Monitoring Data Collection Tool with Airflow & Docker

This guide provides a detailed walkthrough for deploying and operating a data collection tool using Apache Airflow within a Docker environment. It is intended for users with minimal prior knowledge of Docker, Airflow, or automation pipelines. The system is designed to automatically run Python scripts that generate outputs (e.g., CSVs) and upload them to SharePoint.

# 1. Prerequisites

Ensure the following are installed on your system:

* Docker: <https://docs.docker.com/get-docker/>
* Docker Compose: <https://docs.docker.com/compose/install/>
* Git (optional but recommended): <https://git-scm.com/downloads>
* A SharePoint account with appropriate folder permissions
* VS Code (Optional, but recommended): <https://code.visualstudio.com/>

## Verifying Docker Installation

Open a terminal and run:

docker --version  
docker compose version

If you encounter a 'docker command not found' error, restart your computer and verify Docker Desktop is running

# 2. Project Structure Overview

Upon cloning or unzipping the project, the directory should look like this:

monitoring\_data\_collection\_tool/  
│  
├── Dockerfile  
├── docker-compose.yaml  
├── dags/  
│ └── monitoring\_tool\_dag.py  
├── scripts/  
│ └── script1.py (and others)  
├── documentation/  
│ ├── output\_dir.txt  
│ └── logs/  
│ └── workflow\_log.txt  
├── outputs/  
└── run\_tool.py

# 3. Creating the Dockerfile

The Dockerfile defines your environment:

FROM apache/airflow:2.8.1-python3.10  
  
USER root  
RUN apt-get update && apt-get install -y gcc python3-dev libffi-dev  
  
COPY requirements.txt /requirements.txt  
RUN pip install --no-cache-dir -r /requirements.txt  
  
USER airflow  
COPY monitoring\_data\_collection\_tool /opt/airflow/monitoring\_data\_collection\_tool  
COPY dags /opt/airflow/dags

To build the Docker image:

docker build -t monitoring-tool .

# 4. Writing requirements.txt

Ensure all required packages are listed:

pandas  
requests  
office365-rest-python-client

You may add others based on your scripts.

# 5. Configuring docker-compose.yaml

version: '3.7'  
  
services:  
 postgres:  
 image: postgres:13  
 environment:  
 POSTGRES\_USER: airflow  
 POSTGRES\_PASSWORD: airflow  
 POSTGRES\_DB: airflow  
 volumes:  
 - postgres-db-volume:/var/lib/postgresql/data  
  
 redis:  
 image: redis:latest  
  
 airflow-webserver:  
 image: apache/airflow:2.8.1-python3.10  
 depends\_on:  
 - postgres  
 - redis  
 environment:  
 AIRFLOW\_\_CORE\_\_EXECUTOR: LocalExecutor  
 AIRFLOW\_\_CORE\_\_SQL\_ALCHEMY\_CONN: postgresql+psycopg2://airflow:airflow@postgres/airflow  
 ports:  
 - "8080:8080"  
 volumes:  
 - ./dags:/opt/airflow/dags  
 - ./monitoring\_data\_collection\_tool:/opt/airflow/monitoring\_data\_collection\_tool  
 command: webserver  
  
 airflow-scheduler:  
 image: apache/airflow:2.8.1-python3.10  
 depends\_on:  
 - airflow-webserver  
 volumes:  
 - ./dags:/opt/airflow/dags  
 - ./monitoring\_data\_collection\_tool:/opt/airflow/monitoring\_data\_collection\_tool  
 command: scheduler  
  
volumes:  
 postgres-db-volume:

To bring up the service:

docker-compose up –build

# 6. Initialising and Running Airflow

In terminal:

docker compose up airflow-init  
docker compose up

Visit http://localhost:8080. Default login: `admin / admin`

# 6. Airflow DAG

The DAG should be saved in `dags/monitoring\_tool\_dag.py`. It defines the task to run `run\_tool.py`.

from airflow import DAG  
from airflow.operators.bash import BashOperator  
from datetime import datetime  
  
with DAG(  
 "monitoring\_data\_collection\_tool",  
 start\_date=datetime(2025, 1, 1),  
 schedule\_interval="@daily",  
 catchup=False,  
) as dag:  
 run\_tool = BashOperator(  
 task\_id="run\_tool",  
 bash\_command="python /opt/airflow/monitoring\_data\_collection\_tool/run\_tool.py"  
 )

# 7. Running and Validating

To check the container is running:

docker ps

To open the Airflow web UI, navigate to http://localhost:8080

To trigger the DAG manually:

docker exec -it <container\_id> airflow dags trigger monitoring\_data\_collection\_tool

To view logs:

docker exec -it <container\_id> tail -f /opt/airflow/monitoring\_data\_collection\_tool/documentation/logs/workflow\_log.txt

# 8. Storing SharePoint Credentials in Airflow

Use the Airflow UI to add a connection named `sharepoint\_conn`.

1. Navigate to Admin > Connections.  
2. Add a new connection:  
 - Conn Id: `sharepoint\_conn`  
 - Conn Type: `Generic`  
 - Extra JSON:

{  
 "username": "your@domain.com",  
 "password": "your\_password"  
 }

Use `BaseHook.get\_connection("sharepoint\_conn")` in code.

# 9. Troubleshooting

* Airflow not starting: ensure ports aren’t blocked and Docker Compose is configured properly.
* No DAGs visible: ensure DAG file is mounted correctly and has no syntax errors.
* Credentials error: ensure the SharePoint credentials are correctly stored in Airflow Connections.
* Script not producing output: check `outputs/` folder and the log file for exceptions.

# 10. References

* Docker Docs: <https://docs.docker.com/>
* Apache Airflow Docs: <https://airflow.apache.org/docs/apache-airflow/stable/>
* SharePoint Python SDK: <https://github.com/vgrem/Office365-REST-Python-Client>
* Airflow Connections Guide: <https://airflow.apache.org/docs/apache-airflow/stable/howto/connection/index.html>