● Run a Java MapReduce wordcount Job using Oozie in HDP Sandbox.

● Explain the procedures and codes used with their importance and Screenshot.

1) Untar the jar hadoop-mapreduce-examples.jar. You can find this jar under /usr/lib/gphd/hadoop-mapreduce directory on a Pivotal Hadoop cluster.

[hadoop@hdm1 test]$ jar xf hadoop-mapreduce-examples.jar  
[hadoop@hdm1 test]$ ls  
hadoop-mapreduce-examples.jar META-INF org

2) Navigate to the directory to see the list of class files associated with WordCount.

[hadoop@hdm1 test]$ cd org/apache/hadoop/examples/  
[hadoop@hdm1 examples]$ ls WordCount\*  
WordCount.class WordCount$IntSumReducer.class WordCount$TokenizerMapper.class

In the WordCount program, name of the mapper class is WordCount$TokenizerMapper.class and reducer class is WordCount$TokenizerMapper.class. We will use these file when defining the oozie workflow.xml

3) Create a job.properties file. The parameters for the Oozie job are provided in a Java properties file (.properties) or a Hadoop configuration xml (.xml), in this situation we use a .properties file.

nameNode=hdfs://phdha  
jobTracker=hdm1.phd.local:8032  
queueName=default  
examplesRoot=examplesoozie

oozie.wf.application.path=${nameNode}/user/${user.name}/${examplesRoot}/map-reduce  
outputDir=map-reduce  
  
where:  
namenode = Variable to define the namenode path by which HDFS can be accessed. Format: hdfs://<nameservice> or hdfs://<namenode\_host>:<port>  
jobTracker = Variable to define the resource manager address in case of Yarn implementation. Format: <resourcemanager\_hostname>:<port>  
queueName = Name of the queue as defined by Capacity Scheduler, Fail Scheduler etc. By default, it's "default".  
examplesRoot = Environment variable for the workflow.  
oozie.wf.application.path = Environment variable which defines the path on HDFS which holds the workflow.xml to be executed.  
outputDir = Variable to define the output directory

Note: You can define the parameter, oozie.libpath under which all the libraries required for the MapReduce program can be stored. However, this is not applied in this example.

Example:

oozie.libpath=${nameNode}/$(user.name)/share/lib

4) Create a workflow.xml. workflow.xml defines a set of actions to be performed as a sequence or in Control Dependency DAG (Direct Acyclic Graph).

"control dependency" from one action to another means that the second action cannot run until the first action has been completed.

Refer to the documentation:

<http://oozie.apache.org/docs/3.3.2/WorkflowFunctionalSpec.html>.

<workflow-app xmlns="uri:oozie:workflow:0.1" name="map-reduce-wf">  
 <start to="mr-node"/>  
 <action name="mr-node">  
 <map-reduce>  
 <job-tracker>${jobTracker}</job-tracker>  
 <name-node>${nameNode}</name-node>  
 <prepare>  
 <delete path="${nameNode}/user/${wf:user()}/${examplesRoot}/output-data/${outputDir}"/>  
 </prepare>  
   
 <configuration>  
 <property>  
 <name>mapred.mapper.new-api</name>  
 <value>true</value>  
 </property>  
 <property>  
 <name>mapred.reducer.new-api</name>  
 <value>true</value>  
 </property>  
 <property>  
 <name>mapred.job.queue.name</name>  
 <value>${queueName}</value>  
 </property>  
 <property>  
 <name>mapreduce.map.class</name>  
 <value>org.apache.hadoop.examples.WordCount$TokenizerMapper</value>  
 </property>  
 <property>  
 <name>mapreduce.reduce.class</name>  
 <value>org.apache.hadoop.examples.WordCount$IntSumReducer</value>  
 </property>  
 <property>  
 <name>mapreduce.combine.class</name>  
 <value>org.apache.hadoop.examples.WordCount$IntSumReducer</value>  
 </property>  
 <property>  
 <name>mapred.output.key.class</name>  
 <value>org.apache.hadoop.io.Text</value>  
 </property>  
 <property>  
 <name>mapred.output.value.class</name>  
 <value>org.apache.hadoop.io.IntWritable</value>  
 </property>  
 <property>  
 <name>mapred.input.dir</name>  
 <value>/user/${wf:user()}/${examplesRoot}/input-data/text</value>  
 </property>  
 <property>  
 <name>mapred.output.dir</name>  
 <value>/user/${wf:user()}/${examplesRoot}/output-data/${outputDir}</value>  
 </property>  
 </configuration>  
 </map-reduce>  
 <ok to="end"/>  
 <error to="fail"/>  
 </action>  
 <kill name="fail">  
 <message>Map/Reduce failed, error message[${wf:errorMessage(wf:lastErrorNode())}]</message>  
 </kill>  
 <end name="end"/>  
</workflow-app>

5. Create a directory on HDFS under which all the files related to the Oozie job will be kept. In this directory, push the workflow.xml created in the previous step.

[hadoop@hdm1 map-reduce]$ hdfs dfs -mkdir -p /user/hadoop/examplesoozie/map-reduce

[hadoop@hdm1 map-reduce]$ hdfs dfs -copyFromLocal workflow.xml /user/hadoop/examplesoozie/map-reduce/workflow.xml

6. Now under the directory created for the Oozie job, create a folder named lib in which the required library / jar files are kept.

[hadoop@hdm1 map-reduce]$ hdfs dfs -mkdir -p /user/hadoop/examplesoozie/map-reduce/lib

7. Once the directory is created, copy Hadoop MapReduce examples jar under this directory.

[hadoop@hdm1 map-reduce]$ hdfs dfs -copyFromLocal /usr/lib/gphd/hadoop-mapreduce/hadoop-mapreduce-examples.jar /user/hadoop/examplesoozie/map-reduce/lib/hadoop-mapreduce-examples.jar

8. Now you can execute the workflow created, and use it to run Hadoop MapReduce program for WordCount

[hadoop@hdm1 ~]$ oozie job -oozie http://localhost:11000/oozie -config examplesoozie/map-reduce/job.properties -run

9. You can view the status of the job as shown below:

[hadoop@hdm1 ~]$ oozie job -oozie http://localhost:11000/oozie -info 0000009-140529162032574-oozie-oozi-W

Job ID : 0000009-140529162032574-oozie-oozi-W

------------------------------------------------------------------------------------------------------------------------------------

Workflow Name : map-reduce-wf

App Path : hdfs://phdha/user/hadoop/examplesoozie/map-reduce

Status : SUCCEEDED

Run : 0

User : hadoop

Group : -

Created : 2014-05-30 00:31 GMT

Started : 2014-05-30 00:31 GMT

Last Modified : 2014-05-30 00:32 GMT

Ended : 2014-05-30 00:32 GMT

CoordAction ID: -

Actions

------------------------------------------------------------------------------------------------------------------------------------

ID Status Ext ID Ext Status Err Code

------------------------------------------------------------------------------------------------------------------------------------

0000009-140529162032574-oozie-oozi-W@:start: OK - OK -

------------------------------------------------------------------------------------------------------------------------------------

0000009-140529162032574-oozie-oozi-W@mr-node OK job\_1401405229971\_0022 SUCCEEDED -

------------------------------------------------------------------------------------------------------------------------------------

0000009-140529162032574-oozie-oozi-W@end OK - OK -

------------------------------------------------------------------------------------------------------------------------------------

10. Once the job is completed, you can review the output in the directory as specified by workflow.xml.

[hadoop@hdm1 ~]$ hdfs dfs -cat /user/hadoop/examplesoozie/output-data/map-reduce/part-r-00000

SSH:/var/empty/sshd:/sbin/nologin 1

Server:/var/lib/pgsql:/bin/bash 1

User:/var/ftp:/sbin/nologin 1

Yarn:/home/yarn:/sbin/nologin 1

adm:x:3:4:adm:/var/adm:/sbin/nologin 1

bin:x:1:1:bin:/bin:/sbin/nologin 1

console 1

daemon:x:2:2:daemon:/sbin:/sbin/nologin 1

ftp:x:14:50:FTP 1

games:x:12:100:games:/usr/games:/sbin/nologin 1

gopher:x:13:30:gopher:/var/gopher:/sbin/nologin 1

gpadmin:x:500:500::/home/gpadmin:/bin/bash 1

hadoop:x:503:501:Hadoop:/home/hadoop:/bin/bash 1

halt:x:7:0:halt:/sbin:/sbin/halt 1