

AWS Pipeline - Full Design Doc

Version I - 3/28/23

High-Level Design

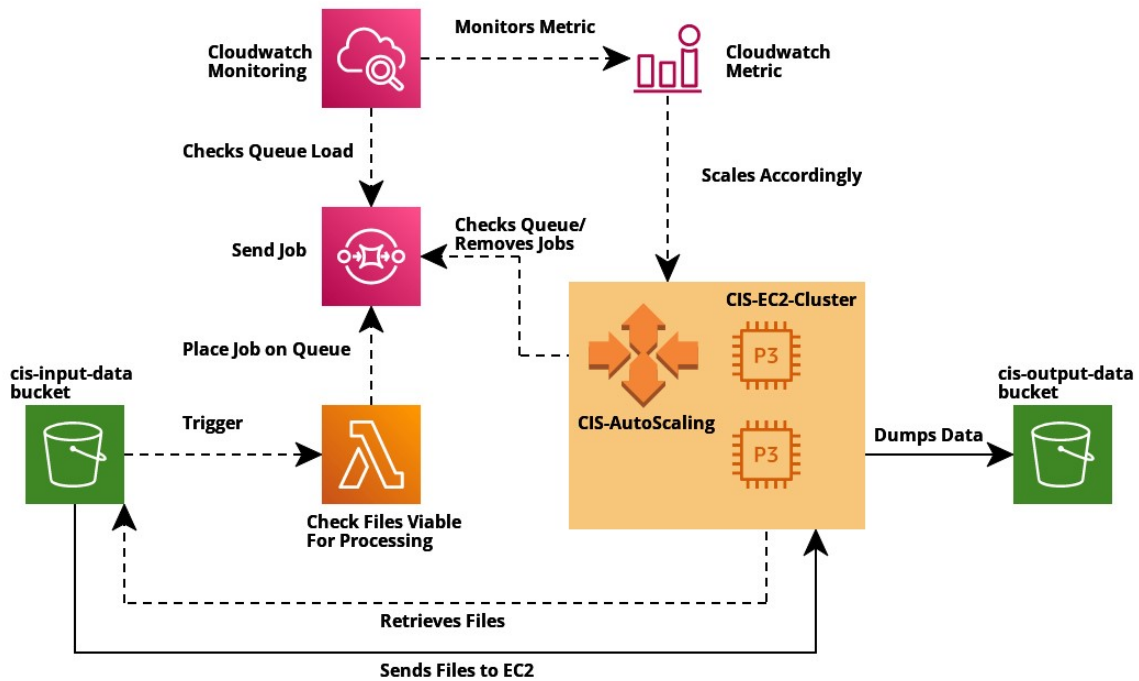


Figure 1 - High-Level Architecture

Overview

This pipeline is designed to process user data via AWS for IMU Spinal Calibration. The process first starts with uploading data to the bucket `cis-input-data` which then is used to trigger a lambda function that then places a job on an SQS queue. Cloudwatch is being utilized to monitor the queue to scale the AWS EC2 group accordingly based on demand. The EC2 instances within the group will always check the SQS queue for additional jobs, and remove them from the queue. Once the job is finished, the data is then dumped into an output S3 bucket called `cis-output-data`

Configuration Descriptions

The following must be done in order to ensure that the pipeline functions. The current pipeline contains these requirements, but any changes may require the installation of the following:

Code

Code that lambda functions, EC2 utilizes, and other services can be found here - <https://github.com/CurveAssure/CISII/tree/main/AWS>

Installations

`boto3` must be installed on any EC2 instances in this pipeline. It is recommended that this is downloaded onto a local machine if one chooses to conduct development there.

`CLI` - Command Line Interface, must be installed on any EC2 instances in this pipeline. It is recommended that is downloaded onto a local machine if one chooses to conduct development there.

Roles

Roles are designed to facilitate permissions and access between AWS services and users.

There is one role active:

- `CIS_Lambda_S3` - which gives permission to lambda to retrieve objects from S3 as well as deposit logs onto Cloudwatch for monitoring. The permissions required goes as follows:
 - `AmazonS3FullAccess`
 - `CloudWatchFullAccess`
 - `AWSLambda_FullAccess`
 - `PoliciesAmazonS3objectLambdaExecutionRolePolicy`

Specific Component Descriptions

S3 Buckets

S3 is an object storage service, where a bucket is a collection that stores files and folders.

There are two S3 buckets listed:

- `cis-data-input` - which is used to store data input, in the form of videos and sensor readings. There are two folders. The first folder is `sensor_data/` which stores all IMU sensor data. The second folder is `video_footage/` which stores video data corresponding to the IMU sensor data.
- `cis-data-output` - which is used to store data outputs. It does not take into consideration what those files are, though that may change with future iterations.

Lambda Functions

Lambda functions serve as short-term, low cost serverless computational systems.

There is one lambda function currently active:

- `CIS_S3_Data_Upload` - is a function that checks if both the data and video file is inputted into the S3 bucket. It is triggered from a new file inputted into `cis-data-input` bucket.

Once a new file has been inputted into a bucket, a series of actions follow:

1. Lambda first checks if both the video and IMU data files are in the bucket. If both are in the bucket, proceed with the next step.
2. Place a message on the SQS queue that a new job is ready to be processed, with the files associated with that job placed as the message.

SQS

SQS is a queue that stores and orders the listing of jobs that need to be processed.

There is currently one queue active:

- `CIS_Data_load_in_fifo` is a first in first out queue that stores the jobs that are needed to be processed. It is subscribed to the topic `CIS-EC2-Scaling-Example` .

Cloudwatch

Cloudwatch is a service that visualizes and monitors metrics.

There are no cloud watch alarms currently active as there are no EC2 groups active. However, below is documentation as to how to create those cloud watch alarms. The following system has been tested:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-using-sqs-queue.html>

EC2

EC2 is a service for provisioning and launching computational resources.

EC2 instances

An image of the instance that is desired to be scaled must be created. Currently, has not been established as our configuration needs to be fine-tuned to the data which is not yet available.

Hardware-wise, this instance will run on `p2.xlarge` instances, which are configured to run machine learning-based applications.

These instances also run a function `getQueue.py` to continuously check if there are new jobs placed on the queue and retrieve them for processing. Jobs are deleted when retrieved from the queue. The corresponding files to the job are retrieved from the S3 bucket.

EC2 Groups

A group of EC2 is provisioned to launch the instances based on the demand, which is determined by the queue and the cloud watch system mentioned above. There are currently no EC2 Groups cause none of them are being deployed at the moment due to insufficient data. This is strictly a cost-saving measure.