CSE546 Project 1 Individual Report

Revanth Bangalore Suresha

IMPLEMENTATION OF DIFFERENT TASKS:

1. **Initial designing architecture stage:** Collectively worked on and came up with the initial architecture of the Cloud Based Image Classification Service.

2. **Setting up all AWS Resources:** Set up all the AWS Resources listed:
    a. IAM: users, groups, roles, policies.
    b. S3: Request bucket, Response bucket.
    c. SQS: Request queue, Response queue.
    d. AMI: Custom AMI based on the AMI shared to the class containing the environment to image classification algorithm. We used the given AMI as base and created a custom AMI containing the python script that would poll messages from the request queue, download images from S3, run the image classification algorithm on the image and push the results to the response queue and S3.
    e. ASG (Auto Scaling Group) and setting up launch configuration with userdata to run the above mentioned python script on startup, setting a few environment variables, setting permissions to certain folders / files.
    f. SNS: Setup notifications, topics and email subscriptions for when alarms are in on state for scale in and scale out policies setup for auto scaling group.

3. **User Data for Launch Configuration:** Collectively worked on and tested UserData script for the ASG Launch Configuration that would be used to produce custom EC2 instances.

4. **Scaling policies and CloudWatch alarms:**
    a. Created a Target tracking scaling policy for the Autoscaling group with Custom Metric Specification. The custom metric published to CloudWatch was BackLogPerInstance = ApproximateNumberOfMessagesVisible (in request queue) / GroupInServiceInstances.
    b. Since the target tracking scaling policy was based on a custom metric which AWS doesn't support on the console at the moment, we created it using AWS CLI through 'aws cloudwatch put-scaling-policy'.
    c. The Custom Target Tracking Scaling Policy automatically created CloudWatch alarms to scale out and scale in to maintain a value of 1.0 for the above mentioned custom metric
    d. Step-scaling policy and CloudWatch alarms: Although Target Tracking Policy satisfied most of our needs, it had a few drawbacks.
        i. The alarms to scale in and out were created automatically and there was no provision to edit the properties like EvaluationPeriod, Threshold, etc.
        ii. The auto scaling group could not have a minimum number of instances set to 0 as that would make the custom metric undefined and the ASG did not scale out.

5. **Group Report:** Contributed in writing the corresponding sections in the group report and review.