Operationalizing-an-AWS-ML

1-Initial setup, training and deployment

I choose ml.t3.medium since it is the lowest cost and my project is on budget. Also it was enough for me to use it. But inside the instance I used larger 2xlarge instances for the training.

2-EC2 Setup

I used ml.m5.2xlarge instance for training and used also Deep Learning AMI GPU Pytorch but I had to install numpy and pytorch packages again because it was not working without them.

3-Comparing EC2 to step 1

EC2 instances main competence power is that it is cheap and can help you in checking models without paying too much from your budget. In EC2 it trains locally and doesn't work with multiple services like s3 and endpoint instances. On the other hand sagemaker notebook instance is more expensive but can interact with s3 and deploy end points. Also sagemaker has more user friendly interface.

4-Setting up a Lambda function

Firstly I created the lumbda function and replaced the code with my "endpoint name" then I added the Full access to the lambda function and after that I used the test case to check it. It worked and gave me the list of 33 numbers

5-Concurrency and Auto-scaling

in concurrency firstly I made a new version and then I used 5 reserved concurrency . Then I made 3 provision concurrency. I made this amount to use more than one concurrency and at the same time not pay a lot.

In auto scaling I used instances from 1-3 because it is a small project and doesn't require a lot of instances. Also the cool downtime is 300 sec because it is not nesesarly to be so responsive in this project to 300 sec is fine.