**TASK ON EC2**

1) Launch one ec2 using Amazon Linux 2 image and add script in user data to install Apache.

Launch an EC2 Instance

- Sign in to the AWS Management Console.

- Navigate to EC2 Dashboard and click Launch Instance.

- Choose Amazon Linux 2 AMI (Amazon Machine Image).

- Select an instance type (e.g., t2.micro for free-tier eligibility).

- Click Next: Configure Instance Details.

Add User Data Script

- Scroll down to Advanced Details.

- In the User Data section, enter the following script:

#!/bin/bash

sudo yum update -y

sudo yum install -y httpd

sudo systemctl start httpd.service

sudo systemctl enable httpd.service

echo "Hello Everyone, Httpd is installed using USERDATA option" > /var/www/html/index.html

- - This script:

- Updates the package manager.

- Installs Apache (httpd).

- Starts and enables the Apache service.

- Creates a simple index.html

Configure Security Group

- In the Security Group settings, allow:

- Port 22 (SSH) for remote access.

- Port 80 (HTTP) for web traffic.

- Port 443 (HTTPS) if needed.

Launch and Verify

- Click Review and Launch.

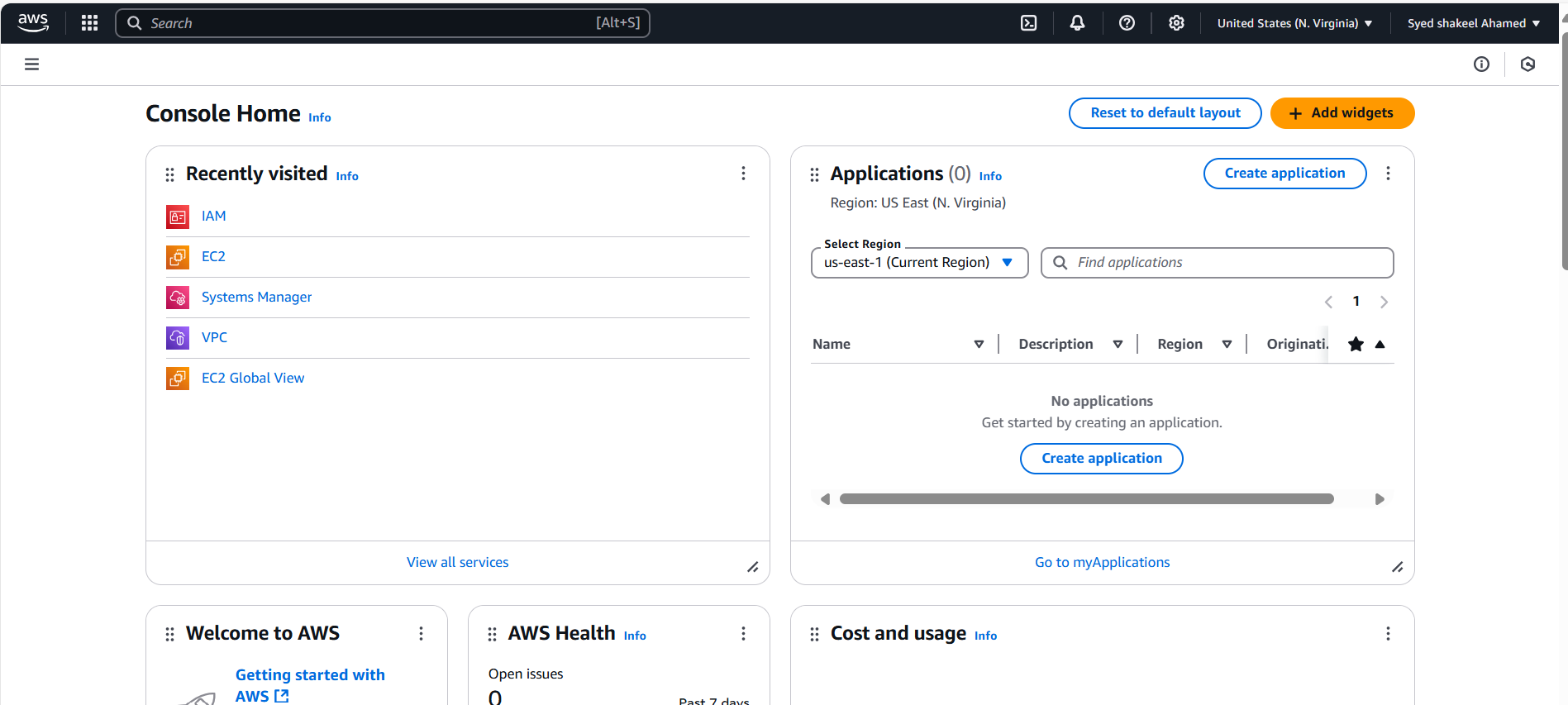
- Select or create a key pair for SSH access.

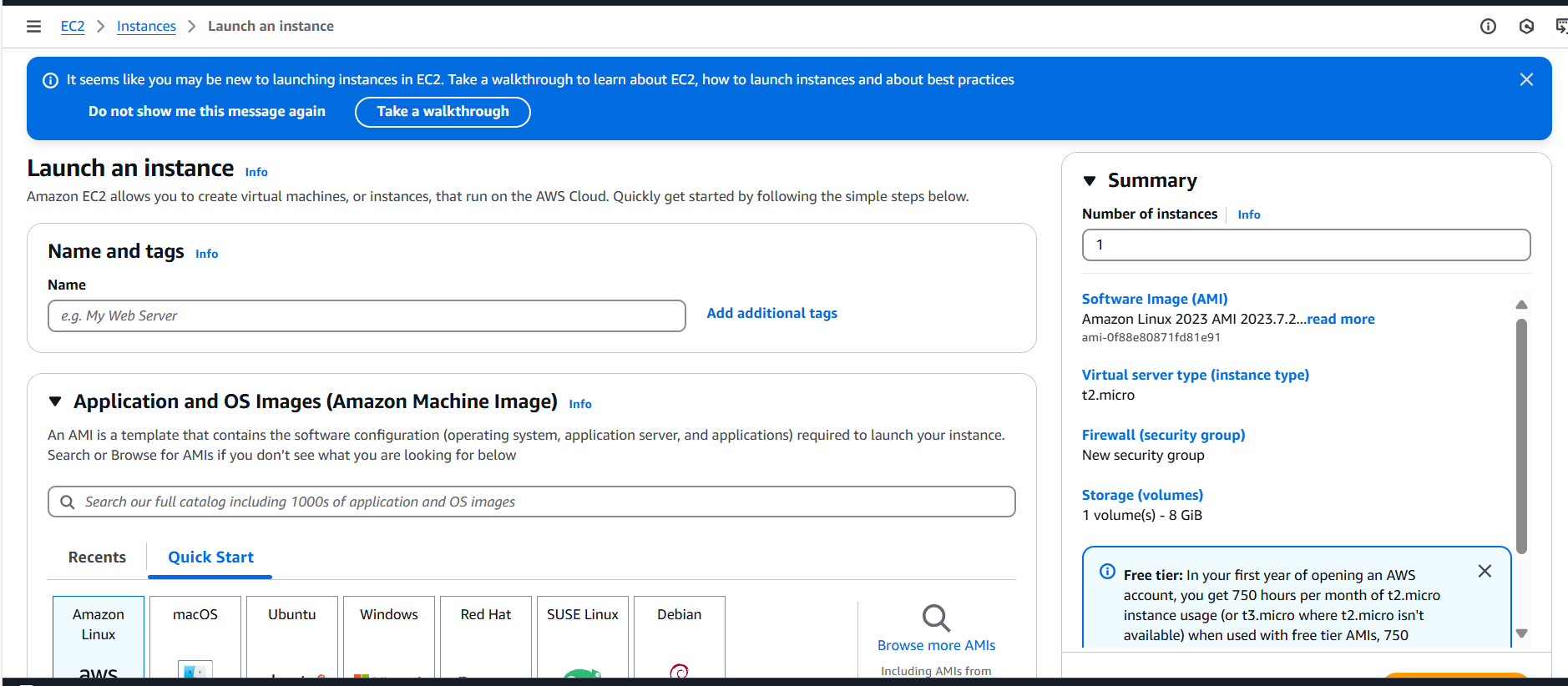
- Click Launch.

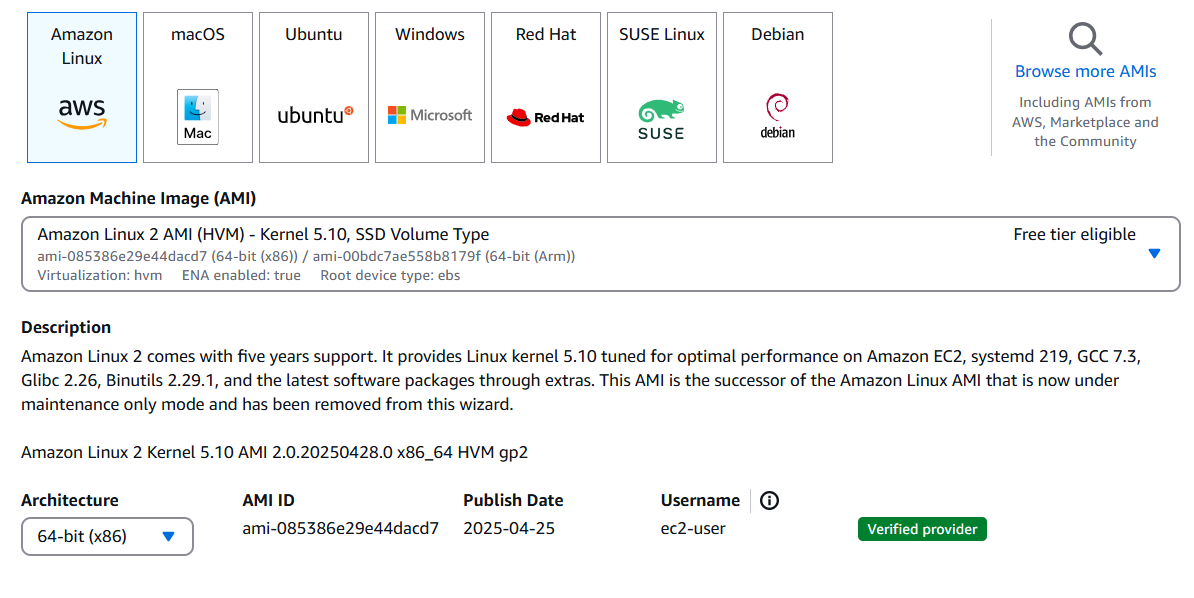
- Once the instance is running, verify Apache by visiting:

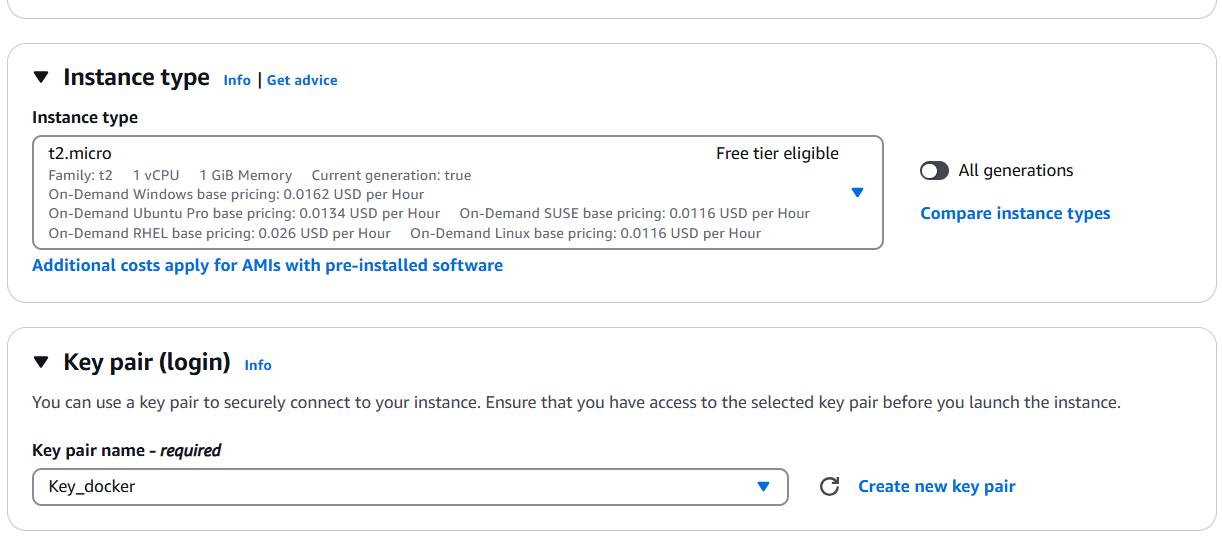
http://34.234.69.7:80>

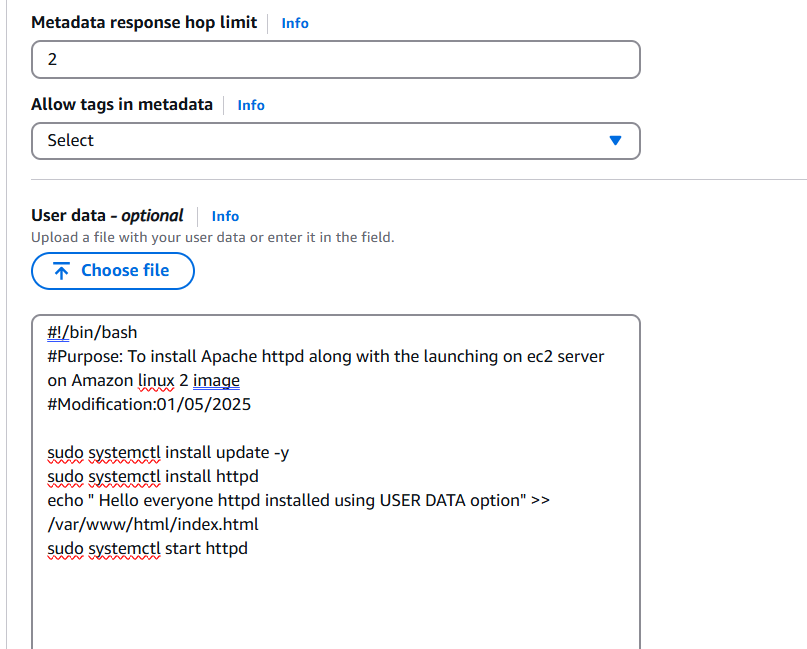
- You should see "Hello Everyone, Httpd is installed using USERDATA option" .



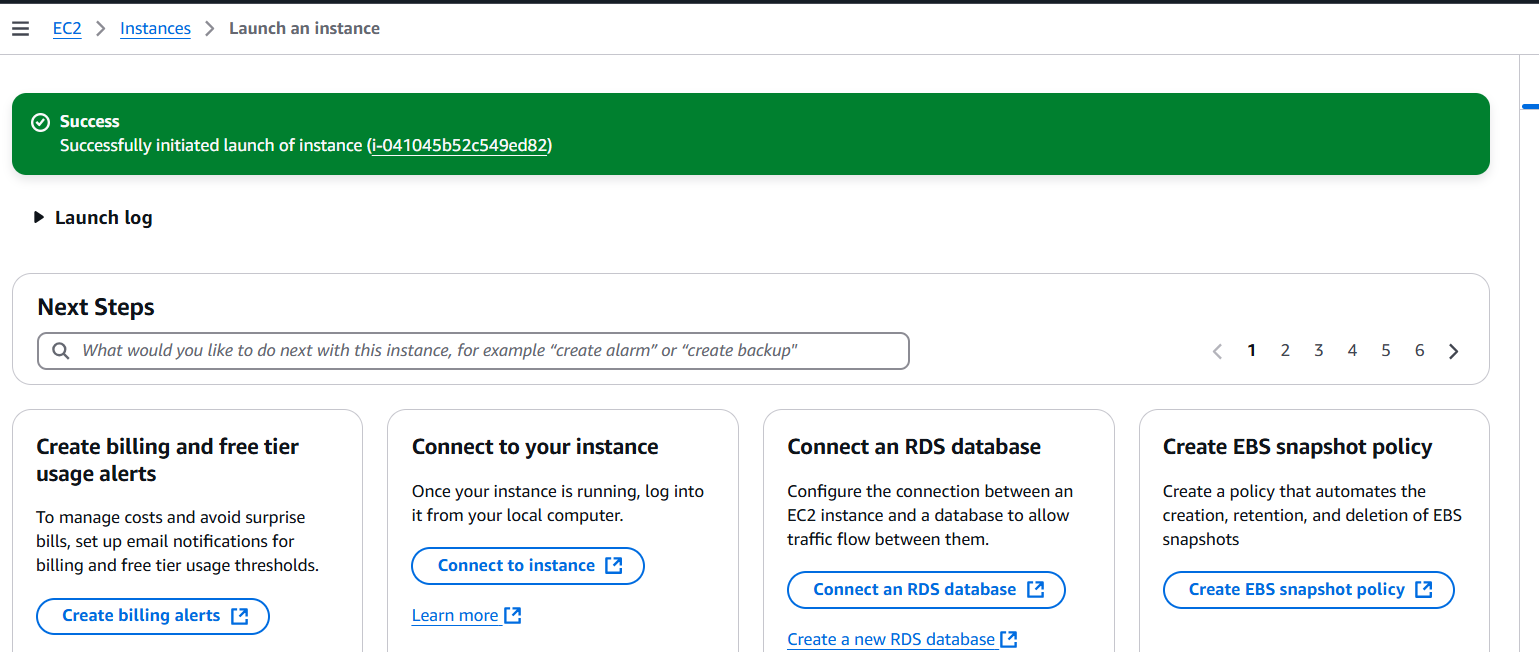


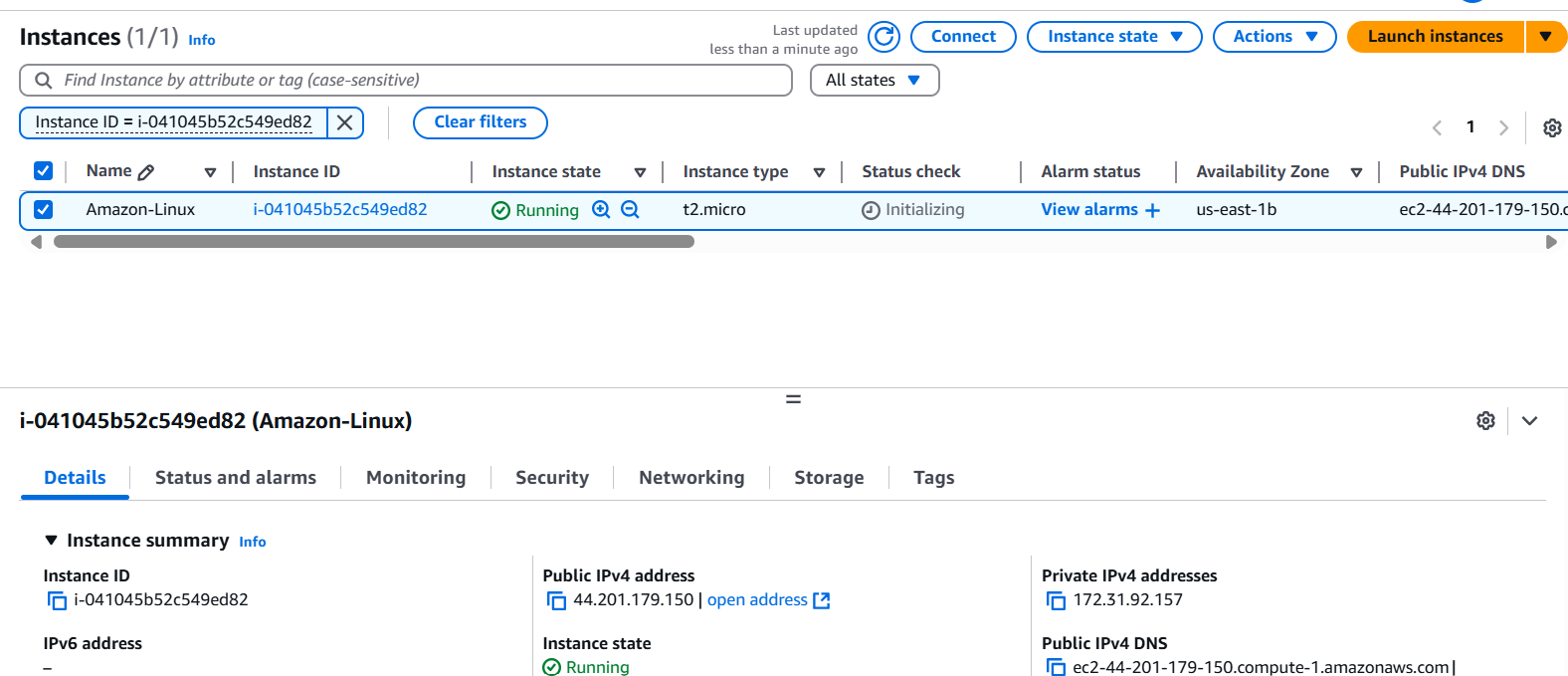


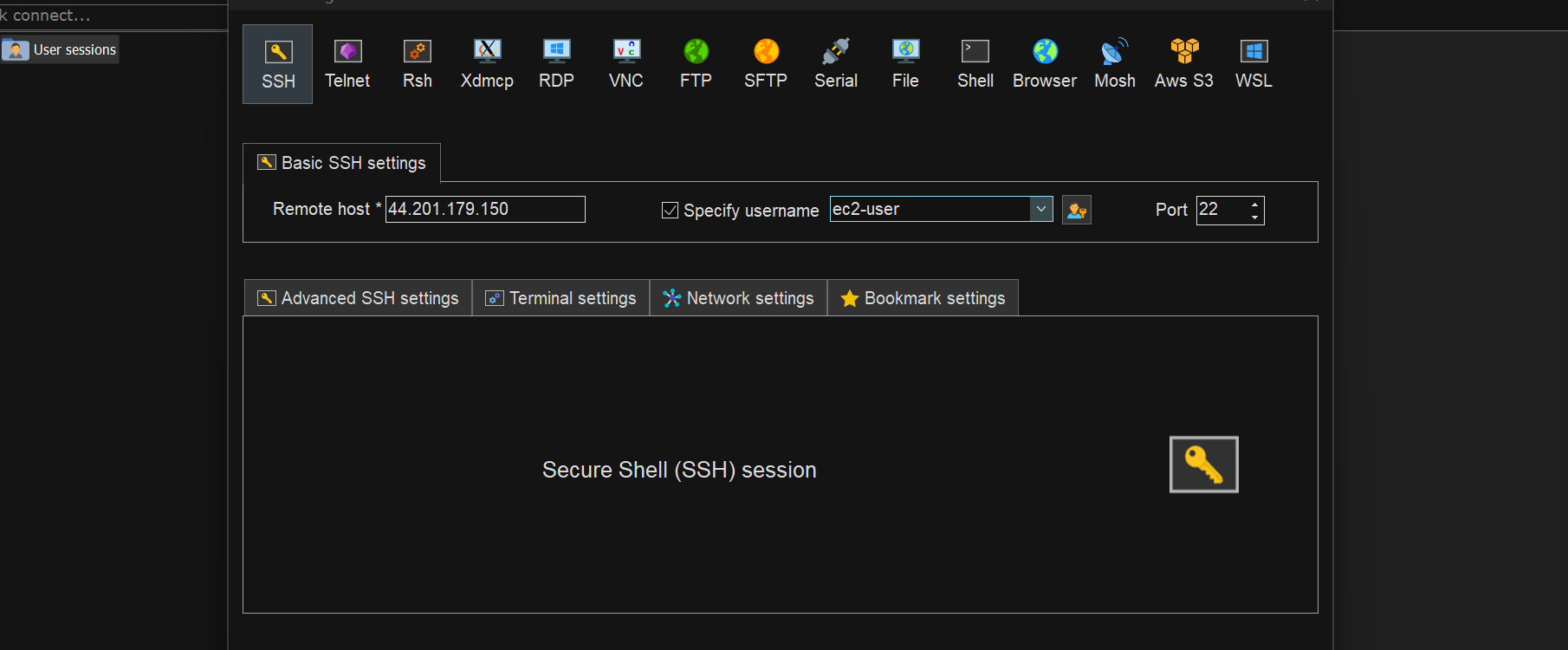
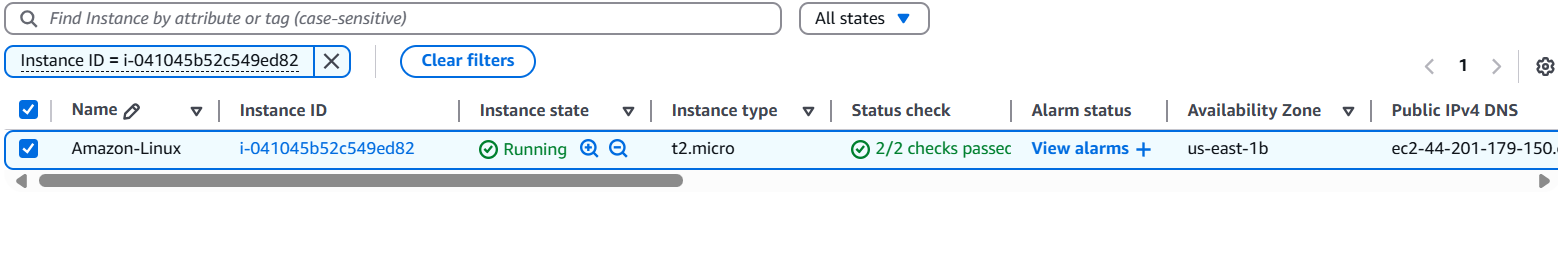


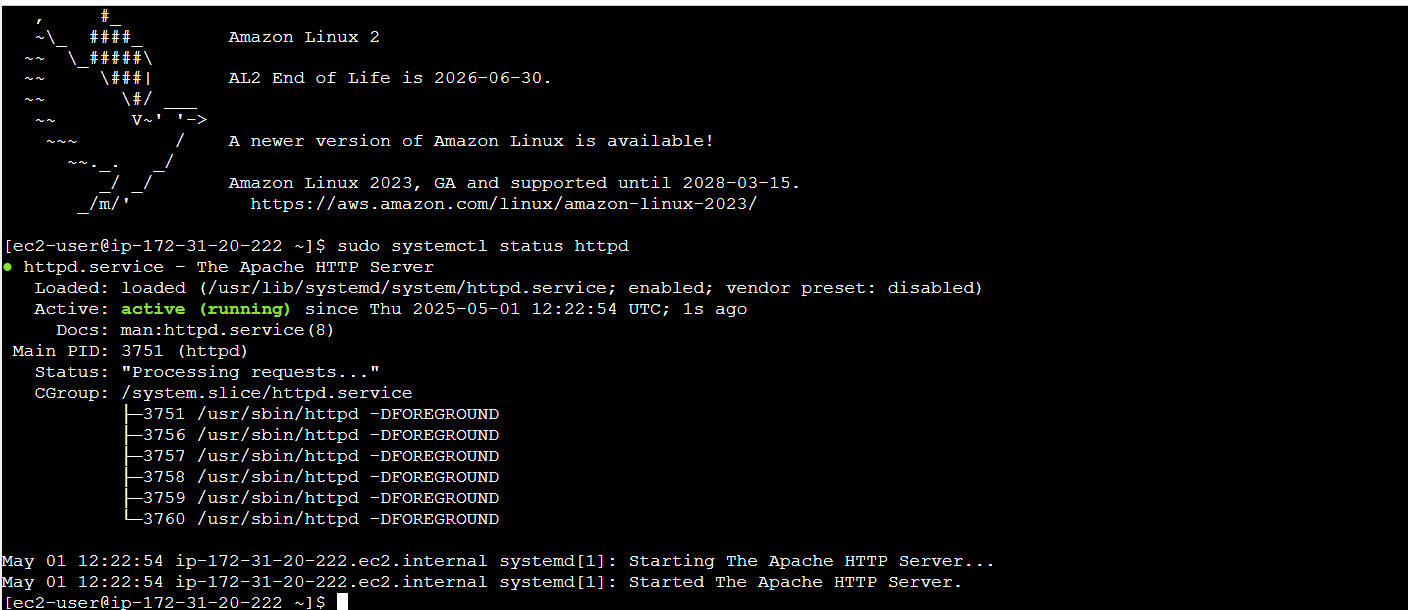














2) Launch one ec2 using Ubuntu image and add script in user data to install Nginx.

Launch an EC2 Instance

- Sign in to the AWS Management Console.

- Navigate to EC2 Dashboard and click Launch Instance.

- Choose Ubuntu AMI (Amazon Machine Image).

- Select an instance type (e.g., t2.micro for free-tier eligibility).

- Click Next: Configure Instance Details.

Add User Data Script

- Scroll down to Advanced Details.

- In the User Data section, enter the following script:

#!/bin/bash

apt update -y

apt install -y nginx

systemctl start nginx

systemctl enable nginx

echo "Welcome to Nginx " > /var/www/html/index.html

-

- This script:

- Updates the package manager.

- Installs Nginx.

- Starts and enables the Nginx service.

- Creates a simple index.html file.

Step 3: Configure Security Group

- In the Security Group settings, allow:

- Port 22 (SSH) for remote access.

- Port 80 (HTTP) for web traffic.

- Port 443 (HTTPS) if needed.

Step 4: Launch and Verify

- Click Review and Launch.

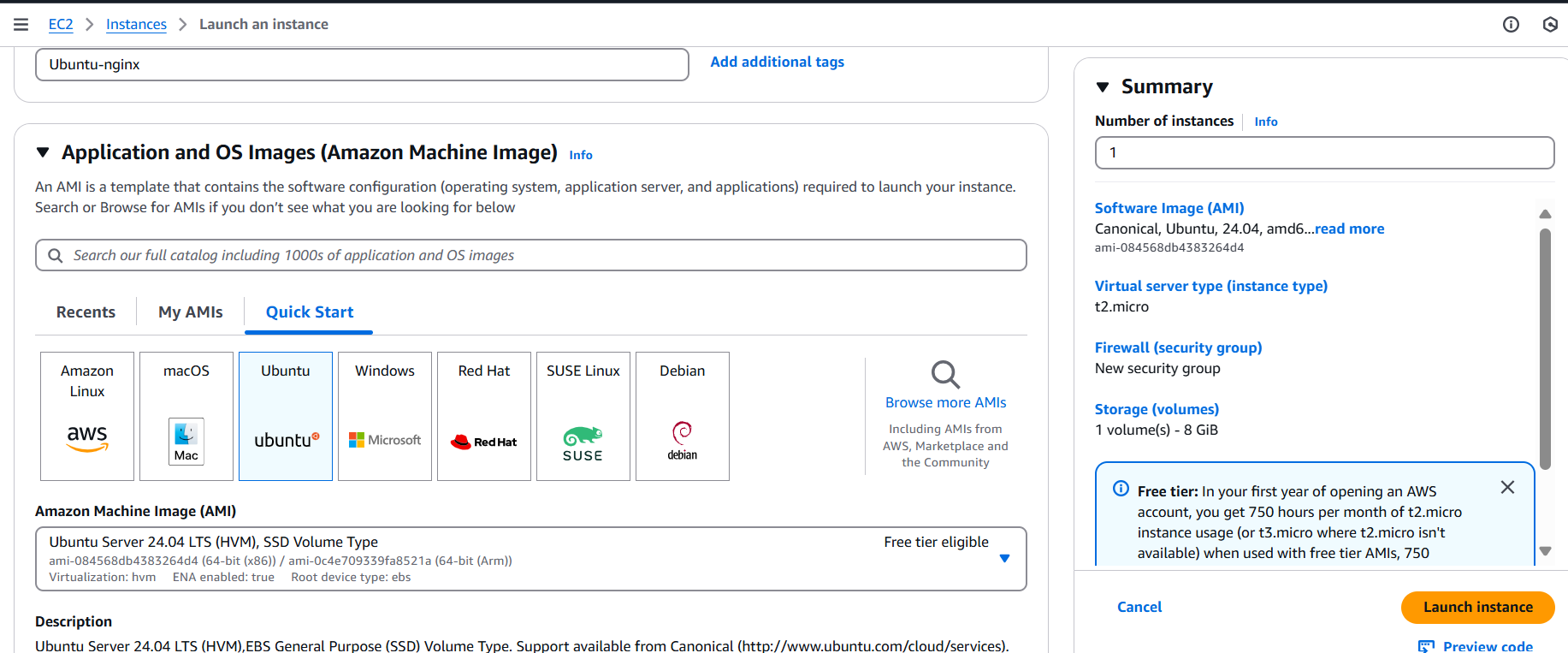
- Select or create a key pair for SSH access.

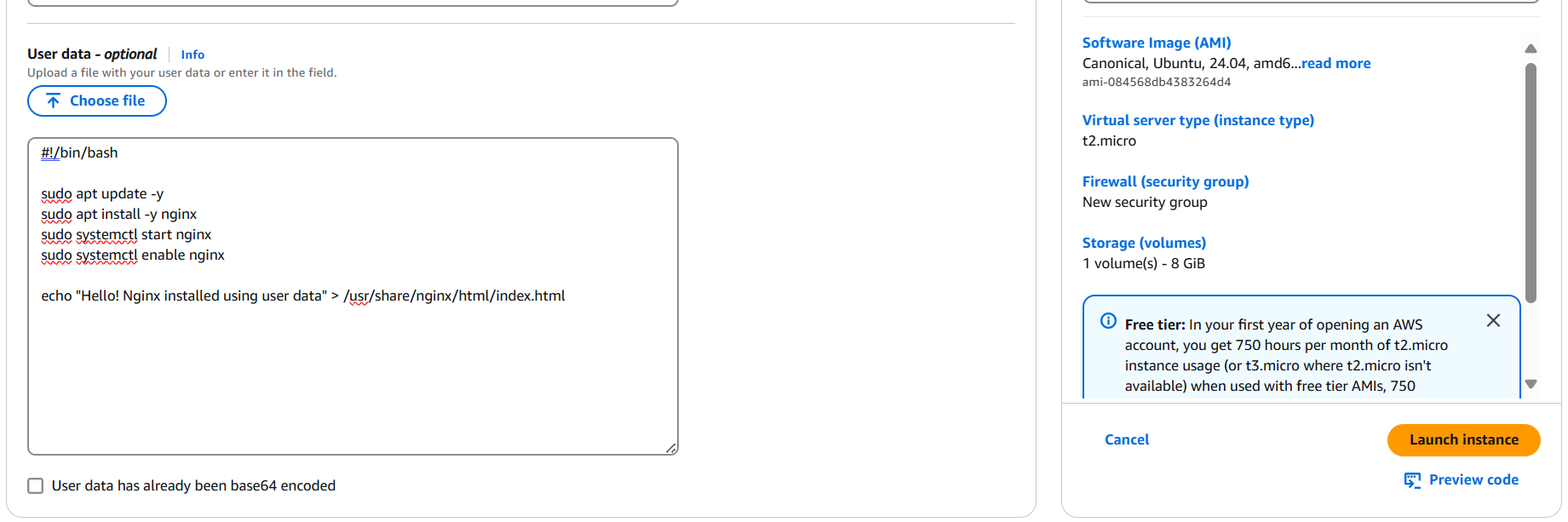
- Click Launch.

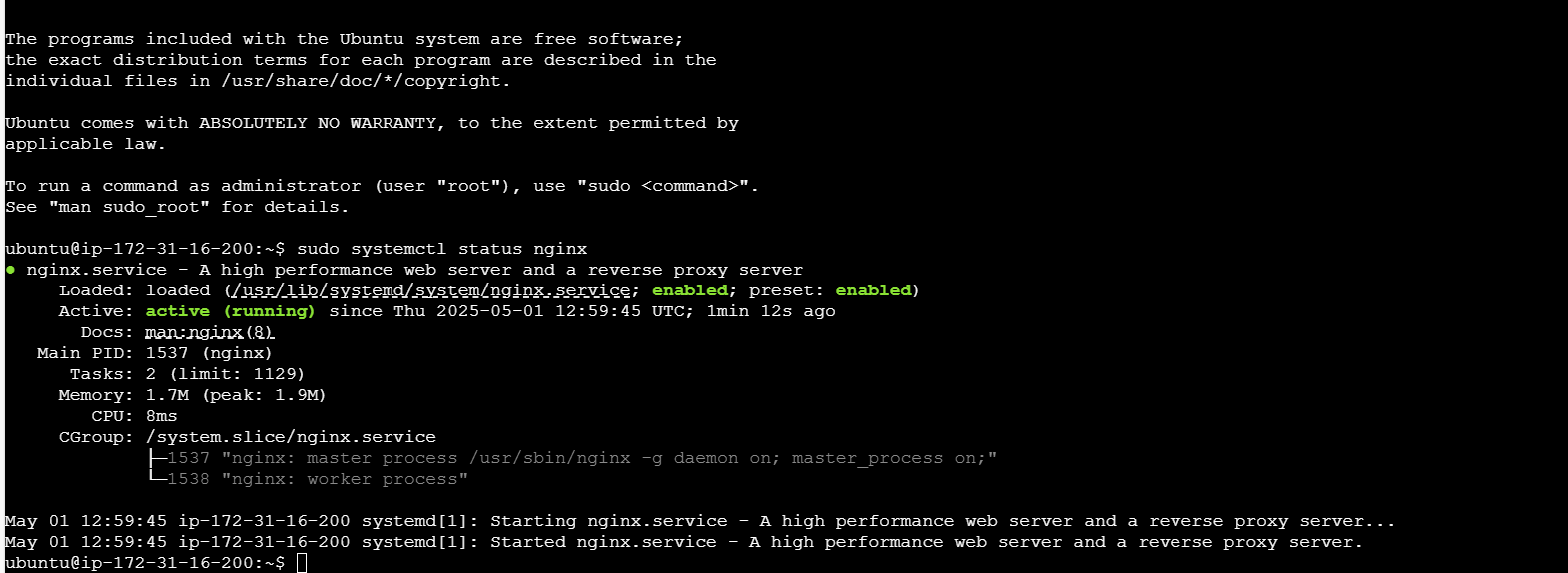
- Once the instance is running, verify Nginx by visiting:

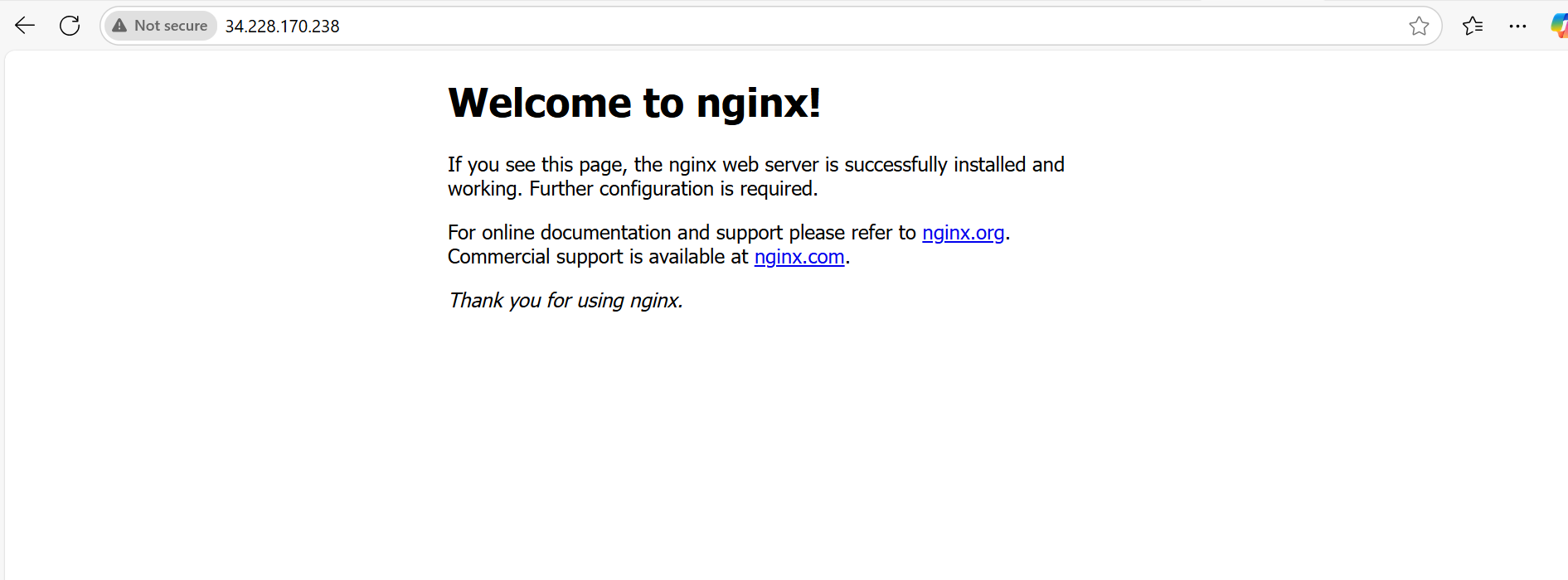
http://<34.228.170.238:80>

- You should see "Welcome to Nginx on " displayed









3) Launch one windows server and install tomcat in windows.

Launch an EC2 Windows Server Instance

- Sign in to the AWS Management Console.

- Navigate to EC2 Dashboard and click Launch Instance.

- Choose a Windows Server AMI (e.g., Windows Server 2019 or 2022).

- Select an instance type (e.g., t2.medium for better performance).

- Click Next: Configure Instance Details.

Configure Security Group

- In the Security Group settings, allow:

- Port 3389 (RDP) for remote access.

- Port 8080 (Tomcat) for web traffic.

- Port 80 (HTTP) if needed.

Connect to the Windows Server

- Once the instance is running, go to EC2 Dashboard → Instances.

- Select your instance and click Connect.

- Use RDP (Remote Desktop Protocol) to connect to the instance.

Install Java (Required for Tomcat)

- Open Microsoft Edge and download Java JDK from Oracle’s website.

- Install Java and set the JAVA\_HOME environment variable:

- Open System Properties → Advanced → Environment Variables.

- Add a new System Variable:

- Variable Name: JAVA\_HOME

- Variable Value: C:\Program Files\Java\jdk-XX.X.X (Replace with actual path).

- Click OK.

Install Apache Tomcat

- Download Apache Tomcat from Tomcat’s official website.

- Extract the downloaded ZIP file to C:\Tomcat.

- Open Command Prompt and navigate to the Tomcat bin directory:

cd C:\Tomcat\bin

startup.bat

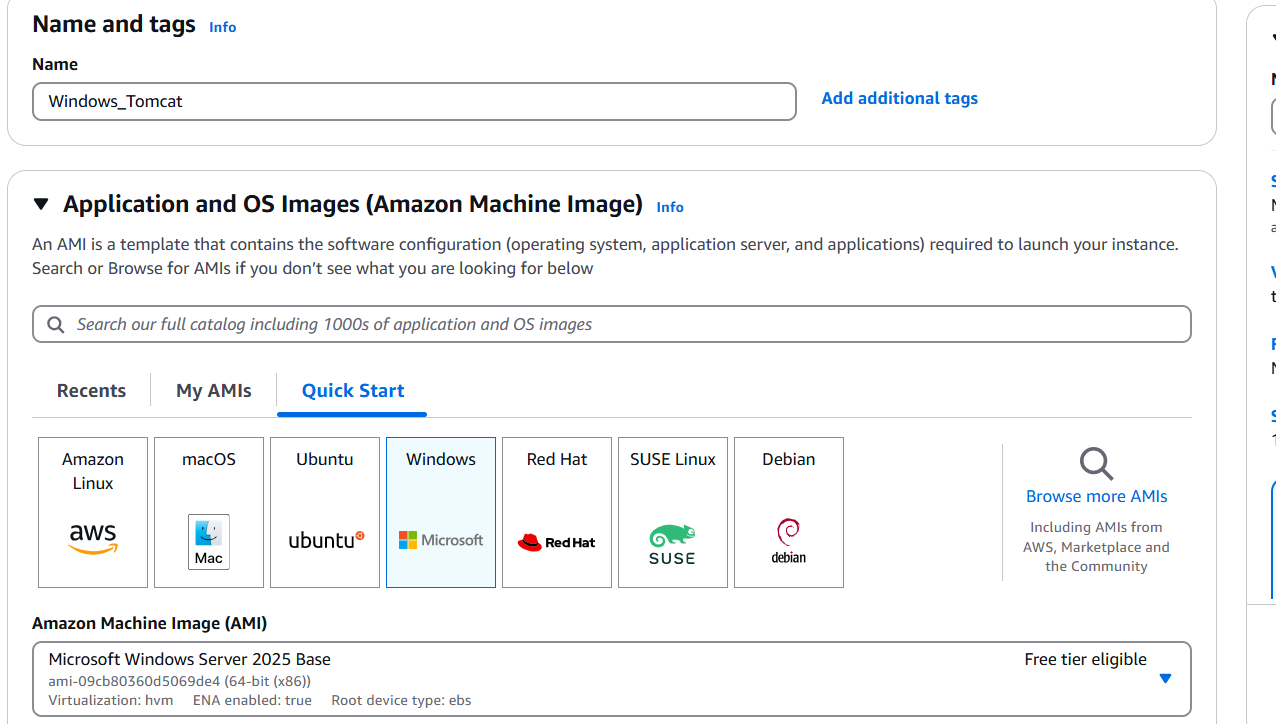
- Tomcat should now be running.

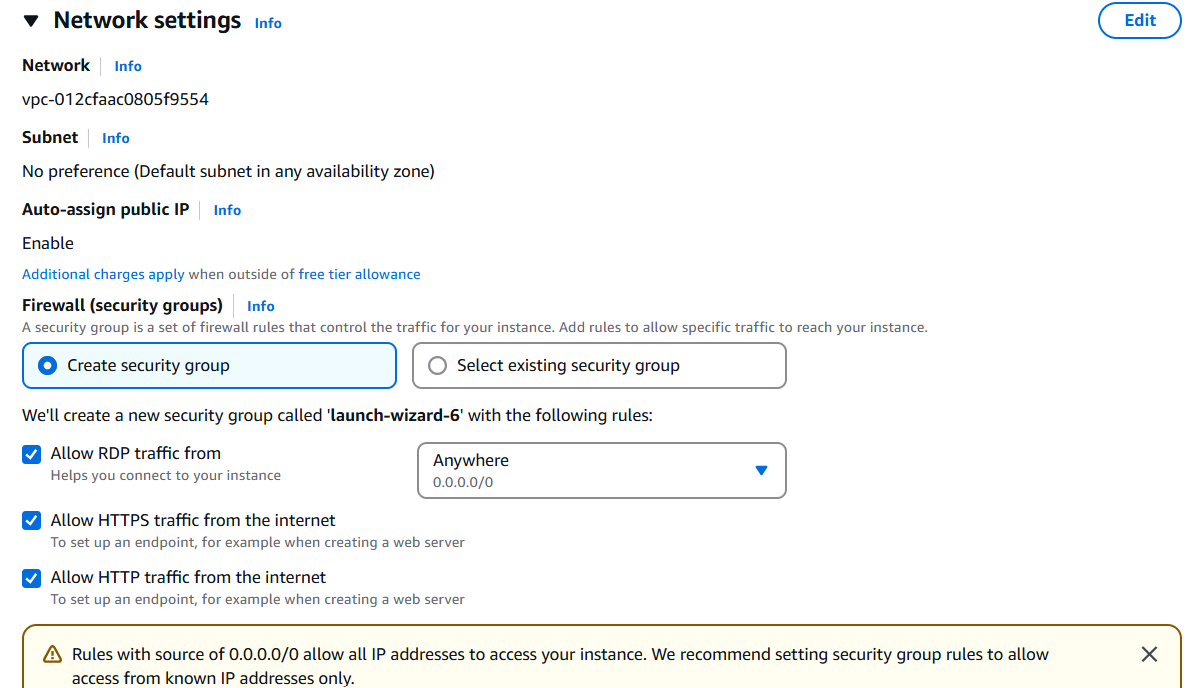
Verify Tomcat Installation

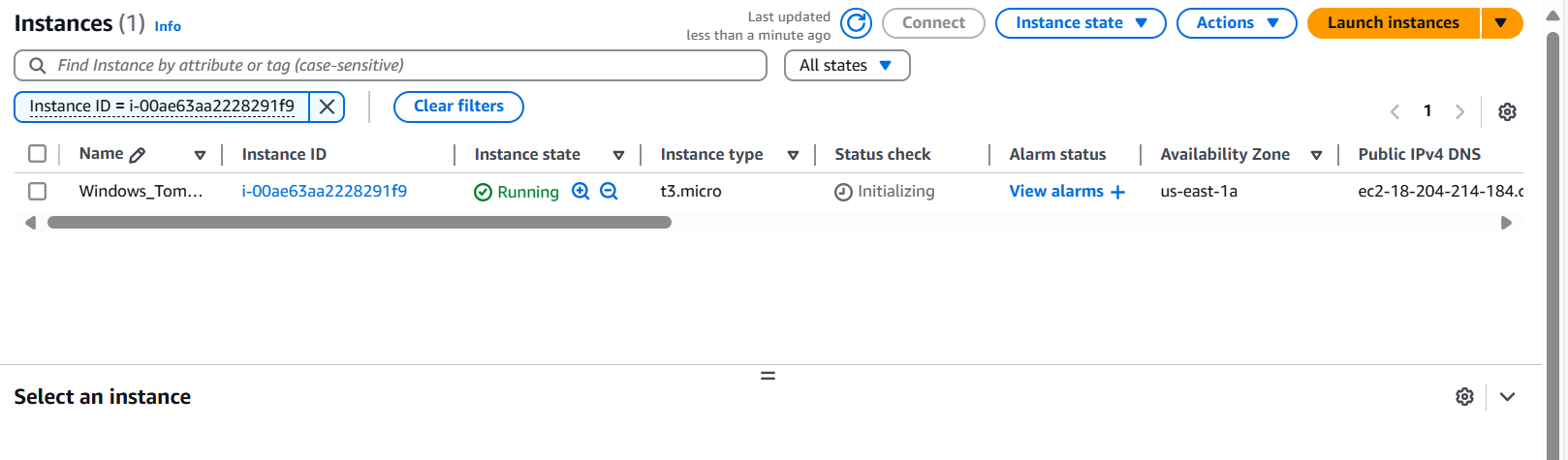
- Open a browser and go to:

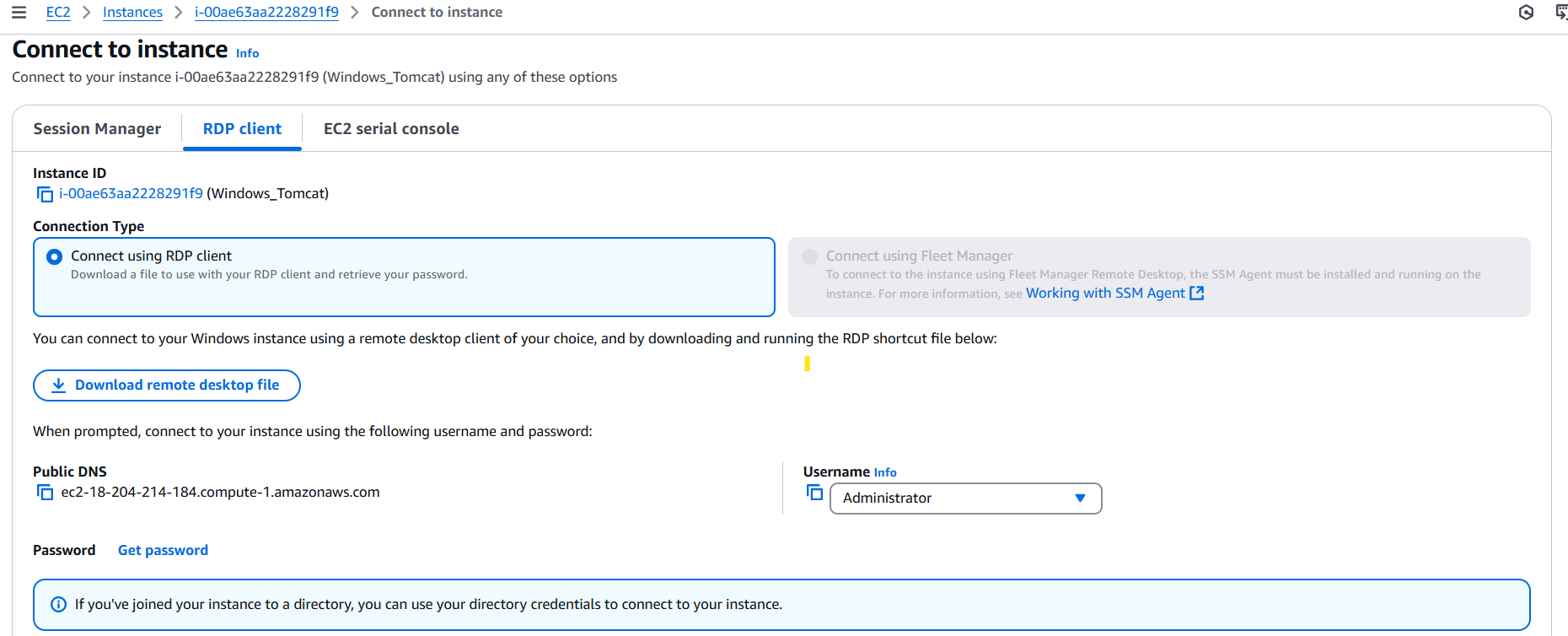
http://<3.215.180.8:8080>

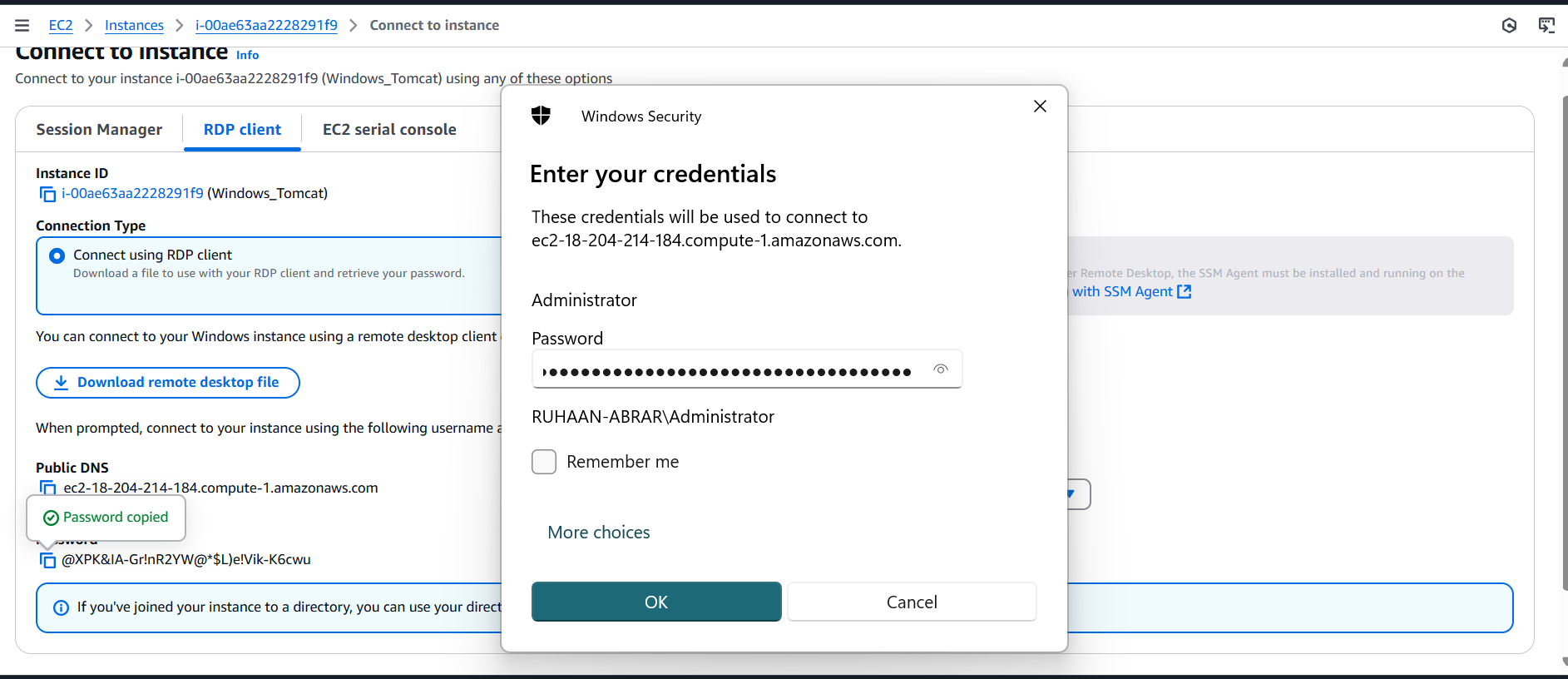
- You should see the Tomcat welcome page.

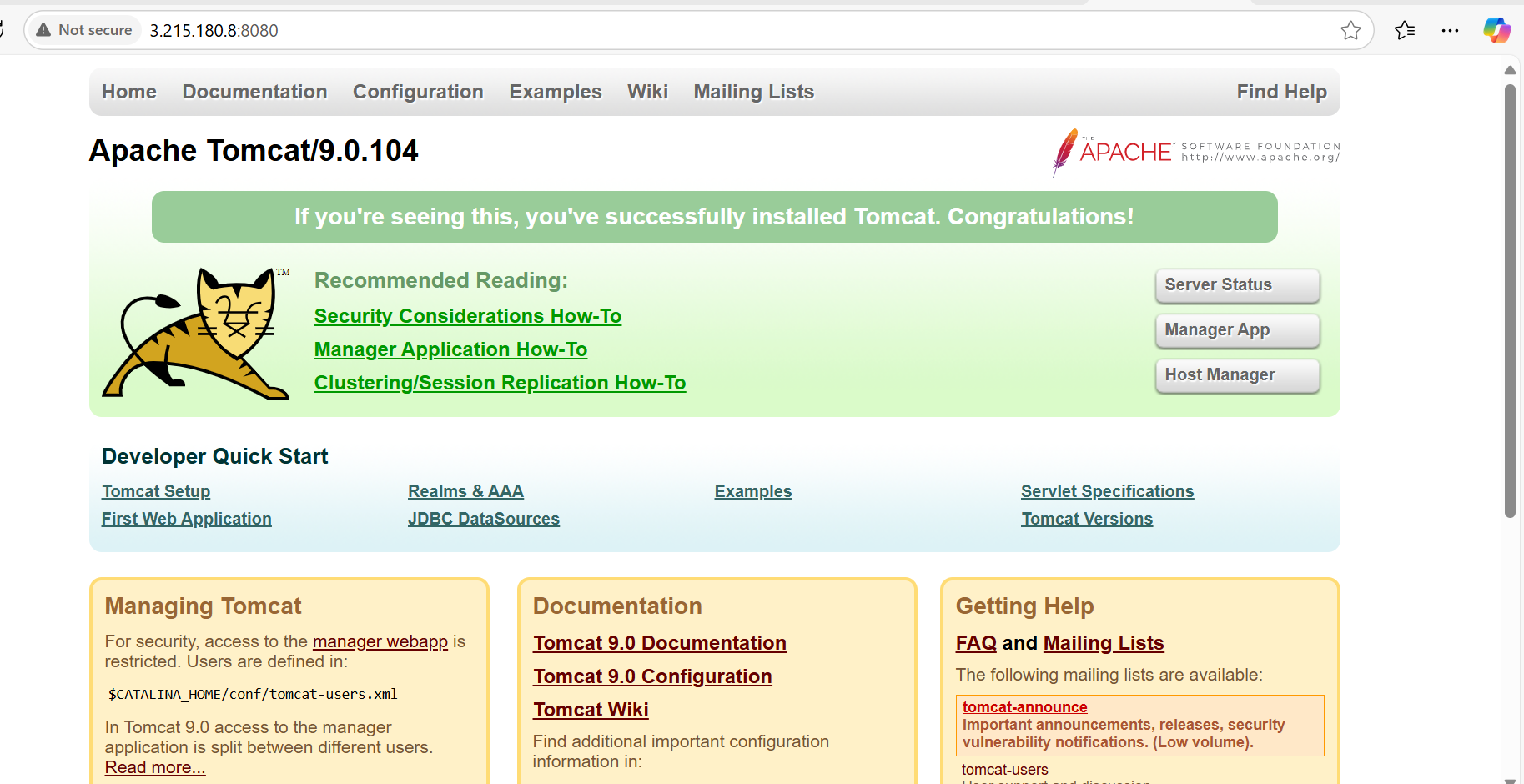




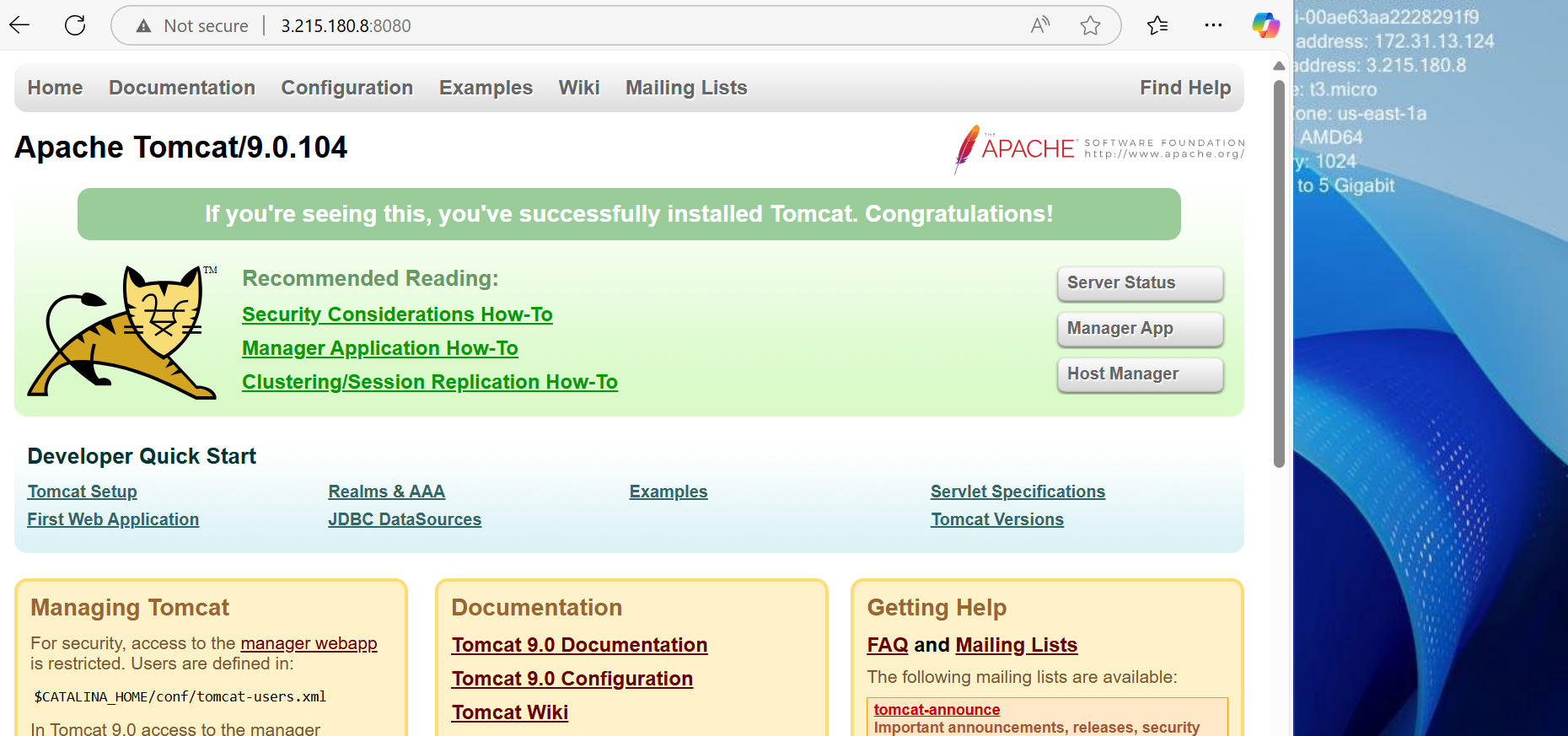






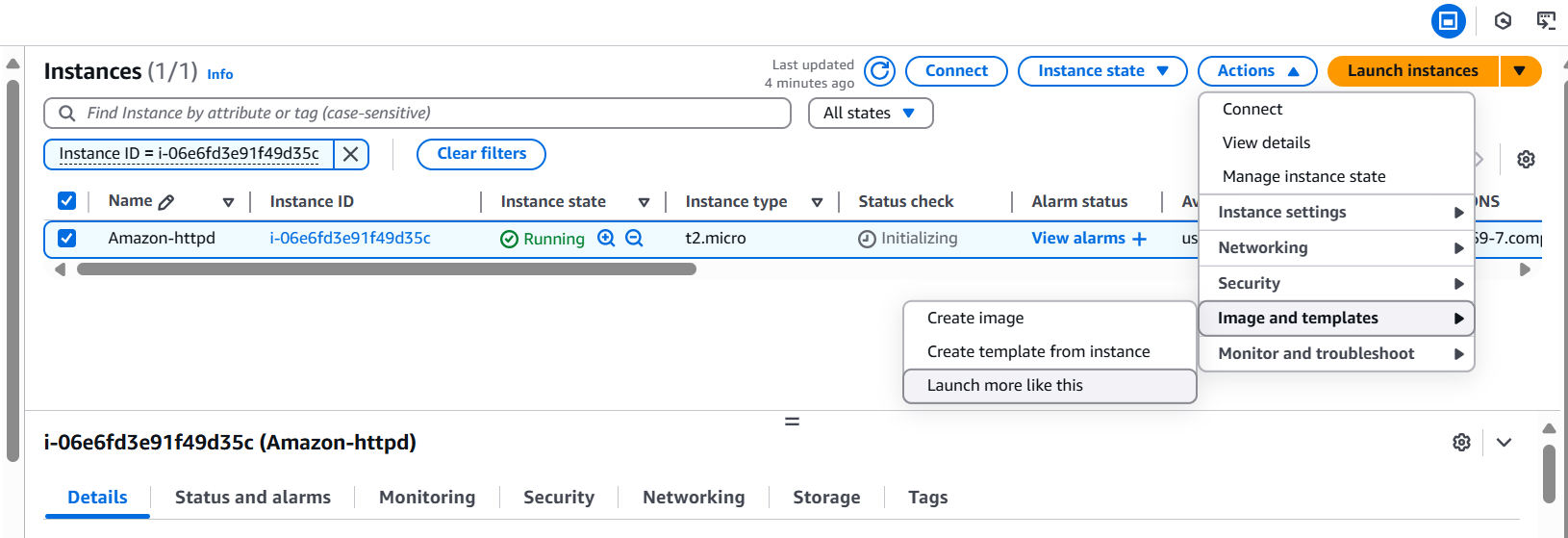


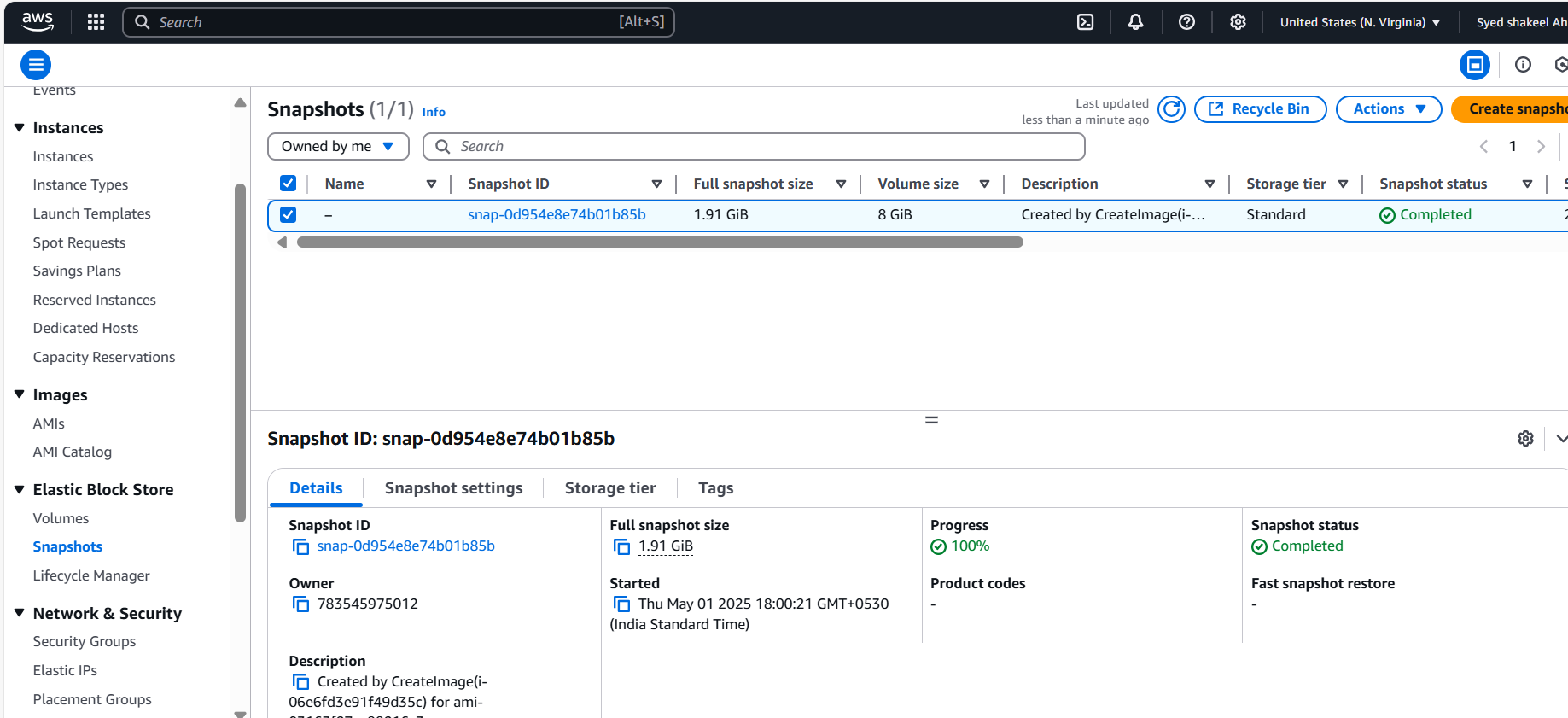


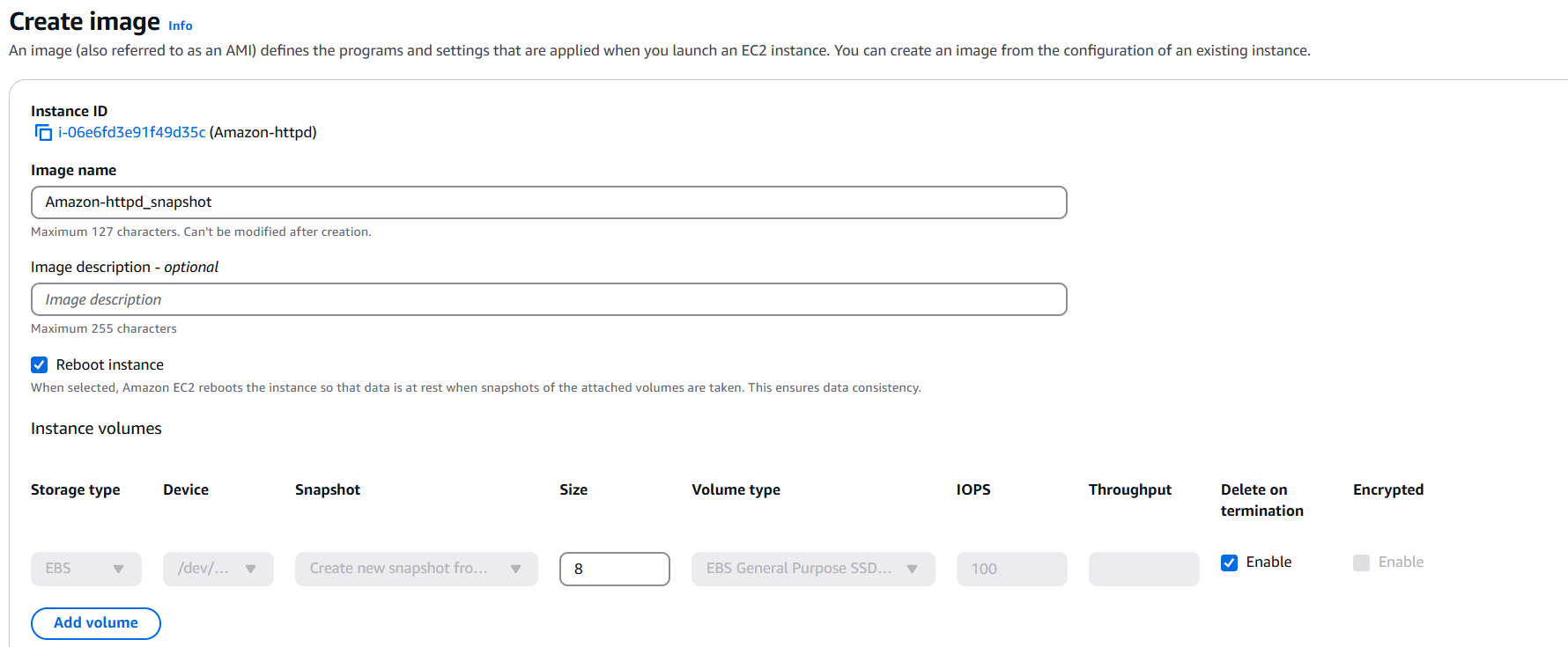


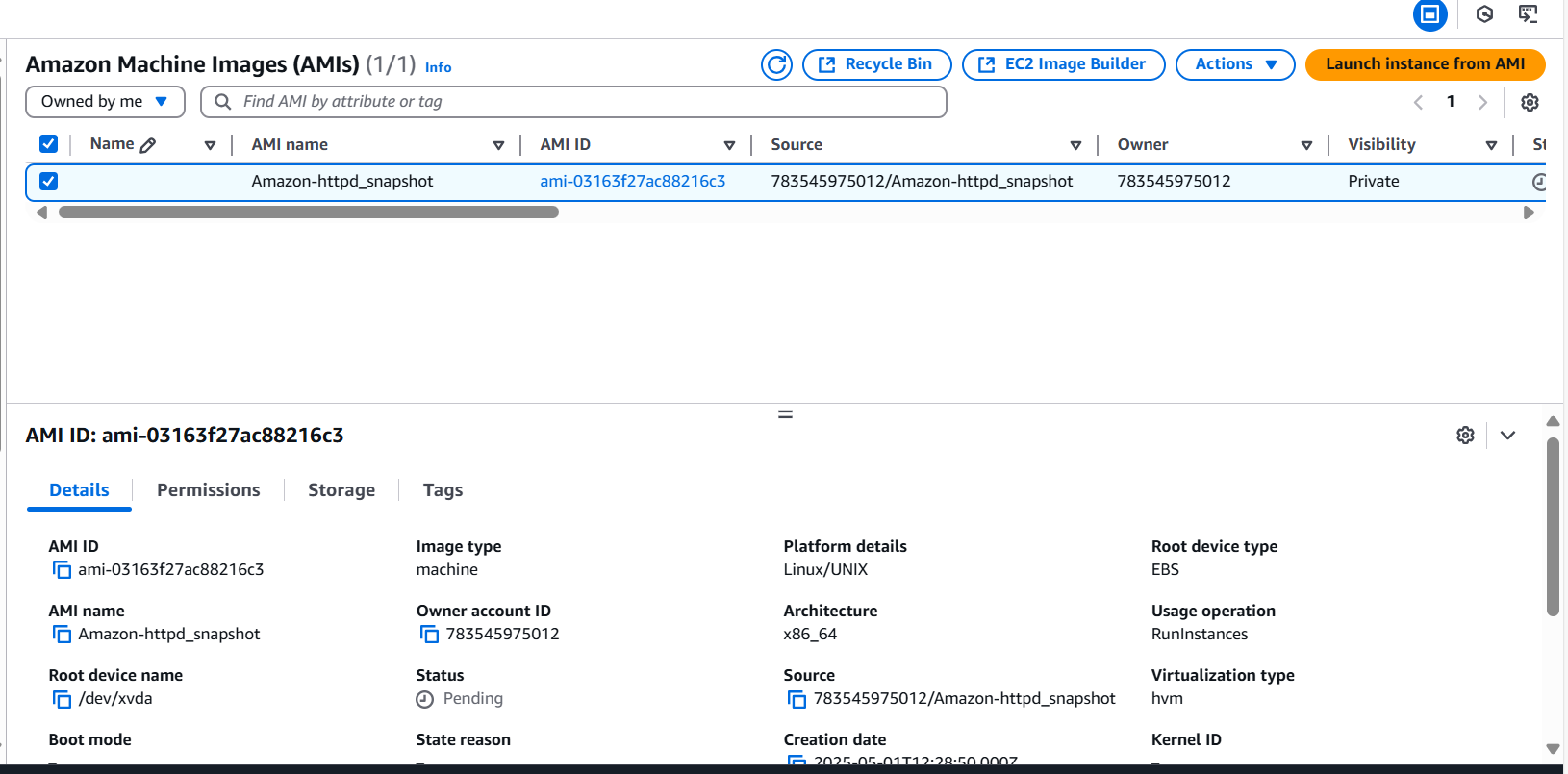
4) Take snapshot of the instance created in Task 1.

* Creating an EBS Snapshot (if your instance has a dedicated EBS volume for httpd data):
* Open the AWS EC2 console.
* Navigate to Volumes under Elastic Block Store (EBS).
* Select the volume attached to your instance.
* Click on Actions → Create Snapshot.
* Creating an Amazon Machine Image (AMI) (to save the entire instance state):
* Go to EC2 instances in the AWS console.
* Select your instance.
* Click Actions → Create image.
* Provide a name and description.
* Choose the necessary storage configurations.
* Click Create Image.

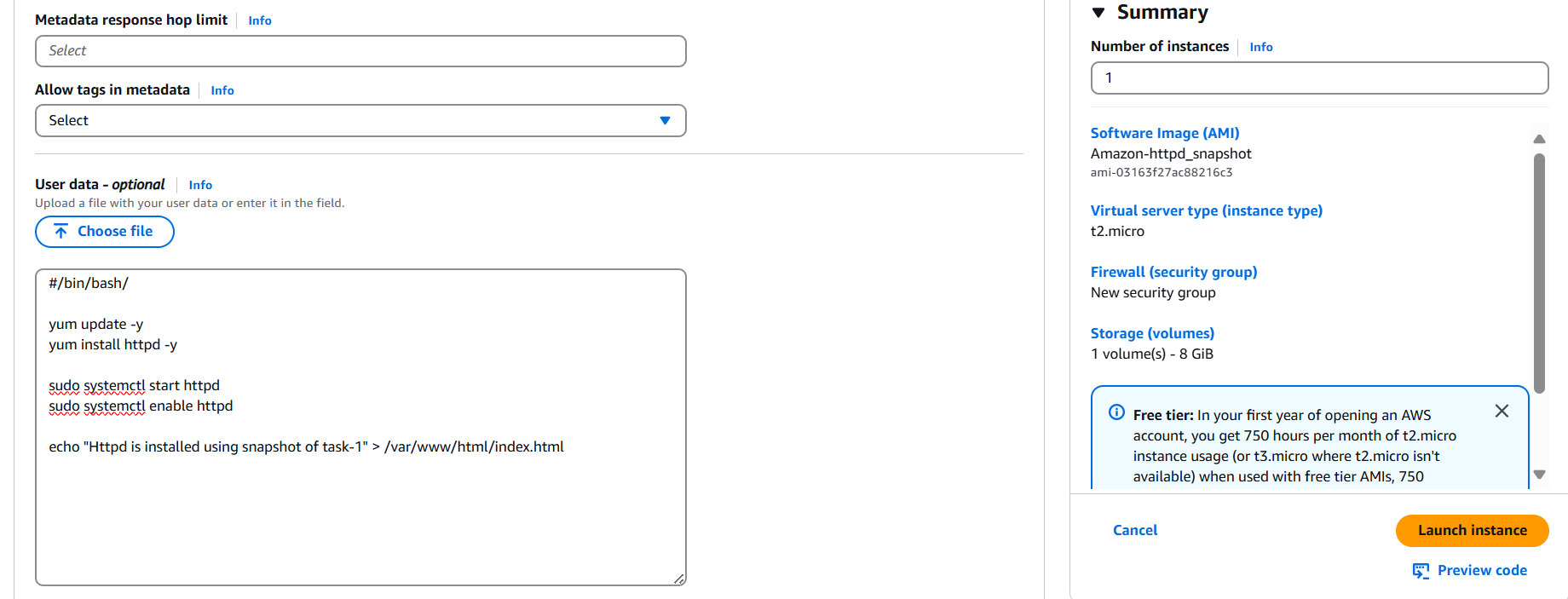


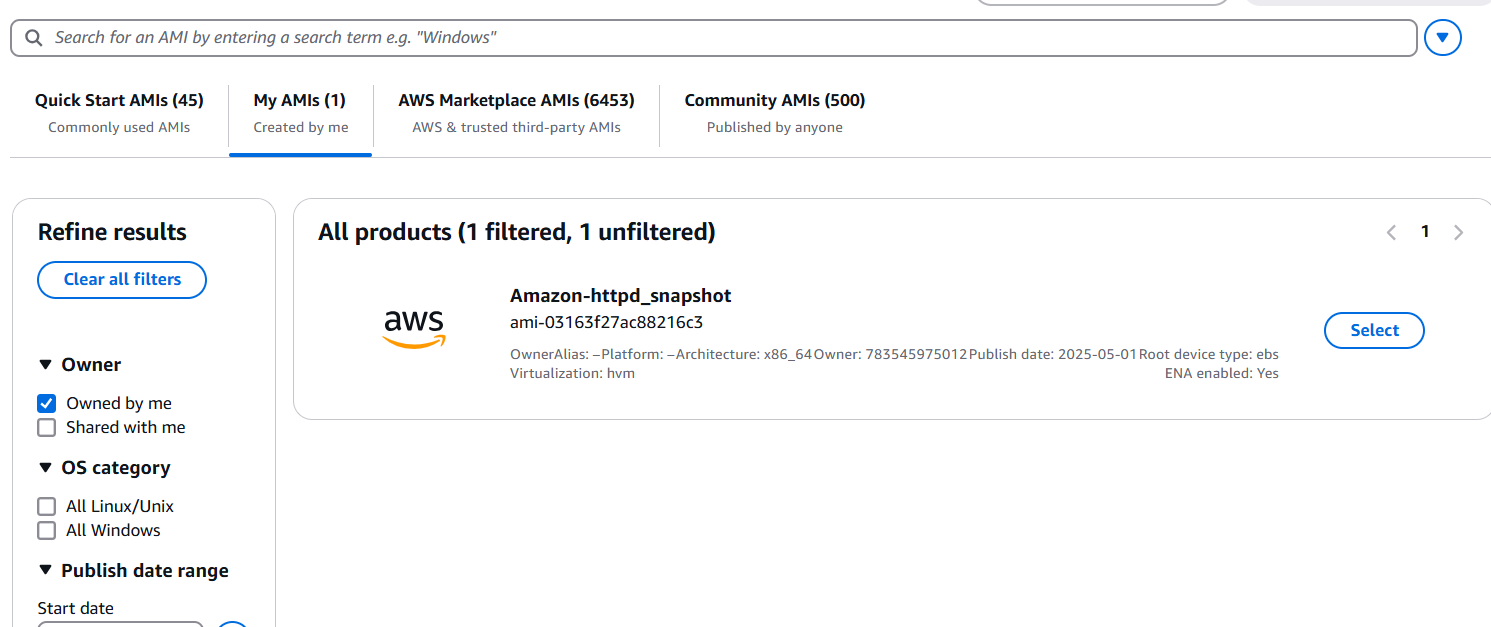


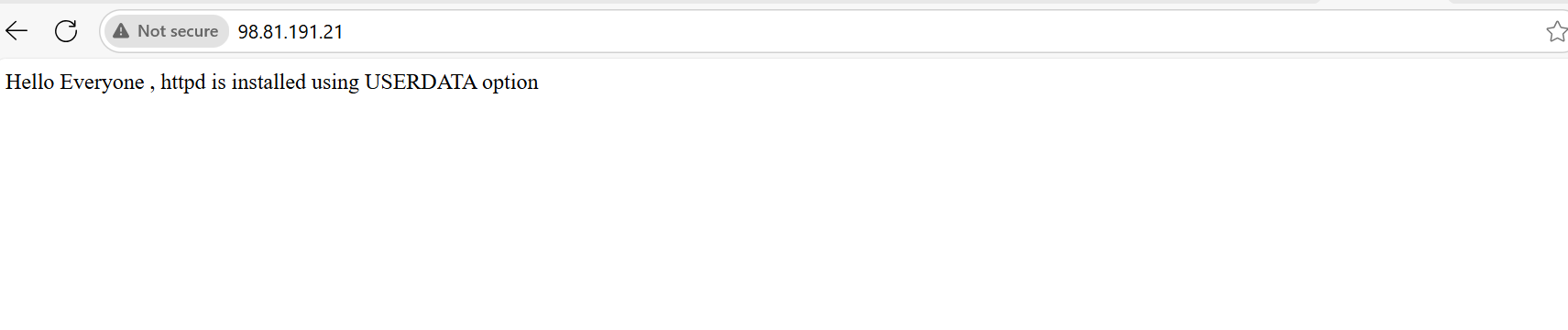












5) Assign password less authentication for ec2 created on Task 2.

As per the Task 2 install and launch the ec2 instance with ubuntu

Please activate run the instance.

Come to the Gitbash terminal try to generate the ssh-keygen and follow the steps provided using: ssh-keygen

Please copy the public key location ~/.ssh/id\_rsa.pub

Login to the instance with Ec2 console connect option

Now go to the directory cd .ssh and ls

Open the authorized\_keys and copy the key

If any user can change to root user and from there they can do the changes to production servers so to avoid this please create new user and passwod

Useradd shakeel and passwd shakeel

Now try to connect in Gitbash terminal using ssh user\_id@Public\_ip adress or ssh –i .pem user\_id@ip address.

Now the permission denied why because the reason is AWS by default it will allow to login for only ec2-user only

To login other users

Sudo su -

Vi /etc/ssh/sshd\_config

Change PasswordAuthentication no/yes

Save the file and

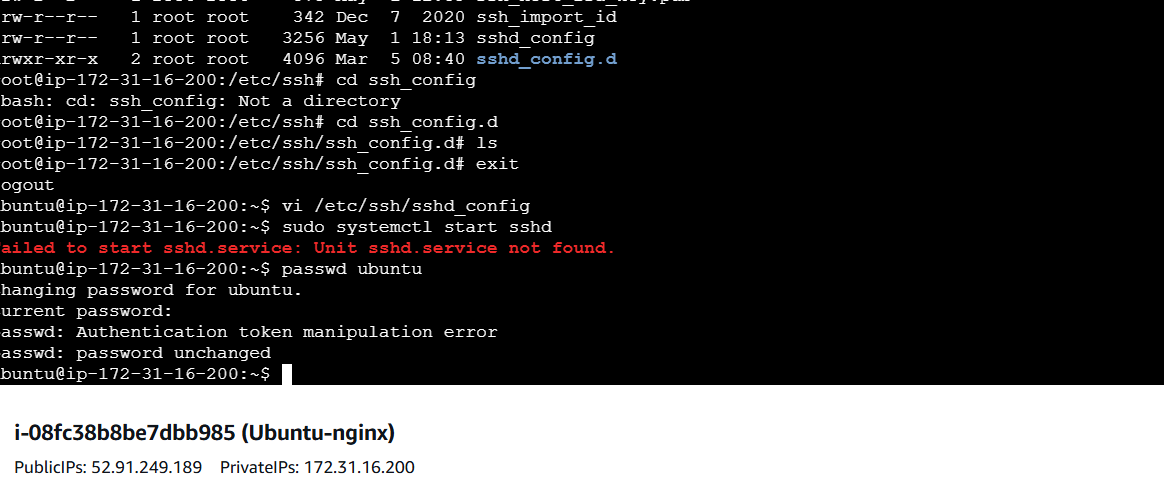
Sytemctl restart sshd

Now try login using ssh username@ip address so it will get connected Now

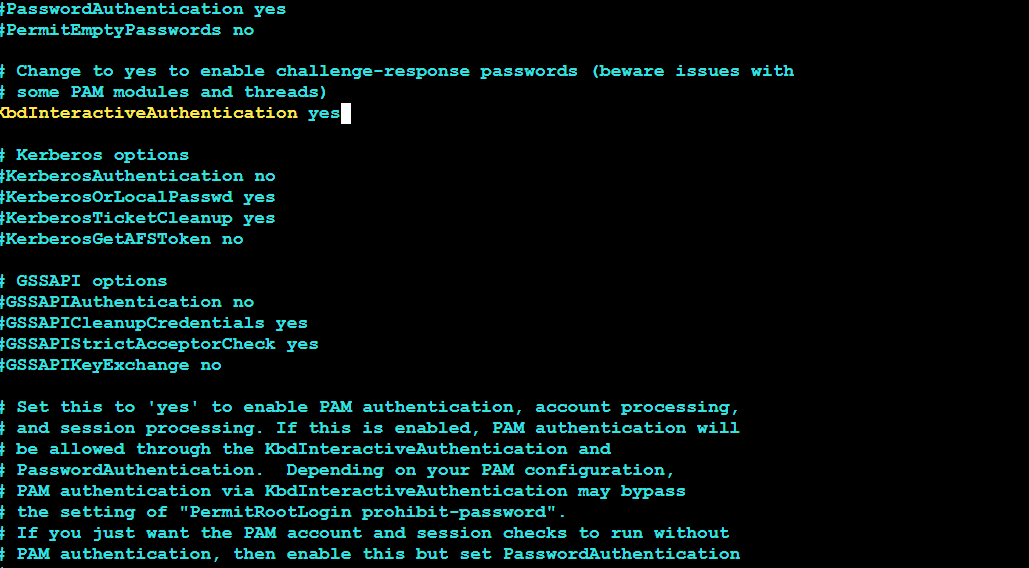
That means your passwordless authentication is successful.

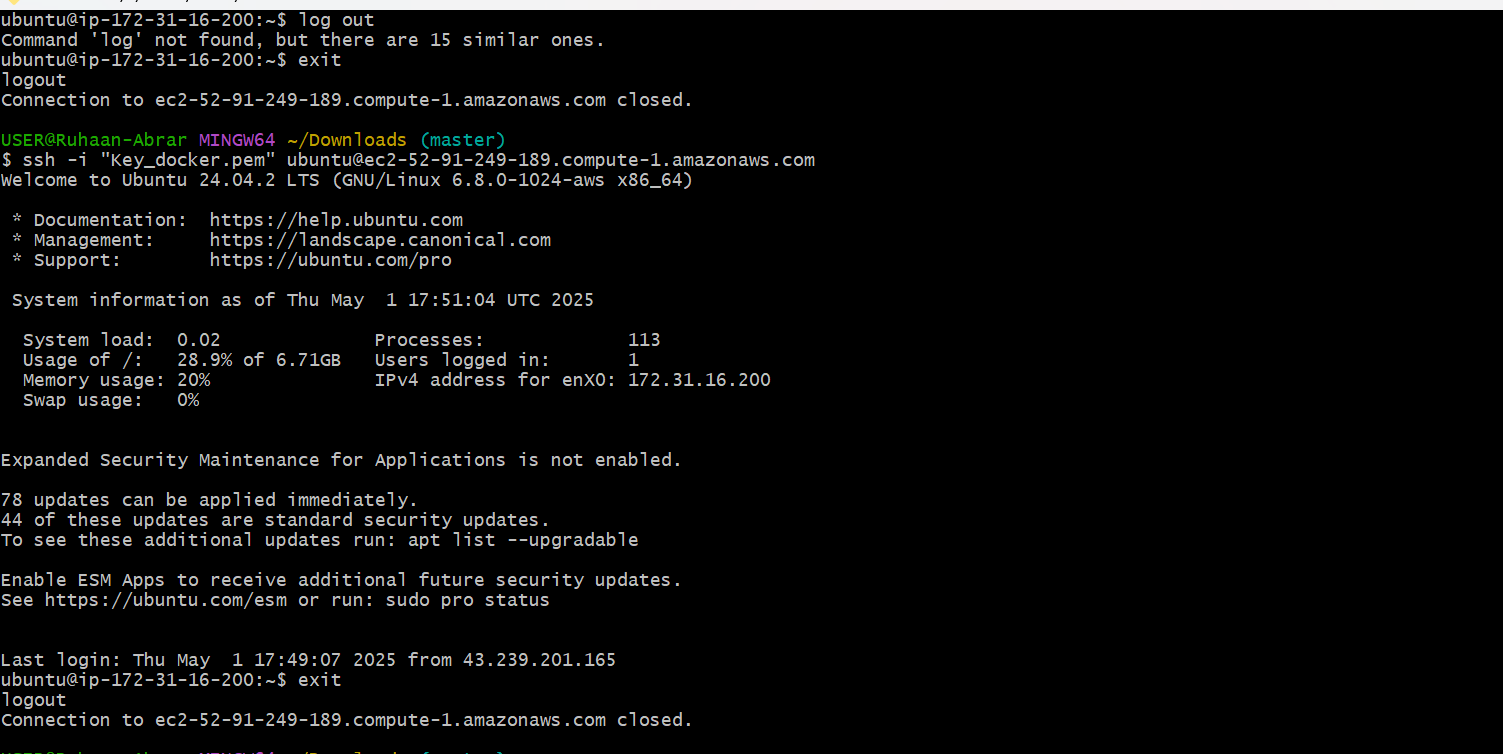
Now try to check the Nginx service status ; systemctl status nginx and also check with ip address:80 in the browser.



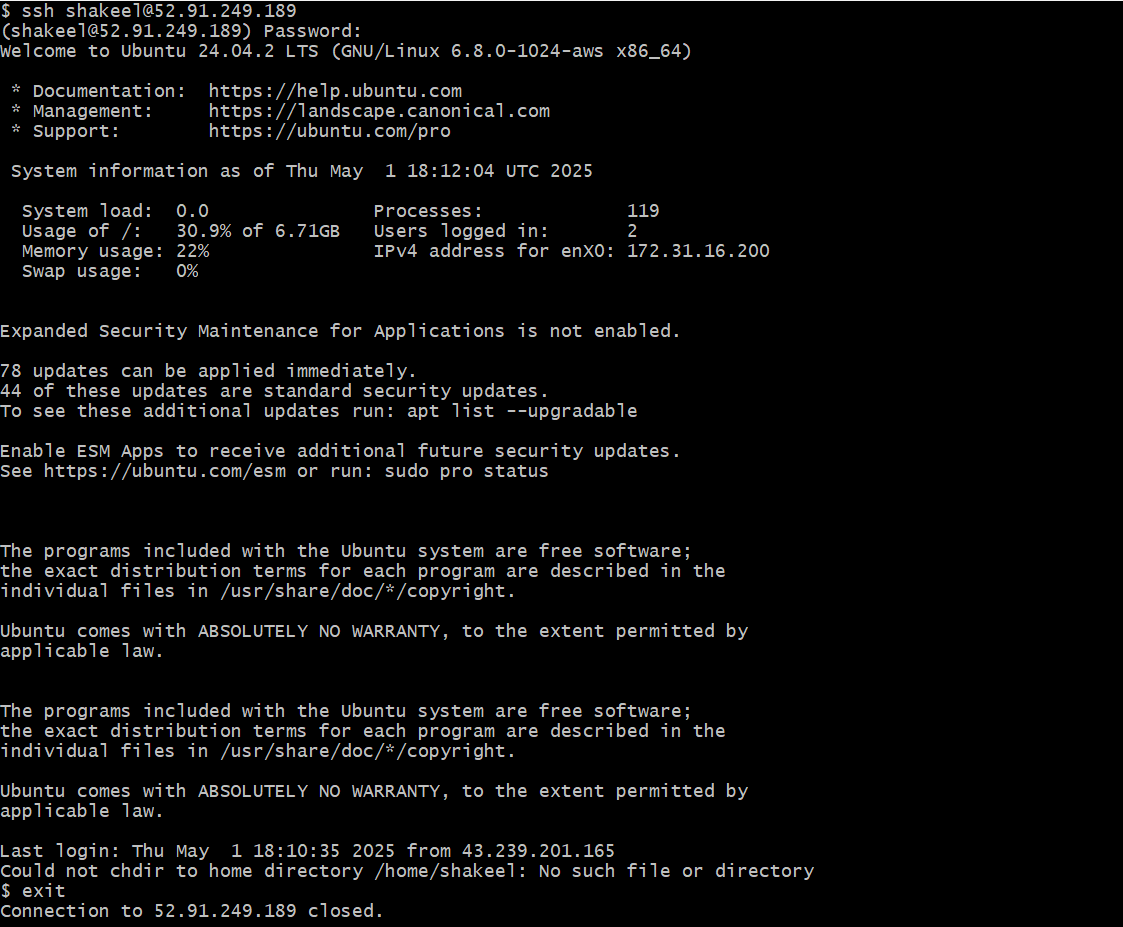


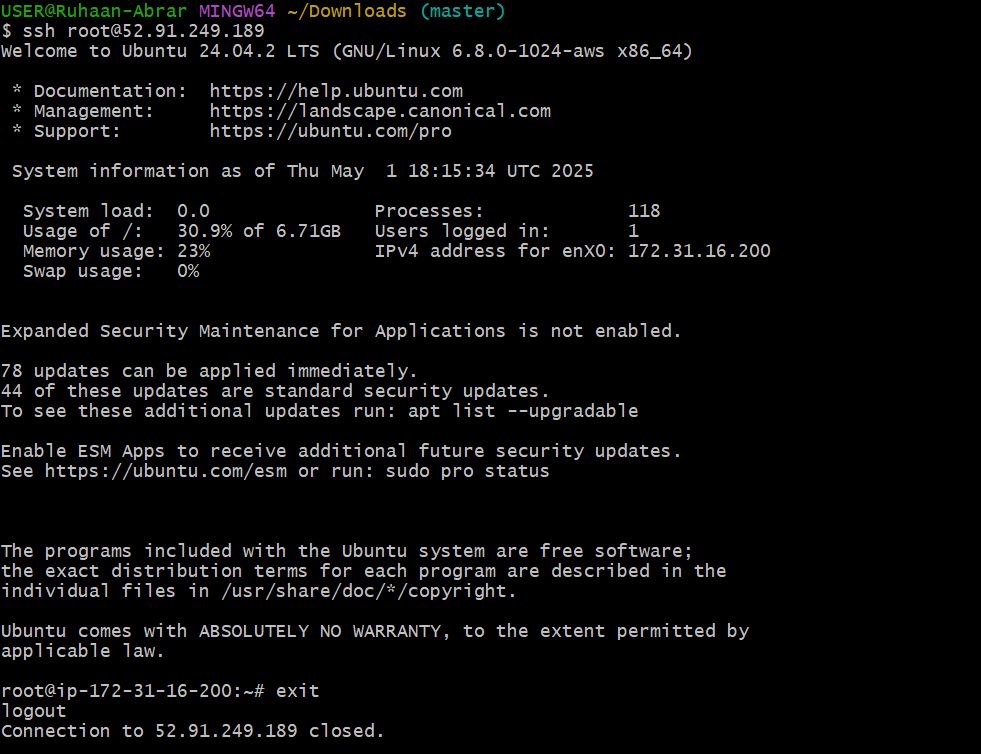












6) Launch any ec2 using spot purchasing option.

Open AWS Console & Navigate to EC2

- Login to your AWS Management Console.

- Go to EC2 Dashboard: Click Launch Instances.

- Select Amazon Machine Image (AMI):

- Search for Amazon Linux (Amazon Linux 2 recommended).

- Select 64-bit (x86) or 64-bit (Arm) based on your architecture

Configure Spot Instance Request

- Instance Type: Choose instance type based on your workload (t3.micro for free tier, c5.large for high performance, etc.).

- Enable Spot Instances:

- Tick the Request Spot Instances option.

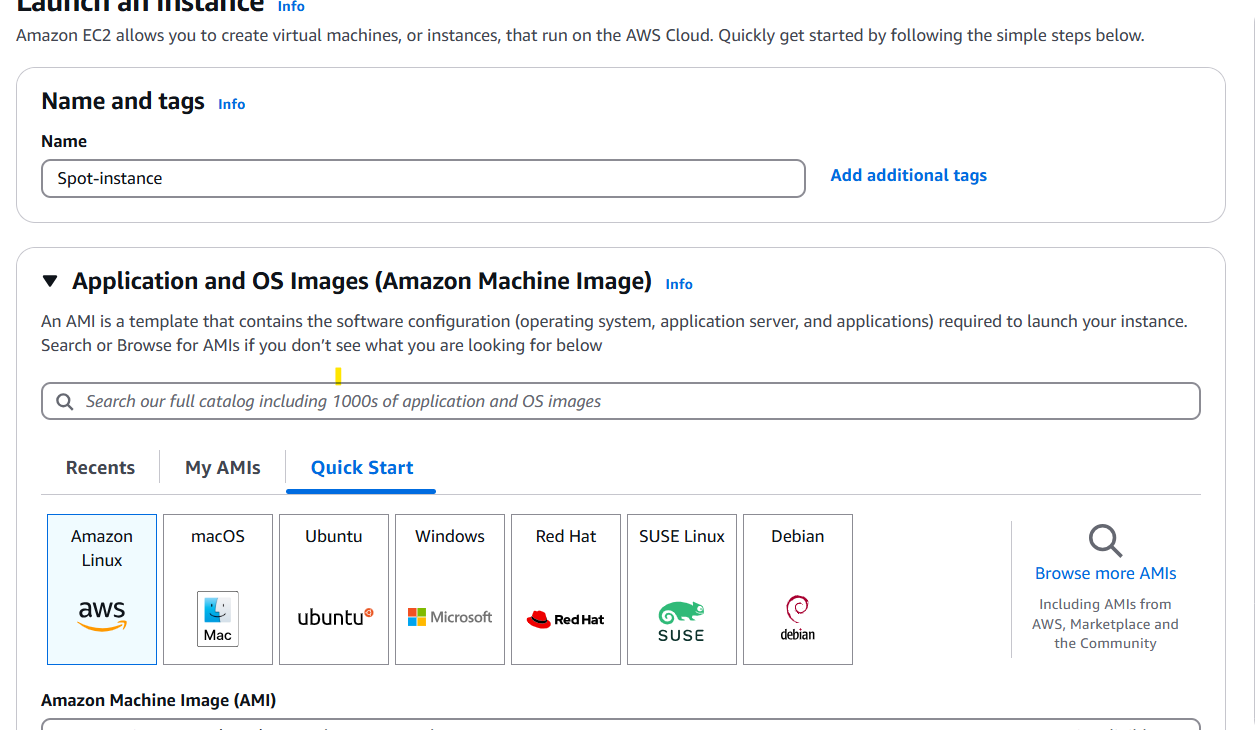
- Define Max Price (optional) or let AWS optimize pricing.

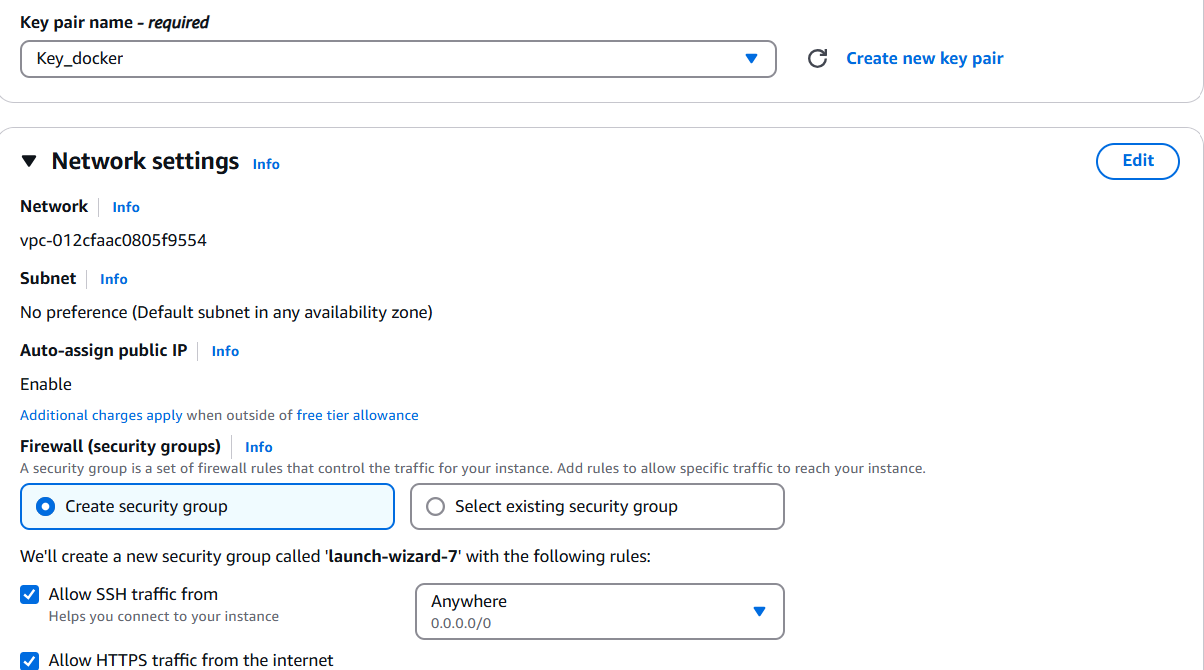
- Select Interruption Behavior (stop, terminate, hibernate).

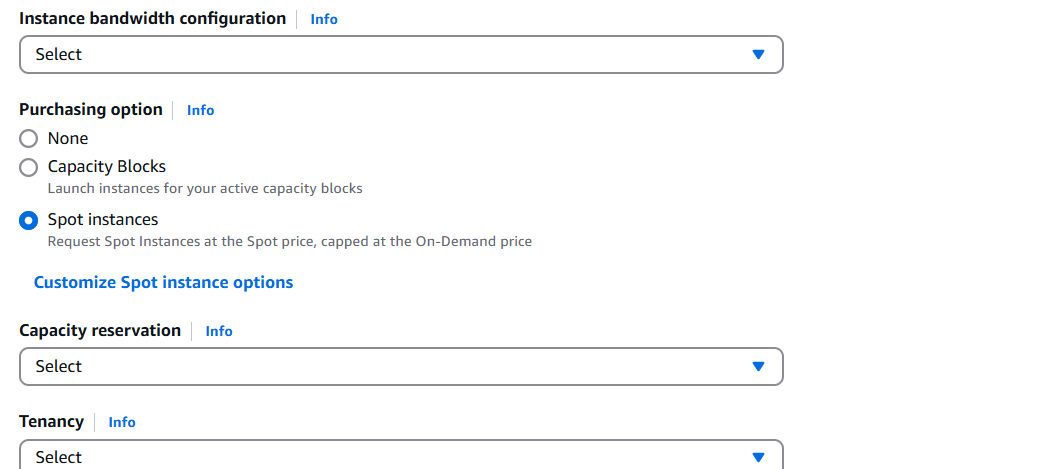
Configure Network & Storage- Configure Instance Details:

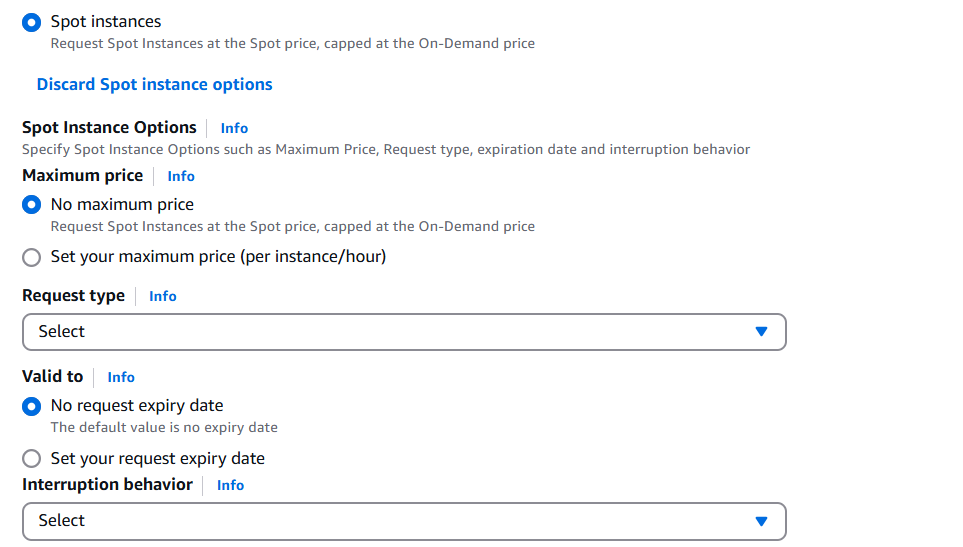
- Set VPC & Subnet (Auto-assign Public IP enabled).

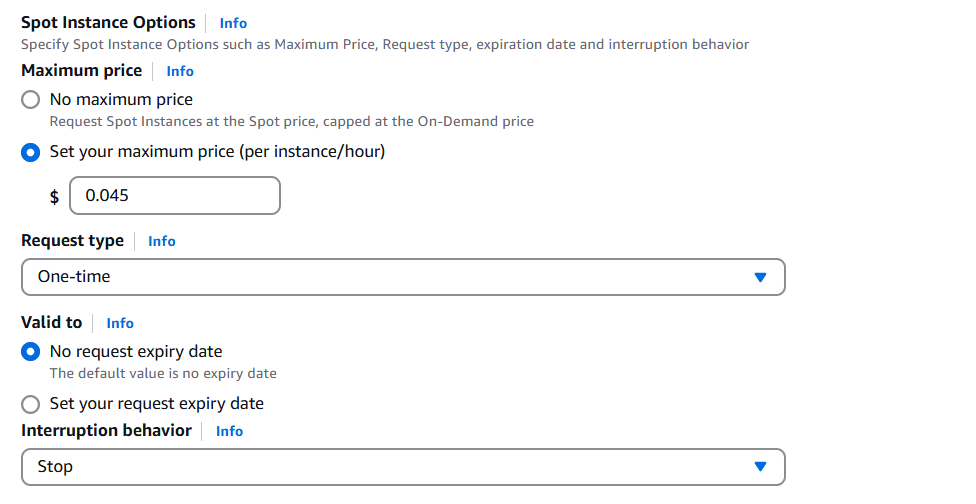
- Enable IAM Role for required permissions.

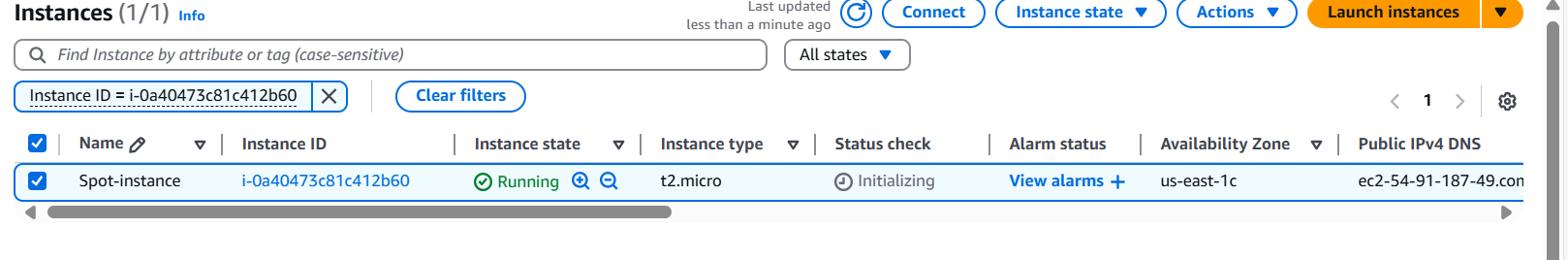












7) Enable Termination policy on ec2 created in Task 2.

Launch the instance by selecting required parameters which we have followed in Task 2

After successful launch of the instance

Enable Termination Protection

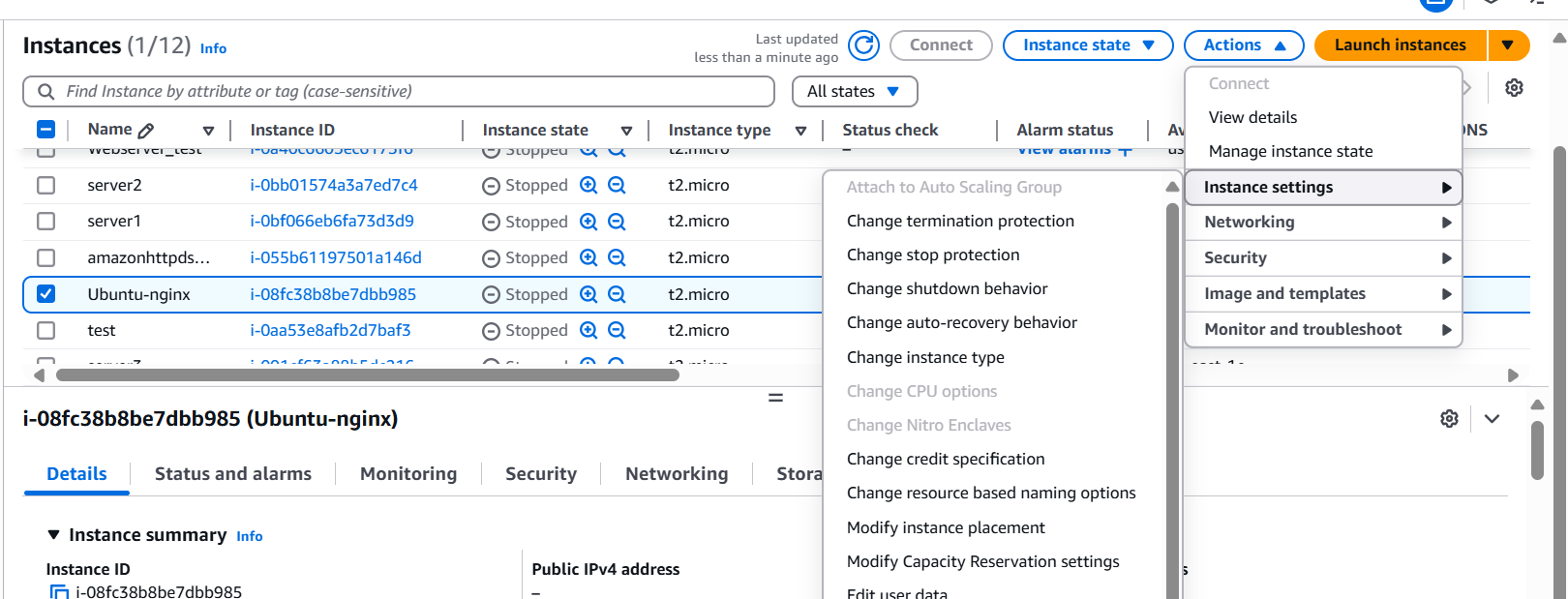
To prevent accidental termination:

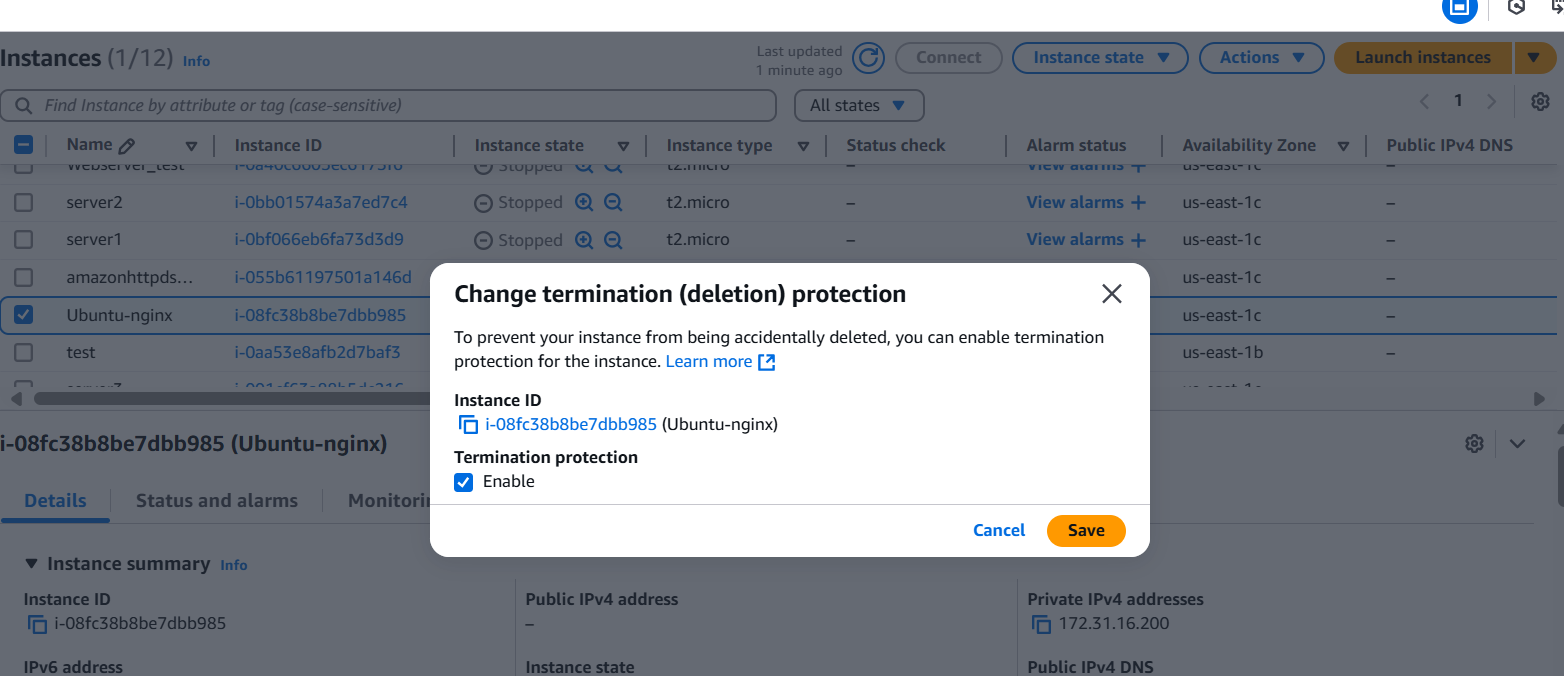
- Go to EC2 Dashboard → Select the instance.

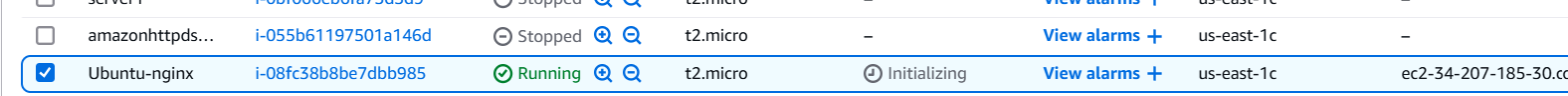
- Click Actions → Instance Settings → Change Termination Protection.

- Enable termination protection and Save.

Now, your instance cannot be terminated via console unless termination protection is disabled.







8) Launch one ec2 using Aws CLI.

To install & Configure AWS CLI in Gitbash Ensure AWS CLI is installed: curl "[https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip"](https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip%22) -o "awscliv2.zip" unzip awscliv2.zip

sudo ./aws/install

Configure AWS CLI with AWS security credentials:

aws configure

We need to provide:

* AWS Access Key ID
* AWS Secret Access Key
* Default Region (e.g., us-east-1)
* Output format (json recommended)

Find Amazon Linux 2 AMI ID Get the latest Amazon Linux 2 AMI ID:

aws ec2 describe-images --owners amazon --filters "Name=name,Values=amzn2-ami-hvm-*-x86\_64-gp2" --query 'Images[*].[ImageId,Name]' --output table

Copy the AMI ID (e.g., ami-xxxxxxxxxxxxxxxxx).

Create SSH Key Pair or else we can use the Same old ssh keypair which we used earlier to launch the instances

Create a new key pair to connect via SSH:

aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text > MyKeyPair.pem

Set correct permissions: chmod 400 MyKeyPair.pem

We need to get the security group and subnet-ids in the particular region where we want to launch the instance. After we got these details we can copy them and paste them in below command

Launch EC2 Instance Run this command:

aws ec2 run-instances

--image-id ami-0c02fb55956c7d316

--count 1

--instance-type t2.micro

--key-name Key\_docker

--security-group-ids sg-0d4b9233d07283c28

--subnet-id subnet-0df05356a4133505a

--associate-public-ip-address

--tag-specifications 'ResourceType=instance,Tags=[{Key=AWSCLI,Value=AmazonLinuxInstance}]' --region us-east-1

aws ec2 describe-subnets

--subnet-ids subnet-0df05356a4133505a

--query 'Subnets[0].VpcId'

--output text

aws ec2 describe-security-groups

--filters,Name=vpc-id Values=vpc-

--query 'SecurityGroups[\*].{ID:GroupId,Name:GroupName}'

--output table

