Airflow Local Setup, VSCode on Windows

Friday, August 01, 2025 4:23 PM

Meeting w/ JvN

--PM

FEATURE 1384537 'Airflow Local Environment'

--Desc:

Setup local Airflow environment on my local PCs so I can write and troubleshoot DAGs locally, before committing them to our Airflow QA environment.

Why:

-Huge time savings writing code

-No need to misuse Gitlab for debugging

STORY 1384545

8/4/2025 Setup a standalone Airflow container for [name] project 'name'

Recreated all the env variables needed for 'lname'

Successful w/ apache/airflow:2.9.1

Retraced previous steps from the meeting with JvN, and created a local, standalone Airflow container on my laptop for project 'name'

Successfully tested with initial 'name_dag.py', which reads 2 input files and returns a (minimally) processed information about a differences dataframe.

Successfully coded and debugged in local Airflow container, deployed to QA Airflow where the DAG also ran successfully.

--Project: Compare Rates

A)Get 2 csv files from a folder in OneDrive

B) run compare2Files script, and

C) output a .csv files with differences

--My request:

Request: i would like to write code for Airflow without having to checkin my code every time just to see if the code would work.

A: recommends to use Podman (no need for Docker license)

Good example of Airflow DAG:

For one drive i have an automation I am currently working on that looks in a personal one drive folder for files to grab, but it should work for other drives as well:

validate_complex_duties/dags/uitls/onedrive_helper.py

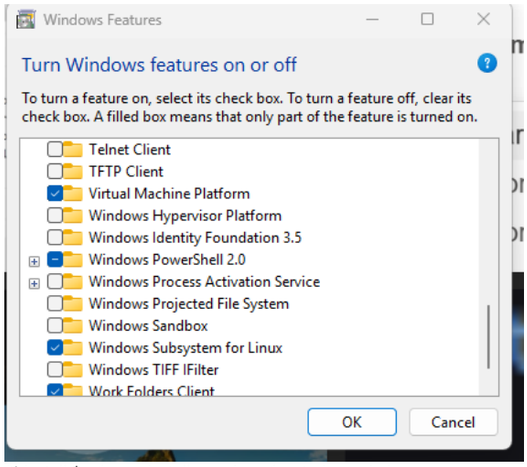
how i use those modules in this file: validate_complex_duties/dags/validate_duties.py

use the az_svc_username creds, make sure that service account has access to the one drive folder

Prerequisites

- Airflow access approved
- Install WSL extensions in VSCode -> get your project opened in WSL

A) Enable Windows Features



B) install $wsl.exe –install, requires elevated permissions

Install with Ubuntu distro

$ wsl –install –d Ubuntu

Get a message: The Windows Subsystem for Linux is not installed. You can install it by running 'wsl.exe --install'. For more info, visit https://aka.ms/wslinstall

- Install podman

$ sudo apt install podman

Want a good Docker image

Docker Image for Apache Airflow — docker-stack Documentation

Docker Image for Apache Airflow — docker-stack Documentation

(I.e: `apache/airflow:latest-python3.12`)

--How we made 'helloWorld' DAG work:

MyUseCase: i want to run an Airflow container using podman so i can run this project's DAG locally

After rebooting my machine (wanted to clear everything and start clean)

Step. Created a brand new Airflow project from a template
Name: `anya-helloworld-dag`
Copied `helloWorld.py` DAG file into `dags` folder
Nothing else. and DAG code = JvN code

Step. In VS Code
Opened project `anya-helloworld-dag` in WSL
(ctrl-shift-p Open WSL this folder)

-ensure no old containers (even after reboot)
$ podman stop airflow-standalone
$ podman rm airflow-standalone

-create container airflow-standalone (with custom pip install module)
$ podman run --name airflow-standalone -d -p 8080:8080 -v "$PWD/dags":/opt/airflow/dags
apache/airflow:2.9.1 bash -c "pip install openpyxl && airflow standalone"
(or create it with a bash script **$ sudo bash start.sh**)

-check got created
$ podman ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
bbc03c3e9989 docker.io/apache/airflow:2.9.1 bash -c pip insta... About an hour ago Up About an hour 0.0.0.0:8080->8080/tcp airflow-standalone

-to see logs (and hopefully password in the beginning
$ podman logs -f airflow-standalone
Or look for password
$ podman logs airflow-standalone | grep -i password

-to restart
$ podman restart airflow-standalone

--Troubleshooting

- Permissions?
  $ chmod -R a+r $PWD/dags

- Exec in?
  $ podman exec -it airflow-standalone bash
  in bash now:
  airflow@8b2163c10209:/opt/airflow**$ env**
  airflow@8b2163c10209:/opt/airflow**$ env | grep admin**

- See what's running and not? Webserver?
  $ hostname -I
  127.0.1.1
  $ lsof -i
  $ lsof -i :8080 | grep LISTEN

- My error1 in Airflow:
  Im running my first "hello world" DAG now. it's taking way too long for printing "hello". so something is not right? it says "queued"
  A: that means the executor is not picking it up

- MyUseCase:
  Azure variables that are needed for the token.
  A: they are in our repo
  username is AZ-Svc-CIS-Automation-Platform@Domain.mail.onmicrosoft.com

--How it didn't work

-this version couldn't run a simple HelloWorld DAG for me
$ podman run --name airflow-standalone -d -p 8080:8080 -v "$PWD/dags":/opt/airflow/dags
apache/airflow:3.0.3 bash -c "pip install openpyxl && airflow standalone"

--Chat 7/31/2025

Support installed WSL for me. Ubuntu distribution installed automatically. not sure if i need it at all?
we can proceed with VSCode setup.

8/1/2025
my goal is to setup Podman. will u have time to help me today?

did you install the WSL extension?
yes

```
podman run --name airflow-standalone \
 -d \
 -p 8080:8080 \
 -v "$PWD/dags":/opt/airflow/dags \
 apache/airflow:2.9.1 standalone
```

Docker Image for Apache Airflow — docker-stack Documentation
Docker Image for Apache Airflow — docker-stack Documentation

apache/airflow:latest-python3.12

sudo podman pull apache/airflow:latest-python3.12

podman logs airflow-standalone

apache/airflow:2.9.1 standalone

podman exec -it airflow-standalone bash

```
podman run --name airflow-standalone \
  -d \
  -p 8080:8080 \
  -v "$PWD/dags":/opt/airflow/dags \
  -e _AIRFLOW_WWW_USER_USERNAME=admin \
  -e _AIRFLOW_WWW_USER_PASSWORD=admin \
  apache/airflow:3.0.3 standalone
```

we will want to make a requirements.txt file in the project that lists the python modules you need. then I can provide the additional commands so they install in the container

that makes sense. good with modules now, but having trouble with Azure variables that are needed for the token.

Im running my first "hello world" DAG now. it's taking way too long for printing "hello". so something is not right?

it says "queued"

yeah that means the executor is not picking it up. once I get off the call we can take alook

```
podman restart airflow-standalone
```

```
lsof -i :8080 | grep LISTEN
```

```
tail -f ~/airflow/logs/scheduler/latest/*.log
```

```
podman exec -it airflow-standalone airflow db reset
```

```
podman restart airflow-standalone
```

```
podman run --name airflow-standalone   -d   -p 8080:8080   -v "$PWD/dags":/opt/airflow/dags
apache/airflow:3.0.3
```