

Data Engineering using Python

Dr. Bambang Purnomosidi D. P,



Agenda

1. Apache Airflow: Installation and Setup
2. A Taste of Apache Airflow
3. Multi DAGs Dirs: DagBag
4. Cases

Apache Airflow Installation and Setup

- Install miniconda
- Create environment: `$ conda create -n de-airflow-37 python=3.7`
- `$ conda install airflow`
- `$ conda install mysql`
- Setup database

```
$ airflow initdb
```

```
...
```

```
DB: sqlite:////home/bpdp/airflow/airflow.db
```

```
[2019-07-12 16:33:10,724] {db.py:350} INFO - Creating tables
```

```
INFO [alembic.runtime.migration] Context impl SQLiteImpl.
```

```
...
```

- See \$HOME/airflow dir
- Set env variable: AIRFLOW_HOME (default to \$HOME/airflow). Currently, this env variable still permitted to be in \$HOME/airflow/airflow.cfg (airflow_home = /home/user/airflow)
- load_examples = False
- Run webserver: **\$ airflow webserver** - or - **\$ airflow webserver -p 3000**
- Run scheduler: **\$ airflow scheduler**

A Taste of Apache Airflow

Let's make a 'hello world' workflow just to understand the taste of Apache Airflow.

just_say_hello.py

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



dag_id = "just_say_hello"













with DAG(dag_id=dag_id, start_date=datetime(2019, 1, 21),
        schedule_interval=None) as dag:

    def say_hello():
        print("Hello, Airflow!")

    PythonOperator(task_id="say_hello", python_callable=say_hello)
```

- Put the file inside \$HOME/airflow/dags
- To check: **\$ python just_say_hello.py**. If nothing is wrong, your Apache Airflow installation is successful
- To actually run DAG from shell: **\$ airflow test just_say_hello say_hello 2019-1-20**
- If you want to run the task from web UI, **un-pause** it and then **trigger** it:

		DAG	Schedule	Owner
	<input type="checkbox"/> Off	just_say_hello	None	Airflow

Runs	Links
 	         

☐ Off DAG: just_say_hello

 Graph View

 Tree View

 Task Duration

 Task Tries

 Landing Times

 Refresh

 Delete

Base date:

2019-07-12 20:28:41

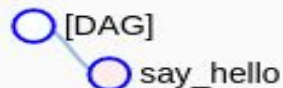
Number of runs:

25



Go

 PythonOperator



Task_id: say_hello
Run: 2019-07-12T20:28:41.131714+00:00
Operator: PythonOperator
Started: 2019-07-12T20:28:43.863235+00:00
Ended: 2019-07-12T20:28:44.805500+00:00
Duration: 0.942265
State: success



☐ Off DAG: just_say_hello

 Graph View

 Tree View

 Task Duration

 Task Tries

 Landing Times

 Refresh

 Delete

success

Base date: 2019-07-12 20:28:42

Number of runs: 25

Run: manual__

Left->Right

Go

Task_id: say_hello

Run: 2019-07-12T20:28:41.131714+00:00

Operator: PythonOperator

Started: 2019-07-12T20:28:43.863235+00:00

Ended: 2019-07-12T20:28:44.805500+00:00

Duration: 0.942265

State: success

success

run

say_hello

Multi DAG Dirs: DagBag

add_dag_bags.py - put this inside \$AIRFLOW_HOME/dags

```
"""
    Add DAG dirs
"""

import os
from airflow.models import DagBag

dags_dirs = ['~/kerjaan/src/airflow/dags-01', '~/kerjaan/src/airflow/dags-02']

for dir in dags_dirs:
    dag_bag = DagBag(os.path.expanduser(dir))

    if dag_bag:
        for dag_id, dag in dag_bag.dags.items():
            globals()[dag_id] = dag
```

- Put example files in ~/kerjaan/src/airflow/dags-01 (the correct one) and ~/kerjaan/src/airflow/dags-02 (the wrong one)

```
~/k/s/airflow ➤ tree ~/kerjaan/src/airflow/  
/home/bpdp/kerjaan/src/airflow/  
├── dags-01  
│   ├── hello_dag_bag_correct.py  
│   └── __pycache__  
│       └── hello_dag_bag_correct.cpython-37.pyc  
└── dags-02  
    ├── hello_dag_bag_wrong.py  
    └── __pycache__  
        └── hello_dag_bag_wrong.cpython-37.pyc  
  
4 directories, 4 files  
~/k/s/airflow ➤
```

hello_dag_bag_correct.py

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

```
dag_id = "hello_dag_bag_correct"
```

```
with DAG(dag_id=dag_id, start_date=datetime(2018, 11, 14),
        schedule_interval=None) as dag:
```

```
    def say_hello_correct():
        print("Hello, correct DAG!")
```

```
    PythonOperator(task_id="say_hello_correct", python_callable=say_hello_correct)
```

hello_dag_bag_wrong.py

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

```
from nonexistent import astral
```

```
dag_id = "hello_dag_bag_wrong"
```








```
with DAG(dag_id=dag_id, start_date=datetime(2018, 11, 14),
        schedule_interval=None) as dag:
```

```
    def say_hello_wrong():
        print("Hello, wrong DAG!")
```

```
    PythonOperator(task_id="say_hello_wrong", python_callable=say_hello_wrong)
```

```
[2019-07-13 04:24:08 +0700] [15002] [INFO] Handling signal: ttou
[2019-07-13 04:24:08 +0700] [3860] [INFO] Worker exiting (pid: 3860)
[2019-07-13 04:24:39 +0700] [15002] [INFO] Handling signal: ttin
[2019-07-13 04:24:39 +0700] [11090] [INFO] Booting worker with pid: 11090
[2019-07-13 04:24:39,087] {__init__.py:51} INFO - Using executor SequentialExecutor
[2019-07-13 04:24:39,252] {__init__.py:305} INFO - Filling up the DagBag from /home/bpdp/airflow/dags
[2019-07-13 04:24:39,254] {__init__.py:305} INFO - Filling up the DagBag from /home/bpdp/kerjaan/src/airflow/dags-01
[2019-07-13 04:24:39,255] {__init__.py:305} INFO - Filling up the DagBag from /home/bpdp/kerjaan/src/airflow/dags-02
```

DAGs

		DAG	Schedule	Owner	Recent T
	<input type="checkbox"/>	hello_dag_bag_correct	None	Airflow	 
	<input checked="" type="checkbox"/>	just_say_hello	None	Airflow	 

DAG yang salah tidak muncul, tapi log error muncul di webserver log

```
2019-07-13 04:07:54 +0700] [15002] [INFO] Handling signal: ttou
2019-07-13 04:07:54 +0700] [29724] [INFO] Worker exiting (pid: 29724)
2019-07-13 04:08:25 +0700] [15002] [INFO] Handling signal: ttin
2019-07-13 04:08:25 +0700] [3304] [INFO] Booting worker with pid: 3304
2019-07-13 04:08:25,146] {__init__.py:51} INFO - Using executor SequentialExecutor
2019-07-13 04:08:25,308] {__init__.py:305} INFO - Filling up the DagBag from /home/bpdp/airflow/dags
2019-07-13 04:08:25,310] {__init__.py:305} INFO - Filling up the DagBag from /home/bpdp/kerjaan/src/airflow/dags-01
2019-07-13 04:08:25,311] {__init__.py:305} INFO - Filling up the DagBag from /home/bpdp/kerjaan/src/airflow/dags-02
2019-07-13 04:08:25,312] {__init__.py:416} ERROR - Failed to import: /home/bpdp/kerjaan/src/airflow/dags-02/hello_dag_bag_wrong.py
Traceback (most recent call last):
  File "/home/bpdp/software/python-dev-tools/miniconda3.7/envs/de-airflow-37/lib/python3.7/site-packages/airflow/models/__init__.py",
line 413, in process_file
    m = imp.load_source(mod_name, filepath)
  File "/home/bpdp/software/python-dev-tools/miniconda3.7/envs/de-airflow-37/lib/python3.7/imp.py", line 171, in load_source
    module = _load(spec)
  File "<frozen importlib._bootstrap>", line 696, in _load
  File "<frozen importlib._bootstrap>", line 677, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 728, in exec_module
  File "<frozen importlib._bootstrap>", line 219, in _call_with_frames_removed
  File "/home/bpdp/kerjaan/src/airflow/dags-02/hello_dag_bag_wrong.py", line 5, in <module>
    from nonexistent import astral
ModuleNotFoundError: No module named 'nonexistent'
```