AWS-Cloud-Assignment-1

1. Launch one EC2 using Amazon Linux 2 image and add a script in user data to install Apache.

Launched the EC2 instance with amazon linux, with t3.micro instance type, opened the advanced details at the bottom and entered following script containing the installation of apache

Script input:

#!/bin/bash

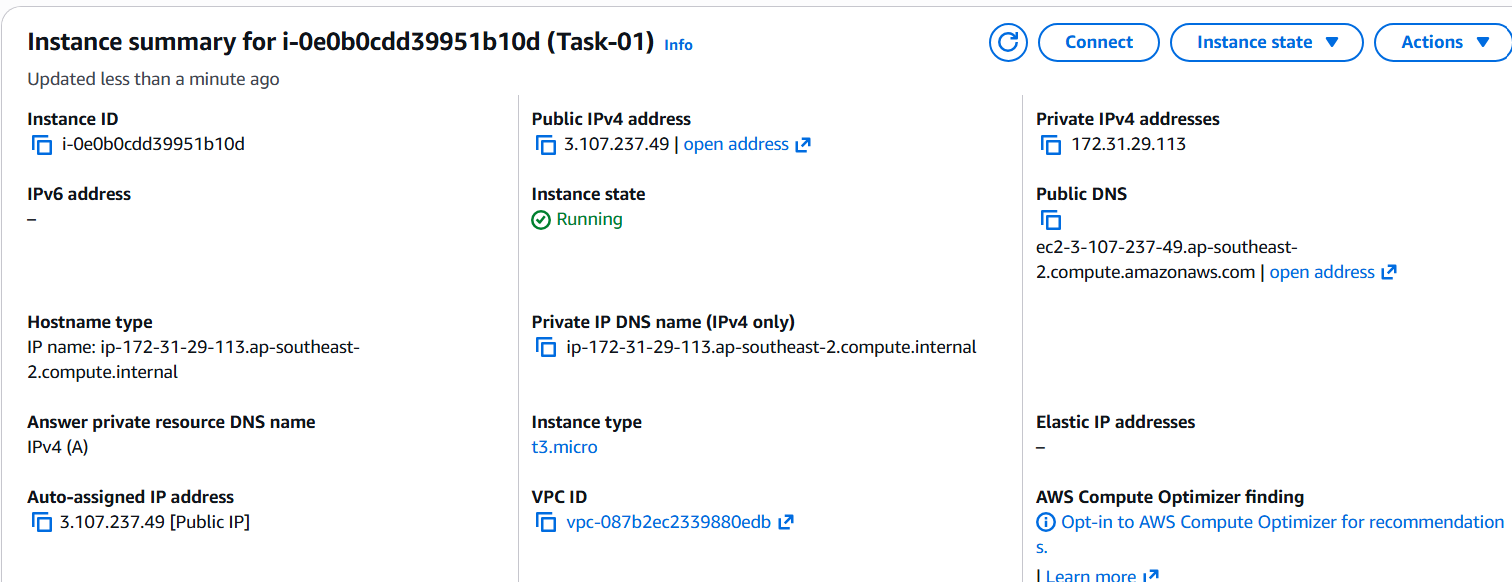
Sudo yum update -y

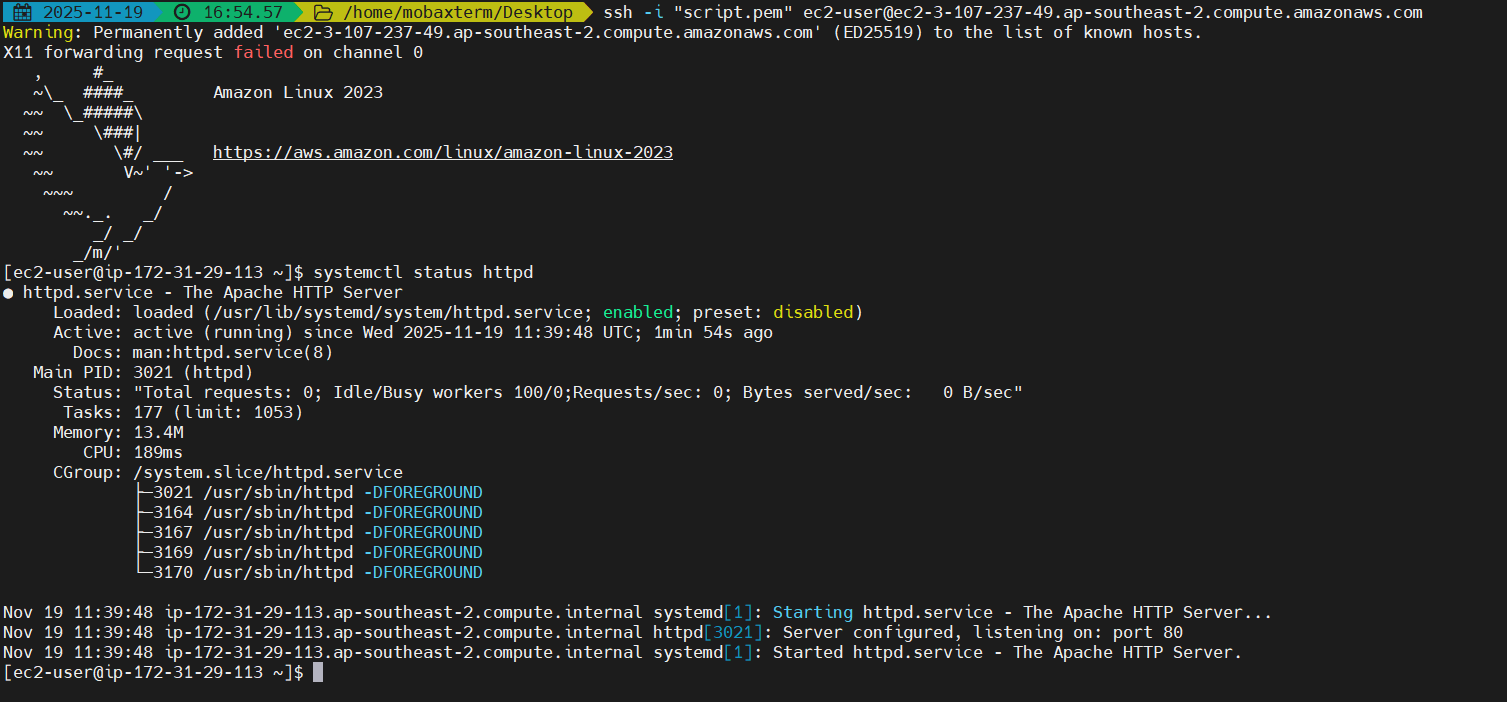
Sudo yum install httpd -y

systemctl start httpd

systemctl enable httpd

echo "<h1>Task-01 Apache Web Server!</h1>" > /var/www/html/index.html

1. Launch one EC2 using Ubuntu image and add a script in user data to install Nginx.

Launched the EC2 instance with ubuntu, with t3.micro instance type, opened the advanced details at the bottom and entered following script containing the installation of nginx

Script input:

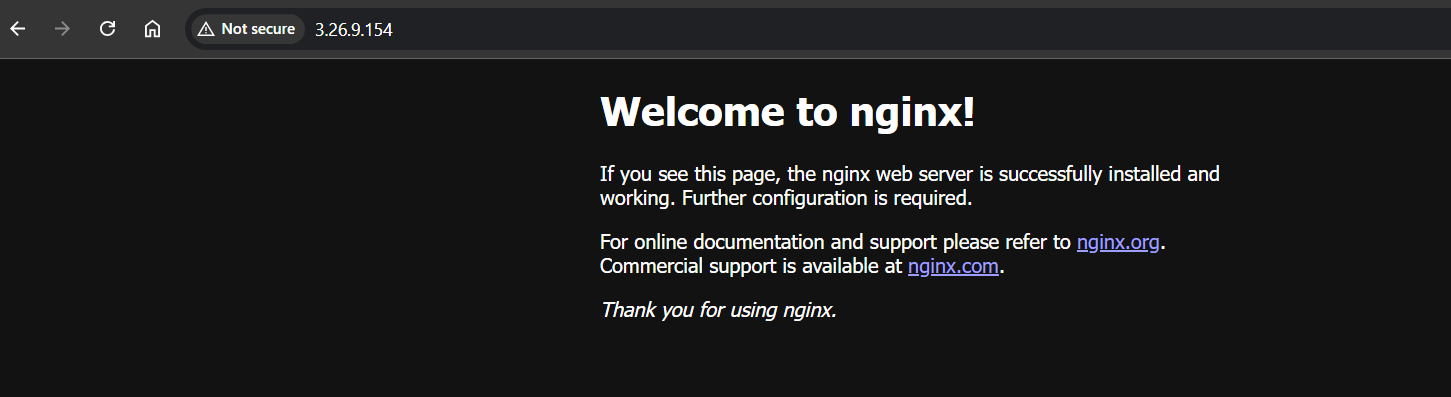
#!/bin/bash

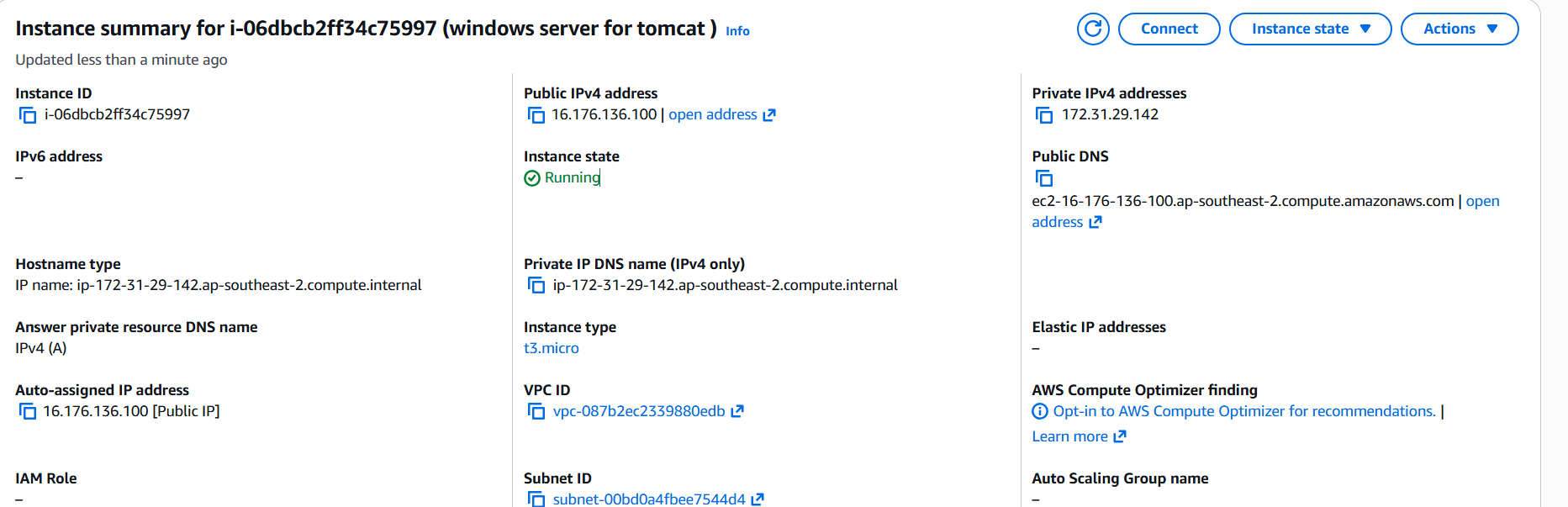
Sudo apt update -y

Sudo apt install nginx -y

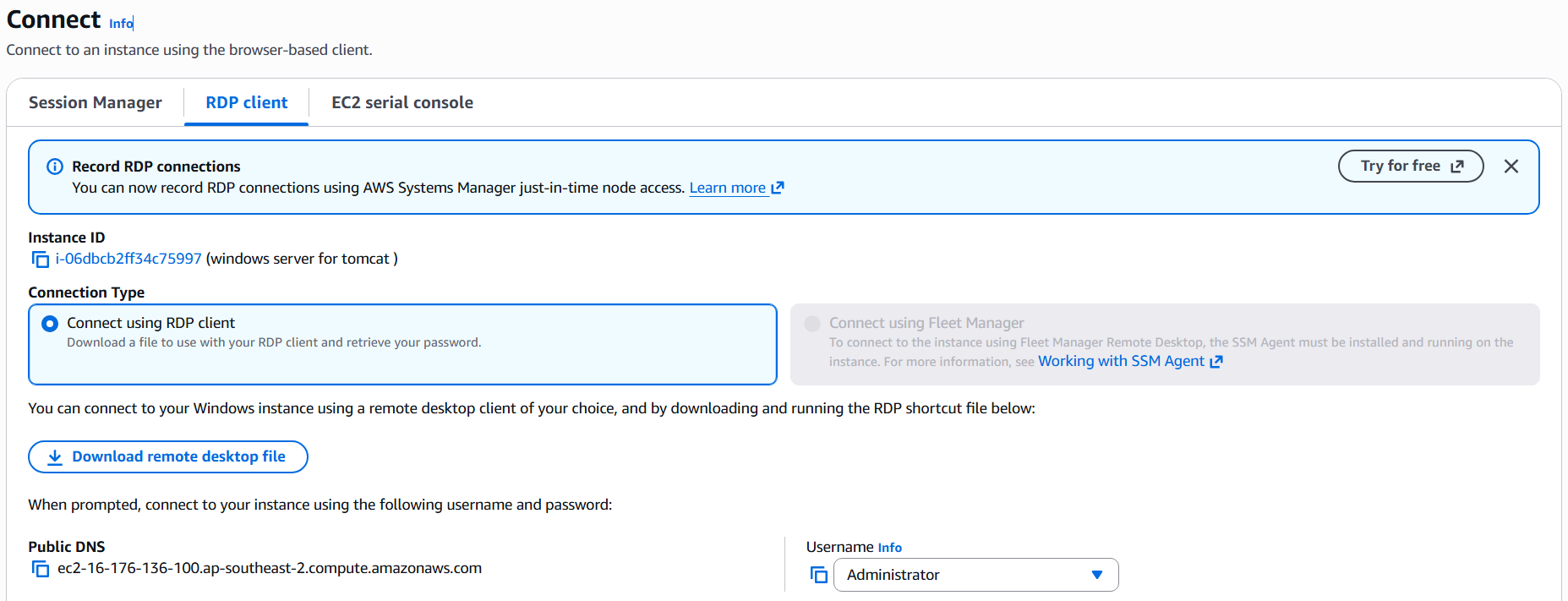
systemctl start nginx

systemctl enable nginx

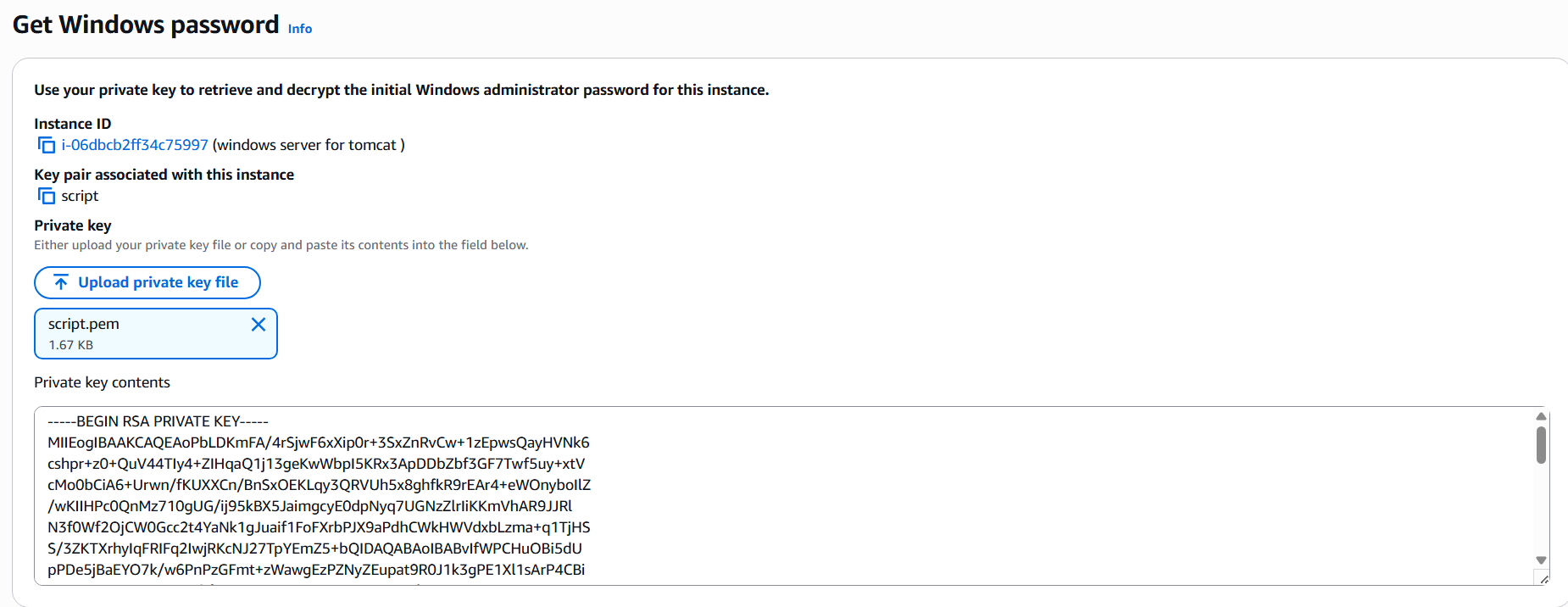
  

1. Launch one Windows server and install Tomcat on Windows. 

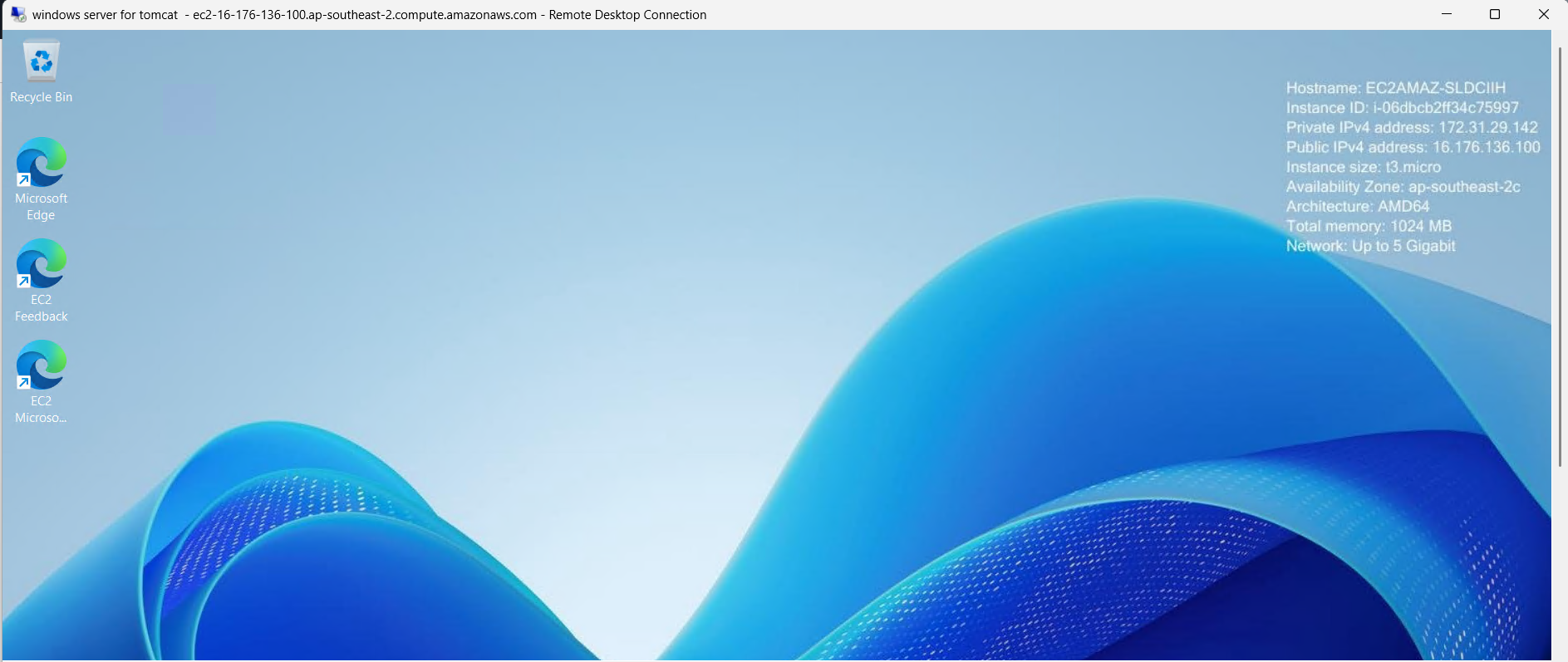
Launch the windows server 2025 base server on AWS console



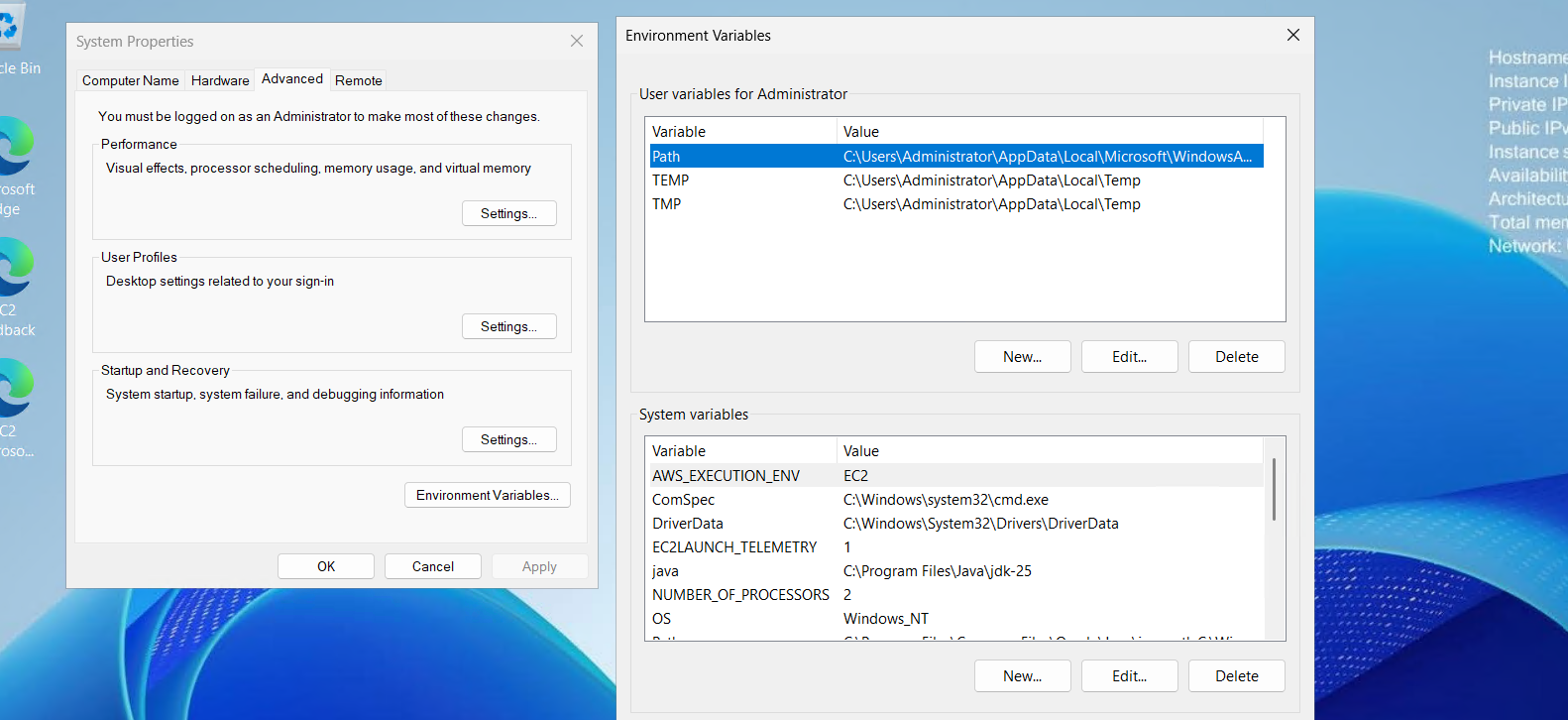
Connect the RDP connection by copying the public DNS of the server



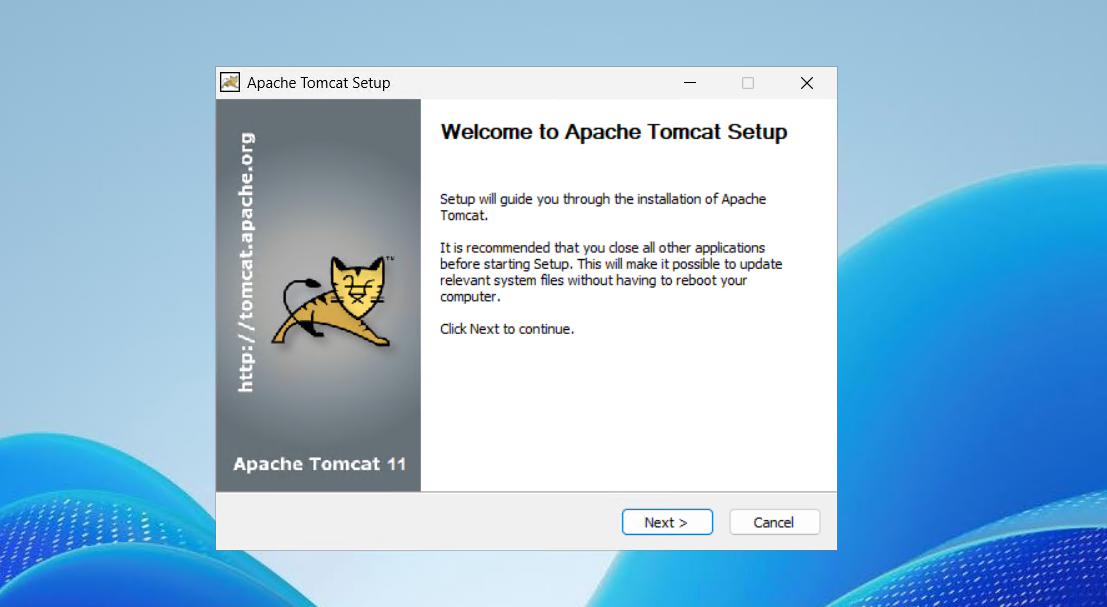
Retrieve the password of the server by submitting the pem file in the get password and copy the password and enter it in the password section



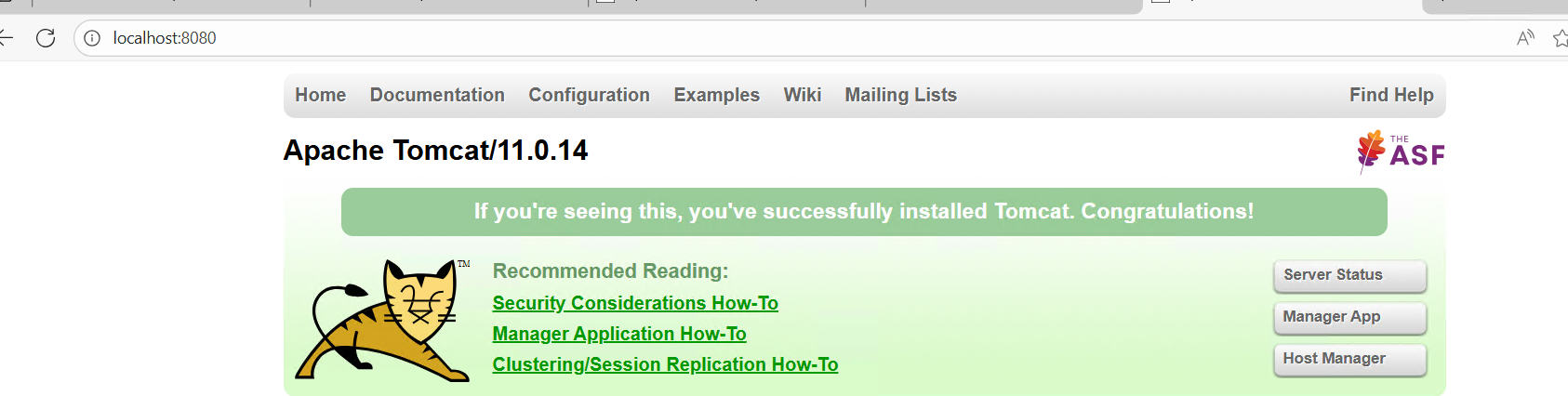
Install jdk latest version in the windows server and copy the path of the bin folder in the c drive, paste the java path In the user variable on the environment variables.



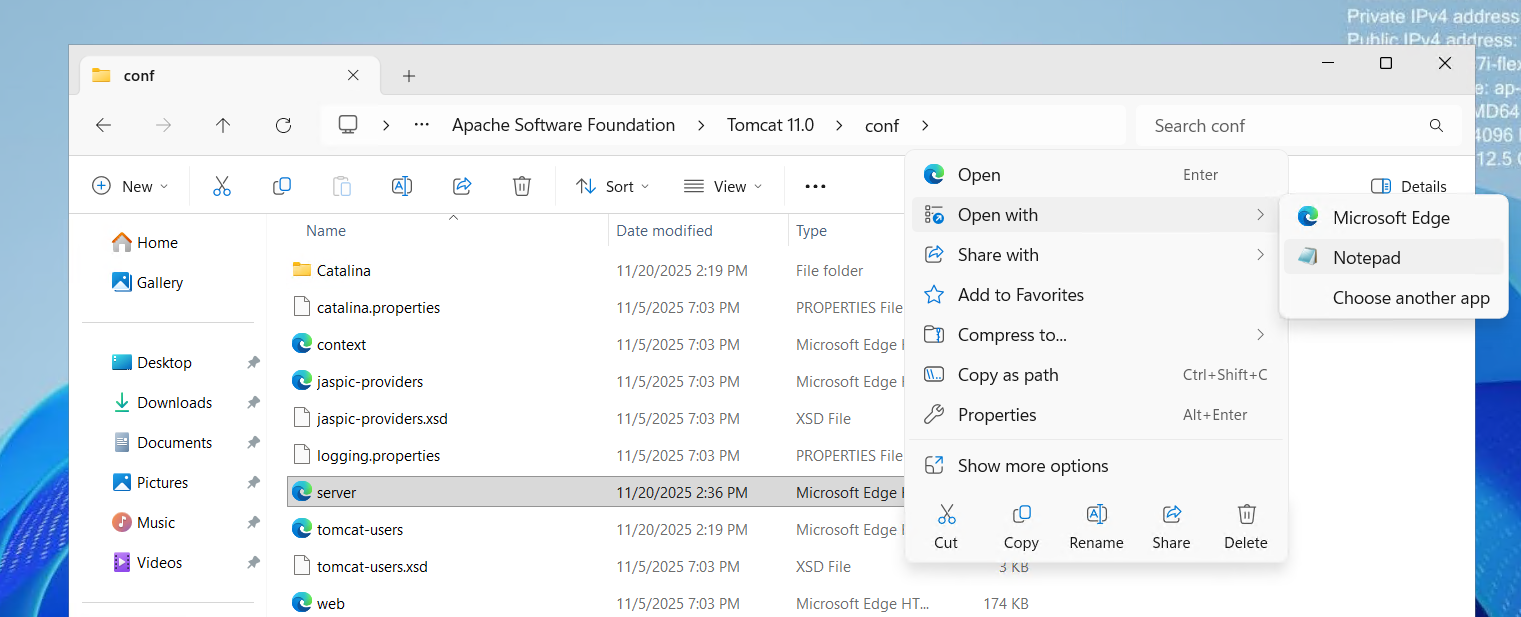
Install Apache Tomcat by downloading the .exe program from the official apache tomcat website and launch the installer



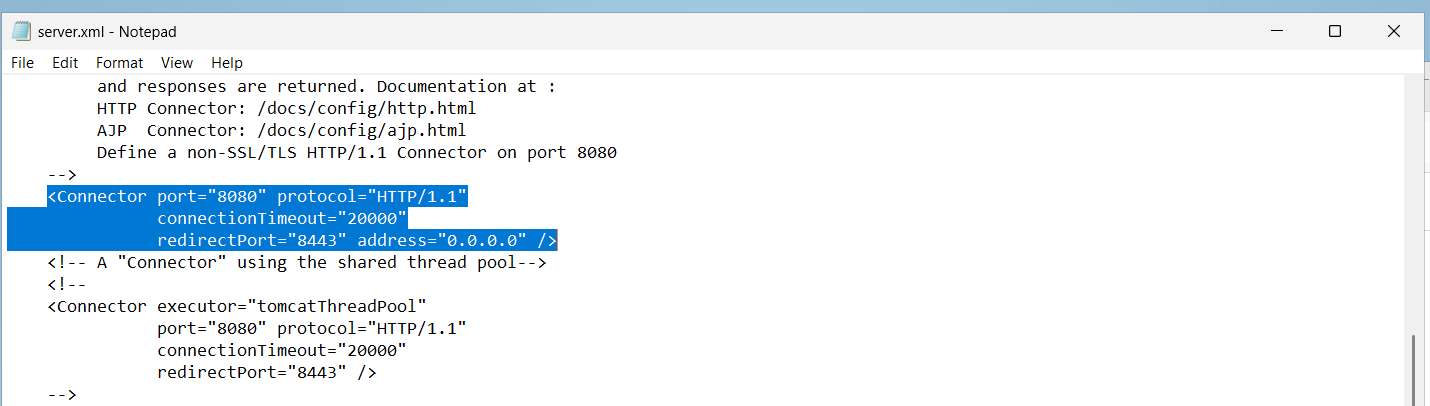
After succesful installation, go to the localhost:8080 to access the tomcat on the windows browser as follows



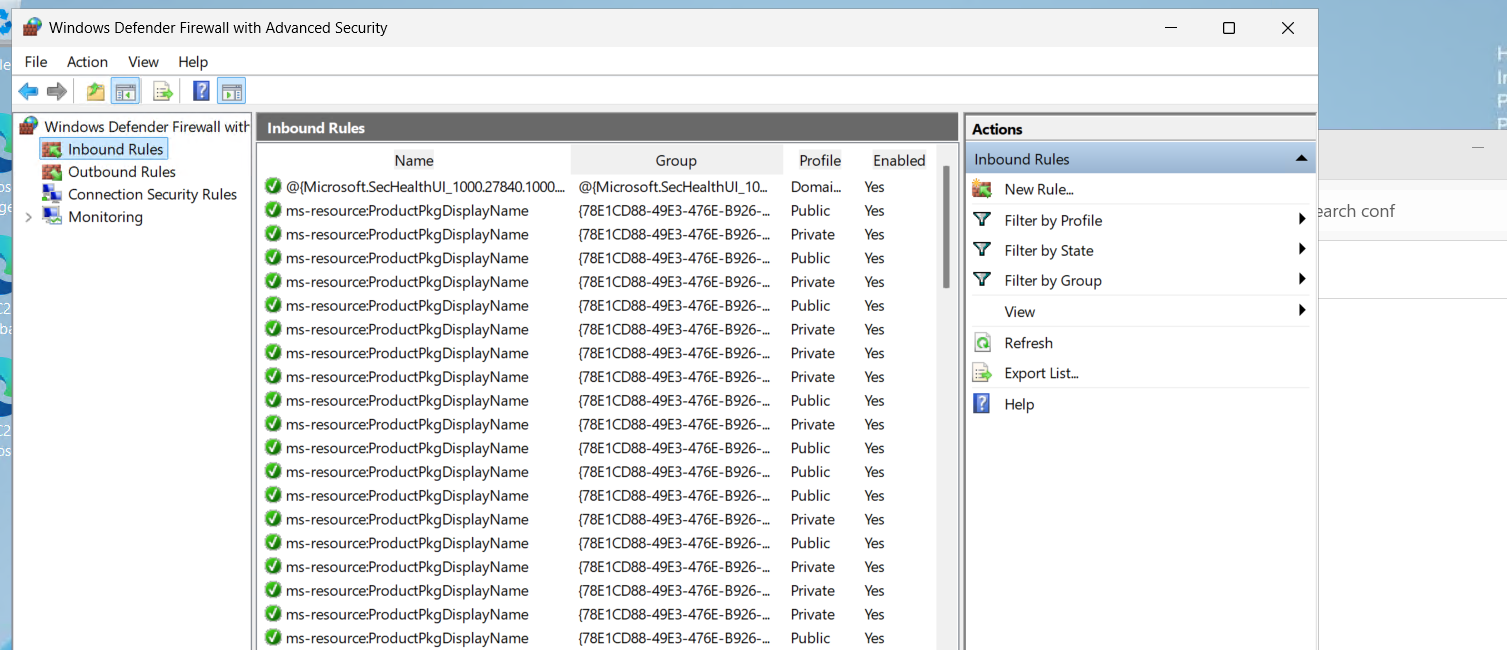
Now want this tomcat server to be able to access via internet so we need to configure the configuration file server.xml, open the file in notepad and enter the text as follows in the designated block



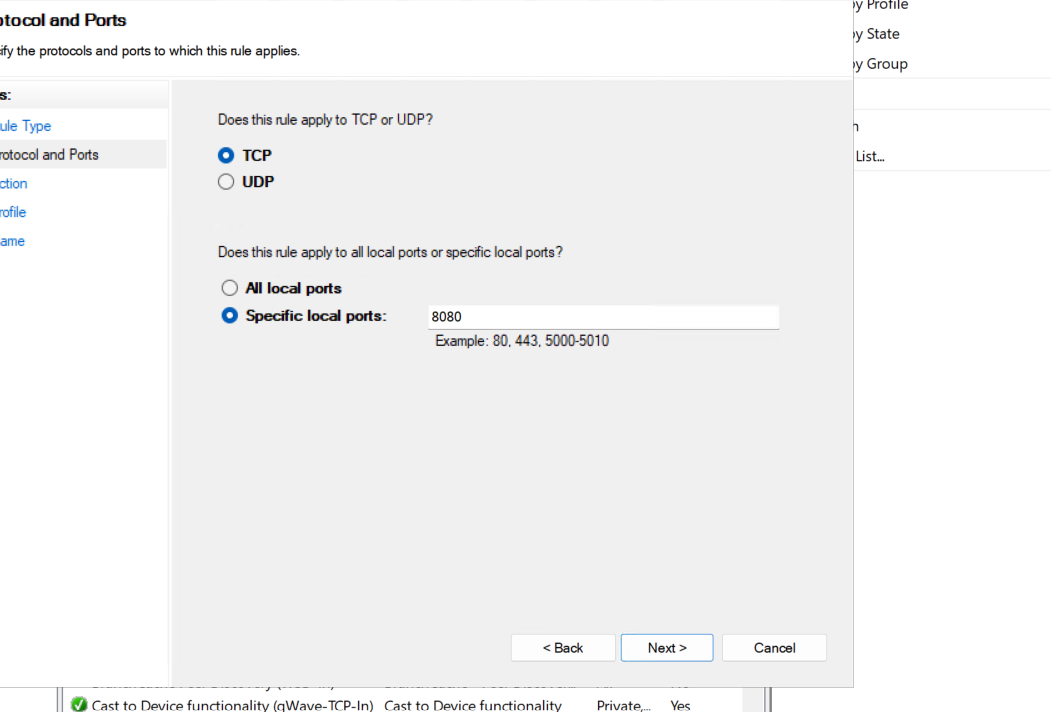
<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" address="0.0.0.0" />

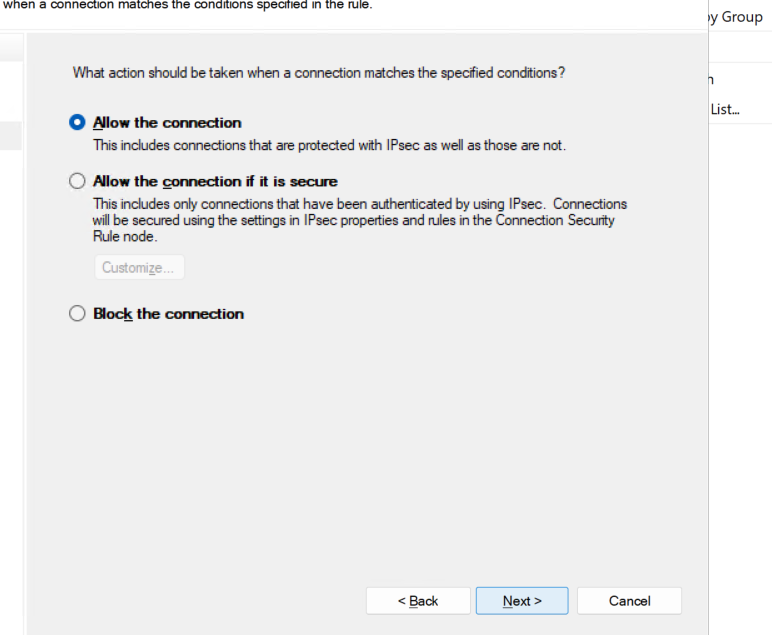


Now go to firewall settings, add new rule in inbound rules

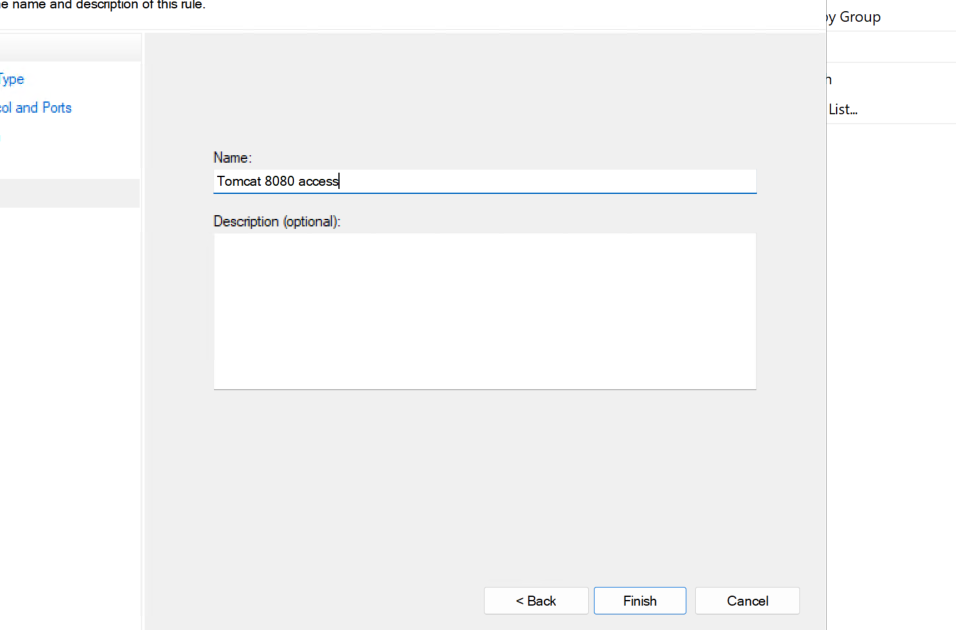


Add specific port to 8080

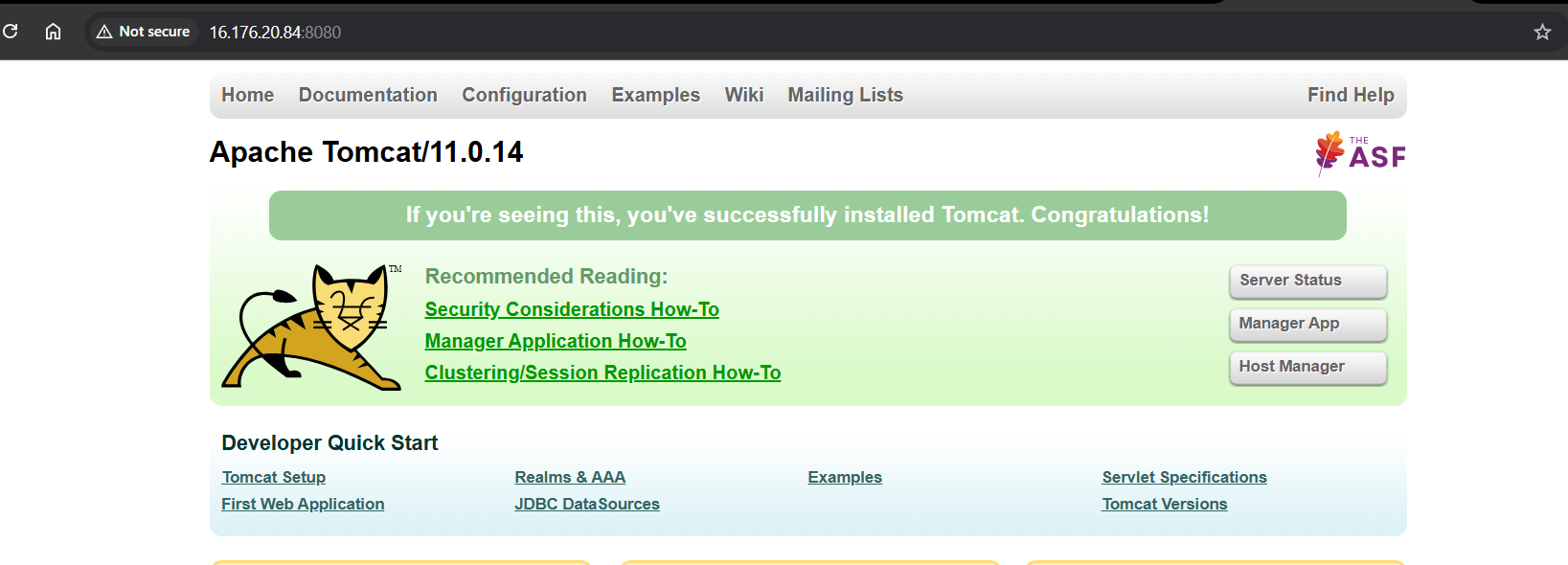




Allow it for all the connections



Give any name I have given name “Tomcat 8080 access:



Now I went to the local machine browser and accessed URL ‘<public\_ip\_address:8080”

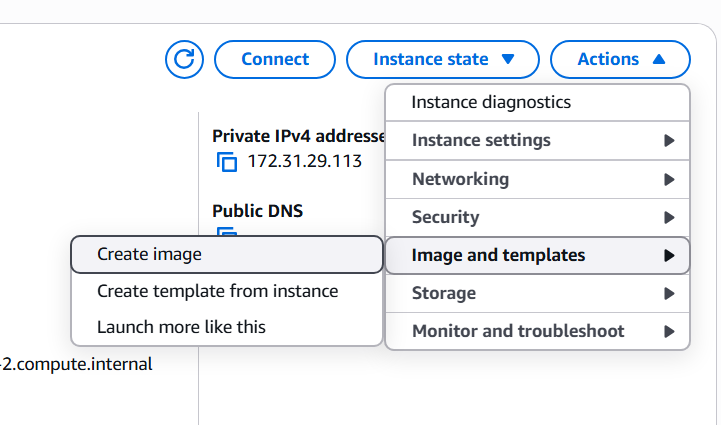
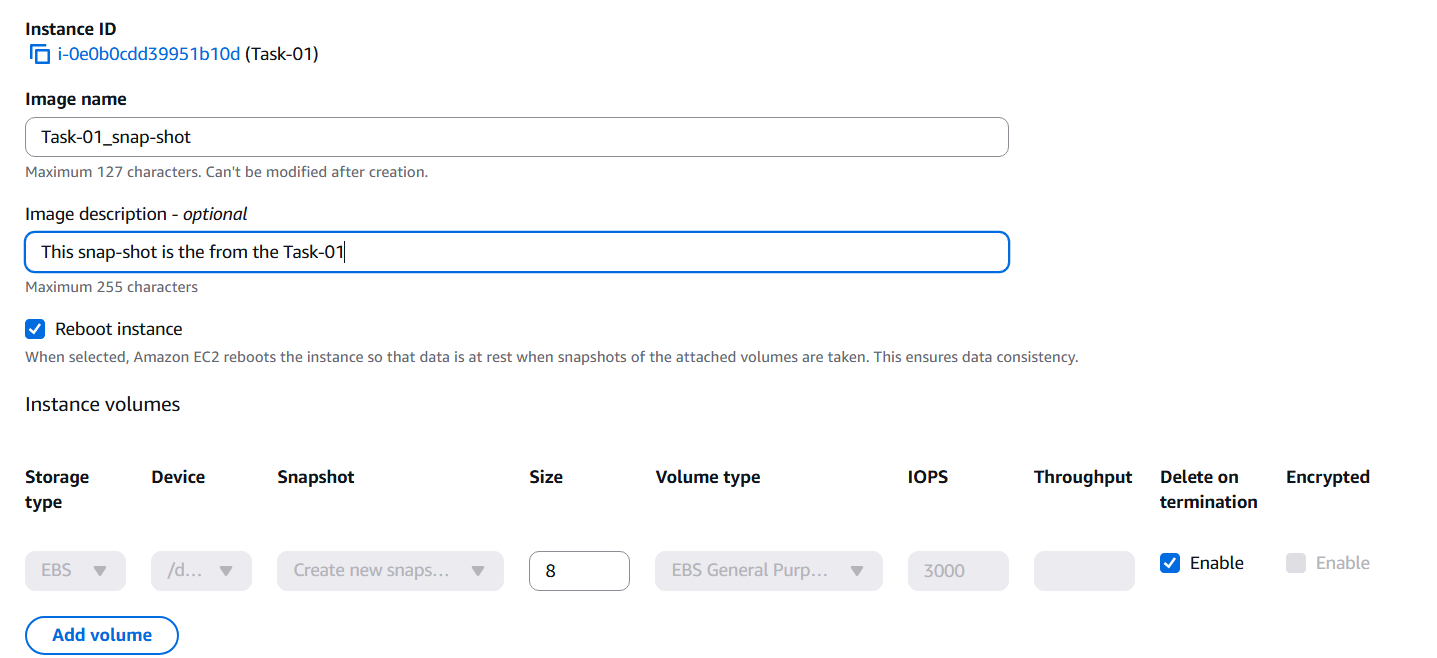
Now we can able to access the tomcat on remote browser.

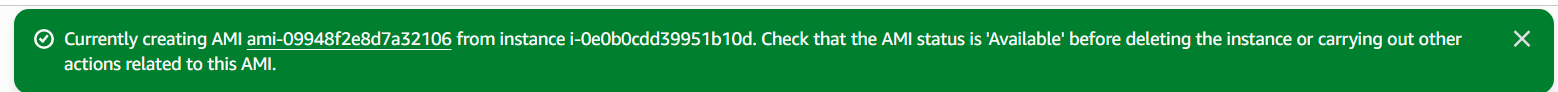
1. Take a snapshot of the instance created in Task 1.

After the instance is launched go to the

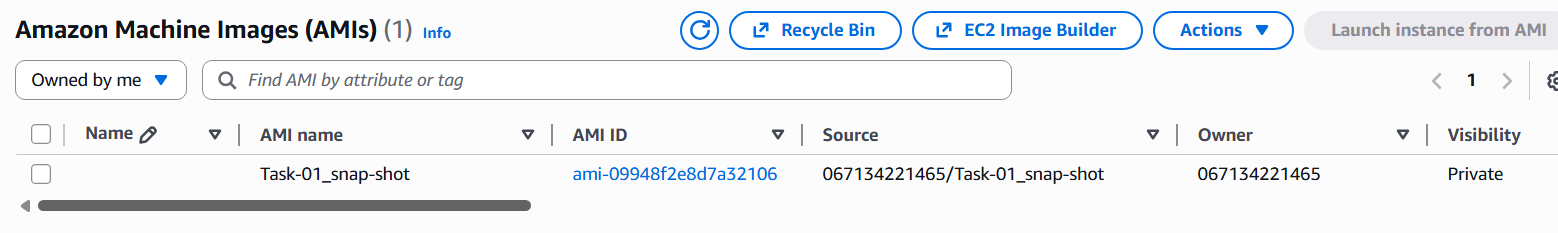
Actions->image and templates->create image

Give the image name and also description

After pressing the create image we will see the following on the top 

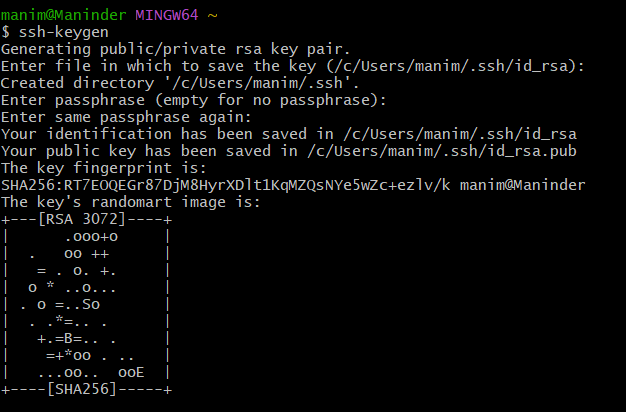
Now goto My AMI section to see the created snapshot



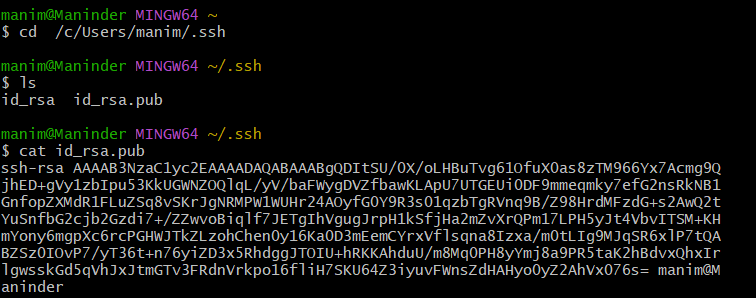
1. Assign passwordless authentication for the EC2 created in Task 2.

On windows Local machine to the gitbash and create ssh keys using the the command ‘ssh-keygen’

This command will create two keys in the windows user home directory in the .ssh hidden folder, navigate to the folder and view the public key using cat command cat id\_rsa.pub



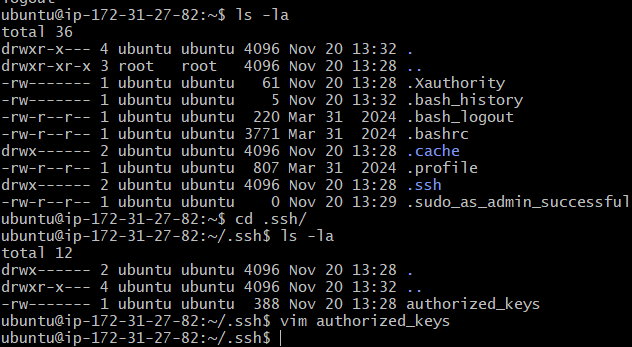
Copy the key



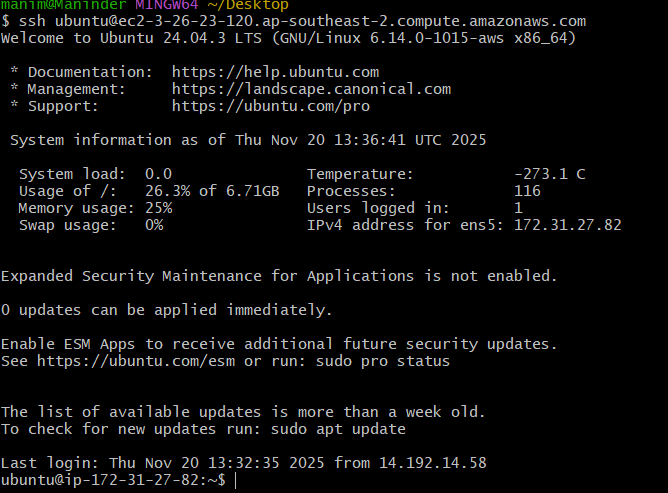
Go to the ubuntu machine from the Task-02

Navigate to the .ssh/authorisedkeys file in the UBUNTU user home folder (Not the root user since we are accessing the ubuntu user not the root user)

Paste the public key we have copied on the top of the already existed ssh key.



Go to the gitbash terminal and access the machine without any key



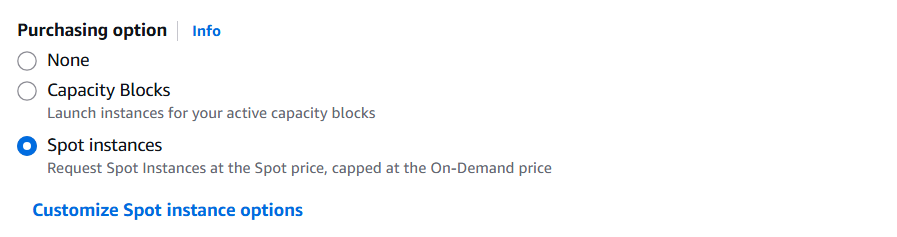
And we can see passwordless authentication is successful as above.

1. Launch any EC2 using the spot purchasing option.

Go to the aws console , choose the amazon linux along with all the requirements

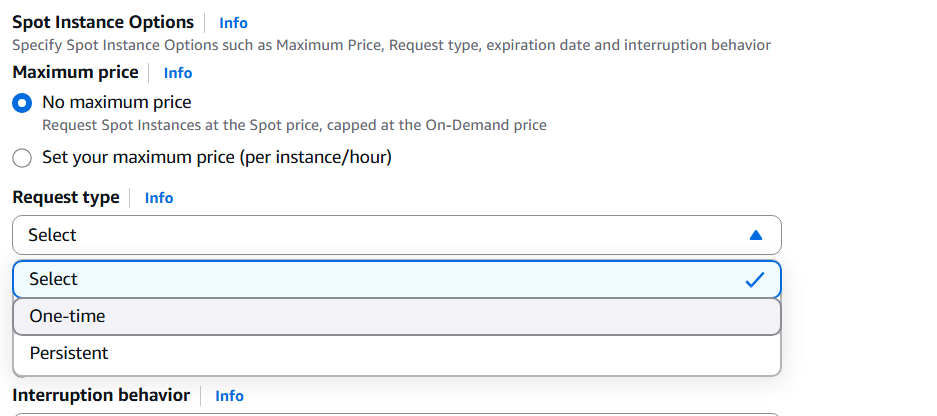
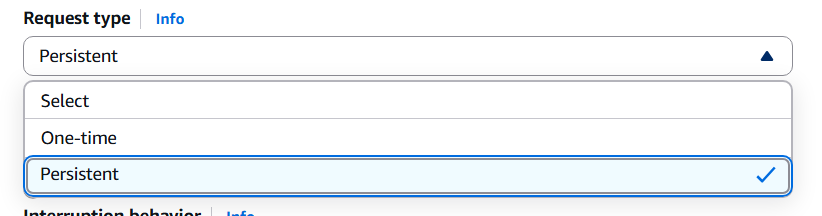
Go to the advanced details and choose spot instances at the purchasing option

Go to customize spot instances



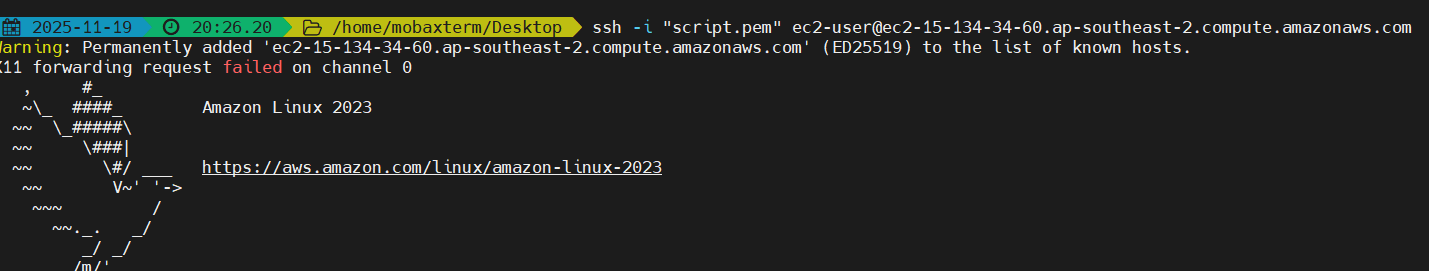
Choose request type to persistent

And choose interrupt behaviout to stop option

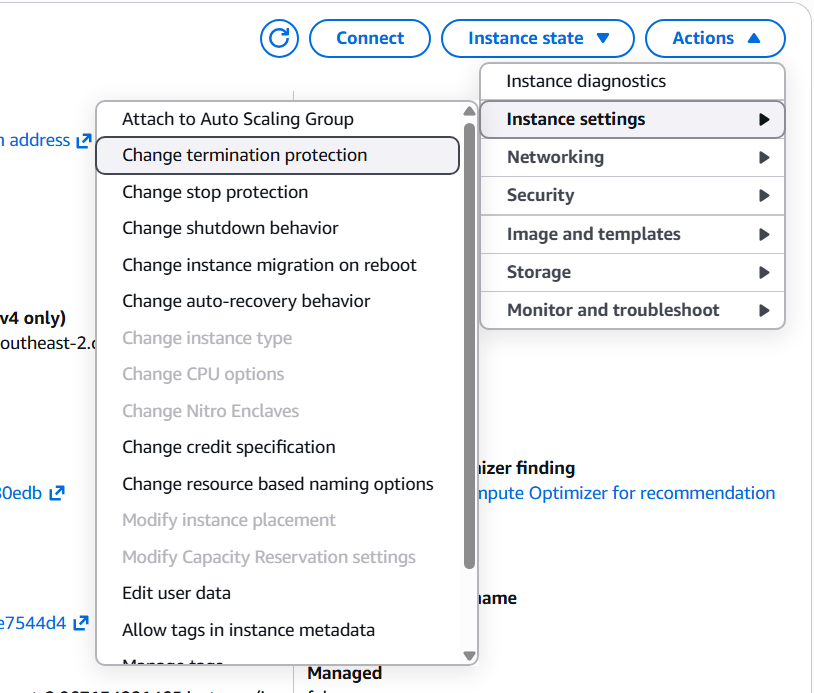
Now we can see the instance is launched, log in using the ssh from the terminal.

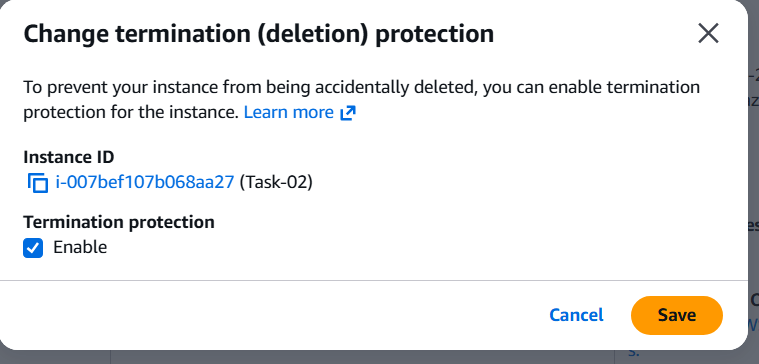




1. Enable termination policy on the EC2 created in Task 2.

Go to Actions-> instance settings-> change termination protection

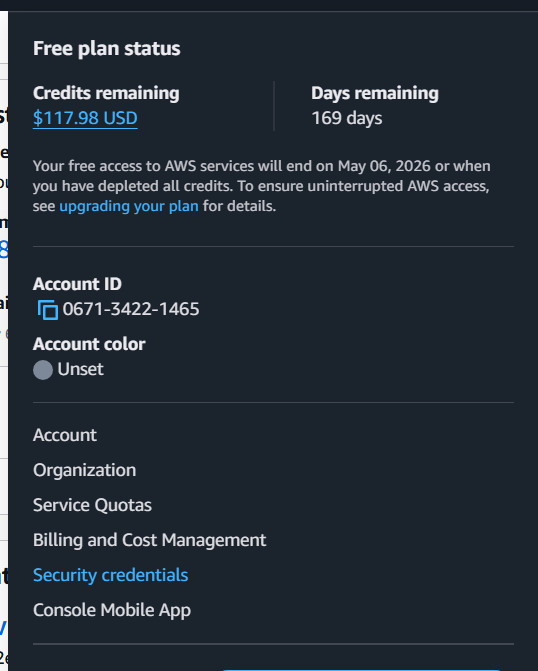


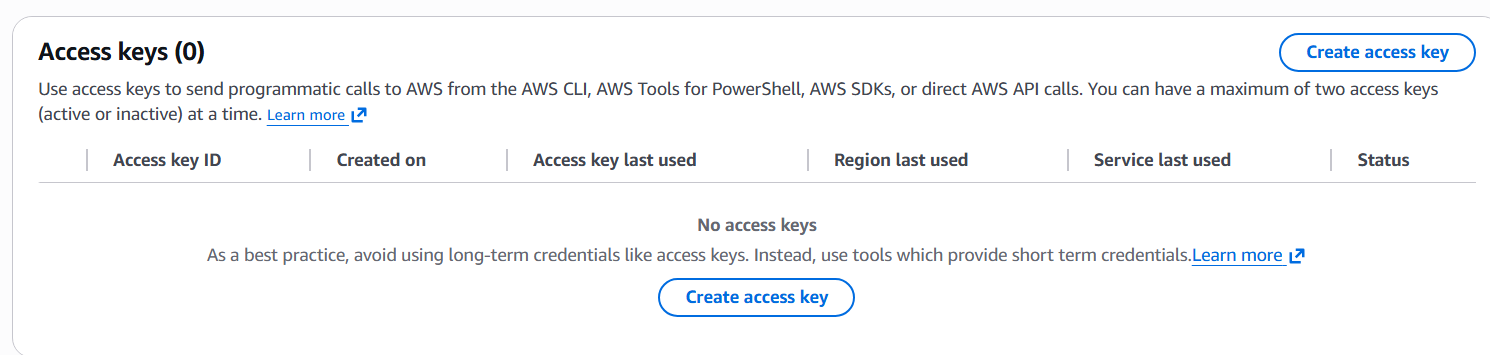


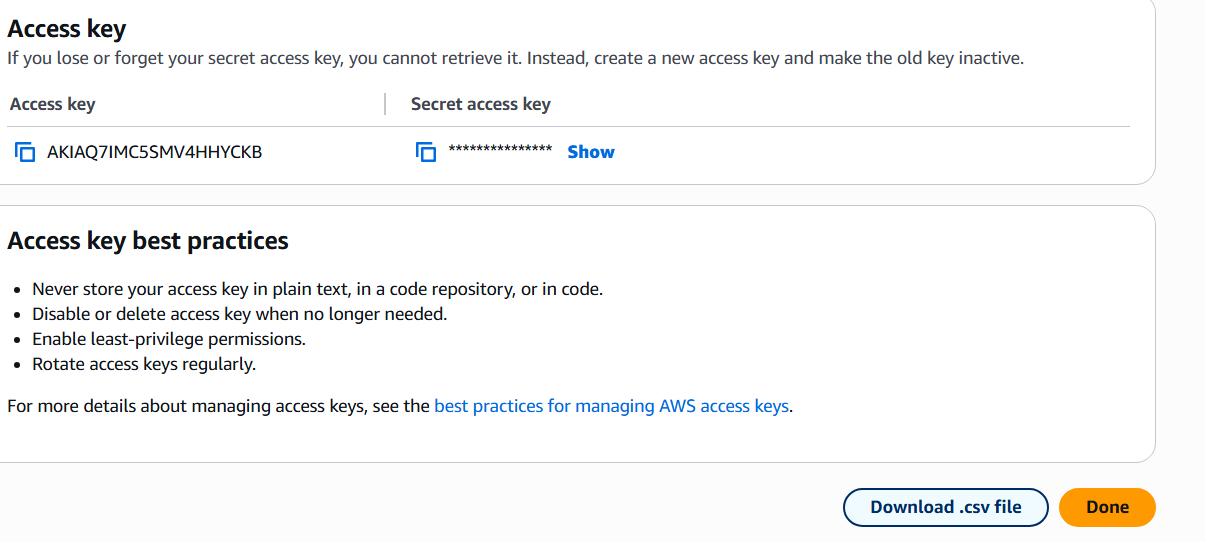
Toogle enable and press ‘save’

1. Launch one EC2 using AWS CLI.

Go to user option on the top right corner and choose ‘security’



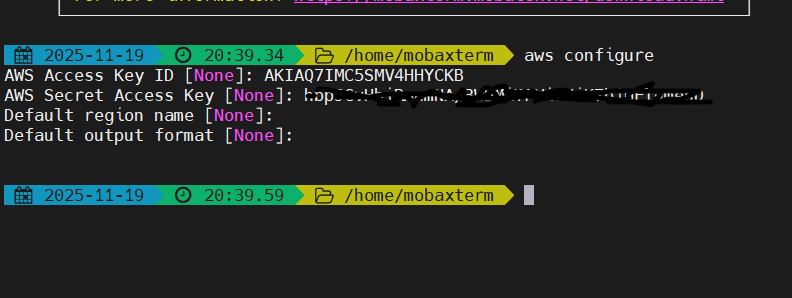


Choose create access key 

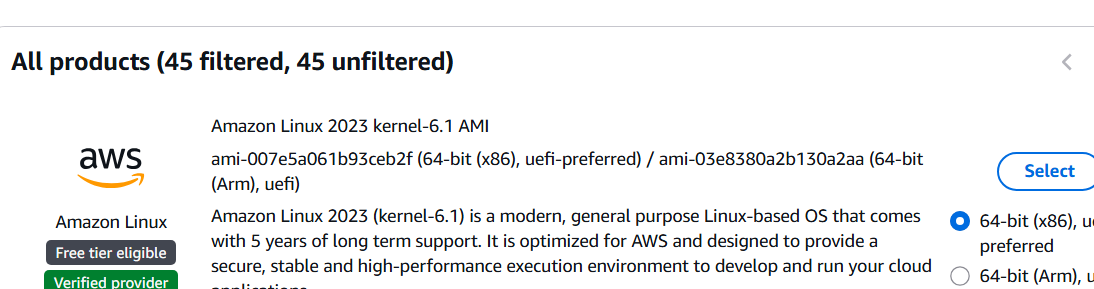
Download access key and secret key

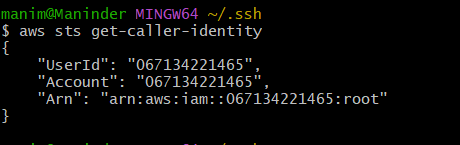
Now go to moboxterm and configure aws using ‘aws configure’

And enter access key and secret key details



Go to amazon instances and copy the ami id of the os you want to launch in your region



Verifying the aws configured values with “aws sts get-caller-identity” 

Set the region to sydney by using “export AWS\_DEFAULT\_REGION="ap-southeast-2"”

Now entered the following command which copied values of os smi, subnet details, pem file and other details

aws ec2 run-instances \

--image-id ami-007e5a061b93ceb2f \

--instance-type t3.micro \

--key-name script \

--security-group-ids sg-04775dd623de7e5dc \

--subnet-id subnet-0272ba8f6d87646fd \

--count 1 \

After executing the above command, I got the following output.



I went to the aws console to check if the server is provisioned or not and found the following.

