

MDS For Quality Pipelines

Outline

SetUp Process	1
Mac User Implementation	1 - 10
Windows User Implementation	11 - 19
Join Orbit Learning Community	19

Setup Process

For you to be able to complete this setup from beginning to the end, you should have these prerequisites:

- Docker Desktop
- Python
- Git
- Visual Studio

For Mac Users:

Checking if prerequisites have been properly set up:

For Docker desktop:

Go to applications, start the docker desktop and ensure that the docker engine has started

Open a terminal and run these commands:

- `docker --version` # to check the version of docker
- `docker run hello-world` # this should run without error

For Python:

Run this command “`python --version`” or “`python3 --version`” # to check the python version from your terminal

For Git:

Run this command “`git --version`” # to check the git version from your terminal

For Visual Studio:

Open the application and check if it opens without an error.

Setting up the MDS

Local SetUp

In this section, we will try to install dbt and great_expectations library locally.

Before you start this phase, check that you can access this url:

<https://raw.githubusercontent.com/apache/airflow/constraints-2.8.4/constraints-3.10.txt>

If you can't access the url above, please change your network.

Local setup for dbt and great_expectations (You can skip this phase if you have done it before)

Open a new terminal and run these commands:

```
python -m venv mds_venv # this creates a virtual environment called mds_venv
```

```
cd mds_venv # this command switch directory to the virtual environment you just created
```

```
source bin/activate # this activates your virtual environment
```

```
pip install dbt-postgres or pip3 install dbt-postgres
```

```
pip install dbt-bigquery or pip3 install dbt-bigquery
```

```
pip install dbt-snowflake or pip3 install dbt-snowflake
```

```
pip install great_expectations or pip3 install great_expectations
```

Initializing a dbt project

Before we do this, let's create a postgres source in docker

On the same terminal you are, run the command below:

```
docker run -d -p5433:5432 -e POSTGRES_PASSWORD=password1234 postgres:15.6
```

When the command above is finished running, run this command:

```
dbt init dbt_test # this command helps you to initialize a dbt project called dbt_test
```

It will request for these credentials:

Source: select the postgres option

Hostname: localhost

Port: 5433

Username: postgres

Password: password1234 (please note that the password won't be visible as you are typing it)

Dbname: postgres

Schema: public

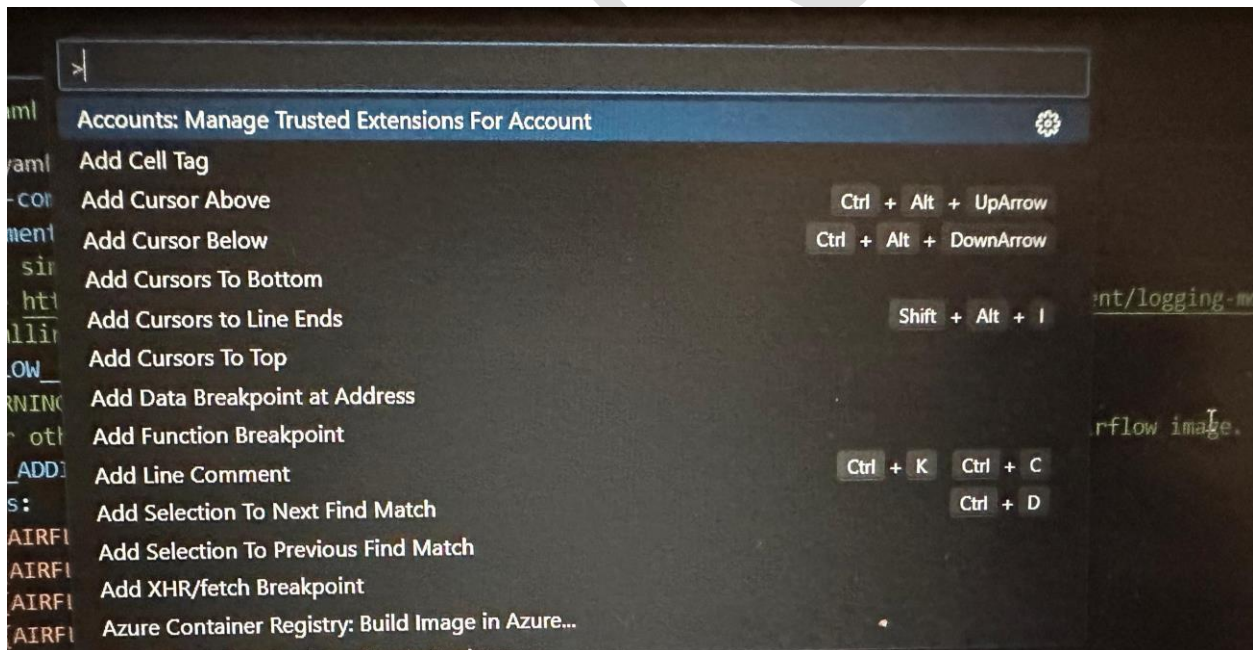
Threads: 1

After you have inputted all these credentials, an output will be displayed on your terminal. This will include the path to the profiles.yaml file (the path should look like this /Users/User/.dbt/profiles.yaml), please copy that path and save somewhere on notepad.

Docker setup for the MDS

Let's first establish the enabling of visual studio code (VS code) to be able to open from your terminal. To do this, open your VS code, and ensure that you have select that you trust the author.

Hold command+shift+p key on your keyboard, you should see something like this come up at the top middle of your VS code:



Scroll down on the options until you see something like this, and then select:

Shell Command: Install 'code' command in PATH

Once you have done this, close your VS code.

Open a new terminal and run this command:

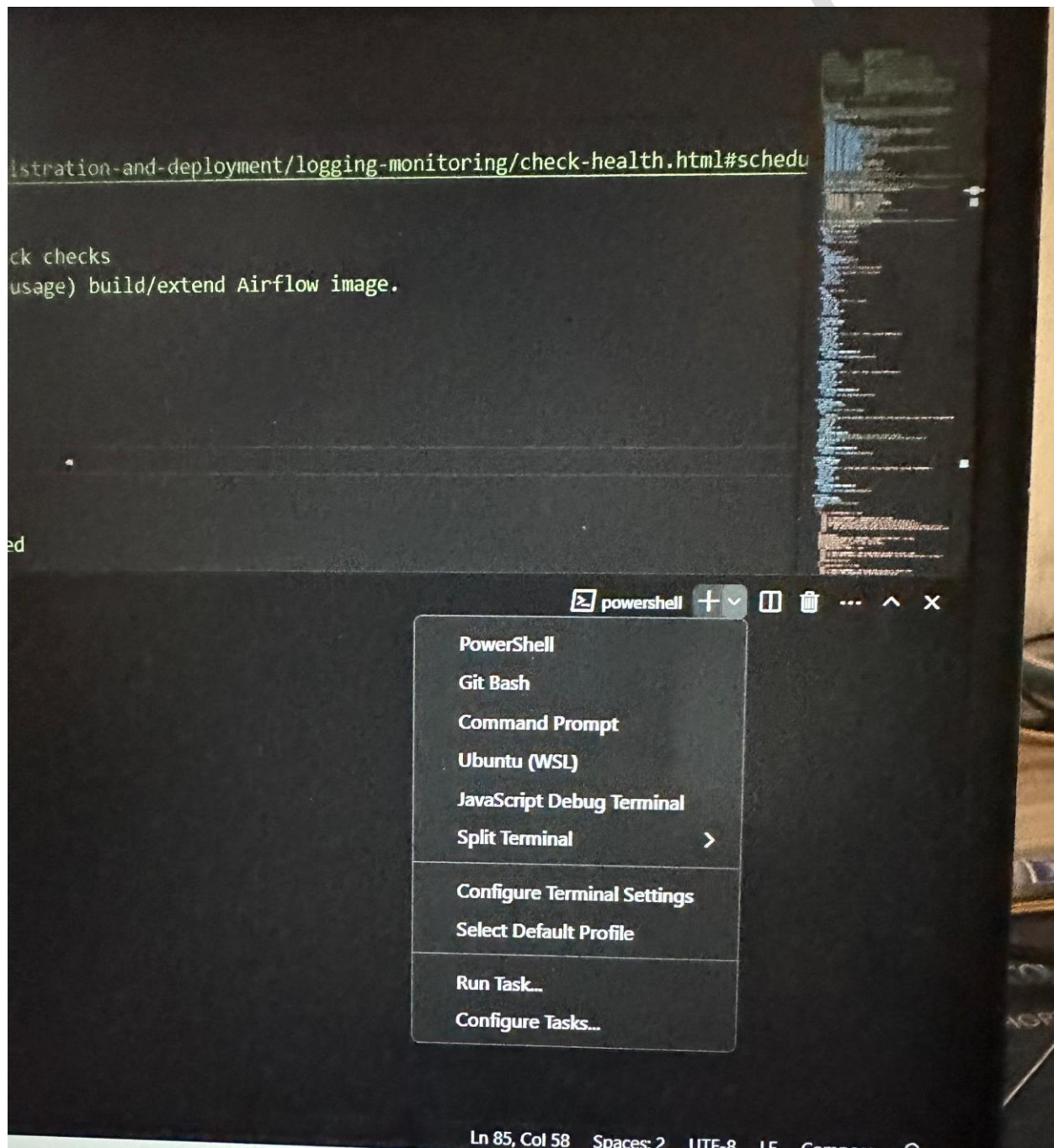
`sudo git config --system core.longpaths true` # It should require you to put your password

Once the command above is done, run this:

`code .` # the full stop there is not an error, include it. It should open a VS code for you

In VS code, open a new terminal (you can do this by taking your mouse to the top of the VS code application, it should show the option to create a new terminal)

Once this is done, you will have to open a new bash terminal. Refer to the image below:



When you click on the dropdown beside the + icon, you should see option for Git Bash. Once this terminal is opened, run the commands below:

git clone --depth=1 <https://github.com/airbytehq/airbyte.git> #(This might not work if you already have a directory called airbyte, try renaming the one you have before rerunning it)

Once the command above is done, run this:

```
cd airbyte
```

```
./run-ab-platform.sh
```

The command above should take some time depending on your internet speed. You can track the process on the terminal you ran the command on VS code. You can open another Git bash terminal to know if the required services have started running for airbyte, you should see these services running:

\$ docker ps	CONTAINER ID	IMAGE	COMMAND	NAMES	CREATED	STATUS	PORTS
395d9b66439f	airbyte/proxy:0.51.0	0-8001/tcp, 0.0.0.0:8003->8003/tcp, 80/tcp, 0.0.0.0:8006->8006/tcp	"/run.sh ./run.sh"	airbyte-proxy	44 hours ago	Up 20 seconds	0.0.0.0:8000-8001->800
a469fc7e0862	airbyte/worker:0.51.0	5->9000/tcp	"/bin/bash -c airbyt..."	airbyte-worker	44 hours ago	Up 23 seconds	5005/tcp, 0.0.0.0:5806
1f54643c81e9	airbyte/server:0.51.0	0.0.0:58064->8001/tcp	"/bin/bash -c airbyt..."	airbyte-server	44 hours ago	Up 23 seconds	5005/tcp, 8000/tcp, 0.
4f7ddb439bc3	airbyte/cron:0.51.0		"/bin/bash -c airbyt..."	airbyte-cron	44 hours ago	Up 23 seconds	
958a01ec562b	airbyte/connector-builder-server:0.51.0		"/bin/bash -c airbyt..."	airbyte-connector-builder-server	44 hours ago	Up 23 seconds	0.0.0.0:58062->80/tcp
ec4e3a262f85	airbyte/webapp:0.51.0		"/docker-entrypoint..."	airbyte-webapp	44 hours ago	Up 24 seconds	80/tcp
8562fba3e9a3->8006/tcp	airbyte/airbyte-api-server:0.51.0		"/bin/bash -c airbyt..."	airbyte-api-server	44 hours ago	Up 23 seconds	5005/tcp, 0.0.0.0:5806
9cb035db391f	airbyte/temporal:0.51.0	p, 7233-7235/tcp, 7239/tcp	"/update-and-start-..."	airbyte-temporal	44 hours ago	Up About a minute	6933-6935/tcp, 6939/tc
a00881b69a9a	airbyte/db:0.51.0		"docker-entrypoint.s..."	airbyte-db	44 hours ago	Up About a minute	5432/tcp

You should now be able to access airbyte from your browser using this url: localhost:8000

Username: airbyte

Password: password

Airflow SetUp (In conjunction with dbt and great_expectations)

While airbyte is trying to setup, you might want to proceed to doing airflow setup at the same time.

Open a new terminal

Create a new directory by running this command:

mkdir airflow # (this will throw up an error if you have done it before), so proceed to next command

cd airflow

code . (this will open another VS code window)

Once VS code is opened, open a new terminal (Git Bash terminal) and run this command:

Please if you have a docker compose.yaml file in this directory before, kindly delete that file before running the command below.

`curl -L -O 'https://airflow.apache.org/docs/apache-airflow/2.8.4/docker-compose.yaml'` # This command will download a docker-compose.yaml file into the airflow directory

Now, let's proceed to edit the file, click on the file from within VS Code, and scroll down to line 53:

```
52 image: ${AIRFLOW_IMAGE_NAME:-apache/airflow:2.8.4}
53 # build: .
54 environment:
```

Remove the comment (#) from that line and should look like this:

```
52 image: ${AIRFLOW_IMAGE_NAME:-apache/airflow:2.8.4}
53 build: .
54 environment:
```

scroll down to line 76 where you have the below:

```
72 volumes:
73   - ${AIRFLOW_PROJ_DIR:-.}/dags:/opt/airflow/dags
74   - ${AIRFLOW_PROJ_DIR:-.}/logs:/opt/airflow/logs
75   - ${AIRFLOW_PROJ_DIR:-.}/config:/opt/airflow/config
76   - ${AIRFLOW_PROJ_DIR:-.}/plugins:/opt/airflow/plugins
```

Edit this part to include these parts:

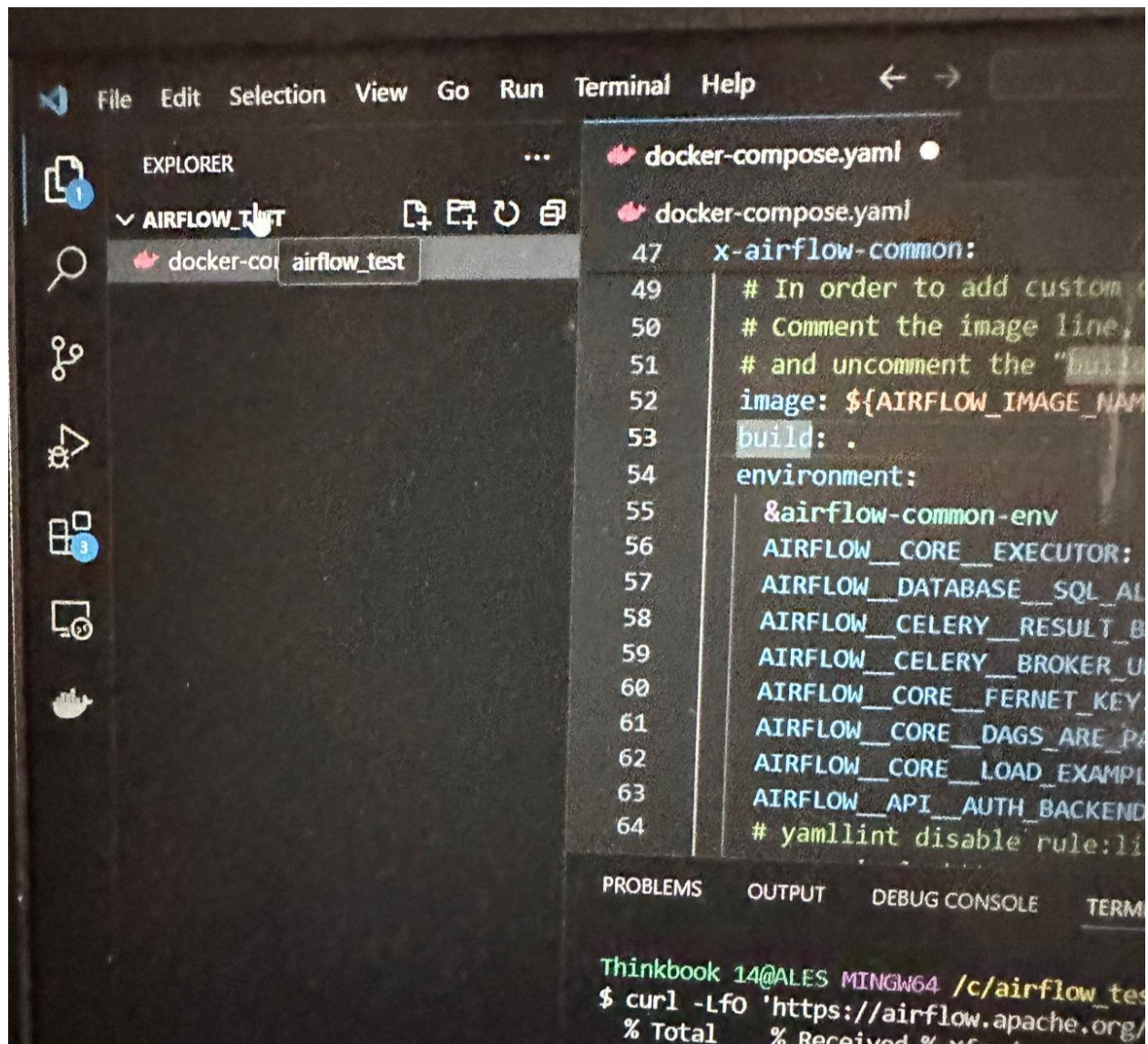
```
- ${AIRFLOW_PROJ_DIR:-.}/dbt:/opt/airflow/dbt
- ${AIRFLOW_PROJ_DIR:-.}/ge:/opt/airflow/ge
- "<to be modified>/.dbt:/home/airflow/.dbt" # to be modified
```

The last line above should be modified to include to the path of your dbt profile as you have noted on your notepad, replace the <to be modified> with /Users/User (i.e the part before the /.dbt as you noted on your notepad, it should look like this:

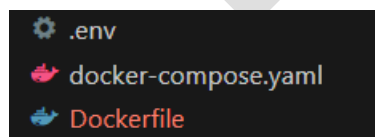
```
72 volumes:
73   - ${AIRFLOW_PROJ_DIR:-.}/dags:/opt/airflow/dags
74   - ${AIRFLOW_PROJ_DIR:-.}/logs:/opt/airflow/logs
75   - ${AIRFLOW_PROJ_DIR:-.}/config:/opt/airflow/config
76   - ${AIRFLOW_PROJ_DIR:-.}/plugins:/opt/airflow/plugins
77   - ${AIRFLOW_PROJ_DIR:-.}/dbt:/opt/airflow/dbt
78   - ${AIRFLOW_PROJ_DIR:-.}/ge:/opt/airflow/ge
79   - "/Users/User/.dbt:/home/airflow/.dbt" # to be modified
```

After all these are done, save the file.

Next, let's create additional files (.env & Dockerfile) in the airflow directory (Don't do this step if you have the files before).



As you can see in the image above, hover your mouse on the airflow directory, there are four icons to the right, the first icon is to create a new file, click on it and create a file named .env and another file named Dockerfile. You should have something like this:



Click on the .env file and add this text and save:

```
AIRFLOW_UID=50000
```

Click on the Dockerfile and add this text and save:

```
FROM apache/airflow:2.8.4-python3.10
#ENV AIRFLOW__CORE__LOAD_EXAMPLES=True
#ENV AIRFLOW__DATABASE__SQL_ALCHEMY_CONN=my_conn_string
#USER root
#RUN apt-get update \
##    && sudo apt update && sudo apt upgrade && sudo apt-get install python3.9 \
#    && apt-get install -y --no-install-recommends \
#        vim
#ADD requirements3.10.txt .
USER root
RUN apt update
RUN apt install git -y
RUN apt update -y && sudo apt install -y build-essential libpq-dev
RUN apt-get update \
    && apt-get install -y --no-install-recommends \
        gcc \
        heimdal-dev \
    && apt-get autoremove -yqq --purge \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/*
#RUN apt-get install vim -y
USER airflow
#RUN pip install apache-airflow==${AIRFLOW_VERSION} -r requirements3.txt
#RUN pip uninstall -y argparse
#ENV AIRFLOW_VERSION=2.8.4
#ENV PYTHON_VERSION="$(python --version | cut -d ' ' -f 2 | cut -d '.' -f 1-2)"
ENV wsl_ip="$(cat /etc/resolv.conf | grep nameserver | cut -d ' ' -f 2)"
ENV PYTHON_VERSION=3.10
ENV CONSTRAINT_URL="https://raw.githubusercontent.com/apache/airflow/constraints-
${AIRFLOW_VERSION}/constraints-${PYTHON_VERSION}.txt"
RUN pip install "apache-
airflow[async,postgres,google,cnfc.kubernetes,snowflake,airbyte]==${AIRFLOW_VERSI
ON}" --constraint "${CONSTRAINT_URL}"
#RUN pip install "apache-
airflow[async,postgres,google,cnfc.kubernetes,snowflake,airbyte]==${AIRFLOW_VERSI
ON}" -r requirements2.txt
RUN pip uninstall -y argparse
RUN pip install dbt-core
RUN pip install dbt-postgres
RUN pip install dbt-redshift
RUN pip install dbt-snowflake
RUN pip install dbt-bigquery
RUN pip install great_expectations
```



```









RUN pip install airflow-provider-great-expectations
RUN pip install astronomer-cosmos[dbt.all]

USER root
RUN mkdir -p -m 777 /opt/airflow/ge
RUN mkdir -p -m 777 /opt/airflow/dbt
RUN mkdir -p -m 777 /opt/airflow/dbt/logs
RUN mkdir -p -m 777 /opt/airflow/dbt/build
RUN mkdir -p -m 777 /opt/airflow/logs
RUN mkdir -p -m 777 /opt/airflow/dags
RUN mkdir -p -m 777 /opt/airflow/plugins
RUN mkdir -p -m 777 /opt/airflow/logs/scheduler
RUN mkdir -p -m 777 /dbt
#RUN mkdir -p -m 777 /opt/airflow/.dbt
RUN chmod a+x /usr/bin/git
#COPY .dbt /opt/airflow/dbt
#COPY .dbt/profiles.yml /home/airflow/.dbt/profiles.yml

```

Once this is done, go to Git Bash terminal you opened within the VS code and run this command:

Note that if you have deployed airflow once before, you have to stop and delete the containers under airflow in your Docker desktop as shown below before running the next command:

<input type="checkbox"/>		airflow	-	Running (6/7)			
<input type="checkbox"/>		airflow-worker-1 20ef2b8c4da4	apache/airflow:2.8.4	Running	4 minutes ago		
<input type="checkbox"/>		airflow-triggerer-1 c224b4f38fb1	apache/airflow:2.8.4	Running	4 minutes ago		
<input type="checkbox"/>		airflow-webserver-1 f4e1567bb59b	apache/airflow:2.8.4	Running	8080:8080		
<input type="checkbox"/>		airflow-scheduler-1 12f748381d2e	apache/airflow:2.8.4	Running	4 minutes ago		
<input type="checkbox"/>		airflow-init-1 b21cb1e49927	apache/airflow:2.8.4	Exited			
<input type="checkbox"/>		postgres-1 67159a525287	postgres:13	Running	4 minutes ago		
<input type="checkbox"/>		redis-1 ah5738205c02	redis:latest	Running	4 minutes ago		

Now run the command below

`docker compose up --build`

This will start pulling the necessary images and build them into containers and start the necessary airflow services.

If everything works fine, you should see these services when you run `docker ps`:

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
20ef2b8c4da4	apache/airflow:2.8.4	"/usr/bin/dumb-init ..."	7 days ago	Up 4 minutes (healthy)	8080/tcp	airflow-airfl
ow-worker-1						
c224b4f38fb1	apache/airflow:2.8.4	"/usr/bin/dumb-init ..."	7 days ago	Up 4 minutes (healthy)	8080/tcp	airflow-airfl
ow-triggerer-1						
f4e1567bb59b	apache/airflow:2.8.4	"/usr/bin/dumb-init ..."	7 days ago	Up 4 minutes (healthy)	0.0.0.0:8080->8080/tcp	airflow-airfl
ow-webserver-1						
12f748381d2e	apache/airflow:2.8.4	"/usr/bin/dumb-init ..."	7 days ago	Up 4 minutes (healthy)	8080/tcp	airflow-airfl
ow-scheduler-1						
67159a525287	postgres:13	"docker-entrypoint.s..."	7 days ago	Up 4 minutes (healthy)	5432/tcp	airflow-postg
res-1						
ab5738205c02	redis:latest	"docker-entrypoint.s..."	7 days ago	Up 4 minutes (healthy)	6379/tcp	airflow-redis
-1						

It should show status healthy for everything

You should now be able to access airflow from your browser by opening this url: localhost:8080

Username: airflow

Password: airflow

Stopping the services:

For airbyte

Open a new terminal

Run these commands:

```
cd airbyte
```

```
docker compose stop # This will stop all the containers
```

For airflow

Open a new terminal

Run these commands:

```
cd airflow
```

```
docker compose stop # This will stop all the containers
```

For Windows Users:

Checking if prerequisites have been properly set up:

For Docker desktop:

Go to applications, start the docker desktop and ensure that the docker engine has started

Open a cmd and run these commands:

- `docker --version` # to check the version of docker
- `docker run hello-world` # this should run without error

For Python:

Run this command “`python --version`” or “`python3 --version`” # to check the python version from your cmd

For Git:

Run this command “`git --version`” # to check the git version from your cmd

For Visual Studio:

Open the application and check if it opens without an error.

Setting up the MDS

Local SetUp

In this section, we will try to install dbt and great_expectations library locally.

Before you start this phase, check that you can access this url:

<https://raw.githubusercontent.com/apache/airflow/constraints-2.8.4/constraints-3.10.txt>

If you can't access the url above, please change your network.

Local setup for dbt and great_expectations (You can skip this phase if you have done it before)

Open a new cmd and run these commands:

```
cd c:\
```

```
python -m venv mds_venv # this creates a virtual environment called mds_venv
```

```
cd mds_venv # this command switch directory to the virtual environment you just created
```

```
scripts\activate # this activates your virtual environment
```

Rerun the commands below if it breaks cos of internet connection

pip install dbt-postgres or pip3 install dbt-postgres
pip install dbt-bigquery or pip3 install dbt-bigquery
pip install dbt-snowflake or pip3 install dbt-snowflake
pip install great_expectations or pip3 install great_expectations

Initializing a dbt project

Before we do this, let's create a postgres source in docker

On the same cmd you are, run the command below:

```
docker run -d -p5433:5432 -e POSTGRES_PASSWORD=password1234 postgres:15.6
```

When the command above is finished running, run this command:

```
dbt init dbt_test # this command helps you to initialize a dbt project called dbt_test
```

It will request for these credentials:

Source: select the postgres option

Hostname: localhost

Port: 5433

Username: postgres

Password: password1234 (please note that the password won't be visible as you are typing it)

Dbname: postgres

Schema: public

Threads: 1

After you have inputted all these credentials, an output will be displayed on your terminal. This will include the path to the profiles.yaml file (the path should look like this C:\Users\Ability\.dbt\profiles.yaml), please copy that path and save somewhere on notepad.

Docker setup for the MDS

Open a new cmd as an administrator and run this command:

```
git config --system core.longpaths true
```

Once the command above is done, close the cmd

Open another cmd without administrator

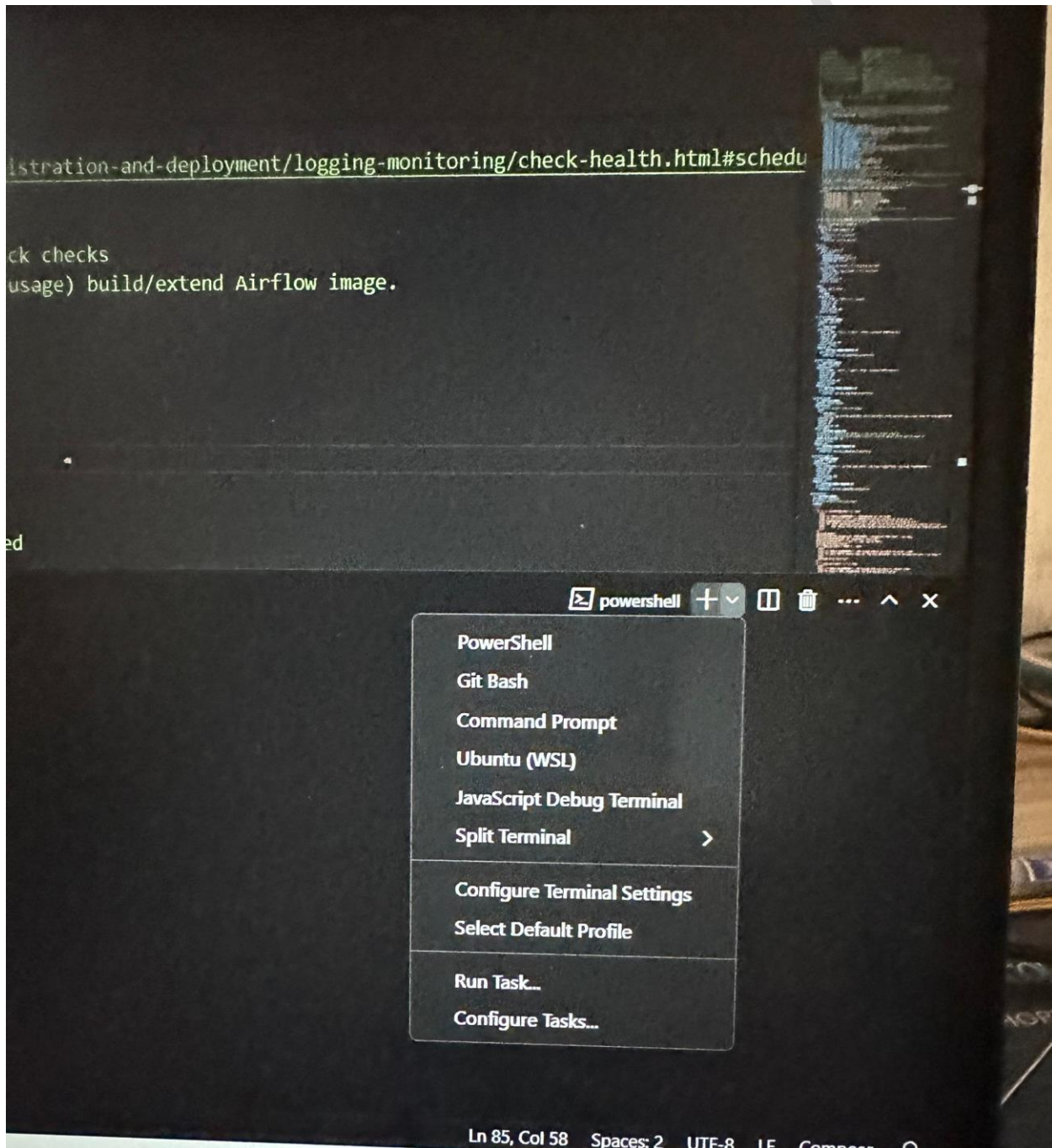
Run these commands:

```
cd c:\
```

code . # the full stop there is not an error, include it. It should open a VS code for you

In VS code, open a new terminal (you can do this by taking your mouse to the top of the VS code application, it should show the option to create a new terminal)

Once this is done, you will have to open a new bash terminal. Refer to the image below:



When you click on the dropdown beside the + icon, you should see option for Git Bash. Once this terminal is opened, run the commands below:

```
git clone --depth=1 https://github.com/airbytehq/airbyte.git
```

Once the command above is done, run this:

```
cd airbyte
```

```
./run-ab-platform.sh
```

The command above should take some time depending on your internet speed. You can track the process on the terminal you ran the command on VS code. You can open another Git bash terminal to know if the required services have started running for airbyte, you should see these services running:

CONTAINER ID	IMAGE	COMMAND	NAMES	CREATED	STATUS	PORTS
395d9b66439f	airbyte/proxy:0.51.0	"/run.sh ./run.sh"	airbyte-proxy	44 hours ago	Up 20 seconds	0.0.0.0:8000-8001->8000-8001/tcp, 0.0.0.0:8003->8003/tcp, 80/tcp, 0.0.0.0:8006->8006/tcp
a469fc7e0862	airbyte/worker:0.51.0	"/bin/bash -c airbyt..."	airbyte-worker	44 hours ago	Up 23 seconds	5005/tcp, 0.0.0.0:5806->5806/tcp
1f54643c81e9	airbyte/server:0.51.0	"/bin/bash -c airbyt..."	airbyte-server	44 hours ago	Up 23 seconds	5005/tcp, 8000/tcp, 0.0.0.0:58064->58064/tcp
4f7ddb439bc3	airbyte/cron:0.51.0	"/bin/bash -c airbyt..."	airbyte-cron	44 hours ago	Up 23 seconds	
958a01ec562b	airbyte/connector-builder-server:0.51.0	"/bin/bash -c airbyt..."	airbyte-connector-builder-server	44 hours ago	Up 23 seconds	0.0.0.0:58062->80/tcp
ec4e3a262f85	airbyte/webapp:0.51.0	"/docker-entrypoint..."	airbyte-webapp	44 hours ago	Up 24 seconds	80/tcp
8562fba3e9a3->8006/tcp	airbyte/airbyte-api-server:0.51.0	"/bin/bash -c airbyt..."	airbyte-api-server	44 hours ago	Up 23 seconds	5005/tcp, 0.0.0.0:5806->5806/tcp
9cb035db391fp, 7233-7235/tcp, 7239/tcp	airbyte/temporal:0.51.0	"/update-and-start..."	airbyte-temporal	44 hours ago	Up About a minute	6933-6935/tcp, 6939/tcp
a00881b69a9a	airbyte/db:0.51.0	"docker-entrypoint.s..."	airbyte-db	44 hours ago	Up About a minute	5432/tcp

You should now be able to access airbyte from your browser using this url: localhost:8000

Username: airbyte

Password: password

Airflow SetUp (In conjunction with dbt and great_expectations)

While airbyte is trying to setup, you might want to proceed to doing airflow setup at the same time.

Open a new cmd

Create a new directory by running this command:

mkdir airflow # (this will throw up an error if you have done it before), so proceed to next command

cd airflow

code . (this will open another VS code window)

Once VS code is opened, open a new terminal (Git Bash terminal) and run this command:

Please if you have a docker compose.yaml file in this directory before, kindly delete that file before running the command below.

`curl -L -O 'https://airflow.apache.org/docs/apache-airflow/2.8.4/docker-compose.yaml'` # This command will download a docker-compose.yaml file into the airflow directory

Now, let's proceed to edit the file, click on the file from within VS Code, and scroll down to line 53:

```
52 image: ${AIRFLOW_IMAGE_NAME:-apache/airflow:2.8.4}
53 # build: .
54 environment:
```

Remove the comment (#) from that line and should look like this:

```
52 image: ${AIRFLOW_IMAGE_NAME:-apache/airflow:2.8.4}
53 build: .
54 environment:
```

scroll down to line 76 where you have the below:

```
72 volumes:
73   - ${AIRFLOW_PROJ_DIR:-.}/dags:/opt/airflow/dags
74   - ${AIRFLOW_PROJ_DIR:-.}/logs:/opt/airflow/logs
75   - ${AIRFLOW_PROJ_DIR:-.}/config:/opt/airflow/config
76   - ${AIRFLOW_PROJ_DIR:-.}/plugins:/opt/airflow/plugins
```

Edit this part to include these parts:

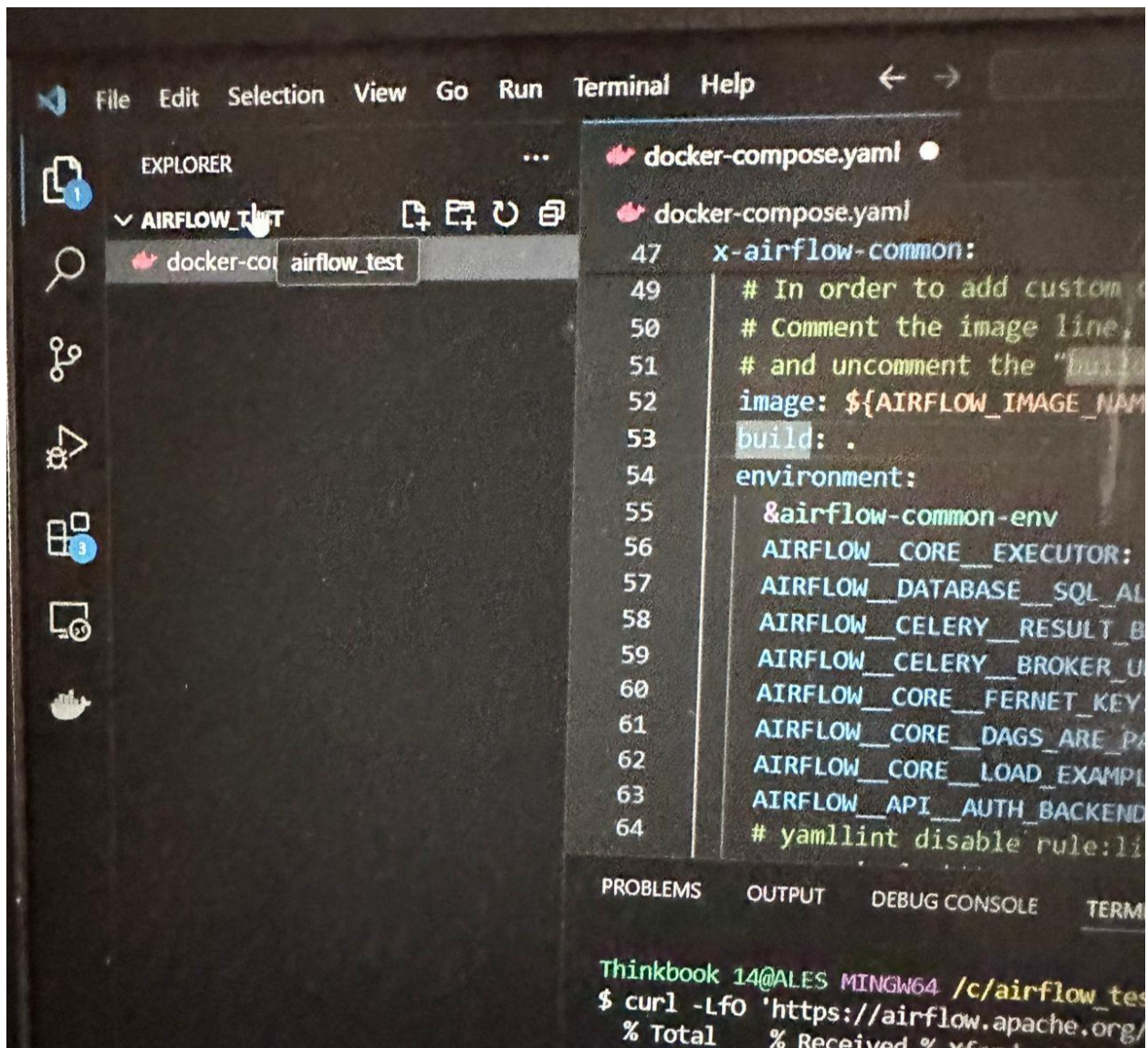
```
- ${AIRFLOW_PROJ_DIR:-.}/dbt:/opt/airflow/dbt
- ${AIRFLOW_PROJ_DIR:-.}/ge:/opt/airflow/ge
- "<to be modified>/.dbt:/home/airflow/.dbt" # to be modified
```

The last line above should be modified to include the path of your dbt profile as you have noted on your notepad, replace the <to be modified> with /C/Users/Ability (i.e the part before the \.dbt as you noted on your notepad, it should look like this:

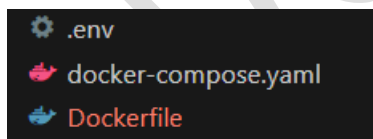
```
72 volumes:
73   - ${AIRFLOW_PROJ_DIR:-.}/dags:/opt/airflow/dags
74   - ${AIRFLOW_PROJ_DIR:-.}/logs:/opt/airflow/logs
75   - ${AIRFLOW_PROJ_DIR:-.}/config:/opt/airflow/config
76   - ${AIRFLOW_PROJ_DIR:-.}/plugins:/opt/airflow/plugins
77   - ${AIRFLOW_PROJ_DIR:-.}/dbt:/opt/airflow/dbt
78   - ${AIRFLOW_PROJ_DIR:-.}/ge:/opt/airflow/ge
79   - "/C/Users/Ability/.dbt:/home/airflow/.dbt" # to be modified
```

After all these are done, save the file.

Next, let's create additional files (.env & Dockerfile) in the airflow directory (Don't do this step if you have the files before).



As you can see in the image above, hover your mouse on the airflow directory, there are four icons to the right, the first icon is to create a new file, click on it and create a file named .env and another file named Dockerfile. You should have something like this:



Click on the .env file and add this text and save:

```
AIRFLOW_UID=50000
```

Click on the Dockerfile and add this text and save:

```
FROM apache/airflow:2.8.4-python3.10
```

```

#ENV AIRFLOW__CORE__LOAD_EXAMPLES=True
#ENV AIRFLOW__DATABASE__SQL_ALCHEMY_CONN=my_conn_string
#USER root
#RUN apt-get update \
##    && sudo apt update && sudo apt upgrade && sudo apt-get install python3.9 \
#    && apt-get install -y --no-install-recommends \
#        vim
#ADD requirements3.10.txt .
USER root
RUN apt update
RUN apt install git -y
RUN apt update -y && sudo apt install -y build-essential libpq-dev
RUN apt-get update \
    && apt-get install -y --no-install-recommends \
        gcc \
        heimdal-dev \
    && apt-get autoremove -yqq --purge \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/*
#RUN apt-get install vim -y
USER airflow
#RUN pip install apache-airflow==${AIRFLOW_VERSION} -r requirements3.txt
#RUN pip uninstall -y argparse
#ENV AIRFLOW_VERSION=2.8.4
#ENV PYTHON_VERSION="$(python --version | cut -d ' ' -f 2 | cut -d '.' -f 1-2)"
ENV wsl_ip="$(cat /etc/resolv.conf | grep nameserver | cut -d ' ' -f 2)"
ENV PYTHON_VERSION=3.10
ENV CONSTRAINT_URL="https://raw.githubusercontent.com/apache/airflow/constraints-
${AIRFLOW_VERSION}/constraints-${PYTHON_VERSION}.txt"
RUN pip install "apache-
airflow[async,postgres,google,cncf.kubernetes,snowflake,airbyte]==${AIRFLOW_VERSI
ON}" --constraint "${CONSTRAINT_URL}"
#RUN pip install "apache-
airflow[async,postgres,google,cncf.kubernetes,snowflake,airbyte]==${AIRFLOW_VERSI
ON}" -r requirements2.txt
RUN pip uninstall -y argparse
RUN pip install dbt-core
RUN pip install dbt-postgres
RUN pip install dbt-redshift
RUN pip install dbt-snowflake
RUN pip install dbt-bigquery
RUN pip install great_expectations
RUN pip install airflow-provider-great-expectations
RUN pip install astronomer-cosmos[dbt.all]

```









```

USER root
RUN mkdir -p -m 777 /opt/airflow/ge
RUN mkdir -p -m 777 /opt/airflow/dbt
RUN mkdir -p -m 777 /opt/airflow/dbt/logs
RUN mkdir -p -m 777 /opt/airflow/dbt/build
RUN mkdir -p -m 777 /opt/airflow/logs
RUN mkdir -p -m 777 /opt/airflow/dags
RUN mkdir -p -m 777 /opt/airflow/plugins
RUN mkdir -p -m 777 /opt/airflow/logs/scheduler
RUN mkdir -p -m 777 /dbt
#RUN mkdir -p -m 777 /opt/airflow/.dbt
RUN chmod a+x /usr/bin/git
#COPY .dbt /opt/airflow/dbt
#COPY .dbt/profiles.yml /home/airflow/.dbt/profiles.yml

```

Once this is done, go to Git Bash terminal you opened within the VS code and run this command:

Note that if you have deployed airflow once before, you have to stop and delete the containers under airflow in your Docker desktop as shown below before running the next command:

<input type="checkbox"/>		airflow	-	Running (6/7)			
<input type="checkbox"/>		airflow-worker-1 20ef2b8c4da4	apache/airflow:2.8.4	Running	4 minutes ago		
<input type="checkbox"/>		airflow-triggerer-1 c224b4f38fb1	apache/airflow:2.8.4	Running	4 minutes ago		
<input type="checkbox"/>		airflow-webserver-1 f4e1567bb59b	apache/airflow:2.8.4	Running	8080:8080		
<input type="checkbox"/>		airflow-scheduler-1 12f748381d2e	apache/airflow:2.8.4	Running	4 minutes ago		
<input type="checkbox"/>		airflow-init-1 b21cb1e49927	apache/airflow:2.8.4	Exited			
<input type="checkbox"/>		postgres-1 67159a525287	postgres:13	Running	4 minutes ago		
<input type="checkbox"/>		redis-1 ah5738205c02	redis:latest	Running	4 minutes ago		

Now run the command below

`docker compose up --build`

This will start pulling the necessary images and build them into containers and start the necessary airflow services.

If everything works fine, you should see these services when you run `docker ps`:


```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
20ef2b8c4da4	apache/airflow:2.8.4	"/usr/bin/dumb-init ..."	7 days ago	Up 4 minutes (healthy)	8080/tcp	airflow-airfl
ow-worker-1						
c224b4f38fb1	apache/airflow:2.8.4	"/usr/bin/dumb-init ..."	7 days ago	Up 4 minutes (healthy)	8080/tcp	airflow-airfl
ow-triggerer-1						
f4e1567bb59b	apache/airflow:2.8.4	"/usr/bin/dumb-init ..."	7 days ago	Up 4 minutes (healthy)	0.0.0.0:8080->8080/tcp	airflow-airfl
ow-webserver-1						
12f748381d2e	apache/airflow:2.8.4	"/usr/bin/dumb-init ..."	7 days ago	Up 4 minutes (healthy)	8080/tcp	airflow-airfl
ow-scheduler-1						
67159a525287	postgres:13	"docker-entrypoint.s..."	7 days ago	Up 4 minutes (healthy)	5432/tcp	airflow-postg
res-1						
ab5738205c02	redis:latest	"docker-entrypoint.s..."	7 days ago	Up 4 minutes (healthy)	6379/tcp	airflow-redis
-1						

It should show status healthy for everything

You should now be able to access airflow from your browser by opening this url: localhost:8080

Username: airflow

Password: airflow

Stopping the services:

For airbyte

Open a new cmd

Run the commands below:

```
cd c:\airbyte
```

```
docker compose stop # This will stop all the containers
```

For airflow

Open a new cmd

Run the commands below:

```
cd c:\airflow
```

```
docker compose stop # This will stop all the containers
```

Join the Online Community

Kindly reach out on the WhatsApp community group if you encounter any difficulty while running these steps. Here is the link to join: <https://chat.whatsapp.com/BEzkst09b2z9LfV5AwPQex>