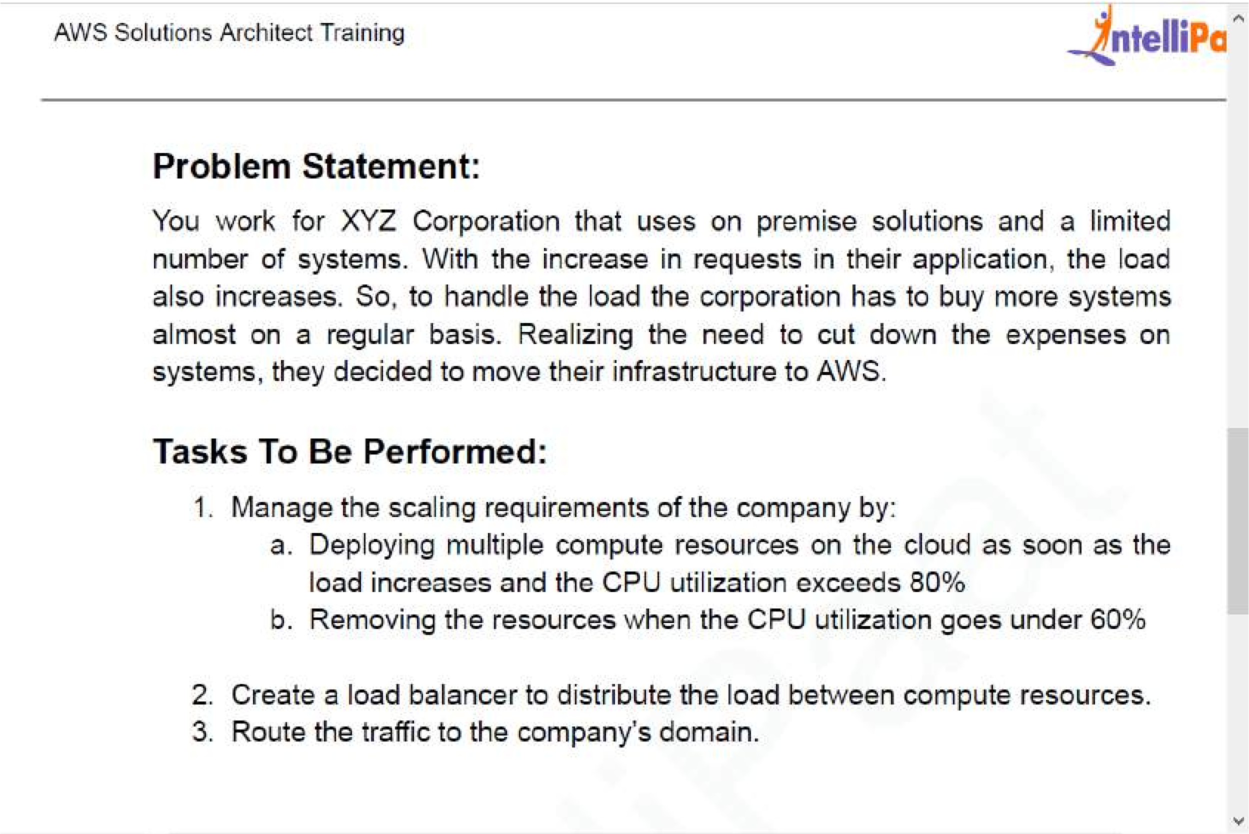
Dated 03.06.2024

Assignmnet for ELB and Route 53

Module 4: Case Study - 1



Problem Statement:

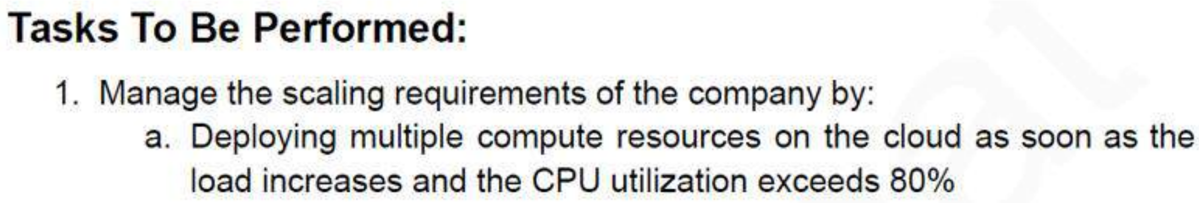
You work for XYZ Corporation that uses on premise solutions and a limited number of systems. With the increase in requests in their application, the load also increases. So, to handle the load the corporation has to buy more systems almost on a regular basis. Realizing the need to cut down the expenses on systems, they decided to move their infrastructure to AWS.

Tasks To Be Performed:

1. Manage the scaling requirements of the company by:
   a. Deploying multiple compute resources on the cloud as soon as the load increases and the CPU utilization exceeds 80%
   b. Removing the resources when the CPU utilization goes under 60%

2. Create a load balancer to distribute the load between compute resources.
3. Route the traffic to the company's domain.

First Deploy the 3 EC2 instances with web application in 3 linux versions

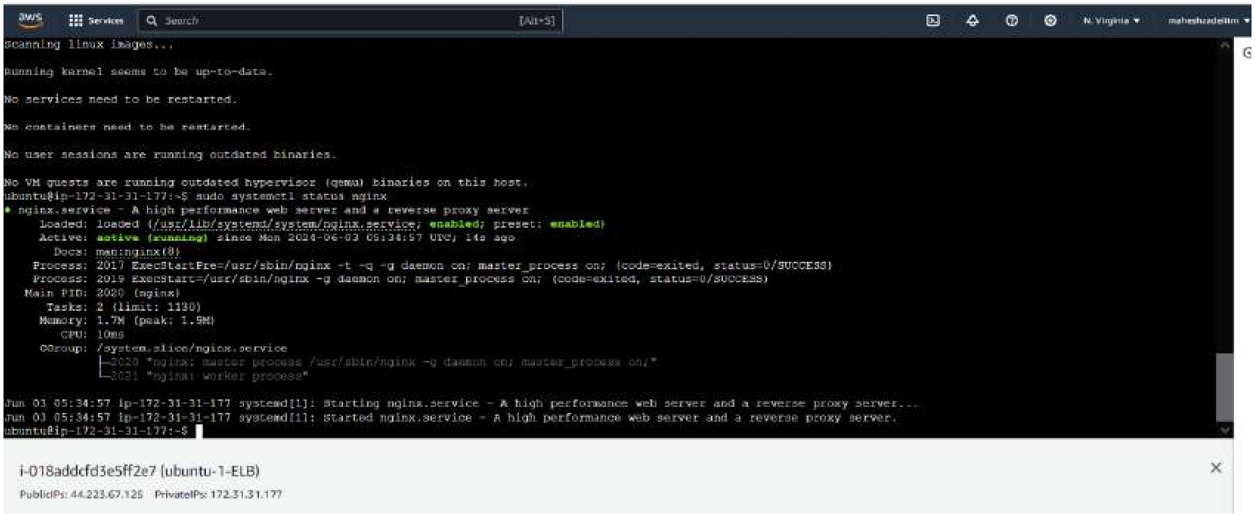


Tasks To Be Performed:

1. Manage the scaling requirements of the company by:
   a. Deploying multiple compute resources on the cloud as soon as the load increases and the CPU utilization exceeds 80%

Now go to AWS Console EC2 dashboard and create 3 instances in N verginia Region

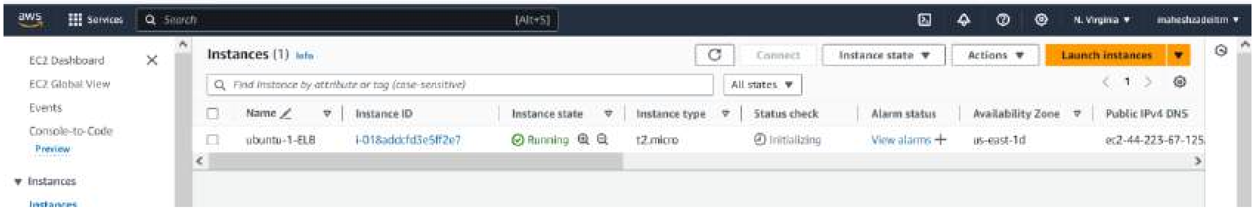
Ubuntu instance launched -1



```
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-31-177:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Mon 2024-06-03 05:34:57 UTC; 14s ago
       Docs: man:nginx(8)
    Process: 2017 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 2019 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 2020 (nginx)
      Tasks: 2 (limit: 1130)
     Memory: 1.7M (peak: 1.9M)
        CPU: 10ms
     CGroup: /system.slice/nginx.service
             ├─2020 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─2021 "nginx: worker process"

Jun 03 05:34:57 ip-172-31-31-177 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Jun 03 05:34:57 ip-172-31-31-177 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-31-177:~$
```

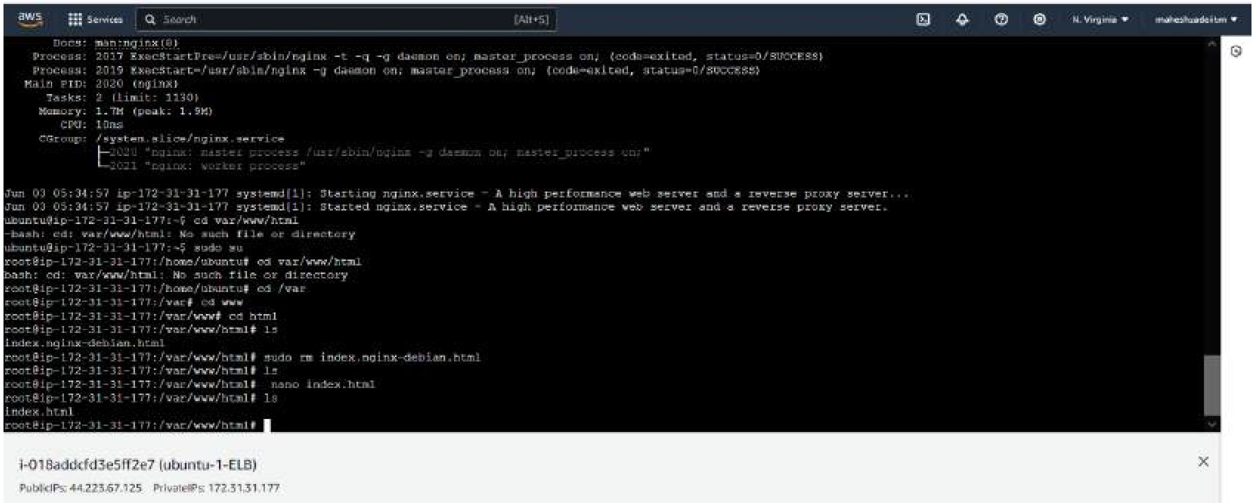
i-018addcfd3e5ff2e7 (ubuntu-1-ELB)

PublicIPs: 44.223.67.125 PrivateIPs: 172.31.31.177



Instances (1) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
□	ubuntu-1-ELB	i-018addcfd3e5ff2e7	Running	t2.micro	Initializing	View alarms +	us-east-1d	ec2-44-223-67-125.

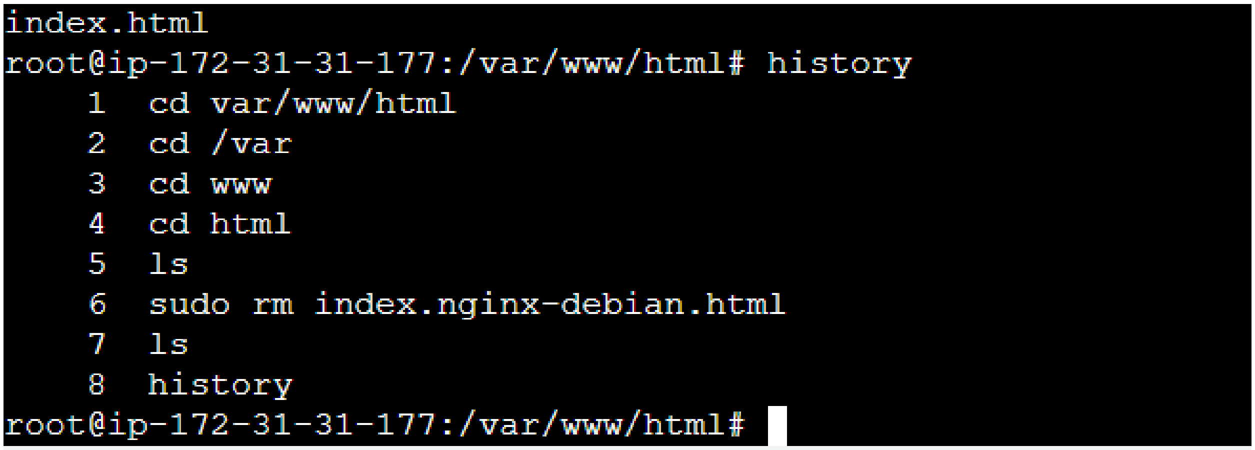


```
       Docs: man:nginx(8)
    Process: 2017 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 2019 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 2020 (nginx)
      Tasks: 2 (limit: 1130)
     Memory: 1.7M (peak: 1.9M)
        CPU: 10ms
     CGroup: /system.slice/nginx.service
             ├─2020 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─2021 "nginx: worker process"

Jun 03 05:34:57 ip-172-31-31-177 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Jun 03 05:34:57 ip-172-31-31-177 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-31-177:~$ cd var/www/html
-bash: cd: var/www/html: No such file or directory
ubuntu@ip-172-31-31-177:~$ sudo su
root@ip-172-31-31-177:/home/ubuntu# cd var/www/html
bash: cd: var/www/html: No such file or directory
root@ip-172-31-31-177:/home/ubuntu# cd /var
root@ip-172-31-31-177:/var# cd www
root@ip-172-31-31-177:/var/www# cd html
root@ip-172-31-31-177:/var/www/html# ls
index.nginx-debian.html
root@ip-172-31-31-177:/var/www/html# sudo rm index.nginx-debian.html
root@ip-172-31-31-177:/var/www/html# ls
root@ip-172-31-31-177:/var/www/html#  nano index.html
root@ip-172-31-31-177:/var/www/html# ls
index.html
root@ip-172-31-31-177:/var/www/html#
```

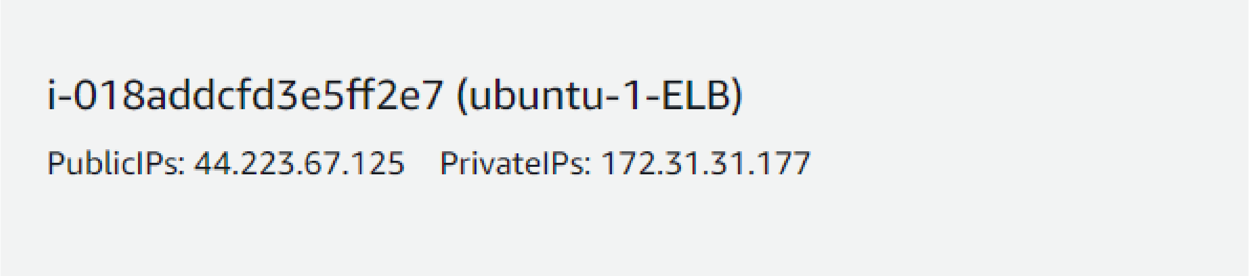
i-018addcfd3e5ff2e7 (ubuntu-1-ELB)

PublicIPs: 44.223.67.125 PrivateIPs: 172.31.31.177

Comands used below



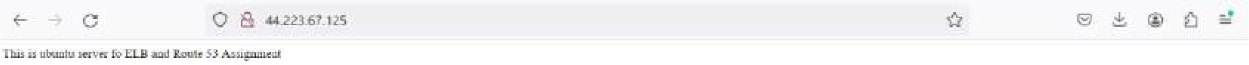
```
index.html
root@ip-172-31-31-177:/var/www/html# history
    1  cd var/www/html
    2  cd /var
    3  cd www
    4  cd html
    5  ls
    6  sudo rm index.nginx-debian.html
    7  ls
    8  history
root@ip-172-31-31-177:/var/www/html# 
```



i-018addcfd3e5ff2e7 (ubuntu-1-ELB)

PublicIPs: 44.223.67.125   PrivateIPs: 172.31.31.177

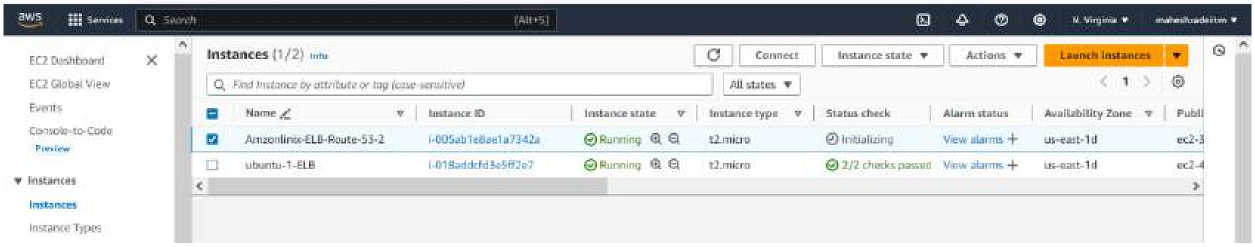
Copy the public IP and check and any browser



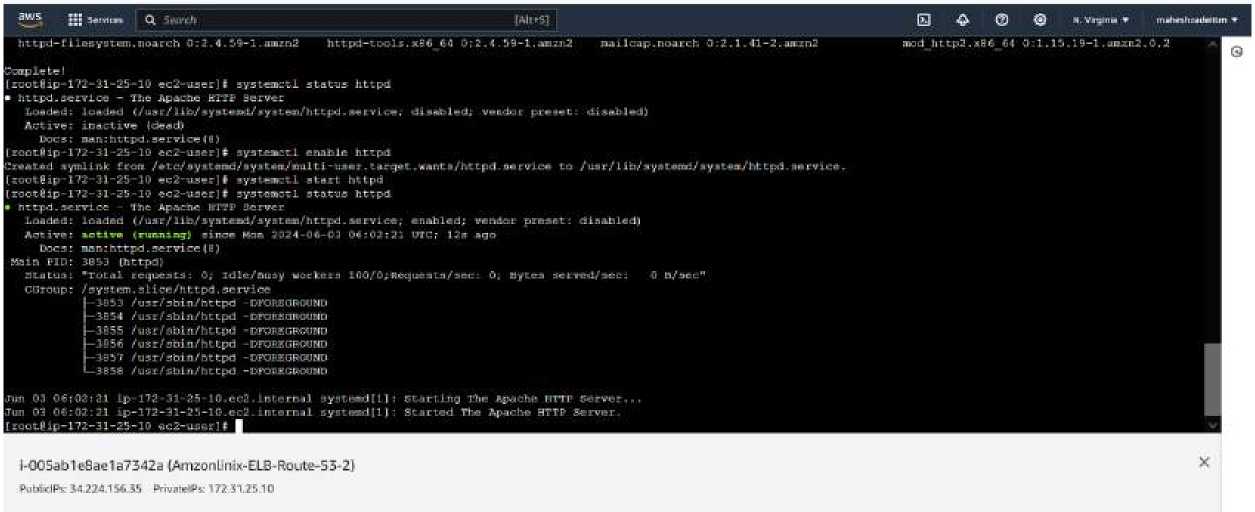
This is ubuntu server fo ELB and Route 53 Assignment

Its working now

Now Install the second linux Machine



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publi
☑	Amzonlinix-ELB-Route-53-2	i-005ab1e8ae1a7342a	Running	t2.micro	Initializing	View alarms +	us-east-1d	ec2-3
□	ubuntu-1-ELB	i-018addcfd3e5ff2e7	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-4

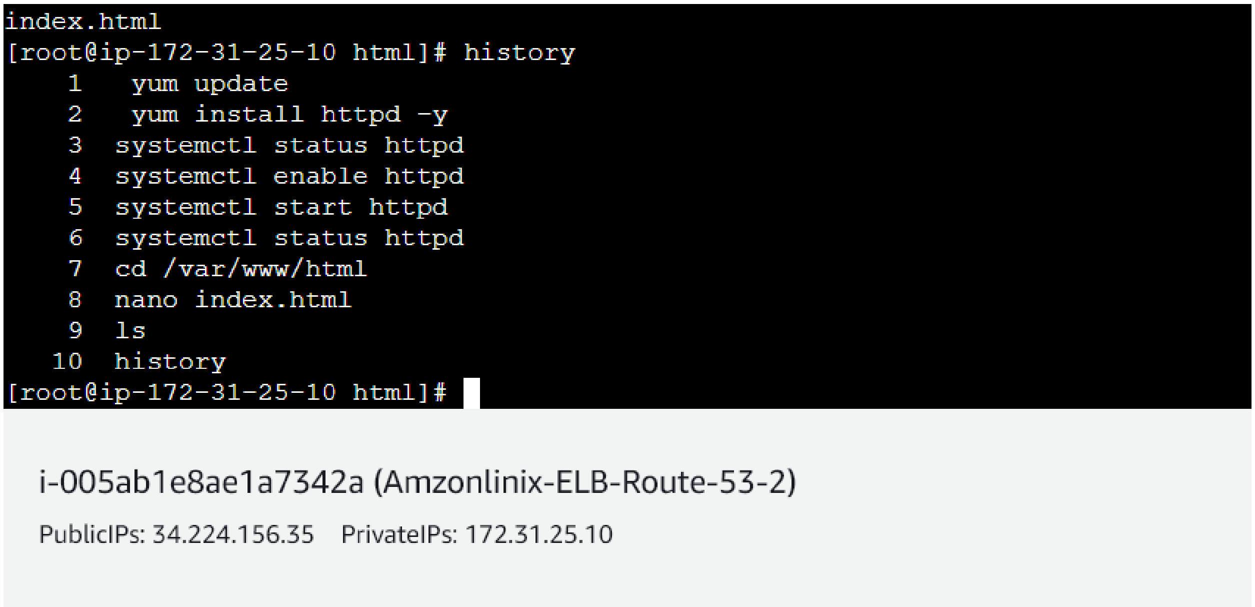


```
  httpd-filesystem.noarch 0:2.4.59-1.amzn2    httpd-tools.x86_64 0:2.4.59-1.amzn2    mailcap.noarch 0:2.1.41-2.amzn2           mod_http2.x86_64 0:1.15.19-1.amzn2.0.2

Complete!
[root@ip-172-31-25-10 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
[root@ip-172-31-25-10 ec2-user]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-25-10 ec2-user]# systemctl start httpd
[root@ip-172-31-25-10 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2024-06-03 06:02:21 UTC; 12s ago
     Docs: man:httpd.service(8)
 Main PID: 3853 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
   CGroup: /system.slice/httpd.service
           ├─3853 /usr/sbin/httpd -DFOREGROUND
           ├─3854 /usr/sbin/httpd -DFOREGROUND
           ├─3855 /usr/sbin/httpd -DFOREGROUND
           ├─3856 /usr/sbin/httpd -DFOREGROUND
           ├─3857 /usr/sbin/httpd -DFOREGROUND
           └─3858 /usr/sbin/httpd -DFOREGROUND

Jun 03 06:02:21 ip-172-31-25-10.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Jun 03 06:02:21 ip-172-31-25-10.ec2.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-172-31-25-10 ec2-user]#
```

i-005ab1e8ae1a7342a (Amzonlinix-ELB-Route-53-2)

PublicIPs: 34.224.156.35   PrivateIPs: 172.31.25.10

Commands run

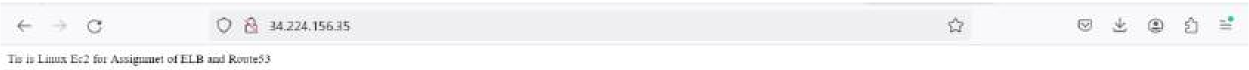


```
index.html
[root@ip-172-31-25-10 html]# history
    1   yum update
    2   yum install httpd -y
    3  systemctl status httpd
    4  systemctl enable httpd
    5  systemctl start httpd
    6  systemctl status httpd
    7  cd /var/www/html
    8  nano index.html
    9  ls
   10  history
[root@ip-172-31-25-10 html]#
```

i-005ab1e8ae1a7342a (Amzonlinix-ELB-Route-53-2)

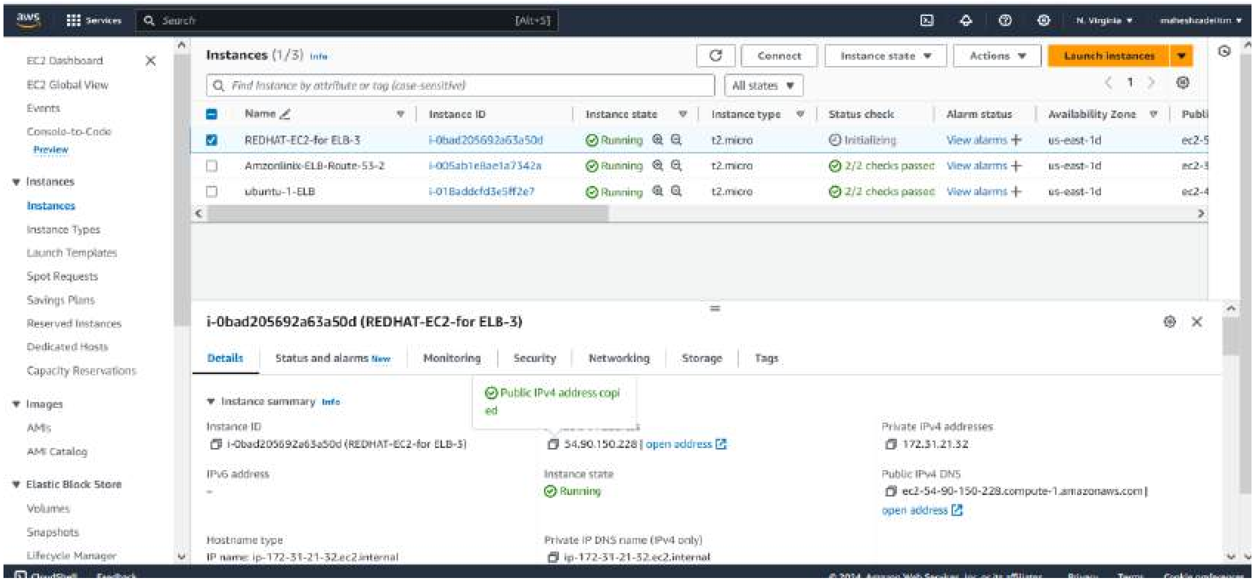
PublicIPs: 34.224.156.35   PrivateIPs: 172.31.25.10

Now copy the public IP and check the website is running or not



Tis is Linux Ec2 for Assigumet of ELB and Route53

Now create the third EC2 Redhat instance



Instances (1/3) Info

Find Instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publi
☑	REDHAT-EC2-for ELB-3	i-0bad205692a63a50d	Running	t2.micro	Initializing	View alarms +	us-east-1d	ec2-5
□	Amzonlinix-ELB-Route-53-2	i-005ab1e8ae1a7342a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-3
□	ubuntu-1-ELB	i-018addcfd3e5ff2e7	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-4

i-0bad205692a63a50d (REDHAT-EC2-for ELB-3)

Details | Status and alarms New | Monitoring | Security | Networking | Storage | Tags

Instance summary Info

Public IPv4 address copied

Instance ID: i-0bad205692a63a50d (REDHAT-EC2-for ELB-3)

54.90.150.228 | open address

Private IPv4 addresses: 172.31.21.32

IPv6 address: –

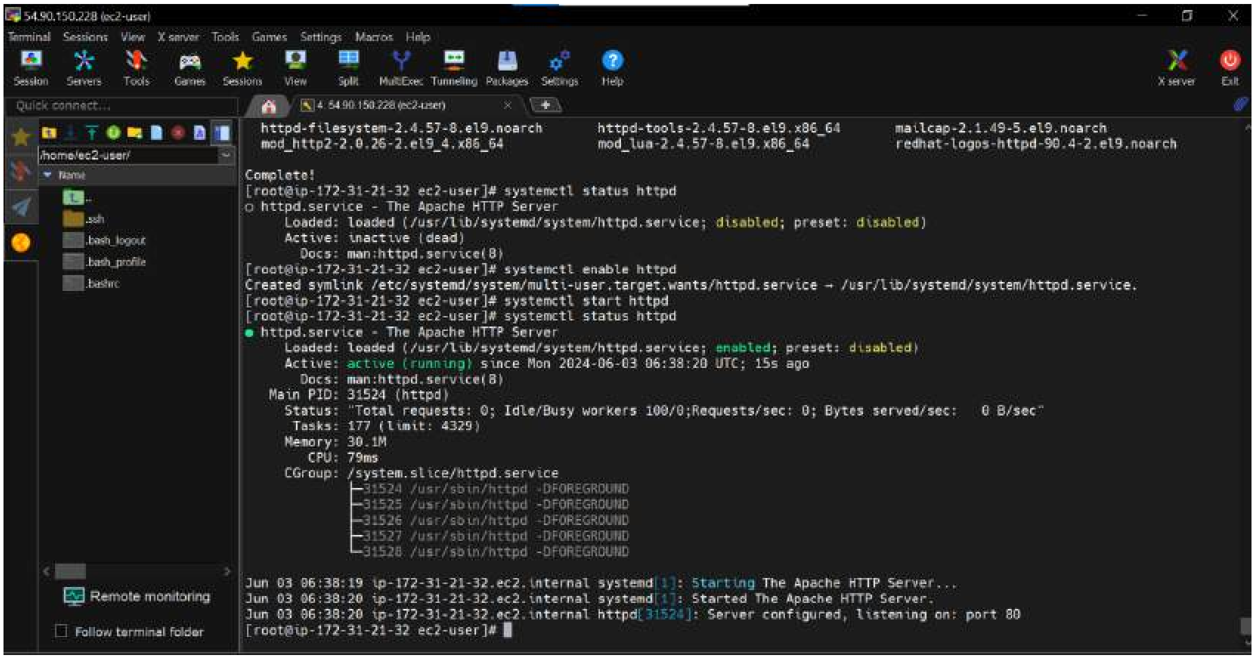
Instance state: Running

Public IPv4 DNS: ec2-54-90-150-228.compute-1.amazonaws.com | open address

Hostname type: IP name: ip-172-31-21-32.ec2.internal

Private IP DNS name (IPv4 only): ip-172-31-21-32.ec2.internal

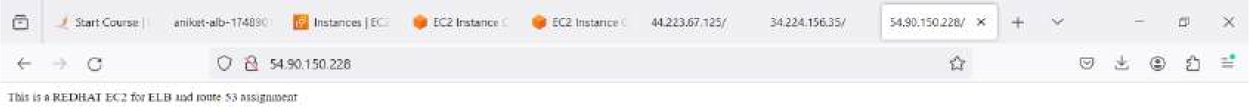
Apache server in running now and is active



54.90.150.228 (ec2-user)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit
Quick connect...
4. 54.90.150.228 (ec2-user)
/home/ec2-user/
Name
..
.ssh
.bash_logout
.bash_profile
.bashrc
Remote monitoring
Follow terminal folder

```
  httpd-filesystem-2.4.57-8.el9.noarch        httpd-tools-2.4.57-8.el9.x86_64        mailcap-2.1.49-5.el9.noarch
  mod_http2-2.0.26-2.el9_4.x86_64             mod_lua-2.4.57-8.el9.x86_64            redhat-logos-httpd-90.4-2.el9.noarch

Complete!
[root@ip-172-31-21-32 ec2-user]# systemctl status httpd
○ httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[root@ip-172-31-21-32 ec2-user]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-21-32 ec2-user]# systemctl start httpd
[root@ip-172-31-21-32 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Mon 2024-06-03 06:38:20 UTC; 15s ago
       Docs: man:httpd.service(8)
   Main PID: 31524 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 4329)
     Memory: 30.1M
        CPU: 79ms
     CGroup: /system.slice/httpd.service
             ├─31524 /usr/sbin/httpd -DFOREGROUND
             ├─31525 /usr/sbin/httpd -DFOREGROUND
             ├─31526 /usr/sbin/httpd -DFOREGROUND
             ├─31527 /usr/sbin/httpd -DFOREGROUND
             └─31528 /usr/sbin/httpd -DFOREGROUND

Jun 03 06:38:19 ip-172-31-21-32.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Jun 03 06:38:20 ip-172-31-21-32.ec2.internal systemd[1]: Started The Apache HTTP Server.
Jun 03 06:38:20 ip-172-31-21-32.ec2.internal httpd[31524]: Server configured, listening on: port 80
[root@ip-172-31-21-32 ec2-user]# 
```

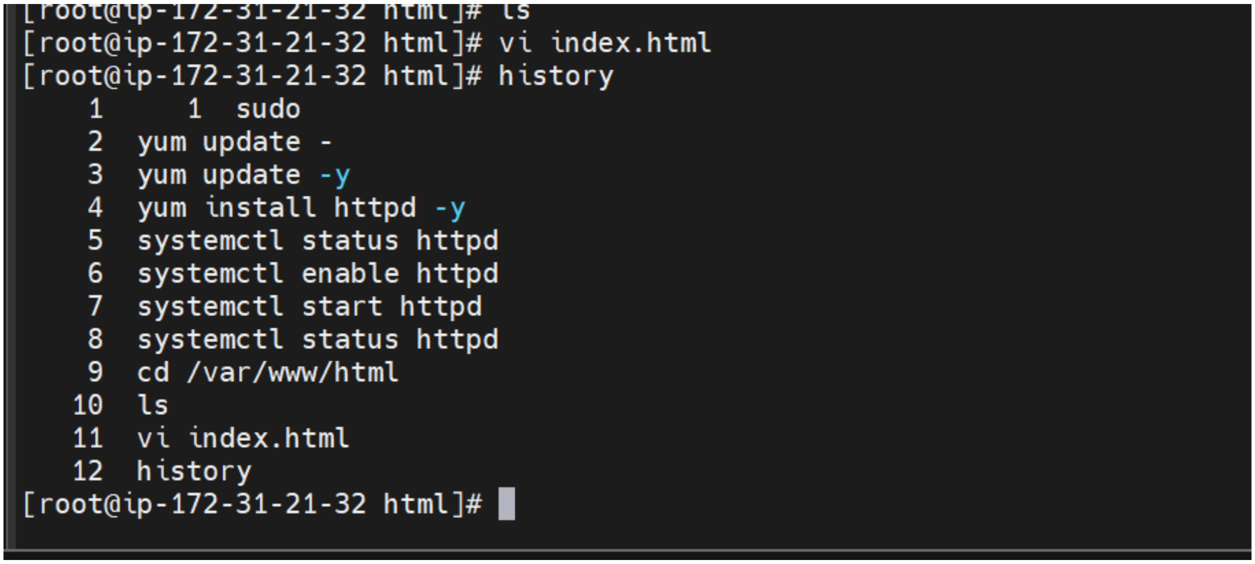
Now copy the public IP and paste in the browser



54.90.150.228

This is a REDHAT EC2 for ELB and route 53 assignment

Command run in red hat machine



```
[root@ip-172-31-21-32 html]# ls
[root@ip-172-31-21-32 html]# vi index.html
[root@ip-172-31-21-32 html]# history
    1      1  sudo
    2  yum update -
    3  yum update -y
    4  yum install httpd -y
    5  systemctl status httpd
    6  systemctl enable httpd
    7  systemctl start httpd
    8  systemctl status httpd
    9  cd /var/www/html
   10  ls
   11  vi index.html
   12  history
[root@ip-172-31-21-32 html]# 
```

Now we have to create the Target group for health check for the Targets for load balancer .

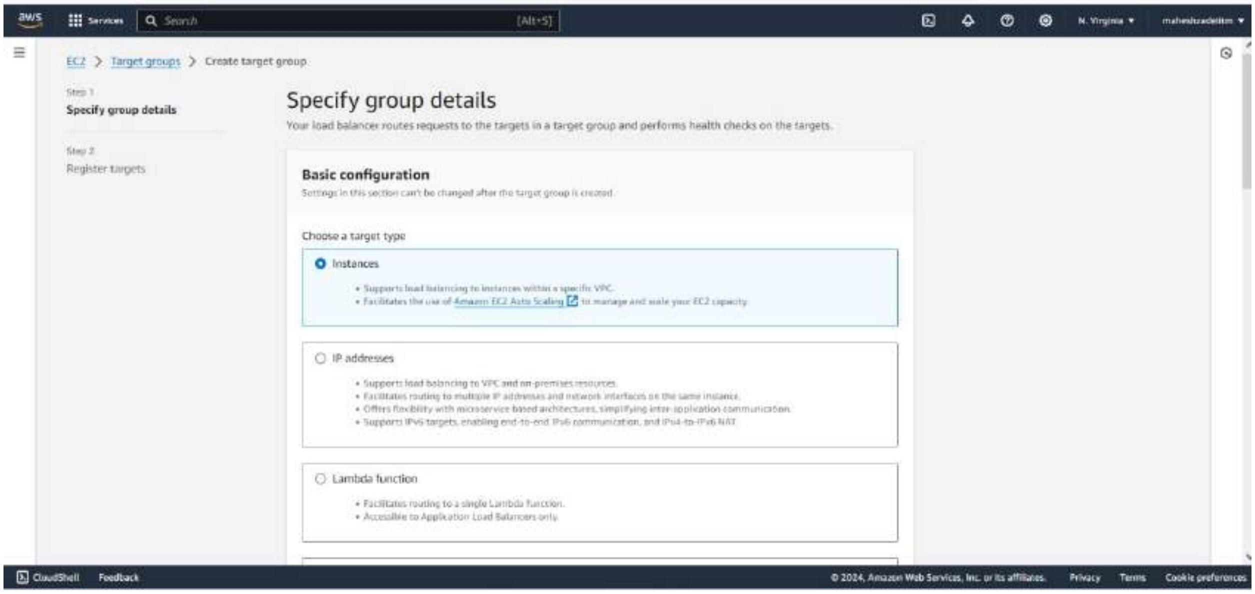
Target group checked the health status on the registered targets to remind the health status .

After this only healthy targests going to receive the traffic.

Target could be anything Lambala Funtion , EC2 , IP address , Load balancer etc.

This the primary purpose of the Target group and load Balancer.

Go to AWS Console in target group and create the Targets in N Virginia region



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 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.

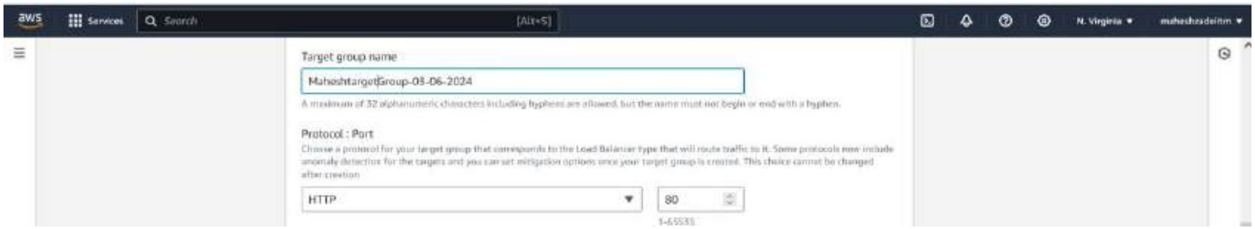
○ IP addresses

- 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.

Give the Target group name



Target group name

MaheshtargetGroup-03-06-2024

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

Protocol : Port

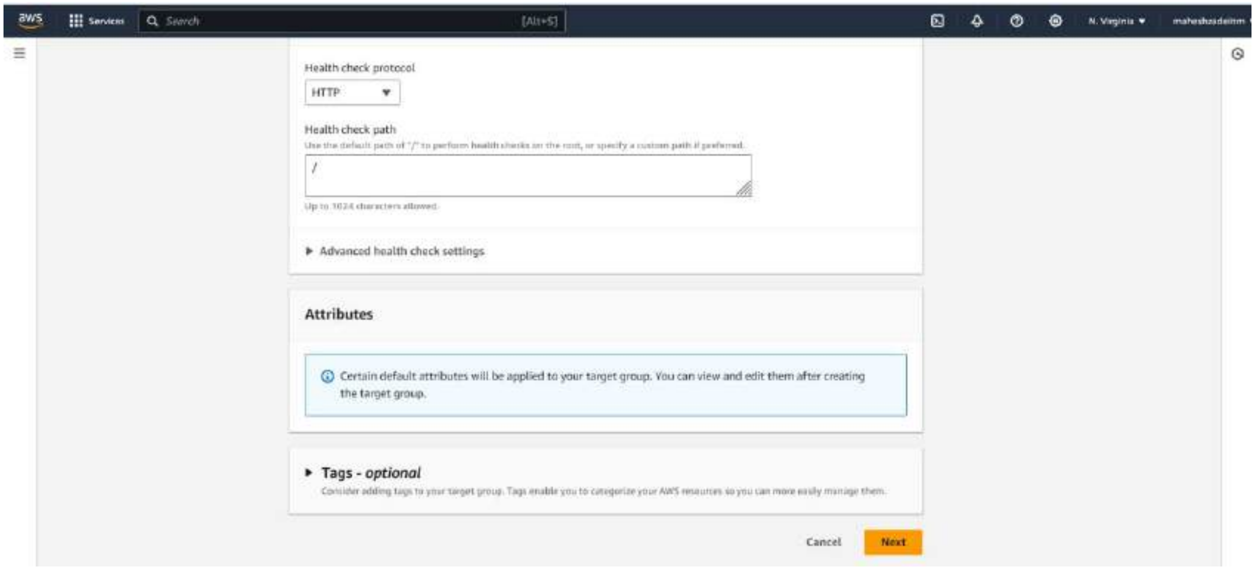
Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

HTTP ▼ 80

1-65535

Keep the http port as 80

Keep the all the setting as it is and click on Next



Health check protocol

HTTP ▼

Health check path

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

/

Up to 1024 characters allowed.

▸ Advanced health check settings

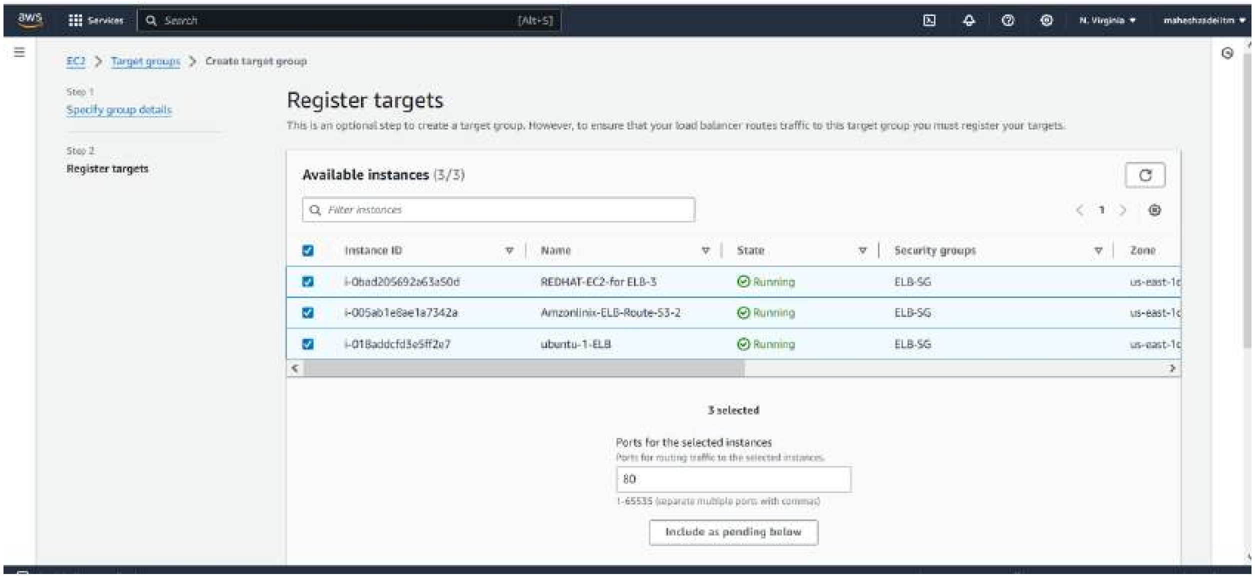
Attributes

Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

▸ Tags - *optional*

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

Click on registered targets



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 (3/3)

☑	Instance ID	Name	State	Security groups	Zone
☑	i-0bad205692a63a50d	REDHAT-EC2-for ELB-3	Running	ELB-SG	us-east-1c
☑	i-005ab1e8ae1a7342a	Amzonlinix-ELB-Route-53-2	Running	ELB-SG	us-east-1c
☑	i-018addcfd3e5ff2e7	ubuntu-1-ELB	Running	ELB-SG	us-east-1c

3 selected

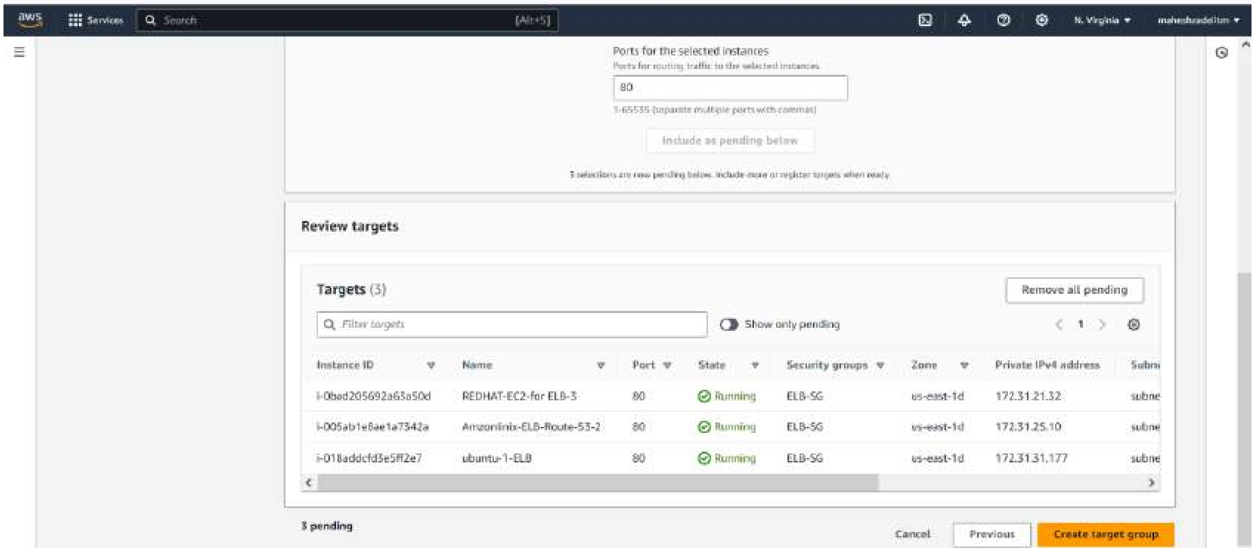
Ports for the selected instances

Ports for routing traffic to the selected instances.

80

1-65535 (separate multiple ports with commas)

And now click on include as pending



Ports for the selected instances

Ports for routing traffic to the selected instances

80

1-65535 (separate multiple ports with commas)

3 selections are now pending below. Include more or register targets when ready.

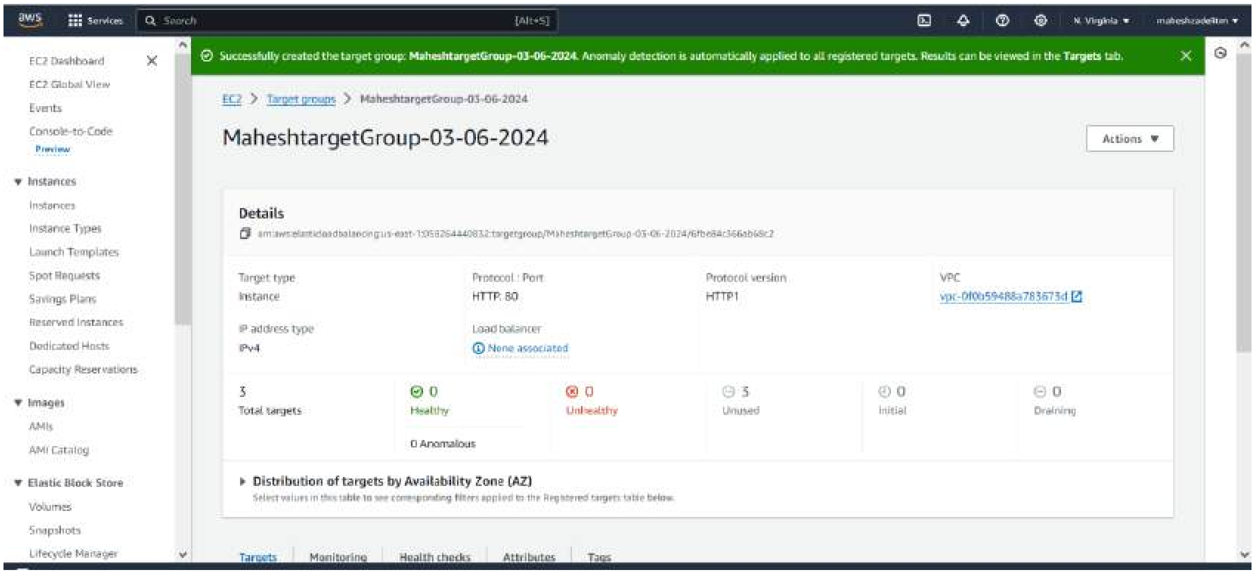
Review targets

Targets (3)

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subne
i-0bad205692a63a50d	REDHAT-EC2-for ELB-3	80	✓ Running	ELB-SG	us-east-1d	172.31.21.32	subne
i-005ab1e8ae1a7342a	Amzonlinix-ELB-Route-53-2	80	✓ Running	ELB-SG	us-east-1d	172.31.25.10	subne
i-018addcfd3e5ff2e7	ubuntu-1-ELB	80	✓ Running	ELB-SG	us-east-1d	172.31.31.177	subne

3 pending

Now click on create Target group



Successfully created the target group: **MaheshtargetGroup-03-06-2024**. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the **Targets** tab.

EC2 > Target groups > MaheshtargetGroup-03-06-2024

MaheshtargetGroup-03-06-2024

Details

arn:aws:elasticloadbalancing:us-east-1:058264440832:targetgroup/MaheshtargetGroup-03-06-2024/6fbe84c366ab68c2

Target type: Instance

Protocol : Port: HTTP: 80

Protocol version: HTTP1

VPC: vpc-0f0b59488a783673d

IP address type: IPv4

Load balancer: None associated

3 Total targets

0 Healthy

0 Unhealthy

3 Unused

0 Initial

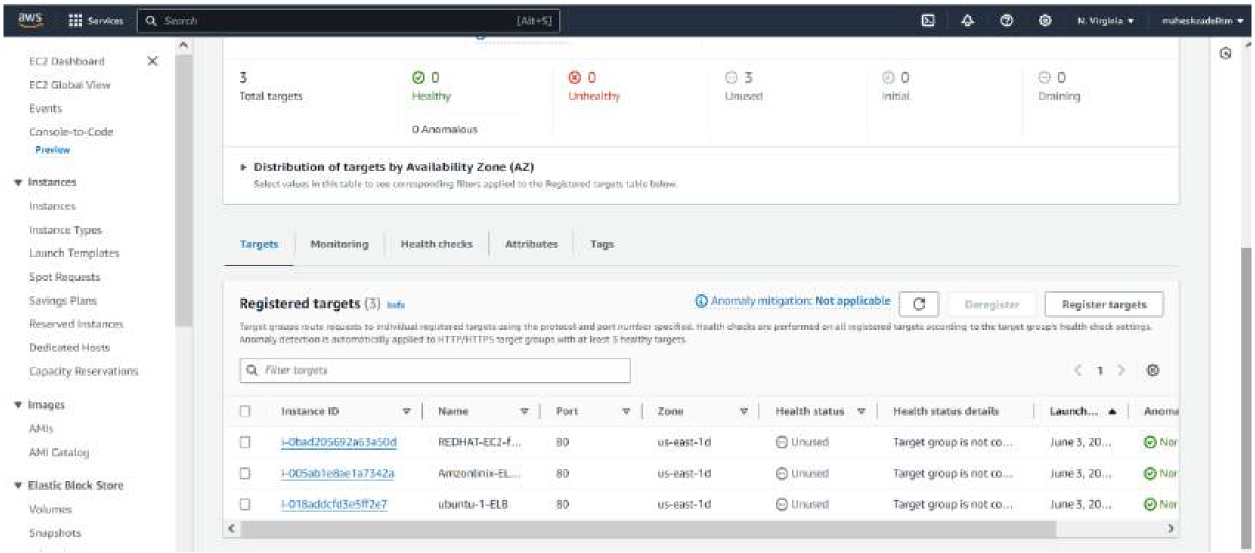
0 Draining

0 Anomalous

Distribution of targets by Availability Zone (AZ)

Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets are showing unused



3	0	0	3	0	0
Total targets	Healthy	Unhealthy	Unused	Initial	Draining
	0 Anomalous				

Distribution of targets by Availability Zone (AZ)

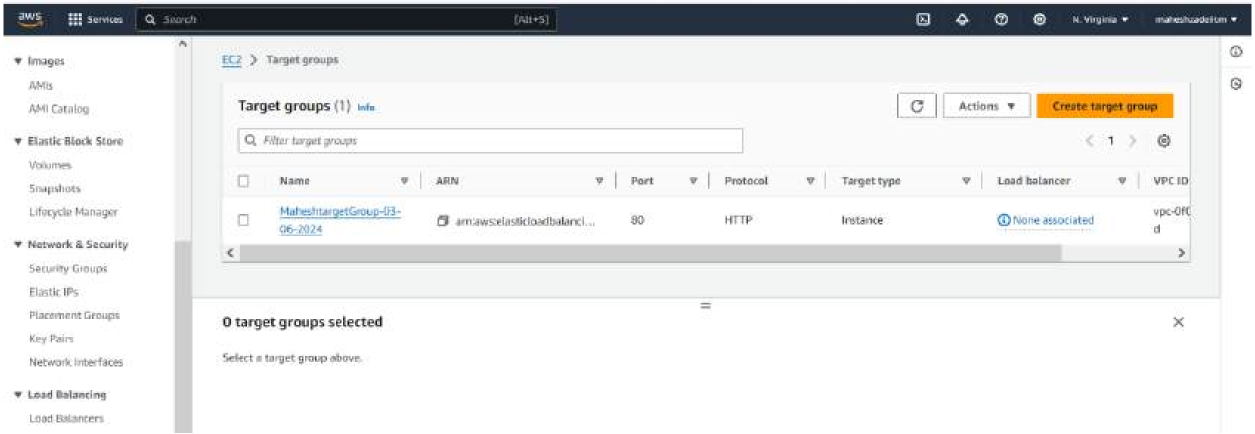
Select values in this table to see corresponding filters applied to the Registered targets table below.

Registered targets (3) Info

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

□	Instance ID	Name	Port	Zone	Health status	Health status details	Launch...	Anoma
□	i-0bad205692a63a50d	REDHAT-EC2-f...	80	us-east-1d	Unused	Target group is not co...	June 3, 20...	Nor
□	i-005ab1e8ae1a7342a	Amzonlinix-EL...	80	us-east-1d	Unused	Target group is not co...	June 3, 20...	Nor
□	i-018addcfd3e5ff2e7	ubuntu-1-ELB	80	us-east-1d	Unused	Target group is not co...	June 3, 20...	Nor

TG Created



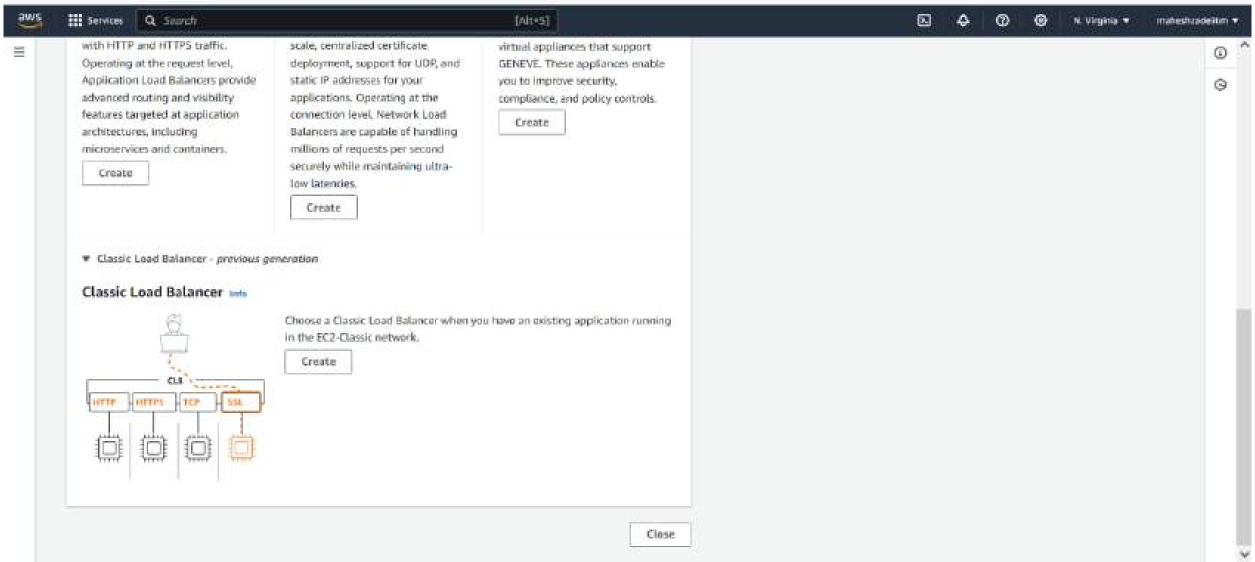
Target groups (1) Info

□	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
□	MaheshtargetGroup-03-06-2024	arn:aws:elasticloadbalanci...	80	HTTP	Instance	None associated	vpc-0f0 d

0 target groups selected

Select a target group above.

Now Go to AWS Console and Create Load Balancer (Classic Load Balancer)



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

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

virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create

▼ Classic Load Balancer - *previous generation*

Classic Load Balancer Info

Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network.

Create

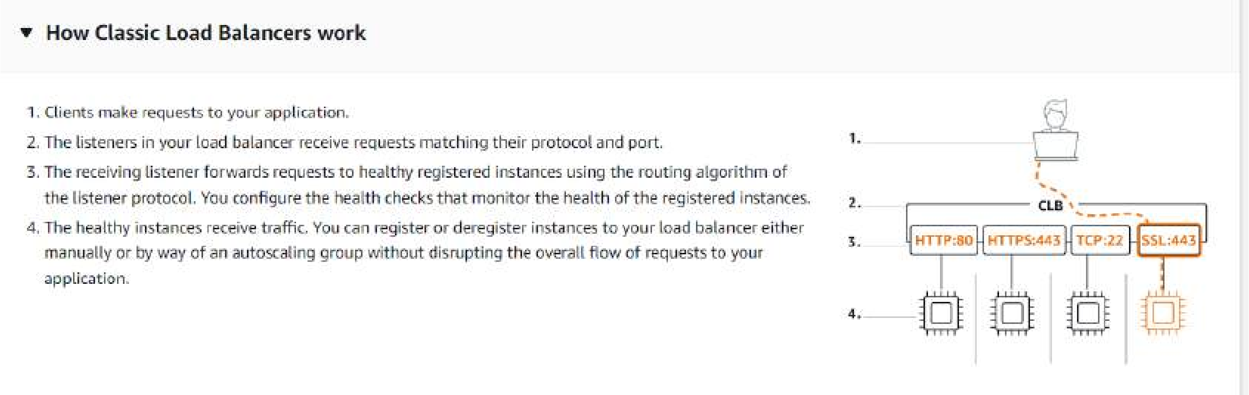
Close

Click on create

Classic Load Balancer

The Classic Load Balancer distributes incoming application traffic across multiple EC2 instance targets in multiple Availability Zones. This increases the fault tolerance of your applications. Elastic Load Balancing detects unhealthy instances and routes traffic only to healthy instances.

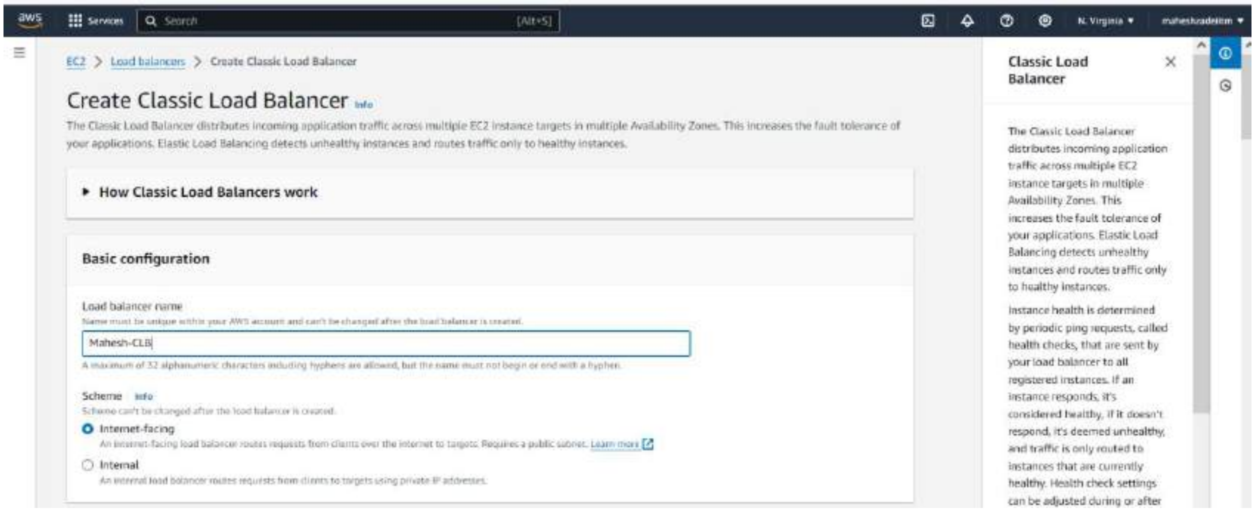
Instance health is determined by periodic ping requests, called health checks, that are sent by your load balancer to all registered instances. If an instance responds, it's considered healthy, if it doesn't respond, it's deemed unhealthy, and traffic is only routed to instances that are currently healthy. Health check settings can be adjusted during or after the creation of your load balancer.



▼ **How Classic Load Balancers work**

1. Clients make requests to your application.
2. The listeners in your load balancer receive requests matching their protocol and port.
3. The receiving listener forwards requests to healthy registered instances using the routing algorithm of the listener protocol. You configure the health checks that monitor the health of the registered instances.
4. The healthy instances receive traffic. You can register or deregister instances to your load balancer either manually or by way of an autoscaling group without disrupting the overall flow of requests to your application.

Give the Name of Load balancer



EC2 > Load balancers > Create Classic Load Balancer

Create Classic Load Balancer Info

The Classic Load Balancer distributes incoming application traffic across multiple EC2 instance targets in multiple Availability Zones. This increases the fault tolerance of your applications. Elastic Load Balancing detects unhealthy instances and routes traffic only to healthy instances.

▸ How Classic 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.

Mahesh-CLB

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

Scheme Info

Scheme can't be changed after the load balancer is created.

● Internet-facing

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more

○ Internal

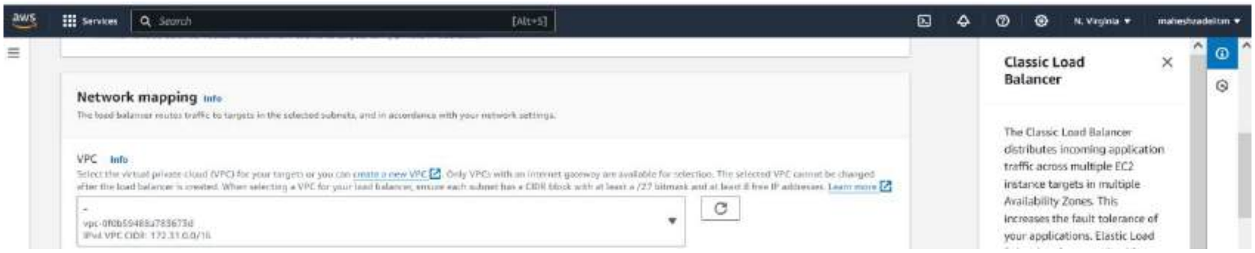
An internal load balancer routes requests from clients to targets using private IP addresses.

Classic Load Balancer

The Classic Load Balancer distributes incoming application traffic across multiple EC2 instance targets in multiple Availability Zones. This increases the fault tolerance of your applications. Elastic Load Balancing detects unhealthy instances and routes traffic only to healthy instances.

Instance health is determined by periodic ping requests, called health checks, that are sent by your load balancer to all registered instances. If an instance responds, it's considered healthy, if it doesn't respond, it's deemed unhealthy, and traffic is only routed to instances that are currently healthy. Health check settings can be adjusted during or after

Select default VPC



Network mapping Info

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

VPC Info

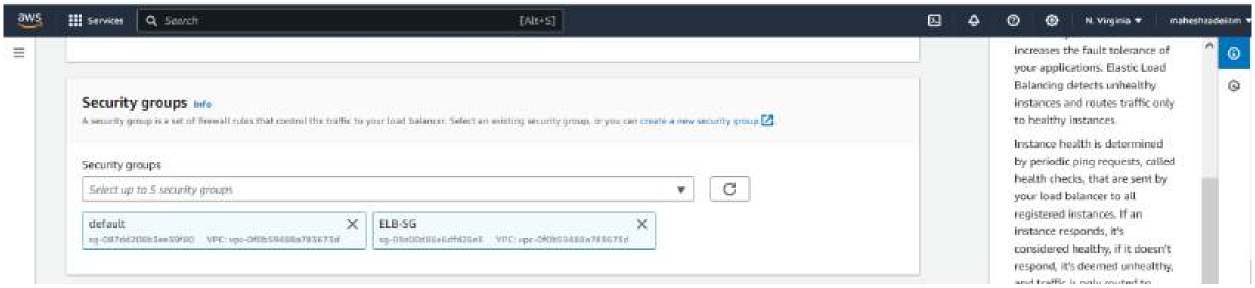
Select the virtual private cloud (VPC) for your targets or you can create a new VPC. Only VPCs with an internet gateway are available for selection. The selected VPC cannot be changed after the load balancer is created. When selecting a VPC for your load balancer, ensure each subnet has a CIDR block with at least a /27 bitmask and at least 8 free IP addresses. Learn more

-
vpc-0f0b59488a783673d
IPv4 VPC CIDR: 172.31.0.0/16

Classic Load Balancer

The Classic Load Balancer distributes incoming application traffic across multiple EC2 instance targets in multiple Availability Zones. This increases the fault tolerance of your applications. Elastic Load

Select SG which we created



Security groups Info

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

Security groups

Select up to 5 security groups

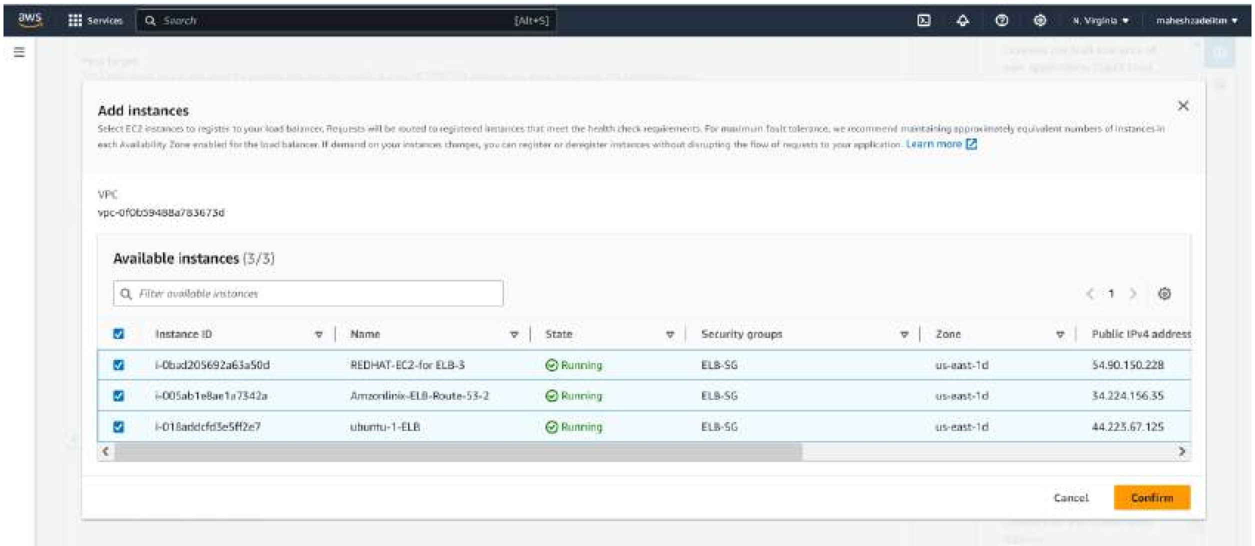
default
sg-087dd2006b1ae59f00 VPC: vpc-0f0b59488a783673d

ELB-SG
sg-0be00e86a6ffd26e8 VPC: vpc-0f0b59488a783673d

increases the fault tolerance of your applications. Elastic Load Balancing detects unhealthy instances and routes traffic only to healthy instances.

Instance health is determined by periodic ping requests, called health checks, that are sent by your load balancer to all registered instances. If an instance responds, it's considered healthy, if it doesn't respond, it's deemed unhealthy,

Add the instances



Add instances

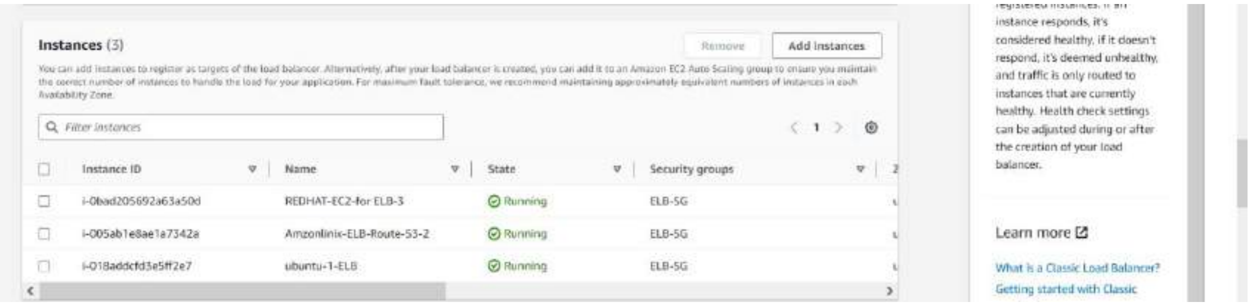
Select EC2 instances to register to your load balancer. Requests will be routed to registered instances that meet the health check requirements. For maximum fault tolerance, we recommend maintaining approximately equivalent numbers of instances in each Availability Zone enabled for the load balancer. If demand on your instances changes, you can register or deregister instances without disrupting the flow of requests to your application. Learn more

VPC
vpc-0f0b59488a783673d

Available instances (3/3)

☑	Instance ID	Name	State	Security groups	Zone	Public IPv4 address
☑	i-0bad205692a63a50d	REDHAT-EC2-for ELB-3	Running	ELB-SG	us-east-1d	54.90.150.228
☑	i-005ab1e8ae1a7342a	Amzonlinix-ELB-Route-53-2	Running	ELB-SG	us-east-1d	34.224.156.35
☑	i-018addcfd3e5ff2e7	ubuntu-1-ELB	Running	ELB-SG	us-east-1d	44.223.67.125

Added instances



Instances (3)

You can add instances to register as targets of the load balancer. Alternatively, after your load balancer is created, you can add it to an Amazon EC2 Auto Scaling group to ensure you maintain the correct number of instances to handle the load for your application. For maximum fault tolerance, we recommend maintaining approximately equivalent numbers of instances in each Availability Zone.

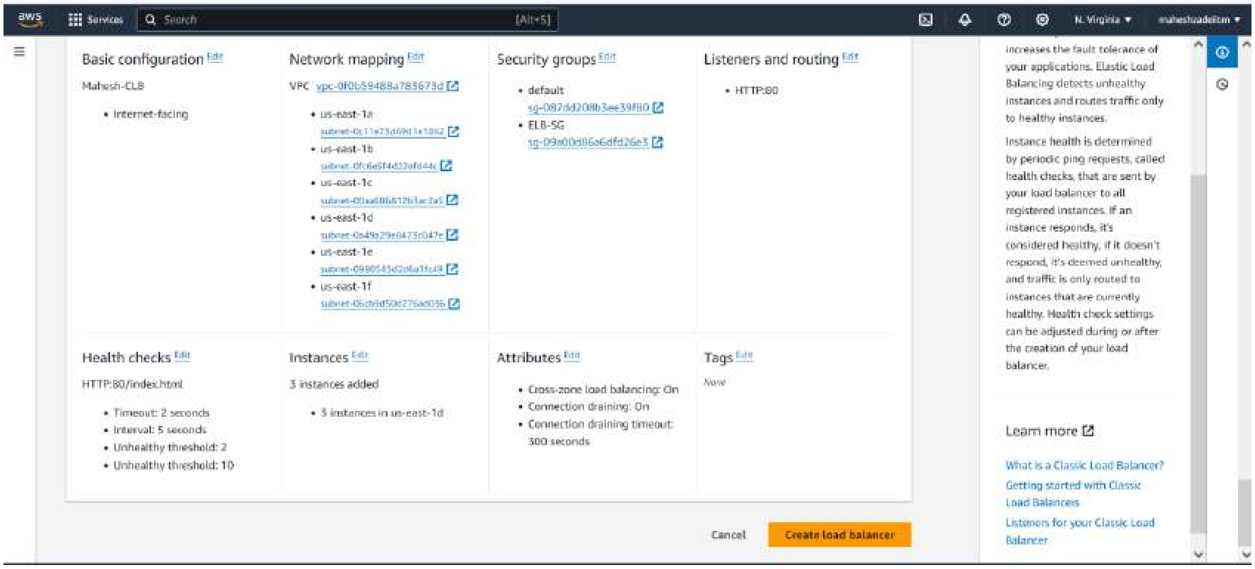
□	Instance ID	Name	State	Security groups
□	i-0bad205692a63a50d	REDHAT-EC2-for ELB-3	Running	ELB-SG
□	i-005ab1e8ae1a7342a	Amzonlinix-ELB-Route-53-2	Running	ELB-SG
□	i-018addcfd3e5ff2e7	ubuntu-1-ELB	Running	ELB-SG

registered instances. If an instance responds, it's considered healthy. If it doesn't respond, it's deemed unhealthy, and traffic is only routed to instances that are currently healthy. Health check settings can be adjusted during or after the creation of your load balancer.

Learn more

What is a Classic Load Balancer?
Getting started with Classic

Keep the remaining setting as it is



Basic configuration Edit
Mahesh-CLB
- Internet-facing

Network mapping Edit
VPC vpc-0f0b59488a783673d
- us-east-1a
  subnet-0c11e73d69d1e1062
- us-east-1b
  subnet-0fc6e5f4d23afd44c
- us-east-1c
  subnet-00aa686812b1ac2a5
- us-east-1d
  subnet-0e49a29e0473c047e
- us-east-1e
  subnet-0980545d2d6a1fc49
- us-east-1f
  subnet-06cb9d50d276ad056

Security groups Edit
- default
  sg-087dd208b3ee39f80
- ELB-SG
  sg-09a00d86a6dfd26e3

Listeners and routing Edit
- HTTP:80

Health checks Edit
HTTP:80/index.html
- Timeout: 2 seconds
- Interval: 5 seconds
- Unhealthy threshold: 2
- Unhealthy threshold: 10

Instances Edit
3 instances added
- 3 instances in us-east-1d

Attributes Edit
- Cross-zone load balancing: On
- Connection draining: On
- Connection draining timeout: 300 seconds

Tags Edit
None

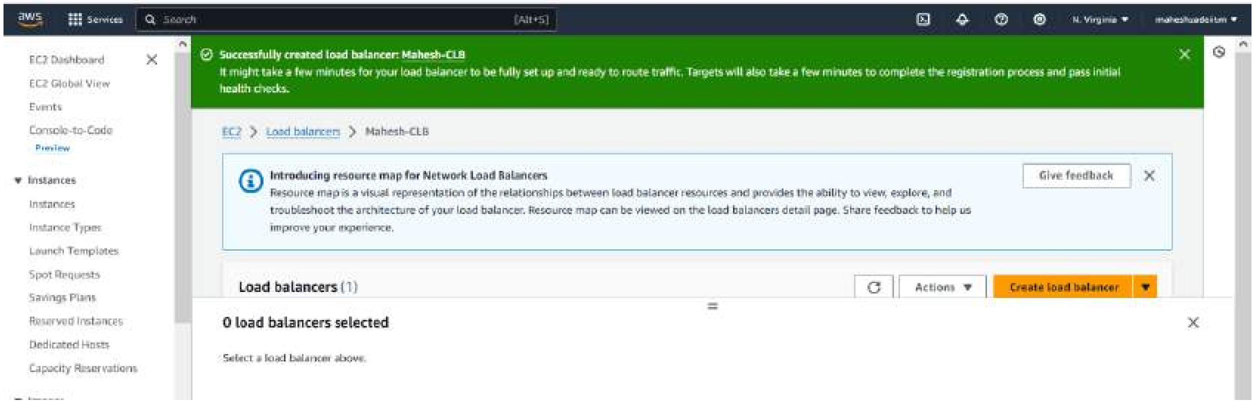
increases the fault tolerance of your applications. Elastic Load Balancing detects unhealthy instances and routes traffic only to healthy instances.

Instance health is determined by periodic ping requests, called health checks, that are sent by your load balancer to all registered instances. If an instance responds, it's considered healthy. If it doesn't respond, it's deemed unhealthy, and traffic is only routed to instances that are currently healthy. Health check settings can be adjusted during or after the creation of your load balancer.

Learn more

What is a Classic Load Balancer?
Getting started with Classic Load Balancers
Listeners for your Classic Load Balancer

Click on Create load balancer



Successfully created load balancer: Mahesh-CLB
It might take a few minutes for your load balancer to be fully set up and ready to route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

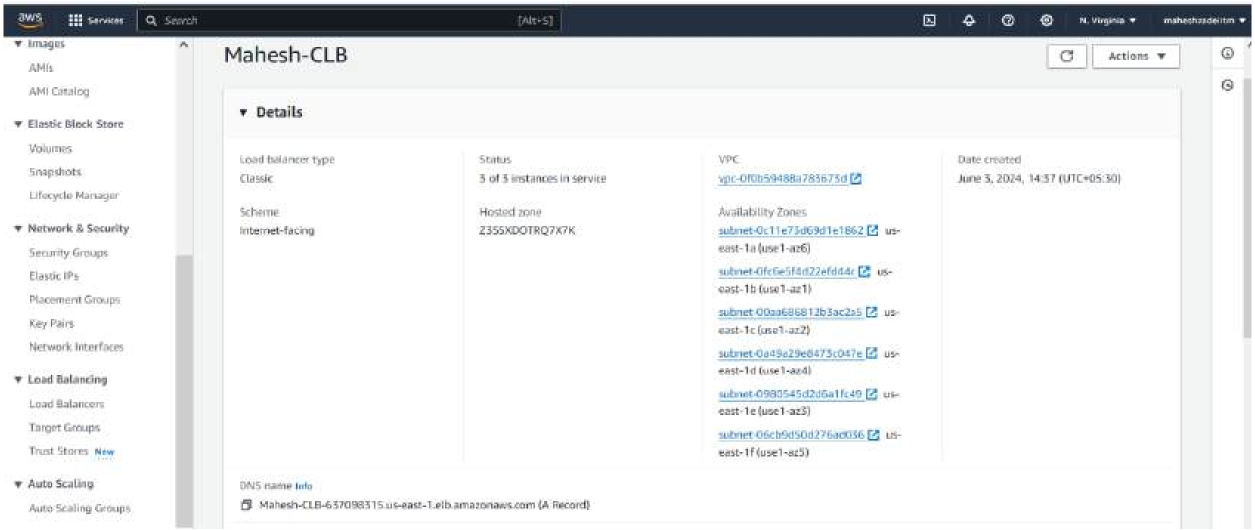
Introducing resource map for Network Load Balancers
Resource map is a visual representation of the relationships between load balancer resources and provides the ability to view, explore, and troubleshoot the architecture of your load balancer. Resource map can be viewed on the load balancers detail page. Share feedback to help us improve your experience.

Load balancers (1)

0 load balancers selected

Select a load balancer above.

LB Created



Mahesh-CLB

Details

Load balancer type
Classic

Status
3 of 3 instances in service

VPC
vpc-0f0b59488a783673d

Date created
June 3, 2024, 14:37 (UTC+05:30)

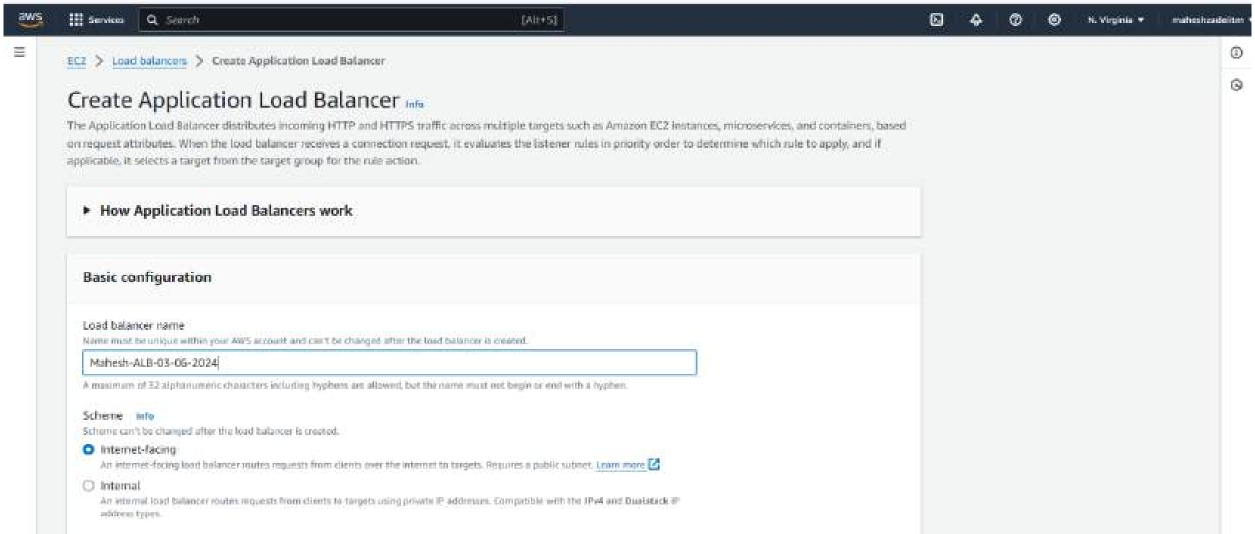
Scheme
Internet-facing

Hosted zone
Z35SXDOTRQ7X7K

Availability Zones
subnet-0c11e73d69d1e1862 us-east-1a (use1-az6)
subnet-0fc6e5f4d22efdd4c us-east-1b (use1-az1)
subnet-00aa686812b3ac2a5 us-east-1c (use1-az2)
subnet-0a49a29e8473c047e us-east-1d (use1-az4)
subnet-0980545d2d6a1fc49 us-east-1e (use1-az3)
subnet-06cb9d50d276ad036 us-east-1f (use1-az5)

DNS name Info
Mahesh-CLB-637098315.us-east-1.elb.amazonaws.com (A Record)

Now create the Application load Balancer



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.

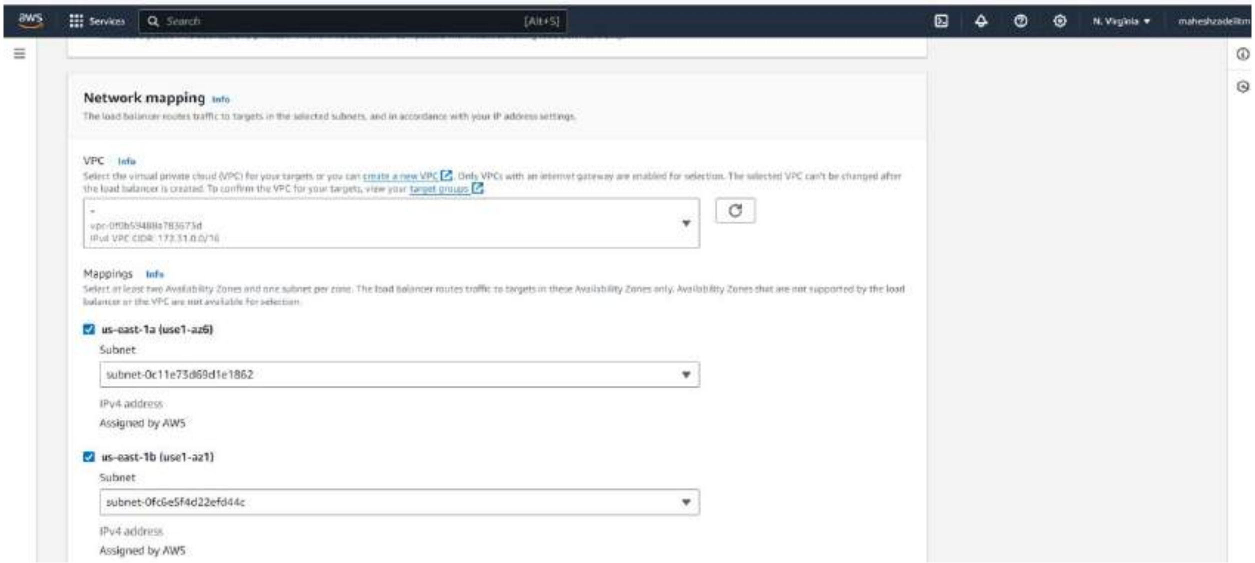
▸ How Application Load Balancers work

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.
Mahesh-ALB-03-06-2024
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info
Scheme can't be changed after the load balancer is created.
● Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more
○ Internal
An internal load balancer routes requests from clients to targets using private IP addresses. Compatible with the IPv4 and Dualstack IP address types.

MAP all the AZ for high availability



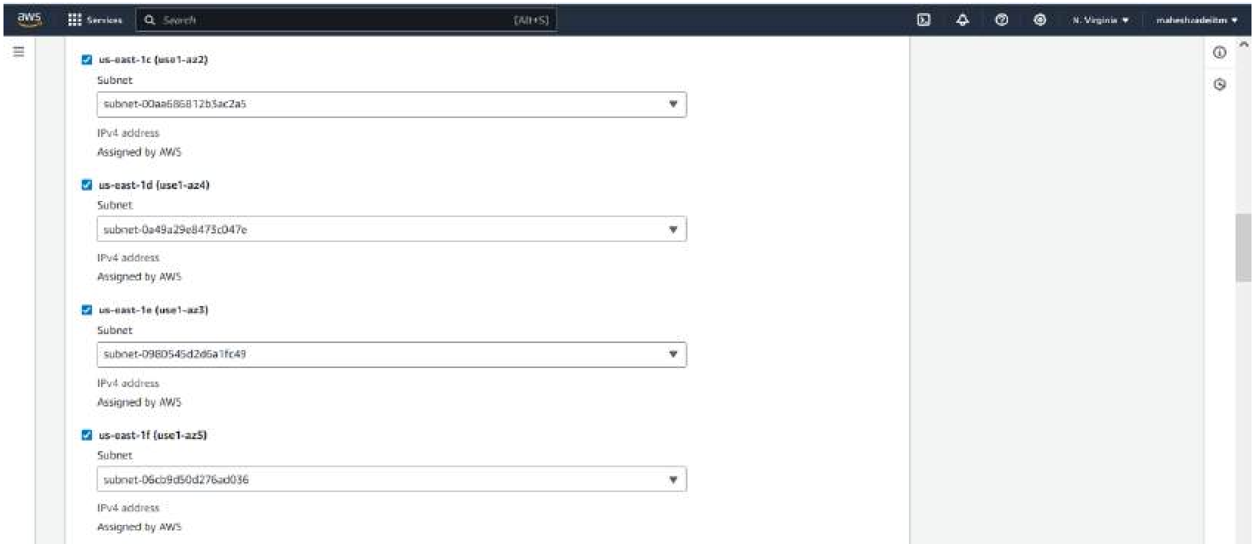
Network mapping Info
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info
Select the virtual private cloud (VPC) for your targets or you can create a new VPC. Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your target groups.
-
vpc-0f0b59488a783673d
IPv4 VPC CIDR: 172.31.0.0/16

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.

☑ us-east-1a (use1-az6)
Subnet
subnet-0c11e73d69d1e1862
IPv4 address
Assigned by AWS

☑ us-east-1b (use1-az1)
Subnet
subnet-0fc6e5f4d22efd44c
IPv4 address
Assigned by AWS



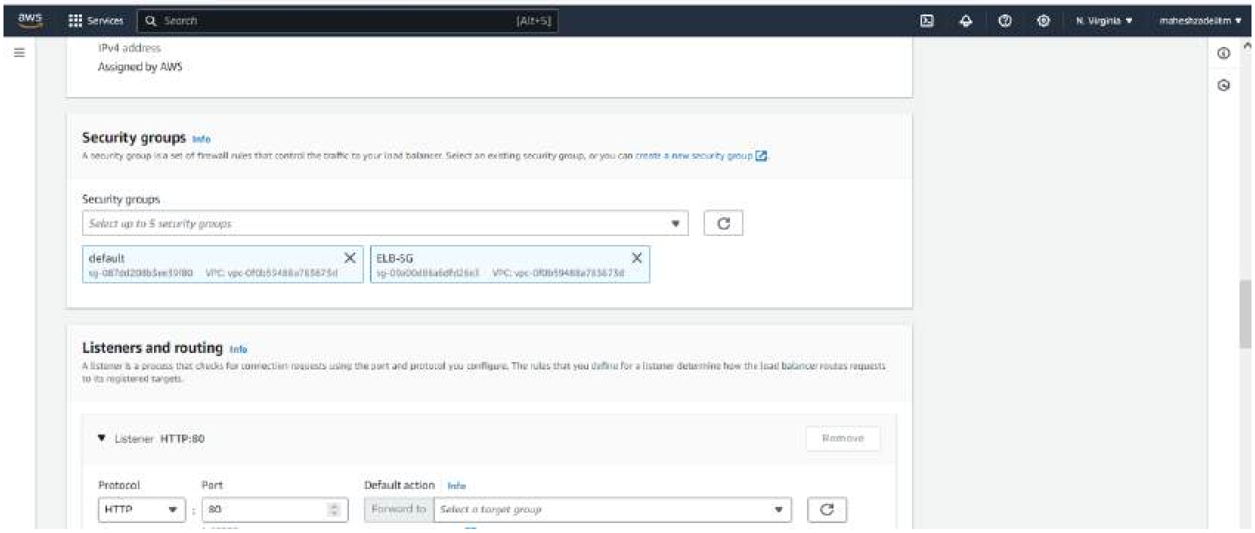
☑ us-east-1c (use1-az2)
Subnet
subnet-00aa686812b5ac2a5
IPv4 address
Assigned by AWS

☑ us-east-1d (use1-az4)
Subnet
subnet-0a49a29e8473c047e
IPv4 address
Assigned by AWS

☑ us-east-1e (use1-az3)
Subnet
subnet-0980545d2d6a1fc49
IPv4 address
Assigned by AWS

☑ us-east-1f (use1-az5)
Subnet
subnet-06cb9d50d276ad036
IPv4 address
Assigned by AWS

Select SG which we created.



IPv4 address
Assigned by AWS

Security groups

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

Security groups
Select up to 5 security groups

default
sg-087dd208b5ee39080 VPC: vpc-0f0b59488a785875d

ELB-SG
sg-00d00d86a6dfd26e3 VPC: vpc-0f0b59488a785875d

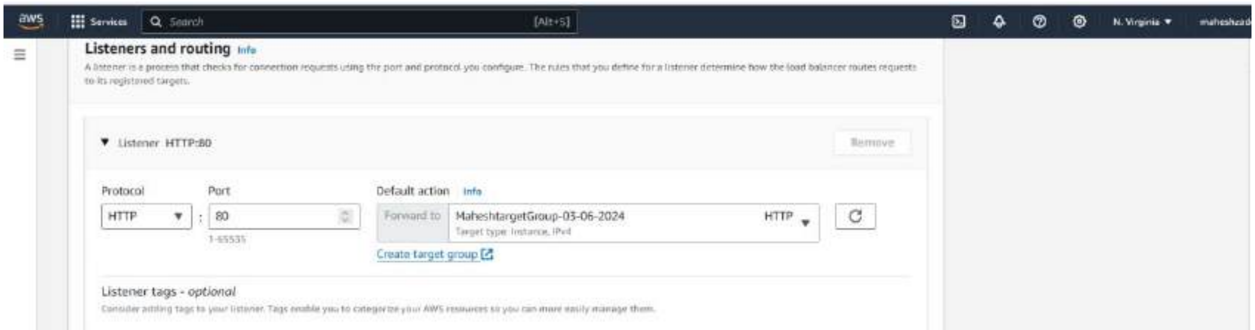
Listeners and routing

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▾ Listener HTTP:80

Protocol: HTTP
Port: 80
Default action: Forward to Select a target group

Select the TG which we created



Listeners and routing

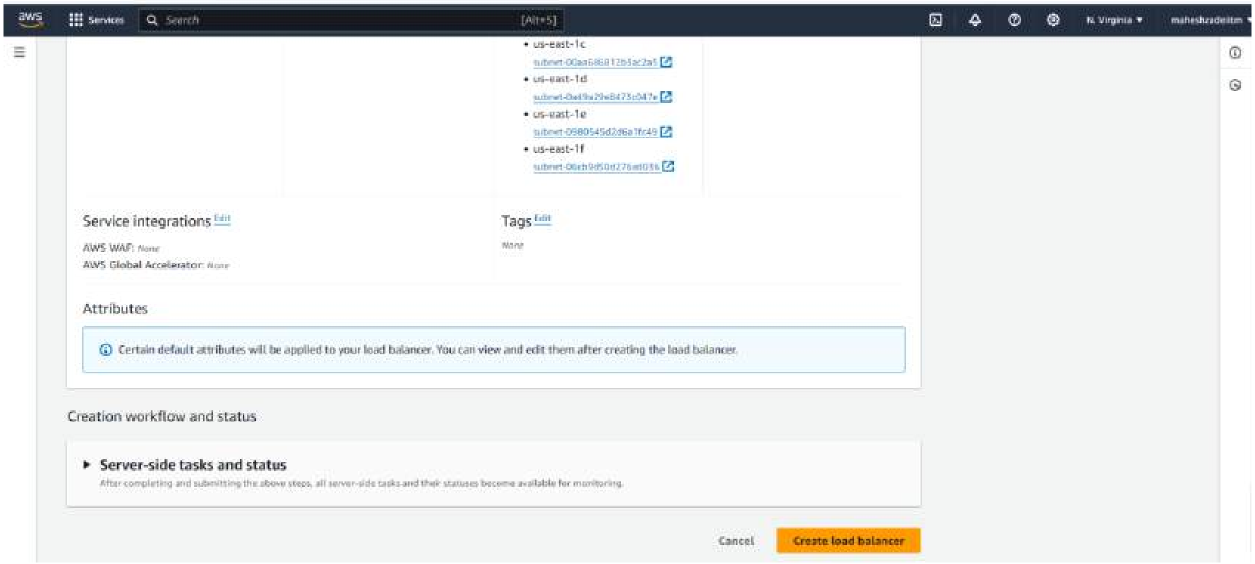
A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▾ Listener HTTP:80

Protocol: HTTP
Port: 80
1-65535
Default action: Forward to MaheshtargetGroup-03-06-2024 HTTP
Target type: Instance, IPv4
Create target group

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

Review the setting and click on create Load balancer



- us-east-1c
  subnet-00aa686812b3ac2a5
- us-east-1d
  subnet-0a49a29e8473c047e
- us-east-1e
  subnet-0980545d2d6a1fc49
- us-east-1f
  subnet-06cb9d50d276ad036

Service integrations Edit

AWS WAF: None

AWS Global Accelerator: None

Tags Edit

None

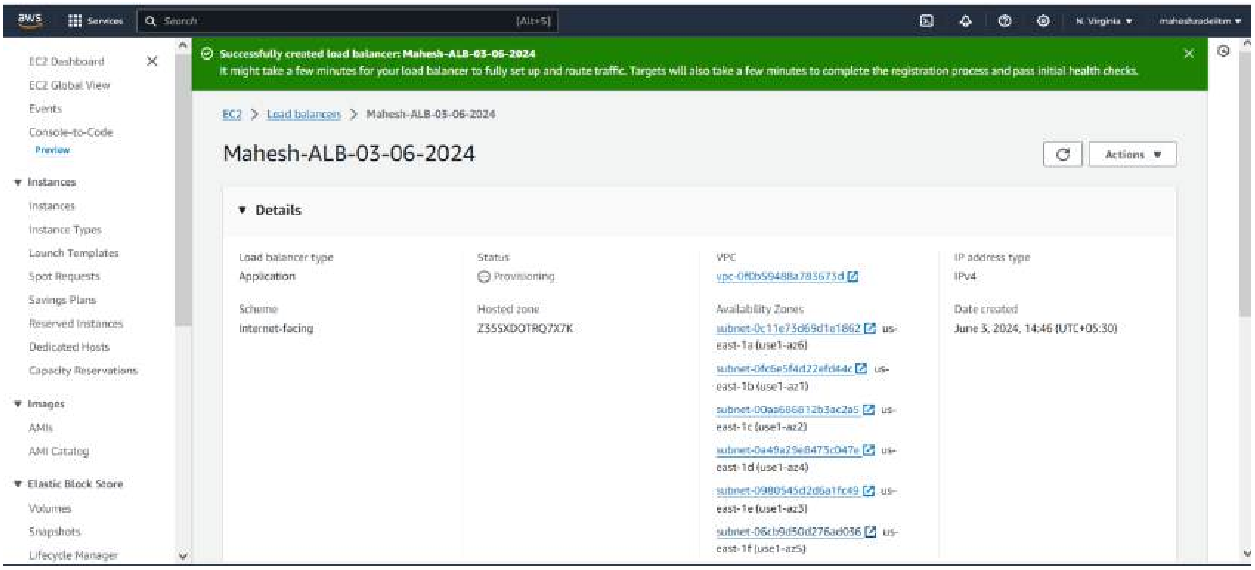
Attributes

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

Creation workflow and status

▸ Server-side tasks and status

After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.



Successfully created load balancer: Mahesh-ALB-03-06-2024

It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

EC2 > Load balancers > Mahesh-ALB-03-06-2024

Mahesh-ALB-03-06-2024

▾ Details

Load balancer type: Application

Scheme: Internet-facing

Status: Provisioning

Hosted zone: Z35SXDOTRQ7X7K

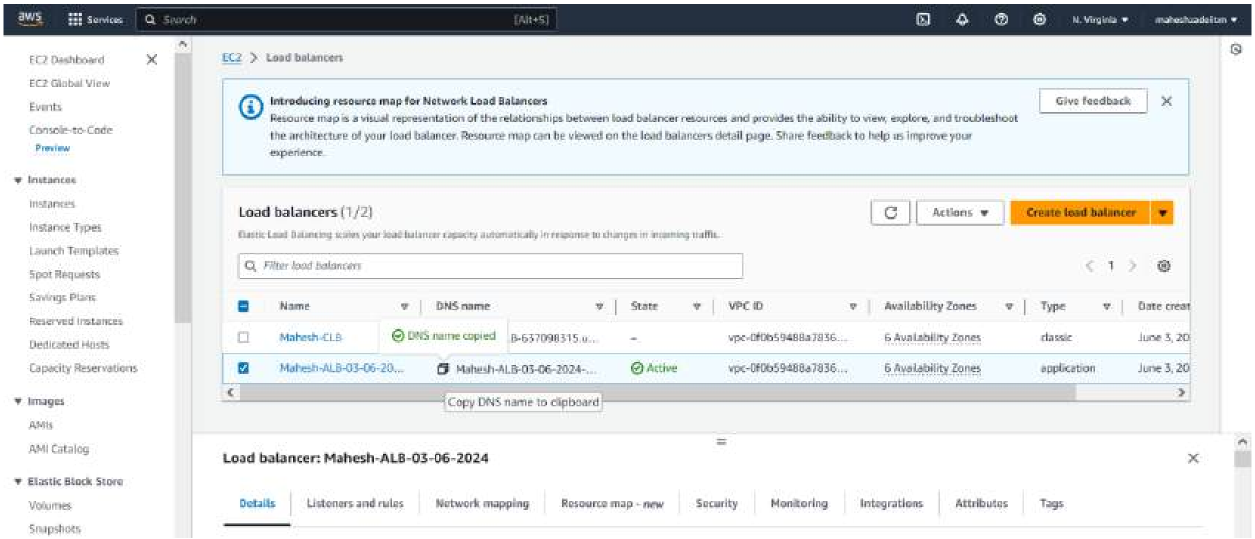
VPC: vpc-0f0b59488a783673d

Availability Zones:
subnet-0c11e73d69d1e1862 us-east-1a (use1-az6)
subnet-0fc6e5f4d22efd44c us-east-1b (use1-az1)
subnet-00aa686812b3ac2a5 us-east-1c (use1-az2)
subnet-0a49a29e8473c047e us-east-1d (use1-az4)
subnet-0980545d2d6a1fc49 us-east-1e (use1-az3)
subnet-06cb9d50d276ad036 us-east-1f (use1-az5)

IP address type: IPv4

Date created: June 3, 2024, 14:46 (UTC+05:30)

Copy the DNS for ALB



Introducing resource map for Network Load Balancers
Resource map is a visual representation of the relationships between load balancer resources and provides the ability to view, explore, and troubleshoot the architecture of your load balancer. Resource map can be viewed on the load balancers detail page. Share feedback to help us improve your experience.

Load balancers (1/2)

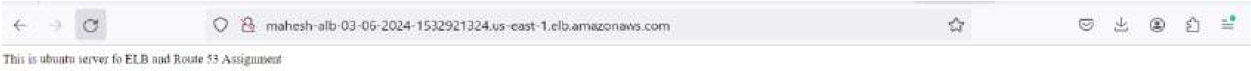
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

	Name	DNS name	State	VPC ID	Availability Zones	Type	Date creat
☐	Mahesh-CLB	✓ DNS name copied B-637098315.u...	-	vpc-0f0b59488a7836...	6 Availability Zones	classic	June 3, 20
☑	Mahesh-ALB-03-06-20...	Mahesh-ALB-03-06-2024-...	✓ Active	vpc-0f0b59488a7836...	6 Availability Zones	application	June 3, 20

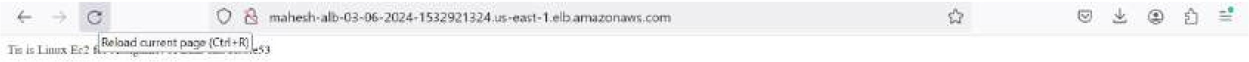
Load balancer: Mahesh-ALB-03-06-2024

DNS of ALB

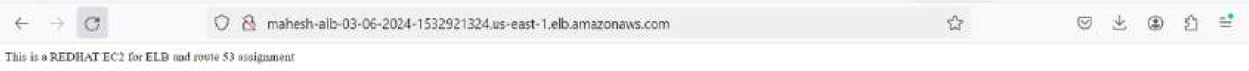
Mahesh-ALB-03-06-2024-1532921324.us-east-1.elb.amazonaws.com


mahesh-alb-03-06-2024-1532921324.us-east-1.elb.amazonaws.com

This is ubuntu server fo ELB and Route 53 Assignment


mahesh-alb-03-06-2024-1532921324.us-east-1.elb.amazonaws.com

Tis is Linux Ec2 f[illegible]e53

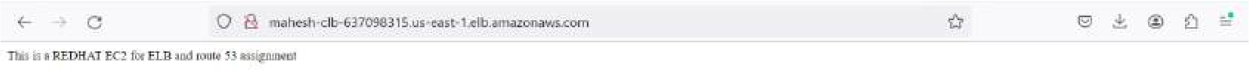

mahesh-alb-03-06-2024-1532921324.us-east-1.elb.amazonaws.com

This is a REDHAT EC2 for ELB and route 53 assignment

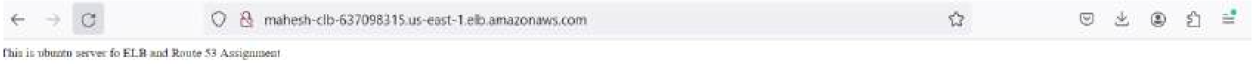
Now Classic Load Balcner is also working

DNS CLB

http://mahesh-clb-637098315.us-east-1.elb.amazonaws.com/

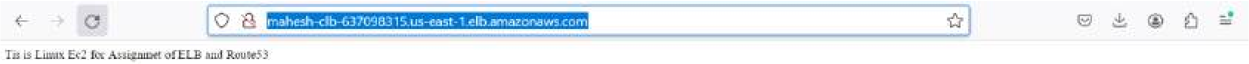

mahesh-clb-637098315.us-east-1.elb.amazonaws.com

This is a REDHAT EC2 for ELB and route 53 assignment



mahesh-clb-637098315.us-east-1.elb.amazonaws.com

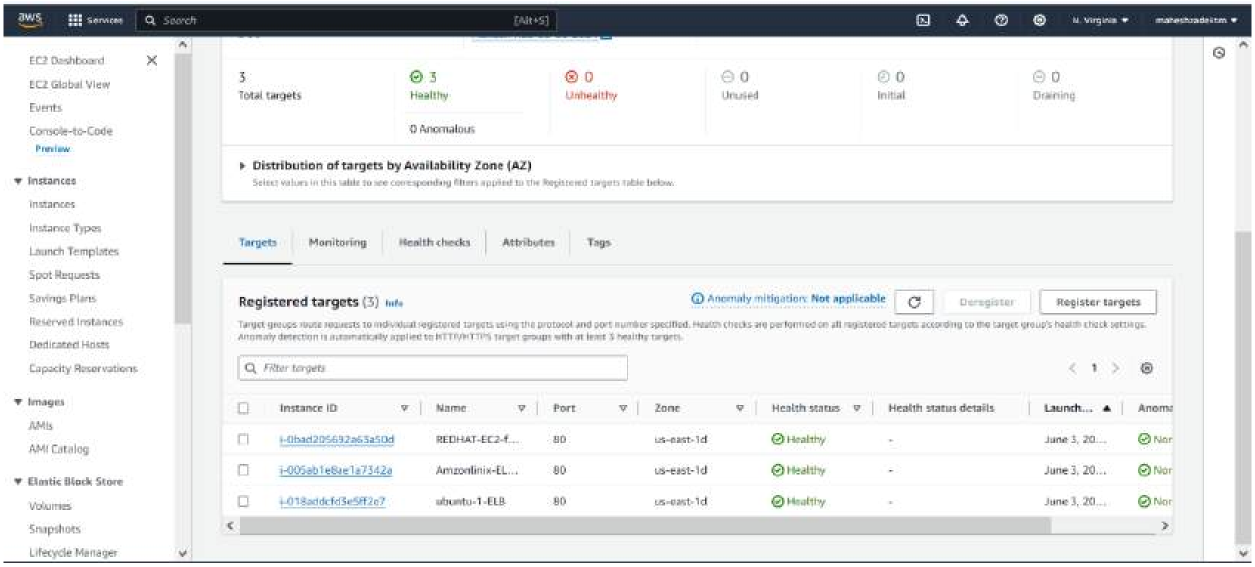
This is ubuntu server fo ELB and Route 53 Assignment



mahesh-clb-637098315.us-east-1.elb.amazonaws.com

Tis is Linux Ec2 for Assignmet of ELB and Route53

Now showing TG as Health for all the targets



3	✓ 3	⊗ 0	⊖ 0	⊖ 0	⊖ 0
Total targets	Healthy	Unhealthy	Unused	Initial	Draining
	0 Anomalous				

▸ Distribution of targets by Availability Zone (AZ)

Select values in this table to see corresponding filters applied to the Registered targets table below.

Registered targets (3)

Anomaly mitigation: Not applicable

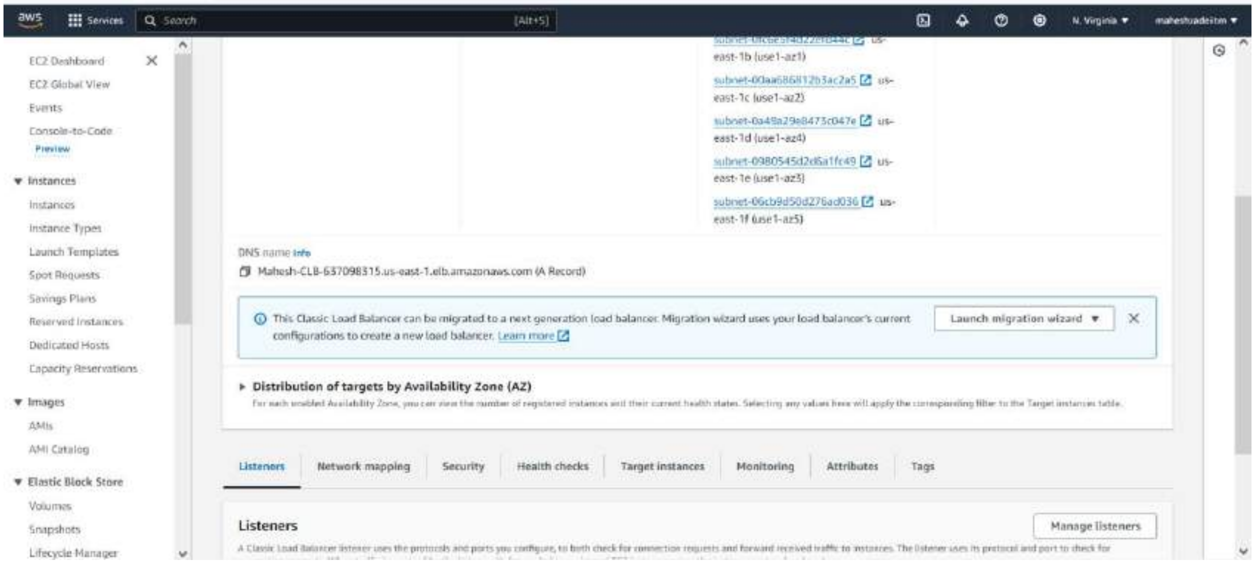
Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

□	Instance ID	Name	Port	Zone	Health status	Health status details	Launch... ▲	Anoma
□	i-0bad205692a63a50d	REDHAT-EC2-f...	80	us-east-1d	✓ Healthy	-	June 3, 20...	✓ Nor
□	i-005ab1e8ae1a7342a	Amzonlinix-EL...	80	us-east-1d	✓ Healthy	-	June 3, 20...	✓ Nor
□	i-018addcfd3e5ff2e7	ubuntu-1-ELB	80	us-east-1d	✓ Healthy	-	June 3, 20...	✓ Nor

Migration of classic Load Balancer

We can do now

Go to Classic Load Balancer and go down



subnet-0fcbe5f4d22efd44c us-east-1b (use1-az1)
subnet-00aa686812b3ac2a5 us-east-1c (use1-az2)
subnet-0a49a29e8473c047e us-east-1d (use1-az4)
subnet-0980545d2d6a1fc49 us-east-1e (use1-az3)
subnet-06cb9d50d276ad036 us-east-1f (use1-az5)

DNS name Info
Mahesh-CLB-637098315.us-east-1.elb.amazonaws.com (A Record)

This Classic Load Balancer can be migrated to a next generation load balancer. Migration wizard uses your load balancer's current configurations to create a new load balancer. Learn more

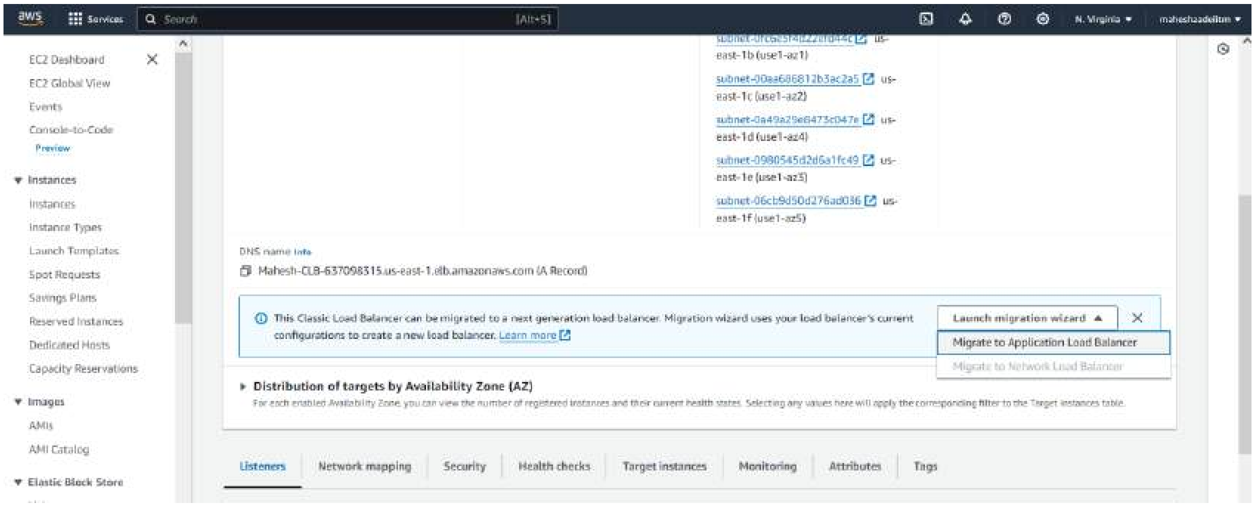
▸ Distribution of targets by Availability Zone (AZ)

For each enabled Availability Zone, you can view the number of registered instances and their current health states. Selecting any values here will apply the corresponding filter to the Target instances table.

Listeners

A Classic Load Balancer listener uses the protocols and ports you configure, to both check for connection requests and forward received traffic to instances. The listener uses its protocol and port to check for

Click on Migrate to application Load Balancer



subnet-0fcbe5f4d22efd44c us-east-1b (use1-az1)
subnet-00aa686812b3ac2a5 us-east-1c (use1-az2)
subnet-0a49a29e8473c047e us-east-1d (use1-az4)
subnet-0980545d2d6a1fc49 us-east-1e (use1-az3)
subnet-06cb9d50d276ad036 us-east-1f (use1-az5)

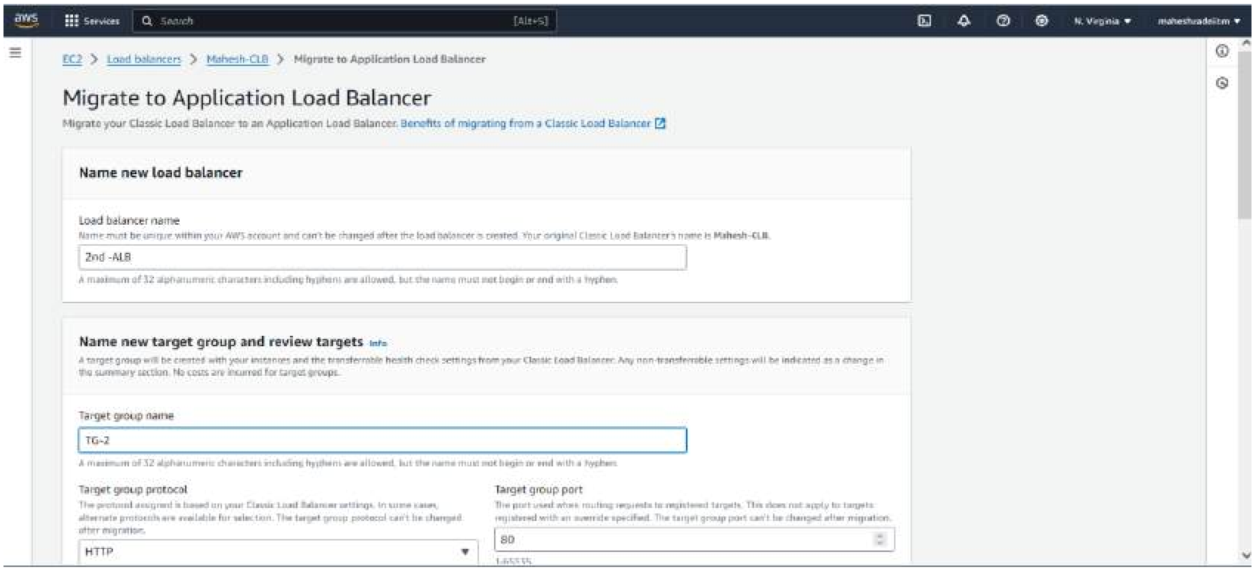
DNS name Info
Mahesh-CLB-637098315.us-east-1.elb.amazonaws.com (A Record)

This Classic Load Balancer can be migrated to a next generation load balancer. Migration wizard uses your load balancer's current configurations to create a new load balancer. Learn more

▸ Distribution of targets by Availability Zone (AZ)

For each enabled Availability Zone, you can view the number of registered instances and their current health states. Selecting any values here will apply the corresponding filter to the Target instances table.

Give the name



EC2 > Load balancers > Mahesh-CLB > Migrate to Application Load Balancer

Migrate to Application Load Balancer

Migrate your Classic Load Balancer to an Application Load Balancer. Benefits of migrating from a Classic Load Balancer

Name new load balancer

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created. Your original Classic Load Balancer's name is **Mahesh-CLB**.

2nd -ALB

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

Name new target group and review targets Info

A target group will be created with your instances and the transferrable health check settings from your Classic Load Balancer. Any non-transferrable settings will be indicated as a change in the summary section. No costs are incurred for target groups.

Target group name

TG-2

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

Target group protocol

The protocol assigned is based on your Classic Load Balancer settings. In some cases, alternate protocols are available for selection. The target group protocol can't be changed after migration.

HTTP

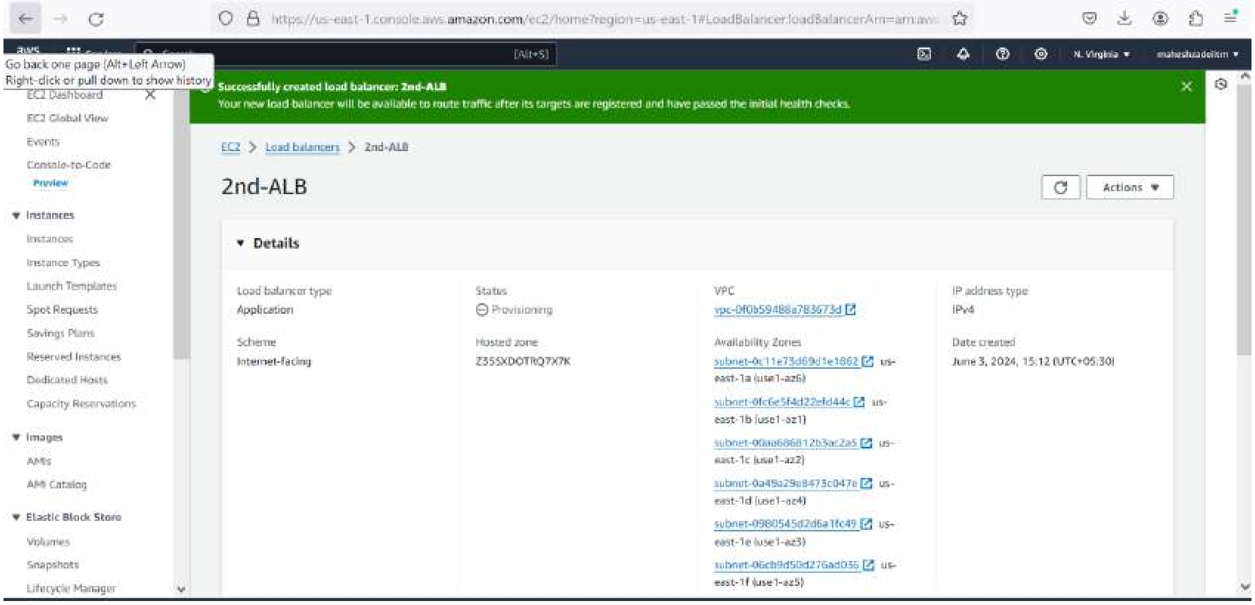
Target group port

The port used when routing requests to registered targets. This does not apply to targets registered with an override specified. The target group port can't be changed after migration.

80

1-65535

Click on application Load Balancer



Successfully created load balancer: 2nd-ALB

Your new load balancer will be available to route traffic after its targets are registered and have passed the initial health checks.

EC2 > Load balancers > 2nd-ALB

2nd-ALB

Details

Load balancer type: Application

Status: Provisioning

VPC: vpc-0f0b59488a783673d

IP address type: IPv4

Scheme: Internet-facing

Hosted zone: Z35SXDOTRQ7X7K

Availability Zones:
subnet-0c11e73d69d1e1862 us-east-1a (use1-az6)
subnet-0fc6e5f4d22efd44c us-east-1b (use1-az1)
subnet-00aa686812b3ac2a5 us-east-1c (use1-az2)
subnet-0a49a29e8473c047e us-east-1d (use1-az4)
subnet-0980545d2d6a1fc49 us-east-1e (use1-az3)
subnet-06cb9d50d276ad036 us-east-1f (use1-az5)

Date created: June 3, 2024, 15:12 (UTC+05:30)

Successfully created load balancer: **2nd-ALB**

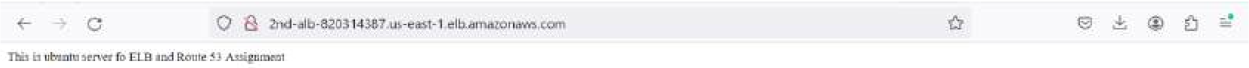
Its showing Classic Load to Application Load Balancer

How Application Load Balancers work

1. Clients make requests to your application.
2. The listeners in your load balancer receive requests matching the protocol and port that you configure.
3. The receiving listener evaluates the incoming request against the rules you specify, and if applicable, routes the request to the appropriate target group. You can use an HTTPS listener to offload the work of TLS encryption and decryption to your load balancer.
4. Healthy targets in one or more target groups receive traffic based on the load balancing algorithm, and the routing rules you specify in the listener.

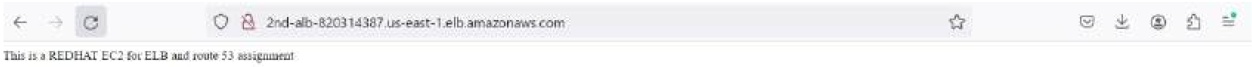
2nd Load balcer DNS check

DNS URL : 2nd-ALB-820314387.us-east-1.elb.amazonaws.com



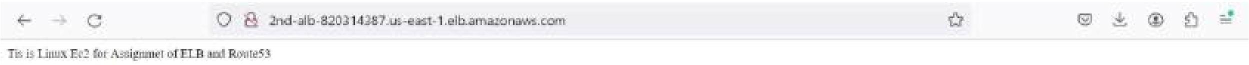
2nd-alb-820314387.us-east-1.elb.amazonaws.com

This is ubuntu server fo ELB and Route 53 Assignment



2nd-alb-820314387.us-east-1.elb.amazonaws.com

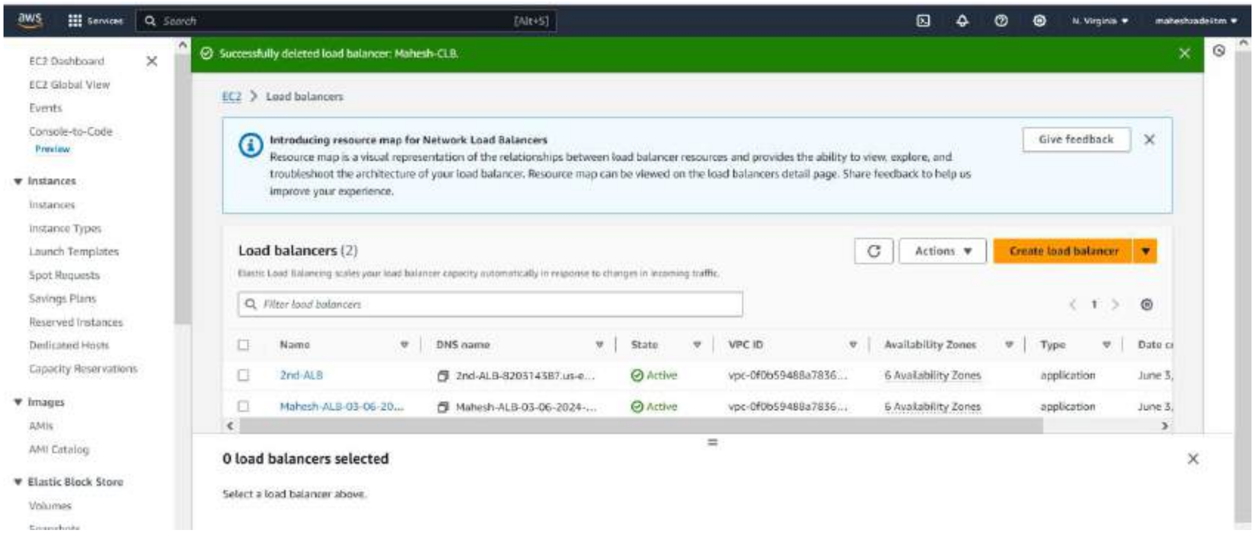
This is a REDHAT EC2 for ELB and route 53 assignment



2nd-alb-820314387.us-east-1.elb.amazonaws.com

Tis is Linux Ec2 for Assignmet of ELB and Route53

Delete the Classic Load Balancer as we Migrate it to application Load Balancer as its Assignment topic taught by Aniket Sir on dated 02-06-2024



Successfully deleted load balancer: Mahesh-CLB.

EC2 > Load balancers

Introducing resource map for Network Load Balancers
Resource map is a visual representation of the relationships between load balancer resources and provides the ability to view, explore, and troubleshoot the architecture of your load balancer. Resource map can be viewed on the load balancers detail page. Share feedback to help us improve your experience.

Load balancers (2)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

□	Name	DNS name	State	VPC ID	Availability Zones	Type	Date c
□	2nd-ALB	2nd-ALB-820314387.us-e...	Active	vpc-0f0b59488a7836...	6 Availability Zones	application	June 3,
□	Mahesh-ALB-03-06-20...	Mahesh-ALB-03-06-2024-...	Active	vpc-0f0b59488a7836...	6 Availability Zones	application	June 3,

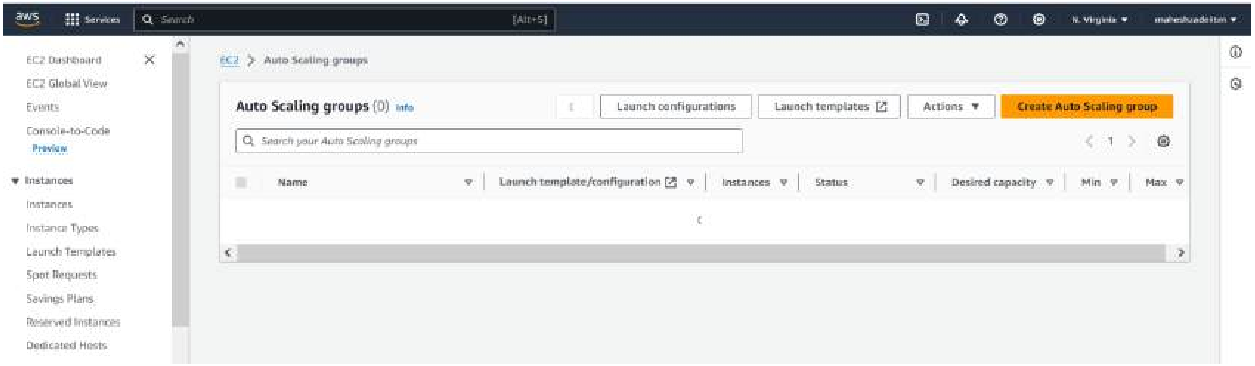
0 load balancers selected

Select a load balancer above.

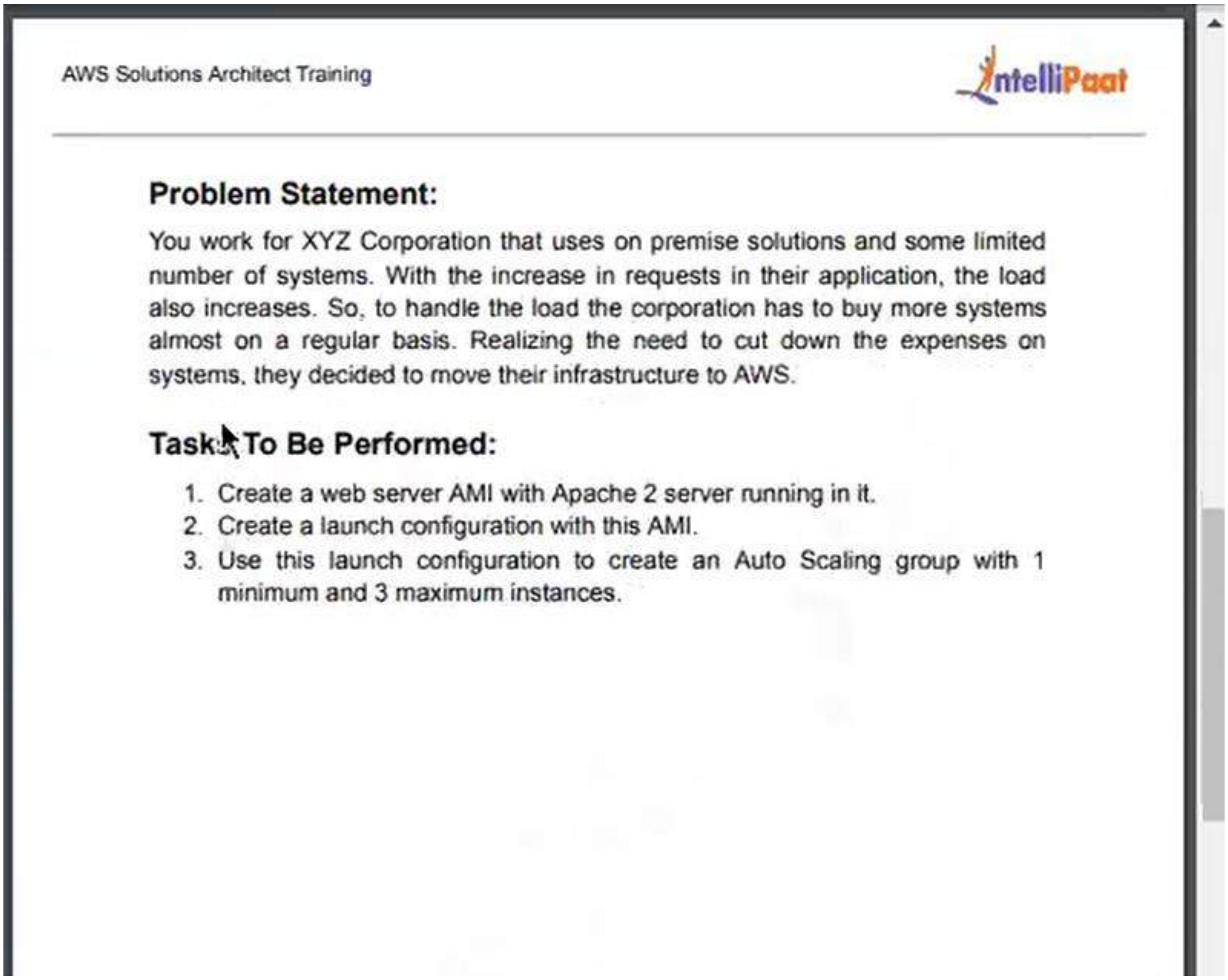
Now Go to Auto scaling in AWS Console

IntelliPaat

Module 4: Auto-Scaling Assignment



Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max



AWS Solutions Architect Training

Problem Statement:

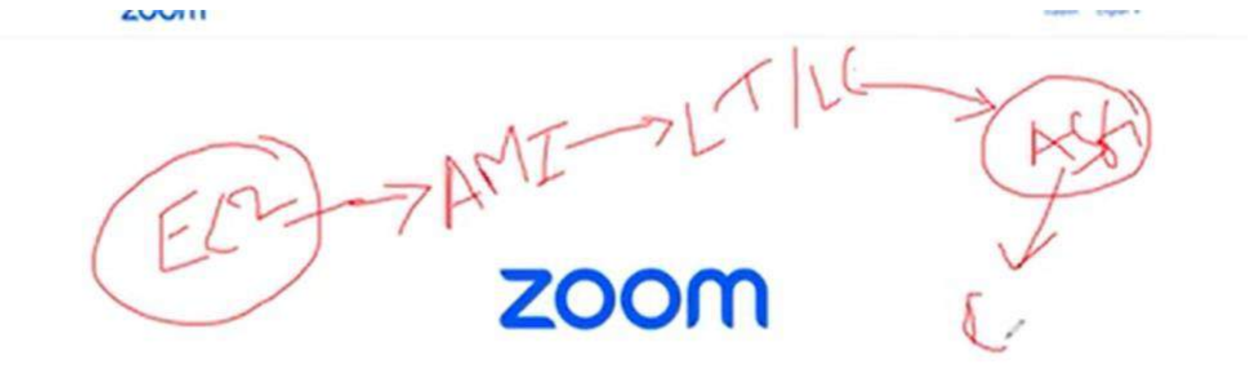
You work for XYZ Corporation that uses on premise solutions and some limited number of systems. With the increase in requests in their application, the load also increases. So, to handle the load the corporation has to buy more systems almost on a regular basis. Realizing the need to cut down the expenses on systems, they decided to move their infrastructure to AWS.

Tasks To Be Performed:

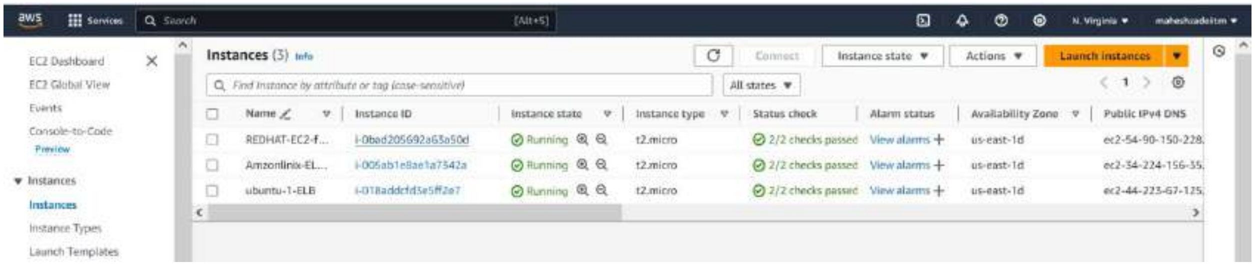
1. Create a web server AMI with Apache 2 server running in it.
2. Create a launch configuration with this AMI.
3. Use this launch configuration to create an Auto Scaling group with 1 minimum and 3 maximum instances.

Steps to Follow

EC2-AMI-Launch Template -Configure Autoscaling Group-max 3 instances



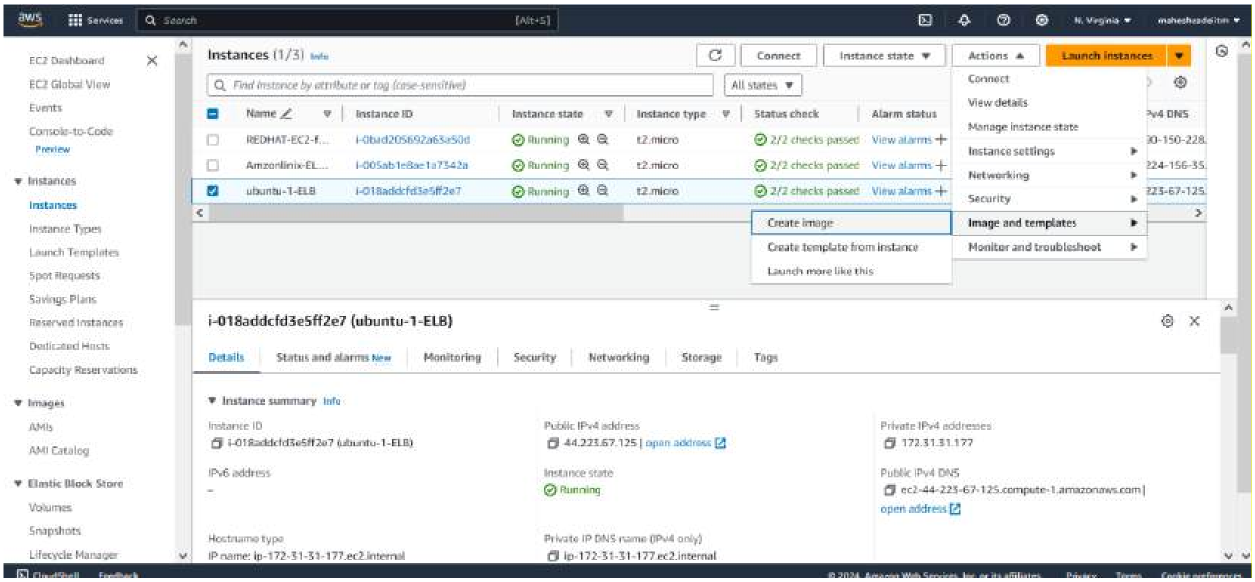
All Ec2 Instances created as web servers



Instances (3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
REDHAT-EC2-f...	i-0bad205692a63a50d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-54-90-150-228.
Amzonlinix-EL...	i-005ab1e8ae1a7542a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-34-224-156-35.
ubuntu-1-ELB	i-018addcfd3e5ff2e7	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-44-223-67-125.

Now Create the AMI using Ec2 instances



Instances (1/3) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	...v4 DNS
□	REDHAT-EC2-f...	i-0bad205692a63a50d	Running	t2.micro	2/2 checks passed	View alarms +	...0-150-228.
□	Amzonlinix-EL...	i-005ab1e8ae1a7542a	Running	t2.micro	2/2 checks passed	View alarms +	...224-156-35.
☑	ubuntu-1-ELB	i-018addcfd3e5ff2e7	Running	t2.micro	2/2 checks passed	View alarms +	...225-67-125.

i-018addcfd3e5ff2e7 (ubuntu-1-ELB)

Instance summary Info

Instance ID: i-018addcfd3e5ff2e7 (ubuntu-1-ELB)

Public IPv4 address: 44.223.67.125 | open address

Private IPv4 addresses: 172.31.31.177

IPv6 address: –

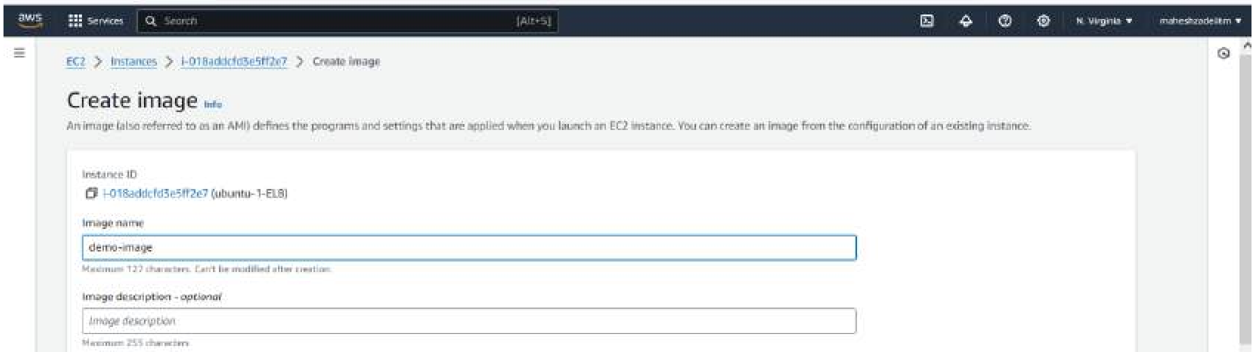
Instance state: Running

Public IPv4 DNS: ec2-44-223-67-125.compute-1.amazonaws.com | open address

Hostname type: IP name: ip-172-31-31-177.ec2.internal

Private IP DNS name (IPv4 only): ip-172-31-31-177.ec2.internal

Give the name demo-image



Create image Info

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID: i-018addcfd3e5ff2e7 (ubuntu-1-ELB)

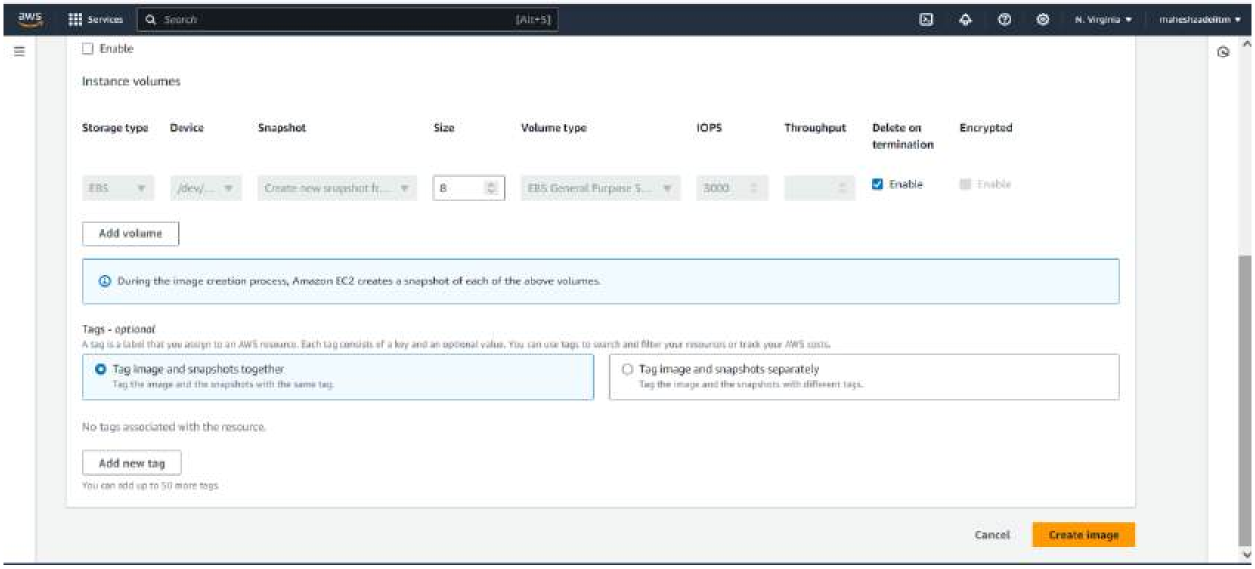
Image name: demo-image

Maximum 127 characters. Can't be modified after creation.

Image description - *optional*: Image description

Maximum 255 characters.

Create Image



□ Enable

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/...	Create new snapshot fr...	8	EBS General Purpose S...	3000		☑ Enable	□ Enable

Add volume

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - *optional*
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

● Tag image and snapshots together
Tag the image and the snapshots with the same tag.

○ Tag image and snapshots separately
Tag the image and the snapshots with different tags.

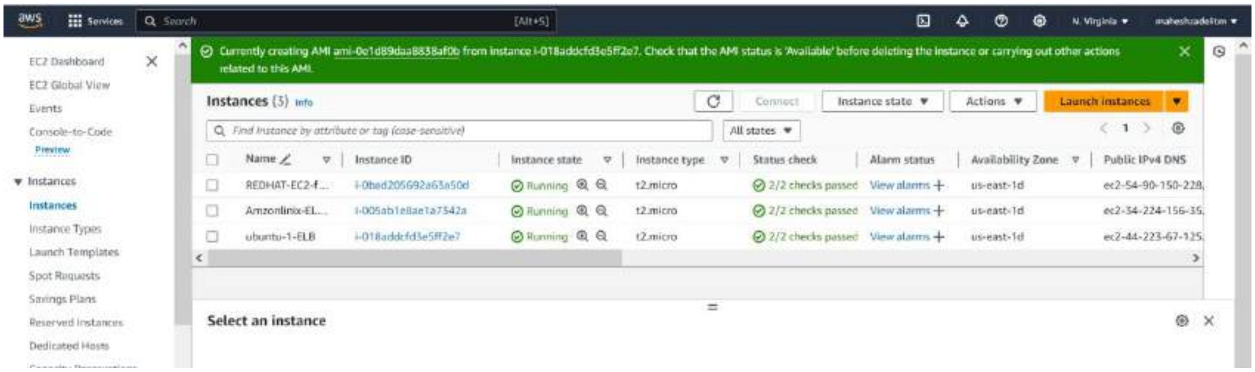
No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel
Create image

Image Created



Currently creating AMI ami-0e1d89daa8838af0b from instance i-018addcfd3e5ff2e7. Check that the AMI status is 'Available' before deleting the instance or carrying out other actions related to this AMI.

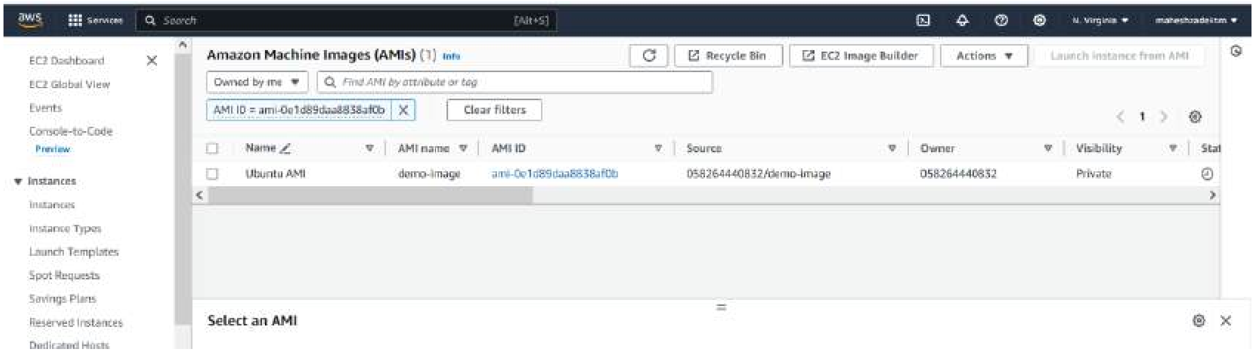
Instances (3) Info

Find instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
REDHAT-EC2-f...	i-0bad205692a63a50d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-54-90-150-228.
Amzonlinix-EL...	i-005ab1e8ae1a7542a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-34-224-156-35.
ubuntu-1-ELB	i-018addcfd3e5ff2e7	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-44-223-67-125.

Select an instance

AMI Created



Amazon Machine Images (AMIs) (1) Info

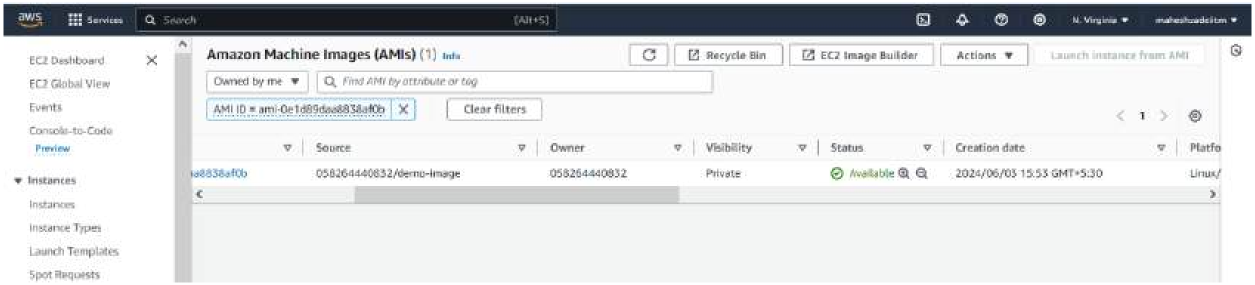
Find AMI by attribute or tag

AMI ID = ami-0e1d89daa8838af0b

Name	AMI name	AMI ID	Source	Owner	Visibility	Stat
Ubuntu AMI	demo-image	ami-0e1d89daa8838af0b	058264440832/demo-image	058264440832	Private	

Select an AMI

Wait till status is available



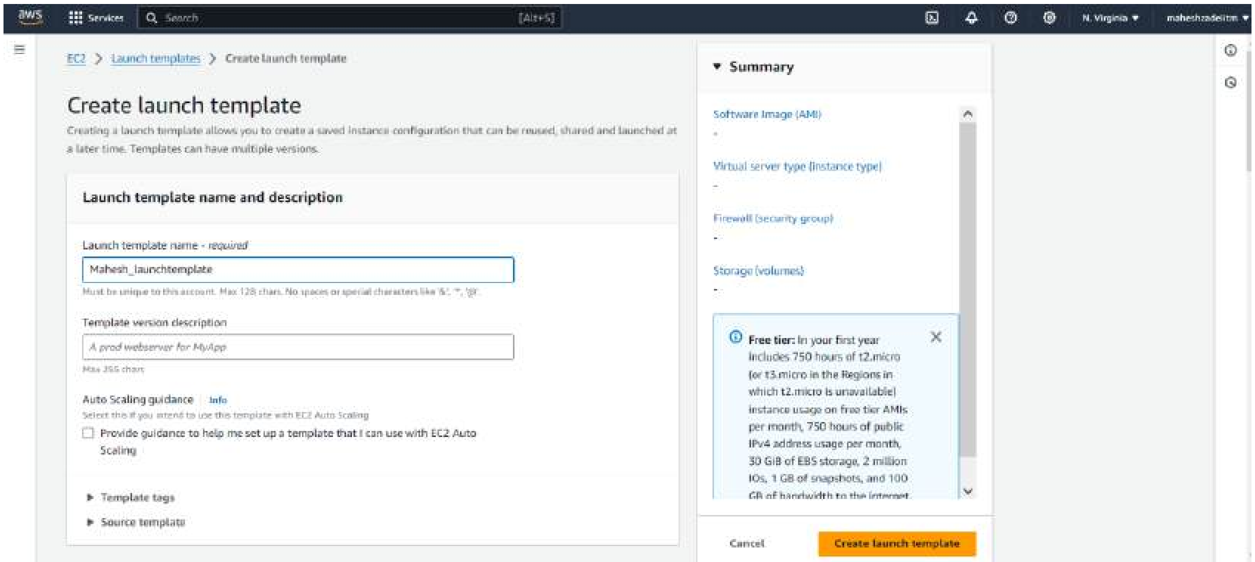
Amazon Machine Images (AMIs) (1) Info

Owned by me ▼ | Find AMI by attribute or tag

AMI ID = ami-0e1d89daa8838af0b ✕

	Source	Owner	Visibility	Status	Creation date	Platfo
a8838af0b	058264440832/demo-image	058264440832	Private	Available	2024/06/03 15:53 GMT+5:30	Linux/

Now go to AWS Console to Launch Template and give the Name



Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

Mahesh_launchtemplate

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

A prod webserver for MyApp

Max 255 chars

Auto Scaling guidance Info

Select this if you intend to use this template with EC2 Auto Scaling

□ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

▼ Summary

Software Image (AMI)
-

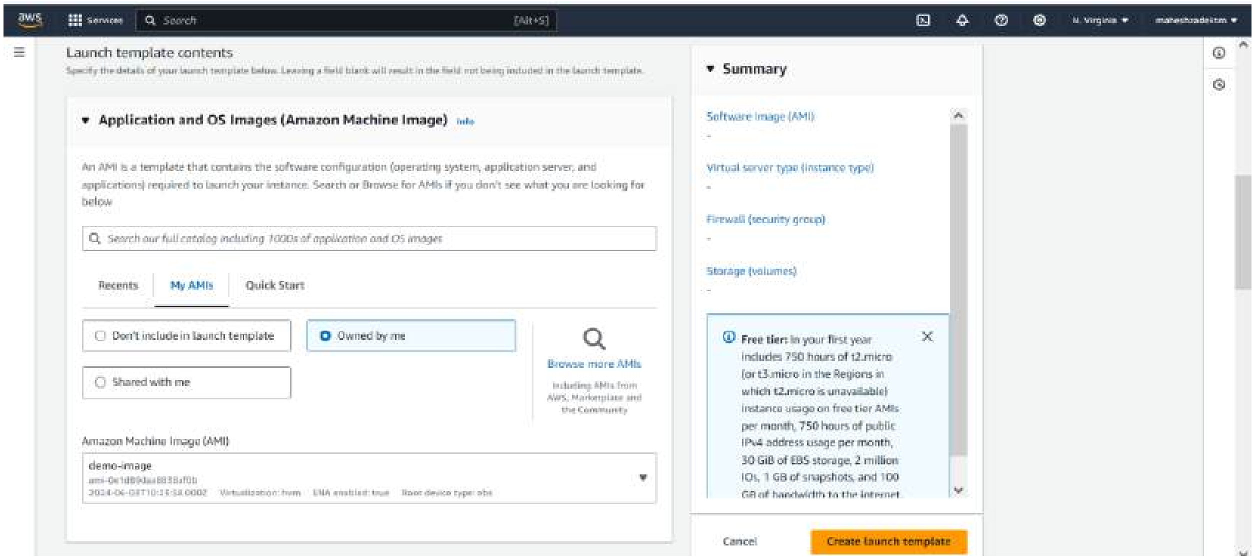
Virtual server type (instance type)
-

Firewall (security group)
-

Storage (volumes)
-

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet

Go to My AMI option and select owned by me image automatically selected



Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

▼ Application and OS Images (Amazon Machine Image) Info

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

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

○ Don't include in launch template
● Owned by me
○ Shared with me

Amazon Machine Image (AMI)

demo-image
ami-0e1d89daa8838af0b
2024-06-03T10:23:58.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

▼ Summary

Software Image (AMI)
-

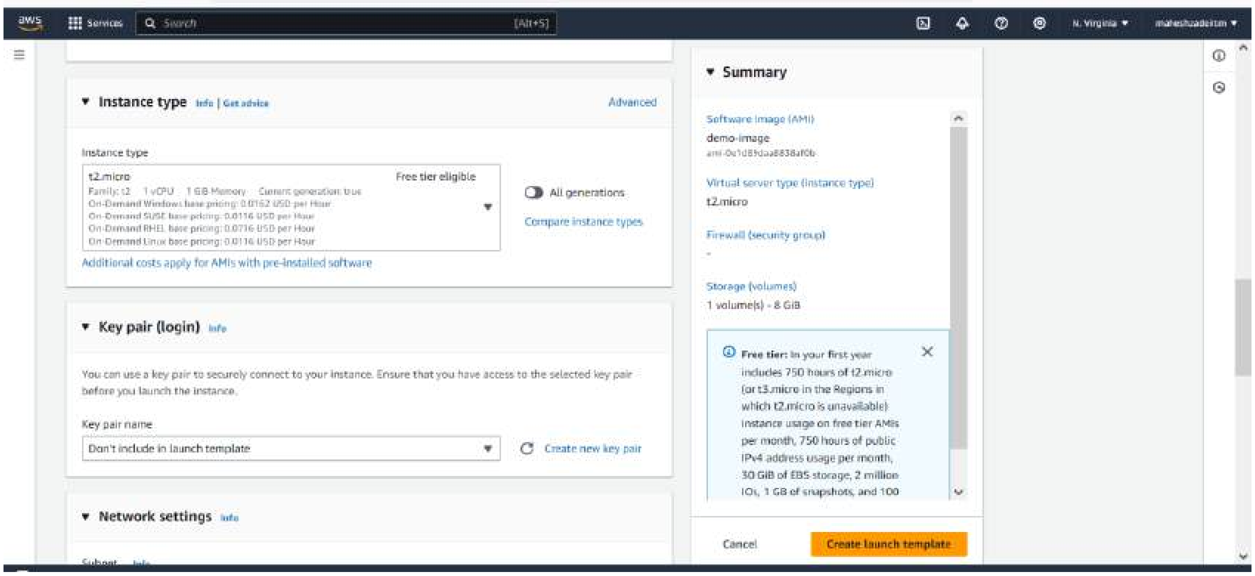
Virtual server type (instance type)
-

Firewall (security group)
-

Storage (volumes)
-

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet

Instance Type selectt2.micro



▼ Instance type Info | Get advice Advanced

Instance type

t2.micro Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.0716 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

Don't include in launch template ▼

Create new key pair

▼ Network settings Info

Subnet Info

▼ Summary

Software image (AMI)
demo-image
ami-0e1d89daa8838af0b

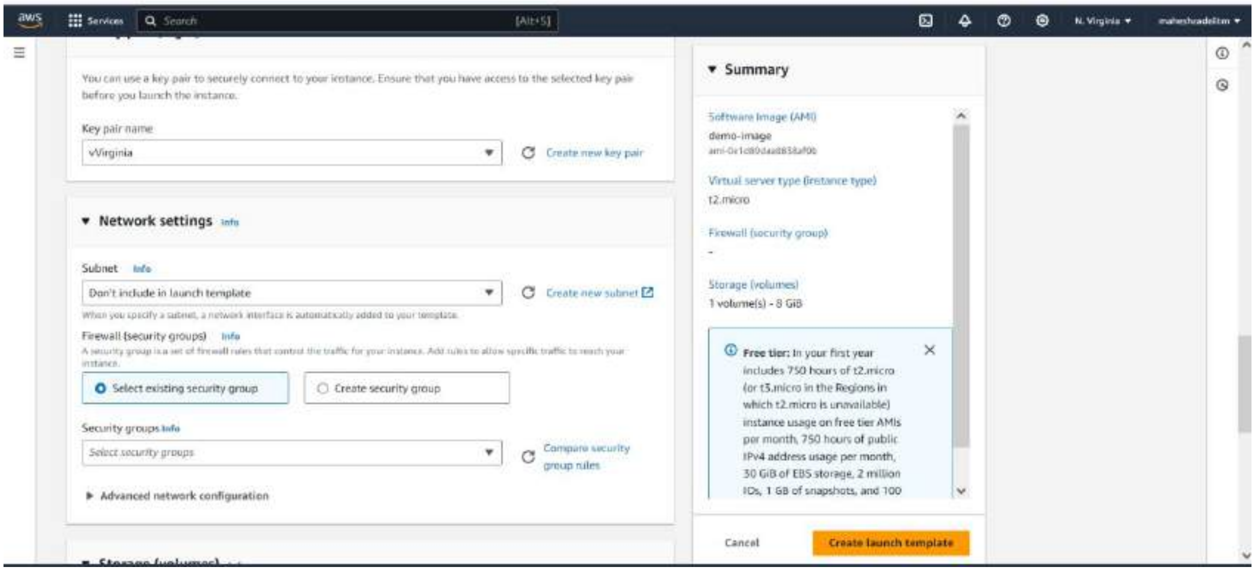
Virtual server type (instance type)
t2.micro

Firewall (security group)
-

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100

Specify keypair on N vergunia



You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

vVirginia ▼

Create new key pair

▼ Network settings Info

Subnet Info

Don't include in launch template ▼

Create new subnet

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) Info

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

● Select existing security group ○ Create security group

Security groups Info

Select security groups ▼

Compare security group rules

▶ Advanced network configuration

▼ Storage (volumes)

▼ Summary

Software image (AMI)
demo-image
ami-0e1d89daa8838af0b

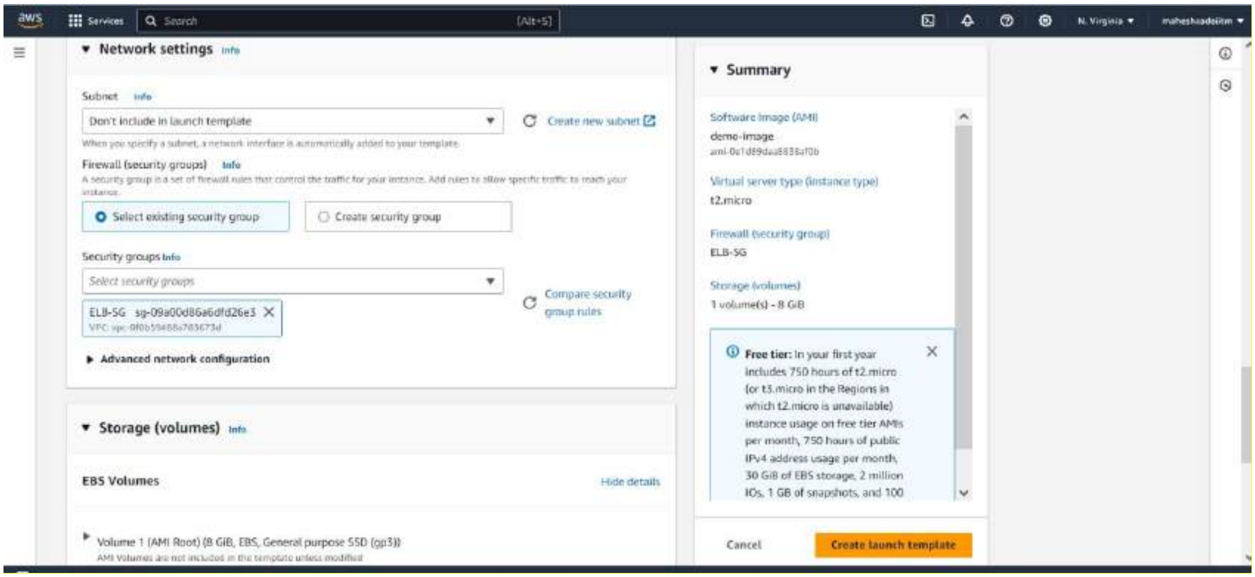
Virtual server type (instance type)
t2.micro

Firewall (security group)
-

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100

Select SG which we created



▼ Network settings Info

Subnet Info

Don't include in launch template ▾

Create new subnet

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) Info

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

● Select existing security group ○ Create security group

Security groups Info

Select security groups ▾

ELB-SG sg-09a00d86a6dfd26e3 ✕
VPC: vpc-0f0b59488a783673d

Compare security group rules

▸ Advanced network configuration

▼ Storage (volumes) Info

EBS Volumes Hide details

▸ Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp3))
AMI Volumes are not included in the template unless modified

▼ Summary

Software Image (AMI)
demo-image
ami-0e1d89daa8838af0b

Virtual server type (instance type)
t2.micro

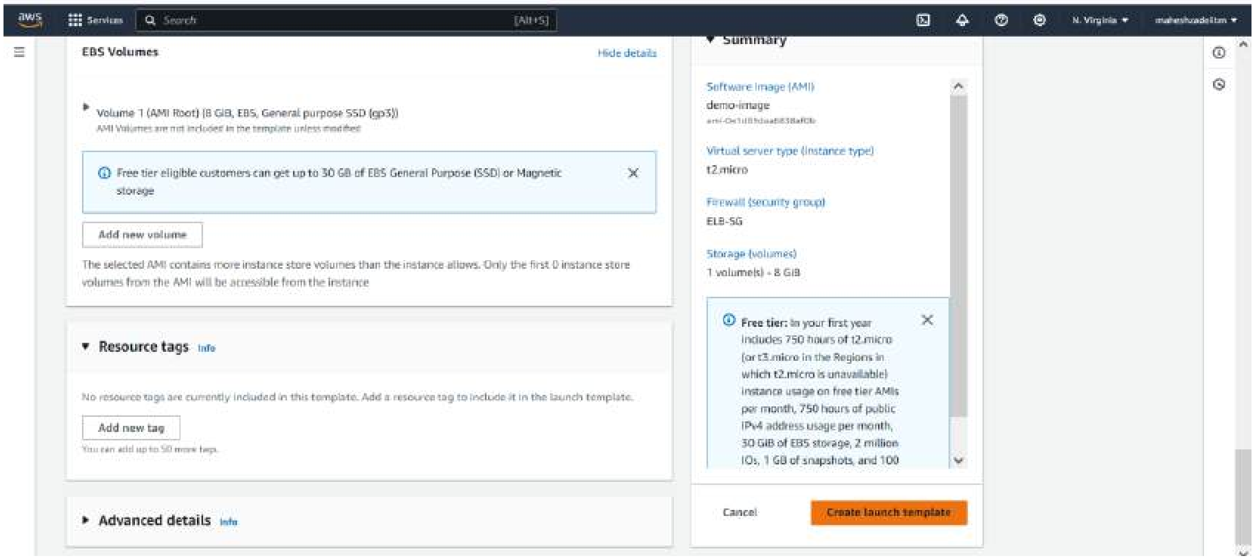
Firewall (security group)
ELB-SG

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 ✕

Cancel Create launch template

Click on launch Template



EBS Volumes Hide details

▸ Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp3))
AMI Volumes are not included in the template unless modified

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage ✕

Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

▼ Resource tags Info

No resource tags are currently included in this template. Add a resource tag to include it in the launch template.

Add new tag

You can add up to 50 more tags.

▸ Advanced details Info

▼ Summary

Software Image (AMI)
demo-image
ami-0e1d89daa8838af0b

Virtual server type (instance type)
t2.micro

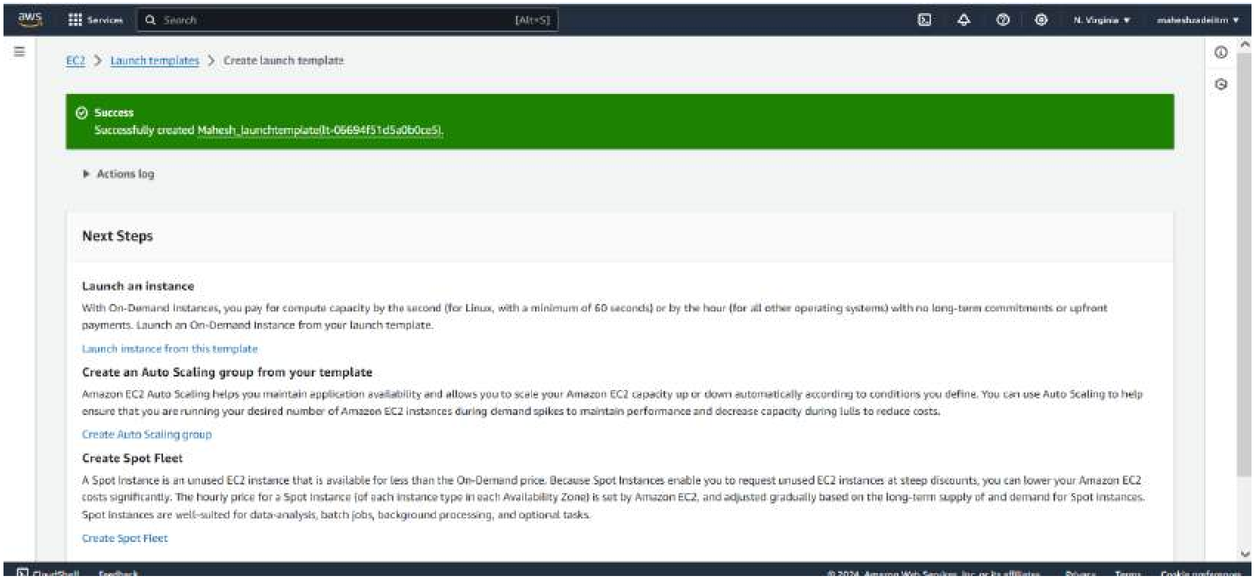
Firewall (security group)
ELB-SG

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 ✕

Cancel Create launch template

Template Launched



EC2 > Launch templates > Create launch template

Success
Successfully created Mahesh_launchtemplate(lt-06694f51d5a0b0ce5).

▶ Actions log

Next Steps

Launch an instance

With On-Demand Instances, you pay for compute capacity by the second (for Linux, with a minimum of 60 seconds) or by the hour (for all other operating systems) with no long-term commitments or upfront payments. Launch an On-Demand Instance from your launch template.

Launch instance from this template

Create an Auto Scaling group from your template

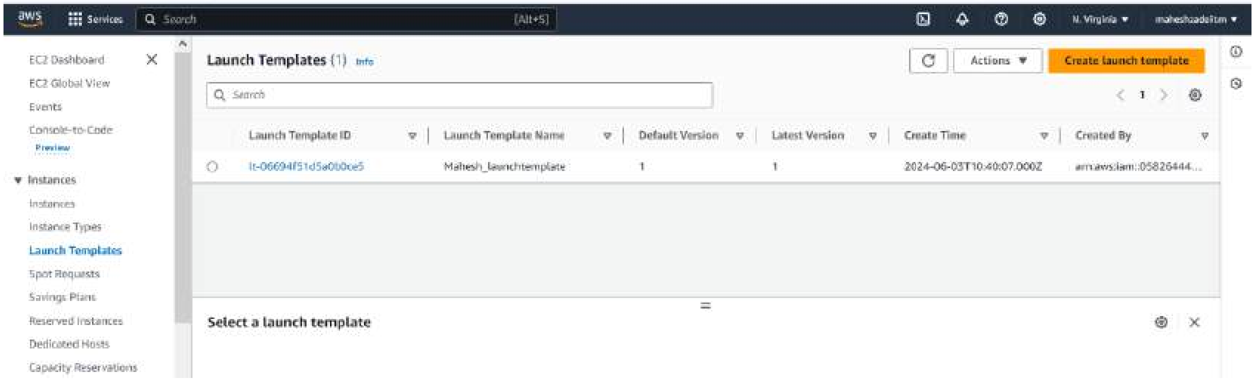
Amazon EC2 Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs.

Create Auto Scaling group

Create Spot Fleet

A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance (of each instance type in each Availability Zone) is set by Amazon EC2, and adjusted gradually based on the long-term supply of and demand for Spot Instances. Spot Instances are well-suited for data-analysis, batch jobs, background processing, and optional tasks.

Create Spot Fleet

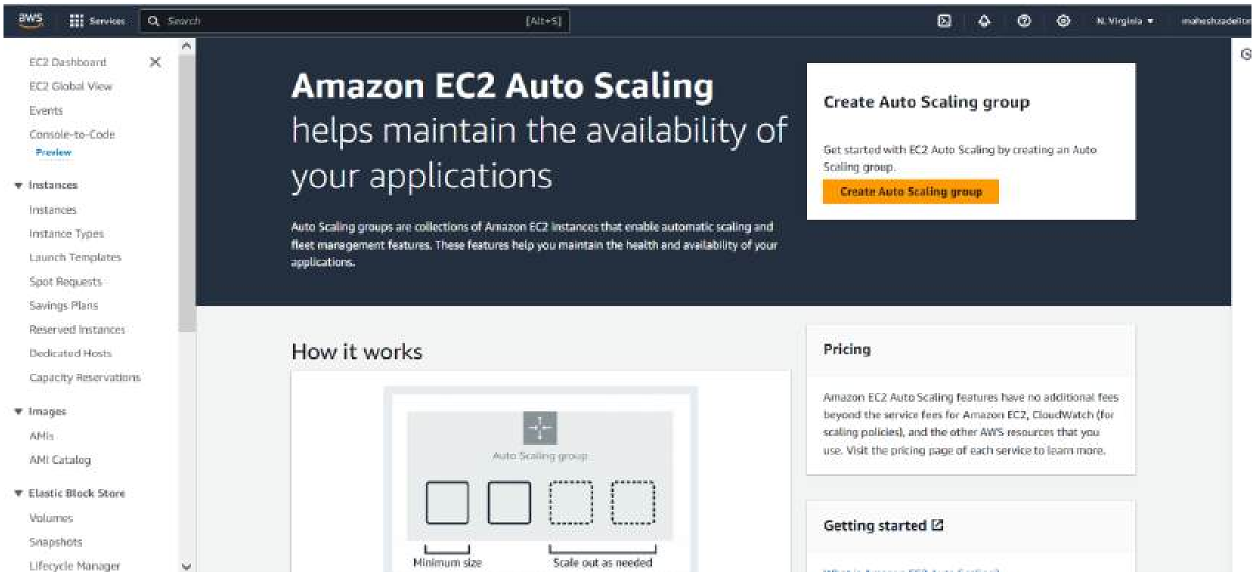


Launch Templates (1) Info

	Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
○	lt-06694f51d5a0b0ce5	Mahesh_launchtemplate	1	1	2024-06-03T10:40:07.000Z	arn:aws:iam::05826444...

Select a launch template

Now Go to AWS Console to create the Auto scaling Group



Amazon EC2 Auto Scaling
helps maintain the availability of your applications

Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. These features help you maintain the health and availability of your applications.

Create Auto Scaling group

Get started with EC2 Auto Scaling by creating an Auto Scaling group.

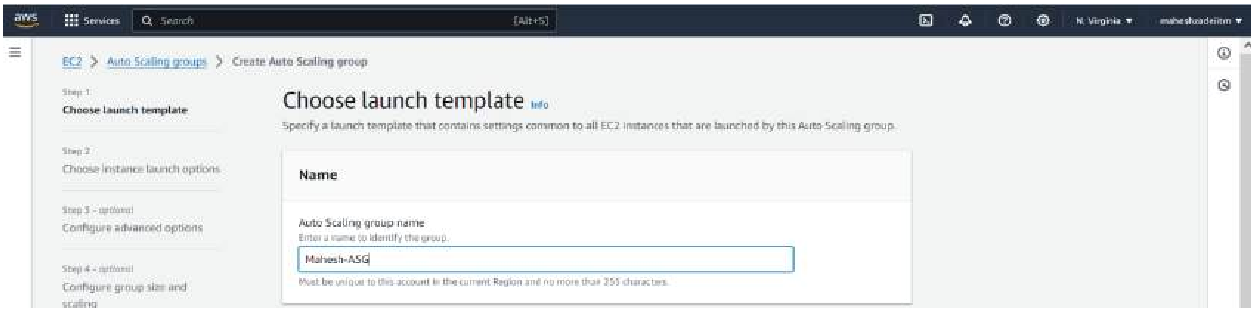
Create Auto Scaling group

How it works

Pricing

Amazon EC2 Auto Scaling features have no additional fees beyond the service fees for Amazon EC2, CloudWatch (for scaling policies), and the other AWS resources that you use. Visit the pricing page of each service to learn more.

Getting started



EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template

Step 2
Choose instance launch options

Step 3 - optional
Configure advanced options

Step 4 - optional
Configure group size and scaling

Choose launch template Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

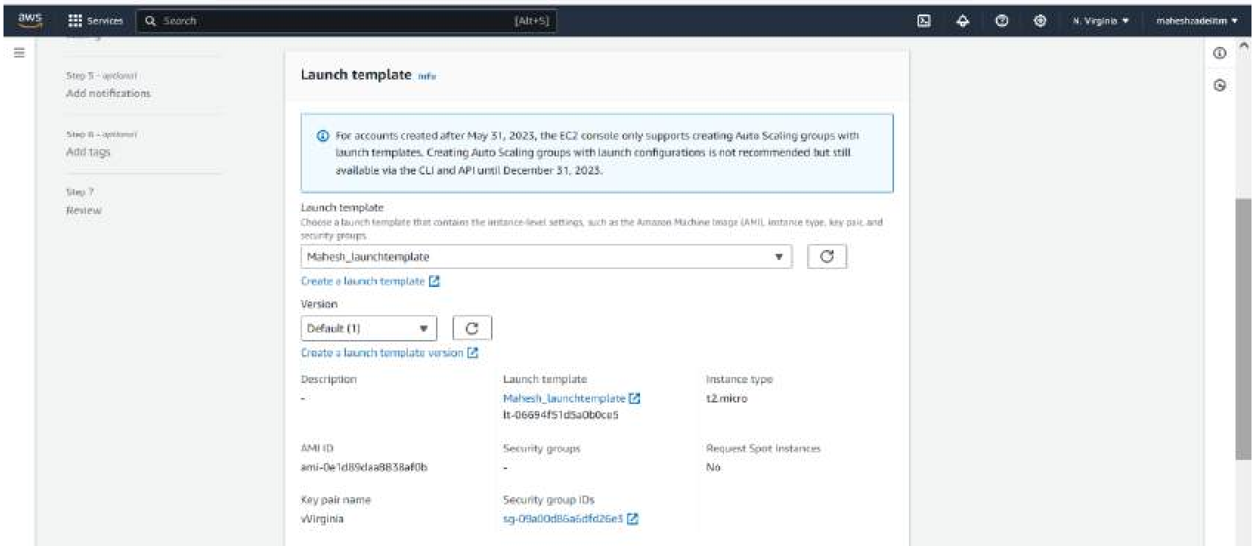
Name

Auto Scaling group name
Enter a name to identify the group.

Mahesh-ASG

Must be unique to this account in the current Region and no more than 255 characters.

Select launch Template



Step 5 - optional
Add notifications

Step 6 - optional
Add tags

Step 7
Review

Launch template Info

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Mahesh_launchtemplate

Create a launch template

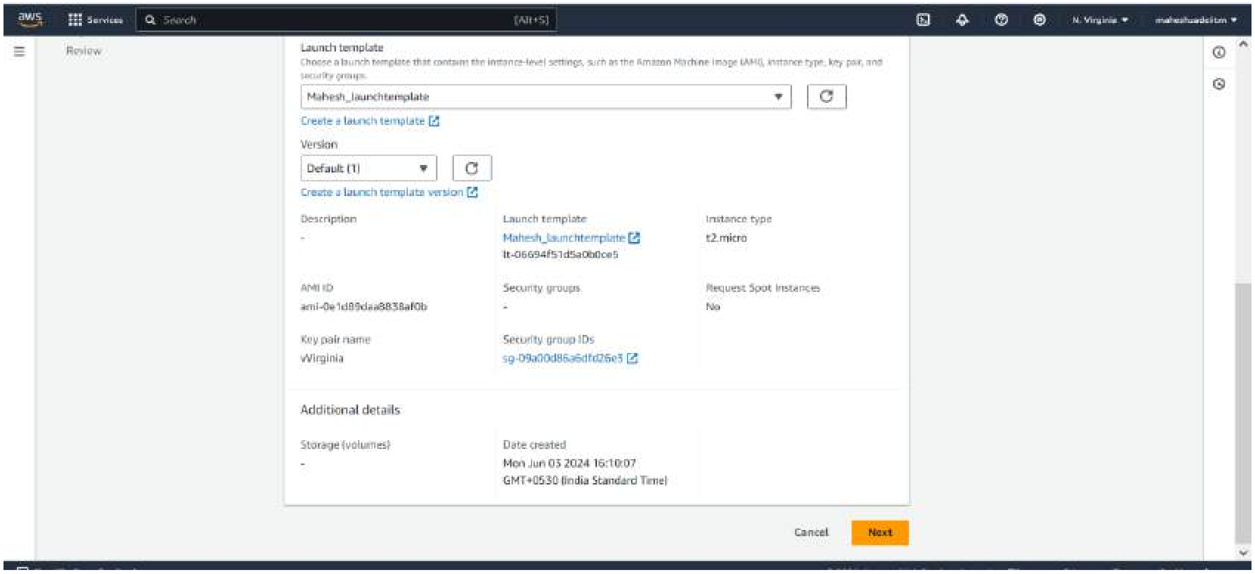
Version

Default (1)

Create a launch template version

Description	Launch template	Instance type
-	Mahesh_launchtemplate lt-06694f51d5a0b0ce5	t2.micro
AMI ID	Security groups	Request Spot Instances
ami-0e1d89daa8838af0b	-	No
Key pair name	Security group IDs	
Wirginia	sg-09a00d86a6dfd26e5	

Next



Review

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Mahesh_launchtemplate

Create a launch template

Version

Default (1)

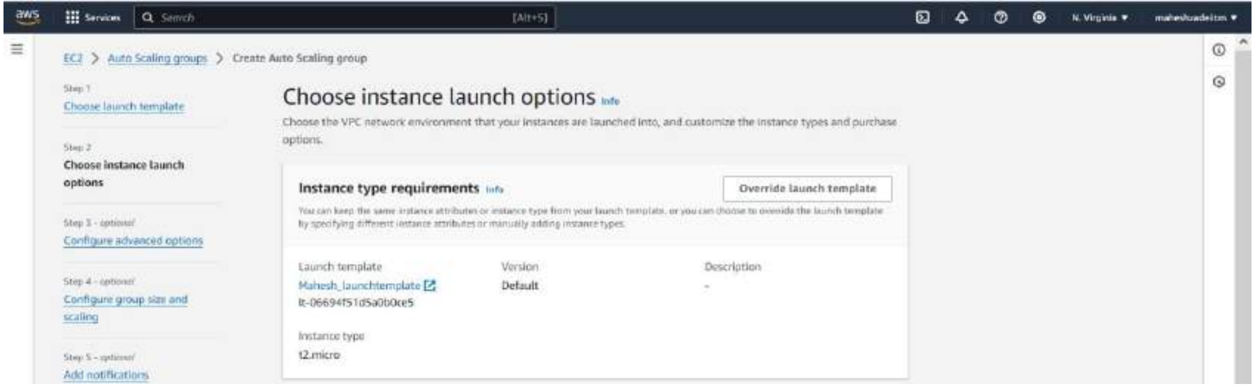
Create a launch template version

Description	Launch template	Instance type
-	Mahesh_launchtemplate lt-06694f51d5a0b0ce5	t2.micro
AMI ID	Security groups	Request Spot Instances
ami-0e1d89daa8838af0b	-	No
Key pair name	Security group IDs	
Wirginia	sg-09a00d86a6dfd26e5	

Additional details

Storage (volumes)	Date created
-	Mon Jun 03 2024 16:10:07 GMT+0530 (India Standard Time)

Instance type t2 micro



EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template

Step 2
Choose instance launch options

Step 3 - optional
Configure advanced options

Step 4 - optional
Configure group size and scaling

Step 5 - optional
Add notifications

Choose instance launch options Info

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Instance type requirements Info

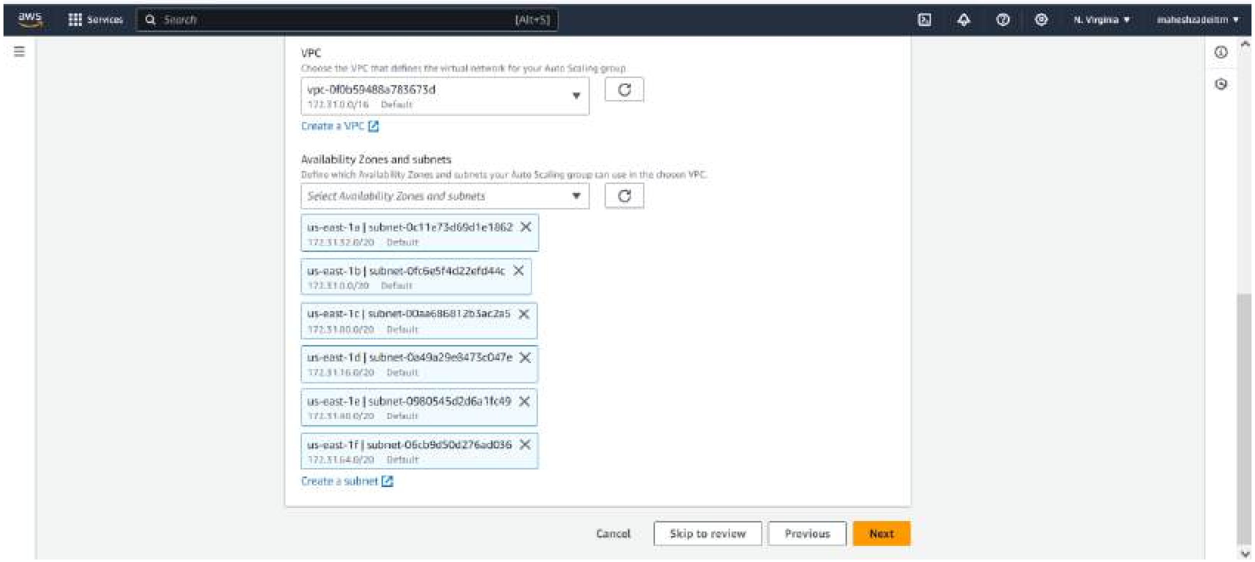
Override launch template

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

Launch template	Version	Description
Mahesh_launchtemplate lt-06694f51d5a0b0ce5	Default	-

Instance type
t2.micro

Select default VPC and all AZ and click on next



VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0f0b59488a783673d
172.31.0.0/16 Default

Create a VPC

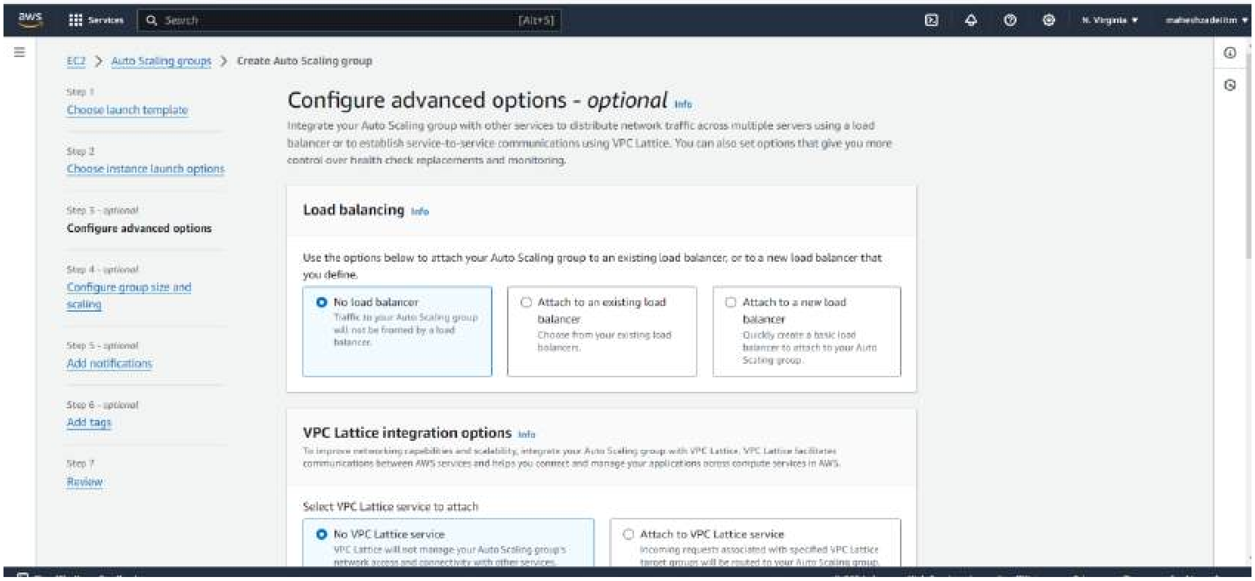
Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

- us-east-1a | subnet-0c11e73d69d1e1862 — 172.31.32.0/20 Default
- us-east-1b | subnet-0fc6e5f4d22efd44c — 172.31.0.0/20 Default
- us-east-1c | subnet-00aa686812b3ac2a5 — 172.31.80.0/20 Default
- us-east-1d | subnet-0a49a29e8473c047e — 172.31.16.0/20 Default
- us-east-1e | subnet-0980545d2d6a1fc49 — 172.31.48.0/20 Default
- us-east-1f | subnet-06cb9d50d276ad036 — 172.31.64.0/20 Default

Create a subnet

Leave all the setting as it is



Configure advanced options - *optional* Info

Integrate your Auto Scaling group with other services to distribute network traffic across multiple servers using a load balancer or to establish service-to-service communications using VPC Lattice. You can also set options that give you more control over health check replacements and monitoring.

Load balancing Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

● No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

○ Attach to an existing load balancer
Choose from your existing load balancers.

○ Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

VPC Lattice integration options Info

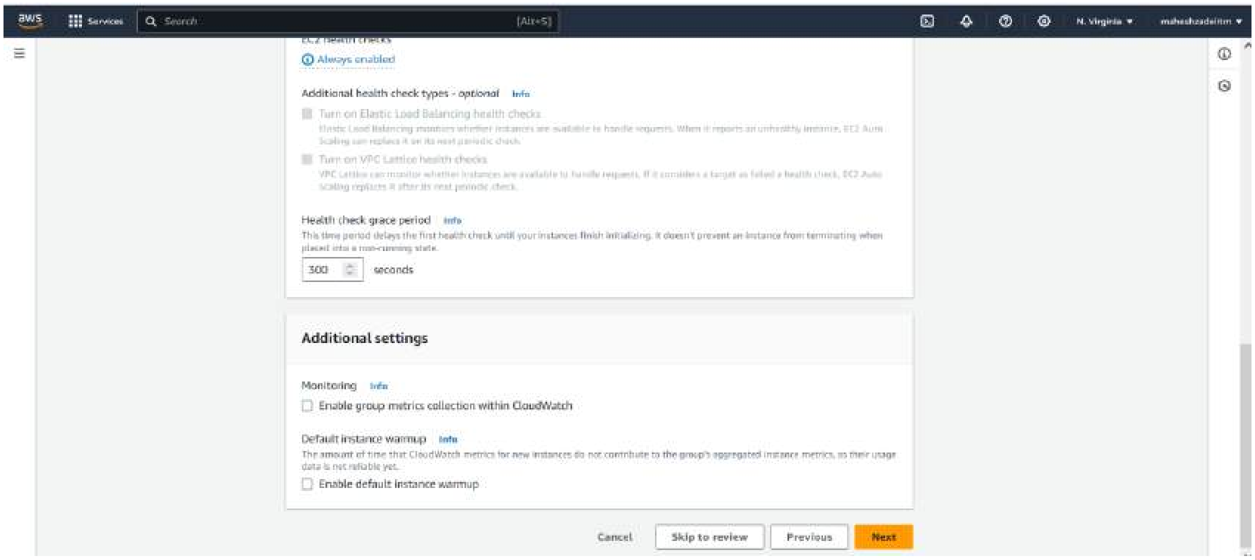
To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

Select VPC Lattice service to attach

● No VPC Lattice service
VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.

○ Attach to VPC Lattice service
Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

Click on next



EC2 health checks
Always enabled

Additional health check types - *optional* Info

□ Turn on Elastic Load Balancing health checks
Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.

□ Turn on VPC Lattice health checks
VPC Lattice can monitor whether instances are available to handle requests. If it considers a target as failed a health check, EC2 Auto Scaling replaces it after its next periodic check.

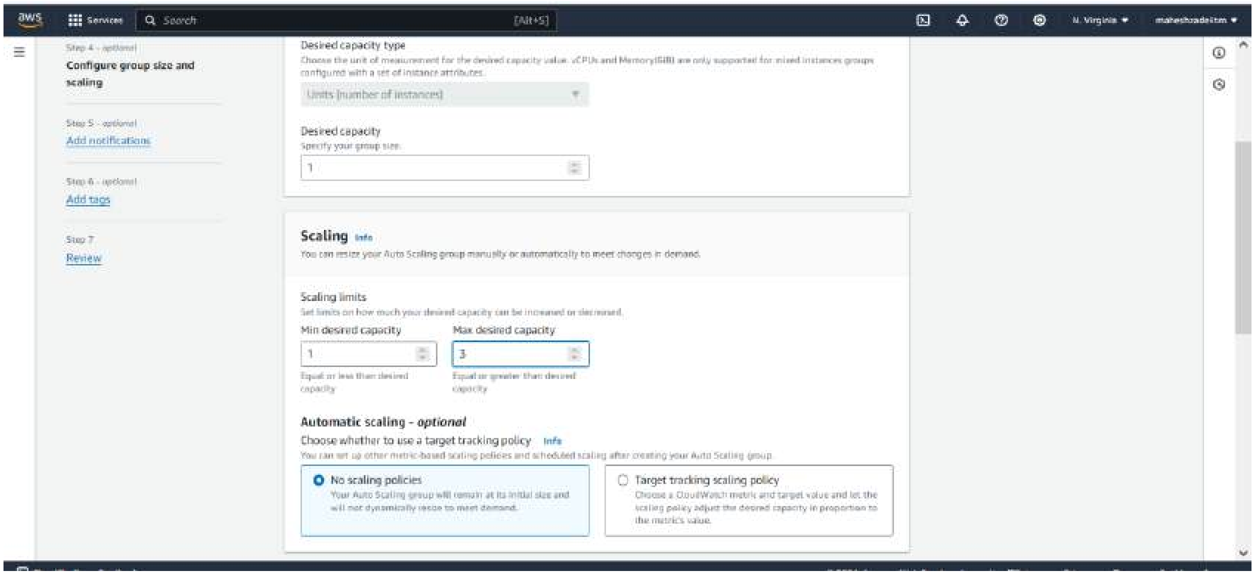
Health check grace period Info
This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.
300 seconds

Additional settings

Monitoring Info
□ Enable group metrics collection within CloudWatch

Default instance warmup Info
The amount of time that CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.
□ Enable default instance warmup

Scaling need to select min 1 and Max 3



Step 4 - optional
Configure group size and scaling

Step 5 - optional
Add notifications

Step 6 - optional
Add tags

Step 7
Review

Desired capacity type
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances)

Desired capacity
Specify your group size.

1

Scaling Info
You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits
Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity: 1
Equal or less than desired capacity

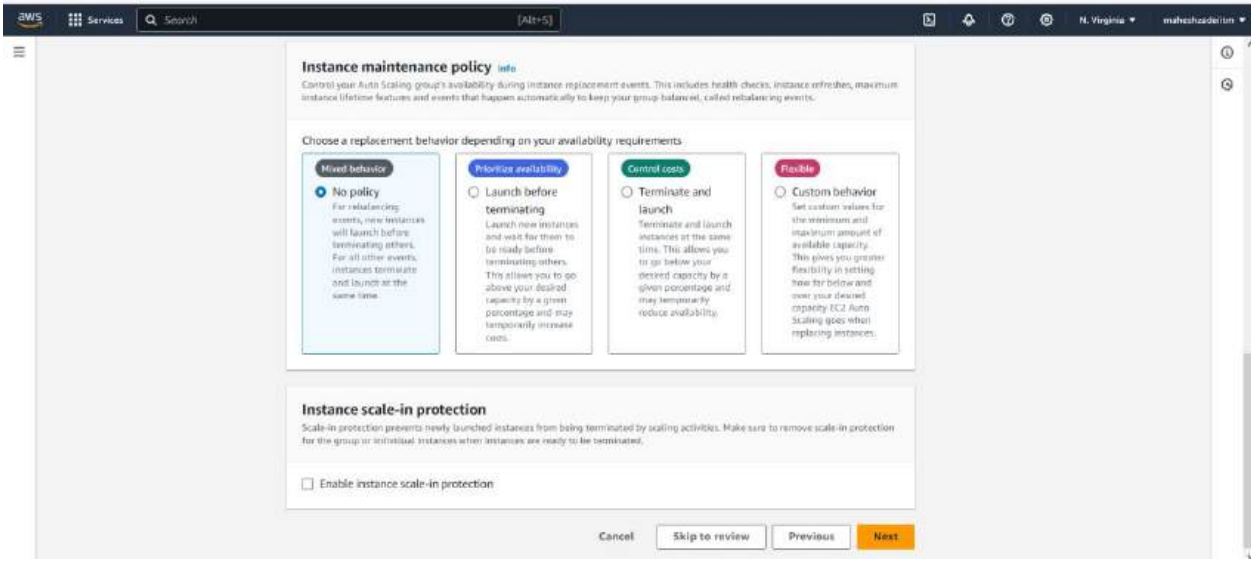
Max desired capacity: 3
Equal or greater than desired capacity

Automatic scaling - *optional*
Choose whether to use a target tracking policy Info
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

● No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

○ Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Next



Instance maintenance policy Info
Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum instance lifetime features and events that happen automatically to keep your group balanced, called rebalancing events.

Choose a replacement behavior depending on your availability requirements

Mixed behavior
● No policy
For rebalancing events, new instances will launch before terminating others. For all other events, instances terminate and launch at the same time.

Prioritize availability
○ Launch before terminating
Launch new instances and wait for them to be ready before terminating others. This allows you to go above your desired capacity by a given percentage and may temporarily increase costs.

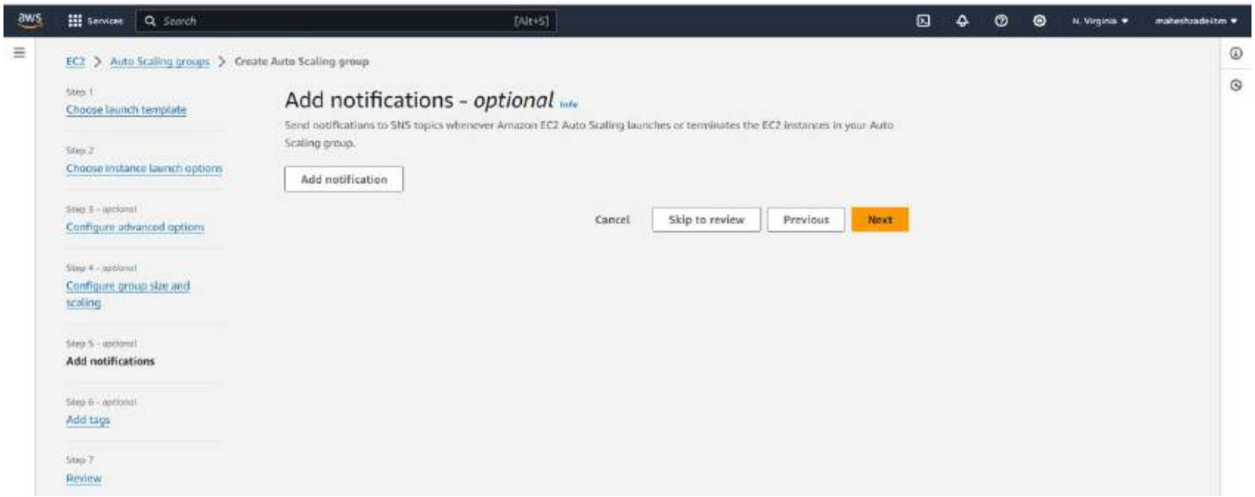
Control costs
○ Terminate and launch
Terminate and launch instances at the same time. This allows you to go below your desired capacity by a given percentage and may temporarily reduce availability.

Flexible
○ Custom behavior
Set custom values for the minimum and maximum amount of available capacity. This gives you greater flexibility in setting how far below and over your desired capacity EC2 Auto Scaling goes when replacing instances.

Instance scale-in protection
Scale-in protection prevents newly launched instances from being terminated by scaling activities. Make sure to remove scale-in protection for the group or individual instances when instances are ready to be terminated.

☐ Enable instance scale-in protection

Next



EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template

Step 2
Choose instance launch options

Step 3 - optional
Configure advanced options

Step 4 - optional
Configure group size and scaling

Step 5 - optional
Add notifications

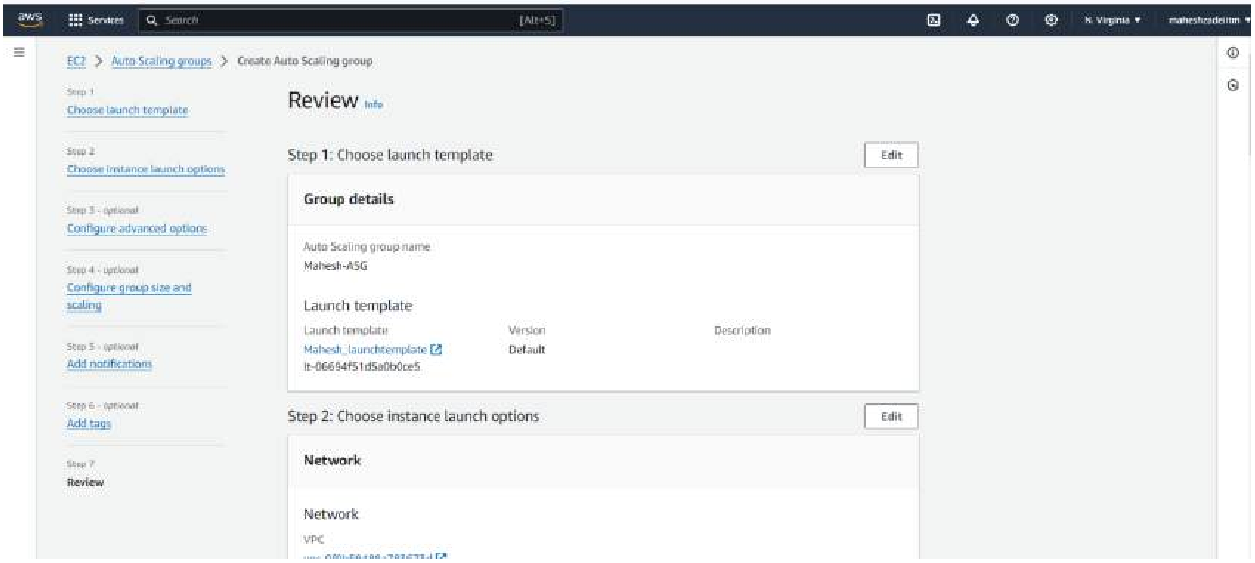
Step 6 - optional
Add tags

Step 7
Review

Add notifications - *optional* Info

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

Next



EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template

Step 2
Choose instance launch options

Step 3 - optional
Configure advanced options

Step 4 - optional
Configure group size and scaling

Step 5 - optional
Add notifications

Step 6 - optional
Add tags

Step 7
Review

Review Info

Step 1: Choose launch template

Group details

Auto Scaling group name
Mahesh-ASG

Launch template

Launch template	Version	Description
Mahesh_launchtemplate lt-06654f51d5a0b0ce5	Default	

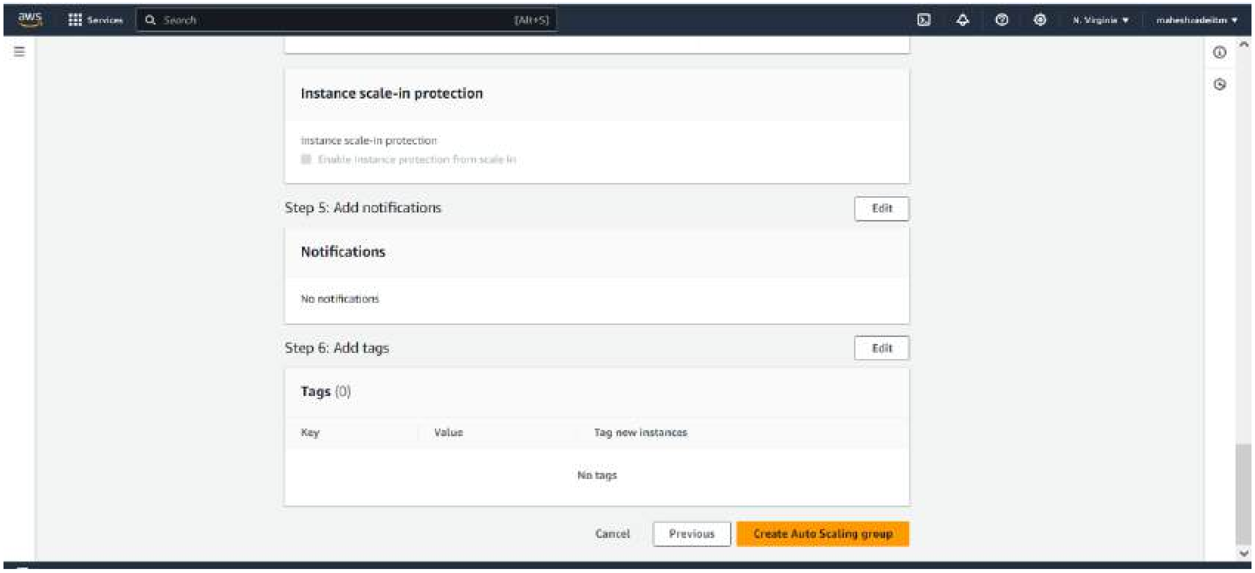
Step 2: Choose instance launch options

Network

Network

VPC

Click on create Auto Scaling Group



Instance scale-in protection

Instance scale-in protection

□ Enable instance protection from scale in

Step 5: Add notifications

Notifications

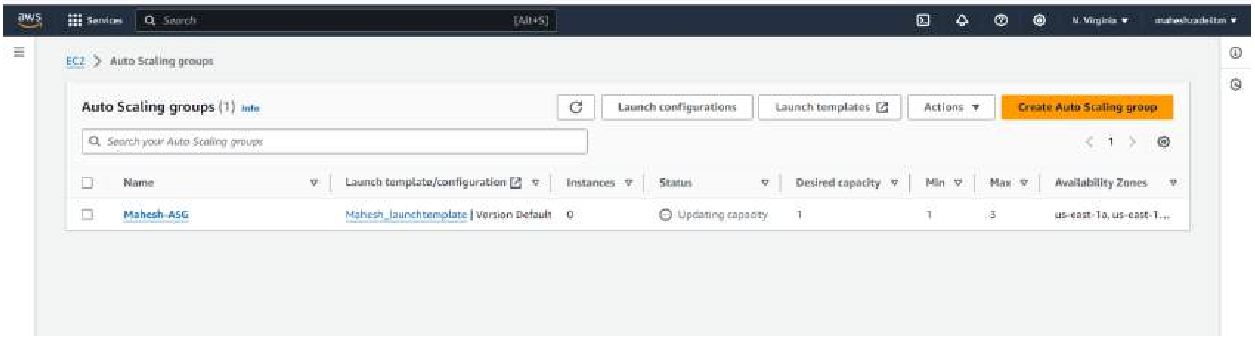
No notifications

Step 6: Add tags

Tags (0)

Key	Value	Tag new instances
No tags		

Crated Auto Scaling group

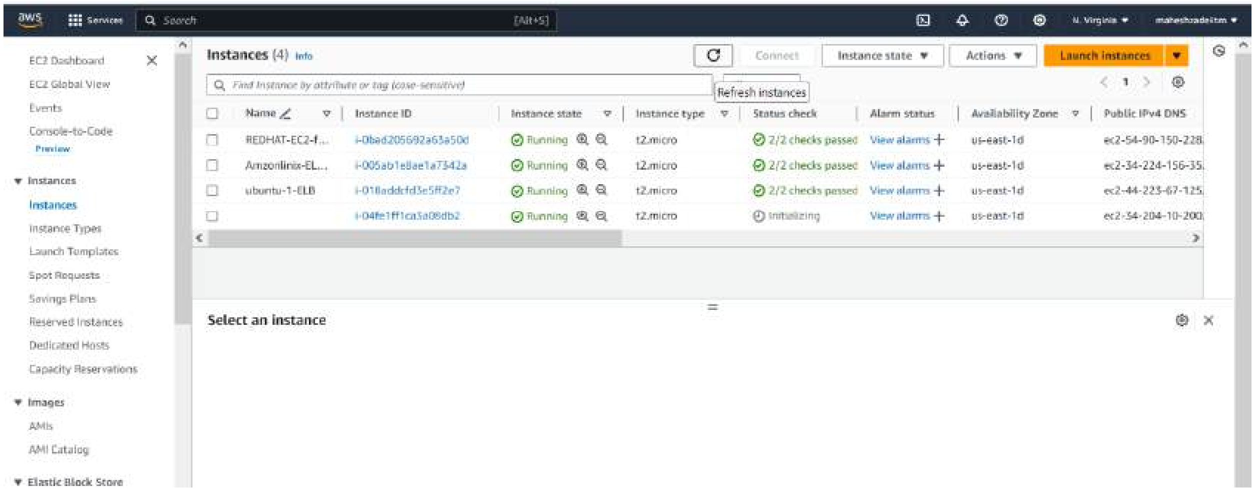


EC2 > Auto Scaling groups

Auto Scaling groups (1) Info

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
Mahesh-ASG	Mahesh_launchtemplate | Version Default	0	Updating capacity	1	1	3	us-east-1a, us-east-1...

EC2 is created in EC2 Dashboard

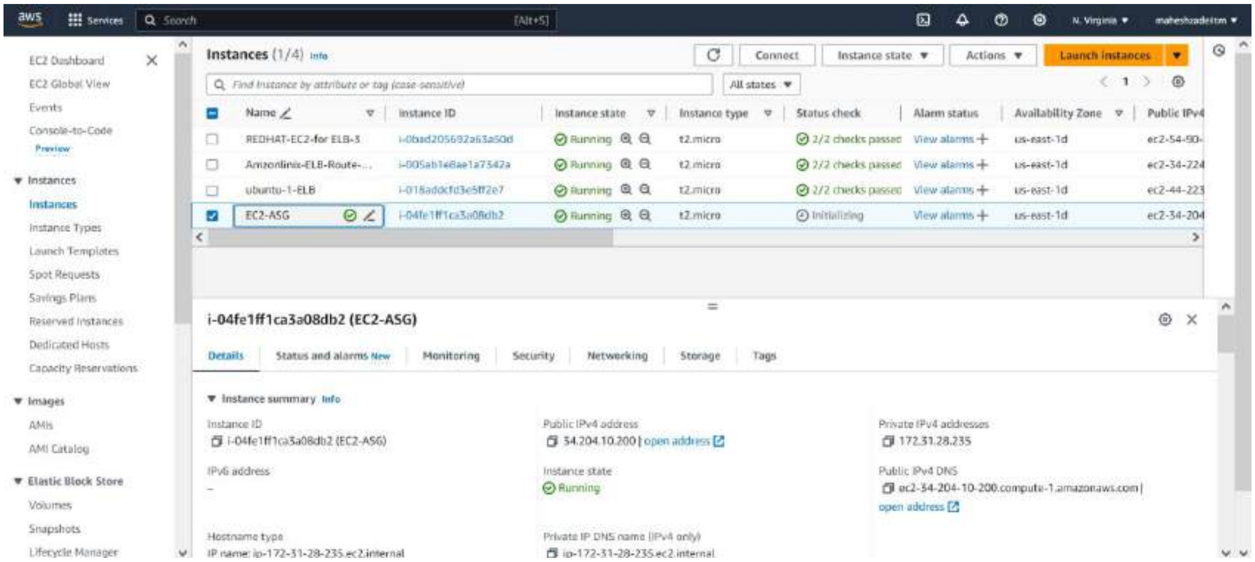


Instances (4) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
REDHAT-EC2-f...	i-0bad205692a63a50d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-54-90-150-228.
Amzonlinix-EL...	i-005ab1e8ae1a7342a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-34-224-156-35.
ubuntu-1-ELB	i-018addcfd3e5ff2e7	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-44-223-67-125.
	i-04fe1ff1ca5a08db2	Running	t2.micro	Initializing	View alarms +	us-east-1d	ec2-54-204-10-200.

Select an instance

Still initializing



Instances (1/4) Info

Find Instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
□	REDHAT-EC2-for ELB-3	i-0bad205692a63a50d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-54-90-
□	Amzonlinix-ELB-Route-...	i-005ab1e8ae1a7342a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-34-224
□	ubuntu-1-ELB	i-018addcfd3e5ff2e7	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-44-223
☑	EC2-ASG	i-04fe1ff1ca3a08db2	Running	t2.micro	Initializing	View alarms +	us-east-1d	ec2-34-204

i-04fe1ff1ca3a08db2 (EC2-ASG)

Details | Status and alarms New | Monitoring | Security | Networking | Storage | Tags

▼ Instance summary Info

Instance ID
i-04fe1ff1ca3a08db2 (EC2-ASG)

Public IPv4 address
54.204.10.200 | open address

Private IPv4 addresses
172.31.28.235

IPv6 address
–

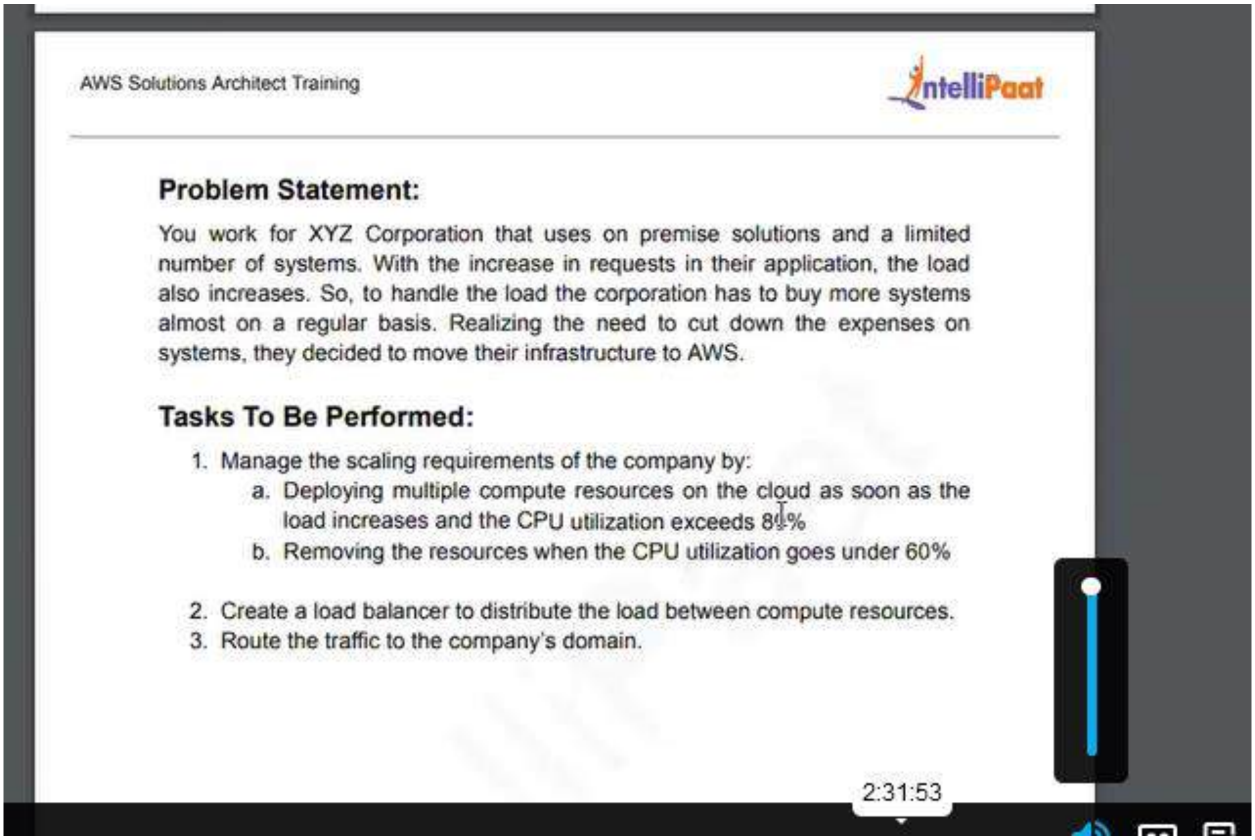
Instance state
Running

Public IPv4 DNS
ec2-34-204-10-200.compute-1.amazonaws.com | open address

Hostname type
IP name: ip-172-31-28-235.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-28-235.ec2.internal

EC2-ASG instacne is Available now by Auto sclaing group



AWS Solutions Architect Training

IntelliPaat

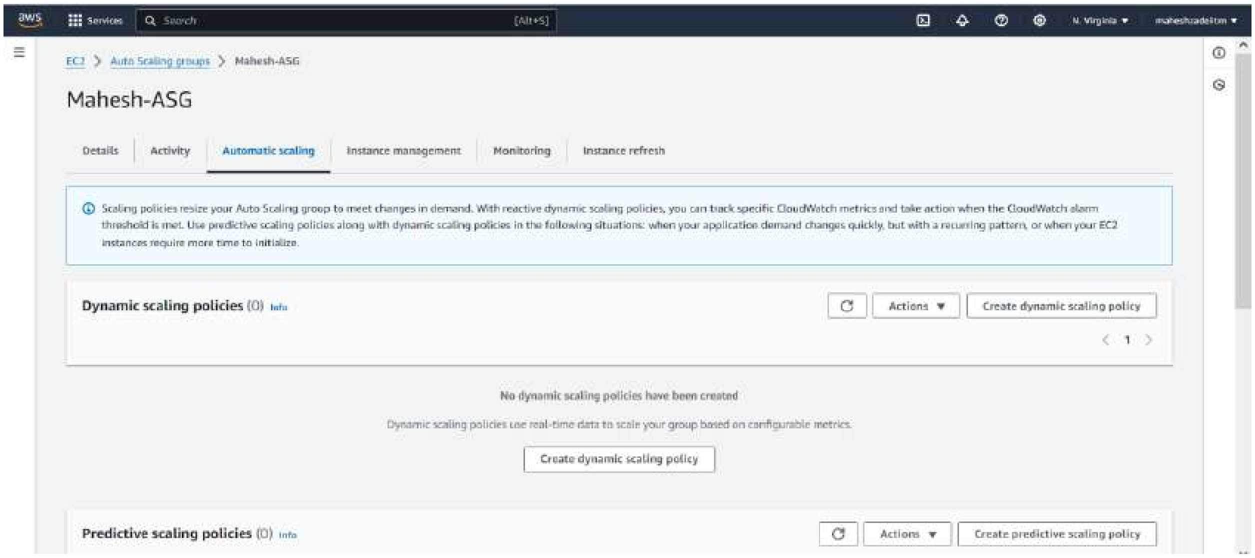
Problem Statement:

You work for XYZ Corporation that uses on premise solutions and a limited number of systems. With the increase in requests in their application, the load also increases. So, to handle the load the corporation has to buy more systems almost on a regular basis. Realizing the need to cut down the expenses on systems, they decided to move their infrastructure to AWS.

Tasks To Be Performed:

1. Manage the scaling requirements of the company by:
   a. Deploying multiple compute resources on the cloud as soon as the load increases and the CPU utilization exceeds 80%
   b. Removing the resources when the CPU utilization goes under 60%
2. Create a load balancer to distribute the load between compute resources.
3. Route the traffic to the company's domain.

Now Go to Auto scaling group for CPU Utilization in Automatic scaling option



Mahesh-ASG

Scaling policies resize your Auto Scaling group to meet changes in demand. With reactive dynamic scaling policies, you can track specific CloudWatch metrics and take action when the CloudWatch alarm threshold is met. Use predictive scaling policies along with dynamic scaling policies in the following situations: when your application demand changes quickly, but with a recurring pattern, or when your EC2 instances require more time to initialize.

Dynamic scaling policies (0) Info

No dynamic scaling policies have been created

Dynamic scaling policies use real-time data to scale your group based on configurable metrics.

Predictive scaling policies (0) Info

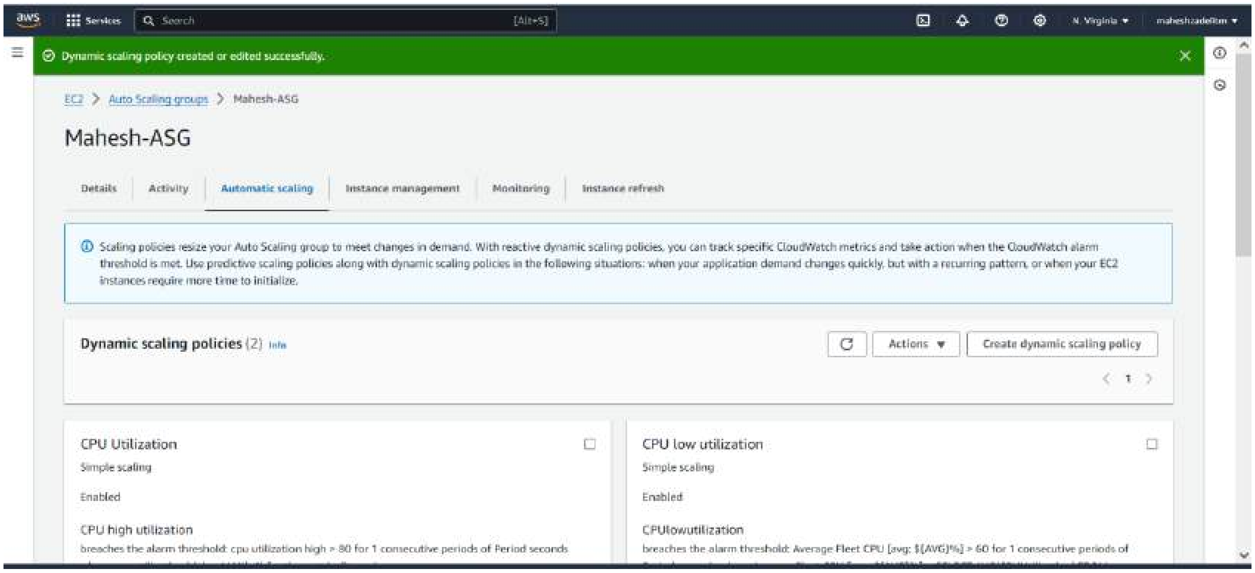
Dynamic scaling policies

Amazon EC2 Auto Scaling can add more instances (referred to as scaling out) to deal with high demand at peak times, and run fewer instances (referred to as scaling in) to reduce costs during periods of low utilization.

When you create a target tracking scaling policy, Amazon EC2 Auto Scaling automatically increases and decreases capacity in response to varying usage levels. For example, a target tracking scaling policy might have a target CPU value of 50 percent. Amazon EC2 Auto Scaling then launches and terminates EC2 instances as required to keep the aggregated CPU usage across all instances in your group at 50 percent.

With step scaling and simple scaling, you must create alarms in Amazon CloudWatch, and then define two policies, one for scaling out and the other for scaling in. Step scaling can make bigger or smaller size adjustments based on the metric value, while simple scaling always makes the same size adjustment.

Click on Create dynamic scaling policy



Dynamic scaling policy created or edited successfully.

Mahesh-ASG

Scaling policies resize your Auto Scaling group to meet changes in demand. With reactive dynamic scaling policies, you can track specific CloudWatch metrics and take action when the CloudWatch alarm threshold is met. Use predictive scaling policies along with dynamic scaling policies in the following situations: when your application demand changes quickly, but with a recurring pattern, or when your EC2 instances require more time to initialize.

Dynamic scaling policies (2) Info

CPU Utilization □
Simple scaling

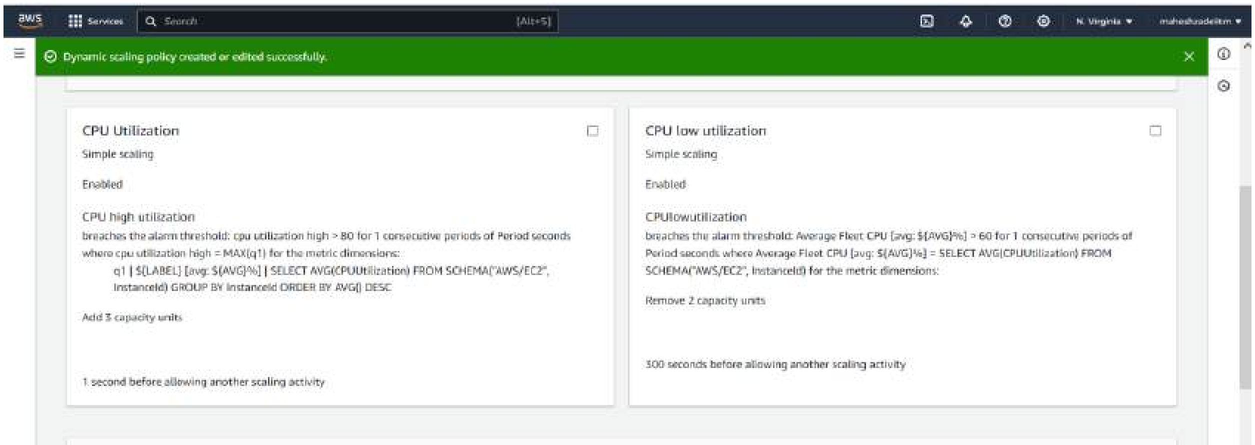
Enabled

CPU high utilization
breaches the alarm threshold: cpu utilization high > 80 for 1 consecutive periods of Period seconds

CPU low utilization □
Simple scaling

Enabled

CPUlowutilization
breaches the alarm threshold: Average Fleet CPU [avg: ${AVG}%] > 60 for 1 consecutive periods of



Dynamic scaling policy created or edited successfully.

CPU Utilization □
Simple scaling

Enabled

CPU high utilization
breaches the alarm threshold: cpu utilization high > 80 for 1 consecutive periods of Period seconds where cpu utilization high = MAX(q1) for the metric dimensions:
q1 | ${LABEL} [avg: ${AVG}%] | SELECT AVG(CPUUtilization) FROM SCHEMA("AWS/EC2", InstanceId) GROUP BY InstanceId ORDER BY AVG() DESC

Add 3 capacity units

1 second before allowing another scaling activity

CPU low utilization □
Simple scaling

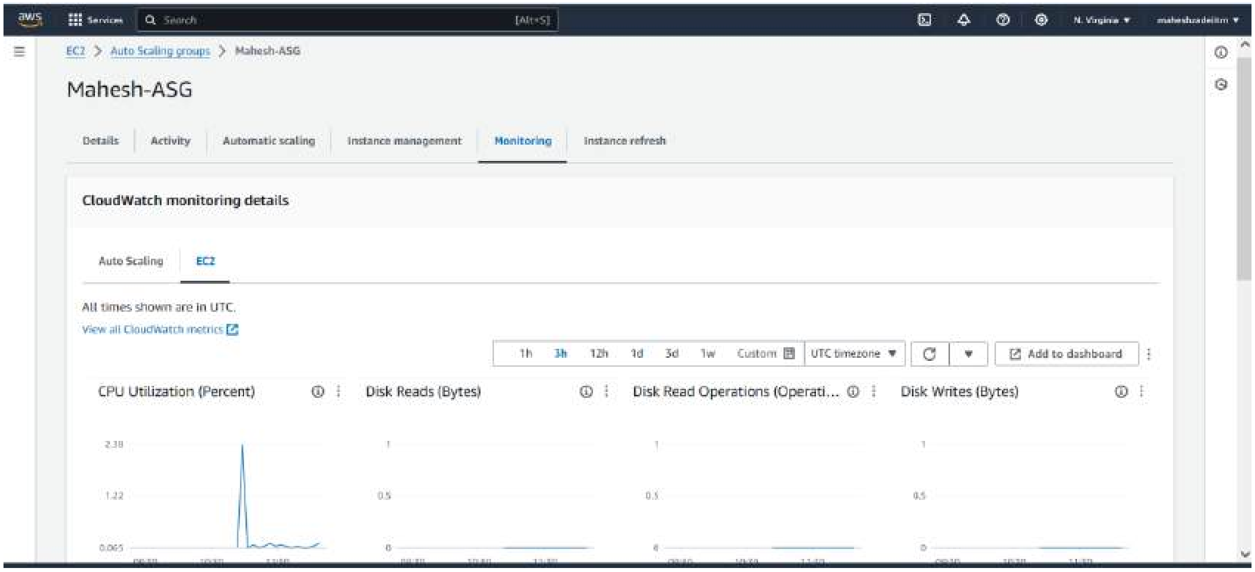
Enabled

CPUlowutilization
breaches the alarm threshold: Average Fleet CPU [avg: ${AVG}%] > 60 for 1 consecutive periods of Period seconds where Average Fleet CPU [avg: ${AVG}%] = SELECT AVG(CPUUtilization) FROM SCHEMA("AWS/EC2", InstanceId) for the metric dimensions:

Remove 2 capacity units

300 seconds before allowing another scaling activity

Now go to Monitoring and check



EC2 > Auto Scaling groups > Mahesh-ASG

Mahesh-ASG

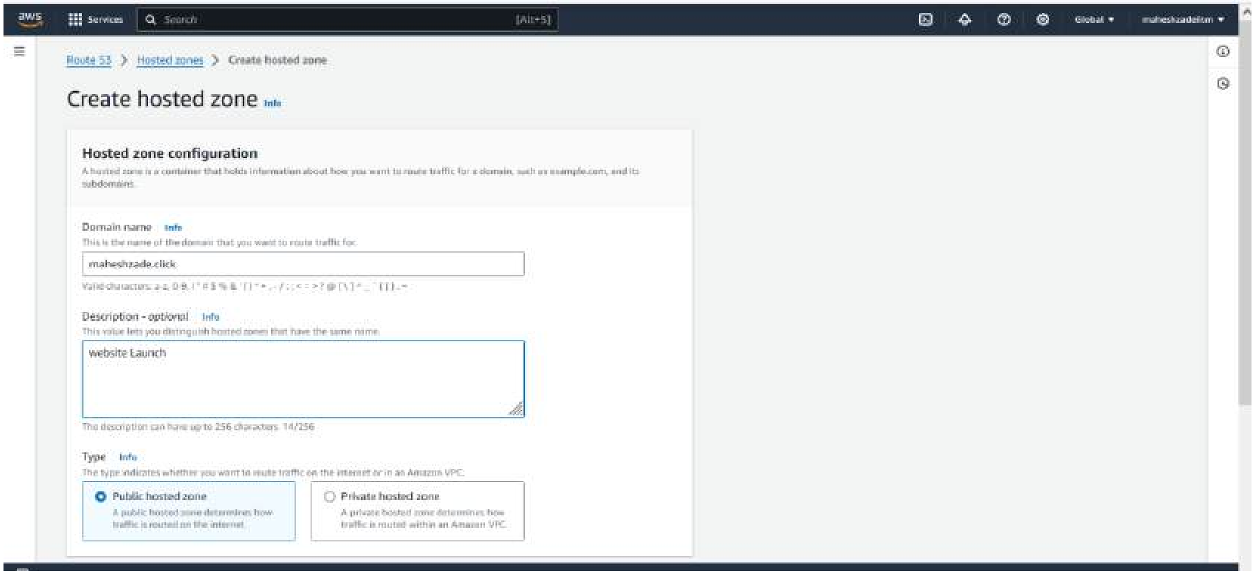
CloudWatch monitoring details

All times shown are in UTC.

View all CloudWatch metrics

Route 53 for Traffic Routing:

Create a hosted zone in Amazon Route 53 with a record set pointing to the ALB's DNS name.



Route 53 > Hosted zones > Create hosted zone

Create hosted zone Info

Hosted zone configuration

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain name Info

This is the name of the domain that you want to route traffic for.

maheshzade.click

Valid characters: a-z, 0-9, ! " # $ % & ' ( ) * + , - / : ; < = > ? @ [\ ] ^ _ ` { | } . ~

Description - optional Info

This value lets you distinguish hosted zones that have the same name.

website Launch

The description can have up to 256 characters. 14/256

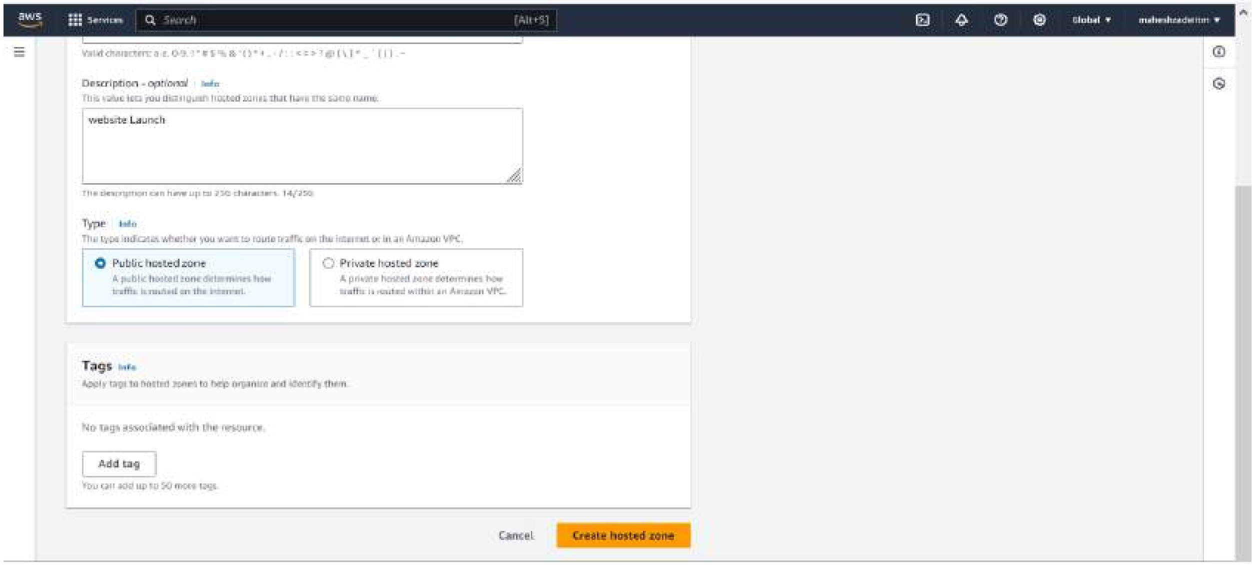
Type Info

The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

● Public hosted zone
A public hosted zone determines how traffic is routed on the internet.

○ Private hosted zone
A private hosted zone determines how traffic is routed within an Amazon VPC.

Create hosted zone



Valid characters: a-z, 0-9, ! " # $ % & ' ( ) * + , - / : ; < = > ? @ [\ ] ^ _ ` { | } . ~

Description - *optional* Info

This value lets you distinguish hosted zones that have the same name.

website Launch

The description can have up to 256 characters. 14/256

Type Info

The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

● Public hosted zone
A public hosted zone determines how traffic is routed on the internet.

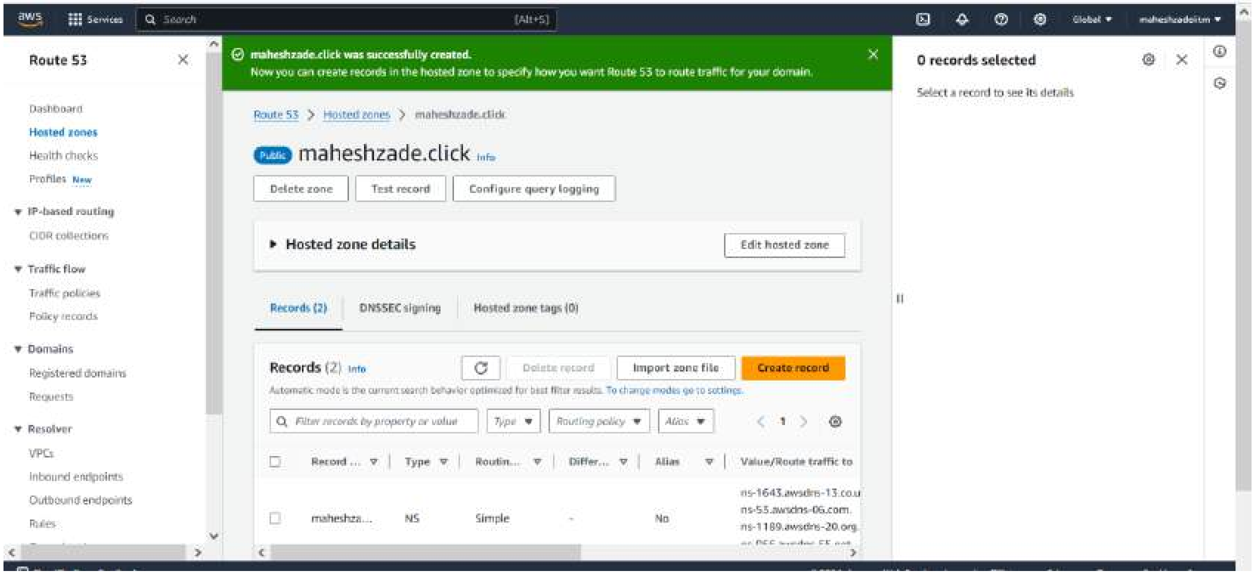
○ Private hosted zone
A private hosted zone determines how traffic is routed within an Amazon VPC.

Tags Info

Apply tags to hosted zones to help organize and identify them.

No tags associated with the resource.

You can add up to 50 more tags.



maheshzade.click was successfully created.
Now you can create records in the hosted zone to specify how you want Route 53 to route traffic for your domain.

maheshzade.click Info

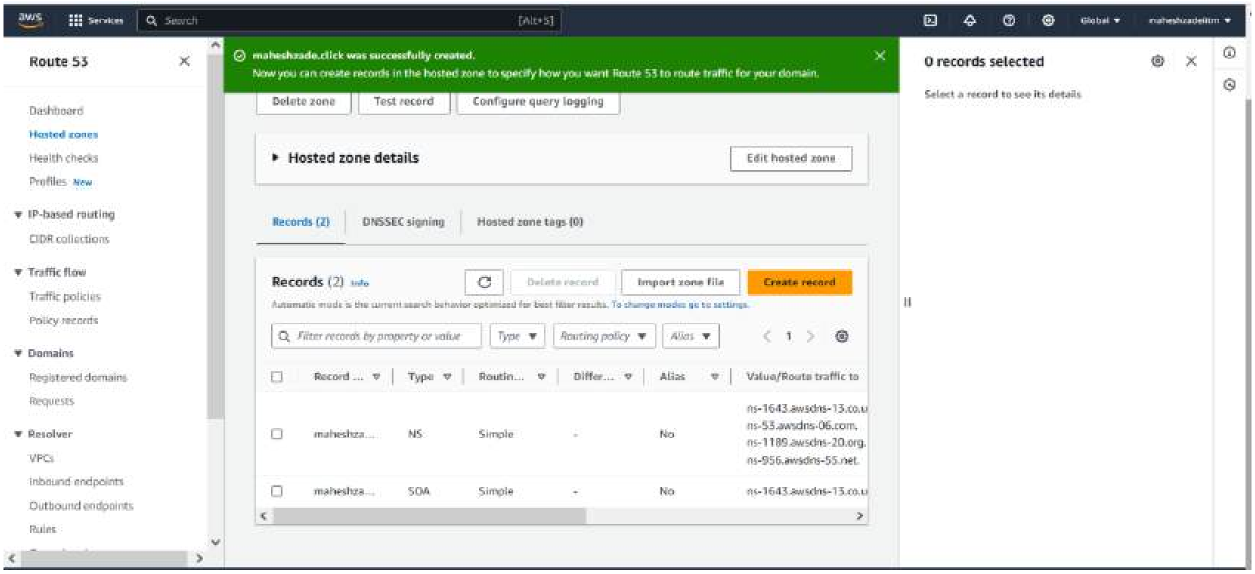
Records (2) Info

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

□	Record ...	Type	Routin...	Differ...	Alias	Value/Route traffic to
□	maheshza...	NS	Simple	-	No	ns-1643.awsdns-13.co.u ns-53.awsdns-06.com. ns-1189.awsdns-20.org.

0 records selected

Select a record to see its details



maheshzade.click was successfully created.
Now you can create records in the hosted zone to specify how you want Route 53 to route traffic for your domain.

Records (2)

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

□	Record ...	Type	Routin...	Differ...	Alias	Value/Route traffic to
□	maheshza...	NS	Simple	-	No	ns-1643.awsdns-13.co.u ns-53.awsdns-06.com. ns-1189.awsdns-20.org. ns-956.awsdns-55.net.
□	maheshza...	SOA	Simple	-	No	ns-1643.awsdns-13.co.u

0 records selected

Select a record to see its details

Configure your domain registrar to point the company's domain (e.g., xyzcorp.com) to the Route 53 hosted zone.(As Comapany domain is not available we are wring the steps).

Configuring Your Domain Registrar to Point to Route 53

Here's how to configure your domain registrar to point XYZ Corporation's domain (xyzcorp.com) to the Route 53 hosted zone:

1. Access Your Domain Registrar Account:

Login to the control panel of your domain registrar (e.g., GoDaddy, Google Domains, Namecheap).

2. Locate Domain Management:

Navigate to the section where you manage your domain name (often labeled "Domains," "DNS Management," or similar).

3. Find DNS Records:

Look for a section related to managing DNS records for your domain. This might be called "DNS Management," "Advanced Settings," or "Name Servers."

4. Update Nameserver (NS) Records:

we need to update the nameserver (NS) records for your domain. These records point to the servers that manage your domain's DNS information.

Replace the existing NS records with the nameservers provided by Route 53 when you created the hosted zone.

Typically, you'll find four Route 53 nameservers starting with "ns-".

5. Save Changes:

Once you've replaced the NS records with Route 53 nameservers, save your changes.

Propagation Time:

It can take up to 24 hours for the changes to propagate throughout the internet. During this time, your domain might still point to the old DNS servers.

Additional Tips:

Consult your domain registrar's documentation for specific instructions on updating NS records.

Consider taking a screenshot of your current DNS records before making changes, in case you need to revert.

By following these steps, you'll successfully configure XYZ Corporation's domain (xyzcorp.com) to point to the Route 53 hosted zone, allowing you to manage its DNS records and route traffic to your AWS application.

Benefits:

Automatic Scaling: Automatically adjusts resources based on demand, eliminating the need for manual provisioning.

Cost Optimization: Utilizes resources efficiently by scaling up during peak loads and down during low periods.

High Availability: The ALB distributes traffic across healthy instances, ensuring application uptime if one instance fails.

Scalability: Easily scales to accommodate future growth in traffic without infrastructure limitations.

Additional Considerations:

Monitoring: Use CloudWatch to monitor key metrics like CPU utilization, network traffic, and application health.

Security: Implement security groups to restrict inbound and outbound traffic for your EC2 instances.

Notifications: Set up notifications for scaling events and any potential issues.

This solution provides XYZ Corporation with an automated and cost-effective way to manage scaling requirements, ensure high availability, and route traffic to their application on AWS.

Assigmnet for ELBS and Route 53 Competed Succesfully .

Thanks

Mahesh Zade

Pls check and update

Thanks You