Use variables to your DAG

Airflow Variables can be mixed with Templating. Templating is a super powerful feature to inject data at runtime. It is particularly useful when you process data based on dates. As you may want rerun past DAG Runs, templating allows you to inject DAG Run logical date at runtime otherwise you would always process the same chunk of data. Let's discover how to do it!

Prerequisites

You should have create the `endpoint` variable and completed the previous activities.

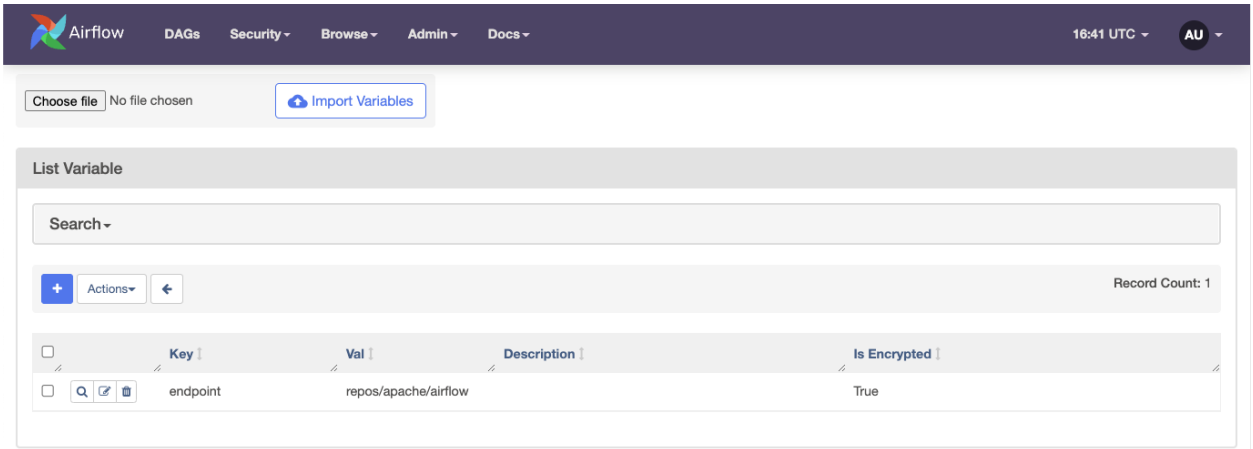
The DAG `extract_stars.py` under the folder `dags` should look like that:

```
from airflow import DAG
from airflow.operators.bash import BashOperator
from airflow.providers.http.operators.http import SimpleHttpOperator

from datetime import datetime

with DAG('extract_stars', schedule_interval='@daily', start_date=datetime(2022, 1, 1), cat
chup=False) as dag:

    get_date = BashOperator(
        task_id="get_date",
        bash_command="date"
    )

    query_github_stats = SimpleHttpOperator(
        task_id="query_github_stats",
        endpoint="repos/apache/airflow",
        method="GET",
        http_conn_id="github_api",
        log_response=True
    )
```

Use a variable to your DAG

Go to `localhost:8080`, click `Admin`, `Variables`

Make sure you have the following variable



Go to your code editor and open `extract_stars.py` under the folder `dags`

In the `query_github_stats` task, change the `endpoint` value by:

`{{ var.value.endpoint }}`

The two pairs of curly brackets indicate a templated placeholder.

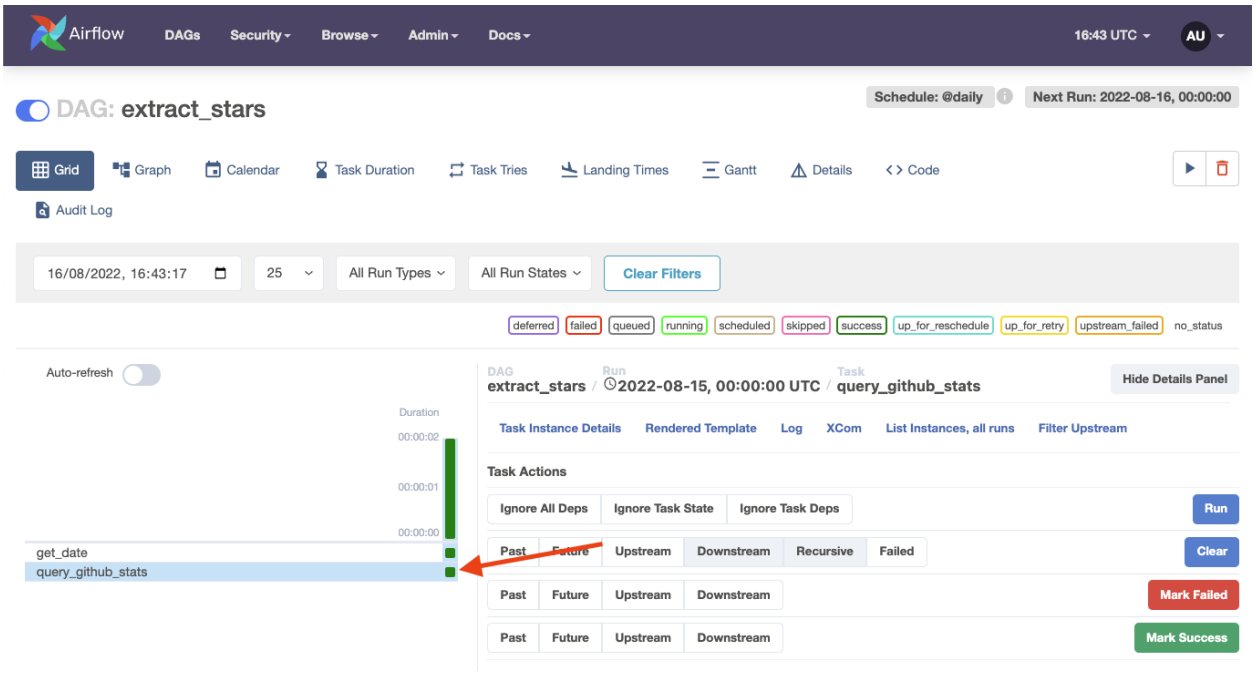
That means `var.value.endpoint` must be replaced by the corresponding value at runtime.

In this case `repos/apache/airflow`

That's it.

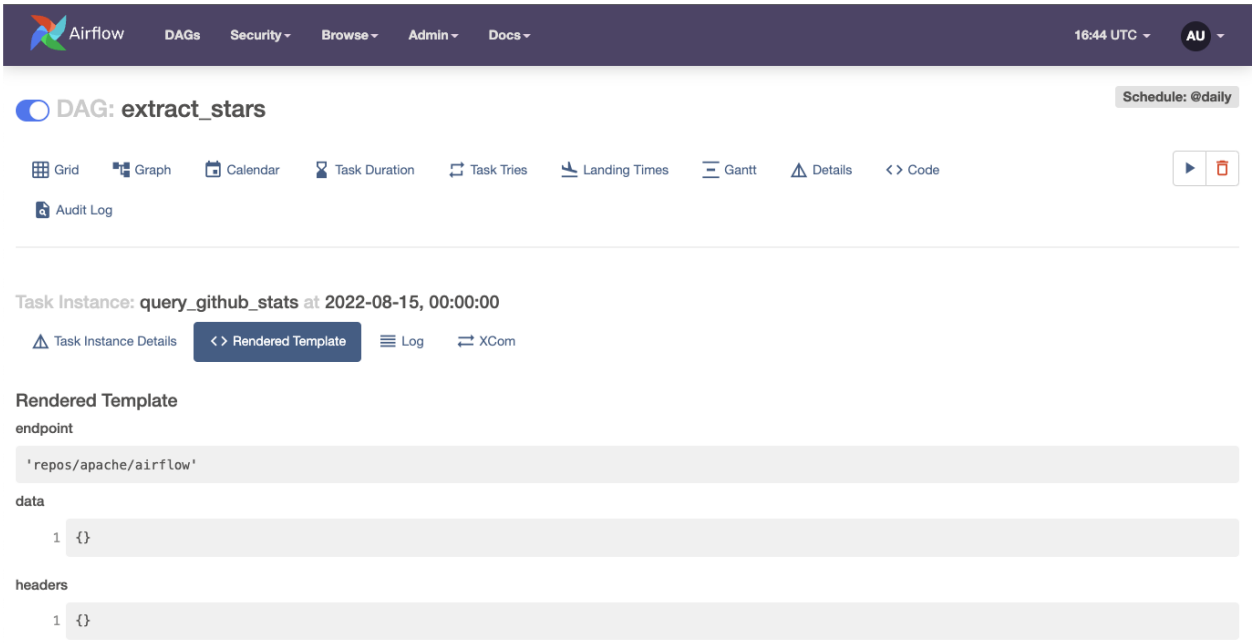
```
query_github_stats = SimpleHttpOperator(
        task_id="query_github_stats",
        endpoint="{{ var.value.endpoint }}",
        method="GET",
        http_conn_id="github_api",
        log_response=True
    )
```

Save the file, go back on the Airflow UI and select the task from the Grid View.



Click `Rendered Template`

You should see the endpoint:



That view gives the rendered template. What you will end up with once the DAG runs and data got injected.

Use the DAG Run date

The DAG pulls Github stars out of the Airflow repository. That number of stars changes everyday.

If you look at the task get_date:

```
get_date = BashOperator(
        task_id="get_date",
        bash_command="date"
    )
```

We execute the bash command `date`

That means we always get the current date.

What if we want to rerun past DAG Runs?

In this case, past DAG Runs would run with the current date and not the dates *at which they got executed*.

To fix that, you have to use templating!

```
get_date = BashOperator(
        task_id="get_date",
        bash_command="echo {{ data_interval_start }}"
    )
```

Now, every time the DAG runs it uses the current date of the DAG Run (`data_interval_start`) and not the current date (`date`).

Don't forget to add `echo` just before.

Well done! You are able to use variables and mix them with templating! 😎

Additional resources

Airflow template ref: https://airflow.apache.org/docs/apache-airflow/stable/templates-ref.html

Astronomer Template guide: https://www.astronomer.io/guides/templating/