**AIRFLOW ORCHESTRATION**

The written pyspark scripts are submitted to the cluster (on-prem or cloud) as spark submit jobs. We will orchestrate the pipeline by creating airflow dags. We can schedule jobs and the target sink gets loaded with the transformed data.

* Setup airflow on the cluster
* Check for dags folder inside home and create a .py file.
* Create a dag under dags and specify the spark submit jobs dependency.
* #Sample script

default\_args = {

'owner': 'hdfs',# It can be any owner

'depends\_on\_past': False,

'email': ['airflow@example.com'],

'email\_on\_failure': False,

'email\_on\_retry': False,

'retries': 1,

}

with DAG(

'tutorial',

default\_args=default\_args,

description='A simple tutorial DAG',

schedule\_interval=timedelta(days=1),

start\_date=datetime(2021, 1, 1),

catchup=False,

tags=['example'],

) as dag:

# t1, t2 and t3 are examples of tasks created by instantiating operators

t1 = BashOperator(

task\_id= ‘DailyDataIngestAndRefine.py’,

bash\_command='spark submit job command,

dag=dag,

retries=3,

)

t2 = BashOperator(

task\_id= ‘EnrichProductReference.py’,

bash\_command='spark submit job command,

dag=dag,

retries=3,

t3 = BashOperator(

task\_id= ‘VendorEnrichment.py’,

bash\_command='spark submit job command,

dag=dag,

retries=3,

)

t1 >> t2 t3

* The above script creates a dag with bitshift dependencies for the job.
* Once t1 gets completed then it triggers t2 and then finally t3.
* After copying the file to airflow home dag on the cluster need to start airflow db
* airflow initdb
* airflow webserver -p
* airflow scheduler.
* Access UI and start the dags, we can edit dags on UI as well.

**DATA PIPELINE IMPLEMENTATION**

* We can deploy this pipeline on Hadoop cluster(HDP) and install all the required binaries to execute the pipeline.
* We can deploy this on cloud on AWS EMR or AZURE HDINSIGHT and execute this pipeline.
* AWS EMR takes long time to spin the cluster and setup needs knowledge.
* We can leverage the processing pipeline with the latest delta lake layer on Databricks which eliminates the limitations of spark and provides optimized and unified analytics platform for both streaming and batch pipeline processing.