AWS Web Application Firewall (WAF)

1. **Overview**

- **AWS WAF** is a security service that protects our web applications from bad or malicious traffic coming from the internet.
- It can be enabled to protect a CloudFront distribution, an Application Load Balancer (ALB), or the API Gateway.
- Because CloudFront can sit in front of any server (AWS or non-AWS), AWS WAF can protect applications running anywhere, not just on AWS.
- If all API requests go through API Gateway, AWS WAF can protect the API even if it's hosted outside AWS.
- WAF does NOT work with Network Load Balancer (NLB) because NLB works at Layer 4.
- It checks every incoming request (visitor).
- Allows good users. Blocks attackers.
- Follows the rules we can define.
- It works at Layer 7 (Application Layer), meaning it understands: URLs, HTTP headers, Query Strings.
- A **Web ACL** is a **rule book**. It decides: What traffic is allowed or blocked. Every WAF setup **must have a Web ACL.** Default Web ACL capacity is 1500 units, can be increased via AWS Support request.
- Rules are **conditions** inside a Web ACL. Example: If IP = 1.2.3.4 → Block then if request contains SQL keywords → Block.
- Each rule takes an **action:**

  - Allow ---> Let the request pass

  - Block ---> Reject the request

  - Count ---> Only monitor, don't block
- If **no rule matches**, WAF follows the **default action**: Allow all or Block all.
- A **Rule Group** is a **collection of rules**. They are of two types: AWS Managed Rule Groups and Custom Rule Groups. They are reusable across multiple Web ACLs.
- Managed rules are **ready-made security rules** maintained by AWS.
- An **IP Set** is a list of IP addresses.

AWS Web Application Firewall (WAF)

2. **Use-cases**

2.1 ***Common Attacks It Protects Against***

- SQL Injection - Attacker tries to steal or modify database data using malicious input.
- Cross-Site Scripting (XSS) - Injecting malicious scripts into web pages.
- Brute Force - Repeated login attempts to guess passwords.
- Bot Attacks - Automated tools flooding your site.
- DDoS (App Layer) - Overloading your application with fake traffic.

2.2 ***Real-World Use-Cases***

- Protect a Public Website: Prevent hackers from exploiting input fields.
- Secure login pages: Block repeated failed login attempts.
- Restrict access by country: Example: Allow traffic only from Saudi.
- Block malicious IP addresses.
- Allow only trusted partners: Example: APIs accessed only from fixed IPs.

3. **Architecture Flow**

- At a high level, **AWS WAF works as a security filter** that sits **in front of your application** and decides whether incoming requests should be **allowed or blocked**.
- The core components involved are Web ACL, Rules and Rule Groups, Protected AWS Resources (such as ALB, CloudFront, or API Gateway).
- AWS WAF works by attaching a Web ACL to supported AWS resources and inspecting every incoming request before it reaches the application. Requests are evaluated against rules in a top-down order, using transformations and attack detection logic. Based on the rule action, the request is either allowed, blocked with a 403 response, or counted for monitoring. Logging and metrics provide visibility without exposing security details to attackers.

- Step 1: Web ACL Is Attached to a Resource.
  - A **Web ACL** is created and attached to one or more supported AWS resources such as: ALB, CloudFront, API Gateway.

AWS Web Application Firewall (WAF)

- The Web ACL acts as the **central rule book** for traffic filtering.

❖ Step 2: Client Sends a Request.

- A user (or attacker) sends an **HTTP or HTTPS request** to the application.

- Example: Opening a website, submitting a login form or Calling an API endpoint.

❖ Step 3: Request Is Forwarded to AWS WAF.

- Before the request reaches the application:

- The **protected AWS resource automatically forwards the request to AWS WAF.**

- The application itself does **not** see the request yet.

- At this point, AWS WAF is the **decision-maker**.

❖ Step 4: Rules Are Evaluated Top to Bottom.

- Inside the Web ACL, **rules are evaluated in order**, from top to bottom.

- Each rule checks specific parts of the request, such as:

- URL (example: `/login`)

- HTTP headers (example: User-Agent)
- Request body (form data or JSON)
- HTTP method (GET, POST, PUT, DELETE)
- Source IP address
- Country of origin

❖ Step 5: How Rules Evaluate Requests.

❖ Rules can evaluate requests using different methods, such as:

- String matching (checking for specific words)

- Regular expressions (regex) for patterns

- Size checks (request too large)

- Built-in attack detection, such as: SQL Injection detection and Cross-Site Scripting (XSS) detection.

❖ Step 6: Text Transformations Are Applied.

❖ Before evaluation, AWS WAF can **transform request data** to reveal hidden attacks.

- Examples of transformations:

- URL decoding

- Converting text to lowercase

- Removing extra whitespace

- This prevents attackers from hiding malicious content using encoding tricks.

- ❖ Step 7: Rule Action Is Applied.

  - If a rule matches, one of three actions occurs: Allow, Block or Count.

  - If a rule with Allow or Block matches, no further rules are evaluated.

  - If a rule with Count matches, evaluation continues.

  - If the request does not match any allow or block rule: The Web ACL applies its default action, usually Allow.

- ❖ Step 8: Blocked Requests Receive a 403 Response.
- ❖ - If AWS WAF blocks a request:

  - The AWS resource returns an **HTTP 403 (Forbidden)** response.

  - The response does not reveal that AWS WAF caused the block.

  - This avoids giving attackers clues about your security setup.

- ❖ Step 9: Allowed Requests Reach the Application.

  - If allowed: The request is forwarded to the backend application. (EC2 instance, container, or API service).

- ❖ Step 10: Logging and Monitoring Flow.

  - AWS WAF can Send logs via Kinesis Firehose.

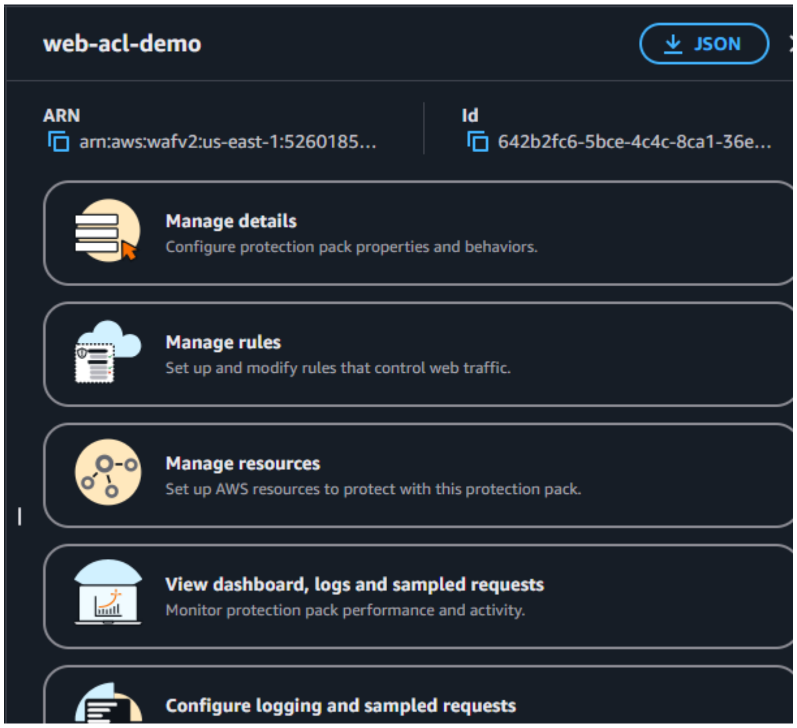
  - It stores logs in s3, redshift or OpenSearch.

  - Sampled requests (last 3 hours) are available for free.

4. **Execution Steps**

- ❖ Open AWS WAF Console
  - ➢ Go to **AWS Console**
  - ➢ Search for **WAF & Shield**
- ❖ Create a Web ACL
  - ➢ Click **Create Web ACL**
  - ➢ Name: `web-acl-demo`
  - ➢ Region: Same as ALB
  - ➢ Resource type: **Application Load Balancer**

AWS Web Application Firewall (WAF)


web-acl-demo

ARN
arn:aws:wafv2:us-east-1:5260185...

Id
642b2fc6-5bce-4c4c-8ca1-36e...

Manage details
Configure protection pack properties and behaviors.

Manage rules
Set up and modify rules that control web traffic.

Manage resources
Set up AWS resources to protect with this protection pack.

View dashboard, logs and sampled requests
Monitor protection pack performance and activity.

Configure logging and sampled requests

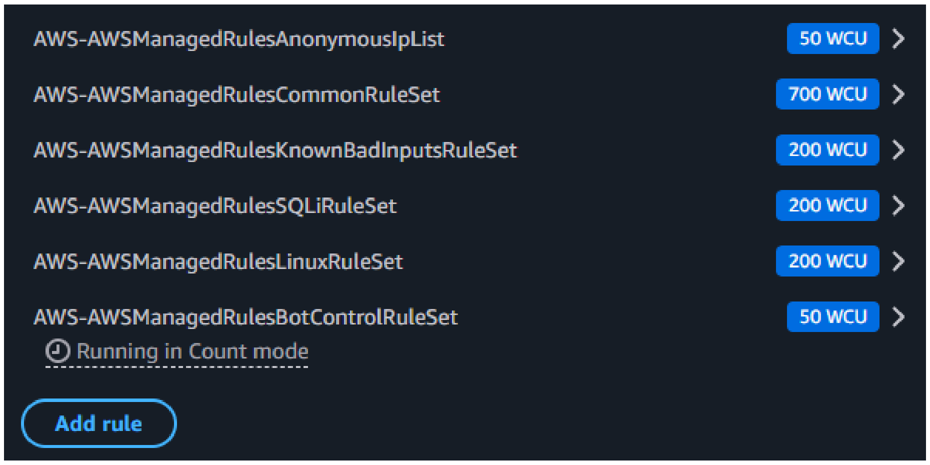
➢

❖ Choose Default Action

- ➢ Set default action to **Allow**
- ➢ (Only malicious traffic will be blocked)

❖ Add Managed Rule Groups

- ➢ AWSManagedRulesCommonRuleSet
- ➢ AWSManagedRulesSQLiRuleSet
- ➢ Purpose: Protect against common web attacks; Block SQL injection attempts.



AWS-AWSManagedRulesAnonymousIpList	50 WCU
AWS-AWSManagedRulesCommonRuleSet	700 WCU
AWS-AWSManagedRulesKnownBadInputsRuleSet	200 WCU
AWS-AWSManagedRulesSQLiRuleSet	200 WCU
AWS-AWSManagedRulesLinuxRuleSet	200 WCU
AWS-AWSManagedRulesBotControlRuleSet Running in Count mode	50 WCU

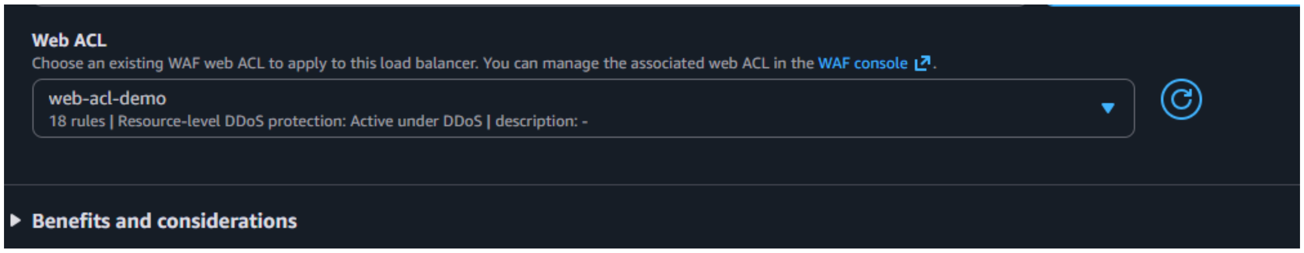
➢

❖ Configure Rule Priority

AWS Web Application Firewall (WAF)

- Rules are evaluated **top to bottom**:
- Lower number = higher priority

❖ Associate Web ACL with ALB

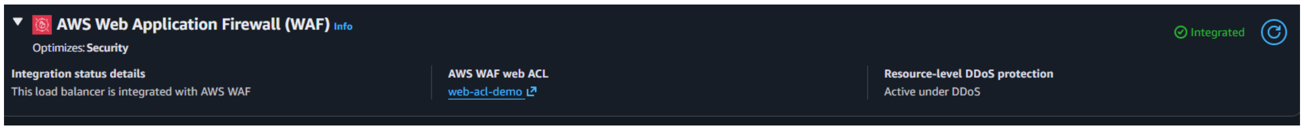
- Select your ALB
- Attach Web ACL
- 

Web ACL

Choose an existing WAF web ACL to apply to this load balancer. You can manage the associated web ACL in the WAF console.

web-acl-demo
18 rules | Resource-level DDoS protection: Active under DDoS | description: -

Benefits and considerations

- 

AWS Web Application Firewall (WAF) Info

Optimizes: Security

Integrated

Integration status details	AWS WAF web ACL	Resource-level DDoS protection
This load balancer is integrated with AWS WAF	web-acl-demo	Active under DDoS

❖ Enable Logging

- Send logs to **CloudWatch Logs**
- Helps in monitoring blocked requests