Kubectl create ns simaproject

Step1:

Make your AIRFLOW code

vim code.py

    database = 'simadb'

    engine = create\_engine(f'mysql+pymysql://{username}:{password}@{hostname}:{port}/{database}')

    conn = engine.connect()

    sql = 'SELECT \* FROM simadb.tradedataTB'

    result = conn.execute(sql)

    data = result.fetchall()

    columns = result.keys()

    df = pd.DataFrame(data, columns=columns)

    return df

def upload\_data\_to\_mysql(df):

    import pandas as pd

    from sqlalchemy import create\_engine

    hostname = "mysql-container.simaproject.svc.cluster.local"  # Use the Kubernetes service DNS

    username = 'root'

    password = '1234'

    port = 3306

    database = 'simadb'

    engine = create\_engine(f'mysql+pymysql://{username}:{password}@{hostname}:{port}/{database}')

    df.to\_sql("tradedataTBnew", con=engine, if\_exists='replace', index=False)

    engine.dispose()

    print('Successfully uploaded to MySQL')

with DAG(

    dag\_id="etl\_pipeline\_from\_mysql",

    schedule="\* \* \* \* \*",

    start\_date=datetime(2024,1,1),

    catchup=False,

    tags=["etl\_by\_airflow"],

) as dag:

    get\_data=PythonOperator(

        task\_id='get\_data\_from\_mysql',

        python\_callable=get\_data\_from\_mysql,

    )

    upload\_data= PythonOperator(

        task\_id='upload\_data\_to\_mysql',

        python\_callable=upload\_data\_to\_mysql,

        op\_args=[get\_data.output],  # Pass the output of get\_data\_task as an argument

    )

    get\_data>>upload\_data

2) step 2 prepare all libraries that are needed to apply your code:

vim requirements.txt

pandas

sqlalchemy

pymysql

apache-airflow

step3: it gives you a dockerfile to build your image(in this project image is run in the host so )

vim Dockerfile

vim # Use a base Python image with the version you need

FROM python:3.9-slim

# Set the working directory inside the container

WORKDIR /app

# Copy your Python script (code.py) into the container

COPY code.py /app/

# Optionally, if your Python script has dependencies, create a requirements.txt file and add it here

# COPY requirements.txt /app/

# RUN pip install -r requirements.txt

# Install required dependencies, such as Airflow, MySQL libraries, and any other libraries in your script

RUN pip install pandas sqlalchemy pymysql apache-airflow

# Set the command to run your Python script when the container starts

CMD ["python", "code.py"]

Step 4: now we build the image inside our host:

docker build -t my-python-etl-image:latest .

step5:

now we prepare the code that provides a container for us to run our database

vim mysqlcontainer.yaml

apiVersion: v1

kind: Pod

metadata:

  name: mysql-container

  namespace: simaproject

spec:

  containers:

    - name: mysql

      image: mysql:5.7

      env:

        - name: MYSQL\_ROOT\_PASSWORD

          value: "1234" # Root password for MySQL

        - name: MYSQL\_DATABASE

          value: "simadb" # Pre-created database

      ports:

        - containerPort: 3306 # MySQL port

      resources:

        requests:

          memory: "256Mi"

          cpu: "500m"

        limits:

          memory: "512Mi"

          cpu: "1000m"

kubectl apply -f mysqlcontainer.yaml -n simaproject

Step 6:

Now we run cronjob:

vim etl-cronjob.yaml

vim apiVersion: batch/v1

kind: CronJob

metadata:

  name: etl-cronjob

  namespace: simaproject

spec:

  schedule: "\*/5 \* \* \* \*"  # Runs every 5 minutes

  jobTemplate:

    spec:

      parallelism: 2  # Number of Pods to run in parallel

      completions: 2  # Number of completions required

      activeDeadlineSeconds: 300  # Timeout for each job (5 minutes)

      template:

        spec:

          nodeSelector:

            kubernetes.io/hostname: minikube  # Use the 'minikube' node name

          containers:

            - name: python-worker

              #image: 192.168.49.1:31326/my-python-etl-image:latest # Updated with Minikube IP

              image: my-python-etl-image:latest

              imagePullPolicy: Never

                #imagePullPolicy: IfNotPresent  # Try to pull only if not already present

              env:

                - name: MYSQL\_HOST

                  value: "mysql-container.simaproject.svc.cluster.local"  # MySQL service hostname

                - name: MYSQL\_PORT

                  value: "3306"

                - name: MYSQL\_USER

                  value: "root"

                - name: MYSQL\_PASSWORD

                  value: "1234"

                - name: MYSQL\_DATABASE

                  value: "simadb"

              resources:

                requests:

                  memory: "256Mi"  # Adjust based on your actual needs

                  cpu: "100m"      # Further reduced CPU request (0.1 CPU cores)

                limits:

                  memory: "512Mi"  # Limit memory usage

                  cpu: "200m"      # Further reduced CPU limit (0.2 CPU cores)

          restartPolicy: Never  # Pods will not restart once the script completes

            #imagePullSecrets:

            #- name: my-registry-secret

kubectl apply -f etl-cronjob.yaml -n simaproject

kubectl get pods -n simaproject -w

Possible errors can be removed by these commands:

Kubectl descrime pods PODS\_NAME -n simaproject