Growth Spurt Components Architecture and Design

Project Purpose:

The purpose of this project is to automate the process of getting record attachment hints from Family Search for our database. This should help our other applications that use these hints be more reliable since they will not run out of hints anymore which has been a constant issue.

Design Overview:

There are two main applications that we will be building for this project which are HarvesterGS and ShakerGS. HarvesterGS will be the application that retrieves hints from Family Search based on hints that have already been completed by other applications. It will be built off of AWS resources. It will have a lambda that triggers an ECS instance that will retrieve hints and pass it to another application called SerializerGS.

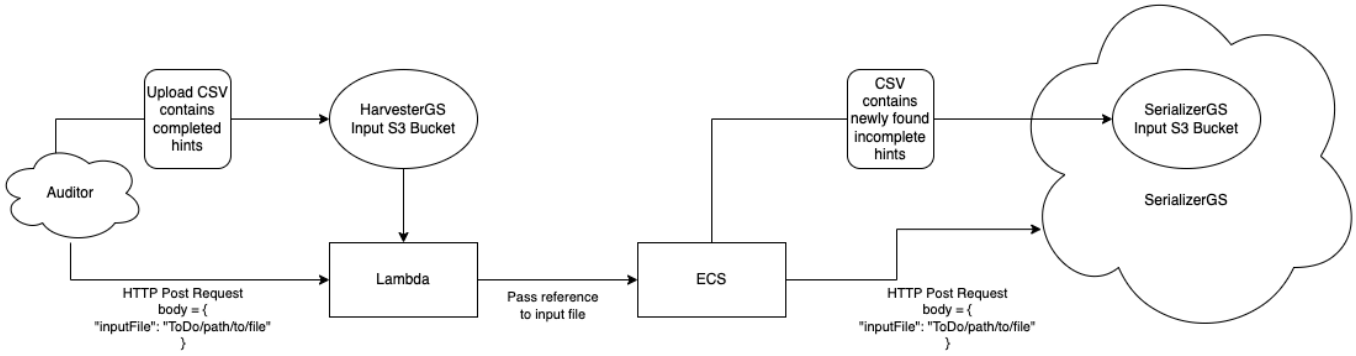
ShakerGS will have a similar structure to HarvesterGS with a lambda that triggers an ECS instance. We will not be building full functionality for ShakerGS but it will put the new hints into our DynamoDB database.

HarvesterGS:

When a hint is marked as complete by our Auditor application, it will be put into a csv that is uploaded to an AWS S3 bucket. The HarvesterGS lambda will then be triggered which has a reference to that csv file in the S3 bucket. The lambda then starts the HarvesterGS ECS task and sets certain environment variables that the ecs task uses to find the csv file. The ECS then uses the completed hints from the csv file to query Family Search for more hints. These new hints are then placed in a csv and sent to a different S3 bucket that SerializerGS uses. The HarvesterGS ECS finishes by triggering the SerializerGS lambda. The most difficult problem will be getting new hints from the completed hints.

Diagrams:

The full layout of HarvesterGS will look something like this:



The full flow of our Growth Spurt Components will look something like this: