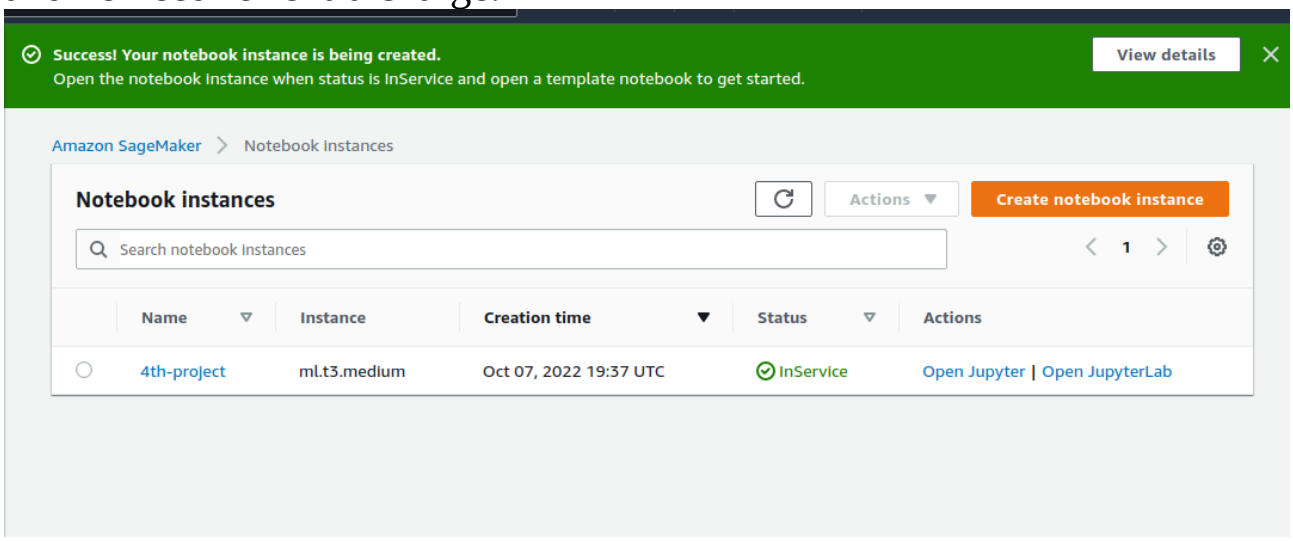
Step1:Training and Deployment

Initial step:

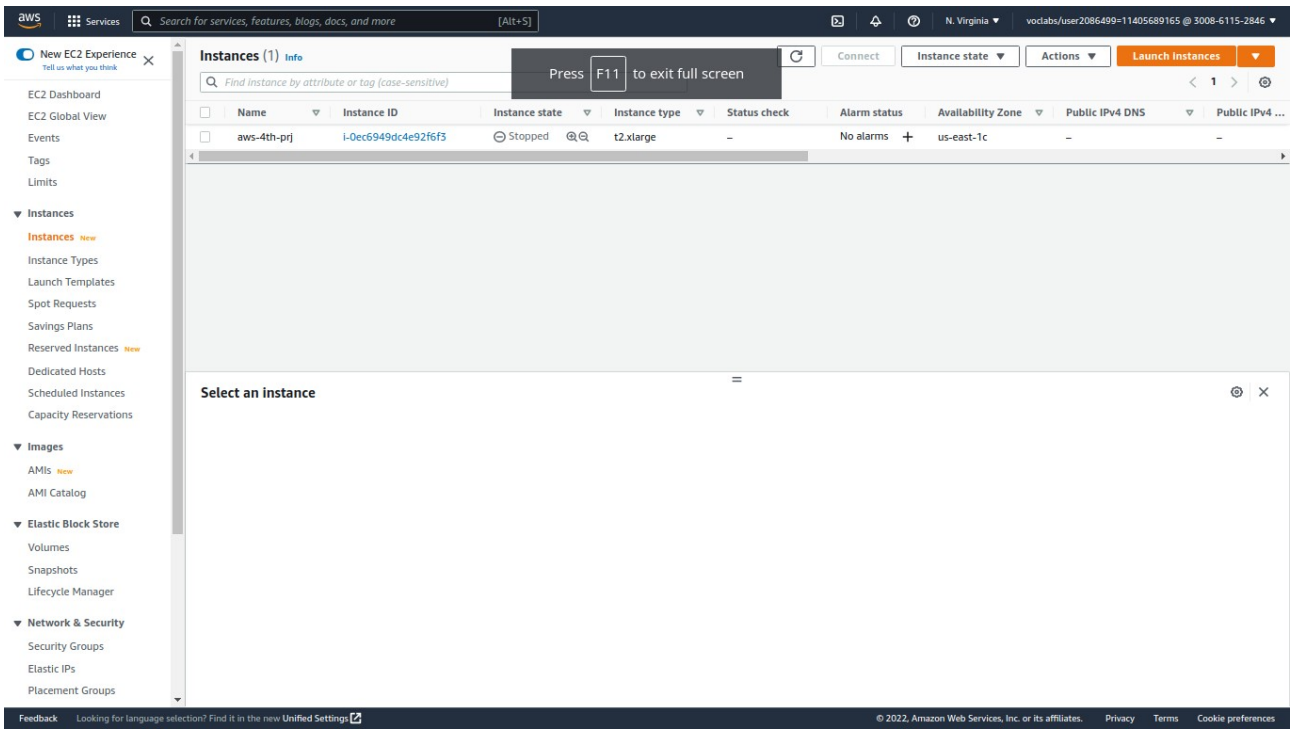
I used “ml.t3.medium” for my notebook instance as it was the default and no need for extra charge.



Step2:EC2 Setup

1.Type of EC2 instance:

I’ve chosen the “t2.xlarge” instance as it gave me the desired speed with affordable price.



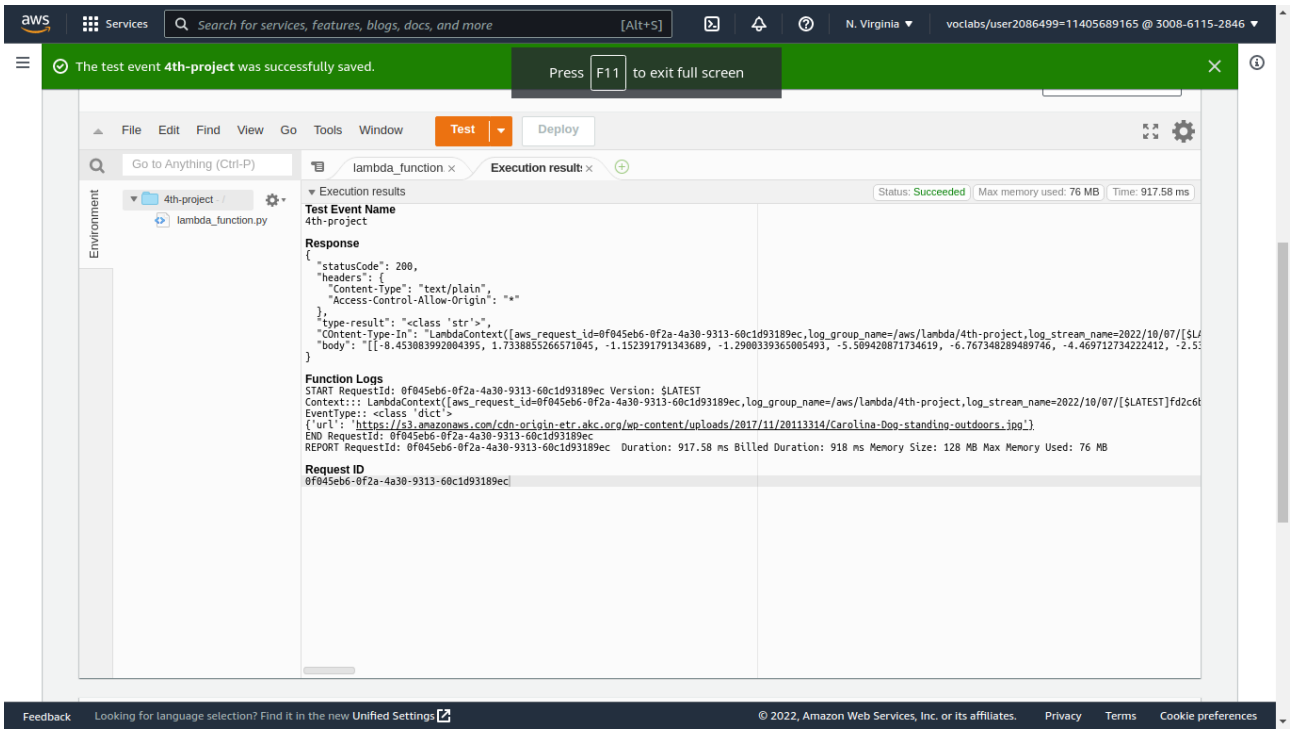
2. Comparison between Sagemaker code and EC2 code:

- In EC2 code we had to save the model data locally, while in Sagemaker model data is saved in S3 bucket, and it was done internally through the whole training process.
- I didn't mention any hyperparameters in EC2 as it's already mentioned in the script unlike Sagemaker.

Step3:Lambda Function

Write about lambda function:

- I changed the end point name for my endpoint deployed.
- I face a problem while testing the lambda function as it gave me "denied permission" and I had to set the IAM role to invoke the end point, also I had to attach "amazon SageMakerFullAccess" to add my permission.
- Then tested it and all worked fine.



Step4:Security And Testing

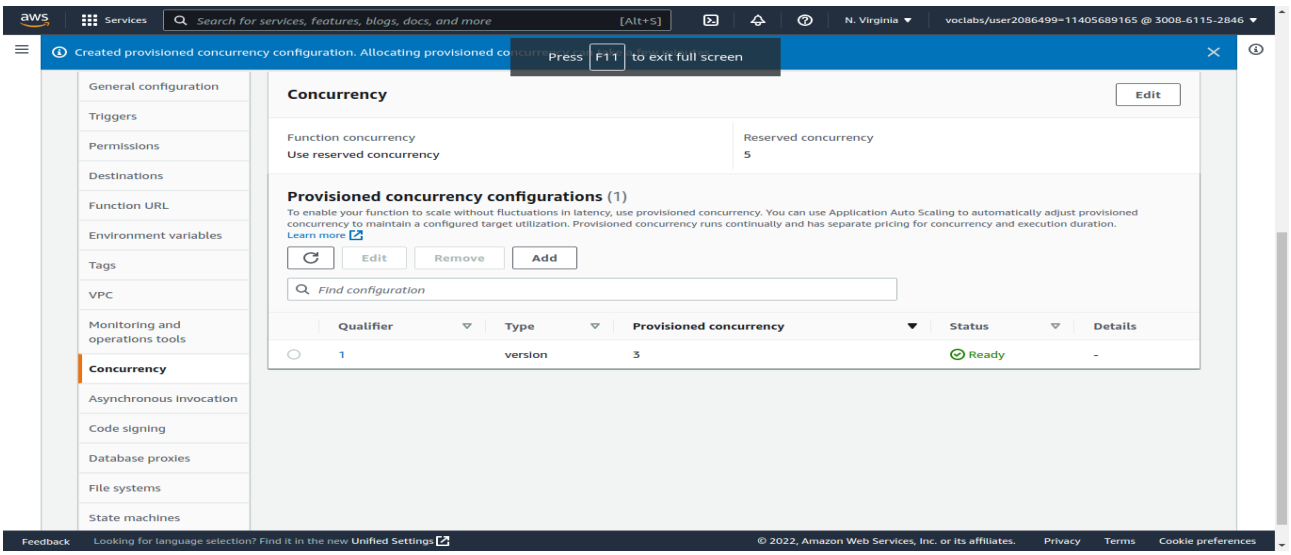
AWS workspace security:

I've checked my cloud watch and found that "logs" are familiar to me, also I have set the IAM role permission according to my "IP address" from the first place, so I made sure it was secured before proceesing.

Step5:Concurrency & Auto-scaling

Write about Concurrency & Auto-scaling configurations:

- First I have chosen the concurrency version for my lambda function
- Then set the reserved concurrency to “5”, with “3” provisioned concurrency.
- Tested Lambda again and “succeeded”



Regarding the Auto-scaling:

- I have set maximum instances “3”
- Also set “Scale-in” and “Scale-out” into “30” seconds which sounds reasonable enough for conigurations.