Secure VPC Architecture for Production Environments

Welcome to our comprehensive guide on implementing a secure Virtual Private Cloud (VPC) architecture for production environments. This presentation will explore the intricacies of designing a robust network infrastructure that leverages both public and private subnets across multiple Availability Zones. We'll delve into the key components, including NAT gateways, load balancers, and Auto Scaling groups, that work in harmony to create a scalable, resilient, and secure cloud environment. By the end of this presentation, you'll have a deep understanding of how to architect a VPC that meets the demanding requirements of modern production workloads.

MA **By, N. Levis NGAKOP**

VPC Fundamentals and Multi-AZ Design

VPC Basics

A Virtual Private Cloud (VPC) is a logically isolated section of the AWS cloud where you can launch resources in a virtual network that you define. It provides complete control over your virtual networking environment, including IP address ranges, subnets, and route tables.

Multi-AZ Architecture

Distributing resources across multiple Availability Zones (AZs) is crucial for high availability and fault tolerance. This design ensures that your application remains operational even if one AZ experiences an outage.

Subnet Strategy

Implementing both public and private subnets allows for a layered security approach. Public subnets host internet-facing resources, while private subnets contain sensitive backend systems, protected from direct internet access.

Public Subnet Components

NAT Gateways

Network Address Translation (NAT) gateways enable instances in private subnets to initiate outbound traffic to the internet while preventing inbound traffic from reaching these instances. They provide a secure way for private resources to download updates or access external services.

Load Balancer Nodes

Load balancers distribute incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones. This improves the fault tolerance of your applications and ensures smooth traffic flow even during peak times.

Internet Gateways

An Internet Gateway allows communication between your VPC and the internet. It provides a target in your VPC route tables for internet-routable traffic and performs network address translation for instances with public IP addresses.

Security Groups

Security groups act as a virtual firewall for your instances to control inbound and outbound traffic. They operate at the instance level and can be configured to allow specific types of traffic to and from your public-facing resources.

Private Subnet Infrastructure

1

EC2 Instances

Amazon EC2 instances in private subnets host your application servers, databases, and other backend systems. These instances are not directly accessible from the internet, providing an additional layer of security.

2

Auto Scaling Groups

Auto Scaling groups automatically adjust the number of EC2 instances based on defined conditions. This ensures that you have the right number of instances available to handle your application's load, improving both performance and cost-efficiency.

3

Private Subnet Routing

Route tables for private subnets are configured to direct internet-bound traffic through the NAT gateway in the public subnet. This allows instances to access the internet for updates and external services while remaining protected from inbound connections.

4

VPC Endpoints

VPC endpoints enable private connections to supported AWS services without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect.

Network Security Measures

1 **Network Access Control Lists (NACLs)**

NACLs act as a firewall at the subnet level, controlling inbound and outbound traffic. They provide an additional layer of security beyond security groups and can be used to block malicious IP addresses or restrict traffic patterns.

2 **VPC Flow Logs**

VPC Flow Logs capture information about the IP traffic going to and from network interfaces in your VPC. This data can be used for network monitoring, security analysis, and troubleshooting, providing valuable insights into your VPC's network behavior.

3 **AWS WAF Integration**

Integrating AWS WAF (Web Application Firewall) with your load balancer allows you to protect your web applications from common web exploits. It can be configured to filter traffic based on rules that you specify, such as IP addresses, HTTP headers, and custom URI strings.

4 **Encryption in Transit and at Rest**

Implement SSL/TLS for all communications between your load balancer and EC2 instances. Use AWS KMS (Key Management Service) to manage encryption keys for data at rest, ensuring that sensitive information remains secure throughout its lifecycle.

High Availability and Fault Tolerance

1 2 3 4

Multi-AZ Deployment

Distribute resources across multiple Availability Zones to ensure high availability. This includes deploying EC2 instances, databases, and other critical components in at least two AZs to protect against localized

Load Balancer Configuration

Configure your load balancer to distribute traffic across multiple AZs. Use health checks to ensure that traffic is only routed to healthy instances, automatically removing unhealthy instances from the rotation.

Auto Scaling Policies

Implement dynamic scaling policies that automatically adjust the number of EC2 instances based on metrics such as CPU utilization or network traffic. This ensures that your application can handle sudden spikes in demand

Disaster Recovery Planning

Develop and regularly test a disaster recovery plan that includes data backups, failover procedures, and recovery time objectives. Consider implementing a multi-region strategy for critical applications to protect against region-

Monitoring and Logging

CloudWatch Metrics

Utilize Amazon CloudWatch to collect and track metrics for your VPC resources. Set up custom dashboards to visualize key performance indicators and configure alarms to notify you of potential issues before they impact your users.

Centralized Logging

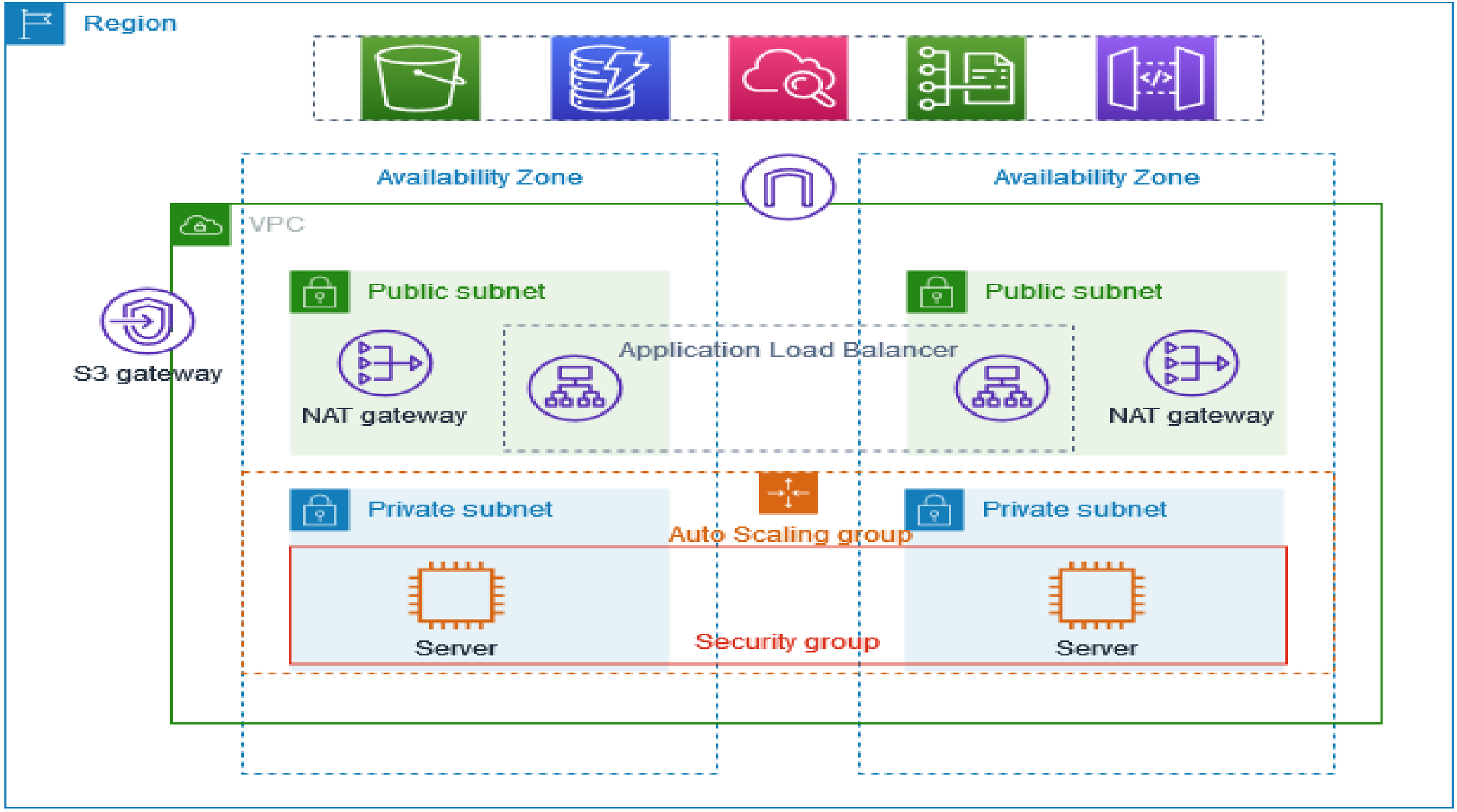
Implement a centralized logging solution using services like Amazon CloudWatch Logs or a third-party log management tool. This allows you to aggregate logs from multiple sources, making it easier to troubleshoot issues and perform security analysis.

Real-time Alerting

Configure alerts based on predefined thresholds for critical metrics such as CPU utilization, memory usage, and network traffic. Use SNS (Simple Notification Service) to send notifications to relevant team members or integrate with incident management systems.

AWS Config and CloudTrail

Enable AWS Config to assess, audit, and evaluate the configurations of your AWS resources. Use AWS CloudTrail to log, continuously monitor, and retain account activity related to actions across your AWS infrastructure.


Region
Availability Zone
Availability Zone
VPC
Public subnet
Public subnet
S3 gateway
NAT gateway
Application Load Balancer
NAT gateway
Private subnet
Private subnet
Auto Scaling group
Server
Security group
Server

Step 1 :

Create the VPC :

Open the Amazon VPC console by visiting https://console.aws.amazon.com/vpc/.

On the dashboard, click on “Create VPC.”

Under “Resources to create,” select “VPC and more.”

Configure the VPC:

a. Provide a name for the VPC in the “Name tag auto-generation” field.
b. For the IPv4 CIDR block, leave it as default suggestion.

Configure the subnets:

a. Set the “Number of Availability Zones” to 2 for increased resiliency across multiple Availability Zones.

b. Specify the “Number of public subnets” as 2.

c. Specify the “Number of private subnets” as 2.

d. For NAT gateways, choose “1 per AZ” to enhance resiliency.

g. For VPC endpoints, you can choose “None” .

h. Regarding DNS options, clear the checkbox for “Enable DNS hostnames.”

Once you’ve configured all the settings, click “Create VPC.”

VPC dashboard
EC2 Global View New
Filter by VPC:
Select a VPC

Virtual private cloud
Your VPCs New
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections

Security
Network ACLs

Create VPC | Launch EC2 Instances

Note: Your Instances will launch in the US East region.

Resources by Region

Refresh Resources

You are using the following Amazon VPC resources

VPCs (See all regions)	US East 1	NAT Gateways (See all regions)	US East 0
Subnets (See all regions)	US East 6	VPC Peering Connections (See all regions)	US East 0
Route Tables (See all regions)	US East 1	Network ACLs (See all regions)	US East 1
Internet Gateways (See all regions)	US East 1	Security Groups (See all regions)	US East 1
Egress-only Internet Gateways (See all regions)	US East 0	Customer Gateways (See all regions)	US East 0
DHCP option sets (See all regions)	US East 1	Virtual Private Gateways (See all regions)	US East 0

Service Health

View complete service health details

Settings

Zones
Console Experiments

Additional Information

VPC Documentation
All VPC Resources
Forums
Report an Issue

AWS Network Manager

AWS Network Manager provides tools and features to help you manage and monitor your network on AWS. Network Manager makes it easier to perform connectivity management, network monitoring and troubleshooting, IP management, and network security and governance.

Get started with Network Manager

Site-to-Site VPN Connections

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

- ○ VPC only
- ◉ VPC and more

Name tag auto-generation Info
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

- ☑ Auto-generate

aws-prod-demo

IPv4 CIDR block Info
Determine the starting IP and the size of your VPC using CIDR notation.

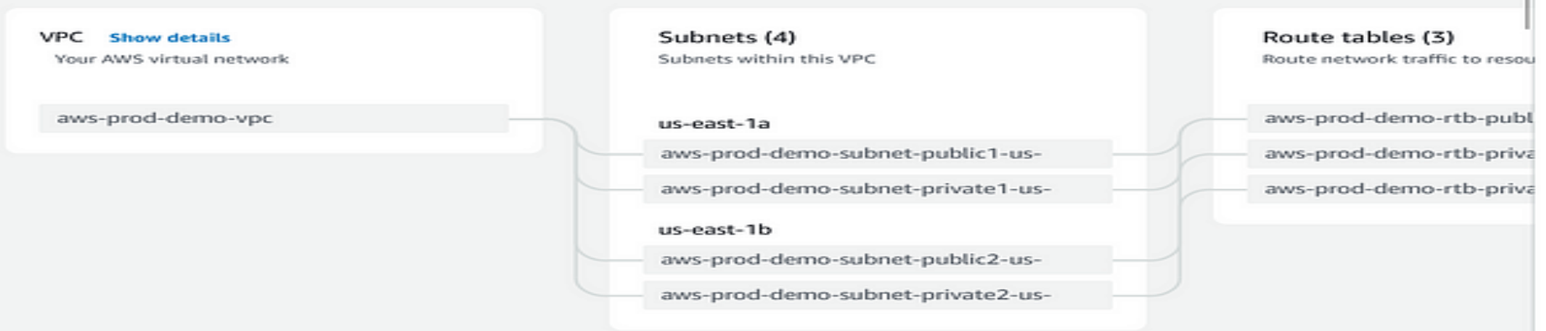
10.0.0.0/16 — 65,536 IPs

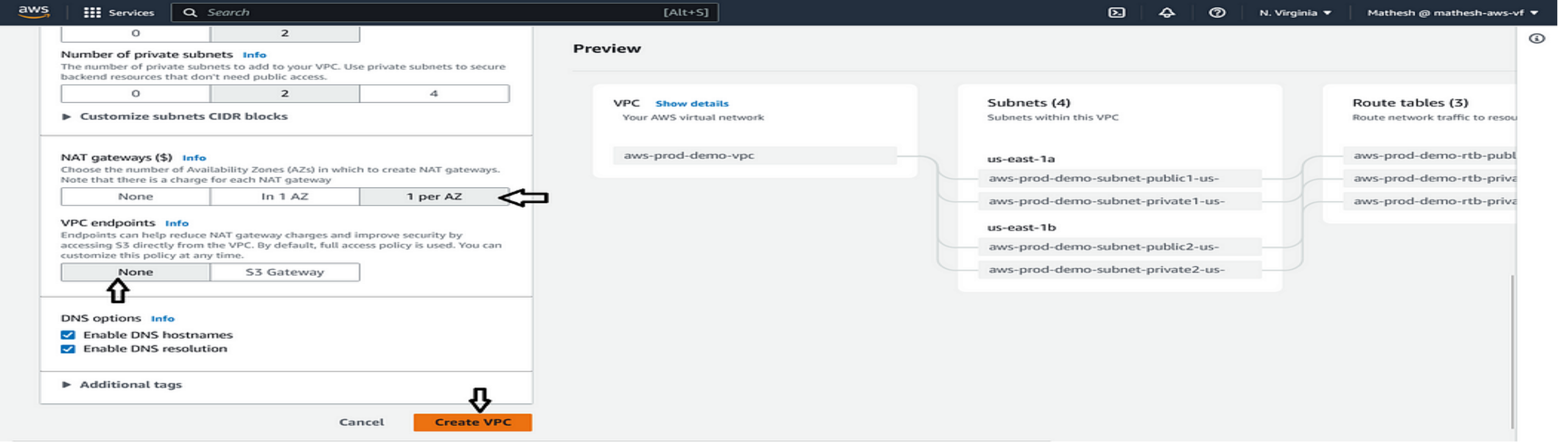
IPv6 CIDR block Info

- ◉ No IPv6 CIDR block
- ○ Amazon-provided IPv6 CIDR block

Tenancy Info

Preview

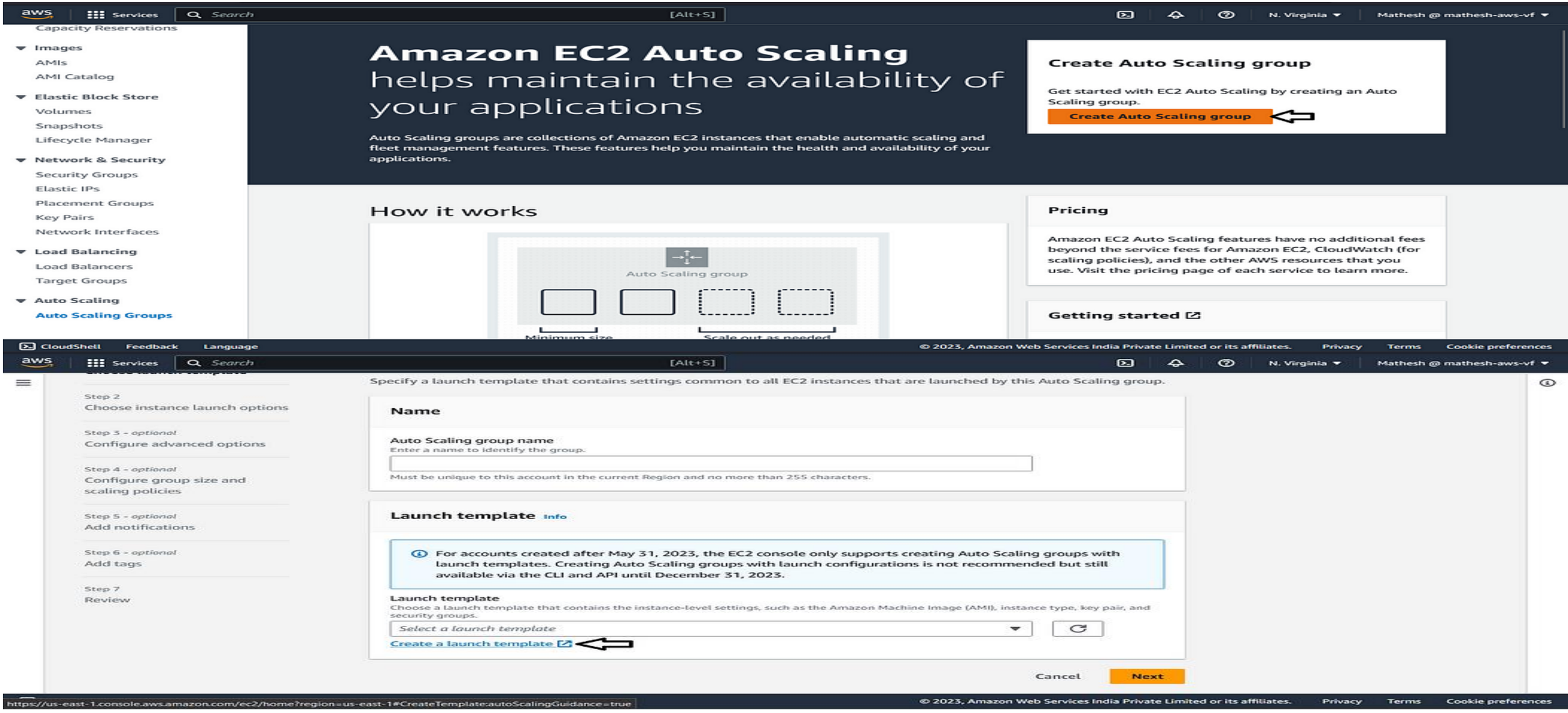




6. Now you can see the VPC is successfully Created .

Step 2:

Creating the Auto Scaling Group :



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*

aws-prod-template

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

Template version description

Demo Prod Template

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

▶ Template tags

▶ Source template

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, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Create launch template

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) - required Info

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

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

Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Li

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-053b0d53c279acc90 (64-bit (x86)) / ami-0a0c8eebcdd6dcbd0 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-05-16

Summary

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-053b0d53c279acc90

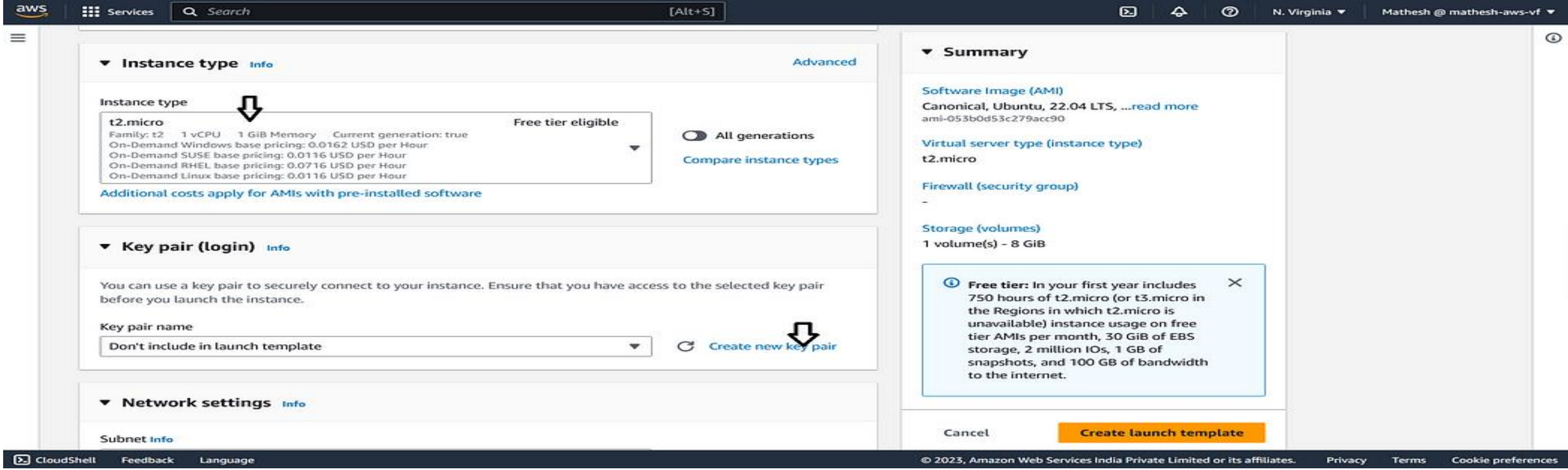
Virtual server type (instance type)
-

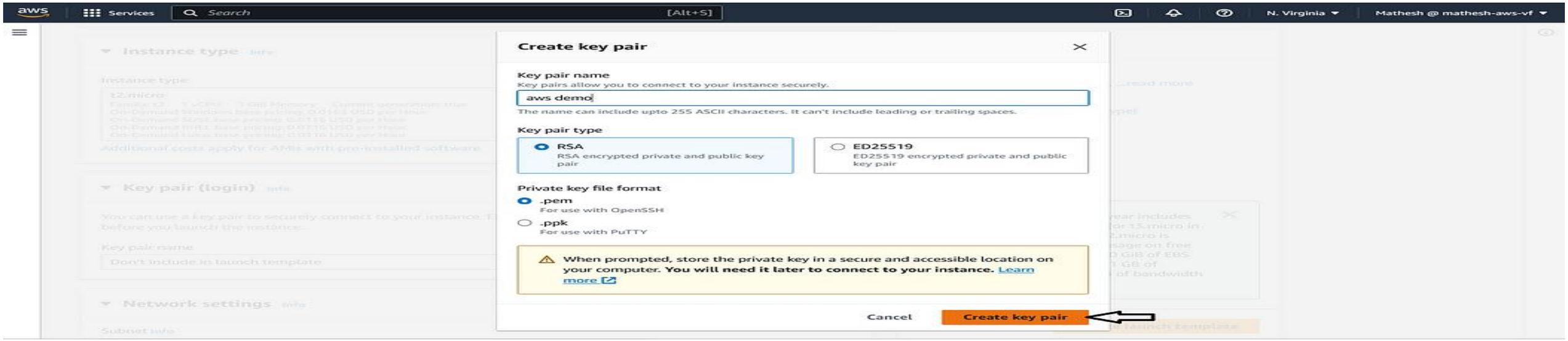
Firewall (security group)
-

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, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Create launch template





Now you have to choose the Key-pair you 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 group name - *required*

aws-prod-sg

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

Description - *required* Info

Allow SSH Access

VPC - *required* Info

vpc-0eb3f78ab19d3a68b (aws-prod-demo-vpc)
10.0.0.0/16

Inbound Security Group Rules

No security group rules are currently included in this template. Add a new rule to include it in the launch template.

Add security group rule

Summary

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-053b0d53c279acc90

Virtual server type (instance type)
t2.micro

Firewall (security group)
New 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, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Create launch template

aws | Services | Search [Alt+S] | N. Virginia | Mathesh @ mathesh-aws-vf

Inbound Security Group Rules

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

Type Info	Protocol Info	Port range Info
ssh	TCP	22

Source type Info	Source Info	Description - *optional* Info
Custom	Add CIDR, prefix list or security: 0.0.0.0/0	e.g. SSH for admin desktop

Security group rule 2 (TCP, 8000, 0.0.0.0/0) — Remove

Type Info	Protocol Info	Port range Info
Custom TCP	TCP	8000

Source type Info	Source Info	Description - *optional* Info
Custom	Add CIDR, prefix list or security: 0.0.0.0/0	e.g. SSH for admin desktop

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

Add security group rule

Summary

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-053b0d53c279acc90

Virtual server type (instance type)
t2.micro

Firewall (security group)
New 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, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Create launch template

CloudShell | Feedback | Language

© 2023, Amazon Web Services India Private Limited or its affiliates. | Privacy | Terms | Cookie preferences

EBS Volumes

Hide details

▶ Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp2))
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)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-053b0d53c279acc90

Virtual server type (instance type)
t2.micro

Firewall (security group)
New 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, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Create launch template

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 policies

Step 5 - *optional*
Add notifications

Step 6 - *optional*
Add tags

Step 7
Review

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

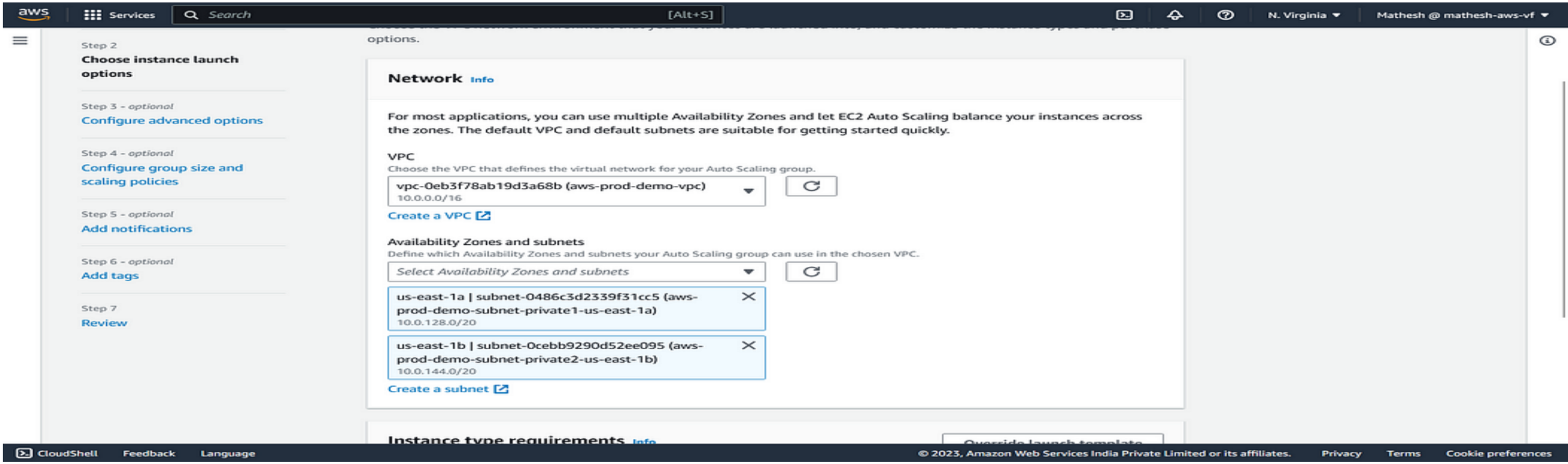
Auto Scaling group name
Enter a name to identify the group.
aws-prod-asg
Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info

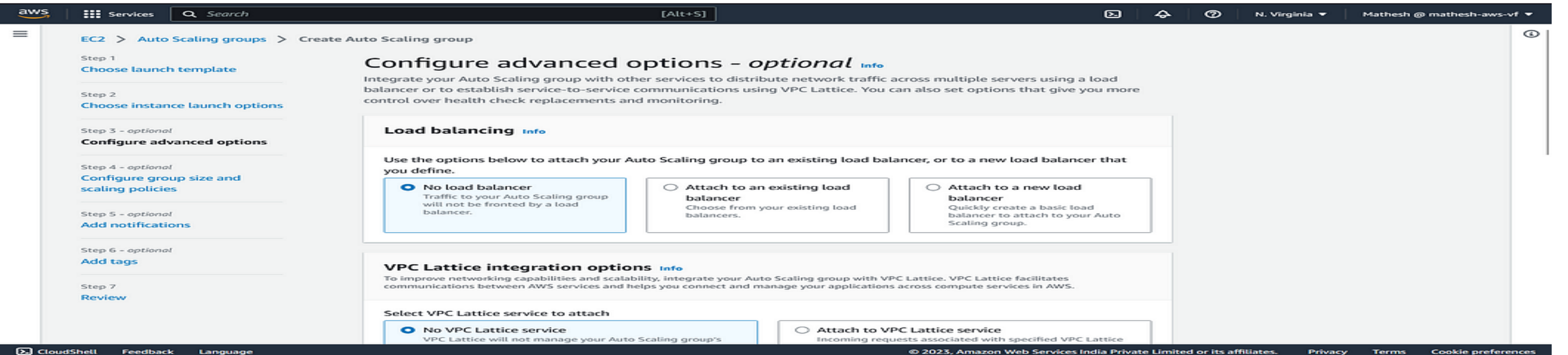
For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

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.
aws-prod-template

Create a launch template



3. Scroll Down and then Click “Next”.



4. Scroll Down and then Click “Next”.

Desired capacity
Specify your group size.

2

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

2

Equal or less than desired capacity

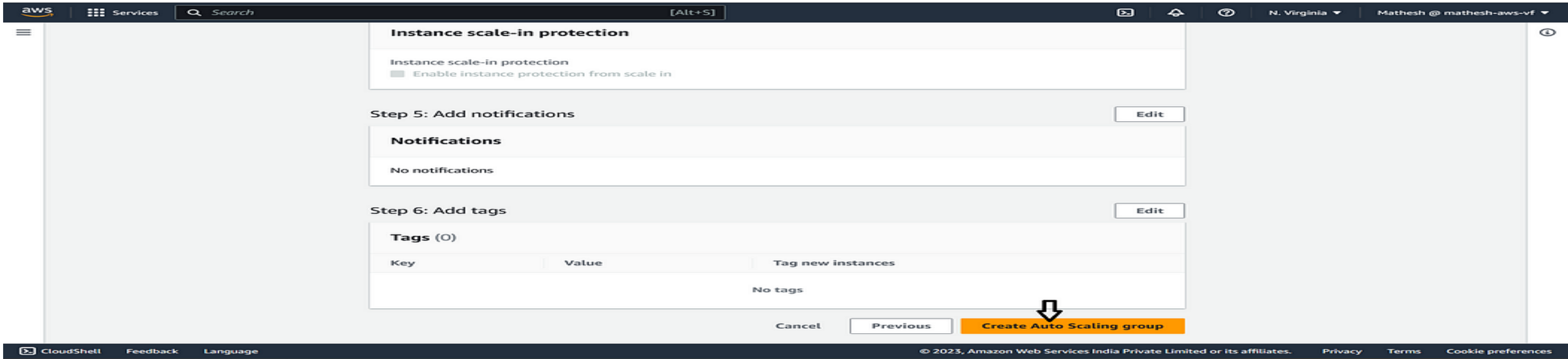
Max desired capacity

4

Equal or greater than desired capacity

Automatic scaling – *optional*

5. Scroll Down and then Click “Skip to Review”.



6. Now your are Successfully Created Auto Scaling Group.

7. Open the AWS Management Console.

8. Navigate to the EC2 console by clicking on “Services” in the top-left corner, then selecting “EC2” under the “Compute” section.

9. In the EC2 dashboard, you’ll find the “Instances” link on the left-hand navigation pane. Click on “Instances.”

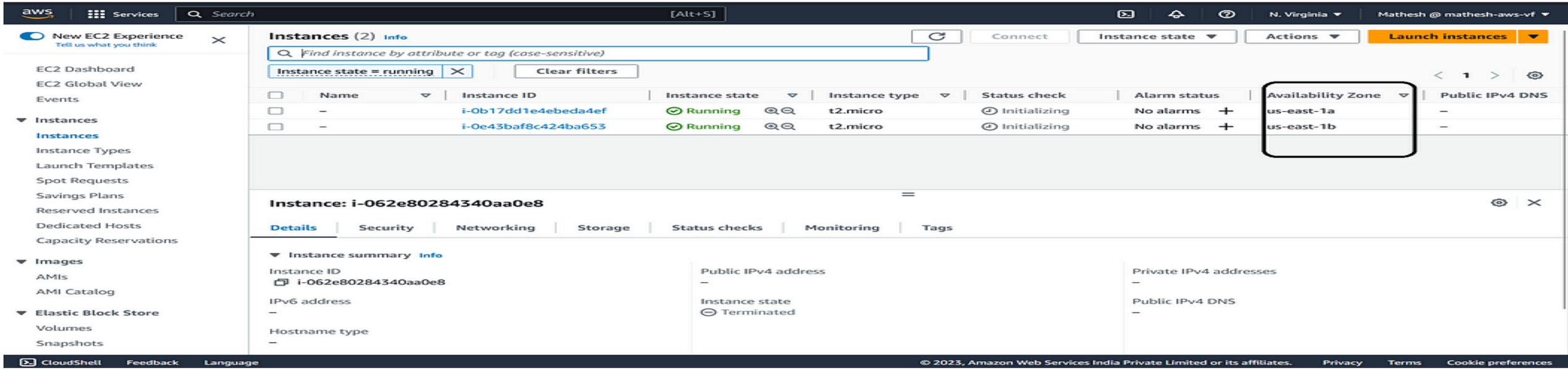
10. Here, you should see the list of EC2 instances associated with your account. Look for the instances created by your Auto Scaling Group.

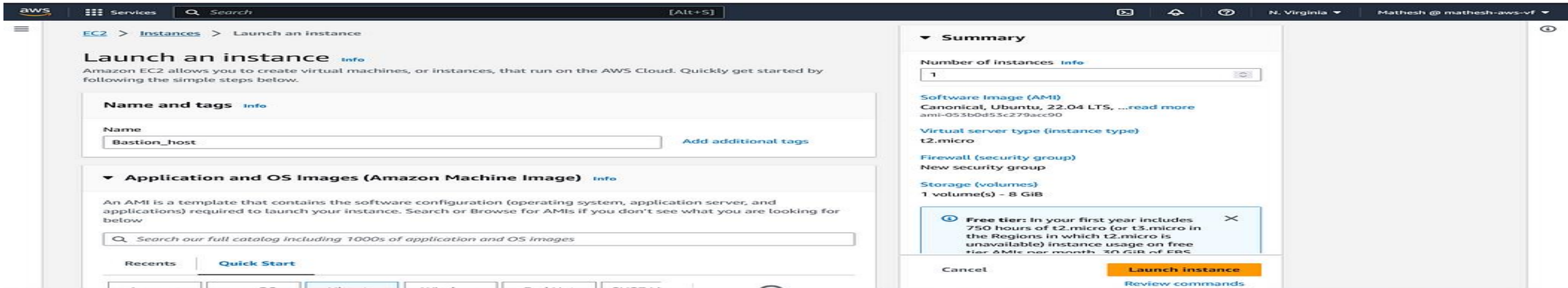
Since you mentioned that the Auto Scaling Group launched instances in different AZs, you can check the “Availability Zone” column to verify that these instances are indeed distributed across multiple AZs.

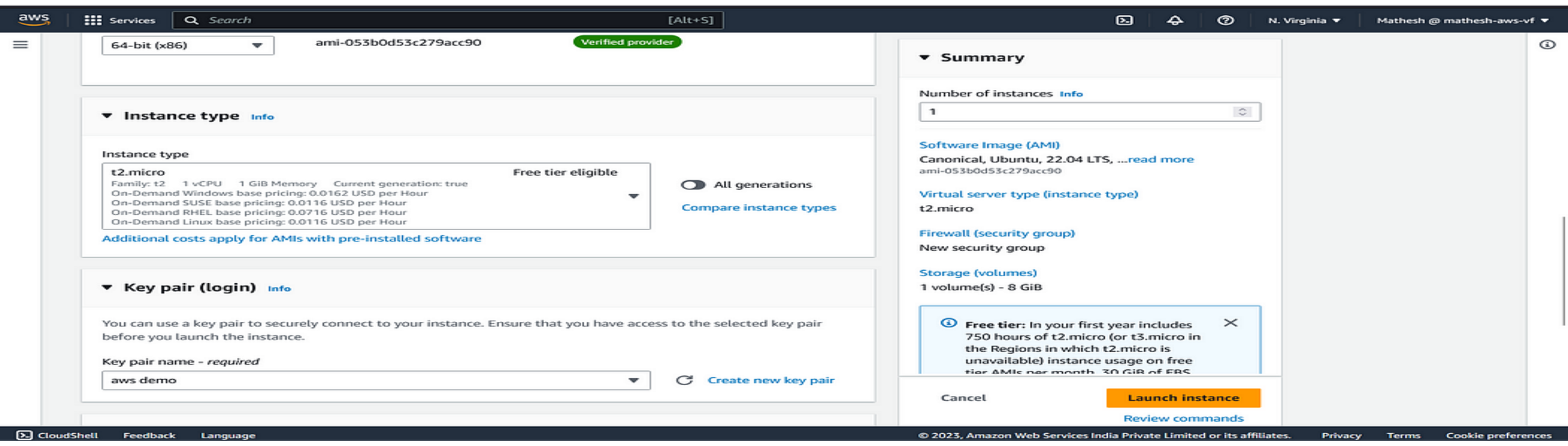
Step 3:

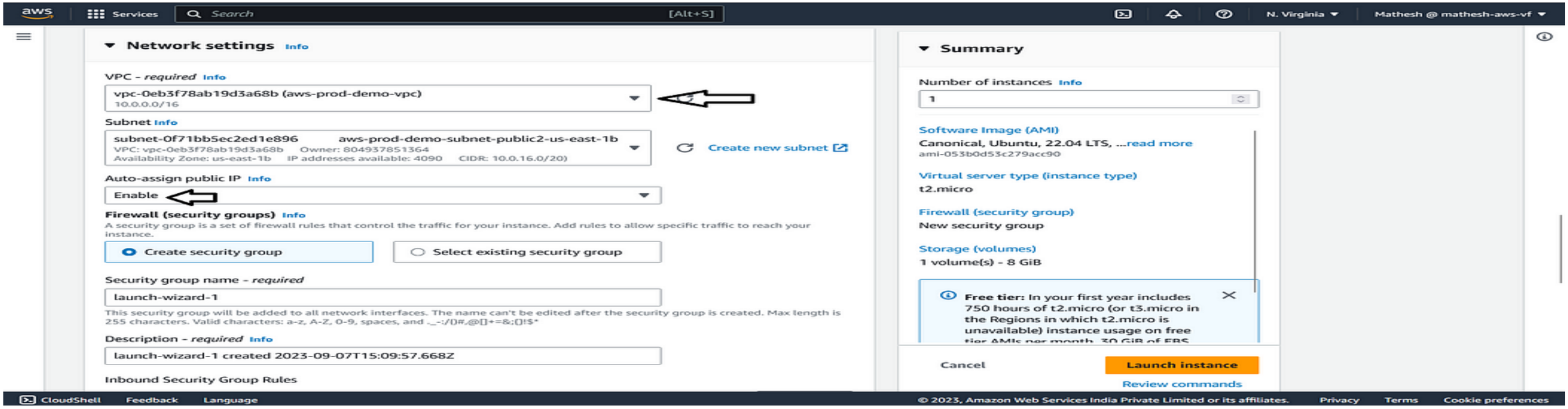
Creating the Bastion Host:

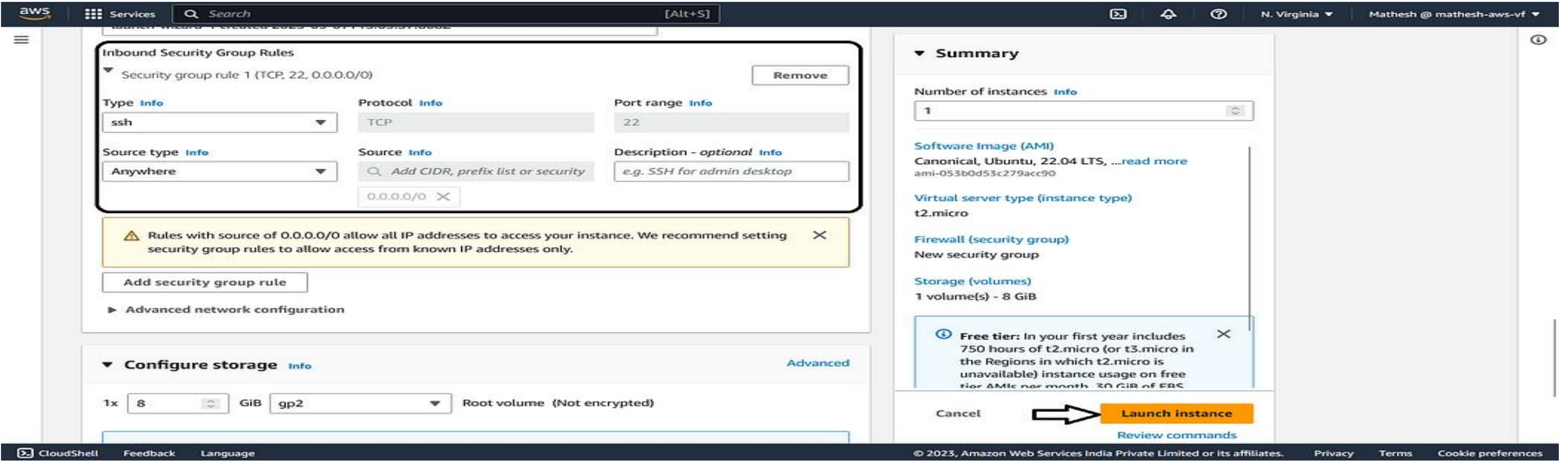
Launch Instance as Specified below.






Services
Search
[Alt+S]
N. Virginia
Mathesh @ mathesh-aws-vf
64-bit (x86)
ami-053b0d53c279acc90
Verified provider
Instance type Info
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 - required
aws demo
Create new key pair
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-053b0d53c279acc90
Virtual server type (instance type)
t2.micro
Firewall (security group)
New 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
Cancel
Launch instance
Review commands
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences


Services
Search
[Alt+S]
N. Virginia
Mathesh @ mathesh-aws-vf
Network settings Info
VPC - required Info
vpc-0eb3f78ab19d3a68b (aws-prod-demo-vpc)
10.0.0.0/16
Subnet Info
subnet-0f71bb5ec2ed1e896 aws-prod-demo-subnet-public2-us-east-1b
VPC: vpc-0eb3f78ab19d3a68b Owner: 804937851364
Availability Zone: us-east-1b IP addresses available: 4090 CIDR: 10.0.16.0/20)
Create new subnet
Auto-assign public IP Info
Enable
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
Security group name - required
launch-wizard-1
This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*
Description - required Info
launch-wizard-1 created 2023-09-07T15:09:57.668Z
Inbound Security Group Rules
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-053b0d53c279acc90
Virtual server type (instance type)
t2.micro
Firewall (security group)
New 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
Cancel
Launch instance
Review commands
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences



Step 4 :

SSH into Private Instance

1) - SSH into the Bastion Host Instance: To SSH into the private instances, we first need to connect to our Bastion host instance. From there, we'll be able to SSH into the private instance.

2) - Ensure the PEM File is Present on the Bastion Host: Additionally, make sure that the PEM file is present on the Bastion host. Without it, you won’t be able to SSH into the private instance from the Bastion host.

3) - Open a Terminal: Open a terminal window on your local machine.

4) - Execute the Following Commands:

- a. If your PEM file is named something like <paradigm_key.pem>, you must remove spaces in the filename. Please rename the file to something like <paradigm_key.pem>.
- b. Copy the PEM file to the Bastion host using the scp command. Replace <pem file location> with the local and remote file paths, and <bastion host public IP> with the Bastion host's public IP address. Example:

```
scp -i /c/Users/GLC/Downloads/paradigm_key.pem  /c/Users/GLC/Downloads/paradigm_key.pem
ubuntu@34.229.240.123:/home/ubuntu
```

- c. The above command will copy the PEM file from your computer to the Bastion host. Once the file is successfully copied, move on to the next step.
- d. SSH into the Bastion host using the following command:

```
ssh -i paradigm_key.pem ubuntu@public ip of bastion host
```

- e. After SSH into the Bastion host, use the ls command to check if the paradigm_key.pem file is present. If it's not there, double-check your previous commands. f. Now, you can SSH into the private instance using the following command, replacing <private IP> with the private instance's IP address:

```
ssh -i paradigm_key.pem ubuntu@<private IP>
```

- g. We will deploy our application on one of the private instances to test the load balancer. h. After successfully SSHing into the private instance, create an HTML file using the Vim text editor:

  vim index.html

- i. This will open the Vim editor. Copy and paste any HTML content you like into the editor. j. For example:

```
<!DOCTYPE html>
<html>
<head>
<title>AWS VPC PROJECT</title>
</head>
<body>

<h1>This Is a test VPC Demo Project</h1>
</body>
</html>
```

k. After pasting the content, save the file by pressing ‘Esc’ to exit insert mode and then entering :x to save.

Finally, can start your Python HTTP server on port 8000 to deploy your application on the private instance:

```
python3 -m http.server 8000
```

Now, your application is deployed on the private instance on port 8000.

Note :

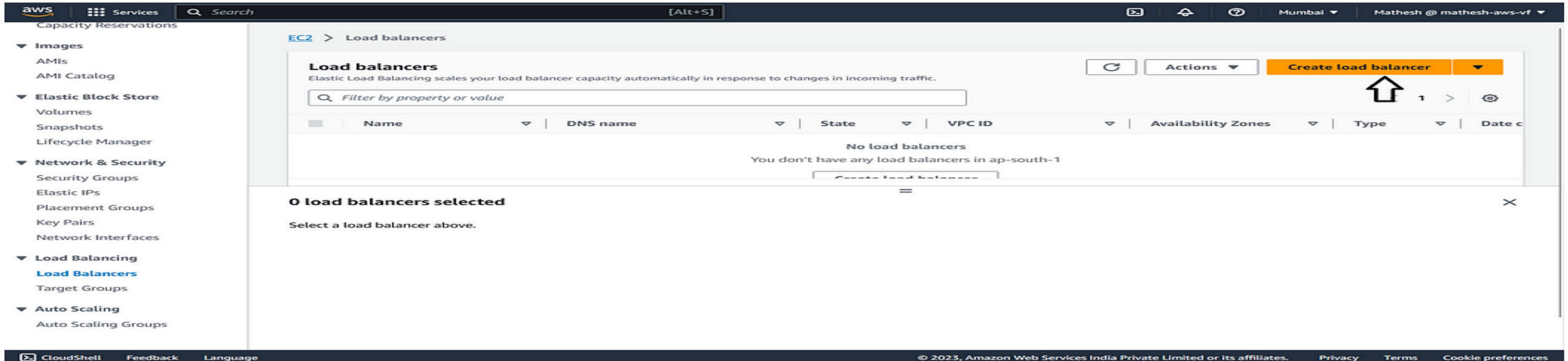
We intentionally deployed the application on only one instance to check if the Load Balancer will distribute 50% of the traffic to one instance (which will receive a response) and 50% to another instance (which will not receive a response).

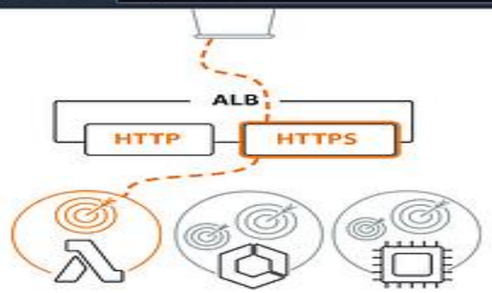
Step 4 :

Creating the Load Balancer:

Access the EC2 Terminal.

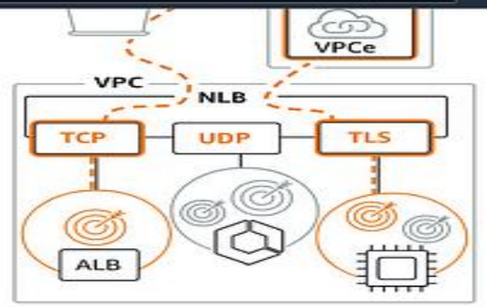
Follow the steps outlined below.





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

Create



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

Create

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

Create

▶ Classic Load Balancer - *previous generation*

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 Elastic Load Balancing works**

Basic configuration

Load balancer name

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

aws-prod-alb

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

Scheme Info

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

- (●) **Internet-facing**
  An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more
- ( ) **Internal**
  An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type Info

Select the type of IP addresses that your subnets use.

IP address type Info
Select the type of IP addresses that your subnets use.

- (•) **IPv4**
  Recommended for internal load balancers.
- ( ) **Dualstack**
  Includes IPv4 and IPv6 addresses.

Network mapping Info

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

VPC Info
Select the virtual private cloud (VPC) for your targets 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.

aws-prod-demo-vpc
vpc-0eb3f78ab19d3a68b
IPv4: 10.0.0.0/16

Choose the Private VPC , You Created

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.

- [x] **us-east-1a (use1-az4)**

Subnet
subnet-0e23e743f1cb89279 aws-prod-demo-subnet-public1-us-east-1a

IPv4 address

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 Elastic Load Balancing works**

Basic configuration

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

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

- (•) **Internet-facing**
  An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more
- ( ) **Internal**
  An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type Info
Select the type of IP addresses that your subnets use.

IP address type Info

Select the type of IP addresses that your subnets use.

- (•) **IPv4**
  Recommended for internal load balancers.
- ( ) **Dualstack**
  Includes IPv4 and IPv6 addresses.

Network mapping Info

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

VPC Info

Select the virtual private cloud (VPC) for your targets 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.

aws-prod-demo-vpc
vpc-0eb3f78ab19d3a68b
IPv4: 10.0.0.0/16

Choose the Private VPC , You Created

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-az4)**

Subnet

subnet-0e23e743f1cb89279 aws-prod-demo-subnet-public1-us-east-1a

IPv4 address

aws | Services | Search [Alt+S] | N. Virginia | Mathesh @ mathesh-aws-vf

the load balancer is created. To confirm the VPC for your targets, view your target groups.

aws-prod-demo-vpc
vpc-0eb3f78ab19d3a68b
IPv4: 10.0.0.0/16

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-az4)**

Subnet

subnet-0e23e743f1cb89279 aws-prod-demo-subnet-public1-us-east-1a

IPv4 address
Assigned by AWS

Choose Public Subnets

☑ **us-east-1b (use1-az6)**

Subnet

subnet-0f71bb5ec2ed1e896 aws-prod-demo-subnet-public2-us-east-1b

IPv4 address
Assigned by AWS

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

aws-prod-sg
sg-0307215e2b24ec234 VPC: vpc-0eb3f78ab19d3a68b

Select a Security Group With SSH access Port 22 , HTTP access Port 80 , Custom TCP access Port 8000 or Create a new Security group with the mentioned access .

Listeners and routing Info

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 Remove

Protocol: HTTP | Port: 80 (1-65535)

Default action Info: Forward to — Select a target group

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.

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.

Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

aws-prod-tg

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

Protocol HTTP : **Port** 8000

Define Port 8000

1-65535

VPC

Select the VPC with the instances that you want to include in the target group.

aws-prod-demo-vpc
vpc-0eb3f78ab19d3a68b
IPv4: 10.0.0.0/16

Protocol version

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

aws | Services | Search [Alt+S] | N. Virginia | Mathesh @ mathesh-aws-vf

Health check protocol

HTTP

Health check path

Use the default path of "/" to ping 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.

Cancel Next

CloudShell | Feedback | Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy | Terms | Cookie preferences

Step 1
Specify group details

Step 2
Register targets

Register targets

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

Available instances (2/3)

Filter resources by property or value

1

	Instance ID	Name	State	Security groups	Zone
☑	i-0b17dd1e4ebeda4ef		Running	aws-prod-sg	us-east-1a
☑	i-0e43baf8c424ba653		Running	aws-prod-sg	us-east-1b
☐	i-0f9bf711b9cc2da54	Bastion_host	Running	launch-wizard-1	us-east-1b

2 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.
8000
1-65535 (separate multiple ports with commas)

Include as pending below

Select two private instances and Click Include as pending below

aws | Services | Search [Alt+S] | N. Virginia | Mathesh @ mathesh-aws-vf

Ports for the selected instances
Ports for routing traffic to the selected instances.
8000
1-65535 (separate multiple ports with commas)

Include as pending below

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

Review targets

Targets (2)

Show only pending | Remove all pending

Filter resources by property or value

1

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone	Subnet ID
✕	Pending	i-0b17dd1e4ebeda4ef		8000	Running	aws-prod-sg	us-east-1a	subnet-0486c3d23
✕	Pending	i-0e43baf8c424ba653		8000	Running	aws-prod-sg	us-east-1b	subnet-0cebb9290

2 pending

Cancel | Previous | **Create target group**

CloudShell | Feedback | Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy | Terms | Cookie preferences

Listener HTTP:80 Remove

Protocol	Port	Default action Info
HTTP	80 (1-65535)	Forward to aws-prod-tg, Target type: Instance, IPv4, HTTP

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.

Add listener tag

You can add up to 50 more tags.

Add listener

Add-on services - *optional*

Additional AWS services can be integrated with this load balancer at launch. You can also add these and other services after your load balancer is created by reviewing the "Integrated Services" tab for the selected load balancer.

AWS Global Accelerator Info

☐ Create an accelerator to get static IP addresses and improve the performance and availability of your applications. Additional charges apply

Summary

Review and confirm your configurations. Estimate cost

Basic configuration Edit

aws-prod-alb

- Internet-facing
- IPv4

Security groups Edit

- aws-prod-sg
  sg-0307215e2b24ec234

Network mapping Edit

VPC vpc-0eb3f78ab19d3a68b
aws-prod-demo-vpc

- us-east-1a
  subnet-0e23e743f1cb89279
  aws-prod-demo-subnet-public1-us-east-1a
- us-east-1b
  subnet-0f71bb5ec2ed1e896
  aws-prod-demo-subnet-public2-us-east-1b

Listeners and routing Edit

- HTTP:80 defaults to aws-prod-tg

Add-on services Edit

None

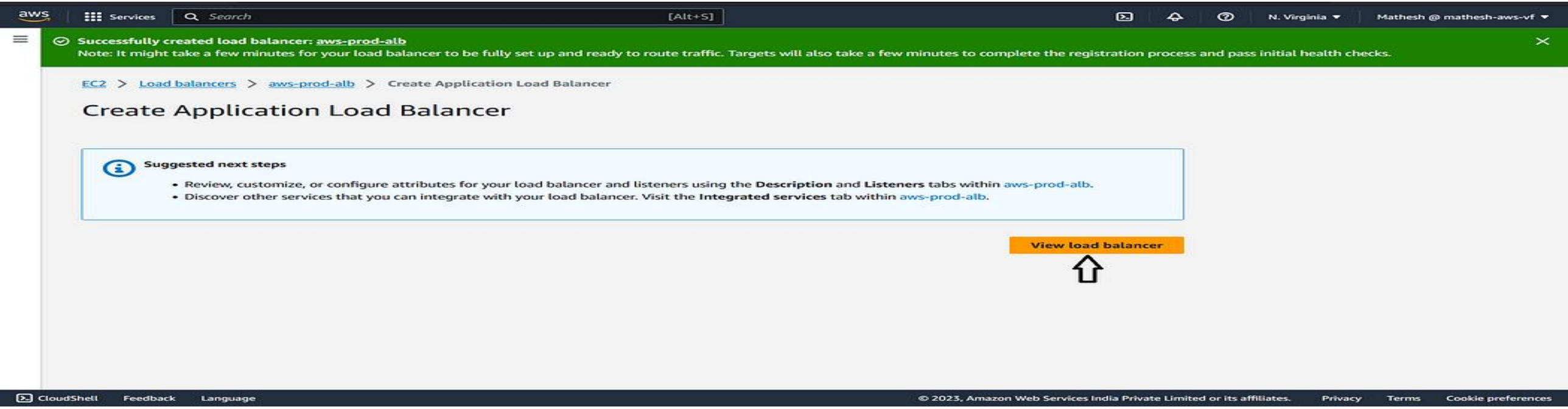
Tags Edit

None

Attributes

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

Cancel **Create load balancer**



EC2 > Load balancers > project-LB

project-LB

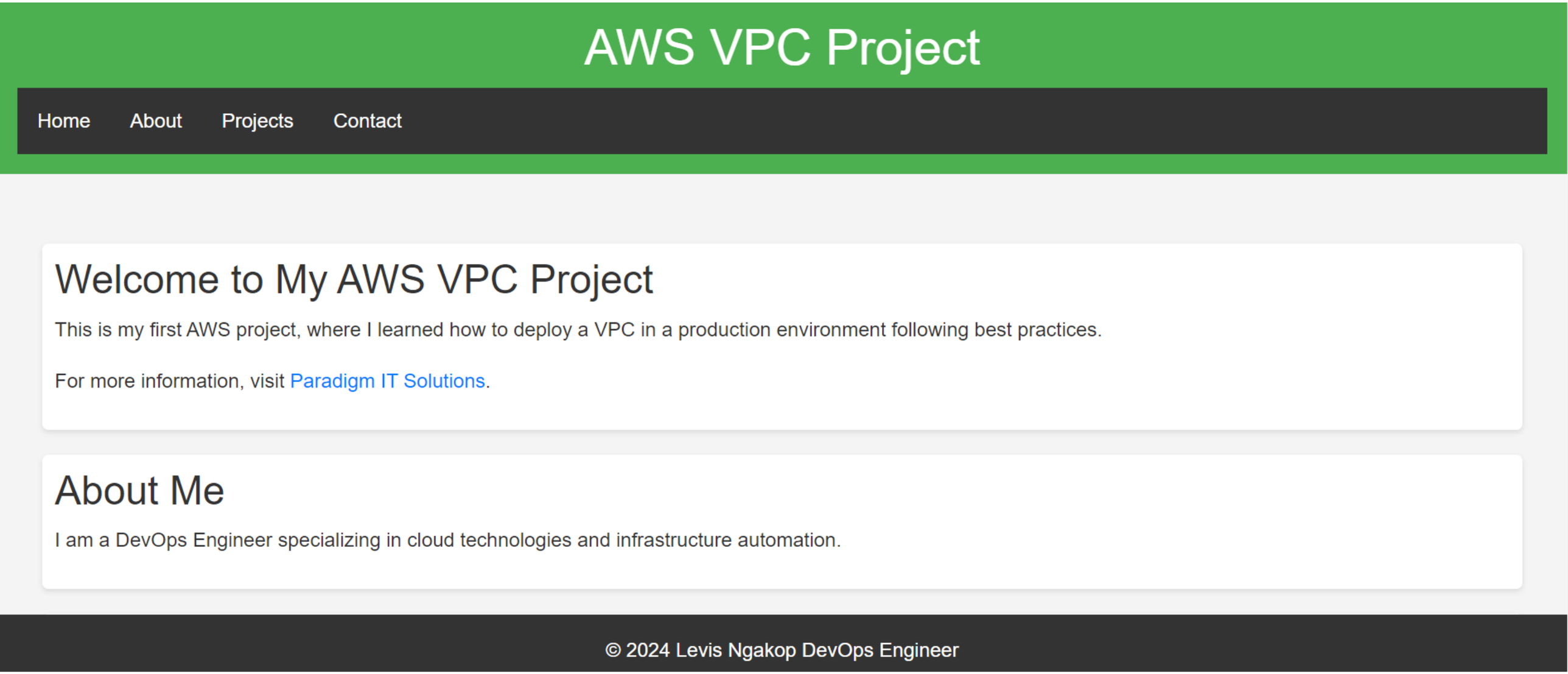
Actions

Details

Load balancer type Application	Status Active	VPC vpc-0786bb49ab4efee67	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z35SXDOTRQ7X7K	Availability Zones subnet-031b574ab2c0b6d9c us-east-1b (use1-az4) subnet-0ac2777cfdc4b58e5 us-east-1a (use1-az2)	Date created August 10, 2024, 17:21 (UTC+02:00)

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:339713075207:loadbalancer/app/project-LB/447465cf67b0ed47

DNS name Info
project-LB-864460842.us-east-1.elb.amazonaws.com (A Record)



Now We Successfully deployed Application securely in Private instance, we can access it through Internet using Load Balancer Securely.