

How to schedule jobs in Oozie

Prepared by,

Rajasekhar Reddy Kodur

Datalake Developer

Table Of contents:

Workflow Job:

- Shell Action
- Sqoop Action
- Spark Action

Coordinator Job (Croning Workflow job):

- Scheduling Shell Action
- Scheduling Sqoop Action
- Scheduling Spark Action

Oozie Server Details: [Test]

Edge Node = APSRT3197

\$OOZIE_HOME = /opt/mapr/oozie/oozie-4.3.0

GROP ACCESS = DAPUSERS_DEV

WEB URL = <https://apsrt3197:11443/oozie/>

Job Id	Name	Status	R...	User	Group	Created	Started	Last Modified	Ended
1 0000473-171026154810092-oozie-ma...	simple-Workflow	KILLED	0	sjarar10		Tue, 14 Nov 2017 17:18:22 ...	Tue, 14 Nov 2017 17:18:22 ...	Tue, 14 Nov 2017 17:33:39 ...	Tue, 14 Nov
2 0000472-171026154810092-oozie-ma...	hive-wf	KILLED	0	agarg107		Tue, 14 Nov 2017 16:42:09 ...	Tue, 14 Nov 2017 16:42:09 ...	Tue, 14 Nov 2017 17:33:38 ...	Tue, 14 Nov
3 0000471-171026154810092-oozie-ma...	hive-wf	KILLED	0	agarg107		Tue, 14 Nov 2017 16:37:56 ...	Tue, 14 Nov 2017 16:37:56 ...	Tue, 14 Nov 2017 16:38:08 ...	Tue, 14 Nov
4 0000470-171026154810092-oozie-ma...	hive-wf	KILLED	0	agarg107		Tue, 14 Nov 2017 16:32:14 ...	Tue, 14 Nov 2017 16:32:14 ...	Tue, 14 Nov 2017 16:32:42 ...	Tue, 14 Nov
5 0000469-171026154810092-oozie-ma...	hive-wf	SUCCEE...	0	agarg107		Tue, 14 Nov 2017 16:27:15 ...	Tue, 14 Nov 2017 16:27:15 ...	Tue, 14 Nov 2017 16:27:31 ...	Tue, 14 Nov
6 0000468-171026154810092-oozie-ma...	hive-wf	KILLED	0	agarg107		Tue, 14 Nov 2017 16:25:35 ...	Tue, 14 Nov 2017 16:25:35 ...	Tue, 14 Nov 2017 16:26:03 ...	Tue, 14 Nov
7 0000467-171026154810092-oozie-ma...	hive-wf	KILLED	0	agarg107		Tue, 14 Nov 2017 16:13:14 ...	Tue, 14 Nov 2017 16:13:14 ...	Tue, 14 Nov 2017 16:13:41 ...	Tue, 14 Nov
8 0000466-171026154810092-oozie-ma...	hive-wf	KILLED	0	agarg107		Tue, 14 Nov 2017 16:09:23 ...	Tue, 14 Nov 2017 16:09:23 ...	Tue, 14 Nov 2017 16:09:23 ...	Tue, 14 Nov

Workflow Jobs:

Workflow job requires two files to submit Oozie Shell Action:

- **Workflow.xml**

A workflow definition is a DAG (directed acyclic graph) with control flow nodes (start, end, decision, fork, join, kill) or action nodes (shell, spark and Sqoop etc.). Nodes are connected by transitions arrows.

- **Job.properties**

The parameters for the job must be provided in a file, either a Java Properties file (.properties) or a Hadoop XML Configuration file (.xml). This file must be specified with the -config option in Oozie. "job.properties" file contains all the parameters required for workflow Job.

1) Shell Action:

Steps to create and submit Simple Workflow Job for Shell Action:

➤ Create a directory for resources (workflow.xml, job.properties) , let's say "workflow".

```
/mapr/datalake/uhclake/tst/developer/rkodurl/oozietst/shell/workflow
/Workflow.xml
/Job.properties
/Test.sh
```

- Create sample bash shell script logic on “Test.sh” to create empty file with timestamp.

```
#!/bin/sh
now=$(date +%s")
filename="my_program.$now.log"
touch      /mapr/datalake/uhclake/tst/developer/rkodurl/oozietst/sqoop/workflow
/$filename
```

- Create “job.properties” which we can pass config/ parameters for workflow Job

```
nameNode=maprfs:///
jobTracker=dbslld0069:8032
queueName=fabbddev_q1
examplesRoot=oozietst
oozie.wf.application.path=${nameNode}/datalake/uhclake/tst/developer/${
user.name}/${examplesRoot}/shell/workflow
myscript=test.sh
myscriptPath=${oozie.wf.application.path}/test.sh#test.sh
```

- Create “workflow.xml” with Shell action

```
<workflow-app xmlns="uri:oozie:workflow:0.4" name="rkodur-shell-wf">
  <start to="shell-node"/>
  <action name="shell-node">
    <shell xmlns="uri:oozie:shell-action:0.2">
      <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>
      <configuration>
        <property>
          <name>mapred.job.queue.name</name>
          <value>${queueName}</value>
        </property>
      </configuration>
      <exec>${myscript}</exec>
      <file>${myscriptPath}</file>
      <capture-output/>
    </shell>
    <ok to="end"/>
    <error to="fail"/>
  </action>
  <kill name="fail">
    <message>Shell action failed, error
message[${wf:errorMessage(wf:lastErrorNode())}]</message>
  </kill>
  <end name="end"/>
</workflow-app>
```

- Submit workflow job to Oozie, by specifying “job.properties” to Oozie with -config option. Then, you will be getting workflow job ID:

```
/opt/mapr/oozie/oozie-4.3.0/bin/oozie job -oozie="https://apsrt3197.uhc.com:11443/oozie" -config
/mapr/datalake/uhclake/tst/developer/rkodurl/oozietst/shell/workflow/job.properties -run
```

- We can check Workflow Job info by executing below command with the workflow job id received from above submission

UHG Datalake

```
/opt/mapr/oozie/oozie-4.3.0/bin/oozie job -oozie="https://apsrt3197.uhc.com:11443/oozie" -info 0000393-171026154810092-oozie-mapr-W
```

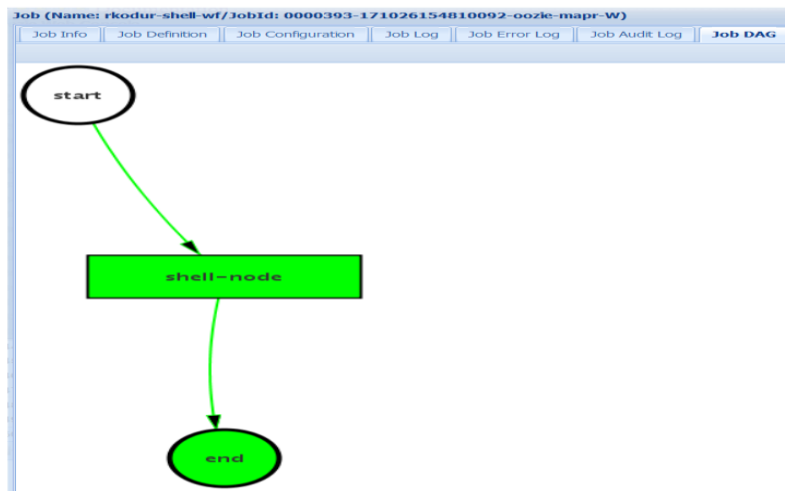
```
$ /opt/mapr/oozie/oozie-4.3.0/bin/oozie job -oozie="https://apsrt3197.uhc.com:11443/oozie" -info 0000393-171026154810092-oozie-mapr-W
Job ID : 0000393-171026154810092-oozie-mapr-W
-----
Workflow Name : rkodur-shell-wf
App Path      : maprfs:///datalake/uhclake/tst/developer/rkodur1/oozietst/shell/workflow
Status        : SUCCEEDED
Run           : 0
User          : rkodur1
Group         : -
Created       : 2017-11-13 21:21 GMT
Started       : 2017-11-13 21:21 GMT
Last Modified : 2017-11-13 21:21 GMT
Ended        : 2017-11-13 21:21 GMT
CoordAction ID: -

Actions
-----
ID                                     Status  Ext ID                               Ext Status Err Code
-----
0000393-171026154810092-oozie-mapr-W@start: OK      -                                     OK          -
0000393-171026154810092-oozie-mapr-W@shell-node OK      job_1508958967521_690352SUCCEEDED -
0000393-171026154810092-oozie-mapr-W@end      OK      -                                     OK          -
-----
```

DAG Visualization:

Go to URL <https://apsrt3197:11443/oozie/> Check Job Workflow ID

Job (Name: rkodur-shell-wf/JobId: 0000393-171026154810092-oozie-mapr-W)							
Job Info Job Definition Job Configuration Job Log Job Error Log Job Audit Log Job DAG							
Actions							
Action Id	Name	Type	Status	Transition	StartTime	EndTime	
1 0000393-171026154810092-oozie-mapr-W@start:	:start:	:START:	OK	shell-node	Mon, 13 Nov 2017 21:21:27 ...	Mon, 13 Nov 2017 21:21:27 ...	
2 0000393-171026154810092-oozie-mapr-W@shell-node	shell-node	shell	OK	end	Mon, 13 Nov 2017 21:21:27 ...	Mon, 13 Nov 2017 21:21:41 ...	
3 0000393-171026154810092-oozie-mapr-W@end	end	:END:	OK		Mon, 13 Nov 2017 21:21:42 ...	Mon, 13 Nov 2017 21:21:42 ...	



2) Sqoop Action

Steps to create and submit Simple Workflow Job for Sqoop Action:

- Create a directory for resources (workflow.xml, job.properties), let's say "workflow".

```
/mapr/datalake/uhclake/tst/developer/rkodur1/oozietst/sqoop/workflow/
```

UHG Datalake

Workflow.xml

Job.properties {Properties for Oozie Workflow}

db.hsqldb.properties {Properties for the Action [Sqoop]}

db.hsqldb.script

- Hsql db is kind of derby database act like source RDBMS system for Sqoop.
- Create "job.properties" which we can pass config/ parameters for workflow.xml

```
nameNode=maprfs:///
jobTracker=dbslld0069:8032
queueName=fabbddev_q1
examplesRoot=oozietst
oozie.libpath=/oozie/share/lib
oozie.use.system.libpath=true
jobOutput=${nameNode}/datalake/uhclake/tst/developer/${user.name}/${examplesRoot}/sqoop/workflow
oozie.wf.application.path=${nameNode}/datalake/uhclake/tst/developer/${user.name}/${examplesRoot}/sqoop/workflow
```

- Create "workflow.xml" with Sqoop action

```
<workflow-app xmlns="uri:oozie:workflow:0.4" name="sqoop-wf">
  <start to="sqoop-node"/>
  <action name="sqoop-node">
    <sqoop xmlns="uri:oozie:sqoop-action:0.4">
      <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>
      <prepare>
        <delete path="${jobOutput}/output-data_sqoop"/>
      </prepare>
      <configuration>
        <property>
          <name>mapred.job.queue.name</name>
          <value>${queueName}</value>
        </property>
      </configuration>
      <command>import --connect jdbc:hsqldb:file:db.hsqldb --table TT --target-dir ${jobOutput}/output-data_sqoop -m 1</command>
      <file>db.hsqldb.properties#db.hsqldb.properties</file>
      <file>db.hsqldb.script#db.hsqldb.script</file>
    </sqoop>
    <ok to="end"/>
    <error to="fail"/>
  </action>
  <kill name="fail">
    <message>Sqoop failed, error
message[${wf:errorMessage(wf:lastErrorNode())}]</message>
  </kill>
</end name="end"/>
```

UHG Datalake

</workflow-app>

- Submit workflow job to Oozie, by specifying “job.properties” to Oozie with `-config` option. Then, you will be getting workflow job ID:

```
/opt/mapr/oozie/oozie-4.3.0/bin/oozie job -oozie="https://apsrt3197.uhc.com:11443/oozie" -config /mapr/datalake/uhclake/tst/developer/rkodurl/oozietst/sqoop/workflow/job.properties -run
```

- We can check Workflow Job info by executing below command with the workflow job id received from above submission

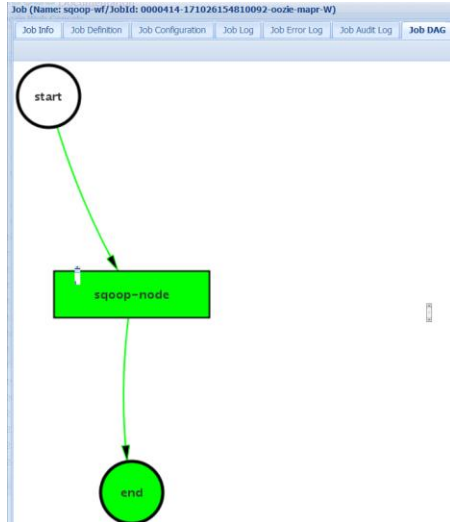
```
/opt/mapr/oozie/oozie-4.3.0/bin/oozie job -oozie="https://apsrt3197.uhc.com:11443/oozie" -info 0000414-171026154810092-oozie-mapr-W
```

```
$ /opt/mapr/oozie/oozie-4.3.0/bin/oozie job -oozie="https://apsrt3197.uhc.com:11443/oozie" -info 0000414-171026154810092-oozie-mapr-W
Job ID : 0000414-171026154810092-oozie-mapr-W
-----
Workflow Name : sqoop-wf
App Path : maprfs:///datalake/uhclake/tst/developer/rkodurl/oozietst/sqoop/workflow
Status : SUCCEEDED
Run : 0
User : rkodurl
Group : -
Created : 2017-11-14 02:23 GMT
Started : 2017-11-14 02:23 GMT
Last Modified : 2017-11-14 02:24 GMT
Ended : 2017-11-14 02:24 GMT
CoordAction ID: -
-----
Actions
-----
ID Status Ext ID Ext Status Err Code
-----
0000414-171026154810092-oozie-mapr-W@start: OK - OK -
0000414-171026154810092-oozie-mapr-W@sqoop-node OK job_1508958967521_694686SUCCEEDED -
0000414-171026154810092-oozie-mapr-W@end OK - OK -
-----
```

DAG Visualization:

Go to URL <https://apsrt3197:11443/oozie/> Check Job Workflow ID

Job (Name: sqoop-wf/JobId: 0000414-171026154810092-oozie-mapr-W)							
Job Info Job Definition Job Configuration Job Log Job Error Log Job Audit Log Job DAG							
Actions							
Action Id	Name	Type	Status	Transition	StartTime	EndTime	
1 0000414-171026154810092-oozie-mapr-W@start:	:start:	:START:	OK	sqoop-node	Tue, 14 Nov 2017 02:23:35 ...	Tue, 14 Nov 2017 02:23:35 ...	
2 0000414-171026154810092-oozie-mapr-W@sqoop-no...	sqoop-node	sqoop	OK	end	Tue, 14 Nov 2017 02:23:35 ...	Tue, 14 Nov 2017 02:24:11 G...	
3 0000414-171026154810092-oozie-mapr-W@end	end	:END:	OK		Tue, 14 Nov 2017 02:24:11 G...	Tue, 14 Nov 2017 02:24:11 G...	



3) Spark Action:

Steps to create and submit Simple Workflow Job for Spark Action:

- Create a directory for resources (workflow.xml, job.properties, lib directory), let's say "workflow".

/mapr/datalake/uhclake/tst/developer/rkodur1/oozietst/sqoop/workflow/

Workflow.xml

Job.properties

/lib/SparkApp.jar

- Create a lib directory for the Spark Application jar to trigger via Oozie.
- Create "job.properties" which we can pass config/ parameters for workflow.xml

```

nameNode=maprfs:///
jobTracker=dbsld0069.uhc.com
master=local[*]
queueName=fabbddev_q1
examplesRoot=oozietst
oozie.libpath=/oozie/share/lib
oozie.use.system.libpath=true
inputDir=${nameNode}/datalake/uhclake/tst/developer/${user.name}/${examplesRoot}/spark/workflow/text/data.txt
outputDir=${nameNode}/datalake/uhclake/tst/developer/${user.name}/${examplesRoot}/spark/workflow/sparkOut
sparkApp=${nameNode}/datalake/uhclake/tst/developer/${user.name}/${examplesRoot}/spark/workflow/lib/oozie-examples.jar
oozie.wf.application.path=${nameNode}/datalake/uhclake/tst/developer/${user.name}/${examplesRoot}/spark/workflow
  
```


- Create “workflow.xml” with Spark action

```
<workflow-app xmlns='uri:oozie:workflow:0.5' name='SparkFileCopy'>
  <start to='spark-node' />
  <action name='spark-node'>
    <spark xmlns="uri:oozie:spark-action:0.1">
      <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>
      <prepare>
        <delete path="${outputDir}"/>
      </prepare>
      <configuration>
        <property>
          <name>mapreduce.job.queueName</name>
          <value>${queueName}</value>
        </property>
      </configuration>
      <master>${master}</master>
      <name>Spark-FileCopy</name>
      <class>org.apache.oozie.example.SparkFileCopy</class>
      <jar>${sparkApp}</jar>
      <arg>${inputDir}</arg>
      <arg>${outputDir}</arg>
    </spark>
    <ok to="end" />
    <error to="fail" />
  </action>

  <kill name="fail">
    <message>Workflow failed, error
      message[${wf:errorMessage(wf:lastErrorNode())}]
    </message>
  </kill>
  <end name='end' />
</workflow-app>
```

- Submit workflow job to Oozie, by specifying “job.properties” to Oozie with *-config* option. Then, you will be getting workflow job ID:

```
/opt/mapr/oozie/oozie-4.3.0/bin/oozie job -oozie="https://apsrt3197.uhc.com:11443/oozie" -
config /mapr/datalake/uhclake/tst/developer/rkodurl/oozietst/spark/workflow/job.properties -
run
job: 0000418-171026154810092-oozie-mapr-W
```

- We can check Workflow Job info by executing below command with the workflow job id received from above submission

```
/opt/mapr/oozie/oozie-4.3.0/bin/oozie job -oozie="https://apsrt3197.uhc.com:11443/oozie" -info
0000418-171026154810092-oozie-mapr-W
```

UHG Datalake

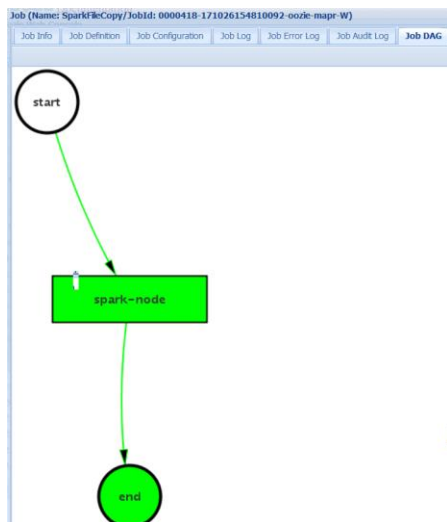
```
$ /opt/mapr/oozie/oozie-4.3.0/bin/oozie job -oozie="https://apsrt3197.uhc.com:11443/oozie" -info 0000418-171026154810092-oozie-mapr-W
Job ID : 0000418-171026154810092-oozie-mapr-W
-----
Workflow Name : SparkFileCopy
App Path : maprfs:///datalake/uhclake/tst/developer/rkodur1/oozietst/spark/workflow
Status : SUCCEEDED
Run : 0
User : rkodur1
Group : -
Created : 2017-11-14 03:05 GMT
Started : 2017-11-14 03:05 GMT
Last Modified : 2017-11-14 03:06 GMT
Ended : 2017-11-14 03:06 GMT
CoordAction ID: -

Actions
-----
ID Status Ext ID Ext Status Err Code
-----
0000418-171026154810092-oozie-mapr-W@start: OK - OK -
0000418-171026154810092-oozie-mapr-W@spark-node OK job_1508958967521_695512SUCCEEDED -
0000418-171026154810092-oozie-mapr-W@end OK - OK -
```

DAG Visualization:

Go to URL <https://apsrt3197:11443/oozie/> Check Job Workflow ID

Job (Name: SparkFileCopy/JobId: 0000418-171026154810092-oozie-mapr-W)							
Job Info Job Definition Job Configuration Job Log Job Error Log Job Audit Log Job DAG							
Actions							
Action Id	Name	Type	Status	Transition	StartTime	EndTime	
1 0000418-171026154810092-oozie-mapr-W@start:	:start:	:START:	OK	spark-node	Tue, 14 Nov 2017 03:05:57 ...	Tue, 14 Nov 2017 03:05:57 ...	
2 0000418-171026154810092-oozie-mapr-W@spark-node	spark-node	spark	OK	end	Tue, 14 Nov 2017 03:05:57 ...	Tue, 14 Nov 2017 03:06:20 ...	
3 0000418-171026154810092-oozie-mapr-W@end	end	:END:	OK		Tue, 14 Nov 2017 03:06:20 ...	Tue, 14 Nov 2017 03:06:20 ...	



Coordinator Job (Croning Workflow job):

Coordinator job allow users to schedule complex workflows, including workflows that are scheduled regularly. Oozie Coordinator models the workflow execution triggers in the form of

time, data or **event** predicates. The workflow job mentioned inside the **Coordinator** is started only after the given conditions are satisfied.

Coordinator job requires three files to Cron Oozie Shell Action:

- **Coordinator.xml**

Definition of coordinator job is defined in this file. Based on what trigger (time based or input based) your workflow will start; and, how long it will continue, workflow wait time - all of this information need to be written on this coordinator.xml file.

- **Coord.properties**

The parameters for the job must be provided in a file, either a Java Properties file (.properties) or a Hadoop XML Configuration file (.xml). This file must be specified with the -config option in Oozie. "coord.properties" file contains all the parameters required for coordinator Job

- **Workflow.xml**

A workflow definition is a DAG with control flow nodes (start, end, decision, fork, join, kill) or action nodes (shell, spark and Sqoop etc.), nodes are connected by transitions arrows.

Frequencies can be expressed using EL [Expression Language] constants and functions that evaluate to a positive integer number in coordinator.xml

EL Constant	Value	Example
<code>\${coord:minutes(int n)}</code>	<i>n</i>	<code>\${coord:minutes(45)} --> 45</code>
<code>\${coord:hours(int n)}</code>	<i>n * 60</i>	<code>\${coord:hours(3)} --> 180</code>
<code>\${coord:days(int n)}</code>	<i>variable</i>	<code>\${coord:days(2)} --> minutes in 2 full days from the current date</code>
<code>\${coord:months(int n)}</code>	<i>variable</i>	<code>\${coord:months(1)} --> minutes in a 1 full month from the current date</code>

1) Scheduling Shell Actions:

Steps to create and submit Simple Cron Job for Shell Action:

- Create a directory for resources (Coordinator.xml, workflow.xml, coord.properties, shell script) to place, let's say "coordinator".

```
/mapr/datalake/uyclake/tst/developer/rkodur1/oozietst/shell/coordinator/
                                                                    coordinator.xml
                                                                    workflow.xml
                                                                    coord.properties
                                                                    test.sh
```

- Create sample bash shell script logic on “Test.sh” to create empty file with timestamp same as workflow Job.

```
#!/bin/sh
now=$(date +%s")
filename="my_program.$now.log"
touch /mapr/datalake/uhclake/tst/developer/rkodurl/oozietst/sqoop/workflow/$filename
```

- Create “coord.properties” with config or parameters for the coordinator Job. In below example, the job is scheduled to run every hour’ start-instance, end-instance, coord App, workflow app path and script path are parameterized.

```
nameNode=maprfs:///
jobTracker=dbslld0069:8032
queueName=fabbddev_q1
examplesRoot=oozietst
oozie.libpath=/oozie/share/lib
startTime=2017-11-13T22:05Z
endTime=2017-11-14T22:15Z
oozie.coord.application.path=${nameNode}/datalake/uhclake/tst/developer/${user.name}/${examplesRoot}/shell/coordinator
workflowAppUri=${oozie.coord.application.path}
myscript=test.sh
myscriptPath=${oozie.wf.application.path}/test.sh#test.sh
```

- Create “coordinator.xml” for a shell-script in workflow App

```
<coordinator-app name="rkodurl-testShell" frequency="${coord:hours(1)}"
start="${startTime}" end="${endTime}" timezone="America/Chicago"
xmlns="uri:oozie:coordinator:0.2">
  <action>
    <workflow>
      <app-path>${workflowAppUri}</app-path>
      <configuration>
        <property>
          <name>jobTracker</name>
          <value>${jobTracker}</value>
        </property>
        <property>
          <name>nameNode</name>
          <value>${nameNode}</value>
        </property>
        <property>
          <name>queueName</name>
          <value>${queueName}</value>
        </property>
        <property>
          <name>coordId</name>
```

```

        <value>${coord:actionId()}</value>
      </property>
    </configuration>
  </workflow>
</action>
</coordinator-app>

```

- Create “workflow.xml” for Shell Action:

```

<workflow-app xmlns="uri:oozie:workflow:0.4" name="rkodur-shell-wf">
  <start to="shell-node"/>
  <action name="shell-node">
    <shell xmlns="uri:oozie:shell-action:0.2">
      <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>
      <configuration>
        <property>
          <name>mapred.job.queue.name</name>
          <value>${queueName}</value>
        </property>
      </configuration>
      <exec>${myscript}</exec>
      <file>${myscriptPath}</file>
      <capture-output/>
    </shell>
    <ok to="end"/>
    <error to="fail"/>
  </action>
  <kill name="fail">
    <message>Shell action failed, error
message[${wf:errorMessage(wf:lastErrorNode())}]</message>
  </kill>
  <end name="end"/>
</workflow-app>

```

- Submit Shell Coordinator job to Oozie, by specifying “coord.properties” to Oozie Config option. Then, we will be getting coordinator job ID:

```

/opt/mapr/oozie/oozie-4.3.0/bin/oozie job -oozie="https://apsrt3197.uhc.com:11443/oozie" -
config
/mapr/datalake/uyclake/tst/developer/rkodurl/oozietst/shell/coordinator/coord.properties -run

```

- By using below command, we can check the status of Coordinator Job:

```

/opt/mapr/oozie/oozie-4.3.0/bin/oozie job -oozie=https://apsrt3197.uhc.com:11443/oozie -info
0000420-171026154810092-oozie-mapr-C

```

UHG Datalake

```
$ /opt/mapr/oozie/oozie-4.3.0/bin/oozie job -oozie="https://apsrt3197.uhc.com:11443/oozie" -info 0000377-171026154810092-oozie-mapr-C
Job ID : 0000377-171026154810092-oozie-mapr-C
-----
Job Name      : rkodur1-testShell
App Path      : maprfs:///datalake/uhclake/tst/developer/rkodur1/examples/apps/shell/ShellPoc/coordinator
Status        : RUNNING
Start Time    : 2017-11-13 12:00 GMT
End Time      : 2020-11-14 13:25 GMT
Pause Time    : -
Concurrency   : 1
-----
ID            Status    Ext ID      Err Code    Created          Nominal Time
-----
0000377-171026154810092-oozie-mapr-C@1  SUCCEEDED 0000378-171026154810092-oozie-mapr-W -      2017-11-13 17:39 GMT 2017-11-13 12:00 GMT
0000377-171026154810092-oozie-mapr-C@2  SUCCEEDED 0000380-171026154810092-oozie-mapr-W -      2017-11-13 17:39 GMT 2017-11-13 13:00 GMT
0000377-171026154810092-oozie-mapr-C@3  SUCCEEDED 0000381-171026154810092-oozie-mapr-W -      2017-11-13 17:39 GMT 2017-11-13 14:00 GMT
0000377-171026154810092-oozie-mapr-C@4  SUCCEEDED 0000382-171026154810092-oozie-mapr-W -      2017-11-13 17:39 GMT 2017-11-13 15:00 GMT
```

➤ By using below command, we can check the status of Workflow Job:

```
/opt/mapr/oozie/oozie-4.3.0/bin/oozie job -oozie=https://apsrt3197.uhc.com:11443/oozie -info 0000421-171026154810092-oozie-mapr-W
```

```
$ /opt/mapr/oozie/oozie-4.3.0/bin/oozie job -oozie="https://apsrt3197.uhc.com:11443/oozie" -info 0000378-171026154810092-oozie-mapr-W
Job ID : 0000378-171026154810092-oozie-mapr-W
-----
Workflow Name : rkodur-shell-wf
App Path      : maprfs:///datalake/uhclake/tst/developer/rkodur1/examples/apps/shell/ShellPoc/coordinator
Status        : SUCCEEDED
Run           : 0
User          : rkodur1
Group         : -
Created       : 2017-11-13 17:39 GMT
Started      : 2017-11-13 17:39 GMT
Last Modified : 2017-11-13 17:45 GMT
Ended        : 2017-11-13 17:45 GMT
CoordAction ID: 0000377-171026154810092-oozie-mapr-C@1
-----
Actions
-----
ID            Status    Ext ID      Ext Status  Err Code
-----
0000378-171026154810092-oozie-mapr-W@start:  OK      -          OK          -
0000378-171026154810092-oozie-mapr-W@shell-node  OK      job_1508958967521_684187 SUCCEEDED  -
0000378-171026154810092-oozie-mapr-W@end        OK      -          OK          -
```

Note: For the coordinator Job, Job ID will ends with "C"; whereas for workflow job, job ID ends with "W"

UI Information:

Go to URL <https://apsrt3197:11443/oozie/> Check Job Coordinator ID

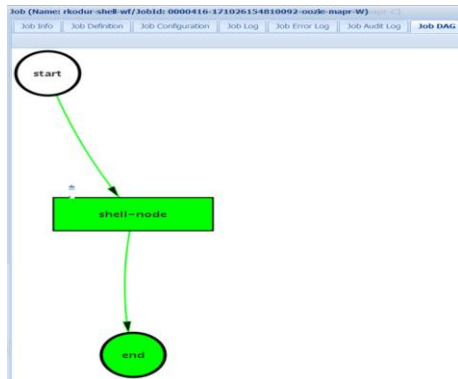
Oozie Web Console									
Workflow Jobs Coordinator Jobs Bundle Jobs System Info Instrumentation Settings									
All Jobs Active Jobs Done Jobs Custom Filter ▼									
Job Id	Name	Status	User	Group	Frequency	Unit	Started	Next Materialization	
1	0000377-171026154810092-oozie-mapr-C rkodur1-testShell	RUNNING	rkodur1		60	MINUTE	Mon, 13 Nov 2017 12:00:00 ...	Tue, 14 Nov 2017 04:00:00 ...	
2	0000081-171026154810092-oozie-mapr-C coord-dmf-md...	RUNNING	achowdh6		1	DAY	Thu, 09 Nov 2017 19:40:00 ...	Tue, 14 Nov 2017 19:40:00 ...	

Job will be in running state until it reaches to end-time instance.

In below mentioned screenshot, 'Nominal time' represents for the future triggers.

Job (Name: rkodur1-testShell/coordJobId: 0000377-171026154810092-oozie-mapr-C)							
Coord Job Info Coord Job Definition Coord Job Configuration Coord Job Log Coord Error Log Coord Audit Log Coord Action Reruns							
Actions							
Action Id	Status	Ext Id	Error Code	Created Time	Nominal Time	Last Mod Time	
3 0000377-171026154810092-oozie-mapr-C@15	SUCCEEDED	0000412-171026154810092-oozie-ma...		Tue, 14 Nov 2017 01:58:49 ...	Tue, 14 Nov 2017 02:00:00 ...	Tue, 14 Nov 2017 02:00:18 ...	
4 0000377-171026154810092-oozie-mapr-C@14	SUCCEEDED	0000406-171026154810092-oozie-ma...		Tue, 14 Nov 2017 00:58:49 ...	Tue, 14 Nov 2017 01:00:00 ...	Tue, 14 Nov 2017 01:00:26 ...	
5 0000377-171026154810092-oozie-mapr-C@13	SUCCEEDED	0000401-171026154810092-oozie-ma...		Mon, 13 Nov 2017 23:58:49 ...	Tue, 14 Nov 2017 00:00:00 ...	Tue, 14 Nov 2017 00:00:24 ...	
6 0000377-171026154810092-oozie-mapr-C@12	SUCCEEDED	0000397-171026154810092-oozie-ma...		Mon, 13 Nov 2017 22:58:49 ...	Mon, 13 Nov 2017 23:00:00 ...	Mon, 13 Nov 2017 23:00:18 ...	
7 0000377-171026154810092-oozie-mapr-C@11	SUCCEEDED	0000394-171026154810092-oozie-ma...		Mon, 13 Nov 2017 21:58:49 ...	Mon, 13 Nov 2017 22:00:00 ...	Mon, 13 Nov 2017 22:00:25 ...	
8 0000377-171026154810092-oozie-mapr-C@10	SUCCEEDED	0000392-171026154810092-oozie-ma...		Mon, 13 Nov 2017 20:58:49 ...	Mon, 13 Nov 2017 21:00:00 ...	Mon, 13 Nov 2017 21:00:15 ...	

DAG for one of the workflow Job:



2) Scheduling Sqoop Action:

- Create a directory for resources (Coordinator.xml, workflow.xml, coord.properties) to place; let's say "coordinator".

/mapr/datalake/uhclake/tst/developer/rkodur1/oozietst/sqoop/coordinator/

Coordinator.xml

coord.properties

workflow.xml

db.hsqldb.properties

db.hsqldb.script

- Hsql db is kind of derby database act like source RDBMS system for Sqoop.

Note: Follow the same steps as Coordinator Job for Shell action for scheduling and status info.

3) Scheduling Spark Action:

- Create a directory for resources (Coordinator.xml, workflow.xml, coord.properties & lib direcorey) to place, let's say "coordinator".

/mapr/datalake/uhclake/tst/developer/rkodur1/oozietst/spark/coordinator/

UHG Datalake

Coordinator.xml

coord.properties

workflow.xml

/lib/SparkApp.jar

Note: Follow the same steps as Coordinator Job for Shell action for scheduling and status info.

Appendix:

Please refer this link for the issues may encounter in sqoop action as mentioned below.

<https://community.cloudera.com/t5/Data-Ingestion-Integration/Running-sqoop-using-oozie-I-get-Main-class-org-apache-oozie/td-p/42221>

Errors like `java.lang.RuntimeException: java.lang.ClassNotFoundException: Class org.apache.oozie.action.hadoop.SqoopMain` Exit[1].