**Operationalise AWS Machine Learning Project**

1. **Sagemaker Instance justification:** I ml.t2.medium instance has 2vCPUs and 4 GB memory. This standard instance was sufficient for my computing needs.
2. **EC2 instance justification:** The t2.micro instance type had 1 cpu and 1GB of memory. T2 instances are Amazon EC2 instance types designed to dramatically reduce costs for applications that benefit from the ability to burst to full core performance whenever required
3. **EC2 code comparison:** The main difference between notebook and EC2 code are the modules that are imported in the notebook code. The Sagemaker notebook imports modules for analytics, hyperparameter tuning and debugging while the EC2 code doesn’t
4. **Lambda function description:** 
   1. Firstly, the name of the deployed endpoint is defined
   2. Next we get the client using boto3 session
   3. The endpoint is invoked using boto3 client by the endpoint name with the image URL received from the request
   4. Process the received result from the endpoint
   5. Return the processed result
5. **Lambda function security:** I attached the SageMakeFullAccess policy to the lambda function’s role. This role may be quite permissive however I was unable to find a more restrictive role that allowed the lambda function to only interact with the endpoint
6. **Concurrency configuration:** I used reserved concurrency and set up 4 maximum concurrent instances for the lambda function. I use ‘Reserved concurrency’ as opposed to “provisioned concurrency” because there’s no charge for configuring reserved concurrency for a function.
7. **Autoscaling configuration:** 
   1. Maximum instance count= 3 and minimum instance count = 1. This was set so that there was some compacity for the model scale in case of high traffic
   2. Scale in and out cool downs were both set to 30 seconds with a target of 30 to ensure that the model was very responsive to scaling compared to the default of 300 seconds for scale in and out.