**AWS WAF & Shield - Complete Guide to Web Application Security**

**Table of Contents**

1. Introduction to AWS WAF & Shield
2. DDoS Attacks Explained
3. AWS WAF vs Shield
4. Step-by-Step Implementation
   * Step 1: Set Up EC2 Instances
   * Step 2: Configure ALB & Route 53
   * Step 3: Create AWS WAF Rules
5. Testing Security Rules
6. AWS Shield Overview
7. Real-World Use Cases
8. Cleanup
9. Interview Questions

**1. Introduction to AWS WAF & Shield**

**What Problems Do They Solve?**

* **AWS WAF (Web Application Firewall)**:
  + Blocks SQL injection, XSS (Cross-Site Scripting), and bad bots.
  + Filters traffic based on IP, country, or custom rules.
* **AWS Shield**:
  + Protects against DDoS attacks (Standard = free, Advanced = $3,000/month).

**Key Features**

| **Feature** | **AWS WAF** | **AWS Shield** |
| --- | --- | --- |
| **SQL Injection Protection** | ✅ | ❌ |
| **IP/Country Blocking** | ✅ | ❌ |
| **DDoS Mitigation** | ❌ | ✅ |
| **Cost** | Pay-per-use | Standard (Free), Advanced ($3K/month) |

**2. DDoS Attacks Explained**

**How DDoS Works**

1. **Attacker** controls a **botnet** (zombie devices infected with malware).
2. **Botnets flood** a target server (e.g., e-commerce site) with fake requests.
3. **Server overloads** → Genuine users can't access the site.

**Types of Attacks**

* **Layer 3/4 (Network)**: UDP floods, SYN floods (handled by Shield).
* **Layer 7 (Application)**: HTTP floods, SQL injection (handled by WAF).

**3. AWS WAF vs Shield**

| **Security Groups** | **AWS WAF** | **AWS Shield** |
| --- | --- | --- |
| Blocks ports/IPs | ✅ | ❌ |
| Stops SQL/XSS | ❌ | ✅ |
| Mitigates DDoS | ❌ | ✅ |
| Geo-blocking | ✅ | ❌ |

🔹 **Key Point**: Use **WAF + Shield Advanced** for full protection.

**4. Step-by-Step Implementation**

**Step 1: Set Up EC2 Instances**

1. Launch **2 EC2(Ubuntu) instances** (us-east-1a, us-east-1b) with NGINX:

Bash

*# User data script*

*#!/bin/bash*

sudo apt update

sudo apt install -y nginx

echo "Server in US-East-1" > /var/www/html/index.ngnix-debian.html

sudo systemctl restart nginx

1. Configure **Security Groups** to allow HTTP/80 traffic.

**Step 2: Configure ALB & Route 53**

1. **Create Target Group**:
   * Add both EC2 instances.
2. **Create ALB**:
   * Attach to the target group.
3. **Route 53**:
   * Add A record pointing waf.cloudvishwakarma.in to ALB DNS.

**Step 3: Create AWS WAF Rules**

1. **Block Your IP**:
   * Go to **AWS WAF → IP Sets → Create**.
   * Add your public IP (e.g., 54.123.141.50/32).
2. **Block a Country**:
   * Go to **Web ACLs → Add Rule**.
   * Select **"Originates from a country"** → Choose **India**.
3. **Priority Order**:
   * Set India rule to **Priority 0** (highest).
   * Set IP block rule to **Priority 1**.

**5. Testing Security Rules**

| **Test Case** | **Expected Result** |
| --- | --- |
| Access from **blocked IP** | HTTP 403 (Forbidden) |
| Access from **India** | HTTP 403 (Forbidden) |
| Access from **US (non-blocked)** | Website loads |

bash

*# Test from different locations*

curl -v http://waf.cloudvishwakarma.in

**6. AWS Shield Overview**

* **Shield Standard**: Free, auto-protects ALB/CloudFront.
* **Shield Advanced** ($3,000/month):
  + 24/7 DDoS response team.
  + Cost protection (credits for scaling during attacks).

**7. Real-World Use Cases**

| **Scenario** | **Solution** |
| --- | --- |
| **E-commerce site** | WAF blocks SQLi/XSS + Shield stops DDoS |
| **Geo-restricted app** | WAF blocks non-allowed countries |
| **API protection** | WAF rate-limiting + IP whitelisting |

**8. Cleanup**

1. Delete **WAF Web ACLs**.
2. Terminate **EC2 instances**.
3. Delete **ALB & Route 53 records**.

**9. Interview Questions**

**Q1: Can security groups replace WAF?**

**A**: No. Security groups work at **network layer** (ports/IPs). WAF protects against **application-layer** attacks (SQLi, XSS).

**Q2: When to use Shield Advanced?**

**A**: For mission-critical apps needing:

* Guaranteed DDoS protection.
* AWS support during attacks.

**Q3: How does WAF block SQL injection?**

**A**: It inspects HTTP requests for malicious SQL patterns (e.g., ' OR 1=1 --).