CSE 546: Cloud Computing

Individual Contribution MSCS portfolio report.

Abhishek Annabathula
ASUID:1225491722
Project I: IaaS

This project required us to implement an AWS-hosted web application that takes concurrent image requests and performs deep learning image classification on the images and provides the recognition result of the content in the image. To achieve the goal, the application architecture consists of a web tier, an app tier, and auto-scaling using cloud watch. As per the architecture, the web tier accepts the concurrent images and pushes them to the AWS SQS request queue, and checks for the recognition results in the SQS response queue.

My part in this project is to build the App tier of the application which handles the reading of messages from the request queue and runs image classification on the image and pushes the result to the response queue. Once the image is read from the request queue, it is saved into the request S3 bucket. When the image classification is run on the images, then the images are fetched from the S3 bucket. After the classification is done, the result is stored in the response S3 bucket.

The implementation details go as follows. The initial setup of the App tier consists of four files namely Apptier Properties.py, AWSUtils.py, ImageClassifier.py, and AppTier.py.

AppTierProperties.py:

This file consists of the request and response queue names, S3 request and response bucket names, and AWS credentials.

AWSUtils.py:

This file consists of the implementation of utility functions like send_message_to_response_queue , receive_message_from_request_queue,delete_message_from_sqs, and download_from_request_s3.

ImageClassifier.py:

This file consists of functionality to read messages from the request queue, perform image classification using image_classification.py, and push the results to the response queue along with uploading images to the S3 bucket and downloading messages from the S3 bucket. The message received is a dictionary with messageID, message-body, and receiptHandle. The message body in base64 format is converted to an image in jpg. After the image classification is done, a response dictionary with the image name and recognition result is created {image_name: recognition_result}. This message is pushed into the response SQS queue and the message in the request queue is deleted after persisting the request message in the request S3 bucket and the response message in the Response S3 bucket. This will persist the request message in the request S3 bucket and the response message in the response S3 bucket. An exit graceful function is implemented to handle the os signal.