

Airflow Project README

This repository is currently an empty Airflow project scaffold. These notes will help you set up, organize, and run your Airflow environment.

Project Overview

This repository will be used to build an Apache Airflow pipeline setup for scheduling, orchestrating, and monitoring workflows.

Repository Structure (Planned)

```
|— dags/           # DAG files will go here
|— plugins/        # Custom plugins (hooks, operators)
|— docker-compose.yml # If using Docker setup
|— requirements.txt # Python dependencies
|— README.md       # Documentation (this file)
```

Getting Started

1. Clone the Repository

```
git clone <repo-url>
cd <repo-folder>
```

2. Install Dependencies

If using pip:

```
pip install -r requirements.txt
```

If using Docker:

```
docker compose up -d
```

Creating Your First DAG

Create a Python file inside `dags/` :

```
from airflow import DAG
from airflow.operators.python import PythonOperator
from datetime import datetime

def sample_task():
    print("Airflow DAG is working!")

with DAG(
    dag_id="sample_dag",
    start_date=datetime(2025, 1, 1),
    schedule_interval="@daily",
    catchup=False,
) as dag:
    run = PythonOperator(
        task_id="run_sample",
        python_callable=sample_task,
    )
```

Airflow Commands

Check DAG list:

```
airflow dags list
```

Trigger DAG manually:

```
airflow dags trigger sample_dag
```

View logs:

```
airflow tasks logs sample_dag run_sample 2025-01-01
```

Notes

- Add custom operators under `plugins/operators`

- Keep DAG code modular and clean
 - Avoid heavy computation inside tasks — use external services
 - Store credentials using Airflow Connections
-



Useful Airflow Links

- Airflow Docs: <https://airflow.apache.org/>
 - DAG Best Practices
 - Scheduler & Executor Details
 - Airflow UI Usage Guide
-

Feel free to update this README as the project grows!