DATA PIPELINE DOCUMENTATION

Table of Contents

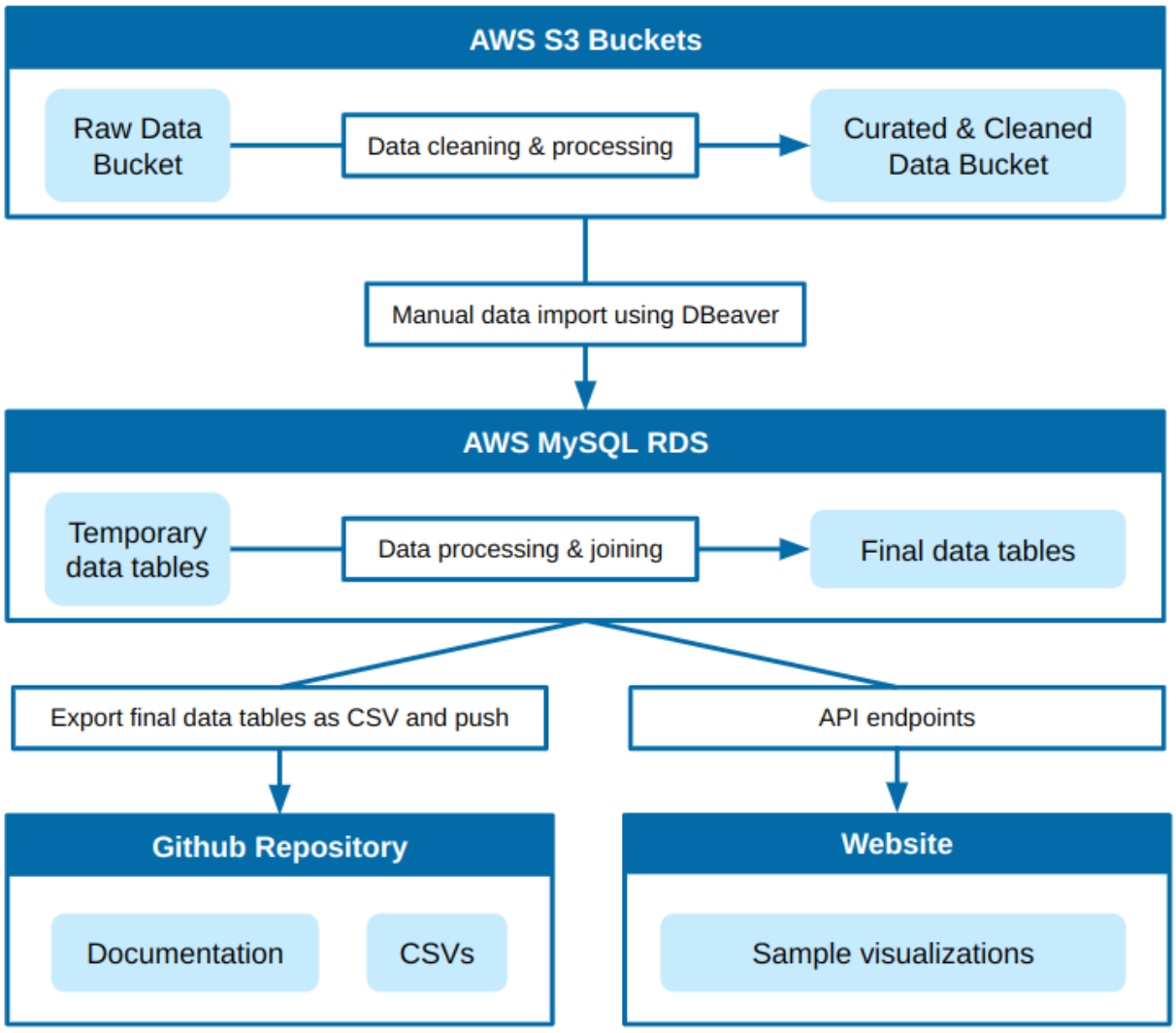
Introduction

This document describes the data pipeline used by the Winter 2023 INFO 498 Capstone class. It provides an overview of the overall pipeline, and then delves deeper into each component, explaining why each tool was chosen and how it is used in the context of this project.

Adapted from Microsoft's Aether Data Documentation Template. More information about this template can be found here.

Data Pipeline Overview

VISUAL OVERVIEW OF THE PIPELINE



The above is a visual representation of the project's data pipeline. Much of the data movement is manual; this is due to the small scale of the project and the need to ensure high-quality data.

AWS S3 BUCKETS

Once the raw data has been procured (either via web scraping or download of data files), it is placed in an AWS S3 Bucket. From there, it is then processed and cleaned using a Jupyter Notebook, and then placed into a separate AWS S3 Bucket that is exclusively for cleaned & curated data.

AWS RDS - MySQL

Our database is a relational database hosted in AWS, using a MySQL engine. Once the data is available in CSV format in the AWS S3 cleaned & curated data bucket, we manually download it and import it into the database using DBeaver. From there, it is processed and joined so that several larger data tables can be used to create visualizations or cross-reference data.

API

In order to provide the web team with the visualization data, we create API endpoints using AWS API Gateway service. Each API endpoint in the gateway service is powered by an AWS Lambda function where we call the queries to get the required data from our AWS RDS Database instance. The API endpoint is called on the client-side through Axios, a JS library used to make HTTP requests Node.js. From there, the web team utilizes the data from the API endpoint to create visualizations on the client side.

GitHub Repository

Once the data is ready in the RDS, it is exported in CSV format and pushed to the Github repository, which is open source.

Data Procurement

OVERVIEW

Our data procurement process began with an in-depth examination of our problem space. We focused on three main areas: existing Long COVID research, intersectionality of life for rural Latinos and Long COVID, and community-level impacts. Long COVID research led us through medical journals, studies, meta-analyses, and reporting that spanned large newspapers and individual newsletters. Looking at intersectionality, we explored: the occupations common to Eastern Washington, health insurance data, availability of medical services, household structures of Latino families, and vaccination rates. Finally, we sought to understand how these facts wove together to impact not only individuals but entire communities.

This research culminated in a thesis that hoped to capture the essence of what we learned, and what we hoped to demonstrate with our data repository. This thesis informed our data collection process and the entire open research project.

METHODOLOGY

Third Party Quantitative Data

As we each conducted individual initial research, our first step was to understand what we already had collectively found. We gathered our sources into a single spreadsheet and began the work of manually collecting the data. In most cases, data was able to be downloaded directly from the study or government database it originated from. For data that was not directly accessible, web scraping tools were built and used to collect the data. These tools can be found in the scripts AWS S3 bucket.

After gathering the sources we had, we began the process of validating our thesis. We broke out the thesis into individual claims – small pieces that could be proven or disproven on their own, such as a lack of medical infrastructure in Eastern Washington. For each claim, we examined our data sources and listed which ones would be applicable to validate that specific claim. This informed our search for additional data sources, as it allowed us to see any gaps left in the original research process. This iterative process was instrumental in creating a broad database that could address many lines of questioning.

First Party Qualitative Data

During the process of building the data repository, we had the opportunity to speak to several people who are directly affected by Long COVID, either as medical professionals, community leaders, or patients themselves. These interviews were semi-structured and were primarily held to inform our understanding of the lived experiences of Latinos with Long COVID – instead of presuming what they needed or had gone through, we wanted to ensure our approach was centered around the people we intended to help. When permitted by the interviewee, we recorded the interview and stored the recording in the raw data S3 bucket. We used the Amazon Transcribe tool to produce transcripts, which helped us understand the findings of the interviews.

We also established a process for gathering larger quantities of quantitative data, in the hopes that future researchers will carry forward this work. A more structured interview process was developed, and an in-depth questionnaire was written as well. More information about these processes can be found in the survey process documentation.

AWS S3 Buckets

OVERVIEW

Amazon Web Service's S3 Buckets are our main storage system. This system was chosen as we need the ability to store data in various formats, maintain file history, and enable remote access, all of which AWS provides. Additionally, the interview and survey responses are protected information. S3 allows for permissions to be set that control who can access files and in which ways. There are two buckets for data storage (`rawdata` and `curateddata`) and one for scripts.

RAW DATA BUCKET

The `rawdata` bucket contains data as it is downloaded from the source indicated in the data documentation. Files are manually uploaded and should be removed from local machines after uploading completes to avoid duplication issues. Files in this bucket have had no cleaning or processing, and in some cases may contain personally identifiable information. Access to this bucket should be carefully restricted in order to maintain the privacy of any survey respondents or interviewees. From here, this data should be processed using the `dataCleaning.ipynb` notebook before moving into the `curateddata` bucket. Any data which does not move to the `curateddata` bucket should still be kept in this folder in case it is needed for future iterations of the project. This bucket has restricted access, and the objects cannot be made public. This is to protect any personal information that may appear in interviews or surveys.

All contents of this bucket appear in the third-party data documentation.

DATA CLEANING & PROCESSING

Data processing and cleaning are done using the `dataCleaning.ipynb` Jupyter Notebook. This notebook takes in a CSV file (i.e. from the `rawdata` bucket) and outputs a CSV file to be added to the `curateddata` bucket. The notebook is divided into four main steps:

1. Load the data
2. Create a data overview
3. Functions to be applied to clean data
4. Generate file output

Further information about the use of these processes can be found within the notebook.

CLEANED & CURATED DATA BUCKET

The `curateddata` bucket contains data that has been processed using the `dataCleaning.ipynb` script. Files are manually uploaded after locally running the script. This

data is free of personally identifiable information and aligns with any feature labeling processes outlined in the data documentation. This data is moved directly into the public database and is all therefore fit for public consumption. Accordingly, the objects in this bucket can be made public.

All contents of this bucket appear in the third-party data documentation.

AWS MySQL RDS

OVERVIEW

We chose to use a MySQL engine for our AWS RDS because it integrates well with AWS' API service. Our database is not very large and does not contain that much data, so having a very powerful database engine was not a priority for us.

DATA IMPORTING

We use a manual process to transfer the data from the AWS S2 Bucket to our database. First, the data is downloaded as a CSV (or several CSVs) locally onto a computer via the AWS S3 interface. Then, using DBeaver, import tables are created to host the data. These tables have to be created in order to import the data; this is a DBeaver requirement. We chose to use DBeaver because we ran into issues with MySQL Community Workbench.

Once the import tables are ready, we follow this process to import the data:
https://dbeaver.com/2022/06/23/import-data-with-dbeaver/

FINAL DATA TABLES

You can find the code for and description of the final data tables in our database in our project GitHub repository README.md:
https://github.com/Adios-COVID/datarepo/blob/main/README.md.

API

You can find the description of our API endpoints in our project GitHub repository README.md:
https://github.com/Adios-COVID/datarepo/blob/main/README.md.