Summary

Thank you for joining us in this module on Airflow Connections.

1. We learnt in order to interact with external systems in a data pipeline, one needs to create a connection, which is a set of parameters, identified by a unique connection ID, where the connection contains login, password, hostname, etc., required for accessing external systems in a data pipeline.
2. We also learnt that one can interact with APIs, Snowflake, DBT etc. using connections in Airflow. To create a connection to interact with an API, one probably may need to get an API key and specify the connection ID, type, description, host, and API key.
3. We also explored that the created connections can be listed, interacted with, and filtered from the connections view in Airflow. However if connections were defined using environment variables, they won't be visible on the UI, but they exist and can be exported using the Airflow CLI.
4. We also interacted with Snowflake from a data pipeline using the Snowflake operator in Airflow which involved creating a task with the Snowflake operator, creating a connection with Snowflake, and specifying the SQL request.
5. The Astro CLI provides the airflow_settings.yaml file, which allows for configuring and programmatically creating airflow connections, pools, and variables. The file can be used to define connection parameters such as connection ID, type, host, login, password, and port for various data sources such as Postgres. This saves time in recreating connections in the Airflow UI every time the project is restarted. However, this functionality is only available locally.
6. Overall, creating connections is essential for accessing external systems in a data pipeline. It enables the user to interact with APIs, Snowflake, and DBT etc. and it can be done using the connections view in Airflow or by programmatically creating connections using the Astro/Airflow CLI.

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