**Notes on Airflow ETL Pipeline**

**1. Required Tools and Libraries**

The ETL pipeline uses:

* **Airflow decorators** → To define a DAG and its tasks in a simpler way.
* **Datetime** → To specify when the workflow should start.
* **Pandas** → To read, clean, and process data.

**2. DAG (Directed Acyclic Graph)**

A DAG represents the overall workflow.

* The DAG is named **"etl\_pipeline"**.
* It starts running from **January 1, 2023**.
* The schedule is set to **manual** (it does not run automatically).
* **Catchup** is disabled, meaning missed runs will not be executed later.
* The DAG is tagged with **"example"** and **"etl"**, which helps in identifying it in the Airflow interface.

**3. Workflow Structure**

The pipeline is divided into three main tasks: **Extract**, **Transform**, and **Load**.

**a) Extract**

* This step reads the raw data from a CSV file.
* It checks and prints the shape (rows and columns) of the raw dataset.
* The data is then converted into a dictionary format so it can be passed to the next step.
* **Note:** The file path used is specific to a local machine. In real production, the path should be accessible to the Airflow worker.

**b) Transform**

* The data received from the extraction step is converted back into a DataFrame.
* A simple transformation is applied: **all rows containing missing values are removed**.
* After cleaning, the shape of the new dataset is printed.
* The cleaned data is then passed forward in dictionary format.

**c) Load**

* The cleaned data is converted back into a DataFrame.
* The first few rows of this final dataset are displayed.
* In a real-world scenario, this step would involve **loading the data into a database, data warehouse, or storage system**.

**4. Task Flow**

The tasks run in a strict order:

1. **Extract** → Reads raw data.
2. **Transform** → Cleans the extracted data.
3. **Load** → Outputs or stores the final dataset.

This ensures a proper **ETL flow**: data is always processed step by step.

**5. Execution**

At the end, the DAG is registered with Airflow so it can appear in the Airflow UI.  
Since the schedule is manual, it must be **triggered manually by the user**.

**Final Understanding**

This Airflow pipeline demonstrates a **basic ETL process**:

* **Extract** → Collect raw data.
* **Transform** → Clean and prepare data.
* **Load** → Display or save the processed data.

It serves as an introductory example of how Airflow can be used for data workflows.