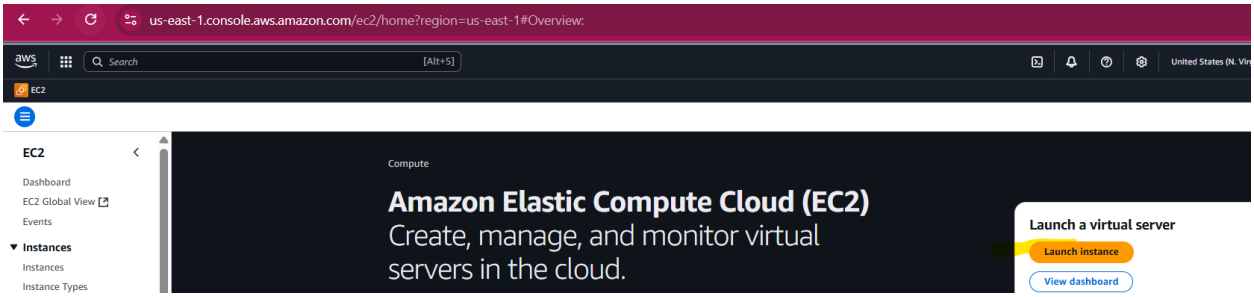
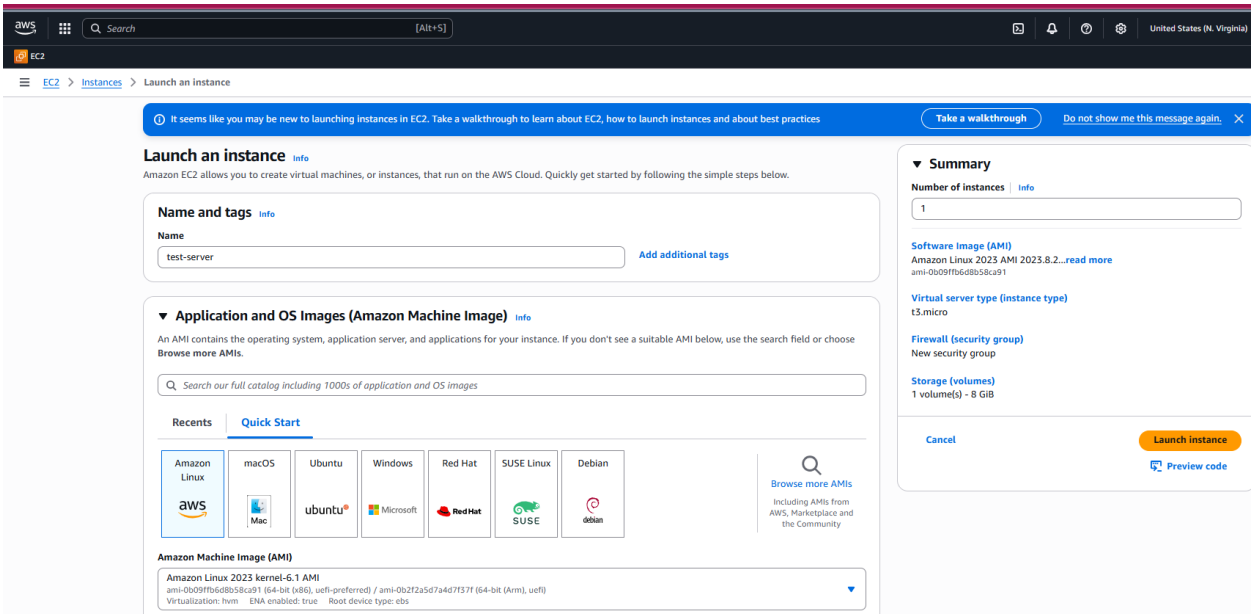
1. Launch one EC2 using Amazon Linux 2 image and add a script in user data to install Apache.

Login to aws account and goto ec2 and click on Launch instance:

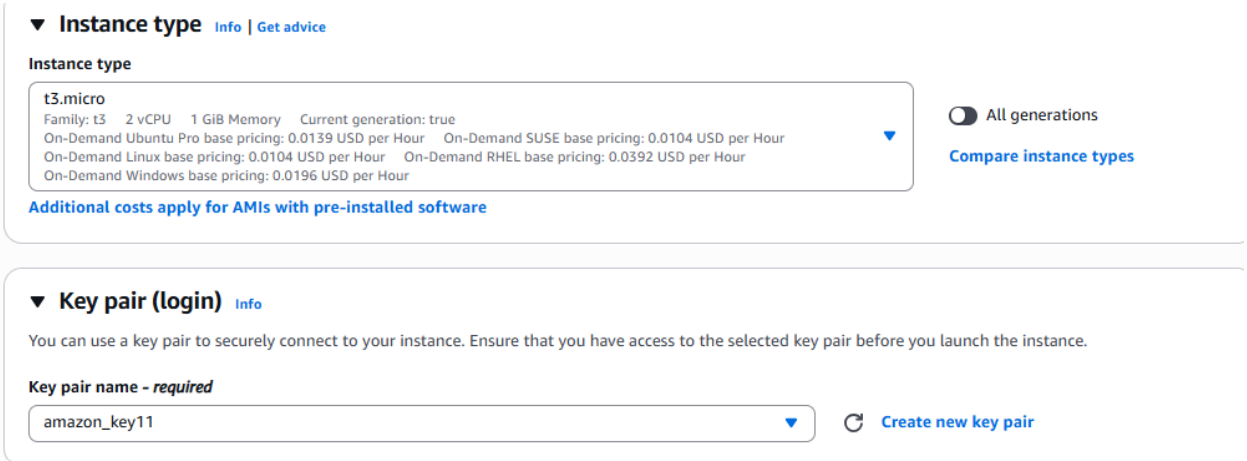


And provide the Name tag and select AMI 'Amazon Linux 2023':



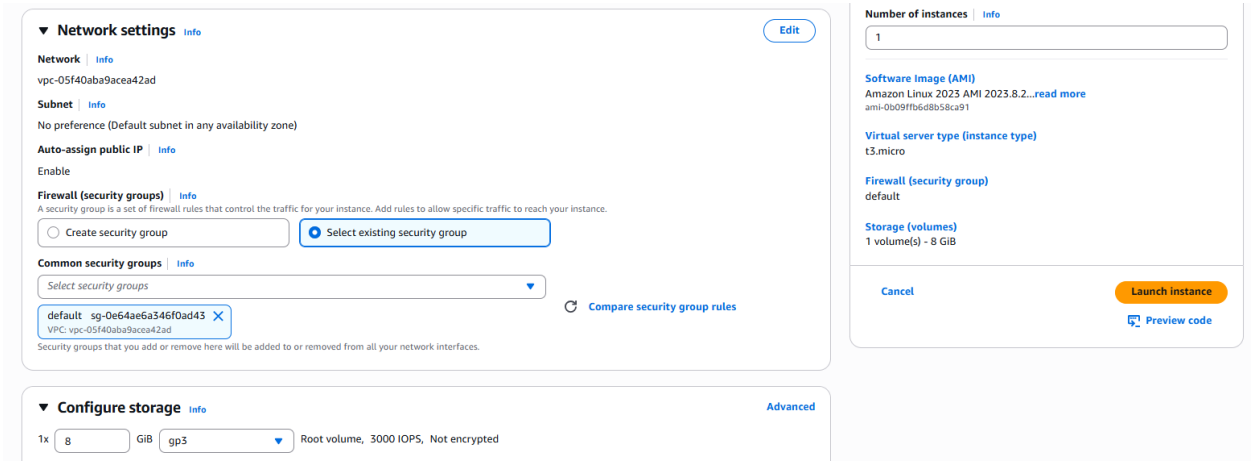
Select the instance type as t3.micro and create a 'key pair'.

Note: I have previously created key pair so I just selected existing key pair:

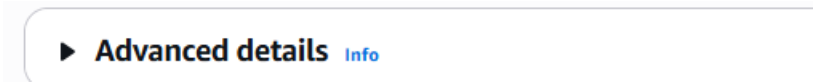


Create the Security group,

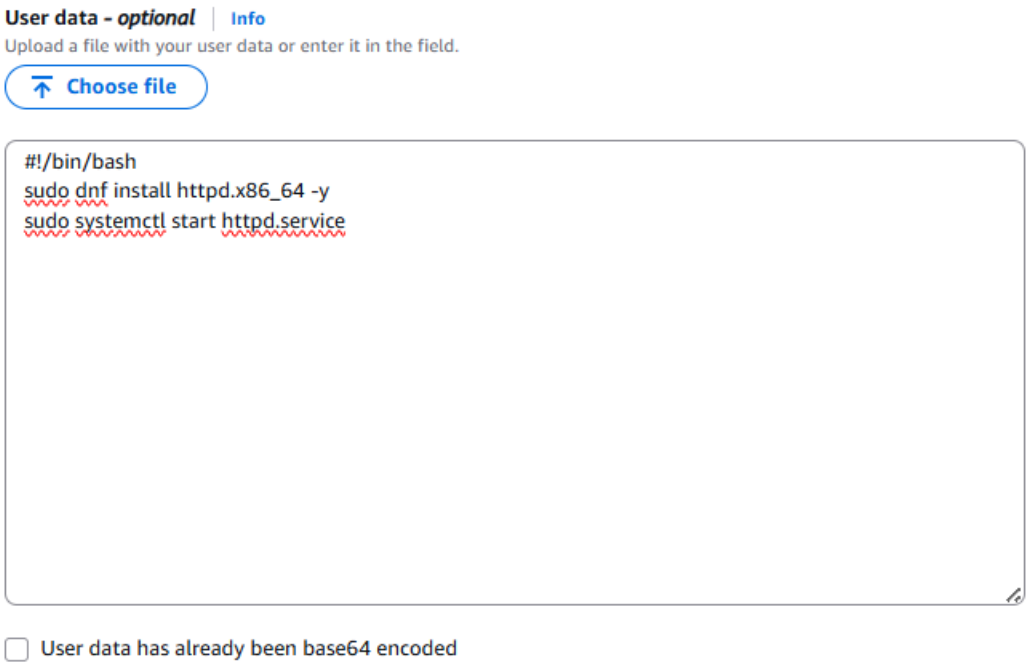
Note: I have used the default Security group:



And click on Launch Template:



and write the script in the user_data:



And Launch the instance by clicking on ‘Launch instance’ in right hand side.

Then login to the server using ssh and pem key and then check the apache is running or not status:

```
[ec2-user@ip-172-31-21-198 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: d
     Active: active (running) since Wed 2025-09-17 07:30:08 UTC; 5min ago
       Docs: man:httpd.service(8)
   Main PID: 3202 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes
      Tasks: 177 (limit: 1057)
     Memory: 13.3M
        CPU: 416ms
     CGroup: /system.slice/httpd.service
             ├─3202 /usr/sbin/httpd -DFOREGROUND
             ├─3213 /usr/sbin/httpd -DFOREGROUND
             ├─3215 /usr/sbin/httpd -DFOREGROUND
             ├─3216 /usr/sbin/httpd -DFOREGROUND
             └─3313 /usr/sbin/httpd -DFOREGROUND
```

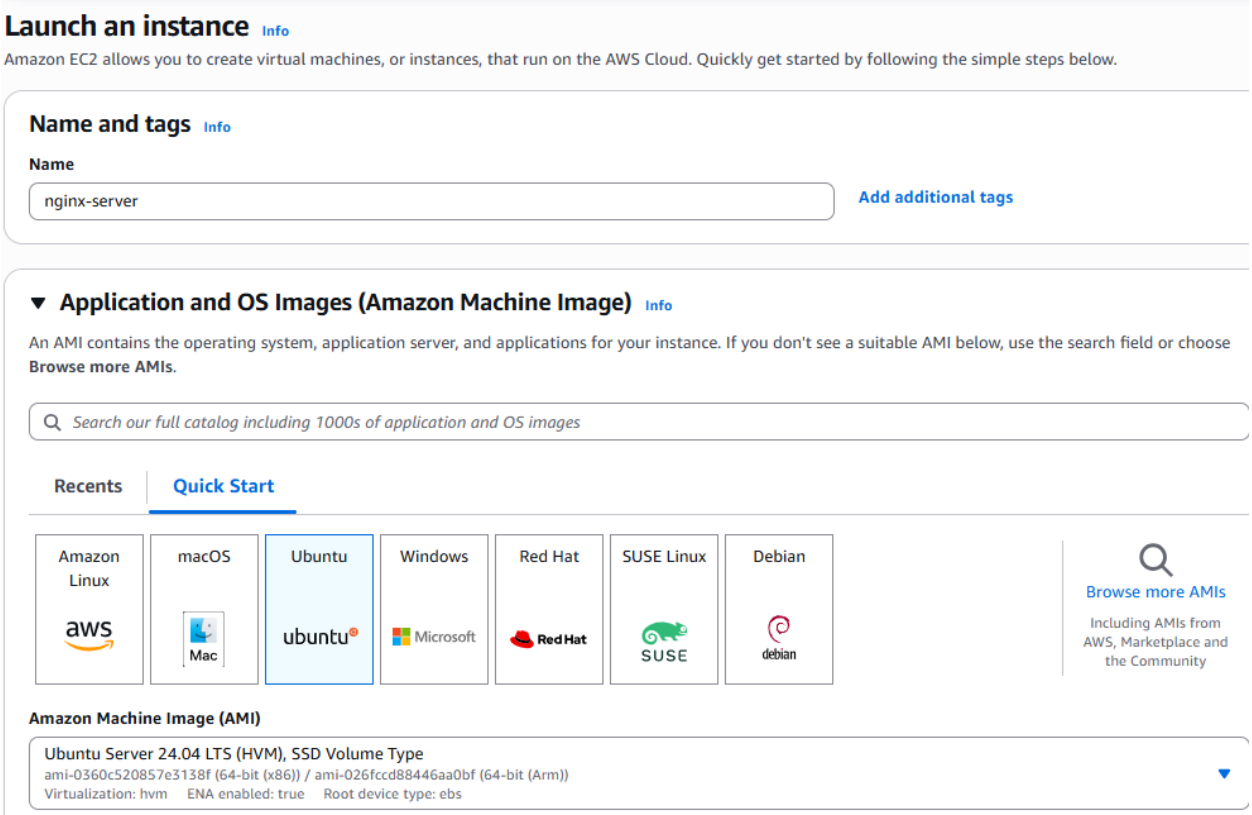
Check the script which has been written in the user_data while launching the instance:

```
[ec2-user@ip-172-31-21-198 ~]$ sudo cat /var/lib/cloud/instance/user-data.txt
#!/bin/bash
sudo dnf install httpd.x86_64 -y
sudo systemctl start httpd.service
```

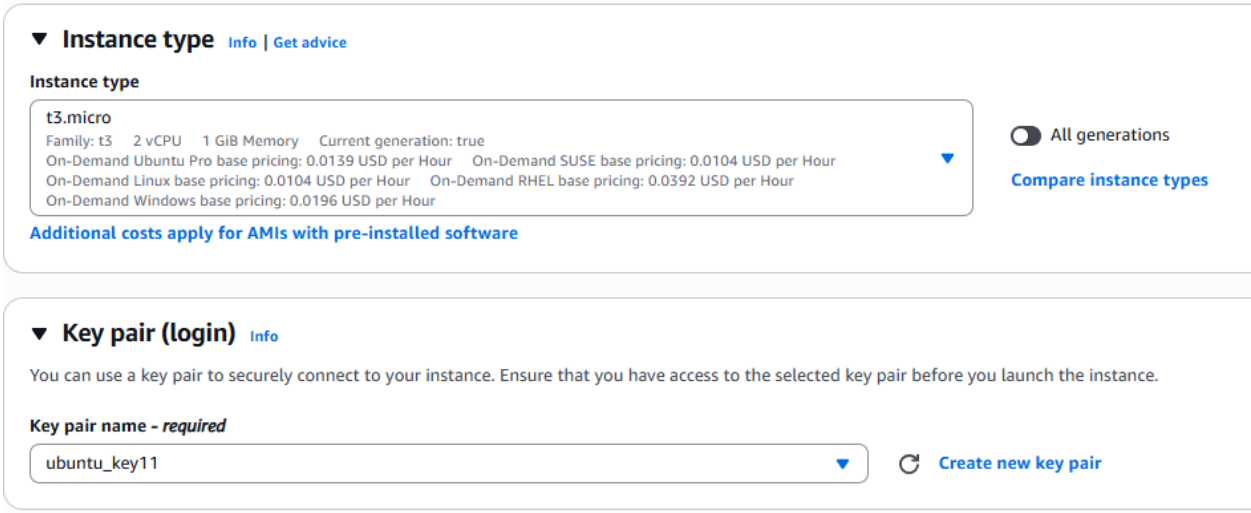
---

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

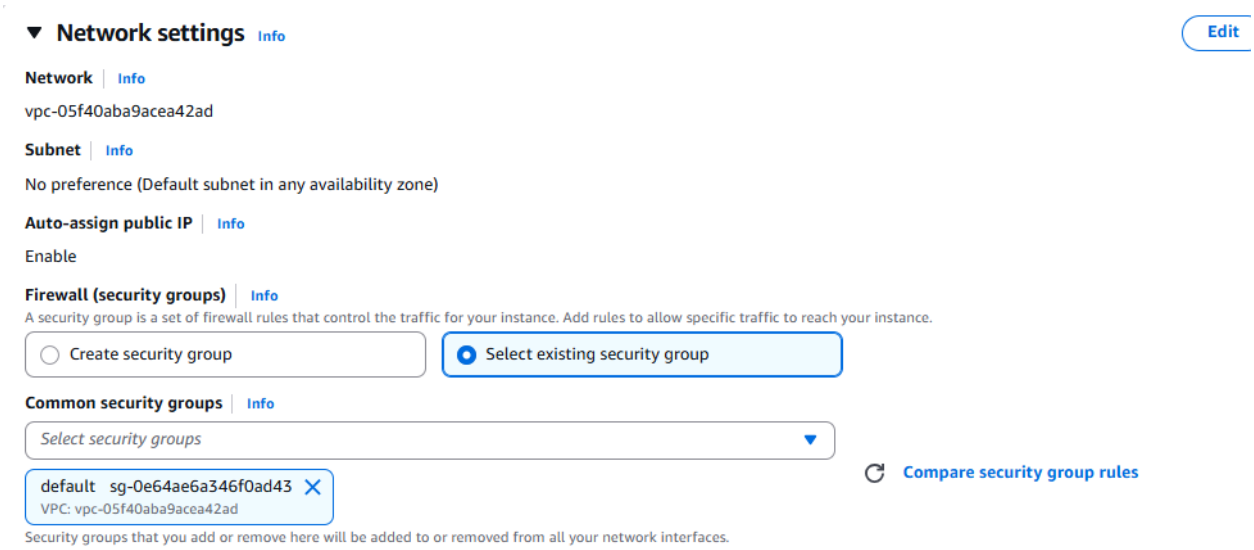
   Launch the ec2 instance and give the name and select AMI as ubuntu:



Select the instance type as ‘t3.micro’ and create a key pair by giving the name:

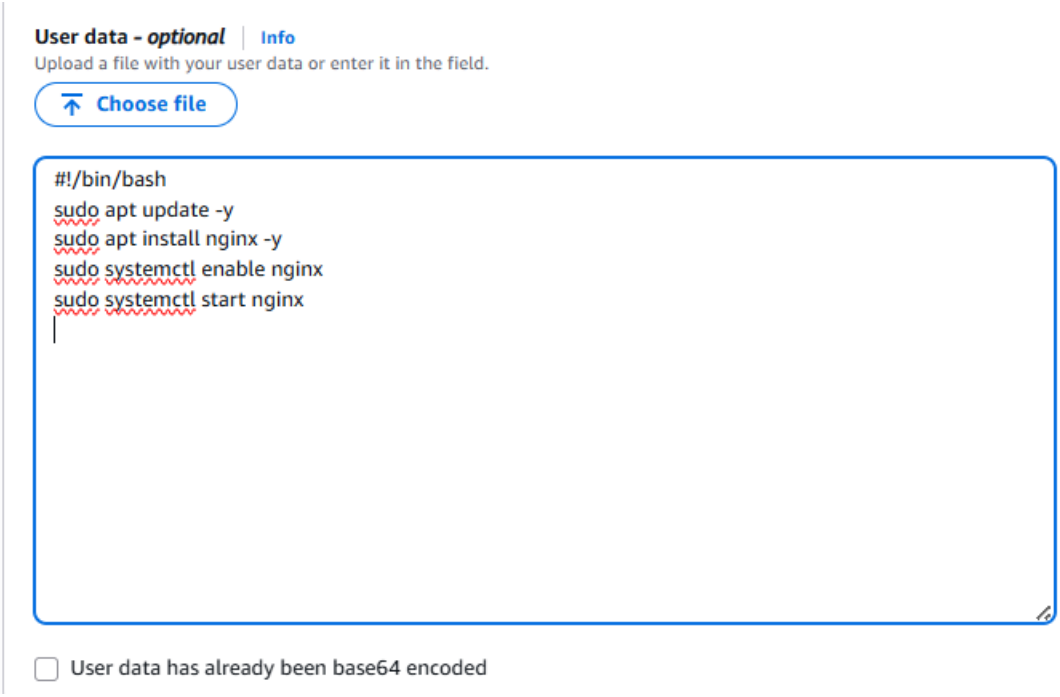


Select the Default security group:

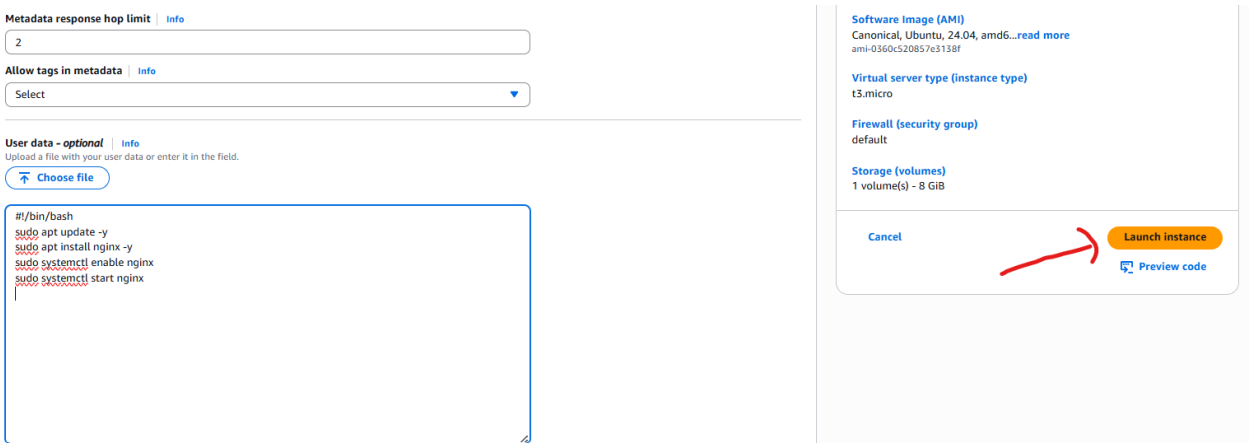


Expand the Advanced details:

Write the script in user_data:



And launch the instance:



Login to ubuntu server using pem key and check nginx is running or not:

```
ubuntu@ip-172-31-17-175:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: en>
     Active: active (running) since Wed 2025-09-17 07:48:47 UTC; 5min ago
       Docs: man:nginx(8)
   Main PID: 1600 (nginx)
      Tasks: 3 (limit: 1008)
     Memory: 2.5M (peak: 5.3M)
```

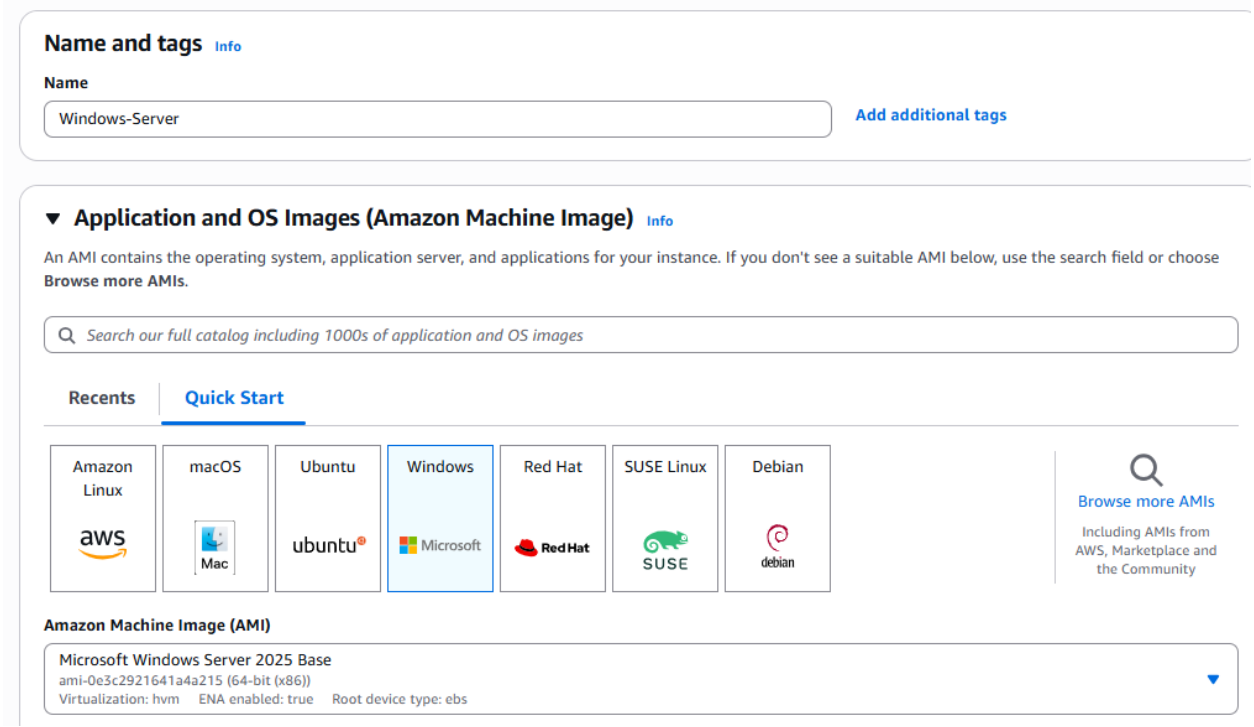
To check the script which is written in the user_data while launching the instance:

```
ubuntu@ip-172-31-17-175:~$ sudo cat /var/lib/cloud/instance/user-data.txt
#!/bin/bash
sudo apt update -y
sudo apt install nginx -y
sudo systemctl enable nginx
sudo systemctl start nginx
```

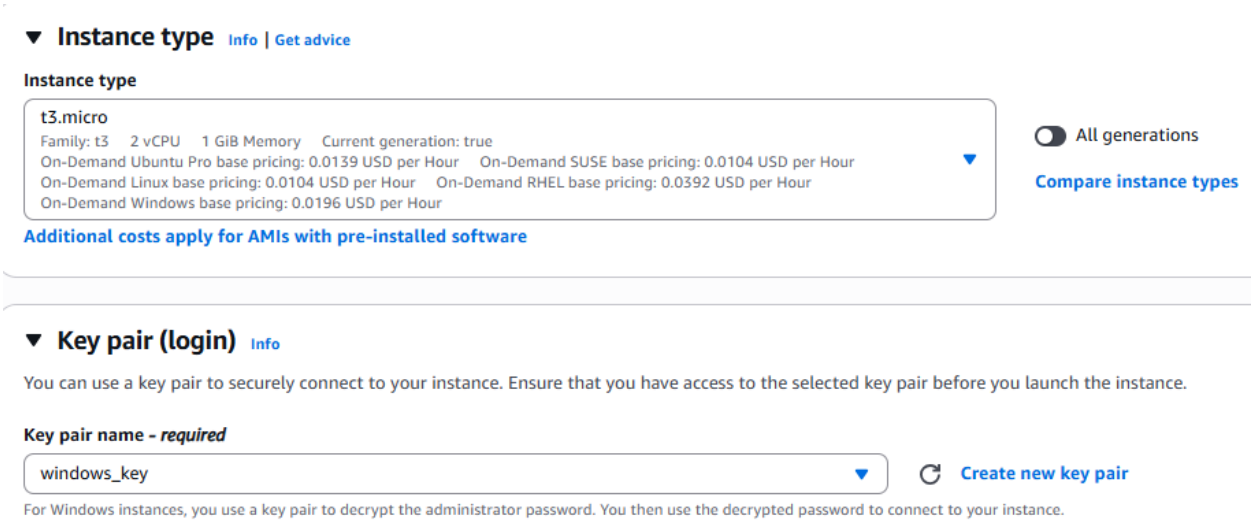
---

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

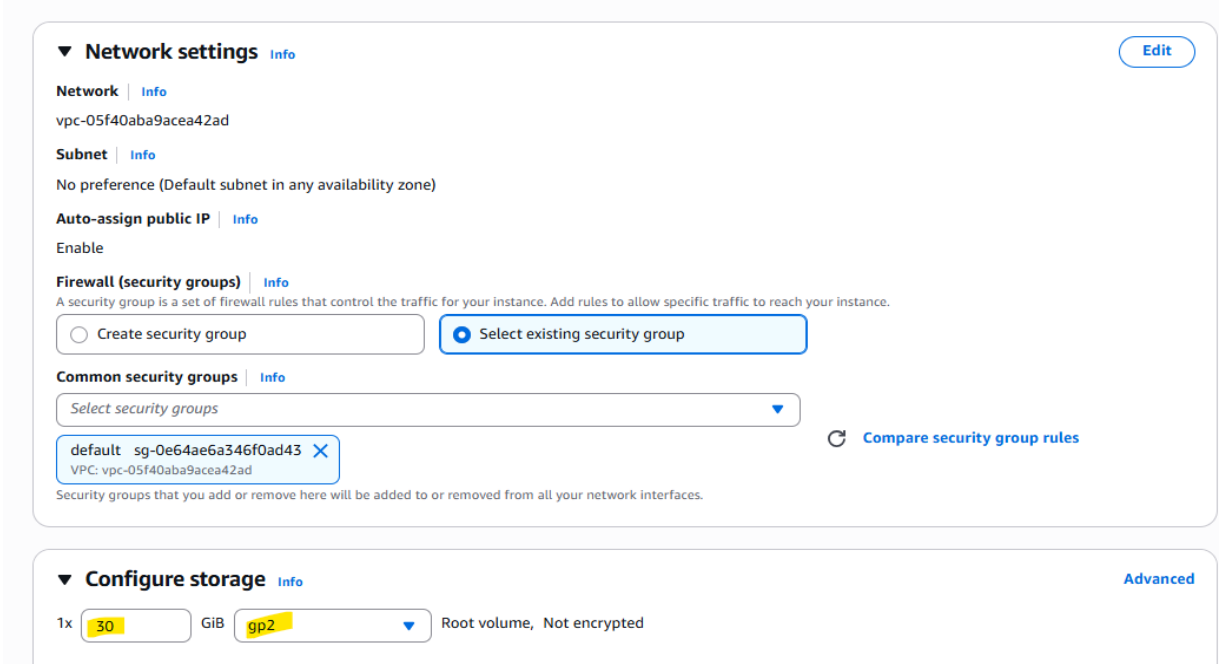
   Launch the ec2 instance with Windows 2025 base Server:



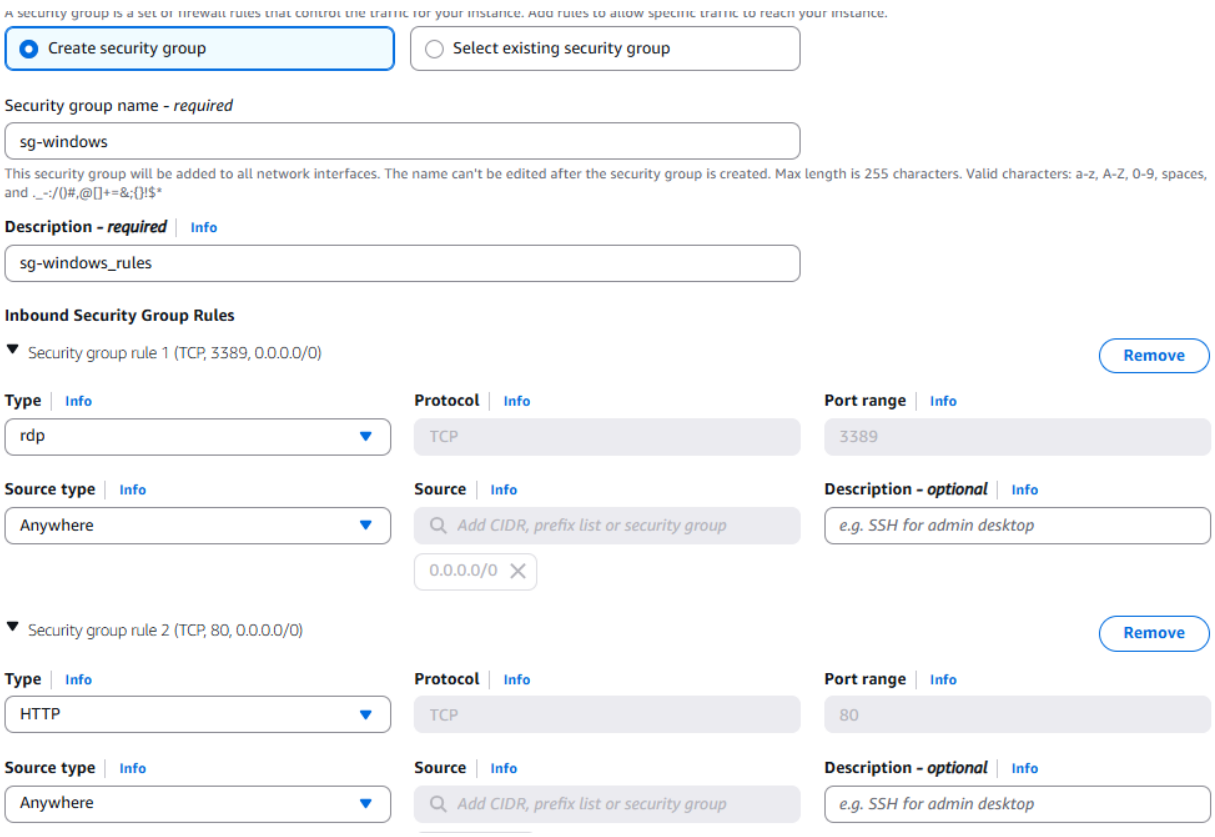
Provide the instance type and create a key pair:

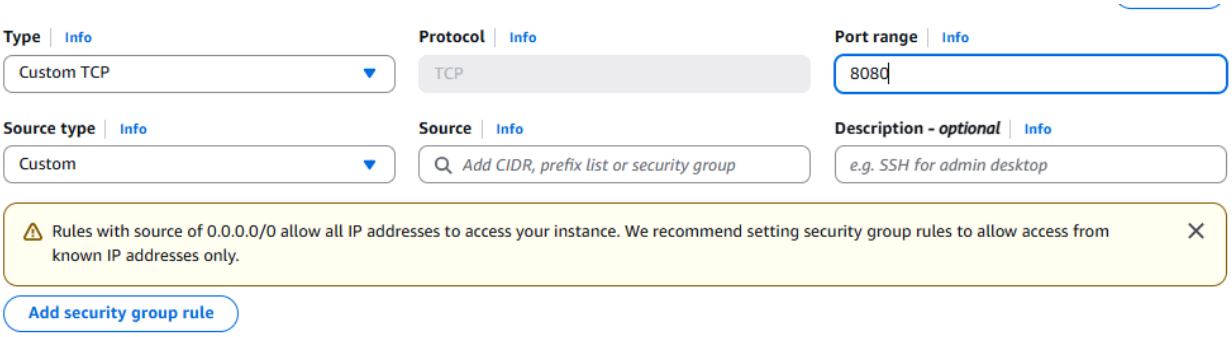


And storage give 30gb, gp2

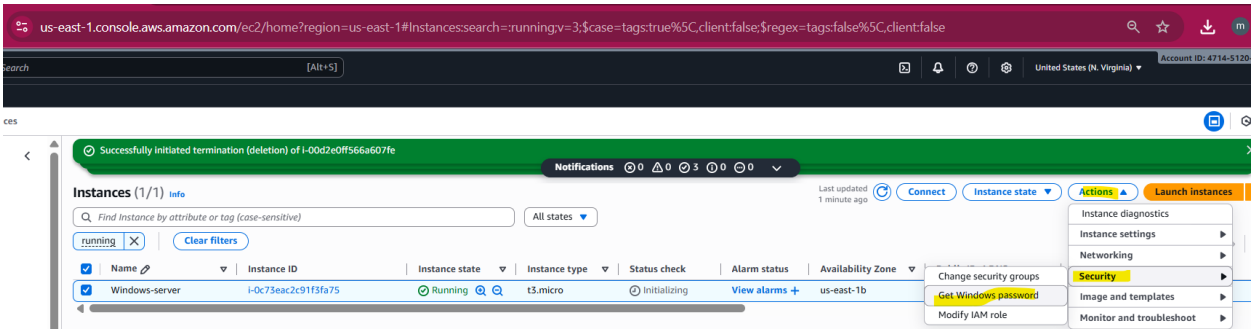


Create security group and open ports: 3389 for rdp, 80 for http, 8080 for tomcat



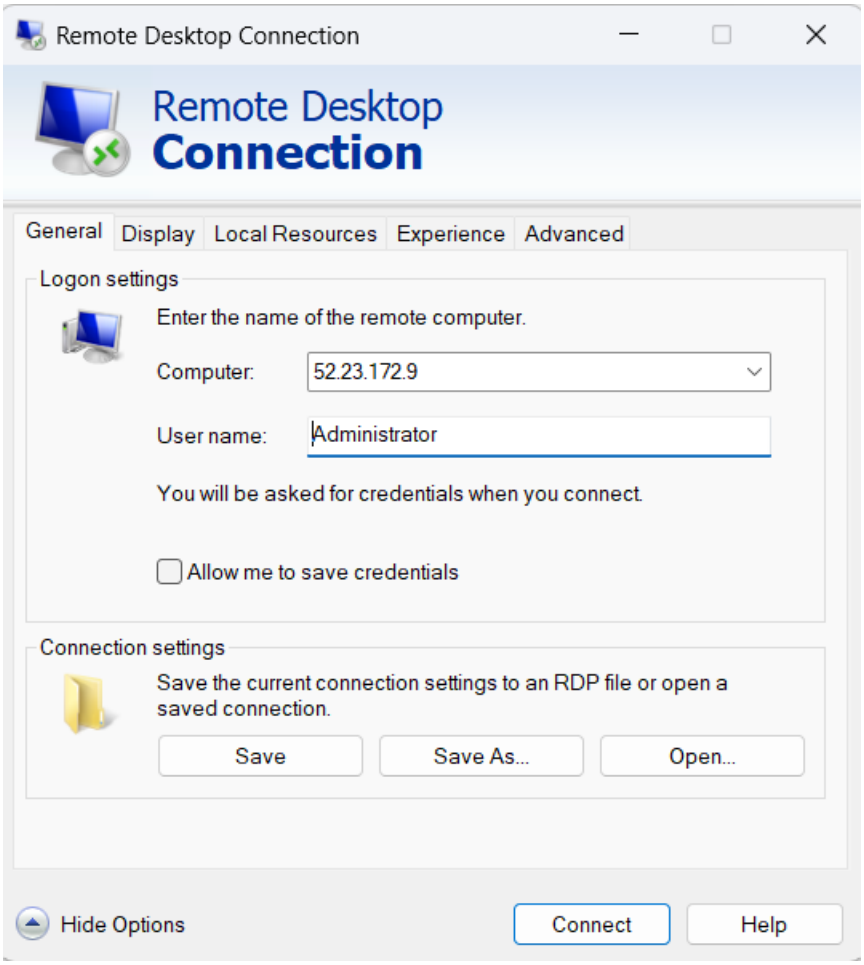


And get Windows password by clicking on ->Actions->Security->Get Windows Password

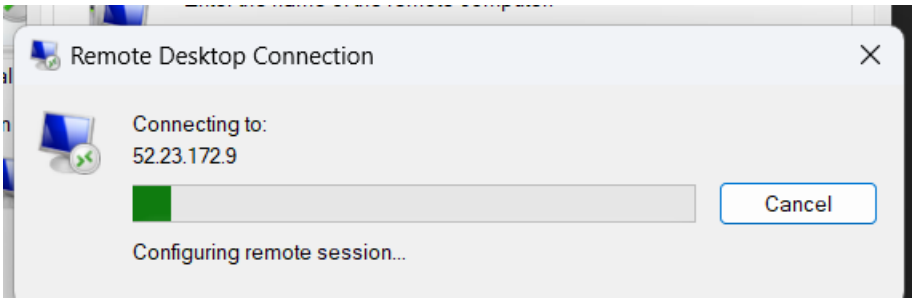


And then decrypt the password by providing the generated .pem key.

And to connect to windows : mstsc->ip, username: Administrator, password: copied password:



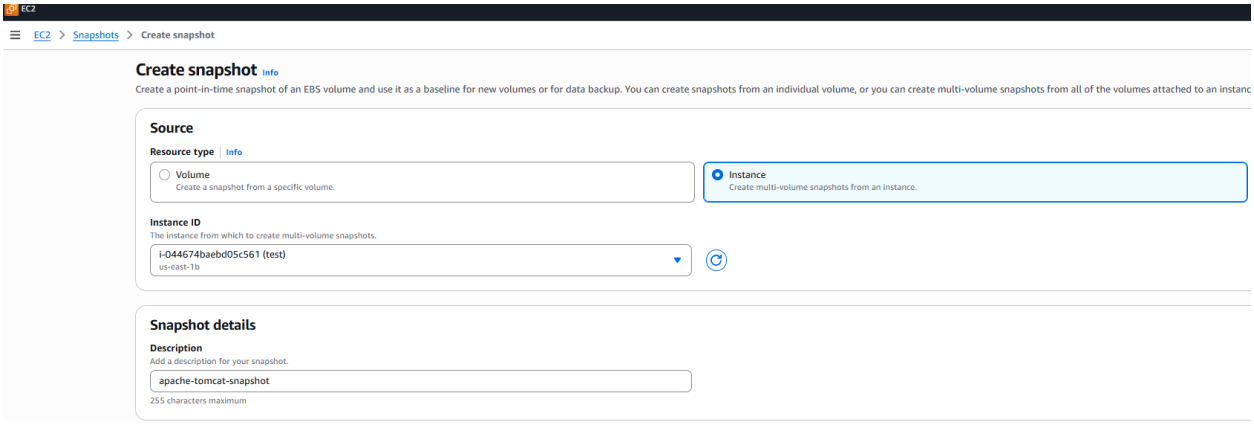
And click on connect



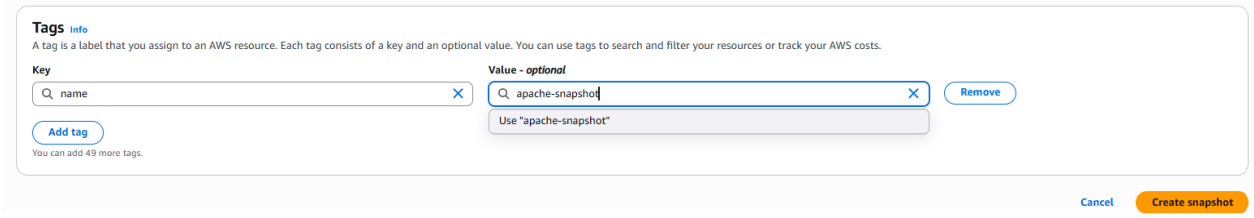
4. Take a snapshot of the instance created in Task 1.

In the Task1 Launched the ec2 instance with apache installed on it using user_data .

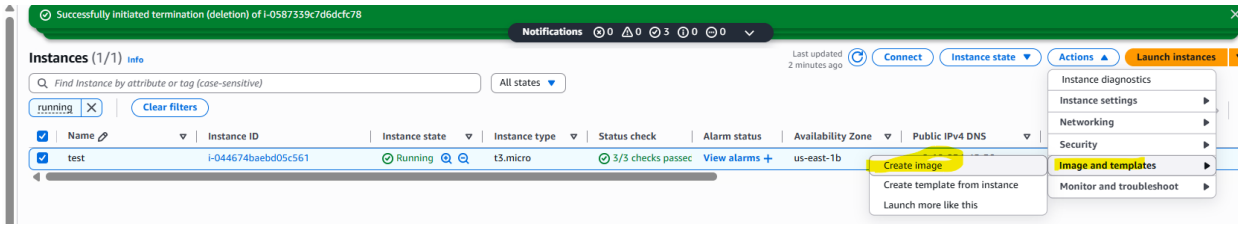
Click on snapshot and select the instance from which want to create snapshot:



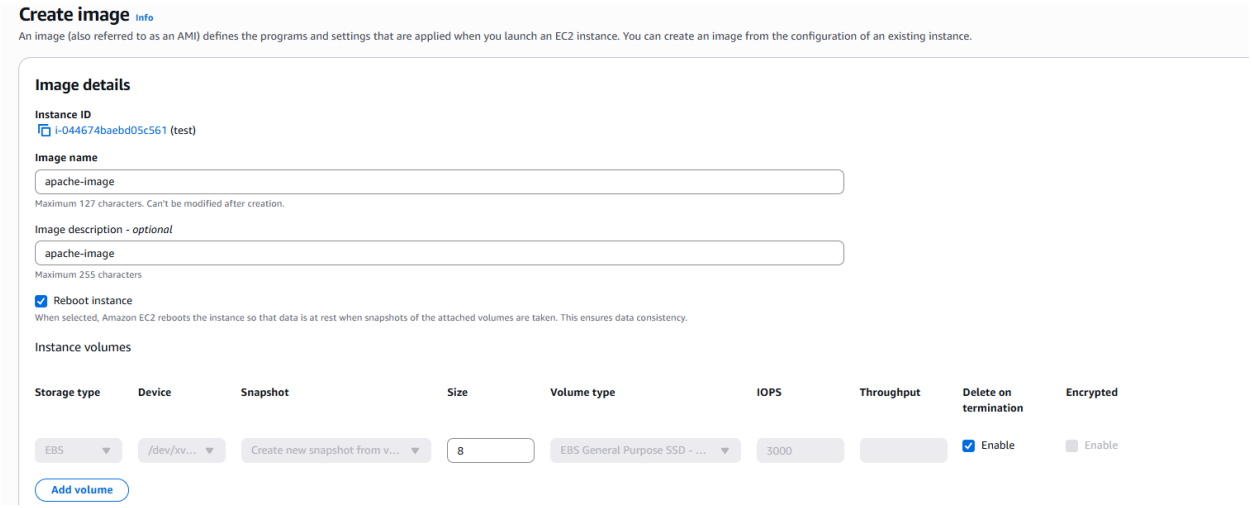
Click on create snapshot:



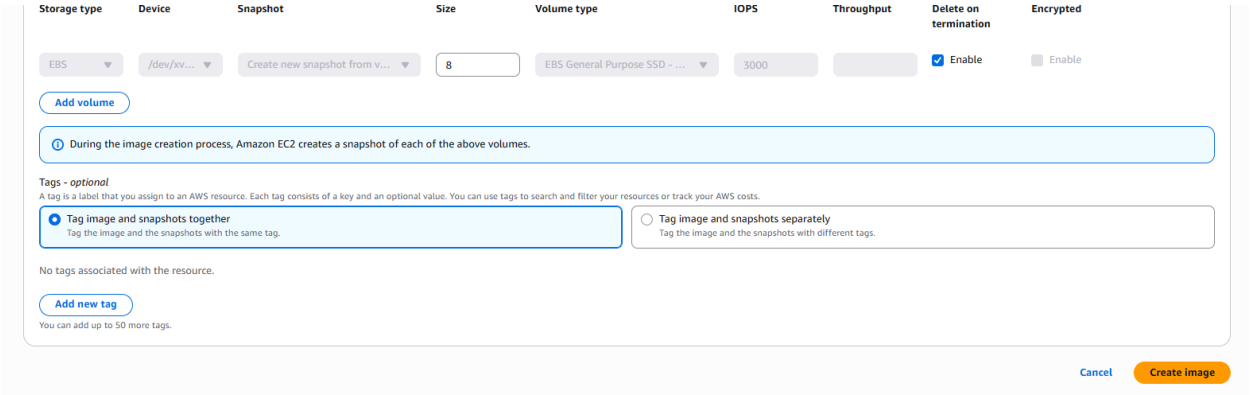
Create an Image from instance:



Provide the image name and description:



Cick on create image:



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

   In task2 we have launched an ec2 instance with ubuntu and install nginx in user_data.

Generate ssh-keygen:

```
dell@DESKTOP-6ORBKUF MINGW64 ~/Downloads
$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/dell/.ssh/id_ed25519):
/c/Users/dell/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase for "/c/Users/dell/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/dell/.ssh/id_ed25519
Your public key has been saved in /c/Users/dell/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:W5l/G3r1CJ+y32tlQ7Yt6njP3pOUC+XuS1cSbDoc21E dell@DESKTOP-6ORBKUF
The key's randomart image is:
```

And copy the public key:

```
dell@DESKTOP-6ORBKUF MINGW64 ~/Downloads
$ cat /c/Users/dell/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAII3H17TSMIosON5kZL6sFvXFxcLKjluG2Vp5JjTC7sii
 dell@DESKTOP-6ORBKUF
```

Into the ~/.ssh/authorized_keys:

```
[ec2-user@ip-172-31-46-215 ~]$ cat ~/.ssh/authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAII3H17TSMIosON5kZL6sFvXFxcLKjluG2Vp5JjTC7sii
dell@DESKTOP-6ORBKUF
```

Change the ownership means it should be ec2-user not the root:

```
[root@ip-172-31-46-215 ~]# chown ec2-user:ec2-user ~/.ssh
[root@ip-172-31-46-215 ~]# chown ec2-user:ec2-user ~/.ssh/authorized_keys
```

```
[ec2-user@ip-172-31-46-215 ~]$ ls -ld ~/.ssh
drwx------. 2 ec2-user ec2-user 61 Sep 17 16:58 /home/ec2-user/.ssh
[ec2-user@ip-172-31-46-215 ~]$ ls -l ~/.ssh/authorized_keys
-rw-------. 1 ec2-user ec2-user 102 Sep 17 16:58 /home/ec2-user/.ssh/authorized_
keys
```

And also change the permission:

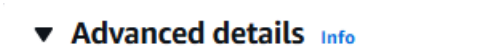
```
drwx------. 2 root root   29 Sep 17 16:37 .ssh
-rw-r--r--. 1 root root  129 Feb  2  2023 .tcshrc
-rw-------. 1 root root 2172 Sep 17 16:37 .viminfo
[root@ip-172-31-46-215 ~]# chmod 700 ~/.ssh
[root@ip-172-31-46-215 ~]# ls -lA
total 28
-rw-------. 1 root root   37 Sep 17 16:37 .bash_history
-rw-r--r--. 1 root root   18 Feb  2  2023 .bash_logout
-rw-r--r--. 1 root root  141 Feb  2  2023 .bash_profile
-rw-r--r--. 1 root root  429 Feb  2  2023 .bashrc
-rw-r--r--. 1 root root  100 Feb  2  2023 .cshrc
drwx------. 2 root root   29 Sep 17 16:37 .ssh
-rw-r--r--. 1 root root  129 Feb  2  2023 .tcshrc
```

```
[root@ip-172-31-46-215 ~]# chmod 600 ~/.ssh/authorized_keys
```

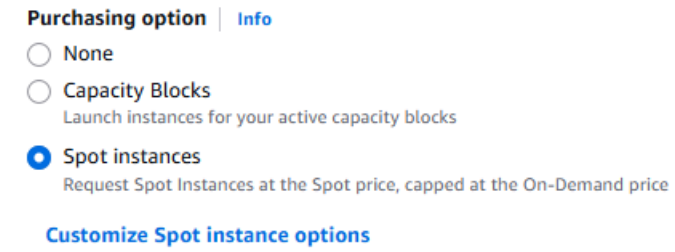
```
dell@DESKTOP-6ORBKUF MINGW64 ~/Downloads
$ ssh ec2-user@13.60.105.19
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Wed Sep 17 16:53:44 2025 from 103.143.169.218
```

6. Launch any EC2 using the spot purchasing option.

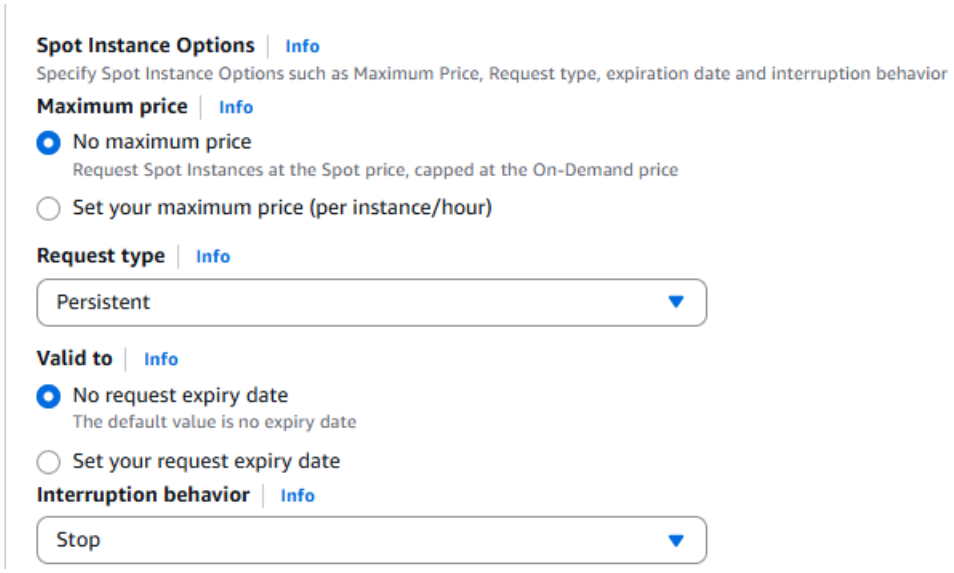
   During ec2 instance Launch click on Advanced details:



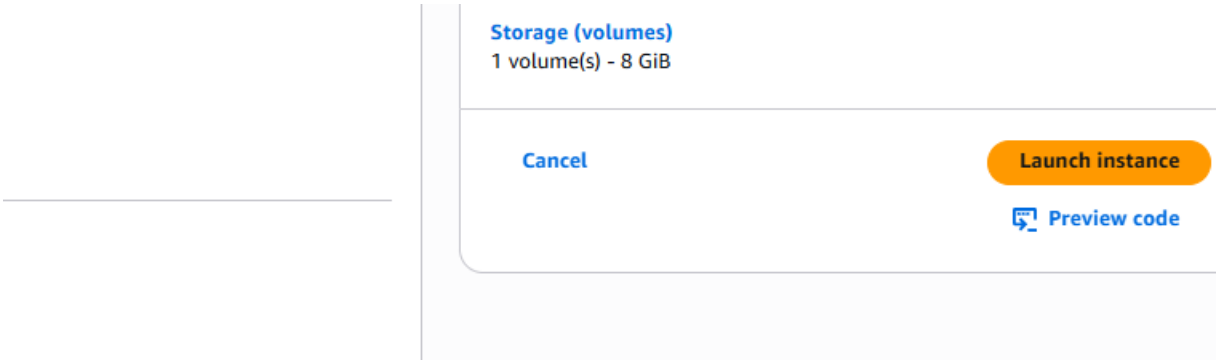
   In purchasing options select ‘Spot instances’:



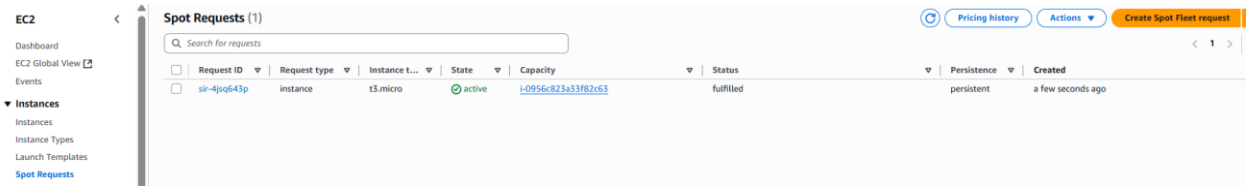
   Click on customize spot instance options, and select Request type as Persistent, and Interruption behavior as Stop:



Click on Launch instance:



the created spot instance we can see in spot Request:

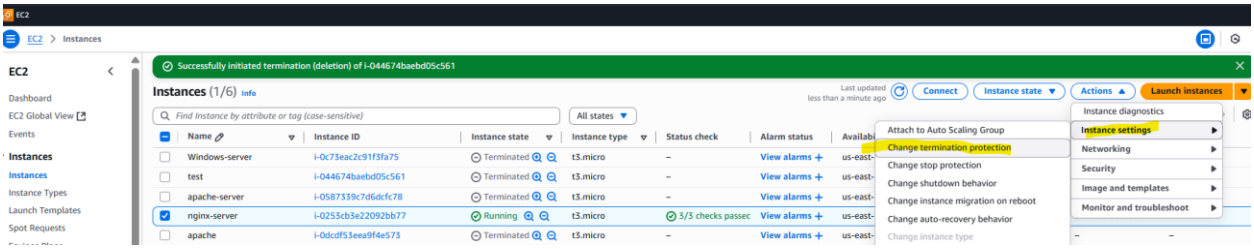


---

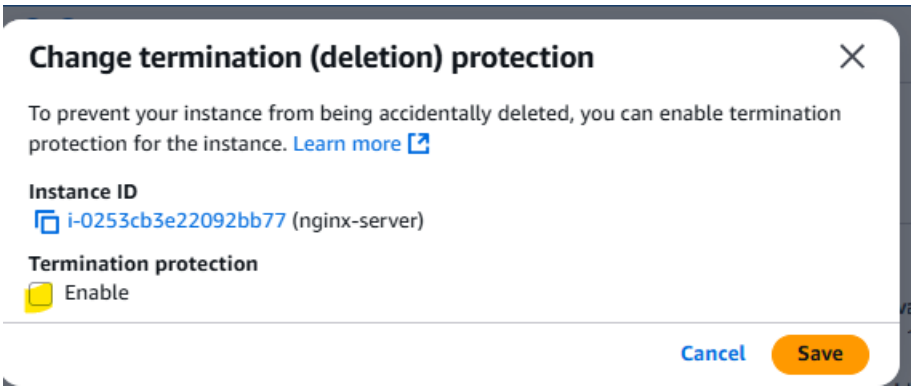
7. Enable termination policy on the EC2 created in Task 2.

   In task2 we have launched an ec2 instance with ubuntu and install nginx in user_data.

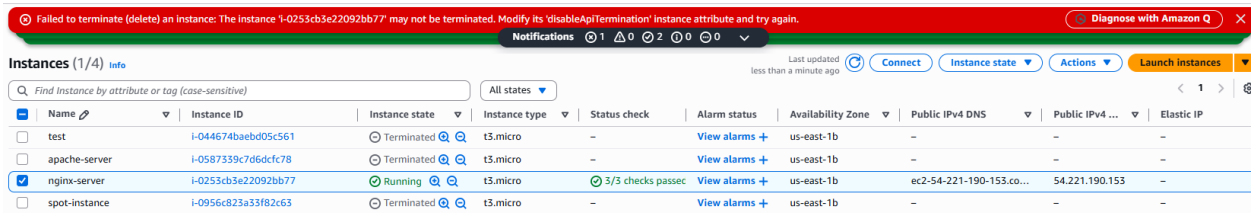
   Select the instance and click on Actions->instance Settings->change termination protection:



Enable the Termination protection:



Then try to delete the instance, it won't allow us to delete because of this Termination protection policy:



---

8. Launch one EC2 using AWS CLI.

Download aws cli:

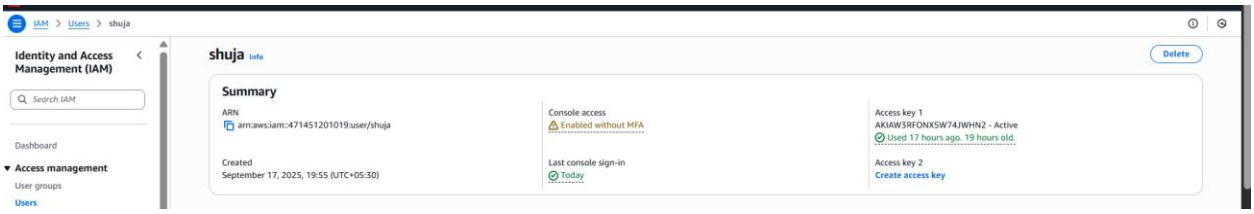
For various parameters that can be used with `msiexec`, see msiexec on the *Microsoft Docs* website. For example, you can use the `/qn` flag for a silent installation.

```
C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi /qn
```

To confirm the installation, open the **Start** menu, search for `cmd` to open a command prompt window, and at the command prompt use the `aws --version` command.

```
C:\> aws --version
aws-cli/2.27.41 Python/3.11.6 Windows/10 exe/AMD64 prompt/off
```

Get the access key and secret key by generating from IAM user:



Provide **Access Key, Secret Key, Region, Output format**.

```
dell@DESKTOP-6ORBKUF MINGW64 ~
$ aws configure
AWS Access Key ID [****************WHN2]:
AWS Secret Access Key [****************lxte]:
Default region name [us-east-1]:
Default output format [None]:
```

Create the pem key using aws cli:

```
dell@DESKTOP-6ORBKUF MINGW64 ~
$ aws ec2 create-key-pair --key-name my-key --query 'KeyMaterial' --output text
> my-key.pem
chmod 400 my-key.pem
```

Create an ec2 instance using aws cli:

```
dell@DESKTOP-6ORBKUF MINGW64 ~
$ aws ec2 run-instances   --image-id ami-08982f1c5bf93d976   --count 1   --insta
nce-type t3.micro   --key-name my-key   --security-group-ids sg-0e64ae6a346f0ad4
3   --subnet-id subnet-0242f46eea951417e   --tag-specifications 'ResourceType=in
stance,Tags=[{Key=Name,Value=MyFirstEC2}]'
{
```

We can see the created instance from the aws cli:

----------Completed----------