**Building Data Pipelines with Apache Airflow and MongoDB**

**Overview**

***E - L - T Process***

Extract (E): I will extract data from the VnExpress website using Airflow, parallelizing the process with Celery to optimize performance.

Load (L): The extracted data will be loaded into a MongoDB architecture consisting of 3 shards, 1 config server, and 1 router. The data will be stored in the "articles" collection.

Transform (T): I will use the pre-trained VnCoreNLP machine learning model to transform the data, and the transformed results will be saved in the "articles\_transformed" collection.

***Docker Compose file***

**Config Server (configsvr):**

Image: MongoDB version 6.0.

Purpose: Acts as the configuration server to manage metadata for the cluster.

Command: Runs MongoDB in replica set mode with --configsvr enabled on port 27019.

Volumes: Stores configuration data and the keyfile for authentication.

Ports: Maps port 27019 on the container to the host.

Environment: Sets root credentials (username and password).

Network: Connects to a custom network mongo-cluster-network.

**Shards:**

Image: MongoDB version 6.0.

Purpose: First shard in the MongoDB sharded cluster, responsible for storing part of the data.

Command: Runs MongoDB in replica set mode with --shardsvr enabled on port 27018.

Volumes: Stores data and keyfile for authentication.

Ports: Maps port 27018 on the container to the host.

Environment: Sets root credentials.

Network: Also connects to mongo-cluster-network.

**Mongos Router (mongos)**:

Image: MongoDB version 6.0.

Purpose: Acts as the query router for the sharded cluster, directing client requests to the appropriate shard.

Command: Connects to the config server at configReplSet/configsvr:27019 and binds to all available IPs, running on port 27017.

Ports: Exposes port 27017 for external access.

Dependencies: Depends on the config server (configsvr) and all three shards (shard1, shard2, and shard3).

Volumes: Shares the keyfile for authentication.

Network: Connected to mongo-cluster-network.

**Volumes**:

Defines persistent storage for configuration data and shard data (configdb, shard1data, shard2data, shard3data).

**Network (mongo-cluster-network)**:

The custom network used by all services to communicate with each other. It uses the bridge driver for container networking.

**Practice**

First, move the code file to the Airflow DAGs directory and install the required libraries:

cp DAG\_etl.py ~/airflow/dags

source ~/airflow\_env/bin/activate

chmod +x install\_me.sh

./install\_me.sh

Next, deploy the Docker Compose file using the following command:

docker-compose up -d

In a new tab, start the Airflow web server:

source ~/airflow\_env/bin/activate

airflow webserver --port 8080

Then, access the web interface at http://localhost:8080

In another new tab, run the scheduler to schedule tasks:

source ~/airflow\_env/bin/activate

airflow scheduler

In a separate tab, start the Celery workers:

source ~/airflow\_env/bin/activate

airflow celery worker

At this point, the data pipeline is fully set up. Now you just need to monitor the following:

Logs on the Airflow web Server

A screenshot of a computer

Description automatically generated

The number of commands and inserts on MongoDB Compass

A screenshot of a computer

Description automatically generated

RAM consumption on WSL2 Ubuntu

A screenshot of a computer screen

Description automatically generated