Project 1 Report

Shriya Srinivasan
ASURITE ID: ssrin103

TASKS DONE:

- Initial designing architecture stage
- App Tier Python script
- User data
- Step scaling policy and Cloud Watch Alarms
- AWS Architecture diagram

IMPLEMENTATION OF DIFFERENT TASKS:

- Initial designing architecture stage: The team brainstormed and came up with a cloud architecture design for our Cloud-based Image classification application.
- App tier(python script):
    - Read AWS and Boto3 documentation and experimented with AWS console and CLI. Created simple resources and explored their capabilities.
    - Got familiar with AWS Free Tier limits.
    - Configured AWS credentials using the AWS CLI. Generated an access key and secret access key and then used the ‘aws configure’ command to complete the configuration.
    - Wrote app tier code: started a continuous loop to poll for messages in the request queue, upload image to input s3 bucket and save to an input image directory, run the Deep learning image classification model, put the result into the response queue and into output s3 bucket.
    - Tested the code on my local system and remote EC2 instance.

- User Data: The team modified and tested the user data to make sure the app tier code runs smoothly on scaled-out instances. We encountered issues around python module installations and permissions.

- Step-scaling policy and CloudWatch alarms: The team created a Step scaling policy for our autoscaling group and set up alarms in CloudWatch.
  For this, I helped perform the following tasks:
    - Read the documentation and experimented with the AWS Autoscaling group, different dynamic scaling policies, and CloudWatch metrics and alarms.
    - Modified the policy into a step scaling policy with different scaling values(scaling in and out) for different ranges of the metric value = (Number of messages in the request queue - Number of running instances).

- AWS Architecture diagram: Used draw.io to create an AWS Architecture diagram to accurately depict the working of our project.