**Documentation**

1. **Install Required Packages:** Install all required Python libraries:

bash

pip install apache-airflow dvc bs4 requests

1. **Initialize Airflow:** After installation, initialize the Airflow database:

bash

airflow db init airflow webserver -p 8080

1. **DVC Setup:** Initialize DVC and configure it to use Google Drive as remote storage:

bash

dvc init dvc remote add -d storage gdrive://<your-gdrive-folder-id>

**Configuration**

* Configure the **airflow\_dag.py** script to set the schedule and define the task dependencies and retries.
* Ensure the **data\_extraction.py** and **data\_storage.py** scripts are properly linked to your Airflow and DVC setups.

**Running the Pipeline**

1. **Start Airflow Webserver and Scheduler:**

bash

airflow webserver -p 8080 airflow scheduler

1. **Access Airflow UI:** Open a web browser and navigate to **http://localhost:8080** to access the Airflow dashboard and monitor pipeline runs.
2. **Trigger DAG:** Manually trigger the DAG from the Airflow UI to start the pipeline or let it run according to its schedule.

**Version Control with DVC**

* Data versions are controlled and tracked using DVC.
* Use commands like **dvc push**, **dvc pull**, and **dvc status** to manage and track data changes over time.

**Conclusion**

This MLOps pipeline efficiently automates the process of extracting, transforming, and storing data with robust scheduling and version control, making it suitable for production environments in data-driven applications.