Summary

Thank you for joining us in this module on Airflow Concepts

1. We have learnt that Airflow is an open-source tool that allows users to author, schedule, and monitor workflows in data pipelines.
2. It is coded in Python and is scalable with a user-friendly interface.
3. We explored the several core components, including the web server, scheduler, meta database, triggerer, executor, queue, and worker.
4. We also learnt about the Directed Acyclic Graph (DAG), which is the most crucial concept, and it represents a data pipeline with nodes as tasks and directed edges as dependencies.
5. Moreover, the Operators are objects that encapsulate tasks, and there are three types of operators: action, transfer, and sensor operators. Providers are packages that contain operators for interacting with specific tools.
6. Airflow works by triggering data pipelines through the scheduler, which creates a DAGRun object and a task instance object for the first task. The task instance is then pushed into a queue and executed by the executor.
7. To create a DAG in Airflow, create a file in the "dags/" folder, instantiate the DAG object with parameters such as the unique DAG ID, start date, scheduling interval, and catchup parameter. Once these parameters are defined, tasks can be implemented within the DAG.
8. To create a task, look up the appropriate operator in the registry.astronomer.io and define the task ID and parameters needed for the operator.
9. Airflow is useful for scheduling batch jobs, training machine learning models, and building ETL or ELT pipelines. It saves time by better scheduling and monitoring data pipelines. However, it is not a data streaming solution or a data processing framework, but an orchestrator for batch processing jobs.
10. We also learnt how to define dependencies in Airflow is simple using the right and left bitshift operators, which can be seen in the Airflow UI. Dependencies can be defined between tasks, such as "start >> end" meaning "end" is executed after "start".

Thank you for tuning in. See you in the next one.