Data Pipelines with Airflow

A Data Pipeline is simply a series of steps in which Data is processed
A Data Pipeline to accomplish this task would likely

- Load application event data from a source such as S3 or Kafka
- Load the Data into an Analytics warehouse such as Redshift
- Perform data transformations that identify high-traffic bike docks so the business can determine where to build additional locations

Data validation is the process of ensuring that data is present, correct, and meaningful, ensuring the quality of your data through automated validity checks.

Directed Acyclic Graphs(DAGs): they are a special subset of graphs in which the edges between nodes have a specific direction, and no cycles exist. When we say "no cycles exist exist" what is meant is that nodes can recreate the path to themselves.
Nodes: A step in the data pipeline process, they are tasks
Edges: The dependencies or relationships other between nodes, ordering, and dependencies between tasks.
What makes up Airflow.

- Scheduler => for orchestrating the scheduling of jobs on a trigger or schedule.
- Work Queue => this holds the state of running DAGs and tags
- Worker processor => they execute the operations defined in the DAG
- Database => this stores your credentials, configurations, history, and connection

AirFlow in its self is not a data preprocessing tool, we just mainly use AIrflow to co-ordinate the processes.

Order of operation of Airflow

- The Airflow Scheduler starts DAGs based on time or external triggers.
- Once a DAG is started, the Scheduler looks at the steps within the DAG and determines which steps can run by looking at their dependencies.
- The Scheduler places runnable steps in the queue.
- Workers pick up those tasks and run them.
- Once the worker has finished running the step, the final status of the task is recorded and additional tasks are placed by the scheduler until all tasks are complete.

- Once all tasks have been completed, the DAG is complete.

If you don't schedule the DAG it will run the default value of running once a day.
Airflow uses the Double carrot to schedule when tasks would run. ">>"
Airflow hooks provide a reusable interface to external systems and databases.
Data lineage is a combination of discrete steps involved in the creation, movement, calculation of that dataset.
The less data you process the faster your pipeline runs.
Pipeline **Data partitioning** is the process of isolating data to be analyzed by one or more attributes such as time, logical type, or data size.
Logical **Data partitioning** breaks conceptually related data into discrete groups for processing.
Size partitioning separates data for processing based on the required size or storage limits, this is very key because Airflow workers do not at any time pull worker memory unlike Apache Spark

The way it usually works is to Create the function => Write the dag => Execute the dag in an operator.
As an AirFlow user, you should always check Airflow control to see if what you need has been built by someone else.
Every task in your dag should perform only one job.
Commonly repeated tasks within a DAG can be captured and used as a sub-dag
Drawbacks of using SubDags

- Limit the visibility within the airflow UI
- Abstraction makes understanding what the DAG is doing more difficult
- Encourages premature optimization