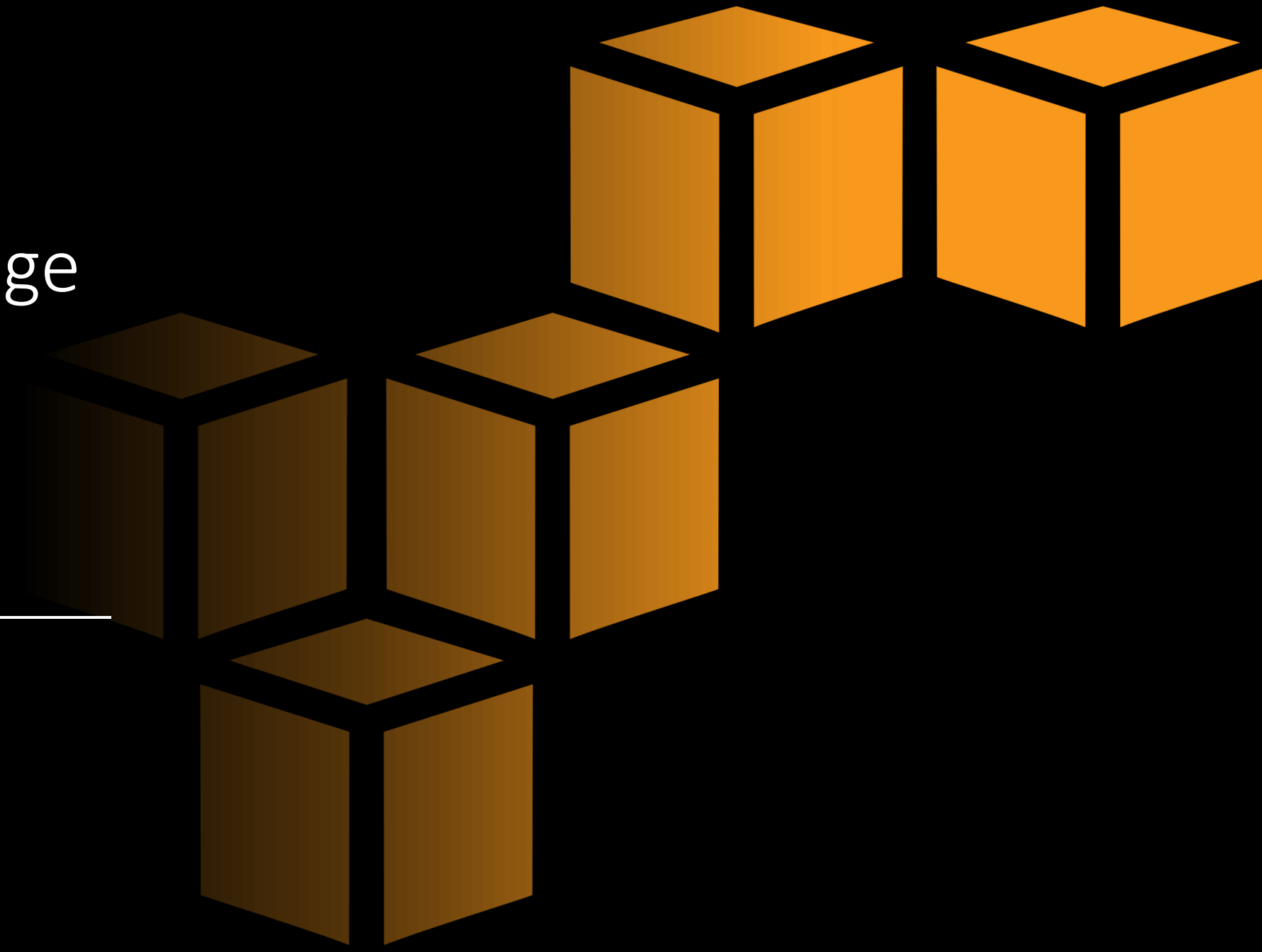
Serverless Image Recognition Application on AWS



Agenda

- Introduction
- Executive Summary
- Architecture
- Security Policy
- Design Considerations

- *Team # 3*
- *<u>Team Members :</u>*
- *1) Rahul Sinha*
- *2) Soumitra Mishra*

Introduction

- Problem Statement:
  - Automate and innovate the inspection process of a widget manufacturing company that currently take pictures of manufactured widget and inspect it manually to ensure the compliance standards set by QA group.

- Proposed Solution:
  - Create an AWS serverless architecture to fully automate the manual inspection process.
  - Create an event-based architecture and make use of Amazon Rekognition to inspect and generated reports for the widget images stored in the cloud

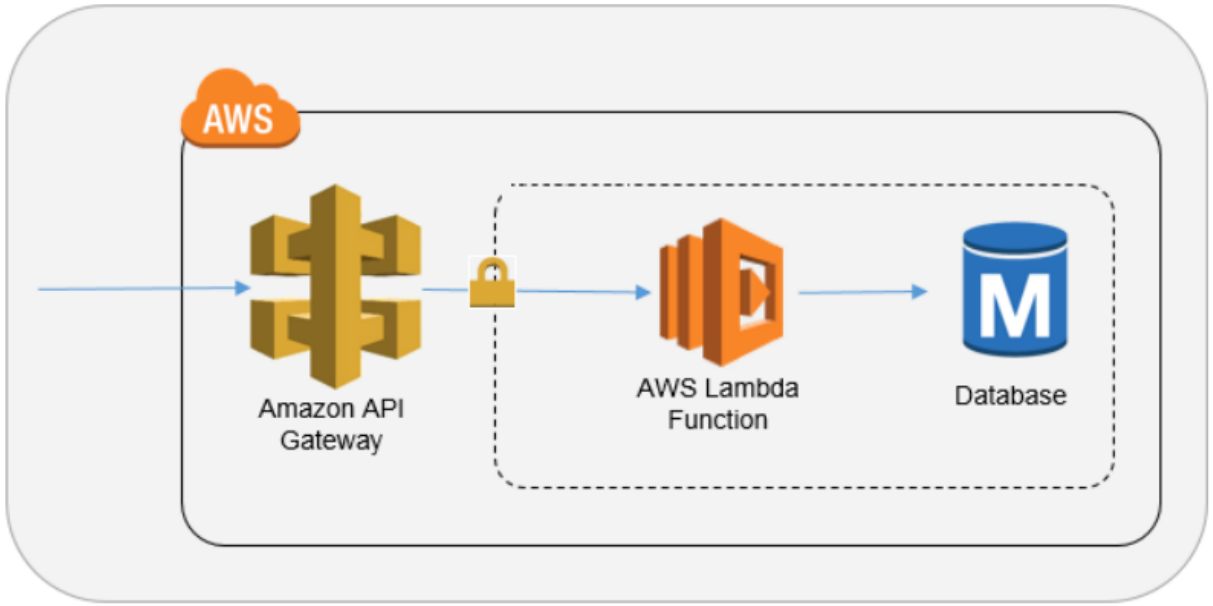
Architecture -Design Summary

3 tier serverless Architecture

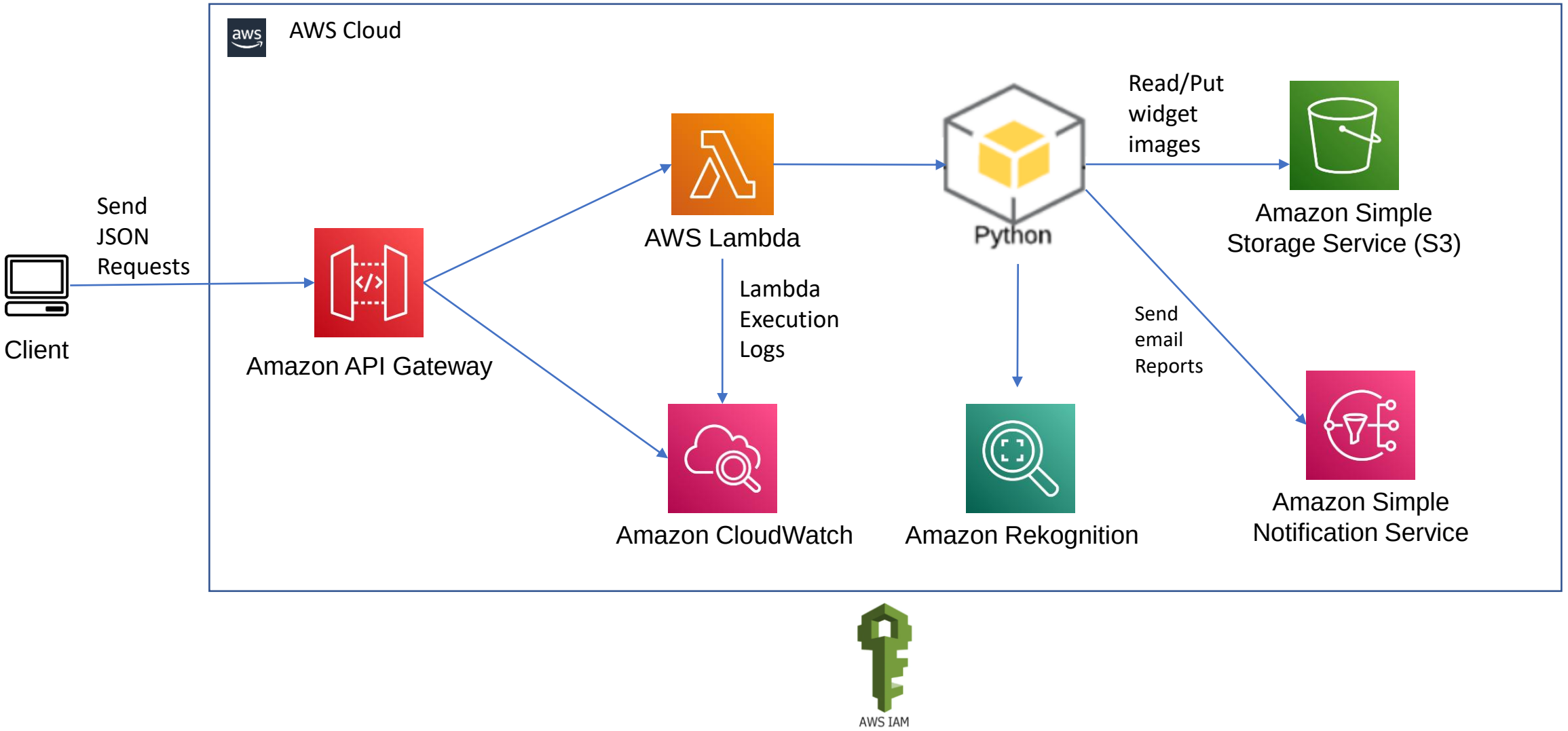
API Gateway to accept the API calls to upload images from the application.

AWS Lambda as the business logic tier

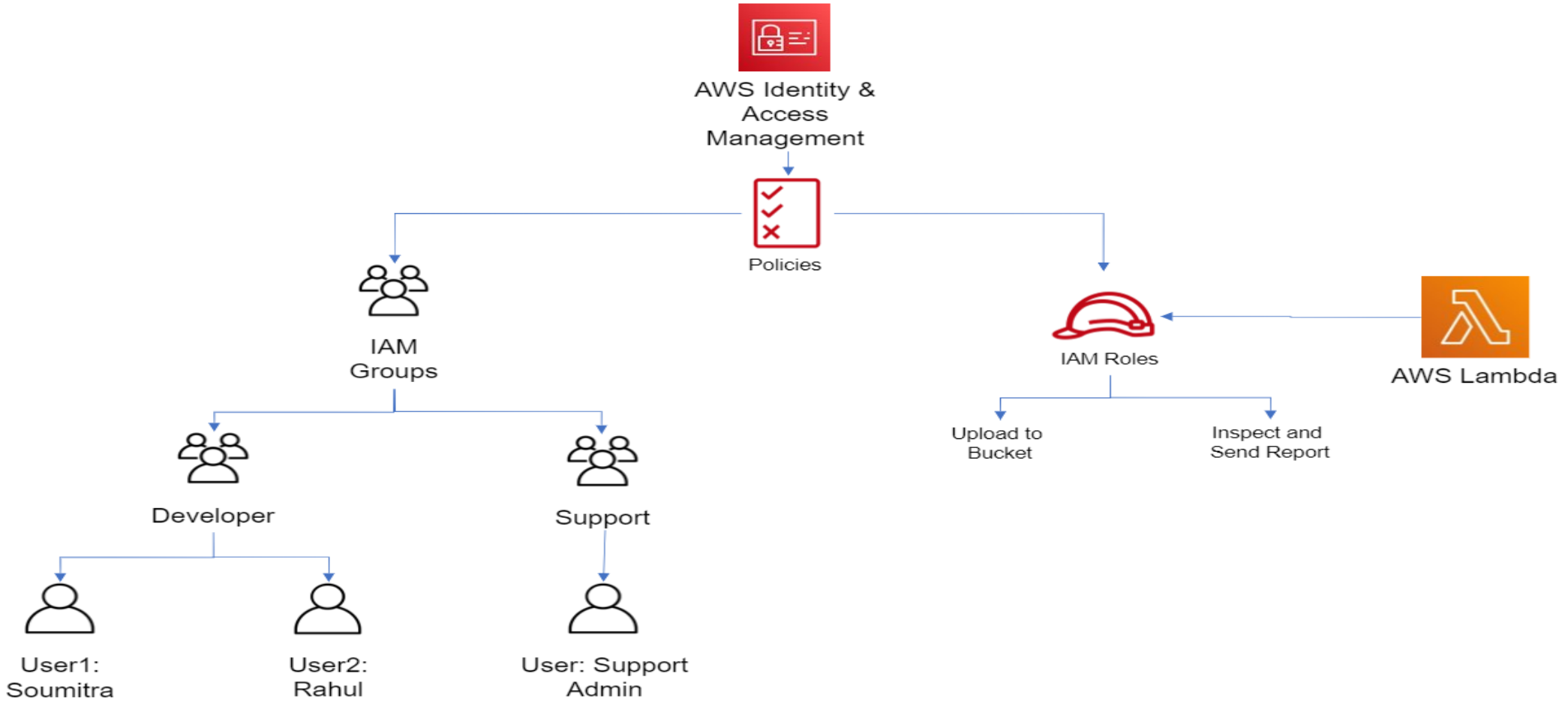
AWS S3 as the data tier to store original/inspected widget images



Architecture Diagram: AWS Components



Architecture : IAM Detail



Architecture Diagram : Component Summary

- Amazon API gateway acts as a reverse proxy to accept all application programming interface (API) calls .
- AWS Lambda lets us run Python codes when events are triggered without provisioning or managing servers.
- AWS Lambda interacts with other services to provide application functionalities.
- Amazon Simple Storage Service (Amazon S3) is an object storage service which is used in our application to store the original as well as the inspected widget images.
- AWS SNS used to send widget inspection reports to the Quality control group via Lambda.
- AWS Rekognition : Uses machine learning to inspect widget images and generate a detailed report
- AWS Cloudwatch : CloudWatch provides us with data and actionable insights to monitor the applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health

Design Considerations

- Created a new policy to allow the required AWS services access to Lambda functions (AWS S3, Rekognition, SNS)
- Created a quality control group for the support staff to monitor the health of the application.
- Designed the application with the AWS well architected framework policies to ensure reliability and running workloads on the cloud
- Limited access provided to Lambda functions using IAM policy to only be able to the required services.

Thank you