Recap

Time to sum up!

Summary

- A plugin is a simple way to add functionalities to Apache Airflow without altering its operation.
- Airflow has a simple plugin manager built-in that can integrate external features to its core by simply dropping files into the `$AIRFLOW_HOME/plugins` folder .
- The python modules in the plugin folder get imported, and hooks, operators, sensors, macros, executors and web views get integrated to Airflow's main collections and become available for use as any Airflow's component.
- To create a plugin you must derive the `airflow.plugins_manager.AirflowPlugin` class and reference the objects you want to plug into Airflow.

Summary

- When you create a plugin, be sure to understand exactly what you are doing. Some components require to implement specific functions, for example:
  - An Operator plugin must have an `execute` method.
  - A Sensor plugin must have a `poke` method returning a boolean value.
- You can create an operator, sensor, hook, executor, view, blueprint, menu link and your own macros.

What's Next?

In this section we have seen how can we customize our Airflow installation by creating our own plugin and modifying the user interface to interact with both PostgreSQL and Elasticsearch.

In the following section, we are going to see how can we make Apache Airflow flexible using Docker.

Be ready!