Session 10

Assignment 1

|  |  |
| --- | --- |
| **Prepared For:** | AcadGild |
|  |  |
| **Document Approval:** | **AcadGild** |
|  |  |
|  |  |
|  |  |
|  |  |
| **Project Title:** | Session 10 – Assignment 1 |
|  |  |
| **Prepared By:** | Duncan Burgess |
|  |  |
|  | dburgess@duncb.com |
|  |  |
| **Primary Engineer:** | Duncan Burgess |
|  |  |
| **Document Reference:** | **Session 10 – Assignment 1** |
|  |  |
| **Start Date:** | 05/10/2017 |
|  |  |
|  |  |

# 

# Contents

[Contents 2](#_Toc494931945)

[Change History 3](#_Toc494931946)

[1. Problem Statement 4](#_Toc494931947)

[2. How to Schedule Hive Job Using Oozie 5](#_Toc494931948)

[2.1. Job.properties 5](#_Toc494931949)

[2.2. Workflow.xml 6](#_Toc494931950)

[2.3. create\_table.hql 7](#_Toc494931951)

[2.4. Create a directory in HDFS called workflows 7](#_Toc494931952)

[2.5. Check files on HDFS 7](#_Toc494931953)

[2.6. Running the Job 7](#_Toc494931954)

[2.7. Test if job worked correctly 8](#_Toc494931955)

# Change History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Document Revision** | **Date** | **Authored By** | **Authorised By** | **Sections Affected** | **Reason for Change** |
| Rev 01 | 05/10/2017 | Duncan Burgess |  | All | Initial release. |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 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/

# How to Schedule Hive Job Using Oozie

In this assignment, 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.

* workflow.xml
* job.properties
* Hive script

Let us look at each of them individually.

**Note**: This was completed using a **Cloudera Sandbox**.

## 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://localhost:8020*

*jobTracker=localhost:8032*

*oozie.libpath=hdfs://localhost:8020/user/oozie/share/lib/hive*

*oozie.use.system.libpath=true*

*oozie.wf.application.path=hdfs://localhost:8020/user/oozie/workflows*

*ooziePath=hdfs://localhost:8020/user/oozie/workflows*

*appPath=hdfs://localhost:8020/user/oozie/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.

*<workflow-app name="HiveOozieDemo" 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>*

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 ‘HiveOozieDemo’. 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**

## create\_table.hql

*USE default;*

*CREATE TABLE HiveWithOozie(*

*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.

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

*[cloudera@quickstart stuff]$ ls*

*create\_table.hql job.properties workflow.xml*

## Create a directory in HDFS called workflows

*[cloudera@quickstart stuff]$ hadoop fs -mkdir -p /user/oozie/workflows/*

Put workflow.xml, Hive script, (create\_table.hql) and hive-site.xml in the directory workflow directory.

*[cloudera@quickstart stuff]$ hadoop fs -copyFromLocal create\_table.hql*

*[cloudera@quickstart stuff]$ hadoop fs -copyFromLocal workflow.xml*

*[cloudera@quickstart stuff]$ hadoop fs –copyFromLocal etc/alternatives/hive-conf/hive-site.xml*

## Check files on HDFS

*[cloudera@quickstart stuff]$ hadoop fs -ls /user/oozie/workflows*

*Found 3 items*

*-rw-r--r-- 1 cloudera supergroup 105 2017-10-03 07:29 /user/oozie/workflows/create\_table.hql*

*-rw-r--r-- 1 cloudera supergroup 1937 2017-09-28 03:13 /user/oozie/workflows/hive-site.xml*

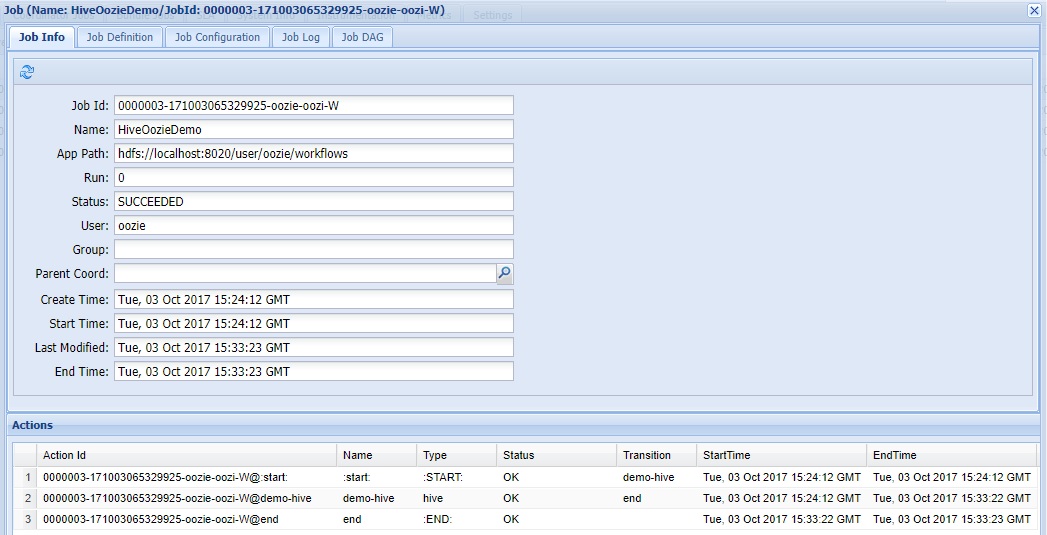
*-rw-r--r-- 1 cloudera supergroup 1325 2017-10-03 07:29 /user/oozie/workflows/workflow.xml*

## Running the Job

*[cloudera@quickstart stuff]$ sudo -u oozie oozie job -oozie http://localhost:11000/oozie -config /home/cloudera/stuff/job.properties –run*

job: 0000003-171003065329925-oozie-oozi-W

**Oozie Job**

****

## Test if job worked correctly

Run Hive an check for presence of table **hivewithoozie**

*hive> show tables;*

*OK*

*hivewithoozie*

*Time taken: 0.051 seconds, Fetched: 1 row(s)*

*hive> describe hivewithoozie;*

*OK*

*id int*

*name string*

*Time taken: 0.439 seconds, Fetched: 2 row(s)*

*hive>*