Karande, Sarwesh Onkareshwar

Abstract

This Proof of Concept demonstrates the implementation of DataBridge, a versatile data integration application, connecting diverse APIs to databases.

Runbook: DataBridge Application Deployment and Operation

Contents

[Application Overview 2](#_Toc148278368)

[Pre-requisites 2](#_Toc148278369)

[Pre-Implementation Checklist 2](#_Toc148278370)

[DataBridge Application Deployment 2](#_Toc148278371)

[Post Deployment Validation of Services 2](#_Toc148278372)

[Test Cases 3](#_Toc148278373)

[Issues and Resolution During Installation 3](#_Toc148278374)

[References 3](#_Toc148278375)

# Application Overview

DataBridge is a versatile data integration application designed to connect any API to any database. It acts as a mediator, fetching data from REST APIs, processing it, and storing it in various databases, including MongoDB. The application architecture consists of a Flask API server, Apache Airflow DAGs (Directed Acyclic Graphs), and MongoDB for seamless data processing and storage.

It can work as a pluggable module in numerous data applications solving widely encountered data integration problems.

# Pre-requisites

* Docker installed on the host machine.
* Python 3.8 or higher installed.
* Proficiency in Docker and Docker Compose usage.
* Access to the required API endpoints for data retrieval.

Pre-Implementation Checklist

* Confirm the presence of Docker and Docker Compose on the host system.
* Verify accessibility of the API endpoints.
* Validate the MongoDB connection string and ensure the MongoDB server is reachable.
* Review and update configurations in the **.env** file if necessary.
* Ensure that ports 80, 8080, 8081, and 27017 are not in use.

DataBridge Application Deployment

1. **Clone the Repository:**

git clone <repository\_url> cd <repository\_directory>

1. **Configure Environment Variables:** Update essential environment variables in the **.env** file as needed.
2. **Build and Start the Application:**

docker compose up --build -d

1. **Accessing Services:**
   * Flask API: [http://localhost:8081/api/products](http://localhost/api/products)
   * Airflow UI: [http://localhost:8080](http://localhost:8080/)
   * MongoDB: mongodb://localhost:27017

# 

# Post Deployment Validation of Services

* Access the Flask API endpoint to ensure it returns the expected JSON response with product data.
* Log in to the Airflow UI (if credentials are set) and verify that the "DataBridgeDAG" is active and scheduled to run at regular intervals.
* Connect via MongoDB compatible database client to validate data.

# Test Cases

* **Test Case 1: Verify API Endpoint**
  + **Steps:**
    1. Access the Flask API endpoint: [http://localhost:8081/api/products](http://localhost/api/products)
    2. Verify if it returns a JSON response containing product data.
  + **Expected Result:** JSON response with product data.
* **Test Case 2: Airflow DAG Execution**
  + **Steps:**
    1. Access the Airflow UI: [http://localhost:8080](http://localhost:8080/)
    2. Check if the "DataBridgeDAG" is active and scheduled to run periodically.
  + **Expected Result:** Active "DataBridgeDAG" scheduled at regular intervals.

# Issues and Resolution During Installation

* **Issue 1: Unable to Access API Endpoint**
  + **Resolution:** Inspect API server logs for errors and confirm the correct API endpoint URL in **data\_ingress.py**.
* **Issue 2: Airflow DAG Fails to Execute**
  + **Resolution:** Examine Airflow logs (**docker logs <airflow\_container\_id>**) for errors and ensure accurate configurations in **DataBridgeDAG.py**.
* **Issue 3: MongoDB Connection Error**
  + **Resolution:** Validate the MongoDB connection string in **data\_store.py** and confirm the MongoDB server is operational and accessible.

# References

* Flask Documentation: [Link to Flask Documentation](https://flask.palletsprojects.com/en/2.1.x/)
* Apache Airflow Documentation: [Link to Airflow Documentation](https://airflow.apache.org/docs/apache-airflow/stable/index.html)
* MongoDB Documentation: [Link to MongoDB Documentation](https://docs.mongodb.com/)