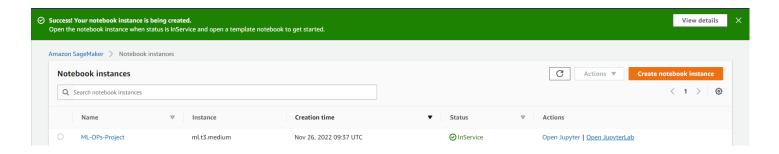
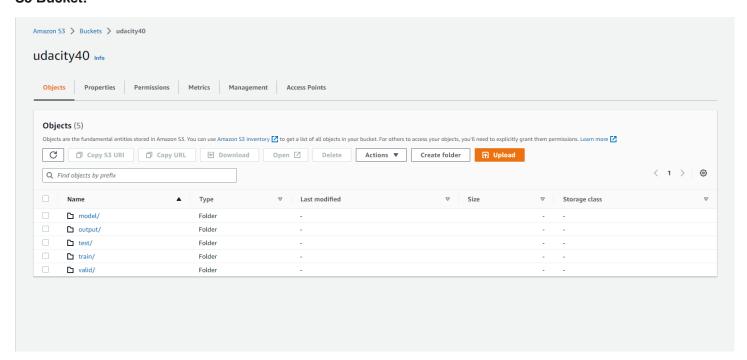
Operationalizing an AWS ML Project

SageMaker instance:

I have chosen an ml.t3.medium instance because it has sufficient computing power (vCPU =2, Memory = 4 GiB) and it's cheap (Price per Hour = \$0.05). So, it does the job while maintaining a low cost.

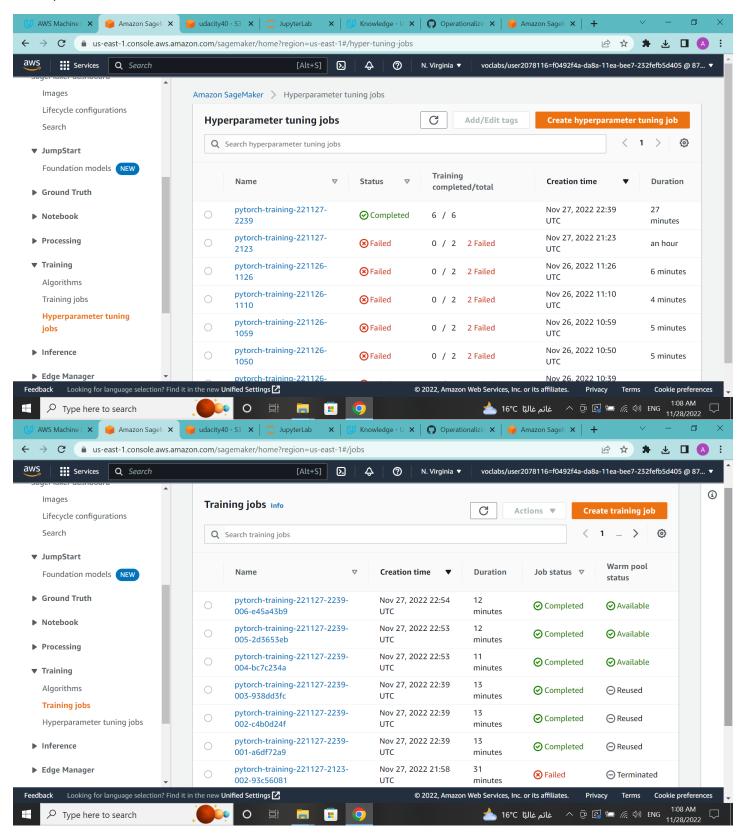


S3 Bucket:

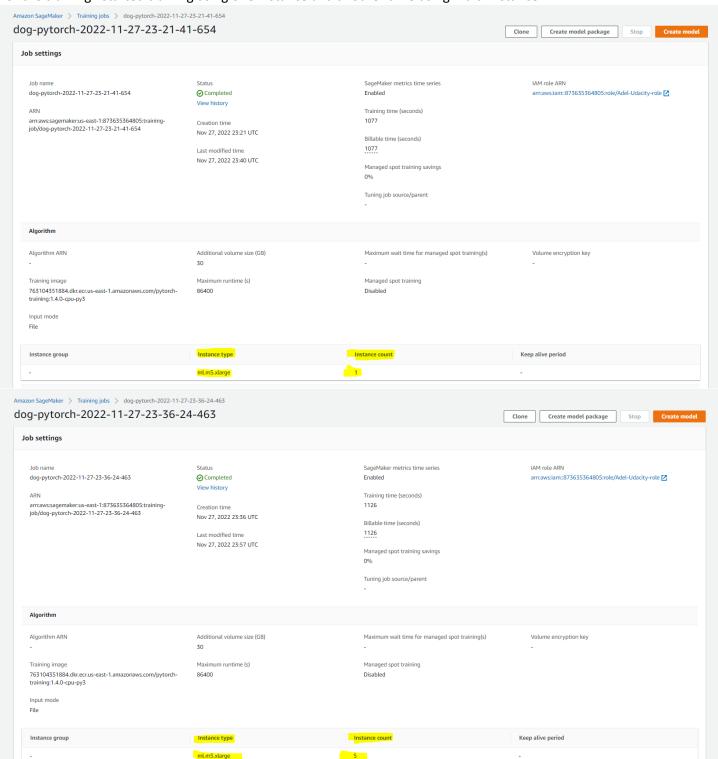


Training and Deployment:

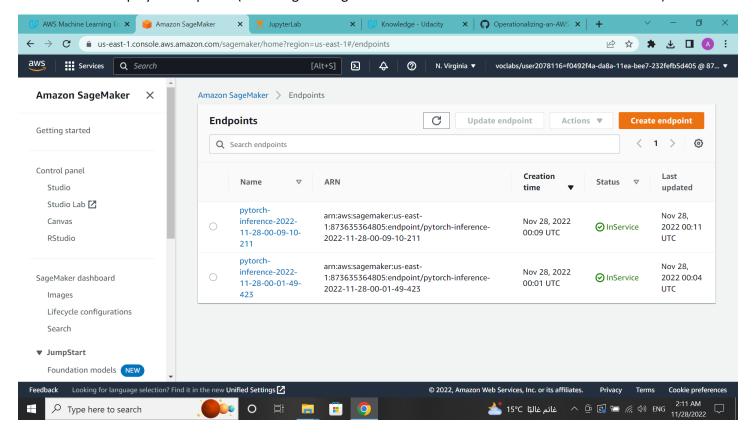
I have had several problems when training the model using the provided instance, first with the error "limit exceeded" and another because some problems in hpo.py. But I managed to overcome these problems by using "ml.m5.2xlarge" instance and max_jobs=6, max_parallel_jobs=3. The hyper-parameter and training jobs are show in the below pictures.



For the training I started training using one instance and another time using multi-instance.

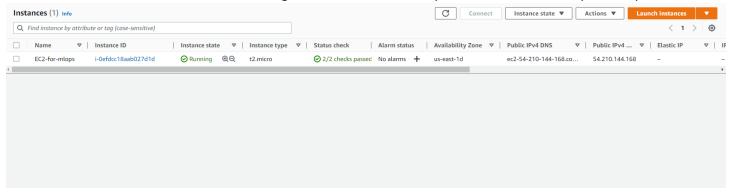


And here is the deployed end points (one using training with one instance and the other with multi-instance)



EC2:

I have used a t2.micro instance because it's eligible for the free tier and provide the needed computation power.



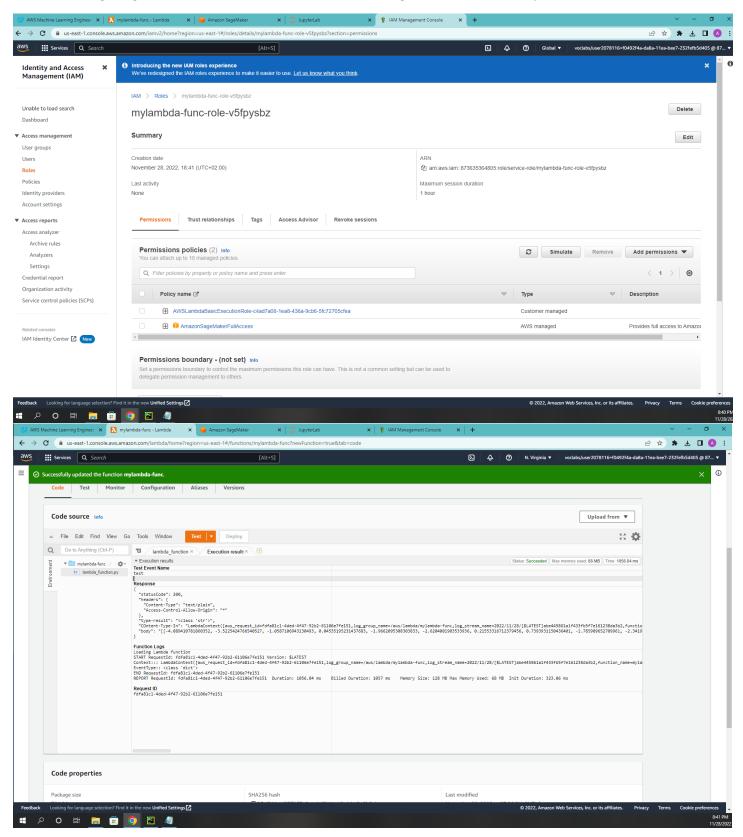
Difference between EC2 training code and SageMaker's:

- In EC2 code there is no calling for any Estimator or Tuner functions. The code in the EC2 script is responsible for saving the model to the local path. While in the SageMaker scripts this was handled internally by SageMaker where the model data was stored to a S3 location.
- In the EC2 code, all the variables already mentioned in the code itself.
- In the EC2 the training happens on the same while in the SageMaker the training job runs on a separate container than the one on which the SageMaker notebook is running.

Lambda function:

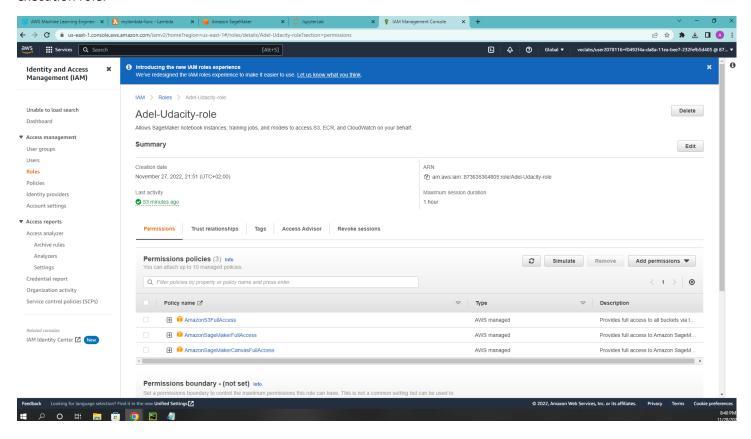
Lambda functions are used for invoking our deployed endpoints. And the end point used is the one of the mult-instance training (pytorch-inference-2022-11-28-18-38-11-439). Also we attached an Amazon SageMaker full Accesses policy to be able to interact with SageMaker successfully with no errors.

In the following images there are the role and result of the testing (also included in a separate file called lambda.txt)



Security:

It's not a good idea to give full access permission because it may result in a security preach, but always choose the right policies and permission and remove them when the task is done. Below is the policies attached to my SageMaker execution role.



Concurrency and Auto Scaling:

I have configured a provisioned concurrency after publishing a version for lambda. Also I have configured Auto Scaling to cope with the traffic requests.

