Airflow Flat Project Structure

Simple Airflow setup with flat DAGs folder. All DAGs in one directory, shared utilities, and minimal configuration.

#airflow #python #data #orchestration #etl #pipelines

PNG PDF Copy Prompt

Project Directory

```
airflow/
  dags/ DAG definitions
    __init__.py
    etl_daily.py Daily ETL pipel…
    ml_training.py
    reporting.py
    utils/ Shared helpers
      __init__.py
      slack_alerts.py
      sql_helpers.py
  plugins/ Custom operator…
    __init__.py
    custom_operators.py
  include/ SQL, configs, s…
    sql/
      extract_users.sql
  tests/
    __init__.py
    test_etl_daily.py
  airflow.cfg Airflow config
  requirements.txt
  .gitignore
  README.md
```

Why This Structure?

This flat structure keeps all DAGs in one folder—simple to navigate for small teams. The `include/` folder stores SQL files and scripts referenced by DAGs. Utilities live in `dags/utils/` so they're importable from any DAG.

Key Directories

`dags/` - Python files defining DAGs (auto-discovered by Airflow)

`plugins/` - Custom operators, hooks, and sensors

`include/` - SQL queries, shell scripts, config files

`dags/utils/` - Shared Python utilities for DAGs

Getting Started

1. `pip install apache-airflow`
2. `airflow db init`
3. `airflow users create --username admin --password admin --role Admin --email admin@example.com --firstname Admin --lastname User`
4. `airflow webserver -p 8080` (new terminal)
5. `airflow scheduler` (another terminal)

DAG Example

```python
# dags/etl_daily.py
from airflow import DAG
from airflow.operators.python import PythonOperator
from datetime import datetime

with DAG(
    "etl_daily",
    start_date=datetime(2024, 1, 1),
    schedule="@daily",
    catchup=False,
) as dag:

    def extract():
        # Extract logic
        pass

    extract_task = PythonOperator(
        task_id="extract",
        python_callable=extract,
    )
```

When To Use This

- Small teams with <20 DAGs
- Getting started with Airflow
- Simple ETL pipelines without complex dependencies
- Single-project deployments

When To Upgrade

- DAGs folder becoming hard to navigate (20+ DAGs)
- Multiple teams contributing to same repository
- Need versioned, tested DAG packages
- Complex shared logic across many DAGs