**Aws Assignment**

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

* Launch a ec2 instance with a pem key and your public ip.
* Then check the security inbound rules ssh should be 22 and http in 80
* Then go to the bottom instance and additional info open it
* And use a bash script for it
* Then go the git bash connect to the server and take the public ip and paste it with last gave that pot number of :80
* Here the results are

**Bash script download apache process:**

#!/bin/bash

# Update system

yum update -y

# Install Apache (httpd)

yum install -y httpd

# Enable Apache to start on boot

systemctl enable httpd

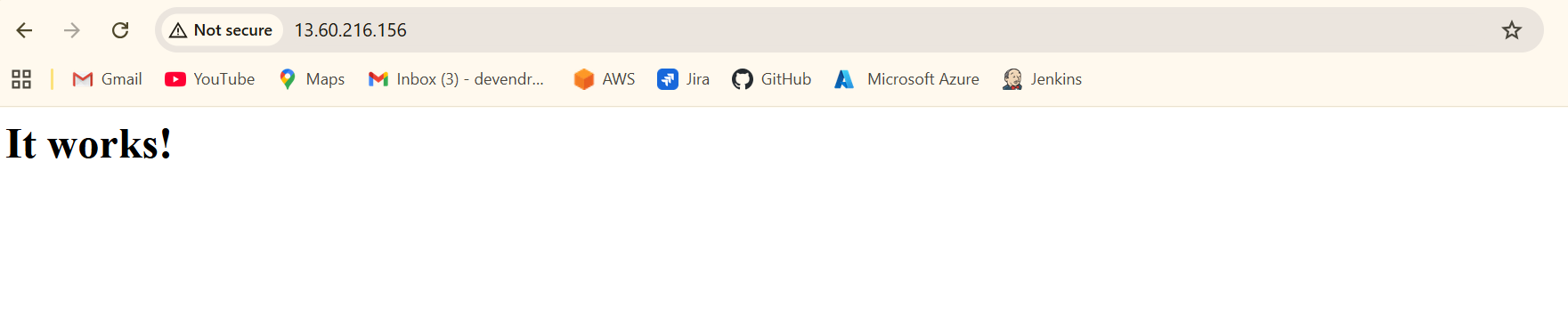
# Start Apache service

systemctl start httpd

# Create a test index.html

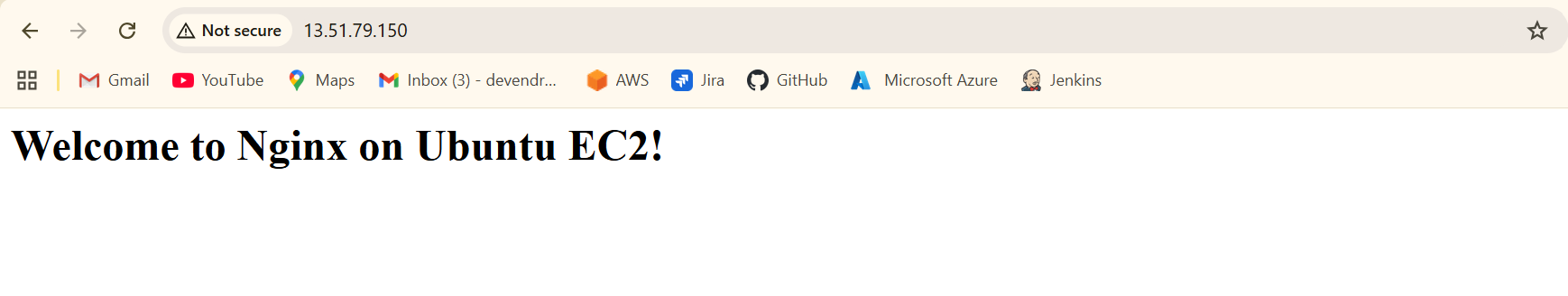
echo "<h1>Welcome to Apache on Amazon Linux 2!</h1>" > /var/www/html/index.html

**Note:** Using # it means gave a comment it will gave you a message who ever check it that one he will understand easily.



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

* Lauch a ec2 instance with a pem key and your public ip.
* Then check the security inbound rules ssh should be 22 and httpd in 80
* Then go to the bottom instance and additional info open it
* And use a bash script for it nginx
* Then go the git bash connect to the server and take the publc ip and paste it with last gave that port number of :80
* Here the results are



* **Bash script for nginx:**

#1/bin/bash

#update system packages

Apt-get update –y

#install nginx

Apt-get install –y nginx

#Enable nginx to start on boot

Systemctl enable nginx

#start nginx service

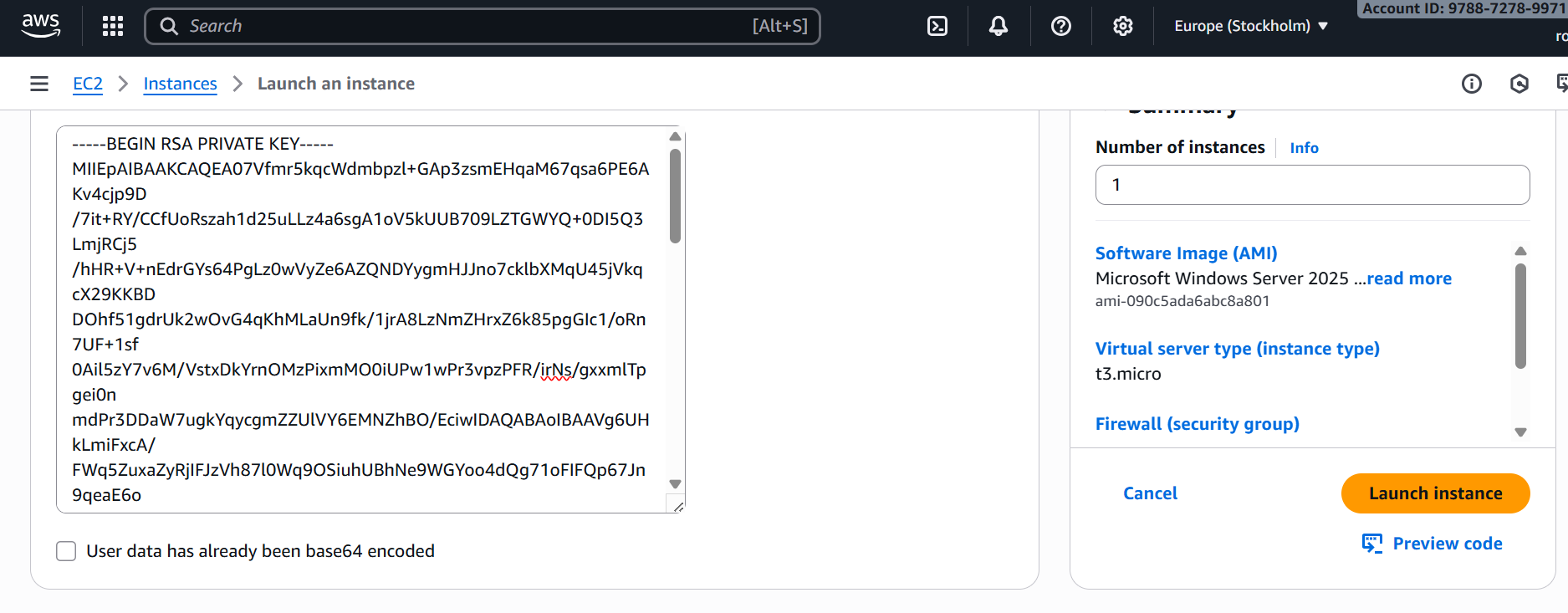
Systemctl start nginx

#ceation a custom index page

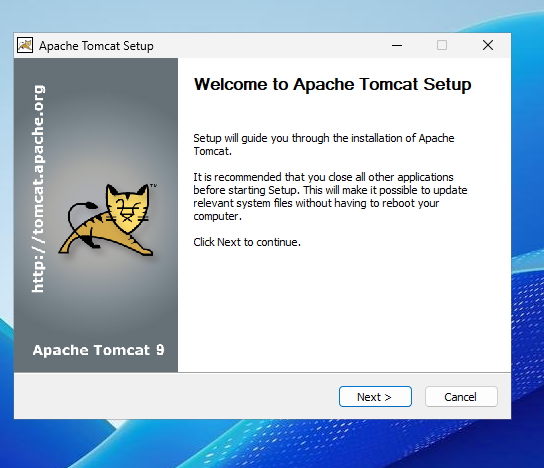
Echo “<h1> welcome to nginx on Ubuntu EC2!</h1>”> /var/ww/html/index.nginx-debian.html

Note: # uses for only gave a comments just want to know the purpose what we are doing in bash scripting.

3) Launch one Windows server and install Tomcat on Windows

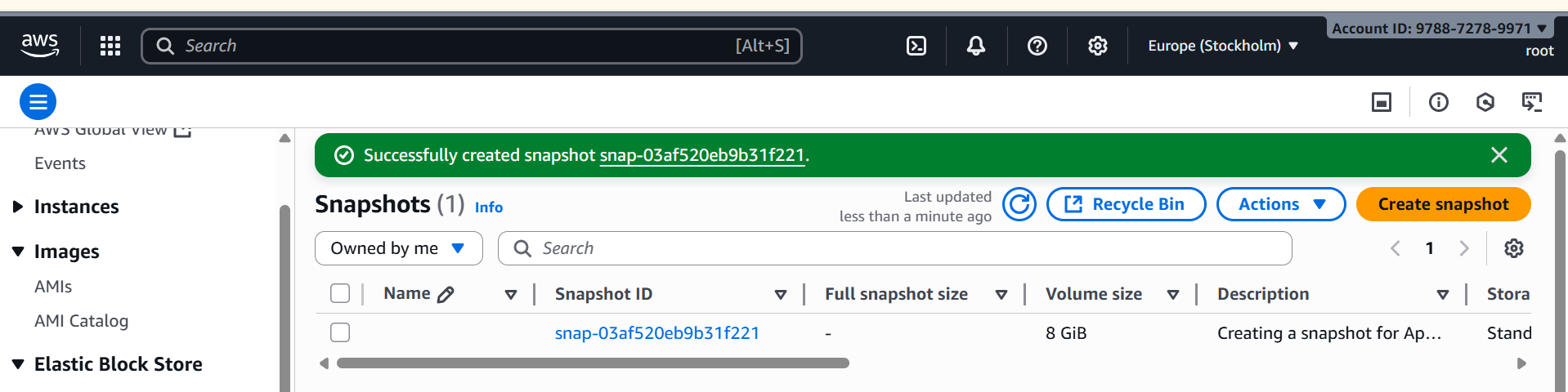






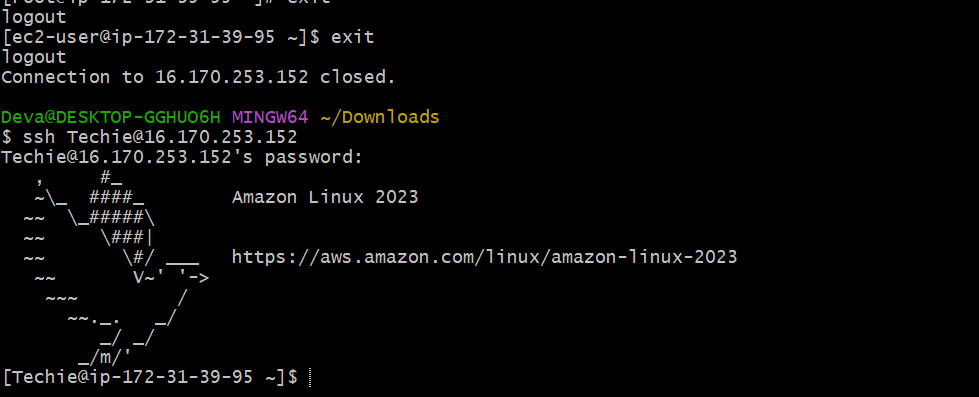
4) Take a snapshot of the instance created in Task 1

* Go to EC2 dashboard 🡪 click on instances
* Select the instance you created in nginx web check
* Scroll down to storage 🡪 find the volume ID (attached root volume)
* Click the Volume Id 🡪 you’ll be redirected to the EBS volumes page.
* Create snapshot and select the instance which one you want here the results are..



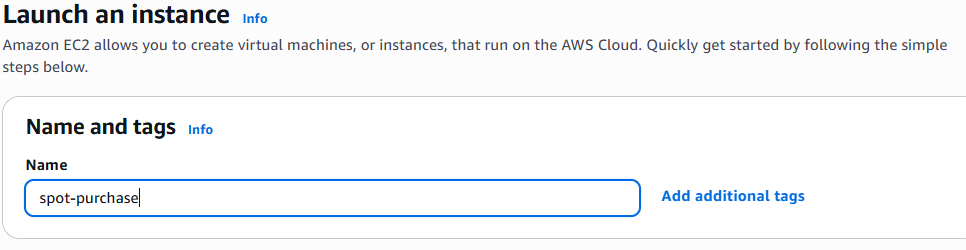
5) Assign password less authentication for the EC2 created in Task 2.

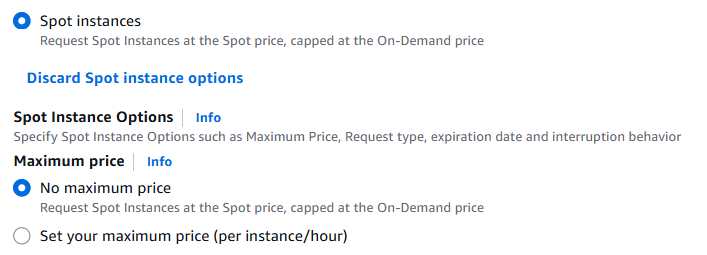
* To create password less authentication:
* Crate a ssh-key in our local machine – ssh –keygen
* Then copy your public key using --- at/c/users/DELL/ .ssh/id\_rsa.pub
* Create an user --- useradd techie
* Password for user --- passwd techie
* Ssh-keygen --- crate a ssh key in ec2 machine
* Vi /root/.ssh/id\_rsa.pub (paste your local machine key here by keeping)
* Present key as same ..
* Vi/etc/ssh/sshd\_config --- enable password authentication as yes
* Systemctl restart sshd --- restart your machine :
* Ssh techie@public-ip
* Allow fingerprint authentication :yes

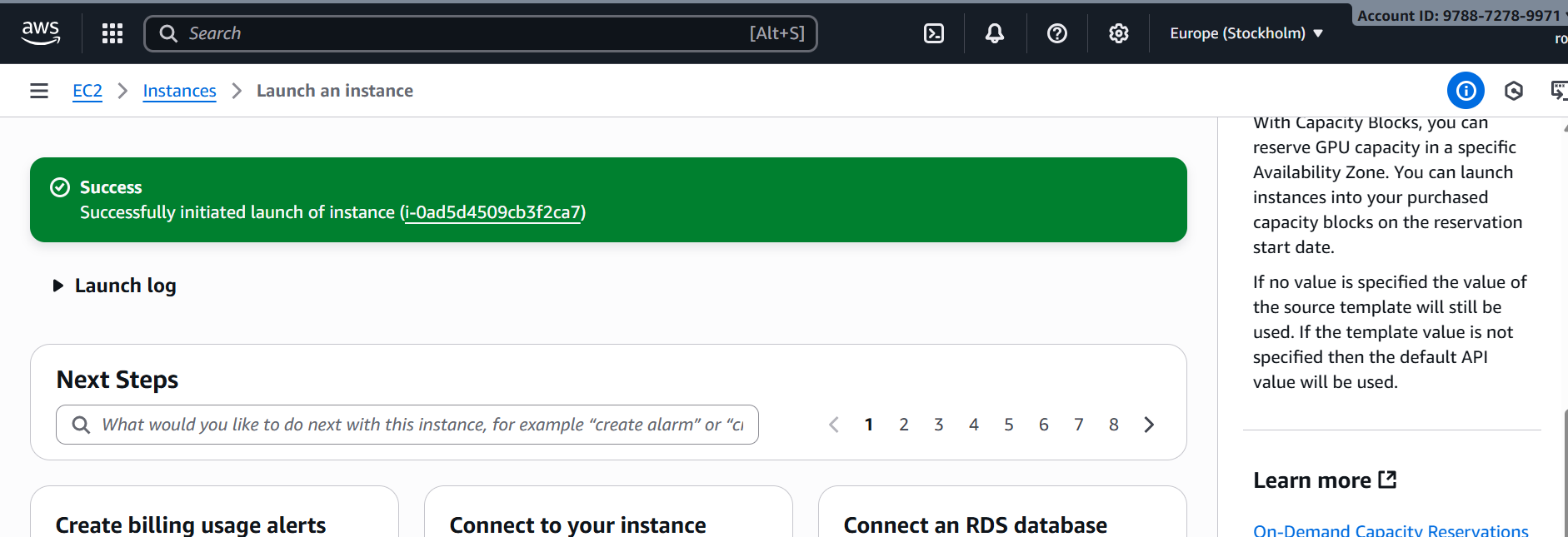


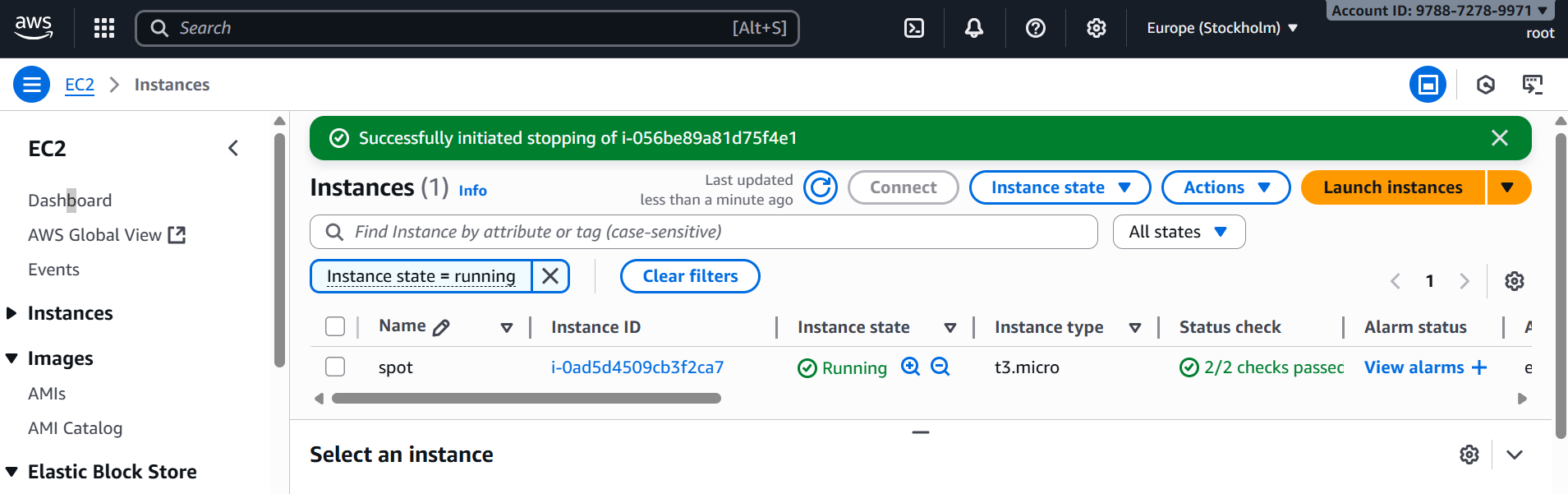
6) Launch any EC2 using the spot purchasing option.

* Click on launch instance
* By using spot instance launch one instance



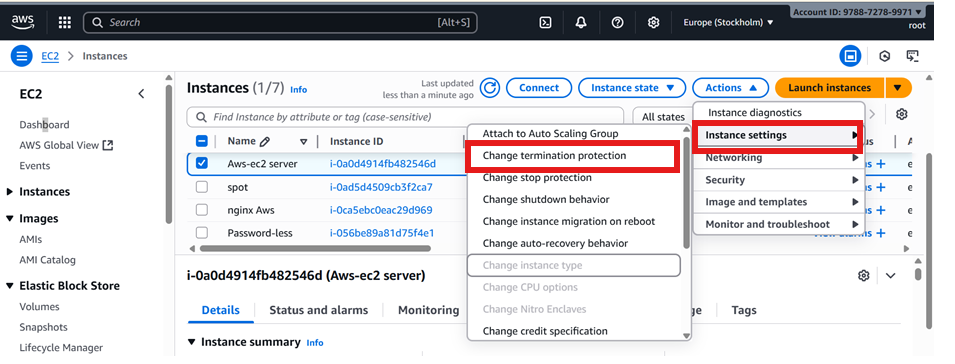


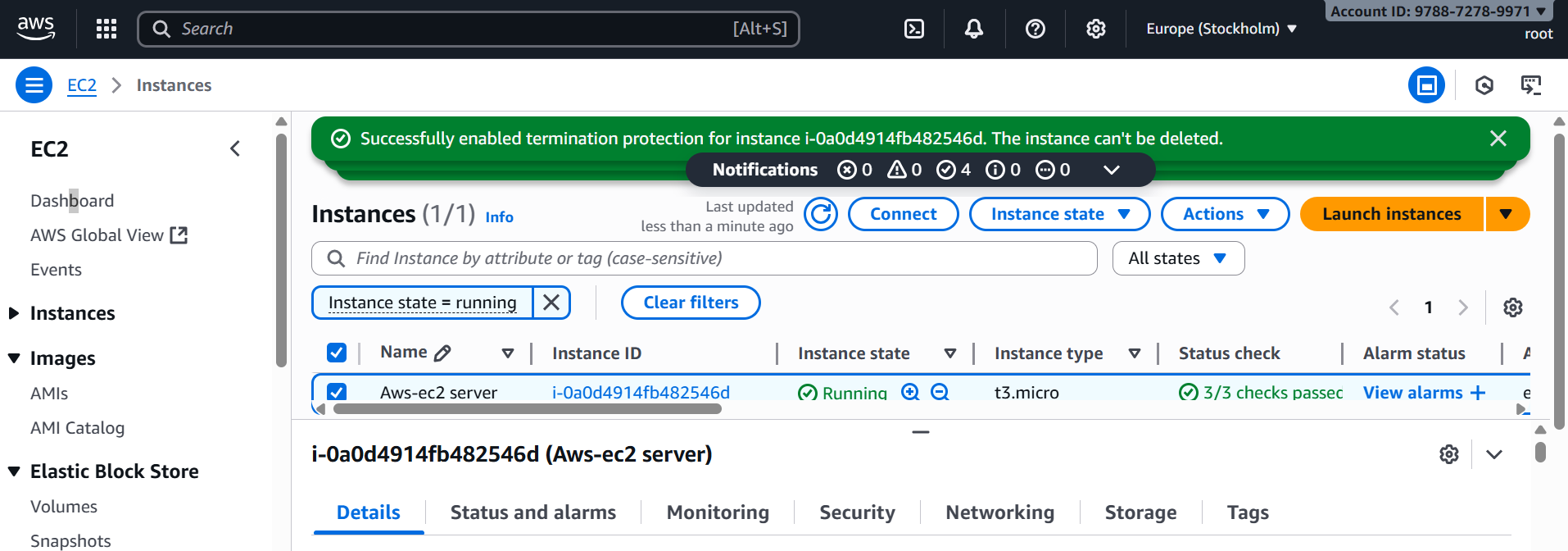




7) Enable termination policy on the EC2 created in Task 2.

* Go to the EC2 Dashboard in AWS console
* Select the EC2 instance you created in task 2
* In the instance settings menu, click change termination protection
* Select Enablr and save .





8) Launch one EC2 using AWS CLI.

* Launch instance name with AWS
* And to connect to the server then
* Wget download cli [**https://awscli.amazonaws.com/AWSCLIV2.msi**](https://awscli.amazonaws.com/AWSCLIV2.msi)
* Then gave a command of aws configure it will show you a options like
* Access key
* Secret access key
* Region and format
* For access key and secrete key open our profile then security credentials and it will generate the keys
* Then aws ec2 describe-instances
* Here the results

