

UNDERSTANDING DAG (PROGRAM PERSPECTIVE)

By Guha Rajan



Program flow

- Airflow DAGs are python files with following sections



Activity I

Lets now create a simple program with two tasks namely.

1. Hello – which would display ‘Hello world’ (using python operator).
2. Bye – which would display ‘Bye’ (echo command in Unix / Linux bash scripting – using bash operator)

‘Hello’ task(T1) to run first and after successful completion of this task, ‘Bye’ task(T2) would need to run.

I. Imports

- `from __future__ import print_function`
- `import datetime`
- `from airflow import models`
- `from airflow.operators import bash_operator`
- `from airflow.operators import python_operator`

2. Arguments

- **Define defaults and DAG specific arguments.**
- `default_dag_args = {
 'start_date': datetime.datetime(2019, 3, 22),
}`

Set ups

- ***Provide DAG with a Name and schedule intervals***
- with `models.DAG(`
 - `'composer_sample_simple_greeting_two',`
 - `schedule_interval=datetime.timedelta(days=1),`
 - `default_args=default_dag_args) as dag:`

Tasks

- ***Define the function, tasks with operators***

```
def greeting():
```

```
    import logging
```

```
    logging.info('Hello World!')
```

```
hello_python = python_operator.PythonOperator(  
    task_id='hello',
```

```
    python_callable=greeting)
```

```
goodbye_bash = bash_operator.BashOperator(  
    task_id='bye',
```

```
    bash_command='echo Goodbye.')
```


Schedule

- ***Define the schedule:***
- `hello_python >> goodbye_bash`