

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



1h 59m 44s left



End Lab

Open Console

Validation

Lab Credentials

User Name ⓘ

Whiz_User_80425.29856740



Password ⓘ

bddd6310-4d44-49f9-bc41-138d172bfc69



Access Key ⓘ

AKIAZ5QR5CK2XMGNWWXL



Secret Key ⓘ

e1265Ro0cMzXJ9eD3asfcHLJNPrCeq3cow9lKncN



Lab Resources

No Lab Resources Found

Support Documents

No Support Documents Found

Need help?



How to use Hands on Lab



Troubleshooting Lab



FAQs

Submit Feedback

Share

Lab Overview

Lab Steps

Lab Validation



Cloud Network Engineer, Cloud Security Engineer



Security, Compute, Networking

Lab Steps

Task 1: Sign in to AWS Management Console

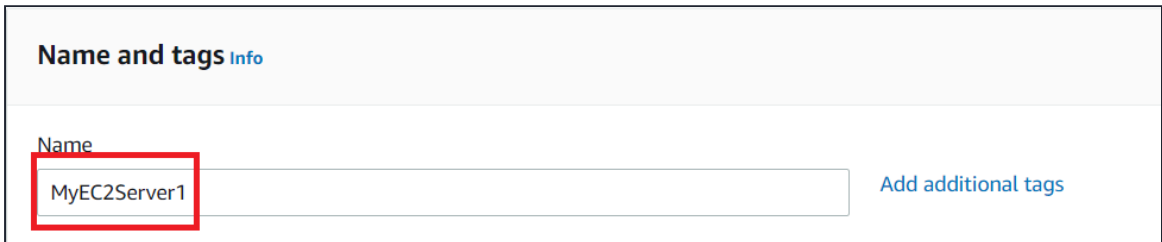
1. Click on the **Open Console** button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
 - Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
 - Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button.
3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1**.

Task 2: Launch First EC2 Instance

In this task, we are going to launch the first EC2 instance by providing the required configurations like name, AMI selection, security group , instance type and other settings.

Furthermore, we will provide the user data as well.

1. Make sure you are in the **N. Virginia(us-east-1)** Region.
2. Navigate to **EC2** by clicking on the **Services** menu in the top left, then click on **EC2** in the **Compute** section.
3. Navigate to **Instances** from the left side menu and click on **Launch Instances** button.
4. Under the **Name and tags** section :
 - Name : Enter **MyEC2Server1**



Name and tags [Info](#)

Name

MyEC2Server1

[Add additional tags](#)

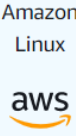
5. Under the **Application and OS Images (Amazon Machine Image)** section :
 - Select **Quick Start** tab and **Amazon Linux** under it
 - Amazon Machine Image (AMI) : select **Amazon Linux 2 AMI**
 - **Note: if there are two AMI's present for Amazon Linux 2 AMI, choose any of them.**

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)


An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images


Quick Start




Amazon Linux




Ubuntu



Windows



Red Hat



SUSE Linux

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-0cff7528ff583bf9a (64-bit (x86)) / ami-00bf5f1c358708486 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20220606.1 x86_64 HVM gp2

Architecture	AMI ID
64-bit (x86) ▼	ami-0cff7528ff583bf9a

6. Under the **Instance Type** section :

- Instance Type : Select **t2.micro**

▼ **Instance type** [Info](#)

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0116 USD per Hour
On-Demand Windows pricing: 0.0162 USD per Hour

Free tier eligible ▼

[Compare instance types](#)

7. Under the **Key Pair (login)** section :

- Click on **Create new key pair** hyperlink
- Key pair name: **MyWebserverKey**
- Key pair type: **RSA**
- Private key file format: **.pem** or **.ppk**
- Click on **Create key pair** and then select the created key pair from the drop-down.

Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

MyWebserverKey

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA
RSA encrypted private and public key pair

☐ ED25519
ED25519 encrypted private and public key pair

Private key file format

☒ .pem
For use with OpenSSH

☐ .ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn](#)

Cancel

Create key pair

8. Under the **Network Settings** section :

- Click on **Edit** button
- Auto-assign public IP: select *Enable*
- Firewall (security groups) : Select **Create a new security group**
- Security group name : Enter **MyWebserverSG**
- Description : Enter **My EC2 Security Group**
- To add **SSH**:
 - Choose Type: **SSH**
 - Source: **Anywhere** (From ALL IP addresses accessible).
- For **HTTP**, click on **Add security group rule**,
 - Choose Type: **HTTP**
 - Source: **Anywhere** (From ALL IP addresses accessible).

- For **HTTPS**, click on **Add security group rule**,
- Choose Type: **HTTPS**
- Source: **Anywhere** (From ALL IP addresses accessible).

VPC - *required* [Info](#)

vpc-abdbb6d6 (Default VPC) (default)

172.31.0.0/16

Subnet [Info](#)

No preference [Create new subnet](#)

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

Security group name - *required*

MyWebserverSG

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!\$*

Description - *required* [Info](#)

My EC2 Security Group

Inbound security groups rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0) Remove

Type Info	Protocol Info	Port range Info
ssh	TCP	22
Source type Info	Source Info	Description - <i>optional</i> Info
Anywhere	Add CIDR, prefix list or security group 0.0.0.0/0	e.g. SSH for admin desktop

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Add security group rule

9. Under the **Advanced details** section :

- Under the **User data**: copy and paste the following script to create an HTML page served by an Apache HTTPD web server.







```
#!/bin/bash
```



```
systemctl start httpd
systemctl enable httpd
echo "<html><h1> Welcome to Whizlabs Server 1 </h1><html>" >>
/var/www/html/index.html
```

10. Keep everything else as default and click on the **Launch instance** button.

11. **Launch Status:** Your instance is now launching, Navigate to **Instances** page from the left menu and wait until the status of the EC2 Instance changes to **running**.

Instances (1/1) Info								  Instance state ▼	Actions ▼	Launch instances ▼
<input type="text" value="Filter instances"/>								< 1 > 		
<input checked="" type="checkbox"/>	Name ▼	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status	Availability Zone			
<input checked="" type="checkbox"/>	MyEC2Server1	i-0e216b97173131875	 Running 	t2.micro	 Initializing	No alarms +	us-east-1a			

Task 3: Launch Second EC2 Instances

In this task, we are going to launch the second EC2 instance by providing the required configurations like name, AMI selection, security group, instance type and other settings. Furthermore, we will provide the user data as well.

1. Now again click on **Launch Instances** button.

2. Under the **Name and tags** section :

- Name : Enter **MyEC2Server2**

Name and tags [Info](#)

Name

MyEC2Server2

Add additional tags

3. Under the **Application and OS Images (Amazon Machine Image)** section :

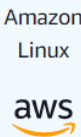
- Select **Quick Start** tab and **Amazon Linux** under it
- Amazon Machine Image (AMI) : select **Amazon Linux 2 AMI**
- Note: if there are two AMI's present for Amazon Linux 2 AMI, choose any of them.**

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

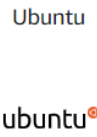
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images


Quick Start



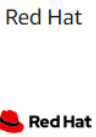
Amazon Linux



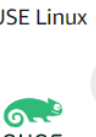
Ubuntu



Windows



Red Hat



SUSE Linux

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-0cff7528ff583bf9a (64-bit (x86)) / ami-00bf5f1c358708486 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20220606.1 x86_64 HVM gp2

Architecture	AMI ID
64-bit (x86) ▼	ami-0cff7528ff583bf9a

4. Under the **Instance Type** section :

- Instance Type : Select **t2.micro**

▼ **Instance type** [Info](#)

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0116 USD per Hour
On-Demand Windows pricing: 0.0162 USD per Hour

Free tier eligible ▼

[Compare instance types](#)

5. Under the **Key Pair (login)** section :

- Select **MyWebserverKey** from the list.

6. Under the **Network Settings** section :

- Click on **Edit** button
- Auto-assign public IP: select **Enable**
- Firewall (security groups) : Select **Select existing security group**

- Common security groups : Select Security group with name **MyWebserverSG**

7. Under the **Advanced details** section :

- Under the **User data**: copy and paste the following script to create an HTML page served by Apache httpd web server:

```
#!/bin/bash
sudo su
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "<html><h1> Welcome to Whizlabs Server 2 </h1><html>" >>
/var/www/html/index.html
```



8. Keep everything else as default and then click on the **Launch Instance** button.

9. Your instances are now launching. Navigate to the EC2 instance page and wait until the status changes to the **Running**. It will usually take 1-2 minutes.

Instances (2) Info

Refresh

Connect

Instance state ▾

Actions ▾

Launch instances

▾

Q Filter instances

< 1 > ⚙

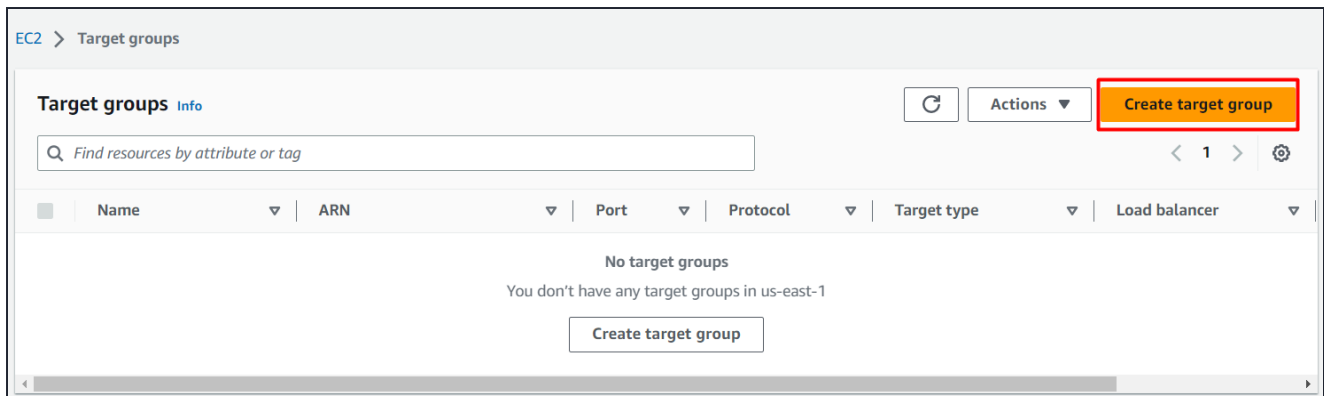
<input type="checkbox"/>	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	MyEC2Server1	i-0e216b97173131875	<div>✔ Running</div> <div>🔍🔍</div>	t2.micro	<div>✔ 2/2 checks passed</div>	No alarms +	us-east-1a
<input type="checkbox"/>	MyEC2Server2	i-0011e80f7ce3cbcd	<div>✔ Running</div> <div>🔍🔍</div>	t2.micro	<div>✔ 2/2 checks passed</div>	No alarms +	us-east-1a

Task 4: Create a Target Group

In this task, we are going to create a target group for the load balancer and will add the target instances so that the load balancer can distribute the traffic among these instances.

1. In the EC2 console, navigate to **Target Groups** in the left-side panel under **Load Balancer** in the **Load Balancing** section.
2. Click on **Create target group** button on the top right corner.





3. Basic configuration:

- Choose a target type : Select **Instances**
- Target group name : Enter **MyWAFTargetGroup**
- Protocol : Select **HTTP**
- Port : Enter **80**

4. Health Checks:

- Health check protocol : Select **HTTP**

Healthy threshold
The number of consecutive health checks successes required before considering an unhealthy target healthy.

2-10

Unhealthy threshold
The number of consecutive health check failures required before considering a target unhealthy.

2-10

Timeout
The amount of time, in seconds, during which no response means a failed health check.

seconds 2-120

Interval
The approximate amount of time between health checks of an individual target

seconds 5-300

5. Leave everything as default and click on **Next** button.

6. Register targets:

- Select the two instances we have created i.e **MyEC2Server1** and **MyEC2Server2**.

- Click on **Include as pending below** and scroll down.

Available instances (2/2)

Q Filter resources by property or value

<input checked="" type="checkbox"/>	Instance ID	Name	Status	Security groups
<input checked="" type="checkbox"/>	i-048434eea99d77038	MyEC2Server1	Running	MyWebserverSG
<input checked="" type="checkbox"/>	i-02ead111eb23b1d54	MyEC2Server2	Running	MyWebserverSG

2 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.

80

1-65535 (separate multiple ports with commas)

Include as pending below

7. Review targets:

- Review the targets and click on **Create target group** button.

Review targets

Targets (2) Remove all pending

All Q Filter resources by property or value

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone	Subnet
X	Pending	i-048434eea99d77038	MyEC2Server1	80	Running	MyWebserverSG	us-east-1c	subnet
X	Pending	i-02ead111eb23b1d54	MyEC2Server2	80	Running	MyWebserverSG	us-east-1d	subnet

2 pending

Cancel Previous **Create target group**

8. Your Target group has been successfully created.

Successfully created target group: MyWAFTargetGroup

EC2 > Target groups

Target groups (1) Info

Find resources by attribute or tag

	Name	ARN	Port	Protocol	Target type	Load balancer
<input type="checkbox"/>	MyWAFTargetGroup	arn:aws:elasticloadbalanci...	80	HTTP	Instance	None associated

Task 5: Create an Application Load Balancer

In this task, we are going to create an Application Load balancer by providing the required configurations like name, target group etc.

1. In the EC2 console, navigate to **Load Balancers** in the left-side panel under **Load Balancing**.
2. Click on **Create Load Balancer** at the top-left to create a new load balancer for our web servers.
3. On the next screen, choose **Application Load Balancer** since we are testing the high availability of the web application and click on **Create** button.
4. Basic configuration:
 - Load balancer name: Enter **MyWAFLoadBalancer**
 - Scheme: Select **Internet-facing**
 - Ip address type: Choose **Ipv4**

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

MyWAFLoadBalancer

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)
Scheme can't be changed after the load balancer is created.

☒ **Internet-facing**
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

☐ **Internal**
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [Info](#)
Select the type of IP addresses that your subnets use.

☒ **IPv4**
Recommended for internal load balancers.

☐ **Dualstack**
Includes IPv4 and IPv6 addresses.

5. Network mapping:
 - VPC : Select **Default**
 - Mappings : Check **All Availability Zones**

6. Security groups:

- Security groups : Select **an existing security group** i.e **MyWebserverSG** from the drop down menu.

Security groups [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups

default sg-e6f0d3fe X
VPC: vpc-2591f458

7. Listeners and routing:

- Protocol : Select **HTTP**
- Port : Enter **80**
- Default action : Select **MyWAFTargetGroup** from the drop down menu

▼ Listener HTTP:80 Remove

Protocol Port

HTTP : 80
1-65535

Default action [Info](#)

Forward to MyWAFTargetGroup
Target type: Instance, IPv4

HTTP Refresh

[Create target group](#)

8. Leave everything as default and click on **Create load balancer** button.

9. You have successfully created Application Load Balancer.

Task 6: Test Load Balancer DNS

In this task, we will test the working of load balancer by copying the DNS to the browser and find out whether it is able to distribute the traffic or not.

1. Now navigate to the **Target Groups** from the left side menu under **Load balancing**.
2. Click on the **MyWAFTargetGroup** Target group name.
3. Now select the **Targets** tab and **wait till both the targets become healthy (Important)**.

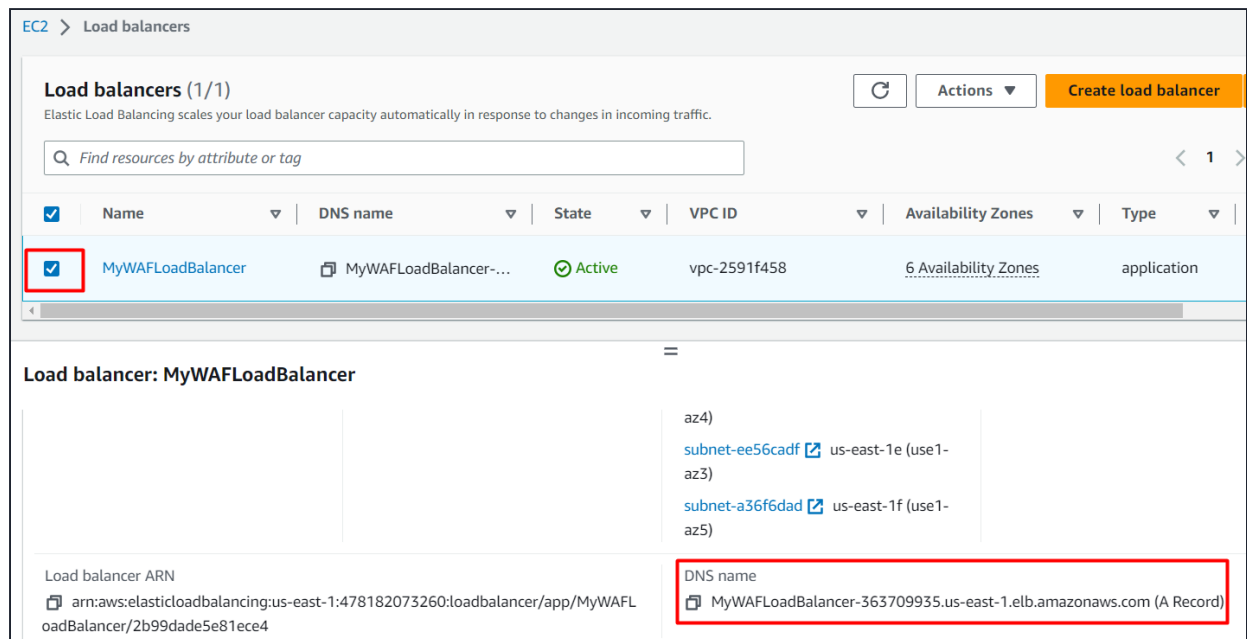
Registered targets (2) Refresh Deregister Register targets

Filter resources by property or value

	Instance ID	Name	Port	Zone	Health status	Health status details
<input type="checkbox"/>	i-0011e80f7ce3cbcd	MyEC2Server2	80	us-east-1a	healthy	
<input type="checkbox"/>	i-068cc6266f84a868e	MyEC2Server1	80	us-east-1a	healthy	

4. Now again navigate to **Load Balancers** from the left side menu under **Load balancing**.

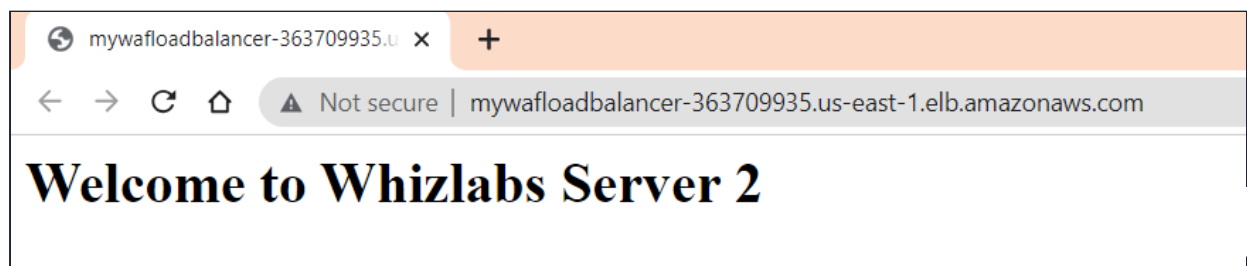
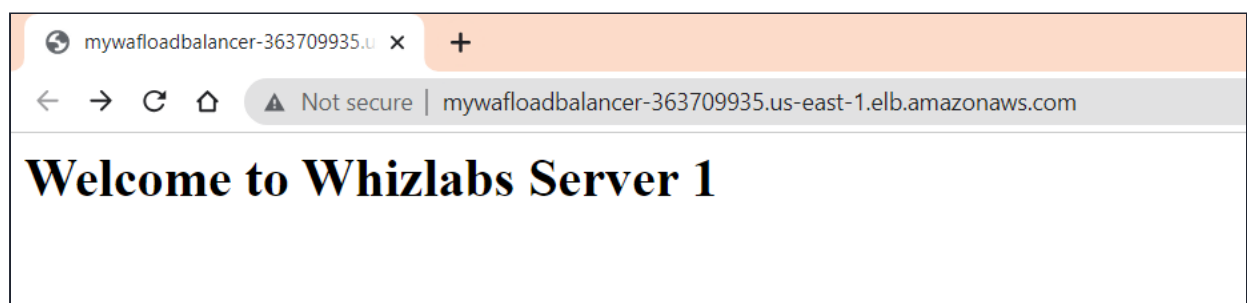
5. Select the **MyWAFLoadBalancer** Load Balancer and copy the **DNS name** under **Description** tab.



6. Copy the **DNS name** of the ELB and enter the address in the **browser**.

- **DNS Example: MyWAFLoadBalancer-2020171322.us-east-1.elb.amazonaws.com**

7. You should see the **index.html** page content of Web Server 1 or Web Server 2

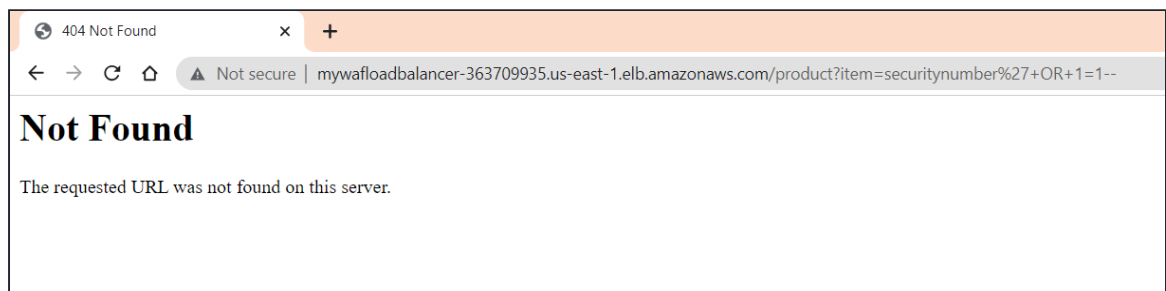


8. Now **Refresh** the page a **few times**. You will observe that the index pages change each time you refresh.

- **Note: The ELB will equally divide the incoming traffic to both servers in a Round Robin manner.**

9. Test SQL Injection :

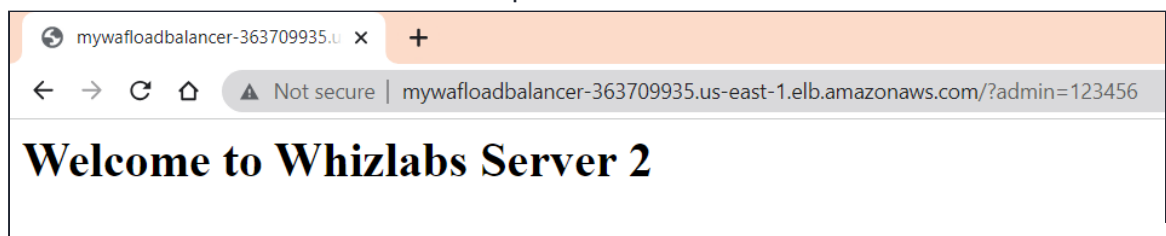
- Along with the ELB DNS add the following URL parameter: ***/product?item=securitynumber'+OR+1=1--***
- Syntax : ***http://<ELB DNS>/product?item=securitynumber'+OR+1=1--***
- Example : ***MyWAFLoadBalancer-2020171322.us-east-1.elb.amazonaws.com/product?item=securitynumber'+OR+1=1--***
- You will be able to see the below output.



- Here the **SQL Injection went inside the server** and since we only have an index page, the server doesn't know how to solve the URL that is why you got **Not Found** page.

10. Test Query String Parameter :

- Along with the ELB DNS add the following URL parameter: ***/?admin=123456***
- Syntax : ***http://<ELB DNS>/?admin=123456***
- Example : ***MyWAFLoadBalancer-2020171322.us-east-1.elb.amazonaws.com/?admin=123456***
- You will be able to see the below output.

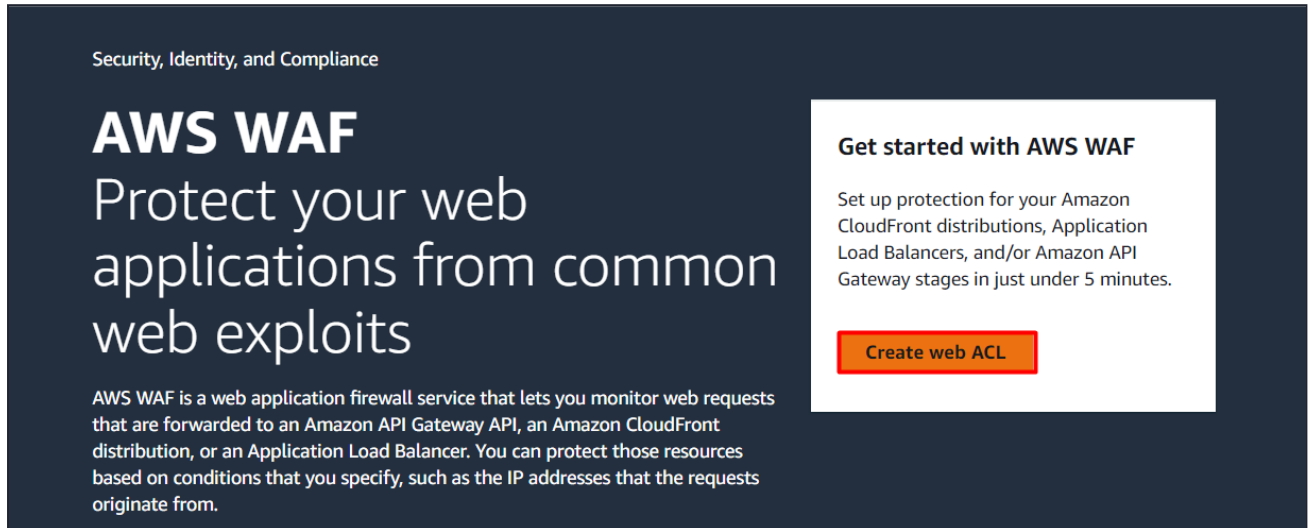


- Here also the **Query string went inside the server** and the server always passes the query string inside and it is resolved by the code that you write. Here the query string is passed and there is no code to resolve this but it won't throw any error it just becomes an unused value. so you got a response back.

Task 7: Create AWS WAF Web ACL

In this task , we are going to create an AWS WAF Web ACL where we will add some customized rules for location restriction, query strings and

1. Navigate to **WAF** by clicking on the **Services** menu in the top, then click on **WAF & Shield** in the **Security, Identity & Compliance** section.
2. On the left side menu, select **Web ACL's** and then click on **Create web ACL** button.



3. Describe web ACL and associate it to AWS resources :

- Name : Enter **MyWAFWebAcl**
- Description : Enter **WAF for SQL Injection, Geo location and Query String parameters**
- CloudWatch metric name : Automatically selects the WAF name, so no changes required.
- Resource type : Select **Regional resources**
- Region : Select **US East (N.Virginia)** from the dropdown.
- **Associated AWS resources :**
 - Click on the **Add AWS resources** button.
 - Resource type : Select **Application Load Balancer**
 - Select **MyWAFLoadBalancer** Load balancer from the list.

Add AWS resources [X]

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

☐ Amazon API Gateway ☒ Application Load Balancer

☐ AWS AppSync ☐ Amazon Cognito User Pools

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

Find AWS resources to associate < 1 > [Settings]

<input checked="" type="checkbox"/>	Name
<input checked="" type="checkbox"/>	MyWAFLoadBalancer

Cancel Add

- Now click on the **Add** button.
- Click on the **Next** button.

4. Add rules and rule groups :

- Under **Rules**, click on **Add rules** and then select **Add my own rules and rule groups**.
 - Rule type : Select **Rule builder**
 - Name : Enter **GeoLocationRestriction**
 - Type : Select **Regular type**
 - If a request : Select **Doesn't match the statement (NOT)**
 - Inspect : Select **Originates from a country in**
 - Country codes : Select **<Your Country>** In this example we select **India-IN**
 - **Note** : You can also select multiple countries also.
 - IP address to use to determine the country of origin : Select **Source IP address**

If a request doesn't match the statement (NOT)

Statement

Inspect

Originates from a country in

Country codes

Choose country codes

India - IN ✕

IP address to use to determine the country of origin
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

- Under **Then : Action** Select **Block**.
- Click on **Add rule**.
- Here we are only allowing requests to come from India and all the requests that come from other countries will be blocked.
- Under **Rules**, click on **Add rules** and then select **Add my own rules and rule groups**.
 - Rule type : Select **Rule builder**
 - Name : Enter **QueryStringRestriction**
 - Type : Select **Regular type**
 - If a request : Select **matches the statement**
 - Inspect : Select **Query string**
 - Match type : Select **Contains string**
 - String to match : Enter **admin**
 - Text transformation : Leave as default.
 - Under **Then : Action** Select **Block**.
 - Click on **Add rules**.
- Anytime in the request URL contains a query string as **admin** WAF will block that request.
- Under **Rules**, click on **Add rules** and then select **Add managed rule groups**.

- It will take a few minutes to load the page. It lists all the rules which are managed by AWS.
- Click on **AWS managed rule groups**.
- Scroll down to **SQL database** and enable the corresponding **Add to web ACL** button.

SQL database
Contains rules that allow you to block request patterns associated with exploitation of SQL databases, like SQL injection attacks. This can help prevent remote injection of unauthorized queries.

200

 Add to web ACL

Edit

- Scroll down to the end and click on **Add rules** button.
- Now you have 3 rules added.

<input type="checkbox"/>	Name	Capacity	Action
<input type="checkbox"/>	GeoLocationRestriction	1	Block
<input type="checkbox"/>	QueryStringRestriction	10	Block
<input type="checkbox"/>	AWS-AWSManagedRulesSQLiRuleSet	200	Use rule actions

- Under **Default web ACL action for requests that don't match any rules, Default action** Select **Allow**.
- Click on the **Next** button.

5. Set rule priority :

- No changes required, leave as default.
- **Note** : You can move the rules based on your priority.
- Click on the **Next** button.

6. Configure metrics :

- Leave it as default.
- Click on the **Next** button.

7. Review and create web ACL :

- Review the configuration done, scroll to the end and click on **Create web ACL** button.

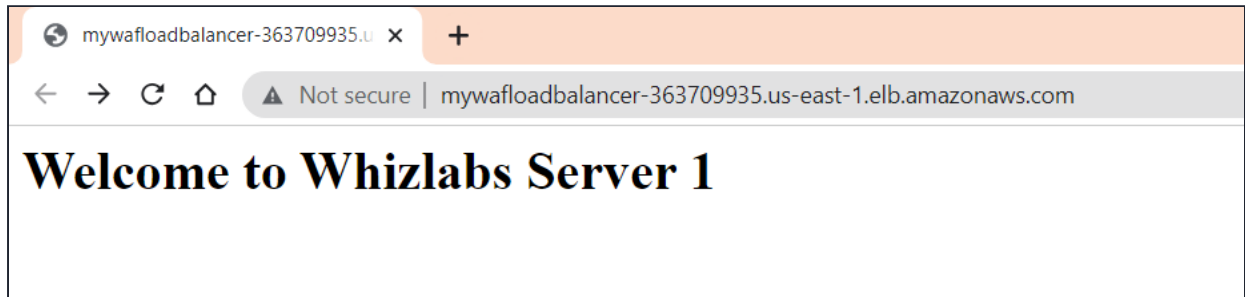
8. It will take a few seconds to create the Web ACL, so wait till its completed.

Task 8: Test Load Balancer DNS

1. Now again navigate to **Load Balancers** from the left side menu under **Load balancing**.
2. Select the **MyWAFLoadBalancer** Load Balancer and copy the **DNS name** under **Description** tab.
3. Copy the **DNS name** of the ELB and enter the address in the **browser**.

- **DNS Example: MyWAFLoadBalancer-2020171322.us-east-1.elb.amazonaws.com**

4. You should see the **index.html** page content of Web Server 1 or Web Server 2.

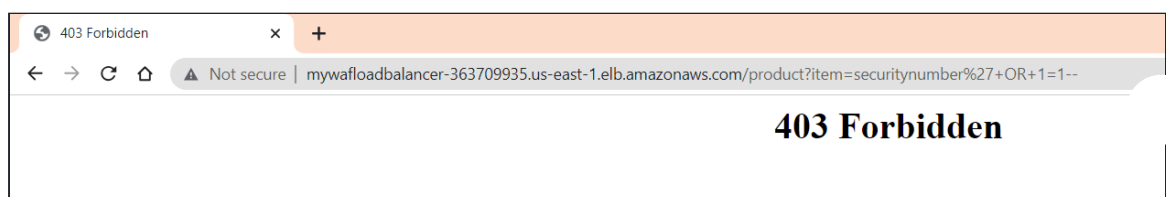


5. Now **Refresh** the page **a few times**. You will observe that the index pages change each time you refresh.

- **Note: The ELB will equally divide the incoming traffic to both servers in a Round Robin manner.**

6. Test **SQL Injection** :

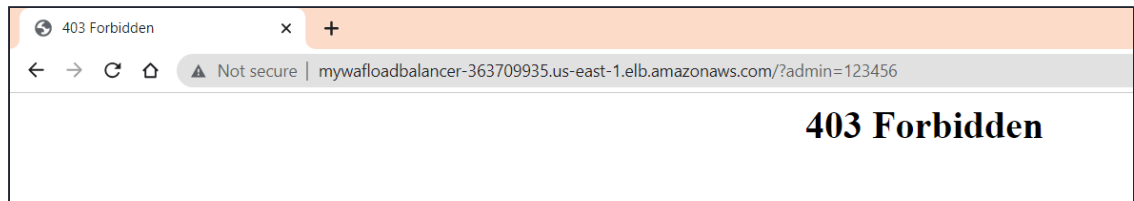
- Along with the ELB DNS add the following URL parameter: **/product?item=securitynumber'+OR+1=1--**
- Syntax : **http://<ELB DNS>/product?item=securitynumber'+OR+1=1--**
- Example : **MyWAFLoadBalancer-2020171322.us-east-1.elb.amazonaws.com/product?item=securitynumber'+OR+1=1--**
- You will be able to see the below output.



- Here the **SQL Injection is blocked by WAF before it goes inside the server.**

7. Test Query String Parameter :

- Along with the ELB DNS add the following URL parameter: ***/?admin=123456***
- Syntax : ***http://<ELB DNS>/?admin=123456***
- Example : ***MyWAFLoadBalancer-2020171322.us-east-1.elb.amazonaws.com/?admin=123456***
- You will be able to see the below output.



- Here also the **Query string which contains admin is blocked by WAF before it could go inside the server.**

Do you know?

WAF can offer protection against Distributed Denial of Service (DDoS) attacks by analyzing traffic patterns, detecting abnormal behavior, and mitigating the impact of such attacks.

Task 9: Validation Test

1. Once the lab steps are completed, please click on the **Validation** button on the left side panel.
2. This will validate the resources in the AWS account and displays whether you have completed this lab successfully or not.
3. Sample output :

The screenshot shows the Whizlabs lab interface. At the top, there's a navigation bar with 'WHIZLABS' and links for 'Lab Library', 'Cloud Sandboxes', and 'My Activity'. Below this, a blue header contains the lab title 'Implementing AWS WAF with ALB to block SQL Injection, Geo Location and Query string' and its level 'Advanced'. A row of tags includes 'Amazon EC2', 'Amazon Web Services', 'Elastic Load Balancing', and 'AWS WAF'. The main content area has tabs for 'Lab Overview', 'Lab Steps', and 'Lab Validation'. Under 'Lab Validation', there's a section 'Check your Validation' with instructions on what to do if checks fail or pass, and a green 'Validate My Lab' button. Below that, a task 'Launch two Amazon EC2 Instance' is listed with a description to check if two Amazon Linux 2 instances are created or not.

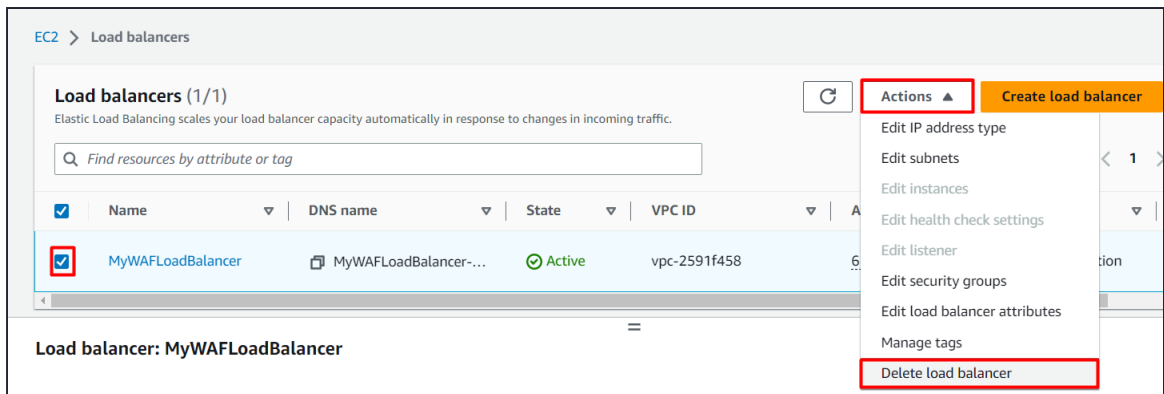
Task 10: Delete AWS Resources

10.1 Deleting an EC2 Instance

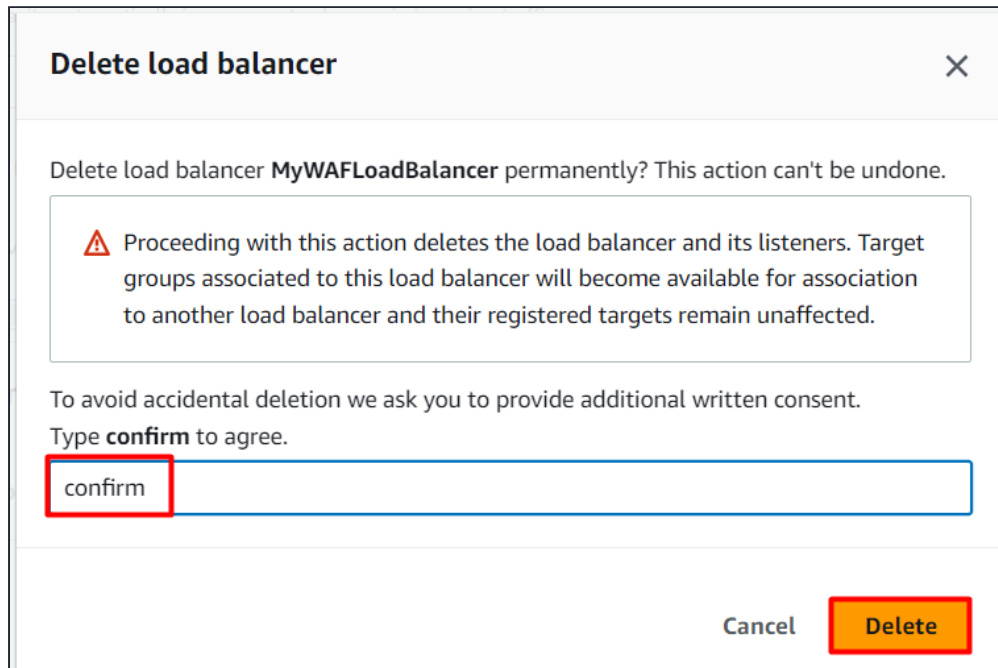
- Make sure you are in the **US East (N. Virginia) us east-1** Region.
- Navigate to **EC2** by clicking on the **Services** menu in the top, then click on **EC2** under **Compute** section.
- Now select the EC2 instance that you have created, click on the **Instance State** and click on the **Terminate** option.
- Click on **Yes, Terminate** button and your EC2 will start terminating.

10.2 Deleting Elastic LoadBalancer and Target Group

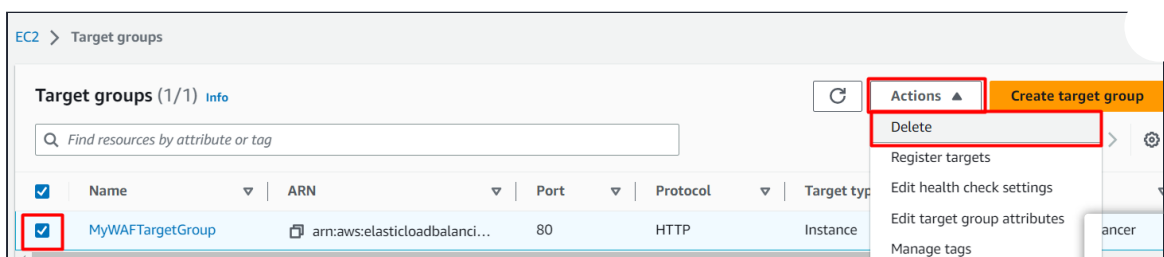
- In the EC2 console, navigate to **Load Balancer** in the left-side paneol.
- **MyWAFLoadBalancer** will be listed here.
- To **delete** the load balancer, need to perform the following actions:
 - **Select** the load balancer,
 - Click on the **Actions** button,
 - Select the **Delete** option.



- Confirm by typing **confirm** and then click on **Delete** button when a pop-up is shown.



- **MyWAFLoadBalancer** be deleted immediately.
- In the EC2 console, navigate to **Target Groups** in the left-side panel.
- **MyWAFTargetGroup** will be listed here.
- To delete the **target group**, need to perform the following actions:
 - **Select** the target group,
 - Click on the **Actions** button,
 - Select the **Delete** option.



- Now click on the **Yes, delete** button to confirm deletion.

- **MyWAFTargetGroup** will be deleted immediately.

10.3 Deleting Web ACL

- Navigate to **WAF** by clicking on the **Services** menu in the top, then click on **WAF & Shield** in the **Security, Identity & Compliance** section.
- On the left side menu, select **Web ACLs** and then click on the Web ACL name that you created, **MyWAFWebAcl**.
- Select **Associated AWS resources** tab, select the application load balancer and click on **Disassociate** button.
- In the textbox enter **remove** and click on **Disassociate** button.
- On the left side menu, select **Web ACLs** and then select the radio button of the Web ACL that you created, **MyWAFWebAcl**.
- Click on the **Delete** button, In the textbox enter **delete** and click on **Delete** button.
- Now the WAF will be successfully deleted.

Completion and Conclusion

1. You have successfully launched First EC2 Instance.
2. You have successfully launched Second EC2 Instance.
3. You have successfully created an Application Load Balancer and Target Group.
4. You have successfully tested Load Balancer DNS.
5. You have successfully created AWS WAF Web ACL.
6. You have successfully tested Load Balancer DNS.

End Lab

1. Sign out of the AWS Account.
2. You have successfully completed the lab.
3. Once you have completed the steps, click on **End Lab** from your whizlabs lab console and wait till the process gets completed.

