



UNIVERSITAT DE
BARCELONA

Master in Fundamental Principles of Data Science

Dr Rohit Kumar



UNIVERSITAT DE
BARCELONA

Airflow Advance

Airflow Variables

- Variables are key-value stores in Airflow's metadata database.
- It is used to store and retrieve arbitrary content or settings from the metadata database.

When to use Variables

- Variables are mostly used to store static values like:
 - config variables
 - a configuration file
 - list of tables
 - list of IDs to dynamically generate tasks from
- Separate the constants and variables from pipeline code:
 - It is useful to have some variables or configuration items accessible and modifiable through the UI.

Working with Variables

- Variables can be listed, created, updated and deleted from the UI (Admin -> Variables).
- In addition, json settings files can be bulk uploaded through the UI.
 - Example:

```
{  
    "example_variables_config": {  
        "var1": "value1",  
        "var2": [1, 2, 3],  
        "var3": {  
            "k": "value3"  
        }  
    }  
}
```

You have to keep inside Airflow home folder/dags/config



Airflow Variables

Variables

Choose File

No file chosen

Import Variables









List (4)

Create

Add Filter ▾

With selected ▾

Search

<input type="checkbox"/>		Key	Val
<input type="checkbox"/>	 	example_variables_config	{"var1": "value1", "var2": [1, 2, 3], "var3": {"k": "value3"}}
<input type="checkbox"/>	 	var1	value1
<input type="checkbox"/>	 	var2	[1, 2, 3]
<input type="checkbox"/>	 	var3	{'k': 'value3'}

Accessing Variables

```
var1 = Variable.get("var1")  
var2 = Variable.get("var2")  
var3 = Variable.get("var3")
```

!! Problem 3 DB connection.

Recommendation:

```
dag_config = Variable.get("example_variables_config",  
deserialize_json=True)  
var1 = dag_config["var1"]  
var2 = dag_config["var2"]  
var3 = dag_config["var3"]
```

Accessing Variables

You can directly use a variable from a jinja template

1) **`{{ var.value.<variable_name> }}`**

```
t2 = BashOperator( task_id="get_variable_value",  
    bash_command='echo {{ var.value.var3 }}')
```

2) **`{{ var.json.<variable_name> }}`**

```
t3 = BashOperator( task_id="get_variable_json",  
    bash_command='echo {{  
var.json.example_variables_config.var3 }} '
```


Carefull with Variables!!

Since Airflow Variables are stored in Metadata Database, so any call to variables would mean a connection to Metadata DB.

Instead of storing a large number of variable in your DAG, which may end up saturating the number of allowed connections to your database.

It is recommended you store all your DAG configuration inside a single Airflow variable with JSON value.

References

- <https://airflow.apache.org/docs/stable/tutorial.html>
- <http://michal.karzynski.pl/blog/2017/03/19/developing-workflows-with-apache-airflow/>
- <https://www.polidea.com/blog/apache-airflow-tutorial-and-beginners-guide/>
- <https://towardsdatascience.com/getting-started-with-apache-airflow-df1aa77d7b1b>