**Extra Materials**

Below are some additional links to help to understand on the related concepts:

* [Airflow: Lesser Known Tips, Tricks, and Best Practises](https://medium.com/datareply/airflow-lesser-known-tips-tricks-and-best-practises-cf4d4a90f8f)
* [Airflow Tips, Tricks & Pitfalls](https://caserta.com/data-blog/airflow-tips-tricks-pitfalls/)
* [Getting started with Apache Airflow](https://towardsdatascience.com/getting-started-with-apache-airflow-df1aa77d7b1b)
* [DAG Writing Best Practices in Apache Airflow](https://www.astronomer.io/guides/dag-best-practices/)

To make the DAG even more compact, try to use the [SubDag](https://www.astronomer.io/guides/subdags" \t "_blank) operator with the dimension loads and hide the repetitive parts behind that. Depending on your set up, using a **subdag** operator could make your DAG cleaner.

### Pro Tip

To set groups of dependencies, you can use lists or tuples as well.

**For example:**

d1 >> d2 >> (d3, d4)

d1 >> d2 >> [d3, d4]

**Suggestions**  
Dependencies are one of Airflow's most powerful and popular features - they allow for previously long, brittle jobs to be broken down into granular parts that are safer, more modular, and reusable. You can check this [link](https://airflow.apache.org/docs/apache-airflow/stable/howto/operator/external_task_sensor.html) to know more about **Managing Dependencies in Apache Airflow**.

**default\_args**

If a dictionary of default\_args is passed to a DAG, it will apply them to any of its operators. This makes it easy to apply a common parameter to many operators without having to type it many times.  
External Resources  
[Backfill and Catchup](https://airflow.apache.org/docs/apache-airflow/stable/scheduler.html#backfill-and-catchup)

**A DAG Run**

is an object representing an instantiation of the DAG in time. Each DAG may or may not have a schedule, which informs how DAG Runs are created. schedule\_interval is defined as a DAG argument, which can be passed a cron expression as a str, a datetime.timedelta object, or one of of the following cron "presets". You may refer to this DAG Runs documentation for more details on scheduling a DAG.

**External Resources**

* [Airflow Scheduling & Triggers](https://airflow.apache.org/scheduler.html)
* [Scheduling Tasks in Airflow](https://www.astronomer.io/guides/scheduling-tasks/)
* [Templating and Macros in Airflow](https://www.astronomer.io/guides/templating)
* [How to add template variable in the Operator task](https://stackoverflow.com/questions/46645001/how-to-add-template-variable-in-the-filename-of-an-emailoperator-task-airflow)
* [Logging in Airflow](https://www.astronomer.io/guides/logging)
* [Adding logs to Airflow Logs](https://stackoverflow.com/questions/40120467/adding-logs-to-airflow-logs)

**Hooks**

are interfaces to external platforms and databases, implementing a common interface when possible and acting as building blocks for operators. Check out the following links for more info.

* [Automate AWS Tasks using Airflow Hooks](https://www.sicara.ai/blog/2019-01-28-automate-aws-tasks-thanks-to-airflow-hooks)
* [Source code for airflow.hooks.S3\_hook](https://airflow.readthedocs.io/en/1.9.0/_modules/airflow/hooks/S3_hook.html)