "SampledRequestsEnabled": true
 }
}
```

Supported status codes for custom responses

This section lists the status codes that you can use in a custom response. For detailed information about HTTP status codes, see Status Codes by the Internet Engineering Task Force (IETF) and List of HTTP status codes on Wikipedia.

The following are the HTTP status codes that AWS WAF supports for custom responses.

- 2xx Successful
  - 200 – OK
  - 201 – Created
  - 202 – Accepted
  - 204 – No Content
  - 206 – Partial Content

- 3xx Redirection
  - 300 – Multiple Choices
  - 301 – Moved Permanently
  - 302 – Found
  - 303 –See Other
  - 304 – Not Modified
  - 307 – Temporary Redirect
  - 308 – Permanent Redirect
- 4xx Client Error
  - 400 – Bad Request
  - 401 – Unauthorized
  - 403 – Forbidden
  - 404 – Not Found
  - 405 – Method Not Allowed
  - 408 – Request Timeout
  - 409 – Conflict
  - 411 – Length Required
  - 412 – Precondition Failed
  - 413 – Request Entity Too Large
  - 414 – Request-URI Too Long
  - 415 – Unsupported Media Type
  - 416 – Requested Range Not Satisfiable
  - 421 – Misdirected Request
  - 429 – Too Many Requests
- **402 (Payment Required)** – Used by the Monetize action to return a price manifest to AI agents. The response body contains machine-readable pricing, accepted payment methods, and license terms in JSON format.
- 5xx Server Error
  - 500 – Internal Server Error
  - 501 – Not Implemented
  - 502 – Bad Gateway

- `503 – Service Unavailable`
- `504 – Gateway Timeout`
- `505 – HTTP Version Not Supported`

Web request labeling in AWS WAF

This section explains what AWS WAF labels are.

A label is metadata added to a web request by a rule when the rule matches the request. Once added, a label remains available on the request until the protection pack (web ACL) evaluation ends. You can access labels in rules that run later in the protection pack (web ACL) evaluation by using a label match statement. For details, see Label match rule statement.

Labels on web requests generate Amazon CloudWatch label metrics. For a list of metrics and dimensions, see Label metrics and dimensions. For information about accessing metrics and metric summaries through CloudWatch and through the AWS WAF console, see Monitoring and tuning your AWS WAF protections.

Labeling use cases

Common use cases for AWS WAF labels include the following:

- **Evaluating a web request against multiple rule statements before taking action on the request** – After a match is found with a rule in a protection pack (web ACL), AWS WAF continues evaluating the request against the protection pack (web ACL) if the rule action doesn't terminate the protection pack (web ACL) evaluation. You can use labels to evaluate and collect information from multiple rules before you decide to allow or block the request. To do this, change the actions for your existing rules to Count and configure them to add labels to matching requests. Then, add one or more new rules to run after your other rules, and configure them to evaluate the labels and manage the requests according to the label match combinations.
- **Managing web requests by geographical region** – You can use the geographic match rule alone to manage web requests by the country of origin. To fine-tune the location down to the region level, you use the geo match rule with a Count action followed by a label match rule. For information about the geo match rule, see Geographic match rule statement.
- **Reusing logic across multiple rules** – If you need to reuse the same logic across multiple rules, you can use labels to single-source the logic and just test for the results. When you have multiple complex rules that use a common subset of nested rule statements, duplicating the common

rule set across your complex rules can be time consuming and error prone. With labels, you can create a new rule with the common rule subset that counts matching requests and adds a label to them. You add the new rule to your protection pack (web ACL) so that it runs before your original complex rules. Then, in your original rules, you replace the shared rule subset with a single rule that checks for the label.

For example, say you have multiple rules that you want to only apply to your login paths. Rather than have each rule specify the same logic to match potential login paths, you can implement a single new rule that contains that logic. Have the new rule add a label to matching requests to indicate that the request is on a login path. In your protection pack (web ACL), give this new rule a lower numeric priority setting than your original rules so that it runs first. Then, in your original rules, replace the shared logic with a check for the presence of the label. For information about priority settings, see Setting rule priority.

- **Creating exceptions to rules in rule groups** – This option is particularly useful for managed rule groups, which you can't view or alter. Many managed rule group rules add labels to matching web requests, to indicate the rules that matched and possibly to provide additional information about the match. When you use a rule group that adds labels to requests, you can override the rule group rules to count matches, and then run a rule after the rule group that handles the web request based on the rule group labels. All AWS Managed Rules add labels to matching web requests. For details, see the rule descriptions at AWS Managed Rules rule groups list.
- **Using label metrics to monitor traffic patterns** – You can access metrics for labels that you add through your rules and for metrics added by any managed rule groups that you use in your protection pack (web ACL). All of the AWS Managed Rules rule groups add labels to the web requests that they evaluate. For a list of label metrics and dimensions, see Label metrics and dimensions. You can access metrics and metric summaries through CloudWatch and through the protection pack (web ACL) page in the AWS WAF console. For information, see Monitoring and tuning your AWS WAF protections.
- **Dynamic label interpolation** – You can use `${namespace:}` syntax to resolve label values at evaluation time in custom request headers, custom response headers, and custom response bodies. One rule with interpolation covers an entire namespace, so you don't need a separate rule for each label value. This is useful for forwarding managed rule group labels, such as Bot Control categories, to your origin as request headers. For more information, see Dynamic label interpolation.

How labeling works in AWS WAF

This section explains how AWS WAF labels work.

When a rule matches a web request, if the rule has labels defined, AWS WAF adds the labels to the request at the end of the rule evaluation. Rules that are evaluated after the matching rule in the protection pack (web ACL) can match against the labels that the rule has added.

Who adds labels to requests

The protection pack (web ACL) components that evaluate requests can add labels to the requests.

- Any rule that isn't a rule group reference statement can add labels to matching web requests. The labeling criteria is part of the rule definition, and when a web request matches the rule, AWS WAF adds the rule's labels to the request. For information, see the section called “Rules that add labels”.
- The geo match rule statement adds country and region labels to any request that it inspects, regardless of whether the statement results in a match. For information, see the section called “Geographic match”.
- The AWS Managed Rules for AWS WAF all add labels to requests that they inspect. They add some labels based on rule matches in the rule group and they add some based on AWS processes that the managed rule groups use, such as the token labeling added when you use an intelligent threat mitigation rule group. For information about the labels that each managed rule group adds, see the section called “AWS Managed Rules rule groups list”.

How AWS WAF manages labels

AWS WAF adds the rule's labels to the request at the end of the rule's inspection of the request. Labeling is part of a rule's match activities, similar to the action.

Labels don't persist with the web request after the protection pack (web ACL) evaluation ends. In order for other rules to match against a label that your rule adds, your rule action must not terminate the evaluation of the web request by the protection pack (web ACL). The rule action must be set to Count, CAPTCHA, or Challenge. When the protection pack (web ACL) evaluation doesn't terminate, subsequent rules in the protection pack (web ACL) can run their label matching criteria against the request. For more information about rule actions, see Using rule actions in AWS WAF.

Access to labels during protection pack (web ACL) evaluation

Once added, labels remain available on the request as long as AWS WAF is evaluating the request against the protection pack (web ACL). Any rule in a protection pack (web ACL) can access labels that have been added by the rules that have already run in the same protection pack (web ACL). This includes rules that are defined directly inside the protection pack (web ACL) and rules defined inside rule groups that are used in the protection pack (web ACL).

- You can match against a label in your rule's request inspection criteria using the label match statement. You can match against any label that's attached to the request. For statement details, see Label match rule statement.
- The geographic match statement adds labels with or without a match, but they're only available after the statement's containing protection pack (web ACL) rule has completed the request evaluation.
  - You can't use a single rule, for example a logical `AND` statement, to run a geo match statement followed by a label match statement against the geographic labels. You must put the label match statement in a separate rule that runs after the rule that contains the geo match statement.
  - If you use a geo match statement as a scope-down statement inside a rate-based rule statement or managed rule group reference statement, the labels that the geo match statement adds are not available for inspection by the containing rule's statement. If you need to inspect geographic labeling in a rate-based rule statement or a rule group, you must run the geo match statement in a separate rule that runs beforehand.

Access to label information outside of protection pack (web ACL) evaluation

Labels don't persist with the web request after the protection pack (web ACL) evaluation ends, but AWS WAF records label information in the logs and in metrics.

- AWS WAF stores Amazon CloudWatch metrics for the first 100 labels on any single request. For information about accessing label metrics, see Monitoring with Amazon CloudWatch and Label metrics and dimensions.
- AWS WAF summarizes CloudWatch label metrics in the protection pack (web ACL) traffic overview dashboards in the AWS WAF console. You can access the dashboards on any protection pack (web ACL) page. For more information, see Traffic overview dashboards for protection packs (web ACLs).

- AWS WAF records labels in the logs for the first 100 labels on a request. You can use labels, along with the rule action, to filter the logs that AWS WAF records. For information, see Logging AWS WAF protection pack (web ACL) traffic.

Your protection pack (web ACL) evaluation can apply more than 100 labels to a web request and match against more than 100 labels, but AWS WAF only records the first 100 in the logs and metrics.

Label syntax and naming requirements in AWS WAF

This section explains how to construct and match against an AWS WAF label.

A label is a string made up of a prefix, optional namespaces, and a name. The components of a label are delimited with a colon. Labels have the following requirements and characteristics:

- Labels are case-sensitive.
- Each label namespace or label name can have up to 128 characters.
- You can specify up to five namespaces in a label.
- Components of a label are separated by colon (`:`).
- You can't use the following reserved strings in the namespaces or name that you specify for a label: `awswaf`, `aws`, `waf`, `rulegroup`, `webacl`, `regexpatternset`, `ipset`, and `managed`.

Label syntax

A fully qualified label has a prefix, optional namespaces, and label name. The prefix identifies the rule group or protection pack (web ACL) context of the rule that added the label. Namespaces might be used to add more context for the label. The label name provides the lowest level of detail for a label. It often indicates the specific rule that added the label to the request.

The label prefix varies depending on its origin.

- **Your labels** – The following shows the full label syntax for labels that you create in your protection pack (web ACL) and rule group rules. The entity types are `rulegroup` and `webacl`.

```
awswaf:<entity owner account id>:<entity type>:<entity name>:<custom
 namespace>:...:<label name>
```

- Label namespace prefix: `awswaf:<entity owner account id>:<entity type>:<entity name>:`
- Custom namespace additions: `<custom namespace>:...:`

When you define a label for a rule in a rule group or protection pack (web ACL), you control the custom namespace strings and the label name. The rest is generated for you by AWS WAF. AWS WAF automatically prefixes all labels with `awswaf` and the account and protection pack (web ACL) or rule group entity settings.

- **Managed rule group labels** – The following shows the full label syntax for labels that are created by rules in managed rule groups.

```
awswaf:managed:<vendor>:<rule group name>:<custom namespace>:...:<label name>
```

  - Label namespace prefix: `awswaf:managed:<vendor>:<rule group name>:`
  - Custom namespace additions: `<custom namespace>:...:`

  All AWS Managed Rules rule groups add labels. For information about managed rule groups, see Using managed rule groups in AWS WAF.

- **Labels from other AWS processes** – These processes are used by AWS Managed Rules rule groups, so you see them added to web requests that you evaluate using managed rule groups. The following shows the full label syntax for labels that are created by processes that are called by managed rule groups.

```
awswaf:managed:<process>:<custom namespace>:...:<label name>
```

  - Label namespace prefix: `awswaf:managed:<process>:`
  - Custom namespace additions: `<custom namespace>:...:`

  Labels of this type are listed for the managed rule groups that call the AWS process. For information about managed rule groups, see Using managed rule groups in AWS WAF.

Label examples for your rules

The following example labels are defined by rules in a rule group named `testRules` that belongs to the account, 111122223333.

```
awswaf:111122223333:rulegroup:testRules:testNS1:testNS2:LabelNameA
```

```
awswaf:111122223333:rulegroup:testRules:testNS1:LabelNameQ
```

```
awswaf:111122223333:rulegroup:testRules:LabelNameZ
```

The following listing shows an example label specification in JSON. These label names include custom namespace strings before the ending label name.

```
Rule: {
    Name: "label_rule",
    Statement: {...}
    RuleLabels: [
        Name: "header:encoding:utf8",
        Name: "header:user_agent:firefox"
    ],
    Action: { Count: {} }
}
```

Note

You can access this type of listing in the console through the rule JSON editor.

If you run the preceding rule in the same rule group and account as the preceding label examples, the resulting, fully qualified labels would be the following:

```
awswaf:111122223333:rulegroup:testRules:header:encoding:utf8
```

```
awswaf:111122223333:rulegroup:testRules:header:user_agent:firefox
```

Label examples for managed rule groups

The following show example labels from AWS Managed Rules rule groups and processes that they invoke.

```
awswaf:managed:aws:core-rule-set:NoUserAgent_Header
```

```
awswaf:managed:aws:sql-database:SQLiExtendedPatterns_QueryArguments
```

```
awswaf:managed:aws:atp:aggregate:attribute:compromised_credentials
```

```
awswaf:managed:token:accepted
```

AWS WAF rules that add labels

In almost all rules, you can define labels and AWS WAF will apply them to any matching request.

The following rule types are the only exceptions:

- **Rate-based rules label only while rate limiting** – Rate-based rules only add labels to web requests for a specific aggregation instance while that instance is being rate limited by AWS WAF. For information about rate-based rules, see Using rate-based rule statements in AWS WAF.
- **Labeling isn't allowed in rule group reference statements** – The console doesn't accept labels for rule group statements or managed rule group statements. Through the API, specifying a label for either statement type results in a validation exception. For information about these statement types, see Using managed rule group statements in AWS WAF and Using rule group statements in AWS WAF.

WCUs – 1 WCU for every 5 labels that you define in your protection pack (web ACL) or rule group rules.

Where to find this

- **Rule builder** on the console – Under the rule's **Action** settings, under **Label**.
- **API data type** – `Rule RuleLabels`

You define a label in a rule by specifying the custom namespace strings and name to append to the label namespace prefix. AWS WAF derives the prefix from the context in which you define the rule. For information about this, see the label syntax information under Label syntax and naming requirements in AWS WAF.

AWS WAF rules that match labels

This section explains how to use a label match statement to evaluate web request labels. You can match against *Label*, which requires the label name, or against *Namespace*, which requires a namespace specification. For either label or namespace, you can optionally include preceding namespaces and the prefix in your specification. For general information about this statement type, see Label match rule statement.

A label's prefix defines the context of the rule group or protection pack (web ACL) where the label's rule is defined. In a rule's label match statement, if your label or namespace match string doesn't specify the prefix, AWS WAF uses the prefix for the label match rule.

- Labels for rules that are defined directly inside a protection pack (web ACL) have a prefix that specifies the protection pack (web ACL) context.
- Labels for rules that are inside a rule group have a prefix that specifies the rule group context. This could be your own rule group or a rule group that's managed for you.

For information about this, see label syntax under Label syntax and naming requirements in AWS WAF.

> **Note**
>
> Some managed rule groups add labels. You can retrieve these through the API by calling `DescribeManagedRuleGroup`. The labels are listed in the `AvailableLabels` property in the response.

If you want to match against a rule that's in a different context than the context of your rule, you must provide the prefix in your match string. For example, if you want to match against labels that are added by rules in a managed rule group, you could add a rule in your protection pack (web ACL) with a label match statement whose match string specifies the rule group's prefix followed by your additional match criteria.

In the match string for the label match statement, you specify either a label or a namespace:

- **Label** – The label specification for a match consists of the ending part of the label. You can include any number of the contiguous namespaces that immediately precede the label name

followed by the name. You can also provide the fully qualified label by starting the specification with the prefix.

Example specifications:

- `testNS1:testNS2:LabelNameA`
- `awswaf:managed:aws:managed-rule-set:testNS1:testNS2:LabelNameA`

- **Namespace** – The namespace specification for a match consists of any contiguous subset of the label specification excluding the name. You can include the prefix and you can include one or more namespace strings.

  Example specifications:

  - `testNS1:testNS2:`
  - `awswaf:managed:aws:managed-rule-set:testNS1:`

AWS WAF label match examples

This section provides examples of match specifications, for the label match rule statement.

> **Note**
>
> These JSON listings were created in the console by adding a rule to a protection pack (web ACL) with the label match specifications and then editing the rule and switching to the **Rule JSON editor**. You can also get the JSON for a rule group or protection pack (web ACL) through the APIs or the command line interface.

Topics

- Match against a local label
- Match against a label from another context
- Match against a managed rule group label
- Match against a local namespace
- Match against a managed rule group namespace

Match against a local label

The following JSON listing shows a label match statement for a label that's been added to the web request locally, in the same context as this rule.

```
Rule: {
    Name: "match_rule",
    Statement: {
        LabelMatchStatement: {
            Scope: "LABEL",
            Key: "header:encoding:utf8"
        }
    },
    RuleLabels: [
        ...generate_more_labels...
    ],
    Action: { Block: {} }
}
```

If you use this match statement in account 111122223333, in a rule that you define for protection pack (web ACL) `testWebACL`, it would match the following labels.

```
awswaf:111122223333:webacl:testWebACL:header:encoding:utf8
```

```
awswaf:111122223333:webacl:testWebACL:testNS1:testNS2:header:encoding:utf8
```

It wouldn't match the following label, because the label string isn't an exact match.

```
awswaf:111122223333:webacl:testWebACL:header:encoding2:utf8
```

It wouldn't match the following label, because the context isn't the same, so the prefix doesn't match. This is true even if you added the rule group `productionRules` to the protection pack (web ACL) `testWebACL`, where the rule is defined.

```
awswaf:111122223333:rulegroup:productionRules:header:encoding:utf8
```

Match against a label from another context

The following JSON listing shows a label match rule that matches against a label from a rule inside a user-created rule group. The prefix is required in the specification for all rules running

in the protection pack (web ACL) that aren't part of the named rule group. This example label specification matches only the exact label.

```
Rule: {
    Name: "match_rule",
    Statement: {
        LabelMatchStatement: {
            Scope: "LABEL",
            Key: "awswaf:111122223333:rulegroup:testRules:header:encoding:utf8"
        }
    },
    RuleLabels: [
        ...generate_more_labels...
    ],
    Action: { Block: {} }
}
```

Match against a managed rule group label

This is a special case of matching against a label that's from another context than that of the match rule. The following JSON listing shows a label match statement for a managed rule group label. This matches only the exact label that's specified in the label match statement's key setting.

```
Rule: {
    Name: "match_rule",
    Statement: {
        LabelMatchStatement: {
            Scope: "LABEL",
            Key: "awswaf:managed:aws:managed-rule-set:header:encoding:utf8"
        }
    },
    RuleLabels: [
        ...generate_more_labels...
    ],
    Action: { Block: {} }
}
```

Match against a local namespace

The following JSON listing shows a label match statement for a local namespace.

```
Rule: {
```

```
    Name: "match_rule",
    Statement: {
        LabelMatchStatement: {
            Scope: "NAMESPACE",
            Key: "header:encoding:"
        }
    },
    Labels: [
        ...generate_more_labels...
    ],
    Action: { Block: {} }
}
```

Similar to the local `Label` match, if you use this statement in account 111122223333, in a rule that you define for protection pack (web ACL) `testWebACL`, it would match the following label.

```
awswaf:111122223333:webacl:testWebACL:header:encoding:utf8
```

It wouldn't match the following label, because the account isn't the same, so the prefix doesn't match.

```
awswaf:444455556666:webacl:testWebACL:header:encoding:utf8
```

The prefix also doesn't match any labels applied by managed rule groups, like the following.

```
awswaf:managed:aws:managed-rule-set:header:encoding:utf8
```

Match against a managed rule group namespace

The following JSON listing shows a label match statement for a managed rule group namespace. For a rule group that you own, you'd also need to provide the prefix in order to match for a namespace that's outside of the rule's context.

```
Rule: {
    Name: "match_rule",
    Statement: {
        LabelMatchStatement: {
            Scope: "NAMESPACE",
            Key: "awswaf:managed:aws:managed-rule-set:header:"
        }
```

```
    },
    RuleLabels: [
        ...generate_more_labels...
    ],
    Action: { Block: {} }
}
```

This specification matches against the following example labels.

```
awswaf:managed:aws:managed-rule-set:header:encoding:utf8
```

```
awswaf:managed:aws:managed-rule-set:header:encoding:unicode
```

It doesn't match the following label.

```
awswaf:managed:aws:managed-rule-set:query:badstring
```

Intelligent threat mitigation in AWS WAF

This section covers the managed intelligent threat mitigation features provided by AWS WAF. These are advanced, specialized protections that you can implement to protect against threats such as malicious bots and account takeover attempts.

> **Note**
>
> The features described here incur additional costs, beyond the basic fees for using AWS WAF. For more information, see AWS WAF Pricing.

The guidance provided in this section is intended for users who know generally how to create and manage AWS WAF web ACLs, rules, and rule groups. Those topics are covered in prior sections of this guide.

Topics

- Options for intelligent threat mitigation in AWS WAF
- Best practices for intelligent threat mitigation in AWS WAF
- Token use in AWS WAF intelligent threat mitigation

- AWS WAF Fraud Control account creation fraud prevention (ACFP)
- AWS WAF Fraud Control account takeover prevention (ATP)
- AWS WAF Bot Control
- AWS WAF Distributed Denial of Service (DDoS) prevention
- Client application integrations in AWS WAF
- CAPTCHA and Challenge in AWS WAF

Options for intelligent threat mitigation in AWS WAF

This section provides a detailed comparison of the options for implementing intelligent threat mitigation.

AWS WAF offers the following types of protections for intelligent threat mitigation.

- **AWS WAF Fraud Control account creation fraud prevention (ACFP)** – Detects and manages malicious account creation attempts on your application's sign-up page. The core functionality is provided by the ACFP managed rule group. For more information, see AWS WAF Fraud Control account creation fraud prevention (ACFP) and AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group.
- **AWS WAF Fraud Control account takeover prevention (ATP)** – Detects and manages malicious takeover attempts on your application's login page. The core functionality is provided by the ATP managed rule group. For more information, see AWS WAF Fraud Control account takeover prevention (ATP) and AWS WAF Fraud Control account takeover prevention (ATP) rule group.
- **AWS WAF Bot Control** – Identifies, labels, and manages both friendly and malicious bots. This feature provides management for common bots with signatures that are unique across applications, and also for targeted bots that have signatures specific to an application. The core functionality is provided by the Bot Control managed rule group. For more information, see AWS WAF Bot Control and AWS WAF Bot Control rule group.
- **Client application integration SDKs** – Validate client sessions and end users on your web pages and acquire AWS WAF tokens for clients to use in their web requests. If you use ACFP, ATP, or Bot Control, implement the application integration SDKs in your client application if you can, to take full advantage of all of the rule group features. We only recommend using these rule groups without an SDK integration as a temporary measure, when a critical resource needs to be quickly secured and there isn't enough time for the SDK integration. For information about implementing the SDKs, see Client application integrations in AWS WAF.

- **Challenge and CAPTCHA rule actions** – Validate client sessions and end users and acquire AWS WAF tokens for clients to use in their web requests. You can implement these anywhere that you specify a rule action, in your rules and as overrides in rule groups that you use. These actions use AWS WAF JavaScript interstitials to interrogate the client or end user, and they require client applications that support JavaScript. For more information, see CAPTCHA and Challenge in AWS WAF.

The intelligent threat mitigation AWS Managed Rules rule groups ACFP, ATP, and Bot Control use tokens for advanced detection. For information about the features that tokens enable in the rule groups, see Using application integration SDKs with ACFP, Using application integration SDKs with ATP, and Using application integration SDKs with Bot Control.

Your options for implementing intelligent threat mitigation run from the basic use of rule actions to run challenges and enforce token acquisition, to the advanced features offered by the intelligent threat mitigation AWS Managed Rules rule groups.

The following tables provide detailed comparisons of the options for the basic and advanced features.

Topics

- Options for challenges and token acquisition
- Options for intelligent threat mitigation managed rule groups
- Options for rate limiting in rate-based rules and targeted Bot Control rules

Options for challenges and token acquisition

This section compares challenge and token management options.

You can provide challenges and acquire tokens using the AWS WAF application integration SDKs or the rule actions Challenge and CAPTCHA. Broadly speaking, the rule actions are easier to implement, but they incur added costs, intrude more on your customer experience, and require JavaScript. The SDKs require programming in your client applications, but they can provide a better customer experience, they're free to use, and they can be used with JavaScript or in Android or iOS applications. You can only use the application integration SDKs with protection packs (web ACLs) that use one of the paid intelligent threat mitigation managed rule groups, described in the following section.

Comparison of options for challenges and token acquisition

	Challenge rule action	CAPTCHA rule action	JavaScript SDK challenge	Mobile SDK challenge
What it is	Rule action that enforces acquisition of the AWS WAF token by presenting the browser client with a silent challenge interstitial	Rule action that enforces acquisition of the AWS WAF token by presenting the client end user with a visual or audio challenge interstitial	Application integration layer, for client browsers and other devices that execute JavaScript. Renders the silent challenge and acquires a token	Application integration layer, for Android and iOS applications. Natively renders the silent challenge and acquires a token
Good choice for...	Silent validation against bot sessions and enforcement of token acquisition for clients that support JavaScript	End user and silent validation against bot sessions and enforcement of token acquisition, for clients that support JavaScript	Silent validation against bot sessions and enforcement of token acquisition for clients that support JavaScript. The SDKs provide the lowest latency and best control over where the challenge script runs in the application.	Silent validation against bot sessions and enforcement of token acquisition for native mobile applications on Android and iOS. The SDKs provide the lowest latency and best control over where the challenge script runs in the application.

	Challenge rule action	CAPTCHA rule action	JavaScript SDK challenge	Mobile SDK challenge
Implementation considerations	Implemented as a rule action setting	Implemented as a rule action setting	Requires one of the ACFP, ATP, or Bot Control paid rule groups in the protection pack (web ACL). Requires coding in the client application.	Requires one of the ACFP, ATP, or Bot Control paid rule groups in the protection pack (web ACL). Requires coding in the client application.
Runtime considerations	Intrusive flow for requests without valid tokens. Client is redirected to an AWS WAF challenge interstitial. Adds network round trips and requires a second evaluation of the web request.	Intrusive flow for requests without valid tokens. Client is redirected to an AWS WAF CAPTCHA interstitial. Adds network round trips and requires a second evaluation of the web request.	Can be run behind the scenes. Gives you more control over the challenge experience.	Can be run behind the scenes. Gives you more control over the challenge experience.
Requires JavaScript	Yes	Yes	Yes	No
Supported clients	Browser and devices that execute Javascript	Browser and devices that execute Javascript	Browser and devices that execute Javascript	Android and iOS devices

	Challenge rule action	CAPTCHA rule action	JavaScript SDK challenge	Mobile SDK challenge
Supports single-page applications (SPA)	Enforcement only. You can use the Challenge action in conjunction with the SDKs, to ensure that requests have a valid challenge token. You can't use the rule action to deliver the challenge script to the page.	Enforcement only. You can use the CAPTCHA action in conjunction with the SDKs, to ensure that requests have a valid CAPTCHA token. You can't use the rule action to deliver the CAPTCHA script to the page.	Yes	N/A
Additional cost	Yes, for action settings that you explicitly specify, either in the rules that you define or as rule action overrides in rule groups that you use. No in all other cases.	Yes, for action settings that you explicitly specify, either in the rules that you define or as rule action overrides in rule groups that you use. No in all other cases.	No, but requires one of the paid rule groups ACFP, ATP, or Bot Control.	No, but requires one of the paid rule groups ACFP, ATP, or Bot Control .

For details about costs associated with these options, see the intelligent threat mitigation information at AWS WAF Pricing.

It can be simpler to run challenges and provide basic token enforcement by just adding a rule with a Challenge or CAPTCHA action. You might be required to use the rule actions, for example if you don't have access to the application code.

If you can implement the SDKs however, you can save costs and reduce latency in your protection pack (web ACL) evaluation of client web requests, compared to using the Challenge action:

- You can write your SDK implementation to run the challenge at any point in your application. You can acquire the token in the background, prior to any customer action that would send a web request to your protected resource. This way, the token is available to send with your client's first request.
- If instead you acquire tokens by implementing a rule with the Challenge action, the rule and action require additional web request evaluation and processing when the client first sends a request and anytime the token expires. The Challenge action blocks the request that doesn't have a valid, unexpired token, and sends the challenge interstitial back to the client. After the client successfully responds to the challenge, the interstitial resends the original web request with the valid token, which is then evaluated a second time by the protection pack (web ACL).

Options for intelligent threat mitigation managed rule groups

This section compares managed rule group options.

The intelligent threat mitigation AWS Managed Rules rule groups provide management of basic bots, detection and mitigation of sophisticated, malicious bots, detection and mitigation of account takeover attempts, and detection and mitigation of fraudulent account creation attempts. These rule groups, combined with the application integration SDKs described in the prior section, provide the most advanced protections and secure coupling with your client applications.

Comparison of the managed rules group options

	ACFP	ATP	Bot Control common level	Bot Control targeted level
What it is	Manages requests that might be part of fraudulen t account cr eation attempts	Manages requests that might be part of malicious takeover a ttempts on an	Manages common bots that self-identify, with signatures that are unique	Manages targeted bots that don't self-identify, with signatures that

	ACFP	ATP	Bot Control common level	Bot Control targeted level
	on an applicati on's registration and sign-up p ages. Does not manage bots. See AWS WAF Fraud Control account creation fraud preventio n (ACFP) rule group.	application's login page. Does not manage bots. See AWS WAF Fraud Control account takeover prevention (ATP) rule group.	across applicati ons. See AWS WAF Bot Control rule group.	are specific to an application. See AWS WAF Bot Control rule group.

	ACFP	ATP	Bot Control common level	Bot Control targeted level
Good choice for...	Inspection of account creation traffic for fraudulent account creation attacks such creation attempts with username traversal and many new accounts created from a single IP address.	Inspection of login traffic for account takeover attacks such login attempts with password traversal and many login attempts from the same IP address. When used with tokens, also provides aggregate protections such as rate limiting of IPs and client sessions for high volumes of failed login attempts.	Basic bot protection and labeling of common, automated bot traffic.	Targeted protection against sophisticated bots, including rate limiting at the client session level and detection and mitigation of browser automation tools such as Selenium and Puppeteer.
Adds labels that indicate evaluation results	Yes	Yes	Yes	Yes
Adds token labels	Yes	Yes	Yes	Yes

	ACFP	ATP	Bot Control common level	Bot Control targeted level
Blocking for requests that don't have a valid token	Not included. See Blocking requests that don't have a valid AWS WAF token.	Not included. See Blocking requests that don't have a valid AWS WAF token.	Not included. See Blocking requests that don't have a valid AWS WAF token.	Blocks client sessions that send 5 requests without a token.
Requires the AWS WAF token `aws-waf-token`	Required for all rules. See Using application integration SDKs with ACFP.	Required for many rules. See Using application integration SDKs with ATP.	No	Yes
Acquires the AWS WAF token `aws-waf-token`	Yes, enforced by the rule `AllRequests`	No	No	Some rules use Challenge or CAPTCHA rule actions, which acquire tokens.

For details about costs associated with these options, see the intelligent threat mitigation information at AWS WAF Pricing.

If your application faces bots that rotate IP addresses and mimic browser behavior, the common protection level alone won't detect them. These bots don't self-identify and require the behavioral analysis and machine learning that targeted protection provides. For guidance on choosing the right protection level for your specific application, see Choosing and configuring Bot Control for your use case.

Options for rate limiting in rate-based rules and targeted Bot Control rules

This section compares rate-based mitigation options.

The targeted level of the AWS WAF Bot Control rule group and the AWS WAF rate-based rule statement both provide web request rate limiting. The following table compares the two options.

Comparison of options for rate-based detection and mitigation

	AWS WAF rate-based rule	AWS WAF Bot Control targeted rules
How rate limiting is applied	Acts on groups of requests that are coming at too high a rate. You can apply any action except for Allow.	Enforces human-like access patterns and applies dynamic rate limiting, through the use of request tokens.
Based on historical traffic baselines?	No	Yes
Time required to accumulate historic traffic baselines	N/A	Five minutes for dynamic thresholds. N/A for token absent.
Mitigation lag	Usually 30-50 seconds. Can be up to several minutes.	Usually less than 10 seconds. Can be up to several minutes.
Mitigation targets	Configurable. You can group requests using a scope-down statement and by one or more aggregation keys, such as IP address, HTTP method, and query string.	IP addresses and client sessions

	AWS WAF rate-based rule	AWS WAF Bot Control targeted rules
Traffic volume level required to trigger mitigations	Medium - can be as low as 10 requests in the specified time window	Low - intended to detect client patterns such as slow scrapers
Customizable thresholds	Yes	No
Default mitigation action	Console default is Block. No default setting in the API; the setting is required. You can set this to any rule action except Allow.	The rule group rule action settings are Challenge for token absent and CAPTCHA for high volume traffic from a single client session. You can set either of these rules to any valid rule action.
Resiliency against highly distributed attacks	Medium - 10,000 IP address maximum for IP address limiting on its own	Medium - limited to 50,000 total between IP addresses and tokens
AWS WAF Pricing	Included in the standard fees for AWS WAF.	Included in the fees for the targeted level of Bot Control intelligent threat mitigation.
For more information	Using rate-based rule statements in AWS WAF	AWS WAF Bot Control rule group

Best practices for intelligent threat mitigation in AWS WAF

Follow the best practices in this section for the most efficient, cost-effective implementation of the intelligent threat mitigation features.

- **Implement the JavaScript and mobile application integration SDKs** – Implement application integration to enable the full set of ACFP, ATP, or Bot Control functionality in the most effective way possible. The managed rule groups use the tokens provided by the SDKs to separate legitimate client traffic from unwanted traffic at the session level. The application integration SDKs ensure that these tokens are always available. For details, see the following:
  - Using application integration SDKs with ACFP
  - Using application integration SDKs with ATP
  - Using application integration SDKs with Bot Control

  Use the integrations to implement challenges in your client and, for JavaScript, to customize how CAPTCHA puzzles are presented to your end users. For details, see Client application integrations in AWS WAF.

  If you customize CAPTCHA puzzles using the JavaScript API and you use the CAPTCHA rule action anywhere in your protection pack (web ACL), follow the guidance for handling the AWS WAF CAPTCHA response in your client at Handling a CAPTCHA response from AWS WAF. This guidance applies to any rules that use the CAPTCHA action, including those in the ACFP managed rule group and the targeted protection level of the Bot Control managed rule group.
- **Limit the requests that you send to the ACFP, ATP, and Bot Control rule groups** – You incur additional fees for using the intelligent threat mitigation AWS Managed Rules rule groups. The ACFP rule group inspects requests to the account registration and creation endpoints that you specify. The ATP rule group inspects requests to the login endpoint that you specify. The Bot Control rule group inspects every request that reaches it in the protection pack (web ACL) evaluation.

  Consider the following approaches to reduce your use of these rule groups:
  - Exclude requests from inspection with a scope-down statement in the managed rule group statement. You can do this with any nestable statement. For information, see Using scope-down statements in AWS WAF.
  - Exclude requests from inspection by adding rules before the rule group. For rules that you can't use in a scope-down statement and for more complex situations, such as labeling followed by label matching, you might want to add rules that run before the rule groups. For

information, see Using scope-down statements in AWS WAF and Using rule statements in AWS WAF.

- Run the rule groups after less expensive rules. If you have other standard AWS WAF rules that block requests for any reason, run them before these paid rule groups. For more information about rules and rule management, see Using rule statements in AWS WAF.
- If you're using more than one of the intelligent threat mitigation managed rule groups, run them in the following order to keep costs down: Bot Control, ATP, ACFP.

For detailed pricing information, see AWS WAF Pricing.

You can also place less expensive rules (IP reputation, geo-blocking) before Bot Control so that requests blocked by cheaper rules never reach the paid inspection. For more cost management guidance, see Managing costs.

- **Do not limit the requests that you send to the Anti-DDoS rule group** – This rule group operates best when you configure it in to monitor all web traffic that you aren't explicitly allowing through. Position it in your web ACL to be evaluated only after rules with the Allow rule action and before all other rules.
- **For distributed denial of service (DDoS) protection, use either Anti-DDoS or Shield Advanced automatic application layer DDoS mitigation** – The other intelligent threat mitigation rule groups don't provide DDoS protection. ACFP protects against fraudulent account creation attempts to your application's sign-up page. ATP protects against account takeover attempts to your login page. Bot Control focuses on enforcing human-like access patterns using tokens and dynamic rate limiting on client sessions.

  Anti-DDoS allows you to monitor and control DDoS attacks, allowing for quick response and mitigation of threats. Shield Advanced with automatic application layer DDoS mitigation automatically responds to detected DDoS attacks by creating, evaluating, and deploying custom AWS WAF mitigations on your behalf.

  For more information about Shield Advanced, see AWS Shield Advanced overview, and Protecting the application layer (layer 7) with AWS Shield Advanced and AWS WAF.

  For more information about Distributed Denial of Service (DDoS) prevention, see Anti-DDoS rule group and Distributed Denial of Service (DDoS) prevention.
- **Enable the Anti-DDoS rule group and the targeted protection level of the Bot Control rule group during normal web traffic** – These rule categories need time to establish baselines for normal traffic.

Enable the targeted protection level of the Bot Control rule group during normal web traffic – Some rules of the targeted protection level need time to establish baselines for normal traffic patterns before they can recognize and respond to irregular or malicious traffic patterns. For example, the `TGT_ML_*` rules need up to 24 hours to warm up.

Add these protections when you are not experiencing an attack and give them time to establish their baselines before expecting them to respond appropriately. If you add these rules during an attack, you will need to enable the Anti-DDoS rule group in count mode. After the attack subsides, the time to establish a baseline is usually from double to triple the normal required time, because of the skewing added by the attack traffic. For additional information about the rules and any warm-up times that they require, see Rules listing.

- **For distributed denial of service (DDoS) protection, use Shield Advanced automatic application layer DDoS mitigation** – The intelligent threat mitigation rule groups don't provide DDoS protection. ACFP protects against fraudulent account creation attempts to your application's sign-up page. ATP protects against account takeover attempts to your login page. Bot Control focuses on enforcing human-like access patterns using tokens and dynamic rate limiting on client sessions.

  When you use Shield Advanced with automatic application layer DDoS mitigation enabled, Shield Advanced automatically responds to detected DDoS attacks by creating, evaluating, and deploying custom AWS WAF mitigations on your behalf. For more information about Shield Advanced, see AWS Shield Advanced overview, and Protecting the application layer (layer 7) with AWS Shield Advanced and AWS WAF.

- **Use production traffic loads when you establish baselines for the Anti-DDoS rule group** – It is common practice to test other rule groups using artificial test traffic. However, when you test and establish baselines for the Anti-DDoS rule group, we recommend that you use traffic flows that reflect the loads in your production environment. Establishing Anti-DDoS baselines using typical traffic is the best way to ensure your resources will be protected when the rule group is enabled in a production environment.

- **Tune and configure token handling** – Adjust the protection pack (web ACL)'s token handling for the best user experience.

  - To reduce operating costs and improve your end user's experience, tune your token management immunity times to the longest that your security requirements permit. This keeps the use of CAPTCHA puzzles and silent challenges to a minimum. For information, see Setting timestamp expiration and token immunity times in AWS WAF.

  - To enable token sharing between protected applications, configure a token domain list for your protection pack (web ACL). For information, see Specifying token domains and domain lists in AWS WAF.
- **Reject requests with arbitrary host specifications** – Configure your protected resources to require that the `Host` headers in web requests match the targeted resource. You can accept one value or a specific set of values, for example `myExampleHost.com` and `www.myExampleHost.com`, but don't accept arbitrary values for the host.
- **For Application Load Balancers that are origins for CloudFront distributions, configure CloudFront and AWS WAF for proper token handling** – If you associate your protection pack (web ACL) to an Application Load Balancer and you deploy the Application Load Balancer as the origin for a CloudFront distribution, see Required configuration for Application Load Balancers that are CloudFront origins.
- **Test and tune before deploying** – Before you implement any changes to your protection pack (web ACL), follow the testing and tuning procedures in this guide to be sure that you're getting the behavior you expect. This is especially important for these paid features. For general guidance, see Testing and tuning your AWS WAF protections. For information specific to the paid managed rule groups, see Testing and deploying ACFP, Testing and deploying ATP, and Testing and deploying AWS WAF Bot Control.

Token use in AWS WAF intelligent threat mitigation

This section explains what AWS WAF tokens do.

AWS WAF tokens are an integral part of the enhanced protections offered by AWS WAF intelligent threat mitigation. A token, sometimes called a fingerprint, is a collection of information about a single client session that the client stores and provides with every web request that it sends. AWS WAF uses tokens to identify and separate malicious client sessions from legitimate sessions, even when both originate from a single IP address. Token use imposes costs that are negligible for legitimate users, but expensive at scale for botnets.

AWS WAF uses tokens to support its browser and end user challenge functionality, which is provided by the application integration SDKs and by the rule actions Challenge and CAPTCHA. Additionally, tokens enable features of the AWS WAF Bot Control and account takeover prevention managed rule groups.

AWS WAF creates, updates, and encrypts tokens for clients that successfully respond to silent challenges and CAPTCHA puzzles. When a client with a token sends a web request, it includes the encrypted token, and AWS WAF decrypts the token and verifies its contents.

Topics

- How AWS WAF uses tokens
- AWS WAF token characteristics
- Setting timestamp expiration and token immunity times in AWS WAF
- Specifying token domains and domain lists in AWS WAF
- Types of token labels in AWS WAF
- Blocking requests that don't have a valid AWS WAF token
- Required configuration for Application Load Balancers that are CloudFront origins

How AWS WAF uses tokens

This section explains how AWS WAF uses tokens.

AWS WAF uses tokens to record and verify the following types of client session validation:

- **CAPTCHA** – CAPTCHA puzzles help distinguish bots from human users. A CAPTCHA is run only by the CAPTCHA rule action. Upon successful completion of the puzzle, the CAPTCHA script updates the token's CAPTCHA timestamp. For more information, see CAPTCHA and Challenge in AWS WAF.
- **Challenge** – Challenges run silently to help distinguish regular client sessions from bot sessions and to make it more costly for bots to operate. When the challenge completes successfully, the challenge script automatically procures a new token from AWS WAF if needed, and then updates the token's challenge timestamp.

  AWS WAF runs challenges in the following situations:

  - **Application integration SDKs** – The application integration SDKs run inside your client application sessions and help ensure that login attempts are only allowed after the client has successfully responded to a challenge. For more information, see Client application integrations in AWS WAF.
  - **Challenge rule action** – For more information, see CAPTCHA and Challenge in AWS WAF.
  - **CAPTCHA** – When a CAPTCHA interstitial runs, if the client doesn't have a token yet, the script automatically runs a challenge first, to verify the client session and to initialize the token.

Tokens are required by many of the rules in the intelligent threat AWS Managed Rules rule groups. The rules use tokens to do things like distinguish between clients at the session level, to determine browser characteristics, and to understand the level of human interactivity on the application web page. These rule groups invoke AWS WAF token management, which applies token labeling that the rule groups then inspect.

- **AWS WAF Fraud Control account creation fraud prevention (ACFP)** – The ACFP rules require web requests with valid tokens. For more information about the rules, see AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group.
- **AWS WAF Fraud Control account takeover prevention (ATP)** – The ATP rules that prevent high volume and long lasting client sessions require web requests that have a valid token with an unexpired challenge timestamp. For more information, see AWS WAF Fraud Control account takeover prevention (ATP) rule group.
- **AWS WAF Bot Control** – The targeted rules in this rule group place a limit on the number of web requests that a client can send without a valid token, and they use token session tracking for session-level monitoring and management. As needed, the rules apply the Challenge and CAPTCHA rule actions to enforce token acquisition and valid client behavior. For more information, see AWS WAF Bot Control rule group.

AWS WAF token characteristics

Each token has the following characteristics:

- The token is stored in a cookie named `aws-waf-token`.
- The token is encrypted.
- The token fingerprints the client session with a sticky granular identifier that contains the following information:
  - The timestamp of the client's latest successful response to a silent challenge.
  - The timestamp of the end user's latest successful response to a CAPTCHA. This is only present if you use CAPTCHA in your protections.
  - Additional information about the client and client behavior that can help separate your legitimate clients from unwanted traffic. The information includes various client identifiers and client-side signals that can be used to detect automated activities. The information gathered is non-unique and can't be mapped to an individual human being.

- All tokens include data from client browser interrogation, such as indications of automation and browser setting inconsistencies. This information is retrieved by the scripts that are run by the Challenge action and by the client application SDKs. The scripts actively interrogate the browser and put the results into the token.
- Additionally, when you implement a client application integration SDK, the token includes passively collected information about the end user's interactivity with the application page. Interactivity includes mouse movements, key presses, and interactions with any HTML form that's present on the page. This information helps AWS WAF detect the level of human interactivity in the client, to challenge users that do not seem to be human. For information about client side integrations, see Client application integrations in AWS WAF.

For security reasons, AWS doesn't provide a complete description of the contents of AWS WAF tokens or detailed information about the token encryption process.

Setting timestamp expiration and token immunity times in AWS WAF

This section explains how challenge and CAPTCHA timestamps expire.

AWS WAF uses challenge and CAPTCHA immunity times to control how frequently a single client session can be presented with a challenge or CAPTCHA. After an end user successfully responds to a CAPTCHA, the CAPTCHA immunity time determines how long the end user remains immune from being presented with another CAPTCHA. Similarly, the challenge immunity time determines how long a client session remains immune from being challenged again after successfully responding to a challenge.

How AWS WAF token immunity times work

AWS WAF records a successful response to a challenge or CAPTCHA by updating the corresponding timestamp inside the token. When AWS WAF inspects the token for challenge or CAPTCHA, it subtracts the timestamp from the current time. If the result is greater than the configured immunity time, the timestamp is expired.

Configurable aspects of AWS WAF token immunity times

You can configure the challenge and CAPTCHA immunity times in the protection pack (web ACL) and also in any rule that uses the CAPTCHA or Challenge rule action.

- The default protection pack (web ACL) setting for both immunity times is 300 seconds.

- You can specify the immunity time for any rule that uses the CAPTCHA or Challenge action. If you don't specify the immunity time for the rule, it inherits the setting from the protection pack (web ACL).
- For a rule inside a rule group that uses the CAPTCHA or Challenge action, if you don't specify the immunity time for the rule, it will inherit the setting from each protection pack (web ACL) where you use the rule group.
- The application integration SDKs use the protection pack (web ACL)'s challenge immunity time.
- The minimum value for the challenge immunity time is 300 seconds. The minimum value for the CAPTCHA immunity time is 60 seconds. The maximum value for both immunity times is 259,200 seconds, or three days.

You can use the protection pack (web ACL) and rule level immunity time settings to tune the CAPTCHA action, Challenge, or SDK challenge management behavior. For example, you might configure rules that control access to highly sensitive data with low immunity times, and then set higher immunity times in your protection pack (web ACL) for your other rules and the SDKs to inherit.

In particular for CAPTCHA, solving a puzzle can degrade your customer's website experience, so tuning the CAPTCHA immunity time can help you mitigate the impact on customer experience while still providing the protections that you want.

For additional information about tuning the immunity times for your use of the Challenge and CAPTCHA rule actions, see Best practices for using the CAPTCHA and Challenge actions.

Where to set the AWS WAF token immunity times

You can set the immunity times in your protection pack (web ACL) and in your rules that use the Challenge and CAPTCHA rule actions.

For general information about managing a protection pack (web ACL) and its rules, see Viewing web traffic metrics in AWS WAF.

Where to set the immunity time for a protection pack (web ACL)

- **Console** – When you edit the protection pack (web ACL), in the **Rules** tab, edit and change the settings in the **protection pack (web ACL) CAPTCHA configuration** and **protection pack (web ACL) Challenge configuration** panes. In the console, you can configure the protection pack (web ACL) CAPTCHA and challenge immunity times only after you've created the protection pack (web ACL).

- **Outside of the console** – The protection pack (web ACL) data type has CAPTCHA and challenge configuration parameters, which you can configure and provide to your create and update operations on the protection pack (web ACL).

Where to set the immunity time for a rule

- **Console** – When you create or edit a rule and specify the CAPTCHA or Challenge action, you can modify the rule's immunity time setting.
- **Outside of the console** – The rule data type has CAPTCHA and challenge configuration parameters, which you can configure when you define the rule.

Specifying token domains and domain lists in AWS WAF

This section explains how to configure the domains that AWS WAF uses in tokens and that it accepts in tokens.

When AWS WAF creates a token for a client, it configures it with a token domain. When AWS WAF inspects a token in a web request, it rejects the token as invalid if its domain doesn't match any of the domains that are considered valid for the protection pack (web ACL).

By default, AWS WAF only accepts tokens whose domain setting exactly matches the host domain of the resource that's associated with the protection pack (web ACL). This is the value of the `Host` header in the web request. In a browser, you can find this domain in the JavaScript `window.location.hostname` property and in the address that your user sees in their address bar.

You can also specify acceptable token domains in your protection pack (web ACL) configuration, as described in the following section. In this case, AWS WAF accepts both exact matches with the host header and matches with domains in the token domain list.

You can specify token domains for AWS WAF to use when setting the domain and when evaluating a token in a protection pack (web ACL). The domains that you specify can't be public suffixes such as `gov.au`. For the domains that you can't use, see the list https://publicsuffix.org/list/public_suffix_list.dat under Public suffix list.

AWS WAF protection pack (web ACL) token domain list configuration

You can configure a protection pack (web ACL) to share tokens across multiple protected resources by providing a token domain list with the additional domains that you want AWS WAF to accept.

With a token domain list, AWS WAF still accepts the resource's host domain. Additionally, it accepts all domains in the token domain list, including their prefixed subdomains.

For example, a domain specification `example.com` in your token domain list matches `example.com` (from `http://example.com/`), `api.example.com`, (from `http://api.example.com/`), and `www.example.com` (from `http://www.example.com/`). It doesn't match `example.api.com`, (from `http://example.api.com/`), or `apiexample.com` (from `http://apiexample.com/`).

You can configure the token domain list in your protection pack (web ACL) when you create or edit it. For general information about managing a protection pack (web ACL), see Viewing web traffic metrics in AWS WAF.

AWS WAF token domain settings

AWS WAF creates tokens at the request of the challenge scripts, which are run by the application integration SDKs and the Challenge and CAPTCHA rule actions.

The domain that AWS WAF sets in a token is determined by the type of challenge script that's requesting it and any additional token domain configuration that you provide. AWS WAF sets the domain in the token to the shortest, most general setting that it can find in the configuration.

- **JavaScript SDK** – You can configure the JavaScript SDK with a token domain specification, which can include one or more domains. The domains that you configure must be domains that AWS WAF will accept, based on the protected host domain and the protection pack (web ACL)'s token domain list.

  When AWS WAF issues a token for the client, it sets the token domain to one that matches the host domain and is the shortest, from among the host domain and the domains in your configured list. For example, if the host domain is `api.example.com` and the token domain list has `example.com`, AWS WAF uses `example.com` in the token, because it matches the host domain and is shorter. If you don't provide a token domain list in the JavaScript API configuration, AWS WAF sets the domain to the host domain of the protected resource.

  For more information, see Providing domains for use in the tokens.
- **Mobile SDK** – In your application code, you must configure the mobile SDK with a token domain property. This property must be a domain that AWS WAF will accept, based on the protected host domain and the protection pack (web ACL)'s token domain list.

When AWS WAF issues a token for the client, it uses this property as the token domain. AWS WAF doesn't use the host domain in the tokens that it issues for the mobile SDK client.

For more information, see the `WAFConfiguration` `domainName` setting at AWS WAF mobile SDK specification.

- **Challenge action** – If you specify a token domain list in the protection pack (web ACL), AWS WAF sets the token domain to one that matches the host domain and is the shortest, from among the host domain and the domains in the list. For example, if the host domain is `api.example.com` and the token domain list has `example.com`, AWS WAF uses `example.com` in the token, because it matches the host domain and is shorter. If you don't provide a token domain list in the protection pack (web ACL), AWS WAF sets the domain to the host domain of the protected resource.

Types of token labels in AWS WAF

This section describes the labels that AWS WAF token management adds to web requests. For general information about labels, see Web request labeling in AWS WAF.

When you use any of the AWS WAF bot or fraud control managed rule groups, the rule groups use AWS WAF token management to inspect the web request tokens and apply token labeling to the requests. For information about the managed rule groups, see AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group, AWS WAF Fraud Control account takeover prevention (ATP) rule group, and AWS WAF Bot Control rule group .

> **Note**
>
> AWS WAF applies token labels only when you use one of these intelligent threat mitigation managed rule groups.

Token management can add the following labels to web requests.

Client session label

The label `awswaf:managed:token:id:`*`identifier`* contains a unique identifier that AWS WAF token management uses to identify the client session. The identifier can change if the client acquires a new token, for example after discarding the token it was using.

> **Note**
>
> AWS WAF doesn't report Amazon CloudWatch metrics for this label.

Browser fingerprint label

The label `awswaf:managed:token:fingerprint:`*`fingerprint-identifier`* contains a robust browser fingerprint identifier that AWS WAF token management computes from various client browser signals. This identifier stays the same across multiple token acquisition attempts. The fingerprint identifier is not unique to a single client.

> **Note**
>
> AWS WAF doesn't report Amazon CloudWatch metrics for this label.

Token status labels: Label namespace prefixes

Token status labels report on the status of the token and of the challenge and CAPTCHA information that it contains.

Each token status label begins with one of the following namespace prefixes:

- `awswaf:managed:token:` – Used to report the general status of the token and to report on the status of the token's challenge information.
- `awswaf:managed:captcha:` – Used to report on the status of the token's CAPTCHA information.

Token status labels: Label names

Following the prefix, the rest of the label provides detailed token status information:

- `accepted` – The request token is present and contains the following:
  - A valid challenge or CAPTCHA solution.
  - An unexpired challenge or CAPTCHA timestamp.
  - A domain specification that's valid for the protection pack (web ACL).

Example: The label `awswaf:managed:token:accepted` indicates that the web requests's token has a valid challenge solution, an unexpired challenge timestamp, and a valid domain.

- `rejected` – The request token is present but doesn't meet the acceptance criteria.

  Along with the rejected label, token management adds a custom label namespace and name to indicate the reason.

  - `rejected:not_solved` – The token is missing the challenge or CAPTCHA solution.
  - `rejected:expired` – The token's challenge or CAPTCHA timestamp has expired, according to your protection pack (web ACL)'s configured token immunity times.
  - `rejected:domain_mismatch` – The token's domain isn't a match for your protection pack (web ACL)'s token domain configuration.
  - `rejected:invalid` – AWS WAF couldn't read the indicated token.

  Example: The labels `awswaf:managed:captcha:rejected` and `awswaf:managed:captcha:rejected:expired` together indicate that the request didn't have a valid CAPTCHA solve because the CAPTCHA timestamp in the token has exceeded the CAPTCHA token immunity time that's configured in the protection pack (web ACL).

- `absent` – The request doesn't have the token or the token manager couldn't read it.

  Example: The label `awswaf:managed:captcha:absent` indicates that the request doesn't have the token.

Blocking requests that don't have a valid AWS WAF token

This section explains how to block login requests that are missing their tokens when using the AWS WAF mobile SDK.

When you use the intelligent threat AWS Managed Rules rule groups `AWSManagedRulesACFPRuleSet`, `AWSManagedRulesATPRuleSet`, and `AWSManagedRulesBotControlRuleSet`, the rule groups invoke AWS WAF token management to evaluate the status of the web request token and to label the requests accordingly.

> **Note**
>
> Token labeling is only applied to web requests that you evaluate using one of these managed rule groups.

For information about the labeling that token management applies, see the preceding section, Types of token labels in AWS WAF.

The intelligent threat mitigation managed rule groups then handle token requirements as follows:

- The `AWSManagedRulesACFPRuleSet` `AllRequests` rule is configured to run the Challenge action against all requests, effectively blocking any that don't have the `accepted` token label.
- The `AWSManagedRulesATPRuleSet` blocks requests that have the `rejected` token label, but it doesn't block requests with the `absent` token label.
- The `AWSManagedRulesBotControlRuleSet` targeted protection level challenges clients after they send five requests without an `accepted` token label. It doesn't block an individual request that doesn't have a valid token. The common protection level of the rule group doesn't manage token requirements.

For additional details about the intelligent threat rule groups, see AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group, AWS WAF Fraud Control account takeover prevention (ATP) rule group and AWS WAF Bot Control rule group.

To block requests that are missing tokens when using the Bot Control or ATP managed rule group

With the Bot Control and ATP rule groups, it's possible for a request without a valid token to exit the rule group evaluation and continue to be evaluated by the protection pack (web ACL).

To block all requests that are missing their token or whose token is rejected, add a rule to run immediately after the managed rule group to capture and block requests that the rule group doesn't handle for you.

The following is an example JSON listing for a protection pack (web ACL) that uses the ATP managed rule group. The protection pack (web ACL) has an added rule to capture the `awswaf:managed:token:absent` label and handle it. The rule narrows its evaluation to web requests going to the login endpoint, to match the scope of the ATP rule group. The added rule is listed in bold.

```
{
  "Name": "exampleWebACL",
  "Id": "55555555-6666-7777-8888-999999999999",
  "ARN": "arn:aws:wafv2:us-east-1:111111111111:regional/webacl/
exampleWebACL/55555555-4444-3333-2222-111111111111",
```

```
  "DefaultAction": {
    "Allow": {}
  },
  "Description": "",
  "Rules": [
    {
      "Name": "AWS-AWSManagedRulesATPRuleSet",
      "Priority": 1,
      "Statement": {
        "ManagedRuleGroupStatement": {
          "VendorName": "AWS",
          "Name": "AWSManagedRulesATPRuleSet",
          "ManagedRuleGroupConfigs": [
            {
              "AWSManagedRulesATPRuleSet": {
                "LoginPath": "/web/login",
                "RequestInspection": {
                  "PayloadType": "JSON",
                  "UsernameField": {
                    "Identifier": "/form/username"
                  },
                  "PasswordField": {
                    "Identifier": "/form/password"
                  }
                },
                "ResponseInspection": {
                  "StatusCode": {
                    "SuccessCodes": [
                      200
                    ],
                    "FailureCodes": [
                      401,
                      403,
                      500
                    ]
                  }
                }
              }
            }
          ]
        }
      },
      "OverrideAction": {
        "None": {}
```

```
      },
      "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "AWS-AWSManagedRulesATPRuleSet"
      }
    },
    {
      "Name": "RequireTokenForLogins",
      "Priority": 2,
      "Statement": {
        "AndStatement": {
          "Statements": [
            {
              "Statement": {
                "LabelMatchStatement": {
                  "Scope": "LABEL",
                  "Key": "awswaf:managed:token:absent"
                }
              }
            },
            {
              "ByteMatchStatement": {
                "SearchString": "/web/login",
                "FieldToMatch": {
                  "UriPath": {}
                },
                "TextTransformations": [
                  {
                    "Priority": 0,
                    "Type": "NONE"
                  }
                ],
                "PositionalConstraint": "STARTS_WITH"
              }
            },
            {
              "ByteMatchStatement": {
                "SearchString": "POST",
                "FieldToMatch": {
                  "Method": {}
                },
                "TextTransformations": [
                  {
```

```
                    "Priority": 0,
                    "Type": "NONE"
                  }
                ],
                "PositionalConstraint": "EXACTLY"
              }
            }
          ]
        }
      },
      "Action": {
        "Block": {}
      },
      "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "RequireTokenForLogins"
      }
    }
  ],
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "exampleWebACL"
  },
  "Capacity": 51,
  "ManagedByFirewallManager": false,
  "RetrofittedByFirewallManager": false,
  "LabelNamespace": "awswaf:111111111111:webacl:exampleWebACL:"
}
```

Required configuration for Application Load Balancers that are CloudFront origins

Read this section if you associate your protection pack (web ACL) to an Application Load Balancer and you deploy the Application Load Balancer as the origin for a CloudFront distribution.

With this architecture, you need to provide the following additional configuration in order for the token information to be handled correctly.

- Configure CloudFront to forward the `aws-waf-token` cookie to the Application Load Balancer. By default, CloudFront removes cookies from the web request before forwarding it to the origin. To keep the token cookie with the web request, configure CloudFront cache behavior to include

either just the token cookie or all cookies. For information about how to do this, see Caching content based on cookies in the *Amazon CloudFront Developer Guide*.

- Configure AWS WAF so that it recognizes the domain of the CloudFront distribution as a valid token domain. By default, CloudFront sets the `Host` header to the Application Load Balancer origin, and AWS WAF uses that as the domain of the protected resource. The client browser, however, sees the CloudFront distribution as the host domain, and tokens that are generated for the client use the CloudFront domain as the token domain. Without any additional configuration, when AWS WAF checks the protected resource domain against the token domain, it will get a mismatch. To fix this, add the CloudFront distribution domain name to the token domain list in your protection pack (web ACL) configuration. For information about how to do this, see AWS WAF protection pack (web ACL) token domain list configuration.

AWS WAF Fraud Control account creation fraud prevention (ACFP)

This section explains what AWS WAF Fraud Control account creation fraud prevention (ACFP) does.

Account creation fraud is an online illegal activity in which an attacker tries to create one or more fake accounts. Attackers use fake accounts for fraudulent activities such as abusing promotional and sign up bonuses, impersonating someone, and cyberattacks like phishing. The presence of fake accounts can negatively impact your business by damaging your reputation with customers and exposure to financial fraud.

You can monitor and control account creation fraud attempts by implementing the ACFP feature. AWS WAF offers this feature in the AWS Managed Rules rule group `AWSManagedRulesACFPRuleSet` with companion application integration SDKs.

The ACFP managed rule group labels and manages requests that might be part of malicious account creation attempts. The rule group does this by inspecting account creation attempts that clients send to your application's account sign-up endpoint.

ACFP protects your account sign-up pages by monitoring account sign-up requests for anomalous activity and by automatically blocking suspicious requests. The rule group uses request identifiers, behavioral analysis, and machine learning to detect fraudulent requests.

- **Request inspection** – ACFP gives you visibility and control over anomalous account creation attempts and attempts that use stolen credentials, to prevent the creation of fraudulent accounts. ACFP checks email and password combinations against its stolen credential database, which is updated regularly as new leaked credentials are found on the dark web. ACFP evaluates

the domains used in email addresses, and monitors the use of phone numbers and address fields to verify the entries and to detects fraudulent behavior. ACFP aggregates data by IP address and client session, to detect and block clients that send too many requests of a suspicious nature.

- **Response inspection** – For CloudFront distributions, in addition to inspecting incoming account creation requests, the ACFP rule group inspects your application's responses to account creation attempts, to track success and failure rates. Using this information, ACFP can temporarily block client sessions or IP addresses that have too many failed attempts. AWS WAF performs response inspection asynchronously, so this doesn't increase latency in your web traffic. AWS WAF doesn't inspect responses for web requests that clients send over HTTP/3 (QUIC).

> **Note**
>
> You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

> **Note**
>
> The ACFP feature is not available for Amazon Cognito user pools.

Topics

- AWS WAF ACFP components
- Using application integration SDKs with ACFP
- Adding the ACFP managed rule group to your web ACL
- Testing and deploying ACFP
- AWS WAF Fraud Control account creation fraud prevention (ACFP) examples

AWS WAF ACFP components

The primary components of AWS WAF Fraud Control account creation fraud prevention (ACFP) are the following:

- **`AWSManagedRulesACFPRuleSet`** – The rules in this AWS Managed Rules rule group detect, label, and handle various types of fraudulent account creation activity. The rule group inspects

HTTP GET text/html requests that clients send to the specified account registration endpoint and POST web requests that clients send to the specified account sign-up endpoint. For protected CloudFront distributions, the rule group also inspects the responses that the distribution sends back to account creation requests. For a list of this rule group's rules, see AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group. You include this rule group in your protection pack (web ACL) using a managed rule group reference statement. For information about using this rule group, see Adding the ACFP managed rule group to your web ACL.

> **Note**
>
> You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

- **Details about your application's account registration and creation pages** – You must provide information about your account registration and creation pages when you add the `AWSManagedRulesACFPRuleSet` rule group to your protection pack (web ACL). This lets the rule group narrow the scope of the requests it inspects and properly validate account creation web requests. The registration page must accept GET text/html requests. The account creation path must accept POST requests. The ACFP rule group works with usernames that are in email format. For more information, see Adding the ACFP managed rule group to your web ACL.
- **For protected CloudFront distributions, details about how your application responds to account creation attempts** – You provide details about your application's responses to account creation attempts, and the ACFP rule group tracks and manages bulk account creation attempts from a single IP address or single client session. For information about configuring this option, see Adding the ACFP managed rule group to your web ACL.
- **JavaScript and mobile application integration SDKs** – Implement the AWS WAF JavaScript and mobile SDKs with your ACFP implementation to enable the full set of capabilities that the rule group offers. Many of the ACFP rules use the information provided by the SDKs for session level client verification and behavior aggregation, required to separate legitimate client traffic from bot traffic. For more information about the SDKs, see Client application integrations in AWS WAF.

You can combine your ACFP implementation with the following to help you monitor, tune, and customize your protections.

- **Logging and metrics** – You can monitor your traffic, and understand how the ACFP managed rule group affects it, by configuring and enabling logs, Amazon Security Lake data collection,

and Amazon CloudWatch metrics for your protection pack (web ACL). The labels that `AWSManagedRulesACFPRuleSet` adds to your web requests are included in the data. For information about the options, see Logging AWS WAF protection pack (web ACL) traffic, Monitoring with Amazon CloudWatch, and What is Amazon Security Lake?.

Depending on your needs and the traffic that you see, you might want to customize your `AWSManagedRulesACFPRuleSet` implementation. For example, you might want to exclude some traffic from ACFP evaluation, or you might want to alter how it handles some of the account creation fraud attempts that it identifies, using AWS WAF features like scope-down statements or label matching rules.

- **Labels and label matching rules** – For any of the rules in `AWSManagedRulesACFPRuleSet`, you can switch the blocking behavior to count, and then match against the labels that are added by the rules. Use this approach to customize how you handle web requests that are identified by the ACFP managed rule group. For more information about labeling and using label match statements, see Label match rule statement and Web request labeling in AWS WAF.
- **Custom requests and responses** – You can add custom headers to the requests that you allow and you can send custom responses for requests that you block. To do this, you pair your label matching with the AWS WAF custom request and response features. For more information about customizing requests and responses, see Customized web requests and responses in AWS WAF.

Using application integration SDKs with ACFP

We highly recommend implementing the application integration SDKs, for the most efficient use of the ACFP rule group.

- **Complete rule group functionality** – The ACFP rule `SignalClientHumanInteractivityAbsentLow` only works with tokens that are populated by the application integrations. This rule detects and manages abnormal human interactivity with the application page. The application integration SDKs can detect normal human interactivity through mouse movements, key presses, and other measurements. The interstitials that are sent by the rule actions CAPTCHA and Challenge can't provide this type of data.
- **Reduced latency** – The rule group rule `AllRequests` applies the Challenge rule action to any request that doesn't already have a challenge token. When this happens, the request is evaluated by the rule group twice: once without the token, and then a second time after the token is acquired by means of the Challenge action interstitial. You aren't charged any added fees for only using the `AllRequests` rule, but this approach adds overhead to your web traffic and adds

latency to your end user experience. If you acquire the token client-side using the application integrations, before sending the account creation request, the ACFP rule group evaluates the request once.

For more information about the rule group capabilities see AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group.

For information about the SDKs, see Client application integrations in AWS WAF. For information about AWS WAF tokens, see Token use in AWS WAF intelligent threat mitigation. For information about the rule actions, see CAPTCHA and Challenge in AWS WAF.

Adding the ACFP managed rule group to your web ACL

This section explains how to add and configure the `AWSManagedRulesACFPRuleSet` rule group.

To configure the ACFP managed rule group to recognize account creation fraud activities in your web traffic, you provide information about how clients access your registration page and send account creation requests to your application. For protected Amazon CloudFront distributions, you also provide information about how your application responds to account creation requests. This configuration is in addition to the normal configuration for a managed rule group.

For the rule group description and rules listing, see AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group.

> **Note**
>
> The ACFP stolen credentials database only contains usernames in email format.

This guidance is intended for users who know generally how to create and manage AWS WAF protection packs (web ACLs), rules, and rule groups. Those topics are covered in prior sections of this guide. For basic information about how to add a managed rule group to your protection pack (web ACL), see Adding a managed rule group to a protection pack (web ACL) through the console.

Follow best practices

Use the ACFP rule group in accordance with the best practices at Best practices for intelligent threat mitigation in AWS WAF.

To use the `AWSManagedRulesACFPRuleSet` rule group in your protection pack (web ACL)

1. Add the AWS managed rule group, `AWSManagedRulesACFPRuleSet` to your protection pack (web ACL), and **Edit** the rule group settings before saving.

   > **Note**
   >
   > You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

2. In the **Rule group configuration** pane, provide the information that the ACFP rule group uses to inspect account creation requests.

   a. For **Use regular expression in paths**, toggle this on if you want AWS WAF to perform regular expression matching for your registration and account creation page path specifications.

      AWS WAF supports the pattern syntax used by the PCRE library `libpcre` with some exceptions. The library is documented at PCRE - Perl Compatible Regular Expressions. For information about AWS WAF support, see Supported regular expression syntax in AWS WAF.

   b. For **Registration page path**, provide the path of the registration page endpoint for your application. This page must accept `GET` text/html requests. The rule group inspects only HTTP `GET` text/html requests to your specified registration page endpoint.

      > **Note**
      >
      > Matching for endpoints is case insensitive. Regex specifications must not contain the flag `(?-i)`, which disables case insensitive matching. String specifications must start with a forward slash `/`.

      For example, for the URL `https://example.com/web/registration`, you could provide the string path specification `/web/registration`. Registration page paths that start with the path that you provide are considered a match. For example `/web/registration` matches the registration paths `/web/registration`, `/web/registration/`, `/web/registrationPage`, and `/web/registration/thisPage`, but doesn't match the path `/home/web/registration` or `/website/registration`.

> **Note**
>
> Ensure that your end users load the registration page before they submit an account creation request. This helps ensure that the account creation requests from the client include valid tokens.

c. For **Account creation path**, provide the URI in your website that accepts completed new user details. This URI must accept `POST` requests.

> **Note**
>
> Matching for endpoints is case insensitive. Regex specifications must not contain the flag `(?-i)`, which disables case insensitive matching. String specifications must start with a forward slash `/`.

For example, for the URL `https://example.com/web/newaccount`, you could provide the string path specification `/web/newaccount`. Account creation paths that start with the path that you provide are considered a match. For example `/web/newaccount` matches the account creation paths `/web/newaccount`, `/web/newaccount/`, `/web/newaccountPage`, and `/web/newaccount/thisPage`, but doesn't match the path `/home/web/newaccount` or `/website/newaccount`.

d. For **Request inspection**, specify how your application accepts account creation attempts by providing the request payload type and the names of the fields within the request body where the username, password, and other account creation details are provided.

> **Note**
>
> For the primary address and phone number fields, provide the fields in the order in which they appear in the request payload.

Your specification of the field names depends on the payload type.

- **JSON payload type** – Specify the field names in JSON pointer syntax. For information about the JSON Pointer syntax, see the Internet Engineering Task Force (IETF) documentation JavaScript Object Notation (JSON) Pointer.

For example, for the following example JSON payload, the username field specification is `/signupform/username` and the primary address field specifications are `/signupform/addrp1`, `/signupform/addrp2`, and `/signupform/addrp3`.

```
{
    "signupform": {
        "username": "THE_USERNAME",
        "password": "THE_PASSWORD",
        "addrp1": "PRIMARY_ADDRESS_LINE_1",
        "addrp2": "PRIMARY_ADDRESS_LINE_2",
        "addrp3": "PRIMARY_ADDRESS_LINE_3",
        "phonepcode": "PRIMARY_PHONE_CODE",
        "phonepnumber": "PRIMARY_PHONE_NUMBER"
    }
}
```

- **FORM_ENCODED payload type** – Use the HTML form names.

  For example, for an HTML form with user and password input elements named `username1` and `password1`, the username field specification is `username1` and the password field specification is `password1`.

e. If you're protecting Amazon CloudFront distributions, then under **Response inspection**, specify how your application indicates success or failure in its responses to account creation attempts.

> **Note**
>
> ACFP response inspection is available only in protection packs (web ACLs) that protect CloudFront distributions. AWS WAF doesn't inspect responses for web requests that clients send over HTTP/3 (QUIC).

Specify a single component in the account creation response that you want ACFP to inspect. For the **Body** and **JSON** component types, AWS WAF can inspect the first 65,536 bytes (64 KB) of the component.

Provide your inspection criteria for the component type, as indicated by the interface. You must provide both success and failure criteria to inspect for in the component.

For example, say your application indicates the status of an account creation attempt in the status code of the response, and uses `200 OK` for success and `401 Unauthorized` or `403 Forbidden` for failure. You would set the response inspection **Component type** to **Status code**, then in the **Success** text box enter `200` and in the **Failure** text box, enter `401` on the first line and `403` on the second.

The ACFP rule group only counts responses that match your success or failure inspection criteria. The rule group rules act on clients while they have too high a success rate among the responses that are counted, in order to mitigate bulk account creation attempts. For accurate behavior by the rule group rules, be sure to provide complete information for both successful and failed account creation attempts.

To see the rules that inspect account creation responses, look for `VolumetricIPSuccessfulResponse` and `VolumetricSessionSuccessfulResponse` in the rules listing at AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group.

3. Provide any additional configuration that you want for the rule group.

   You can further limit the scope of requests that the rule group inspects by adding a scope-down statement to the managed rule group statement. For example, you can inspect only requests with a specific query argument or cookie. The rule group will only inspect requests that match the criteria in your scope-down statement and that are sent to the account registration and account creation paths that you specified in the rule group configuration. For information about scope-down statements, see Using scope-down statements in AWS WAF.

4. Save your changes to the protection pack (web ACL).

Before you deploy your ACFP implementation for production traffic, test and tune it in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune the rules in count mode with your production traffic before enabling them. See the section that follows for guidance.

Testing and deploying ACFP

This section provides general guidance for configuring and testing an AWS WAF Fraud Control account creation fraud prevention (ACFP) implementation for your site. The specific steps that you choose to follow will depend on your needs, resources, and web requests that you receive.

This information is in addition to the general information about testing and tuning provided at Testing and tuning your AWS WAF protections.

Note

AWS Managed Rules are designed to protect you from common web threats. When used in accordance with the documentation, AWS Managed Rules rule groups add another layer of security for your applications. However, AWS Managed Rules rule groups aren't intended as a replacement for your security responsibilities, which are determined by the AWS resources that you select. Refer to the Shared Responsibility Model to ensure that your resources in AWS are properly protected.

Production traffic risk

Before you deploy your ACFP implementation for production traffic, test and tune it in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune the rules in count mode with your production traffic before enabling them.

AWS WAF provides test credentials that you can use to verify your ACFP configuration. In the following procedure, you'll configure a test protection pack (web ACL) to use the ACFP managed rule group, configure a rule to capture the label added by the rule group, and then run an account creation attempt using these test credentials. You'll verify that your protection pack (web ACL) has properly managed the attempt by checking the Amazon CloudWatch metrics for the account creation attempt.

This guidance is intended for users who know generally how to create and manage AWS WAF protection packs (web ACLs), rules, and rule groups. Those topics are covered in prior sections of this guide.

To configure and test an AWS WAF Fraud Control account creation fraud prevention (ACFP) implementation

Perform these steps first in a test environment, then in production.

1. **Add the AWS WAF Fraud Control account creation fraud prevention (ACFP) managed rule group in count mode**

   > **Note**
   >
   > You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

   Add the AWS Managed Rules rule group `AWSManagedRulesACFPRuleSet` to a new or existing protection pack (web ACL) and configure it so that it doesn't alter the current protection pack (web ACL) behavior. For details about the rules and labels for this rule group, see AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group.

   - When you add the managed rule group, edit it and do the following:
     - In the **Rule group configuration** pane, provide the details of your application's account registration and creation pages. The ACFP rule group uses this information to monitor sign-in activities. For more information, see Adding the ACFP managed rule group to your web ACL.
     - In the **Rules** pane, open the **Override all rule actions** dropdown and choose **Count**. With this configuration, AWS WAF evaluates requests against all of the rules in the rule group and only counts the matches that result, while still adding labels to requests. For more information, see Overriding rule actions in a rule group.

       With this override, you can monitor the potential impact of the ACFP managed rules to determine whether you want to add exceptions, such as exceptions for internal use cases.
   - Position the rule group so that it's evaluated after your existing rules in the protection pack (web ACL), with a priority setting that's numerically higher than any rules or rule groups that you're already using. For more information, see Setting rule priority.

     This way, your current handling of traffic isn't disrupted. For example, if you have rules that detect malicious traffic such as SQL injection or cross-site scripting, they'll continue to detect and log that. Alternately, if you have rules that allow known non-malicious traffic, they can continue to allow that traffic, without having it blocked by the ACFP managed rule group. You might decide to adjust the processing order during your testing and tuning activities.

2. **Implement the application integration SDKs**

   Integrate the AWS WAF JavaScript SDK into your browser's account registration and account creation paths. AWS WAF also provides mobile SDKs to integrate iOS and Android devices. For more information about the integration SDKs, see Client application integrations in AWS WAF. For information about this recommendation, see Using application integration SDKs with ACFP.

   > **Note**
   >
   > If you are unable to use the application integration SDKs, it's possible to test the ACFP rule group by editing it in your protection pack (web ACL) and removing the override that you placed on the `AllRequests` rule. This enables the rule's Challenge action setting, to ensure that requests include a valid challenge token.
   > *Do this first in a test environment and then with great care in your production environment.* This approach has the potential to block users. For example, if your registration page path doesn't accept `GET` text/html requests, then this rule configuration can effectively block all requests at the registration page.

3. **Enable logging and metrics for the protection pack (web ACL)**

   As needed, configure logging, Amazon Security Lake data collection, request sampling, and Amazon CloudWatch metrics for the protection pack (web ACL). You can use these visibility tools to monitor the interaction of the ACFP managed rule group with your traffic.

   - For information about logging, see Logging AWS WAF protection pack (web ACL) traffic.
   - For information about Amazon Security Lake, see What is Amazon Security Lake? and Collecting data from AWS services in the *Amazon Security Lake user guide*.
   - For information about Amazon CloudWatch metrics, see Monitoring with Amazon CloudWatch.
   - For information about web request sampling, see Viewing a sample of web requests.

4. **Associate the protection pack (web ACL) with a resource**

   If the protection pack (web ACL) isn't already associated with a test resource, associate it. For information, see Associating or disassociating protection with an AWS resource.

5. **Monitor traffic and ACFP rule matches**

   Make sure that your normal traffic is flowing and that the ACFP managed rule group rules are adding labels to matching web requests. You can see the labels in the logs and see the ACFP and label metrics in the Amazon CloudWatch metrics. In the logs, the rules that you've overridden to count in the rule group show up in the `ruleGroupList` with `action` set to count, and with `overriddenAction` indicating the configured rule action that you overrode.

6. **Test the rule group's credential checking capabilities**

   Perform an account creation attempt with test compromised credentials and check that the rule group matches against them as expected.

   a. Access your protected resource's account registration page and try to add a new account. Use the following AWS WAF test credential pair and enter any test

      - User: `WAF_TEST_CREDENTIAL@wafexample.com`
      - Password: `WAF_TEST_CREDENTIAL_PASSWORD`

      These test credentials are categorized as compromised credentials, and the ACFP managed rule group will add the `awswaf:managed:aws:acfp:signal:credential_compromised` label to the account creation request, which you can see in the logs.

   b. In your protection pack (web ACL) logs, look for the `awswaf:managed:aws:acfp:signal:credential_compromised` label in the `labels` field on the log entries for your test account creation request. For information about logging, see Logging AWS WAF protection pack (web ACL) traffic.

   After you've verified that the rule group captures compromised credentials as expected, you can take steps to configure its implementation as you need for your protected resource.

7. **For CloudFront distributions, test the rule group's management of bulk account creation attempts**

   Run this test for each success response criteria that you configured for the ACFP rule group. Wait at least 30 minutes between tests.

   a. For each of your success criteria, identify an account creation attempt that will succeed with that success criteria in the response. Then, from a single client session, perform at

least 5 successful account creation attempts in under 30 minutes. A user would normally only create a single account on your site.

After the first successful account creation, the `VolumetricSessionSuccessfulResponse` rule should start matching against the rest of your account creation responses, labeling them and counting them, based on your rule action override. The rule might miss the first one or two due to latency.

b. In your protection pack (web ACL) logs, look for the `awswaf:managed:aws:acfp:aggregate:volumetric:session:successful_creation_` label in the `labels` field on the log entries for your test account creation web requests. For information about logging, see Logging AWS WAF protection pack (web ACL) traffic.

These tests verify that your success criteria match your responses by checking that the successful counts aggregated by the rule surpass the rule's threshold. After you've reached the threshold, if you continue to send account creation requests from the same session, the rule will continue to match until the success rate drops below the threshold. While the threshold is exceeded, the rule matches both successful or failed account creation attempts from the session address.

8. **Customize ACFP web request handling**

   As needed, add your own rules that explicitly allow or block requests, to change how ACFP rules would otherwise handle them.

   For example, you can use ACFP labels to allow or block requests or to customize request handling. You can add a label match rule after the ACFP managed rule group to filter labeled requests for the handling that you want to apply. After testing, keep the related ACFP rules in count mode, and maintain the request handling decisions in your custom rule. For an example, see ACFP example: Custom response for compromised credentials.

9. **Remove your test rules and enable the ACFP managed rule group settings**

   Depending on your situation, you might have decided that you want to leave some ACFP rules in count mode. For the rules that you want to run as configured inside the rule group, disable count mode in the protection pack (web ACL) rule group configuration. When you're finished testing, you can also remove your test label match rules.

10. **Monitor and tune**

    To be sure that web requests are being handled as you want, closely monitor your traffic after you enable the ACFP functionality that you intend to use. Adjust the behavior as needed with the rules count override on the rule group and with your own rules.

After you finish testing your ACFP rule group implementation, if you haven't already integrated the AWS WAF JavaScript SDK into your browser's account registration and account creation pages, we strongly recommend that you do so. AWS WAF also provides mobile SDKs to integrate iOS and Android devices. For more information about the integration SDKs, see Client application integrations in AWS WAF. For information about this recommendation, see Using application integration SDKs with ACFP.

AWS WAF Fraud Control account creation fraud prevention (ACFP) examples

This section shows example configurations that satisfy common use cases for the AWS WAF Fraud Control account creation fraud prevention (ACFP) implementations.

Each example provides a description of the use case and then shows the solution in JSON listings for the custom configured rules.

> **Note**
>
> You can retrieve JSON listings like the ones shown in these examples through the console protection pack (web ACL) JSON download or rule JSON editor, or through the `getWebACL` operation in the APIs and the command line interface.

Topics

- ACFP example: Simple configuration
- ACFP example: Custom response for compromised credentials
- ACFP example: Response inspection configuration

ACFP example: Simple configuration

The following JSON listing shows an example protection pack (web ACL) with an AWS WAF Fraud Control account creation fraud prevention (ACFP) managed rule group. Note the additional

`CreationPath` and `RegistrationPagePath` configurations, along with the payload type and the information needed to locate new account information in the payload, in order to verify it. The rule group uses this information to monitor and manage your account creation requests. This JSON includes the protection pack (web ACL)'s automatically generated settings, like the label namespace and the protection pack (web ACL)'s application integration URL.

```
{
  "Name": "simpleACFP",
  "Id": "... ",
  "ARN": "arn:aws:wafv2:us-east-1:111122223333:regional/webacl/simpleACFP/... ",
  "DefaultAction": {
    "Allow": {}
  },
  "Description": "",
  "Rules": [
    {
      "Name": "AWS-AWSManagedRulesACFPRuleSet",
      "Priority": 0,
      "Statement": {
        "ManagedRuleGroupStatement": {
          "VendorName": "AWS",
          "Name": "AWSManagedRulesACFPRuleSet",
          "ManagedRuleGroupConfigs": [
            {
              "AWSManagedRulesACFPRuleSet": {
                "CreationPath": "/web/signup/submit-registration",
                "RegistrationPagePath": "/web/signup/registration",
                "RequestInspection": {
                  "PayloadType": "JSON",
                  "UsernameField": {
                    "Identifier": "/form/username"
                  },
                  "PasswordField": {
                    "Identifier": "/form/password"
                  },
                  "EmailField": {
                    "Identifier": "/form/email"
                  },
                  "PhoneNumberFields": [
                    {
                      "Identifier": "/form/country-code"
                    },
                    {
```

```
                    "Identifier": "/form/region-code"
                  },
                  {
                    "Identifier": "/form/phonenumber"
                  }
                ],
                "AddressFields": [
                  {
                    "Identifier": "/form/name"
                  },
                  {
                    "Identifier": "/form/street-address"
                  },
                  {
                    "Identifier": "/form/city"
                  },
                  {
                    "Identifier": "/form/state"
                  },
                  {
                    "Identifier": "/form/zipcode"
                  }
                ]
              },
              "EnableRegexInPath": false
            }
          }
        ]
      }
    },
    "OverrideAction": {
      "None": {}
    },
    "VisibilityConfig": {
      "SampledRequestsEnabled": true,
      "CloudWatchMetricsEnabled": true,
      "MetricName": "AWS-AWSManagedRulesACFPRuleSet"
    }
  }
],
"VisibilityConfig": {
  "SampledRequestsEnabled": true,
  "CloudWatchMetricsEnabled": true,
  "MetricName": "simpleACFP"
```

```
  },
  "Capacity": 50,
  "ManagedByFirewallManager": false,
  "RetrofittedByFirewallManager": false,
  "LabelNamespace": "awswaf:111122223333:webacl:simpleACFP:"
}
```

ACFP example: Custom response for compromised credentials

By default, the credentials check that's performed by the rule group `AWSManagedRulesACFPRuleSet` handles compromised credentials by labeling the request and blocking it. For details about the rule group and rule behavior, see [AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group].

To inform the user that the account credentials they've provided have been compromised, you can do the following:

- **Override the `SignalCredentialCompromised` rule to Count** – This causes the rule to only count and label matching requests.
- **Add a label match rule with custom handling** – Configure this rule to match against the ACFP label and to perform your custom handling.

The following protection pack (web ACL) listings shows the ACFP managed rule group from the prior example, with the `SignalCredentialCompromised` rule action overridden to count. With this configuration, when this rule group evaluates any web request that uses compromised credentials, it will label the request, but not block it.

In addition, the protection pack (web ACL) now has a custom response named `aws-waf-credential-compromised` and a new rule named `AccountSignupCompromisedCredentialsHandling`. The rule priority is a higher numeric setting than the rule group, so it runs after the rule group in the protection pack (web ACL) evaluation. The new rule matches any request with the rule group's compromised credentials label. When the rule finds a match, it applies the Block action to the request with the custom response body. The custom response body provides information to the end user that their credentials have been compromised and proposes an action to take.

```
{
  "Name": "compromisedCreds",
  "Id": "... ",
```

```
  "ARN": "arn:aws:wafv2:us-east-1:111122223333:regional/webacl/compromisedCreds/...",
  "DefaultAction": {
    "Allow": {}
  },
  "Description": "",
  "Rules": [
    {
      "Name": "AWS-AWSManagedRulesACFPRuleSet",
      "Priority": 0,
      "Statement": {
        "ManagedRuleGroupStatement": {
          "VendorName": "AWS",
          "Name": "AWSManagedRulesACFPRuleSet",
          "ManagedRuleGroupConfigs": [
            {
              "AWSManagedRulesACFPRuleSet": {
                "CreationPath": "/web/signup/submit-registration",
                "RegistrationPagePath": "/web/signup/registration",
                "RequestInspection": {
                  "PayloadType": "JSON",
                  "UsernameField": {
                    "Identifier": "/form/username"
                  },
                  "PasswordField": {
                    "Identifier": "/form/password"
                  },
                  "EmailField": {
                    "Identifier": "/form/email"
                  },
                  "PhoneNumberFields": [
                    {
                      "Identifier": "/form/country-code"
                    },
                    {
                      "Identifier": "/form/region-code"
                    },
                    {
                      "Identifier": "/form/phonenumber"
                    }
                  ],
                  "AddressFields": [
                    {
                      "Identifier": "/form/name"
                    },
```

```
                    {
                      "Identifier": "/form/street-address"
                    },
                    {
                      "Identifier": "/form/city"
                    },
                    {
                      "Identifier": "/form/state"
                    },
                    {
                      "Identifier": "/form/zipcode"
                    }
                  ]
                },
                "EnableRegexInPath": false
              }
            }
          ],
          "RuleActionOverrides": [
            {
              "Name": "SignalCredentialCompromised",
              "ActionToUse": {
                "Count": {}
              }
            }
          ]
        }
      },
      "OverrideAction": {
        "None": {}
      },
      "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "AWS-AWSManagedRulesACFPRuleSet"
      }
    },
    {
      "Name": "AccountSignupCompromisedCredentialsHandling",
      "Priority": 1,
      "Statement": {
        "LabelMatchStatement": {
          "Scope": "LABEL",
          "Key": "awswaf:managed:aws:acfp:signal:credential_compromised"
```

```
        }
      },
      "Action": {
        "Block": {
          "CustomResponse": {
            "ResponseCode": 406,
            "CustomResponseBodyKey": "aws-waf-credential-compromised",
            "ResponseHeaders": [
              {
                "Name": "aws-waf-credential-compromised",
                "Value": "true"
              }
            ]
          }
        }
      },
      "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "AccountSignupCompromisedCredentialsHandling"
      }
    }
  ],
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "compromisedCreds"
  },
  "Capacity": 51,
  "ManagedByFirewallManager": false,
  "RetrofittedByFirewallManager": false,
  "LabelNamespace": "awswaf:111122223333:webacl:compromisedCreds:",
  "CustomResponseBodies": {
    "aws-waf-credential-compromised": {
      "ContentType": "APPLICATION_JSON",
      "Content": "{\n  \"credentials-compromised\": \"The credentials you provided have
 been found in a compromised credentials database.\\n\\nTry again with a different
 username, password pair.\"\n}"
    }
  }
}
```

ACFP example: Response inspection configuration

The following JSON listing shows an example protection pack (web ACL) with an AWS WAF Fraud Control account creation fraud prevention (ACFP) managed rule group that is configured to inspect origin responses. Note the response inspection configuration, which specifies success and response status codes. You can also configure success and response settings based on header, body, and body JSON matches. This JSON includes the protection pack (web ACL)'s automatically generated settings, like the label namespace and the protection pack (web ACL)'s application integration URL.

> **Note**
>
> ACFP response inspection is available only in protection packs (web ACLs) that protect CloudFront distributions. AWS WAF doesn't inspect responses for web requests that clients send over HTTP/3 (QUIC).

```
{
  "Name": "simpleACFP",
  "Id": "... ",
  "ARN": "arn:aws:wafv2:us-east-1:111122223333:regional/webacl/simpleACFP/... ",
  "DefaultAction": {
    "Allow": {}
  },
  "Description": "",
  "Rules": [
    {
      "Name": "AWS-AWSManagedRulesACFPRuleSet",
      "Priority": 0,
      "Statement": {
        "ManagedRuleGroupStatement": {
          "VendorName": "AWS",
          "Name": "AWSManagedRulesACFPRuleSet",
          "ManagedRuleGroupConfigs": [
            {
              "AWSManagedRulesACFPRuleSet": {
                "CreationPath": "/web/signup/submit-registration",
                "RegistrationPagePath": "/web/signup/registration",
                "RequestInspection": {
                  "PayloadType": "JSON",
                  "UsernameField": {
                    "Identifier": "/form/username"
```

```
      },
      "PasswordField": {
        "Identifier": "/form/password"
      },
      "EmailField": {
        "Identifier": "/form/email"
      },
      "PhoneNumberFields": [
        {
          "Identifier": "/form/country-code"
        },
        {
          "Identifier": "/form/region-code"
        },
        {
          "Identifier": "/form/phonenumber"
        }
      ],
      "AddressFields": [
        {
          "Identifier": "/form/name"
        },
        {
          "Identifier": "/form/street-address"
        },
        {
          "Identifier": "/form/city"
        },
        {
          "Identifier": "/form/state"
        },
        {
          "Identifier": "/form/zipcode"
        }
      ]
    },
    "ResponseInspection": {
      "StatusCode": {
        "SuccessCodes": [
          200
        ],
        "FailureCodes": [
          401
        ]
```

```
                }
              },
              "EnableRegexInPath": false
            }
          }
        ]
      }
    },
    "OverrideAction": {
      "None": {}
    },
    "VisibilityConfig": {
      "SampledRequestsEnabled": true,
      "CloudWatchMetricsEnabled": true,
      "MetricName": "AWS-AWSManagedRulesACFPRuleSet"
    }
  }
],
"VisibilityConfig": {
  "SampledRequestsEnabled": true,
  "CloudWatchMetricsEnabled": true,
  "MetricName": "simpleACFP"
},
"Capacity": 50,
"ManagedByFirewallManager": false,
"RetrofittedByFirewallManager": false,
"LabelNamespace": "awswaf:111122223333:webacl:simpleACFP:"
}
```

AWS WAF Fraud Control account takeover prevention (ATP)

This section explains what AWS WAF Fraud Control account takeover prevention (ATP) does.

Account takeover is an online illegal activity in which an attacker gains unauthorized access to a person's account. The attacker might do this in a number of ways, such as using stolen credentials or guessing the victim's password through a series of attempts. When the attacker gains access, they might steal money, information, or services from the victim. The attacker might pose as the victim to gain access to other accounts that the victim owns, or to gain access to the accounts of other people or organizations. Additionally, they might attempt to change the user's password in order to block the victim from their own accounts.

You can monitor and control account takeover attempts by implementing the ATP feature. AWS WAF offers this feature in the AWS Managed Rules rule group `AWSManagedRulesATPRuleSet` and companion application integration SDKs.

The ATP managed rule group labels and manages requests that might be part of malicious account takeover attempts. The rule group does this by inspecting login attempts that clients send to your application's login endpoint.

- **Request inspection** – ATP gives you visibility and control over anomalous login attempts and login attempts that use stolen credentials, to prevent account takeovers that might lead to fraudulent activity. ATP checks email and password combinations against its stolen credential database, which is updated regularly as new leaked credentials are found on the dark web. ATP aggregates data by IP address and client session, to detect and block clients that send too many requests of a suspicious nature.
- **Response inspection** – For CloudFront distributions, in addition to inspecting incoming login requests, the ATP rule group inspects your application's responses to login attempts, to track success and failure rates. Using this information, ATP can temporarily block client sessions or IP addresses that have too many login failures. AWS WAF performs response inspection asynchronously, so this doesn't increase latency in your web traffic. AWS WAF doesn't inspect responses for web requests that clients send over HTTP/3 (QUIC).

> **Note**
>
> You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

> **Note**
>
> The ATP feature is not available for Amazon Cognito user pools.

Topics

- AWS WAF ATP components
- Using application integration SDKs with ATP
- Adding the ATP managed rule group to your protection pack (web ACL)

- Testing and deploying ATP
- AWS WAF Fraud Control account takeover prevention (ATP) examples

AWS WAF ATP components

The primary components of AWS WAF Fraud Control account takeover prevention (ATP) are the following:

- **`AWSManagedRulesATPRuleSet`** – The rules in this AWS Managed Rules rule group detect, label, and handle various types of account takeover activity. The rule group inspects HTTP POST web requests that clients send to the specified login endpoint. For protected CloudFront distributions, the rule group also inspects the responses that the distribution sends back to these requests. For a list of the rule group's rules, see AWS WAF Fraud Control account takeover prevention (ATP) rule group. You include this rule group in your protection pack (web ACL) using a managed rule group reference statement. For information about using this rule group, see Adding the ATP managed rule group to your protection pack (web ACL).

  > **Note**
  >
  > You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

- **Details about your application's login page** – You must provide information about your login page when you add the `AWSManagedRulesATPRuleSet` rule group to your protection pack (web ACL). This lets the rule group narrow the scope of the requests it inspects and properly validate credentials usage in web requests. The ATP rule group works with usernames that are in email format. For more information, see Adding the ATP managed rule group to your protection pack (web ACL).
- **For protected CloudFront distributions, details about how your application responds to login attempts** – You provide details about your application's responses to login attempts, and the rule group tracks and manages clients that are sending too many failed login attempts. For information about configuring this option, see Adding the ATP managed rule group to your protection pack (web ACL).
- **JavaScript and mobile application integration SDKs** – Implement the AWS WAF JavaScript and mobile SDKs with your ATP implementation to enable the full set of capabilities that the rule group offers. Many of the ATP rules use the information provided by the SDKs for session level

client verification and behavior aggregation, required to separate legitimate client traffic from bot traffic. For more information about the SDKs, see Client application integrations in AWS WAF.

You can combine your ATP implementation with the following to help you monitor, tune, and customize your protections.

- **Logging and metrics** – You can monitor your traffic, and understand how the ACFP managed rule group affects it, by configuring and enabling logs, Amazon Security Lake data collection, and Amazon CloudWatch metrics for your protection pack (web ACL). The labels that `AWSManagedRulesATPRuleSet` adds to your web requests are included in the data. For information about the options, see Logging AWS WAF protection pack (web ACL) traffic, Monitoring with Amazon CloudWatch, and What is Amazon Security Lake?.

  Depending on your needs and the traffic that you see, you might want to customize your `AWSManagedRulesATPRuleSet` implementation. For example, you might want to exclude some traffic from ATP evaluation, or you might want to alter how it handles some of the account takeover attempts that it identifies, using AWS WAF features like scope-down statements or label matching rules.
- **Labels and label matching rules** – For any of the rules in `AWSManagedRulesATPRuleSet`, you can switch the blocking behavior to count, and then match against the labels that are added by the rules. Use this approach to customize how you handle web requests that are identified by the ATP managed rule group. For more information about labeling and using label match statements, see Label match rule statement and Web request labeling in AWS WAF.
- **Custom requests and responses** – You can add custom headers to the requests that you allow and you can send custom responses for requests that you block. To do this, you pair your label matching with the AWS WAF custom request and response features. For more information about customizing requests and responses, see Customized web requests and responses in AWS WAF.

Using application integration SDKs with ATP

This section explains how to use application integration SDKs with ATP.

The ATP managed rule group requires the challenge tokens that the application integration SDKs generate. The tokens enable the full set of protections that the rule group offers.

We highly recommend implementing the application integration SDKs, for the most effective use of the ATP rule group. The challenge script must run before the ATP rule group in order for the

rule group to benefit from the tokens that the script acquires. This happens automatically with the application integration SDKs. If you are unable to use the SDKs, you can alternately configure your protection pack (web ACL) so that it runs the Challenge or CAPTCHA rule action against all requests that will be inspected by the ATP rule group. Using the Challenge or CAPTCHA rule action can incur additional fees. For pricing details, see AWS WAF Pricing.

Capabilities of the ATP rule group that don't require a token

When web requests don't have a token, the ATP managed rule group is capable of blocking the following types of traffic:

- Single IP addresses that make a lot of login requests.
- Single IP addresses that make a lot of failed login requests in a short amount of time.
- Login attempts with password traversal, using the same username but changing passwords.

Capabilities of the ATP rule group that require a token

The information provided in the challenge token expands the capabilities of the rule group and of your overall client application security.

The token provides client information with each web request that enables the ATP rule group to separate legitimate client sessions from ill-behaved client sessions, even when both originate from a single IP address. The rule group uses information in the tokens to aggregate client session request behavior for fine-tuned detection and mitigation.

When the token is available in web requests, the ATP rule group can detect and block the following additional categories of clients at the session level:

- Client sessions that fail the silent challenge that the SDKs manage.
- Client sessions that traverse usernames or passwords. This is also known as credential stuffing.
- Client sessions that repeatedly use stolen credentials to log in.
- Client sessions that spend a long time trying to log in.
- Clients sessions that make a lot of login requests. The ATP rule group provides better client isolation than the AWS WAF rate-based rule, which can block clients by IP address. The ATP rule group also uses a lower threshold.
- Clients sessions that make a lot of failed login requests in a short amount of time. This functionality is available for protected Amazon CloudFront distributions.

For more information about the rule group capabilities see AWS WAF Fraud Control account takeover prevention (ATP) rule group.

For information about the SDKs, see Client application integrations in AWS WAF. For information about AWS WAF tokens, see Token use in AWS WAF intelligent threat mitigation. For information about the rule actions, see CAPTCHA and Challenge in AWS WAF.

Adding the ATP managed rule group to your protection pack (web ACL)

This section explains how to add and configure the `AWSManagedRulesATPRuleSet` rule group.

To configure the ATP managed rule group to recognize account takeover activities in your web traffic, you provide information about how clients send login requests to your application. For protected Amazon CloudFront distributions, you also provide information about how your application responds to login requests. This configuration is in addition to the normal configuration for a managed rule group.

For the rule group description and rules listing, see AWS WAF Fraud Control account takeover prevention (ATP) rule group.

> **Note**
>
> The ATP stolen credentials database only contains usernames in email format.

This guidance is intended for users who know generally how to create and manage AWS WAF protection packs (web ACLs), rules, and rule groups. Those topics are covered in prior sections of this guide. For basic information about how to add a managed rule group to your protection pack (web ACL), see Adding a managed rule group to a protection pack (web ACL) through the console.

Follow best practices

Use the ATP rule group in accordance with the best practices at Best practices for intelligent threat mitigation in AWS WAF.

To use the `AWSManagedRulesATPRuleSet` rule group in your protection pack (web ACL)

1. Add the AWS managed rule group, `AWSManagedRulesATPRuleSet` to your protection pack (web ACL), and **Edit** the rule group settings before saving.

> **Note**
>
> You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

2. In the **Rule group configuration** pane, provide the information that the ATP rule group uses to inspect login requests.

   a. For **Use regular expression in paths**, toggle this on if you want AWS WAF to perform regular expression matching for your login page path specifications.

      AWS WAF supports the pattern syntax used by the PCRE library `libpcre` with some exceptions. The library is documented at PCRE - Perl Compatible Regular Expressions. For information about AWS WAF support, see Supported regular expression syntax in AWS WAF.

   b. For **Login path**, provide the path of the login endpoint for your application. The rule group inspects only HTTP `POST` requests to your specified login endpoint.

      > **Note**
      >
      > Matching for endpoints is case insensitive. Regex specifications must not contain the flag `(?-i)`, which disables case insensitive matching. String specifications must start with a forward slash `/`.

      For example, for the URL `https://example.com/web/login`, you could provide the string path specification `/web/login`. Login paths that start with the path that you provide are considered a match. For example `/web/login` matches the login paths `/web/login`, `/web/login/`, `/web/loginPage`, and `/web/login/thisPage`, but doesn't match the login path `/home/web/login` or `/website/login`.

   c. For **Request inspection**, specify how your application accepts login attempts by providing the request payload type and the names of the fields within the request body where the username and password are provided. Your specification of the field names depends on the payload type.

- **JSON payload type** – Specify the field names in JSON pointer syntax. For information about the JSON Pointer syntax, see the Internet Engineering Task Force (IETF) documentation JavaScript Object Notation (JSON) Pointer.

  For example, for the following example JSON payload, the username field specification is `/login/username` and the password field specification is `/login/password`.

  ```
  {
      "login": {
          "username": "THE_USERNAME",
          "password": "THE_PASSWORD"
      }
  }
  ```

- **FORM_ENCODED payload type** – Use the HTML form names.

  For example, for an HTML form with input elements named `username1` and `password1`, the username field specification is `username1` and the password field specification is `password1`.

d. If you're protecting Amazon CloudFront distributions, then under **Response inspection**, specify how your application indicates success or failure in its responses to login attempts.

> **Note**
>
> ATP response inspection is available only in protection packs (web ACLs) that protect CloudFront distributions. AWS WAF doesn't inspect responses for web requests that clients send over HTTP/3 (QUIC).

Specify a single component in the login response that you want ATP to inspect. For the **Body** and **JSON** component types, AWS WAF can inspect the first 65,536 bytes (64 KB) of the component.

Provide your inspection criteria for the component type, as indicated by the interface. You must provide both success and failure criteria to inspect for in the component.

For example, say your application indicates the status of a login attempt in the status code of the response, and uses `200 OK` for success and `401 Unauthorized` or `403 Forbidden` for failure. You would set the response inspection **Component type** to **Status**

code, then in the **Success** text box enter `200` and in the **Failure** text box, enter `401` on the first line and `403` on the second.

The ATP rule group only counts responses that match your success or failure inspection criteria. The rule group rules act on clients while they have too high a failure rate among the responses that are counted. For accurate behavior by the rule group rules, be sure to provide complete information for both successful and failed login attempts.

To see the rules that inspect login responses, look for `VolumetricIpFailedLoginResponseHigh` and `VolumetricSessionFailedLoginResponseHigh` in the rules listing at AWS WAF Fraud Control account takeover prevention (ATP) rule group.

3. Provide any additional configuration that you want for the rule group.

   You can further limit the scope of requests that the rule group inspects by adding a scope-down statement to the managed rule group statement. For example, you can inspect only requests with a specific query argument or cookie. The rule group will inspect only HTTP POST requests to your specified login endpoint that match the criteria in your scope-down statement. For information about scope-down statements, see Using scope-down statements in AWS WAF.

4. Save your changes to the protection pack (web ACL).

Before you deploy your ATP implementation for production traffic, test and tune it in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune the rules in count mode with your production traffic before enabling them. See the section that follows for guidance.

Testing and deploying ATP

This section provides general guidance for configuring and testing an AWS WAF Fraud Control account takeover prevention (ATP) implementation for your site. The specific steps that you choose to follow will depend on your needs, resources, and web requests that you receive.

This information is in addition to the general information about testing and tuning provided at Testing and tuning your AWS WAF protections.

Note

AWS Managed Rules are designed to protect you from common web threats. When used in accordance with the documentation, AWS Managed Rules rule groups add another layer of security for your applications. However, AWS Managed Rules rule groups aren't intended as a replacement for your security responsibilities, which are determined by the AWS resources that you select. Refer to the Shared Responsibility Model to ensure that your resources in AWS are properly protected.

Production traffic risk

Before you deploy your ATP implementation for production traffic, test and tune it in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune the rules in count mode with your production traffic before enabling them.

AWS WAF provides test credentials that you can use to verify your ATP configuration. In the following procedure, you'll configure a test protection pack (web ACL) to use the ATP managed rule group, configure a rule to capture the label added by the rule group, and then run a login attempt using these test credentials. You'll verify that your web ACL has properly managed the attempt by checking the Amazon CloudWatch metrics for the login attempt.

This guidance is intended for users who know generally how to create and manage AWS WAF protection packs (web ACLs), rules, and rule groups. Those topics are covered in prior sections of this guide.

To configure and test an AWS WAF Fraud Control account takeover prevention (ATP) implementation

Perform these steps first in a test environment, then in production.

1. **Add the AWS WAF Fraud Control account takeover prevention (ATP) managed rule group in count mode**

   > **Note**
   >
   > You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

   Add the AWS Managed Rules rule group `AWSManagedRulesATPRuleSet` to a new or existing protection pack (web ACL) and configure it so that it doesn't alter the current protection pack (web ACL) behavior. For details about the rules and labels for this rule group, see AWS WAF Fraud Control account takeover prevention (ATP) rule group.

   - When you add the managed rule group, edit it and do the following:
     - In the **Rule group configuration** pane, provide the details of your application's login page. The ATP rule group uses this information to monitor sign-in activities. For more information, see Adding the ATP managed rule group to your protection pack (web ACL).
     - In the **Rules** pane, open the **Override all rule actions** dropdown and choose **Count**. With this configuration, AWS WAF evaluates requests against all of the rules in the rule group and only counts the matches that result, while still adding labels to requests. For more information, see Overriding rule actions in a rule group.

       With this override, you can monitor the potential impact of the ATP managed rules to determine whether you want to add exceptions, such as exceptions for internal use cases.
   - Position the rule group so that it's evaluated after your existing rules in the protection pack (web ACL), with a priority setting that's numerically higher than any rules or rule groups that you're already using. For more information, see Setting rule priority.

     This way, your current handling of traffic isn't disrupted. For example, if you have rules that detect malicious traffic such as SQL injection or cross-site scripting, they'll continue to detect and log that. Alternately, if you have rules that allow known non-malicious traffic, they can continue to allow that traffic, without having it blocked by the ATP managed rule group. You might decide to adjust the processing order during your testing and tuning activities.

2. **Enable logging and metrics for the protection pack (web ACL)**

   As needed, configure logging, Amazon Security Lake data collection, request sampling, and Amazon CloudWatch metrics for the protection pack (web ACL). You can use these visibility tools to monitor the interaction of the ATP managed rule group with your traffic.

   - For information about configuring and using logging, see Logging AWS WAF protection pack (web ACL) traffic.
   - For information about Amazon Security Lake, see What is Amazon Security Lake? and Collecting data from AWS services in the *Amazon Security Lake user guide*.
   - For information about Amazon CloudWatch metrics, see Monitoring with Amazon CloudWatch.
   - For information about web request sampling, see Viewing a sample of web requests.

3. **Associate the protection pack (web ACL) with a resource**

   If the protection pack (web ACL) isn't already associated with a test resource, associate it. For information, see Associating or disassociating protection with an AWS resource.

4. **Monitor traffic and ATP rule matches**

   Make sure that your normal traffic is flowing and that the ATP managed rule group rules are adding labels to matching web requests. You can see the labels in the logs and see the ATP and label metrics in the Amazon CloudWatch metrics. In the logs, the rules that you've overridden to count in the rule group show up in the `ruleGroupList` with `action` set to count, and with `overriddenAction` indicating the configured rule action that you overrode.

5. **Test the rule group's credential checking capabilities**

   Perform a login attempt with test compromised credentials and check that the rule group matches against them as expected.

   a. Log in to your protected resource's login page using the following AWS WAF test credential pair:

      - User: `WAF_TEST_CREDENTIAL@wafexample.com`
      - Password: `WAF_TEST_CREDENTIAL_PASSWORD`

      These test credentials are categorized as compromised credentials, and the ATP managed rule group will add the

`awswaf:managed:aws:atp:signal:credential_compromised` label to the login request, which you can see in the logs.

b. In your protection pack (web ACL) logs, look for the `awswaf:managed:aws:atp:signal:credential_compromised` label in the `labels` field on the log entries for your test login web requests. For information about logging, see Logging AWS WAF protection pack (web ACL) traffic.

After you've verified that the rule group captures compromised credentials as expected, you can take steps to configure its implementation as you need for your protected resource.

6. **For CloudFront distributions, test the rule group's login failure management**

a. Run a test for each failure response criteria that you configured for the ATP rule group. Wait at least 10 minutes between tests.

To test a single failure criteria, identify a login attempt that will fail with that criteria in the response. Then, from a single client IP address, perform at least 10 failed login attempts in under 10 minutes.

After the first 6 failures, the volumetric failed login rule should start matching against the rest of your attempts, labeling and counting them. The rule might miss the first one or two due to latency.

b. In your protection pack (web ACL) logs, look for the `awswaf:managed:aws:atp:aggregate:volumetric:ip:failed_login_response:high` label in the `labels` field on the log entries for your test login web requests. For information about logging, see Logging AWS WAF protection pack (web ACL) traffic.

These tests verify that your failure criteria match your responses by checking that the failed login counts surpass the thresholds for the rule `VolumetricIpFailedLoginResponseHigh`. After you've reached the thresholds, if you continue to send login requests from the same IP address, the rule will continue to match until the failure rate drops below the threshold. While the thresholds are exceeded, the rule matches both successful or failed logins from the IP address.

7. **Customize ATP web request handling**

As needed, add your own rules that explicitly allow or block requests, to change how ATP rules would otherwise handle them.

For example, you can use ATP labels to allow or block requests or to customize request handling. You can add a label match rule after the ATP managed rule group to filter labeled requests for the handling that you want to apply. After testing, keep the related ATP rules in count mode, and maintain the request handling decisions in your custom rule. For an example, see ATP example: Custom handling for missing and compromised credentials.

8. **Remove your test rules and enable the ATP managed rule group settings**

   Depending on your situation, you might have decided that you want to leave some ATP rules in count mode. For the rules that you want to run as configured inside the rule group, disable count mode in the protection pack (web ACL) rule group configuration. When you're finished testing, you can also remove your test label match rules.

9. **Monitor and tune**

   To be sure that web requests are being handled as you want, closely monitor your traffic after you enable the ATP functionality that you intend to use. Adjust the behavior as needed with the rules count override on the rule group and with your own rules.

After you finish testing your ATP rule group implementation, if you haven't already done so, we strongly recommend that you integrate the AWS WAF JavaScript SDK into your browser login page, for enhanced detection capabilities. AWS WAF also provides mobile SDKs to integrate iOS and Android devices. For more information about the integration SDKs, see Client application integrations in AWS WAF. For information about this recommendation, see Using application integration SDKs with ATP.

AWS WAF Fraud Control account takeover prevention (ATP) examples

This section shows example configurations that satisfy common use cases for the AWS WAF Fraud Control account takeover prevention (ATP) implementations.

Each example provides a description of the use case and then shows the solution in JSON listings for the custom configured rules.

> **Note**
>
> You can retrieve JSON listings like the ones shown in these examples through the console protection pack (web ACL) JSON download or rule JSON editor, or through the `getWebACL` operation in the APIs and the command line interface.

Topics

- ATP example: Simple configuration
- ATP example: Custom handling for missing and compromised credentials
- ATP example: Response inspection configuration

ATP example: Simple configuration

The following JSON listing shows an example protection pack (web ACL) with an AWS WAF Fraud Control account takeover prevention (ATP) managed rule group. Note the additional sign-in page configuration, which gives the rule group the information it needs to monitor and manage your login requests. This JSON includes the protection pack (web ACL)'s automatically generated settings, like the label namespace and the protection pack (web ACL)'s application integration URL.

```
{
    "WebACL": {
        "LabelNamespace": "awswaf:111122223333:webacl:ATPModuleACL:",
        "Capacity": 50,
        "Description": "This is a test protection pack (web ACL) for ATP.",
        "Rules": [
            {
                "Priority": 1,
                "OverrideAction": {
                    "None": {}
                },
                "VisibilityConfig": {
                    "SampledRequestsEnabled": true,
                    "CloudWatchMetricsEnabled": true,
                    "MetricName": "AccountTakeOverValidationRule"
                },
                "Name": "DetectCompromisedUserCredentials",
                "Statement": {
                    "ManagedRuleGroupStatement": {
                        "VendorName": "AWS",
                        "Name": "AWSManagedRulesATPRuleSet",
                        "ManagedRuleGroupConfigs": [
                          {
                            "AWSManagedRulesATPRuleSet": {
                              "LoginPath": "/web/login",
                              "RequestInspection": {
                                "PayloadType": "JSON",
```

```
                                "UsernameField": {
                                  "Identifier": "/form/username"
                                },
                                "PasswordField": {
                                  "Identifier": "/form/password"
                                }
                              },
                              "EnableRegexInPath": false
                            }
                          }
                        ]
                    }
                }
            }
        ],
        "VisibilityConfig": {
            "SampledRequestsEnabled": true,
            "CloudWatchMetricsEnabled": true,
            "MetricName": "ATPValidationAcl"
        },
        "DefaultAction": {
            "Allow": {}
        },
        "ManagedByFirewallManager": false,
        "RetrofittedByFirewallManager": false,
        "Id": "32q10987-65rs-4tuv-3210-98765wxyz432",
        "ARN": "arn:aws:wafv2:us-east-1:111122223333:regional/webacl/
ATPModuleACL/32q10987-65rs-4tuv-3210-98765wxyz432",
        "Name": "ATPModuleACL"
    },
    "ApplicationIntegrationURL": "https://9z87abce34ea.us-
east-1.sdk.awswaf.com/9z87abce34ea/1234567a1b10/",
    "LockToken": "6d0e6966-95c9-48b6-b51d-8e82e523b847"
}
```

ATP example: Custom handling for missing and compromised credentials

By default, the credentials checks that are performed by the rule group `AWSManagedRulesATPRuleSet` handle web requests as follows:

- **Missing credentials** – Label and block request.
- **Compromised credentials** – Label request but don't block or count it.

For details about the rule group and rule behavior, see AWS WAF Fraud Control account takeover prevention (ATP) rule group.

You can add custom handling for web requests that have missing or compromised credentials by doing the following:

- **Override the `MissingCredential` rule to Count** – This rule action override causes the rule to only count and label matching requests.
- **Add a label match rule with custom handling** – Configure this rule to match against both of the ATP labels and to perform your custom handling. For example, you might redirect the customer to your sign-up page.

The following rule shows the ATP managed rule group from the prior example, with the `MissingCredential` rule action overridden to count. This causes the rule to apply its label to matching requests, and then only count the requests, instead of blocking them.

```
"Rules": [
    {
        "Priority": 1,
        "OverrideAction": {
            "None": {}
        },
        "VisibilityConfig": {
            "SampledRequestsEnabled": true,
            "CloudWatchMetricsEnabled": true,
            "MetricName": "AccountTakeOverValidationRule"
        },
        "Name": "DetectCompromisedUserCredentials",
        "Statement": {
            "ManagedRuleGroupStatement": {
                "ManagedRuleGroupConfigs": [
                  {
                    "AWSManagedRulesATPRuleSet": {
                      "LoginPath": "/web/login",
                      "RequestInspection": {
                        "PayloadType": "JSON",
                        "UsernameField": {
                          "Identifier": "/form/username"
                        },
                        "PasswordField": {
                          "Identifier": "/form/password"
```

```
                        }
                      },
                      "EnableRegexInPath": false
                    }
                  }
                ]
                "VendorName": "AWS",
                "Name": "AWSManagedRulesATPRuleSet",
                "RuleActionOverrides": [
                  {
                    "ActionToUse": {
                      "Count": {}
                    },
                    "Name": "MissingCredential"
                  }
                ],
                "ExcludedRules": []
            }
        }
    }
],
```

With this configuration, when this rule group evaluates any web request that has missing or compromised credentials, it will label the request, but not block it.

The following rule has a priority setting that is higher numerically than the preceding rule group. AWS WAF evaluates rules in numeric order, starting from the lowest, so this rule will be evaluated after the rule group evaluation. The rule is configured to match either of the credentials labels and to send a custom response for matching requests.

```
"Name": "redirectToSignup",
      "Priority": 10,
      "Statement": {
        "OrStatement": {
          "Statements": [
            {
              "LabelMatchStatement": {
                "Scope": "LABEL",
                "Key": "awswaf:managed:aws:atp:signal:missing_credential"
              }
            },
            {
              "LabelMatchStatement": {
```

```
                "Scope": "LABEL",
                "Key": "awswaf:managed:aws:atp:signal:credential_compromised"
              }
            }
          ]
        }
      },
      "Action": {
        "Block": {
          "CustomResponse": {
             your custom response settings
          }
        }
      },
      "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "redirectToSignup"
      }
```

ATP example: Response inspection configuration

The following JSON listing shows an example protection pack (web ACL) with an AWS WAF Fraud Control account takeover prevention (ATP) managed rule group that is configured to inspect origin responses. Note the response inspection configuration, which specifies success and response status codes. You can also configure success and response settings based on header, body, and body JSON matches. This JSON includes the protection pack (web ACL)'s automatically generated settings, like the label namespace and the protection pack (web ACL)'s application integration URL.

> **Note**
>
> ATP response inspection is available only in protection packs (web ACLs) that protect CloudFront distributions. AWS WAF doesn't inspect responses for web requests that clients send over HTTP/3 (QUIC).

```
{
    "WebACL": {
        "LabelNamespace": "awswaf:111122223333:webacl:ATPModuleACL:",
        "Capacity": 50,
```

```
        "Description": "This is a test protection pack (web ACL) for ATP.",
        "Rules": [
            {
                "Priority": 1,
                "OverrideAction": {
                    "None": {}
                },
                "VisibilityConfig": {
                    "SampledRequestsEnabled": true,
                    "CloudWatchMetricsEnabled": true,
                    "MetricName": "AccountTakeOverValidationRule"
                },
                "Name": "DetectCompromisedUserCredentials",
                "Statement": {
                    "ManagedRuleGroupStatement": {
                        "VendorName": "AWS",
                        "Name": "AWSManagedRulesATPRuleSet",
                        "ManagedRuleGroupConfigs": [
                          {
                            "AWSManagedRulesATPRuleSet": {
                              "LoginPath": "/web/login",
                              "RequestInspection": {
                                "PayloadType": "JSON",
                                "UsernameField": {
                                  "Identifier": "/form/username"
                                },
                                "PasswordField": {
                                  "Identifier": "/form/password"
                                }
                              },
                              "ResponseInspection": {
                                "StatusCode": {
                                  "SuccessCodes": [
                                    200
                                  ],
                                  "FailureCodes": [
                                    401
                                  ]
                                }
                              },
                              "EnableRegexInPath": false
                            }
                          }
                        ]
```

```
                    }
                }
            }
        ],
        "VisibilityConfig": {
            "SampledRequestsEnabled": true,
            "CloudWatchMetricsEnabled": true,
            "MetricName": "ATPValidationAcl"
        },
        "DefaultAction": {
            "Allow": {}
        },
        "ManagedByFirewallManager": false,
        "RetrofittedByFirewallManager": false,
        "Id": "32q10987-65rs-4tuv-3210-98765wxyz432",
        "ARN": "arn:aws:wafv2:us-east-1:111122223333:regional/webacl/
ATPModuleACL/32q10987-65rs-4tuv-3210-98765wxyz432",
        "Name": "ATPModuleACL"
    },
    "ApplicationIntegrationURL": "https://9z87abce34ea.us-
east-1.sdk.awswaf.com/9z87abce34ea/1234567a1b10/",
    "LockToken": "6d0e6966-95c9-48b6-b51d-8e82e523b847"
}
```

AWS WAF Bot Control

This section explains what Bot Control does.

With Bot Control, you can easily monitor, block, or rate limit bots such as scrapers, scanners, crawlers, status monitors, and search engines. If you use the targeted inspection level of the rule group, you can also challenge bots that don't self identify, making it harder and more expensive for malicious bots to operate against your website. You can protect your applications using the Bot Control managed rule group alone, or in combination with other AWS Managed Rules rule groups and your own custom AWS WAF rules.

Bot Control also provides AI traffic monetization, which lets you charge AI bots and agents for content access. When you add rules with the Monetize action, you can use Bot Control labels to differentiate pricing based on bot identity and verification status. For more information, see AI traffic monetization.

Important

To implement your AI traffic monetization policies, we use multiple detection techniques such as behavioral signals and risk-based systems to inspect and categorize inbound traffic. While these methods are designed to provide high-confidence classification, they are probabilistic and might not correctly identify or categorize all bot traffic in all cases. We continuously test and update our analysis methods to increase accuracy. We recommend using Test mode to validate that your policies produce the expected results before enabling live monetization.

Monetize

Returns an HTTP 402 Payment Required Challenge containing payment instructions. The client can complete payment and resubmit the request to gain access. Use the Monetize action to charge AI bots for content access. The Monetize action is available only for web ACLs associated with Amazon CloudFront distributions. To use this action, your web ACL must have a MonetizationConfig configured. For more information, see AI traffic monetization.

Monetize action parameters

Parameter	Required	Description
PriceMultiplier	No	Integer multiplier (1–100) applied to the base price. Default: 1.

Bot Control includes a console dashboard that shows how much of your current traffic is coming from bots, based on request sampling. With the Bot Control managed rule group added to your protection pack (web ACL), you can take action against bot traffic and receive detailed, real-time information about common bot traffic coming to your applications.

Note

You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

The Bot Control managed rule group provides a basic, common protection level that adds labels to self-identifying bots, verifies generally desirable bots, and detects high confidence bot signatures. This gives you the ability to monitor and control common categories of bot traffic.

The Bot Control rule group also provides a targeted protection level that adds detection for sophisticated bots that don't self identify. Targeted protections use detection techniques such as browser interrogation, fingerprinting, and behavior heuristics to identify bad bot traffic. Additionally, targeted protections provide optional automated, machine-learning analysis of website traffic statistics to detect bot-related activity. When you enable machine learning, AWS WAF uses statistics about website traffic, such as timestamps, browser characteristics, and previous URL visited, to improve the Bot Control machine learning model.

When AWS WAF evaluates a web request against the Bot Control managed rule group, it labels requests that it detects as bot-related. Labels include the bot category and bot name. You can match against these labels in your own AWS WAF rules to customize handling. The labels that are generated by the Bot Control managed rule group are included in Amazon CloudWatch metrics and your protection pack (web ACL) logs.

You can also use AWS Firewall Manager AWS WAF policies to deploy the Bot Control managed rule group across your applications in multiple accounts that are part of your organization in AWS Organizations.

For more information about the Bot Control managed rule group, see AWS WAF Bot Control rule group.

For guidance on choosing the right protection level and configuring Bot Control for specific application scenarios, see Choosing and configuring Bot Control for your use case.

Web bot authentication for AI agents

AWS WAF Bot Control now supports Web Bot Authentication (WBA) as a cryptographic verification method for bots and AI agents. WBA applies to your CloudFront distributions and Regional resources in the commercial AWS Regions. This feature enables legitimate AI crawlers and agents to prove their identity without requiring traditional challenge-response mechanisms.

Version requirement: `AWSManagedRulesBotControlRuleSet` Version_4.0 or later. (The static version must be explicitly selected.) For detailed label taxonomy and rule behavior, see:

- AWS WAF Bot Control rule group
- Web request labeling in AWS WAF

- AWS Managed Rules changelog

Choosing and configuring Bot Control for your use case

Bot Control protects against a range of automated threats. The right configuration depends on what you're protecting and what threats you face. Use this topic to choose the right protection level and configure Bot Control for common application scenarios.

Matching Bot Control to your application

Bot threats generally fall into three categories:

- **Fraud threats** – Credential stuffing, fake accounts, and automated purchasing.
- **Content threats** – Scraping of pricing data, product catalogs, and published content.
- **Availability threats** – Bot traffic volume that degrades performance or increases infrastructure costs.

For most of these threats, we recommend the targeted protection level with the client application integration SDKs. The following scenarios describe how to configure Bot Control for each.

Protecting login and account pages

Credential stuffing and account takeover attacks use automated tools to test stolen credentials against your login endpoints. Account creation fraud uses bots to create fake accounts at scale. Integrate the application integration SDK across your site, and pair Bot Control with the AWS WAF Fraud Control managed rule groups for login and account creation protection.

Protecting product catalogs and pricing data

If your application displays product information, pricing, inventory levels, or other competitive data, bots may scrape this content to feed competitor intelligence, build comparison sites, or undercut your pricing. These scrapers often mimic real browsers and rotate through residential IP addresses to avoid detection. Apply Bot Control across your product and catalog pages, not just checkout. This way Bot Control can detect the behavioral patterns of automated scraping even when the bot appears to be a normal browser.

Protecting published content

Content publishers, news sites, and media platforms face bots that copy articles, images, and other original content. This content theft can dilute your search rankings, reduce ad revenue, and

undermine your competitive position. Apply Bot Control across your content pages. Use the Bot Control dashboard to monitor which bots are accessing your content and decide how to handle each category. For more information, see AWS WAF Bot Control components.

Reducing infrastructure costs from unwanted crawlers

Scrapers, crawlers, and automated tools can generate high traffic volume, increasing your compute and data transfer costs without providing business value. Bot Control identifies self-declaring bots by category, such as scrapers, HTTP libraries, and crawling frameworks. You can then block or rate-limit each category independently using labels. For evasive scraping, targeted protection detects bots that disguise their identity, including those using residential proxies and headless browsers.

Protecting high-value transactions

Checkout flows, limited-inventory purchases, and ticket sales are targets for automated purchasing bots that compete with legitimate customers. Bot Control detects automated browsers such as Selenium and Puppeteer, and applies session-level rate limiting to prevent a single client from monopolizing inventory. Integrate the SDKs on your transaction pages for the strongest detection accuracy.

Managing AI crawlers and training data scrapers

AI bots collect content from your application to train models or surface data in AI applications. Bot Control categorizes known AI bots. Write custom rules to allow specific AI bots you want to permit, such as verified search engine crawlers, and block or rate-limit others. For AI agents that use browser automation frameworks to interact with your application, Bot Control detects and challenges these automated sessions. With Web Bot Authentication (WBA), legitimate AI agents can cryptographically prove their identity. This gives you a reliable signal to distinguish authorized agents from unauthorized ones. For more information about WBA, see Web bot authentication for AI agents.

Protecting mobile applications

Mobile applications face bot threats similar to web applications, but their traffic has different characteristics. Mobile clients use non-browser user agents. HTTP requests from native mobile application frameworks are excluded from the Bot Control `SignalNonBrowserUserAgent` rule, but non-standard HTTP libraries and in-app browsers are not. Integrate the AWS WAF Mobile SDK into your iOS or Android application to provide the client-side tokens that targeted protection uses for accurate detection.

Choosing between common and targeted protection

Bot Control offers two protection levels. The following guidance helps you choose the right level for your application.

Common protection level

Common protection detects bots that identify themselves through their user-agent strings, IP addresses, and other request characteristics. It verifies that bots claiming to be from known organizations (such as search engines) actually originate from those organizations. Common protection gives you visibility into who is visiting your application and lets you control each bot category independently. It does not require the SDKs and has a lower per-request cost. Common protection is a good starting point when you primarily need to manage known bot traffic, such as blocking unwanted scrapers while allowing search engine crawlers.

Targeted protection level

The targeted protection level detects everything that the common protection level detects, and adds detection for bots that don't self-identify. It uses browser interrogation, TLS fingerprinting, behavioral heuristics, and machine learning to distinguish automated clients from humans. The machine learning analyzes your specific website's traffic patterns, including timestamps, browser characteristics, and navigation behavior. It updates its detection as your traffic changes and as bot tactics change. Targeted protection is recommended for applications facing credential stuffing, advanced scraping, automated purchasing, or any bot activity where the attacker actively tries to evade detection.

For most production web applications, we recommend targeted protection. For guidance on configuring targeted protection for specific scenarios, see Matching Bot Control to your application. For guidance on managing costs at high traffic volumes, see Managing costs.

	Common	Targeted
Detects self-identifying bots	Yes	Yes
Verifies legitimate bots (search engines, monitoring services)	Yes	Yes
Detects bots that hide their identity	No	Yes
Machine learning adapted to your traffic	No	Yes

	Common	Targeted
Session-level rate limiting	No	Yes
Detects browser automation tools	Yes (self-identifying)	Yes (including evasive)
Client application integration SDKs	Not required	Strongly recommend ed

Checking your rule group version

The Bot Control managed rule group is updated over time with new detection capabilities. Check which version you are currently using and whether a newer static version is available. Newer versions include additional detections that can increase your coverage against new bot threats. You can review the available versions and their changes in the AWS Managed Rules changelog.

How Bot Control detection works

Bot Control labels every request it evaluates with granular classification: bot name, category, organization, and verification status. You write rules that match on these labels to decide what action to take for each type of bot. This gives you full control rather than a single allow-or-block decision for all bot traffic.

At the targeted protection level, Bot Control combines multiple detection techniques. These include signature matching, browser interrogation, TLS fingerprinting, behavioral heuristics, and machine learning tuned to your website's traffic patterns. A bot that evades one technique may be caught by another. The machine learning also detects coordinated bot activity: patterns where multiple clients work together in ways that individual request analysis would miss. For a detailed description of all Bot Control components, see AWS WAF Bot Control components. For a walkthrough of how the Challenge and CAPTCHA interactions work between the client and AWS WAF, see CAPTCHA and Challenge in AWS WAF.

Managing costs

Your costs for using the Bot Control managed rule group increase with the number of web requests that AWS WAF evaluates with it. For most applications, the cost of Bot Control is justified by the protection it provides. Blocking bot traffic reduces your compute, bandwidth, and data transfer

costs, often by more than the cost of the inspection itself. If you need to optimize costs further, the following strategies can help.

Exclude static assets

The simplest cost optimization is to exclude requests for static file types such as `.css`, `.png`, `.jpg`, and `.svg` from Bot Control inspection. Bots that target your application data and functionality hit dynamic endpoints, not your stylesheets or images. This reduces inspected requests without reducing detection effectiveness. For an example, see Bot Control example: Using Bot Control only for dynamic content.

Order rules for efficiency

Place less expensive rules before the Bot Control managed rule group in your protection pack (web ACL). Rules such as IP reputation lists, geographic restrictions, and rate-based rules can block clearly unwanted traffic before it reaches the paid Bot Control inspection. Requests that are blocked by earlier rules are never evaluated by Bot Control, so you don't incur the additional per-request fee for them.

Scope to specific endpoints when appropriate

For some applications, you may want to apply Bot Control only to specific parts of your site. For example, if your application has a public informational section that doesn't contain sensitive data or valuable content, you can exclude it from inspection. Be careful not to exclude pages that contain content worth protecting, such as product catalogs, pricing data, or published articles. For an example, see Bot Control example: Using Bot Control only for the login page.

Integrating the client application integration SDKs

Use the JavaScript and Mobile SDKs to maximize the effectiveness of targeted protection. The SDKs provide browser fingerprinting, challenge token management, and behavioral telemetry that targeted rules use for session-level detection. A session here is identified by the client token that the SDK acquires, which represents a specific browser or device. Without the SDKs, targeted protection has much less context to work with, and cannot reliably distinguish advanced bots from legitimate users.

Integrate the SDK from the start

When you deploy targeted protection, integrate the SDKs at the same time. This applies whether you are deploying to production or evaluating Bot Control in a test environment. Evaluating

targeted protection without the SDKs does not give you an accurate picture of its detection capabilities, because many of the targeted rules depend on the client-side signals that only the SDKs provide.

Broad integration recommended

For the strongest detection, integrate the JavaScript SDK across all pages that serve dynamic content, not just login or checkout. Bot Control builds behavioral profiles from how clients navigate your application, including request timing, page sequences, and interaction patterns. When the SDK is present on only a single page, Bot Control has limited behavioral context to distinguish a real user from a bot that has learned to mimic one page load. Broader integration gives Bot Control a richer picture of each client session, which improves detection of advanced bots such as residential proxy botnets. At minimum, integrate the SDK on your most critical pages, but treat this as a starting point rather than the end state.

Mobile applications

For iOS and Android applications, use the AWS WAF Mobile SDK. The Mobile SDK handles token acquisition and renewal within your app. For more information about the SDKs, see Client application integrations in AWS WAF.

Common configuration adjustments

Most Bot Control deployments require some configuration adjustments to match the specific characteristics of your application traffic. The following are the most common adjustments.

Allowing monitoring and health-check tools

Internal monitoring tools, uptime checkers, and load balancer health checks generate automated traffic that Bot Control may identify as bot activity. Exclude these requests from Bot Control inspection using a scope-down statement that matches on the source IP address range or a custom header that your monitoring tools include.

Excluding in-app browsers and non-standard HTTP libraries

If users access your site through in-app browsers, such as links opened from social media apps, or if your mobile app uses a non-standard HTTP library, these clients may trigger the `SignalNonBrowserUserAgent` rule. Standard native mobile frameworks are excluded from this rule, but other non-browser user agents are not. Configure a user-agent exception in the Bot Control rule group for these clients. For an example, see Bot Control example: Creating an exception for a blocked user agent.

Managing verified bot crawlers

Bot Control allows verified bots by default. If you want to rate-limit even verified crawlers, for example to reduce the load from aggressive but legitimate search engine crawling, use the bot labels to write custom rate-based rules that limit the request rate for specific verified bot categories. For an example, see Bot Control example: Explicitly allowing verified bots.

Forwarding bot signals to your origin

You can pass Bot Control classification labels to your application as custom request headers. Your origin can then use the labels to serve simplified content to suspected scrapers, log bot activity in your own analytics, or include a request identifier in block pages for false positive reporting. Use the AWS WAF custom request header feature to insert headers based on label matches. For more information about dynamic label values in headers, see Customized web requests and responses in AWS WAF.

Triggering step-up authentication

Instead of blocking suspicious traffic outright, you can forward Bot Control signals to your application and use them to trigger additional verification. For example, if Bot Control labels a session as suspicious, your application can require multi-factor authentication or present an additional verification step before completing a sensitive action. Step-up authentication reduces the impact of false positives while still protecting against automated threats. Use custom request headers to pass the relevant Bot Control labels to your origin.

Testing before enforcing

Always deploy Bot Control in count mode first. In count mode, Bot Control labels all requests but does not block any traffic. Review the labels in your AWS WAF logs to understand what Bot Control detects in your traffic, verify that legitimate traffic is not being mislabeled, and then switch to block mode once you are confident in the detection accuracy. For detailed guidance on testing and deployment, see Testing and deploying AWS WAF Bot Control.

Deploying behind a CDN or reverse proxy

If your application sits behind a third-party CDN, by default AWS WAF sees the CDN's IP address as the client. The Bot Control managed rule group recognizes traffic from Amazon CloudFront, Akamai, Cloudflare, and Fastly automatically and uses the originating client IP from the CDN's standard client IP header instead. Both the common protection level and the targeted protection level use this detection, including session-level rate limiting and the targeted bot detection rules

that depend on accurate client identity. You don't need additional forwarded IP configuration on the Bot Control rule group for these CDNs. For other rule statements in your protection pack (web ACL) that use IP addresses, such as your custom IP set match, geo match, ASN match, and rate-based rules, configure them with a forwarded IP configuration if you want them to evaluate the upstream client IP. For information about forwarded IP addresses, see Using forwarded IP addresses in AWS WAF. If your application is behind a different reverse proxy or a CDN that Bot Control doesn't recognize automatically, configure your reverse proxy to forward the client IP in a header such as `X-Forwarded-For` and use a forwarded IP configuration on the rule statements that evaluate IP addresses.

AWS WAF Bot Control components

The main components of a Bot Control implementation are the following:

- **`AWSManagedRulesBotControlRuleSet`** – The Bot Control managed rule group whose rules detect and handle various categories of bots. This rule group add labels to web requests that it detects as bot traffic.

  > **Note**
  >
  > You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

  The Bot Control managed rule group provides two levels of protection that you can choose from:

  - **Common** – Detects a variety of self-identifying bots, such as web scraping frameworks, search engines, and automated browsers. Bot Control protections at this level identify common bots using traditional bot detection techniques, such as static request data analysis. The rules label traffic from these bots and block the ones that they cannot verify.
  - **Targeted** – Includes the common-level protections and adds targeted detection for sophisticated bots that do not self identify. Targeted protections mitigate bot activity using a combination of rate limiting and CAPTCHA and background browser challenges.
    - **`TGT_`** – Rules that provide targeted protection have names that begin with `TGT_`. All targeted protections use detection techniques such as browser interrogation, fingerprinting, and behavior heuristics to identify bad bot traffic.
    - **`TGT_ML_`** – Targeted protection rules that use machine learning have names that begin with `TGT_ML_`. These rules use automated, machine-learning analysis of website traffic statistics

to detect anomalous behavior indicative of distributed, coordinated bot activity. AWS WAF analyzes statistics about your website traffic such as timestamps, browser characteristics, and previous URL visited, to improve the Bot Control machine learning model. Machine learning capabilities are enabled by default, but you can disable them in your rule group configuration. When machine learning is disabled, AWS WAF does not evaluate these rules.

For details including information about the rule group's rules, see AWS WAF Bot Control rule group.

You include this rule group in your protection pack (web ACL) using a managed rule group reference statement and indicating the inspection level that you want to use. For the targeted level, you also indicate whether to enable machine learning. For more information about adding this managed rule group to your protection pack (web ACL), see Adding the AWS WAF Bot Control managed rule group to your web ACL.

- **Bot Control dashboard** – The bot monitoring dashboard for your protection pack (web ACL), available through the protection pack (web ACL) Bot Control tab. Use this dashboard to monitor your traffic and understand how much of it comes from various types of bots. This can be a starting point for customizing your bot management, as described in this topic. You can also use it to verify your changes and monitor activity for various bots and bot categories.
- **AI Traffic Analysis dashboard** – Specialized dashboard for detailed AI bot and agent activity analysis, available through the protection pack (web ACL) AI Traffic Analysis tab. Provides enhanced visibility into AI-specific traffic patterns, bot intent, and access behaviors beyond standard Bot Control metrics.
- **JavaScript and mobile application integration SDKs** – You should implement the AWS WAF JavaScript and mobile SDKs if you use the targeted protection level of the Bot Control rule group. The targeted rules use information provided by the SDKs in the client tokens for enhanced detection against malicious bots. For more information about the SDKs, see Client application integrations in AWS WAF.
- **Logging and metrics** – You can monitor your bot traffic and understand how the Bot Control managed rule group evaluates and handles your traffic by studying the data that's collected for your protection pack (web ACL) by AWS WAF logs, Amazon Security Lake, and Amazon CloudWatch. The labels that Bot Control adds to your web requests are included in the data. For information about these options, see Logging AWS WAF protection pack (web ACL) traffic, Monitoring with Amazon CloudWatch, and What is Amazon Security Lake?.

Depending on your needs and the traffic that you see, you might want to customize your Bot Control implementation. The following are some of the most commonly used options.

- **Scope-down statements** – You can exclude some traffic from the web requests that the Bot Control managed rule group evaluates by adding a scope-down statement inside the Bot Control managed rule group reference statement. A scope-down statement can be any nestable rule statement. When a request doesn't match the scope-down statement, AWS WAF evaluates it as not matching the rule group reference statement without evaluating it against the rule group. For more information about scope-down statements, see Using scope-down statements in AWS WAF.

  Your costs for using the Bot Control managed rule group increase with the number of web requests that AWS WAF evaluates with it. You can help reduce these costs by using a scope-down statement to limit the requests that the rule group evaluates. For example, you might want to allow your homepage to load for everyone, including bots, and then apply the rule group rules to requests that are going to your application APIs or that contain a particular type of content.
- **Labels and label matching rules** – You can customize how the Bot Control rule group handles some of the bot traffic that it identifies using the AWS WAF label match rule statement. The Bot Control rule group adds labels to your web requests. You can add label matching rules after the Bot Control rule group that match on Bot Control labels and apply the handling that you need. For more information about labeling and using label match statements, see Label match rule statement and Web request labeling in AWS WAF.
- **Custom requests and responses** – You can add custom headers to requests that you allow and you can send custom responses for requests that you block by pairing label matching with the AWS WAF custom request and response features. With dynamic label interpolation, you can use ${namespace:} syntax to resolve Bot Control label values at evaluation time. A single rule can then forward an entire label namespace as a header value, instead of requiring one rule for each label. For more information about customizing requests and responses, see Customized web requests and responses in AWS WAF. For information about dynamic interpolation, see Dynamic label interpolation.

Using application integration SDKs with Bot Control

This section explains how to use application integration SDKs with Bot Control.

Most of the targeted protections of the Bot Control managed rule group require the challenge tokens that the application integration SDKs generate. The rules that don't require a challenge token on the request are the Bot Control common level protections and the targeted level machine learning rules. For descriptions of the protection levels and rules in the rule group, see AWS WAF Bot Control rule group.

We highly recommend implementing the application integration SDKs, for the most effective use of the Bot Control rule group. The challenge script must run before the Bot Control rule group in order for the rule group to benefit from the tokens that the script acquires.

At minimum, add the JavaScript challenge script tag to your most critical pages such as login, checkout, and account creation. For the strongest detection, integrate the SDK more broadly across pages that serve dynamic content. The more behavioral data Bot Control has, the better it can distinguish real users from advanced bots. For detailed integration guidance, see Integrating the client application integration SDKs.

- With the application integration SDKs, the script runs automatically.
- If you're unable to use the SDKs, you can configure your protection pack (web ACL) so that it runs the Challenge or CAPTCHA rule action against all requests that will be inspected by the Bot Control rule group. Using the Challenge or CAPTCHA rule action can incur additional fees. For pricing details, see AWS WAF Pricing.

When you implement the application integration SDKs in your clients or use one of the rule actions that runs the challenge script, you expand the capabilities of the rule group and of your overall client application security.

Tokens provide client information with each web request. This additional information enables the Bot Control rule group to separate legitimate client sessions from ill-behaved client sessions, even when both originate from a single IP address. The rule group uses information in the tokens to aggregate client session request behavior for the fine-tuned detection and mitigation that the targeted protections level provide.

For information about the SDKs, see Client application integrations in AWS WAF. For information about AWS WAF tokens, see Token use in AWS WAF intelligent threat mitigation. For information about the rule actions, see CAPTCHA and Challenge in AWS WAF.

Adding the AWS WAF Bot Control managed rule group to your web ACL

This section explains how to add and configure the `AWSManagedRulesBotControlRuleSet` rule group.

The Bot Control managed rule group `AWSManagedRulesBotControlRuleSet` requires additional configuration to identify the protection level that you want to implement.

For the rule group description and rules listing, see AWS WAF Bot Control rule group.

This guidance is intended for users who know generally how to create and manage AWS WAF protection packs (web ACLs), rules, and rule groups. Those topics are covered in prior sections of this guide. For basic information about how to add a managed rule group to your protection pack (web ACL), see Adding a managed rule group to a protection pack (web ACL) through the console.

Follow best practices

Use the Bot Control rule group in accordance with the best practices at Best practices for intelligent threat mitigation in AWS WAF.

To use the `AWSManagedRulesBotControlRuleSet` rule group in your protection pack (web ACL)

1. Add the AWS managed rule group, `AWSManagedRulesBotControlRuleSet` to your protection pack (web ACL). For the full rule group description, see the section called "Bot Control rule group".

   > **Note**
   >
   > You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

   When you add the rule group, edit it to open the configuration page for the rule group.
2. On the rule group's configuration page, in the **Inspection level** pane, select the inspection level that you want to use.

   - **Common** – Detects a variety of self-identifying bots, such as web scraping frameworks, search engines, and automated browsers. Bot Control protections at this level identify common bots using traditional bot detection techniques, such as static request data analysis. The rules label traffic from these bots and block the ones that they cannot verify.
   - **Targeted** – Includes the common-level protections and adds targeted detection for sophisticated bots that do not self identify. Targeted protections mitigate bot activity using a combination of rate limiting and CAPTCHA and background browser challenges.
     - **`TGT_`** – Rules that provide targeted protection have names that begin with `TGT_`. All targeted protections use detection techniques such as browser interrogation, fingerprinting, and behavior heuristics to identify bad bot traffic.

- **`TGT_ML_`** – Targeted protection rules that use machine learning have names that begin with `TGT_ML_`. These rules use automated, machine-learning analysis of website traffic statistics to detect anomalous behavior indicative of distributed, coordinated bot activity. AWS WAF analyzes statistics about your website traffic such as timestamps, browser characteristics, and previous URL visited, to improve the Bot Control machine learning model. Machine learning capabilities are enabled by default, but you can disable them in your rule group configuration. When machine learning is disabled, AWS WAF does not evaluate these rules.

3. If you're using the targeted protection level and you don't want AWS WAF to use machine learning (ML) to analyze web traffic for distributed, coordinated bot activity, disable the machine learning option. Machine learning is required for the Bot Control rules whose names start with `TGT_ML_`. For details about these rules, see Bot Control rules listing.
4. Add a scope-down statement for the rule group, to contain the costs of using it. A scope-down statement narrows the set of requests that the rule group inspects. For example use cases, start with Bot Control example: Using Bot Control only for the login page and Bot Control example: Using Bot Control only for dynamic content.
5. Provide any additional configuration that you need for the rule group.
6. Save your changes to the protection pack (web ACL).

Before you deploy your Bot Control implementation for production traffic, test and tune it in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune the rules in count mode with your production traffic before enabling them. See the sections that follow for guidance.

Example scenarios of false positives with AWS WAF Bot Control

This section provides example situations where you might encounter false positives with AWS WAF Bot Control.

We have carefully selected the rules in the AWS WAF Bot Control managed rule group to minimize false positives. We test the rules against global traffic and monitor their impact on test protection packs (web ACLs). However, it's still possible to get false positives due to changes in traffic patterns. Additionally, some use cases are known to cause false positives and will require customization specific to your web traffic.

Situations where you might encounter false positives include the following:

- Mobile apps typically have non-browser user agents, which the `SignalNonBrowserUserAgent` rule blocks by default. If you expect traffic from mobile apps, or any other legitimate traffic with non-browser user agents, you'll need to add an exception to allow it.
- You might rely on some specific bot traffic for things like uptime monitoring, integration testing, or marketing tools. If Bot Control identifies and blocks the bot traffic that you want to allow, you need to alter the handling by adding your own rules. While this isn't a false positive scenario for all customers, if it is for you, you will need to handle it the same as for a false positive.
- The Bot Control managed rule group verifies bots using the IP addresses from AWS WAF. If you use Bot Control and you have verified bots that route through a proxy or load balancer, you might need to explicitly allow them using a custom rule. For information about how to create a custom rule of this type, see Using forwarded IP addresses in AWS WAF.
- A Bot Control rule with a low global false positive rate might heavily impact specific devices or applications. For example, in testing and validation, we might not have observed requests from applications with low traffic volumes or from less common browsers or devices.
- A Bot Control rule that has a historically low false positive rate might have increased false positives for valid traffic. This might be due to new traffic patterns or request attributes that emerge with valid traffic, causing it to match the rule where it didn't before. These changes might be due to situations like the following:
  - Traffic details that are altered as traffic flows through network appliances, such as load balancers or content distribution networks (CDN).
  - Emerging changes in traffic data, for example new browsers or new versions for existing browsers.

Important

If your mobile app uses a non-standard HTTP library, or if users access your site through in-app browsers (such as links opened from social media apps), these clients may trigger the `SignalNonBrowserUserAgent` rule. Standard native mobile frameworks are excluded from this rule, but other non-browser user agents are not. Plan to test and add exceptions before enabling Bot Control in block mode. For an example, see Bot Control example: Creating an exception for a blocked user agent.

For information about how to handle false positives that you might get from the AWS WAF Bot Control managed rule group, see the guidance in the section that follows, Testing and deploying AWS WAF Bot Control.

Testing and deploying AWS WAF Bot Control

This section provides general guidance for configuring and testing an AWS WAF Bot Control implementation for your site. The specific steps that you choose to follow will depend on your needs, resources, and the web requests that you receive.

This information is in addition to the general information about testing and tuning provided at Testing and tuning your AWS WAF protections.

> **Note**
>
> AWS Managed Rules are designed to protect you from common web threats. When used in accordance with the documentation, AWS Managed Rules rule groups add another layer of security for your applications. However, AWS Managed Rules rule groups aren't intended as a replacement for your security responsibilities, which are determined by the AWS resources that you select. Refer to the Shared Responsibility Model to ensure that your resources in AWS are properly protected.

> **Production traffic risk**
>
> Before you deploy your Bot Control implementation for production traffic, test and tune it in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune the rules in count mode with your production traffic before enabling them.

This guidance is intended for users who know generally how to create and manage AWS WAF protection packs (web ACLs), rules, and rule groups. Those topics are covered in prior sections of this guide.

To configure and test a Bot Control implementation

Perform these steps first in a test environment, then in production.

1. **Add the Bot Control managed rule group**

> **Note**
>
> You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

Add the managed AWS rule group `AWSManagedRulesBotControlRuleSet` to a new or existing protection pack (web ACL) and configure it so that it doesn't alter current protection pack (web ACL) behavior.

- When you add the managed rule group, edit it and do the following:
  - In the **Inspection level** pane, select the inspection level that you want to use.
    - **Common** – Detects a variety of self-identifying bots, such as web scraping frameworks, search engines, and automated browsers. Bot Control protections at this level identify common bots using traditional bot detection techniques, such as static request data analysis. The rules label traffic from these bots and block the ones that they cannot verify.
    - **Targeted** – Includes the common-level protections and adds targeted detection for sophisticated bots that do not self identify. Targeted protections mitigate bot activity using a combination of rate limiting and CAPTCHA and background browser challenges.
      - **`TGT_`** – Rules that provide targeted protection have names that begin with `TGT_`. All targeted protections use detection techniques such as browser interrogation, fingerprinting, and behavior heuristics to identify bad bot traffic.
      - **`TGT_ML_`** – Targeted protection rules that use machine learning have names that begin with `TGT_ML_`. These rules use automated, machine-learning analysis of website traffic statistics to detect anomalous behavior indicative of distributed, coordinated bot activity. AWS WAF analyzes statistics about your website traffic such as timestamps, browser characteristics, and previous URL visited, to improve the Bot Control machine learning model. Machine learning capabilities are enabled by default, but you can disable them in your rule group configuration. When machine learning is disabled, AWS WAF does not evaluate these rules.

    For more information about this choice, see AWS WAF Bot Control rule group.

- In the **Rules** pane, open the **Override all rule actions** dropdown and choose **Count**. With this configuration, AWS WAF evaluates requests against all of the rules in the rule group and only counts the matches that result, while still adding labels to requests. For more information, see Overriding rule actions in a rule group.

  With this override, you can monitor the potential impact of the Bot Control rules on your traffic, to determine whether you want to add exceptions for things like internal use cases or desired bots.

- Position the rule group so that it's evaluated last in the protection pack (web ACL), with a priority setting that's numerically higher than any other rules or rule groups that you're already using. For more information, see Setting rule priority.

  This way, your current handling of traffic isn't disrupted. For example, if you have rules that detect malicious traffic such as SQL injection or cross-site scripting, they'll continue to detect and log those requests. Alternately, if you have rules that allow known non-malicious traffic, they can continue to allow that traffic, without having it blocked by the Bot Control managed rule group. You might decide to adjust the processing order during your testing and tuning activities, but this is a good way to start.

2. **Enable logging and metrics for the protection pack (web ACL)**

   As needed, configure logging, Amazon Security Lake data collection, request sampling, and Amazon CloudWatch metrics for the protection pack (web ACL). You can use these visibility tools to monitor the interaction of the Bot Control managed rule group with your traffic.

   - For information about logging, see Logging AWS WAF protection pack (web ACL) traffic.
   - For information about Amazon Security Lake, see What is Amazon Security Lake? and Collecting data from AWS services in the *Amazon Security Lake user guide*.
   - For information about Amazon CloudWatch metrics, see Monitoring with Amazon CloudWatch.
   - For information about web request sampling, see Viewing a sample of web requests.

3. **Associate the protection pack (web ACL) with a resource**

   If the protection pack (web ACL) isn't already associated with a resource, associate it. For information, see Associating or disassociating protection with an AWS resource.

4. **Monitor traffic and Bot Control rule matches**

   Make sure that traffic is flowing and that the Bot Control managed rule group rules are adding labels to matching web requests. You can see the labels in the logs and see bot and label metrics in the Amazon CloudWatch metrics. In the logs, the rules that you've overridden to count in the rule group show up in the `ruleGroupList` with `action` set to count, and with `overriddenAction` indicating the configured rule action that you overrode.

   > **Note**
   >
   > The Bot Control managed rule group verifies bots using the IP addresses from AWS WAF. If you use Bot Control and you have verified bots that route through a proxy or load balancer, you might need to explicitly allow them using a custom rule. For information about how to create a custom rule, see Using forwarded IP addresses in AWS WAF. For information about how you can use the rule to customize Bot Control web request handling, see the next step.

   Carefully review the web request handling for any false positives that you might need to mitigate with custom handling. For examples of false positives, see Example scenarios of false positives with AWS WAF Bot Control.

5. **Customize Bot Control web request handling**

   As needed, add your own rules that explicitly allow or block requests, to change how Bot Control rules would otherwise handle them.

   How you do this depends on your use case, but the following are common solutions:

   - Explicitly allow requests with a rule that you add before the Bot Control managed rule group. With this, the allowed requests never reach the rule group for evaluation. This can help contain the cost of using the Bot Control managed rule group.
   - Exclude requests from Bot Control evaluation by adding a scope-down statement inside the Bot Control managed rule group statement. This functions the same as the preceding option. It can help contain the cost of using the Bot Control managed rule group because the requests that don't match the scope-down statement never reach rule group evaluation. For information about scope-down statements, see Using scope-down statements in AWS WAF.

For examples, see the following:

- Excluding IP range from bot management
- Allowing traffic from a bot that you control

- Use Bot Control labels in request handling to allow or block requests. Add a label match rule after the Bot Control managed rule group to filter out labeled requests that you want to allow from those that you want to block.

  After testing, keep the related Bot Control rules in count mode, and maintain the request handling decisions in your custom rule. For information about label match statements, see Label match rule statement.

  For examples of this type of customization, see the following:

  - Creating an exception for a blocked user agent
  - Allowing a specific blocked bot
  - Blocking verified bots

For additional examples, see AWS WAF Bot Control examples.

6. **As needed, enable the Bot Control managed rule group settings**

   Depending on your situation, you might have decided that you want to leave some Bot Control rules in count mode or with a different action override. For the rules that you want to have run as they are configured inside the rule group, enable the regular rule configuration. To do this, edit the rule group statement in your protection pack (web ACL) and make your changes in the **Rules** pane.

AWS WAF Bot Control examples

This section shows example configurations that satisfy a variety of common use cases for AWS WAF Bot Control implementations.

Each example provides a description of the use case and then shows the solution in JSON listings for the custom configured rules.

> **Note**
>
> The JSON listings shown in these examples were created in the console by configuring the rule and then editing it using the **Rule JSON editor**.

Topics

- Bot Control example: Simple configuration
- Bot Control example: Explicitly allowing verified bots
- Bot Control example: Blocking verified bots
- Bot Control example: Allowing a specific blocked bot
- Bot Control example: Creating an exception for a blocked user agent
- Bot Control example: Using Bot Control only for the login page
- Bot Control example: Using Bot Control only for dynamic content
- Bot Control example: Excluding IP range from bot management
- Bot Control example: Allowing traffic from a bot that you control
- Bot Control example: Enabling targeted inspection level
- Bot Control example: Using two statements to limit the use of the targeted inspection level

Bot Control example: Simple configuration

The following JSON listing shows an example protection pack (web ACL) with an AWS WAF Bot Control managed rule group. Note the visibility configuration, which causes AWS WAF to store request samples and metrics for monitoring purposes.

```
{
  "Name": "Bot-WebACL",
  "Id": "...",
  "ARN": "...",
  "DefaultAction": {
    "Allow": {}
  },
  "Description": "Bot-WebACL",
  "Rules": [
      {
        ...
```

```
      },
      {
         "Name": "AWS-AWSBotControl-Example",
         "Priority": 5,
         "Statement": {
            "ManagedRuleGroupStatement": {
               "VendorName": "AWS",
               "Name": "AWSManagedRulesBotControlRuleSet",
               "ManagedRuleGroupConfigs": [
                 {
                   "AWSManagedRulesBotControlRuleSet": {
                     "InspectionLevel": "COMMON"
                   }
                 }
               ],
               "RuleActionOverrides": [],
               "ExcludedRules": []
            },
            "VisibilityConfig": {
               "SampledRequestsEnabled": true,
               "CloudWatchMetricsEnabled": true,
               "MetricName": "AWS-AWSBotControl-Example"
             }
          }
      }
    ],
    "VisibilityConfig": {
      ...
    },
    "Capacity": 1496,
    "ManagedByFirewallManager": false,
    "RetrofittedByFirewallManager": false
}
```

Bot Control example: Explicitly allowing verified bots

AWS WAF Bot Control doesn't block bots that are known by AWS to be common and verifiable bots. When Bot Control identifies a web request as coming from a verified bot, it adds a label that names the bot and a label that indicates that it's a verified bot. Bot Control doesn't add any other labels, such as signals labels, in order to prevent known good bots from being blocked.

You might have other AWS WAF rules that block verified bots. If you want to ensure that verified bots are allowed, add a custom rule to allow them based on the Bot Control labels. Your new

rule must run after the Bot Control managed rule group, so that the labels are available to match against.

The following rule explicitly allows verified bots.

```
{
    "Name": "match_rule",
    "Statement": {
      "LabelMatchStatement": {
        "Scope": "LABEL",
        "Key": "awswaf:managed:aws:bot-control:bot:verified"
      }
    },
    "RuleLabels": [],
    "Action": {
      "Allow": {}
    }
}
```

Bot Control example: Blocking verified bots

In order to block verified bots, you must add a rule to block them that runs after the AWS WAF Bot Control managed rule group. To do this, identify the bot names that you want to block and use a label match statement to identify and block them. If you want to just block all verified bots, you can omit the match against the `bot:name:` label.

The following rule blocks only the `bingbot` verified bot. This rule must run after the Bot Control managed rule group.

```
{
    "Name": "match_rule",
    "Statement": {
      "AndStatement": {
        "Statements": [
          {
            "LabelMatchStatement": {
              "Scope": "LABEL",
              "Key": "awswaf:managed:aws:bot-control:bot:name:bingbot"
            }
          },
          {
            "LabelMatchStatement": {
```

```
              "Scope": "LABEL",
              "Key": "awswaf:managed:aws:bot-control:bot:verified"
            }
          }
        ]
      }
    },
    "RuleLabels": [],
    "Action": {
      "Block": {}
    }
  }
```

The following rule blocks all verified bots.

```
{
    "Name": "match_rule",
    "Statement": {
      "LabelMatchStatement": {
        "Scope": "LABEL",
        "Key": "awswaf:managed:aws:bot-control:bot:verified"
      }
    },
    "RuleLabels": [],
    "Action": {
      "Block": {}
    }
}
```

Bot Control example: Allowing a specific blocked bot

It's possible for a bot to be blocked by more than one of the Bot Control rules. Run through the following procedure for each blocking rule.

If an AWS WAF Bot Control rule is blocking a bot that you do not want to block, do the following:

1. Identify the Bot Control rule that's blocking the bot by checking the logs. The blocking rule will be specified in the logs in the fields whose names start with `terminatingRule`. For information about the protection pack (web ACL) logs, see Logging AWS WAF protection pack (web ACL) traffic. Note the label that the rule is adds to the requests.
2. In your protection pack (web ACL), override the action of the blocking rule to count. To do this in the console, edit the rule group rule in the protection pack (web ACL) and choose a rule action

override of Count for the rule. This ensures that the bot is not blocked by the rule, but the rule will still apply its label to matching requests.

3. Add a label matching rule to your protection pack (web ACL), after the Bot Control managed rule group. Configure the rule to match against the overridden rule's label and to block all matching requests except for the bot that you don't want to block.

   Your protection pack (web ACL) is now configured so that the bot you want to allow is no longer blocked by the blocking rule that you identified through the logs.

Check traffic and your logs again, to be sure that the bot is being allowed through. If not, run through the above procedure again.

For example, suppose you want to block all monitoring bots except for `pingdom`. In this case, you override the `CategoryMonitoring` rule to count and then write a rule to block all monitoring bots except for those with the bot name label `pingdom`.

The following rule uses the Bot Control managed rule group but overrides the rule action for `CategoryMonitoring` to count. The category monitoring rule applies its labels as usual to matching requests, but only counts them instead of performing its usual action of block.

```
{
  "Name": "AWS-AWSBotControl-Example",
  "Priority": 5,
  "Statement": {
    "ManagedRuleGroupStatement": {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesBotControlRuleSet",
      "ManagedRuleGroupConfigs": [
        {
          "AWSManagedRulesBotControlRuleSet": {
            "InspectionLevel": "COMMON"
          }
        }
      ],
   "RuleActionOverrides": [
        {
          "ActionToUse": {
            "Count": {}
          },
          "Name": "CategoryMonitoring"
        }
```

```
      ],
      "ExcludedRules": []
    }
  },
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "AWS-AWSBotControl-Example"
  }
}
```

The following rule matches against the category monitoring label that the preceding `CategoryMonitoring` rule adds to matching web requests. Among the category monitoring requests, this rule blocks all but those that have a label for the bot name `pingdom`.

The following rule must run after the preceding Bot Control managed rule group in the protection pack (web ACL) processing order.

```
{
      "Name": "match_rule",
      "Priority": 10,
      "Statement": {
        "AndStatement": {
          "Statements": [
            {
              "LabelMatchStatement": {
                "Scope": "LABEL",
                "Key": "awswaf:managed:aws:bot-control:bot:category:monitoring"
              }
            },
            {
              "NotStatement": {
                "Statement": {
                  "LabelMatchStatement": {
                    "Scope": "LABEL",
                    "Key": "awswaf:managed:aws:bot-control:bot:name:pingdom"
                  }
                }
              }
            }
          ]
        }
      },
```

```
      "Action": {
        "Block": {}
      },
      "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "match_rule"
      }
}
```

Bot Control example: Creating an exception for a blocked user agent

If traffic from some non-browser user agents is being erroneously blocked, you can create an exception by setting the offending AWS WAF Bot Control rule `SignalNonBrowserUserAgent` to Count and then combining the rule's labeling with your exception criteria.

> **Note**
>
> Mobile apps typically have non-browser user agents, which the `SignalNonBrowserUserAgent` rule blocks by default.

The following rule uses the Bot Control managed rule group but overrides the rule action for `SignalNonBrowserUserAgent` to Count. The signal rule applies its labels as usual to matching requests, but only counts them instead of performing its usual action of block.

```
{
  "Name": "AWS-AWSBotControl-Example",
  "Priority": 5,
  "Statement": {
    "ManagedRuleGroupStatement": {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesBotControlRuleSet",
      "ManagedRuleGroupConfigs": [
        {
          "AWSManagedRulesBotControlRuleSet": {
            "InspectionLevel": "COMMON"
          }
        }
      ],
   "RuleActionOverrides": [
        {
```

```
          "ActionToUse": {
            "Count": {}
          },
          "Name": "SignalNonBrowserUserAgent"
        }
      ],
      "ExcludedRules": []
    }
  },
  "VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "AWS-AWSBotControl-Example"
  }
}
```

The following rule matches against the signal label that the Bot Control `SignalNonBrowserUserAgent` rule adds to its matching web requests. Among the signal requests, this rule blocks all but those that have the user agent that we want to allow.

The following rule must run after the preceding Bot Control managed rule group in the protection pack (web ACL) processing order.

```
{
    "Name": "match_rule",
    "Statement": {
      "AndStatement": {
        "Statements": [
          {
            "LabelMatchStatement": {
              "Scope": "LABEL",
              "Key": "awswaf:managed:aws:bot-control:signal:non_browser_user_agent"
            }
          },
          {
            "NotStatement": {
              "Statement": {
                "ByteMatchStatement": {
                  "FieldToMatch": {
                    "SingleHeader": {
                      "Name": "user-agent"
                    }
                  },
```

```
                  "PositionalConstraint": "EXACTLY",
                  "SearchString": "PostmanRuntime/7.29.2",
                  "TextTransformations": [
                    {
                      "Priority": 0,
                      "Type": "NONE"
                    }
                  ]
                }
              }
            }
          }
        ]
      }
    },
    "RuleLabels": [],
    "Action": {
      "Block": {}
    },
    "VisibilityConfig": {
      "SampledRequestsEnabled": true,
      "CloudWatchMetricsEnabled": true,
      "MetricName": "match_rule"
    }
}
```

Bot Control example: Using Bot Control only for the login page

The following example uses a scope-down statement to apply AWS WAF Bot Control only for traffic that's coming to a website's login page, which is identified by the URI path `login`. The URI path to your login page might be different from the example, depending on your application and environment.

```
{
  "Name": "AWS-AWSBotControl-Example",
  "Priority": 5,
  "Statement": {
    "ManagedRuleGroupStatement": {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesBotControlRuleSet",
   "ManagedRuleGroupConfigs": [
        {
          "AWSManagedRulesBotControlRuleSet": {
```

```
            "InspectionLevel": "COMMON"
          }
        }
      ],
      "RuleActionOverrides": [],
      "ExcludedRules": []
    },
    "VisibilityConfig": {
      "SampledRequestsEnabled": true,
      "CloudWatchMetricsEnabled": true,
      "MetricName": "AWS-AWSBotControl-Example"
    },
    "ScopeDownStatement": {
      "ByteMatchStatement": {
        "SearchString": "login",
        "FieldToMatch": {
          "UriPath": {}
        },
        "TextTransformations": [
          {
            "Priority": 0,
            "Type": "NONE"
          }
        ],
        "PositionalConstraint": "CONTAINS"
      }
    }
  }
}
```

Bot Control example: Using Bot Control only for dynamic content

This example uses a scope-down statement to apply AWS WAF Bot Control only to dynamic content.

The scope-down statement excludes static content by negating the match results for a regex pattern set:

- The regex pattern set is configured to match extensions of *static content*. For example, the regex pattern set specification might be `(?i)\.(jpe?g|gif|png|svg|ico|css|js|woff2?)$`. For information about regex pattern sets and statements, see Regex pattern set match rule statement.

- In the scope-down statement, we exclude the matching static content by nesting the regex pattern set statement inside a NOT statement. For information about the NOT statement, see NOT rule statement.

```
{
  "Name": "AWS-AWSBotControl-Example",
  "Priority": 5,
  "Statement": {
    "ManagedRuleGroupStatement": {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesBotControlRuleSet",
   "ManagedRuleGroupConfigs": [
        {
          "AWSManagedRulesBotControlRuleSet": {
            "InspectionLevel": "COMMON"
          }
        }
      ],
      "RuleActionOverrides": [],
      "ExcludedRules": []
    },
    "VisibilityConfig": {
      "SampledRequestsEnabled": true,
      "CloudWatchMetricsEnabled": true,
      "MetricName": "AWS-AWSBotControl-Example"
    },
    "ScopeDownStatement": {
      "NotStatement": {
        "Statement": {
          "RegexPatternSetReferenceStatement": {
            "ARN": "arn:aws:wafv2:us-east-1:123456789:regional/regexpatternset/
excludeset/00000000-0000-0000-0000-000000000000",
            "FieldToMatch": {
              "UriPath": {}
            },
            "TextTransformations": [
              {
                "Priority": 0,
                "Type": "NONE"
              }
            ]
          }
```

```
        }
      }
    }
  }
}
```

Bot Control example: Excluding IP range from bot management

If you want to exclude a subset of web traffic from AWS WAF Bot Control management, and you can identify that subset using a rule statement, then exclude it by adding a scope-down statement to your Bot Control managed rule group statement.

The following rule performs normal Bot Control bot management on all web traffic except for web requests coming from a specific IP address range.

```
{
  "Name": "AWS-AWSBotControl-Example",
  "Priority": 5,
  "Statement": {
    "ManagedRuleGroupStatement": {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesBotControlRuleSet",
      "ManagedRuleGroupConfigs": [
        {
          "AWSManagedRulesBotControlRuleSet": {
            "InspectionLevel": "COMMON"
          }
        }
      ],
      "RuleActionOverrides": [],
      "ExcludedRules": []
    },
    "VisibilityConfig": {
      "SampledRequestsEnabled": true,
      "CloudWatchMetricsEnabled": true,
      "MetricName": "AWS-AWSBotControl-Example"
    },
    "ScopeDownStatement": {
      "NotStatement": {
        "Statement": {
          "IPSetReferenceStatement": {
            "ARN": "arn:aws:wafv2:us-east-1:123456789:regional/ipset/
friendlyips/00000000-0000-0000-0000-000000000000"
```

```
          }
        }
      }
    }
  }
}
```

Bot Control example: Allowing traffic from a bot that you control

You can configure some site monitoring bots and custom bots to send custom headers. If you want to allow traffic from these types of bots, you can configure them to add a shared secret in a header. You then can exclude messages that have the header by adding a scope-down statement to the AWS WAF Bot Control managed rule group statement.

The following example rule excludes traffic with a secret header from Bot Control inspection.

```
{
  "Name": "AWS-AWSBotControl-Example",
  "Priority": 5,
  "Statement": {
    "ManagedRuleGroupStatement": {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesBotControlRuleSet",
      "ManagedRuleGroupConfigs": [
        {
          "AWSManagedRulesBotControlRuleSet": {
            "InspectionLevel": "COMMON"
          }
        }
      ],
      "RuleActionOverrides": [],
      "ExcludedRules": []
    },
    "VisibilityConfig": {
      "SampledRequestsEnabled": true,
      "CloudWatchMetricsEnabled": true,
      "MetricName": "AWS-AWSBotControl-Example"
    },
    "ScopeDownStatement": {
      "NotStatement": {
        "Statement": {
          "ByteMatchStatement": {
            "SearchString": "YSBzZWNyZXQ=",
```

```
            "FieldToMatch": {
              "SingleHeader": {
                "Name": "x-bypass-secret"
              }
            },
            "TextTransformations": [
              {
                "Priority": 0,
                "Type": "NONE"
              }
            ],
            "PositionalConstraint": "EXACTLY"
          }
        }
      }
    }
  }
}
```

Bot Control example: Enabling targeted inspection level

For an enhanced level of protection, you can enable the targeted inspection level in your AWS WAF Bot Control managed rule group.

In the following example, machine learning features are enabled. You can opt out of this behavior by setting `EnableMachineLearning` to `false`.

```
{
  "Name": "AWS-AWSBotControl-Example",
  "Priority": 5,
  "Statement": {
    "ManagedRuleGroupStatement": {
      "VendorName": "AWS",
      "Name": "AWSManagedRulesBotControlRuleSet",
      "ManagedRuleGroupConfigs": [
        {
          "AWSManagedRulesBotControlRuleSet": {
            "InspectionLevel": "TARGETED",
            "EnableMachineLearning": true
          }
        }
      ],
      "RuleActionOverrides": [],
```

```
      "ExcludedRules": []
    },
    "VisibilityConfig": {
      "SampledRequestsEnabled": true,
      "CloudWatchMetricsEnabled": true,
      "MetricName": "AWS-AWSBotControl-Example"
    }
  }
}
```

Bot Control example: Using two statements to limit the use of the targeted inspection level

As a cost optimization, you can use two AWS WAF Bot Control managed rule group statements in your protection pack (web ACL), with separate inspection levels and scoping. For instance, you could scope the targeted inspection level statement only to more sensitive application endpoints.

The two statements in the following example have mutually exclusive scoping. Without this configuration, a request could result in two billed Bot Control evaluations.

> **Note**
>
> Multiple statements referencing `AWSManagedRulesBotControlRuleSet` are not supported in the visual editor in the console. Instead, use the JSON editor.

```
{
  "Name": "Bot-WebACL",
  "Id": "...",
  "ARN": "...",
  "DefaultAction": {
    "Allow": {}
  },
  "Description": "Bot-WebACL",
  "Rules": [
      {
        ...
      },
      {
       "Name": "AWS-AWSBotControl-Common",
       "Priority": 5,
       "Statement": {
          "ManagedRuleGroupStatement": {
```

```
            "VendorName": "AWS",
            "Name": "AWSManagedRulesBotControlRuleSet",
            "ManagedRuleGroupConfigs": [
              {
                "AWSManagedRulesBotControlRuleSet": {
                  "InspectionLevel": "COMMON"
                }
              }
            ],
            "RuleActionOverrides": [],
            "ExcludedRules": []
         },
         "VisibilityConfig": {
            "SampledRequestsEnabled": true,
            "CloudWatchMetricsEnabled": true,
            "MetricName": "AWS-AWSBotControl-Common"
          },
          "ScopeDownStatement": {
             "NotStatement": {
               "Statement": {
                 "ByteMatchStatement": {
                   "FieldToMatch": {
                     "UriPath": {}
                   },
                   "PositionalConstraint": "STARTS_WITH",
                   "SearchString": "/sensitive-endpoint",
                   "TextTransformations": [
                     {
                       "Type": "NONE",
                       "Priority": 0
                     }
                   ]
                 }
               }
             }
           }
       }
      },
      {
       "Name": "AWS-AWSBotControl-Targeted",
       "Priority": 6,
       "Statement": {
          "ManagedRuleGroupStatement": {
             "VendorName": "AWS",
```

```
            "Name": "AWSManagedRulesBotControlRuleSet",
            "ManagedRuleGroupConfigs": [
              {
                "AWSManagedRulesBotControlRuleSet": {
                  "InspectionLevel": "TARGETED",
                  "EnableMachineLearning": true
                }
              }
            ],
            "RuleActionOverrides": [],
            "ExcludedRules": []
          },
          "VisibilityConfig": {
            "SampledRequestsEnabled": true,
            "CloudWatchMetricsEnabled": true,
            "MetricName": "AWS-AWSBotControl-Targeted"
          },
          "ScopeDownStatement": {
             "Statement": {
               "ByteMatchStatement": {
                 "FieldToMatch": {
                   "UriPath": {}
                 },
                 "PositionalConstraint": "STARTS_WITH",
                 "SearchString": "/sensitive-endpoint",
                 "TextTransformations": [
                   {
                     "Type": "NONE",
                     "Priority": 0
                   }
                 ]
               }
             }
           }
        }
      }
    ],
    "VisibilityConfig": {
      ...
    },
    "Capacity": 1496,
    "ManagedByFirewallManager": false,
    "RetrofittedByFirewallManager": false
```

```
}
```

AWS WAF Distributed Denial of Service (DDoS) prevention

AWS WAF offers sophisticated and customizable protection against DDoS attacks in your AWS resources. Review the options described in this section and select the level of Anti-DDoS protection that meets your security and business needs.

You can choose from two tiers of DDoS protection in AWS WAF:

Resource-level DDoS protection

The standard tier works within Application Load Balancers to defend against known malicious sources through on-host filtering. You can configure the protective behavior to best react to potential DDoS events.

Resource-level DDoS protection:

- Monitors your traffic patterns automatically.
- Updates threat intelligence in real time.
- Protects against known malicious sources.

To optimize web ACL request costs for your Application Load Balancer

You must associate a web ACL with your Application Load Balancer to enable resource-level protection. If your Application Load Balancer is associated with a web ACL that has no configuration, you will not incur charges from AWS WAF requests, however, AWS WAF will not provide sampled requests or report on the Application Load Balancer in CloudWatch metrics. You can take the following actions to enable observability features for the Application Load Balancer:

- Use the `Block` action or `Allow` action with custom request headers in the `DefaultAction`. For information, see Inserting custom request headers for non-blocking actions.
- Add any rules to the web ACL. For information, see AWS WAF rules.
- Enable a logging destination. For information, see Configuring logging for a protection pack (web ACL).
- Associate the web ACL with an AWS Firewall Manager policy. For information, see Creating an AWS Firewall Manager policy for AWS WAF.

AWS WAF will not provide sampled requests or publish CloudWatch metrics without these configurations.

AWS managed rule group DDoS protection

The advanced tier of DDoS protections is offered through the `AWSManagedRulesAntiDDoSRuleSet`. The managed rule group complements the resource-level tier of protection, with the following notable differences:

- Protection extends to CloudFront distributions, Application Load Balancers, and other AWS services supported by AWS WAF
- Traffic baselines are created for your protected resources to improve detection of novel attack patterns.
- Protective behavior is activated according to sensitivity levels you select.
- Manages and labels requests to protected resources during probable DDoS events.

For a comprehensive list of the rules and functionality included, see AWS WAF Distributed Denial of Service (DDoS) prevention rule group.

Note

You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

Topics

- Resource-level DDoS protection for Application Load Balancers
- Advanced Anti-DDoS protection using the AWS WAF Anti-DDoS managed rule group

Resource-level DDoS protection for Application Load Balancers

Resource level DDoS protection adds immediate defense to Application Load Balancers without incurring charges of deploying AWS WAF managed rule groups. This standard tier of Anti-DDoS protection uses AWS threat intelligence and traffic pattern analysis to protect Application Load Balancers. To identify known malicous sources, Anti-DDoS protection performs on-host filtering of

both direct client IP addresses and X-Forwarded-For (XFF) headers. After a known malicious source is identified, protection is activated through one of two modes:

Active under DDoS is the default protective mode and is recommended for most use cases.

This mode:

- Activates protection automatically when detecting high load conditions or potential DDoS events
- Rate-limits traffic from known malicious sources only during attack conditions
- Minimizes impact on legitimate traffic during normal operations
- Uses Application Load Balancer health metrics and AWS WAF response data to determine when to engage protection

Always on is an optional mode that is always active once enabled.

This mode:

- Maintains continuous protection against known malicious sources
- Rate-limits traffic from known malicious sources in real time
- Applies protection to both direct connections and requests with malicious IPs in XFF headers
- May have higher impact on legitimate traffic but provides maximum security

Requests blocked by resource-level DDoS protection are recorded in CloudWatch logs as either `LowReputationPacketsDropped` or `LowReputationRequestsDenied` metrics. For information, see AWS WAF core metrics and dimensions.

Enable standard DDoS protection on an existing webACL

You can enable DDoS protection when you create a web ACL or update an existing web ACL associated with Application Load Balancer.

> **Note**
>
> If you have an existing web ACL that is associated with an Application Load Balancer, Anti-DDoS protection is enabled by default with **Active under DDoS** mode.

To enable Anti-DDoS protection in the AWS WAF console

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. Choose **Web ACLs** in the navigation pane, and then open any web ACL that is associated with an Application Load Balancer.
3. Choose **Associated AWS resources**.
4. Under **Resource level DDoS protection**, choose **Edit**.
5. Select one of the following protection modes:
   - **Active under DDoS** (recommended) - Protection engages only during high load conditions
   - **Always on** - Always-on protection against known malicious sources
6. Choose **Save changes**.

Note

For information about creating a web ACL, see Creating a protection pack (web ACL) in AWS WAF.

To optimize web ACL request costs for your Application Load Balancer

You must associate a web ACL with your Application Load Balancer to enable resource-level protection. If your Application Load Balancer is associated with a web ACL that has no configuration, you will not incur charges from AWS WAF requests, however, AWS WAF will not provide sampled requests or report on the Application Load Balancer in CloudWatch metrics. You can take the following actions to enable observability features for the Application Load Balancer:

- Use the `Block` action or `Allow` action with custom request headers in the `DefaultAction`. For information, see Inserting custom request headers for non-blocking actions.
- Add any rules to the web ACL. For information, see AWS WAF rules.
- Enable a logging destination. For information, see Configuring logging for a protection pack (web ACL).

- Associate the web ACL with an AWS Firewall Manager policy. For information, see Creating an AWS Firewall Manager policy for AWS WAF.

AWS WAF will not provide sampled requests or publish CloudWatch metrics without these configurations.

Advanced Anti-DDoS protection using the AWS WAF Anti-DDoS managed rule group

The `AWSManagedRulesAntiDDoSRuleSet` managed rule group is the most advanced tier of Anti-DDoS protections available in AWS WAF.

Note

You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

AWS WAF Anti-DDoS protection components

The main components for implementing advanced Anti-DDoS protection in AWS WAF include the following:

`AWSManagedRulesAntiDDoSRuleSet` – Detects, labels, and challenges requests that are likely participating in a DDoS attack. It also labels all requests to a protected resource during an event. For details about the rule group's rules and labels, see AWS WAF Distributed Denial of Service (DDoS) prevention rule group. To use this rule group, include it in your protection pack (web ACL) using a managed rule group reference statement. For information, see Adding the Anti-DDoS managed rule group to your protection pack (web ACL).

- **Web ACL traffic overview dashboards** – Provide monitoring for DDoS activity and anti-DDoS responses in the console. For more information, see Traffic overview dashboards for protection packs (web ACLs).
- **Logging and metrics** – Allow you to monitor traffic and understand Anti-DDoS protection effects. Configure logs, Amazon Security Lake data collection, and Amazon CloudWatch metrics for your protection pack (web ACL). For information about these options, see Logging AWS WAF

protection pack (web ACL) traffic, Monitoring with Amazon CloudWatch, and What is Amazon Security Lake?.

- **Labels and label matching rules** – Allow you to customize handling of web requests identified by the Anti-DDoS managed rule group. For any rule in `AWSManagedRulesAntiDDoSRuleSet`, you can switch to count mode and match against added labels. For more information, see Label match rule statement and Web request labeling in AWS WAF.
- **Custom requests and responses** – Allow you to add custom headers to allowed requests and send custom responses for blocked requests. Pair label matching with AWS WAF custom request and response features. For more information, see Customized web requests and responses in AWS WAF.

Adding the Anti-DDoS managed rule group to your protection pack (web ACL)

This section explains how to add and configure the `AWSManagedRulesAntiDDoSRuleSet` rule group.

To configure the Anti-DDoS managed rule group, you provide settings that include how sensitive the rule group is to DDoS attacks and the actions that it takes on requests that are or might be participating in the attacks. This configuration is in addition to the normal configuration for a managed rule group.

For the rule group description and rules and labels listing, see AWS WAF Distributed Denial of Service (DDoS) prevention rule group.

This guidance is intended for users who know generally how to create and manage AWS WAF protection packs (web ACLs), rules, and rule groups. Those topics are covered in prior sections of this guide. For basic information about how to add a managed rule group to your protection pack (web ACL), see Adding a managed rule group to a protection pack (web ACL) through the console.

Follow best practices

Use the Anti-DDoS rule group in accordance with the best practices at Best practices for intelligent threat mitigation in AWS WAF.

To use the `AWSManagedRulesAntiDDoSRuleSet` rule group in your protection pack (web ACL)

1. Add the AWS managed rule group, `AWSManagedRulesAntiDDoSRuleSet` to your protection pack (web ACL), and **Edit** the rule group settings before saving.

> **Note**
>
> You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

2. In the **Rule group configuration** pane, provide any custom configuration for the `AWSManagedRulesAntiDDoSRuleSet` rule group.

   a. For **Block sensitivity level**, specify how sensitive you want the rule `DDoSRequests` to be when matching on the rule group's DDoS suspicion labeling. The higher the sensitivity, the lower the levels of labeling that the rule matches:

   - Low sensitivity is less sensitive, causing the rule to match only on the most obvious participants in an attack, which have the high suspicion label `awswaf:managed:aws:anti-ddos:high-suspicion-ddos-request`.
   - Medium sensitivity causes the rule to match on the medium and high suspicion labels.
   - High sensitivity causes the rule to match on all of the suspicion labels: low, medium, and high.

   This rule provides the most severe handling of web requests that are suspected of participating in DDoS attacks.

   b. For **Enable challenge**, choose whether to enable the rules `ChallengeDDoSRequests` and `ChallengeAllDuringEvent`, which by default apply the Challenge action to matching requests.

   These rules provide request handling that's intended to permit legitimate users to proceed with their requests while blocking participants in the DDoS attack. You can override their action settings to Allow or Count or you can disable their use entirely.

   If you enable these rules, then provide any additional configuration that you want:

   - For **Challenge sensitivity level**, specify how sensitive you want the rule `ChallengeDDoSRequests` to be.

     The higher the sensitivity, the lower the levels of labeling that the rule matches:

  - Low sensitivity is less sensitive, causing the rule to match only on the most obvious participants in an attack, which have the high suspicion label `awswaf:managed:aws:anti-ddos:high-suspicion-ddos-request`.
  - Medium sensitivity causes the rule to match on the medium and high suspicion labels.
  - High sensitivity causes the rule to match on all of the suspicion labels: low, medium, and high.
- For **Exempt URI regular expressions**, provide a regular expression that matches against URIs for web requests that can't handle a silent browser challenge. The Challenge action will effectively block requests from URIs that are missing the challenge token unless they can handle the silent browser challenge.

  The Challenge action can only be handled properly by a client that's expecting HTML content. For more information about how the action works, see CAPTCHA and Challenge action behavior.

  Review the default regular expression and update it as needed. The rules use the specified regular expression to identify request URIs that can't handle the Challenge action and prevent the rules from sending a challenge back. Requests that you exclude in this way can only be blocked by the rule group with the rule `DDoSRequests`.

  The default expression that's provided in the console covers most use cases, but you should review and adapt it for your application.

  AWS WAF supports the pattern syntax used by the PCRE library `libpcre` with some exceptions. The library is documented at PCRE - Perl Compatible Regular Expressions. For information about AWS WAF support, see Supported regular expression syntax in AWS WAF.

3. Provide any additional configuration that you want for the rule group and save the rule.

> **Note**
>
> AWS recommends against using a scope-down statement with this managed rule group. The scope-down statement limits the requests that the rule group observes, and so can result in an inaccurate traffic baseline and diminished DDoS event detection. The scope-down statement option is available for all of the managed rule

group statements, but should not be used for this one. For information about scope-down statements, see Using scope-down statements in AWS WAF.

4. In the **Set rule priority** page, move the new anti-DDoS managed rule group rule up so that it runs only after any Allow action rules that you have and before any other rules. This gives the rule group the ability to track the most traffic for Anti-DDoS protection.
5. Save your changes to the protection pack (web ACL).

Before you deploy your anti-DDoS implementation for production traffic, test and tune it in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune the rules in count mode with your production traffic before enabling them. See the section that follows for guidance.

Testing and deploying Anti-DDoS

You will want to configure and test AWS WAF Distributed Denial of Service (DDoS) prevention before deploying the feature. This section provides general guidance for configuring and testing, however the specific steps that you choose to follow will depend on your needs, resources, and web requests that you receive.

This information is in addition to the general information about testing and tuning provided at Testing and tuning your AWS WAF protections.

Note

AWS Managed Rules are designed to protect you from common web threats. When used in accordance with the documentation, AWS Managed Rules rule groups add another layer of security for your applications. However, AWS Managed Rules rule groups aren't intended as a replacement for your security responsibilities, which are determined by the AWS resources that you select. Refer to the Shared Responsibility Model to ensure that your resources in AWS are properly protected.

Production traffic risk

Test and tune your anti-DDoS implementation in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune the rules in count mode with your production traffic before enabling them.

This guidance is intended for users who know generally how to create and manage AWS WAF protection packs (web ACLs), rules, and rule groups. Those topics are covered in prior sections of this guide.

To configure and test an AWS WAF Distributed Denial of Service (DDoS) prevention implementation

Perform these steps first in a test environment, then in production.

1. **Add the AWS WAF Distributed Denial of Service (DDoS) prevention managed rule group in count mode**

   Note

   You are charged additional fees when you use this managed rule group. For more information, see AWS WAF Pricing.

   Add the AWS Managed Rules rule group `AWSManagedRulesAntiDDoSRuleSet` to a new or existing protection pack (web ACL) and configure it so that it doesn't alter the current protection pack (web ACL) behavior. For details about the rules and labels for this rule group, see AWS WAF Distributed Denial of Service (DDoS) prevention rule group.

   - When you add the managed rule group, edit it and do the following:
     - In the **Rule group configuration** pane, provide the details needed to perform anti-DDoS activities for your web traffic. For more information, see Adding the Anti-DDoS managed rule group to your protection pack (web ACL).
     - In the **Rules** pane, open the **Override all rule actions** dropdown and choose **Count**. With this configuration, AWS WAF evaluates requests against all of the rules in the rule group and only counts the matches that result, while still adding labels to requests. For more information, see Overriding rule actions in a rule group.

With this override, you can monitor the potential impact of the Anti-DDoS managed rules to determine whether you want to make modifications, such as expanding the regex for the URIs that can't handle a silent browser challenge.

- Position the rule group so that it's evaluated as early as possible, immediately after any rules that allow traffic. Rules are evaluated in ascending numeric priority order. The console sets the order for you, starting at the top of your rule list. For more information, see Setting rule priority.

2. **Enable logging and metrics for the protection pack (web ACL)**

   As needed, configure logging, Amazon Security Lake data collection, request sampling, and Amazon CloudWatch metrics for the protection pack (web ACL). You can use these visibility tools to monitor the interaction of the Anti-DDoS managed rule group with your traffic.

   - For information about configuring and using logging, see Logging AWS WAF protection pack (web ACL) traffic.
   - For information about Amazon Security Lake, see What is Amazon Security Lake? and Collecting data from AWS services in the *Amazon Security Lake user guide*.
   - For information about Amazon CloudWatch metrics, see Monitoring with Amazon CloudWatch.
   - For information about web request sampling, see Viewing a sample of web requests.

3. **Associate the protection pack (web ACL) with a resource**

   If the protection pack (web ACL) isn't already associated with a test resource, associate it. For information, see Associating or disassociating protection with an AWS resource.

4. **Monitor traffic and Anti-DDoS rule matches**

   Make sure that your normal traffic is flowing and that the Anti-DDoS managed rule group rules are adding labels to matching web requests. You can see the labels in the logs and see the Anti-DDoS and label metrics in the Amazon CloudWatch metrics. In the logs, the rules that you've overridden to count in the rule group show up in the `ruleGroupList` with `action` set to count, and with `overriddenAction` indicating the configured rule action that you overrode.

5. **Customize Anti-DDoS web request handling**

   As needed, add your own rules that explicitly allow or block requests, to change how Anti-DDoS rules would otherwise handle them.

For example, you can use Anti-DDoS labels to allow or block requests or to customize request handling. You can add a label match rule after the Anti-DDoS managed rule group to filter labeled requests for the handling that you want to apply. After testing, keep the related Anti-DDoS rules in count mode, and maintain the request handling decisions in your custom rule.

6. **Remove test rules and configure Anti-DDoS settings**

   Review your testing results to determine which Anti-DDoS rules you want to keep in count mode for monitoring only. For any rules you want to run with active protection, disable count mode in the protection pack (web ACL) rule group configuration to allow them to perform their configured actions. Once you've finalized these settings, remove any temporary test label match rules while retaining any custom rules you created for production use. For additional Anti-DDoS configuration considerations, see Best practices for intelligent threat mitigation in AWS WAF.

7. **Monitor and tune**

   To be sure that web requests are being handled as you want, closely monitor your traffic after you enable the Anti-DDoS functionality that you intend to use. Adjust the behavior as needed with the rules count override on the rule group and with your own rules.

Best Practices for Anti-DDoS

- **Enable protection during normal traffic periods** – This allows the protection to establish baseline traffic patterns before responding to attacks. Add protection when you are not experiencing an attack and allow time for baseline establishment.
- **Monitor metrics regularly** – Review CloudWatch metrics to understand traffic patterns and protection effectiveness.
- **Consider proactive mode for critical applications** – While reactive mode is recommended for most use cases, consider using proactive mode for applications that require continuous protection against known threats.
- **Test in staging environments** – Before enabling protection in production, test and tune settings in a staging environment to understand the impact on legitimate traffic.

Client application integrations in AWS WAF

This section explains how to use the intelligent threat integration APIs and JavaScript CAPTCHA integration API with your AWS WAF features.

Use AWS WAF client application integration APIs to couple client-side protections with your AWS server-side protection pack (web ACL) protections, to help verify that the client applications that send web requests to your protected resources are the intended clients and that your end users are human beings.

Use the client integrations to manage silent browser challenges and CAPTCHA puzzles, obtain tokens with proof of successful browser and end user responses, and to include these tokens in requests to your protected endpoints. For general information about AWS WAF tokens, see Token use in AWS WAF intelligent threat mitigation.

Combine your client integrations with protection pack (web ACL) protections that require valid tokens for access to your resources. You can use rule groups that check and monitor challenge tokens, like the ones listed in the next section, at Intelligent threat integration and AWS Managed Rules, and you can use the CAPTCHA and Challenge rule actions to check, as described in CAPTCHA and Challenge in AWS WAF.

Use the AWS WAF SDK to improve the detection accuracy of the targeted protection level of Bot Control. Add the challenge `script` tag to your key pages (login, checkout, and account creation). For more guidance on when and how to integrate, see Integrating the client application integration SDKs.

AWS WAF provides two levels of integration for JavaScript applications, and one for mobile applications:

- **Intelligent threat integration** – Verify the client application and provide AWS token acquisition and management. This is similar to the functionality provided by the AWS WAF Challenge rule action. This functionality fully integrates your client application with the `AWSManagedRulesACFPRuleSet` managed rule group, the `AWSManagedRulesATPRuleSet` managed rule group, and the targeted protection level of the `AWSManagedRulesBotControlRuleSet` managed rule group.

  The intelligent threat integration APIs use the AWS WAF silent browser challenge to help ensure that login attempts and other calls to your protected resource are only allowed after the client has acquired a valid token. The APIs manage token authorization for your client application

sessions and gather information about the client to help determine whether it's being operated by a bot or by a human being.

> **Note**
>
> This is available for JavaScript and for Android and iOS mobile applications.

- **CAPTCHA integration** – Verify end users with customized CAPTCHA puzzle that you manage in your application. This is similar to the functionality provided by the AWS WAF CAPTCHA rule action, but with added control over the puzzle placement and behavior.

  This integration leverages the JavaScript intelligent threat integration to run silent challenges and provide AWS WAF tokens to the customer's page.

  > **Note**
  >
  > This is available for JavaScript applications.

Topics

- Intelligent threat integration and AWS Managed Rules
- Accessing the AWS WAF client application integration APIs
- AWS WAF JavaScript integrations
- AWS WAF mobile application integration

Intelligent threat integration and AWS Managed Rules

This section explains how the intelligent threat integration APIs work with the AWS Managed Rules rule groups.

The intelligent threat integration APIs work with protection packs (web ACLs) that use the intelligent threat rule groups to enable the full functionality of these advanced managed rule groups.

- AWS WAF Fraud Control account creation fraud prevention (ACFP) managed rule group `AWSManagedRulesACFPRuleSet`.

Account creation fraud is an online illegal activity in which an attacker creates invalid accounts in your application for purposes such as receiving sign-up bonuses or impersonating someone. The ACFP managed rule group provides rules to block, label, and manage requests that might be part of fraudulent account creation attempts. The APIs enable fine-tuned client browser verification and human interactivity information that the ACFP rules use to separate valid client traffic from malicious traffic.

For more information, see AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group and AWS WAF Fraud Control account creation fraud prevention (ACFP).

- AWS WAF Fraud Control account takeover prevention (ATP) managed rule group `AWSManagedRulesATPRuleSet`.

  Account takeover is an online illegal activity in which an attacker gains unauthorized access to a person's account. The ATP managed rule group provides rules to block, label, and manage requests that might be part of malicious account takeover attempts. The APIs enable fine-tuned client verification and behavior aggregation that the ATP rules use to separate valid client traffic from malicious traffic.

  For more information, see AWS WAF Fraud Control account takeover prevention (ATP) rule group and AWS WAF Fraud Control account takeover prevention (ATP).

- Targeted protection level of the AWS WAF Bot Control managed rule group `AWSManagedRulesBotControlRuleSet`.

  Bots run from self-identifying and useful ones, such as most search engines and crawlers, to malicious bots that operate against your website and don't self-identify. The Bot Control managed rule group provides rules to monitor, label, and manage the bot activity in your web traffic. When you use the targeted protection level of this rule group, the targeted rules use the client session information that the APIs provide to better detect malicious bots.

  For more information, see AWS WAF Bot Control rule group and AWS WAF Bot Control.

To add one of these managed rule groups to your protection pack (web ACL), see the procedures Adding the ACFP managed rule group to your web ACL, Adding the ATP managed rule group to your protection pack (web ACL), and Adding the AWS WAF Bot Control managed rule group to your web ACL.

> **Note**
>
> The managed rule groups currently don't block requests that are missing tokens. In order to block requests that are missing tokens, after you implement your application integration APIs, follow the guidance at Blocking requests that don't have a valid AWS WAF token.

Accessing the AWS WAF client application integration APIs

This section explains where to find the application integration APIs in the AWS WAF console.

The JavaScript integration APIs are generally available, and you can use them for your browsers and other devices that execute JavaScript.

AWS WAF offers custom intelligent threat integration SDKs for Android and iOS mobile apps.

- For Android mobile and TV apps, the SDKs work for Android API version 23 (Android version 6) and later. For information about Android versions, see SDK Platform release notes.
- For iOS mobile apps, the SDKs work for iOS version 13 and later. For information about iOS versions, see iOS & iPadOS Release Notes.
- For Apple TV apps, the SDKs work for tvOS version 14 or later. For information about tvOS versions, see tvOS Release Notes.

To access the integration APIs through the console

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. Choose **Application integration** in the navigation pane, and then choose the tab you're interested in.

   - **Intelligent threat integration** is available for JavaScript and mobile applications.

     The tab contains the following:

     - A list of the protection packs (web ACLs) that are enabled for intelligent threat application integration. The list includes each protection pack (web ACL) that uses the `AWSManagedRulesACFPRuleSet` managed rule group, the `AWSManagedRulesATPRuleSet` managed rule group, or the targeted protection level of the `AWSManagedRulesBotControlRuleSet` managed rule group. When you implement

the intelligent threat APIs, you use the integration URL for the protection pack (web ACL) that you want to integrate with.

- The APIs that you have access to. The JavaScript APIs are always available. For access to the mobile SDKs, contact support at Contact AWS.

- **CAPTCHA integration** is available for JavaScript applications.

  The tab contains the following:

  - The integration URL for use in your integration.
  - The API keys that you've created for your client application domains. Your use of the CAPTCHA API requires an encrypted API key that gives clients the right to access AWS WAF CAPTCHA from their domains. For each client that you integrate with, use an API key that contains the client's domain. For more information these requirements and about managing these keys, see Managing API keys for the JS CAPTCHA API.

AWS WAF JavaScript integrations

This section explains how to use the AWS WAF JavaScript integrations.

You can use the JavaScript integration APIs to implement AWS WAF application integrations in your browsers and other devices that execute JavaScript.

CAPTCHA puzzles and silent challenges can only run when browsers are accessing HTTPS endpoints. Browser clients must be running in secure contexts in order to acquire tokens.

- The intelligent threat APIs let you manage token authorization through a silent client-side browser challenge, and to include the tokens in the requests that you send to your protected resources.
- The CAPTCHA integration API adds to the intelligent threat APIs, and lets you customize the placement and characteristics of the CAPTCHA puzzle in your client applications. This API leverages the intelligent threat APIs to acquire AWS WAF tokens for use in the page after the end user successfully completes the CAPTCHA puzzle.

By using these integrations, you ensure that the remote procedure calls by your client contain a valid token. When these integration APIs are in place on your application's pages, you can implement mitigating rules in your protection pack (web ACL), such as blocking requests that don't contain a valid token. You can also implement rules that enforce the use of the tokens that your client applications obtain, by using the Challenge or CAPTCHA actions in your rules.

Example implementation of intelligent threat APIs

The following listing shows basic components of a typical implementation of the intelligent threat APIs in a web application page.

```
<head>
<script type="text/javascript" src="protection pack (web ACL) integration URL/
challenge.js" defer></script>
</head>
<script>
const login_response = await AwsWafIntegration.fetch(login_url, {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: login_body
  });
</script>
```

Example implementation of CAPTCHA JavaScript API

The CAPTCHA integration API lets you customize your end users' CAPTCHA puzzle experience. The CAPTCHA integration leverages the JavaScript intelligent threat integration, for browser verification and token management, and adds a function for configuring and rendering the CAPTCHA puzzle.

The following listing shows basic components of a typical implementation of the CAPTCHA JavaScript API in a web application page.

```
<head>
    <script type="text/javascript" src="<Integration URL>/jsapi.js" defer></script>
</head>

<script type="text/javascript">
    function showMyCaptcha() {
        var container = document.querySelector("#my-captcha-container");

        AwsWafCaptcha.renderCaptcha(container, {
            apiKey: "...API key goes here...",
            onSuccess: captchaExampleSuccessFunction,
            onError: captchaExampleErrorFunction,
            ...other configuration parameters as needed...
```

```
        });
    }

    function captchaExampleSuccessFunction(wafToken) {
        // Use WAF token to access protected resources
        AwsWafIntegration.fetch("...WAF-protected URL...", {
            method: "POST",
            ...
        });
    }

    function captchaExampleErrorFunction(error) {
        /* Do something with the error */
    }
</script>

<div id="my-captcha-container">
    <!-- The contents of this container will be replaced by the captcha widget -->
</div>
```

Topics

- Providing domains for use in the tokens
- Using the JavaScript API with content security policies
- Using the intelligent threat JavaScript API
- Using the CAPTCHA JavaScript API

Providing domains for use in the tokens

This section explains how to provide additional domains for tokens.

By default, when AWS WAF creates a token, it uses the host domain of the resource that's associated with the protection pack (web ACL). You can provide additional domains for the tokens that AWS WAF creates for the JavaScript APIs. To do this, configure the global variable `window.awsWafCookieDomainList`, with one or more token domains.

When AWS WAF creates a token, it uses the most appropriate, shortest domain from among the combination of the domains in `window.awsWafCookieDomainList` and the host domain of the resource that's associated with the protection pack (web ACL).

Example settings:

```
window.awsWafCookieDomainList = ['.aws.amazon.com']
```

```
window.awsWafCookieDomainList = ['.aws.amazon.com', 'abc.aws.amazon.com']
```

You can't use public suffixes in this list. For example, you can't use `gov.au` or `co.uk` as token domains in the list.

The domains that you specify in this list must be compatible with your other domains and domain configurations:

- The domains must be ones that AWS WAF will accept, based on the protected host domain and the token domain list that's configured for the protection pack (web ACL). For more information, see AWS WAF protection pack (web ACL) token domain list configuration.
- If you use the JavaScript CAPTCHA API, at least one domain in your CAPTCHA API key must be an exact match for one of the token domains in `window.awsWafCookieDomainList` or it must be the apex domain of one of those token domains.

  For example, for the token domain `mySubdomain.myApex.com`, the API key `mySubdomain.myApex.com` is an exact match and the API key `myApex.com` is the apex domain. Either key matches the token domain.

  For more information about the API keys, see Managing API keys for the JS CAPTCHA API.

If you use the `AWSManagedRulesACFPRuleSet` managed rule group, you might configure a domain that matches the one in the account creation path that you provided to the rule group configuration. For more information about this configuration, see Adding the ACFP managed rule group to your web ACL.

If you use the `AWSManagedRulesATPRuleSet` managed rule group, you might configure a domain that matches the one in the login path that you provided to the rule group configuration. For more information about this configuration, see Adding the ATP managed rule group to your protection pack (web ACL).

Using the JavaScript API with content security policies

This section provides an example configuration to allowlist the AWS WAF apex domain.

If you apply content security policies (CSP) to your resources, for your JavaScript implementation to work, you need to allowlist the AWS WAF apex domain `awswaf.com`. The JavaScript SDKs make

calls to different AWS WAF endpoints, so allowlisting this domain provides the permissions that the SDKs need to operate.

The following shows an example configuration to allowlist the AWS WAF apex domain:

```
connect-src 'self' https://*.awswaf.com;
script-src 'self' https://*.awswaf.com;
script-src-elem 'self' https://*.awswaf.com;
```

If you try to use the JavaScript SDKs with resources that use CSP, and you haven't allowlisted the AWS WAF domain, you'll receive errors like the following:

```
Refused to load the script ...awswaf.com/<> because it violates the following Content
 Security Policy directive: “script-src ‘self’
```

Using the intelligent threat JavaScript API

This section provides instructions for using the intelligent threat JavaScript API in your client application.

The intelligent threat APIs provide operations for running silent challenges against the user's browser, and for handling the AWS WAF tokens that provide proof of successful challenge and CAPTCHA responses.

Implement the JavaScript integration first in a test environment, then in production. For additional coding guidance, see the sections that follow.

To use the intelligent threat APIs

1. **Install the APIs**

   If you use the CAPTCHA API, you can skip this step. When you install the CAPTCHA API, the script automatically installs the intelligent threat APIs.

   a. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
   b. In the navigation pane, choose **Application integration**. On the **Application integration** page, you can see tabbed options.
   c. Select **Intelligent threat integration**

d. In the tab, select the protection pack (web ACL) that you want to integrate with. The protection pack (web ACL) list includes only protection packs (web ACLs) that use the `AWSManagedRulesACFPRuleSet` managed rule group, the `AWSManagedRulesATPRuleSet` managed rule group, or the targeted protection level of the `AWSManagedRulesBotControlRuleSet` managed rule group.

e. Open the **JavaScript SDK** pane, and copy the script tag for use in your integration.

f. In your application page code, in the `<head>` section, insert the script tag that you copied for the protection pack (web ACL). This inclusion causes your client application to automatically retrieve a token in the background on page load.

```
<head>
    <script type="text/javascript" src="protection pack (web ACL) integration
 URL/challenge.js” defer></script>
<head>
```

This `<script>` listing is configured with the `defer` attribute, but you can change the setting to `async` if you want a different behavior for your page.

2. **(Optional) Add domain configuration for the client's tokens** – By default, when AWS WAF creates a token, it uses the host domain of the resource that’s associated with the protection pack (web ACL). To provide additional domains for the JavaScript APIs, follow the guidance at Providing domains for use in the tokens.

3. **Code your intelligent threat integration** – Write your code to ensure that token retrieval completes before the client sends its requests to your protected endpoints. If you are already using the `fetch` API to make your call, you can substitute the AWS WAF integration `fetch` wrapper. If you don't use the `fetch` API, you can use the AWS WAF integration `getToken` operation instead. For coding guidance, see the following sections.

4. **Add token verification in your protection pack (web ACL)** – Add at least one rule to your protection pack (web ACL) that checks for a valid challenge token in the web requests that your client sends. You can use rule groups that check and monitor challenge tokens, like the targeted level of the Bot Control managed rule group, and you can use the Challenge rule action to check, as described in CAPTCHA and Challenge in AWS WAF.

   The protection pack (web ACL) additions verify that requests to your protected endpoints include the token that you've acquired in your client integration. Requests that include a valid, unexpired token pass the Challenge inspection and do not send another silent challenge to your client.

5. **(Optional) Block requests that are missing tokens** – If you use the APIs with the ACFP managed rule group, the ATP managed rule group, or the targeted rules of the Bot Control rule group, these rules don't block requests that are missing tokens. To block requests that are missing tokens, follow the guidance at Blocking requests that don't have a valid AWS WAF token.

Topics

- Intelligent threat API specification
- How to use the integration fetch wrapper
- How to use the integration getToken

Intelligent threat API specification

This section lists the specification for the methods and properties of the intelligent threat mitigation JavaScript APIs. Use these APIs for intelligent threat and CAPTCHA integrations.

`AwsWafIntegration.fetch()`

Sends the HTTP `fetch` request to the server using the AWS WAF integration implementation.

`AwsWafIntegration.getToken()`

Retrieves the stored AWS WAF token and stores it in a cookie on the current page with name `aws-waf-token`, and the value set to the token value.

`AwsWafIntegration.hasToken()`

Returns a boolean indicating whether the `aws-waf-token` cookie currently holds an unexpired token.

If you're also using the CAPTCHA integration, see the specification for that at CAPTCHA JavaScript API specification.

How to use the integration `fetch` wrapper

This section provides instructions for using the integration `fetch` wrapper.

You can use the AWS WAF `fetch` wrapper by changing your normal `fetch` calls to the `fetch` API under the `AwsWafIntegration` namespace. The AWS WAF wrapper supports all of the same

options as the standard JavaScript `fetch` API call and adds the token handling for the integration. This approach is generally the simplest way to integrate your application.

Before the wrapper implementation

The following example listing shows standard code before implementing the `AwsWafIntegration` `fetch` wrapper.

```
const login_response = await fetch(login_url, {
     method: 'POST',
     headers: {
       'Content-Type': 'application/json'
     },
     body: login_body
   });
```

After the wrapper implementation

The following listing shows the same code with the `AwsWafIntegration` `fetch` wrapper implementation.

```
const login_response = await AwsWafIntegration.fetch(login_url, {
     method: 'POST',
     headers: {
       'Content-Type': 'application/json'
     },
     body: login_body
   });
```

How to use the integration `getToken`

This section explains how to use the `getToken` operation.

AWS WAF requires your requests to protected endpoints to include the cookie named `aws-waf-token` with the value of your current token.

The `getToken` operation is an asynchronous API call that retrieves the AWS WAF token and stores it in a cookie on the current page with name `aws-waf-token`, and the value set to the token value. You can use this token cookie as needed in your page.

When you call `getToken`, it does the following:

- If an unexpired token is already available, the call returns it immediately.
- Otherwise, the call retrieves a new token from the token provider, waiting for up to 2 seconds for the token acquisition workflow to complete before timing out. If the operation times out, it throws an error, which your calling code must handle.

The `getToken` operation has an accompanying `hasToken` operation, which indicates whether the `aws-waf-token` cookie currently holds an unexpired token.

`AwsWafIntegration.getToken()` retrieves a valid token and stores it as a cookie. Most client calls automatically attach this cookie, but some don't. For example, calls made across host domains don't attach the cookie. In the implementation details that follow, we show how to work with both types of client calls.

Basic `getToken` implementation, for calls that attach the `aws-waf-token` cookie

The following example listing shows standard code for implementing the `getToken` operation with a login request.

```
const login_response = await AwsWafIntegration.getToken()
     .catch(e => {
         // Implement error handling logic for your use case
     })
     // The getToken call returns the token, and doesn't typically require special
 handling
     .then(token => {
         return loginToMyPage()
     })

 async function loginToMyPage() {
     // Your existing login code
 }
```

Submit form only after token is available from `getToken`

The following listing shows how to register an event listener to intercept form submissions until a valid token is available for use.

```
<body>
   <h1>Login</h1>
   <p></p>
```

```
   <form id="login-form" action="/web/login" method="POST" enctype="application/x-www-
form-urlencoded">
     <label for="input_username">USERNAME</label>
     <input type="text" name="input_username" id="input_username"><br>
     <label for="input_password">PASSWORD</label>
     <input type="password" name="input_password" id="input_password"><br>
     <button type="submit">Submit<button>
   </form>

 <script>
   const form = document.querySelector("#login-form");

   // Register an event listener to intercept form submissions
   form.addEventListener("submit", (e) => {
       // Submit the form only after a token is available
       if (!AwsWafIntegration.hasToken()) {
           e.preventDefault();
           AwsWafIntegration.getToken().then(() => {
               e.target.submit();
           }, (reason) => { console.log("Error:"+reason) });
         }
     });
 </script>
 </body>
```

Attaching the token when your client doesn't attach the `aws-waf-token` cookie by default

`AwsWafIntegration.getToken()` retrieves a valid token and stores it as a cookie, but not all client calls attach this cookie by default. For example, calls made across host domains don't attach the cookie.

The `fetch` wrapper handles these cases automatically, but if you aren't able to use the `fetch` wrapper, you can handle this by using a custom `x-aws-waf-token` header. AWS WAF reads tokens from this header, in addition to reading them from the `aws-waf-token` cookie. The following code shows an example of setting the header.

```
const token = await AwsWafIntegration.getToken();
const result = await fetch('/url', {
    headers: {
        'x-aws-waf-token': token,
    },
});
```

By default, AWS WAF only accepts tokens that contain the same domain as the requested host domain. Any cross-domain tokens require corresponding entries in the protection pack (web ACL) token domain list. For more information, see AWS WAF protection pack (web ACL) token domain list configuration.

For additional information about cross-domain token use, see aws-samples/aws-waf-bot-control-api-protection-with-captcha.

Using the CAPTCHA JavaScript API

This section provides instructions for using the CAPTCHA integration API.

The CAPTCHA JavaScript API allows you to configure the CAPTCHA puzzle and place it where you want in your client application. This API leverages the features of the intelligent threat JavaScript APIs to acquire and use AWS WAF tokens after an end user successfully completes a CAPTCHA puzzle.

Implement the JavaScript integration first in a test environment, then in production. For additional coding guidance, see the sections that follow.

To use the CAPTCHA integration API

1. **Install the API**

   a. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.

   b. In the navigation pane, choose **Application integration**. On the **Application integration** page, you can see tabbed options.

   c. Select **CAPTCHA integration**.

   d. Copy the listed JavaScript integration script tag for use in your integration.

   e. In your application page code, in the <head> section, insert the script tag that you copied. This inclusion makes the CAPTCHA puzzle available for configuration and use.

```
<head>
    <script type="text/javascript" src="integrationURL/jsapi.js" defer></
script>
</head>
```

This `<script>` listing is configured with the `defer` attribute, but you can change the setting to `async` if you want a different behavior for your page.

The CAPTCHA script also automatically loads the intelligent threat integration script if it isn't already present. The intelligent threat integration script causes your client application to automatically retrieve a token in the background on page load, and provides other token management functionality that you need for your use of the CAPTCHA API.

2. **(Optional) Add domain configuration for the client's tokens** – By default, when AWS WAF creates a token, it uses the host domain of the resource that's associated with the protection pack (web ACL). To provide additional domains for the JavaScript APIs, follow the guidance at Providing domains for use in the tokens.
3. **Get the encrypted API key for the client** – The CAPTCHA API requires an encrypted API key that contains a list of valid client domains. AWS WAF uses this key to verify that the client domain you're using with the integration is approved to use AWS WAF CAPTCHA. To generate your API key, follow the guidance at Managing API keys for the JS CAPTCHA API.
4. **Code your CAPTCHA widget implementation** – Implement the `renderCaptcha()` API call in your page, at the location where you want to use it. For information about configuring and using this function, see the following sections, CAPTCHA JavaScript API specification and How to render the CAPTCHA puzzle.

   The CAPTCHA implementation integrates with the intelligent threat integration APIs for token management and to run fetch calls that use the AWS WAF tokens. For guidance about using these APIs, see Using the intelligent threat JavaScript API.
5. **Add token verification in your protection pack (web ACL)** – Add at least one rule to your protection pack (web ACL) that checks for a valid CAPTCHA token in the web requests that your client sends. You can use the CAPTCHA rule action to check, as described in CAPTCHA and Challenge in AWS WAF.

   The protection pack (web ACL) additions verify that requests going to your protected endpoints include the token that you've acquired in your client integration. Requests that include a valid, unexpired CAPTCHA token pass the CAPTCHA rule action inspection and do not present your end user with another CAPTCHA puzzle.

After you've implemented the JavaScript API, you can review the CloudWatch metrics for CAPTCHA puzzle attempts and solutions. For metrics and dimension details, see Account metrics and dimensions.

Topics

- CAPTCHA JavaScript API specification
- How to render the CAPTCHA puzzle
- Handling a CAPTCHA response from AWS WAF
- Managing API keys for the JS CAPTCHA API

CAPTCHA JavaScript API specification

This section lists the specification for the methods and properties of the CAPTCHA JavaScript APIs. Use the CAPTCHA JavaScript APIs to run custom CAPTCHA puzzles in your client applications.

This API builds on the intelligent threat APIs, which you use to configure and manage AWS WAF token acquisition and use. See Intelligent threat API specification. .

`AwsWafCaptcha.renderCaptcha(container, configuration)`

Presents an AWS WAF CAPTCHA puzzle to the end user and, upon success, updates the client token with the CAPTCHA validation. This is available only with the CAPTCHA integration. Use this call along with the intelligent threat APIs to manage token retrieval and to provide the token in your `fetch` calls. See the intelligent threat APIs at Intelligent threat API specification.

Unlike the CAPTCHA interstitial that AWS WAF sends, the CAPTCHA puzzle rendered by this method displays the puzzle immediately, without an initial title screen.

`container`

The `Element` object for the target container element on the page. This is commonly retrieved by calling `document.getElementById()` or `document.querySelector()`.

Required: Yes

Type: `Element`

configuration

An object containing CAPTCHA configuration settings, as follows:

`apiKey`

The encrypted API key that enables permissions for the client's domain. Use the AWS WAF console to generate your API keys for your client domains. You can use one key for up to five domains. For information, see Managing API keys for the JS CAPTCHA API.

Required: Yes

Type: `string`

`onSuccess: (wafToken: string) => void;`

Called with a valid AWS WAF token when the end user successfully completes a CAPTCHA puzzle. Use the token in the requests that you send to the endpoints that you protect with an AWS WAF protection pack (web ACL). The token provides proof and the timestamp of the latest successful puzzle completion.

Required: Yes

`onError?: (error: CaptchaError) => void;`

Called with an error object when an error occurs during the CAPTCHA operation.

Required: No

`CaptchaError` class definition – The `onError` handler supplies an error type with the following class definition.

```
CaptchaError extends Error {
    kind: "internal_error" | "network_error" | "token_error" | "client_error";
    statusCode?: number;
}
```

- `kind` – The kind of error returned.
- `statusCode` – The HTTP status code, if available. This is used by `network_error` if the error is due to an HTTP error.

`onLoad?: () => void;`

Called when a new CAPTCHA puzzle loads.

Required: No

`onPuzzleTimeout?: () => void;`

Called when a CAPTCHA puzzle isn't completed before it expires.

Required: No

`onPuzzleCorrect?: () => void;`

Called when a correct answer is provided to a CAPTCHA puzzle.

Required: No

`onPuzzleIncorrect?: () => void;`

Called when an incorrect answer is provided to a CAPTCHA puzzle.

Required: No

`defaultLocale`

The default locale to use for the CAPTCHA puzzle. The written instructions for CAPTCHA puzzles are available in Arabic (ar-SA), simplified Chinese (zh-CN), Dutch (nl-NL), English (en-US), French (fr-FR), German (de-DE), Italian (it-IT), Japanese (ja-JP), Brazilian Portuguese (pt-BR), Spanish (es-ES), and Turkish (tr-TR). Audio instructions are available for all of the written languages except Chinese and Japanese, which default to English. To change the default language, provide the international language and locale code, for example, `ar-SA`.

Default: The language currently in use in the end user's browser

Required: No

Type: `string`

`disableLanguageSelector`

If set to `true`, the CAPTCHA puzzle hides the language selector.

Default: `false`

Required: No

Type: `boolean`

`dynamicWidth`

If set to `true`, the CAPTCHA puzzle changes width for compatibility with the browser window width.

Default: `false`

Required: No

Type: `boolean`

`skipTitle`

If set to `true`, the CAPTCHA puzzle doesn't display the puzzle title heading **Solve the puzzle**.

Default: `false`

Required: No

Type: `boolean`

How to render the CAPTCHA puzzle

This section provides an example `renderCaptcha` implementation.

You can use the AWS WAF `renderCaptcha` call where you want to in your client interface. The call retrieves a CAPTCHA puzzle from AWS WAF, renders it, and sends the results to AWS WAF for verification. When you make the call, you provide the puzzle rendering configuration and the callbacks that you want to run when your end users complete the puzzle. For details about the options, see the preceding section, CAPTCHA JavaScript API specification.

Use this call in conjunction with the token management functionality of the intelligent threat integration APIs. This call gives your client a token that verifies the successful completion of the CAPTCHA puzzle. Use the intelligent threat integration APIs to manage the token and to provide the token in your client's calls to the endpoints that are protected with AWS WAF protection packs (web ACLs). For information about the intelligent threat APIs, see Using the intelligent threat JavaScript API.

Example implementation

The following example listing shows a standard CAPTCHA implementation, including the placement of the AWS WAF integration URL in the `<head>` section.

This listing configures the `renderCaptcha` function with a success callback that uses the `AwsWafIntegration.fetch` wrapper of the intelligent threat integration APIs. For information about this function, see How to use the integration `fetch` wrapper.

```
<head>
    <script type="text/javascript" src="<Integration URL>/jsapi.js" defer></script>
</head>

<script type="text/javascript">
```

```
    function showMyCaptcha() {
        var container = document.querySelector("#my-captcha-container");

        AwsWafCaptcha.renderCaptcha(container, {
            apiKey: "...API key goes here...",
            onSuccess: captchaExampleSuccessFunction,
            onError: captchaExampleErrorFunction,
            ...other configuration parameters as needed...
        });
    }

    function captchaExampleSuccessFunction(wafToken) {
        // Captcha completed. wafToken contains a valid WAF token. Store it for
        // use later or call AwsWafIntegration.fetch() to use it easily.
        // It will expire after a time, so calling AwsWafIntegration.getToken()
        // again is advised if the token is needed later on, outside of using the
        // fetch wrapper.

        // Use WAF token to access protected resources
        AwsWafIntegration.fetch("...WAF-protected URL...", {
            method: "POST",
            headers: {
                "Content-Type": "application/json",
            },
            body: "{ ... }" /* body content */
        });
    }

    function captchaExampleErrorFunction(error) {
        /* Do something with the error */
    }
</script>

<div id="my-captcha-container">
    <!-- The contents of this container will be replaced by the captcha widget -->
</div>
```

Example configuration settings

The following example listing shows the `renderCaptcha` with non-default settings for the width and the title options.

```
        AwsWafCaptcha.renderCaptcha(container, {
            apiKey: "...API key goes here...",
            onSuccess: captchaExampleSuccessFunction,
            onError: captchaExampleErrorFunction,
            dynamicWidth: true,
            skipTitle: true
        });
```

For full information about the configuration options, see CAPTCHA JavaScript API specification.

Handling a CAPTCHA response from AWS WAF

This section provides an example of handling a CAPTCHA response.

An AWS WAF rule with a CAPTCHA action terminates the evaluation of a matching web request if the request doesn't have a token with a valid CAPTCHA timestamp. If the request is a GET text/html call, the CAPTCHA action then serves the client an interstitial with a CAPTCHA puzzle. When you don't integrate the CAPTCHA JavaScript API, the interstitial runs the puzzle and, if the end user successfully solves it, automatically resubmits the request.

When you integrate the CAPTCHA JavaScript API and customize your CAPTCHA handling, you need to detect the terminating CAPTCHA response, serve your custom CAPTCHA, and then if the end user successfully solves the puzzle, resubmit the client's web request.

The following code example shows how to do this.

> **Note**
>
> The AWS WAF CAPTCHA action response has a status code of HTTP 405, which we use to recognize the CAPTCHA response in this code. If your protected endpoint uses an HTTP 405 status code to communicate any other type of response for the same call, this example code will render a CAPTCHA puzzle for those responses as well.

```
<!DOCTYPE html>
<html>
<head>
    <script type="text/javascript" src="<Integration URL>/jsapi.js" defer></script>
</head>
<body>
```

```
    <div id="my-captcha-box"></div>
    <div id="my-output-box"></div>

    <script type="text/javascript">
    async function loadData() {
        // Attempt to fetch a resource that's configured to trigger a CAPTCHA
        // action if the rule matches. The CAPTCHA response has status=HTTP 405.
        const result = await AwsWafIntegration.fetch("/protected-resource");

        // If the action was CAPTCHA, render the CAPTCHA and return

        // NOTE: If the endpoint you're calling in the fetch call responds with HTTP
 405
        // as an expected response status code, then this check won't be able to tell
 the
        // difference between that and the CAPTCHA rule action response.

        if (result.status === 405) {
            const container = document.querySelector("#my-captcha-box");
            AwsWafCaptcha.renderCaptcha(container, {
                apiKey: "...API key goes here...",
                onSuccess() {
                    // Try loading again, now that there is a valid CAPTCHA token
                    loadData();
                },
            });
            return;
        }

        const container = document.querySelector("#my-output-box");
        const response = await result.text();
        container.innerHTML = response;
    }

    window.addEventListener("load", () => {
        loadData();
    });
    </script>
</body>
</html>
```

Managing API keys for the JS CAPTCHA API

This section provides instructions for generating and deleting API keys.

To integrate AWS WAF CAPTCHA into a a client application with the JavaScript API, you need the JavaScript API integration tag and the encrypted API key for the client domain where you want to run your CAPTCHA puzzle.

The CAPTCHA application integration for JavaScript uses the encrypted API keys to verify that the client application domain has permission to use the AWS WAF CAPTCHA API. When you call the CAPTCHA API from your JavaScript client, you provide an API key with a domain list that includes a domain for the current client. You can list up to 5 domains in a single encrypted key.

API key requirements

The API key that you use in your CAPTCHA integration must contain a domain that applies to the client where you use the key.

- If you specify a `window.awsWafCookieDomainList` in your client's intelligent threat integration, then at least one domain in your API key must be an exact match for one of the token domains in `window.awsWafCookieDomainList` or it must be the apex domain of one of those token domains.

  For example, for the token domain `mySubdomain.myApex.com`, the API key `mySubdomain.myApex.com` is an exact match and the API key `myApex.com` is the apex domain. Either key matches the token domain.

  For information about the setting the token domain list, see Providing domains for use in the tokens.
- Otherwise, the current domain must be contained in the API key. The current domain is the domain that you can see in the browser address bar.

The domains that you use must be ones that AWS WAF will accept, based on the protected host domain and the token domain list that's configured for the web ACL. For more information, see AWS WAF protection pack (web ACL) token domain list configuration.

How to choose the Region for your API key

AWS WAF can generate CAPTCHA API keys in any Region where AWS WAF is available.

As a general rule, you should use the same Region for your CAPTCHA API key as you use for your protection pack (web ACL). If you expect a global audience for a regional protection pack (web ACL), however, you can obtain a CAPTCHA JavaScript integration tag that's scoped to CloudFront and an API key that's scoped to CloudFront, and use them with a regional protection pack (web

ACL). This approach allows clients to load a CAPTCHA puzzle from the Region that's closest to them, which reduces latency.

CAPTCHA API keys that are scoped to Regions other than CloudFront are not supported for use across multiple Regions. They can only be used in the Region they are scoped to.

To generate an API key for your client domains

To obtain the integration URL and generate and retrieve the API keys through the console.

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **Application integration**.
3. In the pane, **protection packs (web ACLs) that are enabled for application integration**, select the Region that you want to use for your API key. You can also select the Region in the **API keys** pane of the **CAPTCHA integration** tab.
4. Choose the tab **CAPTCHA integration**. This tab provides the CAPTCHA JavaScript integration tag, which you can use in your integration, and the API keys listing. Both are scoped to the selected Region.
5. In the **API keys** pane, choose **Generate key**. The key generation dialogue appears.
6. Enter the client domains that you want to include in the key. You can enter up to 5. When you're finished, choose **Generate key**. The interface returns to the CAPTCHA integration tab, where your new key is listed.

   Once created, an API key is immutable. If you need to make changes to a key, generate a new key and use that instead.
7. (Optional) Copy the newly generated key for use in your integration.

You can also use the REST APIs or one of the language-specific AWS SDKs for this work. The REST API calls are CreateAPIKey and ListAPIKeys.

To delete an API key

To delete an API key, you must use the REST API or one of the language specific AWS SDKs. The REST API call is DeleteAPIKey. You can't use the console to delete a key.

After you delete a key, it can take up to 24 hours for AWS WAF to disallow use of the key in all regions.

AWS WAF mobile application integration

This section introduces the topic of using the AWS WAF mobile SDKs to implement AWS WAF intelligent threat integration SDKs for Android and iOS mobile and TV apps. For TV apps, the SDKs are compatible with major smart TV platforms, including Android TV and Apple TV.

- For Android mobile and TV apps, the SDKs work for Android API version 23 (Android version 6) and later. For information about Android versions, see SDK Platform release notes.
- For iOS mobile apps, the SDKs work for iOS version 13 and later. For information about iOS versions, see iOS & iPadOS Release Notes.
- For Apple TV apps, the SDKs work for tvOS version 14 or later. For information about tvOS versions, see tvOS Release Notes.

With the mobile AWS WAF SDK, you can manage token authorization, and include the tokens in the requests that you send to your protected resources. By using the SDKs, you ensure that these remote procedure calls by your client contain a valid token. Additionally, when this integration is in place on your application's pages, you can implement mitigating rules in your protection pack (web ACL), such as blocking requests that don't contain a valid token.

For access to the mobile SDKs, contact support at Contact AWS.

> **Note**
>
> The AWS WAF mobile SDKs aren't available for CAPTCHA customization.

The basic approach for using the SDK is to create a token provider using a configuration object, then to use the token provider to retrieve tokens from AWS WAF. By default, the token provider includes the retrieved tokens in your web requests to your protected resource.

The following is a partial listing of an SDK implementation, which shows the main components. For more detailed examples, see Code examples for the AWS WAF mobile SDK.

iOS

```
let url: URL = URL(string: "protection pack (web ACL) integration URL")!
 let configuration = WAFConfiguration(applicationIntegrationUrl: url, domainName:
 "Domain name")
```

```
let tokenProvider = WAFTokenProvider(configuration)
let token = tokenProvider.getToken()
```

Android

```
URL applicationIntegrationURL = new URL("protection pack (web ACL) integration
 URL");
 String domainName = "Domain name";
 WAFConfiguration configuration =
 WAFConfiguration.builder().applicationIntegrationURL(applicationIntegrationURL).domainName(
 WAFTokenProvider tokenProvider = new WAFTokenProvider(Application context,
 configuration);
 WAFToken token = tokenProvider.getToken();
```

Installing the AWS WAF mobile SDK

This section provides instructions for installing the AWS WAF mobile SDK.

For access to the mobile SDKs, contact support at Contact AWS.

Implement the mobile SDK first in a test environment, then in production.

To install the AWS WAF mobile SDK

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **Application integration**.
3. In the **Intelligent threat integrations** tab, do the following:

    a. In the pane **protection packs (web ACLs) that are enabled for application integration**, locate the protection pack (web ACL) that you're integrating with. Copy and save the protection pack (web ACL) integration URL for use in your implementation. You can also obtain this URL through the API call `GetWebACL`.

    b. Choose the mobile device type and version, then choose **Download**. You can choose any version you like, but we recommend using the latest version. AWS WAF downloads the `zip` file for your device to your standard download location.

4. In your app development environment, unzip the file to a work location of your choice. In the top-level directory of the zip file, locate and open the `README`. Follow the instructions in the `README` file to install the AWS WAF mobile SDK for use in your mobile app code.
5. Program your app according to the guidance in the following sections.

AWS WAF mobile SDK specification

This section lists the SDK objects, operations, and configuration settings for the latest available version of the AWS WAF mobile SDK. For detailed information about how the token provider and operations work for the various combinations of configuration settings, see How the AWS WAF mobile SDK works.

`WAFToken`

Holds an AWS WAF token.

`getValue()`

Retrieves the `String` representation of the `WAFToken`.

`WAFTokenProvider`

Manages tokens in your mobile app. Implement this using a `WAFConfiguration` object.

`getToken()`

If background refresh is enabled, this returns the cached token. If background refresh is disabled, this makes a synchronous, blocking call to AWS WAF to retrieve a new token.

`loadTokenIntoProvider(WAFToken)`

Loads the specified token into the `WAFTokenProvider`, replacing any token that the provider was managing. The token provider takes ownership of the new token and handles refreshing it going forward. This operation also updates the token in the cookie store, if `setTokenCookie` is enabled in the `WAFConfiguration`.

`onTokenReady(WAFTokenResultCallback)`

Instructs the token provider to refresh the token and invoke the provided callback when an active token is ready. The token provider will invoke your callback in a background thread when the token is cached and ready. Call this when your app first loads and also when it comes back to an active state. For more information about returning to an active state, see the section called “Retrieving a token following app inactivity”.

For Android or iOS apps, you can set `WAFTokenResultCallback` to the operation that you want the token provider to invoke when a requested token is ready. Your implementation of `WAFTokenResultCallback` must take the parameters `WAFToken`, `SdkError`. For iOS apps, you can alternately create an inline function.

`storeTokenInCookieStorage(WAFToken)`

Instructs the `WAFTokenProvider` to store the specified AWS WAF token into the SDK's cookie manager. By default, the token is only added to the cookie store when it's first acquired and when it's refreshed. If the application clears the shared cookie store for any reason, the SDK doesn't automatically add the AWS WAF token back until the next refresh.

`WAFConfiguration`

Holds the configuration for the implementation of the `WAFTokenProvider`. When you implement this, you provide your protection pack (web ACL)'s integration URL, the domain name to use in the token, and any non-default settings that you want the token provider to use.

The following list specifies the configuration settings that you can manage in the `WAFConfiguration` object.

`applicationIntegrationUrl`

The application integration URL. Get this from the AWS WAF console or through the `getWebACL` API call.

Required: Yes

Type: App-specific URL. For iOS, see iOS URL. For Android, see java.net URL.

`backgroundRefreshEnabled`

Indicates whether you want the token provider to refresh the token in the background. If you set this, the token provider refreshes your tokens in the background according to the configuration settings that govern automatic token refresh activities.

Required: No

Type: `Boolean`

Default value: TRUE

`domainName`

The domain to use in the token, which is used in token acquisition and cookie storage. For example, `example.com` or `aws.amazon.com`. This is usually the host domain of your resource that's associated with the protection pack (web ACL), where you'll be sending web requests. For the ACFP managed rule group, `AWSManagedRulesACFPRuleSet`, this will usually be a single domain that matches the domain in the account creation path that you provided in the rule group configuration. For the ATP managed rule group, `AWSManagedRulesATPRuleSet`, this will usually be a single domain that matches the domain in the login path that you provided in the rule group configuration.

Public suffixes aren't allowed. For example, you can't use `gov.au` or `co.uk` as the token domain.

The domain must be one that AWS WAF will accept, based on the protected host domain and the protection pack (web ACL)'s token domain list. For more information, see AWS WAF protection pack (web ACL) token domain list configuration.

Required: Yes

Type: `String`

`maxErrorTokenRefreshDelayMsec`

The maximum time in milliseconds to wait before repeating a token refresh after a failed attempt. For each auto-retry for a failed attempt, it will add an exponential backoff up until the given input delay time. This value is used after token retrieval has failed and been retried `maxRetryCount` times.

Required: No

Type: `Integer`

Default value: `5000` (5 seconds)

Minimum value allowed: `1` (1 millisecond)

Maximum value allowed: `30000` (30 seconds)

`maxRetryCount`

The maximum number of retries to perform with exponential backoff when a token is requested.

Required: No

Type: `Integer`

Default value: `Infinity`

Minimum value allowed: `0`

Maximum value allowed: `100`

`setTokenCookie`

Indicates whether you want the SDK's cookie manager to add a token cookie into requests and in other areas.

With a `TRUE` value:

- The cookie manager adds a token cookie to all requests whose path is under the path specified in `tokenCookiePath`.
- The `WAFTokenProvider` operation `loadTokenIntoProvider()` updates the token in the cookie store, in addition to loading it into the token provider.

Required: No

Type: `Boolean`

Default value: `TRUE`

`tokenCookiePath`

Used when `setTokenCookie` is `TRUE`. Indicates the top-level path where you want the SDK's cookie manager to add a token cookie. The manager adds a token cookie to all requests that you send to this path and to all child paths.

For example, if you set this to `/web/login`, then the manager includes the token cookie for everything sent to `/web/login` and any of its child paths, like `/web/login/help`. It doesn't include the token for requests sent to other paths, like `/`, `/web`, or `/web/order`.

Required: No

Type: `String`

Default value: `/`

`tokenRefreshDelaySec`

Used for background refresh. The maximum amount of time in seconds between background token refreshes.

Required: No

Type: `Integer`

Default value: 88

Minimum value allowed: 88

Maximum value allowed: `300` (5 minutes)

AWS WAF mobile SDK errors

This section lists the possible errors for the current AWS WAF mobile SDK version.

`SdkError`

The error type returned when failing to retrieve a token. The Android and iOS SDK have the same error types.

The AWS WAF mobile SDK has the following error types:

`invalidChallenge`

This error is returned when the token server returns invalid challenge data, or the response blob is mutated by an attacker.

`errorInvokingGetChallengeEndpoint`

This error is returned when the token server sends a non-success response code back to the client or when a network error occurs.

`invalidVerifyChallengeResponse`

This error is returned when there is an error retrieving the `aws-waf-token` from the AWS WAF server's verification response, or the server response was tampered with.

`errorInvokingVerifyEndpoint`

This error is returned when the client receives a bad response from the AWS WAF server or network error when verifying the solved challenge.

`internalError`

This error is returned on all other errors that might occur within the SDK itself.

`socketTimeoutException`

This error is often returned when encountering network errors during token retrieval.

This error might be caused by the following:

- Low network bandwidth: Confirm your network connectivity settings
- Mutated Application Integration URL: Confirm that the integration URL is not modified from what appears on the AWS WAF console

How the AWS WAF mobile SDK works

This section explains how the AWS WAF mobile SDK classes, properties, and operations work together.

The mobile SDKs provide you with a configurable token provider that you can use for token retrieval and use. The token provider verifies that the requests that you allow are from legitimate customers. When you send requests to the AWS resources that you protect with AWS WAF, you include the token in a cookie, to validate the request. You can handle the token cookie manually or have the token provider do it for you.

This section covers the interactions between the classes, properties, and methods that are included in the mobile SDK. For the SDK specification, see AWS WAF mobile SDK specification.

Token retrieval and caching

When you create the token provider instance in your mobile app, you configure how you want it to manage tokens and token retrieval. Your main choice is how to maintain valid, unexpired tokens for use in your app's web requests:

- **Background refresh enabled** – This is the default. The token provider automatically refreshes the token in the background and caches it. With background refresh enabled, when you call `getToken()`, the operation retrieves the cached token.

  The token provider performs the token refresh at configurable intervals, so that an unexpired token is always available in the cache while the application is active. Background refresh is

paused while your application is in an inactive state. For information about this, see Retrieving a token following app inactivity.

- **Background refresh disabled** – You can disable background token refresh, and then retrieve tokens only on demand. Tokens retrieved on demand aren't cached, and you can retrieve more than one if you want. Each token is independent of any others that you retrieve, and each has its own timestamp that's used to calculate expiration.

  You have the following choices for token retrieval when background refresh is disabled:

  - **`getToken()`** – When you call `getToken()` with background refresh disabled, the call synchronously retrieves a new token from AWS WAF. This is a potentially blocking call that might affect app responsiveness if you invoke it on the main thread.
  - **`onTokenReady(WAFTokenResultCallback)`** – This call asynchronously retrieves a new token and then invokes the provided result callback in a background thread when a token is ready.

How the token provider retries failed token retrievals

The token provider automatically retries token retrieval when retrieval fails. Retries are initially performed using exponential backoff with a starting retry wait time of 100 ms. For information about exponential retries, see Error retries and exponential backoff in AWS.

When the number of retries reaches the configured `maxRetryCount`, the token provider either stops trying or switches to trying every `maxErrorTokenRefreshDelayMsec` milliseconds, depending on the type of token retrieval:

- **`onTokenReady()`** – The token provider switches to waiting `maxErrorTokenRefreshDelayMsec` milliseconds between attempts, and continues trying to retrieve the token.
- **Background refresh** – The token provider switches to waiting `maxErrorTokenRefreshDelayMsec` milliseconds between attempts, and continues trying to retrieve the token.
- **On-demand `getToken()` calls, when background refresh is disabled** – The token provider stops trying to retrieve a token and returns the previous token value, or a null value if there is no previous token.

Token retrieval retry scenarios

When the token provider tries to retrieve a token, it might result in auto-retries depending on where token retrieval fails in the token acquisition flow. This section lists the possible places where you might see an auto-retry.

- **Obtaining or verifying the AWS WAF Challenge through /inputs or /verify:**
  - When a request to obtain and verify a AWS WAF challenge is made and fails, it can result in an auto-retry.
  - You might observe auto-retries happening here along with a `socketTimeoutException` error. This can have multiple causes including:
    - Low network bandwidth: Confirm your network connectivity settings
    - Mutated Application Integration URL: Confirm that the integration URL is not modified from what appears on the AWS WAF console
  - The auto-retry count is configurable with the `maxRetryCount()` function
- **Refreshing the token:**
  - When a request to refresh the token is made through the token handler, it might result in an auto-retry.
  - The auto-retry count here is configurable with the `maxRetryCount()` function.

A configuration with no auto-retries is possible by setting `maxRetryCount(0)`.

Token immunity time and background refresh

The token immunity time that you configure in the Web ACL is independent of the token refresh interval you set in the AWS WAF mobile SDK. When you enable background refresh, the SDK refreshes the token at the interval you specify using `tokenRefreshDelaySec()`. This can result in multiple valid tokens existing simultaneously, depending on your configured immunity time.

To prevent multiple valid tokens, you can disable background refresh and use the `getToken()` function to manage the token lifecycle in your mobile app.

Retrieving a token following app inactivity

Background refresh is only performed while your app is considered active for your app type:

- **iOS** – Background refresh is performed when the app is in the foreground.

- **Android** – Background refresh is performed when the app isn't closed, whether it's in the foreground or background.

If your app remains in any state that doesn’t support background refresh for longer than your configured `tokenRefreshDelaySec` seconds, the token provider pauses background refresh. For example, for an iOS app, if `tokenRefreshDelaySec` is 300 and the app closes or goes into the background for more than 300 seconds, the token provider stops refreshing the token. When the app returns to an active state, the token provider automatically restarts background refresh.

When your app comes back to an active state, call `onTokenReady()` so you can be notified when the token provider has retrieved and cached a new token. Don't just call `getToken()`, because the cache might not yet contain a current, valid token.

Application Integration URL

The AWS WAF mobile SDK application integration URL points to a Web ACL that you've enabled for application integration. This URL routes requests to the correct backend server and associates them with your customer. It doesn't serve as a hard security control, so exposing an integration URL doesn't pose a security risk.

You can technically modify the provided integration URL and still obtain a token. However, we don't recommend this because you might lose visibility into challenge solve rates or encounter token retrieval failures with `socketTimeoutException` errors.

Dependencies

Each downloadable AWS WAF mobile SDK includes a README file that lists out the dependencies for its specific version of the SDK. Refer to the README for the dependencies for your version of the mobile SDK.

Obfuscation/ProGuard (Android SDK Only)

If you use an obfuscation or minification product like ProGuard, you might need to exclude certain namespaces to ensure the mobile SDK works correctly. Check the README for your version of the mobile SDK to find the list of namespaces and exclusion rules.

Code examples for the AWS WAF mobile SDK

This section provides code examples for using the mobile SDK.

Initializing the token provider and getting tokens

You initiate your token provider instance using a configuration object. Then you can retrieve tokens using the available operations. The following shows the basic components of the required code.

iOS

```
let url: URL = URL(string: "protection pack (web ACL) integration URL")!
let configuration = WAFConfiguration(applicationIntegrationUrl: url, domainName:
 "Domain name")
let tokenProvider = WAFTokenProvider(configuration)

//onTokenReady can be add as an observer for
 UIApplication.willEnterForegroundNotification
self.tokenProvider.onTokenReady() { token, error in
 if let token = token {
 //token available
 }

 if let error = error {
 //error occurred after exhausting all retries
 }
}

//getToken()
let token = tokenProvider.getToken()
```

Android

Java example:

```
String applicationIntegrationURL = "protection pack (web ACL) integration URL";
//Or
URL applicationIntegrationURL = new URL("protection pack (web ACL) integration
 URL");

String domainName = "Domain name";

WAFConfiguration configuration =
 WAFConfiguration.builder().applicationIntegrationURL(applicationIntegrationURL).domainName(
WAFTokenProvider tokenProvider = new WAFTokenProvider(Application context,
 configuration);
```

```
// implement a token result callback
WAFTokenResultCallback callback = (wafToken, error) -> {
 if (wafToken != null) {
 // token available
 } else {
 // error occurred in token refresh
 }
};

// Add this callback to application creation or activity creation where token will
 be used
tokenProvider.onTokenReady(callback);

// Once you have token in token result callback
// if background refresh is enabled you can call getToken() from same tokenprovider
 object
// if background refresh is disabled you can directly call getToken()(blocking call)
 for new token
WAFToken token = tokenProvider.getToken();
```

Kotlin example:

```
import com.amazonaws.waf.mobilesdk.token.WAFConfiguration
import com.amazonaws.waf.mobilesdk.token.WAFTokenProvider

private lateinit var wafConfiguration: WAFConfiguration
private lateinit var wafTokenProvider: WAFTokenProvider

private val WAF_INTEGRATION_URL = "protection pack (web ACL) integration URL"
private val WAF_DOMAIN_NAME = "Domain name"

fun initWaf() {
 // Initialize the tokenprovider instance
 val applicationIntegrationURL = URL(WAF_INTEGRATION_URL)
 wafConfiguration =
  WAFConfiguration.builder().applicationIntegrationURL(applicationIntegrationURL)
    .domainName(WAF_DOMAIN_NAME).backgroundRefreshEnabled(true).build()
 wafTokenProvider = WAFTokenProvider(getApplication(), wafConfiguration)

  // getToken from tokenprovider object
  println("WAF: "+ wafTokenProvider.token.value)

  // implement callback for where token will be used
```

```
  wafTokenProvider.onTokenReady {
   wafToken, sdkError ->
  run {
   println("WAF Token:" + wafToken.value)
  }
 }
}
```

Allowing the SDK to provide the token cookie in your HTTP requests

If `setTokenCookie` is TRUE, the token provider includes the token cookie for you in your web requests to all locations under the path that's specified in `tokenCookiePath`. By default,`setTokenCookie` is TRUE and `tokenCookiePath` is `/`.

You can narrow the scope of the requests that include a token cookie by specifying the token cookie path, for example, `/web/login`. If you do this, check that your AWS WAF rules don't inspect for tokens in the requests that you send to other paths. When you use the `AWSManagedRulesACFPRuleSet` rule group, you configure the account registration and creation paths, and the rule group checks for tokens in requests that are sent to those paths. For more information, see Adding the ACFP managed rule group to your web ACL. Similarly, when you use the `AWSManagedRulesATPRuleSet` rule group, you configure the login path, and the rule group checks for tokens in requests that are sent to that path. For more information, see Adding the ATP managed rule group to your protection pack (web ACL).

iOS

When `setTokenCookie` is TRUE, the token provider stores the AWS WAF token in a `HTTPCookieStorage.shared` and automatically includes the cookie in requests to the domain that you specified in `WAFConfiguration`.

```
let request = URLRequest(url: URL(string: domainEndpointUrl)!)
//The token cookie is set automatically as cookie header
let task = URLSession.shared.dataTask(with: request) { data, urlResponse, error  in
}.resume()
```

Android

When `setTokenCookie` is TRUE, the token provider stores the AWS WAF token in a `CookieHandler` instance that's shared application wide. The token provider automatically includes the cookie in requests to the domain that you specified in `WAFConfiguration`.

Java example:

```
URL url = new URL("Domain name");
//The token cookie is set automatically as cookie header
HttpsURLConnection connection = (HttpsURLConnection) url.openConnection();
connection.getResponseCode();
```

Kotlin example:

```
val url = URL("Domain name")
//The token cookie is set automatically as cookie header
val connection = (url.openConnection() as HttpsURLConnection)
connection.responseCode
```

If you already have the `CookieHandler` default instance initialized, the token provider will use it to manage cookies. If not, the token provider will initialize a new `CookieManager` instance with the AWS WAF token and `CookiePolicy.ACCEPT_ORIGINAL_SERVER` and then set this new instance as the default instance in `CookieHandler`.

The following code shows how the SDK initializes the cookie manager and cookie handler when they aren't available in your app.

Java example:

```
CookieManager cookieManager = (CookieManager) CookieHandler.getDefault();
if (cookieManager == null) {
 // Cookie manager is initialized with CookiePolicy.ACCEPT_ORIGINAL_SERVER
 cookieManager = new CookieManager();
 CookieHandler.setDefault(cookieManager);
}
```

Kotlin example:

```
var cookieManager = CookieHandler.getDefault() as? CookieManager
if (cookieManager == null) {
 // Cookie manager is initialized with CookiePolicy.ACCEPT_ORIGINAL_SERVER
 cookieManager = CookieManager()
 CookieHandler.setDefault(cookieManager)
}
```

Manually providing the token cookie in your HTTP requests

If you set `setTokenCookie` to FALSE, then you need to provide the token cookie manually, as a Cookie HTTP request header, in your requests to your protected endpoint. The following code shows how to do this.

iOS

```
var request = URLRequest(url: wafProtectedEndpoint)
request.setValue("aws-waf-token=token from token provider", forHTTPHeaderField:
 "Cookie")
request.httpShouldHandleCookies = true
URLSession.shared.dataTask(with: request) { data, response, error in }
```

Android

Java example:

```
URL url = new URL("Domain name");
HttpsURLConnection connection = (HttpsURLConnection) url.openConnection();
String wafTokenCookie = "aws-waf-token=token from token provider";
connection.setRequestProperty("Cookie", wafTokenCookie);
connection.getInputStream();
```

Kotlin example:

```
val url = URL("Domain name")
val connection = (url.openConnection() as HttpsURLConnection)
val wafTokenCookie = "aws-waf-token=token from token provider"
connection.setRequestProperty("Cookie", wafTokenCookie)
connection.inputStream
```

CAPTCHA and Challenge in AWS WAF

This section explains how CAPTCHA and Challenge work with AWS WAF.

You can configure your AWS WAF rules to run a CAPTCHA or Challenge action against web requests that match your rule's inspection criteria. You can also program your JavaScript client applications to run CAPTCHA puzzles and browser challenges locally.

CAPTCHA puzzles and silent challenges can only run when browsers are accessing HTTPS endpoints. Browser clients must be running in secure contexts in order to acquire tokens.

- **CAPTCHA** – Requires the end user to solve a CAPTCHA puzzle to prove that a human being is sending the request. CAPTCHA puzzles are intended to be fairly easy and quick for humans to complete successfully and hard for computers to either complete successfully or to randomly complete with any meaningful rate of success.

  In protection pack (web ACL) rules, CAPTCHA is commonly used when a Block action would stop too many legitimate requests, but letting all traffic through would result in unacceptably high levels of unwanted requests, such as from bots. For information about the rule action behavior, see How the AWS WAF CAPTCHA and Challenge rule actions work.

  You can also program a CAPTCHA puzzle implementation in your client application integration APIs. When you do this, you can customize the behavior and placement of the puzzle in your client application. For more information, see Client application integrations in AWS WAF.
- **Challenge** – Runs a silent challenge that requires the client session to verify that it's a browser, and not a bot. The verification runs in the background without involving the end user. This is a good option for verifying clients that you suspect of being invalid without negatively impacting the end user experience with a CAPTCHA puzzle. For information about the rule action behavior, see How the AWS WAF CAPTCHA and Challenge rule actions work.

  The Challenge rule action is similar to the challenge run by the client intelligent threat integration APIs, described at Client application integrations in AWS WAF.

> **Note**
>
> You are charged additional fees when you use the CAPTCHA or Challenge rule action in one of your rules or as a rule action override in a rule group. For more information, see AWS WAF Pricing.

For descriptions of all of the rule action options, see Using rule actions in AWS WAF.

Topics

- AWS WAF CAPTCHA puzzles
- How the AWS WAF CAPTCHA and Challenge rule actions work

- Best practices for using the CAPTCHA and Challenge actions

AWS WAF CAPTCHA puzzles

This section explains the features and functionality of the AWS WAF CAPTCHA puzzle.

AWS WAF provides standard CAPTCHA functionality that challenges users to confirm that they are human beings. CAPTCHA stands for Completely Automated Public Turing test to tell Computers and Humans Apart. CAPTCHA puzzles are designed to verify that a human is sending requests and to prevent activity like web scraping, credential stuffing, and spam. CAPTCHA puzzles can't weed out all unwanted requests. Many puzzles have been solved using machine learning and artificial intelligence. In an effort to circumvent CAPTCHA, some organizations supplement automated techniques with human intervention. In spite of this, CAPTCHA continues to be a useful tool to prevent less sophisticated bot traffic and to increase the resources required for large-scale operations.

AWS WAF randomly generates its CAPTCHA puzzles and rotates through them to ensure that users are presented with unique challenges. AWS WAF regularly adds new types and styles of puzzles to remain effective against automation techniques. In addition to the puzzles, the AWS WAF CAPTCHA script gathers data about the client to ensure that the task is being completed by a human and to prevent replay attacks.

Each CAPTCHA puzzle includes a standard set of controls for the end user to request a new puzzle, switch between audio and visual puzzles, access additional instructions, and submit a puzzle solution. All puzzles include support for screen readers, keyboard controls, and contrasting colors.

The AWS WAF CAPTCHA puzzles meet the requirements of the Web Content Accessibility Guidelines (WCAG). For information, see Web Content Accessibility Guidelines (WCAG) Overview at the World Wide Web Consortium (W3C) website.

Topics

- CAPTCHA puzzle language support
- CAPTCHA puzzle examples

CAPTCHA puzzle language support

This section lists what languages are supported in AWS WAF CAPTCHA puzzles.

The CAPTCHA puzzle starts with written instructions in the client browser language or, if the browser language is unsupported, in English. The puzzle provides alternate language options through a dropdown menu.

The user can switch to audio instructions by selecting the headphone icon at the bottom of the page. The audio version of the puzzle provides spoken instructions about text that the user should type into a text box, overlaid by background noise.

The following table lists the languages that you can select for the written instructions in a CAPTCHA puzzle and the audio support for each selection.

AWS WAF CAPTCHA puzzle supported languages

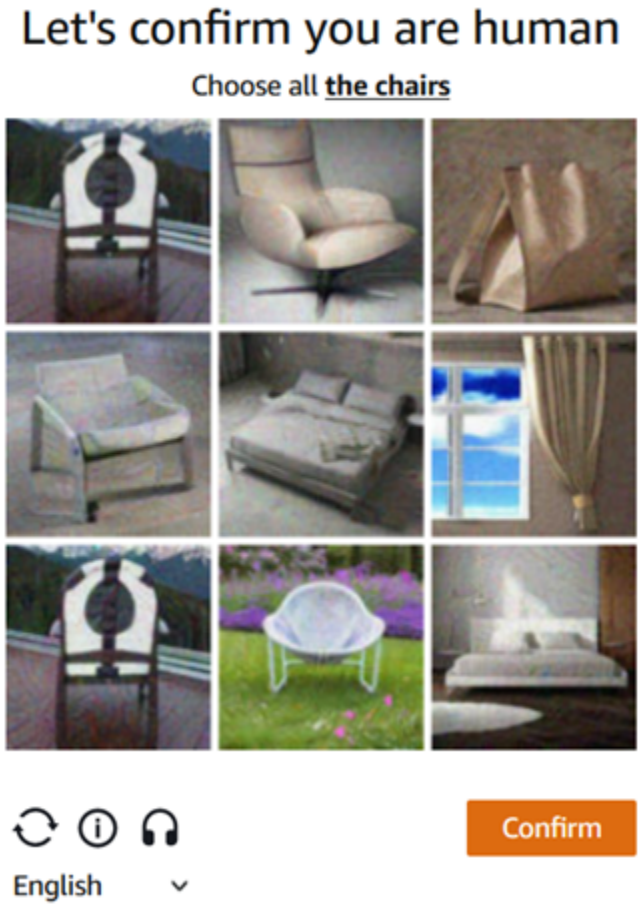
Written instructions support	Locale code	Audio instructions support
Arabic	ar-SA	Arabic
Simplified Chinese	zh-CN	Audio in English
Dutch	nl-NL	Dutch
English	en-US	English
French	fr-FR	French
German	de-DE	German
Italian	it-IT	Italian
Japanese	ja-JP	Audio in English
Brazilian Portuguese	pt-BR	Brazilian Portuguese
Spanish	es-ES	Spanish

Written instructions support	Locale code	Audio instructions support
Turkish	tr-TR	Turkish

CAPTCHA puzzle examples

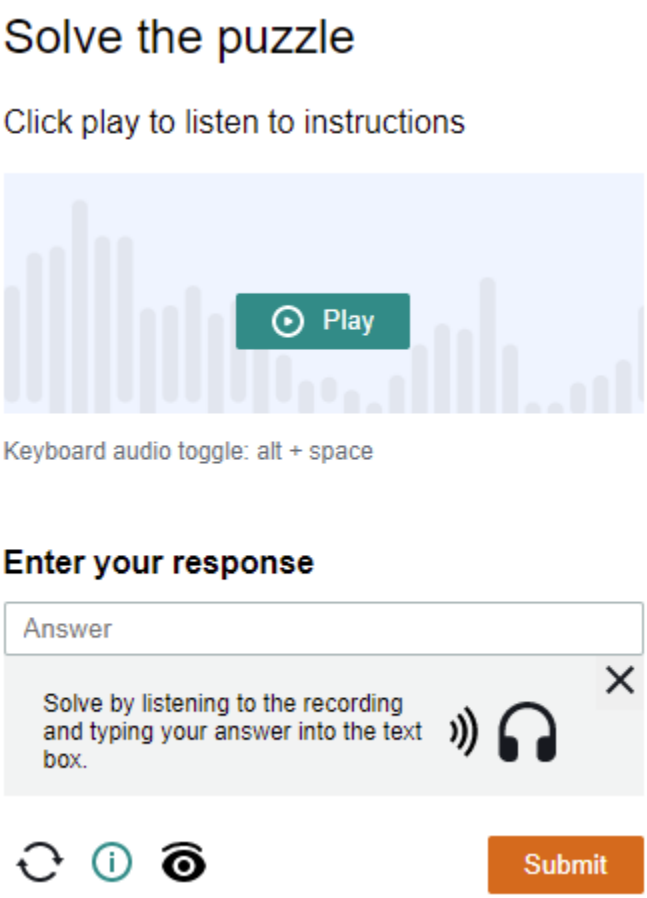
A typical visual CAPTCHA puzzle requires interaction to show that the user can comprehend and interact with one or more images.

The following screenshot shows an example of a picture grid puzzle. This puzzle requires you to select all of the pictures in the grid that include a specific type of object.



An audio puzzle provides background noise overlaid with spoken instructions about text that the user should type into a text box.

The following screenshot shows the display for the audio puzzle choice.



How the AWS WAF CAPTCHA and Challenge rule actions work

This section explains how CAPTCHA and Challenge work.

AWS WAF CAPTCHA and Challenge are standard rule actions, so they're relatively easy to implement. To use either of them, you create the inspection criteria for your rule that identifies the requests that you want to inspect, and then specify one of the two rule actions. For general information about rule action options, see Using rule actions in AWS WAF.

In addition to implementing silent challenges and CAPTCHA puzzles from the server side, you can integrate silent challenges in your JavaScript and iOS and Android client applications, and you can render CAPTCHA puzzles in your JavaScript clients. These integrations allow you to provide your end users with better performance and CAPTCHA puzzle experiences, and they can reduce costs associated with using the rule actions and the intelligent threat mitigation rule groups. For more information about these options, see Client application integrations in AWS WAF. For pricing information, see AWS WAF Pricing.

Topics

- CAPTCHA and Challenge action behavior
- CAPTCHA and Challenge actions in the logs and metrics

CAPTCHA and Challenge action behavior

This section explains what the CAPTCHA and Challenge actions do.

When a web request matches the inspection criteria of a rule with CAPTCHA or Challenge action, AWS WAF determines how to handle the request according to the state of its token and immunity time configuration. AWS WAF also considers whether the request can handle the CAPTCHA puzzle or challenge script interstitials. The scripts are designed to be handled as HTML content, and they can only be handled properly by a client that's expecting HTML content.

> **Note**
>
> You are charged additional fees when you use the CAPTCHA or Challenge rule action in one of your rules or as a rule action override in a rule group. For more information, see AWS WAF Pricing.

How the action handles the web request

AWS WAF applies the CAPTCHA or Challenge action to a web request as follows:

- **Valid token** – AWS WAF handles this similar to a Count action. AWS WAF applies any labels and request customizations that you've configured for the rule action, and then continues evaluating the request using the remaining rules in the protection pack (web ACL).
- **Missing, invalid, or expired token** – AWS WAF discontinues the protection pack (web ACL) evaluation of the request and blocks it from going to its intended destination.

  AWS WAF generates a response that it sends back to the client, according to the rule action type:

  - **Challenge** – AWS WAF includes the following in the response:
    - The header `x-amzn-waf-action` with a value of `challenge`.

> **Note**
>
> For Javascript applications running in the client browser, this header is only available within the application's domain. The header isn't available for cross-domain retrieval. For details, see the section that follows.

- The HTTP status code `202 Request Accepted`.
- If the request contains an `Accept` header with a value of `text/html`, the response includes a JavaScript page interstitial with a challenge script.

- **CAPTCHA** – AWS WAF includes the following in the response:
  - The header `x-amzn-waf-action` with a value of `captcha`.

> **Note**
>
> For Javascript applications running in the client browser, this header is only available within the application's domain. The header isn't available for cross-domain retrieval. For details, see the section that follows.

  - The HTTP status code `405 Method Not Allowed`.
  - If the request contains an `Accept` header with a value of `text/html`, the response includes a JavaScript page interstitial with a CAPTCHA script.

To configure the timing of token expiration at the protection pack (web ACL) or rule level, see Setting timestamp expiration and token immunity times in AWS WAF.

Headers are unavailable to JavaScript applications that run in the client browser

When AWS WAF responds to a client request with a CAPTCHA or challenge response, it doesn't include cross-origin resource sharing (CORS) headers. CORS headers are a set of access control headers that tell the client web browser which domains, HTTP methods, and HTTP headers can be used by JavaScript applications. Without CORS headers, JavaScript applications running in a client browser are not granted access to HTTP headers and so are unable to read the `x-amzn-waf-action` header that's provided in the CAPTCHA and Challenge responses.

What the challenge and CAPTCHA interstitials do

When a challenge interstitial runs, after the client responds successfully, if it doesn't already have a token, the interstitial initializes one for it. Then it updates the token with the challenge solve timestamp.

When a CAPTCHA interstitial runs, if the client doesn't have a token yet, the CAPTCHA interstitial invokes the challenge script first to challenge the browser and initialize the token. Then the interstitial runs its CAPTCHA puzzle. When the end user successfully completes the puzzle, the interstitial updates the token with the CAPTCHA solve timestamp.

In either case, after the client responds successfully and the script updates the token, the script resubmits the original web request using the updated token.

You can configure how AWS WAF handles tokens. For information, see Token use in AWS WAF intelligent threat mitigation.

CAPTCHA and Challenge actions in the logs and metrics

This section explains how AWS WAF handles logging and metrics for the CAPTCHA and Challenge actions.

The CAPTCHA and Challenge actions can be non-terminating, like Count, or terminating, like Block. The outcome depends on whether the request has a valid token with an unexpired timestamp for the action type.

- **Valid token** – When the action finds a valid token and doesn't block the request, AWS WAF captures metrics and logs as follows:
  - Increments the metrics for either `CaptchaRequests` and `RequestsWithValidCaptchaToken` or `ChallengeRequests` and `RequestsWithValidChallengeToken`.
  - Logs the match as a `nonTerminatingMatchingRules` entry with action of CAPTCHA or Challenge. The following listing shows the section of a log for this type of match with the CAPTCHA action.

```
"nonTerminatingMatchingRules": [
{
  "ruleId": "captcha-rule",
  "action": "CAPTCHA",
  "ruleMatchDetails": [],
  "captchaResponse": {
    "responseCode": 0,
```

```
        "solveTimestamp": 1632420429
      }
    }
  ]
```

- **Missing, invalid, or expired token** – When the action blocks the request due to a missing or invalid token, AWS WAF captures metrics and logs as follows:
  - Increments the metric for `CaptchaRequests` or `ChallengeRequests`.
  - Logs the match as a `CaptchaResponse` entry with HTTP `405` status code or as a `ChallengeResponse` entry with HTTP `202` status code. The log indicates whether the request was missing the token or had an expired timestamp. The log also indicates whether AWS WAF sent a CAPTCHA interstitial page to the client or a silent challenge to the client browser. The following listing shows the sections of a log for this type of match with the CAPTCHA action.

```
"terminatingRuleId": "captcha-rule",
"terminatingRuleType": "REGULAR",
"action": "CAPTCHA",
"terminatingRuleMatchDetails": [],
...
"responseCodeSent": 405,
...
"captchaResponse": {
  "responseCode": 405,
  "solveTimestamp": 0,
  "failureReason": "TOKEN_MISSING"
}
```

For information about the AWS WAF logs, see Logging AWS WAF protection pack (web ACL) traffic.

For information about AWS WAF metrics, see AWS WAF metrics and dimensions.

For general information about rule action options, see Using rule actions in AWS WAF.

Requests with no token seem to show up twice in logs and metrics

Based on the CAPTCHA and Challenge action behavior and the logging and metrics described in this section, a request with no token will generally be represented twice in the logs and metrics. This is because the one intended request is actually sent twice by the client.

- The first request, with no token, receives the logging and metrics handling described above for missing, invalid, or expired token. The CAPTCHA or Challenge action terminates this first request, and then responds back to the client with either a silent challenge or CAPTCHA puzzle.
- The client evaluates the challenge or puzzle and, if the client browser or end user responds successfully, sends the request a second time with the newly acquired token. This second request receives the logging and metrics handling described above for a request with a valid token.

Best practices for using the CAPTCHA and Challenge actions

Follow the guidance in this section to plan and implement AWS WAF CAPTCHA or challenge.

Plan your CAPTCHA and challenge implementation

Determine where to place CAPTCHA puzzles or silent challenges based on your website usage, the sensitivity of the data that you want to protect, and the type of requests. Select the requests where you'll apply CAPTCHA so that you present the puzzles as needed, but avoid presenting them where they wouldn't be useful and might degrade the user experience. Use the Challenge action to run silent challenges that have less impact on the end user, but still help verify that the request is coming from a JavaScript enabled browser.

CAPTCHA puzzles and silent challenges can only run when browsers are accessing HTTPS endpoints. Browser clients must be running in secure contexts in order to acquire tokens.

Decide where to run CAPTCHA puzzles and silent challenges on your clients

Identify requests that you don't want to have impacted by CAPTCHA, for example, requests for CSS or images. Use CAPTCHA only when necessary. For example, if you plan to have a CAPTCHA check at login, and the user is always taken directly from the login to another screen, requiring a CAPTCHA check at the second screen would probably not be needed and might degrade your end user experience.

Configure your Challenge and CAPTCHA use so that AWS WAF only sends CAPTCHA puzzles and silent challenges in response to `GET` `text/html` requests. You can't run either the puzzle or the challenge in response to `POST` requests, Cross-Origin Resource Sharing (CORS) preflight `OPTIONS` requests, or any other non-`GET` request types. Browser behavior for other request types can vary and might not be able to handle the interstitials properly.

It's possible for a client to accept HTML but still not be able to handle the CAPTCHA or challenge interstitial. For example, a widget on a webpage with a small iFrame might accept HTML but not be

able to display a CAPTCHA or process it. Avoid placing the rule actions for these types of requests, the same as for requests that don't accept HTML.

Use CAPTCHA or Challenge to verify prior token acquisition

You can use the rule actions solely to verify the existence of a valid token, at locations where legitimate users should always have one. In these situations, it doesn't matter whether the request can handle the interstitials.

For example, if you implement the JavaScript client application CAPTCHA API, and run the CAPTCHA puzzle on the client immediately before you send the first request to your protected endpoint, your first request should always include a token that's valid for both challenge and CAPTCHA. For information about JavaScript client application integration, see AWS WAF JavaScript integrations.

For this situation, in your protection pack (web ACL), you can add a rule that matches against this first call and configure it with the Challenge or CAPTCHA rule action. When the rule matches for a legitimate end user and browser, the action will find a valid token, and therefore will not block the request or send a challenge or CAPTCHA puzzle in response. For more information about how the rule actions work, see CAPTCHA and Challenge action behavior.

Protect your sensitive non-HTML data with CAPTCHA and Challenge

You can use CAPTCHA and Challenge protections for sensitive non-HTML data, like APIs, with the following approach.

1. Identify requests that take HTML responses and that are run in close proximity to the requests for your sensitive, non-HTML data.
2. Write CAPTCHA or Challenge rules that match against the requests for HTML and that match against the requests for your sensitive data.
3. Tune your CAPTCHA and Challenge immunity time settings so that, for normal user interactions, the tokens that clients obtain from the HTML requests are available and unexpired in their requests for your sensitive data. For tuning information, see Setting timestamp expiration and token immunity times in AWS WAF.

When a request for your sensitive data matches a CAPTCHA or Challenge rule, it won't be blocked if the client still has a valid token from the prior puzzle or challenge. If the token isn't available or the timestamp is expired, the request to access your sensitive data will fail. For more information about how the rule actions work, see CAPTCHA and Challenge action behavior.

Use CAPTCHA and Challenge to tune your existing rules

Review your existing rules, to see if you want to alter or add to them. The following are some common scenarios to consider.

- If you have a rate-based rule that blocks traffic, but you keep the rate limit relatively high to avoid blocking legitimate users, consider adding a second rate-based rule after the blocking rule. Give the second rule a lower limit than the blocking rule and set the rule action to CAPTCHA or Challenge. The blocking rule will still block requests that are coming at too high a rate, and the new rule will block most automated traffic at an even lower rate. For information about rate-based rules, see Using rate-based rule statements in AWS WAF.
- If you have a managed rule group that blocks requests, you can switch the behavior for some or all of the rules from Block to CAPTCHA or Challenge. To do this, in the managed rule group configuration, override the rule action setting. For information about overriding rule actions, see Rule group rule action overrides.

Test your CAPTCHA and challenge implementations before you deploy them

As for all new functionality, follow the guidance at the section called “Testing and tuning your protections”.

During testing, review your token timestamp expiration requirements and set your web ACL and rule level immunity time configurations so that you achieve a good balance between controlling access to your website and providing a good experience for your customers. For information, see Setting timestamp expiration and token immunity times in AWS WAF.

AI traffic monetization

AI traffic monetization enables content and API providers to charge AI bots and agents for access to protected resources directly at the edge. When a client requests a monetized resource, AWS WAF returns an HTTP 402 Payment Required response containing pricing and accepted payment networks. The client submits a signed payment authorization on their payment network of choice, AWS WAF verifies it, fetches the content, integrates with third-party facilitator services for settling the payment on-chain, and serves the response.

AI traffic monetization uses an open payment protocol for machine-to-machine payments. Any compatible client or agent runtime can complete payments.

> **Note**
>
> AI traffic monetization is available globally with Amazon CloudFront at no additional charge beyond standard AWS WAF pricing.

Topics

- How AI traffic monetization works
- Getting started with AI traffic monetization
- Pricing configuration
- Payment networks and settlement
- Revenue analytics
- Communicating license terms to AI agents

How AI traffic monetization works

AI traffic monetization uses the x402 open protocol for machine-to-machine payments. The following describes the request lifecycle for a monetized resource:

1. **Request** – A client (typically an AI agent) sends a request to a AWS WAF protected resource on your CloudFront distribution.
2. **Rule evaluation** – AWS WAF evaluates the request against your rules in priority order. If a rule with a Monetize action matches and the request does not include a valid payment authorization, AWS WAF returns an HTTP 402 Payment Required response. For more details, see Rule actions.
3. **Payment Required Challenge** – AWS WAF returns an HTTP 402 response (the "Payment Required Challenge"). The response includes payment instructions containing:
   a. Content price (per request) in USDC
   b. Accepted payment networks (Base, Solana)
   c. Publisher wallet address (payTo)
   d. Maximum timeout
   e. Payment scheme
4. **Payment authorization** – The client signs a payment authorization using their wallet's private key or a server wallet API. The client resubmits the original request with a `payment-signature` header containing the signed authorization.

5. **Verification** – AWS WAF verifies the payment credentials, confirming transfer of sufficient funds and valid authorization. This occurs synchronously in the request path. If the verification fails, the client is served a 402 and the content is not served.
6. **Content fetch** – On successful verification, the request for content is allowed.
7. **Settlement** – If content fetch is successful (2xx status code), the payment is settled on the blockchain via Coinbase Developer Platform's x402 facilitator service. Settlement occurs synchronously – content is served after confirmed payment. If the payment settlement fails, the client is served a 402 and the content is not served.
8. **Response** – The content is served to the client with a `payment-response` header containing settlement confirmation details.

Key behaviors:

- **No payment for failed origins** – If origin returns 4xx or 5xx, settlement is skipped and the client is not charged.
- **Idempotency** – The x402 protocol supports a payment-identifier extension that allows clients to retry requests without double-payment for up to 15 minutes, as long as the extension is used by the client.
- **Replay protection** – Payment authorizations are single-use. Reusing a payment header without a valid payment-identifier results in a new 402 response.

For more details about the x402 open payment protocol, see x402 documentation.

Supported resource types

AI traffic monetization protects resources on Amazon CloudFront distributions. You can monetize any path or content zone served through CloudFront, including:

- Web pages and articles
- API endpoints
- Data feeds
- Media assets
- Structured datasets

Getting started with AI traffic monetization

Prerequisites

- An AWS WAF protection pack (web ACL) associated with an Amazon CloudFront distribution
- A USDC wallet address on Base, Solana, or both to receive payments.
- (Recommended) AWS WAF Bot Control enabled to classify AI bot traffic by identity

Quick start

1. **Enable Bot Control** – If not already enabled, add the AWS WAF Bot Control managed rule group to your web ACL.
2. **Create a protection pack (webACL)** – In the AWS WAF console, navigate to Protection packs (webACL) in the Cloudfront (Global) region, and create a new pack with your pricing, bot and agent policies, and payment configuration. Associate with a CloudFront distribution.
3. **Configure MonetizationConfig** – To use AI traffic monetization, you configure a `MonetizationConfig` on your web ACL. This configuration defines the payment networks you accept and the base price for monetized requests.

The MonetizationConfig contains:

- **CurrencyMode** – REAL for production payments or TEST for test network payments.
- **CryptoConfig** – Payment network configuration including chains, wallet addresses, and pricing.

MonetizationConfig structure

```
{
  "MonetizationConfig": {
    "CryptoConfig": {
      "PaymentNetworks": [
        {
          "Chain": "BASE",
          "WalletAddress": "0x1234567890abcdef1234567890abcdef12345678",
          "Prices": [
            {
              "Amount": "0.001",
              "Currency": "USDC"
```

```
            }
          ]
        },
        {
          "Chain": "SOLANA",
          "WalletAddress": "7xKXtg2CW87d97TXJSDpbD5jBkheTqA83TZRuJosgAsU",
          "Prices": [
            {
              "Amount": "0.001",
              "Currency": "USDC"
            }
          ]
        }
      ]
    }
  }
}
```

Important

The `Amount` is specified in USDC as a decimal string with up to 3 decimal places (for example, `"0.001"` = $0.001 USDC per request).

Supported payment networks

Chain	Network	Currency	Use Case
BASE	Base mainnet (EVM L2)	USDC	Production Payments
SOLANA	Solana mainnet	USDC	Production Payments
BASE_SEPOLIA	Base Sepolia testnet	USDC	Test Mode
SOLANA_DEVNET	Solana Devnet	USDC	Test Mode

Test mode

Test mode (`CurrencyMode: TEST`) enables you to validate your monetization configuration using test blockchain networks before going live with real payments. This configuration is applied to the entire protection pack (web ACL) (web ACL).

> **Important**
>
> To implement your AI traffic monetization policies, we use multiple detection techniques such as behavioral signals and risk-based systems to inspect and categorize inbound traffic. While these methods are designed to provide high-confidence classification, they are probabilistic and might not correctly identify or categorize all bot traffic in all cases. We continuously test and update our analysis methods to increase accuracy. We recommend using Test mode to validate that your policies produce the expected results before enabling live monetization.

Enabling test mode

Set `CurrencyMode` to `TEST` in your MonetizationConfig and configure test network chains.

Test mode behavior

In test mode:

- Payments use test networks: **Base Sepolia** and **Solana Devnet**
- You use test funds such as Testnet Faucet (available from public faucets) instead of real USDC
- The full payment flow executes identically to production – verification, origin fetch, and settlement all occur on the test blockchain
- All events are logged with `CurrencyMode: TEST`
- Revenue analytics are queryable with the `CurrencyMode: ["TEST"]` filter

> **Important**
>
> Use test mode only on non-production traffic for internal validation before enforcing monetization policies in production.

Validating configuration

Use test mode to verify:

- Correct pricing for different rules and content paths
- Payment flow end-to-end (402 → payment authorization → verification → settlement)
- Wallet address correctness (funds arrive in your test wallet)
- Analytics and log fields are populated correctly

When satisfied, update your MonetizationConfig to `CurrencyMode: REAL` and configure production chains (BASE, SOLANA) to begin processing real payments.

- You can start with `CurrencyMode: TEST` for initial testing, or directly start monetizing using `CurrencyMode: REAL`.
- **Add Monetize rules** – Create rules with the Monetize action. Use match statements to target specific paths, Bot Control labels, or other request characteristics.
- **Test** – Send test requests to your CloudFront distribution. Requests matching your Monetize rules will receive Payment Required Challenges (402 responses). Use a compatible payment client to complete the flow.
- **Go live** – Update `CurrencyMode` to `REAL` and configure production payment networks (BASE, SOLANA). Fund a test wallet and verify end-to-end before opening to production traffic.

Next steps

- Review the AI traffic analysis dashboard to understand your current bot traffic before setting prices.
- Configure differentiated pricing based on agent tier and content path.
- Set up revenue analytics to monitor earnings.

Pricing configuration

Base price

Set a base price per request in the MonetizationConfig. The price is specified as a decimal USD string with up to 3 decimal places. The minimum price per request is $0.001 USDC.

Price multipliers

Use the `PriceMultiplier` parameter on individual Monetize rule actions to adjust pricing per rule. The effective price is the base price multiplied by this value.

For example, if your base price is `"0.001"` ($0.001) and a rule has PriceMultiplier `"3"`, the effective price for requests matching that rule is $0.003.

Configuring pricing by intent

You can set different prices for each verification tier. Common patterns include:

- Lower prices for verified AI search crawlers that drive referral traffic
- Standard prices for known agents performing RAG or summarization
- Higher prices for unknown or unverified agents
- Blocking training crawlers entirely or pricing at a premium

> **Important**
>
> To implement your AI traffic monetization policies, we use multiple detection techniques such as behavioral signals and risk-based systems to inspect and categorize inbound traffic. While these methods are designed to provide high-confidence classification, they are probabilistic and might not correctly identify or categorize all bot traffic in all cases. We continuously test and update our analysis methods to increase accuracy. We recommend using Test mode to validate that your policies produce the expected results before enabling live monetization.

Configuring rule action

For each tier, you can set one of the following actions:

- **Monetize** – A terminating action. When a rule with the Monetize action matches, AWS WAF stops evaluating subsequent rules. If the request does not include a valid payment authorization, AWS WAF returns an HTTP 402 response directly and the request is blocked.
- **Allow** – Grant free access (for example, verified search crawlers under a referral agreement).
- **Block** – Deny access entirely.

- **Count** – Allow access and log the request without charging.
- **Captcha** – Requires the end user to solve a CAPTCHA puzzle to prove that a human being is sending the request.
- **Challenge** – Runs a silent challenge that requires the client session to verify that it's a browser, and not a bot.

Important: Monetize action and human traffic

The Monetize action is designed for automated AI agent traffic. When a request matches a Monetize rule, AWS WAF returns an HTTP 402 Payment Required response containing machine-readable payment instructions in the x402 protocol format. Standard web browsers and human users cannot interpret or complete this payment flow – the 402 response will effectively block access for non-automated clients.

To avoid inadvertently blocking human visitors, use the Monetize action only on traffic that has been identified as automated. We recommend combining the Monetize action with AWS WAF Bot Control labels so that only bot traffic receives the 402 challenge, while human users continue to access your content normally.

Example: Monetize only bot traffic using Bot Control labels

The following rule configuration uses Bot Control to classify traffic first, then applies the Monetize action only to requests labeled as bots:

```
{
  "Name": "MonetizeBotTrafficOnly",
  "Priority": 5,
  "Statement": {
    "LabelMatchStatement": {
      "Scope": "LABEL",
      "Key": "awswaf:managed:aws:bot-control:bot"
    }
  },
  "Action": {
    "Monetize": {
    }
  },
  "VisibilityConfig": {
    "CloudWatchMetricsEnabled": true,
    "MetricName": "MonetizeBotTrafficOnly",
```

```
    "SampledRequestsEnabled": true
  }
}
```

In this example, requests from human users (which do not carry the `awswaf:managed:aws:bot-control:bot` label) pass through this rule without matching and continue to subsequent rules. Only requests classified as bots by Bot Control receive the 402 Payment Required Challenge.

You can further refine this pattern by creating separate rules for different bot tiers – for example, allowing verified search crawlers for free while monetizing unverified bots at a higher price multiplier.

Latency characteristics

AI traffic monetization adds several seconds of additional latency to requests that require payment processing. This overhead covers payment authorization verification and on-chain settlement. The additional latency applies only to requests where a transaction is being attempted – requests that do not match the Monetize action or requests without a payment signature attached are unaffected.

The exact latency depends on blockchain network conditions at the time of settlement.

Payment processing errors and throttling

Because AI traffic monetization relies on third-party blockchain settlement services, payment processing may occasionally fail for reasons outside of our control. Possible causes include temporary unavailability of the Coinbase Developer Platform's x402 facilitator service, blockchain network congestion, or transient errors during on-chain settlement. When payment processing fails, content is not served to the client. The client receives a response indicating the failure and can retry the request.

Additionally, AWS might throttle excessively high volumes of payment traffic to protect the integrity of the payment processing infrastructure and prevent abuse. If throttled, payment requests receive an error response and should be retried after a brief backoff. Normal request volumes are not affected by throttling.

Payment networks and settlement

You choose how to receive payments by configuring wallet addresses in your MonetizationConfig. Payments settle directly to your wallet. AWS is not a party to or in the flow of funds for any payment you receive.

How settlement works

1. The client's payment authorization is verified before content is fetched from origin.
2. After origin returns a successful response, the payment is settled on the blockchain.
3. USDC is transferred from the client's wallet to your configured wallet address.
4. Settlement confirmation is included in the response to the client.

Payment settlement is provided to you by Coinbase Developer Platform's x402 facilitator. You agree to Coinbase's terms of service. You instruct us to share pricing and payment configuration information with Coinbase and the relevant client.

Revenue analytics

Revenue dashboard

The revenue dashboard shows:

- Revenue summary – Total revenue, verified vs unverified revenue breakdown, total settlements, and total 402 responses served (GetRevenueStatisticsSummary)
- Top sources by revenue – Ranked AI bots by revenue amount, grouped by name, category, intent, or organization (GetRevenueStatistics with TOP_SOURCES_BY_REVENUE)
- Top paths by revenue – Ranked content paths by revenue amount (GetRevenueStatistics with TOP_PATHS_BY_REVENUE)
- Revenue over time – Time series charts at minutely, 5-minute, hourly, or daily intervals (GetRevenueStatisticsTimeSeries with DATE_HISTOGRAM)
- Payment traffic over time – Payment attempt volume over time (GetRevenueStatisticsTimeSeries with PAYMENT_TRAFFIC)
- Settlement records – Individual transaction details including status (SETTLED, PENDING, FAILED, SERVICE_ERROR, SKIPPED_ORIGIN_ERROR), payer address, amount, and blockchain transaction ID (ListSettlementRecords)

All revenue data can be filtered by source name, category, organization, and intent. By default, only real currency (REAL) transactions are shown – use the CurrencyMode filter to view test transactions.

Log fields

When you enable AWS WAF logging, monetization events include the following additional fields:

Field	Description
`action`	WAF action taken on request
`monetizeVerifyResponse.verificationStatus`	Verification result: VERIFIED, INSUFFICIENT_FUNDS, INVALID_PAYLOAD, INVALID_PAYMENT_REQUIREMENTS, SERVICE_ERROR
`monetizeVerifyResponse.failureReason`	Reason for verification failure (empty on success)
`monetizeVerifyResponse.chainName`	Blockchain chain used
`monetizeVerifyResponse.network`	Blockchain network identifier in CAIP-2 format
`monetizeVerifyResponse.amount`	Payment amount in atomic token units
`monetizeVerifyResponse.currency`	Payment currency (for example, USDC)
`monetizeVerifyResponse.asset`	Token contract address
`monetizeVerifyResponse.payerAddress`	Paying client's wallet address
`monetizeVerifyResponse.idempotencyKey`	x402 payment idempotency key for retry correlation
`monetizeVerifyResponse.currencyMode`	Currency mode: REAL or TEST

For more information, see the section called “Logging”.

Metrics

Metrics are published in Amazon CloudWatch under namespace `AWS/WAFV2`.

Metric Name	Description
`MonetizeRequests`	Emitted when the terminating rule has action Monetize
`MonetizeRequestsFailedVerification`	Emitted when terminating rule has action Monetize AND verification status is not VERIFIED

These metrics have the following dimensions: Region, Rule, WebACL, Resource, ResourceType, Device, Country.

For more information, see the section called “AWS WAF metrics and dimensions”.

Communicating license terms to AI agents

AI traffic monetization tells agents how much to pay but not what they're allowed to do with the content. To communicate machine-readable usage terms, you can use RSL (Really Simple Licensing) – an open standard for machine-readable content licensing.

RSL defines several discovery mechanisms for AI agents to find your license terms:

- **robots.txt** – Add RSL directives to your existing robots.txt file. Agents discover terms during their standard crawl preflight.
- **HTTP Link header** – Inject a `Link` header referencing your RSL license URL into HTTP responses. Can be configured with CloudFront (see below).
- **HTML link element** – Embed a `<link rel="license">` tag in your HTML pages. Suitable for web content but not applicable to API or non-HTML responses.
- **RSS/Atom module** – Attach licensing metadata to feed entries.

Using CloudFront Response Header Policies for RSL

For HTTP `Link` header discovery, you can configure CloudFront to inject the RSL license reference into responses:

1. Create an RSL license XML file describing your usage terms and host it at a stable HTTPS URL (for example, `https://example.com/license.xml`). You can use a standard license from the RSL Collective or define custom terms.
2. Create a CloudFront Response Header Policy with a custom header:
   - Header name: `Link`
   - Value: `<https://example.com/license.xml>; rel="license"; type="application/rsl+xml"`
3. Associate this policy with the cache behaviors serving monetized content.

AI agents that support RSL will discover your terms automatically on every origin response.

> **Note**
>
> CloudFront Response Header Policies apply to responses served from origin. The 402 Payment Required Challenge served by the Monetize action will not include this header.

For more information on response header policies, see Adding HTTP response headers with CloudFront in the *Amazon CloudFront Developer Guide*.

For the RSL specification, see https://rslstandard.org/rsl.

Data protection and logging for AWS WAF protection pack (web ACL) traffic

This section explains the data logging, collection, and protection options that you can use with AWS WAF. The options are the following:

- **Logging** – You can configure your protection pack (web ACL) to send logs for web request traffic to a logging destination of your choice. You can configure field redaction and filtering for this choice. Logging uses the data that's available after any data protection setting are applied.

  For information about this option, see Logging AWS WAF protection pack (web ACL) traffic.
- **Request sampling** – You can configure your protection pack (web ACL) to sample the web requests that it evaluates, to get an idea of the type of traffic that your application is receiving. Request sampling uses the data that's available after any data protection settings are applied.

For information about this option, see Viewing a sample of web requests.

- **Amazon Security Lake** – You can configure Security Lake to collect protection pack (web ACL) data. Security Lake collects log and event data from various AWS sources for normalization, analysis, and management. Security Lake collects from the data that's available after any data protection settings are applied.

  For information about this option, see What is Amazon Security Lake? and Collecting data from AWS services in the *Amazon Security Lake user guide*.

  AWS WAF doesn't charge you for using this option. For pricing information, see Security Lake Pricing and How Security Lake pricing is determined in the *Amazon Security Lake user guide*.
- **Data protection** – You can configure data protections for web traffic data at two levels:
  - **Data protection for the protection pack (web ACL)** – You can configure data protection for each protection pack (web ACL), which enables you to substitute certain web traffic data with static strings or cryptographic hashing. Data protection at this level can be configured centrally, and applies across all logging and data collection options.

    For information about this option, see Data protection.
  - **Logging redaction and filtering** – For logging only, you can configure some of the web traffic data for redaction from the logs, and you can filter the data that you log. This option is in addition to any data protection setting you've configured, and it only affects the data that AWS WAF sends to the configured logging destination.

Topics

- Logging AWS WAF protection pack (web ACL) traffic
- Data protection

Logging AWS WAF protection pack (web ACL) traffic

This section explains the logging options for your AWS WAF protection packs (web ACLs).

You can enable logging to get detailed information about traffic that is analyzed by your web ACL. Logged information includes the time that AWS WAF received a web request from your AWS resource, detailed information about the request, and details about the rules that the request matched. You can send protection pack (web ACL) logs to an Amazon CloudWatch Logs log group,

an Amazon Simple Storage Service (Amazon S3) bucket, or an Amazon Data Firehose delivery stream.

In addition to logs that you can enable for your protection packs (web ACLs), AWS also uses service logs of website or application traffic processed by AWS WAF to provide support for and protect the security of AWS customers and services.

> **Note**
>
> The protection pack (web ACL) logging configuration only affects the AWS WAF logs. In particular, the redacted fields configuration for logging has no impact on request sampling or Security Lake data collection. You can exclude fields from collection or sampling by configuring protection pack (web ACL) data protection. Other than data protection, Security Lake data collection is configured entirely through the Security Lake service.

Topics

- Pricing for logging protection pack (web ACL) traffic information
- AWS WAF logging destinations
- Configuring logging for a protection pack (web ACL)
- Finding your protection pack (web ACL) records
- Log fields for protection pack (web ACL) traffic
- Log examples for protection pack (web ACL) traffic

Other data collection and analysis options

In addition to logging, you can enable the following options for data collection and analysis:

- **Amazon Security Lake** – You can configure Security Lake to collect protection pack (web ACL) data. Security Lake collects log and event data from various sources for normalization, analysis, and management. For information about this option, see What is Amazon Security Lake? and Collecting data from AWS services in the *Amazon Security Lake user guide*.

  AWS WAF doesn't charge you for using this option. For pricing information, see Security Lake Pricing and How Security Lake pricing is determined in the *Amazon Security Lake user guide*.

- **Request sampling** – You can configure your protection pack (web ACL) to sample the web requests that it evaluates, to get an idea of the type of traffic that your application is receiving. For information about this option, see Viewing a sample of web requests.

Pricing for logging protection pack (web ACL) traffic information

This section explains the pricing considerations for using protection pack (web ACL) traffic logs.

You are charged for logging protection pack (web ACL) traffic information according to the costs associated with each log destination type. These charges are in addition to the charges for using AWS WAF. Your costs can vary depending on factors such as the destination type that you choose and the amount of data that you log.

The following provides links to the pricing information for each logging destination type:

- **CloudWatch Logs** – The charges are for vended log delivery. See Amazon CloudWatch Logs Pricing. Under **Paid Tier**, choose the **Logs** tab, and then under **Vended Logs**, see the information for **Delivery to CloudWatch Logs**.
- **Amazon S3 buckets** – The Amazon S3 charges are the combined charges for CloudWatch Logs vended log delivery to the Amazon S3 buckets and for using Amazon S3.
  - For Amazon S3, see Amazon S3 Pricing.
  - For CloudWatch Logs vended log delivery to the Amazon S3, see Amazon CloudWatch Logs Pricing. Under **Paid Tier**, choose the **Logs** tab, and then under **Vended Logs**, see the information for **Delivery to S3**
- **Firehose** – See Amazon Data Firehose Pricing.

For information about AWS WAF pricing, see AWS WAF Pricing.

AWS WAF logging destinations

This section describes the logging options that you can choose from for your AWS WAF logs. Each section provides guidance for configuring logging including information about any behavior that's specific to the destination type. After you've configured the logging destination, you can provide its specifications to your protection pack (web ACL) logging configuration to start logging to it.

Topics

- Sending protection pack (web ACL) traffic logs to a Amazon CloudWatch Logs log group

- Sending protection pack (web ACL) traffic logs to an Amazon Simple Storage Service bucket
- Sending protection pack (web ACL) traffic logs to an Amazon Data Firehose delivery stream

Sending protection pack (web ACL) traffic logs to a Amazon CloudWatch Logs log group

This topic provides information for sending your protection pack (web ACL) traffic logs to a CloudWatch Logs log group.

> **Note**
>
> You are charged for logging in addition to the charges for using AWS WAF. For information, see Pricing for logging protection pack (web ACL) traffic information.

To send logs to Amazon CloudWatch Logs, you create a CloudWatch Logs log group. When you enable logging in AWS WAF, you provide the log group ARN. After you enable logging for your protection pack (web ACL), AWS WAF delivers logs to the CloudWatch Logs log group in log streams.

When you use CloudWatch Logs, you can explore the logs for your protection pack (web ACL) in the AWS WAF console. In your protection pack (web ACL) page, select the tab **Logging insights**. This option is in addition to the logging insights that are provided for CloudWatch Logs through the CloudWatch console.

Configure the log group for AWS WAF protection pack (web ACL) logs in the same Region as the protection pack (web ACL) and using the same account as you use to manage the protection pack (web ACL). For information about configuring a CloudWatch Logs log group, see Working with Log Groups and Log Streams.

Quotas for CloudWatch Logs log groups

CloudWatch Logs has a default maximum quota for throughput, shared across all log groups within a region, which you can request to increase. If your logging requirements are too high for the current throughput setting, you'll see throttling metrics for `PutLogEvents` for your account. To view the limit in the Service Quotas console and request an increase, see the CloudWatch Logs PutLogEvents quota.

Log group naming

Your log group names must start with `aws-waf-logs-` and can end with any suffix you like, for example, `aws-waf-logs-testLogGroup2`.

The resulting ARN format is as follows:

```
arn:aws:logs:Region:account-id:log-group:aws-waf-logs-log-group-suffix
```

The log streams have the following naming format:

```
Region_web-acl-name_log-stream-number
```

The following shows an example log stream for protection pack (web ACL) `TestWebACL` in Region `us-east-1`.

```
us-east-1_TestWebACL_0
```

Permissions required to publish logs to CloudWatch Logs

Configuring protection pack (web ACL) traffic logging for a CloudWatch Logs log group requires the permissions settings described in this section. The permissions are set for you when you use one of the AWS WAF full access managed policies, `AWSWAFConsoleFullAccess` or `AWSWAFFullAccess`. If you want to manage finer-grained access to your logging and AWS WAF resources, you can set the permissions yourself. For information about managing permissions, see Access management for AWS resources in the *IAM User Guide*. For information about the AWS WAF managed policies, see AWS managed policies for AWS WAF.

These permissions allow you to change the protection pack (web ACL) logging configuration, to configure log delivery for CloudWatch Logs, and to retrieve information about your log group. These permissions must be attached to the user that you use to manage AWS WAF.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
```

```
            "Action": [
                "wafv2:PutLoggingConfiguration",
                "wafv2:DeleteLoggingConfiguration"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "LoggingConfigurationAPI"
        },
        {
            "Sid": "WebACLLoggingCWL",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        }
    ]
}
```

When actions are permitted on all AWS resources, it's indicated in the policy with a `"Resource"` setting of `"*"`. This means that the actions are permitted on all AWS resources *that each action supports*. For example, the action `wafv2:PutLoggingConfiguration` is supported only for `wafv2` logging configuration resources.

Sending protection pack (web ACL) traffic logs to an Amazon Simple Storage Service bucket

This topic provides information for sending your protection pack (web ACL) traffic logs to an Amazon S3 bucket.

> **Note**
>
> You are charged for logging in addition to the charges for using AWS WAF. For information, see Pricing for logging protection pack (web ACL) traffic information.

To send your protection pack (web ACL) traffic logs to Amazon S3, you set up an Amazon S3 bucket from the same account as you use to manage the protection pack (web ACL), and you name the bucket starting with `aws-waf-logs-`. When you enable logging in AWS WAF, you provide the bucket name. For information about creating a logging bucket, see Create a Bucket in the *Amazon Simple Storage Service User Guide*.

You can access and analyze your Amazon S3 logs using the Amazon Athena interactive query service. Athena makes it easy to analyze data directly in Amazon S3 using standard SQL. With a few actions in the AWS Management Console, you can point Athena at data stored in Amazon S3 and quickly begin using standard SQL to run ad-hoc queries and get results. For more information, see Querying AWS WAF logs in the *Amazon Athena user guide*. For additional sample Amazon Athena queries, see aws-samples/waf-log-sample-athena-queries on the GitHub website.

> **Note**
>
> AWS WAF supports encryption with Amazon S3 buckets for key type Amazon S3 key (SSE-S3) and for AWS Key Management Service (SSE-KMS) AWS KMS keys. AWS WAF doesn't support encryption for AWS Key Management Service keys that are managed by AWS.

Log files from your protection pack (web ACL) are published to the Amazon S3 bucket at 5-minute intervals. Each log file contains log records for the traffic recorded in the previous 5 minutes.

Multiple log files may be published within a 5-minute period. The log files may contain some or all of the AWS WAF log records for the traffic recorded in the previous five minutes.

The log files are compressed. If you open the files using the Amazon S3 console, Amazon S3 decompresses the log records and displays them. If you download the log files, you must decompress them to view the records.

A single log file contains interleaved entries with multiple records. To see all the log files for a protection pack (web ACL), look for entries aggregated by the protection pack (web ACL) name, Region, and your account ID.

Naming requirements and syntax

Bucket names for AWS WAF logging must start with `aws-waf-logs-` and can end with any suffix you want. For example, `aws-waf-logs-`*`LOGGING-BUCKET-SUFFIX`*.

Bucket location

The bucket locations use the following syntax:

```
s3://aws-waf-logs-LOGGING-BUCKET-SUFFIX/
```

Bucket ARN

The format of the bucket Amazon Resource Name (ARN) is as follows:

```
arn:aws:s3:::aws-waf-logs-LOGGING-BUCKET-SUFFIX
```

Bucket locations with prefixes

If you use prefixes in your object keys name to organize the data that you store in your buckets, you can provide your prefixes in your logging bucket names.

> **Note**
>
> This option is not available through the console. Use the AWS WAF APIs, CLI, or AWS CloudFormation.

For information about using prefixes in Amazon S3, see Organizing objects using prefixes in the *Amazon Simple Storage Service User Guide*.

The bucket locations with prefixes use the following syntax:

```
s3://aws-waf-logs-LOGGING-BUCKET-SUFFIX/KEY-NAME-PREFIX/
```

Bucket folders and file names

Inside your buckets, and following any prefixes that you provide, your AWS WAF logs are written under a folder structure that's determined by your account ID, the Region, the protection pack (web ACL) name, and the date and time.

```
AWSLogs/account-id/WAFLogs/Region/web-acl-name/YYYY/MM/dd/HH/mm
```

Inside the folders, the log file names follow a similar format:

```
account-id_waflogs_Region_web-acl-name_timestamp_hash.log.gz
```

The time specifications used in the folder structure and in the log file name adhere to the timestamp format specification YYYYMMddTHHmmZ.

The following shows an example log file in an Amazon S3 bucket for a bucket named `aws-waf-logs-LOGGING-BUCKET-SUFFIX`. The AWS account is 11111111111. The protection pack (web ACL) is TEST-WEBACL and the Region is `us-east-1`.

```
s3://aws-waf-logs-LOGGING-BUCKET-SUFFIX/AWSLogs/11111111111/WAFLogs/us-
east-1/TEST-WEBACL/2021/10/28/19/50/11111111111_waflogs_us-east-1_TEST-
WEBACL_20211028T1950Z_e0ca43b5.log.gz
```

> **Note**
>
> Your bucket names for AWS WAF logging must start with `aws-waf-logs-` and can end with any suffix you want.

Permissions required to publish logs to Amazon S3

Configuring protection pack (web ACL) traffic logging for an Amazon S3 bucket requires the following permissions settings. These permissions are set for you when you use one of the AWS WAF full access managed policies, `AWSWAFConsoleFullAccess` or `AWSWAFFullAccess`. If you want to further manage access to your logging and AWS WAF resources, you can set these permissions yourself. For information about managing permissions, see Access management for AWS resources in the *IAM User Guide*. For information about the AWS WAF managed policies, see AWS managed policies for AWS WAF.

The following permissions allow you to change the protection pack (web ACL) logging configuration and to configure log delivery to your Amazon S3 bucket. These permissions must be attached to the user that you use to manage AWS WAF.

> **Note**
>
> When you set the permissions listed below, you might see errors in your AWS CloudTrail logs that indicate access denied, but the permissions are correct for AWS WAF logging.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "wafv2:PutLoggingConfiguration",
            "wafv2:DeleteLoggingConfiguration"
         ],
         "Resource":[
            "*"
         ],
         "Effect":"Allow",
         "Sid":"LoggingConfigurationAPI"
      },
    {


       "Sid":"WebACLLogDelivery",


       "Action":[


          "logs:CreateLogDelivery",


          "logs:DeleteLogDelivery"


       ],


       "Resource": "*",


       "Effect":"Allow"


    },
      {
         "Sid":"WebACLLoggingS3",
```

```
         "Action":[
            "s3:PutBucketPolicy",
            "s3:GetBucketPolicy"
         ],
         "Resource": [
         "arn:aws-eusc:s3:::aws-waf-logs-amzn-s3-demo-destination-bucket-suffix"
         ],
         "Effect":"Allow"
      }
   ]
}
```

When actions are permitted on all AWS resources, it's indicated in the policy with a `"Resource"` setting of `"*"`. This means that the actions are permitted on all AWS resources *that each action supports*. For example, the action `wafv2:PutLoggingConfiguration` is supported only for `wafv2` logging configuration resources.

By default, Amazon S3 buckets and the objects that they contain are private. Only the bucket owner can access the bucket and the objects stored in it. The bucket owner, however, can grant access to other resources and users by writing an access policy.

If the user creating the log owns the bucket, the service automatically attaches the following policy to the bucket to give the log permission to publish logs to it:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite",
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws-eusc:s3:::aws-waf-logs-amzn-s3-demo-destination-
bucket-suffix/AWSLogs/123456789012/*",
      "Condition": {
        "StringEquals": {
```

```
            "s3:x-amz-acl": "bucket-owner-full-control",
            "aws:SourceAccount": ["123456789012"]
          },
          "ArnLike": {
          "aws:SourceArn": ["arn:aws-eusc:logs:eusc-de-east-1:123456789012:*"]
          }
        }
      },
      {
        "Sid": "AWSLogDeliveryAclCheck",
        "Effect": "Allow",
        "Principal": {
          "Service": "delivery.logs.amazonaws.com"
        },
        "Action": "s3:GetBucketAcl",
        "Resource": "arn:aws-eusc:s3:::aws-waf-logs-amzn-s3-demo-destination-
bucket-suffix",
        "Condition": {
          "StringEquals": {
          "aws:SourceAccount": ["123456789012"]
          },
          "ArnLike": {
          "aws:SourceArn": ["arn:aws-eusc:logs:eusc-de-east-1:123456789012:*"]
          }
        }
      }
    ]
}
```

Note

Your bucket names for AWS WAF logging must start with `aws-waf-logs-` and can end with any suffix you want.

If the user creating the log doesn't own the bucket, or doesn't have the `GetBucketPolicy` and `PutBucketPolicy` permissions for the bucket, the log creation fails. In this case, the bucket owner must manually add the preceding policy to the bucket and specify the log creator's AWS account ID. For more information, see How Do I Add an S3 Bucket Policy? in the *Amazon Simple Storage*

Service User Guide. If the bucket receives logs from multiple accounts, add a `Resource` element entry to the `AWSLogDeliveryWrite` policy statement for each account.

For example, the following bucket policy allows AWS account 111122223333 to publish logs to a bucket named `aws-waf-logs-`*`LOGGING-BUCKET-SUFFIX`*:

JSON

```
{
    "Version":"2012-10-17",
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws-eusc:s3:::aws-waf-logs-amzn-s3-demo-destination-
bucket-suffix/AWSLogs/111122223333/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": ["111122223333"]
                },
                "ArnLike": {
                    "aws:SourceArn": ["arn:aws-eusc:logs:eusc-de-
east-1:111122223333:*"]
                }
            }
        },
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws-eusc:s3:::aws-waf-logs-amzn-s3-demo-destination-
bucket-suffix",
            "Condition": {
```

```
                "StringEquals": {
                "aws:SourceAccount": ["111122223333"]
                },
                "ArnLike": {
                "aws:SourceArn": ["arn:aws-eusc:logs:eusc-de-
east-1:111122223333:*"]
                }
            }
        }
    ]
}
```

Note

In some cases, you may see `AccessDenied` errors in AWS CloudTrail if the `s3:ListBucket` permission has not been granted to `delivery.logs.amazonaws.com`. To avoid these errors in your CloudTrail logs, you must grant the `s3:ListBucket` permission to `delivery.logs.amazonaws.com` and you must include the `Condition` parameters shown with the `s3:GetBucketAcl` permission set in the preceding bucket policy. To make this simpler, instead of creating a new `Statement`, you can directly update the `AWSLogDeliveryAclCheck` to be `"Action": ["s3:GetBucketAcl", "s3:ListBucket"]`.

Permissions for using AWS Key Management Service with a KMS key

If your logging destination uses server-side encryption with keys that are stored in AWS Key Management Service (SSE-KMS) and you use a customer managed key (KMS key), you must give AWS WAF permission to use your KMS key. To do this, you add a key policy to the KMS key for your chosen destination. This permits AWS WAF logging to write your log files to your destination.

Add the following key policy to your KMS key to allow AWS WAF to log to your Amazon S3 bucket.

```
{
    "Sid": "Allow AWS WAF to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "delivery.logs.amazonaws.com"
        ]
```

```
    },
    "Action": "kms:GenerateDataKey*",
    "Resource": "*"
}
```

Permissions required to access Amazon S3 log files

Amazon S3 uses access control lists (ACLs) to manage access to the log files created by an AWS WAF log. By default, the bucket owner has `FULL_CONTROL` permissions on each log file. The log delivery owner, if different from the bucket owner, has no permissions. The log delivery account has `READ` and `WRITE` permissions. For more information, see Access Control List (ACL) Overview in the *Amazon Simple Storage Service User Guide*.

Sending protection pack (web ACL) traffic logs to an Amazon Data Firehose delivery stream

This section provides information for sending your protection pack (web ACL) traffic logs to an Amazon Data Firehose delivery stream.

> **Note**
>
> You are charged for logging in addition to the charges for using AWS WAF. For information, see Pricing for logging protection pack (web ACL) traffic information.

To send logs to Amazon Data Firehose, you send logs from your protection pack (web ACL) to an Amazon Data Firehose delivery stream which you configure in Firehose. After you enable logging, AWS WAF delivers logs to your storage destination through the HTTPS endpoint of Firehose.

One AWS WAF log is equivalent to one Firehose record. If you typically receive 10,000 requests per second and you enable full logs, you should have a 10,000 records per second setting in Firehose. If you don't configure Firehose correctly, AWS WAF won't record all logs. For more information, see Amazon Kinesis Data Firehose quotas.

For information about how to create an Amazon Data Firehose delivery stream and review your stored logs, see What is Amazon Data Firehose?

For information about creating your delivery stream, see Creating an Amazon Data Firehose delivery stream.

Configuring an Amazon Data Firehose delivery stream for your protection pack (web ACL)

Configure an Amazon Data Firehose delivery stream for your protection pack (web ACL) as follows.

- Create it using the same account as you use to manage the protection pack (web ACL).
- Create it in the same Region as the protection pack (web ACL). If you are capturing logs for Amazon CloudFront, create the firehose in US East (N. Virginia) Region, `us-east-1`.
- Give the data firehose a name that starts with the prefix `aws-waf-logs-`. For example, `aws-waf-logs-us-east-2-analytics`.
- Configure it for direct put, which allows applications to access the delivery stream directly. In the Amazon Data Firehose console, for the delivery stream **Source** setting, choose **Direct PUT or other sources**. Through the API, set the delivery stream property `DeliveryStreamType` to `DirectPut`.

> **Note**
>
> Do not use a `Kinesis stream` as your source.

Permissions required to publish logs to an Amazon Data Firehose delivery stream

To understand the permissions required for your Kinesis Data Firehose configuration, see Controlling Access with Amazon Kinesis Data Firehose.

You must have the following permissions to successfully enable protection pack (web ACL) logging with an Amazon Data Firehose delivery stream.

- `iam:CreateServiceLinkedRole`
- `firehose:ListDeliveryStreams`
- `wafv2:PutLoggingConfiguration`

For information about service-linked roles and the `iam:CreateServiceLinkedRole` permission, see Using service-linked roles for AWS WAF.

Configuring logging for a protection pack (web ACL)

This section provides instructions for configuring data protection for a protection pack (web ACL).

> **Note**
>
> You are charged for logging in addition to the charges for using AWS WAF. For information, see Pricing for logging protection pack (web ACL) traffic information.

To enable logging for a protection pack (web ACL), you must have already configured the logging destination that you're going to use. For information about your destination choices and the requirements for each, see AWS WAF logging destinations.

To configure logging for a protection pack (web ACL)

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **protection packs (web ACLs)**.
3. Choose the name of the protection pack (web ACL) that you want to enable logging for. The console takes you to the protection pack (web ACL)'s description, where you can edit it.
4. On the **Logging and metrics** tab, choose **Enable logging**.
5. Choose the logging destination type, and then choose the logging destination that you configured. You must choose a logging destination whose name begins with `aws-waf-logs-`.
6. (Optional) If you don't want some fields included in the logs, redact them. Choose the field to redact, and then choose **Add**. Repeat as necessary to redact additional fields. Redacted fields appear in the logs as `xxx`.

   > **Note**
   >
   > This setting has no impact on request sampling. You can exclude fields from request sampling by configuring protection pack (web ACL) data protection or by disabling sampling for the protection pack (web ACL).

7. (Optional) If you don't want to send all requests to the logs, add your filtering criteria and behavior. Under **Filter logs**, for each filter that you want to apply, choose **Add filter**, then choose your filtering criteria and specify whether you want to keep or drop requests that match the criteria. When you finish adding filters, if needed, modify the **Default logging behavior**.

> **Note**
>
> If you add multiple filters, AWS WAF evaluates them starting from the top.

8. Choose **Enable logging**.

> **Note**
>
> When you successfully enable logging, AWS WAF will create a service-linked role with the necessary permissions to write logs to the logging destination. For more information, see Using service-linked roles for AWS WAF.

Finding your protection pack (web ACL) records

This section explains how to find your protection pack (web ACL) records.

> **Note**
>
> You are charged for logging in addition to the charges for using AWS WAF. For information, see Pricing for logging protection pack (web ACL) traffic information.

If you can't find a log record in your logs

On rare occasions, it's possible for AWS WAF log delivery to fall below 100%, with logs delivered on a best effort basis. The AWS WAF architecture prioritizes the security of your applications over all other considerations. In some situations, such as when logging flows experience traffic throttling, this can result in records being dropped. This shouldn't affect more than a few records. If you notice a number of missing log entries, contact the AWS Support Center.

In the logging configuration for your protection pack (web ACL), you can customize what AWS WAF sends to the logs.

- **Field redaction** – You can redact the following fields from the log records for the rules that use the corresponding match settings: **URI path**, **Query string**, **Single header**, and **HTTP method**. Redacted fields appear as `REDACTED` in the logs. For example, if you redact the **Query string** field, in the logs, it will be listed as `REDACTED` for all rules that use the **Query string** match

component setting. Redaction applies only to the request component that you specify for matching in the rule, so the redaction of the **Single header** component doesn't apply to rules that match on **Headers**. For a list of the log fields, see Log fields for protection pack (web ACL) traffic.

> **Note**
>
> This setting has no impact on request sampling. You can exclude fields from request sampling by configuring protection pack (web ACL) data protection or by disabling sampling for the protection pack (web ACL).

- **Log filtering** – You can add filtering to specify which web requests are kept in the logs and which are dropped. You filter on the settings that AWS WAF applies during the web request evaluation. You can filter on the following settings:
  - **Fully qualified label** – Fully qualified labels have a prefix, optional namespaces, and label name. The prefix identifies the rule group or protection pack (web ACL) context of the rule that added the label. For information about labels, see Web request labeling in AWS WAF.
  - **Rule action** – You can filter on any normal rule action setting and also on the legacy `EXCLUDED_AS_COUNT` override option for rule group rules. For information about rule action settings, see Using rule actions in AWS WAF. For information about current and legacy rule action overrides for rule group rules, see Overriding rule group actions in AWS WAF.
    - The normal rule action filters apply to actions that are configured in rules and also to actions that are configured using the current option for overriding a rule group rule action.
    - The `EXCLUDED_AS_COUNT` log filter overlaps with the `Count` action log filter. `EXCLUDED_AS_COUNT` filters both the current and legacy options for overriding a rule group rule action to Count.

You can filter logs to include or exclude monetization events:

- To log only monetization activity: filter on `action` = `MONETIZE`
- To exclude monetization: filter where `monetization` field is not present

Log fields for protection pack (web ACL) traffic

The following list describes the possible log fields.

action

The terminating action that AWS WAF applied to the request. This indicates either allow, block, CAPTCHA, or challenge. The CAPTCHA and Challenge actions are terminating when the web request doesn't contain a valid token.

args

The query string.

captchaResponse

The CAPTCHA action status for the request, populated when a CAPTCHA action is applied to the request. This field is populated for any CAPTCHA action, whether terminating or non-terminating. If a request has the CAPTCHA action applied multiple times, this field is populated from the last time the action was applied.

The CAPTCHA action terminates web request inspection when the request either doesn't include a token or the token is invalid or expired. If the CAPTCHA action is terminating, this field includes a response code and failure reason. If the action is non-terminating, this field includes a solve timestamp. To differentiate between a terminating and non-terminating action, you can filter for a non-empty `failureReason` attribute in this field.

cfDistributionTenantId

The identifier for the CloudFront distribution tenant associated with the web request. This field is optional and only applies to protection packs (web ACLs) associated with CloudFront distribution tenants.

challengeResponse

The challenge action status for the request, populated when a Challenge action is applied to the request. This field is populated for any Challenge action, whether terminating or non-terminating. If a request has the Challenge action applied multiple times, this field is populated from the last time the action was applied.

The Challenge action terminates web request inspection when the request either doesn't include a token or the token is invalid or expired. If the Challenge action is terminating, this field includes a response code and failure reason. If the action is non-terminating, this field includes a solve timestamp. To differentiate between a terminating and non-terminating action, you can filter for a non-empty `failureReason` attribute in this field.

clientAsn

The Autonomous System Number (ASN) associated with the IP address of the web request's origin.

> **Note**
>
> **clientAsn** is logged in AWS WAF logs only when an ASN match statement is used. This field is not logged otherwise.

clientIp

The IP address of the client sending the request.

country

The source country of the request. If AWS WAF is unable to determine the country of origin, it sets this field to `-`.

excludedRules

Used only for rule group rules. The list of rules in the rule group that you have excluded. The action for these rules is set to Count.

If you override a rule to count using the override rule action option, matches aren't listed here. They're listed as the action pairs `action` and `overriddenAction`.

exclusionType

A type that indicates that the excluded rule has the action Count.

ruleId

The ID of the rule within the rule group that is excluded.

formatVersion

The format version for the log.

forwardedAsn

The Autonomous System Number (ASN) associated with the IP address of from the entity that forwarded the web request.

headers

The list of headers.

httpMethod

The HTTP method in the request.

httpRequest

The metadata about the request.

httpSourceId

The ID of the associated resource:

- For an Amazon CloudFront distribution, the ID is the *distribution-id* in the ARN syntax:

  arn:*partition*cloudfront::*account-id*:distribution/*distribution-id*

- For an Application Load Balancer, the ID is the *load-balancer-id* in the ARN syntax:

  arn:*partition*:elasticloadbalancing:*region*:*account-id*:loadbalancer/app/*load-balancer-name/load-balancer-id*

- For an Amazon API Gateway REST API, the ID is the *api-id* in the ARN syntax:

  arn:*partition*:apigateway:*region*::/restapis/*api-id*/stages/*stage-name*

- For an AWS AppSync GraphQL API, the ID is the *GraphQLApiId* in the ARN syntax:

  arn:*partition*:appsync:*region*:*account-id*:apis/*GraphQLApiId*

- For an Amazon Cognito user pool, the ID is the *user-pool-id* in the ARN syntax:

  arn:*partition*:cognito-idp:*region*:*account-id*:userpool/*user-pool-id*

- For an AWS App Runner service, the ID is the *apprunner-service-id* in the ARN syntax:

  arn:*partition*:apprunner:*region*:*account-id*:service/*apprunner-service-name/apprunner-service-id*

httpSourceName

The source of the request. Possible values: `CF` for Amazon CloudFront, `APIGW` for Amazon API Gateway, `ALB` for Application Load Balancer, `APPSYNC` for AWS AppSync, `COGNITOIDP` for Amazon Cognito, `APPRUNNER` for App Runner, and `VERIFIED_ACCESS` for Verified Access.

httpVersion

The HTTP version.

ja3Fingerprint

The JA3 fingerprint of the request.

> **Note**
>
> JA3 fingerprint inspection is available only for Amazon CloudFront distributions and Application Load Balancers.

The JA3 fingerprint is a 32-character hash derived from the TLS Client Hello of an incoming request. This fingerprint serves as a unique identifier for the client's TLS configuration. AWS WAF calculates and logs this fingerprint for each request that has enough TLS Client Hello information for the calculation.

You provide this value when you configure a JA3 fingerprint match in your protection pack (web ACL) rules. For information about creating a match against the JA3 fingerprint, see JA3 fingerprint in the Request components in AWS WAF for a rule statement.

ja4Fingerprint

The JA4 fingerprint of the request.

> **Note**
>
> JA4 fingerprint inspection is available only for Amazon CloudFront distributions and Application Load Balancers.

The JA4 fingerprint is a 36-character hash derived from the TLS Client Hello of an incoming request. This fingerprint serves as a unique identifier for the client's TLS configuration. AWS WAF calculates and logs this fingerprint for each request that has enough TLS Client Hello information for the calculation.

You provide this value when you configure a JA4 fingerprint match in your protection pack (web ACL) rules. For information about creating a match against the JA4 fingerprint, see JA4 fingerprint in the Request components in AWS WAF for a rule statement.

labels

The labels on the web request. These labels were applied by rules that were used to evaluate the request. AWS WAF logs the first 100 labels.

nonTerminatingMatchingRules

The list of non-terminating rules that matched the request. Each item in the list contains the following information.

action

The action that AWS WAF applied to the request. This indicates either count, CAPTCHA, or challenge. The CAPTCHA and Challenge are non-terminating when the web request contains a valid token.

ruleId

The ID of the rule that matched the request and was non-terminating.

ruleMatchDetails

Detailed information about the rule that matched the request. This field is only populated for SQL injection and cross-site scripting (XSS) match rule statements. A matching rule might require a match for more than one inspection criteria, so these match details are provided as an array of match criteria.

Any additional information provided for each rule varies according factors such as the rule configuration, rule match type, and details of the match. For example for rules with a CAPTCHA or Challenge action, the `captchaResponse` or `challengeResponse` will be listed. If the matching rule is in a rule group and you've overridden its configured rule action, the configured action will be provided in `overriddenAction`.

oversizeFields

The list of fields in the web request that were inspected by the protection pack (web ACL) and that are over the AWS WAF inspection limit. If a field is oversize but the protection pack (web ACL) doesn't inspect it, it won't be listed here.

This list can contain zero or more of the following values: `REQUEST_BODY`, `REQUEST_JSON_BODY`, `REQUEST_HEADERS`, and `REQUEST_COOKIES`. For more information about oversize fields, see Oversize web request components in AWS WAF.

rateBasedRuleList

The list of all rate-based rules in the web ACL that were evaluated against the request. This includes rules that could not be fully evaluated. For information about rate-based rules, see Using rate-based rule statements in AWS WAF.

rateBasedRuleId

The ID of the rate-based rule. If this has terminated the request, the ID for `rateBasedRuleId` is the same as the ID for `terminatingRuleId`. This field is set to `NO_EVALUATION_PERFORMED` when the rule could not be evaluated. This can happen when the request is missing a custom aggregation key such as a header or cookie required by `CUSTOMKEYS` aggregation.

rateBasedRuleName

The name of the rate-based rule.

limitKey

The type of aggregation that the rule is using. Possible values are `IP` for web request origin, `FORWARDED_IP` for an IP forwarded in a header in the request, `CUSTOMKEYS` for custom aggregate key settings. and `CONSTANT` for count all requests together, with no aggregation.

limitValue

Used only when rate limiting by a single IP address type. If a request contains an IP address that isn't valid, the `limitvalue` is `INVALID`.

maxRateAllowed

The maximum number of requests allowed in the specified time window for a specific aggregation instance. The aggregation instance is defined by the `limitKey` plus any additional key specifications that you've provided in the rate-based rule configuration.

evaluationWindowSec

The amount of time that AWS WAF included in its request counts, in seconds.

customValues

Unique values identified by the rate-based rule in the request. For string values, the logs print the first 32 characters of the string value. Depending on the key type, these values might be for just a key, such as for HTTP method or query string, or they might be for a key and name, such as for header and the header name.

requestHeadersInserted

The list of headers inserted for custom request handling.

requestId

The ID of the request, which is generated by the underlying host service. For Application Load Balancer, this is the trace ID. For all others, this is the request ID.

responseCodeSent

The response code sent with a custom response.

ruleGroupId

The ID of the rule group. If the rule blocked the request, the ID for `ruleGroupID` is the same as the ID for `terminatingRuleId`.

ruleGroupList

The list of rule groups that acted on this request, with match information.

terminatingRule

The rule that terminated the request. If this is present, it contains the following information.

action

The terminating action that AWS WAF applied to the request. This indicates either allow, block, CAPTCHA, or challenge. The CAPTCHA and Challenge actions are terminating when the web request doesn't contain a valid token.

ruleId

The ID of the rule that matched the request.

ruleMatchDetails

Detailed information about the rule that matched the request. This field is only populated for SQL injection and cross-site scripting (XSS) match rule statements. A matching rule might require a match for more than one inspection criteria, so these match details are provided as an array of match criteria.

Any additional information provided for each rule varies according factors such as the rule configuration, rule match type, and details of the match. For example for rules with a CAPTCHA or Challenge action, the `captchaResponse` or `challengeResponse` will be listed. If the

matching rule is in a rule group and you've overridden its configured rule action, the configured action will be provided in `overriddenAction`.

terminatingRuleId

The ID of the rule that terminated the request. If nothing terminates the request, the value is `Default_Action`.

terminatingRuleMatchDetails

Detailed information about the terminating rule that matched the request. A terminating rule has an action that ends the inspection process against a web request. Possible actions for a terminating rule include Allow, Block, CAPTCHA, and Challenge. During the inspection of a web request, at the first rule that matches the request and that has a terminating action, AWS WAF stops the inspection and applies the action. The web request might contain other threats, in addition to the one that's reported in the log for the matching terminating rule.

This is only populated for SQL injection and cross-site scripting (XSS) match rule statements. The matching rule might require a match for more than one inspection criteria, so these match details are provided as an array of match criteria.

terminatingRuleType

The type of rule that terminated the request. Possible values: RATE_BASED, REGULAR, GROUP, and MANAGED_RULE_GROUP.

timestamp

The timestamp in milliseconds.

uri

The URI of the request.

fragment

The part of a URL that follows the "#" symbol, providing additional information about the resource, for example, #section2.

webaclId

The GUID of the protection pack (web ACL).

monetizeVerifyResponse

Present when a Monetize action triggers payment verification. Contains the verification result and payment details.

monetizeVerifyResponse.verificationStatus

The result of the payment verification. Possible values: `VERIFIED`, `INSUFFICIENT_FUNDS`, `INVALID_PAYLOAD`, `INVALID_PAYMENT_REQUIREMENTS`, `SERVICE_ERROR`.

monetizeVerifyResponse.failureReason

When verification fails, the reason for the failure.

monetizeVerifyResponse.chainName

The blockchain chain used for payment. Possible values: `BASE`, `SOLANA`, `BASE_SEPOLIA`, `SOLANA_DEVNET`.

monetizeVerifyResponse.network

The blockchain network identifier (for example, `eip155:84532` for Base Sepolia).

monetizeVerifyResponse.amount

The payment amount as a decimal string.

monetizeVerifyResponse.currency

The cryptocurrency used for payment. Currently `USDC`.

monetizeVerifyResponse.asset

The token contract address on the blockchain (for example, the USDC contract address on the specified chain).

monetizeVerifyResponse.payerAddress

The wallet address of the paying AI agent.

monetizeVerifyResponse.idempotencyKey

A unique key for the payment transaction, used to prevent duplicate processing.

monetizeVerifyResponse.currencyMode

Whether real or test currency was used. Possible values: `REAL`, `TEST`.

Log examples for protection pack (web ACL) traffic

This section provides examples for logging protection pack (web ACL) traffic.

Example Rate-based rule 1: Rule configuration with one key, set to `Header:dogname`

```
    {
      "Name": "RateBasedRule",
      "Priority": 1,
      "Statement": {
        "RateBasedStatement": {
          "Limit": 100,
          "AggregateKeyType": "CUSTOM_KEYS",
          "CustomKeys": [
            {
              "Header": {
                "Name": "dogname",
                "TextTransformations": [
                  {
                    "Priority": 0,
                    "Type": "NONE"
                  }
                ]
              }
            }
          ]
        }
      },
      "Action": {
        "Block": {}
      },
      "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "RateBasedRule"
      }
    }
```

Example Rate-based rule 1: Log entry for request blocked by rate-based rule

```
{
   "timestamp":1683355579981,
   "formatVersion":1,
   "webaclId": ...,
   "terminatingRuleId":"RateBasedRule",
   "terminatingRuleType":"RATE_BASED",
   "action":"BLOCK",
```

```
    "terminatingRuleMatchDetails":[

    ],
    "httpSourceName":"APIGW",
    "httpSourceId":"EXAMPLE11:rjvegx5guh:CanaryTest",
    "ruleGroupList":[

    ],
    "rateBasedRuleList":[
       {
          "rateBasedRuleId": ...,
          "rateBasedRuleName":"RateBasedRule",
          "limitKey":"CUSTOMKEYS",
          "maxRateAllowed":100,
          "evaluationWindowSec":"120",
          "customValues":[
             {
                "key":"HEADER",
                "name":"dogname",
                "value":"ella"
             }
          ]
       }
    ],
    "nonTerminatingMatchingRules":[

    ],
    "requestHeadersInserted":null,
    "responseCodeSent":null,
    "httpRequest":{
       "clientIp":"52.46.82.45",
       "country":"FR",
       "headers":[
          {
             "name":"X-Forwarded-For",
             "value":"52.46.82.45"
          },
          {
             "name":"X-Forwarded-Proto",
             "value":"https"
          },
          {
             "name":"X-Forwarded-Port",
             "value":"443"
```

```
         },
         {
            "name":"Host",
            "value":"rjvegx5guh.execute-api.eu-west-3.amazonaws.com"
         },
         {
            "name":"X-Amzn-Trace-Id",
            "value":"Root=1-645566cf-7cb058b04d9bb3ee01dc4036"
         },
         {
            "name":"dogname",
            "value":"ella"
         },
         {
            "name":"User-Agent",
            "value":"RateBasedRuleTestKoipOneKeyModulePV2"
         },
         {
            "name":"Accept-Encoding",
            "value":"gzip,deflate"
         }
      ],
      "uri":"/CanaryTest",
      "args":"",
      "httpVersion":"HTTP/1.1",
      "httpMethod":"GET",
      "requestId":"Ed0AiHF_CGYF-DA="
   }
}
```

Example Rate-based rule 2: Rule configuration with two keys, set to `Header:dogname` and `Header:catname`

```
    {
      "Name": "RateBasedRule",
      "Priority": 1,
      "Statement": {
        "RateBasedStatement": {
          "Limit": 100,
          "AggregateKeyType": "CUSTOM_KEYS",
          "CustomKeys": [
            {
              "Header": {
```

```
                "Name": "dogname",
                "TextTransformations": [
                  {
                    "Priority": 0,
                    "Type": "NONE"
                  }
                ]
              }
            },
            {
              "Header": {
                "Name": "catname",
                "TextTransformations": [
                  {
                    "Priority": 0,
                    "Type": "NONE"
                  }
                ]
              }
            }
          ]
        }
      },
      "Action": {
        "Block": {}
      },
      "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "RateBasedRule"
      }
    }
```

Example Rate-based rule 2: Log entry for request blocked by rate-based rule

```
{
   "timestamp":1633322211194,
   "formatVersion":1,
   "webaclId":...,
   "terminatingRuleId":"RateBasedRule",
   "terminatingRuleType":"RATE_BASED",
   "action":"BLOCK",
   "terminatingRuleMatchDetails":[
```

```
  ],
  "httpSourceName":"APIGW",
  "httpSourceId":"EXAMPLE11:rjvegx5guh:CanaryTest",
  "ruleGroupList":[

  ],
  "rateBasedRuleList":[
     {
        "rateBasedRuleId":...,
        "rateBasedRuleName":"RateBasedRule",
        "limitKey":"CUSTOMKEYS",
        "maxRateAllowed":100,
        "evaluationWindowSec":"120",
        "customValues":[
           {
              "key":"HEADER",
              "name":"dogname",
              "value":"ella"
           },
           {
              "key":"HEADER",
              "name":"catname",
              "value":"goofie"
           }
        ]
     }
  ],
  "nonTerminatingMatchingRules":[

  ],
  "requestHeadersInserted":null,
  "responseCodeSent":null,
  "httpRequest":{
     "clientIp":"52.46.82.35",
     "country":"FR",
     "headers":[
        {
           "name":"X-Forwarded-For",
           "value":"52.46.82.35"
        },
        {
           "name":"X-Forwarded-Proto",
           "value":"https"
```

```
         },
         {
            "name":"X-Forwarded-Port",
            "value":"443"
         },
         {
            "name":"Host",
            "value":"23llbyn8v3.execute-api.eu-west-3.amazonaws.com"
         },
         {
            "name":"X-Amzn-Trace-Id",
            "value":"Root=1-64556629-17ac754c2ed9f0620e0f2a0c"
         },
         {
            "name":"catname",
            "value":"goofie"
         },
         {
            "name":"dogname",
            "value":"ella"
         },
         {
            "name":"User-Agent",
            "value":"Apache-HttpClient/UNAVAILABLE (Java/11.0.19)"
         },
         {
            "name":"Accept-Encoding",
            "value":"gzip,deflate"
         }
      ],
      "uri":"/CanaryTest",
      "args":"",
      "httpVersion":"HTTP/1.1",
      "httpMethod":"GET",
      "requestId":"EdzmlH5OCGYF1vQ="
   }
}
```

Example Log output for a rule that triggered on SQLi detection (terminating)

```
{
    "timestamp": 1576280412771,
    "formatVersion": 1,
```

```
    "webaclId": "arn:aws:wafv2:ap-southeast-2:111122223333:regional/webacl/
STMTest/1EXAMPLE-2ARN-3ARN-4ARN-123456EXAMPLE",
    "terminatingRuleId": "STMTest_SQLi_XSS",
    "terminatingRuleType": "REGULAR",
    "action": "BLOCK",
    "terminatingRuleMatchDetails": [
        {
            "conditionType": "SQL_INJECTION",
            "sensitivityLevel": "HIGH",
            "location": "HEADER",
            "matchedData": [
                "10",
                "AND",
                "1"
            ]
        }
    ],
    "httpSourceName": "-",
    "httpSourceId": "-",
    "ruleGroupList": [],
    "rateBasedRuleList": [],
    "nonTerminatingMatchingRules": [],
    "httpRequest": {
        "clientIp": "1.1.1.1",
        "country": "AU",
        "headers": [
            {
                "name": "Host",
                "value": "localhost:1989"
            },
            {
                "name": "User-Agent",
                "value": "curl/7.61.1"
            },
            {
                "name": "Accept",
                "value": "*/*"
            },
            {
                "name": "x-stm-test",
                "value": "10 AND 1=1"
            }
        ],
        "uri": "/myUri",
```

```
        "args": "",
        "httpVersion": "HTTP/1.1",
        "httpMethod": "GET",
        "requestId": "rid"
    },
    "labels": [
        {
            "name": "value"
        }
    ]
}
```

Example Log output for a rule that triggered on SQLi detection (non-terminating)

```
{
    "timestamp":1592357192516
    ,"formatVersion":1
    ,"webaclId":"arn:aws:wafv2:us-east-1:123456789012:global/webacl/hello-
world/5933d6d9-9dde-js82-v8aw-9ck28nv9"
    ,"terminatingRuleId":"Default_Action"
    ,"terminatingRuleType":"REGULAR"
    ,"action":"ALLOW"
    ,"terminatingRuleMatchDetails":[]
    ,"httpSourceName":"-"
    ,"httpSourceId":"-"
    ,"ruleGroupList":[]
    ,"rateBasedRuleList":[]
    ,"nonTerminatingMatchingRules":
    [{
        "ruleId":"TestRule"
        ,"action":"COUNT"
        ,"ruleMatchDetails":
        [{
            "conditionType":"SQL_INJECTION"
            ,"sensitivityLevel": "HIGH"
            ,"location":"HEADER"
            ,"matchedData":[
                "10"
                ,"and"
                ,"1"]
            }]
    }]
    ,"httpRequest":{
```

```
        "clientIp":"3.3.3.3"
        ,"country":"US"
        ,"headers":[
            {"name":"Host","value":"localhost:1989"}
            ,{"name":"User-Agent","value":"curl/7.61.1"}
            ,{"name":"Accept","value":"*/*"}
            ,{"name":"myHeader","myValue":"10 AND 1=1"}
            ]
            ,"uri":"/myUri","args":""
            ,"httpVersion":"HTTP/1.1"
            ,"httpMethod":"GET"
            ,"requestId":"rid"
    },
    "labels": [
        {
            "name": "value"
        }
    ]
}
```

Example Log output for multiple rules that triggered inside a rule group (RuleA-XSS is terminating and Rule-B is non-terminating)

```
{
    "timestamp":1592361810888,
    "formatVersion":1,
    "webaclId":"arn:aws:wafv2:us-east-1:123456789012:global/webacl/hello-
world/5933d6d9-9dde-js82-v8aw-9ck28nv9"
    ,"terminatingRuleId":"RG-Reference"
    ,"terminatingRuleType":"GROUP"
    ,"action":"BLOCK",
    "terminatingRuleMatchDetails":
    [{
        "conditionType":"XSS"
        ,"location":"HEADER"
        ,"matchedData":["<","frameset"]
    }]
    ,"httpSourceName":"-"
    ,"httpSourceId":"-"
    ,"ruleGroupList":
    [{
        "ruleGroupId":"arn:aws:wafv2:us-east-1:123456789012:global/rulegroup/hello-
world/c05lb698-1f11-4m41-aef4-99a506d53f4b"
```

```
        ,"terminatingRule":{
            "ruleId":"RuleA-XSS"
            ,"action":"BLOCK"
            ,"ruleMatchDetails":null
            }
        ,"nonTerminatingMatchingRules":
        [{
            "ruleId":"RuleB-SQLi"
            ,"action":"COUNT"
            ,"ruleMatchDetails":
            [{
                "conditionType":"SQL_INJECTION"
                ,"sensitivityLevel": "LOW"
                ,"location":"HEADER"
                ,"matchedData":[
                    "10"
                    ,"and"
                    ,"1"]
            }]
        }]
        ,"excludedRules":null
    }]
    ,"rateBasedRuleList":[]
    ,"nonTerminatingMatchingRules":[]
    ,"httpRequest":{
        "clientIp":"3.3.3.3"
        ,"country":"US"
        ,"headers":
        [
            {"name":"Host","value":"localhost:1989"}
            ,{"name":"User-Agent","value":"curl/7.61.1"}
            ,{"name":"Accept","value":"*/*"}
            ,{"name":"myHeader1","value":"<frameset onload=alert(1)>"}
            ,{"name":"myHeader2","value":"10 AND 1=1"}
            ]
        ,"uri":"/myUri"
        ,"args":""
        ,"httpVersion":"HTTP/1.1"
        ,"httpMethod":"GET"
        ,"requestId":"rid"
    },
    "labels": [
        {
            "name": "value"
```

```
        }
    ]
}
```

Example Log output for a rule that triggered for the inspection of the request body with content type JSON

AWS WAF currently reports the location for JSON body inspection as UNKNOWN.

```
{
    "timestamp": 1576280412771,
    "formatVersion": 1,
    "webaclId": "arn:aws:wafv2:ap-southeast-2:123456789012:regional/webacl/test/111",
    "terminatingRuleId": "STMTest_SQLi_XSS",
    "terminatingRuleType": "REGULAR",
    "action": "BLOCK",
    "terminatingRuleMatchDetails": [
        {
            "conditionType": "SQL_INJECTION",
            "sensitivityLevel": "LOW",
            "location": "UNKNOWN",
            "matchedData": [
                "10",
                "AND",
                "1"
            ]
        }
    ],
    "httpSourceName": "ALB",
    "httpSourceId": "alb",
    "ruleGroupList": [],
    "rateBasedRuleList": [],
    "nonTerminatingMatchingRules": [],
    "requestHeadersInserted":null,
    "responseCodeSent":null,
    "httpRequest": {
        "clientIp": "1.1.1.1",
        "country": "AU",
        "headers": [],
        "uri": "",
        "args": "",
        "httpVersion": "HTTP/1.1",
        "httpMethod": "POST",
```

```
        "requestId": "null"
    },
    "labels": [
        {
            "name": "value"
        }
    ]
}
```

Example Log output for a CAPTCHA rule against a web request with a valid, unexpired CAPTCHA token

The following log listing is for a web request that matched a rule with CAPTCHA action. The web request has a valid and unexpired CAPTCHA token, and is only noted as a CAPTCHA match by AWS WAF, similar to the behavior for the Count action. This CAPTCHA match is noted under `nonTerminatingMatchingRules`.

```
{
  "timestamp": 1632420429309,
  "formatVersion": 1,
  "webaclId": "arn:aws:wafv2:us-east-1:123456789012:regional/webacl/captcha-web-
acl/585e38b5-afce-4d2a-b417-14fb08b66c67",
  "terminatingRuleId": "Default_Action",
  "terminatingRuleType": "REGULAR",
  "action": "ALLOW",
  "terminatingRuleMatchDetails": [],
  "httpSourceName": "APIGW",
  "httpSourceId": "123456789012:b34myvfw0b:pen-test",
  "ruleGroupList": [],
  "rateBasedRuleList": [],
  "nonTerminatingMatchingRules": [
    {
      "ruleId": "captcha-rule",
      "action": "CAPTCHA",
      "ruleMatchDetails": [],
      "captchaResponse": {
        "responseCode": 0,
        "solveTimestamp": 1632420429
      }
    }
  ],
  "requestHeadersInserted": [
    {
```

```
      "name": "x-amzn-waf-test-header-name",
      "value": "test-header-value"
    }
  ],
  "responseCodeSent": null,
  "httpRequest": {
    "clientIp": "72.21.198.65",
    "country": "US",
    "headers": [
      {
        "name": "X-Forwarded-For",
        "value": "72.21.198.65"
      },
      {
        "name": "X-Forwarded-Proto",
        "value": "https"
      },
      {
        "name": "X-Forwarded-Port",
        "value": "443"
      },
      {
        "name": "Host",
        "value": "b34myvfw0b.gamma.execute-api.us-east-1.amazonaws.com"
      },
      {
        "name": "X-Amzn-Trace-Id",
        "value": "Root=1-614cc24d-5ad89a09181910c43917a888"
      },
      {
        "name": "cache-control",
        "value": "max-age=0"
      },
      {
        "name": "sec-ch-ua",
        "value": "\"Chromium\";v=\"94\", \"Google Chrome\";v=\"94\", \";Not A Brand
\";v=\"99\""
      },
      {
        "name": "sec-ch-ua-mobile",
        "value": "?0"
      },
      {
        "name": "sec-ch-ua-platform",
```

```
        "value": "\"Windows\""
      },
      {
        "name": "upgrade-insecure-requests",
        "value": "1"
      },
      {
        "name": "user-agent",
        "value": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/94.0.4606.54 Safari/537.36"
      },
      {
        "name": "accept",
        "value": "text/html,application/xhtml+xml,application/xml;q=0.9,image/
avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9"
      },
      {
        "name": "sec-fetch-site",
        "value": "same-origin"
      },
      {
        "name": "sec-fetch-mode",
        "value": "navigate"
      },
      {
        "name": "sec-fetch-user",
        "value": "?1"
      },
      {
        "name": "sec-fetch-dest",
        "value": "document"
      },
      {
        "name": "referer",
        "value": "https://b34myvfw0b.gamma.execute-api.us-east-1.amazonaws.com/pen-
test/pets"
      },
      {
        "name": "accept-encoding",
        "value": "gzip, deflate, br"
      },
      {
        "name": "accept-language",
        "value": "en-US,en;q=0.9"
```

```
      },
      {
        "name": "cookie",
        "value": "aws-waf-token=51c71352-41f5-4f6d-b676-c24907bdf819:EQoAZ/J
+AAQAAAAA:t9wvxbw042wva7E2Y6lgud/
bS6YG0CJKVAJqaRqDZ140ythKW0Zj9wKB2O8lSkYDRqf1yONcVBFo5u0eYi0tvT4rtQCXsu
+KanAardW8go4QSLw4yoED59lgV7oAhGyCalAzE7ra29j+RvvZPsQyoQuDCrtoY/TvQyMTXIXzGPDC/rKBbg=="
      }
    ],
    "uri": "/pen-test/pets",
    "args": "",
    "httpVersion": "HTTP/1.1",
    "httpMethod": "GET",
    "requestId": "GINMHHUgoAMFxug="
  }
}
```

Example Log output for a CAPTCHA rule against a web request that doesn't have a CAPTCHA token

The following log listing is for a web request that matched a rule with CAPTCHA action. The web request didn't have a CAPTCHA token, and was blocked by AWS WAF.

```
{
  "timestamp": 1632420416512,
  "formatVersion": 1,
  "webaclId": "arn:aws:wafv2:us-east-1:123456789012:regional/webacl/captcha-web-
acl/585e38b5-afce-4d2a-b417-14fb08b66c67",
  "terminatingRuleId": "captcha-rule",
  "terminatingRuleType": "REGULAR",
  "action": "CAPTCHA",
  "terminatingRuleMatchDetails": [],
  "httpSourceName": "APIGW",
  "httpSourceId": "123456789012:b34myvfw0b:pen-test",
  "ruleGroupList": [],
  "rateBasedRuleList": [],
  "nonTerminatingMatchingRules": [],
  "requestHeadersInserted": null,
  "responseCodeSent": 405,
  "httpRequest": {
    "clientIp": "72.21.198.65",
    "country": "US",
    "headers": [
```

```
      {
        "name": "X-Forwarded-For",
        "value": "72.21.198.65"
      },
      {
        "name": "X-Forwarded-Proto",
        "value": "https"
      },
      {
        "name": "X-Forwarded-Port",
        "value": "443"
      },
      {
        "name": "Host",
        "value": "b34myvfw0b.gamma.execute-api.us-east-1.amazonaws.com"
      },
      {
        "name": "X-Amzn-Trace-Id",
        "value": "Root=1-614cc240-18b57ff33c10e5c016b508c5"
      },
      {
        "name": "sec-ch-ua",
        "value": "\"Chromium\";v=\"94\", \"Google Chrome\";v=\"94\", \";Not A Brand
\";v=\"99\""
      },
      {
        "name": "sec-ch-ua-mobile",
        "value": "?0"
      },
      {
        "name": "sec-ch-ua-platform",
        "value": "\"Windows\""
      },
      {
        "name": "upgrade-insecure-requests",
        "value": "1"
      },
      {
        "name": "user-agent",
        "value": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/94.0.4606.54 Safari/537.36"
      },
      {
        "name": "accept",
```

```
        "value": "text/html,application/xhtml+xml,application/xml;q=0.9,image/
avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9"
      },
      {
        "name": "sec-fetch-site",
        "value": "cross-site"
      },
      {
        "name": "sec-fetch-mode",
        "value": "navigate"
      },
      {
        "name": "sec-fetch-user",
        "value": "?1"
      },
      {
        "name": "sec-fetch-dest",
        "value": "document"
      },
      {
        "name": "accept-encoding",
        "value": "gzip, deflate, br"
      },
      {
        "name": "accept-language",
        "value": "en-US,en;q=0.9"
      }
    ],
    "uri": "/pen-test/pets",
    "args": "",
    "httpVersion": "HTTP/1.1",
    "httpMethod": "GET",
    "requestId": "GINKHEssoAMFsrg="
  },
  "captchaResponse": {
    "responseCode": 405,
    "solveTimestamp": 0,
    "failureReason": "TOKEN_MISSING"
  }
}
```

Data protection

AWS WAF data protection settings let you implement customized and granular protection of sensitive information (passwords, API keys, authentication tokens, and other confidential data) on specific data fields such as headers, parameters, and body content.

You can configure data protection at either:

- The protection pack (web ACL) level, which applies across all output destinations.
- Logging only, which only affects the data that AWS WAF sends to the configured logging destination.

Data protection can be specified as either a substitution or hashing.

Substitution refers to replacing content with the word `REDACTED`.

Hashing refers to replacing content, from string to SHA-256 binary to Base64:

1. First, the algorithm builds a string from account_number and content.
2. It then applies SHA-256 to produce a binary hash.
3. Finally, it encodes those bytes using Base64.

> **Tip**
>
> You should review the characteristics of SHA-256 hashing to determine if it meets your requirements before you select the appropriate data protection method. We do not recommend relying on SHA-256 hashing if you intend to achieve an outcome equivalent to encryption or tokenization.

Topics

- Enabling data protection
- Data protection exceptions
- Data protection limitations
- Examples of data protection
- Configuring data protection for a protection pack (web ACL)

Enabling data protection

This section explains the data protection and log configuration options you can select from the console. You can protect data that appears in logs by enabling data protection on certain fields. Data protection can be applied to transform sensitive information in various types of outputs, including full logs, sample requests, and Security Lake.

To enable data protection in the AWS WAF console

Navigate to the **protection packs (web ACLs)** page in the console to **enable protection settings**. To enable data protection for your logs, choose whether to apply it to all logs or to a specific logging destination. For information, see Log fields for protection pack (web ACL) traffic.

> **Note**
>
> You don't need to enable logging to apply data protection on all logging. Data protection will be applied across all output destinations, regardless of whether logging is enabled.

At the bottom of the **Enable protection settings** page, select the **Add field** button on the **Data protection fields** panel. Select the field type from the drop down menu. For information about how each field's data is protected with data protection, see the table below.

Field type	Details
`Single header`	Permanently transform the specified header key value according to the specified option (hashing or subsitution). The transformed value will also be reflected in full Logs.
Body	Permanently transforms the body value. Only applicable for `RuleMatchDetails` in the log.
`Query string`	Permanently transform the query string according to the specified option (hashing or subsitution). The transformed value will also be reflected in full Logs.

Field type	Details
`Single query argument`	Permanently transform the specified query arg value according to the specified option (hashing or subsitution). The transformed value will also be reflected in full Logs.
`Single cookie`	Permanently transform the cookie value according to the specified option (hashing or subsitution). The transformed value will also be reflected in full Logs.

Data protection exceptions

When enabled, data protection will apply to the fields it is enabled on, including `RuleMatchDetails` and `rateBasedRuleList`. However, there are instances when you may want to include the protected data and content in `RuleMatchDetails` and `rateBasedRuleList` for troubleshooting and visibility purposes. In these scenarios, you can specify exceptions to the data protection for that field.

- **`ExcludeRuleMatchDetails`**: If you specify this exception for a specific field, `RuleMatchDetails` will show the value of the field and won't be in scope for data protection.
- **`ExcludeRateBasedDetails`**: If you specify this exception for a specific field, `rateBasedRuleList` will show the value of the field and won't be in scope for data protection.

  Example: The `ExcludeRateBasedDetails` rule is enabled on **SINGLE_HEADER** and **HEADER_NAME** for "dogname".

  If an exception is not applied to the rule, the value for "dogname" will appear as `REDACTED`.

```
"rateBasedRuleList":[ {"rateBasedRuleId": ...,
                       "rateBasedRuleName":"RateBasedRule", "limitKey":"CUSTOMKEYS",
                       "maxRateAllowed":100, "evaluationWindowSec":"120",
 "customValues":[
                       {"key":"HEADER", "name":"dogname", "value":"REDACTED" } ] } ]
```

  If an exception is enabled on the rule, the "dogname" value will appear in the log.

```
"rateBasedRuleList":[ {"rateBasedRuleId": ...,
                       "rateBasedRuleName":"RateBasedRule", "limitKey":"CUSTOMKEYS",
                       "maxRateAllowed":100, "evaluationWindowSec":"120",
"customValues":[
                       {"key":"HEADER", "name":"dogname", "value":"ELLA" } ] } ]
```

Warning

The data protection feature may potentially affect troubleshooting AWS WAF capabilities. These settings can cause unexpected detection and mitigation behaviors. Limit data protection for specific parameters to only those that are absolutely necessary.

Data protection limitations

The following are limitations to consider when using data protection.

QueryString and SingleQueryArg

QueryString Protection

- Data protection on `QueryString` applies to all query arguments, substituting/hashing both keys and values according to the specified settings.

QueryString in `RuleMatch` details and `RateBased` rule lists

- If data protection is applied to a single-query argument, then the entire query string will be substituted/hashed in the `RuleMatchDetails` and `RateBasedRule` section in full logs.
- If different protection methods are specified (substitution and hashing) in multiple single-query arguments, the stricter method, substitution, will be applied to the entire query string in the `RuleMatchDetails` and `RateBasedRule` section in full logs.

Cookies

> **Note**
>
> Data protection is only applied to the values of the cookie when the single header cookie is protected.

Single cookie in `RuleMatchDetails` and `RateBasedRule` lists

- If data protection is applied to a single cookie, then the entire cookie header will be substituted/hashed in the `RuleMatchDetails` and `RateBasedRule` section in full logs.
- If different protection methods are specified (substitution and hashing), the stricter method, substitution, will be applied to the entire cookie in the `RuleMatchDetails` and `RateBasedRule` section in full logs.

Examples of data protection

This section provides log examples of data protection logging of protection pack (web ACL) traffic.

DataProtection hashing

Webacl config

```
"data_protection_config": {
            "data_protections": [
                {
                    "field": {
                        "field_type": "SINGLE_QUERY_ARGUMENT",
                        "field_keys": [
                            "hoppy"
                        ]
                    },
                    "action": "HASH",
                    "exclude_rule_match_details": false,
                    "exclude_rate_based_details": false
                }
             ]
           }
```

Example DataProtection hashing: Log entry with the SingleQuery argument "hoppy" protected.

```
{
    "timestamp": 1738705092889,
    "formatVersion": 1,
    "webaclId": "arn:aws:wafv2:us-east-1:111122223333:regional/webacl/
DataProtectionhashACL/4eede063-e611-44f5-b357-ffc9d7b7fed5",
    "terminatingRuleId": "Default_Action",
    "terminatingRuleType": "REGULAR",
    "action": "ALLOW",
    "terminatingRuleMatchDetails": [],
    "httpSourceName": "APIGW",
    "httpSourceId": "746533260405:xt7v59bhn7:ABC",
    "ruleGroupList": [],
    "rateBasedRuleList": [],
    "nonTerminatingMatchingRules": [{
        "ruleId": "ProtectedSQLIHeadersVisibleInSTM",
        "action": "COUNT",
        "ruleMatchDetails": [{
                "conditionType": "SQL_INJECTION",
                "sensitivityLevel": "HIGH",
                "location": "SINGLE_QUERY_ARG",
                "matchedData": [ "z6hpYAFaMYdtiTeHhxnN5ydgRE5E1WgyVIdgqH0D3iM=" ],
                "matchedFieldName": "hoppy"
        }]
    }],
"requestHeadersInserted": null,
"responseCodeSent": null,
"httpRequest": {
    "clientIp": "54.239.98.137",
    "country": "US",
    "headers": [{
        "name": "X-Forwarded-For",
        "value": "54.239.98.137"
    }, {
        "name": "X-Forwarded-Proto",
        "value": "https"
    }, {
        "name": "X-Forwarded-Port",
        "value": "443"
    }, {
        "name": "Host",
        "value": "xt7xxx9bhn7.gamma.execute-api.us-east-1.amazonaws.com"
    }, {
        "name": "X-Amzn-Trace-Id",
```

```
        "value": "Root=1-67a288c4-27acb3cd5795dd8456b7e3c3"
    }, {
        "name": "Accept-Encoding",
        "value": "gzip"
    }, {
        "name": "User-Agent",
        "value": "okhttp/3.12.1"
    }],
    "uri": "/CanaryTest",
    "args": "hoppy=z6hpYAFaMYdtiTeHhxnN5ydgRE5E1WgyVIdgqH0D3iM=&yellow=hello&x-hoppy-
extra=generic-%3Cwords%3E-in-angle-brackets",
    "httpVersion": "HTTP/1.1",
    "httpMethod": "GET",
    "requestId": "FepO0F8fIAMEqoQ="
},
"labels": [{
    "name": "awswaf:forwardedip:geo:country:US"
}, {
    "name": "awswaf:forwardedip:geo:region:US-VA"
}]
}
```


DataProtection substitution

Webacl Config

```
"data_protection_config": {
            "data_protections": [
                {
                    "field": {
                        "field_type": "SINGLE_QUERY_ARGUMENT",
                        "field_keys": [
                            "hoppy"
                        ]
                    },
                    "action": "SUBSTITUTION",
                    "exclude_rule_match_details": false,
                    "exclude_rate_based_details": false
                }
             ]
           }
```

Example DataProtection substitution: Log entry with Single Query Argument "hoppy" protected

```
{
    "timestamp": 1738705092889,
    "formatVersion": 1,
    "webaclId": "arn:aws:wafv2:us-east-1:111122223333:regional/webacl/
DataProtectionhashACL/4eede063-e611-44f5-b357-ffc9d7b7fed5",
    "terminatingRuleId": "Default_Action",
    "terminatingRuleType": "REGULAR",
    "action": "ALLOW",
    "terminatingRuleMatchDetails": [],
    "httpSourceName": "APIGW",
    "httpSourceId": "746533260405:xt7v59bhn7:ABC",
    "ruleGroupList": [],
    "rateBasedRuleList": [],
    "nonTerminatingMatchingRules": []
"requestHeadersInserted": null,
"responseCodeSent": null,
"httpRequest": {
    "clientIp": "54.239.98.137",
    "country": "US",
    "headers": [{
        "name": "X-Forwarded-For",
        "value": "54.239.98.137"
    }, {
        "name": "X-Forwarded-Proto",
        "value": "https"
    }, {
        "name": "X-Forwarded-Port",
        "value": "443"
    }, {
        "name": "Host",
        "value": "xt7xxx9bhn7.gamma.execute-api.us-east-1.amazonaws.com"
    }, {
        "name": "X-Amzn-Trace-Id",
        "value": "Root=1-67a288c4-27acb3cd5795dd8456b7e3c3"
    }, {
        "name": "Accept-Encoding",
        "value": "gzip"
    }, {
        "name": "User-Agent",
        "value": "okhttp/3.12.1"
    }],
    "uri": "/CanaryTest",
```

```
    "args": "hoppy=REDACTED&yellow=hello&x-hoppy-extra=generic-%3Cwords%3E-in-angle-
brackets",
    "httpVersion": "HTTP/1.1",
    "httpMethod": "GET",
    "requestId": "FepO0F8fIAMEqoQ="
},
"labels": [{
    "name": "awswaf:forwardedip:geo:country:US"
}, {
    "name": "awswaf:forwardedip:geo:region:US-VA"
}]
}
```

Retaining data in RuleMatchDetails

Webacl config

```
"data_protection_config": {
            "data_protections": [
                {
                    "field": {
                        "field_type": "SINGLE_HEADER",
                        "field_keys": [
                            "hoppy"
                        ]
                    },
                    "action": "HASH",
                    "exclude_rule_match_details": true,
                    "exclude_rate_based_details": false
                }
             ]
           }
```

Example of retaining data in RuleMatchDetails: Log entry with single `Header` "hoppy" protected but the value is retained only in `RuleMatchDetails`.

```
{
    "timestamp": 1738705092889,
    "formatVersion": 1,
    "webaclId": "arn:aws:wafv2:us-east-1:111122223333:regional/webacl/
DataProtectionhashACL/4eede063-e611-44f5-b357-ffc9d7b7fed5",
    "terminatingRuleId": "Default_Action",
    "terminatingRuleType": "REGULAR",
```

```
    "action": "ALLOW",
    "terminatingRuleMatchDetails": [],
    "httpSourceName": "APIGW",
    "httpSourceId": "746533260405:xt7v59bhn7:ABC",
    "ruleGroupList": [],
    "rateBasedRuleList": [],
    "nonTerminatingMatchingRules": [{
        "ruleId": "ProtectedSQLIHeadersVisibleInSTM",
        "action": "COUNT",
        "ruleMatchDetails": [{
                "conditionType": "SQL_INJECTION",
                "sensitivityLevel": "HIGH",
                "location": "HEADER",
                "matchedData": [ "10", "AND", "1" ],
                "matchedFieldName": "hoppy"
        }]
    }],
"requestHeadersInserted": null,
"responseCodeSent": null,
"httpRequest": {
    "clientIp": "54.239.98.137",
    "country": "US",
    "headers": [{
        "name": "X-Forwarded-For",
        "value": "54.239.98.137"
    }, {
        "name": "X-Forwarded-Proto",
        "value": "https"
    }, {
        "name": "X-Forwarded-Port",
        "value": "443"
    }, {
        "name": "Host",
        "value": "xt7xxx9bhn7.gamma.execute-api.us-east-1.amazonaws.com"
    }, {
        "name": "X-Amzn-Trace-Id",
        "value": "Root=1-67a288c4-27acb3cd5795dd8456b7e3c3"
    }, {
        "name": "hoppy",
        "value": "zuomr2mxQxofg6EI6f7hMNGaJhhPxt0rFVAXog6FLxE="
    }, {
        "name": "Accept-Encoding",
        "value": "gzip"
    }, {
```

```
        "name": "User-Agent",
        "value": "okhttp/3.12.1"
    }, {
        "name": "hoppy",
        "value": "z6hpYAFaMYdtiTeHhxnN5ydgRE5E1WgyVIdgqH0D3iM="
    }],
    "uri": "/CanaryTest",
    "args": "happy=true",
    "httpVersion": "HTTP/1.1",
    "httpMethod": "GET",
    "requestId": "FepO0F8fIAMEqoQ="
},
"labels": [{
    "name": "awswaf:forwardedip:geo:country:US"
}, {
    "name": "awswaf:forwardedip:geo:region:US-VA"
}]
}
```

Retaining data in rateBasedRule

```
 "data_protection_config": {
           "data_protections": [
               {
                   "field": {
                       "field_type": "SINGLE_HEADER",
                       "field_keys": [
                           "hoppy"
                       ]
                   },
                   "action": "HASH",
                   "exclude_rule_match_details": false,
                   "exclude_rate_based_details": true
               }
            ]
          }
```

Example Retaining data in rateBasedRuleList: Log entry with the Single `Header` "hoppy" protected but the value is retained only in `rateBasedRuleList`

```
{
    "timestamp": 1683355579981,
    "formatVersion": 1,
```

```
    "webaclId": ...,
    "terminatingRuleId": "RateBasedRule",
    "terminatingRuleType": "RATE_BASED",
    "action": "BLOCK",
    "terminatingRuleMatchDetails": [],
    "httpSourceName": "APIGW",
    "httpSourceId": "EXAMPLE11:rjvegx5guh:CanaryTest",
    "ruleGroupList": [],
    "rateBasedRuleList": [{
        "rateBasedRuleId": ...,
        "rateBasedRuleName": "RateBasedRule",
        "limitKey": "CUSTOMKEYS",
        "maxRateAllowed": 100,
        "evaluationWindowSec": "120",
        "customValues": [{
            "key": "HEADER",
            "name": "hoppy",
            "value": "ella"
        }]
    }],
    "nonTerminatingMatchingRules": [],
    "requestHeadersInserted": null,
    "responseCodeSent": null,
    "httpRequest": {
        "clientIp": "52.46.82.45",
        "country": "FR",
        "headers": [{
            "name": "X-Forwarded-For",
            "value": "52.46.82.45"
        }, {
            "name": "X-Forwarded-Proto",
            "value": "https"
        }, {
            "name": "X-Forwarded-Port",
            "value": "443"
        }, {
            "name": "Host",
            "value": "rjvegx5guh.execute-api.eu-west-3.amazonaws.com"
        }, {
            "name": "X-Amzn-Trace-Id",
            "value": "Root=1-645566cf-7cb058b04d9bb3ee01dc4036"
        }, {
            "name": "hoppy",
            "value": "zuomr2mxQxofg6EI6f7hMNGaJhhPxt0rFVAXog6FLxE="
```

```
        }, {
            "name": "User-Agent",
            "value": "RateBasedRuleTestKoipOneKeyModulePV2"
        }, {
            "name": "Accept-Encoding",
            "value": "gzip,deflate"
        }],
        "uri": "/CanaryTest",
        "args": "",
        "httpVersion": "HTTP/1.1",
        "httpMethod": "GET",
        "requestId": "Ed0AiHF_CGYF-DA="
    }
}
```

Data protection for Body

AWS WAF only log subsets of Body in `RuleMatchDetails`.

Webacl config

```
 "data_protection_config": {
            "data_protections": [
                {
                    "field": {
                        "field_type": "BODY"
                    },
                    "action": "SUBSTITUTE",
                    "exclude_rule_match_details": false,
                    "exclude_rate_based_details": false
                }
             ]
           }
```

Example DataProtection for Body: Log entry with Body Subsituted in `ruleMatchDetails`.

```
{
    "timestamp": 1738705092889,
    "formatVersion": 1,
    "webaclId": "arn:aws:wafv2:us-east-1:111122223333:regional/webacl/
DataProtectionhashACL/4eede063-e611-44f5-b357-ffc9d7b7fed5",
    "terminatingRuleId": "Default_Action",
    "terminatingRuleType": "REGULAR",
```

```
    "action": "ALLOW",
    "terminatingRuleMatchDetails": [],
    "httpSourceName": "APIGW",
    "httpSourceId": "746533260405:xt7v59bhn7:ABC",
    "ruleGroupList": [],
    "rateBasedRuleList": [],
    "nonTerminatingMatchingRules": [{
        "ruleId": "ProtectedSQLIBody",
        "action": "COUNT",
        "ruleMatchDetails": [{
            "conditionType": "SQL_INJECTION",
            "sensitivityLevel": "HIGH",
            "location": "BODY",
            "matchedData": ["REDACTED"]
        }]
    }],
    "requestHeadersInserted": null,
    "responseCodeSent": null,
    "httpRequest": {
        "clientIp": "54.239.98.137",
        "country": "US",
        "headers": [{
            "name": "X-Forwarded-For",
            "value": "54.239.98.137"
        }, {
            "name": "X-Forwarded-Proto",
            "value": "https"
        }, {
            "name": "X-Forwarded-Port",
            "value": "443"
        }, {
            "name": "Host",
            "value": "xt7xxx9bhn7.gamma.execute-api.us-east-1.amazonaws.com"
        }, {
            "name": "X-Amzn-Trace-Id",
            "value": "Root=1-67a288c4-27acb3cd5795dd8456b7e3c3"
        }, {
            "name": "Accept-Encoding",
            "value": "gzip"
        }, {
            "name": "User-Agent",
            "value": "okhttp/3.12.1"
        }, {
            "name": "cookie",
```

```
            "value": "hoppy=dog;"
        }],
        "uri": "/CanaryTest",
        "args": "baloo=abc&hoppy-query=xyz&x-hoppy-extra=generic-%3Cwords%3E-in-angle-
brackets",
        "httpVersion": "HTTP/1.1",
        "httpMethod": "GET",
        "requestId": "FepO0F8fIAMEqoQ="
    },
    "labels": [{
        "name": "awswaf:forwardedip:geo:country:US"
    }, {
        "name": "awswaf:forwardedip:geo:region:US-VA"
    }]
}
```

Data protection for `SINGLE_COOKIE`

Webacl config

```
 "data_protection_config": {
           "data_protections": [
               {
                   "field": {
                       "field_type": "SINGLE_COOKIE",
                       "field_keys": [
                           "MILO"
                       ]
                   },
                   "action": "HASH",
                   "exclude_rule_match_details": false,
                   "exclude_rate_based_details": false
               }
            ]
          }
```

Example DataProtection for `SINGLE_COOKIE`: Log entry with a `SINGLE_COOKIE` named "MILO" protected.

The full Log shows the Cookie named MILO is protected in `ruleMatchDetails` and the cookie header. Only cookie values are protected and key names are excluded.

Note

All protected fields (single header, cookie, query arg) are not case sensitive. So, for this example, "MILO" matches "milo".


```
{
    "timestamp": 1738705092889,
    "formatVersion": 1,
    "webaclId": "arn:aws:wafv2:us-east-1:111122223333:regional/webacl/
DataProtectionhashACL/4eede063-e611-44f5-b357-ffc9d7b7fed5",
    "terminatingRuleId": "Default_Action",
    "terminatingRuleType": "REGULAR",
    "action": "ALLOW",
    "terminatingRuleMatchDetails": [],
    "httpSourceName": "APIGW",
    "httpSourceId": "746533260405:xt7v59bhn7:ABC",
    "ruleGroupList": [],
    "rateBasedRuleList": [],
    "nonTerminatingMatchingRules": [{
        "ruleId": "ProtectedSQLIHeadersVisibleInSTM",
        "action": "COUNT",
        "ruleMatchDetails": [{
            "conditionType": "SQL_INJECTION",
            "sensitivityLevel": "HIGH",
            "location": "COOKIE",
            "matchedData": ["zuomr2mxQxofg6EI6f7hMNGaJhhPxt0rFVAXog6FLxE="],
            "matchedFieldName": "milo"
        }]
    }],
    "requestHeadersInserted": null,
    "responseCodeSent": null,
    "httpRequest": {
        "clientIp": "54.239.98.137",
        "country": "US",
        "headers": [{
            "name": "X-Forwarded-For",
            "value": "54.239.98.137"
        }, {
            "name": "X-Forwarded-Proto",
            "value": "https"
        }, {
```

```
            "name": "X-Forwarded-Port",
            "value": "443"
        }, {
            "name": "Host",
            "value": "xt7xxx9bhn7.gamma.execute-api.us-east-1.amazonaws.com"
        }, {
            "name": "X-Amzn-Trace-Id",
            "value": "Root=1-67a288c4-27acb3cd5795dd8456b7e3c3"
        }, {
            "name": "Accept-Encoding",
            "value": "gzip"
        }, {
            "name": "User-Agent",
            "value": "okhttp/3.12.1"
        }, {
            "name": "cookie",
            "value": "hoppy=dog;milo=zuomr2mxQxofg6EI6f7hMNGaJhhPxt0rFVAXog6FLxE=;aws-
waf-token=51c71352-41f5-4f6d-b676-c24907bdf819:EQoAZ/J+AAQAAAAA:t9wvxbw042wva7E2Y6lgud/
bS6YG0CJKVAJqaRqDZ140ythKW0Zj9wKB2O8lSkYDRqf1yONcVBFo5u0eYi0tvT4rtQCXsu
+KanAardW8go4QSLw4yoED59lgV7oAhGyCalAzE7ra29j+RvvZPsQyoQuDCrtoY/TvQyMTXIXzGPDC/rKBbg=="
        }],
        "uri": "/CanaryTest",
        "args": "baloo=abc&hoppy-query=xyz&x-hoppy-extra=generic-%3Cwords%3E-in-angle-
brackets",
        "httpVersion": "HTTP/1.1",
        "httpMethod": "GET",
        "requestId": "FepO0F8fIAMEqoQ="
    },
    "labels": [{
        "name": "awswaf:forwardedip:geo:country:US"
    }, {
        "name": "awswaf:forwardedip:geo:region:US-VA"
    }]
}
```

Data protection for all cookies

You can configure data protection for cookies by using `SINGLE_HEADER`. Only cookie values are protected and key names are excluded.

```
"DataProtectionConfig": {
    "DataProtections": [
        {
```

```
            "Field": {
                "FieldType": "SINGLE_HEADER",
                "FieldKeys": ["cookie"]
            },
            "Action": "SUBSTITUTION",
            "ExcludeRuleMatchDetails": false,
            "ExcludeRateBasedDetails": false
        }
    ]
}
```

Example DataProtection for the `header "COOKIE"`: Log entry with the cookie header protected.

> **Note**
>
> The cookie name `AWS-WAF-TOKEN` is out of scope for data protection.

```
{
    "timestamp": 1738705092889,
    "formatVersion": 1,
    "webaclId": "arn:aws:wafv2:us-east-1:111122223333:regional/webacl/
DataProtectionhashACL/4eede063-e611-44f5-b357-ffc9d7b7fed5",
    "terminatingRuleId": "Default_Action",
    "terminatingRuleType": "REGULAR",
    "action": "ALLOW",
    "terminatingRuleMatchDetails": [],
    "httpSourceName": "APIGW",
    "httpSourceId": "746533260405:xt7v59bhn7:ABC",
    "ruleGroupList": [],
    "rateBasedRuleList": [],
    "nonTerminatingMatchingRules": [],
    "requestHeadersInserted": null,
    "responseCodeSent": null,
    "httpRequest": {
        "clientIp": "54.239.98.137",
        "country": "US",
        "headers": [{
            "name": "X-Forwarded-For",
            "value": "54.239.98.137"
        }, {
            "name": "X-Forwarded-Proto",
```

```
            "value": "https"
        }, {
            "name": "X-Forwarded-Port",
            "value": "443"
        }, {
            "name": "Host",
            "value": "xt7xxx9bhn7.gamma.execute-api.us-east-1.amazonaws.com"
        }, {
            "name": "X-Amzn-Trace-Id",
            "value": "Root=1-67a288c4-27acb3cd5795dd8456b7e3c3"
        }, {
            "name": "Accept-Encoding",
            "value": "gzip"
        }, {
            "name": "User-Agent",
            "value": "okhttp/3.12.1"
        }, {
            "name": "cookie",
            "value": "hoppy=REDACTED;milo=REDACTED;aws-waf-
token=51c71352-41f5-4f6d-b676-c24907bdf819:EQoAZ/J+AAQAAAAA:t9wvxbw042wva7E2Y6lgud/
bS6YG0CJKVAJqaRqDZ140ythKW0Zj9wKB2O8lSkYDRqf1yONcVBFo5u0eYi0tvT4rtQCXsu
+KanAardW8go4QSLw4yoED59lgV7oAhGyCalAzE7ra29j+RvvZPsQyoQuDCrtoY/TvQyMTXIXzGPDC/rKBbg=="
        }],
        "uri": "/CanaryTest",
        "args": "baloo=xyz=&hoppy-query=abc&x-hoppy-extra=abc",
        "httpVersion": "HTTP/1.1",
        "httpMethod": "GET",
        "requestId": "FepO0F8fIAMEqoQ="
    },
    "labels": [{
        "name": "awswaf:forwardedip:geo:country:US"
    }, {
        "name": "awswaf:forwardedip:geo:region:US-VA"
    }]
}
```


Data protection for single query arguments

You can configure data protection for a query string by using `SINGLE_QUERY_ARGUMENT`. This affects the keys and values of all query args. For the following examples, the original query string was `baloo=10 AND 1=1&hoppy=10 AND 1=1&x-hoppy-extra=generic-%3Cwords`.

Webacl config

```
"DataProtectionConfig": {
   "DataProtections": [
        {
            "Field": {
                "FieldType": "SINGLE_QUERY_ARGUMENT",
                "FieldKeys": ["hoppy"]
            },
            "Action": "SUBSTITUTION",
            "ExcludeRuleMatchDetails": false,
            "ExcludeRateBasedDetails": false
        }
    ]
}
```

Example DataProtection for SINGLE_QUERY_ARGUEMENT: Log entry with "hoppy" query string protected with substitution.

```
{
    "timestamp": 1738705092889,
    "formatVersion": 1,
    "webaclId": "arn:aws:wafv2:us-east-1:111122223333:regional/webacl/
DataProtectionSubstituteQueryString/4eede063-e611-44f5-b357-ffc9d7b7fed5",
    "terminatingRuleId": "Default_Action",
    "terminatingRuleType": "REGULAR",
    "action": "ALLOW",
    "terminatingRuleMatchDetails": [],
    "httpSourceName": "APIGW",
    "httpSourceId": "746533260405:xt7v59bhn7:ABC",
    "ruleGroupList": [],
    "rateBasedRuleList": [],
    "nonTerminatingMatchingRules": [
      {
        "ruleId": "ProtectedHoppyQueryArg",
        "action": "COUNT",
        "ruleMatchDetails": [
            {
                "conditionType": "SQL_INJECTION",
                "sensitivityLevel": "HIGH",
                "location": "SINGLE_QUERY_ARG",
                "matchedData": ["REDACTED"],
                "matchedFieldName": "hoppy"
            }]
      },
```

```
      {
        "ruleId":
 "FullQueryStringInspectionWhichDetectsTheFirstFieldWithSQLi_Baloo_IsAlsoMaskedMasked",
        "action": "COUNT",
        "ruleMatchDetails": [
            {
                "conditionType": "SQL_INJECTION",
                "sensitivityLevel": "HIGH",
                "location": "QUERY_ARGS",
                "matchedData": ["REDACTED"],
            }]
      },
      {
        "ruleId": "ProtectedBalooQueryArg",
        "action": "COUNT",
        "ruleMatchDetails": [
            {
                "conditionType": "SQL_INJECTION",
                "sensitivityLevel": "HIGH",
                "location": "SINGLE_QUERY_ARG",
                "matchedData": [ "10", "AND", "1" ],
                "matchedFieldName": "baloo"
            }]
      }
    ],
    "requestHeadersInserted": null,
    "responseCodeSent": null,
    "httpRequest": {
        "clientIp": "54.239.98.137",
        "country": "US",
        "headers": [{
            "name": "X-Forwarded-For",
            "value": "54.239.98.137"
        }, {
            "name": "X-Forwarded-Proto",
            "value": "https"
        }, {
            "name": "X-Forwarded-Port",
            "value": "443"
        }, {
            "name": "Host",
            "value": "xt7xxx9bhn7.gamma.execute-api.us-east-1.amazonaws.com"
        }, {
            "name": "X-Amzn-Trace-Id",
```

```
            "value": "Root=1-67a288c4-27acb3cd5795dd8456b7e3c3"
        }, {
            "name": "Accept-Encoding",
            "value": "gzip"
        }, {
            "name": "User-Agent",
            "value": "okhttp/3.12.1"
        }],
        "uri": "/CanaryTest",
        "args": "baloo=10 AND 1=1&hoppy=REDACTED&x-hoppy-extra=generic-%3Cwords",
        "httpVersion": "HTTP/1.1",
        "httpMethod": "GET",
        "requestId": "FepO0F8fIAMEqoQ="
    },
    "labels": [{
        "name": "awswaf:forwardedip:geo:country:US"
    }, {
        "name": "awswaf:forwardedip:geo:region:US-VA"
    }]
}
```

Data protection for query strings

You can configure data protection for a query string by using `QUERY_STRING`. This affects the keys and values of all query args. For the following examples, the original query string was `baloo=10 AND 1=1&hoppy-query=10 AND 1=1&x-hoppy-extra=generic-%3Cwords`.

Webacl config

```
"DataProtectionConfig": {
 "DataProtections": [
 {
 "Field": {
 "FieldType": "QUERY_STRING"
 },
 "Action": "SUBSTITUTION",
 "ExcludeRuleMatchDetails": false,
 "ExcludeRateBasedDetails": false
 }
 ]
}
```

Example DataProtection for QUERY_STRING: Log entry with query string protected with substitution.


```
{
    "timestamp": 1738705092889,
    "formatVersion": 1,
    "webaclId": "arn:aws:wafv2:us-east-1:111122223333:regional/webacl/
DataProtectionSubstituteQueryString/4eede063-e611-44f5-b357-ffc9d7b7fed5",
    "terminatingRuleId": "Default_Action",
    "terminatingRuleType": "REGULAR",
    "action": "ALLOW",
    "terminatingRuleMatchDetails": [],
    "httpSourceName": "APIGW",
    "httpSourceId": "746533260405:xt7v59bhn7:ABC",
    "ruleGroupList": [],
    "rateBasedRuleList": [],
    "nonTerminatingMatchingRules": [
      {
        "ruleId": "ProtectedHoppyQueryArg",
        "action": "COUNT",
        "ruleMatchDetails": [
            {
                "conditionType": "SQL_INJECTION",
                "sensitivityLevel": "HIGH",
                "location": "QUERY_STRING",
                "matchedData": ["REDACTED"]
            }]
      },
      {
        "ruleId": "ProtectedBalooQueryArg",
        "action": "COUNT",
        "ruleMatchDetails": [
            {
                "conditionType": "SQL_INJECTION",
                "sensitivityLevel": "HIGH",
                "location": "SINGLE_QUERY_ARG",
                "matchedData": [ "REDACTED" ],
                "matchedFieldName": "REDACTED"
            }]
      }
    ],
    "requestHeadersInserted": null,
    "responseCodeSent": null,
```

```
    "httpRequest": {
        "clientIp": "54.239.98.137",
        "country": "US",
        "headers": [{
            "name": "X-Forwarded-For",
            "value": "54.239.98.137"
        }, {
            "name": "X-Forwarded-Proto",
            "value": "https"
        }, {
            "name": "X-Forwarded-Port",
            "value": "443"
        }, {
            "name": "Host",
            "value": "xt7xxx9bhn7.gamma.execute-api.us-east-1.amazonaws.com"
        }, {
            "name": "X-Amzn-Trace-Id",
            "value": "Root=1-67a288c4-27acb3cd5795dd8456b7e3c3"
        }, {
            "name": "Accept-Encoding",
            "value": "gzip"
        }, {
            "name": "User-Agent",
            "value": "okhttp/3.12.1"
        }],
        "uri": "/CanaryTest",
        "args": "REDACTED",
        "httpVersion": "HTTP/1.1",
        "httpMethod": "GET",
        "requestId": "FepO0F8fIAMEqoQ="
    },
    "labels": [{
        "name": "awswaf:forwardedip:geo:country:US"
    }, {
        "name": "awswaf:forwardedip:geo:region:US-VA"
    }]
}
```


Data protection for multiple query arguments

You can configure data protection for individual query args by using `SINGLE_QUERY_ARGUMENT`. When reporting local information we use local protections. However, strings that matched in query string and cookie header have many protection configs that could apply. To simplify, the strictest

protection for `RuleMatchDetails` is applied, even if it doesn't overlap with the specific data range that matched.

For the following examples, the original query string was `baloo=is_a_good_boy&hoppy=likes_to_sleep&x-hoppy-extra=10 AND 1=1`.

```
"DataProtectionConfig": {
    "DataProtections": [
        {
            "Field": {
                "FieldType": "SINGLE_QUERY_ARGUMENT",
                "FieldKeys": ["hoppy"]
            },
            "Action": "SUBSTITUTION",
            "ExcludeRuleMatchDetails": false,
            "ExcludeRateBasedDetails": false
        },
        {
            "Field": {
                "FieldType": "SINGLE_QUERY_ARGUMENT",
                "FieldKeys": ["baloo"]
            },
            "Action": "HASH",
            "ExcludeRuleMatchDetails": false,
            "ExcludeRateBasedDetails": false
        }
    ]
}
```

Example DataProtection for multiple query arguments.

```
{
    "timestamp": 1738705092889,
    "formatVersion": 1,
    "webaclId": "arn:aws:wafv2:us-east-1:111122223333:regional/webacl/
DataProtectionSubstituteQueryString/4eede063-e611-44f5-b357-ffc9d7b7fed5",
    "terminatingRuleId": "Default_Action",
    "terminatingRuleType": "REGULAR",
    "action": "ALLOW",
    "terminatingRuleMatchDetails": [],
    "httpSourceName": "APIGW",
    "httpSourceId": "746533260405:xt7v59bhn7:ABC",
    "ruleGroupList": [],
```

```
    "rateBasedRuleList": [],
    "nonTerminatingMatchingRules": [
      {
        "ruleId": "ProtectedHoppyQueryArg",
        "action": "COUNT",
        "ruleMatchDetails": [
            {
                "conditionType": "SQL_INJECTION",
                "sensitivityLevel": "HIGH",
                "location": "SINGLE_QUERY_ARG",
                "matchedData": ["REDACTED"],
                "matchedFieldName": "hoppy"
            }]
      },
      {
        "ruleId": "ProtectedBalooQueryArg",
        "action": "COUNT",
        "ruleMatchDetails": [
            {
                "conditionType": "SQL_INJECTION",
                "sensitivityLevel": "HIGH",
                "location": "SINGLE_QUERY_ARG",
                "matchedData": ["zuomr2mxQxofg6EI6f7hMNGaJhhPxt0rFVAXog6FLxE="],
                "matchedFieldName": "baloo"
            }]
      },
      {
        "ruleId": "FullQueryStringDetects_x-hoppy-extra_IsSubstituted",
        "action": "COUNT",
        "ruleMatchDetails": [
            {
                "conditionType": "SQL_INJECTION",
                "sensitivityLevel": "HIGH",
                "location": "QUERY_ARGS",
                "matchedData": ["REDACTED"],  // Harshest of Protection Config
            }]
      }
    ],
    "requestHeadersInserted": null,
    "responseCodeSent": null,
    "httpRequest": {
        "clientIp": "54.239.98.137",
        "country": "US",
        "headers": [{
```

```
            "name": "X-Forwarded-For",
            "value": "54.239.98.137"
        }, {
            "name": "X-Forwarded-Proto",
            "value": "https"
        }, {
            "name": "X-Forwarded-Port",
            "value": "443"
        }, {
            "name": "Host",
            "value": "xt7xxx9bhn7.gamma.execute-api.us-east-1.amazonaws.com"
        }, {
            "name": "X-Amzn-Trace-Id",
            "value": "Root=1-67a288c4-27acb3cd5795dd8456b7e3c3"
        }, {
            "name": "Accept-Encoding",
            "value": "gzip"
        }, {
            "name": "User-Agent",
            "value": "okhttp/3.12.1"
        }],
        "uri": "/CanaryTest",
        "args": "baloo=zuomr2mxQxofg6EI6f7hMNGaJhhPxt0rFVAXog6FLxE=&hoppy=REDACTED&x-
hoppy-extra=10 AND 1=1",
        "httpVersion": "HTTP/1.1",
        "httpMethod": "GET",
        "requestId": "FepO0F8fIAMEqoQ="
    },
    "labels": [{
        "name": "awswaf:forwardedip:geo:country:US"
    }, {
        "name": "awswaf:forwardedip:geo:region:US-VA"
    }]
}
```


Note

You cannot specify both **QueryString Masking** and **Single Query Arg Masking** in the same webACL.

Configuring data protection for a protection pack (web ACL)

This section provides instructions for configuring data protection for a protection pack (web ACL).

To configure data protection for a protection pack (web ACL)

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **protection packs (web ACLs)**.
3. Choose the name of the protection pack (web ACL) that you want to enable data protection for. The console takes you to the protection pack (web ACL)'s description, where you can edit it.
4. On the **Logging and metrics** tab, in the **Data protection settings** pane, choose **Enable** or **Edit**.
5. Choose the scope **Global** and then make your field data protection selections. For each field data protection configuration, you can also specify exceptions to exclude from the protection behavior.
6. When you've completed your selections, choose **Save**. The interface returns to the **Logging and metrics** tab where your selections are summarized.

Testing and tuning your AWS WAF protections

This section provides guidance for testing and tuning your AWS WAF protection packs (web ACLs), rules, rule groups, IP sets, and regex pattern sets.

We recommend that you test and tune any changes to your AWS WAF protection pack (web ACL) before applying them to your website or web application traffic.

> ⚠ **Production traffic risk**
>
> Before you deploy your protection pack (web ACL) implementation for production traffic, test and tune it in a staging or testing environment until you are comfortable with the potential impact to your traffic. Then test and tune the rules in count mode with your production traffic before enabling them.

This section also provides general guidance for testing your use of rule groups that are managed by someone else. These include AWS Managed Rules rule groups, AWS Marketplace managed rule

groups, and rule groups that are shared with you by another account. For these rule groups, also follow any guidance that you get from the rule group provider.

- For the Bot Control AWS Managed Rules rule group, also see Testing and deploying AWS WAF Bot Control.
- For the account takeover prevention AWS Managed Rules rule group, also see Testing and deploying ATP.
- For the account creation fraud prevention AWS Managed Rules rule group, also see Testing and deploying ACFP.

Temporary inconsistencies during updates

When you create or change a protection pack (web ACL) or other AWS WAF resources, the changes take a small amount of time to propagate to all areas where the resources are stored. The propagation time can be from a few seconds to a number of minutes.

The following are examples of the temporary inconsistencies that you might notice during change propagation:

- After you create a protection pack (web ACL), if you try to associate it with a resource, you might get an exception indicating that the protection pack (web ACL) is unavailable.
- After you add a rule group to a protection pack (web ACL), the new rule group rules might be in effect in one area where the protection pack (web ACL) is used and not in another.
- After you change a rule action setting, you might see the old action in some places and the new action in others.
- After you add an IP address to an IP set that is in use in a blocking rule, the new address might be blocked in one area while still allowed in another.

Testing and tuning high-level steps

This section provides a checklist of the steps for testing changes to your web ACL, including any rules or rule groups that it uses.

> **Note**
>
> To follow the guidance in this section, you need to understand how to create and manage AWS WAF protections like protection packs (web ACLs), rules, and rule groups. That information is covered in earlier sections of this guide.

To test and tune your protection pack (web ACL)

Perform these steps first in a test environment, then in production.

1. **Prepare for testing**

   Prepare your monitoring environment, switch your new AWS WAF protections to count mode for testing, and create any resource associations that you need.

   See Preparing for testing your AWS WAF protections.

2. **Monitor and tune in test and production environments**

   Monitor and adjust your AWS WAF protections first in a test or staging environment, then in production, until you're satisfied that they can handle traffic as you need them to.

   See Monitoring and tuning your AWS WAF protections.

3. **Enable your protections in production**

   When you're satisfied with your test protections, switch them to production mode, clean up any unnecessary testing artifacts, and continue monitoring.

   See Enabling your protections in production.

After you've finished implementing your changes, continue monitoring your web traffic and protections in production to make sure that they're working as you want them to. Web traffic patterns can change over time, so you might need to adjust the protections occasionally.

Preparing for testing your AWS WAF protections

This section describes how to get set up to test and tune your AWS WAF protections.

> **Note**
>
> To follow the guidance in this section, you need to understand generally how to create and manage AWS WAF protections like protection packs (web ACLs), rules, and rule groups. That information is covered in earlier sections of this guide.

To prepare for testing

1. **Enable protection pack (web ACL) logging, Amazon CloudWatch metrics, and web request sampling for the protection pack (web ACL)**

   Use logging, metrics, and sampling to monitor the interaction of the protection pack (web ACL) rules with your web traffic.

   - **Logging** – You can configure AWS WAF to log the web requests that a protection pack (web ACL) evaluates. You can send logs to CloudWatch logs, an Amazon S3 bucket, or an Amazon Data Firehose delivery stream. You can redact fields and apply filtering. For more information, see Logging AWS WAF protection pack (web ACL) traffic.
   - **Amazon Security Lake** – You can configure Security Lake to collect protection pack (web ACL) data. Security Lake collects log and event data from various sources for normalization, analysis, and management. For information about this option, see What is Amazon Security Lake? and Collecting data from AWS services in the *Amazon Security Lake user guide*.
   - **Amazon CloudWatch metrics** – In your protection pack (web ACL) configuration, provide metric specifications for everything that you want to monitor. You can view metrics through the AWS WAF and CloudWatch consoles. For more information, see Monitoring with Amazon CloudWatch.
   - **Web request sampling** – You can view a sample of all web requests that your protection pack (web ACL) evaluates. For information about web request sampling, see Viewing a sample of web requests.

2. **Set your protections to Count mode**

   In your protection pack (web ACL) configuration, switch anything that you want to test to count mode. This causes the test protections to record matches against web requests without altering how the requests are handled. You'll be able to see the matches in your metrics, logs, and sampled requests, to verify the match criteria and to understand what the effects might

be on your web traffic. Rules that add labels to matching requests will add labels regardless of the rule action.

- **Rule defined in the protection pack (web ACL)** – Edit the rules in the protection pack (web ACL) and set their actions to Count.
- **Rule group** – In your protection pack (web ACL) configuration, edit the rule statement for the rule group and, in the **Rules** pane, open the **Override all rule actions** dropdown and choose **Count**. If you manage the protection pack (web ACL) in JSON, add the rules to the `RuleActionOverrides` settings in the rule group reference statement, with `ActionToUse` set to Count. The following example listing shows overrides for two rules in the `AWSManagedRulesAnonymousIpList` AWS Managed Rules rule group.

```
"ManagedRuleGroupStatement": {
  "VendorName": "AWS",
  "Name": "AWSManagedRulesAnonymousIpList",
    "RuleActionOverrides": [
      {
        "ActionToUse": {
          "Count": {}
        },
        "Name": "AnonymousIPList"
      },
      {
        "ActionToUse": {
          "Count": {}
        },
        "Name": "HostingProviderIPList"
      }
    ],
    "ExcludedRules": []
  }
},
```

For more information about rule action overrides, see Overriding rule actions in a rule group.

For your own rule group, don't modify the rule actions in the rule group itself. Rule group rules with Count action don't generate the metrics or other artifacts that you need for your testing. In addition, changing a rule group affects all protection packs (web ACLs) that use it, while the changes inside the protection pack (web ACL) configuration only affect the single protection pack (web ACL).

- **protection pack (web ACL)** – If you're testing a new protection pack (web ACL), set the default action for the protection pack (web ACL) to allow requests. This lets you try out the web ACL without affecting traffic in any way.

  In general, count mode generates more matches than production. This is because a rule that counts requests doesn't stop the evaluation of the request by the protection pack (web ACL), so rules that run later in the protection pack (web ACL) might also match the request. When you change your rule actions to their production settings, rules that allow or block requests will terminate the evaluation of requests that they match. As a result, matching requests will generally be inspected by fewer rules in the protection pack (web ACL). For more information about the effects of rule actions on the overall evaluation of a web request, see Using rule actions in AWS WAF.

  With these settings, your new protections won't alter web traffic, but will generate match information in metrics, protection pack (web ACL) logs, and request samples.

3. **Associate the protection pack (web ACL) with a resource**

   If the protection pack (web ACL) isn't already associated with the resource, associate it.

   See Associating or disassociating protection with an AWS resource.

You're now ready to monitor and tune your protection pack (web ACL).

Monitoring and tuning your AWS WAF protections

Monitor and tune your AWS WAF protections.

> **Note**
>
> To follow the guidance in this section, you need to understand generally how to create and manage AWS WAF protections like protection packs (web ACLs), rules, and rule groups. That information is covered in earlier sections of this guide.

Monitor web traffic and rule matches to verify the behavior of the protection pack (web ACL). If you find problems, adjust your rules to correct and then monitor to verify the adjustments.

Repeat the following procedure until the protection pack (web ACL) is managing your web traffic as you need it to.

To monitor and tune

1. **Monitor traffic and rule matches**

   Make sure that traffic is flowing and that your test rules are finding matching requests.

   Look for the following information for the protections that you're testing:

   - **Logs** – Access information about the rules that match a web request:
     - **Your rules** - Rules in the protection pack (web ACL) that have Count action are listed under `nonTerminatingMatchingRules`. Rules with Allow or Block are listed as the `terminatingRule`. Rules with CAPTCHA or Challenge can be either terminating or non-terminating, and so are listed under one of the two categories, according to the result of the rule match.
     - **Rule groups** - Rule groups are identified in the `ruleGroupId` field, with their rule matches categorized the same as for standalone rules.
     - **Labels** - Labels that rules have applied to the request are listed in the `Labels` field.

     For more information, see Log fields for protection pack (web ACL) traffic.
   - **Amazon CloudWatch metrics** – You can access the following metrics for your protection pack (web ACL) request evaluation.
     - **Your rules** – Metrics are grouped by the rule action. For example, when you test a rule in Count mode, its matches are listed as `Count` metrics for the protection pack (web ACL).
     - **Your rule groups** – The metrics for your rule groups are listed under the rule group metrics.
     - **Rule groups owned by another account** – Rule group metrics are generally visible only to the rule group owner. However, if you override the rule action for a rule, the metrics for that rule will be listed under your protection pack (web ACL) metrics. Additionally, labels added by any rule group are listed in your protection pack (web ACL) metrics.

       Count action rules in rule groups do NOT emit web ACL dimension metrics - only Rule, RuleGroup, and Region dimensions. This applies even when the rule group is referenced in a web ACL.

Rule groups in this category are AWS Managed Rules for AWS WAF, AWS Marketplace rule groups, Recognizing rule groups provided by other services, and rule groups that are shared with you by another account. When a protection pack (web ACL) is deployed through Firewall Manager, any rules within the WebACL that have a Count action will not display their metrics in the member account.

- **Labels** - Labels that were added to a web request during evaluation are listed in the protection pack (web ACL) label metrics. You can access the metrics for all labels, regardless of whether they were added by your rules and rule groups or by rules in a rule group that another account owns.

For more information, see Viewing metrics for your web ACL.

- **protection pack (web ACL) traffic overview dashboards** – Access summaries of the web traffic that a protection pack (web ACL) has evaluated by going to the protection pack (web ACL)'s page in the AWS WAF console and opening the **Traffic overview** tab.

  The traffic overview dashboards provide near real-time summaries of the Amazon CloudWatch metrics that AWS WAF collects when it evaluates your application web traffic.

  For more information, see Traffic overview dashboards for protection packs (web ACLs).

- **Sampled web requests** – Access information for the rules that match a sampling of the web requests. The sample information identifies matching rules by the metric name for the rule in the protection pack (web ACL). For rule groups, the metric identifies the rule group reference statement. For rules inside rule groups, the sample lists the matching rule name in `RuleWithinRuleGroup`.

  For more information, see Viewing a sample of web requests.

2. **Configure mitigations to address false positives**

   If you determine that a rule is generating false positives, by matching web requests when it shouldn't, the following options can help you tune your protection pack (web ACL) protections to mitigate.

   Correcting rule inspection criteria

   For your own rules, you often just need to adjust the settings that you're using to inspect web requests. Examples include changing the specifications in a regex pattern set, adjusting the text transformations that you apply to a request component before inspection, or switching

to using a forwarded IP address. See the guidance for the rule type that's causing problems, under Using rule statements in AWS WAF.

Correcting more complex problems

For inspection criteria that you don't control and for some complex rules, you might need to make other changes, like adding rules that explicitly allow or block requests or that eliminate requests from evaluation by the problematic rule. Managed rule groups most commonly need this type of mitigation, but other rules can too. Examples include the rate-based rule statement and the SQL injection attack rule statement.

What you do to mitigate false positives depends on your use case. The following are common approaches:

- **Add a mitigating rule** – Add a rule that runs before the new rule and that explicitly allows requests that are causing false positives. For information about rule evaluation order in a web ACL, see Setting rule priority.

  With this approach, the allowed requests are sent to the protected resource, so they never reach the new rule for evaluation. If the new rule is a paid managed rule group, this approach can also help contain the cost of using the rule group.
- **Add a logical rule with a mitigating rule** – Use logical rule statements to combine the new rule with a rule that excludes the false positives. For information, see Using logical rule statements in AWS WAF.

  For example, say you're adding an SQL injection attack match statement that's generating false positives for a category of requests. Create a rule that matches those requests, and then combine the rules using logical rule statements so that you match only on requests that both don't match the false positives criteria and do match the SQL injection attack criteria.
- **Add a scope-down statement** – For rate-based statements and managed rule group reference statements, exclude requests that result in false positives from evaluation by adding a scope-down statement inside the main statement.

  A request that doesn't match the scope-down statement never reaches the rule group or rate-based evaluation. For information about scope-down statements, see Using scope-down statements in AWS WAF. For an example, see Excluding IP range from bot management.

- **Add a label match rule** – For rule groups that use labeling, identify the label that the problematic rule is applying to requests. You might need to set the rule group rules in count mode first, if you haven't already done that. Add a label match rule, positioned to run after the rule group, that matches against the label that's being added by the problematic rule. In the label match rule, you can filter the requests that you want to allow from those that you want to block.

  If you use this approach, when you're finished testing, keep the problematic rule in count mode in the rule group, and keep your custom label match rule in place. For information about label match statements, see Label match rule statement. For examples, see Allowing a specific blocked bot and ATP example: Custom handling for missing and compromised credentials.
- **Change the version of a managed rule group** – For versioned managed rule groups, change the version that you're using. For example, you could switch back to the last static version that you were using successfully.

  This is usually a temporary fix. You might change the version for your production traffic while you continue testing the latest version in your test or staging environment, or while you wait for a more compatible version from the provider. For information about managed rule group versions, see Using managed rule groups in AWS WAF.

When you're satisfied that the new rules are matching requests as you need them to, move to the next stage of your testing and repeat this procedure. Perform the final stage of testing and tuning in your production environment.

Viewing metrics for your web ACL

This section describes how to view metrics for your protection pack (web ACL).

After you've associated a protection pack (web ACL) with one or more AWS resources, you can view the resulting metrics for the association in an Amazon CloudWatch graph.

For information about AWS WAF metrics, see AWS WAF metrics and dimensions. For information about CloudWatch metrics, see the Amazon CloudWatch User Guide.

For each of your rules in a protection pack (web ACL) and for all the requests that an associated resource forwards to AWS WAF for a protection pack (web ACL), CloudWatch lets you do the following:

- View data for the preceding hour or preceding three hours.
- Change the interval between data points.
- Change the calculation that CloudWatch performs on the data, such as maximum, minimum, average, or sum.

> **Note**
>
> AWS WAF with CloudFront is a global service and metrics are available only when you choose the **US East (N. Virginia)** Region in the AWS Management Console. If you choose another Region, no AWS WAF metrics will appear in the CloudWatch console.

To view data for the rules in a protection pack (web ACL)

1. Sign in to the AWS Management Console and open the CloudWatch console at https://eusc-de-east-1.console.amazonaws-eusc.eu/cloudwatch/.
2. If necessary, change the Region to the one where your AWS resources are located. For CloudFront, choose the US East (N. Virginia) Region.
3. In the navigation pane, under **Metrics**, choose **All metrics** and then search under the **Browse** tab for `AWS::WAFV2`.
4. Select the check box for the protection pack (web ACL) that you want to view data for.
5. Change the applicable settings:

   Statistic

   Choose the calculation that CloudWatch performs on the data.

   Time range

   Choose whether you want to view data for the preceding hour or the preceding three hours.

   Period

   Choose the interval between data points in the graph.

   Rules

   Choose the rules for which you want to view data.

> **Note**
>
> If you change the name of a rule and you want the rule's metric name to reflect the change, you must update the metric name as well. AWS WAF doesn't automatically update the metric name for a rule when you change the rule name. You can change the metric name when you edit the rule in the console, by using the rule JSON editor. You can also change both names through the APIs and in any JSON listing that you use to define your protection pack (web ACL) or rule group.

Note the following:

- If you recently associated a protection pack (web ACL) with an AWS resource, you might need to wait a few minutes for data to appear in the graph and for the metric for the protection pack (web ACL) to appear in the list of available metrics.
- If you associate more than one resource with a protection pack (web ACL), the CloudWatch data will include requests for all of them.
- You can hover the cursor over a data point to get more information.
- The graph doesn't refresh itself automatically. To update the display, choose the refresh ( ) icon.

For more information about CloudWatch metrics, see Monitoring with Amazon CloudWatch.

Traffic overview dashboards for protection packs (web ACLs)

This section describes the protection pack (web ACL) traffic overview dashboards in the AWS WAF console. After you associate a protection pack (web ACL) with one or more AWS resources and enable metrics for the protection pack (web ACL), you can access summaries of the web traffic that the protection pack (web ACL) evaluates by going to the protection pack (web ACL)'s **Traffic overview** tab in the AWS WAF console. The dashboards include near real-time summaries of the Amazon CloudWatch metrics that AWS WAF collects when it evaluates your application web traffic, including specialized AI bot and agent activity analysis.

> **Note**
>
> If you don't see anything on the dashboards, make sure you have metrics enabled for the protection pack (web ACL).

The protection pack (web ACL)'s **Traffic overview** tab contains tabbed dashboards with the following categories of information:

- **Top security insights** – Insights into your AWS WAF protections that AWS WAF obtains by directly querying the Amazon CloudWatch logs. The rest of the dashboard uses the CloudWatch metrics. These insights provide richer information, but incur the added costs of querying the CloudWatch logs. For information about the additional costs, see Amazon CloudWatch Logs Pricing.
- **AI Traffic Analysis** – Web requests analyzed for AI bot and agent activity, including bot identification, intent classification, access patterns, and temporal trends. This tab is available when your protection pack (web ACL) receives AI bot traffic
- **All traffic** – All web requests that the protection pack (web ACL) evaluates.

  The dashboard focus is on terminating actions, but you can view the matches for count rules in the following locations:

  - **Top 10 rules** pane of this dashboard. Toggle **Switch to count action** to show count rule matches.
  - **Sampled requests** tab of the protection pack (web ACL) page. This new tab includes a graph of all rule matches. For information, see Viewing a sample of web requests.
- **Anti-DDoS** – Web requests that the protection pack (web ACL) evaluates using the `AntiDDoSRuleSet` Anti-DDoS managed rule group.

  This tab is only available if you're using this rule group in your protection pack (web ACL).
- **Bot Control** – Web requests that the protection pack (web ACL) evaluates using the Bot Control managed rule group.
- If you aren't using this rule group in your protection pack (web ACL), this tab shows the results of evaluating a sampling of your web traffic against the Bot Control rules. This gives you an idea of the bot traffic that your application receives and it's free of charge.

  This rule group is part of the intelligent threat mitigation options that AWS WAF offers. For more information, see AWS WAF Bot Control and AWS WAF Bot Control rule group.

- **Account takeover prevention** – Web requests that the protection pack (web ACL) evaluates using the AWS WAF Fraud Control account takeover prevention (ATP) managed rule group. This tab is only available if you're using this rule group in your protection pack (web ACL).

  The ATP rule group is part of the AWS WAF intelligent threat mitigation offerings. For more information, see AWS WAF Fraud Control account takeover prevention (ATP) and AWS WAF Fraud Control account takeover prevention (ATP) rule group.
- **Account creation fraud prevention** – Web requests that the protection pack (web ACL) evaluates using the AWS WAF Fraud Control account creation fraud prevention (ACFP) managed rule group. This tab is only available if you're using this rule group in your protection pack (web ACL).

  The ACFP rule group is part of the AWS WAF intelligent threat mitigation offerings. For more information, see AWS WAF Fraud Control account creation fraud prevention (ACFP) and AWS WAF Fraud Control account creation fraud prevention (ACFP) rule group.

The dashboards are based on the protection pack (web ACL)'s CloudWatch metrics, and the graphs provide access to the corresponding metrics in CloudWatch. For the intelligent threat mitigation dashboards, like Bot Control, the metrics used are primarily the label metrics.

- For a list of the metrics that AWS WAF provides, see AWS WAF metrics and dimensions.
- For information about CloudWatch metrics, see the Amazon CloudWatch User Guide.

The dashboards provide summaries of your traffic patterns for the terminating actions and date range that you select. The intelligent threat mitigation dashboards include requests that the corresponding managed rule group evaluated, regardless of whether the managed rule group itself applied the terminating action. For example, if Block is selected, the **Account takeover prevention** dashboard includes information for all web requests that were both evaluated by the ATP managed rule group and blocked at some point during the protection pack (web ACL) evaluation. The requests can be blocked by the ATP managed rule group, by a rule that ran after the rule group in the protection pack (web ACL), or by the protection pack (web ACL) default action.

Viewing the dashboards for a protection pack (web ACL)

Follow the procedure in this section to access the protection pack (web ACL) dashboards and set the data filtering criteria. If you recently associated a protection pack (web ACL) with an AWS resource, you might need to wait a few minutes for data to become available in the dashboards.

The dashboards include the requests for all of the resources that you've associated with the protection pack (web ACL).

To view the Traffic overview dashboards for a protection pack (web ACL)

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **protection packs (web ACLs)** and then search for the web ACL that you're interested in.
3. Select the protection pack (web ACL). The console takes you to the protection pack (web ACL)'s page. The **Traffic overview** tab is selected by default.
4. Change the **Data filters** settings as needed.

    - **Terminating rule actions** – Select the terminating actions to include in the dashboards. The dashboards summarize the metrics for the web requests that had one of the selected actions applied by the protection pack (web ACL) evaluation. If you select all of the available actions, the dashboards include all evaluated web requests. For information about the actions, see How AWS WAF handles rule and rule group actions.
    - **Time range** – Select the time interval to view in the dashboards. You can choose to view a time frame relative to now, for example the last 3 hours or the last week, and you can select an absolute time range from a calendar.
    - **Time zone** – This setting applies when you specify an absolute time range. You can use your browser's local time zone or UTC (Coordinated Universal Time).

Review the information in the tabs that you're interested in. The data filter selections apply to all of the dashboards. In the graph panes, you can hover the cursor over a data point or an area to see any additional details.

Count action rules

You can view information for count action matches in one of two places.

- In this **Traffic overview** tab, on the **All traffic** dashboard, find the **Top 10 rules** pane and toggle **Switch to count action**. With this toggle on, the pane shows count rule matches instead of terminating rule matches.

- In the protection pack (web ACL)'s **Sampled requests** tab, see a graph of all rule matches and actions for the time range that you've set on the **Traffic overview** tab. For information about the **Sampled requests** tab, see Viewing a sample of web requests.

Amazon CloudWatch metrics

In the dashboard graph panes, you can access the CloudWatch metrics for the graphed data. Choose the option at the top of the graph pane or from the ⋮ (vertical ellipsis) dropdown menu inside the pane.

Refreshing the dashboards

The dashboards don't refresh automatically. To update the display, choose the refresh

icon.

Examples of the traffic overview dashboards for protection packs (web ACLs)

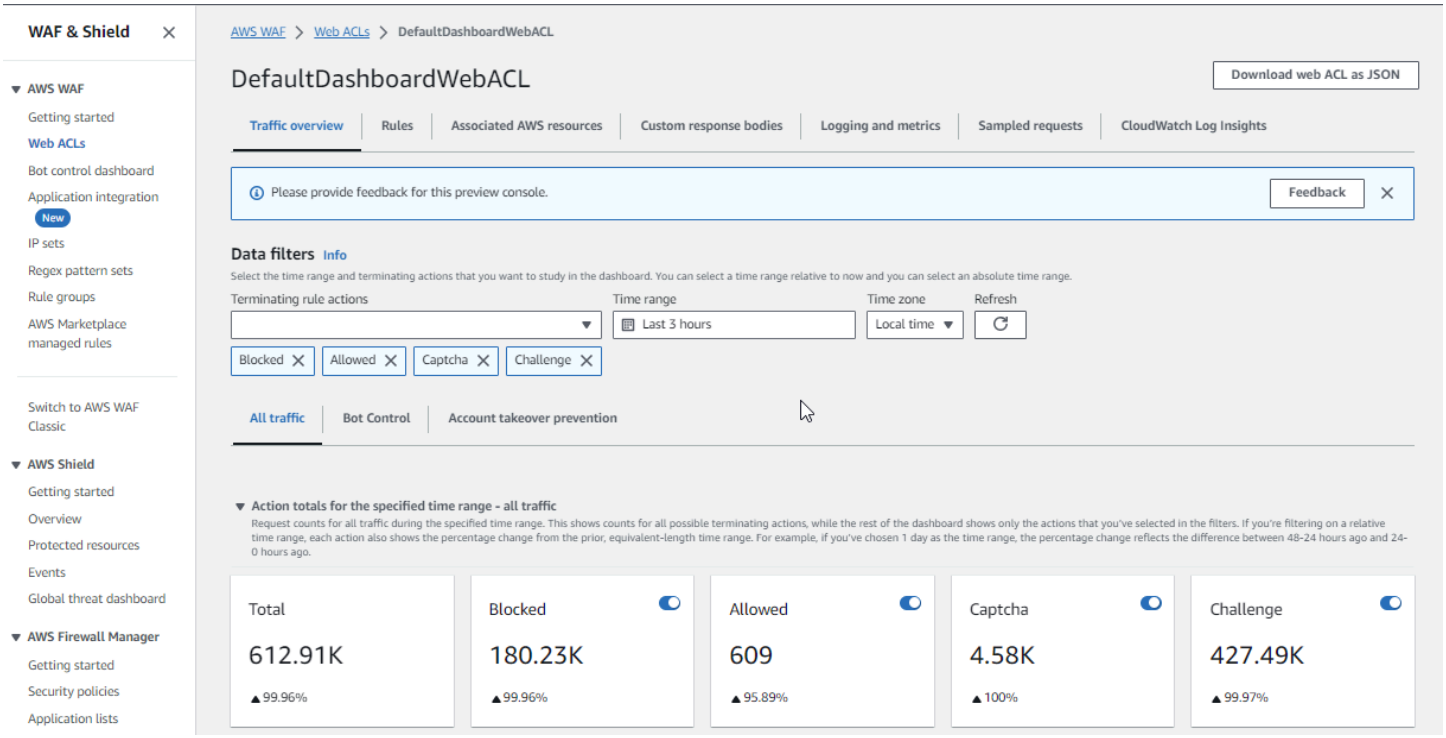
This section shows example screens of the traffic overview dashboards for protection packs (web ACLs).

> **Note**
>
> If you're already using AWS WAF to protect your application resources, you can see the dashboards for any of your protection packs (web ACLs) at its page in the AWS WAF console. For information, see Viewing the dashboards for a protection pack (web ACL).

Example screen: Data filters and All traffic dashboard action counts

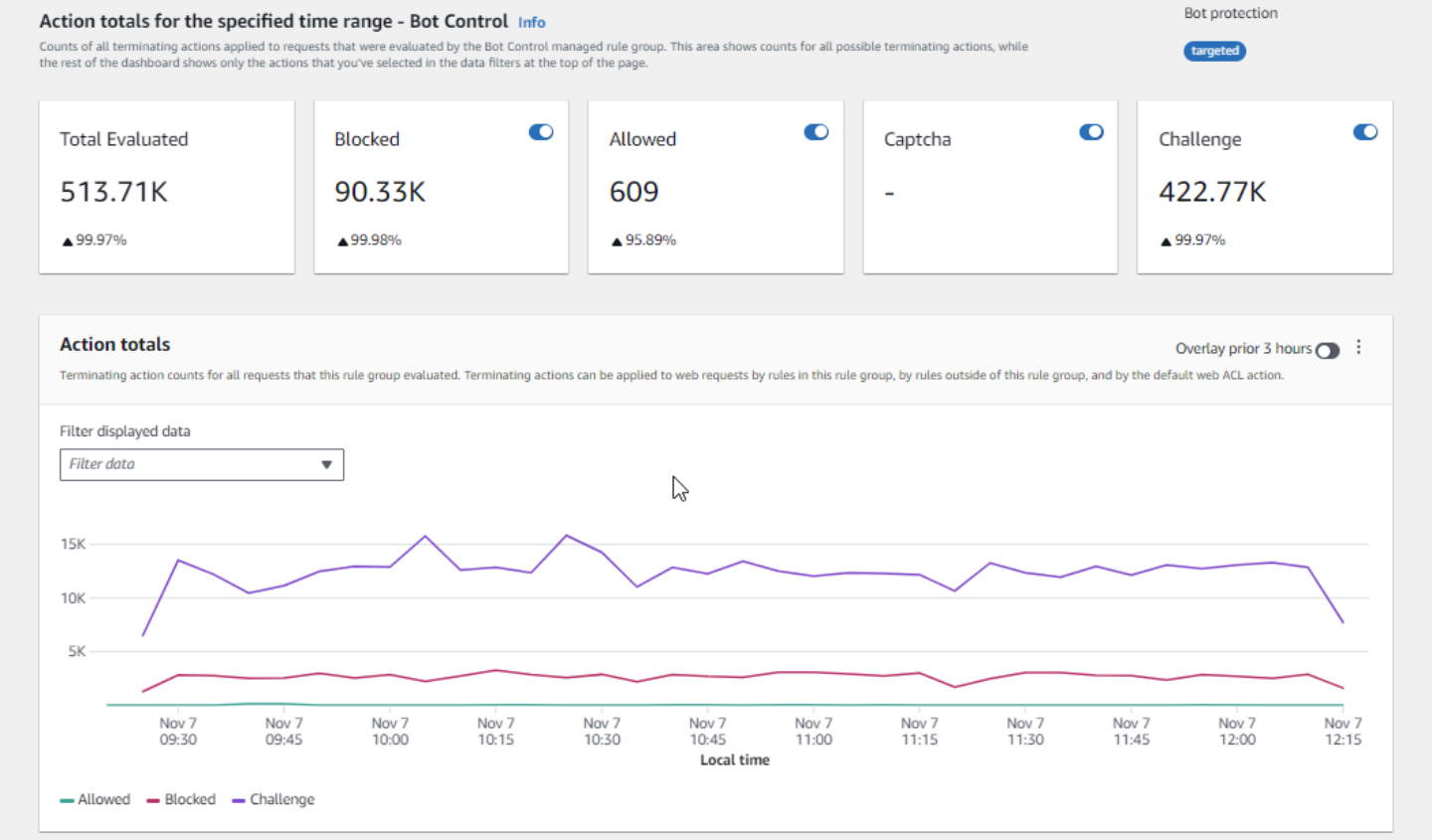
The following screenshot depicts the traffic overview for a protection pack (web ACL) with the **All traffic** tab selected. The data filters are set to the defaults: all terminating actions for the last three hours.

Inside the all traffic dashboard are the action totals for the various terminating actions. Each pane lists the request count and shows an up/down arrow indicating the change since the prior three hours time range.



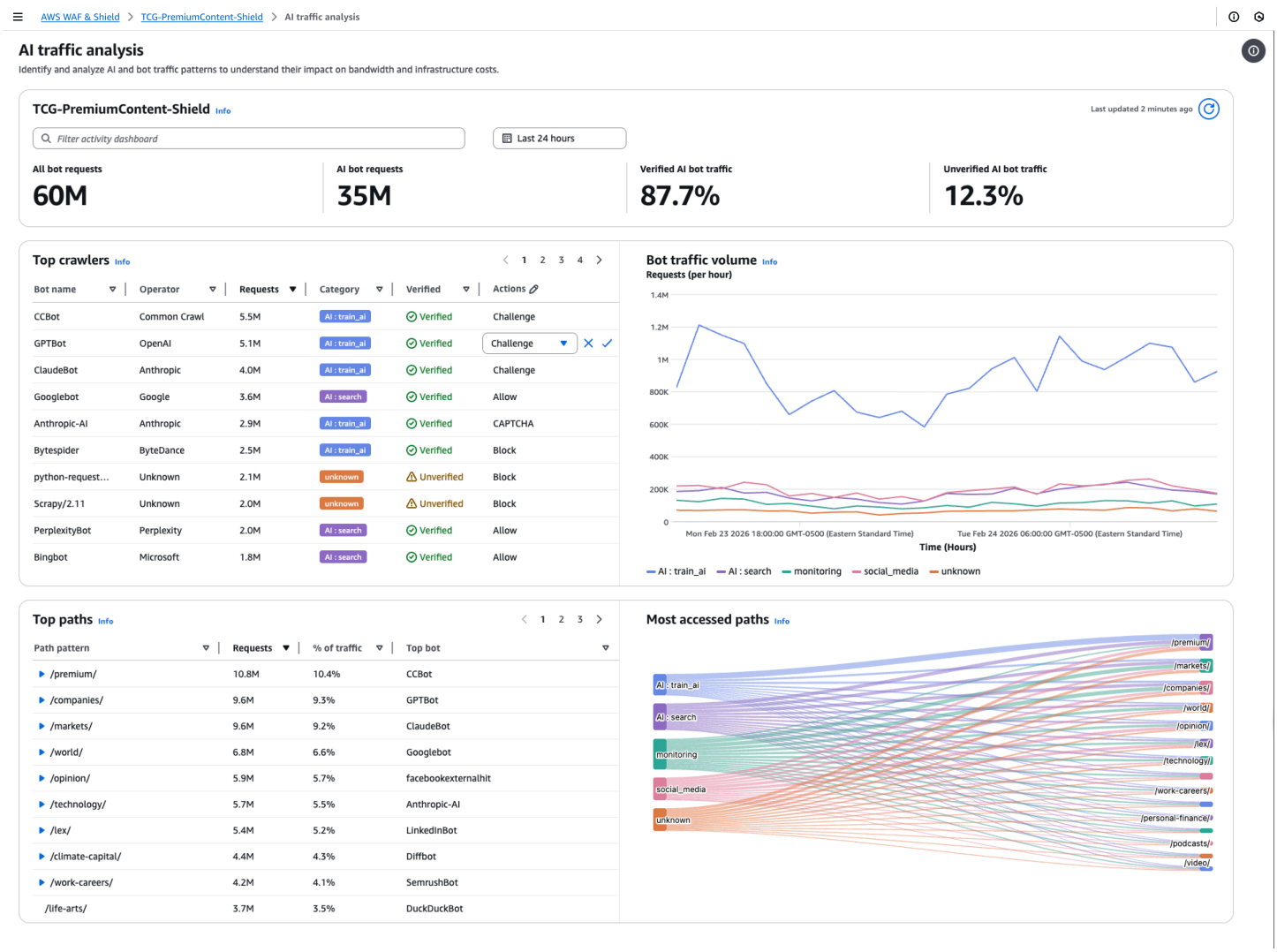
Example screen: Bot Control dashboard action counts

The following screenshot depicts action counts for the Bot Control dashboard. This shows the same totals panes for the time range, but the counts are only for requests that the Bot Control rule group evaluated. Farther down, in the **Action totals** pane, you can see the action counts throughout the specified three-hour time range. For this time range, the CAPTCHA action wasn't applied to any of the requests that the rule group evaluated.



Example screen: AI Traffic Analysis dashboard dashboard action counts

The following screenshot depicts the AI Traffic Analysis dashboard for a protection pack (web ACL). The dashboard shows AI bot activity over the selected time range with filters for bot organization, intent type, and verification status.



The dashboard includes:

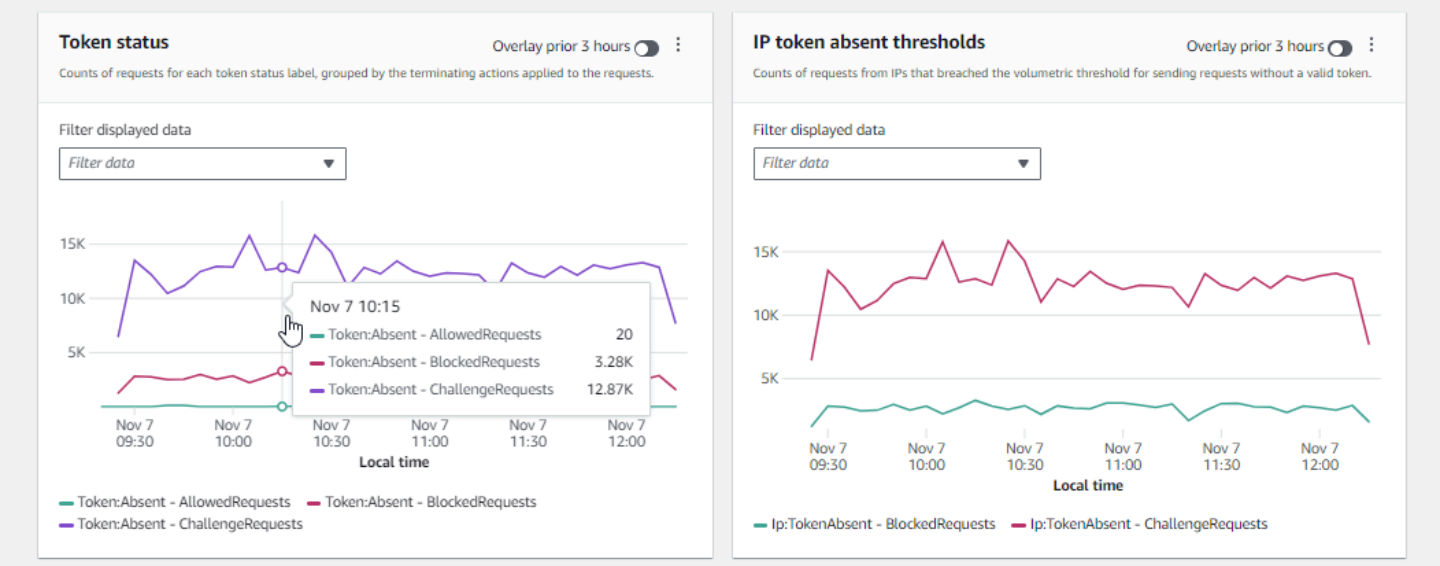
- **Bot Identity panel** – Lists detected AI bots with names and organizations
- **Intent Classification** – Categorizes bot purposes (crawling, indexing, research, etc.)
- **Access Patterns** – Top URLs accessed by AI agents with request counts
- **Temporal Analysis** – Hourly and daily activity trends with 14-day historical view
- **Organization Breakdown** – Traffic volume by bot owner organization

Example screen: Bot Control dashboard token status summary graphs

The following screenshot depicts two of the summary graphics available in the Bot Control dashboard. The **Token status** pane shows counts for the various token status labels, paired with the

rule action that was applied to the request. The **IP token absent thresholds** pane shows data for requests from IPs that were sending too many requests without a token.

Hovering over any area in the graph brings up the available information details. In the **Token status** pane in this screenshot, the mouse is hovering over a point in time, without being on any graph line, so the console displays the data for all lines at that point in time.



This section shows just a few of the traffic summaries that are provided in the protection pack (web ACL) traffic overview dashboards. To see the dashboards for any of your protection packs (web ACLs), open the protection pack (web ACL)'s page in the console. For information about how to do this, see the guidance at Viewing the dashboards for a protection pack (web ACL).

Viewing a sample of web requests

This section describes the protection pack (web ACL) **Sampled requests** tab in the AWS WAF console. In this tab, you can view a graph of all of the rule matches for web requests that AWS WAF has inspected. Additionally, if you have request sampling enabled for the protection pack (web ACL), you can see a table view of a sample of the web requests that AWS WAF has inspected. You can also retrieve sampled request information through the API call `GetSampledRequests`.

The sample of requests contains up to 100 requests that matched the criteria for a rule in the protection pack (web ACL) and another 100 requests for requests that didn't match any rules and had the protection pack (web ACL) default action applied. The requests in the sample come from all the protected resources that have received requests for your content in the previous three hours.

When a web request matches the criteria in a rule and the action for that rule doesn't terminate the request evaluation, AWS WAF continues inspecting the web request using the subsequent rules

in the protection pack (web ACL). Because of this, a web request could appear multiple times. For information about rule action behaviors, see Using rule actions in AWS WAF.

To view the all rules graph and sampled requests

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. In the navigation pane, choose **protection packs (web ACLs)**.
3. Choose the name of the protection pack (web ACL) for which you want to view requests. The console takes you to the protection pack (web ACL)'s description, where you can edit it.
4. In the **Sampled requests** tab, you can see the following:

   - **All rules graph** – This graph shows the matching rules and rule actions for all web request evaluations that were performed during the indicated time range.

     > **Note**
     >
     > The time range for this graph is set in the protection pack (web ACL)'s **Traffic overview** tab, in the **Data filters** section. For information, see Viewing the dashboards for a protection pack (web ACL).

   - **Sampled requests table** – This table displays sampled request data for the last 3 hours.

     > **Note**
     >
     > If you aren't seeing the samples that you expect for a managed rule group, see the section below this procedure.

     For each entry, the table displays the following data:

     Metric name

     The CloudWatch metric name for the rule in the protection pack (web ACL) that matched the request. If a web request doesn't match any rule in the protection pack (web ACL), this value is **Default**.

> **Note**
>
> If you change the name of a rule and you want the rule's metric name to reflect the change, you must update the metric name as well. AWS WAF doesn't automatically update the metric name for a rule when you change the rule name. You can change the metric name when you edit the rule in the console, by using the rule JSON editor. You can also change both names through the APIs and in any JSON listing that you use to define your protection pack (web ACL) or rule group.

Source IP

Either the IP address that the request originated from or, if the viewer used an HTTP proxy or an Application Load Balancer to send the request, the IP address of the proxy or Application Load Balancer.

URI

The part of a URL that identifies a resource, for example, `/images/daily-ad.jpg`.

Rule inside rule group

If the metric name identifies a rule group reference statement, this identifies the rule inside the rule group that matched the request.

Action

Indicates the action for the corresponding rule. For information about the possible rule actions, see Using rule actions in AWS WAF.

> **Note**
>
> Sampled requests for rules with Count action in rule groups are not available in the Web ACL view. Count metrics and sampled requests for rule group rules are only visible to the rule group owner.

Time

The time that AWS WAF received the request from the protected resource.

To display additional information about the components of a web request, choose the name of the URI in the row of the request.

Sampled requests for rules in managed rule groups

The console shows metrics for rule groups with "rule inside rule group" specifying the rule that was triggered. You can view metrics for default action rulesets and rules using the most recent `RuleActionOverrides` setting. For rules using the older `ExcludedRules` setting, select the specific rule from within the ruleset from the **Sampled Requests** metrics rule dropdown.

If you see the older settings, replace them with the new settings to start making the sampled requests available through the console. You can do this through the console by editing the managed rule group in the protection pack (web ACL) and saving it. AWS WAF automatically replaces any older settings with the `RuleActionOverrides` settings and sets the rule action override to Count. For more information about these two settings, see JSON listing: `RuleActionOverrides` replaces `ExcludedRules`.

You can access sampled requests for a rule that has the old override in place through the AWS WAF REST API, SDKs, or command line. For information, see GetSampledRequests in the *AWS WAF API Reference*.

The following shows the syntax for the command line request:

```
aws wafv2 get-sampled-requests \
  --web-acl-arn webACL ARN \
  --rule-metric-name Metric name of the rule in the managed rule group \
  --scope=REGIONAL or CLOUDFRONT \
  --time-window StartTime=UTC timestamp,EndTime=UTC timestamp \
  --max-items 100
```

Enabling your protections in production

This section provides instructions for enabling your tuned protections in production.

When you've finished the final stage of testing and tuning in your production environment, enable your protections in production mode.

> **Production traffic risk**
>
> Before you deploy your protection pack (web ACL) implementation for production traffic, test and tune it in a test environment until you are comfortable with the potential impact to your traffic. Also test and tune it in count mode with your production traffic before enabling your protections for production traffic.

> **Note**
>
> To follow the guidance in this section, you need to understand generally how to create and manage AWS WAF protections like protection packs (web ACLs), rules, and rule groups. That information is covered in earlier sections of this guide.

Perform these steps first in your test environment, then in production.

Enable your AWS WAF protections in production

1. **Switch to your production protections**

   Update your protection pack (web ACL) and switch your settings for production.

   a. **Remove any test rules that you don't need**

      If you added test rules that you don't need in production, remove them. If you're using any label matching rules to filter the results of managed rule group rules, be sure to leave those in place.

   b. **Switch to production actions**

      Change the action settings for your new rules to their intended production settings.

      - **Rule defined in the protection pack (web ACL)** – Edit the rules in the protection pack (web ACL) and change their actions from Count to their production actions.
      - **Rule group** – In your protection pack (web ACL) configuration of the rule group, switch rules to use their own actions or leave them with the Count action override, according to

the results of your testing and tuning activities. If you're using a label matching rule to filter the results of a rule group rule, be sure to leave the override for that rule in place.

To switch to using a rule's action, in your protection pack (web ACL) configuration, edit the rule statement for the rule group and remove the Count override for the rule. If you manage the protection pack (web ACL) in JSON, in the rule group reference statement, remove the entry for the rule from the `RuleActionOverrides` list.

- **protection pack (web ACL)** – If you changed the protection pack (web ACL) default action for your tests, switch it to its production setting.

With these settings, your new protections will be managing web traffic as you intend.

When you save your protection pack (web ACL), the resources that it's associated with will be using your production settings.

2. **Monitor and tune**

   To be sure that web requests are being handled as you want, closely monitor your traffic after you enable the new functionality. You'll be monitoring metrics and logs for your production rule actions, instead of the count actions that you were monitoring for in your tuning work. Keep monitoring and adjust the behavior as needed to adapt to changes in your web traffic.

Using AWS WAF with Amazon CloudFront

Learn how to use AWS WAF with Amazon CloudFront features.

When you create a protection pack (web ACL), you can specify one or more CloudFront distributions that you want AWS WAF to inspect. CloudFront supports two types of distributions: standard distributions that protect individual tenants, and multi-tenant distributions that protect multiple tenants through a single, shared configuration template. AWS WAF inspects web requests for both distribution types based on the rules you define in your protection packs (web ACLs), with different implementation patterns for each type.

Topics

- How AWS WAF works with different distribution types
- Using AWS WAF with CloudFront Flat-Rate Pricing Plans
- AI traffic monetization with CloudFront

- Common use cases for protecting CloudFront distributions with AWS WAF

How AWS WAF works with different distribution types

Distribution types

AWS WAF provides web application firewall capabilities for both standard and multi-tenant distribution CloudFront distributions.

Standard distributions

For standard distributions, AWS WAF adds protection using a single protection pack (web ACL) for each distribution. You can enable this protection by associating an existing protection pack (web ACL) with a CloudFront distribution or by using one-click protection in the CloudFront console. This lets you manage the security controls for each of your distributions independently, since any changes to a protection pack (web ACL) will only affect the distribution associated with it.

This straightforward method of protecting CloudFront distributions is optimal for providing individual domains with specific protections from a single protection pack (web ACL).

Standard distribution considerations

- Changes to a protection pack (web ACL) affect only its associated distribution
- Each distribution requires independent protection pack (web ACL) configuration
- Rules and rule groups are managed separately for each distribution

Multi-tenant distributions

For multi-tenant distributions, AWS WAF adds protection across multiple domains using a single protection pack (web ACL). Domains that are managed by multi-tenant distributions are known as distribution tenants. You can only enable AWS WAF protection for multi-tenant distributions in the CloudFront console, either during or after the multi-tenant distribution creation process. However, changes to a protection pack (web ACL) are still managed through the AWS WAF console or API.

Multi-tenant distributions offer the flexibility to enable AWS WAF protections at two levels:

- **Multi-tenant distribution level** – Associated protection packs (web ACLs) provide baseline security controls that apply to all applications sharing that distribution

- **Distribution tenant level** – Individual tenants within a multi-tenant distribution can have their own protection packs (web ACLs) to implement additional security controls or override multi-tenant distribution settings

These two tiers make multi-tenant distributions optimal for sharing AWS WAF protections across multiple domains without losing the ability to customize security for an individual distribution.

Multi-tenant distribution considerations

- Individual distribution tenants inherit changes made to protection packs (web ACLs) that are associated with related multi-tenant distributions
- The protection packs (web ACLs) associated with specific distribution tenants can override settings configured at the multi-tenant protection pack (web ACL) level
- Managed rule groups can be implemented at both distribution and distribution tenant levels
- Application identifiers can be located in logs to track security events by distribution

AWS WAF features by distribution type

Compare protection pack (web ACL) implementations

AWS WAF Feature	Standard distributions	Multi-tenant distributions
Associating protection packs (web ACLs)	One protection pack (web ACL) per distribution	You can share protection packs (web ACLs) across tenants, with optional tenant-specific protection packs (web ACLs)
Rule management	Rules affect a single distribut ion	Multi-tenant distribution rules affect all associated tenants; distribution tenant-specific rules affect only that tenant
Managed rule groups	Applied to individual distribut ions	Can be applied at multi-ten ant distribution level for all tenants or at tenant level for specific applications

AWS WAF Feature	Standard distributions	Multi-tenant distributions
Logging	Standard AWS WAF logs	Logs include tenant identifiers for security event attribution

Using AWS WAF with CloudFront Flat-Rate Pricing Plans

CloudFront flat-rate pricing plans combine the Amazon CloudFront global content delivery network (CDN) with multiple AWS services and features into a monthly price with no overage charges, regardless of traffic spikes or attacks.

Flat-rate pricing plans include the following AWS services and features for a simple monthly price:

- CloudFront CDN
- AWS WAF and DDoS protection
- Bot management and analytics
- Amazon Route 53 DNS
- Amazon CloudWatch Logs ingestion
- TLS certificate
- Serverless edge compute
- Amazon S3 storage credits each month

Plans are available in Free, Pro, Business, and Premium tiers to match your application's needs. Plans do not need an annual commitment to get the best available rates. Start with the Free plan and upgrade to access more capabilities and larger usage allowances.

For more information and a complete list of plans and features, see CloudFront flat-rate pricing plans in the *Amazon CloudFront Developer Guide*.

> **Important**
>
> A valid AWS WAF protection pack (web ACL) must remain associated with your CloudFront distribution when using any pricing plan. You cannot remove the protection pack (web ACL) association unless you switch back to pay-as-you-go pricing.

While a AWS WAF web ACL must remain associated with your distribution, you maintain full control over your security configuration. You can customize your protection by adjusting which rules are enabled or disabled in your web ACL, and modify rule settings to match your security requirements. For information about managing web ACL rules, see AWS WAF Rules.

AI traffic monetization with CloudFront

AI traffic monetization is available exclusively for AWS WAF web ACL resources associated with Amazon CloudFront distributions. Payment verification and settlement occur at CloudFront edge locations, minimizing latency for agents worldwide.

Considerations when using AI Traffic Monetization with CloudFront Functions and Lambda@Edge

When you use CloudFront Functions or Lambda@Edge with distributions that have Monetize rules in AWS WAF, note the following behavior for AWS WAF-generated responses.

AWS WAF-generated responses (402 Payment Required Challenge)

Viewer-response function (CloudFront Functions and Lambda@Edge) does not run on AWS WAF-generated 402 response. You cannot use these functions to customize the Payment Required Challenge. You also cannot add headers to it or modify it. If you need to add custom headers to 402 responses (for example, CORS or analytics headers), use a Response Headers Policy instead. Response Headers Policies apply to AWS WAF-generated responses.

For more information about CloudFront Functions and response header policies, see the following:

- Viewer-response functions don't run on HTTP status codes 400 and higher. For more information about edge function restrictions, see HTTP status codes.
- For more information about response header policies, see Understanding response headers policies.

Paid responses (200 after settlement)

After a successful payment settlement, the origin response passes through the normal CloudFront response pipeline, including viewer-response function. A viewer-response function can modify

the response that the agent receives after paying. For example, it can change the status code or remove headers.

Common use cases for protecting CloudFront distributions with AWS WAF

The following AWS WAF features work the same way for all CloudFront distributions. Considerations for multi-tenant distributions are listed following each feature scenario.

Using AWS WAF with CloudFront custom error pages

By default, when AWS WAF blocks a web request based on the criteria that you specify, it returns HTTP status code `403 (Forbidden)` to CloudFront, and CloudFront returns that status code to the viewer. The viewer then displays a brief and sparsely formatted default message similar to the following:

```
Forbidden: You don't have permission to access /myfilename.html on this server.
```

You can override this behavior in your AWS WAF protection pack (web ACL) rules by defining custom responses. For more information about customizing response behavior using AWS WAF rules, see Sending custom responses for Block actions.

> **Note**
>
> Responses that you customize using AWS WAF rules take precedence over any response specifications that you define in CloudFront custom error pages.

If you'd rather display a custom error message through CloudFront, possibly using the same formatting as the rest of your website, you can configure CloudFront to return to the viewer an object (for example, an HTML file) that contains your custom error message.

> **Note**
>
> CloudFront can't distinguish between an HTTP status code 403 that is returned by your origin and one that is returned by AWS WAF when a request is blocked. This means that you can't return different custom error pages based on the different causes of an HTTP status code 403.

For more information about CloudFront custom error pages, see Generating custom error responses in the *Amazon CloudFront Developer Guide*.

Custom error pages in multi-tenant distributions

For CloudFront multi-tenant distributions, you can configure custom error pages in the following ways:

- At the multi-tenant level - These settings apply to all tenant distributions that use the multi-tenant distribution template
- Through AWS WAF rules - Custom responses defined in protection packs (web ACLs) take precedence over both multi-tenant distribution and tenant-level custom error pages

Using AWS WAF with CloudFront for applications running on your own HTTP server

When you use AWS WAF with CloudFront, you can protect your applications running on any HTTP webserver, whether it's a webserver that's running in Amazon Elastic Compute Cloud (Amazon EC2) or a webserver that you manage privately. You can also configure CloudFront to require HTTPS between CloudFront and your own webserver, as well as between viewers and CloudFront.

Requiring HTTPS between CloudFront and your own webserver

To require HTTPS between CloudFront and your own webserver, you can use the CloudFront custom origin feature and configure the **Origin Protocol Policy** and the **Origin Domain Name** settings for specific origins. In your CloudFront configuration, you can specify the DNS name of the server along with the port and the protocol that you want CloudFront to use when fetching objects from your origin. You should also ensure that the SSL/TLS certificate on your custom origin server matches the origin domain name you've configured. When you use your own HTTP webserver outside of AWS, you must use a certificate that is signed by a trusted third-party certificate authority (CA), for example, Comodo, DigiCert, or Symantec. For more information about requiring HTTPS for communication between CloudFront and your own webserver, see the topic Requiring HTTPS for Communication Between CloudFront and Your Custom Origin in the *Amazon CloudFront Developer Guide*.

Requiring HTTPS between a viewer and CloudFront

To require HTTPS between viewers and CloudFront, you can change the **Viewer Protocol Policy** for one or more cache behaviors in your CloudFront distribution. For more information about

using HTTPS between viewers and CloudFront, see the topic Requiring HTTPS for Communication Between Viewers and CloudFront in the *Amazon CloudFront Developer Guide*. You can also bring your own SSL certificate so viewers can connect to your CloudFront distribution over HTTPS using your own domain name, for example *https://www.mysite.com*. For more information, see the topic Configuring Alternate Domain Names and HTTPS in the *Amazon CloudFront Developer Guide*.

For multi-tenant distributions, HTTP method configurations follow this hierarchy:

- Template-level settings define the baseline HTTP methods allowed for all tenant distributions
- Tenant distributions can override these settings to:
  - Allow fewer methods than the multi-tenant distribution (using AWS WAF rules to block additional methods)
  - Allow more methods if the multi-tenant distribution is configured to support them
- AWS WAF rules at both multi-tenant distribution and tenant levels can further restrict HTTP methods regardless of the CloudFront configuration

Choosing the HTTP methods that CloudFront responds to

When you create an Amazon CloudFront web distribution, you choose the HTTP methods that you want CloudFront to process and forward to your origin. You can choose from the following options:

- **`GET, HEAD`** – You can use CloudFront only to get objects from your origin or to get object headers.
- **`GET, HEAD, OPTIONS`** – You can use CloudFront only to get objects from your origin, get object headers, or retrieve a list of the options that your origin server supports.
- **`GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE`** – You can use CloudFront to get, add, update, and delete objects, and to get object headers. In addition, you can perform other `POST` operations such as submitting data from a web form.

You also can use AWS WAF byte match rule statements to allow or block requests based on the HTTP method, as described in String match rule statement. If you want to use a combination of methods that CloudFront supports, such as `GET` and `HEAD`, then you don't need to configure AWS WAF to block requests that use the other methods. If you want to allow a combination of methods that CloudFront doesn't support, such as `GET`, `HEAD`, and `POST`, you can configure CloudFront to respond to all methods, and then use AWS WAF to block requests that use other methods.

For more information about choosing the methods that CloudFront responds to, see Allowed HTTP Methods in the topic Values that You Specify When You Create or Update a Web Distribution in the *Amazon CloudFront Developer Guide*.

Allowed HTTP method configurations in multi-tenant distributions

For multi-tenant distributions, HTTP method configurations set at the multi-tenant distribution level apply to all tenant distributions by default. Tenant distributions can override these settings if needed.

- If you want to use a combination of methods that CloudFront supports, such as `GET` and `HEAD`, you don't need to configure AWS WAF to block requests that use other methods.
- If you want to allow a combination of methods that CloudFront doesn't support by default, such as `GET`, `HEAD`, and `POST`, you can configure CloudFront to respond to all methods, and then use AWS WAF to block requests that use other methods.

When implementing security headers in multi-tenant distributions, consider the following:

- Template-level security headers provide baseline protection across all tenant distributions
- Tenant distributions can:
  - Add new security headers not defined in the multi-tenant distribution
  - Modify values for tenant-specific headers
  - Cannot remove or override security headers set at the multi-tenant distribution level
- Consider using multi-tenant distribution-level headers for critical security controls that should apply to all tenants

Logging considerations

Both standard and multi-tenant distributions support AWS WAF logging, but there are important differences in how logs are structured and managed:

Logging comparison

Standard distributions	Multi-tenant distributions
One log configuration per distribution	Template and tenant-level logging options
Standard log fields	Additional tenant identifier fields

Standard distributions	Multi-tenant distributions
Single destination per distribution	Separate destinations possible for multi-tenant distribution and tenant logs

Additional resources

- To learn more about multi-tenant distributions, see Configure distributions in the *Amazon CloudFront Developer Guide*.
- To learn more about using AWS WAF with CloudFront, see Using AWS WAF protection in the *Amazon CloudFront Developer Guide*.
- To learn more about AWS WAF logs, see Log fields for protection pack (web ACL) traffic.

Security in your use of the AWS WAF service

This section explains how the shared responsibility model applies to AWS WAF.

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

> **Note**
>
> This section provides standard AWS security guidance for your use of the AWS WAF service and its AWS resources, such as AWS WAF protection packs (web ACLs) and rule groups. For information about protecting your AWS resources using AWS WAF, see the rest of the AWS WAF guide.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. The effectiveness of our security is regularly tested and verified by third-party auditors as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS WAF, see AWS Services in Scope by Compliance Program.

- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your organization's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using AWS WAF. The following topics show you how to configure AWS WAF to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS WAF resources.

Topics

- Protecting your data in AWS WAF
- Using IAM with AWS WAF
- Logging and monitoring in AWS WAF
- Validating compliance in AWS WAF
- Building for resilience in AWS WAF
- Infrastructure security in AWS WAF

Protecting your data in AWS WAF

The AWS shared responsibility model applies to data protection in AWS WAF. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see Data Privacy FAQ. For information about data protection in Europe, see the General Data Protection Regulation (GDPR) Center.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.

- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with AWS WAF or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

AWS WAF entities—such as protection packs (web ACLs), rule groups, and IP sets—are encrypted at rest, except in certain Regions where encryption is not available, including China (Beijing) and China (Ningxia). Unique encryption keys are used for each Region.

Deleting AWS WAF resources

You can delete the resources that you create in AWS WAF. See the guidance for each resource type in following sections.

- Deleting a protection pack (web ACL)
- Deleting a rule group
- Deleting an IP set
- Deleting a regex pattern set

Using IAM with AWS WAF

This section explains how to use IAM with AWS WAF.

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use AWS WAF resources. IAM is an AWS service that you can use with no additional charge.

Topics

- Audience
- Authenticating with identities
- Managing access using policies
- How AWS WAF works with IAM
- Identity-based policy examples for AWS WAF
- AWS managed policies for AWS WAF
- Troubleshooting AWS WAF identity and access
- Using service-linked roles for AWS WAF

Audience

How you use AWS Identity and Access Management (IAM) differs based on your role:

- **Service user** - request permissions from your administrator if you cannot access features (see Troubleshooting AWS WAF identity and access)
- **Service administrator** - determine user access and submit permission requests (see How AWS WAF works with IAM)
- **IAM administrator** - write policies to manage access (see Identity-based policy examples for AWS WAF)

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be authenticated as the AWS account root user, an IAM user, or by assuming an IAM role.

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you don't use the root user for everyday tasks. For tasks that require root user credentials, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users to use federation with an identity provider to access AWS services using temporary credentials.

A *federated identity* is a user from your enterprise directory, web identity provider, or Directory Service that accesses AWS services using credentials from an identity source. Federated identities assume roles that provide temporary credentials.

For centralized access management, we recommend AWS IAM Identity Center. For more information, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see Require human users to use federation with an identity provider to access AWS using temporary credentials in the *IAM User Guide*.

An *IAM group* specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity with specific permissions that provides temporary credentials. You can assume a role by switching from a user to an IAM role (console) or by calling an AWS CLI or AWS API operation. For more information, see Methods to assume a role in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you attach to an identity (user, group, or role). These policies control what actions identities can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples include IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. You must specify a principal in a resource-based policy.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Other policy types

AWS supports additional policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries** – Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – Set the maximum available permissions for resources in your accounts. For more information, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Advanced policies passed as a parameter when creating a temporary session for a role or federated user. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How AWS WAF works with IAM

This section explains how to use the features of IAM with AWS WAF.

> **Note**
>
> Introducing new actions in IAM policies that give you more granular control when invoking web ACL Association related operations for Application Load Balancer and AWS AppSync. See Actions that require additional permissions settings.

Before you use IAM to manage access to AWS WAF, learn what IAM features are available to use with AWS WAF.

IAM features you can use with AWS WAF

IAM feature	AWS WAF support
Identity-based policies	Yes

IAM feature	AWS WAF support
Resource-based policies	Yes
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	Yes
ACLs	No
ABAC (tags in policies)	Partial
Temporary credentials	Yes
Forward access sessions (FAS)	Yes
Service roles	Yes
Service-linked roles	Yes

To get a high-level view of how AWS WAF and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for AWS WAF

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

To view examples of AWS WAF identity-based policies, see Identity-based policy examples for AWS WAF.

Resource-based policies within AWS WAF

Supports resource-based policies: Yes

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

AWS WAF uses resource based policies to support the sharing of rule groups across accounts. You share a rule group that you own with another AWS account by providing the resource-based policy settings to the AWS WAF API call `PutPermissionPolicy` or to an equivalent CLI or SDK call. For additional information, including examples and links to documentation for the other available languages, see PutPermissionPolicy in the AWS WAF API Reference. This functionality isn't available through other means, such as the console or CloudFormation.

Policy actions for AWS WAF

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Include actions in a policy to grant permissions to perform the associated operation.

To see a list of AWS WAF actions and permissions for each, see Actions defined by AWS WAF V2 in the *Service Authorization Reference*.

Policy actions in AWS WAF use the following prefix before the action:

```
wafv2
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "wafv2:action1",
      "wafv2:action2"
         ]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions in AWS WAF that begin with `List`, include the following action:

```
"Action": "wafv2:List*"
```

To view examples of AWS WAF identity-based policies, see Identity-based policy examples for AWS WAF.

Actions that require additional permissions settings

Some actions require permissions that can't be completely described in Actions defined by AWS WAF V2 in the *Service Authorization Reference*. This section provides additional permissions information.

Topics

- Permissions for AssociateWebACL
- Permissions for DisassociateWebACL
- Permissions for GetWebACLForResource
- Permissions for ListResourcesForWebACL

Permissions for `AssociateWebACL`

This section lists the permissions required to associate a protection pack (web ACL) to a resource using the AWS WAF action `AssociateWebACL`.

For Amazon CloudFront distributions, instead of this action, use the CloudFront action `UpdateDistribution`. For information, see UpdateDistribution in the *Amazon CloudFront API Reference*.

Amazon API Gateway REST API

Requires permission to call API Gateway `SetWebACL` on the REST API resource type and to call AWS WAF `AssociateWebACL` on a protection pack (web ACL).

```
{
    "Sid": "AssociateWebACL1",
    "Effect": "Allow",
    "Action": [
        "wafv2:AssociateWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
},
{
    "Sid": "AssociateWebACL2",
    "Effect": "Allow",
    "Action": [
        "apigateway:SetWebACL"
    ],
    "Resource": [
        "arn:aws:apigateway:*::/restapis/*/stages/*"
    ]
}
```

Application Load Balancer

Old Permission Setting

Requires permission to call `elasticloadbalancing:SetWebACL` action on the Application Load Balancer resource type and to call AWS WAF `AssociateWebACL` on a protection pack (web ACL).

```
{
                "Sid": "AssociateWebACL1",
                "Effect": "Allow",
                "Action": [
                   "wafv2:AssociateWebACL"
                ],
                "Resource": [
                   "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
                ]
            },
            {
                "Sid": "AssociateWebACL2",
```

```
                "Effect": "Allow",
                "Action": [
                   "elasticloadbalancing:SetWebACL"
                ],
                "Resource": [
                   "arn:aws:elasticloadbalancing:*:account-id:loadbalancer/app/*/*"
                ]
             }
```

New Permission Setting

Requires permission to call `elasticloadbalancing:CreateWebACLAssociation` action on the Application Load Balancer resource type and to call AWS WAF `AssociateWebACL` on a protection pack (web ACL).

```
{
   "Sid": "AssociateWebACL1",
   "Effect": "Allow",
   "Action": [
      "wafv2:AssociateWebACL"
   ],
   "Resource": [
      "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
   ]
},
{
   "Sid": "AssociateWebACL2",
   "Effect": "Allow",
   "Action": [
      "elasticloadbalancing:CreateWebACLAssociation"
   ],
   "Resource": [
         "arn:aws:elasticloadbalancing:*:account-id:loadbalancer/app/*/*"
   ]
}
```

AWS AppSync GraphQL API

Old Permission Setting

Requires permission to call AWS AppSync `SetWebACL` on the GraphQL API resource type and to call AWS WAF `AssociateWebACL` on a protection pack (web ACL).

```
{
    "Sid": "AssociateWebACL1",
    "Effect": "Allow",
    "Action": [
        "wafv2:AssociateWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
},
{
    "Sid": "AssociateWebACL2",
    "Effect": "Allow",
    "Action": [
        "appsync:SetWebACL"
    ],
    "Resource": [
        "arn:aws:appsync:*:account-id:apis/*"
    ]
}
```

New Permission Setting

Requires permission to call `appsync:AssociateWebACL` action on the GraphQL API resource type and to call AWS WAF `AssociateWebACL` on a protection pack (web ACL).

```
{
    "Sid": "AssociateWebACL1",
    "Effect": "Allow",
    "Action": [
        "wafv2:AssociateWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
},
{
    "Sid": "AssociateWebACL2",
    "Effect": "Allow",
    "Action": [
        "appsync:AssociateWebACL"
    ],
    "Resource": [
```

```
        "arn:aws:appsync:*:account-id:apis/*"
    ]
}
```

Amazon Cognito user pool

Requires permission to call the Amazon Cognito `AssociateWebACL` action on the user pool resource type and to call AWS WAF `AssociateWebACL` on a protection pack (web ACL).

```
{
    "Sid": "AssociateWebACL1",
    "Effect": "Allow",
    "Action": [
        "wafv2:AssociateWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
},
{
    "Sid": "AssociateWebACL2",
    "Effect": "Allow",
    "Action": [
        "cognito-idp:AssociateWebACL"
    ],
    "Resource": [
        "arn:aws:cognito-idp:*:account-id:userpool/*"
    ]
}
```

AWS App Runner service

Requires permission to call the App Runner `AssociateWebACL` action on the App Runner service resource type and to call AWS WAF `AssociateWebACL` on a web ACL.

```
{
    "Sid": "AssociateWebACL1",
    "Effect": "Allow",
    "Action": [
        "wafv2:AssociateWebACL"
    ],
    "Resource": [
```

```
            "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
        ]
    },
    {
        "Sid": "AssociateWebACL2",
        "Effect": "Allow",
        "Action": [
            "apprunner:AssociateWebAcl"
        ],
        "Resource": [
            "arn:aws:apprunner:*:account-id:service/*/*"
        ]
    }
```

Amazon Bedrock AgentCore Gateway

Requires permission to call the `bedrock-agentcore:GatewayAssociateWebACL` action on the Bedrock AgentCore Gateway resource type and to call AWS WAF `AssociateWebACL` for a web ACL.

```
{
  "Sid": "AssociateWebACL1",
  "Effect": "Allow",
  "Action": [
    "wafv2:AssociateWebACL"
  ],
  "Resource": [
    "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
  ]
},
{
  "Sid": "AssociateWebACL2",
  "Effect": "Allow",
  "Action": [
    "bedrock-agentcore:GatewayAssociateWebACL"
  ],
  "Resource": [
    "arn:aws:bedrock-agentcore:*:account-id:gateway/*"
  ]
}
```

AWS Verified Access instance

Requires permission to call the `ec2:AssociateVerifiedAccessInstanceWebAcl` action on the Verified Access instance resource type and to call AWS WAF `AssociateWebACL` on a web ACL.

```
{
    "Sid": "AssociateWebACL1",
    "Effect": "Allow",
    "Action": [
        "wafv2:AssociateWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
},
{
    "Sid": "AssociateWebACL2",
    "Effect": "Allow",
    "Action": [
        "ec2:AssociateVerifiedAccessInstanceWebAcl"
    ],
    "Resource": [
        "arn:aws:ec2:*:account-id:verified-access-instance/*"
    ]
}
```

Permissions for `DisassociateWebACL`

This section lists the permissions required to disassociate a protection pack (web ACL) from a resource using the AWS WAF action `DisassociateWebACL`.

For Amazon CloudFront distributions, instead of this action, use the CloudFront action `UpdateDistribution` with an empty protection pack (web ACL) ID. For information, see UpdateDistribution in the *Amazon CloudFront API Reference*.

Amazon API Gateway REST API

Requires permission to call API Gateway `SetWebACL` on the REST API resource type. Does not require permission to call AWS WAF `DisassociateWebACL`.

```
{
    "Sid": "DisassociateWebACL",
    "Effect": "Allow",
    "Action": [
        "apigateway:SetWebACL"
```

```
    ],
    "Resource": [
        "arn:aws:apigateway:*::/restapis/*/stages/*"
    ]
}
```

Application Load Balancer

Old Permission Setting

Requires permission to call the `elasticloadbalancing:SetWebACL` action on the Application Load Balancer resource type. Does not require permission to call AWS WAF `DisassociateWebACL`.

```
{
    "Sid": "DisassociateWebACL",
    "Effect": "Allow",
    "Action": [
        "elasticloadbalancing:SetWebACL"
    ],
    "Resource": [
        "arn:aws:elasticloadbalancing:*:account-id:loadbalancer/app/*/*"
    ]
}
```

New Permission Setting

Requires permission to call the `elasticloadbalancing:DeleteWebACLAssociation` action on the Application Load Balancer resource type. Also requires permission to call AWS WAF `DisassociateWebACL`.

```
{
      "Sid": "DisassociateWebACL",
      "Effect": "Allow",
      "Action": "wafv2:DisassociateWebACL",
      "Resource": "*"
    },
    {
      "Sid": "DisassociateWebACL2",
      "Effect": "Allow",
      "Action": [
         "elasticloadbalancing:DeleteWebACLAssociation"
      ],
      "Resource": [
```

```
        "arn:aws:elasticloadbalancing:*:account-id:loadbalancer/app/*/*"
      ]
   }
}
```

AWS AppSync GraphQL API

Old Permission Setting

Requires permission to call AWS AppSync `SetWebACL` on the GraphQL API resource type. Does not require permission to call AWS WAF `DisassociateWebACL`.

```
{
    "Sid": "DisassociateWebACL",
    "Effect": "Allow",
    "Action": [
        "appsync:SetWebACL"
    ],
    "Resource": [
        "arn:aws:appsync:*:account-id:apis/*"
    ]
}
```

New Permission Setting

Requires permission to call the `appsync:DisassociateWebACL` action on the GraphQL API resource type. Also requires permission to call AWS WAF `DisassociateWebACL`.

```
{
    "Sid": "DisassociateWebACL1",
    "Effect": "Allow",
    "Action": "wafv2:DisassociateWebACL",
    "Resource": "*"
},
{
    "Sid": "DisassociateWebACL2",
    "Effect": "Allow",
    "Action": [
        "appsync:DisassociateWebACL"
    ],
    "Resource": [
        "arn:aws:appsync:*:account-id:apis/*"
    ]
```

```
}
```

Amazon Cognito user pool

Requires permission to call the Amazon Cognito `DisassociateWebACL` action on the user pool resource type and to call AWS WAF `DisassociateWebACL`.

```
{
    "Sid": "DisassociateWebACL1",
    "Effect": "Allow",
    "Action": "wafv2:DisassociateWebACL",
    "Resource": "*"
},
{
    "Sid": "DisassociateWebACL2",
    "Effect": "Allow",
    "Action": [
        "cognito-idp:DisassociateWebACL"
    ],
    "Resource": [
        "arn:aws:cognito-idp:*:account-id:userpool/*"
    ]
}
```

AWS App Runner service

Requires permission to call the App Runner `DisassociateWebACL` action on the App Runner service resource type and to call AWS WAF `DisassociateWebACL`.

```
{
    "Sid": "DisassociateWebACL1",
    "Effect": "Allow",
    "Action": "wafv2:DisassociateWebACL",
    "Resource": "*"
},
{
    "Sid": "DisassociateWebACL2",
    "Effect": "Allow",
    "Action": [
        "apprunner:DisassociateWebAcl"
    ],
    "Resource": [
        "arn:aws:apprunner:*:account-id:service/*/*"
```

```
    ]
}
```

Amazon Bedrock AgentCore Gateway

Requires permission to call the `bedrock-agentcore:GatewayDisassociateWebACL` action on the Bedrock AgentCore Gateway resource type and to call AWS WAF `DisassociateWebACL`.

```
{
  "Sid": "DisassociateWebACL",
  "Effect": "Allow",
  "Action": [
    "wafv2:DisassociateWebACL"
  ],
  "Resource": "*"
},
{
  "Sid": "DisassociateWebACL2",
  "Effect": "Allow",
  "Action": [
    "bedrock-agentcore:GatewayDisassociateWebACL"
  ],
  "Resource": [
    "arn:aws:bedrock-agentcore:*:account-id:gateway/*"
  ]
}
```

AWS Verified Access instance

Requires permission to call the `ec2:DisassociateVerifiedAccessInstanceWebAcl` action on the Verified Access instance resource type and to call AWS WAF `DisassociateWebACL`.

```
{
    "Sid": "DisassociateWebACL1",
    "Effect": "Allow",
    "Action": "wafv2:DisassociateWebACL",
    "Resource": "*"
},
{
    "Sid": "DisassociateWebACL2",
    "Effect": "Allow",
    "Action": [
        "ec2:DisassociateVerifiedAccessInstanceWebAcl"
```

```
    ],
    "Resource": [
        "arn:aws:ec2:*:account-id:verified-access-instance/*"
    ]
}
```

Permissions for `GetWebACLForResource`

This section lists the permissions required to get the protection pack (web ACL) for a protected resource using the AWS WAF action `GetWebACLForResource`.

For Amazon CloudFront distributions, instead of this action, use the CloudFront action `GetDistributionConfig`. For information, see GetDistributionConfig in the *Amazon CloudFront API Reference*.

> **Note**
>
> `GetWebACLForResource` requires the permission to call `GetWebACL`. In this context, AWS WAF uses `GetWebACL` only to verify that your account has the permission it needs to access the protection pack (web ACL) that `GetWebACLForResource` returns. When you call `GetWebACLForResource`, you might get an error indicating that your account is not authorized to perform `wafv2:GetWebACL` on the resource. AWS WAF doesn't add this type of error to the AWS CloudTrail event history.

Amazon API Gateway REST API

Require permission to call AWS WAF `GetWebACLForResource` and `GetWebACL` for a protection pack (web ACL).

```
{
    "Sid": "GetWebACLForResource",
    "Effect": "Allow",
    "Action": [
        "wafv2:GetWebACLForResource",
        "wafv2:GetWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
}
```

Application Load Balancer

Old Permission Setting

Require permission to call AWS WAF `GetWebACLForResource` and `GetWebACL` for a protection pack (web ACL).

```
{
    "Sid": "GetWebACLForResource",
    "Effect": "Allow",
    "Action": [
        "wafv2:GetWebACLForResource",
        "wafv2:GetWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
}
```

New Permission Setting

Require permission to call AWS WAF `GetWebACLForResource` and `GetWebACL` for a protection pack (web ACL). Also require permission to call `elasticloadbalancing:GetLoadBalancerWebACL`.

```
{
    "Sid": "GetWebACLForResource1",
    "Effect": "Allow",
    "Action": [
        "wafv2:GetWebACLForResource",
        "wafv2:GetWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
},
{
   "Sid": "GetWebACLForResource2",
    "Effect": "Allow",
    "Action": [
        "elasticloadbalancing:GetLoadBalancerWebACL"
    ],
    "Resource": [
```

```
        "arn:aws:elasticloadbalancing:*:account-id:loadbalancer/app/*/*"
    ]
}
```

AWS AppSync GraphQL API

Old Permission Setting

Require permission to call AWS WAF `GetWebACLForResource` and `GetWebACL` for a protection pack (web ACL).

```
{
    "Sid": "GetWebACLForResource",
    "Effect": "Allow",
    "Action": [
        "wafv2:GetWebACLForResource",
        "wafv2:GetWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
}
```

New Permission Setting

Require permission to call AWS WAF `GetWebACLForResource` and `GetWebACL` for a protection pack (web ACL). Also require permission to call `appsync:GetWebACLForResource`.

```
{
    "Sid": "GetWebACLForResource1",
    "Effect": "Allow",
    "Action": [
        "wafv2:GetWebACLForResource",
        "wafv2:GetWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
},
{
   "Sid": "GetWebACLForResource2",
    "Effect": "Allow",
    "Action": [
```

```
        "appsync:GetWebACLForResource"
    ],
    "Resource": [
        "arn:aws:appsync:*:account-id:apis/*"
    ]
}
```

Amazon Cognito user pool

Requires permission to call the Amazon Cognito `GetWebACLForResource` action on the user pool resource type and to call AWS WAF `GetWebACLForResource` and `GetWebACL`.

```
{
    "Sid": "GetWebACLForResource1",
    "Effect": "Allow",
    "Action": [
        "wafv2:GetWebACLForResource",
        "wafv2:GetWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
},
{
    "Sid": "GetWebACLForResource2",
    "Effect": "Allow",
    "Action": [
        "cognito-idp:GetWebACLForResource"
    ],
    "Resource": [
        "arn:aws:cognito-idp:*:account-id:userpool/*"
    ]
}
```

AWS App Runner service

Requires permission to call the App Runner `DescribeWebAclForService` action on the App Runner service resource type and to call AWS WAF `GetWebACLForResource` and `GetWebACL`.

```
{
    "Sid": "GetWebACLForResource1",
    "Effect": "Allow",
```

```
    "Action": [
        "wafv2:GetWebACLForResource",
        "wafv2:GetWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
},
{
    "Sid": "GetWebACLForResource2",
    "Effect": "Allow",
    "Action": [
        "apprunner:DescribeWebAclForService"
    ],
    "Resource": [
        "arn:aws:apprunner:*:account-id:service/*/*"
    ]
}
```

Amazon Bedrock AgentCore Gateway

Requires permission to call the `bedrock-agentcore:GatewayGetWebACLForResource` action on the Bedrock AgentCore Gateway resource type and to call AWS WAF `GetWebACLForResource` and `GetWebACL` for a web ACL.

```
{
  "Sid": "GetWebACLForResource1",
  "Effect": "Allow",
  "Action": [
    "wafv2:GetWebACLForResource",
    "wafv2:GetWebACL"
  ],
  "Resource": [
    "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
  ]
},
{
  "Sid": "GetWebACLForResource2",
  "Effect": "Allow",
  "Action": [
    "bedrock-agentcore:GatewayGetWebACLForResource"
  ],
  "Resource": [
```

```
    "arn:aws:bedrock-agentcore:*:account-id:gateway/*"
  ]
}
```

AWS Verified Access instance

Requires permission to call the `ec2:GetVerifiedAccessInstanceWebAcl` action on the Verified Access instance resource type and to call AWS WAF `GetWebACLForResource` and `GetWebACL`.

```
{
    "Sid": "GetWebACLForResource1",
    "Effect": "Allow",
    "Action": [
        "wafv2:GetWebACLForResource",
        "wafv2:GetWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
},
{
    "Sid": "GetWebACLForResource2",
    "Effect": "Allow",
    "Action": [
        "ec2:GetVerifiedAccessInstanceWebAcl"
    ],
    "Resource": [
        "arn:aws:ec2:*:account-id:verified-access-instance/*"
    ]
}
```

Permissions for `ListResourcesForWebACL`

This section lists the permissions required to retrieve the list of protected resources for a protection pack (web ACL) using the AWS WAF action `ListResourcesForWebACL`.

For Amazon CloudFront distributions, instead of this action, use the CloudFront action `ListDistributionsByWebACLId`. For information, see ListDistributionsByWebACLId in the *Amazon CloudFront API Reference*.

Amazon API Gateway REST API

Require permission to call AWS WAF `ListResourcesForWebACL` for a web ACL.

```
{
    "Sid": "ListResourcesForWebACL",
    "Effect": "Allow",
    "Action": [
        "wafv2:ListResourcesForWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
}
```

Application Load Balancer

Old Permission Setting

Require permission to call AWS WAF `ListResourcesForWebACL` for a protection pack (web ACL).

```
{
    "Sid": "ListWebACLForResource",
    "Effect": "Allow",
    "Action": [
        "wafv2:GetWebACLForResource"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
}
```

New Permission Setting

Require permission to call AWS WAF `ListResourcesForWebACL` for a protection pack (web ACL). Also require permission to call `elasticloadbalancing:DescribeWebACLAssociation`.

```
{
    "Sid": "ListResourcesForWebACL",
    "Effect": "Allow",
    "Action": [
        "wafv2:ListResourcesForWebACL",
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
```

```
    ]
},
{
   "Sid": "ListResourcesForWebACL2",
    "Effect": "Allow",
    "Action": [
        "elasticloadbalancing:DescribeWebACLAssociation"
    ],
    "Resource": "*"
}
```

AWS AppSync GraphQL API

Old Permission Setting

Require permission to call AWS WAF `ListResourcesForWebACL` for a protection pack (web ACL).

```
{
    "Sid": "ListResourcesForWebACL",
    "Effect": "Allow",
    "Action": [
        "wafv2:ListResourcesForWebACL",
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
}
```

New Permission Setting

Require permission to call AWS WAF `ListResourcesForWebACL` for a protection pack (web ACL). Also require permission to call `appsync:ListResourcesForWebACL`.

```
{
    "Sid": "ListResourcesForWebACL",
    "Effect": "Allow",
    "Action": [
        "wafv2:ListResourcesForWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
},
```

```
{
    "Sid": "ListResourcesForWebACL2",
    "Effect": "Allow",
    "Action": [
        "appsync:ListResourcesForWebACL"
    ],
    "Resource": "*"
}
```

Amazon Cognito user pool

Requires permission to call the Amazon Cognito `ListResourcesForWebACL` action on the user pool resource type and to call AWS WAF `ListResourcesForWebACL`.

```
{
    "Sid": "ListResourcesForWebACL1",
    "Effect": "Allow",
    "Action": [
        "wafv2:ListResourcesForWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
},
{
    "Sid": "ListResourcesForWebACL2",
    "Effect": "Allow",
    "Action": [
        "cognito-idp:ListResourcesForWebACL"
    ],
    "Resource": [
        "arn:aws:cognito-idp:*:account-id:userpool/*"
    ]
}
```

AWS App Runner service

Requires permission to call the App Runner `ListAssociatedServicesForWebAcl` action on the App Runner service resource type and to call AWS WAF `ListResourcesForWebACL`.

```
{
    "Sid": "ListResourcesForWebACL1",
    "Effect": "Allow",
```

```
    "Action": [
        "wafv2:ListResourcesForWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
},
{
    "Sid": "ListResourcesForWebACL2",
    "Effect": "Allow",
    "Action": [
        "apprunner:ListAssociatedServicesForWebAcl"
    ],
    "Resource": [
        "arn:aws:apprunner:*:account-id:service/*/*"
    ]
}
```

Amazon Bedrock AgentCore Gateway

Requires permission to call the `bedrock-agentcore:GatewayListResourcesForWebACL` action and to call AWS WAF `ListResourcesForWebACL` for a web ACL.

```
{
  "Sid": "ListResourcesForWebACL",
  "Effect": "Allow",
  "Action": [
    "wafv2:ListResourcesForWebACL"
  ],
  "Resource": [
    "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
  ]
},
{
  "Sid": "ListResourcesForWebACL2",
  "Effect": "Allow",
  "Action": [
    "bedrock-agentcore:GatewayListResourcesForWebACL"
  ],
  "Resource": "*"
}
```

AWS Verified Access instance

Requires permission to call the `ec2:DescribeVerifiedAccessInstanceWebAclAssociations` action on the Verified Access instance resource type and to call AWS WAF `ListResourcesForWebACL`.

```
{
    "Sid": "ListResourcesForWebACL1",
    "Effect": "Allow",
    "Action": [
        "wafv2:ListResourcesForWebACL"
    ],
    "Resource": [
        "arn:aws:wafv2:region:account-id:regional/webacl/*/*"
    ]
},
{
    "Sid": "ListResourcesForWebACL2",
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeVerifiedAccessInstanceWebAclAssociations"
    ],
    "Resource": [
        "arn:aws:ec2:*:account-id:verified-access-instance/*"
    ]
}
```

Policy resources for AWS WAF

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. As a best practice, specify a resource using its Amazon Resource Name (ARN). For actions that don't support resource-level permissions, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see the list of AWS WAF resource types and their ARNs, see Resources defined by AWS WAF V2 in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each

resource, see Actions defined by AWS WAF V2. To allow or deny access to a subset of AWS WAF resources, include the ARN of the resource in the `resource` element of your policy.

The ARNs of AWS WAF `wafv2` resources have the following format:

```
arn:partition:wafv2:region:account-id:scope/resource-type/resource-name/resource-id
```

For general information about ARN specifications, see Amazon Resource Names (ARNs) in the Amazon Web Services General Reference.

The following lists requirements that are specific to the ARNs of `wafv2` resources:

- *region*: For AWS WAF resources that you use to protect Amazon CloudFront distributions, set this to `us-east-1`. Otherwise, set this to the Region you're using with your protected regional resources.
- *scope*: Set the scope to `global` for use with an Amazon CloudFront distribution or `regional` for use with any of the regional resources that AWS WAF supports. The regional resources are an Amazon API Gateway REST API, an Application Load Balancer, an AWS AppSync GraphQL API, an Amazon Cognito user pool, an AWS App Runner service, an Amazon Bedrock AgentCore Gateway, and an AWS Verified Access instance.
- *resource-type*: Specify one of the following values: `webacl`, `rulegroup`, `ipset`, `regexpatternset`, or `managedruleset`.
- *resource-name*: Specify the name that you gave the AWS WAF resource, or specify a wildcard (*) to indicate all resources that satisfy the other specifications in the ARN. You must either specify the resource name and resource ID or specify a wildcard for both.
- *resource-id*: Specify the ID of the AWS WAF resource, or specify a wildcard (*) to indicate all resources that satisfy the other specifications in the ARN. You must either specify the resource name and resource ID or specify a wildcard for both.

For example, the following ARN specifies all protection packs (web ACLs) with regional scope for the account `111122223333` in Region `us-west-1`:

```
arn:aws:wafv2:us-west-1:111122223333:regional/webacl/*/*
```

The following ARN specifies the rule group named `MyIPManagementRuleGroup` with global scope for the account `111122223333` in Region `us-east-1`:

```
arn:aws:wafv2:us-east-1:111122223333:global/rulegroup/MyIPManagementRuleGroup/1111aaaa-bbbb-cccc-dddd-example-id
```

To view examples of AWS WAF identity-based policies, see Identity-based policy examples for AWS WAF.

Policy condition keys for AWS WAF

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element specifies when statements execute based on defined criteria. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

In addition, AWS WAF supports the following condition keys that you can use to provide fine-grained filtering for your IAM policies:

- **wafv2:LogDestinationResource**

  This condition key takes an Amazon Resource Name (ARN) specification for the logging destination. This is the ARN that you provide for the logging destination when you use the REST API call `PutLoggingConfiguration`.

  You can explicitly specify an ARN and you can specify filtering for the ARN. The following example specifies filtering for Amazon S3 bucket ARNs that have a specific location and prefix.

  ```
  "Condition": { "ArnLike": { "wafv2:LogDestinationResource": "arn:aws:s3:::aws-waf-logs-suffix/custom-prefix/*" } }
  ```

- **wafv2:LogScope**

  This condition key defines the source of the logging configuration in a string. Currently, this is always set to the default of `Customer`, which indicates that the logging destination is owned and managed by you.

To see a list of AWS WAF condition keys, see Condition keys for AWS WAF V2 in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by AWS WAF V2.

To view examples of AWS WAF identity-based policies, see Identity-based policy examples for AWS WAF.

ACLs in AWS WAF

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ABAC with AWS WAF

Supports ABAC (tags in policies): Partial

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes called tags. You can attach tags to IAM entities and AWS resources, then design ABAC policies to allow operations when the principal's tag matches the tag on the resource.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see Define permissions with ABAC authorization in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with AWS WAF

Supports temporary credentials: Yes

Temporary credentials provide short-term access to AWS resources and are automatically created when you use federation or switch roles. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM and AWS services that work with IAM in the *IAM User Guide*.

Forward access sessions for service AWS WAF

Supports forward access sessions (FAS): Yes

Forward access sessions (FAS) use the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. For policy details when making FAS requests, see Forward access sessions.

Service roles for AWS WAF

Supports service roles: Yes

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.

> **Warning**
>
> Changing the permissions for a service role might break AWS WAF functionality. Edit service roles only when AWS WAF provides guidance to do so.

Service-linked roles for AWS WAF

Supports service-linked roles: Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing AWS WAF service-linked roles, see Using service-linked roles for AWS WAF.

Identity-based policy examples for AWS WAF

This section provides identity-based policy examples for AWS WAF.

By default, users and roles don't have permission to create or modify AWS WAF resources. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Create IAM policies (console) in the *IAM User Guide*.

For details about actions and resource types defined by AWS WAF, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for AWS WAF V2 in the *Service Authorization Reference*.

Topics

- Policy best practices
- Using the AWS WAF console
- Allow users to view their own permissions
- Grant read-only access to AWS WAF, CloudFront, and CloudWatch
- Grant full access to AWS WAF, CloudFront, and CloudWatch
- Grant access to a single AWS account
- Grant access to a single protection pack (web ACL)
- Grant CLI access to a protection pack (web ACL) and rule group

Policy best practices

Identity-based policies determine whether someone can create, access, or delete AWS WAF resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to

specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.

- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the AWS WAF console

To access the AWS WAF console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the AWS WAF resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that users and roles can use the AWS WAF console, also attach at least the AWS WAF `AWSWAFConsoleReadOnlyAccess` AWS managed policy to the entities. For information about this managed policy, see AWS managed policy: AWSWAFConsoleReadOnlyAccess. For more information about attaching a managed policy to a user, see Adding permissions to a user in the *IAM User Guide*.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws-eusc:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Grant read-only access to AWS WAF, CloudFront, and CloudWatch

The following policy grants users read-only access to AWS WAF resources, to Amazon CloudFront web distributions, and to Amazon CloudWatch metrics. It's useful for users who need permission to view the settings in AWS WAF conditions, rules, and protection packs (web ACLs) to see which distribution is associated with a protection pack (web ACL), and to monitor metrics and a sample of requests in CloudWatch. These users can't create, update, or delete AWS WAF resources.

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Action": [
               "wafv2:Get*",
               "wafv2:List*",
               "cloudfront:GetDistribution",
               "cloudfront:GetDistributionConfig",
               "cloudfront:ListDistributions",
               "cloudfront:ListDistributionsByWebACLId",
               "cloudfront:ListDistributionTenantsByCustomization",
               "cloudfront:ListDistributionTenants",
               "cloudfront:GetDistributionTenant",
               "cloudwatch:GetMetricData",
               "cloudwatch:ListMetrics",
               "cloudwatch:GetMetricStatistics",
               "ec2:DescribeRegions",
               "pricingplanmanager:GetSubscription",
               "pricingplanmanager:ListSubscriptions",
               "route53:ListHostedZones",
               "route53:GetHostedZone"
           ],
           "Effect": "Allow",
           "Resource": "*"
       }
   ]
}
```

Grant full access to AWS WAF, CloudFront, and CloudWatch

The following policy lets users perform any AWS WAF operation, perform any operation on CloudFront web distributions, and monitor metrics and a sample of requests in CloudWatch. It's useful for users who are AWS WAF administrators.

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Action": [
               "wafv2:*",
```

```
                "cloudfront:CreateDistribution",
                "cloudfront:ListDistributions",
                "cloudfront:ListDistributionsByWebACLId",
                "cloudfront:UpdateDistribution",
                "cloudfront:GetDistributionConfig",
                "cloudfront:GetDistribution",
                "cloudfront:DisassociateDistributionTenantWebACL",
                "cloudfront:DisassociateDistributionWebACL",
                "cloudfront:AssociateDistributionTenantWebACL",
                "cloudfront:AssociateDistributionWebACL",
                "cloudfront:ListDistributionTenantsByCustomization",
                "cloudfront:ListDistributionTenants",
                "cloudfront:DeleteDistribution",
                "cloudfront:GetDistributionTenant",
                "cloudfront:DeleteDistributionTenant",
                "cloudwatch:GetMetricData",
                "cloudwatch:ListMetrics",
                "cloudwatch:GetMetricStatistics",
                "ec2:DescribeRegions",
                "pricingplanmanager:GetSubscription",
                "pricingplanmanager:ListSubscriptions",
                "pricingplanmanager:UpdateSubscription",
                "pricingplanmanager:CancelSubscription",
                "pricingplanmanager:CancelSubscriptionChange",
                "pricingplanmanager:AssociateResourcesToSubscription",
                "pricingplanmanager:DisassociateResourcesFromSubscription",
                "route53:ListHostedZones",
                "route53:GetHostedZone"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

We strongly recommend that you configure multi-factor authentication (MFA) for users who have administrative permissions. For more information, see Using Multi-Factor Authentication (MFA) Devices with AWS in the *IAM User Guide*.

Grant access to a single AWS account

This policy grants the following permissions to the account 444455556666:

- Full access to all AWS WAF operations and resources.
- Read and update access to all CloudFront distributions, which allows you to associate protection packs (web ACLs) and CloudFront distributions.
- Read access to all CloudWatch metrics and metric statistics, so that you can view CloudWatch data and a sample of requests in the AWS WAF console.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "wafv2:*"
         ],
         "Resource": [
            "arn:aws-eusc:wafv2:eusc-de-east-1:444455556666:*"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "cloudfront:GetDistribution",
            "cloudfront:GetDistributionConfig",
            "cloudfront:ListDistributions",
            "cloudfront:ListDistributionsByWebACLId",
            "cloudfront:UpdateDistribution",
            "cloudwatch:ListMetrics",
            "cloudwatch:GetMetricStatistics",
            "ec2:DescribeRegions"
         ],
         "Resource": [
            "*"
         ]
      }
   ]
}
```

Grant access to a single protection pack (web ACL)

The following policy lets users perform any AWS WAF operation through the console on a specific protection pack (web ACL) in the account 444455556666.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "wafv2:*"
            ],
            "Resource": [
                "arn:aws-eusc:wafv2:eusc-de-east-1:444455556666:regional/webacl/
test123/112233d7c-86b2-458b-af83-51c51example"
            ]
        },
        {
            "Sid": "consoleAccess",
            "Effect": "Allow",
            "Action": [
                "wafv2:ListWebACLs",
                "ec2:DescribeRegions"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Grant CLI access to a protection pack (web ACL) and rule group

The following policy lets users perform any AWS WAF operation through the CLI on a specific protection pack (web ACL) and a specific rule group in the account 444455556666.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "wafv2:*"
         ],
         "Resource": [
        "arn:aws-eusc:wafv2:eusc-de-east-1:444455556666:regional/webacl/
test123/112233d7c-86b2-458b-af83-51c51example",
        "arn:aws-eusc:wafv2:eusc-de-east-1:444455556666:regional/rulegroup/
test123rulegroup/555555555-6666-1234-abcd-00d11example"
         ]
      }
   ]
}
```

The following policy lets users perform any AWS WAF operation through the console on a specific protection pack (web ACL) in the account 444455556666.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "wafv2:*"
            ],
            "Resource": [
                "arn:aws-eusc:wafv2:eusc-de-east-1:444455556666:regional/webacl/
test123/112233d7c-86b2-458b-af83-51c51example"
            ]
        },
        {
            "Sid": "consoleAccess",
```

```
            "Effect": "Allow",
            "Action": [
                "wafv2:ListWebACLs",
                "ec2:DescribeRegions"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

AWS managed policies for AWS WAF

This section explains how to use AWS managed policies for AWS WAF.

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

AWS managed policy: AWSWAFReadOnlyAccess

This policy grants read-only permissions that allow users to access AWS WAF resources and resources for integrated services, such as Amazon CloudFront, Amazon API Gateway, Application Load Balancer, AWS AppSync, Amazon Cognito, AWS App Runner, AWS Amplify, Amazon CloudWatch, Amazon Bedrock AgentCore Gateway, and AWS Verified Access. You can attach this policy to your IAM identities. AWS WAF also attaches this policy to a service role that allows AWS WAF to perform actions on your behalf.

For details about this policy, see AWSWAFReadOnlyAccess in the IAM console.

AWS managed policy: AWSWAFFullAccess

This policy grants full access to AWS WAF resources and resources for integrated services, such as Amazon CloudFront, Amazon API Gateway, Application Load Balancer, AWS AppSync, Amazon Cognito, AWS App Runner, AWS Amplify, Amazon CloudWatch, Amazon Bedrock AgentCore Gateway, and AWS Verified Access. You can attach this policy to your IAM identities. AWS WAF also attaches this policy to a service role that allows AWS WAF to perform actions on your behalf.

For details about this policy, see AWSWAFFullAccess in the IAM console.

AWS managed policy: AWSWAFConsoleReadOnlyAccess

This policy grants read-only permissions to the AWS WAF console, which includes resources for AWS WAF and for integrated services, such as Amazon CloudFront, Amazon API Gateway, Application Load Balancer, AWS AppSync, Amazon Cognito, AWS App Runner, AWS Amplify, Amazon CloudWatch, Amazon Bedrock AgentCore Gateway, and AWS Verified Access. You can attach this policy to your IAM identities.

For details about this policy, see AWSWAFConsoleReadOnlyAccess in the IAM console.

AWS managed policy: AWSWAFConsoleFullAccess

This policy grants full access to the AWS WAF console, which includes resources for AWS WAF and for integrated services, such as Amazon CloudFront, Amazon API Gateway, Application Load Balancer, AWS AppSync, Amazon Cognito, AWS App Runner, AWS Amplify, Amazon CloudWatch, Amazon Bedrock AgentCore Gateway, and AWS Verified Access. You can attach this policy to your IAM identities. AWS WAF also attaches this policy to a service role that allows AWS WAF to perform actions on your behalf.

For details about this policy, see AWSWAFConsoleFullAccess in the IAM console.

AWS managed policy: WAFV2LoggingServiceRolePolicy

This policy allows AWS WAF to write logs to Amazon Data Firehose. This policy is used only if you enable logging in AWS WAF. This policy is attached to the service-linked role `AWSServiceRoleForWAFV2Logging`. For more information about the service-linked role, see Using service-linked roles for AWS WAF.

For details about this policy, see WAFV2LoggingServiceRolePolicy in the IAM console.

AWS WAF updates to AWS managed policies

View details about updates to AWS managed policies for AWS WAF since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the AWS WAF document history page at Document history.

Policy	Description of change	Date
`AWSWAFConsoleFullAccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleFullAccess.	Added the following permissions for CloudFront pricing plans. For more detail, see Using AWS WAF with CloudFront Flat-Rate Pricing Plans • `pricingplanmanager:GetSubscription` – Grants permission to get pricing plan subscription information associated with your AWS account • `pricingplanmanager:ListSubscriptions` – Grants permission to list the pricing plan subscriptions associated with your AWS account • `pricingplanmanager:UpdateSubscription` – Grants permission to update the pricing plan subscriptions associated with your AWS account • `pricingplanmanager:CancelSubscription` – Grants permission	2025-11-18

Policy	Description of change	Date
	to cancel the pricing plan subscriptions associated with your AWS account • `pricingplanmanager:CancelSubscriptionChange` – Grants permission to cancel pricing plan subscription changes associated with your AWS account • `route53:ListHostedZones` – Grants permission to list the Route 53 hosted zones associated with your AWS account • `route53:GetHostedZone` – Grants permission to view details about Route 53 hosted zones associated with your AWS account • `cloudfront:GetDistribution` – Grants permission to view details about CloudFront distributions associated with your AWS account	

Policy	Description of change	Date
`AWSWAFConsoleReadOnlyAccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleReadOnlyAccess.	Added the following permissions for CloudFront pricing plans. For more detail, see Using AWS WAF with CloudFront Flat-Rate Pricing Plans • `pricingplanmanager:GetSubscription` – Grants permission to get pricing plan subscription information associated with your AWS account • `pricingplanmanager:ListSubscriptions` – Grants permission to list the pricing plan subscriptions associated with your AWS account • `route53:ListHostedZones` – Grants permission to list the Route 53 hosted zones associated with your AWS account • `route53:GetHostedZone` – Grants permission to view details about Route 53 hosted zones associated with your AWS account • `cloudfront:GetDistribution` – Grants permission to view details about CloudFront distribut	2025-11-18

Policy	Description of change	Date
	ions associated with your AWS account	

Policy	Description of change	Date
`AWSWAFConsoleReadOnlyAccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleReadOnlyAccess.	Updated the following permissions: • `apigateway:GET` – Updated the resource scope to restrict retrievable resources to only Amazon API Gateway resources Added the following permissions: **AWS AppSync** • `appsync:ListApis` – Grants permission to retrieve AWS AppSync APIs in your AWS account **Amazon CloudWatch** • `logs:StartQuery` – Grants permission to start a query of one or more Amazon CloudWatch Logs insights log groups • `logs:DescribeQueryDefinitions` – Grants permission to get a list of Amazon CloudWatch Logs insights query definitions and save them in your AWS or a source AWS account • `logs:GetQueryResults` – Grants permissio	2025-11-03

Policy	Description of change	Date
	n to get the results from a specified Amazon CloudWatch Logs insights query **Amazon Data Firehose** • `firehose:ListDeliveryStreams` – Grants permission to retrieve the Firehose streams in your AWS account **AWS Price List** • `pricing:GetPriceListFileUrl` – Grants permission to get the URL used to retrieve a Price List file from AWS Price List • `pricing:ListPriceLists` – Grants permission to retrieve a list of Price List references from AWS Price List **AWS Marketplace** • `aws-marketplace:ViewSubscriptions` – Grants permission to view AWS Marketplace software subscriptions associated with your AWS account	

Policy	Description of change	Date
`AWSWAFConsoleFullAccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleFullAccess.	Updated the following permissions: • `apigateway:GET` – Updated the resource scope to restrict retrievable resources to only Amazon API Gateway resources Added the following permissions: **AWS AppSync** • `appsync:ListApis` – Grants permission to retrieve AWS AppSync APIs in your AWS account **Amazon CloudWatch** • `logs:StartQuery` – Grants permission to start a query of one or more Amazon CloudWatch Logs insights log groups • `logs:DescribeQueryDefinitions` – Grants permission to get a list of Amazon CloudWatch Logs insights query definitions and save them in your AWS or a source AWS account • `logs:GetQueryResults` – Grants permissio	2025-11-03

Policy	Description of change	Date
	n to get the results from a specified Amazon CloudWatch Logs insights query **Amazon Data Firehose** • `firehose:ListDeliveryStreams` – Grants permission to retrieve the Firehose streams in your AWS account **AWS Price List** • `pricing:GetPriceListFileUrl` – Grants permission to get the URL used to retrieve a Price List file from AWS Price List • `pricing:ListPriceLists` – Grants permission to retrieve a list of Price List references from AWS Price List **AWS Marketplace** • `aws-marketplace:ViewSubscriptions` – Grants permission to view AWS Marketplace software subscriptions associated with your AWS account	

Policy	Description of change	Date
`AWSWAFFullAccess` This policy allows AWS WAF to manage AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFFullAccess.	Added permissions AssociateWebACL, DisassociateWebACL, GetWebACLForResource, and ListResourcesForWebACL required for AWS Amplify.	2025-05-05
`AWSWAFReadOnlyAccess` This policy allows AWS WAF to manage AWS resources on your behalf in AWS WAF and in integrated services. For details about this policy, see AWSWAFReadOnlyAccess in the IAM console.	Added permissions GetWebACLForResource and ListResourcesForWebACL required for AWS Amplify.	2025-05-05

Policy	Description of change	Date
`AWSWAFConsoleReadOnlyAccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleReadOnlyAccess.	Added the following permissions: **Amplify** • `amplify:GetWebACLForResource` – Grants permission to retrieve the AWS WAF protection pack (web ACL) associated with an Amplify resource • `amplify:ListApps` – Grants permission to retrieve the Amplify apps in your AWS account • `amplify:ListResourcesForWebACL` – Grants permission to retrieve the Amplify apps associated with an AWS WAF protection pack (web ACL) **CloudFront** • `cloudfront:GetDistributionConfig` – Grants permission to get the configuration information about a CloudFront distribution • `cloudfront:GetDistributionTenant` – Grants permission to get the information about a	2025-05-05

Policy	Description of change	Date
	CloudFront distribution tenant • `cloudfront:ListDistributionTenants` – Grants permission to list the CloudFront distribution tenants associated with your AWS account • `cloudfront:ListDistributionTenantsByCustomization` – Grants permission to list filtered CloudFront distribution tenants associated with your AWS account	

Policy	Description of change	Date
`AWSWAFConsoleFullAccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleFullAccess.	Added the following permissions: **Amplify** • `amplify:AssociateWebACL` – Grants permission to associate an AWS WAF protection pack (web ACL) to an Amplify resource • `amplify:DisassociateWebACL` – Grants permission to disassociate an AWS WAF protection pack (web ACL) from an Amplify resource • `amplify:GetWebACLForResource` – Grants permission to retrieve the AWS WAF protection pack (web ACL) associated with an Amplify resource • `amplify:ListApps` – Grants permission to retrieve the Amplify apps in your AWS account • `amplify:ListResourcesForWebACL` – Grants permission to retrieve the Amplify apps associated with an AWS WAF protection pack (web ACL)	2025-05-05

Policy	Description of change	Date
	CloudFront • `cloudfront:AssociateDistributionTenantWebACL` – Grants permission to associate an AWS WAF protection pack (web ACL) with a CloudFront distribution tenant • `cloudfront:AssociateDistributionWebACL` – Grants permission to associate an AWS WAF protection pack (web ACL) with a CloudFront distribution • `cloudfront:DisassociateDistributionTenantWebACL` – Grants permission to disassociate an AWS WAF protection pack (web ACL) with a CloudFront distribution tenant • `cloudfront:DisassociateDistributionWebACL` – Grants permission to disassociate an AWS WAF protection pack (web ACL) with a CloudFront distribution • `cloudfront:GetDistributionConfig` – Grants permission to get	

Policy	Description of change	Date
	the configuration information about a CloudFront distribution • `cloudfront:GetDistributionTenant` – Grants permission to get the information about a CloudFront distribution tenant • `cloudfront:ListDistributionTenants` – Grants permission to list the CloudFront distribution tenants associated with your AWS account • `cloudfront:ListDistributionTenantsByCustomization` – Grants permission to list filtered CloudFront distribution tenants associated with your AWS account	
`WAFV2LoggingServiceRolePolicy` This policy allows AWS WAF to write logs to Amazon Data Firehose. It's used only if you enable logging. Details in IAM console: WAFV2LoggingServiceRolePolicy.	Added Statement IDs (Sids) to the permissions settings in the service-linked role that this policy is attached to.	2024-06-03

Policy	Description of change	Date
`AWSServiceRoleForWAFV2Logging` This service-linked role provides permissions policies that allow AWS WAF to write logs to Amazon Data Firehose. Details in IAM console: AWSServiceRoleForWAFV2Logging.	Added Statement IDs (Sids) to the permissions settings.	2024-06-03
AWS WAF additions to change tracking	AWS WAF started tracking changes for the managed policy `WAFV2LoggingServiceRolePolicy` and the service-linked role `AWSServiceRoleForWAFV2Logging` .	2024-06-03
`AWSWAFFullAccess` This policy allows AWS WAF to manage AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFFullAccess.	Expanded permissions to add AWS Verified Access instances to the resource types that you can protect with AWS WAF.	2023-06-17

Policy	Description of change	Date
`AWSWAFReadOnlyAccess` This policy allows AWS WAF to manage AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFReadOnlyAccess.	Expanded permissions to add AWS Verified Access instances to the resource types that you can protect with AWS WAF.	2023-06-17
`AWSWAFConsoleFullAccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleFullAccess.	Expanded permissions to add AWS Verified Access instances to the resource types that you can protect with AWS WAF.	2023-06-17
`AWSWAFConsoleReadOnlyAccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleReadOnlyAccess.	Expanded permissions to add AWS Verified Access instances to the resource types that you can protect with AWS WAF.	2023-06-17

Policy	Description of change	Date
`AWSWAFFullAccess` This policy allows AWS WAF to manage AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFFullAccess.	Expanded permissions to correct the access settings for AWS App Runner services.	2023-06-06
`AWSWAFReadOnlyAccess` This policy allows AWS WAF to manage AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFReadOnlyAccess.	Expanded permissions to correct the access settings for AWS App Runner services.	2023-06-06
`AWSWAFConsoleFullAccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleFullAccess.	Expanded permissions to correct the access settings for AWS App Runner services.	2023-06-06

Policy	Description of change	Date
`AWSWAFConsoleReadOnlyAccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleReadOnlyAccess.	Expanded permissions to correct the access settings for AWS App Runner services.	2023-06-06
`AWSWAFFullAccess` This policy allows AWS WAF to manage AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFFullAccess.	Expanded permissions to add AWS App Runner services to the resource types that you can protect with AWS WAF.	2023-03-30
`AWSWAFReadOnlyAccess` This policy allows AWS WAF to manage AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFReadOnlyAccess.	Expanded permissions to add AWS App Runner services to the resource types that you can protect with AWS WAF.	2023-03-30

Policy	Description of change	Date
`AWSWAFConsoleFullAccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleFullAccess.	Expanded permissions to add AWS App Runner services to the resource types that you can protect with AWS WAF.	2023-03-30
`AWSWAFConsoleReadOnlyAccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleReadOnlyAccess.	Expanded permissions to add AWS App Runner services to the resource types that you can protect with AWS WAF.	2023-03-30
`AWSWAFFullAccess` This policy allows AWS WAF to manage AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFFullAccess.	Expanded permissions to add Amazon Cognito user pools to the resource types that you can protect with AWS WAF.	2022-08-25

Policy	Description of change	Date
`AWSWAFReadOnlyAccess` This policy allows AWS WAF to manage AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFReadOnlyAccess.	Expanded permissions to add Amazon Cognito user pools to the resource types that you can protect with AWS WAF.	2022-08-25
`AWSWAFConsoleFullAccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleFullAccess.	Expanded permissions to add Amazon Cognito user pools to the resource types that you can protect with AWS WAF.	2022-08-25
`AWSWAFConsoleReadOnlyAccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleReadOnlyAccess.	Expanded permissions to add Amazon Cognito user pools to the resource types that you can protect with AWS WAF.	2022-08-25

Policy	Description of change	Date
`AWSWAFFullAccess` This policy allows AWS WAF to manage AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFFullAccess.	Corrected the permissions settings for log delivery for Amazon Simple Storage Service (Amazon S3) and Amazon CloudWatch Logs. This change resolves access denied errors that were occurring during logging configuration. For informati on about logging your protection pack (web ACL) traffic, see Logging AWS WAF protection pack (web ACL) traffic.	2022-01-11
`AWSWAFConsoleFullA ccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleFullAccess.	Corrected the permissions settings for log delivery for Amazon Simple Storage Service (Amazon S3) and Amazon CloudWatch Logs. This change resolves access errors that were occurring during logging configura tion. For information about logging your protection pack (web ACL) traffic, see Logging AWS WAF protection pack (web ACL) traffic.	2022-01-11

Policy	Description of change	Date
`AWSWAFFullAccess` This policy allows AWS WAF to manage AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFFullAccess.	Added new permissions for expanded logging options. This change gives AWS WAF access to the additional logging destinations Amazon Simple Storage Service (Amazon S3) and Amazon CloudWatch Logs. For information about logging your protection pack (web ACL) traffic, see Logging AWS WAF protection pack (web ACL) traffic.	2021-11-15
`AWSWAFConsoleFullAccess` This policy allows AWS WAF to manage AWS console resources and other AWS resources on your behalf in AWS WAF and in integrated services. Details in IAM console: AWSWAFConsoleFullAccess.	Added new permissions for expanded logging options. This change gives AWS WAF access to the additional logging destinations Amazon Simple Storage Service (Amazon S3) and Amazon CloudWatch Logs. For information about logging your protection pack (web ACL) traffic, see Logging AWS WAF protection pack (web ACL) traffic.	2021-11-15
AWS WAF started tracking changes	AWS WAF started tracking changes for its AWS managed policies.	2021-3-01

Troubleshooting AWS WAF identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with AWS WAF and IAM.

Topics

- I am not authorized to perform an action in AWS WAF
- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my AWS WAF resources

I am not authorized to perform an action in AWS WAF

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `wafv2:`*`GetWidget`* permissions.

```
User: arn:aws-eusc:iam::123456789012:user/mateojackson is not authorized to perform:
 wafv2:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `wafv2:`*`GetWidget`* action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to AWS WAF.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in AWS WAF. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws-eusc:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my AWS WAF resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether AWS WAF supports these features, see How AWS WAF works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Using service-linked roles for AWS WAF

This section explains how to use service-linked roles to give AWS WAF access to resources in your AWS account.

AWS WAF uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to AWS WAF. Service-linked roles are predefined by AWS WAF and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up AWS WAF easier because you don’t have to manually add the necessary permissions. AWS WAF defines the permissions of its service-linked roles, and unless

defined otherwise, only AWS WAF can assume its roles. The defined permissions include the trust policy and the permissions policy. That permissions policy can't be attached to any other IAM entity.

You can delete a service-linked role only after first deleting the role's related resources. This protects your AWS WAF resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for AWS WAF

AWS WAF uses the service-linked role `AWSServiceRoleForWAFV2Logging` to write logs to Amazon Data Firehose. This role is used only if you enable logging in AWS WAF. For information about logging, see Logging AWS WAF protection pack (web ACL) traffic.

This service-linked role is attached to the AWS managed policy `WAFV2LoggingServiceRolePolicy`. For more information about the managed policy, see AWS managed policy: WAFV2LoggingServiceRolePolicy.

The `AWSServiceRoleForWAFV2Logging` service-linked role trusts the service to assume the role `wafv2.amazonaws.com`.

The permissions policies of the role allow AWS WAF to complete the following actions on the specified resources:

- Amazon Data Firehose actions: `PutRecord` and `PutRecordBatch` on Firehose data stream resources with a name that starts with `aws-waf-logs-`. For example, `aws-waf-logs-us-east-2-analytics`.
- AWS Organizations action: `DescribeOrganization` on Organizations organizations resources.
- AWS Key Management Service actions: `GenerateDataKey` and `Decrypt` to allow Amazon Data Firehose to use customer managed AWS KMS keys to encrypt data.

See the full service-linked role in the IAM console: AWSServiceRoleForWAFV2Logging.

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

Creating a service-linked role for AWS WAF

You don't need to manually create a service-linked role. When you enable AWS WAF logging on the AWS Management Console, or you make a `PutLoggingConfiguration` request in the AWS WAF CLI or the AWS WAF API, AWS WAF creates the service-linked role for you.

You must have the `iam:CreateServiceLinkedRole` permission to enable logging.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you enable AWS WAF logging, AWS WAF creates the service-linked role for you again.

Editing a service-linked role for AWS WAF

AWS WAF doesn't allow you to edit the `AWSServiceRoleForWAFV2Logging` service-linked role. After you create a service-linked role, you can't change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting a service-linked role for AWS WAF

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don’t have an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

> **Note**
>
> If the AWS WAF service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To delete AWS WAF resources used by the `AWSServiceRoleForWAFV2Logging`

1. On the AWS WAF console, remove logging from every web ACL. For more information, see Logging AWS WAF protection pack (web ACL) traffic.
2. Using the API or CLI, submit a `DeleteLoggingConfiguration` request for each web ACL that has logging enabled. For more information, see AWS WAF API Reference.

To manually delete the service-linked role using IAM

Use the IAM console, the IAM CLI, or the IAM API to delete the `AWSServiceRoleForWAFV2Logging` service-linked role. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

Supported Regions for AWS WAF service-linked roles

AWS WAF supports using service-linked roles in all of the regions where the service is available. For more information, see AWS WAF endpoints and quotas.

Logging and monitoring in AWS WAF

This section explains how to use AWS tools for monitoring and responding to events in AWS WAF.

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS WAF and your AWS solutions. You should collect monitoring data from all parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your AWS WAF resources and responding to potential events:

Amazon CloudWatch alarms

Using CloudWatch alarms, you watch a single metric over a time period that you specify. If the metric exceeds a given threshold, CloudWatch sends a notification to an Amazon SNS topic or AWS Auto Scaling policy. For more information, see Monitoring with Amazon CloudWatch.

AWS CloudTrail logs

CloudTrail provides a record of actions taken by a user, role, or an AWS service in AWS WAF. Using the information collected by CloudTrail, you can determine the request that was made to AWS WAF, the IP address from which the request was made, who made the request, when it was made, and additional details. For more information, see Logging API calls with AWS CloudTrail.

AWS WAF protection pack (web ACL) traffic logging

AWS WAF offers logging for the traffic that your protection packs (web ACLs) analyze. The logs include information such as the time that AWS WAF received the request from your protected AWS resource, detailed information about the request, and the action setting for the rule that the request matched. For more information, see Logging AWS WAF protection pack (web ACL) traffic.

Validating compliance in AWS WAF

This section explains your compliance responsibility when using AWS WAF.

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. For more information about your compliance responsibility when using AWS services, see AWS Security Documentation.

Building for resilience in AWS WAF

This section explains how AWS architecture supports data redundancy for AWS WAF.

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure security in AWS WAF

This section explains how AWS WAF isolates service traffic.

As a managed service, AWS WAF is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access AWS WAF through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

AWS WAF quotas

Note

This is the latest version of AWS WAF. For AWS WAF Classic, see AWS WAF Classic.

AWS WAF is subject to the following quotas (formerly referred to as limits). These quotas are the same for all Regions in which AWS WAF is available. Each Region is subject to these quotas individually. The quotas are not cumulative across Regions.

AWS WAF has default quotas on the maximum number of entities you can have per account. You can request an increase in these quotas.

Resource	Default quota per account per Region
Maximum number of protection packs (web ACLs)	100
Maximum number of rule groups	100
Maximum number of IP sets	100
Maximum number of requests per second per protection pack (web ACL)	100,000
Maximum number of custom request headers per protection pack (web ACL) or rule group	100
Maximum number of custom response headers per protection pack (web ACL) or rule group	100
Maximum number of custom response bodies per protection pack (web ACL) or rule group	50
Maximum number of token domains in a protection pack (web ACL) token domain list	10
Maximum number of regex pattern sets	10
Maximum number of Application Load Balancer associations per protection pack (web ACL)	100
Maximum number of Amazon API Gateway REST API associations per protection pack (web ACL)	100

Resource	Default quota per account per Region
Maximum number of Amazon Cognito user pool associations per protection pack (web ACL)	100
Maximum number of AWS AppSync GraphQL API associations per protection pack (web ACL)	100
Maximum number of AWS App Runner service associations per protection pack (web ACL)	100
Maximum number of AWS Verified Access instance associations per protectio n pack (web ACL)	100
Maximum number of AWS Amplify associations per protection pack (web ACL)	100
Maximum number of Amazon Bedrock AgentCore Gateway associations per protection pack (web ACL)	100

The maximum requests per second (RPS) allowed for AWS WAF on CloudFront is set by CloudFront and described in the CloudFront Developer Guide.

AWS WAF has fixed quotas on the following entity settings per account per Region. These quotas can't be changed.

Resource	Quota per account per Region
Maximum protection pack (web ACL) capacity units (WCUs) per protection pack (web ACL)*	5,000
Maximum WCUs per rule group	5,000

Resource	Quota per account per Region
Maximum number of reference statements per rule group. In a rule group, a reference statement can reference an IP set or a regex pattern set.	50
Maximum number of reference statements per protection pack (web ACL). In a protection pack (web ACL), a reference statement can reference a rule group, an IP set, or a regex pattern set.	50
Maximum number of IP addresses in CIDR notation per IP set	10,000
Maximum number of rate-based rules per protection pack (web ACL)	10
Maximum number of rate-based rules per rule group	4
Minimum request rate that can be defined for a rate-based rule	10
Maximum number of unique IP addresses that can be rate limited per rate-based rule	10,000
Maximum number of characters in a string match statement	200
Maximum number of characters in each regex pattern	200
Maximum number of unique regex patterns per regex pattern set	10
Maximum size of a web request body that can be inspected for Application Load Balancer and AWS AppSync protections	8 KB
Maximum size of a web request body that can be inspected for Amazon Bedrock AgentCore Gateway protections	16 KB
Maximum size of a web request body that can be inspected for CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway protections**	64 KB
Maximum number of text transformations per rule statement	10

Resource	Quota per account per Region
Maximum size of the custom response body content for a single custom response definition	4 KB
Maximum number of custom headers for a single custom response definitio n	10
Maximum number of custom headers for a single custom request definition	10
Maximum combined size of all response body content for a single rule group or a single protection pack (web ACL)	50 KB
Maximum number of Geo match country codes inside a single `GeoMatchS tatement`	50

*Using more than 1,500 WCUs in a protection pack (web ACL) incurs costs beyond the basic protection pack (web ACL) price. For more information, see Web ACL capacity units (WCUs) in AWS WAF and AWS WAF Pricing.

**By default, the body inspection limit is set to 16 KB for CloudFront, API Gateway, Amazon Cognito, App Runner, Verified Access, and Amazon Bedrock AgentCore Gateway resources, but you can increase this for any of these resources in your protection pack (web ACL) configuration, up to the listed maximum. For more information, see Considerations for managing body inspection in AWS WAF.

AWS WAF has the following fixed quotas on calls per account per Region. These quotas apply to the total calls to the service through any available means, including the console, CLI, AWS CloudFormation, the REST API, and the SDKs. These quotas can't be changed.

Call type	Quota per account per Region
Maximum number of calls to `AssociateWebACL`	One request every 2 seconds

Call type	Quota per account per Region
Maximum number of calls to `DisassociateWebACL`	One request every 2 seconds
Maximum number of calls to `GetWebACLForResource`	One request per second
Maximum number of calls to `ListResourcesForWebACL`	One request per second
Maximum number of calls to `GetDecryptedAPIKey`	One request every 2 seconds
Maximum number of calls to any individual `Get` or `List` action, if no other quota is defined for it	Five requests per second
Maximum number of calls to any individual `Create`, `Put`, or `Update` action, if no other quota is defined for it	One request per second

AWS WAF has the following fixed quotas on calls by all accounts in a single organization in AWS Organizations. These quotas apply to the total calls to the service through any available means, including the console, CLI, AWS CloudFormation, the REST API, and the SDKs. These quotas can't be changed.

Call type	Quota per organization in a single Region
Maximum number of calls by all accounts in an organization to `ListResourcesForWebACL`, in any single Region for the Regions US East (N. Virginia) (us-east-1), US West (Oregon) (us-west-2), or Europe (Ireland) (eu-west-1).	12 requests per second

Call type	Quota per organization in a single Region
Maximum number of calls by all accounts in an organization to `ListResources ForWebACL` , in any single Region that doesn't have a different quota listed in this table.	6 requests per second

AWS WAF has the following quotas for AI traffic monetization features. Quotas marked as adjustable can be increased through a quota increase request.

Resource	Default quota	Adjustable
Protection packs per account	25	Yes
Protection packs per protection pack (web ACL)	10	Yes
Pricing tiers per protection pack	20	No
Content zones per protection pack	50	Yes
Payment providers per protection pack	3	No
Minimum price per request	$0.00001 (10^-5 USD)	No
Maximum price per request	$100	No
Access token TTL range	60-86400 seconds	No

Migrating your AWS WAF Classic resources to AWS WAF

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

This section provides guidance for migrating your rules and protection packs (web ACLs) from AWS WAF Classic to AWS WAF. AWS WAF was released in November 2019. If you created resources like rules and protection packs (web ACLs) using AWS WAF Classic, you either need to work with them using AWS WAF Classic or migrate them to this latest version.

Warning

AWS WAF Classic support will end on September 30, 2025.

Before you start your migration work, familiarize yourself with AWS WAF by reading through AWS WAF.

Topics

- Why migrate to AWS WAF?
- Migration caveats and limitations
- How the migration works
- Migrating a protection pack (web ACL) from AWS WAF Classic to AWS WAF

Why migrate to AWS WAF?

The latest version of AWS WAF provides many improvements over the prior version, while maintaining most of the concepts and terminology that you're accustomed to.

The following list describes the major changes in the latest AWS WAF. Before you continue with your migration, please take some time to review this list and to familiarize yourself with the rest of the AWS WAF guide.

- **Support for AWS WAF Classic will end on September 30, 2025.**
- **AWS Managed Rules for AWS WAF** – The rule groups now available through AWS Managed Rules provide protection against common web threats. Most of these rule groups are included free of charge with AWS WAF. For more information, see AWS Managed Rules rule groups list and the blog post Announcing AWS Managed Rules for AWS WAF.
- **New AWS WAF API** – The new API allows you to configure all of your AWS WAF resources using a single set of APIs. To distinguish between regional and global applications, the new API includes a `scope` setting. For more information about the API, see the AWS WAFV2 Actions and AWS WAFV2 Data Types.

  In the APIs, SDKs, CLIs, and AWS CloudFormation, AWS WAF Classic retains its naming schemes and this latest version of AWS WAF is referred to with an added `V2` or `v2`, depending on the context.
- **Simplified service quotas (limits)** – AWS WAF now allows more rules per protection pack (web ACL) and allows you to express longer regex patterns. For more information, see AWS WAF quotas.
- **Computing needs determine capacity limits** – The limits for protection packs (web ACLs) are now based on protection pack (web ACL) capacity units (WCUs). AWS WAF calculates a rule's WCUs according to its required operating capacity. The total WCUs for a protection pack (web ACL) equals the sum of WCUs from all its rules and rule groups.

  For general information about WCU, see How AWS WAF works. For information about each rule's WCU usage, see Using rule statements in AWS WAF.
- **Document-based rule writing** – You can now write and express rules, rule groups, and protection packs (web ACLs) in JSON format. You no longer need to use individual API calls to create different conditions and then associate the conditions to a rule. This greatly simplifies how you write and maintain your code. You can access a JSON format of your protection packs (web ACLs) through the console when you're viewing the protection pack (web ACL), by choosing **Download**

protection pack (web ACL) as JSON. When you are creating your own rule, you can access its JSON representation by choosing **Rule JSON editor**.

- **Rule nesting and full logical operation support** – You can write complex combined rules by using logical rule statements and by using nesting. You can create statements such as `[A AND NOT(B OR C)]`. For more information, see Using logical rule statements in AWS WAF.
- **Improved rate-based rules** – In the latest version of AWS WAF, you can customize the time window that the rule evaluates and how the rule aggregates requests. You can customize aggregation using combinations of a number of web request characteristics. Additionally the latest rate-based rules react more quickly to changes in traffic. For more information, see Using rate-based rule statements in AWS WAF.
- **Variable CIDR range support for IP set** – IP set specifications now have more flexibility in the IP ranges. For IPv4, AWS WAF supports `/1` to `/32`. For IPv6, AWS WAF supports `/1` to `/128`. For more information about IP sets, see IP set match rule statement.
- **Chainable text transformations** – AWS WAF can perform multiple text transformations against web request content before inspecting it. For more information, see Using text transformations in AWS WAF.
- **Improved console experience** – The new AWS WAF console features visual rule builder and a more user intuitive console design.
- **Expanded options for Firewall Manager AWS WAF policies** – In the Firewall Manager management of AWS WAF protection packs (web ACLs), you can now create a set of rule groups that AWS WAF processes first and a set of rule groups that AWS WAF processes last. After you apply the AWS WAF policy, local account owners can add their own rule groups that AWS WAF processes in between these two sets. For more information about Firewall Manager AWS WAF policies, see Using AWS WAF policies with Firewall Manager.
- **AWS CloudFormation support for all rule statement types** – AWS WAF in AWS CloudFormation supports all rule statement types that the AWS WAF console and API support. Additionally, you can easily convert the rules that you write in JSON format to YAML format.

Migration caveats and limitations

The migration only handles protection pack (web ACL) configurations, and the protection pack (web ACL) migration doesn't bring over all settings exactly as you have them in AWS WAF Classic. Some configuration items require manual configuration in AWS WAF (v2). A few things don't map exactly between the two versions, and you'll need to decide how you want to configure the

functionality in AWS WAF (v2). Some settings, like the protection pack (web ACL)'s associations with AWS resources, are disabled initially in the new version so you can add them when you're ready.

The following list describes the caveats of the migration and describes any steps you might want to take in response. Use this overview to plan your migration. The detailed migration steps, later on, walk you through the recommended mitigation steps.

- **Single account migration** – You can only migrate AWS WAF Classic resources for any account to AWS WAF resources for the same account.
- **Only protection pack (web ACL) configurations** – The migration only migrates protection packs (web ACLs) and resources that the protection packs (web ACLs) are using. To migrate a resource, such as a rule group or IP set, that's not used by any migrated web ACL, manually create the resource in AWS WAF (v2).
- **No AWS Marketplace managed rules** – The migration doesn't bring over any managed rules from AWS Marketplace sellers. Some AWS Marketplace sellers have equivalent managed rules for AWS WAF that you can subscribe to again. Before you do this, review the AWS Managed Rules that are provided with the latest version of AWS WAF. Most of these are free of charge for AWS WAF users. For information about managed rules, see Using managed rule groups in AWS WAF.
- **No protection pack (web ACL) associations** – The migration doesn't bring over any associations between the protection pack (web ACL) and protected resources. This is by design, to avoid affecting your production workload. After you verify that everything is migrated correctly, associate the new protection pack (web ACL) with your resources.
- **Logging disabled** – Logging for the migrated protection pack (web ACL) is disabled by default. This is by design. Enable logging when you are ready to switch over from AWS WAF Classic to AWS WAF.
- **No AWS Firewall Manager rule groups** – The migration doesn't handle rule groups that are managed by Firewall Manager. You can migrate a protection pack (web ACL) that's managed by Firewall Manager, but the migration doesn't bring over the rule group. Instead of using the migration tool for these protection packs (web ACLs), recreate the policy for the new AWS WAF in Firewall Manager.

> **Note**
>
> The rule groups that Firewall Manager managed for AWS WAF Classic were Firewall Manager rule groups. With the new version of AWS WAF, the rule groups are AWS WAF rule groups. Functionally, they are the same.

- **AWS WAF Security Automations caveat** – Don't try to migrate any AWS WAF Security Automations. The migration doesn't convert Lambda functions, which might be in use by the automations. Consider deploying the automations for the latest version instead. For information, see AWS WAF Security Automations.

How the migration works

You can migrate your web ACLs from AWS WAF Classic to AWS WAF v2 using several methods. Follow these steps to complete your migration.

To migrate from AWS WAF Classic to AWS WAF v2

1. Identify your AWS WAF Classic web ACLs:
   - View a list of your web ACLs in the AWS Health dashboard.
   - Use the AWS WAF Classic web ACL cleanup script to get a list of all your web ACLs and their associations. This helps you identify which web ACLs are actively protecting resources and allows you to delete unused web ACLs.
2. Migrate individual web ACLs:
   - Follow the migration process in the AWS WAF Developer Guide.
   - Use the migration wizard to parse your AWS WAF Classic web ACL and generate an AWS CloudFormation template.
   - Use the generated template to create an equivalent AWS WAF v2 web ACL and complete the migration.
3. For multiple eligible web ACLs:
   - Use the AWS WAF bulk migration script to migrate multiple eligible AWS WAF Classic web ACLs simultaneously.
4. For web ACLs managed by AWS Firewall Manager:

- Firewall Manager policies use AWS WAF Classic web ACLs with AWS WAF Classic policies. For Shield Advanced policies created before January 2022, Firewall Manager also uses AWS WAF Classic web ACLs. You must migrate these policies to use AWS WAF v2 web ACLs.

  Follow the instructions at Migrating AWS WAF Classic Web ACLs in Firewall Manager.

Important

We recommend reviewing each migrated web ACLto ensure it meets your security requirements before associating it with your resources.

Migrating a protection pack (web ACL) from AWS WAF Classic to AWS WAF

The automated migration carries over most of your AWS WAF Classic protection pack (web ACL) configuration, leaving some things that you need to handle manually.

Note

Some protection configurations cannot be automatically migrated, and require manual configuration in AWS WAF (v2). See the list at Migration caveats and limitations.

The following lists the high-level steps for migrating a protection pack (web ACL).

1. The automated migration reads everything related to your existing protection pack (web ACL), without modifying or deleting anything in AWS WAF Classic. It creates a representation of the web ACL and its related resources, compatible with AWS WAF. It generates an CloudFormation template for the new protection pack (web ACL) and stores it in an Amazon S3 bucket.
2. You deploy the template into CloudFormation, in order to recreate the protection pack (web ACL) and related resources in AWS WAF.
3. You review the protection pack (web ACL), and manually complete the migration, making sure that your new protection pack (web ACL) takes full advantage of the capabilities of the latest AWS WAF.
4. You manually switch your protected resources over to the new protection pack (web ACL).

Topics

- Migrating a protection pack (web ACL): automated migration
- Migrating a protection pack (web ACL): manual follow-up
- Migrating a protection pack (web ACL): additional considerations
- Migrating a protection pack (web ACL): switchover

Migrating a protection pack (web ACL): automated migration

To automatically migrate a protection pack (web ACL) configuration from AWS WAF Classic to AWS WAF

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. Choose **Switch to AWS WAF Classic** and review your configuration settings for the protection pack (web ACL). Make note of the settings, considering the caveats and limitations described in the preceding section, Migration caveats and limitations.
3. In the informational dialogue at the top, locate the sentence that starts with **Migrate protection packs (web ACLs)** and choose the link to the **migration wizard**. This launches the migration wizard.

   If you don't see the informational dialogue, you might have closed it since you launched the AWS WAF Classic console. In the navigation bar, choose **Switch to new AWS WAF** then choose **Switch to AWS WAF Classic**, and the informational dialogue should reappear.
4. Select the protection pack (web ACL) that you want to migrate.
5. For **Migration configuration**, provide an Amazon S3 bucket to use for the template. You need an Amazon S3 bucket that's configured properly for the migration API, to store the AWS CloudFormation template that it generates.

   - If the bucket is encrypted, the encryption must use Amazon S3 (SSE-S3) keys. The migration doesn't support encryption with AWS Key Management Service (SSE-KMS) keys.
   - The bucket name must start with `aws-waf-migration-`. For example, `aws-waf-migration-my-web-acl`.
   - The bucket must be in the Region where you are deploying the template. For example, for a protection pack (web ACL) in `us-west-2`, you must use an Amazon S3 bucket in `us-west-2` and you must deploy the template stack to `us-west-2`.

6. For **S3 bucket policy**, we recommend choosing **Auto apply the bucket policy required for migration**. Alternatively, if you want to manage the bucket on your own, you must manually apply the following bucket policy:

   - For global Amazon CloudFront applications (`waf`):

     JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "apiv2migration.waf.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws-eusc:s3:::<BUCKET_NAME>/AWSWAF/
<CUSTOMER_ACCOUNT_ID>/*"
        }
    ]
}
```

   - For regional Amazon API Gateway or Application Load Balancer applications (`waf-regional`):

     JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "apiv2migration.waf-regional.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws-eusc:s3:::<BUCKET_NAME>/AWSWAF/
<CUSTOMER_ACCOUNT_ID>/*"
        }
    ]
```

```
}
```

7. For **Choose how to handle rules that cannot be migrated**, choose either to exclude rules that can't be migrated, or to stop the migration. For information about rules that can't be migrated, see Migration caveats and limitations.
8. Choose **Next**.
9. For **Create CloudFormation template**, verify your settings, then choose **Start creating CloudFormation template** to begin the migration process. This can take a few minutes, depending on the complexity of your protection pack (web ACL).
10. In **Create and run CloudFormation stack to complete migration**, you can choose to go to the AWS CloudFormation console to create a stack from the template, to create the new protection pack (web ACL) and its resources. To do this, choose **Create CloudFormation stack**.

After the automatic migration process completes, you're ready to proceed to the manual follow-up steps. See Migrating a protection pack (web ACL): manual follow-up.

Migrating a protection pack (web ACL): manual follow-up

After the automated migration is complete, review the newly created protection pack (web ACL) and fill in the components that the migration doesn't bring over for you. The following procedure covers the aspects of protection pack (web ACL) management that the migration doesn't handle. For the list, see Migration caveats and limitations.

To finish the basic migration - manual steps

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/homev2.
2. The console should automatically use the latest version of AWS WAF. To verify this, in the navigation pane, check that you can see the option **Switch to AWS WAF Classic**. If you see **Switch to new AWS WAF**, choose that to switch to the latest version.
3. In the navigation pane, choose **protection packs (web ACLs)**.
4. In the **protection packs (web ACLs)** page, locate your new protection pack (web ACL) in the list for the Region where you created it. Choose the protection pack (web ACL)'s name to bring up the settings for the protection pack (web ACL).
5. Review all of the settings for the new protection pack (web ACL) against your prior AWS WAF Classic web ACL. By default, logging and protected resource associations are disabled. You enable those when you're ready to switch over.

6. If your AWS WAF Classic protection pack (web ACL) had a managed rule group, the rule group inclusion wasn't brought over in the migration. You can add managed rule groups to the new protection pack (web ACL). Review the information about managed rule groups, including the list of AWS Managed Rules that are available with the new version of AWS WAF, at Using managed rule groups in AWS WAF. To add a managed rule group, do the following:

   a. In your protection pack (web ACL) settings page, choose the protection pack (web ACL) **Rules** tab.
   b. Choose **Add rules**, then choose **Add managed rule groups**.
   c. Expand the listing for the vendor of your choice and select the rule groups that you want to add. For AWS Marketplace sellers, you might need to subscribe to the rule groups. For more information about using managed rule groups in your protection pack (web ACL), see Using managed rule groups in AWS WAF and Using protection packs (web ACLs) with rules and rule groups in AWS WAF.

After you finish the basic migration process, we recommend that you review your needs and consider additional options, to be sure that the new configuration is as efficient as possible and that it's using the latest available security options. See Migrating a protection pack (web ACL): additional considerations.

Migrating a protection pack (web ACL): additional considerations

Review your new protection pack (web ACL) and consider the options available to you in the new AWS WAF to be sure that the configuration is as efficient as possible and that it's using the latest available security options.

Additional AWS Managed Rules

Consider implementing additional AWS Managed Rules in your protection pack (web ACL) to increase the security posture for your application. These are included with AWS WAF at no additional cost. AWS Managed Rules feature the following types of rule groups:

- Baseline rule groups provide general protection against a variety of common threats, such as stopping known bad inputs from making it into your application and preventing admin page access.
- Use-case specific rule groups provide incremental protection for many diverse use cases and environments.
- IP reputation lists provide threat intelligence based on the client's source IP.

For more information, see AWS Managed Rules for AWS WAF.

Rule optimization and cleanup

Revisit your old rules and consider optimizing them by rewriting them or removing outdated ones. For example, if in the past, you deployed an AWS CloudFormation template from the technical paper for OWASP Top 10 Web Application Vulnerabilities, Prepare for the OWASP Top 10 Web Application Vulnerabilities Using AWS WAF and Our New White Paper, you should consider replacing that with AWS Managed Rules. While the concept found within the document is still applicable and may assist you in writing your own rules, the rules created by the template have been largely superseded by AWS Managed Rules.

Amazon CloudWatch metrics and alarms

Revisit your Amazon CloudWatch metrics and set up alarms as needed. The migration doesn't carry over CloudWatch alarms and it's possible that your metric names aren't what you want.

Review with your application team

Work with your application team and check your security posture. Find out what fields are parsed frequently by the application and add rules to sanitize the input accordingly. Check for any edge cases and add rules to catch these cases if the application's business logic fails to process them.

Plan the switchover

Plan the timing of the switch with your application team. The switch from the old protection pack (web ACL) association to the new one can take a small amount of time to propagate to all areas where your resources are stored. The propagation time can be from a few seconds to a number of minutes. During this time, some requests will be processed by the old protection pack (web ACL) and others will be processed by the new protection pack (web ACL). Your resources will be protected throughout the switch, but you might notice inconsistencies in request handling while the switch is underway.

When you are ready to switch over, follow the procedure at Migrating a protection pack (web ACL): switchover.

Migrating a protection pack (web ACL): switchover

After you've verified your new protection pack (web ACL) settings, you can start to use it in place of your AWS WAF Classic protection pack (web ACL).

To begin using your new AWS WAF protection pack (web ACL)

1. Associate the AWS WAF protection pack (web ACL) with the resources that you want to protect, following the guidance at Associating or disassociating protection with an AWS resource. This automatically disassociates the resources from the old protection pack (web ACL).

   The switch can take from a few seconds to a number of minutes to propagate. During this time, some requests might be processed by the old protection pack (web ACL) and others by the new protection pack (web ACL). Your resources will be protected throughout the switch, but you might notice inconsistencies in request handling until it's complete.
2. Configure logging for the new protection pack (web ACL), following the guidance at Logging AWS WAF protection pack (web ACL) traffic.
3. (Optional) If your AWS WAF Classic protection pack (web ACL) is no longer associated with any resources, consider removing it entirely from AWS WAF Classic. For information, see Deleting a Web ACL.

AWS WAF Classic

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

AWS WAF Classic is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to an Amazon API Gateway API, Amazon CloudFront or an Application Load Balancer. AWS WAF Classic also lets you control access to your content. Based on conditions that you specify, such as the IP addresses that requests originate from or the values of query strings, API Gateway, CloudFront or an Application Load Balancer responds to requests either with the requested content or with an HTTP 403 status code (Forbidden). You also can configure CloudFront to return a custom error page when a request is blocked.

Topics

- Setting up AWS WAF Classic
- How AWS WAF Classic works
- AWS WAF Classic pricing
- Getting started with AWS WAF Classic
- Creating and configuring a Web Access Control List (Web ACL)
- Working with AWS WAF Classic rule groups for use with AWS Firewall Manager
- Getting started with AWS Firewall Manager to enable AWS WAF Classic rules
- Tutorial: Creating an AWS Firewall Manager policy with hierarchical rules

- Logging Web ACL traffic information
- Listing IP addresses blocked by rate-based rules
- How AWS WAF Classic works with Amazon CloudFront features
- Security in AWS WAF Classic
- AWS WAF Classic quotas

Setting up AWS WAF Classic

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

This topic describes preliminary steps, such as creating a user account, to prepare you to use AWS WAF Classic. You aren't charged for these. You are charged only for AWS services that you use.

Note

If you're a new user to AWS WAF, don't follow these setup steps for AWS WAF Classic. Instead, follow the steps for the latest version of AWS WAF, at Setting up your account to use the services.

After you complete these steps, see Getting started with AWS WAF Classic to continue getting started with AWS WAF Classic.

> **Note**
>
> AWS Shield Standard is included with AWS WAF Classic and does not require additional setup. For more information, see How AWS Shield and Shield Advanced work.

Before you use AWS WAF Classic or AWS Shield Advanced for the first time, complete the steps in this section.

Topics

- Sign up for an AWS account
- Download tools

Sign up for an AWS account

To get started with AWS, you need an AWS account. For information about creating an AWS account, see Getting started with an AWS account in the *AWS Account Management Reference Guide*.

Download tools

The AWS Management Console includes a console for AWS WAF Classic, but if you want to access AWS WAF Classic programmatically, see the following:

- If you want to call the AWS WAF Classic API without having to handle low-level details like assembling raw HTTP requests, you can use an AWS SDK. The AWS SDKs provide functions and data types that encapsulate the functionality of AWS WAF Classic and other AWS services. To download an AWS SDK, see the applicable page, which also includes prerequisites and installation instructions:
  - Java
  - JavaScript
  - .NET
  - Node.js
  - PHP
  - Python
  - Ruby

For a complete list of AWS SDKs, see Tools for Amazon Web Services.

- If you're using a programming language for which AWS doesn't provide an SDK, the AWS WAF API Reference documents the operations that AWS WAF Classic supports.
- The AWS Command Line Interface (AWS CLI) supports AWS WAF Classic. The AWS CLI lets you control multiple AWS services from the command line and automate them through scripts. For more information, see AWS Command Line Interface.
- AWS Tools for Windows PowerShell supports AWS WAF Classic. For more information, see AWS Tools for PowerShell Cmdlet Reference.

How AWS WAF Classic works

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

You use AWS WAF Classic to control how API Gateway, Amazon CloudFront or an Application Load Balancer responds to web requests. You start by creating conditions, rules, and web access control lists (web ACLs). You define your conditions, combine your conditions into rules, and combine the rules into a web ACL.

Note

You can also use AWS WAF Classic to protect your applications that are hosted in Amazon Elastic Container Service (Amazon ECS) containers. Amazon ECS is a highly scalable,

fast container management service that makes it easy to run, stop, and manage Docker containers on a cluster. To use this option, you configure Amazon ECS to use an AWS WAF Classic enabled Application Load Balancer to route and protect HTTP/HTTPS (layer 7) traffic across the tasks in your service. For more information, see the topic Service Load Balancing in the *Amazon Elastic Container Service Developer Guide*.

Conditions

Conditions define the basic characteristics that you want AWS WAF Classic to watch for in web requests:

- Scripts that are likely to be malicious. Attackers embed scripts that can exploit vulnerabilities in web applications. This is known as *cross-site scripting*.
- IP addresses or address ranges that requests originate from.
- Country or geographical location that requests originate from.
- Length of specified parts of the request, such as the query string.
- SQL code that is likely to be malicious. Attackers try to extract data from your database by embedding malicious SQL code in a web request. This is known as *SQL injection*.
- Strings that appear in the request, for example, values that appear in the `User-Agent` header or text strings that appear in the query string. You can also use regular expressions (regex) to specify these strings.

Some conditions take multiple values. For example, you can specify up to 10,000 IP addresses or IP address ranges in an IP condition.

Rules

You combine conditions into rules to precisely target the requests that you want to allow, block, or count. AWS WAF Classic provides two types of rules:

Regular rule

Regular rules use only conditions to target specific requests. For example, based on recent requests that you've seen from an attacker, you might create a rule that includes the following conditions:

- The requests come from 192.0.2.44.
- They contain the value `BadBot` in the `User-Agent` header.

- They appear to include SQL-like code in the query string.

When a rule includes multiple conditions, as in this example, AWS WAF Classic looks for requests that match all conditions—that is, it ANDs the conditions together.

Add at least one condition to a regular rule. A regular rule without conditions can't match any requests, so the rule's action (allow, count, or block) is never triggered.

Rate-based rule

Rate-based rules are like regular rules with an added rate limit. A rate-based rule counts the requests that arrive from IP addresses that satisfy the rule's conditions. If the requests from an IP address exceed the rate limit in a five-minute period, the rule can trigger an action. It can take a minute or two for the action to trigger.

Conditions are optional for rate-based rules. If you don't add any conditions in a rate-based rule, the rate limit applies to all IP addresses. If you combine conditions with the rate limit, the rate limit applies to IP addresses that match the conditions.

For example, based on recent requests that you've seen from an attacker, you might create a rate-based rule that includes the following conditions:

- The requests come from 192.0.2.44.
- They contain the value `BadBot` in the `User-Agent` header.

In this rate-based rule, you also define a rate limit. In this example, let's say that you create a rate limit of 1,000. Requests that meet both of the preceding conditions and exceed 1,000 requests per five minutes trigger the rule's action (block or count), which is defined in the web ACL.

Requests that don't meet both conditions aren't counted towards the rate limit and aren't affected by this rule.

As a second example, suppose that you want to limit requests to a particular page on your website. To do this, you could add the following string match condition to a rate-based rule:

- The **Part of the request to filter on** is `URI`.
- The **Match Type** is `Starts with`.
- A **Value to match** is `login`.

Further, you specify a `RateLimit` of 1,000.

By adding this rate-based rule to a web ACL, you could limit requests to your login page without affecting the rest of your site.

Web ACLs

After you combine your conditions into rules, you combine the rules into a web ACL. This is where you define an action for each rule—allow, block, or count—and a default action:

An action for each rule

When a web request matches all the conditions in a rule, AWS WAF Classic can either block the request or allow the request to be forwarded to the API Gateway API, CloudFront distribution or an Application Load Balancer. You specify the action that you want AWS WAF Classic to perform for each rule.

AWS WAF Classic compares a request with the rules in a web ACL in the order in which you listed the rules. AWS WAF Classic then takes the action that is associated with the first rule that the request matches. For example, if a web request matches one rule that allows requests and another rule that blocks requests, AWS WAF Classic will either allow or block the request depending on which rule is listed first.

If you want to test a new rule before you start using it, you also can configure AWS WAF Classic to count the requests that meet all the conditions in the rule. As with rules that allow or block requests, a rule that counts requests is affected by its position in the list of rules in the web ACL. For example, if a web request matches a rule that allows requests and another rule that counts requests, and if the rule that allows requests is listed first, the request isn't counted.

A default action

The default action determines whether AWS WAF Classic allows or blocks a request that doesn't match all the conditions in any of the rules in the web ACL. For example, suppose you create a web ACL and add only the rule that you defined before:

- The requests come from 192.0.2.44.
- They contain the value `BadBot` in the `User-Agent` header.
- They appear to include malicious SQL code in the query string.

If a request doesn't meet all three conditions in the rule and if the default action is `ALLOW`, AWS WAF Classic forwards the request to API Gateway, CloudFront or an Application Load Balancer, and the service responds with the requested object.

If you add two or more rules to a web ACL, AWS WAF Classic performs the default action only if a request doesn't satisfy all the conditions in any of the rules. For example, suppose you add a second rule that contains one condition:

- Requests that contain the value `BIGBadBot` in the `User-Agent` header.

AWS WAF Classic performs the default action only when a request doesn't meet all three conditions in the first rule and doesn't meet the one condition in the second rule.

On some occasions, AWS WAF might encounter an internal error that delays the response to Amazon API Gateway, Amazon CloudFront or an Application Load Balancer about whether to allow or block a request. On those occasions CloudFront will typically allow the request or serve the content. API Gateway and an Application Load Balancer typically will deny the request and not serve the content.

AWS WAF Classic pricing

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

With AWS WAF Classic, you pay only for the web ACLs and rules that you create, and for the number of HTTP requests that AWS WAF Classic inspects. For more information, see AWS WAF Classic Pricing.

Getting started with AWS WAF Classic

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

This tutorial shows how to use AWS WAF Classic to perform the following tasks:

- Set up AWS WAF Classic.
- Create a web access control list (web ACL) using the AWS WAF Classic console, and specify the conditions that you want to use to filter web requests. For example, you can specify the IP addresses that the requests originate from and values in the request that are used only by attackers.
- Add the conditions to a rule. Rules let you target the web requests that you want to block or allow. A web request must match all the conditions in a rule before AWS WAF Classic blocks or allows requests based on the conditions that you specify.
- Add the rules to your web ACL. This is where you specify whether you want to block web requests or allow them based on the conditions that you add to each rule.
- Specify a default action, either block or allow. This is the action that AWS WAF Classic takes when a web request doesn't match any of your rules.
- Choose the Amazon CloudFront distribution that you want AWS WAF Classic to inspect web requests for. This tutorial covers the steps only for CloudFront, but the process for an Application Load Balancer and Amazon API Gateway APIs essentially is the same. AWS WAF Classic for CloudFront is available for all AWS Regions. AWS WAF Classic for use with API Gateway or an Application Load Balancer is available in the Regions listed at AWS service endpoints.

> **Note**
>
> AWS typically bills you less than US $0.25 per day for the resources that you create during this tutorial. When you're finished with the tutorial, we recommend that you delete the resources to prevent incurring unnecessary charges.

Topics

- Step 1: Set up AWS WAF Classic
- Step 2: Create a Web ACL
- Step 3: Create an IP match condition
- Step 4: Create a geo match condition
- Step 5: Create a string match condition
- Step 5A: Create a regex condition (optional)
- Step 6: Create a SQL injection match condition
- Step 7: (Optional) create additional conditions
- Step 8: Create a rule and add conditions
- Step 9: Add the rule to a Web ACL
- Step 10: Clean up your resources

Step 1: Set up AWS WAF Classic

If you haven't already followed the general setup steps in Setting up AWS WAF Classic, do that now.

Step 2: Create a Web ACL

The AWS WAF Classic console guides you through the process of configuring AWS WAF Classic to block or allow web requests based on conditions that you specify, such as the IP addresses that the requests originate from or values in the requests. In this step, you create a web ACL.

To create a web ACL

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

If you see **Switch to AWS WAF Classic** in the navigation pane, select it.

2. If this is your first time using AWS WAF Classic, choose **Go to AWS WAF Classic**, and then choose **Configure web ACL**.

   If you've used AWS WAF Classic before, choose **Web ACLs** in the navigation pane, and then choose **Create web ACL**.

3. On the **Name web ACL** page, for **Web ACL name**, enter a name.

   > **Note**
   >
   > You can't change the name after you create the web ACL.

4. For **CloudWatch metric name**, enter a name. The name can contain only alphanumeric characters (A-Z, a-z, 0-9). It can't contain white space.

   > **Note**
   >
   > You can't change the name after you create the web ACL.

5. For **Region**, choose a Region. If you will associate this web ACL with a CloudFront distribution, choose **Global (CloudFront)**.

6. For **AWS resource to associate**, choose the resource that you want to associate with your web ACL, and then choose **Next**.

Step 3: Create an IP match condition

An IP match condition specifies the IP addresses or IP address ranges that requests originate from. In this step, you create an IP match condition. In a later step, you specify whether you want to allow requests or block requests that originate from the specified IP addresses.

> **Note**
>
> For more information about IP match conditions, see Working with IP match conditions.

To create an IP match condition

1. On the **Create conditions** page, for **IP match conditions**, choose **Create condition**.
2. In the **Create IP match condition** dialog box, for **Name**, enter a name. The name can contain only alphanumeric characters (A-Z, a-z, 0-9) or the following special characters: _-!"#`+*},./ .
3. For **Address**, enter **192.0.2.0/24**. This IP address range, specified in CIDR notation, includes the IP addresses from 192.0.2.0 to 192.0.2.255. (The 192.0.2.0/24 IP address range is reserved for examples, so no web requests will originate from these IP addresses.)

   AWS WAF Classic supports IPv4 address ranges: /8 and any range between /16 through /32. AWS WAF Classic supports IPv6 address ranges: /24, /32, /48, /56, /64, and /128. (To specify a single IP address, such as 192.0.2.44, enter **192.0.2.44/32**.) Other ranges aren't supported.

   For more information about CIDR notation, see the Wikipedia article Classless Inter-Domain Routing.
4. Choose **Create**.

Step 4: Create a geo match condition

A geo match condition specifies the country or countries that requests originate from. In this step, you create a geo match condition. In a later step, you specify whether you want to allow requests or block requests that originate from the specified countries.

> **Note**
>
> For more information about geo match conditions, see Working with geographic match conditions.

To create a geo match condition

1. On the **Create conditions** page, for **Geo match conditions**, choose **Create condition**.
2. In the **Create geo match condition** dialog box, for **Name**, enter a name. The name can contain only alphanumeric characters (A-Z, a-z, 0-9) or the following special characters: _-!"#`+*},./ .
3. Choose a **Location type** and a country. Currently, **Location type** can only be **Country**.
4. Choose **Add location**.
5. Choose **Create**.

Step 5: Create a string match condition

A string match condition identifies the strings that you want AWS WAF Classic to search for in a request, such as a specified value in a header or in a query string. Usually, a string consists of printable ASCII characters, but you can specify any character from hexadecimal 0x00 to 0xFF (decimal 0 to 255). In this step, you create a string match condition. In a later step, you specify whether you want to allow or block requests that contain the specified strings.

> **Note**
>
> For more information about string match conditions, see Working with string match conditions.

To create a string match condition

1. On the **Create conditions** page, for **String and regex match conditions**, choose **Create condition**.
2. In the **Create string match condition** dialog box, enter the following values:

   Name

   Enter a name. The name can contain only alphanumeric characters (A-Z, a-z, 0-9) or the following special characters: _-!"#`+*},./ .

   Type

   Choose **String match**.

   Part of the request to filter on

   Choose the part of the web request that you want AWS WAF Classic to inspect for a specified string.

   For this example, choose **Header**.

   > **Note**
   >
   > If you choose **Body** for the value of **Part of the request to filter on**, AWS WAF Classic inspects only the first 8192 bytes (8 KB) because CloudFront forwards only the first 8192 bytes for inspection. To allow or block requests for which the

body is longer than 8192 bytes, you can create a size constraint condition. (AWS WAF Classic gets the length of the body from the request headers.) For more information, see Working with size constraint conditions.

Header (Required if "Part of the request to filter on" is "Header")

Because you chose **Header** for **Part of the request to filter on**, you must specify which header you want AWS WAF Classic to inspect. Enter **User-Agent**. (This value is not case sensitive.)

Match type

Choose where the specified string must appear in the **User-Agent** header, for example, at the beginning, at the end, or anywhere in the string.

For this example, choose **Exactly matches**, which indicates that AWS WAF Classic inspects web requests for a header value that is identical to the value that you specify.

Transformation

In an effort to bypass AWS WAF Classic, attackers use unusual formatting in web requests, for example, by adding white space or by URL-encoding some or all of the request. Transformations convert the web request to a more standard format by removing white space, by URL-decoding the request, or by performing other operations that eliminate much of the unusual formatting that attackers commonly use.

You can only specify a single type of text transformation.

For this example, choose **None**.

Value is base64 encoded

When the value that you enter in **Value to match** is already base64-encoded, select this check box.

For this example, don't select the check box.

Value to match

Specify the value that you want AWS WAF Classic to search for in the part of web requests that you indicated in **Part of the request to filter on**.

For this example, enter **BadBot**. AWS WAF Classic will inspect the `User-Agent` header in web requests for the value **BadBot**.

The maximum length of **Value to match** is 50 characters. If you want to specify a base64-encoded value, you can provide up to 50 characters before encoding.

3. If you want AWS WAF Classic to inspect web requests for multiple values, such as a `User-Agent` header that contains `BadBot` and a query string that contains `BadParameter`, you have two choices:

   - If you want to allow or block web requests only when they contain both values (`AND`), you create one string match condition for each value.
   - If you want to allow or block web requests when they contain either value or both (`OR`), you add both values to the same string match condition.

   For this example, choose **Create**.

Step 5A: Create a regex condition (optional)

A regular expression condition is a type of string match condition and similar in that it identifies the strings that you want AWS WAF Classic to search for in a request, such as a specified value in a header or in a query string. The primary difference is that you use a regular expression (regex) to specify the string pattern that you want AWS WAF Classic to search for. In this step, you create a regex match condition. In a later step, you specify whether you want to allow or block requests that contain the specified strings.

> **Note**
>
> For more information about regex match conditions, see Working with regex match conditions.

To create a regex match condition

1. On the **Create conditions** page, for **String match and regex conditions**, choose **Create condition**.
2. In the **Create string match condition** dialog box, enter the following values:

Name

Enter a name. The name can contain only alphanumeric characters (A-Z, a-z, 0-9) or the following special characters: _-!"#`+*},./ .

Type

Choose **Regex match**.

Part of the request to filter on

Choose the part of the web request that you want AWS WAF Classic to inspect for a specified string.

For this example, choose **Body**.

> **Note**
>
> If you choose **Body** for the value of **Part of the request to filter on**, AWS WAF Classic inspects only the first 8192 bytes (8 KB) because CloudFront forwards only the first 8192 bytes for inspection. To allow or block requests for which the body is longer than 8192 bytes, you can create a size constraint condition. (AWS WAF Classic gets the length of the body from the request headers.) For more information, see Working with size constraint conditions.

Transformation

In an effort to bypass AWS WAF Classic, attackers use unusual formatting in web requests, for example, by adding white space or by URL-encoding some or all of the request. Transformations convert the web request to a more standard format by removing white space, by URL-decoding the request, or by performing other operations that eliminate much of the unusual formatting that attackers commonly use.

You can only specify a single type of text transformation.

For this example, choose **None**.

Regex patterns to match to request

Choose **Create regex pattern set**.

New pattern set name

Enter a name and then specify the regex pattern that you want AWS WAF Classic to search for.

Next, enter the regular expression **I[a@]mAB[a@]dRequest**. AWS WAF Classic will inspect the `User-Agent` header in web requests for the values:

- **IamABadRequest**
- **IamAB@dRequest**
- **I@mABadRequest**
- **I@mAB@dRequest**

3. Choose **Create pattern set and add filter**.
4. Choose **Create**.

Step 6: Create a SQL injection match condition

A SQL injection match condition identifies the part of web requests, such as a header or a query string, that you want AWS WAF Classic to inspect for malicious SQL code. Attackers use SQL queries to extract data from your database. In this step, you create a SQL injection match condition. In a later step, you specify whether you want to allow requests or block requests that appear to contain malicious SQL code.

> **Note**
>
> For more information about string match conditions, see Working with SQL injection match conditions.

To create a SQL injection match condition

1. On the **Create conditions** page, for **SQL injection match conditions**, choose **Create condition**.
2. In the **Create SQL injection match condition** dialog box, enter the following values:

Name

Enter a name.

Part of the request to filter on

Choose the part of web requests that you want AWS WAF Classic to inspect for malicious SQL code.

For this example, choose **Query string**.

> **Note**
>
> If you choose **Body** for the value of **Part of the request to filter on**, AWS WAF Classic inspects only the first 8192 bytes (8 KB) because CloudFront forwards only the first 8192 bytes for inspection. To allow or block requests for which the body is longer than 8192 bytes, you can create a size constraint condition. (AWS WAF Classic gets the length of the body from the request headers.) For more information, see Working with size constraint conditions.

Transformation

For this example, choose **URL decode**.

Attackers use unusual formatting, such as URL encoding, in an effort to bypass AWS WAF Classic. The **URL decode** option eliminates some of that formatting in the web request before AWS WAF Classic inspects the request.

You can only specify a single type of text transformation.

3. Choose **Create**.
4. Choose **Next**.

Step 7: (Optional) create additional conditions

AWS WAF Classic includes other conditions, including the following:

- **Size constraint conditions** – Identifies the part of web requests, such as a header or a query string, that you want AWS WAF Classic to check for length. For more information, see Working with size constraint conditions.

- **Cross-site scripting match conditions** – Identifies the part of web requests, such as a header or a query string, that you want AWS WAF to inspect for malicious scripts. For more information, see Working with cross-site scripting match conditions.

You can optionally create these conditions now, or you can skip to Step 8: Create a rule and add conditions.

Step 8: Create a rule and add conditions

You create a rule to specify the conditions that you want AWS WAF Classic to search for in web requests. If you add more than one condition to a rule, a web request must match all the conditions in the rule for AWS WAF Classic to allow or block requests based on that rule.

> **Note**
>
> For more information about rules, see Working with rules.

To create a rule and add conditions

1. On the **Create rules** page, choose **Create rule**.
2. In the **Create rule** dialog box, enter the following values:

   Name

   Enter a name.

   CloudWatch metric name

   Enter a name for the CloudWatch metric that AWS WAF Classic will create and will associate with the rule. The name can contain only alphanumeric characters (A-Z, a-z, 0-9). It can't contain white space.

   Rule type

   Choose either **Regular rule** or **Rate-based rule**. Rate-based rules are identical to regular rules but also take into account how many requests arrive from the identified IP address in any five-minute period. For more information about the rule types, see How AWS WAF Classic works. For this example, choose `Regular rule`.

Rate limit

For a rate-based rule, enter the maximum number of requests to allow in any five-minute period from an IP address that matches the rule's conditions.

3. For the first condition that you want to add to the rule, specify the following settings:

   - Choose whether you want AWS WAF Classic to allow or block requests based on whether a web request does or does not match the settings in the condition.

     For this example, choose **does**.
   - Choose the type of condition that you want to add to the rule: an IP match set condition, a string match set condition, or a SQL injection match set condition.

     For this example, choose **originate from IP addresses in**.
   - Choose the condition that you want to add to the rule.

     For this example, choose the IP match condition that you created in previous tasks.
4. Choose **Add condition**.
5. Add the geo match condition that you created earlier. Specify the following values:

   - **When a request does**
   - **originate from a geographic location in**
   - Choose your geo match condition.
6. Choose **Add another condition**.
7. Add the string match condition that you created earlier. Specify the following values:

   - **When a request does**
   - **match at least one of the filters in the string match condition**
   - Choose your string match condition.
8. Choose **Add condition**.
9. Add the SQL injection match condition that you created earlier. Specify the following values:

   - **When a request does**
   - **match at least one of the filters in the SQL injection match condition**
   - Choose your SQL injection match condition.
10. Choose **Add condition**.

11. Add the size constraint condition that you created earlier. Specify the following values:
    - **When a request does**
    - **match at least one of the filters in the size constraint condition**
    - Choose your size constraint condition.
12. If you created any other conditions, such as a regex condition, add those in a similar manner.
13. Choose **Create**.
14. For the **Default action**, choose **Allow all requests that don't match any rules**.
15. Choose **Review and create**.

Step 9: Add the rule to a Web ACL

When you add the rule to a web ACL, you specify the following settings:

- The action that you want AWS WAF Classic to take on web requests that match all the conditions in the rule: allow, block, or count the requests.
- The default action for the web ACL. This is the action that you want AWS WAF Classic to take on web requests that *do not* match all the conditions in the rule: allow or block the requests.

AWS WAF Classic starts blocking CloudFront web requests that match all the following conditions (and any others you might have added):

- The value of the `User-Agent` header is `BadBot`
- (If you created and added the regex condition) The value of the `Body` is any of the four strings that matches the pattern `I[a@]mAB[a@]dRequest`
- The requests originate from IP addresses in the range 192.0.2.0-192.0.2.255
- The requests originate from the country that you selected in your geo match condition
- The requests appear to include malicious SQL code in the query string

AWS WAF Classic allows CloudFront to respond to any requests that don't meet all three of these conditions.

Step 10: Clean up your resources

You've now successfully completed the tutorial. To prevent your account from accruing additional AWS WAF Classic charges, you should clean up the AWS WAF Classic objects that you created. Alternatively, you can change the configuration to match the web requests that you really want to allow, block, and count.

> **Note**
>
> AWS typically bills you less than US $0.25 per day for the resources that you create during this tutorial. When you're finished, we recommend that you delete the resources to prevent incurring unnecessary charges.

To delete the objects that AWS WAF Classic charges for

1. Disassociate your web ACL from your CloudFront distribution:

   a. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

      If you see **Switch to AWS WAF Classic** in the navigation pane, select it.

   b. Choose the name of the web ACL that you want to delete. This opens a page with the web ACL's details in the right pane.

   c. In the right pane, on the **Rules** tab, go to the **AWS resources using this web ACL** section. For the CloudFront distribution that you associated the web ACL with, choose the **x** in the **Type** column.

2. Remove the conditions from your rule:

   a. In the navigation pane, choose **Rules**.

   b. Choose the rule that you created during the tutorial.

   c. Choose **Edit rule**.

   d. Choose the **x** at the right of each condition heading.

   e. Choose **Update**.

3. Remove the rule from your web ACL, and delete the web ACL:

   a. In the navigation pane, choose **Web ACLs**.

b. Choose the name of the web ACL that you created during the tutorial. This opens a page with the web ACL's details in the right pane.

c. On the **Rules** tab, choose **Edit web ACL**.

d. Choose the **x** at the right of the rule heading.

e. Choose **Actions**, and then choose **Delete web ACL**.

4. Delete your rule:

a. In the navigation pane, choose **Rules**.

b. Choose the rule that you created during the tutorial.

c. Choose **Delete**.

d. In the **Delete** dialog box, choose **Delete** again to confirm.

AWS WAF Classic doesn't charge for conditions, but if you want to complete the cleanup, perform the following procedure to remove filters from conditions and delete the conditions.

To delete filters and conditions

1. Delete the IP address range in your IP match condition, and delete the IP match condition:

a. In the navigation pane of the AWS WAF Classic console, choose **IP addresses**.

b. Choose the IP match condition that you created during the tutorial.

c. Select the check box for the IP address range that you added.

d. Choose **Delete IP address or range**.

e. In the **IP match conditions** pane, choose **Delete**.

f. In the **Delete** dialog box, choose **Delete** again to confirm.

2. Delete the filter in your SQL injection match condition, and delete the SQL injection match condition:

a. In the navigation pane, choose **SQL injection**.

b. Choose the SQL injection match condition that you created during the tutorial.

c. Select the check box for the filter that you added.

d. Choose **Delete filter**.

e. In the **SQL injection match conditions** pane, choose **Delete**.

f. In the **Delete** dialog box, choose **Delete** again to confirm.

3. Delete the filter in your string match condition, and delete the string match condition:

    a. In the navigation pane, choose **String and regex matching**.

    b. Choose the string match condition that you created during the tutorial.

    c. Select the check box for the filter that you added.

    d. Choose **Delete filter**.

    e. In the **String match conditions** pane, choose **Delete**.

    f. In the **Delete** dialog box, choose **Delete** again to confirm.

4. If you created one, delete the filter in your regex match condition, and delete the regex match condition:

    a. In the navigation pane, choose **String and regex matching**.

    b. Choose the regex match condition that you created during the tutorial.

    c. Select the check box for the filter that you added.

    d. Choose **Delete filter**.

    e. In the **Regex match conditions** pane, choose **Delete**.

    f. In the **Delete** dialog box, choose **Delete** again to confirm.

5. Delete the filter in your size constraint condition, and delete the size constraint condition:

    a. In the navigation pane, choose **Size constraints**.

    b. Choose the size constraint condition that you created during the tutorial.

    c. Select the check box for the filter that you added.

    d. Choose **Delete filter**.

    e. In the **Size constraint conditions** pane, choose **Delete**.

    f. In the **Delete** dialog box, choose **Delete** again to confirm.

Creating and configuring a Web Access Control List (Web ACL)

> **Warning**
>
> AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

> **Note**
>
> This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
> **For the latest version of AWS WAF**, see AWS WAF.

A web access control list (web ACL) gives you fine-grained control over the web requests that your Amazon API Gateway API, Amazon CloudFront distribution or Application Load Balancer responds to. You can allow or block the following types of requests:

- Originate from an IP address or a range of IP addresses
- Originate from a specific country or countries
- Contain a specified string or match a regular expression (regex) pattern in a particular part of requests
- Exceed a specified length
- Appear to contain malicious SQL code (known as SQL injection)
- Appear to contain malicious scripts (known as cross-site scripting)

You can also test for any combination of these conditions, or block or count web requests that not only meet the specified conditions, but also exceed a specified number of requests in any 5-minute period.

To choose the requests that you want to allow to have access to your content or that you want to block, perform the following tasks:

1. Choose the default action, allow or block, for web requests that don't match any of the conditions that you specify. For more information, see Deciding on the default action for a Web ACL.
2. Specify the conditions under which you want to allow or block requests:
   - To allow or block requests based on whether the requests appear to contain malicious scripts, create cross-site scripting match conditions. For more information, see Working with cross-site scripting match conditions.

- To allow or block requests based on the IP addresses that they originate from, create IP match conditions. For more information, see Working with IP match conditions.
- To allow or block requests based on the country that they originate from, create geo match conditions. For more information, see Working with geographic match conditions.
- To allow or block requests based on whether the requests exceed a specified length, create size constraint conditions. For more information, see Working with size constraint conditions.
- To allow or block requests based on whether the requests appear to contain malicious SQL code, create SQL injection match conditions. For more information, see Working with SQL injection match conditions.
- To allow or block requests based on strings that appear in the requests, create string match conditions. For more information, see Working with string match conditions.
- To allow or block requests based on a regex pattern that appear in the requests, create regex match conditions. For more information, see Working with regex match conditions.

3. Add the conditions to one or more rules. If you add more than one condition to the same rule, web requests must match all the conditions for AWS WAF Classic to allow or block requests based on the rule. For more information, see Working with rules. Optionally, you can use a rate-based rule instead of a regular rule to limit the number of requests from any IP address that meets the conditions.
4. Add the rules to a web ACL. For each rule, specify whether you want AWS WAF Classic to allow or block requests based on the conditions that you added to the rule. If you add more than one rule to a web ACL, AWS WAF Classic evaluates the rules in the order that they're listed in the web ACL. For more information, see Working with web ACLs.

   When you add a new rule or update existing rules, it can take up to one minute for those changes to appear and be active across your web ACLs and resources.

Topics

- Working with conditions
- Working with rules
- Working with web ACLs

Working with conditions

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

Conditions specify when you want to allow or block requests.

- To allow or block requests based on whether the requests appear to contain malicious scripts, create cross-site scripting match conditions. For more information, see Working with cross-site scripting match conditions.
- To allow or block requests based on the IP addresses that they originate from, create IP match conditions. For more information, see Working with IP match conditions.
- To allow or block requests based on the country that they originate from, create geo match conditions. For more information, see Working with geographic match conditions.
- To allow or block requests based on whether the requests exceed a specified length, create size constraint conditions. For more information, see Working with size constraint conditions.
- To allow or block requests based on whether the requests appear to contain malicious SQL code, create SQL injection match conditions. For more information, see Working with SQL injection match conditions.
- To allow or block requests based on strings that appear in the requests, create string match conditions. For more information, see Working with string match conditions.
- To allow or block requests based on a regex pattern that appear in the requests, create regex match conditions. For more information, see Working with regex match conditions.

Topics

- Working with cross-site scripting match conditions
- Working with IP match conditions
- Working with geographic match conditions
- Working with size constraint conditions
- Working with SQL injection match conditions
- Working with string match conditions
- Working with regex match conditions

Working with cross-site scripting match conditions

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

Attackers sometimes insert scripts into web requests in an effort to exploit vulnerabilities in web applications. You can create one or more cross-site scripting match conditions to identify the parts of web requests, such as the URI or the query string, that you want AWS WAF Classic to inspect for possible malicious scripts. Later in the process, when you create a web ACL, you specify whether to allow or block requests that appear to contain malicious scripts.

Topics

- Creating cross-site scripting match conditions
- Values that you specify when you create or edit cross-site scripting match conditions

- Adding and deleting filters in a cross-site scripting match condition
- Deleting cross-site scripting match conditions

Creating cross-site scripting match conditions

When you create cross-site scripting match conditions, you specify filters. The filters indicate the part of web requests that you want AWS WAF Classic to inspect for malicious scripts, such as the URI or the query string. You can add more than one filter to a cross-site scripting match condition, or you can create a separate condition for each filter. Here's how each configuration affects AWS WAF Classic behavior:

- **More than one filter per cross-site scripting match condition (recommended)** – When you add a cross-site scripting match condition that contains multiple filters to a rule and add the rule to a web ACL, a web request must match only one of the filters in the cross-site scripting match condition for AWS WAF Classic to allow or block the request based on that condition.

  For example, suppose you create one cross-site scripting match condition, and the condition contains two filters. One filter instructs AWS WAF Classic to inspect the URI for malicious scripts, and the other instructs AWS WAF Classic to inspect the query string. AWS WAF Classic allows or blocks requests if they appear to contain malicious scripts *either* in the URI *or* in the query string.
- **One filter per cross-site scripting match condition** – When you add the separate cross-site scripting match conditions to a rule and add the rule to a web ACL, web requests must match all the conditions for AWS WAF Classic to allow or block requests based on the conditions.

  Suppose you create two conditions, and each condition contains one of the two filters in the preceding example. When you add both conditions to the same rule and add the rule to a web ACL, AWS WAF Classic allows or blocks requests only when both the URI and the query string appear to contain malicious scripts.

> **Note**
>
> When you add a cross-site scripting match condition to a rule, you also can configure AWS WAF Classic to allow or block web requests that *do not* appear to contain malicious scripts.

To create a cross-site scripting match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **Cross-site scripting**.
3. Choose **Create condition**.
4. Specify the applicable filter settings. For more information, see Values that you specify when you create or edit cross-site scripting match conditions.
5. Choose **Add another filter**.
6. If you want to add another filter, repeat steps 4 and 5.
7. When you're done adding filters, choose **Create**.

Values that you specify when you create or edit cross-site scripting match conditions

When you create or update a cross-site scripting match condition, you specify the following values:

Name

The name of the cross-site scripting match condition.

The name can contain only the characters A-Z, a-z, 0-9, and the special characters: _-!"#`+*},./ . You can't change the name of a condition after you create it.

Part of the request to filter on

Choose the part of each web request that you want AWS WAF Classic to inspect for malicious scripts:

Header

A specified request header, for example, the `User-Agent` or `Referer` header. If you choose **Header**, specify the name of the header in the **Header** field.

HTTP method

The HTTP method, which indicates the type of operation that the request is asking the origin to perform. CloudFront supports the following methods: `DELETE`, `GET`, `HEAD`, `OPTIONS`, `PATCH`, `POST`, and `PUT`.

Query string

The part of a URL that appears after a ? character, if any.

> **Note**
>
> For cross-site scripting match conditions, we recommend that you choose **All query parameters (values only)** instead of **Query string** for **Part of the request to filter on**.

URI

The URI path of the request, which identifies the resource, for example, `/images/daily-ad.jpg`. This doesn't include the query string or fragment components of the URI. For information, see Uniform Resource Identifier (URI): Generic Syntax.

Unless a **Transformation** is specified, a URI is not normalized and is inspected just as AWS receives it from the client as part of the request. A **Transformation** will reformat the URI as specified.

Body

The part of a request that contains any additional data that you want to send to your web server as the HTTP request body, such as data from a form.

> **Note**
>
> If you choose **Body** for the value of **Part of the request to filter on**, AWS WAF Classic inspects only the first 8192 bytes (8 KB). To allow or block requests for which the body is longer than 8192 bytes, you can create a size constraint condition. (AWS WAF Classic gets the length of the body from the request headers.) For more information, see Working with size constraint conditions.

Single query parameter (value only)

Any parameter that you have defined as part of the query string. For example, if the URL is "www.xyz.com?UserName=abc&SalesRegion=seattle" you can add a filter to either the *UserName* or *SalesRegion* parameter.

If you choose **Single query parameter (value only)**, you will also specify a **Query parameter name**. This is the parameter in the query string that you will inspect, such as *UserName* or *SalesRegion*. The maximum length for **Query parameter name** is 30 characters. **Query parameter name** is not case sensitive. For example, it you specify *UserName* as the **Query parameter name**, this will match all variations of *UserName*, such as *username* and *UsERName*.

All query parameters (values only)

Similar to **Single query parameter (value only)**, but rather than inspecting the values of a single parameter, AWS WAF Classic inspects all parameter values within the query string for possible malicious scripts. For example, if the URL is "www.xyz.com?UserName=abc&SalesRegion=seattle," and you choose **All query parameters (values only)**, AWS WAF Classic will trigger a match if either the value of *UserName* or *SalesRegion* contain possible malicious scripts.

Header

If you chose **Header** for **Part of the request to filter on**, choose a header from the list of common headers, or enter the name of a header that you want AWS WAF Classic to inspect for malicious scripts.

Transformation

A transformation reformats a web request before AWS WAF Classic inspects the request. This eliminates some of the unusual formatting that attackers use in web requests in an effort to bypass AWS WAF Classic.

You can only specify a single type of text transformation.

Transformations can perform the following operations:

None

AWS WAF Classic doesn't perform any text transformations on the web request before inspecting it for the string in **Value to match**.

Convert to lowercase

AWS WAF Classic converts uppercase letters (A-Z) to lowercase (a-z).

HTML decode

AWS WAF Classic replaces HTML-encoded characters with unencoded characters:

- Replaces `"` with &

- Replaces ` ` with a non-breaking space
- Replaces `<` with <
- Replaces `>` with >
- Replaces characters that are represented in hexadecimal format, `&#xhhhh;`, with the corresponding characters
- Replaces characters that are represented in decimal format, `&#nnnn;`, with the corresponding characters

Normalize white space

AWS WAF Classic replaces the following characters with a space character (decimal 32):

- \f, formfeed, decimal 12
- \t, tab, decimal 9
- \n, newline, decimal 10
- \r, carriage return, decimal 13
- \v, vertical tab, decimal 11
- non-breaking space, decimal 160

In addition, this option replaces multiple spaces with one space.

Simplify command line

For requests that contain operating system command line commands, use this option to perform the following transformations:

- Delete the following characters: \ " ' ^
- Delete spaces before the following characters: / (
- Replace the following characters with a space: , ;
- Replace multiple spaces with one space
- Convert uppercase letters (A-Z) to lowercase (a-z)

URL decode

Decode a URL-encoded request.

Adding and deleting filters in a cross-site scripting match condition

You can add or delete filters in a cross-site scripting match condition. To change a filter, add a new one and delete the old one.

To add or delete filters in a cross-site scripting match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **Cross-site scripting**.
3. Choose the condition that you want to add or delete filters in.
4. To add filters, perform the following steps:

   a. Choose **Add filter**.

   b. Specify the applicable filter settings. For more information, see Values that you specify when you create or edit cross-site scripting match conditions.

   c. Choose **Add**.
5. To delete filters, perform the following steps:

   a. Select the filter that you want to delete.

   b. Choose **Delete filter**.

Deleting cross-site scripting match conditions

If you want to delete a cross-site scripting match condition, you must first delete all filters in the condition and remove the condition from all the rules that are using it, as described in the following procedure.

To delete a cross-site scripting match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **Cross-site scripting**.
3. In the **Cross-site scripting match conditions** pane, choose the cross-site scripting match condition that you want to delete.
4. In the right pane, choose the **Associated rules** tab.

If the list of rules using this cross-site scripting match condition is empty, go to step 6. If the list contains any rules, make note of the rules, and continue with step 5.

5. To remove the cross-site scripting match condition from the rules that are using it, perform the following steps:

   a. In the navigation pane, choose **Rules**.

   b. Choose the name of a rule that is using the cross-site scripting match condition that you want to delete.

   c. In the right pane, select the cross-site scripting match condition that you want to remove from the rule, and choose **Remove selected condition**.

   d. Repeat steps b and c for all the remaining rules that are using the cross-site scripting match condition that you want to delete.

   e. In the navigation pane, choose **Cross-site scripting**.

   f. In the **Cross-site scripting match conditions** pane, choose the cross-site scripting match condition that you want to delete.

6. Choose **Delete** to delete the selected condition.

Working with IP match conditions

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

If you want to allow or block web requests based on the IP addresses that the requests originate from, create one or more IP match conditions. An IP match condition lists up to 10,000 IP addresses or IP address ranges that your requests originate from. Later in the process, when you create a web ACL, you specify whether to allow or block requests from those IP addresses.

Topics

- Creating an IP Match Condition
- Editing IP match conditions
- Deleting IP match conditions

Creating an IP Match Condition

If you want to allow some web requests and block others based on the IP addresses that the requests originate from, create an IP match condition for the IP addresses that you want to allow and another IP match condition for the IP addresses that you want to block.

> **Note**
>
> When you add an IP match condition to a rule, you also can configure AWS WAF Classic to allow or block web requests that *do not* originate from the IP addresses that you specify in the condition.

To create an IP match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **IP addresses**.
3. Choose **Create condition**.
4. Enter a name in the **Name** field.

   The name can contain only alphanumeric characters (A-Z, a-z, 0-9) or the following special characters: _-!"#`+*},./ . You can't change the name of a condition after you create it.
5. Select the correct IP version and specify an IP address or range of IP addresses by using CIDR notation. Here are some examples:

- To specify the IPv4 address 192.0.2.44, type **192.0.2.44/32**.
- To specify the IPv6 address 0:0:0:0:0:ffff:c000:22c, type **0:0:0:0:0:ffff:c000:22c/128**.
- To specify the range of IPv4 addresses from 192.0.2.0 to 192.0.2.255, type **192.0.2.0/24**.
- To specify the range of IPv6 addresses from 2620:0:2d0:200:0:0:0:0 to 2620:0:2d0:200:ffff:ffff:ffff:ffff, enter **2620:0:2d0:200::/64**.

AWS WAF Classic supports IPv4 address ranges: /8 and any range between /16 through /32. AWS WAF Classic supports IPv6 address ranges: /24, /32, /48, /56, /64, and /128. For more information about CIDR notation, see the Wikipedia entry Classless Inter-Domain Routing.

6. Choose **Add another IP address or range**.
7. If you want to add another IP address or range, repeat steps 5 and 6.
8. When you're finished adding values, choose **Create IP match condition**.

Editing IP match conditions

You can add an IP address range to an IP match condition or delete a range. To change a range, add a new one and delete the old one.

To edit an IP match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **IP addresses**.
3. In the **IP match conditions** pane, choose the IP match condition that you want to edit.
4. To add an IP address range:

   a. In the right pane, choose **Add IP address or range**.

   b. Select the correct IP version and enter an IP address range by using CIDR notation. Here are some examples:

   - To specify the IPv4 address 192.0.2.44, enter **192.0.2.44/32**.
   - To specify the IPv6 address 0:0:0:0:0:ffff:c000:22c, enter **0:0:0:0:0:ffff:c000:22c/128**.

- To specify the range of IPv4 addresses from 192.0.2.0 to 192.0.2.255, enter **192.0.2.0/24**.
- To specify the range of IPv6 addresses from 2620:0:2d0:200:0:0:0:0 to 2620:0:2d0:200:ffff:ffff:ffff:ffff, enter **2620:0:2d0:200::/64**.

AWS WAF Classic supports IPv4 address ranges: /8 and any range between /16 through /32. AWS WAF Classic supports IPv6 address ranges: /24, /32, /48, /56, /64, and /128. For more information about CIDR notation, see the Wikipedia entry Classless Inter-Domain Routing.

c. To add more IP addresses, choose **Add another IP address** and enter the value.

d. Choose **Add**.

5. To delete an IP address or range:

a. In the right pane, select the values that you want to delete.

b. Choose **Delete IP address or range**.

Deleting IP match conditions

If you want to delete an IP match condition, you must first delete all IP addresses and ranges in the condition and remove the condition from all the rules that are using it, as described in the following procedure.

To delete an IP match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.

2. In the navigation pane, choose **IP addresses**.
3. In the **IP match conditions** pane, choose the IP match condition that you want to delete.
4. In the right pane, choose the **Rules** tab.

   If the list of rules using this IP match condition is empty, go to step 6. If the list contains any rules, make note of the rules, and continue with step 5.

5. To remove the IP match condition from the rules that are using it, perform the following steps:

a. In the navigation pane, choose **Rules**.
b. Choose the name of a rule that is using the IP match condition that you want to delete.
c. In the right pane, select the IP match condition that you want to remove from the rule, and choose **Remove selected condition**.
d. Repeat steps b and c for all the remaining rules that are using the IP match condition that you want to delete.
e. In the navigation pane, choose **IP match conditions**.
f. In the **IP match conditions** pane, choose the IP match condition that you want to delete.

6. Choose **Delete** to delete the selected condition.

Working with geographic match conditions

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

If you want to allow or block web requests based on the country that the requests originate from, create one or more geo match conditions. A geo match condition lists countries that your requests originate from. Later in the process, when you create a web ACL, you specify whether to allow or block requests from those countries.

You can use geo match conditions with other AWS WAF Classic conditions or rules to build sophisticated filtering. For example, if you want to block certain countries, but still allow specific IP addresses from that country, you could create a rule containing a geo match condition and an IP

match condition. Configure the rule to block requests that originate from that country and do not match the approved IP addresses. As another example, if you want to prioritize resources for users in a particular country, you could include a geo match condition in two different rate-based rules. Set a higher rate limit for users in the preferred country and set a lower rate limit for all other users.

> **Note**
>
> If you are using the CloudFront geo restriction feature to block a country from accessing your content, any request from that country is blocked and is not forwarded to AWS WAF Classic. So if you want to allow or block requests based on geography plus other AWS WAF Classic conditions, you should *not* use the CloudFront geo restriction feature. Instead, you should use an AWS WAF Classic geo match condition.

Topics

- Creating a geo match condition
- Editing geo match conditions
- Deleting geo match conditions

Creating a geo match condition

If you want to allow some web requests and block others based on the countries that the requests originate from, create a geo match condition for the countries that you want to allow and another geo match condition for the countries that you want to block.

> **Note**
>
> When you add a geo match condition to a rule, you also can configure AWS WAF Classic to allow or block web requests that *do not* originate from the country that you specify in the condition.

To create a geo match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

If you see **Switch to AWS WAF Classic** in the navigation pane, select it.

2. In the navigation pane, choose **Geo match**.
3. Choose **Create condition**.
4. Enter a name in the **Name** field.

   The name can contain only alphanumeric characters (A-Z, a-z, 0-9) or the following special characters: _-!"#`+*},./ . You can't change the name of a condition after you create it.
5. Choose a **Region**.
6. Choose a **Location type** and a country. **Location type** can currently only be **Country**.
7. Choose **Add location**.
8. Choose **Create**.

Editing geo match conditions

You can add countries to or delete countries from your geo match condition.

To edit a geo match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **Geo match**.
3. In the **Geo match conditions** pane, choose the geo match condition that you want to edit.
4. To add a country:

   a. In the right pane, choose **Add filter**.

   b. Choose a **Location type** and a country. **Location type** can currently only be **Country**.

   c. Choose **Add**.
5. To delete a country:

   a. In the right pane, select the values that you want to delete.

   b. Choose **Delete filter**.

Deleting geo match conditions

If you want to delete a geo match condition, you must first remove all countries in the condition and remove the condition from all the rules that are using it, as described in the following procedure.

To delete a geo match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. Remove the geo match condition from the rules that are using it:
   a. In the navigation pane, choose **Rules**.
   b. Choose the name of a rule that is using the geo match condition that you want to delete.
   c. In the right pane, choose **Edit rule**.
   d. Choose the **X** next to the condition you want to delete.
   e. Choose **Update**.
   f. Repeat for all the remaining rules that are using the geo match condition that you want to delete.
3. Remove the filters from the condition you want to delete:
   a. In the navigation pane, choose **Geo match**.
   b. Choose the name of the geo match condition that you want to delete.
   c. In the right pane, choose the check box next to **Filter** in order to select all of the filters.
   d. Choose the **Delete filter**.
4. In the navigation pane, choose **Geo match**.
5. In the **Geo match conditions** pane, choose the geo match condition that you want to delete.
6. Choose **Delete** to delete the selected condition.

Working with size constraint conditions

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

If you want to allow or block web requests based on the length of specified parts of requests, create one or more size constraint conditions. A size constraint condition identifies the part of web requests that you want AWS WAF Classic to look at, the number of bytes that you want AWS WAF Classic to look for, and an operator, such as greater than (>) or less than (<). For example, you can use a size constraint condition to look for query strings that are longer than 100 bytes. Later in the process, when you create a web ACL, you specify whether to allow or block requests based on those settings.

Note that if you configure AWS WAF Classic to inspect the request body, for example, by searching the body for a specified string, AWS WAF Classic inspects only the first 8192 bytes (8 KB). If the request body for your web requests will never exceed 8192 bytes, you can create a size constraint condition and block requests that have a request body greater than 8192 bytes.

Topics

- Creating size constraint conditions
- Values that you specify when you create or edit size constraint conditions
- Adding and deleting filters in a size constraint condition
- Deleting size constraint conditions

Creating size constraint conditions

When you create size constraint conditions, you specify filters that identify the part of web requests for which you want AWS WAF Classic to evaluate the length. You can add more than one filter to a size constraint condition, or you can create a separate condition for each filter. Here's how each configuration affects AWS WAF Classic behavior:

- **One filter per size constraint condition** – When you add the separate size constraint conditions to a rule and add the rule to a web ACL, web requests must match all the conditions for AWS WAF Classic to allow or block requests based on the conditions.

  For example, suppose you create two conditions. One matches web requests for which query strings are greater than 100 bytes. The other matches web requests for which the request body is greater than 1024 bytes. When you add both conditions to the same rule and add the rule to a web ACL, AWS WAF Classic allows or blocks requests only when both conditions are true.
- **More than one filter per size constraint condition** – When you add a size constraint condition that contains multiple filters to a rule and add the rule to a web ACL, a web request needs only to match one of the filters in the size constraint condition for AWS WAF Classic to allow or block the request based on that condition.

  Suppose you create one condition instead of two, and the one condition contains the same two filters as in the preceding example. AWS WAF Classic allows or blocks requests if either the query string is greater than 100 bytes or the request body is greater than 1024 bytes.

> **Note**
>
> When you add a size constraint condition to a rule, you also can configure AWS WAF Classic to allow or block web requests that *do not* match the values in the condition.

To create a size constraint condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **Size constraints**.
3. Choose **Create condition**.

4. Specify the applicable filter settings. For more information, see Values that you specify when you create or edit size constraint conditions.
5. Choose **Add another filter**.
6. If you want to add another filter, repeat steps 4 and 5.
7. When you're finished adding filters, choose **Create size constraint condition**.

Values that you specify when you create or edit size constraint conditions

When you create or update a size constraint condition, you specify the following values:

Name

Enter a name for the size constraint condition.

The name can contain only alphanumeric characters (A-Z, a-z, 0-9) or the following special characters: _-!"#`+*},./. You can't change the name of a condition after you create it.

Part of the request to filter on

Choose the part of each web request for which you want AWS WAF Classic to evaluate the length:

Header

A specified request header, for example, the `User-Agent` or `Referer` header. If you choose **Header**, specify the name of the header in the **Header** field.

HTTP method

The HTTP method, which indicates the type of operation that the request is asking the origin to perform. CloudFront supports the following methods: `DELETE`, `GET`, `HEAD`, `OPTIONS`, `PATCH`, `POST`, and `PUT`.

Query string

The part of a URL that appears after a `?` character, if any.

URI

The URI path of the request, which identifies the resource, for example, `/images/daily-ad.jpg`. This doesn't include the query string or fragment components of the URI. For information, see Uniform Resource Identifier (URI): Generic Syntax.

Unless a **Transformation** is specified, a URI is not normalized and is inspected just as AWS receives it from the client as part of the request. A **Transformation** will reformat the URI as specified.

Body

The part of a request that contains any additional data that you want to send to your web server as the HTTP request body, such as data from a form.

Single query parameter (value only)

Any parameter that you have defined as part of the query string. For example, if the URL is "www.xyz.com?UserName=abc&SalesRegion=seattle" you can add a filter to either the *UserName* or *SalesRegion* parameter.

If you choose **Single query parameter (value only)**, you will also specify a **Query parameter name**. This is the parameter in the query string that you will inspect, such as *UserName*. The maximum length for **Query parameter name** is 30 characters. **Query parameter name** is not case sensitive. For example, it you specify *UserName* as the **Query parameter name**, this will match all variations of *UserName*, such as *username* and *UsERName*.

All query parameters (values only)

Similar to **Single query parameter (value only)**, but rather than inspecting the value of a single parameter, AWS WAF Classic inspects the values of all parameters within the query string for the size constraint. For example, if the URL is "www.xyz.com?UserName=abc&SalesRegion=seattle," and you choose **All query parameters (values only)**, AWS WAF Classic will trigger a match the value of if either *UserName* or *SalesRegion* exceed the specified size.

Header (Only When "Part of the request to filter on" is "Header")

If you chose **Header** for **Part of the request to filter on**, choose a header from the list of common headers, or type the name of a header for which you want AWS WAF Classic to evaluate the length.

Comparison operator

Choose how you want AWS WAF Classic to evaluate the length of the query string in web requests with respect to the value that you specify for **Size**.

For example, if you choose **Is greater than** for **Comparison operator** and type **100** for **Size**, AWS WAF Classic evaluates web requests for a query string that is longer than 100 bytes.

Size

Enter the length, in bytes, that you want AWS WAF Classic to watch for in query strings.

> **Note**
>
> If you choose **URI** for the value of **Part of the request to filter on**, the **/** in the URI counts as one character. For example, the URI path `/logo.jpg` is nine characters long.

Transformation

A transformation reformats a web request before AWS WAF Classic evaluates the length of the specified part of the request. This eliminates some of the unusual formatting that attackers use in web requests in an effort to bypass AWS WAF Classic.

> **Note**
>
> If you choose **Body** for **Part of the request to filter on**, you can't configure AWS WAF Classic to perform a transformation because only the first 8192 bytes are forwarded for inspection. However, you can still filter your traffic based on the size of the HTTP request body and specify a transformation of **None**. (AWS WAF Classic gets the length of the body from the request headers.)

You can only specify a single type of text transformation.

Transformations can perform the following operations:

None

AWS WAF Classic doesn't perform any text transformations on the web request before checking the length.

Convert to lowercase

AWS WAF Classic converts uppercase letters (A-Z) to lowercase (a-z).

HTML decode

AWS WAF Classic replaces HTML-encoded characters with unencoded characters:

- Replaces `"` with &

- Replaces   with a non-breaking space
- Replaces < with <
- Replaces > with >
- Replaces characters that are represented in hexadecimal format, &#xhhhh;, with the corresponding characters
- Replaces characters that are represented in decimal format, &#nnnn;, with the corresponding characters

Normalize white space

AWS WAF Classic replaces the following characters with a space character (decimal 32):

- \f, formfeed, decimal 12
- \t, tab, decimal 9
- \n, newline, decimal 10
- \r, carriage return, decimal 13
- \v, vertical tab, decimal 11
- non-breaking space, decimal 160

In addition, this option replaces multiple spaces with one space.

Simplify command line

For requests that contain operating system command line commands, use this option to perform the following transformations:

- Delete the following characters: \ " ' ^
- Delete spaces before the following characters: / (
- Replace the following characters with a space: , ;
- Replace multiple spaces with one space
- Convert uppercase letters (A-Z) to lowercase (a-z)

URL decode

Decode a URL-encoded request.

Adding and deleting filters in a size constraint condition

You can add or delete filters in a size constraint condition. To change a filter, add a new one and delete the old one.

To add or delete filters in a size constraint condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **Size constraint**.
3. Choose the condition that you want to add or delete filters in.
4. To add filters, perform the following steps:

   a. Choose **Add filter**.

   b. Specify the applicable filter settings. For more information, see Values that you specify when you create or edit size constraint conditions.

   c. Choose **Add**.
5. To delete filters, perform the following steps:

   a. Select the filter that you want to delete.

   b. Choose **Delete filter**.

Deleting size constraint conditions

If you want to delete a size constraint condition, you need to first delete all filters in the condition and remove the condition from all the rules that are using it, as described in the following procedure.

To delete a size constraint condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **Size constraints**.
3. In the **Size constraint conditions** pane, choose the size constraint condition that you want to delete.
4. In the right pane, choose the **Associated rules** tab.

If the list of rules using this size constraint condition is empty, go to step 6. If the list contains any rules, make note of the rules, and continue with step 5.

5. To remove the size constraint condition from the rules that are using it, perform the following steps:

   a. In the navigation pane, choose **Rules**.

   b. Choose the name of a rule that is using the size constraint condition that you want to delete.

   c. In the right pane, select the size constraint condition that you want to remove from the rule, and then choose **Remove selected condition**.

   d. Repeat steps b and c for all the remaining rules that are using the size constraint condition that you want to delete.

   e. In the navigation pane, choose **Size constraint**.

   f. In the **Size constraint conditions** pane, choose the size constraint condition that you want to delete.

6. Choose **Delete** to delete the selected condition.

Working with SQL injection match conditions

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

Attackers sometimes insert malicious SQL code into web requests in an effort to extract data from your database. To allow or block web requests that appear to contain malicious SQL code, create one or more SQL injection match conditions. A SQL injection match condition identifies the part of web requests, such as the URI path or the query string, that you want AWS WAF Classic to inspect. Later in the process, when you create a web ACL, you specify whether to allow or block requests that appear to contain malicious SQL code.

Topics

- Creating SQL injection match conditions
- Values that you specify when you create or edit SQL injection match conditions
- Adding and deleting filters in a SQL injection match condition
- Deleting SQL injection match conditions

Creating SQL injection match conditions

When you create SQL injection match conditions, you specify filters, which indicate the part of web requests that you want AWS WAF Classic to inspect for malicious SQL code, such as the URI or the query string. You can add more than one filter to a SQL injection match condition, or you can create a separate condition for each filter. Here's how each configuration affects AWS WAF Classic behavior:

- **More than one filter per SQL injection match condition (recommended)** – When you add a SQL injection match condition containing multiple filters to a rule and add the rule to a web ACL, a web request needs only to match one of the filters in the SQL injection match condition for AWS WAF Classic to allow or block the request based on that condition.

  For example, suppose you create one SQL injection match condition, and the condition contains two filters. One filter instructs AWS WAF Classic to inspect the URI for malicious SQL code, and the other instructs AWS WAF Classic to inspect the query string. AWS WAF Classic allows or blocks requests if they appear to contain malicious SQL code *either* in the URI *or* in the query string.
- **One filter per SQL injection match condition** – When you add the separate SQL injection match conditions to a rule and add the rule to a web ACL, web requests must match all the conditions for AWS WAF Classic to allow or block requests based on the conditions.

  Suppose you create two conditions, and each condition contains one of the two filters in the preceding example. When you add both conditions to the same rule and add the rule to a web

ACL, AWS WAF Classic allows or blocks requests only when both the URI and the query string appear to contain malicious SQL code.

> **Note**
>
> When you add a SQL injection match condition to a rule, you also can configure AWS WAF Classic to allow or block web requests that *do not* appear to contain malicious SQL code.

To create a SQL injection match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **SQL injection**.
3. Choose **Create condition**.
4. Specify the applicable filter settings. For more information, see Values that you specify when you create or edit SQL injection match conditions.
5. Choose **Add another filter**.
6. If you want to add another filter, repeat steps 4 and 5.
7. When you're finished adding filters, choose **Create**.

Values that you specify when you create or edit SQL injection match conditions

When you create or update a SQL injection match condition, you specify the following values:

Name

The name of the SQL injection match condition.

The name can contain only alphanumeric characters (A-Z, a-z, 0-9) or the following special characters: _-!"#`+*},./. You can't change the name of a condition after you create it.

Part of the request to filter on

Choose the part of each web request that you want AWS WAF Classic to inspect for malicious SQL code:

Header

A specified request header, for example, the `User-Agent` or `Referer` header. If you choose **Header**, specify the name of the header in the **Header** field.

HTTP method

The HTTP method, which indicates the type of operation that the request is asking the origin to perform. CloudFront supports the following methods: `DELETE`, `GET`, `HEAD`, `OPTIONS`, `PATCH`, `POST`, and `PUT`.

Query string

The part of a URL that appears after a ? character, if any.

> **Note**
>
> For SQL injection match conditions, we recommend that you choose **All query parameters (values only)** instead of **Query string** for **Part of the request to filter on**.

URI

The URI path of the request, which identifies the resource, for example, `/images/daily-ad.jpg`. This doesn't include the query string or fragment components of the URI. For information, see Uniform Resource Identifier (URI): Generic Syntax.

Unless a **Transformation** is specified, a URI is not normalized and is inspected just as AWS receives it from the client as part of the request. A **Transformation** will reformat the URI as specified.

Body

The part of a request that contains any additional data that you want to send to your web server as the HTTP request body, such as data from a form.

> **Note**
>
> If you choose **Body** for the value of **Part of the request to filter on**, AWS WAF Classic inspects only the first 8192 bytes (8 KB). To allow or block requests for which the body is longer than 8192 bytes, you can create a size constraint condition. (AWS WAF

Classic gets the length of the body from the request headers.) For more information, see Working with size constraint conditions.

Single query parameter (value only)

Any parameter that you have defined as part of the query string. For example, if the URL is "www.xyz.com?UserName=abc&SalesRegion=seattle" you can add a filter to either the *UserName* or *SalesRegion* parameter.

If you choose **Single query parameter (value only)** you will also specify a **Query parameter name**. This is the parameter in the query string that you will inspect, such as *UserName* or *SalesRegion*. The maximum length for **Query parameter name** is 30 characters. **Query parameter name** is not case sensitive. For example, it you specify *UserName* as the **Query parameter name**, this will match all variations of *UserName*, such as *username* and *UsERName*.

All query parameters (values only)

Similar to **Single query parameter (value only)**, but rather than inspecting the value of a single parameter, AWS WAF Classic inspects the value of all parameters within the query string for possible malicious SQL code. For example, if the URL is "www.xyz.com?UserName=abc&SalesRegion=seattle," and you choose **All query parameters (values only)**, AWS WAF Classic will trigger a match if the value of either *UserName* or *SalesRegion* contain possible malicious SQL code.

Header

If you chose **Header** for **Part of the request to filter on**, choose a header from the list of common headers, or enter the name of a header that you want AWS WAF Classic to inspect for malicious SQL code.

Transformation

A transformation reformats a web request before AWS WAF Classic inspects the request. This eliminates some of the unusual formatting that attackers use in web requests in an effort to bypass AWS WAF Classic.

You can only specify a single type of text transformation.

Transformations can perform the following operations:

None

AWS WAF Classic doesn't perform any text transformations on the web request before inspecting it for the string in **Value to match**.

Convert to lowercase

AWS WAF Classic converts uppercase letters (A-Z) to lowercase (a-z).

HTML decode

AWS WAF Classic replaces HTML-encoded characters with unencoded characters:

- Replaces `"` with &
- Replaces ` ` with a non-breaking space
- Replaces `<` with <
- Replaces `>` with >
- Replaces characters that are represented in hexadecimal format, `&#xhhhh;`, with the corresponding characters
- Replaces characters that are represented in decimal format, `&#nnnn;`, with the corresponding characters

Normalize white space

AWS WAF Classic replaces the following characters with a space character (decimal 32):

- \f, formfeed, decimal 12
- \t, tab, decimal 9
- \n, newline, decimal 10
- \r, carriage return, decimal 13
- \v, vertical tab, decimal 11
- non-breaking space, decimal 160

In addition, this option replaces multiple spaces with one space.

Simplify command line

For requests that contain operating system command line commands, use this option to perform the following transformations:

- Delete the following characters: \ " ' ^

- Delete spaces before the following characters: / (
- Replace the following characters with a space: , ;
- Replace multiple spaces with one space
- Convert uppercase letters (A-Z) to lowercase (a-z)

URL decode

Decode a URL-encoded request.

Adding and deleting filters in a SQL injection match condition

You can add or delete filters in a SQL injection match condition. To change a filter, add a new one and delete the old one.

To add or delete filters in a SQL injection match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **SQL injection**.
3. Choose the condition that you want to add or delete filters in.
4. To add filters, perform the following steps:

   a. Choose **Add filter**.
   b. Specify the applicable filter settings. For more information, see Values that you specify when you create or edit SQL injection match conditions.
   c. Choose **Add**.
5. To delete filters, perform the following steps:

   a. Select the filter that you want to delete.
   b. Choose **Delete filter**.

Deleting SQL injection match conditions

If you want to delete a SQL injection match condition, you need to first delete all filters in the condition and remove the condition from all the rules that are using it, as described in the following procedure.

To delete a SQL injection match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **SQL injection**.
3. In the **SQL injection match conditions** pane, choose the SQL injection match condition that you want to delete.
4. In the right pane, choose the **Associated rules** tab.

   If the list of rules using this SQL injection match condition is empty, go to step 6. If the list contains any rules, make note of the rules, and continue with step 5.
5. To remove the SQL injection match condition from the rules that are using it, perform the following steps:

   a. In the navigation pane, choose **Rules**.

   b. Choose the name of a rule that is using the SQL injection match condition that you want to delete.

   c. In the right pane, select the SQL injection match condition that you want to remove from the rule, and choose **Remove selected condition**.

   d. Repeat steps b and c for all of the remaining rules that are using the SQL injection match condition that you want to delete.

   e. In the navigation pane, choose **SQL injection**.

   f. In the **SQL injection match conditions** pane, choose the SQL injection match condition that you want to delete.
6. Choose **Delete** to delete the selected condition.

Working with string match conditions

> **Warning**
>
> AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

> **Note**
>
> This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
> **For the latest version of AWS WAF**, see AWS WAF.

If you want to allow or block web requests based on strings that appear in the requests, create one or more string match conditions. A string match condition identifies the string that you want to search for and the part of web requests, such as a specified header or the query string, that you want AWS WAF Classic to inspect for the string. Later in the process, when you create a web ACL, you specify whether to allow or block requests that contain the string.

Topics

- Creating a string match condition
- Values that you specify when you create or edit string match conditions
- Adding and deleting filters in a string match condition
- Deleting string match conditions

Creating a string match condition

When you create string match conditions, you specify filters that identify the string that you want to search for and the part of web requests that you want AWS WAF Classic to inspect for that string, such as the URI or the query string. You can add more than one filter to a string match condition, or you can create a separate string match condition for each filter. Here's how each configuration affects AWS WAF Classic behavior:

- **One filter per string match condition** – When you add the separate string match conditions to a rule and add the rule to a web ACL, web requests must match all the conditions for AWS WAF Classic to allow or block requests based on the conditions.

  For example, suppose you create two conditions. One matches web requests that contain the value `BadBot` in the `User-Agent` header. The other matches web requests that contain the value `BadParameter` in query strings. When you add both conditions to the same rule and add

the rule to a web ACL, AWS WAF Classic allows or blocks requests only when they contain both values.

- **More than one filter per string match condition** – When you add a string match condition that contains multiple filters to a rule and add the rule to a web ACL, a web request needs only to match one of the filters in the string match condition for AWS WAF Classic to allow or block the request based on the one condition.

  Suppose you create one condition instead of two, and the one condition contains the same two filters as in the preceding example. AWS WAF Classic allows or blocks requests if they contain *either* `BadBot` in the `User-Agent` header *or* `BadParameter` in the query string.

> **Note**
>
> When you add a string match condition to a rule, you also can configure AWS WAF Classic to allow or block web requests that *do not* match the values in the condition.

To create a string match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **String and regex matching**.
3. Choose **Create condition**.
4. Specify the applicable filter settings. For more information, see Values that you specify when you create or edit string match conditions.
5. Choose **Add filter**.
6. If you want to add another filter, repeat steps 4 and 5.
7. When you're finished adding filters, choose **Create**.

Values that you specify when you create or edit string match conditions

When you create or update a string match condition, you specify the following values:

Name

Enter a name for the string match condition. The name can contain only alphanumeric characters (A-Z, a-z, 0-9) or the following special characters: _-!"#`+*},./. You can't change the name of a condition after you create it.

Type

Choose **String match**.

Part of the request to filter on

Choose the part of each web request that you want AWS WAF Classic to inspect for the string that you specify in **Value to match**:

Header

A specified request header, for example, the `User-Agent` or `Referer` header. If you choose **Header**, specify the name of the header in the **Header** field.

HTTP method

The HTTP method, which indicates the type of operation that the request is asking the origin to perform. CloudFront supports the following methods: `DELETE`, `GET`, `HEAD`, `OPTIONS`, `PATCH`, `POST`, and `PUT`.

Query string

The part of a URL that appears after a `?` character, if any.

URI

The URI path of the request, which identifies the resource, for example, `/images/daily-ad.jpg`. This doesn't include the query string or fragment components of the URI. For information, see Uniform Resource Identifier (URI): Generic Syntax.

Unless a **Transformation** is specified, a URI is not normalized and is inspected just as AWS receives it from the client as part of the request. A **Transformation** will reformat the URI as specified.

Body

The part of a request that contains any additional data that you want to send to your web server as the HTTP request body, such as data from a form.

> **Note**
>
> If you choose **Body** for the value of **Part of the request to filter on**, AWS WAF Classic inspects only the first 8192 bytes (8 KB). To allow or block requests for which the body is longer than 8192 bytes, you can create a size constraint condition. (AWS WAF Classic gets the length of the body from the request headers.) For more information, see Working with size constraint conditions.

Single query parameter (value only)

Any parameter that you have defined as part of the query string. For example, if the URL is "www.xyz.com?UserName=abc&SalesRegion=seattle" you can add a filter to either the *UserName* or *SalesRegion* parameter.

If duplicate parameters appear in the query string, the values are evaluated as an "OR." That is, either value will trigger a match. For example, in the URL "www.xyz.com?SalesRegion=boston&SalesRegion=seattle", either "boston" or "seattle" in **Value to match** will trigger a match.

If you choose **Single query parameter (value only)** you will also specify a **Query parameter name**. This is the parameter in the query string that you will inspect, such as *UserName* or *SalesRegion*. The maximum length for **Query parameter name** is 30 characters. **Query parameter name** is not case sensitive. For example, it you specify *UserName* as the **Query parameter name**, this will match all variations of *UserName*, such as *username* and *UsERName*.

All query parameters (values only)

Similar to **Single query parameter (value only)**, but rather than inspecting the value of a single parameter, AWS WAF Classic inspects the value of all parameters within the query string for the **Value to match**. For example, if the URL is "www.xyz.com?UserName=abc&SalesRegion=seattle," and you choose **All query parameters (values only)**, AWS WAF Classic will trigger a match if the value of either *UserName* or *SalesRegion* is specified as the **Value to match**.

Header (Only When "Part of the request to filter on" is "Header")

If you chose **Header** from the **Part of the request to filter on** list, choose a header from the list of common headers, or enter the name of a header that you want AWS WAF Classic to inspect.

Match type

Within the part of the request that you want AWS WAF Classic to inspect, choose where the string in **Value to match** must appear to match this filter:

Contains

The string appears anywhere in the specified part of the request.

Contains word

The specified part of the web request must include **Value to match**, and **Value to match** must contain only alphanumeric characters or underscore (A-Z, a-z, 0-9, or _). In addition, **Value to match** must be a word, which means one of the following:

- **Value to match** exactly matches the value of the specified part of the web request, such as the value of a header.
- **Value to match** is at the beginning of the specified part of the web request and is followed by a character other than an alphanumeric character or underscore (_), for example, `BadBot;`.
- **Value to match** is at the end of the specified part of the web request and is preceded by a character other than an alphanumeric character or underscore (_), for example, `;BadBot`.
- **Value to match** is in the middle of the specified part of the web request and is preceded and followed by characters other than alphanumeric characters or underscore (_), for example, `-BadBot;`.

Exactly matches

The string and the value of the specified part of the request are identical.

Starts with

The string appears at the beginning of the specified part of the request.

Ends with

The string appears at the end of the specified part of the request.

Transformation

A transformation reformats a web request before AWS WAF Classic inspects the request. This eliminates some of the unusual formatting that attackers use in web requests in an effort to bypass AWS WAF Classic.

You can only specify a single type of text transformation.

Transformations can perform the following operations:

None

AWS WAF Classic doesn't perform any text transformations on the web request before inspecting it for the string in **Value to match**.

Convert to lowercase

AWS WAF Classic converts uppercase letters (A-Z) to lowercase (a-z).

HTML decode

AWS WAF Classic replaces HTML-encoded characters with unencoded characters:

- Replaces `"` with &
- Replaces ` ` with a non-breaking space
- Replaces `<` with <
- Replaces `>` with >
- Replaces characters that are represented in hexadecimal format, `&#xhhhh;`, with the corresponding characters
- Replaces characters that are represented in decimal format, `&#nnnn;`, with the corresponding characters

Normalize white space

AWS WAF Classic replaces the following characters with a space character (decimal 32):

- \f, formfeed, decimal 12
- \t, tab, decimal 9
- \n, newline, decimal 10
- \r, carriage return, decimal 13
- \v, vertical tab, decimal 11
- non-breaking space, decimal 160

In addition, this option replaces multiple spaces with one space.

Simplify command line

When you're concerned that attackers are injecting an operating system command line command and using unusual formatting to disguise some or all of the command, use this option to perform the following transformations:

- Delete the following characters: \ " ' ^

- Delete spaces before the following characters: / (
- Replace the following characters with a space: , ;
- Replace multiple spaces with one space
- Convert uppercase letters (A-Z) to lowercase (a-z)

URL decode

Decode a URL-encoded request.

Value is base64 encoded

If the value in **Value to match** is base64-encoded, select this check box. Use base64-encoding to specify non-printable characters, such as tabs and linefeeds, that attackers include in their requests.

Value to match

Specify the value that you want AWS WAF Classic to search for in web requests. The maximum length is 50 bytes. If you're base64-encoding the value, the 50-byte maximum length applies to the value before you encode it.

Adding and deleting filters in a string match condition

You can add filters to a string match condition or delete filters. To change a filter, add a new one and delete the old one.

To add or delete filters in a string match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **String and regex matching**.
3. Choose the condition that you want to add or delete filters in.
4. To add filters, perform the following steps:

   a. Choose **Add filter**.

   b. Specify the applicable filter settings. For more information, see Values that you specify when you create or edit string match conditions.

   c. Choose **Add**.

5. To delete filters, perform the following steps:

    a. Select the filter that you want to delete.

    b. Choose **Delete Filter**.

Deleting string match conditions

If you want to delete a string match condition, you need to first delete all filters in the condition and remove the condition from all the rules that are using it, as described in the following procedure.

To delete a string match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

    If you see **Switch to AWS WAF Classic** in the navigation pane, select it.

2. Remove the string match condition from the rules that are using it:

    a. In the navigation pane, choose **Rules**.

    b. Choose the name of a rule that is using the string match condition that you want to delete.

    c. In the right pane, choose **Edit rule**.

    d. Choose the **X** next to the condition you want to delete.

    e. Choose **Update**.

    f. Repeat for all the remaining rules that are using the string match condition that you want to delete.

3. Remove the filters from the condition you want to delete:

    a. In the navigation pane, choose **String and regex matching**.

    b. Choose the name of the string match condition that you want to delete.

    c. In the right pane, choose the check box next to **Filter** in order to select all of the filters.

    d. Choose the **Delete filter**.

4. In the navigation pane, choose **String and regex matching**.

5. In the **String and regex match conditions** pane, choose the string match condition that you want to delete.

6. Choose **Delete** to delete the selected condition.

Working with regex match conditions

> **Warning**
>
> AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

> **Note**
>
> This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
> **For the latest version of AWS WAF**, see AWS WAF.

If you want to allow or block web requests based on strings that match a regular expression (regex) pattern that appears in the requests, create one or more regex match conditions. A regex match condition is a type of string match condition that identifies the pattern that you want to search for and the part of web requests, such as a specified header or the query string, that you want AWS WAF Classic to inspect for the pattern. Later in the process, when you create a web ACL, you specify whether to allow or block requests that contain the pattern.

Topics

- Creating a regex match condition
- Values that you specify when you create or edit RegEx match conditions
- Editing a regex match condition

Creating a regex match condition

When you create regex match conditions, you specify pattern sets that identify the string (using a regular expression) that you want to search for. You then add those pattern sets to filters that specify the part of web requests that you want AWS WAF Classic to inspect for that pattern set, such as the URI or the query string.

You can add multiple regular expressions to a single pattern set. If you do so, those expressions are combined with an *OR*. That is, a web request will match the pattern set if the appropriate part of the request matches any of the expressions listed.

When you add a regex match condition to a rule, you also can configure AWS WAF Classic to allow or block web requests that *do not* match the values in the condition.

AWS WAF Classic supports most standard Perl Compatible Regular Expressions (PCRE). However, the following are not supported:

- Backreferences and capturing subexpressions
- Arbitrary zero-width assertions
- Subroutine references and recursive patterns
- Conditional patterns
- Backtracking control verbs
- The \C single-byte directive
- The \R newline match directive
- The \K start of match reset directive
- Callouts and embedded code
- Atomic grouping and possessive quantifiers

To create a regex match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **String and regex matching**.
3. Choose **Create condition**.
4. Specify the applicable filter settings. For more information, see Values that you specify when you create or edit RegEx match conditions.
5. Choose **Create pattern set and add filter** (if you created a new pattern set) or **Add filter** if you used an existing pattern set.
6. Choose **Create**.

Values that you specify when you create or edit RegEx match conditions

When you create or update a regex match condition, you specify the following values:

Name

Enter a name for the regex match condition. The name can contain only alphanumeric characters (A-Z, a-z, 0-9) or the following special characters: _-!"#`+*},./. You can't change the name of a condition after you create it.

Type

Choose **Regex match**.

Part of the request to filter on

Choose the part of each web request that you want AWS WAF Classic to inspect for the pattern that you specify in **Value to match**:

Header

A specified request header, for example, the `User-Agent` or `Referer` header. If you choose **Header**, specify the name of the header in the **Header** field.

HTTP method

The HTTP method, which indicates the type of operation that the request is asking the origin to perform. CloudFront supports the following methods: `DELETE`, `GET`, `HEAD`, `OPTIONS`, `PATCH`, `POST`, and `PUT`.

Query string

The part of a URL that appears after a ? character, if any.

URI

The URI path of the request, which identifies the resource, for example, `/images/daily-ad.jpg`. This doesn't include the query string or fragment components of the URI. For information, see Uniform Resource Identifier (URI): Generic Syntax.

Unless a **Transformation** is specified, a URI is not normalized and is inspected just as AWS receives it from the client as part of the request. A **Transformation** will reformat the URI as specified.

Body

The part of a request that contains any additional data that you want to send to your web server as the HTTP request body, such as data from a form.

> **Note**
>
> If you choose **Body** for the value of **Part of the request to filter on**, AWS WAF Classic inspects only the first 8192 bytes (8 KB). To allow or block requests for which the body is longer than 8192 bytes, you can create a size constraint condition. (AWS WAF Classic gets the length of the body from the request headers.) For more information, see Working with size constraint conditions.

Single query parameter (value only)

Any parameter that you have defined as part of the query string. For example, if the URL is "www.xyz.com?UserName=abc&SalesRegion=seattle" you can add a filter to either the *UserName* or *SalesRegion* parameter.

If duplicate parameters appear in the query string, the values are evaluated as an "OR." That is, either value will trigger a match. For example, in the URL "www.xyz.com?SalesRegion=boston&SalesRegion=seattle", a pattern that matches either "boston" or "seattle" in **Value to match** will trigger a match.

If you choose **Single query parameter (value only)** you will also specify a **Query parameter name**. This is the parameter in the query string that you will inspect, such as *UserName* or *SalesRegion*. The maximum length for **Query parameter name** is 30 characters. **Query parameter name** is not case sensitive. For example, it you specify *UserName* as the **Query parameter name**, this will match all variations of *UserName*, such as *username* and *UsERName*.

All query parameters (values only)

Similar to **Single query parameter (value only)**, but rather than inspecting the value of a single parameter, AWS WAF Classic inspects the value of all parameters within the query string for the pattern specified in the **Value to match**. For example, in the URL "www.xyz.com?UserName=abc&SalesRegion=seattle", a pattern in **Value to match** that matches either the value in *UserName* or *SalesRegion* will trigger a match.

Header (Only When "Part of the request to filter on" is "Header")

If you chose **Header** from the **Part of the request to filter on** list, choose a header from the list of common headers, or enter the name of a header that you want AWS WAF Classic to inspect.

Transformation

A transformation reformats a web request before AWS WAF Classic inspects the request. This eliminates some of the unusual formatting that attackers use in web requests in an effort to bypass AWS WAF Classic.

You can only specify a single type of text transformation.

Transformations can perform the following operations:

None

AWS WAF Classic doesn't perform any text transformations on the web request before inspecting it for the string in **Value to match**.

Convert to lowercase

AWS WAF Classic converts uppercase letters (A-Z) to lowercase (a-z).

HTML decode

AWS WAF Classic replaces HTML-encoded characters with unencoded characters:

- Replaces `"` with &
- Replaces ` ` with a non-breaking space
- Replaces `<` with <
- Replaces `>` with >
- Replaces characters that are represented in hexadecimal format, `&#xhhhh;`, with the corresponding characters
- Replaces characters that are represented in decimal format, `&#nnnn;`, with the corresponding characters

Normalize white space

AWS WAF Classic replaces the following characters with a space character (decimal 32):

- \f, formfeed, decimal 12
- \t, tab, decimal 9

- \n, newline, decimal 10
- \r, carriage return, decimal 13
- \v, vertical tab, decimal 11
- non-breaking space, decimal 160

In addition, this option replaces multiple spaces with one space.

Simplify command line

When you're concerned that attackers are injecting an operating system command line command and using unusual formatting to disguise some or all of the command, use this option to perform the following transformations:

- Delete the following characters: \ " ' ^
- Delete spaces before the following characters: / (
- Replace the following characters with a space: , ;
- Replace multiple spaces with one space
- Convert uppercase letters (A-Z) to lowercase (a-z)

URL decode

Decode a URL-encoded request.

Regex pattern to match to request

You can choose an existing pattern set, or create a new one. If you create a new one specify the following:

New pattern set name

Enter a name and then specify the regex pattern that you want AWS WAF Classic to search for.

If you add multiple regular expressions to a pattern set, those expressions are combined with an *OR*. That is, a web request will match the pattern set if the appropriate part of the request matches any of the expressions listed.

The maximum length of **Value to match** is 70 characters.

Editing a regex match condition

You can make the following changes to an existing regex match condition:

- Delete a pattern from an existing pattern set
- Add a pattern to an existing pattern set
- Delete a filter to an existing regex match condition
- Add a filter to an existing regex match condition (You can have only one filter in a regex match condition. Therefore, in order to add a filter, you must delete the existing filter first.)
- Delete an existing regex match condition

> **Note**
>
> You cannot add or delete a pattern set from an existing filter. You must either edit the pattern set, or delete the filter and create a new filter with a new pattern set.

To delete a pattern from an existing pattern set

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **String and regex matching**.
3. Choose **View regex pattern sets**.
4. Choose the name of the pattern set you want to edit.
5. Choose **Edit**.
6. Choose the **X** next to the pattern you want to delete.
7. Choose **Save**.

To add a pattern to an existing pattern set

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **String and regex matching**.
3. Choose **View regex pattern sets**.
4. Choose the name of the pattern set to edit.

5. Choose **Edit**.
6. Enter a new regex pattern.
7. Choose the **+** next to the new pattern.
8. Choose **Save**.

To delete a filter from an existing regex match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **String and regex matching**.
3. Choose the name of the condition with the filter you want to delete.
4. Choose the box next to the filter you want to delete.
5. Choose **Delete filter**.

To delete a regex match condition

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. Delete the filter from the regex condition. See To delete a filter from an existing regex match condition for instructions to do this.)
3. Remove the regex match condition from the rules that are using it:

   a. In the navigation pane, choose **Rules**.

   b. Choose the name of a rule that is using the regex match condition that you want to delete.

   c. In the right pane, choose **Edit rule**.

   d. Choose the **X** next to the condition you want to delete.

   e. Choose **Update**.

   f. Repeat for all the remaining rules that are using the regex match condition that you want to delete.

4. In the navigation pane, choose **String and regex matching**.
5. Select the button next to the condition you want to delete.
6. Choose **Delete**.

To add or change a filter to an existing regex match condition

You can have only one filter in a regex match condition. If you want to add or change the filter, you must first delete the existing filter.

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. Delete the filter from the regex condition you want to change. See To delete a filter from an existing regex match condition for instructions to do this.)
3. In the navigation pane, choose **String and regex matching**.
4. Choose the name of the condition you want to change.
5. Choose **Add filter**.
6. Enter the appropriate values for the new filter and choose **Add**.

Working with rules

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

Rules let you precisely target the web requests that you want AWS WAF Classic to allow or block by specifying the exact conditions that you want AWS WAF Classic to watch for. For example, AWS WAF Classic can watch for the IP addresses that requests originate from, the strings that the requests contain and where the strings appear, and whether the requests appear to contain malicious SQL code.

Topics

- Creating a rule and adding conditions
- Adding and removing conditions in a rule
- Deleting a rule
- AWS Marketplace rule groups

Creating a rule and adding conditions

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

If you add more than one condition to a rule, a web request must match all the conditions for AWS WAF Classic to allow or block requests based on that rule.

To create a rule and add conditions

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

If you see **Switch to AWS WAF Classic** in the navigation pane, select it.

2. In the navigation pane, choose **Rules**.
3. Choose **Create rule**.
4. Enter the following values:

   Name

   Enter a name.

   CloudWatch metric name

   Enter a name for the CloudWatch metric that AWS WAF Classic will create and will associate with the rule. The name can contain only alphanumeric characters (A-Z, a-z, 0-9), with maximum length 128 and minimum length one. It can't contain white space or metric names reserved for AWS WAF Classic, including "All" and "Default_Action.

   Rule type

   Choose either `Regular rule` or `Rate-based rule`. Rate–based rules are identical to regular rules, but also take into account how many requests arrive from an IP address in a five-minute period. For more information about these rule types, see How AWS WAF Classic works.

   Rate limit

   For a rate-based rule, enter the maximum number of requests to allow in any five-minute period from an IP address that matches the rule's conditions. The rate limit must be at least 100.

   You can specify a rate limit alone, or a rate limit and conditions. If you specify only a rate limit, AWS WAF places the limit on all IP addresses. If you specify a rate limit and conditions, AWS WAF places the limit on IP addresses that match the conditions.

   When an IP address reaches the rate limit threshold, AWS WAF applies the assigned action (block or count) as quickly as possible, usually within 30 seconds. Once the action is in place, if five minutes pass with no requests from the IP address, AWS WAF resets the counter to zero.

5. To add a condition to the rule, specify the following values:

When a request does/does not

If you want AWS WAF Classic to allow or block requests based on the filters in a condition, choose **does**. For example, if an IP match condition includes the IP address range 192.0.2.0/24 and you want AWS WAF Classic to allow or block requests that come from those IP addresses, choose **does**.

If you want AWS WAF Classic to allow or block requests based on the inverse of the filters in a condition, choose **does not**. For example, if an IP match condition includes the IP address range 192.0.2.0/24 and you want AWS WAF Classic to allow or block requests that *do not* come from those IP addresses, choose **does not**.

match/originate from

Choose the type of condition that you want to add to the rule:

- Cross-site scripting match conditions – choose **match at least one of the filters in the cross-site scripting match condition**
- IP match conditions – choose **originate from an IP address in**
- Geo match conditions – choose **originate from a geographic location in**
- Size constraint conditions – choose **match at least one of the filters in the size constraint condition**
- SQL injection match conditions – choose **match at least one of the filters in the SQL injection match condition**
- String match conditions – choose **match at least one of the filters in the string match condition**
- Regular expression match conditions – choose **match at least one of the filters in the regex match condition**

condition name

Choose the condition that you want to add to the rule. The list displays only conditions of the type that you chose in the preceding step.

6. To add another condition to the rule, choose **Add another condition**, and repeat steps 4 and 5. Note the following:

- If you add more than one condition, a web request must match at least one filter in every condition for AWS WAF Classic to allow or block requests based on that rule

- If you add two IP match conditions to the same rule, AWS WAF Classic will only allow or block requests that originate from IP addresses that appear in both IP match conditions

7. When you're finished adding conditions, choose **Create**.

Adding and removing conditions in a rule

> **Warning**
>
> AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

> **Note**
>
> This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
> **For the latest version of AWS WAF**, see AWS WAF.

You can change a rule by adding or removing conditions.

To add or remove conditions in a rule

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **Rules**.
3. Choose the name of the rule in which you want to add or remove conditions.
4. Choose **Add rule**.
5. To add a condition, choose **Add condition** and specify the following values:

When a request does/does not

If you want AWS WAF Classic to allow or block requests based on the filters in a condition, for example, web requests that originate from the range of IP addresses 192.0.2.0/24, choose **does**.

If you want AWS WAF Classic to allow or block requests based on the inverse of the filters in a condition, choose **does not**. For example, if an IP match condition includes the IP address range 192.0.2.0/24 and you want AWS WAF Classic to allow or block requests that *do not* come from those IP addresses, choose **does not**.

match/originate from

Choose the type of condition that you want to add to the rule:

- Cross-site scripting match conditions – choose **match at least one of the filters in the cross-site scripting match condition**
- IP match conditions – choose **originate from an IP address in**
- Geo match conditions – choose **originate from a geographic location in**
- Size constraint conditions – choose **match at least one of the filters in the size constraint condition**
- SQL injection match conditions – choose **match at least one of the filters in the SQL injection match condition**
- String match conditions – choose **match at least one of the filters in the string match condition**
- Regular expression match conditions – choose **match at least one of the filters in the regex match condition**

condition name

Choose the condition that you want to add to the rule. The list displays only conditions of the type that you chose in the preceding step.

6. To remove a condition, select the **X** to the right of the condition name
7. Choose **Update**.

Deleting a rule

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

If you want to delete a rule, you need to first remove the rule from the web ACLs that are using it and remove the conditions that are included in the rule.

To delete a rule

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.

2. To remove the rule from the web ACLs that are using it, perform the following steps for each of the web ACLs:

   a. In the navigation pane, choose **Web ACLs**.

   b. Choose the name of a web ACL that is using the rule that you want to delete.

      Note

      If you don't see the web ACL, make sure the Region selection is correct. Web ACLs that protect Amazon CloudFront distributions are in **Global (CloudFront)**.

   c. Choose the **Rules** tab.

d. Choose **Edit web ACL**.

e. Choose the **X** to the right of the rule that you want to delete, and then choose **Update**.

3. In the navigation pane, choose **Rules**.
4. Select the name of the rule you want to delete.

> **Note**
>
> If you don't see the rule, make sure the Region selection is correct. Rules that protect Amazon CloudFront distributions are in **Global (CloudFront)**.

5. Choose **Delete**.

AWS Marketplace rule groups

> **Warning**
>
> AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

> **Note**
>
> This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
> **For the latest version of AWS WAF**, see AWS WAF.

AWS WAF Classic provides *AWS Marketplace rule groups* to help you protect your resources. AWS Marketplace rule groups are collections of predefined, ready-to-use rules that are written and updated by AWS and AWS partner companies.

Some AWS Marketplace rule groups are designed to help protect specific types of web applications like WordPress, Joomla, or PHP. Other AWS Marketplace rule groups offer broad protection against known threats or common web application vulnerabilities, such as those listed in the OWASP Top 10.

You can install a single AWS Marketplace rule group from your preferred AWS partner, and you can also add your own customized AWS WAF Classic rules for increased protection. If you are subject to regulatory compliance like PCI or HIPAA, you might be able to use AWS Marketplace rule groups to satisfy web application firewall requirements.

AWS Marketplace rule groups are available with no long-term contracts, and no minimum commitments. When you subscribe to a rule group, you are charged a monthly fee (prorated hourly) and ongoing request fees based on volume. For more information, see AWS WAF Classic Pricing and the description for each AWS Marketplace rule group on AWS Marketplace.

Automatic updates

Keeping up to date on the constantly changing threat landscape can be time consuming and expensive. AWS Marketplace rule groups can save you time when you implement and use AWS WAF Classic. Another benefit is that AWS and our AWS partners automatically update AWS Marketplace rule groups when new vulnerabilities and threats emerge.

Many of our partners are notified of new vulnerabilities before public disclosure. They can update their rule groups and deploy them to you even before a new threat is widely known. Many also have threat research teams to investigate and analyze the most recent threats in order to write the most relevant rules.

Access to the rules in an AWS Marketplace rule group

Each AWS Marketplace rule group provides a comprehensive description of the types of attacks and vulnerabilities that it's designed to protect against. To protect the intellectual property of the rule group providers, you can't view the individual rules within a rule group. This restriction also helps to keep malicious users from designing threats that specifically circumvent published rules.

Because you can't view individual rules in an AWS Marketplace rule group, you also can't edit any rules in an AWS Marketplace rule group. However, you can exclude specific rules from a rule group. This is called a "rule group exception." Excluding rules does not remove those rules. Rather, it changes the action for the rules to `COUNT`. Therefore, requests that match an excluded rule are counted but not blocked. You will receive COUNT metrics for each excluded rule.

Excluding rules can be helpful when troubleshooting rule groups that are blocking traffic unexpectedly (false positives). One troubleshooting technique is to identify the specific rule within the rule group that is blocking the desired traffic and then disable (exclude) that particular rule.

In addition to excluding specific rules, you can refine your protection by enabling or disabling entire rule groups, as well as choosing the rule group action to perform. For more information, see Using AWS Marketplace rule groups.

Quotas

You can enable only one AWS Marketplace rule group. You can also enable one custom rule group that you create using AWS Firewall Manager. These rule groups count towards the 10 rule maximum quota per web ACL. Therefore, you can have one AWS Marketplace rule group, one custom rule group, and up to eight custom rules in a single web ACL.

Pricing

For AWS Marketplace rule group pricing, see AWS WAF Classic Pricing and the description for each AWS Marketplace rule group on AWS Marketplace.

Using AWS Marketplace rule groups

You can subscribe to and unsubscribe from AWS Marketplace rule groups on the AWS WAF Classic console. You can also exclude specific rules from a rule group.

To subscribe to and use an AWS Marketplace rule group

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **Marketplace**.
3. In the **Available marketplace products** section, choose the name of a rule group to view the details and pricing information.
4. If you want to subscribe to the rule group, choose **Continue**.

   > **Note**
   >
   > If you don't want to subscribe to this rule group, simply close this page in your browser.

5. Choose **Set up your account**.
6. Add the rule group to a web ACL, just as you would add an individual rule. For more information, see Creating a Web ACL or Editing a Web ACL.

> **Note**
>
> When adding a rule group to a web ACL, the action that you set for the rule group (either **No override** or **Override to count**) is called the rule group override action. For more information, see Rule group override.

To unsubscribe from an AWS Marketplace rule group

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. Remove the rule group from all web ACLs. For more information, see Editing a Web ACL.
3. In the navigation pane, choose **Marketplace**.
4. Choose **Manage your subscriptions**.
5. Choose **Cancel subscription** next to the name of the rule group that you want to unsubscribe from.
6. Choose **Yes, cancel subscription**.

To exclude a rule from a rule group (rule group exception)

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. If not already enabled, enable AWS WAF Classic logging. For more information, see Logging Web ACL traffic information. Use the AWS WAF Classic logs to identify the IDs of the rules that you want to exclude. These are typically rules that are blocking legitimate requests.
3. In the navigation pane, choose **Web ACLs**.
4. Choose the name of the web ACL that you want to edit. This opens a page with the web ACL's details in the right pane.

> **Note**
>
> The rule group that you want to edit must be associated with a web ACL before you can exclude a rule from that rule group.

5. On the **Rules** tab in the right pane, choose **Edit web ACL**.
6. In the **Rule group exceptions** section, expand the rule group that you want to edit.
7. Choose the **X** next to the rule that you want to exclude. You can identify the correct rule ID by using the AWS WAF Classic logs.
8. Choose **Update**.

Excluding rules does not remove those rules from the rule group. Rather, it changes the action for the rules to `COUNT`. Therefore, requests that match an excluded rule are counted but not blocked. You will receive `COUNT` metrics for each excluded rule.

> **Note**
>
> You can use this same procedure to exclude rules from custom rule groups that you have created in AWS Firewall Manager. However, rather than excluding a rule from a custom rule group using these steps, you can also simply edit a custom rule group using the steps described in Adding and deleting rules from an AWS WAF Classic rule group.

Rule group override

AWS Marketplace rule groups have two possible actions: **No override** and **Override to count**. If you want to test the rule group, set the action to **Override to count**. This rule group action overrides any *block* action that is specified by individual rules contained within the group. That is, if the rule group's action is set to **Override to count**, instead of potentially blocking matching requests based on the action of individual rules within the group, those requests will be counted. Conversely, if you set the rule group's action to **No override**, actions of the individual rules within the group will be used.

Troubleshooting AWS Marketplace rule groups

If you find that an AWS Marketplace rule group is blocking legitimate traffic, perform the following steps.

To troubleshoot an AWS Marketplace rule group

1. Exclude the specific rules that are blocking legitimate traffic. You can identify which rules are blocking which requests using the AWS WAF Classic logs. For more information about excluding rules, see To exclude a rule from a rule group (rule group exception).
2. If excluding specific rules does not solve the problem, you can change the action for the AWS Marketplace rule group from **No override** to **Override to count**. This allows the web request to pass through, regardless of the individual rule actions within the rule group. This also provides you with Amazon CloudWatch metrics for the rule group.
3. After setting the AWS Marketplace rule group action to **Override to count**, contact the rule group provider's customer support team to further troubleshoot the issue. For contact information, see the rule group listing on the product listing pages on AWS Marketplace.

Contacting customer support

For problems with AWS WAF Classic or a rule group that is managed by AWS, contact AWS Support. For problems with a rule group that is managed by an AWS partner, contact that partner's customer support team. To find partner contact information, see the partner's listing on AWS Marketplace.

Creating and selling AWS Marketplace rule groups

If you want to sell AWS Marketplace rule groups on AWS Marketplace, see How to Sell Your Software on AWS Marketplace.

Working with web ACLs

> **Warning**
>
> AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

> **Note**
>
> This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
> **For the latest version of AWS WAF**, see AWS WAF.

When you add rules to a web ACL, you specify whether you want AWS WAF Classic to allow or block requests based on the conditions in the rules. If you add more than one rule to a web ACL, AWS WAF Classic evaluates each request against the rules in the order that you list them in the web ACL. When a web request matches all the conditions in a rule, AWS WAF Classic immediately takes the corresponding action—allow or block—and doesn't evaluate the request against the remaining rules in the web ACL, if any.

If a web request doesn't match any of the rules in a web ACL, AWS WAF Classic takes the default action that you specified for the web ACL. For more information, see Deciding on the default action for a Web ACL.

If you want to test a rule before you start using it to allow or block requests, you can configure AWS WAF Classic to count the web requests that match the conditions in the rule. For more information, see Testing web ACLs.

Topics

- Deciding on the default action for a Web ACL
- Creating a Web ACL
- Associating or disassociating a Web ACL with an Amazon API Gateway API, a CloudFront distribution or an Application Load Balancer
- Editing a Web ACL
- Deleting a Web ACL
- Testing web ACLs

Deciding on the default action for a Web ACL

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

When you create and configure a web ACL, the first and most important decision that you must make is whether the default action should be for AWS WAF Classic to allow web requests or to block web requests. The default action indicates what you want AWS WAF Classic to do after it inspects a web request for all the conditions that you specify, and the web request doesn't match any of those conditions:

- **Allow** – If you want to allow most users to access your website, but you want to block access to attackers whose requests originate from specified IP addresses, or whose requests appear to contain malicious SQL code or specified values, choose **Allow** for the default action.
- **Block** – If you want to prevent most would-be users from accessing your website, but you want to allow access to users whose requests originate from specified IP addresses, or whose requests contain specified values, choose **Block** for the default action.

Many decisions that you make after you've decided on a default action depend on whether you want to allow or block most web requests. For example, if you want to *allow* most requests, then the match conditions that you create generally should specify the web requests that you want to *block*, such as the following:

- Requests that originate from IP addresses that are making an unreasonable number of requests

- Requests that originate from countries that either you don't do business in or are the frequent source of attacks
- Requests that include fake values in the **User-Agent** header
- Requests that appear to include malicious SQL code

Creating a Web ACL

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

To create a web ACL

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. If this is your first time using AWS WAF Classic, choose **Go to AWS WAF Classic** and then **Configure Web ACL**. If you've used AWS WAF Classic before, choose **Web ACLs** in the navigation pane, and then choose **Create web ACL**.
3. For **Web ACL name**, enter a name.

   Note

   You can't change the name after you create the web ACL.

4. For **CloudWatch metric name**, change the default name if applicable. The name can contain only alphanumeric characters (A-Z, a-z, 0-9), with maximum length 128 and minimum length one. It can't contain white space or metric names reserved for AWS WAF Classic, including "All" and "Default_Action."

   > **Note**
   >
   > You can't change the name after you create the web ACL.

5. For **Region**, choose a Region.
6. For **AWS resource**, choose the resource that you want to associate with this web ACL, and then choose **Next**.
7. If you've already created the conditions that you want AWS WAF Classic to use to inspect your web requests, choose **Next**, and then continue to the next step.

   If you haven't already created conditions, do so now. For more information, see the following topics:

   - Working with cross-site scripting match conditions
   - Working with IP match conditions
   - Working with geographic match conditions
   - Working with size constraint conditions
   - Working with SQL injection match conditions
   - Working with string match conditions
   - Working with regex match conditions
8. If you've already created the rules or rule groups (or subscribed to an AWS Marketplace rule group) that you want to add to this web ACL, add the rules to the web ACL:

   a. In the **Rules** list, choose a rule.

   b. Choose **Add rule to web ACL**.

   c. Repeat steps a and b until you've added all the rules that you want to add to this web ACL.

   d. Go to step 10.
9. If you haven't created rules yet, you can add rules now:

   a. Choose **Create rule**.

b. Enter the following values:

Name

Enter a name.

CloudWatch metric name

Enter a name for the CloudWatch metric that AWS WAF Classic will create and will associate with the rule. The name can contain only alphanumeric characters (A-Z, a-z, 0-9), with maximum length 128 and minimum length one. It can't contain white space or metric names reserved for AWS WAF Classic, including "All" and "Default_Action."

> **Note**
>
> You can't change the metric name after you create the rule.

c. To add a condition to the rule, specify the following values:

When a request does/does not

If you want AWS WAF Classic to allow or block requests based on the filters in a condition, for example, web requests that originate from the range of IP addresses 192.0.2.0/24, choose **does**.

If you want AWS WAF Classic to allow or block requests based on the inverse of the filters in a condition, choose **does not**. For example, if an IP match condition includes the IP address range 192.0.2.0/24 and you want AWS WAF Classic to allow or block requests that *do not* come from those IP addresses, choose **does not**.

match/originate from

Choose the type of condition that you want to add to the rule:

- Cross-site scripting match conditions – choose **match at least one of the filters in the cross-site scripting match condition**
- IP match conditions – choose **originate from an IP address in**
- Geo match conditions – choose **originate from a geographic location in**
- Size constraint conditions – choose **match at least one of the filters in the size constraint condition**

- SQL injection match conditions – choose **match at least one of the filters in the SQL injection match condition**
- String match conditions – choose **match at least one of the filters in the string match condition**
- Regex match conditions – choose **match at least one of the filters in the regex match condition**

condition name

Choose the condition that you want to add to the rule. The list displays only conditions of the type that you chose in the preceding list.

d. To add another condition to the rule, choose **Add another condition**, and then repeat steps b and c. Note the following:

- If you add more than one condition, a web request must match at least one filter in every condition for AWS WAF Classic to allow or block requests based on that rule.
- If you add two IP match conditions to the same rule, AWS WAF Classic will only allow or block requests that originate from IP addresses that appear in both IP match conditions.

e. Repeat step 9 until you've created all the rules that you want to add to this web ACL.

f. Choose **Create**.

g. Continue with step 10.

10. For each rule or rule group in the web ACL, choose the kind of management you want AWS WAF Classic to provide, as follows:

- For each rule, choose whether you want AWS WAF Classic to allow, block, or count web requests based on the conditions in the rule:
  - **Allow** – API Gateway, CloudFront or an Application Load Balancer responds with the requested object. In the case of CloudFront, if the object isn't in the edge cache, CloudFront forwards the request to the origin.
  - **Block** – API Gateway, CloudFront or an Application Load Balancer responds to the request with an HTTP 403 (Forbidden) status code. CloudFront also can respond with a custom error page. For more information, see Using AWS WAF Classic with CloudFront custom error pages.
  - **Count** – AWS WAF Classic increments a counter of requests that match the conditions in the rule, and then continues to inspect the web request based on the remaining rules in the web ACL.

For information about using **Count** to test a web ACL before you start to use it to allow or block web requests, see Counting the web requests that match the rules in a web ACL.

- For each rule group, set the override action for the rule group:
  - **No override** – Causes the actions of the individual rules within the rule group to be used.
  - **Override to count** – Overrides any block actions that are specifieid by individual rules in the group, so that all matching requests are only counted.

  For more information, see Rule group override.

11. If you want to change the order of the rules in the web ACL, use the arrows in the **Order** column. AWS WAF Classic inspects web requests based on the order in which rules appear in the web ACL.
12. If you want to remove a rule that you added to the web ACL, choose the **x** in the row for the rule.
13. Choose the default action for the web ACL. This is the action that AWS WAF Classic takes when a web request doesn't match the conditions in any of the rules in this web ACL. For more information, see Deciding on the default action for a Web ACL.
14. Choose **Review and create**.
15. Review the settings for the web ACL, and choose **Confirm and create**.

Associating or disassociating a Web ACL with an Amazon API Gateway API, a CloudFront distribution or an Application Load Balancer

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.

For the latest version of AWS WAF, see AWS WAF.

To associate or disassociate a web ACL, perform the applicable procedure. Note that you also can associate a web ACL with a CloudFront distribution when you create or update the distribution. For more information, see Using AWS WAF Classic to Control Access to Your Content in the *Amazon CloudFront Developer Guide*.

The following restrictions apply when associating a web ACL:

- Each API Gateway API, Application Load Balancer and CloudFront distribution can be associated with only one web ACL.
- Web ACLs associated with a CloudFront distribution cannot be associated with an Application Load Balancer or API Gateway API. The web ACL can, however, be associated with other CloudFront distributions.

To associate a web ACL with an API Gateway API, CloudFront distribution or Application Load Balancer

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **Web ACLs**.
3. Choose the name of the web ACL that you want to associate with an API Gateway API, CloudFront distribution or Application Load Balancer. This opens a page with the web ACL's details in the right pane.
4. On the **Rules** tab, under **AWS resources using this web ACL**, choose **Add association**.
5. When prompted, use the **Resource** list to choose the API Gateway API, CloudFront distribution or Application Load Balancer that you want to associate this web ACL with. If you choose an Application Load Balancer, you also must specify a Region.
6. Choose **Add**.
7. To associate this web ACL with an additional API Gateway API, CloudFront distribution or another Application Load Balancer, repeat steps 4 through 6.

To disassociate a web ACL from an API Gateway API, CloudFront distribution or Application Load Balancer

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **Web ACLs**.
3. Choose the name of the web ACL that you want to disassociate from an API Gateway API, CloudFront distribution or Application Load Balancer. This opens a page with the web ACL's details in the right pane.
4. On the **Rules** tab, under **AWS resources using this web ACL**, choose the **x** for each API Gateway API, CloudFront distribution or Application Load Balancer that you want to disassociate this web ACL from.

Editing a Web ACL

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

To add or remove rules from a web ACL or change the default action, perform the following procedure.

To edit a web ACL

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **Web ACLs**.
3. Choose the name of the web ACL that you want to edit. This opens a page with the web ACL's details in the right pane.
4. On the **Rules** tab in the right pane, choose **Edit web ACL**.
5. To add rules to the web ACL, perform the following steps:

   a. In the **Rules** list, choose the rule that you want to add.

   b. Choose **Add rule to web ACL**.

   c. Repeat steps a and b until you've added all the rules that you want.
6. If you want to change the order of the rules in the web ACL, use the arrows in the **Order** column. AWS WAF Classic inspects web requests based on the order in which rules appear in the web ACL.
7. To remove a rule from the web ACL, choose the **x** at the right of the row for that rule. This doesn't delete the rule from AWS WAF Classic, it just removes the rule from this web ACL.
8. To change the action for a rule or the default action for the web ACL, choose the preferred option.

   > **Note**
   >
   > When setting the action for a rule group or an AWS Marketplace rule group (as opposed to a single rule), the action you set for the rule group (either **No override** or **Override to count**) is called the override action. For more information, see Rule group override
9. Choose **Save changes**.

Deleting a Web ACL

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

Important

Deleting a web ACL is permanent and can't be undone. If the selected web ACL contains any rules or is associated with any CloudFront distributions, Application load balancer or API Gateway, remove the rules and associations before deleting. Otherwise, the delete will fail.

To delete a web ACL, you must remove the rules that are included in the web ACL and disassociate all CloudFront distributions and Application Load Balancers from the web ACL. Perform the following procedure.

To delete a web ACL

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **Web ACLs**.
3. Choose the name of the web ACL that you want to delete. This opens a page with the web ACL's details in the right pane.

> **Note**
>
> If you don't see the web ACL, make sure the Region selection is correct. Web ACLs that protect Amazon CloudFront distributions are in **Global (CloudFront)**.

4. On the **Rules** tab in the right pane, choose **Edit web ACL**.
5. To remove all rules from the web ACL, choose the **x** at the right of the row for each rule. This doesn't delete the rules from AWS WAF Classic, it just removes the rules from this web ACL.
6. Choose **Update**.
7. Disassociate the web ACL from all CloudFront distributions and Application Load Balancers. On the **Rules** tab, under **AWS resources using this web ACL**, choose the **x** for each API Gateway API, CloudFront distribution or Application Load Balancer.
8. On the **Web ACLs** page, confirm that the web ACL that you want to delete is selected, and then choose **Delete**.

Testing web ACLs

> **Warning**
>
> AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

> **Note**
>
> This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
> **For the latest version of AWS WAF**, see AWS WAF.

To ensure that you don't accidentally configure AWS WAF Classic to block web requests that you want to allow or allow requests that you want to block, we recommend that you test your web ACL thoroughly before you start using it on your website or web application.

Topics

- Counting the web requests that match the rules in a web ACL
- Viewing a sample of the web requests that API Gateway CloudFront or an Application Load Balancer has forwarded to AWS WAF Classic

Counting the web requests that match the rules in a web ACL

When you add rules to a web ACL, you specify whether you want AWS WAF Classic to allow, block, or count the web requests that match all the conditions in that rule. We recommend that you begin with the following configuration:

- Configure all the rules in a web ACL to count web requests
- Set the default action for the web ACL to allow requests

In this configuration, AWS WAF Classic inspects each web request based on the conditions in the first rule. If the web request matches all the conditions in that rule, AWS WAF Classic increments a counter for that rule. Then AWS WAF Classic inspects the web request based on the conditions in the next rule. If the request matches all the conditions in that rule, AWS WAF Classic increments a counter for the rule. This continues until AWS WAF Classic has inspected the request based on the conditions in all of your rules.

After you've configured all the rules in a web ACL to count requests and associated the web ACL with an Amazon API Gateway API, CloudFront distribution or Application Load Balancer, you can view the resulting counts in an Amazon CloudWatch graph. For each rule in a web ACL and for all the requests that API Gateway, CloudFront or an Application Load Balancer forwards to AWS WAF Classic for a web ACL, CloudWatch lets you:

- View data for the preceding hour or preceding three hours,
- Change the interval between data points
- Change the calculation that CloudWatch performs on the data, such as maximum, minimum, average, or sum

> **Note**
>
> AWS WAF Classic with CloudFront is a global service and metrics are available only when you choose the **US East (N. Virginia) Region** in the AWS Management Console. If you choose another region, no AWS WAF Classic metrics will appear in the CloudWatch console.

To view data for the rules in a web ACL

1. Sign in to the AWS Management Console and open the CloudWatch console at https://eusc-de-east-1.console.amazonaws-eusc.eu/cloudwatch/.
2. In the navigation pane, under **Metrics**, choose **WAF**.
3. Select the check box for the web ACL that you want to view data for.
4. Change the applicable settings:

   Statistic

   Choose the calculation that CloudWatch performs on the data.

   Time range

   Choose whether you want to view data for the preceding hour or the preceding three hours.

   Period

   Choose the interval between data points in the graph.

   Rules

   Choose the rules for which you want to view data.

   Note the following:

   - If you just associated a web ACL with an API Gateway API, CloudFront distribution or Application Load Balancer, you might need to wait a few minutes for data to appear in the graph and for the metric for the web ACL to appear in the list of available metrics.
   - If you associate more than one API Gateway API, CloudFront distribution or Application Load Balancer with a web ACL, the CloudWatch data will include all the requests for all the distributions that are associated with the web ACL.

- You can hover the mouse cursor over a data point to get more information.
- The graph doesn't refresh itself automatically. To update the display, choose the refresh ( ) icon.

5. (Optional) View detailed information about individual requests that API Gateway CloudFront or an Application Load Balancer has forwarded to AWS WAF Classic. For more information, see Viewing a sample of the web requests that API Gateway CloudFront or an Application Load Balancer has forwarded to AWS WAF Classic.
6. If you determine that a rule is intercepting requests that you don't want it to intercept, change the applicable settings. For more information, see Creating and configuring a Web Access Control List (Web ACL).

   When you're satisfied that all of your rules are intercepting only the correct requests, change the action for each of your rules to **Allow** or **Block**. For more information, see Editing a Web ACL.

Viewing a sample of the web requests that API Gateway CloudFront or an Application Load Balancer has forwarded to AWS WAF Classic

In the AWS WAF Classic console, you can view a sample of the requests that API Gateway CloudFront or an Application Load Balancer has forwarded to AWS WAF Classic for inspection. For each sampled request, you can view detailed data about the request, such as the originating IP address and the headers included in the request. You also can view which rule the request matched, and whether the rule is configured to allow or block requests.

The sample of requests contains up to 100 requests that matched all the conditions in each rule and another 100 requests for the default action, which applies to requests that didn't match all the conditions in any rule. The requests in the sample come from all the API Gateway APIs, CloudFront edge locations or Application Load Balancers that have received requests for your content in the previous 15 minutes.

To view a sample of the web requests that API Gateway; CloudFront or an Application Load Balancer has forwarded to AWS WAF Classic

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

If you see **Switch to AWS WAF Classic** in the navigation pane, select it.

2. In the navigation pane, choose the web ACL for which you want to view requests.
3. In the right pane, choose the **Requests** tab.

   The **Sampled requests** table displays the following values for each request:

   Source IP

   Either the IP address that the request originated from or, if the viewer used an HTTP proxy or an Application Load Balancer to send the request, the IP address of the proxy or Application Load Balancer.

   URI

   The URI path of the request, which identifies the resource, for example, `/images/daily-ad.jpg`. This doesn't include the query string or fragment components of the URI. For information, see Uniform Resource Identifier (URI): Generic Syntax.

   Matches rule

   Identifies the first rule in the web ACL for which the web request matched all the conditions. If a web request doesn't match all the conditions in any rule in the web ACL, the value of **Matches rule** is **Default**.

   Note that when a web request matches all the conditions in a rule and the action for that rule is **Count**, AWS WAF Classic continues inspecting the web request based on subsequent rules in the web ACL. In this case, a web request could appear twice in the list of sampled requests: once for the rule that has an action of **Count** and again for a subsequent rule or for the default action.

   Action

   Indicates whether the action for the corresponding rule is **Allow**, **Block**, or **Count**.

   Time

   The time that AWS WAF Classic received the request from API Gateway, CloudFront or your Application Load Balancer.

4. To display additional information about the request, choose the arrow on the left side of the IP address for that request. AWS WAF Classic displays the following information:

Source IP

The same IP address as the value in the **Source IP** column in the table.

Country

The two-letter country code of the country that the request originated from. If the viewer used an HTTP proxy or an Application Load Balancer to send the request, this is the two-letter country code of the country that the HTTP proxy or an Application Load Balancer is in.

For a list of two-letter country codes and the corresponding country names, see the Wikipedia entry ISO 3166-1 alpha-2.

Method

The HTTP request method for the request: `GET`, `HEAD`, `OPTIONS`, `PUT`, `POST`, `PATCH`, or `DELETE`.

URI

The same URI as the value in the **URI** column in the table.

Request headers

The request headers and header values in the request.

5. To refresh the list of sample requests, choose **Get new samples**.

Working with AWS WAF Classic rule groups for use with AWS Firewall Manager

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

An AWS WAF Classic *rule group* is a set of rules that you add to an AWS WAF Classic AWS Firewall Manager policy. You can create your own rule group, or you can purchase a managed rule group from AWS Marketplace.

Important

If you want to add an AWS Marketplace rule group to your Firewall Manager policy, each account in your organization must first subscribe to that rule group. After all accounts have subscribed, you can then add the rule group to a policy. For more information, see AWS Marketplace rule groups.

Topics

- Creating an AWS WAF Classic rule group
- Adding and deleting rules from an AWS WAF Classic rule group

Creating an AWS WAF Classic rule group

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and

you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

When you create an AWS WAF Classic rule group to use with AWS Firewall Manager, you specify which rules to add to the group.

To create a rule group (console)

1. Sign in to the AWS Management Console using the AWS Firewall Manager administrator account that you set up in the prerequisites, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fms.

   Note

   For information about setting up a Firewall Manager administrator account, see Creating an AWS Firewall Manager default administrator account.

2. In the navigation pane, choose **Switch to AWS WAF Classic**.
3. In the AWS WAF Classic navigation pane, choose **Rule groups**.
4. Choose **Create rule group**.

   Note

   You can't add rate-based rules to a rule group.

5. If you have already created the rules that you want to add to the rule group, choose **Use existing rules for this rule group** . If you want to create new rules to add to the rule group, choose **Create rules and conditions for this rule group**.
6. Choose **Next**.
7. If you chose to create rules, follow the steps to create them at Creating a rule and adding conditions.

   Note

   Use the AWS WAF Classic console to create your rules.

When you've created all the rules you need, go to the next step.

8. Type a rule group name.
9. To add a rule to the rule group, select a rule then choose **Add rule**. Choose whether to allow, block, or count requests that match the rule's conditions. For more information on the choices, see How AWS WAF Classic works.
10. When you are finished adding rules, choose **Create**.

You can test your rule group by adding it to an AWS WAF WebACL and setting the WebACL action to **Override to Count**. This action overrides any action that you choose for the rules contained in the group, and only counts matching requests. For more information, see Creating a Web ACL.

Adding and deleting rules from an AWS WAF Classic rule group

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

You can add or delete rules in an AWS WAF Classic rule group.

Deleting a rule from the rule group does not delete the rule itself. It only removes the rule from the rule group.

To add or delete rules in a rule group (console)

1. Sign in to the AWS Management Console using the AWS Firewall Manager administrator account that you set up in the prerequisites, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fms.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see Creating an AWS Firewall Manager default administrator account.

2. In the navigation pane, choose **Switch to AWS WAF Classic**.
3. In the AWS WAF Classic navigation pane, choose **Rule groups**.
4. Choose the rule group that you want to edit.

   > **Note**
   >
   > If you don't see the rule group that you want to edit, make sure you have the correct Region selected. For rule groups used to protect Amazon CloudFront distributions, use the **Global (CloudFront)** setting.

5. Choose **Edit rule group**.
6. To add rules, perform the following steps:

   a. Select a rule, and then choose **Add rule to rule group**. Choose whether to allow, block, or count requests that match the rule's conditions. For more information on the choices, see How AWS WAF Classic works. Repeat to add more rules to the rule group.

      > **Note**
      >
      > You cannot add rate-based rules to rule group.

   b. Choose **Update**.

7. To delete rules, perform the following steps:

   a. Choose the **X** next to the rule to delete. Repeat to delete more rules from the rule group.

   b. Choose **Update**.

Getting started with AWS Firewall Manager to enable AWS WAF Classic rules

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

You can use AWS Firewall Manager to enable AWS WAF rules, AWS WAF Classic rules, AWS Shield Advanced protections, and Amazon VPC security groups. The steps for getting set up are slightly different for each:

- To use Firewall Manager to enable rules using the latest version of AWS WAF, don't use this topic. Instead, follow the steps in Setting up AWS Firewall Manager AWS WAF policies.
- To use Firewall Manager to enable AWS Shield Advanced protections, follow the steps in Setting up AWS Firewall Manager AWS Shield Advanced policies.
- To use Firewall Manager to enable Amazon VPC security groups, follow the steps in Setting up AWS Firewall Manager Amazon VPC security group policies.

To use Firewall Manager to enable AWS WAF Classic rules, perform the following steps in sequence.

Topics

- Step 1: Complete the prerequisites
- Step 2: Create rules
- Step 3: Create a rule group

- Step 4: Create and apply an AWS Firewall Manager AWS WAF Classic policy

Step 1: Complete the prerequisites

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

There are several mandatory steps to prepare your account for AWS Firewall Manager. Those steps are described in AWS Firewall Manager prerequisites. Complete all the prerequisites before proceeding to Step 2: Create rules.

Step 2: Create rules

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.

For the latest version of AWS WAF, see AWS WAF.

In this step, you create rules using AWS WAF Classic. If you already have AWS WAF Classic rules that you want to use with AWS Firewall Manager, skip this step and go to Step 3: Create a rule group.

Note

Use the AWS WAF Classic console to create your rules.

To create AWS WAF Classic rules (console)

- Create your rules, and then add your conditions to your rules. For more information, see Creating a rule and adding conditions.

You are now ready to go to Step 3: Create a rule group.

Step 3: Create a rule group

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

A rule group is a set of rules that defines what actions to take when a particular set of conditions is met. You can use managed rule groups from AWS Marketplace, and you can create your own rule groups. For information about managed rule groups, see AWS Marketplace rule groups.

To create your own rule group, perform the following procedure.

To create a rule group (console)

1. Sign in to the AWS Management Console using the AWS Firewall Manager administrator account that you set up in the prerequisites, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fms.
2. In the navigation pane, choose **Security policies**.
3. If you have not met the prerequisites, the console displays instructions about how to fix any issues. Follow the instructions, and then begin this step (create a rule group) again. If you have met the prerequisites, choose **Close**.
4. Choose **Create policy**.

   For **Policy type**, choose **AWS WAF Classic**.
5. Choose **Create an AWS Firewall Manager policy and add a new rule group**.
6. Choose an AWS Region, and then choose **Next**.
7. Because you already created rules, you don't need to create conditions. Choose **Next**.
8. Because you already created rules, you don't need to create rules. Choose **Next**.
9. Choose **Create rule group**.
10. For **Name**, enter a friendly name.
11. Enter a name for the CloudWatch metric that AWS WAF Classic will create and will associate with the rule group. The name can contain only alphanumeric characters (A-Z, a-z, 0-9) or the following special characters: _-!"#`+*},./. It can't contain white space.
12. Select a rule, and then choose **Add rule**. A rule has an action setting that allows you to choose whether to allow, block, or count requests that match the rule's conditions. For this tutorial, choose **Count**. Repeat adding rules until you have added all the rules that you want to the rule group.
13. Choose **Create**.

You are now ready to go to Step 4: Create and apply an AWS Firewall Manager AWS WAF Classic policy.

Step 4: Create and apply an AWS Firewall Manager AWS WAF Classic policy

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

After you create the rule group, you create an AWS Firewall Manager AWS WAF policy. A Firewall Manager AWS WAF policy contains the rule group that you want to apply to your resources.

To create a Firewall Manager AWS WAF policy (console)

1. After you create the rule group (the last step in the preceding procedure, Step 3: Create a rule group), the console displays the **Rule group summary** page. Choose **Next**.
2. For **Name**, enter a friendly name.
3. For **Policy type**, choose **WAF**.
4. For **Region**, choose an AWS Region. To protect Amazon CloudFront resources, choose **Global**.

   To protect resources in multiple regions (other than CloudFront resources), you must create separate Firewall Manager policies for each Region.
5. Select a rule group to add, and then choose **Add rule group**.
6. A policy has two possible actions: **Action set by rule group** and **Count**. If you want to test the policy and rule group, set the action to **Count**. This action overrides any *block* action specified by the rule group contained in the policy. That is, if the policy's action is set to **Count**, those requests are only counted and not blocked. Conversely, if you set the policy's action to **Action**

set by rule group, actions of the rule group in the policy are used. For this tutorial, choose **Count**.

7. Choose **Next**.
8. If you want to include only specific accounts in the policy, or alternatively exclude specific accounts from the policy, select **Select accounts to include/exclude from this policy (optional)**. Choose either **Include only these accounts in this policy** or **Exclude these accounts from this policy**. You can choose only one option. Choose **Add**. Select the account numbers to include or exclude, and then choose **OK**.

   > **Note**
   >
   > If you don't select this option, Firewall Manager applies a policy to all accounts in your organization in AWS Organizations. If you add a new account to the organization, Firewall Manager automatically applies the policy to that account.

9. Choose the types of resources that you want to protect.
10. If you want to protect only resources with specific tags, or alternatively exclude resources with specific tags, select **Use tags to include/exclude resources**, enter the tags, and then choose either **Include** or **Exclude**. You can choose only one option.

    If you enter more than one tag (separated by commas), and if a resource has any of those tags, it is considered a match.

    For more information about tags, see Working with Tag Editor.
11. Choose **Create and apply this policy to existing and new resources**.

    This option creates a web ACL in each applicable account within an organization in AWS Organizations, and associates the web ACL with the specified resources in the accounts. This option also applies the policy to all new resources that match the preceding criteria (resource type and tags). Alternatively, if you choose **Create but do not apply this policy to existing or new resources**, Firewall Manager creates a web ACL in each applicable account within the organization, but doesn't apply the web ACL to any resources. You must apply the policy to resources later.
12. Leave the choice for **Replace existing associated web ACLs** at the default setting.

    When this option is selected, Firewall Manager removed all existing web ACL associations from in-scope resources before it associates the new policy's web ACLs to them.

13. Choose **Next**.
14. Review the new policy. To make any changes, choose **Edit**. When you are satisfied with the policy, choose **Create policy**.

Tutorial: Creating an AWS Firewall Manager policy with hierarchical rules

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

With AWS Firewall Manager, you can create and apply AWS WAF Classic protection policies that contain hierarchical rules. That is, you can create and enforce certain rules centrally, but delegate the creation and maintenance of account-specific rules to other individuals. You can monitor the centrally applied (common) rules for any accidental removal or mishandling, thereby ensuring that they are applied consistently. The account-specific rules add further protection customized for the needs of individual teams.

Note

In the latest version of AWS WAF, this capability is built in and doesn't require any special handling. If you aren't already using AWS WAF Classic, use the latest version instead. See Creating an AWS Firewall Manager policy for AWS WAF.

The following tutorial describes how to create a hierarchical set of protection rules.

Topics

- Step 1: Designate a Firewall Manager administrator account
- Step 2: Create a rule group using the Firewall Manager administrator account
- Step 3: Create a Firewall Manager policy and attach the common rule group
- Step 4: Add account-specific rules
- Conclusion

Step 1: Designate a Firewall Manager administrator account

To use AWS Firewall Manager, you must designate an account in your organization as the Firewall Manager administrator account. This account can be either the management account or a member account in the organization.

You can use the Firewall Manager administrator account to create a set of common rules that you apply to other accounts in the organization. Other accounts in the organization can't change these centrally applied rules.

To designate an account as a Firewall Manager administrator account and complete other prerequisites for using Firewall Manager, see the instructions in AWS Firewall Manager prerequisites. If you've already completed the prerequisites, you can skip to step 2 of this tutorial.

In this tutorial, we refer to the administrator account as **`Firewall-Administrator-Account`**.

Step 2: Create a rule group using the Firewall Manager administrator account

Next, create a rule group using **`Firewall-Administrator-Account`**. This rule group contains the common rules that you will apply to all member accounts governed by the policy that you create in the next step. Only **`Firewall-Administrator-Account`** can make changes to these rules and the container rule group.

In this tutorial, we refer to this container rule group as **`Common-Rule-Group`**.

To create a rule group, see the instructions in Creating an AWS WAF Classic rule group. Remember to sign in to the console using your Firewall Manager administrator account (**`Firewall-Administrator-Account`**) when following these instructions.

Step 3: Create a Firewall Manager policy and attach the common rule group

Using **`Firewall-Administrator-Account`**, create a Firewall Manager policy. When you create this policy, you must do the following:

- Add **`Common-Rule-Group`** to the new policy.
- Include all accounts in the organization that you want **`Common-Rule-Group`** applied to.
- Add all resources that you want **`Common-Rule-Group`** applied to.

For instructions on creating a policy, see Creating an AWS Firewall Manager policy.

This creates a web ACL in each specified account and adds **`Common-Rule-Group`** to each of those web ACLs. After you create the policy, this web ACL and the common rules are deployed to all specified accounts.

In this tutorial, we refer to this web ACL as **`Administrator-Created-ACL`**. A unique **`Administrator-Created-ACL`** now exists in each specified member account of the organization.

Step 4: Add account-specific rules

Each member account in the organization can now add their own account-specific rules to the **`Administrator-Created-ACL`** that exists in their account. The common rules already in **`Administrator-Created-ACL`** continue to apply, along with the new, account-specific rules. AWS WAF inspects web requests based on the order in which rules appear in the web ACL. This applies to both **`Administrator-Created-ACL`** and account-specific rules.

To add rules to **`Administrator-Created-ACL`**, see Editing a protection pack (web ACL) in AWS WAF.

Conclusion

You now have a web ACL that contains common rules administered by the Firewall Manager administrator account as well as account-specific rules maintained by each member account.

The **`Administrator-Created-ACL`** in each account references the single **`Common-Rule-Group`**. Therefore, future changes by the Firewall Manager administrator account to **`Common-Rule-Group`** will immediately take effect in each member account.

Member accounts can't change or remove the common rules in **`Common-Rule-Group`**.

Account-specific rules don't affect other accounts.

Logging Web ACL traffic information

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

Note

You cannot use Amazon Security Lake to collect AWS WAF Classic data.

You can enable logging to get detailed information about traffic that is analyzed by your web ACL. Information that is contained in the logs include the time that AWS WAF Classic received the request from your AWS resource, detailed information about the request, and the action for the rule that each request matched.

To get started, you set up an Amazon Kinesis Data Firehose. As part of that process, you choose a destination for storing your logs. Next, you choose the web ACL that you want to enable logging for. After you enable logging, AWS WAF delivers logs through the firehose to your storage destination.

For information about how to create an Amazon Kinesis Data Firehose and review your stored logs, see What Is Amazon Data Firehose? To understand the permissions required for your Kinesis Data Firehose configuration, see Controlling Access with Amazon Kinesis Data Firehose.

You must have the following permissions to successfully enable logging:

- `iam:CreateServiceLinkedRole`
- `firehose:ListDeliveryStreams`
- `waf:PutLoggingConfiguration`

For more information about service-linked roles and the `iam:CreateServiceLinkedRole` permission, see Using service-linked roles for AWS WAF Classic.

To enable logging for a web ACL

1. Create an Amazon Kinesis Data Firehose using a name starting with the prefix "aws-waf-logs-" For example, `aws-waf-logs-us-east-2-analytics`. Create the data firehose with a PUT source and in the region that you are operating. If you are capturing logs for Amazon CloudFront, create the firehose in US East (N. Virginia). For more information, see Creating an Amazon Data Firehose Delivery Stream.

   > **Important**
   >
   > Do not choose `Kinesis stream` as your source.
   > One AWS WAF Classic log is equivalent to one Firehose record. If you typically receive 10,000 requests per second and you enable full logs, you should have a 10,000 records per second setting in Firehose. If you don't configure Firehose correctly, AWS WAF Classic won't record all logs. For more information, see Amazon Kinesis Data Firehose Quotas.

2. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.

3. In the navigation pane, choose **Web ACLs**.
4. Choose the name of the web ACL that you want to enable logging for. This opens a page with the web ACL's details in the right pane.

5. On the **Logging** tab, choose **Enable logging**.
6. Choose the Kinesis Data Firehose that you created in the first step. You must choose a firehose that begins with "aws-waf-logs-."
7. (Optional) If you don't want certain fields and their values included in the logs, redact those fields. Choose the field to redact, and then choose **Add**. Repeat as necessary to redact additional fields. The redacted fields appear as `REDACTED` in the logs. For example, if you redact the **cookie** field, the **cookie** field in the logs will be `REDACTED`.
8. Choose **Enable logging**.

> **Note**
>
> When you successfully enable logging, AWS WAF Classic will create a service linked role with the necessary permissions to write logs to the Amazon Kinesis Data Firehose. For more information, see Using service-linked roles for AWS WAF Classic.

To disable logging for a web ACL

1. In the navigation pane, choose **Web ACLs**.
2. Choose the name of the web ACL that you want to disable logging for. This opens a page with the web ACL's details in the right pane.
3. On the **Logging** tab, choose **Disable logging**.
4. In the dialog box, choose **Disable logging**.

Example log


```
{

 "timestamp":1533689070589,
 "formatVersion":1,
 "webaclId":"385cb038-3a6f-4f2f-ac64-09ab912af590",
 "terminatingRuleId":"Default_Action",
 "terminatingRuleType":"REGULAR",
 "action":"ALLOW",
 "httpSourceName":"CF",
 "httpSourceId":"i-123",
 "ruleGroupList":[
```

```
                        {
                         "ruleGroupId":"41f4eb08-4e1b-2985-92b5-e8abf434fad3",
                         "terminatingRule":null,
                         "nonTerminatingMatchingRules":[
                                                         {"action" : "COUNT",
                                                         "ruleId" :
 "4659b169-2083-4a91-bbd4-08851a9aaf74"}
                                                        ],
                         "excludedRules":               [
                                                         {"exclusionType" :
 "EXCLUDED_AS_COUNT",
                                                          "ruleId" :
 "5432a230-0113-5b83-bbb2-89375c5bfa98"}
                                                        ]
                        }
                       ],

 "rateBasedRuleList":[
                           {
                            "rateBasedRuleId":"7c968ef6-32ec-4fee-96cc-51198e412e7f",

                            "limitKey":"IP",
                            "maxRateAllowed":100

                           },
                           {
                            "rateBasedRuleId":"462b169-2083-4a93-bbd4-08851a9aaf30",
                            "limitKey":"IP",
                            "maxRateAllowed":100
                            }
                            ],

 "nonTerminatingMatchingRules":[
                                    {"action" : "COUNT",

                                    "ruleId" : "4659b181-2011-4a91-
bbd4-08851a9aaf52"}
                                   ],

 "httpRequest":{
                      "clientIp":"192.10.23.23",

                      "country":"US",
```

```
                    "headers":[

                                {
                                 "name":"Host",
                                 "value":"127.0.0.1:1989"
                                },
                                {
                                 "name":"User-Agent",
                                 "value":"curl/7.51.2"
                                },
                                {
                                 "name":"Accept",
                                 "value":"*/*"
                                }
                              ],
                   "uri":"REDACTED",
                   "args":"usernam=abc",
                   "httpVersion":"HTTP/1.1",
                   "httpMethod":"GET",
                   "requestId":"cloud front Request id"
                   }
}
```

Following is an explanation of each item listed in these logs:

timestamp

The timestamp in milliseconds.

formatVersion

The format version for the log.

webaclId

The GUID of the web ACL.

terminatingRuleId

The ID of the rule that terminated the request. If nothing terminates the request, the value is `Default_Action`.

terminatingRuleType

The type of rule that terminated the request. Possible values: RATE_BASED, REGULAR, and GROUP.

action

The action. Possible values for a terminating rule: ALLOW and BLOCK. COUNT is not a valid value for a terminating rule.

terminatingRuleMatchDetails

Detailed information about the terminating rule that matched the request. A terminating rule has an action that ends the inspection process against a web request. Possible actions for a terminating rule are ALLOW and BLOCK. This is only populated for SQL injection and cross-site scripting (XSS) match rule statements. As with all rule statements that inspect for more than one thing, AWS WAF applies the action on the first match and stops inspecting the web request. A web request with a terminating action could contain other threats, in addition to the one reported in the log.

httpSourceName

The source of the request. Possible values: CF (if the source is Amazon CloudFront), APIGW (if the source is Amazon API Gateway), and ALB (if the source is an Application Load Balancer).

httpSourceId

The source ID. This field shows the ID of the associated Amazon CloudFront distribution, the REST API for API Gateway, or the name for an Application Load Balancer.

ruleGroupList

The list of rule groups that acted on this request. In the preceding code example, there is only one.

ruleGroupId

The ID of the rule group. If the rule blocked the request, the ID for `ruleGroupID` is the same as the ID for `terminatingRuleId`.

terminatingRule

The rule within the rule group that terminated the request. If this is a non-null value, it also contains a **ruleid** and **action**. In this case, the action is always BLOCK.

nonTerminatingMatchingRules

The list of rules in the rule group that match the request. These are always COUNT rules (non-terminating rules that match).

action (nonTerminatingMatchingRules group)

This is always COUNT (non-terminating rules that match).

ruleId (nonTerminatingMatchingRules group)

The ID of the rule within the rule group that matches the request and was non-terminating. That is, COUNT rules.

excludedRules

The list of rules in the rule group that you have excluded. The action for these rules is set to COUNT.

exclusionType (excludedRules group)

A type that indicates that the excluded rule has the action COUNT.

ruleId (excludedRules group)

The ID of the rule within the rule group that is excluded.

rateBasedRuleList

The list of rate-based rules that acted on the request.

rateBasedRuleId

The ID of the rate-based rule that acted on the request. If this has terminated the request, the ID for `rateBasedRuleId` is the same as the ID for `terminatingRuleId`.

limitKey

The field that AWS WAF uses to determine if requests are likely arriving from a single source and thus subject to rate monitoring. Possible value: IP.

maxRateAllowed

The maximum number of requests, which have an identical value in the field that is specified by `limitKey`, allowed in a five-minute period. If the number of requests exceeds the `maxRateAllowed` and the other predicates specified in the rule are also met, AWS WAF triggers the action that is specified for this rule.

httpRequest

The metadata about the request.

clientIp

The IP address of the client sending the request.

country

The source country of the request. If AWS WAF is unable to determine the country of origin, it sets this field to `-`.

headers

The list of headers.

uri

The URI of the request. The preceding code example demonstrates what the value would be if this field had been redacted.

args

The query string.

httpVersion

The HTTP version.

httpMethod

The HTTP method in the request.

requestId

The ID of the request.

Listing IP addresses blocked by rate-based rules

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and

you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

AWS WAF Classic provides a list of IP addresses that are blocked by rate-based rules.

To view addresses blocked by rate-based rules

1. Sign in to the AWS Management Console and open the AWS WAF console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

   If you see **Switch to AWS WAF Classic** in the navigation pane, select it.
2. In the navigation pane, choose **Rules**.
3. In the **Name** column, choose a rate-based rule.

   The list shows the IP addresses that the rule currently blocks.

How AWS WAF Classic works with Amazon CloudFront features

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

When you create a web ACL, you can specify one or more CloudFront distributions that you want AWS WAF Classic to inspect. AWS WAF Classic starts to allow, block, or count web requests for

those distributions based on the conditions that you identify in the web ACL. CloudFront provides some features that enhance the AWS WAF Classic functionality. This chapter describes a few ways that you can configure CloudFront to make CloudFront and AWS WAF Classic work better together.

Topics

- Using AWS WAF Classic with CloudFront custom error pages
- Using AWS WAF Classic with CloudFront for applications running on your own HTTP server
- Choosing the HTTP methods that CloudFront responds to

Using AWS WAF Classic with CloudFront custom error pages

When AWS WAF Classic blocks a web request based on the conditions that you specify, it returns HTTP status code 403 (Forbidden) to CloudFront. Next, CloudFront returns that status code to the viewer. The viewer then displays a brief and sparsely formatted default message similar to this:

```
Forbidden: You don't have permission to access /myfilename.html on this server.
```

If you'd rather display a custom error message, possibly using the same formatting as the rest of your website, you can configure CloudFront to return to the viewer an object (for example, an HTML file) that contains your custom error message.

> **Note**
>
> CloudFront can't distinguish between an HTTP status code 403 that is returned by your origin and one that is returned by AWS WAF Classic when a request is blocked. This means that you can't return different custom error pages based on the different causes of an HTTP status code 403.

For more information about CloudFront custom error pages, see Customizing Error Responses in the *Amazon CloudFront Developer Guide*.

Using AWS WAF Classic with CloudFront for applications running on your own HTTP server

When you use AWS WAF Classic with CloudFront, you can protect your applications running on any HTTP webserver, whether it's a webserver that's running in Amazon Elastic Compute Cloud

(Amazon EC2) or a webserver that you manage privately. You can also configure CloudFront to require HTTPS between CloudFront and your own webserver, as well as between viewers and CloudFront.

Requiring HTTPS Between CloudFront and Your Own Webserver

To require HTTPS between CloudFront and your own webserver, you can use the CloudFront custom origin feature and configure the **Origin Protocol Policy** and the **Origin Domain Name** settings for specific origins. In your CloudFront configuration, you can specify the DNS name of the server along with the port and the protocol that you want CloudFront to use when fetching objects from your origin. You should also ensure that the SSL/TLS certificate on your custom origin server matches the origin domain name you've configured. When you use your own HTTP webserver outside of AWS, you must use a certificate that is signed by a trusted third-party certificate authority (CA), for example, Comodo, DigiCert, or Symantec. For more information about requiring HTTPS for communication between CloudFront and your own webserver, see the topic Requiring HTTPS for Communication Between CloudFront and Your Custom Origin in the *Amazon CloudFront Developer Guide*.

Requiring HTTPS Between a Viewer and CloudFront

To require HTTPS between viewers and CloudFront, you can change the **Viewer Protocol Policy** for one or more cache behaviors in your CloudFront distribution. For more information about using HTTPS between viewers and CloudFront, see the topic Requiring HTTPS for Communication Between Viewers and CloudFront in the *Amazon CloudFront Developer Guide*. You can also bring your own SSL certificate so viewers can connect to your CloudFront distribution over HTTPS using your own domain name, for example *https://www.mysite.com*. For more information, see the topic Configuring Alternate Domain Names and HTTPS in the *Amazon CloudFront Developer Guide*.

Choosing the HTTP methods that CloudFront responds to

When you create an Amazon CloudFront web distribution, you choose the HTTP methods that you want CloudFront to process and forward to your origin. You can choose from the following options:

- **GET, HEAD** – You can use CloudFront only to get objects from your origin or to get object headers.
- **GET, HEAD, OPTIONS** – You can use CloudFront only to get objects from your origin, get object headers, or retrieve a list of the options that your origin server supports.

- **GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE** – You can use CloudFront to get, add, update, and delete objects, and to get object headers. In addition, you can perform other POST operations such as submitting data from a web form.

You also can use AWS WAF Classic string match conditions to allow or block requests based on the HTTP method, as described in Working with string match conditions. If you want to use a combination of methods that CloudFront supports, such as GET and HEAD, then you don't need to configure AWS WAF Classic to block requests that use the other methods. If you want to allow a combination of methods that CloudFront doesn't support, such as GET, HEAD, and POST, you can configure CloudFront to respond to all methods, and then use AWS WAF Classic to block requests that use other methods.

For more information about choosing the methods that CloudFront responds to, see Allowed HTTP Methods in the topic Values that You Specify When You Create or Update a Web Distribution in the *Amazon CloudFront Developer Guide*.

Security in AWS WAF Classic

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. The effectiveness of our security is regularly tested and verified by third-party auditors as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS WAF Classic, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your organization's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using AWS WAF Classic. The following topics show you how to configure AWS WAF Classic to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS WAF Classic resources.

Topics

- Data protection in AWS WAF Classic
- Identity and access management for AWS WAF Classic
- Logging and monitoring in AWS WAF Classic
- Compliance validation for AWS WAF Classic
- Resilience in AWS WAF Classic
- Infrastructure security in AWS WAF Classic

Data protection in AWS WAF Classic

> **Warning**
>
> AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

> **Note**
>
> This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
> **For the latest version of AWS WAF**, see AWS WAF.

The AWS shared responsibility model applies to data protection in AWS WAF Classic. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see Data Privacy FAQ. For information about data protection in Europe, see the General Data Protection Regulation (GDPR) Center.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes

when you work with AWS WAF Classic or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

AWS WAF Classic entities—such as web ACLs, rules, and conditions—are encrypted at rest, except in certain Regions where encryption is not available, including China (Beijing) and China (Ningxia). Unique encryption keys are used for each Region.

Deleting AWS WAF Classic resources

You can delete the resources that you create in AWS WAF Classic. See the guidance for each resource type in following sections.

- Deleting a Web ACL
- Adding and deleting rules from an AWS WAF Classic rule group
- Deleting a rule

Identity and access management for AWS WAF Classic

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in)

and *authorized* (have permissions) to use AWS WAF Classic resources. IAM is an AWS service that you can use with no additional charge.

Topics

- Audience
- Authenticating with identities
- Managing access using policies
- How AWS WAF Classic works with IAM
- Identity-based policy examples for AWS WAF Classic
- Troubleshooting AWS WAF Classic identity and access
- Using service-linked roles for AWS WAF Classic

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in AWS WAF Classic.

Service user – If you use the AWS WAF Classic service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more AWS WAF Classic features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in AWS WAF Classic, see Troubleshooting AWS WAF Classic identity and access.

Service administrator – If you're in charge of AWS WAF Classic resources at your company, you probably have full access to AWS WAF Classic. It's your job to determine which AWS WAF Classic features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with AWS WAF Classic, see How AWS WAF Classic works with IAM.

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to AWS WAF Classic. To view example AWS WAF Classic identity-based policies that you can use in IAM, see Identity-based policy examples for AWS WAF Classic.

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be authenticated as the AWS account root user, an IAM user, or by assuming an IAM role.

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you don't use the root user for everyday tasks. For tasks that require root user credentials, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users to use federation with an identity provider to access AWS services using temporary credentials.

A *federated identity* is a user from your enterprise directory, web identity provider, or Directory Service that accesses AWS services using credentials from an identity source. Federated identities assume roles that provide temporary credentials.

For centralized access management, we recommend AWS IAM Identity Center. For more information, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see Require human users to use federation with an identity provider to access AWS using temporary credentials in the *IAM User Guide*.

An *IAM group* specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity with specific permissions that provides temporary credentials. You can assume a role by switching from a user to an IAM role (console) or by calling an AWS CLI or AWS API operation. For more information, see Methods to assume a role in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you attach to an identity (user, group, or role). These policies control what actions identities can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples include IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-

based policies, service administrators can use them to control access to a specific resource. You must specify a principal in a resource-based policy.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries** – Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – Set the maximum available permissions for resources in your accounts. For more information, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Advanced policies passed as a parameter when creating a temporary session for a role or federated user. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How AWS WAF Classic works with IAM

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

Before you use IAM to manage access to AWS WAF Classic, learn what IAM features are available to use with AWS WAF Classic.

IAM features you can use with AWS WAF Classic

IAM feature	AWS WAF Classic support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	Yes
ACLs	No
ABAC (tags in policies)	Partial
Temporary credentials	Yes

IAM feature	AWS WAF Classic support
Forward access sessions (FAS)	Yes
Service roles	Yes
Service-linked roles	Yes

To get a high-level view of how AWS WAF Classic and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for AWS WAF Classic

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

To view examples of AWS WAF Classic identity-based policies, see Identity-based policy examples for AWS WAF Classic.

Resource-based policies within AWS WAF Classic

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for AWS WAF Classic

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Include actions in a policy to grant permissions to perform the associated operation.

To see a list of AWS WAF Classic actions, see Actions defined by AWS WAF and Actions defined by AWS WAF Regional in the *Service Authorization Reference*.

Policy actions in AWS WAF Classic use the following prefix before the action:

```
waf
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "waf:action1",
      "waf:action2"
         ]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions in AWS WAF Classic that begin with `List`, include the following action:

```
"Action": "waf:List*"
```

To view examples of AWS WAF Classic identity-based policies, see Identity-based policy examples for AWS WAF Classic.

Policy resources for AWS WAF Classic

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. As a best practice, specify a resource using its Amazon Resource Name (ARN). For actions that don't support resource-level permissions, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see the list of AWS WAF Classic resource types and their ARNs, see Resources defined by AWS WAF and Resources defined by AWS WAF Regional in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by AWS WAF and Actions defined by AWS WAF Regional. To allow or deny access to a subset of AWS WAF Classic resources, include the ARN of the resource in the `resource` element of your policy.

In AWS WAF Classic, the resources are *web ACLs* and *rules*. AWS WAF Classic also supports conditions such as *byte match*, *IP match*, and *size constraint*.

These resources and conditions have unique Amazon Resource Names (ARNs) associated with them, as shown in the following table.

Name in AWS WAF Console	Name in AWS WAF SDK/CLI	ARN Format
Web ACL	`WebACL`	`arn:aws:waf::` *`account`*`:`*`webacl`*`/`*`ID`*
Rule	`Rule`	`arn:aws:waf::` *`account`*`:`*`rule`*`/`*`ID`*
String match condition	`ByteMatchSet`	`arn:aws:waf::` *`account`*`:`*`bytematchset`*` /`*`ID`*
SQL injection match condition	`SqlInjectionMatchSet`	`arn:aws:waf::` *`account`*`:`*`sqlinjectionset`*` /`*`ID`*
Size constraint condition	`SizeConstraintSet`	`arn:aws:waf::` *`account`*`:`*`sizeconstraintset`*` /`*`ID`*

Name in AWS WAF Console	Name in AWS WAF SDK/CLI	ARN Format	
IP match condition	IPSet	arn:aws:waf:: *account*:*ipset*/*ID*	
Cross-site scripting match condition	XssMatchSet	arn:aws:waf:: *account*:*xssmatchset* /*ID*	

To allow or deny access to a subset of AWS WAF Classic resources, include the ARN of the resource in the `resource` element of your policy. The ARNs for AWS WAF Classic have the following format:

```
arn:aws:waf::account:resource/ID
```

Replace the *account*, *resource*, and *ID* variables with valid values. Valid values can be the following:

- *account*: The ID of your AWS account. You must specify a value.
- *resource*: The type of AWS WAF Classic resource.
- *ID*: The ID of the AWS WAF Classic resource, or a wildcard (*) to indicate all resources of the specified type that are associated with the specified AWS account.

For example, the following ARN specifies all web ACLs for the account 111122223333:

```
arn:aws:waf::111122223333:webacl/*
```

Policy condition keys for AWS WAF Classic

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element specifies when statements execute based on defined criteria. You can create conditional expressions that use condition operators, such as equals or less than, to match

the condition in the policy with values in the request. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of AWS WAF Classic condition keys, see Condition keys for AWS WAF and Resources defined by AWS WAF Regional in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by AWS WAF and Actions defined by AWS WAF Regional.

To view examples of AWS WAF Classic identity-based policies, see Identity-based policy examples for AWS WAF Classic.

ACLs in AWS WAF Classic

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ABAC with AWS WAF Classic

Supports ABAC (tags in policies): Partial

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes called tags. You can attach tags to IAM entities and AWS resources, then design ABAC policies to allow operations when the principal's tag matches the tag on the resource.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see Define permissions with ABAC authorization in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with AWS WAF Classic

Supports temporary credentials: Yes

Temporary credentials provide short-term access to AWS resources and are automatically created when you use federation or switch roles. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM and AWS services that work with IAM in the *IAM User Guide*.

Forward access sessions for AWS WAF Classic

Supports forward access sessions (FAS): Yes

Forward access sessions (FAS) use the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. For policy details when making FAS requests, see Forward access sessions.

Service roles for AWS WAF Classic

Supports service roles: Yes

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.

> **Warning**
>
> Changing the permissions for a service role might break AWS WAF Classic functionality. Edit service roles only when AWS WAF Classic provides guidance to do so.

Service-linked roles for AWS WAF Classic

Supports service-linked roles: Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing AWS WAF Classic service-linked roles, see Using service-linked roles for AWS WAF Classic.

Identity-based policy examples for AWS WAF Classic

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

By default, users and roles don't have permission to create or modify AWS WAF Classic resources. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Create IAM policies (console) in the *IAM User Guide*.

For details about actions and resource types defined by AWS WAF Classic, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for AWS WAF and Actions, resources, and condition keys for AWS WAF Regional in the *Service Authorization Reference*.

Topics

- Policy best practices
- Using the AWS WAF Classic console
- Allow users to view their own permissions

Policy best practices

Identity-based policies determine whether someone can create, access, or delete AWS WAF Classic resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the AWS WAF Classic console

To access the AWS WAF Classic console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the AWS WAF Classic resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

Users who can access and use the AWS console can also access the AWS WAF Classic console. No additional permissions are required.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws-eusc:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
```

```
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Troubleshooting AWS WAF Classic identity and access

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

Use the following information to help you diagnose and fix common issues that you might encounter when working with AWS WAF Classic and IAM.

Topics

- I am not authorized to perform an action in AWS WAF Classic
- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my AWS WAF Classic resources

I am not authorized to perform an action in AWS WAF Classic

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `waf:`*`GetWidget`* permissions.

```
User: arn:aws-eusc:iam::123456789012:user/mateojackson is not authorized to perform:
 waf:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `waf:`*`GetWidget`* action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to AWS WAF Classic.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in AWS WAF Classic. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws-eusc:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my AWS WAF Classic resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether AWS WAF Classic supports these features, see How AWS WAF Classic works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.

Using service-linked roles for AWS WAF Classic

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

AWS WAF Classic uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to AWS WAF Classic. Service-linked roles are predefined by AWS WAF Classic and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up AWS WAF Classic easier because you don't have to manually add the necessary permissions. AWS WAF Classic defines the permissions of its service-linked roles, and unless defined otherwise, only AWS WAF Classic can assume its roles. The defined permissions include the trust policy and the permissions policy. That permissions policy can't be attached to any other IAM entity.

You can delete a service-linked role only after first deleting the role's related resources. This protects your AWS WAF Classic resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for AWS WAF Classic

AWS WAF Classic uses the following service-linked roles:

- `AWSServiceRoleForWAFLogging`
- `AWSServiceRoleForWAFRegionalLogging`

AWS WAF Classic uses these service-linked roles to write logs to Amazon Data Firehose. These roles are used only if you enable logging in AWS WAF. For more information, see Logging Web ACL traffic information.

The `AWSServiceRoleForWAFLogging` and `AWSServiceRoleForWAFRegionalLogging` service-linked roles trust the following services (respectively) to assume the role:

- `waf.amazonaws.com`

  `waf-regional.amazonaws.com`

The permissions policies of the roles allow AWS WAF Classic to complete the following actions on the specified resources:

- Action: `firehose:PutRecord` and `firehose:PutRecordBatch` on Amazon Data Firehose data stream resources with a name that starts with "aws-waf-logs-." For example, `aws-waf-logs-us-east-2-analytics`.
- Action: `kms:GenerateDataKey` and `kms:Decrypt` on AWS Key Management Service to allow Amazon Data Firehose to use customer managed AWS KMS keys to encrypt data.

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

Creating a service-linked role for AWS WAF Classic

You don't need to manually create a service-linked role. When you enable AWS WAF Classic logging on the AWS Management Console, or you make a `PutLoggingConfiguration` request in the AWS WAF Classic CLI or the AWS WAF Classic API, AWS WAF Classic creates the service-linked role for you.

You must have the `iam:CreateServiceLinkedRole` permission to enable logging.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you enable AWS WAF Classic logging, AWS WAF Classic creates the service-linked role for you again.

Editing a service-linked role for AWS WAF Classic

AWS WAF Classic doesn't allow you to edit the `AWSServiceRoleForWAFLogging` and `AWSServiceRoleForWAFRegionalLogging` service-linked roles. After you create a service-linked role, you can't change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting a service-linked role for AWS WAF Classic

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don’t have an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

> **Note**
>
> If the AWS WAF Classic service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To delete AWS WAF Classic resources used by the `AWSServiceRoleForWAFLogging` and `AWSServiceRoleForWAFRegionalLogging`

1. On the AWS WAF Classic console, remove logging from every web ACL. For more information, see Logging Web ACL traffic information.
2. Using the API or CLI, submit a `DeleteLoggingConfiguration` request for each web ACL that has logging enabled. For more information, see AWS WAF Classic API Reference.

To manually delete the service-linked role using IAM

Use the IAM console, the IAM CLI, or the IAM API to delete the `AWSServiceRoleForWAFLogging` and `AWSServiceRoleForWAFRegionalLogging` service-linked roles. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

Supported Regions for AWS WAF Classic service-linked roles

AWS WAF Classic supports using service-linked roles in the following AWS Regions.

Region Name	Region Identity	Support in AWS WAF Classic
US East (N. Virginia)	`eusc-de-east-1`	Yes
	`eusc-de-east-1`	Yes
US West (N. California)	`us-west-1`	Yes
US West (Oregon)	`us-west-2`	Yes
Asia Pacific (Mumbai)	`ap-south-1`	Yes
Asia Pacific (Osaka)	`ap-northeast-3`	Yes

Region Name	Region Identity	Support in AWS WAF Classic
Asia Pacific (Seoul)	ap-northeast-2	Yes
Asia Pacific (Singapore)	ap-southeast-1	Yes
Asia Pacific (Sydney)	ap-southeast-2	Yes
Asia Pacific (Tokyo)	ap-northeast-1	Yes
Canada (Central)	ca-central-1	Yes
Europe (Frankfurt)	eu-central-1	Yes
Europe (Ireland)	eu-west-1	Yes
Europe (London)	eu-west-2	Yes
Europe (Paris)	eu-west-3	Yes
South America (São Paulo)	sa-east-1	Yes

Logging and monitoring in AWS WAF Classic

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS WAF Classic and your AWS solutions. You should collect monitoring data from all parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your AWS WAF Classic resources and responding to potential events:

Amazon CloudWatch Alarms

Using CloudWatch alarms, you watch a single metric over a time period that you specify. If the metric exceeds a given threshold, CloudWatch sends a notification to an Amazon SNS topic or AWS Auto Scaling policy. For more information, see Monitoring with Amazon CloudWatch.

AWS CloudTrail Logs

CloudTrail provides a record of actions taken by a user, role, or an AWS service in AWS WAF Classic. Using the information collected by CloudTrail, you can determine the request that was made to AWS WAF Classic, the IP address from which the request was made, who made the request, when it was made, and additional details. For more information, see Logging API calls with AWS CloudTrail.

Compliance validation for AWS WAF Classic

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. For more information about your compliance responsibility when using AWS services, see AWS Security Documentation.

Resilience in AWS WAF Classic

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure security in AWS WAF Classic

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and

you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

As a managed service, AWS WAF Classic is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access AWS WAF Classic through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

AWS WAF Classic quotas

Warning

AWS WAF Classic is is going through a planned end-of-life process. Refer to your AWS Health dashboard for the milestones and dates specific to your Region.

Note

This is **AWS WAF Classic** documentation. You should only use this version if you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November 2019, and you have not migrated them over to the latest version yet. To migrate your web ACLs, see Migrating your AWS WAF Classic resources to AWS WAF.
For the latest version of AWS WAF, see AWS WAF.

AWS WAF Classic is subject to the following quotas (formerly referred to as limits).

AWS WAF Classic has default quotas on the number of entities per account per Region. You can request an increase to these.

Resource	Default quota per account per Region
Web ACLs	50
Rules	100
Rate-based-rules	5
Conditions per account per Region	For all conditions except for regex match and geo match, 100 of each condition type. For example, 100 size constraint conditions and 100 IP match conditions. For regex and geo match conditions, see the following table.
Requests per Second	25,000 per web ACL*

*This quota applies only to AWS WAF Classic on an Application Load Balancer. Requests per Second (RPS) quotas for AWS WAF Classic on CloudFront are the same as the RPS quotas support by CloudFront that is described in the CloudFront Developer Guide.

The following quotas on AWS WAF Classic entities can't be changed.

Resource	Quota per account per Region
Rule groups per web ACL	2: 1 customer-created rule group and 1 AWS Marketplace rule group
Rules per web ACL	10
Conditions per rule	10
IP address ranges (in CIDR notation) per IP match condition	10,000 You can update up to 1,000 addresses at a time. The API call `UpdateIPSet` accepts a maximum of 1,000 addresses in a single request.
IP addresses blocked per rate-based rule	10,000
Minimum rate-based rule rate limit per 5 minute period	100
Filters per cross-site scripting match condition	10
Filters per size constraint condition	10
Filters per SQL injection match condition	10

Resource	Quota per account per Region
Filters per string match condition	10
In string match conditions, the number of characters in HTTP header names, when you've configured AWS WAF Classic to inspect the headers in web requests for a specified value	40
In string match conditions, the number of characters in the value that you want AWS WAF Classic to search for	50
Regex match conditions	10
In regex match conditions, the number of characters in the pattern that you want AWS WAF Classic to search for	70
In regex match conditions, the number of patterns per pattern set	10
In regex match conditions, the number of pattern sets per regex condition	1
Pattern sets	5
Geo match conditions	50
Locations per geo match condition	50

AWS WAF Classic has the following fixed quotas on calls per account per Region. These quotas apply to the total calls to the service through any available means, including the console, CLI, AWS CloudFormation, the REST API, and the SDKs. These quotas can't be changed.

Call type	Quota per account per Region
Maximum number of calls to `AssociateWebACL`	1 request every 2 seconds

Call type	Quota per account per Region
Maximum number of calls to `DisassociateWebACL`	1 request every 2 seconds
Maximum number of calls to `GetWebACLForResource`	1 request per second
Maximum number of calls to `ListResourcesForWebACL`	1 request per second
Maximum number of calls to `CreateWebACLMigrationStack`	1 request per second
Maximum number of calls to `GetChangeToken`	10 requests per second
Maximum number of calls to `GetChangeTokenStatus`	1 request per second
Maximum number of calls to any individual `List` action, if no other quota is defined for it	5 requests per second
Maximum number of calls to any individual `Create`, `Put`, `Get`, or `Update` action, if no other quota is defined for it	1 request per second

AWS Shield

Protection against Distributed Denial of Service (DDoS) attacks is of primary importance for your internet-facing applications. When you build your application on AWS, you can make use of protections that AWS provides at no additional cost. Additionally, you can use the AWS Shield Advanced managed threat protection service to improve your security posture with additional DDoS detection, mitigation, and response capabilities.

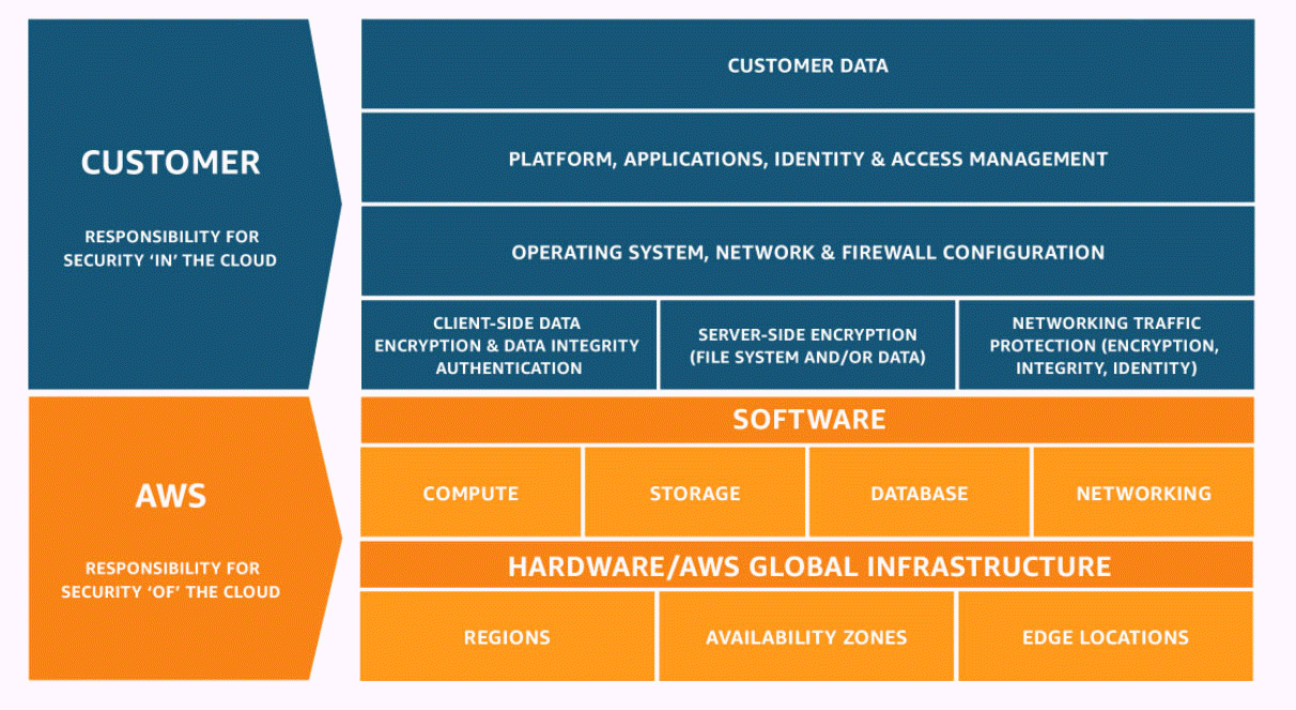
AWS is committed to providing you with the tools, best practices, and services to help ensure high availability, security, and resiliency in your defense against bad actors on the internet. This guide is provided to help IT decision makers and security engineers understand how to use Shield and Shield Advanced to better protect their applications from DDoS attacks and other external threats.

When you build your application on AWS, you receive automatic protection by AWS against common volumetric DDoS attack vectors, like UDP reflection attacks and TCP SYN floods. You can leverage these protections to ensure the availability of the applications that you run on AWS by designing and configuring your architecture for DDoS resiliency.

This guide provides recommendations that can help you design, create, and configure your application architectures for DDoS resiliency. Applications that adhere to the best practices provided in this guide can benefit from an improved continuity of availability when they are targeted by larger DDoS attacks and by wider ranges of DDoS attack vectors. Additionally, this guide shows you how to use Shield Advanced to implement an optimized DDoS protection posture for your critical applications. These include applications for which you've guaranteed a certain level of availability to your customers and those that require operational support from AWS during DDoS events.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. The effectiveness of our security is regularly tested and verified by third-party auditors as part of the AWS compliance programs. To learn about the compliance programs that apply to Shield Advanced, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your organization's requirements, and applicable laws and regulations.



How AWS Shield and Shield Advanced work

This page explains the difference between AWS Shield Standard and AWS Shield Advanced. It also describes the classes of attacks that Shield detects.

AWS Shield Standard and AWS Shield Advanced provide protections against Distributed Denial of Service (DDoS) attacks for AWS resources at the network and transport layers (layer 3 and 4) and the application layer (layer 7). A DDoS attack is an attack in which multiple compromised systems try to flood a target with traffic. A DDoS attack can prevent legitimate end users from accessing the target services and can cause the target to crash due to overwhelming traffic volume.

AWS Shield provides protection against a wide range of known DDoS attack vectors and zero-day attack vectors. Shield detection and mitigation is designed to provide coverage against threats even if they are not explicitly known to the service at the time of detection.

Shield Standard is provided automatically and at no extra charge when you use AWS. For higher levels of protection against attacks, you can subscribe to AWS Shield Advanced.

Classes of attacks that Shield detects include the following:

- **Network volumetric attacks (layer 3)** – This is a sub category of infrastructure layer attack vectors. These vectors attempt to saturate the capacity of the targeted network or resource, to deny service to legitimate users.

- **Network protocol attacks (layer 4)** – This is a sub category of infrastructure layer attack vectors. These vectors abuse a protocol to deny service to the targeted resource. A common example of a network protocol attack is a TCP SYN flood, which can exhaust connection state on resources like servers, load balancers, or firewalls. A network protocol attack can also be volumetric. For example, a larger TCP SYN flood may intend to saturate the capacity of a network while also exhausting the state of the targeted resource or intermediate resources.
- **Application layer attacks (layer 7)** – This category of attack vector attempts to deny service to legitimate users by flooding an application with queries that are valid for the target, such as web request floods.

Contents

- AWS Shield Standard overview
- AWS Shield Advanced overview
- List of AWS resources that AWS Shield Advanced protects
- AWS Shield Advanced capabilities and options
- Deciding whether to subscribe to AWS Shield Advanced and apply additional protections
- Examples of DDoS attacks
- How AWS Shield detects events
  - AWS Shield detection logic for infrastructure layer threats (layer 3 and layer 4)
  - Shield Advanced detection logic for application layer threats (layer 7)
  - Shield Advanced detection logic for multiple resources in an application
- How AWS Shield mitigates events
  - List of AWS Shield DDoS mitigation features
  - AWS Shield mitigation logic for CloudFront and Route 53
  - AWS Shield mitigation logic for AWS Regions
  - AWS Shield mitigation logic for AWS Global Accelerator standard accelerators
  - AWS Shield Advanced mitigation logic for Elastic IPs
  - AWS Shield Advanced mitigation logic for web applications

AWS Shield Standard overview

AWS Shield is a managed threat protection service that protects the perimeter of your application. The perimeter is the first point of entry for application traffic coming from outside the AWS network.

To determine where your application perimeter lies, consider how users access your application from the internet. If the first point of entry is in an AWS Region, then the application perimeter is your Amazon Virtual Private Cloud (VPC). If users are directed to your application by Amazon Route 53, and first access the application using Amazon CloudFront or AWS Global Accelerator, then the application perimeter begins at the edge of the AWS network.

Shield provides DDoS detection and mitigation benefits for all applications running on AWS, but the decisions that you make when you design your application architecture will influence your level of DDoS resiliency. DDoS Resiliency is your application's ability to continue operating within expected parameters during an attack.

All AWS customers benefit from the automatic protection of Shield Standard, at no additional charge. Shield Standard defends against the most common, frequently occurring network and transport layer DDoS attacks that target your website or applications. While Shield Standard helps protect all AWS customers, you get particular benefit with Amazon Route 53 hosted zones, Amazon CloudFront distributions, and AWS Global Accelerator standard accelerators. These resources receive comprehensive availability protection against all known network and transport layer attacks.

AWS Shield Advanced overview

AWS Shield Advanced is a managed service that helps you protect your application against external threats, like DDoS attacks, volumetric bots, and vulnerability exploitation attempts. For higher levels of protection against attacks, you can subscribe to AWS Shield Advanced.

When you subscribe to Shield Advanced and add protection to your resources, Shield Advanced provides expanded DDoS attack protection for those resources. The protections that you receive from Shield Advanced can vary depending on your architecture and configuration choices. Use the information in this guide to build and protect resilient applications using Shield Advanced, and to escalate when you need expert help.

Shield Advanced subscriptions and AWS WAF costs

Your Shield Advanced subscription covers the costs of using standard AWS WAF capabilities for resources that you protect with Shield Advanced. The standard AWS WAF fees that are covered by your Shield Advanced protections are the cost per protection pack (web ACL), the cost per rule, and the base price per million requests for web request inspection, up to 1,500 WCUs and up to the default body size.

Enabling Shield Advanced automatic application layer DDoS mitigation adds a rule group to your protection pack (web ACL) that uses 150 web ACL capacity units (WCUs). These WCUs count against the WCU usage in your protection pack (web ACL). For more information, see Automating application layer DDoS mitigation with Shield Advanced, Protecting the application layer with the Shield Advanced rule group, and Web ACL capacity units (WCUs) in AWS WAF.

Your subscription to Shield Advanced does not cover the use of AWS WAF for resources that you do not protect using Shield Advanced. It also does not cover any additional non-standard AWS WAF costs for protected resources. Examples of non-standard AWS WAF costs are those for Bot Control, for the CAPTCHA rule action, for web ACLs that use more than 1,500 WCUs, and for inspecting the request body beyond the default body size. The full list is provided on the AWS WAF pricing page. Your subscription to Shield Advanced includes access to the Layer 7 Anti-DDoS Amazon Managed Rule group. As part of your subscription, you will get up to 50 billion requests to Shield Advanced protected AWS WAF resources in a calendar month. Requests beyond 50 billion will be billed as per the AWS Shield Advanced pricing page.

For full information and pricing examples, see Shield Pricing and AWS WAF Pricing.

Shield Advanced subscription billing

If you're an AWS Channel Reseller, talk to your account team for information and guidance. This billing information is for customers that are not AWS Channel Resellers.

For all others, the following subscription and billing guidelines apply:

- For accounts that are members of an AWS Organizations organization, AWS bills the Shield Advanced subscriptions against the payer account for the organization, regardless of whether the payer account itself is subscribed.
- When you subscribe multiple accounts that are in the same AWS Organizations consolidated billing account family, one subscription price covers all subscribed accounts in the family. The organization must own all of the AWS accounts and all of their resources.
- When you subscribe multiple accounts for multiple organizations, you can still pay one subscription fee across all of the organizations, accounts, and resources providing you own all

of them. Contact your account manager or AWS support and request a fee waiver on the AWS Shield Advanced subscription charges for all but one of the organizations.

For detailed pricing information and examples, see AWS Shield Pricing.

List of AWS resources that AWS Shield Advanced protects

> **Note**
>
> Shield Advanced protections are only enabled for resources that you have explicitly specified in Shield Advanced or that you protect through an AWS Firewall Manager Shield Advanced policy. Shield Advanced doesn't automatically protect your resources.

You can use Shield Advanced for advanced monitoring and protection with the following resource types:

- Amazon CloudFront distributions. For CloudFront continuous deployment, Shield Advanced protects any staging distribution that's associated with a protected primary distribution.
- Amazon Route 53 hosted zones.
- AWS Global Accelerator standard accelerators.
- Amazon EC2 Elastic IP addresses. Shield Advanced protects the resources that are associated with protected Elastic IP addresses.
- Amazon EC2 instances, through association to Amazon EC2 Elastic IP addresses.
- The following Elastic Load Balancing (ELB) load balancers:
  - Application Load Balancers.
  - Classic Load Balancers.
  - Network Load Balancers, through associations to Amazon EC2 Elastic IP addresses.

For additional information about protections for these resource types, see List of resources that AWS Shield Advanced protects.

AWS Shield Advanced capabilities and options

AWS Shield Advanced subscription includes the following capabilities and options. These supplement the DDoS detection and mitigation capabilities that you already receive with AWS.

- **AWS WAF integration** – Shield Advanced uses AWS WAF web ACLs, rules, and rule groups as part of its application layer protections. For more information about AWS WAF, see How AWS WAF works.

  > **Note**
  >
  > Your Shield Advanced subscription covers the costs of using standard AWS WAF capabilities for resources that you protect with Shield Advanced. The standard AWS WAF fees that are covered by your Shield Advanced protections are the cost per protection pack (web ACL), the cost per rule, and the base price per million requests for web request inspection, up to 1,500 WCUs and up to the default body size.
  > Enabling Shield Advanced automatic application layer DDoS mitigation adds a rule group to your protection pack (web ACL) that uses 150 web ACL capacity units (WCUs). These WCUs count against the WCU usage in your protection pack (web ACL). For more information, see Automating application layer DDoS mitigation with Shield Advanced, Protecting the application layer with the Shield Advanced rule group, and Web ACL capacity units (WCUs) in AWS WAF.
  > Your subscription to Shield Advanced does not cover the use of AWS WAF for resources that you do not protect using Shield Advanced. It also does not cover any additional non-standard AWS WAF costs for protected resources. Examples of non-standard AWS WAF costs are those for Bot Control, for the CAPTCHA rule action, for web ACLs that use more than 1,500 WCUs, and for inspecting the request body beyond the default body size. The full list is provided on the AWS WAF pricing page. Your subscription to Shield Advanced includes access to the Layer 7 Anti-DDoS Amazon Managed Rule group. As part of your subscription, you will get up to 50 billion requests to Shield Advanced protected AWS WAF resources in a calendar month. Requests beyond 50 billion will be billed as per the AWS Shield Advanced pricing page.
  > For full information and pricing examples, see Shield Pricing and AWS WAF Pricing.

- **Automatic application layer DDoS mitigation** – You can configure Shield Advanced to respond automatically to mitigate application layer (layer 7) attacks against your protected resources. With automatic mitigation, Shield Advanced enforces AWS WAF rate limiting on requests from known DDoS sources, and it automatically adds and manages custom AWS WAF protections in response to detected DDoS attacks. You can configure automatic mitigation to count or block the web requests that are part of an attack.

  For more information, see Automating application layer DDoS mitigation with Shield Advanced.

- **Health-based detection** – You can use Amazon Route 53 health checks with Shield Advanced to inform event detection and mitigation. Health checks monitor your application according to your specifications, reporting healthy when your specifications are met and unhealthy when they aren't. Using health checks with Shield Advanced helps prevent false positives and provides faster detection and mitigation when a protected resource is unhealthy. You can use health-based detection for any resource type except Route 53 hosted zones. Shield Advanced proactive engagement is available only for resources that have health-based detection enabled.

  For more information, see Health-based detection using health checks with Shield Advanced and Route 53.

- **Protection groups** – You can use protection groups to create logical groupings of your protected resources, for enhanced detection and mitigation of the group as a whole. You can define the criteria for membership in a protection group so that newly protected resources are automatically included. A protected resource can belong to multiple protection groups.

  For more information, see Grouping your AWS Shield Advanced protections.

- **Enhanced visibility into DDoS events and attacks** – Shield Advanced gives you access to advanced, real-time metrics and reports for extensive visibility into events and attacks on your protected AWS resources. You can access this information through the Shield Advanced API and console, and through Amazon CloudWatch metrics.

  For more information, see Visibility into DDoS events with Shield Advanced.

- **Centralized management of Shield Advanced protections by AWS Firewall Manager** – You can use Firewall Manager to automatically apply Shield Advanced protections to your new accounts and resources and to deploy AWS WAF rules to your web ACLs. Firewall Manager Shield Advanced protection policies are included at no additional charge for Shield Advanced customers. You can also centralize your Shield Advanced monitoring activities for your accounts by using Firewall Manager with an Amazon Simple Notification Service (SNS) topic or AWS Security Hub CSPM.

  For more information about using Firewall Manager to manage Shield Advanced protections, see AWS Firewall Manager and Using AWS Shield Advanced policies in Firewall Manager. For information about Firewall Manager pricing, see AWS Firewall Manager Pricing.

- **AWS Shield Response Team (SRT)** – The SRT has deep experience in protecting AWS, Amazon.com, and its subsidiaries. As an AWS Shield Advanced customer, you can contact the SRT at any time for assistance during a DDoS attack that affects the availability of your application. You can also work with the SRT to create and manage custom mitigations for your resources.

To use the services of the SRT, you must also be subscribed to the Business Support plan or the Enterprise Support plan.

For more information, see Managed DDoS event response with Shield Response Team (SRT) support.

- **Proactive engagement** – With proactive engagement, the Shield Response Team (SRT) contacts you directly if the Amazon Route 53 health check that you have associated with your protected resource becomes unhealthy during an event that's detected by Shield Advanced. This gives you quicker engagement with experts when the availability of your application might be affected by a suspected attack.

  For more information, see Setting up proactive engagement for the SRT to contact you directly.
- **Cost protection opportunities** – Shield Advanced offers some cost protection against spikes in your AWS bill that might result from a DDoS attack against your protected resources. This can include coverage for spikes in Shield Advanced data transfer out (DTO) usage fees. Shield Advanced provides any cost protection in the form of Shield Advanced service credits.

  For more information, see Requesting a credit in AWS Shield Advanced after an attack.

Deciding whether to subscribe to AWS Shield Advanced and apply additional protections

Review the scenarios in this section for help deciding which accounts to subscribe to AWS Shield Advanced and where to apply additional protections. With Shield Advanced, you pay one monthly subscription fee for all accounts created under a consolidated billing account, plus usage fees based on GB of data transferred out. For information about Shield Advanced pricing, see AWS Shield Advanced Pricing.

To protect an application and its resources with Shield Advanced, you subscribe the accounts that manage the application to Shield Advanced and then you add protections to the application's resources. For information about subscribing accounts and protecting resources, see Setting up AWS Shield Advanced.

Shield Advanced subscriptions and AWS WAF costs

Your Shield Advanced subscription covers the costs of using standard AWS WAF capabilities for resources that you protect with Shield Advanced. The standard AWS WAF fees that are covered by your Shield Advanced protections are the cost per protection pack (web ACL), the cost per rule, and

the base price per million requests for web request inspection, up to 1,500 WCUs and up to the default body size.

Enabling Shield Advanced automatic application layer DDoS mitigation adds a rule group to your protection pack (web ACL) that uses 150 web ACL capacity units (WCUs). These WCUs count against the WCU usage in your protection pack (web ACL). For more information, see Automating application layer DDoS mitigation with Shield Advanced, Protecting the application layer with the Shield Advanced rule group, and Web ACL capacity units (WCUs) in AWS WAF.

Your subscription to Shield Advanced does not cover the use of AWS WAF for resources that you do not protect using Shield Advanced. It also does not cover any additional non-standard AWS WAF costs for protected resources. Examples of non-standard AWS WAF costs are those for Bot Control, for the CAPTCHA rule action, for web ACLs that use more than 1,500 WCUs, and for inspecting the request body beyond the default body size. The full list is provided on the AWS WAF pricing page. Your subscription to Shield Advanced includes access to the Layer 7 Anti-DDoS Amazon Managed Rule group. As part of your subscription, you will get up to 50 billion requests to Shield Advanced protected AWS WAF resources in a calendar month. Requests beyond 50 billion will be billed as per the AWS Shield Advanced pricing page.

For full information and pricing examples, see Shield Pricing and AWS WAF Pricing.

Shield Advanced subscription billing

If you're an AWS Channel Reseller, talk to your account team for information and guidance. This billing information is for customers that are not AWS Channel Resellers.

For all others, the following subscription and billing guidelines apply:

- For accounts that are members of an AWS Organizations organization, AWS bills the Shield Advanced subscriptions against the payer account for the organization, regardless of whether the payer account itself is subscribed.
- When you subscribe multiple accounts that are in the same AWS Organizations consolidated billing account family, one subscription price covers all subscribed accounts in the family. The organization must own all of the AWS accounts and all of their resources.
- When you subscribe multiple accounts for multiple organizations, you can still pay one subscription fee across all of the organizations, accounts, and resources providing you own all of them. Contact your account manager or AWS support and request a fee waiver on the AWS Shield Advanced subscription charges for all but one of the organizations.

For detailed pricing information and examples, see AWS Shield Pricing.

Identifying the applications to protect

Consider implementing Shield Advanced protections for applications where you need any of the following:

- Guaranteed availability for the users of the application.
- Rapid access to DDoS mitigation experts if the application is affected by a DDoS attack.
- Awareness by AWS that the application might be affected by a DDoS attack and notification of attacks from AWS and escalation to your security or operations teams.
- Predictability in your cloud costs, including when a DDoS attack affects your use of AWS services.

If an application or its resources require any of the above, consider creating subscriptions for the related accounts.

Identifying the resources to protect

For each subscribed account, consider adding a Shield Advanced protection to each resource that has any of the following characteristics:

- The resource serves external users on the internet.
- The resource is exposed to the internet and is also part of a critical application. Consider every exposed resource, regardless of whether you intend it to be accessed by users on the internet.
- The resource is protected by an AWS WAF web ACL.

To learn more about creating and managing protections for your resources, see Resource protections in AWS Shield Advanced.

Additionally, follow the recommendations in this guide to help ensure that you architect your application for DDoS resiliency and that you have properly configured the features of Shield Advanced for optimal protections.

Examples of DDoS attacks

AWS Shield Advanced provides expanded protection against many types of attacks.

The following list describes some common attack types:

User Datagram Protocol (UDP) reflection attacks

In UDP reflection attacks, an attacker can spoof the source of a request and use UDP to elicit a large response from the server. The extra network traffic directed towards the spoofed, attacked IP address can slow the targeted server and prevent legitimate end users from accessing needed resources.

TCP SYN flood

The intent of an TCP SYN flood attack is to exhaust the available resources of a system by leaving connections in a half-open state. When a user connects to a TCP service like a web server, the client sends a TCP SYN packet. The server returns an acknowledgment, and the client returns its own acknowledgement, completing the three-way handshake. In a TCP SYN flood, the third acknowledgment is never returned, and the server is left waiting for a response. This can prevent other users from connecting to the server.

DNS query flood

In a DNS query flood, an attacker uses multiple DNS queries to exhaust the resources of a DNS server. AWS Shield Advanced can help provide protection against DNS query flood attacks on Route 53 DNS servers.

HTTP flood/cache-busting (layer 7) attacks

With an HTTP flood, including GET and P0ST floods, an attacker sends multiple HTTP requests that appear to be from a real user of the web application. Cache-busting attacks are a type of HTTP flood that uses variations in the HTTP request's query string that prevent use of edge-located cached content and forces the content to be served from the origin web server, causing additional and potentially damaging strain on the origin web server.

How AWS Shield detects events

AWS operates service-level detection systems for the AWS network and individual AWS services, to ensure that they remain available during a DDoS attack. Additionally, resource-level detection systems monitor each individual AWS resource to ensure that traffic toward the resource remains within expected parameters. This combination protects both the targeted AWS resource and AWS services, by applying mitigations that drop known bad packets, highlight potentially malicious traffic, and prioritize traffic from end users.

Detected events appear in your Shield Advanced event summaries, attack details, and Amazon CloudWatch metrics as either the name of the DDoS attack vector or as `Volumetric` if the

evaluation was based on traffic volume instead of signature. For more information on the attack vector dimensions that are available within the `DDoSDetected` CloudWatch metric, see AWS Shield Advanced metrics.

Topics

- AWS Shield detection logic for infrastructure layer threats (layer 3 and layer 4)
- Shield Advanced detection logic for application layer threats (layer 7)
- Shield Advanced detection logic for multiple resources in an application

AWS Shield detection logic for infrastructure layer threats (layer 3 and layer 4)

This page explains how event detection works for the infrastructure (network and transport) layers.

The detection logic used to protect targeted AWS resources against DDoS attacks in the infrastructure layers (layer 3 and layer 4) depends on the resource type and whether the resource is protected with AWS Shield Advanced.

Detection for Amazon CloudFront and Amazon Route 53

When you serve your web application with CloudFront and Route 53, all packets to the application are inspected by a fully inline DDoS mitigation system, which does not introduce any observable latency. DDoS attacks against CloudFront distributions and Route 53 hosted zones are mitigated in real time. These protections apply regardless of whether you use AWS Shield Advanced.

Follow the best practice of using CloudFront and Route 53 as the entry point of your web application wherever possible for the fastest detection and mitigation of DDoS events.

Detection for AWS Global Accelerator and regional services

Resource-level detection protects AWS Global Accelerator standard accelerators and resources that are launched in AWS Regions, like Classic Load Balancers, Application Load Balancers, and Elastic IP addresses (EIPs). These resource types are monitored for traffic elevations that may indicate the presence of a DDoS attack that requires a mitigation. Every minute, traffic to each AWS resource is evaluated. If traffic to a resource is elevated, additional checks are performed to measure the capacity of the resource.

Shield performs the following standard checks:

- **Amazon Elastic Compute Cloud (Amazon EC2) instances, EIPs attached to Amazon EC2 instances** – Shield retrieves capacity from the protected resource. The capacity depends on the

target's instance type, instance size, and other factors such as whether the instance is using enhanced networking.

- **Classic Load Balancers and Application Load Balancers** – Shield retrieves capacity from the targeted load balancer node.
- **EIPs attached to Network Load Balancers** – Shield retrieves capacity from the targeted load balancer. The capacity is independent of the target load balancer's group configuration.
- **AWS Global Accelerator standard accelerators** – Shield retrieves capacity, which is based on the endpoint configuration.

These evaluations occur across multiple dimensions of network traffic, such as port and protocol. If the capacity of the targeted resource is exceeded, Shield places a DDoS mitigation. The mitigations placed by Shield will reduce DDoS traffic, but might not eliminate it. Shield may also place a mitigation if a fraction of the resource's capacity is exceeded on a traffic dimension that's consistent with known DDoS attack vectors. Shield places this mitigation with a limited time to live (TTL), which it extends as long as the attack is ongoing.

> **Note**
>
> Mitigations placed by Shield will reduce DDoS traffic, but may not eliminate it. You can augment Shield with solutions like AWS Network Firewall or an on-host firewall like iptables to prevent your application from processing traffic that is not valid for your application or was not generated by legitimate end users.

Shield Advanced protections add the following to the existing Shield detection activities:

- **Lower detection thresholds** – Shield Advanced places mitigations at one half of the calculated capacity. This can provide faster mitigations for attacks that ramp up slowly and mitigation of attacks that have a more ambiguous volumetric signature.
- **Intermittent attack protection** – Shield Advanced places mitigations with an exponentially increasing time to live (TTL), based on the frequency and duration of attacks. This keeps mitigations in place longer when a resource is frequently targeted and when an attack occurs in short bursts.
- **Health-based detection** – When you associate a Route 53 health check with a Shield Advanced protected resource, the status of the health check is used in the detection logic. During a detected event, if the health check is healthy, Shield Advanced requires greater confidence that

the event is an attack before placing a mitigation. If instead the health check is unhealthy, Shield Advanced might place a mitigation even before confidence has been established. This feature helps avoid false positives and provides quicker reactions to attacks that affect your application. For information about health checks with Shield Advanced, see Health-based detection using health checks with Shield Advanced and Route 53.

Shield Advanced detection logic for application layer threats (layer 7)

This page explains how event detection works for the application layer.

AWS Shield Advanced provides web application layer detection for protected Amazon CloudFront distributions and Application Load Balancers. When you protect these resource types with Shield Advanced, you can associate an AWS WAF web ACL with your protection to enable web application layer detection. Shield Advanced consumes request data for the associated web ACL and builds a traffic baseline for your application. Web application layer detection relies on the native integration between Shield Advanced and AWS WAF. To learn more about application layer protections, including associating an AWS WAF web ACL to a Shield Advanced protected resource, see Protecting the application layer (layer 7) with AWS Shield Advanced and AWS WAF.

For web application layer detection, Shield Advanced monitors application traffic and compares it to historic baselines looking for anomalies. This monitoring covers total volume and the composition of traffic. During a DDoS attack, we expect both the volume and composition of traffic to change, and Shield Advanced requires a statistically significant deviation in both to declare an event.

Shield Advanced performs its measurements against historical time windows. This approach reduces false positive notifications from legitimate changes in traffic volume or from changes in traffic that match an expected pattern, such as a sale that's offered at the same time each day.

> **Note**
>
> Avoid false positives in your Shield Advanced protections by giving Shield Advanced time to establish baselines that represent normal, legitimate traffic patterns. Shield Advanced begins to collect information for its baseline when you associate a web ACL with your protected resource. Associate a web ACL with your protected resource at least 24 hours before any planned event that might cause unusual patterns in your web traffic. Shield Advanced web application layer detection is most accurate when it has observed 30 days of normal traffic.

The time that Shield Advanced takes to detect an event is affected by how much change it observes in the volume of traffic. For lower volume changes, Shield Advanced observes traffic for a longer period, in order to build confidence that an event is occurring. For higher volume changes, Shield Advanced detects and reports an event more quickly.

A rate-based rule in your web ACL, whether added by you or by the Shield Advanced automatic application layer mitigation feature, can mitigate an attack before it reaches a detectable level. For more information about automatic application layer DDoS mitigation, see Automating application layer DDoS mitigation with Shield Advanced.

> **Note**
>
> You can architect your application to scale in response to elevated traffic or load to ensure that it is not affected by smaller request floods. With Shield Advanced, your protected resources are covered by cost protection. This helps protect you against unexpected increases in your cloud bill that might occur as the result of a DDoS attack. To learn more about Shield Advanced cost protection, see Requesting a credit in AWS Shield Advanced after an attack.

Shield Advanced detection logic for multiple resources in an application

This page explains how event detection works for multiple resources in an application.

You can use AWS Shield Advanced protection groups to create collections of protected resources that are part of the same application. You can choose which protected resources to place in a group or indicate that all resources of the same type should be treated as one group. For example, you might create a group of all Application Load Balancers. When you create a protection group, Shield Advanced detection aggregates all traffic for the protected resources within the group. This is useful if you have many resources that each have a small amount of traffic, but with a large aggregated volume. You can also use protection groups to preserve application baselines, for the case of blue-green deployments where traffic is transferred between protected resources.

You can choose to aggregate the traffic in your protection group in one of the following ways:

- **Sum** – This aggregation combines all traffic across resources in the protection group. You can use this aggregation to ensure that newly created resources have an existing baseline and to reduce detection sensitivity, which can help prevent false positives.

- **Mean** – This aggregation uses the average of all traffic across the protection group. You can use this aggregation for applications where traffic across resources is uniform, like load balancers.
- **Max** – This aggregation uses the highest traffic of any resource in the protection group. You can use this aggregation when there are multiple tiers of an application in a protection group. For example, you may have a protection group that includes a CloudFront distribution, its Application Load Balancer origin, and the Application Load Balancer's Amazon EC2 instance targets.

You can also use protection groups to improve the speed at which Shield Advanced places mitigations, for attacks that targets multiple internet-facing Elastic IPs or AWS Global Accelerator standard accelerators. When one resource in a protection group is targeted, Shield Advanced establishes confidence for the other resources in the group. This places Shield Advanced detection on alert and can reduce the time required to create additional mitigations.

To learn more about protection groups, see Grouping your AWS Shield Advanced protections.

How AWS Shield mitigates events

This page introduces how AWS Shield event mitigation works.

The mitigation logic that protects your application can vary depending on your application architecture. When you protect a web application with Amazon CloudFront and Amazon Route 53, you benefit from mitigations that are specific to web and DNS use cases and that protect all traffic for the services. When your application's entry point is a resource that runs in an AWS Region, the mitigation logic varies depending on the service, the resource type, and your use of AWS Shield Advanced.

AWS DDoS mitigation systems are developed by Shield engineers and they're closely integrated with AWS services. The engineers take into account aspects of your architecture such as the capacity and health of targeted resources. Shield engineers continually monitor the efficacy and performance of DDoS mitigation systems and are able to respond quickly when new threats are discovered or anticipated.

You can architect your application to scale in response to elevated traffic or load, to help ensure that it's not affected by smaller request floods. If you use Shield Advanced to protect your resources, you receive coverage against unexpected increases in your cloud bill that might occur as the result of a DDoS attack.

Infrastructure mitigations

For infrastructure layer attacks, AWS Shield DDoS mitigation systems are present at the AWS network border and at AWS edge locations. The placement of multiple levels of security controls throughout the AWS infrastructure provides defense-in-depth to your cloud applications.

Shield maintains DDoS mitigation systems at all points of ingress from the internet. When Shield detects a DDoS attack, for each point of ingress, it reroutes the traffic through the DDoS mitigation systems in the same location. This doesn't introduce any observable additional latency, and provides a mitigation capacity of more than 100 TeraBits Per Second (Tbps) across all AWS Regions and all edge locations. Shield protects your resource availability without rerouting traffic to external or remote scrubbing centers, which could increase latency.

- At the AWS network border, for any AWS service or resource, DDoS mitigation systems mitigate infrastructure layer attacks coming from the internet. The systems perform their mitigations when signaled by Shield detection or by an engineer on the Shield Response Team (SRT).
- At AWS edge locations, DDoS mitigation systems continuously inspect every packet that's forwarded to Amazon CloudFront distributions and Amazon Route 53 hosted zones, regardless of their origin. When needed, the systems apply mitigations that are specifically designed for web and DNS traffic. An added benefit of using Amazon CloudFront and Amazon Route 53 to protect your web applications is that DDoS attacks are immediately mitigated, without requiring a signal from Shield detection.

Application layer mitigations

Shield Advanced provides web application layer mitigations for the Amazon CloudFront distributions and Application Load Balancers where you've enabled Shield Advanced protections. When you enable protection, you associate an AWS WAF web ACL with the resource, to enable web application layer detection. Additionally, you have the option of enabling automatic application layer mitigation, which instructs Shield Advanced to manage protections for you during a DDoS attack.

Shield only provides custom mitigations for application layer attacks on resources for which you've enabled Shield Advanced and automatic application layer mitigation. With automatic mitigation, Shield Advanced enforces AWS WAF rate limiting on requests from known DDoS sources, and it automatically adds and manages custom AWS WAF protections in response to detected DDoS attacks. For detailed information about mitigations of this type, see How Shield Advanced manages automatic mitigation.

A rate-based rule in your web ACL, whether added by you or added by the Shield Advanced automatic application layer mitigation feature, can mitigate an attack before it reaches a detectable level. For more information about detection, see Shield Advanced detection logic for application layer threats (layer 7).

Topics

- List of AWS Shield DDoS mitigation features
- AWS Shield mitigation logic for CloudFront and Route 53
- AWS Shield mitigation logic for AWS Regions
- AWS Shield mitigation logic for AWS Global Accelerator standard accelerators
- AWS Shield Advanced mitigation logic for Elastic IPs
- AWS Shield Advanced mitigation logic for web applications

List of AWS Shield DDoS mitigation features

The main features of AWS Shield DDoS mitigation are the following:

- **Packet validation** – This ensures that every inspected packet conforms to an expected structure and is valid for its protocol. Supported protocol validations include IP, TCP (including header and options), UDP, ICMP, DNS, and NTP.
- **Access Control Lists (ACLs) and shapers** – An ACL evaluates traffic against specific attributes and either drops matching traffic or maps it to a shaper. The shaper limits the packet rate for the matching traffic, dropping excess packets in order to contain the volume that reaches the destination. AWS Shield detection and Shield Response Team (SRT) engineers can provide dedicated rate allocations to expected traffic and more restrictive rate allocations to traffic with attributes that match known DDoS attack vectors. The attributes that an ACL can match include the port, protocol, TCP flags, destination address, source country, and arbitrary patterns in the packet payload.
- **Suspicion scoring** – This uses the knowledge that Shield has of expected traffic to apply a score to every packet. Packets that more closely adhere to patterns of known good traffic are assigned a lower suspicion score. Observation of known bad traffic attributes can increase the suspicion score for a packet. When it's necessary to rate limit packets, Shield drops packets with higher suspicion scores first. This helps Shield to mitigate both known and zero-day DDoS attacks while avoiding false positives.

- **TCP SYN proxy** – This provides protection against TCP SYN floods by sending TCP SYN cookies to challenge new connections before allowing them to pass to the protected service. The TCP SYN proxy provided by Shield DDoS mitigation is stateless, which allows it to mitigate the largest known TCP SYN flood attacks without reaching state exhaustion. This is achieved by integrating with AWS services to hand off connection state instead of maintaining a continuous proxy between the client and the protected service. TCP SYN proxy is currently available on Amazon CloudFront, Amazon Route 53, and AWS Global Accelerator.
- **Rate distribution** – This continuously adjusts per-location shaper values based on the ingress pattern of traffic toward a protected resource. This prevents rate limiting of customer traffic that might not enter the AWS network evenly.

AWS Shield mitigation logic for CloudFront and Route 53

This page explains how Shield DDoS mitigation continually inspects traffic for CloudFront and Route 53. These services operate from a globally distributed network of AWS edge locations that provide you with broad access to Shield's DDoS mitigation capacity and deliver your application from infrastructure that's closer to your end users.

> **Important**
>
> AWS Shield Advanced doesn't support CloudFront tenants.

- **CloudFront** – Shield DDoS mitigations only allow traffic that's valid for web applications to pass through to the service. This provides automatic protection against many common DDoS vectors, like UDP reflection attacks.

  CloudFront maintains persistent connections to your application origin, TCP SYN floods are automatically mitigated through integration with the Shield TCP SYN proxy feature, and Transport Layer Security (TLS) is terminated at the edge. These combined features ensure that your application origin only receives well-formed web requests and that it's protected against lower-layer DDoS attacks, connection floods, and TLS abuse.

  CloudFront uses a combination of DNS traffic direction and anycast routing. These techniques improve the resilience of your application by mitigating attacks close to the source, providing fault isolation, and ensuring access to capacity to mitigate the largest known attacks.

- **Route 53** – Shield mitigations only allow valid DNS requests to reach the service. Shield mitigates DNS query floods using suspicion scoring that prioritizes known good queries and deprioritizes queries that contain suspicious or known DDoS attack attributes.

  Route 53 uses shuffle sharding to provide a unique set of four resolver IP addresses to every hosted zone, for both IPv4 and IPv6. Each IP address corresponds to a different subset of Route 53 locations. Each location subset consists of authoritative DNS servers that only partially overlap with infrastructure in any other subset. This ensures that if a user query fails for any reason, it will be successfully served on retry.

  Route 53 uses anycast routing to direct DNS queries to the nearest edge location, based on network proximity. Anycast also fans out DDoS traffic to many edge locations, which prevents attacks from focusing on a single location.

In addition to the speed of mitigation, CloudFront and Route 53 provide broad access to the globally distributed capacity of Shield. To take advantage of these capabilities, use these services as the entry point of your dynamic or static web applications.

To learn more about using CloudFront and Route 53 to protect web applications, see How to Help Protect Dynamic Web Applications Against DDoS Attacks by Using Amazon CloudFront and Amazon Route 53. To learn more about fault isolation on Route 53, see A Case Study in Global Fault Isolation.

AWS Shield mitigation logic for AWS Regions

This page explains how Shield event mitigation logic works in AWS Regions.

Resources that are launched in AWS Regions are protected by AWS Shield DDoS mitigation systems placed by Shield resource-level detection. Regional resources include Elastic IPs (EIPs), Classic Load Balancers, and Application Load Balancers.

Prior to placing a mitigation, Shield identifies the targeted resource and its capacity. Shield uses the capacity to determine the maximum total traffic that its mitigations should allow to be forwarded to the resource. Access control lists (ACLs) and other shapers within the mitigation might decrease the allowed volumes for some traffic, for example traffic that matches known DDoS attack vectors or that isn't expected to come in large volume. This further limits the amount of traffic that the mitigations allow for UDP reflection attacks or for TCP traffic that has TCP SYN or FIN flags.

Shield determines capacity and places mitigations differently for each resource type.

- For an Amazon EC2 instance, or an EIP that's attached to an Amazon EC2 instance, Shield calculates the capacity based on the instance type and other instance attributes, such as whether the instance has enhanced networking enabled.
- For an Application Load Balancer or Classic Load Balancer, Shield calculates capacity individually for each targeted node of the load balancer. DDoS attack mitigations for these resources are provided by a combination of Shield DDoS mitigations and automatic scaling by the load balancer. When the Shield Response Team (SRT) is engaged on an attack against an Application Load Balancer or Classic Load Balancer resource, they might accelerate scaling as an additional protection measure.
- Shield calculates capacity for some AWS resources is based on the available capacity of the underlying AWS infrastructure. These resource types include Network Load Balancers (NLBs) and resources that route traffic through Gateway Load Balancers or AWS Network Firewall.

> **Note**
>
> Protect your Network Load Balancers by attaching EIPs that are protected by Shield Advanced. You can work with SRT to build custom mitigations that are based on the expected traffic and capacity of the underlying application.

When Shield places a mitigation, the initial rate limits that Shield defines in the mitigation logic are applied equally to every Shield DDoS mitigation system. For example, if Shield places a mitigation with a 100,000 packets per second (pps) limit, it will initially allow 100,000 pps at every location. Then, Shield continuously aggregates mitigation metrics to determine the actual ratio of traffic, and uses the ratio to adapt the rate limit for each location. This prevents false positives and ensures that the mitigations are not overly permissive.

AWS Shield mitigation logic for AWS Global Accelerator standard accelerators

This page explains how Shield event mitigation logic works for AWS Global Accelerator standard accelerators. Shield mitigations only allow valid traffic to reach the listener endpoints of a Global Accelerator standard accelerator.

Standard accelerators are deployed globally, and they provide you with IP addresses that you can use to route traffic to AWS resources in any AWS Region. The rate limits that Shield enforces for a

Global Accelerator mitigation are based on the capacities of the resources to which the standard accelerator routes traffic. Shield places mitigations when the total traffic exceeds the determined rate, and also when a fraction of that rate is exceeded for known DDoS vectors.

When you configure a standard accelerator, you define endpoint groups for each AWS Region where you'll route traffic for your application. When Shield places a mitigation, it calculates the capacity of each endpoint group and updates rate limits at each Shield DDoS mitigation system accordingly. The rate varies for each location, based on assumptions made by Shield about how traffic will route from the internet to your AWS resources. The capacity for an endpoint group is calculated as the number of resources in the group multiplied by the lowest capacity for any resource in the group. At regular intervals, Shield recalculates the capacity for your application and updates the rate limits as needed.

> **Note**
>
> Using traffic dials to change the percentage of traffic that's directed to an endpoint group doesn't change how Shield calculates or distributes rate limits to its DDoS mitigation systems. If you use traffic dials, configure your endpoint groups to mirror one another in terms of resource type and quantity. This helps ensure that the capacity calculated by Shield is representative of the resources that are serving traffic for your application.

For more information about endpoint groups and traffic dials in Global Accelerator, see Endpoint groups in AWS Global Accelerator standard accelerators.

AWS Shield Advanced mitigation logic for Elastic IPs

This page explains how Shield event mitigation logic works for Elastic IPs with AWS Shield Advanced. When you protect an Elastic IP (EIP) with AWS Shield Advanced, Shield Advanced enhances the mitigations that Shield places during a DDoS event.

Shield Advanced DDoS mitigation systems replicate the Network ACL (NACL) configuration for the public subnet to which the EIP is associated. For example, if your NACL is configured to block all UDP traffic, Shield Advanced merges that rule into the mitigations that Shield places.

This additional functionality can help you to avoid availability risks due to traffic that's not valid for your application. You can also use NACLs to block individual source IP addresses or source IP address CIDR ranges. This can be a useful mitigation tool for DDoS attacks that aren't

distributed. It also lets you easily manage your own allow lists or to block IP addresses that shouldn't communicate with your application, without relying on intervention by AWS engineers.

AWS Shield Advanced mitigation logic for web applications

AWS Shield Advanced uses AWS WAF to mitigate web application layer attacks. AWS WAF is included with Shield Advanced at no additional cost.

Standard application layer protection

When you protect an Amazon CloudFront distribution or Application Load Balancer with Shield Advanced, you can use Shield Advanced to associate an AWS WAF web ACL with your protected resource, if you don't already have one associated. If you haven't already configured a web ACL, you can use the Shield Advanced console wizard to create one and add a rate-based rule to it. A rate-based rule limits the number of requests per five minute time window for each IP address, providing basic protections against web application layer request floods. You can configure the rate, starting as low as 10. For more information, see Protecting the application layer with AWS WAF web ACLs and Shield Advanced.

You can also use the AWS WAF service to manage the web ACL. Through AWS WAF, you can expand the web ACL configuration to do things such as inspect specific web request components for string matches or patterns, add custom request and response handling, and match against the geolocation of the request origin. For more information about AWS WAF rules, see AWS WAF rules.

Automatic application layer mitigation

For enhanced protection, enable Shield Advanced automatic application layer mitigation. With this option, Shield Advanced maintains an AWS WAF rate limiting rule for requests from known DDoS sources and it provides custom mitigations for detected DDoS attacks.

When Shield Advanced detects an attack on a protected resource, it attempts to identify an attack signature that isolates the attack traffic from the normal traffic to your application. Shield Advanced evaluates the identified attack signature against the historical traffic patterns for the resource that's under attack, as well as for any other resource that's associated with the same web ACL.

If Shield Advanced determines that the attack signature isolates only the traffic that's involved in the DDoS attack, it implements the signature in AWS WAF rules inside the associated web ACL. You can instruct Shield Advanced to place mitigations that only count the traffic that they match against, or that block it, and you can change the setting at any time. When Shield Advanced determines that its mitigating rules are no longer needed, it removes them from the web ACL. For

more information about application layer event mitigation, see Automating application layer DDoS mitigation with Shield Advanced.

For more information about Shield Advanced application layer mitigations, see Protecting the application layer (layer 7) with AWS Shield Advanced and AWS WAF.

Building basic DDoS resilient architectures with Shield Advanced

This page explains Distributed Denial of Service (DDoS) resiliency and introduces two example architectures.

DDoS resiliency is the ability of your application architecture to withstand DDoS attacks while continuing to serve legitimate end users. An application that is highly resilient can remain available during an attack with minimal impact on performance metrics such as errors or latency. This section shows some common example architectures and describes how to use the DDoS detection and mitigation capabilities that are provided by AWS and Shield Advanced to increase their DDoS resiliency.

The example architectures in this section highlight the AWS services that provide the greatest DDoS resiliency benefits for your deployed applications. The benefits of the highlighted services include the following:

- **Access to globally distributed network capacity** – The services Amazon CloudFront, AWS Global Accelerator, and Amazon Route 53 provide you with access to internet and DDoS mitigation capacity across the AWS global edge network. This is useful in mitigating larger volumetric attacks, which can reach terabits in scale. You can run your application in any AWS Region and use these services to protect availability and optimize performance for your legitimate users.
- **Protection against web application layer DDoS attack vectors** – Web application layer DDoS attacks are best mitigated using a combination of application scale and a web application firewall (WAF). Shield Advanced uses web request inspection logs from AWS WAF to detect anomalies that can be mitigated either automatically or via engagement with the AWS Shield Response Team (SRT). Automatic mitigation is available through deployed AWS WAF rate-based rules and also through the Shield Advanced automatic application layer DDoS mitigation.

In addition to reviewing these examples, review and follow the applicable best practices at AWS Best Practices for DDoS Resiliency.

Topics

- Example Shield Advanced DDoS resiliency architecture for common web applications
- Example Shield Advanced DDoS resiliency architecture for TCP and UDP applications

Example Shield Advanced DDoS resiliency architecture for common web applications

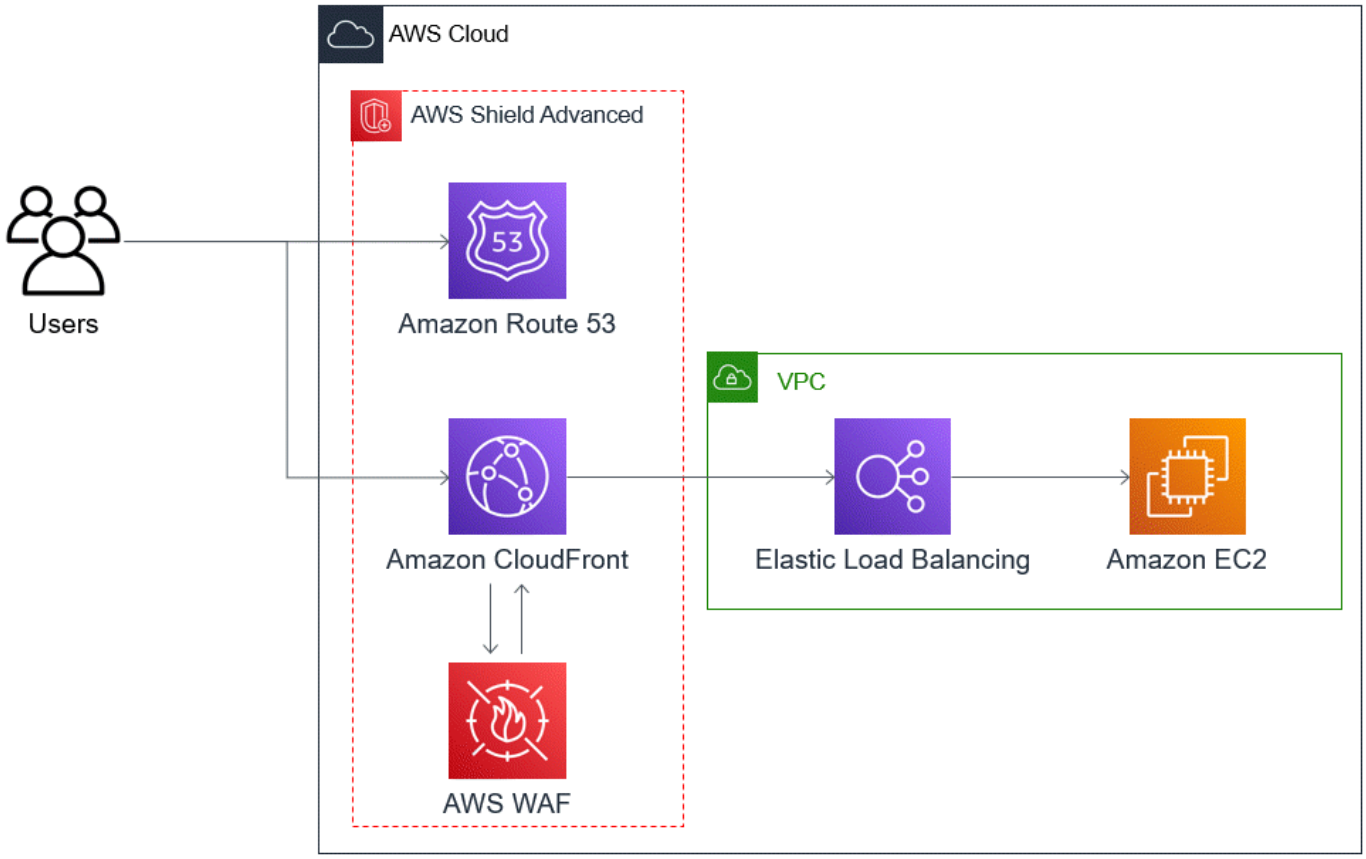
This page provides an example architecture for maximizing resiliency against DDoS attacks with AWS web applications.

You can build a web application in any AWS Region and receive automatic DDoS protection from the detection and mitigation capabilities that AWS provides in the Region.

This example is for architectures that route users to a web application using resources like Classic Load Balancers, Application Load Balancers, Network Load Balancers, AWS Marketplace solutions, or your own proxy layer. You can improve DDoS resiliency by inserting Amazon Route 53 hosted zones, Amazon CloudFront distributions, and AWS WAF web ACLs between these web application resources and your users. These insertions can obfuscate the application origin, serve requests closer to your end users, and detect and mitigate application layer request floods. Applications that serve static or dynamic content to your users with CloudFront and Route 53 are protected by an integrated, fully inline DDoS mitigation system that mitigates infrastructure layer attacks in real time.

With these architectural improvements in place, you can then protect your Route 53 hosted zones and your CloudFront distributions with Shield Advanced. When you protect CloudFront distributions, Shield Advanced prompts you to associate AWS WAF web ACLs and create rate-based rules for them, and gives you the option of enabling automatic application layer DDoS mitigation or proactive engagement. Proactive engagement and automatic application layer DDoS mitigation use Route 53 health checks that you associate with the resource. To learn more about these options, see Resource protections in AWS Shield Advanced.

The following reference diagram depicts this DDoS resilient architecture for a web application.



The benefits that this approach provides to your web application include the following:

- Protection against frequently used infrastructure layer (layer 3 and layer 4) DDoS attacks, without detection delay. In addition, if a resource is frequently targeted, Shield Advanced places mitigations for longer periods of time. Shield Advanced also uses application context inferred from Network ACLs (NACLs) to block unwanted traffic further upstream. This isolates failures closer to their source, minimizing the effect on legitimate users.
- Protection against TCP SYN floods. The DDoS mitigation systems that are integrated with CloudFront, Route 53, and AWS Global Accelerator provide a TCP SYN proxy capability that challenges new connection attempts and only serves legitimate users.
- Protection against DNS application layer attacks, because Route 53 is responsible for serving authoritative DNS responses.
- Protection against web application layer request floods. The rate-based rule that you configure in your AWS WAF web ACL blocks source IPs when they are sending more requests than the rule allows.

- Automatic application layer DDoS mitigation for your CloudFront distributions, if you choose to enable this option. With automatic DDoS mitigation, Shield Advanced maintains a rate-based rule in the distribution's associated AWS WAF web ACL that limits the volume of requests from known DDoS sources. Additionally, when Shield Advanced detects an event that affects the health of your application, it automatically creates, tests, and manages mitigating rules in web ACL.
- Proactive engagement with the Shield Response Team (SRT), if you choose to enable this option. When Shield Advanced detects an event that affects the health of your application, the SRT responds and proactively engages with your security or operations teams using the contact information that you provide. The SRT analyzes patterns in your traffic and can update your AWS WAF rules to block the attack.

Example Shield Advanced DDoS resiliency architecture for TCP and UDP applications

This example shows a DDoS resilient architecture for TCP and UDP applications in an AWS Region that uses Amazon Elastic Compute Cloud (Amazon EC2) instances or Elastic IP (EIP) addresses.

You can follow this general example to improve DDoS resiliency for the following application types:

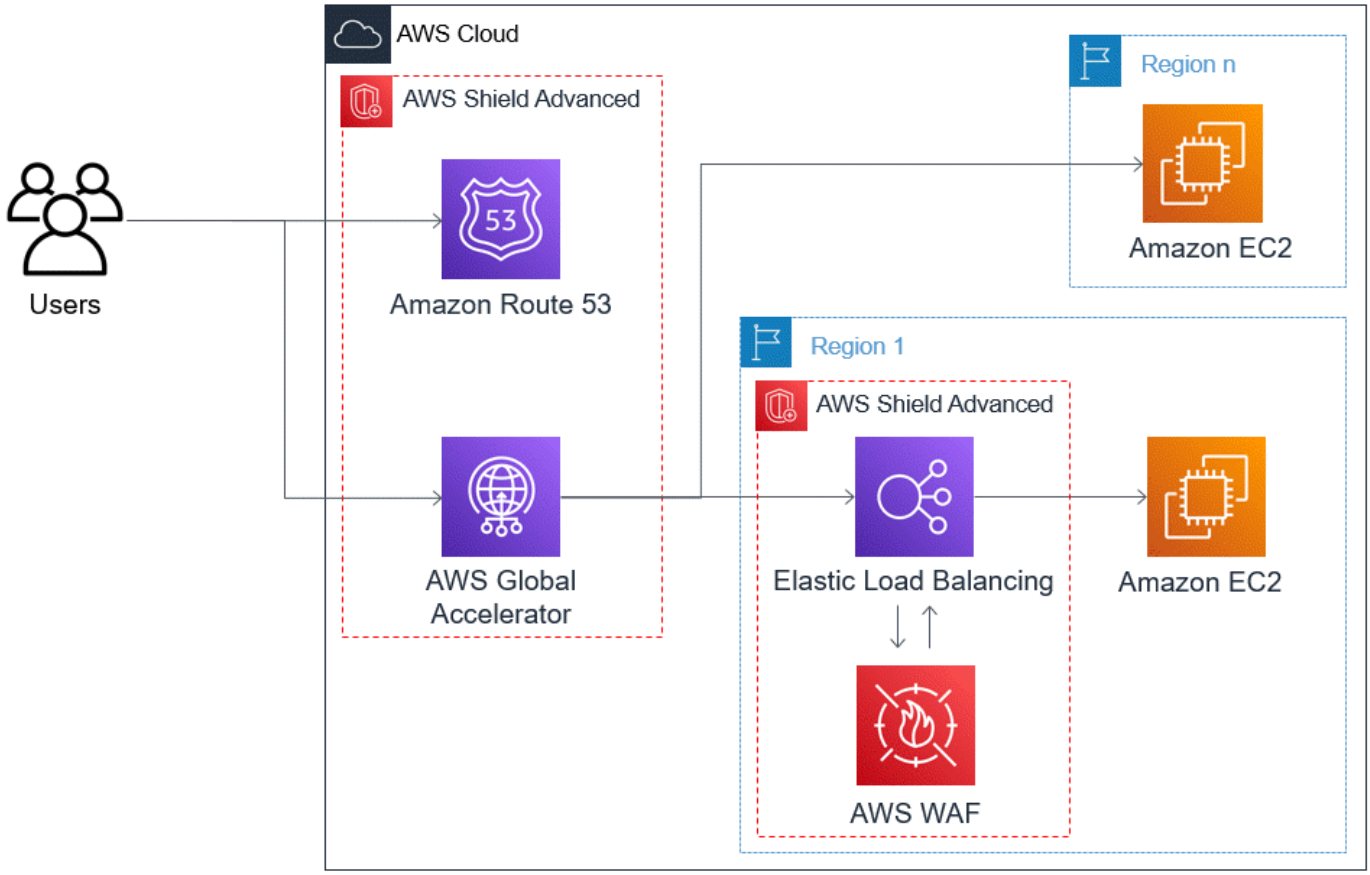
- TCP or UDP applications. For example, applications used for gaming, IoT, and voice over IP.
- Web applications that require static IP addresses or that use protocols that Amazon CloudFront doesn't support. For example, your application might require IP addresses that your users can add to their firewall allow lists, and that aren't used by any other AWS customers.

You can improve DDoS resiliency for these application types by introducing Amazon Route 53 and AWS Global Accelerator. These services can route users to your application and they can provide your application with static IP addresses that are anycast routed across the AWS global edge network. Global Accelerator standard accelerators can improve user latency by up to 60%. If you have a web application, you can detect and mitigate web application layer request floods by running the application on an Application Load Balancer, and then protecting the Application Load Balancer with an AWS WAF web ACL.

After you've built your application, protect your Route 53 hosted zones, Global Accelerator standard accelerators, and any Application Load Balancers with Shield Advanced. When you protect your Application Load Balancers, you can associate AWS WAF web ACLs and create rate-based rules

for them. You can configure proactive engagement with the SRT for both your Global Accelerator standard accelerators and your Application Load Balancers by associating new or existing Route 53 health checks. To learn more about the options, see Resource protections in AWS Shield Advanced.

The following reference diagram depicts an example DDoS resilient architecture for TCP and UDP applications.



The benefits that this approach provides to your application include the following:

- Protection against the largest known infrastructure layer (layer 3 and layer 4) DDoS attacks. If the volume of an attack causes congestion upstream from AWS, the failure will be isolated closer to its source and will have a minimized effect on your legitimate users.
- Protection against DNS application layer attacks, because Route 53 is responsible for serving authoritative DNS responses.
- If you have a web application, this approach provides protection against web application layer request floods. The rate-based rule that you configure in your AWS WAF web ACL blocks source IPs while they are sending more requests than the rule allows.

- Proactive engagement with the Shield Response Team (SRT), if you choose to enable this option for eligible resources. When Shield Advanced detects an event that affects the health of your application, the SRT responds and proactively engages with your security or operations teams using the contact information that you provide.

Combining Shield Advanced with other AWS services

You can use Shield Advanced to protect your resources in many types of scenarios. However, in some cases you should use other services or combine other services with Shield Advanced to offer the best protection. Following are examples of how to use Shield Advanced or other AWS services to help protect your resources.

Goal	Suggested services	Related service documentation
Protect a web application and RESTful APIs against a DDoS attack	Shield Advanced protecting an Amazon CloudFront distribution and an Application Load Balancer	Elastic Load Balancing documentation, Amazon CloudFront Documentation
Protect a TCP-based application against a DDoS attack	Shield Advanced protecting an AWS Global Accelerator standard accelerator; attached to an Elastic IP address	AWS Global Accelerator Documentation, Elastic Load Balancing documentation
Protect a UDP-based game server against a DDoS attack	Shield Advanced protecting an Amazon EC2 instance attached to an Elastic IP address	Amazon Elastic Compute Cloud Documentation

For example, if you use Shield Advanced to protect an Elastic IP address, Shield Advanced protects whatever resource is associated with it. During an attack, Shield Advanced automatically deploys your network ACLs to the border of the AWS network. When your network ACLs are at the border of the network, Shield Advanced can provide protection against larger DDoS events. Typically, network ACLs are applied near your Amazon EC2 instances within your Amazon VPC. The network

ACL can mitigate attacks only as large as your Amazon VPC and instance can handle. If the network interface attached to your Amazon EC2 instance can process up to 10 Gbps, volumes over 10 Gbps slow down and possibly block traffic to that instance. During an attack, Shield Advanced promotes your network ACL to the AWS border, which can process multiple terabytes of traffic. Your network ACL is able to provide protection for your resource well beyond your network's typical capacity. For more information about network ACLs, see Network ACLs.

Setting up AWS Shield Advanced

This tutorial walks you through getting started with AWS Shield Advanced using the Shield Advanced console.

> **Note**
>
> Shield Advanced requires a subscription, while AWS Shield Standard does not. The protections provided by Shield Standard are available free of charge to all AWS customers.

Shield Advanced provides advanced DDoS detection and mitigation protection for network layer (layer 3), transport layer (layer 4), and application layer (layer 7) attacks. For more information about Shield Advanced, see AWS Shield Advanced overview.

The AWS technical community has published an example of an automated process for configuring Shield Advanced using the infrastructure as code (IaC) tools, AWS CloudFormation and Terraform. You can use AWS Firewall Manager with this solution if your accounts are part of an organization in AWS Organizations and if you're protecting any resource types except for Amazon Route 53 or AWS Global Accelerator. To explore this option, see the code repository at aws-samples / aws-shield-advanced-one-click-deployment and the tutorial at One-click deployment of Shield Advanced.

> **Note**
>
> It's important that you fully configure Shield Advanced prior to a Distributed Denial of Service (DDoS) event. Complete the configuration to help ensure that your application is protected and that you are ready to respond if your application is affected by a DDoS attack.

Perform the following steps in sequence to get started using Shield Advanced.

Contents

- Subscribing to AWS Shield Advanced
- Adding and configuring resource protections with Shield Advanced
  - Configuring application layer (layer 7) DDoS protections with AWS WAF
  - Configuring health-based detection for your protections with Shield Advanced and Route 53
  - Configuring alarms and notifications with Shield Advanced and Amazon SNS
  - Reviewing and finishing your protection configuration in Shield Advanced
- Setting up AWS Shield Response Team (SRT) support for DDoS event response
- Creating a DDoS dashboard in CloudWatch and setting CloudWatch alarms

Subscribing to AWS Shield Advanced

This page explains how to subscribe your accounts to Shield Advanced, to start using the service.

You must subscribe to Shield Advanced for each AWS account that you want to protect. You do not need to subscribe to Shield Standard.

Shield Advanced subscription billing

If you're an AWS Channel Reseller, talk to your account team for information and guidance. This billing information is for customers that are not AWS Channel Resellers.

For all others, the following subscription and billing guidelines apply:

- For accounts that are members of an AWS Organizations organization, AWS bills the Shield Advanced subscriptions against the payer account for the organization, regardless of whether the payer account itself is subscribed.
- When you subscribe multiple accounts that are in the same AWS Organizations consolidated billing account family, one subscription price covers all subscribed accounts in the family. The organization must own all of the AWS accounts and all of their resources.
- When you subscribe multiple accounts for multiple organizations, you can still pay one subscription fee across all of the organizations, accounts, and resources providing you own all of them. Contact your account manager or AWS support and request a fee waiver on the AWS Shield Advanced subscription charges for all but one of the organizations.

For detailed pricing information and examples, see AWS Shield Pricing.

Consider simplifying subscriptions with AWS Firewall Manager

If your accounts are part of an organization, we recommend that you use AWS Firewall Manager if you can, to automate your subscriptions and protections for the organization. Firewall Manager supports all protected resource types except for Amazon Route 53 and AWS Global Accelerator. To use Firewall Manager, see AWS Firewall Manager and Setting up AWS Firewall Manager AWS Shield Advanced policies.

If you don't use Firewall Manager, for each account with resources to protect, subscribe and add protections using the following procedures.

To subscribe an account to AWS Shield Advanced

1. Sign in to the AWS Management Console and open the AWS WAF & Shield console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.
2. In the **AWS Shield** navigation bar, choose **Getting started**. Choose **Subscribe to Shield Advanced**.
3. In the **Subscribe to Shield Advanced** page, read each term of the agreement, and then select all of the check boxes to indicate that you accept the terms. For accounts in a consolidated billing family, you must agree to the terms for each account.

   > **Important**
   >
   > When you are subscribed, to unsubscribe you must contact AWS Support.
   > To disable autorenewal for your subscription, you must use the Shield API operation UpdateSubscription or the CLI command update-subscription.

   Choose **Subscribe to Shield Advanced**. This subscribes your account to Shield Advanced and activates the service.

Your account is subscribed. Continue through the following steps to protect your account's resources with Shield Advanced.

> **Note**
>
> Shield Advanced doesn't automatically protect your resources after you subscribe. You must specify the resources you want Shield Advanced to protect.

Adding and configuring resource protections with Shield Advanced

This page provides instructions for adding and configuring protections for your resources.

Shield Advanced only protects the resources that you specify, either through Shield Advanced or in a Firewall Manager Shield Advanced policy. It doesn't automatically protect the resources of a subscribed account.

> **Note**
>
> If you use an AWS Firewall Manager Shield Advanced policy for your protections, you don't need to do this step. You configure the policy with the types of resource to protect, and Firewall Manager automatically adds protections to resources that are within scope of the policy.

If you don't use Firewall Manager, go through the following procedures for each account that has resources to protect.

To choose the resources to protect using Shield Advanced

1. Choose **Add resources to protect** from the subscription confirmation page of the prior procedure, or from the **Protected resources** or **Overview** page.
2. In the **Choose resources to protect with Shield Advanced** page, in **Specify the Region and resource types**, provide the Region and resource type specifications for the resources that you want to protect. You can protect resources in multiple Regions by selecting **All Regions** and you can narrow the selection to global resources by selecting **Global**. You can deselect any resource types that you do not want to protect. For information about protections for your resource types, see List of resources that AWS Shield Advanced protects.
3. Choose **Load resources**. Shield Advanced populates the **Select Resources** section with the AWS resources that match your criteria.

4. In the **Select Resources** section, you can filter the list of resources by entering a string to search for in the resource listings.

   Select the resources that you want to protect.

5. In the **Tags** section, if you want to add tags to the Shield Advanced protections that you are creating, specify those. For information about tagging AWS resources, see Working with Tag Editor.

6. Choose **Protect with Shield Advanced**. This adds Shield Advanced protections to the resources.

Continue through the console wizard screens to complete the configuration of your resource protections.

Topics

- Configuring application layer (layer 7) DDoS protections with AWS WAF
- Configuring health-based detection for your protections with Shield Advanced and Route 53
- Configuring alarms and notifications with Shield Advanced and Amazon SNS
- Reviewing and finishing your protection configuration in Shield Advanced

Configuring application layer (layer 7) DDoS protections with AWS WAF

This page provides instructions for configuring application layer protections with AWS WAF web ACLs.

To protect an application layer resource, Shield Advanced uses an AWS WAF web ACL with a rate-based rule as a starting point. AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to your application layer resources, and lets you control access to your content based on the characteristics of the requests. A rate-based rule limits the volume of traffic based on your request aggregation criteria, providing basic DDoS protection to your application. For more information, see How AWS WAF works and Using rate-based rule statements in AWS WAF.

You can also optionally enable Shield Advanced automatic application layer DDoS mitigation, to have Shield Advanced rate limit requests from known DDoS sources and automatically provide incident-specific protections for you.

Important

If you manage your Shield Advanced protections through AWS Firewall Manager using a Shield Advanced policy, you can't manage application layer protections here. You must manage them in your Firewall Manager Shield Advanced policy.

Shield Advanced subscriptions and AWS WAF costs

Your Shield Advanced subscription covers the costs of using standard AWS WAF capabilities for resources that you protect with Shield Advanced. The standard AWS WAF fees that are covered by your Shield Advanced protections are the cost per protection pack (web ACL), the cost per rule, and the base price per million requests for web request inspection, up to 1,500 WCUs and up to the default body size.

Enabling Shield Advanced automatic application layer DDoS mitigation adds a rule group to your protection pack (web ACL) that uses 150 web ACL capacity units (WCUs). These WCUs count against the WCU usage in your protection pack (web ACL). For more information, see Automating application layer DDoS mitigation with Shield Advanced, Protecting the application layer with the Shield Advanced rule group, and Web ACL capacity units (WCUs) in AWS WAF.

Your subscription to Shield Advanced does not cover the use of AWS WAF for resources that you do not protect using Shield Advanced. It also does not cover any additional non-standard AWS WAF costs for protected resources. Examples of non-standard AWS WAF costs are those for Bot Control, for the CAPTCHA rule action, for web ACLs that use more than 1,500 WCUs, and for inspecting the request body beyond the default body size. The full list is provided on the AWS WAF pricing page. Your subscription to Shield Advanced includes access to the Layer 7 Anti-DDoS Amazon Managed Rule group. As part of your subscription, you will get up to 50 billion requests to Shield Advanced protected AWS WAF resources in a calendar month. Requests beyond 50 billion will be billed as per the AWS Shield Advanced pricing page.

For full information and pricing examples, see Shield Pricing and AWS WAF Pricing.

To configure layer 7 DDoS protections for a Region

Shield Advanced gives you the option to configure layer 7 DDoS mitigation for each Region where your chosen resources are located. If you're adding protections in multiple regions, the wizard walks you through the following procedure for each Region.

1. The **Configure layer 7 DDoS protections** page lists each resource that isn't yet associated with a web ACL. For each of these, either choose an existing web ACL or create a new web ACL. For any resource that already has an associated web ACL, you can change web ACLs by first disassociating the current one through AWS WAF. For more information, see Associating or disassociating protection with an AWS resource.

   For web ACLs that don't already have a rate-based rule, the configuration wizard prompts you to add one. A rate-based rule limits traffic from IP addresses when they are sending a high volume of requests. Rate-based rules help protect your application against web request floods and can provide alerts about sudden spikes in traffic that might indicate a potential DDoS attack. Add a rate-based rule to a web ACL by choosing **Add rate limit rule** and then providing a rate limit and rule action. You can configure additional protections in the web ACL through AWS WAF.

   For information about using web ACLs and rate-based rules in your Shield Advanced protections, including additional configuration options for rate-based rules, see Protecting the application layer with AWS WAF web ACLs and Shield Advanced.

2. For **Automatic application layer DDoS mitigation**, if you want to have Shield Advanced automatically mitigate DDoS attacks against your application layer resources, choose **Enable** and then select the AWS WAF rule action that you want Shield Advanced to use in its custom rules. This setting applies to all of the web ACLs for the resources that you are managing in this wizard session.

   With automatic application layer DDoS mitigation, Shield Advanced maintains a rate-based rule in the resource's AWS WAF web ACL that limits the volume of requests from known DDoS sources. Additionally, Shield Advanced compares current traffic patterns against historic traffic baselines to detect deviations that might indicate a DDoS attack. When Shield Advanced detects a DDoS attack, it responds by creating, evaluating, and deploying custom AWS WAF rules to respond. You specify whether the custom rules count or block attacks on your behalf.

   > **Note**
   >
   > Automatic application layer DDoS mitigation works only with protection packs (web ACLs) that were created using the latest version of AWS WAF (v2).

For more information about Shield Advanced automatic application layer DDoS mitigation, including caveats and best practices for using this feature, see Automating application layer DDoS mitigation with Shield Advanced.

3. Choose **Next**. The console wizard advances to the health-based detection page.

Configuring health-based detection for your protections with Shield Advanced and Route 53

This page provides instructions for configuring Shield Advanced to use health-based detection. This can help improve responsiveness and accuracy in attack detection and mitigation.

Well-configured health checks are essential for accurate detection of events. You can configure health-based detection for any resource type except for Route 53 hosted zones.

To use health-based detection, define a health check for your resource in Route 53, and then associate the health check with your Shield Advanced protection. It's important that the health check that you configure accurately reflect the health of the resource. For information and examples for configuring health checks to use with Shield Advanced, see Health-based detection using health checks with Shield Advanced and Route 53.

Health checks are required for Shield Response Team (SRT) proactive engagement support. For information about proactive engagement, see Setting up proactive engagement for the SRT to contact you directly.

> **Note**
>
> Health checks must be reporting healthy when you associate them with your Shield Advanced protections.

To configure health-based detection

1. Under **Associated Health Check**, choose the ID of the health check that you want to associate with the protection.

> **Note**
>
> If you do not see the health check you need, go to the Route 53 console and verify the health check and its ID. For information, see Creating and Updating Health Checks.

2. Choose **Next**. The console wizard advances to the alarms and notifications page.

Configuring alarms and notifications with Shield Advanced and Amazon SNS

This page provides instructions to optionally configure Amazon Simple Notification Service notifications for detected Amazon CloudWatch alarms and rate-based rule activity. You can use these to receive notification when Shield detects an event on a protected resource or when a rate-limit configured in a rate-based rule is exceeded.

For information about Shield Advanced CloudWatch metrics, see AWS Shield Advanced metrics. For information about Amazon SNS, see the Amazon Simple Notification Service Developer Guide.

To configure alarms and notifications

1. Select the Amazon SNS topics that you want notification for. You can use a single Amazon SNS topic for all protected resources and rate-based rules, or you can choose different topics, customized to your organization. For example, you can create an SNS topic for each team that's responsible for incident response for a specific set of resources.
2. Choose **Next**. The console wizard advances to the resource protection review page.

Reviewing and finishing your protection configuration in Shield Advanced

To review and finish your settings

1. In the **Review and configure DDoS mitigation and visibility** page, review your settings. To make modifications, choose **Edit** in the area that you want to modify. This takes you back to the associated page in the console wizard. Make your changes, then choose **Next** in the subsequent pages until you return to the **Review and configure DDoS mitigation and visibility** page.
2. Choose **Finish configuration**. The **Protected resources** page lists your newly protected resources.

Setting up AWS Shield Response Team (SRT) support for DDoS event response

This page provides instructions for setting up Shield Response Team (SRT) support.

The SRT includes security engineers who specialize in DDoS event response. You can optionally add permissions that allow the SRT to manage resources on your behalf during a DDoS event. In addition, you can configure the SRT to proactively engage with you if the Route 53 health checks associated with your protected resources are unhealthy during a detected event. Both of these additions to your protections enable quicker responses to DDoS events.

> **Note**
>
> To use the services of the Shield Response Team (SRT), you must be subscribed to the Business Support plan or the Enterprise Support plan.

The SRT can monitor AWS WAF request data and logs during application layer events to identify anomalous traffic. They can help craft custom AWS WAF rules to mitigate offending traffic sources. As needed, the SRT might make architectural recommendations to help you better align your resources with AWS recommendations.

For more information about the SRT, see Managed DDoS event response with Shield Response Team (SRT) support.

To grant permissions to the SRT

1. In the AWS Shield console **Overview** page, under **Configure AWS SRT support**, choose **Edit SRT access**. The **Edit AWS Shield Response Team (SRT) access** page opens.
2. For **SRT access setting** select one of the options:

   - **Do not grant the SRT access to my account** – Shield removes any permissions you previously gave to the SRT to access your account and resources.
   - **Create a new role for the SRT to access my account** – Shield creates a role that trusts the service principal `drt.shield.amazonaws.com`, which represents the SRT, and attaches the managed policy `AWSShieldDRTAccessPolicy` to it. The managed policy allows the SRT to make AWS Shield Advanced and AWS WAF API calls on your behalf and to access your

AWS WAF logs. For more information about the managed policy, see AWS managed policy: AWSShieldDRTAccessPolicy.

- **Choose an existing role for the SRT to access my accounts** – For this option, you must modify the configuration of the role in AWS Identity and Access Management (IAM) as follows:
  - Attach the managed policy `AWSShieldDRTAccessPolicy` to the role. This managed policy allows the SRT to make AWS Shield Advanced and AWS WAF API calls on your behalf and to access your AWS WAF logs. For more information about the managed policy, see AWS managed policy: AWSShieldDRTAccessPolicy. For information about attaching the managed policy to your role, see Attaching and Detaching IAM Policies.
  - Modify the role to trust the service principal `drt.shield.amazonaws.com`. This is the service principal that represents the SRT. For more information, see IAM JSON Policy Elements: Principal.

3. Choose **Save** to save your changes.

For more information about giving the SRT access to your protections and data, see Granting access for the SRT.

To enable SRT proactive engagement

1. In the AWS Shield console **Overview** page, under **Proactive engagement and contacts**, in the contacts area, choose **Edit**.

   In the **Edit contacts** page, provide the contact information for the people that you want the SRT to contact for proactive engagement.

   If you provide more than one contact, in the **Notes**, indicate the circumstances under which each contact should be used. Include primary and secondary contact designations, and provide the hours of availability and time zones for each contact.

   Example contact notes:

   - This is a hotline that's staffed 24x7x365. Please work with the responding analyst and they will get the appropriate person on the call.
   - Please contact me if the hotline doesn't respond within 5 minutes.

2. Choose **Save**.

The **Overview** page reflects the updated contact information.

3. Choose **Edit proactive engagement feature**, choose **Enable**, and then choose **Save** to enable proactive engagement.

For more information about proactive engagement, see Setting up proactive engagement for the SRT to contact you directly.

Creating a DDoS dashboard in CloudWatch and setting CloudWatch alarms

This page provides instructions for creating a DDoS dashboard in CloudWatch and setting CloudWatch alarms.

You can monitor potential DDoS activity using Amazon CloudWatch, which collects raw data from Shield Advanced and processes it into readable, near real-time metrics. You can use statistics in CloudWatch to gain a perspective on how your web application or service is performing. For more information about using CloudWatch, see What is CloudWatch in the *Amazon CloudWatch User Guide*.

- For instructions for creating a CloudWatch dashboard, see Monitoring with Amazon CloudWatch.
- For descriptions of the Shield Advanced metrics that you can add to your dashboard, see AWS Shield Advanced metrics.

Shield Advanced reports resource metrics to CloudWatch more frequently during DDoS events than while no events are underway. Shield Advanced reports metrics once a minute during an event, and then once right after the event ends. While no events are underway, Shield Advanced reports metrics once a day, at a time assigned to the resource. This periodic report keeps the metrics active and available for use in your custom CloudWatch alarms.

This completes the tutorial for getting started with Shield Advanced. To take full advantage of the protections you've chosen, continue exploring the features and options of Shield Advanced. To start, familiarize yourself with your options for viewing and responding to events at Visibility into DDoS events with Shield Advanced and Responding to DDoS events in AWS.

Managed DDoS event response with Shield Response Team (SRT) support

This page describes the function of the Shield Response Team (SRT).

The SRT provides added support for Shield Advanced customers. The SRT are security engineers who specialize in DDoS event response. As an additional layer of support to your AWS Support plan, you can work directly with the SRT, leveraging their expertise as part of your event response workflow. For information about the options and for configuration guidance, see the topics that follow.

Note

- To use the services of the Shield Response Team (SRT), you must be subscribed to the Business Support plan or the Enterprise Support plan.
- Shield Response Team (SRT) provides services in regions where Shield Advanced is available, and for customers in GovCloud regions, AWS GovCloud (US-East) and AWS GovCloud (US-West).

SRT support activities

The primary goal in an engagement with the SRT is to protect the availability and performance of your application. Depending on the type of DDoS event and the architecture of your application, the SRT may take one or more of the following actions:

- **AWS WAF log analysis and rules** – For resources that use an AWS WAF web ACL, the SRT can analyze your AWS WAF logs to identify attack characteristics in your application web requests. With your approval during engagement, the SRT can apply changes to your web ACL to block the attacks that they've identified.
- **Build custom network mitigations** – The SRT can write custom mitigations for you for infrastructure layer attacks. The SRT can work with you to understand traffic that's expected for your application, to block unexpected traffic, and to optimize packet per second rate limits. For more information, see Setting up custom mitigations against DDoS attacks with the SRT.
- **Network traffic engineering** – The SRT works closely with AWS networking teams to protect Shield Advanced customers. When required, AWS can change how internet traffic arrives on the AWS network in order to allocate more mitigation capacity to your application.

- **Architectural recommendations** – The SRT may determine that the best mitigation for an attack requires architectural changes to better align with the AWS best practices, and they will help support your implementation of these practices. For information, see AWS Best Practices for DDoS Resiliency.

The following sections provide instructions for engaging with the SRT

Topics

- Granting access for the SRT
- Setting up proactive engagement for the SRT to contact you directly
- Contacting the SRT for help with a suspected DDoS event
- Setting up custom mitigations against DDoS attacks with the SRT

Granting access for the SRT

This page provides instructions for granting permission to the SRT to act on your behalf, so that they can access your AWS WAF logs and make calls to the AWS Shield Advanced and AWS WAF APIs to manage protections.

During application layer DDoS events, the SRT can monitor AWS WAF requests to identify anomalous traffic and help craft custom AWS WAF rules to mitigate offending traffic sources.

Additionally, you can grant the SRT access to other data that you have stored in Amazon S3 buckets, such as packet captures or logs from an Application Load Balancer, Amazon CloudFront, or from third party sources.

> **Note**
>
> To use the services of the Shield Response Team (SRT), you must be subscribed to the Business Support plan or the Enterprise Support plan.

To manage permissions for the SRT

1. In the AWS Shield console **Overview** page, under **Configure AWS SRT support**, choose **Edit SRT access**. The **Edit AWS Shield Response Team (SRT) access** page opens.

2. For **SRT access setting** select one of the options:

   - **Do not grant the SRT access to my account** – Shield removes any permissions you previously gave to the SRT to access your account and resources.
   - **Create a new role for the SRT to access my account** – Shield creates a role that trusts the service principal `drt.shield.amazonaws.com`, which represents the SRT, and attaches the managed policy `AWSShieldDRTAccessPolicy` to it. The managed policy allows the SRT to make AWS Shield Advanced and AWS WAF API calls on your behalf and to access your AWS WAF logs. For more information about the managed policy, see AWS managed policy: AWSShieldDRTAccessPolicy.
   - **Choose an existing role for the SRT to access my accounts** – For this option, you must modify the configuration of the role in AWS Identity and Access Management (IAM) as follows:
     - Attach the managed policy `AWSShieldDRTAccessPolicy` to the role. This managed policy allows the SRT to make AWS Shield Advanced and AWS WAF API calls on your behalf and to access your AWS WAF logs. For more information about the managed policy, see AWS managed policy: AWSShieldDRTAccessPolicy. For information about attaching the managed policy to your role, see Attaching and Detaching IAM Policies.
     - Modify the role to trust the service principal `drt.shield.amazonaws.com`. This is the service principal that represents the SRT. For more information, see IAM JSON Policy Elements: Principal.

3. For **(Optional): Grant SRT access to an Amazon S3 bucket**, if you need to share data that isn't in your AWS WAF web ACL logs, configure this. For example, Application Load Balancer access logs, Amazon CloudFront logs, or logs from third party sources.

   > **Note**
   >
   > You don't need to do this for your AWS WAF web ACL logs. The SRT gains access to those when you grant access to your account.

   a. Configure the Amazon S3 buckets according to the following guidelines:

      - The bucket locations must be in the same AWS account as the one you gave the SRT general access to, in the prior step **AWS Shield Response Team (SRT) access**.

- The buckets can be either plaintext or SSE-S3 encrypted. For more information about Amazon S3 SSE-S3 encryption, see Protecting Data Using Server-Side Encryption with Amazon S3-Managed Encryption Keys (SSE-S3) in the Amazon S3 User Guide.

  The SRT cannot view or process logs that are stored in buckets that are encrypted with keys stored in AWS Key Management Service (AWS KMS).

b. In the Shield Advanced **(Optional): Grant SRT access to an Amazon S3 bucket** section, for each Amazon S3 bucket where your data or logs are stored, enter the name of the bucket and choose **Add Bucket**. You can add up to 10 buckets.

   This grants the SRT the following permissions on each bucket: `s3:GetBucketLocation`, `s3:GetObject`, and `s3:ListBucket`.

   If you want to give the SRT permission to access more than 10 buckets, you can do this by editing the additional bucket policies and manually granting the permissions listed here for the SRT.

   The following shows an example policy listing.

```
{
    "Sid": "AWSDDoSResponseTeamAccessS3Bucket",
    "Effect": "Allow",
    "Principal": {
        "Service": "drt.shield.amazonaws.com"
    },
    "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket"
    ],
    "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
    ]
}
```

4. Choose **Save** to save your changes.

You can also authorize the SRT through the API by creating an IAM role, attaching the policy AWSShieldDRTAccessPolicy to it, and then passing the role to the operation AssociateDRTRole.

Setting up proactive engagement for the SRT to contact you directly

> **Note**
>
> Proactive engagement is not available for events detected by the Anti-DDoS managed rule group (AMR). The Shield Advanced console gives you visibility into events for both Shield Advanced and Anti-DDoS AMR detections. Only an event that's detected by Shield Advanced can trigger the proactive engagement. To be notified for Anti-DDoS managed rule group events, set up alarms using events and metrics generated by the Anti-DDoS AMR. Contact support to engage with SRT if needed.

This page provides instructions for setting up proactive engagement with the SRT.

With proactive engagement, the SRT contacts you directly when the availability or performance of your application is affected because of a possible attack. We recommend this engagement model because it provides the quickest SRT response and it allows the SRT to begin troubleshooting even before they've established contact with you.

Proactive engagement is available for network-layer and transport-layer events on Elastic IP addresses and AWS Global Accelerator standard accelerators, and for web request floods on Amazon CloudFront distributions and Application Load Balancers. Proactive engagement is available only for Shield Advanced resource protections that have an associated Amazon Route 53 health check. For information about managing and using health checks, see Health-based detection using health checks with Shield Advanced and Route 53.

During an event that's detected by Shield Advanced, the SRT uses the state of your health checks to determine whether the event qualifies for proactive engagement. If so, the SRT will contact you according to the contact guidance that you provide in your proactive engagement configuration.

You can configure up to ten contacts for proactive engagement, and you can provide notes to guide the SRT in reaching out to you. Your proactive engagement contacts should be available to engage with the SRT during events. If you don't have a 24/7 operations center, you can provide a pager contact and indicate this contact preference in your contact notes.

Proactive engagement requires you to do the following:

- You must be subscribed to the Business Support plan or the Enterprise Support plan.

- You must associate an Amazon Route 53 health check with any resource that you want to protect with proactive engagement. The SRT uses the status of your health checks to help determine whether an event requires proactive engagement, so it's important that your health checks accurately reflect the state of your protected resources. For more information and guidance, see Health-based detection using health checks with Shield Advanced and Route 53.
- For a resource that has an AWS WAF web ACL associated, you must create the web ACL using AWS WAF (v2), which is the latest version of AWS WAF.
- You must provide at least one contact for the SRT to use for proactive engagement during an event. Keep your contact information complete and up to date.

To enable SRT proactive engagement

1. In the AWS Shield console **Overview** page, under **Proactive engagement and contacts**, in the contacts area, choose **Edit**.

   In the **Edit contacts** page, provide the contact information for the people that you want the SRT to contact for proactive engagement.

   If you provide more than one contact, in the **Notes**, indicate the circumstances under which each contact should be used. Include primary and secondary contact designations, and provide the hours of availability and time zones for each contact.

   Example contact notes:

   - This is a hotline that's staffed 24x7x365. Please work with the responding analyst and they will get the appropriate person on the call.
   - Please contact me if the hotline doesn't respond within 5 minutes.
2. Choose **Save**.

   The **Overview** page reflects the updated contact information.
3. Choose **Edit proactive engagement feature**, choose **Enable**, and then choose **Save** to enable proactive engagement.

Contacting the SRT for help with a suspected DDoS event

You can contact the SRT in one of the following ways:

Support case

You can open a case under **AWS Shield** in the **AWS Support Center** console.

For guidance on creating a support case, see AWS Support Center.

Select the severity appropriate to your situation and provide your contact details. In the description, provide as much detail as possible. Provide information about any protected resources that you think might be affected, and the current state of your end-user experience. For example, if your user experience is degraded or parts of your application are currently unavailable, provide that information.

- **For suspected DDoS attacks** – If the availability or performance of your application is currently affected by a possible DDoS attack, choose the following severity and contact options:
  - For severity, choose the highest severity available for your support plan:
    - For Business support this is **Production system down: < 1 hour**.
    - For Enterprise support this is **Business-critical system down: < 15 minutes**.
  - For contact option, select either **Phone** or **Chat** and provide your details. Using a live contact method provides the fastest response.

Proactive engagement

With AWS Shield Advanced proactive engagement, the SRT contacts you directly if the Amazon Route 53 health check associated with your protected resource becomes unhealthy during a detected event. For more information about this option, see Setting up proactive engagement for the SRT to contact you directly.

Setting up custom mitigations against DDoS attacks with the SRT

This page provides instructions for working with the SRT to build custom mitigations against DDoS attacks.

For your Elastic IPs (EIPs) and your AWS Global Accelerator standard accelerators, you can work with the SRT to configure custom mitigations. This is useful in case you know of specific logic that should be enforced when a mitigation is placed. For example, you may wish to only allow traffic from certain countries, enforce specific rate limits, configure optional validations, disallow fragments, or only allow traffic that matches a specific pattern in the packet payload.

Examples of common custom mitigations include the following:

- **Pattern matching** – If you operate a service that interacts with client-side applications, you can choose to match on known patterns that are unique to those applications. For example, you may operate a gaming or communications service that requires the end-user to install specific software that you distribute. You can include a magic number in every packet sent by the application to your service. You can match on up to 128 bytes (separate or contiguous) of a non-fragmented TCP or UDP packet payload and headers. The match can be expressed in hexadecimal notation as a specific offset from the beginning of the packet payload or a dynamic offset following a known value. For example, the mitigation can look for the byte `0x01` and then expect `0x12345678` as the next four bytes.
- **DNS specific** – If you operate your own authoritative DNS service using services like Global Accelerator or Amazon Elastic Compute Cloud (Amazon EC2), you can request a custom mitigation that validates packets to ensure that they are valid DNS queries and apply suspicion scoring that evaluates attributes that are specific to DNS traffic.

To inquire about working with SRT to build custom mitigations, create a support case under AWS Shield. To learn more about creating AWS Support cases, see Getting started with AWS Support.

Resource protections in AWS Shield Advanced

You can add and configure AWS Shield Advanced protections for your resources. You can manage protections for a single resource and you can group your protected resources into logical collections for better event management. You can also track changes to your Shield Advanced protections using AWS Config.

> **Note**
>
> Shield Advanced protects only resources that you have specified either in Shield Advanced or through an AWS Firewall Manager Shield Advanced policy. It doesn't automatically protect your resources.

If you're using an AWS Firewall Manager Shield Advanced policy, you don't need to manage protections for resources that are in scope of the policy. Firewall Manager automatically manages protections for accounts and resources that are in scope of a policy, according to the policy configuration. For more information, see Using AWS Shield Advanced policies in Firewall Manager.

Topics

- List of resources that AWS Shield Advanced protects
- Protecting Amazon EC2 instances and Network Load Balancers with Shield Advanced
- Protecting the application layer (layer 7) with AWS Shield Advanced and AWS WAF
- Health-based detection using health checks with Shield Advanced and Route 53
- Adding AWS Shield Advanced protection to AWS resources
- Editing AWS Shield Advanced protections
- Creating alarms and notifications for resources protected by Shield Advanced
- Removing AWS Shield Advanced protection from an AWS resource
- Grouping your AWS Shield Advanced protections
- Tracking Shield Advanced resource protection changes in AWS Config

List of resources that AWS Shield Advanced protects

This section provides information about Shield Advanced protections for each resource type.

Shield Advanced protects AWS resources in the network and transport layers (layers 3 and 4) and in the application layer (layer 7). You can protect some resources directly and others through association with protected resources. Shield Advanced supports IPv4, and does not support IPv6.

> **Note**
>
> Shield Advanced protects only resources that you have specified either in Shield Advanced or through an AWS Firewall Manager Shield Advanced policy. It doesn't automatically protect your resources.

You can use Shield Advanced for advanced monitoring and protection with the following resource types:

- Amazon CloudFront distributions. For CloudFront continuous deployment, Shield Advanced protects any staging distribution that's associated with a protected primary distribution.
- Amazon Route 53 hosted zones.
- AWS Global Accelerator standard accelerators.
- Amazon EC2 Elastic IP addresses. Shield Advanced protects the resources that are associated with protected Elastic IP addresses.

- Amazon EC2 instances, through association to Amazon EC2 Elastic IP addresses.
- The following Elastic Load Balancing (ELB) load balancers:
  - Application Load Balancers.
  - Classic Load Balancers.
  - Network Load Balancers, through associations to Amazon EC2 Elastic IP addresses.

> **Note**
>
> You can't use Shield Advanced to protect any other resource type. For example, you can't protect AWS Global Accelerator custom routing accelerators or Gateway Load Balancers.

> **Note**
>
> NAT Gateways handle outbound traffic only, whereas Shield Advanced protects against inbound DDoS. For outbound traffic protection, use AWS Network Firewall.

You can monitor and protect up to 1,000 resources for each resource type per AWS account. For example, in a single account, you could protect 1,000 Amazon EC2 Elastic IP addresses, 1,000 CloudFront distributions, and 1,000 Application Load Balancers. You can request an increase to the number of resources that you can protect with Shield Advanced through the Service Quotas console at https://eusc-de-east-1.console.amazonaws-eusc.eu/servicequotas/.

Protecting Amazon EC2 instances and Network Load Balancers with Shield Advanced

This page explains how to use AWS Shield Advanced protections for Amazon EC2 instances and Network Load Balancers.

You can protect Amazon EC2 instances and Network Load Balancers by first attaching these resources to Elastic IP addresses, and then protecting the Elastic IP addresses in Shield Advanced.

When you protect Elastic IP addresses, Shield Advanced identifies and protects the resources that they're attached to. Shield Advanced automatically identifies the type of resource that's attached to an Elastic IP address and applies the appropriate detections and mitigations for that

resource. This includes configuring network ACLs that are specific to the Elastic IP address. For more information about using Elastic IP addresses with your AWS resources, see the following guides: Amazon Elastic Compute Cloud documentation or Elastic Load Balancing documentation.

During an attack, Shield Advanced automatically deploys your network ACLs to the border of the AWS network. When your network ACLs are at the border of the network, Shield Advanced can provide protection against larger DDoS events. Typically, network ACLs are applied near your Amazon EC2 instances within your Amazon VPC. The network ACL can mitigate attacks only as large as your Amazon VPC and instance can handle. For example, if the network interface attached to your Amazon EC2 instance can process up to 10 Gbps, then volumes over 10 Gbps will slow down and possibly block traffic to that instance. During an attack, Shield Advanced promotes your network ACL to the AWS border, which can process multiple terabytes of traffic. Your network ACL is able to provide protection for your resource well beyond your network's typical capacity. For more information about network ACLs, see Network ACLs.

Some scaling tools, like AWS Elastic Beanstalk, don't let you automatically attach an Elastic IP address to a Network Load Balancer. For those cases, you need to manually attach the Elastic IP address.

Protecting the application layer (layer 7) with AWS Shield Advanced and AWS WAF

This page explains how Shield Advanced and AWS WAF work together to protect resources at the application layer (layer 7).

To protect your application layer resources with Shield Advanced, you start by associating an AWS WAF web ACL with the resource and adding one or more rate-based rules to it. You can additionally enable automatic application layer DDoS mitigation, which causes Shield Advanced to automatically create and manage web ACL rules on your behalf in response to DDoS attacks.

When you protect an application layer resource with Shield Advanced, Shield Advanced analyzes traffic over time to establish and maintain baselines. Shield Advanced uses these baselines to detect anomalies in traffic patterns that might indicate a DDoS attack. The point at which Shield Advanced detects an attack depends on the traffic that Shield Advanced has been able to observe prior to the attack and on the architecture you use for your web applications. The architectural variations that can affect Shield Advanced behavior include the type of instance you use, your instance size, and whether the instance type supports enhanced networking. You can also configure Shield Advanced to automatically place mitigations for application layer attacks.

Shield Advanced subscriptions and AWS WAF costs

Your Shield Advanced subscription covers the costs of using standard AWS WAF capabilities for resources that you protect with Shield Advanced. The standard AWS WAF fees that are covered by your Shield Advanced protections are the cost per protection pack (web ACL), the cost per rule, and the base price per million requests for web request inspection, up to 1,500 WCUs and up to the default body size.

Enabling Shield Advanced automatic application layer DDoS mitigation adds a rule group to your protection pack (web ACL) that uses 150 web ACL capacity units (WCUs). These WCUs count against the WCU usage in your protection pack (web ACL). For more information, see Automating application layer DDoS mitigation with Shield Advanced, Protecting the application layer with the Shield Advanced rule group, and Web ACL capacity units (WCUs) in AWS WAF.

Your subscription to Shield Advanced does not cover the use of AWS WAF for resources that you do not protect using Shield Advanced. It also does not cover any additional non-standard AWS WAF costs for protected resources. Examples of non-standard AWS WAF costs are those for Bot Control, for the CAPTCHA rule action, for web ACLs that use more than 1,500 WCUs, and for inspecting the request body beyond the default body size. The full list is provided on the AWS WAF pricing page. Your subscription to Shield Advanced includes access to the Layer 7 Anti-DDoS Amazon Managed Rule group. As part of your subscription, you will get up to 50 billion requests to Shield Advanced protected AWS WAF resources in a calendar month. Requests beyond 50 billion will be billed as per the AWS Shield Advanced pricing page.

For full information and pricing examples, see Shield Pricing and AWS WAF Pricing.

Topics

- List of factors that affect application layer event detection and mititgation with Shield Advanced
- Protecting the application layer with AWS WAF web ACLs and Shield Advanced
- Protecting the application layer with AWS WAF rate-based rules and Shield Advanced
- Automating application layer DDoS mitigation with Shield Advanced

List of factors that affect application layer event detection and mititgation with Shield Advanced

This section describes the factors that affect the detection and mitigation of application layer events by Shield Advanced.

Health checks

Health checks that accurately report the overall health of your application provide Shield Advanced with information about the traffic conditions that your application is experiencing. Shield Advanced requires less information pointing to a potential attack when your application is reporting unhealthy and it requires more evidence of an attack if your application is reporting healthy.

It's important to configure your health checks so that they accurately report application health. For more information and guidance, see Health-based detection using health checks with Shield Advanced and Route 53.

Traffic baselines

Traffic baselines give Shield Advanced information about the characteristics of normal traffic for your application. Shield Advanced uses these baselines to recognize when your application isn't receiving normal traffic, so it can notify you and, as configured, start devising and testing mitigation options to counter a potential attack. For additional information about how Shield Advanced uses traffic baselines to detect potential events, see the overview section Shield Advanced detection logic for application layer threats (layer 7).

Shield Advanced creates its baselines from information provided by the web ACL that's associated with the protected resource. The web ACL must be associated with the resource for at least 24 hours and up to 30 days before Shield Advanced can reliably determine the application's baselines. The time required begins when you associate the web ACL, either through Shield Advanced or through AWS WAF.

For more information about using a web ACL with your Shield Advanced application layer protections, see Protecting the application layer with AWS WAF web ACLs and Shield Advanced.

Rate-based rules

Rate-based rules can help mitigate attacks. They can also obscure attacks, by mitigating them before they become a large enough problem to show up against normal traffic baselines or in health check status reporting.

We recommend using rate-based rules in your web ACL when you protect an application resource with Shield Advanced. Even though their mitigations can obscure a potential attack, they are a valuable first line of defense, helping ensure that your application stays available to your legitimate customers. The traffic that your rate-based rules detect and rate limit is visible in your AWS WAF metrics.

In addition to your own rate-based rules, if you enable automatic application layer DDoS mitigation, Shield Advanced adds a rule group to your web ACL that it uses to mitigate attacks. In this rule group, Shield Advanced always has a rate-based rule in place that limits the volume of requests from IP addresses that are known to be sources of DDoS attacks. Metrics for the traffic that the Shield Advanced rules mitigate aren't available for you to view.

For more information about rate-based rules, see Using rate-based rule statements in AWS WAF. For information about the rate-based rule that Shield Advanced uses for automatic application layer DDoS mitigation, see Protecting the application layer with the Shield Advanced rule group.

For more information about Shield Advanced and AWS WAF metrics, see Monitoring with Amazon CloudWatch.

Protecting the application layer with AWS WAF web ACLs and Shield Advanced

This page explains how AWS WAF web ACLs and Shield Advanced work together to create basic application layer protections.

To protect an application layer resource with Shield Advanced, you start by associating an AWS WAF web ACL with the resource. AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to your application layer resources, and lets you control access to your content based on the characteristics of the requests. You can configure a web ACL to monitor and manage requests based on factors such as where the request originated, the contents of query strings and cookies, and the rate of requests coming from a single IP address. At a minimum, your Shield Advanced protection requires you to associate a web ACL with a rate-based rule, which limits the rate of requests for each IP address.

If the associated web ACL doesn't have a rate-based rule defined, Shield Advanced prompts you to define at least one. Rate-based rules automatically block traffic from source IPs when they exceed the thresholds that you define. They help protect your application against web request floods and can provide alerts about sudden spikes in traffic that might indicate a potential DDoS attack.

> **Note**
>
> A rate-based rule responds very quickly to spikes in the traffic that the rule is monitoring. Because of this, a rate-based rule can prevent not only an attack, but also the detection of a potential attack by Shield Advanced detection. This trade off favors prevention over complete visibility into attack patterns. We recommend using a rate-based rule as your first line of defense against attacks.

With your web ACL in place, if a DDoS attack occurs, you apply mitigations by adding and managing rules in the web ACL. You can do this directly, with the assistance of the Shield Response Team (SRT), or automatically through automatic application layer DDoS mitigation.

> **Important**
>
> If you also use automatic application layer DDoS mitigation, see the best practices for managing your web ACL at Best practices for using automatic application layer DDoS mitigation.

For information about using AWS WAF to manage your web request monitoring and management rules, see Creating a protection pack (web ACL) in AWS WAF.

Protecting the application layer with AWS WAF rate-based rules and Shield Advanced

This page explains how AWS WAF rate-based rules and Shield Advanced work together to create basic application layer protections.

When you use a rate-based rule with its default configuration, AWS WAF periodically evaluates traffic for the prior 5-minute time window. AWS WAF blocks requests from any IP address that exceeds the rule's threshold until the request rate drops down to an acceptable level. When you configure a rate-based rule through Shield Advanced, configure its rate threshold to a value that's greater than the normal traffic rate that you expect from any one source IP in any five minute time window.

You might want to use more than one rate-based rule in a web ACL. For example, you could have one rate-based rule for all traffic that has a high threshold plus one or more additional rules that are configured to match select parts of your web application and that have lower thresholds. For example, you might match on the URI `/login.html` with a lower threshold, to mitigate abuse against a login page.

You can configure a rate-based rule to use a different evaluation time window and to aggregate requests by a number of request components, like header values, labels, and query arguments. For more information, see Using rate-based rule statements in AWS WAF.

For additional information and guidance, see the security blog post The three most important AWS WAF rate-based rules.

Expanded configuration options through AWS WAF

The Shield Advanced console enables you to add a rate-based rule and configure it with the basic, default settings. You can define additional configuration options by managing your rate-based rules through AWS WAF. For example, you can configure the rule to aggregate requests based on keys such as a forwarded IP address, a query string, and a label. You can also add a scope-down statement to the rule to filter out some requests from evaluation and rate limiting. For more information, see Using rate-based rule statements in AWS WAF.

Automating application layer DDoS mitigation with Shield Advanced

> **Note**
>
> Starting March 26, 2026, the Anti-DDoS Managed Rule Group (Anti-DDOS AMR) for AWS WAF becomes the default solution for protection against HTTP request flood attacks (see the Anti-DDoS AMR launch blog). It supersedes the Layer 7 Auto Mitigation (L7AM) feature. If you're an existing Shield Advanced customer, you can continue to use the legacy solution with existing or new AWS accounts. However, we encourage you to adopt the Anti-DDoS Managed Rule Group. The Anti-DDoS Managed Rule Group detects and mitigates attacks within seconds rather than minutes. If you're a new Shield Advanced customer and require access to the legacy solution, contact AWS Support.

This page introduces the topic of automatic application layer DDoS mitigation and lists associated caveats.

You can configure Shield Advanced to respond automatically to mitigate application layer (layer 7) attacks against your protected application layer resources, by counting or blocking web requests that are part of the attack. This option is an addition to the application layer protection that you add through Shield Advanced with an AWS WAF web ACL and your own rate-based rule.

When automatic mitigation is enabled for a resource, Shield Advanced maintains a rule group in the resource's associated web ACL where it manages mitigation rules on behalf of the resource. The rule group contains a rate-based rule that tracks the volume of requests from IP addresses that are known to be sources of DDoS attacks.

Additionally, Shield Advanced compares current traffic patterns against historic traffic baselines to detect deviations that might indicate a DDoS attack. Shield Advanced responds to detected DDoS attacks by creating, evaluating, and deploying additional, custom AWS WAF rules in the rule group.

Caveats for using automatic application layer DDoS mitigation

The following list describes the caveats of Shield Advanced automatic application layer DDoS mitigation, and describes steps that you might want to take in response.

- Automatic application layer DDoS mitigation works only with protection packs (web ACLs) that were created using the latest version of AWS WAF (v2).
- Shield Advanced requires time to establish a baseline of your application's normal, historic traffic, which it leverages to detect and isolate attack traffic from normal traffic, to mitigate attack traffic. The time to establish a baseline is between 24 hours and 30 days from the time you associate a web ACL with the protected application resource. For additional information about traffic baselines, see List of factors that affect application layer event detection and mititgation with Shield Advanced.
- Enabling automatic application layer DDoS mitigation adds a rule group to your protection pack (web ACL) that uses 150 web ACL capacity units (WCUs). These WCUs count against the WCU usage in your protection pack (web ACL). For more information, see Protecting the application layer with the Shield Advanced rule group, and Web ACL capacity units (WCUs) in AWS WAF.
- The Shield Advanced rule group generates AWS WAF metrics, but they are not available to view. This is the same as for any other rule groups that you use in your protection pack (web ACL) but do not own, such as AWS Managed Rules rule groups. For more information about AWS WAF metrics, see AWS WAF metrics and dimensions. For information about this Shield Advanced protection option, see Automating application layer DDoS mitigation with Shield Advanced.
- For web ACLs that protect multiple resources, automatic mitigation only deploys custom mitigations that don't negatively impact any of the protected resources.
- The time between the start of a DDoS attack and when Shield Advanced places custom automatic mitigation rules varies with each event. Some DDoS attacks might end before the custom rules are deployed. Other attacks might happen when a mitigation is already in place, and so might be mitigated by those rules from the start of the event. Additionally, rate-based rules in the web ACL and Shield Advanced rule group might mitigate attack traffic before it's detected as a possible event.
- For Application Load Balancers that receive any traffic through a content delivery network (CDN), such as Amazon CloudFront, the application-layer automatic mitigation capabilities of Shield Advanced for those Application Load Balancer resources will be reduced. Shield Advanced uses client traffic attributes to identify and isolate attack traffic from normal traffic to your application, and CDNs may not preserve or forward the original client traffic attributes. If you use CloudFront, we recommend enabling automatic mitigation on the CloudFront distribution.

- Automatic application layer DDoS mitigation does not interact with protection groups. You can enable automatic mitigation for resources that are in protection groups, but Shield Advanced does not automatically apply attack mitigations based on protection group findings. Shield Advanced applies automatic attack mitigations for individual resources.

Contents

- Best practices for using automatic application layer DDoS mitigation
- Enabling automatic application layer DDoS mitigation
  - What happens when you enable automatic mitigation
- How Shield Advanced manages automatic mitigation
  - How Shield Advanced responds to DDoS attacks with automatic mitigation
  - How Shield Advanced manages the rule action setting
  - How Shield Advanced manages mitigations when an attack subsides
  - What happens when you disable automatic mitigation
- Protecting the application layer with the Shield Advanced rule group
- Viewing the automatic application layer DDoS mitigation configuration for a resource
- Enabling and disabling automatic application layer DDoS mitigation
- Changing the action used for automatic application layer DDoS mitigation
- Using AWS CloudFormation with automatic application layer DDoS mitigation

Best practices for using automatic application layer DDoS mitigation

Adhere to the guidance provided in this section when you use automatic mitigation.

General protections management

Follow these guidelines for planning and implementing your automatic mitigation protections.

- Manage all of your automatic mitigation protections either through Shield Advanced or, if you're using AWS Firewall Manager to manage your Shield Advanced automatic mitigation settings, through Firewall Manager. Don't mix your use of Shield Advanced and Firewall Manager to manage these protections.
- Manage similar resources using the same web ACLs and protection settings, and manage dissimilar resources using different web ACLs. When Shield Advanced mitigates a DDoS attack on a protected resource, it defines rules for the web ACL that's associated with the resource

and then tests the rules against traffic of all resources that are associated with the web ACL. Shield Advanced will only apply the rules if they don't negatively impact any of the associated resources. For more information, see How Shield Advanced manages automatic mitigation.

- For Application Load Balancers that have all their internet traffic proxied through a Amazon CloudFront distribution, only enable automatic mitigation on the CloudFront distribution. The CloudFront distribution will always have the greatest number of original traffic attributes, which Shield Advanced leverages to mitigate attacks.

Detection and mitigation optimization

Follow these guidelines to optimize the protections that automatic mitigation provides to protected resources. For an overview of application layer detection and mitigation, see List of factors that affect application layer event detection and mititgation with Shield Advanced.

- Configure health checks for your protected resources and use them to enable health-based detection in your Shield Advanced protections. For guidance, see Health-based detection using health checks with Shield Advanced and Route 53.
- Enable automatic mitigation in Count mode until Shield Advanced has established a baseline for normal, historic traffic. Shield Advanced needs from 24 hours to 30 days to establish a baseline.

  Establishing a baseline of normal traffic patterns requires the following:

  - The association of a web ACL with the protected resource. You can use AWS WAF directly to associate your web ACL or you can have Shield Advanced associate it when you enable the Shield Advanced application layer protection and specify a web ACL to use.
  - Normal traffic flow to your protected application. If your application isn't experiencing normal traffic, such as before the application is launched or if it lacks production traffic for extended periods of time, the historical data can't be gathered.

Web ACL management

Follow these guidelines for managing the web ACLs that you use with automatic mitigation.

- If you need to replace the web ACL that's associated with the protected resource, make the following changes in order:

  1. In Shield Advanced, disable automatic mitigation.
  2. In AWS WAF, disassociate the old web ACL and associate the new web ACL.

3. In Shield Advanced, enable automatic mitigation.

   Shield Advanced doesn't automatically transfer automatic mitigation from the old web ACL to the new one.

- Don't delete any rule group rule from your web ACLs whose name starts with `ShieldMitigationRuleGroup`. If you do delete this rule group, you disable the protections provided by Shield Advanced automatic mitigation for every resource that's associated with the web ACL. Additionally, it can take Shield Advanced some time to receive notice of the change and to update its settings. During this time, the Shield Advanced console pages will provide incorrect information.

  For more information about the rule group, see Protecting the application layer with the Shield Advanced rule group.
- Don't modify the name of a rule group rule whose name starts with `ShieldMitigationRuleGroup`. Doing so can interfere with the protections provided by Shield Advanced automatic mitigation through the web ACL.
- When you create rules and rule groups, don't use names that start with `ShieldMitigationRuleGroup`. This string is used by Shield Advanced to manage your automatic mitigations.
- In your management of your web ACL rules, don't assign a priority setting of 10,000,000. Shield Advanced assigns this priority setting to its automatic mitigation rule group rule when it adds it.
- Keep the `ShieldMitigationRuleGroup` rule prioritized so that it runs when you want it to in relation to the other rules in your web ACL. Shield Advanced adds the rule group rule to the web ACL with priority 10,000,000, to run after your other rules. If you use the AWS WAF console wizard to manage your web ACL, adjust the priority settings as needed after you add rules to the web ACL.
- If you use AWS CloudFormation to manage your web ACLs, you don't need to manage the `ShieldMitigationRuleGroup` rule group rule. Follow the guidance at Using AWS CloudFormation with automatic application layer DDoS mitigation.

Enabling automatic application layer DDoS mitigation

This page explains how to configure Shield Advanced to automatically respond to application layer attacks.

You enable Shield Advanced automatic mitigation as part of the application layer DDoS protections for your resource. For information about doing this through the console, see Configure application layer DDoS protections.

The automatic mitigation functionality requires you to do the following:

- **Associate a web ACL with the resource** – This is required for any Shield Advanced application layer protection. You can use the same web ACL for multiple resources. We recommend doing this only for resources that have similar traffic. For information about web ACLs, including the requirements for using them with multiple resources, see How AWS WAF works.
- **Enable and configure Shield Advanced automatic application layer DDoS mitigation** – When you enable this, you specify whether you want Shield Advanced to automatically block or count web requests that it determines to be part of a DDoS attack. Shield Advanced adds a rule group to the associated web ACL and uses it to dynamically manage its response to DDoS attacks on the resource. For information about the rule action options, see Using rule actions in AWS WAF.
- **(Optional, but recommended) Add a rate-based rule to the web ACL** – By default, the rate-based rule provides your resource with basic protection against DDoS attacks by preventing any individual IP address from sending too many requests in a short time. For information about rate-based rules, including custom request aggregation options and examples, see Using rate-based rule statements in AWS WAF.

What happens when you enable automatic mitigation

Shield Advanced does the following when you enable automatic mitigation:

- **As needed, adds a rule group for Shield Advanced use** – If the AWS WAF web ACL that you have associated with the resource doesn't already have an AWS WAF rule group rule that's dedicated to automatic application layer DDoS mitigation, Shield Advanced adds one.

  The name of the rule group rule starts with `ShieldMitigationRuleGroup`. The rule group always contains a rate-based rule named `ShieldKnownOffenderIPRateBasedRule`, which limits the volume of requests from IP addresses that are known to be sources of DDoS attacks. For additional details about the Shield Advanced rule group and the web ACL rule that references it, see Protecting the application layer with the Shield Advanced rule group.
- **Starts responding to DDoS attacks against the resource** – Shield Advanced automatically responds to DDoS attacks for the protected resource. In addition to the rate-based rule, which is always present, Shield Advanced uses its rule group to deploy custom AWS WAF rules for DDoS

attack mitigation. Shield Advanced tailors these rules to your application and to the attacks that your application experiences, and tests them against the resource's historical traffic before deploying them.

Shield Advanced uses a single rule group rule in any web ACL that you use for automatic mitigation. If Shield Advanced has already added the rule group for another protected resource, it doesn't add another rule group to the web ACL.

Automatic application layer DDoS mitigation depends on the presence of the rule group to mitigate attacks. If the rule group is removed from the AWS WAF web ACL for any reason, the removal disables automatic mitigation for all resources that are associated with the web ACL.

How Shield Advanced manages automatic mitigation

The topics in this section describe how Shield Advanced handles your configuration changes for automatic application layer DDoS mitigation and how it handles DDoS attacks when automatic mitigation is enabled.

Topics

- How Shield Advanced responds to DDoS attacks with automatic mitigation
- How Shield Advanced manages the rule action setting
- How Shield Advanced manages mitigations when an attack subsides
- What happens when you disable automatic mitigation

How Shield Advanced responds to DDoS attacks with automatic mitigation

When you have automatic mitigation enabled on a protected resource, the rate-based rule `ShieldKnownOffenderIPRateBasedRule` in the Shield Advanced rule group responds automatically to elevated traffic volumes from known DDoS sources. This rate-limiting is applied quickly and acts as a front-line defense against attacks.

When Shield Advanced detects an attack, it does the following:

1. Attempts to identify an attack signature that isolates the attack traffic from the normal traffic to your application. The goal is to produce high quality DDoS mitigation rules that, when placed, affect only the attack traffic and don't impact normal traffic to your application.

2. Evaluates the identified attack signature against the historical traffic patterns for the resource that's under attack as well as for any other resource that's associated with the same web ACL. Shield Advanced does this before it deploys any rules in response to the event.

   Depending on the evaluation results, Shield Advanced does one of the following:

   - If Shield Advanced determines that the attack signature isolates only the traffic that is involved in the DDoS attack, it implements the signature in AWS WAF rules in the Shield Advanced mitigation rule group in the web ACL. Shield Advanced gives these rules the action setting that you've configured for the resource's automatic mitigation - either Count or Block.
   - Otherwise, Shield Advanced doesn't place a mitigation.

Throughout an attack, Shield Advanced sends the same notifications and provides the same event information as for basic Shield Advanced application layer protections. You can see the information about events and DDoS attacks, and about any Shield Advanced mitigations for attacks, in the Shield Advanced event console. For information, see Visibility into DDoS events with Shield Advanced.

If you've configured automatic mitigation to use the Block rule action and you experience false positives from the mitigation rules that Shield Advanced has deployed, you can change the rule action to Count. For information about how to this, see Changing the action used for automatic application layer DDoS mitigation.

How Shield Advanced manages the rule action setting

You can set the rule action for your automatic mitigations to Block or Count.

When you change the automatic mitigation rule action setting for a protected resource, Shield Advanced updates all rule settings for the resource. It updates any rules that are currently in place for the resource in the Shield Advanced rule group and it uses the new action setting when it creates new rules.

For resources that use the same web ACL, if you specify different actions, Shield Advanced uses the Block action setting for the rule group's rate-based rule `ShieldKnownOffenderIPRateBasedRule`. Shield Advanced creates and manages other rules in the rule group on behalf of a specific protected resource, and uses the action setting that you've specified for the resource. All rules in the Shield Advanced rule group in a web ACL are applied to the web traffic of all of the associated resources.

Changing the action setting can take a few seconds to propagate. During this time, you might see the old setting in some places where the rule group is in use, and the new setting in other places.

You can change the rule action setting for your automatic mitigation configuration in the events page of the console, and through the application layer configuration page. For information about the events page, see Responding to DDoS events in AWS. For information about the configuration page, see Configure application layer DDoS protections.

How Shield Advanced manages mitigations when an attack subsides

When Shield Advanced determines that mitigation rules that were deployed for a particular attack are no longer needed, it removes them from the Shield Advanced mitigation rule group.

The removal of mitigating rules won't necessarily coincide with the end of an attack. Shield Advanced monitors patterns of attack that it detects on your protected resources. It might proactively defend against the recurrence of an attack with a specific signature by keeping the rules that it has deployed against the initial occurrence of that attack in place. As needed, Shield Advanced increases the window of time that it keeps rules in place. This way, Shield Advanced might mitigate repeated attacks with a specific signature before they impact your protected resources.

Shield Advanced never removes the rate-based rule `ShieldKnownOffenderIPRateBasedRule`, which limits the volume of requests from IP addresses that are known to be sources of DDoS attacks.

What happens when you disable automatic mitigation

Shield Advanced does the following when you disable automatic mitigation for a resource:

- **Stops automatically responding to DDoS attacks** – Shield Advanced discontinues its automatic response activities for the resource.
- **Removes unneeded rules from the Shield Advanced rule group** – If Shield Advanced is maintaining any rules in its managed rule group on behalf of the protected resource, it removes them.
- **Removes the Shield Advanced rule group, if it's no longer in use** – If the web ACL that you have associated with the resource isn't associated to any other resource that has automatic mitigation enabled, Shield Advanced removes its rule group rule from the web ACL.

Protecting the application layer with the Shield Advanced rule group

This page explains how the Shield Advanced rule group works in your web ACL.

Shield Advanced manages automatic mitigation activities using rules in a rule group that it owns and manages for you. Shield Advanced references the rule group with a rule in the web ACL that you have associated with your protected resource.

The rule group rule in your web ACL

The Shield Advanced rule group rule in your web ACL has the following properties:

- **Name** – `ShieldMitigationRuleGroup_`*`account-id_web-acl-id_unique-identifier`*
- **Web ACL capacity units (WCU)** – 150. These WCUs count against the WCU usage in your web ACL.

Shield Advanced creates this rule in your web ACL with a priority setting of 10,000,000, so that it runs after your other rules and rule groups in the web ACL. AWS WAF runs the rules in a web ACL from the lowest numeric priority setting on up. During your management of the web ACL, this priority setting might change.

The automatic mitigation functionality doesn't consume any additional AWS WAF resources in your account, other than the WCUs used by the rule group in your web ACL. For example, the Shield Advanced rule group isn't counted as one of your account's rule groups. For information about account limits in AWS WAF, see AWS WAF quotas.

Rules in the rule group

Within the referenced Shield Advanced rule group, Shield Advanced maintains a rate-based rule `ShieldKnownOffenderIPRateBasedRule`, which limits the volume of requests from IP addresses that are known to be sources of DDoS attacks. This rule serves as the first line of defense against any attack, because it's always present in the rule group and it doesn't rely on the analysis of traffic patterns to contain attacks. This rule's action is set to the action that you choose for your automatic mitigations, just like the other rules in the rule group. For information about rate-based rules, see Using rate-based rule statements in AWS WAF.

> **Note**
>
> The rate-based rule `ShieldKnownOffenderIPRateBasedRule` operates independent of Shield Advanced event detection. While automatic mitigation is enabled, this rule rate

limits IP addresses that are known to be sources of DDoS attacks. For these IP addresses, the rule's rate limiting can prevent attacks and also keep attacks from appearing in the Shield Advanced detection information. This trade off favors prevention over complete visibility into attack patterns.

In addition to the permanent rate-based rule described above, the rule group contains any rules that Shield Advanced is currently using to mitigate DDoS attacks. Shield Advanced adds, modifies, and removes these rules as needed. For information, see How Shield Advanced manages automatic mitigation.

Metrics

The rule group generates AWS WAF metrics, but because this rule group is owned by Shield Advanced, these metrics aren't available to view. For more information, see AWS WAF metrics and dimensions.

Viewing the automatic application layer DDoS mitigation configuration for a resource

You can view the automatic application layer DDoS mitigation configuration for a resource in the **Protected resources** page and in the individual protections pages.

To view the automatic application layer DDoS mitigation configuration

1. Sign in to the AWS Management Console and open the AWS WAF & Shield console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.
2. In the AWS Shield navigation pane, choose **Protected resources**. In the list of protected resources, the column **Automatic application layer DDoS mitigation** indicates whether automatic mitigation is enabled and, where enabled, the action that Shield Advanced is to use in its mitigations.

   You can also select any application layer resource to see the same information listed on the protections page for the resource.

Enabling and disabling automatic application layer DDoS mitigation

The following procedure shows how to enable or disable automatic response for a protected resource.

To enable or disable automatic application layer DDoS mitigation for a single resource

1. Sign in to the AWS Management Console and open the AWS WAF & Shield console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.
2. In the AWS Shield navigation pane, choose **Protected resources**.
3. In the **Protections** tab, select the application layer resource that you want to enable automatic mitigation for. The protections page opens for the resource.
4. In the resource's protections page, choose **Edit**.
5. In the page **Configure layer 7 DDoS mitigation for global resources - *optional***, for **Automatic application layer DDoS mitigation**, choose the option that you want to use for automatic mitigations. The options in the console are the following:

   - **Keep current settings** – Make no changes to the automatic mitigation settings of the protected resource.
   - **Enable** – Enable automatic mitigation for the protected resource. When you choose this, also select the rule action that you want the automatic mitigations to use in the web ACL rules. For information about rule action settings, see Using rule actions in AWS WAF.

     If your protected resource doesn't yet have a history of normal application traffic, enable automatic mitigation in Count mode until Shield Advanced can establish a baseline. Shield Advanced begins to collect information for its baseline when you associate a web ACL with your protected resource, and it can take 24 hour to 30 days to establish a good baseline of normal traffic.
   - **Disable** – Disable automatic mitigation for the protected resource.
6. Walk through the rest of the pages until you finish and save the configuration.

In the **Protections** page, the automatic mitigation settings are updated for the resource.

Changing the action used for automatic application layer DDoS mitigation

You can change the action that Shield Advanced uses for its application layer automatic response in multiple locations in the console:

- **Automatic mitigation configuration** – Change the action when you configure automatic mitigation for your resource. For the procedure, see the preceding section Enabling and disabling automatic application layer DDoS mitigation.

- **Event details page** – Change the action in the event details page, when you're viewing the event information in the console. For information, see Viewing AWS Shield Advanced event details.

If you have two protected resources that share a web ACL, and you set the action to Count for one and Block for the other, Shield Advanced sets the action for the rule group's rate-based rule `ShieldKnownOffenderIPRateBasedRule` to Block.

Using AWS CloudFormation with automatic application layer DDoS mitigation

This page explains how to use CloudFormation to manage your protections and AWS WAF web ACLs.

Enabling or disabling automatic application layer DDoS mitigation

You can enable and disable automatic application layer DDoS mitigation through AWS CloudFormation, using the `AWS::Shield::Protection` resource. The effect is the same as when you enable or disable the feature through the console or any other interface. For information about the CloudFormation resource, see AWS::Shield::Protection in the *AWS CloudFormation user guide*.

Managing web ACLs used with automatic mitigation

Shield Advanced manages automatic mitigation for your protected resource using a rule group rule in the protected resource's AWS WAF web ACL. Through the AWS WAF console and APIs, you'll see the rule listed in your web ACL rules, with a name that starts with `ShieldMitigationRuleGroup`. This rule is dedicated to your automatic application layer DDoS mitigation and it's managed for you by Shield Advanced and AWS WAF. For more information, see Protecting the application layer with the Shield Advanced rule group and How Shield Advanced manages automatic mitigation.

If you use CloudFormation to manage your web ACLs, don't add the Shield Advanced rule group rule to your web ACL template. When you update a web ACL that's being used with your automatic mitigation protections, AWS WAF automatically manages the rule group rule in the web ACL.

You'll see the following differences compared to other web ACLs that you manage through CloudFormation:

- CloudFormation won't report any drift in the stack drift status between the actual configuration of the web ACL, with the Shield Advanced rule group rule, and your web ACL template, without the rule. The Shield Advanced rule won't appear in the actual listing for the resource in the drift details.

You will be able to see the Shield Advanced rule group rule in web ACL listings that you retrieve from AWS WAF, such as through the AWS WAF console or AWS WAF APIs.

- If you modify the web ACL template in a stack, AWS WAF and Shield Advanced automatically maintain the Shield Advanced automatic mitigation rule in the updated web ACL. The automatic mitigation protections provided by Shield Advanced are not interrupted by your update to the web ACL.

Don't manage the Shield Advanced rule in your CloudFormation web ACL template. The web ACL template shouldn't list the Shield Advanced rule. Follow the best practices for web ACL management at Best practices for using automatic application layer DDoS mitigation.

Health-based detection using health checks with Shield Advanced and Route 53

You can configure Shield Advanced to use health-based detection for improved responsiveness and accuracy in attack detection and mitigation. You can use this option with any resource type except for Route 53 hosted zones.

To configure health-based detection, you define a health check for your resource in Route 53, verify that it's reporting healthy, and then associate it with your Shield Advanced protection. For information about Route 53 health checks, see How Amazon Route 53 checks the health of your resources and Creating, updating, and deleting health checks in the Amazon Route 53 Developer Guide.

> **Note**
>
> Health checks are required for Shield Response Team (SRT) proactive engagement support. For information about proactive engagement, see Setting up proactive engagement for the SRT to contact you directly.

Health checks measure the health of your resources based on the requirements that you define. The health check status provides vital input to the Shield Advanced detection mechanisms, giving them greater sensitivity to the current state of your specific applications.

You can enable health-based detection for any resource type except for Route 53 hosted zones.

- **Network and transport layer (layer 3/layer 4) resources** – Health-based detection improves the accuracy of network-layer and transport-layer event detection and mitigation for Network Load Balancers, Elastic IP addresses, and Global Accelerator standard accelerators. When you protect these resource types with Shield Advanced, Shield Advanced can provide mitigations for smaller attacks and faster mitigation for attacks, even when traffic is within the application's capacity.

  When you add health-based detection, during periods when the associated health check is unhealthy, Shield Advanced can place mitigations even more quickly and at even lower thresholds.

- **Application layer (layer 7) resources** – Health-based detection improves the accuracy of web request flood detection for CloudFront distributions and Application Load Balancers. When you protect these resource types with Shield Advanced, you receive web request flood detection alerts when there's a statistically significant deviation in traffic volume that's combined with significant changes in traffic patterns, based on request characteristics.

  With health-based detection, when the associated Route 53 health check is unhealthy, Shield Advanced requires smaller deviations to alert and it reports events more quickly. Conversely, when the associated Route 53 health check is healthy, Shield Advanced requires larger deviations to alert.

You'll benefit the most from using a health check with Shield Advanced if the health check only reports healthy when your application is running within acceptable parameters and only reports unhealthy when it's not. Use the guidance in this section to manage your health check associations in Shield Advanced.

> **Note**
>
> Shield Advanced doesn't automatically manage your health checks.

The following are required to use a health check with Shield Advanced:

- The health check must report healthy when you associate it with your Shield Advanced protection.
- The health check must be relevant to the health of your protected resource. You are responsible for defining and maintaining health checks that accurately report the health of your application, based on your application's specific requirements.

- The health check must remain available for use by the Shield Advanced protection. Don't delete a health check in Route 53 that you're using for a Shield Advanced protection.

Contents

- Best practices for using health checks with Shield Advanced
- CloudWatch metrics commonly used for health checks with Shield Advanced
  - Metrics used to monitor application health
  - Amazon CloudWatch metrics for each resource type
- Associating a health check with your resource protected by Shield Advanced
- Disassociating a health check from your resource protected by Shield Advanced
- Viewing health check association status in Shield Advanced
- Health check examples for Shield Advanced
  - Amazon CloudFront distributions
  - Load balancers
  - Amazon EC2 elastic IP address (EIP)

Best practices for using health checks with Shield Advanced

Follow the best practices in this section when you create and use health checks with Shield Advanced.

- Plan your health checks by identifying the components of your infrastructure that you want to monitor. Consider the following resource types for health checks:
  - Critical resources.
  - Any resources where you want higher sensitivity in Shield Advanced detection and mitigation.
  - Resources for which you want Shield Advanced to proactively reach out to you. Proactive engagement is informed by the status of your health checks.

  Examples of resources that you might want to monitor include Amazon CloudFront distributions, internet-facing load balancers, and Amazon EC2 instances.
- Define health checks that accurately reflect the health of your application origin with as few notifications as possible.

- Write health checks so that they're only unhealthy when your application is unavailable or isn't performing within acceptable parameters. You are responsible for defining and maintaining health checks based on your application's specific requirements.
- Use as few health checks as possible while still accurately reporting on the health of your application. For example, multiple alarms from multiple areas of your application that all report the same problem might add overhead to your response activities without adding informational value.
- Use calculated health checks to monitor application health using a combination of Amazon CloudWatch metrics. For example, you can calculate combined health based on the latency of your application servers and their 5xx error rates, which indicate that the origin server didn't fulfill the request.
- Create and publish your own application health indicators to CloudWatch custom metrics as needed and use them in a calculated health check.

- Implement and manage your health checks to improve detection and reduce unnecessary maintenance activities.
  - Before you associate a health check with a Shield Advanced protection, make sure that it's in a healthy state. Associating a health check that's reporting unhealthy can skew the Shield Advanced detection mechanisms for your protected resources.
  - Keep your health checks available for use by Shield Advanced. Don't delete a health check in Route 53 that you're using for a Shield Advanced protection.
  - Use staging and test environments only to test your health checks. Only maintain health check associations for environments that require production-level performance and availability. Don't maintain health check association in Shield Advanced for staging and test environments.

CloudWatch metrics commonly used for health checks with Shield Advanced

This section lists the Amazon CloudWatch metrics that are commonly used in health checks to measure application health during distributed denial of service (DDoS) events. For full information about the CloudWatch metrics for each resource type, see the list that follows the table.

Topics

- Metrics used to monitor application health
- Amazon CloudWatch metrics for each resource type

Metrics used to monitor application health

Resource	Metric	Description
Route 53	`HealthCheckStatus`	The status of the health check endpoint.
CloudFront	`5xxErrorRate`	The percentage of all requests for which the HTTP status code is 5xx. This indicates an attack that's impacting the application.
Application Load Balancer	`HTTPCode_ELB_5XX_Count`	The number of HTTP 5xx client error codes generated by the load balancer.
Application Load Balancer	`RejectedConnectionCount`	The number of connections that were rejected because the load balancer reached its maximum number of connections.
Application Load Balancer	`TargetConnectionErrorCount`	The number of connections that were not successfully established between the load balancer and the target.
Application Load Balancer	`TargetResponseTime`	The time elapsed in seconds after the request leaves the load balancer and when it receives a response from the target.
Application Load Balancer	`UnHealthyHostCount`	The number of targets that are considered unhealthy.

Resource	Metric	Description
Amazon EC2	`CPUUtilization`	The percentage of allocated EC2 compute units that are currently in use.

Amazon CloudWatch metrics for each resource type

For additional information about the metrics that are available for your protected resources, see the following sections in the resource guides:

- Amazon Route 53 – Monitoring your resources with Amazon Route 53 health checks and Amazon CloudWatch in the Amazon Route 53 Developer Guide.
- Amazon CloudFront – Monitoring CloudFront with Amazon CloudWatch in the Amazon CloudFront Developer Guide.
- Application Load Balancer – CloudWatch metrics for your Application Load Balancer in the User Guide for Application Load Balancers.
- Network Load Balancer – CloudWatch metrics for your Network Load Balancer in the User Guide for Network Load Balancers.
- AWS Global Accelerator – Using Amazon CloudWatch with AWS Global Accelerator in the AWS Global Accelerator Developer Guide.
- Amazon Elastic Compute Cloud – List the available CloudWatch metrics for your instances in the https://docs.aws.eu/AWSEC2/latest/UserGuide/.
- Amazon EC2 Auto Scaling – Monitoring CloudWatch metrics for your Auto Scaling groups and instances in the Amazon EC2 Auto Scaling User Guide.

Associating a health check with your resource protected by Shield Advanced

The following procedure shows how to associate an Amazon Route 53 health check with a protected resource.

> **Note**
>
> Before you associate a health check with a Shield Advanced protection, make sure that it's in a healthy state. For information, see Monitoring health check status and getting notifications in the Amazon Route 53 Developer Guide.

To associate a health check

1. Sign in to the AWS Management Console and open the AWS WAF & Shield console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.
2. In the AWS Shield navigation pane, choose **Protected resources**.
3. On the **Protections** tab, select the resource that you want to associate with a health check.
4. Choose **Configure protections**.
5. Choose **Next** until you get to the page **Configure health check based DDoS detection - *optional***.
6. Under **Associated Health Check**, choose the ID of the health check that you want to associate with the protection.

   > **Note**
   >
   > If you do not see the health check you need, go to the Route 53 console and verify the health check and its ID. For information, see Creating and Updating Health Checks.

7. Walk through the rest of the pages until you finish the configuration. On the **Protections** page, your updated health check association is listed for the resource.
8. On the **Protections** page, check that your newly associated health check is reporting healthy.

   You can't successfully begin using a health check in Shield Advanced while the health check is reporting unhealthy. Doing so causes Shield Advanced to detect false positives at very low thresholds and can also negatively impact the ability of the Shield Response Team (SRT) to provide proactive engagement for the resource.

   If the newly associated health check is reporting unhealthy, do the following:

   a. Disassociate the health check from your protection in Shield Advanced.

b. Revisit your health check specifications in Amazon Route 53 and verify your overall application performance and availability.

c. When your application is performing within your parameters for good health and your health check is reporting healthy, try again to associate the health check in Shield Advanced.

The health check association procedure is complete when you've established your new health check association and it reports healthy in Shield Advanced.

Disassociating a health check from your resource protected by Shield Advanced

The following procedure shows how to disassociate an Amazon Route 53 health check from a protected resource.

To disassociate a health check

1. Sign in to the AWS Management Console and open the AWS WAF & Shield console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.
2. In the AWS Shield navigation pane, choose **Protected resources**.
3. On the **Protections** tab, select the resource that you want to disassociate from a health check.
4. Choose **Configure protections**.
5. Choose **Next** until you get to the page **Configure health check based DDoS detection - *optional***.
6. Under **Associated Health Check**, choose the empty option, listed as **-**.
7. Walk through the rest of the pages until you finish the configuration.

On the **Protections** page, the health check field for your resource is set to **-**, indicating no health check association.

Viewing health check association status in Shield Advanced

You can see the status of the health check that's associated with a protection on the AWS WAF & Shield console **Protected resources** page and on the details page of each resource.

- **Healthy** – The health check is available and is reporting healthy.
- **Unhealthy** – The health check is available and is reporting unhealthy.

- **Unavailable** – The health check is not available for use by Shield Advanced.

To resolve an Unavailable health check

Create and use a new health check. Don't try to associate a health check again after it has had a status of unavailable in Shield Advanced.

For detailed guidance on following these steps, see the preceding topics.

1. In Shield Advanced, disassociate the health check from the resource.
2. In Route 53, create a new health check for the resource and note its ID. For information, see Creating and Updating Health Checks in the Amazon Route 53 Developer Guide.
3. In Shield Advanced, associate the new health check with the resource.

Health check examples for Shield Advanced

This section shows examples of health checks that you could use in a calculated health check. A calculated health check uses a number of individual health checks to determine a combined status. The status of each individual health check is based on the health of an endpoint or on the state of an Amazon CloudWatch metric. You combine health checks into a calculated health check and then configure your calculated health check to report health based on the combined health status of the individual health checks. Tune the sensitivity of your calculated health checks according to your requirements for application performance and availability.

For information about calculated health checks, see Monitoring other health checks (calculated health checks) in the Amazon Route 53 Developer Guide. For additional information, see the blog post Route 53 Improvements – Calculated Health Checks and Latency Checks.

Topics

- Amazon CloudFront distributions
- Load balancers
- Amazon EC2 elastic IP address (EIP)

Amazon CloudFront distributions

The following examples describe health checks that could be combined into a calculated health check for a CloudFront distribution:

- Monitor an endpoint by specifying a domain name to a path on the distribution that's serving dynamic content. A healthy response would include HTTP response codes 2xx and 3xx.
- Monitor the state of a CloudWatch alarm that's measuring the health of the CloudFront origin. For example, you can maintain a CloudWatch alarm on the Application Load Balancer metric `TargetResponseTime`, and create a health check that reflects the status of the alarm. The health check can be unhealthy when the response time, between request leaving the load balancer and when the load balancer receives a response from the target, exceeds the threshold configured in the alarm.
- Monitor the state of a CloudWatch alarm that measures the percentage of requests for which the response's HTTP status code is 5xx. If the CloudFront distribution's 5xx error rate is higher than the threshold defined in the CloudWatch alarm, the status of this health check will switch to unhealthy.

Load balancers

The following examples describe health checks that could be used in calculated health checks for an Application Load Balancer, Network Load Balancer, or Global Accelerator standard accelerator.

- Monitor the state of a CloudWatch alarm that measures the number of new connections established by clients to the load balancer. You can set the alarm threshold for the average number of new connections at some degree higher than your every day average. The metrics for each resource type are the following:
  - Application Load Balancer: `NewConnectionCount`
  - Network Load Balancer: `ActiveFlowCount`
  - Global Accelerator: `NewFlowCount`
- For Application Load Balancer and Network Load Balancer, monitor the state of a CloudWatch alarm that measures the number of load balancers that are considered healthy. You can set the alarm threshold either on Availability Zone or on the minimum number of healthy hosts that your load balancer requires. The available metrics for the load balancer resources are as follows:
  - Application Load Balancer: `HealthyHostCount`
  - Network Load Balancer: `HealthyHostCount`
- For Application Load Balancer, monitor the state of a CloudWatch alarm that measures the number of HTTP 5xx response codes generated by the load balancer targets. For an Application Load Balancer, you can use the metric `HTTPCode_Target_5XX_Count` and base the alarm threshold on the sum of all 5xx errors for the load balancer.

Amazon EC2 elastic IP address (EIP)

The following example health checks could be combined into a calculated health check for an Amazon EC2 elastic IP address:

- Monitor an endpoint by specifying an IP address to the Elastic IP address. The health check will remain healthy as long as a TCP connection can be established with the resource behind the IP address.
- Monitor the state of a CloudWatch alarm that measures the percentage of allocated Amazon EC2 compute units that are currently in use on the instance. You can use the Amazon EC2 metric `CPUUtilization` and base the alarm threshold on what you consider to be a high CPU utilization rate for your application, such as 90%.

Adding AWS Shield Advanced protection to AWS resources

Follow the guidance in this section to add Shield Advanced protection to one or more resources.

To add protection for an AWS resource

1. Sign in to the AWS Management Console and open the AWS WAF & Shield console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.
2. In the navigation pane, under AWS Shield choose **Protected resources**.
3. Choose **Add resources to protect**.
4. In the **Choose resources to protect with Shield Advanced** page, in **Specify the Region and resource types**, provide the Region and resource type specifications for the resources that you want to protect. You can protect resources in multiple Regions by selecting **All Regions** and you can narrow the selection to global resources by selecting **Global**. You can deselect any resource types that you do not want to protect. For information about protections for your resource types, see List of resources that AWS Shield Advanced protects.
5. Choose **Load resources**. Shield Advanced populates the **Select Resources** section with the AWS resources that match your criteria.
6. In the **Select Resources** section, you can filter the list of resources by entering a string to search for in the resource listings.

   Select the resources that you want to protect.

7. In the **Tags** section, if you want to add tags to the Shield Advanced protections that you are creating, specify those. For information about tagging AWS resources, see Working with Tag Editor.
8. Choose **Protect with Shield Advanced**. This adds Shield Advanced protections to the resources.

Editing AWS Shield Advanced protections

You can change the settings for your AWS Shield Advanced protections at any time. To do this, walk through the options for your selected protections and modify the settings that you need to change.

To manage protected resources

1. Sign in to the AWS Management Console and open the AWS WAF & Shield console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.
2. In the AWS Shield navigation pane, choose **Protected resources**.
3. In the **Protections** tab, select the resources that you want to protect.
4. Choose **Configure protections** and the resource specification option that you want.
5. Walk through each of the resource protection options, making changes as needed.

Configure application layer DDoS protections

For protection against attacks on Amazon CloudFront and Application Load Balancer resources, you can add AWS WAF web ACLs and add rate-based rules. For information about this, see Protecting the application layer with AWS WAF web ACLs and Shield Advanced.

You can also enable the Shield Advanced automatic application layer DDoS mitigation. For information about how AWS WAF works, see AWS WAF. For information about the automatic mitigation feature, see Automating application layer DDoS mitigation with Shield Advanced.

> **Important**
>
> If you manage your Shield Advanced protections through AWS Firewall Manager using a Shield Advanced policy, you can't manage the application layer protections here. For

all other resources, we recommend that, at a minimum, you attach a web ACL to each resource, even if web ACL doesn't contain any rules.

Note

When you enable automatic application layer DDoS mitigation for a resource, if needed, the operation automatically adds a service-linked role to your account to give Shield Advanced the permissions it needs to manage your web ACL protections. For information, see Using service-linked roles for Shield Advanced.

To configure application layer DDoS protections

1. In the **Configure layer 7 DDoS protections** page, if the resource isn't already associated with a web ACL, you can choose an existing web ACL or create your own.

   To create a web ACL, follow these steps:

   a. Choose **Create web ACL**.

   b. Enter a name. You can't change the name after you create the web ACL.

   c. Choose **Create**.

   Note

   If a resource is already associated with a web ACL, you can't change to a different web ACL. If you want to change the web ACL, you must first remove the associated web ACLs from the resource. For more information, see Associating or disassociating protection with an AWS resource.

2. If the web ACL doesn't have a rate-based rule defined, you can add one by choosing **Add rate limit rule** and then performing the following steps:

   a. Enter a name.

   b. Enter a rate limit. This is the maximum number of requests allowed in any five minute period from any single IP address before the rate-based rule action is applied to the

IP address. When the requests from the IP address fall below the limit, the action is discontinued.

c. Set the rule action to count or block requests from IP addresses while their request counts are over the limit. The application and removal of the rule action might take effect a minute or two after the IP address request rate changes.

d. Choose **Add rule**.

3. For **Automatic application layer DDoS mitigation**, choose whether you want Shield Advanced to automatically mitigate DDoS attacks on your behalf, as follows:

   - To enable automatic mitigation, choose **Enable** and then select the AWS WAF rule action that you want Shield Advanced to use in its custom rules. Your choices are Count and Block. For information about these AWS WAF rule actions, see Using rule actions in AWS WAF. For information about how Shield Advanced manages this action setting, see How Shield Advanced manages the rule action setting.
   - To disable automatic mitigation, choose **Disable**.
   - To leave the automatic mitigation settings unchanged for the resources that you're managing, leave the default choice **Keep current settings**.

   For information about Shield Advanced automatic application layer DDoS mitigation, see Automating application layer DDoS mitigation with Shield Advanced.

4. Choose **Next**.

Creating alarms and notifications for resources protected by Shield Advanced

The following procedure shows how to manage CloudWatch alarms for protected resources.

> **Note**
>
> CloudWatch incurs additional costs. For CloudWatch pricing, see Amazon CloudWatch Pricing.

To create alarms and notifications

1. In the protections page **Create alarms and notifications - *optional***, configure the SNS topics for the alarms and notifications that you want to receive. For resources that you don't want notifications for, choose **No topic**. You can add an Amazon SNS topic or create a new topic.
2. To create an Amazon SNS topic, follow these steps:
   a. In the dropdown list, choose **Create an SNS topic**.
   b. Enter a topic name.
   c. Optionally enter an email address that the Amazon SNS messages will be sent to, and then choose **Add email**. You can enter more than one.
   d. Choose **Create**.
3. Choose **Next**.

Removing AWS Shield Advanced protection from an AWS resource

You can remove AWS Shield Advanced protection from any of your AWS resources at any time.

> **Important**
>
> Deleting an AWS resource doesn't remove the resource from AWS Shield Advanced. You must also remove the protection on the resource from AWS Shield Advanced, as described in this procedure.

Remove AWS Shield Advanced protection from an AWS resource

1. Sign in to the AWS Management Console and open the AWS WAF & Shield console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.
2. In the AWS Shield navigation pane, choose **Protected resources**.
3. In the **Protections** tab, select the resources whose protections you want to remove.
4. Choose **Delete protections**.
   - If you have an Amazon CloudWatch alarm configured for a protection, you are given the option to delete the alarm along with the protection. If you choose not to delete the alarm at this point, you can instead delete it later using the CloudWatch console.

> **Note**
>
> For protections that have an Amazon Route 53 health check configured, if you add the protection again later, the protection still includes the health check.

The preceding steps remove AWS Shield Advanced protection from specific AWS resources. They don't cancel your AWS Shield Advanced subscription. You will continue to be charged for the service. For information about your AWS Shield Advanced subscription, contact the AWS Support Center.

Removing a CloudWatch alarm from your Shield Advanced protections

To remove a CloudWatch alarm from your Shield Advanced protections, do one of the following:

- Delete the protection as described in Removing AWS Shield Advanced protection from an AWS resource. Be sure to select the check box next to **Also delete related DDoSDetection alarm**.
- Delete the alarm using the CloudWatch console. The name of the alarm to delete starts with **DDoSDetectedAlarmForProtection**.

Grouping your AWS Shield Advanced protections

Use protection groups to create logical collections of your protected resources and manage their protections as a group. For information about managing resource protections, see Editing AWS Shield Advanced protections.

> **Note**
>
> Automatic application layer DDoS mitigation does not interact with protection groups. You can enable automatic mitigation for resources that are in protection groups, but Shield Advanced does not automatically apply attack mitigations based on protection group findings. Shield Advanced applies automatic attack mitigations for individual resources.

AWS Shield Advanced protection groups give you a self-service way to customize the scope of detection and mitigation by treating multiple protected resources as a single unit. Resource grouping can provide a number of benefits.

- Improve accuracy of detection.
- Reduce unactionable event notifications.
- Increase coverage of mitigation actions to include protected resources that also might be affected during an event.
- Accelerate time to mitigation of attacks with multiple similar targets.
- Facilitate automatic protection of newly created protected resources.

Protection groups can help reduce false positives in situations such as blue/green swap, where resources alternate between being near zero load and fully loaded. Another example is when you create and delete resources frequently while maintaining a load level that's shared among the members of the group. For situations such as these, monitoring individual resources can lead to false positives, while monitoring the health of the group of resources does not.

You can configure protection groups to include all protected resources, all resources of specific resource types, or individually specified resources. Newly protected resources that satisfy your protection group criteria are automatically included in your protection group. A protected resource can belong to multiple protection groups.

Creating a Shield Advanced protection group

To create a protection group

1. Sign in to the AWS Management Console and open the AWS WAF & Shield console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.
2. In the AWS Shield navigation pane, choose **Protected resources**.
3. Choose the **Protection groups** tab, then choose **Create protection group**.
4. In the **Create protection group** page, provide a name for your group. You'll use this name to identify the group in your list of protected resources. You can't change the name of a protection group after you create it.
5. For **Protection grouping criteria**, select the criteria that you want Shield Advanced to use to identify the protected resources to include in the group. Make your additional selections based on the criteria that you've chosen.

6. For **Aggregation**, select how you want Shield Advanced to combine resource data for the group in order to detect, mitigate, and report events.

   - **Sum** – Use the total traffic across the group. This is a good choice for most cases. Examples include Elastic IP addresses for Amazon EC2 instances that scale manually or automatically.
   - **Mean** – Use the average of the traffic across the group. This is a good choice for resources that share traffic uniformly. Examples include accelerators and load balancers.
   - **Max** – Use the highest traffic from each resource. This is useful for resources that don't share traffic, and for resources that share traffic in a non-uniform way. Examples include Amazon CloudFront distributions and origin resources for CloudFront distributions.

7. Choose **Save** to save your protection group and return to the **Protected resources** page.

In the **Shield Events** page, you can view events for your protection group and drill down to see additional information for the protected resources that are in the group.

Updating a Shield Advanced protection group

To update a protection group

1. Sign in to the AWS Management Console and open the AWS WAF & Shield console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.
2. In the AWS Shield navigation pane, choose **Protected resources**.
3. In the **Protection groups** tab, select the check box next to the protection group that you want to modify.
4. In the protection group's page, choose **Edit**. Make your changes to the protection group settings.
5. Choose **Save** to save your changes.

Deleting a Shield Advanced protection group

To delete a protection group

1. Sign in to the AWS Management Console and open the AWS WAF & Shield console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.
2. In the AWS Shield navigation pane, choose **Protected resources**.

3. In the **Protection groups** tab, select the check box next to the protection group that you want to remove.
4. In the protection group's page, choose **Delete** and confirm the action.

Tracking Shield Advanced resource protection changes in AWS Config

This page explains how to record changes to the AWS Shield Advanced protection of your resources using AWS Config. You can then use this information to maintain a configuration change history for audit and troubleshooting purposes.

To record protection changes, enable AWS Config for each resource that you want to track. For more information, see Getting Started with AWS Config in the *AWS Config Developer Guide*.

You must enable AWS Config for each AWS Region that contains the tracked resources. You can enable AWS Config manually, or you can use the CloudFormation template "Enable AWS Config" at CloudFormation StackSets Sample Templates in the *CloudFormation User Guide*.

If you enable AWS Config, you're charged as detailed on the AWS Config Pricing page.

> **Note**
>
> If you already have AWS Config enabled for the necessary Regions and resources, you don't need to do anything. AWS Config logs regarding protection changes to your resources start populating automatically.

After enabling AWS Config, use the US East (N. Virginia) Region in the AWS Config console to view the configuration change history for AWS Shield Advanced global resources.

View the change history for AWS Shield Advanced regional resources via the AWS Config console in the US East (N. Virginia), US East (Ohio), US West (Oregon), US West (N. California), Europe (Ireland), Europe (Frankfurt), Asia Pacific (Tokyo), and Asia Pacific (Sydney) Regions.

Visibility into DDoS events with Shield Advanced

AWS Shield provides visibility into the following categories of events and event activities:

- **Global** – All customers can access an aggregated view of global threat activity over the last two weeks. You can see this information under the **Getting Started** and **Global threat dashboard**

pages of the AWS Shield console. For more information, see Viewing AWS Shield global and account activity.

- **Account** – All customers can access a summary of the events for their account over the prior year. You can see this information under the **Getting Started** page of the AWS Shield console. For more information, see Viewing AWS Shield global and account activity.

When you subscribe to Shield Advanced and add protections to your resources, you gain access to additional information about the events and DDoS attacks on the protected resources:

- **Events on protected resources** – Shield Advanced provides detailed information for each event through the **Events** page of the AWS Shield console. For more information, see Viewing AWS Shield Advanced events.
- **Event metrics for protected resources** – Shield Advanced publishes detection, mitigation, and top contributor Amazon CloudWatch metrics for all resources that it protects. You can use these metrics to configure CloudWatch dashboards and alarms. For more information, see AWS Shield Advanced metrics.
- **Cross-account event visibility for protected resources** – If you use AWS Firewall Manager to manage your Shield Advanced protections, you can enable visibility into protections across multiple accounts by using Firewall Manager combined with AWS Security Hub CSPM. For more information, see Viewing Shield Advanced events across multiple AWS accounts with AWS Firewall Manager and AWS Security Hub CSPM.

If you enable automatic application layer DDoS mitigation for an application layer protection, Shield Advanced adds a rule group to your protection pack (web ACL) that it uses to manage automated protections. This rule group generates AWS WAF metrics, but they are not available to view. This is the same as for any other rule groups that you use in your protection pack (web ACL) but do not own, such as AWS Managed Rules rule groups. For more information about AWS WAF metrics, see AWS WAF metrics and dimensions. For information about this Shield Advanced protection option, see Automating application layer DDoS mitigation with Shield Advanced.

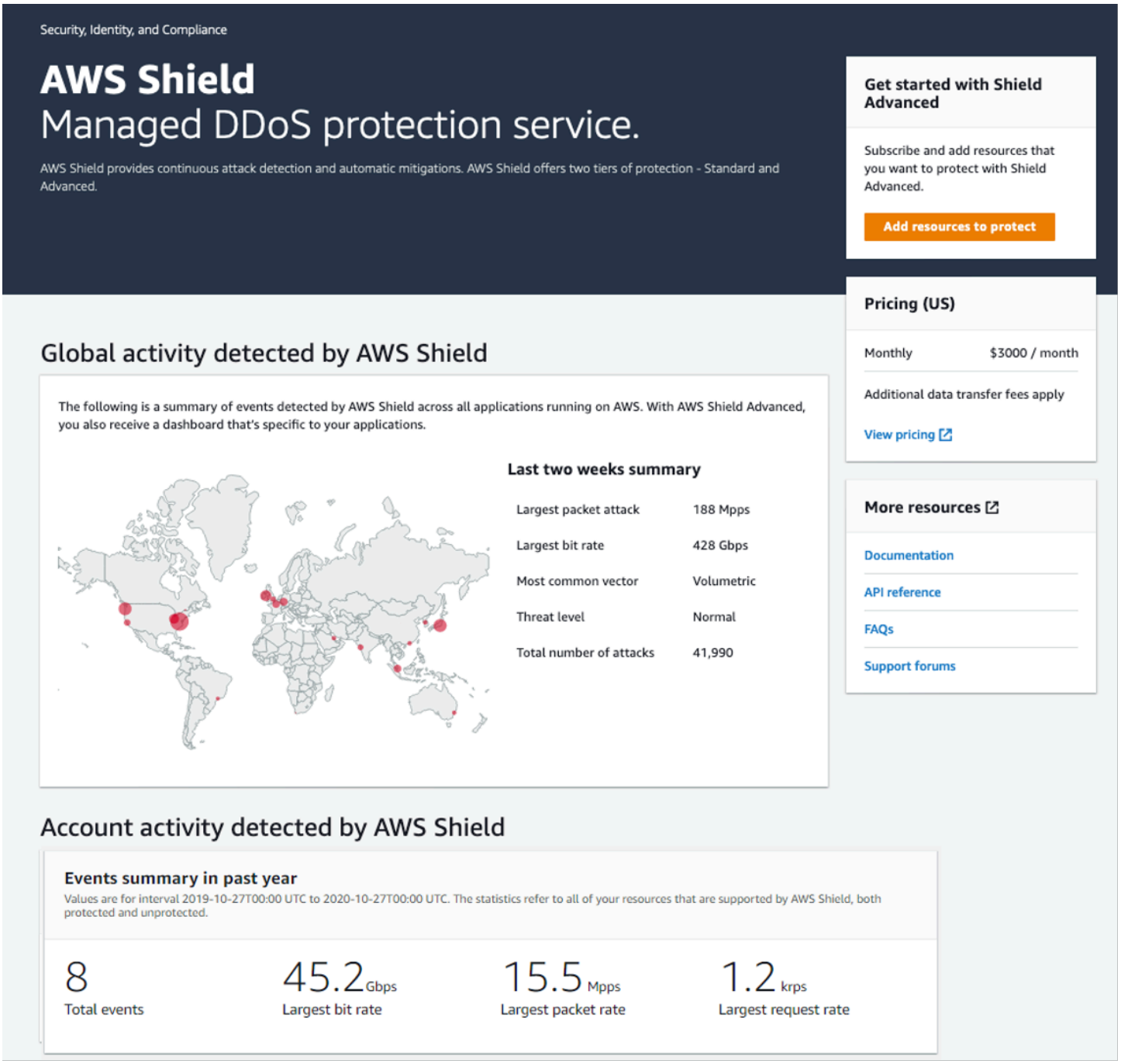
Topics

- Viewing AWS Shield global and account activity
- Viewing AWS Shield Advanced events
- Viewing Shield Advanced events across multiple AWS accounts with AWS Firewall Manager and AWS Security Hub CSPM

Viewing AWS Shield global and account activity

This page provides instructions for accessing an aggregated view of global threat activity and a per-account event summary in the AWS Shield console **Getting Started** and **Global threat dashboard** pages.

The following screenshot shows an example **Getting Started** page.



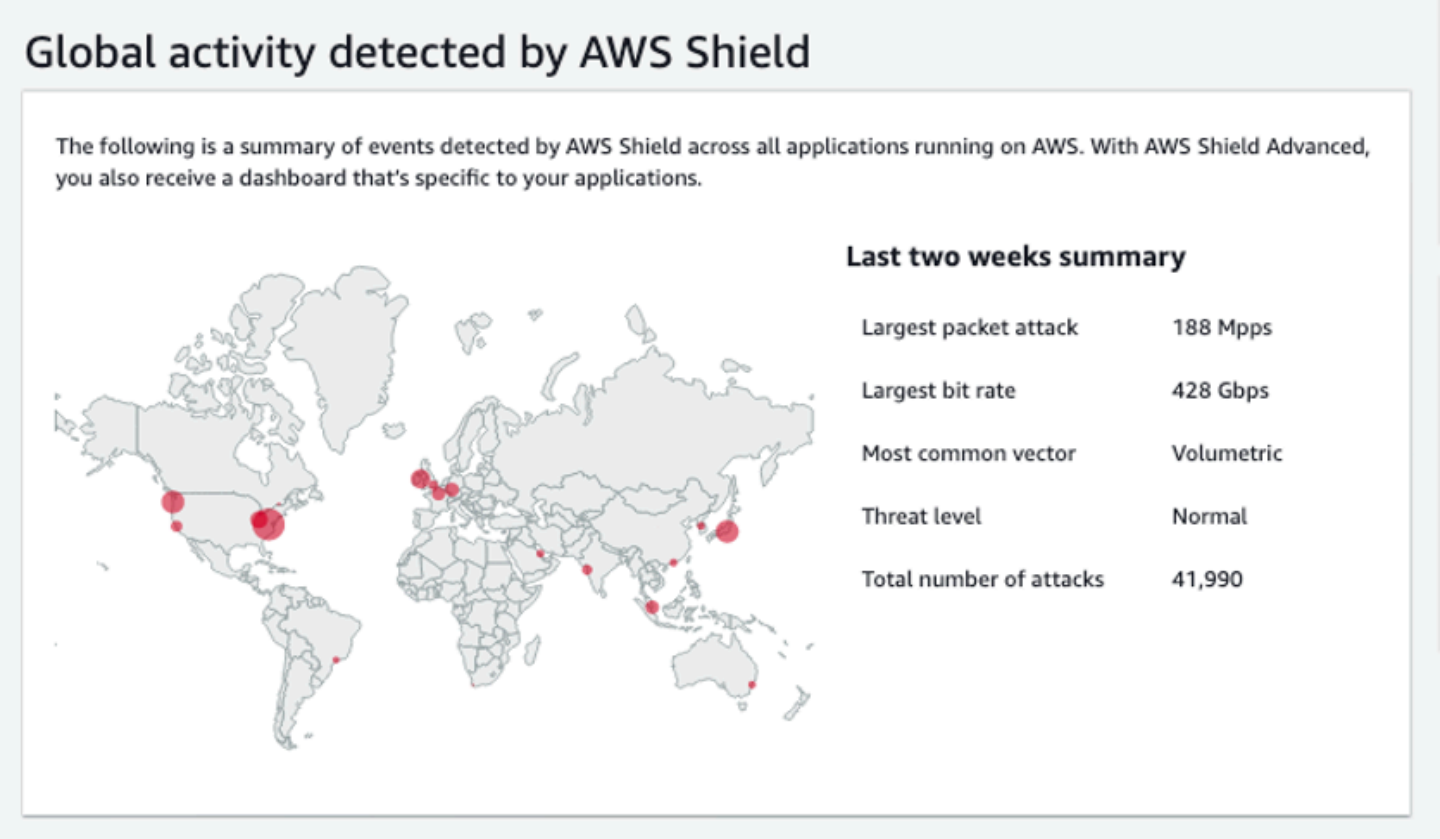
To access the AWS Shield console

- Sign in to the AWS Management Console and open the AWS WAF & Shield console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.

You don't need a subscription to Shield Advanced to access global activity and account event summary information.

Global activity

This information is available through the AWS Shield console **Global threat dashboard** and **Getting Started** pages. The following screenshot shows an example of the global activity pane.



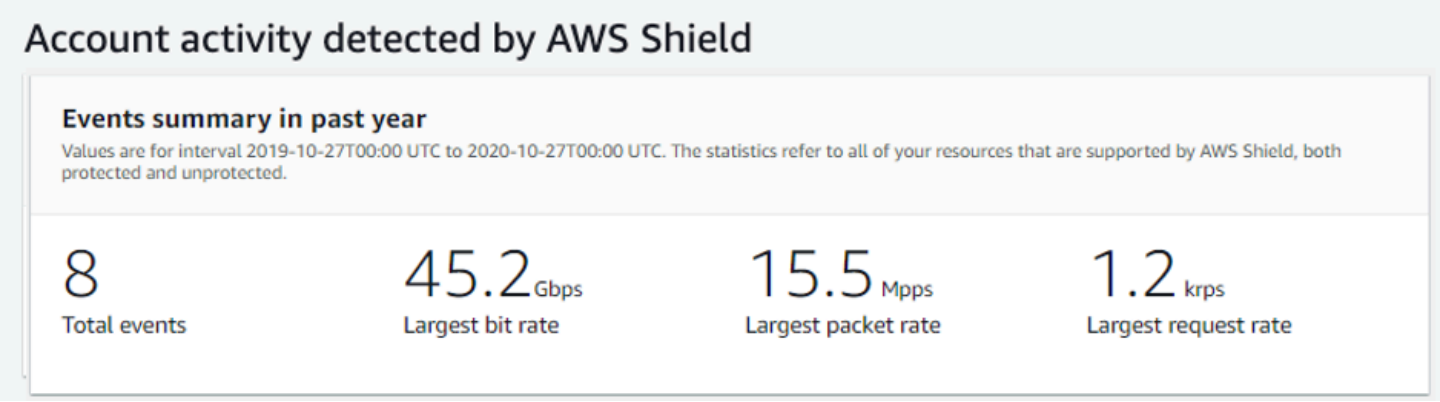
Global activity describes DDoS events observed across all AWS customers. Once per hour, AWS updates the information for the prior two weeks. In the console pane, you can see the results, partitioned by AWS Region and displayed on a world heat map. Next to the map, Shield displays summary information such as the largest packet attack, largest bit rate, most common vector, total number of attacks, and threat level. The threat level is an assessment of the current global activity compared to what AWS typically observes. The default threat level value is **Normal**. AWS automatically updates the value to **High** for elevated DDoS activity.

The **Global threat dashboard** also provides time-series metrics and gives you the ability to change between time durations. To view the history of significant DDoS attacks, you can customize the dashboard for views from the last day to the last two weeks. Time-series metrics provide a view of the largest bit rate, packet rate, or request rate for all events detected by AWS Shield for applications running on AWS during the time window that you select.

Account activity

This information is available in the AWS Shield console **Getting Started** page.

The following screenshot shows an example account activity pane.



Account activity describes DDoS events that Shield detected for your resources that are eligible for protection by Shield Advanced. Each day, Shield creates summary metrics for the year ending at 00:00 UTC the prior day, and then displays total events, largest bit rate, largest packet rate, and largest request rate.

- The total events metric reflects every time that Shield observed suspicious attributes in traffic that was destined to your application. Suspicious attributes can include traffic that is at a higher than normal volume, traffic that does not match your application's historical profile, or traffic that does not match heuristics that are defined by Shield for valid application traffic.
- Largest bit rate and largest packet rate statistics are available for every resource.
- The largest request rate statistic is available only for Amazon CloudFront distributions and Application Load Balancers that have an associated AWS WAF web ACL.

> **Note**
>
> You can also access the account level event summary through the AWS Shield API operation DescribeAttackStatistics.

Viewing AWS Shield Advanced events

This page provides instructions for accessing information about events in Shield Advanced.

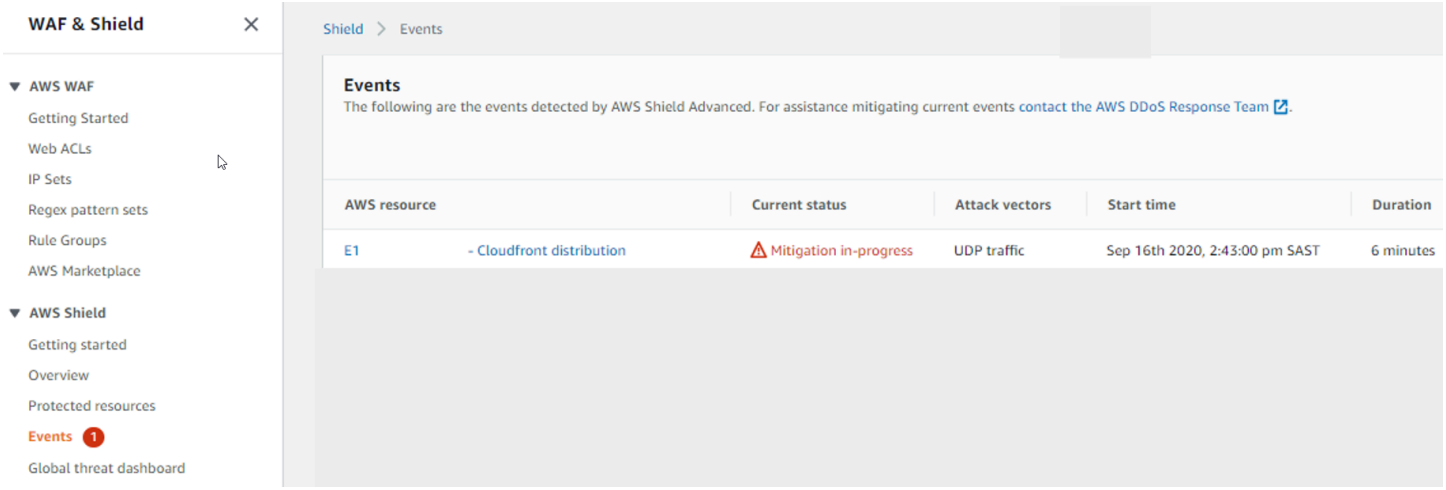
When you subscribe to Shield Advanced, and protect your resources, you gain access to additional visibility features for the resources. These include near real-time notification of events that are detected by Shield Advanced and additional information about detected events and mitigations.

> **Note**
>
> Your event information in the Shield Advanced console is based on Shield Advanced metrics. For information about Shield Advanced metrics, see AWS Shield Advanced metrics

AWS Shield evaluates traffic to your protected resource along multiple dimensions. When an anomaly is detected, Shield Advanced creates a separate event for each resource that's affected.

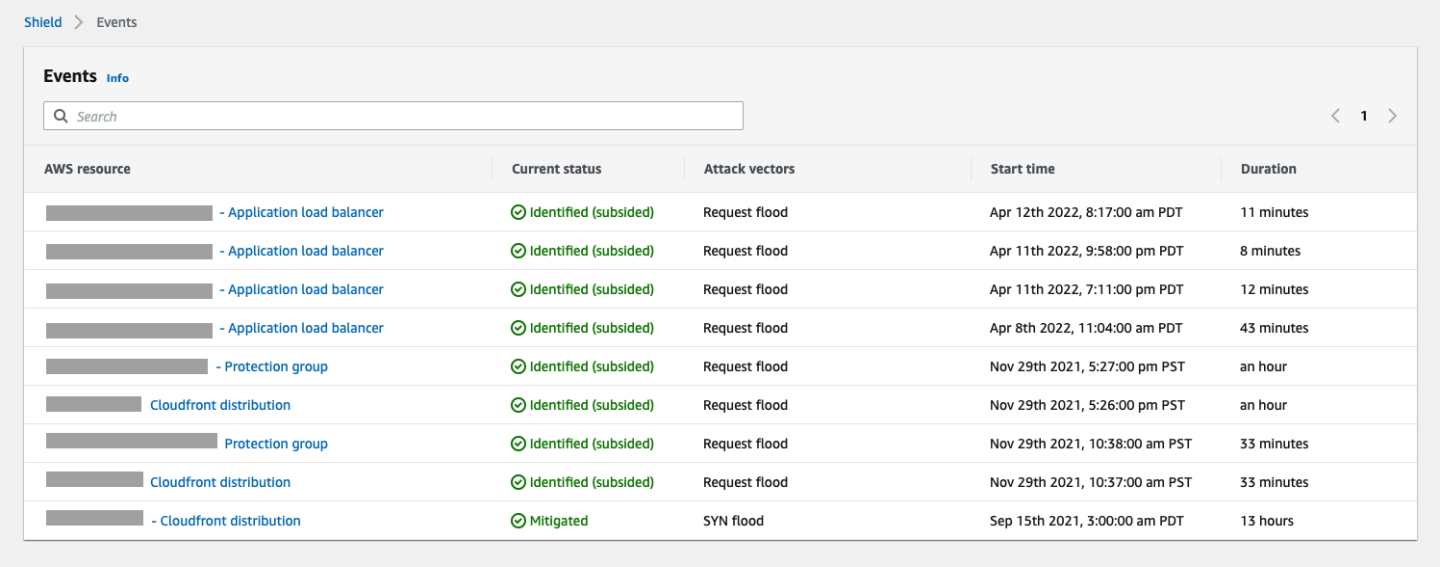
You can access event summaries and details through the **Events** page of the Shield console. The top level **Events** page provides an overview of current and past events.

The following screenshot shows an example **Events** page with a single ongoing event. This active event is also flagged in the left navigation pane.



Shield Advanced might also automatically place mitigations against attacks, depending on the traffic type and on your configured protections. These mitigations can protect your resource from receiving excess traffic or traffic that matches a known DDoS attack signature.

The following screenshot shows an example **Events** listing where all events have been mitigated by Shield Advanced or have subsided on their own.



Protect your resources before an event

Improve the accuracy of event detection by protecting resources with Shield Advanced while they are receiving the normal expected traffic, before they are subject to a DDoS attack.

In order to accurately report events for a protected resource, Shield Advanced must first establish a baseline of expected traffic patterns for it.

- Shield Advanced reports infrastructure layer events for resources after they've been protected for at least 15 minutes.
- Shield Advanced reports web application layer events for resources after they've been protected for at least 24 hours. The accuracy of detection for application layer events is best after Shield Advanced has observed expected traffic for 30 days.

To access events information in the AWS Shield console

1. Sign in to the AWS Management Console and open the AWS WAF & Shield console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/.
2. In the AWS Shield navigation pane, choose **Events**. The console shows the **Events** page.

3. From the **Events** page, you can select any event in the list to see additional summary information and details for the event.

Topics

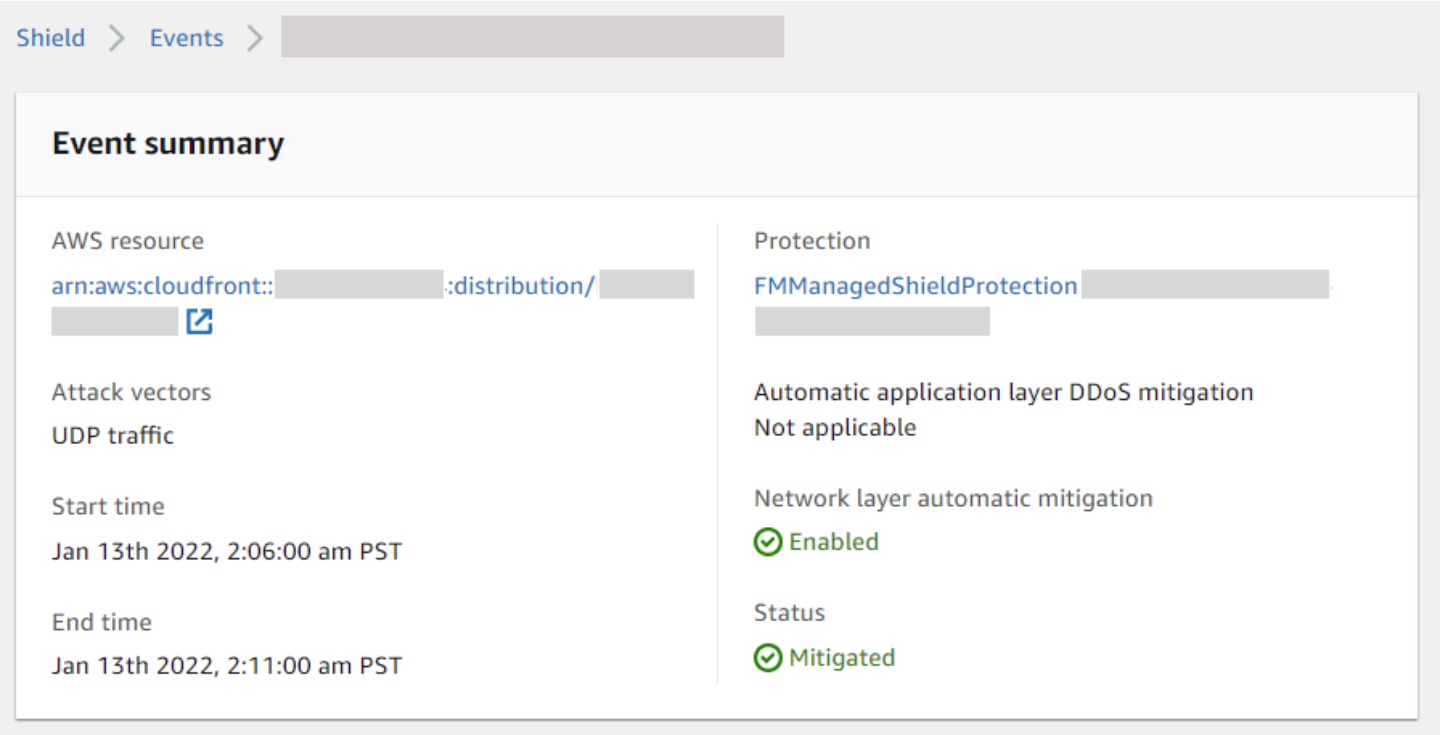
- List of fields in AWS Shield Advanced event summaries
- Viewing AWS Shield Advanced event details

List of fields in AWS Shield Advanced event summaries

This page lists and defines the fields in Shield Advanced event summaries.

You can view summary and detail information for an event in the event's console page. To open the page for an event, select its AWS resource name from the **Events** page list.

The following screenshot shows an example event summary for a network layer event.



The event page summary information includes the following.

- **Current status** – Values that indicate the state of the event and the actions that Shield Advanced has taken on the event. Status values apply to infrastructure layer (layer 3 or 4) and application layer (layer 7) events.

  - **Identified (ongoing)** and **Identified (subsided)** – These indicate that Shield Advanced detected an event, but has taken no action on it so far. **Identified (subsided)** indicates that the suspicious traffic that Shield detected has stopped without intervention.
  - **Mitigation in progress** and **Mitigated** – These indicate that Shield Advanced detected an event and has taken action on it. **Mitigated** is also used when the targeted resource is an Amazon CloudFront distribution or Amazon Route 53 hosted zone, which have their own automatic inline mitigations.
- **Attack vectors** – DDoS attack vectors like TCP SYN floods and Shield Advanced detection heuristics like request flood. These can be indicators of a DDoS attack.
- **Start time** – The date and time that the first anomalous traffic data point was detected.
- **Duration or end time** – Indicates the time elapsed between the event start time and the last observed anomalous data point that Shield Advanced observed. While an event is ongoing, these values will continue to increase.
- **Protection** – Names the Shield Advanced protection that's in associated with the resource, and provides a link to its protection page. This is available in the individual event's page.
- **Automatic application layer DDoS mitigation** – Used for application layer protections, to indicate whether the Shield Advanced automatic application layer DDoS mitigation is enabled for the resource. If it is enabled, this provides a link to access and manage the configuration. This is available in the individual event's page.
- **Network layer automatic mitigation** – Indicates whether the resource has automatic mitigation at the network layer. If a resource has a network layer component, it will have this enabled. This information is available in the individual event's page.

For resources that are frequently targeted, Shield may leave mitigations in place after excess traffic has subsided, to prevent further recurring events.

> **Note**
>
> You can also access event summaries for protected resources through the AWS Shield API operation ListAttacks.

Viewing AWS Shield Advanced event details

You can see details about an event's detection, mitigation, and top contributors in the bottom section of the console page for the event. This section can include a mix of legitimate and potentially unwanted traffic, and may represent both traffic that was passed to your protected resource and traffic that was blocked by Shield mitigations.

- **Detection and mitigation** – Provides information about the observed event and any applied mitigations against it. For information about event mitigation, see Responding to DDoS events in AWS.
- **Top contributors** – Categorizes the traffic that's involved in the event, and lists the primary sources of traffic that Shield has identified for each category. For application layer events, use the top contributors information to get a general idea of the nature of an event, but use the AWS WAF logs for your security decisions. For more information, see the sections that follow.

Your event information in the Shield Advanced console is based on Shield Advanced metrics. For information about Shield Advanced metrics, see AWS Shield Advanced metrics

Mitigation metrics aren't included for Amazon CloudFront or Amazon Route 53 resources, because these services are protected by a mitigation system that's always enabled and doesn't require mitigations for individual resources.

The details sections vary according to whether the information is for an infrastructure layer or application layer event.

Topics

- Viewing application layer (layer 7) event details in Shield Advanced
- Viewing infrastructure layer (layer 3 or 4) event details in Shield Advanced

Viewing application layer (layer 7) event details in Shield Advanced

You can see details about an application layer event's detection, mitigation, and top contributors in the bottom section of the console page for the event. This section can include a mix of legitimate and potentially unwanted traffic, and may represent both traffic that was passed to your protected resource and traffic that was blocked by Shield Advanced mitigations.

The mitigation details are for any rules in the web ACL that's associated with the resource, including rules that are deployed specifically in response to an attack and rate-based rules that

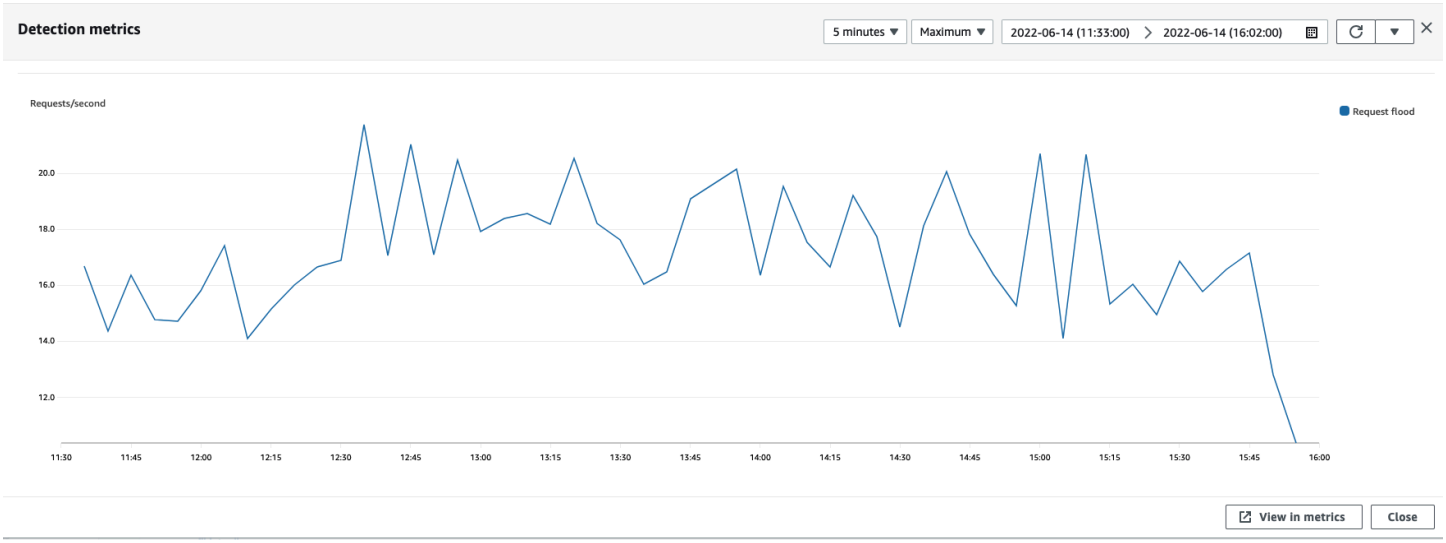
are defined in the web ACL. If you enable automatic application layer DDoS mitigation for an application, the mitigation metrics include metrics for those additional rules. For information about these application layer protections, see Protecting the application layer (layer 7) with AWS Shield Advanced and AWS WAF.

Detection and mitigation

For an application layer (layer 7) event, the **Detection and mitigation** tab shows detection metrics that are based on information obtained from the AWS WAF logs. Mitigation metrics are based on AWS WAF rules in the associated web ACL that are configured to block the unwanted traffic.

For Amazon CloudFront distributions, you can configure Shield Advanced to apply automatic mitigations for you. With any application layer resources, you can choose to define your own mitigating rules in your web ACL and you can request help from the Shield Response Team (SRT). For information about these options, see Responding to DDoS events in AWS.

The following screenshot shows an example of the detection metrics for an application layer event that subsided after a number of hours.



Event traffic that subsides before a mitigating rule takes effect isn't represented in mitigation metrics. This can result in a difference between the web request traffic shown in the detection graphs and the allow and block metrics shown in the mitigation graphs.

Top contributors

The **Top contributors** tab for application layer events displays the top 5 contributors that Shield has identified for the event, based on the AWS WAF logs that it has retrieved. Shield categorizes

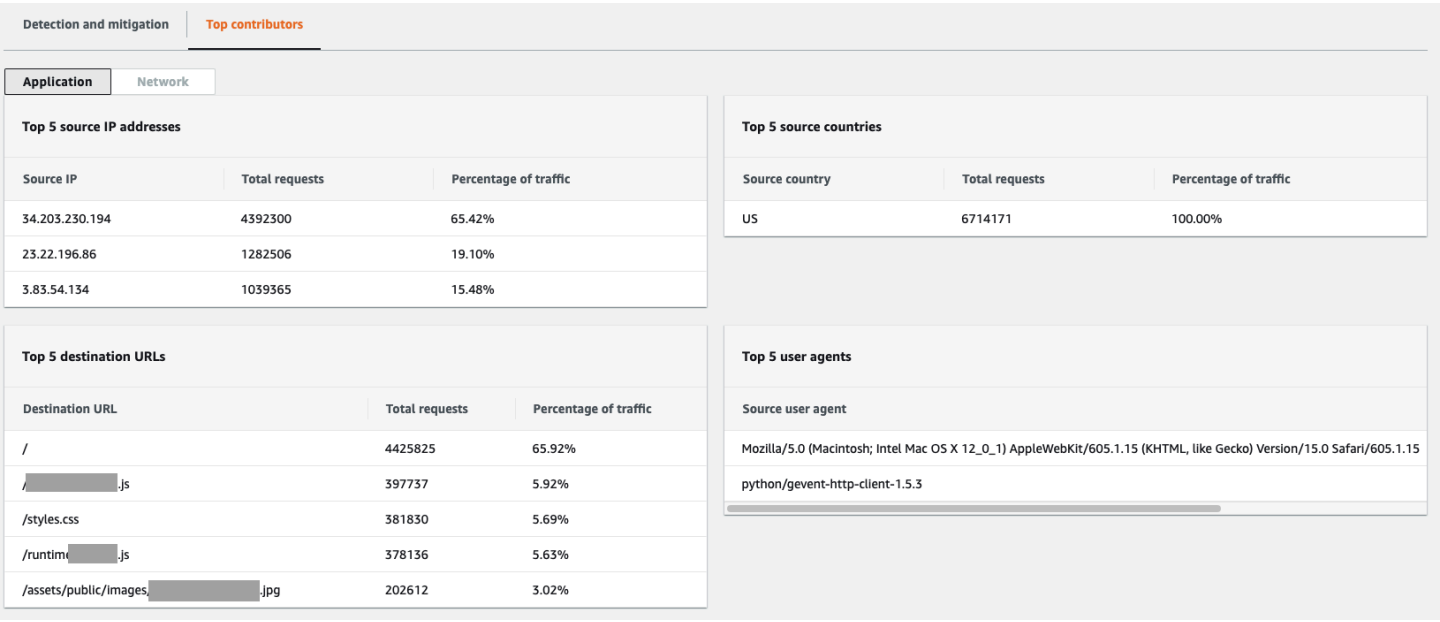
the top contributors information by dimensions such as source IP, source country, and destination URL.

> **Note**
>
> For the most accurate information about the traffic that's contributing to an application layer event, use the AWS WAF logs.

Use the Shield application layer top contributors information only to get a general idea of the nature of an attack, and do not base your security decisions on it. For application layer events, the AWS WAF logs are the best source of information for understanding the contributors to an attack and for devising your mitigation strategies.

The Shield top contributors information doesn't always completely reflect the data in the AWS WAF logs. When it ingests the logs, Shield prioritizes reducing the impact to system performance over retrieving the complete set of data from the logs. This can result in a loss of granularity in the data that's available to Shield for analysis. In most cases, the majority of the information is available, but it's possible for the top contributor data to be skewed to some degree for any attack.

The following screenshot shows an example **Top contributors** tab for an application layer event.



Contributor information is based on requests for both legitimate and potentially unwanted traffic. Larger volume events and events where the request sources aren't highly distributed are more likely to have identifiable top contributors. A significantly distributed attack could have any

number of sources, making it hard to identify top contributors to the attack. If Shield Advanced doesn't identify significant contributors for a specific category, it displays the data as unavailable.

Viewing infrastructure layer (layer 3 or 4) event details in Shield Advanced

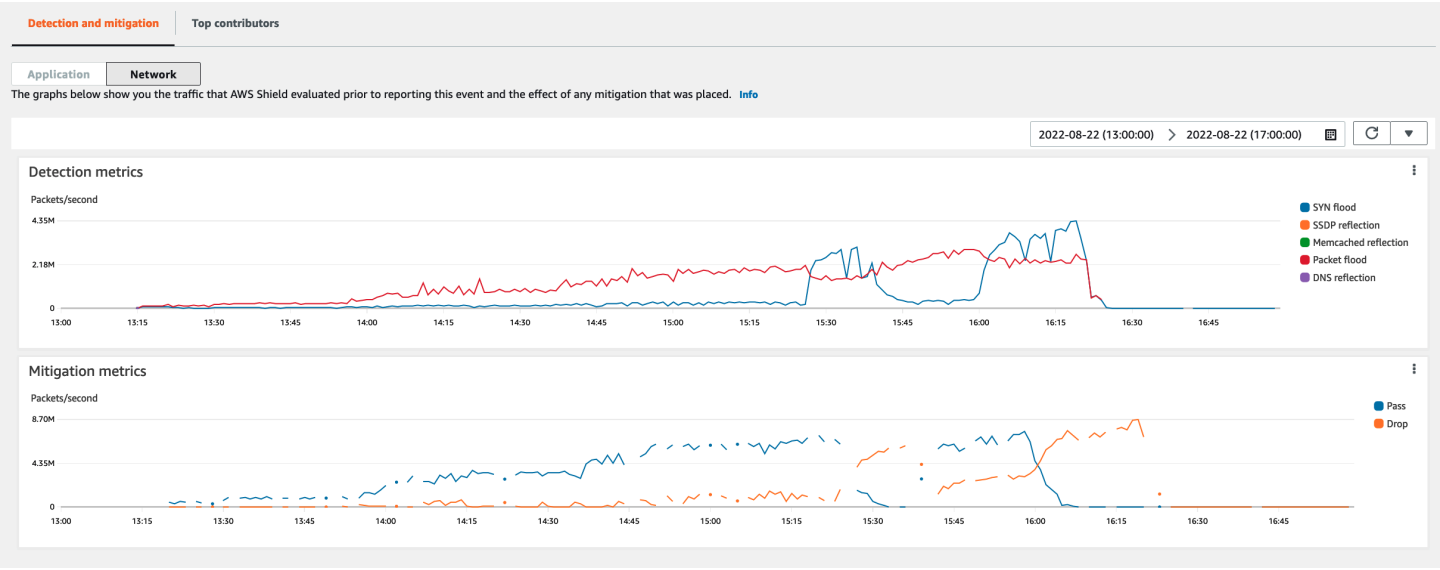
You can see details about an infrastructure layer event's detection, mitigation, and top contributors in the bottom section of the console page for the event. This section can include a mix of legitimate and potentially unwanted traffic, and may represent both traffic that was passed to your protected resource and traffic that was blocked by Shield mitigations.

Detection and mitigation

For an infrastructure layer (layer 3 or 4) event, the **Detection and mitigation** tab shows detection metrics that are based on sampled network flows and mitigation metrics that are based on traffic observed by the mitigation systems. Mitigation metrics are a more precise measurement of the traffic into your resource.

Shield automatically creates a mitigation for the protected resource types Elastic IP (EIP), Classic Load Balancer (CLB), Application Load Balancer (ALB), and AWS Global Accelerator standard accelerator. Mitigation metrics for EIP addresses and AWS Global Accelerator standard accelerators indicate the number of passed and dropped packets.

The following screenshot shows an example **Detection and mitigation** tab for an infrastructure layer event.

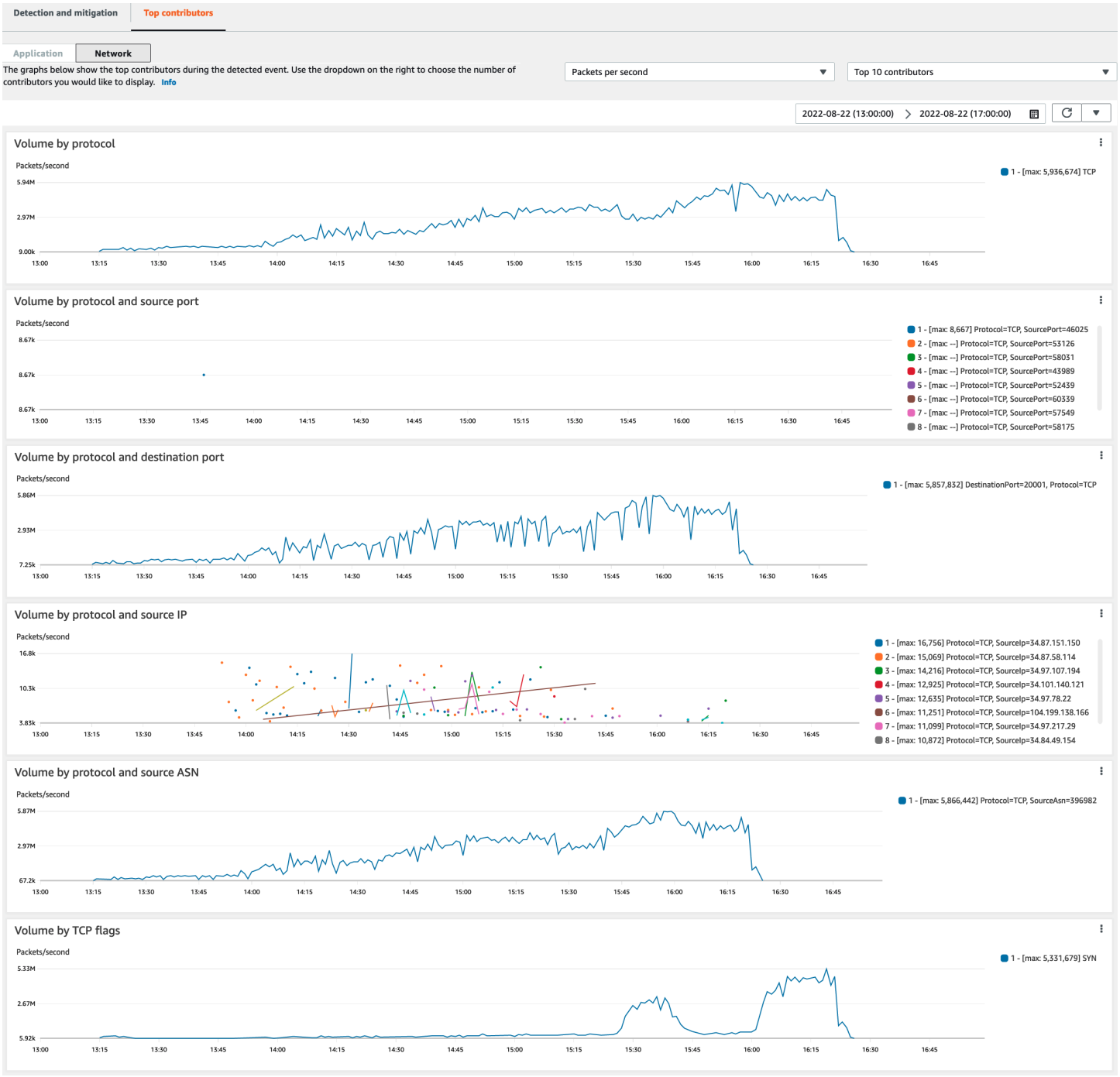


Event traffic that subsides before Shield places a mitigation isn't represented in the mitigation metrics. This can result in a difference between the traffic shown in the detection graphs and the pass and drop metrics shown in the mitigation graphs.

Top contributors

The **Top contributors** tab for infrastructure layer events lists metrics for up to 100 top contributors on several traffic dimensions. The details include network layer properties for any dimension where at least five significant sources of traffic could be identified. Examples of sources of traffic are source IP and source ASN.

The following screenshot shows an example **Top contributors** tab for an infrastructure layer event.



Contributor metrics are based on sampled network flows for both legitimate and potentially unwanted traffic. Larger volume events and events where the traffic sources aren't highly distributed are more likely to have identifiable top contributors. A significantly distributed attack could have any number of sources, making it hard to identify top contributors to the attack. If Shield doesn't identify any significant contributors for a specific metric or category, it displays the data as unavailable.

In an infrastructure layer DDoS attack, traffic sources might be spoofed or reflected. A spoofed source is intentionally forged by the attacker. A reflected source is the real source of detected traffic, but it's not a willing participant in the attack. For example, an attacker might generate a large, amplified flood of traffic to a target by reflecting the attack off of services on the internet that are usually legitimate. In this case, the source information might be valid while it's not the actual source of the attack. These factors can limit the viability of mitigation techniques that block sources based on packet headers.

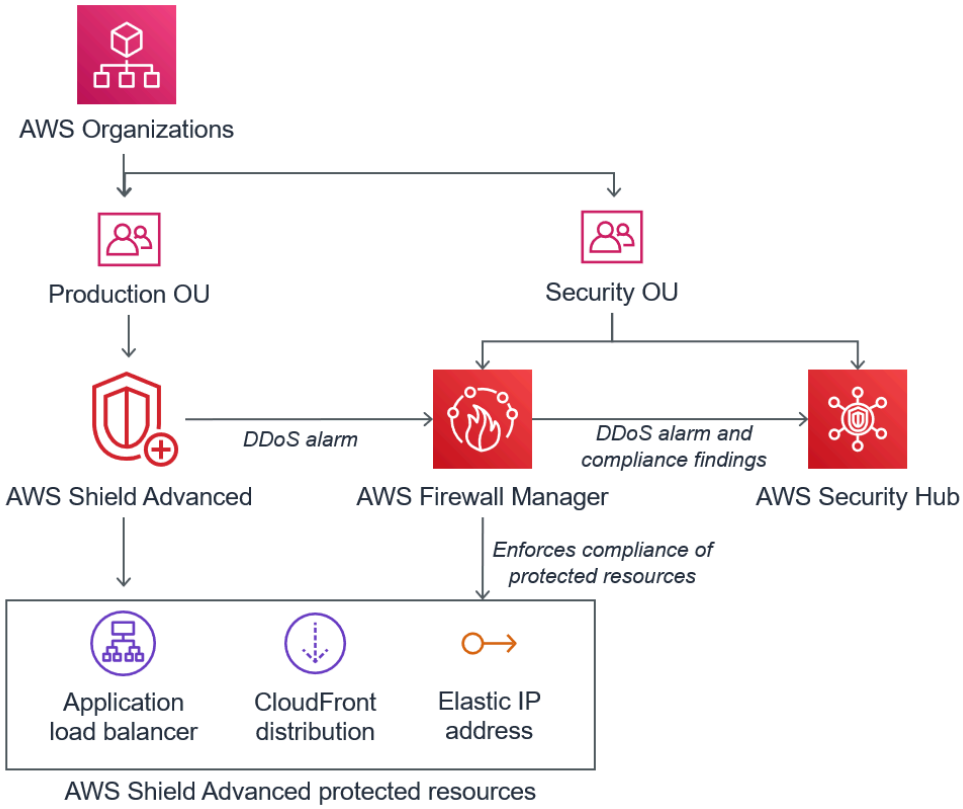
Viewing Shield Advanced events across multiple AWS accounts with AWS Firewall Manager and AWS Security Hub CSPM

You can use AWS Firewall Manager and AWS Security Hub CSPM to manage and monitor AWS Shield Advanced protected resources across multiple accounts.

With Firewall Manager, you can create a Shield Advanced security policy that reports and enforces DDoS protection compliance across all of your accounts. Firewall Manager monitors your protected resources, including adding protections to new resources that come into scope of the Shield Advanced policy.

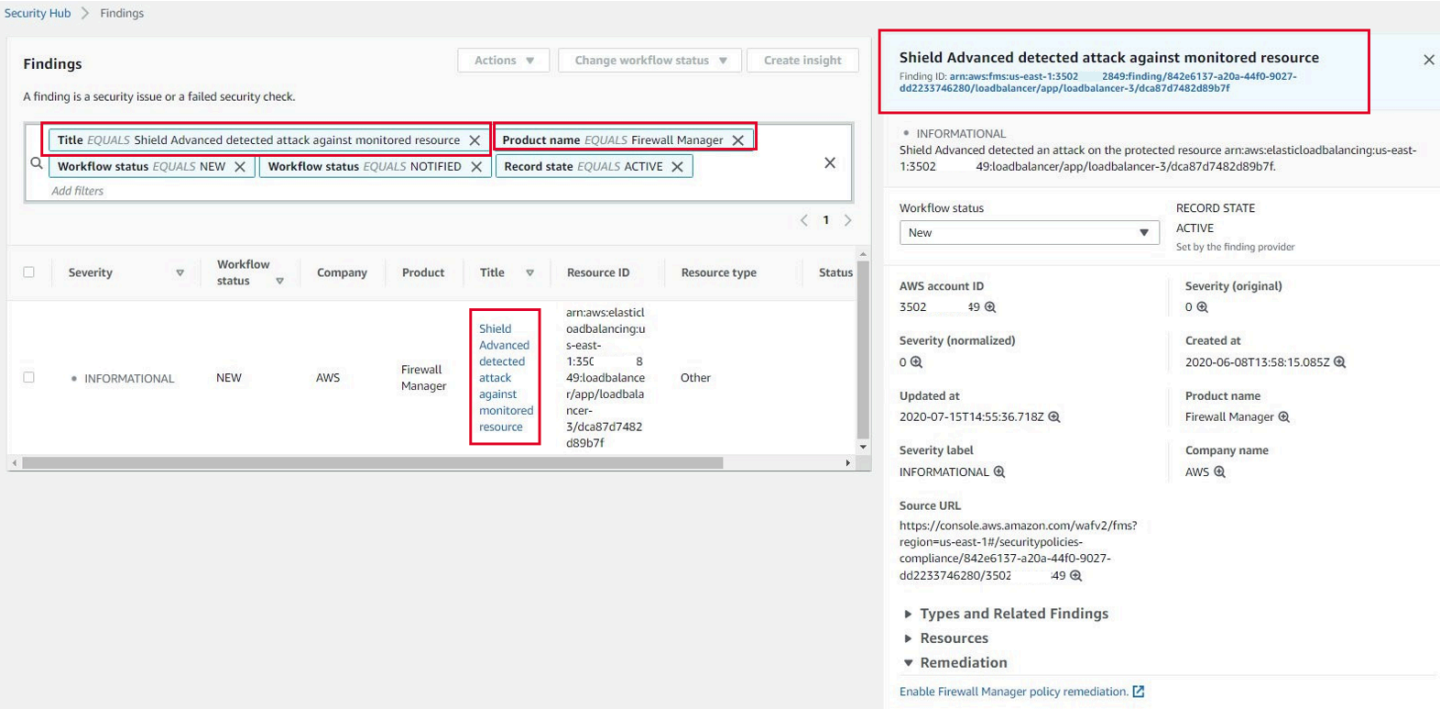
You can integrate Firewall Manager with AWS Security Hub CSPM to get a single dashboard that reports DDoS events that are detected by Shield Advanced and Firewall Manager compliance findings, when Firewall Manager identifies a resource that's out of compliance with your Shield Advanced security policy.

The following figure depicts a typical architecture for monitoring Shield Advanced protected resources with Firewall Manager and Security Hub CSPM.



When you integrate Firewall Manager with Security Hub CSPM, you can view security findings in a single place, alongside other alerts and compliance status information for the applications that you run on AWS.

The following screenshot highlights the information that you can see for a Shield Advanced event inside the Security Hub CSPM console when you have an integration of this type.



To learn how to integrate Firewall Manager and Security Hub CSPM with Shield Advanced to centralize event and compliance monitoring across your protected accounts, see the AWS security blog Set up centralized monitoring for DDoS events and auto-remediate noncompliant resources.

Shield Advanced attack flow logs

Flow logs enable you to capture information about traffic going to network interfaces in your Shield Advanced protected resources. Flow log data is published to Amazon S3, Amazon CloudWatch Logs, or Amazon Data Firehose, where you can retrieve and view your data after you've enabled flow logs.

> **Note**
>
> You must view CloudWatch metrics and logs for Shield Advanced protected resources in the US East (N. Virginia) Region. This applies to both the console and the AWS CLI. When you use the AWS CLI, include the following parameter: `--region us-east-1`

> **Note**
>
> CloudWatch Logs charges apply when you use flow logs, even when logs are published directly to Amazon S3. For more information, see Vended Logs under the Logs tab at Amazon CloudWatch Pricing.

Enable publishing flow logs to Amazon S3

To publish flow logs to Amazon S3, you must configure IAM permissions for the log delivery actions and for the Shield service.

IAM permissions for publishing flow logs

An IAM principal, such as an IAM role or user, must have sufficient permissions to publish flow logs to the Amazon S3 bucket. The IAM policy must include the following permissions:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery",
                "logs:UpdateDeliveryConfiguration"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:accountID:delivery:*",
                "arn:aws:logs:us-east-1:accountID:delivery-source:*",
```

```
                "arn:aws:logs:us-east-1:accountID:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries",
                "logs:DescribeConfigurationTemplates"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyS3",
            "Effect": "Allow",
            "Action": [
                "s3:PutBucketPolicy",
                "s3:GetBucketPolicy"
            ],
            "Resource": "arn:aws:s3:::bucket-name"
        }
    ]
}
```

In the preceding policy, replace *accountID* with your AWS account ID and *bucket-name* with the name of your Amazon S3 bucket.

Shield service-specific permissions

In addition to the destination-specific permissions, AWS Shield requires explicit authorization that you can send logs from your resources. This provides an additional layer of security. Shield authorizes the `AllowVendedLogDeliveryForResource` action for protection resources that vend logs:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ServiceLevelAccessForLogDelivery",
            "Effect": "Allow",
            "Action": [
```

```
                "shield:AllowVendedLogDeliveryForResource"
            ],
            "Resource": "arn:aws:shield::accountID:protection/*"
        }
    ]
}
```

Replace *accountID* with your AWS account ID.

To enable flow log delivery

A working log delivery consists of three elements. Use the following procedure to configure each element using the AWS CLI.

1. Create a `DeliverySource`, which is a logical object that represents the resources that send the logs. Run the following command:

   ```
   aws logs put-delivery-source \
     --name delivery-source-name \
     --resource-arn "arn:aws:shield::accountID:protection/protectionID" \
     --log-type FLOW_LOGS \
     --region us-east-1
   ```

   Replace *delivery-source-name* with a name for your delivery source, *accountID* with your AWS account ID, and *protectionID* with your Shield Advanced protection ID.

   Ensure that the user issuing this command has the service-level permission `shield:AllowVendedLogDeliveryForResource`.

2. Create a `DeliveryDestination`, which is a logical object that represents the actual delivery destination. Run the following command:

   ```
   aws logs put-delivery-destination \
     --name delivery-destination-name \
     --output-format json \
     --delivery-destination-configuration "destinationResourceArn=arn:aws:s3:::bucket-name" \
     --region us-east-1
   ```

   Replace *delivery-destination-name* with a name for your delivery destination and *bucket-name* with the name of your Amazon S3 bucket.

3. Create a `Delivery`, which connects a delivery source to a delivery destination. Run the following command:

```
aws logs create-delivery \
  --delivery-source-name delivery-source-name-from-step1 \
  --delivery-destination-arn "arn-returned-in-step2" \
  --region us-east-1
```

   Replace *`delivery-source-name-from-step1`* with the delivery source name from step 1, and *`arn-returned-in-step2`* with the ARN returned in step 2.

Flow log files

Flow logs from your Shield protection are published to an Amazon S3 bucket at 5-minute intervals during an attack. Log files are written every five minutes, and each log file contains flow log records for the IP address traffic recorded in the previous five minutes.

The maximum file size for a log file is 75 MB. If the log file reaches this limit within the 5-minute period, the flow log stops adding records to it. The flow log then publishes the file to the Amazon S3 bucket and creates a new log file.

Log files are compressed. If you open the files using the Amazon S3 console, Amazon S3 decompresses the log records and displays them. If you download the log files, you must decompress them to view the records.

A single log file contains interleaved entries with multiple records. To see all the log files for a protection, look for entries aggregated by the protection name, Region, and your account ID.

Flow log record syntax

A flow log record is a space-separated string with the following fields.

Field	Description
`version`	Flow log version number.
`protection_arn`	AWS protection ARN that identifies the resource protected in Shield Advanced.

Field	Description
`srcaddr`	Source IP address of the packet.
`dstaddr`	Destination IP address of the packet.
`srcport`	Source port of the packet.
`dstport`	Destination port of the packet.
`protocol`	Protocol of the packet.
`packets`	Number of packets within the aggregation window.
`bytes`	Number of bytes within the aggregation window.
`starttime`	Aggregation window start time.
`endtime`	Aggregation window end time.
`action`	Action taken by Shield Advanced.
`tcp_flags`	TCP flags field from the packet.
`sampling_rate`	Sampling rate used during packet processing.
`location`	AWS ingress location.
`srccountry`	Two-letter country code representing the country of ingress traffic.

Responding to DDoS events in AWS

This page explains how AWS responds to DDoS attacks, and provides options for how you can further respond.

AWS automatically mitigates network and transport layer (layer 3 and layer 4) DDoS attacks. If you use Shield Advanced to protect your Amazon EC2 instances, during an attack Shield Advanced automatically deploys your Amazon VPC network ACLs to the border of the AWS network. This

allows Shield Advanced to provide protection against larger DDoS events. For more information about network ACLs, see Network ACLs.

For application layer (layer 7) DDoS attacks, AWS attempts to detect and notify AWS Shield Advanced customers through CloudWatch alarms. By default, it doesn't automatically apply mitigations, to avoid inadvertently blocking valid user traffic.

For application layer (layer 7) resources, you have the following options available for responding to an attack.

- **Provide your own mitigations** – You can investigate and mitigate the attack on your own. For information, see Manually mitigating an application layer DDoS attack.
- **Contact support** – If you're a Shield Advanced customer, you can contact the AWS Support Center to get help with mitigations. Critical and urgent cases are routed directly to DDoS experts. For information, see Contacting the support center during an application layer DDoS attack.

Additionally, before an attack occurs, you can proactively enable the following mitigation options:

- **Automatic mitigations on Amazon CloudFront distributions** – With this option, Shield Advanced defines and manages mitigating rules for you in your web ACL. For information about automatic application layer mitigation, see Automating application layer DDoS mitigation with Shield Advanced.
- **Proactive engagement** – When AWS Shield Advanced detects a large application layer attack against one of your applications, the SRT can proactively contact you. The SRT triages the DDoS event and creates AWS WAF mitigations. The SRT contacts you and, with your consent, can apply the AWS WAF rules. For more information about this option, see Setting up proactive engagement for the SRT to contact you directly.

Contacting the support center during an application layer DDoS attack

This page provides instructions for contacting the support center during an application layer DDoS attack.

If you're an AWS Shield Advanced customer, you can contact the AWS Support Center to get help with mitigations. Critical and urgent cases are routed directly to DDoS experts. With AWS Shield Advanced, complex cases can be escalated to the AWS Shield Response Team (SRT), which has deep experience in protecting AWS, Amazon.com, and its subsidiaries. For more information about the SRT, see Managed DDoS event response with Shield Response Team (SRT) support.

To get Shield Response Team (SRT) support, contact the AWS Support Center. The response time for your case depends on the severity that you select and the response times, which are documented on the AWS Support Plans page.

Select the following options:

- Case type: Technical Support
- Service: Distributed Denial of Service (DDoS)
- Category: Inbound to AWS
- Severity: *Choose an appropriate option*

When discussing with our representative, explain that you're an AWS Shield Advanced customer experiencing a possible DDoS attack. Our representative will direct your call to the appropriate DDoS experts. If you open a case with the AWS Support Center using the **Distributed Denial of Service (DDoS)** service type, you can speak directly with a DDoS expert by chat or telephone. DDoS support engineers can help you identify attacks, recommend improvements to your AWS architecture, and provide guidance in the use of AWS services for DDoS attack mitigation.

For application layer attacks, the SRT can help you analyze the suspicious activity. If you have automatic mitigation enabled for your resource, the SRT can review the mitigations that Shield Advanced is automatically placing against the attack. In any case, the SRT can assist you to review and mitigate the issue. Mitigations that the SRT recommends often require the SRT to create or update AWS WAF web access control lists (web ACLs) in your account. The SRT will need your permission to do this work.

> **Important**
>
> We recommend that as part of enabling AWS Shield Advanced, you follow the steps in Granting access for the SRT to proactively provide the SRT with the permissions that they need to assist you during an attack. Providing permission ahead of time helps to prevent any delays in the event of an actual attack.

The SRT helps you triage the DDoS attack to identify attack signatures and patterns. With your consent, the SRT creates and deploys AWS WAF rules to mitigate the attack.

You can also contact the SRT before or during a possible attack to review mitigations and to develop and deploy custom mitigations. For example, if you're running a web application and need

only ports 80 and 443 open, you can work with the SRT to preconfigure a web ACL to "allow" only ports 80 and 443.

You authorize and contact the SRT at the account level. That is, if you use Shield Advanced within a Firewall Manager Shield Advanced policy, the account owner, not the Firewall Manager administrator, must contact the SRT for support. The Firewall Manager administrator can contact the SRT only for accounts that they own.

Manually mitigating an application layer DDoS attack

This page provides instructions for manually mitigating an application layer DDoS attack.

If you determine that the activity in the events page for your resource represents a DDoS attack, you can create your own AWS WAF rules in your web ACL to mitigate the attack. This is the only option available if you aren't a Shield Advanced customer. AWS WAF is included with AWS Shield Advanced at no additional cost. For information about creating rules in your web ACL, see Configuring protection in AWS WAF.

If you use AWS Firewall Manager, you can add your AWS WAF rules to a Firewall Manager AWS WAF policy.

To manually mitigate a potential application layer DDoS attack

1. Create rule statements in your web ACL with criteria that matches the unusual behavior. To start with, configure them to count matching requests. For information about configuring your web ACL and rule statements, see Using protection packs (web ACLs) with rules and rule groups in AWS WAF and Testing and tuning your AWS WAF protections.

   > **Note**
   >
   > Always test your rules first by initially using the rule action Count instead of Block. After you're comfortable that your new rules are identifying the correct requests, you can modify them to block the requests.

2. Monitor the request counts to determine whether you want to block the matching requests. If the volume of requests continues to be unusually high and you're confident that your rules are capturing the requests that are causing the high volume, change the rules in your web ACL to block the requests.

3. Continue monitoring the events page to ensure that your traffic is being handled as you want it to be.

AWS provides preconfigured templates to get you started quickly. The templates include a set of AWS WAF rules that you can customize and use to block common web-based attacks. For more information, see AWS WAF Security Automations.

Requesting a credit in AWS Shield Advanced after an attack

If you're subscribed to AWS Shield Advanced and you experience a DDoS attack that increases utilization of a Shield Advanced protected resource, you can request a Shield Advanced service credit for charges related to the increased utilization, to the extent that it is not mitigated by Shield Advanced.

> **Note**
>
> You can apply any credits received through this process only to Shield Advanced usage. Shield Advanced credits are not available for use with other services.

Credits are available only for the following types of charges:

- Shield Advanced data transfer out
- Amazon CloudFront HTTP/HTTPS requests
- CloudFront data transfer out
- Amazon Route 53 queries
- AWS Global Accelerator standard accelerator data transfer
- Load balancer capacity units for Application Load Balancer
- Instance costs for protected Amazon Elastic Compute Cloud (Amazon EC2) instances that were created by an auto-scaling policy in response to the attack

Prerequisites for requesting a credit

To be eligible to receive a credit, before the attack began, you must have done the following:

- You must have added Shield Advanced protection to the resources for which you want to request a credit. Protected resources added during an attack are not eligible for cost protection.

> **Note**
>
> Enabling Shield Advanced on your AWS account does not automatically enable Shield Advanced protection for individual resources.

For more information about how to protect AWS resources using Shield Advanced, see Adding AWS Shield Advanced protection to AWS resources.

- For applicable CloudFront and Application Load Balancer protected resources, you must have associated an AWS WAF web ACL and implemented a rate-based rule in the web ACL in Block mode. For information about AWS WAF rate-based rules, see Using rate-based rule statements in AWS WAF. For information about how to associate web ACLs with AWS resources, see Configuring protection in AWS WAF.
- You must have implemented the appropriate best practices in AWS Best Practices for DDoS Resiliency to configure your application in a way that minimizes cost during a DDoS attack.

How to apply for a credit

To be eligible for a credit, you must submit your credit request within the 15 day period immediately following the billing month in which the attack occurred.

To apply for a credit, submit a billing case through the AWS Support Center. Include the following in your request:

- The words "DDoS Concession" in the subject line
- The dates and times of each event or availability interruption for which you're requesting a credit
- The AWS services and specific resources that were affected

After you submit a request, the AWS Shield Response Team (SRT) will validate whether a DDoS attack occurred and, if so, whether any protected resources scaled to absorb the DDoS attack. If AWS determines that protected resources scaled to absorb the DDoS attack, AWS will issue a credit for that portion of traffic that AWS determines was caused by the DDoS attack. Credits are valid for 12 months.

Security in your use of the AWS Shield service

This section explains how the shared responsibility model applies to AWS Shield.

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

> **Note**
>
> This section provides standard AWS security guidance for your use of the AWS Shield service and its AWS resources, such as Shield Advanced protections.
> For information about protecting your AWS resources using Shield and Shield Advanced, see the rest of the AWS Shield guide.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. The effectiveness of our security is regularly tested and verified by third-party auditors as part of the AWS compliance programs. To learn about the compliance programs that apply to Shield, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your organization's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Shield. The following topics show you how to configure Shield to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Shield resources.

Topics

- Protecting your data in Shield
- Using IAM with AWS Shield
- Logging and monitoring in Shield

- Validating compliance in Shield
- Building for resilience in Shield
- Infrastructure security in AWS Shield

Protecting your data in Shield

This section explains how the AWS shared responsibility model applies to data protection in Shield.

The AWS shared responsibility model applies to data protection in AWS Shield. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see Data Privacy FAQ. For information about data protection in Europe, see the General Data Protection Regulation (GDPR) Center.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Shield or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or

diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Shield entities—such as protections—are encrypted at rest, except in certain Regions where encryption is not available, including China (Beijing) and China (Ningxia). Unique encryption keys are used for each Region.

Using IAM with AWS Shield

This section explains how to use IAM with AWS Shield.

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Shield resources. IAM is an AWS service that you can use with no additional charge.

Topics

- Audience
- Authenticating with identities
- Managing access using policies
- How AWS Shield works with IAM
- Identity-based policy examples for AWS Shield
- AWS managed policies for AWS Shield
- Troubleshooting AWS Shield identity and access
- Using service-linked roles for Shield Advanced

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Shield.

Service user – If you use the Shield service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Shield features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Shield, see Troubleshooting AWS Shield identity and access.

Service administrator – If you're in charge of Shield resources at your company, you probably have full access to Shield. It's your job to determine which Shield features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Shield, see How AWS Shield works with IAM.

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Shield. To view example Shield identity-based policies that you can use in IAM, see Identity-based policy examples for AWS Shield.

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be authenticated as the AWS account root user, an IAM user, or by assuming an IAM role.

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you don't use the root user for everyday tasks. For tasks that require root user credentials, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users to use federation with an identity provider to access AWS services using temporary credentials.

A *federated identity* is a user from your enterprise directory, web identity provider, or Directory Service that accesses AWS services using credentials from an identity source. Federated identities assume roles that provide temporary credentials.

For centralized access management, we recommend AWS IAM Identity Center. For more information, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see Require human users to use federation with an identity provider to access AWS using temporary credentials in the *IAM User Guide*.

An *IAM group* specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity with specific permissions that provides temporary credentials. You can assume a role by switching from a user to an IAM role (console) or by calling an AWS CLI or AWS API operation. For more information, see Methods to assume a role in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you attach to an identity (user, group, or role). These policies control what actions identities can perform, on which resources, and

under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples include IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. You must specify a principal in a resource-based policy.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries** – Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see Service control policies in the *AWS Organizations User Guide*.

- **Resource control policies (RCPs)** – Set the maximum available permissions for resources in your accounts. For more information, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Advanced policies passed as a parameter when creating a temporary session for a role or federated user. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How AWS Shield works with IAM

This section explains how to use the features of IAM with AWS Shield.

Before you use IAM to manage access to Shield, learn what IAM features are available to use with Shield.

IAM features you can use with AWS Shield

IAM feature	Shield support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	Yes
ACLs	No
ABAC (tags in policies)	Partial
Temporary credentials	Yes

IAM feature	Shield support
Forward access sessions (FAS)	Yes
Service roles	Yes
Service-linked roles	Yes

To get a high-level view of how Shield and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for Shield

This section provides identity-based policy examples for AWS Shield.

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

To view examples of Shield identity-based policies, see Identity-based policy examples for AWS Shield.

Resource-based policies within Shield

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for Shield

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Include actions in a policy to grant permissions to perform the associated operation.

To see a list of Shield actions, see Actions defined by AWS Shield in the *Service Authorization Reference*.

Policy actions in Shield use the following prefix before the action:

```
shield
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "shield:action1",
      "shield:action2"
         ]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions in Shield that begin with `List`, include the following action:

```
"Action": "shield:List*"
```

To view examples of Shield identity-based policies, see Identity-based policy examples for AWS Shield.

Policy resources for Shield

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. As a best practice, specify a resource using its Amazon Resource Name (ARN). For actions that don't support resource-level permissions, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see the list of Shield resource types and their ARNs, see Resources defined by AWS Shield in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by AWS Shield. To allow or deny access to a subset of Shield resources, include the ARN of the resource in the `resource` element of your policy.

In AWS Shield, the resources are *protections* and *attacks*. These resources have unique Amazon Resource Names (ARNs) associated with them, as shown in the following table.

Name in AWS Shield Console	Name in AWS Shield SDK/ CLI	ARN Format
Event or attack	`AttackDetail`	`arn:aws:shield:: account:attack/ID`
Protection	`Protection`	`arn:aws:shield:: account:protection/ ID`

To allow or deny access to a subset of Shield resources, include the ARN of the resource in the `resource` element of your policy. The ARNs for Shield have the following format:

```
arn:partition:shield::account:resource/ID
```

Replace the *`account`*, *`resource`*, and *`ID`* variables with valid values. Valid values can be the following:

- *`account`*: The ID of your AWS account. You must specify a value.

- *resource*: The type of Shield resource, either `attack` or `protection`.
- *ID*: The ID of the Shield resource, or a wildcard (*) to indicate all resources of the specified type that are associated with the specified AWS account.

For example, the following ARN specifies all protections for the account 111122223333:

```
arn:aws:shield::111122223333:protection/*
```

The ARNs of Shield resources have the following format:

```
arn:partition:shield:region:account-id:scope/resource-type/resource-name/resource-id
```

For general information about ARN specifications, see Amazon Resource Names (ARNs) in the Amazon Web Services General Reference.

The following lists requirements that are specific to the ARNs of `wafv2` resources:

- *region*: For Shield resources that you use to protect Amazon CloudFront distributions, set this to `us-east-1`. Otherwise, set this to the Region you're using with your protected regional resources.
- *scope*: Set the scope to `global` for use with an Amazon CloudFront distribution or `regional` for use with any of the regional resources that AWS WAF supports. The regional resources are an Amazon API Gateway REST API, an Application Load Balancer, an AWS AppSync GraphQL API, an Amazon Cognito user pool, an AWS App Runner service, an Amazon Bedrock AgentCore Gateway, and an AWS Verified Access instance.
- *resource-type*: Specify one of the following values: `attack` for events or attacks, `protection` for protections.
- *resource-name*: Specify the name that you gave the Shield resource, or specify a wildcard (*) to indicate all resources that satisfy the other specifications in the ARN. You must either specify the resource name and resource ID or specify a wildcard for both.
- *resource-id*: Specify the ID of the Shield resource, or specify a wildcard (*) to indicate all resources that satisfy the other specifications in the ARN. You must either specify the resource name and resource ID or specify a wildcard for both.

For example, the following ARN specifies all web ACLs with regional scope for the account `111122223333` in Region `us-west-1`:

```
arn:aws:wafv2:us-west-1:111122223333:regional/webacl/*/*
```

The following ARN specifies the rule group named `MyIPManagementRuleGroup` with global scope for the account `111122223333` in Region `us-east-1`:

```
arn:aws:wafv2:us-east-1:111122223333:global/rulegroup/MyIPManagementRuleGroup/1111aaaa-bbbb-cccc-dddd-example-id
```

To view examples of Shield identity-based policies, see Identity-based policy examples for AWS Shield.

Policy condition keys for Shield

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element specifies when statements execute based on defined criteria. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of Shield condition keys, see Condition keys for AWS Shield in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by AWS Shield.

To view examples of Shield identity-based policies, see Identity-based policy examples for AWS Shield.

ACLs in Shield

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ABAC with Shield

Supports ABAC (tags in policies): Partial

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes called tags. You can attach tags to IAM entities and AWS resources, then design ABAC policies to allow operations when the principal's tag matches the tag on the resource.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see Define permissions with ABAC authorization in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with Shield

Supports temporary credentials: Yes

Temporary credentials provide short-term access to AWS resources and are automatically created when you use federation or switch roles. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM and AWS services that work with IAM in the *IAM User Guide*.

Forward access sessions for Shield

Supports forward access sessions (FAS): Yes

Forward access sessions (FAS) use the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. For policy details when making FAS requests, see Forward access sessions.

Service roles for Shield

Supports service roles: Yes

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.

> **Warning**
>
> Changing the permissions for a service role might break Shield functionality. Edit service roles only when Shield provides guidance to do so.

Service-linked roles for Shield

Supports service-linked roles: Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing Shield service-linked roles, see Using service-linked roles for Shield Advanced.

Identity-based policy examples for AWS Shield

By default, users and roles don't have permission to create or modify Shield resources. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Create IAM policies (console) in the *IAM User Guide*.

For details about actions and resource types defined by Shield, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for AWS Shield in the *Service Authorization Reference*.

Topics

- Policy best practices
- Using the Shield console
- Allow users to view their own permissions
- Grant read access to your Shield Advanced protections
- Grant read-only access to Shield, CloudFront, and CloudWatch
- Grant full access to Shield, CloudFront, and CloudWatch

Policy best practices

Identity-based policies determine whether someone can create, access, or delete Shield resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the Shield console

To access the AWS Shield console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Shield resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

Users who can access and use the AWS console can also access the AWS Shield console. No additional permissions are required.

Console-only APIs

You can access the following Distributed Denial of Service (DDoS) attack information in the console. Specify the following API permissions in an IAM policy to allow or deny specific actions.

Action	Description
`DescribeAttackContributors`	Grants permission to get detailed informati on about the contributors to a specific DDoS attack.
`ListMitigations`	Grants permission to retrieve a list of mitigatio n actions that have been applied during DDoS attacks.
`GetGlobalThreatData`	Grants permission to retrieve global threat intelligence data and trends from AWS Shield's threat monitoring systems.

This example shows how you might create a policy that allows you to see DDoS attack information in the console.

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "shield:DescribeAttackContributors"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "shield:ListMitigations"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "shield:GetGlobalThreatData"
            ],
            "Resource": "*"
        }
    ]
}
```

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
```

```
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws-eusc:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Grant read access to your Shield Advanced protections

AWS Shield allows cross-account resource access, but it doesn't allow you to create cross-account resource protections. You can only create protections for resources from within the account that owns those resources.

The following is an example policy that grants permissions for the `shield:ListProtections` action on all resources. Shield doesn't support identifying specific resources using the resource ARNs (also referred to as resource-level permissions) for some of the API actions, so you specify a wildcard character (*). This only permits access to the resources that you can retrieve through the action `ListProtections`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ListProtections",
```

```
            "Effect": "Allow",
            "Action": [
                "shield:ListProtections"
            ],
            "Resource": "*"
        }
    ]
}
```

Grant read-only access to Shield, CloudFront, and CloudWatch

The following policy grants users read-only access to Shield and associated resources, including Amazon CloudFront resources, and Amazon CloudWatch metrics. It's useful for users who need permission to view the settings in Shield protections and attacks and to monitor metrics in CloudWatch. These users can't create, update, or delete Shield resources.

JSON

```
{
        "Version":"2012-10-17",
        "Statement": [
            {
                "Sid": "ProtectedResourcesReadAccess",
                "Effect": "Allow",
                "Action": [
                    "cloudfront:List*",
                    "route53:List*",
                    "cloudfront:Describe*",
                    "elasticloadbalancing:Describe*",
                    "cloudwatch:Describe*",
                    "cloudwatch:Get*",
                    "cloudwatch:List*",
                    "cloudfront:GetDistribution*",
                    "globalaccelerator:ListAccelerators",
                    "globalaccelerator:DescribeAccelerator"
                ],
                "Resource": [
                    "arn:aws-eusc:elasticloadbalancing:*:*:*",
                    "arn:aws-eusc:cloudfront::*:*",
                    "arn:aws-eusc:route53:::hostedzone/*",
                    "arn:aws-eusc:cloudwatch:*:*:*:*",
```

```
                "arn:aws-eusc:globalaccelerator::*:*"
            ]
        },
        {
            "Sid": "ShieldReadOnly",
            "Effect": "Allow",
            "Action": [
                "shield:List*",
                "shield:Describe*",
                "shield:Get*"
            ],
            "Resource": "*"
        }
    ]
}
```

Grant full access to Shield, CloudFront, and CloudWatch

The following policy lets users perform any Shield operation, perform any operation on CloudFront web distributions, and monitor metrics and a sample of requests in CloudWatch. It's useful for users who are Shield administrators.

JSON

```
{
        "Version":"2012-10-17",
        "Statement": [
            {
                "Sid": "ProtectedResourcesReadAccess",
                "Effect": "Allow",
                "Action": [
                    "cloudfront:List*",
                    "route53:List*",
                    "cloudfront:Describe*",
                    "elasticloadbalancing:Describe*",
                    "cloudwatch:Describe*",
                    "cloudwatch:Get*",
                    "cloudwatch:List*",
                    "cloudfront:GetDistribution*",
                    "globalaccelerator:ListAccelerators",
                    "globalaccelerator:DescribeAccelerator"
```

```
                ],
                "Resource": [
                    "arn:aws-eusc:elasticloadbalancing:*:*:*",
                    "arn:aws-eusc:cloudfront::*:*",
                    "arn:aws-eusc:route53:::hostedzone/*",
                    "arn:aws-eusc:cloudwatch:*:*:*:*",
                    "arn:aws-eusc:globalaccelerator::*:*"
                ]
            },
            {
                "Sid": "ShieldFullAccess",
                "Effect": "Allow",
                "Action": [
                    "shield:*"
                ],
                "Resource": "*"
            }
        ]
}
```

We strongly recommend that you configure multi-factor authentication (MFA) for users who have administrative permissions. For more information, see Using Multi-Factor Authentication (MFA) Devices with AWS in the *IAM User Guide*.

AWS managed policies for AWS Shield

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed

policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

AWS managed policy: AWSShieldDRTAccessPolicy

This section explains how to use AWS managed policies for Shield.

AWS Shield uses this managed policy when you grant permission to the Shield Response Team (SRT) to act on your behalf. This policy gives the SRT limited access to your AWS account, to assist with DDoS attack mitigation during high-severity events. This policy allows the SRT to manage your AWS WAF rules and Shield Advanced protections and to access your AWS WAF logs.

For information about granting permission to the SRT to operate on your behalf, see Granting access for the SRT.

For details about this policy, see AWSShieldDRTAccessPolicy in the IAM console.

AWS managed policy: AWSShieldServiceRolePolicy

Shield Advanced uses this managed policy when you enable automatic application layer DDoS mitigation, to set the permissions it needs to manage resources for your account. This policy allows Shield Advanced to create and apply AWS WAF rules and rule groups in the web ACLs that you've associated with your protected resources, to automatically respond to DDoS attacks.

You can't attach AWSShieldServiceRolePolicy to your IAM entities. Shield attaches this policy to the service-linked role `AWSServiceRoleForAWSShield` to allow Shield to perform actions on your behalf.

Shield Advanced enables the use of this policy when you enable automatic application layer DDoS mitigation. For more information about the use for this policy, see Automating application layer DDoS mitigation with Shield Advanced.

For information about the service-linked role AWSServiceRoleForAWSShield that uses this policy, see Using service-linked roles for Shield Advanced

For details about this policy, see AWSShieldServiceRolePolicy in the IAM console.

Shield updates to AWS managed policies

View details about updates to AWS managed policies for Shield since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Shield document history page at Document history.

Policy	Description of change	Date
`AWSShieldServiceRolePolicy` This policy allows Shield to access and manage AWS resources in order to automatically respond to application layer DDoS attacks on your behalf. Details in IAM console: AWSShieldServiceRolePolicy The service-linked role `AWSServiceRoleForAWSShield` uses this policy. For information, see Using service-linked roles for Shield Advanced.	Added this policy to provide Shield Advanced with the permissions required for the automatic application layer DDoS mitigation functionality. For information about this feature, see Automating application layer DDoS mitigation with Shield Advanced.	December 1, 2021
Shield started tracking changes	Shield started tracking changes for its AWS managed policies.	March 3, 2021

Troubleshooting AWS Shield identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Shield and IAM.

Topics

- I am not authorized to perform an action in Shield

- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my Shield resources

I am not authorized to perform an action in Shield

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `shield:`*`GetWidget`* permissions.

```
User: arn:aws-eusc:iam::123456789012:user/mateojackson is not authorized to perform:
 shield:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `shield:`*`GetWidget`* action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to Shield.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Shield. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws-eusc:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my Shield resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Shield supports these features, see How AWS Shield works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.

Using service-linked roles for Shield Advanced

This section explains how to use service-linked roles to give Shield Advanced access to resources in your AWS account.

AWS Shield Advanced uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to Shield Advanced. Service-linked roles are predefined by Shield Advanced and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up Shield Advanced easier because you don't have to manually add the necessary permissions. Shield Advanced defines the permissions of its service-linked roles, and unless defined otherwise, only Shield Advanced can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting their related resources. This protects your Shield Advanced resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-Linked Role Permissions for Shield Advanced

Shield Advanced uses the service-linked role named **AWSServiceRoleForAWSShield**. This role allows Shield Advanced to access and manage AWS resources in order to automatically respond to application layer DDoS attacks on your behalf. For more information about this functionality, see Automating application layer DDoS mitigation with Shield Advanced.

The AWSServiceRoleForAWSShield service-linked role trusts the following services to assume the role:

- `shield.amazonaws.com`

The role permissions policy named AWSShieldServiceRolePolicy allows Shield Advanced to complete the following actions on all AWS resources:

- `wafv2:GetWebACL`
- `wafv2:UpdateWebACL`
- `wafv2:GetWebACLForResource`
- `wafv2:ListResourcesForWebACL`
- `cloudfront:ListDistributions`
- `cloudfront:GetDistribution`

When actions are permitted on all AWS resources, this is indicated in the policy as `"Resource": "*"`. This only means that the service-linked role can take each indicated action on all AWS resources *that the action supports*. For example, the action `wafv2:GetWebACL` is supported only for `wafv2` web ACL resources.

Shield Advanced only makes resource-level API calls for protected resources for which you've enabled the application layer protections feature and for web ACLs that are associated with those protected resources.

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

Creating a Service-Linked Role for Shield Advanced

You don't need to manually create a service-linked role. When you enable automatic application layer DDoS mitigation for a resource in the AWS Management Console, the AWS CLI, or the AWS API, Shield Advanced creates the service-linked role for you.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you enable automatic application layer DDoS mitigation for a resource, Shield Advanced creates the service-linked role for you again.

Editing a Service-Linked Role for Shield Advanced

Shield Advanced does not allow you to edit the AWSServiceRoleForAWSShield service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting a Service-Linked Role for Shield Advanced

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

> **Note**
>
> If Shield Advanced is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To delete the Shield Advanced resources that are used by the AWSServiceRoleForAWSShield

For all of your resources that have application layer DDoS protections configured, disable automatic application layer DDoS mitigation. For console instructions, see Configure application layer DDoS protections.

To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the AWSServiceRoleForAWSShield service-linked role. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

Supported Regions for Shield Advanced Service-Linked Roles

Shield Advanced supports using service-linked roles in all of the Regions where the service is available. For more information, see Shield Advanced endpoints and quotas.

Logging and monitoring in Shield

This section explains how to use AWS tools for monitoring and responding to events in AWS Shield.

Monitoring is an important part of maintaining the reliability, availability, and performance of Shield and your AWS solutions. You should collect monitoring data from all parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your Shield resources and responding to potential events:

Amazon CloudWatch Alarms

Using CloudWatch alarms, you watch a single metric over a time period that you specify. If the metric exceeds a given threshold, CloudWatch sends a notification to an Amazon SNS topic or AWS Auto Scaling policy. For more information, see Monitoring with Amazon CloudWatch.

AWS CloudTrail Logs

CloudTrail provides a record of actions taken by a user, role, or an AWS service in Shield. Using the information collected by CloudTrail, you can determine the request that was made to Shield, the IP address from which the request was made, who made the request, when it was made, and additional details. For more information, see Logging API calls with AWS CloudTrail.

Validating compliance in Shield

This section explains your compliance responsibility when using AWS Shield.

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. For more information about your compliance responsibility when using AWS services, see AWS Security Documentation.

Building for resilience in Shield

This section explains how AWS architecture supports data redundancy for AWS Shield.

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure security in AWS Shield

This section explains how AWS Shield isolates service traffic.

As a managed service, AWS Shield is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access Shield through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

AWS Shield Advanced quotas

AWS Shield Advanced has default quotas on the number of entities per Region. You can request an increase in these quotas.

Resource	Default quota
Maximum number of protected resources for each resource type that AWS Shield Advanced offers protection for, per account.	1,000
Maximum number of protection groups, per account.	100
Maximum number of individual protected resources that you can specifically include in a protection group. In the API, this applies to the `Members` that you specify when you set the protection group `Pattern` to `ARBITRARY`. In the console, this applies to the resources that you select for the protection grouping **Choose from protected resources**.	1,000

AWS Shield network security director (preview)

> **Note**
>
> AWS Shield network security director is in public preview release and is subject to change.

AWS Shield network security director helps secure your AWS environment by discovering your compute, networking, and network security resources across your account. network security director evaluates each resource's security configuration by analyzing network topology and security configurations against AWS best practices and threat intelligence. To help you strengthen your security, network security director rates its findings from low to critical severity and shares specific remediation steps, which you can explore using natural language queries through Amazon Q Developer.

AWS Shield network security director pricing

AWS currently does not charge for use of network security director. However, you are responsible for fees incurred for the underlying services you use, such as AWS WAF. When network security director becomes generally available, pricing will differ from the preview release.

AWS Shield network security director use cases

> **Note**
>
> AWS Shield network security director is in public preview release and is subject to change.

AWS Shield network security director helps secure your AWS environment by discovering your compute, networking, and network security resources across your account. network security director evaluates each resource's security configuration by analyzing network topology and security configurations against AWS best practices and threat intelligence. To help you strengthen your security, network security director rates its findings from low to critical severity and shares specific remediation steps, which you can explore using natural language queries through Amazon Q Developer.

Network security director and Amazon Q Developer help you identify security issues in your network security configuration and provide mitigation options:

- **Overly permissive access to your EC2 instances** – Identify security groups and network ACLs (NACLs) that are associated with your VPCs and Amazon Elastic Compute Cloud instances that allow unrestricted access to high-risk ports, such as ports 22 and 3389. Follow step-by-step instructions for implementing the right rules for security groups or NACLs to restrict access for these high-risk ports.
- **Compute and networking resources that are open to the internet** – Identify resources that are reachable from the internet via connectivity with an internet gateway.
- **Internet-facing resources that aren't fully protected by AWS WAF** – Identify resources that are reachable from the internet and understand the status of their AWS WAF protections. Follow step-by-step instructions for configuring and deploying AWS WAF, including recommendations for using rules such as rate-limiting rules and AWS Managed Rules rule groups.
- **Resources that are exposed to known threats** – Identify resources that are exposed to known threats, such as distributed denial of service (DDoS) attacks, SQL injection attacks, and cross-site scripting (XSS) attacks. Follow step-by-step instructions for implementing custom rules or AWS WAF AWS Managed Rules rule groups to defend against these threats.
- **Network security services that are enabled but aren't attached to any compute or networking resources** – Identify AWS WAF web ACLs and VPC security groups and NACLs that are currently not protecting any of your compute or networking resources. Follow instructions for removing them or for adding recommended rules to improve protections in case you decide to associate them with compute or networking resources in the future.

Key concepts in network security director

> **Note**
>
> AWS Shield network security director is in public preview release and is subject to change.

Resources

The compute, networking, and security resources that handle your application traffic:

- *Compute* – Amazon Elastic Compute Cloud instances

- *Networking* – Application Load Balancers, Amazon API Gateways, Amazon CloudFront distributions, VPC subnets, and VPC elastic network interfaces (ENIs)
- *Security* – AWS WAF web ACLs, VPC security groups, and VPC network access control lists (NACLs)

Findings

Alerts about missing or misconfigured network security services, with severity levels of NONE, INFORMATIONAL, LOW, MEDIUM, HIGH, or CRITICAL. network security director generates findings by evaluating configuration settings and threat intelligence for each resource.

Severity

A measure of a resource's vulnerability to potential security events, based on AWS best practices and threat intelligence. Severity assessment considers both potential vulnerabilities and existing protections. A resource's severity level matches its most severe finding, or shows as none if there are no findings.

Network topology

A visual representation of your network that shows resource connections, internet exposure, and tag-based relationships. Use the topology view to investigate resources and their findings.

Understanding network security director findings

> **Note**
>
> AWS Shield network security director is in public preview release and is subject to change.

Network security director generates specific findings for each type of resource it analyzes. These findings help you identify security issues and take appropriate action. The following table lists all possible findings by resource type.

network security director findings by resource type

Resource type	Finding description
Application Load Balancer	• CloudFront origin is also Internet accessible without CloudFront protections • AWS WAF missing bot and scraper rules

Resource type	Finding description
	• DDoS activity detected • Resource has no firewall attached for protection • AWS WAF missing all rules - no protection, possible misconfiguration • AWS WAF missing key AWS Managed Rules (IP Reputation, Common Rules, or Bad Inputs)
Amazon API Gateway	• AWS WAF missing bot and scraper rules • Resource has no firewall attached for protection • AWS WAF missing all rules - no protection, possible misconfiguration • AWS WAF missing key AWS Managed Rules (IP Reputation, Common Rules, or Bad Inputs)
Amazon CloudFront	• AWS WAF missing bot and scraper rules • DDoS activity detected • Resource has no firewall attached for protection • AWS WAF missing all rules - no protection, possible misconfiguration • AWS WAF missing key AWS Managed Rules (IP Reputation, Common Rules, or Bad Inputs)

Resource type	Finding description
Amazon Elastic Compute Cloud (EC2) instance	• Allows unrestricted inbound access (0.0.0.0/0) on all ports • Allows unrestricted inbound access (0.0.0.0/0) to RDP port 3389 • Allows unrestricted inbound access (0.0.0.0/0) to SSH port 22 • Allows unrestricted outbound access (0.0.0.0/0) on all ports • Resource has no firewall attached for protection • CloudFront origin is also Internet accessible without CloudFront protections • Resource has no firewall attached for protection • AWS WAF missing bot and scraper rules • AWS WAF missing all rules - no protection, possible misconfiguration • AWS WAF missing key AWS Managed Rules (IP Reputation, Common Rules, or Bad Inputs)
VPC security group	• Allows unrestricted inbound access (0.0.0.0/0) on all ports • Allows unrestricted inbound access (0.0.0.0/0) to RDP port 3389 • Allows unrestricted inbound access (0.0.0.0/0) to SSH port 22 • Allows unrestricted outbound access (0.0.0.0/0) on all ports
VPC network access control list (NACL)	• Allows unrestricted inbound access (0.0.0.0/0) on all ports • Allows unrestricted inbound access (0.0.0.0/0) to RDP port 3389 • Allows unrestricted inbound access (0.0.0.0/0) to SSH port 22 • Allows unrestricted outbound access (0.0.0.0/0) on all ports

Resource type	Finding description
AWS WAF web ACL	• Bot activity detected • AWS WAF missing bot and scraper rules • AWS WAF WebACL is not associated with any resources • AWS WAF missing all rules - no protection, possible misconfiguration • AWS WAF missing key AWS Managed Rules (IP Reputation, Common Rules, or Bad Inputs)

Setting up your account to use AWS Shield network security director

> **Note**
>
> AWS Shield network security director is in public preview release and is subject to change.

AWS Shield network security director requires AWS Organizations to manage security across multiple accounts in your organization. This topic describes the preliminary steps to prepare your AWS environment, including setting up Organizations, designating a delegated administrator, and configuring the necessary IAM permissions. You aren't charged for these preliminary setup steps. You are charged only for AWS services that you use.

Prerequisites

Before you can use AWS Shield network security director, you must have the following in place:

- **AWS Organizations** - AWS Shield network security director works exclusively with AWS Organizations to provide security analysis across multiple accounts. You cannot use AWS Shield network security director with a single standalone account.
- **Management account access** - You need access to the AWS Organizations management account to designate a delegated administrator for AWS Shield network security director.

- **Delegated administrator account** - You need to identify or create an account that will serve as the delegated administrator for AWS Shield network security director. This cannot be the Organizations management account.

> **Important**
>
> AWS Shield network security director cannot be used with standalone AWS accounts. You must have AWS Organizations configured with at least one member account in addition to the management account.

Understanding AWS Organizations integration

AWS Organizations is a global account management service that lets AWS administrators consolidate and manage multiple AWS accounts. AWS Shield network security director integrates with Organizations to provide centralized security analysis and management across your entire organization.

When you integrate AWS Shield network security director with AWS Organizations:

- The Organizations management account designates a delegated administrator for AWS Shield network security director
- The delegated administrator can enable AWS Shield network security director across multiple accounts and regions
- Security analysis and findings are centrally managed through the delegated administrator account
- Service-linked roles are automatically created in member accounts to enable analysis

This approach is similar to other AWS security services like AWS Security Hub and provides consistent governance across your security tools.

Choosing a delegated administrator

A delegated administrator is an AWS account in your organization that has been granted permissions to manage AWS Shield network security director on behalf of the organization. The delegated administrator can enable the service, create policies, and manage security findings across all member accounts.

Delegated administrator requirements:

- Must be a member account in your AWS Organizations structure
- Cannot be the Organizations management account
- Should have appropriate IAM permissions configured (see next section)

> **Note**
>
> As a best practice, we recommend using the same delegated administrator account across AWS security services (such as Security Hub, GuardDuty, and AWS Shield network security director) for consistent governance and simplified management.

IAM requirements for the delegated administrator

The delegated administrator account requires specific IAM permissions to manage AWS Shield network security director effectively. You must attach the following policy to the IAM user or role that will be managing AWS Shield network security director in the delegated administrator account.

Required IAM policy for AWS Shield network security director delegated administrator:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "network-security-director:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:CreateServiceLinkedRole"
            ],
```

```
            "Resource": [
                "arn:aws:iam:::role/aws-service-role/
AWSServiceRoleForNetworkSecurityDirector"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "organizations:ListRoots",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:ListAccountsForParent",
                "organizations:ListAccounts",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:ListDelegatedAdministrators",
                "organizations:DescribeOrganization",
                "organizations:CreatePolicy",
                "organizations:UpdatePolicy",
                "organizations:DeletePolicy",
                "organizations:AttachPolicy",
                "organizations:DetachPolicy",
                "organizations:EnablePolicyType",
                "organizations:DisablePolicyType",
                "organizations:DescribeOrganization",
                "organizations:DescribeOrganizationalUnit",
                "organizations:DescribeAccount",
                "organizations:ListRoots",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:ListParents",
                "organizations:ListChildren",
                "organizations:ListAccounts",
                "organizations:ListAccountsForParent",
                "organizations:ListTagsForResource",
                "organizations:ListDelegatedAdministrators",
                "organizations:ListHandshakesForAccount",
                "organizations:DescribePolicy",
                "organizations:DescribeEffectivePolicy",
                "organizations:ListPolicies",
                "organizations:ListPoliciesForTarget",
                "organizations:ListTargetsForPolicy"
            ],
            "Resource": [
                "*"
            ]
        }
```

```
    ]
}
```

Policy explanation:

- **network-security-director:*** - Grants full access to all AWS Shield network security director operations, including enabling the service, creating policies, and managing findings.
- **IAM permissions** - Allows the delegated administrator to manage the service-linked role that AWS Shield network security director uses to perform analysis across member accounts.

To create and attach the IAM policy

1. Sign in to the AWS Management Console using the delegated administrator account.
2. Open the IAM console at https://console.aws.amazon.com/iam/.
3. In the navigation pane, choose **Policies**, then choose **Create policy**.
4. Choose the **JSON** tab and paste the policy document shown above.
5. Choose **Next: Tags**, then **Next: Review**.
6. For **Name**, enter **`NetworkSecurityDirectorDelegatedAdminPolicy`**.
7. Choose **Create policy**.
8. Attach this policy to the IAM user or role that will be managing AWS Shield network security director in the delegated administrator account.

Setup checklist

Before proceeding to enable AWS Shield network security director, ensure you have completed the following setup tasks:

- ✓ AWS Organizations is configured with a management account and at least one member account
- ✓ You have identified a delegated administrator account (cannot be the management account)
- ✓ The required IAM policy has been created and attached in the delegated administrator account
- ✓ You have access to both the Organizations management account and the delegated administrator account

Once you have completed these setup tasks, you can proceed to Enabling AWS Shield network security director to enable AWS Shield network security director for your organization.

Enabling AWS Shield network security director

> **Note**
>
> AWS Shield network security director is in public preview release and is subject to change.

AWS Shield network security director is enabled for AWS accounts through AWS Organizations. This section of the documentation describes all the steps required to enable AWS Shield network security director for an AWS Organization.

This section includes two steps, both of which are necessary to complete network security director setup:

1. The AWS Organization management account enables AWS Shield network security director, designates a delegated administrator for the organization, and creates the corresponding delegated administrator policy.
2. The delegated administrator for the organization creates a policy that enables AWS Shield network security director for user-selected regions and target member accounts in the organization.

Enabling AWS Shield network security director and delegating a service administrator

When assigning the delegated administrator account for AWS Shield network security director, AWS Shield network security director will recommend an existing delegated administrator if one is already configured for another AWS security service, such as **AWS Security Hub**. If no delegated administrator exists, you will be prompted to select a member account from your organization. The organization's management account cannot be designated as the delegated administrator.

To designate an administrator for AWS Shield network security director

1. Sign in to your AWS Account with your AWS Organization management account credentials and open the AWS Shield network security director console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/network-security-director/.

2. From the network security director home page, choose **Get started**.
3. For **Delegated administrator account**, choose an administrator account based on the provided options. As a best practice, we recommend using the same delegated administrator across security services for consistent governance.
4. For **Delegated administrator policy**, choose one of the following options to add the policy statement:

    a. (Option 1) Choose **Update this for me**. Select the box under the policy statement to confirm AWS Shield network security director will automatically create a delegation policy granting all required permissions to the delegated administrator.

    b. (Option 2) Choose **I want to attach this manually**. Choose **Copy and attach**. In the AWS Organizations console, under **Delegated administrator for AWS Organizations**, choose **Delegate**, and paste the resource policy in the delegation policy editor and then Choose **Create Policy**. Open the tab where you are in the AWS Shield network security director console.
5. Choose **Complete get started**.

At the end of this step the following actions will be complete:

- Trusted Access enablement for AWS Shield network security director. This will allow network security director to create service-linked roles within member accounts that are in scope of the policy.
- Creation of the service-linked role **AWSServiceRoleForNetworkSecurityDirector** for the organization's management account.
- Registration of the delegated administrator for AWS Shield network security director.
- Update of the resource policy, allowing the delegated administrator for AWS Shield network security director to make necessary calls to AWS Organizations APIs.

Now that the setup is complete, you will be redirected to the **Settings** page, where you can update or remove the delegated administrator, manage delegation policy, and disable network security director as a service. To access this settings page in the future with the organization's management account, navigate to the network security director console and choose **Manage settings**.

Enabling AWS Shield network security director for member accounts with delegated administrator

This step must be completed by the delegated administrator. Once the AWS Organization's management account designates a delegated administrator, that administrator must create a policy that grants permission to enable regions within the organization. All configured policies are available in the **Region and Account Policies** section of the AWS Shield network security director console. The procedure below outlines how to create this policy.

To create and attach a policy that enables regions for targeted accounts

1. Sign in to your AWS account with your delegated administrator credentials and open the AWS Shield network security director console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/network-security-director/.
2. From the AWS Shield network security director home page, choose **Enable**.
3. For **Details**, enter a name and an optional description for the policy.
4. For **Account selection**, select one of the following options. Choose **All organizational units and accounts** if you want to apply the policy to all organizational units and accounts. Choose **Specific organizational units and accounts** if you want to apply the policy to specific organizational units and accounts. Use the search bar or organizational structure tree to specify the organizational units and accounts where the policy will be applied.
5. For **Regions**, select the regions you want to enable or disable for this policy. Please refer to Performance Considerations before completing your selections.
6. Review your changes, and then choose **Enable network security director**.

At the end of this step the following actions will be complete:

- Creation of the service-linked role **AWSServiceRoleForNetworkSecurityDirector** for the current delegated administrator account.
- Creation of the policy that enables AWS Shield network security director to run scans in the enabled regions and on the attached targets.
- Redirection to the **Summary dashboard**, where you can view organization-wide insights as well as resource-level details for each account.

Now that the setup is complete, you will be redirected to the **Summary dashboard** page, where you can view organization-wide insights as well as resource-level details for each account. To manage the policies in the future with the delegated administrator account, navigate to the network security director console and choose **Manage settings**.

Exploring resources and findings

The AWS Shield network security director dashboard provides a summary of the most severe findings, a comparison of findings by included regions, a table of accounts that have been analyzed, a panel that appears when an account is selected, and a network topology that populates once a resource within that panel is selected.

> **Note**
>
> You must sign in with delegated administrator credentials to view resources and findings on the network security director dashboard.

To identify which accounts have findings

1. Sign in with your delegated administrator account and open the AWS Shield network security director console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/network-security-director/.
2. In the navigation pane, under network security director, choose **Dashboard**.
3. The Summary dashboard displays with the following widgets.

Regions Widget

The Regions widget highlights the **most critical findings**, which are a representation of all the findings on resources and their combined severity. Next to it is a chart that includes the current region and provides a comparative summary of finding severities across all enabled regions. This allows for quick identification of critical issues by region and an indicator that you may need to switch regions if there are findings in that region to review.

Accounts Widget

Accounts that are analyzed in the current region appear in the table with the default sort set to show accounts with resources by the highest composite finding severities. You can select an account to open a panel that displays resources by their composite finding severity.

To identify which resources have findings

1. From the network security director Dashboard, navigate to the **Accounts** widget.
2. Select the account which shows the number of composite severity findings.
3. This will open the resources in the **Account and topology explorer** panel widget.

To identify your affected resources and find specific remediation recommendations, see the following sections.

When a network analysis completes, network security director provides detailed recommendations to remediate vulnerabilities identified in resource findings. You can filter for any vulnerable resource based on **Resource ID**, **Severity level**, **Resource type**, or associated **Findings**. By default, the **Resources** table displays resources in order of highest to lowest severity.

Account and topology explorer panel widget

When an account is selected the account and topology explorer panel widget opens to display the resources and findings within that account. Filter the table within to find specific resources or findings or remove filters to see a summary of resources types and finding types.

AWS Shield network security director assigns severity levels to each finding from the most recent network analysis. Resources can be assigned **None**, **Informational**, **Low**, **Medium**, **High**, or **Critical**. This severity level represents the severity level of the most severe finding identified on a resource. For example, if your latest network analysis determines that your Amazon Amazon EC2 instance has one Medium severity finding and two Low severity findings, that resource is assigned a Medium composite finding severity level.

The **Findings overview** widget inside the **Account and topology explorer panel widget**, which can be accessed by removing any resource or finding type filters, provides two ways to understand the findings that network security director found in your resources:

- From **Highest severity resources**, you can quickly understand which severity level is the most severe across all your networking resources. You can also see a list of how many of your

resources are affected and the number of resources assigned each severity level by network security director.

- From **Severity distribution**, you can view the number of resources with a specific severity level for each resource type and compare it with those of other resource types.

To explore your network topology

1. In the navigation pane, under network security director, choose **Dashboard**.
2. From the network security director Dashboard, navigate to the **Accounts** widget.
3. Select the account which shows the number of composite severity findings.
4. This will open the resources in the **Account and topology explorer** panel widget.

AWS Shield network security director maps the connections of your resources during its analyses. These connections are visualized in a network topology that is shown in a widget on the dashboard below the **Accounts** widget. When you select an account, and then select a resource in the **Account and topology explorer** panel widget, the network topology visualization appears in the context of the selected resource.

Network topology widget

The network topology exists in an empty state until a specific resource has been selected in the Account and topology explorer panel widget which appears when an account has been selected in the **Accounts** table widget. The network topology can be navigated by dragging the canvas, or by using the zoom controls in the lower right corner of the container. There are also controls for resetting the zoom and location of the canvas along with exporting its contents. The topology can be refreshed in the case of updated analysis findings and can also expand its view to display the topology across the entire screen. Select a resource in the topology to view its details in the **Account and topology explorer** panel widget. Select a resource edge connection to learn more about the nature of the relationship between two resources in the topology.

> **Note**
>
> The network topology is built by traversing from the selected resource to its connected network resources. Only the first 100 connected resources are displayed in the topology visualization, sorted by severity. This sorting is applied only to the initial set of 100 fetched results, not the complete dataset.

To find recommendations for improving your security

1. From the network security director Dashboard, navigate to the **Accounts** widget.
2. Select the account which shows the number of composite severity findings.
3. This will open the resources in the **Account and topology explorer** panel widget.
4. Selecting a resource in the widget displays its details, with an option to navigate directly to the full **Resource Details page** for deeper investigation.

A resource can have multiple findings identified by network security director. Each finding represents a security issue found during your most recent network analysis.

- Expand the **Remediation recommendations** for the finding to learn more about it.
- Follow the steps suggested by network security director or choose the documentation link included to learn more.

After reviewing and implementing the remediation recommendations for your affected resources, you may want to get additional insights about your overall security configuration. Continue to Analyze network security with Amazon Q Developer to learn how to use Amazon Q Developer for further analysis.

Analyze network security with Amazon Q Developer

Amazon Q Developer is a generative artificial intelligence (generative AI)-powered assistant that works with network security director to help answer your questions and provide recommendations about network security and remediation options.

You can interact with Amazon Q Developer by choosing the Q button or **Explore with Amazon Q Developer** anywhere it appears in the network security director console. This section guides you through the steps to ask questions of Q from the network security director **Dashboard**.

To engage with Amazon Q Developer

> **Note**
>
> You must have a completed network analysis before you can chat with Amazon Q Developer.

1. Sign in to the AWS Management Console and open the AWS Shield network security director console at https://eusc-de-east-1.console.amazonaws-eusc.eu/wafv2/network-security-director/.
2. From the network security director home page, choose **Dashboard**.
3. In the **Ask Amazon Q Developer** widget, choose a question to use as a prompt in the Amazon Q Developer chat interface.
4. In the Amazon Q Developer chat interface, submit your request.

Example questions

Following are example questions about network security that you can ask Amazon Q Developer:

- Identify my top network security findings
- Identify my top network security findings in the account 123456789010
- Identify my top network security findings in us-west-2
- Summarize the network security of my environment
- Are my systems at risk of DDoS attacks?
- How can I improve my network security?
- Do I have any resources without WAF protection?
- Which resources are not protected from common web vulnerabilities?
- What are the common network security issues on my EC2 instances?
- Do I have any WAF web ACLs that aren't protecting anything?

Data protection considerations

For information about how Amazon Q Developer stores your conversations, see Data protection in Amazon Q Developer in the *Amazon Q Developer User Guide*.

For information about how Amazon Q Developer uses cross region processing, see Cross region processing in Amazon Q Developer in the *Amazon Q Developer User Guide*.

AWS Shield network security director quotas

AWS accounts have default quotas, formerly referred to as limits, for each AWS service. The following table describes the quota for network security director. For information about quotas that can be changed, see Service Quotas.

Resource	Default Quota
Maximum resources processed per scan	300,000
Maximum number of edges per resource	200
Maximum number of accounts enabled per region per organization	15,000
Maximum Q prompts per organization per month	200

When network security director reaches the maximum number of resources that it can process in a network analysis or the maximum number of edges per resource, the network analysis fails. You are not charged for the failed network analysis.

Troubleshooting AWS Shield network security director

Unsupported Cross-Account Shared Resources

AWS Shield network security director does not support certain cross-account shared resources. When attempting to scan these resources, you'll receive error messages indicating the resources cannot be analyzed.

Unsupported Shared Resources and Error Messages

Resource Type	Error Message
Network Firewall FirewallPolicy	network-firewall:DescribeFirewallPolicy not supported on shared resources

Resource Type	Error Message
Network Firewall Stateful rule group	network-firewall:DescribeRuleGroup not supported on shared resources
Network Firewall Stateless rule group	network-firewall:DescribeRuleGroup not supported on shared resources
EC2 PrefixList	ec2:GetManagedPrefixListEntries not supported on shared resources

Availability of Resources, Findings, and Suppression

If an account leaves an organization or network security director is disabled for an account, the following occurs:

- **Findings and Resources:** Findings from the account will be removed once the service is disabled for the account. This process typically takes a few minutes but could be longer.
- **Suppressions:** Suppressions are deleted within 90 days of disabling the service for an account. If the service is re-enabled for an account within this 90-day period, existing suppressions might still be available, but availability is not guaranteed. Suppressions must be removed before disabling the service for an account to avoid this uncertainty.

Performance Considerations

AWS Shield network security director is designed to provide daily data refreshes for your organization's network analysis. However, performance can vary based on your organization's size and region.

Organizations with a large number of accounts may experience longer refresh cycles, with data refreshes occurring after multiple days for individual accounts. Additionally, performance can vary significantly by Region, with opt-in regions in particular experiencing slow performance and extended refresh times.

For improved performance and more frequent data refreshes, we recommend enabling network security director for accounts that are specifically relevant to each region. This recommendation is especially critical for opt-in regions.

Additional Resources

If you encounter issues not addressed in this troubleshooting guide, please contact AWS Support for additional assistance.

Security in your use of the AWS Shield network security director

> **Note**
>
> AWS Shield network security director is in public preview release and is subject to change.

This section describes the key security considerations for using this network security director preview.

Data sources

When you run an analysis, network security director retrieves information about your AWS resources using public AWS API endpoints. The information retrieved includes resource attributes that are available to your account through the public AWS APIs.

AWS Shield network security director also uses internal AWS data sources and threat intelligence to identify findings and recommend remediations.

Data encryption

Review the following encryption considerations when using network security director.

- **Encryption at rest** – All data is protected at rest.
- **Encryption in transit** – All data is protected in transit using Transport Layer Security (TLS) encryption. All communication is authenticated using Amazon Simple Storage Service AWS Signature Version 4 (SigV4). For information about SigV4, see Authenticating Requests (AWS Signature Version 4) in the *Amazon S3 User Guide*.
- **Key management** – Customer-managed keys are not currently supported.

Topics

- Identity and Access Management for AWS Shield network security director

- Identity-based policy examples for AWS Shield network security director
- Using service-linked roles for AWS Shield network security director

Identity and Access Management for AWS Shield network security director

> **Note**
>
> AWS Shield network security director is in public preview release and is subject to change.

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use AWS Shield network security director resources. IAM is an AWS service that you can use with no additional charge.

Review the guidance in this section to understand how to use supported policies and roles for AWS Shield network security director.

How AWS Shield network security director works with IAM

This section explains how to use the features of IAM with AWS Shield network security director.

Before you use IAM to manage access to network security director, learn what IAM features are available to use with network security director.

IAM features you can use with AWS Shield network security director

IAM feature	AWS Shield network security director support
Identity-based policies	Yes
Service-linked roles	Yes

To get a high-level view of how network security director and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for network security director

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

To view examples of AWS Shield network security director identity-based policies, see Identity-based policy examples for AWS Shield network security director.

Service-linked roles for network security director

Supports service-linked roles: Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing network security director service-linked roles, see Using service-linked roles for AWS Shield network security director.

Identity-based policy examples for AWS Shield network security director

> **Note**
>
> When you start using AWS Shield network security director, we automatically create a service-linked role that satisfies all the minimum permissions requirements. Creating and managing your own identity-based policies is optional.

To provide appropriate access to network security director, you can create identity-based policies that grant the necessary permissions for administrative and read-only access.

For more information about creating and managing IAM policies, see Managed policies and inline policies in the *IAM User Guide*.

These permissions allow AWS Shield network security director to perform comprehensive security analysis and provide accurate network security recommendations. The example policies provided in this guide are designed for common use cases. You can use these policies as a starting point and modify them as needed to meet your specific requirements.

Example policies in this guide

- Administrative access identity-based policy
- Read-only access identity-based policy

Policy best practices

Identity-based policies determine whether someone can create, access, or delete network security director resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies

adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.

- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Updates to identity-based policies

As updates and features are added to network security director, you may need to update your identity-based policies to include additional permissions. Monitor this guide for information about new permissions that may be required.

Unlike AWS managed policies, customer managed policies are not automatically updated. You are responsible for maintaining and updating these policies as needed.

For more information, see Adding permissions to a user in the *IAM User Guide*.

Administrative access identity-based policy

Create an identity-based policy with the following example to provide full administrative access to network security director operations and the ability to create the required service-linked role.

Policy name: NetworkSecurityDirectorAdminPolicy

Policy description: Allows full administrative access to AWS Shield network security director operations and also provides access to create or delete the service linked role for Network Security Director.

JSON

```
{
```

```
   "Version":"2012-10-17",
   "Statement": [
     {
       "Effect": "Allow",
       "Action": [
         "network-security-director:*"
       ],
       "Resource": "*"
     },
     {
       "Effect": "Allow",
       "Action": [
         "iam:CreateServiceLinkedRole"
       ],
       "Resource": "arn:aws-eusc:iam::*:role/aws-service-role/network-security-
director.amazonaws.com/AWSServiceRoleForNetworkSecurityDirector"
     }
   ]
 }
```

Read-only access identity-based policy

Create an identity-based policy with the following policy example to provide read-only access to network security director operations.

Policy name: NetworkSecurityDirectorReadOnlyPolicy

Policy description: Allows read-only access to AWS Shield network security director.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "network-security-director:Get*",
        "network-security-director:List*"
```

```
      ],
      "Resource": "*"
    }
  ]
}
```

Using service-linked roles for AWS Shield network security director

This section explains how to use service-linked roles to give AWS Shield network security director access to resources in your AWS account.

AWS Shield network security director uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to AWS Shield network security director. Service-linked roles are predefined by AWS Shield network security director and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up AWS Shield network security director easier because you don’t have to manually add the necessary permissions. AWS Shield network security director defines the permissions of its service-linked roles, and unless defined otherwise, only AWS Shield network security director can assume its roles. The defined permissions include the trust policy and the permissions policy. That permissions policy can't be attached to any other IAM entity.

See the full service-linked role in the IAM console: NetworkSecurityDirectorServiceLinkedRolePolicy.

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for AWS Shield network security director

The `NetworkSecurityDirectorServiceLinkedRolePolicy` service-linked role trusts the following services to assume the role:

- `network-director.amazonaws.com`

The `NetworkSecurityDirectorServiceLinkedRolePolicy` grants AWS Shield network security director permissions to access and analyze various AWS resources and services on your behalf. This includes:

- Retrieving network configuration and security settings from Amazon EC2 resources
- Accessing CloudWatch metrics to analyze network traffic patterns
- Gathering information about load balancers and target groups
- Collecting AWS WAF configurations and rules
- Accessing AWS Direct Connect gateway information
- And more, as detailed in the permissions list below

The following listing is for permissions that don't support downscoping to specific resources. The rest are downscoped for the indicated service resources.

```
{
 "Sid": "ResourceLevelPermissionNotSupported",
 "Effect": "Allow",
 "Action": [
   "cloudwatch:GetMetricData",
   "cloudwatch:GetMetricStatistics",
   "ec2:DescribeAvailabilityZones",
   "ec2:DescribeCustomerGateways",
   "ec2:DescribeInstances",
   "ec2:DescribeInternetGateways",
   "ec2:DescribeManagedPrefixLists",
   "ec2:DescribeNatGateways",
   "ec2:DescribeNetworkAcls",
   "ec2:DescribeNetworkInterfaces",
   "ec2:DescribePrefixLists",
   "ec2:DescribeRegions",
   "ec2:DescribeRouteTables",
   "ec2:DescribeSecurityGroups",
   "ec2:DescribeSubnets",
   "ec2:DescribeTransitGateways",
   "ec2:DescribeTransitGatewayVpcAttachments",
   "ec2:DescribeTransitGatewayAttachments",
```

```
    "ec2:DescribeTransitGatewayPeeringAttachments",
    "ec2:DescribeTransitGatewayRouteTables",
    "ec2:DescribeVpcEndpoints",
    "ec2:DescribeVpcEndpointServiceConfigurations",
    "ec2:DescribeVpcPeeringConnections",
    "ec2:DescribeVpcs",
    "ec2:DescribeVpnConnections",
    "ec2:DescribeVpnGateways",
    "ec2:GetTransitGatewayRouteTablePropagations",
    "ec2:GetManagedPrefixListEntries",
    "elasticloadbalancing:DescribeLoadBalancers",
    "elasticloadbalancing:DescribeTargetGroups",
    "elasticloadbalancing:DescribeTags",
    "elasticloadbalancing:DescribeListeners",
    "elasticloadbalancing:DescribeTargetHealth",
    "elasticloadbalancing:DescribeTargetGroupAttributes",
    "elasticloadbalancing:DescribeRules",
    "elasticloadbalancing:DescribeLoadBalancencerAttributes",
    "wafv2:ListWebACLs",
    "cloudfront:ListDistributions",
    "cloudfront:ListTagsForResource",
    "directconnect:DescribeDirectConnectGateways",
    "directconnect:DescribeVirtualInterfaces"
  ],
  "Resource": "*"
}
```

`NetworkSecurityDirectorServiceLinkedRolePolicy` service-linked role permissions

The following list covers all permissions enabled by the `NetworkSecurityDirectorServiceLinkedRolePolicy` service-linked role.

Amazon CloudFront

```
 {
  "Sid": "cloudfront",
  "Effect": "Allow",
  "Action": [
    "cloudfront:GetDistribution"
  ],
  "Resource": "arn:aws:cloudfront::*:distribution/*"
```

```
}
```

AWS WAF

```
{
 "Sid": "wafv2",
 "Effect": "Allow",
 "Action": [
   "wafv2:ListResourcesForWebACL",
   "wafv2:ListRuleGroups",
   "wafv2:ListAvailableManagedRuleGroups",
   "wafv2:GetRuleGroup",
   "wafv2:DescribeManagedRuleGroup",
   "wafv2:GetWebACL"
 ],
 "Resource": [
   "arn:aws:wafv2:*:*:global/rulegroup/*",
   "arn:aws:wafv2:*:*:regional/rulegroup/*",
   "arn:aws:wafv2:*:*:global/managedruleset/*",
   "arn:aws:wafv2:*:*:regional/managedruleset/*",
   "arn:aws:wafv2:*:*:global/webacl/*/*",
   "arn:aws:wafv2:*:*:regional/webacl/*/*",
   "arn:aws:apprunner:*:*:service/*",
   "arn:aws:cognito-idp:*:*:userpool/*",
   "arn:aws:ec2:*:*:verified-access-instance/*"
 ]
}
```

AWS WAF Classic

```
{
 "Sid": "classicWaf",
 "Effect": "Allow",
 "Action": [
   "waf:ListWebACLs",
   "waf:GetWebACL"
 ],
 "Resource": [
   "arn:aws:waf::*:webacl/*",
```

```
    "arn:aws:waf-regional:*:*:webacl/*"
  ]
}
```

AWS Direct Connect

```
{
 "Sid": "directconnect",
 "Effect": "Allow",
 "Action": [
   "directconnect:DescribeConnections",
   "directconnect:DescribeDirectConnectGatewayAssociations",
   "directconnect:DescribeDirectConnectGatewayAttachments",
   "directconnect:DescribeVirtualGateways"
 ],
 "Resource": [
   "arn:aws:directconnect::*:dx-gateway/*",
   "arn:aws:directconnect:*:*:dxcon/*",
   "arn:aws:directconnect:*:*:dxlag/*",
   "arn:aws:directconnect:*:*:dxvif/*"
 ]
}
```

AWS Transit Gateway routes

```
{
 "Sid": "ec2Get",
 "Effect": "Allow",
 "Action": [
   "ec2:SearchTransitGatewayRoutes"
 ],
 "Resource": [
   "arn:aws:ec2:*:*:transit-gateway-route-table/*"
 ]
}
```

AWS Network Firewall

```
 {
  "Sid": "networkFirewall",
  "Effect": "Allow",
  "Action": [
    "network-firewall:ListFirewalls",
    "network-firewall:ListFirewallPolicies",
    "network-firewall:ListRuleGroups",
    "network-firewall:DescribeFirewall",
    "network-firewall:DescribeFirewallPolicy",
    "network-firewall:DescribeRuleGroup"
  ],
  "Resource": [
    "arn:aws:network-firewall:*:*:*/*"
  ]
}
```

Amazon API Gateway

```
 {
   "Sid": "apiGatewayGetAPI",
   "Effect": "Allow",
   "Action": [
     "apigateway:GET"
   ],
  "Resource": [
    "arn:aws:apigateway:*::/restapis",
    "arn:aws:apigateway:*::/restapis/*",
    "arn:aws:apigateway:*::/apis",
    "arn:aws:apigateway:*::/apis/*",
    "arn:aws:apigateway:*::/tags/*",
    "arn:aws:apigateway:*::/vpclinks",
    "arn:aws:apigateway:*::/vpclinks/*"
  ]
 }
```

Creating a service-linked role for AWS Shield network security director

You don't need to manually create a service-linked role. When you run your first network analysis, AWS Shield network security director creates the service-linked role for you.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you enable AWS Shield network security director logging, AWS Shield network security director creates the service-linked role for you again.

Editing a service-linked role for AWS Shield network security director

AWS Shield network security director doesn't allow you to edit the `NetworkSecurityDirectorServiceLinkedRolePolicy` service-linked role. After you create a service-linked role, you can't change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting a service-linked role for AWS Shield network security director

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don’t have an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

This protects your AWS Shield network security director resources because you can't inadvertently remove permission to access the resources.

> **Note**
>
> If the AWS Shield network security director service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To manually delete the service-linked role using IAM

Use the IAM console, the IAM CLI, or the IAM API to delete the `NetworkSecurityDirectorServiceLinkedRolePolicy` service-linked role. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

Supported Regions for AWS Shield network security director service-linked roles

> **Note**
>
> AWS Shield network security director is in public preview release and is subject to change.

AWS Shield network security director supports using service-linked roles in following regions and can only retrieve data about your resources in these regions.

Region Name	Region
US East (N. Virginia)	us-east-1
Europe (Stockholm)	eu-north-1
Asia Pacific (Thailand)	ap-southeast-7
Africa (Cape Town)	ap-south-1
US East (Ohio)	us-east-2
Asia Pacific (Malaysia)	ap-southeast-5
Asia Pacific (Tokyo)	ap-northeast-1
US West (Oregon)	us-west-2
Europe (Spain)	eu-south-2
Europe (Ireland)	eu-west-1
Europe (Frankfurt)	eu-central-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2

Logging AWS Shield network security director API calls with AWS CloudTrail

AWS Shield network security director integrates with AWS CloudTrail to record all API calls as events. This integration captures calls made from the network security director console, programmatic calls to network security director APIs, and calls made from other AWS services.

With CloudTrail, you can view recent events in the Event history or create a trail to deliver ongoing logs to an Amazon Simple Storage Service bucket. These logs provide details about each request, including the identity of the caller, the time, the request parameters, and the response.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

network security director information in CloudTrail

CloudTrail is automatically enabled on your AWS account. When activity occurs in network security director, it's recorded as an event in CloudTrail. For an ongoing record of events, create a trail that delivers log files to an Amazon S3 bucket.

For more information about creating and managing trails, see:

- Creating a Trail for Your AWS Account
- AWS Service Integrations with CloudTrail Logs
- Receiving CloudTrail Log Files from Multiple Regions and Accounts

network security director API operations logged by CloudTrail

All network security director API operations are logged by CloudTrail and documented in the API Reference. The following operations are included:

- *ListResources*: Lists resources available in the service
- *GetResource*: Retrieves detailed information about a specific resource
- *ListFindings*: Lists security findings
- *GetFinding*: Retrieves detailed information about a specific finding
- *UpdateFinding*: Updates the status or other attributes of a finding
- *ListRemediations*: Lists remediation recommendations for a finding

- *ListInsights*: Lists insights based on findings and resources
- *ListAccountSummaries*: Lists account summaries for an organization

Understanding network security director log file entries

CloudTrail log entries contain information about who made the request, when it was made, and what parameters were used. Here's an example of a ListAccountSummaries action:


```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::111122223333:user/janedoe",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/janedoe",
        "accountId": "111122223333",
        "userName": "janedoe"
      },
      "attributes": {
        "creationDate": "2025-11-11T02:57:20Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2025-11-11T02:59:53Z",
  "eventSource": "network-security-director.amazonaws.com",
  "eventName": "ListAccountSummaries",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "aws-cli/1.18.147 Python/2.7.18 Linux/5.10.244-220.970.amzn2int.x86_64
 botocore/1.18.6",
  "requestParameters": {
    "status": "ACTIVE",
    "sortBy": "SEVERITY",
    "maxResults": 2
```

```
  },
  "responseElements": null,
  "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}
```


Monitoring CloudTrail logs with Amazon CloudWatch

You can use Amazon CloudWatch to monitor and alert on specific API activity in CloudTrail logs. This helps you detect unauthorized access attempts, configuration changes, or unusual activity patterns.

To set up CloudWatch monitoring:

1. Configure your CloudTrail trail to send logs to CloudWatch Logs
2. Create metric filters to extract specific information from log events
3. Create alarms based on these metrics

For detailed instructions, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs.

Best practices for CloudTrail with network security director

To maximize security and auditability with CloudTrail:

- *Enable CloudTrail in all regions* for comprehensive coverage
- *Enable log file integrity validation* to detect unauthorized modifications
- *Use IAM to control access to CloudTrail logs* following least privilege principles
- *Set up alerts for critical events* using CloudWatch alarms
- *Regularly review CloudTrail logs* to identify unusual activity

AWS Firewall Manager

AWS Firewall Manager simplifies your administration and maintenance tasks across multiple accounts and resources for a variety of protections, including AWS WAF, AWS Shield Advanced, Amazon VPC security groups and network ACLs, AWS Network Firewall, and Amazon Route 53 Resolver DNS Firewall. With Firewall Manager, you set up your protections just once and the service automatically applies them across your accounts and resources, even as you add new accounts and resources.

Firewall Manager provides these benefits:

- Helps to protect resources across accounts
- Helps to protect all resources of a particular type, such as all Amazon CloudFront distributions
- Helps to protect all resources with specific tags
- Automatically adds protection to resources that are added to your account
- Allows you to subscribe all member accounts in an AWS Organizations organization to AWS Shield Advanced, and automatically subscribes new in-scope accounts that join the organization
- Allows you to apply security group rules to all member accounts or specific subsets of accounts in an AWS Organizations organization, and automatically applies the rules to new in-scope accounts that join the organization
- Lets you use your own rules, or purchase managed rules from AWS Marketplace

Firewall Manager is particularly useful when you want to protect your entire organization rather than a small number of specific accounts and resources, or if you frequently add new resources that you want to protect. Firewall Manager also provides centralized monitoring of DDoS attacks across your organization.

> **Note**
>
> Charges incurred by AWS Firewall Manager are for the underlying services, such as AWS WAF and AWS Config. For more information, see AWS Firewall Manager Pricing.

Topics

- AWS Firewall Manager prerequisites

- Using AWS Firewall Manager administrators
- Setting up AWS Firewall Manager policies
- Using AWS Firewall Manager policies
- Using Firewall Manager managed lists
- Grouping your resources in Firewall Manager
- Viewing compliance information for an AWS Firewall Manager policy
- AWS Firewall Manager integration with AWS Security Hub CSPM
- Security in your use of the AWS Firewall Manager service
- AWS Firewall Manager quotas
- Migrating AWS WAF Classic Web ACLs in Firewall Manager

AWS Firewall Manager prerequisites

This topic shows you how to get ready to administer AWS Firewall Manager. You use one Firewall Manager administrator account to manage all Firewall Manager security policies for your organization in AWS Organizations. Except where noted, perform the prerequisite steps using the account that you will use as the Firewall Manager administrator.

Before you use Firewall Manager for the first time, perform the following steps in sequence.

Topics

- Joining and configuring AWS Organizations for using Firewall Manager
- Creating an AWS Firewall Manager default administrator account
- Enabling AWS Config for using Firewall Manager
- Subscribing in the AWS Marketplace and configuring third-party settings for Firewall Manager third-party policies
- Enabling resource sharing for Network Firewall and DNS Firewall policies with AWS RAM
- Using AWS Firewall Manager in Regions that are disabled by default

Joining and configuring AWS Organizations for using Firewall Manager

To use Firewall Manager, your account must be a member of the organization in the AWS Organizations service where you want to use your Firewall Manager policies.

> **Note**
>
> For information about Organizations, see AWS Organizations User Guide.

To establish the required AWS Organizations membership and configuration

1. Choose an account to use as the Firewall Manager administrator for the organization in Organizations.
2. If your chosen account isn't already a member of the organization, have it join. Follow the guidance at Inviting an AWS account to join your organization.
3. AWS Organizations has two available feature sets: *consolidated billing features* and *all features*. To use Firewall Manager, your organization must be enabled for all features. If your organization is configured only for consolidated billing, follow the guidance at Enabling All Features in Your Organization.

Creating an AWS Firewall Manager default administrator account

This page provides instructions for creating an AWS Firewall Manager default administrator account.

> **Note**
>
> This procedure uses the account and organization that you chose and configured in the preceding step.

Only the organization's management account can create Firewall Manager default administrator accounts. The first administrator account that you create is the *default admininstrator* account. The default administrator account can manage third-party firewalls and has full administrative scope. When you set the default administrator account, Firewall Manager automatically sets it as an AWS Organizations delegated administrator for Firewall Manager. This allows Firewall Manager to access information about the organizational units (OUs) in the organization. You can use OUs to specify the scope of your Firewall Manager policies. For more information about setting policy scope, see the guidance for the individual policy types under Creating an AWS Firewall Manager policy. For more information about Organizations and management accounts, see Managing the AWS Accounts in Your Organization.

Required settings for the organization's management account

The organization's management account must have the following settings in order to onboard the organization to Firewall Manager and create a default administrator:

- It must be a member of the organization in AWS Organizations where you want to apply your Firewall Manager policies.

To set the default administrator account

1. Sign in to the Firewall Manager AWS Management Console using an existing AWS Organizations management account.
2. Open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2.
3. In the navigation pane, choose **Settings**.
4. Type the AWS account ID of the account that you've chosen to use as the Firewall Manager administrator.

   > **Note**
   >
   > The default administrator has full administrative scope. Full administrative scope means that this account can apply policies to all accounts and organizational units (OUs) within the organization, take actions in all Regions, and manage all Firewall Manager policy types.

5. Choose **Create administrator account** to create the account.

For more information about managing the Firewall Manager administrator account, see Using AWS Firewall Manager administrators.

Enabling AWS Config for using Firewall Manager

To use Firewall Manager, you must enable AWS Config.

> **Note**
>
> You incur charges for your AWS Config settings, according to AWS Config pricing. For more information, see Getting Started with AWS Config.

> **Note**
>
> In order for Firewall Manager to monitor policy compliance, AWS Config must continuously record configuration changes for protected resources. In your AWS Config configuration, the recording frequency must be set to **Continuous**, which is the default setting.

To enable AWS Config for Firewall Manager

1. Enable AWS Config for each of your AWS Organizations member accounts, including the Firewall Manager administrator account. For more information, see Getting Started with AWS Config.
2. Enable AWS Config for each AWS Region that contains the resources that you want to protect. You can enable AWS Config manually, or you can use the CloudFormation template "Enable AWS Config" at AWS CloudFormation StackSets Sample Templates.

   If you don't want to enable AWS Config for all resources, then you must enable the following according to the type of Firewall Manager policies that you use:

   - **WAF policy** – Enable Config for the resource types CloudFront Distribution, Application Load Balancer (choose **ElasticLoadBalancingV2** from the list), API Gateway, WAF WebACL, WAF Regional WebACL, and WAFv2 WebACL. To enable AWS Config to protect a CloudFront distribution, you must be in the US East (N. Virginia) Region. Other Regions don't have CloudFront as an option.
   - **Shield policy** – Enable Config for the resource types Shield Protection, ShieldRegional Protection, Application Load Balancer, EC2 EIP, WAF WebACL, WAF Regional WebACL, and WAFv2 WebACL.
   - **Common security group policy** – Enable Config for the resource types Amazon EC2 SecurityGroup and Amazon EC2 VPC, plus the resource types in the policy's resource type list (typically Amazon EC2 Instance and Amazon EC2 NetworkInterface). If you enable security group references distribution on the policy, also enable Amazon EC2 VPCPeeringConnection.
   - **Security group content audit policy** – Enable Config for the resource types Amazon EC2 SecurityGroup, Amazon EC2 NetworkInterface, and Amazon EC2 Instance.
   - **Security group usage audit policy** – Enable Config for the resource type Amazon EC2 SecurityGroup.

- **Network ACL policy** – Enable Config for the resource types Amazon EC2 Subnet and Amazon EC2 network ACL.
- **Network Firewall policy** – Enable Config for the resource types NetworkFirewall FirewallPolicy, NetworkFirewall RuleGroup, EC2 VPC, EC2 InternetGateway, EC2 RouteTable, and EC2 Subnet.
- **DNS Firewall policy** – Enable Config for the resource type EC2 VPC and Amazon Route 53 FirewallRuleGroupAssociation.
- **Third-party firewall policy** – Enable Config for the resource types Amazon EC2 VPC, Amazon EC2 InternetGateway, Amazon EC2 RouteTable, Amazon EC2 Subnet, and Amazon EC2 VPCEndpoint.

> **Note**
>
> If you configure your AWS Config recorder to use a custom IAM role, you need to make sure the IAM policy has the proper permissions to record the Firewall Manager policy's required resource types. Without the proper permissions, the required resources may not be recorded which prevents Firewall Manager from properly protecting your resources. Firewall Manager doesn't have visibility into these permission misconfigurations. For information about using IAM with AWS Config, see IAM for AWS Config.

Subscribing in the AWS Marketplace and configuring third-party settings for Firewall Manager third-party policies

Complete the following prerequisites to set up Firewall Manager third-party firewall policies.

Fortigate Cloud Native Firewall (CNF) as a Service policy prerequisites

To use Fortigate CNF for Firewall Manager

1. Subscribe to the Fortigate Cloud Native Firewall (CNF) as a Service service in the AWS Marketplace.
2. First, register a tenant on the Fortigate CNF product portal. Then Add your Firewall Manager administrator account under your tenant on the Fortigate CNF product portal. For more information, see the Fortigate CNF documentation.

For information about working with Fortigate CNF policies, see Using Fortigate Cloud Native Firewall (CNF) as a Service policies for Firewall Manager.

Palo Alto Networks Cloud Next Generation Firewall policy prerequisites

To use Palo Alto Networks Cloud NGFW for Firewall Manager

1. Subscribe to the Palo Alto Networks Cloud Next Generation Firewall Pay-As-You-Go service in the AWS Marketplace.
2. Complete the Palo Alto Networks Cloud NGFW deployment steps listed in the Deploy Palo Alto Networks Cloud NGFW for AWS with the AWS Firewall Manager topic in the *Palo Alto Networks Cloud Next Generation Firewall for AWS deployment guide*.

For information about working with Palo Alto Networks Cloud NGFW policies, see Using Palo Alto Networks Cloud NGFW policies for Firewall Manager.

Enabling resource sharing for Network Firewall and DNS Firewall policies with AWS RAM

To manage Firewall Manager Network Firewall and DNS Firewall policies, you must enable sharing with AWS Organizations in AWS Resource Access Manager. This allows Firewall Manager to deploy protections across your accounts when you create these policy types.

To enable sharing with AWS Organizations in AWS Resource Access Manager

- Follow the guidance at Enable Sharing with AWS Organizations in the *AWS Resource Access Manager User Guide*.

If you run into problems with resource sharing, see the guidance at Resource sharing for Network Firewall and DNS Firewall policies.

Using AWS Firewall Manager in Regions that are disabled by default

To use Firewall Manager in a Region that's disabled by default, you must enable the Region for both the management account of your AWS organization and the Firewall Manager default administrator account. For information about Regions that are disabled by default and how to enable them, see Managing AWS Regions in the *AWS General Reference*.

To enable a disabled Region

- For both the Organizations management account and the Firewall Manager default administrator account, follow the guidance at Enabling a Region in the *AWS General Reference*.

After you follow these steps, you can configure Firewall Manager to begin protecting your resources. For more information, see Setting up AWS Firewall Manager AWS WAF policies.

Using AWS Firewall Manager administrators

This page explains what Firewall Manager administrators are and defines related terms.

With AWS Firewall Manager you can have one or multiple administrators who can manage the firewall resources of your organization. If you want to use multiple Firewall Manager administrators in your organization, you can apply administrative scope conditions to each administrator to define the resources that they can manage. This gives you the flexibility to have different administrator roles within your organization, and helps you maintain the principal of least privileged access. For example, you can have one administrator manage a set of organizational units (OUs) for your organization, while delegating another administrator to manage only specific Firewall Manager policy types. For more information about Organizations and management accounts, see Managing the AWS Accounts in Your Organization.

For the maximum number of administrators that you can have per organization, see AWS Firewall Manager quotas

Getting started using Firewall Manager administrators

Before you begin using Firewall Manager administrators, you must complete the prerequisites listed in AWS Firewall Manager prerequisites. In the prerequisites, you'll onboard an AWS Organizations organization to Firewall Manager and create a default administrator account for Firewall Manager. A default administrator account has the ability to manage third-party firewalls and has full administrative scope.

Administrative scope

Administrative scope defines the resources that the Firewall Manager administrator can manage. After an AWS Organizations management account onboards an organization to Firewall Manager, the management account can create additional Firewall Manager administrators with different

administrative scopes. An AWS Organizations management account can either grant the administrator **full** or **restricted** administrative scope. Full scope gives the administrator full access to all of the preceding resource types. Restricted scope refers to granting administrative permission to only a subset of the preceding resources. We recommend that you only grant administrators the permissions they need to perform the duties of their role. You can apply any combination of these administrative scope conditions to an administrator:

- Accounts or OUs in your organization that the administrator can apply policies to.
- Regions that the administrator can perform actions in.
- Firewall Manager policy types that the administrator can manage.

Administrator roles

There are two types of administrator roles in Firewall Manager: a default administrator, and Firewall Manager administrators.

- Default administrator - The organization's management account creates a Firewall Manager *default administrator* account when they onboard their organization to Firewall Manager while completing the AWS Firewall Manager prerequisites. The default administrator can manage third-party firewalls and has full administrative scope, but is otherwise at the same peer level as other administrators, if you choose to have multiple administrators.
- Firewall Manager administrators - A Firewall Manager administrator can manage the resources that the AWS Organizations management account designates for them in their administrative scope configuration. For the maximum number of administrators that you can have per organization, see AWS Firewall Manager quotas. Upon creation of a Firewall Manager administrator account, the service checks with AWS Organizations to see if the account is already a delegated administrator for Firewall Manager within the organization. If not, then Firewall Manager calls Organizations to set the account as a delegated administrator for Firewall Manager. For information about Organizations delegated administrators, see AWS Organizations terminology and concepts in the *AWS Organizations User Guide*.

Existing administrators

If you are an existing Firewall Manager customer and have set already set an administrator, then this existing administrator will be the Firewall Manager default administrator. There should be no impacts to your existing flow. If you wish to add more administrators, you can do so by following the procedures in this chapter.

Creating a Firewall Manager administrator account

The following procedure describes how to create a Firewall Manager administrator account using the Firewall Manager console.

> **Note**
>
> Only an organization's managment account can create Firewall Manager administrator accounts.

To create a Firewall Manager administrator account

1. Sign in to the Firewall Manager AWS Management Console using an existing AWS Organizations management account.
2. Open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2.
3. In the navigation pane, choose **Settings**.
4. Choose **Create administrator account**.
5. In the **Details** pane, for **AWS account ID** type the AWS ID of a member account that you'd like to add as a Firewall Manager administrator.
6. For **Administrative scope**, choose one of the following options:

   - **Full** – This grants the administrator the ability to apply policies to all accounts and organizational units (OUs) within the organization, take actions in all Regions, and apply all Firewall Manager policy types, except for third-party firewalls. Only the default administrator can create and manage third-party firewalls. Take caution if granting this level of permissions to the administrator. In the spirit of least privilege, we recommend only granting the administrator the permissions they need to perform the duties of their role.
   - **Restricted** – If applying a **Restricted** scope, then in **Configure administrative scope** configure the accounts and organizational units, Regions, and policy types that the account can manage.

     For **Accounts and organizational units**, choose the options as follows:

     - If you want to apply policies to all accounts or organizational units in your organization, choose **Include all accounts under my AWS organization**.
     - If you want to apply policies only to specific accounts or accounts that are in specific AWS Organizations organizational units (OUs), choose **Include only the specified accounts**

and organizational units, and then add the accounts and OUs that you want to include. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

- If you want to apply policies to all but a specific set of accounts or AWS Organizations organizational units (OUs), choose **Exclude the specified accounts and organizational units, and include all others**, and then add the accounts and OUs that you want to exclude. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

For **Regions**, choose the options as follows:

- If you want to allow the administrator to perform actions in all available Regions, choose **Include all Regions**.
- If you want the administrator to perform actions only in specific Regions, choose **Include only the specified Regions**, and then specify the Regions that you want to include.

> **Note**
>
> To include a Region that is disabled by default, you must enable the Region for both the AWS Organizations organization management account and the default administration account. For information about enabling Regions for an account, see Enable a Region in the *Amazon Web Services General Reference*.

For **Policy types**, choose the options as follows:.

- If you want to allow the administrator to manage all policy types, choose **Include all policy types**.
- If you want the administrator to manage only specific policy types, choose **Include only the specified policy types**, and then specify the policy types that you want to include.

7. Choose **Create administrator account** to create the administrator account. Upon creation, Firewall Manager calls AWS Organizations to see if the administrator is already a delegated administrator for your organization. If not, Firewall Manager will designate the account as a delegated administrator. For information about delegated administrators in Organizations see AWS Organizations terminology and concepts in the *AWS Organizations User Guide*.

If you apply **Restricted** administrative scope, Firewall Manager automatically evaluates any new resources against your settings. For example, if you include only specific accounts, Firewall Manager

doesn't apply the policy to any new accounts. As another example, if you include an OU, when you add an account to the OU or to any of its child OUs, Firewall Manager automatically includes the account within the administrative scope.

Updating a Firewall Manager administrator account

The following procedure describes how to update a Firewall Manager administrator account using the Firewall Manager console.

> **Note**
>
> To update an administrator's scope to include a Region that's disabled by default, you must enable the Region for both the AWS Organizations organization management account and the default administration account. For information about enabling Regions for an account, see Enable a Region in the *Amazon Web Services General Reference*.
> Only an organization's managment account can update Firewall Manager administrator accounts.

To update an administrator account (console)

1. Sign in to the Firewall Manager AWS Management Console using an existing AWS Organizations management account.
2. Open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2.
3. In the navigation pane, choose **Settings**.
4. in the **Firewall Manager administrators** table, choose the account that you'd like to update.
5. Select **Edit** to change details of administrator's account. You can't change the **account ID**.
6. Choose **Save** to save your changes.

Revoking a Firewall Manager administrator account

The following procedure describes how to revoke a Firewall Manager administrator account. If you are the default administrator, before you can revoke your account all of the Firewall Manager administrator accounts within your organization must first revoke their own accounts.

> **Note**
>
> Only an individual Firewall Manager administrator can revoke their own administrator account.

To revoke an administrator account (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.
2. In the navigation pane, choose **Settings**.
3. In the **Administrator account** pane, select **Revoke administrator account** to revoke your account.

> **Important**
>
> When you revoke administrator privileges from an administrator account, all Firewall Manager policies created by that account are deleted.

Changing the default Firewall Manager administrator account

The following procedure describes how to change the default Firewall Manager administrator account.

You can designate only one account in an organization as the default Firewall Manager administrator account. The default administrator account follows the principle of first in, last out. To designate a different default administrator account, each individual administrator account must first revoke their own account. Then, the existing default administrator can revoke their own account, which also will offboard the organization from Firewall Manager. When an administrator revokes their account, all Firewall Manager policies created by that account are deleted. To designate a new default administrator account, you then must sign into Firewall Manager with the AWS Organizations management account to designate a new administrator account. To change the default administrator account for an organization, perform the following procedure.

To change the default administrator account

1. Sign in to the Firewall Manager AWS Management Console using an existing AWS Organizations management account.
2. Open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2.
3. In the navigation pane, choose **Settings**.
4. Type the ID of the account that you've chosen to use as the Firewall Manager administrator.

> **Note**
>
> This account is given permission to create and manage Firewall Manager policies across all accounts within your organization.

5. Choose **Create administrator account**.
6. Type the AWS ID of the account that you've chosen to use as the Firewall Manager administrator.

> **Note**
>
> This account is given full administrative scope. Full administrative scope means that this account can apply policies to all accounts and organizational units (OUs) within the organization, take actions in all Regions, and manage all Firewall Manager policy types.

7. Choose **Create administrator account** to create the default administrator account.

Disqualifying changes to a Firewall Manager administrator account

Some changes to an administrator account can disqualify it from remaining an administrator account.

This section describes the changes that can disqualify the an administrator account, and how AWS and Firewall Manager handle these changes.

Account removed from the organization in AWS Organizations

If the AWS Firewall Manager administrator account is removed from the organization in AWS Organizations, it can no longer administer policies for the organization. Firewall Manager takes one of the following actions:

- **Account with no policies** – If the Firewall Manager administrator account has no Firewall Manager policies, Firewall Manager revokes the administrator account.
- **Account with Firewall Manager policies** – If the Firewall Manager administrator account has Firewall Manager policies, Firewall Manager sends an email to inform you of the situation and to provide options that you can take, with the help of your AWS sales account representative.

Account closed

If you close the account that you're using for the AWS Firewall Manager administrator, AWS and Firewall Manager handle the closure as follows:

- AWS revokes the account's administrator access from Firewall Manager and Firewall Manager deactivates any policies that were managed by the administrator account. The protections that were provided by those policies are stopped across the organization.
- AWS retains the Firewall Manager policy data for the account for 90 days from the effective date of the administrator account closure. During this 90-day period, you can reopen the closed account.
  - If you reopen the closed account during the 90-day period, AWS reassigns the account as the Firewall Manager administrator and recovers the Firewall Manager policy data for the account.
  - Otherwise, at the end of the 90-day period, AWS permanently deletes all Firewall Manager policy data for the account.

Setting up AWS Firewall Manager policies

You can use AWS Firewall Manager to enable a number of different types of security policies. The steps for getting set up are slightly different for each.

Topics

- Setting up AWS Firewall Manager AWS WAF policies
- Setting up AWS Firewall Manager AWS Shield Advanced policies
- Setting up AWS Firewall Manager Amazon VPC security group policies
- Setting up AWS Firewall Manager Amazon VPC network ACL policies
- Setting up AWS Firewall Manager AWS Network Firewall policies
- Setting up AWS Firewall Manager DNS Firewall policies

- Setting up AWS Firewall Manager Palo Alto Networks Cloud Next Generation Firewall policies
- Setting up AWS Firewall Manager Fortigate CNF policies

Setting up AWS Firewall Manager AWS WAF policies

To use AWS Firewall Manager to enable AWS WAF rules across your organization, perform the following steps in sequence.

Topics

- Step 1: Completing the prerequisites
- Step 2: Creating and applying an AWS WAF policy
- Step 3: Cleaning Up

Step 1: Completing the prerequisites

There are several mandatory steps to prepare your account for AWS Firewall Manager. Those steps are described in AWS Firewall Manager prerequisites. Complete all of the prerequisites before proceeding to Step 2: Creating and applying an AWS WAF policy.

Step 2: Creating and applying an AWS WAF policy

A Firewall Manager AWS WAF policy contains the rule groups that you want to apply to your resources. Firewall Manager creates a Firewall Manager web ACL in each account where you apply the policy. The individual account managers can add rules and rule groups to the resulting web ACL, in addition to the rule groups that you define here. For information about Firewall Manager AWS WAF policies, see Using AWS WAF policies with Firewall Manager.

> **Note**
>
> Web ACLs that Firewall Manager creates can't participate in CloudFront flat-rate pricing plans. If you attempt to manage a web ACL that's part of a CloudFront flat-rate pricing plan, Firewall Manager does not report compliance status for both the CloudFront distribution and the Firewall Manager policy. This state can't be remediated through Firewall Manager or any other action.

To create a Firewall Manager AWS WAF policy (console)

Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

1. In the navigation pane, choose **Security policies**.
2. Choose **Create policy**.
3. For **Policy type**, choose **AWS WAF**.
4. For **Region**, choose an AWS Region. To protect Amazon CloudFront distributions, choose **Global**.

   To protect resources in multiple Regions (other than CloudFront distributions), you must create separate Firewall Manager policies for each Region.
5. Choose **Next**.
6. For **Policy name**, enter a descriptive name. Firewall Manager includes the policy name in the names of the web ACLs that it manages. The web ACL names have `FMManagedWebACLV2-` followed by the policy name that you enter here, `-`, and the web ACL creation timestamp, in UTC milliseconds. For example, `FMManagedWebACLV2-MyWAFPolicyName-1621880374078`.

   > **Important**
   >
   > Web ACL names can't change after creation. If you update your policy's name, Firewall Manager won't update the associated web ACL name. To have Firewall Manager create a web ACL with a different name, you must create a new policy.

7. Under **Policy rules**, for **First rule groups**, choose **Add rule groups**. Expand the **AWS managed rule groups**. For **Core rule set**, toggle **Add to web ACL**. For **AWS known bad inputs**, toggle **Add to web ACL**. Choose **Add rules**.

   For **Last rule groups**, choose **Add rule groups**. Expand the **AWS managed rule groups** and for the **Amazon IP reputation list**, toggle **Add to web ACL**. Choose **Add rules**.

   Under **First rule groups**, select **Core rule set** and choose **Move down**. AWS WAF evaluates web requests against the **AWS known bad inputs** rule group before it evaluates against the **Core rule set**.

You can also create your own AWS WAF rule groups if you want, using the AWS WAF console. Any rule groups that you create show up under **Your rule groups** in the **Describe policy : Add rule groups page**.

The first and last AWS WAF rule groups that you manage through Firewall Manager have names that begin with `PREFMManaged-` or `POSTFMManaged-`, respectively, followed by the Firewall Manager policy name, and the rule group creation timestamp, in UTC milliseconds. For example, `PREFMManaged-MyWAFPolicyName-1621880555123`.

8. Leave the default action for the web ACL at **Allow**.
9. Leave the **Policy action** at the default, to not automatically remediate noncompliant resources. You can change the option later.
10. Choose **Next**.
11. For **Policy scope**, you provide the settings for the accounts, resource types, and tagging that identify the resources you want to apply the policy to. For this tutorial, leave the **AWS accounts** and **Resources** settings, and choose one or more resource types.
12. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.
13. Choose **Next**.
14. For **Policy tags**, add any identifying tags that you want to add to the Firewall Manager policy resource. For more information about tags, see Working with Tag Editor.
15. Choose **Next**.
16. Review the new policy settings and return to any pages where you need to any adjustments.

    Check to be sure that **Policy actions** is set to **Identify resources that don't comply with the policy rules, but don't auto remediate.** This allows you to review the changes that your policy would make before you enable them.
17. When you are satisfied with the policy, choose **Create policy**.

In the **AWS Firewall Manager policies** pane, your policy should be listed. It will probably indicate **Pending** under the accounts headings and it will indicate the status of the **Automatic remediation** setting. The creation of a policy can take several minutes. After the **Pending** status is replaced with account counts, you can choose the policy name to explore the compliance status of the accounts and resources. For information, see Viewing compliance information for an AWS Firewall Manager policy

Step 3: Cleaning Up

To avoid extraneous charges, delete any unnecessary policies and resources.

To delete a policy (console)

1. On the **AWS Firewall Manager policies** page, choose the radio button next to the policy name, and then choose **Delete**.
2. In the **Delete** confirmation box, select **Delete all policy resources**, and then choose **Delete** again.

   AWS WAF removes the policy and any associated resources, like web ACLs, that it created in your account. The changes might take a few minutes to propagate to all accounts.

Setting up AWS Firewall Manager AWS Shield Advanced policies

You can use AWS Firewall Manager to enable AWS Shield Advanced protections across your organization.

> **Important**
>
> Firewall Manager doesn't support Amazon Route 53 or AWS Global Accelerator. If you need to protect these resources with Shield Advanced, you can't use a Firewall Manager policy. Instead, follow the instructions in Adding AWS Shield Advanced protection to AWS resources.

To use Firewall Manager to enable Shield Advanced protection, perform the following steps in sequence.

Topics

- Step 1: Completing the prerequisites
- Step 2: Creating and applying a Shield Advanced policy
- Step 3: (Optional) Authorizing the Shield Response Team (SRT)
- Step 4: Configuring Amazon SNS notifications and Amazon CloudWatch alarms

Step 1: Completing the prerequisites

There are several mandatory steps to prepare your account for AWS Firewall Manager. Those steps are described in AWS Firewall Manager prerequisites. Complete all the prerequisites before proceeding to Step 2: Creating and applying a Shield Advanced policy.

Step 2: Creating and applying a Shield Advanced policy

After completing the prerequisites, you create an AWS Firewall Manager Shield Advanced policy. A Firewall Manager Shield Advanced policy contains the accounts and resources that you want to protect with Shield Advanced.

> **Important**
>
> Firewall Manager does not support Amazon Route 53 or AWS Global Accelerator. If you need to protect these resources with Shield Advanced, you can't use a Firewall Manager policy. Instead, follow the instructions in Adding AWS Shield Advanced protection to AWS resources.

To create a Firewall Manager Shield Advanced policy (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. Choose **Create policy**.
4. For **Policy type**, choose **Shield Advanced**.

   To create a Shield Advanced policy, your Firewall Manager administrator account must be subscribed to Shield Advanced. If you are not subscribed, you are prompted to do so. For information about the cost for subscribing, see AWS Shield Advanced Pricing.

   > **Note**
   >
   > You don't need to manually subscribe each member account to Shield Advanced. Firewall Manager does this for you when it creates the policy. Each account must remain subscribed for Firewall Manager and Shield Advanced to continue to protect resources in the account.

5. For **Region**, choose an AWS Region. To protect Amazon CloudFront resources, choose **Global**.

   To protect resources in multiple Regions (other than CloudFront resources), you must create separate Firewall Manager policies for each Region.
6. Choose **Next**.
7. For **Name**, enter a descriptive name.
8. (Global Region only) For **Global** Region policies, you can choose whether you want to manage Shield Advanced automatic application layer DDoS mitigation. For this tutorial, leave this choice at the default setting of **Ignore**.
9. For **Policy action**, choose the option that doesn't automatically remediate.
10. Choose **Next**.
11. **AWS accounts this policy applies to** allows you to narrow the scope of your policy by specifying accounts to include or exclude. For this tutorial, choose **Include all accounts under my organization.**
12. Choose the types of resources that you want to protect.

    Firewall Manager doesn't support Amazon Route 53 or AWS Global Accelerator. If you need to protect these resources with Shield Advanced, you can't use a Firewall Manager policy. Instead, follow the Shield Advanced guidance at Adding AWS Shield Advanced protection to AWS resources.

13. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.
14. Choose **Next**.
15. For **Policy tags**, add any identifying tags that you want to add to the Firewall Manager policy resource. For more information about tags, see Working with Tag Editor.
16. Choose **Next**.
17. Review the new policy settings and return to any pages where you need to any adjustments.

    Check to be sure that **Policy actions** is set to **Identify resources that don't comply with the policy rules, but don't auto remediate.** This allows you to review the changes that your policy would make before you enable them.
18. When you are satisfied with the policy, choose **Create policy**.

    In the **AWS Firewall Manager policies** pane, your policy should be listed. It will probably indicate **Pending** under the accounts headings and it will indicate the status of the **Automatic remediation** setting. The creation of a policy can take several minutes. After the **Pending** status is replaced with account counts, you can choose the policy name to explore the compliance status of the accounts and resources. For information, see Viewing compliance information for an AWS Firewall Manager policy

Continue to Step 3: (Optional) Authorizing the Shield Response Team (SRT).

Step 3: (Optional) Authorizing the Shield Response Team (SRT)

One of the benefits of AWS Shield Advanced is support from the Shield Response Team (SRT). When you experience a potential DDoS attack, you can contact the AWS Support Center. If necessary, the Support Center escalates your issue to the SRT. The SRT helps you analyze the suspicious activity and assists you in mitigating the issue. This mitigation often involves creating or updating AWS WAF rules and web ACLs in your account. The SRT can inspect your AWS WAF configuration and create or update AWS WAF rules and web ACLs for you, but the team needs your authorization to do so. We recommend that as part of setting up AWS Shield Advanced, you

proactively provide the SRT with the needed authorization. Providing authorization ahead of time helps prevent mitigation delays in the event of an actual attack.

You authorize and contact the SRT at the account level. That is, the account owner, not the Firewall Manager administrator, must perform the following steps to authorize the SRT to mitigate potential attacks. The Firewall Manager administrator can authorize the SRT only for accounts that they own. Likewise, only the account owner can contact the SRT for support.

> **Note**
>
> To use the services of the SRT, you must be subscribed to the Business Support plan or the Enterprise Support plan.

To authorize the SRT to mitigate potential attacks on your behalf, follow the instructions in Managed DDoS event response with Shield Response Team (SRT) support. You can change SRT access and permissions at any time by using the same steps.

Continue to Step 4: Configuring Amazon SNS notifications and Amazon CloudWatch alarms.

Step 4: Configuring Amazon SNS notifications and Amazon CloudWatch alarms

You can continue from this step without configuring Amazon SNS notifications or CloudWatch alarms. However, configuring these alarms and notifications significantly increases your visibility into possible DDoS events.

You can monitor your protected resources for potential DDoS activity using Amazon SNS. To receive notification of possible attacks, create an Amazon SNS topic for each Region.

> **Important**
>
> Amazon SNS notifications of potential DDoS activity are not sent in real time and can be delayed. Additionally, if you exceed the Shield Advanced quota of 1,000 protected resources for each resource type for each account, Firewall Manager performance constraints might prevent the successful delivery of DDoS attack notifications entirely. For more information, see AWS Shield Advanced quotas.
>
> To enable real-time notifications of potential DDoS activity, you can use a CloudWatch alarm. Your alarm must be based on the `DDoSDetected` metric from the account in which the protected resource exists.

To create an Amazon SNS topic in Firewall Manager (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, under **AWS FMS**, choose **Settings**.
3. Choose **Create new topic**.
4. Enter a topic name.
5. Enter an email address that the Amazon SNS messages will be sent to, and then choose **Add email address**.
6. Choose **Update SNS configuration**.

Configuring Amazon CloudWatch alarms

Shield Advanced records detection, mitigation, and top contributor metrics in CloudWatch that you can monitor. For more information, see AWS Shield Advanced metrics. CloudWatch incurs additional costs. For CloudWatch pricing, see Amazon CloudWatch Pricing.

To create a CloudWatch alarm, follow the instructions in Using Amazon CloudWatch Alarms. By default, Shield Advanced configures CloudWatch to alert you after just one indicator of a potential DDoS event. If needed, you can use the CloudWatch console to change this setting to alert you only after multiple indicators are detected.

> **Note**
>
> In addition to the alarms, you can also use a CloudWatch dashboard to monitor potential DDoS activity. The dashboard collects and processes raw data from Shield Advanced into readable, near real-time metrics. You can use statistics in Amazon CloudWatch to gain a perspective on how your web application or service is performing. For more information, see What is CloudWatch in the *Amazon CloudWatch User Guide*.

For instructions about creating a CloudWatch dashboard, see Monitoring with Amazon CloudWatch. For information about specific Shield Advanced metrics that you can add to your dashboard, see AWS Shield Advanced metrics.

When you've completed your Shield Advanced configuration, familiarize yourself with your options for viewing events at Visibility into DDoS events with Shield Advanced.

Setting up AWS Firewall Manager Amazon VPC security group policies

To use AWS Firewall Manager to enable Amazon VPC security groups across your organization, perform the following steps in sequence.

Topics

- Step 1: Completing the prerequisites
- Step 2: Creating a security group to use in your policy
- Step 3: Creating and applying a common security group policy

Step 1: Completing the prerequisites

There are several mandatory steps to prepare your account for AWS Firewall Manager. Those steps are described in AWS Firewall Manager prerequisites. Complete all the prerequisites before proceeding to Step 2: Creating a security group to use in your policy.

Step 2: Creating a security group to use in your policy

In this step, you create a security group that you could apply across your organization using Firewall Manager.

Note

For this tutorial, you won't apply your security group policy to the resources in your organization. You'll just create the policy and see what would happen if you applied the policy's security group to your resources. You do this by disabling automatic remediation on the policy.

If you already have a general security group defined, skip this step and go to Step 3: Creating and applying a common security group policy.

To create a security group to use in a Firewall Manager common security group policy

- Create a security group that you could apply to all accounts and resources in your organization, following the guidance under Security Groups for Your VPC in the Amazon VPC User Guide.

  For information on the security group rules options, see Security Group Rules Reference.

You are now ready to go to Step 3: Creating and applying a common security group policy.

Step 3: Creating and applying a common security group policy

After completing the prerequisites, you create an AWS Firewall Manager common security group policy. A common security group policy provides a centrally controlled security group for your entire AWS organization. It also defines the AWS accounts and resources that the security group applies to. In addition to common security group policies, Firewall Manager supports content audit security group policies, to manage the security group rules in use in your organization, and usage audit security group policies, to manage unused and redundant security groups. For more information, see Using security group policies in Firewall Manager to manage Amazon VPC security groups.

For this tutorial, you create a common security group policy and set its action to not automatically remediate. This allows you to see what effect the policy would have without making changes to your AWS organization.

To create a Firewall Manager common security group policy (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.

3. If you have not met the prerequisites, the console displays instructions about how to fix any issues. Follow the instructions, and then return to this step, to create a common security group policy.
4. Choose **Create policy**.
5. For **Policy type**, choose **Security group**.
6. For **Security group policy type**, choose **Common security groups**.
7. For **Region**, choose an AWS Region.
8. Choose **Next**.
9. For **Policy name**, enter a descriptive name.
10. **Policy rules** allow you to choose how the security groups in this policy are applied and maintained. For this tutorial, leave the options unchecked.
11. Choose **Add primary security group**, select the security group that you created for this tutorial, and choose **Add security group**.
12. For **Policy action**, choose **Identify resources that don't comply with the policy rules, but don't auto remediate.**
13. Choose **Next**.
14. **AWS accounts affected by this policy** allows you to narrow the scope of your policy by specifying accounts to include or exclude. For this tutorial, choose **Include all accounts under my organization.**
15. For **Resource type**, choose one or more types, according to the resources you have defined for your AWS organization.
16. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.
17. Choose **Next**.
18. For **Policy tags**, add any identifying tags that you want to add to the Firewall Manager policy resource. For more information about tags, see Working with Tag Editor.
19. Choose **Next**.

20. Review the new policy settings and return to any pages where you need to any adjustments.

    Check to be sure that **Policy actions** is set to **Identify resources that don't comply with the policy rules, but don't auto remediate.** This allows you to review the changes that your policy would make before you enable them.

21. When you are satisfied with the policy, choose **Create policy**.

    In the **AWS Firewall Manager policies** pane, your policy should be listed. It will probably indicate **Pending** under the accounts headings and it will indicate the status of the **Automatic remediation** setting. The creation of a policy can take several minutes. After the **Pending** status is replaced with account counts, you can choose the policy name to explore the compliance status of the accounts and resources. For information, see Viewing compliance information for an AWS Firewall Manager policy

22. When you are finished exploring, if you don't want to keep the policy you created for this tutorial, choose the policy name, choose **Delete**, choose **Clean up resources created by this policy.**, and finally choose **Delete**.

For more information about Firewall Manager security group policies, see Using security group policies in Firewall Manager to manage Amazon VPC security groups.

Setting up AWS Firewall Manager Amazon VPC network ACL policies

To use AWS Firewall Manager to enable network ACLs across your organization, perform the steps in this section in sequence.

For information about network ACLs, see Control traffic to subnets using network ACLs in the *Amazon VPC User Guide*.

Topics

- Step 1: Completing the prerequisites
- Step 2: Creating a network ACL policy

Step 1: Completing the prerequisites

There are several mandatory steps to prepare your account for AWS Firewall Manager. Those steps are described in AWS Firewall Manager prerequisites. Complete all the prerequisites before proceeding to Step 2: Creating a network ACL policy.

Step 2: Creating a network ACL policy

After completing the prerequisites, you create a Firewall Manager network ACL policy. A network ACL policy provides a centrally controlled network ACL definition for your entire AWS organization. It also defines the AWS accounts and subnets that the network ACL applies to.

For information about Firewall Manager network ACL policies, see Network ACL policies.

For general information about Firewall Manager network ACL policies, see Network ACL policies.

> **Note**
>
> For this tutorial, you won't apply your network ACL policy to the subnets in your organization. You'll just create the policy and see what would happen if you applied the policy's network ACL to your subnets. You do this by disabling automatic remediation on the policy.

To create a Firewall Manager network ACL policy (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. If you have not met the prerequisites, the console displays instructions about how to fix any issues. Follow the instructions, and then return to this step, to create a network ACL policy.
4. Choose **Create policy**.
5. For **Region**, choose an AWS Region.
6. For **Policy type**, choose **Network ACL**.
7. Choose **Next**.

8. For **Policy name**, enter a descriptive name.
9. For **Network ACL policy rules**, define the first and last rules for both inbound and outbound traffic.

   You define network ACL rules in Firewall Manager similar to how you define them through Amazon VPC. The only difference is that, instead of assigning rule numbers yourself, you assign the order to run each set of rules, and then Firewall Manager assigns the numbers for you when you save the policy. You can define up to 5 inbound rules, divided in any way between first and last, and you can define up to 5 outbound rules.

   For guidance specifying network ACL rules, see Add and delete network ACL rules in the *Amazon VPC User Guide*.

   The rules that you define in the Firewall Manager policy specify the minimum rule configuration that a network ACL must have to be compliant with the network ACL policy. For example, a network ACL's inbound rules cannot be compliant with the policy unless they start with as the policy's inbound first rules, in the same order as they're specified in the policy. For more information, see Network ACL policies.
10. For **Policy action**, choose **Identify resources that don’t comply with the policy rules, but don’t auto remediate.**
11. Choose **Next**.
12. **AWS accounts affected by this policy** allows you to narrow the scope of your policy by specifying accounts to include or exclude. For this tutorial, choose **Include all accounts under my organization.**

    The **Resource type** for a network ACL policy is always subnet.
13. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.
14. Choose **Next**.
15. For **Policy tags**, add any identifying tags that you want to add to the Firewall Manager policy resource. For more information about tags, see Working with Tag Editor.

16. Choose **Next**.
17. Review the new policy settings and return to any pages where you need to any adjustments.

    Check to be sure that **Policy actions** is set to **Identify resources that don't comply with the policy rules, but don't auto remediate.** This allows you to review the changes that your policy would make before you enable them.
18. When you are satisfied with the policy, choose **Create policy**.

    In the **AWS Firewall Manager policies** pane, your policy should be listed. It will probably indicate **Pending** under the accounts headings and it will indicate the status of the **Automatic remediation** setting. The creation of a policy can take several minutes. After the **Pending** status is replaced with account counts, you can choose the policy name to explore the compliance status of the accounts and resources. For information, see Viewing compliance information for an AWS Firewall Manager policy
19. When you are finished exploring, if you don't want to keep the policy that you created for this tutorial, choose the policy name, choose **Delete**, choose **Clean up resources created by this policy.**, and finally choose **Delete**.

For more information about Firewall Manager network ACL policies, see Network ACL policies.

Setting up AWS Firewall Manager AWS Network Firewall policies

To use AWS Firewall Manager to enable an AWS Network Firewall firewall across your organization, perform the following steps in sequence. For information about Firewall Manager Network Firewall policies, see Using AWS Network Firewall policies in Firewall Manager.

Topics

- Step 1: Completing the prerequisites
- Step 2: Creating a Network Firewall rule group to use in your policy
- Step 3: Creating and applying a Network Firewall policy

Step 1: Completing the prerequisites

There are several mandatory steps to prepare your account for AWS Firewall Manager. Those steps are described in AWS Firewall Manager prerequisites. Complete all the prerequisites before proceeding to the next step.

Step 2: Creating a Network Firewall rule group to use in your policy

To follow this tutorial, you should be familiar with AWS Network Firewall and know how to configure its rule groups and firewall policies.

You must have at least one rule group in Network Firewall that will be used in your AWS Firewall Manager policy. If you haven't already created a rule group in Network Firewall, do so now. For information about using Network Firewall, see the AWS Network Firewall Developer Guide.

Step 3: Creating and applying a Network Firewall policy

After completing the prerequisites, you create an AWS Firewall Manager Network Firewall policy. A Network Firewall policy provides a centrally controlled AWS Network Firewall firewall for your entire AWS organization. It also defines the AWS accounts and resources that the firewall applies to.

For more information about how Firewall Manager manages your Network Firewall policies, see Using AWS Network Firewall policies in Firewall Manager.

To create a Firewall Manager Network Firewall policy (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. If you haven't met the prerequisites, the console displays instructions about how to fix any issues. Follow the instructions, and then return to this step, to create a Network Firewall policy.
4. Choose **Create security policy**.
5. For **Policy type**, choose **AWS Network Firewall**.
6. For **Region**, choose an AWS Region.
7. Choose **Next**.

8. For **Policy name**, enter a descriptive name.
9. The policy configuration allows you to define the firewall policy. This is the same process as the one you use in the AWS Network Firewall console. You add the rule groups that you want to use in your policy and provide the default stateless actions. For this tutorial, configure this policy as you would a firewall policy in Network Firewall.

> **Note**
>
> Auto remediation happens automatically for AWS Firewall Manager Network Firewall policies, so you won't see an option to choose not to auto remediate here.

10. Choose **Next**.
11. For **Firewall endpoints**, choose **Multiple firewall endpoints**. This option provides high availability for your firewall. When you create the policy, Firewall Manager creates a firewall subnet in each Availability Zone where you have public subnets to protect.
12. For **AWS Network Firewall route configuration**, choose **Monitor** to have Firewall Manager monitor your VPCs for route configuration violations and alert you with remediation suggestions to help you to bring the routes into compliance. Optionally, if you don't want to have your route configurations monitored by Firewall Manager and receive these alerts, choose **Off**.

> **Note**
>
> Monitoring provides you with details about non-compliant resources due to faulty route configuration, and suggests remediation actions from the Firewall Manager `GetViolationDetails` API. For example, Network Firewall alerts you if traffic is not routed through the firewall endpoints that are created by your policy.

> **Warning**
>
> If you choose **Monitor**, you can't change it to **Off** in the future for the same policy. You must create a new policy.

13. For **Traffic type**, select **Add to firewall policy** to route traffic through the internet gateway.

14. **AWS accounts affected by this policy** allows you to narrow the scope of your policy by specifying accounts to include or exclude. For this tutorial, choose **Include all accounts under my organization**.

    The **Resource type** for a Network Firewall policy is always **VPC**.

15. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.

16. Choose **Next**.

17. For **Policy tags**, add any identifying tags that you want to add to the Firewall Manager policy resource. For more information about tags, see Working with Tag Editor.

18. Choose **Next**.

19. Review the new policy settings and return to any pages where you need to any adjustments.

    Check to be sure that **Policy actions** is set to **Identify resources that don't comply with the policy rules, but don't auto remediate.** This allows you to review the changes that your policy would make before you enable them.

20. When you are satisfied with the policy, choose **Create policy**.

    In the **AWS Firewall Manager policies** pane, your policy should be listed. It will probably indicate **Pending** under the accounts headings and it will indicate the status of the **Automatic remediation** setting. The creation of a policy can take several minutes. After the **Pending** status is replaced with account counts, you can choose the policy name to explore the compliance status of the accounts and resources. For information, see Viewing compliance information for an AWS Firewall Manager policy

21. When you are finished exploring, if you don't want to keep the policy that you created for this tutorial, choose the policy name, choose **Delete**, choose **Clean up resources created by this policy.**, and finally choose **Delete**.

For more information about Firewall Manager Network Firewall policies, see Using AWS Network Firewall policies in Firewall Manager.

Setting up AWS Firewall Manager DNS Firewall policies

To use AWS Firewall Manager to enable Amazon Route 53 Resolver DNS Firewall across your organization, perform the following steps in sequence. For information about Firewall Manager DNS Firewall policies, see Using Amazon Route 53 Resolver DNS Firewall policies in Firewall Manager.

Topics

- Step 1: Completing the prerequisites
- Step 2: Creating your DNS Firewall rule groups to use in your policy
- Step 3: Creating and applying a DNS Firewall policy

Step 1: Completing the prerequisites

There are several mandatory steps to prepare your account for AWS Firewall Manager. Those steps are described in AWS Firewall Manager prerequisites. Complete all the prerequisites before proceeding to the next step.

Step 2: Creating your DNS Firewall rule groups to use in your policy

To follow this tutorial, you should be familiar with Amazon Route 53 Resolver DNS Firewall and know how to configure its rule groups.

You must have least one rule group in DNS Firewall that will be used in your AWS Firewall Manager policy. If you haven't already created a rule group in DNS Firewall, do so now. For information about using DNS Firewall, see Amazon Route 53 Resolver DNS Firewall in the Amazon Route 53 Developer Guide.

Step 3: Creating and applying a DNS Firewall policy

After completing the prerequisites, you create an AWS Firewall Manager DNS Firewall policy. A DNS Firewall policy provides a set of centrally controlled DNS Firewall rule group associations for your entire AWS organization. It also defines the AWS accounts and resources that the firewall applies to.

For more information about how Firewall Manager manages your DNS Firewall rule group associations, see Using Amazon Route 53 Resolver DNS Firewall policies in Firewall Manager.

To create a Firewall Manager DNS Firewall policy (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.
2. In the navigation pane, choose **Security policies**.
3. If you haven't met the prerequisites, the console displays instructions about how to fix any issues. Follow the instructions, and then return to this step, to create a DNS Firewall policy.
4. Choose **Create security policy**.
5. For **Policy type**, choose **Amazon Route 53 Resolver DNS Firewall**.
6. For **Region**, choose an AWS Region.
7. Choose **Next**.
8. For **Policy name**, enter a descriptive name.
9. The policy configuration allows you to define the DNS Firewall rule group associations that you want to manage from Firewall Manager. You add the rule groups that you want to use in your policy. You can define an association to evaluate first for your VPCs and one to evaluate last. For this tutorial, add one or two rule group associations, depending on your needs.
10. Choose **Next**.
11. **AWS accounts affected by this policy** allows you to narrow the scope of your policy by specifying accounts to include or exclude. For this tutorial, choose **Include all accounts under my organization**.

    The **Resource type** for a DNS Firewall policy is always **VPC**.
12. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.
13. Choose **Next**.
14. For **Policy tags**, add any identifying tags that you want to add to the Firewall Manager policy resource. For more information about tags, see Working with Tag Editor.

15. Choose **Next**.
16. Review the new policy settings and return to any pages where you need to any adjustments.

    Check to be sure that **Policy actions** is set to **Identify resources that don't comply with the policy rules, but don't auto remediate.** This allows you to review the changes that your policy would make before you enable them.
17. When you are satisfied with the policy, choose **Create policy**.

    In the **AWS Firewall Manager policies** pane, your policy should be listed. It will probably indicate **Pending** under the accounts headings and it will indicate the status of the **Automatic remediation** setting. The creation of a policy can take several minutes. After the **Pending** status is replaced with account counts, you can choose the policy name to explore the compliance status of the accounts and resources. For information, see Viewing compliance information for an AWS Firewall Manager policy
18. When you are finished exploring, if you don't want to keep the policy that you created for this tutorial, choose the policy name, choose **Delete**, choose **Clean up resources created by this policy.**, and finally choose **Delete**.

For more information about Firewall Manager DNS Firewall policies, see Using Amazon Route 53 Resolver DNS Firewall policies in Firewall Manager.

Setting up AWS Firewall Manager Palo Alto Networks Cloud Next Generation Firewall policies

To use AWS Firewall Manager to enable Palo Alto Networks Cloud Next Generation Firewall (NGFW) policies, perform the following steps in sequence. For information about Palo Alto Networks Cloud NGFW policies, see Using Palo Alto Networks Cloud NGFW policies for Firewall Manager.

Topics

- Step 1: Completing the general prerequisites
- Step 2: Completing the Palo Alto Networks Cloud NGFW policy prerequisites
- Step 3: Creating and applying a Palo Alto Networks Cloud NGFW policy

Step 1: Completing the general prerequisites

There are several mandatory steps to prepare your account for AWS Firewall Manager. Those steps are described in AWS Firewall Manager prerequisites. Complete all the prerequisites before proceeding to the next step.

Step 2: Completing the Palo Alto Networks Cloud NGFW policy prerequisites

There are a couple of additional mandatory steps that you must complete in order to use Palo Alto Networks Cloud NGFW policies. Those steps are described in Palo Alto Networks Cloud Next Generation Firewall policy prerequisites. Complete all the prerequisites before proceeding to the next step.

Step 3: Creating and applying a Palo Alto Networks Cloud NGFW policy

After completing the prerequisites, you create an AWS Firewall Manager Palo Alto Networks Cloud NGFW policy.

For more information about Firewall Manager policies for Palo Alto Networks Cloud NGFW, see Using Palo Alto Networks Cloud NGFW policies for Firewall Manager.

To create a Firewall Manager policy for Palo Alto Networks Cloud NGFW (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. Choose **Create policy**.
4. For **Policy type**, choose **Palo Alto Networks Cloud NGFW**. If you haven't already subscribed to the Palo Alto Networks Cloud NGFW service in the AWS Marketplace, you'll need to do that first. To subscribe in the AWS Marketplace, choose **View AWS Marketplace details**.

5. For **Deployment model**, choose either the **Distributed model** or **Centralized model**. The deployment model determines how Firewall Manager manages endpoints for the policy. With the distributed model, Firewall Manager maintains firewall endpoints in each VPC that's within policy scope. With the centralized model, Firewall Manager maintains a single endpoint in an inspection VPC.
6. For **Region**, choose an AWS Region. To protect resources in multiple Regions, you must create separate policies for each Region.
7. Choose **Next**.
8. For **Policy name**, enter a descriptive name.
9. In the policy configuration, choose the Palo Alto Networks Cloud NGFW firewall policy to associate with this policy. The list of Palo Alto Networks Cloud NGFW firewall policies contains all of the Palo Alto Networks Cloud NGFW firewall policies that are associated with your Palo Alto Networks Cloud NGFW tenant. For information about creating and managing Palo Alto Networks Cloud NGFW firewall policies, see the *Deploy Palo Alto Networks Cloud NGFW for AWS with the AWS Firewall Manager* topic in the *Palo Alto Networks Cloud NGFW for AWS deployment guide*.
10. For **Palo Alto Networks Cloud NGFW logging - optional**, optionally choose which Palo Alto Networks Cloud NGFW log type(s) to log for your policy. For information about Palo Alto Networks Cloud NGFW log types, see Configure Logging for Palo Alto Networks Cloud NGFW on AWS in the *Palo Alto Networks Cloud NGFW for AWS deployment guide*.

    For **log destination**, specify when Firewall Manager should write logs to.
11. Choose **Next**.
12. Under **Configure third-party firewall endpoint** do one of the following, depending on whether you're using the distributed or centralized deployment model to create your firewall endpoints:

    - If you're using the distributed deployment model for this policy, under **Availability Zones**, select which Availability Zones to create firewall endpoints in. You can select Availability Zones by **Availability Zone name** or by **Availability Zone ID**.
    - If you're using the centralized deployment model for this policy, in **AWS Firewall Manager endpoint configuration** under **Inspection VPC configuration**, enter the AWS account ID of the owner of the inspection VPC, and the VPC ID of the inspection VPC.
        - Under **Availability Zones**, select which Availability Zones to create firewall endpoints in. You can select Availability Zones by **Availability Zone name** or by **Availability Zone ID**.

13. Choose **Next**.
14. For **Policy scope**, under **AWS accounts this policy applies to**, choose the option as follows:

    - If you want to apply the policy to all accounts in your organization, leave the default selection, **Include all accounts under my AWS organization**.
    - If you want to apply the policy only to specific accounts or accounts that are in specific AWS Organizations organizational units (OUs), choose **Include only the specified accounts and organizational units**, and then add the accounts and OUs that you want to include. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.
    - If you want to apply the policy to all but a specific set of accounts or AWS Organizations organizational units (OUs), choose **Exclude the specified accounts and organizational units, and include all others**, and then add the accounts and OUs that you want to exclude. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

    You can only choose one of the options.

    After you apply the policy, Firewall Manager automatically evaluates any new accounts against your settings. For example, if you include only specific accounts, Firewall Manager doesn't apply the policy to any new accounts. As another example, if you include an OU, when you add an account to the OU or to any of its child OUs, Firewall Manager automatically applies the policy to the new account.

    The **Resource type** for Network Firewall policies is **VPC**.
15. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.
16. For **Grant cross-account access**, choose **Download CloudFormation template**. This downloads an CloudFormation template that you can use to create an CloudFormation stack. This stack creates an AWS Identity and Access Management role that grants Firewall Manager cross-

account permissions to manage Palo Alto Networks Cloud NGFW resources. For information about stacks, see Working with stacks in the *CloudFormation User Guide*.

17. Choose **Next**.
18. For **Policy tags**, add any identifying tags that you want to add to the Firewall Manager policy resource. For more information about tags, see Working with Tag Editor.
19. Choose **Next**.
20. Review the new policy settings and return to any pages where you need to any adjustments.

    Check to be sure that **Policy actions** is set to **Identify resources that don't comply with the policy rules, but don't auto remediate.** This allows you to review the changes that your policy would make before you enable them.
21. When you are satisfied with the policy, choose **Create policy**.

    In the **AWS Firewall Manager policies** pane, your policy should be listed. It will probably indicate **Pending** under the accounts headings and it will indicate the status of the **Automatic remediation** setting. The creation of a policy can take several minutes. After the **Pending** status is replaced with account counts, you can choose the policy name to explore the compliance status of the accounts and resources. For information, see Viewing compliance information for an AWS Firewall Manager policy

For more information about Firewall Manager Palo Alto Networks Cloud NGFW policies, see Using Palo Alto Networks Cloud NGFW policies for Firewall Manager.

Setting up AWS Firewall Manager Fortigate CNF policies

Fortigate Cloud Native Firewall (CNF) as a Service is a third-party firewall service that you can use for your AWS Firewall Manager policies. With Fortigate CNF for Firewall Manager, you can create and centrally deploy Fortigate CNF resources and policy sets across all of your AWS accounts. To use AWS Firewall Manager to enable Fortigate CNF policies, perform the following steps in sequence. For more information about Fortigate CNF policies, see Using Fortigate Cloud Native Firewall (CNF) as a Service policies for Firewall Manager.

Topics

- Step 1: Completing the general prerequisites
- Step 2: Completing the Fortigate CNF policy prerequisites
- Step 3: Creating and applying a Fortigate CNF policy

Step 1: Completing the general prerequisites

There are several mandatory steps to prepare your account for AWS Firewall Manager. Those steps are described in AWS Firewall Manager prerequisites. Complete all the prerequisites before proceeding to the next step.

Step 2: Completing the Fortigate CNF policy prerequisites

There are additional mandatory steps that you must complete in order to use Fortigate CNF policies. Those steps are described in Fortigate Cloud Native Firewall (CNF) as a Service policy prerequisites. Complete all the prerequisites before proceeding to the next step.

Step 3: Creating and applying a Fortigate CNF policy

After completing the prerequisites, you create an AWS Firewall Manager Fortigate CNF policy.

For more information about Firewall Manager policies for Fortigate CNF, see Using Fortigate Cloud Native Firewall (CNF) as a Service policies for Firewall Manager.

To create a Firewall Manager policy for Fortigate CNF (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. Choose **Create policy**.
4. For **Policy type**, choose Fortigate CNF. If you haven't already subscribed to the Fortigate CNF service in the AWS Marketplace, you'll need to do that first. To subscribe in the AWS Marketplace, choose **View AWS Marketplace details**.
5. For **Deployment model**, choose either the **Distributed model** or **Centralized model**. The deployment model determines how Firewall Manager manages endpoints for the policy. With the distributed model, Firewall Manager maintains firewall endpoints in each VPC that's within

policy scope. With the centralized model, Firewall Manager maintains a single endpoint in an inspection VPC.

6. For **Region**, choose an AWS Region. To protect resources in multiple Regions, you must create separate policies for each Region.
7. Choose **Next**.
8.
9. In the policy configuration, choose the Fortigate CNF firewall policy to associate with this policy. The list of Fortigate CNF firewall policies contains all of the Fortigate CNF firewall policies that are associated with your Fortigate CNF tenant. For information about creating and managing Fortigate CNF firewall policies, see the Fortigate CNF documentation.
10. Choose **Next**.
11. Under **Configure third-party firewall endpoint** do one of the following, depending on whether you're using the distributed or centralized deployment model to create your firewall endpoints:

    - If you're using the distributed deployment model for this policy, under **Availability Zones**, select which Availability Zones to create firewall endpoints in. You can select Availability Zones by **Availability Zone name** or by **Availability Zone ID**.
    - If you're using the centralized deployment model for this policy, in **AWS Firewall Manager endpoint configuration** under **Inspection VPC configuration**, enter the AWS account ID of the owner of the inspection VPC, and the VPC ID of the inspection VPC.
      - Under **Availability Zones**, select which Availability Zones to create firewall endpoints in. You can select Availability Zones by **Availability Zone name** or by **Availability Zone ID**.
12. Choose **Next**.
13. For **Policy scope**, under **AWS accounts this policy applies to**, choose the option as follows:

    - If you want to apply the policy to all accounts in your organization, leave the default selection, **Include all accounts under my AWS organization**.
    - If you want to apply the policy only to specific accounts or accounts that are in specific AWS Organizations organizational units (OUs), choose **Include only the specified accounts and organizational units**, and then add the accounts and OUs that you want to include. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.
    - If you want to apply the policy to all but a specific set of accounts or AWS Organizations organizational units (OUs), choose **Exclude the specified accounts and organizational**

units, and include all others, and then add the accounts and OUs that you want to exclude. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

You can only choose one of the options.

After you apply the policy, Firewall Manager automatically evaluates any new accounts against your settings. For example, if you include only specific accounts, Firewall Manager doesn't apply the policy to any new accounts. As another example, if you include an OU, when you add an account to the OU or to any of its child OUs, Firewall Manager automatically applies the policy to the new account.

The **Resource type** for Fortigate CNF policies is **VPC**.

14. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.

15. For **Grant cross-account access**, choose **Download CloudFormation template**. This downloads an CloudFormation template that you can use to create an CloudFormation stack. This stack creates an AWS Identity and Access Management role that grants Firewall Manager cross-account permissions to manage Fortigate CNF resources. For information about stacks, see Working with stacks in the *CloudFormation User Guide*. To create a stack, you'll need the account ID from the Fortigate CNF portal.
16. Choose **Next**.
17. For **Policy tags**, add any identifying tags that you want to add to the Firewall Manager policy resource. For more information about tags, see Working with Tag Editor.
18. Choose **Next**.
19. Review the new policy settings and return to any pages where you need to any adjustments.

    Check to be sure that **Policy actions** is set to **Identify resources that don’t comply with the policy rules, but don’t auto remediate.** This allows you to review the changes that your policy would make before you enable them.

20. When you are satisfied with the policy, choose **Create policy**.

    In the **AWS Firewall Manager policies** pane, your policy should be listed. It will probably indicate **Pending** under the accounts headings and it will indicate the status of the **Automatic remediation** setting. The creation of a policy can take several minutes. After the **Pending** status is replaced with account counts, you can choose the policy name to explore the compliance status of the accounts and resources. For information, see Viewing compliance information for an AWS Firewall Manager policy

For more information about Firewall Manager Fortigate CNF policies, see Using Fortigate Cloud Native Firewall (CNF) as a Service policies for Firewall Manager.

Using AWS Firewall Manager policies

AWS Firewall Manager provides the following types of policies. For each policy type, you define the:

- **AWS WAF policy** – Firewall Manager supports AWS WAF and AWS WAF Classic policies. For both versions, you define which resources are protected by the policy.
  - The AWS WAF policy type takes sets of rule groups to run first and last in the web ACL. Then, in the accounts where you apply the web ACL, the account owner can add rules and rule groups to run in between the two sets.
  - The AWS WAF Classic policy type takes a single rule group to run in the web ACL.
- **Shield Advanced policy** – This policy type applies Shield Advanced protections throughout your organization for the resource types that you specify.
- **Amazon VPC security group policy** – This policy type gives you control over security groups that are in use throughout your organization and lets you enforce a baseline set of rules across your organization.
- **Amazon VPC network access control list (ACL) policy** – This policy type gives you control over network ACLs that are in use throughout your organization and lets you enforce a baseline set of network ACLs across your organization.
- **Network Firewall policy** – This policy type applies AWS Network Firewall protection to your organization's VPCs.
- **Amazon Route 53 Resolver DNS Firewall policy** – This policy applies DNS Firewall protections to your organization's VPCs.

- **Third-party firewall policy** – This policy type applies third-party firewall protections. Third-party firewalls are available by subscription through the AWS Marketplace console at AWS Marketplace.
  - **Palo Alto Networks Cloud NGFW policy** – This policy type applies Palo Alto Networks Cloud Next Generation Firewall (NGFW) protections and Palo Alto Networks Cloud NGFW rulestacks to your organization's VPCs.
  - **Fortigate Cloud Native Firewall (CNF) as a Service policy** – This policy type applies Fortigate Cloud Native Firewall (CNF) as a Service protections. Fortigate CNF is a cloud-centered solution that blocks Zero-Day threats and secures cloud infrastructures with industry-leading advanced threat prevention, smart web application firewalls (WAF), and API protection.

A Firewall Manager policy is specific to the individual policy type. If you want to enforce multiple policy types across accounts, you can create multiple policies. You can create more than one policy for each type.

If you add a new account to an organization that you created with AWS Organizations, Firewall Manager automatically applies the policy to the resources in that account that are within scope of the policy.

General settings for AWS Firewall Manager policies

AWS Firewall Manager managed policies have some common settings and behaviors. For all, you specify a name and define the scope of the policy, and you can use resource tagging to control policy scope. You can choose to view the accounts and resources that are out of compliance without taking corrective action or to automatically remediate noncompliant resources.

For information about policy scope, see Using the AWS Firewall Manager policy scope.

Creating an AWS Firewall Manager policy

The steps for creating a policy vary between the different policy types. Make sure to use the procedure for the type of policy that you need.

> **Important**
>
> AWS Firewall Manager doesn't support Amazon Route 53 or AWS Global Accelerator. If you want to protect these resources with Shield Advanced, you can't use a Firewall Manager

policy. Instead, follow the instructions in Adding AWS Shield Advanced protection to AWS resources.

Topics

- Creating an AWS Firewall Manager policy for AWS WAF
- Creating an AWS Firewall Manager policy for AWS WAF Classic
- Creating an AWS Firewall Manager policy for AWS Shield Advanced
- Creating an AWS Firewall Manager common security group policy
- Creating an AWS Firewall Manager content audit security group policy
- Creating an AWS Firewall Manager usage audit security group policy
- Creating an AWS Firewall Manager network ACL policy
- Creating an AWS Firewall Manager policy for AWS Network Firewall
- Creating an AWS Firewall Manager policy for Amazon Route 53 Resolver DNS Firewall
- Creating an AWS Firewall Manager policy for Palo Alto Networks Cloud NGFW
- Creating an AWS Firewall Manager policy for Fortigate Cloud Native Firewall (CNF) as a Service

Creating an AWS Firewall Manager policy for AWS WAF

In a Firewall Manager AWS WAF policy, you can use managed rule groups, which AWS and AWS Marketplace sellers create and maintain for you. You can also create and use your own rule groups. For more information about rule groups, see AWS WAF rule groups.

If you want to use your own rule groups, create those before you create your Firewall Manager AWS WAF policy. For guidance, see Managing your own rule groups. To use an individual custom rule, you must define your own rule group, define your rule within that, and then use the rule group in your policy.

For information about Firewall Manager AWS WAF policies, see Using AWS WAF policies with Firewall Manager.

To create a Firewall Manager policy for AWS WAF (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/

fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

> **Note**
>
> For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. Choose **Create policy**.
4. For **Policy type**, choose **AWS WAF**.
5. For **Region**, choose an AWS Region. To protect Amazon CloudFront distributions, choose **Global**.

   To protect resources in multiple Regions (other than CloudFront distributions), you must create separate Firewall Manager policies for each Region.
6. Choose **Next**.
7. For **Policy name**, enter a descriptive name. Firewall Manager includes the policy name in the names of the web ACLs that it manages. The web ACL names have `FMManagedWebACLV2-` followed by the policy name that you enter here, `-`, and the web ACL creation timestamp, in UTC milliseconds. For example, `FMManagedWebACLV2-MyWAFPolicyName-1621880374078`.
8. For **Web request body inspection**, optionally change the body size limit. For information about body inspection size limits, including pricing considerations, see Considerations for managing body inspection in AWS WAF in the *AWS WAF Developer Guide*.
9. Under **Policy rules**, add the rule groups that you want AWS WAF to evaluate first and last in the web ACL. To use AWS WAF managed rule group versioning, toggle **Enable versioning**. The individual account managers can add rules and rule groups in between your first rule groups and your last rule groups. For more information about using AWS WAF rule groups in Firewall Manager policies for AWS WAF, see Using AWS WAF policies with Firewall Manager.

   (Optional) To customize how your web ACL uses the rule group, choose **Edit**. The following are common customization settings:

   - For managed rule groups, override the rule actions for some or all rules. If you don't define an override action for a rule, the evaluation uses the rule action that's defined inside the rule

group. For information about this option, see Overriding rule group actions in AWS WAF in the *AWS WAF Developer Guide*.

- Some managed rule groups require you to provide additional configuration. See the documentation from your managed rule group provider. For information specific to the AWS Managed Rules rule groups, see AWS Managed Rules for AWS WAF in the *AWS WAF Developer Guide*.

When you're finished with your settings, choose **Save rule**.

10. Set the default action for the web ACL. This is the action that AWS WAF takes when a web request doesn't match any of the rules in the web ACL. You can add custom headers with the **Allow** action, or custom responses for the **Block** action. For more information about default web ACL actions, see Setting the protection pack (web ACL) default action in AWS WAF. For information about setting custom web requests and responses, see Customized web requests and responses in AWS WAF.
11. For **Logging configuration**, choose **Enable logging** to turn on logging. Logging provides detailed information about traffic that is analyzed by your web ACL. Choose the **Logging destination**, and then choose the logging destination that you configured. You must choose a logging destination whose name begins with `aws-waf-logs-`. For information about configuring an AWS WAF logging destination, see Using AWS WAF policies with Firewall Manager.
12. (Optional) If you don't want certain fields and their values included in the logs, redact those fields. Choose the field to redact, and then choose **Add**. Repeat as necessary to redact additional fields. The redacted fields appear as `REDACTED` in the logs. For example, if you redact the **URI** field, the **URI** field in the logs will be `REDACTED`.
13. (Optional) If you don't want to send all requests to the logs, add your filtering criteria and behavior. Under **Filter logs**, for each filter that you want to apply, choose **Add filter**, then choose your filtering criteria and specify whether you want to keep or drop requests that match the criteria. When you finish adding filters, if needed, modify the **Default logging behavior**. For more information, see Finding your protection pack (web ACL) records in the *AWS WAF Developer Guide*.
14. You can define a **Token domain list** to enable token sharing between protected applications. Tokens are used by the CAPTCHA and Challenge actions and by the application integration SDKs that you implement when you use the AWS Managed Rules rule groups for AWS WAF Fraud Control account takeover prevention (ATP) and AWS WAF Bot Control.

Public suffixes aren't allowed. For example, you can't use `gov.au` or `co.uk` as a token domain.

By default, AWS WAF accepts tokens only for the domain of the protected resource. If you add token domains in this list, AWS WAF accepts tokens for all domains in the list and for the domain of the associated resource. For more information, see AWS WAF protection pack (web ACL) token domain list configuration in the *AWS WAF Developer Guide*.

You can only change the web ACL's CAPTCHA and challenge **immunity times** when you edit an existing web ACL. You can find these settings under the Firewall Manager **Policy details** page. For information about these settings, see Setting timestamp expiration and token immunity times in AWS WAF. If you update the **Association config**, **CAPTCHA**, **Challenge**, or **Token domain list** settings in an existing policy, Firewall Manager will overwrite the your local web ACLs with the new values. However, if you don't update the policy's **Association config**, **CAPTCHA**, **Challenge**, or **Token domain list** settings, then the values in your local web ACLs will remain unchanged. For information about this option, see CAPTCHA and Challenge in AWS WAF in the *AWS WAF Developer Guide*.

15. Under **Web ACL management**, choose how Firewall Manager manages web ACL creation and clean up.

    a. For **Manage unassociated web ACLs**, choose whether Firewall Manager manages unassociated web ACLs. With this option, Firewall Manager creates web ACLs for the accounts within policy scope only if the web ACLs will be used by at least one resource. When an account comes into policy scope, Firewall Manager automatically creates a web ACL in the account if at least one resource will use it.

    When you enable this option, Firewall Manager performs a one-time cleanup of unassociated web ACLs in your account. The cleanup process can take several hours. If a resource leaves policy scope after Firewall Manager creates a web ACL, Firewall Manager disassociates the resource from the web ACL, but doesn't clean up the unassociated web ACL. Firewall Manager only cleans up unassociated web ACLs when you first enable management of unassociated web ACLs in the policy.

    b. For **Web ACL source**, specify whether to create all new web ACLs for in-scope resources or to retrofit existing web ACLs where possible. Firewall Manager can retrofit web ACLs that are owned by in-scope accounts.

    The default behavior is to create all new web ACLs. If you choose this, all web ACLs managed by Firewall Manager will have names that begin with `FMManagedWebACLV2`.

If you choose to retrofit existing web ACLs, the retrofitted web ACLs will have their original names and the ones created by Firewall Manager will have names that begin with `FMManagedWebACLV2`.

16. For **Policy action**, if you want to create a web ACL in each applicable account within the organization, but not apply the web ACL to any resources yet, choose **Identify resources that don't comply with the policy rules, but don't auto remediate** and don't choose **Manage unassociated web ACLs**. You can change these options later.

    If instead you want to automatically apply the policy to existing in-scope resources, choose **Auto remediate any noncompliant resources**. If **Manage unassociated web ACLs** is disabled, the **Auto remediate any noncompliant resources** option creates a web ACL in each applicable account within the organization and associates the web ACL with the resources in the accounts. If **Manage unassociated web ACLs** is enabled, the **Auto remediate any noncompliant resources** option only creates and associates a web ACL in accounts that have resources eligible for association to the web ACL.

    When you choose **Auto remediate any noncompliant resources**, you can also choose to remove existing web ACL associations from in-scope resources, for the web ACLs that aren't managed by another active Firewall Manager policy. If you choose this option, Firewall Manager first associates the policy's web ACL with the resources, and then removes the prior associations. If a resource has an association with another web ACL that's managed by a different active Firewall Manager policy, this choice doesn't affect that association.

17. Choose **Next**.

18. For **AWS accounts this policy applies to**, choose the option as follows:

    - If you want to apply the policy to all accounts in your organization, leave the default selection, **Include all accounts under my AWS organization**.
    - If you want to apply the policy only to specific accounts or accounts that are in specific AWS Organizations organizational units (OUs), choose **Include only the specified accounts and organizational units**, and then add the accounts and OUs that you want to include. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.
    - If you want to apply the policy to all but a specific set of accounts or AWS Organizations organizational units (OUs), choose **Exclude the specified accounts and organizational units, and include all others**, and then add the accounts and OUs that you want to exclude.

Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

You can only choose one of the options.

After you apply the policy, Firewall Manager automatically evaluates any new accounts against your settings. For example, if you include only specific accounts, Firewall Manager doesn't apply the policy to any new accounts. As another example, if you include an OU, when you add an account to the OU or to any of its child OUs, Firewall Manager automatically applies the policy to the new account.

19. For **Resource type**, choose the types of resources that you want to protect.
20. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.
21. Choose **Next**.
22. For **Policy tags**, add any identifying tags that you want to add to the Firewall Manager policy resource. For more information about tags, see Working with Tag Editor.
23. Choose **Next**.
24. Review the new policy settings and return to any pages where you need to any adjustments.

    When you are satisfied with the policy, choose **Create policy**. In the **AWS Firewall Manager policies** pane, your policy should be listed. It will probably indicate **Pending** under the accounts headings and it will indicate the status of the **Automatic remediation** setting. The creation of a policy can take several minutes. After the **Pending** status is replaced with account counts, you can choose the policy name to explore the compliance status of the accounts and resources. For information, see Viewing compliance information for an AWS Firewall Manager policy

Creating an AWS Firewall Manager policy for AWS WAF Classic

To create a Firewall Manager policy for AWS WAF Classic (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. Choose **Create policy**.
4. For **Policy type**, choose **AWS WAF Classic**.
5. If you already created the AWS WAF Classic rule group that you want to add to the policy, choose **Create an AWS Firewall Manager policy and add existing rule groups**. If you want to create a new rule group, choose **Create a Firewall Manager policy and add a new rule group**.
6. For **Region**, choose an AWS Region. To protect Amazon CloudFront resources, choose **Global**.

   To protect resources in multiple Regions (other than CloudFront resources), you must create separate Firewall Manager policies for each Region.
7. Choose **Next**.
8. If you are creating a rule group, follow the instructions in Creating an AWS WAF Classic rule group. After you create the rule group, continue with the following steps.
9. Enter a policy name.
10. If you are adding an existing rule group, use the dropdown menu to select a rule group to add, and then choose **Add rule group**.
11. A policy has two possible actions: **Action set by rule group** and **Count**. If you want to test the policy and rule group, set the action to **Count**. This action overrides any *block* action specified by the rules in the rule group. That is, if the policy's action is set to **Count**, those requests are only counted and not blocked. Conversely, if you set the policy's action to **Action set by rule group**, actions of the rule group rules are used. Choose the appropriate action.
12. Choose **Next**.

13. For **AWS accounts this policy applies to**, choose the option as follows:

    - If you want to apply the policy to all accounts in your organization, leave the default selection, **Include all accounts under my AWS organization**.
    - If you want to apply the policy only to specific accounts or accounts that are in specific AWS Organizations organizational units (OUs), choose **Include only the specified accounts and organizational units**, and then add the accounts and OUs that you want to include. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.
    - If you want to apply the policy to all but a specific set of accounts or AWS Organizations organizational units (OUs), choose **Exclude the specified accounts and organizational units, and include all others**, and then add the accounts and OUs that you want to exclude. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

    You can only choose one of the options.

    After you apply the policy, Firewall Manager automatically evaluates any new accounts against your settings. For example, if you include only specific accounts, Firewall Manager doesn't apply the policy to any new accounts. As another example, if you include an OU, when you add an account to the OU or to any of its child OUs, Firewall Manager automatically applies the policy to the new account.
14. Choose the type of resource that you want to protect.
15. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.
16. If you want to automatically apply the policy to existing resources, choose **Create and apply this policy to existing and new resources**.

    This option creates a web ACL in each applicable account within an AWS organization and associates the web ACL with the resources in the accounts. This option also applies the policy

to all new resources that match the preceding criteria (resource type and tags). Alternatively, if you choose **Create policy but do not apply the policy to existing or new resources**, Firewall Manager creates a web ACL in each applicable account within the organization, but doesn't apply the web ACL to any resources. You must apply the policy to resources later. Choose the appropriate option.

17. For **Replace existing associated web ACLs**, you can choose to remove any web ACL associations that are currently defined for in-scope resources, and then replace them with associations to the web ACLs that you are creating with this policy. By default, Firewall Manager doesn't remove existing web ACL associations before it adds the new ones. If you want to remove the existing ones, choose this option.
18. Choose **Next**.
19. Review the new policy. To make any changes, choose **Edit**. When you are satisfied with the policy, choose **Create and apply policy**.

Creating an AWS Firewall Manager policy for AWS Shield Advanced

To create a Firewall Manager policy for Shield Advanced (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. Choose **Create policy**.
4. For **Policy type**, choose **Shield Advanced**.

   To create a Shield Advanced policy, you must be subscribed to Shield Advanced. If you are not subscribed, you are prompted to do so. For information about the cost for subscribing, see AWS Shield Advanced Pricing.

5. For **Region**, choose an AWS Region. To protect Amazon CloudFront distributions, choose **Global**.

   For Region choices other than **Global**, to protect resources in multiple Regions, you must create a separate Firewall Manager policy for each Region.

6. Choose **Next**.

7. For **Name**, enter a descriptive name.

8. For **Global** Region policies only, you can choose whether you want to manage Shield Advanced automatic application layer DDoS mitigation. For information about this Shield Advanced feature, see Automating application layer DDoS mitigation with Shield Advanced.

   You can choose to enable or disable automatic mitigation, or you can choose to ignore it. If you choose to ignore it, Firewall Manager doesn't manage automatic mitigation at all for the Shield Advanced protections. For more information about these policy options, see Using Automatic application layer DDoS mitigation with Firewall Manager Shield Advanced policies.

9. Under **Web ACL management**, if you want Firewall Manager to manage unassociated web ACLs, then enable **Manage unassociated web ACLs**. With this option, Firewall Manager creates web ACLs in the accounts within policy scope only if the web ACLs will be used by at least one resource. If at any time an account comes into policy scope, Firewall Manager automatically creates a web ACL in the account if at least one resource will use the web ACL. Upon enablement of this option, Firewall Manager performs a one-time cleanup of unassociated web ACLs in your account. The cleanup process can take several hours. If a resource leaves policy scope after Firewall Manager creates a web ACL, Firewall Manager will not disassociate the resource from the web ACL. To include the web ACL in the one-time cleanup, you must first manually disassociate the resources from the web ACL and then enable **Manage unassociated web ACLs**.

10. For **Policy action**, we recommend creating the policy with the option that doesn't automatically remediate noncompliant resources. When you disable automatic remediation, you can assess the effects of your new policy before you apply it. When you are satisfied that the changes are what you want, then edit the policy and change the policy action to enable automatic remediation.

    If instead you want to automatically apply the policy to existing in-scope resources, choose **Auto remediate any noncompliant resources**. This option applies Shield Advanced protections for each applicable account within the AWS organization and each applicable resource in the accounts.

For **Global** Region policies only, if you choose **Auto remediate any noncompliant resources**, you can also choose to have Firewall Manager automatically replace any existing AWS WAF Classic web ACL associations with new associations to web ACLs that were created using the latest version of AWS WAF (v2). If you choose this, Firewall Manager removes the associations with the earlier version web ACLs and creates new associations with latest version web ACLs, after creating new empty web ACLs in any in-scope accounts that don't already have them for the policy. For more information about this option, see Replace AWS WAF Classic web ACLs with latest version web ACLs.

11. Choose **Next**.
12. For **AWS accounts this policy applies to**, choose the option as follows:

    - If you want to apply the policy to all accounts in your organization, keep the default selection, **Include all accounts under my AWS organization**.
    - If you want to apply the policy only to specific accounts or accounts that are in specific AWS Organizations organizational units (OUs), choose **Include only the specified accounts and organizational units**, and then add the accounts and OUs that you want to include. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.
    - If you want to apply the policy to all but a specific set of accounts or AWS Organizations organizational units (OUs), choose **Exclude the specified accounts and organizational units, and include all others**, and then add the accounts and OUs that you want to exclude. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

    You can only choose one of the options.

    After you apply the policy, Firewall Manager automatically evaluates any new accounts against your settings. For example, if you include only specific accounts, Firewall Manager doesn't apply the policy to any new accounts. As another example, if you include an OU, when you add an account to the OU or to any of its child OUs, Firewall Manager automatically applies the policy to the new account.
13. Choose the type of resource that you want to protect.

    Firewall Manager does not support Amazon Route 53 or AWS Global Accelerator. If you need to use Shield Advanced to protect resources from these services, you can't use a Firewall

Manager policy. Instead, follow the Shield Advanced guidance at Adding AWS Shield Advanced protection to AWS resources.

14. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.

15. Choose **Next**.

16. For **Policy tags**, add any identifying tags that you want to add to the Firewall Manager policy resource. For more information about tags, see Working with Tag Editor.

17. Choose **Next**.

18. Review the new policy settings and return to any pages where you need to any adjustments.

    When you are satisfied with the policy, choose **Create policy**. In the **AWS Firewall Manager policies** pane, your policy should be listed. It will probably indicate **Pending** under the accounts headings and it will indicate the status of the **Automatic remediation** setting. The creation of a policy can take several minutes. After the **Pending** status is replaced with account counts, you can choose the policy name to explore the compliance status of the accounts and resources. For information, see Viewing compliance information for an AWS Firewall Manager policy

Creating an AWS Firewall Manager common security group policy

For information about how common security group policies work, see Using common security group policies with Firewall Manager.

To create a common security group policy, you must have a security group already created in your Firewall Manager administrator account that you want to use as the primary for your policy. You can manage security groups through Amazon Virtual Private Cloud (Amazon VPC) or Amazon Elastic Compute Cloud (Amazon EC2). For information, see Working with Security Groups in the *Amazon VPC User Guide*.

To create a common security group policy (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. Choose **Create policy**.
4. For **Policy type**, choose **Security group**.
5. For **Security group policy type**, choose **Common security groups**.
6. For **Region**, choose an AWS Region.
7. Choose **Next**.
8. For **Policy name**, enter a friendly name.
9. For **Policy rules**, do the following:

   a. From the rules option, choose the restrictions that you want to apply to the security group rules and the resources that are within policy scope. If you choose **Distribute tags from the primary security group to the security groups created by this policy**, then you must also select **Identify and report when the security groups created by this policy become non-compliant**.

   > **Important**
   >
   > Firewall Manager won't distribute system tags added by AWS services into the replica security groups. System tags begin with the `aws:` prefix. Additionally, Firewall Manager won't update the tags of existing security groups or create new security groups if the policy has tags that conflict with the organization's tag policy. For information about tag policies, see Tag policies in the AWS Organizations User Guide.

If you choose **Distribute security group references from the primary security group to the security groups created by this policy**, Firewall Manager only distributes the security group references if they have an active peering connection in Amazon VPC. For information about this option, see Using common security group policies with Firewall Manager.

b. For **Primary security groups**, choose **Add security groups**, and then choose the security groups that you want to use. Firewall Manager populates the list of security groups from all Amazon VPC instances in the Firewall Manager administrator account.

   By default, the maximum number of primary security groups per policy is 3. For information about this setting, see AWS Firewall Manager quotas.

c. For **Policy action**, we recommend creating the policy with the option that doesn't automatically remediate. This allows you to assess the effects of your new policy before you apply it. When you are satisfied that the changes are what you want, then edit the policy and change the policy action to enable automatic remediation of noncompliant resources.

10. Choose **Next**.
11. For **AWS accounts this policy applies to**, choose the option as follows:

    - If you want to apply the policy to all accounts in your organization, leave the default selection, **Include all accounts under my AWS organization**.
    - If you want to apply the policy only to specific accounts or accounts that are in specific AWS Organizations organizational units (OUs), choose **Include only the specified accounts and organizational units**, and then add the accounts and OUs that you want to include. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.
    - If you want to apply the policy to all but a specific set of accounts or AWS Organizations organizational units (OUs), choose **Exclude the specified accounts and organizational units, and include all others**, and then add the accounts and OUs that you want to exclude. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

    You can only choose one of the options.

After you apply the policy, Firewall Manager automatically evaluates any new accounts against your settings. For example, if you include only specific accounts, Firewall Manager doesn't apply the policy to any new accounts. As another example, if you include an OU, when you add an account to the OU or to any of its child OUs, Firewall Manager automatically applies the policy to the new account.

12. For **Resource type**, choose the types of resources that you want to protect.

    For the resource type **EC2 instance**, you can choose to remediate all Amazon EC2 instances or only remediate instances that have just the default, primary elastic network interface (ENI). For the latter option, Firewall Manager doesn't remediate instances that have additional ENI attachments. Instead, when automatic remediation is enabled, Firewall Manager only marks the compliance status of these EC2 instances, and doesn't apply any remediation actions. See additional caveats and limitations for the Amazon EC2 resource type at Security group policy caveats and limitations.

13. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.

14. For **Shared VPC resources**, if you want to apply the policy to resources in shared VPCs, in addition to the VPCs that the accounts own, select **Include resources from shared VPCs**.

15. Choose **Next**.

16. Review the policy settings to be sure they're what you want, and then choose **Create policy**.

Firewall Manager creates a replica of the primary security group in every Amazon VPC instance contained within the in-scope accounts up to the supported Amazon VPC maximum quota per account. Firewall Manager associates the replica security groups to the resources that are within policy scope for each in-scope account. For more information about how this policy works, see Using common security group policies with Firewall Manager.

Creating an AWS Firewall Manager content audit security group policy

For information about how content audit security group policies work, see Using content audit security group policies with Firewall Manager.

For some content audit policy settings, you must provide an audit security group for Firewall Manager to use as a template. For example, you might have an audit security group that contains all of the rules that you don't allow in any security group. You must create these audit security groups using your Firewall Manager administrator account, before you can use them in your policy. You can manage security groups through Amazon Virtual Private Cloud (Amazon VPC) or Amazon Elastic Compute Cloud (Amazon EC2). For information, see Working with Security Groups in the *Amazon VPC User Guide*.

To create a content audit security group policy (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. Choose **Create policy**.
4. For **Policy type**, choose **Security group**.
5. For **Security group policy type**, choose **Auditing and enforcement of security group rules**.
6. For **Region**, choose an AWS Region.
7. Choose **Next**.
8. For **Policy name**, enter a friendly name.
9. For **Policy rules**, choose the managed or custom policy rules option that you want to use.

   a. For **Configure managed audit policy rules**, do the following:

      i. For **Configure security group rules to audit**, select the type of security group rules that you want your audit policy to apply to.

ii. If you want to do things like audit rules based on the protocols, ports, and CIDR range settings that in your security groups, choose **Audit overly permissive security group rules** and select the options that you want.

For the selection **Rule allows all traffic**, you can provide a custom application list to designate the applications that you want to audit. For information about custom application lists and how to use them in your policy, see Using managed lists and Using managed lists.

For selections that use protocol lists, you can use existing lists and you can create new lists. For information about protocol lists and how to use them in your policy, see Using managed lists and Using managed lists.

iii. If you want to audit high-risk based on their access to either reserved or non-reserved CIDR ranges, choose **Audit high risk applications** and select the options that you want.

The following selections are mutually exclusive: **Applications that can access only reserved CIDR ranges** and **Applications allowed to access non-reserved CIDR ranges**. You can select at most one of them in any policy.

For selections that use application lists, you can use existing lists and you can create new lists. For information about application lists and how to use them in your policy, see Using managed lists and Using managed lists.

iv. Use the **Overrides** settings to explicitly override other settings in the policy. You can choose to always allow or always deny specific security group rules, regardless of whether they comply with the other options that you've set for the policy.

For this option, you provide an audit security group as your allowed rules or denied rules template. For **Audit security groups**, choose **Add audit security groups**, and then choose the security group that you want to use. Firewall Manager populates the list of audit security groups from all Amazon VPC instances in the Firewall Manager administrator account. The default maximum quota for the number of audit security groups for a policy is one. For information about increasing the quota, see AWS Firewall Manager quotas.

b. For **Configure custom policy rules**, do the following:

i. From the rules options, choose whether to allow only the rules defined in the audit security groups or deny all the rules. For information about this choice, see Using content audit security group policies with Firewall Manager.

ii. For **Audit security groups**, choose **Add audit security groups**, and then choose the security group that you want to use. Firewall Manager populates the list of audit security groups from all Amazon VPC instances in the Firewall Manager administrator account. The default maximum quota for the number of audit security groups for a policy is one. For information about increasing the quota, see AWS Firewall Manager quotas.

iii. For **Policy action**, you must create the policy with the option that doesn't automatically remediate. This allows you to assess the effects of your new policy before you apply it. When you are satisfied that the changes are what you want, edit the policy and change the policy action to enable automatic remediation of noncompliant resources.

10. Choose **Next**.
11. For **AWS accounts this policy applies to**, choose the option as follows:

- If you want to apply the policy to all accounts in your organization, leave the default selection, **Include all accounts under my AWS organization**.
- If you want to apply the policy only to specific accounts or accounts that are in specific AWS Organizations organizational units (OUs), choose **Include only the specified accounts and organizational units**, and then add the accounts and OUs that you want to include. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.
- If you want to apply the policy to all but a specific set of accounts or AWS Organizations organizational units (OUs), choose **Exclude the specified accounts and organizational units, and include all others**, and then add the accounts and OUs that you want to exclude. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

You can only choose one of the options.

After you apply the policy, Firewall Manager automatically evaluates any new accounts against your settings. For example, if you include only specific accounts, Firewall Manager doesn't apply the policy to any new accounts. As another example, if you include an OU, when you add

an account to the OU or to any of its child OUs, Firewall Manager automatically applies the policy to the new account.

12. For **Resource type**, choose the types of resource that you want to protect.
13. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.
14. Choose **Next**.
15. Review the policy settings to be sure they're what you want, and then choose **Create policy**.

Firewall Manager compares the audit security group against the in-scope security groups in your AWS organization, according to your policy rules settings. You can review the policy status in the AWS Firewall Manager policy console. After the policy is created, you can edit it and enable automatic remediation to put your auditing security group policy into effect. For more information about how this policy works, see Using content audit security group policies with Firewall Manager.

Creating an AWS Firewall Manager usage audit security group policy

For information about how usage audit security group policies work, see Using usage audit security group policies with Firewall Manager.

To create a usage audit security group policy (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.

3. Choose **Create policy**.
4. For **Policy type**, choose **Security group**.
5. For **Security group policy type**, choose **Auditing and cleanup of unassociated and redundant security groups**.
6. For **Region**, choose an AWS Region.
7. Choose **Next**.
8. For **Policy name**, enter a friendly name.
9. For **Policy rules**, choose one or both of the options available.

   - If you choose **Security groups within this policy scope must be used by at least one resource**, Firewall Manager removes any security groups that it determines are unused. When this rule is enabled, Firewall Manager runs it last when you save the policy.

     For details about how Firewall Manager determines usage and the timing of the remediation, see Using usage audit security group policies with Firewall Manager.

     > **Note**
     >
     > When you use this usage audit security group policy type, avoid making multiple changes to the association status of in-scope security groups in a short amount of time. Doing so can cause Firewall Manager to miss corresponding events.

   By default, Firewall Manager considers security groups as noncompliant with this policy rule as soon as they're unused. You can optionally specify a number of minutes that a security group can exist unused before it's considered noncompliant, up to 525,600 minutes (365 days). You can use this setting to allow yourself time to associate new security groups with resources.

   > **Important**
   >
   > If you specify a number of minutes other than the default value of zero, you must enable indirect relationships in AWS Config. Otherwise, your usage audit security group policies will not work as intended. For information about indirect relationships in AWS Config, see Indirect Relationships in AWS Config in the *AWS Config Developer Guide*.

- If you choose **Security groups within this policy scope must be unique**, Firewall Manager consolidates redundant security groups, so that only one is associated with any resources. If you choose this, Firewall Manager runs it first when you save the policy.

10. For **Policy action**, we recommend creating the policy with the option that doesn't automatically remediate. This allows you to assess the effects of your new policy before you apply it. When you are satisfied that the changes are what you want, then edit the policy and change the policy action to enable automatic remediation of noncompliant resources.
11. Choose **Next**.
12. For **AWS accounts this policy applies to**, choose the option as follows:

    - If you want to apply the policy to all accounts in your organization, leave the default selection, **Include all accounts under my AWS organization**.
    - If you want to apply the policy only to specific accounts or accounts that are in specific AWS Organizations organizational units (OUs), choose **Include only the specified accounts and organizational units**, and then add the accounts and OUs that you want to include. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.
    - If you want to apply the policy to all but a specific set of accounts or AWS Organizations organizational units (OUs), choose **Exclude the specified accounts and organizational units, and include all others**, and then add the accounts and OUs that you want to exclude. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

    You can only choose one of the options.

    After you apply the policy, Firewall Manager automatically evaluates any new accounts against your settings. For example, if you include only specific accounts, Firewall Manager doesn't apply the policy to any new accounts. As another example, if you include an OU, when you add an account to the OU or to any of its child OUs, Firewall Manager automatically applies the policy to the new account.
13. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.

14. Choose **Next**.
15. If you haven't excluded the Firewall Manager administrator account from the policy scope, Firewall Manager prompts you to do this. Doing this leaves the security groups in the Firewall Manager administrator account, which you use for common and audit security group policies, under your manual control. Choose the option you want in this dialogue.
16. Review the policy settings to be sure they're what you want, and then choose **Create policy**.

If you chose to require unique security groups, Firewall Manager scans for redundant security groups in each in-scope Amazon VPC instance. Then, if you chose to require that each security group be used by at least one resource, Firewall Manager scans for security groups that have remained unused for the minutes specified in the rule. You can review the policy status in the AWS Firewall Manager policy console. For more information about how this policy works, see Using usage audit security group policies with Firewall Manager.

Creating an AWS Firewall Manager network ACL policy

For information about how network ACL policies work, see Network ACL policies.

To create a network ACL policy, you must know how to define a network ACL for use with your Amazon VPC subnets. For information, see Control traffic to subnets using network ACLs and Work with network ACLs in the *Amazon VPC User Guide*.

To create a network ACL policy (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

> **Note**
>
> For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. Choose **Create policy**.
4. For **Policy type**, choose **Network ACL**.
5. For **Region**, choose an AWS Region.
6. Choose **Next**.
7. For **Policy name**, enter a descriptive name.
8. For **Policy rules**, define the rules that you want to always run in the network ACLs that Firewall Manager manages for you. Network ACLs monitor and handle inbound and outbound traffic, so in your policy, you define the rules for both directions.

   For either direction, you define rules that you want to always run first and rules you want to always run last. In the network ACLs that Firewall Manager manages, account owners can define custom rules to run in between these first and last rules.
9. For **Policy action**, if you want to identify noncompliant subnets and network ACLs, but not take any corrective action yet, choose **Identify resources that don't comply with the policy rules, but don't auto remediate**. You can change these options later.

   If instead you want to automatically apply the policy to existing in-scope subnets, choose **Auto remediate any noncompliant resources**. With this option, you also specify whether to force remediation when the traffic handling behavior of policy rules conflicts with custom rules that are in the network ACL. Regardless of whether you force remediation, Firewall Manager reports conflicting rules in its compliance violations.
10. Choose **Next**.
11. For **AWS accounts this policy applies to**, choose the option as follows:

    - If you want to apply the policy to all accounts in your organization, leave the default selection, **Include all accounts under my AWS organization**.
    - If you want to apply the policy only to specific accounts or accounts that are in specific AWS Organizations organizational units (OUs), choose **Include only the specified accounts and organizational units**, and then add the accounts and OUs that you want to include. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.
    - If you want to apply the policy to all but a specific set of accounts or AWS Organizations organizational units (OUs), choose **Exclude the specified accounts and organizational units, and include all others**, and then add the accounts and OUs that you want to exclude.

Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

You can only choose one of the options.

After you apply the policy, Firewall Manager automatically evaluates any new accounts against your settings. For example, if you include only specific accounts, Firewall Manager doesn't apply the policy to any different, new accounts. As another example, if you include an OU, when you add an account to the OU or to any of its child OUs, Firewall Manager automatically applies the policy to the new account.

12. For **Resource type**, the setting is fixed at **Subnets**.
13. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.
14. Choose **Next**.
15. Review the policy settings to be sure they're what you want, and then choose **Create policy**.

Firewall Manager creates the policy and begins monitoring and managing the in scope network ACLs according to your settings. For more information about how this policy works, see Network ACL policies.

Creating an AWS Firewall Manager policy for AWS Network Firewall

In a Firewall Manager Network Firewall policy, you use rule groups that you manage in AWS Network Firewall. For information about managing your rule groups, see AWS Network Firewall rule groups in the *Network Firewall Developer Guide*.

For information about Firewall Manager Network Firewall policies, see Using AWS Network Firewall policies in Firewall Manager.

To create a Firewall Manager policy for AWS Network Firewall (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. Choose **Create policy**.
4. For **Policy type**, choose **AWS Network Firewall**.
5. Under **Firewall management type**, choose how you'd like Firewall Manager to manage the policy's firewalls. Choose from the following options:

   - **Distributed** - Firewall Manager creates and maintains firewall endpoints in each VPC that's in the policy scope.
   - **Centralized** - Firewall Manager creates and maintains endpoints in a single inspection VPC.
   - **Import existing firewalls** - Firewall Manager imports existing firewalls from Network Firewall using resource sets. For information about resource sets, see Grouping your resources in Firewall Manager.

6. For **Region**, choose an AWS Region. To protect resources in multiple Regions, you must create separate policies for each Region.
7. Choose **Next**.
8. For **Policy name**, enter a descriptive name. Firewall Manager includes the policy name in the names of the Network Firewall firewalls and firewall policies that it creates.
9. In the **AWS Network Firewall policy configuration**, configure the firewall policy as you would in Network Firewall. Add your stateless and stateful rule groups and specify the policy's default actions. You can optionally set the policy's stateful rule evaluation order and default actions, as well as logging configuration. For information about Network Firewall firewall policy management, see AWS Network Firewall firewall policies in the *AWS Network Firewall Developer Guide*.

When you create the Firewall Manager Network Firewall policy, Firewall Manager creates firewall policies for the accounts that are within scope. Individual account managers can add rule groups to the firewall policies, but they can't change the configuration that you provide here.

10. Choose **Next**.
11. Do one of the following, depending on the **Firewall management type** you selected in the previous step:

- If you're using a **distributed** firewall management type, in **AWS Firewall Manager endpoint configuration** under **Firewall endpoint location**, choose one of the following options:
  - **Custom endpoint configuration** - Firewall Manager creates firewalls for each VPC within the policy scope, in the Availability Zones that you specify. Each firewall contains at least one firewall endpoint.
    - Under **Availability Zones**, select which Availability Zones to create firewall endpoints in. You can select Availability Zones by **Availability Zone name** or by **Availability Zone ID**.
    - If you want to provide the CIDR blocks for Firewall Manager to use for firewall subnets in your VPCs, they must all be /28 CIDR blocks. Enter one block per line. If you omit these, Firewall Manager chooses IP addresses for you from those that are available in the VPCs.

      > **Note**
      >
      > Auto remediation happens automatically for AWS Firewall Manager Network Firewall policies, so you won't see an option to choose not to auto remediate here.

  - **Automatic endpoint configuration** - Firewall Manager automatically creates firewall endpoints in the Availability Zones with public subnets in your VPC.
    - For the **Firewall endpoints** configuration, specify how you want the firewall endpoints to be managed by Firewall Manager. We recommend using multiple endpoints for high availability.
- If you're using a **centralized** firewall management type, in **AWS Firewall Manager endpoint configuration** under **Inspection VPC configuration**, enter the AWS account ID of the owner of the inspection VPC, and the VPC ID of the inspection VPC.

- Under **Availability Zones**, select which Availability Zones to create firewall endpoints in. You can select Availability Zones by **Availability Zone name** or by **Availability Zone ID**.
- If you want to provide the CIDR blocks for Firewall Manager to use for firewall subnets in your VPCs, they must all be /28 CIDR blocks. Enter one block per line. If you omit these, Firewall Manager chooses IP addresses for you from those that are available in the VPCs.

> **Note**
>
> Auto remediation happens automatically for AWS Firewall Manager Network Firewall policies, so you won't see an option to choose not to auto remediate here.

- If you're using a **import existing firewalls** firewall management type, in **Resource sets** add one or more resource sets. A resource set defines the existing Network Firewall firewalls owned by your organization's account that you want to centrally manage in this policy. To add a resource set to the policy, you must first create a resource set using the console or the PutResourceSet API. For information about resource sets, see Grouping your resources in Firewall Manager. For more information about importing existing firewalls from Network Firewall, see How Firewall Manager creates firewall endpoints.

12. Choose **Next**.
13. If your policy uses a distributed firewall management type, under **Route management**, choose whether or not Firewall Manager will monitor and alert on the traffic that must be routed through the respective firewall endpoints.

> **Note**
>
> If you choose **Monitor**, you can't change the setting to **Off** at a later date. Monitoring continues until you delete the policy.

14. For **Traffic type**, optionally add the traffic endpoints that you want to route traffic through for firewall inspection.
15. For **Allow required cross-AZ traffic**, if you enable this option then Firewall Manager treats as compliant routing that sends traffic out of an Availability Zone for inspection, for Availability Zones that don't have their own firewall endpoint. Availability Zones that have endpoints must always inspect their own traffic.
16. Choose **Next**.
17. For **Policy scope**, under **AWS accounts this policy applies to**, choose the option as follows:

- If you want to apply the policy to all accounts in your organization, leave the default selection, **Include all accounts under my AWS organization**.
- If you want to apply the policy only to specific accounts or accounts that are in specific AWS Organizations organizational units (OUs), choose **Include only the specified accounts and organizational units**, and then add the accounts and OUs that you want to include. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.
- If you want to apply the policy to all but a specific set of accounts or AWS Organizations organizational units (OUs), choose **Exclude the specified accounts and organizational units, and include all others**, and then add the accounts and OUs that you want to exclude. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

You can only choose one of the options.

After you apply the policy, Firewall Manager automatically evaluates any new accounts against your settings. For example, if you include only specific accounts, Firewall Manager doesn't apply the policy to any new accounts. As another example, if you include an OU, when you add an account to the OU or to any of its child OUs, Firewall Manager automatically applies the policy to the new account.

18. The **Resource type** for Network Firewall policies is **VPC**.
19. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.
20. Choose **Next**.
21. For **Policy tags**, add any identifying tags that you want to add to the Firewall Manager policy resource. For more information about tags, see Working with Tag Editor.
22. Choose **Next**.
23. Review the new policy settings and return to any pages where you need to any adjustments.

When you are satisfied with the policy, choose **Create policy**. In the **AWS Firewall Manager policies** pane, your policy should be listed. It will probably indicate **Pending** under the accounts headings and it will indicate the status of the **Automatic remediation** setting. The creation of a policy can take several minutes. After the **Pending** status is replaced with account counts, you can choose the policy name to explore the compliance status of the accounts and resources. For information, see Viewing compliance information for an AWS Firewall Manager policy

Creating an AWS Firewall Manager policy for Amazon Route 53 Resolver DNS Firewall

In a Firewall Manager DNS Firewall policy, you use rule groups that you manage in Amazon Route 53 Resolver DNS Firewall. For information about managing your rule groups, see Managing rule groups and rules in DNS Firewall in the *Amazon Route 53 Developer Guide*.

For information about Firewall Manager DNS Firewall policies, see Using Amazon Route 53 Resolver DNS Firewall policies in Firewall Manager.

To create a Firewall Manager policy for Amazon Route 53 Resolver DNS Firewall (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. Choose **Create policy**.
4. For **Policy type**, choose **Amazon Route 53 Resolver DNS Firewall**.
5. For **Region**, choose an AWS Region. To protect resources in multiple Regions, you must create separate policies for each Region.
6. Choose **Next**.

7. For **Policy name**, enter a descriptive name.
8. In the policy configuration, add the rule groups that you want DNS Firewall to evaluate first and last among your VPCs' rule group associations. You can add up to two rule groups to the policy.

   When you create the Firewall Manager DNS Firewall policy, Firewall Manager creates the rule group associations, with the association priorities that you've provided, for the VPCs and accounts that are within scope. The individual account managers can add rule group associations in between your first and last associations, but they can't change the associations that you define here. For more information, see Using Amazon Route 53 Resolver DNS Firewall policies in Firewall Manager.
9. Choose **Next**.
10. For **AWS accounts this policy applies to**, choose the option as follows:

    - If you want to apply the policy to all accounts in your organization, leave the default selection, **Include all accounts under my AWS organization**.
    - If you want to apply the policy only to specific accounts or accounts that are in specific AWS Organizations organizational units (OUs), choose **Include only the specified accounts and organizational units**, and then add the accounts and OUs that you want to include. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.
    - If you want to apply the policy to all but a specific set of accounts or AWS Organizations organizational units (OUs), choose **Exclude the specified accounts and organizational units, and include all others**, and then add the accounts and OUs that you want to exclude. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

    You can only choose one of the options.

    After you apply the policy, Firewall Manager automatically evaluates any new accounts against your settings. For example, if you include only specific accounts, Firewall Manager doesn't apply the policy to any new accounts. As another example, if you include an OU, when you add an account to the OU or to any of its child OUs, Firewall Manager automatically applies the policy to the new account.
11. The **Resource type** for DNS Firewall policies is **VPC**.

12. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.
13. Choose **Next**.
14. For **Policy tags**, add any identifying tags that you want to add to the Firewall Manager policy resource. For more information about tags, see Working with Tag Editor.
15. Choose **Next**.
16. Review the new policy settings and return to any pages where you need to any adjustments.

    When you are satisfied with the policy, choose **Create policy**. In the **AWS Firewall Manager policies** pane, your policy should be listed. It will probably indicate **Pending** under the accounts headings and it will indicate the status of the **Automatic remediation** setting. The creation of a policy can take several minutes. After the **Pending** status is replaced with account counts, you can choose the policy name to explore the compliance status of the accounts and resources. For information, see Viewing compliance information for an AWS Firewall Manager policy

Creating an AWS Firewall Manager policy for Palo Alto Networks Cloud NGFW

With a Firewall Manager policy for Palo Alto Networks Cloud Next Generation Firewall (Palo Alto Networks Cloud NGFW), you use Firewall Manager to deploy Palo Alto Networks Cloud NGFW resources, and manage NGFW rulestacks centrally across all of your AWS accounts.

For information about Firewall Manager Palo Alto Networks Cloud NGFW policies, see Using Palo Alto Networks Cloud NGFW policies for Firewall Manager. For information about how to configure and manage Palo Alto Networks Cloud NGFW for Firewall Manager, see the *Palo Alto Networks Palo Alto Networks Cloud NGFW on AWS* documentation.

Prerequisites

There are several mandatory steps to prepare your account for AWS Firewall Manager. Those steps are described in AWS Firewall Manager prerequisites. Complete all the prerequisites before proceeding to the next step.

To create a Firewall Manager policy for Palo Alto Networks Cloud NGFW (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. Choose **Create policy**.
4. For **Policy type**, choose **Palo Alto Networks Cloud NGFW**. If you haven't already subscribed to the Palo Alto Networks Cloud NGFW service in the AWS Marketplace, you'll need to do that first. To subscribe in the AWS Marketplace, choose **View AWS Marketplace details**.
5. For **Deployment model**, choose either the **Distributed model** or **Centralized model**. The deployment model determines how Firewall Manager manages endpoints for the policy. With the distributed model, Firewall Manager maintains firewall endpoints in each VPC that's within policy scope. With the centralized model, Firewall Manager maintains a single endpoint in an inspection VPC.
6. For **Region**, choose an AWS Region. To protect resources in multiple Regions, you must create separate policies for each Region.
7. Choose **Next**.
8. For **Policy name**, enter a descriptive name.
9. In the policy configuration, choose the Palo Alto Networks Cloud NGFW firewall policy to associate with this policy. The list of Palo Alto Networks Cloud NGFW firewall policies contains all of the Palo Alto Networks Cloud NGFW firewall policies that are associated with your Palo Alto Networks Cloud NGFW tenant. For information about creating and managing Palo Alto Networks Cloud NGFW firewall policies, see the *Deploy Palo Alto Networks Cloud NGFW for AWS with the AWS Firewall Manager* topic in the *Palo Alto Networks Cloud NGFW for AWS deployment guide*.
10. For **Palo Alto Networks Cloud NGFW logging - optional**, optionally choose which Palo Alto Networks Cloud NGFW log type(s) to log for your policy. For information about Palo Alto

Networks Cloud NGFW log types, see Configure Logging for Palo Alto Networks Cloud NGFW on AWS in the *Palo Alto Networks Cloud NGFW for AWS deployment guide*.

For **log destination**, specify when Firewall Manager should write logs to.

11. Choose **Next**.
12. Under **Configure third-party firewall endpoint** do one of the following, depending on whether you're using the distributed or centralized deployment model to create your firewall endpoints:

    - If you're using the distributed deployment model for this policy, under **Availability Zones**, select which Availability Zones to create firewall endpoints in. You can select Availability Zones by **Availability Zone name** or by **Availability Zone ID**.
    - If you're using the centralized deployment model for this policy, in **AWS Firewall Manager endpoint configuration** under **Inspection VPC configuration**, enter the AWS account ID of the owner of the inspection VPC, and the VPC ID of the inspection VPC.
      - Under **Availability Zones**, select which Availability Zones to create firewall endpoints in. You can select Availability Zones by **Availability Zone name** or by **Availability Zone ID**.
13. If you want to provide the CIDR blocks for Firewall Manager to use for firewall subnets in your VPCs, they must all be /28 CIDR blocks. Enter one block per line. If you omit these, Firewall Manager chooses IP addresses for you from those that are available in the VPCs.

    > **Note**
    >
    > Auto remediation happens automatically for AWS Firewall Manager Network Firewall policies, so you won't see an option to choose not to auto remediate here.

14. Choose **Next**.
15. For **Policy scope**, under **AWS accounts this policy applies to**, choose the option as follows:

    - If you want to apply the policy to all accounts in your organization, leave the default selection, **Include all accounts under my AWS organization**.
    - If you want to apply the policy only to specific accounts or accounts that are in specific AWS Organizations organizational units (OUs), choose **Include only the specified accounts and organizational units**, and then add the accounts and OUs that you want to include. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

- If you want to apply the policy to all but a specific set of accounts or AWS Organizations organizational units (OUs), choose **Exclude the specified accounts and organizational units, and include all others**, and then add the accounts and OUs that you want to exclude. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

You can only choose one of the options.

After you apply the policy, Firewall Manager automatically evaluates any new accounts against your settings. For example, if you include only specific accounts, Firewall Manager doesn't apply the policy to any new accounts. As another example, if you include an OU, when you add an account to the OU or to any of its child OUs, Firewall Manager automatically applies the policy to the new account.

16. The **Resource type** for Network Firewall policies is **VPC**.
17. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.
18. For **Grant cross-account access**, choose **Download CloudFormation template**. This downloads an CloudFormation template that you can use to create an CloudFormation stack. This stack creates an AWS Identity and Access Management role that grants Firewall Manager cross-account permissions to manage Palo Alto Networks Cloud NGFW resources. For information about stacks, see Working with stacks in the *CloudFormation User Guide*.
19. Choose **Next**.
20. For **Policy tags**, add any identifying tags that you want to add to the Firewall Manager policy resource. For more information about tags, see Working with Tag Editor.
21. Choose **Next**.
22. Review the new policy settings and return to any pages where you need to any adjustments.

    When you are satisfied with the policy, choose **Create policy**. In the **AWS Firewall Manager policies** pane, your policy should be listed. It will probably indicate **Pending** under the accounts headings and it will indicate the status of the **Automatic remediation** setting. The

creation of a policy can take several minutes. After the **Pending** status is replaced with account counts, you can choose the policy name to explore the compliance status of the accounts and resources. For information, see Viewing compliance information for an AWS Firewall Manager policy

Creating an AWS Firewall Manager policy for Fortigate Cloud Native Firewall (CNF) as a Service

With a Firewall Manager policy for Fortigate CNF, you can use Firewall Manager to deploy and manage Fortigate CNF resources across all of your AWS accounts.

For information about Firewall Manager Fortigate CNF policies, see Using Fortigate Cloud Native Firewall (CNF) as a Service policies for Firewall Manager. For information about how to configure Fortigate CNF for use with Firewall Manager, see the Fortinet documentation.

Prerequisites

There are several mandatory steps to prepare your account for AWS Firewall Manager. Those steps are described in AWS Firewall Manager prerequisites. Complete all the prerequisites before proceeding to the next step.

To create a Firewall Manager policy for Fortigate CNF (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. Choose **Create policy**.
4. For **Policy type**, choose **Fortigate Cloud Native Firewall (CNF) as a Service**. If you haven't already subscribed to the Fortigate CNF service in the AWS Marketplace, you'll need to do that first. To subscribe in the AWS Marketplace, choose **View AWS Marketplace details**.

5. For **Deployment model**, choose either the **Distributed model** or **Centralized model**. The deployment model determines how Firewall Manager manages endpoints for the policy. With the distributed model, Firewall Manager maintains firewall endpoints in each VPC that's within policy scope. With the centralized model, Firewall Manager maintains a single endpoint in an inspection VPC.
6. For **Region**, choose an AWS Region. To protect resources in multiple Regions, you must create separate policies for each Region.
7. Choose **Next**.
8. For **Policy name**, enter a descriptive name.
9. In the policy configuration, choose the Fortigate CNF firewall policy to associate with this policy. The list of Fortigate CNF firewall policies contains all of the Fortigate CNF firewall policies that are associated with your Fortigate CNF tenant. For information about creating and managing Fortigate CNF tenants, see the Fortinet documentation.
10. Choose **Next**.
11. Under **Configure third-party firewall endpoint** do one of the following, depending on whether you're using the distributed or centralized deployment model to create your firewall endpoints:

    - If you're using the distributed deployment model for this policy, under **Availability Zones**, select which Availability Zones to create firewall endpoints in. You can select Availability Zones by **Availability Zone name** or by **Availability Zone ID**.
    - If you're using the centralized deployment model for this policy, in **AWS Firewall Manager endpoint configuration** under **Inspection VPC configuration**, enter the AWS account ID of the owner of the inspection VPC, and the VPC ID of the inspection VPC.
      - Under **Availability Zones**, select which Availability Zones to create firewall endpoints in. You can select Availability Zones by **Availability Zone name** or by **Availability Zone ID**.
12. If you want to provide the CIDR blocks for Firewall Manager to use for firewall subnets in your VPCs, they must all be /28 CIDR blocks. Enter one block per line. If you omit these, Firewall Manager chooses IP addresses for you from those that are available in the VPCs.

> **Note**
>
> Auto remediation happens automatically for AWS Firewall Manager Network Firewall policies, so you won't see an option to choose not to auto remediate here.

13. Choose **Next**.
14. For **Policy scope**, under **AWS accounts this policy applies to**, choose the option as follows:

    - If you want to apply the policy to all accounts in your organization, leave the default selection, **Include all accounts under my AWS organization**.
    - If you want to apply the policy only to specific accounts or accounts that are in specific AWS Organizations organizational units (OUs), choose **Include only the specified accounts and organizational units**, and then add the accounts and OUs that you want to include. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.
    - If you want to apply the policy to all but a specific set of accounts or AWS Organizations organizational units (OUs), choose **Exclude the specified accounts and organizational units, and include all others**, and then add the accounts and OUs that you want to exclude. Specifying an OU is the equivalent of specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.

    You can only choose one of the options.

    After you apply the policy, Firewall Manager automatically evaluates any new accounts against your settings. For example, if you include only specific accounts, Firewall Manager doesn't apply the policy to any new accounts. As another example, if you include an OU, when you add an account to the OU or to any of its child OUs, Firewall Manager automatically applies the policy to the new account.
15. The **Resource type** for Network Firewall policies is **VPC**.
16. For **Resources**, you can narrow the scope of the policy using tagging, by either including or excluding resources with the tags that you specify. You can use inclusion or exclusion, and not both. For more information about tags to define policy scope, see Using the AWS Firewall Manager policy scope.

    Resource tags can only have non-null values. If you omit the value for a tag, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.
17. For **Grant cross-account access**, choose **Download CloudFormation template**. This downloads an CloudFormation template that you can use to create an CloudFormation stack. This stack creates an AWS Identity and Access Management role that grants Firewall Manager cross-account permissions to manage Fortigate CNF resources. For information about stacks, see

Working with stacks in the *CloudFormation User Guide*. To create a stack, you'll need the account ID from the Fortigate CNF portal.

18. Choose **Next**.
19. For **Policy tags**, add any identifying tags that you want to add to the Firewall Manager policy resource. For more information about tags, see Working with Tag Editor.
20. Choose **Next**.
21. Review the new policy settings and return to any pages where you need to any adjustments.

    When you are satisfied with the policy, choose **Create policy**. In the **AWS Firewall Manager policies** pane, your policy should be listed. It will probably indicate **Pending** under the accounts headings and it will indicate the status of the **Automatic remediation** setting. The creation of a policy can take several minutes. After the **Pending** status is replaced with account counts, you can choose the policy name to explore the compliance status of the accounts and resources. For information, see Viewing compliance information for an AWS Firewall Manager policy

Deleting an AWS Firewall Manager policy

You can delete a Firewall Manager policy by performing the following steps.

To delete a policy (console)

1. In the navigation pane, choose **Security policies**.
2. Choose the option next to the policy that you want to delete.
3. Choose **Delete**.

> **Note**
>
> When you delete a Firewall Manager common security group policy, to remove the policy's replica security groups, choose the option to clean up the resources created by the policy. Otherwise, after the primary is deleted, the replicas remain and require manual management in each Amazon VPC instance.

Important

When you delete a Firewall Manager Shield Advanced policy, the policy is deleted, but your accounts remain subscribed to Shield Advanced.

Using the AWS Firewall Manager policy scope

This page explains what the Firewall Manager policy scope is and how it works.

The policy scope defines where the policy applies. You can apply centrally controlled policies to:

- All of your accounts and resources within your organization in AWS Organizations.
- A subset of your accounts and resources within your organization in AWS Organizations.

For instructions on how to set policy scope, see Creating an AWS Firewall Manager policy.

Policy scope options in AWS Firewall Manager

When you add a new account or resource to your organization, Firewall Manager automatically assesses it against your settings for each policy and applies the policy based on these settings. For example, you can choose to apply a policy to all accounts except the account numbers in a specified list. Resource tags can also be used to define policy scope. You can choose to apply a policy by excluding or including resources that have all of the tags in a list. Alternatively, you can choose to apply a policy only to resources that have any of a specified tag in a list.

AWS accounts in scope

The settings that you provide to define the AWS accounts affected by the policy determine which of the accounts in your AWS organization to apply the policy to. You can choose to apply the policy in one of the following ways:

- To all accounts in your organization
- To only a specific list of included account numbers and AWS Organizations organizational units (OUs)
- To all except a specific list of excluded account numbers and AWS Organizations organizational units (OUs)

For information about AWS Organizations, see AWS Organizations User Guide.

Resources in scope

Similarly to the settings for accounts in scope, the settings that you provide for resources determine which in-scope resource types to apply the policy to. You can choose one of the following:

- All resources
- Resources that have all of the tags that you specify
- All resources except those that have all of the tags that you specify
- Only resources that have any of the tags that you specify
- All resources except only resources that have any of tags that you specify

You can only specify resource tags with non-null values. If you don't provide anything for the value, Firewall Manager saves the tag with an empty string value: "". Resource tags only match with tags that have the same key and the same value.

For more information about tagging your resources, see Working with Tag Editor.

Policy scope management in AWS Firewall Manager

When policies are in place, Firewall Manager manages them continuously and applies them to new AWS accounts and resources as they are added, in accordance with the policy scope.

How Firewall Manager manages AWS accounts and resources

If an account or resource goes out of scope for any reason, AWS Firewall Manager doesn't automatically remove protections or delete Firewall Manager-managed resources unless you select the **Automatically remove protections from resources that leave the policy scope** check box.

> **Note**
>
> The option **Automatically remove protections from resources that leave the policy scope** is not available for AWS Shield Advanced or AWS WAF Classic policies.

Selecting this check box directs AWS Firewall Manager to automatically clean up resources that Firewall Manager manages for accounts when those accounts leave the policy scope. For example,

Firewall Manager will disassociate a Firewall Manager-managed web ACL from a protected customer resource when the customer resource leaves the policy scope.

To determine which resources should be removed from protection when a customer resource leaves the policy scope, Firewall Manager follows these guidelines:

- *Default behavior*:
  - The associated AWS Config managed rules are deleted. This behavior is independent of the check box.
  - Any associated AWS WAF web access control lists (web ACLs) that don't contain any resources are deleted. This behavior is independent of the check box.
  - Any protected resource that goes out of scope remains associated and protected. For example, an Application Load Balancer or API from API Gateway that's associated with a web ACL remains associated with the web ACL, and the protection remains in place.
- *With the **Automatically remove protections from resources that leave the policy scope** check box selected*:
  - The associated AWS Config managed rules are deleted. This behavior is independent of the check box.
  - Any associated AWS WAF web access control lists (web ACLs) that don't contain any resources are deleted. This behavior is independent of the check box.
  - Any protected resource that goes out of scope is automatically disassociated and removed from Firewall Manager protection when it leaves the policy scope. For example, for a security group policy, an Elastic Inference accelerator or Amazon EC2 instance is automatically disassociated from the replicated security group when it leaves the policy scope. The replicated security group and its resources are automatically removed from protection.
  - Firewall Manager deletes the logging configuration regardless of the resource cleanup option value when a member account leaves the scope or when the policy is deleted.

Using AWS WAF policies with Firewall Manager

This section explains how to use AWS WAF policies with Firewall Manager. In a Firewall Manager AWS WAF policy, you specify the AWS WAF rule groups that you want to use to protect all resources that are within policy scope. When you apply the policy, Firewall Manager begins managing web ACLs for in-scope resources, using the specified rule groups and other policy configurations.

You can configure the policy to create and manage all new web ACLs for in-scope resources, replacing any web ACLs that are already in use. Alternately, you can configure the policy to keep any web ACLs that are already associated with in-scope resources, and retrofit them for use by the policy. With this second option, Firewall Manager only creates new web ACLs for resources that don't already have a web ACL association.

Regardless of how they're created, in the web ACLs that Firewall Manager manages, individual accounts can manage their own rules and rule groups, in addition to the rule groups that you define in the Firewall Manager policy.

For the procedure to create a Firewall Manager AWS WAF policy, see Creating an AWS Firewall Manager policy for AWS WAF.

Rule group management for AWS WAF policies

The web ACLs that are managed by Firewall Manager AWS WAF policies contain three sets of rules. These sets provide a higher level of prioritization for the rules and rule groups in the web ACL:

- First rule groups, defined by you in the Firewall Manager AWS WAF policy. AWS WAF evaluates these rule groups first.
- Rules and rule groups that are defined by the account managers in the web ACLs. AWS WAF evaluates any account-managed rules or rule groups next.
- Last rule groups, defined by you in the Firewall Manager AWS WAF policy. AWS WAF evaluates these rule groups last.

Within each of these sets of rules, AWS WAF evaluates rules and rule groups as usual, according to their priority settings within the set.

In the policy's first and last rule groups sets, you can only add rule groups and not individual rules. You can use managed rule groups, which AWS Managed Rules and AWS Marketplace sellers create and maintain for you. You can also manage and use your own rule groups. For more information about all of these options, see AWS WAF rule groups.

If you want to use your own rule groups, you create those before you create your Firewall Manager AWS WAF policy. For guidance, see Managing your own rule groups. To use an individual custom rule, you must define your own rule group, define your rule within that, and then use the rule group in your policy.

The first and last AWS WAF rule groups that you manage through Firewall Manager have names that begin with `PREFMManaged-` or `POSTFMManaged-`, respectively, followed by the Firewall Manager policy name, and the rule group creation timestamp, in UTC milliseconds. For example, `PREFMManaged-MyWAFPolicyName-1621880555123`.

For information about how AWS WAF evaluates web requests, see Using protection packs (web ACLs) with rules and rule groups in AWS WAF.

Firewall Manager enables sampling and Amazon CloudWatch metrics for the rule groups that you define for the AWS WAF policy.

Individual account owners have complete control over the metrics and sampling configuration for any rule or rule group that they add to the policy's managed web ACLs.

> **Note**
>
> If you don't have a subscription to AWS WAF marketplace rule groups in your member account, Firewall Manager can't propagate custom or managed rule groups to that account.

Web ACL management for AWS WAF policies

Firewall Manager creates and manages web ACLs for in-scope resources according to your configuration settings and general policy management.

> **Note**
>
> If a resource that's configured with advanced automatic application layer DDoS mitigation by another Firewall Manager Shield policy comes into scope of an AWS WAF policy, where the web ACL on the resource was created by that Firewall Manager Shield policy, Firewall Manager will be unable to apply the AWS WAF policy protections to the resource and will mark the resource noncompliant.
> If a customer manually associates a customer-owned web ACL with the resource, the Firewall Manager AWS WAF policy will still override that customer web ACL with the Firewall Manager AWS WAF policy web ACL.

Manage unassociated web ACLs configuration

Policy configuration setting that specifies how Firewall Manager manages web ACLs for accounts when the web ACLs won't be used by any resource. If you enable management of unassociated web ACLs, Firewall Manager creates web ACLs in accounts that are within policy scope only if the web ACLs will be used by at least one resource. If you don't enable this option, Firewall Manager automatically ensures that each account has a web ACL regardless of whether the web ACL will be used.

When this is enabled, when an account comes into policy scope, Firewall Manager automatically creates a web ACL in the account only if at least one resource will use the web ACL.

Additionally, when you enable management of unassociated web ACLs, at policy creation, Firewall Manager performs a one-time cleanup of unassociated web ACLs in your account. During this cleanup, Firewall Manager skips any web ACLs that you've modified after their creation, for example, if you added a rule group to the web ACL or modified its settings. The cleanup process can take several hours. If a resource leaves policy scope after Firewall Manager creates a web ACL, Firewall Manager disassociates the resource from the web ACL, but won't clean up the unassociated web ACL. Firewall Manager only cleans up unassociated web ACLs when you first enable management of unassociated web ACLs in a policy.

In the API, this setting is `optimizeUnassociatedWebACL` in the SecurityServicePolicyData data type. Example: `\"optimizeUnassociatedWebACL\":false`

Web ACL source configuration: Create all new or retrofit existing?

Policy configuration setting that specifies what Firewall Manager does with existing web ACLs that are associated with in-scope resources.

By default, Firewall Manager creates all new web ACLs for in-scope resources. With retrofitting, Firewall Manager uses any existing web ACLs that are already in use, and only creates new web ACLs for resources that don't already have one associated.

When a policy is configured for retrofitting, all web ACLs that are associated with in-scope resources are retrofitted or marked noncompliant.

Firewall Manager only retrofits a web ACL if it satisfies the following requirements:

- The web ACL is owned by a customer account.
- The web ACL is only associated with in-scope resources.

Tip

Before you configure an AWS WAF policy for retrofitting, make sure that the web ACLs that are associated with the policy's in-scope resources aren't associated with any out-of-scope resources.

Tip

If you want to delete an associated resource, first disassociate it from the web ACL. If a web ACL is noncompliant due to an association with an out-of-scope resource, deleting the out-of-scope resource without first disassociating it from the web ACL can bring the web ACL into compliance, and Firewall Manager can then retrofit the web ACL through remediation, but the remediation in this situation can be delayed by up to 24 hours.

For information about accessing compliance violation details, see Viewing compliance information for an AWS Firewall Manager policy.

If a web ACL can be retrofitted, Firewall Manager modifies it as follows:

- Firewall Manager inserts the AWS WAF policy's first rule groups in front of the web ACL's existing rules and appends the AWS WAF policy's last rule groups at the end. For information about rule group management, see Rule group management for AWS WAF policies.
- If the policy has a logging configuration, then Firewall Manager adds it to the web ACL only if the web ACL isn't already configured for logging. If the web ACL has logging configured by the account, Firewall Manager leaves it in place both during the retrofitting and for any subsequent updates to the policy's logging configuration.
- Firewall Manager doesn't verify or configure any other web ACL properties. For example, Firewall Manager doesn't modify the web ACL's default action, custom request headers, CAPTCHA or Challenge configurations, or token domain lists. Firewall Manager only configures these other properties on web ACLs that Firewall Manager creates.

After Firewall Manager retrofits all existing associated web ACLs, for any in-scope resource that doesn't have a web ACL, Firewall Manager handles the resource following the default policy

behavior. If it's a resource that AWS WAF can protect, then Firewall Manager creates and associates a Firewall Manager web ACL with that resource.

In the API, the web ACL source setting is `webACLSource` in the SecurityServicePolicyData data type. Example: `\"webACLSource\":\"RETROFIT_EXISTING\"`

Sampling and CloudWatch metrics

AWS Firewall Manager enables sampling and Amazon CloudWatch metrics for the web ACLs and rule groups that it creates for an AWS WAF policy.

Web ACL naming

A web ACL that Firewall Manager creates is named after the AWS WAF policy as follows: `FMManagedWebACLV2-`*`policy name`*`-`*`timestamp`*. The timestamp is in UTC milliseconds. For example, `FMManagedWebACLV2-MyWAFPolicyName-1621880374078`.

A web ACL that Firewall Manager retrofits has the name that the customer account specified at creation. A web ACL name can't be changed after creation.

> **Note**
>
> If a resource configured with advanced automatic application layer DDoS mitigation comes into scope of a AWS WAF policy, Firewall Manager will be unable to associate the web ACL created by the AWS WAF policy to the resource.

Logging for an AWS WAF policy

You can enable centralized logging for your AWS WAF policies to get detailed information about traffic that's analyzed by your web ACL within your organization. AWS Firewall Manager supports this option for AWS WAFV2, not for AWS WAF Classic.

The information in the logs includes the time that AWS WAF received the request from your protected AWS resource, detailed information about the request, and the action for the rule that each request matched from all in-scope accounts. For information about AWS WAF logging, see Logging AWS WAF protection pack (web ACL) traffic in the *AWS WAF Developer Guide*.

You can send your logs to an Amazon Data Firehose data stream or Amazon Simple Storage Service (S3) bucket. Each destination type requires some additional configuration in order for Firewall

Manager to be able to manage the AWS WAF logging across your in-scope resources and accounts. The sections that follow provide details.

If the policy has web ACL retrofitting enabled, Firewall Manager doesn't override any logging configuration that's in place in existing web ACLs. For information about retrofitting, see the web ACL source configuration information at Web ACL management for AWS WAF policies.

> **Note**
>
> Only modify or disable logging for Firewall Manager policies through the Firewall Manager interface. If you use AWS WAF to update or delete the logging configuration of a web ACL that's managed by Firewall Manager, Firewall Manager won't detect the change automatically. If you have used AWS WAF, you can manually prompt an update to the Firewall Manager AWS WAF policy by re-evaluating the policy's rule in AWS Config. To do this, in the AWS Config console, locate the AWS Config rule for the Firewall Manager policy and select the re-evaluate action.

Topics

- Logging destinations
- Enabling logging for an AWS WAF policy in Firewall Manager
- Disabling logging for an AWS WAF policy in Firewall Manager

Logging destinations

This section describes the logging destinations that you can choose to send your AWS WAF policy logs. Each section provides guidance for configuring logging for the destination type and information about any behavior that's specific to the destination type. After you've configured your logging destination, you can provide its specifications to your Firewall Manager AWS WAF policy to start logging to it.

Firewall Manager doesn't have visibility into log failures after creating the logging configuration. It's your responsibility to verify that log delivery is working as you intended.

Firewall Manager doesn't modify any existing logging configurations in your organization's member accounts.

Topics

- Amazon Data Firehose data streams
- Amazon Simple Storage Service buckets

Amazon Data Firehose data streams

This topic provides information for sending your web ACL traffic logs to an Amazon Data Firehose data stream.

When you enable Amazon Data Firehose logging, Firewall Manager sends logs from your policy's web ACLs to an Amazon Data Firehose where you've configured a storage destination. After you enable logging, AWS WAF delivers logs for each configured web ACL, through the HTTPS endpoint of Kinesis Data Firehose to the configured storage destination. Before you use it, test your delivery stream to be sure that it has enough throughput to accommodate your organization's logs. For more information about how to create an Amazon Kinesis Data Firehose and review the stored logs, see What Is Amazon Data Firehose?

You must have the following permissions to successfully enable logging with a Kinesis:

- `iam:CreateServiceLinkedRole`
- `firehose:ListDeliveryStreams`
- `wafv2:PutLoggingConfiguration`

When you configure a Amazon Data Firehose logging destination on an AWS WAF policy, Firewall Manager creates a web ACL for the policy in the Firewall Manager administrator account as follows:

- Firewall Manager creates the web ACL in the Firewall Manager administrator account regardless of whether the account is in scope of the policy.
- The web ACL has logging enabled, with a log name `FMManagedWebACLV2-Logging`*`policy name`*`-`*`timestamp`*, where the timestamp is the UTC time that the log was enabled for the web ACL, in milliseconds. For example, `FMManagedWebACLV2-LoggingMyWAFPolicyName-1621880565180`. The web ACL has no rule groups and no associated resources.
- You are charged for the web ACL according to the AWS WAF pricing guidelines. For more information, see AWS WAF Pricing.
- Firewall Manager deletes the web ACL when you delete the policy.

For information about service-linked roles and the `iam:CreateServiceLinkedRole` permission, see Using service-linked roles for AWS WAF.

For more information about creating your delivery stream, see Creating an Amazon Data Firehose Delivery Stream.

Amazon Simple Storage Service buckets

This topic provides information for sending your web ACL traffic logs to an Amazon S3 bucket.

The bucket that you choose as your logging destination must be owned by a Firewall Manager administrator account. For information about the requirements for creating your Amazon S3 bucket for logging and bucket naming requirements, see Amazon Simple Storage Service in the *AWS WAF Developer Guide*.

Eventual consistency

When you make change to AWS WAF policies configured with an Amazon S3 logging destination, Firewall Manager updates the bucket policy to add the permissions necessary for logging. When doing so, Firewall Manager follows the last-writer-wins semantics and data consistency models that Amazon Simple Storage Service follows. If you concurrently make multiple policy updates to a Amazon S3 destination in the Firewall Manager console or through the PutPolicy API, some permissions may not be saved. For more information about the Amazon S3 data consistency model, see Amazon S3 data consistency model in the *Amazon Simple Storage Service User Guide*.

Permissions to publish logs to an Amazon S3 bucket

Configuring web ACL traffic logging for an Amazon S3 bucket in an AWS WAF policy requires the following permissions settings. Firewall Manager automatically attaches these permissions to your Amazon S3 bucket when you configure Amazon S3 as your logging destination to give the service permission to publish logs to the bucket. If you want to manage finer-grained access to your logging and Firewall Manager resources, you can set these permissions yourself. For information about managing permissions, see Access management for AWS resources in the *IAM User Guide*. For information about the AWS WAF managed policies, see AWS managed policies for AWS WAF.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "AWSLogDeliveryForFirewallManager",
```

```
    "Statement": [
        {
            "Sid": "AWSLogDeliveryAclCheckFMS",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws-eusc:s3:::aws-waf-logs-amzn-s3-demo-destination-
bucket-suffix"
        },
        {
            "Sid": "AWSLogDeliveryWriteFMS",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws-eusc:s3:::aws-waf-logs-amzn-s3-demo-destination-
bucket-suffix/policy-id/AWSLogs/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
```

To prevent the cross-service confused deputy problem, you can add the `aws:SourceArn` and `aws:SourceAccount` global condition context keys to your bucket's policy. To add these keys, you can either modify the policy that Firewall Manager creates for you when you configure the logging destination, or if you want fine grained control, you can create your own policy. If you add these conditions to your logging destination policy, Firewall Manager won't validate or monitor the confused deputy protections. For general information about the confused deputy problem, see The confused deputy problem
in the *IAM User Guide*.

When you add the `sourceAccount` add `sourceArn` properties it'll increase the bucket policy size. If you're adding a long list of `sourceAccount` add `sourceArn` properties, take care not to exceed the Amazon S3 bucket policy size quota.

The following example shows how to prevent the confused deputy problem by using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in your bucket's policy. Replace *`member-account-id`* with the account IDs of the members in your organization.

JSON

```
{
   "Version":"2012-10-17",
   "Id":"AWSLogDeliveryForFirewallManager",
   "Statement":[
      {
         "Sid":"AWSLogDeliveryAclCheckFMS",
         "Effect":"Allow",
         "Principal":{
            "Service":"delivery.logs.amazonaws.com"
         },
         "Action":"s3:GetBucketAcl",
         "Resource":"arn:aws-eusc:s3:::aws-waf-logs-amzn-s3-demo-destination-
bucket-suffix",
         "Condition":{
            "StringEquals":{
               "aws:SourceAccount":[
               "111122223333",
               "444455556666"
               ]
            },
            "ArnLike":{
               "aws:SourceArn":[
               "arn:aws-eusc:logs:*:111122223333:*",
               "arn:aws-eusc:logs:*:444455556666:*"
               ]
            }
         }
      },
      {
         "Sid":"AWSLogDeliveryWriteFMS",
         "Effect":"Allow",
         "Principal":{
            "Service":"delivery.logs.amazonaws.com"
         },
         "Action":"s3:PutObject",
```

```
         "Resource":"arn:aws-eusc:s3:::aws-waf-logs-amzn-s3-demo-destination-
bucket-suffix/policy-id/AWSLogs/*",
         "Condition":{
            "StringEquals":{
               "s3:x-amz-acl":"bucket-owner-full-control",
               "aws:SourceAccount":[
               "111122223333",
               "444455556666"
               ]
            },
            "ArnLike":{
               "aws:SourceArn":[
               "arn:aws-eusc:logs:*:111122223333:*",
               "arn:aws-eusc:logs:*:444455556666:*"
               ]
            }
         }
      }
   ]
}
```

Server-side encryption for Amazon S3 buckets

You can enable Amazon S3 server-side encryption or use a AWS Key Management Service customer managed key on your S3 bucket. If you choose to use the default Amazon S3 encryption on your Amazon S3 bucket for AWS WAF logs, you don't need to take any special action. However, if you choose to use a customer-provided encryption key to encrypt your Amazon S3 data at rest, you must add the following permission statement to your AWS Key Management Service key policy:

```
{
            "Sid": "Allow Logs Delivery to use the key",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
```

```
            "Resource": "*"
}
```

For information about using customer-provided encryption keys with Amazon S3, see Using server-side encryption with customer-provided keys (SSE-C) in the *Amazon Simple Storage Service User Guide*.

Enabling logging for an AWS WAF policy in Firewall Manager

The following procedure describes how to enable logging for an AWS WAF policy in the Firewall Manager console.

To enable logging for an AWS WAF policy

1. Before you can enable logging, you must configure your logging destination resources as the following:

   - **Amazon Kinesis Data Streams** - Create an Amazon Data Firehose using your Firewall Manager administrator account. Use a name starting with the prefix `aws-waf-logs-`. For example, `aws-waf-logs-firewall-manager-central`. Create the data firehose with a PUT source and in the Region that you are operating. If you are capturing logs for Amazon CloudFront, create the firehose in US East (N. Virginia). Before you use it, test your delivery stream to be sure that it has enough throughput to accommodate your organization's logs. For more information, see Creating an Amazon Data Firehose delivery stream.
   - **Amazon Simple Storage Service buckets** - Create an Amazon S3 bucket according to the guidelines in the Amazon Simple Storage Service topic in the *AWS WAF Developer Guide*. You must also configure your Amazon S3 bucket with the permissions listed in Permissions to publish logs to an Amazon S3 bucket.
2. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

3. In the navigation pane, choose **Security Policies**.

4. Choose the AWS WAF policy that you want to enable logging for. For more information about AWS WAF logging, see Logging AWS WAF protection pack (web ACL) traffic.
5. On the **Policy details** tab, in the **Policy rules** section, choose **Edit**.
6. For **Logging configuration**, choose **Enable logging** to turn on logging. Logging provides detailed information about traffic that is analyzed by your web ACL. Choose the **Logging destination**, and then choose the logging destination that you configured. You must choose a logging destination whose name begins with `aws-waf-logs-`. For information about configuring an AWS WAF logging destination, see Using AWS WAF policies with Firewall Manager.
7. (Optional) If you don't want certain fields and their values included in the logs, redact those fields. Choose the field to redact, and then choose **Add**. Repeat as necessary to redact additional fields. The redacted fields appear as `REDACTED` in the logs. For example, if you redact the **URI** field, the **URI** field in the logs will be `REDACTED`.
8. (Optional) If you don't want to send all requests to the logs, add your filtering criteria and behavior. Under **Filter logs**, for each filter that you want to apply, choose **Add filter**, then choose your filtering criteria and specify whether you want to keep or drop requests that match the criteria. When you finish adding filters, if needed, modify the **Default logging behavior**. For more information, see Finding your protection pack (web ACL) records in the *AWS WAF Developer Guide*.
9. Choose **Next**.
10. Review your settings, then choose **Save** to save your changes to the policy.

Disabling logging for an AWS WAF policy in Firewall Manager

The following procedure describes how to disable logging for an AWS WAF policy in the Firewall Manager console.

To disable logging for an AWS WAF policy

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

> **Note**
>
> For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security Policies**.
3. Choose the AWS WAF policy that you want to disable logging for.
4. On the **Policy details** tab, in the **Policy rules** section, choose **Edit**.
5. For **Logging configuration status**, choose **Disabled**.
6. Choose **Next**.
7. Review your settings, then choose **Save** to save your changes to the policy.

> **Note**
>
> Only modify or disable logging for Firewall Manager policies through the Firewall Manager interface. If you use AWS WAF to update or delete the logging configuration of a web ACL that's managed by Firewall Manager, Firewall Manager won't detect the change automatically. If you have used AWS WAF, you can manually prompt an update to the Firewall Manager AWS WAF policy by re-evaluating the policy's rule in AWS Config. To do this, in the AWS Config console, locate the AWS Config rule for the Firewall Manager policy and select the re-evaluate action.

Using AWS Shield Advanced policies in Firewall Manager

This page explains how to use AWS Shield policies with Firewall Manager. In a Firewall Manager AWS Shield policy, you choose the resources that you want to protect. When you apply the policy with auto remediation enabled, for each in-scope resource that's not already associated with a AWS WAF web ACL, Firewall Manager associates an empty AWS WAF web ACL. The empty web ACL is used for Shield monitoring purposes. If you then associate any other web ACL to the resource, Firewall Manager removes the empty web ACL association.

> **Note**
>
> When a resource that's in scope of an AWS WAF policy comes into the scope of a Shield Advanced policy configured with automatic application layer DDoS mitigation, Firewall Manager applies the Shield Advanced protection only after associating the web ACL created by the AWS WAF policy.

How AWS Firewall Manager manages unassociated web ACLs in Shield policies

You can configure whether Firewall Manager manages unassociated web ACLs for you through the **Manage unassociated web ACLs** setting in your policy, or the `optimizeUnassociatedWebACLs` setting in the SecurityServicePolicyData data type in the API. If you enable management of unassociated web ACLs in your policy, Firewall Manager creates web ACLs in the accounts within policy scope only if the web ACLs will be used by at least one resource. If at any time an account comes into policy scope, Firewall Manager automatically creates a web ACL in the account if at least one resource will use the web ACL.

When you enable management of unassociated web ACLs, Firewall Manager performs a one-time cleanup of unassociated web ACLs in your account. The cleanup process can take several hours. If a resource leaves policy scope after Firewall Manager creates a web ACL, Firewall Manager doesn't disassociate the resource from the web ACL. If you want Firewall Manager to clean up the web ACL, you must first manually disassociate the resources from the web ACL, and then enable the manage unassociated web ACLs option in your policy.

If you don't enable this option, Firewall Manager doesn't manage unassociated web ACLs, and Firewall Manager automatically creates a web ACL in each account that's within policy scope.

How AWS Firewall Manager manages scope changes in Shield policies

Accounts and resources can go out of scope of an AWS Firewall Manager Shield Advanced policy due to a number of changes, such as changes to policy scope settings, changes to the tags on a resource, and the removal of an account from an organization. For general information about policy scope settings, see Using the AWS Firewall Manager policy scope.

With an AWS Firewall Manager Shield Advanced policy, if an account or resource goes out of scope, Firewall Manager stops monitoring the account or resource.

If an account goes out of scope by being removed from the organization, it will continue to be subscribed to Shield Advanced. Because the account is no longer part of the consolidated billing

family, the account will incur a prorated Shield Advanced subscription fee. On the other hand, an account that goes out of scope but remains in the organization doesn't incur additional fees.

For Shield policy using Classic WebACL, if a resource goes out of scope, it continues to be protected by Shield Advanced and continues to incur Shield Advanced data transfer charges.

Using Automatic application layer DDoS mitigation with Firewall Manager Shield Advanced policies

This page explains how Automatic application layer DDoS mitigation works with Firewall Manager.

When you apply a Shield Advanced policy to Amazon CloudFront distributions or Application Load Balancers, you have the option of configuring Shield Advanced automatic application layer DDoS mitigation in the policy.

For information about Shield Advanced automatic mitigation, see Automating application layer DDoS mitigation with Shield Advanced.

Shield Advanced automatic application layer DDoS mitigation has the following requirements:

- Automatic application layer DDoS mitigation works only with Amazon CloudFront distributions and Application Load Balancers.

  If applying your Shield Advanced policy to Amazon CloudFront distributions, you can choose this option for Shield Advanced policies that you create for the **Global** Region. If applying protections to Application Load Balancers, you can apply the policy to any Region that Firewall Manager supports.
- Automatic application layer DDoS mitigation works only with protection packs (web ACLs) that were created using the latest version of AWS WAF (v2).

  Because of this, if you have a policy that uses AWS WAF Classic web ACLs, you need to either replace the policy with a new policy, which will automatically use the latest version of AWS WAF, or have Firewall Manager create new version web ACLs for your existing policy and switch over to using them. For more information about the options, see Replace AWS WAF Classic web ACLs with latest version web ACLs.

Automatic mitigation configuration

The automatic application layer DDoS mitigation option for Firewall Manager Shield Advanced policies applies Shield Advanced automatic mitigation functionality to your policy's in-scope

accounts and resources. For detailed information about this Shield Advanced feature, see Automating application layer DDoS mitigation with Shield Advanced.

You can choose to have Firewall Manager enable or disable automatic mitigation for the CloudFront distributions or Application Load Balancers that are in scope of the policy, or you can choose to have the policy ignore Shield Advanced automatic mitigation settings:

- **Enable** – If you choose to enable automatic mitigation, you also specify whether mitigating Shield Advanced rules should count or block matching web requests. Firewall Manager will mark in-scope resources as noncompliant if they either don't have automatic mitigation enabled, or are using a rule action that doesn't match the one you specify for the policy. If you configure the policy for automatic remediation, Firewall Manager updates noncompliant resources as needed.
- **Disable** – If you choose to disable automatic mitigation, Firewall Manager will mark in-scope resources as noncompliant if they have automatic mitigation enabled. If you configure the policy for automatic remediation, Firewall Manager updates noncompliant resources as needed.
- **Ignore** – If you choose to ignore automatic mitigation, Firewall Manager won't consider any of the automatic mitigation settings in your Shield policy when it performs remediation activities for the policy. This setting allows you to control automatic mitigation through Shield Advanced, without having those settings overwritten by Firewall Manager. This setting doesn't apply to any Classic Load Balancers or Elastic IPs resources manged through Shield Advanced, because Shield Advanced doesn't currently support L7 automatic mitigation for those resources.

Replace AWS WAF Classic web ACLs with latest version web ACLs

Automatic application layer DDoS mitigation works only with protection packs (web ACLs) that were created using the latest version of AWS WAF (v2).

To determine the web ACL version for your Shield Advanced policy, see Determining the version of AWS WAF that's used by a Shield Advanced policy.

If you want to use automatic mitigation in your Shield Advanced policy, and your policy currently uses AWS WAF Classic web ACLs, you can either create a new Shield Advanced policy to replace your current one, or you can use the options described in this section to replace earlier version web ACLs with new (v2) web ACLs inside your current Shield Advanced policy. New policies always create web ACLs using the latest version of AWS WAF. If you replace the entire policy, when you delete it, you can have Firewall Manager delete all of the earlier version web ACLs as well. The rest of this section describes your options for replacing the web ACLs inside your existing policy.

When you modify an existing Shield Advanced policy for Amazon CloudFront resources, Firewall Manager can automatically create a new empty AWS WAF (v2) web ACL for the policy, in any in-scope account that doesn't already have a v2 web ACL. When Firewall Manager creates a new web ACL, if the policy already has an AWS WAF Classic web ACL in the same account, Firewall Manager configures the new version web ACL with the same default action setting as the existing web ACL. If there is no existing AWS WAF Classic web ACL, Firewall Manager sets the default action to Allow in the new web ACL. After Firewall Manager creates a new web ACL, you can customize it as needed through the AWS WAF console.

When you choose any of the following policy configuration options, Firewall Manager creates new (v2) web ACLs for in-scope accounts that don't already have them:

- When you enable or disable automatic application layer DDoS mitigation. This choice alone only causes Firewall Manager to create the new web ACLs, and not to replace any existing AWS WAF Classic web ACL associations on the policy's in-scope resources.
- When you choose the policy action of automatic remediation and you choose the option to replace AWS WAF Classic web ACLs with AWS WAF (v2) web ACLs. You can choose to replace earlier version web ACLs regardless of your configuration choices for automatic application layer DDoS mitigation.

  When you choose the replacement option, Firewall Manager creates the new version web ACLs as needed and then does the following for the policy's in-scope resources:
  - If a resource is associated with a web ACL from any other active Firewall Manager policy, Firewall Manager leaves the association alone.
  - For any other case, Firewall Manager removes any association with an AWS WAF Classic web ACL and associates the resource with the policy's AWS WAF (v2) web ACL.

You can choose to have Firewall Manager replace the earlier version web ACLs with the new version web ACLs when you want to. If you've previously customized the policy's AWS WAF Classic web ACLs, you can update new version web ACLs to comparable settings before you choose to have Firewall Manager perform the replacement step.

You can access either version of web ACL for a policy through the same-version console for AWS WAF or AWS WAF Classic.

Firewall Manager doesn't delete any replaced AWS WAF Classic web ACLs until you delete the policy itself. After the AWS WAF Classic web ACLs are no longer used by the policy, you can delete them if you want to.

Determining the version of AWS WAF that's used by a Shield Advanced policy

This page explains how to determine which version of AWS WAF web ACL your Shield Advanced policy uses.

You can determine which version of AWS WAF your Firewall Manager Shield Advanced policy uses by looking at the parameter keys in the policy's AWS Config service-linked rule. If the AWS WAF version that's in use is the latest, the parameter keys include `policyId` and `webAclArn`. If it's the earlier version, AWS WAF Classic, the parameter keys include `webAclId` and `resourceTypes`.

The AWS Config rule only lists keys for the web ACLs that the policy is currently using with in-scope resources.

To determine which version of AWS WAF your Firewall Manager Shield Advanced policy uses

1. Retrieve the policy ID for the Shield Advanced policy:

   a. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   b. In the navigation pane, choose **Security Policies**.

   c. Choose the Region for the policy. For CloudFront distributions, this is `Global`.

   d. Find the policy that you want and copy the value of its **Policy ID**.

      Example policy ID: `1111111-2222-3333-4444-a55aa5aaa555`.

2. Create the policy's AWS Config rule name by appending the policy ID to the string `FMManagedShieldConfigRule`.

   Example AWS Config rule name: `FMManagedShieldConfigRule1111111-2222-3333-4444-a55aa5aaa555`.

3. Search the parameters for the associated AWS Config rule for keys named `policyId` and `webAclArn`:

   a. Open the AWS Config console at https://eusc-de-east-1.console.amazonaws-eusc.eu/config/home.

   b. In the navigation pane, choose **Rules**.

c. Find your Firewall Manager policy's AWS Config rule name in the list and select it. The rule's page opens.

d. Under **Rule details**, in the **Parameters** section, look at the keys. If you find keys named `policyId` and `webAclArn`, the policy uses web ACLs that were created using the latest version of AWS WAF. If you find keys named `webAclId` and `resourceTypes`, the policy uses web ACLs that were created using the earlier version, AWS WAF Classic.

Using security group policies in Firewall Manager to manage Amazon VPC security groups

This page explains how to use AWS Firewall Manager security group policies to manage Amazon Virtual Private Cloud security groups for your organization in AWS Organizations. You can apply centrally controlled security group policies to your entire organization or to a select subset of your accounts and resources. You can also monitor and manage the security group policies that are in use in your organization, with auditing and usage security group policies.

Firewall Manager continuously maintains your policies and applies them to accounts and resources as they are added or updated across your organization. For information about AWS Organizations, see AWS Organizations User Guide.

For information about Amazon Virtual Private Cloud security groups, see Security Groups for Your VPC in the *Amazon VPC User Guide*.

You can use Firewall Manager security group policies to do the following across your AWS organization:

- Apply common security groups to specified accounts and resources.
- Audit security group rules, to locate and remediate noncompliant rules.
- Audit usage of security groups, to clean up unused and redundant security groups.

This section covers how Firewall Manager security groups policies work and provides guidance for using them. For procedures to create security group policies, see Creating an AWS Firewall Manager policy.

Best practices for security group policies

This section lists recommendations for managing security groups using AWS Firewall Manager.

Exclude the Firewall Manager administrator account

When you set the policy scope, exclude the Firewall Manager administrator account. When you create a usage audit security group policy through the console, this is the default option.

Start with automatic remediation disabled

For content or usage audit security group policies, start with automatic remediation disabled. Review the policy details information to determine the effects that automatic remediation would have. When you are satisfied that the changes are what you want, edit the policy to enable automatic remediation.

Avoid conflicts if you also use outside sources to manage security groups

If you use a tool or service other than Firewall Manager to manage security groups, take care to avoid conflicts between your settings in Firewall Manager and the settings in your outside source. If you use automatic remediation and your settings conflict, you can create a cycle of conflicting remediation that consumes resources on both sides.

For example, say you configure another service to maintain a security group for a set of AWS resources, and you configure a Firewall Manager policy to maintain a different security group for some or all of the same of resources. If you configure either side to disallow any other security group to be associated with the in-scope resources, that side will remove the security group association that's maintained by the other side. If both sides are configured in this way, you can end up with a cycle of conflicting disassociations and associations.

Additionally, say that you create a Firewall Manager audit policy to enforce a security group configuration that conflicts with the security group configuration from the other service. Remediation applied by the Firewall Manager audit policy can update or delete that security group, putting it out of compliance for the other service. If the other service is configured to monitor and automatically remediate any problems it finds, it will recreate or update the security group, putting it again out of compliance with the Firewall Manager audit policy. If the Firewall Manager audit policy is configured with automatic remediation, it will again update or delete the outside security group, and so on.

To avoid conflicts like these, create configurations that are mutually exclusive, between Firewall Manager and any outside sources.

You can use tagging to exclude outside security groups from automatic remediation by your Firewall Manager policies. To do this, add one or more tags to the security groups or other

resources that are managed by the outside source. Then, when you define the Firewall Manager policy scope, in your resources specification, exclude resources that have the tag or tags that you've added.

Similarly, in your outside tool or service, exclude the security groups that Firewall Manager manages from any management or auditing activities. Either don't import the Firewall Manager resources or use Firewall Manager-specific tagging to exclude them from outside management.

Best practices for usage audit security group policies

Follow these guidelines when you use usage audit security group policies.

- Avoid making multiple changes to the association status of a security group in a short amount of time, such as within a 15-minute window. Doing so can cause Firewall Manager to miss some or all of the corresponding events. For example, don't quickly associate and disassociate a security group with an elastic network interface.

Security group policy caveats and limitations

This section lists the caveats and limitations for using Firewall Manager security group policies.

Resource type: Amazon EC2 instance

This section lists the caveats and limitations for protecting Amazon EC2 instances with Firewall Manager security group policies.

- With security groups that protect Amazon EC2 elastic network interfaces (ENIs), changes to a security group aren't immediately visible to Firewall Manager. Firewall Manager usually detects changes within several hours, but detection can be delayed as much as six hours.
- Firewall Manager doesn't support security groups for Amazon EC2 ENIs that were created by the Amazon Relational Database Service.
- Firewall Manager doesn't support updating security groups for Amazon EC2 ENIs that were created using the Fargate service type. You can, however, update security groups for Amazon ECS ENIs with the Amazon EC2 service type.
- Firewall Manager doesn't support updating security groups for requester-managed Amazon EC2 ENIs, because Firewall Manager doesn't have permission to modify them.
- For common security group policies, these caveats concern the interaction between the number of elastic network interfaces (ENIs) that are attached to the EC2 instance and the policy option

that specifies whether to remediate only EC2 instances with no added attachments or to remediate all instances. Every EC2 instance has a default primary ENI, and you can attach more ENIs. In the API, the policy option setting for this choice is `ApplyToAllEC2InstanceENIs`.

If an in-scope EC2 instance has additional ENIs attached and the policy is configured to include only EC2 instances with just the primary ENI, then Firewall Manager won't attempt any remediation for the EC2 instance. Additionally, If the instance goes out of policy scope, Firewall Manager doesn't attempt to disassociate any security group associations that it might have established for the instance.

For the following edge cases, during resource cleanup, Firewall Manager can leave replicated security group associations intact, regardless of the policy's resource cleanup specifications:

- When an instance with additional ENIs was previously remediated by a policy that was configured to include all EC2 instances, and then either the instance went out of policy scope or the policy setting was changed to include only instances without additional ENIs.
- When an instance with no additional ENIs was remediated by an policy that was configured to include only instances with no additional ENIs, then another ENI was attached to the instance, and then the instance went out of policy scope.

Other caveats and limitations

The following are miscellaneous caveats and limitations for Firewall Manager security group policies.

- Firewall Manager security group policies do not support security groups shared through AWS RAM.
- The use of AWS RAM to share prefix lists across an organization is not an officially supported feature of Firewall Manager. While it may happen to work today, there is neither an obligation for AWS to offer support for that use case nor any guarantee that it will continue to work.
- Updating Amazon ECS ENIs is possible only for Amazon ECS services that use the rolling update (Amazon ECS) deployment controller. For other Amazon ECS deployment controllers such as CODE_DEPLOY or external controllers, Firewall Manager currently can't update the ENIs.
- Firewall Manager doesn't support updating security groups in ENIs for Network Load Balancers.
- In common security group policies, if a shared VPC is later unshared with an account Firewall Manager won't delete the replica security groups in the account.

- With usage audit security group policies, if you create multiple policies with a custom delay time setting that all have the same scope, the first policy with compliance findings will be the policy that reports the findings.

Security group policy use cases

You can use AWS Firewall Manager common security group policies to automate the host firewall configuration for communication between Amazon VPC instances. This section lists standard Amazon VPC architectures and describes how to secure each using Firewall Manager common security group policies. These security group policies can help you apply a unified set of rules to select resources in different accounts and avoid per-account configurations in Amazon Elastic Compute Cloud and Amazon VPC.

With Firewall Manager common security group policies, you can tag just the EC2 elastic network interfaces that you need for communication with instances in another Amazon VPC. The other instances in the same Amazon VPC are then more secure and isolated.

Use case: Monitoring and controlling requests to Application Load Balancers and Classic Load Balancers

You can use a Firewall Manager common security group policy to define which requests your in-scope load balancers should serve. You can configure this through the Firewall Manager console. Only requests that comply with the security group's inbound rules can reach your load balancers, and the load balancers will only distribute requests that meet the outbound rules.

Use case: Internet-accessible, public Amazon VPC

You can use a Firewall Manager common security group policy to secure a public Amazon VPC, for example, to allow only inbound port 443. This is the same as only allowing inbound HTTPS traffic for a public VPC. You can tag public resources within the VPC (for example, as "PublicVPC"), and then set the Firewall Manager policy scope to only resources with that tag. Firewall Manager automatically applies the policy to those resources.

Use case: Public and Private Amazon VPC instances

You can use the same common security group policy for public resources as recommended in the prior use case for internet-accessible, public Amazon VPC instances. You can use a second common security group policy to limit communication between the public resources and the private ones. Tag the resources in the public and private Amazon VPC instances with something

like "PublicPrivate" to apply the second policy to them. You can use a third policy to define the allowed communication between the private resources and other corporation or private Amazon VPC instances. For this policy, you can use another identifying tag on the private resources.

Use case: Hub and spoke Amazon VPC instances

You can use a common security group policy to define communications between the hub Amazon VPC instance and spoke Amazon VPC instances. You can use a second policy to define communication from each spoke Amazon VPC instance to the hub Amazon VPC instance.

Use case: Default network interface for Amazon EC2 instances

You can use a common security group policy to allow only standard communications, for example internal SSH and patch/OS update services, and to disallow other insecure communication.

Use case: Identify resources with open permissions

You can use an audit security group policy to identify all resources within your organization that have permission to communicate with public IP addresses or that have IP addresses that belong to third-party vendors.

Using common security group policies with Firewall Manager

This page explains how Firewall Manager common security group policies work.

With a common security group policy, Firewall Manager provides a centrally controlled association of security groups to accounts and resources across your organization. You specify where and how to apply the policy in your organization.

You can apply common security group policies to the following resource types:

- Amazon Elastic Compute Cloud (Amazon EC2) instance
- Elastic Network Interface
- Application Load Balancer
- Classic Load Balancer

For guidance on creating a common security group policy using the console, see Creating a common security group policy.

Shared VPCs

In the policy scope settings for a common security group policy, you can choose to include shared VPCs. This choice includes VPCs that are owned by another account and shared with an in-scope account. VPCs that in-scope accounts own are always included. For information about shared VPCs, see Working with shared VPCs in the *Amazon VPC User Guide*.

The following caveats apply to including shared VPCs. These are in addition to the general caveats for security group policies at Security group policy caveats and limitations.

- Firewall Manager replicates the primary security group into the VPCs for each in-scope account. For a shared VPC, Firewall Manager replicates the primary security group once for each in-scope account that the VPC is shared with. This can result in multiple replicas in a single shared VPC.
- When you create a new shared VPC, you won't see it represented in the Firewall Manager security group policy details until after you create at least one resource in the VPC that's within the scope of the policy.
- When you disable shared VPCs in a policy that had shared VPCs enabled, in the shared VPCs, Firewall Manager deletes the replica security groups that aren't associated with any resources. Firewall Manager leaves the remaining replica security groups in place, but stops managing them. Removal of these remaining security groups requires manual management in each shared VPC instance.

Primary security groups

For each common security group policy, you provide AWS Firewall Manager with one or more primary security groups:

- Primary security groups must be created by the Firewall Manager administrator account and can reside in any Amazon VPC instance in the account.
- You manage your primary security groups through Amazon Virtual Private Cloud (Amazon VPC) or Amazon Elastic Compute Cloud (Amazon EC2). For information, see Working with Security Groups in the *Amazon VPC User Guide*.
- You can name one or more security groups as primaries for a Firewall Manager security group policy. By default, the number of security groups allowed in a policy is one, but you can submit a request to increase it. For information, see AWS Firewall Manager quotas.
- Primary security groups must contain at least one ingress or egress rule. If you remove all rules from a primary security group, Firewall Manager continues to manage replica security groups based on the last version of the primary security groups that had rules.

Policy rules settings

You can choose one or more of the following change control behaviors for the security groups and resources of your common security group policy:

- Identify and report on any changes made by local users to replica security groups.
- Disassociate any other security groups from the AWS resources that are within the policy scope.
- Distribute tags from the primary group to the replica security groups.

> **Important**
>
> Firewall Manager won't distribute system tags added by AWS services into the replica security groups. System tags begin with the `aws:` prefix. Additionally, Firewall Manager won't update the tags of existing security groups or create new security groups if the policy has tags that conflict with the organization's tag policy. For information about tag policies, see Tag policies in the AWS Organizations User Guide.

- Distribute security group references from the primary group to the replica security groups.

  This enables you to easily establish common security group referencing rules across all in-scope resources to instances associated with the specified security group's VPC. When you enable this option, Firewall Manager only propagates the security group references if the security groups reference peer security groups in Amazon Virtual Private Cloud. If the replica security groups don't correctly reference the peer security group, Firewall Manager marks these replicated security groups as non-compliant. For information about how to reference peer security groups in Amazon VPC, see Update your security groups to reference peer security groups in the Amazon VPC Peering Guide.

  If you don't enable this option, Firewall Manager doesn't propagate security group references to the replica security groups. For information about VPC peering in Amazon VPC, see the Amazon VPC Peering Guide.

Policy creation and management

When you create your common security group policy, Firewall Manager replicates the primary security groups to every Amazon VPC instance within the policy scope, and associates the replicated security groups to accounts and resources that are in scope of the policy. When you modify a primary security group, Firewall Manager propagates the change to the replicas.

When you delete a common security group policy, you can choose whether to clean up the resources created by the policy. For Firewall Manager common security groups, these resources are the replica security groups. Choose the cleanup option unless you want to manually manage each individual replica after the policy is deleted. For most situations, choosing the cleanup option is the simplest approach.

How replicas are managed

The replica security groups in the Amazon VPC instances are managed like other Amazon VPC security groups. For information, see Security Groups for Your VPC in the *Amazon VPC User Guide*.

Using content audit security group policies with Firewall Manager

This page explains how Firewall Manager content audit security group policies work.

Use AWS Firewall Manager content audit security group policies to audit and apply policy actions to the rules that are in use in your organization's security groups. Content audit security group policies apply to all customer-created security groups in use in your AWS organization, according to the scope that you define in the policy.

For guidance on creating a content audit security group policy using the console, see Creating a content audit security group policy.

Policy scope resource type

You can apply content audit security group policies to the following resource types:

- Amazon Elastic Compute Cloud (Amazon EC2) instance
- Elastic Network Interface
- Amazon VPC security group

Security groups are considered in scope of the policy if they explicitly are in scope or if they're associated with resources that are in scope.

Policy rule options

You can use either managed policy rules or custom policy rules for each content audit policy, but not both.

- **Managed policy rules** – In a policy with managed rules, you can use application and protocol lists to control which rules that Firewall Manager audits and either marks as compliant or non-

compliant. You can use lists that are managed by Firewall Manager. You can also create and use your own application and protocol lists. For information about these types of lists and your management options for custom lists, see Using Firewall Manager managed lists.

- **Custom policy rules** – In a policy with custom policy rules, you specify an existing security group as the audit security group for your policy. You can use the audit security group rules as a template that defines the rules that Firewall Manager audits and either marks as compliant or non-compliant.

Audit security groups

You must create audit security groups using your Firewall Manager administrator account, before you can use them in your policy. You can manage security groups through Amazon Virtual Private Cloud (Amazon VPC) or Amazon Elastic Compute Cloud (Amazon EC2). For information, see Working with Security Groups in the *Amazon VPC User Guide*.

A security group that you use for a content audit security group policy is used by Firewall Manager only as a comparison reference for the security groups that are in scope of the policy. Firewall Manager doesn't associate it with any resources in your organization.

The way that you define the rules in the audit security group depends on your choices in the policy rules settings:

- **Managed policy rules** – For managed policy rules settings, you use an audit security group to override other settings in the policy, to explicitly allow or deny rules that otherwise might have another compliance outcome.
  - If you choose to always *allow* the rules that are defined in the audit security group, any rule that matches one that's defined in the audit security group is considered *compliant* with the policy, regardless of the other policy settings.
  - If you choose to always *deny* the rules that are defined in the audit security group, any rule that matches one that's defined in the audit security group is considered *noncompliant* with the policy, regardless of the other policy settings.
- **Custom policy rules** – For custom policy rules settings, the audit security group provides the example of what is acceptable or not acceptable in the in-scope security group rules:
  - If you choose to *allow* the use of the rules, all in-scope security groups must only have rules that are *within* the allowed range of the policy's audit security group rules. *In this case, the policy's security group rules provide the example of what's acceptable to do.*

- If you choose to *deny* the use of the rules, all in-scope security groups must only have rules that are *not within* the allowed range of the policy's audit security group rules. *In this case, the policy's security group provides the example of what's not acceptable to do.*

Policy creation and management

When you create an audit security group policy, you must have automatic remediation disabled. The recommended practice is to review the effects of policy creation before enabling automatic remediation. After you review the expected effects, you can edit the policy and enable automatic remediation. When automatic remediation is enabled, Firewall Manager updates or removes rules that are noncompliant in in-scope security groups.

Security groups affected by an audit security group policy

All security groups in your organization that are customer-created are eligible to be in scope of an audit security group policy.

Replica security groups are not customer-created and so aren't eligible to be directly in scope of an audit security group policy. However, they can be updated as a result of the policy's automatic remediation activities. A common security group policy's primary security group is customer-created and can be in scope of an audit security group policy. If an audit security group policy makes changes to a primary security group, Firewall Manager automatically propagates those changes to the replicas.

Caveats and limitations for content audit security group policies

You cannot reference peer security groups in a content audit security group policy.

For information on other considerations across all Firewall Manager security groups, see Security group policy caveats and limitations.

Using usage audit security group policies with Firewall Manager

This page explains how Firewall Manager usage audit security group policies work.

Use AWS Firewall Manager usage audit security group policies to monitor your organization for unused and redundant security groups and optionally perform cleanup. When you enable automatic remediation for this policy, Firewall Manager does the following:

1. Consolidates redundant security groups, if you've chosen that option.

2. Removes unused security groups, if you've chosen that option.

You can apply usage audit security group policies to the following resource type:

- Amazon VPC security group

For guidance on creating a usage audit security group policy using the console, see Creating a usage audit security group policy.

How Firewall Manager detects and remediates redundant security groups

For security groups to be considered redundant, they must have exactly the same rules set and be in the same Amazon VPC instance.

To remediate a redundant security group set, Firewall Manager selects one of the security groups in the set to keep, and then associates it to all resources that are associated with the other security groups in the set. Firewall Manager then disassociates the other security groups from the resources they were associated with, which renders them unused.

> **Note**
>
> If you have also chosen to remove unused security groups, Firewall Manager does that next. This can result in the removal of the security groups that are in the redundant set.

How Firewall Manager detects and remediates unused security groups

Firewall Manager considers a security group to be unused if both of the following are true:

- The security group is not used by any Amazon EC2 instance or Amazon EC2 elastic network interface.
- Firewall Manager hasn't received a configuration item for it within the number of minutes specified in the policy rule time period.

The policy rule time period has a default setting of zero minutes, but you can increase the time up to 365 days (525,600 minutes), to give yourself time to associate new security groups with resources.

> **Important**
>
> If you specify a number of minutes other than the default value of zero, you must enable indirect relationships in AWS Config. Otherwise, your usage audit security group policies will not work as intended. For information about indirect relationships in AWS Config, see Indirect Relationships in AWS Config in the *AWS Config Developer Guide*.

Firewall Manager remediates unused security groups by deleting them from your account according to your rules settings, if possible. If Firewall Manager is unable to delete a security group, it marks it as noncompliant with the policy. Firewall Manager can't delete a security group that's referenced by another security group.

The timing of the remediation varies according to whether you use the default time period setting or a custom setting:

- **Time period set to zero, the default** – With this setting, a security group is considered unused as soon as it's not being used by an Amazon EC2 instance or elastic network interface.

  For this zero time period setting, Firewall Manager remediates the security group immediately.
- **Time period greater than zero** – With this setting, a security group is considered unused when it's not being used by an Amazon EC2 instance or elastic network interface and Firewall Manager hasn't received a configuration item for it within the specified number of minutes.

  For the non-zero time period setting, Firewall Manager remediates the security group after it's remained in the unused state for 24 hours.

Default account specification

When you create a usage audit security group policy through the console, Firewall Manager automatically chooses **Exclude the specified accounts and include all others**. The service then puts the Firewall Manager administrator account in the list to exclude. This is the recommended approach, and allows you to manually manage the security groups that belong to the Firewall Manager administrator account.

Using Amazon VPC network access control list (ACL) policies with Firewall Manager

This section covers how AWS Firewall Manager network ACL policies work and provides guidance for using them. For guidance creating a network ACL policy using the console, see Creating a network ACL policy.

For information about Amazon VPC network access control lists (ACLs), see Control traffic to subnets using network ACLs in the *Amazon VPC User Guide*.

You can use Firewall Manager network ACL policies to manage Amazon Virtual Private Cloud (Amazon VPC) network access control lists (ACLs) for your organization in AWS Organizations. You define the policy's network ACL rule settings and the accounts and subnets where you want the settings enforced. Firewall Manager continuously applies your policy settings to accounts and subnets as they are added or updated across your organization. For information about policy scope and AWS Organizations, see Using the AWS Firewall Manager policy scope and the AWS Organizations User Guide.

When you define a Firewall Manager network ACL policy, in addition to the standard Firewall Manager policy settings, such as name and scope, you provide the following:

- First and last rules for inbound and outbound traffic handling. Firewall Manager enforces the presence and ordering of these in the network ACLs that are in scope of the policy, or reports noncompliance. Your individual accounts can create custom rules to run in between the policy's first and last rules.
- Whether to force remediation when remediation would result in traffic management conflicts between the rules in the network ACL. This applies only when remediation is enabled for the policy.

Best practices for using Firewall Manager network ACL policies

This section lists recommendations for working with Firewall Manager network ACL policies and managed network ACLs.

Refer to the FMManaged tag to identify network ACLs that are managed by Firewall Manager

The network ACLs that Firewall Manager manages have the `FMManaged` tag set to `true`. Use this tag to help distinguish your own custom network ACLs from those that you're managing through Firewall Manager.

Don't modify the value of the FMManaged tag on a network ACL

Firewall Manager uses this tag to set and determine its management status with a network ACL.

Don't modify the associations for subnets that have Firewall Manager managed network ACLs

Don't manually change the associations between your subnets and any network ACLs that are managed by Firewall Manager. Doing so can disable the ability of Firewall Manager to manage protections for those subnets. You can identify network ACLs that are managed by Firewall Manager by looking for the `FMManaged` tag settings of `true`.

To remove a subnet from Firewall Manager policy management, use the Firewall Manager policy scope settings to exclude the subnet. For example, you can tag the subnet and then exclude that tag from policy scope. For more information, see Using the AWS Firewall Manager policy scope.

When you update a managed network ACL, don't modify the rules that are managed by Firewall Manager

In a network ACL that's managed by Firewall Manager, keep your custom rules separated from the policy rules by adhering to the numbering scheme described in Using network ACL rules and tagging in Firewall Manager. Only add or modify rules that have numbers between 5,000 and 32,000.

Avoid adding too many rules for your account limits

During remediation of a network ACL, Firewall Manager usually increases the network ACL rule count temporarily. To avoid noncompliance problems, make sure you have enough room for the rules you're using. For more information, see How Firewall Manager remediates noncompliant managed network ACLs.

Start with automatic remediation disabled

Start with automatic remediation disabled, and then review the policy details information to determine the effects that automatic remediation would have. When you are satisfied that the changes are what you want, edit the policy to enable automatic remediation.

Firewall Manager network ACL policy caveats

This section lists the caveats and limitations for using Firewall Manager network ACL policies.

- **Slower update times than with other policies** – Firewall Manager generally applies network ACL policies and policy changes more slowly than with other Firewall Manager policies, due to

limitations in the rate at which the Amazon EC2 network ACL APIs are able to process requests. You might notice that policy changes take longer than similar changes with other Firewall Manager policies, in particular when you first add a policy.

- **For initial subnet protection, Firewall Manager prefers older policies** – This applies only to subnets that aren't yet protected by a Firewall Manager network ACL policy. If a subnet comes into scope of more than one network ACL policy at the same time, then Firewall Manager uses the oldest policy to protect the subnet.
- **Reasons for a policy to stop protecting a subnet** – A policy that's managing the network ACL for a subnet retains management until one of the following happens:
  - The subnet goes out of scope of the policy.
  - The policy is deleted.
  - You manually change the subnet's association to a network ACL that's managed by a different Firewall Manager policy and for which the subnet is in scope.

Topics

- Using network ACL rules and tagging in Firewall Manager
- How Firewall Manager initiates network ACL management for a subnet
- How Firewall Manager remediates noncompliant managed network ACLs
- Deleting a Firewall Manager network ACL policy

Using network ACL rules and tagging in Firewall Manager

This section describes the network ACL policy rule specifications and the network ACLs that are managed by Firewall Manager.

Tagging on a managed network ACL

Firewall Manager tags a managed network ACL with a `FMManaged` tag that has a value of `true`. Firewall Manager only performs remediation on network ACLs that have this tag setting.

Rules that you define in the policy

In your network ACL policy specification, you define the rules that you want to run first and last for inbound traffic and the rules that you want to run first and last for outbound traffic.

By default, you can define up to 5 inbound rules, for use in any combination of first and last rules in the policy. Similarly, you can define up to 5 outbound rules. For more about these limits, see

Soft quotas. For information about the general limits on network ACLs, see Amazon VPC quotas on network ACLs in the *Amazon VPC User Guide*.

You don't assign rule numbers to the policy rules. Instead, you specify the rules in the order you want them to be evaluated, and Firewall Manager uses that ordering to assign rule numbers in the network ACLs that it manages.

Other than this, you manage the policy's network ACL rules specifications as you would manage the rules in a network ACL through Amazon VPC. For information about network ACL management in Amazon VPC, see Control traffic to subnets using network ACLs and Work with network ACLs in the Amazon VPC User Guide.

Rules in a managed network ACL

Firewall Manager configures the rules in a network ACL that it manages by placing the policy's first and last rules before and after any custom rules that an individual account manager defines. Firewall Manager preserves the order of the custom rules. Network ACLs are evaluated starting with the lowest numbered rule.

When Firewall Manager first creates a network ACL, it defines the rules with the following numbering:

- **First rules: 1, 2, ...** – Defined by you in the Firewall Manager network ACL policy.

  Firewall Manager assigns rule numbers starting from 1 with increments of 1, with the rules ordered as you have ordered them in the policy specification.
- **Custom rules: 5,000, 5,100, ...** – Managed by individual account managers through Amazon VPC.

  Firewall Manager assigns numbers to these rules starting from 5,000 and incrementing by 100 for each subsequent rule.
- **Last rules: ... 32,765, 32,766** – Defined by you in the Firewall Manager network ACL policy.

  Firewall Manager assigns rule numbers that end at the highest possible number, 32766 with increments of 1, with the rules ordered as you have ordered them in the policy specification.

After network ACL initialization, Firewall Manager doesn't control changes that individual accounts make in its managed network ACLs. Individual accounts can change a network ACL without taking it out of compliance, providing any custom rules remain numbered in between the policy's first and

last rules, and the first and last rules maintain their specified ordering. As a best practice, when managing custom rules, adhere to the numbering described in this section.

How Firewall Manager initiates network ACL management for a subnet

This section describes how Firewall Manager initiates network ACL management for a subnet.

Firewall Manager begins management of the network ACL for a subnet when it associates the subnet with a network ACL that Firewall Manager has created and tagged with `FMManaged` set to `true`.

Compliance with a network ACL policy requires the subnet's network ACL to have the policy's first rules positioned first, in the order specified in the policy, the last rules positioned last, in order, and any other custom rules positioned in the middle. These requirements can be satisfied by an unmanaged network ACL that the subnet is already associated with or by a managed network ACL.

When Firewall Manager applies a network ACL policy to a subnet that's associated with an unmanaged network ACL, Firewall Manager checks the following in order, stopping when it identifies a viable option:

1. **The associated network ACL is already compliant** – If the network ACL that's currently associated with the subnet is compliant, then Firewall Manager leaves that association in place and does not start network ACL management for the subnet.

   Firewall Manager doesn't alter or otherwise manage a network ACL that it doesn't own, but as long as it's compliant, Firewall Manager leaves it in place and just monitors it for policy compliance.
2. **A compliant managed network ACL is available** – If Firewall Manager is already managing a network ACL that complies with the required configuration, then this is an option. If remediation is enabled, Firewall Manager associates the subnet to it. If remediation is disabled, Firewall Manager marks the subnet noncompliant and offers replacing the network ACL association as a remediation option.
3. **Create a new compliant managed network ACL** – If remediation is enabled, Firewall Manager creates a new network ACL and associates it with the subnet. Otherwise, Firewall Manager marks the subnet noncompliant and offers the remediation options of creating the new network ACL and replacing the network ACL association.

If these steps fail, Firewall Manager reports noncompliance for the subnet.

Firewall Manager follows these steps when a subnet first comes into scope and when a subnet's unmanaged network ACL is out of compliance.

How Firewall Manager remediates noncompliant managed network ACLs

This section describes how Firewall Manager remediates its managed network ACLs when they're out of compliance with the policy. Firewall Manager only remediates managed network ACLs—with the `FMManaged` tag set to `true`. For network ACLs that aren't managed by Firewall Manager, see Initial network ACL management.

Remediation restores the relative locations of the first, custom, and last rules and restores the ordering for first and last rules. During remediation, Firewall Manager won't necessarily move rules to the rule numbers that it uses in network ACL initialization. For the initial number settings and descriptions of these rule categories, see Initial network ACL management.

In order to establish compliant rules and rule ordering, Firewall Manager might need to move rules around inside the network ACL. As much as possible, Firewall Manager preserves the network ACL's protections by maintaining existing compliant rule ordering as it does this. For example, it might temporarily duplicate rules to new locations, and then perform an ordered removal of the original rules, preserving relative locations during the process.

This approach protects your settings, but it also requires space in the network ACL for the interim rules. If Firewall Manager hits the limit for rules in a network ACL, it will halt remediation. When this happens, the network ACL remains out of compliance and Firewall Manager reports the reason.

If an account adds custom rules to a network ACL that's managed by Firewall Manager, and those rules interfere with Firewall Manager remediation, Firewall Manager stops any remediation activities on the network ACL and reports the conflict.

Forced remediation

If you choose auto remediation for the policy, you also specify whether to force remediation for the first rules or last rules.

When Firewall Manager encounters a conflict in traffic handling between a custom rule and a policy rule, it refers to the corresponding forced remediation setting. If forced remediation is enabled, Firewall Manager applies the remediation, in spite of the conflict. If this option isn't enabled, Firewall Manager halts remediation. In either case, Firewall Manager reports the rule conflict and offers remediation options.

Rule count requirements and limitations

During remediation, Firewall Manager might temporarily duplicate rules in order to move them without altering the protections that they provide.

For either inbound or outbound rules, the greatest number of rules that Firewall Manager might require to perform remediation is the following:

```
2 * (the number of rules defined in the policy for the traffic direction)
+
the number of custom rules defined in the network ACL for the traffic direction
```

Network ACLs and network ACL policies are bound by mutable rule limits. If Firewall Manager hits a limit in its remediation efforts, it stops trying to remediate and reports the noncompliance.

To make room for Firewall Manager to perform its remediation activities, you might request a limit increase. Alternately, you can change the configuration in the policy or network ACL to reduce the number of rules used.

For information about the network ACL limits, see Amazon VPC quotas on network ACLs in the *Amazon VPC User Guide*.

When remediation fails

While updating a network ACL, if Firewall Manager needs to stop for any reason, it doesn't roll back the changes, but instead leaves the network ACL in an interim state. If you see duplicate rules in a network ACL that has the `FMManaged` tag set to `true`, Firewall Manager is probably in the middle of remediating it. Changes might be partially complete for a period, but because of the approach Firewall Manager takes to remediation, this won't interrupt traffic or reduce the protection for associated subnets.

When Firewall Manager doesn't completely remediate network ACLs that are out of compliance, it reports the noncompliance for the associated subnets and suggests possible remediation options.

Retrying after remediation fails

In most cases, if Firewall Manager fails to complete remediation changes to a network ACL, it will eventually retry the change.

The exception to this is when remediation reaches the network ACL rule count limit or the VPC network ACL count limit. Firewall Manager can't perform remediation activities that take AWS resources over their limit settings. In these cases, you need to reduce counts or increase limits in

order to proceed. For information about the limits, see Amazon VPC quotas on network ACLs in the *Amazon VPC User Guide*.

Firewall Manager network ACL compliance reporting

Firewall Manager monitors and reports compliance for all network ACLs that are attached to in-scope subnets.

Generally speaking, noncompliance occurs for situations such as incorrect rule ordering or a conflict in traffic handling behavior between policy rules and custom rules. Noncompliance reporting includes compliance violations and remediation options.

Firewall Manager reports compliance violations for a network ACL policy in the same way as for other policy types. For information about compliance reporting, see Viewing compliance information for an AWS Firewall Manager policy.

Noncompliance during policy updates

After you modify a network ACL policy, until Firewall Manager updates the network ACLs that are in scope of the policy, Firewall Manager marks those network ACLs noncompliant. Firewall Manager does this even if the network ACLs might, strictly speaking, be in compliance.

For example, if you remove rules from the policy specification, while in-scope network ACLs still have the extra rules, their rule definitions might still comply with the policy. However, since the extra rules are part of the rules that Firewall Manager is managing, Firewall Manager views them as violations of current policy settings. This is different from how Firewall Manager views custom rules that you add to the Firewall Manager managed network ACLs.

Deleting a Firewall Manager network ACL policy

This section describes what happens in Firewall Manager when you delete a Firewall Manager network ACL policy.

When you delete a Firewall Manager network ACL policy, Firewall Manager changes the `FMManaged` tag values to `false` on all network ACLs that it's been managing for the policy.

Additionally, you can choose whether to clean up the resources created by the policy. If you choose clean up, Firewall Manager tries the following steps in order:

1. **Put the association back to the original** – Firewall Manager tries to associate the subnet back to the network ACL that it was associated with before Firewall Manager started managing it.

2. **Remove first and last rules from the network ACL** – If it can't change the association, Firewall Manager tries to remove the policy's first and last rules, leaving only the custom rules in the network ACL that's associated with the subnet.
3. **Do nothing to the rules or the association** – If it can't do either of the above things, Firewall Manager leaves the network ACL and its association as they are.

If you don't choose the cleanup option, you'll need to manually manage each network ACL after the policy is deleted. For most situations, choosing the cleanup option is the simplest approach.

Using AWS Network Firewall policies in Firewall Manager

This section explains how to use AWS Network Firewall policies with Firewall Manager.

You can use AWS Firewall Manager Network Firewall *policies* to manage AWS Network Firewall *firewalls* for your Amazon Virtual Private Cloud *VPCs* across your *organization* in AWS Organizations. You can apply centrally controlled firewalls to your entire organization or to a select subset of your accounts and VPCs.

Network Firewall provides network traffic filtering protections for the public subnets in your VPCs. Firewall Manager creates and manages your firewalls based on the *firewall management type* defined by your policy. Firewall Manager provides the following firewall management models:

- **Distributed** - For each account and VPC that's within policy scope, Firewall Manager creates a Network Firewall firewall and deploys firewall endpoints to VPC subnets, to filter network traffic.
- **Centralized** - Firewall Manager creates a single Network Firewall firewall in a single Amazon VPC.
- **Import existing firewalls** - Firewall Manager imports existing firewalls for management in a single Firewall Manager policy. You can apply additional rules to the imported firewalls managed by your policy to ensure that your firewalls meet your security standards.

> **Note**
>
> Firewall Manager Network Firewall policies are Firewall Manager policies that you use to manage Network Firewall protections for your VPCs across your organization.
> The Network Firewall protections are specified in resources in the Network Firewall service that are called firewall policies.

For information about using Network Firewall, see the AWS Network Firewall Developer Guide.

The following sections cover requirements for using Firewall Manager Network Firewall policies and describe how the policies work. For the procedure for creating the policy, see Creating an AWS Firewall Manager policy for AWS Network Firewall.

> **Important**
>
> **You must enable resource sharing.** A Network Firewall policy shares Network Firewall rule groups across the accounts in your organization. For this to work, you must have resource sharing enabled for AWS Organizations. For information about how to enable resource sharing, see Resource sharing for Network Firewall and DNS Firewall policies.

> **Important**
>
> **You must have your Network Firewall rule groups defined.** When you specify a new Network Firewall policy, you define the firewall policy the same as you do when you're using AWS Network Firewall directly. You specify the stateless rule groups to add, default stateless actions, and stateful rule groups. Your rule groups must already exist in the Firewall Manager administrator account for you to include them in the policy. For information about creating Network Firewall rule groups, see AWS Network Firewall rule groups.

Topics

- How Firewall Manager creates firewall endpoints
- How Firewall Manager manages your firewall subnets
- How Firewall Manager manages your Network Firewall resources
- How Firewall Manager manages and monitors VPC route tables for your policy
- Configuring logging for an AWS Network Firewall policy

How Firewall Manager creates firewall endpoints

This section explains how Firewall Manager creates firewall endpoints.

The *Firewall management type* in your policy determines how Firewall Manager creates firewalls. Your policy can create *distributed* firewalls, a *centralized* firewall, or you can **import existing firewalls**:

- **Distributed** - With the distributed deployment model, Firewall Manager creates endpoints for each VPC that's within policy scope. You can either customize the endpoint location by specifying which Availability Zones to create firewall endpoints in, or Firewall Manager can automatically create endpoints in the Availability Zones with public subnets. If you manually choose the Availability Zones, you have the option to restrict the set of allowed CIDRs per Availability Zone. If you decide to let Firewall Manager automatically create the endpoints, you must also specify whether the service will create a single endpoint or multiple firewall endpoints within your VPCs.
  - For multiple firewall endpoints, Firewall Manager deploys a firewall endpoint in each Availability Zone where you have a subnet with an internet gateway or a Firewall Manager-created firewall endpoint route in the route table. This is the default option for a Network Firewall policy.
  - For a single firewall endpoint, Firewall Manager deploys a firewall endpoint in a single Availability Zone in any subnet that has an internet gateway route. With this option, traffic in other zones needs to cross zone boundaries in order to be filtered by the firewall.

    > **Note**
    >
    > For both of these options, there must be a subnet associated to a route table that has an IPv4/prefixlist route in it. Firewall Manager does not check for any other resources.

- **Centralized** - With the centralized deployment model, Firewall Manager creates one or more firewall endpoints within an *inspection VPC*. An inspection VPC is a central VPC where Firewall Manager launches your endpoints. When you use the centralized deployment model, you also specify which Availability Zones to create firewall endpoints in. You can't change the inspection VPC after you create your policy. To use a different inspection VPC, you must create a new policy.
- **Import existing firewalls** - When you import existing firewalls, you choose the firewalls to manage in your policy by adding one or more *resource sets* to your policy. A resource set is a collection of resources, in this case existing firewalls in Network Firewall, that are managed by an account in your organization. Before you use resource sets in your policy, you must first create a resource set. For information about Firewall Manager resource sets, see Grouping your resources in Firewall Manager.

  Keep in mind the following considerations when working with imported firewalls:

- If an imported firewall become non-compliant, Firewall Manager will try to automatically resolve the violation, except for under the following circumstances:
  - If there's a mismatch between the Firewall Manager and Network Firewall policy's stateful or stateless default actions.
  - If a rule group in an imported firewall's firewall policy has the same priority as a rule group in the Firewall Manager policy.
  - If an imported firewall uses a firewall policy that's associated with a firewall that's not part of the policy's resource set. This can happen because a firewall can have exactly one firewall policy, but a single firewall policy can be associated with multiple firewalls.
  - If a pre-existing rule group belonging to an imported firewall's firewall policy that is also specified in the Firewall Manager policy is given a different priority.
- If you enable resource cleanup in the policy, Firewall Manager removes the rule groups which have been in FMS import policy from the firewalls in scope of the resource set.
- Firewalls managed by that are managed by a Firewall Manager import existing firewall management type can only be managed by one policy at a time. If the same resource set is added to multiple import network firewall policies, the firewalls in the resource set will be managed by the first policy the resource set was added to and will be ignored by the second policy.
- Firewall Manager doesn't currently stream exception policy configurations. For information about stream exception policies, see Stream exception policy in the *AWS Network Firewall Developer Guide*.
- Import existing firewalls does not support importing Transit gateway-attached firewalls.

If you change the list of Availability Zones for policies using distributed or centralized firewall management, Firewall Manager will try to clean up any endpoints that were created in the past, but that aren't currently in policy scope. Firewall Manager will remove the endpoint only if there are no route table routes that reference the out of scope endpoint. If Firewall Manager finds that it is unable to delete these endpoints, it will mark the firewall subnet as being non-compliant and will continue attempting to remove the endpoint until such time as it is safe to delete.

How Firewall Manager manages your firewall subnets

This section explains how Firewall Manager manages your firewall subnets.

Firewall subnets are the VPC subnets that Firewall Manager creates for the firewall endpoints that filter your network traffic. Each firewall endpoint must be deployed in a dedicated VPC subnet. Firewall Manager creates at least one firewall subnet in each VPC that's within scope of the policy.

For policies that use the distributed deployment model with automatic endpoint configuration, Firewall Manager only creates firewall subnets in Availability Zones that have a subnet with an internet gateway route, or a subnet with a route to the firewall endpoints that Firewall Manager created for their policy. For more information, see VPCs and subnets in the *Amazon VPC User Guide*.

For policies that use either the distributed or centralized model where you specify which Availability Zones Firewall Manager creates the firewall endpoints in, Firewall Manager creates an endpoint in those specific Availability Zones irrespective of whether there are other resources in the Availability Zone.

When you first define a Network Firewall policy, you specify how Firewall Manager manages the firewall subnets in each of the VPCs that are in scope. You cannot change this choice later.

For policies that use the distributed deployment model with automatic endpoint configuration, you can choose between the following options:

- Deploy a firewall subnet for every Availability Zone that has public subnets. This is the default behavior. This provides high availability of your traffic filtering protections.
- Deploy a single firewall subnet in one Availability Zone. With this choice, Firewall Manager identifies a zone in the VPC that has the most public subnets and creates the firewall subnet there. The single firewall endpoint filters all network traffic for the VPC. This can reduce firewall costs, but it isn't highly available and it requires traffic from other zones to cross zone boundaries in order to be filtered.

For policies that use distributed deployment model with custom endpoint configuration or the centralized deployment model, Firewall Manager creates the subnets in the specified Availability Zones that are within the policy scope.

You can provide VPC CIDR blocks for Firewall Manager to use for the firewall subnets or you can leave the choice of firewall endpoint addresses up to Firewall Manager to determine.

- If you don't provide CIDR blocks, Firewall Manager queries your VPCs for available IP addresses to use.
- If you provide a list of CIDR blocks, Firewall Manager searches for new subnets only in the CIDR blocks that you provide. You must use /28 CIDR blocks. For each firewall subnet that Firewall

Manager creates, it walks your CIDR block list and uses the first one that it finds that is applicable to the Availability Zone and VPC and has available addresses. If Firewall Manager is unable to find open space in the VPC (with or without the restriction), the service won't create a firewall in the VPC.

If Firewall Manager can't create a required firewall subnet in an Availability Zone, it marks the subnet as non-compliant with the policy. While the zone is in this state, traffic for the zone must cross zone boundaries in order to be filtered by an endpoint in another zone. This is similar to the single firewall subnet scenario.

How Firewall Manager manages your Network Firewall resources

This section describes how you manage your Network Firewall resources in Firewall Manager.

When you define the policy in Firewall Manager, you provide the network traffic filtering behavior of a standard AWS Network Firewall firewall policy. You add stateless and stateful Network Firewall rule groups and specify default actions for packets that don't match any stateless rules. For information on working with firewall policies in AWS Network Firewall, see the AWS Network Firewall firewall policies.

For distributed and centralized policies, when you save the Network Firewall policy, Firewall Manager creates a firewall and firewall policy in each VPC that's within scope of the policy. Firewall Manager names these Network Firewall resources by concatenating the following values:

- A fixed string, either `FMManagedNetworkFirewall` or `FMManagedNetworkFirewallPolicy`, depending on the resource type.
- Firewall Manager policy name. This is the name you assign when you create the policy.
- Firewall Manager policy ID. This is the AWS resource ID for the Firewall Manager policy.
- Amazon VPC ID. This is the AWS resource ID for the VPC where Firewall Manager creates the firewall and firewall policy.

The following shows an example name for a firewall that's managed by Firewall Manager:

```
FMManagedNetworkFirewallEXAMPLENameEXAMPLEFirewallManagerPolicyIdEXAMPLEVPCId
```

The following shows an example firewall policy name:

```
FMManagedNetworkFirewallPolicyEXAMPLENameEXAMPLEFirewallManagerPolicyIdEXAMPLEVPCId
```

After you create the policy, member accounts in the VPCs can't override your firewall policy settings or your rule groups, but they can add rule groups to the firewall policy that Firewall Manager has created.

How Firewall Manager manages and monitors VPC route tables for your policy

This section explains how Firewall Manager manages and monitors your VPC route tables.

> **Note**
>
> Route table management isn't currently supported for policies that use the centralized deployment model.

When Firewall Manager creates your firewall endpoints, it also creates the VPC route tables for them. However, Firewall Manager doesn't manage your VPC route tables. You must configure your VPC route tables to direct network traffic to the firewall endpoints that are created by Firewall Manager. Using Amazon VPC ingress routing enhancements, change your routing tables to route traffic through the new firewall endpoints. Your changes must insert the firewall endpoints between the subnets that you want to protect and outside locations. The exact routing that you need to do depends on your architecture and its components.

Currently, Firewall Manager allows monitoring of your VPC route table routes for any traffic destined to the internet gateway, that is bypassing the firewall. Firewall Manager doesn't support other target gateways like NAT gateways.

For information about managing route tables for your VPC, see Managing route tables for your VPC in the *Amazon Virtual Private Cloud User Guide*. For information about managing your route tables for Network Firewall, see Route table configurations for AWS Network Firewall in the *AWS Network Firewall Developer Guide*.

When you enable monitoring for a policy, Firewall Manager continuously monitors VPC route configurations and alerts you about traffic that bypasses firewall inspection for that VPC. If a subnet has a firewall endpoint route, Firewall Manager looks for the following routes:

- Routes to send traffic to the Network Firewall endpoint.
- Routes to forward the traffic from the Network Firewall endpoint to the internet gateway.

- Inbound routes from the internet gateway to the Network Firewall endpoint.
- Routes from the firewall subnet.

If a subnet has a Network Firewall route but there's asymmetric routing in Network Firewall and your internet gateway route table, Firewall Manager reports the subnet as non-compliant. Firewall Manager also detects routes to the internet gateway in the firewall route table that Firewall Manager created, as well as the route table for your subnet, and reports them as non-compliant. Additional routes in the Network Firewall subnet route table and your internet gateway route table are also reported as non-compliant. Depending on the violation type, Firewall Manager suggests remediation actions to bring the route configuration into compliance. Firewall Manager doesn't offer suggestions in all cases. For example, if your customer subnet has a firewall endpoint that was created outside of Firewall Manager, Firewall Manager doesn't suggest remediation actions.

By default, Firewall Manager will mark any traffic that crosses the Availability Zone boundary for inspection as being non-compliant. However, if you choose to automatically create a single endpoint in your VPC, Firewall Manager won't mark traffic that crosses the Availability Zone boundary as non-compliant.

For policies that use distributed deployment models with custom endpoint configuration, you can choose whether the traffic crossing the Availability Zone boundary from an Availability Zone without a firewall endpoint is marked as compliant or non-compliant.

Note

- Firewall Manager does not suggest remediation actions for non-IPv4 routes, such as IPv6 and prefix list routes.
- Calls made using the `DisassociateRouteTable` API call can take up to 12 hours to detect.
- Firewall Manager creates a Network Firewall route table for a subnet that contains the firewall endpoints. Firewall Manager assumes that this route table contains only valid internet gateway and VPC default routes. Any extra or invalid routes in this route table are considered to be non-compliant.

When you configure your Firewall Manager policy, if you choose **Monitor** mode, Firewall Manager provides resource violation and remediation details about your resources. You can use these

suggested remediation actions to fix route issues in your route tables. If you choose **Off** mode, Firewall Manager doesn't monitor your route table content for you. With this option, you manage your VPC route tables for yourself. For more information about these resource violations, see Viewing compliance information for an AWS Firewall Manager policy.

> **Warning**
>
> If you choose **Monitor** under **AWS Network Firewall route configuration** when creating your policy, you can't turn it off for that policy. However, if you choose **Off**, you can enable it later.

Configuring logging for an AWS Network Firewall policy

This section explains how you can enable centralized logging for your Network Firewall policies to get detailed information about traffic within your organization. You can select flow logging to capture network traffic flow, or alert logging to report traffic that matches a rule with the rule action set to `DROP` or `ALERT`. For more information about AWS Network Firewall logging, see Logging network traffic from AWS Network Firewall in the *AWS Network Firewall Developer Guide*.

You send logs from your policy's Network Firewall firewalls to an Amazon S3 bucket. After you enable logging, AWS Network Firewall delivers logs for each configured Network Firewall by updating the firewall settings to deliver logs to your selected Amazon S3 buckets with the reserved AWS Firewall Manager prefix, `<policy-name>-<policy-id>`.

> **Note**
>
> This prefix is used by Firewall Manager to determine whether a logging configuration was added by Firewall Manager, or whether it was added by the account owner. If the account owner attempts to use the reserved prefix for their own custom logging, it is overwritten by the logging configuration in the Firewall Manager policy.

For more information about how to create an Amazon S3 bucket and review the stored logs, see What is Amazon S3? in the *Amazon Simple Storage Service User Guide*.

To enable logging you must meet the following requirements:

- The Amazon S3 that you specify in your Firewall Manager policy must exist.

- You must have the following permissions:
  - `logs:CreateLogDelivery`
  - `s3:GetBucketPolicy`
  - `s3:PutBucketPolicy`
- If the Amazon S3 bucket that's your logging destination uses server-side encryption with keys that are stored in AWS Key Management Service, you must add the following policy to your AWS KMS customer-managed key to allow Firewall Manager to log to your CloudWatch Logs log group:

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "delivery.logs.amazonaws.com"
    },
    "Action": [
        "kms:Encrypt*",
        "kms:Decrypt*",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:Describe*"
    ],
    "Resource": "*"
}
```

Note that only buckets in the Firewall Manager administrator account may be used for AWS Network Firewall central logging.

When you enable centralized logging on a Network Firewall policy, Firewall Manager takes these actions on your account:

- Firewall Manager updates the permissions on selected S3 buckets to allow for log delivery.
- Firewall Manager creates directories in the S3 bucket for each member account in the scope of the policy. The logs for each account can be found at `<bucket-name>/<policy-name>-<policy-id>/AWSLogs/<account-id>`.

To enable logging for a Network Firewall policy

1. Create an Amazon S3 bucket using your Firewall Manager administrator account. For more information, see Creating a bucket in the *Amazon Simple Storage Service User Guide*.
2. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

3. In the navigation pane, choose **Security Policies**.
4. Choose the Network Firewall policy that you want to enable logging for. For more information about AWS Network Firewall logging, see Logging network traffic from AWS Network Firewall in the *AWS Network Firewall Developer Guide*.
5. On the **Policy details** tab, in the **Policy rules** section, choose **Edit**.
6. To enable and aggregate logs, choose one or more options under **Logging configuration**:
   - **Enable and aggregate flow logs**
   - **Enable and aggregate alert logs**
7. Choose the Amazon S3 bucket where you want your logs to be delivered. You must choose a bucket for each log type that you enable. You can use the same bucket for both log types.
8. (Optional) If you want custom member account-created logging to be replaced with the policy's logging configuration, choose **Override existing logging configuration**.
9. Choose **Next**.
10. Review your settings, then choose **Save** to save your changes to the policy.

To disable logging for a Network Firewall policy

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

> **Note**
>
> For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security Policies**.
3. Choose the Network Firewall policy that you want to disable logging for.
4. On the **Policy details** tab, in the **Policy rules** section, choose **Edit**.
5. Under **Logging configuration status**, deselect **Enable and aggregate flow logs** and **Enable and aggregate alert logs** if they are selected.
6. Choose **Next**.
7. Review your settings, then choose **Save** to save your changes to the policy.

Using Amazon Route 53 Resolver DNS Firewall policies in Firewall Manager

This page describes how you can use AWS Firewall Manager DNS Firewall policies to manage associations between Amazon Route 53 Resolver DNS Firewall rule groups and your Amazon Virtual Private Cloud *VPCs* across your *organization* in AWS Organizations. You can apply centrally controlled rule groups to your entire organization, or to a select subset of your accounts and VPCs.

DNS Firewall provides filtering and regulation of outbound DNS traffic for your VPCs. You create reusable collections of filtering rules in DNS Firewall rule groups and you associate the rule groups to your VPCs. When you apply the Firewall Manager policy, for each account and VPC that's within policy scope, Firewall Manager creates an association between each DNS Firewall rule group in the policy and each VPC that's within scope of the policy, using the association priority settings that you specify in the Firewall Manager policy.

For information about using DNS Firewall, see Amazon Route 53 Resolver DNS Firewall in the Amazon Route 53 Developer Guide.

The following sections cover requirements for using Firewall Manager DNS Firewall policies and describe how the policies work. For the procedure for creating the policy, see Creating an AWS Firewall Manager policy for Amazon Route 53 Resolver DNS Firewall.

Important

You must enable resource sharing. A DNS Firewall policy shares DNS Firewall rule groups across the accounts in your organization. For this to work, you must have resource sharing enabled with AWS Organizations. For information about how to enable resource sharing, see Resource sharing for Network Firewall and DNS Firewall policies.

Important

You must have your DNS Firewall rule groups defined. When you specify a new DNS Firewall policy, you define the rule groups the same as you do when you're using Amazon Route 53 Resolver DNS Firewall directly. Your rule groups must already exist in the Firewall Manager administrator account for you to include them in the policy. For information about creating DNS Firewall rule groups, see DNS Firewall rule groups and rules.

You define the lowest and highest priority rule group associations

The DNS Firewall rule group associations that you manage through Firewall Manager DNS Firewall policies contain the lowest priority associations and the highest priority associations for your VPCs. In your policy configuration, these appear as first and last rule groups.

DNS Firewall filters DNS traffic for the VPC in the following order:

1. First rule groups, defined by you in the Firewall Manager DNS Firewall policy. Valid values are between 1 and 99.
2. DNS Firewall rule groups that are associated by individual account managers through DNS Firewall.
3. Last rule groups, defined by you in the Firewall Manager DNS Firewall policy. Valid values are between 9,901 and 10,000.

How Firewall Manager names the rule group associations that it creates

When you save the DNS Firewall policy, if you enabled autoremediation, Firewall Manager creates a DNS Firewall association between the rule groups that you provided in the policy and the VPCs that are in scope of the policy. Firewall Manager names these associations by concatenating the following values:

- The fixed string, `FMManaged_`.
- The Firewall Manager policy ID. This is the AWS resource ID for the Firewall Manager policy.

The following shows an example name for a firewall that's managed by Firewall Manager:

```
FMManaged_EXAMPLEDNSFirewallPolicyId
```

After you create the policy, if account owners in the VPCs override your firewall policy settings or your rule group associations then Firewall Manager will mark the policy as non-compliant and try to propose a remedial action. Account owners can associate other DNS Firewall rule groups to the VPCs that are in scope of the DNS Firewall policy. Any associations that are created by the individual account owners must have priority settings between your first and last rule group associations.

Deleting a rule group from a Firewall Manager DNS Firewall policy

Deleting a rule group

To delete a rule group from a Firewall Manager DNS Firewall policy, you must perform the following steps:

> **Important**
>
> Removing a rule group from your Firewall Manager DNS Firewall policy removes its effect from VPCs that have the policy applied, regardless of whether you also delete the rule group from your DNS Firewall rule groups. Deleting a rule group is a permanent action and can't be undone.

1. Remove the rule group from your Firewall Manager DNS Firewall policy.
2. Unshare the rule group in AWS Resource Access Manager. To unshare a rule group that you own, you must remove it from the resource share. You can do this using the AWS RAM console or the AWS CLI. For information about unsharing a resource, see Update a resource share in AWS RAM in the *AWS RAM User Guide*.
3. Delete the rule group using the DNS Firewall console or AWS CLI.

Using Palo Alto Networks Cloud NGFW policies for Firewall Manager

The Palo Alto Networks Cloud Next Generation Firewall (NGFW) is a third-party firewall service that you can use for your AWS Firewall Manager policies. With Palo Alto Networks Cloud NGFW for Firewall Manager, you can create and centrally deploy Palo Alto Networks Cloud NGFW resources and rulestacks across all of your AWS accounts.

To use Palo Alto Networks Cloud NGFW with Firewall Manager, you first subscribe to the Palo Alto Networks Cloud NGFW Pay-As-You-Go service in the AWS Marketplace. After subscribing, you perform a series of steps in the Palo Alto Networks Cloud NGFW service to configure your account and Cloud NGFW settings. Then, you create a Firewall Manager Cloud FMS policy to centrally deploy and manage Palo Alto Networks Cloud NGFW resources and rules across all of the accounts in your AWS Organizations.

For the procedure for creating the Firewall Manager policy, see Creating an AWS Firewall Manager policy for Palo Alto Networks Cloud NGFW. For information about how to configure and manage Palo Alto Networks Cloud NGFW for Firewall Manager, see the *Palo Alto Networks Palo Alto Networks Cloud NGFW on AWS* documentation. For supported AWS Regions, see *Cloud NGFW for AWS Supported Regions and Zones*.

Using Fortigate Cloud Native Firewall (CNF) as a Service policies for Firewall Manager

Fortigate Cloud Native Firewall (CNF) as a Service is a third-party firewall service that you can use for your AWS Firewall Manager policies. Fortigate CNF is a next generation firewall service that makes it easy for you to protect your cloud networks and manage your security policies. With Fortigate CNF for Firewall Manager, you can create and centrally deploy Fortigate CNF resources and policy sets across all of your AWS accounts.

To use Fortigate CNF with Firewall Manager, you first subscribe to the Fortigate Cloud Native Firewall (CNF) as a Service in the AWS Marketplace. After subscribing, you perform a series of steps in the Fortigate CNF service to configure your global policy sets and other settings. Then, you create a Firewall Manager policy to centrally deploy and manage Fortigate CNF resources across all of the accounts in your AWS Organizations.

For the procedure for creating a Fortigate CNF Firewall Manager policy, see Creating an AWS Firewall Manager policy for Fortigate Cloud Native Firewall (CNF) as a Service. For information about how to configure and manage Fortigate CNF for use with Firewall Manager, see the Fortigate CNF documentation.

Resource sharing for Network Firewall and DNS Firewall policies

To manage Firewall Manager Network Firewall and DNS Firewall policies, you must enable resource sharing with AWS Organizations in AWS Resource Access Manager. This allows Firewall Manager to deploy protections across your accounts when you create these policy types.

To enable resource sharing, follow the instructions at Enable Sharing with AWS Organizations in the *AWS Resource Access Manager User Guide*.

Problems with resource sharing

You might encounter problems with resource sharing, either when you use AWS RAM to enable it, or when you're working on Firewall Manager policies that require it.

Examples of these problems include the following:

- When you follow the instructions to enable sharing, in the AWS RAM console, the choice **Enable sharing with AWS Organizations** is grayed out and not available for selection.
- When you work in Firewall Manager on a policy that requires resource sharing, the policy is marked as non-compliant and you see messages indicating that resource sharing or AWS RAM isn't enabled.

If you encounter problems with resource sharing, use the following procedure to try to enable it.

Try again to enable resource sharing

- Try again to enable sharing using one of the following options:
  - (Option) Through the AWS RAM console, follow the instructions at Enable Sharing with AWS Organizations in the *AWS Resource Access Manager User Guide*.
  - (Option) Using the AWS RAM API, call `EnableSharingWithAwsOrganization`. See the documentation at EnableSharingWithAwsOrganization.

Using Firewall Manager managed lists

This section explains what managed lists are and how to use them.

Managed application and protocol lists streamline your configuration and management of AWS Firewall Manager content audit security group policies. You use managed lists to define the

protocols and applications that your policy allows and disallows. For information about content audit security group policies, see Using content audit security group policies with Firewall Manager.

You can use the following types of managed lists in a content audit security group policy:

- **Firewall Manager application lists and protocol lists** – Firewall Manager manages these lists.
  - The application lists include `FMS-Default-Public-Access-Apps-Allowed` and `FMS-Default-Public-Access-Apps-Denied`, which describe commonly used applications that should be allowed or denied to the general public.
  - The protocol lists include `FMS-Default-Protocols-Allowed`, a list of commonly used protocols that should be allowed to the general public. You can use any list that Firewall Manager manages, but you can't edit or delete it.
- **Custom application lists and protocol lists** – You manage these lists. You can create lists of either type with the settings that you need. You have full control over your own custom managed lists, and you can create, edit, and delete them as needed.

> **Note**
>
> Currently, Firewall Manager doesn't check references to a custom managed list when you delete it. This means that you can delete a custom managed application list or protocol list even when it is in use by an active policy. This can cause the policy to stop functioning. Delete an application list or protocol list only after you have verified that it isn't referenced by any active polices.

Managed lists are AWS resources. You can tag a custom managed list. You can't tag a Firewall Manager managed list.

Managed list versioning

Custom managed lists don't have versions. When you edit a custom list, policies that reference the list automatically use the updated list.

Firewall Manager managed lists are versioned. The Firewall Manager service team publishes new versions as needed, in order to apply the best security practices to the lists.

When you use a Firewall Manager managed list in a policy, you choose your versioning strategy as follows:

- **Latest available version** – If you don't specify an explicit version setting for the list, then your policy automatically uses the latest version. This is the only option available through the console.
- **Explicit version** – If you specify a version for the list, then your policy uses that version. Your policy remains locked to the version that you specified until you modify the version setting. To specify the version, you must define the policy outside of the console, for example through the CLI or one of the SDKs.

For more information about choosing the version setting for a list, see Using managed lists in your content audit security group policies.

Using managed lists in your content audit security group policies

When you create a content audit security group policy, you can choose to use managed audit policy rules. Some of the settings for this option require a managed application list or protocol list. Examples of these settings include protocols that are allowed in security group rules and applications can access the internet.

The following restrictions apply for each policy setting that uses a managed list:

- You can specify at most one Firewall Manager managed list for any setting. By default, you can specify at most one custom list. The custom list limit is a soft quota, so you can request an increase to it. For more information, see AWS Firewall Manager quotas.
- In the console, if you select a Firewall Manager managed list, you can't specify the version. The policy will always use the latest version of the list. To specify the version, you must define the policy outside of the console, for example through the CLI or one of the SDKs. For information about versioning for Firewall Manager managed lists, see Managed list versioning.

For information about creating a content audit security group policy through the console, see Creating a content audit security group policy.

Creating a custom managed list in Firewall Manager

Follow these procedures to create a custom managed application list or custom managed protocol list.

Topics

- Creating a custom managed application list

- Creating a custom managed protocol list

Creating a custom managed application list

To create a custom managed application list

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Application lists**.
3. In the **Application lists** page, choose **Create application list**.
4. In the **Create application list** page, give your list a name. Don't use the prefix `fms-` as this is reserved for Firewall Manager.
5. Specify an application either by providing the protocol and port number or by selecting an application from the **Type** drop down. Give your application specification a name.
6. Choose **Add another** as needed and fill in the application information until you have completed your list.
7. (Optional) Apply tags to your list.
8. Choose **Save** to save your list and return to the **Application lists** page.

Creating a custom managed protocol list

To create a custom managed protocol list

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

> **Note**
>
> For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Protocol lists**.
3. In the **Protocol lists** page, choose **Create protocol list**.
4. In the protocol list creation page, give your list a name. Don't use the prefix `fms-` as this is reserved for Firewall Manager.
5. Specify a protocol.
6. Choose **Add another** as needed and fill in the protocol information until you have completed your list.
7. (Optional) Apply tags to your list.
8. Choose **Save** to save your list and return to the **Protocol lists** page.

Viewing a managed list in Firewall Manager

To view an application list or protocol list

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Application lists** or **Protocol lists**.

   The page displays all of the lists of the selected type that are available for your use. The lists that Firewall Manager manages have a **Y** in the **ManagedList** column.
3. To see the details of a list, choose its name. The detail page displays the list's content and any tags.

For Firewall Manager managed lists, you can also see the available versions by selecting the **Version** drop down.

Deleting a custom managed list in Firewall Manager

You can delete custom managed lists. You can't edit or delete lists that Firewall Manager manages.

> **Note**
>
> Currently, Firewall Manager doesn't check references to a custom managed list when you delete it. This means that you can delete a custom managed application list or protocol list even when it is in use by an active policy. This can cause the policy to stop functioning. Only delete an application list or protocol list after you have verified that it isn't referenced by any active polices.

To delete a custom managed application or protocol list

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. Make sure that the list that you want to delete isn't in use in any of your audit security group policies by doing the following:

   a. In the navigation pane, choose **Security policies**.

   b. In the **AWS Firewall Manager policies** page, select and edit your audit security groups, and remove any references to the custom list that you want to delete.

      If you delete a custom managed list that's in use in an audit security group policy, the policy that's using it can stop functioning.

3. In the navigation pane, choose **Application lists** or **Protocol lists**, depending on the type of list you want to delete.
4. In the list page, select the custom list that you want to delete and choose **Delete**.

Grouping your resources in Firewall Manager

This section decribes what a resource set is and lists considerations for using resource sets.

An AWS Firewall Manager *resource set* is a collection of resources, such as firewalls, that you can group together and manage in a Firewall Manager policy. Resource sets enable members in your organization to have granular control over what resources to manage in a policy. To use resource sets, create a resource set in the console or using the PutResourceSet API, then add the resource set to your Firewall Manager policy.

You can create and manage resource sets for the following resource and security policy types:

Resource type	Firewall Manager security policy type
AWS Network Firewall - firewalls	Network Firewall policy - Use resource sets to import existing firewalls from Network Firewall. For information about using resource sets in a Network Firewall policy, see the Importing existing firewalls step in the Creating an AWS Firewall Manager policy for AWS Network Firewall procedure.

The following sections cover requirements for creating and deleting resource sets.

Topics

- Considerations when working with resource sets in Firewall Manager
- Creating resource sets in Firewall Manager
- Deleting a resource set in Firewall Manager

Considerations when working with resource sets in Firewall Manager

Note the following considerations when working with resource sets.

References to non-existent resources

When you add a resource to a resource set, you create a reference to the resource using an Amazon Resource Name (ARN). Firewall Manager validates that Amazon Resource Name (ARN) is the correct

format, but Firewall Manager doesn't check that the referenced resource exists. If the resource doesn't exist yet passes ARN validation, Firewall Manager includes the resource reference in the resource set. If a new resource with the same ARN is later created, Firewall Manager applies rule groups from the resource set's associated policy to the new resource.

Deleted resources

When a resource in a resource set is deleted, the reference to the resource remains in the resource set until it's removed by the Firewall Manager administrator.

Resources owned by member account that leaves the AWS Organizations organization

If a member account leaves the organization, any references to resources owned by that member account will remain in the resource set but will no longer be managed by any policies the resource set is associated with.

Association to multiple policies

A resource set can be associated with multiple policies, but not all policy types support multiple policies managing the same resource. See the documentation for your specific policy type for information about unsupported scenarios.

Creating resource sets in Firewall Manager

To create a resource set (console)

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

   > **Note**
   >
   > For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Resource sets**.
3. Choose **Create resource set**.
4. For **Resource set name**, enter a descriptive name.

5. (Optional) enter a **Description** for the resource set.
6. Choose **Next**.
7. For **Choose resources**, select an **AWS account ID** then select **Choose resources** to add resources owned and managed by this account to the resource set. After you select the resources, select **Add** to add the resources to the resource set.
8. Choose **Next**.
9. For **Resource set tags**, add any identifying tags that you want for the resource set. For more information about tags, see Working with Tag Editor.
10. Choose **Next**.
11. Review the new resource set. To make any changes, choose **Edit** in the area that you want to change. This returns you to the corresponding step in the creation wizard. When you are satisfied with the resource set, choose **Create resource set**.

Deleting a resource set in Firewall Manager

Before you can delete a resource set, the resource set must be disassociated from all policies using the resource set. You can disassociate resource groups in the policy detail page using the console, or with the PutPolicy API.

To delete a resource set (console)

1. In the navigation pane, choose **Resource sets**.
2. Choose the option next to the resource set that you want to delete.
3. Choose **Delete**.

Viewing compliance information for an AWS Firewall Manager policy

This section provides guidance for viewing the compliance status of accounts and resources that are in scope of an AWS Firewall Manager policy. For information about the controls in place at AWS to maintain security and compliance of the cloud, see Compliance validation for Firewall Manager.

> **Note**
>
> In order for Firewall Manager to monitor policy compliance, AWS Config must continuously record configuration changes for protected resources. In your AWS Config configuration, the recording frequency must be set to **Continuous**, which is the default setting.

> **Note**
>
> To maintain proper compliance state in your protected resources, avoid repeatedly changing the state of the Firewall Manager protections, either automatically or manually. Firewall Manager uses information from AWS Config to detect changes to resource configurations. If changes are applied quickly enough, AWS Config can lose track of some of them, which can result in the loss of information about compliance or remediation state in Firewall Manager.
>
> If you see that a resource you're protecting with Firewall Manager has an incorrect compliance or remediation status, first make sure you're not running any process that alters or resets your Firewall Manager protections, then refresh the AWS Config tracking for the resource by reevaluating the associated configuration rules in AWS Config.
>
> If you modify the policy or in-scope resources, it may take several minutes before updates to the compliance status and details are visible.

For all AWS Firewall Manager policies, you can view the compliance status for accounts and resources that are in scope of the policy. An account or resource is in compliance with a Firewall Manager policy if the settings in the policy are reflected in the settings for the account or resource. Each policy type has its own compliance requirements, which you can tune when you define the policy. For some policies, you can also view detailed violation information for in scope resources, to help you to better understand and manage your security risk.

To view the compliance information for a policy

1. Sign in to the AWS Management Console using your Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fmsv2. For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

> **Note**
>
> For information about setting up a Firewall Manager administrator account, see AWS Firewall Manager prerequisites.

2. In the navigation pane, choose **Security policies**.
3. Choose a policy. In the **Accounts and resources** tab of the policy page, Firewall Manager lists the accounts in your organization, grouped by those that are within scope of the policy and those that are outside of scope.

   The **Accounts within policy scope** pane lists the compliancy status for each account. A **Compliant** status indicates that the policy has successfully been applied to all of in-scope resources for the account. A **Noncompliant** status indicates that the policy hasn't been applied to one or more of the in-scope resources for the account.
4. Choose an account that's noncompliant. In the account page, Firewall Manager lists the ID and type for each noncompliant resource and the reason that the resource is in violation of the policy.

   > **Note**
   >
   > For the resource types `AWS::EC2::NetworkInterface` (ENI) and `AWS::EC2::Instance`, Firewall Manager might show a limited number of noncompliant resources. To list additional noncompliant resources, fix the ones that are initially displayed for the account.
5. If the Firewall Manager policy type is a content audit security group policy, you can access detailed violation information for a resource.

   To view violation details, choose the resource.

   > **Note**
   >
   > Resources that Firewall Manager found to be noncompliant before the addition of the detailed resource violation page might not have violation details.

In the resource page, Firewall Manager lists specific details about the violation, according to resource type.

- **`AWS::EC2::NetworkInterface` (ENI)** – Firewall Manager displays information about the security group that the resource doesn't comply with. Choose the security group to see more detail about it.
- **`AWS::EC2::Instance`** – Firewall Manager displays the ENI attached to the EC2 instance that's noncompliant. It also displays information about the security group that the resources don't comply with. Choose the security group to see more detail about it.
- **`AWS::EC2::SecurityGroup`** – Firewall Manager displays the following violation details:
  - **Noncompliant security group rule** – The rule that's in violation, including its protocol, port range, IP CIDR range, and description.
  - **Referenced rule** – The audit security group rule that the noncompliant security group rule violates, with its details.
  - **Violation reasons** – Explanation of the noncompliance finding.
  - **Remediation action** – Suggested action to take. If Firewall Manager can't determine a safe remediation action, this field is blank.
- **`AWS::EC2::Subnet`** – This is used for network ACL and Network Firewall policies.

  Firewall Manager displays the subnet ID, VPC ID, and Availability Zone. If applicable, Firewall Manager includes additional information about the violation. The violation description component contains a description of the expected state of the resource, the current, noncompliant state, and if available, a description of what caused the discrepancy.

  Network Firewall violations

  - **Route management violations** – For Network Firewall policies that use Monitor mode, Firewall Manager displays basic subnet information, as well as expected and actual routes in the subnet, internet gateway, and Network Firewall subnet route table. Firewall Manager alerts you that there's a violation if the actual routes don’t match the expected routes in the route table.
  - **Remediation actions for route management violations** – For Network Firewall policies that use Monitor mode, Firewall Manager suggests possible remediation actions on route configurations that have violations.

For example, say a subnet is expected to send traffic through the firewall endpoints, but the current subnet is sending traffic directly to the internet gateway. This is a route management violation. The suggested remediation in this case might be a list of ordered actions. The first being a recommendation to add the required routes to the Network Firewall subnet's route table to direct outgoing traffic to the internet gateway and to direct incoming traffic for destinations inside the VPC to `local`. The second recommendation is to replace the internet gateway route or the invalid Network Firewall route in the subnet's route table to direct outgoing traffic to the firewall endpoints. The third recommendation is to add required routes to the internet gateway's route table to direct incoming traffic to the firewall endpoints.

- **`AWS::EC2::InternetGateway`** – This is used for Network Firewall policies that have Monitor mode enabled.
  - **Route management violations** – The internet gateway is noncompliant if the internet gateway is not associated with a route table, or if there is an invalid route in the internet gateway route table.
  - **Remediation actions for route management violations** – Firewall Manager suggests possible remediation actions to remedy route management violations.

  Example 1 – Route management violation and remediation suggestions

  An internet gateway is not associated with a route table. The suggested remediation actions might be a list of ordered actions. The first action is to create a route table. The second action is to associate the route table with the internet gateway. The third action is to add the required route to the internet gateway route table.

  Example 2 – Route management violation and remediation suggestions

  The internet gateway is associated with a valid route table, but the route is configured improperly. The suggested remediation might be a list of ordered actions. The first suggestion is to remove the invalid route. The second is to add the required route to the internet gateway route table.

- **`AWS::NetworkFirewall::FirewallPolicy`** – This is used for Network Firewall policies. Firewall Manager displays information about a Network Firewall firewall policy that's been modified in a way that makes it noncompliant. The information provides the expected firewall policy and the policy that it found in the customer account, so you can compare stateless and stateful rule groups names and priority settings, custom action names,

and default stateless actions settings. The violation description component contains a description of the expected state of the resource, the current, noncompliant state, and if available, a description of what caused the discrepancy.

- **`AWS::EC2::VPC`** – This is used for DNS Firewall policies. Firewall Manager displays information about a VPC that's in scope of a Firewall Manager DNS Firewall policy, and that is noncompliant with the policy. The information provided includes the expected rule groups that are expected to be associated with the VPC and the actual rule groups. The violation description component contains a description of the expected state of the resource, the current, noncompliant state, and if available, a description of what caused the discrepancy.
- **`AWS::WAFv2::WebACL`** – This is used for AWS WAF policies whose configuration specifies retrofitting for existing web ACLs. Firewall Manager displays information about a web ACL that is associated with an in-scope resource, but is not fully compatible with retrofitting by Firewall Manager. For example, if the web ACL is also associated with a resource that's not in scope of the policy, Firewall Manager can't retrofit it.

AWS Firewall Manager integration with AWS Security Hub CSPM

This page explains how to use Firewall Manager and Security Hub CSPM together.

AWS Firewall Manager creates findings for resources that are out of compliance and for attacks that it detects, and it sends them to AWS Security Hub CSPM. For information about Security Hub CSPM findings, see Findings in AWS Security Hub CSPM.

When you use Security Hub CSPM and Firewall Manager, Firewall Manager automatically sends your findings to Security Hub CSPM. For information about getting started with Security Hub CSPM, see Setting Up AWS Security Hub CSPM in the AWS Security Hub CSPM User Guide.

Note

Firewall Manager only updates findings for policies that are under its management and for resources that it's monitoring.
Firewall Manager doesn't resolve findings for the following:

- Policies that have been deleted.
- Resources that have been deleted.

- Resources that have gone out of scope of the Firewall Manager policy, for example due to tag change or policy definition change.

How do I view my Firewall Manager findings?

To view your Firewall Manager findings in Security Hub CSPM, follow the guidance at Working with Findings in Security Hub CSPM and create a filter using the following settings:

- Attribute set to **Product Name**.
- Operator set to **EQUALS**.
- Value set to `Firewall Manager`. This setting is case sensitive.

Can I disable this?

You can disable the integration of AWS Firewall Manager findings with Security Hub CSPM through the Security Hub CSPM console. Choose **Integrations** in the navigation bar, then in the Firewall Manager pane, choose **Disable Integration**. For more information, see the AWS Security Hub CSPM User Guide.

AWS Firewall Manager finding types

- AWS WAF policy Firewall Manager findings
- AWS Shield Advanced policy Firewall Manager findings
- Security group common policy Firewall Manager findings
- Security group content audit policy Firewall Manager findings
- Security group usage audit policy Firewall Manager findings
- Amazon Route 53 Resolver DNS Firewall policy Firewall Manager findings
- AWS Config Firewall Manager findings

AWS WAF policy Firewall Manager findings

This page explains Firewall Manager findings for AWS WAF policies.

You can use Firewall Manager AWS WAF policies to apply AWS WAF rule groups to your resources in AWS Organizations. For more information, see Using AWS Firewall Manager policies.

Resource is missing Firewall Manager managed web ACL.

An AWS resource doesn't have the AWS Firewall Manager managed web ACL association in accordance with the Firewall Manager policy. You can enable Firewall Manager remediation on the policy to correct this.

- Severity – 80
- Status settings – PASSED/FAILED
- Updates – If Firewall Manager performs the remediation action, it will update the finding and the severity will lower from `HIGH` to `INFORMATIONAL`. If you perform the remediation, Firewall Manager will not update the finding.

Firewall Manager managed web ACL has misconfigured rule groups.

This is a AWS WAF Classic policy finding. The rule groups in a web ACL that's managed by Firewall Manager are not configured correctly, according to the Firewall Manager policy. This means that the web ACL is missing the rule groups that the policy requires. You can enable Firewall Manager remediation on the policy to correct this.

- Severity – 80
- Status settings – PASSED/FAILED
- Updates – If Firewall Manager performs the remediation action, it will update the finding and the severity will lower from `HIGH` to `INFORMATIONAL`. If you perform the remediation, Firewall Manager will not update the finding.

AWS Shield Advanced policy Firewall Manager findings

This page explains Firewall Manager findings for AWS Shield Advanced policies.

For information about AWS Shield Advanced policies, see Using security group policies in Firewall Manager to manage Amazon VPC security groups.

Resource lacks Shield Advanced protection.

An AWS resource that should have Shield Advanced protection, according to the Firewall Manager policy, doesn't have it. You can enable Firewall Manager remediation on the policy, which will enable the protection for the resource.

- Severity – 60
- Status settings – PASSED/FAILED

- Updates – If Firewall Manager performs the remediation action, it will update the finding and the severity will lower from `HIGH` to `INFORMATIONAL`. If you perform the remediation, Firewall Manager will not update the finding.

Shield Advanced detected attack against monitored resource.

Shield Advanced detected an attack on a protected AWS resource. You can enable Firewall Manager remediation on the policy.

- Severity – 70
- Status settings – None
- Updates – Firewall Manager does not update this finding.

Security group common policy Firewall Manager findings

This page explains Firewall Manager findings for security group common policies.

For information about security group common policies, see Using security group policies in Firewall Manager to manage Amazon VPC security groups.

Resource has misconfigured security group.

Firewall Manager has identified a resource that is missing the Firewall Manager managed security group associations that it should have, according to the Firewall Manager policy. You can enable Firewall Manager remediation on the policy, which creates the associations according to the policy settings.

- Severity – 70
- Status settings – PASSED/FAILED
- Updates – Firewall Manager updates this finding.

Firewall Manager replica security group is out of sync with primary security group.

A Firewall Manager replica security group is out of sync with its primary security group, according to their common security group policy. You can enable Firewall Manager remediation on the policy, which syncs the replica security groups with the primary.

- Severity – 80

- Status settings – PASSED/FAILED
- Updates – Firewall Manager updates this finding.

Security group content audit policy Firewall Manager findings

This page explains Firewall Manager findings for security group content audit policies.

For information about security group content audit policies, see Using security group policies in Firewall Manager to manage Amazon VPC security groups.

Security group is not in compliance with content audit security group.

A Firewall Manager security group content audit policy has identified a noncompliant security group. This is a customer-created security group that's in scope of the content audit policy and that doesn't comply with the settings defined by the policy and its audit security group. You can enable Firewall Manager remediation on the policy, which modifies the noncompliant security group to bring it into compliance.

- Severity – 70
- Status settings – PASSED/FAILED
- Updates – Firewall Manager updates this finding.

Security group usage audit policy Firewall Manager findings

This page explains Firewall Manager findings for security group usage audit policies.

For information about security group usage audit policies, see Using security group policies in Firewall Manager to manage Amazon VPC security groups.

Firewall Manager found redundant security group.

The Firewall Manager security group usage audit has identified a redundant security group. This is a security group with an identical rules set as another security group within the same Amazon Virtual Private Cloud instance. You can enable Firewall Manager automatic remediation on the usage audit policy, which replaces redundant security groups and with a single security group.

- Severity – 30
- Status settings – None

- Updates – Firewall Manager does not update this finding.

Firewall Manager found unused security group.

The Firewall Manager security group usage audit has identified an unused security group. This is a security group that's not referenced by any Firewall Manager common security group policy. You can enable Firewall Manager automatic remediation on the usage audit policy, which removes unused security groups.

- Severity – 30
- Status settings – None
- Updates – Firewall Manager does not update this finding.

Amazon Route 53 Resolver DNS Firewall policy Firewall Manager findings

This page explains Firewall Manager findings for Amazon Route 53 Resolver DNS Firewall policies.

For information about DNS Firewall policies, see Using Amazon Route 53 Resolver DNS Firewall policies in Firewall Manager.

Resource is missing DNS Firewall protection

A VPC is missing a DNS Firewall rule group association that's defined in the Firewall Manager DNS Firewall policy. The finding lists the rule group that's specified by the policy.

- Severity – 80

AWS Config Firewall Manager findings

This page explains Firewall Manager findings for AWS Config.

For information about AWS Config, see Enabling AWS Config for using Firewall Manager.

Account does not have AWS Config enabled in the Region.

Firewall Manager requires AWS Config to be enabled in your account and Region. To resolve this issue, enable AWS Config in the account and Region where you want to use Firewall Manager.

- Status settings – PASSED/FAILED
- Updates – Firewall Manager updates this finding.

> **Note**
>
> After you enable AWS Config, the compliance status changes to PASS, but the severity remains HIGH.

> **Note**
>
> In order for Firewall Manager to monitor policy compliance, AWS Config must continuously record configuration changes for protected resources. In your AWS Config configuration, the recording frequency must be set to **Continuous**, which is the default setting.

Security in your use of the AWS Firewall Manager service

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

> **Note**
>
> This section provides standard AWS security guidance for your use of the AWS Firewall Manager service and its AWS resources, such as Firewall Manager Network Firewall policies and security group policies.
> For information about protecting your AWS resources using Firewall Manager, see the rest of the Firewall Manager guide.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. The effectiveness of our security is regularly tested and verified by third-party auditors as part of

the AWS compliance programs. To learn about the compliance programs that apply to Firewall Manager, see AWS Services in Scope by Compliance Program.

- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your organization's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Firewall Manager. The following topics show you how to configure Firewall Manager to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Firewall Manager resources.

Topics

- Data protection in Firewall Manager
- Identity and Access Management for AWS Firewall Manager
- Logging and monitoring in Firewall Manager
- Compliance validation for Firewall Manager
- Resilience in Firewall Manager
- Infrastructure security in AWS Firewall Manager

Data protection in Firewall Manager

The AWS shared responsibility model applies to data protection in AWS Firewall Manager. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see Data Privacy FAQ. For information about data protection in Europe, see the General Data Protection Regulation (GDPR) Center.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.

- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Firewall Manager or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Firewall Manager entities—such as policies—are encrypted at rest, except in certain Regions where encryption is not available, including China (Beijing) and China (Ningxia). Unique encryption keys are used for each Region.

Identity and Access Management for AWS Firewall Manager

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Firewall Manager resources. IAM is an AWS service that you can use with no additional charge.

Topics

- Audience
- Authenticating with identities
- Managing access using policies
- How AWS Firewall Manager works with IAM
- Identity-based policy examples for AWS Firewall Manager

- AWS managed policies for AWS Firewall Manager
- Troubleshooting AWS Firewall Manager identity and access
- Using service-linked roles for Firewall Manager
- Cross-service confused deputy prevention

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Firewall Manager.

Service user – If you use the Firewall Manager service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Firewall Manager features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Firewall Manager, see Troubleshooting AWS Shield identity and access.

Service administrator – If you're in charge of Firewall Manager resources at your company, you probably have full access to Firewall Manager. It's your job to determine which Firewall Manager features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Firewall Manager, see How AWS Shield works with IAM.

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Firewall Manager. To view example Firewall Manager identity-based policies that you can use in IAM, see Identity-based policy examples for AWS Shield.

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be authenticated as the AWS account root user, an IAM user, or by assuming an IAM role.

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you don't use the root user for everyday tasks. For tasks that require root user credentials, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users to use federation with an identity provider to access AWS services using temporary credentials.

A *federated identity* is a user from your enterprise directory, web identity provider, or Directory Service that accesses AWS services using credentials from an identity source. Federated identities assume roles that provide temporary credentials.

For centralized access management, we recommend AWS IAM Identity Center. For more information, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see Require human users to use federation with an identity provider to access AWS using temporary credentials in the *IAM User Guide*.

An *IAM group* specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity with specific permissions that provides temporary credentials. You can assume a role by switching from a user to an IAM role (console) or by calling an AWS CLI or AWS API operation. For more information, see Methods to assume a role in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you attach to an identity (user, group, or role). These policies control what actions identities can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples include IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. You must specify a principal in a resource-based policy.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries** – Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – Set the maximum available permissions for resources in your accounts. For more information, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Advanced policies passed as a parameter when creating a temporary session for a role or federated user. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How AWS Firewall Manager works with IAM

Before you use IAM to manage access to Firewall Manager, learn what IAM features are available to use with Firewall Manager.

IAM features you can use with AWS Firewall Manager

IAM feature	Firewall Manager support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	No
ACLs	No
ABAC (tags in policies)	Yes
Temporary credentials	Yes
Forward access sessions (FAS)	Yes
Service roles	Partial
Service-linked roles	Yes

To get a high-level view of how Firewall Manager and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for Firewall Manager

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

To view examples of Firewall Manager identity-based policies, see Identity-based policy examples for AWS Firewall Manager.

Identity-based policy examples for Firewall Manager

To view examples of Firewall Manager identity-based policies, see Identity-based policy examples for AWS Firewall Manager.

Resource-based policies within Firewall Manager

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for Firewall Manager

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Include actions in a policy to grant permissions to perform the associated operation.

To see a list of Firewall Manager actions, see Actions defined by AWS Firewall Manager in the *Service Authorization Reference*.

Policy actions in Firewall Manager use the following prefix before the action:

```
fms
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "fms:action1",
      "fms:action2"
         ]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Describe`, include the following action:

```
"Action": "fms:Describe*"
```

To view examples of Firewall Manager identity-based policies, see Identity-based policy examples for AWS Firewall Manager.

Policy resources for Firewall Manager

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. As a best practice, specify a resource using its Amazon Resource Name (ARN). For actions that don't support resource-level permissions, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of Firewall Manager resource types and their ARNs, see Resources defined by AWS Firewall Manager in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by AWS Firewall Manager.

To view examples of Firewall Manager identity-based policies, see Identity-based policy examples for AWS Firewall Manager.

Policy condition keys for Firewall Manager

Supports service-specific policy condition keys: No

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element specifies when statements execute based on defined criteria. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of Firewall Manager condition keys, see Condition keys for AWS Firewall Manager in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by AWS Firewall Manager.

To view examples of Firewall Manager identity-based policies, see Identity-based policy examples for AWS Firewall Manager.

ACLs in Firewall Manager

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ABAC with Firewall Manager

Supports ABAC (tags in policies): Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes called tags. You can attach tags to IAM entities and AWS resources, then design ABAC policies to allow operations when the principal's tag matches the tag on the resource.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see Define permissions with ABAC authorization in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with Firewall Manager

Supports temporary credentials: Yes

Temporary credentials provide short-term access to AWS resources and are automatically created when you use federation or switch roles. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM and AWS services that work with IAM in the *IAM User Guide*.

Forward access sessions for Firewall Manager

Supports forward access sessions (FAS): Yes

Forward access sessions (FAS) use the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. For policy details when making FAS requests, see Forward access sessions.

Service roles for Firewall Manager

Supports service roles: Partial

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.

> **Warning**
>
> Changing the permissions for a service role might break Firewall Manager functionality. Edit service roles only when Firewall Manager provides guidance to do so.

Choosing an IAM role in Firewall Manager

To use the *`PutNotificationChannel`* API action in Firewall Manager, you must choose a role to allow Firewall Manager to access Amazon SNS so that the service can publish Amazon SNS messages on your behalf. For more information, see PutNotificationChannel in the *AWS Firewall Manager API Reference*.

The following shows an example SNS topic permission setting. To use this policy with your own custom role, replace the replace the `AWSServiceRoleForFMS` Amazon Resource Name (ARN) with the `SnsRoleName` ARN.

```
{
  "Sid": "AWSFirewallManagerSNSPolicy",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::account ID:role/aws-service-role/
fms.amazonaws.com/AWSServiceRoleForFMS"
  },
  "Action": "sns:Publish",
  "Resource": "SNS topic ARN"
}
```

For more information about Firewall Manager actions and resources, see the AWS Identity and Access Management guide topic Actions Defined by AWS Firewall Manager

Service-linked roles for Firewall Manager

Supports service-linked roles: Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing service-linked roles, see AWS services that work with IAM. Find a service in the table that includes a `Yes` in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Identity-based policy examples for AWS Firewall Manager

By default, users and roles don't have permission to create or modify Firewall Manager resources. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Create IAM policies (console) in the *IAM User Guide*.

For details about actions and resource types defined by Firewall Manager, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for AWS Firewall Manager in the *Service Authorization Reference*.

Topics

- Policy best practices
- Using the Firewall Manager console
- Allow users to view their own permissions
- Grant read access to your Firewall Manager security groups

Policy best practices

Identity-based policies determine whether someone can create, access, or delete Firewall Manager resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API

operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the Firewall Manager console

To access the AWS Firewall Manager console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Firewall Manager resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that users and roles can still use the Firewall Manager console, also attach the Firewall Manager *ConsoleAccess* or *ReadOnly* AWS managed policy to the entities. For more information, see Adding permissions to a user in the *IAM User Guide*.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
```

```
            "Resource": ["arn:aws-eusc:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Grant read access to your Firewall Manager security groups

Firewall Manager allows cross-account resource access, but it doesn't allow you to create cross-account resource protections. You can only create protections for resources from within the account that owns those resources.

The following is an example policy that grants permissions for the `fms:Get`,`fms:List`, and `ec2:DescribeSecurityGroups` actions on all resources.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "fms:Get*",
                "fms:List*",
                "ec2:DescribeSecurityGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
```

```
        }
    ]
}
```

AWS managed policies for AWS Firewall Manager

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

AWS managed policy: `AWSFMAdminFullAccess`

Use the `AWSFMAdminFullAccess` AWS managed policy to allow your administrators to access AWS Firewall Manager resources, including all Firewall Manager policy types. This policy doesn't include permissions for setting up Amazon Simple Notification Service notifications in AWS Firewall Manager. For information about how to setting up access for Amazon Simple Notification Service, see Setting up access for Amazon Simple Notification Service.

For the policy listing and details, see the IAM console at AWSFMAdminFullAccess. The rest of this section gives an overview of the policy settings.

Permission statements

This policy is grouped into statements based on the set of permissions.

- **AWS Firewall Manager policy resources** - Allows full administrative permissions to resources in AWS Firewall Manager, including all Firewall Manager policy types.
- **Write AWS WAF logs to Amazon Simple Storage Service** - Allows Firewall Manager to write and read AWS WAF logs in Amazon S3.
- **Create service-linked role** – Allows the administrator to create a service-linked role, which allows Firewall Manager to access resources in other services on your behalf. This permission allows creating the service-linked role only for use by Firewall Manager. For information about how Firewall Manager uses service-linked roles, see Using service-linked roles for Firewall Manager.
- **AWS Organizations** – Allows administrators to use Firewall Manager for an organization in AWS Organizations. After enabling trusted access for Firewall Manager in AWS Organizations, members of the admin account can view findings across their organization. For information about using AWS Organizations with AWS Firewall Manager, see Using AWS Organizations with other AWS services in the *AWS Organizations User Guide*.

Permission categories

The following lists the types of permissions in the policy and the permissions that they provide.

- `fms` – Work with AWS Firewall Manager resources.
- `waf` and `waf-regional` – Work with AWS WAF Classic policies.
- `elasticloadbalancing` – Associate AWS WAF web ACLsto Elastic Load Balancers.
- `firehose` – View information about AWS WAF logs.
- `organizations` – Work with AWS Organizations resources.
- `shield` – View the subscription state of AWS Shield policies.
- `route53resolver` – Work with Route 53 Private DNS for VPCs rule groups in an Route 53 Private DNS for VPCs policy.
- `wafv2` – Work with AWS WAFV2 policies.
- `network-firewall` – Work with AWS Network Firewall policies.
- `ec2` – View policy Availability Zones and Regions.
- `s3` – View information about AWS WAF logs.

AWS managed policy: FMSServiceRolePolicy

This policy allows AWS Firewall Manager to manage AWS resources on your behalf in Firewall Manager and in integrated services. This policy is attached to the service-linked role AWSServiceRoleForFMS. For more information about the service-linked role, see Using service-linked roles for Firewall Manager.

For policy details, see the IAM console at FMSServiceRolePolicy.

AWS managed policy: AWSFMAdminReadOnlyAccess

Grants read-only access to all AWS Firewall Manager resources.

For the policy listing and details, see the IAM console at AWSFMAdminReadOnlyAccess. The rest of this section gives an overview of the policy settings.

Permission categories

The following lists the types of permissions in the policy and the information that the permissions allow read only access to.

- fms – AWS Firewall Manager resources.
- waf and waf-regional – AWS WAF Classic policies.
- firehose – AWS WAF logs.
- organizations – AWS Organizations resources.
- shield – AWS Shield policies.
- route53resolver – Route 53 Private DNS for VPCs rule groups in an Route 53 Private DNS for VPCs policy.
- wafv2 – Your AWS WAFV2 rule groups and AWS Managed Rules rule groups that are available in AWS WAFV2.
- network-firewall – AWS Network Firewall rule groups and rule group metadata.
- ec2 – AWS Network Firewall policy Availability Zones and Regions .
- s3 – AWS WAF logs.

AWS managed policy: AWSFMMemberReadOnlyAccess

Grants read-only access to AWS Firewall Manager member resources. For the policy listing and details, see the IAM console at AWSFMMemberReadOnlyAccess.

Firewall Manager updates to AWS managed policies

View details about updates to AWS managed policies for Firewall Manager since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Firewall Manager document history page at Document history.

Change	Description	Date
FMSServiceRolePolicy – Updated policy	Added permissions to the Firewall Manager service policy. Added the following permissions required for Amazon CloudFront: • `cloudfront:AssociateDistributionWebACL` – Grants permission to associate an AWS WAF web ACL with a CloudFront distribution • `cloudfront:DisassociateDistributionWebACL` – Grants permission to disassociate an AWS WAF web ACL with a CloudFront distribution	2025-05-21
FMSServiceRolePolicy – Updated policy	Added permissions to the Firewall Manager service policy. Added `BatchGetResourceConfig` permissio	2025-02-10

Change	Description	Date
	ns to get resource configura tion statuses in batches. See the updated policy in the IAM console: FMSServiceRolePoli cy.	
FMSServiceRolePolicy – Updated policy	Added permissions to the Firewall Manager service role policy. Added the ability to read Network Firewall TLS configuration information. See the updated policy in the IAM console: FMSServic eRolePolicy.	2024-07-22
FMSServiceRolePolicy – Updated policy	Added permissions for managing network ACLs. See the updated policy in the IAM console: FMSServic eRolePolicy.	2024-04-22
FMSServiceRolePolicy – Updated policy	Added permissions that allow Firewall Manager to describe whether the specified AWS Config rules are compliant. See the updated policy in the IAM console: FMSServic eRolePolicy.	2023-04-21

Change	Description	Date
FMSServiceRolePolicy – Updated policy	Added permissions that allow Firewall Manager to describe Amazon EC2 instance and network interface attributes. See the updated policy in the IAM console: FMSServiceRolePolicy.	2022-11-15
AWSFMAdminReadOnlyAccess – Updated policy	Added permissions to support AWS WAFV2, Shield, Network Firewall, DNS Firewall, Amazon VPC security group, policies. See the updated policy in the IAM console: AWSFMAdminReadOnlyAccess.	2022-11-02
AWSFMAdminFullAccess – Updated policy	Added permissions to support AWS WAFV2, Shield, Network Firewall, DNS Firewall, Amazon VPC security group, policies. Removed Amazon SNS permissions. See the updated policy in the IAM console: AWSFMAdminFullAccess.	2022-10-21
`FMSServiceRolePolicy` – New permissions for AWS Firewall Manager third-party firewall policies	This change allows Firewall Manager to create and delete the Amazon EC2 VPC endpoints associated with a third-party firewall policy.	2022-03-30

Change	Description	Date
`FMSServiceRolePolicy` – New permissions for AWS Network Firewall policies	Added new permissions to support deployment of firewalls for Network Firewall policies. The new permissio ns allow the retrieval of information about Availability Zones for accounts that are in scope of a policy.	2022-02-16
`FMSServiceRolePolicy` – New permissions for AWS Shield policies	Added new permissions to retrieve tags for AWS WAF regional and AWS WAF global resources. Added AWS WAF regional permissio ns to retrieve web ACLs using a resource ARN. Added permissions to support Shield automatic application layer DDoS mitigation.	2022-01-07
`FMSServiceRolePolicy` – New permissions for AWS Shield policies	Added new permission to retrieve tags for Elastic Load Balancing resources.	2021-11-18
`FMSServiceRolePoli cy` – New permissions for security group and AWS Network Firewall policies	Added new permissions to enable centralized logging for AWS Network Firewall policies. Additionally, read-only Amazon EC2 permissio ns were added to support changes to the Config service that impact how AWS Firewall Manager queries resources for security group policies.	2021-09-29

Change	Description	Date
`FMSServiceRolePolicy` – ARN formats for AWS WAF resources	Updated the `FMSServiceRolePolicy` to standardize the ARN formats for AWS WAF resources. The updated ARN formats are `arn:aws:waf:*:*:*` and `arn:aws:waf-regional:*:*:*`.	2021-08-12
`FMSServiceRolePolicy` – Additional regions in China	AWS Firewall Manager has enabled `FMSServiceRolePolicy` for the BJS and ZHY regions in China.	2021-08-12
`FMSServiceRolePolicy` – Update to the existing policy	Added new permissions to allow AWS Firewall Manager to manage Amazon Route 53 Resolver DNS Firewall. This change allows Firewall Manager to configure Amazon Route 53 Resolver DNS Firewall associations. This permits you to use Firewall Manager to provide DNS Firewall protections for your VPCs throughout your organization in AWS Organizations.	2021-03-17
Firewall Manager started tracking changes	Firewall Manager started tracking changes for its AWS managed policies.	2021-03-02

Troubleshooting AWS Firewall Manager identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Firewall Manager and IAM.

Topics

- I am not authorized to perform an action in Firewall Manager
- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my Firewall Manager resources

I am not authorized to perform an action in Firewall Manager

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `fms:`*`GetWidget`* permissions.

```
User: arn:aws-eusc:iam::123456789012:user/mateojackson is not authorized to perform:
 fms:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `fms:`*`GetWidget`* action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to Firewall Manager.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Firewall Manager. However, the action requires the service to have

permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws-eusc:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my Firewall Manager resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Firewall Manager supports these features, see How AWS Shield works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.

Using service-linked roles for Firewall Manager

AWS Firewall Manager uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to Firewall Manager. Service-linked roles are predefined by Firewall Manager and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up Firewall Manager easier because you don’t have to manually add the necessary permissions. Firewall Manager defines the permissions of its service-linked roles,

and unless defined otherwise, only Firewall Manager can assume its roles. The defined permissions include the trust policy and the permissions policy. That permissions policy can't be attached to any other IAM entity.

You can delete a service-linked role only after first deleting the role's related resources. This protects your Firewall Manager resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for Firewall Manager

AWS Firewall Manager uses the service-linked role name AWSServiceRoleForFMS to allow Firewall Manager to call AWS services on your behalf for management of firewall policies and AWS Organizations account resources. This policy is attached to the AWS managed role `AWSServiceRoleForFMS`. For more information about the managed role, see AWS managed policy: `FMSServiceRolePolicy`.

The AWSServiceRoleForFMS service-linked role trusts the service to assume the role `fms.amazonaws.com`.

The role permissions policy allows Firewall Manager to complete the following actions on the specified resources:

- `waf` - Manage AWS WAF Classic web ACLs, rule group permissions, and the web ACLs associations in your account.
- `ec2` - Manage security groups on elastic network interfaces and Amazon EC2 instances. Manage network ACLs on Amazon VPC subnets.
- `vpc` - Manage subnets, route tables, tags, and endpoints in Amazon VPC.
- `wafv2` - Manage AWS WAF web ACLs, rule group permissions, and the web ACLs associations in your account.
- `cloudfront` - Create web ACLs to protect CloudFront distributions.
- `config` - Manage Firewall Manager-owned AWS Config rules in your account.
- `iam` - Manage this service-linked role, and creates required AWS WAF and Shield service-linked roles if configuring logging for AWS WAF and Shield policies.

- `organization` - Create a service-linked role owned by Firewall Manager to manage AWS Organizations resources used by Firewall Manager.
- `shield` - Manage AWS Shield protections and L7 mitigation configurations for resources in your account.
- `ram` - Manage AWS RAM resource sharing for DNS Firewall rule groups and Network Firewall rule groups.
- `network-firewall` - Manage Firewall Manager-owned AWS Network Firewall resources and dependent Amazon VPC resources in your account.
- `route53resolver` - Manage Firewall Manager-owned DNS Firewall associations in your account.

See the full policy in the IAM console: FMSServiceRolePolicy.

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

Creating a service-linked role for Firewall Manager

You don't need to manually create a service-linked role. When you enable Firewall Manager logging on the AWS Management Console, or you make a `PutLoggingConfiguration` request in the Firewall Manager CLI or the Firewall Manager API, Firewall Manager creates the service-linked role for you.

You must have the `iam:CreateServiceLinkedRole` permission to enable logging.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you enable Firewall Manager logging, Firewall Manager creates the service-linked role for you again.

Editing a service-linked role for Firewall Manager

Firewall Manager doesn't allow you to edit the AWSServiceRoleForFMS service-linked role. After you create a service-linked role, you can't change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting a service-linked role for Firewall Manager

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

> **Note**
>
> If the Firewall Manager service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To delete the service-linked role using IAM

Use the IAM console, the IAM CLI, or the IAM API to delete the AWSServiceRoleForFMS service-linked role. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

Supported Regions for Firewall Manager service-linked roles

Firewall Manager supports using service-linked roles in all of the regions where the service is available. For more information, see Firewall Manager endpoints and quotas.

Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. In AWS, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the *calling service*) calls another service (the *called service*). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it should not otherwise have permission to access. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in resource policies to limit the permissions that AWS Firewall Manager gives another service to the resource. Use `aws:SourceArn` if you want only one resource to be associated with the cross-service access. Use `aws:SourceAccount` if you want to allow any resource in that account to be associated with the cross-service use.

The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` global condition context key with the full ARN of the resource. If you don't know the full ARN of the resource or if you are specifying multiple resources, use the `aws:SourceArn` global context condition key with wildcard characters (*) for the unknown portions of the ARN. For example, `arn:aws-eusc:fms:*:account-id:*`.

If the `aws:SourceArn` value does not contain the account ID, such as an Amazon S3 bucket ARN, you must use both global condition context keys to limit permissions.

The value of `aws:SourceArn` must be the AWS Firewall Manager administrator's AWS account.

The following examples show how you can use the `aws:SourceArn` global condition context key in Firewall Manager to prevent the confused deputy problem.

The following example shows how to prevent the confused deputy problem by using the `aws:SourceArn` global condition context key in the Firewall Manager role trust policy. Replace *Region* and *account-id* with your own information.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Sid": "ConfusedDeputyPreventionExamplePolicy",
        "Effect": "Allow",
        "Principal": {
            "Service": "servicename.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "ArnLike": {
                "aws:SourceArn": [
                "arn:aws-eusc:fms:eusc-de-east-1:123456789012:${*}",
    "arn:aws-eusc:fms:eusc-de-east-1:123456789012:policy/*"
                ]
            },
            "StringEquals": {
                "aws:SourceAccount": "123456789012"
            }
        }
    }
```

```
}
```

Logging and monitoring in Firewall Manager

Monitoring is an important part of maintaining the reliability, availability, and performance of Firewall Manager and your AWS solutions. You should collect monitoring data from all parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your Firewall Manager resources and responding to potential events:

Amazon CloudWatch Alarms

Using CloudWatch alarms, you watch a single metric over a time period that you specify. If the metric exceeds a given threshold, CloudWatch sends a notification to an Amazon SNS topic or AWS Auto Scaling policy. For more information, see Monitoring with Amazon CloudWatch.

AWS CloudTrail Logs

CloudTrail provides a record of actions taken by a user, role, or an AWS service in Firewall Manager. Using the information collected by CloudTrail, you can determine the request that was made to Firewall Manager, the IP address from which the request was made, who made the request, when it was made, and additional details. For more information, see Logging API calls with AWS CloudTrail.

Compliance validation for Firewall Manager

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. For more information about your compliance responsibility when using AWS services, see AWS Security Documentation.

Resilience in Firewall Manager

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure security in AWS Firewall Manager

As a managed service, AWS Firewall Manager is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access Firewall Manager through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

AWS Firewall Manager quotas

AWS Firewall Manager is subject to the following quotas (formerly referred to as limits).

AWS Firewall Manager has default quotas that you might be able to increase and fixed quotas.

The security group policies and network ACL policies that are managed by Firewall Manager are subject to standard Amazon VPC quotas. For more information, see Amazon VPC Quotas in the Amazon VPC User Guide.

Each Firewall Manager Network Firewall policy creates a Network Firewall firewall with an associated firewall policy and its rule groups. These Network Firewall resources are subject to the quotas listed at AWS Network Firewall quotas in the *Network Firewall Developer Guide*.

Soft quotas

AWS Firewall Manager has default quotas on the number of entities per Region. You can request an increase in these quotas.

All policy types

Resource	Default quota per Region
Accounts per organization in AWS Organizations	Varies. An invitation sent to an account counts against this quota. The count is returned if the invited account declines, the management account cancels the invitation, or the invitation expires.
Firewall Manager policies per organization in AWS Organizations.	50. The Region specifications `Global` and `US East (N.`

Resource	Default quota per Region
	`Virginia) Region` refer to the same Region, so this limit applies to the total combined policies for the two of them.
Organizational units in scope per Firewall Manager policy.	20
Accounts in scope of a Firewall Manager policy if you explicitly include and exclude individual accounts.	200
Accounts in scope of a Firewall Manager policy if you do not explicitly include or exclude individual accounts.	2,500
Accounts that an organization in AWS Organizations can contain for the organization to be onboarded by Firewall Manager. The count includes the Firewall Manager administrator account.	10,000
Tags that include or exclude resources per Firewall Manager policy.	8
Number of resource sets per account.	20
Number of resources per resource set.	100
Number of resources sets per Firewall Manager policy.	5

AWS WAF policies

Resource	Default quota per Region
AWS WAF rule groups per Firewall Manager administrator account.	100
AWS WAF Classic rule groups per Firewall Manager administrator account.	10

Resource	Default quota per Region
Rule groups per AWS WAF policy.	50
Partner rule groups per AWS WAF policy.	1

Common security group policies

Resource	Default quota per Region.
Primary security groups per policy.	3
Amazon VPC instances in scope per policy per account, including shared VPCs.	100

Content audit security group policies

Resource	Default quota per Region
Audit security groups per policy.	1
Applications per application list.	50
Custom managed application lists for rules that allow all traffic.	1
Custom managed application lists per policy rules.	1
Custom managed application lists per account.	10
Protocols per protocol list.	5
Custom managed protocol lists for any setting in a policy.	1
Custom managed protocol lists per account.	10

Network ACL policies

Resource	Default quota per Region
Number of inbound rules per network ACL policy, used for first or last rules. For example, you can have 5 first and 0 last inbound rules, or 2 first and 3 last, but you can't have 4 first and 2 last.	5
Number of outbound rules per network ACL policy, used for first or last rules. For example, you can have 5 first and 0 last outbound rules, or 2 first and 3 last, but you can't have 4 first and 2 last.	5

Network Firewall policies

Resource	Default quota per Region
The number of IPV4 CIDRs that you can provide for a single policy.	50
Stateful rule group capacity per Network Firewall policy.	30,000

DNS Firewall policies

Resource	Default quota per Region
DNS Firewall rule groups per DNS Firewall policy.	2

Hard quotas

The following per-Region quotas related to AWS Firewall Manager can't be changed.

All policy types

Resource	Quota per Region
The maximum number of Firewall Manager administrators you can have in an AWS Organizations organization. You must have at one default administrator, and as many as nine additional Firewall Manager administrators.	10

AWS WAF policies

Resource	Quota per Region
Total web ACL capacity units (WCU) for the rule groups in an AWS WAF policy.	5,000

AWS WAF Classic policies

Resource	Quota per Region
AWS WAF Classic rule groups per policy.	2: 1 customer-created rule group and 1 AWS Marketplace rule group.
AWS WAF Classic rules per Firewall Manager AWS WAF Classic rule group.	10

Network Firewall policies

Resource	Quota per Region
Number of VPCs that can be automatically remediated for a single policy.	1,000
Stateless rule groups per Network Firewall policy.	20
Stateful rule groups per Network Firewall policy.	20

Resource	Quota per Region
Stateless rule group capacity per Network Firewall policy.	30,000

Migrating AWS WAF Classic Web ACLs in Firewall Manager

There are two scenarios where Firewall Manager might use AWS WAF Classic WebACLs:

1. With a AWS WAF Classic policy
2. With a Shield Advanced policy created before January 2022

Migrating Web ACLs in AWS WAF Classic Policies

To migrate web ACLs from a AWS WAF Classic policy, you must first migrate any AWS WAF Classic rule groups to AWS WAF (v2) rule groups. Then you can create a new policy using the migrated rule groups.

1. Migrate your AWS WAF Classic rule groups to AWS WAF (v2) rule groups using this migration script: AWS WAF bulk migration script.
2. Create a new AWS WAF policy with the following settings:
   - Use the newly migrated AWS WAF (v2) rule groups
   - Enable automatic remediation
3. For each account you want to migrate:
   a. Remove the account from the old AWS WAF Classic policy
   b. Wait approximately 2-3 minutes
   c. Add the account to the scope of the new AWS WAF policy
   d. (Optional) Use resource tag filtering to narrow the policy scope to specific resources
4. Verify the migration:
   a. Confirm that the new AWS WAF policy has created v2 web ACLs
   b. Verify that Firewall Manager has associated the new web ACLs with the appropriate resources

Migrating Web ACLs in Shield Advanced Policies

Automatic application layer DDoS mitigation in Firewall Manager works only with web ACLs that were created using AWS WAF (v2). If you want to use automatic mitigation in your Firewall Manager policies, and your policies currently use AWS WAF Classic web ACLs, you must migrate them to AWS WAF (v2). You can either migrate all web ACLs at once or migrate them one account at a time.

Migrating All Web ACLs at Once

To migrate all web ACLs in your Shield Advanced policy at once, you can use the policy's automatic remediation feature:

1. Open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fms.
2. Choose your Shield Advanced policy.
3. Enable automatic remediation and choose the option to replace AWS WAF Classic web ACLs with AWS WAF (v2) web ACLs.

Firewall Manager creates new AWS WAF (v2) web ACLs as needed and manages the migration of resource associations from Classic to v2 web ACLs.

Migrating Web ACLs One Account at a Time

To migrate web ACLs one account at a time, follow these steps:

1. Create a new Shield Advanced policy with the following settings:
   - Set automatic application layer DDoS mitigation to *Disabled*
   - Enable automatic remediation
2. For each account you want to migrate:
   a. Remove the account from the old Shield Advanced policy
   b. Wait approximately 2-3 minutes
   c. Add the account to the scope of the new Shield Advanced policy
   d. (Optional) Use resource tag filtering to narrow the policy scope to specific resources
3. Verify the migration:
   a. Confirm that the new Shield Advanced policy has created AWS WAF (v2) web ACLs

b. Verify that Firewall Manager has associated the new web ACLs with the appropriate resources

Monitoring AWS WAF, AWS Firewall Manager, and AWS Shield Advanced

Monitoring is an important part of maintaining the reliability, availability, and performance of your services.

> **Note**
>
> For information about monitoring your Shield Advanced resources and identifying possible DDoS events using Shield Advanced, see AWS Shield.

As you start monitoring these services, you should create a monitoring plan that includes answers to the following questions:

- What are your monitoring goals?
- What resources will you monitor?
- How often will you monitor these resources?
- What monitoring tools will you use?
- Who will perform the monitoring tasks?
- Who should be notified when something goes wrong?

The next step is to establish a baseline for normal performance in your environment, by measuring performance at various times and under different load conditions. As you monitor AWS WAF, Firewall Manager, Shield Advanced and related services, store historical monitoring data so that you can compare it with current performance data, identify normal performance patterns and performance anomalies, and devise methods to address issues.

For AWS WAF, you should monitor the following items at a minimum to establish a baseline:

- The number of allowed web requests
- The number of blocked web requests

Topics

- Monitoring tools

- Monitoring with Amazon CloudWatch
- Logging API calls with AWS CloudTrail

Monitoring tools

AWS provides various tools that you can use to monitor AWS WAF and AWS Shield Advanced. You can configure some of these tools to do the monitoring for you, while other tools require manual intervention. We recommend that you automate monitoring tasks as much as possible.

Automated monitoring tools

You can use the following automated monitoring tools to watch AWS WAF and AWS Shield Advanced and report when something is wrong:

- **protection pack (web ACL) traffic overview dashboards** – Access summaries of the web traffic that a protection pack (web ACL) evaluates by going to the web ACL's page in the AWS WAF console and opening the **Traffic overview** tab.

  The traffic overview dashboards provide near real-time summaries of the Amazon CloudWatch metrics that AWS WAF collects when it evaluates your application web traffic. You can see summaries for all of your web traffic and for traffic evaluated by the intelligent threat mitigation rule groups, inlcuding specialized AI Traffic Analysis for AI bot and agent activity.

  For more information, see Traffic overview dashboards for protection packs (web ACLs) or go to the dashboards in the console.
- **Amazon CloudWatch Alarms** – Watch a single metric over a time period you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon Simple Notification Service (Amazon SNS) topic or Amazon EC2 Auto Scaling policy. Alarms invoke actions for sustained state changes only. CloudWatch alarms will not invoke actions simply because they are in a particular state; the state must have changed and been maintained for a specified number of periods. For more information, see Monitoring CloudFront Activity Using CloudWatch.

  > **Note**
  >
  > CloudWatch metrics and alarms are not enabled for AWS Firewall Manager.

Not only can you use CloudWatch to monitor AWS WAF and Shield Advanced metrics as described in Monitoring with Amazon CloudWatch, you also should use CloudWatch to monitor activity for your protected resources. For more information, see the following:

- Monitoring CloudFront Activity Using CloudWatch in the *Amazon CloudFront Developer Guide*
- Logging and monitoring in Amazon API Gateway in the *API Gateway Developer Guide*
- CloudWatch Metrics for Your Application Load Balancer in the *Elastic Load Balancing User Guide*
- Monitoring and Logging in the *AWS AppSync Developer Guide*
- Logging and monitoring in Amazon Cognito in the *Amazon Cognito Developer Guide*
- Viewing App Runner logs streamed to CloudWatch Logs and Viewing App Runner service metrics reported to CloudWatch in the *AWS App Runner Developer Guide*

- **Amazon CloudWatch Logs** – Monitor, store, and access your log files from AWS CloudTrail or other sources. For more information, see What is Amazon CloudWatch Logs?.
- **Amazon CloudWatch Events** – Automate your AWS services and respond automatically to system events. Events from AWS services are delivered to CloudWatch Events in near real time, and you can specify automated actions to take when an event matches a rule that you write. For more information, see What is Amazon CloudWatch Events?
- **AWS CloudTrail Log Monitoring** – Share log files between accounts, monitor CloudTrail log files in real time by sending them to CloudWatch Logs, write log-processing applications in Java, and validate that your log files have not changed after delivery by CloudTrail. For more information, see Logging API calls with AWS CloudTrail and Working with CloudTrail Log Files in the *AWS CloudTrail User Guide*.
- **AWS Config** – View the configuration of AWS resources in your AWS account, including how the resources are related to one another and how they were configured in the past so that you can see how the configurations and relationships change over time.

Manual monitoring tools

Another important part of monitoring AWS WAF and AWS Shield Advanced involves manually monitoring those items that the CloudWatch alarms don't cover. You can view the AWS WAF, Shield Advanced, CloudWatch, and other AWS Management Console dashboards to see the state of your AWS environment. We recommend that you also check the log files for your web ACLs and rules.

- For example, to view the AWS WAF dashboard:
  - On the **Requests** tab of the AWS WAF **Web ACLs** page, view a graph of total requests and requests that match each rule that you have created. For more information, see Viewing a sample of web requests.
- View the CloudWatch home page for the following:
  - Current alarms and status
  - Graphs of alarms and resources
  - Service health status

  In addition, you can use CloudWatch to do the following:
  - Create customized dashboards to monitor the services that you care about.
  - Graph metric data to troubleshoot issues and discover trends.
  - Search and browse all of your AWS resource metrics.
  - Create and edit alarms to be notified of problems.

Monitoring with Amazon CloudWatch

You can monitor web requests and web ACLs and rules using Amazon CloudWatch, which collects and processes raw data from AWS WAF and AWS Shield Advanced into readable, near real-time metrics. You can use statistics in Amazon CloudWatch to gain a perspective on how your web application or service is performing. For more information, see What is CloudWatch in the *Amazon CloudWatch User Guide*.

> **Note**
>
> CloudWatch metrics and alarms are not enabled for Firewall Manager.

You can create an Amazon CloudWatch alarm that sends an Amazon SNS message when the alarm changes state. An alarm watches a single metric over a time period that you specify, and performs one or more actions based on the value of the metric relative to a specified threshold over a number of time periods. The action is a notification sent to an Amazon SNS topic or Auto Scaling policy. Alarms invoke actions for sustained state changes only. CloudWatch alarms do not invoke actions simply because they are in a particular state; the state must have changed and been maintained for a specified number of periods.

Topics

- Viewing metrics and dimensions
- AWS WAF metrics and dimensions
- AWS Shield Advanced metrics
- AWS Firewall Manager notifications

Viewing metrics and dimensions

Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace. AWS Firewall Manager doesn't record metrics.

- The AWS WAF namespace is `AWS/WAFV2`
- The Shield Advanced namespace is `AWS/DDoSProtection`

> **Note**
>
> AWS WAF reports metrics once a minute.
> Shield Advanced reports metrics once a minute during an event and less frequently other times.

Use the following procedures to view the metrics for AWS WAF and AWS Shield Advanced.

To view metrics using the CloudWatch console

1. Sign in to the AWS Management Console and open the CloudWatch console at https://eusc-de-east-1.console.amazonaws-eusc.eu/cloudwatch/.
2. If necessary, change the Region to the one where your AWS resources are located. For CloudFront, choose the US East (N. Virginia) Region.
3. In the navigation pane, under **Metrics**, choose **All metrics** and then search under the **Browse** tab for the service.

To view metrics using the AWS CLI

- For AWS/WAFV2, at a command prompt use the following command:

```
aws cloudwatch list-metrics --namespace "AWS/WAFV2"
```

For Shield Advanced, at a command prompt use the following command:

```
aws cloudwatch list-metrics --namespace "AWS/DDoSProtection"
```

AWS WAF metrics and dimensions

AWS WAF reports metrics once a minute. AWS WAF provides metrics and dimensions in the `AWS/WAFV2` namespace.

You can see summary information for AWS WAF metrics through the AWS WAF console, in the protection pack (web ACL)'s traffic overview tab. For more information, go to the console or see Traffic overview dashboards for protection packs (web ACLs).

You can see the following metrics for protection packs (web ACLs), rules, rule groups, and labels.

> **Note**
>
> If your Application Load Balancer is associated with a protection pack (web ACL) that has no rules or other active configurations, AWS WAF does not provide sampled requests or publish CloudWatch metrics for that Application Load Balancer. For more information about this behavior and how to enable observability features, see AWS WAF Distributed Denial of Service (DDoS) prevention.

- **Your rules** – Metrics are grouped by the rule action. For example, when you test a rule in Count mode, its matches are listed as `Count` metrics for the protection pack (web ACL).
- **Your rule groups** – The metrics for your rule groups are listed under the rule group metrics.
- **Rule groups owned by another account** – Rule group metrics are generally visible only to the rule group owner. However, if you override the rule action for a rule, the metrics for that rule will be listed under your protection pack (web ACL) metrics. Additionally, labels added by any rule group are listed in your protection pack (web ACL) metrics.

Count action rules in rule groups do NOT emit web ACL dimension metrics - only Rule, RuleGroup, and Region dimensions. This applies even when the rule group is referenced in a web ACL.

Rule groups in this category are AWS Managed Rules for AWS WAF, AWS Marketplace rule groups, Recognizing rule groups provided by other services, and rule groups that are shared with you by another account. When a protection pack (web ACL) is deployed through Firewall Manager, any rules within the WebACL that have a Count action will not display their metrics in the member account.

- **Labels** - Labels that were added to a web request during evaluation are listed in the protection pack (web ACL) label metrics. You can access the metrics for all labels, regardless of whether they were added by your rules and rule groups or by rules in a rule group that another account owns.

Topics

- AWS WAF core metrics and dimensions
- Label metrics and dimensions
- Free bot visibility metrics and dimensions
- Account metrics and dimensions
- AWS WAF usage metrics

AWS WAF core metrics and dimensions

AWS WAF core metrics

Metric	Description
`AllowedRequests`	The number of allowed web requests. Reporting criteria: There is a nonzero value. Valid statistics: Sum
`BlockedRequests`	The number of blocked web requests. Reporting criteria: There is a nonzero value. Valid statistics: Sum

Metric	Description
CountedRequests	The number of counted web requests. Reporting criteria: There is a nonzero value. A counted web request is one that matches at least one of the rules. Request counting is typically used for testing. Valid statistics: Sum
CaptchaRequests	The number of web requests that had CAPTCHA controls applied. It represents a terminating rule and does not include RequestsWithValidCaptchaToken . Reporting criteria: There is a nonzero value. A CAPTCHA web request is one that matches a rule that has a CAPTCHA action setting. This metric records all requests that match, regardless of whether the CAPTCHA token is expired, invalid, absent, or has a domain mismatch. Valid statistics: Sum
RequestsWithValidCaptchaToken	The number of web requests that had CAPTCHA controls applied and that had a valid CAPTCHA token. Reporting criteria: There is a nonzero value. Valid statistics: Sum

Metric	Description
CaptchasAttempted	The number of solutions that were submitted by an end user in response to a CAPTCHA puzzle challenge. Reporting criteria: There is a nonzero value. Valid statistics: Sum
CaptchasSolved	The number of CAPTCHA puzzle solutions submitted that successfully solved the puzzle. Reporting criteria: There is a nonzero value. Valid statistics: Sum
ChallengeRequests	The number of web requests that had challenge controls applied. It represents a terminating rule and does not include RequestsWithValidChallengeToken . Reporting criteria: There is a nonzero value. A challenge web request is one that matches a rule that has a Challenge action setting. This metric records all requests that match, regardless of whether the challenge token is expired, invalid, absent, or has a domain mismatch. Valid statistics: Sum

Metric	Description
ChallengesAttempted	The number of attempts that were submitted by an end user in response to a silent challenge served by AWS WAF. This includes challenges served by a Challenge rule action and challenges run as part of a CAPTCHA action when no token is present. Reporting criteria: There is a nonzero value. Valid statistics: Sum
ChallengesSolved	The number of silent challenge solutions submitted that successfully passed a silent challenge served by AWS WAF. This includes challenges served by a Challenge rule action and challenges run as part of a CAPTCHA action when no token is present. Reporting criteria: There is a nonzero value. Valid statistics: Sum
PassedRequests	The number of passed requests. This is only used for requests that go through a rule group evaluation without matching any of the rule group rules. Reporting criteria: There is a nonzero value. Valid statistics: Sum
RequestsWithValidChallengeToken	The number of web requests that had challenge controls applied and that had a valid challenge token. Reporting criteria: There is a nonzero value. Valid statistics: Sum

Metric	Description
LowReputationPacketsDropped	The number of packets dropped from known malicious sources. This metric is recorded when a request is blocked by resource-level DDoS protectio n. Reporting criteria: There is a nonzero value. Valid statistics: Sum This metric is published to the AWS/Appli cationELB namespace.
LowReputationRequestsDenied	The number of HTTP requests denied with HTTP 403 responses. This metric is recorded when a request is blocked by resource-level DDoS protectio n. Reporting criteria: There is a nonzero value. Valid statistics: Sum This metric is published to the AWS/Appli cationELB namespace.
MonetizeRequests	The number of web requests that had monetize controls applied. It represents a terminating rule that has a Monetize action. Reporting criteria: There is a nonzero value. A monetize web request is one that matches a rule that has a Monetize action setting. This metric records all requests that match. Valid statistics: Sum

Metric	Description
MonetizeRequestsFailedVerification	The number of web requests that had monetize controls applied and failed monetize verification. Reporting criteria: There is a nonzero value. Valid statistics: Sum

AWS WAF core dimensions

Dimension	Description
Region	Required for all protected resource types except for Amazon CloudFront distributions.
Rule	One of the following: • The metric name of the Rule. • ALL, which represents all rules within a WebACL or RuleGroup . • Default_Action (only when combined with the WebACL dimension), which represents the action assigned to any request whose evaluation wasn't terminated by the action of a rule in the protection pack (web ACL).
RuleGroup	The metric name of the RuleGroup .
WebACL	The metric name of the WebACL.
WebACLArn	The Amazon Resource Name (ARN) of the web ACL. This dimension is only available when AWS WAF is enabled.
ResourceType	The type of the protected resource, such as CF, APIGW, or ALB.

Dimension	Description
`Resource`	The Amazon Resource Name (ARN) of the protected resource. This dimension does not include App Runner resource ARNs.
`Country`	The country of origin of the request. This is the two-character designation from the Internati onal Organization for Standardization (ISO) 3166 standard. For example, US for the United States and UA for Ukraine. If a request has an `X-Forwarded-For` header, AWS WAF uses that to determine this setting. Otherwise, AWS WAF uses the country of the client IP. This determination is independent of any logic you use in your rules to determine country of origin. AWS WAF determines the locations of the IPs using MaxMind GeoIP databases.
`Attack`	The type of attack that AWS WAF identified in the request, based on the rules and rule groups that you use in your web ACL. Your rules and the rules in the baseline AWS managed rule groups can identify attack types. For example, cross-site scripting (XSS) rule matches identify XSS attack types, and rate-based rules identify volumetric attack types. The attack type usually indicates the type of rule that terminated the web request evaluation.
`Device`	The device type of the client that sent the request, obtained from the web request's `user-agent` header.

Dimension	Description
LoadBalancerArn	The Amazon Resource Name (ARN) of the load balancer.
LoadBalancerArnAvailabilityZone	The combination of the load balancer ARN and the Availability Zone.
ManagedRuleGroup	The metric name of the ManagedRuleGroup .
ManagedRuleGroupRule	The rule within the ManagedRuleGroup that was matched.
VulnerabilityCategory	The vulnerability category that the request matches, based on AWS managed rule IP sets.

Label metrics and dimensions

Metrics for the labels added to requests during evaluation by your rules and by the managed rule groups that you use in your protection pack (web ACL). For information, see Web request labeling.

For any single web request, AWS WAF stores metrics for at most 100 labels. Your protection pack (web ACL) evaluation can apply more than 100 labels and match against more than 100 labels, but only the first 100 are reflected in the metrics.

Label metrics

Metric	Description
AllowedRequests	The number of labels on web requests that had the action setting Allow applied. The labels can have been added at any point during the web request evaluation. Reporting criteria: There is a nonzero value. Valid statistics: Sum
BlockedRequests	The number of labels on web requests that had the action setting Block applied. The labels can have

Metric	Description
	been added at any point during the web request evaluation. Reporting criteria: There is a nonzero value. Valid statistics: Sum
`CountedRequests`	The number of labels added to web requests by rule group rules that have a Count action setting. This metric is only available to the owner of a rule group, for rules inside the rule group. For other cases, the count label metrics are rolled up into the terminating action that was applied to the request, like Allow or Block. Reporting criteria: There is a nonzero value. Valid statistics: Sum
`CaptchaRequests`	The number of labels on web requests that had a terminating CAPTCHA action applied. The labels can have been added at any point during the web request evaluation. Reporting criteria: There is a nonzero value. Valid statistics: Sum
`ChallengeRequests`	The number of labels on web requests that had a terminating Challenge action applied. The labels can have been added at any point during the web request evaluation. Reporting criteria: There is a nonzero value. Valid statistics: Sum

Metric	Description
AllowRuleMatch	The number of matched rules that both generated the associated label and terminated request evaluation with an Allow action. Reporting criteria: There is a nonzero value. Valid statistics: Sum
BlockRuleMatch	The number of matched rules that both generated the associated label and terminated request evaluation with a Block action. Reporting criteria: There is a nonzero value. Valid statistics: Sum
CountRuleMatch	The number of matched rules that both generated the associated label and applied a Count action. One request could result in multiple instances of this metric, if multiple rules are configured with the same label and action. Reporting criteria: There is a nonzero value. Valid statistics: Sum
CaptchaRuleMatch	The number of matched rules that both generated the associated label and terminated request evaluation with a CAPTCHA action. Reporting criteria: There is a nonzero value. Valid statistics: Sum

Metric	Description
ChallengeRuleMatch	The number of matched rules that both generated the associated label and terminated request evaluation with a Challenge action. Reporting criteria: There is a nonzero value. Valid statistics: Sum
CaptchaRuleMatchWithValidToken	The number of matched rules that both generated the associated label and applied a non-terminating CAPTCHA action. One request could result in multiple instances of this metric, if multiple rules are configured with the same label and action. Reporting criteria: There is a nonzero value. Valid statistics: Sum
ChallengeRuleMatchWithValidToken	The number of matched rules that both generated the associated label and applied a non-terminating Challenge action. One request could result in multiple instances of this metric, if multiple rules are configured with the same label and action. Reporting criteria: There is a nonzero value. Valid statistics: Sum

Label dimensions

Dimension	Description
`Region`	Required for all protected resource types except for Amazon CloudFront distributions.
`RuleGroup`	The metric name of the `RuleGroup` . Used for the metric `CountedRequests` .
`WebACL`	The metric name of the `WebACL`.
`ResourceType`	The type of the protected resource, such as `CF`, `APIGW`, or `ALB`.
`Resource`	The Amazon Resource Name (ARN) of the protected resource.
`LabelNamespace`	The namespace prefix of the label that was added to the request.
`Label`	The name of the label that was added to the request.
`Context`	The managed rule group that served as the context of the label addition. For example, the context for token management labels such as `awswaf:managed:token:accepted` is the AWS WAF managed rule group that uses token management on the request, such as the Bot Control or ATP managed rule group. This dimension doesn't apply to all labels.

Free bot visibility metrics and dimensions

When you don't use Bot Control in your protection pack (web ACL), AWS WAF applies the Bot Control managed rule group to a sampling of your web requests, at no additional cost. This can provide an idea of the bot traffic that is coming to your protected resources. For information about Bot Control, see AWS WAF Bot Control rule group.

Free bot visibility metrics

Metric	Description
SampleAllowedRequest	The number of sampled requests that have Allow action. Reporting criteria: There is a nonzero value. Valid statistics: Sum
SampleBlockedRequest	The number of sampled requests that have Block action. Reporting criteria: There is a nonzero value. Valid statistics: Sum
SampleCaptchaRequest	The number of sampled requests that have CAPTCHA action. Reporting criteria: There is a nonzero value. Valid statistics: Sum
SampleChallengeRequest	The number of sampled requests that have Challenge action. Reporting criteria: There is a nonzero value. Valid statistics: Sum
SampleCountRequest	The number of sampled requests that have Count action. Reporting criteria: There is a nonzero value. Valid statistics: Sum

Free bot visibility dimensions

Dimension	Description
`Region`	Required for all protected resource types except for Amazon CloudFront distributions.
`WebACL`	The metric name of the `WebACL`.
`BotCategory`	The name of the of the detected bot category, based on the web request labels.
`VerificationStatus`	The name of the of the detected bot verification status, based on the web request labels.
`Signal`	The name of the of the detected bot signals, based on the web request labels.

Account metrics and dimensions

Account metrics provide account-wide information about CAPTCHA puzzles and silent Challenge rule actions that were serviced through the JavaScript API.

Account metrics

Metric	Description
`CaptchasAttemptedSdk`	The number of solutions that were submitted by an end user in response to a CAPTCHA puzzle challenge, for puzzles that were served via the CAPTCHA JavaScript API. Reporting criteria: There is a nonzero value. Valid statistics: Sum
`CaptchasSolvedSdk`	The number of CAPTCHA puzzle solutions submitted that successfully solved the puzzle, for puzzles that were served via the CAPTCHA JavaScrip t API.

Metric	Description
	Reporting criteria: There is a nonzero value. Valid statistics: Sum
`ChallengesAttemptedSdk`	The number of attempts that were submitted by an end user in response to a silent challenge served through the Challenge JavaScript API. This includes challenges initiated by a Challenge action and challenges run as part of a CAPTCHA action when no token is present. Reporting criteria: There is a nonzero value. Valid statistics: Sum
`ChallengesSolvedSdk`	The number of silent challenge solutions submitted that successfully passed a silent challenge served through the Challenge JavaScript API. This includes challenges initiated by a Challenge action and challenges run as part of a CAPTCHA action when no token is present. Reporting criteria: There is a nonzero value. Valid statistics: Sum

Account dimensions

Dimension	Description
`Region`	Required for all protected resource types except for Amazon CloudFront distributions.

AWS WAF usage metrics

You can use CloudWatch usage metrics to provide visibility into your account's usage of resources. Use these metrics to visualize your current service usage on CloudWatch graphs and dashboards.

AWS WAF usage metrics correspond to AWS service quotas. You can configure alarms that alert you when your usage approaches a service quota. For more information about CloudWatch integration with service quotas, see AWS usage metrics in the *Amazon CloudWatch User Guide.*

AWS WAF publishes the following metrics in the `AWS/Usage` namespace.

Usage metrics

Metric	Description
`ResourceCount`	The number of the specified resources in your account. The resources are defined by the dimensions associated with the metric. The most useful statistic for this metric is `MAXIMUM`, which represents the maximum number of resources used during the 1-minute period.

The following dimension is used to refine the usage metrics that are published by AWS WAF.

Usage dimensions

Dimension	Description
`Resource`	The type of resource for which the usage is being reported.

The following are the supported values for the `Resource` dimension.

`Resource` values

Value	Description
`WebAclsPerAccountCloudFront`	The number of protection packs (web ACLs) the customer has in CloudFront per account. This metric is only available when there is at least one protectio n pack (web ACL) in CloudFront.
`WebAclsPerAccountRegional`	The number of protection packs (web ACLs) the customer has in a region per account. This metric is

Value	Description
	only available when there is at least one protection pack (web ACL) in that region.
`RuleGroupsPerAccountCloudFront`	The number of rule groups the customer has in CloudFront per account. This metric is only available when there is at least one rule group in CloudFront.
`RuleGroupsPerAccountRegional`	The number of rule groups the customer has in a region per account. This metric is only available when there is at least one rule group in that region.
`IpSetsPerAccountCloudFront`	The number of IP sets the customer has in CloudFront per account. This metric is only available when there is at least one IP set in CloudFront.
`IpSetsPerAccountRegional`	The number of IP sets the customer has in a region per account. This metric is only available when there is at least one IP set in that region.
`RegexPatternSetsPerAccountCloudFront`	The number of regex pattern sets the customer has in CloudFront per account. This metric is only available when there is at least one regex pattern set in CloudFront.
`RegexPatternSetsPerAccountRegional`	The number of regex pattern sets the customer has in a region per account. This metric is only available when there is at least one regex pattern set in that region.

AWS Shield Advanced metrics

Shield Advanced publishes Amazon CloudWatch detection, mitigation, and top contributor metrics for all resources that it protects. These metrics improve your ability to monitor your resources by making it possible to create and configure CloudWatch dashboards and alarms for them.

The Shield Advanced console presents summaries of many of the metrics that it records. For information, see Visibility into DDoS events with Shield Advanced.

If you enable automatic application layer DDoS mitigation for an application layer protection, Shield Advanced adds a rule group to your protection pack (web ACL) that it uses to manage automated protections. This rule group generates AWS WAF metrics, but they are not available to view. This is the same as for any other rule groups that you use in your protection pack (web ACL) but do not own, such as AWS Managed Rules rule groups. For more information about AWS WAF metrics, see AWS WAF metrics and dimensions. For information about this Shield Advanced protection option, see Automating application layer DDoS mitigation with Shield Advanced.

Metric reporting locations

Shield Advanced reports metrics in the US East (N. Virginia) Region, `us-east-1` for the following:

- The global services Amazon CloudFront and Amazon Route 53.
- Protection groups. For information about protection groups, see Grouping your AWS Shield Advanced protections.

For other resource types, Shield Advanced reports metrics in the resource's Region.

Timing of metric reporting

Shield Advanced reports metrics to Amazon CloudWatch on an AWS resource more frequently during DDoS events than while no events are underway. Shield Advanced reports metrics once a minute during an event, and then once right after the event ends.

While no events are underway, Shield Advanced reports metrics once a day, at a time assigned to the resource. This periodic report keeps the metrics active and available for use in custom CloudWatch alarms and dashboards.

Alarm recommendations

We recommend that you create alarms to notify you of circumstances that require attention. As a starting point, you could create an alarm for each protected resource that reports when the `DDoSDetected` detection metric is non zero. A non-zero value in this metric doesn't necessarily imply that a DDoS attack is underway, but we recommend looking closer at the resource status when the metric is in this state.

For request floods, we recommend that you create alarms for composite checks that also consider factors such as application health and web request volume. You may choose to alarm on the other 3 metrics that report on the volume of traffic for various attack vector dimensions. By

considering the capacity of your application and alarming when traffic is approaching your application limitations, you can create a set of rules that notify you as needed, without too much unwanted noise.

Topics

- Detection metrics
- Mitigation metrics
- Top contributors metrics

Detection metrics

Shield Advanced provides the metrics and dimensions in the `AWS/DDoSProtection` namespace.

Detection metrics

Metric	Description
`DDoSDetected`	Indicates whether a DDoS event is underway for a particular Amazon Resource Name (ARN). This metric has a non-zero value during an event.
`DDoSAttackBitsPerSecond`	The number of bits observed during a DDoS event for a particular Amazon Resource Name (ARN). This metric is available only for network and transport layer (layer 3 and layer 4) DDoS events. This metric has a non-zero value during an event. Units: Bits
`DDoSAttackPacketsPerSecond`	The number of packets observed during a DDoS event for a particular Amazon Resource Name (ARN). This metric is available only for network and transport layer (layer 3 and layer 4) DDoS events.

Metric	Description
	This metric has a non-zero value during an event. Units: Packets
DDoSAttackRequestsPerSecond	The number of requests observed during a DDoS event for a particular Amazon Resource Name (ARN). This metric is available only for layer 7 DDoS events. The metric is reported only for the most significant layer 7 events. This metric has a non-zero value during an event. Units: Requests
DDoSAttackRequests	The number of requests observed during a DDoS event for a particular Amazon Resource Name (ARN). This metric is available only for Anti-DDoS Managed Rules (AMR) DDoS events. This metic is in the AWS/WAFV2 namespace and has a non-zero value during an event. Units: Requests

Shield Advanced posts the DDoSDetected metric with no other dimensions. The remaining detection metrics include the AttackVector dimensions that correspond to the type of attack, from the following list:

- ACKFlood
- ChargenReflection
- DNSReflection
- AWS/WAFV2
- GenericUDPReflection
- MemcachedReflection

- MSSQLReflection
- NetBIOSReflection
- NTPReflection
- PortMapper
- RequestFlood
- RIPReflection
- SNMPReflection
- SSDPReflection
- SYNFlood
- UDPFragment
- UDPTraffic
- UDPReflection

Mitigation metrics

Shield Advanced provides metrics and dimensions in the AWS/DDoSProtection namespace.

Mitigation metrics

Metric	Description
VolumePacketsPerSecond	The number of packets per second that were dropped or passed by a mitigation that was deployed in response to a detected event. Units: packets

Mitigation dimensions

Dimension	Description
ResourceArn	Amazon Resource Name (ARN)
MitigationAction	The outcome of an applied mitigation. Possible values are Pass or Drop.

Top contributors metrics

Shield Advanced provides metrics in the `AWS/DDoSProtection` namespace.

Top contributors metrics

Metric	Description
`VolumePacketsPerSecond`	The number of packets per second for a top contributor. Units: packets
`VolumeBitsPerSecond`	The number of bits per second for a top contributor. Units: bits

Shield Advanced posts top contributors metrics by dimension combinations that characterize the event contributors. You can use any of the following combinations of dimensions for any of the top contributors metrics:

- `ResourceArn`, `Protocol`
- `ResourceArn`, `Protocol`, `SourcePort`
- `ResourceArn`, `Protocol`, `DestinationPort`
- `ResourceArn`, `Protocol`, `SourceIp`
- `ResourceArn`, `Protocol`, `SourceAsn`
- `ResourceArn`, `TcpFlags`

Top contributors dimensions

Dimension	Description
`ResourceArn`	Amazon Resource Name (ARN).
`Protocol`	IP protocol name, either TCP or UDP.
`SourcePort`	Source TCP or UDP port.

Dimension	Description
DestinationPort	Destination TCP or UDP port.
SourceIp	Source IP address.
SourceAsn	Source autonomous system number (ASN).
TcpFlags	A combination of flags present in a TCP packet, separated by a dash (-). Monitored flags are ACK, FIN, RST, SYN. This dimension value always appears sorted alphabetically. For example, ACK-FIN-RST-SYN , ACK-SYN, and FIN-RST.

AWS Firewall Manager notifications

AWS Firewall Manager doesn't record metrics, so you can't create Amazon CloudWatch alarms specifically for Firewall Manager. However, you can configure Amazon SNS notifications to alert you to potential attacks. To create Amazon SNS notifications in Firewall Manager, see Step 4: Configuring Amazon SNS notifications and Amazon CloudWatch alarms.

Logging API calls with AWS CloudTrail

AWS WAF, AWS Shield Advanced, and AWS Firewall Manager are integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures a subset of API calls for these services as events, including calls from the AWS WAF, Shield Advanced or Firewall Manager consoles and from code calls to the AWS WAF, Shield Advanced, or Firewall Manager APIs. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS WAF, Shield Advanced, or Firewall Manager. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to these services, the IP address that the request was made from, who made the request, when it was made, and additional details.

To learn more about CloudTrail, including how to configure and enable it, see the AWS CloudTrail User Guide.

CloudTrail is enabled on your AWS account when you create the account. When supported event activity occurs in AWS WAF, Shield Advanced, or Firewall Manager, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for AWS WAF, Shield Advanced, or Firewall Manager, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail on the console, the trail applies to all Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

AWS WAF information in AWS CloudTrail

All AWS WAF actions are logged by AWS CloudTrail and are documented in the AWS WAF API Reference. For example, calls to `ListWebACL`, `UpdateWebACL`, and `DeleteWebACL` generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root user credentials
- Whether the request was made with temporary security credentials for a role or federated user
- Whether the request was made by another AWS service

For more information, see CloudTrail userIdentity Element.

Cross-account CloudTrail events

For certain AWS WAF API operations that support cross-account access through resource-based policies, CloudTrail events are logged to both the caller account and the resource owner account. This applies to the following API operations:

- `CreateWebACL`
- `UpdateWebACL`
- `GetWebACL`
- `GetRuleGroup`
- `PutFirewallManagerRuleGroups`

When these operations are performed across accounts, CloudTrail events are generated for both successful API calls and access denied calls in both the caller's account and the resource owner's account.

Example: AWS WAF log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. AWS CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

The following are examples of CloudTrail log entries for AWS WAF protection pack (web ACL) operations.

Example: CloudTrail log entry for `CreateWebACL`


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "principalId",
    "arn": "arn:aws:sts::112233445566:assumed-role/Admin",
    "accountId": "112233445566",
    "accessKeyId": "accessKeyId",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
```

```
        "principalId": "principalId",
        "arn": "arn:aws:iam::112233445566:role/Admin",
        "accountId": "112233445566",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2019-11-06T03:43:07Z"
      }
    }
  },
  "eventTime": "2019-11-06T03:44:21Z",
  "eventSource": "wafv2.amazonaws.com",
  "eventName": "CreateWebACL",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "10.0.0.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/78.0.3904.87 Safari/537.36",
  "requestParameters": {
    "name": "foo",
    "scope": "CLOUDFRONT",
    "defaultAction": {
      "block": {}
    },
    "description": "foo",
    "rules": [
      {
        "name": "foo",
        "priority": 1,
        "statement": {
          "geoMatchStatement": {
            "countryCodes": [
              "AF",
              "AF"
            ]
          }
        },
        "action": {
          "block": {}
        },
        "visibilityConfig": {
          "sampledRequestsEnabled": true,
          "cloudWatchMetricsEnabled": true,
```

```
          "metricName": "foo"
        }
      }
    ],
    "visibilityConfig": {
      "sampledRequestsEnabled": true,
      "cloudWatchMetricsEnabled": true,
      "metricName": "foo"
    }
  },
  "responseElements": {
    "summary": {
      "name": "foo",
      "id": "ebbcb976-8d59-4d20-8ca8-4ab2f6b7c07b",
      "description": "foo",
      "lockToken": "67551e73-49d8-4363-be48-244deea72ea9",
      "aRN": "arn:aws:wafv2:us-east-1:112233445566:global/webacl/foo/
ebbcb976-8d59-4d20-8ca8-4ab2f6b7c07b"
    }
  },
  "requestID": "c51521ba-3911-45ca-ba77-43aba50471ca",
  "eventID": "afd1a60a-7d84-417f-bc9c-7116cf029065",
  "eventType": "AwsApiCall",
  "apiVersion": "2019-04-23",
  "recipientAccountId": "112233445566"
}
```

Example: CloudTrail log entry for `GetWebACL`

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AssumedRole",
    "arn": "arn:aws:sts::112233445566:assumed-role/Admin/admin",
    "accountId": "112233445566",
    "accessKeyId": "accessKeyId",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AssumedRole",
        "arn": "arn:aws:iam::112233445566:role/Admin",
        "accountId": "112233445566",
```

```
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2019-11-06T19:17:20Z"
      }
    }
  },
  "eventTime": "2019-11-06T19:18:28Z",
  "eventSource": "wafv2.amazonaws.com",
  "eventName": "GetWebACL",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "10.0.0.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/78.0.3904.87 Safari/537.36",
  "requestParameters": {
    "name": "foo",
    "scope": "CLOUDFRONT",
    "id": "webacl"
  },
  "responseElements": null,
  "requestID": "f2db4884-4eeb-490c-afe7-67cbb494ce3b",
  "eventID": "7d563cd6-4123-4082-8880-c2d1fda4d90b",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "apiVersion": "2019-04-23",
  "recipientAccountId": "112233445566"
}
```


Example: CloudTrail log entry for UpdateWebACL


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "principalId",
    "arn": "arn:aws:sts::112233445566:assumed-role/Admin",
    "accountId": "112233445566",
    "accessKeyId": "accessKeyId",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
```

```
        "principalId": "principalId",
        "arn": "arn:aws:iam::112233445566:role/Admin",
        "accountId": "112233445566",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2019-11-06T19:17:20Z"
      }
    }
  },
  "eventTime": "2019-11-06T19:20:56Z",
  "eventSource": "wafv2.amazonaws.com",
  "eventName": "UpdateWebACL",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "10.0.0.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/78.0.3904.87 Safari/537.36",
  "requestParameters": {
    "name": "foo",
    "scope": "CLOUDFRONT",
    "id": "ebbcb976-8d59-4d20-8ca8-4ab2f6b7c07b",
    "defaultAction": {
      "block": {}
    },
    "description": "foo",
    "rules": [
      {
        "name": "foo",
        "priority": 1,
        "statement": {
          "geoMatchStatement": {
            "countryCodes": [
              "AF"
            ]
          }
        },
        "action": {
          "block": {}
        },
        "visibilityConfig": {
          "sampledRequestsEnabled": true,
          "cloudWatchMetricsEnabled": true,
```

```
          "metricName": "foo"
        }
      }
    ],
    "visibilityConfig": {
      "sampledRequestsEnabled": true,
      "cloudWatchMetricsEnabled": true,
      "metricName": "foo"
    },
    "lockToken": "67551e73-49d8-4363-be48-244deea72ea9"
  },
  "responseElements": {
    "nextLockToken": "a6b54c01-7975-4e6d-b7d0-2653cb6e231d"
  },
  "requestID": "41c96e12-9790-46ab-b145-a230f358f2c2",
  "eventID": "517a10e6-4ca9-4828-af90-a5cff9756594",
  "eventType": "AwsApiCall",
  "apiVersion": "2019-04-23",
  "recipientAccountId": "112233445566"
}
```

Example: CloudTrail log entry for `DeleteWebACL`

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "principalId",
    "arn": "arn:aws:sts::112233445566:assumed-role/Admin/session-name",
    "accountId": "112233445566",
    "accessKeyId": "accessKeyId",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "principalId",
        "arn": "arn:aws:iam::112233445566:role/Admin",
        "accountId": "112233445566",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2019-11-06T19:17:20Z"
```

```
      }
    }
  },
  "eventTime": "2019-11-06T19:25:17Z",
  "eventSource": "wafv2.amazonaws.com",
  "eventName": "DeleteWebACL",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "10.0.0.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/78.0.3904.87 Safari/537.36",
  "requestParameters": {
    "name": "foo",
    "scope": "CLOUDFRONT",
    "id": "ebbcb976-8d59-4d20-8ca8-4ab2f6b7c07b",
    "lockToken": "a6b54c01-7975-4e6d-b7d0-2653cb6e231d"
  },
  "responseElements": null,
  "requestID": "71703f89-e139-440c-96d4-9c77f4cd7565",
  "eventID": "2f976624-b6a5-4a09-a8d0-aa3e9f4e5187",
  "eventType": "AwsApiCall",
  "apiVersion": "2019-04-23",
  "recipientAccountId": "112233445566"
}
```

Example: AWS WAF classic log file entries

AWS WAF Classic is the prior version of AWS WAF. For information, see AWS WAF Classic.

The log entry demonstrates the `CreateRule`, `GetRule`, `UpdateRule`, and `DeleteRule` operations:

```
{
  "Records": [
   {
      "eventVersion": "1.03",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAIEP4IT4TPDEXAMPLE",
        "arn": "arn:aws:iam::777777777777:user/nate",
        "accountId": "777777777777",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "nate"
      },
```

```
      "eventTime": "2016-04-25T21:35:14Z",
      "eventSource": "waf.amazonaws.com",
      "eventName": "CreateRule",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "AWS Internal",
      "userAgent": "console.amazonaws.com",
      "requestParameters": {
        "name": "0923ab32-7229-49f0-a0e3-66c81example",
        "changeToken": "l9434322-8685-4ed2-9c5b-9410bexample",
        "metricName": "0923ab32722949f0a0e366c81example"
      },
      "responseElements": {
        "rule": {
          "metricName": "0923ab32722949f0a0e366c81example",
          "ruleId": "12132e64-6750-4725-b714-e7544example",
          "predicates": [

          ],
          "name": "0923ab32-7229-49f0-a0e3-66c81example"
        },
        "changeToken": "l9434322-8685-4ed2-9c5b-9410bexample"
      },
      "requestID": "4e6b66f9-d548-11e3-a8a9-73e33example",
      "eventID": "923f4321-d378-4619-9b72-4605bexample",
      "eventType": "AwsApiCall",
      "apiVersion": "2015-08-24",
      "recipientAccountId": "777777777777"
    },
    {
      "eventVersion": "1.03",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAIEP4IT4TPDEXAMPLE",
        "arn": "arn:aws:iam::777777777777:user/nate",
        "accountId": "777777777777",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "nate"
      },
      "eventTime": "2016-04-25T21:35:22Z",
      "eventSource": "waf.amazonaws.com",
      "eventName": "GetRule",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "AWS Internal",
      "userAgent": "console.amazonaws.com",
```

```
    "requestParameters": {
      "ruleId": "723c2943-82dc-4bc1-a29b-c7d73example"
    },
    "responseElements": null,
    "requestID": "8e4f3211-d548-11e3-a8a9-73e33example",
    "eventID": "an236542-d1f9-4639-bb3d-8d2bbexample",
    "eventType": "AwsApiCall",
    "apiVersion": "2015-08-24",
    "recipientAccountId": "777777777777"
  },
  {
    "eventVersion": "1.03",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "AIDAIEP4IT4TPDEXAMPLE",
      "arn": "arn:aws:iam::777777777777:user/nate",
      "accountId": "777777777777",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "userName": "nate"
    },
    "eventTime": "2016-04-25T21:35:13Z",
    "eventSource": "waf.amazonaws.com",
    "eventName": "UpdateRule",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
      "ruleId": "7237b123-7903-4d9e-8176-9d71dexample",
      "changeToken": "32343a11-35e2-4dab-81d8-6d408example",
      "updates": [
        {
          "predicate": {
            "type": "SizeConstraint",
            "dataId": "9239c032-bbbe-4b80-909b-782c0example",
            "negated": false
          },
          "action": "INSERT"
        }
      ]
    },
    "responseElements": {
      "changeToken": "32343a11-35e2-4dab-81d8-6d408example"
    },
    "requestID": "11918283-0b2d-11e6-9ccc-f9921example",
```

```
      "eventID": "00032abc-5bce-4237-a8ee-5f1a9example",
      "eventType": "AwsApiCall",
      "apiVersion": "2015-08-24",
      "recipientAccountId": "777777777777"
    },
    {
      "eventVersion": "1.03",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAIEP4IT4TPDEXAMPLE",
        "arn": "arn:aws:iam::777777777777:user/nate",
        "accountId": "777777777777",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "nate"
      },
      "eventTime": "2016-04-25T21:35:28Z",
      "eventSource": "waf.amazonaws.com",
      "eventName": "DeleteRule",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "AWS Internal",
      "userAgent": "console.amazonaws.com",
      "requestParameters": {
        "changeToken": "fd232003-62de-4ea3-853d-52932example",
        "ruleId": "3e3e2d11-fd8b-4333-8b03-1da95example"
      },
      "responseElements": {
        "changeToken": "fd232003-62de-4ea3-853d-52932example"
      },
      "requestID": "b23458a1-0b2d-11e6-9ccc-f9928example",
      "eventID": "a3236565-1a1a-4475-978e-81c12example",
      "eventType": "AwsApiCall",
      "apiVersion": "2015-08-24",
      "recipientAccountId": "777777777777"
    }
  ]
}
```


AWS Shield Advanced information in CloudTrail

AWS Shield Advanced supports logging the following actions as events in CloudTrail log files:

- ListAttacks
- DescribeAttack

- CreateProtection
- DescribeProtection
- DeleteProtection
- ListProtections
- CreateSubscription
- DescribeSubscription
- GetSubscriptionState

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root user credentials
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity Element.

Example: Shield Advanced log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `DeleteProtection` and `ListProtections` actions.


```
[
 {
   "eventVersion": "1.05",
   "userIdentity": {
     "type": "IAMUser",
     "principalId": "123456789098765432123 1",
     "arn": "arn:aws:iam::123456789012:user/SampleUser",
     "accountId": "123456789012",
```

```
      "accessKeyId": "1AFGDT647FHU83JHFI81H",
      "userName": "SampleUser"
    },
    "eventTime": "2018-01-10T21:31:14Z",
    "eventSource": "shield.amazonaws.com",
    "eventName": "DeleteProtection",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "aws-cli/1.14.10 Python/3.6.4 Darwin/16.7.0 botocore/1.8.14",
    "requestParameters": {
      "protectionId": "12345678-5104-46eb-bd03-agh4j8rh3b6n"
    },
    "responseElements": null,
    "requestID": "95bc0042-f64d-11e7-abd1-1babdc7aa857",
    "eventID": "85263bf4-17h4-43bb-b405-fh84jhd8urhg",
    "eventType": "AwsApiCall",
    "apiVersion": "AWSShield_20160616",
    "recipientAccountId": "123456789012"
  },
  {
    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "123456789098765432123",
      "arn": "arn:aws:iam::123456789012:user/SampleUser",
      "accountId": "123456789012",
      "accessKeyId": "1AFGDT647FHU83JHFI81H",
      "userName": "SampleUser"
    },
    "eventTime": "2018-01-10T21:30:03Z",
    "eventSource": "shield.amazonaws.com",
    "eventName": "ListProtections",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "aws-cli/1.14.10 Python/3.6.4 Darwin/16.7.0 botocore/1.8.14",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "6accca40-f64d-11e7-abd1-1bjfi8urhj47",
    "eventID": "ac0570bd-8dbc-41ac-a2c2-987j90j3h78f",
    "eventType": "AwsApiCall",
    "apiVersion": "AWSShield_20160616",
    "recipientAccountId": "123456789012"
  }
```

```
]
```

AWS Firewall Manager information in CloudTrail

AWS Firewall Manager supports logging the following actions as events in CloudTrail log files:

- AssociateAdminAccount
- DeleteNotificationChannel
- DeletePolicy
- DisassociateAdminAccount
- PutNotificationChannel
- PutPolicy
- GetAdminAccount
- GetComplianceDetail
- GetNotificationChannel
- GetPolicy
- ListComplianceStatus
- ListPolicies

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root user credentials
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity Element.

Example: Firewall Manager log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of

the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `GetAdminAccount-->` action.


```
 {
                "eventVersion": "1.05",
                "userIdentity": {
                                "type": "AssumedRole",
                                "principalId": "123456789098765432123 1",
                                "arn": "arn:aws:sts::123456789012:assumed-role/Admin/
SampleUser",
                                "accountId": "123456789012",
                                "accessKeyId": "1AFGDT647FHU83JHFI81H",
                                "sessionContext": {
                                                "attributes": {
                                                                "mfaAuthenticated":
 "false",
                                                                "creationDate":
 "2018-04-14T02:51:50Z"
                                                               },
                                                "sessionIssuer": {
                                                                "type": "Role",
                                                                "principalId":
 "123456789098765432123 1",
                                                                "arn":
 "arn:aws:iam::123456789012:role/Admin",
                                                                "accountId":
 "123456789012",
                                                                "userName": "Admin"
                                                                 }
                                                  }
                                },
                "eventTime": "2018-04-14T03:12:35Z",
                "eventSource": "fms.amazonaws.com",
                "eventName": "GetAdminAccount",
                "awsRegion": "us-east-1",
                "sourceIPAddress": "72.21.198.65",
                "userAgent": "console.amazonaws.com",
                "requestParameters": null,
                "responseElements": null,
```

```
            "requestID": "ae244f41-3f91-11e8-787b-dfaafef95fc1",
            "eventID": "5769af1e-14b1-4bd1-ba75-f023981d0a4a",
            "eventType": "AwsApiCall",
            "apiVersion": "2018-01-01",
            "recipientAccountId": "123456789012"
    }
```

Using the AWS WAF and AWS Shield Advanced API

This section describes how to make requests to the AWS WAF and Shield Advanced API for creating and managing match sets, rules, and protection packs (web ACLs) in AWS WAF as well as your subscription and protections in Shield Advanced. This section will acquaint you with the components of requests, the content of responses, and how to authenticate requests.

Topics

- Using the AWS SDKs
- Making HTTPS requests to AWS WAF or Shield Advanced
- HTTP responses
- Authenticating requests

Using the AWS SDKs

If you use a language that AWS provides an SDK for, use the SDK rather than trying to work your way through the APIs. The SDKs make authentication simpler, integrate easily with your development environment, and provide easy access to AWS WAF and Shield Advanced commands. For more information about the AWS SDKs, see Download tools in the topic Setting up your account to use the services.

Making HTTPS requests to AWS WAF or Shield Advanced

AWS WAF and Shield Advanced requests are HTTPS requests, as defined by RFC 2616. Like any HTTP request, a request to AWS WAF or Shield Advanced contains a request method, a URI, request headers, and a request body. The response contains an HTTP status code, response headers, and sometimes a response body.

Request URI

The request URI is always a single forward slash, /.

HTTP headers

AWS WAF and Shield Advanced require the following information in the header of an HTTP request:

Host (Required)

The endpoint that specifies where your resources are created. For information about endpoints, see AWS service endpoints. For example, the value of the `Host` header for AWS WAF for a CloudFront distribution is `waf.amazonaws.com:443`.

x-amz-date or Date (Required)

The date used to create the signature that is contained in the `Authorization` header. Specify the date in ISO 8601 standard format, in UTC time, as shown in the following example:

```
x-amz-date: 20151007T174952Z
```

You must include either `x-amz-date` or `Date`. (Some HTTP client libraries don't let you set the `Date` header). When an `x-amz-date` header is present, AWS WAF ignores any `Date` header when authenticating the request.

The time stamp must be within 15 minutes of the AWS system time when the request is received. If it isn't, the request fails with the `RequestExpired` error code to prevent someone else from replaying your requests.

Authorization (Required)

The information required for request authentication. For more information about constructing this header, see Authenticating requests.

X-Amz-Target (Required)

A concatenation of `AWSWAF_` or `AWSShield_`, the API version without punctuation, a period (`.`), and the name of the operation, for example:

`AWSWAF_20150824.CreateWebACL`

Content-Type (Conditional)

Specifies that the content type is JSON as well as the version of JSON, as shown in the following example:

```
Content-Type: application/x-amz-json-1.1
```

Condition: Required for POST requests.

Content-Length (Conditional)

Length of the message (without the headers) according to RFC 2616.

Condition: Required if the request body itself contains information (most toolkits add this header automatically).

The following is an example header for an HTTP request to create a protection pack (web ACL) in AWS WAF:

```
POST / HTTP/1.1
Host: waf.amazonaws.com:443
X-Amz-Date: 20151007T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/20151007/us-east-2/waf/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,

 Signature=145b1567ab3c50d929412f28f52c45dbf1e63ec5c66023d232a539a4afd11fd9
X-Amz-Target: AWSWAF_20150824.CreateWebACL
Accept: */*
Content-Type: application/x-amz-json-1.1; charset=UTF-8
Content-Length: 231
Connection: Keep-Alive
```

HTTP request body

Many AWS WAF and Shield Advanced API actions require you to include JSON-formatted data in the body of the request.

The following example request uses a simple JSON statement to update an `IPSet` to include the IP address 192.0.2.44 (represented in CIDR notation as 192.0.2.44/32):

```
POST / HTTP/1.1
Host: waf.amazonaws.com:443
X-Amz-Date: 20151007T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/20151007/us-east-2/waf/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,

 Signature=145b1567ab3c50d929412f28f52c45dbf1e63ec5c66023d232a539a4afd11fd9
X-Amz-Target: AWSWAF_20150824.UpdateIPSet
```

```
Accept: */*
Content-Type: application/x-amz-json-1.1; charset=UTF-8
Content-Length: 283
Connection: Keep-Alive

{
   "ChangeToken": "d4c4f53b-9c7e-47ce-9140-0ee5ffffffff",
   "IPSetId": "69d4d072-170c-463d-ab82-0643ffffffff",
   "Updates": [
      {
         "Action": "INSERT",
         "IPSetDescriptor": {
            "Type": "IPV4",
            "Value": "192.0.2.44/32"
         }
      }
   ]
}
```

HTTP responses

All AWS WAF and Shield Advanced API actions include JSON-formatted data in the response.

Here are some important headers in the HTTP response and how you should handle them in your application, if applicable:

HTTP/1.1

This header is followed by a status code. Status code 200 indicates a successful operation.

Type: String

x-amzn-RequestId

A value created by AWS WAF or Shield Advanced that uniquely identifies your request, for example, K2QH8DN0U907N97FNA2GDLL80BVV4KQNS05AEMVJF66Q9ASUAAJG. If you have a problem with AWS WAF, AWS can use this value to troubleshoot the problem.

Type: String

Content-Length

The length of the response body in bytes.

Type: String

Date

The date and time that AWS WAF or Shield Advanced responded, for example, Wed, 07 Oct 2015 12:00:00 GMT.

Type: String

Error responses

If a request results in an error, the HTTP response contains the following values:

- A JSON error document as the response body
- Content-Type
- The applicable 3xx, 4xx, or 5xx HTTP status code

The following is an example of a JSON error document:

```
HTTP/1.1 400 Bad Request
x-amzn-RequestId: b0e91dc8-3807-11e2-83c6-5912bf8ad066
x-amzn-ErrorType: ValidationException
Content-Type: application/json
Content-Length: 125
Date: Mon, 26 Nov 2012 20:27:25 GMT

{"message":"1 validation error detected: Value null at 'TargetString' failed to satisfy
 constraint: Member must not be null"}
```

Authenticating requests

If you use a language that AWS provides an SDK for, we recommend that you use the SDK. All the AWS SDKs greatly simplify the process of signing requests and save you a significant amount of time when compared with using the AWS WAF or Shield Advanced API. In addition, the SDKs integrate easily with your development environment and provide easy access to related commands.

AWS WAF and Shield Advanced require that you authenticate every request that you send by signing the request. To sign a request, you calculate a digital signature using a cryptographic hash function, which returns a hash value based on the input. The input includes the text of your request

and your secret access key. The hash function returns a hash value that you include in the request as your signature. The signature is part of the `Authorization` header of your request.

After receiving your request, AWS WAF or Shield Advanced recalculates the signature using the same hash function and input that you used to sign the request. If the resulting signature matches the signature in the request, AWS WAF or Shield Advanced processes the request. If not, the request is rejected.

AWS WAF and Shield Advanced supports authentication using AWS Signature Version 4. The process for calculating a signature can be broken into three tasks:

Task 1: Create a Canonical Request

Create your HTTP request in canonical format as described in Task 1: Create a Canonical Request For Signature Version 4 in the *Amazon Web Services General Reference*.

Task 2: Create a String to Sign

Create a string that you will use as one of the input values to your cryptographic hash function. The string, called the string to sign, is a concatenation of the following values:

- Name of the hash algorithm
- Request date
- Credential scope string
- Canonicalized request from the previous task

The credential scope string itself is a concatenation of date, region, and service information.

For the `X-Amz-Credential` parameter, specify the following:

- The code for the endpoint to which you're sending the request, `us-east-2`
- `waf` for the service abbreviation

For example:

```
X-Amz-Credential=AKIAIOSFODNN7EXAMPLE/20130501/us-east-2/waf/
aws4_request
```

Task 3: Create a Signature

Create a signature for your request by using a cryptographic hash function that accepts two input strings:

- Your string to sign, from Task 2.
- A derived key. The derived key is calculated by starting with your secret access key and using the credential scope string to create a series of hash-based message authentication codes (HMACs).

Related information

The following related resources can help you as you work with this service.

The following resources are available for AWS WAF, AWS Shield Advanced, and AWS Firewall Manager.

- **Guidelines for Implementing AWS WAF** – Technical publication with current recommendations for implementing AWS WAF to protect existing and new web applications.
- **AWS discussion forums** – A community-based forum for discussing technical questions related to this and other AWS services.
- **AWS WAF Discussion Forum** – A community-based forum for developers to discuss technical questions related to AWS WAF.
- **Shield Advanced Discussion Forum** – A community-based forum for developers to discuss technical questions related to Shield Advanced.
- **AWS WAF product information** – The primary web page for information about AWS WAF, including features, pricing, and more.
- **Shield Advanced product information** – The primary web page for information about Shield Advanced, including features, pricing, and more.

The following resources are available for Amazon Web Services.

- Classes & Workshops – Links to role-based and specialty courses, in addition to self-paced labs to help sharpen your AWS skills and gain practical experience.
- AWS Developer Center – Explore tutorials, download tools, and learn about AWS developer events.
- AWS Developer Tools – Links to developer tools, SDKs, IDE toolkits, and command line tools for developing and managing AWS applications.
- Getting Started Resource Center – Learn how to set up your AWS account, join the AWS community, and launch your first application.
- Hands-On Tutorials – Follow step-by-step tutorials to launch your first application on AWS.
- AWS Whitepapers – Links to a comprehensive list of technical AWS whitepapers, covering topics such as architecture, security, and economics and authored by AWS Solutions Architects or other technical experts.

- AWS Support Center – The hub for creating and managing your AWS Support cases. Also includes links to other helpful resources, such as forums, technical FAQs, service health status, and AWS Trusted Advisor.
- Support – The primary webpage for information about Support, a one-on-one, fast-response support channel to help you build and run applications in the cloud.
- Contact Us – A central contact point for inquiries concerning AWS billing, account, events, abuse, and other issues.
- AWS Site Terms – Detailed information about our copyright and trademark; your account, license, and site access; and other topics.

FAQs on supported protocols

The AWS Firewall Manager service is adding SDK support for the Smithy RPC v2 CBOR protocol in addition to the existing AWS JSON protocol. The Smithy RPC v2 CBOR protocol is more performant than AWS JSON for most workloads. AWS SDKs will prioritize the protocol that is the most performant for each language.

What are the AWS JSON and Smithy RPC v2 CBOR protocols, and how do they differ?

A wire protocol acts as a mediator between a client and the service. The AWS SDK handles the serialization (converting an object to the wire format) and deserialization (converting the wire format back to an object) between the client and server transparently for both protocols.

AWS JSON - JSON is one of the most widely used and accepted data formats for communication between systems. This service uses AWS JSON as a medium to communicate between an AWS SDK client and the server. An HTTP request for a service API operation accepts input in the form of JSON, the operation is executed, and the response is returned to the SDK client in the form of JSON.

Smithy RPC v2 CBOR - Smithy RPC v2 CBOR (Concise Binary Object Representation) is a protocol developed by Amazon for more efficient data transmission. It is a data format designed to keep message sizes compact and to allow the format to evolve without requiring version negotiation between systems. The key difference from AWS JSON is that AWS JSON uses human-readable text (like `{"name": "John"}`), while Smithy RPC v2 CBOR uses a binary format (sequences of 1s and 0s). This binary approach creates smaller data packages that travel faster over the network and reduces serialization time since computers process binary data directly.

How do I get started with the Smithy RPC v2 CBOR protocol?

To get started, use the latest AWS SDK/CLI version to use the Smithy RPC v2 CBOR protocol for this service. The SDK will automatically use the Smithy RPC v2 CBOR protocol when available. Upgrade your AWS SDK to the specified version or any subsequent version.

Starting 2026-10-06, the following AWS SDKs will use the Smithy RPC v2 CBOR protocol, while the other SDKs will continue using the AWS JSON protocol:

- AWS SDK for C++
- AWS SDK for Go V2
- AWS SDK for Java 2.x
- AWS SDK for .NET v4
- AWS Tools for PowerShell v5
- AWS SDK for Swift
- AWS SDK for Kotlin

What are the risks of enabling the Smithy RPC v2 CBOR protocol?

The added support of Smithy RPC v2 CBOR in the SDKs is expected to be fully transparent and effortless for customers. However, if you are using a custom implementation of the AWS SDK, or a combination of custom clients and the AWS SDK, that depends on AWS JSON-specific serialization behavior, it may be incompatible with the Smithy RPC v2 CBOR protocol.

What if my solution does not support Smithy RPC v2 CBOR?

To use the AWS JSON protocol, downgrade your SDK to a version released before 2026-10-06.

What AWS Regions support the Smithy RPC v2 CBOR protocol?

The Smithy RPC v2 CBOR protocol is supported in all AWS Regions where this service is available.

What latency improvements can I expect when upgrading to SDK versions that support Smithy RPC v2 CBOR?

Smithy RPC v2 CBOR provides up to 13% lower latency and up to 15% reduction in payload size compared to AWS JSON. The binary encoding eliminates the overhead of JSON text parsing and string escaping, which reduces both network transfer time and serialization cost.

Will the AWS JSON protocol be deprecated?

AWS JSON protocol will continue to be supported through older versions of SDKs. Also, some language-specific SDKs will continue using AWS JSON as it is more performant than Smithy RPC

v2 CBOR due to the available JSON implementation for the language being faster than its CBOR implementation.

Where can I find more information about Smithy RPC v2 CBOR?

You can find more information about the AWS JSON protocol at AWS JSON 1.0 protocol and AWS JSON 1.1 protocol in the Smithy documentation, and about Smithy RPC v2 CBOR at Smithy RPC v2 CBOR protocol in the Smithy documentation.

Document history

This page lists significant changes to this documentation.

Service features are sometimes rolled out incrementally to the AWS Regions where a service is available. We update this documentation for the first release only. We don't provide information about Region availability or announce subsequent Region rollouts. For information about Region availability of service features and to subscribe to notifications about updates, see What's New with AWS?.

Change	Description	Date
Added FAQ page for supported protocols	Added a new FAQ page with answers to common questions about the AWS JSON and Smithy RPC v2 CBOR wire protocols. For more information, see FAQs on supported protocols.	October 6, 2026
Updated AWS Managed Rules for AWS WAF	Released static version 2.5 of the SQL database managed rule group.	September 16, 2026
Updated AWS Managed Rules for AWS WAF	Released static version 1.23 of the core rule set (CRS) managed rule group.	August 28, 2026
Updated AWS Managed Rules for AWS WAF	Released static version 2.4 of the SQL database managed rule group.	August 12, 2026
AWS WAF pre-parse text transformations	Added pre-parse text transformations that normalize raw query strings before AWS WAF parses	July 29, 2026

	them into individual query arguments.	
AWS WAF new text transformations and pre-parse text transformations	Added 10 new text transformations and a new pre-parse text transformations feature that normalizes raw query strings before parsing.	July 29, 2026
Updated AWS Managed Rules for AWS WAF	Released static version 6.1 of the Bot Control rule group with expanded bot detection across additional bot categories.	July 24, 2026
Updated AWS Managed Rules for AWS WAF	Released static version 1.22 of the core rule set (CRS) managed rule group.	July 20, 2026
Updated AWS Managed Rules for AWS WAF	Released static version 1.26 of the Known bad inputs managed rule group.	July 15, 2026
AI traffic monetization	Added documentation for AI traffic monetization, including : How AI traffic monetization works, Getting started with AI traffic monetization, Pricing configuration, Payment network and settlemen t, Revenue analytics, and Communicating license terms to AI agents. Minor updates to Data protection and logging.	June 15, 2026
Updated AWS Managed Rules for AWS WAF	Released static version 2.5 of the Windows operating system managed rule group.	June 1, 2026

HTTP/2 body inspection control for Application Load Balancer	You can now configure how AWS WAF inspects HTTP/2 request bodies on your Application Load Balancer when routing to HTTP/2 targets.	June 1, 2026
Updated AWS Managed Rules for AWS WAF	Released static version 6.0 of the Bot Control rule group with Web Bot Auth support to commercial partition regional AWS WAF integrations and expanded bot detection across additional bot categories.	May 22, 2026
Resource associations per web ACL	Add new quotas for Amazon API Gateway REST API, Amazon Cognito user pool, AWS AppSync GraphQL API, AWS App Runner service, AWS Verified Access instance, and AWS Amplify associations per web ACL. The default quota per web ACL is 100 for each.	May 1, 2026
Updated AWS Managed Rules for AWS WAF	Released static version 2.3 of the SQL database managed rule group.	April 29, 2026
Updated AWS Managed Rules for AWS WAF	Released static version 2.4 of the Windows operating system managed rule group.	April 28, 2026

AWS WAF API permission updates for Application Load Balancer and AppSync	Updated guide to align with the IAM best practices . For more information, see Security best practices in IAM.	April 8, 2026
Updated AWS Managed Rules for AWS WAF	Released static version 1.21 of the core rule set (CRS) managed rule group.	April 6, 2026
Dynamic label interpolation	You can now use `${namespace:}` syntax to dynamically resolve label values in custom request headers, custom response headers, and custom response bodies.	April 1, 2026
Updated AWS Managed Rules for AWS WAF	Released static version 2.2 of the PHP Application rule group.	March 24, 2026
Updated AWS Managed Rules for AWS WAF	Released static version 5.0 of the Bot Control rule group with expanded bot detection capabilities.	February 25, 2026
AWS WAF console has a new AI Traffic Analysis dashboard	The web ACL page in the AWS WAF console has a new AI Traffic Analysis dashboard.	February 25, 2026
Updated AWS Shield network security director network topology widget documentation	Updated AWS Shield network security director network topology widget documentation to reflect how it is generated and displayed.	February 9, 2026
Updated AWS Managed Rules for AWS WAF	Released static version of 3.2 of the POSIX operating system rule group.	January 15, 2026

Updated AWS Shield network security director public preview	Updated AWS Shield network security director public preview to include multi-account features and integration with AWS Organizations.	December 12, 2025
Updated AWS Managed Rules for AWS WAF	Released static version 1.25 of the Known bad inputs rule group.	December 8, 2025
Updated AWS Managed Rules for AWS WAF	Released static version of 3.1 of the POSIX operating system rule group.	December 8, 2025
Updated AWS Managed Rules for AWS WAF	Released static version 1.24 of the Known bad inputs rule group.	December 4, 2025
Updated AWS Managed Rules for AWS WAF	Updated Bot Control rule group with support for expanded bot detection across multiple categores like Advertising, AI, Content Fetcher, and Social Media.	November 20, 2025
AWS WAF managed policy changes	Added Amazon CloudFront Flat rate pricing permissions to the `AWSWAFConsoleFullAccess`, and `AWSWAFConsoleReadOnlyAccess` managed policies.	November 18, 2025

AWS WAF integration with CloudFront flat-rate pricing plans	You can now subscribe CloudFront distributions associated with AWS WAF to a flat-rate pricing plan. This new pricing plan is different from pay-as-you-go pricing in that pricing plans have preset service quotas for usage and resources. Once you subscribe a distribution for a pricing plan, you won't receive unexpected usage charges for that distribut ion. You will have a single price for all resources and features that are included in the pricing plan.	November 18, 2025
Updated AWS Managed Rules for AWS WAF	Added support for Web Bot Authentication for AI Agents to the Bot Control rule group.	November 17, 2025
AWS WAF managed policy changes	Updated Amazon API Gateway permissions to the `AWSWAFConsoleFullA ccess` and `AWSWAFCon soleReadOnlyAccess` managed policies. Added Amazon CloudWatch, AWS AppSync, Amazon Data Firehose, AWS Price List, and AWS Marketplace to the `AWSWAFConsoleFullA ccess` and `AWSWAFCon soleReadOnlyAccess` managed policies.	November 3, 2025

ALB associations per web ACL	Add a new quota for ALB associations per web ACL. The default quota per web ACL is 100.	October 22, 2025
Updated AWS Managed Rules for AWS WAF	Updated the core, or common, rule set (CRS).	October 2, 2025
Updated thresholds for AWS WAF Bot Control	Updated thresholds for `TGT_TokenReuseIpLow` and `TGT_TokenReuseIpMedium` .	August 29, 2025
Add quota for Geo match countries per rule	The maximum number of Geo match countries for a rule is 50.	August 29, 2025
Updated AWS Managed Rules for AWS WAF	Updated the core, or common, rule set (CRS).	August 14, 2025
Updated AWS Managed Rules for AWS WAF	Updated the core, or common, rule set (CRS).	June 18, 2025
First preview release of network security director	AWS Shield network security director is now available in preview, providing insights into your AWS security configuration.	June 17, 2025
Added identity-based customer managed policies for AWS Shield network security director	You can create and manage your own identity-based policies to grant AWS Shield network security director appropriate access to your AWS resources.	June 17, 2025

AWS WAF adds an new console user experience	The AWS WAF console now has a simplified onboarding workflow and an improved way to manage web ACLs, through protection packs (web ACLs).	June 17, 2025
Updated AWS WAF metrics and dimensions	Two new Distributed Denial of Service (DDoS) prevention metrics are now published to the `AWS/ApplicationELB` namespace: `LowReputationRequestsDenied` and `LowReputationPacketsDropped` .	June 11, 2025
New Anti-DDoS managed rule group for AWS WAF	`AWSManagedRulesAntiDDoSRuleSet` protects your resources by detecting , labeling, and challenging requests that are suspected to be participating in DDoS attacks.	June 11, 2025
AWS WAF adds resource-level DDoS protection	You can now use Anti-DDoS functionality to detect and prevent DDoS attacks in Application Load Balancers.	June 11, 2025
AWS WAF adds ASN match statements	You can now match web requests based on the Autonomous System Number (ASN) of the originating IP address.	June 5, 2025

AWS WAF adds ASN as a custom key aggregation option	You can now limit requests from specific Autonomous System Number (ASNs) using custom key aggregation.	June 5, 2025
Updated AWS Managed Rules for AWS WAF	Updated the AWS WAF Bot Control rule set.	May 29, 2025
AWS Firewall Manager security policy updates	Updates to the `FMSServiceRolePolicy` to add permissions required for Amazon CloudFront.	May 21, 2025
Updated AWS WAF metrics and dimensions for silent Challenge	Added `ChallengesAttempted`, `ChallengesSolved`, `ChallengesAttemptedSdk`, and `ChallengesSolvedSdk` to the AWS AWS WAF metrics and dimensions section.	May 16, 2025
AWS WAF managed policy changes	Added AWS Amplify permissions to the `AWSWAFFullAccess`, `AWSWAFReadOnlyAccess`, `AWSWAFConsoleFullAccess`, and `AWSWAFConsoleReadOnlyAccess` managed policies and Amazon CloudFront permissions to the `AWSWAFConsoleFullAccess`, and `AWSWAFConsoleReadOnlyAccess` managed policies.	May 5, 2025

AWS WAF adds support for new CloudFront distributions	You can now associate AWS WAF web ACLs with CloudFront multi-tenant distributions and distribution tenants.	April 28, 2025
URI fragment in log match details	Rule match details in the logs now include the URI fragment from the web request. You can configure logging to redact this field from the logs.	March 17, 2025
New AWS WAF request component	You can now inspect the URI fragment.	March 17, 2025
Added client-side protections to AWS WAF	Client-side protections from AWS Marketplace are now available. You can subscribe and unsubscribe to client-side protections through the AWS Marketplace console.	March 10, 2025
AWS WAF supports new JA4 field matching	You can detect and block traffic based on advanced JavaScript Fingerprinting (JA4) characteristics and use the JA4 fingerprint as one of the supported request keys within WAF rate-based rules.	March 4, 2025
Updated AWS Managed Rules for AWS WAF	Updated the core, or common, rule set (CRS).	March 3, 2025
Updated AWS WAF metrics and dimensions	Added information on usage metrics to the AWS WAF metrics and dimensions section.	February 21, 2025

AWS WAF adds data protection options	AWS WAF now lets you configure data protection either at the protection pack (web ACL) level or at the logging only level.	February 14, 2025
AWS Firewall Manager security policy updates	Updates to the `FMSServiceRolePolicy` to add permissions for getting resource configuration statuses in batches.	February 10, 2025
AWS Firewall Manager quota updates	Updated the Firewall Manager quotas section to reflect new AWS WAF and Network Firewall policies.	February 10, 2025
Updated AWS Managed Rules for AWS WAF	Updated the SQLi database rule group.	January 24, 2025
Updated AWS Managed Rules for AWS WAF	Updated the Linux operating system rule group.	January 24, 2025
Updated AWS Managed Rules for AWS WAF	Updated the SQLi database rule group.	January 24, 2025
AWS Firewall Manager updates to resource tags	Firewall Manager now lets you combine multiple resource tags using the logical AND operator or the logical OR operator. You can also use a new wildcard operator in a resource tag to match any key or value.	January 9, 2025

AWS WAF web ACL dashboard adds top security insights	The AWS WAF console web ACL traffic overview dashboards have a new top insights tab.	January 2, 2025
Rate-based rule aggregation on JA3 and JA4 fingerprints	You can now specify the JA3 fingerprint and JA4 fingerprint in your custom aggregation keys for rate-based rules.	December 20, 2024
AWS WAF adds inspection of JA4 fingerprint	You can now perform an exact match against the web request's JA4 fingerprint, for Amazon CloudFront distributions and Application Load Balancers.	December 20, 2024
Update to the AWS WAF mobile SDK specification	Added the `loadTokenIntoProvider` operation to `WAFTokenProvider`.	November 19, 2024
Application integration SDKs add TV apps	You can use the Android and iOS integration SDKs for TV apps as well as mobile apps.	November 19, 2024
AWS WAF token labeling adds browser fingerprint	Token management now adds a label for the browser fingerprint.	November 13, 2024
Updated AWS Managed Rules for AWS WAF	Updated the Bot Control rule group.	November 7, 2024
Firewall Manager AWS WAF policy can use existing web ACLs	Firewall Manager AWS WAF policies can now retrofit existing account-owned web ACLs, and create new web ACLs only where needed.	October 22, 2024

Updated AWS Managed Rules for AWS WAF	Updated the core rule set (CRS) rule group.	October 16, 2024
Updated AWS Managed Rules for AWS WAF	Updated the Bot Control, ATP, and ACFP managed rule groups.	September 13, 2024
Updated AWS Managed Rules for AWS WAF	Updated the Linux operating system rule group.	September 2, 2024
Updated AWS Managed Rules for AWS WAF	Updated the core rule set (CRS) rule group.	August 30, 2024
Lower rate-based rule threshold	The minimum request rate for a rate-based rule is now 10. Before this, it was 100.	August 30, 2024
Updated AWS Managed Rules for AWS WAF	Updated the Windows operating system rule group.	August 28, 2024
AWS WAF metrics added new metrics for CAPTCHA JavaScript API	AWS WAF added 2 new metrics, `CaptchasAttemptedSdk` and `CaptchasSolvedSdk`, to show account-wide CAPTCHA puzzle attempts using the CAPTCHA JavaScript API.	August 28, 2024
Add quotas on calls per organization for `ListResourcesForWebACL`	AWS WAF now limits the number of calls to `ListResourcesForWebACL` by the accounts in an organization for any single Region.	July 26, 2024

AWS Firewall Manager security policy updates	Updates to `FMSServiceRolePolicy` to add permissions for reading Network Firewall TLS configuration information.	July 22, 2024
Updated AWS Managed Rules for AWS WAF	Updated the WordPress application rule group.	July 15, 2024
Updated AWS Managed Rules for AWS WAF	Updated the Linux operating system rule group.	July 12, 2024
Updated AWS Managed Rules for AWS WAF	Updated the core rule set (CRS) rule group.	July 9, 2024
Updated AWS Managed Rules for AWS WAF	Updated the PHP application and Windows operating system rule groups.	July 3, 2024
Clarify how JSON body parsing works	Updated coverage for JSON body inspection to clarify how AWS WAF handles parsing and the body parsing fallback behavior.	June 25, 2024
Updated AWS Managed Rules for AWS WAF	Updated the Linux operating system rule group.	June 6, 2024
AWS WAF managed policy changes	Updated `WAFV2LoggingServiceRolePolicy` and `AWSServiceRoleForWAFV2Logging` to add Statement IDs (Sids) to the permissions settings.	June 3, 2024

AWS WAF managed policy change tracking	AWS WAF started tracking changes for the managed policy `WAFV2LoggingServiceRolePolicy` and the service-linked role `AWSServiceRoleForWAFV2Logging` .	June 3, 2024
Updated AWS Managed Rules for AWS WAF	The Bot Control, ATP, and ACFP managed rule groups are now versioned and will provide SNS notifications for version updates, the same as other versioned AWS Managed Rules.	May 29, 2024
Updated AWS Managed Rules for AWS WAF	Updated the POSIX operating system rule group, `AWSManagedRulesUnixRuleSet` .	May 28, 2024
CAPTCHA and Challenge actions	Added clarification that browser clients require HTTPS to run CAPTCHA puzzles and silent challenges.	May 24, 2024
Integration with Amazon Security Lake	You can now use Security Lake to collect protection pack (web ACL) traffic data. For information, see Collecting data from AWS services in the *Amazon Security Lake user guide*.	May 22, 2024

Integration with Amazon Security Lake	You can now use Security Lake to collect protection pack (web ACL) traffic data. For information, see Collecting data from AWS services in the *Amazon Security Lake user guide*.	May 22, 2024
Updated AWS Managed Rules for AWS WAF	Updated the core rule set (CRS) rule group.	May 21, 2024
Updated AWS Managed Rules for AWS WAF	Updated the SQLi database rule group.	May 14, 2024
Updated AWS Managed Rules for AWS WAF	Updated the known bad inputs and POSIX operating system rule groups.	May 8, 2024
Updated AWS Managed Rules for AWS WAF	Updated the Windows operating system rule group.	May 3, 2024
AWS WAF mobile SDK Android Kotlin code samples	Added example code for Kotlin-based Android integrations.	May 2, 2024
AWS WAF metrics added dimensions and new metrics	AWS WAF added new dimension for `ManagedRuleSetRule` in rule metrics and new metrics for the matched rule action for label metrics.	May 2, 2024
AWS Firewall Manager supports network ACL policies	Firewall Manager now supports the management of Amazon VPC network access control lists (ACLs) through Firewall Manager network ACL policies.	April 25, 2024

AWS Firewall Manager security policy updates	Updates to `FMSServiceRolePolicy` to add permissions for managing network ACLs.	April 22, 2024
Updated health check metrics list	We removed some metrics from the list of those that are commonly used in health checks.	April 16, 2024
Updates for Firewall Manager security group policies	We've updated our usage audit security group policies and improved the documentation. See the usage audit policy section and the sections on best practices and limitations.	April 2, 2024
Updated Bot Control examples	Added examples depicting the targeted inspection level and updated existing examples to reflect best practices.	March 27, 2024
Updated ATP examples	Added example depicting response inspection configuration and updated existing examples to reflect best practices.	March 27, 2024
Updated ACFP examples	Added example depicting response inspection configuration.	March 27, 2024
Update Amazon CloudWatch Logs log stream limits	AWS WAF no longer has per-protection pack (web ACL) limits on publishing logs to CloudWatch Logs log streams.	March 27, 2024

AWS Shield Advanced application layer (layer 7) protections	Updated general and best practice guidance for application layer detection and mitigation, web ACL use, rate-based rules, and automatic application layer DDoS mitigation.	March 14, 2024
Updated AWS Managed Rules for AWS WAF	Updated the IP reputation rule group.	March 13, 2024
Changes to body inspection size limits	AWS WAF now supports larger body inspection size limits for some regional resources.	March 7, 2024
Configurable evaluation window for AWS WAF rate-based rules	You can now configure the time window that rate-based rules use to count requests, to 1, 2, 5, or 10 minutes. The default is 5, which was the only option before this release.	February 28, 2024
Expanded logging information for CAPTCHA and Challenge	The top level `captchaResponse` and `challengeResponse` fields are now populated with the last of these actions to be applied to a request, whether terminating or non-terminating. Prior to this, these fields were populated only for terminating actions.	February 22, 2024

JavaScript CAPTCHA API key management	You can now delete CAPTCHA JS API keys through the AWS WAF APIs.	February 6, 2024
AWS WAF CAPTCHA puzzles audio	The audio version of the CAPTCHA puzzle now supports multiple languages.	February 6, 2024
AWS WAF challenge and CAPTCHA token labeling	Token management now adds labels for the CAPTCHA token and has enhanced the token labeling for the challenge token.	December 20, 2023
Updated AWS Managed Rules for AWS WAF	Updated the known bad inputs rule group.	December 16, 2023
Updated AWS Managed Rules for AWS WAF	Updated the known bad inputs rule group.	December 14, 2023
Updated AWS Managed Rules for AWS WAF	Updated the core rule set (CRS) rule group.	December 6, 2023
Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: AWS WAF Bot Control.	December 5, 2023

New web authentication labels for AI agents Bot Control managed rule group.	The Bot Control managed rule group now supports Web Bot Authentication (WBA) as a cryptographic verification method for bots and AI agents accessing your CloudFront distributions. This feature enables legitimate AI crawlers and agents to prove their identity without requiring traditional challenge-response mechanisms.	November 20, 2023
Updated Firewall Manager AWS Config prerequisites	If you use a custom IAM role instead of the Firewall Manager managed role for AWS Config, you must ensure that your permission policy allows AWS Config recorder to record Firewall Manager resources.	November 17, 2023
AWS WAF console dashboards	We corrected the guidance for viewing all rules and sampled requests for a protection pack (web ACL) in the AWS WAF console.	November 17, 2023
Updated AWS Managed Rules for AWS WAF	Updated the Bot Control rule group.	November 14, 2023
AWS WAF console has new web ACL dashboards	The web ACL page in the AWS WAF console has new web traffic overview dashboards.	November 14, 2023

Updated ATP managed rule group	Corrected label information for the rules `VolumetricIpFailedLoginResponseHigh` and `VolumetricSessionFailedLoginResponseHigh` .	November 13, 2023
Updated ACFP managed rule group	Corrected label information for the rules `VolumetricIPSuccessfulResponse` and `VolumetricSessionSuccessfulResponse` .	November 13, 2023
Updated AWS Managed Rules for AWS WAF	Updated the core rule set (CRS) rule group.	November 2, 2023
Shield Advanced automatic application layer DDoS mitigation	Shield Advanced now maintains a rate-based rule in the automatic mitigation rule group that limits the volume of requests from IP addresses known to be sources of DDoS attacks.	October 31, 2023
Updated AWS Managed Rules for AWS WAF	Updated the core rule set (CRS) rule group.	October 30, 2023
Bot Control managed rule group removed signal label for the request CSP	The Bot Control managed rule group removed the signal label that indicates the cloud service provider (CSP).	October 28, 2023
Bot Control managed rule group signal label for the request CSP	The Bot Control managed rule group signal labels include a label that indicates the cloud service provider (CSP).	October 27, 2023

Updated AWS WAF IAM permissions information	For the AWS WAF actions that manage protection pack (web ACL) associations, the policy actions section now lists the permissions requirements for each web application resource type.	October 25, 2023
Firewall Manager management of modified web ACLs	When you enable management of unassociated web ACLs, Firewall Manager doesn't include the modified web ACLs in the one-time cleanup of unused resources.	October 19, 2023
Updated AWS Managed Rules for AWS WAF	Updated the POSIX operating system rule group, `AWSManagedRulesUnixRuleSet`.	October 12, 2023
AWS WAF metrics added dimensions	AWS WAF added new dimensions for viewing web ACL metrics.	October 12, 2023
Updated AWS Managed Rules for AWS WAF	Updated the core rule set (CRS) rule group.	October 11, 2023
Update to the AWS WAF mobile SDK specification	Added the `storeTokenInCookieStorage` operation to `WAFTokenProvider`.	October 11, 2023
Exception deployments AWS Managed Rules for AWS WAF	Updated two static versions of the known bad inputs rule group and updated the default version to point to the most recent static version.	October 4, 2023

AWS WAF HTML entity decode text transformation	Expanded the functionality of the HTML entity decode text transformation.	October 4, 2023
Added new option to Firewall Manager security group common policy	Firewall Manager now can distribute security group references to replica security groups.	October 3, 2023
AWS WAF adds inspection of JA3 fingerprint	You can now perform an exact match against the web request's JA3 fingerprint, for Amazon CloudFront distributions and Application Load Balancers.	September 26, 2023
Updates to Firewall Manager security group policy rules settings	Firewall Manager now supports security group referencing from primary security groups to replica security groups.	September 25, 2023
Updated Shield Advanced automatic application layer DDoS mitigation	Firewall Manager now supports Application Load Balancer resources for Shield Advanced policies configured with automatic application layer DDoS mitigation.	September 14, 2023
Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: AWS WAF Bot Control.	September 6, 2023

AWS WAF Bot Control	The targeted protection level of the Bot Control managed rule group now inspects for token reuse between IP addresses. It also now provides optional, machine-learning analysis of traffic statistics to detect some bot-related activity.	September 6, 2023
Update to the AWS WAF mobile SDK specification	Lowered the min, max, and default values for `tokenRefreshDelaySec` from min 300, max 600, and default 300 to min 88, max 300, and default 88.	September 5, 2023
Updated AWS Managed Rules for AWS WAF	Updated the AWS WAF Bot Control rule group.	August 30, 2023
Shield Advanced automatic application layer DDoS mitigation	Added guidance for using CloudFormation to manage the web ACLs that you use with automatic application layer DDoS mitigation.	August 30, 2023
New Firewall Manager content audit security group policy option	Added new option for auditing overly permissive rule groups, and improved console procedure descriptions.	August 29, 2023

New Firewall Manager Shield and AWS WAF policy option	If you enable management of unassociated web ACLs in AWS WAF and Shield, Firewall Manager only creates web ACLs in the accounts within policy scope only if the web ACLs will be used by at least one resource.	August 9, 2023
Updated AWS Managed Rules for AWS WAF	Updated the core rule set (CRS) rule group.	July 26, 2023
Rate-based rule aggregation on URI path	You can now specify the URI path in your custom aggregation keys for rate-based rules.	July 19, 2023
New AWS WAF policy rule option in AWS Firewall Manager	AWS Firewall Manager adds support for configuring AWS WAF web request body inspection size limits.	July 18, 2023
AWS WAF managed policy changes	Updated `AWSWAFFullAccessPolicy`, `AWSWAFConsoleFullAccess`, `AWSWAFReadOnlyAccess`, and `AWSWAFConsoleReadOnlyAccess` to add AWS Verified Access to the resource types that you can protect with AWS WAF.	June 17, 2023
Updated AWS Managed Rules for AWS WAF	Updated the rule group `AWSManagedRulesACFPRuleSet`.	June 13, 2023

Update to AWS WAF Fraud Control account takeover prevention (ATP)	You can now specify the login endpoint for the ATP managed rule group using a r egular expression.	June 13, 2023
New information for the CAPTCHA JavaScript API	New section describes how to serve a custom CAPTCHA puzzle when AWS WAF responds to a request with a CAPTCHA.	June 13, 2023
New ACFP managed rule group	Use the new rule group `AWSManagedRulesACFPRuleSet` to detect and block fraudulent account creation attempts.	June 13, 2023
New AWS WAF Fraud Control account creation fraud prevention (ACFP)	You can detect and block fraudulent account creation attempts with the new AWS WAF Fraud Control account creation fraud prevention (ACFP) managed rule group `AWSManagedRulesACFPRuleSet` . With protected CloudFront distributions, you can also use ACFP to block new account creation attempts from clients that have recently submitted too many failed account creation attempts.	June 13, 2023

AWS WAF managed policy changes	Updated `AWSWAFFullAccessPolicy`, `AWSWAFConsoleFullAccess`, `AWSWAFReadOnlyAccess`, and `AWSWAFConsoleReadOnlyAccess` to correct the access settings for AWS App Runner services.	June 6, 2023
Added limitation for Firewall Manager security group policies	If a shared VPC is later unshared, Firewall Manager won't delete the replica security groups in the associated account.	June 2, 2023
New AWS WAF request component: Header order	You can now match against an ordered list of the names of the headers in the request.	May 30, 2023
Updated AWS Managed Rules for AWS WAF	Updated the Linux operating system rule set.	May 22, 2023
Updated the organization of the AWS WAF rules section	The rules statement listings are now grouped by statement type.	May 16, 2023
Moved topic: Listing IP addresses that are being rate limited	The topic for listing IP addresses that are being rate limited by a rate-based rule is now under the rate-based rules topic.	May 16, 2023

Expanded options for rate-based rules	You can now rate limit web requests based on aggregation keys other th an IP addresses, and you can aggregate using combinati ons of keys. You can also rate limit all requests that match a scope-down statement, without further aggregation.	May 16, 2023
Firewall Manager quota increases	Increased the number of Firewall Manager policies per organization in AWS Organizations from 20 to 50. Increased maximum number of primary security groups per policy from one to three. Changed the maximum number of WCUs from a soft quota to a hard quota.	May 5, 2023
Increased maximum WCUs per rule group	You can now use up to 5,000 protection pack (web ACL) capacity units (WCUs) per rule group without requesting an increase from support. This new limit can't be increased.	May 1, 2023
AWS WAF Amazon S3 log bucket locations with prefixes	AWS WAF now allows prefixes in Amazon S3 log bucket names.	May 1, 2023
Updated AWS Managed Rules for AWS WAF	Updated the core rule set (CRS) rule group.	April 28, 2023

Added support for AWS Verified Access instances to AWS WAF	You can now associate an AWS WAF web ACL with a Verified Access instance. This change is only available in the latest version of AWS WAF and not in AWS WAF Classic.	April 28, 2023
Revised chapter on working with multiple Firewall Manager administrators	You can now designate multiple Firewall Manager administrators to create and manage the firewall resources of your organization.	April 24, 2023
AWS Firewall Manager managed policy update	Updated `FMSServiceRolePolicy` .	April 21, 2023
New JavaScript client application integration for CAPTCHA	You can now customize the placement and characteristics of the CAPTCHA puzzle in your JavaScript client applications.	April 20, 2023
Application integration renamed to intelligent threat integration	We renamed the existing functionality for client application integrations to intelligent threat integrations, to help distinguish between that and the new CAPTCHA application integration for JavaScript.	April 20, 2023

Variable pricing for web ACL WCUs beyond 1,500	Using more than 1,500 web ACL capacity units (WCUs) in your web ACL incurs additional costs, which are adjusted automatically as your web ACL WCU usage increases and decreases. The web ACL maximum is 5,000 WCUs.	April 11, 2023
Increased maximum WCUs per protection pack (web ACL)	You can now use up to 5,000 protection pack (web ACL) capacity units (WCUs) per protection pack (web ACL) without requesting an increase from support. This new limit can't be increased.	April 11, 2023
Body inspection size limits for CloudFront protection packs (web ACLs)	For protection packs (web ACLs) that protect Amazon CloudFront distributions, you can increase the body inspection size limit up to 64 KB in your protection pack (web ACL) configuration.	April 11, 2023
Body inspection size increase for CloudFront	The maximum AWS WAF body inspection size limit for Amazon CloudFront distributions is increased from 8 KB to 64 KB. The default inspection size limit for CloudFront is 16 KB.	April 11, 2023

New AWS WAF policy rule options in AWS Firewall Manager	AWS Firewall Manager adds support for AWS WAF Fraud Control account takeover pr evention (ATP) and AWS WAF Bot Control AWS Managed Rules rule groups, Amazon S3 logging destinations, rule action overrides, `CAPTCHA` and `Challenge` rule actions, and token domain lists.	April 7, 2023
AWS WAF managed policy changes	Updated `AWSWAFFul lAccessPolicy` , `AWSWAFConsoleFullA ccess` , `AWSWAFRea dOnlyAccess` , and `AWSWAFConsoleRead0 nlyAccess` to add AWS App Runner services to the resource types that you can protect with AWS WAF.	March 30, 2023
Added warning about the usage of tags within security group policies	Firewall Manager won't update the tags of existing security groups or create new security groups if the policy has tags that conflict with the organization's tag policy.	March 28, 2023
Updating service role information	Updated how to use a service role with Firewall Manager.	March 8, 2023

Corrected information about how rate-based rules perform rate limiting	Rate based rules with scope-down statements only rate limit requests that match the rule's scope-down statement . We were stating that the limiting applied to all requests for any rate limited IP address.	March 1, 2023
Updated AWS Managed Rules for AWS WAF	Updated the PHP application rule group.	February 27, 2023
Added support for AWS App Runner to AWS WAF	You can now associate an AWS WAF web ACL with an AWS App Runner service. This change is only available in the latest version of AWS WAF and not in AWS WAF Classic.	February 23, 2023
Updated the IAM guidance for AWS Firewall Manager	Updated guide to align with the IAM best practices . For more information, see Security best practices in IAM.	February 16, 2023
Updated AWS Managed Rules for AWS WAF	Updated the rule group `AWSManagedRulesATPRuleSet` to add login response inspection in web ACLs that protect Amazon CloudFront distributions.	February 15, 2023
AWS WAF Fraud Control account takeover prevention (ATP) login response inspection	For protected CloudFront distributions, you can now use ATP to block new login attempts from clients that have recently submitted too many failed login attempts.	February 15, 2023

Updated AWS Managed Rules for AWS WAF	Updated the core rule set.	January 25, 2023
Best practices for intelligent threat mitigation	Added a section with best practices for implementing Bot Control, ATP, and other intelligent threat mitigation features.	January 22, 2023
How to inspect HTTP/2 pseudo headers	Added a section that maps HTTP/2 pseudo headers to their corresponding web request components.	January 20, 2023
Updated the IAM guidance for AWS WAF Classic	Updated guide to align with the IAM best practices . For more information, see Security best practices in IAM.	January 3, 2023
Updated the IAM guidance for AWS Shield	Updated guide to align with the IAM best practices . For more information, see Security best practices in IAM.	January 3, 2023
Updating Amazon Route 53 Resolver DNS Firewall policies	Added information about deleting Amazon Route 53 Resolver DNS Firewall rule groups.	December 29, 2022
Updated AWS Managed Rules for AWS WAF	Updated the Linux operating system rule set.	December 15, 2022
Updated AWS Managed Rules for AWS WAF	Updated the core rule set.	December 5, 2022

Firewall Manager adds support for Fortigate Cloud Native Firewall (CNF) as a Service policies	Firewall Manager now supports the Fortigate CNF policies.	December 2, 2022
Removed AWS Config requirement for DNS Firewall policies	For DNS Firewall policies, you now only need to enable Config for the resource type EC2 VPC.	November 17, 2022
AWS Firewall Manager managed policy update	Updated `FMSServiceRolePolicy` .	November 15, 2022
Expansion of language options for the AWS WAF CAPTCHA puzzle	The CAPTCHA puzzle now offers its written instructions in multiple languages. The instructions inside each audio puzzle are still provided in English only.	November 11, 2022
New Firewall Manager quotas for resource sets	Added new quotas for resource sets.	November 8, 2022
Add support for resource sets	You can create resource sets to group resources to manage in an Firewall Manager policy.	November 8, 2022
Add support for importing firewalls from Network Firewall	You can now import and manage existing firewalls in Network Firewall policies using resource sets.	November 8, 2022
AWS Firewall Manager managed policy update	Updated `AWSFMAdminReadOnlyAccess` .	November 2, 2022

Geo match statement now adds labels to requests for country and region	You can now manage geographical request origins at the region level by combining geo matching with label matching.	October 31, 2022
Renamed the top-level section: Managed protections	The section is now named AWS WAF intelligent threat mitigation, which aligns with our marketing pages.	October 27, 2022
New targeted protection level in the Bot Control managed rule group	The Bot Control managed rule group now offers additiona l, targeted rules for the detection and mitigation of sophisticated bots. This protection level is available for additional fees.	October 27, 2022
New section on AWS WAF tokens	Understand how AWS WAF uses tokens for intelligent threat mitigation.	October 27, 2022
Added important note about updating Firewall Manager Network Firewall policies	When you update a Firewall Manager policy, all Network Firewall policies that were created by the policy will be updated with the Firewall Manager policy's Network Firewall policy configuration.	October 27, 2022

Action overrides in rule groups	You can now override the actions of the rules in a rule group to any rule action setting. As with the prior Count action override, you can apply your overrides to all rules in a rule group and to individual rules.	October 27, 2022
AWS WAF new Challenge rule action option	You can configure rules to use a Challenge, to verify that requests are being sent by browsers.	October 27, 2022
AWS WAF allows token sharing across multiple protected applications	You can enable the use of tokens across multiple protected applications by configuring a token domain list for your protection pack (web ACL).	October 27, 2022
All headers specification is not case sensitive	Changed the all headers specification to be case insensitive. This matches the single header behavior.	October 26, 2022
AWS Firewall Manager managed policy changes	Corrections to `AWSFMAdminFullAccess` .	October 21, 2022
Updated AWS Managed Rules for AWS WAF	Updated the known bad inputs rule group.	October 20, 2022
Updated AWS Managed Rules for AWS WAF	Updated the known bad inputs rule group.	October 5, 2022

Update to the AWS WAF mobile SDK specification	Lowered the default value for `tokenRefreshDelaySec` from 600 (10 minutes) to 300 (5 minutes).	September 30, 2022
Updated AWS Managed Rules for AWS WAF	Corrected the label names provided in this documentation for the following rule groups: POSIX operating system, PHP application, WordPress application.	September 19, 2022
New AWS WAF policy rule option in AWS Firewall Manager	AWS Firewall Manager now supports customized web requests and responses for default web actions in AWS WAF policies.	September 9, 2022
Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: IP reputation.	August 30, 2022
AWS WAF managed policy changes	Updated `AWSWAFFullAccessPolicy`, `AWSWAFConsoleFullAccess`, `AWSWAFReadOnlyAccess`, and `AWSWAFConsoleReadOnlyAccess` to add Amazon Cognito user pools to the resource types that you can protect with AWS WAF.	August 25, 2022
AWS WAF Fraud Control account takeover prevention (ATP)	You can now use the AWS WAF Fraud Control account takeover prevention (ATP) functionality with Amazon CloudFront distributions.	August 24, 2022

Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: Known bad inputs.	August 22, 2022
Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: `AWSManagedRulesATPRuleSet` .	August 11, 2022
Added support for Amazon Cognito user pools to AWS WAF	You can now associate an AWS WAF web ACL with an Amazon Cognito user pool. This change is only available in the latest version of AWS WAF and not in AWS WAF Classic.	August 11, 2022
Added a section on deployments for versioned AWS Managed Rules rule groups	Added a new section documenting deployments for versioned AWS Managed Rules rule groups. The section includes information about how default versions are named during release candidate deployments.	July 29, 2022
Updated requirements for configuring logging for Network Firewall policies	Added requirements for Network Firewall policies that use an encrypted Amazon S3 bucket as the log destination.	July 26, 2022
Sensitivity level option for SQLi rule statement	You can now raise the sensitivity of your SQL injection rule statements. This doesn't change the behavior of existing statements, whose sensitivity level at the default of LOW.	July 15, 2022

Added Network Firewall policy configuration option	Firewall Manager now supports stateful evaluation order and default actions in Network Firewall firewall policy configurations.	July 14, 2022
Updates to Firewall Manager security group policy rules settings	Firewall Manager now supports tag distribution from primary security groups to replica security groups.	July 7, 2022
Updates to the AWS Shield guide	Expanded the information in the Shield guide to describe how Shield performs event mitigation.	June 24, 2022
Updated guidance for testing and tuning AWS WAF protections	The general guidance for testing and tuning AWS WAF is updated and is now a top-level topic.	June 20, 2022
Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: Core rule set (CRS).	June 9, 2022
New Firewall Manager confused deputy guidance	Added guidance on how to prevent the confused deputy problem for Firewall Manager.	June 1, 2022
Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: Core rule set (CRS).	May 24, 2022
New AWS WAF request components: Headers and Cookies	You can now inspect cookies and you can inspect all headers, in addition to just a single header.	April 29, 2022

New AWS WAF request components: Headers and Cookies	You can now inspect the cookies in a web request and you can inspect all headers in a web request, in addition to just a single header.	April 29, 2022
AWS WAF handling for oversize body, headers, and cookies request components	You can now specify how AWS WAF should handle oversize request bodies, headers, and cookies inside your rules that inspect these components. Rules that you already created that inspect these component s have behavior that matches the new Continue option for oversize handling.	April 29, 2022
AWS WAF Amazon S3 log policy changes	Updated the Amazon S3 log permission policy and example.	April 12, 2022
Automatic application layer DDoS mitigation option now available with AWS Shield Advanced for Application Load Balancer	Shield Advanced now supports automatic applicati on layer DDoS mitigation for Application Load Balancers , making it available for all application layer protectio ns. You can configure Shield Advanced to automatic ally count or block the web requests that are part of an application layer DDoS attack on a protected resource.	April 8, 2022

Added an indicator of the current default version setting for managed rule groups	Managed rule group version lists now indicate which version is the current default.	April 8, 2022
Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: AWS WAF Bot Control.	April 6, 2022
Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: Known bad inputs.	March 31, 2022
Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: Known bad inputs.	March 30, 2022
Firewall Manager adds support for the Palo Alto Networks Cloud Next Generation Firewall (NGFW)	Firewall Manager now supports the Palo Alto Networks Cloud Next Generation Firewall (NGFW).	March 30, 2022
Add support for Palo Alto Networks Cloud NGFW to AWS Firewall Manager	AWS Firewall Manager now supports Palo Alto Networks Cloud Next Generation Firewall (NGFW) policies.	March 30, 2022
Updates to the AWS Shield guide	Expanded the information in the Shield guide to describe how Shield performs event detection and to provide examples of DDoS resilient architectures.	March 16, 2022

Updates to the AWS Shield guide	Expanded the information in the Shield guide and improved the organization of various sections. The main changes are in the following Shield guide sections: Shield Response Team (SRT) support, Resource protections in AWS Shield Advanced, and Visibility into DDoS events.	February 28, 2022
Firewall Manager now supports the Network Firewall centralized deployment model	Added a new procedure that explains how to configure policies that use distributed and centralized deployment models.	February 24, 2022
Firewall Manager adds support for the AWS Network Firewall centralized deployment model	You can now configure your AWS Network Firewall policies to use either the distributed or centralized deployment model. With the distributed deployment model, Firewall Manager creates and maintains firewall endpoints in each VPC that's within the policy scope. With the centralized deployment model, Firewall Manager creates and maintains firewall endpoints in a single inspection VPC.	February 24, 2022

Add support for AWS WAF managed rule group versioning to AWS Firewall Manager	AWS Firewall Manager now supports AWS WAF managed rule group versioning in Firewall Manager AWS WAF policies.	February 18, 2022
AWS Firewall Manager managed policy change	Update to `FMSServiceRolePolicy` .	February 16, 2022
Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: IP reputation lists.	February 15, 2022
Updated AWS Managed Rules for AWS WAF	Updated the AWS WAF Fraud Control account takeover prevention (ATP) rule group `AWSManagedRulesATPRuleSet` .	February 11, 2022
Changes to the organization of the AWS WAF guide	Added a new top-level section for managed protections. Moved the CAPTCHA section from under rules to under the new managed protections section. Moved the labels section from under rules to its own top-level section.	February 11, 2022
AWS WAF client application integrations	Use the AWS WAF JavaScript and mobile client APIs to integrate your client applications with the intelligent threat mitigation AWS Managed Rules rule groups for enhanced detection.	February 11, 2022

AWS WAF Fraud Control account takeover prevention (ATP)	You can detect and block account takeover attempts with the new AWS WAF Fraud Control account takeover prevention (ATP) managed rule group `AWSManagedRulesATPRuleSet`.	February 11, 2022
Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: Known bad inputs.	January 28, 2022
AWS WAF managed policy changes	Updated `AWSWAFFullAccessPolicy` and `AWSWAFConsoleFullAccess` to correct logging permissions.	January 11, 2022
Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: core rule set (CRS), SQLi database.	January 10, 2022
Firewall Manager supports Shield Advanced automatic application layer DDoS mitigation	Firewall Manager Shield Advanced policies for Amazon CloudFront resources now include support for automatic application layer DDoS mitigation.	January 7, 2022
AWS Firewall Manager managed policy change	Update to `FMSServiceRolePolicy`.	January 7, 2022
Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: Known bad inputs.	December 17, 2021
Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: Known bad inputs.	December 11, 2021

Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: Known bad inputs.	December 10, 2021
New AWS Shield Advanced service-linked role	Added `AWSServiceRoleForAWSShield` to support the automatic application layer DDoS mitigation functionality.	December 1, 2021
New AWS Shield managed policy	Added `AWSShieldServiceRolePolicy` to support the automatic application layer DDoS mitigation functionality.	December 1, 2021
Automatic application layer DDoS mitigation option now available with AWS Shield Advanced for CloudFront	Shield Advanced now supports automatic application layer DDoS mitigation for Amazon CloudFront distributions. You can configure Shield Advanced to automatically count or block the web requests that are part of an application layer DDoS attack on a CloudFront distribution.	December 1, 2021
Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: core rule set (CRS), Windows operating system, Linux operating system, and IP reputation lists.	November 23, 2021
AWS Firewall Manager managed policy change	Update to `FMSServiceRolePolicy` .	November 18, 2021

Expanded logging options for AWS WAF	You can now log protection pack (web ACL) traffic to an Amazon CloudWatch Logs log group or an Amazon Simple Storage Service (Amazon S3) bucket. These options are in addition to the existing option of logging to an Amazon Data Firehose delivery stream.	November 15, 2021
AWS WAF managed policy changes	Updated `AWSWAFFullAccessPolicy` and `AWSWAFConsoleFullAccess` to support additional logging destinations.	November 15, 2021
AWS WAF new CAPTCHA rule action option	You can configure rules to run a CAPTCHA against web requests and, as needed, send a CAPTCHA problem to the client.	November 8, 2021
Updated AWS Managed Rules for AWS WAF	Updated the core rule set (CRS) rule group.	October 27, 2021
Updated AWS Managed Rules for AWS WAF	All AWS Managed Rules rule groups now support labeling. The rule descriptions include the label specifications.	October 25, 2021
Firewall Manager supports Network Firewall log filtering	AWS Firewall Manager now supports log filtering for Network Firewall policies.	October 4, 2021
AWS Firewall Manager managed policy change	Update to `FMSServiceRolePolicy` .	September 29, 2021

Added regex match statement	You can now match web requests against a single regular expression.	September 22, 2021
Rate-based rules inside AWS WAF rule groups	You can now define rate-based rules inside AWS WAF rule groups. In AWS Firewall Manager, this capability is fully supported for AWS WAF policies.	September 13, 2021
Automatically remove out-of-scope resource protections in AWS Firewall Manager	AWS Firewall Manager allows you to automatically remove protections from resources that leave policy scope.	August 25, 2021
AWS Firewall Manager managed policy change	Update to `FMSServiceRolePolicy` .	August 12, 2021
Added versioning to managed rule groups	Managed rule group providers can now version their rule groups.	August 9, 2021
Modify AWS Firewall Manager administrator requirements	You can use the organization's management account as the Firewall Manager administrator account. This had been disallowed.	August 2, 2021
Firewall Manager quota increase	Increased the number of Amazon VPC instances that you can have in scope of a Firewall Manager policy from 10 to 100.	July 28, 2021

AWS Firewall Manager support for AWS Network Firewall route table monitoring	AWS Firewall Manager now supports route table monitoring, and provides remediation action recommendations to security administrators for AWS Network Firewall policies with misconfigured routes.	July 8, 2021
AWS WAF additional text transformation options	Expanded options for text transformations, which you can apply to web request components before inspecting them.	June 24, 2021
Modified naming for Firewall Manager AWS WAF policy resources	The naming for the web ACLs, rule groups, and logging that Firewall Manager manages for your AWS WAF policies has changed.	May 26, 2021
Updated AWS Managed Rules for AWS WAF	Updated support for labeling to IP reputation lists and removed suffixes on rule names for Amazon IP reputation list.	May 4, 2021

Add support for AWS Organizations Delegated Administrator	When you set the AWS Firewall Manager administrator account, Firewall Manager now designates the account as the AWS Organizations delegated administrator for Firewall Manager. With this change, when you set the Firewall Manager administrator account, you must provide a member account other than the organization's management account. This change doesn't affect your existing settings.	April 30, 2021
Updated AWS Managed Rules for AWS WAF	Updated the AWS WAF Bot Control rule group.	April 1, 2021
Set individual rule actions to Count in a rule group	You can now set the individual rule actions in a rule group to Count. The information for the existing override, which is at the rule group level, has been corrected.	April 1, 2021
Scope-down statement for managed rule groups	You can now use a scope-down statement with managed rule groups in the same way as you can with a rate-based statement.	April 1, 2021
Log filtering	You can now filter the protection pack (web ACL) traffic that you log based on rule action and label.	April 1, 2021

AWS WAF labels on web requests	You can configure rules to add labels to matching web requests and to match on labels that are added by other rules.	April 1, 2021
AWS WAF Bot Control	You can monitor and control bot traffic with the new AWS WAF Bot Control feature, which combines the Bot Control managed rule group with web request labeling, scope-down statements, and log filtering.	April 1, 2021
Firewall Manager supports Amazon Route 53 Resolver DNS Firewall policies	AWS Firewall Manager supports central management of Amazon Route 53 Resolver DNS Firewall outbound DNS traffic filtering for your VPCs.	March 31, 2021
Custom request and response handling	You can include custom headers for web requests that AWS WAF doesn't block and you can send custom responses for web requests that AWS WAF blocks. This is available for protection pack (web ACL) default action and rule action settings.	March 29, 2021
AWS Firewall Manager managed policy change	Update to `FMSServiceRolePolicy` .	March 17, 2021

Updated AWS Managed Rules for AWS WAF	Updated the following rule groups: core rule set (CRS), admin protection, known bad inputs, and Linux operating system.	March 3, 2021
AWS Shield managed policy change tracking	Shield started tracking changes for its AWS managed policies.	March 3, 2021
AWS Firewall Manager managed policy change tracking	Firewall Manager started tracking changes for its AWS managed policies.	March 2, 2021
AWS WAF managed policy change tracking	AWS WAF started tracking changes for its AWS managed policies.	March 1, 2021
Inspect a web request body as parsed JSON	Added the option to inspect the web request body as parsed and filtered JSON. This is in addition to the existing option to inspect th e web request body as plain text.	February 12, 2021
Firewall Manager supports AWS Network Firewall policies	AWS Firewall Manager supports central managemen t of AWS Network Firewall network traffic filtering for your VPCs.	November 17, 2020
Add support for AWS Shield Advanced protection groups	You can now group your protected resources into logical groups and manage their protections collectively.	November 13, 2020

Added support for AWS AppSync to AWS WAF	You can now associate an AWS WAF web ACL with your AWS AppSync GraphQL API. This change is only available in the latest version of AWS WAF and not in AWS WAF Classic.	October 1, 2020
Updated AWS Managed Rules for AWS WAF	Updated the Windows operating system rule set.	September 23, 2020
Updated AWS Managed Rules for AWS WAF	Updated the rule sets PHP application and POSIX operating system.	September 16, 2020
Updated AWS Shield console	AWS Shield offers a new console option, with an improved user experience. The console guidance in the documentation is for the new console.	September 1, 2020
Firewall Manager updates to common security group policies	AWS Firewall Manager common security group policies now support Application Load Balancers and Classic Load Balancers resource types through the console implementation. The new options are available in the common policy's **Policy scope** settings.	August 11, 2020
Updated AWS Managed Rules for AWS WAF	Updated the core rule set.	August 7, 2020

Specify IP address location in web request	Added the option to use IP addresses from an HTTP header that you specify, instead of using the web request origin. The alternate header is commonly `X-Forwarded-For` (XFF), but you can specify any header name. You can use this option for IP set matching, geo matching, and rate-based rule count aggregation.	July 9, 2020
Firewall Manager updates to content audit security group policies	AWS Firewall Manager has expanded functionality for content audit security group policies including a managed rules option, that uses managed application and protocol lists, and details for resource violations.	July 7, 2020
Firewall Manager managed lists	AWS Firewall Manager now supports managed applicati on and protocol lists. Firewall Manager manages some lists and you can create and manage your own.	July 7, 2020
Firewall Manager supports shared VPCs in common security group policies	AWS Firewall Manager now supports using common security group policies in shared VPCs. You can do this in addition to using them in the VPCs owned by in-scope accounts.	May 26, 2020

Updated AWS Managed Rules for AWS WAF	Added documentation for each rule in the AWS Managed Rules for AWS WAF.	May 20, 2020
Updated AWS Managed Rules for AWS WAF	Updated the Linux operating system rule group.	May 19, 2020
Add support for migrating AWS WAF Classic resources to AWS WAF (v2)	You can now use the console or API to export your AWS WAF Classic resources for migration to the latest version of AWS WAF.	April 27, 2020
Add support for AWS Organizations organizational units in policy scope	AWS Firewall Manager now supports using AWS Organizations organizational units (OUs) to specify policy scope. You can use OUs to include or exclude accounts from the scope, in addition to including or excluding specific accounts. Specifying an OU is the same as specifying all accounts in the OU and in any of its child OUs, including any child OUs and accounts that are added at a later time.	April 6, 2020
Add support for AWS WAF (v2) to AWS Firewall Manager	AWS Firewall Manager now supports the latest version of AWS WAF, in addition to the prior version, AWS WAF Classic.	March 31, 2020

Update to AWS Firewall Manager common security group policies	AWS Firewall Manager common security group policy now has the option to apply the policy to all elastic network interfaces in your in-scope Amazon EC2 instances . You can still choose to only apply the policy to the default elastic network interface.	March 11, 2020
Updated AWS Managed Rules for AWS WAF	AWS Managed Rules for AWS WAF added an `AWSManagedRulesAnonymousIpList` rule group.	March 6, 2020
Updated AWS Managed Rules for AWS WAF	AWS Managed Rules for AWS WAF updated the WordPress application and `AWSManagedRulesCommonRuleSet` rule groups.	March 3, 2020
Added Amazon Route 53 health check to AWS Shield Advanced protection options	Shield Advanced now supports the use of Amazon Route 53 health check associations, to improve the accuracy of threat detection and mitigation.	February 14, 2020
Updated AWS Managed Rules for AWS WAF	AWS Managed Rules for AWS WAF has updated the SQL Database rule group to add checking the message URI.	January 23, 2020

Firewall Manager new option for security group usage audit policy	Firewall Manager has a new option for security group usage audit policies. You can now set a minimum number of minutes a security group must remain unused before it's considered noncompli ant. By default, this minutes setting is zero.	January 14, 2020
Firewall Manager new option for AWS WAF policy	Firewall Manager has a new option for AWS WAF policies. You can now choose to remove all existing web ACL associations from in-scope resources before associating the policy's new web ACLs to them.	January 14, 2020
Updated AWS Managed Rules for AWS WAF	AWS Managed Rules for AWS WAF has updated text transformations for rules in the Core Rule Set and the SQL Database rule groups.	December 20, 2019
AWS Firewall Manager integrated with AWS Security Hub CSPM	AWS Firewall Manager now creates findings for resources that are out of compliance and for attacks and sends them to AWS Security Hub CSPM.	December 18, 2019

Release of AWS WAF version 2	New version of the AWS WAF developer guide. You can manage a web ACL or rule group in JSON format. Expanded capabilities include logical rule statements, rule statement nesting, and full CIDR support for IP addresses and address ranges. Rules are no longer AWS resources , but exist only in the context of a web ACL or rule group. For existing customers, the prior version of the service is now called AWS WAF Classic. In the APIs, SDKs, and CLIs, AWS WAF Classic retains its naming schemes and this latest version of AWS WAF is referred to with an added "V2" or "v2", depending on the context. AWS WAF can't access AWS resources that were created in AWS WAF Classic. To use those resources in AWS WAF, you need to migrate them.	November 25, 2019
AWS Managed Rules rule groups for AWS WAF	Added AWS Managed Rules rule groups. These are free of charge for AWS WAF customers.	November 25, 2019
AWS Firewall Manager support for Amazon Virtual Private Cloud security groups	Added support for Amazon VPC security groups to Firewall Manager.	October 10, 2019

AWS Firewall Manager support for AWS Shield Advanced	Added support for Shield Advanced to Firewall Manager.	March 15, 2019
Tutorial: Creating hierarchical policies	Added tutorial on creating hierarchical policies in AWS Firewall Manager.	February 11, 2019
Rule-level control in rule groups	You can now exclude individual rules from AWS Marketplace rule groups, as well as your own rule groups.	December 12, 2018
AWS Shield Advanced support for AWS Global Accelerator standard accelerators	Shield Advanced can now protect AWS Global Accelerator standard accelerators.	November 26, 2018
AWS WAF support for Amazon API Gateway	AWS WAF now protects Amazon API Gateway APIs.	October 25, 2018
Expanded AWS shield advanced getting started wizard	New wizard provides opportunity to create rate-based rules and Amazon CloudWatch Events.	August 31, 2018
AWS WAF logging	Enable logging to get detailed information about traffic that is analyzed by your web ACL.	August 31, 2018
Support for query parameters in conditions	When creating a condition , you can now search the requests for specific parameters.	June 5, 2018
Shield advanced getting started wizard	Introduces a new streamlined process for subscribing to AWS Shield Advanced.	June 5, 2018

Expanded allowed CIDR ranges	When creating an IP match condition, AWS WAF now supports IPv4 address ranges: /8 and any range between /16 through /32.	June 5, 2018

Updates before 2018

The following table describes important changes in each release of the *AWS WAF Developer Guide* that were made before 2018.

Change	API Version	Description	Release Date
Update	2016-08-24	AWS Marketplace rule groups	November, 2017
Update	2016-08-24	Shield Advanced support for Elastic IP addresses	November, 2017
Update	2016-08-24	Global threat dashboard	November, 2017
Update	2016-08-24	DDoS-resistant website tutorial	October, 2017
Update	2016-08-24	Geo and regex conditions	October, 2017
Update	2016-08-24	Rate-based rules	June, 2017
Update	2016-08-24	Reorganization	April, 2017
Update	2016-08-24	Added information about DDOS protection and support for Application Load Balancers.	November, 2016

Change	API Version	Description	Release Date
New Features	2015-08-24	You can now log all your API calls to AWS WAF through AWS CloudTrail, the AWS service that records API calls for your account and delivers log files to your S3 bucket. CloudTrail logs can be used to enable security analysis, track changes to your AWS resources, and aid in compliance auditing. Integrating AWS WAF and CloudTrail lets you determine which requests were made to the AWS WAF API, the source IP address from which each request was made, who made the request, when it was made, and more. If you are already using AWS CloudTrail, you will start seeing AWS WAF API calls in your CloudTrail log. If you haven't enabled CloudTrail for your account, you can enable it on CloudTrail from the AWS Management Console. There is no additional charge for enabling CloudTrail, but standard rates for Amazon S3 and Amazon SNS usage apply.	April 28, 2016
New Features	2015-08-24	You can now use AWS WAF to allow, block, or count web requests that appear to contain malicious scripts, known as cross-site scripting or XSS. Attackers sometimes insert malicious scripts into web requests in an effort to exploit vulnerabilities in web applications. For more information, see Cross-site scripting attack rule statement.	March 29, 2016

Change	API Version	Description	Release Date
New Features	2015-08-24	With this release, AWS WAF adds the following features: • You can configure AWS WAF to allow, block, or count web requests based on the lengths of specified parts of the requests, such as query strings or URIs. For more information, see Size constraint rule statement. • You can configure AWS WAF to allow, block, or count web requests based on the content in the request body. This is the part of a request that contains any additional data that you want to send to your web server as the HTTP request body, such as data from a form. This feature applies to string match conditions, SQL injection match conditions, and the new size constraint conditions mentioned in the first bullet. For more information, see Adjusting rule statement settings in AWS WAF.	January 27, 2016
New Feature	2015-08-24	You can now use the AWS WAF console to choose the CloudFront distributions that you want to associate a web ACL with. For more information, see Associating or Disassociating a Web ACL and a CloudFront Distribution.	November 16, 2015
Initial Release	2015-08-24	This is the first release of the *AWS WAF Developer Guide*.	October 6, 2015