Recap

Let’s review what we’ve learned so far

Summary

- A DAG represents a collection of tasks to run, organized in a way that represent their dependencies and relationships. Each node is a task, each edge is a dependency.
- An operator describes a single task in a workflow. A Task is created by instantiating an Operator class. When an Operator is instantiated, this task becomes a node in your DAG. Three types of Operators (Action Operators, Transfer Operators, Sensor Operators).

Summary

- Two ways of describing a dependency between operators in Apache Airflow: set_upstream() / set_downstream or << / >>
- The scheduler's role is to monitor all tasks and DAGs to ensure that everything is executed based on the **start_date** and the **schedule_interval** parameters**. There is also an execution_date which is the latest time your DAG has been executed (last(date) + schedule_interval).** When the Scheduler parses a DAG, it automatically creates a DagRun which is an instantiation of a DAG in time according to the start_date and schedule_interval. The schedule_interval parameter is set to indicate at which interval the Scheduler should run your DAG. It receives preferably a CRON expression as a `str` or a `datetime.timedelta` object.

Summary

- If you run a DAG on a schedule_interval of one day, the run stamped 2016-01-01 will be triggered soon after 2016-01-01T23:59.
- The Scheduler runs your job one schedule_interval AFTER the `start_date`, at the END of the period.
- The Scheduler triggers tasks soon after the start_date + schedule_interval is passed

Summary

- An Airflow DAG with a start_date and a schedule_interval defines a serie of intervals which the Scheduler turns into individual DagRuns to execute.
- Let's assume the start_date of your DAG is 2016-01-01T10:00 and you have started the DAG at 2016-01-01T10:30 with the schedule_interval of */10 * * * * (AFTER every 10 minutes).
- Apache Airflow will run past DAGs for any interval that has not been run. This concept is called Catchup / Backfill.
- This feature allows you to backfill your DB with data produced from your ETL as if it were run from the past.
- If you want to avoid this behavior, you can set the parameter `catchup=False` into the DAG arguments.

Summary

- Workflows are basically the combination of Dags, Operators, Tasks and TaskInstances.

What's next?

Well done!

In this section, we have seen a lot of concepts to understand and you have done your first DAG.

In the following section we are going to see another very important component of Apache Airflow which is the metadata database used in coordination with the executors.