AWS WAF – Web Application Firewall

AWS WAF is a web application firewall that helps protect your web applications or APIs against common web exploits and bots that may affect availability, compromise security, or consume excessive resources. AWS WAF gives you control over how traffic reaches your applications by enabling you to create security rules that control bot traffic and block common attack patterns, such as SQL injection or cross-site scripting. You can also customize rules that filter out specific traffic patterns. You can get started quickly using Managed Rules for AWS WAF, a pre-configured set of rules managed by AWS or AWS Marketplace Sellers to address issues like the OWASP Top 10 security risks and automated bots that consume excess resources, skew metrics, or can cause downtime. These rules are regularly updated as new issues emerge. AWS WAF includes a full-featured API that you can use to automate the creation, deployment, and maintenance of security rules.

AWS WAF Overview

AWS Web Application Firewall (WAF) is a security tool that helps you to protect the application against web attacks. WAF monitors and controls unusual bot traffic, blocks common attack patterns, such as **SQL Injection or Cross-site scripting**, etc. It also lets you monitor the HTTP and HTTPS requests that are forwarded to an Amazon API Gateway API, Amazon CloudFront or an Application Load Balancer.

- Amazon WAF allows you to control your content by using an IP address from where the request originates.
- Three things make Amazon WAF work – **Access control lists (ACL), Rules and Rule Group**.
- Amazon WAF manages Web ACL capacity units (WCU) for rules, rule groups and web ACLs.
- Amazon WAF includes a full-featured API that you can use to automate the creation, deployment, and maintenance of security rules.

Common Web Attacks

Before protecting your applications, you need to know the most common web attacks mention below.

DDoS(Denial-Of-Service) attacks: This is probably the most common attack. Attackers overload an application by sending bulk requests to the web servers. Thousands of hosts infected with malware are used in this attack, which utilizes more than one unique IP address or machine. This slows down the application and significantly hurt the value of a brand.
SQL injections: SQL injection is a code injection procedure that might destroy your SQL database. Attackers can run malicious SQL queries on your web applications.
Cross-Site Scripting: If your application is vulnerable to cross-site scripting, then the attacker can run or inject malicious scripts, generally in the form of a browser side script. These scripts can even rewrite the content of the HTML pages.

AWS WAF Features

Amazon Web Application Firewall offer lots of features to its users mentioned below.

- **Protection Against Web Attacks:** With minimum latency impact on incoming traffic, AWS WAF offers many rules to inspect any element of a web request. AWS WAF protects web applications against threats by filtering traffic according to the rules created.
- **Establish Rules Accordingly:** AWS WAF is a versatile and valuable tool for protecting the infrastructures of applications. And this is because it allows users to establish rules according to their needs and vulnerabilities that they wish to stop. We can consider it as a great solution to protect any web applications environment at the enterprise level.
- **Web traffic filtering:** WAF allows users to create rules to filter web traffic. It filters IP addresses, HTTP headers, HTTP body, or URI strings from a web request.
- **Flexible Integration With AWS Services:** AWS Web Application Firewall offers easy integration with other AWS services like Amazon EC2, CloudFront, Load balancer etc.

- **Monitor Rules:** AWS Web Application Firewall allows us to create rules and review and customize them to prevent unknown attracts.

Benefits

Agile protection against web attacks
AWS WAF rule propagation and updates take under a minute, enabling you to quickly update security across your environment when issues arise. WAF supports hundreds of rules that can inspect any part of the web request with minimal latency impact to incoming traffic. AWS WAF protects web applications from attacks by filtering traffic based on rules that you create. For example, you can filter any part of the web request, such as IP addresses, HTTP headers, HTTP body, or URI strings. This allows you to block common attack patterns, such as SQL injection or cross-site scripting.

Save time with managed rules
With Managed Rules for AWS WAF, you can quickly get started and protect your web application or APIs against common threats. You can select from many rule types, such as ones that address issues like the Open Web Application Security Project (OWASP) Top 10 security risks, threats specific to Content Management Systems (CMS), or emerging Common Vulnerabilities and Exposures (CVE). Managed rules are automatically updated as new issues emerge, so that you can spend more time building applications.

Improved web traffic visibility
AWS WAF gives near real-time visibility into your web traffic, which you can use to create new rules or alerts in Amazon CloudWatch. You have granular control over how the metrics are emitted, allowing you to monitor from the rule level to the entire inbound traffic. In addition, AWS WAF offers comprehensive logging by capturing each inspected web request’s full header data for use in security automation, analytics, or auditing purposes.

Ease of deployment & maintenance

AWS WAF is easy to deploy and protect applications deployed on either Amazon CloudFront as part of your CDN solution, the Application Load Balancer that fronts all your origin servers, Amazon API Gateway for your REST APIs, or AWS AppSync for your GraphQL APIs. There is no additional software to deploy, DNS configuration,

SSL/TLS certificate to manage, or need for a reverse proxy setup. With AWS Firewall Manager integration, you can centrally define and manage your rules, and reuse them across all the web applications that you need to protect.

Easily monitor, block, or rate-limit bots

With AWS WAF Bot Control, you get visibility and control over common and pervasive bot traffic to your applications. Within the AWS WAF console, you can monitor common bots, such as status monitors and search engines, and get detailed, real-time visibility into the category, identity, and other details of bot traffic. You can also block, or rate-limit, traffic from pervasive bots, such as scrapers, scanners, and crawlers. Using AWS Firewall Manager, you can deploy the Bot Control managed rule group across multiple accounts in your AWS Organization.

Security integrated with how you develop applications

Every feature in AWS WAF can be configured using either the AWS WAF API or the AWS Management Console. This allows your DevOps team to define application-specific rules that increase web security as they develop applications. This lets you put web security at multiple points in the development process chain, from the hands of the developer initially writing code, to the DevOps engineer deploying software, to the security administrators enforcing a set of rules across the organization.

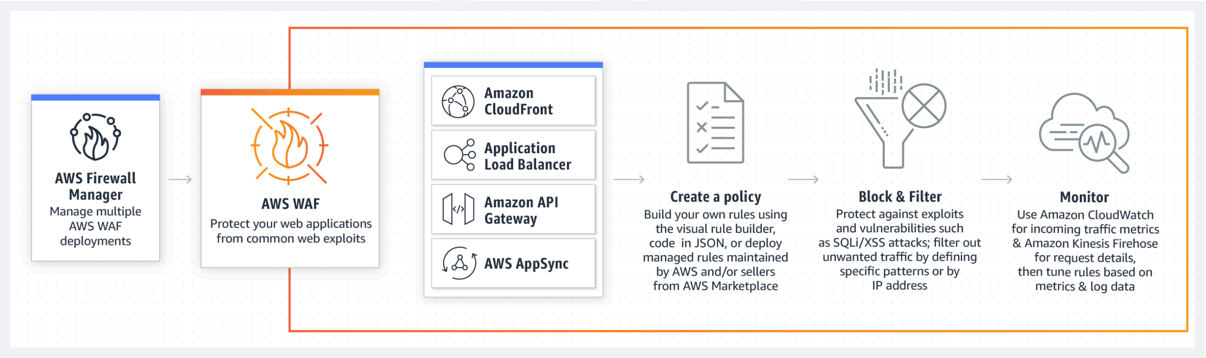
How It Works

AWS Web Application Firewall protect the applications from malicious attacks. Working of WAF mentioned below.

- **AWS Firewall Manage:** It Manages multiple AWS Web Application Firewall Deployments
- **AWS WAF:** Protect deployed application from common web exploits.

- **Create a Policy:** Now you can build your own rules using the visual rule builder.
- **Block Filter:** Block filter protect against exploits and vulnerabilities attacks.

- **Monitor:** Use Amazon CloudWatch for incoming traffic metrics & Amazon kinesis firehose for request details, then tune rules based on metrics and log data.


AWS Firewall Manager
Manage multiple AWS WAF deployments
AWS WAF
Protect your web applications from common web exploits
Amazon CloudFront
Application Load Balancer
Amazon API Gateway
AWS AppSync
Create a policy
Monitor
Block & Filter

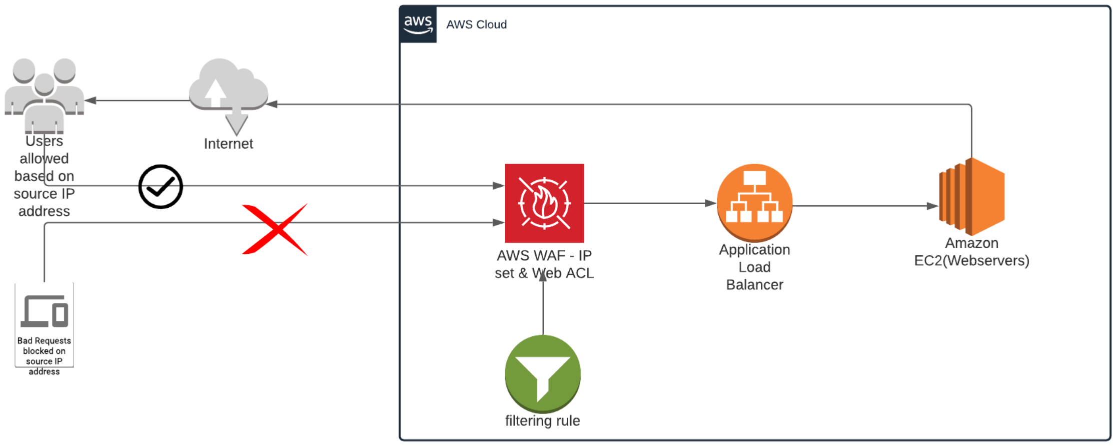
Build your own rules using the visual rule builder, code in JSON, or deploy managed rules maintained by AWS and/or sellers from AWS Marketplace

Protect against exploits and vulnerabilities such as SQLi/XSS attacks; filter out unwanted traffic by defining specific patterns or by IP address

Use Amazon CloudWatch for incoming traffic metrics & Amazon Kinesis Firehose for request details, then tune rules based on metrics & log data

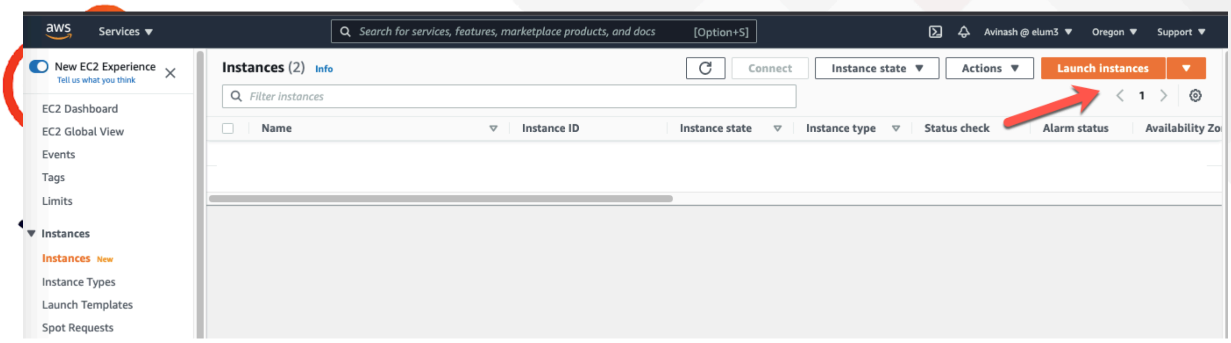
Demo : Control Web Traffic using Web Application Firewall (WAF)

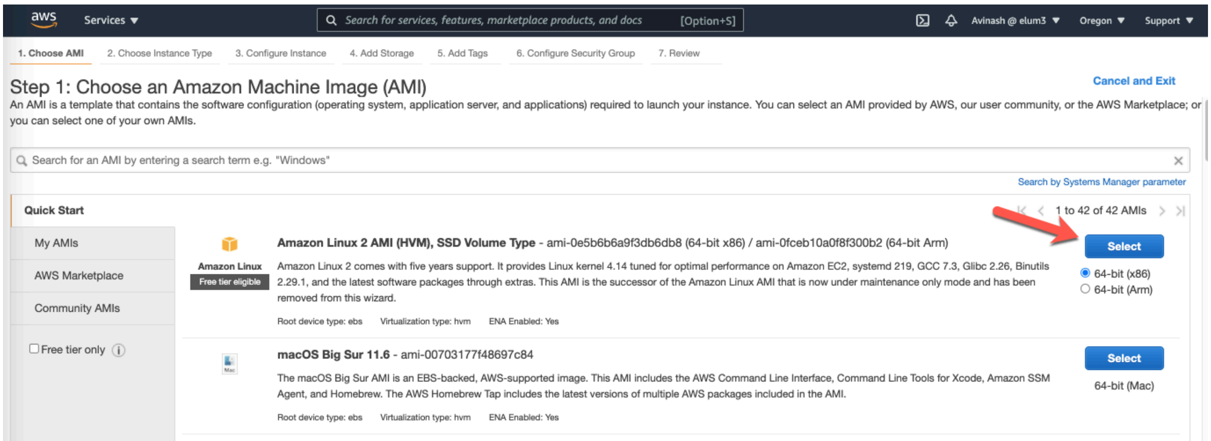
Step 1 : Launch 2 webservers in 2 different Az's
Step 2 : Create Application Load Balancer
Step 3 : Create IP set in WAF
Step 4 : Create Web ACL in WAF
Step 5 : Test the working on WAF


AWS Cloud
Users allowed based on source IP address
Internet
Bad Requests blocked on source IP address
AWS WAF - IP set & Web ACL
Application Load Balancer
Amazon EC2(Webservers)
filtering rule

Step 1 : Launch 2 webservers in 2 different Az's

- Navigate to AWS EC2 console





Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

□ Free tier only

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0e5b6b6a9f3db6db8 (64-bit x86) / ami-0fceb10a0f8f300b2 (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

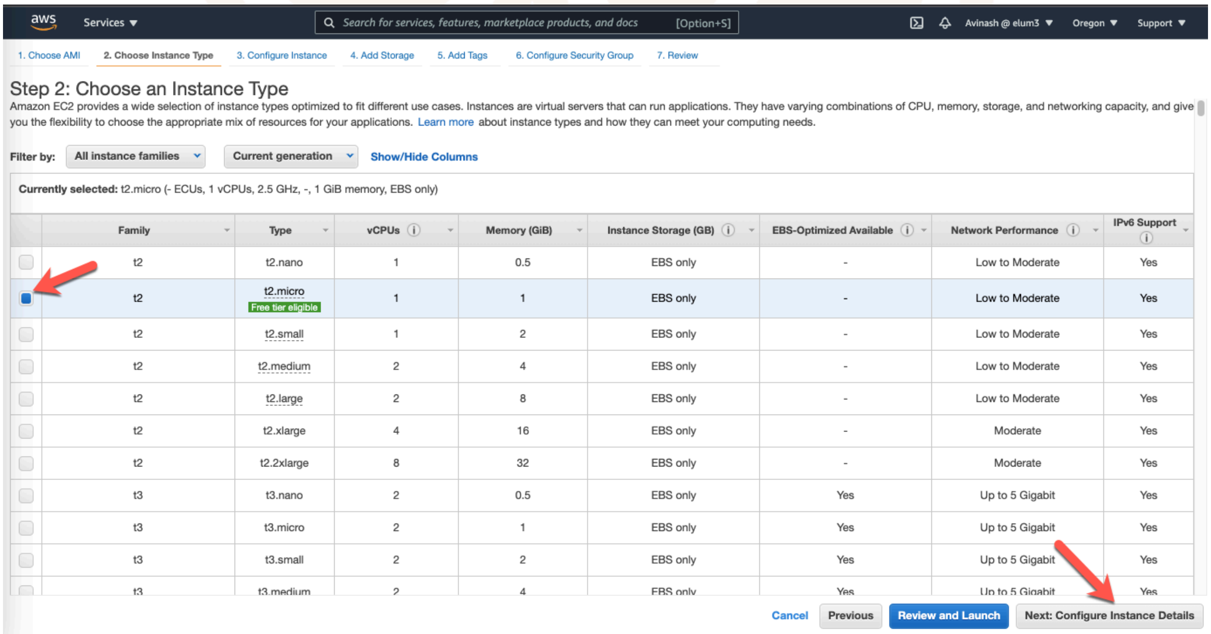
● 64-bit (x86)
○ 64-bit (Arm)

macOS Big Sur 11.6 - ami-00703177f48697c84

The macOS Big Sur AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

64-bit (Mac)



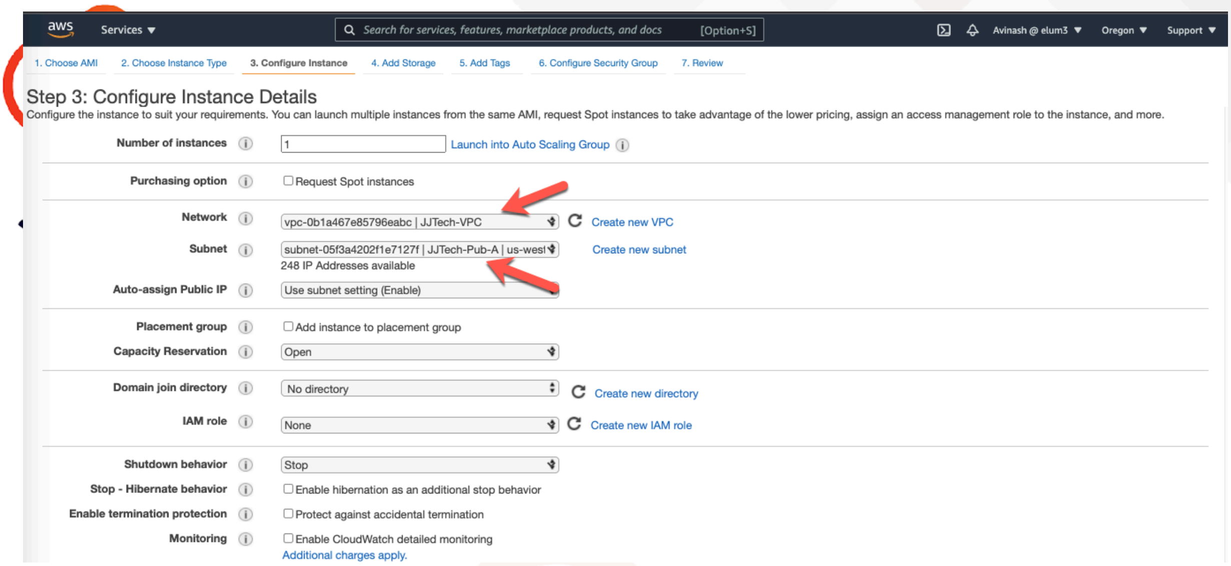
Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance families Current generation

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

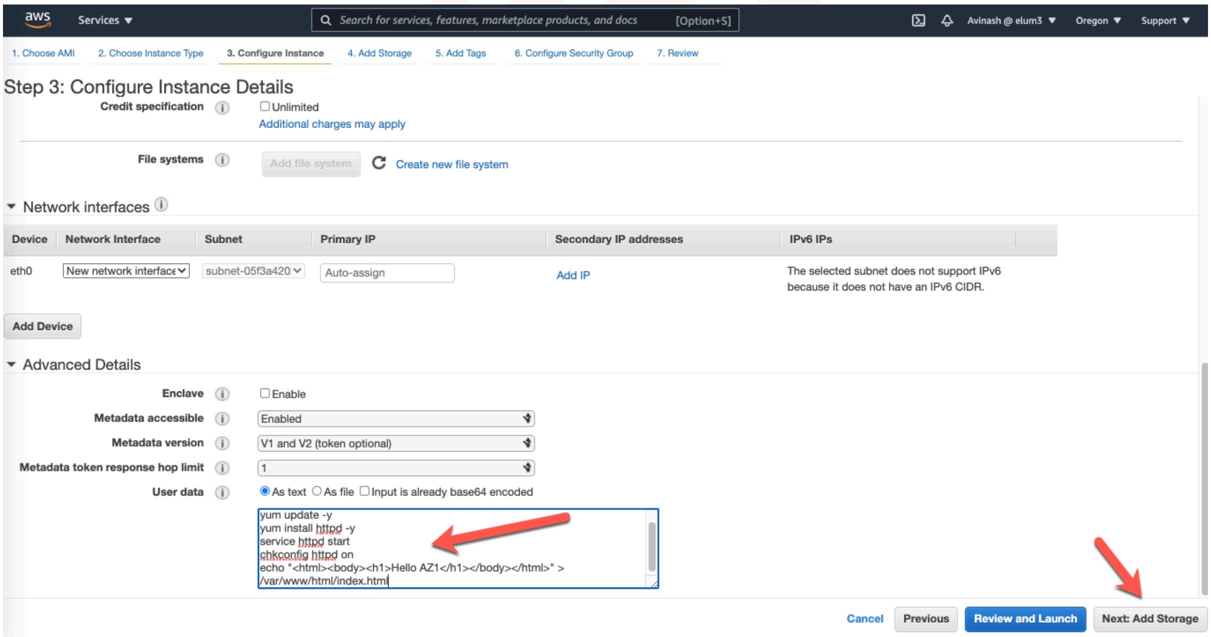
	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
□	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
☑	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
□	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
□	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
□	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
□	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
□	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
□	t3	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
□	t3	t3.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
□	t3	t3.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1	Launch into Auto Scaling Group
Purchasing option	□ Request Spot instances	
Network	vpc-0b1a467e85796eabc | JJTech-VPC	Create new VPC
Subnet	subnet-05f3a4202f1e7127f | JJTech-Pub-A | us-west 248 IP Addresses available	Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)	
Placement group	□ Add instance to placement group	
Capacity Reservation	Open	
Domain join directory	No directory	Create new directory
IAM role	None	Create new IAM role
Shutdown behavior	Stop	
Stop - Hibernate behavior	□ Enable hibernation as an additional stop behavior	
Enable termination protection	□ Protect against accidental termination	
Monitoring	□ Enable CloudWatch detailed monitoring Additional charges apply.	



Step 3: Configure Instance Details

Credit specification	□ Unlimited Additional charges may apply	
File systems	Add file system	Create new file system

Network interfaces

Device	Network Interface	Subnet	Primary IP	Secondary IP addresses	IPv6 IPs
eth0	New network interface	subnet-05f3a420	Auto-assign	Add IP	The selected subnet does not support IPv6 because it does not have an IPv6 CIDR.

Add Device

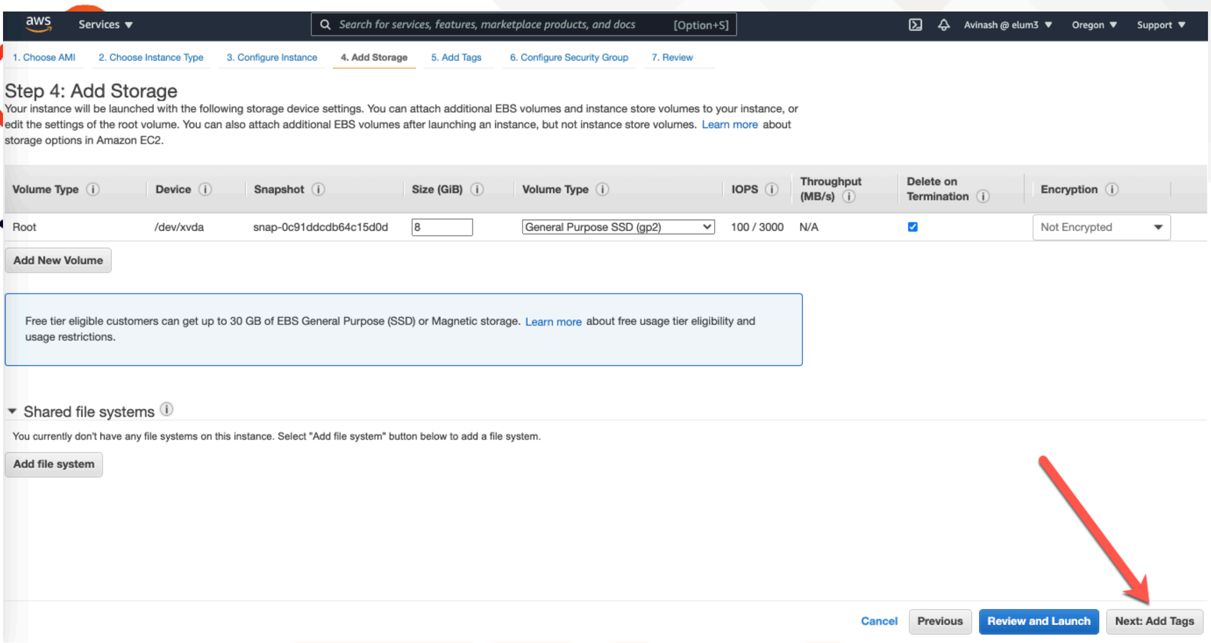
Advanced Details

Enclave	□ Enable
Metadata accessible	Enabled
Metadata version	V1 and V2 (token optional)
Metadata token response hop limit	1
User data	● As text ○ As file □ Input is already base64 encoded

```
yum update -y
yum install httpd -y
service httpd start
chkconfig httpd on
echo "<html><body><h1>Hello AZ1</h1></body></html>" >
/var/www/html/index.html
```

Userdata:

```
#!/bin/bash
yum update -y
yum install httpd -y
service httpd start
chkconfig httpd on
echo "<html><body><h1>Hello AZ1</h1></body></html>" >
/var/www/html/index.html
```



Step 4: Add Storage

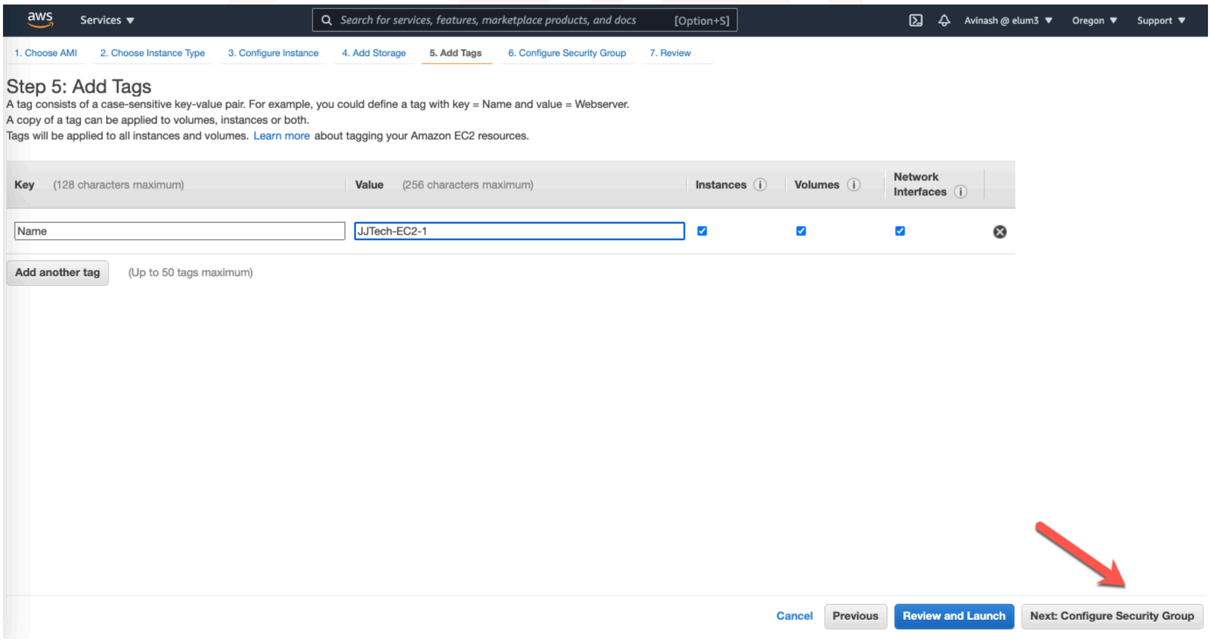
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-0c91ddcdb64c15d0d	8	General Purpose SSD (gp2)	100 / 3000	N/A	☑	Not Encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

Shared file systems

You currently don't have any file systems on this instance. Select "Add file system" button below to add a file system.

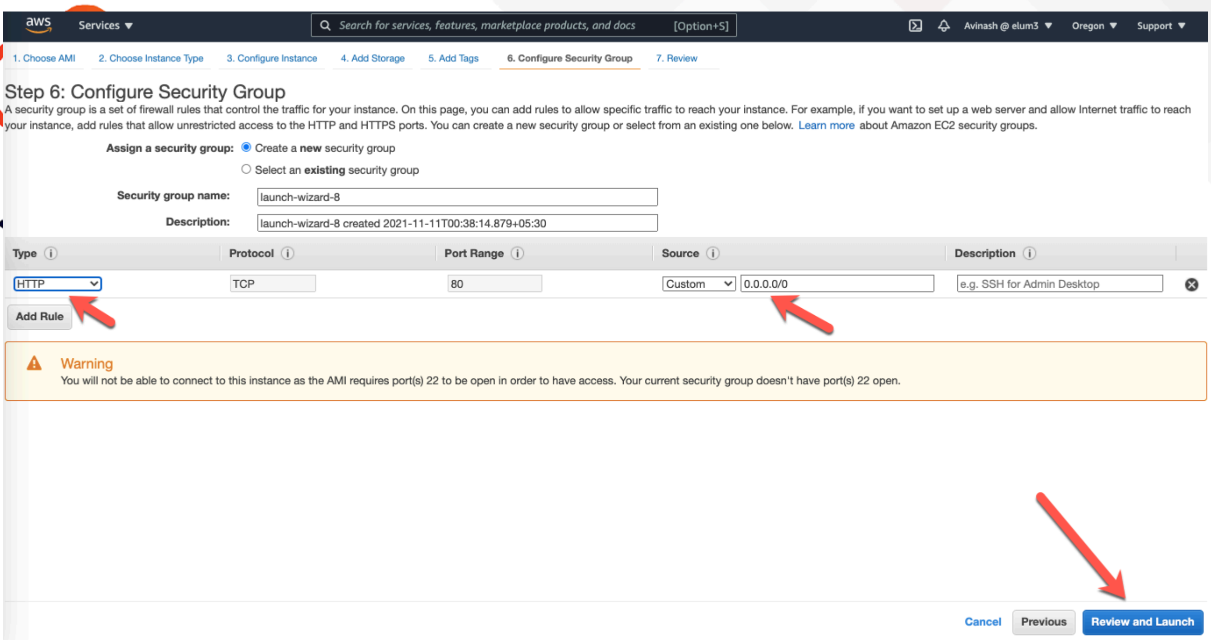


Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes	Network Interfaces	
Name	JJTech-EC2-1	☑	☑	☑	⊗

(Up to 50 tags maximum)



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

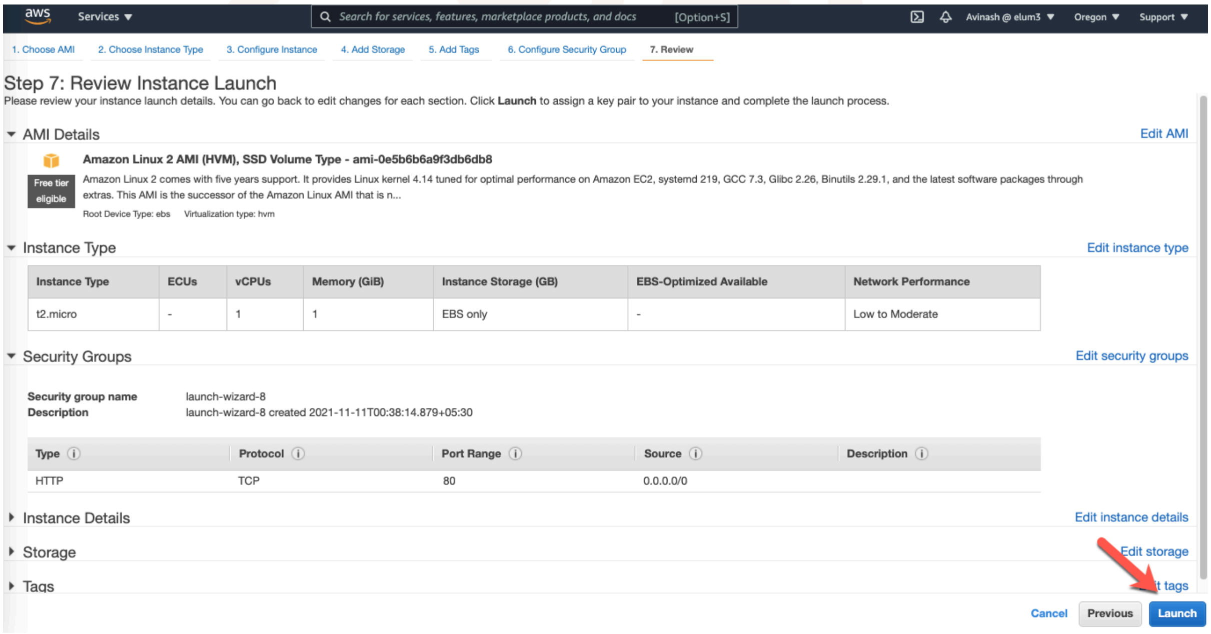
Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: launch-wizard-8

Description: launch-wizard-8 created 2021-11-11T00:38:14.879+05:30

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

⚠ Warning
You will not be able to connect to this instance as the AMI requires port(s) 22 to be open in order to have access. Your current security group doesn't have port(s) 22 open.



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

▼ AMI Details — Edit AMI

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0e5b6b6a9f3db6db8

Free tier eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is n...

Root Device Type: ebs Virtualization type: hvm

▼ Instance Type — Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

▼ Security Groups — Edit security groups

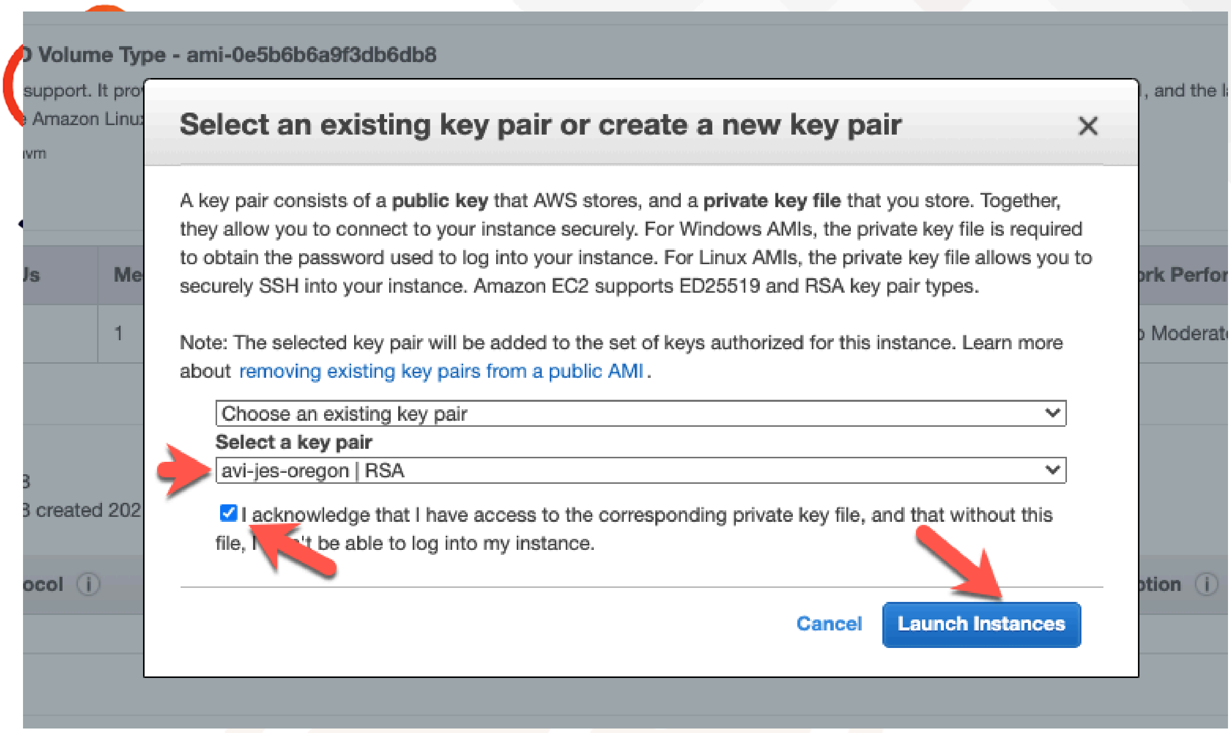
Security group name launch-wizard-8
Description launch-wizard-8 created 2021-11-11T00:38:14.879+05:30

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	

▸ Instance Details — Edit instance details

▸ Storage — Edit storage

▸ Tags — t tags



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

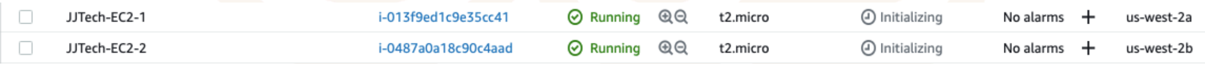
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Choose an existing key pair

Select a key pair: avi-jes-oregon | RSA

☑ I acknowledge that I have access to the corresponding private key file, and that without this file, I won't be able to log into my instance.

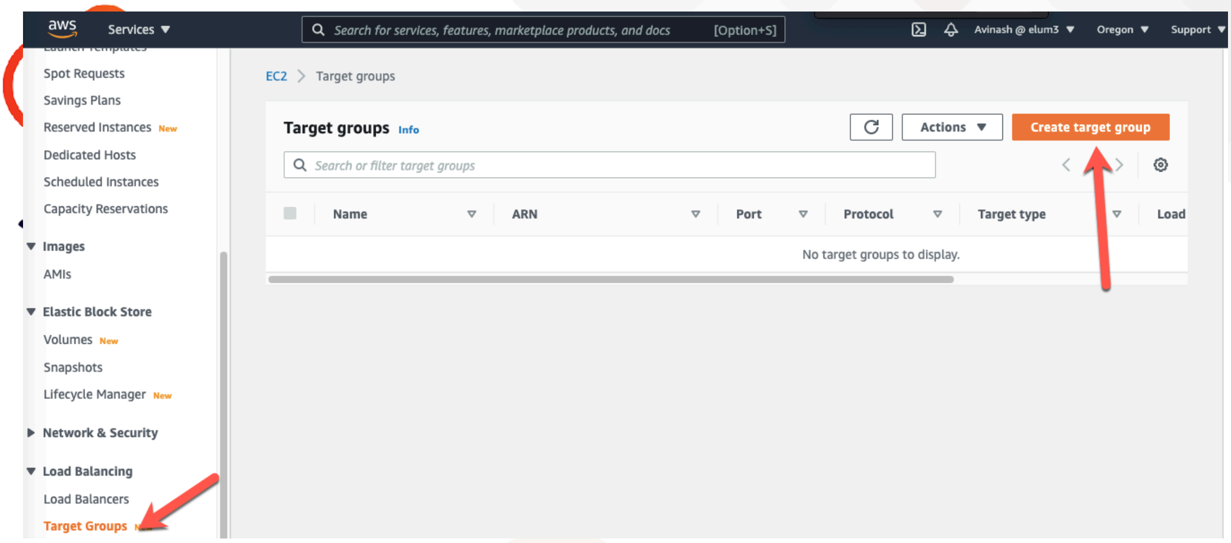
- Please create the second EC2 instance with same configuration and userdata in another AZ

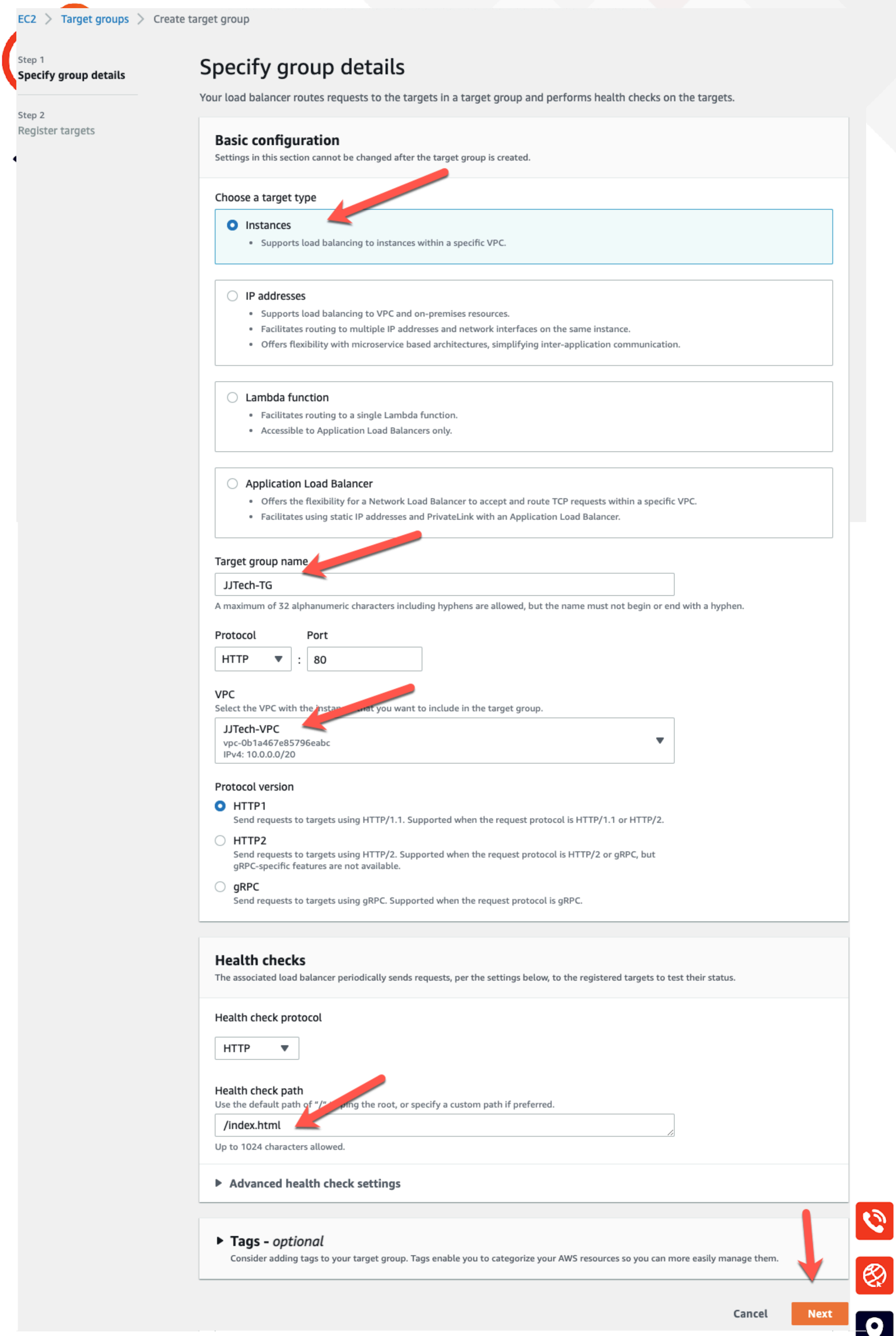


□	JJTech-EC2-1	i-013f9ed1c9e35cc41	Running		t2.micro	Initializing	No alarms	+	us-west-2a
□	JJTech-EC2-2	i-0487a0a18c90c4aad	Running		t2.micro	Initializing	No alarms	+	us-west-2b

Step 2 : Create Application Load Balancer

- Navigate to EC2 console TargetGroup


aws
Services ▼
Search for services, features, marketplace products, and docs [Option+S]
Avinash @ elum3 ▼
Oregon ▼
Support ▼
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
▼ Images
AMIs
▼ Elastic Block Store
Volumes New
Snapshots
Lifecycle Manager New
▶ Network & Security
▼ Load Balancing
Load Balancers
Target Groups
EC2 > Target groups
Target groups Info
Actions ▼
Create target group
Search or filter target groups
Name
ARN
Port
Protocol
Target type
Load
No target groups to display.



EC2 > Target groups > Create target group

Step 1
Specify group details

Step 2
Register targets

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 cannot be changed after the target group is created.

Choose a target type

● Instances
- Supports load balancing to instances within a specific VPC.

○ 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.

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

Protocol: HTTP ▼ : Port: 80

VPC
Select the VPC with the instances that you want to include in the target group.
JJTech-VPC
vpc-0b1a467e85796eabc
IPv4: 10.0.0.0/20 ▼

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.

Health checks

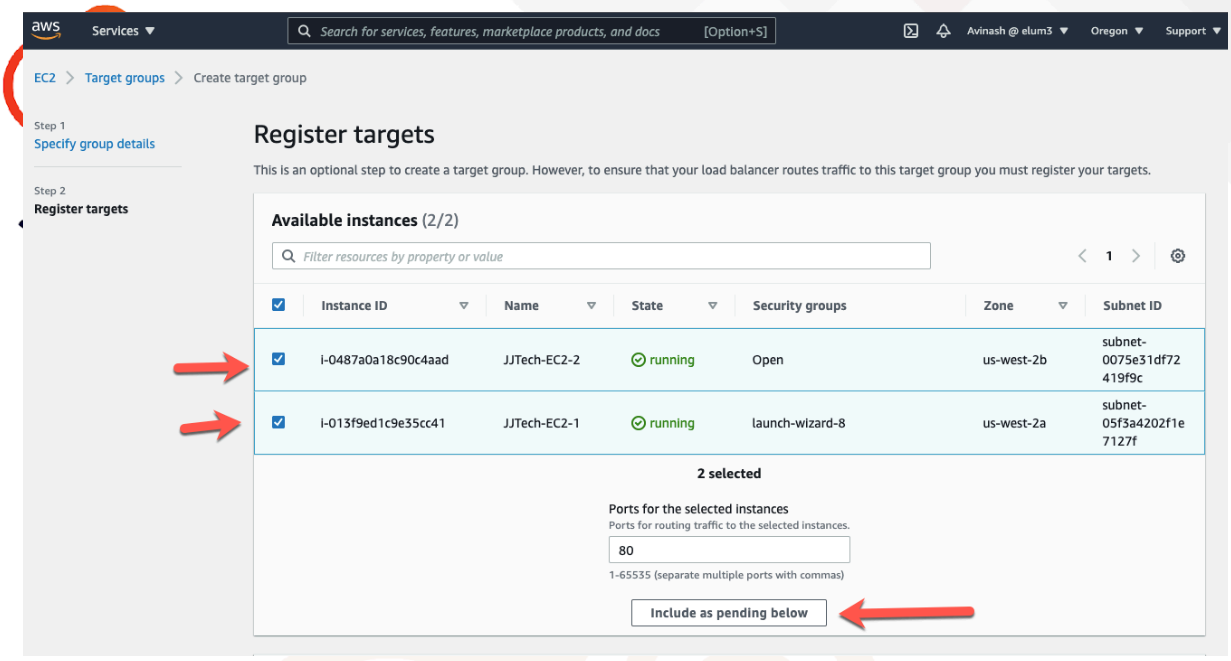
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol
HTTP ▼

Health check path
Use the default path of "/" to ping the root, or specify a custom path if preferred.
/index.html
Up to 1024 characters allowed.

▶ Advanced health check settings

▶ Tags - *optional*
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.


aws
Services ▼
Search for services, features, marketplace products, and docs [Option+S]
Avinash @ elum3 ▼
Oregon ▼
Support ▼

EC2 > Target groups > Create target group

Step 1
Specify group details

Step 2
Register targets

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/2)

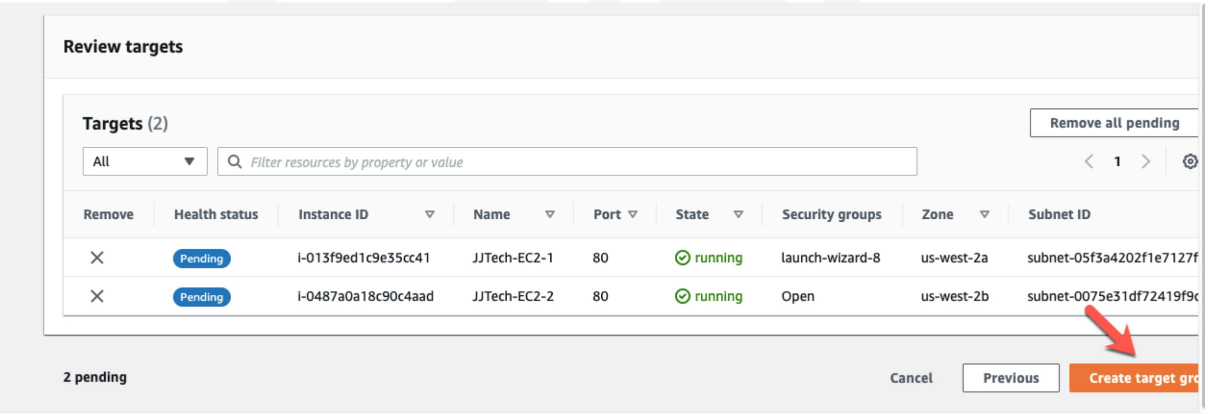
Filter resources by property or value

☑	Instance ID	Name	State	Security groups	Zone	Subnet ID
☑	i-0487a0a18c90c4aad	JJTech-EC2-2	✓ running	Open	us-west-2b	subnet-0075e31df72419f9c
☑	i-013f9ed1c9e35cc41	JJTech-EC2-1	✓ running	launch-wizard-8	us-west-2a	subnet-05f3a4202f1e7127f

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



Review targets

Targets (2)

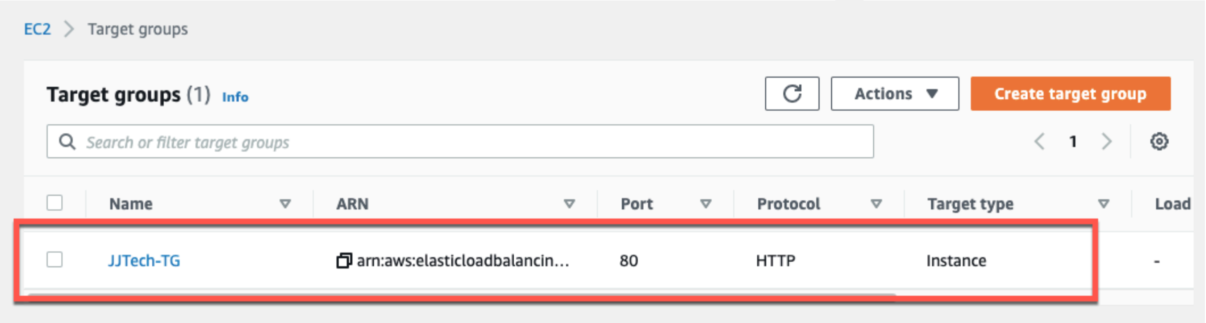
All ▼ Filter resources by property or value

Remove all pending

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone	Subnet ID
✕	Pending	i-013f9ed1c9e35cc41	JJTech-EC2-1	80	✓ running	launch-wizard-8	us-west-2a	subnet-05f3a4202f1e7127f
✕	Pending	i-0487a0a18c90c4aad	JJTech-EC2-2	80	✓ running	Open	us-west-2b	subnet-0075e31df72419f9c

2 pending

Cancel Previous Create target gro



EC2 > Target groups

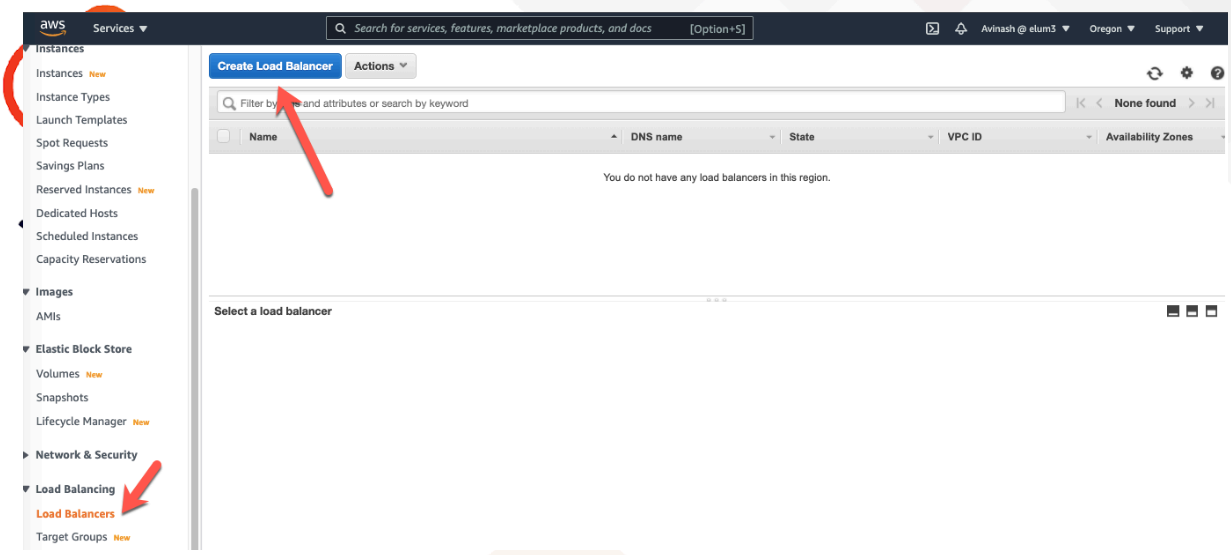
Target groups (1) Info

Actions ▼ Create target group

Search or filter target groups

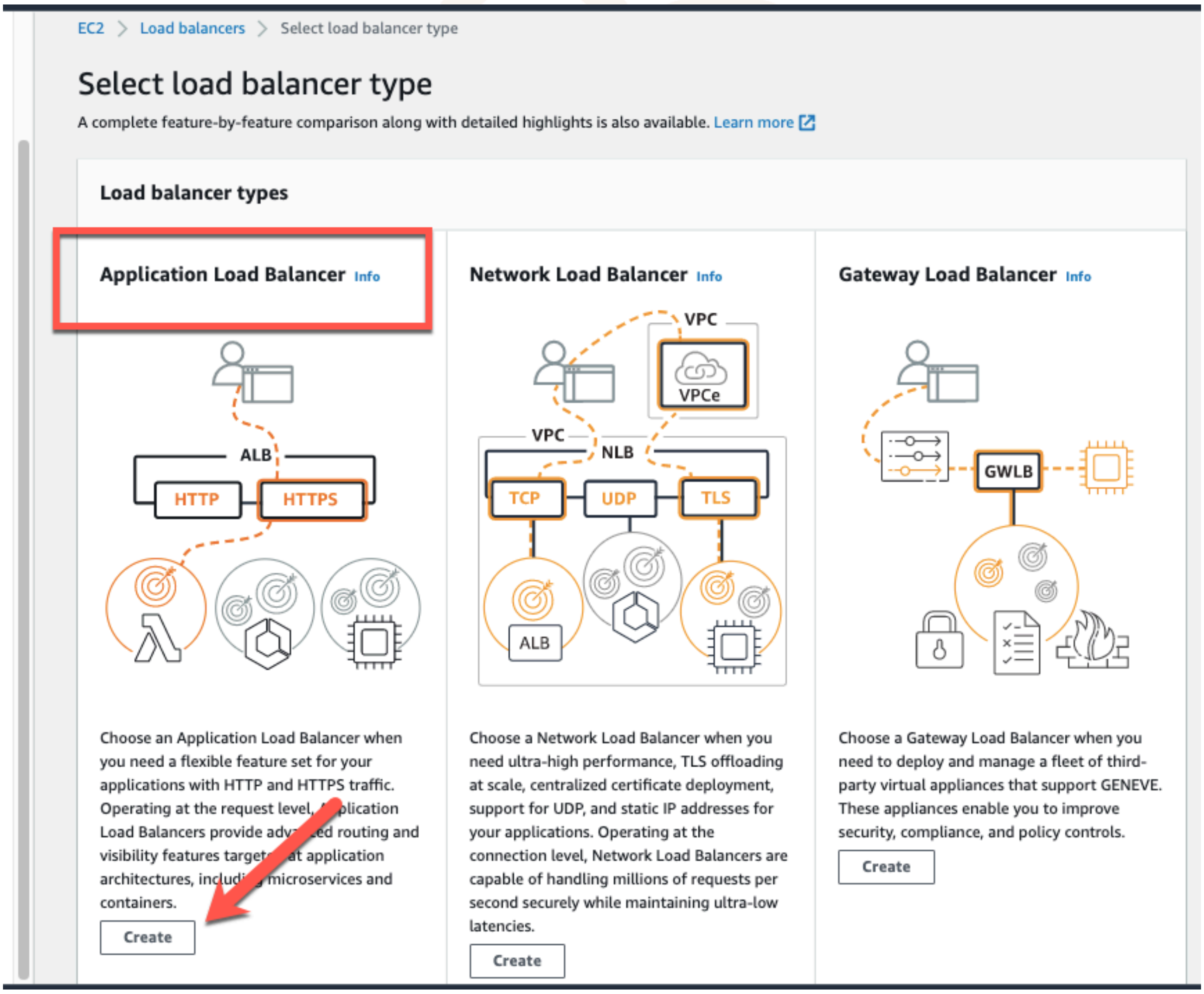
☐	Name	ARN	Port	Protocol	Target type	Load
☐	JJTech-TG	arn:aws:elasticloadbalancin…	80	HTTP	Instance	-

- Navigate to EC2 console □ Load balancer
- Create ALB, with above Target group



You do not have any load balancers in this region.

Select a load balancer



Select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. Learn more

Load balancer types

Application Load Balancer Info

Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Create

Network Load Balancer Info

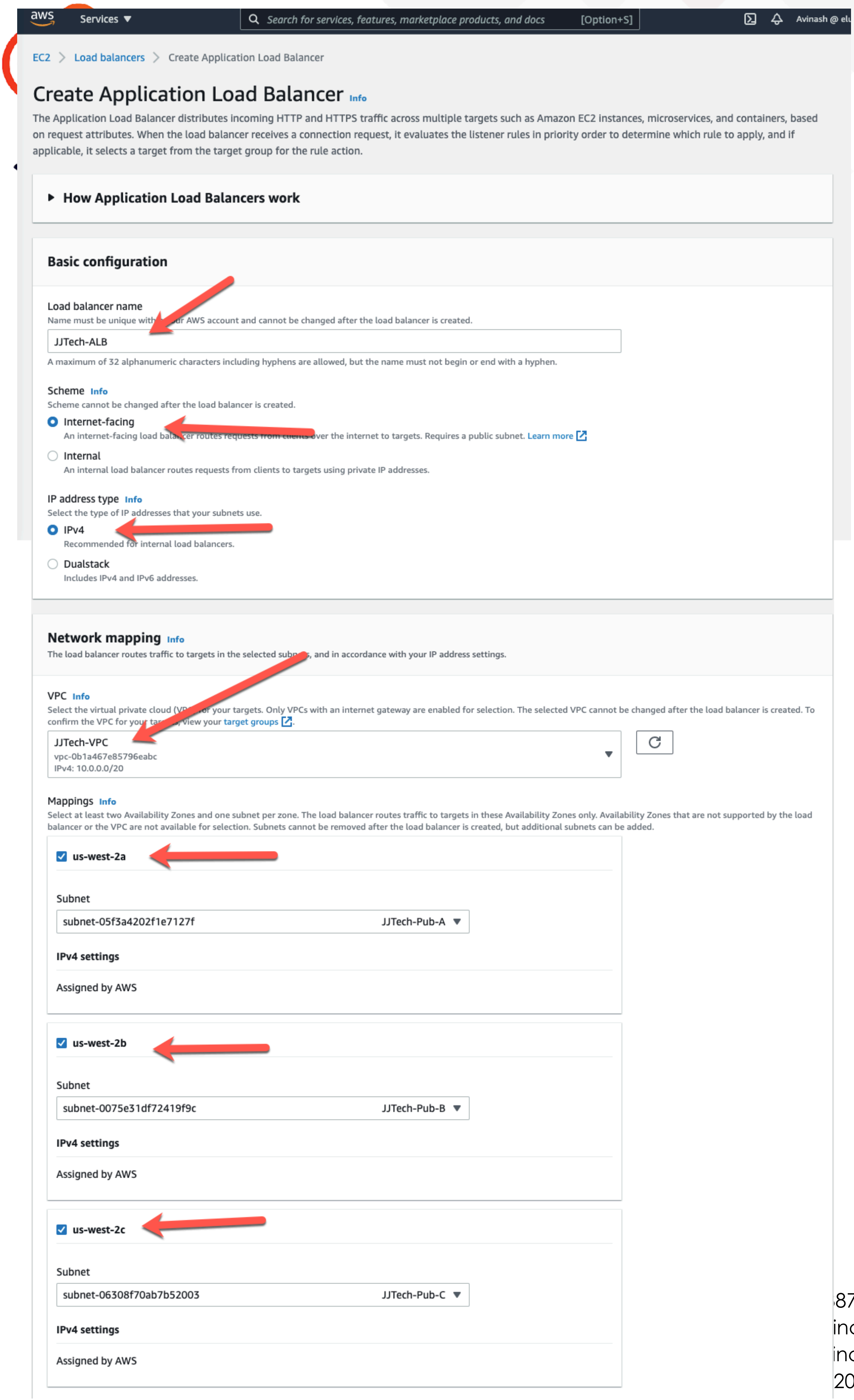
Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Create

Gateway Load Balancer Info

Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create


aws
Services ▼
Search for services, features, marketplace products, and docs
[Option+S]
Avinash @ elu

EC2 > Load balancers > Create Application Load Balancer

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 cannot be changed after the load balancer is created.
JJTech-ALB
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info
Scheme cannot 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.

Network mapping Info

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

VPC Info
Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your target groups.

JJTech-VPC
vpc-0b1a467e85796eabc
IPv4: 10.0.0.0/20

Mappings Info
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection. Subnets cannot be removed after the load balancer is created, but additional subnets can be added.

☑ **us-west-2a**

Subnet
subnet-05f3a4202f1e7127f JJTech-Pub-A ▼

IPv4 settings

Assigned by AWS

☑ **us-west-2b**

Subnet
subnet-0075e31df72419f9c JJTech-Pub-B ▼

IPv4 settings

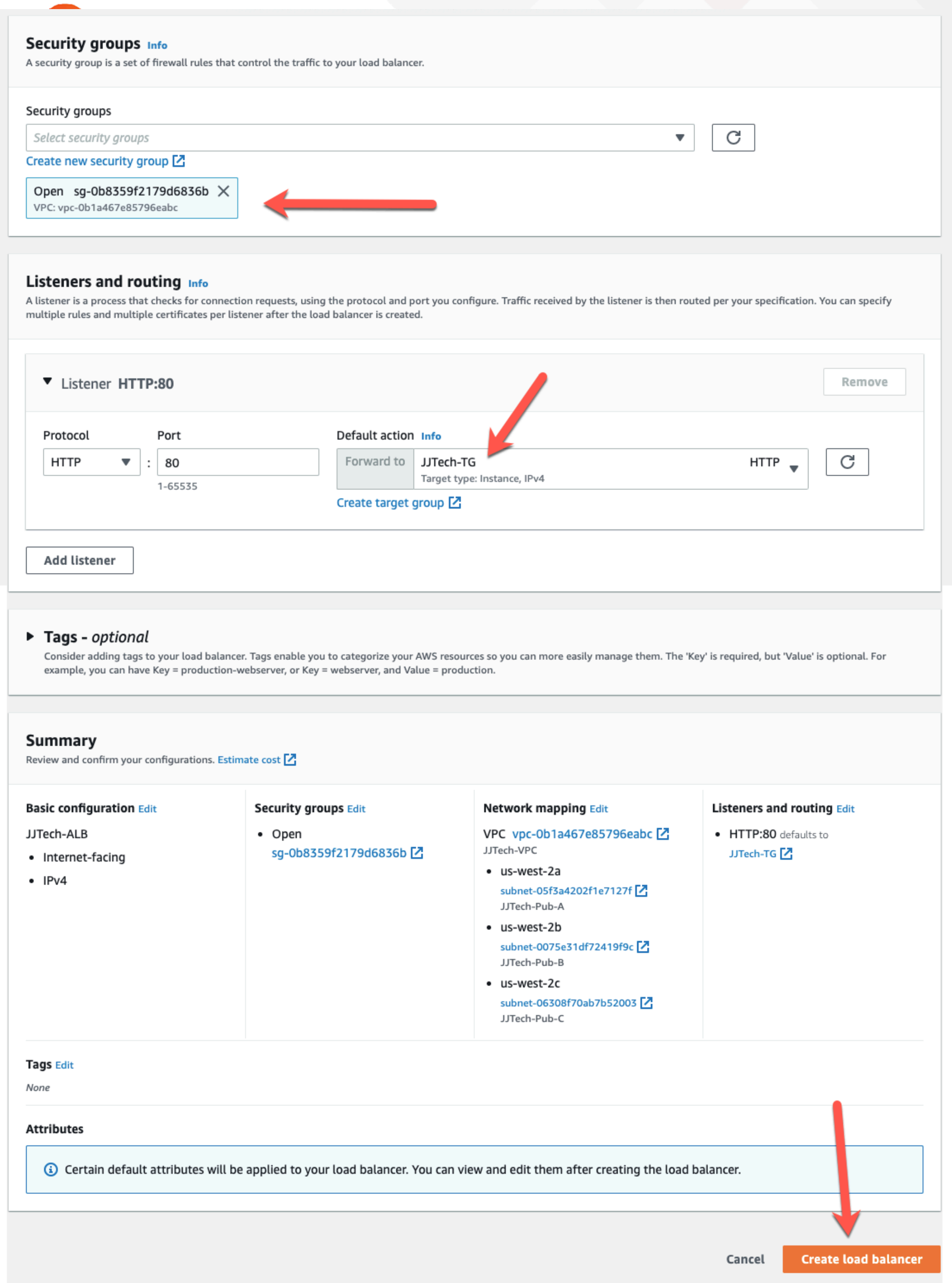
Assigned by AWS

☑ **us-west-2c**

Subnet
subnet-06308f70ab7b52003 JJTech-Pub-C ▼

IPv4 settings

Assigned by AWS



Security groups Info

A security group is a set of firewall rules that control the traffic to your load balancer.

Security groups

Select security groups ▼

Create new security group

Open sg-0b8359f2179d6836b ✕
VPC: vpc-0b1a467e85796eabc

Listeners and routing Info

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

▼ Listener **HTTP:80** Remove

Protocol: HTTP ▼

Port: 80 (1-65535)

Default action Info: Forward to JJTech-TG HTTP ▼
Target type: Instance, IPv4

Create target group

Add listener

▶ **Tags** - *optional*

Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The 'Key' is required, but 'Value' is optional. For example, you can have Key = production-webserver, or Key = webserver, and Value = production.

Summary

Review and confirm your configurations. Estimate cost

Basic configuration Edit

JJTech-ALB

- Internet-facing
- IPv4

Security groups Edit

- Open
  sg-0b8359f2179d6836b

Network mapping Edit

VPC vpc-0b1a467e85796eabc
JJTech-VPC

- us-west-2a
  subnet-05f3a4202f1e7127f
  JJTech-Pub-A
- us-west-2b
  subnet-0075e31df72419f9c
  JJTech-Pub-B
- us-west-2c
  subnet-06308f70ab7b52003
  JJTech-Pub-C

Listeners and routing Edit

- HTTP:80 defaults to
  JJTech-TG

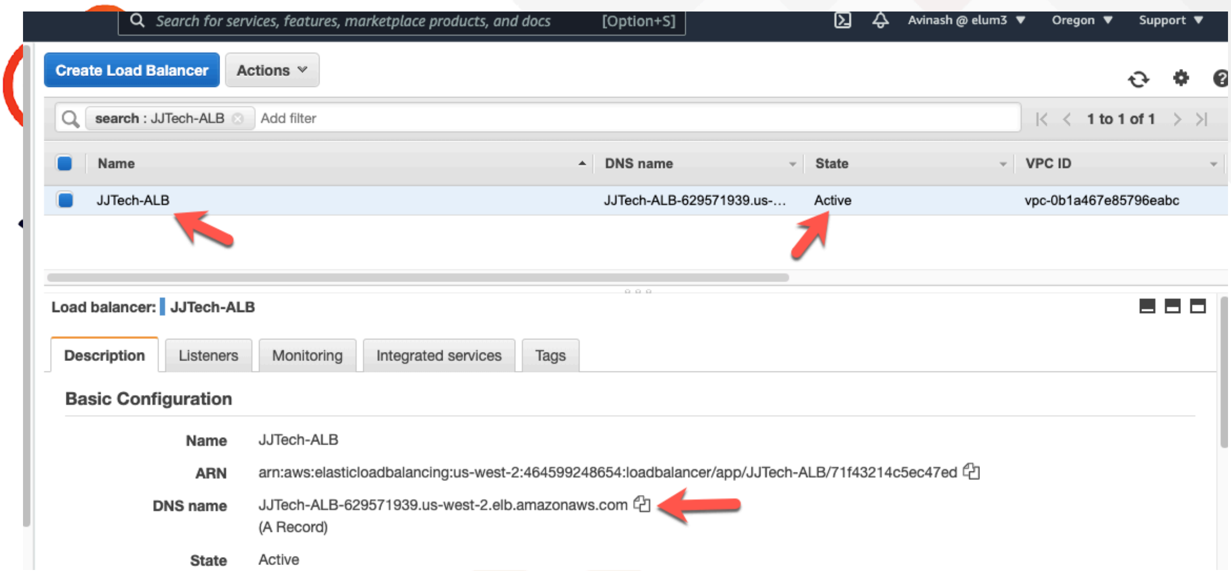
Tags Edit

None

Attributes

ⓘ Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

Cancel **Create load balancer**



Name	DNS name	State	VPC ID
JJTech-ALB	JJTech-ALB-629571939.us-...	Active	vpc-0b1a467e85796eabc

Load balancer: JJTech-ALB

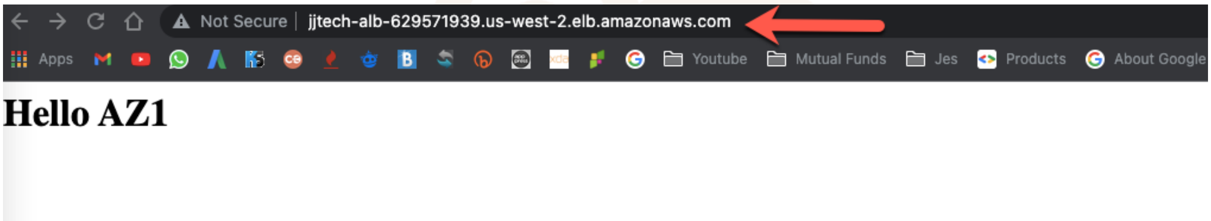
Basic Configuration

Name JJTech-ALB

ARN arn:aws:elasticloadbalancing:us-west-2:464599248654:loadbalancer/app/JJTech-ALB/71f43214c5ec47ed

DNS name JJTech-ALB-629571939.us-west-2.elb.amazonaws.com (A Record)

State Active

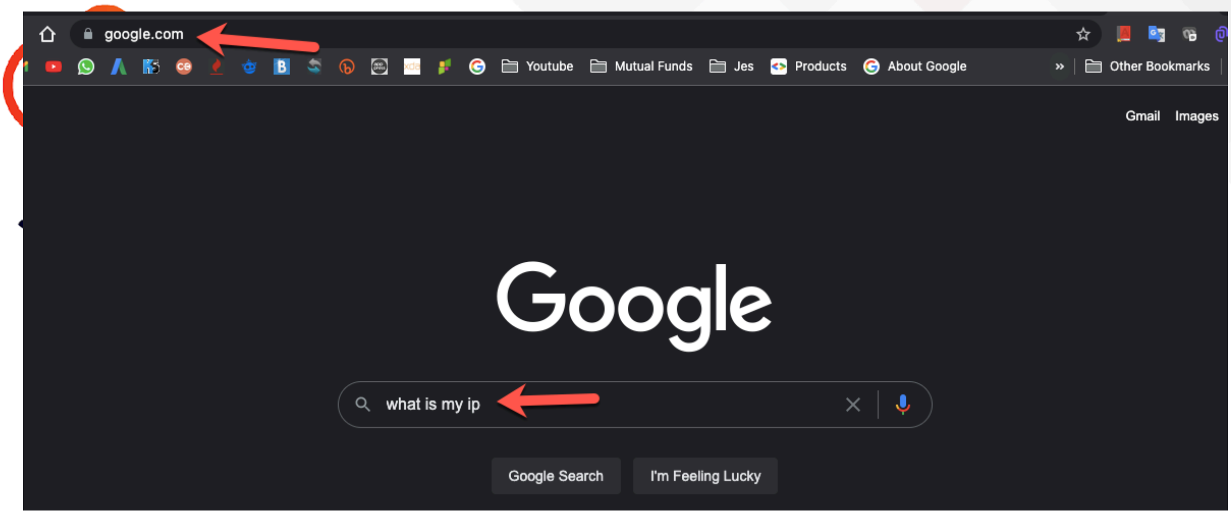


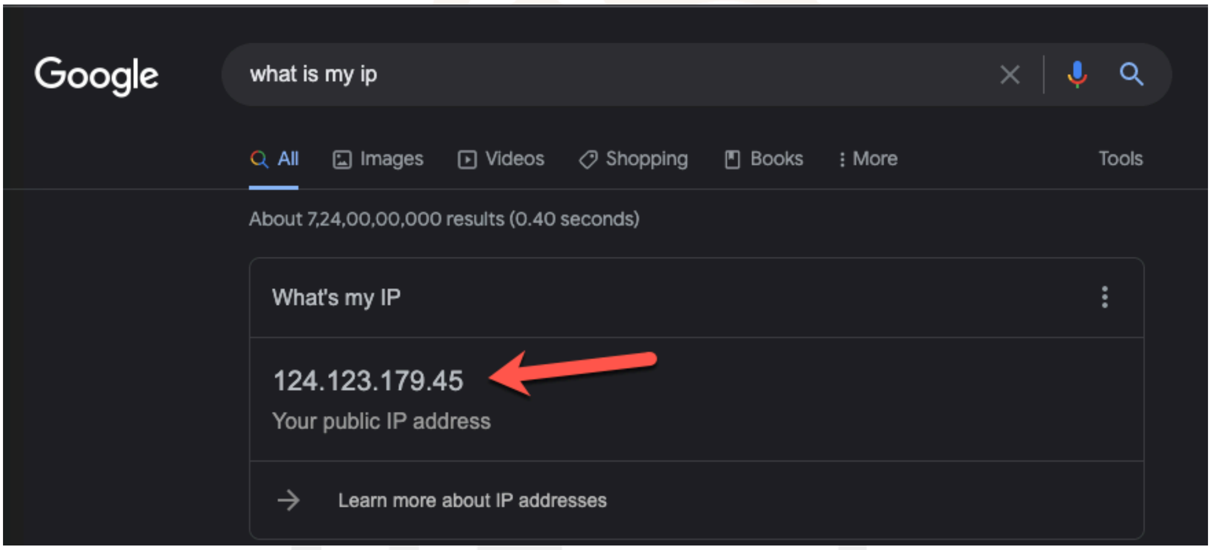
Hello AZ1

- Refresh the browser traffic will be loaded to another EC2 Instance
- Now, you are able to access the app from load balancer
- Let’s restrict the access through WAF

Step 3 : Create IP set in WAF

- Before we setup IP set, let’s get your ip by typing “what is my ip” in Google search





About 7,24,00,00,000 results (0.40 seconds)

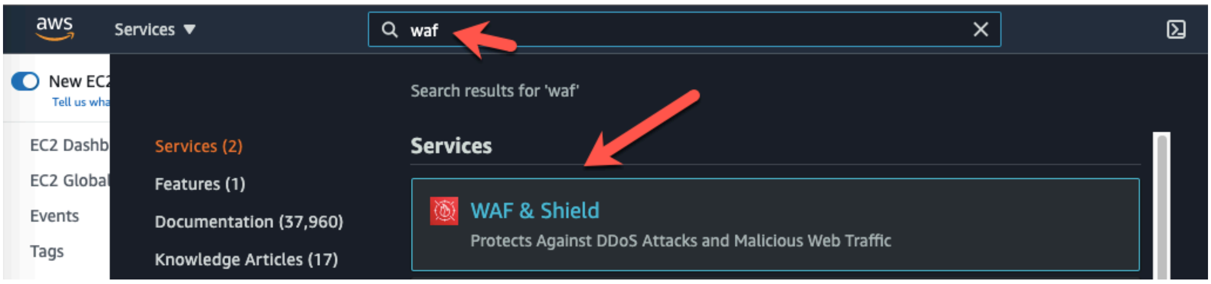
What's my IP

124.123.179.45

Your public IP address

Learn more about IP addresses

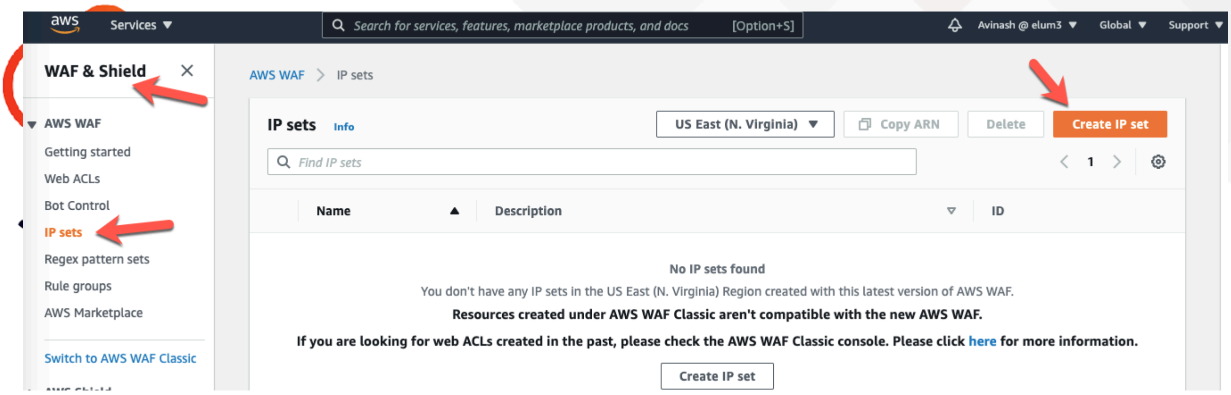
- Navigate to WAF AWS Console and create an IP set



Search results for 'waf'

WAF & Shield

Protects Against DDoS Attacks and Malicious Web Traffic



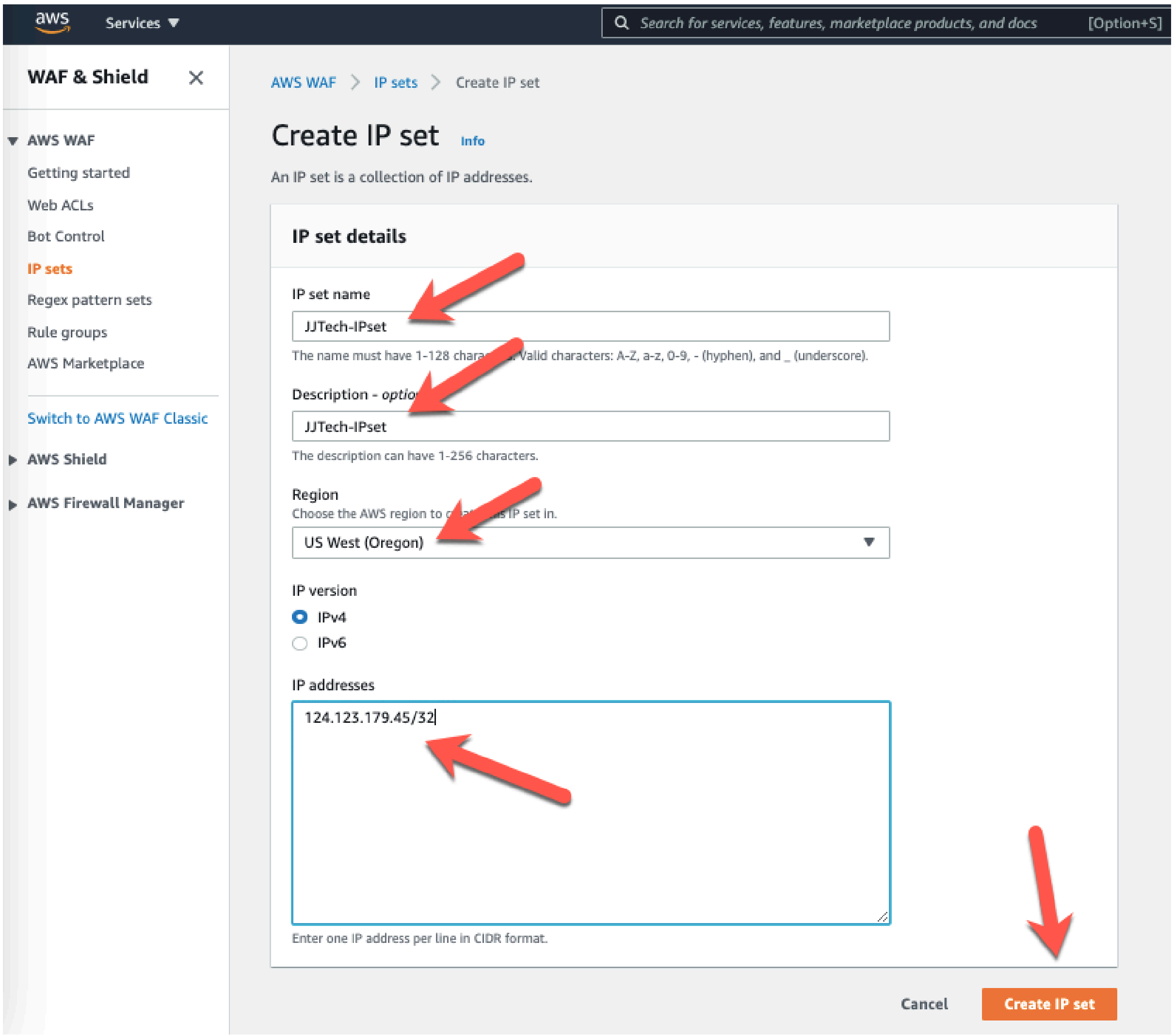
No IP sets found

You don't have any IP sets in the US East (N. Virginia) Region created with this latest version of AWS WAF.

Resources created under AWS WAF Classic aren't compatible with the new AWS WAF.

If you are looking for web ACLs created in the past, please check the AWS WAF Classic console. Please click here for more information.

Create IP set



Create IP set Info

An IP set is a collection of IP addresses.

IP set details

IP set name: JJTech-IPset

The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Description - optional: JJTech-IPset

The description can have 1-256 characters.

Region

Choose the AWS region to create this IP set in.

US West (Oregon) ▼

IP version

● IPv4

○ IPv6

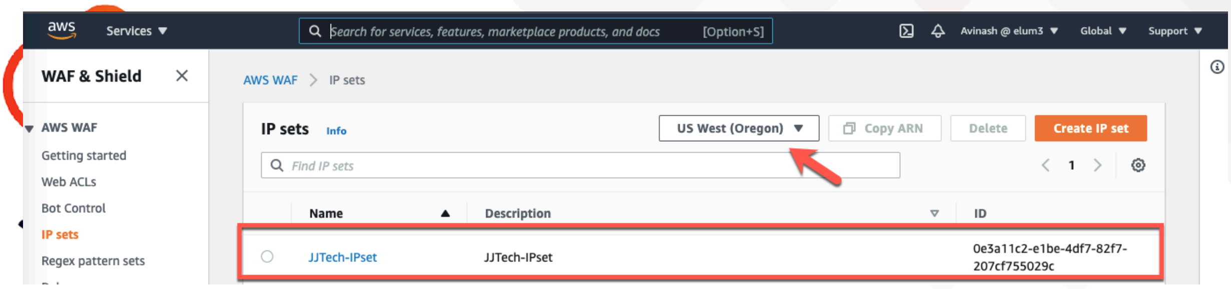
IP addresses

124.123.179.45/32

Enter one IP address per line in CIDR format.

Cancel   Create IP set

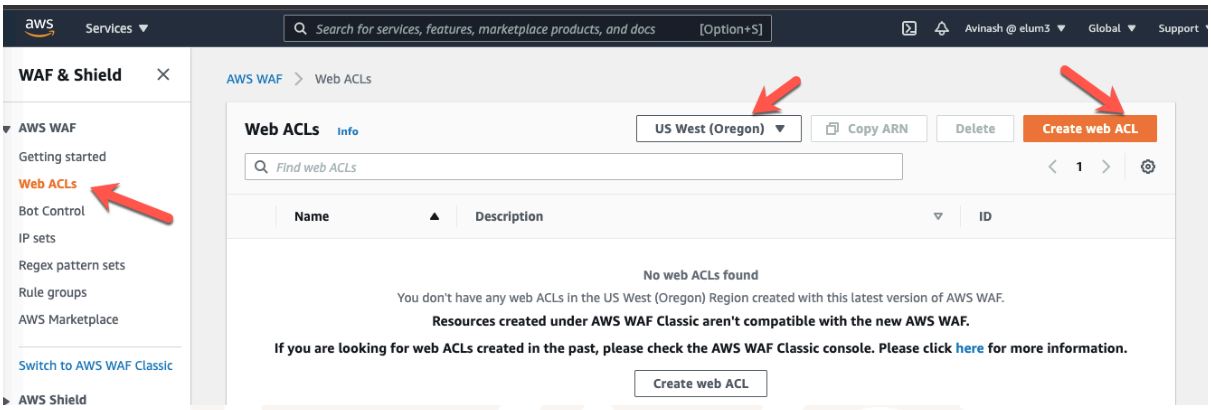
- Add the IP address you got from “what is my ip” with /32



IP sets — US West (Oregon) ▼

Name	Description	ID
○ JJTech-IPset	JJTech-IPset	0e3a11c2-e1be-4df7-82f7-207cf755029c

- Navigate to AWS WAF Web ACl’s console



Web ACLs — US West (Oregon) ▼

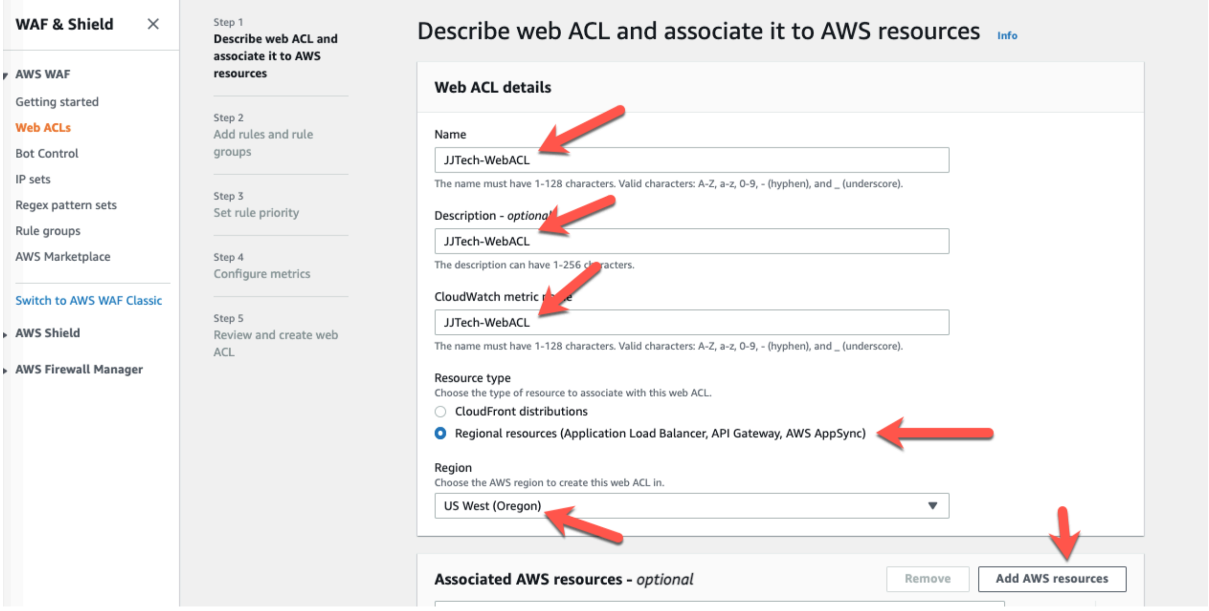
No web ACLs found

You don't have any web ACLs in the US West (Oregon) Region created with this latest version of AWS WAF.

Resources created under AWS WAF Classic aren't compatible with the new AWS WAF.

If you are looking for web ACLs created in the past, please check the AWS WAF Classic console. Please click here for more information.

Create web ACL



Describe web ACL and associate it to AWS resources

Web ACL details

Name: JJTech-WebACL
The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

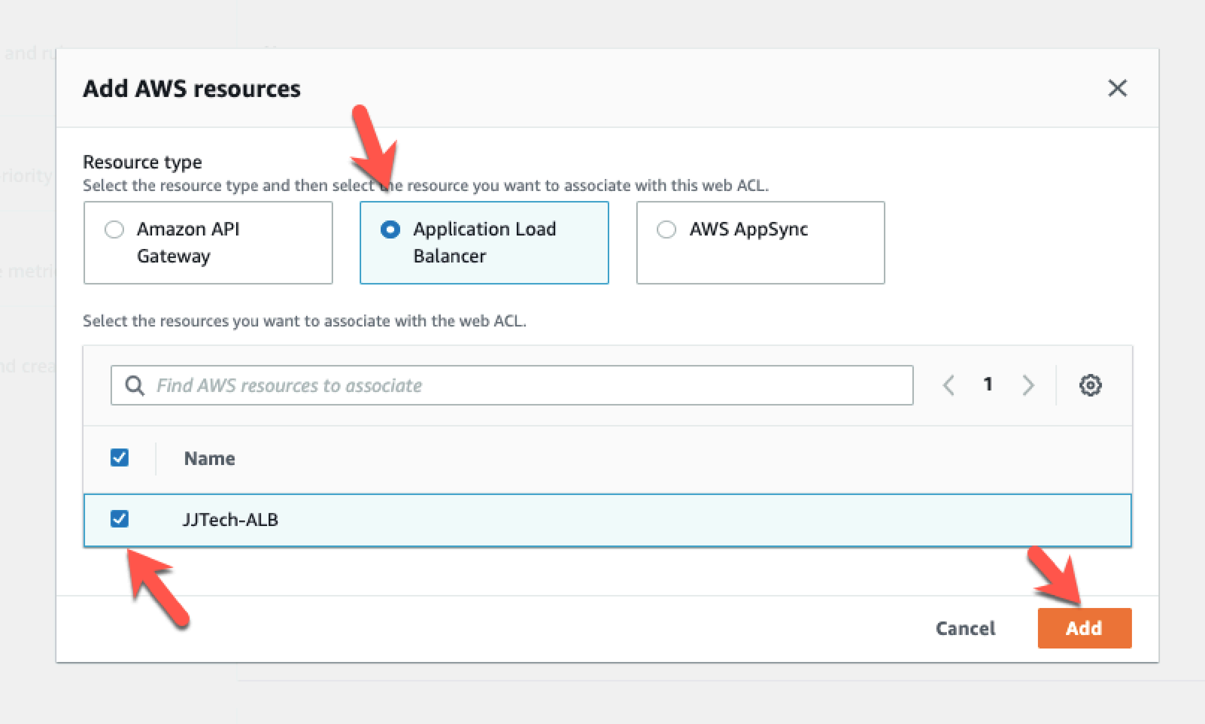
Description - *optional*: JJTech-WebACL
The description can have 1-256 characters.

CloudWatch metric name: JJTech-WebACL
The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Resource type
Choose the type of resource to associate with this web ACL.
○ CloudFront distributions
● Regional resources (Application Load Balancer, API Gateway, AWS AppSync)

Region
Choose the AWS region to create this web ACL in.
US West (Oregon) ▼

Associated AWS resources - *optional* — Remove — Add AWS resources



Add AWS resources

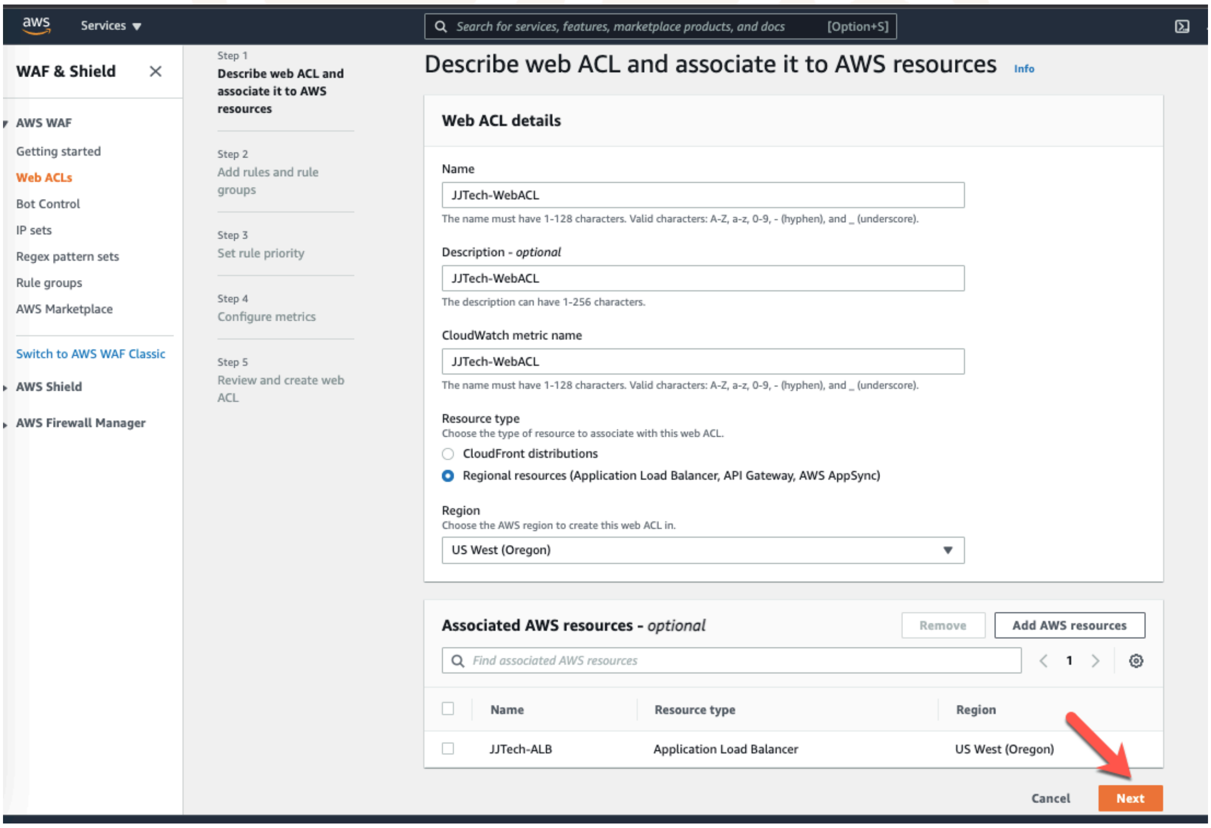
Resource type

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

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

☑	Name
☑	JJTech-ALB



Step 1
Describe web ACL and associate it to AWS resources

Step 2
Add rules and rule groups

Step 3
Set rule priority

Step 4
Configure metrics

Step 5
Review and create web ACL

Describe web ACL and associate it to AWS resources Info

Web ACL details

Name: JJTech-WebACL

The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Description - *optional*: JJTech-WebACL

The description can have 1-256 characters.

CloudWatch metric name: JJTech-WebACL

The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Resource type

Choose the type of resource to associate with this web ACL.

○ CloudFront distributions

● Regional resources (Application Load Balancer, API Gateway, AWS AppSync)

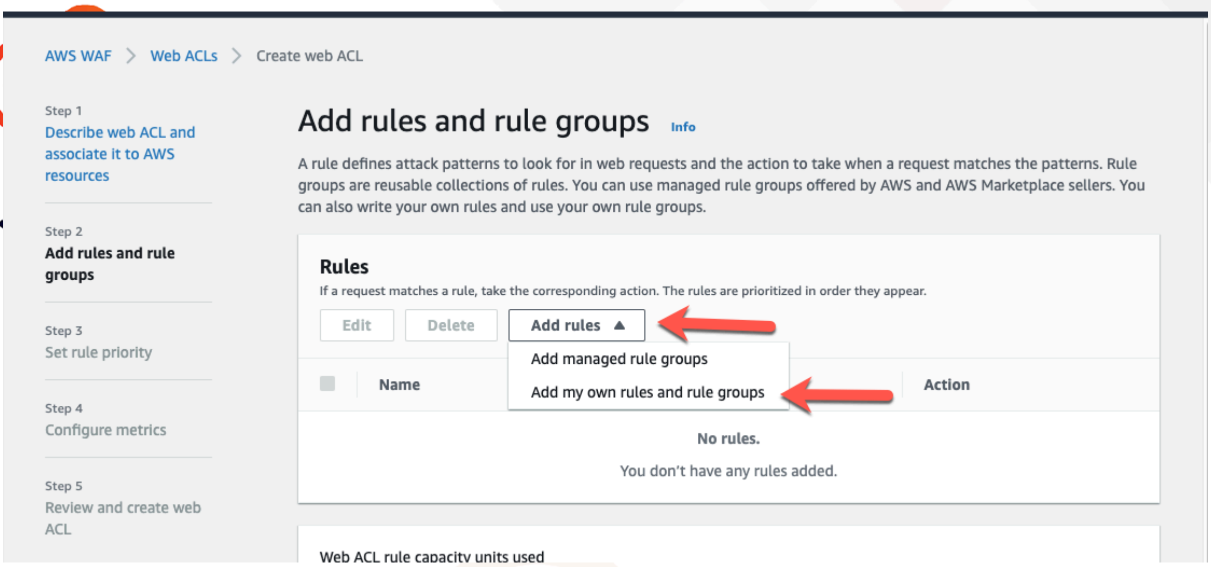
Region

Choose the AWS region to create this web ACL in.

US West (Oregon)

Associated AWS resources - *optional*

□	Name	Resource type	Region
□	JJTech-ALB	Application Load Balancer	US West (Oregon)



AWS WAF > Web ACLs > Create web ACL

Step 1
Describe web ACL and associate it to AWS resources

Step 2
Add rules and rule groups

Step 3
Set rule priority

Step 4
Configure metrics

Step 5
Review and create web ACL

Add rules and rule groups Info

A rule defines attack patterns to look for in web requests and the action to take when a request matches the patterns. Rule groups are reusable collections of rules. You can use managed rule groups offered by AWS and AWS Marketplace sellers. You can also write your own rules and use your own rule groups.

Rules

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

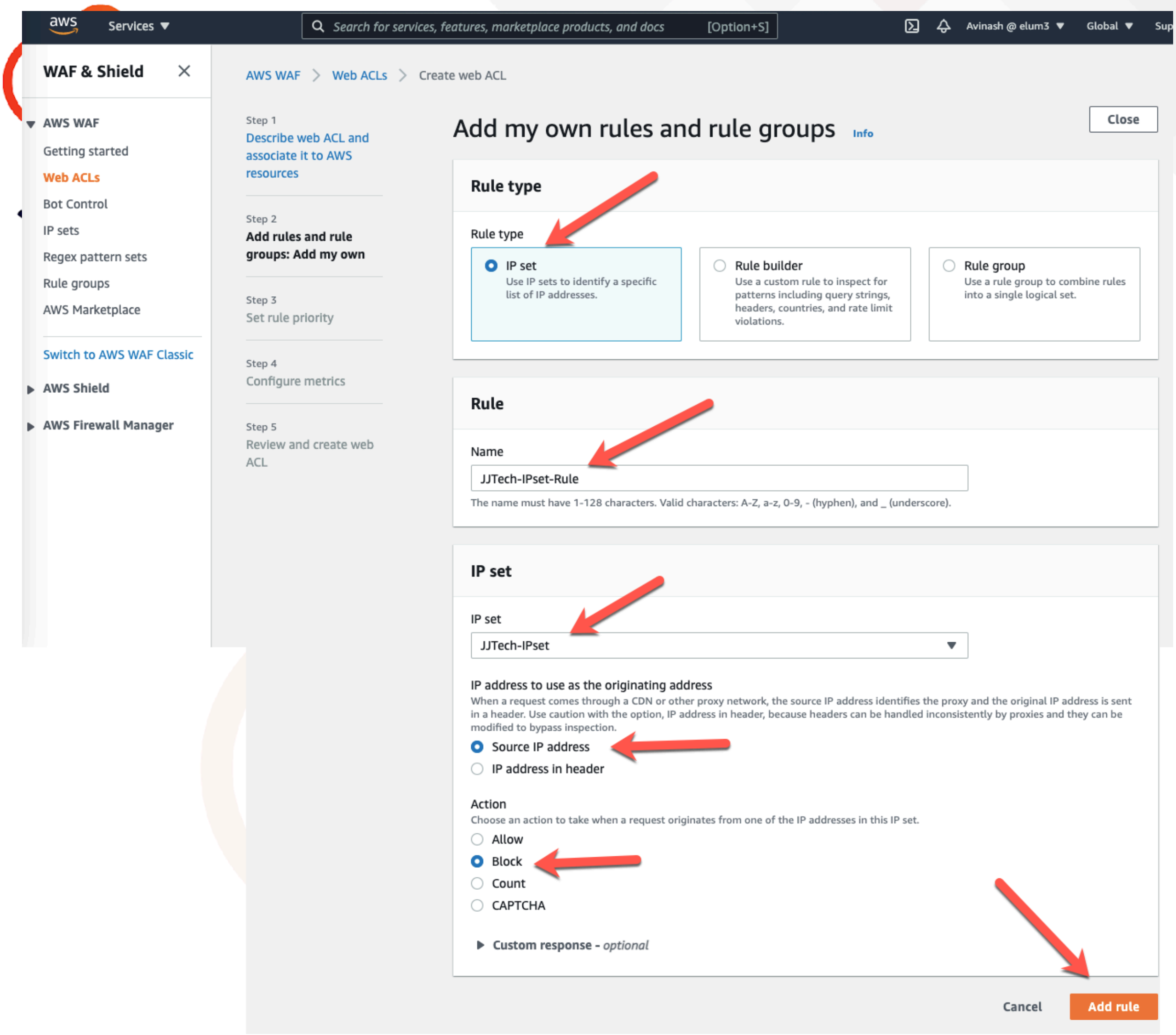
□	Name	Action

No rules.

You don't have any rules added.

Web ACL rule capacity units used

- In this test we are going to block our own IP and allowing remaining all Ips



aws
Services ▼
Search for services, features, marketplace products, and docs [Option+S]
Avinash @ elum3 ▼
Global ▼
Sup

WAF & Shield

- AWS WAF
  - Getting started
  - Web ACLs
  - Bot Control
  - IP sets
  - Regex pattern sets
  - Rule groups
  - AWS Marketplace
- Switch to AWS WAF Classic
- AWS Shield
- AWS Firewall Manager

AWS WAF > Web ACLs > Create web ACL

Step 1
Describe web ACL and associate it to AWS resources

Step 2
Add rules and rule groups: Add my own

Step 3
Set rule priority

Step 4
Configure metrics

Step 5
Review and create web ACL

Add my own rules and rule groups

Info

Close

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

JJTech-IPset-Rule

The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

IP set

IP set

JJTech-IPset ▼

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

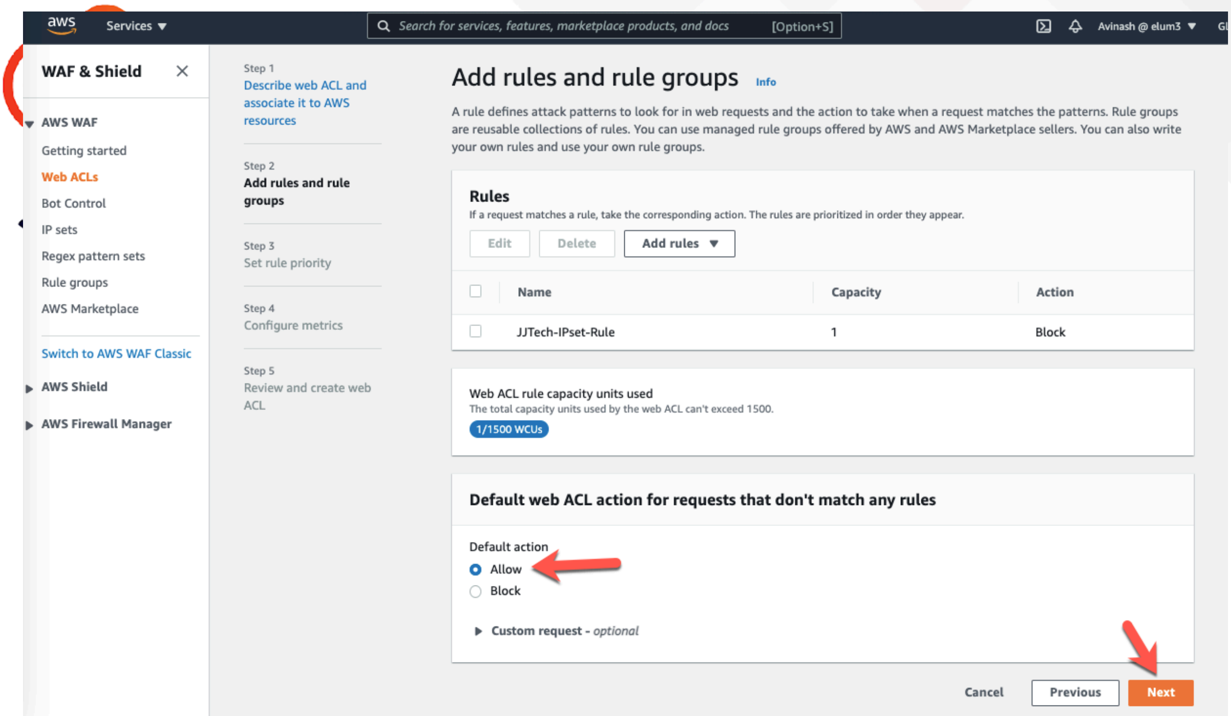
Choose an action to take when a request originates from one of the IP addresses in this IP set.

○ Allow
● Block
○ Count
○ CAPTCHA

▶ Custom response - *optional*

Cancel

Add rule



Add rules and rule groups

A rule defines attack patterns to look for in web requests and the action to take when a request matches the patterns. Rule groups are reusable collections of rules. You can use managed rule groups offered by AWS and AWS Marketplace sellers. You can also write your own rules and use your own rule groups.

Rules

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

□	Name	Capacity	Action
□	JJTech-IPset-Rule	1	Block

Web ACL rule capacity units used

The total capacity units used by the web ACL can't exceed 1500.

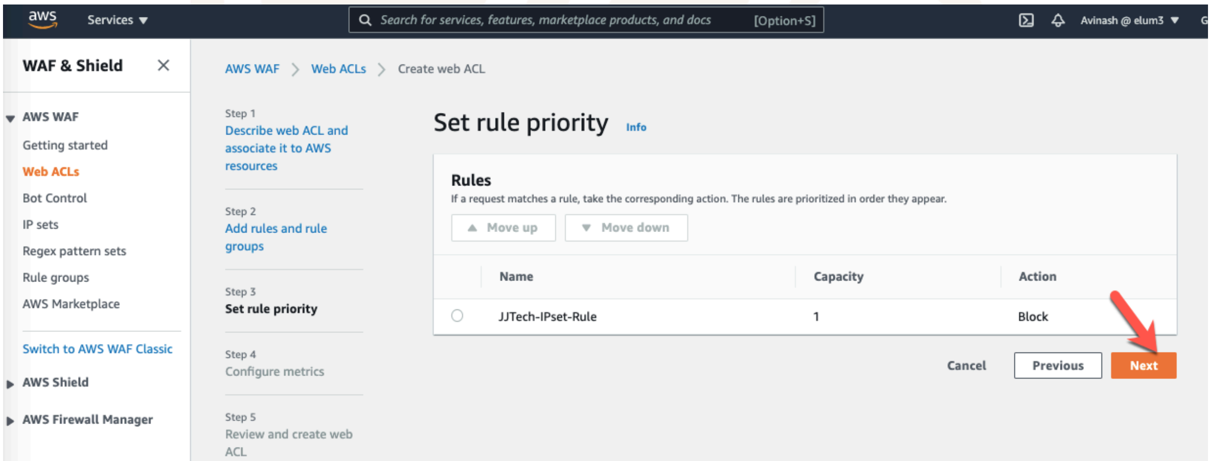
1/1500 WCUs

Default web ACL action for requests that don't match any rules

Default action

● Allow

○ Block

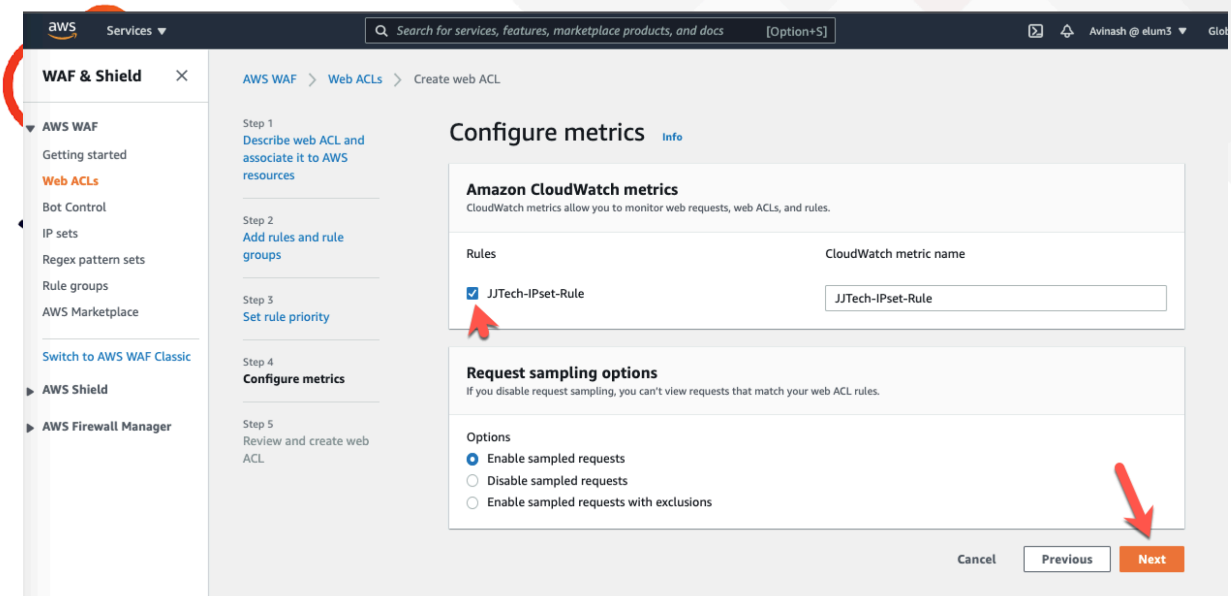


Set rule priority

Rules

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

	Name	Capacity	Action
○	JJTech-IPset-Rule	1	Block



Configure metrics

Amazon CloudWatch metrics

CloudWatch metrics allow you to monitor web requests, web ACLs, and rules.

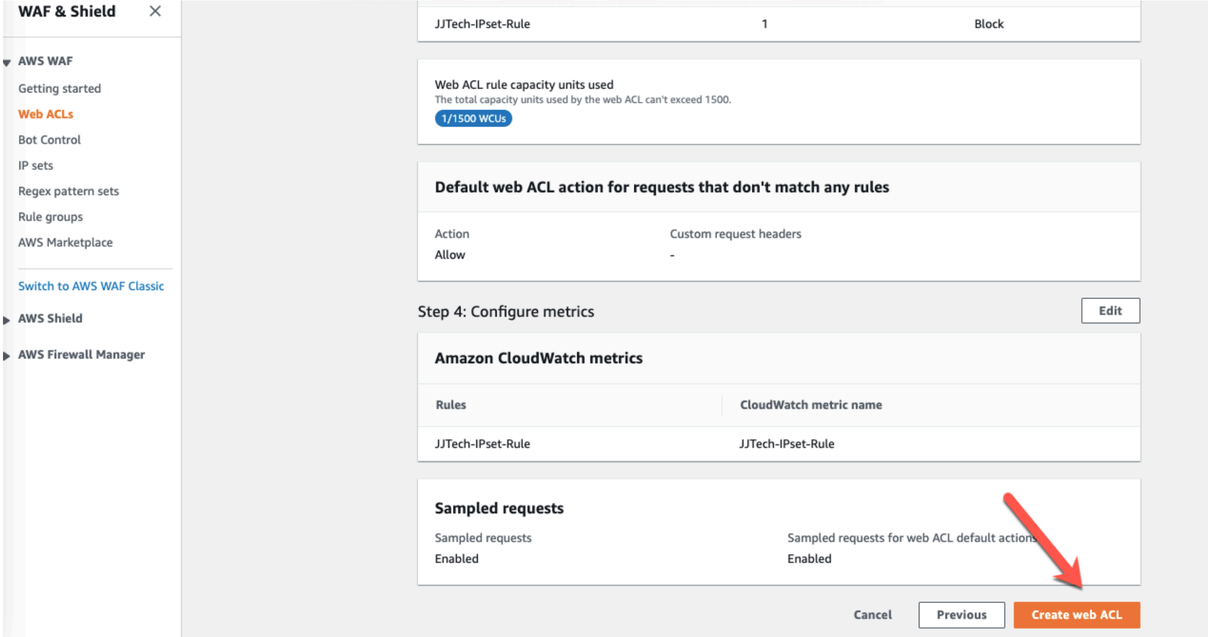
Rules	CloudWatch metric name
☑ JJTech-IPset-Rule	JJTech-IPset-Rule

Request sampling options

If you disable request sampling, you can't view requests that match your web ACL rules.

Options

- ● Enable sampled requests
- ○ Disable sampled requests
- ○ Enable sampled requests with exclusions



JJTech-IPset-Rule	1	Block

Web ACL rule capacity units used

The total capacity units used by the web ACL can't exceed 1500.

1/1500 WCUs

Default web ACL action for requests that don't match any rules

Action	Custom request headers
Allow	-

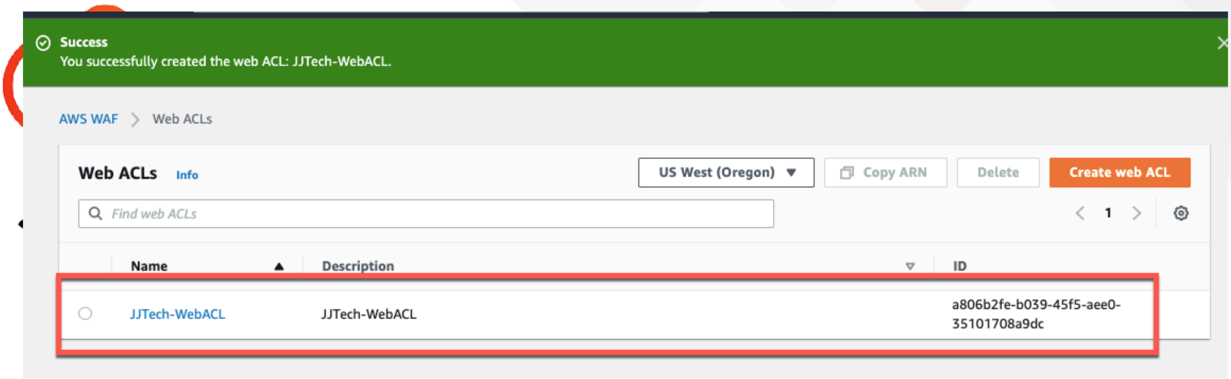
Step 4: Configure metrics

Amazon CloudWatch metrics

Rules	CloudWatch metric name
JJTech-IPset-Rule	JJTech-IPset-Rule

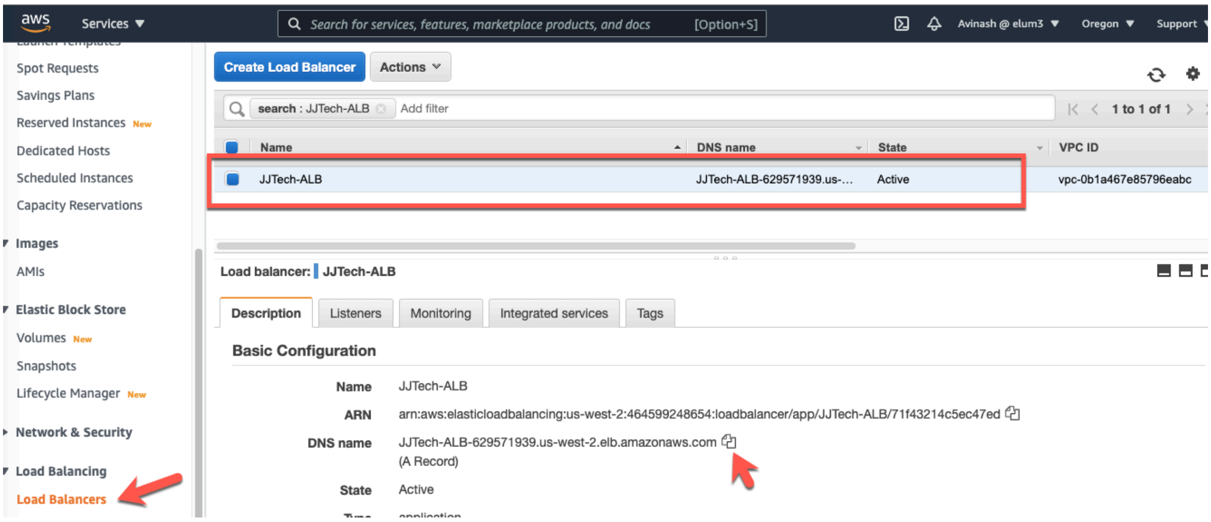
Sampled requests

Sampled requests	Sampled requests for web ACL default actions
Enabled	Enabled



Success
You successfully created the web ACL: JJTech-WebACL.

Name	Description	ID
○ JJTech-WebACL	JJTech-WebACL	a806b2fe-b039-45f5-aee0-35101708a9dc



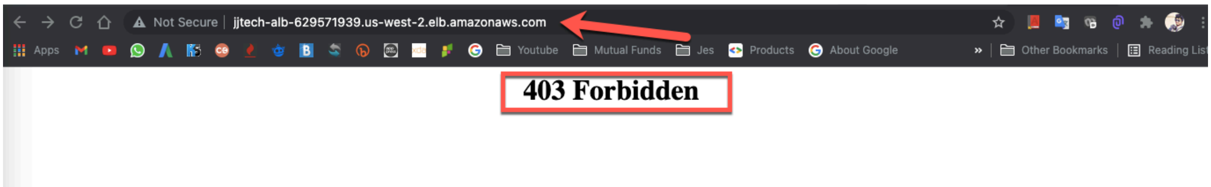
Name	DNS name	State	VPC ID
JJTech-ALB	JJTech-ALB-629571939.us-...	Active	vpc-0b1a467e85796eabc

Load balancer: JJTech-ALB

Basic Configuration

Name: JJTech-ALB
ARN: arn:aws:elasticloadbalancing:us-west-2:464599248654:loadbalancer/app/JJTech-ALB/71f43214c5ec47ed
DNS name: JJTech-ALB-629571939.us-west-2.elb.amazonaws.com (A Record)
State: Active

- Since we blocked our own IP , we cannot access the app from ALB DNS
- Copy the DNS and search in google



403 Forbidden

- It’s not working
- Our test is successful
- This is how we can block any Ips or/and allow any IPs

JJ Tech

JJ Tech

+1 (410) 8887049
contact@jjtechinc.co
www.jjtechinc.co
14103 Hammermil Field Dr Bowie MD 20720