

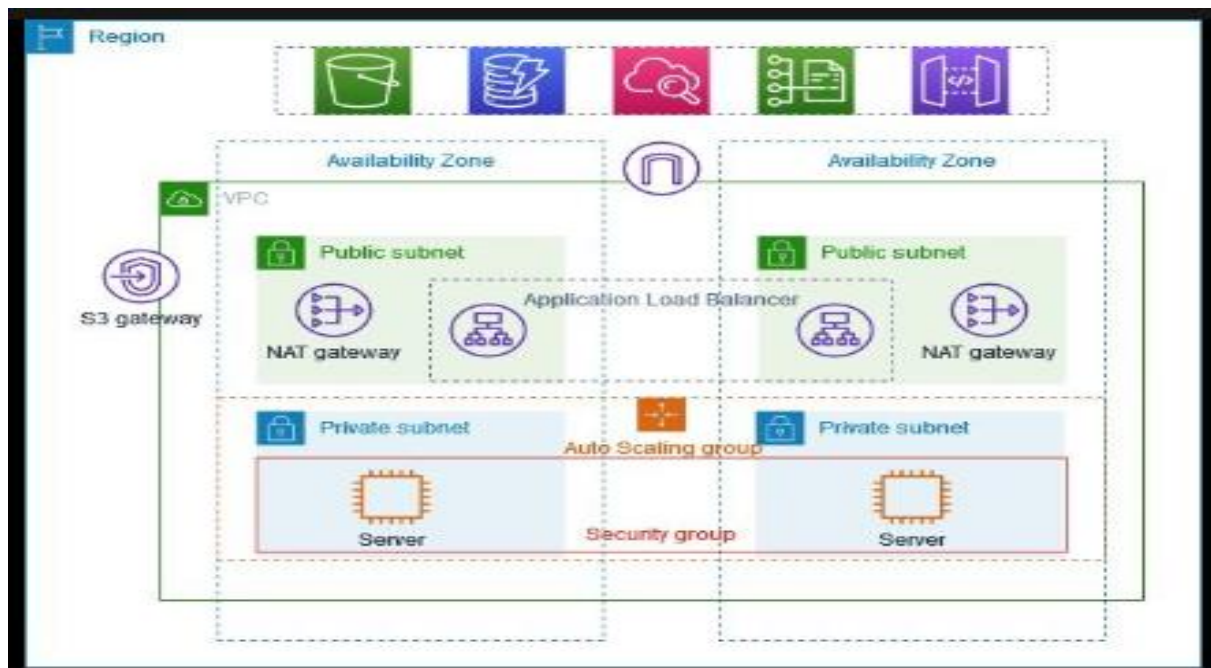
VPC With Public-Private Subnets in Production

About the Project:

This example demonstrates how to create a VPC that you can use for servers in a production environment. To improve resiliency, you deploy the servers in two Availability Zones, by using an Auto Scaling group and an Application Load Balancer. For additional security, you deploy the servers in private subnets. The servers receive requests through the load balancer. The servers can connect to the internet by using a NAT gateway. To improve resiliency, you deploy the NAT gateway in both Availability Zones.

Overview:

The VPC has public subnets and private subnets in two Availability Zones. Each public subnet contains a NAT gateway and a load balancer node. The servers run in the private subnets, are launched and terminated by using an Auto Scaling group, and receive traffic from the load balancer. The servers can connect to the internet by using the NAT gateway.



- Deployed servers in two Availability Zones using Auto Scaling group and Application Load Balancer for high availability.
- Configured VPC with 2 AZs, public/private subnets & Bastion host for secure server access.
- Enabled outbound internet connectivity via NAT Gateways for IP masking.
- Implemented Application Load Balancer for optimized performance & user experience.
- Ran application securely within VPC, ensuring compliance & protection.

First, Create the VPC -name it(vpc-project) and in NAT gateways select the 1per AZ

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
vpc-project

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
CIDR block size must be between /16 and /28.

IPv6 CIDR block: [info](#)
☒ No IPv6 CIDR block
☐ Amazon-provided IPv6 CIDR block

Tenancy: [info](#)
Default

Number of Availability Zones (AZs): [info](#)
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.
1 2 3
▶ Customize AZs

Number of public subnets: [info](#)
The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.
0 2

Number of private subnets: [info](#)
The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.
0 2 4
▶ Customize subnets CIDR blocks

NAT gateways (1): [info](#)
Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway.
None In 1 AZ 1 per AZ

VPC endpoints: [info](#)
Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.
None 53 Gateway

DNS options: [info](#)
☒ Enable DNS hostnames
☒ Enable DNS resolution

Preview

VPC [Show details](#)
Your AWS virtual network

Subnets (4)
Subnets within this VPC

Route tables (3)
Route network traffic to resources

Network connections (4)
Connections to other networks

After click on the create it create the four subnets. Two are public and two are private subnets.

Virtual private cloud

EC2 Global View [EC2](#)

Filter by VPC: [Select a VPC](#)

Virtual private cloud

Your VPCs

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

CloudShell Feedback

VPC > Your VPCs > vpc-0539a5cdf7c2dae51

vpc-0539a5cdf7c2dae51 / vpc-project-vpc

Details [Info](#)

VPC ID: vpc-0539a5cdf7c2dae51

State: Available

DNS hostnames: Enabled

DNS resolution: Enabled

Tenancy: Default

DHCP option set: dopt-0e39e4a4e75ae730b

Main route table: rtb-0d8c0857695b2869

Main network ACL: acl-077386b276ad5d4b6

Default VPC: No

IPv4 CIDR: 10.0.0.0/16

IPv6 pool: -

IPv6 CIDR (Network border group): -

Network Address Usage metrics: Disabled

Route 53 Resolver DNS Firewall rule groups: -

Owner ID: 952047986519

Resource map [Info](#)

Resource map

VPC [Show details](#)
Your AWS virtual network

Subnets (4)
Subnets within this VPC

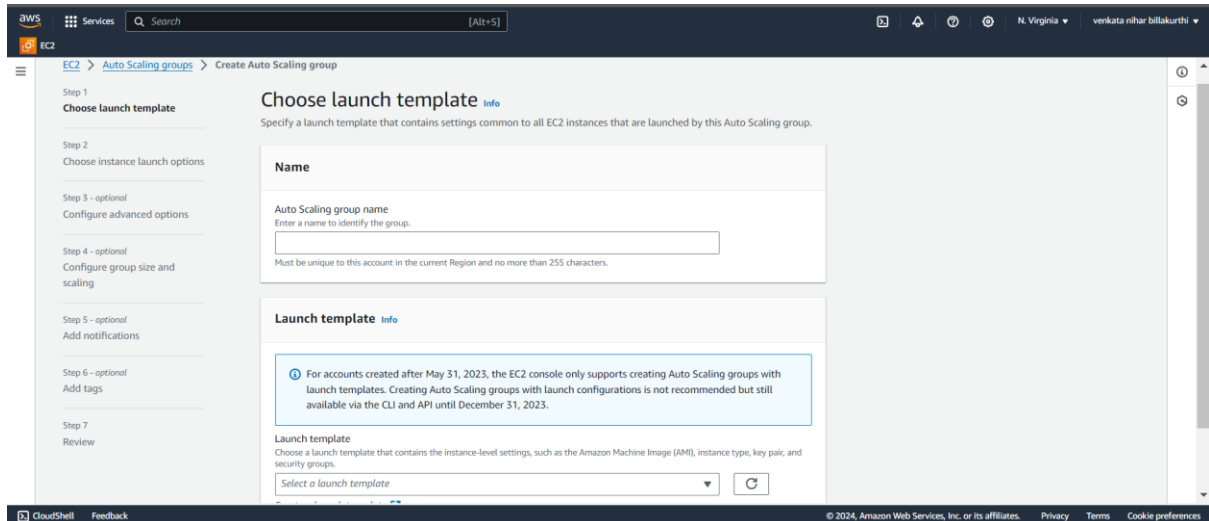
Route tables (4)
Route network traffic to resources

Network connections (4)
Connections to other networks

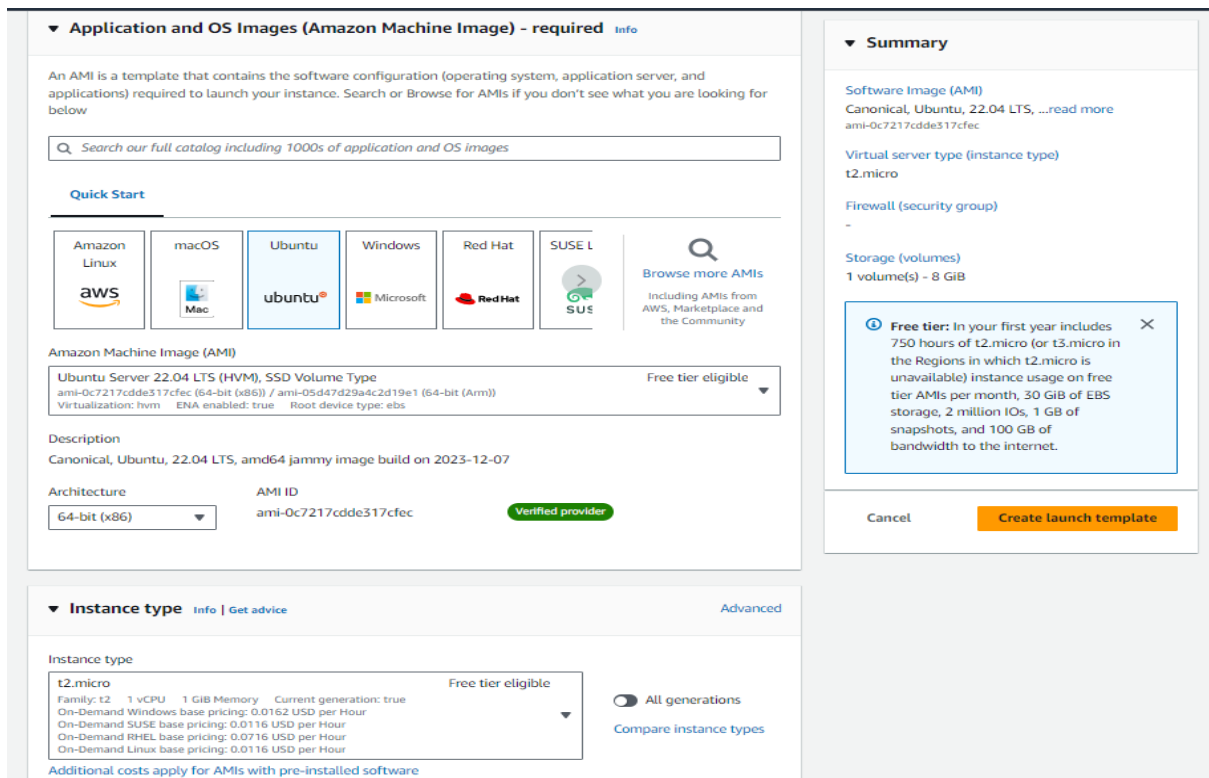
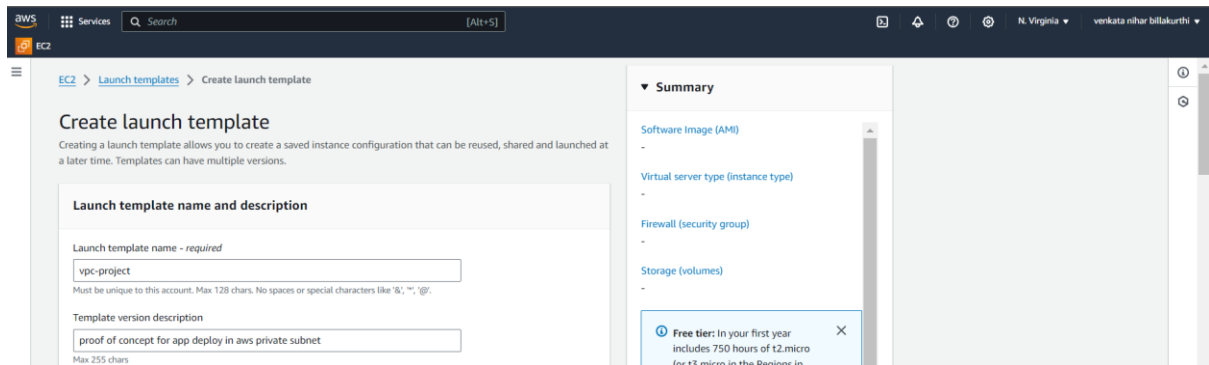
© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Here, vpc is created successfully.

After, then create the Auto Scaling groups – name them(VPC-project) and select our lanuch template.



So create the launch template name it(vpc-project) and description (proof of concept for app deployment in aws private subnets) the name and description is our choice.



Add the custom TCP port:8000 in security group

▼ 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

vpc-project

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 [Info](#)

vpc-0539a5cd7c2dae51 (vpc-project-vpc)

7b0.0.0.0/16

Inbound Security Group Rules

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

Type [Info](#)

ssh

Protocol [Info](#)

TCP

Port range [Info](#)

22

Source type [Info](#)

Anywhere

Source [Info](#)

0.0.0.0/0

Description - optional [Info](#)

e.g. SSH for admin desktop

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

Type [Info](#)

Custom TCP

Protocol [Info](#)

TCP

Port range [Info](#)

8000

Source type [Info](#)

Anywhere

Source [Info](#)

0.0.0.0/0

Description - optional [Info](#)

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.

[Advanced network configuration](#)

▼ Storage (volumes) [Info](#)

EBS Volumes [Hide details](#)

► Volume 1 (AMI Root) (8 GiB, EBS, General purpose SSD (gp2))

▼ Summary

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more

ami-0c7217cd8e517efec

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 I/Os, 1 GiB of snapshots, and 100 GB of bandwidth to the Internet.

Cancel

Then click on the create.

Services [Alt+S]

EC2

EC2 Dashboard EC2 Global View Events Console-to-Code [Preview](#) ▼ Instances Instances Instance Types **Launch Templates** Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations [New](#) ▼ Images AMIs AMI Catalog ▼ Elastic Block Store Volumes

Launch Templates (1) [Info](#)

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
lt-0c58acbe381204fac	vpc-project	1	1	2024-01-18T14:27:30.000Z	arnaws:iam::952047986519:root

Select a launch template

CloudShell [Feedback](#)

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Here the launch template is created.

Now, select our launch template in auto scaling

The screenshot shows the 'Launch template' step in the AWS Management Console. The left sidebar lists steps: 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), and Step 7 (Review). The main content area has a 'Name' section with a text input 'vpc-project' and a 'Launch template' section with a dropdown menu also set to 'vpc-project'. A blue information box states: '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.' At the bottom, there are buttons for 'CloudShell', 'Feedback', and a footer with copyright information.

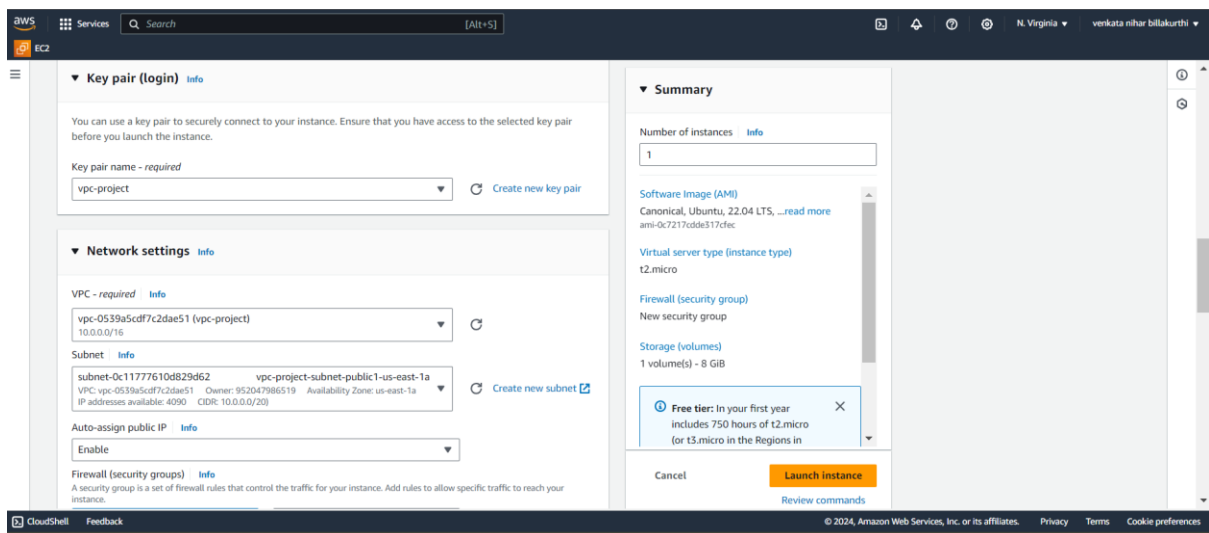
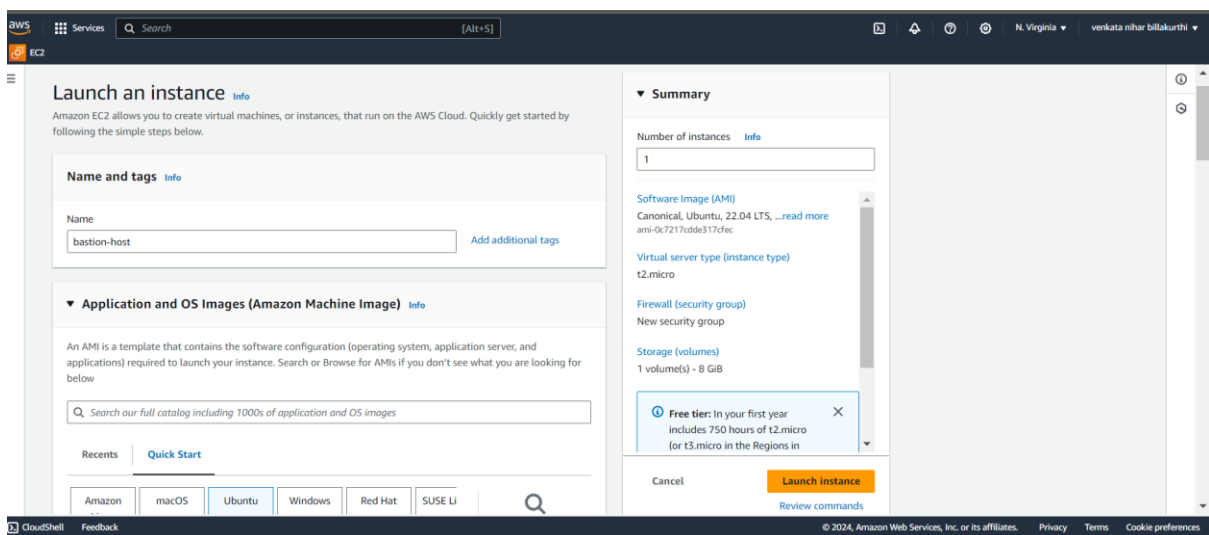
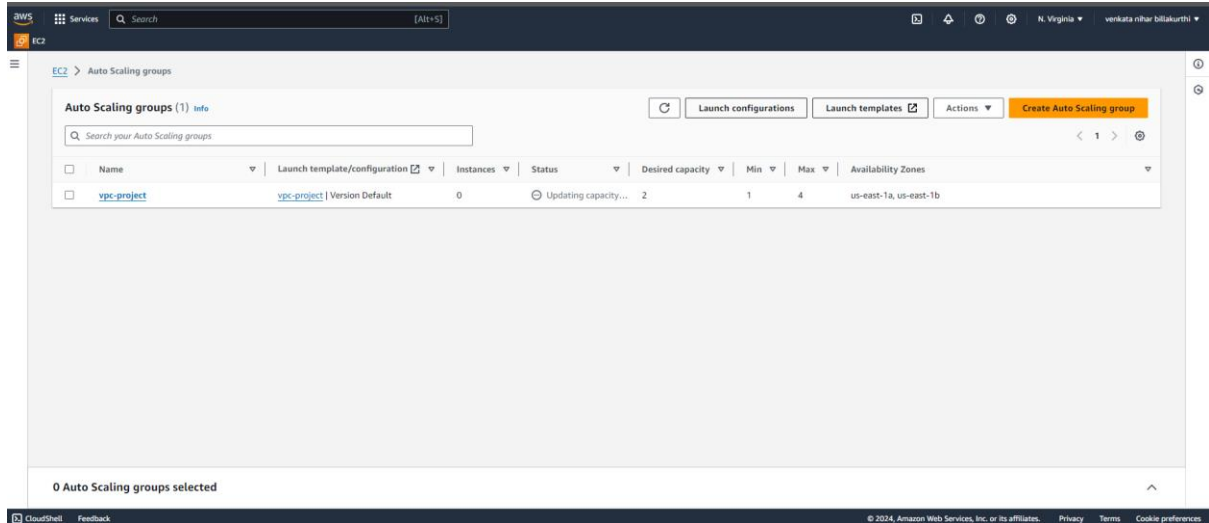
And select out vpc and private subnets also.

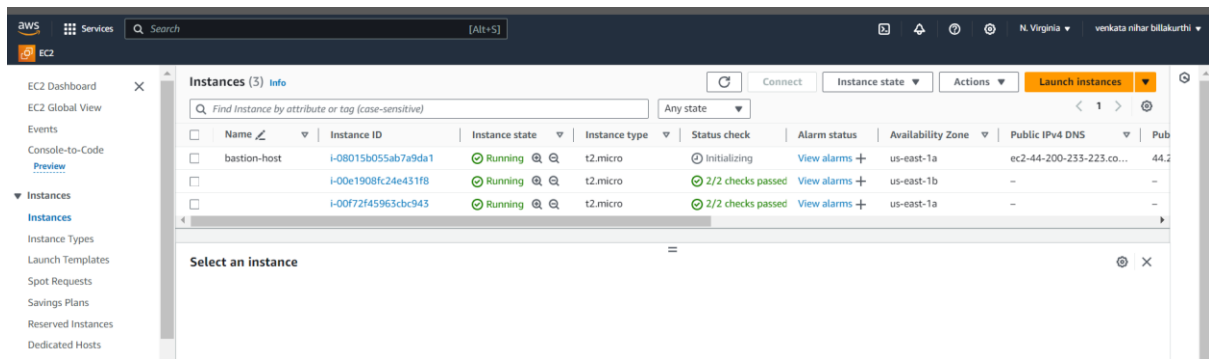
The screenshot shows the 'Network' step in the AWS Management Console. The left sidebar shows Step 6 (optional, Add tags) and Step 7 (Review). The main content area has a 'VPC' section with a dropdown menu showing 'vpc-0539a5cdf7c2dae51 (vpc-project-vpc)' and an 'Availability Zones and subnets' section with a dropdown menu showing 'us-east-1a | subnet-09fc80c9dfb0dbe37 (vpc-project-subnet-private1-us-east-1a)'. There are also buttons for 'Cancel', 'Skip to review', 'Previous', and 'Next'. The footer includes 'CloudShell', 'Feedback', and copyright information.

In the configure group size and scaling, Desired capacity is 2 and max is 4 this our choices.

The screenshot shows the 'Configure group size and scaling' step in the AWS Management Console. The left sidebar lists steps: 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), and Step 7 (Review). The main content area has a 'Group size' section with a 'Desired capacity type' dropdown set to 'Units (number of instances)' and a 'Desired capacity' input field set to '2'. There is also a 'Scaling' section with 'Min desired capacity' set to '1' and 'Max desired capacity' set to '4'. The footer includes 'CloudShell', 'Feedback', and copyright information.

Here, the auto scaling group and two aws instances also created.





```
PS C:\Users\hp> cd downloads
PS C:\Users\hp\downloads> scp -i /Users/hp/Downloads/vpc-project.pem /Users/hp/Downloads/vpc-project.pem ubuntu@44.200.233.223:/home/ubuntu
vpc-project.pem 100% 1674 6.0KB/s 00:00
```

```
ubuntu@ip-10-0-8-134: ~
PS C:\Users\hp\downloads> ssh -i vpc-project.pem ubuntu@44.200.233.223
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Thu Jan 25 05:45:22 UTC 2024

System load:  0.0          Processes:      98
Usage of /:   20.8% of 7.57GB Users logged in:  0
Memory usage: 21%         IPv4 address for eth0: 10.0.8.134
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-8-134:~$ ls
vpc-project.pem
ubuntu@ip-10-0-8-134:~$
```

Now, we try to login in private instances.

Then, copy the private IP Address of a one instances

➔ Ssh -i (passkey file name) ubuntu@privateIP

Then, we can able to login to the private instances.

Now, create a file

➔ Vim index.html

➔ Enter the sample code and save it

```
<!DOCTYPE html>
<html>
<body>
<h1>My First AWS PROJECT to demonstrate apps in private subnet</h1>
</body>
</html>
```



Python3 -m http.server 8000

It is running now.

Now, go to EC2

- > Load Balancers
- > Application I.b
- > Select the our Vpc
- > Select the both subnets
- > Select the our security group
- > Listeners & routing – create target group -name it and port 8000,next -select the private instances -click on include as pending below -create.
- > Select the our target group
- > create.

Then go to security group -> edit inbound rules -> HTTP/anywhere ->create

Go to load balancer >>>copy the (DNS) and paste it on browser.

