DAG: data pipeline diyebiliriz tasklardan oluşan ve tasklarının birbirine bağlı olduğu bir bütün

dag bir döngü halinde olamaz yani önce task1 sonra task2 sonra task3 sonrasında tekrardan task1 çalışmamalı

dag başlamalı ve sona ermeli

Operatör :

aslında tasklarımız bir operatördür tasklarda hangi işi yapmak istiyorsam ilgili operatoru kullanıyorum

<https://airflow.apache.org/docs/apache-airflow/stable/_api/airflow/operators/index.html> tüm operatörlere bu adresten ulaşabilirim

3 tip operatör var

1-) Action Operator: örnek olarak PythonOperator,BashOperator

PythonOperator Python fonkisyonları yazmamızı sağlar

BashOperator Bash komutları yazmamızı sağlar

2-) Transfer Operator:

Kaynak ve hedef arasında data transferi sağlar

Örneğin mysql den hive a data transferi sağlamak için

[airflow.operators.mysql\_to\_hive](https://airflow.apache.org/docs/apache-airflow/stable/_api/airflow/operators/mysql_to_hive/index.html) operatörünü kulancam

3.Sensor Operatörleri:

Sensor operatörini kullanan task bir sonraki taska geçmesi için bir şey olamsını bekler

Örneğin localimde x diznine y dosyasının gelmesini bekliyorum dosya gelince bir sonraki task e geçicek

Variable:

Airflow ui dan oluşturabilir kullanım amacı diyelimki bir çok dagde hep aynı klasörü kullanıyorum ilerde bu klasörün değişmesi gerekti ozman tüm kullandığım daglere gidip bu klasörü güncellemem gerekir

İşte bu durumda eğer variable oluştursaydım arayüzden variable ın value değerini değiştirmem yeterli olucaktı

Dag.py deki değişkenler sadece Dag.py içinde geçerli iken arayüzden oluşturduğum variable tüm

Daglerde geçerli

Xcom:

Taskler arası data alışverişi sağlar

Bir task da oluşan değeri yada oluşturduğum değeri bir sonraki task da kullanmam gerekiyorsa xcom oluşturmam gerekir

Schedule:

tasklerimizi ve dağimizi zamanlamaktan sorumlu

taskin durumuna bağlı olarak hangi taskin hangi sırayla çalışacağına karar verir

dag run u başlatan bileşen scheduler dır

executor:

Taskelrimizin nasıl çalışacağını belirtir

Tasklerimi tek makinede tek tek çalıştırmak istiyorsam Sequential Executor kullancam

Tasklerimi tek makinede paralel bir şekilde çalıştırmak istiyorsam LocalExecutor kullancam

Tasklerimi birçok makinede paralel çalıştırmak istiyorsam CeleryExecutor kullancam

Bir Airflow deploymentında neden CeleryExecutor kullanımına ihtiyaç duyulmuş olabilir? Bu executor kullanılan bir ortamda DAG geliştiricisinin özel olarak dikkat etmesi gereken husus nedir?

Local ve executorda tek makinede tasklarını çalıştırdığı için kaynak sınırlı bu da daha az task demek

Ayrıyaten tek makinede tüm airflow bileşenleri olduğu için olası makine arızasında tüm bileşenler etkilenir

Tasklerimizi tüm makineye dağıtmaya bu sayede daha fazla task execute etmeye olanak sağladığından kullanılabilir.

Bu Executoru kullanan geliştiricinin dikkat etmesi gerekenler

.queue sistemi için extarnool bir tool yüklü olması gerekir redis gibi

Varsayılan olarak gelen db sqlite bu ama bu db paralel çalışma imkanı sağlamadığında bir db ye ihtiyacım var mysql postgresql gibi bunlardan birini yüklemem lazım

Dag klasörüm,configürasyonlarım import ettiğim modüllerin hepsi clusterdaki tüm nodelarda bulunması gerekir

\* 2020-05-20 24:00 zamanında başlayacak ve her gün bir kere çalışacak, çalıştığında da HDFS'de /path/to/**<year>/<month>/<day>** adında bir klasörün olup olmadığını kontrol edecek ve eğer böyle bir klasör varsa o klasörün içine boş bir \_SEEN adında dosya koyacak (touch) bir DAG hazırlayınız. HDFS'de kontrol edilecek folder'da year, day, month kısımları DAG'in schedule zamanından (nominal time) alınmalıdır.

from airflow import DAG

from datetime import datetime,timedelta

from airflow.sensors.hdfs\_sensor import HdfsSensor

from airflow.operators.bash\_operator import BashOperator

default\_args = {

    'start\_date':datetime(2020,5,20)

}

yesterday\_year = datetime.strftime(datetime.now() - timedelta(1), '<%Y>')

yesterday\_month=datetime.strftime(datetime.now() - timedelta(1), '<%m>')

yesterday\_day=datetime.strftime(datetime.now() - timedelta(1), '<%d>')

folder='/home/train/hadoop/'+yesterday\_year+'/'+yesterday\_month+'/'+yesterday\_day

with DAG(dag\_id='quiz1',default\_args=default\_args,schedule\_interval='@daily',catchup=False) as dag:

    check\_folder=HdfsSensor(

        task\_id='check\_folder',

        hdfs\_conn\_id='hdfs\_default',

        filepath='/home/train/hadoop/'+yesterday\_year+'/'+yesterday\_month+'/'+yesterday\_day

    )

    create\_file = BashOperator(

        task\_id="crate\_file",

        bash\_command='hdfs dfs -touch'+folder+'/\_seen'

    )

    check\_folder>>create\_file