# Getting Started with Airflow Using Docker - Beginner

WEB: Apache Airflow, which is “a platform to programmatically author, schedule and monitor workflows”.

Essentially, Airflow is cron on steroids: it allows you to schedule tasks to run, run them in a particular order, and monitor / manage all of your tasks. It’s becoming very popular among data engineers / data scientists as a great tool for orchestrating ETL pipelines and monitor them as they run.

## Follow below steps to do hands-on on Airflow.

### Steps to install Airflow:

#### Install and run docker desktop application

#### create docker hub account

#### run docker desktop and install required libs

#### open command prompt and run command docker - if not running so removed all docker\* from your user variables - edit env variables in windows

#### check docker command running or not in cmd - if running so follow below steps - otherwise re-install.

#### pull airflow image from docker hub - docker pull puckel/docker-airflow

#### check images - docker images

#### create container - docker run -d -p 8080:8080 puckel/docker-airflow webserver

#### check running container - docker ps

#### check Airflow ui - http://localhost:8080/admin

#### Optional - command to go inside container -docker exec -ti <container name> bash

### Steps to create dags in Airflow.

#### I have created two dags(simple - print hello, complex - cross communicatiton between tasks)- url for repository

#### download my airflow repository and copy dags to airflow container(/usr/local/airflow/dags)

#### prune airflow container - docker container prune

#### create airflow container with dag - docker run -d -p 8080:8080 -v D:\vscode\_projects\Airflow\dags\:/usr/local/airflow/dags puckel/docker-airflow webserver

#### wait 2 min - and open airflow ui - http://localhost:8080/admin/ --you can see dags

#### you can start/stop dag using on/off icon

#### click on dag and open graph view - you can see the status

#### now click on task and check the logs - on yours.

#### to stop container run below commands

#### docker container ps -- check container id

#### docker container stop <id>

#### docker container prune - delete all stop container