Scheduling Hive Job Using Oozie

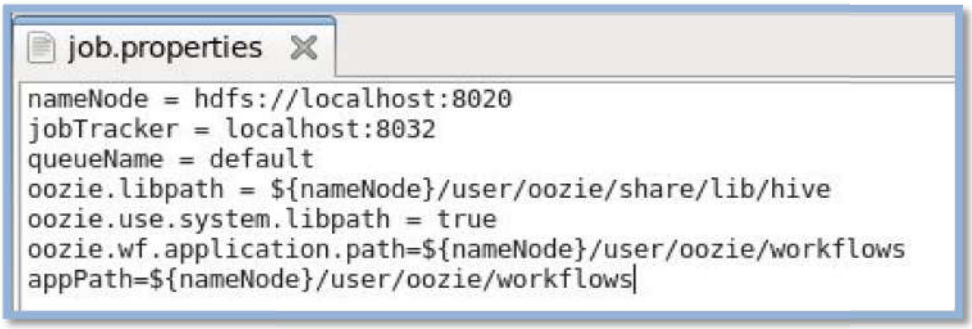
Note: Cloudera quickstart VM is used to schedule Hive job using oozie.

In Cloudera quickstart VM all daemons are started at the time when we start VM, so there is no need to start all the required daemons manually, like as in acadgild VM we start hadoop daemons with "start-all.sh" command and mysql service with "sudo service mysqld start".

To schedule Hive job using Oozie, we need to write a Hive-action. Oozie job consists of mainly three things.

- **workflow.xml**
- **job.properties**
- **hive script**

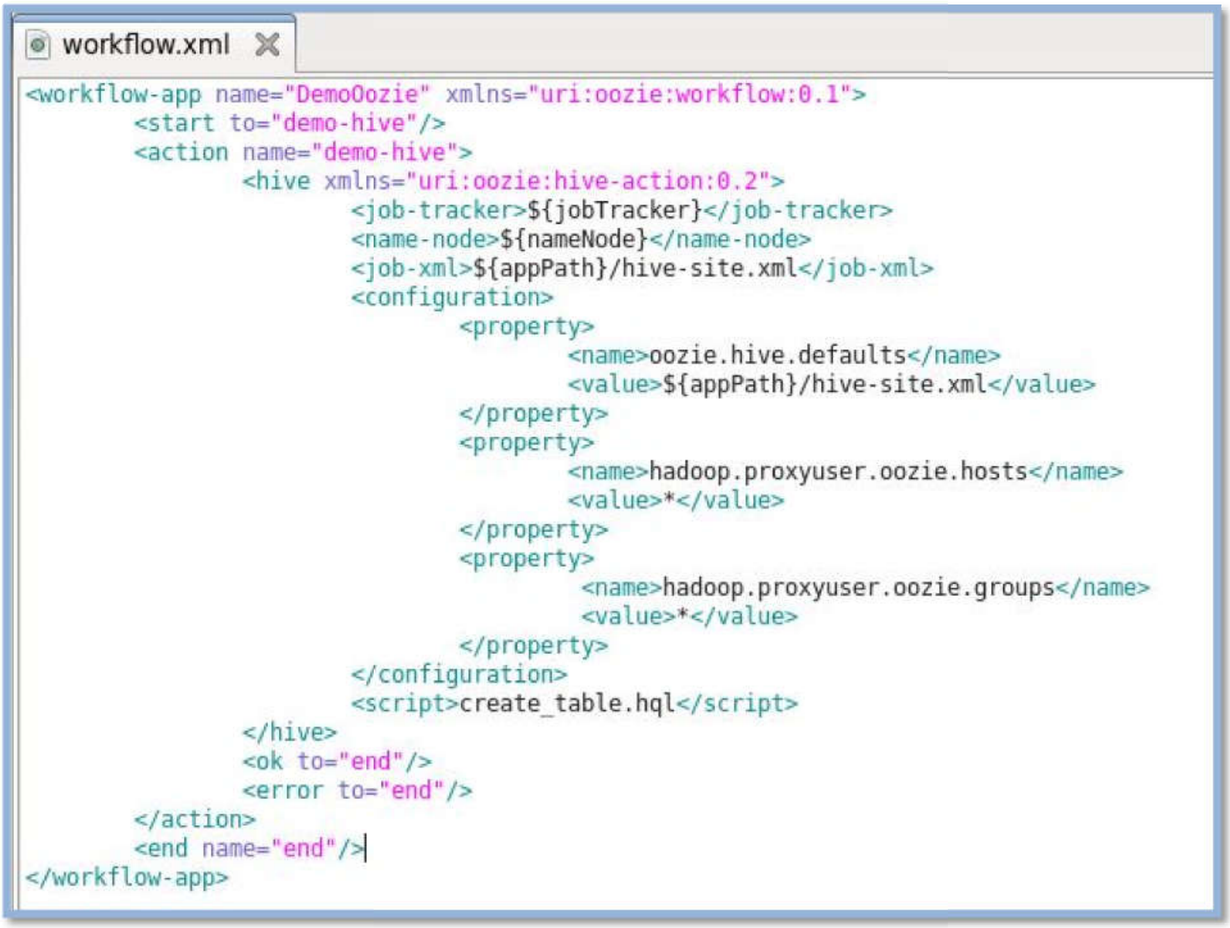
Step 1: Created job.properties file inside /home/cloudera/mydata/oozie/ directory using gedit job.properties command. This file consists of all the variables definitions that are used in workflow.xml.



```
nameNode = hdfs://localhost:8020
jobTracker = localhost:8032
queueName = default
oozie.libpath = ${nameNode}/user/oozie/share/lib/hive
oozie.use.system.libpath = true
oozie.wf.application.path=${nameNode}/user/oozie/workflows
appPath=${nameNode}/user/oozie/workflows
```

Line 1: In nameNode variable, assigning address of namenode
Line 2: In jobTracker variable, assigning address of resource manager
Line 3: queueName variable stores default value
Line 4: oozie.libpath stores that path where all hive .jar files are present
Line 5: oozie.use.system.libpath is set to true so that path specified at Line 4 is picked.
Line 6 & 7: oozie.wf.application.path and appPath store the path where all dependent files are present like workflow.xml, hive-script, hive-site.xml

Step 2: Created workflow.xml file inside /home/cloudera/mydata/oozie/ directory using gedit workflow.xml command. This is the place where we write our 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 we need to mention the details in a proper tag.



```
<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>
```

Explanation:

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”>

First tag <start to/> gives a name to hive action (i.e. ‘demo-hive’) and when <action name> matches with <start to/> then it starts oozie job.

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

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

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

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

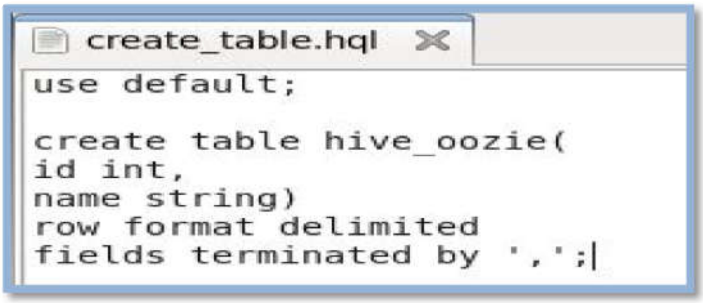
<job-xml>${appPath}/hive-site.xml</job-xml>

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

<script>create_table.hql</script>

In this tag we need to write the exact name of script file (here, it is a Hive script file i.e. create_table.hql) which will be looked for and the query will get executed.

Step 3: Created hive script i.e. create_table.hql which we want to schedule in Oozie, inside /home/cloudera/mydata/oozie/ directory using gedit workflow.xml command.



create_table.hql

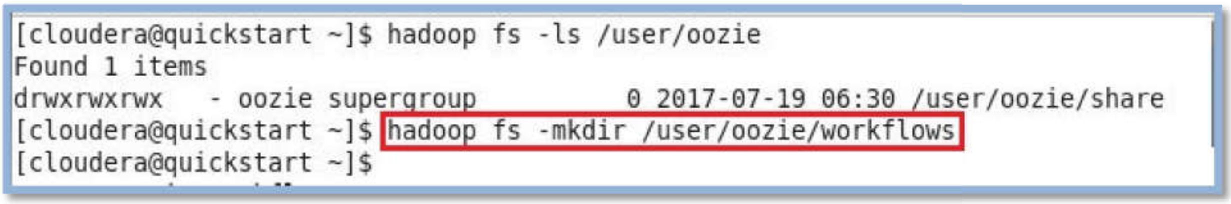
```
use default;

create table hive_oozie(
id int,
name string)
row format delimited
fields terminated by ',';
```

Explanation:

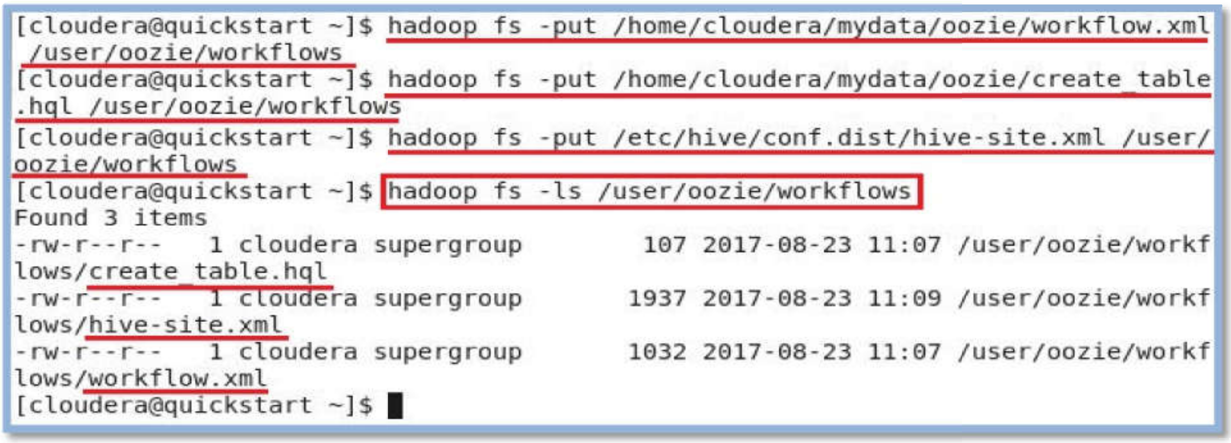
Line 1: "use default" statement allows to use default database where table will get created, and this default database is present inside /user/hive/warehouse in hdfs.

Line 2-5: These statements create table with name hive_oozie and with two columns id (type int), name (type string) and it's specified that row format is delimited where all fields are terminated by ','.

Step 4: Created workflows directory in hdfs inside /user/oozie, and have put workflow.xml, create_table.hql and hive-site.xml inside /user/oozie/workflows location using below commands

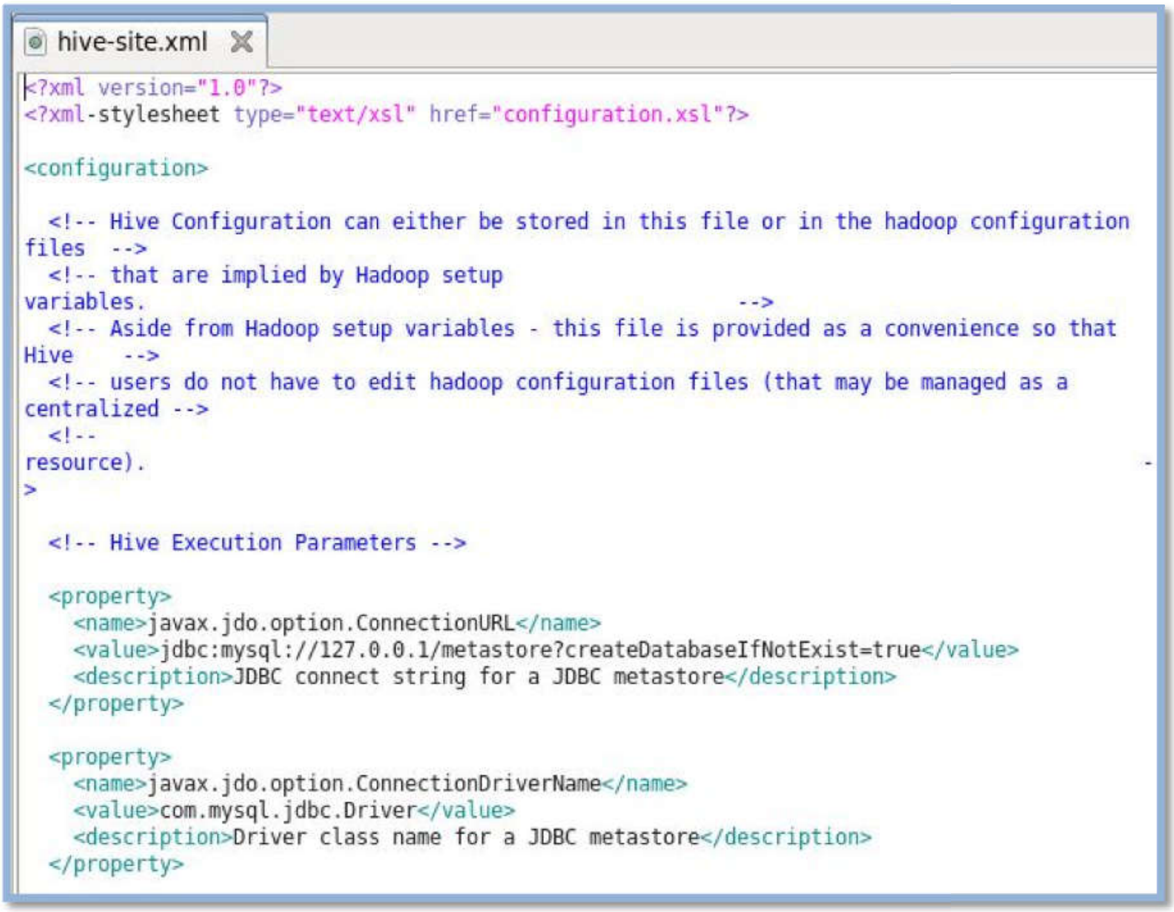


```
[cloudera@quickstart ~]$ hadoop fs -ls /user/oozie
Found 1 items
drwxrwxrwx   - oozie supergroup          0 2017-07-19 06:30 /user/oozie/share
[cloudera@quickstart ~]$ hadoop fs -mkdir /user/oozie/workflows
[cloudera@quickstart ~]$
```

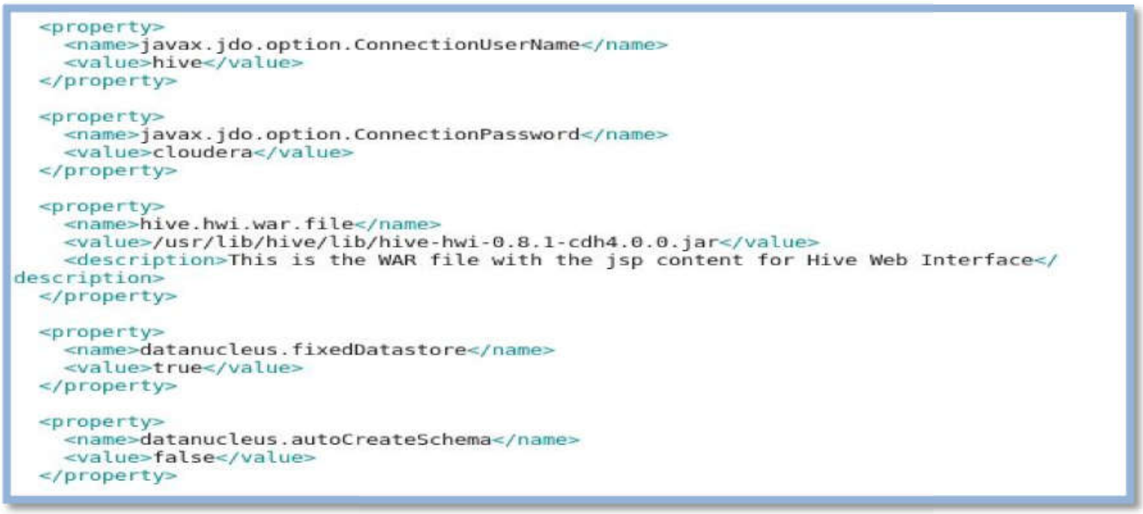


```
[cloudera@quickstart ~]$ hadoop fs -put /home/cloudera/mydata/oozie/workflow.xml
 /user/oozie/workflows
[cloudera@quickstart ~]$ hadoop fs -put /home/cloudera/mydata/oozie/create_table
.hql /user/oozie/workflows
[cloudera@quickstart ~]$ hadoop fs -put /etc/hive/conf.dist/hive-site.xml /user/
oozie/workflows
[cloudera@quickstart ~]$ hadoop fs -ls /user/oozie/workflows
Found 3 items
-rw-r--r--   1 cloudera supergroup        107 2017-08-23 11:07 /user/oozie/workf
lows/create_table.hql
-rw-r--r--   1 cloudera supergroup       1937 2017-08-23 11:09 /user/oozie/workf
lows/hive-site.xml
-rw-r--r--   1 cloudera supergroup       1032 2017-08-23 11:07 /user/oozie/workf
lows/workflow.xml
[cloudera@quickstart ~]$
```

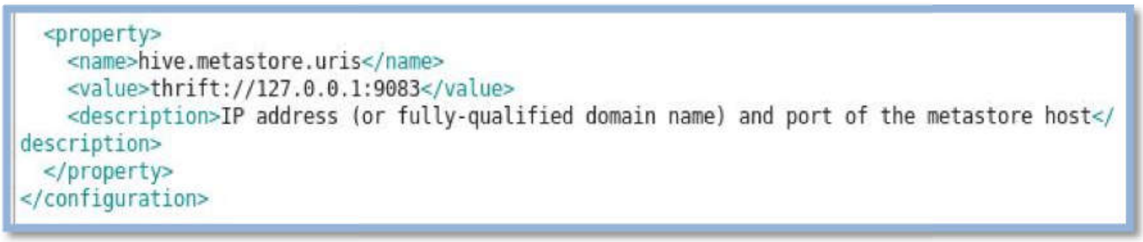
hive-site.xml looks like as below:



```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

  <!-- Hive Configuration can either be stored in this file or in the hadoop configuration
files  -->
  <!-- that are implied by Hadoop setup
variables.                                                 -->
  <!-- Aside from Hadoop setup variables - this file is provided as a convenience so that
Hive    -->
  <!-- users do not have to edit hadoop configuration files (that may be managed as a
centralized -->
  <!--
resource).                                                                                  -
>

  <!-- Hive Execution Parameters -->

  <property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://127.0.0.1/metastore?createDatabaseIfNotExist=true</value>
    <description>JDBC connect string for a JDBC metastore</description>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.jdbc.Driver</value>
    <description>Driver class name for a JDBC metastore</description>
  </property>
```

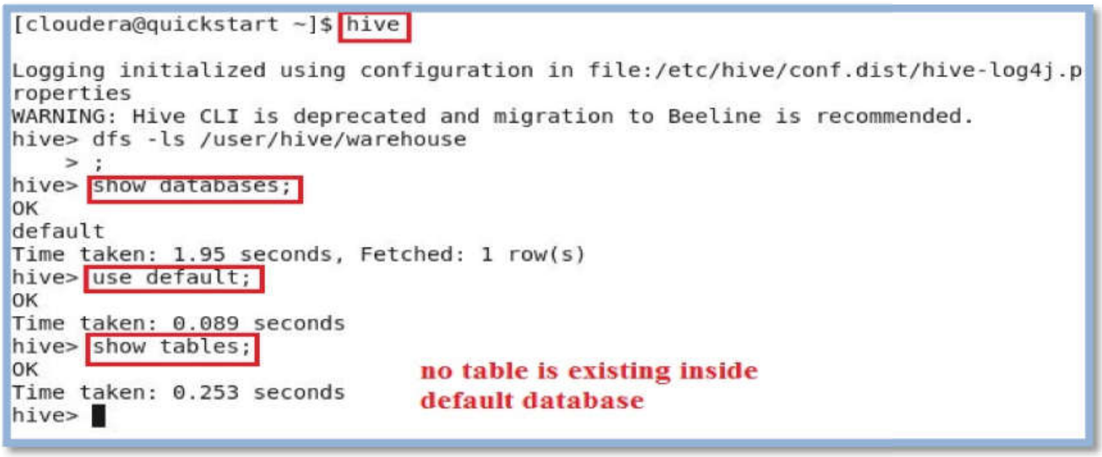


```
  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>hive</value>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>cloudera</value>
  </property>

  <property>
    <name>hive.hwi.war.file</name>
    <value>/usr/lib/hive/lib/hive-hwi-0.8.1-cdh4.0.0.jar</value>
    <description>This is the WAR file with the jsp content for Hive Web Interface</
description>
  </property>

  <property>
    <name>datanucleus.fixedDatastore</name>
    <value>true</value>
  </property>

  <property>
    <name>datanucleus.autoCreateSchema</name>
    <value>false</value>
  </property>
```



```
  <property>
    <name>hive.metastore.uris</name>
    <value>thrift://127.0.0.1:9083</value>
    <description>IP address (or fully-qualified domain name) and port of the metastore host</
description>
  </property>
</configuration>
```

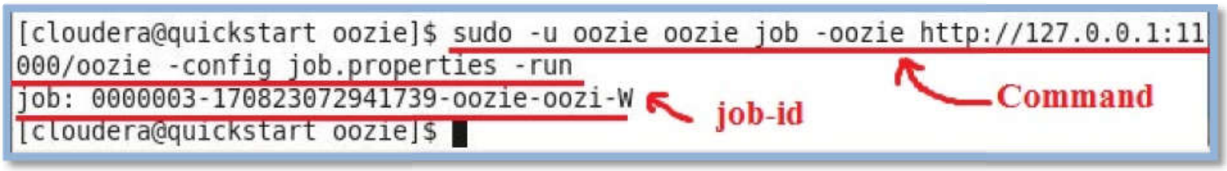
Step 5: Checked whether hive_oozie table already exists inside default database or not



```
[cloudera@quickstart ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.p
roperties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> dfs -ls /user/hive/warehouse
    > ;
hive> show databases;
OK
default
Time taken: 1.95 seconds, Fetched: 1 row(s)
hive> use default;
OK
Time taken: 0.089 seconds
hive> show tables;
OK
Time taken: 0.253 seconds
hive> █
```

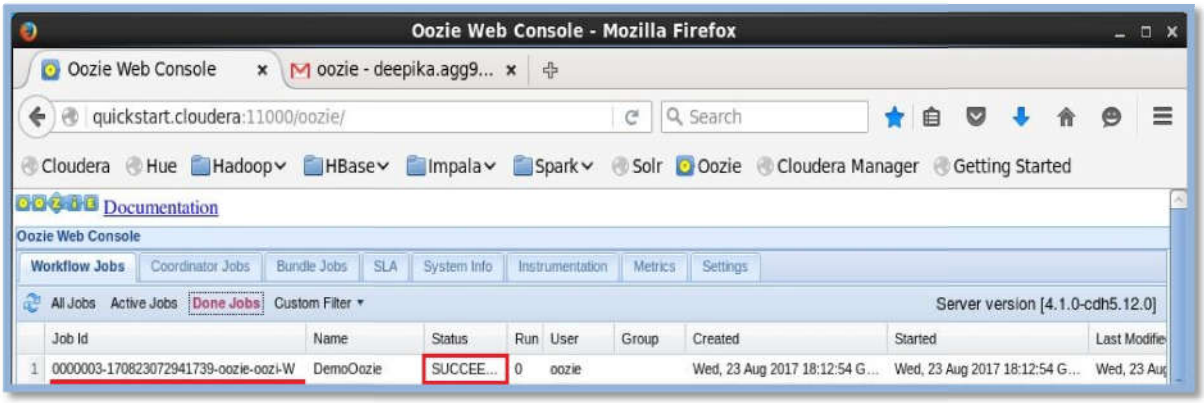
no table is existing inside
default database

Step 6: After completing above steps, ran Oozie job by using the below command.



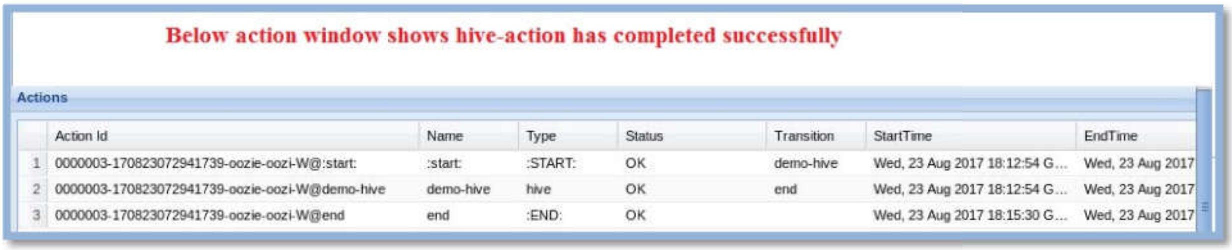
```
[cloudera@quickstart oozie]$ sudo -u oozie oozie job -oozie http://127.0.0.1:11
000/oozie -config job.properties -run
job: 0000003-170823072941739-oozie-oozi-W
[cloudera@quickstart oozie]$ 
```

After running the job, checked the status of job in Oozie web console.



Server version [4.1.0-cdh5.12.0]

	Job Id	Name	Status	Run	User	Group	Created	Started	Last Modifie
1	0000003-170823072941739-oozie-oozi-W	DemoOozie	SUCCEE...	0	oozie		Wed, 23 Aug 2017 18:12:54 G...	Wed, 23 Aug 2017 18:12:54 G...	Wed, 23 Aug

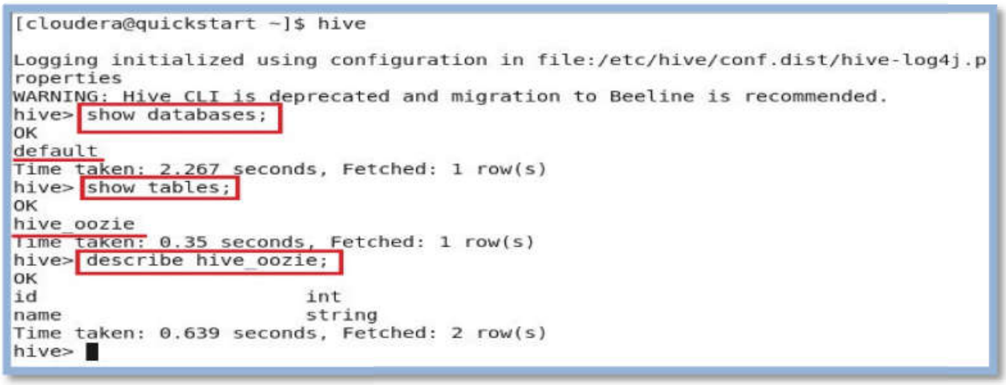


Below action window shows hive-action has completed successfully

Actions

	Action Id	Name	Type	Status	Transition	StartTime	EndTime
1	0000003-170823072941739-oozie-oozi-W@:start:	:start:	:START:	OK	demo-hive	Wed, 23 Aug 2017 18:12:54 G...	Wed, 23 Aug 2017
2	0000003-170823072941739-oozie-oozi-W@demo-hive	demo-hive	hive	OK	end	Wed, 23 Aug 2017 18:12:54 G...	Wed, 23 Aug 2017
3	0000003-170823072941739-oozie-oozi-W@end	end	:END:	OK		Wed, 23 Aug 2017 18:15:30 G...	Wed, 23 Aug 2017

Step 6: Below screenshot shows that hive_oozie table has been created successfully.



```
[cloudera@quickstart ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.p
roperties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> show databases;
OK
default
Time taken: 2.267 seconds, Fetched: 1 row(s)
hive> show tables;
OK
hive_oozie
Time taken: 0.35 seconds, Fetched: 1 row(s)
hive> describe hive_oozie;
OK
id                          int
name                        string
Time taken: 0.639 seconds, Fetched: 2 row(s)
hive> 
```