import airflow

**Databricks Submit Run Now Operator**

from datetime import timedelta

from airflow import DAG

from airflow.operators.bash\_operator import BashOperator

#from airflow.operators.bash\_operator import BashOperator

from airflow.providers.databricks.operators.databricks import DatabricksSubmitRunOperator

#from airflow.contrib.operators.databricks\_operator import DatabricksSubmitRunOperator

# default arguments for all the tasks

default\_args = {

    'owner': 'airflow',

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

    'depends\_on\_past': False,

    'start\_date': airflow.utils.dates.days\_ago(0)

}

# create a DAG definition

dag = DAG(

'databricks\_demo',

default\_args=default\_args,

schedule\_interval=timedelta(days=1),

)

# create a simple task that prints todays date

date\_task = BashOperator(

    task\_id='print\_date',

    bash\_command='date',

    dag=dag,

)

# create a cluster config

new\_cluster\_conf = {

    'spark\_version': '10.4.x-scala2.12',

    'node\_type\_id': 'Standard\_DS3\_v2',

    'autoscale' : {

        'min\_workers': 1,

        'max\_workers': 8

    },

    'spark\_conf': {

        'spark.databricks.delta.preview.enabled': 'true',

        'spark.sql.crossJoin.enabled': 'true',

        },

        'spark\_env\_vars': {

            'PYSPARK\_PYTHON': '/databricks/python3/bin/python3'

        },

}

notebook\_task\_params = {

        'new\_cluster': new\_cluster\_conf,

        'notebook\_task': {

        'notebook\_path': '/Users/manish.mehta@tigeranalytics.com/demo-workbook',

        }

}

# create a task to run a notebook using above config

notebook\_task = DatabricksSubmitRunOperator(

            task\_id='notebook\_task',

            dag=dag,

            json=notebook\_task\_params,

            do\_xcom\_push = True

)

# set the order of tasks

date\_task >> notebook\_task