

**Introduction to Docker**

Shweta Ajay Shinde

Masters in Data Analytics, San Jose State University

Data 226: Data warehousing

Instructor: Keeyong Han

18<sup>th</sup> Oct 2024

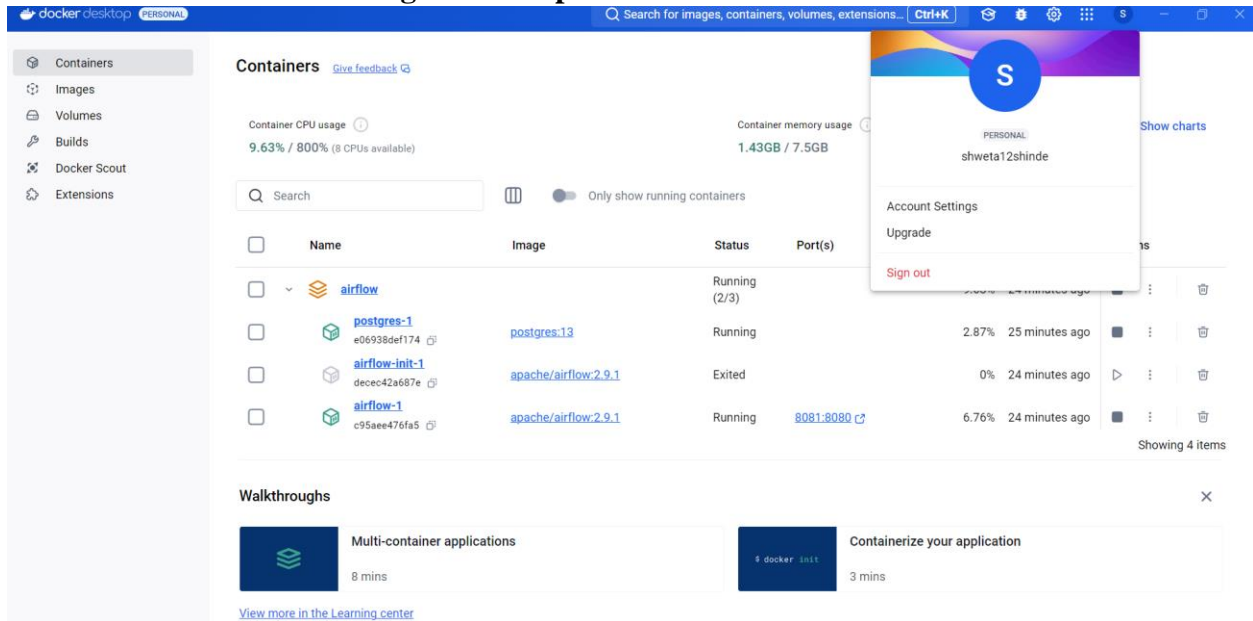
# Docker

## Introduction to Docker

### 1. Install Docker Engine in your laptop (1 pt)

- For Windows, refer to [Docker Setup for Windows](#)[Links to an external site.](#)  
[Minimize Video](#)
- Capture the screenshot of your Docker Desktop running (#1)

### Screenshot of Docker running on desktop.



### Screenshot of all necessary command executed on terminal

```
packages (3.10.0)
airflow-1 | Requirement already satisfied: pandas>=1.3.0 in /home/airflow/.local/lib/python3.12/site-packages (fro
m yfinance) (2.1.4)
airflow-1 | Requirement already satisfied: numpy>=1.16.5 in /home/airflow/.local/lib/python3.12/site-packages (fro
m yfinance) (1.26.4)
airflow-1 | Requirement already satisfied: requests>=2.31 in /home/airflow/.local/lib/python3.12/site-packages (fr
om yfinance) (2.31.0)
airflow-1 | Collecting multitasking>=0.0.7 (from yfinance)
airflow-1 |   Downloading multitasking-0.0.11-py3-none-any.whl.metadata (5.5 kB)
airflow-1 | Requirement already satisfied: lxml>=4.9.1 in /home/airflow/.local/lib/python3.12/site-packages (from
yfinance) (5.2.1)
airflow-1 | Requirement already satisfied: platformdirs>=2.0.0 in /home/airflow/.local/lib/python3.12/site-package
s (from yfinance) (4.2.1)
airflow-1 | Requirement already satisfied: pytz>=2022.5 in /home/airflow/.local/lib/python3.12/site-packages (from
yfinance) (2024.1)
airflow-1 | Collecting frozendict>=2.3.4 (from yfinance)
airflow-1 |   Downloading frozendict-2.4.6-py312-none-any.whl.metadata (23 kB)
airflow-1 | Collecting peewee>=3.16.2 (from yfinance)
airflow-1 |   Downloading peewee-3.17.7.tar.gz (939 kB)
airflow-1 | _____ 939.5/939.5 kB 5.1 MB/s eta 0:00:00
airflow-1 | Installing build dependencies: started
airflow-1 | Installing build dependencies: finished with status 'done'
airflow-1 | Getting requirements to build wheel: started
airflow-1 | Getting requirements to build wheel: finished with status 'done'
airflow-1 | Preparing metadata (pyproject.toml): started
airflow-1 | Preparing metadata (pyproject.toml): finished with status 'done'
airflow-1 | Requirement already satisfied: beautifulsoup4>=4.11.1 in /home/airflow/.local/lib/python3.12/site-pack
ages (from yfinance) (4.12.3)
airflow-1 | Collecting html5lib>=1.1 (from yfinance)
airflow-1 |   Downloading html5lib-1.1-py2.py3-none-any.whl.metadata (16 kB)
```

## Docker

### 2. Run Airflow via Docker Container (2 pt)

- Refer to [sjsu-data226 github docLinks to an external site.](#)
- Capture the screenshot of “docker ps” command (#2)

#### Screenshot of “docker ps” command

```
PS C:\Users\karti> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS              PORTS
c95aee476fa5   apache/airflow:2.9.1               "/usr/bin/dumb-init ..." 10 hours ago   Up 25 seconds (health: starting) 0.0.0.0
:8081->8080/tcp   airflow-airflow-1
e06938def174   postgres:13                         "docker-entrypoint.s..." 10 hours ago   Up 51 seconds (healthy)         5432/tcp
p               airflow-postgres-1
PS C:\Users\karti> vi dags/yfinance_to_snowflake.py
PS C:\Users\karti> docker exec -it c sh
(airflow)airflow dags list
dag_id          | fileloc                                | owners | is_paused
=====+=====+=====+=====
HelloWorld      | /opt/airflow/dags/helloWorld.py       | keeyong | True
YfinanceToSnowflake | /opt/airflow/dags/yfinance_to_snowflake.py | airflow | True
(airflow)
```

### 3. Run the YfinanceToSnowflake DAG in your Docker Container (3 pt)

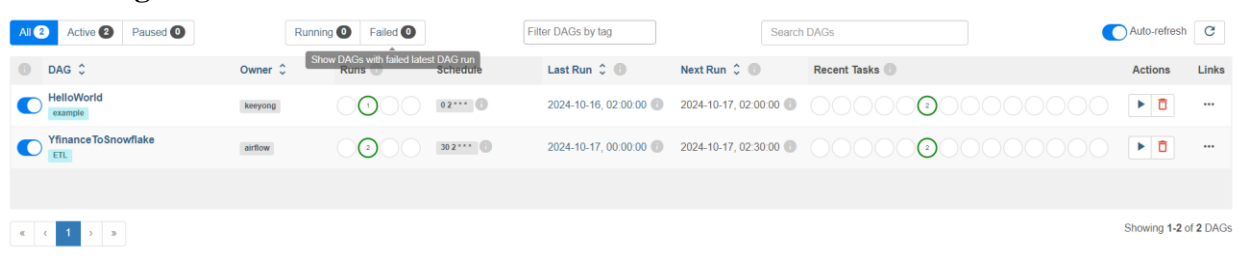
- Modify [the codeLinks to an external site.](#) according to your environment (including Airflow Connections)
- Add a link to your github repo to capture your DAG code.

Github link:-

[https://github.com/ShindeShwetaK/DW\\_Assignment/blob/main/yfinance\\_to\\_snowflake.py](https://github.com/ShindeShwetaK/DW_Assignment/blob/main/yfinance_to_snowflake.py)

- Capture a screenshot of your Airflow Web UI showing the DAG’s detailed page (#3)

#### DAGS Page



DAG	Owner	Runs	Last Run	Next Run	Recent Tasks	Actions	Links
HelloWorld example	keeyong	1	2024-10-16, 02:00:00	2024-10-17, 02:00:00	1	[Play] [Stop] [Refresh]	...
YfinanceToSnowflake ETL	airflow	2	2024-10-17, 00:00:00	2024-10-17, 02:30:00	2	[Play] [Stop] [Refresh]	...

Showing 1-2 of 2 DAGs

## Docker

### Connection Created

Edit Connection

Connection Id \*

snowflake\_conn

Connection Type \*

Snowflake

Connection Type missing? Make sure you've installed the correspond

Description

Schema

snowflake schema

Login

shweta12shinde

Password

snowflake password

Extra

{
"account": "mihzyak-aeb64943",
"warehouse": "compute\_wh",
"database": "dev",
"insecure\_mode": false
}

#### 4. Run your DAG from the command line (3 pt)

- Follow the instructions [hereLinks to an external site.](#) and capture screenshots

### Test YfinanceToSnowflake dags command execution

```
(airflow)airflow dags test YfinanceToSnowflake 2024-10-17
[2024-10-17T16:56:25.594+0000] {dagbag.py:545} INFO - Filling up the DagBag from /opt/airflow/dags
[2024-10-17T16:56:26.932+0000] {dag.py:4206} INFO - dagrun id: YfinanceToSnowflake
[2024-10-17T16:56:26.958+0000] {dag.py:4222} INFO - created dagrun <DagRun YfinanceToSnowflake @ 2024-10-17 00:00:00+00:00: manual__2024-10-17T00:00:00+00:00, state:running, queued_at: None. externally triggered: False>
[2024-10-17T16:56:26.984+0000] {dag.py:4168} INFO - [DAG TEST] starting task_id=extract map_index=-1
[2024-10-17T16:56:26.984+0000] {dag.py:4171} INFO - [DAG TEST] running task <TaskInstance: YfinanceToSnowflake.extract manual__2024-10-17T00:00:00+00:00 [scheduled]>
[2024-10-17 16:56:27,241] {taskinstance.py:2648} INFO - Exporting env vars: AIRFLOW_CTX_DAG_OWNER='airflow' AIRFLOW_CTX_DAG_ID='YfinanceToSnowflake' AIRFLOW_CTX_TASK_ID='extract' AIRFLOW_CTX_EXECUTION_DATE='2024-10-17T00:00:00+00:00' AIRFLOW_CTX_TRY_NUMBER='1' AIRFLOW_CTX_DAG_RUN_ID='manual__2024-10-17T00:00:00+00:00'
[2024-10-17T16:56:27.241+0000] {taskinstance.py:2648} INFO - Exporting env vars: AIRFLOW_CTX_DAG_OWNER='airflow' AIRFLOW_CTX_DAG_ID='YfinanceToSnowflake' AIRFLOW_CTX_TASK_ID='extract' AIRFLOW_CTX_EXECUTION_DATE='2024-10-17T00:00:00+00:00' AIRFLOW_CTX_TRY_NUMBER='1' AIRFLOW_CTX_DAG_RUN_ID='manual__2024-10-17T00:00:00+00:00'
[2024-10-17T16:56:27.243+0000] {taskinstance.py:430} INFO - ::endgroup::
[*****100%*****] 1 of 1 completed
[2024-10-17 16:56:27,771] {python.py:237} INFO - Done. Returned value was: {'Open': [233.44000244140625], 'High': [233.85000610351562], 'Low': [230.52999877929688], 'Close': [232.31500244140625], 'Adj Close': [232.31500244140625], 'Volume': [13513133]}
[2024-10-17T16:56:27.771+0000] {python.py:237} INFO - Done. Returned value was: {'Open': [233.44000244140625], 'High': [233.85000610351562], 'Low': [230.52999877929688], 'Close': [232.31500244140625], 'Adj Close': [232.31500244140625], 'Volume': [13513133]}
[2024-10-17T16:56:27.772+0000] {taskinstance.py:441} INFO - ::group::Post task execution logs
[2024-10-17T16:56:27.818+0000] {taskinstance.py:1206} INFO - Marking task as SUCCESS. dag_id=YfinanceToSnowflake, task_id=extract, run_id>manual__2024-10-17T00:00:00+00:00, execution_date=20241017T000000, start_date=, end_date=20241017T165627
[2024-10-17T16:56:27.829+0000] {dag.py:4182} INFO - [DAG TEST] end task task_id=extract map_index=-1
[2024-10-17T16:56:27.846+0000] {dag.py:4168} INFO - [DAG TEST] starting task_id=load map_index=-1
[2024-10-17T16:56:27.846+0000] {dag.py:4171} INFO - [DAG TEST] running task <TaskInstance: YfinanceToSnowflake.load manual__2024-10-17T00:00:00+00:00 [scheduled]>
```

Docker

DAG run successful in airflow UI

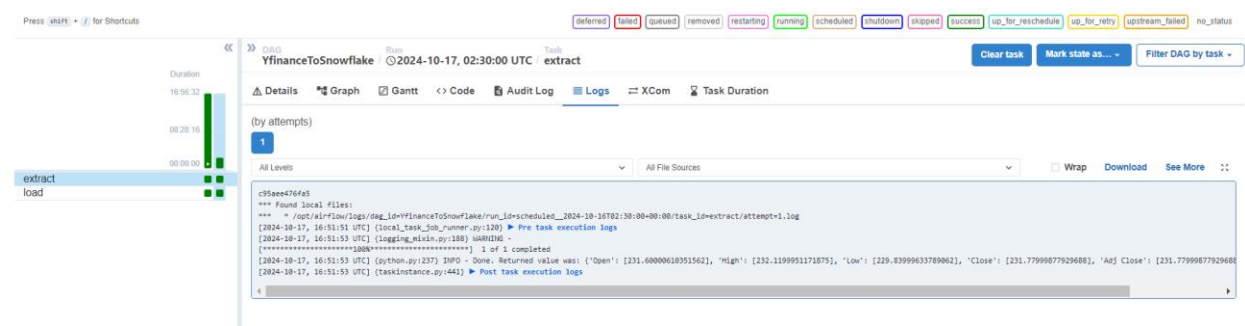


Table created in Snowflake

