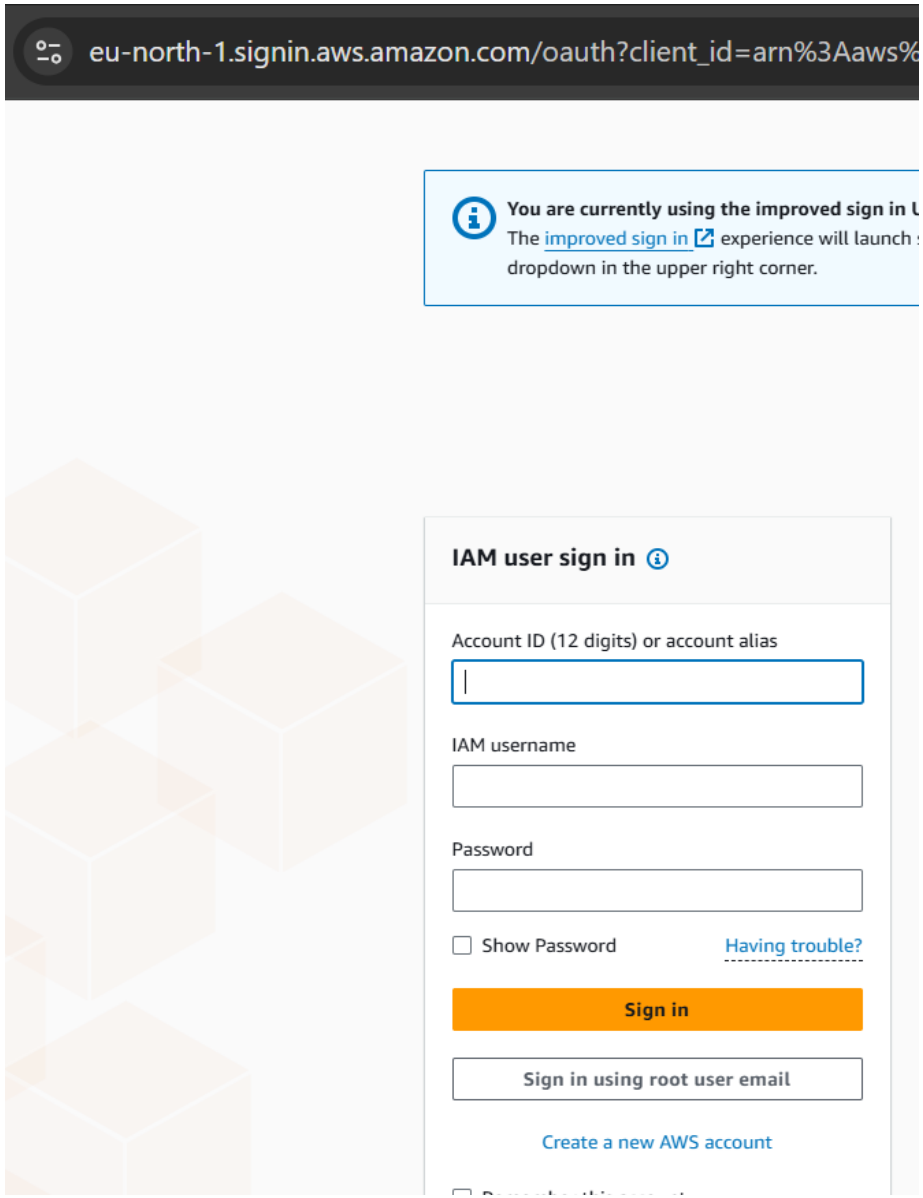
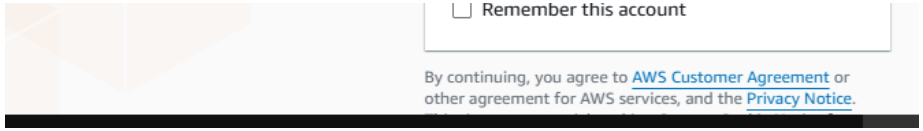
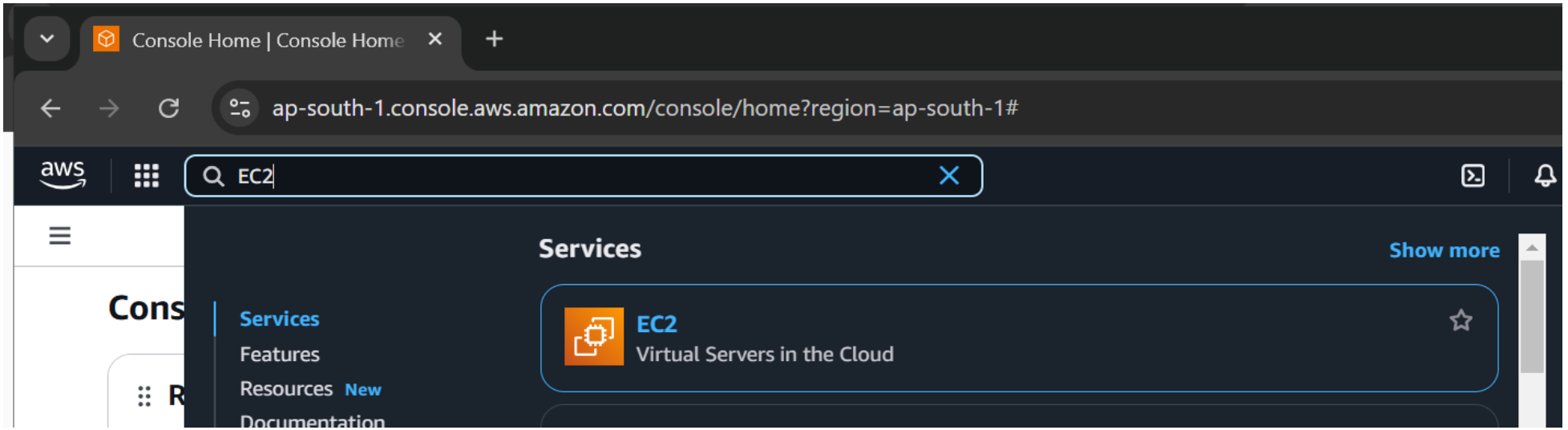
***) Go to https://console.aws.amazon.com/**

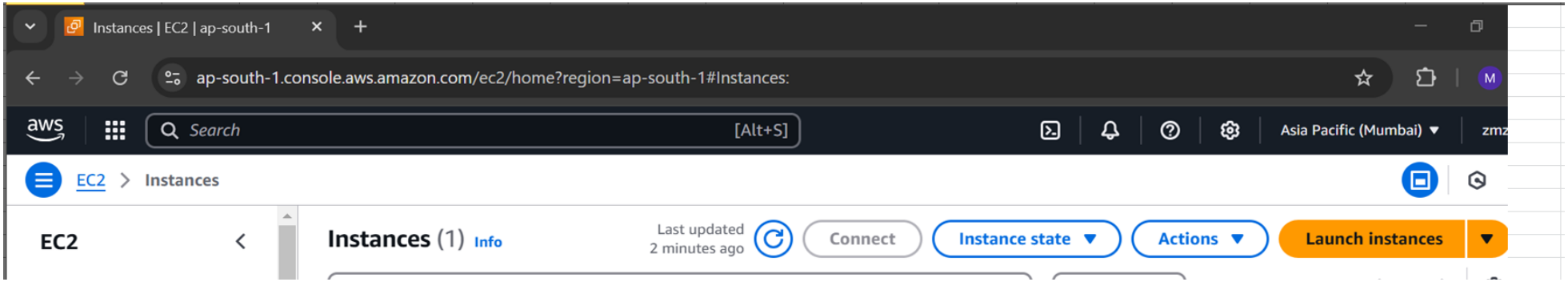




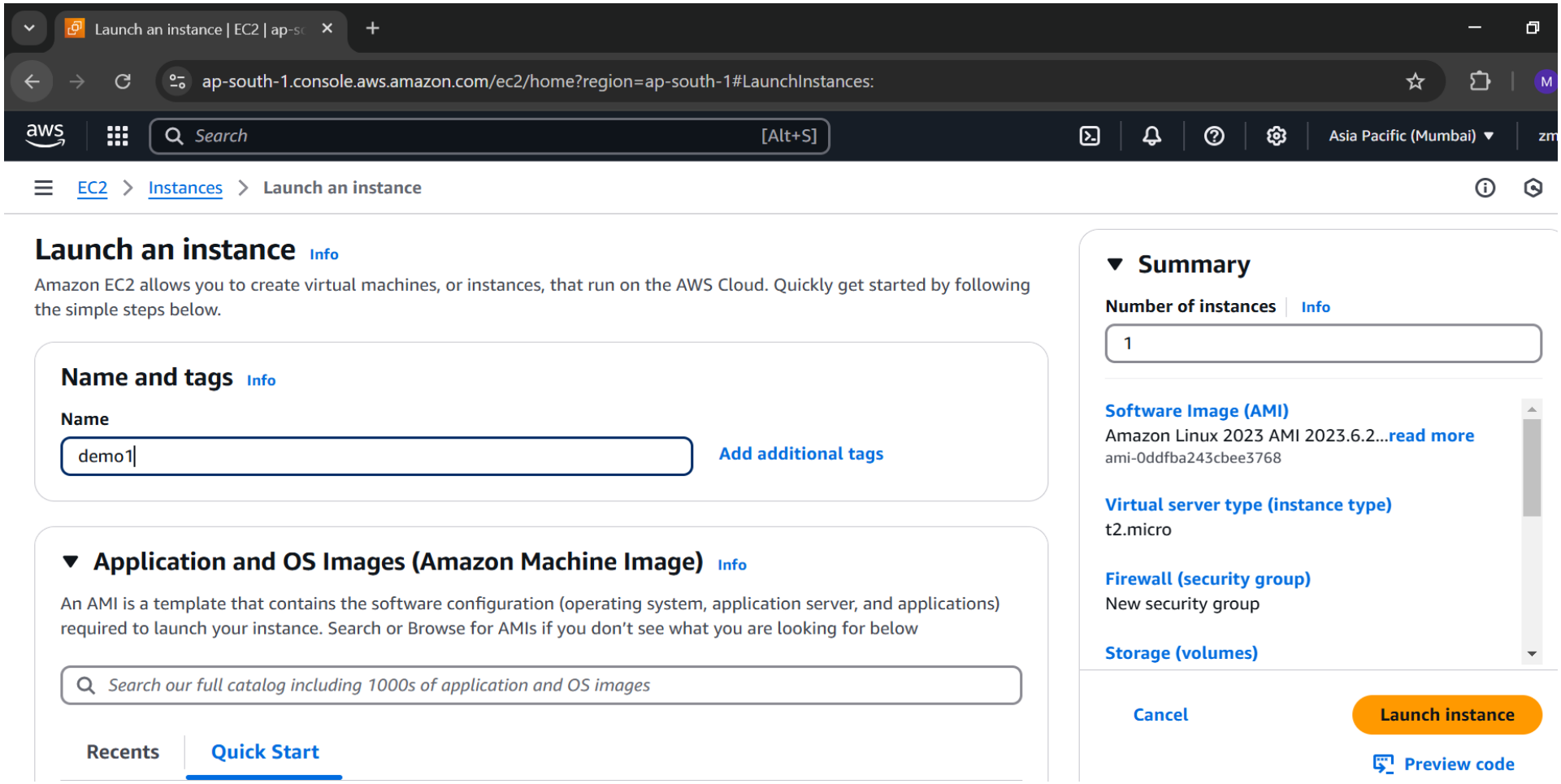
***) Login with your credentials**



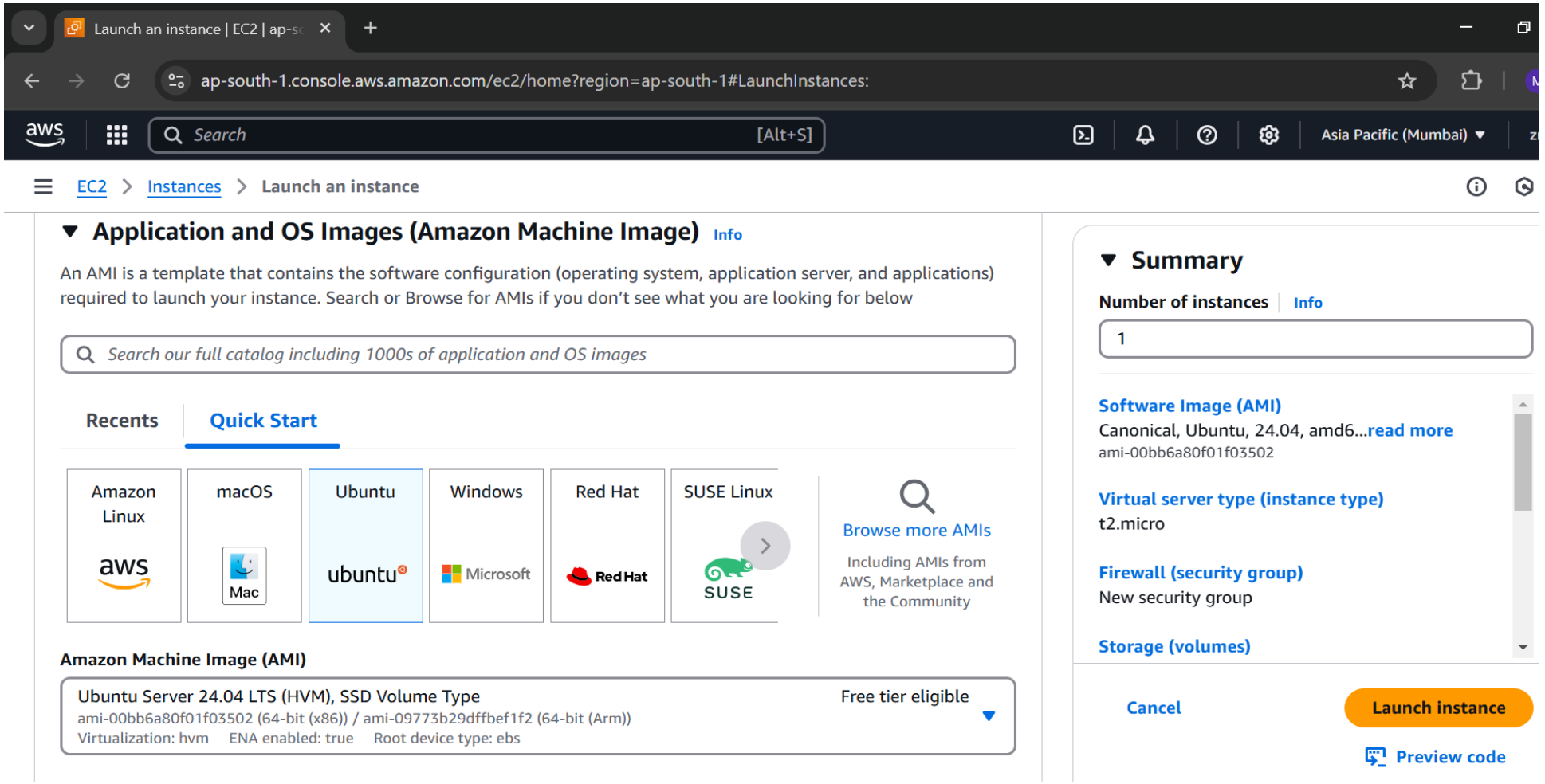
***) Type EC2 and select EC2 services**



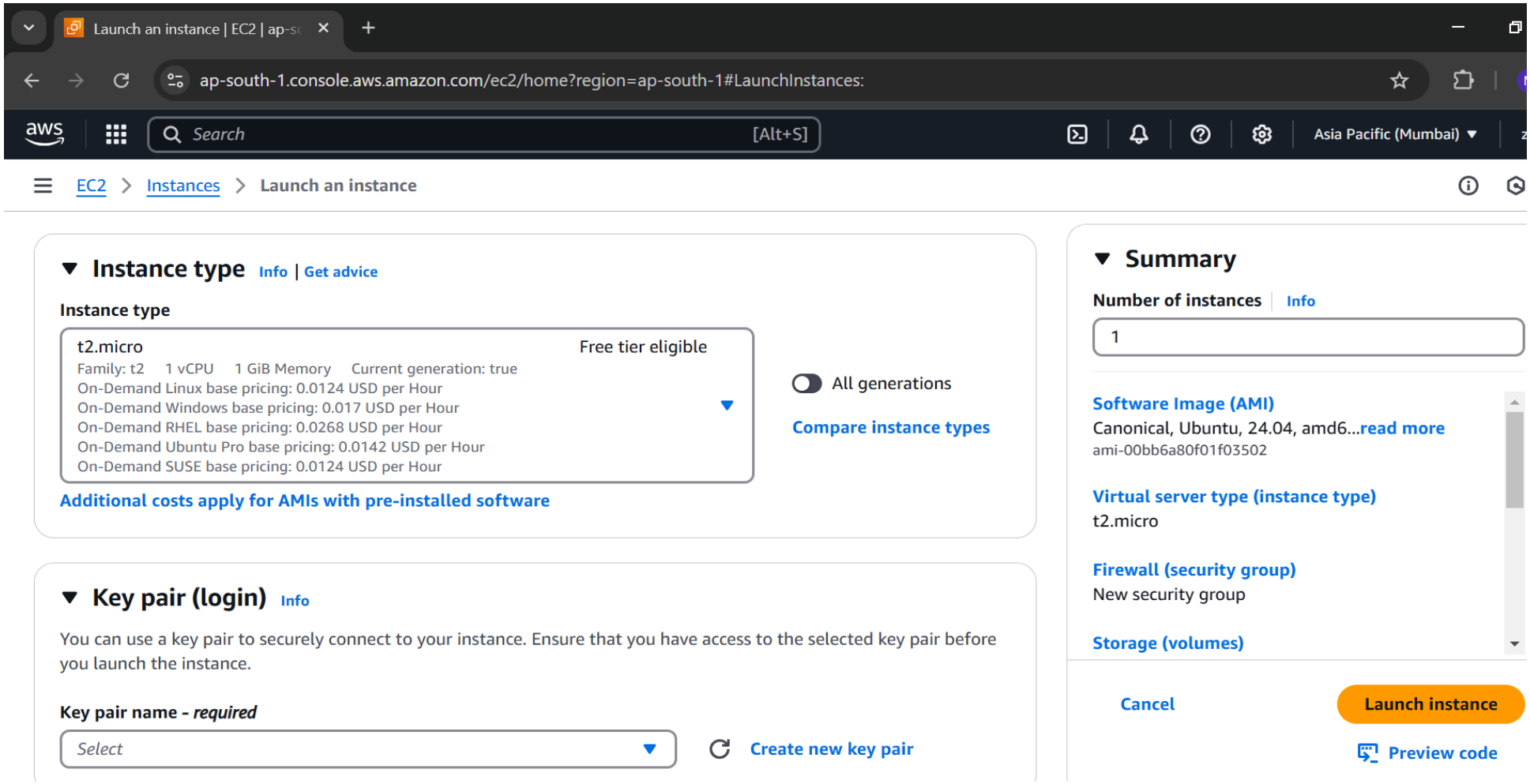
***) Click Launch Instances**



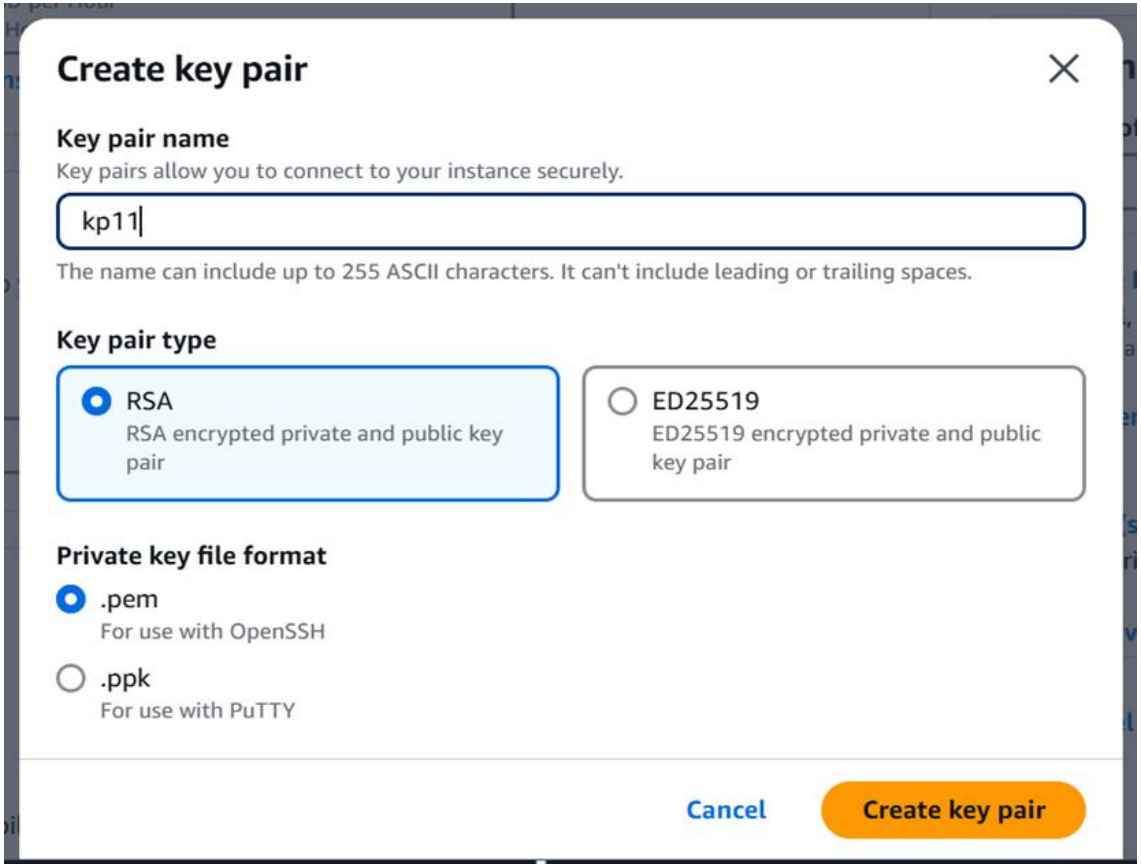
***) Give an instance name**



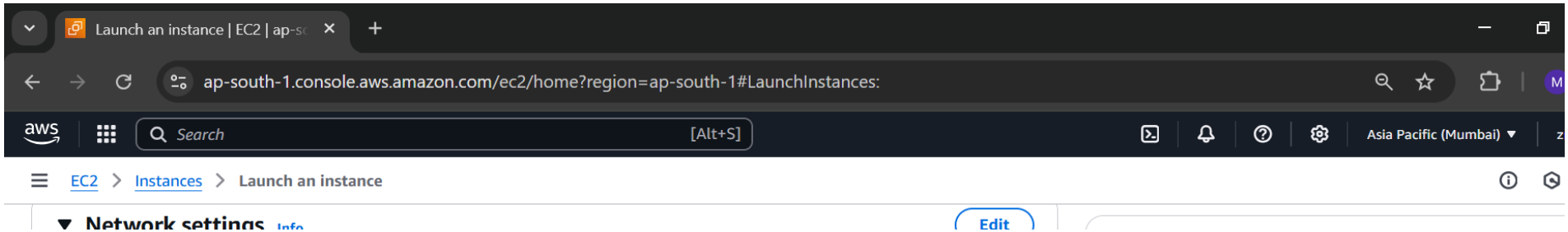
***) Select Ubuntu Server Image (Free tier)**

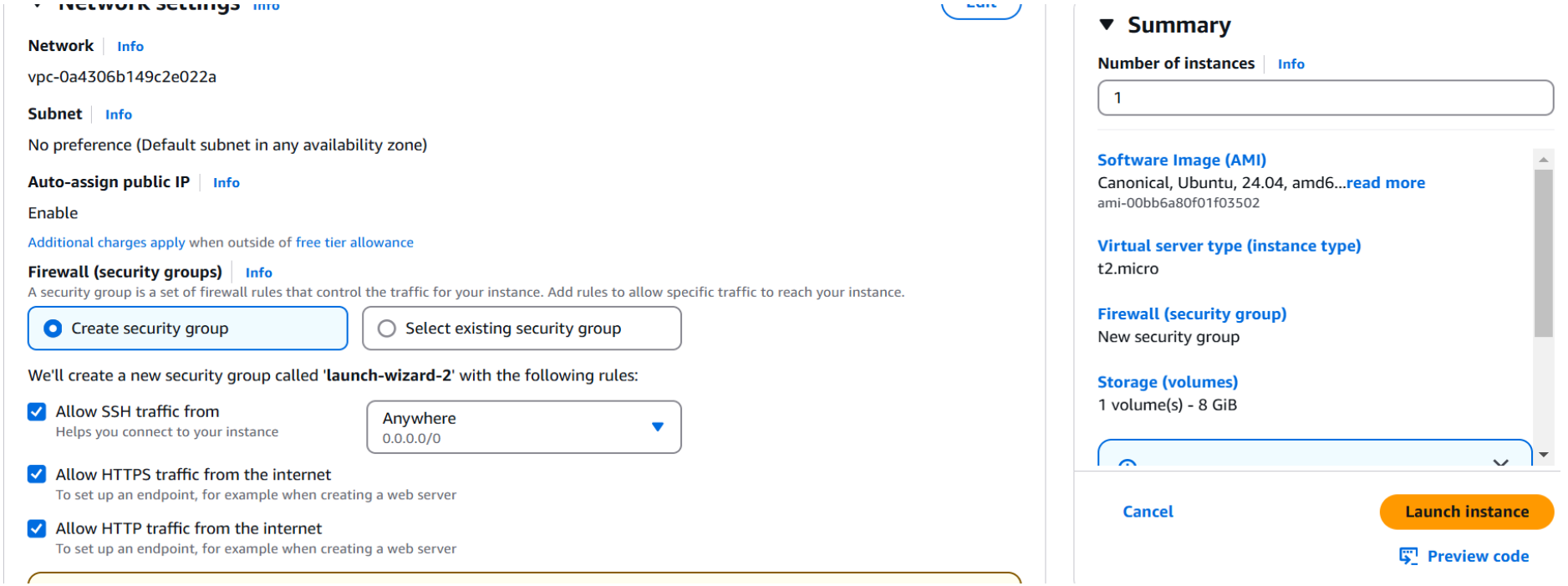


***) Select t2.micro instance type (free tier) and click Create new key pair(for any future server access)**

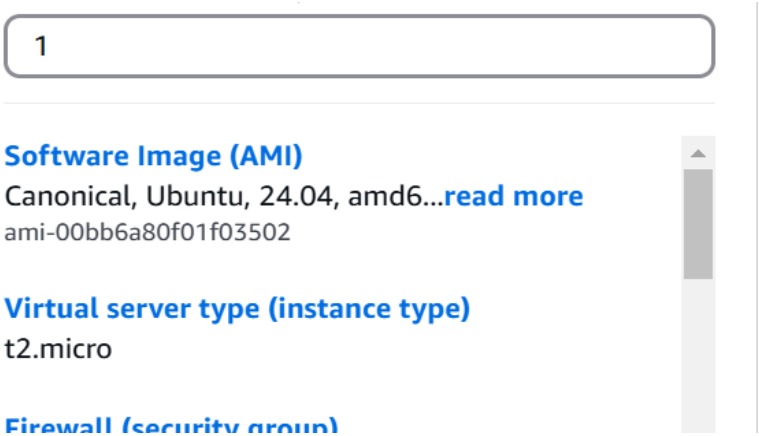


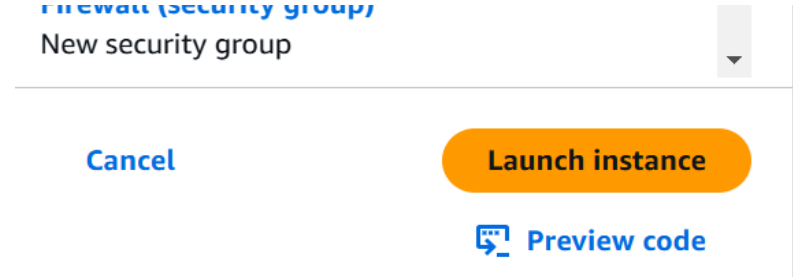
***) give a key pair name and keep rest as it is and click create key pair**

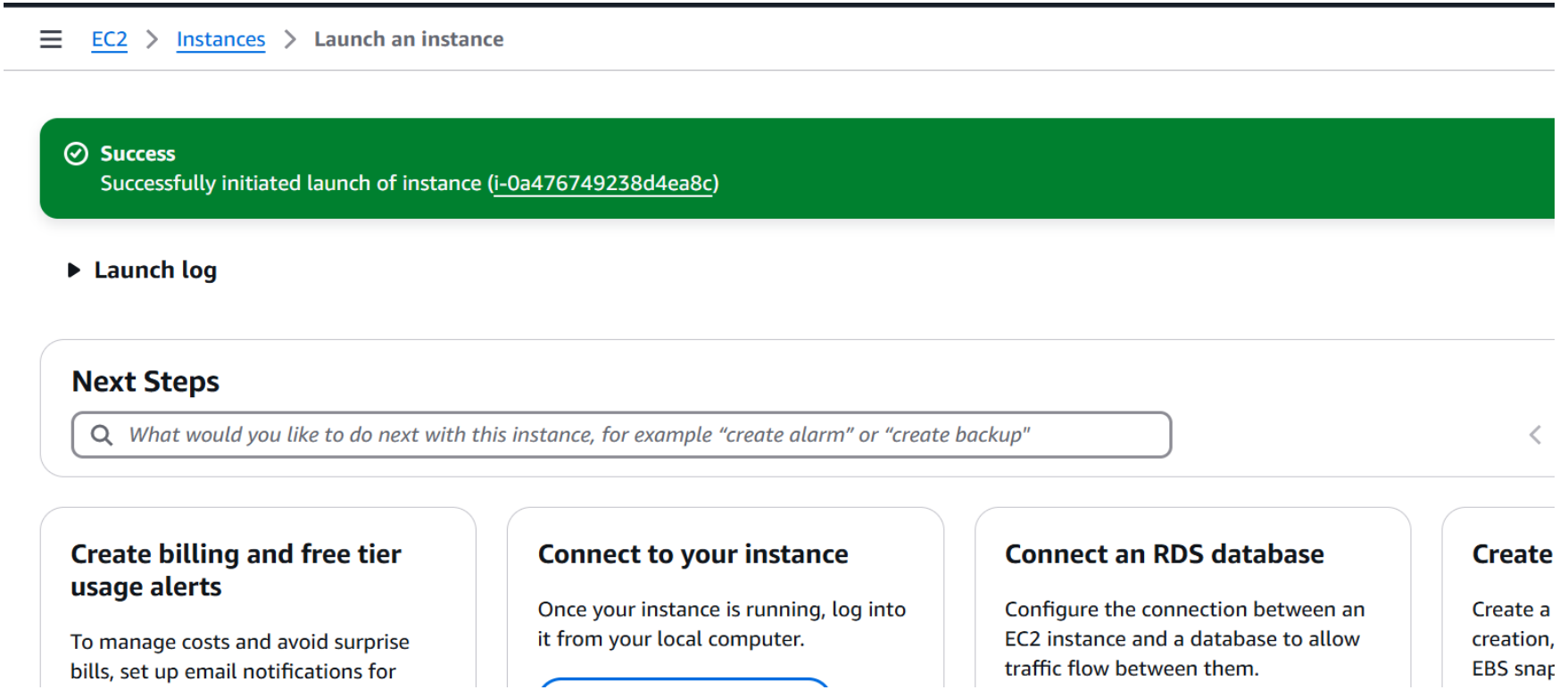




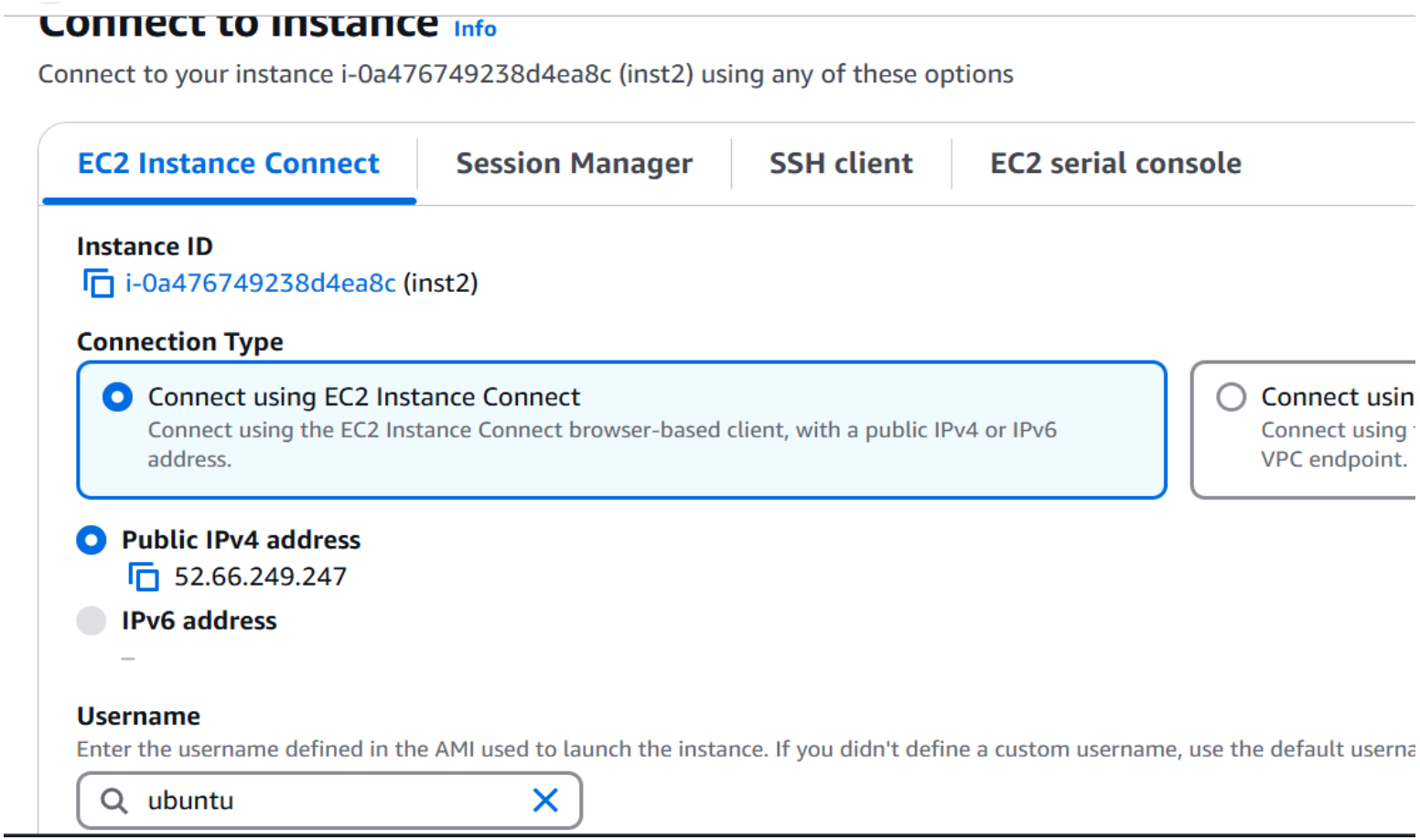
***) Click Launch Instance and your Server is ready**



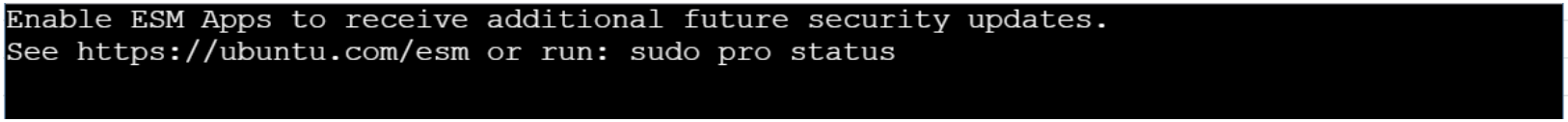

New security group
Cancel
Launch instance
Preview code

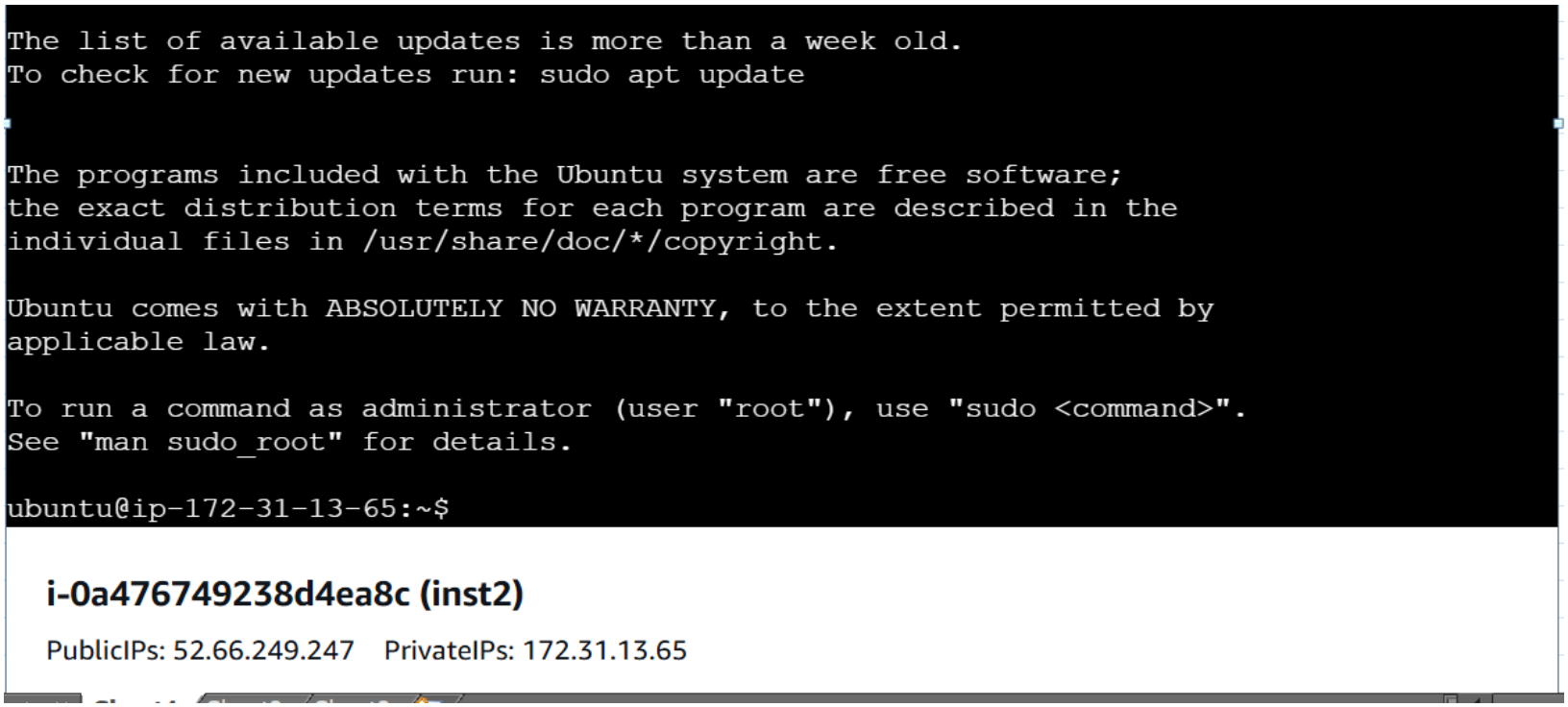

EC2 > Instances > Launch an instance
Success
Successfully initiated launch of instance (i-0a476749238d4ea8c)
Launch log
Next Steps
What would you like to do next with this instance, for example “create alarm” or “create backup"
Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for
Connect to your instance
Once your instance is running, log into it from your local computer.
Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.
Create
Create a
creation,
EBS snap

*) Click connect to your instance to log into server



*) choose Connect using EC2 instance connect as Connection Type





***) You will see a terminal connected to server**

***) Type the following commands to update/install the libraries in the server and activate Mlflow server**

 # to update the server, python and virtual environment library
sudo apt update
sudo apt install python3-pip
sudo apt install python3-virtualenv

 # to create the virtual environment
virtualenv env1

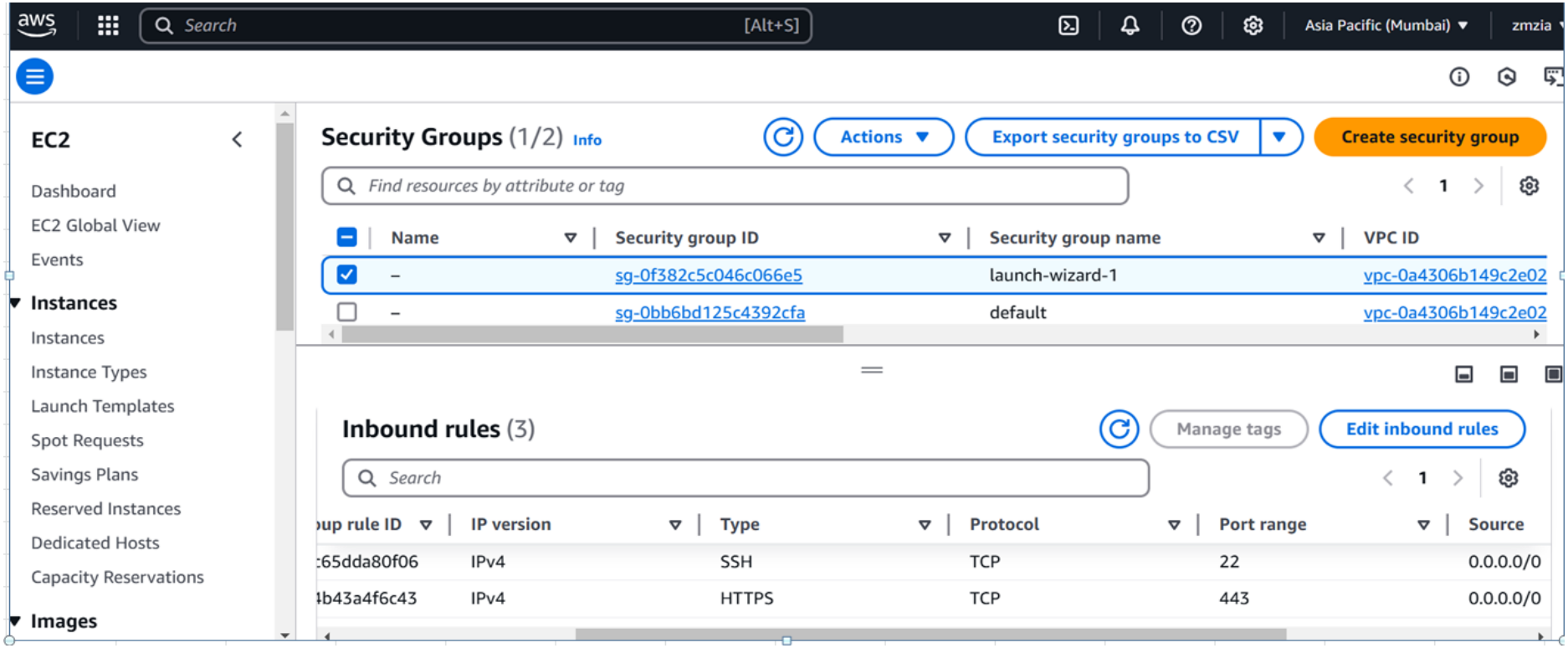
 # go to the bin folder in the newly created virtual environment

cd env1/bin

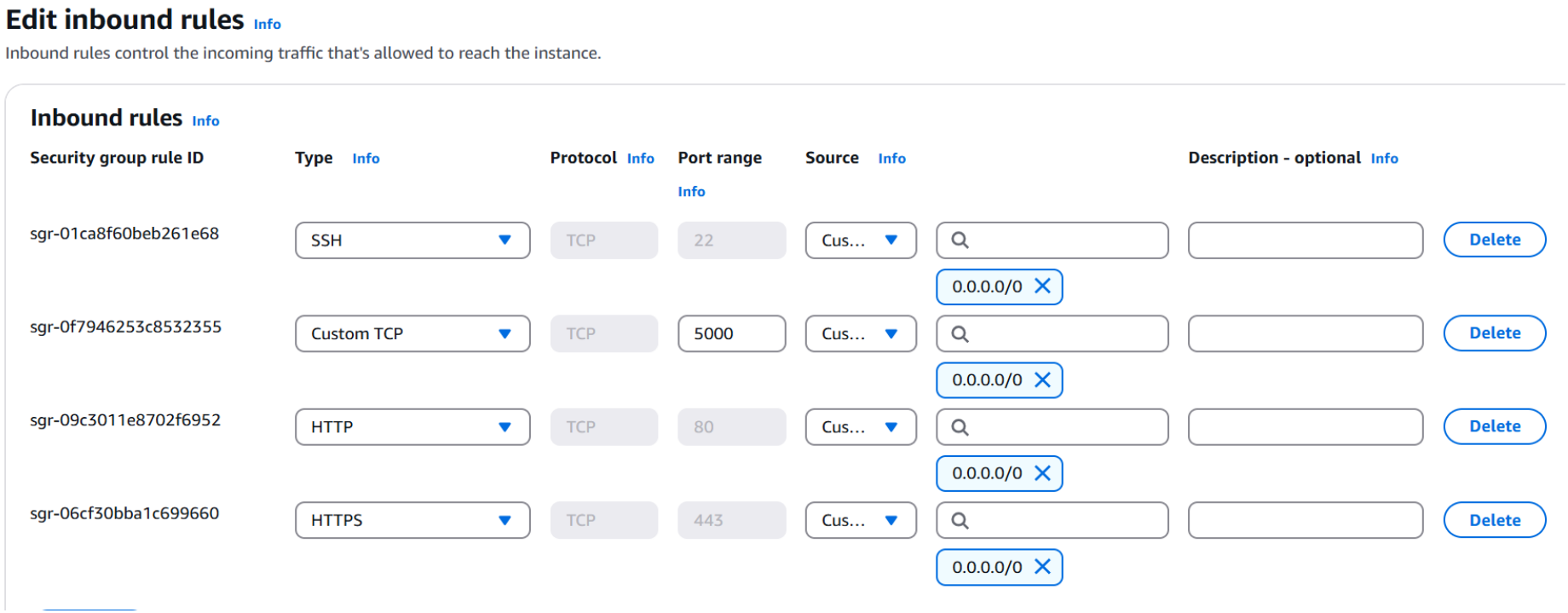
activate the virtual environment
source activate

install MLFlow library
pip install mlflow

*) With the instance created click Security Groups , choose create security group and check the three Allow boxes



*) Choose launch-wizard-1 group, under the Inbound rules tab, and click Edit Inbound rules



*) Add rule, Custom TCP(Type), 5000(port range) and 0.0.0.0/0(Source) and Save

*) go back to the EC2 instance window and start the MLFlow server

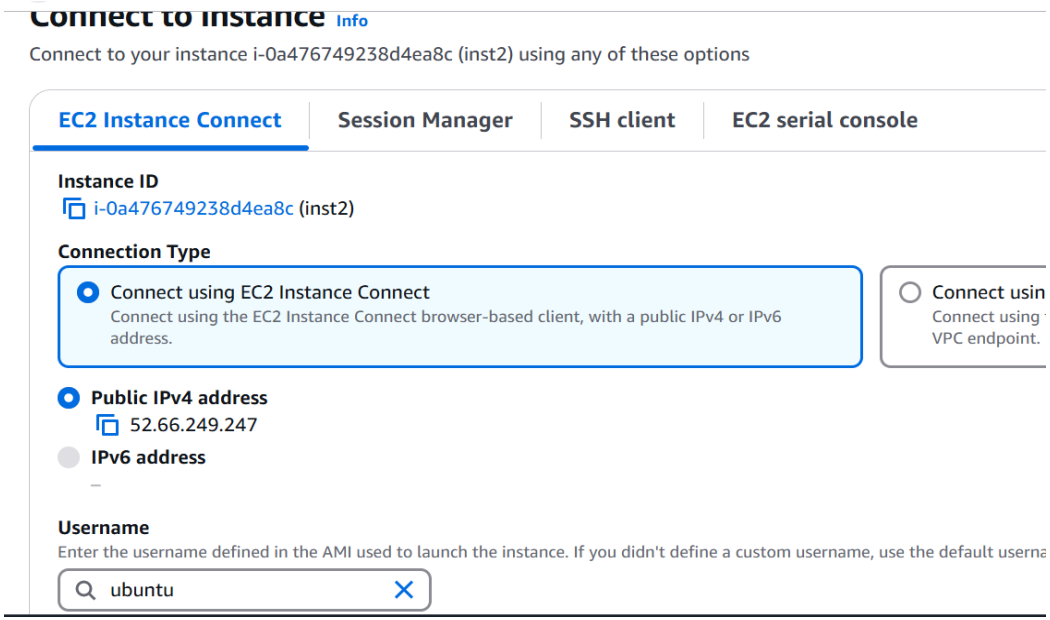
mlflow server -h 0.0.0.0 --port 5000

```
(env1) ubuntu@ip-172-31-13-65:~/env1/bin$ mlflow server -h 0.0.0.0 --port 5000
[2025-02-14 17:13:35 +0000] [5103] [INFO] Starting gunicorn 23.0.0
[2025-02-14 17:13:35 +0000] [5103] [INFO] Listening at: http://0.0.0.0:5000 (5103)
[2025-02-14 17:13:35 +0000] [5103] [INFO] Using worker: sync
[2025-02-14 17:13:35 +0000] [5104] [INFO] Booting worker with pid: 5104
```

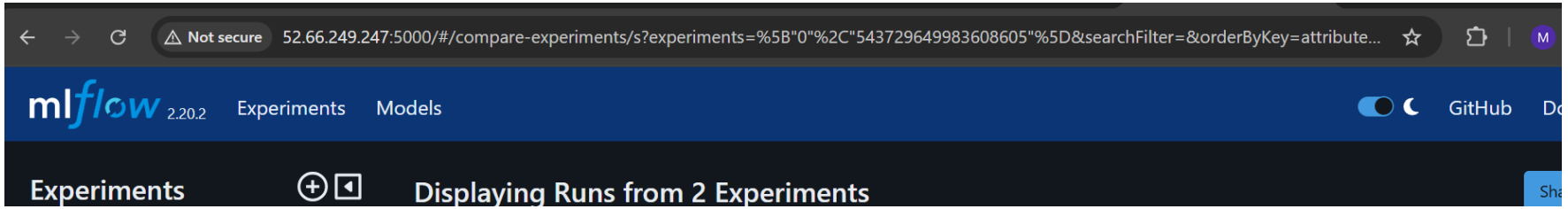
```
[2025-02-14 17:13:35 +0000] [5105] [INFO] Booting worker with pid: 5105
[2025-02-14 17:13:35 +0000] [5106] [INFO] Booting worker with pid: 5106
[2025-02-14 17:13:35 +0000] [5107] [INFO] Booting worker with pid: 5107
```

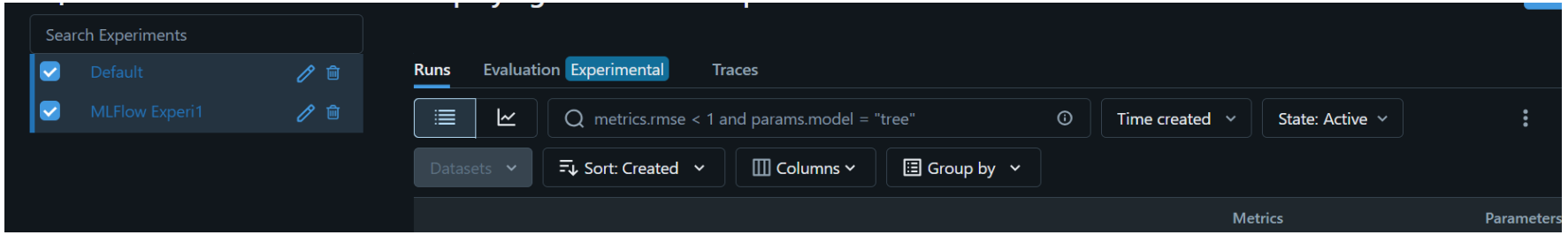
***) server is started and you can copy the public ip of the server and start using in your code for experiment tracking**

***) you can retrieve public ip address from various ways(below is one of them)**



***) access MLFlow Dashboard from browser like http://52.66.249.247:5000**





*) use this server for experiment tracking in the code (sample)

mlflow.set_tracking_uri("http://52.66.249.247:5000")

exp = mlflow.set_experiment("demo1")