How the Scheduler Works?

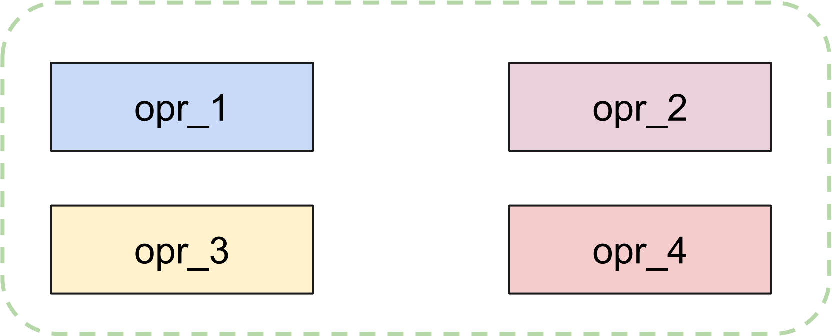
The heart of Apache Airflow

Definition

- 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).**
- The scheduler periodically scans the DAG folder (airflow/dags) to inspect tasks and verifies if they can be triggered or not.

DagRun

- A DAG consists of Tasks and need those tasks to run.
- 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 the schedule_interval.
- When a DagRun is running all tasks inside it will be executed.
- Here is a representation of a DagRun:



Key Parameters

- start_date
  - The first date for which you want to have data produced by the DAG in your database (can be set in the past)
- end_date
  - The date at which your DAG should stop running (usually set to None)
- retries
  - The maximum number of retries before the task fails
- retry_delay
  - The delay between retries.
- schedule_interval
  - The interval at which the Scheduler will trigger your DAG

Schedule Interval

- The schedule_interval parameter is set to indicate at which interval the Scheduler should run your DAG. It preferably receives a CRON expression as a `string` or a `datetime.timedelta` object.
- Alternatively, you can also use a cron “preset” as shown into the following table.

Schedule Interval

Preset	Meaning	Cron
None	Don’t schedule. Manually triggered	
@once	Schedule once and only once	
@hourly	Run once an hour at the beginning of the hour	0 * * * *
@daily	Run once a day at midnight	0 0 * * *
@weekly	Run once a week at midnight on Sunday morning	0 0 * * 0
@monthly	Run once a month at midnight of the first day of the month	0 0 1 * *
@yearly	Run once a year at midnight of January 1	0 0 1 1 *

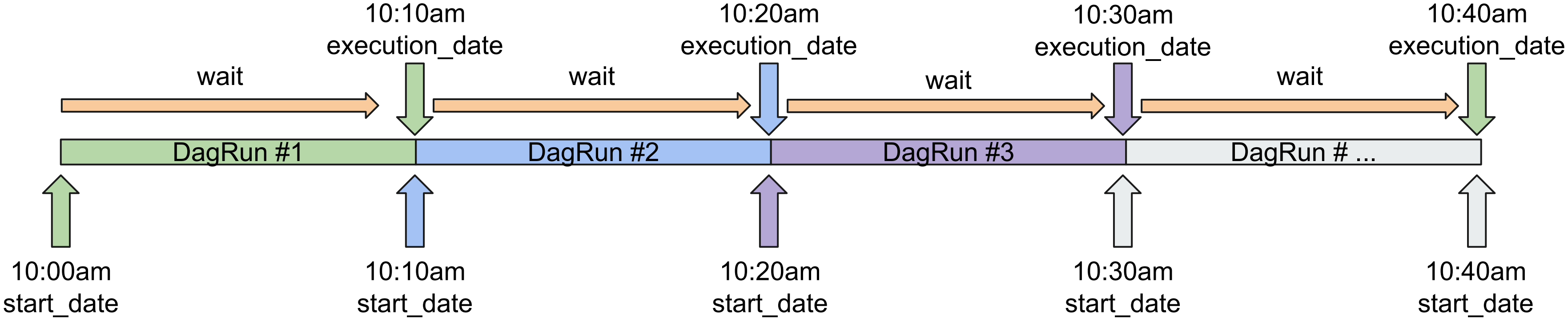
Important Notes

- 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 + scheduler_interval is passed

Backfill and Catchup

- 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.

Example


10:10am
execution_date
10:20am
execution_date
10:30am
execution_date
10:40am
execution_date
wait
wait
wait
wait
DagRun #1
DagRun #2
DagRun #3
DagRun # ...
10:00am
start_date
10:10am
start_date
10:20am
start_date
10:30am
start_date
10:40am
start_date

Final Important Notes

- The first DagRun is created based on the minimum start_date for the tasks in your DAG.
- Subsequent DagRuns are created by the Scheduler based on your DAG's `schedule_interval`, sequentially.