Step by Step documentation of the technical assessment:

Assumptions:

1. Docker and Docker compose is installed.

**Postgres Docker Installation:**

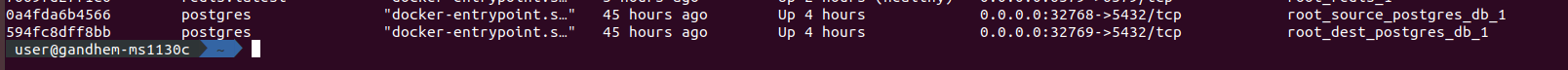
1. Create 2 Postgres Docker containers using docker-compose.yml to have source and destination database respectively

Git Repo:

https://github.com/jameshunt76/glints/tree/main/postgres\_docker\_compose

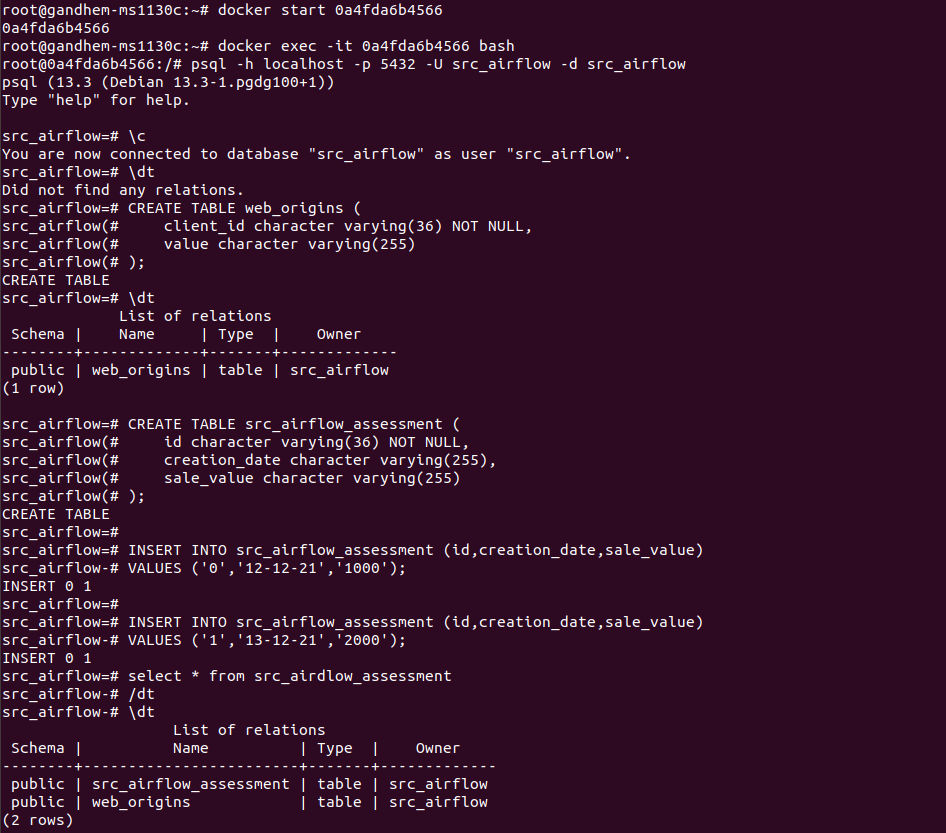
Source – root\_source\_postgres\_db\_1 (0a4fda6b4566)

Destination – root\_dest\_postgres\_db\_1 (594fc8dff8bb)

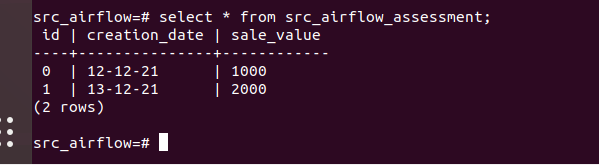


1. Create table in Source Postgres container and add data (Source – root\_source\_postgres\_db\_1)

* Start the container and exec it
* Login into postgres using the ID password created in docker-compose.yml file
* psql -h localhost -p 5432 -U src\_airflow -d src\_airflow
* Create table and insert 2 records

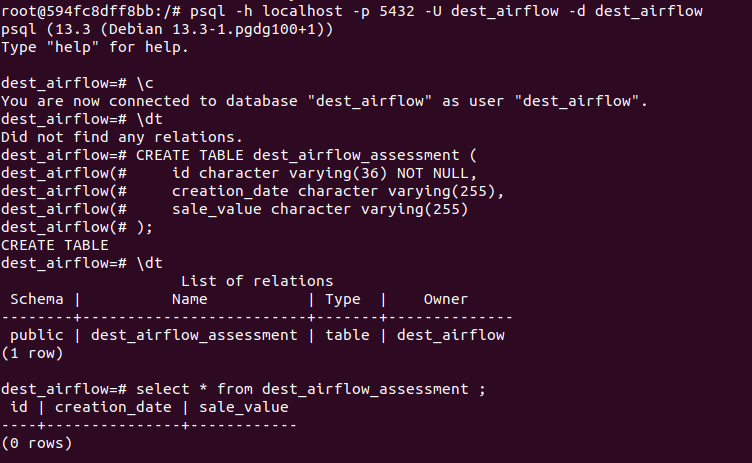


Below are the 2 records inserted



1. Create table in Destination Postgres container and add data (Destination – root\_dest\_postgres\_db\_1)

* Start the container and exec it
* Login into postgres using the ID password created in docker-compose.yml file
* psql -h localhost -p 5432 -U dest\_airflow -d dest\_airflow
* Create table (Must be Empty table)

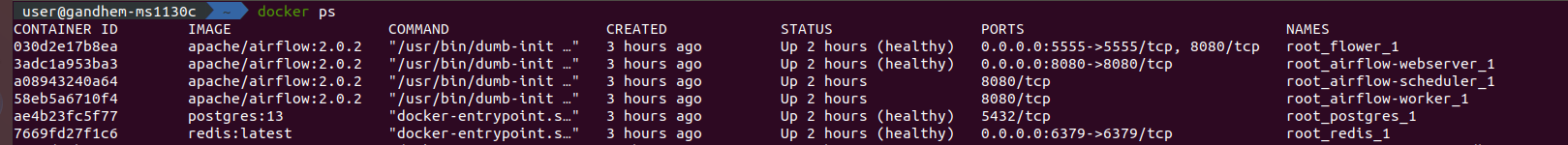


**Airflow Docker Installation:**

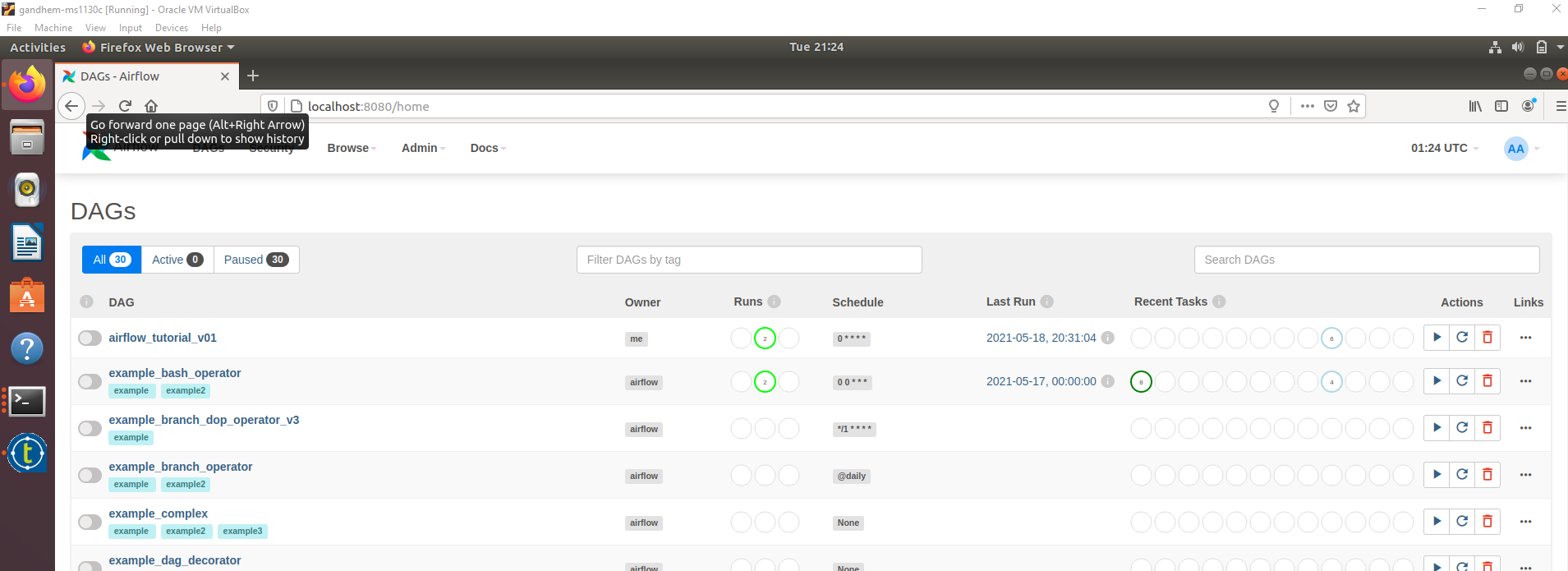
1. Install Airflow using the docker-compose.yml attached to the git repo

* <https://github.com/jameshunt76/glints/tree/main/airflow_docker_compose>

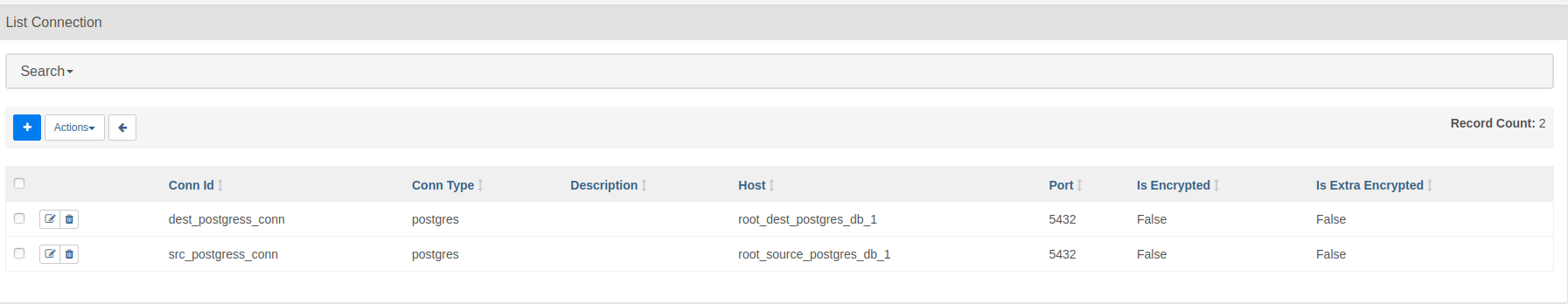
1. Run docker ps to make sure all the containers are up and healthy



1. Run docker-compose up to start the webserver and scheduler



1. Create source and destination connections for postgres



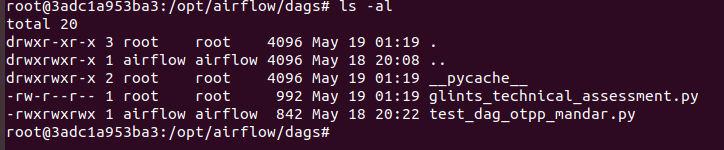
**Create a Dag File:**

1. Create the following file in Airflow container

Git Repo:

<https://github.com/jameshunt76/glints/tree/main/dags>

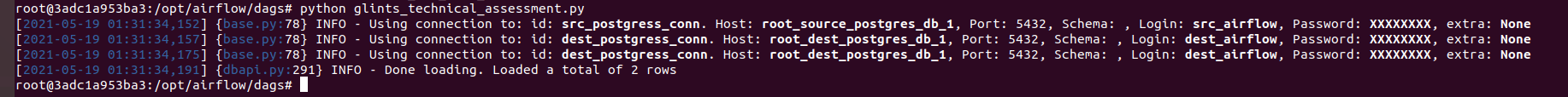
File Name: glints\_technical\_assessment.py

****

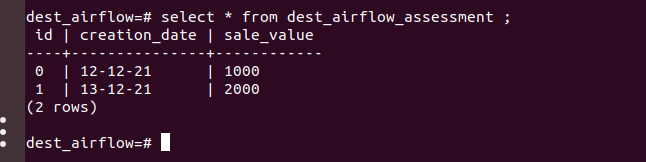
1. Verify the script with below command

Command:

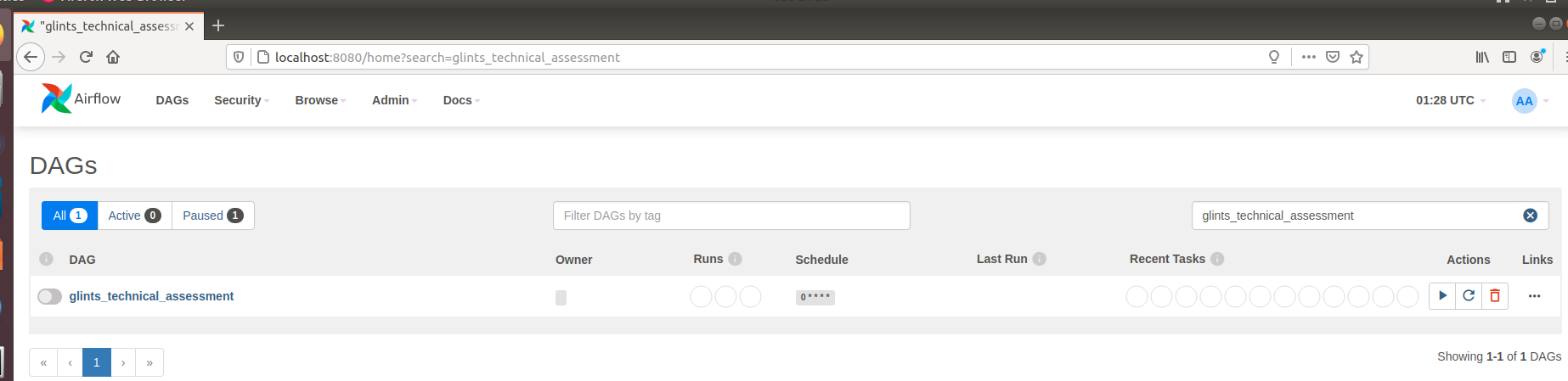
python glints\_technical\_assessment.py

****

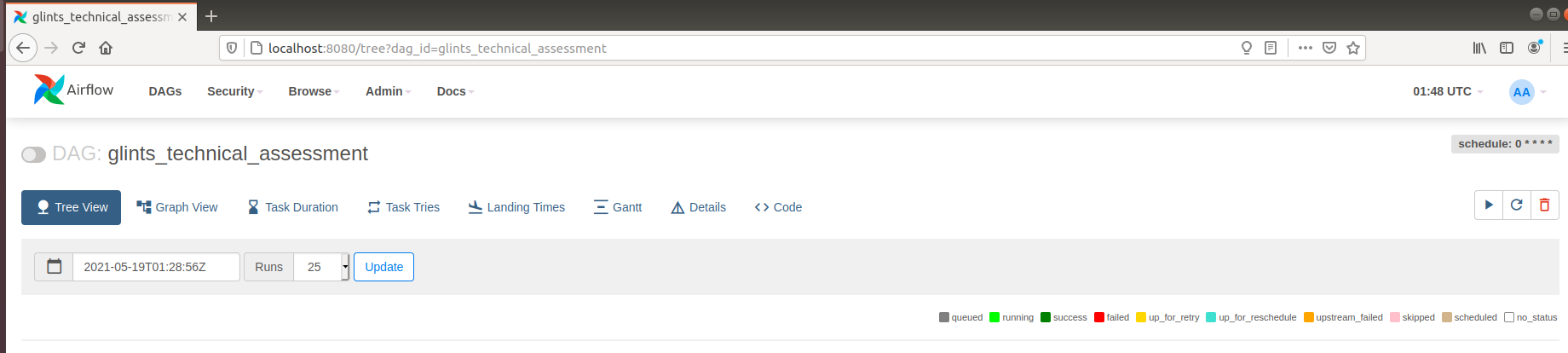
1. You will see the data like below in the destination. Truncate the table after verifying

****

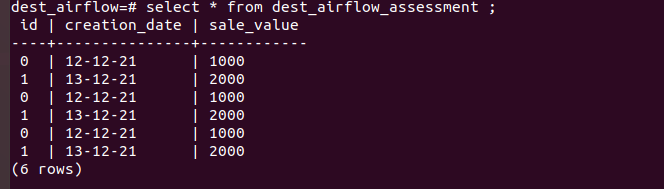
1. You must see the DAG on the Web UI



1. Run the DAG manually. Because of the schedule it keeps running and start inserting records.



1. Check the table again. Since the job is running recurring it keeps on adding record. (Need a fix 😊)



Note: The below were required to run the python script

pip install psycopg2-binary