Hands-On Example: ETL Pipeline with Apache Airflow and Pandas

Step 1: Set Up Apache Airflow

Install Apache Airflow

bash

Copy code

```
# Using pip to install Airflow
pip install apache-airflow
```

1.

Initialize the Database

bash

Copy code

```
airflow db init
```

2.

Start Airflow Web Server and Scheduler

bash

Copy code

```
airflow webserver --port 8080
airflow scheduler
```

3.
4. **Access Airflow UI**
    - Open your browser and go to `http://localhost:8080`.

Step 2: Create a DAG

1. **Create a New DAG File**
    - Create a Python file named `etl_pipeline.py` in the `dags` folder of your Airflow installation.

Define the DAG

python

Copy code

```
from airflow import DAG
from airflow.operators.python import PythonOperator
from datetime import datetime
import pandas as pd

# Define the functions for each task
```

```
def fetch_data():
    # Simulate fetching data and saving to CSV
    data = {
        'date': ['2023-01-01', '2023-01-02', '2023-01-03'],
        'quantity': [10, 20, 15],
        'price': [5.0, 10.0, 7.5]
    }
    df = pd.DataFrame(data)
    df.to_csv('/tmp/sales_data.csv', index=False)

def process_data():
    # Load the CSV file
    df = pd.read_csv('/tmp/sales_data.csv')
    # Data cleaning
    df['total_sales'] = df['quantity'] * df['price']
    # Save the transformed data
    df.to_csv('/tmp/sales_data_processed.csv', index=False)

# Define the DAG
with DAG('etl_pipeline', start_date=datetime(2023, 1, 1),
schedule_interval='@daily', catchup=False) as dag:
    task1 = PythonOperator(task_id='fetch_data',
python_callable=fetch_data)
    task2 = PythonOperator(task_id='process_data',
python_callable=process_data)

    task1 >> task2  # Set task dependencies
```

2.

Step 3: Run the DAG

1. **Trigger the DAG**
    - Go to the Airflow UI, find your `etl_pipeline` DAG, and click the "Trigger DAG" button.
2. **Monitor the Tasks**
    - You can see the status of each task in the Airflow UI. Make sure both tasks complete successfully.

Step 4: Verify the Output

1. **Check the Output Files**
    - After the DAG runs, check the `/tmp/` directory for the files:
        - `sales_data.csv` (raw data)
        - `sales_data_processed.csv` (processed data with total sales)
2. **Open the Processed File**
    - Open `sales_data_processed.csv` to verify that the `total_sales` column has been correctly calculated.

Conclusion

This hands-on example guides you through setting up an ETL pipeline using Apache Airflow and Pandas. You created a simple workflow that fetches, processes, and saves data, allowing you to understand DAGs, task management, and data manipulation with Pandas. Feel free to extend the project by adding more tasks, incorporating error handling, or visualizing the data!