Task on Ec2:  
==================

**1) Launch one ec2 using Amazon Linux 2 image and add script in user**

**data to install Apache.**

* Open aws console
* Select launch instance and name ,now we are working on amazon linux2 and name it as Apacheserver
* Select AMI image as linux 2
* Instance type as t3.micro
* Choose an existing one or create a new key pair
* Allow **HTTP (port 80)** and **SSH (port 22)** in the security group.
* Finally write bash script in user data which is in advanced options then create instance
* **#!/bin/bash**

**yum update -y**

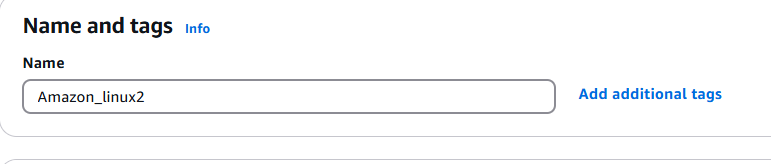
**yum install -y httpd**

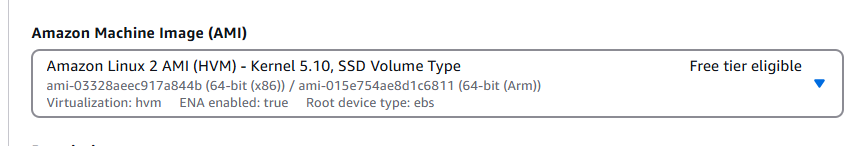
**systemctl start httpd**

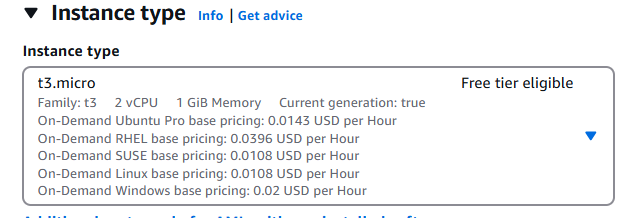
**systemctl enable httpd**

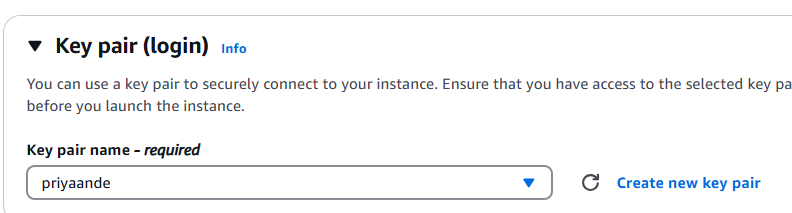
**echo "<h1>Hello from EC2 - Apache installed using User Data!</h1>" > / /var/www/html/index.html**

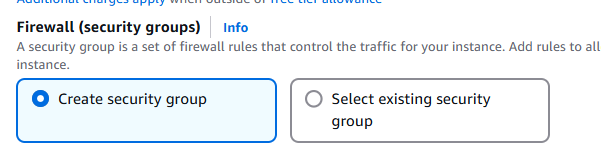
* Then click on create instance
* Copy ip address of instance and paste on web page you can see httpd is running

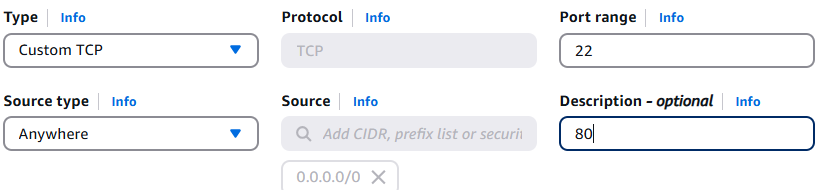


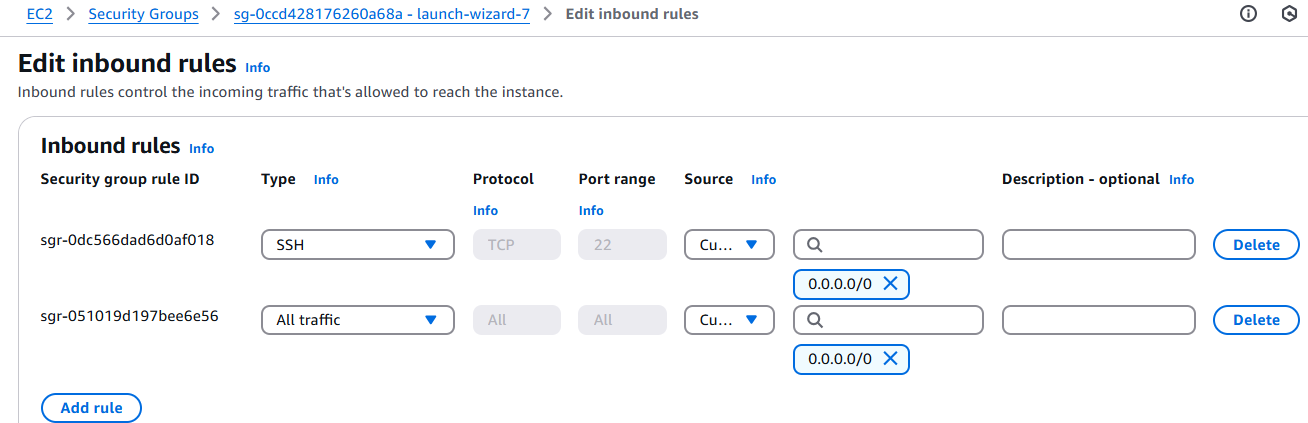




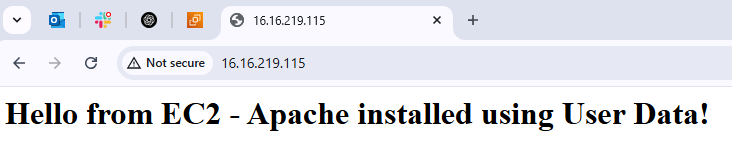












1. **Launch one ec2 using Ubuntu image and add script in user data to install Nginx.**

* Open aws console
* Select launch instance and name ,now we are working on ubuntu image and name it as nginx server
* Select AMI image as Ubuntu
* Instance type as t3.micro
* Choose an existing one or create a new key pair
* Allow ngnix**(port 80)** and **SSH (port 22)** in the security group.
* Finally write bash script in user data which is in advanced options then create instance

**#!/bin/bash**

**apt update -y**

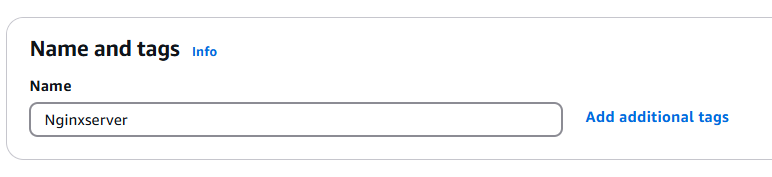
**apt install nginx -y**

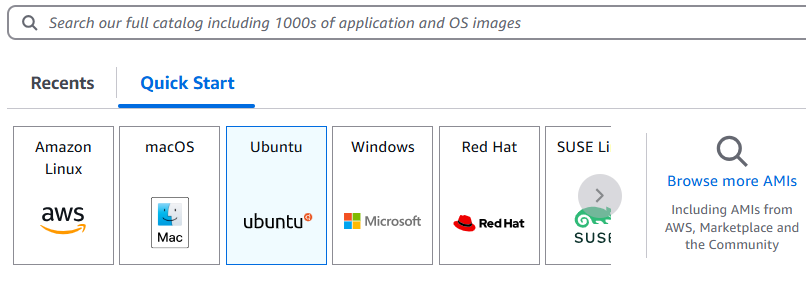
**systemctl start nginx**

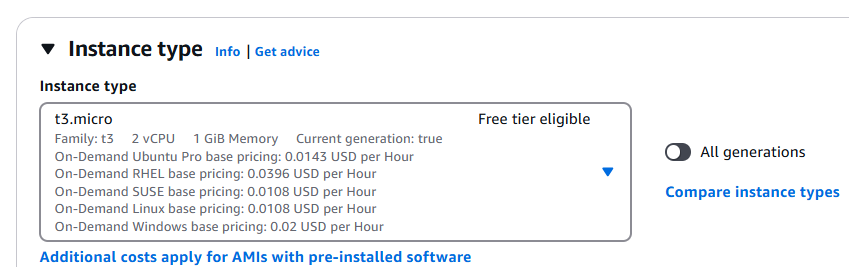
**systemctl enable nginx**

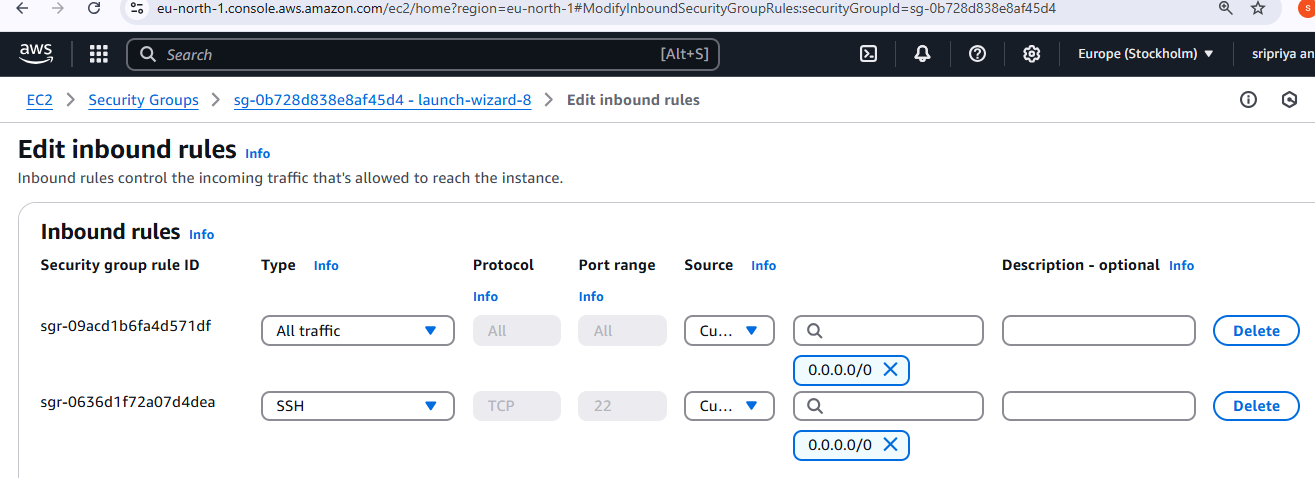
**echo "<h1>Hello from EC2 - Nginx installed using User Data on Ubuntu!</h1>" > /var/www/html/index.nginx-debian.html**

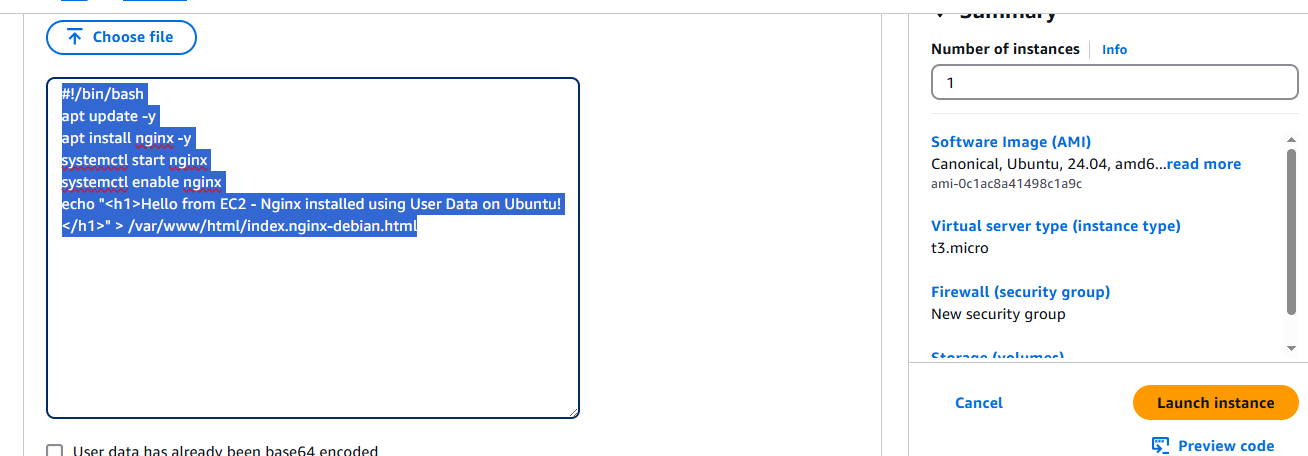
* Then click on create instance
* Copy ip address of instance and paste on web page you can see Ngnix is running

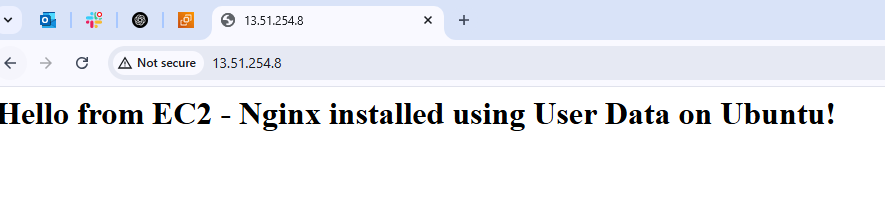












1. **Launch one windows server and install tomcat in windows.**

* Go to Aws console and launch one ec2 instance
* Name it as windows\_server
* Select **Microsoft Windows Server 2019 Base in AMI**
* Choose t3.micro
* Select “Create new key pair”.
* Set name (e.g., windows-keypair).
* Choose .pem format.
* Download and save the .pem
* Select or create a new Security Group.

Add the following Inbound Rules:

Port Protocol Purpose

3389 RDP Remote Desktop

8080 TCP Tomcat Access

80 HTTP Web Access

* Launch instance
* Get Administrator Password:

Select the instance → Actions → Connect → Get Password.

Upload your .pem key → Click Decrypt Password.

* Download and open .rdp file → Use Administrator and the decrypted password to log in.
* Install Java (JDK):

Open Microsoft Edge inside the server.

Go to https://www.oracle.com/java/technologies/javase-downloads.html

Download and install the latest Java SE Development Kit (JDK).

Optionally, set the JAVA\_HOME environment variable (via System Properties → Environment Variables).

* Install Apache Tomcat:

Go to https://tomcat.apache.org

Download the Windows Service Installer.

* During install:

Provide JDK path.

Leave port 8080 as default.

Complete installation

* Start Tomcat Service:

Go to Services → Find Apache Tomcat → Start the service.

* Open Windows Defender Firewall:

Go to Inbound Rules → New Rule:

Type: Port

Port: 8080

Allow the connection

Apply to Domain, Private, Public

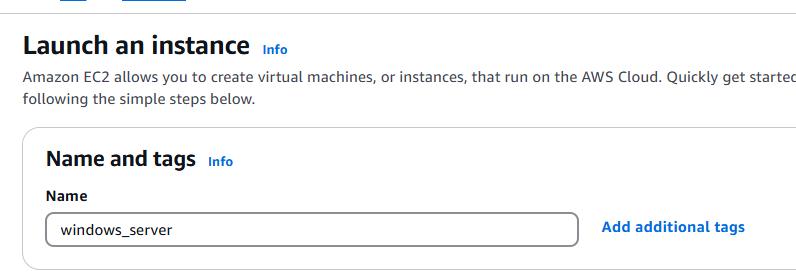
Name: Tomcat 8080

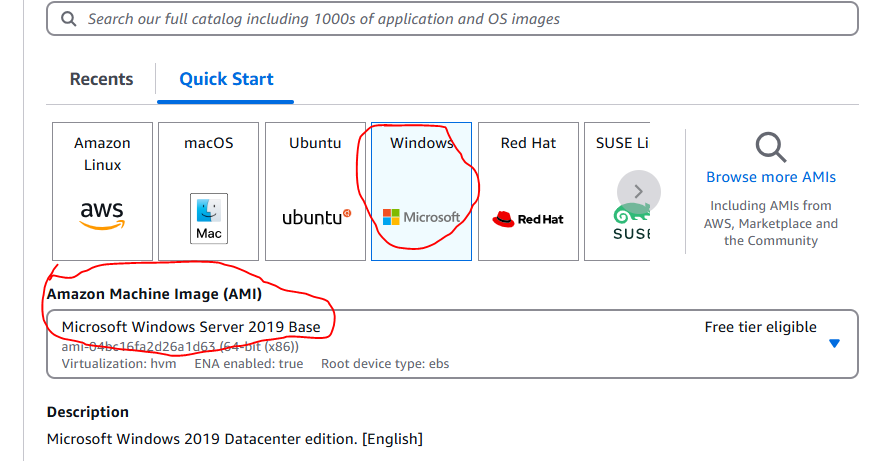
* Open browser inside the server:

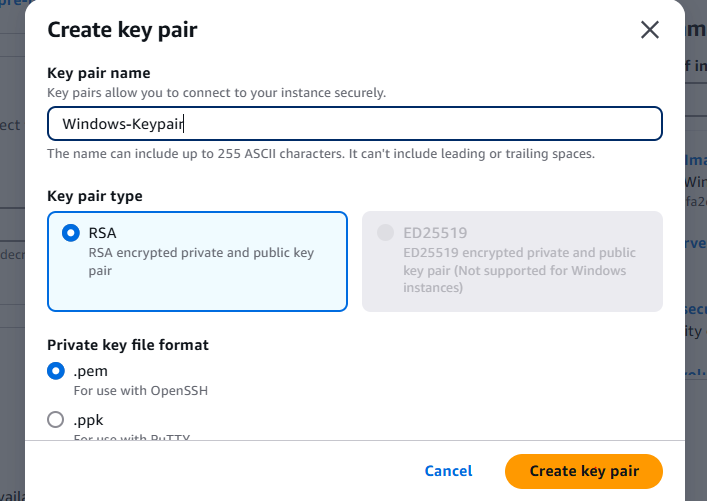
<http://localhost:8080>

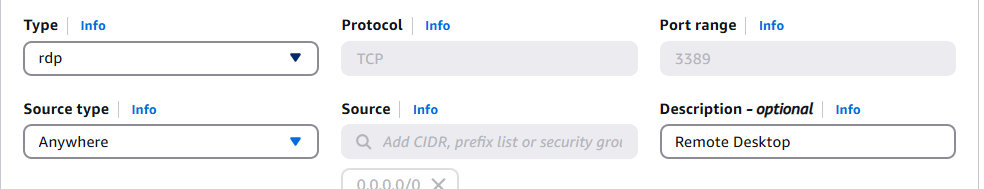
* Open on your own browser:

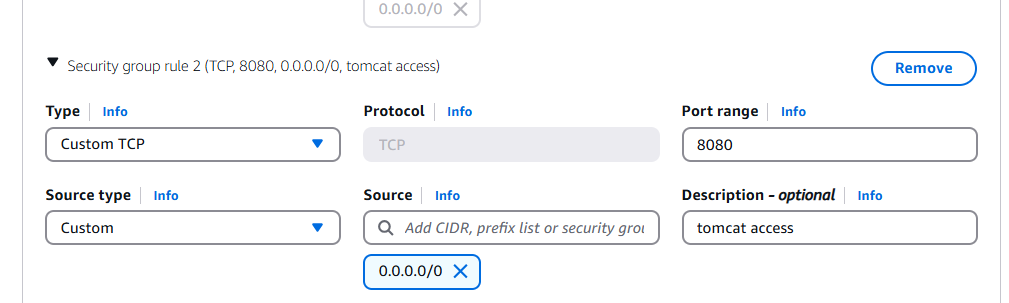
http://<public-ip>:8080

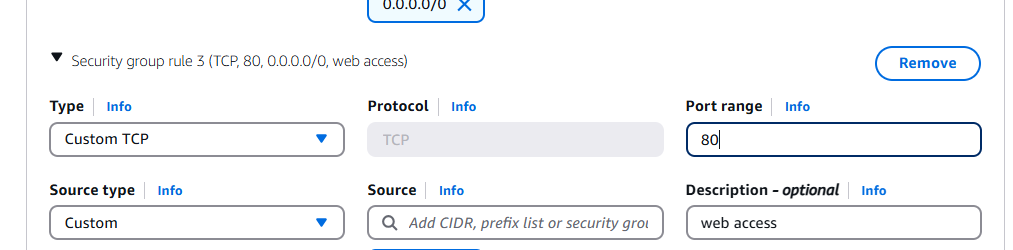


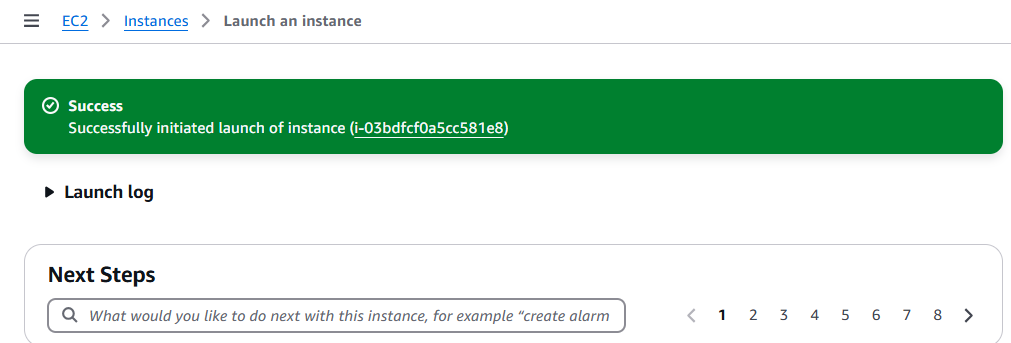


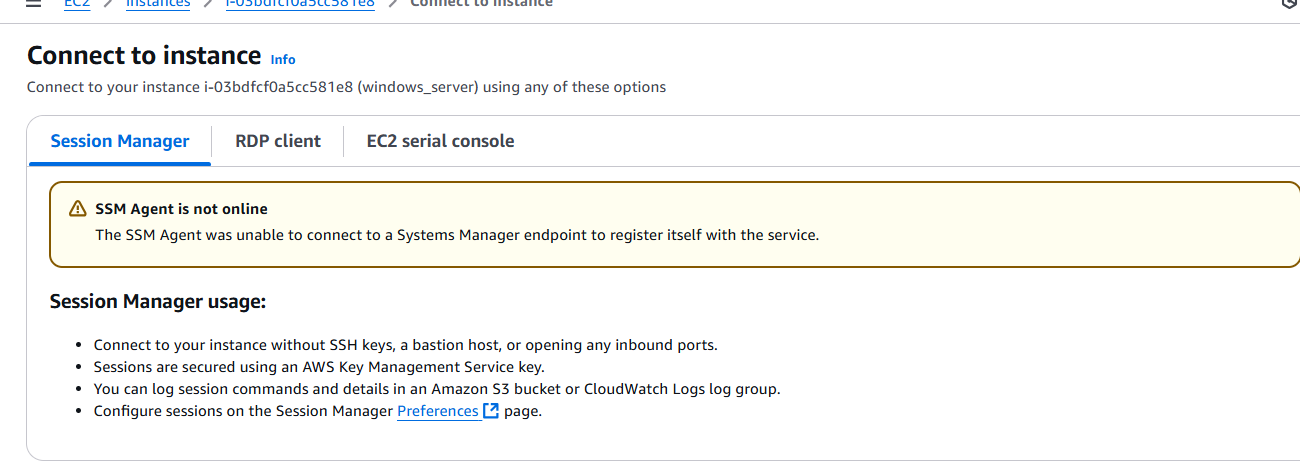


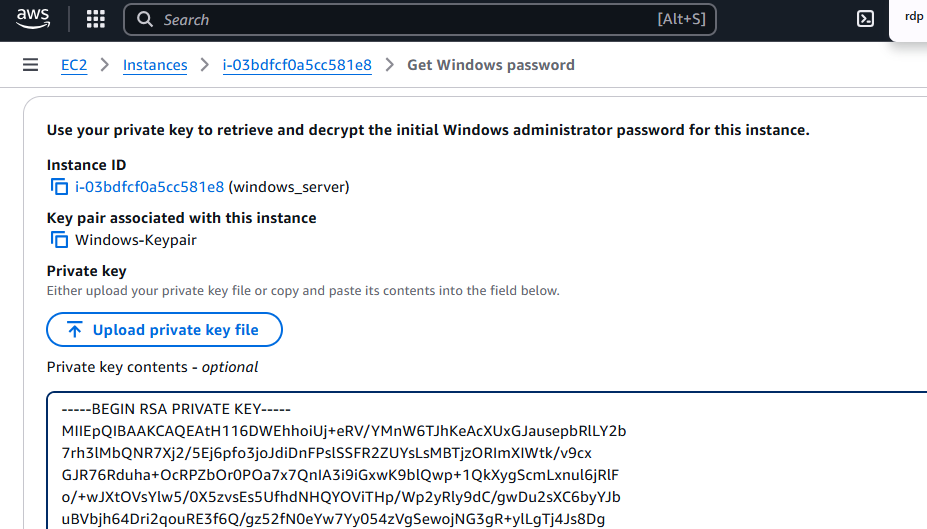


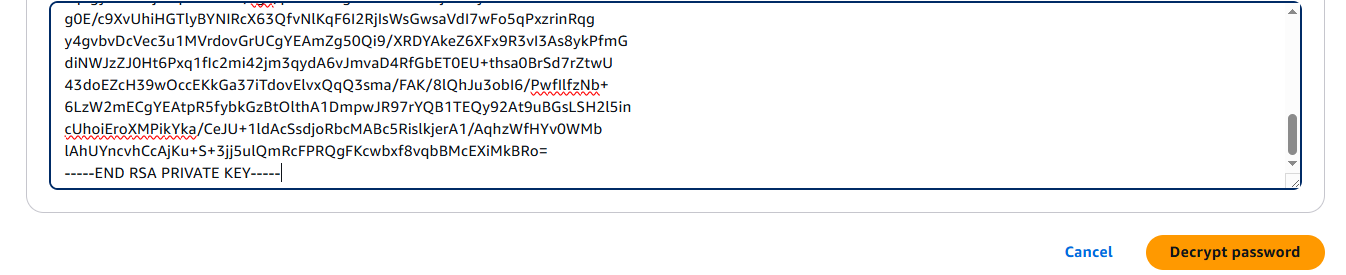


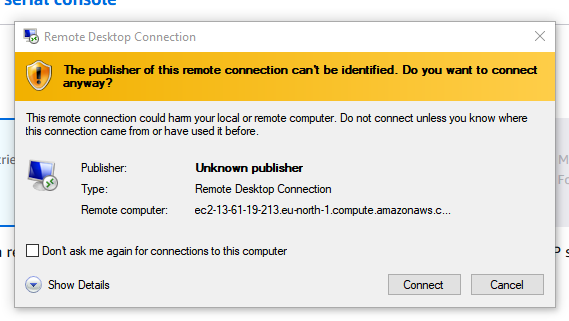


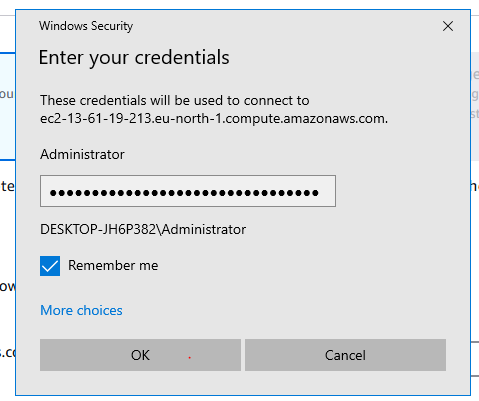


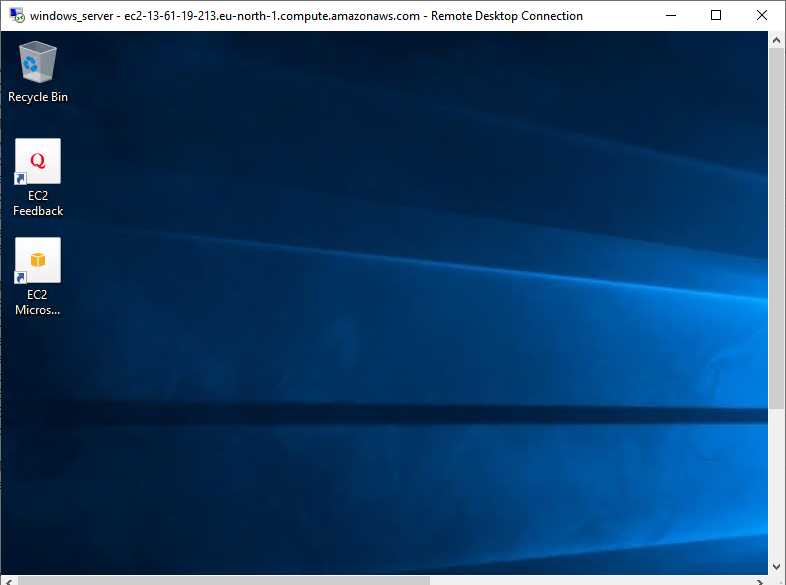


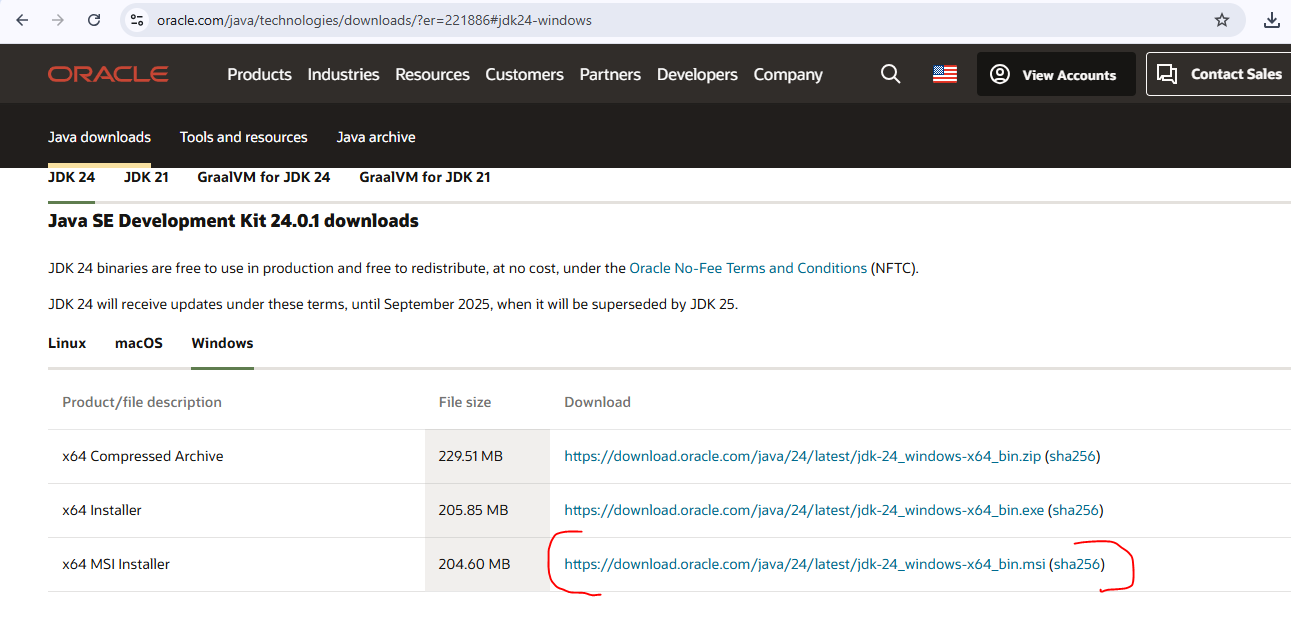


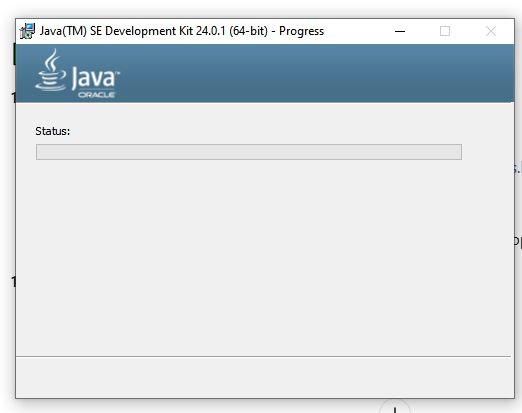


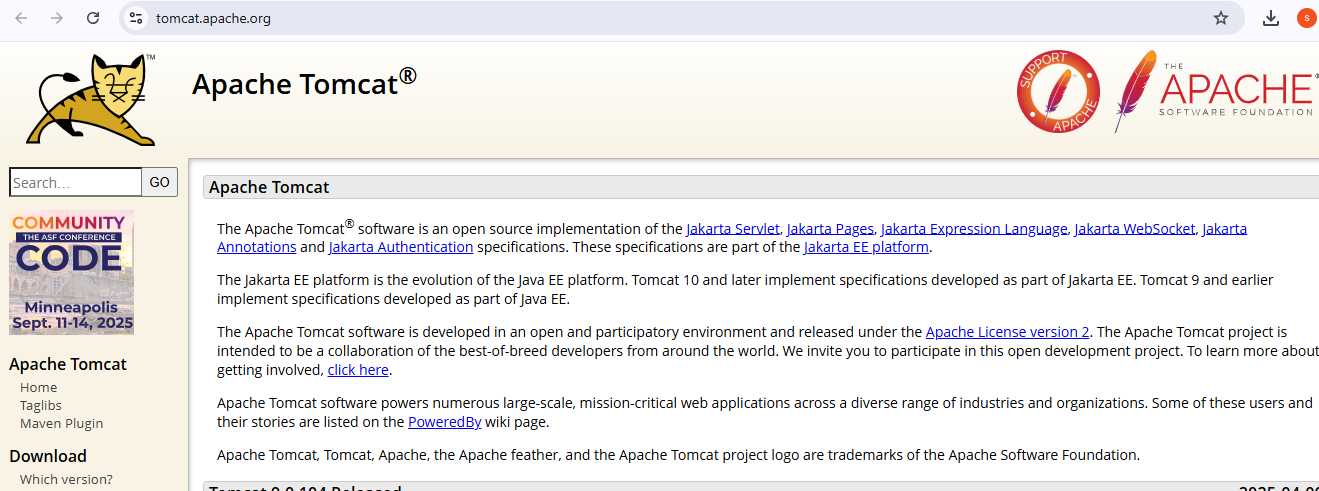


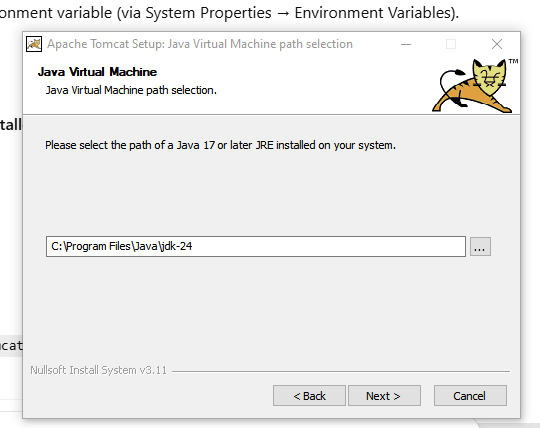


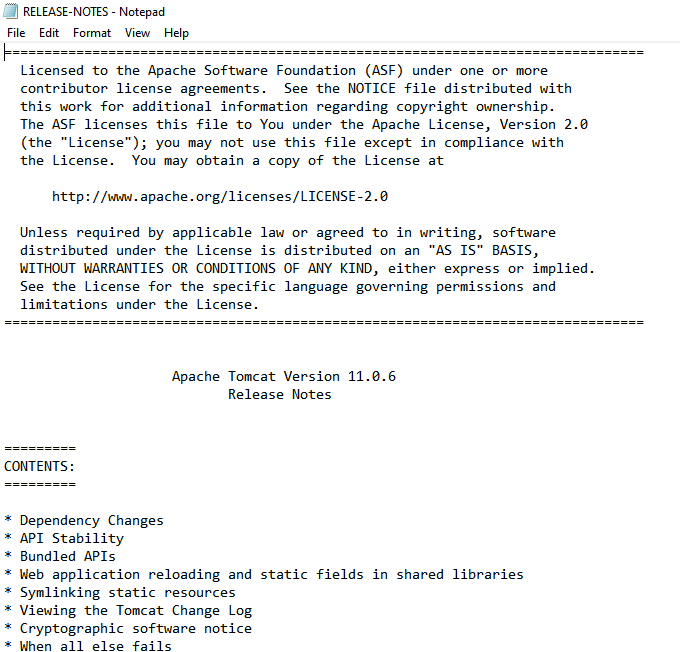


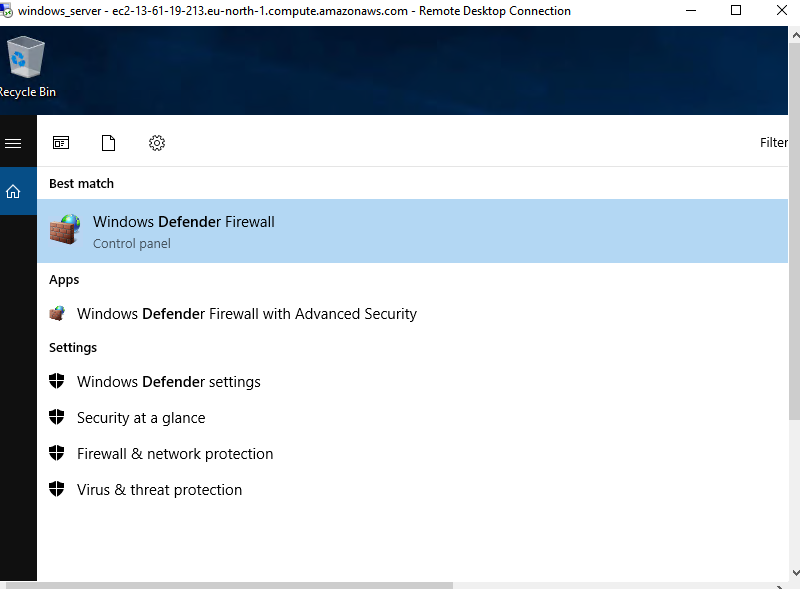


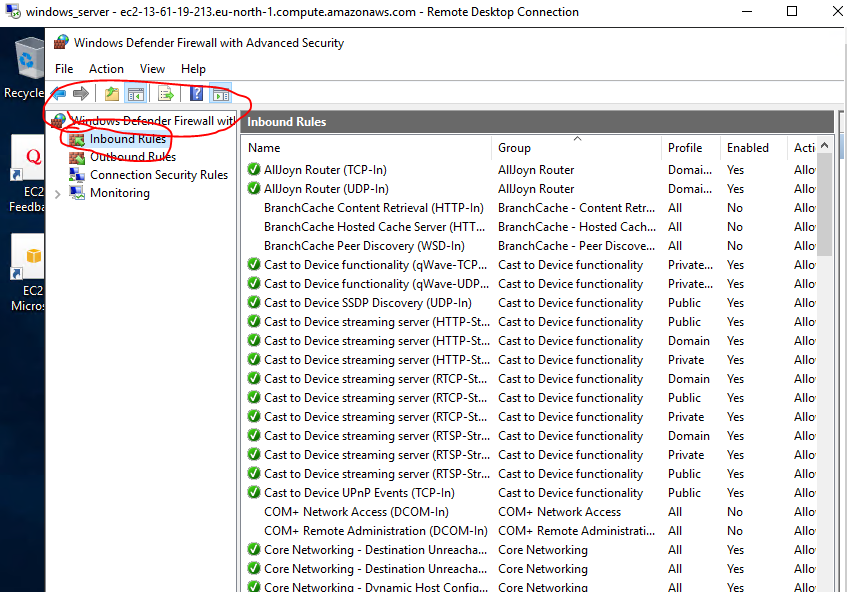


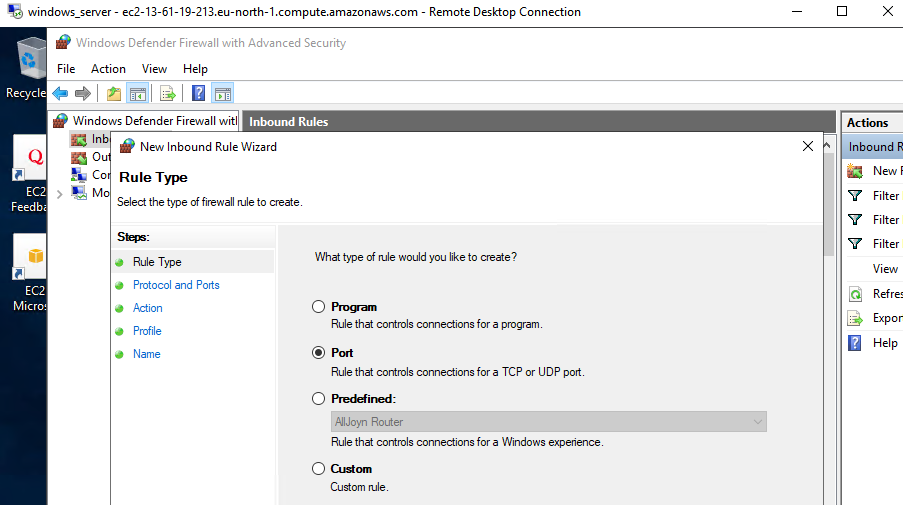


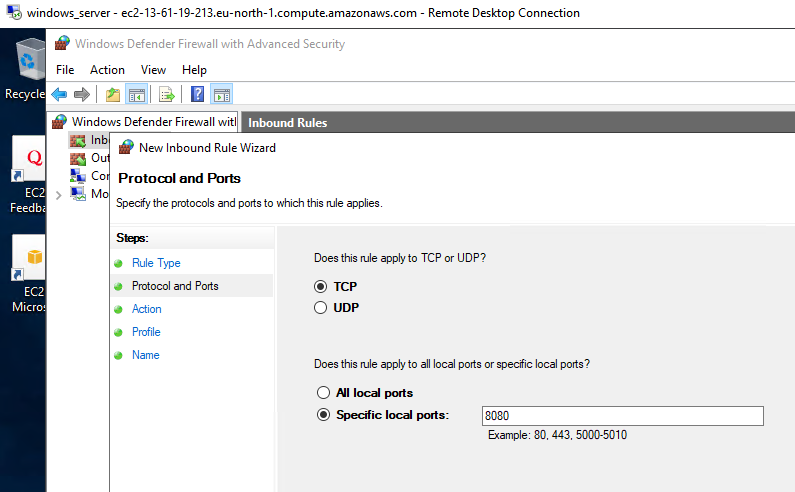


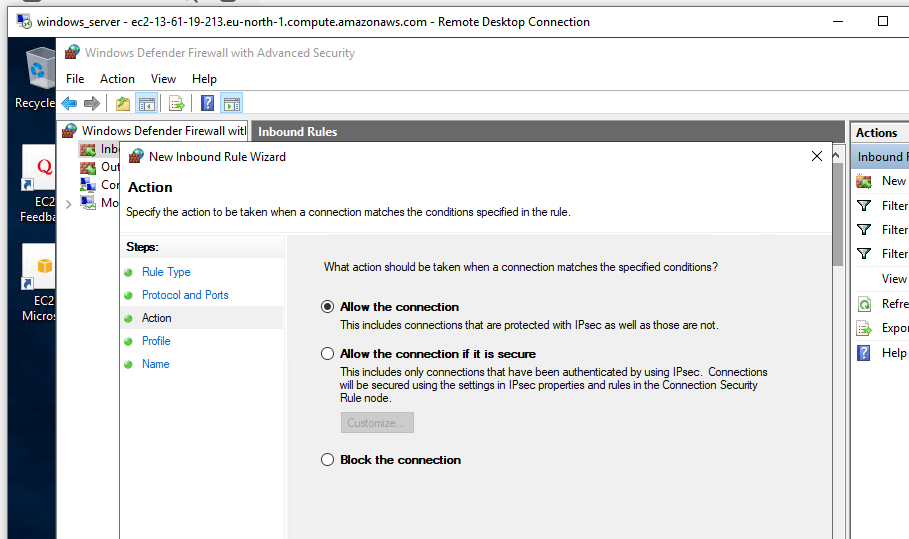


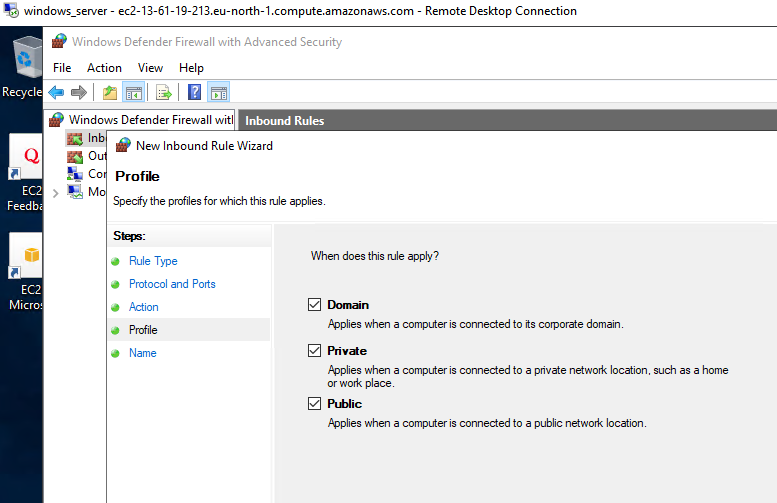


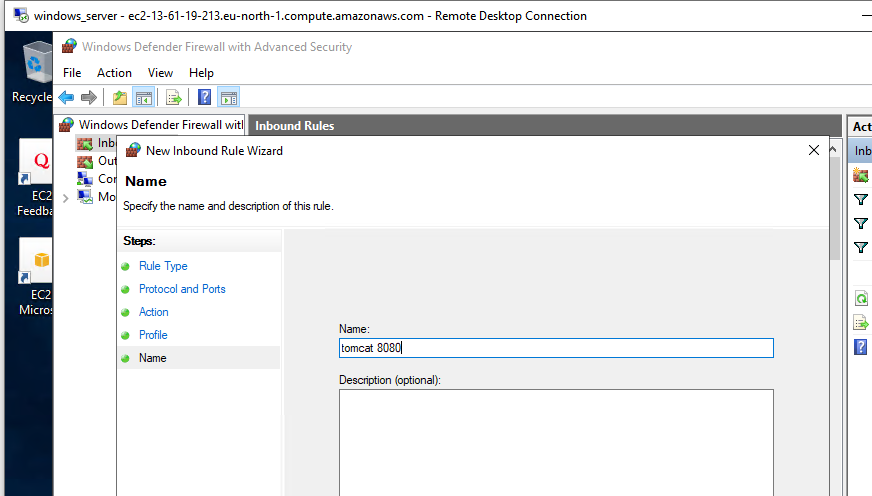


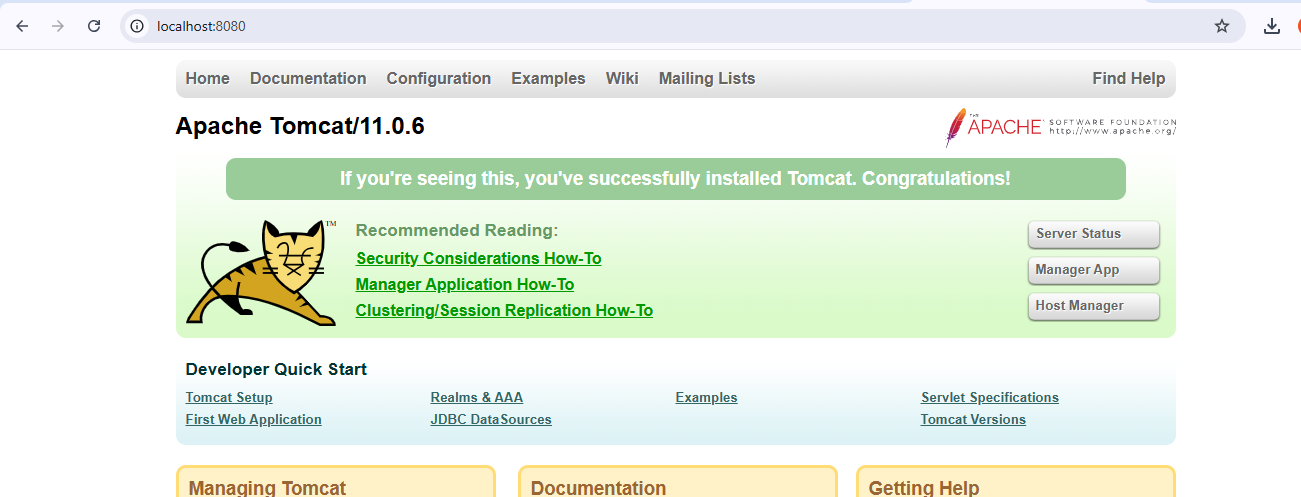






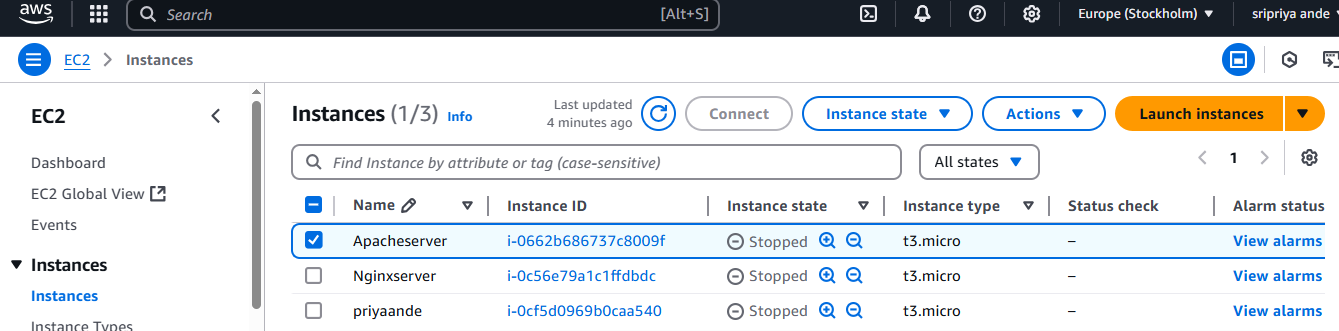


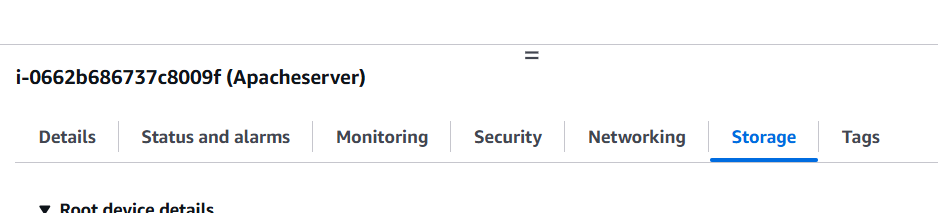


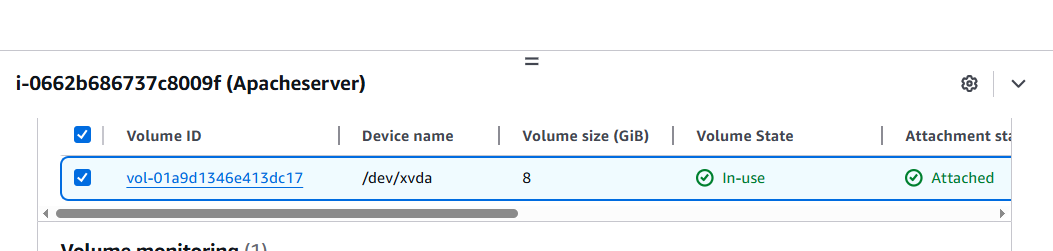


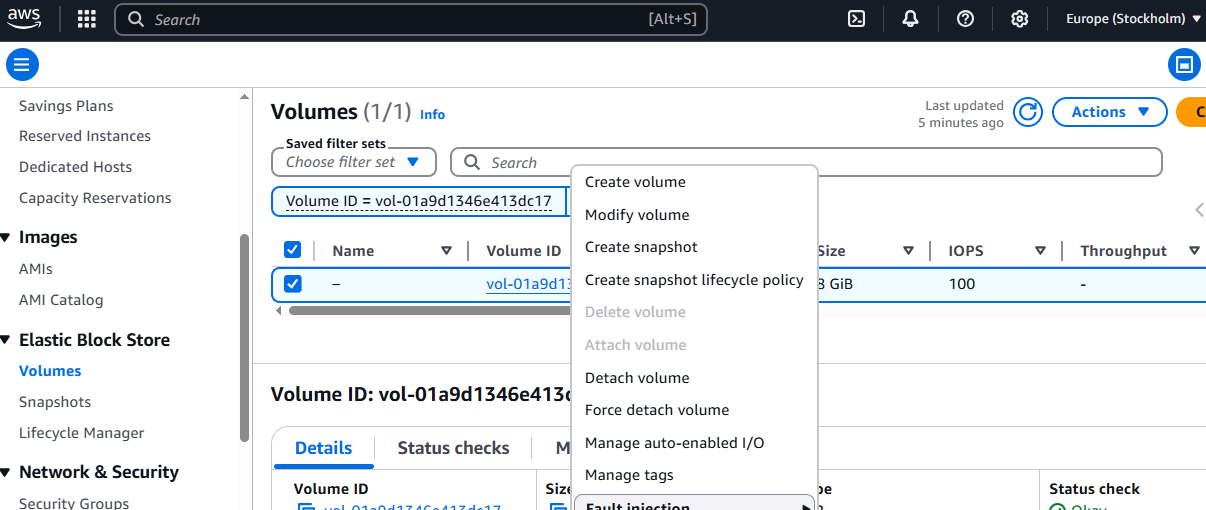
1. **Take snapshot of the instance created in Task 1.**

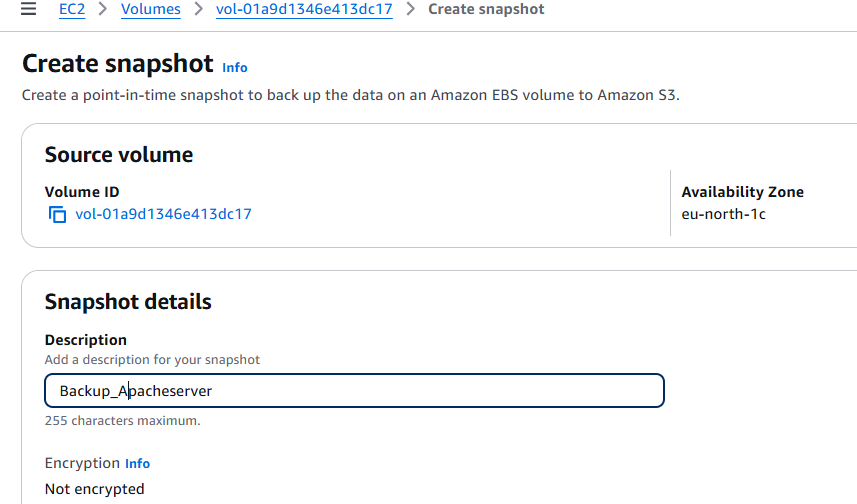
* Go to instance and click on instance type and stop the running sever then click on instance you want and scroll done toy can find storage tab
* You will find volume id left click on that id open one page
* Then right click on volume id you will find some
* Click on create snapshot
* In description you can name anything (backup\_Apacheserver)
* Then click on create snapshot
* Then you can see snapshot successfully created for Apacheserver

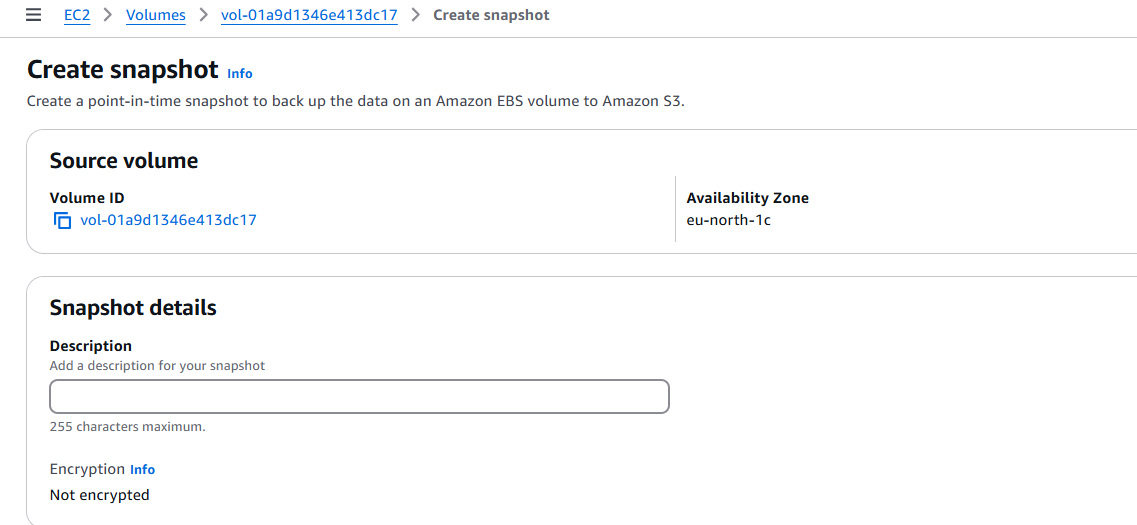


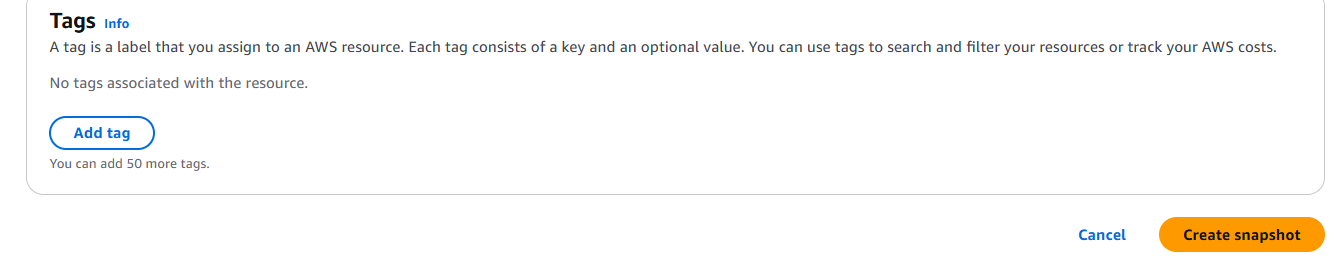


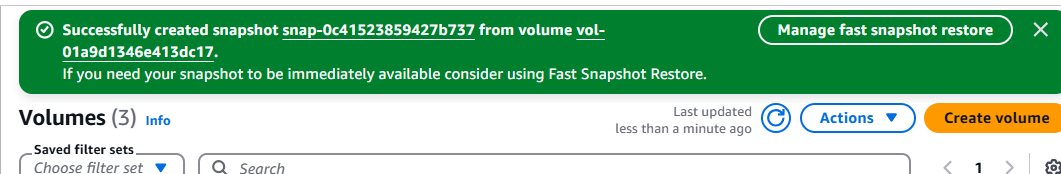






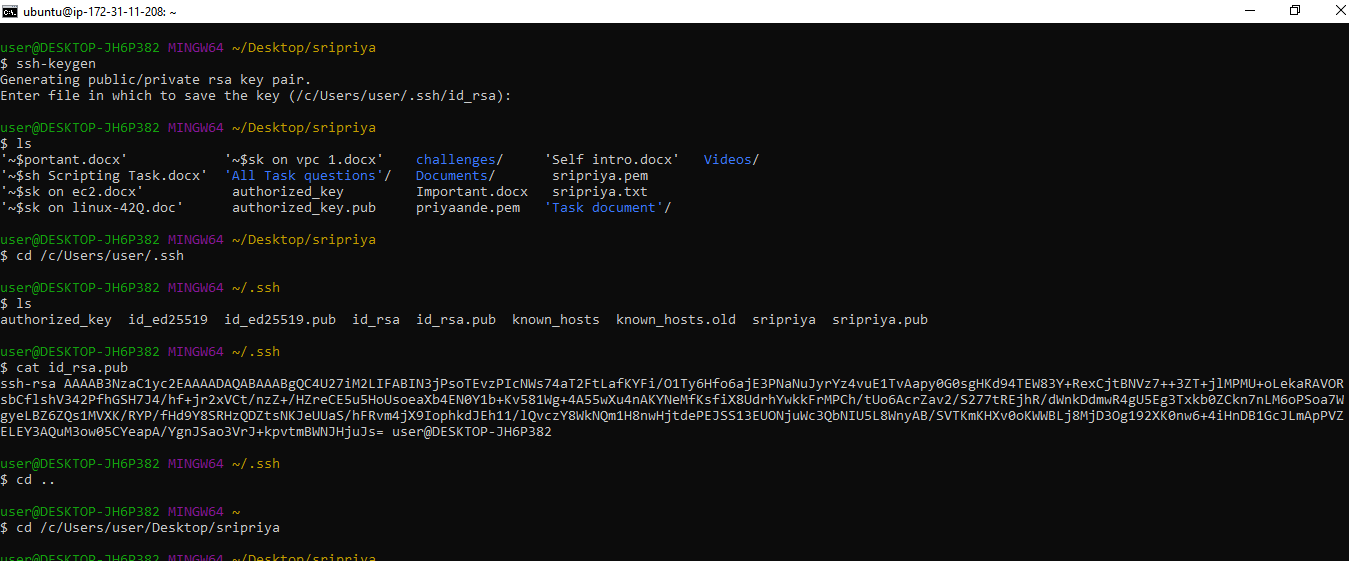


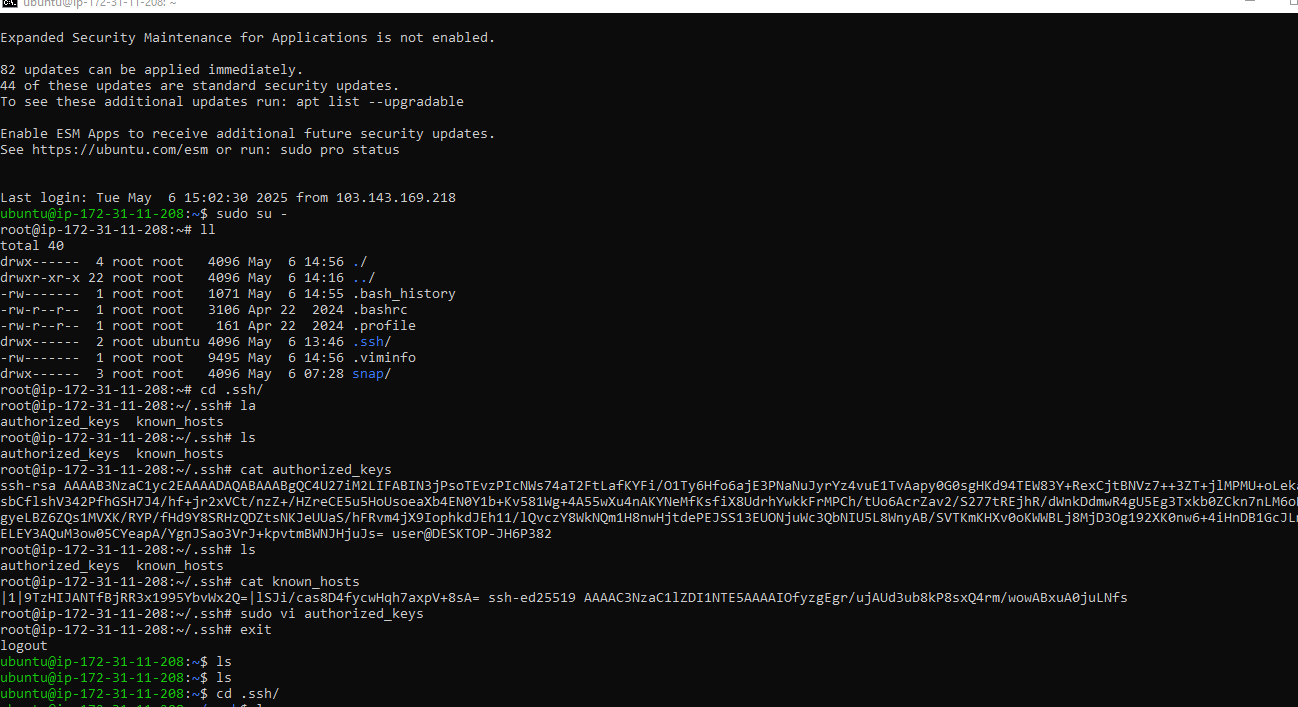


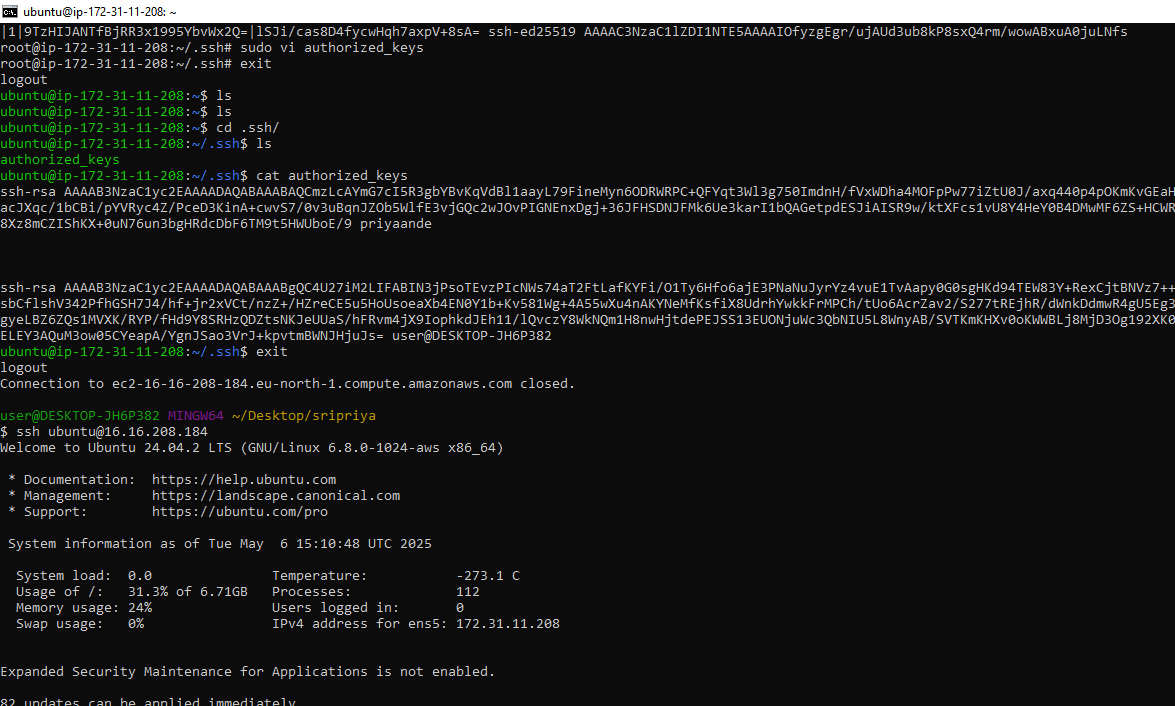


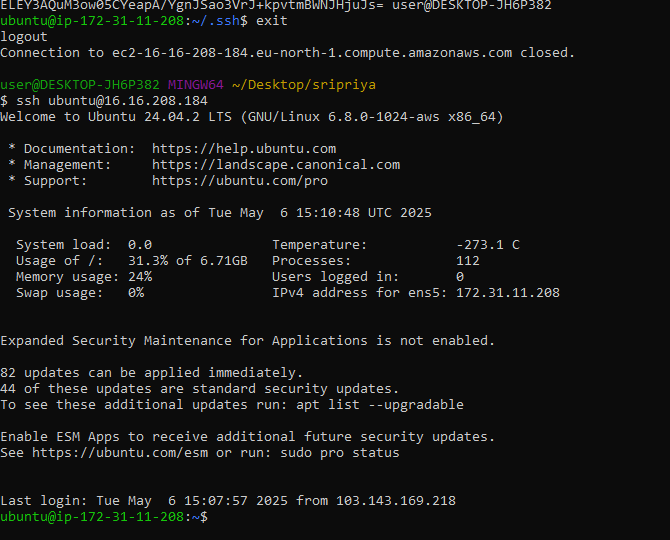
1. **Assign password less authentication for ec2 created on Task 2.**

* Open git bash and enter ssh-keygen
* Then go to /c/Users/user/.ssh/
* $ cat id\_rsa.pub
* Copy publicy key and paste in authorized\_key file
* Then give chomd 600 permissio
* Login to server using pem key
* In server paste this pub key in server and exit and run using ssh ubuntu@ <public ip address>



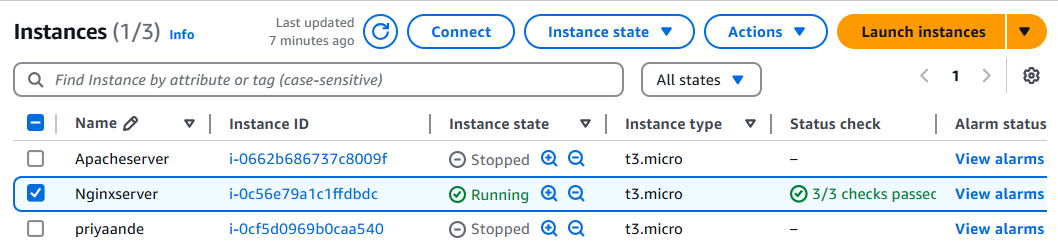


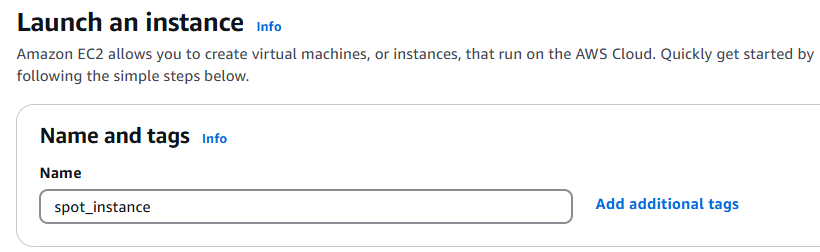


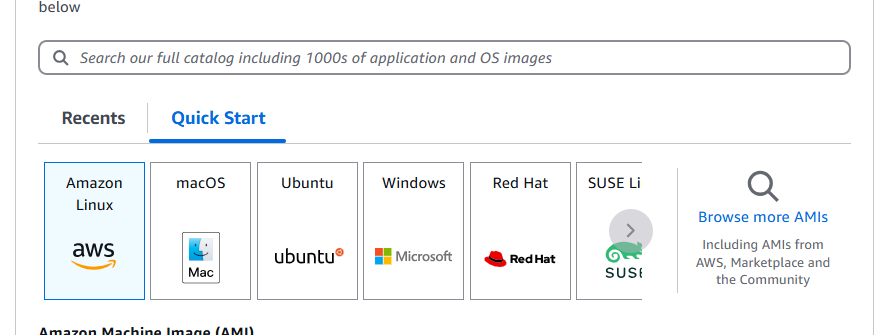


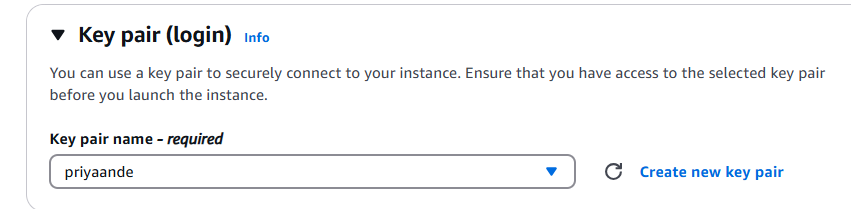
1. **Launch any ec2 using spot purchasing option.**

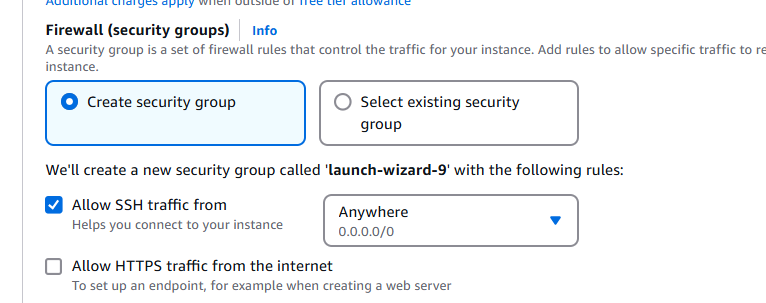
* Launch new instance on ec2
* Name the instance (spot\_instance)
* Select AMI (Aws linux or ubuntu)
* Select instance type
* Select key pair name
* Select security group
* Select advanced options in this option select Purchasing option in that select spot instances
* Click on launch
* Spot\_instance is created

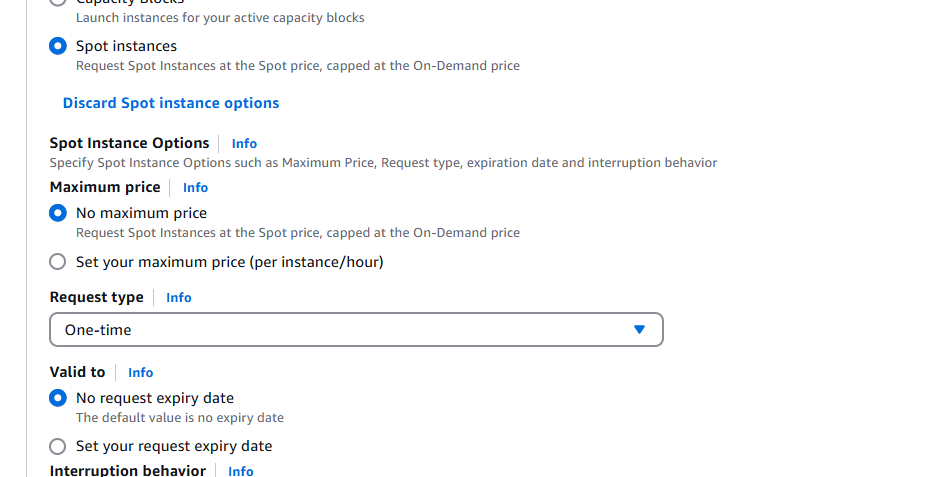


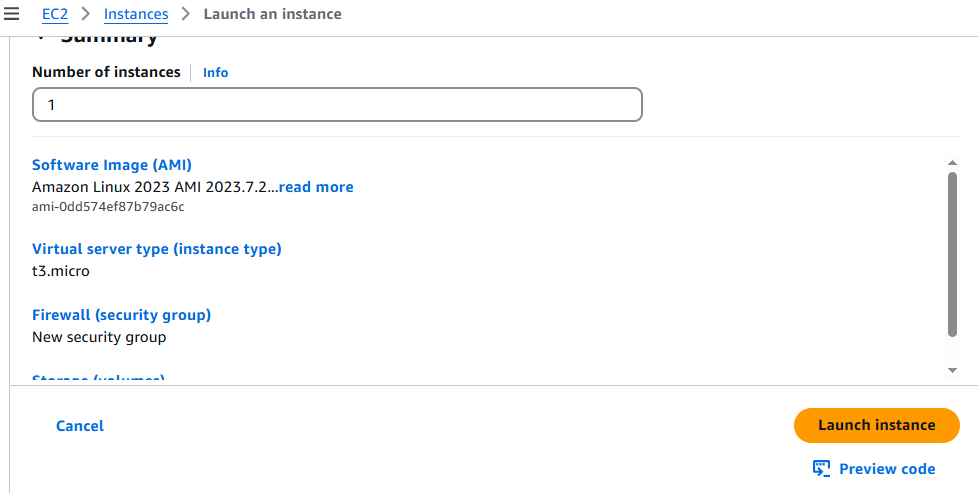


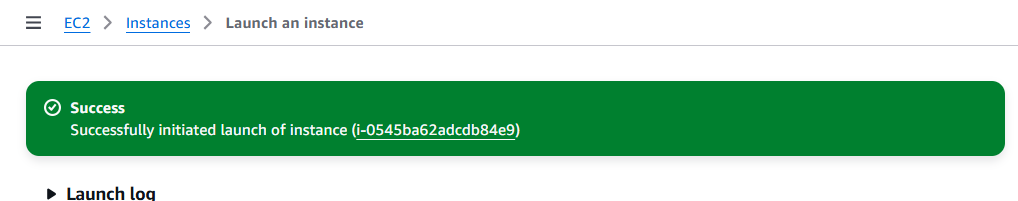










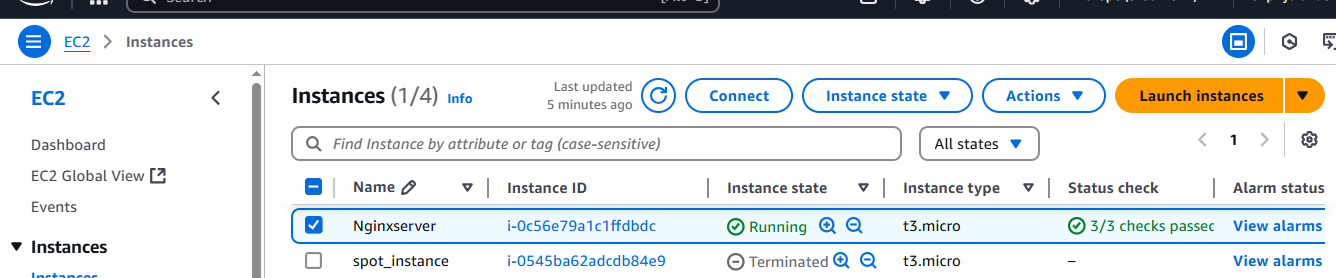


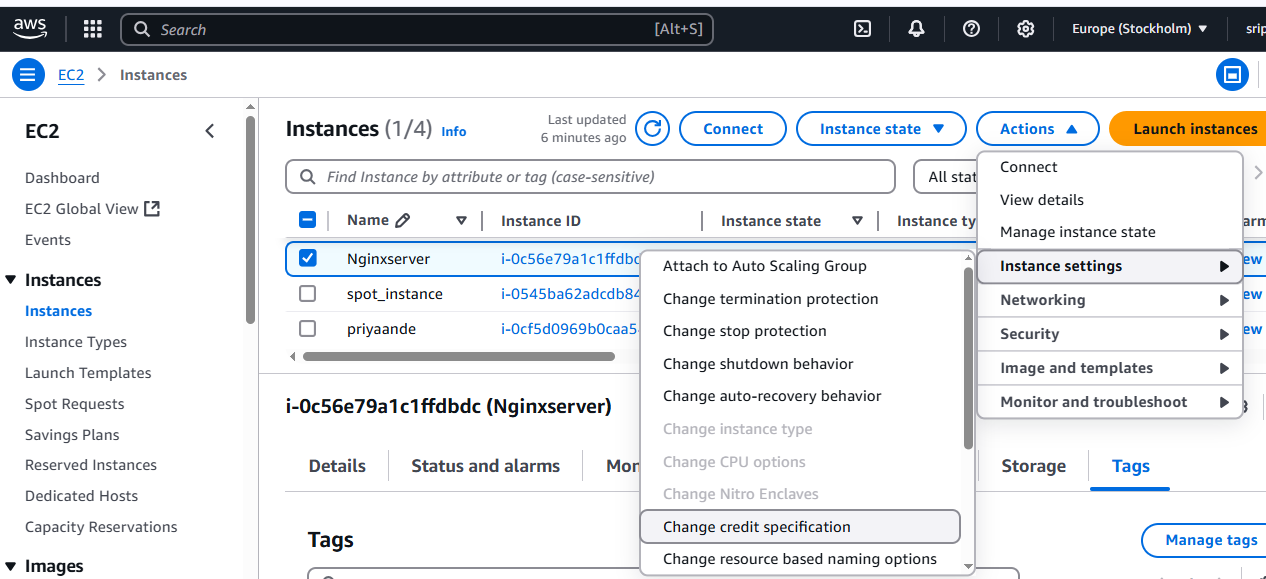


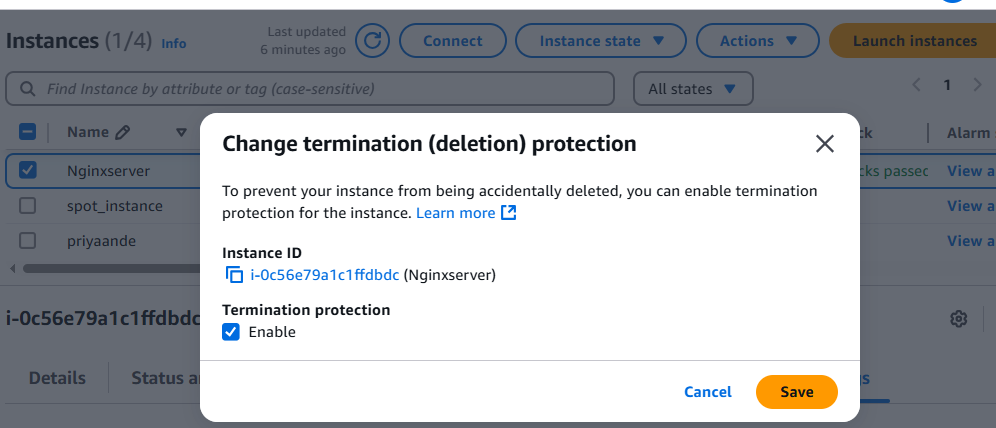
1. **Enable Termination policy on ec2 created in Task 2.**

To prevent your instance from being accidentally deleted, you can enable termination protection for the instance

* Open the **AWS EC2 Console**
* Click **Instances**
* Select the EC2 instance from Ubuntu server
* Click Action In the top menu
* Choose instance settings
* Click on change termination protection and enable
* Click save







**8) Launch one ec2 using Aws CLI.**

* Download AWS CLI from their(windows) website and install
* Go to AwS account and open account top Left side ->click on security credientials
* Scroll down we can see access key click on create access key
* Access key and Security access key is generated
* Open Git bash type aws --version
* Aws configuration give access key and security access key
* aws ec2 run-instances command is used to **launch a new EC2 instance** — think of it as telling AWS, “Create a virtual machine with these specific settings.
* Get ami id,sg id and subnet id from ani of instance
* aws ec2 run-instances \

--image-id ami-0c1ac8a41498c1a9c \

--instance-type t3.micro \

--key-name priyaande \

--security-group-ids sg-0b728d838e8af45d4 \

--subnet-id subnet-07265454bce78a0ca \

--count 1 \

--tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=MyEC2Instance}]'

* If you want to get only the key pair names in a table format

aws ec2 describe-key-pairs --query 'KeyPairs[\*].KeyName' --output table

* aws ec2 describe-instances \

--instance-ids i-0c56e79a1c1ffdbdc \

--query 'Reservations[\*].Instances[\*].PublicIpAddress' \

--output text

* chmod 400 priyaande.pem
* Ssh -i <>pemkey ec2-user<ipaddress>



