AWS WAF, AWS Firewall Manager, and AWS Shield Advanced

Developer Guide

API Version 2019-07-29
# Table of Contents

- What are AWS WAF, AWS shield, and AWS Firewall Manager? ................................................................. 1
  - AWS shield ................................................................................................................................. 2
  - AWS Firewall Manager ........................................................................................................... 2
  - Which should i choose? ........................................................................................................... 2

- Setting up ............................................................................................................................................ 3
  - Step 1: Sign up for an AWS account .................................................................................. 3
  - Step 2: Create an IAM user ................................................................................................. 3
  - Step 3: Download tools ........................................................................................................ 5

- AWS WAF ........................................................................................................................................ 6
  - How AWS WAF works ........................................................................................................ 7
    - AWS WAF Web ACL capacity units (WCU) ........................................................................ 7
    - AWS WAF pricing .............................................................................................................. 7
  - Getting started with AWS WAF .......................................................................................... 7
    - Step 1: Set up AWS WAF .................................................................................................. 8
    - Step 2: Create a Web ACL ............................................................................................... 8
    - Step 3: Add a string match rule ...................................................................................... 9
    - Step 4: Add an AWS Managed Rules rule group .......................................................... 10
    - Step 5: Finish your Web ACL configuration ................................................................... 10
    - Step 6: Clean up your resources ................................................................................... 11

- Tips for migrating your AWS WAF Classic resources to AWS WAF ...................................................... 11

- Managing and using a Web Access Control List (Web ACL) ............................................................ 12
  - How AWS WAF processes a Web ACL ................................................................................ 13
  - Working with web ACLs ........................................................................................................ 14

- Rule groups ....................................................................................................................................... 21
  - Managed rule groups ........................................................................................................... 21
  - Managing your own rule groups .......................................................................................... 29
  - Managing rule group behavior in a web ACL ........................................................................ 31

- Rules ............................................................................................................................................... 31
  - Rule name ............................................................................................................................. 32
  - Rule action ............................................................................................................................ 32
  - Rule statements ..................................................................................................................... 33

- IP sets and regex pattern sets ........................................................................................................... 46
  - Creating and managing an IP set ....................................................................................... 46
  - Creating and managing a regex pattern set ..................................................................... 48

- Logging Web ACL traffic information .............................................................................................. 50

- Listing IP addresses blocked by rate-based rules ............................................................................ 54

- How AWS WAF works with Amazon CloudFront features ............................................................. 54
  - Using AWS WAF with CloudFront custom error pages .................................................... 55
  - Using AWS WAF with CloudFront geo restriction ............................................................. 55
  - Using AWS WAF with CloudFront for applications running on your own HTTP server ....... 55

- Choosing the HTTP methods that CloudFront responds to ......................................................... 56

- Security .......................................................................................................................................... 56
  - Data protection ..................................................................................................................... 57
  - Identity and access management ......................................................................................... 58
  - Logging and monitoring ....................................................................................................... 72
  - Compliance validation .......................................................................................................... 73
  - Resilience ............................................................................................................................ 74
  - Infrastructure security .......................................................................................................... 74

- AWS WAF quotas ............................................................................................................................ 74

- AWS WAF Classic .......................................................................................................................... 76

- Setting up AWS WAF Classic ........................................................................................................ 76
  - Step 1: Sign up for an AWS account .................................................................................. 77
  - Step 2: Create an IAM user ............................................................................................... 77
What are AWS WAF, AWS shield, and AWS Firewall Manager?

AWS WAF 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 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.

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 CloudFront or an Application Load Balancer 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 the requests that match the properties that you specify** – When you want to allow or block requests based on new properties in web requests, you first can configure AWS WAF to count the requests that match those properties without allowing or blocking those requests. This lets you confirm that you didn’t accidentally configure AWS WAF to block all the traffic to your website. When you’re confident that you specified the correct properties, you can change the behavior to allow or block requests.

Using AWS WAF has several benefits:

- Additional protection against web attacks using conditions that you specify. You can define conditions by 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 string 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 conditions. Alternatively, rules can 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.
- 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.
AWS shield

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. AWS Shield Advanced provides expanded DDoS attack protection for your Amazon EC2 instances, Elastic Load Balancing load balancers, CloudFront distributions, and Route 53 hosted zones. AWS Shield Advanced incurs additional charges.

For more information about AWS Shield Standard and AWS Shield Advanced, see AWS Shield (p. 247).

AWS Firewall Manager

AWS Firewall Manager simplifies your administration and maintenance tasks across multiple accounts and resources for AWS WAF rules, AWS Shield Advanced protections, and Amazon VPC security groups. The Firewall Manager service automatically applies your rules and other security protections across your accounts and resources, even as you add new accounts and resources.

For more information about Firewall Manager, see AWS Firewall Manager (p. 200).

Which should I choose?

You can use AWS WAF (p. 6), AWS Firewall Manager (p. 200), and AWS Shield (p. 247) together to create a comprehensive security solution.

It all starts with AWS WAF. You can automate and then simplify AWS WAF management using AWS Firewall Manager. Shield Advanced adds additional features on top of AWS WAF, such as dedicated support from the DDoS Response Team (DRT) and advanced reporting.

If you want granular control over the protection that is added to your resources, AWS WAF alone is the right choice. If you want to use AWS WAF across accounts, accelerate your AWS WAF configuration, or automate protection of new resources, use Firewall Manager with AWS WAF.

Finally, if you own high visibility websites or are otherwise prone to frequent DDoS attacks, you should consider purchasing the additional features that Shield Advanced provides.

Note
To use the services of the DRT, you must be subscribed to the Business Support plan or the Enterprise Support plan.
Setting up

This topic describes preliminary steps, such as creating an AWS account, to prepare you to use AWS WAF, AWS Firewall Manager, and AWS Shield Advanced. You are not charged to set up this account and other preliminary items. You are charged only for AWS services that you use.

After you complete these steps, see Getting started with AWS WAF (p. 7) to continue getting started with AWS WAF.

Note
AWS Shield Standard is included with AWS WAF and does not require additional setup. For more information, see How AWS Shield works (p. 247).

Before you use AWS WAF or AWS Shield Advanced for the first time, complete the following tasks:

• Step 1: Sign up for an AWS account (p. 3)
• Step 2: Create an IAM user (p. 3)
• Step 3: Download tools (p. 5)

Step 1: Sign up for an AWS account

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including AWS WAF. You are charged only for the services that you use.

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To sign up for AWS

2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Note your AWS account number, because you'll need it for the next task.

Step 2: Create an IAM user

To use the AWS WAF console, you must sign in to confirm that you have permission to perform AWS WAF operations. You can use the root credentials for your AWS account, but we don't recommend it. For greater security and control of your account, we recommend that you use AWS Identity and Access Management (IAM) to do the following:

• Create an IAM user account for yourself or your business.
• Either add the IAM user account to an IAM group that has administrative permissions, or grant administrative permissions directly to the IAM user account.

You then can sign in to the AWS WAF console (and other service consoles) by using a special URL and the credentials for the IAM user. You also can add other users to the IAM user account, and control their level of access to AWS services and to your resources.
Step 2: Create an IAM user

Note
For information about creating access keys to access AWS WAF by using the AWS Command Line Interface (AWS CLI), Tools for Windows PowerShell, the AWS SDKs, or the AWS WAF API, see Managing Access Keys for IAM Users.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM console. If you aren't familiar with using the console, see Working with the AWS Management Console for an overview.

To create an administrator user for yourself and add the user to an administrators group (console)

1. Use your AWS account email address and password to sign in as the AWS account root user to the IAM console at https://console.aws.amazon.com/iam/.
   
   Note
   We strongly recommend that you adhere to the best practice of using the Administrator IAM user below and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.

2. In the navigation pane, choose Users and then choose Add user.

3. For User name, enter Administrator.

4. Select the check box next to AWS Management Console access. Then select Custom password, and then enter your new password in the text box.

5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to User must create a new password at next sign-in to allow the new user to reset their password after they sign in.

6. Choose Next: Permissions.

7. Under Set permissions, choose Add user to group.

8. Choose Create group.

9. In the Create group dialog box, for Group name enter Administrators.

10. Choose Filter policies, and then select AWS managed -job function to filter the table contents.

11. In the policy list, select the check box for AdministratorAccess. Then choose Create group.
   
   Note
   You must activate IAM user and role access to Billing before you can use the AdministratorAccess permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.

12. Back in the list of groups, select the check box for your new group. Choose Refresh if necessary to see the group in the list.

13. Choose Next: Tags.

14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM Entities in the IAM User Guide.

15. Choose Next: Review to see the list of group memberships to be added to the new user. When you are ready to proceed, choose Create user.

You can use this same process to create more groups and users and to give your users access to your AWS account resources. To learn about using policies that restrict user permissions to specific AWS resources, see Access Management and Example Policies.

To sign in as this new IAM user, first sign out of the AWS console. Then use the following URL, where your_aws_account_id is your AWS account number without the hyphens. For example, if your AWS account number is 1234-5678-9012, your AWS account ID is 123456789012:

API Version 2019-07-29
Enter the IAM user name and password that you just created. When you're signed in, the navigation bar displays "your_user_name @ your_aws_account_id".

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. From the IAM dashboard, choose Customize and enter an alias, such as your company name. To sign in after you create an account alias, use the following URL:

https://your_account_alias.signin.aws.amazon.com/console/

To verify the sign-in link for IAM users for your account, open the IAM console and check under the IAM users sign-in link on the dashboard.

After you complete these steps, you can stop here and go to Getting started with AWS WAF (p. 7) to continue getting started with AWS WAF using the console. If you want to access AWS WAF programmatically using the AWS WAF API, continue on to the next step, Step 3: Download tools (p. 5).

Step 3: Download tools

The AWS Management Console includes a console for AWS WAF, but if you want to access AWS WAF programmatically, the following documentation and tools will help you:

- If you want to call the AWS WAF 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 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 supports.

- The AWS Command Line Interface (AWS CLI) supports AWS WAF. 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. 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 an Amazon CloudFront distribution, an Amazon API Gateway API, or an Application Load Balancer.

AWS WAF 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, an Amazon CloudFront distribution, an Amazon API Gateway API, or an Application Load Balancer responds to requests either with the requested content or with an HTTP 403 status code (Forbidden). You can also configure CloudFront to return a custom error page when a request is blocked.

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
• How AWS WAF works (p. 6)
• Getting started with AWS WAF (p. 7)
• Tips for migrating your AWS WAF Classic resources to AWS WAF (p. 11)
• Managing and using a Web Access Control List (Web ACL) (p. 12)
• Rule groups (p. 21)
• AWS WAF rules (p. 31)
• IP sets and regex pattern sets (p. 46)
• Logging Web ACL traffic information (p. 50)
• Listing IP addresses blocked by rate-based rules (p. 54)
• How AWS WAF works with Amazon CloudFront features (p. 54)
• Security in AWS WAF (p. 56)
• AWS WAF quotas (p. 74)

How AWS WAF works

You use AWS WAF to control how an Amazon CloudFront distribution, an Amazon API Gateway API, or an Application Load Balancer responds to web requests.

• Web ACLs – You use a web access control list (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 specify how to handle requests that match the criteria. You set a default action for the web ACL that indicates whether to block or allow through those requests that pass the rules inspections.

• 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 use rules to block matching requests or to allow matching requests through. You can also use rules just to count matching requests.
• **Rules groups** – You can use rules individually 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.

After you create your web ACL, you can associate it with one or more AWS resources. The resource types that you can protect using AWS WAF web ACLs are Amazon CloudFront distributions, Amazon API Gateway APIs, and Application Load Balancers.

AWS WAF is available in the Regions listed at AWS Regions and Endpoints.

• For an API Gateway API or an Application Load Balancer, you can use any of the Regions in the list.
• For a CloudFront distribution, AWS WAF is available globally, but when you create AWS resources for use with CloudFront, you must choose the Region US East (N. Virginia). AWS WAF resources are web ACLs, rule groups, IP sets, and regex pattern sets.

**AWS WAF Web ACL capacity units (WCU)**

AWS WAF uses web ACL capacity units (WCU) to calculate and control the operating resources that are used to run your rules, rule groups, and web ACLs. 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.

AWS WAF manages capacity for rules, rule groups, and web ACLs:

• **Rule capacity** – AWS WAF calculates rule capacity when you create or update a rule. For some basic guidelines for rule capacity requirements, see the listings for the various rule statements at AWS WAF rule statements (p. 33). You can also get an idea of the capacity required for the various rule types in the AWS WAF console by creating a web ACL or rule group and adding individual rules to it. The console displays the capacity units used as you add the rules.

• **Rule group capacity** – AWS WAF requires that each rule group is assigned an immutable capacity 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 WCU within its max quote. This ensures that web ACLs that are using the rule group remain within their quotas.

• **Web ACL capacity** – The maximum capacity for a web ACL is 1,500, which is sufficient for most use cases. If you need more capacity, contact the AWS Support Center.

**AWS WAF pricing**

With AWS WAF, you pay only for the web ACLs and rule groups that you create, and for the number of HTTP(S) requests that AWS WAF inspects. For more information, see AWS WAF Pricing.

**Getting started with 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.
• Choose the AWS resources that you want AWS WAF to inspect web requests for. This tutorial covers the steps for Amazon CloudFront. The process is essentially the same for an Application Load Balancer or Amazon API Gateway API.
Step 1: Set up AWS WAF

- Add the rules and rule groups 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. For each rule, you specify whether you want to block matching web requests or allow them. The rules that are defined inside a rule group have their actions defined inside the rule group.
- Specify a default action for the web ACL, either block or allow. This is the action that AWS WAF takes when a web request doesn't match any of the rules.

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 (p. 8)
- Step 2: Create a Web ACL (p. 8)
- Step 3: Add a string match rule (p. 9)
- Step 4: Add an AWS Managed Rules rule group (p. 10)
- Step 5: Finish your Web ACL configuration (p. 10)
- Step 6: Clean up your resources (p. 11)

Step 1: Set up AWS WAF

If you already signed up for an AWS account and created an IAM user as described in Setting up (p. 3), go to Step 2: Create a Web ACL (p. 8).

If not, go to Setting up (p. 3) and perform at least the first two steps. (You can skip downloading tools for now because this Getting Started topic focuses on using the AWS WAF console.)

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 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://console.aws.amazon.com/wafv2/.
2. If this is your first time using AWS WAF, choose Go to AWS WAF, and then choose Configure web ACL.
   If you've used AWS WAF before, 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. 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**

A string match rule statement identifies strings that you want AWS WAF 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 rule with a string match statement and indicate what to do with matching requests.

**Note**
For more information about string match rule statements, see **String match rule statement (p. 41)**.

**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 use the logical statement types for rules, which allow you to combine or negate rule statement results.

5. On **Statement**, for **Inspect**, open the dropdown and choose the web request component that you want AWS WAF to look for your string in. For this example, choose **Header**.

   When you choose **Header**, you also specify which header you want AWS WAF to inspect. Enter **User-Agent**. (This value isn’t case sensitive.)

   **Note**
   If you choose to inspect the web request **Body**, AWS WAF inspects only the first 8192 bytes (8 KB), because the underlying host service 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 gets the length of the body from the request headers. For more information, see **Size constraint rule statement (p. 40)**.

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 **BadBot**. AWS WAF will inspect the **User-Agent** header in web requests for the value **BadBot**.
8. Leave **Text transformation** set to None.

In an effort to bypass AWS WAF, 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 specify multiple transformations. AWS WAF processes them all in order before inspecting the web request component.

9. For **Action**, select the action you want the rule to take when it matches a web request. For this example, choose **Count**. This creates metrics for web requests that match the rule, but doesn't affect whether the rule is allowed or blocked. For information on your choices, see AWS WAF rule action (p. 32) and How AWS WAF processes a Web ACL (p. 13).

10. Choose **Add rule**.

---

**Step 4: Add an AWS Managed Rules rule group**

AWS Managed Rules offers a set of managed rule groups for your use, free of charge to AWS WAF customers. For more information about rule groups, see Rule groups (p. 21). 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, turn on the Add to web ACL toggle in the Action column. Also turn on the Set rules action to count toggle. This sets the action for all rules in the rule group to count only. This allows you to see how the rule group behaves with your web requests before you put it to use.
4. Choose Add rules
5. Choose Next.

---

**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. You can change the processing order by moving them up and down. To do this, select one in the list and choose Move up or Move down.
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. Deselect any you don't want metrics for. As needed, change the names of the ones you want metrics for. For more information about Amazon CloudWatch metrics, see Monitoring with Amazon CloudWatch (p. 282).
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 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 charges for

1. In the **Web ACL** page, select your web ACL from the list and choose **Edit**.
2. On **Associated AWS resources - optional**, select all associated resources, and then choose **Remove**. 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.

## Tips for migrating your AWS WAF Classic resources to AWS WAF

**Note**

This is the latest version of **AWS WAF**. If you created AWS WAF resources, like rules and web ACLs, prior to November, 2019, you either need to use AWS WAF Classic to work with those resources, or you need to migrate them to this latest version.

This section provides high-level guidance for migrating your rules and web ACLs from AWS WAF Classic to AWS WAF.

### Before you begin

Understand the options and expanded capabilities of this latest version of AWS WAF by reading the information in this AWS WAF developer guide. For example, your options for rules management now include logical AND, OR, and NOT statements and you can nest rules. AWS WAF support full CIDR notation in IP address specifications.

Additionally, AWS WAF now provides a set of rule groups for your use, free of charge. You might want to use these in your new web ACL configurations, in place of some of your prior settings. For information, see [Managed rule groups](#) (p. 21).

### General steps to migrate your web ACLs from AWS WAF Classic to AWS WAF

For each web ACL that you use in AWS WAF Classic, do the following:
1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/

Choose Switch to AWS WAF Classic and review your configuration settings for the web ACL. Make note of the settings that you want to migrate for the web ACL. This is a manual process.

2. In another browser session, follow the guidance at Creating a web ACL (p. 15) to create a web ACL in AWS WAF that's based on your AWS WAF Classic web ACL.

3. In AWS WAF Classic, disassociate old web ACL from your AWS resources.

4. In AWS WAF, associate the new web ACL with your AWS resources.

Managing and using a Web Access Control List (Web ACL)

A web access control list (web ACL) gives you fine-grained control over the web requests that your Amazon CloudFront distribution, Amazon API Gateway API, or Application Load Balancer responds to.

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 any 5-minute period. You can combine conditions using logical operators.

This criteria is provided inside the rules that you include in your web ACL and in rule groups that you use in the web ACL. It's specified in the rule statement. For a full list of the options, see AWS WAF rule statements (p. 33).

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, either allow or block, for web requests that don't match any of the rules that you specify. For more information, see Deciding on the default action for a Web ACL (p. 13).

2. Add any rule groups that you want to use in your web ACL. Managed rule groups usually contain rules that block web requests. For information about rule groups, see Rule groups (p. 21).

3. Specify additional conditions under which you want to allow or block requests in one or more rules. To add more than one, start with AND or OR rule statements and nest the rules that you want to combine under those. If you want to negate a rule option, nest the rule in a NOT statement. You can optionally use a rate-based rule instead of a regular rule to limit the number of requests from any single IP address that meets the conditions. For information about rules, see AWS WAF rules (p. 31).

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 How AWS WAF processes a Web ACL (p. 13).

Topics
- How AWS WAF processes a Web ACL (p. 13)
- Working with web ACLs (p. 14)
How AWS WAF processes a Web ACL

The way a web ACL handles a web request depends on the following:

- The action settings on the rules and web ACL
- Any overrides that you place on the rules and rule groups that you add
- The ordering of the rules and rule groups

If you add more than one rule to a web ACL, AWS WAF evaluates each request against the rules in the order that you list them in the web ACL. If you add a rule group to your web ACL, AWS WAF processes the rule group in the order that it's listed in the web ACL and processes the rules in the rule group in the order that they're listed inside that.

How AWS WAF handles the rule actions in a Web ACL

For the rules in a web ACL, you can choose between counting, allowing, or blocking matching web requests:

- Allow and block actions are terminating actions. They stop all other processing of the web ACL on the matching web request. If you include a rule in a web ACL that has an allow or block action and the rule finds a match, that match determines the final disposition of the web request for the web ACL. AWS WAF doesn't process any other rules in the web ACL that come after the matching one. This is true for rules that you add directly to the web ACL and rules that are inside an added rule group.
- The count action 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 web ACL rule set. If the only rules that match have count action set, AWS WAF applies the web ACL default action setting.

The actions that AWS WAF applies to a web request are affected by the relative position of rules in the web ACL. For example, if a web request matches a rule that allows requests and matches another rule that counts requests, if the rule that allows requests is listed first, then AWS WAF doesn't count the request.

Deciding on the default action for a Web ACL

When you create and configure a web ACL, you set the web ACL default action, which determines how AWS WAF handles web requests that don't match any rules in the web ACL:

- **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 web ACL, add rules that identify and block the specific requests that you want to block.
- **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 web ACL, add rules that identify and allow the specific requests that you want to allow in.

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 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 groups usually use the block action. For information about managed rule groups, see Managed rule groups (p. 21).

Overriding the actions of an entire rule group

For each rule group in a web ACL, you can override the contained rule's actions, setting the action for all to count. If any of the rule actions in the group is set to block or allow, this override changes the behavior so that matching rules are only counted. In cases where the rule group might normally determine to allow or block a request, it doesn't, and the web ACL continues processing any rules that follow.

If you want to test a rule before you start using it to allow or block requests, you can configure AWS WAF to count the web requests that match the conditions in the rule. If you have metrics enabled, you'll receive COUNT metrics for all the rules in the group. For more information, see Testing web ACLs (p. 18).

Excluding a rule in a rule group

For each rule group, you can exclude individual rules. This effectively overrides the rule's action to count, so the effect on the rule is the same as the effect of Override to count for all rules in the group. In cases where the rule might normally determine to allow or block a request, it doesn't, and the web ACL continues processing any rules that follow.

Excluding a single rule can help you troubleshoot false positives, which is when a rule group blocks traffic that you aren't expecting it to block. You can identify the rule within the rule group that's doing the unexpected blocking and then disable it by excluding it. Excluding a rule overrides the action that's set on the rule to count. If you have metrics enabled, you'll receive COUNT metrics for each excluded rule.

How AWS resources handle response delays from AWS WAF

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 Amazon API Gateway and Application Load Balancer typically deny the request and don't serve the content.

Working with web ACLs

When you make changes to web ACLs or web ACL components, like rules and rule groups, AWS WAF propagates the changes everywhere that the web ACL and its components are stored and used. Your changes are applied within seconds, but there might be a brief period of inconsistency when the changes have arrived in some places and not in others. So, for example, if you add an IP address to an IP set that's referenced by a blocking rule in a web ACL, the new address might briefly be blocked in one area while still allowed in another. This temporary inconsistency can occur when you first associate a web ACL with an AWS resource and when you change a web ACL that is already associated with a resource. Generally, any inconsistencies of this type last only a few seconds.

Topics
• Creating a web ACL (p. 15)
• Associating or disassociating a Web ACL with an AWS resource (p. 16)
• Editing a Web ACL (p. 17)
• Deleting a Web ACL (p. 17)
• Testing web ACLs (p. 18)
Creating a web ACL

To create a web ACL

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.
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. For Resource type, choose the category of AWS resource that you want to associate with this web ACL. For more information, see Associating or disassociating a Web ACL with an AWS resource (p. 16).
7. For Region, choose the Region where you want AWS WAF to store the web ACL.
8. (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.
9. Choose Next.
10. (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, turn on the Add to web ACL toggle in the Action column.

    If you want to set the action for all rules in the rule group to count only, turn on the Set rules action to count toggle. For information about this option, see Overriding the actions of an entire rule group (p. 14).

    Choose Add rules to finish adding managed rules and return to the Add rules and rule groups page.
11. (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. Choose your rule group from the list. If you want to override the rule actions for the group, turn on the Set rules action to count toggle. For information about this option, see Overriding the actions of an entire rule group (p. 14).

    12. (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 (p. 31).
This procedure covers the Rule visual editor.

a. For Name, enter the name that you want to use to identify this rule.
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 (p. 31).
c. For Action, select the action you want the rule to take when it matches a web request. For information on your choices, see AWS WAF rule action (p. 32) and How AWS WAF processes a Web ACL (p. 13).
d. Choose Add rule.

13. Choose 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. For more information, see Deciding on the default action for a Web ACL (p. 13).


15. 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. For more information, see How AWS WAF processes a Web ACL (p. 13).

16. Choose Next.

17. In the Configure metrics page, deselect anything you don't want metrics for, and update names as needed. You can combine metrics from multiple sources by providing the same CloudWatch metric name.

18. Choose Next.

19. 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.

20. Choose Create web ACL. Your new web ACL is listed in the Web ACLs page.

Associating or disassociating a Web ACL with an AWS resource

You can use AWS WAF to associate a web ACL with the following AWS regional resource types:

- Application Load Balancer
- Amazon API Gateway API

You can associate a 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.

You can only associate a web ACL to an Application Load Balancer within AWS Regions. For example, you can't associate a web ACL to an Application Load Balancer that is on AWS Outposts.

Restrictions on multiple associations

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

- You can associate each AWS resource with only one web ACL. The relationship between web ACL and AWS resources is one-to-many.
- For a single Region and for use with CloudFront distributions, you can't mix resource type associations:
  - You can associate a web ACL with one or more CloudFront distributions. You can't associate a web ACL that you've associated with a CloudFront distribution with any other AWS resource type.
• In a single Region, you can associate a web ACL exclusively with Application Load Balancer resources or exclusively with API Gateway API resources. You can't mix the two resource type associations in one Region.

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://console.aws.amazon.com/wafv2/.
2. In the navigation pane, choose Web ACLs.
3. Choose the web ACL that you want to associate with a resource.
4. On the Associated AWS resources tab, choose Add AWS resources.
5. When prompted, choose your resource that you want to associate this web ACL with. If you choose an Application Load Balancer or API Gateway API, specify a Region.
6. Choose Add.

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://console.aws.amazon.com/wafv2/.
2. In the navigation pane, choose Web ACLs.
3. Choose the web ACL that you want to disassociate from your resource.
4. On the Associated AWS resources tab, deselect the resources that you want to disassociate this web ACL from.
5. Choose Save.

Editing a Web ACL

To add or remove rules from a web ACL or change the default action, access the web ACL using the following procedure:

To edit a web ACL

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.
2. In the navigation pane, choose Web ACLs.
3. Choose the web ACL that you want to edit, and then choose Edit.
4. Page through the web ACL definitions, and make your changes. This is the same as the procedure that you use to create the web ACL in Creating a web ACL (p. 15), but with some fields not modifiable. For example, you can't change the name of a web ACL.

Deleting a Web ACL

To delete a web ACL, you first disassociate all AWS resources 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://console.aws.amazon.com/wafv2/.
2. In the navigation pane, choose **Web ACLs**.
3. Select the web ACL that you want to delete, and then choose **Edit**.
4. Disassociate the web ACL from all AWS resources. On the **Associated AWS resources** tab, select all resources, and then choose **Remove**.
5. On the **Web ACLs** page, select the web ACL that you want to delete, and then choose **Delete**.

### Testing web ACLs

To ensure that you don’t accidentally configure AWS WAF 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 (p. 18)
- Viewing a sample of web requests (p. 19)

### 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 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 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 increments a counter for that rule. Then AWS WAF inspects the web request based on the conditions in the next rule. If the request matches all the conditions in that rule, AWS WAF increments a counter for the rule. This continues until AWS WAF 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 one or more AWS resources (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 an associated resource forwards to AWS WAF for a 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 console. If you choose another region, no AWS WAF metrics will appear in the CloudWatch console.

**To view data for the rules in a web ACL**

2. In the navigation pane, under **Metrics**, choose **AWS/WAFV2**.
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 recently associated a 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 web ACL to appear in the list of available metrics.
- If you associate more than one resource with a web ACL, the CloudWatch data will include requests for all of them.
- 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 an associated AWS resource has forwarded to AWS WAF. For more information, see Viewing a sample of web requests (p. 19).

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 Managing and using a Web Access Control List (Web ACL) (p. 12).

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 (p. 17).

**Viewing a sample of web requests**

In the AWS WAF console, if you have request sampling enabled, you can view a sample of the requests that an associated resource has forwarded to AWS WAF 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 an associated resource has forwarded to AWS WAF**

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.
2. In the navigation pane, choose **Web ACLs**
3. Choose the web ACL for which you want to view requests.

4. In the **Overview** 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 part of a URL that identifies a resource, for example, /images/daily-ad.jpg.

   **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 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 received the request from API Gateway, CloudFront or your Application Load Balancer.

5. To display additional information about the request, choose the arrow on the left side of the IP address for that request. AWS WAF 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.

6. To refresh the list of sample requests, choose **Get new samples**.
Rule groups

A rule group is a reusable set of rules that you can add to a web ACL. For more information about web ACLs, see Managing and using a Web Access Control List (Web ACL) (p. 12).

Rule groups fall into two main categories:

• Managed rule groups, which AWS Managed Rules and AWS Marketplace sellers create and maintain for you
• Your own rule groups, which you create and maintain

Differences between rule groups and web ACLs

Rule groups and web ACLs both contain rules, which are defined in the same manner in both places. Rule groups differ from web ACLs in the following ways:

• Rule groups can't contain the following rule statement types:
  • Rule group reference statements
  • Rate-based rule statements
• You can reuse a single rule group in multiple web ACLs by adding a rule group reference statement to each web ACL. You can't reuse a web ACL.
• Rule groups don't have default actions. In a web ACL, you set a default action for each rule or rule group that you include. Each individual rule inside a rule group or 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 web ACL.
• Web ACLs have a system-defined maximum capacity of 1,500 web ACL capacity units (WCUs). Every 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 web ACL.

For information about rules, see AWS WAF rules (p. 31).

This section describes the types of managed rule groups that are available to you and provides guidance for creating and managing your own rule groups, if you choose to do so.

Topics

• Managed rule groups (p. 21)
• Managing your own rule groups (p. 29)
• Managing rule group behavior in a web ACL (p. 31)

Managed rule groups

Managed rule groups are collections of predefined, ready-to-use rules that AWS and AWS Marketplace sellers write and maintain for you:

• AWS Managed Rules rule groups are available for free to AWS WAF customers.
• AWS Marketplace managed rule groups are available by subscription through AWS Marketplace.

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, such as those 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. AWS and AWS Marketplace sellers automatically update managed rule groups when new vulnerabilities and threats emerge.

AWS and many of the AWS Marketplace sellers are notified of new vulnerabilities before public disclosure. AWS WAF 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 a Managed 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.

You can't view individual rules in a managed rule group, and you can't edit them. However, you can exclude specific rules from a rule group when you add it to your web ACL. You can also override all rule actions in the group to COUNT. For more information, see the following section and also see the steps for adding a managed rule group in the procedure Creating a web ACL (p. 15).

**Topics**

- Working with managed rule groups (p. 22)
- AWS Managed Rules for AWS WAF (p. 24)
- AWS Marketplace managed rule groups (p. 28)

**Working with managed rule groups**

This section provides guidance for accessing and managing managed rule groups.

**To retrieve a 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 vendor names are listed. Expand each vendor listing to see the list of managed rule groups. When you add a managed rule group to your 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

The API and CLI calls return a list of all managed rules that you can reference in the JSON model or through AWS CloudFormation.

**To retrieve the list of rules in a managed rule group**

- **Console**
  1. Add the managed rule group into your web ACL and finish creating your web ACL. For guidance, see the section called "Creating a web ACL" (p. 15).
  2. From the Web ACLs page, choose the web ACL you just created. This takes you to the web ACL edit page.
3. Choose Rules.
4. 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.

In this page, you can optionally put individual rules into count mode by selecting Override rules action.

- API – DescribeManagedRuleGroup
- CLI – `aws wafv2 describe-managed-rule-group --scope=REGIONAL --vendor-name=<vendor> --name=<managedrule_name>`

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.

To add or modify managed rule groups using 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 ExcludedRules specification lists rules whose actions are overridden to count only.

```
{
    "Name": "AWS-AWSManagedRulesCommonRuleSet",
    "Priority": 0,
    "Statement": {
        "ManagedRuleGroupStatement": {
            "VendorName": "AWS",
            "Name": "AWSManagedRulesCommonRuleSet",
            "ExcludedRules": [
                { "Name": "NoUserAgent_HEADER"
            ]
        }
    },
    "OverrideAction": {
        "None": {}
    },
    "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "AWS-AWSManagedRulesCommonRuleSet"
    }
}
```

To add or modify managed rule groups using AWS CloudFormation

You can reference and modify managed rule groups within a rule statement using the AWS CloudFormation template. The following listing shows the AWS Managed Rules rule group, `AWSManagedRulesCommonRuleSet`, in AWS CloudFormation template. The ExcludedRules specification lists rules whose actions are overridden to count only.

```
Description: WebACL With AMR
Resources:
WebACLWithAMR:
    Type: AWS::WAFV2::WebACL
    Properties:
        Name: WebACLWithAMR
        Scope: REGIONAL
        Description: This is a demo
        DefaultAction:
```

API Version 2019-07-29
AWS Managed Rules for AWS WAF

AWS Managed Rules for AWS WAF is a managed service that provides protection against common application vulnerabilities or other unwanted traffic, without having to write your own rules. You have the option of selecting one or more rule groups from AWS Managed Rules for each web ACL, up to the allowed maximum web ACL capacity unit (WCU) limit. You can choose whether to count (monitor) or block requests that are matched by the managed rules.

As a best practice, before using a rule group in production, test it in a non-production environment, with the action override set to count. Evaluate the rule group using Amazon CloudWatch metrics combined with AWS WAF sampled requests or AWS WAF logs. When you're satisfied that the rule group does what you want, remove the override on the group.

Mitigating False Positive Scenarios

If you are encountering false-positive scenarios with AWS Managed Rules rule groups, perform the following steps:

1. In the web ACL configuration, override the actions in the rules of the rule groups, putting them into count (alert) mode. This stops them from blocking legitimate traffic.

2. Use either AWS WAF sampled requests or AWS WAF logs to identify which AWS Managed Rules rule group is triggering the false positive. You can identify the AWS Managed Rules rule group by looking at the ruleGroupId field in the log or the RuleWithinRuleGroup in the sampled request. The rule name follows this pattern: AWS#<AMR RuleGroup Name>#<AMR Rule Name>.

3. On the AWS WAF console, edit the web ACL, locate the AWS Managed Rules rule group that you've identified, and disable the rule that is causing the false positive.

For more information about a rule in an AWS Managed Rules rule group, contact the AWS Support Center.

AWS Managed Rules rule groups list

This section describes the AWS Managed Rules rule groups that are currently available. You see these on the console when you add a managed rule group to your web ACL. Through the API, you can retrieve
this list along with the AWS Marketplace managed rule groups that you’re subscribed to by calling 
ListAvailableManagedRuleGroups.

Topics

- Baseline rule groups (p. 25)
- Use-case specific rule groups (p. 25)
- IP reputation rule groups (p. 26)

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)

VendorName: AWS, Name: AWSManagedRulesCommonRuleSet, WCU: 700

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 those described in OWASP publications. Consider using this rule group for any AWS WAF use case.

Admin Protection

VendorName: AWS, Name: AWSManagedRulesAdminProtectionRuleSet, WCU: 100

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 would like to reduce the risk of a malicious actor gaining administrative access to your application.

Known Bad Inputs

VendorName: AWS, Name: AWSManagedRulesKnownBadInputsRuleSet, WCU: 200

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.

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

VendorName: AWS, Name: AWSManagedRulesSQLiRuleSet, WCU: 200

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.

LINUX operating system

VendorName: AWS, Name: AWSManagedRulesLinuxRuleSet, WCU: 200

The Linux Operating System rule group contains rules that block request patterns associated with exploitation of vulnerabilities specific to Linux, including Linux-specific Local File Inclusion (LFI) attacks.
This can help prevent attacks that expose file contents or execute 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.

**POSIX Operating System**

VendorName: AWS, Name: AWSManagedRulesUnixRuleSet, WCU: 100

The POSIX Operating System rule group contains rules that block request patterns associated with 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 execute 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, OpenBSD, and many others.

**Windows Operating System**

VendorName: AWS, Name: AWSManagedRulesWindowsRuleSet, WCU: 200

The Windows Operating System rule group contains rules that block request patterns associated with exploitation of vulnerabilities specific to Windows, like remote execution of PowerShell commands. This can help prevent exploitation of vulnerabilities that allow an attacker to run unauthorized commands or execute malicious code. Evaluate this rule group if any part of your application runs on a Windows operating system.

**PHP Application**

VendorName: AWS, Name: AWSManagedRulesPHPRuleSet, WCU: 100

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 allow an attacker to remotely execute 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.

**WordPress Application**

VendorName: AWS, Name: AWSManagedRulesWordPressRuleSet, WCU: 100

The WordPress Applications 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.

**IP reputation rule groups**

IP reputation rule groups allow you to block requests based on their source. Choose one or more of these rule groups if you would like to reduce your exposure to bot traffic, exploitation attempts, or if you are enforcing geographic restrictions on your content.

**Amazon IP Reputation**

VendorName: AWS, Name: AWSManagedRulesAmazonIpReputationList, WCU: 25

This 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.
Anonymous IP List

VendorName: AWS, Name: AWSManagedRulesAnonymousIpList, WCU: 50

This rule group contains rules to block requests from services that allow the obfuscation of viewer identity. These include requests from VPNs, proxies, Tor nodes, and 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.

AWS Managed Rules disclaimer

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 changelog

This section lists changes to the AWS Managed Rules for AWS WAF since their release in November, 2019.

<table>
<thead>
<tr>
<th>Rule group</th>
<th>Rules affected</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anonymous IP List</td>
<td>All</td>
<td>New rule group in IP reputation rule groups (p. 26) to block requests from services that allow the obfuscation of viewer identity, to help mitigate bots and evasion of geographic restrictions.</td>
<td>March 06, 2020</td>
</tr>
<tr>
<td>Wordpress application</td>
<td>• WordPressExploitableCommands_QUERYSTRING</td>
<td>New rule that checks for exploitable commands in the query string.</td>
<td>March 03, 2020</td>
</tr>
<tr>
<td>Core Rule Set (CRS)</td>
<td>• SizeRestrictions_QUERYSTRING • SizeRestrictions_Cookie • SizeRestrictions_BODY • SizeRestrictions_URIPATH</td>
<td>Adjusted the size value constraints for improved accuracy.</td>
<td>March 03, 2020</td>
</tr>
<tr>
<td>SQL database</td>
<td>• SQLi_URIPATH</td>
<td>The rules now check the message URI.</td>
<td>January 23, 2020</td>
</tr>
<tr>
<td>SQL database</td>
<td>• SQLi_BODY • SQLi_QUERYARGUMENTS • SQLi_COOKIE</td>
<td>Updated text transformations.</td>
<td>December 20, 2019</td>
</tr>
<tr>
<td>Core Rule Set (CRS)</td>
<td>• CrossSiteScripting_URIPATH • CrossSiteScripting_BODY • CrossSiteScripting_QUERYARGUMENTS • CrossSiteScripting_COOKIE</td>
<td>Updated text transformations.</td>
<td>December 20, 2019</td>
</tr>
</tbody>
</table>
AWS Marketplace managed rule groups

AWS Marketplace managed rule groups are available by subscription through the AWS Marketplace console at AWS Marketplace. After you subscribe to a AWS Marketplace managed rule group, you can use it in AWS WAF.

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. After all accounts have subscribed, you can then add the rule group to a policy.

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 are charged a monthly fee (prorated hourly) and ongoing request fees based on volume. For more information, see AWS WAF Pricing and the description for each AWS Marketplace rule group at AWS Marketplace.

Subscribing to AWS Marketplace managed rule groups

You can subscribe to and unsubscribe from AWS Marketplace rule groups on the AWS WAF console. If you need to, you can exclude specific rules in a managed rule group when you add it to a web ACL.

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. After all accounts have subscribed, you can then add the rule group to a policy.

To subscribe to an AWS Marketplace managed rule group

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.
2. In the navigation pane, choose AWS 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, similar to how you add an individual rule. For more information, see Creating a web ACL (p. 15) or Editing a Web ACL (p. 17).
   
   Note
   When adding a rule group to a web ACL, you can override the actions of all rules in the rule group to COUNT only. For more information, see Overriding rule actions for a rule group (p. 31).

To unsubscribe from an AWS Marketplace managed rule group

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.
2. Remove the rule group from all web ACLs. For more information, see Editing a Web ACL (p. 17).
3. In the navigation pane, choose AWS 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**.

After you're subscribed to an AWS Marketplace rule group, you use it in your web ACLs as you do other managed rule groups. For information, see Creating a web ACL (p. 15).

**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. Exclude the specific rules that are blocking legitimate traffic. You can identify which rules are blocking which 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>#<RuleGroupName>#<RuleName>`.

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 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 AWS Marketplace seller, contact the vendor's customer support team. To find vendor contact information, see the vendors's listing on AWS Marketplace.

**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. When you create your own rule group, you must set an immutable maximum capacity for it.

Rule groups that you create hold rules just like a web ACL does, and you add rules to a rule group in the same way as you do to a web ACL.

**Topics**

- Creating a rule group (p. 29)
- Using your rule group in a Web ACL (p. 30)
- Deleting a rule group (p. 30)

**Creating a rule group**

**To create a rule group**

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.
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 set to manage it and use it.
   
   **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 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 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 web ACL capacity units (WCUs). This is an immutable setting. For information about WCUs, see the section called “AWS WAF Web ACL capacity units (WCU)” (p. 7).

   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**.

### Using your rule group in a Web ACL

To use a rule group in a web ACL, on the console, when you add or update the rules in your web ACL, on the **Add rules and rule groups** page, choose **Add rules**, and then choose **Add my own rules and rule groups**. Then choose **Rule group** and select your rule group from the list.

**Eventual consistency**

When you make changes to web ACLs or web ACL components, like rules and rule groups, AWS WAF propagates the changes everywhere that the web ACL and its components are stored and used. Your changes are applied within seconds, but there might be a brief period of inconsistency when the changes have arrived in some places and not in others. So, for example, if you add an IP address to an IP set that's referenced by a blocking rule in a web ACL, the new address might briefly be blocked in one area while still allowed in another. This temporary inconsistency can occur when you first associate a web ACL with an AWS resource and when you change a web ACL that is already associated with a resource. Generally, any inconsistencies of this type last only a few seconds.

### 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 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 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 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 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**

2. In the navigation pane, choose **Rule groups**.
3. Choose the rule group that you want to delete, and then choose **Delete**.
Managing rule group behavior in a web ACL

This section describes your options for modifying how you use a rule group in your web ACL. This information applies to all rule group types.

Overriding rule actions for a rule group

For each rule group in a web ACL, you can override the contained rule's actions by setting the override action for the rule group to Override to count. If any of the rule actions in the group is set to block or allow, this override changes the behavior to only count. For more information about this option, see How AWS WAF processes a Web ACL (p. 13).

To override the rule actions in a rule group, when you add the rule group to the web ACL, turn on the Set rules action to count toggle. For more information on web ACL management, see Managing and using a Web Access Control List (Web ACL) (p. 12).

Excluding a rule in a rule group

Excluding a single rule can help you troubleshoot false positives, which is when a rule group blocks traffic that you aren't expecting it to block. You can identify the rule within the rule group that's doing the unexpected blocking, and then disable it by excluding it. Excluding a rule overrides the action that's set on the rule to count. If you have metrics enabled, you'll receive COUNT metrics for each excluded rule. For more information about this option, see How AWS WAF processes a Web ACL (p. 13).

To exclude a rule in a rule group

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.
2. If not already enabled, enable AWS WAF logging. For more information, see Logging Web ACL traffic information (p. 50). Use the AWS WAF 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 web ACL that you want to edit. This takes you to a page showing the web ACL overview, with other tabs available.
   - **Note**
     You must associate a rule group with the web ACL and save the web ACL before you can exclude a rule from that rule group.
5. Choose the Rules tab.
6. 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.
7. Select Override rules action for the rules you want to exclude from the rule group. AWS WAF will count matching web requests for the rules that you exclude, and take no other action. You receive count metrics for each rule in this state, if metrics are enabled for the rule group.

AWS WAF rules

In every rule group and every web ACL, rules define how to inspect web requests and what to do when a web request matches the inspection criteria. Each rule requires one top-level statement, which might contain nested statements at any depth, depending on the rule and statement type.

The inspection instructions are included in the JSON format as rule statements and the action in rule actions.
You use the rules in a web ACL to block or allow web requests based on 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 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.

Some rule types take multiple values. For example, you can specify up to 10,000 IP addresses or IP address ranges in an IP address rule.

In addition to statements like the ones in the preceding list, which provide web request inspection criteria, 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, all the statements need to result in a match for the top-level AND statement to match.

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 web ACL where it's defined. You can manage rules and copy them to other web ACLs by using the JSON format of the rule group or web ACL that contains the rule. Or you can manage them through the AWS WAF console Rule Builder, which is available for web ACLs and rule groups.

Topics
- AWS WAF rule name (p. 32)
- AWS WAF rule action (p. 32)
- AWS WAF rule statements (p. 33)

AWS WAF rule name

You must assign a unique name to every rule in your web ACL or rule group.

The name can contain only the characters A–Z, a–z, 0–9, and the following special characters: _-!"#`+*},./. You can’t change the name of a rule after you create it in your rule group or web ACL.

AWS WAF rule action

The rule action tells AWS WAF what to do with a web request when it matches the criteria defined in the rule.
Here are the rule action options:

- **Count** – AWS WAF counts the request but doesn't determine whether to allow it or block it. With this action, AWS WAF continues processing the remaining rules in the web ACL.
- **Allow** – AWS WAF allows the request to be forwarded to the AWS resource for processing and response.
- **Block** – AWS WAF blocks the request and the AWS resource responds with an HTTP 403 (Forbidden) status code.

You can override rule actions when you add them to a web ACL. When you do this, the rule runs with the action set to count. For more information about how web ACL and rule settings interact, see [How AWS WAF processes a Web ACL](p. 13).

## AWS WAF rule statements

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 check your web requests for. 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.

Web ACLs can also contain rule statements that just reference rule groups. On the console, you don’t see these represented as rule statements, but every web ACL has a JSON format representation. In the JSON, you can see these special types of rule statements. For rules of any complexity, managing your web ACL using the JSON editor is the easiest way to go. You can retrieve the complete configuration for a web ACL in JSON format, modify it as you need, and then provide it to AWS WAF through the console, API, or CLI. The same is true for rule groups that you manage on your own.

### Nesting rule statements

AWS WAF supports nesting for many rule statements. You need to use nesting for some scenarios. For example, to control the rate of requests coming from a specific geographical area, you use a rate-based rule and nest a geographical match rule inside it, to narrow the scope of the rate tracking. Some rule statements aren't nestable. For example, you can't nest a rule group statement inside another statement.

To combine rule statement results, you nest the statements under logical AND or OR rule statements. The visual editor on the console supports one level of rule statement nesting, which works for many needs. To nest more levels, you can edit the JSON representation of the rule on the console.

### Topics

- Rule statements list (p. 33)
- Request component settings (p. 43)
- Rule statements that reference a set or a rule group (p. 45)

### Rule statements list

This section describes the statements that you can add to a rule and provides some guidelines for calculating web ACL capacity units (WCU) usage for each.

This page groups the rule statements by category, provides a high-level description for each, and provides a link to a section with more information for the statement type.
Match statements

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

<table>
<thead>
<tr>
<th>Match Statement</th>
<th>Description</th>
<th>WCUs</th>
<th>Nestable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geographic match (p. 35)</td>
<td>Inspects the request’s country of origin.</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>IP set match (p. 36)</td>
<td>Compares the request origin against a set of IP addresses and address ranges.</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>Regex pattern set (p. 39)</td>
<td>Compares regex patterns against a specified request component.</td>
<td>25 per pattern set</td>
<td>Yes</td>
</tr>
<tr>
<td>Size constraint (p. 40)</td>
<td>Checks size constraints against a specified request component.</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>SQLi attack (p. 40)</td>
<td>Inspects for malicious SQL code in a specified request component.</td>
<td>20</td>
<td>Yes</td>
</tr>
<tr>
<td>String match (p. 41)</td>
<td>Compares a string to a specified request component.</td>
<td>Depends on the type of match</td>
<td>Yes</td>
</tr>
<tr>
<td>XSS scripting attack (p. 42)</td>
<td>Inspects for cross-site scripting attacks in a specified request component.</td>
<td>40</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Logical rules statements

Logical rules statements allow you to combine other statements or negate their results. Every logical rule statement takes at least one nested statement.

<table>
<thead>
<tr>
<th>Logical Statement</th>
<th>Description</th>
<th>WCUs</th>
<th>Nestable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND logic (p. 35)</td>
<td>Combines nested statements with AND logic.</td>
<td>Based on nested statements</td>
<td>Yes</td>
</tr>
<tr>
<td>NOT logic (p. 37)</td>
<td>Negates the results of a nested statement.</td>
<td>Based on nested statement</td>
<td>Yes</td>
</tr>
<tr>
<td>OR logic (p. 37)</td>
<td>Combines nested statements with OR logic.</td>
<td>Based on nested statements</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Complex statements
AWS WAF supports the following complex statements.

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
<th>WCUs</th>
<th>Nestable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate-based (p. 38)</td>
<td>Tracks the rate of requests from individual IP addresses. You can narrow the scope with a nested statement.</td>
<td>Equal to the WCUs of any nested scoping statement.</td>
<td>No</td>
</tr>
<tr>
<td>Managed rule group (p. 37)</td>
<td>Runs the rules that are defined in the specified managed rule group.</td>
<td>Defined by rule group.</td>
<td>No</td>
</tr>
<tr>
<td>Rule group (p. 39)</td>
<td>Runs the rules that are defined in a rule group that you manage.</td>
<td>You define this for the rule group when you create it.</td>
<td>No</td>
</tr>
</tbody>
</table>

**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 one nested statement.

**Nestable** – You can nest this statement type.

**WCUs** – Depends on the nested statements.

**Where to find this**

- **Rule builder** on the console – For If a request, choose matches all the statements (AND), and then fill in the nested statements.
- **API statement** – AndStatement

**Geographic match rule statement**

To allow or block web requests based on country of origin, create one or more geographical, or geo, match statements.

**Note**

If you use 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. So if you want to allow or block requests based on geography plus other AWS WAF criteria, you should not use the CloudFront geo restriction feature. Instead, you should use an AWS WAF geo match condition.

You can use this to block access to your site from specific countries or to only allow access from specific countries. If you want to allow some web requests and block others based on country of origin, add a geo match statement for the countries that you want to allow and add a second one for the countries that you want to block.

You can use geo match statements with other AWS WAF statements to build sophisticated filtering. For example, to block certain countries, but still allow requests from a specific set of IP addresses in that country, you could create a rule with the action set to Block and the following nested statements:

- **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

As another example, if you want to prioritize resources for users in a particular country, you could create a different rate-based rules statement for each geo match condition. Set a higher rate limit for users in the preferred country and set a lower rate limit for all other users.

Nestable – You can nest this statement type inside logical rule statements and rate-based statements.

WCUs – 1 WCU.

This statement requires the following settings:

• Geo match – An array of country codes to compare for a geo match. These must be two-character country codes, for example, [ "US", "CN" ], from the alpha-2 country ISO codes of the ISO 3166 international standard.

  Each code must be one or two characters long.

Where to find this

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

IP set match rule statement

The IP set match statement inspects the IP address of a web request's origin 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. AWS WAF supports all IPv4 and IPv6 address ranges. 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. This allows you to use the single set in multiple rules. When you update the referenced IP set, AWS WAF automatically updates all rules that reference it.
  For information about creating and managing an IPSet, see Creating and managing an IP set (p. 46).

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

Nestable – You can nest this statement type inside logical rule statements and rate-based statements.

WCUs – 1 WCU.

This statement requires the following settings:

• IP set specification – Choose the IP set that you want to use from the list or create a new one.

Where to find this

• 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 statement – IPSetReferenceStatement
Managed rule group statement

The managed rule group rule statement adds a reference in your 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 web ACL rules as this type.

A managed rule group is either an, which is free for AWS WAF customers, or a AWS Marketplace managed rule group, which you can subscribe to through AWS Marketplace. For more information, see Managed rule groups (p. 21).

When you add a rule group to a web ACL, you can exclude individual rules in the group from running, and you can override the actions of all rules in the rule group, to count only. For more information, see How AWS WAF processes a Web ACL (p. 13).

**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 web ACL.

**WCUs** – Set for the rule group at creation.

**Where to find this**

- **Console** – During the process of creating a 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 statement** – ManagedRuleGroupStatement

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 geographical match statement that specifies the country.

**Nestable** – You can nest this statement type.

**WCUs** – Depends on the nested statement.

**Where to find this**

- **Rule builder** on the console – For If a request, choose doesn’t match the statement (NOT), and then fill in the nested statement.
- **API statement** – 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 one nested statement.

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 geographics 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 geographics match statement one
level lower, inside a NOT statement. This level of nesting requires you to use the JSON formatting, as the console supports only one level of nesting.

**Nestable** – You can nest this statement type.

**WCUs** – Depends on the nested statements.

**Where to find this**
- **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 statement** – **OrStatement**

**Rate-based rule statement**

A rate-based rule tracks the rate of requests for each originating IP address, and triggers the rule action on IPs with rates that go over a limit. You set the limit as the number of requests per 5-minutes time span. You can use this type of rule to put a temporary block on requests from an IP address that's sending excessive requests.

When the rule action triggers, AWS WAF blocks additional requests from the IP address until the request rate falls below the limit.

You can optionally nest another statement inside the rate-based statement, to narrow the scope so that AWS WAF only counts requests that match the nested statement. For example, based on recent requests that you've seen from an attacker in the United States, you might create a rate-based rule with a nested AND rule statement that contains the following, second level of nested statements:

- A geo-match match statement that specifies requests originating in the United States.
- A string match statement that searches in the User-Agent header for the string BadBot.

Let's say that you also set a rate limit of 1,000. For each IP address, AWS WAF counts requests that meet both of the conditions. Requests that don't meet both conditions aren't counted. If the count for an IP address exceeds 1,000 requests in any 5-minute time span, the rule's action triggers against that IP address.

As another example, you might want to limit requests to the login page on your website. To do this, you could create a rate-based rule with the following nested string match statement:

- The **Request option** is **URI**.
- The **Match Type** is **Starts with**.
- A **Strings to match** is login.

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.

**Not nestable** – You can't nest this statement type inside other statements. You can include it directly in a web ACL. You cannot include it in a rule group.

**WCUs** – Depends on the nested statement.

**Where to find this**
- **Rule builder** on the console – For **Request option**, choose **Rate**.
- **API statement** – **RateBasedStatement**
**Regex pattern set match rule statement**

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.

**Note**
Each regex pattern set match rule references a regex pattern set, which you create and maintain independent of your rules. This allows you to use the single set in multiple rules. When you update the referenced regex pattern 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 (p. 48).

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.

**Nestable** – You can nest this statement type inside logical rule statements and rate-based statements.

**WCUs** – 25 WCUs per regex pattern set.

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

- **Request components** – The part of the web request to inspect, for example, a query string or the body. For more information, see Request component (p. 43).
- **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 more information, see Text transformations (p. 44).

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

**Rule group statement**

The rule group rule statement adds a reference to your 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 web ACL, any of your own rule groups that you've added show up under the web ACL rules as this type. For information about using your own rule groups, see Managing your own rule groups (p. 29).

When you add a rule group to a web ACL, you can exclude individual rules in the group from running, and you can override the actions of all rules in the rule group, to count only. For more information, see How AWS WAF processes a Web ACL (p. 13).

**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 web ACL.
WCUs – Set for the rule group at creation.

Where to find this

- **Console** – During the process of creating a 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 statement** – RuleGroupReferenceStatement

Size constraint rule statement

A size constraint statement compares a number of bytes against the size of a request component, using a comparison 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.

**Note**
If you configure AWS WAF to inspect the request body, AWS WAF inspects only the first 8,192 bytes (8 KB). If the request body for your web requests never exceeds 8,192 bytes, you can create a size constraint condition and block requests that have a request body greater than 8,192 bytes.

**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 /logo.jpg is nine characters long.

**Nestable** – You can nest this statement type inside logical rule statements and rate-based statements.

**WCUs** – 1 WCU.

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

- **Request components** – The part of the web request to inspect, for example, a query string or the body. For more information, see Request component (p. 43).
- **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 more information, see Text transformations (p. 44).

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

SQL injection attack rule statement

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. An SQL injection match condition identifies the part of web requests, such as the URI or the query string, that you want AWS WAF 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.

**Nestable** – You can nest this statement type inside logical rule statements and rate-based statements.

**WCUs** – 20 WCUs.

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

- **Request components** – The part of the web request to inspect, for example, a query string or the body. For more information, see Request component (p. 43).
- **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 more information, see Text transformations (p. 44).

**Where to find this**

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

**String match rule statement**

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).

**Nestable** – You can nest this statement type inside logical rule statements and rate-based statements.

**WCUs** – Depends on the type of match that you use. For example, **Starts with string** requires 2 WCUs, and **Contains string** requires 10 WCUs.

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

- **Request components** – The part of the web request to inspect, for example, a query string or the body. For more information, see Request component (p. 43).
- **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 more information, see Text transformations (p. 44).

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.
  - **Contains string** – The string appears anywhere in the request component.
• **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 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 builder** on the console – For **Match type**, choose **String match condition**, and then fill in the strings that you want to match against.

• **API statement** – **ByteMatchStatement**

**Cross-site scripting attack rule statement**

Attacker 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 part of web requests, such as the URI or the query string, that you want AWS WAF to inspect for possible malicious scripts.

When you create cross-site scripting match conditions, you specify filters. The filters indicate the part of web requests that you want AWS WAF 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.

**Nestable** – You can nest this statement type inside logical rule statements and rate-based statements.

**WCUs** – 40 WCUs.

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

• **Request components** – The part of the web request to inspect, for example, a query string or the body. For more information, see **Request component** (p. 43).

• **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 more information, see **Text transformations** (p. 44).

**Where to find this**

• **Rule builder** on the console – For **Match type**, choose **Attack match conditions** > **Contains XSS injection attacks**.

• **API statement** – **XssMatchStatement**
Request component settings

This section describes the settings that you specify whenever you use a rule statement that inspects a component of the web request.

Topics

- Request component (p. 43)
- Text transformations (p. 44)

Request component

The request component specifies the part of a web request for AWS WAF to inspect. You specify this for standard rule statements that look for patterns inside the web request. These include regex pattern match, SQL injection attack, and size constraint statements.

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 these settings in the following locations:

- Rule builder on the console – For Request option, choose Request components.
- API statement contents – FieldToMatch

Here are the options for the part of the web request to inspect:

Options for the part of the request to inspect

Header

A specific request header. For this option, you also choose the name of the header in the Header type field, for example, User-Agent or Referer.

HTTP method

The HTTP method, which indicates the type of operation that the web request is asking the origin to perform.

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 instead of Query string.

Single query parameter

Any 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 parameter name. For example, if the URL is www.xyz.com?UserName=abc&SalesRegion=seattle, you can specify UserName or SalesRegion for the name. The maximum length for the name is 30 characters. The name is not case sensitive, so if you specify UserName as the name, AWS WAF matches all variations of UserName, including username and UsERName.

If the query string contains more than one instance of the name 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

Similar to Single query parameter, but AWS WAF inspect the values of all parameters 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.

URI

The part of a URL that identifies a resource, for example, /images/daily-ad.jpg. If you don’t use a text transformation with this option, AWS WAF doesn’t normalize the URI and inspects it just as it receives it from the client in the request.

Body

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

Note

Only the first 8 KB (8,192 bytes) of the request body are forwarded to AWS WAF for inspection. If you don’t need to inspect more than 8 KB, you can guarantee that you don’t allow additional bytes in by combining your statement that inspects the body of the web request, such as a string match rule statement, with a size constraint rule statement that enforces an 8 KB max size on the body of the request. For information about size constraint statements, see Size constraint rule statement. AWS WAF doesn’t support inspecting the entire contents of web requests whose bodies exceed 8 KB.

Text transformations

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

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

None

AWS WAF inspects the web request as received.

Convert to lowercase

AWS WAF converts any uppercase letters, A–Z, to lowercase, a–z.

HTML decode

AWS WAF replaces HTML-encoded characters with unencoded characters:

• Replaces &quot; with &
Rule statements

- Replaces &nbsp; with a non-breaking space
- Replaces &lt; with <
- Replaces &gt; with >
- Replaces characters that are represented in hexadecimal format, &amp;#xhhhh;, with the corresponding characters
- Replaces characters that are represented in decimal format, &amp;#nnnn;, with the corresponding characters

Normalize white space

AWS WAF replaces multiple spaces with one space and 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

Simplify command 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

URL decode

Decode a URL-encoded request.

Rule statements that reference a set or a rule group

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 in a 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 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 (p. 46).
- **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 (p. 46).
- **AWS** manages these rule groups. On the console, these are available for your use when you add a managed rule group to your web ACL. For more information about these, see AWS Managed Rules rule groups list (p. 24).
AWS WAF, AWS Firewall Manager, and AWS Shield Advanced Developer Guide
IP sets and regex pattern sets

- **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 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 managed rule groups](p. 28).

- **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 [AWS Marketplace managed rule groups](p. 28).

### 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 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 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.

### 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.

#### Eventual consistency

When you make changes to web ACLs or web ACL components, like rules and rule groups, AWS WAF propagates the changes everywhere that the web ACL and its components are stored and used. Your changes are applied within seconds, but there might be a brief period of inconsistency when the changes have arrived in some places and not in others. So, for example, if you add an IP address to an IP set that's referenced by a blocking rule in a web ACL, the new address might briefly be blocked in one area while still allowed in another. This temporary inconsistency can occur when you first associate a web ACL with an AWS resource and when you change a web ACL that is already associated with a resource. Generally, any inconsistencies of this type last only a few seconds.

#### Topics
- [Creating and managing an IP set](p. 46)
- [Creating and managing a regex pattern set](p. 48)

### Creating and managing an IP set

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 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 web ACL or rule group.

#### Topics
- [Creating an IP set](p. 47)
- [Using an IP set in a rule group or Web ACL](p. 47)
- [Editing an IP set](p. 47)
- [Deleting an IP set](p. 48)
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 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://console.aws.amazon.com/wafv2/.
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 the Region where you want to store the IP set. To use an IP set in 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. 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 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::/64, enter `2620:0:2d0:200::/64`.
7. Review the settings for the IP set, and choose Create IP set.

Using an IP set in a rule group or Web ACL

To use an IP set, add a rule statement that references it to the rule group or web ACL where you need it. For information, see IP set match rule statement (p. 36).

Editing an IP set

To add or remove IP addresses or IP address ranges from an IP set or change its description, perform the following procedure.

**To edit an IP set**

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.
2. In the navigation pane, choose IP sets.
3. Select the IP set that you want to edit and choose Edit.
4. Modify the IP version and addresses as needed. In the **IP addresses** text box, you must have one IP address or IP address range per line, in CIDR notation. AWS WAF supports all IPv4 and IPv6 CIDR
Creating and managing a regex pattern set

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 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 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 an OR. That is, a web request will match the pattern set rule statement if the request component matches any of the patterns in the set.

Regex pattern use limitations

AWS WAF supports standard Perl Compatible Regular Expressions (PCRE) with the following exceptions, which it doesn’t support:

- 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

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 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 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 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 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://console.aws.amazon.com/wafv2/.
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

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://console.aws.amazon.com/wafv2/.
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 the Region where you want to store the regex pattern set. To use a regex pattern set in 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 standard Perl Compatible Regular Expressions (PCRE) with the following exceptions, which it doesn't support:

- 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
6. Review the settings for the regex pattern set, and choose Create regex pattern set.

Using a regex pattern set in a rule group or Web ACL

To use a regex pattern set in a rule group or web ACL, in the console, when you add or update the rules in your rule group or web ACL, in the Rule builder interface, for Request option, choose the request component that you want to compare to your pattern set. Choose Match type > String match condition > Matches pattern from regular expression, and then choose the name of the regex pattern set that you want to use.

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 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 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 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 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://console.aws.amazon.com/wafv2/.
2. In the navigation pane, choose Regex pattern sets.
3. Select the regex pattern set that you want to delete and choose Delete.

Logging Web ACL traffic information

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 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 to your storage destination through the HTTPS endpoint of Kinesis Data Firehose. For more information about how to create an Amazon Kinesis Data Firehose and review the stored logs, see What Is Amazon Kinesis Data Firehose?

You must have the following permissions to successfully enable logging:

- iam:CreateServiceLinkedRole
- firehose:ListDeliveryStreams
- wafv2:PutLoggingConfiguration

For more information about service-linked roles and the iam:CreateServiceLinkedRole permission, see Using service-linked roles for AWS WAF (p. 69).

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 Kinesis Data Firehose Delivery Stream.

   Important
   Do not choose Kinesis stream as your source.
   One AWS WAF log is equivalent to one Kinesis Data 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 Kinesis Data Firehose. If you don't configure Kinesis Data Firehose correctly, AWS WAF 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://console.aws.amazon.com/wafv2/.
3. In the navigation pane, choose **Web ACLs**.
4. Choose the web ACL that you want to enable logging for.
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 XXX in the logs. For example, if you redact the **cookie** field, the **cookie** field in the logs will be XXX.
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 Amazon Kinesis Data Firehose. For more information, see Using service-linked roles for AWS WAF (p. 69).

**To disable logging for a web ACL**
1. In the navigation pane, choose **Web ACLs**.
2. Choose the web ACL that you want to disable logging for.
3. On the **Logging** tab, choose **Disable logging**.
4. In the dialog box, choose **Disable logging**.

**Example Example log**

```json
{
  "timestamp": 1576280412771,
  "formatVersion": 1,
  "terminatingRuleId": "STMTest_SQLi_XSS",
  "terminatingRuleType": "REGULAR",
  "action": "BLOCK",
  "terminatingRuleMatchDetails": [
    {
      "conditionType": "SQL_INJECTION",
      "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"
      }
    ]
  }
}
```
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 non-terminating 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.
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

You can access the list of IP addresses that are currently blocked by a rate-based rule by using the CLI, the API, or any of the SDKs. This topic covers access using the CLI and APIs. The console doesn't provide this functionality 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 blocked IP addresses for a rate-based rule that's being used on an Amazon CloudFront distribution.

```bash
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 API or an Application Load Balancer.

```bash
aws wafv2 get-rate-based-statement-managed-keys --scope=REGIONAL --region=region --web-acl-name=WebACLName --web-acl-id=WebACLId --rule-name=RuleName
```

How AWS WAF works with Amazon CloudFront features

When you create a web ACL, you can specify one or more CloudFront distributions that you want AWS WAF to inspect. AWS WAF 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 functionality. This chapter describes a few ways that you can configure CloudFront to make CloudFront and AWS WAF work better together.

### Topics

- Using AWS WAF with CloudFront custom error pages (p. 55)
- Using AWS WAF with CloudFront geo restriction (p. 55)
- Using AWS WAF with CloudFront for applications running on your own HTTP server (p. 55)
- Choosing the HTTP methods that CloudFront responds to (p. 56)

#### Using AWS WAF with CloudFront custom error pages

When AWS WAF 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 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 with CloudFront geo restriction

You can use the Amazon CloudFront geo restriction feature, also known as geoblocking, to prevent users in specific geographic locations from accessing content that you distribute through a CloudFront web distribution. If you want to block web requests from specific countries and also block requests based on other conditions, you can use CloudFront geo restriction in conjunction with AWS WAF. CloudFront returns the same HTTP status code to viewers—HTTP 403 (Forbidden)—whether they try to access your content from a country on a CloudFront geo restriction blacklist or whether the request is blocked by AWS WAF.

**Note**

You can see the two-letter country code of the country that requests originate from in the sample of web requests for a web ACL. For more information, see Viewing a sample of web requests (p. 19).

For more information about CloudFront geo restriction, see Restricting the Geographic Distribution of Your Content in the Amazon CloudFront Developer Guide.

#### 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
Choosing the HTTP methods that CloudFront responds to

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 byte match rule statements to allow or block requests based on the HTTP method, as described in String match rule statement (p. 41). 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*.

**Security in 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, 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**

- Data protection in AWS WAF (p. 57)
- Identity and access management in AWS WAF (p. 58)
- Logging and monitoring in AWS WAF (p. 72)
- Compliance validation for AWS WAF (p. 73)
- Resilience in AWS WAF (p. 74)
- Infrastructure security in AWS WAF (p. 74)

**Data protection in AWS WAF**

AWS WAF conforms to the AWS shared responsibility model, which includes regulations and guidelines for data protection. AWS is responsible for protecting the global infrastructure that runs all the AWS services. AWS maintains control over data hosted on this infrastructure, including the security configuration controls for handling customer content and personal data. AWS customers and APN partners, acting either as data controllers or data processors, are responsible for any personal data that they put in the AWS Cloud.

AWS WAF entities—such as 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.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM), so that 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.
- Set up API and user activity logging with AWS CloudTrail. See AWS CloudTrail API Reference.
- 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 personal data stored in Amazon S3.

We strongly recommend that you never put sensitive identifying information, such as your customers' account numbers, into free-form 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 piece of data that you enter...
Identity and access management in AWS WAF

Access to AWS WAF requires credentials. Those credentials must have permissions to access AWS resources, such as an AWS WAF resource or an Amazon S3 bucket. The following sections provide details on how you can use AWS Identity and Access Management (IAM) and AWS WAF to help secure access to your resources.

- Authentication (p. 58)
- Access control (p. 59)

Authentication

You can access AWS as any of the following types of identities:

- **AWS account root user** – When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

- **IAM user** – An IAM user is an identity within your AWS account that has specific custom permissions (for example, permissions to create a rule in AWS WAF). You can use an IAM user name and password to sign in to secure AWS webpages like the AWS Management Console, AWS Discussion Forums, or the AWS Support Center.

In addition to a user name and password, you can also generate access keys for each user. You can use these keys when you access AWS services programmatically, either through one of the several SDKs or by using the AWS Command Line Interface (CLI). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don't use AWS tools, you must sign the request yourself. AWS WAF supports Signature Version 4, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 Signing Process in the AWS General Reference.

- **IAM role** – An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user in that it is an AWS identity with permissions policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session. IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as federated users. AWS assigns a role to a federated user when access is requested through an identity
AWS WAF, AWS Firewall Manager, and AWS Shield Advanced Developer Guide
Identity and access management

provider. For more information about federated users, see Federated Users and Roles in the IAM User Guide.

• AWS service access – A service role is an IAM role that a service assumes to perform actions in your account on your behalf. When you set up some AWS service environments, you must define a role for the service to assume. This service role must include all the permissions that are required for the service to access the AWS resources that it needs. Service roles vary from service to service, but many allow you to choose your permissions as long as you meet the documented requirements for that service. Service roles provide access only within your account and cannot be used to grant access to services in other accounts. You can create, modify, and delete a service role from within IAM. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data from that bucket into an Amazon Redshift cluster. For more information, see Creating a Role to Delegate Permissions to an AWS Service in the IAM User Guide.

• Applications running on Amazon EC2 – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances in the IAM User Guide.

Access control
You can have valid credentials to authenticate your requests, but unless you have permissions you can’t create or access AWS WAF resources. For example, you must have permissions to create an AWS WAF web ACL or rule group.

The following sections describe how to manage permissions for AWS WAF. We recommend that you read the overview first.

• Overview of managing access permissions to your AWS WAF resources (p. 60)
• Using identity-based policies (IAM policies) for AWS WAF (p. 63)
• AWS WAF API permissions: Actions, resources, and conditions reference (p. 67)

AWS Identity and Access Management
AWS WAF integrates with AWS Identity and Access Management (IAM), a service that lets your organization do the following:

• Create users and groups under your organization’s AWS account
• Share your AWS account resources with users in the account
• Assign unique security credentials to each user
• Control user access to services and resources

For example, you can use IAM with AWS WAF to control which users in your AWS account can create a new web ACL.

For general information about IAM, see the following documentation:

• AWS Identity and Access Management (IAM)
• IAM Getting Started Guide
Overview of managing access permissions to your AWS WAF resources

Every AWS resource is owned by an AWS account, and permissions to create or access a resource are governed by permissions policies. An account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles). Some services also support attaching permissions policies to resources.

**Note**

An *account administrator* (or administrator user) is a user with administrator privileges. For more information, see IAM Best Practices in the IAM User Guide.

When granting permissions, you decide who is getting the permissions, the resources they get permissions for, and the specific operations that you want to allow on those resources.

Topics

- AWS WAF resources and operations (p. 60)
- Understanding resource ownership (p. 61)
- Managing access to resources (p. 61)
- Specifying policy elements: Actions, effects, resources, and principals (p. 62)
- Specifying conditions in a policy (p. 63)

AWS WAF resources and operations

In AWS WAF, the resources are *web ACLs*, *rulegroups*, *IP sets*, and *regex pattern sets*. These resources have unique Amazon Resource Names (ARNs) associated with them, as shown in the following table.

<table>
<thead>
<tr>
<th>Name in AWS WAF Console</th>
<th>Name in AWS WAF SDK/CLI</th>
<th>ARN Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule group</td>
<td>RuleGroup</td>
<td>arn:aws:wafv2:region:account:rulegroup/ID</td>
</tr>
<tr>
<td>IP set</td>
<td>IPSet</td>
<td>arn:aws:wafv2:region:account:ipset/ID</td>
</tr>
</tbody>
</table>

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 for AWS WAF have the following format:

```
arwn:aws:wafv2:region: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 resource.
• **ID**: The ID of the AWS WAF 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 in Region us-east-1:

```
arn:aws:wafv2:us-east-1:111122223333:webacl/*
```

For more information, see Resources in the IAM User Guide.

AWS WAF provides a set of operations to work with AWS WAF resources. For a list of available operations, see Actions.

### Understanding resource ownership

A **resource owner** is the AWS account that creates the resource. That is, the resource owner is the AWS account of the principal entity (the root account, an IAM user, or an IAM role) that authenticates the request that creates the resource. The following examples illustrate how this works:

- If you use the root account credentials of your AWS account to create an AWS WAF resource, your AWS account is the owner of the resource.
- If you create an IAM user in your AWS account and grant permissions to create an AWS WAF resource to that user, the user can create an AWS WAF resource. However, your AWS account, to which the user belongs, owns the AWS WAF resource.
- If you create an IAM role in your AWS account with permissions to create an AWS WAF resource, anyone who can assume the role can create an AWS WAF resource. Your AWS account, to which the role belongs, owns the AWS WAF resource.

### Managing access to resources

A **permissions policy** describes who has access to what. The following sections explain the available options for creating permissions policies.

**Note**

These sections discuss using IAM in the context of AWS WAF. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see What Is IAM? in the IAM User Guide. For information about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the IAM User Guide.

Policies that are attached to an IAM identity are known as **identity-based** policies, and policies that are attached to a resource are known as **resource-based** policies. AWS WAF supports only identity-based policies.

**Topics**

- [Identity-based policies (IAM policies)](p. 61)
- [Resource-based policies](p. 62)

### Identity-based policies (IAM policies)

You can attach policies to IAM identities. For example, you can do the following:

- **Attach a permissions policy to a user or a group in your account** – An account administrator can use a permissions policy that is associated with a particular user to grant permissions for that user to create an AWS WAF resource.
- **Attach a permissions policy to a role (grant cross-account permissions)** – You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions. For example, the administrator in Account A can create a role to grant cross-account permissions to another AWS account (for example, Account B) or an AWS service as follows:

1. Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions on resources in Account A.
2. Account A administrator attaches a trust policy to the role identifying Account B as the principal who can assume the role.
3. Account B administrator can then delegate permissions to assume the role to any users in Account B. Doing this allows users in Account B to create or access resources in Account A. The principal in the trust policy also can be an AWS service principal if you want to grant an AWS service permissions to assume the role.

For more information about using IAM to delegate permissions, see Access Management in the IAM User Guide.

The following is an example policy that grants permissions for the `wafv2:ListWebACLs` action on all resources. In the current implementation, AWS WAF 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 must specify a wildcard character (*):

```json
{
    "Version": "2019-07-29",
    "Statement": [
        {
            "Sid": "ListWebACLs",
            "Effect": "Allow",
            "Action": [
                "wafv2:ListWebACLs"
            ],
            "Resource": "*"
        }
    ]
}
```

For more information about using identity-based policies with AWS WAF, see Using identity-based policies (IAM policies) for AWS WAF (p. 63). For more information about users, groups, roles, and permissions, see Identities (Users, Groups, and Roles) in the IAM User Guide.

**Resource-based policies**

Other services, such as Amazon S3, also support resource-based permissions policies. For example, you can attach a policy to an S3 bucket to manage access permissions to that bucket. AWS WAF doesn't support resource-based policies.

**Specifying policy elements: Actions, effects, resources, and principals**

For each AWS WAF resource (see AWS WAF resources and operations (p. 60)), the service defines a set of API operations (see AWS WAF API permissions: Actions, resources, and conditions reference (p. 67)). To grant permissions for these API operations, AWS WAF defines a set of actions that you can specify in a policy. Note that performing an API operation can require permissions for more than one action. When granting permissions for specific actions, you also identify the resource on which the actions are allowed or denied.

The following are the most basic policy elements:

- **Resource** – In a policy, you use an Amazon Resource Name (ARN) to identify the resource to which the policy applies. For more information, see AWS WAF resources and operations (p. 60).
• **Action** – You use action keywords to identify resource operations that you want to allow or deny. For example, the `wafv2:CreateRuleGroup` permission allows the user permissions to perform the AWS WAF `CreateRuleGroup` operation.

• **Effect** – You specify the effect when the user requests the specific action. This can be either allow or deny. If you don’t explicitly grant access to a resource, access is implicitly denied. You also can explicitly deny access to a resource, which you might do to make sure that a user cannot access it, even if a different policy grants access.

• **Principal** – In identity-based policies (IAM policies), the user that the policy is attached to is the implicit principal. AWS WAF doesn’t support resource-based policies.


For a table that shows all the AWS WAF API actions and the resources that they apply to, see [AWS WAF API permissions: Actions, resources, and conditions reference](https://docs.aws.amazon.com/waf/latest/APIReference/). (p. 67).

### Specifying conditions in a policy

When you grant permissions, you can use the IAM policy language to specify the conditions when a policy should take effect. For example, you might want a policy to be applied only after a specific date. For more information about specifying conditions in a policy language, see [Condition](https://docs.aws.amazon.com/IAM/latest/UserGuide/index.html) in the IAM User Guide.

To express conditions, you use predefined condition keys. There are no condition keys specific to AWS WAF. However, there are AWS-wide condition keys that you can use as appropriate. For a complete list of AWS-wide keys, see [Available Keys for Conditions](https://docs.aws.amazon.com/IAM/latest/UserGuide/index.html) in the IAM User Guide.

### Using identity-based policies (IAM policies) for AWS WAF

This section provides examples of identity-based policies that demonstrate how an account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles) and thereby grant permissions to perform operations on AWS WAF resources.

**Important**

We recommend that you first review the introductory topics that explain the basic concepts and options available for you to manage access to your AWS WAF resources. For more information, see [Overview of managing access permissions to your AWS WAF resources](https://docs.aws.amazon.com/waf/latest/_gettingstarted/). (p. 60).

The following shows an example of a permissions policy:

```json
{
    "Version": "2019-07-29",
    "Statement": [
        {
            "Sid": "CreateFunctionPermissions",
            "Effect": "Allow",
            "Action": [
                "wafv2:ListWebACLs",
                "wafv2:GetWebACL",
                "cloudwatch:ListMetrics",
                "wafv2:GetSampledRequests"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PermissionToPassAnyRole",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*"
        }
    ]
}
```
The policy has two statements:

- The first statement grants permissions to view statistics for AWS WAF web ACLs, using the \texttt{wafv2:ListWebACLs}, \texttt{wafv2:GetWebACL}, \texttt{cloudwatch:ListMetrics}, and \texttt{wafv2:GetSampledRequests} actions. AWS WAF doesn't support permissions for some of these actions at the resource level. Therefore, the policy specifies a wildcard character (*) as the \texttt{Resource} value.

- The second statement grants permissions for the IAM action \texttt{iam:PassRole} on IAM roles. The wildcard character (*) at the end of the \texttt{Resource} value means that the statement allows permissions for the \texttt{iam:PassRole} action on any IAM role. To only extend these permissions to a specific role, replace the wildcard character (*) in the resource ARN with the specific role name.

The policy doesn't specify the \texttt{Principal} element because in an identity-based policy you don't specify the principal who gets the permissions. When you attach a policy to a user, the user is the implicit principal. When you attach a permissions policy to an IAM role, the principal identified in the role's trust policy gets the permissions.

For a table that shows all the AWS WAF API actions and the resources that they apply to, see \texttt{AWS WAF API permissions: Actions, resources, and conditions reference (p. 67)}.

\textbf{Topics}

- Permissions required to use the AWS WAF console (p. 64)
- AWS managed (predefined) policies for AWS WAF (p. 64)
- Customer managed policy examples (p. 65)

### Permissions required to use the AWS WAF console

The AWS WAF console provides an integrated environment for you to create and manage AWS WAF resources. The console provides many features and workflows that often require permissions to create an AWS WAF resource in addition to the API-specific permissions that are documented in the \texttt{AWS WAF API permissions: Actions, resources, and conditions reference (p. 67)}. For more information about these additional console permissions, see \texttt{Customer managed policy examples (p. 65)}.

### AWS managed (predefined) policies for AWS WAF

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. Managed policies grant necessary permissions for common use cases so you can avoid having to investigate what permissions are needed. For more information, see \texttt{AWS Managed Policies in the IAM User Guide}.

The following AWS managed policies, which you can attach to users in your account, are specific to AWS WAF and are grouped by use case scenario:

- \texttt{AWSWAFReadOnlyAccess} – Grants read-only access to AWS WAF resources.
- \texttt{AWSWAFFullAccess} – Grants full access to AWS WAF resources.

**Note**

You can review these permissions policies by signing in to the IAM console and searching for specific policies there.
You also can create your own custom IAM policies to allow permissions for AWS WAF API operations and resources. You can attach these custom policies to the IAM users or groups that require those permissions or to custom execution roles (IAM roles) that you create for your AWS WAF resources.

**Customer managed policy examples**

The examples in this section provide a group of sample policies that you can attach to a user. If you are new to creating policies, we recommend that you first create an IAM user in your account and attach the policies to the user, in the sequence outlined in the steps in this section.

You can use the console to verify the effects of each policy as you attach the policy to the user. Initially, the user doesn't have permissions, and the user won't be able to do anything on the console. As you attach policies to the user, you can verify that the user can perform various operations on the console.

We recommend that you use two browser windows: one to create the user and grant permissions, and the other to sign in to the AWS Management Console using the user's credentials and verify permissions as you grant them to the user.

For examples that show how to create an IAM role that you can use as an execution role for your AWS WAF resource, see Creating IAM Roles in the *IAM User Guide*.

**Example topics**

- **Example 1**: Give users read-only access to AWS WAF, CloudFront, and CloudWatch (p. 65)
- **Example 2**: Give users full access to AWS WAF, CloudFront, and CloudWatch (p. 66)
- **Example 3**: Granting access to a specified AWS account (p. 66)
- **Example 4**: Granting access to a specified Web ACL (p. 67)

**Create an IAM user**

First, you need to create an IAM user, add the user to an IAM group with administrative permissions, and then grant administrative permissions to the IAM user that you created. You then can access AWS using a special URL and the user's credentials.

For instructions, see Creating Your First IAM User and Administrators Group in the *IAM User Guide*.

**Example 1: Give users 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 web ACLs to see which distribution is associated with a web ACL, and to monitor metrics and a sample of requests in CloudWatch. These users can't create, update, or delete AWS WAF resources.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "wafv2:Get*",
                "wafv2:List*",
                "cloudfront:GetDistribution",
                "cloudfront:GetDistributionConfig",
                "cloudfront:ListDistributions",
                "cloudfront:ListDistributionsByWebACLId",
                "cloudwatch:ListMetrics",
                "cloudwatch:GetMetricStatistics"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

API Version 2019-07-29
Example 2: Give users 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.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Action": [
            "wafv2:*",
            "cloudfront:CreateDistribution",
            "cloudfront:GetDistribution",
            "cloudfront:GetDistributionConfig",
            "cloudfront:UpdateDistribution",
            "cloudfront:ListDistributions",
            "cloudfront:ListDistributionsByWebACLId",
            "cloudfront:DeleteDistribution",
            "cloudwatch:ListMetrics",
            "cloudwatch:GetMetricStatistics"
         ],
         "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.

Example 3: Granting access to a specified 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 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:wafv2:us-east-1:444455556666:*"
         ]
      },
      {
         "Effect": "Allow",
```
Example 4: Granting access to a specified Web ACL

This policy grants the following permissions to the webacl ID 112233d7c-86b2-458b-af83-51c51example in the account 44445555666:

- Full access to AWS WAF Get, Update, and Delete operations and resources

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["wafv2:*"],
      "Resource": ["arn:aws:wafv2:us-east-1:44445555666:webacl/112233d7c-86b2-458b-af83-51c51example"]
    }
  ]
}
```

AWS WAF API permissions: Actions, resources, and conditions reference

When you set up Access control (p. 59) and writing permissions policies that you can attach to an IAM identity (identity-based policies), use the information in this section as a guide. For each AWS WAF API operation, you need to know the actions for which you can grant permissions to perform the action, and the AWS resource for which you can grant the permissions. You specify the actions in the policy's Action field, and you specify the resource value in the policy's Resource field.

**Note**
To specify an action, use the wafv2: prefix followed by the API operation name (for example, wafv2:CreateIPSet).

You can use AWS-wide condition keys in your AWS WAF policies to express conditions. For a complete list of AWS-wide keys, see Available Keys for Conditions in the IAM User Guide.

Global and regional settings

In the resource settings in this section, use the following scope and region settings:

- For CloudFront distributions, set scope to global and set region to us-east-1.
• For API Gateway APIs and Application Load Balancers, set `scope` to regional and set the `region` to the region you’re interested in.

**AWS WAF standard API permissions**

The basic CRUD and list operations on AWS resources follow a standard pattern for permissions granting. This section shows the pattern for web ACL operations.

This pattern applies to web ACLs, rule groups, IP sets, and regex pattern sets.

**CRUD operations**

- **AWS WAF API Operations** – CreateWebACL, GetWebACL, UpdateWebACL, and DeleteWebACL
- **API Actions** – `wafv2:CreateWebACL`, `wafv2:GetWebACL`, `wafv2:UpdateWebACL`, `wafv2:DeleteWebACL`

**List operations**

- **AWS WAF API Operation** – ListWebACLs
- **API Actions** – `wafv2:ListWebACLs`

If you want to list all resources in your account, call the list operation once for global, and once for each regional application region.

**AWS WAF non-standard API permissions**

The following operations don’t follow the standard CRUD and list pattern and require specific resource permissions settings.

For each operation, we list the required policy actions and their associated policy resources.

**AssociateWebACL**

- **API Actions** – `wafv2:AssociateWebACL`, `elasticloadbalancing:SetWebACL`, `apigateway:SetWebACL`
- **Resources** –
  - `arn:aws:wafv2:region:account-id:scope/webacl/entity-name/entity-ID`
  - `arn:aws:elasticloadbalancing:region:account-id:loadbalancer/app/ApplicationLoadBalancerName/ApplicationLoadBalancerID`
  - `arn:aws:apigateway:region::/restapis/api-ID/stages/stage-name`

**CheckCapacity**

- **API Action** – `wafv2:CheckCapacity`
- **Resource** – This requires permissions on all ARNs that are referenced in the contained rules. It doesn't require any other permissions.

**DescribeManagedRuleGroup**

- **API Action** – `wafv2:DescribeManagedRuleGroup`
DisassociateWebACL

**API Actions** – wafv2:DisassociateWebACL, elasticloadbalancing:SetWebACL, apigateway:SetWebACL

**Resources** –

arn:aws:wafv2:region:account-id:scope/webacl/entity-name/entity-ID

arn:aws:elasticloadbalancing:region:account-id:loadbalancer/app/ApplicationLoadBalancerName/ApplicationLoadBalancerID

arn:aws:apigateway:region::/restapis/api-ID/stages/stage-name

GetRateBasedStatementManagedKeys

**API Action** – wafv2:GetRateBasedStatementManagedKeys


GetSampledRequests

**API Action** – wafv2:GetSampledRequests

**Resource** – The resource permissions depend on the parameters that you specify in the API call. You must have access to the web ACL that corresponds to the request for samples. For example: arn:aws:wafv2:region:account-id:scope/webacl/entity-name/entity-ID

ListAvailableManagedRuleGroups

**API Action** – wafv2:ListAvailableManagedRuleGroups


Using service-linked roles for AWS WAF

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.

AWS WAF uses this service-linked role to write logs to Amazon Kinesis Data Firehose. This role is used only if you enable logging in AWS WAF. For more information, see Logging Web ACL traffic information (p. 50).

The AWSServiceRoleForWAFV2Logging service-linked role trusts the service to assume the role wafv2.amazonaws.com.
The permissions policies of the role allows AWS WAF to complete the following actions on the specified resources:

- **Action:** `firehose:PutRecord` and `firehose:PutRecordBatch` on Amazon Kinesis Data Firehose data stream resources with a name that starts with "aws-waf-logs-." For example, `aws-waf-logs-us-east-2-analytics`.

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 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 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 the following AWS Regions.
<table>
<thead>
<tr>
<th>Region Name</th>
<th>Region Identity</th>
<th>Support in AWS WAF</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (N. Virginia)</td>
<td>us-east-1</td>
<td>Yes</td>
</tr>
<tr>
<td>US East (Ohio)</td>
<td>us-east-2</td>
<td>Yes</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>us-west-1</td>
<td>Yes</td>
</tr>
<tr>
<td>US West (Oregon)</td>
<td>us-west-2</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>ap-south-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Osaka-Local)</td>
<td>ap-northeast-3</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>ap-northeast-2</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>ap-southeast-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Sydney)</td>
<td>ap-southeast-2</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>ap-northeast-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Canada (Central)</td>
<td>ca-central-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>eu-central-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>eu-west-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (London)</td>
<td>eu-west-2</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>eu-west-3</td>
<td>Yes</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>sa-east-1</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Logging and monitoring 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 incidents:

**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](p. 282).

**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](p. 286).

**AWS Trusted Advisor**

Trusted Advisor draws upon best practices learned from serving hundreds of thousands of AWS customers. Trusted Advisor inspects your AWS environment and then makes recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps. All AWS customers have access to five Trusted Advisor checks. Customers with a Business or Enterprise support plan can view all Trusted Advisor checks. For more information, see [AWS Trusted Advisor](p. 282).
Compliance validation for AWS WAF

Third-party auditors assess the security and compliance of AWS WAF as part of multiple AWS compliance programs, such as SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. 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 WAF is determined by the sensitivity of your data, your organization's compliance objectives, and applicable laws and regulations. If your use of AWS WAF is subject to compliance with standards like HIPAA, PCI, or FedRAMP, AWS provides resources to help:

- **Security and Compliance Quick Start Guides** – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- **Architecting for HIPAA Security and Compliance Whitepaper** – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- **AWS Compliance Resources** – This collection of workbooks and guides might apply to your industry and location.
- **AWS Config** – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- **AWS Security Hub** – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.
- **AWS Well-Architected Framework** – The AWS Well-Architected Framework helps you build secure cloud applications.
Resilience in 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

As a managed service, AWS WAF is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access AWS WAF through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

AWS WAF quotas

AWS WAF is subject to the following quotas (formerly referred to as limits).

AWS WAF has default quotas on the maximum number of entities you can have per account. You can request an increase in these quotas.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web ACLs per Region</td>
<td>100</td>
</tr>
<tr>
<td>Rule groups per Region</td>
<td>100</td>
</tr>
<tr>
<td>Web ACL capacity units (WCUs) per web ACL</td>
<td>1,500</td>
</tr>
<tr>
<td>WCUs per rule group</td>
<td>1,500</td>
</tr>
<tr>
<td>IP sets per Region</td>
<td>100</td>
</tr>
<tr>
<td>Requests per second per web ACL (applies only to Application Load Balancers)</td>
<td>10,000</td>
</tr>
</tbody>
</table>

The maximum requests per second (RPS) allowed for AWS WAF on CloudFront is set by CloudFront and described in the CloudFront Developer Guide.

The following quotas on AWS WAF entities can't be changed.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of references (to IP sets and regex pattern sets) per rule group</td>
<td>50</td>
</tr>
</tbody>
</table>
### AWS WAF quotas

<table>
<thead>
<tr>
<th>Resource</th>
<th>Quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of references (to IP sets, regex pattern sets, and rule groups) per web ACL</td>
<td>50</td>
</tr>
<tr>
<td>IP addresses in CIDR notation per IP set</td>
<td>10,000</td>
</tr>
<tr>
<td>Unique IP addresses that can be blocked per rate-based rule</td>
<td>10,000</td>
</tr>
<tr>
<td>Maximum characters allowed for a string match statement</td>
<td>200</td>
</tr>
<tr>
<td>Maximum characters allowed for each regex pattern</td>
<td>200</td>
</tr>
<tr>
<td>Unique regex patterns per regex set</td>
<td>10</td>
</tr>
<tr>
<td>Regex sets per Region</td>
<td>10</td>
</tr>
<tr>
<td>Maximum size of a web request body that can be inspected</td>
<td>8 KB</td>
</tr>
<tr>
<td>Maximum number of rate-based rules per account</td>
<td>10</td>
</tr>
<tr>
<td>Minimum request rate that can be defined for a rate-based rule</td>
<td>100</td>
</tr>
</tbody>
</table>
## AWS WAF Classic

**Note**
This is **AWS WAF Classic** documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

*For the latest version of AWS WAF*, see [AWS WAF](#) (p. 6).

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 (p. 76)
- How AWS WAF Classic works (p. 79)
- AWS WAF Classic pricing (p. 82)
- Getting started with AWS WAF Classic (p. 82)
- Tutorials for AWS WAF Classic (p. 92)
- Creating and configuring a Web Access Control List (Web ACL) (p. 119)
- Logging Web ACL traffic information (p. 165)
- Listing IP addresses blocked by rate-based rules (p. 169)
- How AWS WAF Classic works with Amazon CloudFront features (p. 170)
- Security in AWS WAF Classic (p. 172)
- AWS WAF Classic quotas (p. 197)

### Setting up AWS WAF Classic

**Note**
This is **AWS WAF Classic** documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

*For the latest version of AWS WAF*, see [AWS WAF](#) (p. 6).

This topic describes preliminary steps, such as creating an AWS account, to prepare you to use AWS WAF Classic. You are not charged to set up this account and other preliminary items. You are charged only for AWS services that you use.

**Note**
If you are a new user, don't follow these setup steps for AWS WAF Classic. Instead, follow the steps for the latest version of AWS WAF, at [Setting up](#) (p. 3).

After you complete these steps, see [Getting started with AWS WAF Classic](#) (p. 82) 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 works](#) (p. 247).

Before you use AWS WAF Classic or AWS Shield Advanced for the first time, complete the following tasks:

---

**API Version 2019-07-29**

76
Step 1: Sign up for an AWS account

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including AWS WAF Classic. You are charged only for the services that you use.

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To sign up for AWS

2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Note your AWS account number, because you'll need it for the next task.

Step 2: Create an IAM user

To use the AWS WAF Classic console, you must sign in to confirm that you have permission to perform AWS WAF Classic operations. You can use the root credentials for your AWS account, but we don't recommend it. For greater security and control of your account, we recommend that you use AWS Identity and Access Management (IAM) to do the following:

- Create an IAM user account for yourself or your business.
- Either add the IAM user account to an IAM group that has administrative permissions, or grant administrative permissions directly to the IAM user account.

You then can sign in to the AWS WAF Classic console (and other service consoles) by using a special URL and the credentials for the IAM user. You also can add other users to the IAM user account, and control their level of access to AWS services and to your resources.

   Note
   For information about creating access keys to access AWS WAF Classic by using the AWS Command Line Interface (AWS CLI), Tools for Windows PowerShell, the AWS SDKs, or the AWS WAF Classic API, see Managing Access Keys for IAM Users.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM console. If you aren't familiar with using the console, see Working with the AWS Management Console for an overview.

To create an administrator user for yourself and add the user to an administrators group (console)

1. Use your AWS account email address and password to sign in as the AWS account root user to the IAM console at https://console.aws.amazon.com/iam/.

   Note
   We strongly recommend that you adhere to the best practice of using the Administrator IAM user below and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.
Step 2: Create an IAM user

2. In the navigation pane, choose Users and then choose Add user.
3. For User name, enter Administrator.
4. Select the check box next to AWS Management Console access. Then select Custom password, and then enter your new password in the text box.
5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to User must create a new password at next sign-in to allow the new user to reset their password after they sign in.
6. Choose Next: Permissions.
7. Under Set permissions, choose Add user to group.
8. Choose Create group.
9. In the Create group dialog box, for Group name enter Administrators.
10. Choose Filter policies, and then select AWS managed -job function to filter the table contents.
11. In the policy list, select the check box for AdministratorAccess. Then choose Create group.

Note
You must activate IAM user and role access to Billing before you can use the AdministratorAccess permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.

12. Back in the list of groups, select the check box for your new group. Choose Refresh if necessary to see the group in the list.
13. Choose Next: Tags.
14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM Entities in the IAM User Guide.
15. Choose Next: Review to see the list of group memberships to be added to the new user. When you are ready to proceed, choose Create user.

You can use this same process to create more groups and users and to give your users access to your AWS account resources. To learn about using policies that restrict user permissions to specific AWS resources, see Access Management and Example Policies.

To sign in as this new IAM user, first sign out of the AWS console. Then use the following URL, where your_aws_account_id is your AWS account number without the hyphens. For example, if your AWS account number is 1234-5678-9012, your AWS account ID is 123456789012:

https://your_aws_account_id.signin.aws.amazon.com/console/

Enter the IAM user name and password that you just created. When you're signed in, the navigation bar displays "your_user_name @ your_aws_account_id".

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. From the IAM dashboard, choose Customize and enter an alias, such as your company name. To sign in after you create an account alias, use the following URL:

https://your_account_alias.signin.aws.amazon.com/console/

To verify the sign-in link for IAM users for your account, open the IAM console and check under the IAM users sign-in link on the dashboard.

After you complete these steps, you can stop here and go to Getting started with AWS WAF Classic (p. 82) to continue getting started with AWS WAF Classic using the console. If you want to access AWS WAF Classic programmatically using the AWS WAF Classic API, continue on to the next step, Step 3: Download tools (p. 79).
Step 3: Download tools

The AWS Management Console includes a console for AWS WAF Classic, but if you want to access AWS WAF Classic programmatically, the following documentation and tools will help you:

- 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

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).

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.

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 API Gateway, 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.

The following illustration shows how AWS WAF Classic checks the rules and performs the actions based on those rules.
Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see AWS WAF (p. 6).

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

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see AWS WAF (p. 6).
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 Regions. AWS WAF Classic for use with API Gateway or an Application Load Balancer is available in the Regions listed at AWS Regions and 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 (p. 83)
- Step 2: Create a Web ACL (p. 83)
- Step 3: Create an IP match condition (p. 84)
- Step 4: Create a geo match condition (p. 84)
- Step 5: Create a string match condition (p. 85)
- Step 5A: Create a regex condition (optional) (p. 86)
- Step 6: Create a SQL injection match condition (p. 88)
- Step 7: (Optional) create additional conditions (p. 88)
- Step 8: Create a rule and add conditions (p. 89)
- Step 9: Add the rule to a Web ACL (p. 90)
- Step 10: Clean up your resources (p. 90)

**Step 1: Set up AWS WAF Classic**

If you already signed up for an AWS account and created an IAM user as described in Setting up AWS WAF Classic (p. 76), go to Step 2: Create a Web ACL (p. 83).

If not, go to Setting up AWS WAF Classic (p. 76) and perform at least the first two steps. (You can skip downloading tools for now because this Getting Started topic focuses on using the AWS WAF Classic console.)

**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://console.aws.amazon.com/wafv2/.
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 (p. 126).

#### 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.
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.

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 (p. 130).

     **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.)
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 (p. 144).
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 (p. 130).

   **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 (p. 134).

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 (p. 130).
   - 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 (p. 130).
- **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 (p. 121).
You can optionally create these conditions now, or you can skip to Step 8: Create a rule and add conditions (p. 89).

**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 (p. 149).

**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 (p. 79). 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:
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://console.aws.amazon.com/wafv2/.
   b. Choose the web ACL that you want to delete.
   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 web ACL that you created during the tutorial.
   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.

---

**Tutorials for AWS WAF Classic**

**Note**

This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

*For the latest version of AWS WAF, see [AWS WAF (p. 6)].*

This section contains a link to a preconfigured template as well as three tutorials that present complete solutions for common tasks that you can perform in AWS WAF Classic. The tutorials show how to combine several AWS services to automatically configure AWS WAF Classic in response to your CloudFront traffic. Their purpose is to provide general guidance. They are not intended for direct use in your production environment without careful review and adaptation to the unique aspects of your business environment.

---

**AWS WAF Classic Preconfigured Protections**
You can use our preconfigured template to get started quickly with AWS WAF Classic. The template includes a set of AWS WAF Classic rules that are designed to block common web-based attacks. You can customize the template to fit your business needs.

The rules in the template help protect against bad bots, SQL injection, cross-site scripting (XSS), HTTP floods, and other known attacks. After you deploy the template, AWS WAF Classic begins to block the web requests to your CloudFront distribution or to an Application Load Balancer that matches the preconfigured rules in your web access control (web ACL) list. You can use this automated solution in addition to other web ACLs that you configure. For more information, see AWS WAF Classic Security Automations.

**Tutorials for AWS WAF Classic**

- Tutorial: Quickly setting up AWS WAF Classic protection against common attacks (p. 93)
- Tutorial: Implementing a DDoS-resistant website using AWS services (p. 98)

**Tutorial: Quickly setting up AWS WAF Classic protection against common attacks**

**Note**

This is **AWS WAF Classic** documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see AWS WAF (p. 6).

This tutorial shows you how to use **AWS CloudFormation** to quickly configure AWS WAF Classic to protect against the following common attacks:

- **Cross-site scripting attacks** – Attackers sometimes insert scripts into web requests in an effort to exploit vulnerabilities in web applications. Cross-site scripting match conditions 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.
- **SQL injection attacks** – Attackers sometimes insert malicious SQL code into web requests in an effort to extract data from your database. SQL injection match conditions identify the part of web requests that you want AWS WAF Classic to inspect for possible malicious SQL code.
- **Attacks from known bad IP addresses** – You can use IP match conditions to allow, block, or count web requests based on the IP addresses that the requests originate from. An IP match condition lists up to 1,000 IP addresses or IP address ranges that you specify.

**Note**

This tutorial assumes that you have a CloudFront distribution that you use to deliver content for your web application. If you don’t have a CloudFront distribution, see Creating or Updating a Web Distribution Using the CloudFront Console in the **Amazon CloudFront Developer Guide**.

**Topics**

- Solution overview (p. 94)
- Step 1: Create an AWS CloudFormation stack that sets up AWS WAF Classic protection against common attacks (p. 96)
- Step 2: Associate a Web ACL with a CloudFront distribution (p. 96)
- Step 3: (Optional) add IP addresses to the IP Match Condition (p. 97)
- Step 4: (Optional) update the Web ACL to block large bodies (p. 97)
- Step 5: (Optional) delete your AWS CloudFormation stack (p. 98)
- Related resources (p. 98)
Solution overview

AWS CloudFormation uses a template to set up the following AWS WAF Classic conditions, rules, and a web ACL.

Conditions

AWS CloudFormation creates the following conditions.

IP Match Condition

Filters requests that come from known bad IP addresses. This lets you easily add IPs to a list to block access to your website. You might want to do this if you’re receiving a lot of bad requests from one or more IP addresses. If you want to allow, block, or count requests based on the IP addresses that the requests come from, see Step 3: (Optional) add IP addresses to the IP Match Condition (p. 97) later in this tutorial.

The name of the condition is prefixManualBlockSet where prefix is the name that you specify for the web ACL when you create the AWS CloudFormation stack.

Size Constraint Condition

Filters requests for which the body is longer than 8,192 bytes. AWS WAF Classic evaluates only the first 8,192 bytes of the request part that you specify in a filter. If valid request bodies never exceed 8,192 bytes, you can use a size constraint condition to catch malicious requests that might otherwise slip through.

For this tutorial, AWS CloudFormation configures AWS WAF Classic only to count, not block, requests that have a body longer than 8,192 bytes. If the body in your requests never exceeds that length, you can change the configuration to block requests that have longer bodies. For information about how to view the count of requests that exceed 8,192 bytes and how to change the web ACL to block requests that contain bodies larger than 8,192 bytes, see Step 4: (Optional) update the Web ACL to block large bodies (p. 97).

The name of the condition is prefixLargeBodyMatch where prefix is the name that you specify for the web ACL when you create the AWS CloudFormation stack.

SQL Injection Condition

Filters requests that contain possible malicious SQL code. The condition includes filters that evaluate the following parts of requests:
- Query string (URL decode transformation)
- URI (URL decode transformation)
- Body (URL decode transformation)
- Body (HTML decode transformation)

The name of the condition is prefixSqlMatch where prefix is the name that you specify for the web ACL when you create the AWS CloudFormation stack.

Cross-site Scripting Condition

Filters requests that contain possible malicious scripts. The condition includes filters that evaluate the following parts of requests:
- Query string (URL decode transformation)
- URI (URL decode transformation)
- Body (URL decode transformation)
- Body (HTML decode transformation)

The name of the condition is prefixXssMatch where prefix is the name that you specify for the web ACL when you create the AWS CloudFormation stack.
Rules

When you create the AWS CloudFormation stack, AWS CloudFormation creates the following rules and adds the corresponding condition to each rule:

`prefix`ManualIPBlockRule

AWS CloudFormation adds the `prefix`ManualBlockSet condition to this rule.

`prefix`SizeMatchRule

AWS CloudFormation adds the `prefix`LargeBodyMatch condition to this rule.

`prefix`SqlRule

AWS CloudFormation adds the `prefix`SqlMatch condition to this rule.

`prefix`XssRule

AWS CloudFormation adds the `prefix`XssMatch condition to this rule.

Web ACL

AWS CloudFormation creates a web ACL that has the name that you specify when you create the AWS CloudFormation stack. The web ACL contains the following rules with the specified settings:

`prefix`ManualIPBlockRule

By default, the condition in this rule doesn't contain any IP addresses. If you want to allow, block, or count requests based on the IP addresses that the requests come from, see Step 3: (Optional) add IP addresses to the IP Match Condition (p. 97) later in this tutorial.

`prefix`SizeMatchRule

By default, AWS WAF Classic counts requests for which the body is longer than 8,192 bytes.

`prefix`SqlRule

AWS WAF Classic blocks requests based on the settings in this rule.

`prefix`XssRule

AWS WAF Classic blocks requests based on the settings in this rule.

Requirements

This tutorial assumes that you have a CloudFront distribution that you use to deliver content for your web application. If you don't have a CloudFront distribution, see Creating or Updating a Web Distribution Using the CloudFront Console in the Amazon CloudFront Developer Guide. This tutorial also uses AWS CloudFormation to simplify the provisioning process. For more information, see the AWS CloudFormation User Guide.

Estimated time

The estimated time to complete this tutorial is 15 minutes if you already have a CloudFront distribution, or 30 minutes if you need to create a CloudFront distribution.

Costs

There is a cost associated with the resources that you create during this tutorial. You can delete the resources after you finish the tutorial to stop incurring charges. For more information, see AWS WAF Classic Pricing and Amazon CloudFront Pricing.
Step 1: Create an AWS CloudFormation stack that sets up AWS WAF Classic protection against common attacks

In the following procedure, you use an AWS CloudFormation template to create a stack that sets up AWS WAF Classic protection against common attacks.

**Important**
You begin to incur charges for the different services when you create the AWS CloudFormation stack that deploys this solution. Charges continue to accrue until you delete the AWS CloudFormation stack. For more information, see Step 5: (Optional) delete your AWS CloudFormation stack (p. 98).

To create an AWS CloudFormation stack for blocking IP addresses that submit bad requests

1. To start the process that creates an AWS CloudFormation stack, choose the link for the region in which you want to create AWS resources:
   - Create a stack in US East (N. Virginia)
   - Create a stack in US West (Oregon)
   - Create a stack in Europe (Ireland)
   - Create a stack in Asia Pacific (Tokyo)
2. If you are not already signed in to the AWS Management Console, sign in when prompted.
3. On the **Specify template** page, choose Amazon S3 URL. For the template URL, type `https://s3.amazonaws.com/cloudformation-examples/community/common-attacks.json`.
4. Choose Next.
5. On the **Specify stack details** page, specify the following values:
   - **Stack Name**: You can use the default name (CommonAttackProtection), or you can change the name. The stack name must not contain spaces and must be unique within your AWS account.
   - **Name**: Specify a name for the web ACL that AWS CloudFormation will create. The name that you specify is also used as a prefix for the conditions and rules that AWS CloudFormation will create, so you can easily find all the related objects.
6. Choose Next.
7. (Optional) On the **Configure stack options** page, enter tags and advanced settings or leave the boxes blank.
8. Choose Next.
9. On the **Review** page, review the configuration, and then choose Create stack.

   After you choose Create stack, AWS CloudFormation creates the AWS WAF Classic resources that are identified in Solution overview (p. 94).

Step 2: Associate a Web ACL with a CloudFront distribution

After AWS CloudFormation creates the stack, you must associate your CloudFront distribution to activate AWS WAF Classic.

**Note**
You can associate a web ACL with as many distributions as you want, but you can associate only one web ACL with a given distribution.
To associate a web ACL with a CloudFront distribution

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.
2. Choose Go to AWS WAF Classic.
3. In the navigation pane, choose Web ACLs.
4. Choose the web ACL that you want to associate with a CloudFront distribution.
5. On the Rules tab, under AWS resources using this web ACL, choose Add association.
6. When prompted, use the Resource list to choose the distribution that you want to associate this web ACL with.
7. Choose Add.
8. To associate this web ACL with additional CloudFront distributions, repeat steps 4 through 6.

Step 3: (Optional) add IP addresses to the IP Match Condition

When you created the AWS CloudFormation stack, AWS CloudFormation created an IP match condition for you, added it to a rule, added the rule to a web ACL, and configured the web ACL to block requests based on IP addresses. The IP match condition doesn't include any IP addresses, though. If you want to block requests based on IP addresses, perform the following procedure.

To edit AWS CloudFormation parameter values

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. Type an IP address or range by using CIDR notation. Here are two examples:
      • To specify the IP address 192.0.2.44, type 192.0.2.44/32.
      • To specify the range of IP addresses from 192.0.2.0 to 192.0.2.255, type 192.0.2.0/24.

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.

Note
AWS WAF Classic supports both IPv4 and IPv6 IP addresses.

   c. To add more IP addresses, choose Add another IP address, and then type the value.
   d. Choose Add.

Step 4: (Optional) update the Web ACL to block large bodies

When you created the AWS CloudFormation stack, AWS CloudFormation created a size constraint condition that filters requests that have request bodies longer than 8,192 bytes. It also added the condition to a rule, and added the rule to the web ACL. In this example, AWS CloudFormation configured the web ACL to count requests, not to block requests. This is useful when you want to confirm you are not blocking valid requests inadvertently.

If you want to block requests that are longer than 8,192 bytes, perform the following procedure.
To change the action for a rule in a web ACL

2. In the navigation pane, choose Web ACLs.
3. Choose the web ACL that you want to edit.
4. In the right pane, choose the Rules tab.
5. Choose Edit Web ACL.
6. To change the action for the `prefixLargeBodyMatchRule`, choose the preferred option. (`prefix` is the value that you specified for the name of the web ACL.)
7. Choose Save changes.

Step 5: (Optional) delete your AWS CloudFormation stack

If you want to stop protecting from common attacks as described in Solution overview (p. 94), delete the AWS CloudFormation stack that you created in Step 1: Create an AWS CloudFormation stack that sets up AWS WAF Classic protection against common attacks (p. 96). This deletes the AWS WAF Classic resources that AWS CloudFormation created and stops the AWS charges for those resources.

To delete an AWS CloudFormation stack

2. Select the check box for the stack. The default name is CommonAttackProtection.
3. Choose Delete Stack.
4. Choose Yes, Delete to confirm.
5. To track the progress of the stack deletion, select the check box for the stack, and choose the Events tab in the bottom pane.

Related resources

For AWS WAF Classic samples, including Lambda functions, AWS CloudFormation templates, and SDK usage examples, go to GitHub at https://github.com/awslabs/aws-waf-sample.

Tutorial: Implementing a DDoS-resistant website using AWS services

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).

This tutorial provides step-by-step instructions for setting up a website that is resistant to distributed denial of service (DDoS) attacks. A DDoS attack can flood your website with traffic, prevent legitimate users from accessing the site, and even cause your site to crash due to the overwhelming traffic volume.

Topics
- Overview (p. 99)
- Architecture (p. 99)
Overview

This tutorial shows you how to use several AWS services together to build a resilient, highly secure website. For example, you learn how to do the following:

- Use load balancers and edge servers, which distribute traffic to multiple instances across regions and zones and help to protect your instances from SSL-based attacks
- Mitigate infrastructure (layer 3 and layer 4) DDoS attacks with techniques like overprovisioning your capacity
- Use a web application firewall to monitor HTTP and HTTPS requests and control access to your content

The tutorial shows how to integrate AWS services such as Amazon EC2, Elastic Load Balancing, CloudFront, Route 53, and AWS WAF Classic. Although the tutorial is designed as an end-to-end solution, you don't have to complete every step if you're already using some of those AWS services. For example, if you've already registered your website domain with Route 53 and are using Route 53 as your DNS service, you can skip those steps.

The tutorial is intended to help you launch each AWS service quickly. For that reason, it doesn't cover all possible options. For detailed information about each service, see AWS Documentation. For many of the steps, this tutorial provides specific values to enter. Generally you should use those values. However, in certain cases, such as domain name for your website, use what is appropriate for your needs.

Each main step of the tutorial briefly describes the following:

- What you are doing
- Why you are doing it (that is, how it contributes to your DDoS protection)
- How to do it

Important

You are responsible for the cost of the AWS services implemented in this tutorial. For full details, see the pricing webest-practicesage for each AWS service that you use in this solution. You can find links to each service on the Cloud Products page.

Architecture

The following diagram shows the architecture deployed in this tutorial.
To get started go to Prerequisites (p. 100).

Prerequisites

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).

The following tasks are not specifically related to DDoS protection, but are necessary to complete the tutorial.

Topics
- Sign up for AWS (p. 101)
- Create an IAM user (p. 101)
- Create a key pair (p. 102)
- Create a virtual private cloud (VPC) with two subnets (p. 103)
- Create a security group (p. 104)
Sign up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS. You are charged only for the services that you use.

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To create an AWS account
2. Follow the online instructions.
   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Note your AWS account number, because you'll need it for the next task.

Create an IAM user

To access AWS services and resources, you must provide credentials. Although it's possible to sign in with the user name and password that you created when you first opened your AWS account, for security purposes we strongly recommend that you create new credentials through the AWS Identity and Access Management (IAM) service, and that you use those credentials to sign in.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the following procedure.

To create an IAM user for yourself and add the user to an administrators group
1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose Users, and then choose Add user.
3. For User name, type a user name, such as Administrator. The name can consist of letters, digits, and the following characters: plus (+), equal (=), comma (,), period (.), at (@), underscore (_), and hyphen (-). The name is not case sensitive and can be up to 64 characters in length.
4. Select the check box next to AWS Management Console access, select Custom password, and then type the new user's password in the text box.
5. Choose Next: Permissions.
6. On the Set permissions for user page, choose Add user to group.
7. Choose Create group.
8. In the Create group dialog box, type the name for the new group. The name can consist of letters, digits, and the following characters: plus (+), equal (=), comma (,), period (.), at (@), underscore (_), and hyphen (-). The name is not case sensitive and can be up to 128 characters in length.
9. For Filter, choose Job function.
10. In the policy list, select the check box for AdministratorAccess. Then choose Create group.
11. Back in the list of groups, select the check box for your new group. Choose Refresh if necessary to see the group in the list.
12. Choose Next: Review to see the list of group memberships to be added to the new user. When you are ready to proceed, choose Create user.

To sign in as this new IAM user, sign out of the AWS console, then use the following URL, where your_aws_account_id is your AWS account number without the hyphens (for example, if your AWS account number is 1234-5678-9012, your AWS account ID is 123456789012):
Enter the IAM user name (not your email address) and password that you just created. When you're signed in, the navigation bar displays "your_user_name @ your_aws_account_id".

To verify the sign-in link for IAM users for your account, open the IAM console and check under IAM users sign-in link on the dashboard.

For more information about IAM, see the IAM User Guide.

Create a key pair

A key pair is a set of security credentials that you use to prove your identity. A key pair consists of a private key and a public key that you create. You use your key pair to log in to your Amazon EC2 instance, which is a virtual server in the AWS Cloud. You specify the name of the key pair when you initially launch the instance.

To create a key pair

1. Sign in to AWS using the URL that you created in the previous section.
2. From the AWS dashboard, choose EC2 to open the Amazon EC2 console.
3. From the navigation bar, select a region for the key pair. You can select any region that's available to you, regardless of your location. However, key pairs are specific to a region; for example, if you plan to launch an instance in the US West (Oregon) Region, you must create a key pair for the instance in the US West (Oregon) Region. For this tutorial, consider choosing the US West (Oregon) Region.

Note
Later in this tutorial, we use AWS Lambda and Amazon API Gateway, which currently are available only in specific AWS Regions. Therefore, ensure that you select an AWS Region where both Lambda and Amazon API Gateway are available. US West (Oregon), suggested above, supports all the services that are used in this tutorial. For the most current service availability information, see AWS service offerings by region.

4. In the navigation pane, under NETWORK & SECURITY, choose Key Pairs.

Tip
The navigation pane is on the left side of the console. If you do not see the pane, it might be minimized; choose the arrow to expand the pane. You might have to scroll down to see the Key Pairs link.

5. Choose Create Key Pair.

6. Type a name for the new key pair in the Key pair name field of the Create Key Pair dialog box, and then choose Create. Use a name that is easy for you to remember, such as your IAM user name, followed by –key-pair, plus the region name. For example, me-key-pair-uswest2.

7. The private key file is automatically downloaded by your browser. The base file name is the name that you specified as the name of your key pair, and the file name extension is .pem. Save the private key file in a safe place.

Important
This is the only chance for you to save the private key file. You must provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

For more information, see Amazon EC2 Key Pairs.
Create a virtual private cloud (VPC) with two subnets

Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. In this tutorial your VPC will contain the two Amazon EC2 instances that host your website along with two subnets connected to those instances.

For more information about Amazon VPC, see What is Amazon VPC? in the Amazon VPC User Guide.

To create a nondefault VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. From the navigation bar, select a region for the VPC. VPCs are specific to a region, so you should select the same region in which you created your key pair. For this tutorial, we use the US West (Oregon) Region.
3. On the VPC dashboard, choose Start VPC Wizard.
4. On the Step 1: Select a VPC Configuration page, ensure that VPC with a Single Public Subnet is selected, and then choose Select.
5. On the Step 2: VPC with a Single Public Subnet page, specify the following details:
   - For VPC name, type a friendly name for your VPC.
   - For Availability Zone, choose us-west-2a.
   - For Subnet name, type subnet-1.
   - Keep the other default configuration settings.
6. Choose Create VPC. On the confirmation page, choose OK.

Add a second subnet to your VPC

For increased availability, later in this tutorial you configure a load balancer to use different subnets in two different Availability Zones. When you created your Amazon VPC in the previous step, you created the first subnet in an Availability Zone. You now must add a second subnet in a different Availability Zone. Both Availability Zones must be in the same AWS Region.

To add a second subnet to your Amazon VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose Subnets, Create Subnet.
3. Specify the following subnet details:
   - For Name tag, provide a name for your subnet. For example, type subnet-2. Doing so creates a tag with a key of Name and the value that you specify.
   - For VPC, choose the VPC that you just created in the previous steps.
   - For Availability Zone, choose an Availability Zone that your subnet will reside in. This should be different than the Availability Zone that you created with your VPC earlier in this tutorial. The tutorial used us-west-2a as an example. So this time, choose something other than us-west-2a, such as us-west-2b.
   - For IPv4 CIDR block, specify an IPv4 CIDR block for this second subnet. You must specify an IPv4 CIDR block for the subnet from the range of your VPC. The IP addresses for your two subnets cannot overlap. Assuming you used the defaults when setting up your VPC, your first subnet used CIDR block 10.0.0.0/24. So for this second CIDR block, you can use 10.0.1.0/24. For more information, see VPC and Subnet Sizing for IPv4.
4. Choose Yes, create.
5. On the subnets page, choose the first subnet you created, subnet-1.
6. In the details pane, on the Route Table tab, note the Route Table ID. It starts with rtb-.

7. On the subnets page, choose the second subnet that you created, subnet-2.

8. On the details pane, choose Edit.

9. Your second subnet must use the same route table as your first subnet. For Change to, select the name of the route table that you noted earlier.

10. Choose Save.

Create a security group

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You must add rules to a security group that enable you to connect to your instance from your IP address using RDP. You can also add rules that allow inbound and outbound HTTP and HTTPS access from anywhere.

Prerequisites

You need the public IPv4 address of your local computer. The security group editor in the Amazon EC2 console can automatically detect the public IPv4 address for you. Alternatively, you can use the search phrase “what is my IP address” in an internet browser. If you are connecting through an internet service provider (ISP) or from behind a firewall without a static IP address, you must find out the range of IP addresses used by client computers.

To create a security group with least privilege

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.

2. From the navigation bar, select a region for the security group. Security groups are specific to a region, so you should select the same region in which you created your key pair, US West (Oregon).

3. In the navigation pane, choose Security Groups .


5. Type a name for the new security group and a description. Use a name that is easy for you to remember, such as your IAM user name, followed by _SG_, plus the region name. For example, me_SG_uswest2.

6. In the VPC list, select the VPC that you created earlier in this tutorial.

7. On the Inbound tab, create the following rules (choose Add Rule for each new rule):

   • Choose HTTP from the Type list, and make sure that Source is set to Anywhere (0.0.0.0/0).
   
   • Choose HTTPS from the Type list, and make sure that Source is set to Anywhere (0.0.0.0/0).
   
   • Choose RDP from the Type list. In the Source box, choose MyIP to automatically populate the field with the public IPv4 address of your local computer. Alternatively, choose Custom and specify the public IPv4 address of your computer or network in CIDR notation. To specify an individual IP address in CIDR notation, add the routing suffix /32, for example, 203.0.113.25/32. If your company allocates addresses from a range, specify the entire range, such as 203.0.113.0/24.

   Warning

   For security reasons, we don’t recommend that you allow RDP access from all IPv4 addresses (0.0.0.0/0) to your instance, except for testing purposes and only for a short time.

8. After you have added all of the rules, choose Create.

Next: Step 1: Launch a virtual server using Amazon EC2 (p. 105).
Step 1: Launch a virtual server using Amazon EC2

**Note**
This is **AWS WAF Classic** documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. **For the latest version of AWS WAF**, see [AWS WAF](p. 6).

You can mitigate infrastructure (layer 3 and layer 4) DDoS attacks by using techniques like overprovisioning capacity. That is, you can scale your website to absorb larger volumes of traffic without capital-intensive investments or unnecessary complexity. You can use Amazon EC2 to launch virtual servers (known as *instances*) and quickly scale up or down as your requirements change. You can scale horizontally by adding instances to your website as needed. You can also choose to scale vertically by using larger instances. In this step of the tutorial, you create a **c4.8xlarge** Amazon EC2 Windows instance, which includes a 10 GB network interface and enhanced networking, in the US West (Oregon) Region.

**Important**
You are responsible for the cost of the AWS services implemented in this tutorial. For full details about EC2 costs, see the [Amazon EC2 pricing page](#).

**Topics**
- Create an Amazon EC2 instance (p. 105)
- Connect to your instance (p. 106)
- Install a web server and host your site (p. 107)
- Launch a second EC2 instance (p. 107)
- Test your website (p. 108)

**Create an Amazon EC2 instance**

The Amazon EC2 instances you create here will host your website.

**To launch an instance**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose the US West (Oregon) Region (or whatever region you chose for your VPC).
3. From the Amazon EC2 dashboard, choose **Launch Instance**.
4. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations, called *Amazon Machine Images (AMIs)*, that serve as templates for your instance. Choose the AMI for *Windows Server 2016 R2 Base*.
5. On the **Choose an Instance Type** page, choose the **c4.8xlarge** type. This type provides a 10 GB network interface and support for enhanced networking.
6. Choose **Review and Launch**.
7. Choose **Edit Instance Details**.
8. For **Network**, choose the VPC that you created in the prerequisites step, **Create a virtual private cloud (VPC) with two subnets** (p. 103).
9. For **Subnet**, choose **subnet-1**, which you created and named when creating the VPC.
10. For **Auto-assign Public IP**, choose **Enable**.
11. Choose **Review and Launch**.
12. On the **Review Instance Launch** page, under **Security Groups**, use the following steps to choose the security group that you created in the prerequisites step, **Create a security group** (p. 104).
a. Choose **Edit security groups**.

b. On the **Configure Security Group** page, ensure that **Select an existing security group** is selected.

c. Choose the security group that you created earlier from the list of existing security groups, and then choose **Review and Launch**.

13. On the **Review Instance Launch** page, choose **Launch**.

14. When prompted for a key pair, select **Choose an existing key pair**, and then select the key pair that you created in the prerequisites step, *Create a key pair* (p. 102).

**Warning**

Don't select the **Proceed without a key pair** option. If you launch your instance without a key pair, you can't connect to it.

Select the acknowledgement check box, and then choose **Launch Instances**.

15. A confirmation page lets you know that your instance is launching. Choose **View Instances** to close the confirmation page and return to the console.

16. On the **Instances** page, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is **pending**. After the instance starts, its state changes to **running** and it receives a public DNS name. (If the **Public DNS (IPv4)** column is hidden, choose the Show/Hide icon in the top-right corner of the page, and then select **Public DNS (IPv4)**.) Take note of your public IPv4 address. You need this value later in this tutorial.

17. It can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks; you can view this information in the **Status Checks** column.

**Connect to your instance**

You will use Microsoft Remote Desktop to connect to your instances. If you are connecting from a Microsoft Windows computer, Remote Desktop is already installed. If you are using another operating system, you might need to install Remote Desktop before performing the following procedure.

**To connect to your Windows instance using an RDP client**

1. In the Amazon EC2 console, select the instance, and then choose **Connect**.

2. In the **Connect To Your Instance** dialog box, choose **Get Password** (it will take a few minutes after the instance is launched before the password is available).

3. Choose **Browse** and navigate to the private key file that you created when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file into the **Contents** field.

4. Choose **Decrypt Password**. The console displays the default administrator password for the instance in the **Connect To Your Instance** dialog box, replacing the link to **Get Password** shown previously with the actual password.

5. Record the default administrator password, or copy it to the clipboard. You need this password to connect to the instance.

6. Choose **Download Remote Desktop File**. Your browser prompts you to either open or save the .rdp file. Either option is fine. When you have finished, you can choose **Close** to dismiss the **Connect To Your Instance** dialog box.

   - If you opened the .rdp file, you see the **Remote Desktop Connection** dialog box.
   - If you saved the .rdp file, navigate to your downloads directory, and then open the .rdp file to display the dialog box.

7. You might get a warning that the publisher of the remote connection is unknown. You can continue to connect to your instance.
8. When prompted, connect to and log in to the instance, using the administrator account for the operating system and the password that you recorded or copied previously.

   **Note**  
   Sometimes copying and pasting content can corrupt data. If you encounter a "Password Failed" error when you log in, try typing in the password manually.

9. Due to the nature of self-signed certificates, you might get a warning that the security certificate could not be authenticated. Use the following steps to verify the identity of the remote computer, or simply choose Yes or Continue to continue if you trust the certificate.

   a. If you are using Remote Desktop Connection from a Windows PC, choose View certificate. If you are using Microsoft Remote Desktop on a Mac, choose Show Certificate.
   
   b. Choose the Details tab, and scroll down to the Thumbest-practicesrint entry on a Windows PC, or the SHA1 Fingerprints entry on a Mac. This is the unique identifier for the remote computer's security certificate.
   
   c. In the Amazon EC2 console, select the instance, choose Actions, and then choose Get System Log.
   
   d. In the system log output, look for an entry labeled RDPCERTIFICATE-THUMbest-practicesRINT. If this value matches the thumbest-practicesrint or fingerprint of the certificate, you have verified the identity of the remote computer.
   
   e. If you are using Remote Desktop Connection from a Windows PC, return to the Certificate dialog box and choose OK. If you are using Microsoft Remote Desktop on a Mac, return to the Verify Certificate and choose Continue.
   
   f. [Windows] Choose Yes in the Remote Desktop Connection window to connect to your instance.  
   [Mac OS] Log in as prompted, using the default administrator account and the default administrator password that you recorded or copied previously. You might need to switch spaces to see the login screen. For more information about spaces, see the Apple website.
   
   g. If you receive an error while attempting to connect to your instance, see Remote Desktop can't connect to the remote computer.

---

**Install a web server and host your site**

The next step is to install a web hosting service on your Amazon EC2 instance and build your website. You have several options for a web server, such as Microsoft Internet Information Server (IIS), which is already part of your instance, Apache HTTP Server for Windows, and others.

Installing a web server and configuring your website is outside the scope of this tutorial. Refer to the proper product documentation to implement a web server on your instance. However, as an example, at a general level, the steps for installing IIS are the following:

- Connect to your instance as described earlier.
- Using Windows Server Manager, choose Add roles and features.
- Choose Role-based or feature-based installation.
- Choose Web Server (IIS) and begin the installation process.
- After the installation is complete, build your website.

**Launch a second EC2 instance**

You now must repeat this process (launch another EC2 instance and build your website) to create a duplicate of your first EC2 instance. This is necessary to enable load balancing later in the tutorial.

Follow all the same steps just described to launch an instance. Be sure to edit the second instance details and security group as per the previous steps. When editing the instance details, note the following:
Before you begin (p. 108)
Create your load balancer (p. 108)
Test your load balancer (p. 110)

Before you begin
Ensure that the Amazon EC2 instances that you launched earlier in this tutorial are in the Active state.

Create your load balancer
Next, you configure a load balancer that automatically routes traffic to your two Amazon EC2 instances.

To create a load balancer
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation bar, select the same region that you selected for your EC2 instances.

3. In the navigation pane, under **LOAD BALANCING**, choose **Target Groups**.

4. Choose **Create target group**.

5. Specify a name, protocol, port, and VPC for the target group, and then choose **Create**. For this tutorial, use the following values:

   - **Name**: MyWebServers
   - **Protocol**: HTTP
   - **Port**: 80
   - **Target type**: Instance
   - **VPC**: The VPC that contains your EC2 instances
   - Keep the other settings.

6. Select the new target group.

7. On the **Targets** tab, choose **Edit**.

8. For **Instances**, select both of the instances that you created earlier in this tutorial. Choose **Add to registered**, and then choose **Save**.

   The status of the instances is **initial** until the instances are registered and have passed health checks, and then it is **unused** until you configure the target group to receive traffic from the load balancer.

9. In the navigation pane, under **LOAD BALANCING**, choose **Load Balancers**.

10. Choose **Create Load Balancer**.

11. For **Select load balancer type**, choose **Application Load Balancer**.

12. Choose **Create**.

13. Complete the **Configure Load Balancer** page as follows:

   a. For **Name**, type a name for your load balancer.

   b. For **Scheme**, choose **Internet-facing**. An internet-facing load balancer routes requests from clients over the internet to targets. An internal load balancer routes requests to targets using private IP addresses.

   c. For **Listeners**, the default is a listener that accepts HTTP traffic on port 80.

   d. For **Availability Zones**, select the VPC that you used for your EC2 instances. Select at least two Availability Zones. If there is one subnet for an Availability Zone, it is selected. If there is more than one subnet for an Availability Zone, select one of the subnets. You can select only one subnet per Availability Zone.

   e. Choose **Next: Configure Security Settings**.

14. For now, ignore the message about creating a secure listener group. Choose **Next: Configure Security Groups**.

15. Complete the **Configure Security Groups** page as follows:

   a. Select **Create a new security group**.

   b. Type a name and description for the security group, or keep the default name and description. This new security group contains a rule that allows traffic to the port that you selected for your load balancer on the **Configure Load Balancer** page.

   c. Choose **Next: Configure Routing**.

16. Complete the **Configure Routing** page as follows:

   a. For **Target group**, choose **Existing target group**.

   b. For **Name**, choose the target group that you created earlier.

   c. Choose **Next: Register Targets**.
17. On the Register Targets page, the instances that you registered with the target group appear under Registered instances. You can't modify the targets registered with the target group until after you complete the wizard. Choose Next: Review.


19. After you are notified that your load balancer was created successfully, choose Close.

Test your load balancer

You should now be able to view your website using the DNS name of the load balancer.

To test your load balancer

1. On the Amazon EC2 console, in the navigation pane, select Load Balancers.
2. Select the box next to your load balancer.
3. In the details pane, note the DNS name.
4. Enter this address in a web browser. You should be directed to your website.

   Important
   If you make changes to the website, you must make the same changes to both EC2 instances. The load balancer can serve content from either instance, so it is important that both instances are identical.

Next: Step 3: Improve performance and absorb attacks using Amazon CloudFront (p. 110).

Step 3: Improve performance and absorb attacks using Amazon CloudFront

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see AWS WAF (p. 6).

Highly scaled, diverse internet connections can significantly improve the response time of your website, better absorb DDoS attacks, and isolate faults. Amazon CloudFront edge servers along with Route 53 provide the additional layer of network infrastructure that you need to achieve these benefits. Your content is served and DNS queries are resolved from locations that typically are closer to your users than your EC2 origin servers. This reduces the load on your origin EC2 servers.

Important
You are responsible for the cost of the AWS services implemented in this tutorial. For full details about CloudFront costs, see the CloudFront pricing page.

Topics
- Deliver Your Content using Amazon CloudFront (p. 110)

Deliver Your Content using Amazon CloudFront

Amazon CloudFront is a content delivery network (CDN) service that you can use to deliver your entire website, including static, dynamic, streaming, and interactive content. You can use persistent TCP connections and variable time-to-live (TTL) to accelerate the delivery of your content, even if it can't be cached at an edge location. This allows you to use CloudFront to protect your web application, even if you are not serving static content.
CloudFront accepts only well-formed connections to prevent many common DDoS attacks, like SYN floods and UDP reflection attacks, from reaching your origin. CloudFront can automatically close connections that are unusually slow, which can indicate a potential DDoS attack.

Further, DDoS attacks are geographically isolated close to the source, which prevents the traffic from affecting other locations. You can also use the CloudFront geo restriction feature to prevent users in specific geographic locations from accessing your content. This can be useful in case you want to block attacks that are originating from geographic locations where you do not expect to serve users.

All of these capabilities can greatly improve your ability to continue serving traffic to users during large DDoS attacks.

**To implement Amazon CloudFront**

1. Open the CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. Choose Create Distribution.
3. On the Select a delivery method for your content page, in the Web section, choose Get Started.
4. On the Create Distribution page, for Origin name, type the name of the load balancer that you created earlier in the tutorial. To find the name, go to the Amazon EC2 dashboard and choose Load Balancers in the navigation pane. Choose the load balancer that you created earlier.
5. Accept all the default values for the remainder of the Origin Settings fields.
6. Under Default Cache Behavior Settings, accept the default values, and CloudFront will do the following:
   - Forward all requests that use the CloudFront URL for your distribution (for example, http://d111111abcdef8.cloudfront.net/image.jpg) to the load balancer that you specified earlier
   - Allow users to use either HTTP or HTTPS to access your objects
   - Respond to requests for your objects
   - Cache your objects at CloudFront edge locations for 24 hours
   - Forward only the default request headers to your origin and not cache your objects based on the values in the headers
   - Allow everyone to view your content
   - Not automatically compress your content

   For more information, see Cache Behavior Settings.
7. Under Distribution Settings, accept the defaults, other than the following:

   **Price Class**
   
   Select the price class that corresponds with the maximum price that you want to pay for CloudFront service. By default, CloudFront serves your objects from edge locations in all CloudFront regions.

   For more information about price classes and about how your choice of price class affects CloudFront performance for your distribution, see Choosing the Price Class for a CloudFront Distribution. For information about CloudFront pricing, including how price classes map to CloudFront regions, see Amazon CloudFront Pricing.

   **AWS WAF Classic Web ACL**

   Choose None. You configure AWS WAF Classic later in this tutorial.

   **Alternate Domain Names (CNAMEs) (Optional)**

   Specify a domain name that you want to use for your website's URLs. For example, you could enter example.com.
Default Root Object (Optional)

The object that you want CloudFront to request from your origin (for example, index.html) when a viewer requests the root URL of your distribution (http://example.com/) instead of an object in your distribution (http://example.com/product-description.html). Specifying a default root object avoids exposing the contents of your distribution.

Comment (Optional)

Enter any comments that you want to save with the distribution.

8. Choose Create Distribution.

9. After CloudFront creates your distribution, the value of the Status column for your distribution changes from InProgress to Deployed. If you chose to enable the distribution, it will then be ready to process requests. This should take less than 15 minutes.

The domain name that CloudFront assigns to your distribution appears in the list of distributions. (It also appears on the General tab for a selected distribution.) Note both this name and the Distribution ID because you need these later in the tutorial.

10. On the CloudFront console, note the ID of the distribution that you just created. You need this ID later in the tutorial.

To test your CloudFront distribution

1. On the CloudFront console, select the ID of the distribution that you just created. This opens the details page for this distribution. Note the domain name.

2. Open that domain name in a browser. You should see your website. It might take about 15 minutes or so for the distribution to be active. If you get an error that indicates that your origin closed the connection, give it some more time and try again. You might also have to refresh the page in your browser.

Next: Step 4: Register your Domain name and implement DNS service using Route 53 (p. 112).

Step 4: Register your Domain name and implement DNS service using Route 53

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).

You can use Route 53 to register the domain name for your website, route internet traffic to the resources for your domain, and check the health of your web server to verify that it’s reachable, available, and functional. Route 53 helps to protect against DDoS attacks by providing redundancy and load balancing across multiple DNS servers. Route 53 can also detect anomalies in DNS queries and prioritize requests from users that are known to be reliable and, by extension, deprioritize requests that are from potentially less reliable sources.

Important
You are responsible for the cost of the AWS services implemented in this tutorial. For full details about Route 53 costs, see the Route 53 pricing page.

Topics
- Register your Domain with Route 53 (p. 113)
- Create records (p. 114)
Register your Domain with Route 53

If you are new to hosting a website, your next step in this tutorial is to register a domain using Route 53. Following are the steps to do this.

**Important**
If your domain is already registered with another registrar, you must migrate your existing domain from the other registrar’s DNS service to instead use Route 53 as the DNS service. This tutorial does not cover that transfer process. Instead of following the Route 53 procedures described in this tutorial, you must perform four steps to transfer an existing domain:

- Create a hosted zone
- Get your current DNS configuration from your DNS service provider
- Create resource records sets
- Update your registrar’s name servers

For more information about transferring an existing domain registration from another registrar, see Transferring Domains.

**To register a new domain using Route 53**

2. Under *Domain Registration*, choose *Get Started Now*.
3. Choose *Register Domain*.
4. Type the domain name that you want to register, and choose *Check* to find out whether the domain name is available. As an example, this tutorial assumes that you register the domain name example.com.

   For information about how to specify characters other than a-z, 0-9, and - (hyphen) and how to specify internationalized domain names, see DNS Domain Name Format.

5. If the domain is available, choose *Add to cart*. The domain name appears in your shopping cart.
6. In the shopping cart, choose the number of years that you want to register the domain for.
7. To register more domains, repeat steps 4 through 6.
8. Choose *Continue*.
9. On the *Contact Details for Your n Domains* page, enter contact information for the domain registrant, administrator, and technical contacts. The values that you enter here are applied to all the domains that you’re registering.
10. For some top-level domains (TLDs), we’re required to collect additional information. For these TLDs, enter the applicable values after the *Postal/Zip Code* field.
11. Choose whether you want to hide your contact information from WHOIS queries. For more information, see the following topics:

   - Enabling or Disabling Privacy Protection for Contact Information for a Domain
   - Domains That You Can Register with Route 53
12. Choose *Continue*.
13. Review the information that you entered, read the terms of service, and select the check box to confirm that you’ve read the terms of service.
14. Choose *Complete Purchase*.

For generic TLDs, we typically send an email to the registrant for the domain to verify that the registrant contact can be reached at the email address that you specified. (We don’t send an email if we already have confirmation that the email address is valid.) The email comes from one of the following email addresses:
Create records

Your next step is to create records that tell Route 53 how you want to route traffic for the domain and subdomain.

To create records

2. In the navigation pane, choose Hosted zones.
3. Because you registered your domain using Route 53, Route 53 automatically creates a hosted zone for you. Choose this hosted zone.
4. Choose Create Record Set.
5. Enter the applicable values:
   - For Name, leave as is (it should already be example.com).
   - For Type, choose A – IPv4 address.
   - For Alias, choose Yes.
   - For Alias Target, type the domain name of your CloudFront distribution that you created earlier in this tutorial.
6. Choose Create.

Note
Your new record takes time to propagate to the Route 53 DNS servers. Changes generally propagate to all Route 53 name servers within 60 seconds.

To test your Route 53 records

1. Open the domain name you added to the record, such as example.com, in a browser.
2. You should see your website.

Next: Step 5: Detect and filter malicious web requests using AWS WAF Classic (p. 114).

Step 5: Detect and filter malicious web requests using AWS WAF Classic

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.
You can use a web application firewall (WAF) to protect your web applications against attacks that attempt to exploit a vulnerability in your website. Common examples include SQL injection or cross-site request forgery. You can also use a firewall to detect and mitigate web application layer DDoS attacks.

AWS WAF Classic is a web application firewall service that lets you monitor the HTTP and HTTPS requests that are forwarded to 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, CloudFront responds to the requests either with the requested content or with an HTTP 403 status code (Forbidden).

Some attacks consist of web traffic that is disguised to look like regular user traffic. To mitigate this type of attack, you can use AWS WAF Classic rate-based blacklisting. With rate-based blacklisting, you can set a threshold for how many requests your web application can serve. If a bot or crawler exceeds this limit, you can use AWS WAF Classic to automatically block any additional requests.

AWS provides preconfigured templates that include a set of AWS WAF Classic rules, which you can customize to best fit your needs. These templates are designed to block common web-based attacks such as bad bots, SQL injection, cross-site scripting (XSS), HTTP floods, and known-attacker attacks. This tutorial uses these templates to provide firewall protection for your website. The following procedures show you how to deploy the templates using AWS CloudFormation. For more information, including a diagram of the template's solution, see AWS WAF Classic Security Automations.

The template uses some AWS features, such as AWS Lambda and Amazon API Gateway, that are not covered in this tutorial. The template performs all the necessary configuration, so you don't need to perform any additional actions for those services. However, if you want to learn more about Lambda and Amazon API Gateway, see the AWS Lambda Developer Guide and Amazon API Gateway Developer Guide.

**Important**
You are responsible for the cost of all the AWS services that are deployed as part of this template, including Amazon S3, AWS Lambda, Amazon API Gateway, AWS WAF Classic, and others. For full details, see the pricing page for each AWS service.

**Topics**
- Launch the stack (template) (p. 115)
- Associate the Web ACL with your web application (p. 117)
- Configure web access logging (p. 117)

**Launch the stack (template)**

This automated AWS CloudFormation template deploys the AWS WAF Classic Security Automations solution on the AWS Cloud.

**To launch the AWS CloudFormation stack (template)**

1. Sign into the AWS CloudFormation console.
2. If this is your first time using AWS CloudFormation, on the Select Template page, choose Specify an Amazon S3 template URL and then enter https://s3.amazonaws.com/solutions-reference/aws-waf-security-automations/latest/aws-waf-security-automations.template. If you've used AWS CloudFormation in the past, choose Create stack, and then choose Specify an Amazon S3 template URL and enter https://s3.amazonaws.com/solutions-reference/aws-waf-security-automations/latest/aws-waf-security-automations.template.
3. Choose Next.
4. On the Specify Details page, specify the following values:
Stack Name

Type a name for the AWS WAF configuration. This will also be the name of the web ACL that the template creates, for example, MyWebsiteACL.

Activate SQL Injection Protection

Choose yes to enable the component that is designed to block common SQL injection attacks.

Activate Cross-site Scripting Protection

Choose yes to enable the component that is designed to block common XSS attacks.

Activate HTTP Flood Protection

Choose yes. This component configures a rate-based rule to protect against attacks that consist of a large number of requests from a particular IP address, such as a web-layer DDoS attack or a brute-force login attempt. The rate-based rule is automatically triggered when web requests from a client exceed a configurable threshold, which defines the maximum number of incoming requests allowed from a single IP address within a five-minute period. Once this threshold is breached, additional requests from the IP address are blocked until the request rate falls below the threshold. For more information on rate-based rules, see How AWS WAF Classic Works.

Activate Scanner & Probe Protection

Choose yes to enable the component that is designed to block scanners and probes.

Activate Reputation List Protection

Choose yes to block requests from IP addresses on third-party reputation lists (supported lists: spamhaus, torproject, and emergingthreats).

Activate Bad Bot Protection

Choose yes. The template requires this protection to be enabled. However, to take full advantage of this protection, you must complete additional steps outside the scope of this tutorial, such as creating a honeypot link. Those steps are described AWS WAF Security Automations, Step 3. Embed the Honeypot Link in Your Web Application. These additional steps are optional and are not required to complete this tutorial. If you choose to perform these additional steps, complete this tutorial first, and then you can set up the honeypot link.

CloudFront Access Log Bucket Name

Type a name for the Amazon S3 bucket where you want to store access logs for your CloudFront distribution. This is the name of a new bucket that the template creates during stack launch. Do not use an existing name.

Important
This field is required. Although you will not get an immediate error if you omit this information, the formation process will not complete successfully unless you specify a name.

Request Threshold

This is used for the HTTP flood protection, so is not applicable for this tutorial. You can leave the default, which is 2000.

Error Threshold

This is the maximum acceptable bad requests per minute per IP address. This is used by the scanner and probe protection. Use the default value, which is 50.

WAF Block Period

This is the period (in minutes) to block applicable IP addresses that are identified by the scanner and probe protection. Use the default value, which is 240.
Send Anonymous Usage Data

Choose yes to send anonymous data to AWS to help us understand solution usage across our customer base as a whole. To opt out of this feature, choose no.

5. Choose Next.
6. Make no changes on the Options page.
7. Choose Next.
8. On the Review page, review and confirm the settings. Be sure to select the check box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
9. Choose Create to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the Status column. You should see a status of CREATE_COMPLETE in about fifteen (15) minutes.

Associate the Web ACL with your web application

Now associate your Amazon CloudFront web distribution with the web ACL.

To associate the web ACL with your web application
1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.
2. In the navigation pane, choose Web ACLs.
3. Select your newly created WebACL. The name of this ACL is the name that you specified in the previous step, for example, MyWebsiteACL.
5. Choose Add association.
6. For AWS resources using this web ACL, choose the CloudFront distribution that you created earlier in this tutorial.
7. Choose Add to save your changes.

Configure web access logging

As the last step of this tutorial, you configure Amazon CloudFront to send web access logs to the appropriate Amazon S3 bucket so that this data is available for the Log Parser AWS Lambda function.

To store web access logs from a CloudFront distribution
1. Open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. Choose the check box next to your distribution, and then choose Distribution Settings.
3. On the General tab, choose Edit.
4. Confirm that for AWS WAF Web ACL, the web ACL that the solution created (the same name that you assigned to the stack during initial configuration) is already entered.
5. For Logging, choose On.
6. For Bucket for Logs, choose the Amazon S3 bucket that you want use to store web access logs (that you defined in Launch the stack (template) (p. 115)).
7. Choose Yes, edit to save your changes.

Next: Additional best practices (p. 118).
Additional best practices

**Note**
This is **AWS WAF Classic** documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of **AWS WAF**, see [AWS WAF](#).

You now have several components in place to help protect your website from DDoS attacks. However, there is still more you can do. Following are several best practices you should consider. This tutorial does not cover the implementation details of the best practices, but links to relevant documentation are provided.

**Topics**
- Obscuring AWS resources (p. 118)
- Using security groups (p. 118)
- Network access control lists (ACLs) (p. 118)
- Protecting your origin (p. 119)
- Conclusion (p. 119)

**Obscuring AWS resources**

For many website applications, your AWS resources do not need to be fully exposed to the internet. For example, it might not be necessary for Amazon EC2 instances behind an elastic load balancer (ELB) to be publicly accessible. In this scenario, you might decide to allow users to access the ELB on certain TCP ports and to allow only the ELB to communicate with the Amazon EC2 instances. You can do this by configuring security groups and network access control lists (ACLs) within your Amazon Virtual Private Cloud (VPC). Amazon VPC allows you to provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define.

Security groups and network ACLs are similar in that they allow you to control access to AWS resources within your Amazon VPC. Security groups allow you to control inbound and outbound traffic at the instance level. Network ACLs offer similar capabilities, but at the VPC subnet level. Additionally, there is no charge for inbound data transfer on Amazon EC2 security group rules or network ACLs. This ensures that you do not incur any additional charges for traffic that is dropped by your security groups or network ACLs.

**Using security groups**

You can specify security groups when launching an Amazon EC2 instance or associate the instance with a security group later on. All traffic to a security group from the internet is implicitly denied unless you create an `Allow` rule to permit the traffic. For example, in this tutorial, you created a solution that consists of an ELB and two Amazon EC2 instances. You should consider creating one security group for the ELB ("ELB security group") and one for the instances ("web application server security group"). You can then create allow rules to permit traffic from the internet to the ELB security group and to permit traffic from the ELB security group to the web application server security group. As a result, traffic from the internet is unable to directly communicate with your Amazon EC2 instances, which makes it more difficult for an attacker to learn about the design and structure of your website.

**Network access control lists (ACLs)**

With network ACLs, you can specify both `Allow` and `Deny` rules. This is useful in case you want to explicitly deny certain types of traffic to your website. For example, you can define IP addresses (as CIDR ranges), protocols, and destination ports that should be denied for the entire subnet. If your website is used only for TCP traffic, you can create a rule to deny all UDP traffic, or vice versa. This tool is useful
when responding to DDoS attacks because it can allow you to create your own rules to mitigate the attack if you know the source IP addresses or other signature. You can use network ACLs in conjunction with AWS WAF Classic ACLs.

Protecting your origin

You should consider configuring CloudFront to prevent users from bypassing CloudFront and requesting content directly from your origin. This can improve the security of your origin. To learn more, see Using Custom Headers to Restrict Access to Your Content on a Custom Origin.

Conclusion

The best practices outlined in this tutorial can help you to build a DDoS-resilient architecture that can protect the availability of your website against many common infrastructure and application layer DDoS attacks. The degree to which you are able to architect your application according to these best practices influences the type and volume of DDoS attacks that you can mitigate.

For more information, see the following:

- AWS Best Practices for DDoS Resiliency
- AWS Documentation

Blog tutorials

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).

Blog Tutorials

The following tutorial topics link out to the AWS Security Blog.

- How to Import IP Address Reputation Lists to Automatically Update AWS WAF IP Blacklists
- How to Reduce Security Threats and Operating Costs Using AWS WAF and Amazon CloudFront
- How to Prevent Hotlinking by Using AWS WAF, Amazon CloudFront, and Referer Checking
- How to Use AWS CloudFormation to Automate Your AWS WAF Configuration with Example Rules and Match Conditions

Creating and configuring a Web Access Control List (Web ACL)

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).

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:
Creating and configuring a Web Access Control List (Web ACL)

- 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 (p. 156).
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 (p. 121).
   - 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 (p. 126).
   - 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 (p. 128).
   - 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 (p. 130).
   - 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 (p. 134).
   - 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 (p. 139).
   - 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 (p. 144).
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 (p. 149). 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 (p. 156).

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 (p. 121)
- Working with rules (p. 149)
- Working with web ACLs (p. 156)
Working with conditions

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).

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 (p. 121).
- 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 (p. 126).
- 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 (p. 128).
- 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 (p. 130).
- 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 (p. 134).
- 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 (p. 139).
- 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 (p. 144).

Topics
- Working with cross-site scripting match conditions (p. 121)
- Working with IP match conditions (p. 126)
- Working with geographic match conditions (p. 128)
- Working with size constraint conditions (p. 130)
- Working with SQL injection match conditions (p. 134)
- Working with string match conditions (p. 139)
- Working with regex match conditions (p. 144)

Working with cross-site scripting match conditions

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).

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 (p. 122)
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://console.aws.amazon.com/wafv2/.
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 (p. 122).
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 part of a URL that identifies a resource, for example, `/images/daily-ad.jpg`. 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** (p. 130).

**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, if 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 &quot; with &
- Replaces &nbsp; with a non-breaking space
- Replaces &lt; with <
- Replaces &gt; with >
- Replaces characters that are represented in hexadecimal format, #:xhhh:, 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**

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 (p. 122)**.
   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**

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**.
Working with conditions

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

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see AWS WAF (p. 6).

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 (p. 126)
• Editing IP match conditions (p. 127)
• Deleting IP match conditions (p. 127)

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://console.aws.amazon.com/wafv2/.

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: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 2620:0:2d0:200:0:0:0:0, 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://console.aws.amazon.com/wafv2/.
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 2620:0:2d0:200:0:0:0:0 to 2620:0:2d0:200:0:0:0:0, enter 2620:0:2d0:200::/64.
   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://console.aws.amazon.com/wafv2/.
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**.
6. Choose **Delete** to delete the selected condition.

### Working with geographic match conditions

**Note**

This is **AWS WAF Classic** documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of **AWS WAF**, see [AWS WAF](#) (p. 6).

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 (p. 128)
- Editing geo match conditions (p. 129)
- Deleting geo match conditions (p. 129)

### 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.
Working with conditions

**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://console.aws.amazon.com/wafv2/.
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://console.aws.amazon.com/wafv2/.
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://console.aws.amazon.com/wafv2/.
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

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see AWS WAF (p. 6).

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 (p. 130)
• Values that you specify when you create or edit size constraint conditions (p. 131)
• Adding and deleting filters in a size constraint condition (p. 133)
• Deleting size constraint conditions (p. 134)

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://console.aws.amazon.com/wafv2/.
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 (p. 131).
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 part of a URL that identifies a resource, for example, /images/daily-ad.jpg. 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 /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 `&nbsp;` with a non-breaking space
- Replaces `&lt;` with `<`
- Replaces `&gt;` 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://console.aws.amazon.com/wafv2/.
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 (p. 131).
   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://console.aws.amazon.com/wafv2/.
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

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.
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 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.

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://console.aws.amazon.com/wafv2/.
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 (p. 136).
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 part of a URL that identifies a resource, for example, /images/daily-ad.jpg. 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 (p. 130).

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 \textit{SalesRegion}. The maximum length for \textit{Query parameter name} is 30 characters. \textit{Query parameter name} is not case sensitive. For example, it you specify \textit{UserName} as the \textit{Query parameter name}, this will match all variations of \textit{UserName}, such as \textit{username} and \textit{USERName}.

All query parameters (values only)

Similar to \textit{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 \textit{All query parameters (values only)}, AWS WAF Classic will trigger a match if the value of either \textit{UserName} or \textit{SalesRegion} contain possible malicious SQL code.

Header

If you chose \textit{Header} for \textit{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:

\textbf{None}

AWS WAF Classic doesn't perform any text transformations on the web request before inspecting it for the string in \textit{Value to match}.

\textbf{Convert to lowercase}

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

\textbf{HTML decode}

AWS WAF Classic replaces HTML-encoded characters with unencoded characters:

- Replaces \texttt{&quot;} with \\texttt{&}
- Replaces \texttt{\&nbsp;} with a non-breaking space
- Replaces \texttt{\&lt;} with \texttt{<}
- Replaces \texttt{\&gt;} with \texttt{>}
- Replaces characters that are represented in hexadecimal format, \texttt{\&\#hhhh}, with the corresponding characters
- Replaces characters that are represented in decimal format, \texttt{\&\#nnnn}, with the corresponding characters

\textbf{Normalize white space}

AWS WAF Classic replaces the following characters with a space character (decimal 32):

- \texttt{\f}, formfeed, decimal 12
- \texttt{\t}, tab, decimal 9
- \texttt{\n}, newline, decimal 10
- \texttt{\r}, carriage return, decimal 13
- \texttt{\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://console.aws.amazon.com/wafv2/.
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** (p. 136).
   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://console.aws.amazon.com/wafv2/.
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

**Note**

This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).

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 (p. 139)
- Values that you specify when you create or edit string match conditions (p. 140)
- Adding and deleting filters in a string match condition (p. 143)
- Deleting string match conditions (p. 144)

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://console.aws.amazon.com/wafv2/.
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 (p. 140).
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 part of a URL that identifies a resource, for example, /images/daily-ad.jpg. 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 (p. 130).

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, if 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 &quot; with \\n
• Replaces &nbsp; with a non-breaking space

• Replaces &amp; with <

• Replaces &gt; with >

• Replaces characters that are represented in hexadecimal format, &amp;#xhhhh; , with the corresponding characters

• Replaces characters that are represented in decimal format, &amp;#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://console.aws.amazon.com/wafv2/
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 (p. 140).
   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://console.aws.amazon.com/wafv2/.
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

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).

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 (p. 144)
- Values that you specify when you create or edit RegEx match conditions (p. 145)
- Editing a regex match condition (p. 148)

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://console.aws.amazon.com/wafv2/.
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 (p. 145).
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 part of a URL that identifies a resource, for example, /images/daily-ad.jpg. 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 (p. 130).

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, if 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 &quot; with \\&
- Replaces &nbsp; with a non-breaking space
- Replaces &lt; with <
- Replaces &gt; 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://console.aws.amazon.com/wafv2/.
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://console.aws.amazon.com/wafv2/.
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://console.aws.amazon.com/wafv2/.
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://console.aws.amazon.com/wafv2/.
2. Delete the filter from the regex condition. See To delete a filter from an existing regex match condition (p. 149) 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://console.aws.amazon.com/wafv2/.
2. Delete the filter from the regex condition you want to change. See To delete a filter from an existing regex match condition (p. 149) 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

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).
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 (p. 150)
- Adding and removing conditions in a rule (p. 151)
- Deleting a rule (p. 152)
- AWS marketplace rule groups (p. 153)

Creating a rule and adding conditions

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).

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://console.aws.amazon.com/wafv2/.
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 (p. 79).

   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

**Note**

This is **AWS WAF Classic** documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

**For the latest version of AWS WAF**, see [AWS WAF (p. 6)](https://console.aws.amazon.com/wafv2/).

You can change a rule by adding or removing conditions.

**To add or remove conditions in a rule**

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**

**Note**

This is **AWS WAF Classic** documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of **AWS WAF**, see [AWS WAF](#) (p. 6).

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**

2. To remove the rule from the web ACLs that are using it, perform the following steps:
   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.

   c. Choose **Edit web ACL**.

   d. Choose the X to the right of the rule that you want to remove from the web ACL, and then choose **Update**.

   e. Repeat for all of the remaining web ACLs that are using the rule that you want to delete.

3. In the navigation pane, choose **Rules**.

4. Select the name of the rule you want to delete.

5. Choose **Delete**.

## AWS marketplace rule groups

**Note**

This is **AWS WAF Classic** documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see [AWS WAF (p. 6)](https://docs.aws.amazon.com/waf/latest/developerguide/aws-waf.html).

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](https://www.owasp.org/index.php/Top_Ten_2013).

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](https://docs.aws.amazon.com/waf/latest/developerguide/pricing.html) 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 \texttt{COUNT}. Therefore, requests that match an excluded rule are counted but not blocked. You will receive \texttt{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 (p. 154).

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 \url{https://console.aws.amazon.com/wafv2/}.
2. In the navigation pane, choose Marketplace.
3. In the \textbf{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.

   \textbf{Note}
   
   If you don’t want to subscribe to this rule group, simply close this page in your browser.

5. Choose \textbf{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 (p. 157) or Editing a Web ACL (p. 161).

   \textbf{Note}
   
   When adding a rule group to a web ACL, the action that you set for the rule group (either \texttt{No override} or \texttt{Override to count}) is called the rule group override action. For more information, see Rule group override (p. 155).

To unsubscribe from an AWS marketplace rule group

1. Sign in to the AWS Management Console and open the AWS WAF console at \url{https://console.aws.amazon.com/wafv2/}.
2. Remove the rule group from all web ACLs. For more information, see Editing a Web ACL (p. 161).
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://console.aws.amazon.com/wafv2/.
2. If not already enabled, enable AWS WAF Classic logging. For more information, see Logging Web ACL traffic information (p. 165). 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 web ACL that you want to edit.
   
   **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 receiveCOUNTmetrics 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 a rule group (p. 212).

**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) (p. 155).
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**

**Note**

This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see [AWS WAF](#). (p. 6)

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](#) (p. 156).

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](#) (p. 162).

**Topics**

- [Deciding on the default action for a Web ACL](#) (p. 156)
- [Creating a Web ACL](#) (p. 157)
- [Associating or disassociating a Web ACL with an Amazon API Gateway API, a CloudFront distribution or an Application Load Balancer](#) (p. 160)
- [Editing a Web ACL](#) (p. 161)
- [Deleting a Web ACL](#) (p. 161)
- [Testing web ACLs](#) (p. 162)

**Deciding on the default action for a Web ACL**

**Note**

This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.
For the latest version of AWS WAF, see AWS WAF (p. 6).

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

**Note**

This is **AWS WAF Classic** documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see **AWS WAF** (p. 6).

**To create a web ACL**

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.
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 (p. 121)
* Working with IP match conditions (p. 126)
* Working with geographic match conditions (p. 128)
* Working with size constraint conditions (p. 130)
* Working with SQL injection match conditions (p. 134)
* Working with string match conditions (p. 139)
* Working with regex match conditions (p. 144)

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**

cycle 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 (p. 170).
  • **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 (p. 162).

• 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 specified by individual rules in the group, so that all matching requests are only counted.

For more information, see Rule group override (p. 155).

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 (p. 156).
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**

**Note**
This is **AWS WAF Classic** documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of **AWS WAF**, see [AWS WAF (p. 6)](https://docs.aws.amazon.com/waf/latest/developerguide/).

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:

- Web ACLs associated with an Application Load Balancer can be associated only with another Application Load Balancer in the same Region. The same is true for API Gateway APIs.
- Web ACLs associated with a CloudFront distribution cannot be associated with an Application Load Balancer. The web ACL can, however, be associated with other CloudFront distributions.
- Each API Gateway API, Application Load Balancer and CloudFront distribution can be associated with only one web ACL.

**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://console.aws.amazon.com/wafv2/.
2. In the navigation pane, choose **Web ACLs**.
3. Choose the web ACL that you want to associate with an API Gateway API, CloudFront distribution or Application Load Balancer.
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://console.aws.amazon.com/wafv2/.
2. In the navigation pane, choose **Web ACLs**.
3. Choose the web ACL that you want to disassociate from an API Gateway API, CloudFront distribution or Application Load Balancer.
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**

*Note*
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see AWS WAF (p. 6).

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://console.aws.amazon.com/wafv2/.
2. In the navigation pane, choose Web ACLs.
3. Choose the web ACL that you want to edit.
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 (p. 155)
9. Choose Save changes.

**Deleting a Web ACL**

*Note*
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see AWS WAF (p. 6).

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://console.aws.amazon.com/wafv2/.
2. In the navigation pane, choose Web ACLs.
3. Choose the web ACL that you want to delete.
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

**Note**

This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see [AWS WAF](p. 6).

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 (p. 162)
- Viewing a sample of the web requests that API Gateway CloudFront or an Application Load Balancer has forwarded to AWS WAF Classic (p. 164)

### 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
AWS WAF, AWS Firewall Manager, and
AWS Shield Advanced Developer Guide
Working with web ACLs

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 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://console.aws.amazon.com/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](p. 164).

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)](p. 119).

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](p. 161).
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://console.aws.amazon.com/wafv2/.
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 part of a URL that identifies a resource, for example, /images/daily-ad.jpg.

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.

Logging Web ACL traffic information

Note

This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see AWS WAF (p. 6).

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 more information about how to create an Amazon Kinesis Data Firehose and review the stored logs, see What Is 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 (p. 192).

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 Kinesis Data Firehose Delivery Stream.
Important
Do not choose Kinesis stream as your source.
One AWS WAF Classic log is equivalent to one Kinesis Data 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 Kinesis Data Firehose. If you don't configure Kinesis Data 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://console.aws.amazon.com/wafv2/.
3. In the navigation pane, choose Web ACLs.
4. Choose the web ACL that you want to enable logging for.
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 XXX in the logs. For example, if you redact the cookie field, the cookie field in the logs will be XXX.
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 (p. 192).

To disable logging for a web ACL
1. In the navigation pane, choose Web ACLs.
2. Choose the web ACL that you want to disable logging for.
3. On the Logging tab, choose Disable logging.
4. In the dialog box, choose Disable logging.

Example Example log

```json
{
  "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": []
```
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.

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.

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

Note

This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.
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://console.aws.amazon.com/wafv2/.
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

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see AWS WAF (p. 6).

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 (p. 170)
- Using AWS WAF Classic with CloudFront geo restriction (p. 171)
- Using AWS WAF Classic with CloudFront for applications running on your own HTTP server (p. 171)
- Choosing the HTTP methods that CloudFront responds to (p. 172)

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 geo restriction

You can use the Amazon CloudFront geo restriction feature, also known as geoblocking, to prevent users in specific geographic locations from accessing content that you distribute through a CloudFront web distribution. If you want to block web requests from specific countries and also block requests based on other conditions, you can use CloudFront geo restriction in conjunction with AWS WAF Classic. CloudFront returns the same HTTP status code to viewers—HTTP 403 (Forbidden)—whether they try to access your content from a country on a CloudFront geo restriction blacklist or whether the request is blocked by AWS WAF Classic.

Note
You can see the two-letter country code of the country that requests originate from in the sample of web requests for a web ACL. 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 (p. 164).

For more information about CloudFront geo restriction, see Restricting the Geographic Distribution of Your Content 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 (p. 139). 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

**Note**

This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see AWS WAF (p. 6).

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

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see AWS WAF (p. 6).

AWS WAF Classic conforms to the AWS shared responsibility model, which includes regulations and guidelines for data protection. AWS is responsible for protecting the global infrastructure that runs all the AWS services. AWS maintains control over data hosted on this infrastructure, including the security configuration controls for handling customer content and personal data. AWS customers and APN partners, acting either as data controllers or data processors, are responsible for any personal data that they put in the AWS Cloud.

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.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM), so that 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.
- Set up API and user activity logging with AWS CloudTrail. See AWS CloudTrail API Reference.
- 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 personal data stored in Amazon S3.

We strongly recommend that you never put sensitive identifying information, such as your customers' account numbers, into free-form 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 piece of data that you enter into AWS WAF Classic or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server.

For more information about data protection, see the AWS Shared Responsibility Model and GDPR blog post on the AWS Security Blog.

Identity and access management in AWS WAF Classic

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.
For the latest version of AWS WAF, see AWS WAF (p. 6).

Access to AWS WAF Classic requires credentials. Those credentials must have permissions to access AWS resources, such as an AWS WAF Classic resource or an Amazon S3 bucket. The following sections provide details on how you can use AWS Identity and Access Management (IAM) and AWS WAF Classic to help secure access to your resources.

- Authentication (p. 174)
- Access control (p. 175)

**Authentication**

You can access AWS as any of the following types of identities:

- **AWS account root user** – When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

- **IAM user** – An IAM user is an identity within your AWS account that has specific custom permissions (for example, permissions to create a rule in AWS WAF Classic). You can use an IAM user name and password to sign in to secure AWS webpages like the AWS Management Console, AWS Discussion Forums, or the AWS Support Center.

In addition to a user name and password, you can also generate access keys for each user. You can use these keys when you access AWS services programmatically, either through one of the several SDKs or by using the AWS Command Line Interface (CLI). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don’t use AWS tools, you must sign the request yourself. AWS WAF Classic supports Signature Version 4, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 Signing Process in the AWS General Reference.

- **IAM role** – An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user in that it is an AWS identity with permissions policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session. IAM roles with temporary credentials are useful in the following situations:

  - **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as federated users. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated Users and Roles in the IAM User Guide.

  - **AWS service access** – A service role is an IAM role that a service assumes to perform actions in your account on your behalf. When you set up some AWS service environments, you must define a role for the service to assume. This service role must include all the permissions that are required for
the service to access the AWS resources that it needs. Service roles vary from service to service, but many allow you to choose your permissions as long as you meet the documented requirements for that service. Service roles provide access only within your account and cannot be used to grant access to services in other accounts. You can create, modify, and delete a service role from within IAM. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data from that bucket into an Amazon Redshift cluster. For more information, see Creating a Role to Delegate Permissions to an AWS Service in the IAM User Guide.

- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances in the IAM User Guide.

**Access control**

You can have valid credentials to authenticate your requests, but unless you have permissions you can’t create or access AWS WAF Classic resources. For example, you must have permissions to create an AWS WAF Classic web ACL or rule.

The following sections describe how to manage permissions for AWS WAF Classic. We recommend that you read the overview first.

- Overview of managing access permissions to your AWS WAF Classic resources (p. 176)
- Using identity-based policies (IAM policies) for AWS WAF Classic (p. 179)
- AWS WAF Classic API permissions: Actions, resources, and conditions reference (p. 184)

**AWS Identity and Access Management**

**Note**

This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).

AWS WAF Classic integrates with AWS Identity and Access Management (IAM), a service that lets your organization do the following:

- Create users and groups under your organization’s AWS account
- Share your AWS account resources with users in the account
- Assign unique security credentials to each user
- Control user access to services and resources

For example, you can use IAM with AWS WAF Classic to control which users in your AWS account can create a new web ACL.

For general information about IAM, see the following documentation:

- AWS Identity and Access Management (IAM)
- IAM Getting Started Guide
Overview of managing access permissions to your AWS WAF Classic resources

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).

Every AWS resource is owned by an AWS account, and permissions to create or access a resource are governed by permissions policies. An account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles). Some services also support attaching permissions policies to resources.

Note
An account administrator (or administrator user) is a user with administrator privileges. For more information, see IAM Best Practices in the IAM User Guide.

When granting permissions, you decide who is getting the permissions, the resources they get permissions for, and the specific operations that you want to allow on those resources.

Topics

• AWS WAF Classic resources and operations (p. 176)
• Understanding resource ownership (p. 177)
• Managing access to resources (p. 177)
• Specifying policy elements: Actions, effects, resources, and principals (p. 179)
• Specifying conditions in a policy (p. 179)

AWS WAF Classic resources and operations

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.

<table>
<thead>
<tr>
<th>Name in AWS WAF Console</th>
<th>Name in AWS WAF SDK/CLI</th>
<th>ARN Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web ACL</td>
<td>WebACL</td>
<td>arn:aws:waf::account:webacl/ID</td>
</tr>
<tr>
<td>Rule</td>
<td>Rule</td>
<td>arn:aws:waf::account:rule/ID</td>
</tr>
<tr>
<td>String match condition</td>
<td>ByteMatchSet</td>
<td>arn:aws:waf::account:bytematchset/ID</td>
</tr>
<tr>
<td>SQL injection match condition</td>
<td>SqlInjectionMatchSet</td>
<td>arn:aws:waf::account:sqlinjectionset/ID</td>
</tr>
<tr>
<td>Size constraint condition</td>
<td>SizeConstraintSet</td>
<td>arn:aws:waf::account:sizeconstraintset/ID</td>
</tr>
</tbody>
</table>
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:

\[
\text{arn:aws:waf::account:resource/ID}
\]

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:

\[
\text{arn:aws:waf::111122223333:webacl/*}
\]

For more information, see Resources in the IAM User Guide.

AWS WAF Classic provides a set of operations to work with AWS WAF Classic resources. For a list of available operations, see Actions.

### Understanding resource ownership

A resource owner is the AWS account that creates the resource. That is, the resource owner is the AWS account of the principal entity (the root account, an IAM user, or an IAM role) that authenticates the request that creates the resource. The following examples illustrate how this works:

- If you use the root account credentials of your AWS account to create an AWS WAF Classic resource, your AWS account is the owner of the resource.
- If you create an IAM user in your AWS account and grant permissions to create an AWS WAF Classic resource to that user, the user can create an AWS WAF Classic resource. However, your AWS account, to which the user belongs, owns the AWS WAF Classic resource.
- If you create an IAM role in your AWS account with permissions to create an AWS WAF Classic resource, anyone who can assume the role can create an AWS WAF Classic resource. Your AWS account, to which the role belongs, owns the AWS WAF Classic resource.

### Managing access to resources

A permissions policy describes who has access to what. The following sections explain the available options for creating permissions policies.

**Note**

These sections discuss using IAM in the context of AWS WAF Classic. It doesn’t provide detailed information about the IAM service. For complete IAM documentation, see What Is IAM? in the
Policies that are attached to an IAM identity are known as **identity-based** policies, and policies that are attached to a resource are known as **resource-based** policies. AWS WAF Classic supports only identity-based policies.

**Topics**
- Identity-based policies (IAM policies) (p. 178)
- Resource-based policies (p. 179)

**Identity-based policies (IAM policies)**

You can attach policies to IAM identities. For example, you can do the following:

- **Attach a permissions policy to a user or a group in your account** – An account administrator can use a permissions policy that is associated with a particular user to grant permissions for that user to create an AWS WAF Classic resource.

- **Attach a permissions policy to a role (grant cross-account permissions)** – You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions. For example, the administrator in Account A can create a role to grant cross-account permissions to another AWS account (for example, Account B) or an AWS service as follows:
  1. Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions on resources in Account A.
  2. Account A administrator attaches a trust policy to the role identifying Account B as the principal who can assume the role.
  3. Account B administrator can then delegate permissions to assume the role to any users in Account B. Doing this allows users in Account B to create or access resources in Account A. The principal in the trust policy also can be an AWS service principal if you want to grant an AWS service permissions to assume the role.

  For more information about using IAM to delegate permissions, see **Access Management** in the IAM User Guide.

  The following is an example policy that grants permissions for the `waf:ListRules` action on all resources. In the current implementation, AWS WAF Classic 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 must specify a wildcard character (`*`):

  ```json
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "ListRules",
        "Effect": "Allow",
        "Action": ["waf:ListRules"],
        "Resource": "*"
      }
    ]
  }
  ```

  For more information about using identity-based policies with AWS WAF Classic, see **Using identity-based policies (IAM policies) for AWS WAF Classic** (p. 179). For more information about users, groups, roles, and permissions, see **Identities (Users, Groups, and Roles)** in the IAM User Guide.
Resource-based policies

Other services, such as Amazon S3, also support resource-based permissions policies. For example, you can attach a policy to an S3 bucket to manage access permissions to that bucket. AWS WAF doesn't support resource-based policies.

Authorization based on AWS WAF Classic tags

You can attach tags to AWS WAF Classic resources or pass tags in a request to AWS WAF Classic. To control access based on tags, you provide tag information in the condition element of a policy. For more information about tagging your resources, see Working with Tag Editor.

Specifying policy elements: Actions, effects, resources, and principals

For each AWS WAF Classic resource (see AWS WAF Classic resources and operations (p. 176)), the service defines a set of API operations (see AWS WAF Classic API permissions: Actions, resources, and conditions reference (p. 184)). To grant permissions for these API operations, AWS WAF Classic defines a set of actions that you can specify in a policy. Note that performing an API operation can require permissions for more than one action. When granting permissions for specific actions, you also identify the resource on which the actions are allowed or denied.

The following are the most basic policy elements:

- **Resource** – In a policy, you use an Amazon Resource Name (ARN) to identify the resource to which the policy applies. For more information, see AWS WAF Classic resources and operations (p. 176).
- **Action** – You use action keywords to identify resource operations that you want to allow or deny. For example, the waf:CreateRule permission allows the user permissions to perform the AWS WAF Classic CreateRule operation.
- **Effect** – You specify the effect when the user requests the specific action. This can be either allow or deny. If you don't explicitly grant access to allow a resource, access is implicitly denied. You also can explicitly deny access to a resource, which you might do to make sure that a user cannot access it, even if a different policy grants access.
- **Principal** – In identity-based policies (IAM policies), the user that the policy is attached to is the implicit principal. AWS WAF doesn't support resource-based policies.

To learn more about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the IAM User Guide.

For a table that shows all the AWS WAF Classic API actions and the resources that they apply to, see AWS WAF Classic API permissions: Actions, resources, and conditions reference (p. 184).

Specifying conditions in a policy

When you grant permissions, you can use the IAM policy language to specify the conditions when a policy should take effect. For example, you might want a policy to be applied only after a specific date. For more information about specifying conditions in a policy language, see Condition in the IAM User Guide.

To express conditions, you use predefined condition keys. There are no condition keys specific to AWS WAF Classic. However, there are AWS-wide condition keys that you can use as appropriate. For a complete list of AWS-wide keys, see Available Keys for Conditions in the IAM User Guide.

Using identity-based policies (IAM policies) for AWS WAF Classic

**Note**

This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.
For the latest version of AWS WAF, see AWS WAF (p. 6).

This section provides examples of identity-based policies that demonstrate how an account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles) and thereby grant permissions to perform operations on AWS WAF Classic resources.

**Important**

We recommend that you first review the introductory topics that explain the basic concepts and options available for you to manage access to your AWS WAF Classic resources. For more information, see Overview of managing access permissions to your AWS WAF Classic resources (p. 176).

The following shows an example of a permissions policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateFunctionPermissions",
      "Effect": "Allow",
      "Action": [
        "waf:ListWebACLs",
        "waf:ListRules",
        "waf:GetWebACL",
        "waf:GetRule",
        "cloudwatch:ListMetrics",
        "waf:GetSampledRequests"
      ],
      "Resource": "*"
    },
    {
      "Sid": "PermissionToPassAnyRole",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::account-id:role/*"
    }
  ]
}
```

The policy has two statements:

- The first statement grants permissions to view statistics for AWS WAF Classic web ACLs, using the `waf:ListWebACLs`, `waf:ListRules`, `waf:GetWebACL`, `waf:GetRule`, `cloudwatch:ListMetrics`, and `waf:GetSampledRequests` actions. AWS WAF Classic doesn't support permissions for some of these actions at the resource level. Therefore, the policy specifies a wildcard character (*) as the `Resource` value.

- The second statement grants permissions for the IAM action `iam:PassRole` on IAM roles. The wildcard character (*) at the end of the `Resource` value means that the statement allows permissions for the `iam:PassRole` action on any IAM role. To only extend these permissions to a specific role, replace the wildcard character (*) in the resource ARN with the specific role name.

The policy doesn't specify the Principal element because in an identity-based policy you don't specify the principal who gets the permissions. When you attach a policy to a user, the user is the implicit principal. When you attach a permissions policy to an IAM role, the principal identified in the role's trust policy gets the permissions.

For a table that shows all the AWS WAF Classic API actions and the resources that they apply to, see AWS WAF Classic API permissions: Actions, resources, and conditions reference (p. 184).
Topics

- Permissions required to use the AWS WAF Classic console (p. 181)
- AWS managed (predefined) policies for AWS WAF Classic (p. 181)
- Customer managed policy examples (p. 181)

Permissions required to use the AWS WAF Classic console

The AWS WAF Classic console provides an integrated environment for you to create and manage AWS WAF Classic resources. The console provides many features and workflows that often require permissions to create an AWS WAF Classic resource in addition to the API-specific permissions that are documented in the AWS WAF Classic API permissions: Actions, resources, and conditions reference (p. 184). For more information about these additional console permissions, see Customer managed policy examples (p. 181).

AWS managed (predefined) policies for AWS WAF Classic

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. Managed policies grant necessary permissions for common use cases so you can avoid having to investigate what permissions are needed. For more information, see AWS Managed Policies in the IAM User Guide.

The following AWS managed policies, which you can attach to users in your account, are specific to AWS WAF Classic and are grouped by use case scenario:

- AWSWAFReadOnlyAccess – Grants read-only access to AWS WAF Classic resources.
- AWSWAFFullAccess – Grants full access to AWS WAF Classic resources.

Note
You can review these permissions policies by signing in to the IAM console and searching for specific policies there.

You also can create your own custom IAM policies to allow permissions for AWS WAF Classic API operations and resources. You can attach these custom policies to the IAM users or groups that require those permissions or to custom execution roles (IAM roles) that you create for your AWS WAF Classic resources.

Customer managed policy examples

The examples in this section provide a group of sample policies that you can attach to a user. If you are new to creating policies, we recommend that you first create an IAM user in your account and attach the policies to the user, in the sequence outlined in the steps in this section.

You can use the console to verify the effects of each policy as you attach the policy to the user. Initially, the user doesn't have permissions, and the user won't be able to do anything on the console. As you attach policies to the user, you can verify that the user can perform various operations on the console.

We recommend that you use two browser windows: one to create the user and grant permissions, and the other to sign in to the AWS Management Console using the user's credentials and verify permissions as you grant them to the user.

For examples that show how to create an IAM role that you can use as an execution role for your AWS WAF Classic resource, see Creating IAM Roles in the IAM User Guide.

Example topics

- Example 1: Give users read-only access to AWS WAF Classic, CloudFront, and CloudWatch (p. 182)
Identity and access management

- Example 2: Give users full access to AWS WAF Classic, CloudFront, and CloudWatch (p. 182)
- Example 3: Granting access to a specified AWS account (p. 183)
- Example 4: Granting access to a specified Web ACL (p. 183)

Create an IAM user

First, you need to create an IAM user, add the user to an IAM group with administrative permissions, and then grant administrative permissions to the IAM user that you created. You then can access AWS using a special URL and the user's credentials.

For instructions, see Creating Your First IAM User and Administrators Group in the IAM User Guide.

Example 1: Give users read-only access to AWS WAF Classic, CloudFront, and CloudWatch

The following policy grants users read-only access to AWS WAF Classic 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 Classic conditions, rules, and web ACLs to see which distribution is associated with a web ACL, and to monitor metrics and a sample of requests in CloudWatch. These users can't create, update, or delete AWS WAF Classic resources.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "waf:Get*",
        "waf:List*",
        "cloudfront:GetDistribution",
        "cloudfront:GetDistributionConfig",
        "cloudfront:ListDistributions",
        "cloudfront:ListDistributionsByWebACLId",
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricStatistics"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Example 2: Give users full access to AWS WAF Classic, CloudFront, and CloudWatch

The following policy lets users perform any AWS WAF Classic 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 Classic administrators.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "waf:*",
        "cloudfront:CreateDistribution",
        "cloudfront:GetDistribution",
        "cloudfront:GetDistributionConfig",
        "cloudfront:UpdateDistribution",
        "cloudfront:ListDistributions",
        "cloudfront:ListDistributionsByWebACLId",
        "cloudfront:DeleteDistribution",
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricStatistics"
      ],
      "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.

**Example 3: Granting access to a specified AWS account**

This policy grants the following permissions to the account 444455556666:

- Full access to all AWS WAF Classic operations and resources.
- Read and update access to all CloudFront distributions, which allows you to associate 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 Classic console.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "waf:*"
      ],
      "Resource": [
        "arn:aws:waf::444455556666:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudfront:GetDistribution",
        "cloudfront:GetDistributionConfig",
        "cloudfront:ListDistributions",
        "cloudfront:ListDistributionsByWebACLId",
        "cloudfront:UpdateDistribution",
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricStatistics"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

**Example 4: Granting access to a specified Web ACL**

This policy grants the following permissions to the webacl ID 112233d7c-86b2-458b-af83-51c51example in the account 444455556666:

- Full access to AWS WAF Classic Get, Update, and Delete operations and resources

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "waf:GetWebACL",
        "waf:UpdateWebACL",
        "waf:DeleteWebACL"
      ],
      "Resource": [
        "arn:aws:waf::444455556666:webacl/112233d7c-86b2-458b-af83-51c51example"
      ]
    }
  ]
}
```
AWS WAF Classic API permissions: Actions, resources, and conditions reference

**Note**
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. **For the latest version of AWS WAF, see AWS WAF (p. 6).**

When you set up Access control (p. 175) and writing permissions policies that you can attach to an IAM identity (identity-based policies), you can use the following table as a reference. The table lists each AWS WAF Classic API operation, the corresponding actions for which you can grant permissions to perform the action, and the AWS resource for which you can grant the permissions. You specify the actions in the policy's `Action` field, and you specify the resource value in the policy's `Resource` field.

You can use AWS-wide condition keys in your AWS WAF Classic policies to express conditions. For a complete list of AWS-wide keys, see Available Keys for Conditions in the IAM User Guide.

**Note**
To specify an action, use the `waf:` prefix followed by the API operation name (for example, `waf:CreateIPSet`).

AWS WAF Classic API and required permissions for actions

**CreateByteMatchSet**

**Action:** `waf:CreateByteMatchSet`

**Resource:**
- Global (for Amazon CloudFront): `arn:aws:waf::account-id:bytematchset/entity-ID`

**CreateIPSet**

**Action:** `waf:CreateIPSet`

**Resource:**
- Global (for Amazon CloudFront): `arn:aws:waf::account-id:ipset/entity-ID`

**CreateRule**

**Action:** `waf:CreateRule`
Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:rule/entity-ID


CreateRateBasedRule

Action: waf:CreateRateBasedRule

Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:rule/entity-ID


CreateSizeConstraintSet

Action: waf:CreateSizeConstraintSet

Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:sizeconstraintset/entity-ID


CreateSqlInjectionMatchSet

Action: waf:CreateSqlInjectionMatchSet

Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:sqlinjectionmatchset/entity-ID


CreateWebACL

Action: waf:CreateWebACL

Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:webacl/entity-ID


CreateXssMatchSet

Action: waf:CreateXssMatchSet

Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:xssmatchset/entity-ID


DeleteByteMatchSet

Action: waf:DeleteByteMatchSet,
Resource:
Global (for Amazon CloudFront): arn:aws:waf::account-id:bytematchset/entity-ID

DeleteIPSet
Action: waf:DeleteIPSet

Resource:
Global (for Amazon CloudFront): arn:aws:waf::account-id:ipset/entity-ID

DeleteRule
Action: waf:DeleteRule

Resource:
Global (for Amazon CloudFront): arn:aws:waf::account-id:rule/entity-ID

DeleteRateBasedRule
Action: waf:DeleteRateBasedRule

Resource:
Global (for Amazon CloudFront): arn:aws:waf::account-id:rule/entity-ID

DeleteSizeConstraintSet
Action: waf:DeleteSizeConstraintSet

Resource:
Global (for Amazon CloudFront): arn:aws:waf::account-id:sizeconstraintset/entity-ID

DeleteSqlInjectionMatchSet
Action: waf:DeleteSqlInjectionMatchSet

Resource:
Global (for Amazon CloudFront): arn:aws:waf::account-id:sqlinjectionmatchset/entity-ID

DeleteWebACL
Action: waf:DeleteWebACL
Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:webacl/entity-ID


DeleteXssMatchSet

Action: waf:DeleteXssMatchSet

Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:xssmatchset/entity-ID


GetByteMatchSet

Action: waf:GetByteMatchSet

Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:bytematchset/entity-ID


GetChangeToken

Action: waf:GetChangeToken

Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:changetoken/entity-ID


GetChangeTokenStatus

Action: waf:GetChangeTokenStatus

Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:changetoken/token-ID


GetIPSet

Action: waf:GetIPSet

Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:ipset/entity-ID


GetRule

Action: waf:GetRule
Resource:

Global (for Amazon CloudFront): arn:aws:waf::<account-id>rule/entity-ID
Regional (for an Application Load Balancer): arn:aws:waf-regional:region:<account-id>rule/entity-ID

GetRateBasedRule

Action: waf:GetRateBasedRule

Resource:

Global (for Amazon CloudFront): arn:aws:waf::<account-id>rule/entity-ID
Regional (for an Application Load Balancer): arn:aws:waf-regional:region:<account-id>rule/entity-ID

GetRateBasedRuleManagedKeys

Action: waf:GetRateBasedRuleManagedKeys

Resource:

Global (for Amazon CloudFront): arn:aws:waf::<account-id>rule/entity-ID
Regional (for an Application Load Balancer): arn:aws:waf-regional:region:<account-id>rule/entity-ID

GetSampledRequests

Action: waf:GetSampledRequests

Resource: Resource depends on the parameters that are specified in the API call. You must have access to the rule or web ACL that corresponds to the request for samples. For example:

Global (for Amazon CloudFront): arn:aws:waf::<account-id>rule/example1 or arn:aws:waf::<account-id>webacl/example2
Regional (for an Application Load Balancer): arn:aws:waf-regional:region:<account-id>rule/example1 or arn:aws:waf-regional:region:<account-id>webacl/example2

GetSizeConstraintSet

Action: waf:GetSizeConstraintSet

Resource:

Global (for Amazon CloudFront): arn:aws:waf::<account-id>sizeconstraintset/entity-ID
Regional (for an Application Load Balancer): arn:aws:waf-regional:region:<account-id>sizeconstraintset/entity-ID

GetSqlInjectionMatchSet

Action: waf:GetSqlInjectionMatchSet

Resource:

Global (for Amazon CloudFront): arn:aws:waf::<account-id>sqlinjectionmatchset/entity-ID
Regional (for an Application Load Balancer): arn:aws:waf-regional:region:<account-id>sqlinjectionmatchset/entity-ID
GetWebACL

**Action:** waf:GetWebACL

**Resource:**
Global (for Amazon CloudFront): arn:aws:waf::account-id:webacl/entity-ID

GetXssMatchSet

**Action:** waf:GetXssMatchSet

**Resource:**
Global (for Amazon CloudFront): arn:aws:waf::account-id:xssmatchset/entity-ID

ListByteMatchSets

**Action:** waf:ListByteMatchSets

**Resource:**
Global (for Amazon CloudFront): arn:aws:waf::account-id:bytematchset/entity-ID

ListIPSets

**Action:** waf:ListIPSets

**Resource:**
Global (for Amazon CloudFront): arn:aws:waf::account-id:ipset/entity-ID

ListRules

**Action:** waf:ListRules

**Resource:**
Global (for Amazon CloudFront): arn:aws:waf::account-id:rule/entity-ID

ListRateBasedRules

**Action:** waf:ListRateBasedRules

**Resource:**
Global (for Amazon CloudFront): arn:aws:waf::account-id:rule/entity-ID
ListSizeConstraintSets

**Action:** waf:ListSizeConstraintSets

**Resource:**

Global (for Amazon CloudFront): `arn:aws:waf::account-id:sizeconstraintset/entity-ID`


ListSqlInjectionMatchSets

**Action:** waf:ListSqlInjectionMatchSets

**Resource:**

Global (for Amazon CloudFront): `arn:aws:waf::account-id:sqlinjectionmatchset/entity-ID`


ListWebACLs

**Action:** waf:ListWebACLs

**Resource:**

Global (for Amazon CloudFront): `arn:aws:waf::account-id:webacl/entity-ID`


ListXssMatchSets

**Action:** waf:ListXssMatchSets

**Resource:**

Global (for Amazon CloudFront): `arn:aws:waf::account-id:xssmatchset/entity-ID`


UpdateByteMatchSet

**Action:** waf:UpdateByteMatchSet

**Resource:**

Global (for Amazon CloudFront): `arn:aws:waf::account-id:bytematchset/entity-ID`


UpdateIPSet

**Action:** waf:UpdateIPSet

**Resource:**

Global (for Amazon CloudFront): `arn:aws:waf::account-id:ipset/entity-ID`

UpdateRule

Action: waf:UpdateRule

Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:rule/entity-ID


UpdateRateBasedRule

Action: waf:UpdateRateBasedRule

Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:rule/entity-ID


UpdateSizeConstraintSet

Action: waf:UpdateSizeConstraintSet

Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:sizeconstraintset/entity-ID


UpdateSqlInjectionMatchSet

Action: waf:UpdateSqlInjectionMatchSet

Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:sqlinjectionmatchset/entity-ID


UpdateWebACL

Action: waf:UpdateWebACL

Resource:

Global (for Amazon CloudFront): arn:aws:waf::account-id:webacl/entity-ID


UpdateXssMatchSet

Action: waf:UpdateXssMatchSet

Resource:
Global (for Amazon CloudFront): arn:aws:waf::account-id:xssmatchset/entity-ID


**Using service-linked roles for AWS WAF Classic**

**Note**
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).

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
- AWSServiceRoleForWAFFRegionalLogging

AWS WAF Classic uses these service-linked roles to write logs to Amazon Kinesis Data Firehose. These roles are used only if you enable logging in AWS WAF. For more information, see Logging Web ACL traffic information (p. 165).

The AWSServiceRoleForWAFLogging and AWSServiceRoleForWAFFRegionalLogging 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 Kinesis Data Firehose data stream resources with a name that starts with "aws-waf-logs-". For example, aws-waf-logs-us-east-2-analytics.
Identity and access management

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 (p. 165).
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.

<table>
<thead>
<tr>
<th>Region Name</th>
<th>Region Identity</th>
<th>Support in AWS WAF Classic</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (N. Virginia)</td>
<td>us-east-1</td>
<td>Yes</td>
</tr>
</tbody>
</table>

API Version 2019-07-29

193
<table>
<thead>
<tr>
<th>Region Name</th>
<th>Region Identity</th>
<th>Support in AWS WAF Classic</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (Ohio)</td>
<td>us-east-2</td>
<td>Yes</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>us-west-1</td>
<td>Yes</td>
</tr>
<tr>
<td>US West (Oregon)</td>
<td>us-west-2</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>ap-south-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Osaka-Local)</td>
<td>ap-northeast-3</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>ap-northeast-2</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>ap-southeast-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Sydney)</td>
<td>ap-southeast-2</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>ap-northeast-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Canada (Central)</td>
<td>ca-central-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>eu-central-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>eu-west-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (London)</td>
<td>eu-west-2</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>eu-west-3</td>
<td>Yes</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>sa-east-1</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Logging and monitoring in AWS WAF Classic

Note
This is AWS WAF Classic documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. For the latest version of AWS WAF, see AWS WAF (p. 6).

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 incidents:

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 (p. 282).

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 (p. 286).

AWS Trusted Advisor

Trusted Advisor draws upon best practices learned from serving hundreds of thousands of AWS customers. Trusted Advisor inspects your AWS environment and then makes recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps. All AWS customers have access to five Trusted Advisor checks. Customers with a Business or Enterprise support plan can view all Trusted Advisor checks. For more information, see AWS Trusted Advisor.
Compliance validation for AWS WAF Classic

**Note**
This is **AWS WAF Classic** documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version. **For the latest version of AWS WAF**, see [AWS WAF](p. 6).

Third-party auditors assess the security and compliance of AWS WAF Classic as part of multiple AWS compliance programs, such as SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see [AWS Services in Scope by Compliance Program](p. 6). For general information, see [AWS Compliance Programs](p. 6).

You can download third-party audit reports using AWS Artifact. For more information, see [Downloading Reports in AWS Artifact](p. 6).

Your compliance responsibility when using AWS WAF Classic is determined by the sensitivity of your data, your organization's compliance objectives, and applicable laws and regulations. If your use of AWS WAF Classic is subject to compliance with standards like HIPAA, PCI, or FedRAMP, AWS provides resources to help:

- **Security and Compliance Quick Start Guides** – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- **Architecting for HIPAA Security and Compliance Whitepaper** – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- **AWS Compliance Resources** – This collection of workbooks and guides might apply to your industry and location.
- **AWS Config** – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- **AWS Security Hub** – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.
- **AWS Well-Architected Framework** – The AWS Well-Architected Framework helps you build secure cloud applications.
Resilience in AWS WAF Classic

**Note**
This is **AWS WAF Classic** documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see AWS WAF (p. 6).

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

**Note**
This is **AWS WAF Classic** documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see AWS WAF (p. 6).

As a managed service, AWS WAF Classic is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access AWS WAF Classic through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

AWS WAF Classic quotas

**Note**
This is **AWS WAF Classic** documentation. If you created AWS WAF resources, like rules and web ACLs, in AWS WAF prior to November, 2019, and you have not migrated your web ACLs over yet, you need to use AWS WAF Classic to access those resources. Otherwise, do not use this version.

For the latest version of AWS WAF, see AWS WAF (p. 6).

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. You can request an increase to these.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web ACLs per AWS account per Region</td>
<td>50</td>
</tr>
<tr>
<td>Rules per AWS account per Region</td>
<td>100</td>
</tr>
<tr>
<td>Rate-based-rules per AWS account per Region</td>
<td>5</td>
</tr>
</tbody>
</table>

API Version 2019-07-29

197
### AWS WAF Classic quotas

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conditions per AWS account per Region</td>
<td>100 of each condition type (For example: 100 size constraint conditions, 100 IP match conditions, and so on. The exception is regex match conditions. You can have a maximum of 10 regex match conditions per account per Region. This quota cannot be increased.)</td>
</tr>
<tr>
<td>Requests per Second</td>
<td>10,000 per web ACL*</td>
</tr>
</tbody>
</table>

*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](https://docs.aws.amazon.com/omega-waF/latest/userguide/).*

The following quotas on AWS WAF Classic entities can't be changed.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule groups per web ACL</td>
<td>2: 1 customer-created rule group and 1 AWS Marketplace rule group</td>
</tr>
<tr>
<td>Rules per web ACL</td>
<td>10</td>
</tr>
<tr>
<td>Conditions per rule</td>
<td>10</td>
</tr>
<tr>
<td>IP address ranges (in CIDR notation) per IP match condition</td>
<td>10,000</td>
</tr>
<tr>
<td>IP addresses blocked per rate-based rule</td>
<td>10,000</td>
</tr>
<tr>
<td>Minimum rate-based rule rate limit per 5 minute period</td>
<td>100</td>
</tr>
<tr>
<td>Filters per cross-site scripting match condition</td>
<td>10</td>
</tr>
<tr>
<td>Filters per size constraint condition</td>
<td>10</td>
</tr>
<tr>
<td>Filters per SQL injection match condition</td>
<td>10</td>
</tr>
<tr>
<td>Filters per string match condition</td>
<td>10</td>
</tr>
<tr>
<td>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</td>
<td>40</td>
</tr>
</tbody>
</table>
### AWS WAF Classic quotas

<table>
<thead>
<tr>
<th>Resource</th>
<th>Quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>In string match conditions, the number of characters in the value that you want AWS WAF Classic to search for</td>
<td>50</td>
</tr>
<tr>
<td>In regex match conditions, the number of characters in the pattern that you want AWS WAF Classic to search for</td>
<td>70</td>
</tr>
<tr>
<td>In regex match conditions, the number of patterns per pattern set</td>
<td>10</td>
</tr>
<tr>
<td>In regex match conditions, the number of pattern sets per regex condition</td>
<td>1</td>
</tr>
<tr>
<td>The number of pattern sets per AWS account per Region</td>
<td>5</td>
</tr>
<tr>
<td>GeoMatchSets per AWS account per Region</td>
<td>50</td>
</tr>
<tr>
<td>Locations per GeoMatchSet</td>
<td>50</td>
</tr>
</tbody>
</table>
AWS Firewall Manager

AWS Firewall Manager simplifies your AWS WAF Classic, AWS Shield Advanced, and Amazon VPC security groups administration and maintenance tasks across multiple accounts and resources. With Firewall Manager, you set up your AWS WAF Classic firewall rules, Shield Advanced protections, and Amazon VPC security groups just once. The service automatically applies the rules and protections across your accounts and resources, even as you add new 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.

Topics

• AWS Firewall Manager pricing (p. 200)
• AWS Firewall Manager prerequisites (p. 201)
• Getting started with AWS Firewall Manager to enable AWS WAF Classic rules (p. 202)
• Getting started with AWS Firewall Manager to enable AWS Shield Advanced protection (p. 205)
• Getting started with AWS Firewall Manager Amazon VPC security group policies (p. 209)
• Working with rule groups (p. 211)
• Working with AWS Firewall Manager policies (p. 213)
• Tutorial: Creating a policy with hierarchical rules (p. 226)
• Viewing resource compliance with a policy (p. 228)
• AWS Firewall Manager findings (p. 228)
• Designating a different account as the AWS Firewall Manager administrator account (p. 231)
• Security in AWS Firewall Manager (p. 232)
• AWS Firewall Manager quotas (p. 246)

AWS Firewall Manager pricing

Charges incurred by AWS Firewall Manager are for the underlying services, such as AWS WAF Classic and AWS Config. For more information, see AWS Firewall Manager Pricing.
AWS Firewall Manager prerequisites

This topic shows you how to prepare your account to use AWS Firewall Manager. Before you use Firewall Manager for the first time, perform all the following steps in sequence.

Topics

• Step 1: Join AWS Organizations (p. 201)
• Step 2: Set the AWS Firewall Manager administrator account (p. 201)
• Step 3: Enable AWS Config (p. 202)

Step 1: Join AWS Organizations

To use Firewall Manager, your account must be a member of an organization in the AWS Organizations service. If your account is already a member, you can skip this step and go to Step 2: Set the AWS Firewall Manager administrator account (p. 201).

Note

AWS Organizations has two available feature sets: consolidated billing features and all features. To use Firewall Manager, the organization that you belong to must be enabled for all features. If your organization is configured only for consolidated billing, see Enabling All Features in Your Organization.

If your account is not part of an organization, create or join an organization as described in Creating and Managing an AWS Organizations.

After your account is a member of an organization, go to Step 2: Set the AWS Firewall Manager administrator account (p. 201).

Step 2: Set the AWS Firewall Manager administrator account

AWS Firewall Manager must be associated with the master account of your AWS organization or associated with a member account that has the appropriate permissions. The account that you associate with Firewall Manager is called the Firewall Manager administrator account.

For more information about AWS Organizations and master accounts, see Managing the AWS Accounts in Your Organization.

To set the Firewall Manager administrator account (console)

1. Sign in to the AWS Management Console using an existing AWS Organizations master account. You can sign in using the account's root user (not recommended) or another IAM user or IAM role within the account that has equivalent permissions.
3. Choose Get started.
4. Type an account ID to associate with Firewall Manager. This creates the Firewall Manager administrator account. The account ID can be the account that you are signed in with, or a different account. If the account ID that you type is not an AWS Organizations master account, Firewall Manager sets the appropriate permissions for the member account that you specify.

Note

The account that you enter in this step is given permission to create and manage AWS WAF Classic rules across all accounts within your organization.
5. Choose **Set administrator**.

After you set the Firewall Manager administrator account, go to **Step 3: Enable AWS Config (p. 202)**.

**Step 3: Enable AWS Config**

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](#).

**To enable AWS Config for Firewall Manager (console)**

1. Enable AWS Config for each of your AWS Organizations member accounts. 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 AWS CloudFormation template "Enable AWS Config" at [AWS CloudFormation StackSets Sample Templates](#). At a minimum, you must specify one or both of the following resource types to protect with AWS Firewall Manager:
   - Application Load Balancer – When enabling AWS Config to protect an Application Load Balancer, in the provided list of resource types, choose `ElasticLoadBalancingV2`.
   - CloudFront distribution – When enabling AWS Config to protect a CloudFront distribution, you must be in the US East (N. Virginia) Region. Other regions will not have CloudFront as an option.

You can now configure Firewall Manager to begin protecting your resources. For more information, see [Getting started with AWS Firewall Manager to enable AWS WAF Classic rules (p. 202)](#).

**Getting started with AWS Firewall Manager to enable AWS WAF Classic rules**

**Note**
The latest version of AWS WAF was released in November, 2019. AWS Firewall Manager currently works with the prior version, AWS WAF Classic, and not AWS WAF. For information about AWS WAF Classic, see [AWS WAF Classic (p. 76)](#).

You can use AWS Firewall Manager to enable AWS WAF Classic rules, AWS Shield Advanced protections, and Amazon VPC security groups. The steps for getting set up are slightly different for each. If you want to use Firewall Manager to enable AWS Shield Advanced protections, follow the steps in [Getting started with AWS Firewall Manager to enable AWS Shield Advanced protection (p. 205)](#). If you want to use Firewall Manager to enable Amazon VPC security groups, follow the steps in [Getting started with AWS Firewall Manager Amazon VPC security group policies (p. 209)](#).

To use Firewall Manager to enable AWS WAF Classic rules, perform the following steps in sequence.

**Topics**
- Step 1: Complete the prerequisites (p. 203)
- Step 2: Create rules (p. 203)
- Step 3: Create a rule group (p. 203)
- Step 4: Create and apply an AWS Firewall Manager AWS WAF Classic policy (p. 204)
Step 1: Complete the prerequisites

There are several mandatory steps to prepare your account for AWS Firewall Manager. Those steps are described in AWS Firewall Manager prerequisites (p. 201). Complete all the prerequisites before proceeding to Step 2: Create rules (p. 203).

Step 2: Create rules

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 (p. 203).

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 (p. 150).

You are now ready to go to Step 3: Create a rule group (p. 203).

Step 3: Create a rule group

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 Managed Rules or AWS Marketplace, and you can create your own rule groups. For information about managed rule groups, see AWS marketplace rule groups (p. 153).

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.
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.
Step 4: Create and apply an AWS Firewall Manager AWS WAF Classic policy

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 (p. 203)), 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 Next.
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.

Getting started with AWS Firewall Manager to enable AWS Shield Advanced protection

You can use AWS Firewall Manager to enable AWS WAF Classic rules, AWS Shield Advanced protections, and Amazon VPC security groups. The steps for getting set up are slightly different for each. If you want to use Firewall Manager to enable AWS WAF Classic rules, follow the steps in Getting started with AWS Firewall Manager to enable AWS WAF Classic rules (p. 202). If you want to use Firewall Manager to enable Amazon VPC security groups, follow the steps in Getting started with AWS Firewall Manager Amazon VPC security group policies (p. 209).

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 (p. 259).

To use Firewall Manager to enable Shield Advanced protection, perform the following steps in sequence.

Topics
- Step 1: Complete the prerequisites (p. 205)
- Step 2: Create and apply an AWS Firewall Manager Shield Advanced policy (p. 205)
- Step 3: (Optional) authorize the DDoS response team (p. 207)
- Step 4: Configure Amazon SNS notifications and Amazon CloudWatch alarms (p. 207)
- Step 5: Deploy AWS WAF Classic rules (p. 208)
- Step 6: Monitor the global threat environment dashboard (p. 209)

Step 1: Complete the prerequisites

There are several mandatory steps to prepare your account for AWS Firewall Manager. Those steps are described in AWS Firewall Manager prerequisites (p. 201). Complete all the prerequisites before proceeding to Step 2: Create and apply an AWS Firewall Manager Shield Advanced policy (p. 205).

Step 2: Create and apply an AWS Firewall Manager 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 (p. 259).
To create a Firewall Manager-Shield Advanced policy (console)

1. Sign in to the AWS Management Console using the 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 Step 2: Set the AWS Firewall Manager administrator account (p. 201).

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 more information, see AWS Shield pricing (p. 253).

   **Note**
   You don’t need to manually subscribe each member account to Shield Advanced. Firewall Manager does this for you as part of creating the policy.

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 friendly name.

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.

   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 (p. 259).

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 applies Shield Advanced protection to each applicable account within an organization in AWS Organizations, and associates the protection 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 doesn’t apply Shield Advanced protection to any resources. You must apply the policy to resources later.
Step 3: (Optional) authorize the DDoS response team

One of the benefits of AWS Shield Advanced is support from the DDoS response team (DRT). When you experience a potential DDoS attack, you can contact the AWS Support Center. If necessary, the Support Center escalates your issue to the DRT. The DRT helps you analyze the suspicious activity and assists you in mitigating the issue. This mitigation often involves creating or updating AWS WAF Classic rules and web ACLs in your account. The DRT 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 DRT 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 DRT at the account level. That is, the account owner, not the Firewall Manager administrator, must perform the following steps to authorize the DRT to mitigate potential attacks. The Firewall Manager administrator can authorize the DRT only for accounts that they own. Likewise, only the account owner can contact the DRT for support.

To authorize the DRT to mitigate potential attacks on your behalf, follow the instructions in Editing AWS Shield Advanced settings (p. 264). You can change DRT access and permissions at any time by using the same steps.

Continue to Step 4: Configure Amazon SNS notifications and Amazon CloudWatch alarms (p. 207).

Step 4: Configure Amazon SNS notifications and Amazon CloudWatch alarms

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.

To create an Amazon SNS topic in Firewall Manager (console)

1. Sign in to the AWS Management Console using the 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 Step 2: Set the AWS Firewall Manager administrator account (p. 201).

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**.

**Configure Amazon CloudWatch alarms**

Shield Advanced records metrics in CloudWatch that you can monitor. For more information, see [AWS Shield Advanced metrics and alarms](p. 284). CloudWatch incurs additional costs. For CloudWatch pricing, see [Amazon CloudWatch Pricing](p. 284).

To create a CloudWatch alarm, follow the instructions in [Using Amazon CloudWatch Alarms](p. 282). 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. These statistics are recorded for a period of two weeks, so that you can access historical information and gain a better perspective on how your web application or service is performing. For more information, see [What is CloudWatch](p. 284) in the [Amazon CloudWatch User Guide](p. 282).

For instructions about creating a CloudWatch dashboard, see [Monitoring with Amazon CloudWatch](p. 282). For information about specific Shield Advanced metrics that you can add to your dashboard, see [AWS Shield Advanced metrics and alarms](p. 284).

You can continue from this step without configuring Amazon SNS notifications or CloudWatch alarms. However, doing so significantly reduces your visibility of possible DDoS events.

After you configure the Amazon SNS notifications and CloudWatch alarms, we recommend that you follow the instructions in [Step 5: Deploy AWS WAF Classic rules](p. 208).

**Step 5: Deploy AWS WAF Classic rules**

Several resources are available to help you quickly deploy AWS WAF Classic rules. Consider taking advantage of one or more of the following offerings when creating your initial set of rules. You can then use these rules to set up a Firewall Manager-AWS WAF policy, as described in [Working with AWS Firewall Manager policies](p. 213).

**Security automation templates**

AWS provides preconfigured templates that include a set of AWS WAF Classic rules, which you can customize to best fit your needs. These templates are designed to block common web-based attacks such as bad bots, SQL injection, cross-site scripting (XSS), HTTP floods, and known-attacker attacks. In addition to activating Shield Advanced and specifying resources for Shield Advanced protection, you also should use these preconfigured templates.

For more information, see [AWS WAF Security Automations](p. 282). AWS WAF is included with Shield Advanced at no additional cost.

**AWS Marketplace rule groups**

AWS WAF provides [AWS Marketplace rule groups](p. 282) 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 Marketplace sellers. For more information, see [Managed rule groups](p. 21).

**AWS WAF for OWASP Top 10 Web Application Vulnerabilities**

This document outlines how you can use AWS WAF to mitigate the application vulnerabilities that are defined in the Open Web Application Security Project (OWASP) Top 10 list. This list shows the
most common categories of application security flaws. For more information, see AWS WAF Security Automations.

As a final step for getting started with Firewall Manager and Shield Advanced, review the global threat environment dashboard, as described in Step 6: Monitor the global threat environment dashboard (p. 209).

**Step 6: Monitor the global threat environment dashboard**

The global threat environment dashboard provides a near real-time summary of the global AWS threat landscape. The threat landscape includes the largest attack, the top attack vectors, and the relative number of significant attacks. To view the history of significant DDoS attacks, you can customize the dashboard for different time durations. For more information, see Monitoring threats across AWS (p. 299).

**Getting started with AWS Firewall Manager**

**Amazon VPC security group policies**

You can use AWS Firewall Manager to enable AWS WAF Classic rules, AWS Shield Advanced protections, and Amazon VPC security groups. The steps for getting set up are slightly different for each. If you want to use Firewall Manager to enable AWS WAF Classic rules, follow the steps in Getting started with AWS Firewall Manager to enable AWS WAF Classic rules (p. 202). If you want to use Firewall Manager to enable AWS Shield Advanced protections, follow the steps in Getting started with AWS Firewall Manager to enable AWS Shield Advanced protection (p. 205).

To use Firewall Manager to enable a security group across your organization, perform the following steps in sequence.

**Topics**

- **Step 1: Complete the prerequisites (p. 209)**
- **Step 2: Create a security group to use in your policy (p. 209)**
- **Step 3: Create and apply an AWS Firewall Manager common security group policy (p. 210)**

**Step 1: Complete the prerequisites**

There are several mandatory steps to prepare your account for AWS Firewall Manager. Those steps are described in AWS Firewall Manager prerequisites (p. 201). Complete all the prerequisites before proceeding to Step 2: Create a security group to use in your policy (p. 209).

**Step 2: Create 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.
Step 3: Create and apply an AWS Firewall Manager 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: Create and apply an AWS Firewall Manager common security group policy (p. 210).

Step 3: Create and apply an AWS Firewall Manager 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 How security group policies work in AWS Firewall Manager (p. 221).

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 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 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 friendly name.
10. Policy rules allow you to choose how the security groups in this policy are applied and maintained. For this tutorial, choose Apply the primary security groups to every resource within the policy scope. and leave the other 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. **Resources** allows you to narrow the scope of your policy by specifying resource tags for inclusion or exclusion. To use tagging, you need to first tag your resources. For more information about tagging your resources, see Working with Tag Editor. For this tutorial, choose **Include all resources that match the selected resource type**.

17. Choose **Next**.

18. Review your policy settings. 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 have, without making changes at this time.

19. 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 that **Automatic remediation** is disabled. 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 resource compliance with a policy (p. 228)

20. 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 How security group policies work in AWS Firewall Manager (p. 221).

---

### Working with rule groups

A **rule group** is a set of rules that you add to a web ACL or an 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 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 (p. 153).

**Topics**

- Creating a rule group (p. 211)
- Adding and deleting rules from a rule group (p. 212)

### Creating a rule group

When you create a 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.
Adding and deleting rules from a rule group

You can add or delete rules in a 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 Step 2: Set the AWS Firewall Manager administrator account (p. 201).

2. In the navigation pane, choose Rule groups.

3. Choose Create rule group.

   Note
   You can't add rate-based rules to a rule group.

4. 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.

5. Choose Next.

6. If you chose to create rules, follow the steps to create them at Creating a rule and adding conditions (p. 150).

   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.

7. Type a rule group name.

8. 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 (p. 79).

9. 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 (p. 157).
Working with AWS Firewall Manager policies

AWS Firewall Manager provides the following types of policies:

- **AWS WAF policy** – This policy contains a AWS WAF Classic rule group and defines which resources will be protected by that rule group.

  **Note**
  The latest version of AWS WAF was released in November, 2019. AWS Firewall Manager currently works with the prior version, AWS WAF Classic, and not AWS WAF. For information about AWS WAF Classic, see [AWS WAF Classic](p. 76)

- **Shield Advanced policy** – This policy applies AWS Shield Advanced protection to specified accounts and resources.

- **Amazon VPC security group policy** – This type of policy gives you control over security groups that are in use throughout your organization in AWS Organizations and lets you enforce a baseline set of rules across your organization.

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.

**Note**
A rule group is a set of rules, and each rule includes conditions that you specify. You can apply only one rule group to a policy, but you can apply the same rule group to multiple policies.

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.

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](p. 259).

**Topics**
- Creating an AWS Firewall Manager policy for AWS WAF Classic (p. 214)
- Creating an AWS Firewall Manager policy for shield advanced (p. 215)
- Creating an AWS Firewall Manager common security group policy (p. 216)
- Creating an AWS Firewall Manager content audit security group policy (p. 218)
Creating an AWS Firewall Manager policy for AWS WAF Classic

Note
The latest version of AWS WAF was released in November, 2019. AWS Firewall Manager currently works with the prior version, AWS WAF Classic, and not AWS WAF. For information about AWS WAF Classic, see AWS WAF Classic (p. 76)

To create a Firewall Manager policy for AWS WAF Classic (console)

1. Sign in to the AWS Management Console using the Firewall Manager administrator account that you set up in the prerequisites (AWS Firewall Manager prerequisites (p. 201)), 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 Step 2: Set the AWS Firewall Manager administrator account (p. 201).

2. In the navigation pane, choose Security policies.

3. Choose Create policy.

4. For Policy type, choose AWS WAF.

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. If you are creating a rule group, follow the instructions in Creating a rule group (p. 211). After you create the rule group, continue with the following steps.

8. Choose Next.

9. If you are creating a rule group, follow the instructions in Creating a rule group (p. 211). After you create the rule group, continue with the following steps.

10. Enter a policy name.

11. 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.

12. 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.

13. Choose Next.

14. 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 AWS organization. If you add a new account to the organization, Firewall Manager automatically applies the policy to that account.

15. Choose the type of resource that you want to protect.
16. 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), if a resource has any of those tags, it is considered a match.

   For more information about tags, see [Working with Tag Editor](#).

17. 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.

18. 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.

19. Choose **Next**.

20. 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 Shield Advanced

**To create a Firewall Manager policy for Shield Advanced (console)**

1. Sign in to the AWS Management Console using the Firewall Manager administrator account that you set up in the prerequisites ([AWS Firewall Manager prerequisites](#)), and then open the Firewall Manager console at [https://console.aws.amazon.com/wafv2/fms](https://console.aws.amazon.com/wafv2/fms).

   **Note**
   
   For information about setting up a Firewall Manager administrator account, see [Step 2: Set the AWS Firewall Manager administrator account](#).

2. In the navigation pane, choose **Security policies**.

3. Choose **Create policy**.

4. For **Name**, enter a friendly name.

5. 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 more information, see [AWS Shield pricing](#).

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 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**.
Creating an AWS Firewall Manager policy

9. Choose the type of resource that you want to protect.

If you don't select this option, Firewall Manager applies a policy to all accounts in your AWS organization. If you add a new account to the organization, Firewall Manager automatically applies the policy to that account.

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 (p. 259).

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 applies Shield Advanced protection to each applicable account within an AWS organization, and associates the protection 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 doesn't apply Shield Advanced protection to any resources. You must apply the policy to resources later.

12. Choose Next.

13. Review the new policy. To make any changes, choose Edit. When you are satisfied with the policy, choose Create policy.

Creating an AWS Firewall Manager common security group policy

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.

For information about how common security group policies work, see Common security group policies (p. 222).

To create a common security group policy (console)

1. Sign in to the AWS Management Console using the Firewall Manager administrator account that you set up in the prerequisites (AWS Firewall Manager prerequisites (p. 201)), 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 Step 2: Set the AWS Firewall Manager administrator account (p. 201).

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 options, choose the restrictions that you want to apply to the security group rules and the resources that are within policy scope.

   b. For **Primary security groups**, choose **Add primary security group**, and then choose the security group that you want to use. Firewall Manager populates the list of primary security groups from all Amazon VPC instances in the Firewall Manager administrator account. The default maximum number of primary security groups for a policy is one. For information on increasing the maximum, see [AWS Firewall Manager quotas](p. 246).

   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 affected by this policy**, if you want to apply the policy to all accounts in your organization, choose **Include all accounts under my organization**. If you want to include only specific accounts in the policy, or alternatively exclude specific accounts from the policy, choose one of those options, and then use the **AWS accounts** dialog box to add account numbers to include or exclude. You can choose only one option.

    **Note**
    For the choices **Include all accounts under my organization** and **Exclude the specified accounts and include all others**, when you add a new account to your organization, Firewall Manager automatically applies the policy to that account.

12. For **Resource type**, choose the types of resources that you want to protect.

    If you choose **EC2 instance**, you can choose to include all elastic network interfaces in each Amazon EC2 instance or just the default interface in each instance. If you have more than one elastic network interface in any in-scope Amazon EC2 instance, choosing the option to include all interfaces allows Firewall Manager to apply the policy to all of them. When you enable automatic remediation, if Firewall Manager can't apply the policy to all elastic network interfaces in an Amazon EC2 instance, it marks the instance as noncompliant.

13. For **Resources**, if you want to apply the policy to all resources within the AWS accounts and resource type parameters, choose **Include all resources that match the selected resource type**. If you want to include or exclude specific resources, use tagging to specify the resources, and then choose the appropriate option and add the tags to the list. You can apply the policy either to all resources except those that have all the tags that you specify, or you can apply it to only those that have all the tags that you specify. For more information about tagging your resources, see [Working with Tag Editor](p. 222).

    **Note**
    If you enter more than one tag, a resource must have all the tags to be a match.

14. Choose **Next**.

15. 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 [Common security group policies](p. 222).
Creating an AWS Firewall Manager content audit security group policy

To create a content audit security group policy, you must have a security group already created in your Firewall Manager administrator account that you want to use as the audit security group 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.

You can use the audit security group rules as a template for what rules are allowed by the policy or a template for what rules are denied by the policy. For information about how content audit security group policies work, see Content audit security group policies (p. 223).

To create a content audit security group policy (console)

1. Sign in to the AWS Management Console using the Firewall Manager administrator account that you set up in the prerequisites (AWS Firewall Manager prerequisites (p. 201)), 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 Step 2: Set the AWS Firewall Manager administrator account (p. 201).

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, do the following:
   a. From the rules options, choose whether to allow only the rules defined in the audit security groups or deny all the rules. For information on this choice, see Content audit security group policies (p. 223).
   b. 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 on increasing the quota, see AWS Firewall Manager quotas (p. 246).
   c. 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 affected by this policy, if you want to apply the policy to all accounts in your organization, choose Include all accounts under my organization. If you want to include only specific accounts in the policy, or alternatively exclude specific accounts from the policy, choose one of those options, and then use the AWS accounts dialog box to add account numbers to include or exclude. You can choose only one option.
   
   **Note**
   For the choices Include all accounts under my organization and Exclude the specified accounts and include all others, when you add a new account to your organization, Firewall Manager automatically applies the policy to that account.
12. For Resource type, choose the types of resource that you want to protect.
13. For **Resources**, if you want to apply the policy to all resources within the AWS accounts and resource type parameters, choose *Include all resources that match the selected resource type*. If you want to include or exclude specific resources, use tagging to specify the resources, and then choose the appropriate option and add the tags to the list. You can apply the policy either to all resources except those that have all the tags that you specify, or you can apply it to only those that have all the tags that you specify. For more information about tagging your resources, see [Working with Tag Editor](#).

   **Note**
   If you enter more than one tag, a resource must have all the tags to be a match.

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 [Content audit security group policies](#).

**Creating an AWS Firewall Manager usage audit security group policy**

AWS Firewall Manager usage audit security group policies allow you to monitor your organization for unused and redundant security groups and optionally perform cleanup. For information about how usage audit security group policies work, see [Usage audit security group policies](#).

**To create a usage audit security group policy (console)**

1. Sign in to the AWS Management Console using the Firewall Manager administrator account that you set up in the prerequisites ([AWS Firewall Manager prerequisites](#)), and then open the Firewall Manager console at [https://console.aws.amazon.com/wafv2/fms](https://console.aws.amazon.com/wafv2/fms).

   **Note**
   For information about setting up a Firewall Manager administrator account, see [Step 2: Set the AWS Firewall Manager administrator account](#).

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 unused 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. By default, Firewall Manager considers security groups as noncompliant with this policy rule if they are unused for any length of time. You can optionally specify a number of minutes that a security group can exist unused before it is considered noncompliant. If you choose this rule, Firewall Manager runs it last when you save the policy.

   - 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 affected by this policy**, if you want to apply the policy to all accounts in your organization, choose **Include all accounts under my organization**. If you want to include only specific accounts in the policy, or alternatively exclude specific accounts from the policy, choose one of those options, and then use the **AWS accounts** dialog box to add account numbers to include or exclude. You can choose only one option.

   **Note**
   For the choices **Include all accounts under my organization** and **Exclude the specified accounts and include all others**, when you add a new account to your organization, Firewall Manager automatically applies the policy to that account.

13. For **Resources**, if you want to apply the policy to all resources within the AWS accounts and resource type parameters, choose **Include all resources that match the selected resource type**. If you want to include or exclude specific resources, use tagging to specify the resources, and then choose the appropriate option and add the tags to the list. You can apply the policy either to all resources except those that have all the tags that you specify, or you can apply it to only those that have all the tags that you specify. For more information about tagging your resources, see *Working with Tag Editor*.

   **Note**
   If you enter more than one tag, a resource must have all the tags to be a match.

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 *Usage audit security group policies* (p. 224).

### 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.

AWS Shield Advanced policy scope changes

If you modify a Firewall Manager-Shield Advanced policy which causes an account or resource to go out of scope of the policy, Firewall Manager no longer monitors the account or resource. However, the account continues to be subscribed to Shield Advanced. The resources continue to be protected by Shield Advanced and will incur the Shield Advanced data transfer charges.

If the tags assigned to a resource change, causing the resource to fall out of scope of a Firewall Manager-Shield Advanced policy, Firewall Manager no longer monitors the resource. However, the resource continues to be protected by Shield Advanced and will incur the Shield Advanced data transfer charges.

If an account that was part of a Firewall Manager-Shield Advanced policy leaves the organization, it no longer falls under the scope of the policy and no longer is monitored by Firewall Manager. However, the account remains 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.

How security group policies work in AWS Firewall Manager

You can 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 (p. 213).

General settings for security group policies

Security group policies in AWS Firewall Manager are similar to other Firewall Manager managed policies. You pick a name and define the scope of the policy. 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. Once in place, Firewall Manager runs your security group policies continuously, and applies them to new AWS accounts and resources as they are added, according to the policy scope.

AWS accounts in scope
The settings that you provide for the AWS accounts affected by the policy determine which of the accounts in your AWS organization to apply the security group policy to. You can choose to apply the policy in one of the following ways:

- To all accounts in your organization
- To all except a specific list of excluded account numbers
- To only a specific list of included account numbers

Whichever option you choose, when you add a new account to your organization, Firewall Manager automatically assesses it against these settings in each security group policy and applies the policy as indicated. For example, if you choose to apply the policy to all accounts except the account numbers in the list, when you add a new account, Firewall Manager applies the policy if the new account number isn't in the exclude list.

**Resources in scope**

The settings that you provide for resources determine which resources in the in-scope accounts and resource types to apply the policy to. You can choose one of the following:

- All resources
- All resources except those that have all the tags that you specify
- Resources that have all the tags that you specify

For more information about tagging your resources, see [Working with Tag Editor](#).

Whichever option you choose, when you add a new resource to your organization, Firewall Manager automatically assesses the resources against these settings in each security group policy and applies the policy as indicated. For example, if you choose to apply the policy only to resources that have all the tags in the list, when you add or update a resource within your policy’s account and resource type parameters, Firewall Manager compares the resource's tags to the list and applies the policy if the resource has all the tags.

**Common security group policies**

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.

**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](#).

**Policy rules settings**

You can choose one or both of the following change control behaviors for the security groups and resources of your common security group policy:
• Identify and revert 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.

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.

Content audit security group policies

Use AWS Firewall Manager content audit security group policies to check and manage the rules that are in use in your organization's security groups. You can apply a content audit security group policy to the same resource types as common security group policies, and you can also apply them to security groups themselves. 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.

Policy scope resource type

For the resource type of a content audit security group policy, you can choose the same types that are available to the common security group policy. You can also choose security groups as a resource type. 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 rules and audit security groups

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 choice in the policy rules settings, to allow or deny use of the 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.

Usage audit security group policies

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 first consolidates redundant security groups, if you've chosen that option, then removes unused security groups, if you've chosen that option.

How Firewall Manager 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 remediates unused security groups

For security groups to be considered unused, they must remain unused by any resource for the minumum number of minutes specified in the policy rule. By default, this number is zero. You can give this a higher setting, in order to allow yourself time to associate new security groups with resources. Firewall Manager remediates unused security groups by deleting them from your account, according to your rules settings.

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.

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 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.

Security group policy limitations

This section lists the limitations for using AWS Firewall Manager policies:

- Updating security groups for Amazon EC2 elastic network interfaces that were created using the Fargate service type is not supported. You can, however, update security groups for Amazon ECS elastic network interfaces with the Amazon EC2 service type.
- Updating Amazon ECS elastic network interfaces 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 elastic network interfaces.
- Firewall Manager doesn't currently support updating security groups in elastic network interfaces for an Elastic Load Balancing load balancer, an Application Load Balancer, or a Network Load Balancer.

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 to for communication with instances in another Amazon VPC. The other instances in the same Amazon VPC are then more secure and isolated.

**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 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.

**Tutorial: Creating a policy with hierarchical rules**

With AWS FireWall Manager, you can create and apply 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.

The following tutorial describes how to create a hierarchical set of protection rules.

**Topics**

- Step 1: Designate a FireWall Manager administrator account (p. 227)
- Step 2: Create a rule group using the FireWall Manager administrator account (p. 227)
- Step 3: Create a FireWall Manager policy and attach the common rule group (p. 227)
- Step 4: Add account-specific rules (p. 227)
- Conclusion (p. 228)
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 master 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 (p. 201). 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 a rule group (p. 211). 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 (p. 213).

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 Web ACL (p. 17).

**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.

**Viewing resource compliance with a policy**

You can check to see what resources an AWS Firewall Manager policy is being applied to.

**To check what resources an AWS Firewall Manager policy is being applied to (console)**

1. Sign in to the AWS Management Console using the 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 Step 2: Set the AWS Firewall Manager administrator account (p. 201).

2. In the navigation pane, choose Security policies.

3. Choose a policy. Firewall Manager lists each account in the organization and shows the status. A Compliant status indicates that the policy has been applied to all applicable resources in the account. A Noncompliant status indicates that the policy is not applied to all resources in the account.

4. Choose an account. Firewall Manager lists each resource in the account and shows the status. A Compliant status indicates that the policy is applied to the resource. A Noncompliant status indicates that the policy is not applied to the resource. Firewall Manager lists up to 100 noncompliant resources.

**AWS Firewall Manager findings**

AWS Firewall Manager creates findings for resources that are out of compliance and for attacks that it detects and sends them to AWS Security Hub. For information about Security Hub findings, see Findings in AWS Security Hub.


**How do I view my Firewall Manager findings?**

To view your Firewall Manager findings in Security Hub, follow the guidance at Working with Findings in Security Hub 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 through the Security Hub console. Choose **Integrations** in the navigation bar, then in the Firewall Manager pane, choose **Disable Integration**. For more information, see the AWS Security Hub User Guide.

**AWS Firewall Manager Finding Types**

- AWS WAF policy findings (p. 229)
- AWS Shield Advanced policy findings (p. 229)
- Security group common policy findings (p. 230)
- Security group content audit policy findings (p. 230)
- Security group usage audit policy findings (p. 231)

**AWS WAF policy findings**

You can use Firewall Manager AWS WAF Classic policies to apply AWS WAF Classic rules to your resources in AWS Organizations. For more information, see Working with AWS Firewall Manager policies (p. 213).

**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.**

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 findings**

You use Firewall Manager Shield policies to protect accounts and resources AWS Shield Advanced. For more information, see Working with AWS Firewall Manager policies (p. 213).
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 findings

For information about security group common policies, see How security group policies work in AWS Firewall Manager (p. 221).

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 findings

For information about security group content audit policies, see How security group policies work in AWS Firewall Manager (p. 221).

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 findings

For information about security group usage audit policies, see How security group policies work in AWS Firewall Manager (p. 221).

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.

Designating a different account as the AWS Firewall Manager administrator account

To use AWS Firewall Manager, you must log in to the console with a Firewall Manager administrator account. You can designate only one account in an organization as a Firewall Manager administrator account. It can be an AWS Organizations master account or a member account. To set up an administrator account for the first time, see Step 2: Set the AWS Firewall Manager administrator account (p. 201).

If you designate an account as an administrator account, and you later want to designate a different account as the administrator account, perform the following procedure.

**Important**

To designate a different account, you first must revoke administrator privileges from the current administrator account. When you revoke the privileges, all Firewall Manager policies created by
that account are deleted. You then must sign into Firewall Manager with the AWS Organizations master account to designate a new administrator account.

To designate a different account as the AWS Firewall Manager administrator account (console)

1. Sign in to the AWS Management Console using the current Firewall Manager administrator account, and then open the Firewall Manager console at https://console.aws.amazon.com/wafv2/fms.
2. In the navigation pane, choose Settings.
3. Choose Revoke administrator account.
   Important
   When you revoke administrator privileges from the current administrator account, all Firewall Manager policies created by that account are deleted.
4. Sign out of the AWS Management console.
5. Sign in to the AWS Management Console using your AWS Organizations master account. You can sign in using your root user credentials for the account (not recommended) or you can sign in using an IAM user or IAM role within the account that has equivalent permissions.
7. Choose Get started.
8. Type an account ID to associate with Firewall Manager. This account will be the new Firewall Manager administrator account. It can be the master account that you are signed in with or it can be a member account in your organization. If the account ID that you type is a member account and not the master account, Firewall Manager sets the appropriate permissions for the member account.
   Note
   The account is given permission to create and manage AWS WAF Classic rules across all accounts within the organization.
9. Choose Set administrator.

Closing the AWS Firewall Manager administrator account

If you close your AWS Firewall Manager Administrator account without first revoking that account (as described in step three, immediately above), then:

- AWS will revoke the account’s administrator access from Firewall Manager. After AWS revokes the account’s administrator access from Firewall Manager, all Firewall Manager policies applied to any account previously governed by the administrator account will be deactivated and such policy protection will no longer be applied to any of these accounts.
- AWS will retain the Firewall Manager policy data for the account for 90 days from the effective date of your Administrator account closure. If you elect to reopen the previously closed account during this 90-day window, AWS will reassign the account as the Firewall Manager administrator and recover the account’s previous Firewall Manager policy data.
- After the 90-day window expires, if the closed account has not been reopened, AWS will permanently delete all Firewall Manager policy data for that account.

Security in AWS Firewall Manager

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 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 (p. 233)
- Identity and access management in AWS Firewall Manager (p. 234)
- Logging and monitoring in Firewall Manager (p. 244)
- Compliance validation for Firewall Manager (p. 245)
- Resilience in Firewall Manager (p. 245)
- Infrastructure security in AWS Firewall Manager (p. 245)

**Data protection in Firewall Manager**

Firewall Manager conforms to the AWS shared responsibility model, which includes regulations and guidelines for data protection. AWS is responsible for protecting the global infrastructure that runs all the AWS services. AWS maintains control over data hosted on this infrastructure, including the security configuration controls for handling customer content and personal data. AWS customers and APN partners, acting either as data controllers or data processors, are responsible for any personal data that they put in the AWS Cloud.

Firewall Manager entities—such as policies—are encrypted at rest. Unique encryption keys are used for each Region.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM), so that 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.
- Set up API and user activity logging with AWS CloudTrail. See AWS CloudTrail API Reference.
- Use AWS encryption solutions, along with all default security controls within AWS services.

We strongly recommend that you never put sensitive identifying information, such as your customers' account numbers, into free-form 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 piece of data that you enter into Firewall Manager or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server.
Identity and access management in AWS Firewall Manager

Access to AWS Firewall Manager requires credentials. Those credentials must have permissions to access AWS Firewall Manager resources, like policies. The following sections provide details on how you can use AWS Identity and Access Management (IAM) and Firewall Manager to help secure access to your resources.

- Authentication (p. 234)
- Access control (p. 235)

Authentication

You can access AWS as any of the following types of identities:

- **AWS account root user** – When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

- **IAM user** – An IAM user is an identity within your AWS account that has specific custom permissions (for example, permissions to create a rule in Firewall Manager). You can use an IAM user name and password to sign in to secure AWS webpages like the AWS Management Console, AWS Discussion Forums, or the AWS Support Center.

In addition to a user name and password, you can also generate access keys for each user. You can use these keys when you access AWS services programmatically, either through one of the several SDKs or by using the AWS Command Line Interface (CLI). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don’t use AWS tools, you must sign the request yourself. Firewall Manager supports Signature Version 4, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 Signing Process in the AWS General Reference.

- **IAM role** – An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user in that it is an AWS identity with permissions policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session. IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as federated users. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated Users and Roles in the IAM User Guide.
• **AWS service access** – A service role is an IAM role that a service assumes to perform actions in your account on your behalf. When you set up some AWS service environments, you must define a role for the service to assume. This service role must include all the permissions that are required for the service to access the AWS resources that it needs. Service roles vary from service to service, but many allow you to choose your permissions as long as you meet the documented requirements for that service. Service roles provide access only within your account and cannot be used to grant access to services in other accounts. You can create, modify, and delete a service role from within IAM. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data from that bucket into an Amazon Redshift cluster. For more information, see Creating a Role to Delegate Permissions to an AWS Service in the IAM User Guide.

• **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances in the IAM User Guide.

**Access control**

You can have valid credentials to authenticate your requests, but unless you have permissions you can’t create or access AWS Firewall Manager resources. For example, you must have permissions to create a Firewall Manager policy.

The following sections describe how to manage permissions for AWS Firewall Manager. We recommend that you read the overview first.

• Overview of managing access permissions to your AWS Firewall Manager resources (p. 236)
• Using identity-based policies (IAM policies) for AWS Firewall Manager (p. 239)
• Firewall Manager required permissions for API actions (p. 241)

**AWS Identity and Access Management**

Firewall Manager integrates with AWS Identity and Access Management (IAM), a service that lets your organization do the following:

• Create users and groups under your organization’s AWS account
• Share your AWS account resources with users in the account
• Assign unique security credentials to each user
• Control user access to services and resources

For example, you can use IAM with Firewall Manager to control which users in your AWS account can create a new policy.

For general information about IAM, see the following documentation:

• AWS Identity and Access Management (IAM)
• IAM Getting Started Guide
• IAM User Guide
Overview of managing access permissions to your AWS Firewall Manager resources

Every AWS resource is owned by an AWS account, and permissions to create or access a resource are governed by permissions policies. An account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles). Some services also support attaching permissions policies to resources.

Note
An account administrator (or administrator user) is a user with administrator privileges. For more information, see IAM Best Practices in the IAM User Guide.

When granting permissions, you decide who is getting the permissions, the resources they get permissions for, and the specific operations that you want to allow on those resources.

Topics

- AWS Firewall Manager resources and operations (p. 236)
- Understanding resource ownership (p. 237)
- Managing access to resources (p. 237)
- Specifying policy elements: Actions, effects, resources, and principals (p. 238)
- Specifying conditions in a policy (p. 239)

AWS Firewall Manager resources and operations

AWS Firewall Manager has the single resource policy. This resource has a unique Amazon Resource Name (ARN) associated with it, as shown in the following table.

<table>
<thead>
<tr>
<th>Name in AWS Firewall Manager Console</th>
<th>Name in AWS Firewall Manager SDK/CLI</th>
<th>ARN Format</th>
</tr>
</thead>
</table>

To allow or deny access to a subset of Firewall Manager resources, include the ARN of the resource in the resource element of your policy. The ARNs for Firewall Manager have the following format:

arn:aws:wafv2:region: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 Firewall Manager resource.
- ID: The ID of the Firewall Manager 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 policies for the account 111122223333 in Region us-east-1:
Identity and access management

\[ \text{arn:aws:wafv2:us-east-1:111122223333:policy/*} \]

For more information, see Resources in the IAM User Guide.

AWS Firewall Manager provides a set of operations to work with Firewall Manager resources. For a list of available operations, see Actions.

Understanding resource ownership

A resource owner is the AWS account that creates the resource. That is, the resource owner is the AWS account of the principal entity (the root account, an IAM user, or an IAM role) that authenticates the request that creates the resource. The following examples illustrate how this works:

- If you use the root account credentials of your AWS account to create a Firewall Manager resource, your AWS account is the owner of the resource.
- If you create an IAM user in your AWS account and grant permissions to create a Firewall Manager resource to that user, the user can create a Firewall Manager resource. However, your AWS account, to which the user belongs, owns the Firewall Manager resource.
- If you create an IAM role in your AWS account with permissions to create a Firewall Manager resource, anyone who can assume the role can create a Firewall Manager resource. Your AWS account, to which the role belongs, owns the Firewall Manager resource.

Managing access to resources

A permissions policy describes who has access to what. The following sections explain the available options for creating permissions policies.

Note
These sections discuss using IAM in the context of AWS Firewall Manager. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see What Is IAM? in the IAM User Guide. For information about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the IAM User Guide.

Policies that are attached to an IAM identity are known as identity-based policies, and policies that are attached to a resource are known as resource-based policies. AWS Firewall Manager supports only identity-based policies.

Topics

- Identity-based policies (IAM policies) (p. 237)
- Resource-based policies (p. 238)

Identity-based policies (IAM policies)

You can attach policies to IAM identities. For example, you can do the following:

- **Attach a permissions policy to a user or a group in your account** – An account administrator can use a permissions policy that is associated with a particular user to grant permissions for that user to create a Firewall Manager resource.
- **Attach a permissions policy to a role (grant cross-account permissions)** – You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions. For example, the administrator in Account A can create a role to grant cross-account permissions to another AWS account (for example, Account B) or an AWS service as follows:
  1. Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions on resources in Account A.
2. Account A administrator attaches a trust policy to the role identifying Account B as the principal who can assume the role.

3. Account B administrator can then delegate permissions to assume the role to any users in Account B. Doing this allows users in Account B to create or access resources in Account A. The principal in the trust policy also can be an AWS service principal if you want to grant an AWS service permissions to assume the role.

For more information about using IAM to delegate permissions, see Access Management in the IAM User Guide.

The following is an example policy that grants permissions for the fms:GetPolicy action on all policies in two specific regions.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Deny",
      "Action": "fms:GetPolicy",
      "Resource": [
        "arn:aws:fms:us-east-1::*:policy/*",
        "arn:aws:fms:us-west-2::*:policy/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/stage": "prod"
        }
      }
    }
  ]
}
```

For more information about using identity-based policies with Firewall Manager, see Using identity-based policies (IAM policies) for AWS Firewall Manager (p. 239). For more information about users, groups, roles, and permissions, see Identities (Users, Groups, and Roles) in the IAM User Guide.

Resource-based policies

Other services, such as Amazon S3, also support resource-based permissions policies. For example, you can attach a policy to an S3 bucket to manage access permissions to that bucket. AWS Firewall Manager doesn't support resource-based policies.

Specifying policy elements: Actions, effects, resources, and principals

For each AWS Firewall Manager resource (see AWS Firewall Manager resources and operations (p. 236)), the service defines a set of API operations (see Firewall Manager required permissions for API actions (p. 241)). To grant permissions for these API operations, Firewall Manager defines a set of actions that you can specify in a policy. Note that performing an API operation can require permissions for more than one action. When granting permissions for specific actions, you also identify the resource on which the actions are allowed or denied.

The following are the most basic policy elements:

- **Resource** – In a policy, you use an Amazon Resource Name (ARN) to identify the resource to which the policy applies. For more information, see AWS Firewall Manager resources and operations (p. 236).
- **Action** – You use action keywords to identify resource operations that you want to allow or deny. For example, the fms:CreatePolicy permission, coupled with the waf:ListRuleGroups permission, allows the user permissions to perform the AWS Firewall Manager CreatePolicy operation.
• **Effect** – You specify the effect when the user requests the specific action. This can be either allow or deny. If you don't explicitly grant access to a resource, access is implicitly denied. You also can explicitly deny access to a resource, which you might do to make sure that a user cannot access it, even if a different policy grants access.

• **Principal** – In identity-based policies (IAM policies), the user that the policy is attached to is the implicit principal. AWS Firewall Manager doesn't support resource-based policies.

To learn more about IAM policy syntax and descriptions, see [AWS IAM Policy Reference](https://docs.aws.amazon.com/IAM/latest/userguide/reference_policies_elements.html) in the **IAM User Guide**.

For a table that shows all the AWS Firewall Manager API actions and the resources that they apply to, see [Firewall Manager required permissions for API actions](https://docs.aws.amazon.com/waf/latest/APIReference/).  

**Specifying conditions in a policy**

When you grant permissions, you can use the IAM policy language to specify the conditions when a policy should take effect. For example, you might want a policy to be applied only after a specific date. For more information about specifying conditions in a policy language, see [Condition](https://docs.aws.amazon.com/IAM/latest/userguide/reference_policies_elements.html) in the **IAM User Guide**.

To express conditions, you use predefined condition keys. There are no condition keys specific to Firewall Manager. However, there are AWS-wide condition keys that you can use as appropriate. For a complete list of AWS-wide keys, see [Available Keys for Conditions](https://docs.aws.amazon.com/IAM/latest/userguide/reference_policies_elements.html) in the **IAM User Guide**.

**Using identity-based policies (IAM policies) for AWS Firewall Manager**

This section provides examples of identity-based policies that demonstrate how an account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles) and thereby grant permissions to perform operations on AWS Firewall Manager resources.

**Important**

We recommend that you first review the introductory topics that explain the basic concepts and options available for you to manage access to your AWS Firewall Manager resources. For more information, see [Overview of managing access permissions to your AWS Firewall Manager resources](https://docs.aws.amazon.com/waf/latest/userguide/). For a table that shows all the AWS Firewall Manager API actions and the resources that they apply to, see [Firewall Manager required permissions for API actions](https://docs.aws.amazon.com/waf/latest/APIReference/).

**Topics**

- Permissions required to use the AWS Firewall Manager console (p. 239)
- AWS managed (predefined) policies for AWS Firewall Manager (p. 240)
- Customer managed policy examples (p. 240)

**Permissions required to use the AWS Firewall Manager console**

The AWS Firewall Manager console provides an integrated environment for you to create and manage Firewall Manager resources. The console provides many features and workflows that often require permissions to create a Firewall Manager resource in addition to the API-specific permissions that are documented in the [Firewall Manager required permissions for API actions](https://docs.aws.amazon.com/waf/latest/APIReference/). For more information about these additional console permissions, see [Customer managed policy examples](https://docs.aws.amazon.com/waf/latest/APIReference/).
AWS managed (predefined) policies for AWS Firewall Manager

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. Managed policies grant necessary permissions for common use cases so you can avoid having to investigate what permissions are needed. For more information, see AWS Managed Policies in the IAM User Guide.

The following AWS managed policies, which you can attach to users in your account, are specific to AWS Firewall Manager and are grouped by use case scenario:

- **AWSFMAdminFullAccess** – Grants full access to AWS Firewall Manager resources.
- **AWSFMAdminReadOnlyAccess** – Grants read-only access to all AWS Firewall Manager resources.
- **AWSFMMemberReadOnlyAccess** – Grants read-only access to AWS Firewall Manager member resources.

**Note**
You can review these permissions policies by signing in to the IAM console and searching for specific policies there.

You also can create your own custom IAM policies to allow permissions for AWS Firewall Manager API operations and resources. You can attach these custom policies to the IAM users or groups that require those permissions or to custom execution roles (IAM roles) that you create for your Firewall Manager resources.

Customer managed policy examples

The examples in this section provide a group of sample policies that you can attach to a user. If you are new to creating policies, we recommend that you first create an IAM user in your account and attach the policies to the user, in the sequence outlined in the steps in this section.

You can use the console to verify the effects of each policy as you attach the policy to the user. Initially, the user doesn't have permissions, and the user won't be able to do anything on the console. As you attach policies to the user, you can verify that the user can perform various operations on the console.

We recommend that you use two browser windows: one to create the user and grant permissions, and the other to sign in to the AWS Management Console using the user's credentials and verify permissions as you grant them to the user.

For examples that show how to create an IAM role that you can use as an execution role for your Firewall Manager resource, see Creating IAM Roles in the IAM User Guide.

Example topics

- **Example: Give admin user read-only access to Firewall Manager security groups (p. 240)**

Create an IAM user

First, you need to create an IAM user, add the user to an IAM group with administrative permissions, and then grant administrative permissions to the IAM user that you created. You then can access AWS using a special URL and the user's credentials.

For instructions, see Creating Your First IAM User and Administrators Group in the IAM User Guide.

**Example: Give admin user read-only access to Firewall Manager security groups**

The following policy grants admin users read-only access to Firewall Manager security groups and policies. These users can't create, update, or delete the Firewall Manager resources.
Firewall Manager required permissions for API actions

When you set up Access control (p. 235) and writing permissions policies that you can attach to an IAM identity (identity-based policies), use the information in this section as a guide. For each AWS Firewall Manager API operation, you need to know the actions for which to grant permissions, and the AWS resource for which you grant the permissions. You specify the actions in the policy's Action field, and you specify the resource value in the policy's Resource field.

### Note
To specify an action, use the `fms:` prefix followed by the API operation name (for example, `fms:CreatePolicy`). This topic only lists actions that require explicit resource permissions.

You can use AWS-wide condition keys in your AWS Firewall Manager policies to express conditions. For a complete list of AWS-wide keys, see Available Keys for Conditions in the IAM User Guide.

To use the following Firewall Manager API actions, you need permissions on the resource: `arn:aws:fms:region:account:policy/ID`.

- DeletePolicy
- GetComplianceDetail
- GetPolicy
- GetProtectionStatus
- ListComplianceStatus
- PutPolicy

For more information about Firewall Manager actions and resources, see the AWS Identity and Access Management guide topic Actions Defined by AWS Firewall Manager

For the full list of the API actions available for Firewall Manager, see AWS Firewall Manager API Reference.

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

Firewall Manager uses the service-linked role FMSServiceRolePolicy.

AWS Firewall Manager uses this service-linked role to write logs to Amazon Kinesis Data Firehose. This role is used only if you enable logging in AWS Firewall Manager. For more information, see Logging Web ACL traffic information (p. 50).

The FMSServiceRolePolicy service-linked role trusts the service to assume the role fms.amazonaws.com.

The permissions policies of the role allows Firewall Manager to complete the following actions on the specified resources:

- Action: firehose:PutRecord and firehose:PutRecordBatch on Amazon Kinesis Data Firehose data stream resources with a name that starts with "aws-fms-logs-." For example, aws-fms-logs-us-east-2-analytics.

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 FMSServiceRolePolicy 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.
### To delete the service-linked role using IAM

Use the IAM console, the IAM CLI, or the IAM API to delete the FMSServiceRolePolicy 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 the following AWS Regions.

<table>
<thead>
<tr>
<th>Region Name</th>
<th>Region Identity</th>
<th>Support in Firewall Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>US East (N. Virginia)</td>
<td>us-east-1</td>
<td>Yes</td>
</tr>
<tr>
<td>US East (Ohio)</td>
<td>us-east-2</td>
<td>Yes</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>us-west-1</td>
<td>Yes</td>
</tr>
<tr>
<td>US West (Oregon)</td>
<td>us-west-2</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>ap-south-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Osaka-Local)</td>
<td>ap-northeast-3</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>ap-northeast-2</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>ap-southeast-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Sydney)</td>
<td>ap-southeast-2</td>
<td>Yes</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>ap-northeast-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Canada (Central)</td>
<td>ca-central-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>eu-central-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>eu-west-1</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (London)</td>
<td>eu-west-2</td>
<td>Yes</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>eu-west-3</td>
<td>Yes</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>sa-east-1</td>
<td>Yes</td>
</tr>
</tbody>
</table>
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 incidents:

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 (p. 282).

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 (p. 286).

AWS Trusted Advisor

Trusted Advisor draws upon best practices learned from serving hundreds of thousands of AWS customers. Trusted Advisor inspects your AWS environment and then makes recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps. All AWS customers have access to five Trusted Advisor checks. Customers with a Business or Enterprise support plan can view all Trusted Advisor checks. For more information, see AWS Trusted Advisor.
Compliance validation for Firewall Manager

Third-party auditors assess the security and compliance of Firewall Manager as part of multiple AWS compliance programs, such as SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. 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 Firewall Manager is determined by the sensitivity of your data, your organization’s compliance objectives, and applicable laws and regulations. If your use of Firewall Manager is subject to compliance with standards like HIPAA, PCI, or FedRAMP, AWS provides resources to help:

- **Security and Compliance Quick Start Guides** – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- **Architecting for HIPAA Security and Compliance Whitepaper** – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- **AWS Compliance Resources** – This collection of workbooks and guides might apply to your industry and location.
- **AWS Security Hub** – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.
- **AWS Well-Architected Framework** – The AWS Well-Architected Framework helps you build secure cloud applications.

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 the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access Firewall Manager through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.
AWS Firewall Manager quotas

AWS Firewall Manager is subject to the following quotas (formerly referred to as limits).

AWS Firewall Manager has default quotas on the number of entities per account. You can request an increase in these quotas.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounts per organization in AWS Organizations</td>
<td>Varies. An invitation sent to an account counts against this quota. The count is returned if the invited account declines, the master account cancels the invitation, or the invitation expires.</td>
</tr>
<tr>
<td>Firewall Manager policies per organization in AWS Organizations</td>
<td>20</td>
</tr>
<tr>
<td>Tags that include or exclude resources per Firewall Manager policy</td>
<td>8</td>
</tr>
<tr>
<td>Rule groups per Firewall Manager administrator account</td>
<td>10</td>
</tr>
<tr>
<td>Primary security groups per common Firewall Manager policy</td>
<td>1</td>
</tr>
<tr>
<td>Audit security groups per content audit Firewall Manager policy</td>
<td>1</td>
</tr>
<tr>
<td>Amazon VPC instances in scope per Firewall Manager common security group policy</td>
<td>5</td>
</tr>
</tbody>
</table>

The security group policies managed by Firewall Manager are subject to standard Amazon VPC quotas. For more information, see Amazon VPC Quotas in the Amazon VPC User Guide.

The following quotas related to AWS Firewall Manager can't be changed.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Quota</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule groups per Firewall Manager policy</td>
<td>2: 1 customer-created rule group and 1 AWS Marketplace rule group</td>
</tr>
<tr>
<td>Rules per rule group</td>
<td>10</td>
</tr>
</tbody>
</table>
AWS Shield

AWS provides AWS Shield Standard and AWS Shield Advanced for protection against DDoS attacks. AWS Shield Standard is automatically included at no extra cost beyond what you already pay for AWS WAF and your other AWS services. For added protection against DDoS attacks, AWS offers AWS Shield Advanced. AWS Shield Advanced provides expanded DDoS attack protection for your Amazon Elastic Compute Cloud instances, Elastic Load Balancing load balancers, Amazon CloudFront distributions, Amazon Route 53 hosted zones, and your AWS Global Accelerator accelerators.

How AWS Shield works

A distributed denial of service (DDoS) attack is an attack in which multiple compromised systems attempt to flood a target, such as a network or web application, with traffic. A DDoS attack can prevent legitimate users from accessing a service and can cause the system to crash due to the overwhelming traffic volume.

AWS provides two levels of protection against DDoS attacks: AWS Shield Standard and AWS Shield Advanced.

AWS Shield Standard

All AWS customers benefit from the automatic protections of AWS Shield Standard, at no additional charge. AWS Shield Standard defends against most common, frequently occurring network and transport layer DDoS attacks that target your website or applications. While AWS Shield Standard helps protect all AWS customers, you get particular benefit if you are using Amazon CloudFront and Amazon Route 53. These services receive comprehensive availability protection against all known infrastructure (Layer 3 and 4) attacks.

AWS Shield Advanced

For higher levels of protection against attacks, you can subscribe to AWS Shield Advanced. AWS Shield Advanced provides expanded DDoS attack protection for web applications running on the following resources:

- Amazon Elastic Compute Cloud
- Elastic Load Balancing (ELB)
Amazon CloudFront
• Amazon Route 53
• AWS Global Accelerator

For example, if you use Shield Advanced to protect an Elastic IP address, Shield Advanced automatically deploys your network ACLs to the border of the AWS network during an attack. 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.

The point at which Shield Advanced detects attacks and places mitigations depends on the architecture you use for your web applications. It varies based on characteristics like the type of instance you use, your instance size, and whether you use enhanced networking.

As an AWS Shield Advanced customer, you can contact a 24x7 DDoS response team (DRT) for assistance during a DDoS attack. You also have exclusive access to advanced, real-time metrics and reports for extensive visibility into attacks on your AWS resources. With the assistance of the DRT, AWS Shield Advanced includes intelligent DDoS attack detection and mitigation for not only for network layer (layer 3) and transport layer (layer 4) attacks, but also for application layer (layer 7) attacks.

To use the services of the DRT, you must be subscribed to the Business Support plan or the Enterprise Support plan.

AWS Shield Advanced also offers some cost protection against spikes in your AWS bill that could result from a DDoS attack. This cost protection is provided for your Elastic Load Balancing load balancers, Amazon CloudFront distributions, Amazon Route 53 hosted zones, Amazon Elastic Compute Cloud instances, and your AWS Global Accelerator accelerators.

AWS WAF is included with AWS Shield Advanced at no extra cost. For more information about AWS Shield Advanced pricing, see AWS Shield Advanced Pricing.

When you add an AWS Shield Advanced protection to a resource, you can optionally include one or more additions to the protection. The protection additions vary by resource type and can include the following:

• A custom AWS WAF web ACL or rate-based rule, as described in Step 3: Add rate-based rules (p. 256).
• An Amazon CloudWatch alarm, as described in Step 5: Configure Amazon CloudWatch alarms (p. 258).
• An Amazon Route 53 health check for health-based detection, as described in the following section.

Shield Advanced health-based detection

Shield Advanced health-based detection uses the health of your AWS resource to improve responsiveness and accuracy in attack detection and mitigation. To use health-based detection, you define the health check in Route 53 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 and Updating Health Checks.

You can enable health-based detection for any resource type that Shield Advanced supports. The benefits of health-based detection vary according to the type of resource. The following sections describe some of these benefits.
Benefits for Elastic IP addresses and Global Accelerator accelerators

Health-based detection improves the accuracy of network-layer and transport-layer event detection and mitigation.

When you protect an Elastic IP address or Global Accelerator accelerator with Shield Advanced, you reduce the threshold required to place a mitigation. Shield Advanced helps to provide quicker mitigation for attacks and mitigations for smaller attacks, even when traffic is within the application's capacity.

When you add health-based detection, during periods when the associated Route 53 health check is unhealthy, Shield Advanced can place mitigations even more quickly and at lower thresholds.

Benefits for CloudFront distributions and Application Load Balancers

Health-based detection improves the accuracy of web request flood detection.

When you protect a CloudFront distribution or Application Load Balancer with Shield Advanced, you receive web request flood detection alerts when there is a statistically significant deviation in traffic volume combined with significant changes in traffic self-similarity. Self-similarity is determined based on attributes like user agent, referrer, and URI.

When you add health-based detection, you increase the likelihood that the alerts you receive are timely and actionable. With health-based detection, during periods when the associated Route 53 health check is unhealthy, Shield Advanced requires smaller deviations to alert and it reports events more quickly. When the associated Route 53 health check is healthy, Shield Advanced requires larger deviations to alert.

Types of DDoS attacks

AWS Shield Advanced provides expanded protection against many types of attacks. For example:

User Datagram Protocol (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 users from accessing needed resources.

SYN flood

The intent of an 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 SYN packet. The server returns an acknowledgment, and the client returns its own acknowledgement, completing the three-way handshake. In an 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 POST 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.
About the AWS DDoS response team (DRT)

With AWS Shield Advanced, complex DDoS events can be escalated to the AWS DDoS Response team (DRT), which has deep experience in protecting AWS, Amazon.com, and its subsidiaries.

For layer 3 and layer 4 attacks, AWS provides automatic attack detection and proactively applies mitigations on your behalf. For layer 7 DDoS attacks, AWS attempts to detect and notify AWS Shield Advanced customers through CloudWatch alarms, but does not apply mitigations proactively. This is to avoid inadvertently dropping valid user traffic.

You can also contact the DRT before or during a possible attack to develop and deploy custom mitigations. For example, if you are running a web application and only need ports 80 and 443 open, you can work with the DRT to pre-configure an ACL to only "Allow" ports 80 and 443.

AWS Shield Advanced customers have two options to mitigate layer 7 attacks:

- **Provide your own mitigations:** AWS WAF is included with AWS Shield Advanced at no extra cost. You can create your own AWS WAF rules to mitigate the DDoS attacks. AWS provides preconfigured templates to get you started quickly. The templates include a set of AWS WAF rules that are designed to block common web-based attacks. You can customize the templates to fit your business needs. For more information, see AWS WAF Security Automations.

  In this case, the DRT is not involved. You can, however, engage the DRT for guidance on implementing best practices such as AWS WAF common protections.

- **Engage the DRT:** If you want additional support in addressing an attack, you can contact the AWS Support Center. Critical and urgent cases are routed directly to DDoS experts. With AWS Shield Advanced, complex cases can be escalated to the DRT, which has deep experience in protecting AWS, Amazon.com, and its subsidiaries. If you are an AWS Shield Advanced customer, you also can request special handling instructions for high severity cases.

  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.

  The DRT helps you triage the DDoS attack to identify attack signatures and patterns. With your consent, the DRT creates and deploys AWS WAF rules to mitigate the attack.

When AWS Shield Advanced detects a large layer 7 attack against one of your applications, the DRT might proactively contact you. The DRT triages the DDoS incident and creates AWS WAF mitigations. The DRT then contacts you for consent to apply the AWS WAF rules.

**Important**

The DRT can help you to analyze suspicious activity and assist you to mitigate the issue. This mitigation often requires the DRT to create or update web access control lists (web ACLs) in your account. However, they need your permission to do so. We recommend that as part of enabling AWS Shield Advanced, you follow the steps in Step 4: (Optional) authorize the DDoS response team (p. 257) to proactively provide the DRT with the needed permissions. Providing permission ahead of time helps prevent any delays in the event of an actual attack.

To use the services of the DRT, you must be subscribed to the Business Support plan or the Enterprise Support plan.

Help me choose a protection plan

In many cases, AWS Shield Standard protection is sufficient for your needs. AWS services and technologies are built to provide resilience in the face of the most common DDoS attacks. Supplementing this built-in protection with AWS WAF and a combination of other AWS services as a defense-in-depth strategy typically provides adequate attack protection and mitigation. Further, if you
have the technical expertise and want full control over monitoring for and mitigating layer 7 attacks, AWS Shield Standard is likely the appropriate choice.

If your business or industry is a likely target of DDoS attacks, or if you prefer to let AWS handle the majority of DDoS protection and mitigation responsibilities for layer 3, layer 4, and layer 7 attacks, AWS Shield Advanced might be the best choice. AWS Shield Advanced not only provides layer 3 and layer 4 protection and mitigation, but also includes AWS WAF at no extra charge and DRT assistance for layer 7 attacks. If you use AWS WAF and AWS Shield Standard, you must design your own layer 7 protection and mitigation processes.

AWS Shield Advanced customers also benefit from detailed information about DDoS attacks against their AWS resources. While AWS Shield Standard provides automatic protection for the most common layer 3 and layer 4 attacks, visibility into the details of those attacks is limited. AWS Shield Advanced provides you with extensive data about the details of both layer 3, layer 4, and layer 7 DDoS attacks.

AWS Shield Advanced also offers cost protection for DDoS attacks against your AWS resources. This valuable feature helps prevent unexpected spikes in your bill caused by DDoS attacks. If cost predictability is important to you, AWS Shield Advanced can offer that stability.

The following table shows a comparison of AWS Shield Standard and AWS Shield Advanced.

<table>
<thead>
<tr>
<th>Feature</th>
<th>AWS Shield Standard</th>
<th>AWS Shield Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Active Monitoring</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network flow monitoring</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Automatic always-on detection</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Automated application (layer 7) traffic monitoring</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>DDoS Mitigations</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Helps protect against common DDoS attacks, such as SYN flood and UDP reflection attacks</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Access to additional DDoS mitigation capacity, including automatic deployment of network ACLs to the AWS border during an attack</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Custom application</td>
<td>Yes, through user-created AWS WAF ACLs. Incurs</td>
<td>Yes, through user-created or DRT-created AWS WAF ACLs. Included as part of the AWS Shield Advanced subscription.</td>
</tr>
</tbody>
</table>
## Help me choose a protection plan

<table>
<thead>
<tr>
<th>Feature</th>
<th>AWS Shield Standard</th>
<th>AWS Shield Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>layer (layer 7) mitigations</td>
<td>standard AWS WAF charges.</td>
<td>Yes</td>
</tr>
<tr>
<td>Instant rule updates</td>
<td>Yes, through user-created AWS WAF ACLs. Incurs standard AWS WAF charges.</td>
<td>Yes</td>
</tr>
<tr>
<td>AWS WAF for app vulnerability protection</td>
<td>Yes, through user-created AWS WAF ACLs. Incurs standard AWS WAF charges.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Visibility and Reporting

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 3/4 attack notification</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Layer 3/4 attack forensics reports (source IP, attack vector, and more)</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Layer 7 attack notification</td>
<td>Yes, through AWS WAF. Incurs standard AWS WAF charges.</td>
<td>Yes</td>
</tr>
<tr>
<td>Layer 7 attack forensics reports (Top talkers report, sampled requests, and more)</td>
<td>Yes, through AWS WAF. Incurs standard AWS WAF charges.</td>
<td>Yes</td>
</tr>
<tr>
<td>Layer 3/4/7 attack historical report</td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>

**DDoS Response Team Support (Must be subscribed to the Business Support plan or the Enterprise Support plan.**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Incident management during high severity events</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Custom mitigations during attacks</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Post-attack analysis</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

**Cost Protection** (Service credits for DDoS scaling charges)
### Example AWS Shield Advanced use cases

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.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Suggested services</th>
<th>Related service documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protect a web application and RESTful APIs against a DDoS attack</td>
<td>Shield Advanced protecting an Amazon CloudFront distribution and an Application Load Balancer</td>
<td>Amazon Elastic Load Balancing Documentation, Amazon CloudFront Documentation</td>
</tr>
<tr>
<td>Protect a TCP-based application against a DDoS attack</td>
<td>Shield Advanced protecting a Network Load Balancer attached to an Elastic IP address</td>
<td>Amazon Elastic Load Balancing Documentation</td>
</tr>
<tr>
<td>Protect a UDP-based game server against a DDoS attack</td>
<td>Shield Advanced protecting an Amazon EC2 instance attached to an Elastic IP address</td>
<td>Amazon Elastic Compute Cloud Documentation</td>
</tr>
</tbody>
</table>

### AWS Shield pricing

AWS Shield Standard is included with your AWS services at no additional cost.

AWS Shield Advanced pricing is detailed on the [AWS Shield Advanced Pricing page](https://aws.amazon.com/waf/pricing/). AWS Shield Advanced does have an additional cost, but AWS Shield Advanced customers do not pay for AWS WAF separately for resources that they protect with AWS Shield Advanced. Protection for those resources is included as part of the AWS Shield Advanced service. Further, AWS Shield Advanced charges do not increase with attack volume. This provides a predictable cost for the extended protection.
The AWS Shield Advanced fee applies for each business that is subscribed to AWS Shield Advanced. If your business has multiple AWS accounts, you pay just one Shield Advanced monthly fee as long as all the AWS accounts are in the same Consolidated Billing account family. Further, you must own all the AWS accounts and resources in the account.

However, AWS Channel Resellers will pay a separate monthly fee for each member account. AWS Channel Resellers who resell AWS Shield Advanced to customers with more than one member account may contact us for additional billing support. With respect to such AWS Channel Resellers, AWS reserves the right to modify the monthly fee for AWS Shield Advanced. For more information, see the AWS Shield Advanced Pricing page.

**Getting started with AWS Shield Advanced**

This tutorial shows you how to get started with AWS Shield Advanced. For best results, perform the following steps in sequence.

**Topics**
- Step 1: Activate AWS Shield Advanced (p. 254)
- Step 2: Specify your resources to protect (p. 255)
- Step 3: Add rate-based rules (p. 256)
- Step 4: (Optional) authorize the DDoS response team (p. 257)
- Step 5: Configure Amazon CloudWatch alarms (p. 258)
- Step 6: Deploy AWS WAF rules (p. 258)
- Step 7: Monitor the global threat environment dashboard (p. 259)

**Step 1: Activate AWS Shield Advanced**

AWS Shield Advanced provides advanced DDoS detection and mitigation protection for network layer (layer 3), transport layer (layer 4), and application layer (layer 7) attacks.

**Important**
You must activate Shield Advanced for each AWS account that you want to protect. If you want to activate Shield Advanced for multiple accounts, we recommend that you use AWS Firewall Manager. For more information, see Getting started with AWS Firewall Manager to enable AWS Shield Advanced protection (p. 205).

**To activate AWS Shield Advanced**

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.

2. If this is your first time signing in to the AWS WAF console, choose Go to Shield, and then choose Activate AWS Shield Advanced. Read each term of the agreement, and then select each check box to indicate that you accept the terms. Before you can continue, you must select all check boxes, then choose Activate service.

**Important**
By choosing Activate service, you are subscribing to Shield Advanced and activating the service. To unsubscribe, contact AWS Support.

Otherwise, in the navigation pane, under AWS Shield, choose Protected resources.

You can now go to Step 2: Specify your resources to protect (p. 255).
Activating and setting up AWS Shield Advanced for multiple accounts

You must activate Shield Advanced for each AWS account that you want to protect. To activate Shield Advanced for multiple accounts, we recommend that you use AWS Firewall Manager. For more information, see Getting started with AWS Firewall Manager to enable AWS Shield Advanced protection (p. 205). Alternatively, you can follow the procedure in Step 1: Activate AWS Shield Advanced (p. 254) for each account, each time logging in with a different account.

If you activate Shield Advanced for multiple accounts that are in the same consolidated billing account family, the monthly subscription fee covers all those accounts. You don't pay extra subscription fees for individual accounts. You must own all the AWS accounts and resources in the account.

Note

AWS Channel Resellers pay a separate monthly fee for each member account. AWS Channel Resellers who resell AWS Shield Advanced to customers with more than one member account can contact us for additional billing support. With respect to such AWS Channel Resellers, AWS reserves the right to modify the monthly fee for AWS Shield Advanced. For more information, see the AWS Shield Advanced Pricing page.

The first time that you activate Shield Advanced from an account, you are presented with a pricing agreement. The pricing agreement is displayed on the console each time that you activate Shield Advanced from a different account. The pricing agreement covers all activated accounts in a consolidated billing family, but you must agree to the terms each time that you activate an account.

You can now go to Step 2: Specify your resources to protect (p. 255).

Step 2: Specify your resources to protect

After you activate your AWS Shield Advanced subscription, as described in Step 1: Activate AWS Shield Advanced (p. 254), you specify the resources that you want to protect.

If you are using AWS Firewall Manager to create a Firewall Manager-Shield Advanced policy, you don't need to do this step. You have already specified your resources in the policy.

If you aren't using a Firewall Manager-Shield Advanced policy, you can also specify resources later if you want, using the procedure at Adding AWS Shield Advanced protection to AWS resources (p. 259).

Note

Shield Advanced only protects resources that you have specified in Shield Advanced or through a Firewall Manager-Shield Advanced policy. It doesn't automatically protect your resources.

To choose the resources to protect with Shield Advanced

1. Choose the resources to protect. For load balancers or Elastic IP addresses, you also must choose a Region.

You can choose from the dropdown list, or you can enter the Amazon Resource Name (ARN) of specific resources. You can choose or enter any combination of resource types and resources. If you enter an ARN, the ARN must be in the account that you're using.

Shield Advanced lists a maximum of 100 resources at one time. If you have more than 100 resources, choose Next to see the next set.

Note

• If you want to protect an Amazon EC2 instance, you first must associate an Elastic IP address to the instance, and then choose the Elastic IP address as the resource to protect.
• If you choose an Elastic IP address as the resource to protect, Shield Advanced protects whatever resource is associated with that Elastic IP address, either an Amazon EC2
instance or an Elastic Load Balancing load balancer. Shield Advanced automatically identifies the type of resource that is associated with the Elastic IP address and applies the appropriate mitigations for that resource. This includes configuring network ACLs that are specific to that Elastic IP address. For more information about using Elastic IP addresses with your AWS resources, see the appropriate guide: Amazon Elastic Compute Cloud Documentation or Elastic Load Balancing Documentation.

- Shield Advanced does not support EC2-Classic.

2. Choose **Protect selected resources**.

You can now go to Step 3: Add rate-based rules (p. 256).

### Step 3: Add rate-based rules

We recommend that you add rate-based rules as part of your AWS Shield Advanced protections. These rules can alert you to sudden spikes in traffic that might indicate a potential DDoS event. A rate-based rule counts the requests that arrive from any individual address in any five minute period. If the number of requests exceeds the limit defined by you, the rule can trigger an action such as sending you a notification. For more information about rate-base rules, see How AWS WAF works (p. 6).

**Note**

If you have used AWS Firewall Manager to create a Firewall Manager-Shield Advanced policy, do not do this step. Firewall Manager doesn't support rate-based rules.

#### To add rate-based rules

1. For each resource that is listed in the table, choose an existing web ACL or alternatively create a different web ACL. To create a web ACL, follow these steps:
   a. From the dropdown list, choose **Create new web ACL**.
   b. Enter a name. You can't change the name after you create the web ACL.
   c. Choose **Create web ACL**.

   **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 a Web ACL with an AWS resource (p. 16).

   The table shows the resources that you can associate with a web ACL. These are Amazon CloudFront distributions, Amazon API Gateway APIs, and Application Load Balancers. You can't associate web ACLs and rate-based rules with other types of resources.

2. For each resource that is listed in the table, in the **Rate-based rule** dropdown, choose the rule setting that you want to apply. If you want to apply a rate-based rule, you can choose an existing rule or create a different rate-based rule with the following steps:
   a. From the dropdown list, choose **Create new rule**.
   b. Enter a name.
   c. Enter a rate limit. This is the maximum number of requests allowed in a five-minute period from any single IP address.
   d. Choose **Create rule**.
   e. Optionally, on the resource line, use the **Action** dropdown list to modify the action to take if the rule is triggered. **Block** will block the request. **Count** will allow the request but increment a counter tracking how many times this rule was triggered.

3. Choose **Apply web ACLs and rules**.
To continue without adding rate-based rules

- Choose Continue.

  Important
  We recommend that at a minimum you attach a web ACL to each resource, even if that web ACL doesn't contain any rules.

You can now go to Step 4: (Optional) authorize the DDoS response team (p. 257).

Step 4: (Optional) authorize the DDoS response team

One of the benefits of AWS Shield Advanced is support from the DDoS response team (DRT). When you experience a potential DDoS attack, you can contact the AWS Support Center. If necessary, the Support Center escalates your issue to the DRT. The DRT 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 DRT 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 DRT with the needed authorization. Providing authorization ahead of time helps prevent mitigation delays in the event of an actual attack.

If you do not want to authorize the DRT to mitigate potential attacks on your behalf, choose Do not grant the DRT access to my account, and then choose Save settings. Otherwise, continue with the following steps.

  Note
  To use the services of the DRT, you must be subscribed to the Business Support plan or the Enterprise Support plan.

To authorize the DRT to mitigate potential attacks on your behalf

1. Select either Create new role for the DRT to access my account or Choose an existing role for the DRT to access my account.

   If you choose to use an existing role, you must attach the AWSShieldDRTAccessPolicy managed policy to the role. For more information, see Attaching and Detaching IAM Policies. If you choose Create new role for the DRT to access my account, the policy is attached to the role automatically.

   If you choose to use an existing role, the role must also trust the service principal drt.shield.amazonaws.com. For more information, see IAM JSON Policy Elements: Principal.

   The AWSShieldDRTAccessPolicy managed policy gives the DRT full access only to your AWS WAF and Shield resources. The policy enables the DRT to inspect your AWS WAF configuration and create or update AWS WAF rules and web ACLs on your behalf. The DRT takes these actions only if explicitly authorized by you.

2. Complete the necessary information, either the new role name or the existing role name.

3. (Optional) If you want to authorize the DRT to access your AWS WAF logs stored in an Amazon S3 bucket, enter the name of your bucket where those logs are stored. Choose Add bucket. Repeat as necessary, adding more buckets, up to a maximum of 10. For more information, see Logging Web ACL traffic information (p. 50).

   When you add a bucket to the list, the following permissions are granted to the DRT: s3:GetBucketLocation, s3:GetObject, and s3:ListBucket.

4. We send notifications of possible DDoS activity to the email address that is associated with your account. If you want us to send notifications to additional email addresses, enter each address in the box and choose Add email address. Repeat as necessary to add more email addresses, up to a maximum of 10.
Step 5: Configure Amazon CloudWatch alarms

You can monitor your protected resources for potential DDoS activity using CloudWatch alarms. To create an alarm, you specify an Amazon SNS topic for your resources in each Region. AWS Shield Advanced configures the SNS topic in CloudWatch to send you an email that alerts you to potential DDoS activity.

Note
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.

You can continue from this step without configuring CloudWatch alarms. However, doing so significantly reduces your visibility of possible DDoS events.

Note
In addition to the Amazon SNS notifications, 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. These statistics are recorded for a period of two weeks, so that you can access historical information and gain a better 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 on creating a CloudWatch dashboard, see Monitoring with Amazon CloudWatch (p. 282). For information about specific Shield Advanced metrics that you can add to your dashboard, see AWS Shield Advanced metrics and alarms (p. 284).

After you configure the CloudWatch alarms, we recommend that you follow the instructions in Step 6: Deploy AWS WAF rules (p. 258).

Step 6: Deploy AWS WAF rules

Several resources are available to help you quickly deploy AWS WAF rules. Consider taking advantage of one or more of the following offerings when creating your initial set of rules:

Security automation templates

AWS provides preconfigured templates that include a set of AWS WAF rules, which you can customize to best fit your needs. These templates are designed to block common web-based attacks such as bad bots, SQL injection, cross-site scripting (XSS), HTTP floods, and known-attacker attacks. In addition to activating Shield Advanced and specifying resources for Shield Advanced protection, you also should use these preconfigured templates.
For more information, see AWS WAF Security Automations. AWS WAF is included with Shield Advanced at no additional cost.

AWS Marketplace rule groups

AWS WAF 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 Marketplace sellers. For more information, see Managed rule groups (p. 21).

**AWS WAF for OWASP Top 10 Web Application Vulnerabilities**

This document outlines how you can use AWS WAF to mitigate the application vulnerabilities that are defined in the Open Web Application Security Project (OWASP) Top 10 list. This list shows the most common categories of application security flaws. For more information, see AWS WAF Security Automations.

**Managed Rules**

AWS provides managed rules to customers of AWS WAF who are subscribed to AWS Shield Advanced. These rules are written by the AWS Threat Research Team (TRT). They help protect against exploitation of a wide range of vulnerabilities, including those described in OWASP publications and many Common Vulnerabilities and Exposures (CVE). They also provide IP reputation rules that use threat intelligence from Amazon to block known malicious sources. You do not have to take any action to receive automatic updates to managed rules.

To request managed rules, open an “AWS Shield” case in the AWS Support Center. To learn more about opening an AWS support case, see Getting Started with AWS Support.

As a final step for getting started with Shield Advanced, review the global threat environment dashboard, as described in Step 7: Monitor the global threat environment dashboard (p. 259).

**Step 7: Monitor the global threat environment dashboard**

The global threat environment dashboard provides a near real-time summary of the global AWS threat landscape. The threat landscape includes the largest attack, the top attack vectors, and the relative number of significant attacks. To view the history of significant DDoS attacks, you can customize the dashboard for different time durations. For more information, see Monitoring threats across AWS (p. 299).

**Adding AWS Shield Advanced protection to AWS resources**

As part of enabling Shield Advanced for an account, you optionally choose initial resources to protect. You can add protection to more resources at any time. Shield Advanced offers advanced monitoring and protection for the following:

- Elastic IP addresses
- Amazon CloudFront distributions
- Amazon Route 53 hosted zones
- Elastic Load Balancing load balancers
- AWS Global Accelerator accelerators

You can monitor and protect up to 1,000 resources for each of these resource types per AWS account. For example, you could protect 1,000 IP addresses, 1,000 distributions, and 1,000 load balancers in a single account. If you want to increase the number of resources that you can protect, contact the AWS Support Center.
If you are using an AWS Firewall Manager Shield Advanced policy, you don't need to add resources using the procedure in this section. If the resource is within the Firewall Manager policy scope, that is, the resource is of the correct type and has the correct tags as defined by the policy, Firewall Manager automatically includes it within its protection.

**Important**
Before you perform the following procedure, you must complete the procedure in Step 1: Activate AWS Shield Advanced (p. 254).

**To add protection for an AWS resource**

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.
2. Choose **Protected resources**.
3. Choose **Add protected resources**.
4. Choose or enter the resource types and resources to protect. For Classic Load Balancer and Application Load Balancer resources, you also must choose a Region.

You can choose from the provided list or enter the Amazon Resource Name (ARN) of specific resources to protect. You can choose or enter any combination of resource types and resources.

Shield Advanced lists a maximum of 100 resources at one time. If you have more than 100 resources, choose **Next** to see the next set.

If you want to protect an Amazon EC2 instance, you must first associate an Elastic IP address with the instance, and then choose the Elastic IP address as the resource to protect.

**Note**
Shield Advanced does not support EC2-Classic.

5. Choose **Protect selected resources**.
6. (Optional) For **Add web ACLs and rules**, add a web ACL and rate-based rule as described in Step 3: Add rate-based rules (p. 256) in the Getting Started section. If you use Shield Advanced within an AWS Firewall Manager Shield Advanced policy, you can't add a rate-based rule.
7. (Optional) For **Configure enhanced DDos detection**, select an Amazon Route 53 health check to add to the protection and choose **Associate Health Check**.
8. (Optional) For **Create Amazon CloudWatch alarms and notifications**, configure a CloudWatch alarm as described in Step 5: Configure Amazon CloudWatch alarms (p. 258) in the Getting Started section.

**Note**
If you choose an Elastic IP address as the resource to protect, Shield Advanced protects whatever resource is associated with that Elastic IP address, either an Amazon EC2 instance or an Elastic Load Balancing load balancer. Shield Advanced automatically identifies the type of resource associated with the Elastic IP address and applies the appropriate mitigations for that resource, including configuring network ACLs specific to that Elastic IP address. For more information about using Elastic IP addresses with your AWS resources, see the appropriate guide: Amazon Elastic Compute Cloud Documentation or Elastic Load Balancing Documentation. Shield Advanced does not support EC2-Classic.

---

**Removing AWS Shield Advanced 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 console at https://console.aws.amazon.com/wafv2/.
2. Choose Protected resources.
3. Choose the radio button next to the resource.
4. Choose Delete protection.
   - If you have an Amazon CloudWatch alarm configured for this 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 above steps remove AWS Shield Advanced protection from a specific AWS resource. 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.

Managing AWS Shield Advanced protections

You can change the settings for your AWS Shield Advanced protections at any time.

Adding a Web ACL and rate-based rule to your Shield Advanced protections

For protection against attacks on Amazon CloudFront and Application Load Balancer resources, you can add AWS WAF web ACLs and rate-based rules. For information about how AWS WAF works, see AWS WAF (p. 6).

The following procedure shows how to add a web ACL and rate-based rule to a protection.

To add a web ACL and rate-based rule to a protection

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.
2. Choose Protected resources.
3. Choose the radio button next to the resource that you want to edit.
4. Choose Manage existing protections.
5. Choose the check box next to the resource that you want to edit.
6. Choose an existing web ACL and an existing rate-based rule. Alternatively, you can create a different web ACL and rate-based rule by following these steps:
   a. Choose Create new web ACL from the dropdown list.
b. Enter a name. You can't change the name after you create the web ACL.

c. Choose Create web ACL.

d. From the dropdown list, choose Create new rule.

e. Enter a name.

f. Enter a rate limit. This is the maximum number of requests from a single IP address that are allowed in a five-minute period.

g. Choose Create rule.

7. Choose the action to take if the rule is triggered. Block will block the request. Count will allow the request but increment a counter tracking how many times this rule was triggered.

8. Choose Apply web protections.

9. To skip the Amazon CloudWatch alarms page, uncheck all the check boxes and choose Create alarms.

Adding health-based detection to your Shield Advanced protections

You can use your Amazon Route 53 health checks to improve how AWS Shield Advanced detects and mitigates network-layer, transport-layer, and application layer attacks. For information about how this works, see the section called “Shield Advanced health-based detection” (p. 248).

To get started with health-based detection, create a Route 53 health check for the AWS resource that you want to protect with Shield Advanced. For information about Route 53 health checks, see How Amazon Route 53 Checks the Health of Your Resources and Creating and Updating Health Checks. After you create your health check, associate it with your protection using the following procedure.

Note
You are responsible for ensuring that the health check you use is relevant to the health of the protected resource and that it remains available for use by the protection. Shield Advanced doesn't manage the health check in any way.

The following procedure shows how to associate an Amazon Route 53 health check with a protection.

To associate a health check with a Shield Advanced protection

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.

2. Choose Protected resources.

3. Choose the resource that you want to edit.

4. Choose Manage existing protections.

5. If you need to skip the Add web ACLs and rules page, choose Skip and go to next step.

6. For Configure enhanced DDoS detection, choose the resource that you want to edit.

7. Under Associated Health Check, choose the ID of the health check that you want to associate with the protection.

    Note
    If you don't 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.

8. Choose Associate Health Check.

9. To skip the Amazon CloudWatch alarms page, either clear all the check boxes and choose Finish or choose Cancel. Canceling only skips the CloudWatch settings and doesn't cancel your changes to the health check settings.
Shield Advanced health check status settings

The status of the health check that you associate with a protection can have the following values in the Shield console:

- **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 this, first disassociate the health check from the protection in Shield Advanced. Then, in Route 53, create a new health check for the protection and note its ID. Finally, associate the new health check with the protection following the procedure in this topic. Don’t try to reassociate a health check that has been unavailable.

Adding a CloudWatch alarm to your Shield Advanced protections

The following procedure shows how to add a CloudWatch alarm to a protected resource.

Note
CloudWatch incurs additional costs. For CloudWatch pricing, see Amazon CloudWatch Pricing.

To add a CloudWatch alarm to a protected resource

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.
2. Choose Protected resources.
3. Choose the radio button next to the resource.
4. Choose Manage existing protections.
5. To skip the rate-based rule page, choose Continue.
6. For each Region that is listed in the table, choose an existing Amazon SNS topic or create a different topic. To create an Amazon SNS topic, follow these steps:
   a. Choose Create new topic from the dropdown list.
   b. Enter a topic name.
   c. Enter an email address that the Amazon SNS messages will be sent to, and then choose Add email address.
   d. Choose Create topic.
   e. Repeat as necessary for each protection and each rate-based rule.
7. Choose Create alarms.

Removing a CloudWatch alarm from your Shield Advanced protections

To remove a CloudWatch alarm from your Shield Advanced protections, you have two options:

- Delete the protection as described in Removing AWS Shield Advanced from an AWS resource (p. 260). 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 will start with DDoSDetectedAlarmForProtection.
Editing AWS Shield Advanced settings

You can change AWS Shield Advanced settings, such as adding or removing DDoS response team (DRT) access to your account or adding or removing emergency contact information.

**Edit AWS Shield Advanced settings**

2. Choose **Summary** under **AWS Shield** in the navigation pane.
3. Choose **Edit** under either **DDoS response team (DRT) support** or **Emergency contacts**.
4. Make the necessary changes and choose **Save**.

**Note**
To use the services of the DRT, you must be subscribed to the Business Support plan or the Enterprise Support plan.

AWS Shield Advanced: Requesting a credit

If you are subscribed to AWS Shield Advanced and a DDoS attack results in additional charges for your Amazon CloudFront, Elastic Load Balancing, Route 53 or Amazon EC2 services, you can apply for a credit for the charges by submitting a billing case through the AWS Support Center.

If the AWS Shield Advanced team determines that the incident is a valid DDoS attack and that the underlying services scaled to absorb the attack, AWS provides account credit for charges incurred due to the attack. For example, if your legitimate CloudFront data transfer usage during the attack period was 20 GB, but due to the attack you incurred charges for 200 GB of incremental data transfer, AWS provides credit to offset the incremental data transfer charges. AWS automatically applies all credits toward your future monthly bills. Credits are applied towards AWS Shield and cannot be used for payment for other AWS services. Credits are valid for 12 months.

**Important**
To be eligible for a credit, AWS must receive your credit request by the end of the second billing cycle after the incident occurred.

To request your credit, submit a billing query to the AWS Support Center that contains the following information:

- The words "DDoS Concession" in the subject line
- The dates and times of each incident interruption that you are claiming
- The AWS services (Amazon CloudFront, Elastic Load Balancing, Route 53, Amazon EC2) and specific resources that were affected by the DDoS activity

Security in 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.

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**

- Data protection in Shield (p. 265)
- Identity and access management in AWS Shield (p. 266)
- Logging and monitoring in Shield (p. 277)
- Compliance validation for Shield (p. 278)
- Resilience in Shield (p. 278)
- Infrastructure security in AWS Shield (p. 278)

## Data protection in Shield

Shield conforms to the AWS shared responsibility model, which includes regulations and guidelines for data protection. AWS is responsible for protecting the global infrastructure that runs all the AWS services. AWS maintains control over data hosted on this infrastructure, including the security configuration controls for handling customer content and personal data. AWS customers and APN partners, acting either as data controllers or data processors, are responsible for any personal data that they put in the AWS Cloud.

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.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM), so that 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.
- Set up API and user activity logging with AWS CloudTrail. See AWS CloudTrail API Reference.
- 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 personal data stored in Amazon S3.

We strongly recommend that you never put sensitive identifying information, such as your customers' account numbers, into free-form 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 piece of data that you enter into Shield or other services might get picked up for inclusion in diagnostic logs. When you provide a URL
to an external server, don’t include credentials information in the URL to validate your request to that server.

For more information about data protection, see the AWS Shared Responsibility Model and GDPR blog post on the AWS Security Blog.

Identity and access management in AWS Shield

Access to AWS Shield requires credentials. Those credentials must have permissions to access AWS resources, such as an AWS Shield resource or an Amazon S3 bucket. The following sections provide details on how you can use AWS Identity and Access Management (IAM) and Shield to help secure access to your resources.

- Authentication (p. 266)
- Access control (p. 267)

Authentication

You can access AWS as any of the following types of identities:

- **AWS account root user** – When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

- **IAM user** – An IAM user is an identity within your AWS account that has specific custom permissions (for example, permissions to create a rule in Shield). You can use an IAM user name and password to sign in to secure AWS webpages like the AWS Management Console, AWS Discussion Forums, or the AWS Support Center.

In addition to a user name and password, you can also generate access keys for each user. You can use these keys when you access AWS services programmatically, either through one of the several SDKs or by using the AWS Command Line Interface (CLI). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don’t use AWS tools, you must sign the request yourself. Shield supports Signature Version 4, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 Signing Process in the AWS General Reference.

- **IAM role** – An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user in that it is an AWS identity with permissions policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session. IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as federated users. AWS assigns a role to a federated user when access is requested through an identity
AWS service access – A service role is an IAM role that a service assumes to perform actions in your account on your behalf. When you set up some AWS service environments, you must define a role for the service to assume. This service role must include all the permissions that are required for the service to access the AWS resources that it needs. Service roles vary from service to service, but many allow you to choose your permissions as long as you meet the documented requirements for that service. Service roles provide access only within your account and cannot be used to grant access to services in other accounts. You can create, modify, and delete a service role from within IAM. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data from that bucket into an Amazon Redshift cluster. For more information, see Creating a Role to Delegate Permissions to an AWS Service in the IAM User Guide.

Applications running on Amazon EC2 – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances in the IAM User Guide.

Access control

You can have valid credentials to authenticate your requests, but unless you have permissions you can't create or access AWS Shield resources. For example, you must have permissions to create a Shield protection or list attacks.

The following sections describe how to manage permissions for AWS Shield. We recommend that you read the overview first.

- Overview of managing access permissions to your AWS Shield resources (p. 268)
- Using identity-based policies (IAM policies) for AWS Shield (p. 271)
- Shield required permissions for API actions (p. 275)

AWS Identity and Access Management

Shield integrates with AWS Identity and Access Management (IAM), a service that lets your organization do the following:

- Create users and groups under your organization's AWS account
- Share your AWS account resources with users in the account
- Assign unique security credentials to each user
- Control user access to services and resources

For example, you can use IAM with Shield to control which users in your AWS account can create a new web ACL.

For general information about IAM, see the following documentation:

- AWS Identity and Access Management (IAM)
- IAM Getting Started Guide
Overview of managing access permissions to your AWS Shield resources

Every AWS resource is owned by an AWS account, and permissions to create or access a resource are governed by permissions policies. An account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles). Some services also support attaching permissions policies to resources.

Note
An account administrator (or administrator user) is a user with administrator privileges. For more information, see IAM Best Practices in the IAM User Guide.

When granting permissions, you decide who is getting the permissions, the resources they get permissions for, and the specific operations that you want to allow on those resources.

Topics
- AWS Shield resources and operations (p. 268)
- Understanding resource ownership (p. 269)
- Managing access to resources (p. 269)
- Specifying policy elements: Actions, effects, resources, and principals (p. 270)
- Specifying conditions in a policy (p. 271)

AWS Shield resources and operations

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.

<table>
<thead>
<tr>
<th>Name in AWS Shield Console</th>
<th>Name in AWS Shield SDK/CLI</th>
<th>ARN Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incident or attack</td>
<td>AttackDetail</td>
<td>arn:aws:shield::account:attack/ID</td>
</tr>
<tr>
<td>Protection</td>
<td>Protection</td>
<td>arn:aws:shield::account:protection/ID</td>
</tr>
</tbody>
</table>

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:aws: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.
- 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:
Identity and access management

For more information, see Resources in the IAM User Guide.

AWS Shield provides a set of operations to work with Shield resources. For a list of available operations, see Actions.

Understanding resource ownership

A resource owner is the AWS account that creates the resource. That is, the resource owner is the AWS account of the principal entity (the root account, an IAM user, or an IAM role) that authenticates the request that creates the resource. The following examples illustrate how this works:

- If you use the root account credentials of your AWS account to create a Shield resource, your AWS account is the owner of the resource.
- If you create an IAM user in your AWS account and grant permissions to create a Shield resource to that user, the user can create a Shield resource. However, your AWS account, to which the user belongs, owns the Shield resource.
- If you create an IAM role in your AWS account with permissions to create a Shield resource, anyone who can assume the role can create a Shield resource. Your AWS account, to which the role belongs, owns the Shield resource.
- With AWS Shield, to create a protection or describe an attack associated with a specific resource, a user must have an access to the resource itself in addition to having access to the Shield resource. For example to create a protection for an Amazon CloudFront distribution, the user needs read access for the distribution to protect. To describe an attack against a CloudFront distribution, the user needs read access to the distribution.

Managing access to resources

A permissions policy describes who has access to what. The following sections explain the available options for creating permissions policies.

Note

These sections discuss using IAM in the context of AWS Shield. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see What Is IAM? in the IAM User Guide. For information about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the IAM User Guide.

Policies that are attached to an IAM identity are known as identity-based policies, and policies that are attached to a resource are known as resource-based policies. AWS Shield supports only identity-based policies.

Topics

- Identity-based policies (IAM policies) (p. 269)
- Resource-based policies (p. 270)

Identity-based policies (IAM policies)

You can attach policies to IAM identities. For example, you can do the following:

- Attach a permissions policy to a user or a group in your account – An account administrator can use a permissions policy that is associated with a particular user to grant permissions for that user to create an Shield resource.
- Attach a permissions policy to a role (grant cross-account permissions) – You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions. For example,
the administrator in Account A can create a role to grant cross-account permissions to another AWS account (for example, Account B) or an AWS service as follows:

1. Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions on resources in Account A.
2. Account A administrator attaches a trust policy to the role identifying Account B as the principal who can assume the role.
3. Account B administrator can then delegate permissions to assume the role to any users in Account B. Doing this allows users in Account B to create or access resources in Account A. The principal in the trust policy also can be an AWS service principal if you want to grant an AWS service permissions to assume the role.

For more information about using IAM to delegate permissions, see Access Management in the IAM User Guide.

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 must specify a wildcard character (*):

```json
{
    "Version": "2016-06-02",
    "Statement": [
        {
            "Sid": "ListProtections",
            "Effect": "Allow",
            "Action": [ "shield:ListProtections"
            ],
            "Resource": "*
        }
    ]
}
```

For more information about using identity-based policies with Shield, see Using identity-based policies (IAM policies) for AWS Shield (p. 271). For more information about users, groups, roles, and permissions, see Identities (Users, Groups, and Roles) in the IAM User Guide.

Resource-based policies

Other services, such as Amazon S3, also support resource-based permissions policies. For example, you can attach a policy to an S3 bucket to manage access permissions to that bucket. AWS Shield doesn't support resource-based policies.

Specifying policy elements: Actions, effects, resources, and principals

For each AWS Shield resource (see AWS Shield resources and operations (p. 268)), the service defines a set of API operations (see Shield required permissions for API actions (p. 275)). To grant permissions for these API operations, Shield defines a set of actions that you can specify in a policy. Note that performing an API operation can require permissions for more than one action. When granting permissions for specific actions, you also identify the resource on which the actions are allowed or denied.

The following are the most basic policy elements:
• **Resource** – In a policy, you use an Amazon Resource Name (ARN) to identify the resource to which the policy applies. For more information, see AWS Shield resources and operations (p. 268).

• **Action** – You use action keywords to identify resource operations that you want to allow or deny. For example, the `shield:CreateRuleGroup` permission allows the user permissions to perform the AWS Shield CreateRuleGroup operation.

• **Effect** – You specify the effect when the user requests the specific action. This can be either allow or deny. If you don't explicitly grant access to a resource, access is implicitly denied. You also can explicitly deny access to a resource, which you might do to make sure that a user cannot access it, even if a different policy grants access.

• **Principal** – In identity-based policies (IAM policies), the user that the policy is attached to is the implicit principal. AWS Shield doesn't support resource-based policies.

To learn more about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the IAM User Guide.

For a table that shows all the AWS Shield API actions and the resources that they apply to, see Shield required permissions for API actions (p. 275).

**Specifying conditions in a policy**

When you grant permissions, you can use the IAM policy language to specify the conditions when a policy should take effect. For example, you might want a policy to be applied only after a specific date. For more information about specifying conditions in a policy language, see Condition in the IAM User Guide.

To express conditions, you use predefined condition keys. There are no condition keys specific to Shield. However, there are AWS-wide condition keys that you can use as appropriate. For a complete list of AWS-wide keys, see Available Keys for Conditions in the IAM User Guide.

**Using identity-based policies (IAM policies) for AWS Shield**

This section provides examples of identity-based policies that demonstrate how an account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles) and thereby grant permissions to perform operations on AWS Shield resources.

**Important**

We recommend that you first review the introductory topics that explain the basic concepts and options available for you to manage access to your AWS Shield resources. For more information, see Overview of managing access permissions to your AWS Shield resources (p. 268).

The following shows an example of a permissions policy:

```json
{
  "Version": "2016-06-02",
  "Statement": [
    {
      "Sid": "CreateFunctionPermissions",
      "Effect": "Allow",
      "Action": [
        "shield:ListProtections",
        "shield:DescribeProtection",
        "shield:ListAttacks",
        "shield:DescribeAttack"
      ],
      "Resource": "*
    },
    {
      "Sid": "PermissionToPassAnyRole",
      "Effect": "Allow",
      "Action": [
        "shield:PassRole"
      ],
      "Resource": "*"
    }
  ]
}
```
The policy has two statements:

- The first statement grants permissions to view statistics for AWS Shield protections and actions. The policy specifies a wildcard character (*) as the Resource value.
- The second statement grants permissions for the IAM action iam:PassRole on IAM roles. The wildcard character (*) at the end of the Resource value means that the statement allows permissions for the iam:PassRole action on any IAM role. To only extend these permissions to a specific role, replace the wildcard character (*) in the resource ARN with the specific role name.

The policy doesn't specify the Principal element because in an identity-based policy you don't specify the principal who gets the permissions. When you attach a policy to a user, the user is the implicit principal. When you attach a permissions policy to an IAM role, the principal identified in the role's trust policy gets the permissions.

For a table that shows all the AWS Shield API actions and the resources that they apply to, see Shield required permissions for API actions (p. 275).

Topics

- Permissions required to use the AWS Shield console (p. 272)
- AWS managed (predefined) policies for AWS Shield (p. 272)
- Customer managed policy examples (p. 273)

Permissions required to use the AWS Shield console

The AWS Shield console provides an integrated environment for you to create and manage Shield resources. The console provides many features and workflows that often require permissions to create an Shield resource in addition to the API-specific permissions that are documented in the Shield required permissions for API actions (p. 275). For more information about these additional console permissions, see Customer managed policy examples (p. 273).

AWS managed (predefined) policies for AWS Shield

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. Managed policies grant necessary permissions for common use cases so you can avoid having to investigate what permissions are needed. For more information, see AWS Managed Policies in the IAM User Guide.

AWS Shield uses the AWS managed policy AWSShieldDRTAccessPolicy that you can use to grant the AWS Shield DDoS Response Team (DRT) access to your account. This allows the DRT to perform actions on your account, to manage your AWS WAF rules and Shield protections. To use this, you create a role and pass it to the Shield API operation, associate DRT role. In the API, this is AssociateDRTRole. In the CLI, it's associate-drt-role. For more information about this policy, see Step 4: (Optional) authorize the DDoS response team (p. 257).

Note

You can review AWS managed permissions policies by signing in to the IAM console and searching for the policies.
You also can create your own custom IAM policies to allow permissions for AWS Shield API operations and resources. You can attach these custom policies to the IAM users or groups that require those permissions or to custom execution roles (IAM roles) that you create for your Shield resources.

**Customer managed policy examples**

The examples in this section provide a group of sample policies that you can attach to a user. If you are new to creating policies, we recommend that you first create an IAM user in your account and attach the policies to the user, in the sequence outlined in the steps in this section.

You can use the console to verify the effects of each policy as you attach the policy to the user. Initially, the user doesn't have permissions, and the user won't be able to do anything on the console. As you attach policies to the user, you can verify that the user can perform various operations on the console.

We recommend that you use two browser windows: one to create the user and grant permissions, and the other to sign in to the AWS Management Console using the user's credentials and verify permissions as you grant them to the user.

For examples that show how to create an IAM role that you can use as an execution role for your Shield resource, see [Creating IAM Roles](#) in the *IAM User Guide*.

**Example topics**

- Example 1: Give users read-only access to Shield, CloudFront, and CloudWatch (p. 273)
- Example 2: Give users full access to Shield, CloudFront, and CloudWatch (p. 274)

**Create an IAM user**

First, you need to create an IAM user, add the user to an IAM group with administrative permissions, and then grant administrative permissions to the IAM user that you created. You then can access AWS using a special URL and the user's credentials.

For instructions, see [Creating Your First IAM User and Administrators Group](#) in the *IAM User Guide*.

**Example 1: Give users read-only access to Shield, CloudFront, and CloudWatch**

The following policy grants users read-only access to Shield an 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*",
                "elasticloadbalancing:List*",
                "route53:List*",
                "cloudfront:Describe*",
                "elasticloadbalancing:Describe*",
                "route53:Describe*",
                "cloudwatch:Describe*",
                "cloudwatch:Get*",
                "cloudwatch:List*",
                "cloudfront:GetDistribution*",
                "globalaccelerator:ListAccelerators",
...}
```
"globalaccelerator:DescribeAccelerator",
],
"Resource": [
  "arn:aws:elasticloadbalancing:*:*:*",
  "arn:aws:cloudfront::*:*",
  "arn:aws:route53:::hostedzone/*",
  "arn:aws:cloudwatch::*:*:*",
  "arn:aws:globalaccelerator::*:*"
],

"Sid": "ShieldReadOnly",
"Effect": "Allow",
"Action": [
  "shield:List*",
  "shield:Describe*",
  "shield:Get***"
],
"Resource": "*"
}

Example 2: Give users 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*",
        "elasticloadbalancing:List*",
        "route53:List*",
        "cloudfront:Describe*",
        "elasticloadbalancing:Describe*",
        "route53:Describe*",
        "cloudwatch:Describe*",
        "cloudwatch:Get***",
        "cloudwatch:List***",
        "cloudfront:GetDistribution***",
        "globalaccelerator:ListAccelerators",
        "globalaccelerator:DescribeAccelerator"
      ],
      "Resource": [
        "arn:aws:elasticloadbalancing:*:*:*",
        "arn:aws:cloudfront::*:*",
        "arn:aws:route53:::hostedzone/*",
        "arn:aws:cloudwatch::*:*:*",
        "arn:aws: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.

Shield required permissions for API actions

When you set up Access control (p. 267) and writing permissions policies that you can attach to an IAM identity (identity-based policies), use the information in this section as a guide. For each AWS Shield API operation, you need to know the actions for which you can grant permissions to perform the action, and the AWS resource for which you can grant the permissions. You specify the actions in the policy's Action field, and you specify the resource value in the policy's Resource field.

Note
To specify an action, use the shield: prefix followed by the API operation name (for example, shield:CreateProtection). The following list only includes actions that require explicit resource permissions.

You can use AWS-wide condition keys in your AWS Shield policies to express conditions. For a complete list of AWS-wide keys, see Available Keys for Conditions in the IAM User Guide.

For each action, we list the actions and the associated policy resource specifications.

**AssociateDrLogBucket**

**API Actions** – shield:AssociateDrLogBucket, s3:GetBucketPolicy, s3:PutBucketPolicy

**Resource** – arn:aws:s3:::bucket_name/optional_object_key

**AssociateDrtRole**

**API Actions** – shield:AssociateDrtRole, iam:GetRole, iam:ListAttachedRolePolicies, iam:PassRole

**Resource** – arn:aws:iam::account-id:role/role-id

**CreateProtection**

**API Actions** – shield:CreateProtection

**Resource** – arn:aws:shield::account:protection/ID

**DeleteProtection**

**API Actions** – shield:DeleteProtection

**Resource** – arn:aws:shield::account:protection/ID

**DescribeAttack**

**API Actions** – shield:DescribeAttack

**Resource** – arn:aws:shield::account:attack/ID

**DescribeDrtAccess**

**API Actions** – shield:DescribeDrtAccess, s3:GetBucketPolicy

**Resource** – arn:aws:s3:::bucket_name/optional_object_key
DescribeProtection

**API Actions** – shield:DescribeProtection

**Resource** – arn:aws:shield::account:protection/ID

DisassociateDRTLogBucket

**API Actions** – shield:DisassociateDRTLogBucket, s3:DeleteBucketPolicy, s3:GetBucketPolicy, s3:PutBucketPolicy

**Resource** – arn:aws:s3:::bucket_name/optional_object_key

For more information about Shield actions and resources, see the AWS Identity and Access Management guide topic Actions Defined by AWS Shield.

For a full list of the API actions available for Shield, see AWS Shield Advanced API Reference.
Logging and monitoring in 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 incidents:

**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 (p. 282).

**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 (p. 286).
Compliance validation for Shield

Third-party auditors assess the security and compliance of Shield as part of multiple AWS compliance programs, such as SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. 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 Shield is determined by the sensitivity of your data, your organization's compliance objectives, and applicable laws and regulations. If your use of Shield is subject to compliance with standards like HIPAA, PCI, or FedRAMP, AWS provides resources to help:

- **Security and Compliance Quick Start Guides** – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- **Architecting for HIPAA Security and Compliance Whitepaper** – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- **AWS Compliance Resources** – This collection of workbooks and guides might apply to your industry and location.
- **AWS Config** – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- **AWS Well-Architected Framework** – The AWS Well-Architected Framework helps you build secure cloud applications.

Resilience in 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

As a managed service, AWS Shield is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access Shield through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.
AWS Shield Advanced quotas

AWS Shield Advanced is subject to the quotas described here. These were formerly referred to as limits.

AWS Shield Advanced offers advanced monitoring and protection for Elastic IP addresses, Amazon CloudFront distributions, Amazon Route 53 hosted zones, Elastic Load Balancing load balancers or AWS Global Accelerator accelerators. You can monitor and protect up to 1000 of each of these resource types per account. If you want to increase these quotas, contact the AWS Support Center.
Monitoring AWS WAF, AWS Firewall Manager, and AWS Shield Advanced

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS WAF, AWS Firewall Manager, and AWS Shield Advanced and for identifying possible DDoS attacks using Shield Advanced. 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 (p. 280)
• Logging API calls with AWS CloudTrail (p. 286)

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:

• 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 periods.
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 Firewall Manager.

Not only can you use CloudWatch to monitor AWS WAF and Shield Advanced metrics as described in Monitoring with Amazon CloudWatch (p. 282), you also should use CloudWatch to monitor activity for your Amazon API Gateway and Elastic Load Balancing resources and Amazon CloudFront distributions. For more information, see Tracing, Logging, and Monitoring an API Gateway API, CloudWatch Metrics for Your Application Load Balancer, and Monitoring CloudFront Activity Using CloudWatch.

- **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 (p. 286) 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 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 (p. 19).
- 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.
View protection changes of your resources using AWS Config

You can 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 AWS CloudFormation template “Enable AWS Config” at AWS CloudFormation StackSets Sample Templates in the AWS 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.

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. These statistics are recorded for a period of two weeks, so that you can access historical information and gain a better 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.

Creating Amazon CloudWatch alarms

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.

AWS WAF and AWS Shield Advanced metrics and dimensions

You can use the following procedures to view the metrics for AWS WAF and AWS Shield Advanced.

Note
Amazon CloudWatch metrics and alarms are not enabled for AWS Firewall Manager.
To view metrics using the CloudWatch console

Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

2. If necessary, change the Region. From the navigation bar, choose the Region where your AWS resources are located. For more information, see AWS Regions and Endpoints.
   
   To view AWS WAF metrics for CloudFront, you must choose the US East (N. Virginia) Region.
3. In the navigation pane, choose Metrics.
4. On the All metrics tab, choose the appropriate service.

To view metrics using the AWS CLI

- For AWS WAF, at a command prompt use the following command:

```bash
aws cloudwatch list-metrics --namespace "WAF"
```

For Shield Advanced, at a command prompt use the following command:

```bash
aws cloudwatch list-metrics --namespace "DDoSProtection"
```

AWS WAF metrics

The WAF namespace includes the following metrics.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllowedRequests</td>
<td>The number of allowed web requests. Reporting criteria: There is a nonzero value. Valid statistics: Sum</td>
</tr>
<tr>
<td>BlockedRequests</td>
<td>The number of blocked web requests. Reporting criteria: There is a nonzero value. Valid statistics: Sum</td>
</tr>
<tr>
<td>CountedRequests</td>
<td>The number of counted web requests. Reporting criteria: There is a nonzero value. A counted web request is one that matches all the conditions in a particular rule. Counted web requests are typically used for testing. Valid statistics: Sum</td>
</tr>
<tr>
<td>PassedRequests</td>
<td>The number of passed requests for a rule group. Reporting criteria: There is a nonzero value.</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Passed requests</td>
<td>Passed requests are requests that don't match any rule contained in the rule group. Valid statistics: Sum</td>
</tr>
</tbody>
</table>

### AWS WAF dimensions

AWS WAF for Amazon CloudWatch can use the following dimension combinations:

- Rule, WebACL
- RuleGroup, WebACL
- Rule, RuleGroup

AWS WAF for an Application Load Balancer can use the following dimension combinations:

- Region, Rule, WebACL
- Region, RuleGroup, WebACL
- Region, Rule, RuleGroup

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule</td>
<td>One of the following:</td>
</tr>
<tr>
<td></td>
<td>• The metric name of the Rule.</td>
</tr>
<tr>
<td></td>
<td>• ALL, which represents all rules within a WebACL or RuleGroup.</td>
</tr>
<tr>
<td></td>
<td>• Default_Action (only when combined with the WebACL dimension), which represents the action assigned to any request that doesn't match any rule with either an allow or block action.</td>
</tr>
<tr>
<td>RuleGroup</td>
<td>The metric name of the RuleGroup.</td>
</tr>
<tr>
<td>WebACL</td>
<td>The metric name of the WebACL.</td>
</tr>
<tr>
<td>Region</td>
<td>The Region of the Application Load Balancer.</td>
</tr>
</tbody>
</table>

### AWS Shield Advanced metrics and alarms

This section discusses the metrics and alarms available with AWS Shield Advanced.

### AWS Shield Advanced metrics

Shield Advanced reports metrics to Amazon CloudWatch on an AWS resource more frequently during DDoS attacks than while no attacks are underway. Shield Advanced reports metrics once a minute during an attack, and then once right after the attack ends. While no attacks 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.

Shield Advanced provides the following metrics.
### Metric Table

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDoSDetected</td>
<td>Indicates whether a DDoS event is underway for a particular Amazon Resource Name (ARN). This metric has a value of 1 during an attack and a value of 0 otherwise.</td>
</tr>
<tr>
<td>DDoSAttackBitsPerSecond</td>
<td>The number of bits observed during a DDoS event for a particular Amazon Resource Name (ARN). This metric is available only for layer 3 and layer 4 DDoS events. This metric has a non-zero value during an attack and a value of 0 otherwise. Units: Bits</td>
</tr>
<tr>
<td>DDoSAttackPacketsPerSecond</td>
<td>The number of packets observed during a DDoS event for a particular Amazon Resource Name (ARN). This metric is available only for layer 3 and layer 4 DDoS events. This metric has a non-zero value during an attack and a value of 0 otherwise. Units: Packets</td>
</tr>
<tr>
<td>DDoSAttackRequestsPerSecond</td>
<td>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 attack and a value of 0 otherwise. Units: Requests</td>
</tr>
</tbody>
</table>

For the global services Amazon CloudFront and Amazon Route 53, metrics are reported in the US East (N. Virginia) Region.

Shield Advanced posts the DDoSDetected metric with no other dimensions. The other metrics include the appropriate AttackVector dimensions:

- ACKFlood
- ChragenReflection
- DNSSerialization
- GenericUDPRelfection
- MemcachedRelfection
- MSSQLRelfection
- NetBIOSRelfection
- NTPRelfection
- Portmapper
- RequestFlood
Creating AWS Shield Advanced alarms

You can use AWS Shield Advanced metrics for Amazon CloudWatch alarms. CloudWatch alarms send notifications or automatically make changes to the resources that you are monitoring based on rules that you define.

For detailed instructions on creating a CloudWatch alarm, see the Amazon CloudWatch User Guide. When creating the alarm on the CloudWatch console, after choosing Create an alarm, choose AWSDDOSProtectionMetrics to use the Shield Advanced metrics. You can then create an alarm based on a specific volume of traffic, or you can trigger an alarm whenever a metric is non-zero. The second option triggers an alarm for any potential attack observed by Shield Advanced.

Note
The AWSDDOSProtectionMetrics are available only to Shield Advanced customers.

For more information, see What is CloudWatch in the Amazon CloudWatch User Guide.

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 To create an Amazon SNS topic in Firewall Manager (console) (p. 207).

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
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 or IAM 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.

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 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": "2019-11-06T03:43:07Z",
    "eventType": "SampleEvent",
    "resources": [],
    "eventSource": "aws.waf",
    "eventSourceARN": "arn:aws:waf:us-east-1:112233445566:resource/123456789012",
    "eventBusName": "default",
    "awsRegion": "us-east-1",
    "eventName": "CreateWebACL",
    "userId": "A7ZC15QGZJG5K",  "recipientAccountId": "112233445566"
}
Example: CloudTrail log entry for GetWebACL
Example: CloudTrail log entry for UpdateWebACL

```json
{
    "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"
            }
        },
        "webIdFederationData": {},
        "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2019-11-06T19:17:20Z"
        }
    },
    "sessionTime": "2019-11-06T19:18:28Z",
    "eventSource": "wafv2.amazonaws.com",
    "eventName": "GetWebACL",
    "awsRegion": "us-west-2",
    "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": "7d563c6d-4123-4082-8880-c2d1d3a4d90b",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "apiVersion": "2019-04-23",
    "recipientAccountId": "112233445566"
}
```

API Version 2019-07-29
289
Example: CloudTrail log entry for UpdateWebACL

```json
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "principalId",
    "arn": "arn:aws:sts::112233445566:assumed-role/Admin/sheqiang-Isengard",
    "accountId": "112233445566"
  },
  "eventSource": "wafv2.amazonaws.com",
  "eventName": "UpdateWebACL",
  "awsRegion": "us-west-2",
  "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: AWS WAF classic log file entries

AWS WAF Classic is the prior version of AWS WAF. For information, see AWS WAF Classic (p. 76).

The log entry demonstrates the CreateRule, GetRule, UpdateRule, and DeleteRule operations:

```json
{
  "Records": [
    {
      "eventVersion": "1.03",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAIEP4IT4TPDEEXAMPLE",
        "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-west-2",
      "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-west-2",
"sourceIPAddress": "AWS Internal",
"userAgent": "console.amazonaws.com",
"requestParameters": {
"ruleId": "723c2943-82dc-4bc1-a29b-c7d73example"
},
"responseElements": null,
"requestID": "8e4f3211-5d48-11e3-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-west-2",
"sourceIPAddress": "AWS Internal",
"userAgent": "console.amazonaws.com",
"requestParameters": {
"ruleId": "7237b123-7903-4d9e-8176-9d71dexample",
"changeToken": "32343a11-35e2-4dab-81d8-6d408example",
"updates": [

}
AWS Shield Advanced supports logging the following actions as events in CloudTrail log files:

- ListAttacks
- DescribeAttack
- CreateProtection
- DescribeProtection
- DeleteProtection
- ListProtections
- CreateSubscription
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 or IAM 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.

```json
[
  {
    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "1234567890987654321231",
      "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-west-2",
    "sourceIPAddress": "AWS Internal",
    "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": "2018-01-10T21:31:14Z",
    "eventSource": "shield.amazonaws.com",
    "eventName": "ListProtections",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "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"
  }
]```
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 or IAM 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": "12345678908765432131",
        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/SampleUser",
        "accountId": "123456789012",
        "accessKeyId": "1AFGDT647FHU5JHI81H",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-04-14T02:51:50Z"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "1234567890987654321231",
                "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-df9afe95f9c1",
        "eventID": "5769af1e-14b1-4bd1-ba75-f023981d0a4a",
        "eventType": "AwsApiCall",
        "apiVersion": "2018-01-01",
        "recipientAccountId": "123456789012"
    }
}
```
Responding to DDoS attacks

AWS automatically addresses 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, which allows Shield Advanced to provide protection against larger DDoS events. For more information about network ACLs, see Network ACLs.

If DDoS alarms in CloudWatch indicate a possible layer 7 attack, you have two options:

- Investigate and mitigate the attack on your own. If you determine that the activity represents a DDoS attack, you can create your own AWS WAF rules to mitigate the attack. AWS WAF is included with AWS Shield Advanced at no additional cost. AWS provides preconfigured templates to get you started quickly. The templates include a set of AWS WAF rules, which are designed to block common web-based attacks. You can customize the rules to fit your business needs. For more information, see AWS WAF Security Automations and Creating a web ACL (p. 15).

- If you use AWS Firewall Manager, you can add these rules to a Firewall Manager-AWS WAF policy.

- If you are an AWS Shield Advanced customer, you also have the option of contacting 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 DRT, which has deep experience in protecting AWS, Amazon.com, and its subsidiaries.

To get DRT support, contact the AWS Support Center. 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 are 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.

**Important**

For layer 7 attacks, the DRT can help you analyze the suspicious activity, and then assist you to mitigate the issue. This mitigation often requires the DRT to create or update AWS WAF web access control lists (web ACLs) in your account. However, they need your permission to do so. We recommend that as part of enabling AWS Shield Advanced, you follow the steps in Step 4: (Optional) authorize the DDoS response team (p. 257) to proactively provide the DRT with the needed permissions. Providing permission ahead of time helps to prevent any delays in the event of an actual attack.

You can also contact the DRT before or during a possible attack to develop and deploy custom mitigations. For example, if you are running a web application and need only ports 80 and 443 open, you can work with the DRT to preconfigure a web ACL to “allow” only ports 80 and 443.

You authorize and contact the DRT 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 DRT for support. The Firewall Manager administrator can contact the DRT only for accounts that they own.
Reviewing DDoS incidents

AWS Shield Advanced provides real-time metrics and reports for extensive visibility into attacks on your AWS resources.

These metrics and reports are available only for AWS Shield Advanced customers. To activate AWS Shield Advanced, see To activate AWS Shield Advanced (p. 254).

You can view near real-time metrics about attacks, including the following:

- Attack type
- Start time
- Duration
- Blocked packet per second
- HTTP request samples

Details are available for active and past incidents that have occurred in the last 12 months.

Shield Advanced details report

Additionally, AWS Shield Advanced gives you insight into your overall traffic at the time of the attack. You can review details about the most prevalent:

- IP addresses
- URLs
- Referrers
- ASNs
- Countries
- User Agents

Use this information to create AWS WAF rules to help prevent future attacks. For example, if you see that you have a lot of requests coming from a country that you don't typically do business in, you can create an AWS WAF rule to block requests from that country.

**Note**

You should always test your rules first by initially using Count rather than Block. After you are comfortable that the new rule is identifying the correct requests, you can modify your rule to block those requests.

To review DDoS incidents


2. Choose the **Incident type** of the attack that you want to investigate.

The **Incidents** page provides the following possible statuses for current and past events:

**Mitigation in progress**

Indicates a possible layer 3 or 4 attack has been identified and AWS is attempting to address the issue.
Mitigated

Indicates a possible layer 3 or 4 attack was identified. AWS responded to the attack and the incident appears to be over.

Identified (ongoing)

Indicates a possible layer 7 attack has been identified. AWS cannot address layer 7 attacks, you must design your own layer mitigation processes. For more information on responding to possible layer 7 attacks, see Responding to DDoS attacks (p. 297).

Identified (subsided)

Indicates a possible layer 7 attack was identified and appears to be over.

If you determine a possible attack is underway, you can contact the DRT through the AWS Support Center, or attempt to mitigate the attack on your own by creating a new web access control list (web ACL).

To mitigate a potential DDoS attack

1. Create conditions in AWS WAF that match the unusual behavior.
2. Add those conditions to one or more AWS WAF rules.
3. Add those rules to a web ACL and configure the web ACL to count the requests that match the rules.
4. Monitor those counts to determine if the source of the requests should be blocked. If the volume of requests continues to be unusually high, change your web ACL to block those requests.

For more information, see Creating a web ACL (p. 15).

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.

Monitoring threats across AWS

If you are a Shield Advanced customer, in addition to the information provided on the Incidents page about attacks on your own resources, you can find more information on the global threat environment dashboard. On the dashboard, you can view trends and metrics about the DDoS threat landscape across Amazon EC2, Amazon CloudFront, Elastic Load Balancing, and Amazon Route 53.

The global threat environment dashboard provides a near real-time summary of the global AWS threat landscape, including the largest attack, the top attack vectors, and the relative number of significant attacks. You can customize the dashboard view for different time durations to see the history of significant DDoS attacks.

To view the global threat environment dashboard

1. Sign in to the AWS Management Console and open the AWS WAF console at https://console.aws.amazon.com/wafv2/.
2. In the navigation pane, under AWS Shield, choose Global threat environment.
3. Choose a time period.

You can use the information on the global threat environment dashboard to better understand the threat landscape and help you make decisions to better protect your AWS resources.
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 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 (p. 300)
- Making HTTPS requests to AWS WAF or Shield Advanced (p. 300)
- HTTP responses (p. 302)
- Authenticating requests (p. 303)

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 Step 3: Download tools (p. 5) in the topic Setting up (p. 3).

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. The various endpoints can be found in AWS Regions and 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 (p. 303).

**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 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=145b1567ab3c50d929412f28f52c45dfb1e63ec5c66023d232a539a4af11fd9
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 (known in the console as an IP match condition) to include the IP address 192.0.2.44 (represented in CIDR notation as 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, K2QH8DN0U907MN97FNA2GL18BD8V4KQNS0585MVJF66Q9ASUAAJG. 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).
Resources

The following related resources can help you as you work with this service.

AWS resources

Several helpful guides, forums, and other resources are available from Amazon Web 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.
- **Classes & Workshops** – Links to role-based and specialty courses as well as self-paced labs to help sharpen your AWS skills and gain practical experience.
- **AWS Developer Tools** – Links to developer tools, SDKs, IDE toolkits, and command line tools for developing and managing AWS applications.
- **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.
- **AWS Support** – The primary web page for information about AWS 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.
### Document history

<table>
<thead>
<tr>
<th>update-history-change</th>
<th>update-history-description</th>
<th>update-history-date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update to AWS Firewall Manager common security group policies</td>
<td>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.</td>
<td>March 11, 2020</td>
</tr>
<tr>
<td>Updated AWS Managed Rules for AWS WAF</td>
<td>AWS Managed Rules for AWS WAF added a AWSManagedRulesAnonymousIpList rule group.</td>
<td>March 6, 2020</td>
</tr>
<tr>
<td>Updated AWS Managed Rules for AWS WAF</td>
<td>AWS Managed Rules for AWS WAF updated the WordPress application and AWSManagedRulesCommonRuleSet rule groups.</td>
<td>March 3, 2020</td>
</tr>
<tr>
<td>Added Amazon Route 53 health check to AWS Shield Advanced protection options</td>
<td>Shield Advanced now supports the use of Amazon Route 53 health check associations, to improve the accuracy of threat detection and mitigation.</td>
<td>February 14, 2020</td>
</tr>
<tr>
<td>Updated AWS Managed Rules for AWS WAF</td>
<td>AWS Managed Rules for AWS WAF has updated the SQL Database rule group to add checking the message URI.</td>
<td>January 23, 2020</td>
</tr>
<tr>
<td>Firewall Manager new option for security group usage audit policy</td>
<td>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 noncompliant. By default, this minutes setting is zero.</td>
<td>January 14, 2020</td>
</tr>
<tr>
<td>Firewall Manager new option for AWS WAF policy</td>
<td>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.</td>
<td>January 14, 2020</td>
</tr>
<tr>
<td>Updated AWS Managed Rules for AWS WAF</td>
<td>AWS Managed Rules for AWS WAF has updated text transformations for rules in</td>
<td>December 20, 2019</td>
</tr>
</tbody>
</table>
### AWS Firewall Manager integrated with AWS Security Hub
AWS Firewall Manager now creates findings for resources that are out of compliance and for attacks and sends them to AWS Security Hub. **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
Shield Advanced can now protect AWS Global Accelerator. **November 26, 2018**
Earlier updates

The following table describes important changes in each release of the *AWS WAF Developer Guide*.

<table>
<thead>
<tr>
<th>Change</th>
<th>API Version</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS WAF support for Amazon</td>
<td>2016-08-24</td>
<td>AWS WAF now protects API Gateway APIs.</td>
<td>October 25, 2018</td>
</tr>
<tr>
<td>API Gateway</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expanded AWS shield advanced</td>
<td></td>
<td>New wizard provides opportunity to create rate-based rules and Amazon CloudWatch Events.</td>
<td>August 31, 2018</td>
</tr>
<tr>
<td>getting started wizard</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AWS WAF logging</td>
<td></td>
<td>Enable logging to get detailed information about traffic that is analyzed by your web ACL.</td>
<td>August 31, 2018</td>
</tr>
<tr>
<td>Support for query parameters</td>
<td>2016-08-24</td>
<td>When creating a condition, you can now search the requests for specific parameters.</td>
<td>June 5, 2018</td>
</tr>
<tr>
<td>in conditions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shield advanced getting</td>
<td>2016-08-24</td>
<td>Introduces a new streamlined process for subscribing to AWS Shield Advanced.</td>
<td>June 5, 2018</td>
</tr>
<tr>
<td>started wizard</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expanded allowed CIDR ranges</td>
<td></td>
<td>When creating an IP match condition, AWS WAF now supports IPv4 address ranges: /8 and any range between /16 through /32.</td>
<td>June 5, 2018</td>
</tr>
<tr>
<td>New Features</td>
<td>2015-08-24</td>
<td>You can now log all your API calls to AWS WAF through AWS CloudTrail, the AWS service that records API calls.</td>
<td>April 28, 2016</td>
</tr>
</tbody>
</table>

API Version 2019-07-29

308
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 AWS CloudTrail log. If you haven’t turned on AWS CloudTrail for your account, you can turn on CloudTrail from the AWS Management Console. There is no additional charge for turning on CloudTrail, but standard rates for Amazon S3 and Amazon SNS usage apply.

<table>
<thead>
<tr>
<th>New Features</th>
<th>API Version</th>
<th>Description</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-08-24</td>
<td>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 (p. 42).</td>
<td>March 29, 2016</td>
<td></td>
</tr>
</tbody>
</table>
| 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 URLs. For more information, see Size constraint rule statement (p. 40).  
• 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 Request component (p. 43). | January 27, 2016 |
| 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 |
| 2015-08-24   | This is the first release of the AWS WAF Developer Guide. | October 6, 2015 |
AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the AWS General Reference.