Introducing airflowctl

A CLI to streamline getting started with Airflow

Kaxil Naik
Who am I?

Committer & PMC Member of Apache Airflow

Director of Engineering @ Astronomer
Happy Ganesh Chaturthi
The Motivation!
Challenges in getting started with Airflow

- Python dependency Hell
  - Broken installations due to one of Airflow’s transitive dependencies
  - Mitigated partially by Constraints file
  - Isolation of dependencies is still a problem

- Standard Project structure

- Current Options:
  - Docker via Docker-compose
  - Helm
  - Virtual Environments
Challenges in getting started with Airflow

Docker / Helm

- Docker & Helm has a learning curve and needs familiarity
- Pain to debug
- Images can be huge!
- Volume mounts – why are my changes not reflected in the containers!!
- Installing deps needs rebuilding of images

Virtual Environments

- Cumbersome & just too many ways to manage virtual environments
- Settings Airflow Home for multiple Airflow projects is a pain
Introducing “airflowctl”
Goal:
Streamline getting started with Airflow
Quick Start

```
pip install airflowctl

airflowctl init my_airflow_project --build-start
```
Features

Single command to Install & Setup a single Airflow project

Standardized project structure

Airflow Connections & Variables Management

Dependency Isolation & Automatic Virtual Environment Management

Allows Installing a Python version not available on local machine

Management of multiple Airflow projects

Runs LocalExecutor with Sqlite for Airflow 2.6+
airflowctl --help

Usage: airflowctl [OPTIONS] COMMAND [ARGS]...

Streamline getting started with Apache Airflow™ and managing multiple Airflow projects.

Options
---
--install-completion  Install completion for the current shell.
--show-completion     Show completion for the current shell, to copy it or customize the installation.
                      Show this message and exit.

Commands
---
aflow  Forward commands to Airflow CLI.
build Build an Airflow project. This command sets up the project environment, installs Apache Airflow and its dependencies.
info  Display information about the current Airflow project.
init  Initialize a new Airflow project.
list  List all Airflow projects created using this CLI.
logs  Continuously display live logs of the background Airflow processes.
start Start Airflow.
stop Stop a running background Airflow process and its entire process tree.
Project structure

```
my_airflow_project
  .env
  .gitignore
  dags
    example_dag_basic.py
  plugins
  requirements.txt
  settings.yaml
```
# Airflow version to be installed
airflow_version: "2.6.3"

# Python version for the project
python_version: "3.8"

# Airflow connections
connexions:
  # Example connection
  # - conn_id: example
  #   conn_type: http
  #   host: http://example.com
  #   port: 80
  #   login: user
  #   password: pass
  #   schema: http
  #   extra:
  #     example_extra_field: example-value

# Airflow variables
variables:
  # Example variable
  # - key: example
  #   value: example-value
  #   description: example-description
Demo
Non-goals

Not meant to be used in Production (yet)!
Roadmap

Interactive Tutorials

Persona-based experience incl. example DAGs & tutorials

Additional verification checks before **installing** Airflow
  SQLite version & User-provided Python version isn’t supported

Additional verification checks before **running** Airflow
  Check if port needed for webserver is free, if not utilize the next available port
  “airflow standalone” doesn’t exist on older versions

Ability to connect & interact with remote Airflow environments

Currently supports “virtualenv” mode, code is pluggable to allow more modes

Contribute this to the Apache Airflow repo in the next 30 days
Questions?

Twitter/X: @kaxil