

Oozie

“If **Opportunities** don’t knock,
build a door”



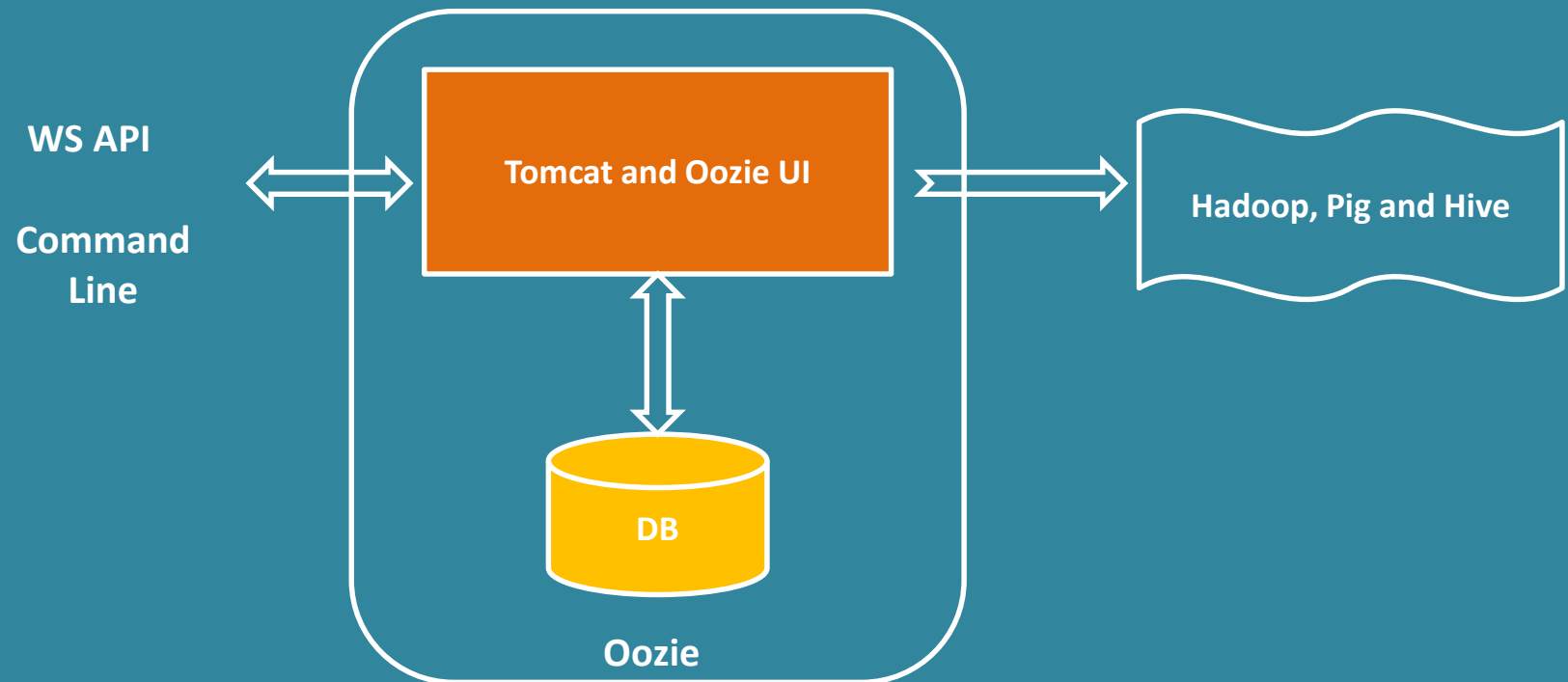
Presented By,
Siva Kumar Bhuchipalli



Oozie



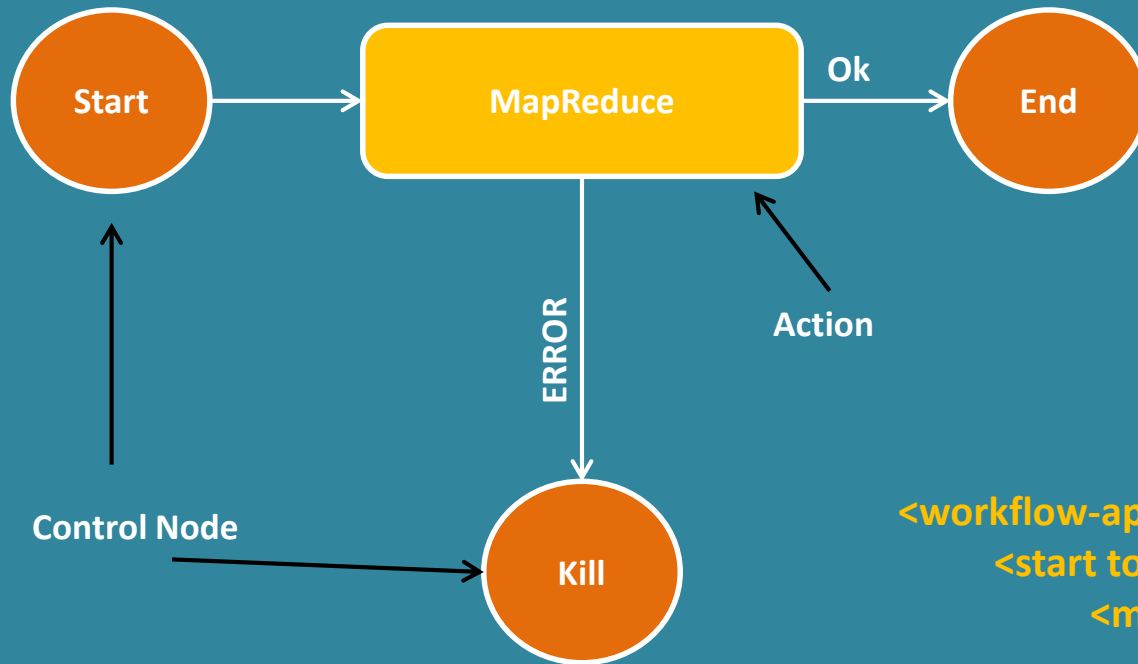
- Workflow scheduler for Hadoop
 - ✓ Java MapReduce Jobs
 - ✓ Streaming Jobs
 - ✓ Pig
- Top level Apache project
 - ✓ – Comes packaged in major Hadoop Distributions
 - ❑ Cloudera Distribution for Hadoop (CDH)
<http://incubator.apache.org/oozie>
- Provides workflow management and coordination of those workflows
- Manages Directed Acyclic Graph (DAG) of actions



- **Runs HTTP service**
 - ✓ Clients interact with the service by submitting workflows
 - ✓ Workflows are executed immediately or later
- **Workflows are defined via XML**
 - ✓ Instead of writing Java code that implements Tool interface and extending Configured class



Action And Control Nodes



```
<workflow-app name="foo-wf"..  
  <start to="[NODE-NAME]"/>  
    <map-reduce>  
      ...  
      ...  
    </map-reduce>  
    <kill name="[NODE-NAME]">  
      <message>Error  
      occurred</message>  
    </kill>  
  <end name="[NODE-NAME]"/>  
</workflow-app>
```

- **Control Flow**

- ✓ start, end, kill
- ✓ decision
- ✓ fork, join



- **Actions**

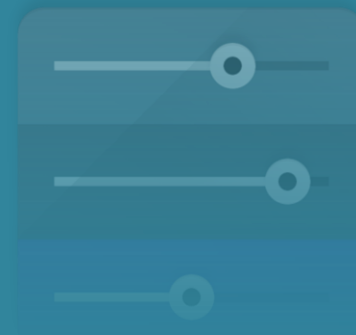
- ✓ map-reduce
- ✓ java
- ✓ pig
- ✓ Hdfs
- ✓ Hive
- ✓ Shell



```
<decision name="check-output">
  <switch>
    <case to="end">
      ${wf:actionData('shell-node')['my_output']} eq
'Hello Oozie'}
    </case>
    <default to="fail-output"/>
  </switch>
</decision>
```

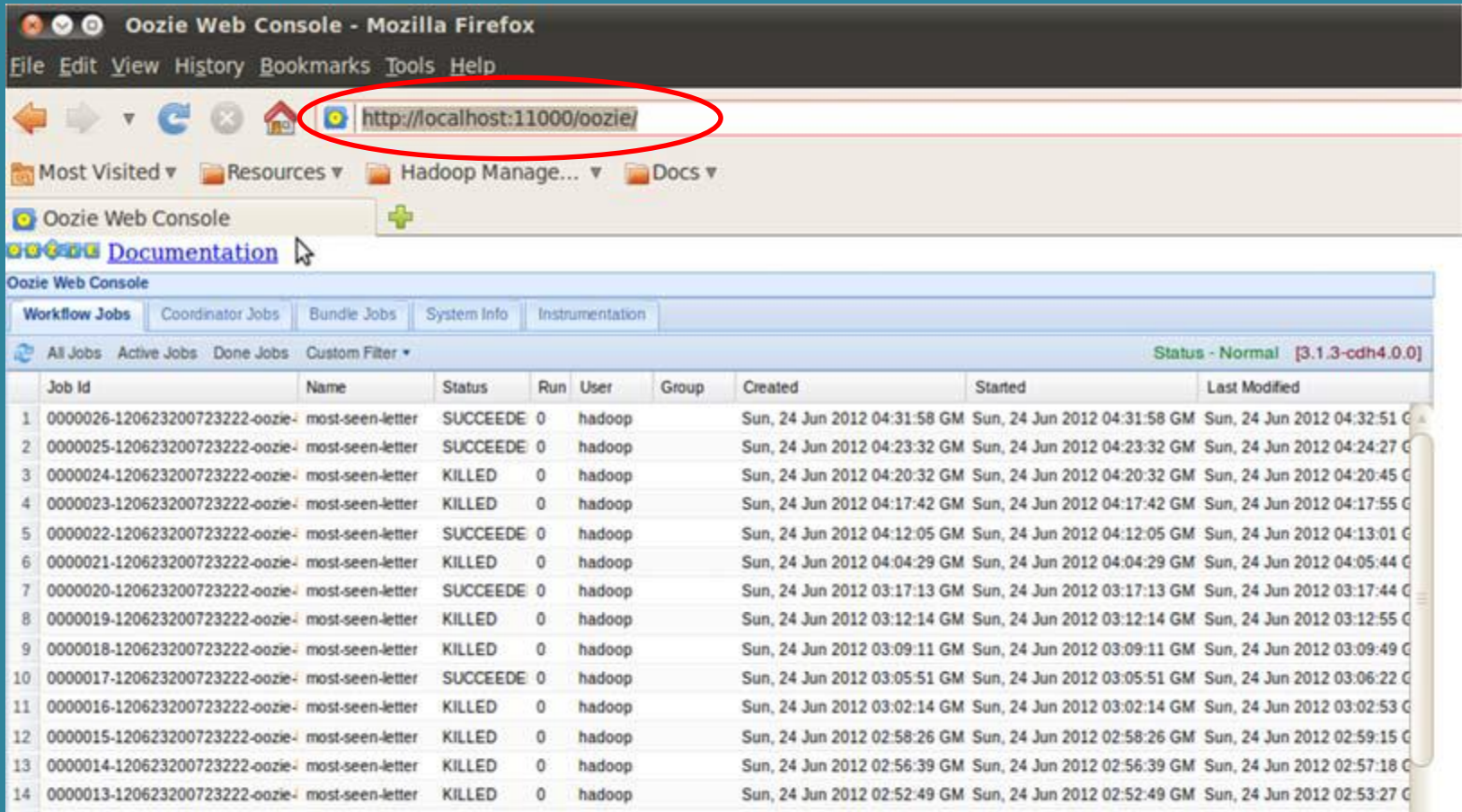
Oozie Coordination Engine

- Oozie Coordination Engine can trigger workflows by
 - ✓ Time (Periodically)
 - ✓ Data Availability (Data appears in a directory)



Test Installation

\$ oozie admin -status -oozie <http://localhost:11000/oozie/>
System mode: NORMAL



Oozie Web Console - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Most Visited Resources Hadoop Manage... Docs

Oozie Web Console

Documentation

Oozie Web Console

Workflow Jobs Coordinator Jobs Bundle Jobs System Info Instrumentation

All Jobs Active Jobs Done Jobs Custom Filter Status - Normal [3.1.3-cdh4.0.0]

Job Id	Name	Status	Run	User	Group	Created	Started	Last Modified
1 0000026-120623200723222-oozie-	most-seen-letter	SUCCEDE	0	hadoop		Sun, 24 Jun 2012 04:31:58 GM	Sun, 24 Jun 2012 04:31:58 GM	Sun, 24 Jun 2012 04:32:51 G
2 0000025-120623200723222-oozie-	most-seen-letter	SUCCEDE	0	hadoop		Sun, 24 Jun 2012 04:23:32 GM	Sun, 24 Jun 2012 04:23:32 GM	Sun, 24 Jun 2012 04:24:27 G
3 0000024-120623200723222-oozie-	most-seen-letter	KILLED	0	hadoop		Sun, 24 Jun 2012 04:20:32 GM	Sun, 24 Jun 2012 04:20:32 GM	Sun, 24 Jun 2012 04:20:45 G
4 0000023-120623200723222-oozie-	most-seen-letter	KILLED	0	hadoop		Sun, 24 Jun 2012 04:17:42 GM	Sun, 24 Jun 2012 04:17:42 GM	Sun, 24 Jun 2012 04:17:55 G
5 0000022-120623200723222-oozie-	most-seen-letter	SUCCEDE	0	hadoop		Sun, 24 Jun 2012 04:12:05 GM	Sun, 24 Jun 2012 04:12:05 GM	Sun, 24 Jun 2012 04:13:01 G
6 0000021-120623200723222-oozie-	most-seen-letter	KILLED	0	hadoop		Sun, 24 Jun 2012 04:04:29 GM	Sun, 24 Jun 2012 04:04:29 GM	Sun, 24 Jun 2012 04:05:44 G
7 0000020-120623200723222-oozie-	most-seen-letter	SUCCEDE	0	hadoop		Sun, 24 Jun 2012 03:17:13 GM	Sun, 24 Jun 2012 03:17:13 GM	Sun, 24 Jun 2012 03:17:44 G
8 0000019-120623200723222-oozie-	most-seen-letter	KILLED	0	hadoop		Sun, 24 Jun 2012 03:12:14 GM	Sun, 24 Jun 2012 03:12:14 GM	Sun, 24 Jun 2012 03:12:55 G
9 0000018-120623200723222-oozie-	most-seen-letter	KILLED	0	hadoop		Sun, 24 Jun 2012 03:09:11 GM	Sun, 24 Jun 2012 03:09:11 GM	Sun, 24 Jun 2012 03:09:49 G
10 0000017-120623200723222-oozie-	most-seen-letter	SUCCEDE	0	hadoop		Sun, 24 Jun 2012 03:05:51 GM	Sun, 24 Jun 2012 03:05:51 GM	Sun, 24 Jun 2012 03:06:22 G
11 0000016-120623200723222-oozie-	most-seen-letter	KILLED	0	hadoop		Sun, 24 Jun 2012 03:02:14 GM	Sun, 24 Jun 2012 03:02:14 GM	Sun, 24 Jun 2012 03:02:53 G
12 0000015-120623200723222-oozie-	most-seen-letter	KILLED	0	hadoop		Sun, 24 Jun 2012 02:58:26 GM	Sun, 24 Jun 2012 02:58:26 GM	Sun, 24 Jun 2012 02:59:15 G
13 0000014-120623200723222-oozie-	most-seen-letter	KILLED	0	hadoop		Sun, 24 Jun 2012 02:56:39 GM	Sun, 24 Jun 2012 02:56:39 GM	Sun, 24 Jun 2012 02:57:18 G
14 0000013-120623200723222-oozie-	most-seen-letter	KILLED	0	hadoop		Sun, 24 Jun 2012 02:52:49 GM	Sun, 24 Jun 2012 02:52:49 GM	Sun, 24 Jun 2012 02:53:27 G

Running Oozie Examples

- Extract examples packaged with Oozie
 - ✓ `$ cd $OOZIE_HOME`
 - ✓ `$ tar xvf oozie-examples.tar.gz`
- Copy examples to HDFS to user's home directory
 - ✓ `$ hdfs dfs -put examples examples`
- Run an example
 - ✓ `$ oozie job -oozie http://localhost:11000/oozie -config examples/apps/map-reduce/job.properties -run`
- Check Web Console
 - ✓ <http://localhost:11000/oozie/>

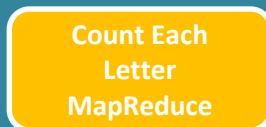
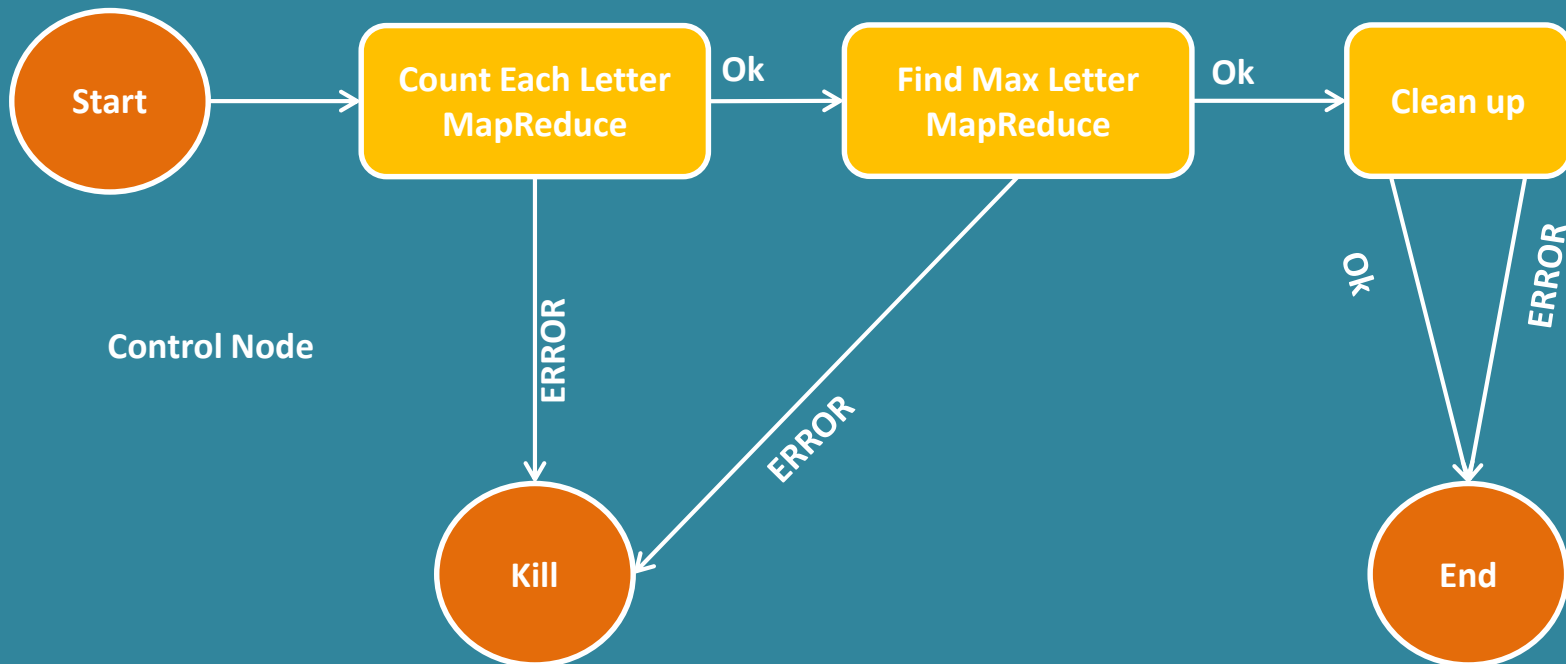
Oozie Web Console									
Workflow Jobs Coordinator Jobs Bundle Jobs System Info Instrumentation									
All Jobs Active Jobs Done Jobs Custom Filter ▾									
Status - Normal [3.1.3-cdh4.0.0]									
Job Id	Name	Status	Run	User	Group	Created	Started	Last Modified	
1 0000000-120623200723222-oozie-	map-reduce-wf	SUCCEEDED	0	hadoop		Sun, 24 Jun 2012 00:25:47 GM	Sun, 24 Jun 2012 00:25:48 GM	Sun, 24 Jun 201	

Oozie WorkFlow



- Workflows consist of
 - ✓ Action nodes
 - ☐ MapReduce, Pig, Hive
 - ☐ Streaming, Java, etc...
 - ✓ Control flow nodes
 - ☐ Logic decisions between action nodes
 - ☐ Execute actions based on conditions or in parallel
- Workflows begin with **START node**
- Workflows succeed with **END node**
- Workflows fail with **KILL node**
- Several actions support JSP Expression Language (**EL**)

Most Occurrences WorkFlows



→ Action Node



→ Control Flow Node



→ Control Node

Most Occurrences WorkFlows

```
<workflow-app xmlns="uri:oozie:workflow:0.2" name="most-seen-letter">
  <start to="count-each-letter"/>
  <action name="count-each-letter">
    <map-reduce>
      <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>
      <prepare>
        <delete path="${nameNode}${outputDir}"/>
        <delete path="${nameNode}${intermediateDir}"/>
      </prepare>
      <configuration>
        ...
        <property>
          <name>mapreduce.job.map.class</name>
          <value>mr.wordcount.StartsWithCountMapper</value>
        </property>
        ...
      </configuration>
    </map-reduce>
    <ok to="find-max-letter"/>
    <error to="fail"/>
  </action>
  ...
</workflow-app>
```

START Action Node to count-each-letter MapReduce action

MapReduce have optional Prepare section

Pass property that will be set on MapReduce job's Configuration object

In case of success, go to the next job; in case of failure go to fail node

...

First Map-Reduce Action

<map-reduce>

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

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

Administrative items to indicate where namenode and resource manager is

<prepare>

<delete path="\${nameNode}\${outputDir}"/>

<delete path="\${nameNode}\${intermediateDir}"/>

</prepare>

Optional prepare section; allows to execute command prior running the job

<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>

By default "old api" is used; specify to use new api

Specify which queue to submit this job to Resource Manager

...


...

First Map-Reduce Action

...

```
<property>
  <name>mapreduce.job.map.class</name>
  <value>mr.wordcount.StartsWithCountMapper</value>
</property>
<property>
  <name>mapreduce.job.combine.class</name>
  <value>mr.wordcount.StartsWithCountReducer</value>
</property>
<property>
  <name>mapreduce.job.reduce.class</name>
  <value>mr.wordcount.StartsWithCountReducer</value>
</property>
<property>
  <name>mapreduce.job.inputformat.class</name>
  <value>org.apache.hadoop.mapreduce.lib.input.TextInputFormat</value>
</property>
<property>
  <name>mapreduce.job.outputformat.class</name>
  <value>org.apache.hadoop.mapreduce.lib.output.TextOutputFormat</value>
</property>
```

Specify Mapper, Reducer, Input and Output formats; this is instead of Tool implementation




...

This action will produce a file of tab separated keyvalue pairs as specified by TextOutputFormat



First Map-Reduce Action (continued)

```
...  
...  
<property>  
  <name>mapreduce.job.output.key.class</name>  
  <value>org.apache.hadoop.io.Text</value>  
</property>  
<property>  
  <name>mapreduce.job.output.value.class</name>  
  <value>org.apache.hadoop.io.IntWritable</value>  
</property>  
<property>  
  <name>mapreduce.input.fileinputformat.inputdir</name>  
  <value>${inputFile}</value>  
</property>  
<property>  
  <name>mapreduce.output.fileoutputformat.outputdir</name>  
  <value>${intermediateDir}</value>  
</property>  
</configuration>  
</map-reduce>
```



These properties are substituted from job.properties file

Most Occurrences WorkFlows

```
<action name="find-max-letter"> ← Second MapReduce job
  <map-reduce>
    <job-tracker>${jobTracker}</job-tracker>
    <name-node>${nameNode}</name-node> ← Namenode and Yarn
    <configuration>                    Resource Manager
    ...                               Location
    ...
    <property>
      <name>mapreduce.job.map.class</name>
      <value>mr.workflows.MostSeenStartLetterMapper</value>
    </property>
    <property>
      <name>mapreduce.job.combine.class</name>
      <value>mr.workflows.MostSeendStartLetterReducer</value>
    ...
    ...
    </configuration>
    </map-reduce>
    <ok to="clean-up"/> ← Control Flow Node
    <error to="fail"/>
  </action>
```

Token substituted from application properties file

Second Map-Reduce Action

...

```
<property>
  <name>mapreduce.job.map.class</name>
  <value>mr.workflows.MostSeenStartLetterMapper</value>
</property>
<property>
  <name>mapreduce.job.combine.class</name>
  <value>mr.workflows.MostSeendStartLetterReducer</value>
</property>
<property>
  <name>mapreduce.job.reduce.class</name>
  <value>mr.workflows.MostSeendStartLetterReducer</value>
</property>
```

...




Specify Mapper, Reducer and Combiner

Second Map-Reduce Action (continued)

First map-reduce action produced a file with tab separated key-value pairs; second step utilizes KeyValueTextInputFormat to read these pairs as text

```
...  
<property>  
  <name>mapreduce.job.inputformat.class</name>  
  <value>org.apache.hadoop.mapreduce.lib.input.KeyValueTextInputFormat</value>  
>  
</property>  
<property>  
  <name>mapreduce.job.outputformat.class</name>  
  <value>org.apache.hadoop.mapreduce.lib.output.TextOutputFormat</value>  
</property>  
<property>  
  <name>mapreduce.job.output.key.class</name>  
  <value>org.apache.hadoop.io.Text</value>  
</property>  
<property>  
  <name>mapreduce.job.output.value.class</name>  
  <value>org.apache.hadoop.io.IntWritable</value>  
</property>  
...
```



Most Occurrences WorkFlows

```
...  
...  
<action name="clean-up">  
  <fs>  
    <delete path='${nameNode}${intermediateDir}'/>  
  </fs>  
  <ok to="end"/>  
  <error to="end"/>  
</action>  
  
<kill name="fail">  
  <message>Map/Reduce failed, error  
    message[${wf:errorMessage(wf:lastErrorNode())}]</message>  
</kill>  
  <end name="end"/>  
</workflow-app>
```

Clean node, remove temporary folder

Workflow has failed, display error message

JSP expression language

Workflow ended with success

Package And Run Your WorkFlow

1. Create application directory structure with workflow definitions and resources
 - ✓ Workflow.xml, jars, etc..
2. Copy application directory to HDFS
3. Create application configuration file
 - ✓ specify location of the application directory on HDFS
 - ✓ specify location of the namenode and resource manager
4. Submit workflow to Oozie
 - ✓ Utilize oozie command line
5. Monitor running workflow(s)



1.Oozie Application Directory

- Must comply to directory structure spec

mostSeenLetter-oozieWorkflow

|--lib/

| |--HadoopSamples.jar

|--workflow.xml

Application
Workflow Root



Libraries should be placed
under lib directory

Workflow.xml
defines workflow

1.Oozie Application Directory

- Can use a build tool to generate this structure
 - ✓ Samples use maven plugins (see pom.xml)
 - ☐ Maven-dependency-plugin
 - ☐ Maven-resources-plugin
 - ✓ Run 'mvn clean package'
 - ☐ Will create 'mostSeenLetter-oozieWorkflow' directory with dependencies and workflow definitions



2.Copy Application Directories To HDFS

- Oozie utilizes HDFS to load applications

hdfs dfs -put mostSeenLetter-oozieWorkflow

Copies directory from local files system onto HDFS;
directory gets copied to user's home directory



3. Create Application Configuration File

- job.properties - Needs to exist locally, required for submission
-

nameNode=hdfs://localhost:8020

jobTracker=localhost:8032

queueName=default

← Properties for required locations such as namenode and resource manager

inputFile=/training/data/hamlet.txt

intermediateDir=/training/playArea/mostSeenLetter-oozieWorkflow-tmp

outputDir=/training/playArea/oozieWorkflow

← Properties needed for the MapReduce actions in the workflow

oozie.wf.application.path=\${nameNode}/user/\${user.name}/mostSeenLetter-oozieWorkflow

↖ Most importantly HDFS location of the application must be specified

4.Submit WorkFlow To Oozie

- Use oozie command line tool
 - ✓ For usage: \$oozie help

Application configuration file

```
$ oozie job -oozie http://localhost:11000/oozie -config job.properties -run  
job: 0000001-120711224224630-oozie-hado-W
```

Application ID; use this ID to get status



5. Monitor Running WorkFlow

- Two options
 - ✓ Command line (**\$oozie**)
 - ✓ Web Interface (<http://localhost:11000/oozie>)



5. Monitor Running WorkFlow(s) – Command Line



```
$ oozie job -info 0000001-120711224224630-oozie-hado-W
```

```
Job ID : 0000001-120711224224630-oozie-hado-W
```

```
-----  
Workflow Name : most-seen-letter
```

```
App Path : hdfs://localhost:8020/user/hadoop/mostSeenLetter-oozieWorkflow
```

```
Status : RUNNING
```

```
Run : 0
```

```
User : hadoop
```

```
Group : -
```

```
Created : 2012-07-13 03:08
```

```
Started : 2012-07-13 03:08
```

```
Last Modified : 2012-07-13 03:08
```

```
Ended : -
```

```
CoordAction ID: -
```

```
Actions
```

Workflow overview

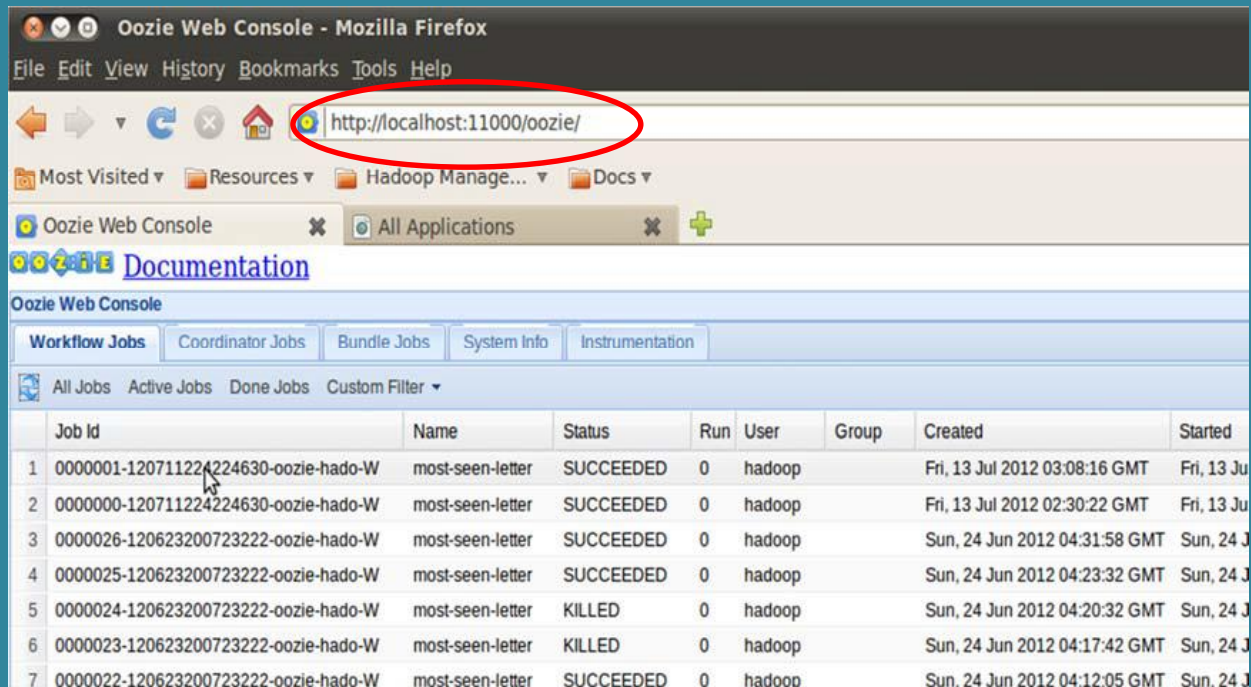
Get info by
Application ID

Completed and
executing tasks

```
-----  
ID                               Status      Ext ID      Ext Status Err Code  
-----  
0000001-120711224224630-oozie-hado-W@count-each-letter OK          job_1342136595052_0006 SUCCEEDED -  
-----  
0000001-120711224224630-oozie-hado-W@find-max-letter RUNNING    job_1342136595052_0008
```

5. Monitor Running WorkFlow(S) – Web Interface

<http://localhost:11000/oozie/>



Oozie Web Console - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:11000/oozie/

Most Visited Resources Hadoop Manage... Docs

Oozie Web Console All Applications

[Documentation](#)

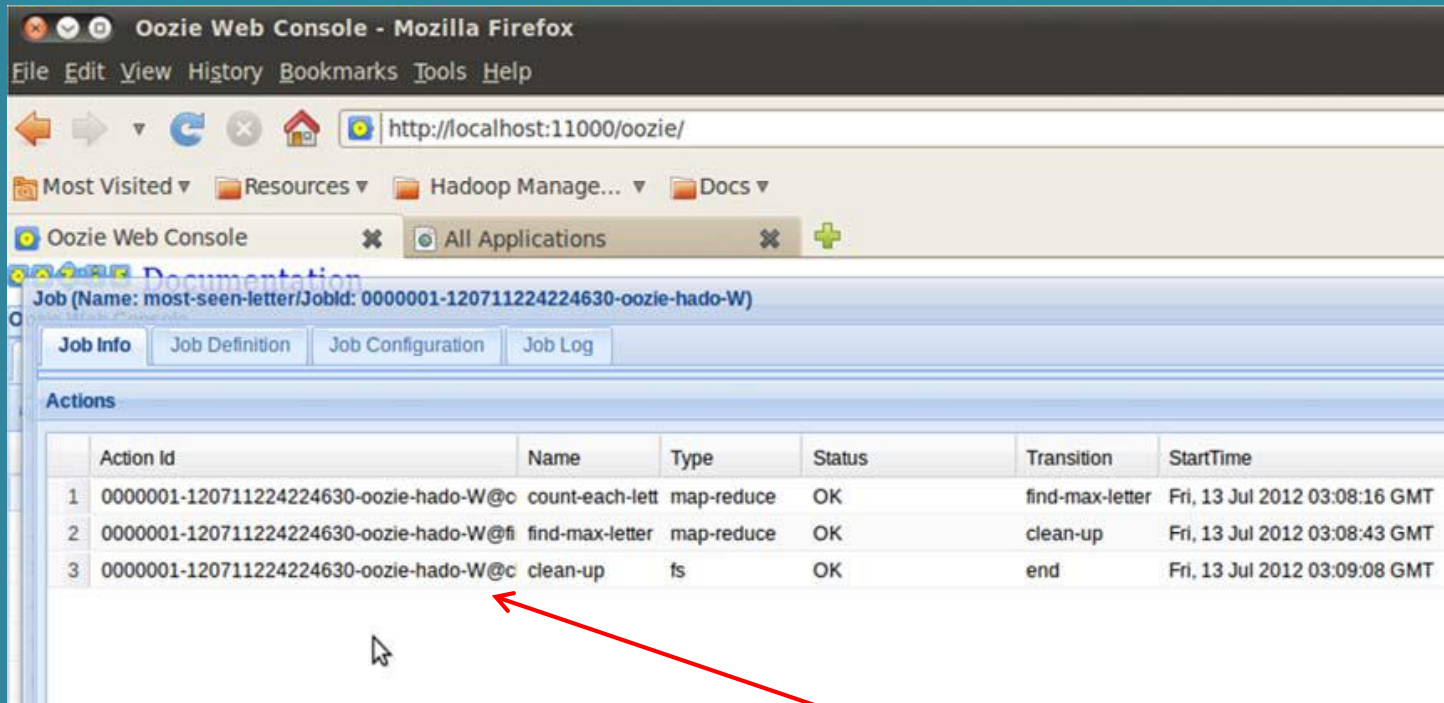
Oozie Web Console

Workflow Jobs Coordinator Jobs Bundle Jobs System Info Instrumentation

All Jobs Active Jobs Done Jobs Custom Filter

	Job Id	Name	Status	Run	User	Group	Created	Started
1	0000001-120711224224630-oozie-hado-W	most-seen-letter	SUCCEEDED	0	hadoop		Fri, 13 Jul 2012 03:08:16 GMT	Fri, 13 Jul 2012 03:08:16 GMT
2	0000000-120711224224630-oozie-hado-W	most-seen-letter	SUCCEEDED	0	hadoop		Fri, 13 Jul 2012 02:30:22 GMT	Fri, 13 Jul 2012 02:30:22 GMT
3	0000026-120623200723222-oozie-hado-W	most-seen-letter	SUCCEEDED	0	hadoop		Sun, 24 Jun 2012 04:31:58 GMT	Sun, 24 Jun 2012 04:31:58 GMT
4	0000025-120623200723222-oozie-hado-W	most-seen-letter	SUCCEEDED	0	hadoop		Sun, 24 Jun 2012 04:23:32 GMT	Sun, 24 Jun 2012 04:23:32 GMT
5	0000024-120623200723222-oozie-hado-W	most-seen-letter	KILLED	0	hadoop		Sun, 24 Jun 2012 04:20:32 GMT	Sun, 24 Jun 2012 04:20:32 GMT
6	0000023-120623200723222-oozie-hado-W	most-seen-letter	KILLED	0	hadoop		Sun, 24 Jun 2012 04:17:42 GMT	Sun, 24 Jun 2012 04:17:42 GMT
7	0000022-120623200723222-oozie-hado-W	most-seen-letter	SUCCEEDED	0	hadoop		Sun, 24 Jun 2012 04:12:05 GMT	Sun, 24 Jun 2012 04:12:05 GMT

5. Monitor Running WorkFlow(S) – Web Interface



The screenshot shows the Oozie Web Console interface in a Mozilla Firefox browser window. The address bar displays `http://localhost:11000/oozie/`. The browser has two tabs: "Oozie Web Console" and "All Applications". The "Oozie Web Console" tab is active, showing a job details page for "Job (Name: most-seen-letter/JobId: 0000001-120711224224630-oozie-hado-W)". The page has four tabs: "Job Info", "Job Definition", "Job Configuration", and "Job Log". The "Job Info" tab is selected, displaying a table of actions.

	Action Id	Name	Type	Status	Transition	StartTime
1	0000001-120711224224630-oozie-hado-W@c	count-each-lett	map-reduce	OK	find-max-letter	Fri, 13 Jul 2012 03:08:16 GMT
2	0000001-120711224224630-oozie-hado-W@fi	find-max-letter	map-reduce	OK	clean-up	Fri, 13 Jul 2012 03:08:43 GMT
3	0000001-120711224224630-oozie-hado-W@c	clean-up	fs	OK	end	Fri, 13 Jul 2012 03:09:08 GMT

A red arrow points from the text below to the third action in the table.

Clicking on a particular application/job will bring up list of all the actions; click on each action to get further details

5. Monitor Running WorkFlow(S) – Web Interface

The screenshot shows the Oozie Job Information window. The 'Job Info' tab is selected, and the 'Action Info' sub-tab is active. The window displays details for the action 'find-max-letter' (JobId: 0000001-120711224224630-oozie-hado-W). The details include:

- Name: find-max-letter
- Type: map-reduce
- Transition: clean-up
- Start Time: Fri, 13 Jul 2012 03:08:43 GMT
- End Time: Fri, 13 Jul 2012 03:09:08 GMT
- Status: OK
- Error Code: (empty)
- Error Message: (empty)
- External ID: job_1342136595052_0008
- External Status: SUCCEEDED
- Console URL: http://localhost:8088/proxy/application_1342136595052_0008/
- Tracker URI: localhost:8021

Below the details, a table shows the job's progress. The table has columns for 'Action', 'StartTime', and 'EndTime'. The data rows show the job's progress across different actions.

Action	StartTime	EndTime
find-max-letter	Fri, 13 Jul 2012 03:08:16 GMT	Fri, 13 Jul 2012 03:08:43 GMT
map	Fri, 13 Jul 2012 03:08:43 GMT	Fri, 13 Jul 2012 03:09:08 GMT
clean-up	Fri, 13 Jul 2012 03:09:08 GMT	Fri, 13 Jul 2012 03:09:08 GMT

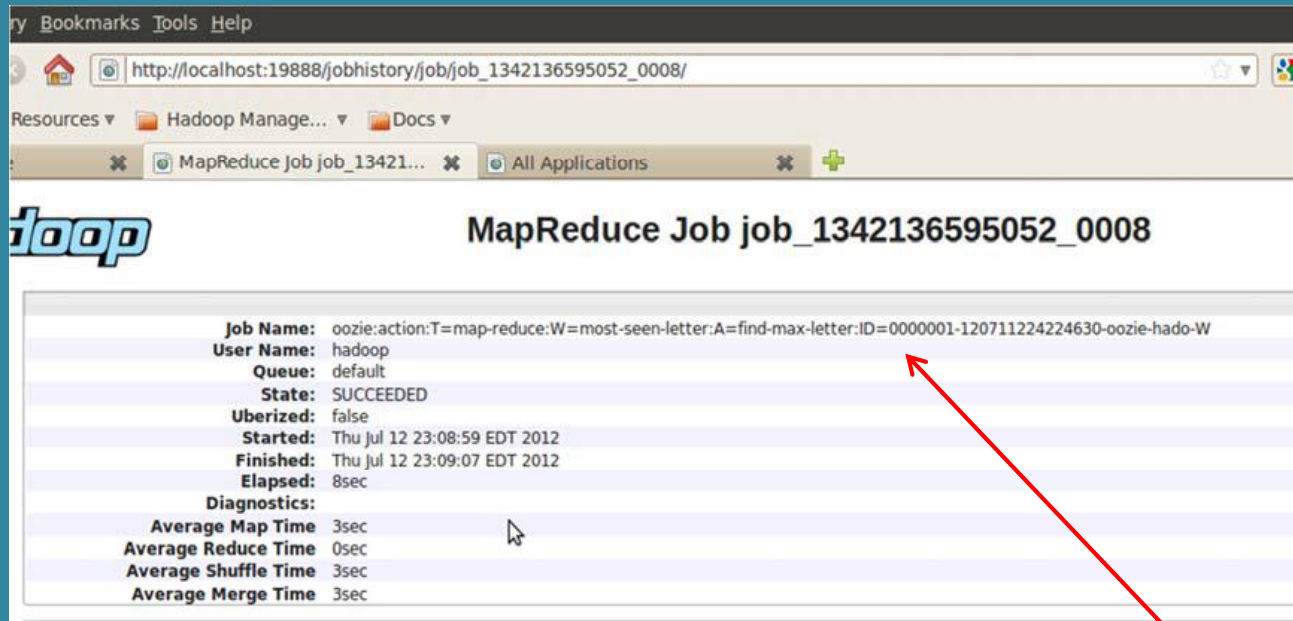
At the bottom of the window, a table shows the job's overall status. The table has columns for 'JobId', 'Job Name', 'Status', 'Exit Code', and 'Hadoop Command'. The data row shows the job's overall status as 'KILLED'.

JobId	Job Name	Status	Exit Code	Hadoop Command
0000001-120623200723222-oozie-hado-W	most-seen-letter	KILLED	0	hadoop

Link to Resource Manager to view details of the job for this particular Oozie Action

This view displays details for a selected action.

5. Monitor Running WorkFlow(S) – Web Interface



The screenshot shows a web browser window with the URL `http://localhost:19888/jobhistory/job/job_1342136595052_0008/`. The page title is "MapReduce Job job_1342136595052_0008". The Hadoop logo is visible on the left. The job details are as follows:

Job Name:	oozie:action:T=map-reduce:W=most-seen-letter:A=find-max-letter:ID=0000001-120711224224630-oozie-hado-W
User Name:	hadoop
Queue:	default
State:	SUCCEEDED
Uberized:	false
Started:	Thu Jul 12 23:08:59 EDT 2012
Finished:	Thu Jul 12 23:09:07 EDT 2012
Elapsed:	8sec
Diagnostics:	
Average Map Time	3sec
Average Reduce Time	0sec
Average Shuffle Time	3sec
Average Merge Time	3sec

Two red arrows point from the text annotations to the "Job Name" field and the "Console Url" link in the "Diagnostics" section.

Oozie assigns a name to each job

Clicking on the "Console Url" from action view will take you to the details of the job for that action

Hive Action

```
<workflow-app xmlns="uri:oozie:workflow:0.2" name="hive-wf">
  <start to="hive-node"/>

  <action name="hive-node">
    <hive xmlns="uri:oozie:hive-action:0.2">
      <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>
      <prepare>
        <delete path="${nameNode}/user/${wf:user()}/${examplesRoot}/output-data/hive"/>
        <mkdir path="${nameNode}/user/${wf:user()}/${examplesRoot}/output-data"/>
      </prepare>
      <configuration>
        <property>
          <name>mapred.job.queue.name</name>
          <value>${queueName}</value>
        </property>
      </configuration>
      <script>script.q</script>
      <param>INPUT=/user/${wf:user()}/${examplesRoot}/input-data/table</param>
      <param>OUTPUT=/user/${wf:user()}/${examplesRoot}/output-data/hive</param>
    </hive>
    <ok to="end"/>
    <error to="fail"/>
  </action>

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

Pig Action

```
<workflow-app xmlns="uri:oozie:workflow:0.2" name="pig-wf">
  <start to="pig-node"/>
  <action name="pig-node">
    <pig>
      <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>
      <prepare>
        <delete path="${nameNode}/user/${wf:user()}/${examplesRoot}/output-data/pig"/>
      </prepare>
      <configuration>
        <property>
          <name>mapred.compress.map.output</name>
          <value>true</value>
        </property>
      </configuration>
      <script>id.pig</script>
      <param>INPUT=/user/${wf:user()}/${examplesRoot}/input-data/text</param>
      <param>OUTPUT=/user/${wf:user()}/${examplesRoot}/output-data/pig</param>
    </pig>
    <ok to="end"/>
    <error to="fail"/>
  </action>
  <kill name="fail">
    <message>Pig failed, error message[${wf:errorMessage(wf:lastErrorNode())}]</message>
  </kill>
  <end name="end"/>
</workflow-app>
```

Sqoop Action

```
<workflow-app xmlns="uri:oozie:workflow:0.2" name="sqoop-wf">
  <start to="sqoop-node"/>

  <action name="sqoop-node">
    <sqoop xmlns="uri:oozie:sqoop-action:0.2">
      <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>
      <prepare>
        <delete path="${nameNode}/user/${wf:user()}/${examplesRoot}/output-data/sqoop"/>
        <mkdir path="${nameNode}/user/${wf:user()}/${examplesRoot}/output-data"/>
      </prepare>
      <command>import --connect jdbc:mysql://localhost/db --table TT --target-dir /user/${wf:user()}/${examplesRoot}/output-
data/sqoop -m 1</command>
    </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"/>
</workflow-app>
```

SSH Action

```
<workflow-app xmlns="uri:oozie:workflow:0.2" name="ssh-wf">
  <start to="ssh"/>

  <action name="ssh">
    <ssh xmlns="uri:oozie:ssh-action:0.1">
      <host>localhost</host>
      <command>echo</command>
      <args>"Hello Oozie!"</args>
    </ssh>
    <ok to="end"/>
    <error to="fail"/>
  </action>

  <kill name="fail">
    <message>SSH action failed, error message[${wf:errorMessage(wf:lastErrorNode())}]</message>
  </kill>

  <end name="end"/>
</workflow-app>
```

Coordinator XML

```
<coordinator-app name="omg_coord_app" frequency="${freq}" start="${startTime}" end="${endTime}" timezone="UTC"
xmlns="uri:oozie:coordinator:0.1">
  <controls>
    <timeout>${timeout}</timeout>
    <concurrency>${concurrency}</concurrency>
  </controls>
  <action>
    <workflow>
      <app-path>${workflowPath}</app-path>
    </workflow>
  </action>
</coordinator-app>
```

Coordinator.properties file

```
oozie.coord.application.path=hdfs://namenode:8020/user/root/omg_coordinator/  
freq=30  
startTime=2015-07-12T00:05Z  
endTime=2015-12-09T00:04Z  
timeout=30  
concurrency=1  
workflowPath=hdfs://namenode:8020/user/root/omg_oozie/  
nameNode=hdfs://namenode:8020  
jobTracker=rm:8032    //8032 in cloudera and 8050 in hortonworks  
queueName=default  
oozie.use.system.libpath=true  
#oozie.service.WorkflowAppService.system.libpath=/user/user/share/lib/lib_20150402185013/  
#oozie.wf.application.path=${nameNode}/user/${user.name}/omg_oozie
```

Any Queries?????



THANK YOU



Come Once Again