How does AWS WAF Protect?

Introduction

AWS WAF protects your web applications from common web exploits

It is a web application firewall service that lets you monitor web requests forwarded to an Amazon API Gateway API, an Amazon CloudFront distribution, or an Application Load Balancer.

You can protect those resources based on conditions you specify, such as the IP addresses from which the requests originate.
It comes under → Security, Identity, and Compliance Services

Objective

We will understand the importance of AWS WAF by implementing it to block my laptop or any IP address from accessing EC2 instances (accessed via Load balancer).
The uses of AWS WAF are boundless. It can be used to set up other rules like CAPTCHA, Count, and Challenge and enable CloudWatch metrics.

Prerequisites

AWS Account: You need an AWS account to access and use AWS WAF. If you do not have an account, you can sign up for one on the AWS website.

Web Application: AWS WAF is designed to protect web applications deployed on AWS services such as Amazon CloudFront, Application Load Balancer, or Amazon API Gateway. Therefore, you should have a web application deployed on these services or plan to deploy one.

AWS Management Console or AWS CLI: You can manage AWS WAF either through the AWS Management Console, which provides a web-based interface, or by using the AWS Command Line Interface (CLI). Ensure that you have access to either of these tools to configure and manage AWS WAF.

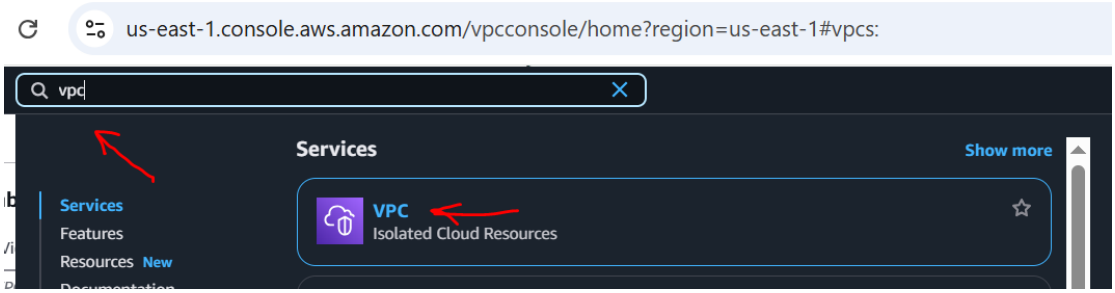
Basic Understanding of Web Application Security: While AWS WAF simplifies the process of protecting your web applications, it is beneficial to have a basic understanding of common web application security vulnerabilities and best practices. This knowledge helps create effective rules and configure AWS WAF to suit your application's needs.

By meeting these prerequisites, you can effectively leverage the benefits of AWS WAF to enhance the security of your web applications in the AWS ecosystem.

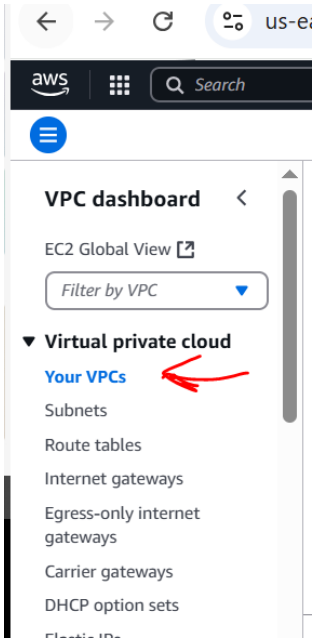
IMPLEMENTATION STEPS

1. Create VPC

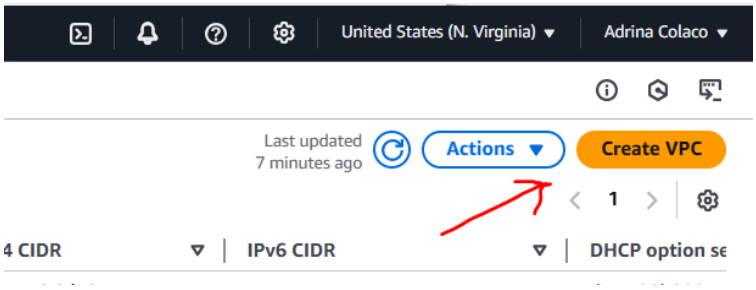
a. Type “VPC” in the search bar and click on “VPC” in the search results



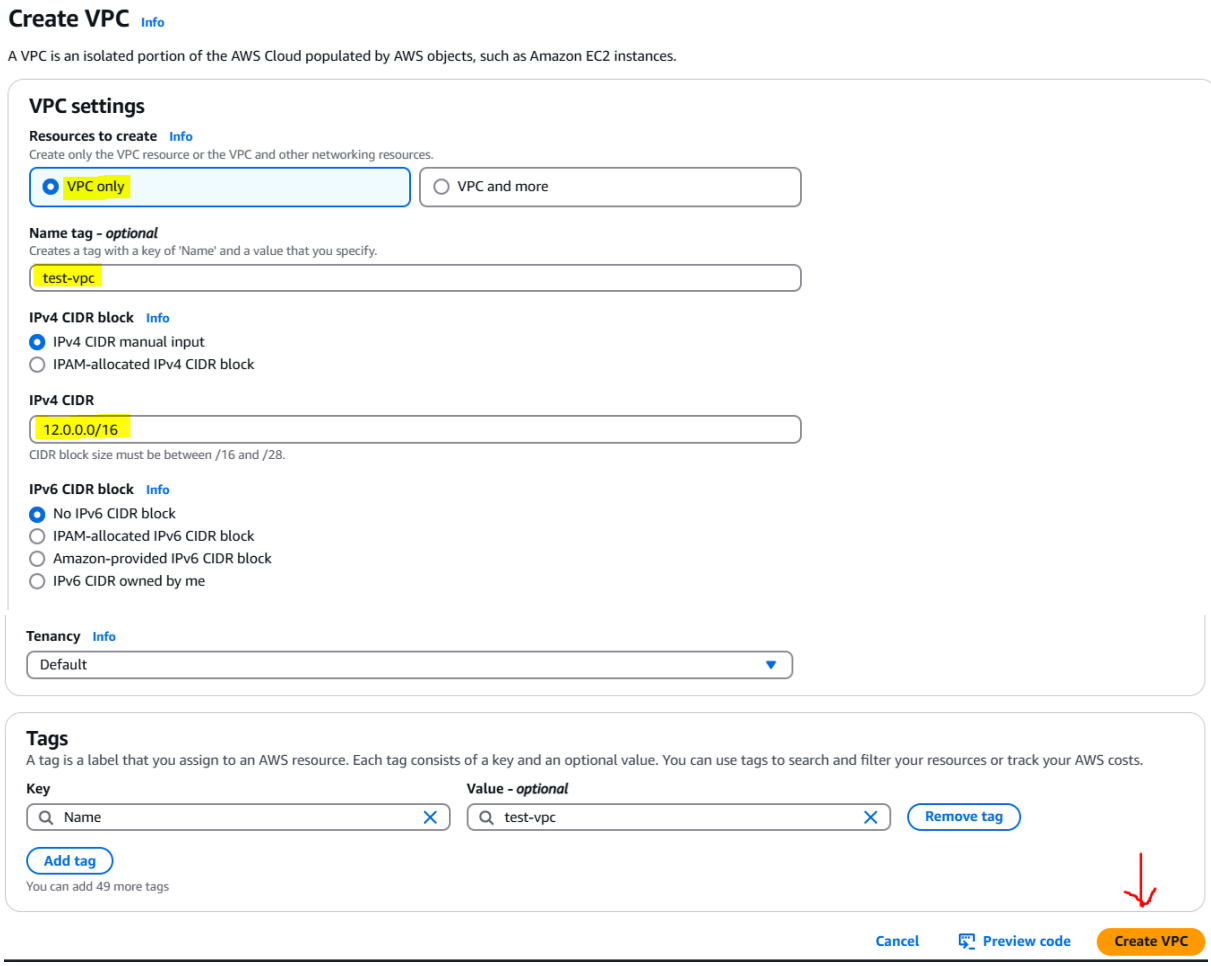
b. Go to “Your VPC” on the left panel.



c. Click on “Create VPC” on the right corner.

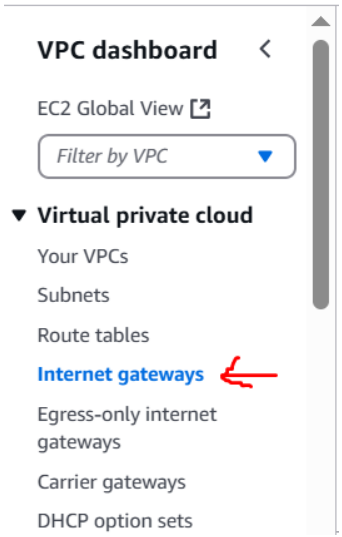


d. Enter the highlighted details in the Launch an instance page and click on “Create VPC”

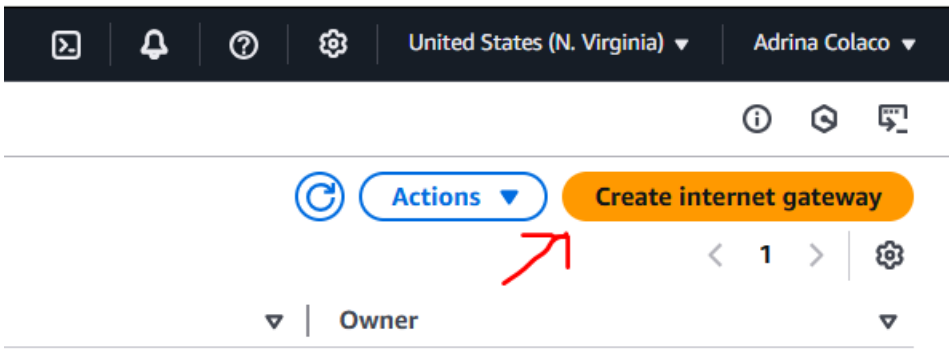


2. Create Internet Gateway

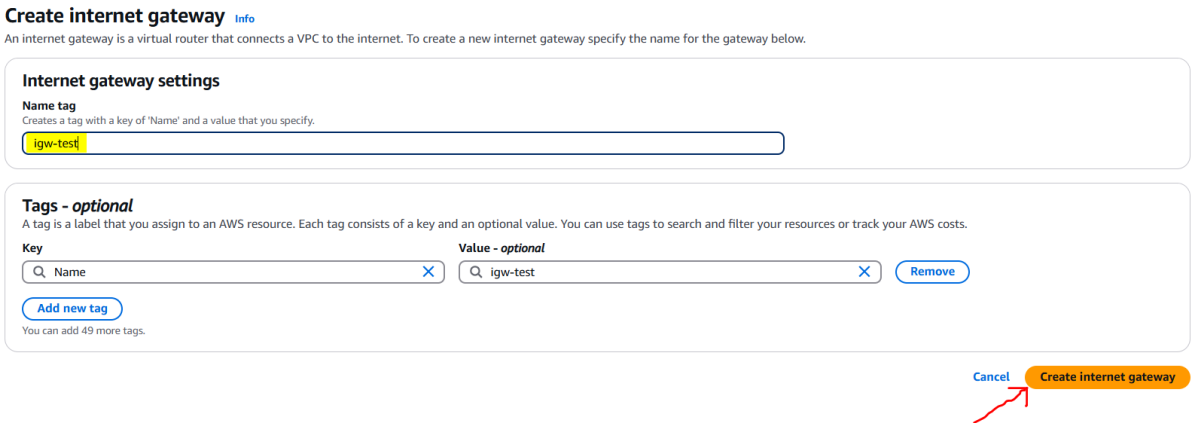
a. Click on “Internet Gateways” in the left panel.



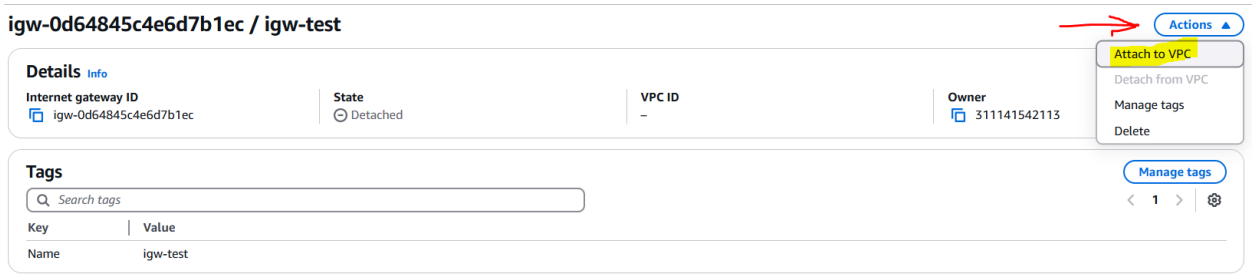
b. Click on “Create internet gateway”



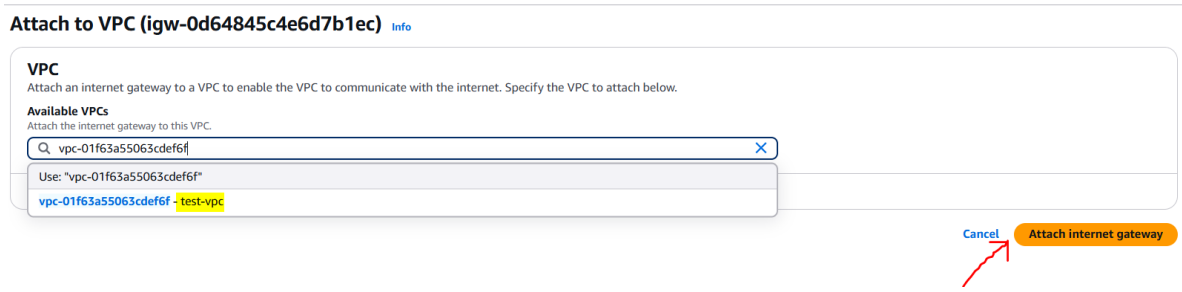
c. Enter the details in highlighted fields and click on “Create internet gateway”.



d. Goto “Actions” and click on “Attach to VPC”

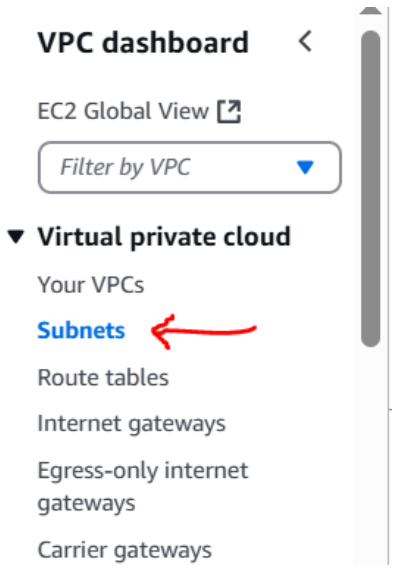


e. Select the respective VPC and click on “Attach internet gateway”.

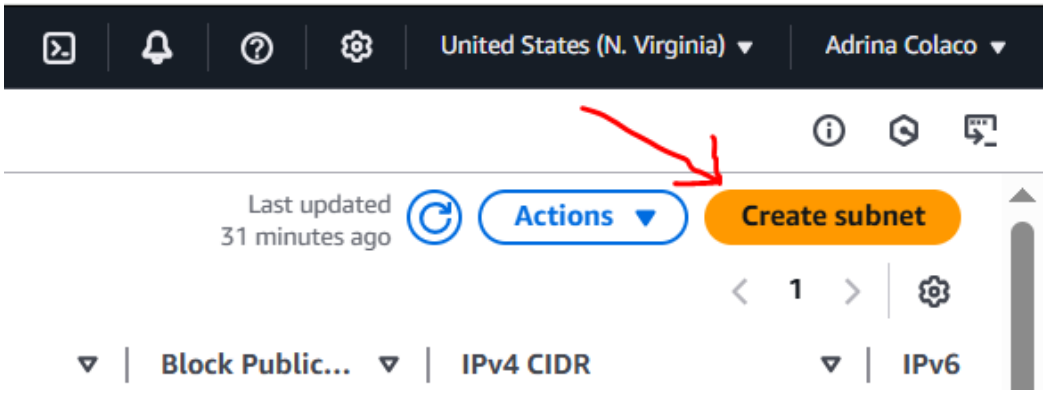


3. Create Public Subnet

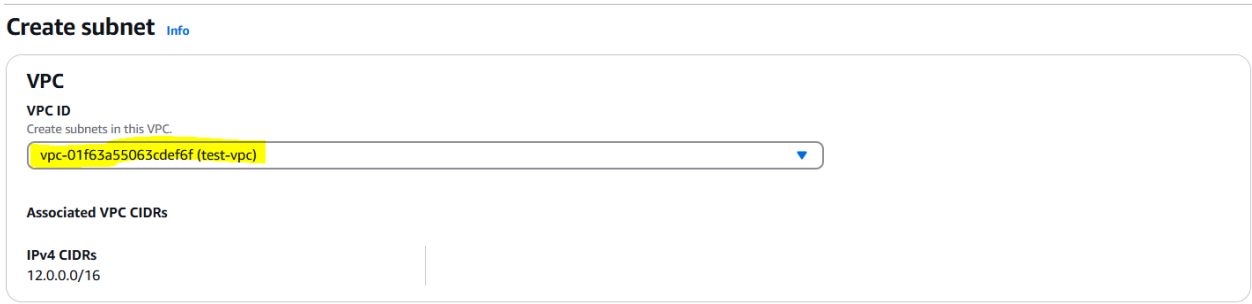
a. Go to “Subnets” on the left panel.



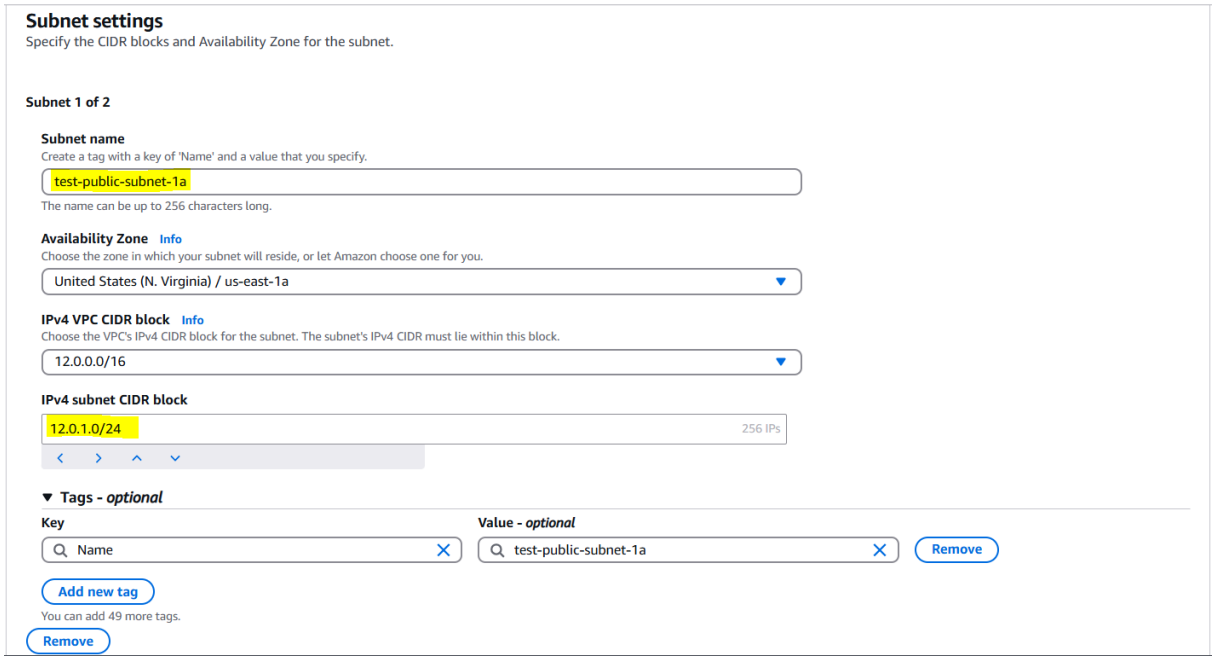
b. Click on “Create subnet”



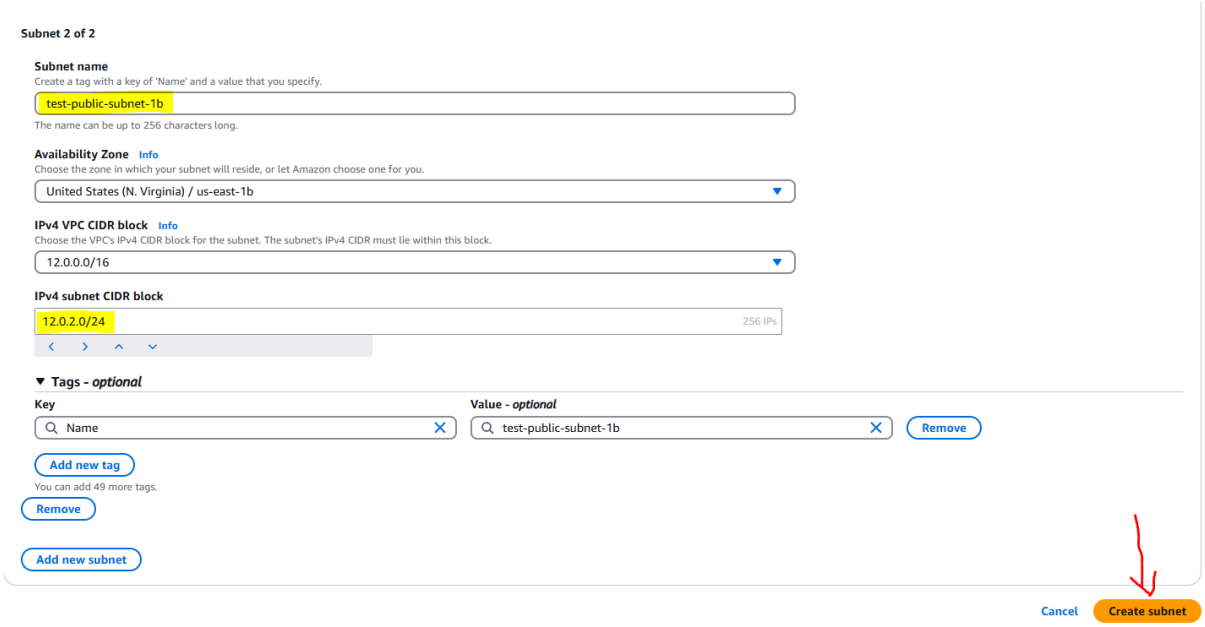
c. Create two subnets in 2 different availability zones and fill out the highlighted fields.



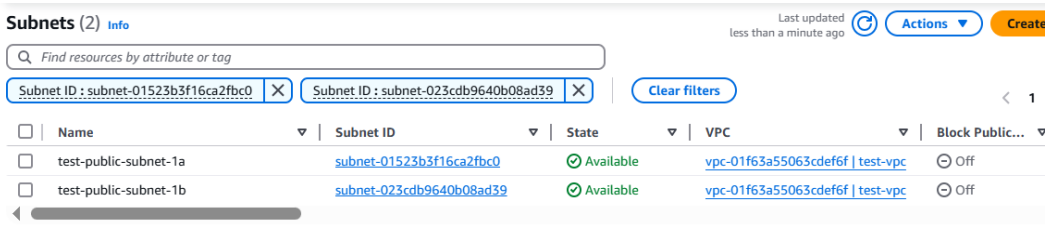
Details of first Subnet



Add details of the second subnet by clicking on the “Add new subnet” button. Enter details as highlighted below and click on “Create subnet”.

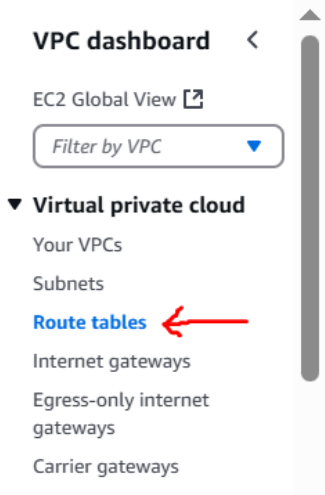


d. Subnets will be created.

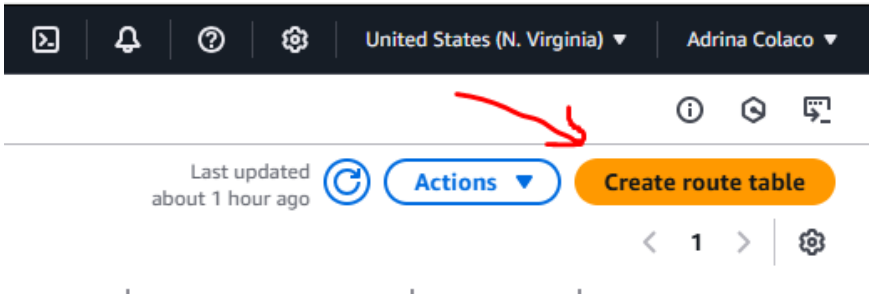


4. Create Route table

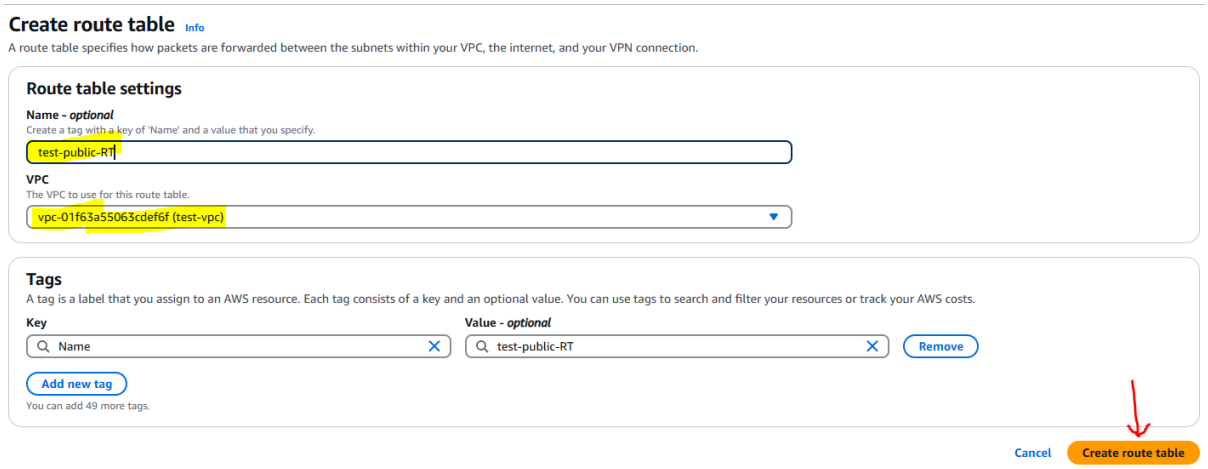
a. Go to “Route tables” in the left panel.



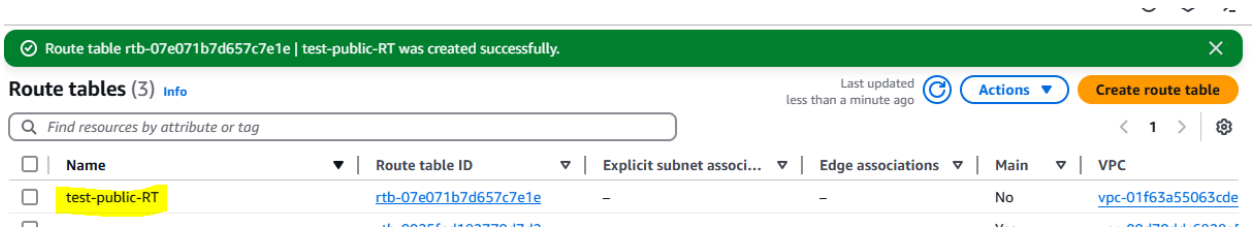
b. Click on the “Create route table” button in the right corner.



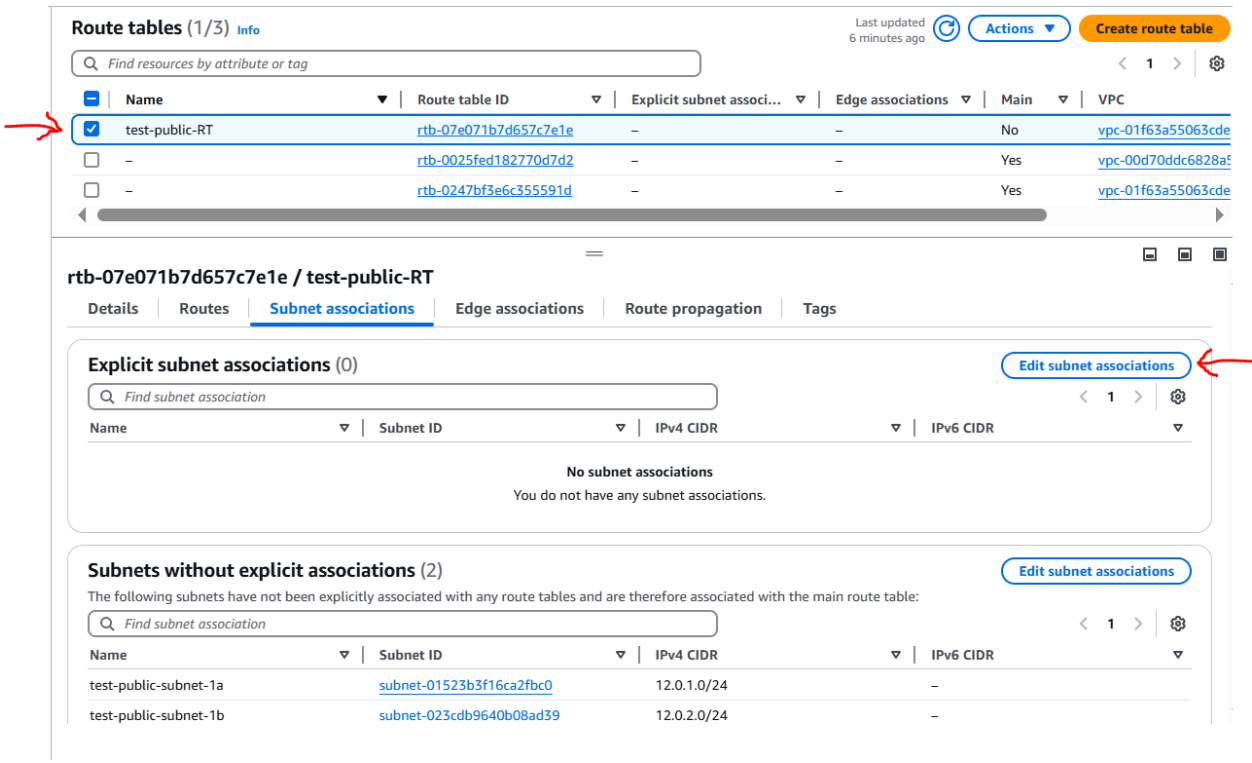
c. Enter details into the highlighted fields and click on the “Create route table” button.



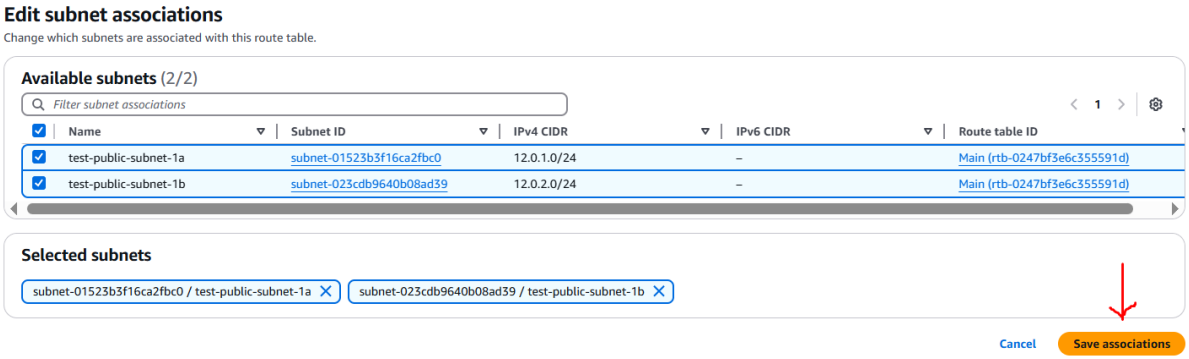
d. The route table will be successfully created.



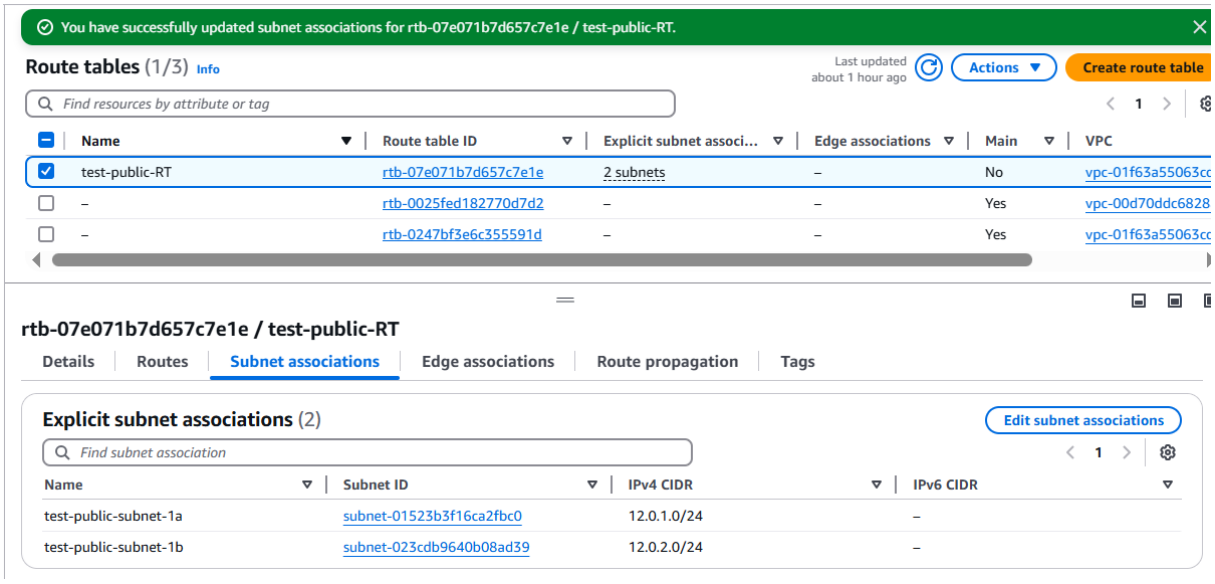
e. Associate Subnets to the Route table by clicking on “Edit subnet associations”.



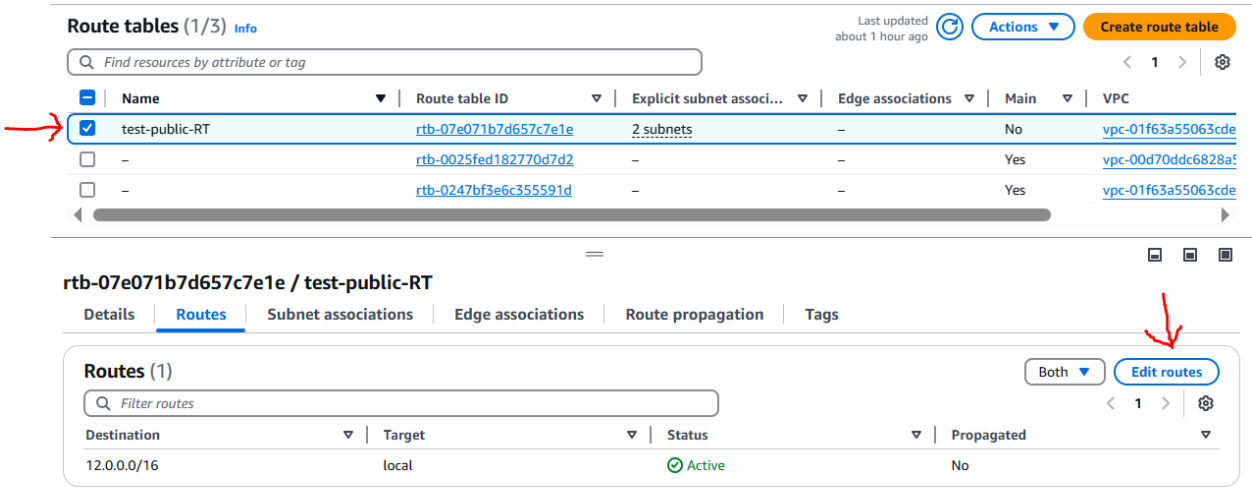
f. Check mark on both the subnets and click on “Save associations”



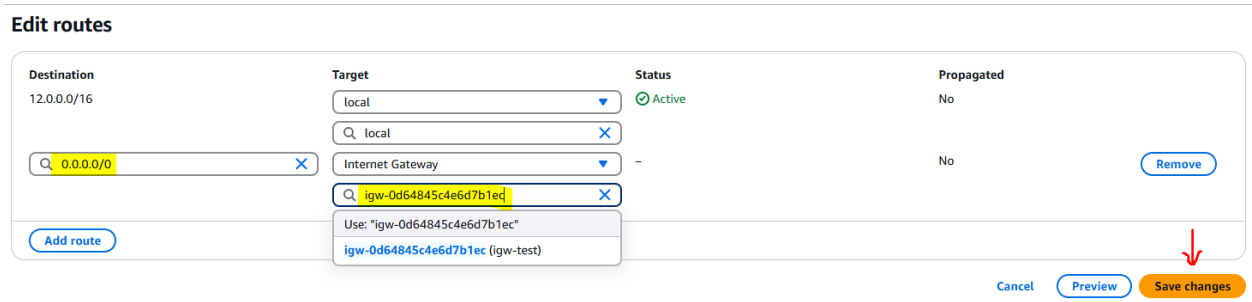
Subnets associated with the Route table.



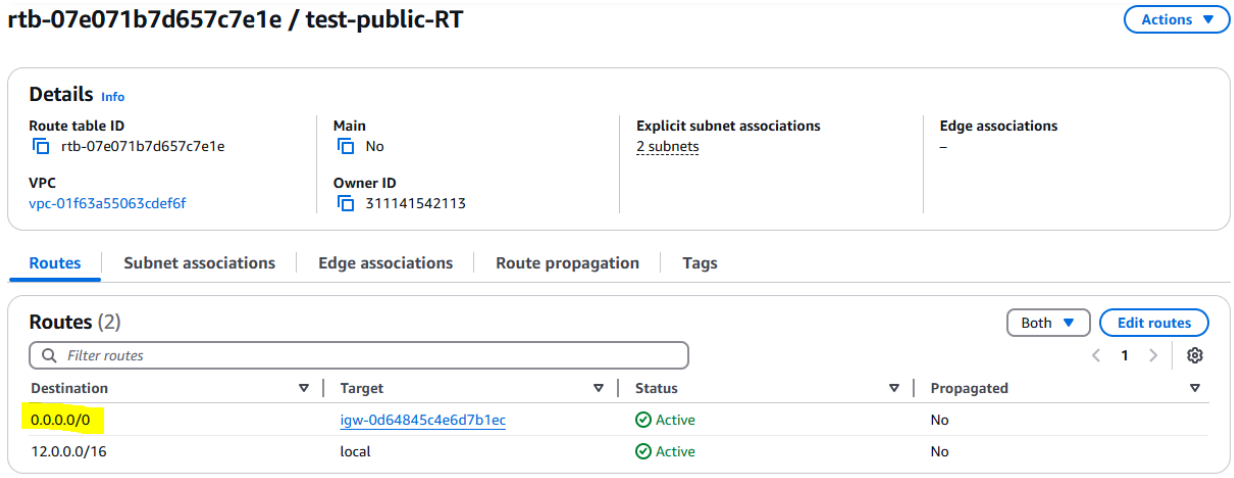
g. Edit the Routes to provide internet access to the Route table.



h. Click on “Add route” and add the highlighted fields. Click on “Save changes”.



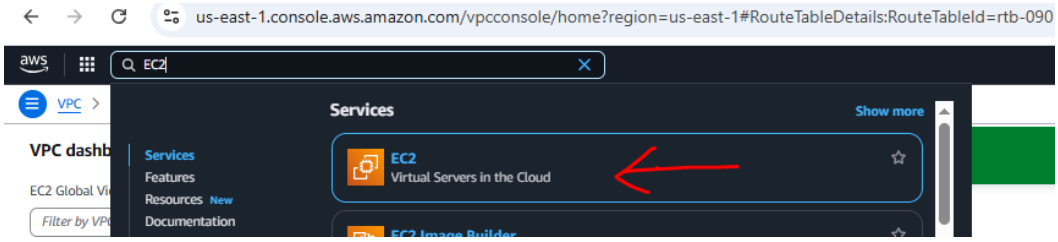
i. Routes should be updated and active.



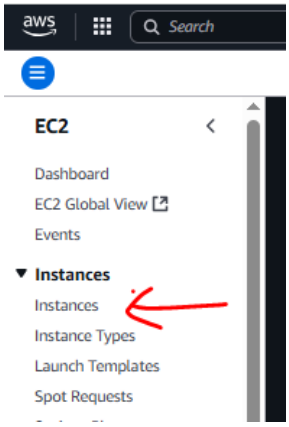
5. Create an "EC2" instance.

a. The goal is to create two different EC2 instances belonging to the 2 respective subnets that we created earlier.

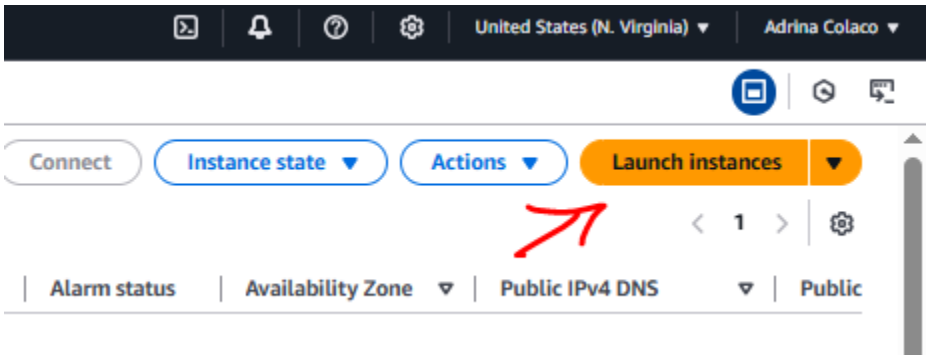
b. Type "EC2" in the search bar and click on "EC2" in the search results.



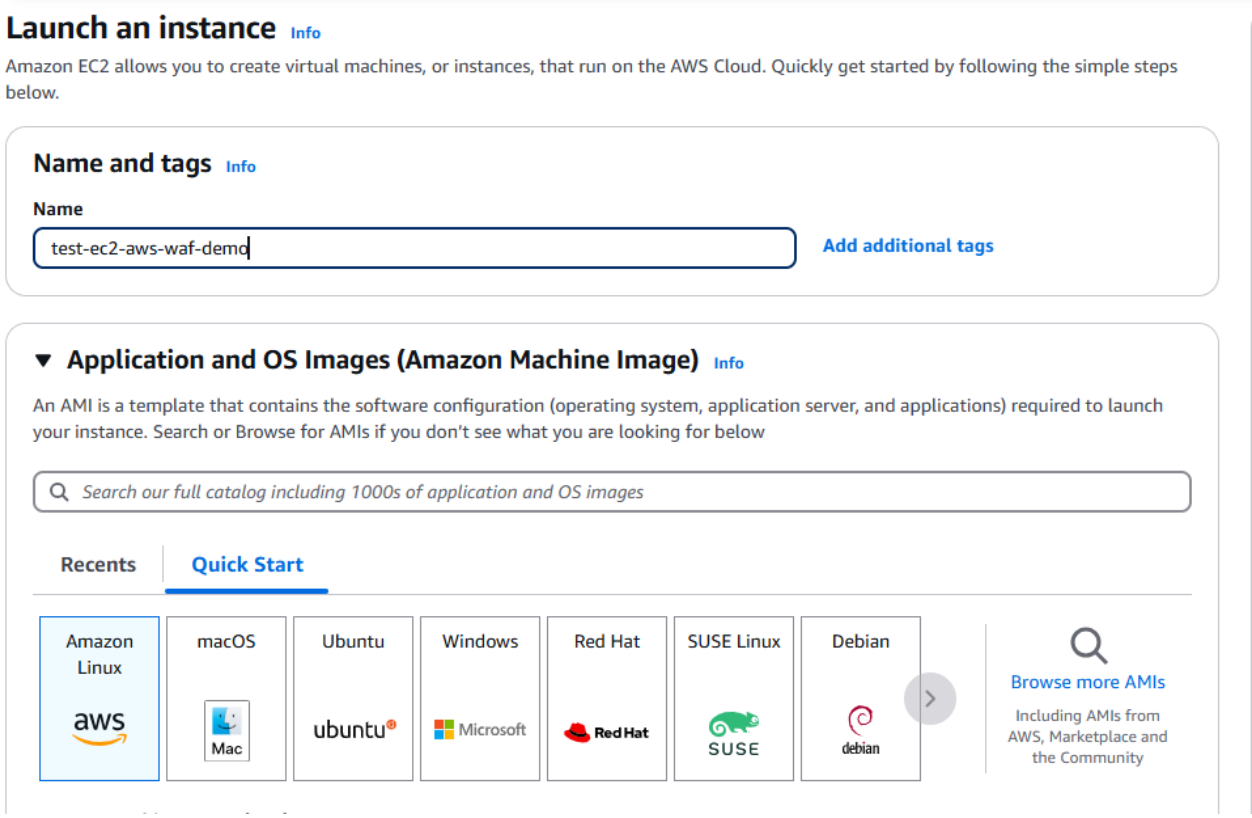
c. Goto "Instances" on the left panel.

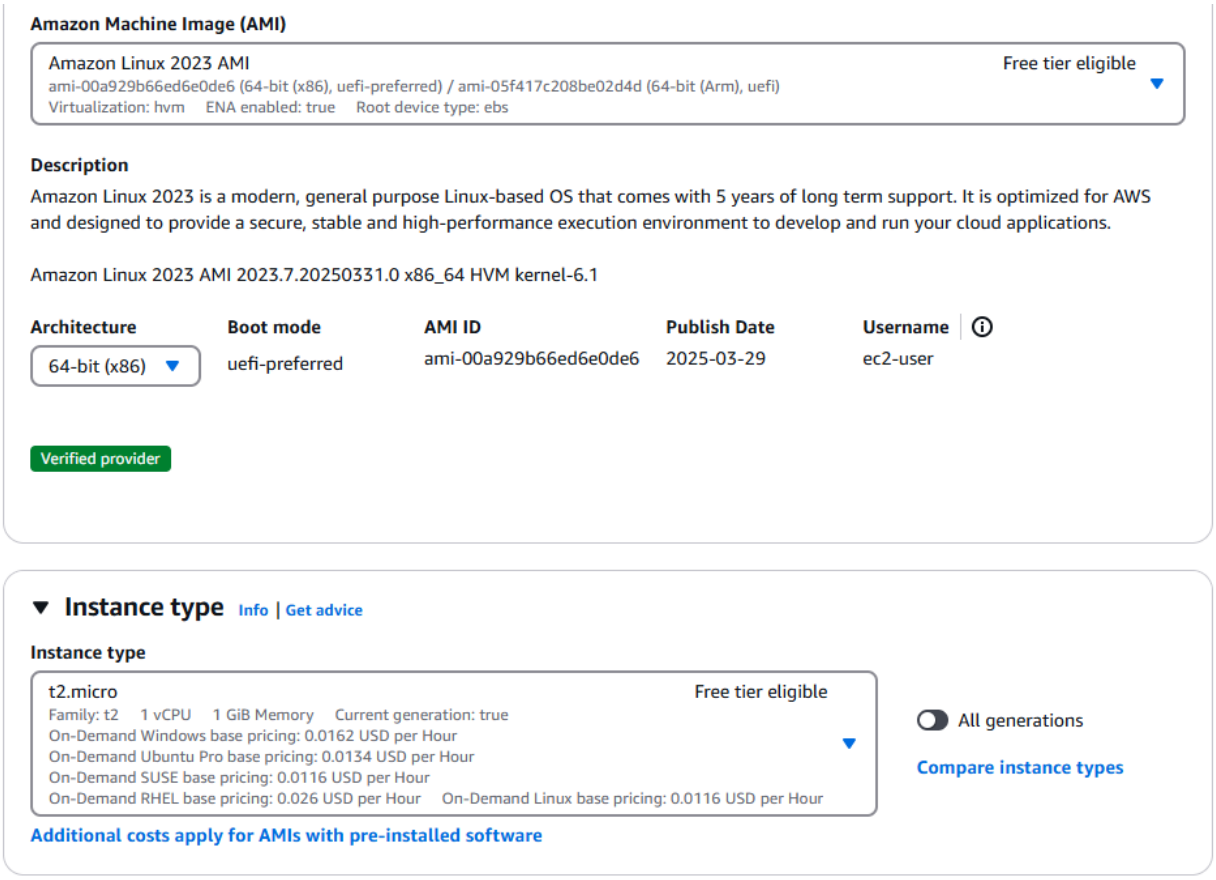


d. Click on "Launch instance" on the right corner.

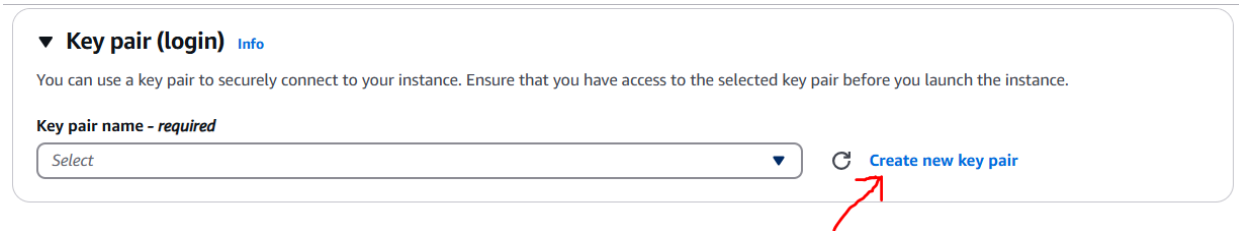


e. Enter the highlighted details in the Launch an instance page.

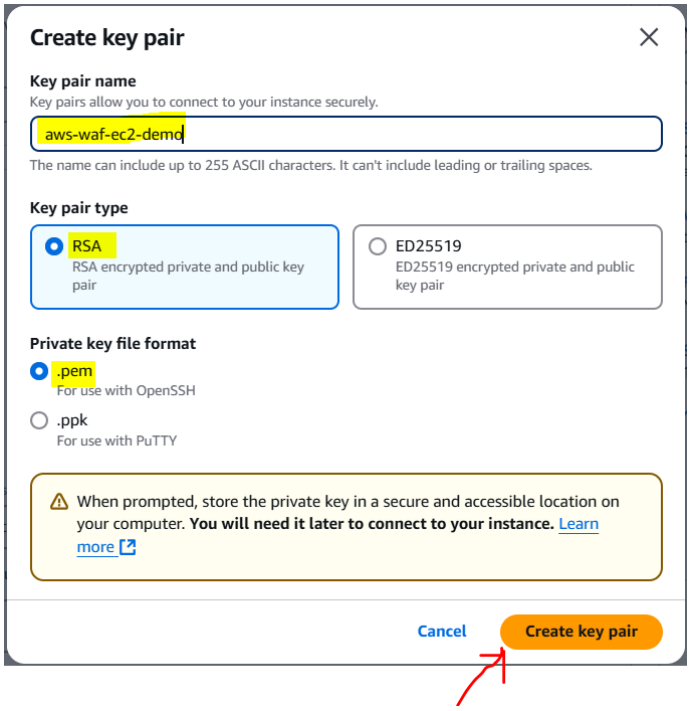




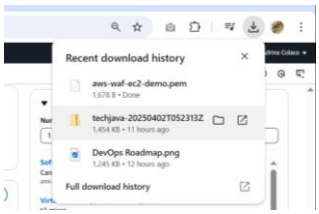
f. Create a new Key Pair



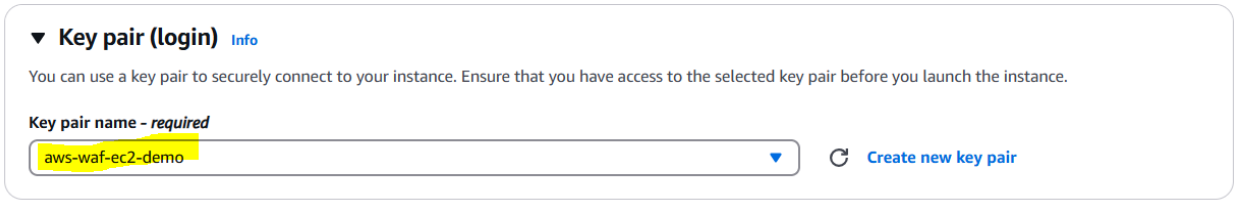
g. Enter the highlighted fields and click on “Create key pair”



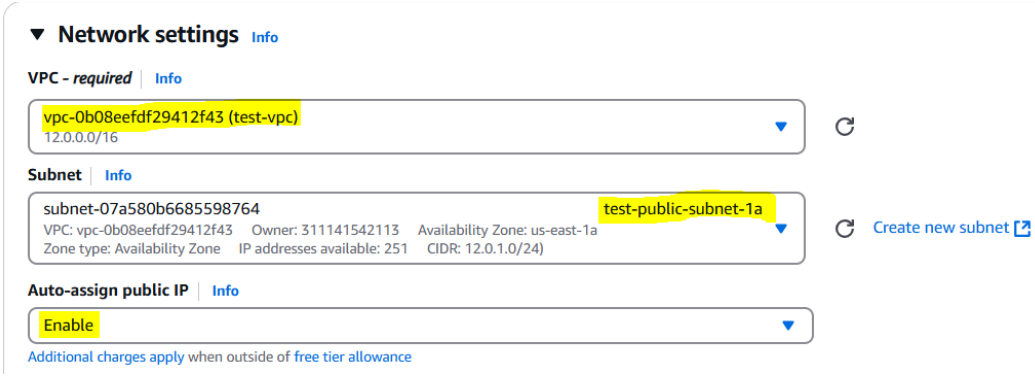
h. The key pair will automatically be downloaded to your “Downloads” folder.



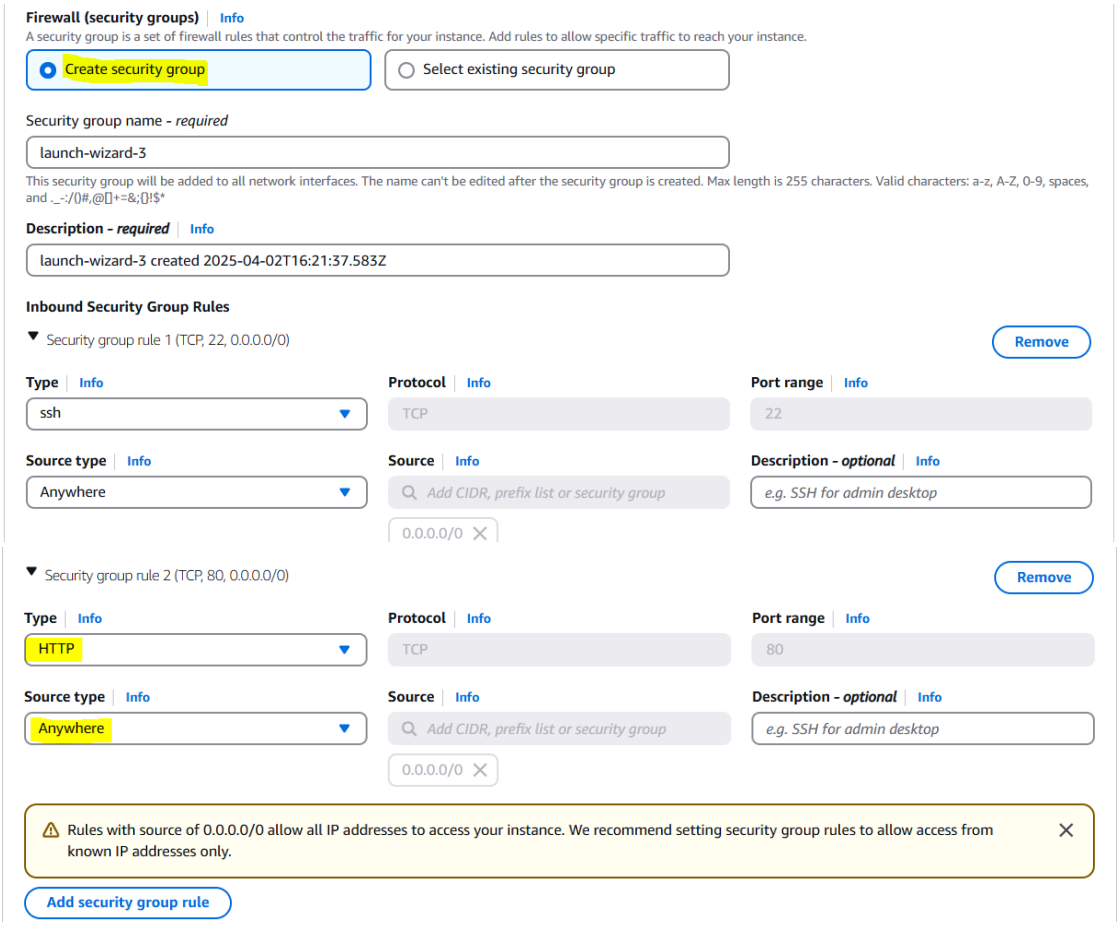
i. The key pair will also attach to the EC2.



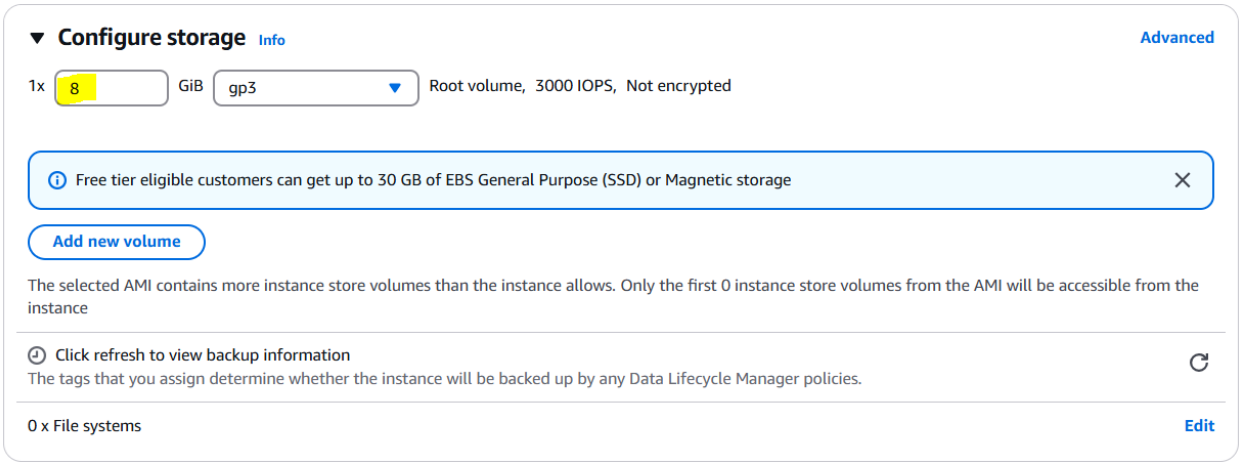
j. Network setting:
   i. Edit and change the highlighted fields
      1. VPC
      2. Subnet (Select the respective subnet for each EC2 instance; we are creating 2 EC2 instances)
      3. Change Auto-assign public IP to “Enable”



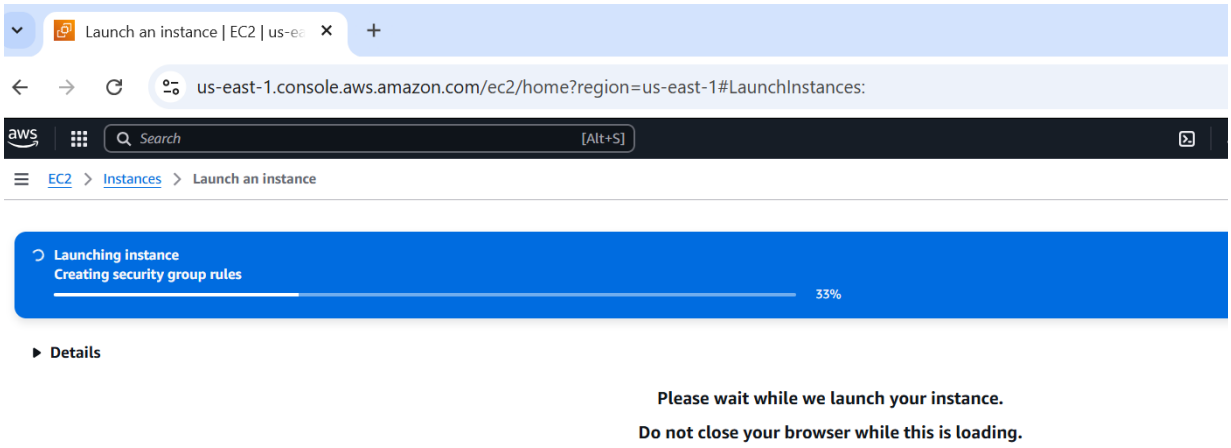
ii. Create a Security Group and add the HTTP security group rule as highlighted below.



k. Leave “Configure storage” and “Advanced details” as is.



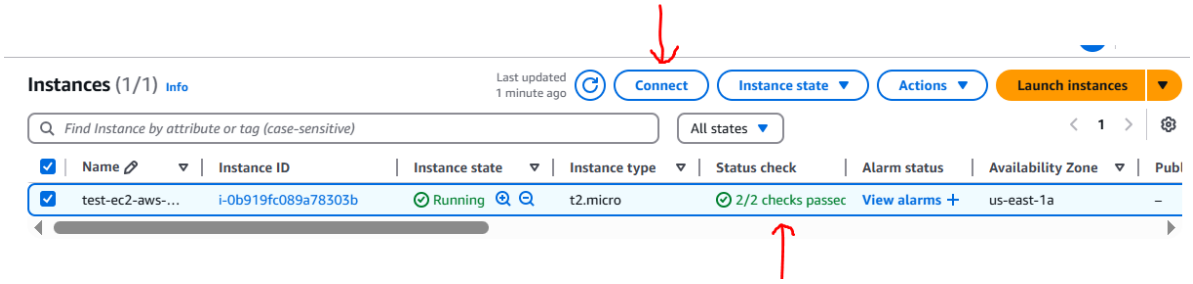
l. Click on “Launch instance”. You will see the screen below.



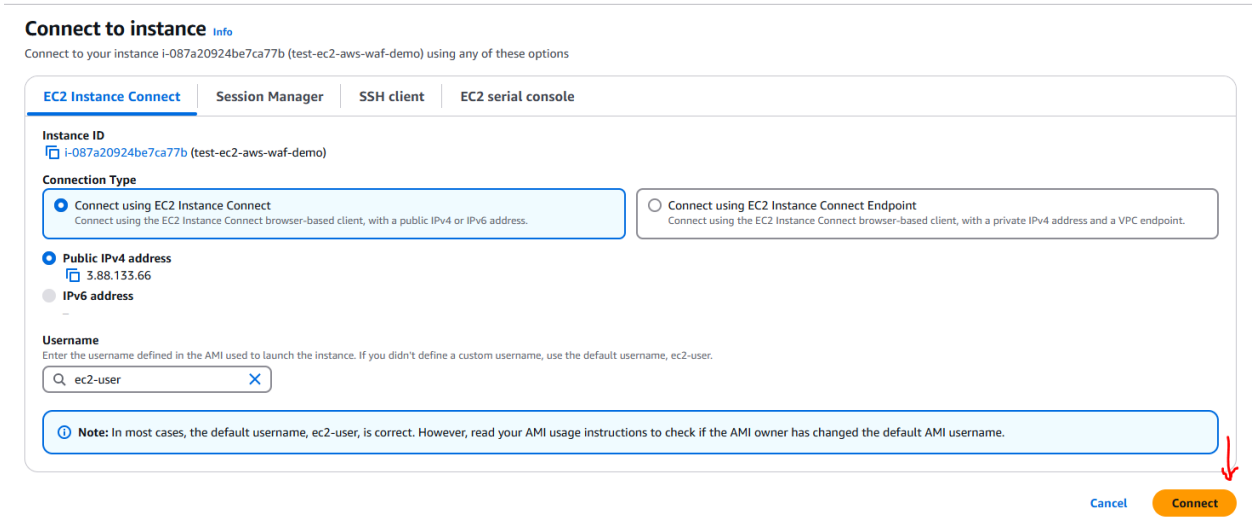
6. Installation of HTTPD service on EC2.

a. Connect to the EC2 instance.

b. Click on the check mark next to the EC2 instance. Make sure the Status Check is **GREEN**. Click on the “Connect” button.



c. Connect to the EC2 instance by clicking on “Connect.”



d. Install, start, and enable the HTTPD service on EC2 using the below commands(One at a time).

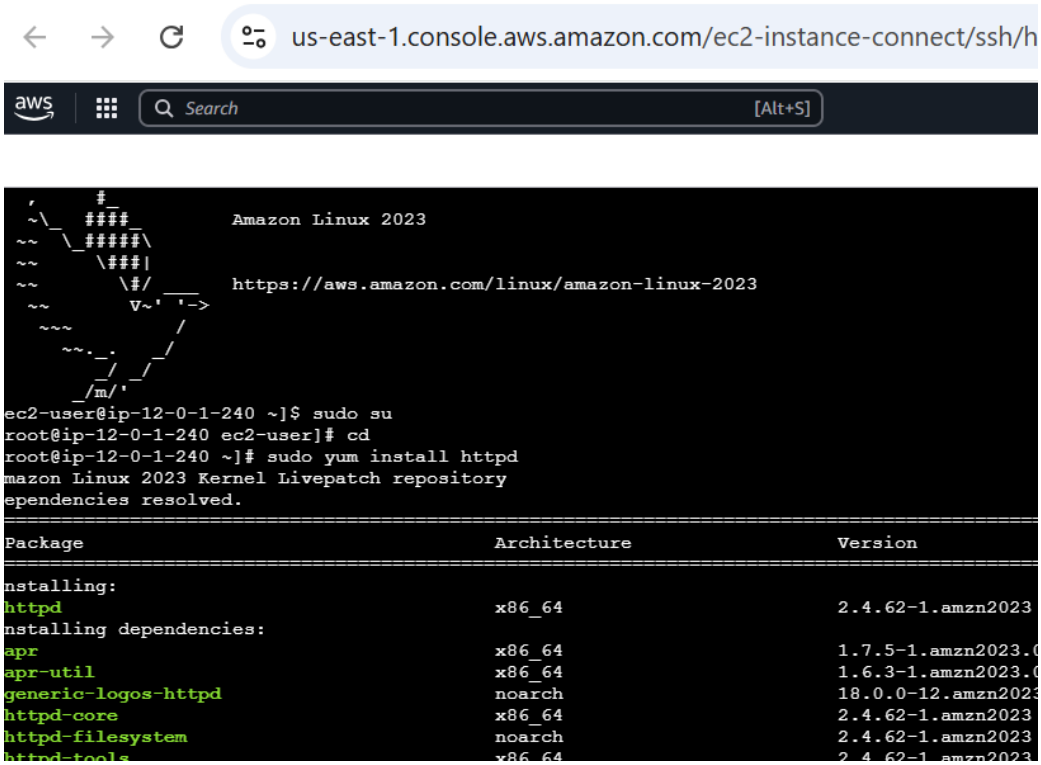
sudo yum install httpd -y

sudo systemctl enable httpd

sudo systemctl status httpd

sudo systemctl start httpd

sudo systemctl status httpd



e. Create “index.html” in /var/www/html folder
   i. Write the below command in the command CLI.
      cd /var/www/html
      Vim index.html

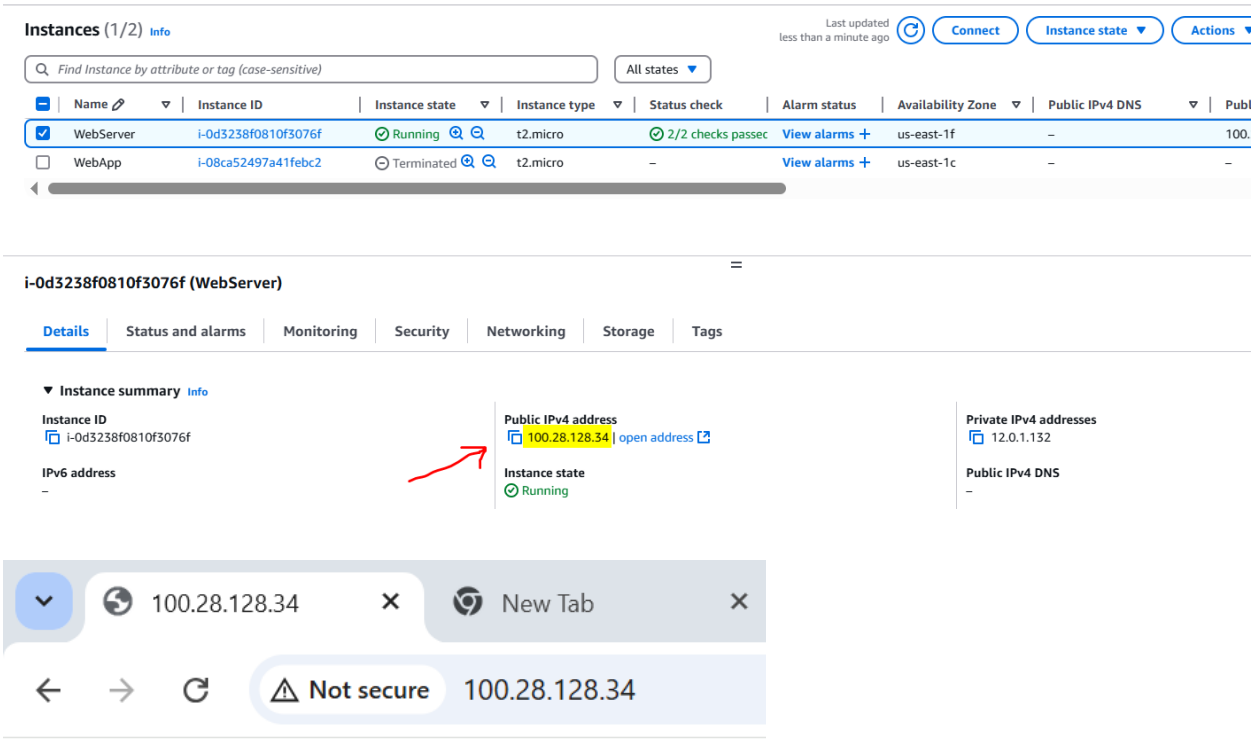
Write the bash code below in index.html and save it by pressing [ESC] →:wq [Press enter]
Replace Linux Server 1 with Linux Server 2 in the below code while creating this file on the second EC2 instance.

```
<!DOCTYPE html>
<html>
<body>

<h1>Hello World!</h1>
<p>You have reached Linux Server 1.</p>

</body>
</html>
```

7. ## Testing of EC2 instance

Copy the Public IPv4 address from the EC2 and place it in the browser to check if the website is displaying.

Use **http**://<IP address>

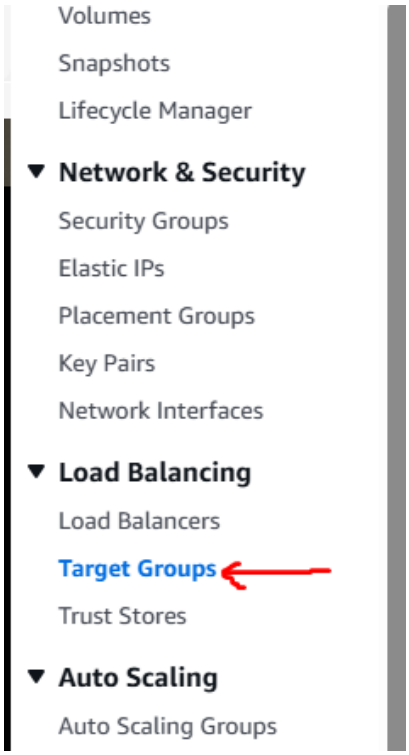


Hello World!

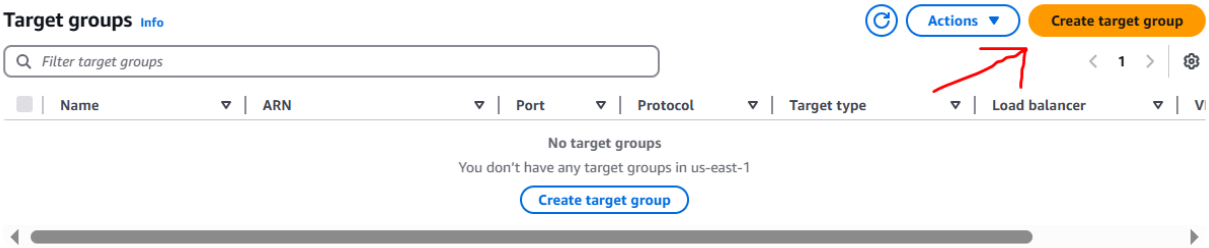
You have reached Linux Server 1.

8. ## Target Group

a. Go to “Target groups” in the left pane.



b. Click on “Create target group”



c. Enter details in the highlighted fields.

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose a target type

- (●) Instances
  - Supports load balancing to instances within a specific VPC.
  - Facilitates the use of Amazon EC2 Auto Scaling to manage and scale your EC2 capacity.
- ( ) IP addresses
  - Supports load balancing to VPC and on-premises resources.
  - Facilitates routing to multiple IP addresses and network interfaces on the same instance.
  - Offers flexibility with microservice based architectures, simplifying inter-application communication.
  - Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.
- ( ) Lambda function
  - Facilitates routing to a single Lambda function.
  - Accessible to Application Load Balancers only.
- ( ) Application Load Balancer
  - Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
  - Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

tg-ec2-instances

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

Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

HTTP | 80

1-65535

IP address type

Only targets with the indicated IP address type can be registered to this target group.

- (●) IPv4
  Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.
- ( ) IPv6
  Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). Learn more

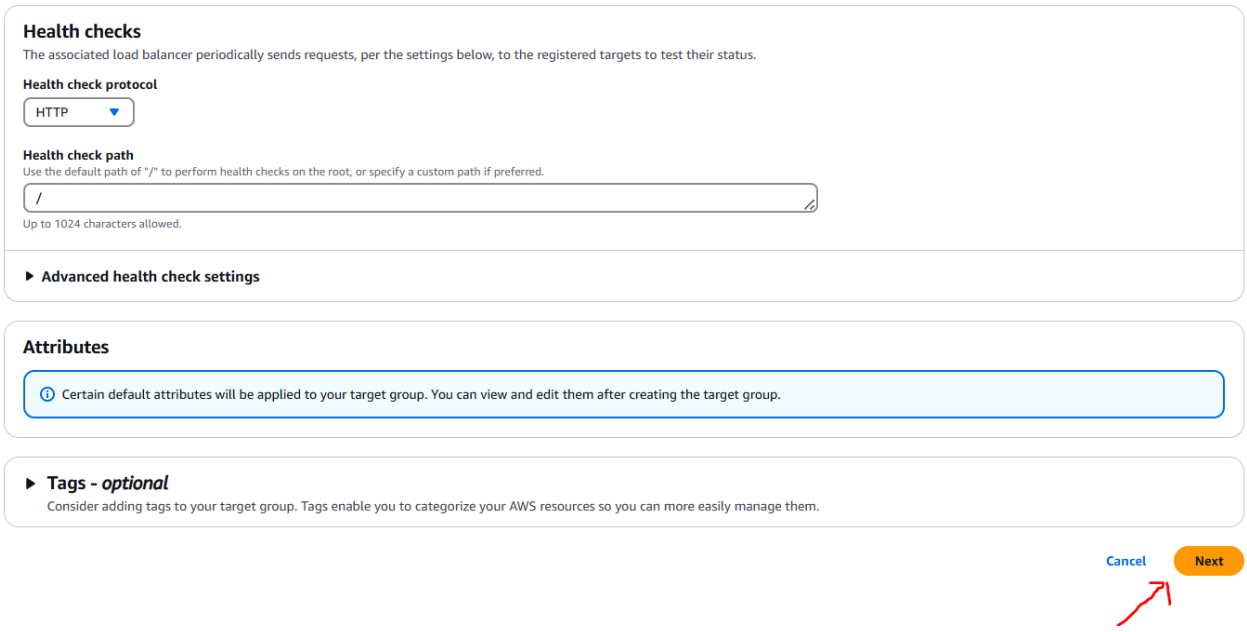
VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

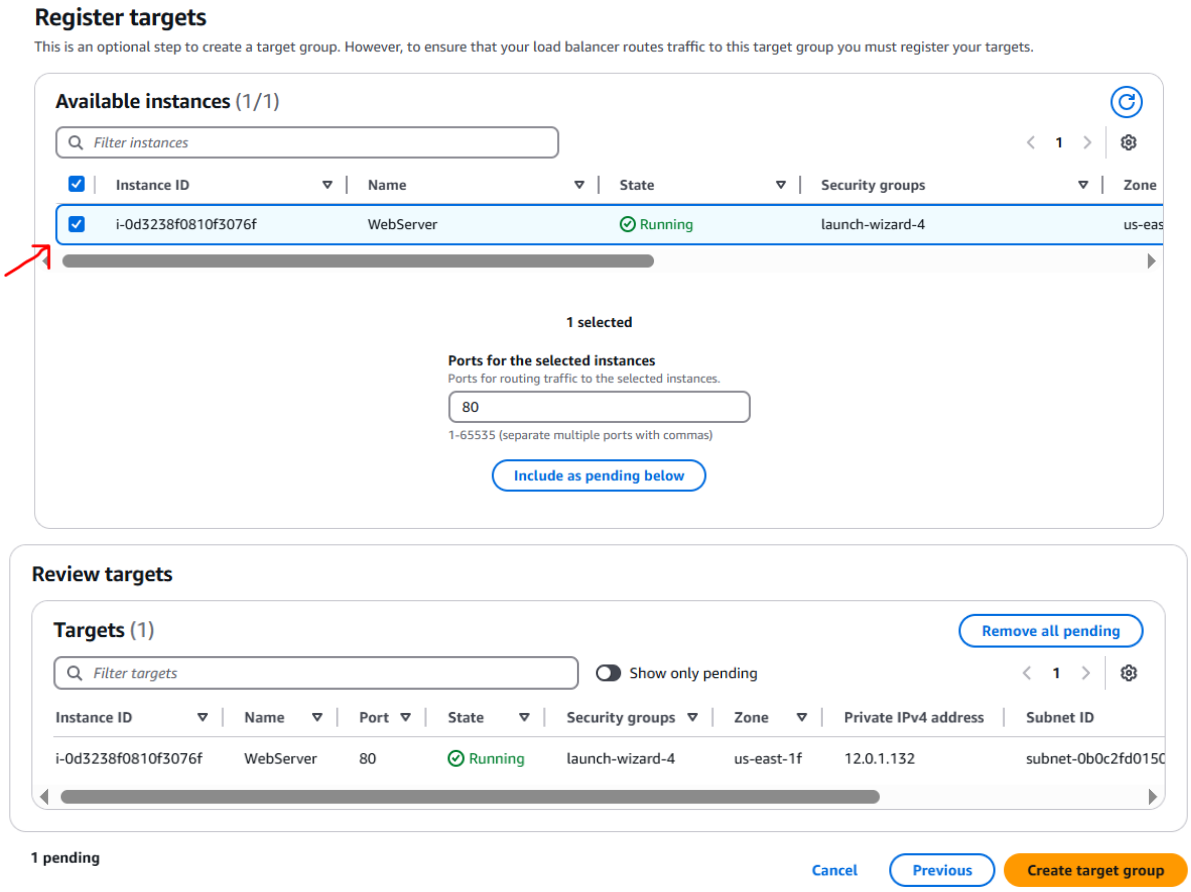
my-vpc
vpc-03ad40e7bb181aaf9
IPv4 VPC CIDR: 12.0.0.0/16

Protocol version

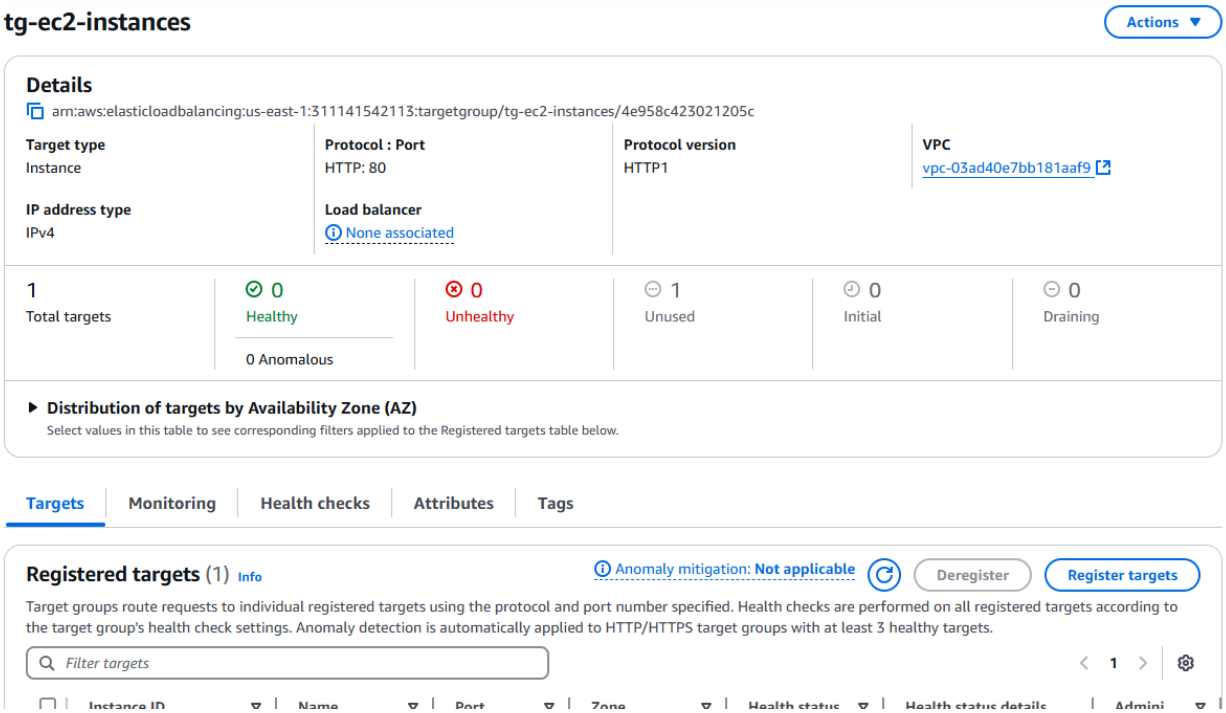
- (●) HTTP1
  Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
- ( ) HTTP2
  Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.
- ( ) gRPC
  Send requests to targets using gRPC. Supported when the request protocol is gRPC.



d. Click on the checkbox, select the instance ID, and click “Include as pending below”.
Then click on “Create target group.”

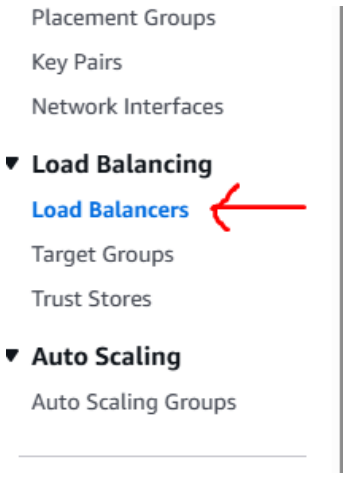


e. Target group will be created.

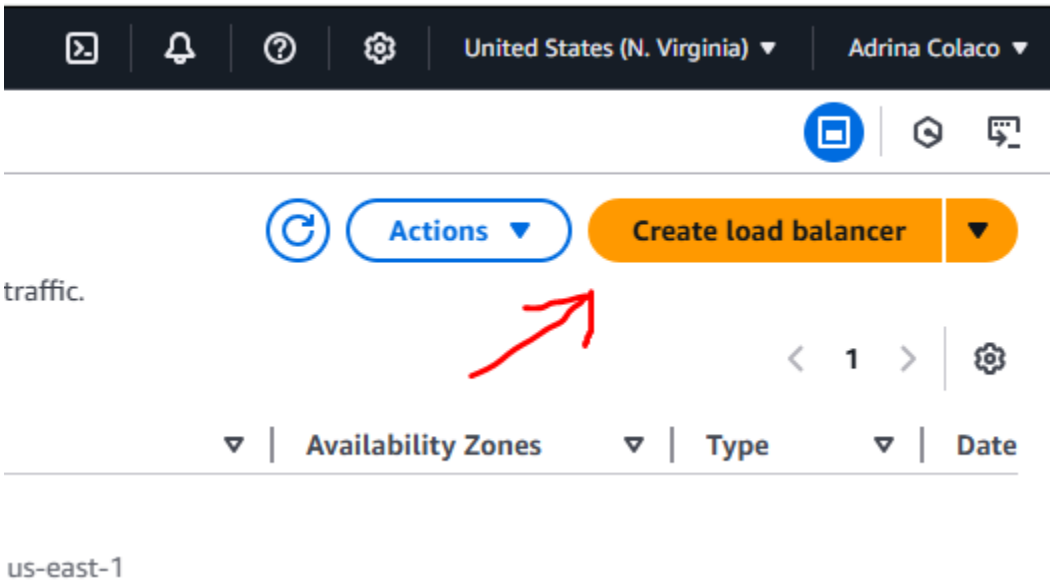


9. Load balancer

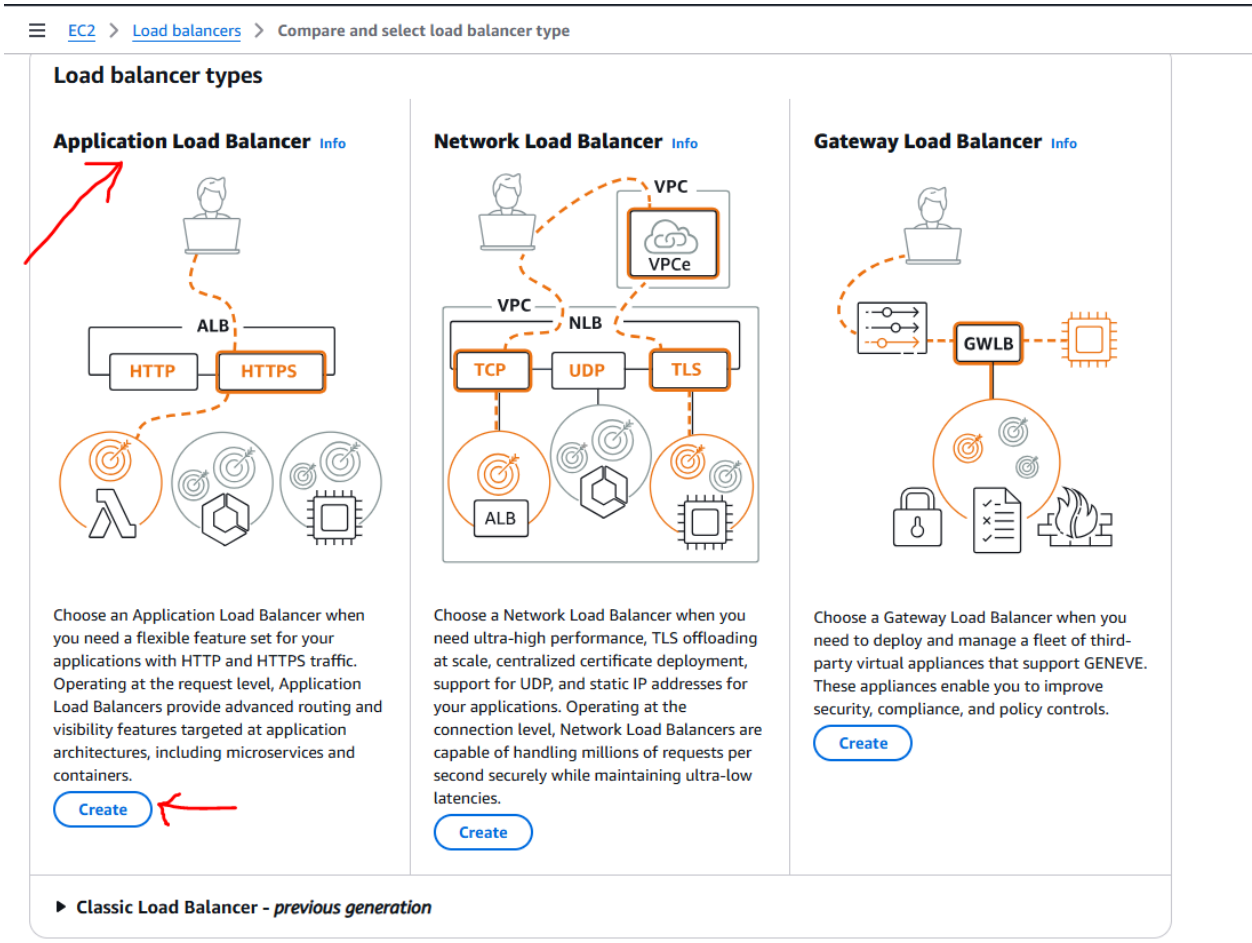
a. Click on “Load Balancers” on the left panel.



b. Click on “Create load balancer”



c. Click on “Application Load Balancer”



d. Enter details into the highlighted fields. Select the relevant VPC and at least two subnets.

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

▶ **How Application Load Balancers work**

Basic configuration

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

lb-waf-demo

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
  - Serves internet-facing traffic.
  - Has public IP addresses.
  - DNS name is publicly resolvable.
  - Requires a public subnet.
- ( ) Internal
  - Serves internal traffic.
  - Has private IP addresses.
  - DNS name is publicly resolvable.
  - Compatible with the **IPv4** and **Dualstack** IP address types.

Load balancer IP address type | Info
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

- (●) IPv4
  Includes only IPv4 addresses.
- ( ) Dualstack
  Includes IPv4 and IPv6 addresses.
- ( ) Dualstack without public IPv4
  Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with **internet-facing** load balancers only.

Network mapping Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC | Info
The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view target groups. For a new VPC, create a VPC.

my-vpc
vpc-03ad40e7bb181aaf9
IPv4 VPC CIDR: 12.0.0.0/16

IP pools - *new* | Info
You can optionally choose to configure an IPAM pool as the preferred source for your load balancers IP addresses. Create or view **Pools** in Amazon VPC IP Address Manager console.

[ ] Use IPAM pool for public IPv4 addresses
The IPAM pool you choose will be the preferred source of public IPv4 addresses. If the pool is depleted IPv4 addresses will be assigned by AWS.

Availability Zones and subnets | Info
Select at least two Availability Zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

[x] **us-east-1d (use1-az1)**
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.

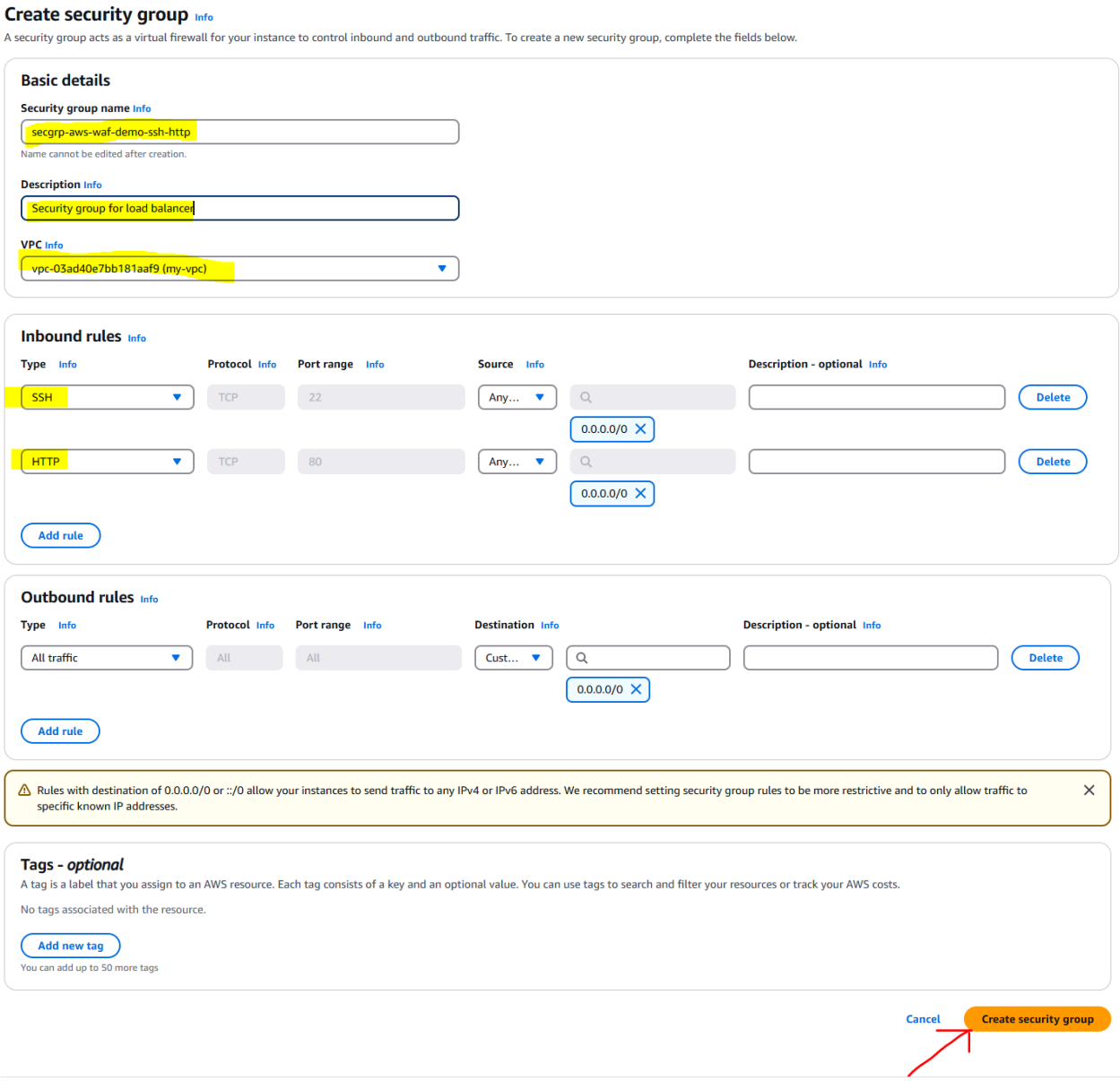
subnet-0d5c1eef5f4a55284 — Public-Subnet
IPv4 subnet CIDR: 12.0.2.0/24

⚠ The selected subnet does not have a route to an internet gateway. This means that your load balancer will not receive internet traffic.
You can proceed with this selection; however, for internet traffic to reach your load balancer, you must update the subnet's route table in the VPC console.

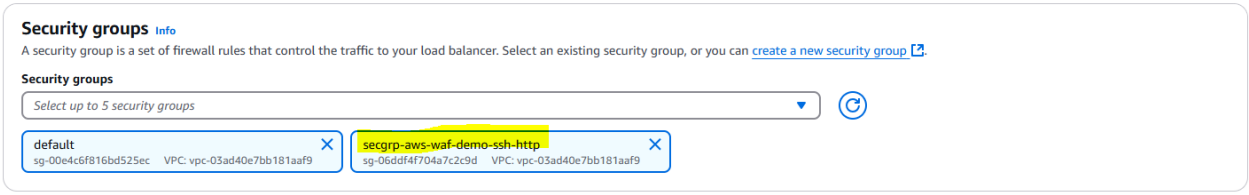
[x] **us-east-1f (use1-az5)**
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.

subnet-0b0c2fd01505ebbc7 — Public-Subnet
IPv4 subnet CIDR: 12.0.1.0/24

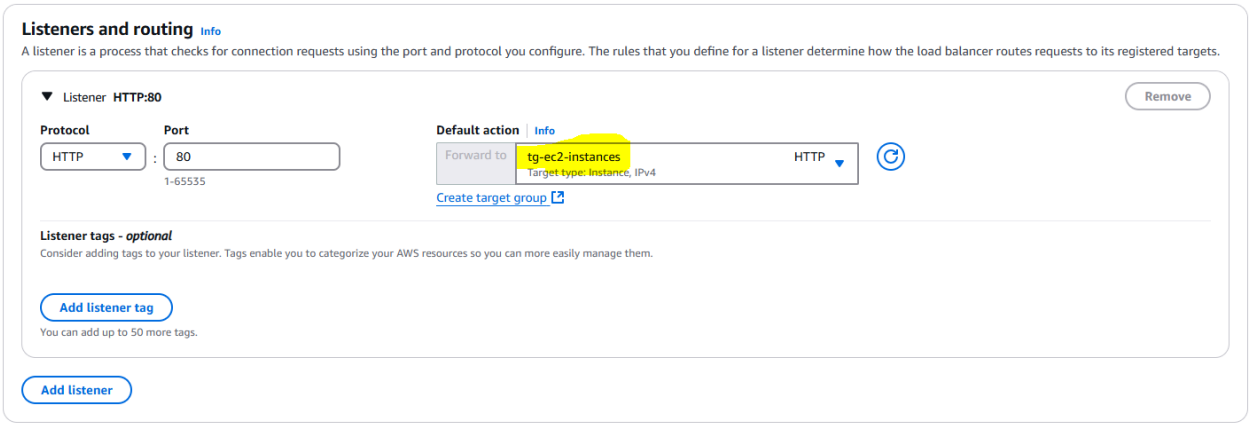
e. Create a security group for the load balancer and add the respective SSH and HTTP Inbound rules and click on “Create security group”.



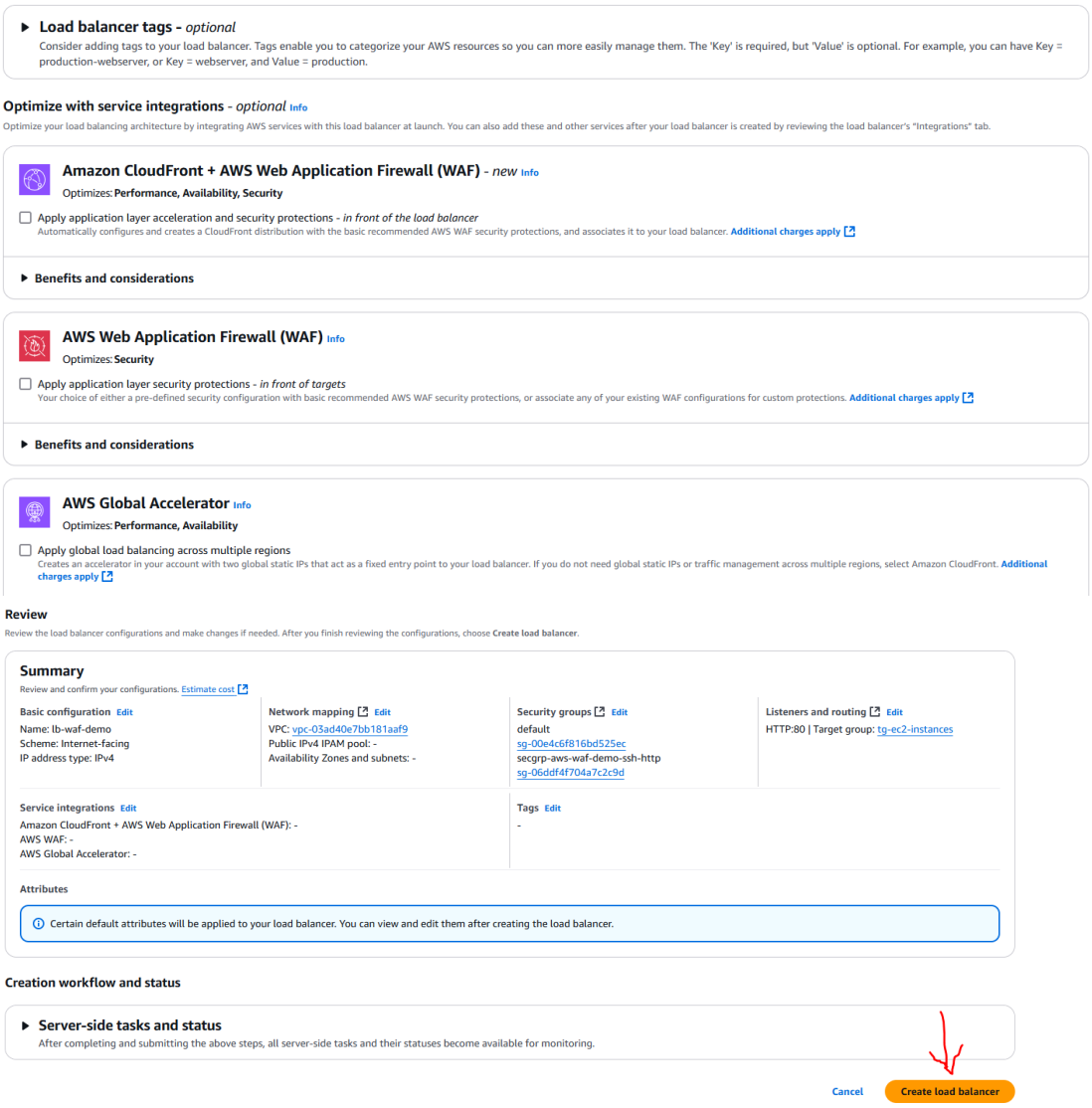
f. Attach the security group to the load balancer.



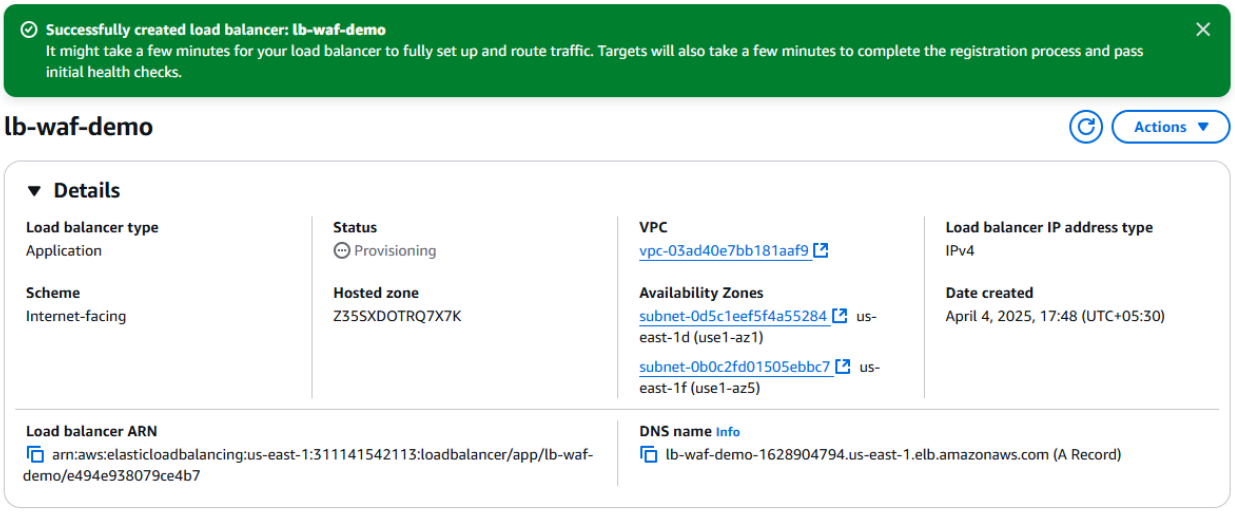
g. Listeners and routing (Important)



h. The rest of the details stay the same. Click on “Create load balancer”.

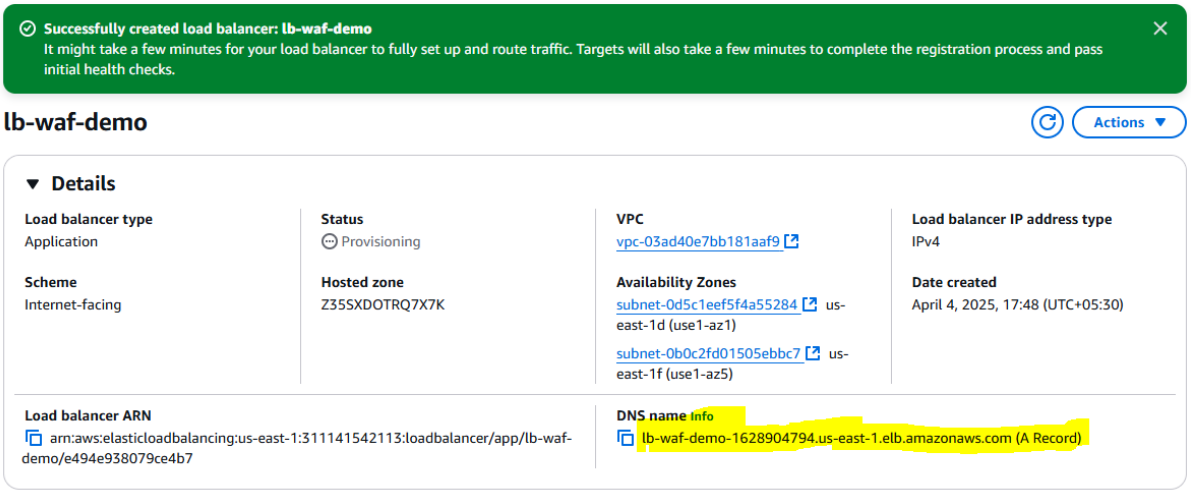


i. Load Balancer created.

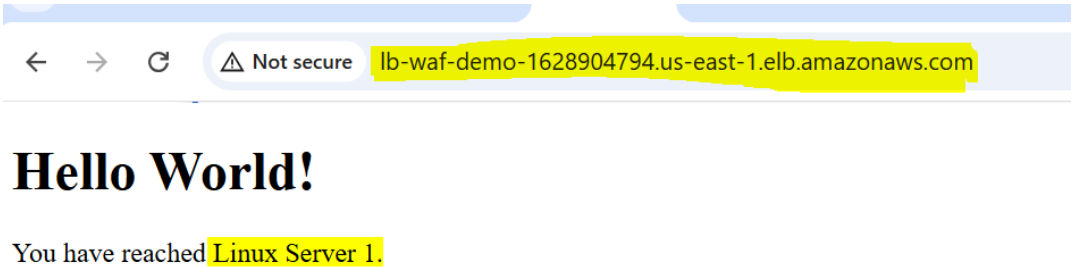


10. Evaluate the load balancer DNS.

a. Copy the DNS name and open it in a browser**(Use HTTP)**

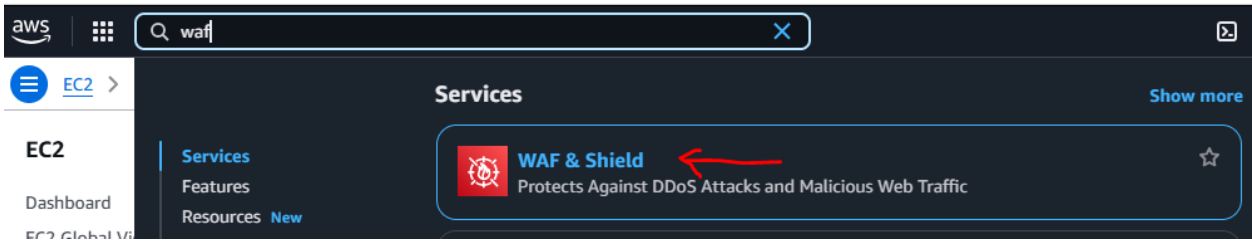


b. The web browser should show a result from one of the EC2s like below screenshot.

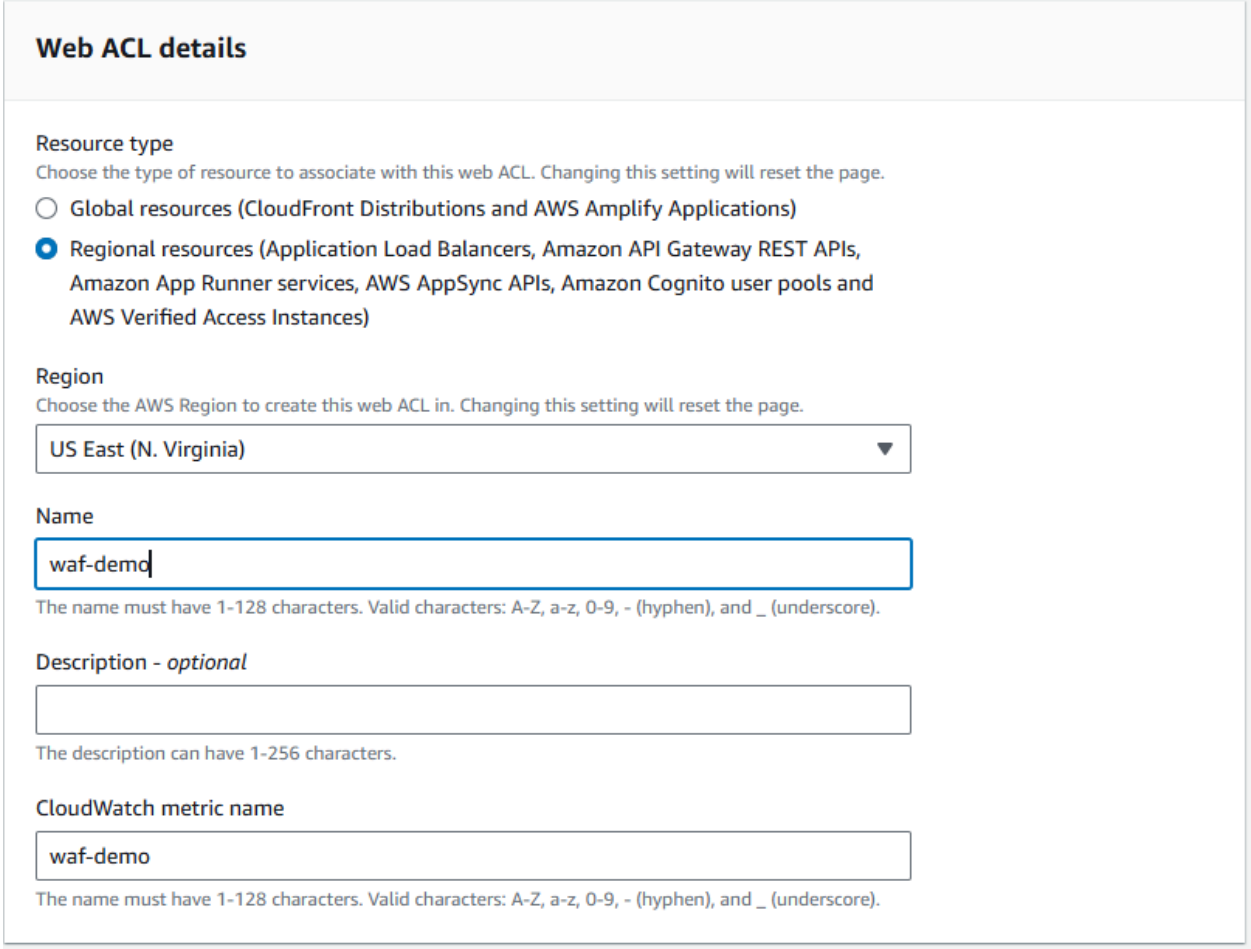


11. AWS WAF

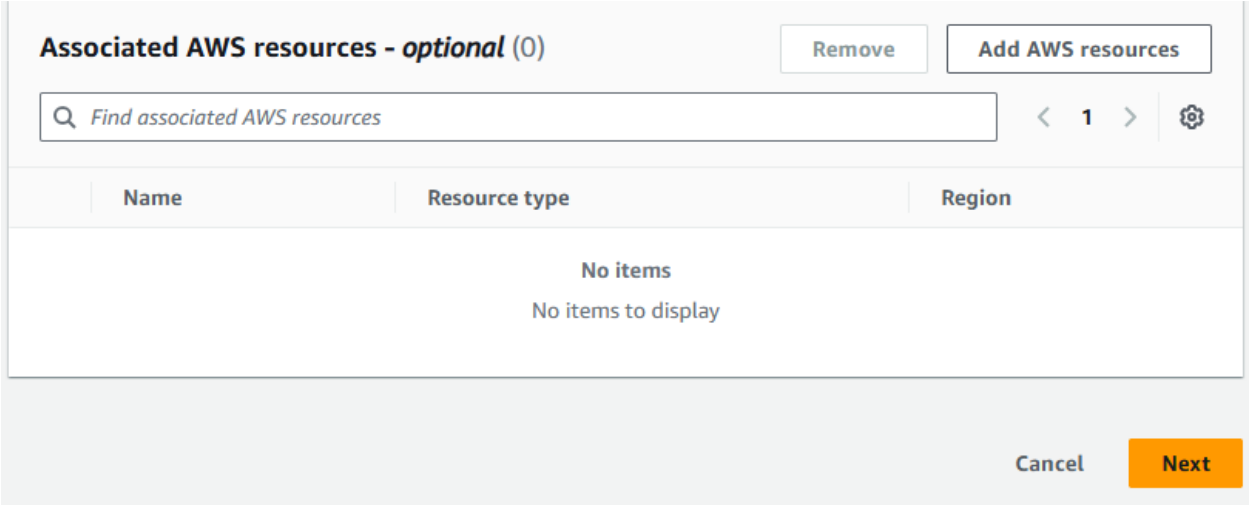
a. Type WAF in the search bar.

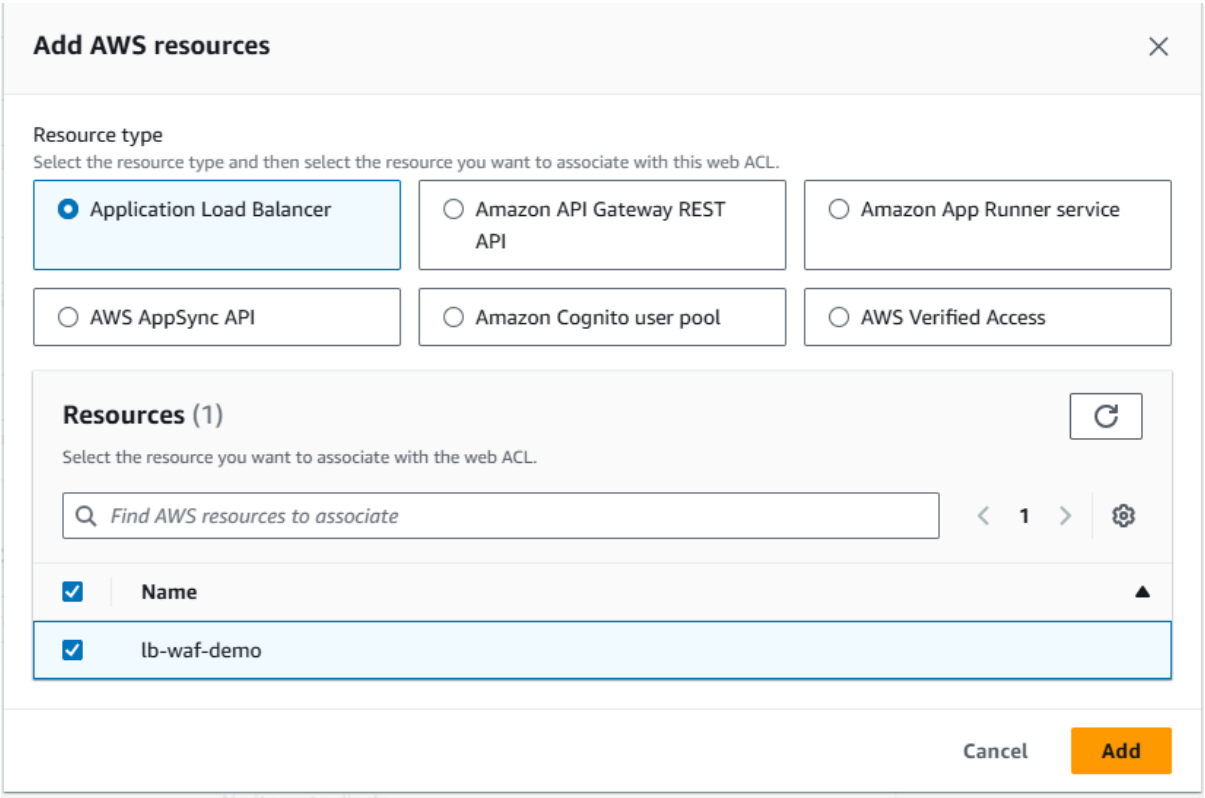


b. Make sure you select the correct region.

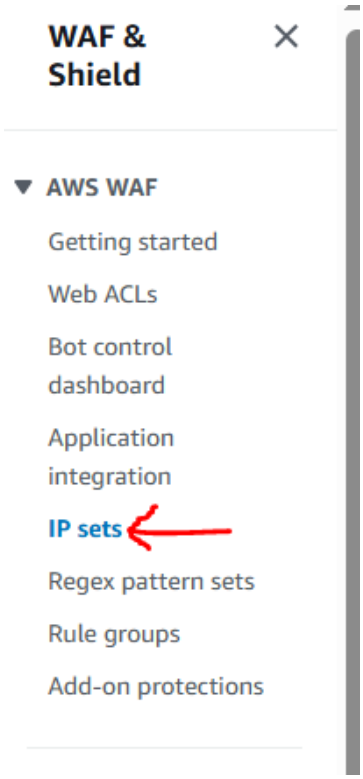


c. Attach Load Balancer by clicking on “Add AWS resources”

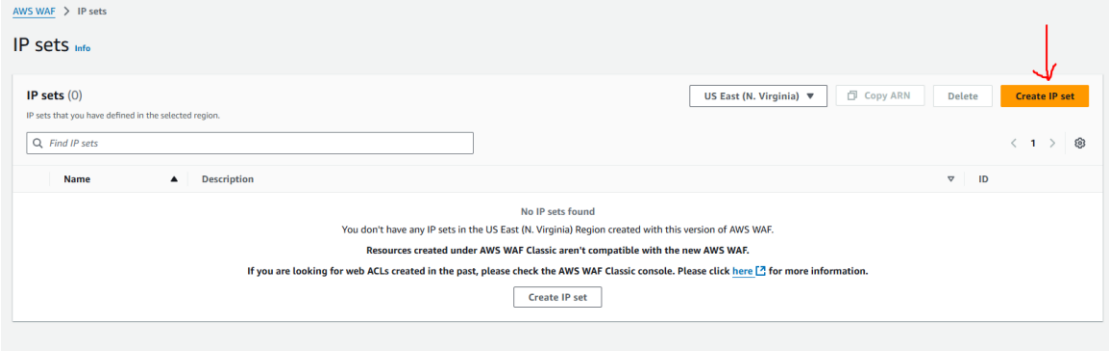




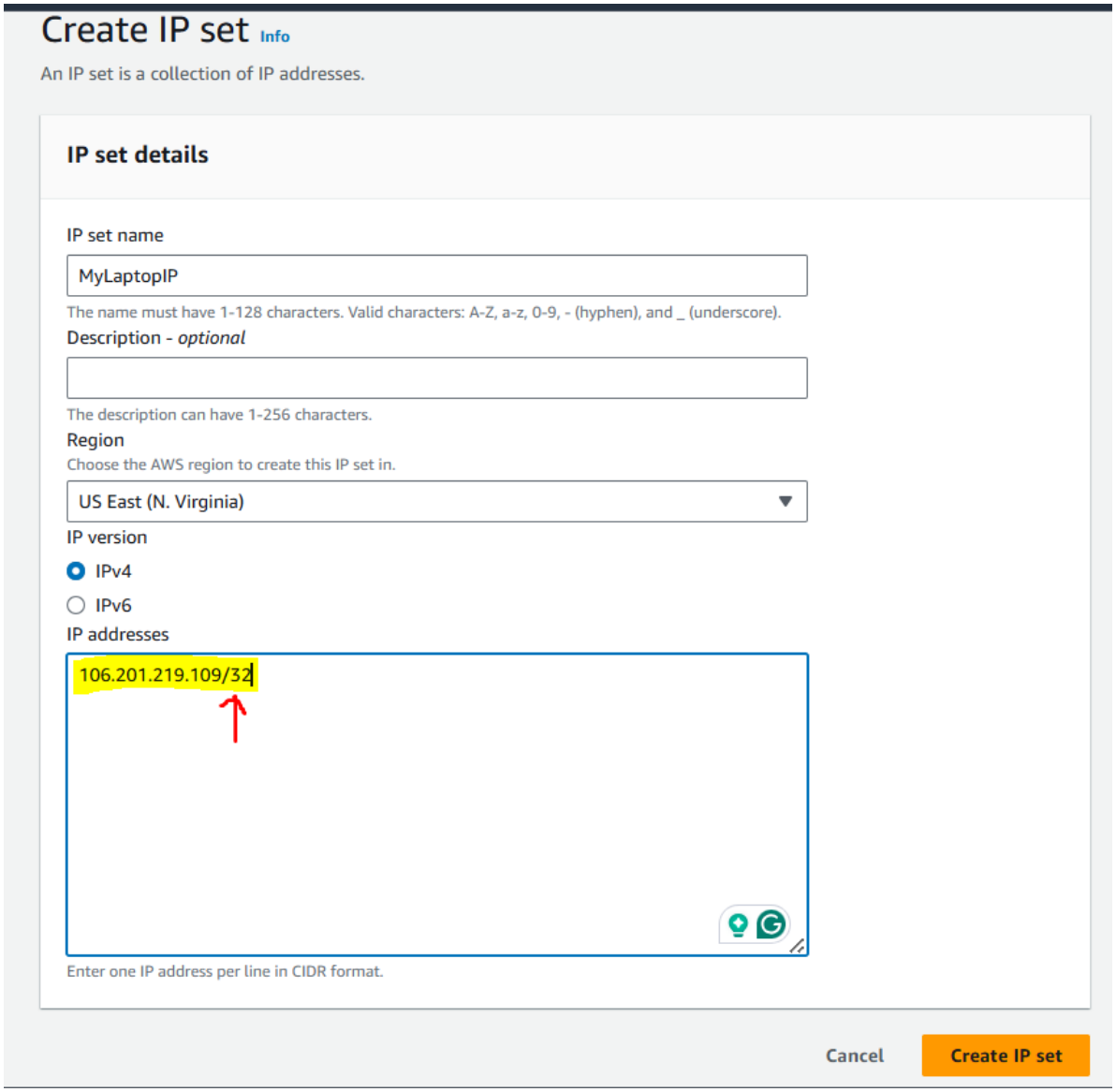
d. Next step is to add rules, but before doing this create IP sets.
Go to left panel and open IP sets in a **new TAB**



e. Click on the “Create IP set” button.

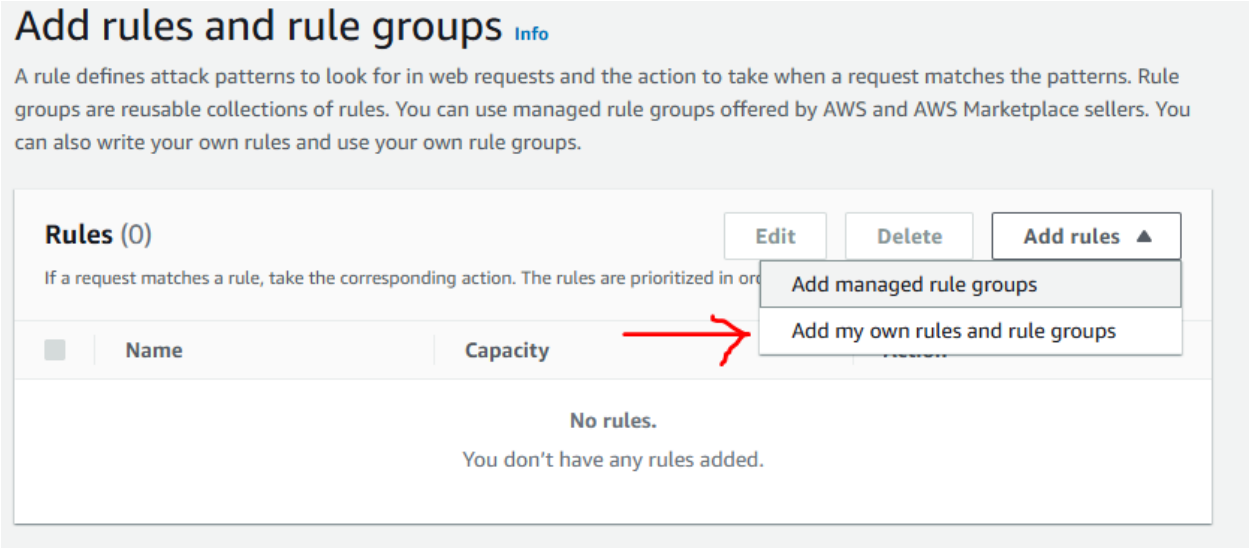


f. Add the IP you want to BLOCK. Do not forget to add the range as highlighted. Click on “Create IP set”.



g. Time to add rules.
   Go back to the Add rules and rule groups page.
   Click on “add rules” drop down.

Select “Add my own rules and rule groups”



h. Enter the highlighted details. Click on “add rule”.

Add my own rules and rule groups Info

Rule type

Rule type

- (●) **IP set**
  Use IP sets to identify a specific list of IP addresses.
- ( ) **Rule builder**
  Use a custom rule to inspect for patterns including query strings, headers, countries, and rate limit violations.
- ( ) **Rule group**
  Use a rule group to combine rules into a single logical set.

Rule

Name

BlockmyLaptop

IP set

IP set

MyLaptopIP

IP address to use as the originating address

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

Action

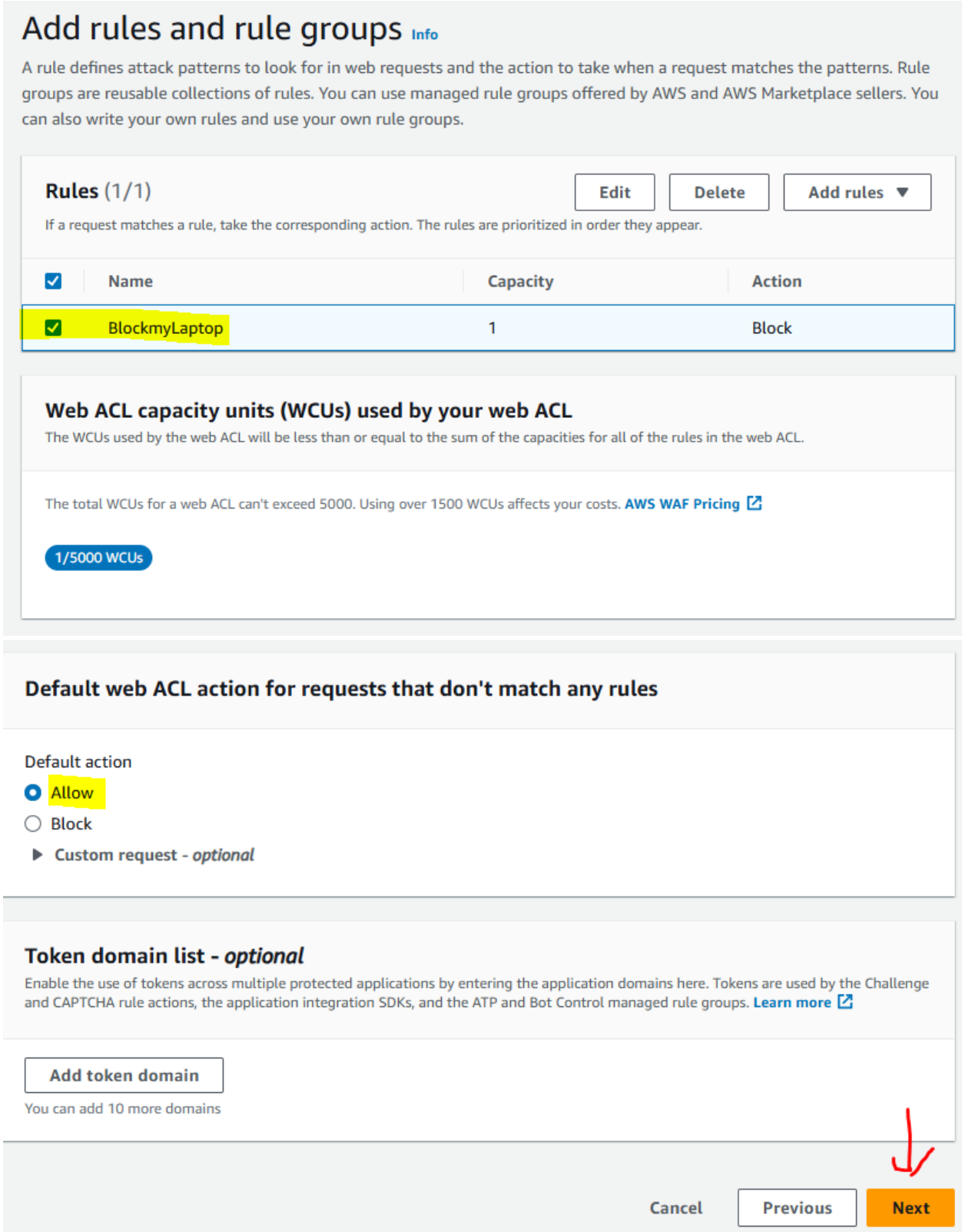
Choose an action to take when a request originates from one of the IP addresses in this IP set.

- ( ) Allow
- (●) Block
- ( ) Count
- ( ) CAPTCHA customize
- ( ) Challenge

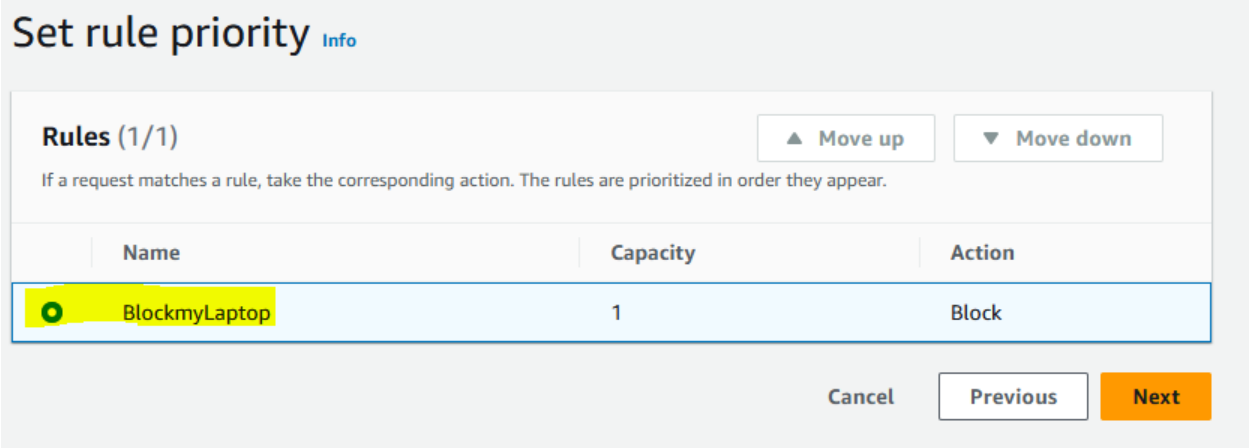
▶ Custom response - *optional*

Cancel | Add rule

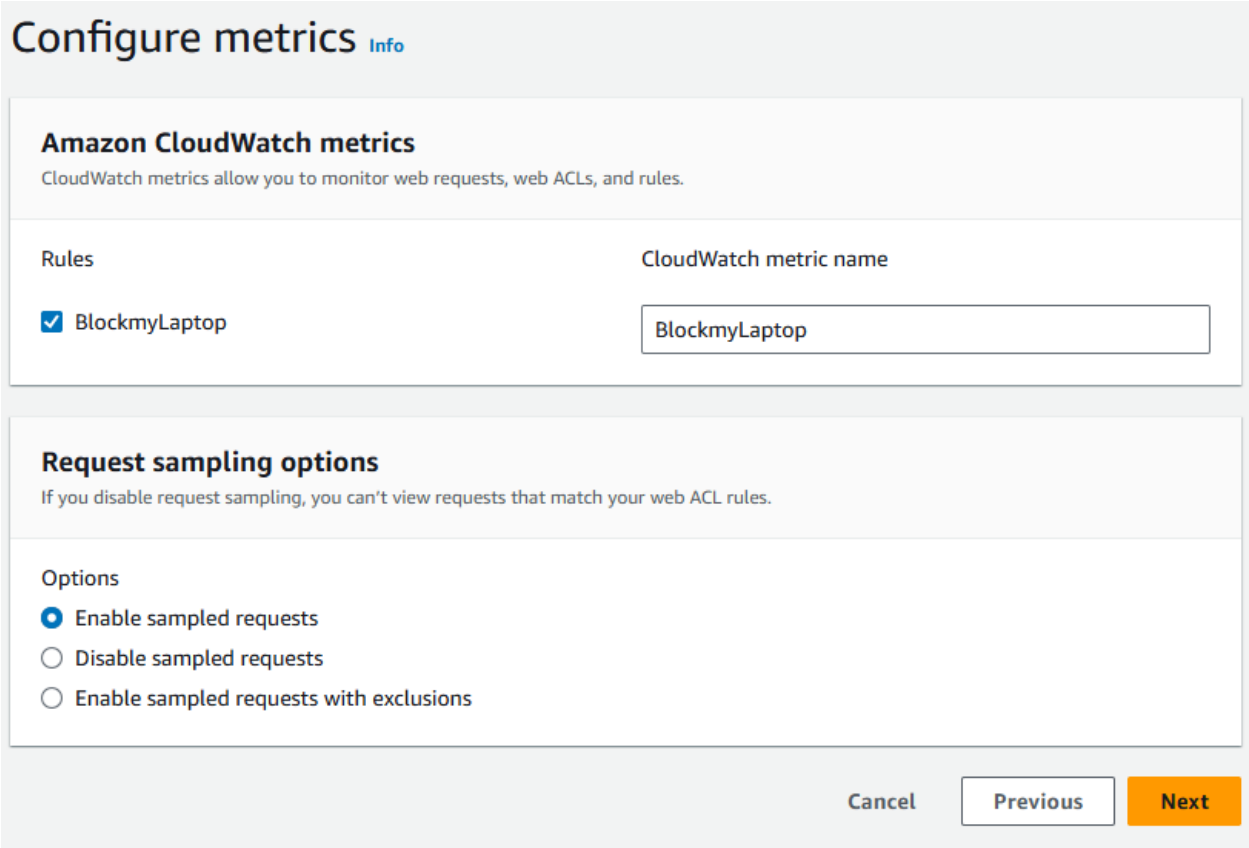
i. Check the newly created rule. Click on “Next”.



j. Set rule priority. Click on “Next”



k. Leave “Configure metrics” as is.



l. Leave “Review and create web ACL” as is and click on “Create web ACL”

Review and create web ACL Info

Step 1: Describe web ACL and associate it to AWS resources

Edit step 1

Web ACL details

Name
waf-demo

Description

CloudWatch metric name
waf-demo

Scope
REGIONAL

Region
us-east-1

Steps 2 and 3: Add rules and set rule priority

Edit steps 2 and 3

Rules (1)

If a request matches a rule, take the corresponding action. The rules are prioritized in order they appear.

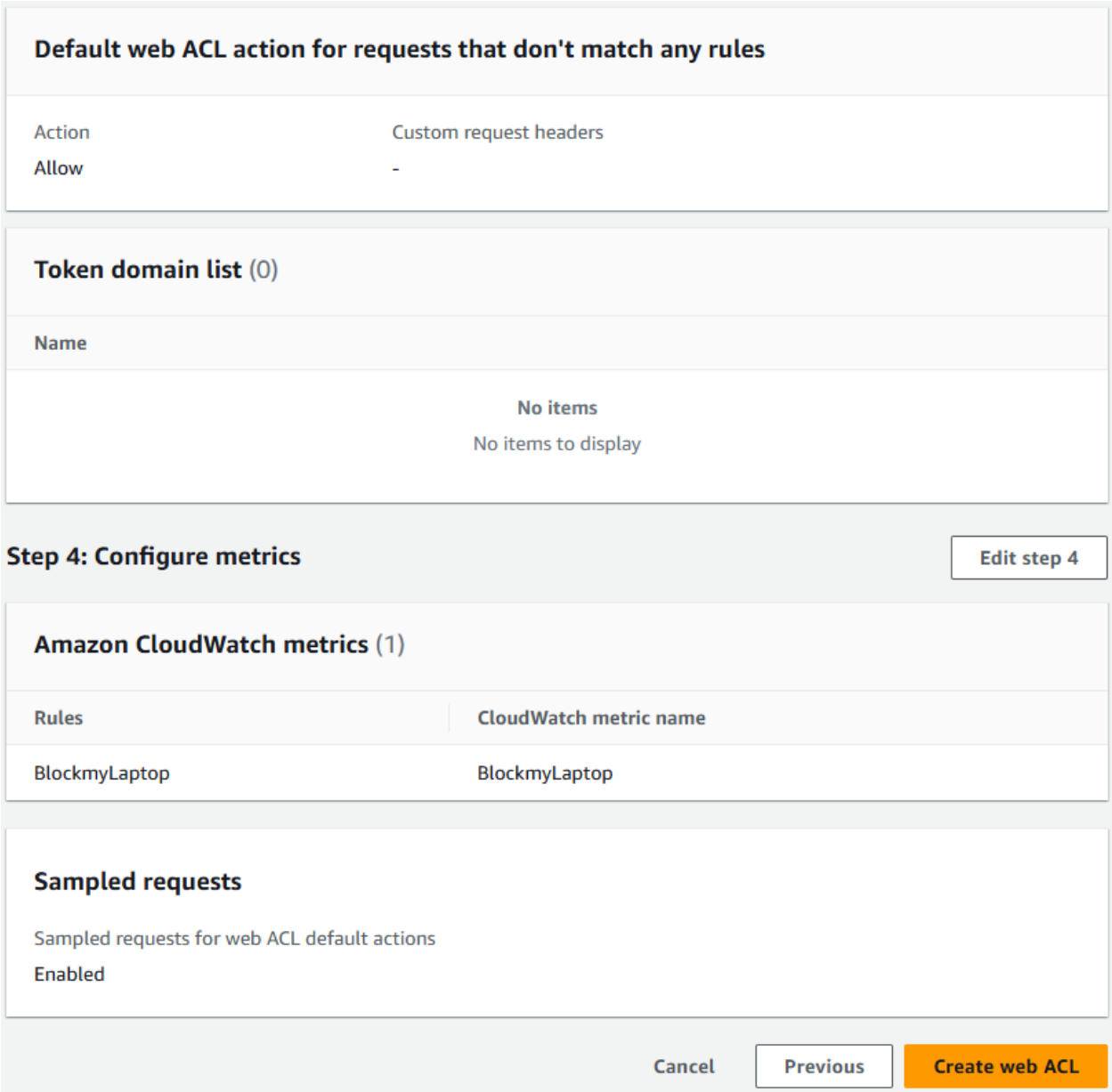
Name	Capacity	Action
BlockmyLaptop	1	Block

Web ACL capacity units (WCUs) used by your web ACL

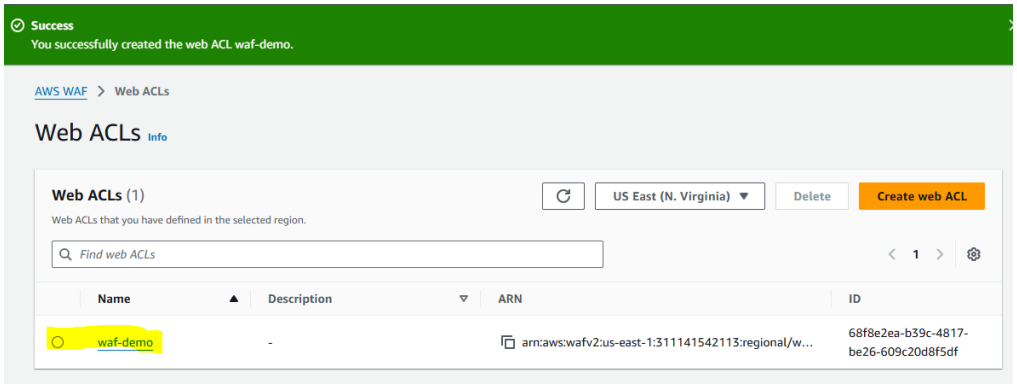
The WCUs used by the web ACL will be less than or equal to the sum of the capacities for all of the rules in the web ACL.

The total WCUs for a web ACL can't exceed 5000. Using over 1500 WCUs affects your costs. AWS WAF Pricing

1/5000 WCUs

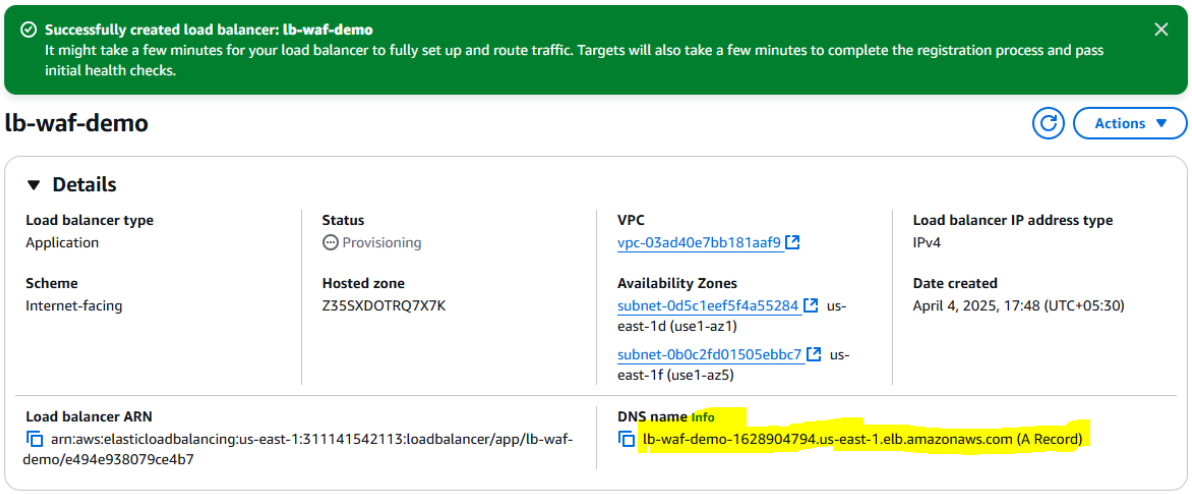


m. Web ACL should be successfully created.

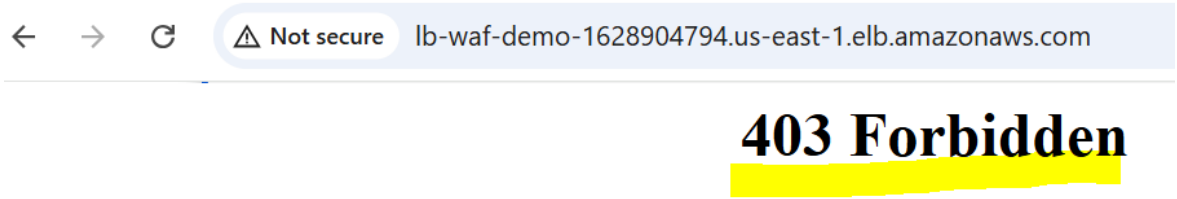


12. Evaluate the load balancer DNS again.

a. Copy the DNS name and open it in a browser**(Use HTTP)**



b. The web browser should show “403 Forbidden” error like the screenshot below.



13. WAF Testing Complete.

a. Similarly, you can allow IP addresses, set up CAPTCHA, calculate the count of requests or set up challenges in the rules.

14. Try it yourself!

Change the rule in WAF to CAPTCHA and see what happens.

AWS SERVICES USED

- **AWS VPC**
- **AWS EC2**
- **AWS WAF**