MEDIRECT

DATA ENGINEERING ASSESSMENT

Python Developer

ENVIRONMENT SETUP

Apache Airflow
Postgres
Jenkins

APACHE AIRFLOW

Installation Procedure

- Setting up of Apache Airflow v2.7.2 docker stack on a local environment (macOS) was carried by following through the official installation guides.
- The official *docker-compose.yml* file includes the provision of CeleryExecutor rather than the LocalExecutor. The latter executor is capable of running tasks sequentially – one task instance at a time and must reside on the same machine as the scheduler.
- Minor changes applied to docker yaml file:
  - Enabling port forwarding for PostgreSQL database server to hosting machine i.e TCP 5432:5432. This is required to provision the database and database tables via Azure Data Studio.
  - A *dockerfile* to enable the provision of additional python libraries via *requirements.txt*
  - Explicit use of a docker env file - docker compose up airflow-init --env-file docker-compose.env (rather than hidden .env file)
- Inclusion of *airflow.sh* wrapper script to invoke commands on the web server via docker compose/terminal CLI.

  e.g Creation of a new application user having admin rights.

```
./airflow.sh users create --username mark --password mark --firstname Mark --lastname Grech --
role Admin --email grechmarkj@gmail.com
```

APACHE AIRFLOW METASTORE

Installation Procedure

- The creation and initialisation of the Airflow's metastore took place automatically during the initial setup process.
- Environment variables pertaining to the metastore were left unchanged.

```
AIRFLOW__DATABASE__SQL_ALCHEMY_CONN
AIRFLOW__DATABASE__SQL_ALCHEMY_SCHEMA
```

- However, there was an instance where `airflow db migrate` command had to to explicitly invoked.
- Specification of a project folder. By default, docker compose creates a project folder in the same directory where the yaml file resides.

```
AIRFLOW_PROJ_DIR=/Volumes/DATA2/Docker/airflow_stack_official/
```

- Avoid the loading of demo DAGS

```
AIRFLOW__CORE__LOAD_EXAMPLES=False
```

- User UID is set in response to the warning message reported during the provision of containers. Documentation states that the UID denotes the user to run containers as and could therefore affect the mounting of shared volumes.

```
AIRFLOW_UID=50000
```

AIRFLOW CONFIGURATION

- The database and API endpoints are defined as Airflow Variables and Connections.
- Defined programmatically via shell script - *init_connections.sh*
  - Deletes existing objects.
  - Creates connections and variables.
- **List of Connections**
  - *api_endpoint_openholidaysapi_org* - airflow connection to openholidaysapi.org REST API service.
  - *api_endpoint_openerapi_com_v6* - airflow connection to open.er-api.com REST API service.

- **List of Variables**
  - *Environment* - Denotes whether the machine is a development, uat, pre-pod or a production environment. The environment variable is read during the execution of the DAG files. In a non-production setting, data is logged for debugging purposes.

AZURE DATA STUDIO / POSTGRES DB

- Installation of PostgreSQL plugin since only MS SQL server is supported by default.
- Testing connectivity to the local database instance using default administrative credentials.
- Creation of *db_currency_exchange* database and database objects via the DDL script.
- The creation of the underlying database objects was performed outside of the data pipelines.

  The rationale behind this approach is based on the separation of data manipulation and database definition tasks as data pipelines are typically complex. Embedding DDL scripting may result in additional maintenance/support efforts.
- The creation of an application database user having restricted permissions on the default database schema (*airflow_app_usr*).

JENKINS

- The provision of a Jenkins docker container to enable CI/CD.
- Reverted to the official and latest version of Jenkins Docker container.
- Added SSH Agent plugin along with other standard plugins.

APACHE AIRFLOW

airflow.sh info

```
Apache Airflow
version                 | 2.7.2
executor                | CeleryExecutor
task_logging_handler    | airflow.utils.log.file_task_handler.FileTaskHandler
sql_alchemy_conn        | postgresql+psycopg2://airflow:airflow@postgres/airflow
dags_folder             | /opt/airflow/dags
plugins_folder          | /opt/airflow/plugins
base_log_folder         | /opt/airflow/logs
remote_base_log_folder |


System info
OS                  | Linux
architecture        | x86_64
uname               | uname_result(system='Linux', node='b081596f7361', release='6.4.16-linuxkit', version='#1 SMP PREEMPT_DYNAMIC Tue Oct 10 20:42:40 UTC 2023',
                    | machine='x86_64', processor='')
locale              | ('en_US', 'UTF-8')
python_version      | 3.8.18 (default, Oct 11 2023, 23:57:43)  [GCC 10.2.1 20210110]
python_location     | /usr/local/bin/python


Tools info
git                 | NOT AVAILABLE
ssh                 | OpenSSH_8.4p1 Debian-5+deb11u2, OpenSSL 1.1.1w  11 Sep 2023
kubectl             | NOT AVAILABLE
gcloud              | NOT AVAILABLE
cloud_sql_proxy     | NOT AVAILABLE
mysql               | mysql  Ver 8.0.34 for Linux on x86_64 (MySQL Community Server - GPL)
sqlite3             | 3.34.1 2021-01-20 14:10:07 10e20c0b43500cfb9bbc0eaa061c57514f715d87238f4d835880cd846b9ealt1
psql                | psql (PostgreSQL) 16.0 (Debian 16.0-1.pgdg110+1)


Paths info
airflow_home        | /opt/airflow
system_path         | /root/bin:/home/airflow/.local/bin:/usr/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
python_path         | /home/airflow/.local/bin:/usr/local/lib/python38.zip:/usr/local/lib/python3.8:/usr/local/lib/python3.8/lib-dynload:/home/airflow/.local/lib/pytho
                    | n3.8/site-packages:/usr/local/lib/python3.8/site-packages:/opt/airflow/dags:/opt/airflow/config:/opt/airflow/plugins
airflow_on_path     | True


Providers info
apache-airflow-providers-amazon           | 8.7.1
apache-airflow-providers-celery           | 3.3.4
apache-airflow-providers-cncf-kubernetes  | 7.6.0
apache-airflow-providers-common-sql       | 1.7.2
apache-airflow-providers-daskexecutor     | 1.0.1
apache-airflow-providers-docker           | 3.7.5
apache-airflow-providers-elasticsearch    | 5.0.2
apache-airflow-providers-ftp              | 3.5.2
apache-airflow-providers-google           | 10.9.0
apache-airflow-providers-grpc             | 3.2.2
apache-airflow-providers-hashicorp        | 3.4.3
apache-airflow-providers-http             | 4.5.2
apache-airflow-providers-imap             | 3.3.2
apache-airflow-providers-microsoft-azure  | 7.0.0
apache-airflow-providers-mysql            | 5.3.1
apache-airflow-providers-odbc             | 4.0.0
apache-airflow-providers-openlineage      | 1.1.0
apache-airflow-providers-postgres         | 5.6.1
apache-airflow-providers-redis            | 3.3.2
apache-airflow-providers-sendgrid         | 3.2.2
apache-airflow-providers-sftp             | 4.6.1
apache-airflow-providers-slack            | 8.1.0
apache-airflow-providers-snowflake        | 5.0.1
apache-airflow-providers-sqlite           | 3.4.3
apache-airflow-providers-ssh              | 3.7.3

(base) mark@imac-9490m Docker_official %
```

APACHE AIRFLOW

`docker ps`

```
(base) mark@imac-9490m ~ % docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED        STATUS                    PORTS                    NAMES
249842710fca   apache/airflow:2.7.2   "/usr/bin/dumb-init …"   22 hours ago   Up 25 minutes (healthy)   8080/tcp                 docker_official-airflow-triggerer-1
1da51203c2c0   apache/airflow:2.7.2   "/usr/bin/dumb-init …"   22 hours ago   Up 25 minutes (healthy)   8080/tcp                 docker_official-airflow-worker-1
a2f7d3c21595   apache/airflow:2.7.2   "/usr/bin/dumb-init …"   22 hours ago   Up 25 minutes (healthy)   8080/tcp                 docker_official-airflow-scheduler-1
2dc893bafcdb   apache/airflow:2.7.2   "/usr/bin/dumb-init …"   22 hours ago   Up 25 minutes (healthy)   0.0.0.0:8080->8080/tcp   docker_official-airflow-webserver-1
22e30d20f4e4   postgres:13            "docker-entrypoint.s…"   22 hours ago   Up 25 minutes (healthy)   0.0.0.0:5432->5432/tcp   docker_official-postgres-1
01c3ded706dd   redis:latest           "docker-entrypoint.s…"   22 hours ago   Up 25 minutes (healthy)   6379/tcp                 docker_official-redis-1
```

PYTHON TRANSFORMATION JOBS

DAGS

DAGS

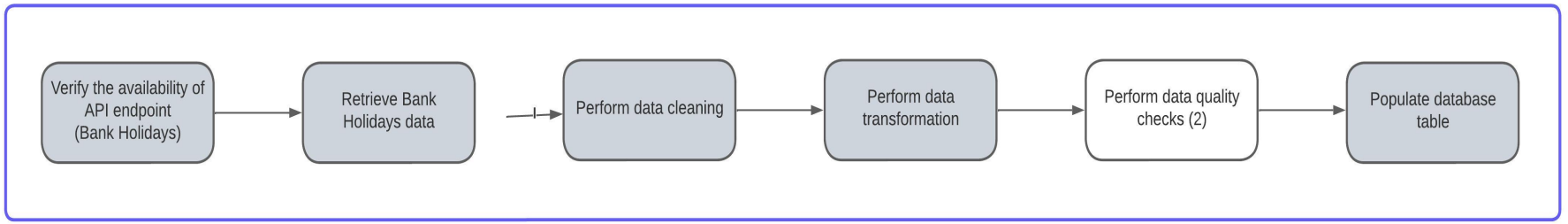
1. Two separate pipelines were designed since each pipeline has different execution times.
2. The bank holiday pipeline is required to be executed on a yearly basis. Data pertaining to the specified country, in this case Belgium is retrieved in a single request.
3. In contrast, the most recent currency exchange rates need to be retrieved daily.
4. The raw bank holiday dataset includes a start date and an end date. It was noted that all the public holidays span over a single day.
5. However, should a public holiday span over multiple days, the data transformation routine caters for such instances whereby multiple data records are created representing each day.

DAGS

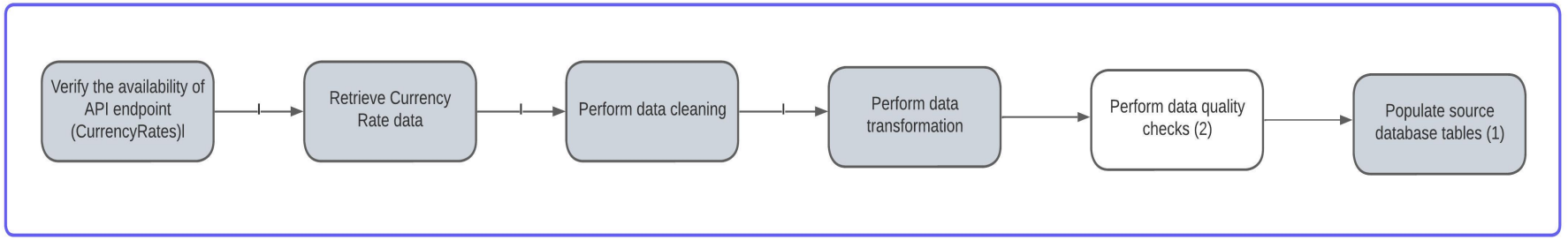
6. Data exchange between tasks operators occurs through XCOM and objects are relayed in form of a dictionary. This approach is thought to be extensible vis-à-vis other data structures such as lists or tuples whenever additional is required to the passed between processes.

DAGS

public_holidays_dag



currency_rate_dag

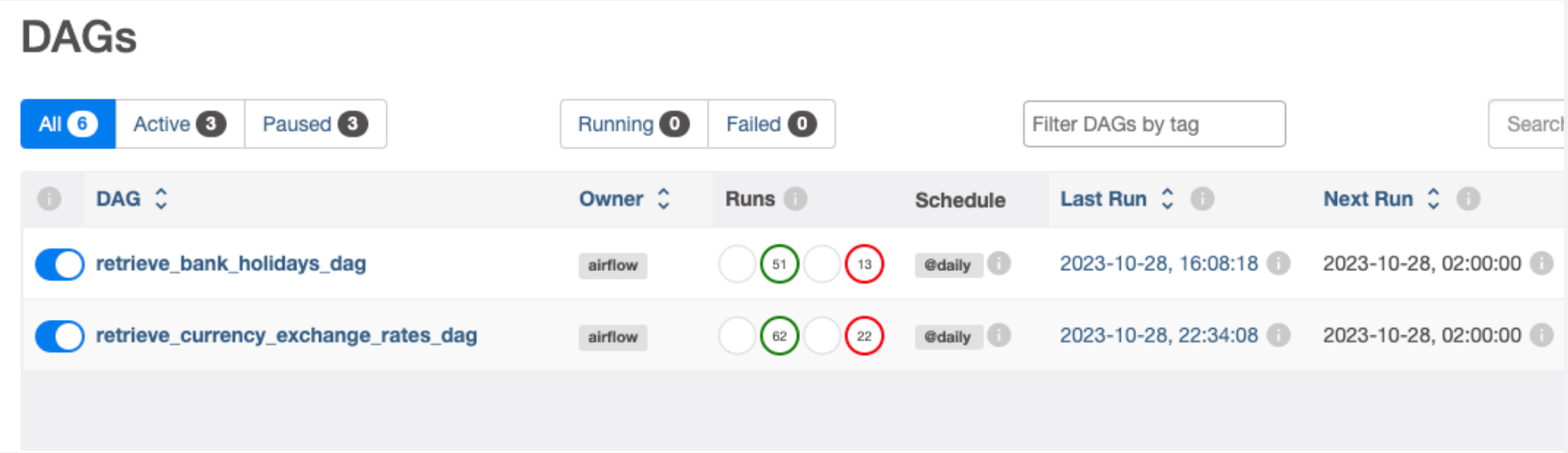


1. Source tables are populated in a single database transaction

2. A task can be added to a data pipeline that would perform data quality checks prior to persisting data in DB

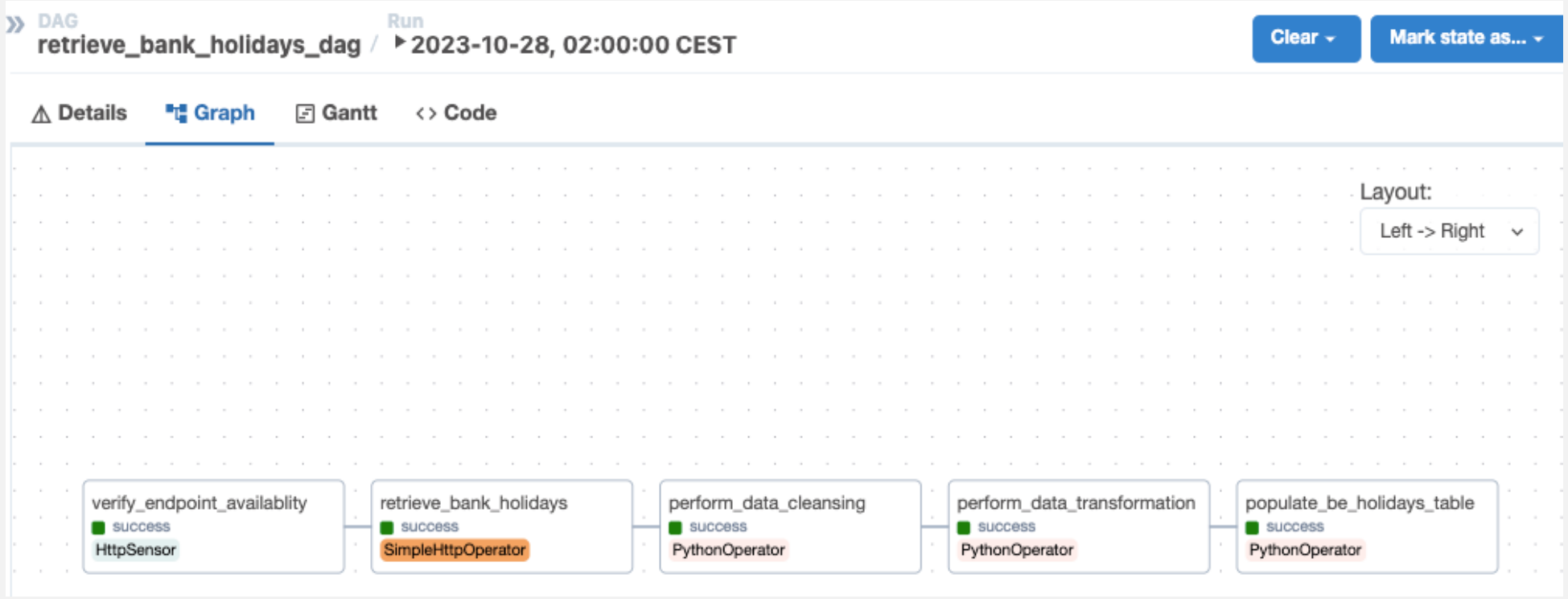
DAGS

The two data pipelines as listed in Airflow's DAGs GUI.



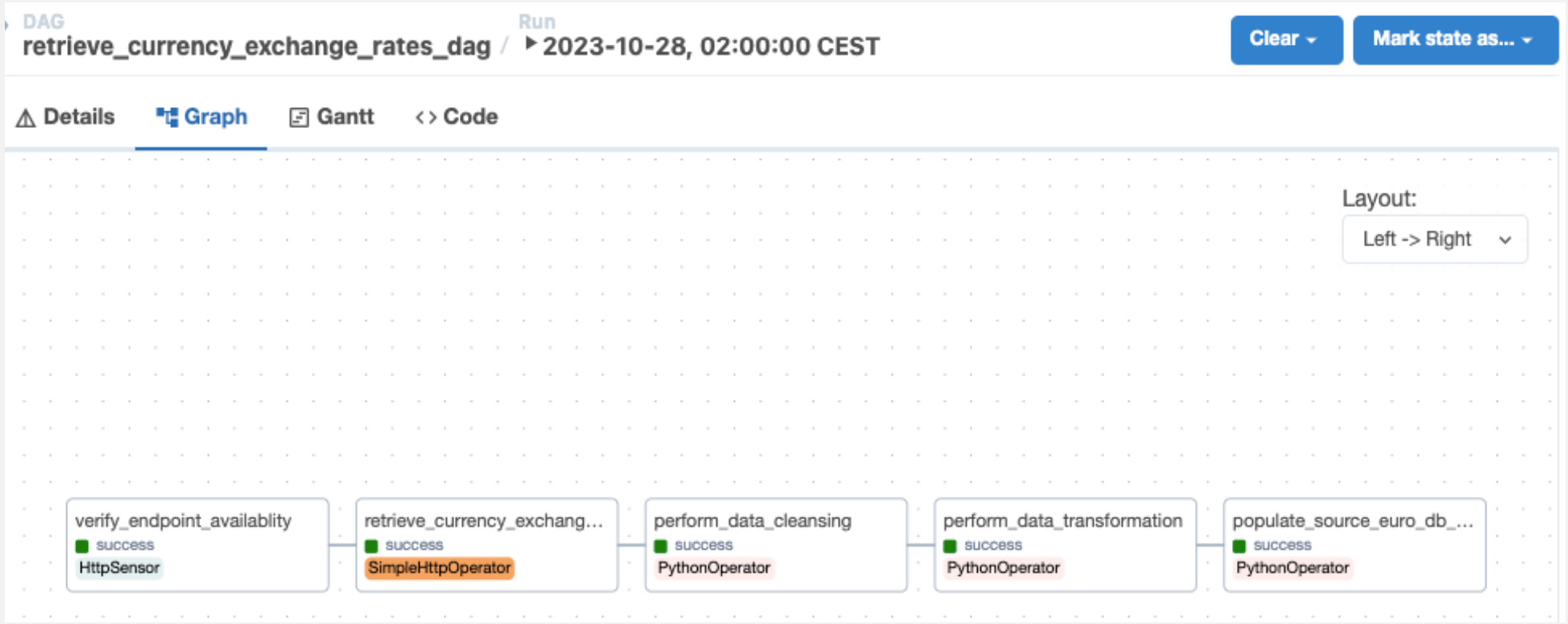
DAGS - GRAPHS

Structure of *retrieve_bank_holidays_dag*



DAGS - GRAPHS

Structure of *retrieve_currency_exchange_rates_dag*



SQLALCHEMY

1. Persistence of data to PostgreSQL database has been implemented based on a *PythonOperator* rather than a *PostgresOperator*.

2. SQLAlchemy ORM is the library of choice since it is natively supported by Airflow.

3. Implementation follows SQLAlchemy Declarative Mapping approach. The three classes, representing each database table are defined in a separate file namely *db_currency_exchange.py*.

4. No relationships between the three table classes were defined in the absence of foreign key constraints.

DATABASE VIEWS

EXCHANGE_RATES VIEW

	collection_dt	base	currency	rate	is_holiday
1	2023-10-28T00:02:31+00:00	EUR	USD	1.0565	NULL
2	2023-10-28T00:02:31+00:00	EUR	AED	3.8800	NULL
3	2023-10-28T00:02:31+00:00	EUR	AFN	78.6437	NULL
4	2023-10-28T00:02:31+00:00	EUR	ALL	105.7952	NULL
5	2023-10-28T00:02:31+00:00	EUR	AMD	424.6313	NULL
6	2023-10-28T00:02:31+00:00	EUR	ANG	1.8911	NULL
7	2023-10-28T00:02:31+00:00	EUR	AOA	879.6182	NULL
8	2023-10-28T00:02:31+00:00	EUR	ARS	369.7834	NULL
9	2023-10-28T00:02:31+00:00	EUR	AUD	1.6665	NULL
1…	2023-10-28T00:02:31+00:00	EUR	AWG	1.8911	NULL
1…	2023-10-28T00:02:31+00:00	EUR	AZN	1.7942	NULL
1…	2023-10-28T00:02:31+00:00	EUR	BAM	1.9558	NULL
1…	2023-10-28T00:02:31+00:00	EUR	BBD	2.1130	NULL
1…	2023-10-28T00:02:31+00:00	EUR	BDT	116.4626	NULL
1…	2023-10-28T00:02:31+00:00	EUR	BGN	1.9563	NULL
1…	2023-10-28T00:02:31+00:00	EUR	BHD	0.3972	NULL
1…	2023-10-28T00:02:31+00:00	EUR	BIF	2988.0583	NULL
1…	2023-10-28T00:02:31+00:00	EUR	BMD	1.0565	NULL
1…	2023-10-28T00:02:31+00:00	EUR	BND	1.4470	NULL
2…	2023-10-28T00:02:31+00:00	EUR	BOB	7.3082	NULL
2…	2023-10-28T00:02:31+00:00	EUR	BRL	5.2667	NULL
2…	2023-10-28T00:02:31+00:00	EUR	BSD	1.0565	NULL
2…	2023-10-28T00:02:31+00:00	EUR	BTN	88.0015	NULL
2…	2023-10-28T00:02:31+00:00	EUR	BWP	14.5722	NULL

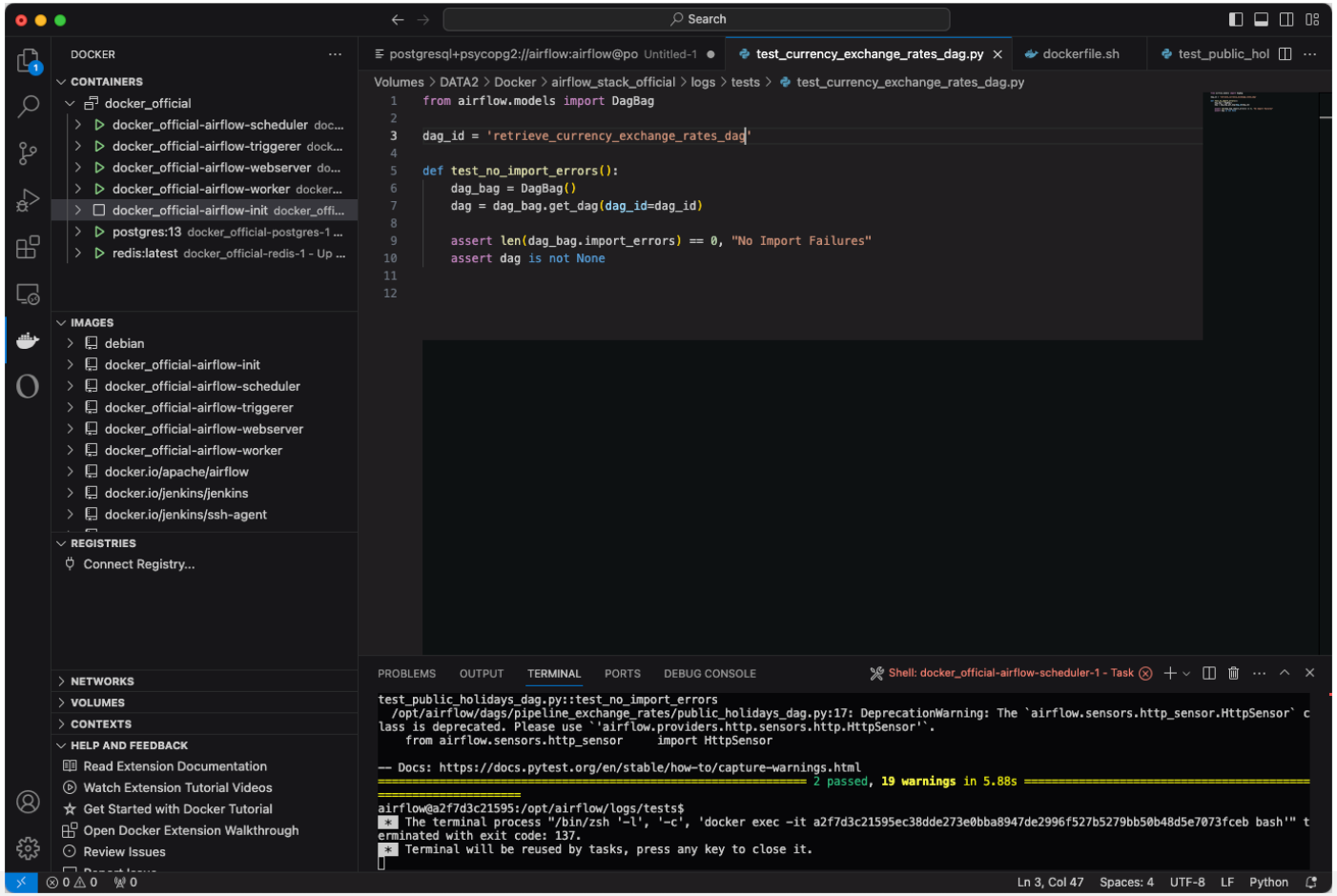
EXCHANGE_RATES VIEW

```
--Creation of VIEW exchange_rates
DROP VIEW IF EXISTS exchange_rates;
CREATE VIEW exchange_rates
AS
WITH cte_base AS
(
SELECT "source_data_EUR".id,
upload_dt,

(SELECT currency
FROM "source_data_EUR" as source_data_EUR_inner
WHERE source_data_EUR_inner.upload_dt = "source_data_EUR".upload_dt and rate=1) as base,

currency,rate,
"holiday_description" as is_holiday
FROM "source_data_EUR" LEFT JOIN "be_holidays" ON date(upload_dt) = "be_holidays".event_date
),
cte_collection_date
AS
(
SELECT id,
upload_dt,
to_timestamp(api_data->>'time_last_update_utc','DY, DD MON YYYY HH24:MI:SS TZH') as collection_dt
FROM "public"."source_data_USD"
)
SELECT cte_collection_date.collection_dt, cte_base.base, cte_base.currency,
cte_base.rate,cte_base.is_holiday
FROM cte_base LEFT JOIN cte_collection_date ON cte_base.upload_dt = cte_collection_date.upload_dt
```


UNIT TESTING

TESTING

1. Installation of pyTest library on airflow-webserver through *dockerfile* and *requirements.txt.*
2. Initiation of a terminal shell on airflow-webserver container via VS code IDE and docker extension.



TESTING

3. Despite having made of the pyTest unit testing framework, a single test was created which verifies the importation of libraries within a DAG.

DELIVERABLES

DELIVERABLES

1. Kindly refer to *deliverables.xlsx*.
2. Deliverables reside in a designated code repository hosted on GitHub.
3. URL to repo: https://github.com/tmpdeproject/assessment.git

FUTURE WORK ENHANCEMENTS

FUTURE WORK / ENHANCEMENTS

- The possibility to include frameworks such as Great Expections to ensure data quality.
- The development of a comprehensive set of unit tests aimed to not only check for errors in the actual DAG scripts but more importantly to verify the functionality and cater for edge-cases.
- Configuration of Jenkins and source code repository to support CI/CD.

THANK YOU