Social Media Analytics for Business Assignment

Implementation Guide

CIS 7029-20276556

2023

Table of Contents

[Introduction 1](#_Toc148175592)

[Project Overview 1](#_Toc148175593)

[Prerequisites 2](#_Toc148175594)

[Configure Reddit application 2](#_Toc148175595)

[Configure YouTube API 3](#_Toc148175596)

[Airflow Project 5](#_Toc148175597)

[Flask API Project 8](#_Toc148175598)

[React Project 9](#_Toc148175599)

[MySQL Database 9](#_Toc148175600)

[Dockerization 10](#_Toc148175601)

[Testing 11](#_Toc148175602)

# **Introduction**

Welcome to the Implementation Guide for the Social Media Assignment. This comprehensive guide aims to provide a step-by-step walkthrough of the process involved in developing an advanced social media analytics system. The system is comprised of three primary projects: Airflow for data scraping, Flask API for data processing, and React for data visualization. Furthermore, these projects will incorporate the utilization of a MySQL database for the effective storage and management of data. Additionally, Docker will be utilized for containerization purposes, ensuring smooth deployment and scalability.

# **Project Overview**

**Airflow Project**: The objective of this project is to implement automation techniques to streamline the process of gathering data from Reddit and YouTube. The utilization of Airflow, a robust workflow scheduler, would facilitate the process of collecting and ingesting data from various social media networks at predetermined intervals. In this study, we will examine the process of establishing Airflow, defining jobs for getting API data, implementing scheduling mechanisms, and monitoring procedures in order to guarantee the dependability of collected data.

**Flask API Project**: The Flask API functions as the central element for data processing. The system will take data from Airflow, perform data processing operations, and provide endpoints for consumption by the React frontend. The course will cover the process of developing a Flask API, executing data transformations, including error handling techniques, and establishing connections with a MySQL database.

**React Project**: The primary objective of the React project is to emphasize the representation of data through visual means. The processed data will be shown in an interactive dashboard that includes graphs and user-friendly functionalities. This tutorial will cover the process of configuring React, creating a dashboard layout, incorporating data from a Flask API, and implementing user interaction functionalities.

**MySQL Database**: I'll use MySQL as our relational database management system to make sure that data stays in place and can be quickly retrieved. You will learn how to use SQL queries to create database schemas, store data, and get that data.

**Dockerization**: The Dockerization section will show you how to make each project into a container so that it is easy to deploy and control. We'll put the Airflow, Flask API, React, and MySQL components in Docker containers and use Docker Compose to organize them.

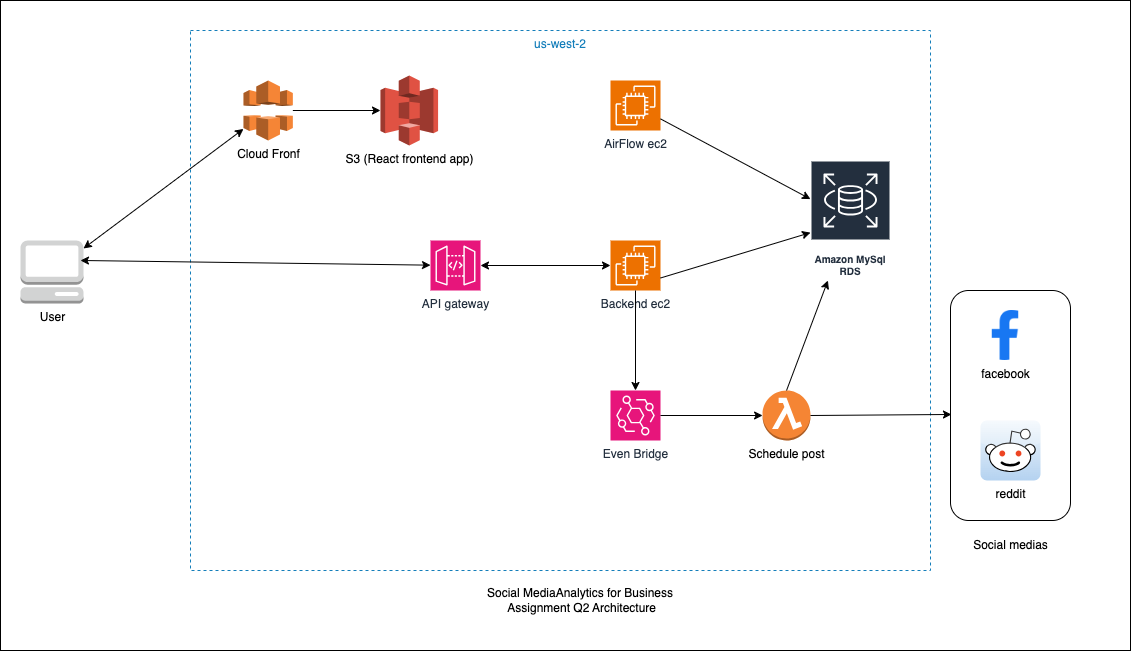


Figure Social Media Analytics for Business Architecture

# **Prerequisites**

Before beginning these tasks, it is necessary to meet certain prerequisites. To follow along, you will need specific software and tools. Ensure the following are installed:

* Docker Desktop: For containerization of project components.
* Node.js and npm (for React).
* MySQL Workbench: To manage the MySQL database.
* Visual Studio Code (VSCode): A code editor for developing and editing project code.

With these essential tools and technologies at your disposal, you will be prepared to embark on the exciting journey of building a social media analytics system from beginning.

## **Configure Reddit application**

In order to set up your Reddit account for development purposes, it is necessary to generate a Reddit application. By utilizing this method, users will be granted access to the Reddit API, enabling them to develop applications and utilities that facilitate interaction with the Reddit platform.

To create a Reddit application:

1. Go to <https://www.reddit.com/prefs/apps>.
2. Click the "Are you a developer? Create an app ..." button.
3. Fill out the form with the following information:

* Name: The name of your application.
* Application type: Select "Web app".
* Description: A brief description of your application.
* About URL: The URL of your application's website, if it has one.
* Redirect URL: The URL that Reddit will redirect users to after they authorize your application. This can be any URL, but it is recommended to use a URL on your application's website.
* Click the "Create app" button.

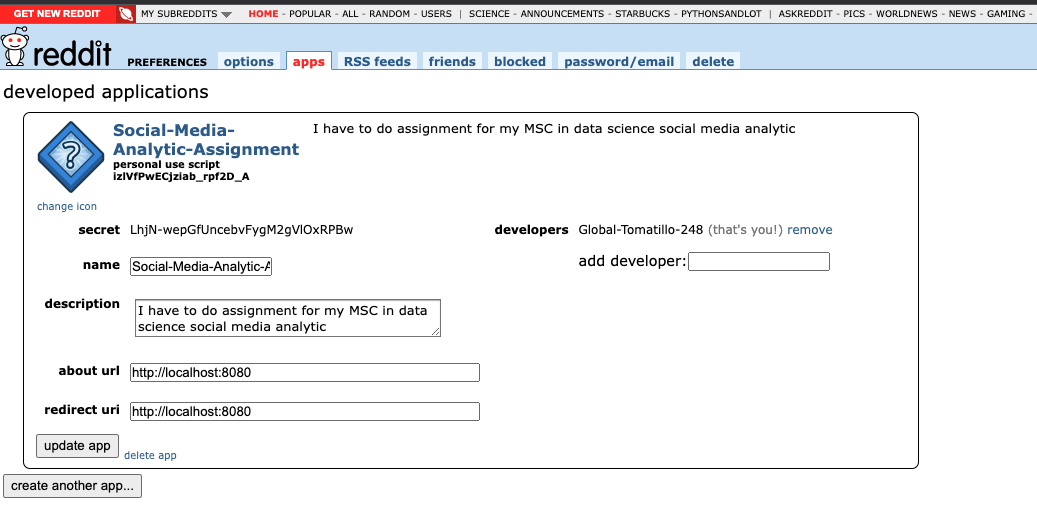


Figure Create Reddit application

Once you have created a Reddit application, you will be given a client ID and a client secret. These are important pieces of information that you will need to use your application to interact with the Reddit API.

To configure your Reddit account to use your new application:

1. Go to <https://www.reddit.com/prefs/apps>.
2. Click the "Manage apps" button.
3. Find your application in the list and click the "Edit" button.
4. Under the "Permissions" section, select the permissions that your application needs.
5. Click the "Save" button.

Once you have configured your Reddit account to use your application, you can start using the Reddit API to build apps and tools that interact with Reddit.

## **Configure YouTube API**

To configure the Google API for use with the YouTube API, you will need to create a project in the Google Developers Console and obtain authorization credentials.

To create a project in the Google Developers Console:

1. Go to the Google Developers Console: <https://console.developers.google.com/>
2. Click the "Create Project" button.
3. Enter a name for your project and click the "Create" button.

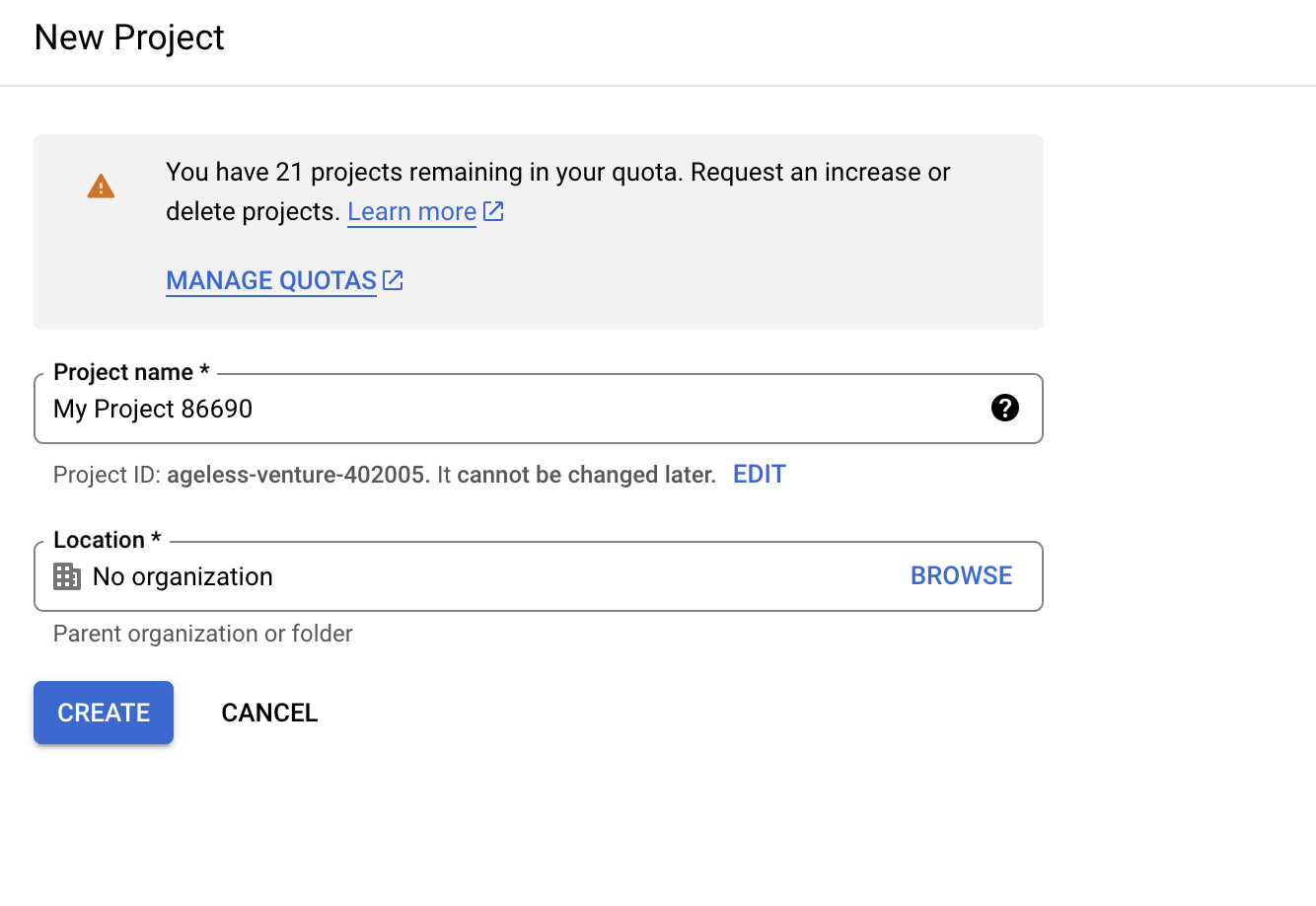


Figure Create Google project

Once you have created a project, you will need to enable the YouTube Data API. To do this:

1. Go to the "Library" page in the Google Developers Console.
2. Search for the "YouTube Data API v3" and click on it.
3. Click the "Enable" button.

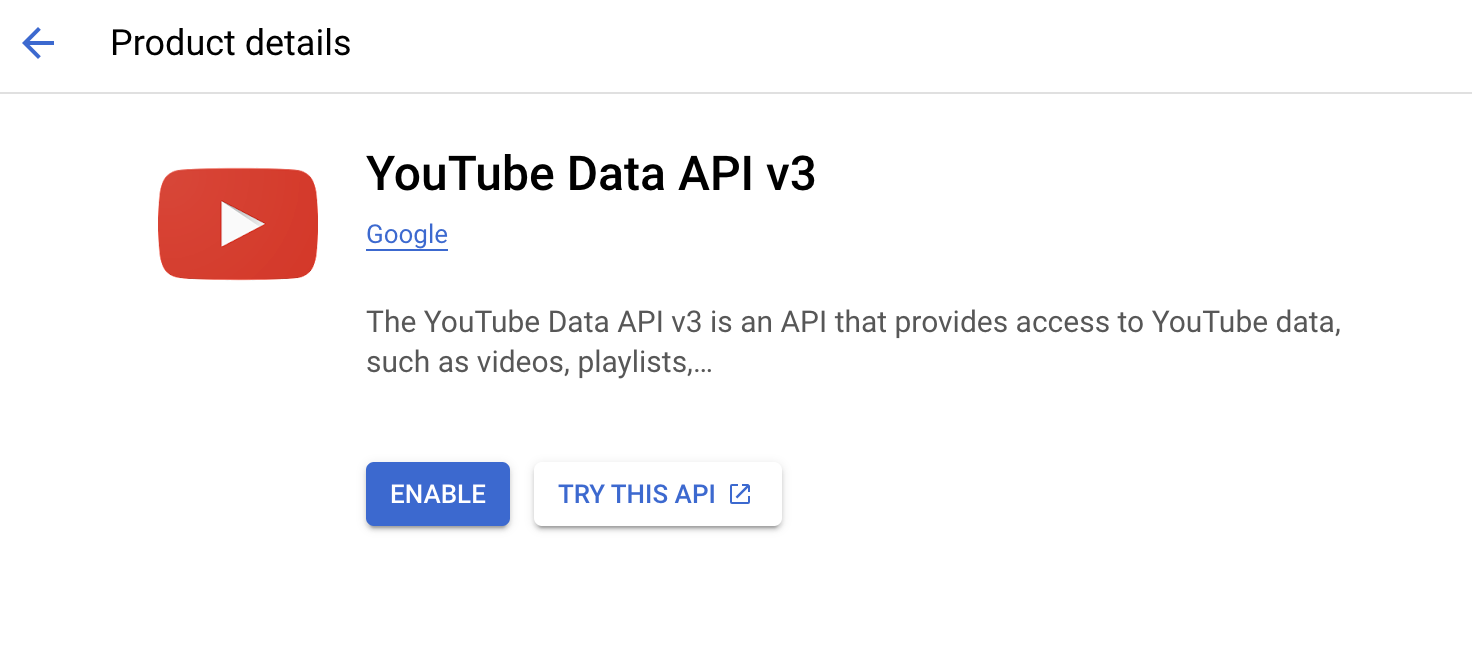


Figure Enable YouTube API

Next, you will need to obtain authorization credentials. You can do this by creating an API key or using OAuth 2.0.

To create an API key:

1. Go to the "Credentials" page in the Google Developers Console.
2. Click the "Create Credentials" button.
3. Select "API Key" from the list of options.
4. Click the "Create" button.
5. Copy the API key and store it in a safe place.

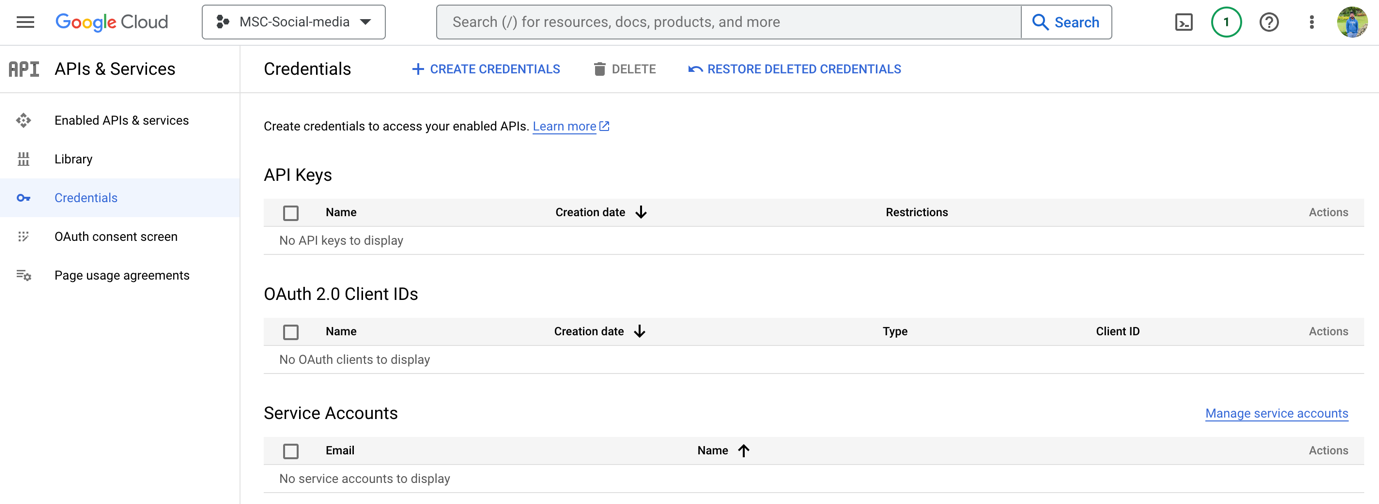


Figure Generate App Credentials

To use OAuth 2.0 authorization, you will need to create a client ID and client secret. To do this:

1. Go to the "Credentials" page in the Google Developers Console.
2. Click the "Create Credentials" button.
3. Select "OAuth client ID" from the list of options.
4. Select "Web application" as the application type.
5. Click the "Create" button.
6. Copy the client ID and client secret and store them in a safe place.

Once you have obtained authorization credentials, you can start using the YouTube API in this application.

# **Airflow Project**

**Step 1:** Make build.sh Executable

1. Open your terminal and navigate to the directory where your build.sh script is located.
2. Make the script executable by running the following command:

*chmod +x build.sh*

**Step 2:** Run build.sh to Set Up the Airflow Project

*./build.sh*

**Step 3:** Verify Configuration

1. Log in to the Airflow Web UI:

* Open a web browser and navigate to the Airflow web interface. Typically, it can be accessed at [http://localhost:8080](http://localhost:8080/) if you're running Airflow locally.
* Log in with your Airflow username and password.
  + Username: airflow
  + Password: airflow

1. Access the Admin Section:

* Once you're logged in, click on the "Admin" menu option in the top navigation bar. This will take you to the Admin section.

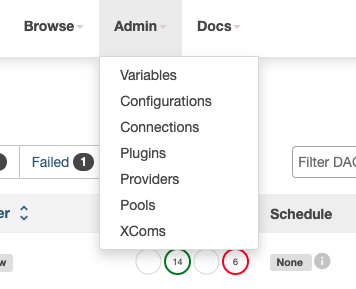


Figure Airflow Admin Menu

1. Click on "Connections":

* In the Admin section, you'll find various options. Click on "Connections" to manage database connections.

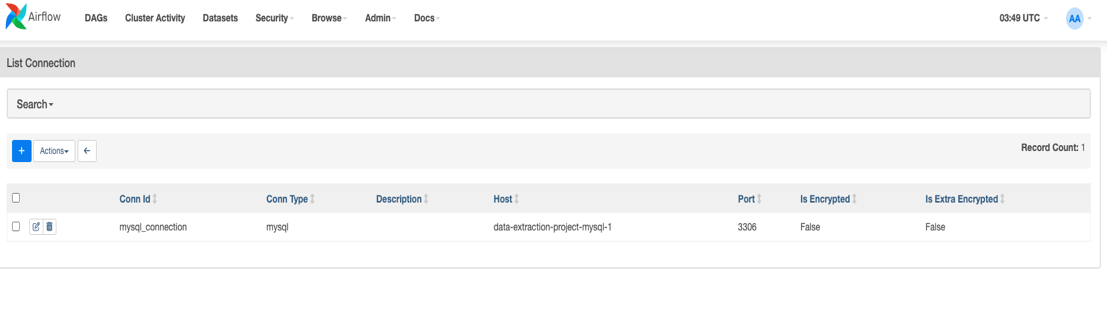


Figure Connection List

1. Create a New Connection:

* To create a new MySQL connection, click the "+ Add a new connection" button or a similar option depending on your Airflow version.

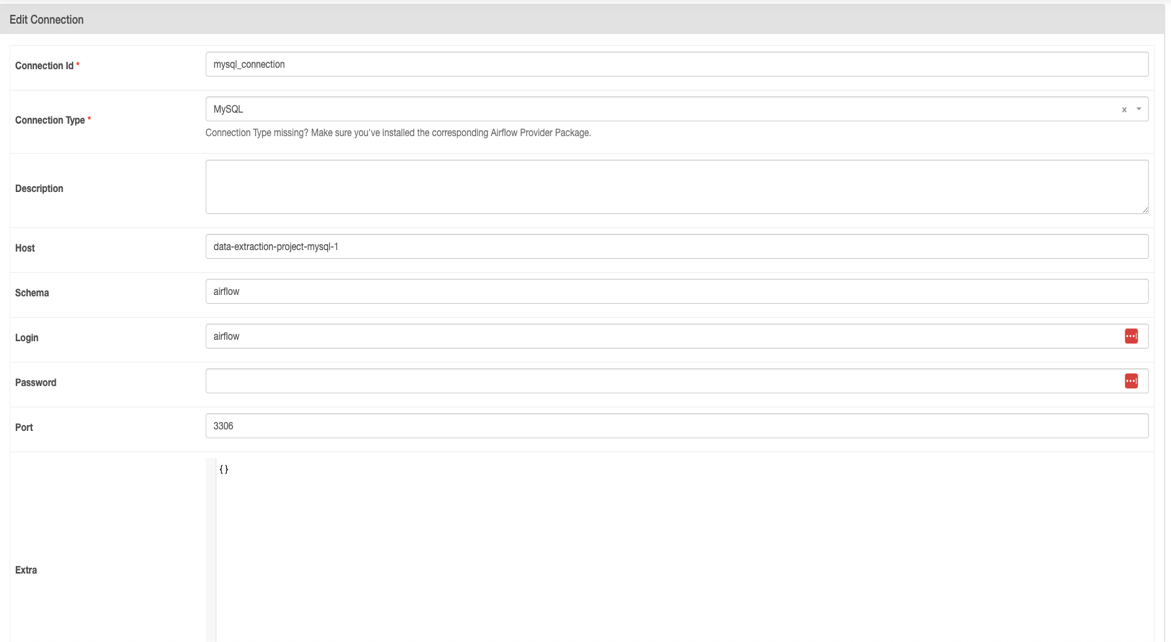


Figure Create MYSQL connection

* Go to your terminal and run this command:

*data-extraction-project % docker ps*

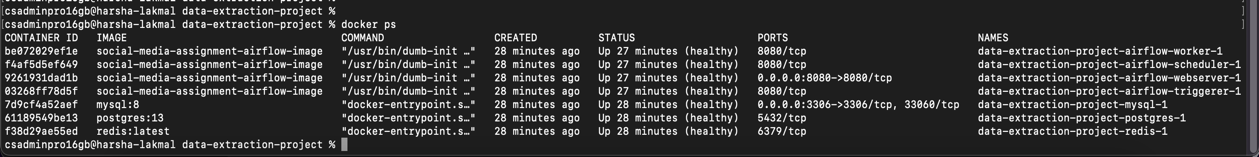


Figure Docker container list

You can see:

data-extraction-project-mysql-1 in your terminal. You can use as Host name.

1. Save the Connection:
2. Verify the Connection:

* You can verify that the connection is successfully established by clicking on the "Test Connection" button or by using it within your Airflow DAGs and tasks.

A white line on a white surface

Description automatically generated

Figure Save MYSQL configuration

1. Click on "Variables":

* In the Admin section, locate and click on the "Variables" option. This is where you can manage and configure variables.

A screenshot of a computer

Description automatically generated

Figure Add connection detail into variable list

After the configuration of Airflow DAGs, connections, and variables, the user can execute all the DAGs or specific DAGs through the Airflow web interface.

A screenshot of a computer

Description automatically generated

Figure Airflow Dag list

# **Flask API Project**

**Step 1:** Make build.sh Executable

1. Open your terminal and navigate to the directory where your build.sh script is located.
2. Make the script executable by running the following command:

*chmod +x build.sh*

**Step 2:** Update a.env File:

MYSQL\_DATABASE\_HOST=assignment-backend-mysql-1

MYSQL\_DATABASE\_USER=root

MYSQL\_DATABASE\_PASSWORD=123123

MYSQL\_DATABASE\_DB=social\_media\_db

REDDIT\_CLIENT\_ID=XXXXX XXXXX

REDDIT\_CLIENT\_SECRET=XXXXX XXXXX

REDDIT\_USER\_AGENT=XXXXX XXXXX

REDDIT\_USER\_NAME=XXXXX XXXXX

REDDIT\_PASSWORD=XXXXX XXXXX

**Step 3:** Run build.sh to Set Up the Flask Project

./build.sh

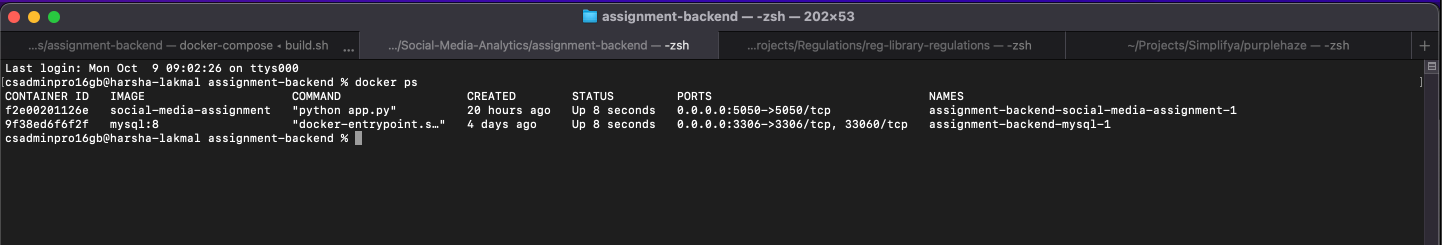


Figure Setup Flask project

# **React Project**

To set up a React project with yarn install and yarn start, you can use the following steps:

**Step 1:** Install Yarn, if you don't already have it:

npm install --global yarn

**Step 2:** Navigate to the project directory:

cd assignment-frontend

**Step 3:** Install the project's dependencies:

yarn install

**Step 4:** Start the development server:

yarn start

# **MySQL Database**

**Step 1:** Open MySQL Workbench

* Launch MySQL Workbench on your computer.

**Step 2:** Create a New Connection

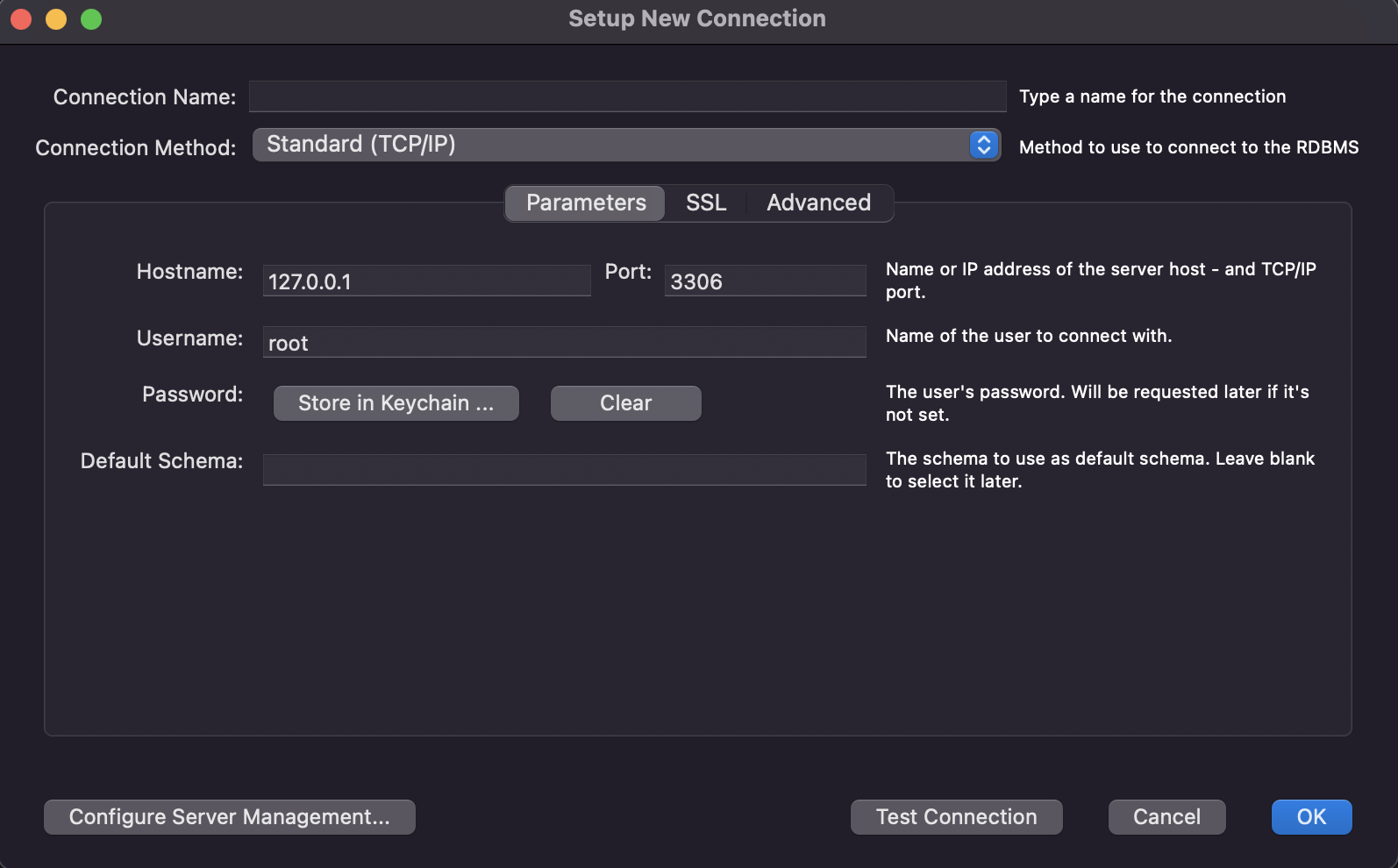


Figure Setup new connection

**Connection Name:** Give your connection a descriptive name (e.g., "Social Media Assignment").

**Connection Method:** Choose "Standard TCP/IP over SSH" or the appropriate method for your setup.

**Hostname:** Enter the hostname or IP address of your MySQL server. Port: Specify the port number for MySQL (usually 3306 by default).

**Username:** student

**Password:** std123

**Step 3:** Import SQL File

File\_name: **social\_media\_assignment\_db.sql**

I have attached these SQL file in dump folder.

# **Dockerization**

Dockerize the Airflow Project

* Check Running Docker Containers:
  + *docker ps*

*A screenshot of a computer

Description automatically generated*

Figure List of docker containers

Dockerize the Flask API Project

* Check Running Docker Containers:
  + *docker ps*

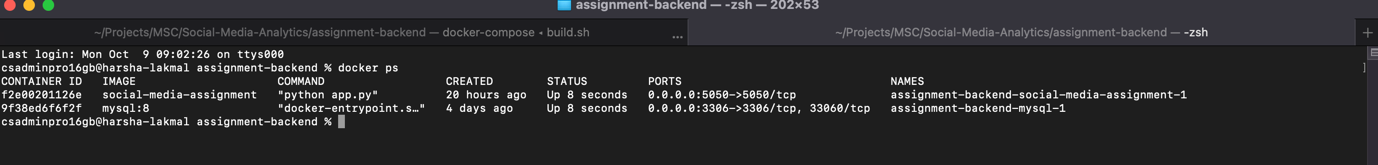


Figure Docker container list

# **Testing**

**Flask project**

**Open Postman:**

* Launch the Postman application on your computer.

**Import the Postman JSON File:**

* Click on the "Import" button in the top left corner of the Postman interface.

**Choose the File:**

* In the import dialog, click the "Choose Files" button to browse for and select the Postman JSON file that you want to import.

**Select the Import Type:**

* Choose the "File" option if you are importing a local file.Alternatively, choose the "Link" option if you have a publicly accessible URL for the JSON file.

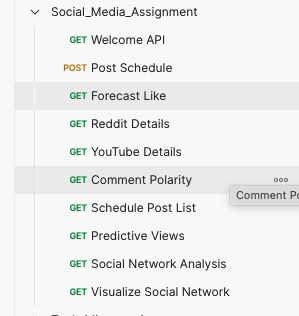


Figure Postman API list

These are APIs for Social media assignment dashboard.

**API Documentation**

Table : API documentation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API** | **Method** | **Description** | **Request payload** | **Response** |
| <http://localhost:5050/> | GET | Welcome API |  | Welcome My Social Media Analytics!! |
| <http://localhost:5050/reddit/reddit-details> | GET | Reddit Details API |  | {  "data": {  "comment\_count": 98460,  "like\_count": 2179195,  "subscriber\_count": 260693  }  } |
| <http://localhost:5050/reddit/youtube-details> | GET | YouTube Details API |  | {  "data": {  "comment\_count": 218,  "like\_count": 978,  "subscriber\_count": 23100,  "view\_count": 104007  }  } |
| <http://localhost:5050/reddit/predictive-likes?forecast_periods=7> | GET | Reddit forecast likes |  | {  "data": [  {  "color": "hsl(99, 70%, 50%)",  "data": [  {  "x": "2023-08-23",  "y": 2582  },  {  "x": "2023-08-24",  "y": 2161  },  {  "x": "2023-08-25",  "y": 2120  },  {  "x": "2023-08-26",  "y": 2116  },  {  "x": "2023-08-27",  "y": 2116  },  {  "x": "2023-08-28",  "y": 2116  },  {  "x": "2023-08-29",  "y": 2116  }  ],  "id": "likes"  }  ]  } |
| <http://localhost:5050/reddit/predictive-views?forecast_periods=7> | GET | YouTube like forecast |  | {  "data": [  {  "color": "hsl(99, 70%, 50%)",  "data": [  {  "x": "2023-09-11",  "y": 3133  },  {  "x": "2023-09-12",  "y": 3143  },  {  "x": "2023-09-13",  "y": 3143  },  {  "x": "2023-09-14",  "y": 3143  },  {  "x": "2023-09-15",  "y": 3143  },  {  "x": "2023-09-16",  "y": 3143  },  {  "x": "2023-09-17",  "y": 3143  }  ],  "id": "likes"  }  ]  } |
| <http://localhost:5050/reddit/social-network-analysis> | GET | Social media analytics details |  | {  "data": {  "basic\_network\_properties": {  "network\_density": 0.0008189568025879035,  "number\_of\_edges(Interactions)": 3200,  "number\_of\_nodes\_(Authors)": 2796  },  "betweenness\_centrality": [  {  "centrality": 0.25006226281441923,  "node": "MA6613"  },  {  "centrality": 0.21631212323629018,  "node": "Cupcake\_Great"  },  {  "centrality": 0.18258620966895325,  "node": "PossiblyUnhinged"  },  {  "centrality": 0.13476842431415212,  "node": "witchezbrew"  },  {  "centrality": 0.10609262817469442,  "node": "evviiieeee"  }  ],  "closeness\_centrality": [  {  "centrality": 0.36907434306087417,  "node": "MA6613"  },  {  "centrality": 0.3637428422696512,  "node": "Cupcake\_Great"  },  {  "centrality": 0.36298701298701297,  "node": "Stir-Bucks-Barista"  },  {  "centrality": 0.35828739905140367,  "node": "evviiieeee"  },  {  "centrality": 0.35033843068438203,  "node": "Successful\_Climate24"  }  ],  "degree\_centrality": [  {  "centrality": 0.14203935599284437,  "node": "MA6613"  },  {  "centrality": 0.12200357781753131,  "node": "Cupcake\_Great"  },  {  "centrality": 0.09803220035778175,  "node": "PossiblyUnhinged"  },  {  "centrality": 0.0740608228980322,  "node": "witchezbrew"  },  {  "centrality": 0.0626118067978533,  "node": "evviiieeee"  }  ]  }  } |
| <http://localhost:5050/reddit/visualize-social-network> | GET | Visualize communities |  |  |
| <http://localhost:5050/reddit/schedule-post> | POST | Schedule posts | {  "title": "Social media analytics assignment post",  "text": "This is my test post",  "tags": ["MSC", "Assignment"],  "url": "http://google.com",  "file": "",  "is\_one\_time": **true**,  "rule": "WE"  } |  |
| <http://localhost:5050/reddit/schedule-post> | GET | List Schedule posts |  |  |