

AWS WAF (Web Application Firewall)

WAF stands for Web Application Firewall, and in the context of AWS (Amazon Web Services), AWS WAF is a service that helps protect web applications from common web exploits. It allows you to control and customize the traffic to your web applications by defining security rules. These rules can be configured to allow, block, or monitor (count) web requests based on specified conditions.

Key features of AWS WAF include:

1. **Rules:** You can create rules to specify the conditions under which certain requests should be allowed or blocked. For example, you can create rules to block requests from specific IP addresses or to block requests that match certain patterns indicative of common web attacks.
2. **Conditions:** Conditions are the building blocks of rules and are used to define what criteria should be met for a rule to take effect. Conditions can be based on various factors such as IP addresses, HTTP headers, or request parameters.
3. **WebACLs (Web Access Control Lists):** WebACLs are sets of rules that you can associate with a CloudFront distribution or an Application Load Balancer. They allow you to apply multiple rules to your web traffic.
4. **Integration with Other AWS Services:** AWS WAF can be integrated with other AWS services like Amazon CloudFront (a content delivery network) and Application Load Balancers, providing a scalable and distributed solution for protecting web applications.
5. **Managed Rules:** AWS WAF also provides managed rule sets, which are pre-configured sets of rules designed to address common threats. These managed rule sets are created and maintained by AWS security experts.

By using AWS WAF, you can enhance the security of your web applications by mitigating common security risks and protecting against various types of attacks, such as SQL injection, cross-site scripting (XSS), and more.

Uses Cases of WAF:

AWS WAF (Web Application Firewall) can be applied to various use cases to enhance the security of web applications. Here are some common use cases for AWS WAF:

1. **Protection Against Common Web Exploits:**
 - SQL Injection (SQLi) Protection:** WAF can be configured to detect and block SQL injection attempts, where attackers try to inject malicious SQL code into input fields to manipulate databases.
 - Cross-Site Scripting (XSS) Prevention:** WAF rules can be set up to identify and block attempts to inject malicious scripts into web pages, protecting against XSS attacks.
2. **Bot and Scraping Protection:** WAF can help identify and block automated bots and scrapers that might be attempting to scrape sensitive information from your website or impact your server's performance.
3. **Protection Against DDoS Attacks:** WAF can be used in conjunction with AWS services like Amazon CloudFront to protect against Distributed Denial of Service (DDoS) attacks by blocking or throttling suspicious traffic.
4. **Geographic Restriction:** WAF can be configured to allow or block traffic based on the geographic location of the source IP address. This can help in enforcing regional restrictions for compliance or security reasons.

5. **Bad Bot Mitigation:** Identify and block requests from known malicious bots or poorly-behaved bots that may be scraping content, attempting to exploit vulnerabilities, or causing other types of harm.
6. **API Security:** Protect APIs from common security threats by applying WAF rules to inspect and filter incoming API requests. This can include protecting against injection attacks, excessive rate limiting, and more.
7. **Custom Security Policies:** Create custom security policies tailored to the specific needs of your application. Define rules based on various parameters such as IP addresses, headers, query parameters, and request payloads.
8. **Compliance Requirements:** Address specific compliance requirements by using WAF to enforce security controls. For example, you can create rules to meet Payment Card Industry Data Security Standard (PCI DSS) requirements.
9. **Application Load Balancer Integration:** Integrate WAF with AWS Application Load Balancers to protect web applications hosted behind the load balancer.
10. **Zero-Day Exploit Protection:** Use WAF to quickly respond to new and emerging threats by updating rules to block or mitigate zero-day exploits before more comprehensive security measures can be implemented.

By utilizing AWS WAF for these use cases, organizations can significantly improve the security posture of their web applications and protect against a wide range of threats and vulnerabilities.

Disadvantages of WAF:

While Web Application Firewalls (WAFs) provide valuable security benefits, they also come with some potential disadvantages and challenges. It's important for organizations to be aware of these aspects when implementing a WAF:

1. **False Positives:** WAFs may generate false positives, flagging legitimate traffic as malicious. Tuning and refining rules are necessary to reduce false positives, but this process can be time-consuming.
2. **False Negatives:** On the flip side, WAFs may also produce false negatives, failing to detect certain types of attacks. Sophisticated attackers might find ways to bypass WAF protections.
3. **Complex Rule Configuration:** Configuring and maintaining WAF rules can be complex, especially for organizations with intricate web applications. Ensuring that rules are correctly configured without disrupting legitimate traffic requires expertise.
4. **Resource Overhead:** Implementing a WAF introduces additional processing overhead for inspecting and filtering web traffic. This can impact the performance of web applications, especially if not properly tuned.
5. **Learning Curve:** Organizations new to WAFs may face a learning curve in understanding how to configure and manage the system effectively. Training and expertise are essential for maximizing the benefits of a WAF.
6. **Continuous Monitoring and Updating:** WAFs require continuous monitoring and updating to stay effective against evolving threats. Regular updates to rules and policies are necessary to address new vulnerabilities and attack vectors.
7. **Limited Protection Against Advanced Threats:** While WAFs provide protection against common web application attacks, they may have limitations in defending against highly sophisticated and targeted threats that go beyond known attack patterns.

8. **Performance Impact:** Depending on the level of inspection and the complexity of rules, WAFs can introduce latency and affect the overall performance of web applications, especially during peak traffic times.
9. **Dependency on Signature-based Detection:** Many WAFs rely on signature-based detection methods to identify known attack patterns. This approach may be less effective against zero-day exploits and new, previously unseen attack techniques.
10. **Cost:** Implementing and maintaining a WAF may involve additional costs, including subscription fees for rule updates, hardware or cloud service costs, and potentially increased operational costs for managing and tuning the WAF.
11. **Configuration Errors:** Misconfigurations in WAF settings can inadvertently block legitimate traffic or allow malicious traffic to pass through. Regular audits and testing are essential to identify and rectify such errors.
12. **Dependence on Vendor Updates:** Organizations relying on managed rule sets provided by WAF vendors may be dependent on the vendor's update schedule. Delays in receiving updates could expose the organization to emerging threats.

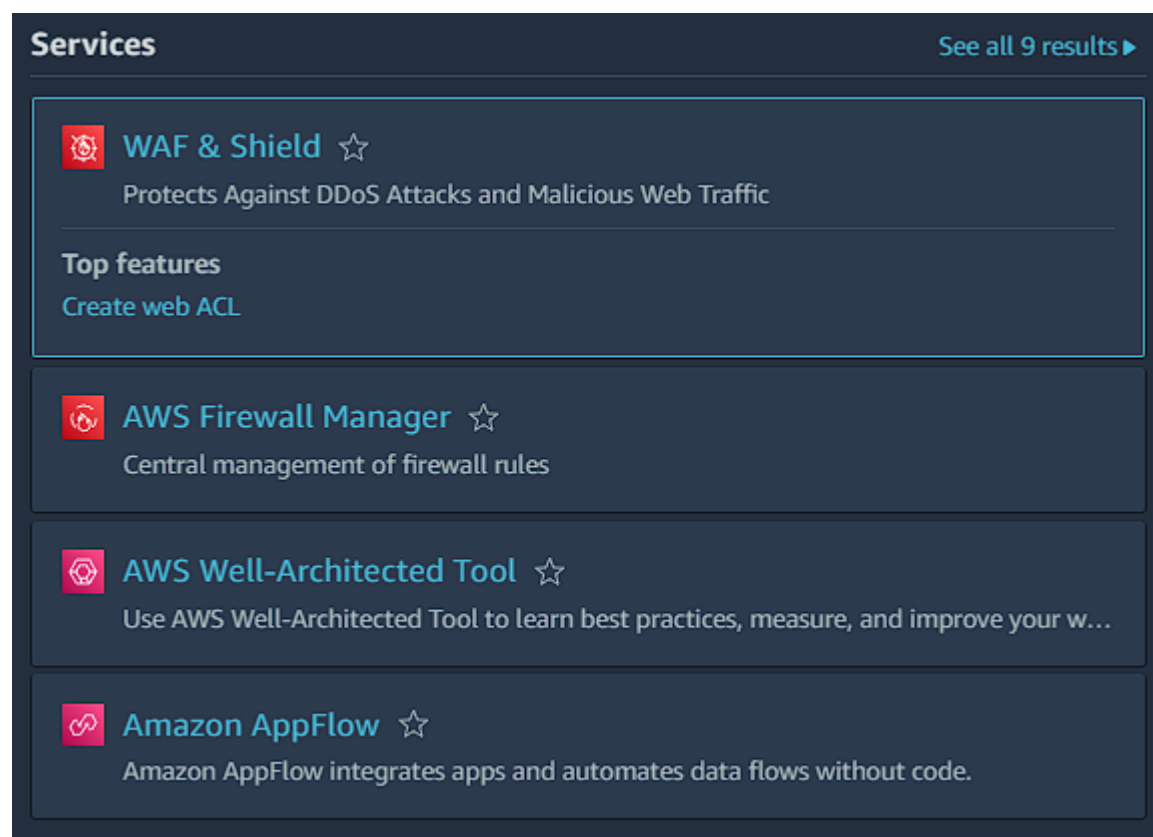
How to create WAF (Web Application Firewall):

Step 1:

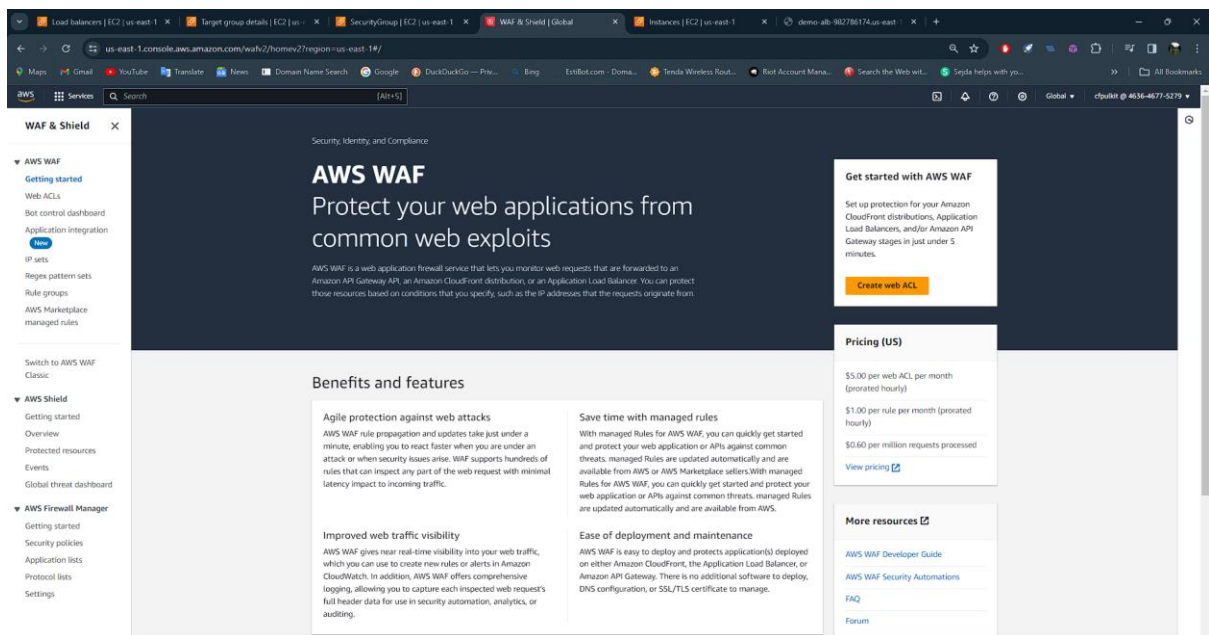
1. Create an Application Load Balancer. Which you might have already created.
2. Then move to AWS WAF.

Step 2: Create IP set

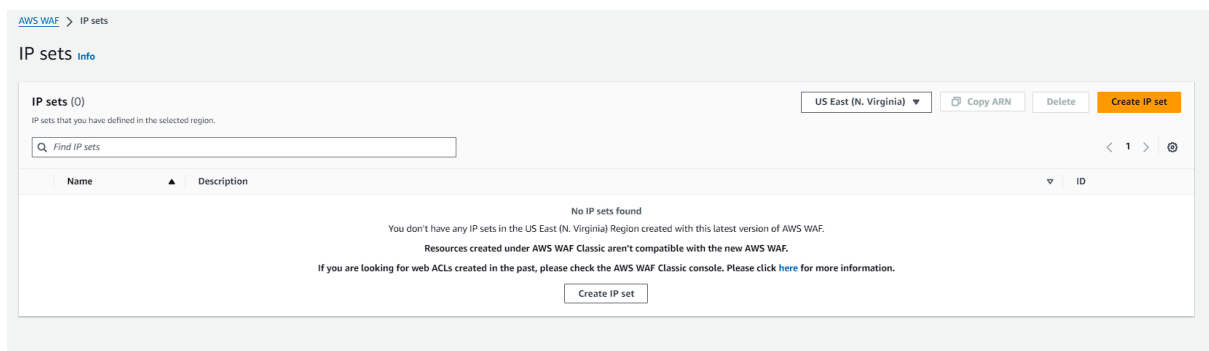
1. Go to AWS WAF



2. Open WAF and Shield.



3. Now go to IP sets and create an IP set. This will allow you to count the received requests.



4. Give IP set a name and select the region of your choice. Then select IPv4 in IP version and give your IP address. It is preferred to give your own IP address.

IP set details

IP set name

Demo-IP-set

The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Description - *optional*

The description can have 1-256 characters.

Region

Choose the AWS region to create this IP set in.

US East (N. Virginia) ▼

IP version

☒ IPv4

☐ IPv6

IP addresses

10.0.0.0/32

Enter one IP address per line in CIDR format.

Cancel

Create IP set

5. Click on Create IP set.

Success

You successfully created the IP set Demo-IP-set in the US East (N. Virginia) region.

[AWS WAF](#) > IP sets

IP sets [Info](#)

IP sets (1/1)

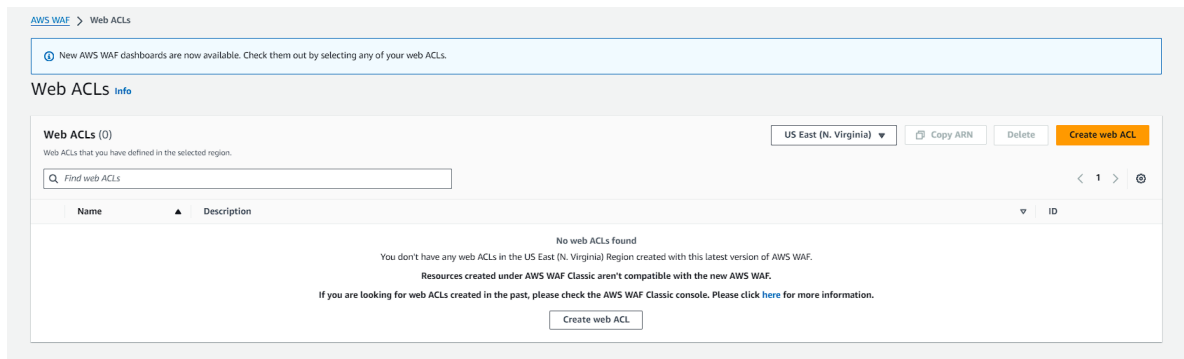
IP sets that you have defined in the selected region.

US East (N. Virginia) ▼ Copy ARN Delete Create IP set

Find IP sets

Name	Description	ID
Demo-IP-set	-	0204da8a-01b2-473a-a6bf-93257e8bb432

Step 2: Create Web ACL



1. Click on create web ACL. Give it a name.
2. Select your region if it is not selected by default.

Describe web ACL and associate it to AWS resources [Info](#)

Web ACL details

Resource type
Choose the type of resource to associate with this web ACL. Changing this setting will reset the page.

☐ Amazon CloudFront distributions

☒ Regional resources (Application Load Balancers, Amazon API Gateway REST APIs, Amazon App Runner services, AWS AppSync GraphQL APIs, Amazon Cognito user pools and AWS Verified Access Instances)

Region
Choose the AWS Region to create this web ACL in. Changing this setting will reset the page.

US East (N. Virginia) ▼

Name

Demo-Web-ACL

The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Description - optional

The description can have 1-256 characters.

CloudWatch metric name

Demo-Web-ACL

The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

3. Now Associate your AWS resource which in your case is Application load balancer.

Associated AWS resources - *optional* (0)

Remove

Add AWS resources

< 1 > ⚙

	Name	Resource type	Region
No items No items to display			

Cancel

Next

4. Click add and it will be selected. Then click next.

Add AWS resources

×

Resource type
Select the resource type and then select the resource you want to associate with this web ACL.

☒ Application Load Balancer

☐ Amazon API Gateway REST API

☐ Amazon App Runner service

☐ AWS AppSync GraphQL API

☐ Amazon Cognito user pool

☐ AWS Verified Access

Select the resources you want to associate with the web ACL.

< 1 > ⚙

<input checked="" type="checkbox"/>	Name
<input checked="" type="checkbox"/>	Demo-ALB

Cancel

Add

Associated AWS resources - *optional* (1)

Remove

Add AWS resources

< 1 > ⚙

<input type="checkbox"/>	Name	Resource type	Region
<input type="checkbox"/>	Demo-ALB	Application Load Balancer	US East (N. Virginia)

Cancel

Next

- On the next page in the add rules, click on it and select Add my own rules and rule group.

Rules (0)

EditDeleteAdd rules ▼

If a request matches a rule, take the corresponding action. The rules are prioritized in order they appear.

	Name	Capacity	Action
No rules. You don't have any rules added.			

Add rules ▲

Add managed rule groups

Add my own rules and rule groups

- Then select IP set over here.

Rule type

Rule type

☒ IP set
Use IP sets to identify a specific list of IP addresses.

☐ Rule builder
Use a custom rule to inspect for patterns including query strings, headers, countries, and rate limit violations.

☐ Rule group
Use a rule group to combine rules into a single logical set.

- Give a name to the Rule. Then select the IP set. Choose Source IP address and in the Action select Count. Now click on add rule.

Rule

Name

Demo-set-rule

The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

IP set

IP set

Demo-IP-set ▼

IP address to use as the originating address
When a request comes through a CDN or other proxy network, the source IP address identifies the proxy and the original IP address is sent in a header. Use caution with the option, IP address in header, because headers can be handled inconsistently by proxies and they can be modified to bypass inspection.

☒ Source IP address
☐ IP address in header

Action
Choose an action to take when a request originates from one of the IP addresses in this IP set.

☐ Allow
☐ Block
☒ Count
☐ CAPTCHA
☐ Challenge

► Custom request - *optional*

Cancel

Add rule

8. It would look like this. Now just click on next. The default action should select to Allow.

Rules (1)

Edit

Delete

Add rules ▼

If a request matches a rule, take the corresponding action. The rules are prioritized in order they appear.

<input type="checkbox"/>	Name	Capacity	Action
<input type="checkbox"/>	Demo-set-rule	1	Count

Web ACL capacity units (WCUs) used by your web ACL

The WCUs used by the web ACL will be less than or equal to the sum of the capacities for all of the rules in the web ACL.

The total WCUs for a web ACL can't exceed 5000. Using over 1500 WCUs affects your costs. [AWS WAF Pricing](#)

1/5000 WCUs

Default web ACL action for requests that don't match any rules

Default action

☒ Allow

☐ Block

Custom request - optional

Token domain list - optional

Enable the use of tokens across multiple protected applications by entering the application domains here. Tokens are used by the Challenge and CAPTCHA rule actions, the application integration SDKs, and the ATP and Bot Control managed rule groups. [Learn more](#)

Add token domain

You can add 10 more domains

Cancel

Previous

Next

Set rule priority [Info](#)

Rules (1/1)

▲ Move up

▼ Move down

If a request matches a rule, take the corresponding action. The rules are prioritized in order they appear.

	Name	Capacity	Action
<input checked="" type="radio"/>	Demo-set-rule	1	Count

Cancel

Previous

Next

Configure metrics [Info](#)

Amazon CloudWatch metrics

CloudWatch metrics allow you to monitor web requests, web ACLs, and rules.

Rules

☒ Demo-set-rule

CloudWatch metric name

Demo-set-rule

Request sampling options

If you disable request sampling, you can't view requests that match your web ACL rules.

Options

- ☒ Enable sampled requests
- ☐ Disable sampled requests
- ☐ Enable sampled requests with exclusions

Cancel

Previous

Next

9. Click on next and review the ACL and create it. It might take some time to get created.

Success

You successfully created the web ACL Demo-Web-ACL.

AWS WAF > Web ACLs

New AWS WAF dashboards are now available. Check them out by selecting any of your web ACLs.

Web ACLs [Info](#)

Web ACLs (1)

Web ACLs that you have defined in the selected region.

Find web ACLs

< 1 > ⌕

Name	Description	ID
Demo-Web-ACL	-	7ba1c52d-799b-42fd-80aa-c2691b3e560c

US East (N. Virginia) Copy ARN Delete Create web ACL

AWS WAF > Web ACLs > Demo-Web-ACL

Demo-Web-ACL

Download web ACL as JSON

Traffic overview

Rules

Associated AWS resources

Custom response bodies

Logging and metrics

Sampled requests New

CloudWatch Log Insights

You can learn more about the dashboards [here](#). Please provide feedback for the dashboards.

Feedback

Sampled requests now has its own tab: Sampled requests.

×

Data filters [Info](#)

Select the time range and terminating actions that you want to view in the dashboard. You can select a time range relative to now and you can select an absolute time range.

Terminating rule actions

Time range

Time zone

Refresh

Blocked

Allowed

Captcha

Challenge

All traffic

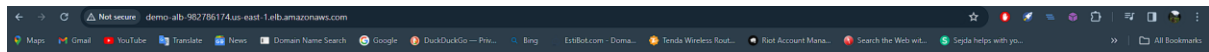
Bot Control

▼ Action totals for the specified time range - all traffic

Request counts for all traffic during the specified time range. This shows counts for all possible terminating actions, while the rest of the dashboard shows only the actions that you've selected in the filters. If you're filtering on a relative time range, each action also shows the percentage change from the prior, equivalent-length time range. For example, if you've chosen 1 day as the time range, the percentage change reflects the difference between 48-24 hours ago and 24-0 hours ago.

Total	Blocked	Allowed	Captcha	Challenge
0	0	0	0	0

10. Once it is created, now go back to Application Load Balancer and copy its DNS name and paste it in a new tab.

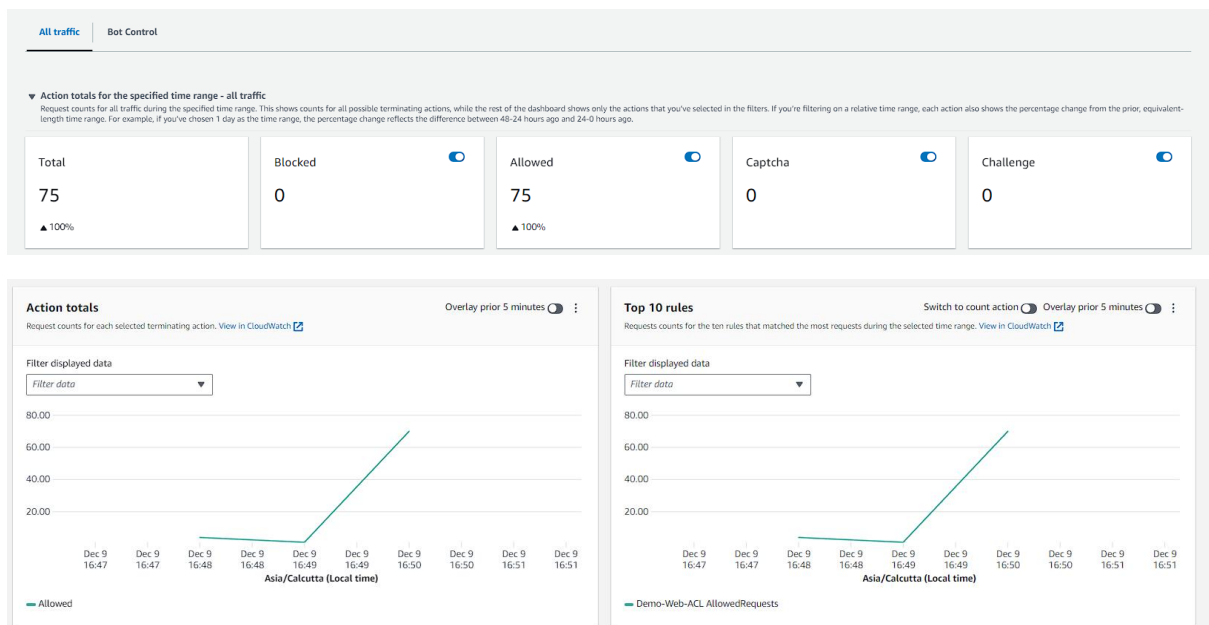


Hello World from ip-172-31-40-108.ec2.internal

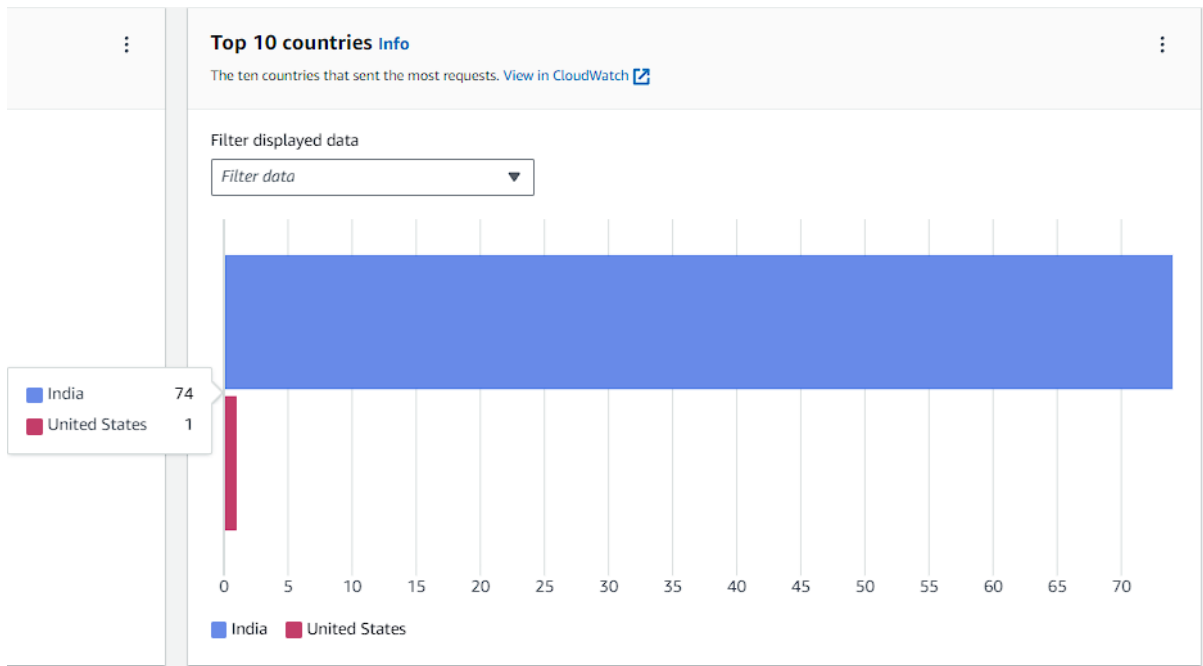


Hello World from ip-172-31-23-149.ec2.internal

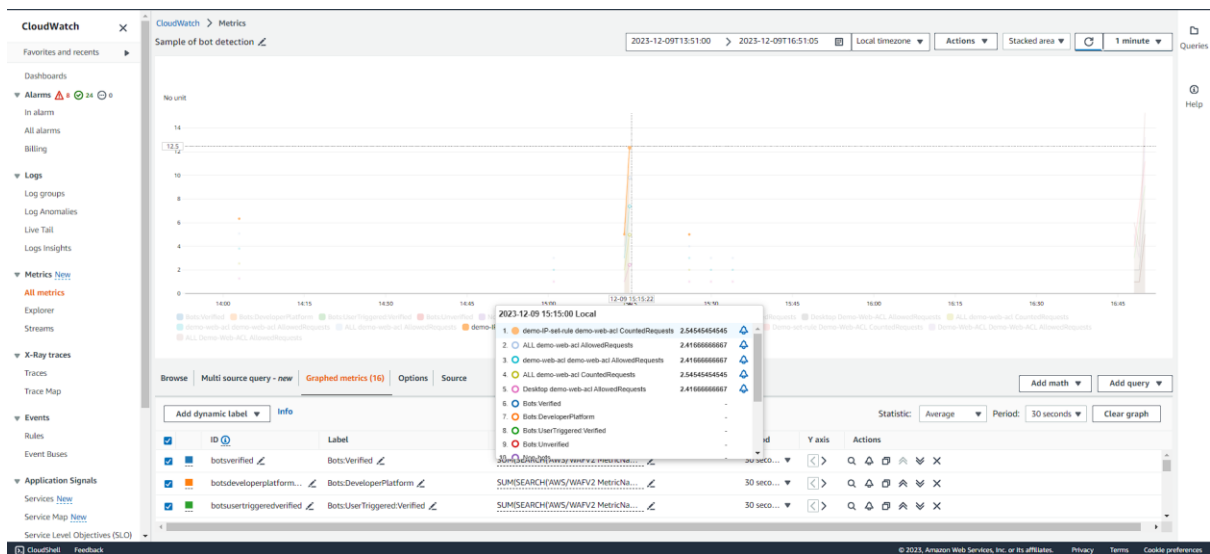
11. It will switch to different IPs and WAF will count it and tell you that how many time the request was approved. So, just refresh it many times as you can and see the metrics in cloud watch.
12. So, as you can see the ALB requested 75 times and it was allowed each and every time it requested to the WAF



13. It will also tell you that from which country the ALB is requesting despite of using any region of your choice.

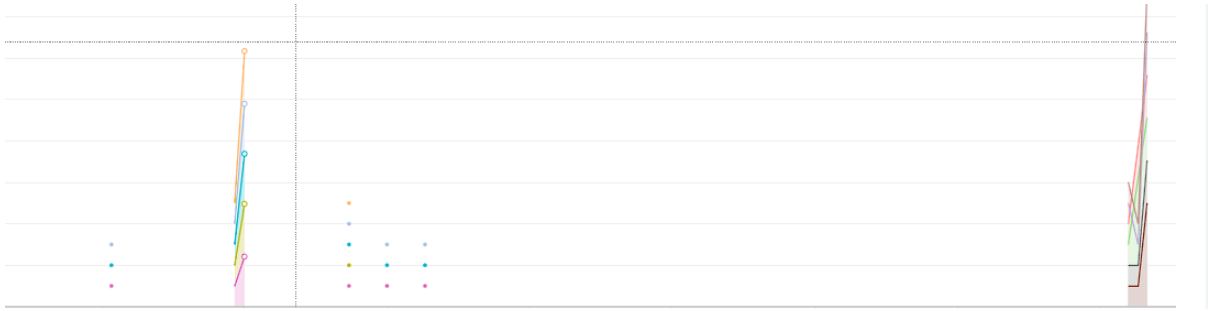


14. Now open Cloud Watch. You can open it by clicking on the view cloudwatch option. And here you'll see the proper graph.



2023-12-09 15:15:00 Local

1.	demo-IP-set-rule demo-web-acl CountedRequests	2.54545454545	
2.	ALL demo-web-acl AllowedRequests	2.41666666667	
3.	demo-web-acl demo-web-acl AllowedRequests	2.41666666667	
4.	ALL demo-web-acl CountedRequests	2.54545454545	
5.	Desktop demo-web-acl AllowedRequests	2.41666666667	
6.	Bots:Verified	-	
7.	Bots:DeveloperPlatform	-	
8.	Bots:UserTriggered:Verified	-	
9.	Bots:Unverified	-	



Step 3: Blocking all the Request From our IP address.

1. Now if you want to block all the request from your IP address or some other IP address. Then it can be done easily.
2. Go back to Web ACL and click on Rules.

Demo-Web-ACL

Traffic overview **Rules** Associated AWS resources Custom response bodies Logging and metrics Sampled requests New CloudWatch Log Insights

3. Here select your rule and click on edit.

Rules (1/1)

☐ Name Action Priority Custom response

<input checked="" type="checkbox"/>	Demo-set-rule	Count	0	-
-------------------------------------	---------------	-------	---	---

4. Now scroll down a little. Select Action to Block and save the rule.

Then

Action

Action

Choose an action to take when a request matches the statements above.

☐ Allow

☒ Block

☐ Count

☐ CAPTCHA

☐ Challenge

► Custom response - optional

► Add label - optional

Add labels to requests that match this rule. Rules that are evaluated later in the same web ACL can reference the labels that this rule adds.

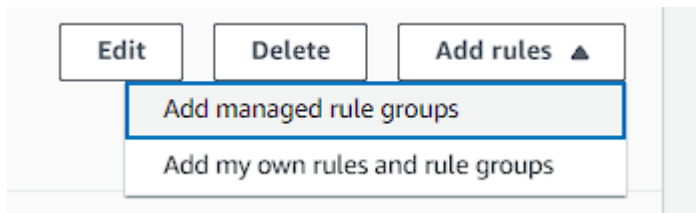
Cancel Save rule

Rules (1)

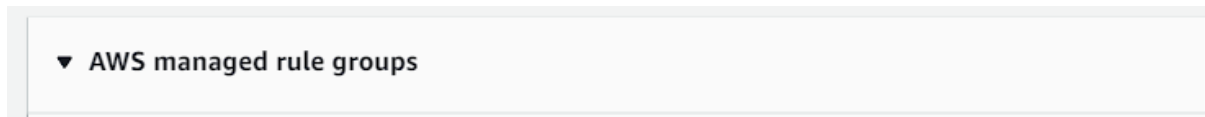
☐ Name Action Priority Custom response

<input type="checkbox"/>	Demo-set-rule	Block	0	-
--------------------------	---------------	-------	---	---




5. Now select your rule again and click on Add managed rule groups.



6. Select AWS managed rule groups. And scroll down to free rule groups.



7. In the free rule groups, add Amazon IP reputation list and click on edit.

Free rule groups You can use the free rule groups without any added charges beyond the standard service charges for AWS WAF. AWS WAF Pricing 		
Name	Capacity	Action
Admin protection Contains rules that allow you to block external access to exposed admin pages. This may be useful if you are running third-party software or would like to reduce the risk of a malicious actor gaining administrative access to your application. Learn More 	100	<input type="checkbox"/> Add to web ACL
Amazon IP reputation list This group contains rules that are based on Amazon threat intelligence. This is useful if you would like to block sources associated with bots or other threats. Learn More 	25	<input checked="" type="checkbox"/> Add to web ACL <div>Edit</div>

8. Set override all rule action to Count. And click on Save rule.

Amazon IP reputation list rules

The rules apply actions and labels to requests that match their criteria. [Learn More](#)

By default, the rule group uses its configured rule actions. You can override the actions for all rules and for individual rules. For a single rule, use the rule dropdown to specify an override action or to remove an override.

Allow and Block actions terminate web ACL evaluation for matching requests. Count action counts matching requests and continues the web ACL evaluation. [Learn More](#)

Override all rule actions

Override to Count ▼

Remove all overrides

AWSManagedIPReputationList

Rule action: **Block**

Override to Count ▼

AWSManagedReconnaissanceList

Rule action: **Block**

Override to Count ▼

AWSManagedIPDDoSList

Rule action: **Count**

Override to Count ▼

► **Override rule group action - optional**

Cancel

Save rule

9. You will need to set the priority. Now click on save.

[AWS WAF](#) > [Web ACLs](#) > [Demo-Web-ACL](#) > Set rule priority

Set rule priority [Info](#)

Rules (1/2)

▲ Move up

▼ Move down

If a request matches a rule, take the corresponding action. The rules are prioritized in order they appear.

	Name	Capacity	Action
<input checked="" type="radio"/>	Demo-set-rule	1	Block
<input type="radio"/>	AWS-AWSManagedRulesAmazonIpReputationList	25	Use rule actions

Cancel

Save

Rules (2)

Find rules

Edit

Delete

Add rules ▼

< 1 >

ⓘ

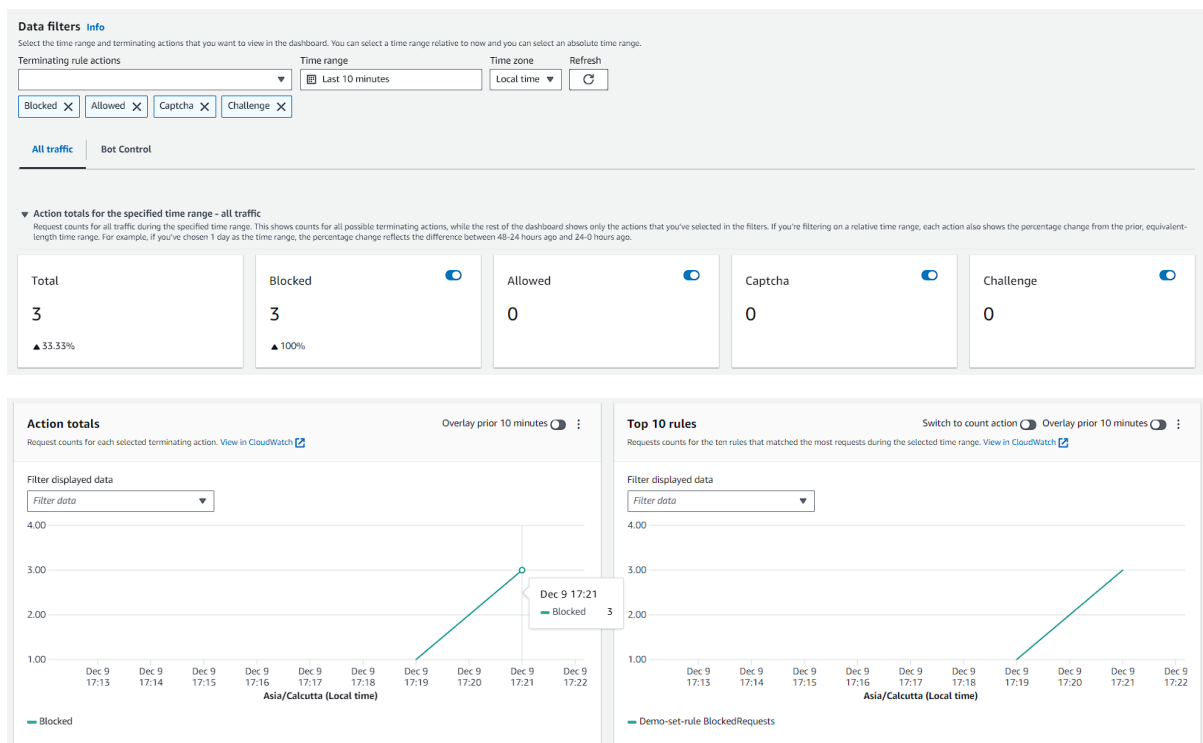
<input type="checkbox"/>	Name	Action	Priority	Custom response
<input type="checkbox"/>	Demo-set-rule	Block	0	-
<input type="checkbox"/>	AWS-AWSManagedRulesAmazonIpReputationList	Use rule actions	1	-

10. After completing these steps now go back to the ALB(Application Load Balancer) and copy the DNS name and copy it in a new tab. It will show you a message like this because the WAF has blocked the request that was made to it by ALB

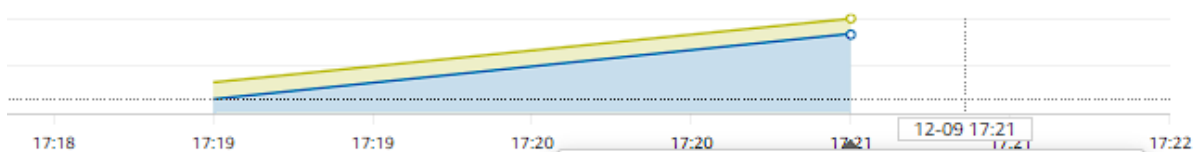
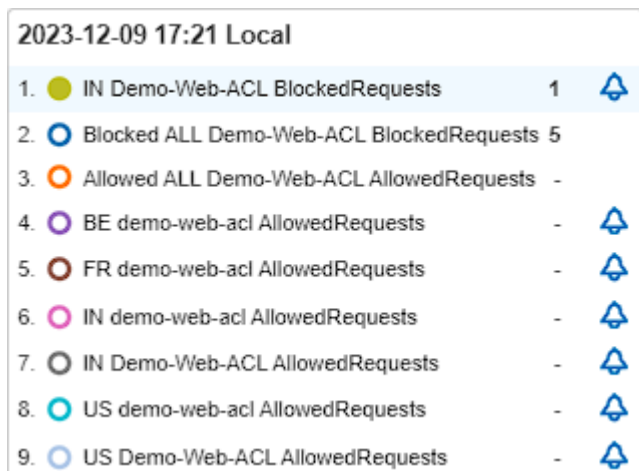


403 The forbidden

11. So now if you go back to Web ACL and refresh it. You'll see the number of requests that were blocked.



12. If you go to see the metrics in cloud watch it'll give you the stats.



Step 4: Deleting all the resources

1. First go to Web ACL, open it and go to Associated AWS resources and Disassociate it.

Traffic overview | Rules | **Associated AWS resources** | Custom response bodies | Logging and metrics | Sampled requests **New** | CloudWatch Log Insights

Associated AWS resources (1/1) Disassociate Add AWS resources

<input checked="" type="checkbox"/>	Name	Resource type	Region
<input checked="" type="checkbox"/>	Demo-ALB	Application Load Balancer	US East (N. Virginia)

2. Now simply delete your Web ACL.
3. Go to IP Sets and delete it.
4. Now go to EC2 and navigate to load balancers. First delete your load balancer then delete your target groups.
5. Then just terminate your EC2 instances.