Assignment 10.1

**Problem Statement**

Implement the concept given in below blog link and share the complete steps along with

screenshots.

<https://acadgild.com/blog/oozie-job-scheduling-in-hive/>

**Solution:**

In this post, we will learn how to schedule the Hive job using Oozie. In production, where you need to run the same job for multiple times, or, you have multiple jobs that should be executed one after another, you need to schedule your job using some scheduler. There are multiple ways to automate jobs, however, here we will work with Oozie. We will begin with understanding what Oozie is and Oozie job scheduling.

Oozie, an open source Apache project is a job scheduler that manages Hadoop jobs. In short, Oozie schedules long list of works sequentially into one job.

To schedule Hive job using Oozie, you need to write a Hive-action. Your Oozie job will consist of mainly three things.

1. workflow.xml
2. job.properties
3. Hive script|

**Job.properties**

This file consists of all the variable definition that you use in your workflow.xml. Let’s say, in workflow.xml, you have mentioned a property as below:

**<name-node>${nameNode}</name-node>**

So, in your Job.properties file, you must declare $nameNode and assign the relative path.

Below are the details for Job.properties:

nameNode=hdfs://sandbox.hortonworks.com:8020

jobTracker=sandbox.hortonworks.com:8050

oozie.libpath=${nameNode}/user/oozie/share/lib/hive

oozie.wf.application.path=${nameNode}/user/${user.name}/workflows

appPath=${nameNode}/user/${user.name}/workflows

Let us understand what each of it means.

**oozie.libpath**=${nameNode}/user/oozie/share/lib/hive

Indicates the path (in hdfs) where all the respective **jars** are present.

**oozie.wf.application.path**=${nameNode}/user/${user.name}/workflows

This is the place where from your application will get the **dependent files**.

**Workflow.xml**

This is the place where you write your Oozie action. It contains all the details of files, scripts,  required to schedule and run Oozie job. As the name suggests, it is an XML file where you need to mention the details in a proper tag. Below is a sample workflow.xml for running Hive action.

**<workflow-app** name="DemoOozie" xmlns="uri:oozie:workflow:0.1">

<start to="demo-hive"/>

<**action** name="demo-hive">

<hive xmlns="**uri:oozie:hive-action:0.2**">

<job-tracker>${jobTracker}</job-tracker>

<name-node>${nameNode}</name-node>

<job-xml>${appPath}/hive-site.xml</job-xml>

<configuration>

<property>

<name>oozie.hive.defaults</name>

<value>${appPath}/hive-site.xml</value>

</property>

<property>

<name>hadoop.proxyuser.oozie.hosts</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.oozie.groups</name>

<value>\*</value>

</property>

</configuration>

<script>create\_table.hql</script>

</hive>

<ok to="end"/>

<error to="end"/>

</**action**>

<end name="end"/>

**</workflow-app>**

Now let us try to understand what exactly the content of workflow.xml means.

The first line creates a workflow app and we assign a name (according to our convenience) to recognize the job.

**<workflow-app name=”DemoOozie”>**

Indicates, we are creating a workflow app whose name is ‘DemoOozie’. All the other properties will remain inside this main tag.

**<start to=”demo-hive”/>**

**<action name=”demo-hive”>**

Quite self-explanatory are the above two tags which says, give a name to your action (here ‘demo-hive’) and when <action name> matches, start your oozie job.

**<hive xmlns=”uri:oozie:hive-action:0.2″>**

The line above is very important as, it says what kind of action you are going to run. It can be a MR action, or a Pig action, or Hive. Here we have given the name as Hive-action.

**<job-tracker>${jobTracker}</job-tracker>**

**<name-node>${nameNode}</name-node>**

**<job-xml>${appPath}/hive-site.xml</job-xml>**

All the above tags point to the variable where your job-tracker, NameNode, and Hive-site.xml is present. The exact declaration of these variables is done in Job.properties file.

**<script>create\_table.hql</script>**

You need to fill in the exact name of your script file (here, it is a Hive script file) which will be looked for and the query will get executed.

**create\_table.hql**

This is the Hive script which you want to schedule in Oozie. Quite simple and self-explanatory it is.

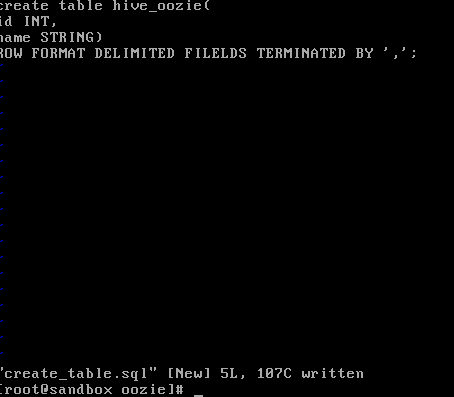
use default;

create table hive\_oozie(

id INT,

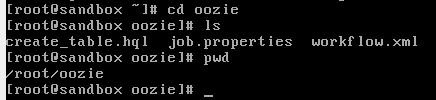
name STRING)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';



Now we will see the step by step procedure to run a Hive-Oozie job.

* Create a directory and keep the above 3 files (Job.properties, workflow.xml, and create\_table.hql) in it.



•Create a directory in HDFS by firing below command.

hadoop fs -mkdir -p /user/oozie/workflows/

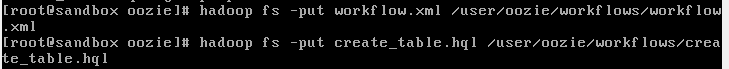
•Put workflow.xml, Hive script (create\_table.hql) and hive-site.xml in the directory created in step 2. You can use the below command.

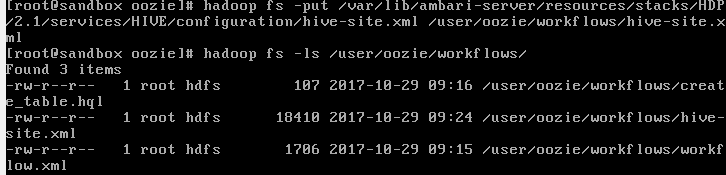
Note: Path may differ

hadoop fs -put workflow.xml /user/oozie/workflows/

hadoop fs -put create\_table.hql /user/oozie/workflows/

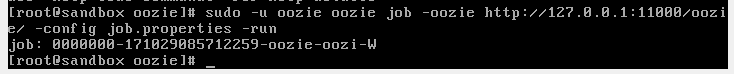
hadoop fs -put /var/lib/ambari-server/resources/stacks/HDP/2.1/services/HIVE/configuration/hive-site.xml /user/oozie/workflows/hive-site.xml

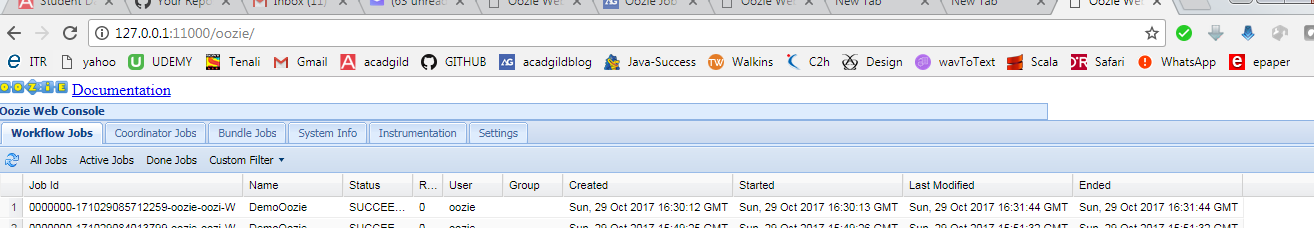
****

****

* Once done, you can run your Oozie job by using the below command.

*sudo -u oozie oozie job -oozie http://127.0.0.1:11000/oozie -config job.properties -run*

****

****