**Introduction**

**WAF (web application firewall)**

1. AWS WAF is a web application firewall that helps you protect your web applications against common web exploits that might affect availability and compromise security.
2. 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.
3. It only allows the request to reach the server based on the rules or patterns you define.
4. Users create their own rules and specify the conditions that AWS WAF searches for in incoming web requests.
5. The cost of WAF is only for what you use.
6. The pricing is based on how many rules you deploy and how many web requests your application receives.
7. For example, you can deploy AWS WAF on Amazon CloudFront with an Application Load Balancer in front of your web servers or servers running on EC2.

**Features of WAF**

Web traffic filtering using custom rules

You can create your own rules, depending on your requirements, whether to block or allow incoming and outgoing requests. You can also customize the string that appears in your web request.

Blocking malicious requests

You can also configure rules in AWS WAF to identify and block web request 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.

**Lab Description**

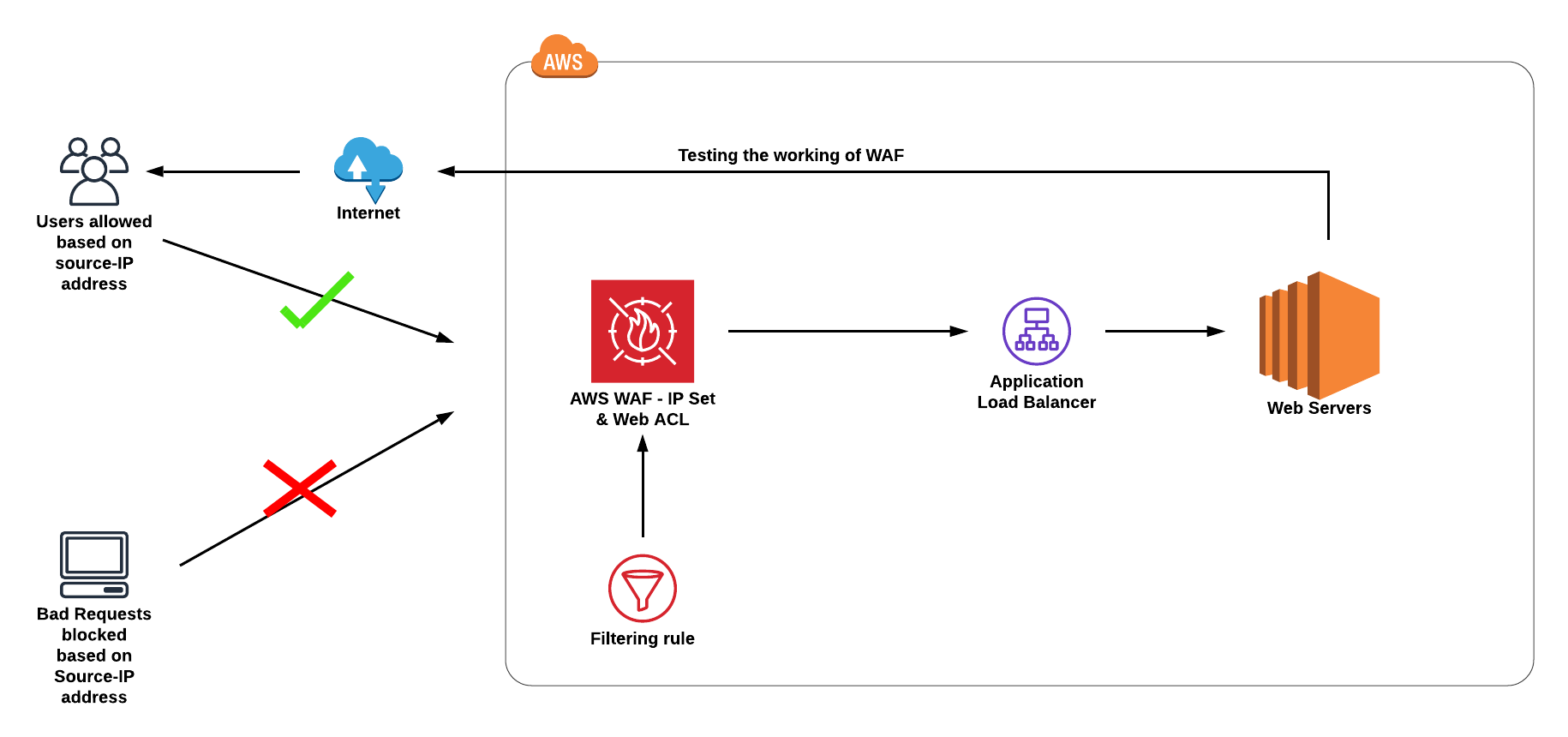
**Application Load Balancer (ALB)**

* **Load Balancer** is a **service** that **allows you to distribute the incoming application or network traffic across multiple targets**, such as Amazon **EC2** instances, containers, and IP addresses, in multiple Availability Zones.
* ALB is used to route the HTTP and HTTPS traffic across the targets based on the rules attached to the target group.
* **Rules** determine what action is taken when a rule matches a client's request.
* The **target group** is used to route requests across registered **targets** as part of an action rule. Target groups consist of a protocol and target port. We can also configure health checks to monitor the status of the target group. A single  ALB can route traffic to multiple target groups.
* **Targets** consist of EC2 instances that are registered with the ALB as part of a target group.

**Web servers**

* Two web servers are launched in the Private subnet to handle the web request.
* The request to web servers is shared using the ALB.
* Web servers are attached to the ALB Target group.
* Servers are pre-installed with HTTPD on both servers and have the test pages **RESPONSE COMING FROM SERVER 1** and **RESPONSE COMING FROM SERVER 2**respectively.
* They are attached to a security group via port 80 that allows the web traffic coming from ALB.

**Architecture Diagram**



# ****Lab Steps****

## ****Task 1: Sign in to the AWS Management Console****

1. Click on the **open console** button, and you will get redirected to the 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 the 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.**
4. Select Maybe later in the New AWS Console Home page pop-up

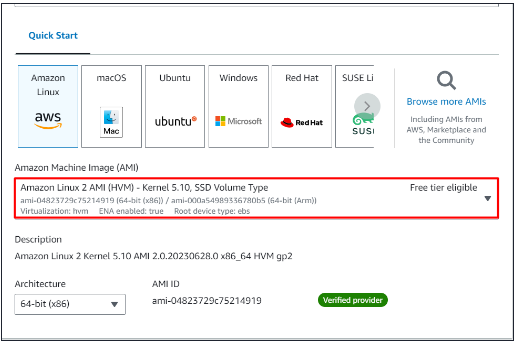
## ****Task 2: Creating a Security group for the Load balancer****

1. Navigate to the EC2 Dashboard and scroll down to **Security Groups**. In the left menu, click on **Create security group.**
2. **Configure the security** group as follows:

* Security group name: Enter ***LoadBalancer-SG***
* Description: Enter ***Security group for the Load balancer***
* VPC: **Leave as default**
* In **Inbound rules**, Click on **Add Rule** and add the port as follows.
* Type:Select **HTTP**
* Protocol: **TCP**
* Port range**: 80**
* Source: Select **Custom**, and enter ***0.0.0.0/0***
* Once you provide the above details, click on Create and the security group for the load balancer will be created.

## ****Task 3: Steps to create the Web-servers****

1. Make sure you are in the **US East (N. Virginia)**Region.
2. Navigate to **Instances** on the left panel and click on **Launch instance.**
3. Name: Enter **webserver-A**
4. **For Amazon Machine Image (AMI):** Search for **Amazon Linux 2 AMI** in the search box and click on the **select** button



1. Instance Type: Select **t2.micro**

****

* 1. **For Key pair:** Select **Create a new key pair** Button
  + Key pair name: **WhizKey**
  + Key pair type: **RSA**
  + Private key file format: **.pem**

1. Select the **Create key pair** Button.
2. In Network Settings Click on the **Edit button**:

* Auto-assign public IP: **Enable**
* Choose **Create security group**
  + Name:  Enter ***webserver-SG***
  + Description: Enter ***security group for the webserver***
* To add **HTTP** Click on **Add security group rule**
  + Choose Type**:** Select **HTTP**
  + Source Type: **Custom**
  + Source: Choose **LoadBalancer-SG**
* To add **SSH**
  + Choose Type: Select **SSH**
  + Source: Choose **Anywhere**

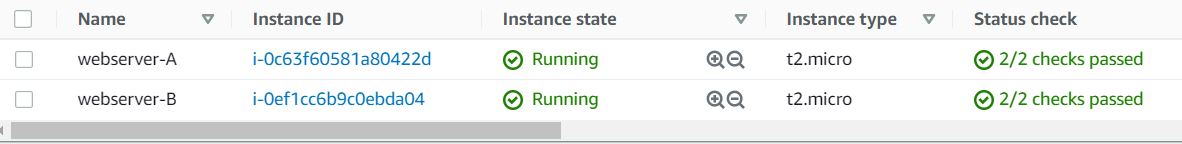
1. Click on **Advanced Details.**
2. Under the **User data** section, enter the following script to create an HTML page served by an Apache HTTPD web server.



1. Click on **Launch Instances**.
2. After a few minutes, you will see a new instance named **webserver-A** running.
3. Repeat the above steps to create **Webserver-B** by selecting the existing security group **webserver-SG** providing the following details.
4. Kindly select the same key-pair (as in you have created for previous instance (**Webserver-A)**to **Webserver-B**instance.



1. Navigate to the EC2 Dashboard to find the two instances (webserver-A and webserver-B) running.

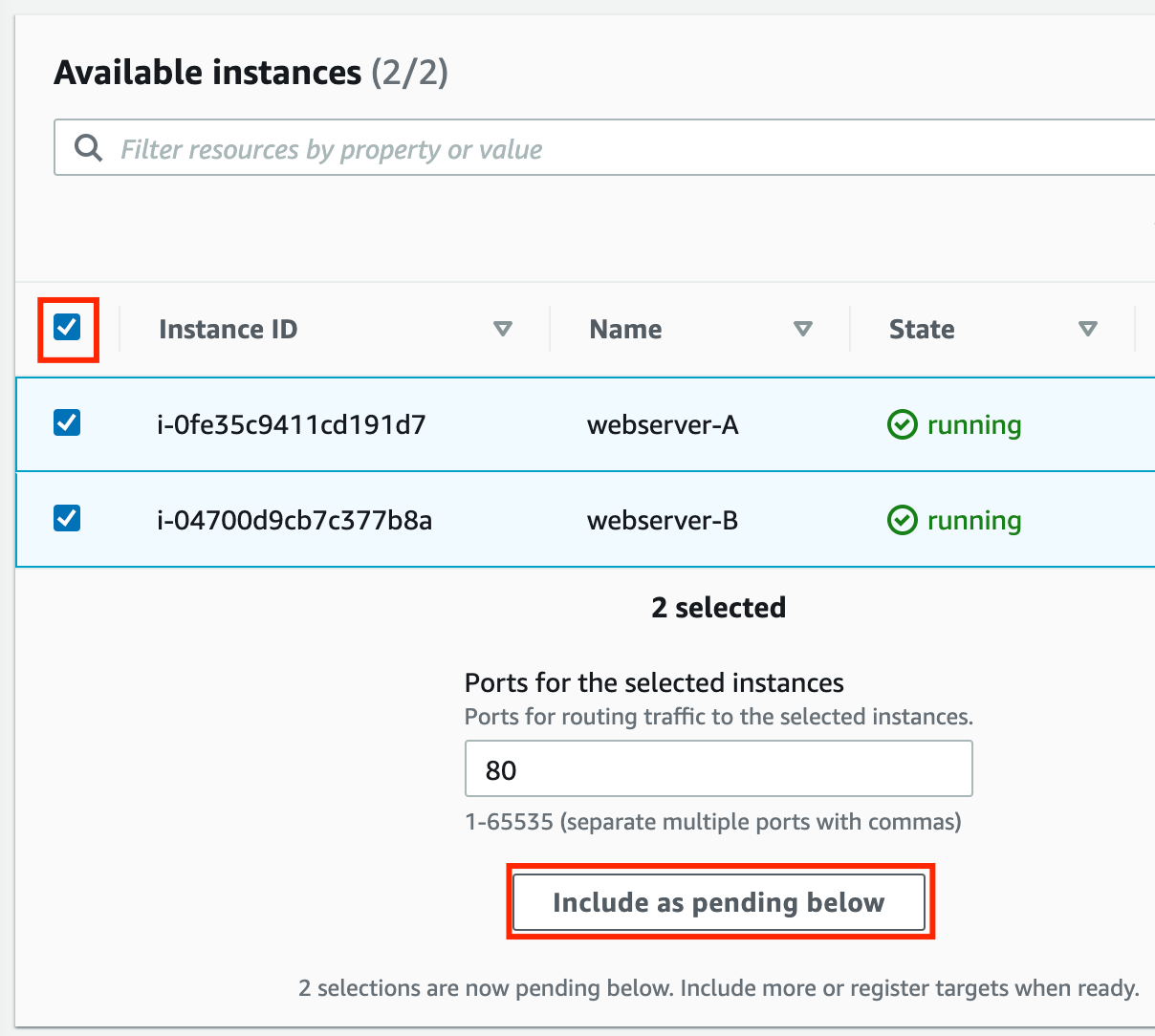


**Task 4: Creating a Load balancer**

1. In the **EC2** Console, Navigate to **Target Groups**, present in the left panel under **Load Balancing**.
2. Click on the **Create target group.**
3. For **Step 1, Specify group details**

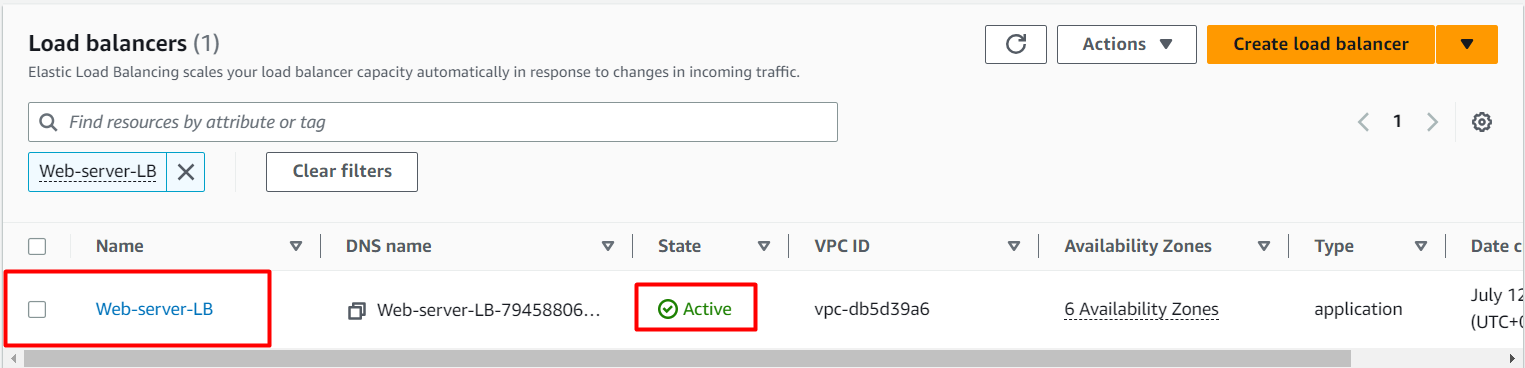
* Under Basic configurations,
  + Choose a target group:  Choose **Instances**
  + Target group name:  Enter ***web-server-TG***
* **Keep all the settings as default.**
* Health check protocol: HTTP
* Health check path: Enter ***/index.html***
* Scroll to the end of the page and click on the **Next** button.

1. For **Step 2, Register targets**
   * Select both instances and click on the **Include as pending below** button.



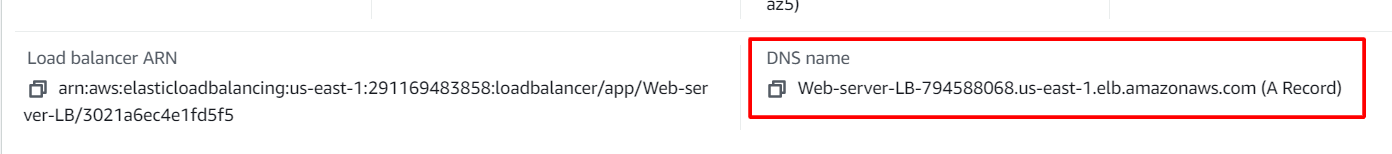
* + Instances will be present in the Review targets part, having health status as **Pending**.
  + Click on the **Create target group** button.

1. **The Target group is now created.**
2. In the EC2 console, navigate to **Load balancers** in the left-side panel.
3. Click on **create load balancer**at the top-left to create a new load balancer for our web servers.
4. **Select Load Balancer Type**: Under the **Application load balancer**, click on the **Create** button.
5. To create an Application load balancer, **configure the load balancer** as below
   * For the **Basic configuration** section,
     + Name: Enter ***Web-server-LB***
     + Scheme: Select**Internet-facing**
     + IP address type: Choose **IPv4**
   * For the **Network mapping** section:
     + VPC: Select **Default**
     + Mappings:**Select all the Availability zones present**
   * For the Security groups section,
     + Select the ***LoadBalancer-SG* Security group** from the dropdown and **remove the default security group**.
   * For the **Listeners and routing** section,
   * The listener is already present with Protocol HTTP and Port 80.
     + **Select the target group *web-server-TG* for the Default action forwards to option.**
6. Keep the tags as default and click on the **Create load balancer** button.
7. **You** **have successfully created the Application Load balancer.**Click on the**View load balancers button.**
8. Wait for 2 to 3 minutes for the load balancer to become **Active**.

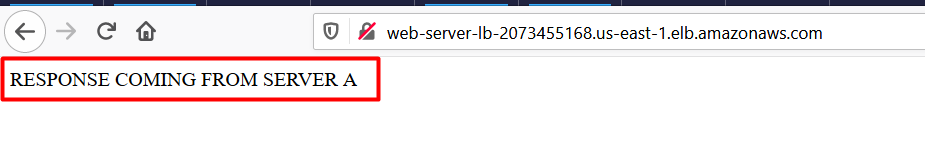


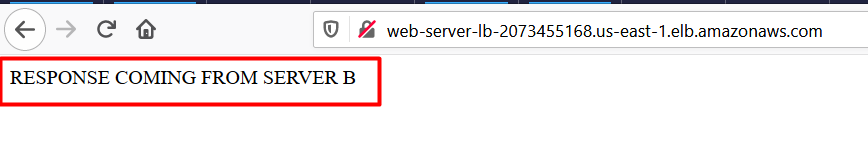
## ****Task 5: Testing the Load Balancer****

1. Navigate to**Load Balancers** and selectthe**load balancer** that you created. Under **details** scroll down you will be able to see the DNS name**,** **copy the DNS name**and paste it into the browser



1. Refresh the browser a few times and you will see the request is serving from both servers. You will see the output as **RESPONSE COMING FROM SERVER A & RESPONSE COMING FROM  SERVER B.** This showsthat load is shared between the two web servers via the Application Load Balancer.





**Task 6: Creating an IP set**

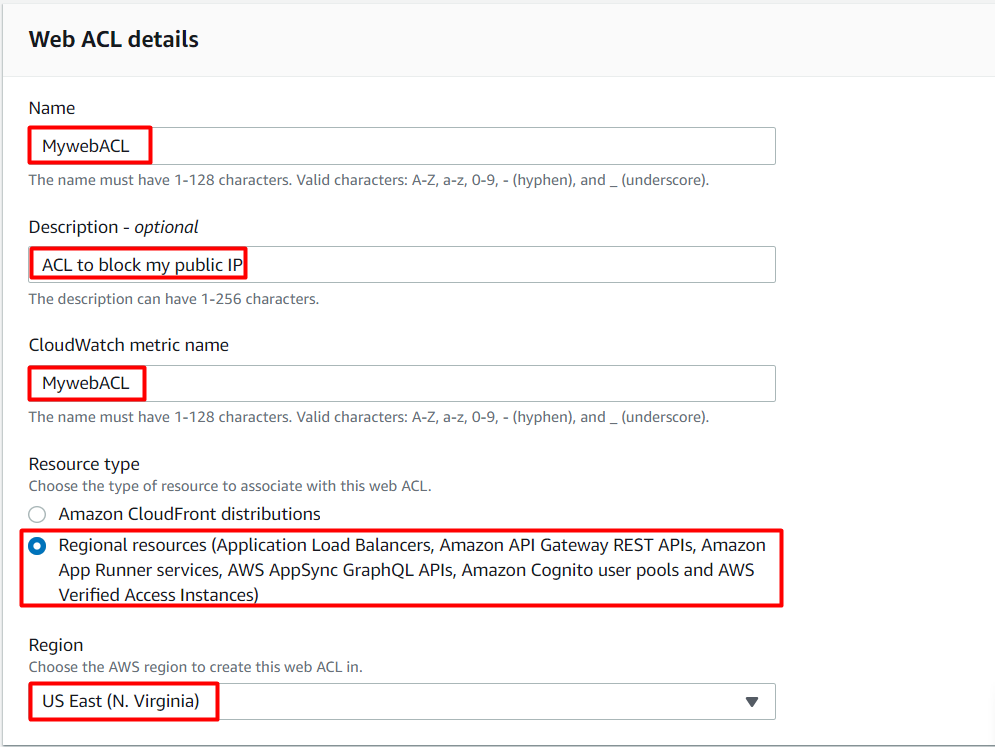
1. Click on **services** and select **WAF & Shield** under the **Security, Identity, & Compliance** section.
2. On the left side, you will be able to see the **IP sets** menu. Click on **IP sets** andclick on **Create IP sets.**
3. On the next screen, fill out the following details under **Create IP set.**

* **IP set details:**
  + IP set name: Enter ***MyIPset***
  + Description: Enter ***IP set to block my public IP***
  + Region: Select **US EAST (N.Virginia )**
  + IP Version: Select **IPv4**
  + IP address: Enter the ***IP of your local network/32*** from <https://www.whatismyip.com/>.
  + Note: You have to give /32 after the IP is pasted or else you won't be able to create an IP set.
* Once you have provided the above details, click on **Create IP set**

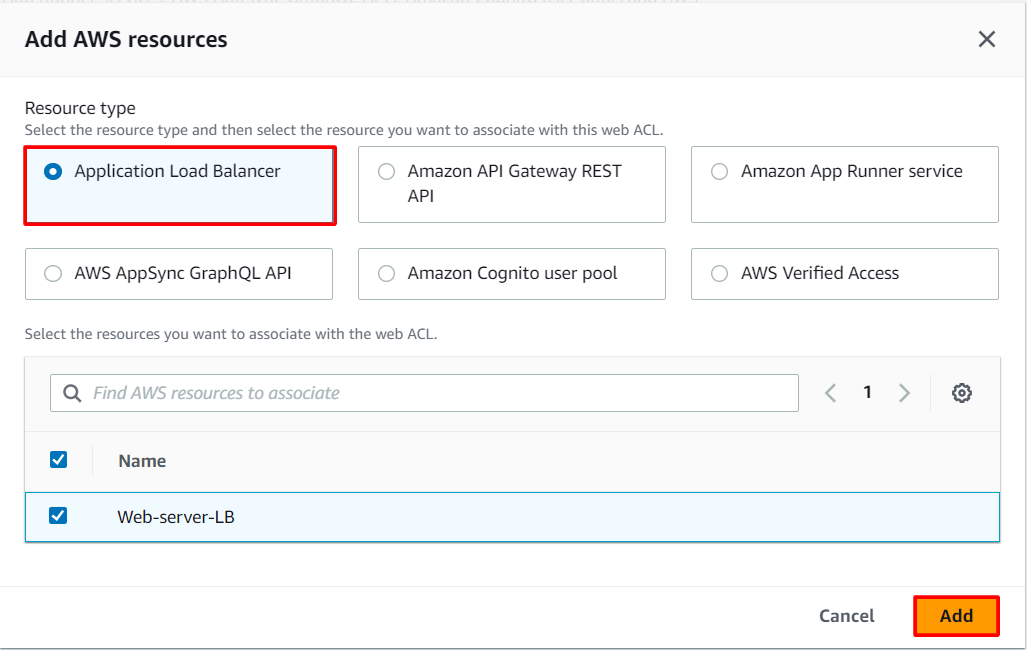
**Task 7: Creating a Web ACL**

1. **Web ACL details**

* Navigate to the **AWS WAF** dashboard and select **Web ACLs**. Click on **Create web ACL** to create a new web ACL.
* Configure the ACL as below:
  + **Web ACL details**
* Name: Enter ***MywebACL***
* Description: Enter ***ACL to block my public IP***
* Resource type:Select **Regional resources  (Application Load Balancer and API Gateway)**
* Region: Select **US EAST (N.Virginia)**



* To associate an AWS resource, click on **Add AWS resources**
* In Add AWS resources select **Application Load Balancer**and select the name of **ALB**. Click on **Add**



* Lastly, click on the **Next**button

1. **Add rules and rule groups**

* Under **Rules** click on **Add rule** and select  **Add my own rules and rule groups** in the drop-down menu.
* In **Rule type**select**IP set** as shown below and fill in the details as given below:
  + Rule type: Select **IP set**
  + Name: Enter ***MywebACL-rule***
  + IP set:  **select the IP set created Above ( MyIPset )**
  + IP address to use as the originating address**: Source IP address**
  + Action: Select **Block**
* Once you provide the above details, click on the **Add rule**.
* Lastly, click on the **Next**button

1. **Set rule priority**

* Leave as default and click on **Next**.

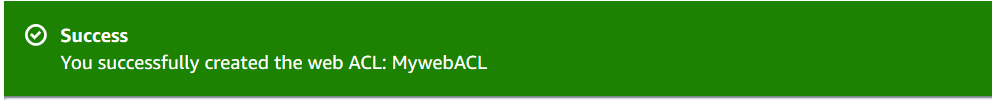
1. **Configure metrics**

* Leave as default and click on **Next.**

1. **Review and create web ACL**

* Review all your inputs and click on **Create web ACL**

1. Wait for 1 or 2 minutes until you will see that your web ACL is successfully created.

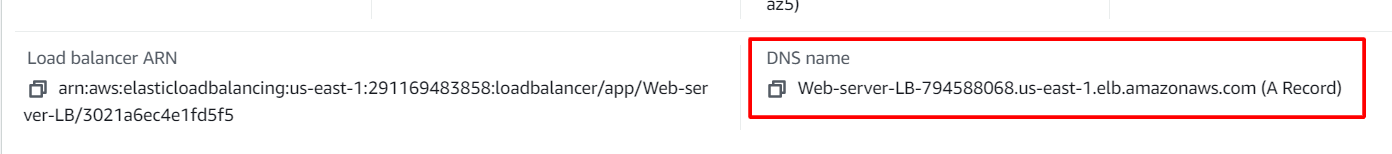


1. You have successfully created a web ACL for ALB with the help of an IP set created with your public IP.

## ****Task 8: Testing the working of the WAF****

1. To test the WAF, navigate  to **Load Balancers** from the EC2 left menu under the sub-heading **Load balancing**
2. Under the Load balancer section, select the Application load balancer**Web-server-LB.**
3. Copy the DNS name Under details scroll down you will be able to see the DNS name copy it and paste it in your desired browser.

* **Example**: web-server-lb-1903855210.us-east-1.elb.amazonaws.com

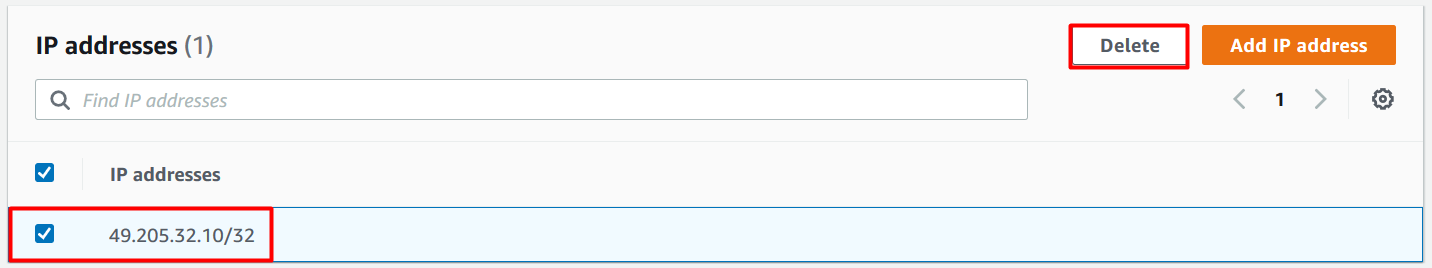


1. You will get a **403 forbidden error** showing that WAF blocked your connection to ALB.



## ****Task 9: Unblocking the IP****

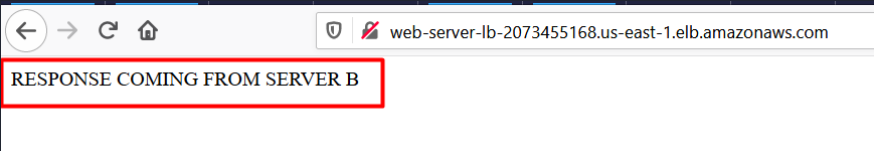
1. To unblock the IP, navigate to **IP sets** and click on **MyIPset.**Select your public IP and then click on **Delete**



1. Type **delete**in the confirmation box and click on **Delete**.
2. You have successfully removed the IP from WAF.
3. Wait for a few minutes.
4. Navigate to **Load Balancers** from the EC2 left menu under the sub-heading **Load balancing**
5. Under the Load balancer section, select the Application load balancer**Web-server-LB.**
6. Copy the DNS name under Description and paste it into your desired browser.

* **Example**: web-server-lb-1903855210.us-east-1.elb.amazonaws.com

1. You will get the response from the web servers either stating **RESPONSE COMING FROM SERVER A** or **RESPONSE COMING FROM SERVER B** as shown below:



### ****Do you know?****

Web Application Firewall (WAF): A Web Application Firewall is a security service that sits between a web application and its users, inspecting and filtering incoming web traffic. In the context of AWS, the AWS WAF service provides this functionality. It helps protect web applications by examining HTTP/HTTPS requests and applying a set of predefined rules or custom rule configurations to block, allow, or filter traffic based on specified criteria.

# ****Completion and Conclusion****

1. You have successfully created an IP set using your public IP.
2. You have successfully created a web ACL rule using an IP set and application load balancer (ALB).
3. You have successfully tested the working of the ALB after implementing a WAF, blocking the web request to the ALB from your local network.
4. You deleted the IP set and tested the working of the ALB.