


Home / AWS / Guided Lab / Implementing AWS WAF with ALB to block SQL Injection, Geo Location and Query string

Implementing AWS WAF with ALB to block SQL Injection, Geo Location and Query string

Level: **Advanced**

[Amazon EC2](#) [Amazon Web Services](#) [Elastic Load Balancing](#) [AWS WAF](#)

Required Points

10 

Lab Duration


02:00:00


Average Start time


Less than a minute

Start Lab →

Need help?

 How to use Hands on Lab



 Troubleshooting Lab

 FAQs

[Submit Feedback](#)

[Share](#)

Lab Overview

 Cloud Network Engineer, Cloud Security Engineer
 Security, Compute, Networking

Lab Details

1. This tutorial guides you through the process of setting up an Application Load Balancer in AWS Elastic Load Balancer. This advanced load balancing solution efficiently divides incoming application traffic among two Amazon EC2 instances. Furthermore, we will establish a series of regulations to prevent access from specific geographical locations, safeguard against SQL injections, and restrict certain Query String parameters.

2. Duration: **120 minutes**

3. AWS Region: **N. Virginia (us-east-1)**

Introduction

What is AWS WAF?

- AWS WAF is a web application firewall that helps you to protect your web applications against common web exploits that might affect availability and compromise security.
- AWS WAF gives you control over how traffic reaches your applications by enabling you to create security rules that block common attack patterns like SQL injection and cross-site scripting.
- It only allows the request to reach the server based on the rules or patterns you define.
- Users create their own rules and specify the conditions that AWS WAF searches for in incoming web requests.
- The cost of WAF is only for what you use.
- The pricing is based on how many rules you deploy and how many web requests your application receives.
- For example, you can deploy AWS WAF on Amazon CloudFront, Load Balancer or API Gateways.

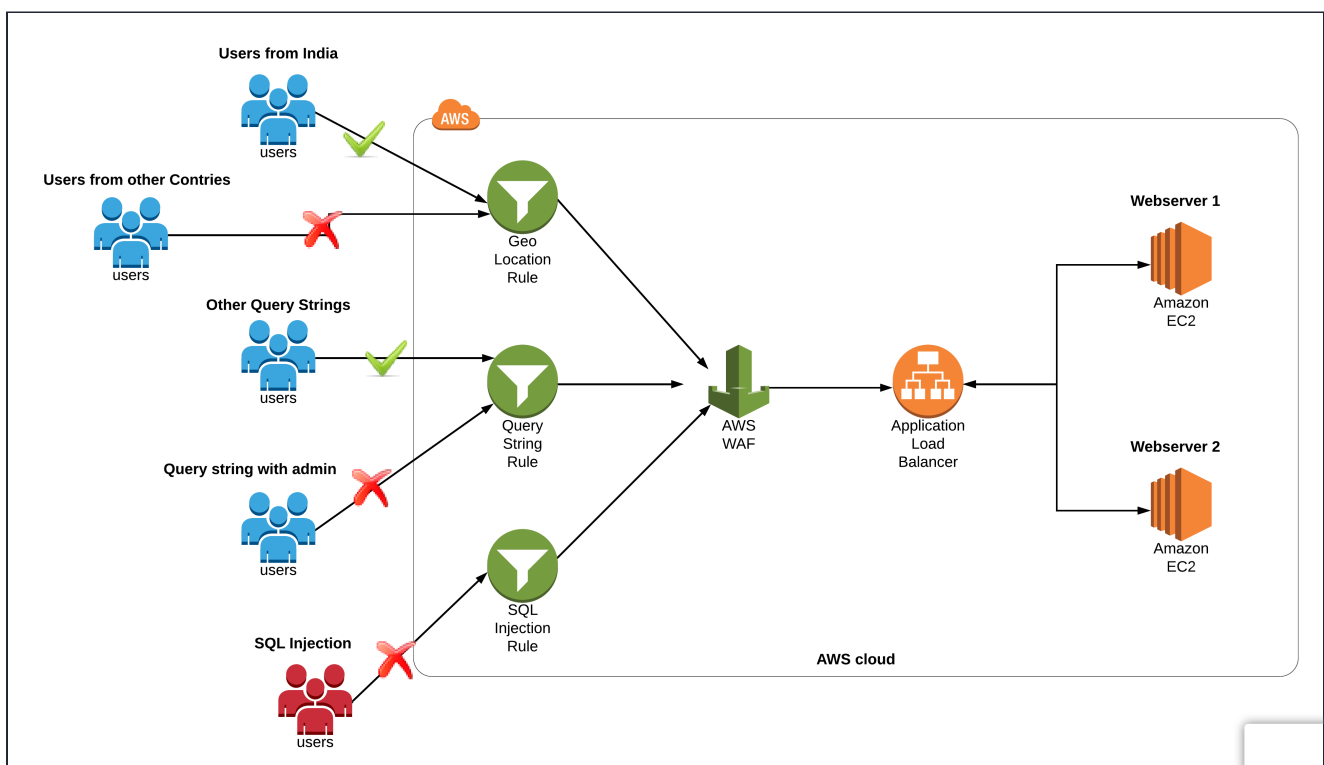
What is Elastic Load Balancing?

- ELB is a service that automatically distributes incoming application traffic and scales resources to meet traffic demands.
- It helps in adjusting capacity according to incoming application and network traffic.
- It can be enabled within a single availability zone or across multiple availability zones to maintain consistent application performance.
- ELB offers features like:
 - Detection of unhealthy EC2 instances.

- Spreading EC2 instances across healthy channels only.
- Centralized management of SSL certificates.
- Optional public key authentication.
- Support for both IPv4 and IPv6.
- ELB accepts incoming traffic from clients and routes requests to its registered targets.
- When an unhealthy target or instance is detected, ELB stops routing traffic to it and resumes only when the instance is healthy again.
- ELB monitors the health of its registered targets and ensures that the traffic is routed only to healthy instances.
- ELB's are configured to accept incoming traffic by specifying one or more **listeners**. A listener is a process that checks for connection requests.
- Listeners are configured with a protocol and port number from the client to the ELB and vice-versa i.e., back from ELB to the client.
- ELB supports the following :
 - Application Load Balancers.
 - Network Load Balancers.
 - Gateway Load Balancers.
 - Classic Load Balancers.
- Each load balancer is configured differently.
- For Application and Network Load Balancers, you register targets in target groups and route traffic to target groups.
- Gateway Load Balancers use Gateway Load Balancer endpoints to securely exchange traffic across VPC boundaries.
- For Classic Load Balancers, you register instances with the load balancer.
- AWS recommends users to work with Application Load Balancer to use multiple Availability Zones because if one availability zone fails, the load balancer can continue to route traffic to the next available one.
- We can have our load balancer be either internal or internet-facing.
- The nodes of an internet-facing load balancer have Public IP addresses, and the DNS name is publicly resolvable to the Public IP addresses of the nodes.
- Due to the point above, internet-facing load balancers can route requests from clients over the Internet.

- The nodes of an internal load balancer have only Private IP addresses, and the DNS name is publicly resolvable to the Private IP addresses of the nodes.
- Due to the point above, internal load balancers can only route requests from clients with access to the VPC for the load balancer.
- Both internet-facing and internal load balancers route requests to your targets using Private IP addresses.
- Your targets do not need Public IP addresses to receive requests from an internal or an internet-facing load balancer.
- You can create your own rules, depending on your requirements, whether to block or allow the incoming and outgoing request. You can also customise the string that appears in your web request.
- Blocking malicious requests
- You can also configure rules in AWS WAF to identify and block web requests threats like SQL injections and cross-site scripting.
- Tune your rules and monitor traffic
- AWS WAF also allows us to review our rules and customize them to prevent new attacks from reaching the server.

Architecture Diagram



Task Details

1. Sign in to AWS Management Console.
2. Launch First EC2 Instance.
3. Launch Second EC2 Instance.
4. Create a Target Group.
5. Create an Application Load Balancer.
6. Test Load Balancer DNS.
7. Create AWS WAF Web ACL.
8. Test Load Balancer DNS.
9. Validation of the lab.
10. Deleting AWS Resources.

Launching Lab Environment

1. To launch the lab environment, Click on the **Start Lab** button.
2. Please wait until the cloud environment is provisioned. It will take less than a minute to provision.
3. Once the Lab is started, you will be provided with **IAM user name, Password, Access Key, and Secret Access Key.**

Note : You can only start one lab at any given time

[About Us](#) [Subscription](#) [Instructions and Guidelines](#) [FAQ's](#) [Contact Us](#)



© 2024, Whizlabs Software Pvt. Ltd.

