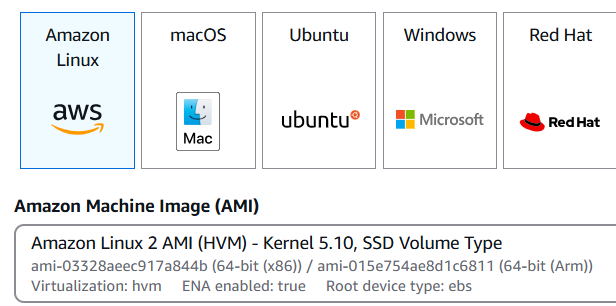
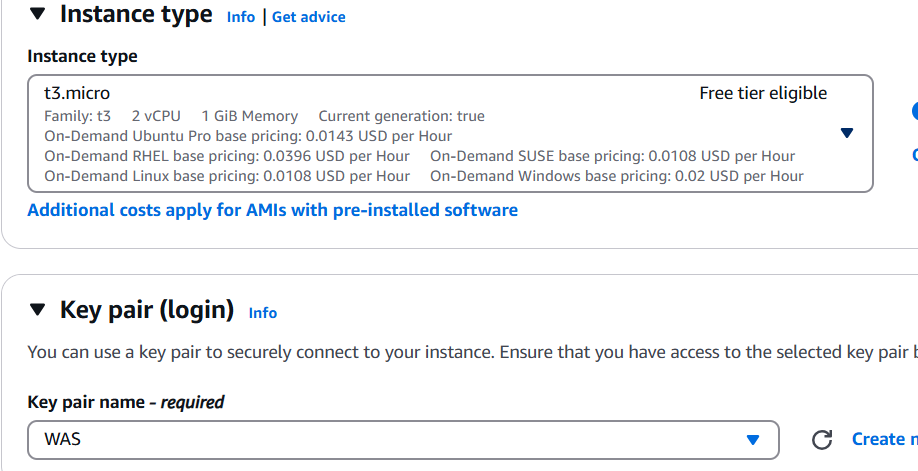
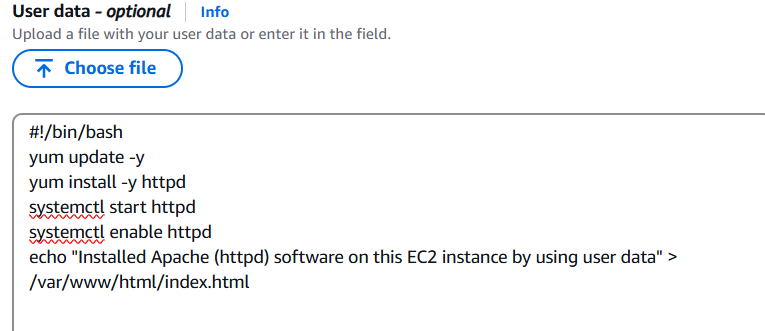
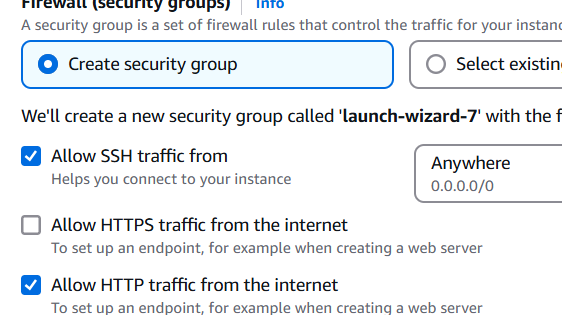
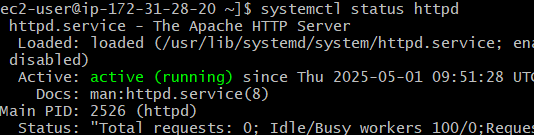
**TASKS ON EC2**

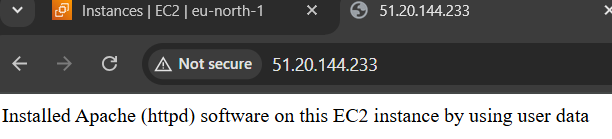
1) Launch one ec2 using Amazon Linux 2 image and add script in user data to install Apache.

* click on launch instance and select amazon linux 2
* select t2.micro or t3.micro(depends on the region)
* choose a key pair or make a new one
* select or create a security group to allow the necessary ports
* under advance setting , go to user data add the script for installing apache and launch the instance
* SSH into the Amazon Linux 2 instance and check the status of apache or copy the public IP of your ec2 instance with the port number to check if its running

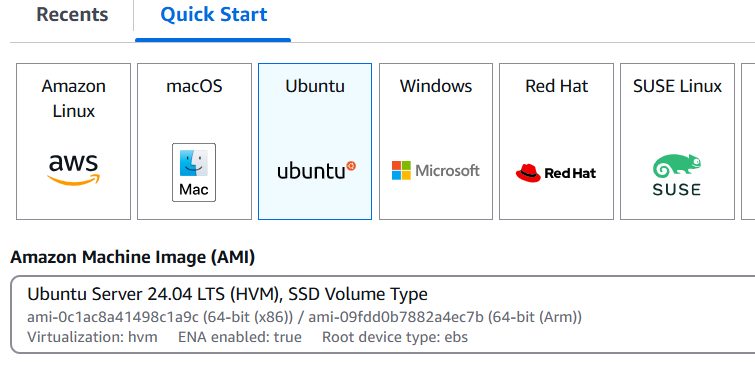
 

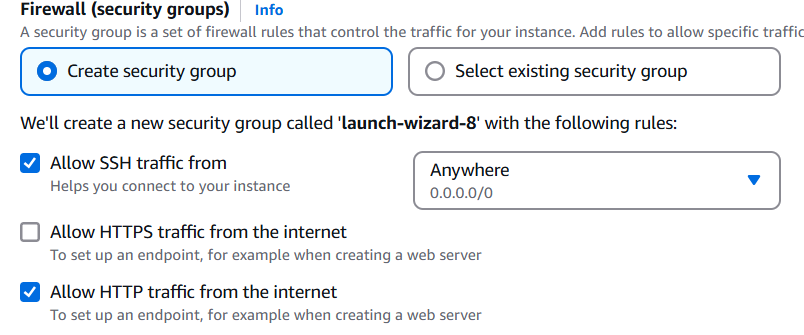
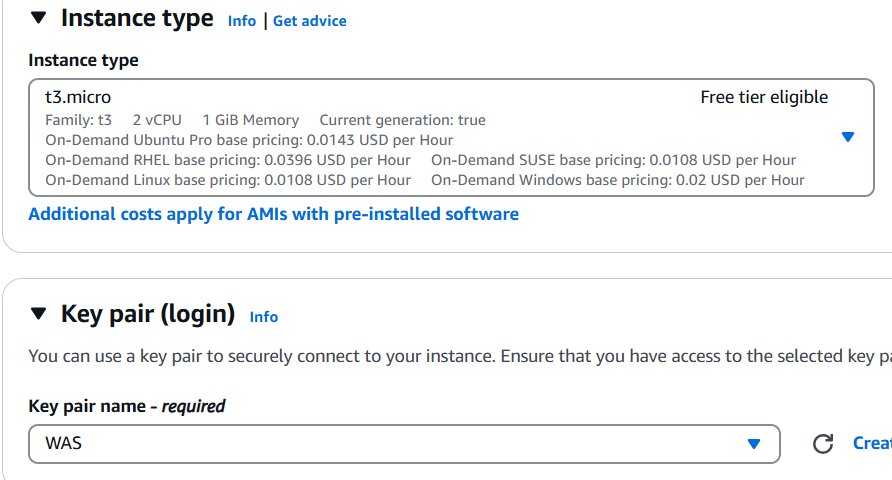


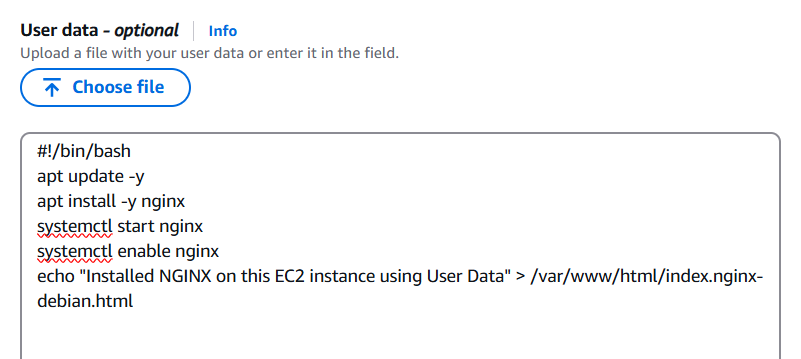


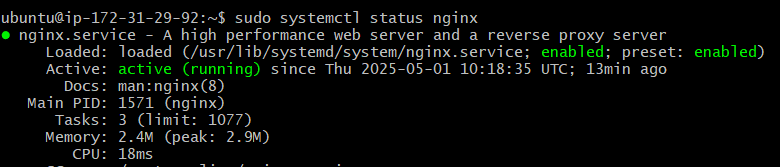


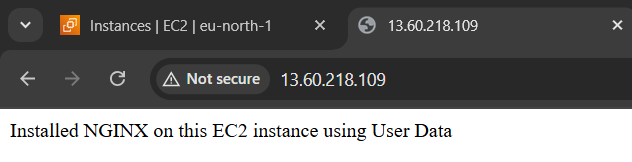
2) Launch one ec2 using Ubuntu image and add script in user data to install Nginx.

* click on launch instance and select Ubuntu
* select t2.micro or t3.micro(depends on the region)
* choose a key pair or make a new one
* select or create a security group to allow the necessary ports
* under advance setting , go to user data add the script for installing nginx and launch the instance
* SSH into the Ubuntu instance and check the status of nginx or copy the public IP of your ec2 instance with the port number to check if its running
* 



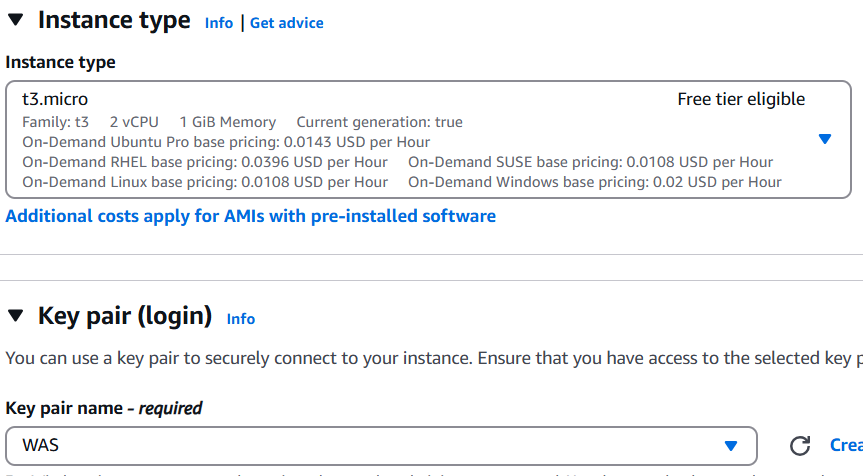
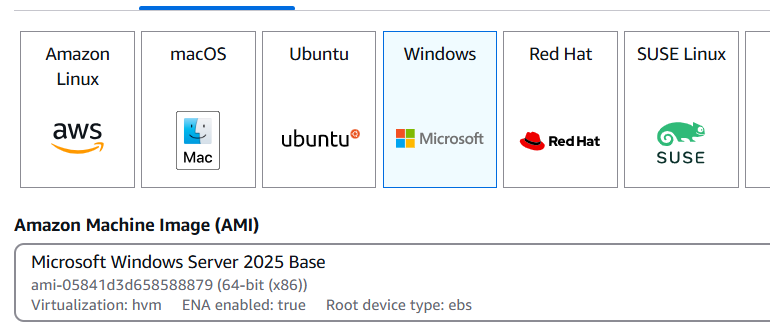


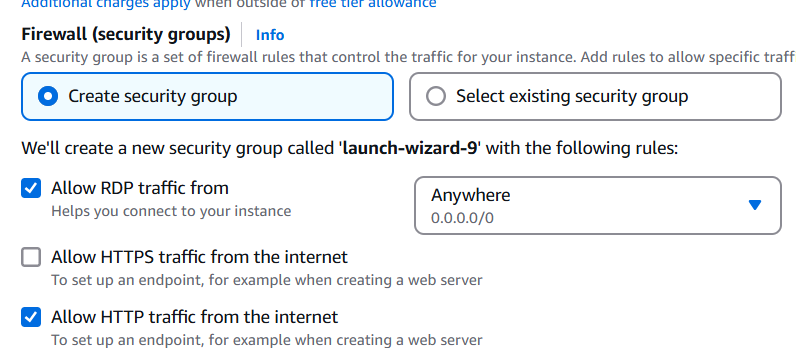


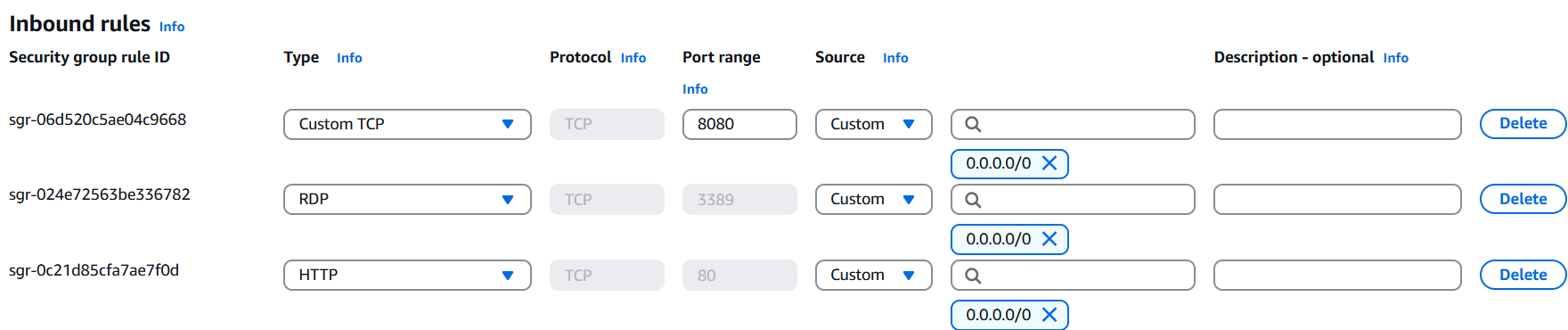


3) Launch one windows server and install tomcat in windows.

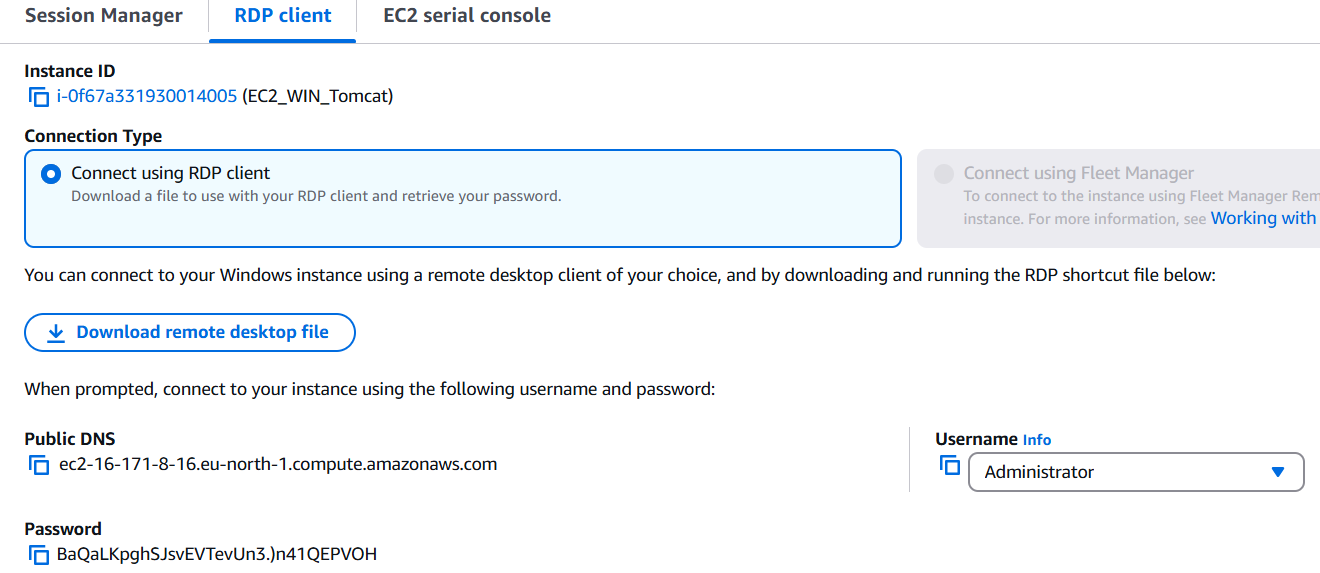
* click on launch instance and select Windows
* select t2.micro or t3.micro(depends on the region)
* choose a key pair or make a new one
* select or create a security group to allow the necessary ports, also add port 8080 to the inbound rules on aws

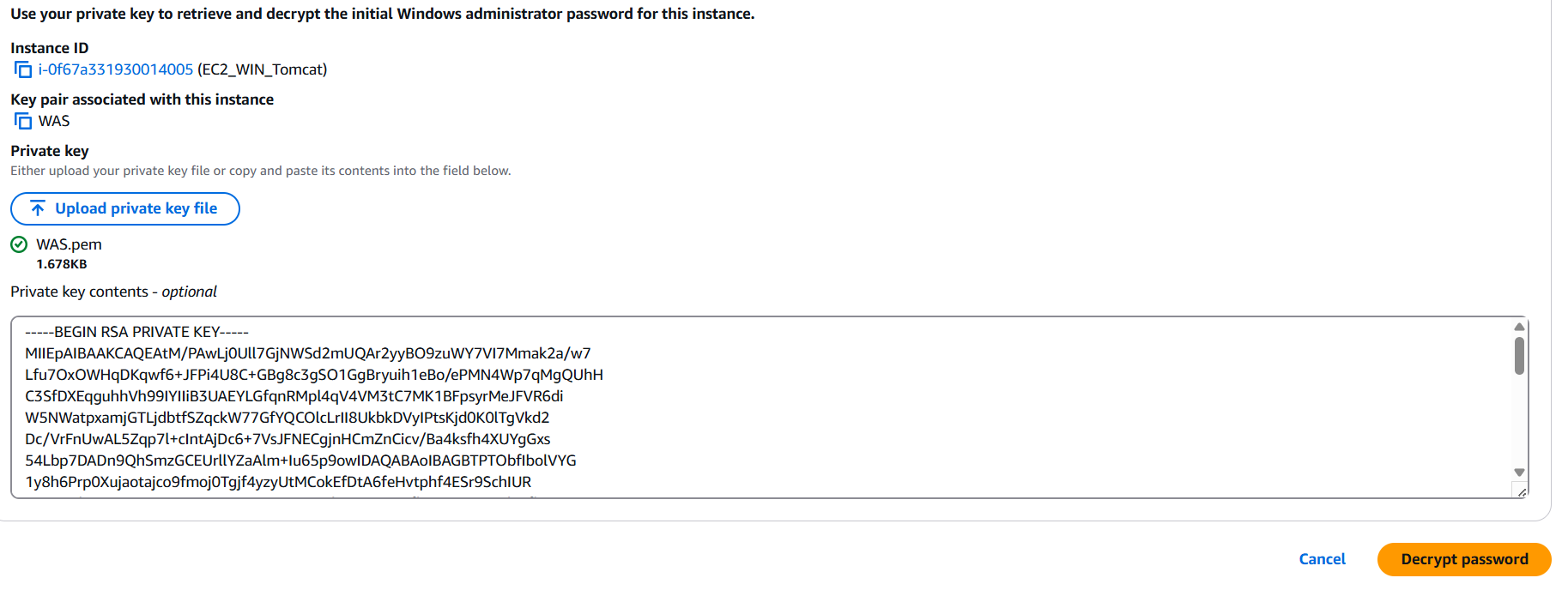


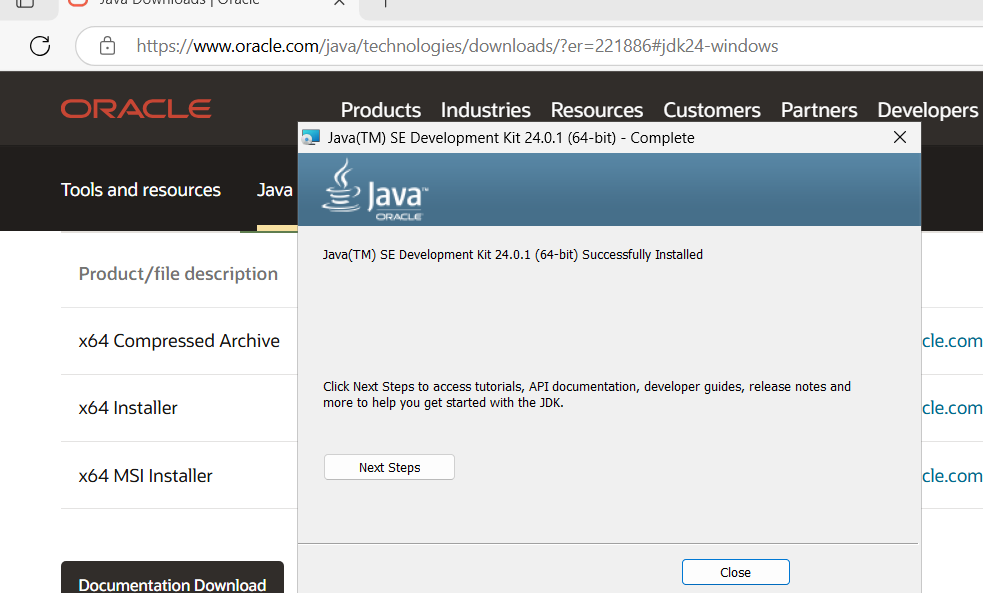


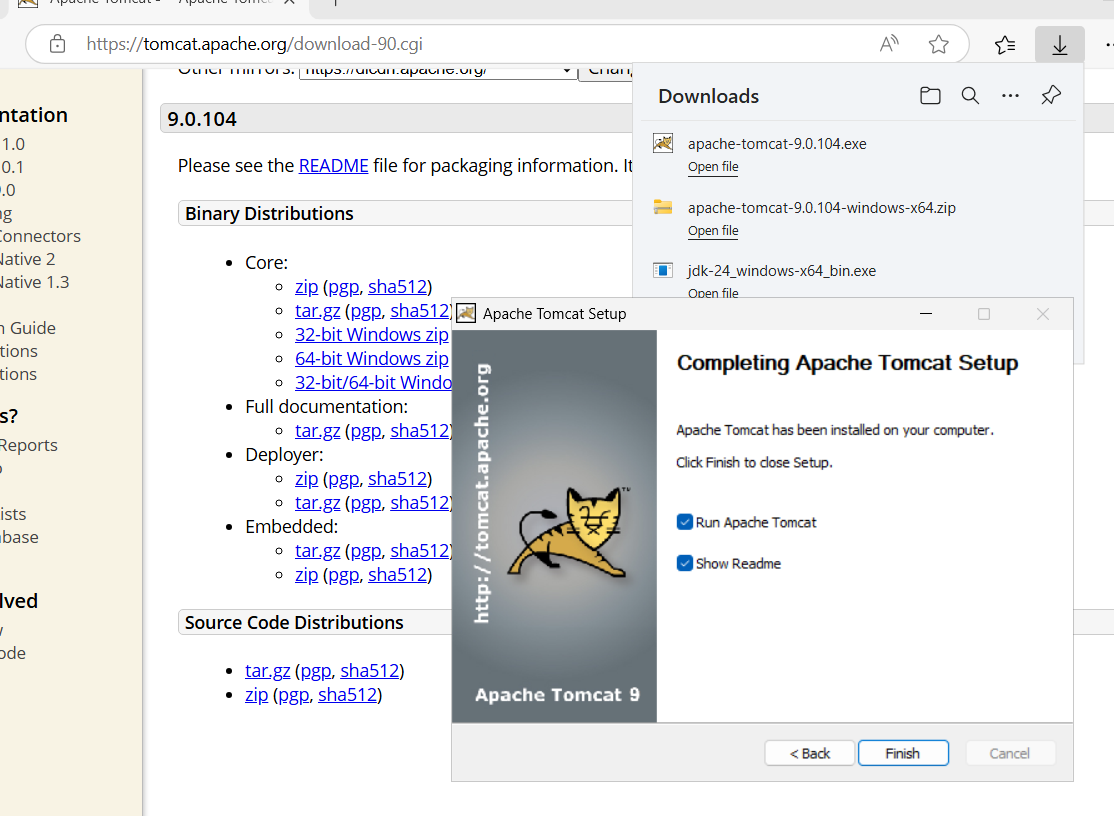


* launch instance and download the RDP(remote desktop client) file
* click on get password, wait for 4 mins, upload your pem key and to get the administrator password, use that to login to the windows server
* in the win. server , go to edge and download and install java from oracle website then download tomcat from its website and install it and run it.

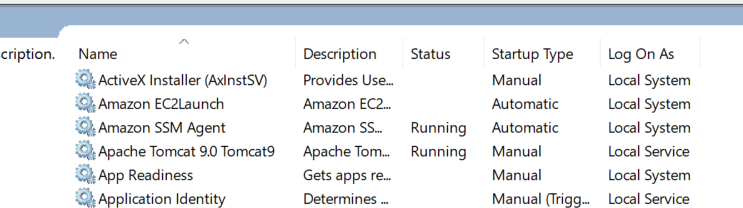


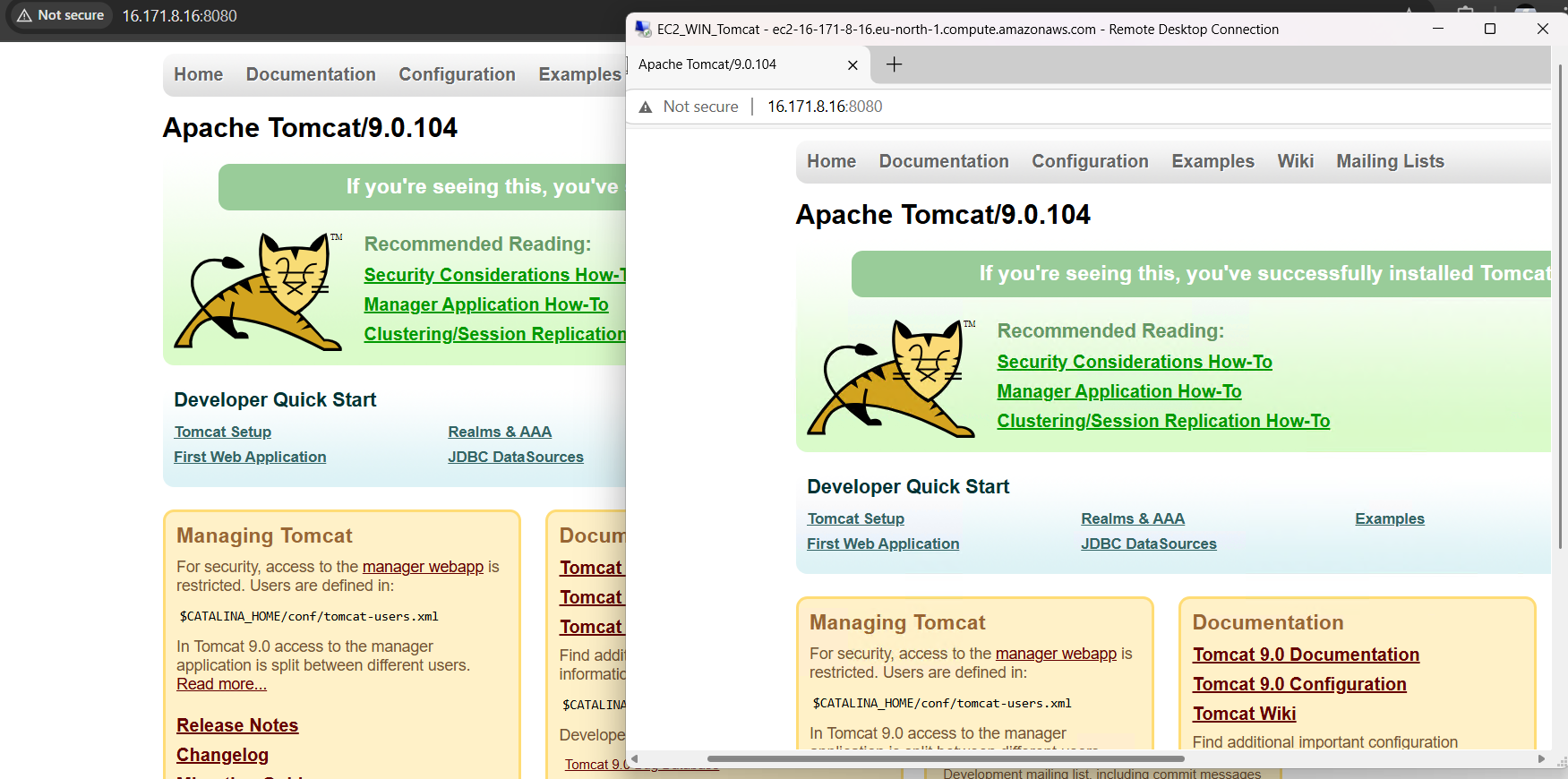






* check if its running by checking the services tab ,also add port 8080 in the windows firewall settings under inbound rules and it should run on the browser inside the remote server and outside too

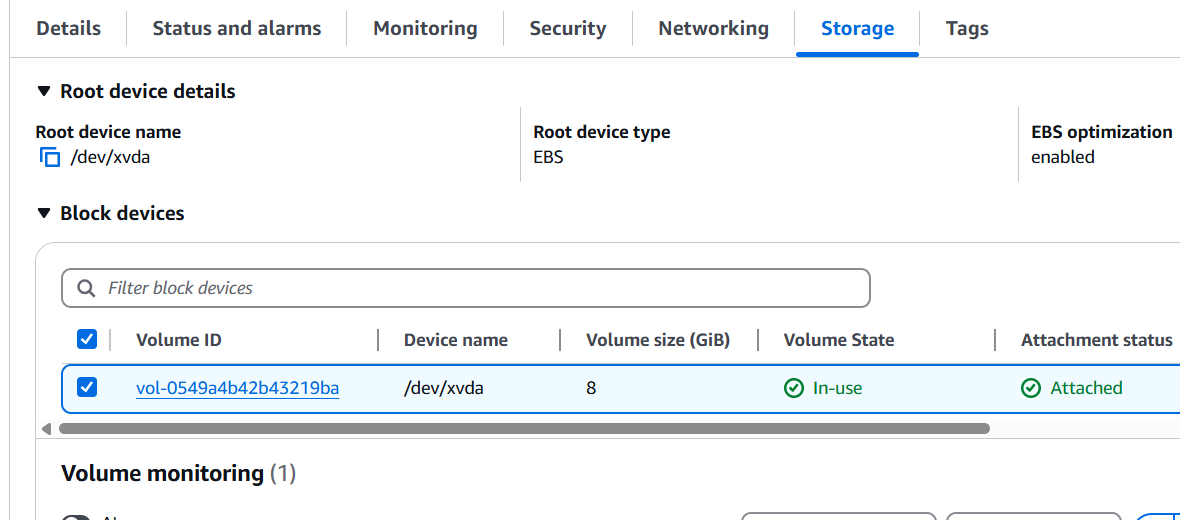


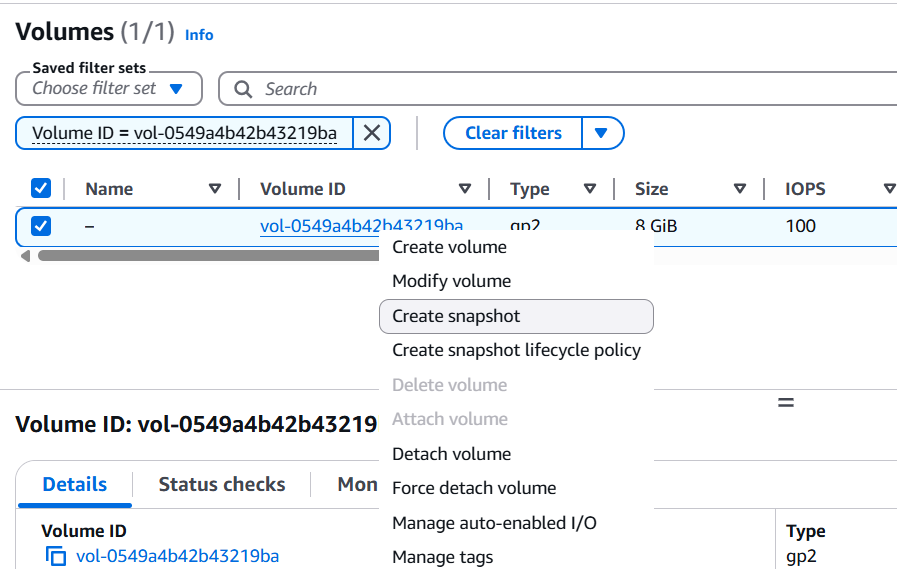


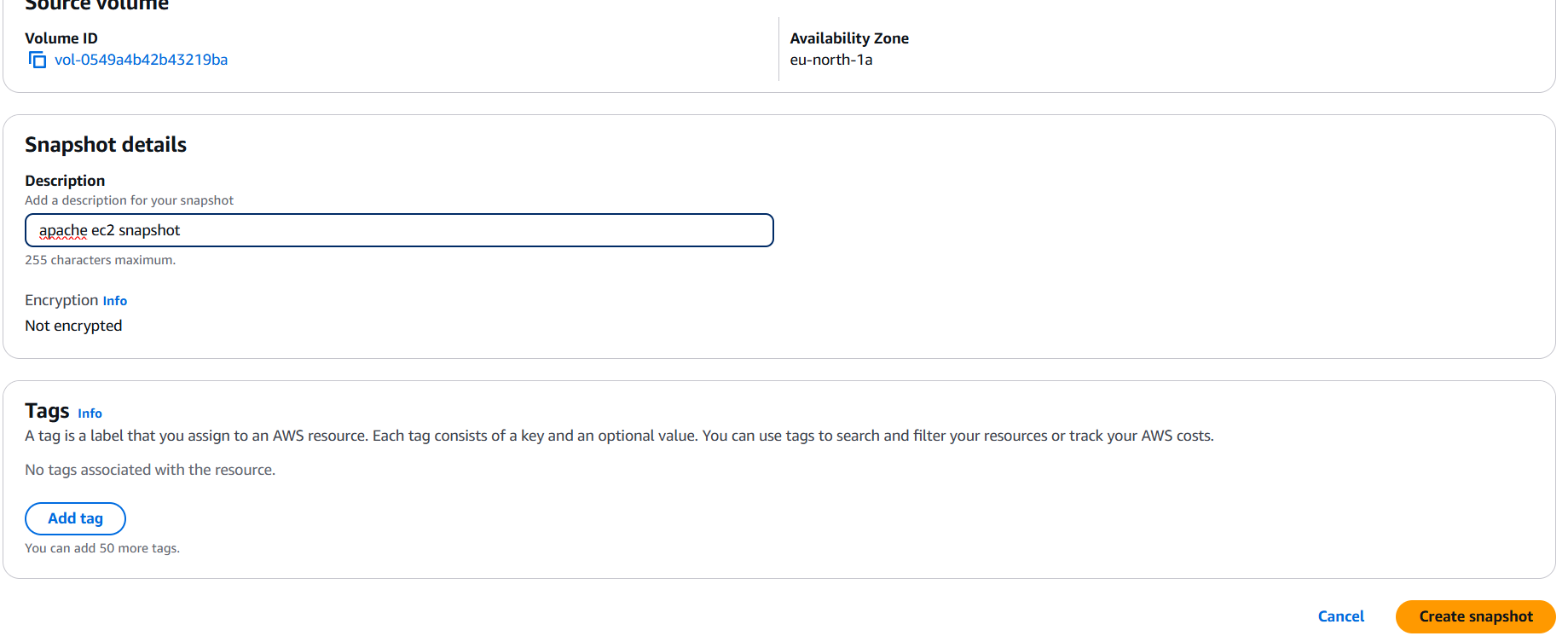
4) Take snapshot of the instane created in Task 1.

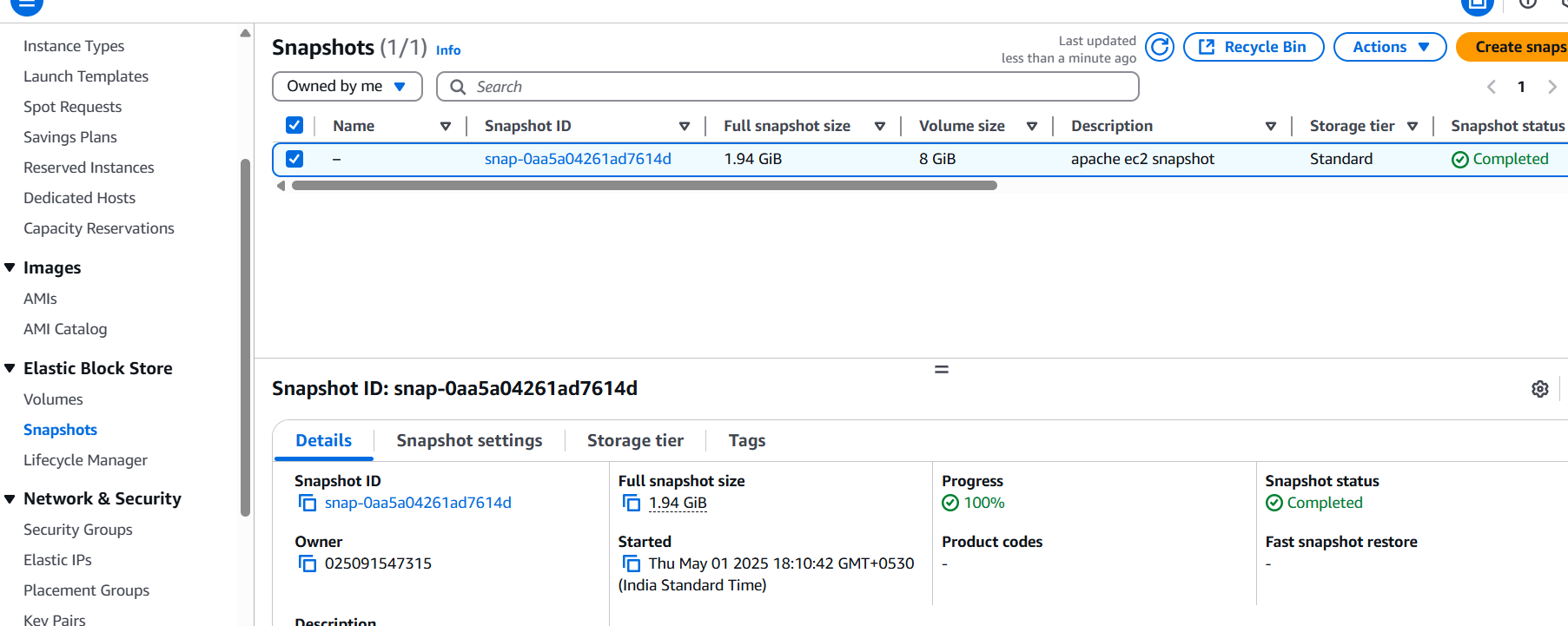
Snapshot capture the current state of the data at a specific moment and can be used to restore data to a previous state if needed

* select the apache task instance
* go down to the storage tab and click on volume ID, a new volume page opens , right click on the ID and select create snapshot.
* add description and tags if you want and create snapshot





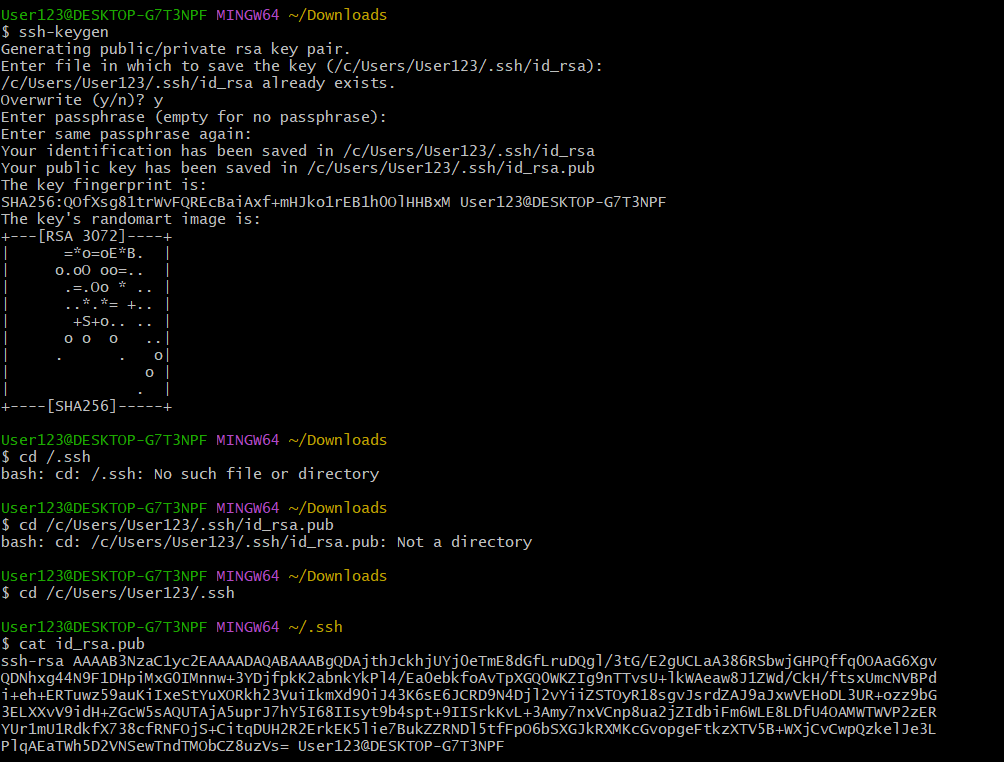




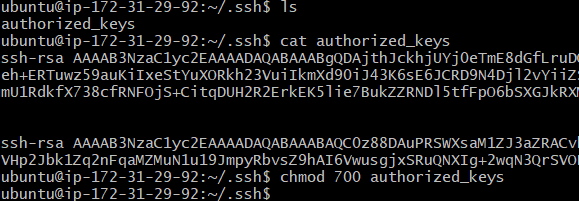
* snapshots created can be viewed by visiting the snapshots tab on the left hand side

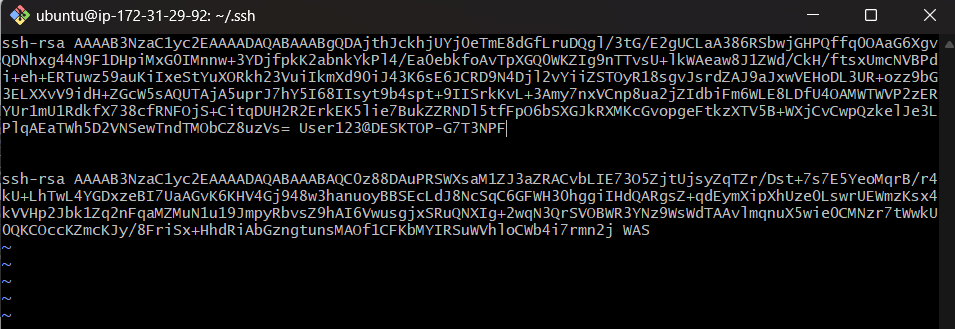
5) Assign password less authentication for ec2 created on Task 2.

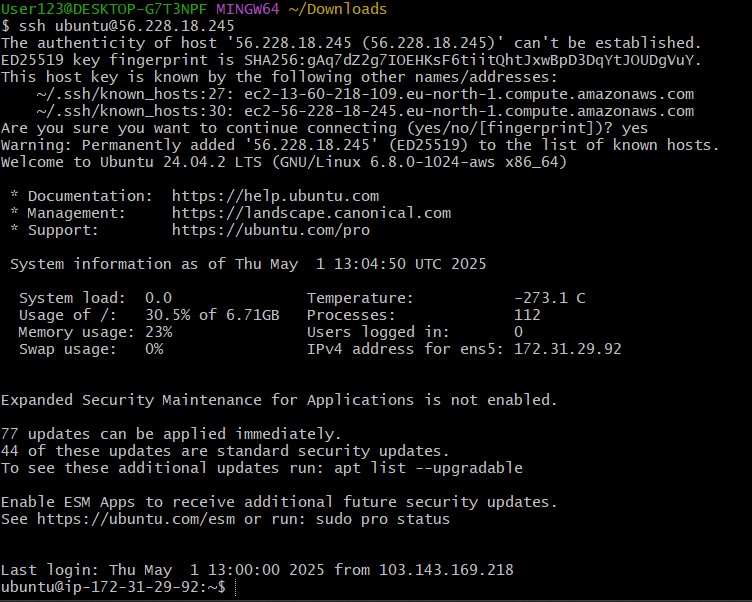
* create a public key pair using ssh-keygen on the local system
* got to its location and copy the public key



* ssh into the instance and add the public key to the authorized\_keys on that ec2 instance, save and quit.
* also change the authorized key permission to 600.
* exit from the instance and now log in to the instance using local machine by ssh ubuntu@<public-ip>
* If it works without asking for a password or pem key file then we have successfully set up password-less authentication



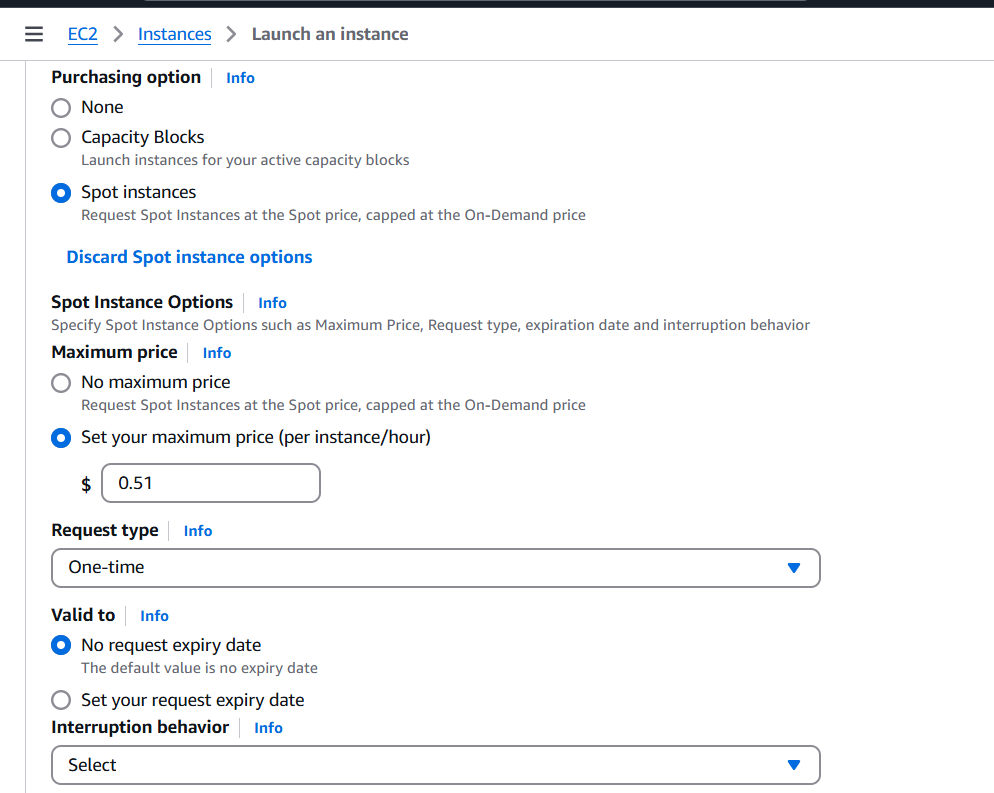


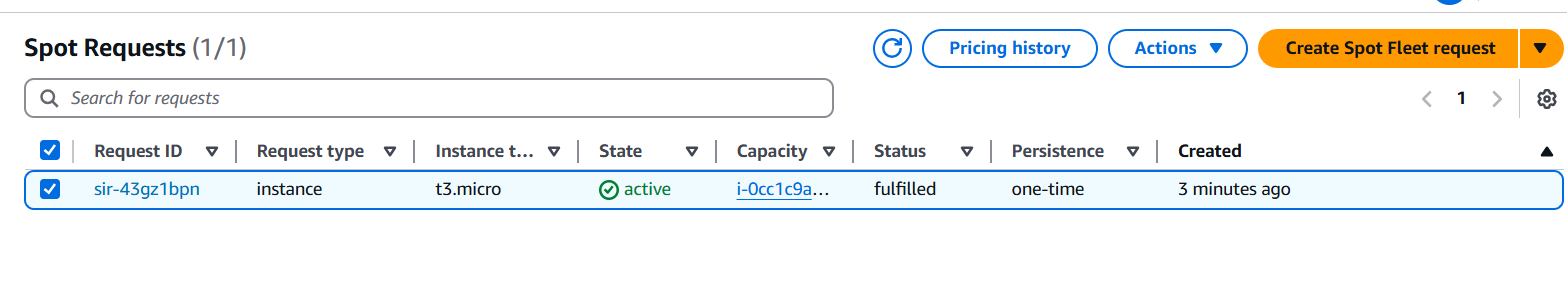


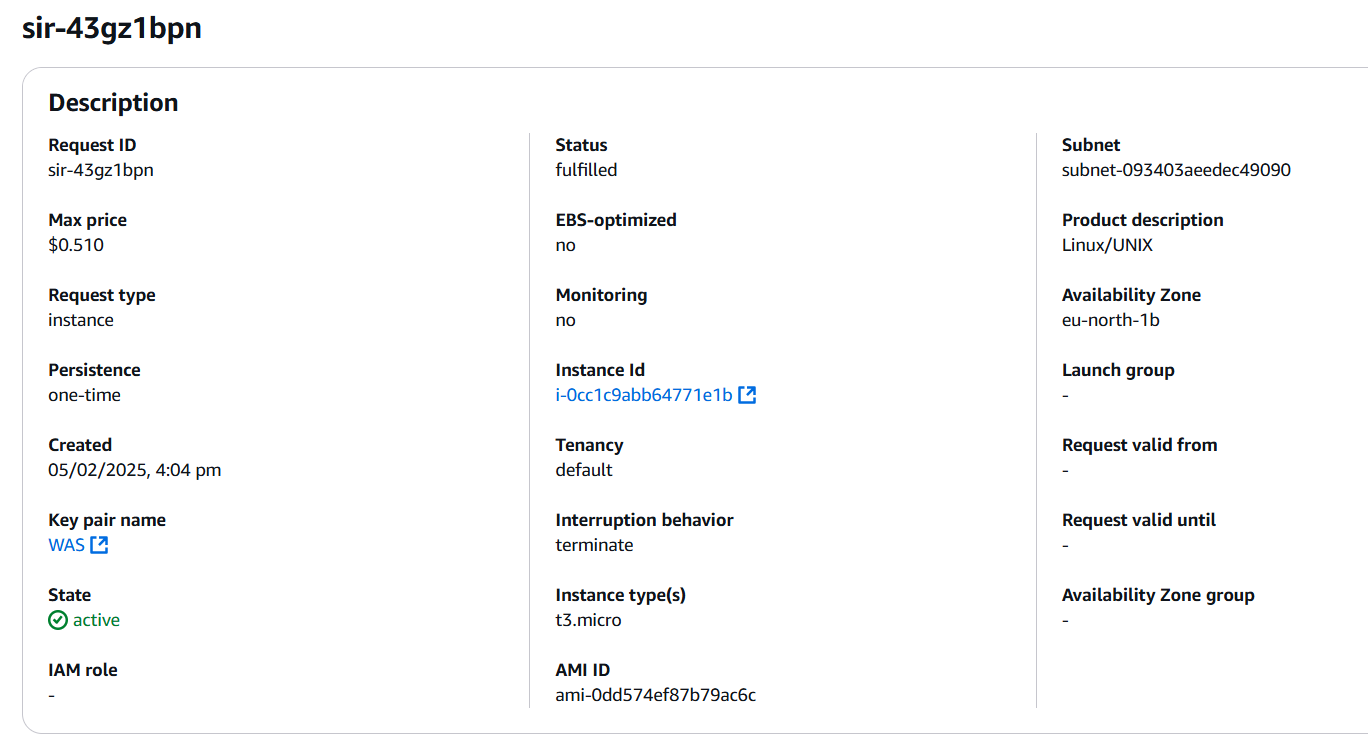
6) Launch any ec2 using spot purchasing option.

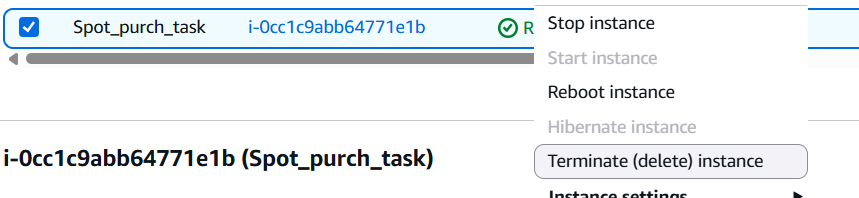
Spot Instances allow you to bid for unused EC2 capacity at a significantly lower price than On-Demand instances. However, they can be terminated by AWS if the spot price exceeds your bid or if there is no longer unused capacity available.

* launch an instance like you usually do and go to the advanced details and choose spot instances under purchasing option.
* click on options and select your price/hour and request type , either one time or persistent . launch it now.
* To stop it you need to terminate as its a one time request and not persistent which you can start and stop.



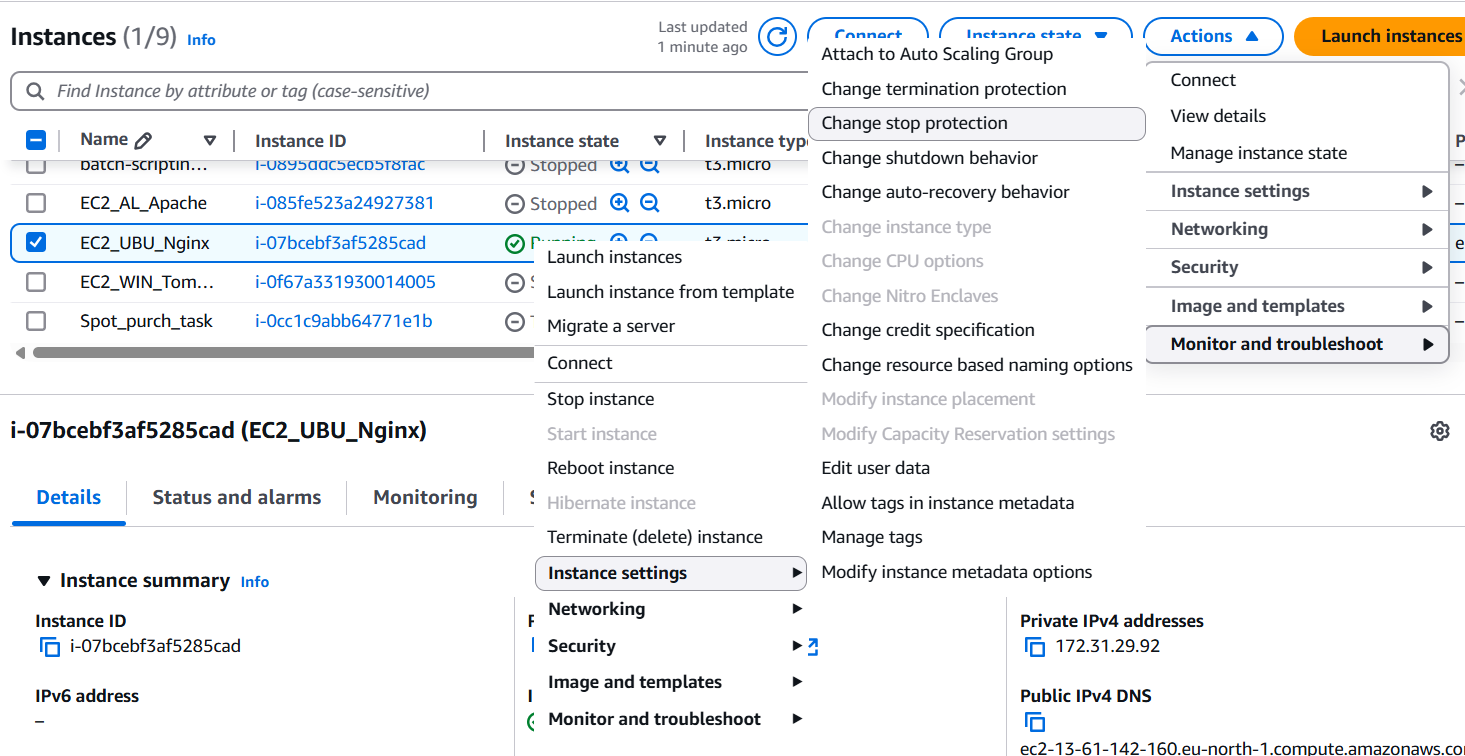


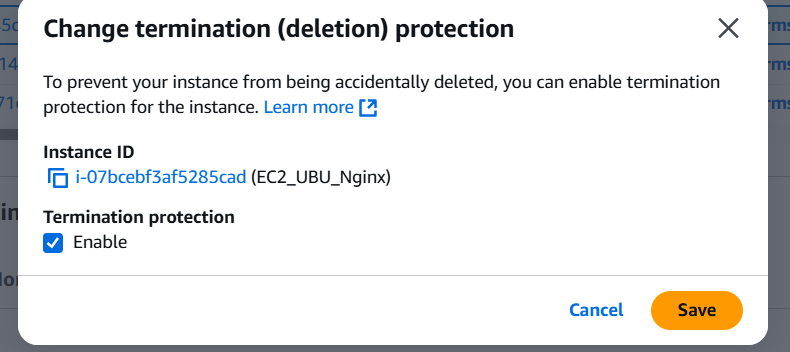


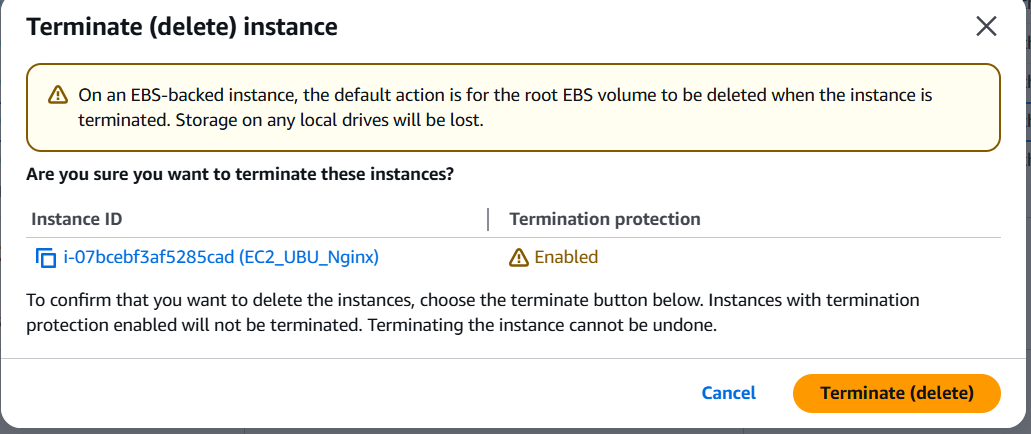


7) Enable Termination policy on ec2 created in Task 2.

* Select the tasks 2 (ubuntu) instance and right click or click the action button on top right and go under instance settings
* click on change termination protect, check the enable option and save
* to verify this, try to terminate it but it wont get terminated .



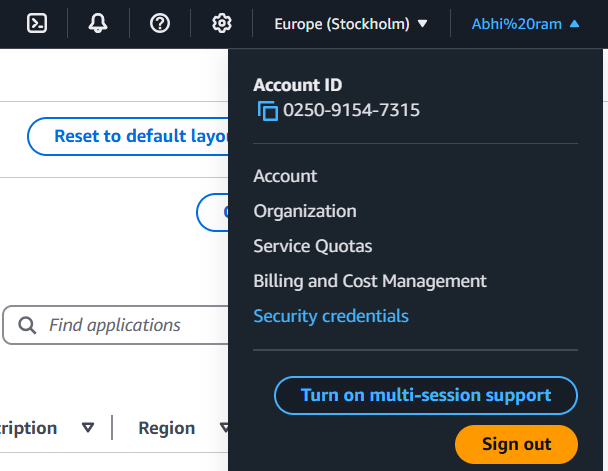


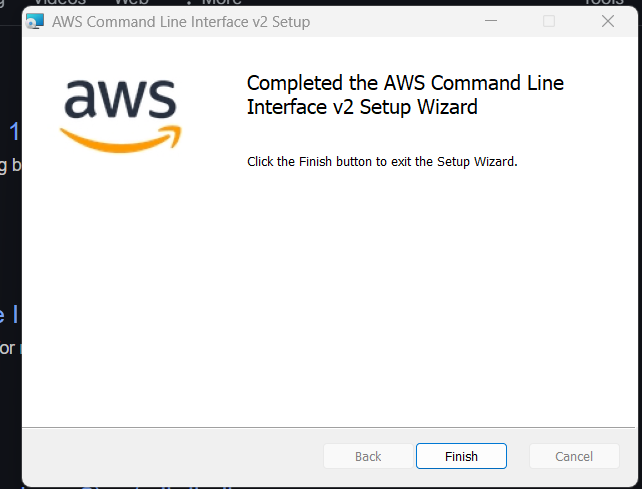


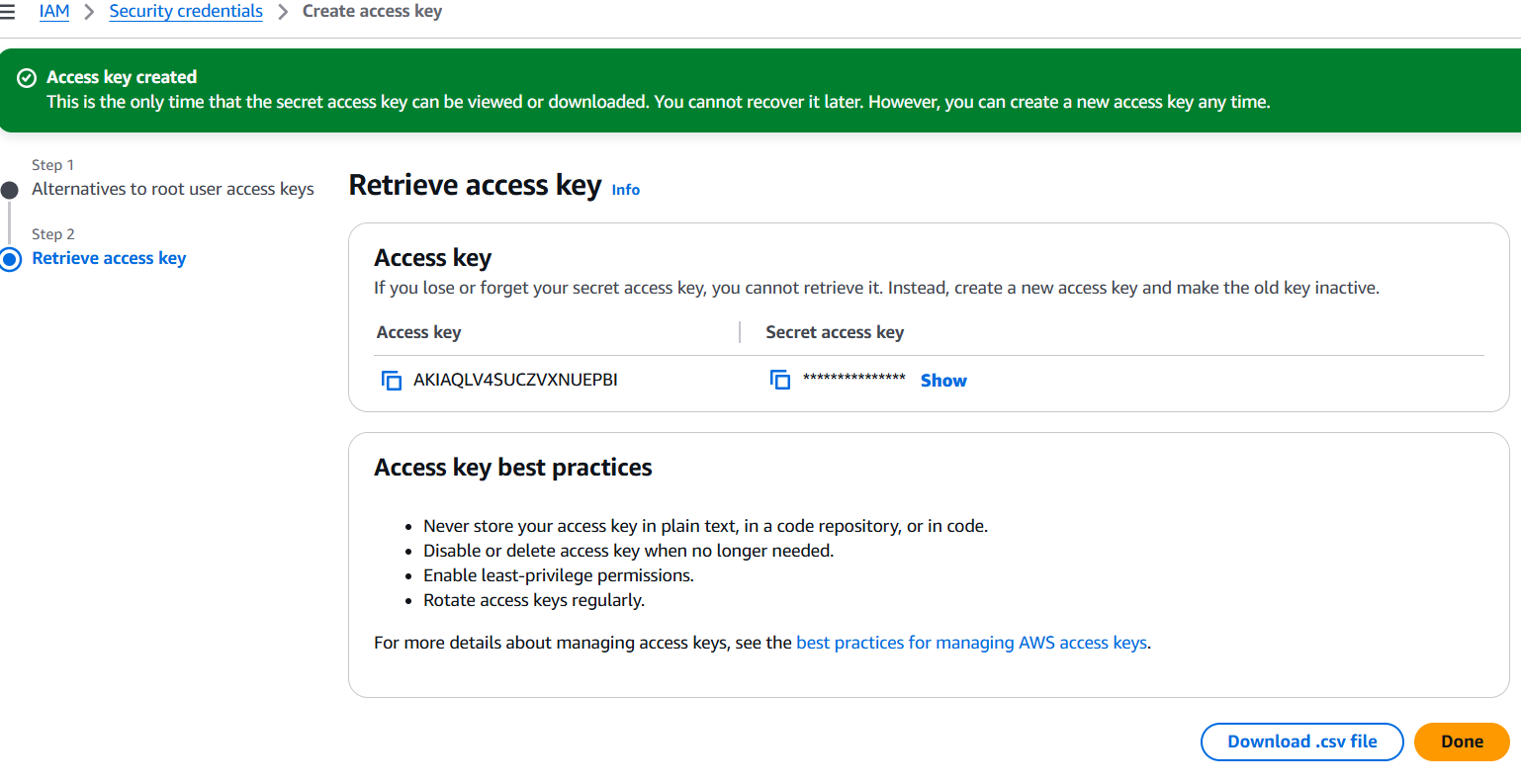
\*\*Terminate protection or termination policy?(doubt)\*\*

8) Launch one ec2 using Aws CLI.

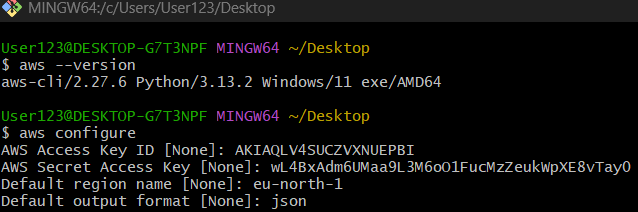
* Download AWS CLI from their website (windows version) and install it
* Go to you AWS home page and click on your name and select security credentials(top right corner)
* Scrolls down to find the access key section and create one, copy or just download the .csv file







* your aws credentials will be stored in C:\users\<username>\.aws\credentials
* Enter the access key id and the secret key along with the region and output format(json/text/table) using the aws configure command



* run this command structure -

aws ec2 run-instances \

--image-id ami-0dd574ef87b79ac6c \ ##what os or ami you want

--count 1 \ ##no. of instances needed

--instance-type t3.micro \ ##instance type t2 or t3

--key-name WAS \ ## my .pem key (private key)

--security-group-ids sg-016a6c8c0f343e16c \ ## group for ports

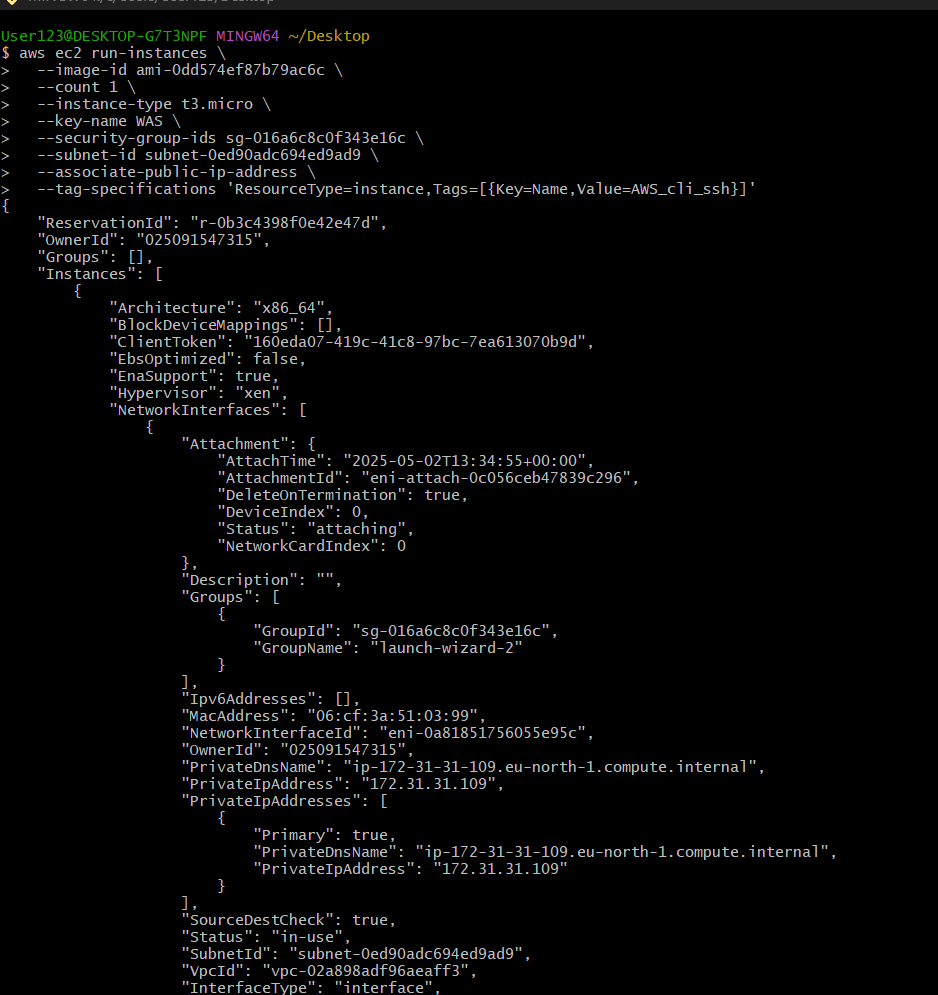
--subnet-id subnet-0ed90adc694ed9ad9 \

--associate-public-ip-address \ ##if sometimes subnet that does not auto - assign public ip

--tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=AWS\_cli\_ssh}]'

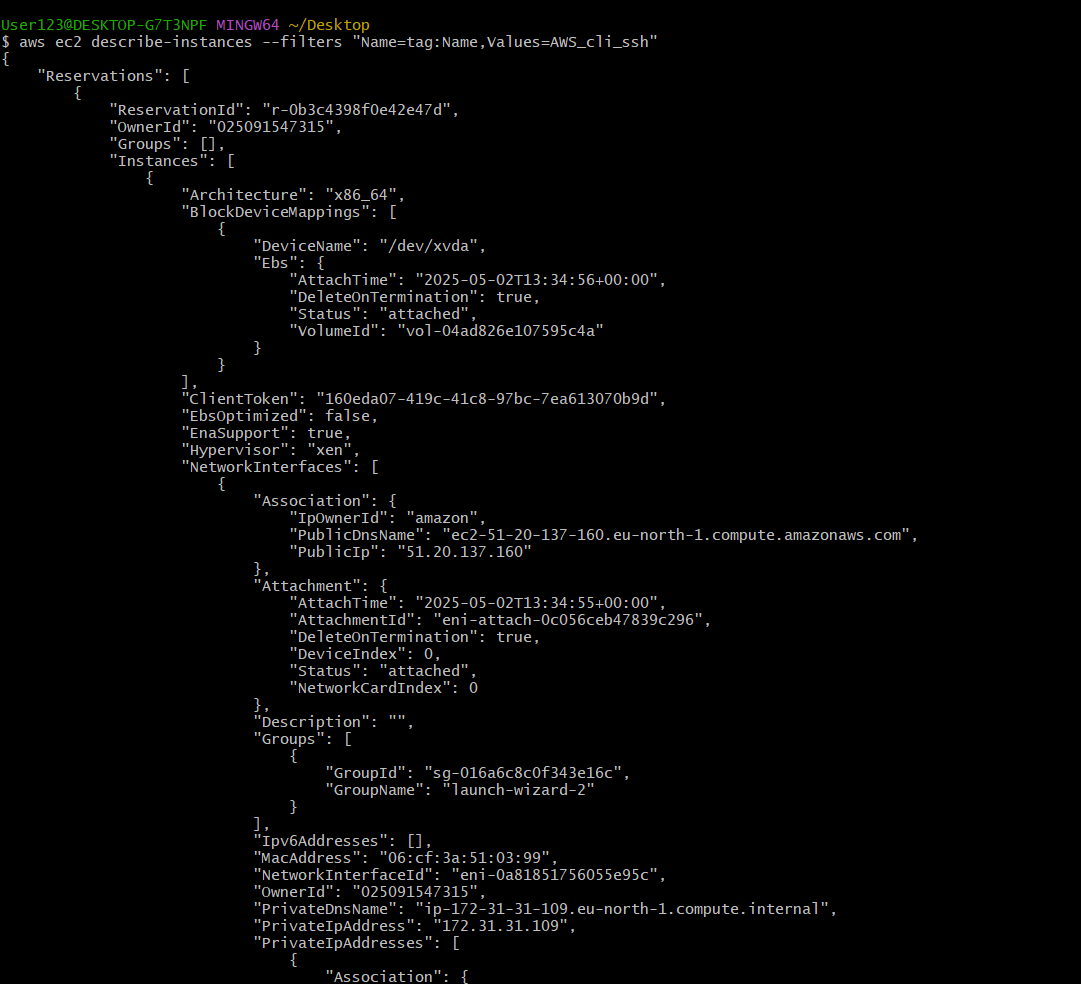
## to name my server, identification reasons

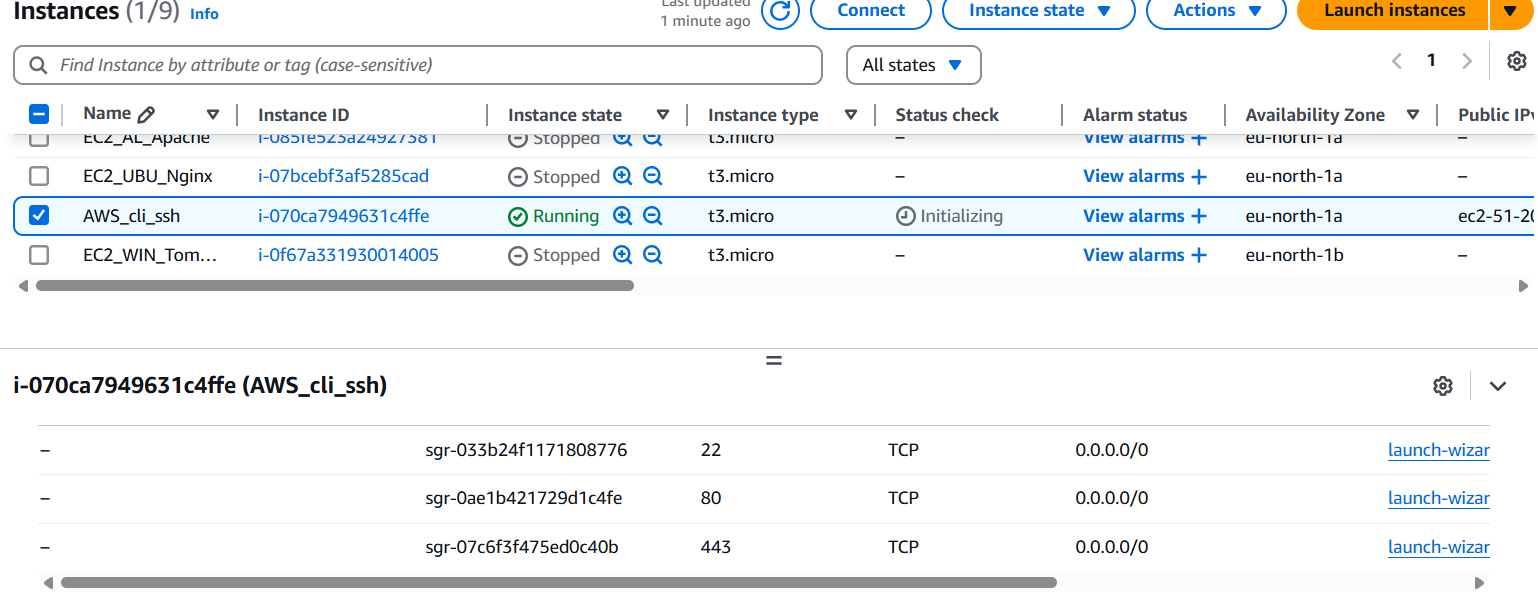
Also \ (backwards slash) is used to tell gitbash that command runs in next line





* aws ec2 describe-instances --filters "Name=tag:Name,Values=AWS\_cli\_ssh" to check if its launched, it will show the public,private IPs and some other details.





* it show should up on the instances webpage too
* now ssh into it like usual using - ssh -i <creasted-key>.pem ec2-user@<public-ip>

