Apache Oozie: Workflow Scheduler

Overview of Oozie

- Oozie is an open-source Apache project that provides a framework for coordinating and scheduling Hadoop jobs. Oozie is not restricted to just MapReduce jobs; you can use Oozie to schedule Pig, Hive, Sqoop, Streaming jobs, and even Java programs.

- Oozie is a Java web application that runs in a Tomcat instance.

Why Oozie?

The Problem

- Doing something on the grid often required multiple steps
  - MapReduce job
  - Pig job
  - Streaming job
  - HDFS operation (mkdir, chmod, etc)…
- Multiple ad-hoc solutions existed
  - custom job control
  - shell scripts
  - cron…
- Cost of building and running apps were high
  - development and applications engineering
  - support, operations, and hardware

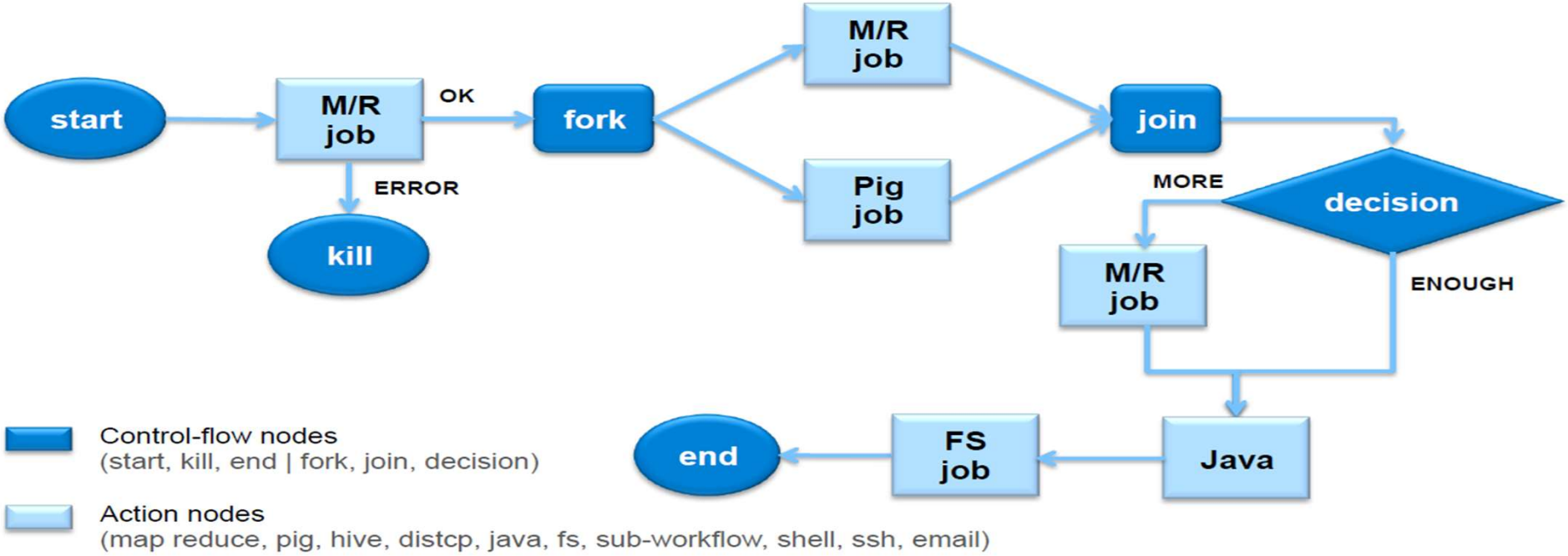
The Need

OOZIE

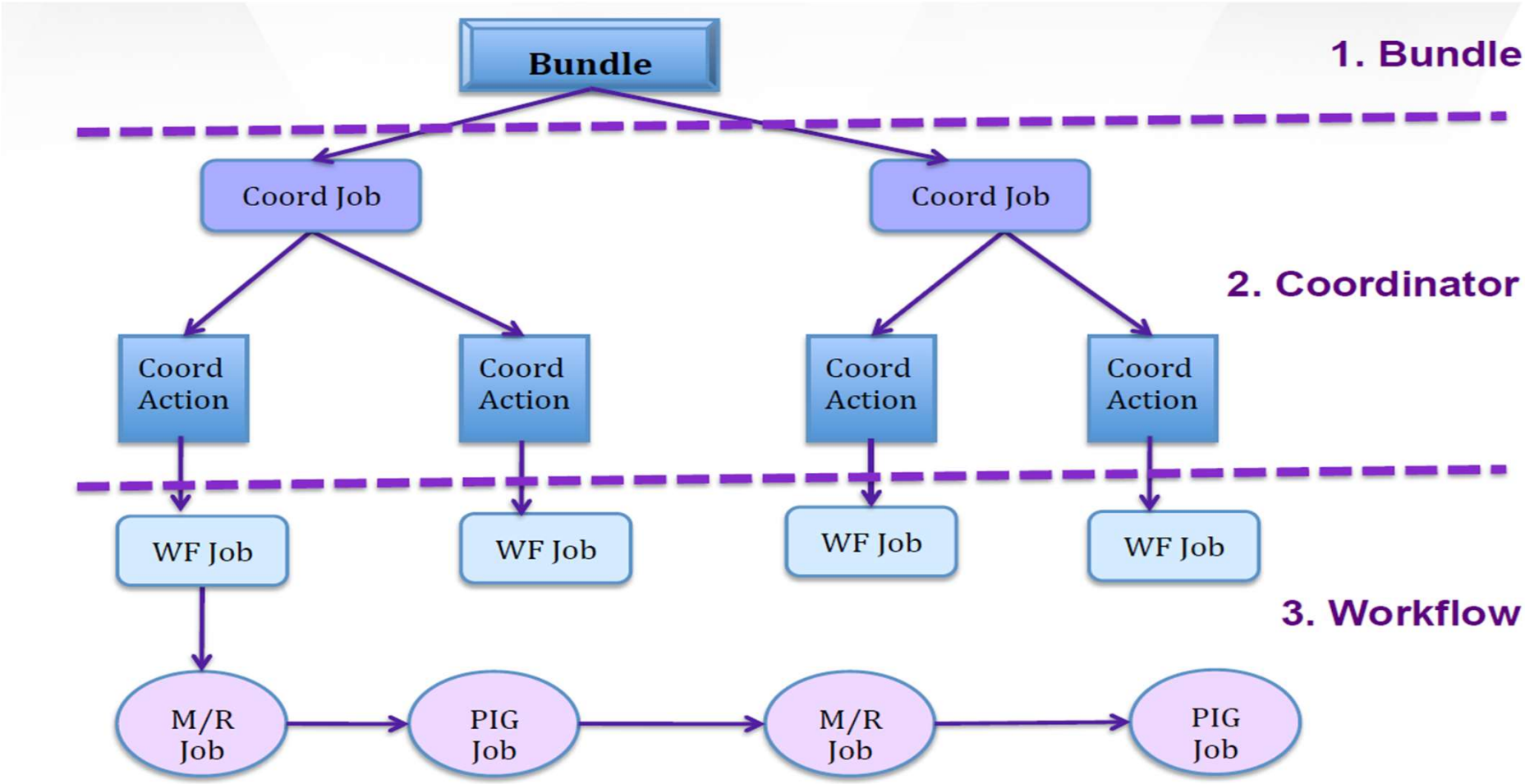
A server-based workflow scheduling system to manage Hadoop jobs

Oozie as a DAG

- Oozie executes workflow defined as DAG of jobs (Directed Acyclic Graph)
- The job type includes MapReduce, Pig, Hive, shell script, custom Java code etc.
- Introduced in Oozie 1.x



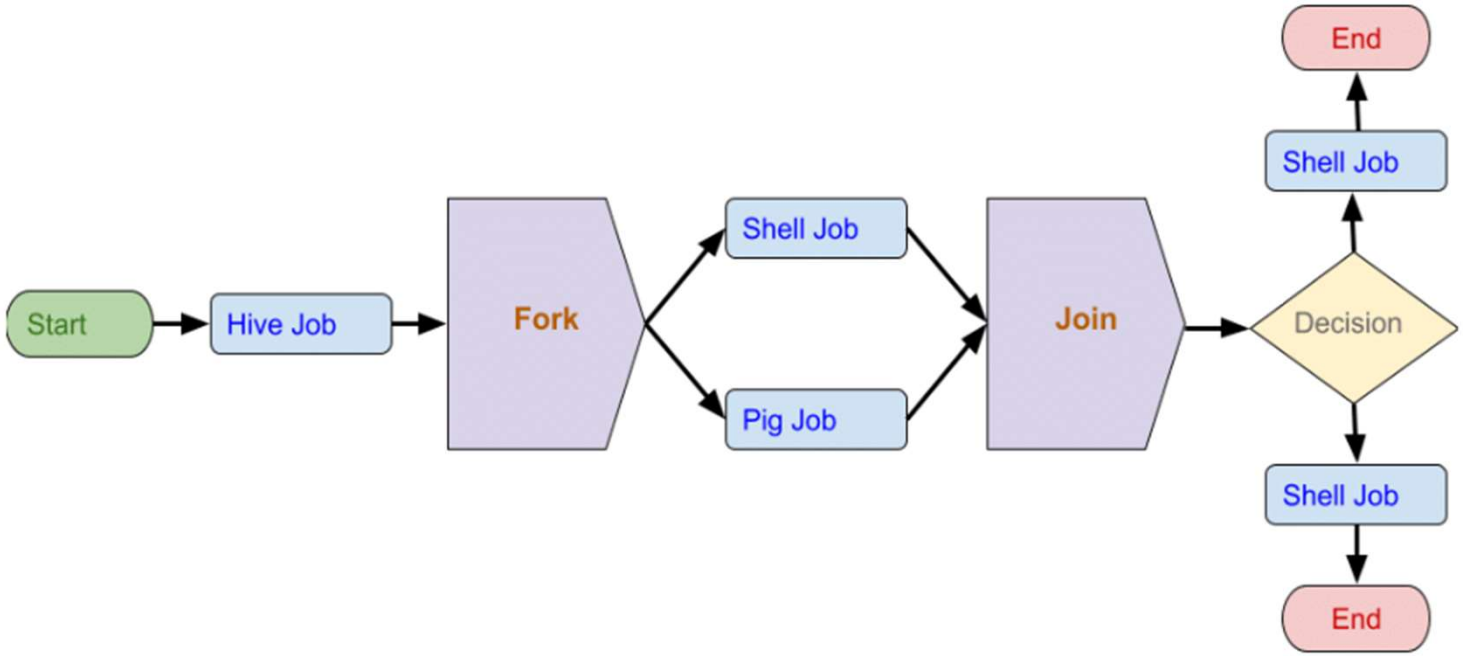
Layer of Abstraction



Oozie-Workflow

- An Apache Oozie workflow is a sequence of actions organized as a directed acyclic graph (DAG). These actions depend on each other, so that the next action can only be executed after the previous action has been completed.
- Different types of actions can be created as required. The workflow and any scripts or .jar files must be positioned in the HDFS path before executing the workflow.
- If we want to run several jobs in parallel, we can use Fork. For each use of Fork, a join must be used at the end of the Fork. Join assumes that all nodes running in parallel are children of a single Fork, as shown in the following diagram.

Sample Workflow



Oozie-Coordinator

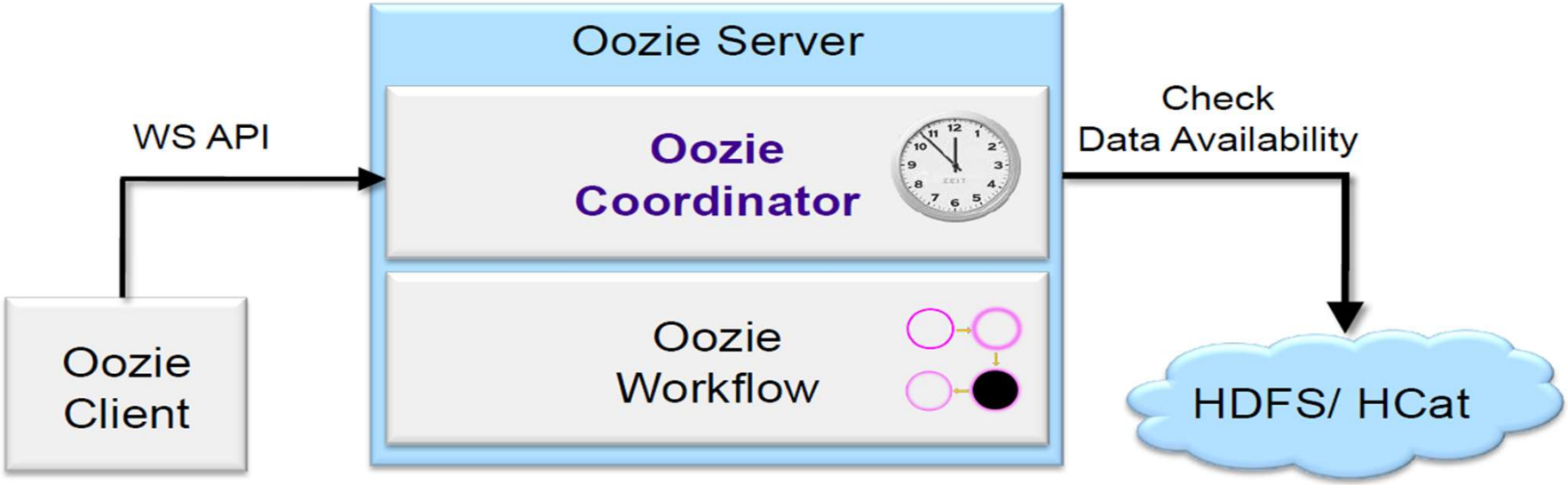
- The Oozie **Coordinator** system allows the user to define and execute recurrent and interdependent workflow jobs (data application pipelines).

- Connect workflow jobs that run regularly, but at different time intervals. The outputs of multiple subsequent runs of a workflow become the input to the next workflow.

Example: The outputs of last 4 runs of a workflow that runs every 15 minutes become the input of another workflow that runs every 60 minutes. Chaining together these workflows result it is referred as a data application pipeline.

Oozie-Coordinator

- Oozie executes workflow based on
  - time dependency (frequency)
  - data dependency
- Introduced in 2.x



Oozie -Bundle

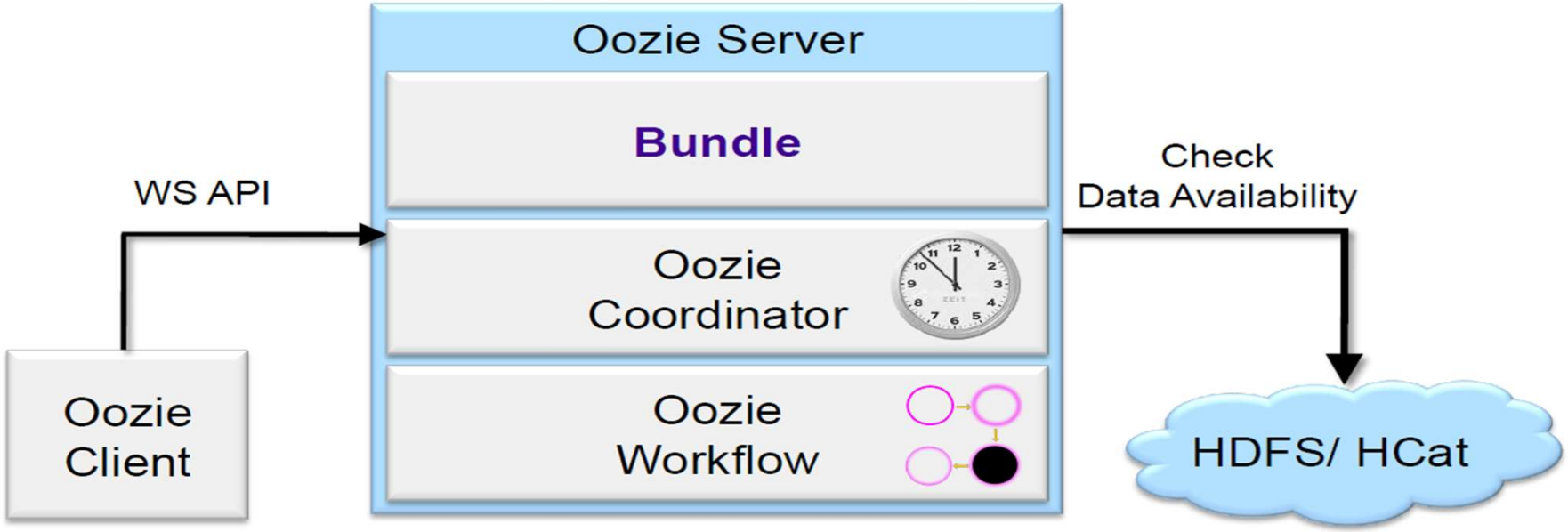
- Bundle is a higher-level oozie abstraction that will batch a set of coordinator applications. The user will be able to start/stop/suspend/resume/rerun in the bundle level resulting a better and easy operational control.
- Oozie **Bundle** system allows the user to define and execute a bunch of coordinator applications often called a data pipeline. There is no explicit dependency among the coordinator applications in a bundle. However, a user could use the data dependency of coordinator applications to create an implicit data application pipeline.

Some Key Terms

- **Kick-off-time:** The time when a bundle should start and submit coordinator applications.
- **Bundle Application:** A bundle application defines a set of coordinator applications and when to start those. Normally, bundle applications are parameterized. A bundle application is written in XML.
- **Bundle Job:** A bundle job is an executable instance of a bundle application. A job submission is done by submitting a job configuration that resolves all parameters in the application definition.
- **Bundle Definition Language:** The language used to describe bundle applications.

Bundle

- Users can define and execute a “bundle” of coordinator apps
  - large scale data processing (inter-related coordinators)
  - operability and manageability of pipelines
- User can start/stop/suspend/resume/rerun in the bundle level
- Introduced in 3.x, bundles are optional



Overview of Oozie

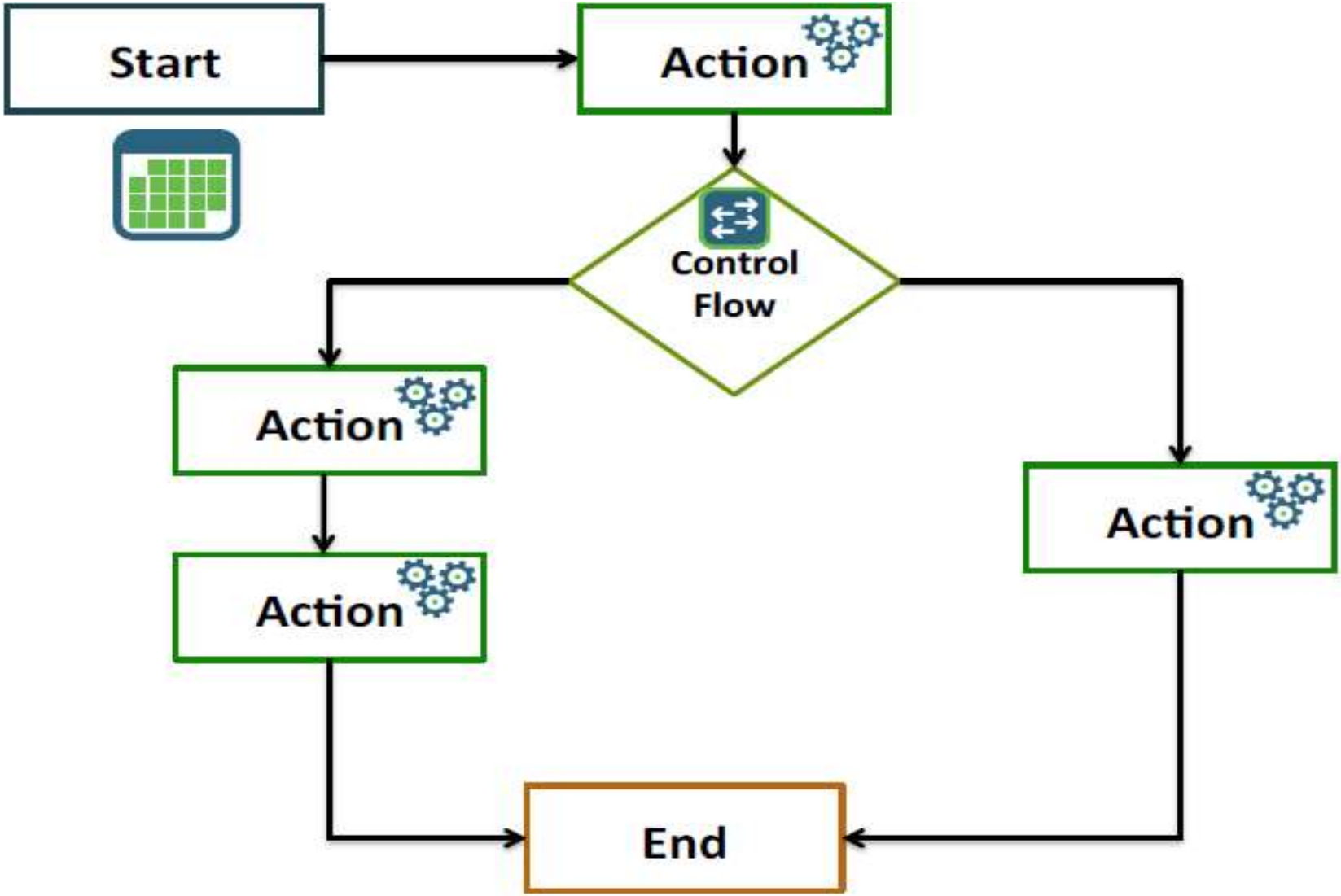
Oozie has two main capabilities:

1. Oozie Workflow: a collection of actions (defined in a **workflow.xml** file).
    - Pig Actions
    - Hive Actions
    - MapReduce Actions

2. Oozie Coordinator: a recurring workflow (defined in a **coordinator.xml** file).

    - Schedule a Job Based on Time
    - Schedule a Job Based on Data Availability

Defining an Oozie Workflow



Three Important Files

1. You need the JAR file
2. workflow.xml
3. job.properties

```
[cloudera@quickstart examples]$ ls
apps  input-data  src
[cloudera@quickstart examples]$ cd apps/map-reduce/
[cloudera@quickstart map-reduce]$ ls
job.properties                    lib                                workflow.xml
job-with-config-class.properties  workflow-with-config-class.xml
[cloudera@quickstart map-reduce]$ cd lib
[cloudera@quickstart lib]$ ls
oozie-examples-4.1.0-cdh5.8.0.jar
[cloudera@quickstart lib]$
```

Jar File...Inside lib Folder

Workflow.xml

```
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<workflow-app xmlns="uri:oozie:workflow:0.2" 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.job.queue.name</name>
                    <value>${queueName}</value>
                </property>
                <property>
```

Start node name

Action node name

JobTracker or Resource Manager parameter

Name Node Parameter

Workflow.xml

```
    <delete path="${nameNode}/user/${wf:user()}/${examplesRoot}/output-data/${outpu
</prepare>
<configuration>
    <property>
        <name>mapred.job.queue.name</name>
        <value>${queueName}</value>
    </property>
    <property>
        <name>mapred.mapper.class</name>
        <value>org.apache.oozie.example.SampleMapper</value>
    </property>
    <property>
        <name>mapred.reducer.class</name>
        <value>org.apache.oozie.example.SampleReducer</value>
    </property>
    <property>
        <name>mapred.map.tasks</name>
        <value>1</value>
    </property>
    <property>
        <name>mapred.input.dir</name>
        <value>/user/${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>
```

Delete the directory if existing

Mapper Class

Reducer class

Mapper Input Directory

Mapper Output Directory

OK if successful

fail if Not Successfull

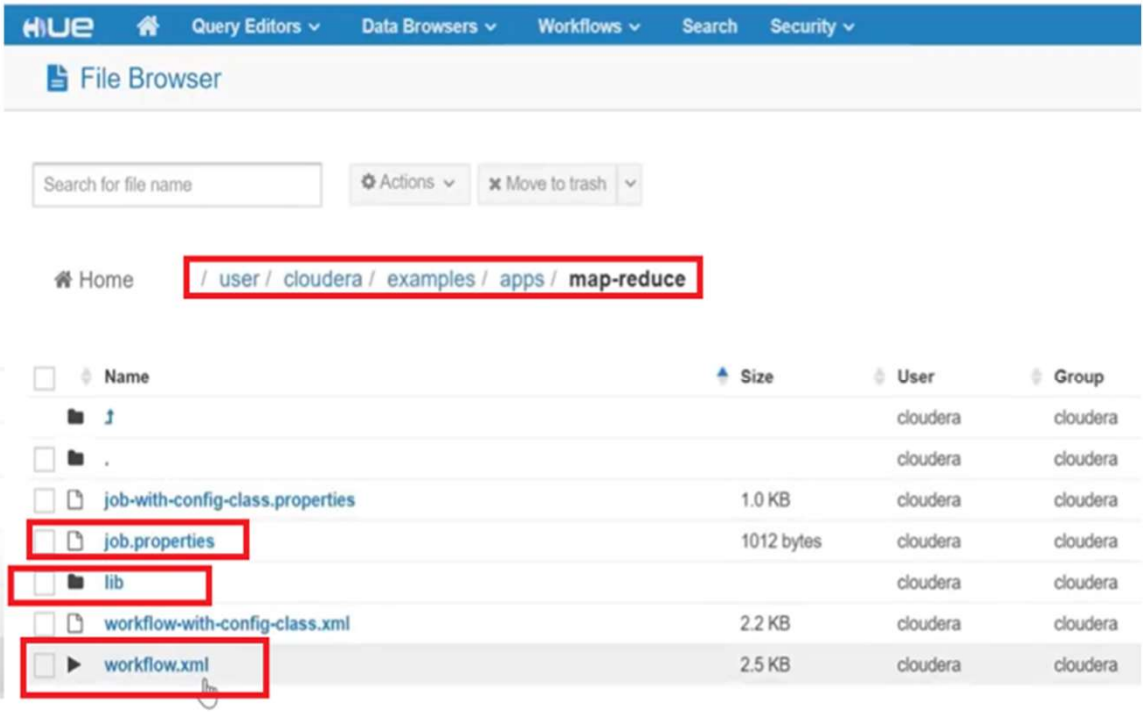
Error Message ..if not successful

job.properties file

```
nameNode=hdfs://localhost:8020
jobTracker=localhost:8032
queueName=default
examplesRoot=examples

oozie.wf.application.path=${nameNode}/user/${user.name}/${examplesRoot}/apps/map-
l
outputDir=map-reduce
```

Defining the Parameter

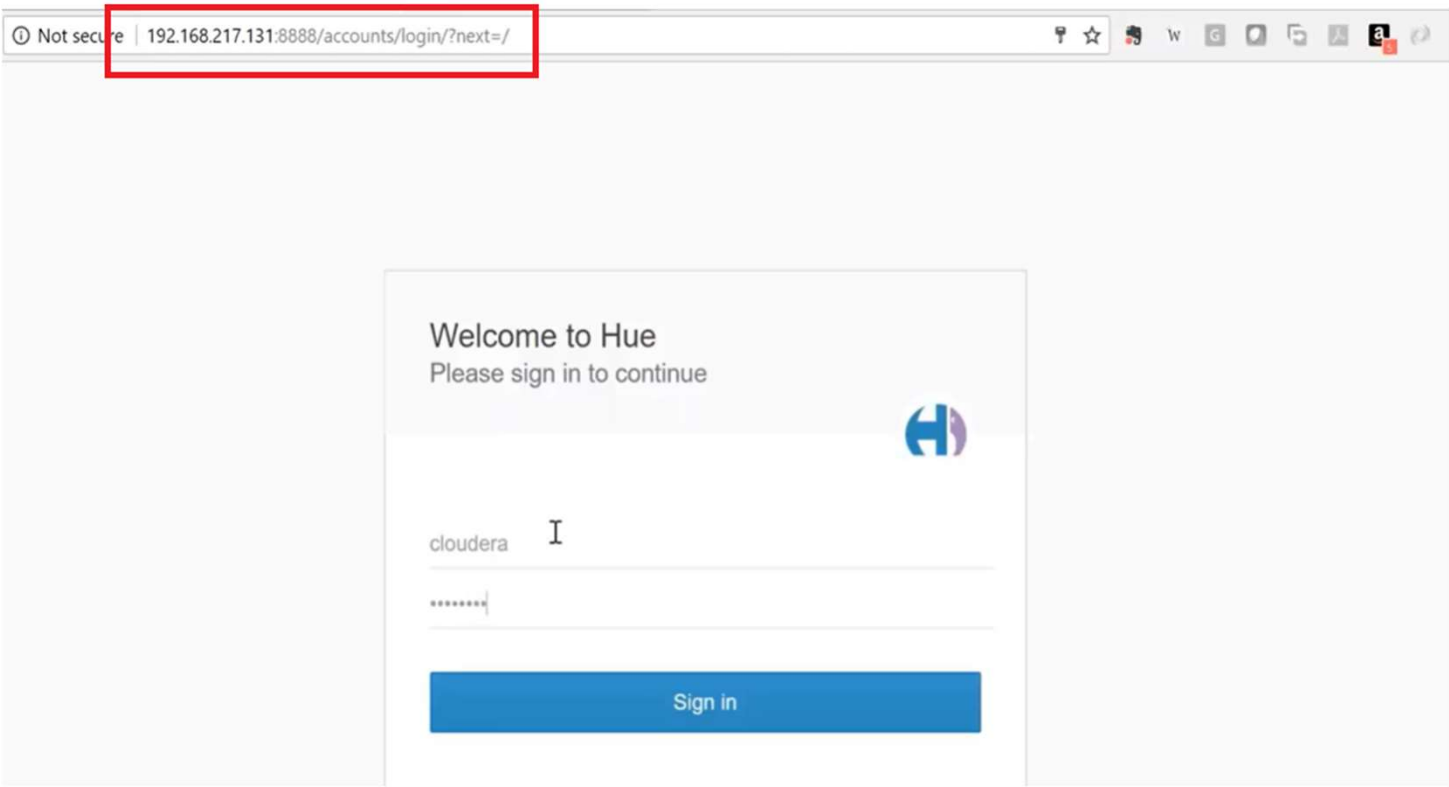


Submit the Job

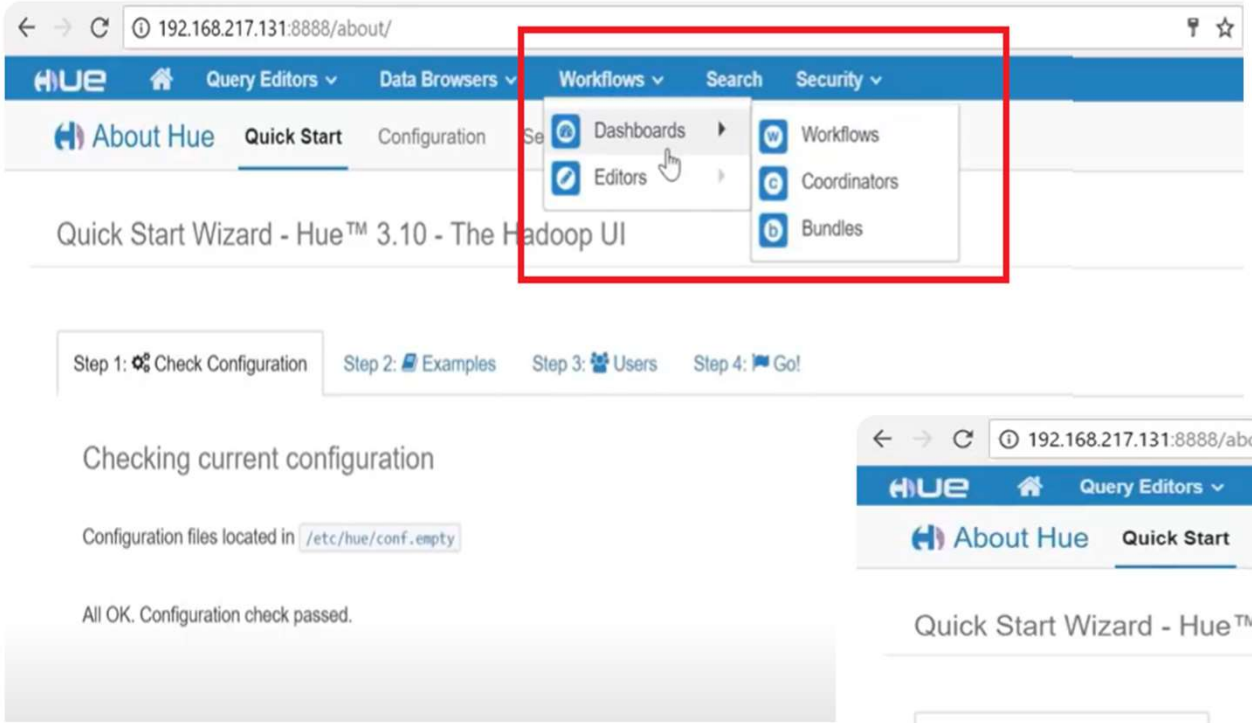
cloudera@quickstart:~/oozie

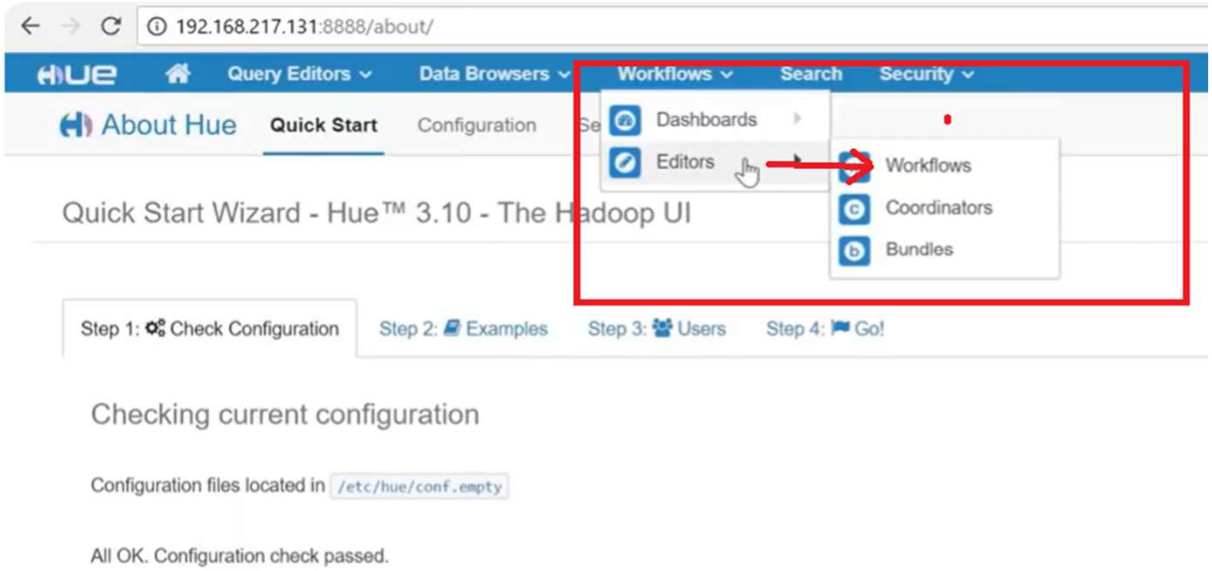
```
[cloudera@quickstart oozie]$ ls
examples  oozie-examples.tar.gz
[cloudera@quickstart oozie]$ oozie job -oozie http://localhost:11000/oozie -config examples/apps/m
p-reduce/job.properties -run
job: 0000000-171030184541100-oozie-oozi-W
[cloudera@quickstart oozie]$
```

Through HUE : Example

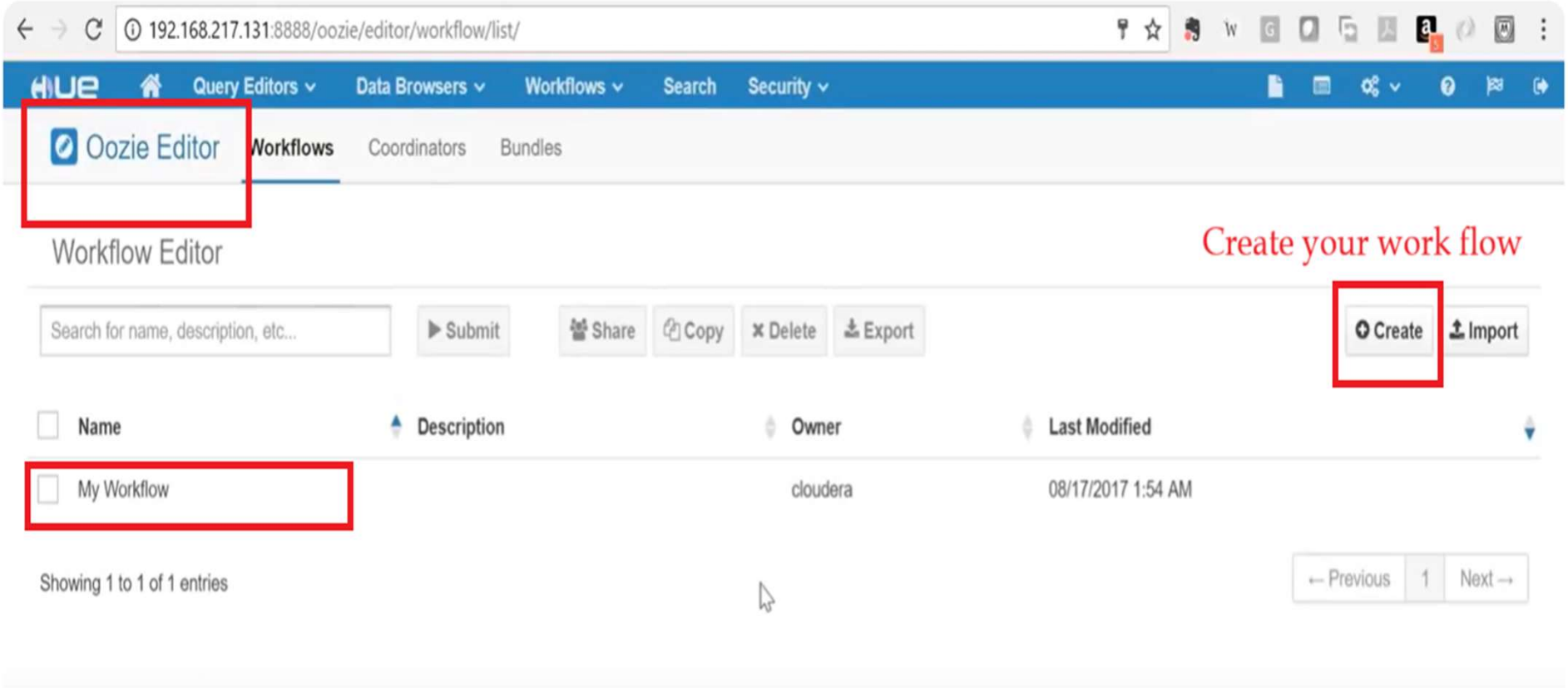


HUE Interface

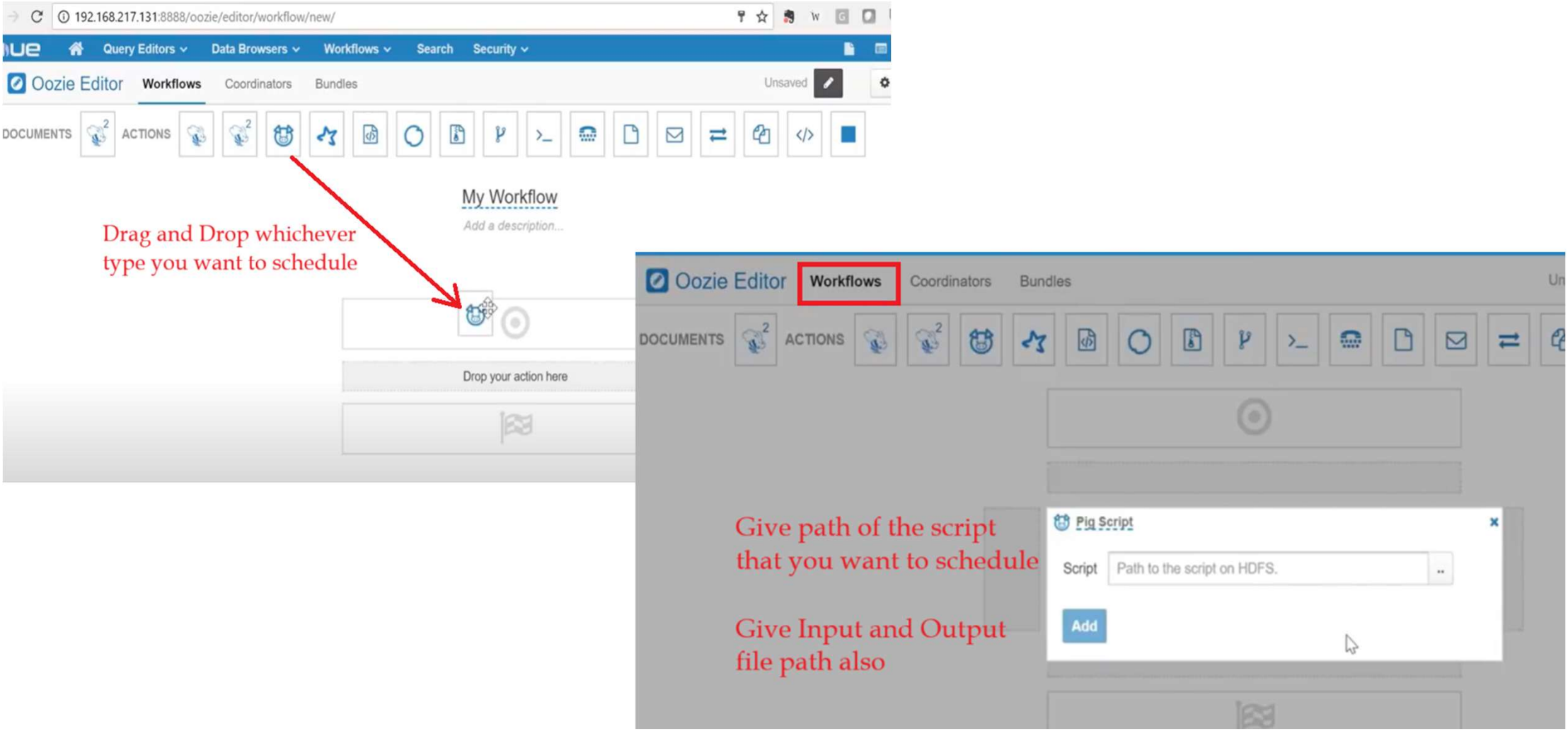




Oozie Editor



Working with Oozie Interface



Example : Simple Map-Reduce, Word Count Program

```
[cloudera@quickstart WordCount3]$ ls -lrt
total 28
-rw-rw-r-- 1 cloudera cloudera 2278 Sep  1  2015 Makefile
-rw-rw-r-- 1 cloudera cloudera 5171 Sep  1  2015 wordcount.jar
-rw-rw-r-- 1 cloudera cloudera   30 Sep  1  2015 stop_words.text
-rw-rw-r-- 1 cloudera cloudera 4713 Sep  1  2015 WordCount.java
drwxrwxr-x 3 cloudera cloudera 4096 May  3 11:08 build
[cloudera@quickstart WordCount3]$ hadoop fs -ls /wctest/
Found 2 items
drwxr-xr-x   - cloudera supergroup          0 2019-05-18 03:09 /wctest/WordCount3
drwxr-xr-x   - cloudera supergroup          0 2019-05-18 03:12 /wctest/input
[cloudera@quickstart WordCount3]$ hadoop fs -ls /wctest/input
Found 1 items
-rw-r--r--   1 cloudera supergroup         17 2019-05-18 03:12 /wctest/input/file.txt
[cloudera@quickstart WordCount3]$ hadoop fs -cat /wctest/input/file.txt
hello how are you[cloudera@quickstart WordCount3]$
```

Word Count Jar FIle

Input File

1

File Content

```
            Failed Shuffles=0
            Merged Map outputs=1
            GC time elapsed (ms)=198
            CPU time spent (ms)=1370
            Physical memory (bytes) snapshot=428949504
            Virtual memory (bytes) snapshot=3015155712
            Total committed heap usage (bytes)=418385920
    Shuffle Errors
            BAD_ID=0
            CONNECTION=0
            IO_ERROR=0
            WRONG_LENGTH=0
            WRONG_MAP=0
            WRONG_REDUCE=0
    File Input Format Counters
            Bytes Read=17
    File Output Format Counters
            Bytes Written=26
```

Successful Execution

2

```
            WRONG_REDUCE=0
    File Input Format Counters
            Bytes Read=17
    File Output Format Counters
            Bytes Written=26
[cloudera@quickstart WordCount3]$ hadoop fs -ls /wctest/
Found 3 items
drwxr-xr-x   - cloudera supergroup          0 2019-05-18 03:09 /wctest/WordCount3
drwxr-xr-x   - cloudera supergroup          0 2019-05-18 03:12 /wctest/input
drwxr-xr-x   - cloudera supergroup          0 2019-05-18 04:49 /wctest/output
[cloudera@quickstart WordCount3]$ hadoop fs -ls /wctest/output
Found 2 items
-rw-r--r--   1 cloudera supergroup          0 2019-05-18 04:49 /wctest/output/_SUCCESS
-rw-r--r--   1 cloudera supergroup         26 2019-05-18 04:49 /wctest/output/part-r-00000
```

3

Output Directory created on successful execution

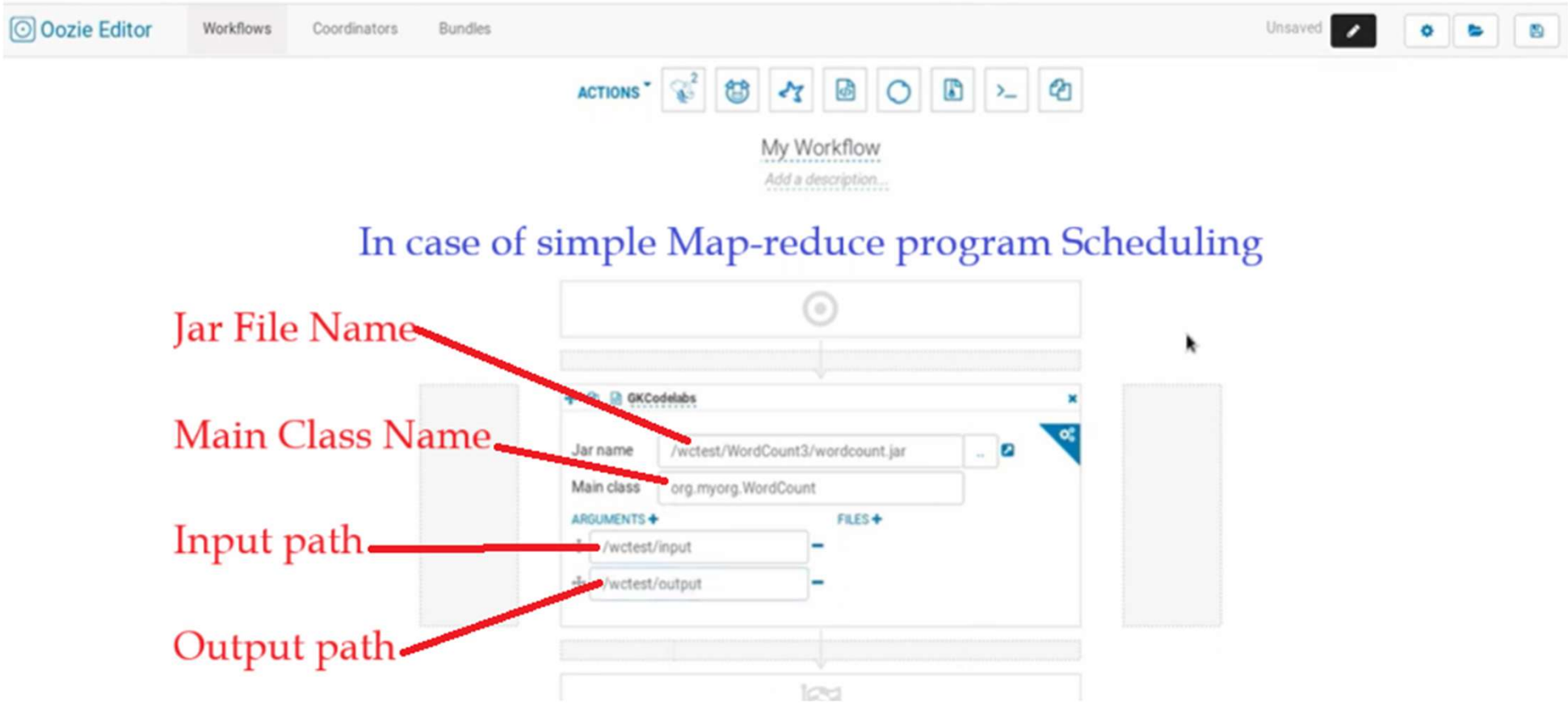
Part file in Output directory

```
[cloudera@quickstart WordCount3]$ hadoop fs -ls /wctest/output
Found 2 items
-rw-r--r--   1 cloudera supergroup          0 2019-05-18 04:49 /wctes
-rw-r--r--   1 cloudera supergroup         26 2019-05-18 04:49 /wctes
[cloudera@quickstart WordCount3]$ hadoop fs -cat /wctest/output/part*
are     1
hello   1
how     1
you     1
[cloudera@quickstart WordCount3]$
```

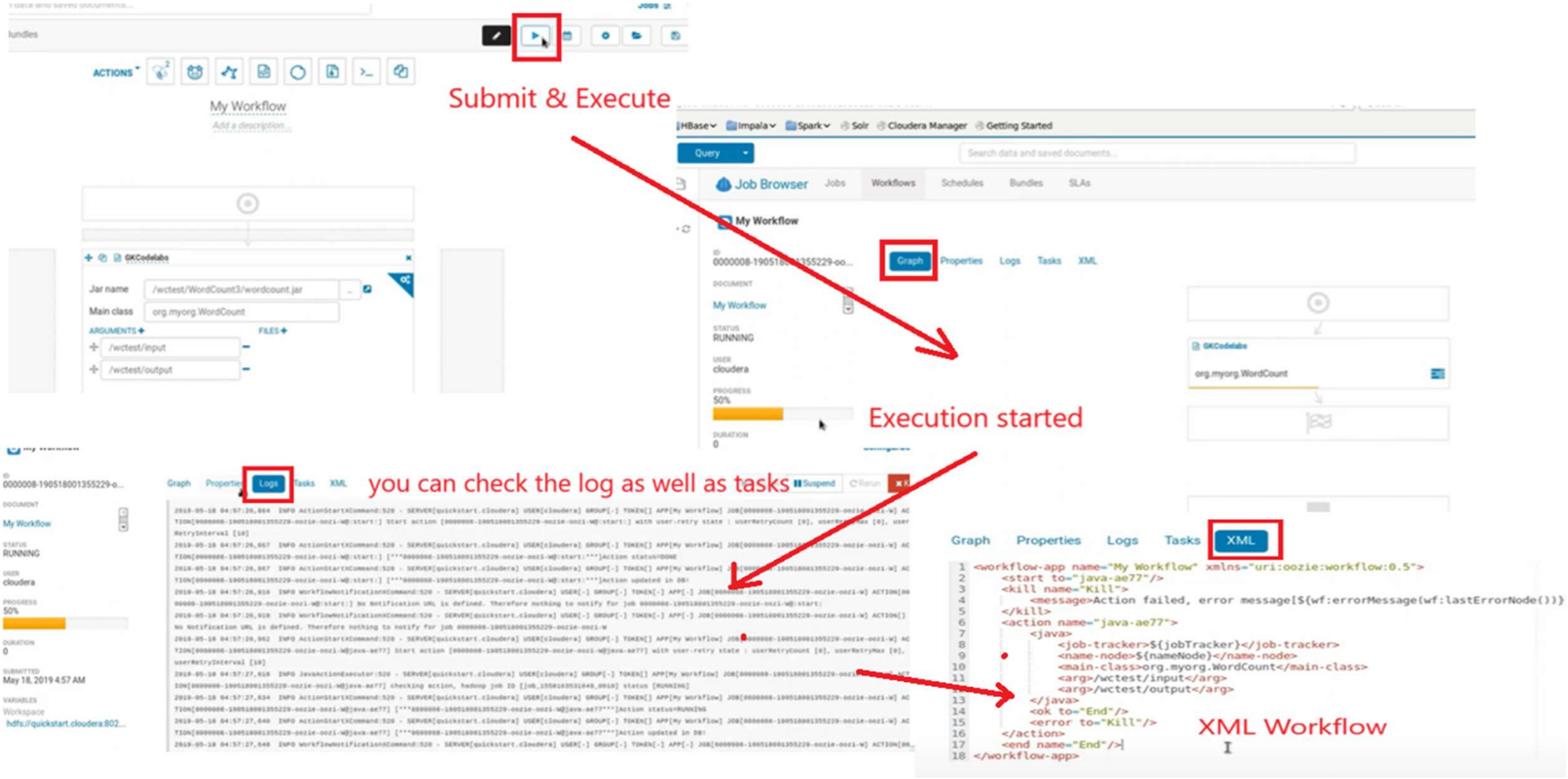
Display Output from Part File

4

Scheduling simple Map-Reduce through UI



Submit, Execute & check O/p



O/p

Query | Search data and saved documents... | Jobs

Job Browser | Jobs | Workflows | Schedules | Bundles | SLAs

user:cloudera | Succeeded | Running | Failed in the last 7 days

Id	Name	User	Type	Status	Progress	Group	Started	Duration
application_1558163531849_0011	wordcount	cloudera	MAPREDUCE	SUCCEEDED	100	root.cloudera	May 18, 2019 4:57 AM	27.86s
application_1558163531849_0010	oozie:launcher:T=java:W=My Workflow:A=java-ae77:ID=0000008-190518001355229-oozie-oozi-W	cloudera	MAPREDUCE	SUCCEEDED	100	root.cloudera	May 18, 2019 4:57 AM	51.53s
application_1558163531849_0009	wordcount	cloudera	MAPREDUCE	SUCCEEDED	100	root.cloudera	May 18, 2019 4:49 AM	24.57s
application_1558163531849_0008	wordcount	cloudera	MAPREDUCE	SUCCEEDED	100	root.cloudera	May 18, 2019 3:40 AM	23.8s
application_1558163531849_0007	wordcount	cloudera	MAPREDUCE	SUCCEEDED	100	root.cloudera	May 18, 2019 3:28 AM	36.53s

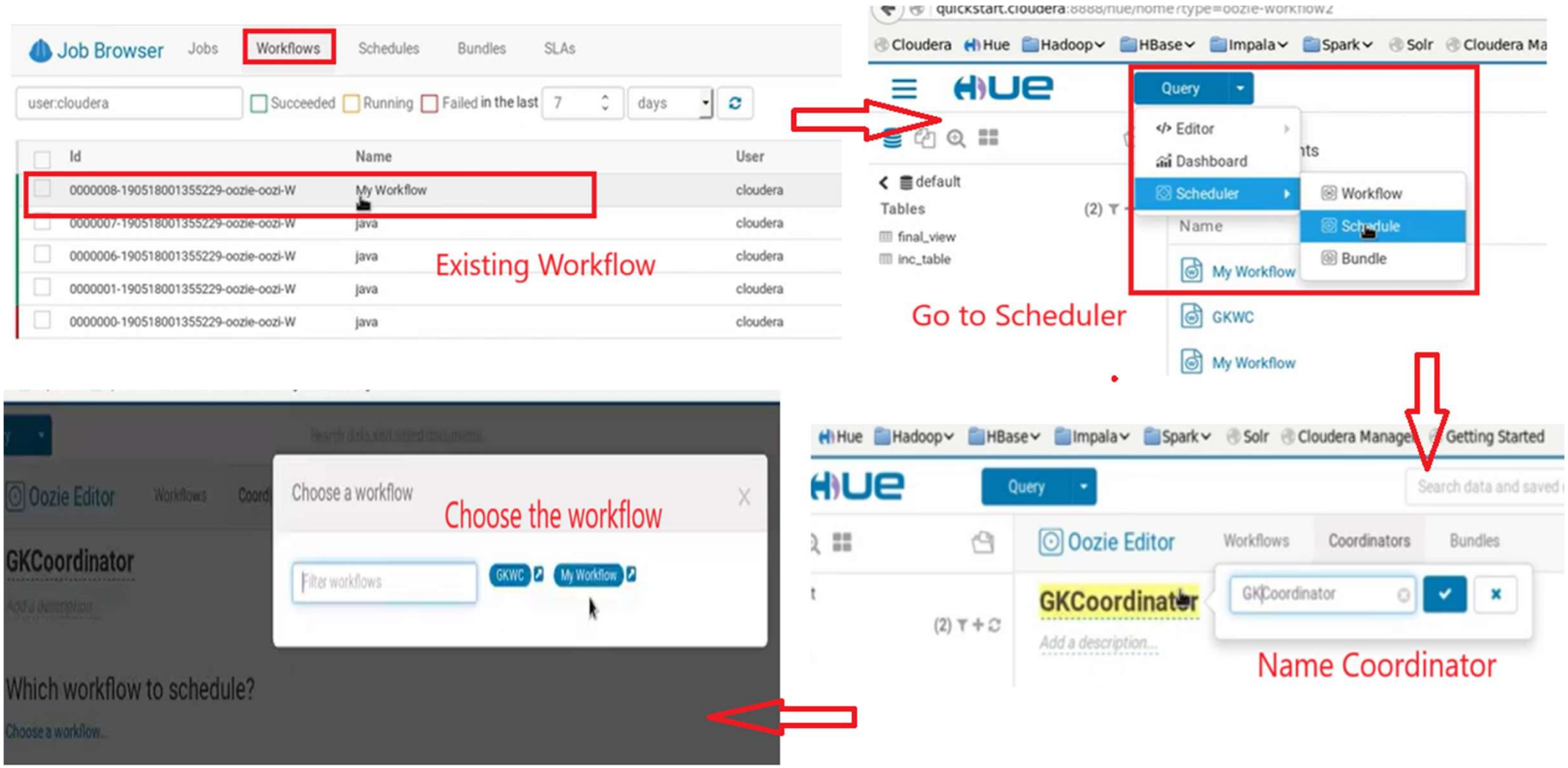
```
[cloudera@quickstart WordCount3]$ hadoop fs -rm -r /wctest/output
Deleted /wctest/output
[cloudera@quickstart WordCount3]$ hadoop fs -ls /wctest/output
Found 2 items
-rw-r--r--   1 cloudera supergroup          0 2019-05-18 04:58 /wctest/output/_SUCCESS
-rw-r--r--   1 cloudera supergroup         26 2019-05-18 04:58 /wctest/output/part-r-00000
[cloudera@quickstart WordCount3]$ hadoop fs -cat /wctest/input/file.txt
hello how are you[cloudera@quickstart WordCount3]$ hadoop fs -cat /wctest/output/part*
are     1
hello   1
how     1
you     1
[cloudera@quickstart WordCount3]$
```

Remove previous data ..if any

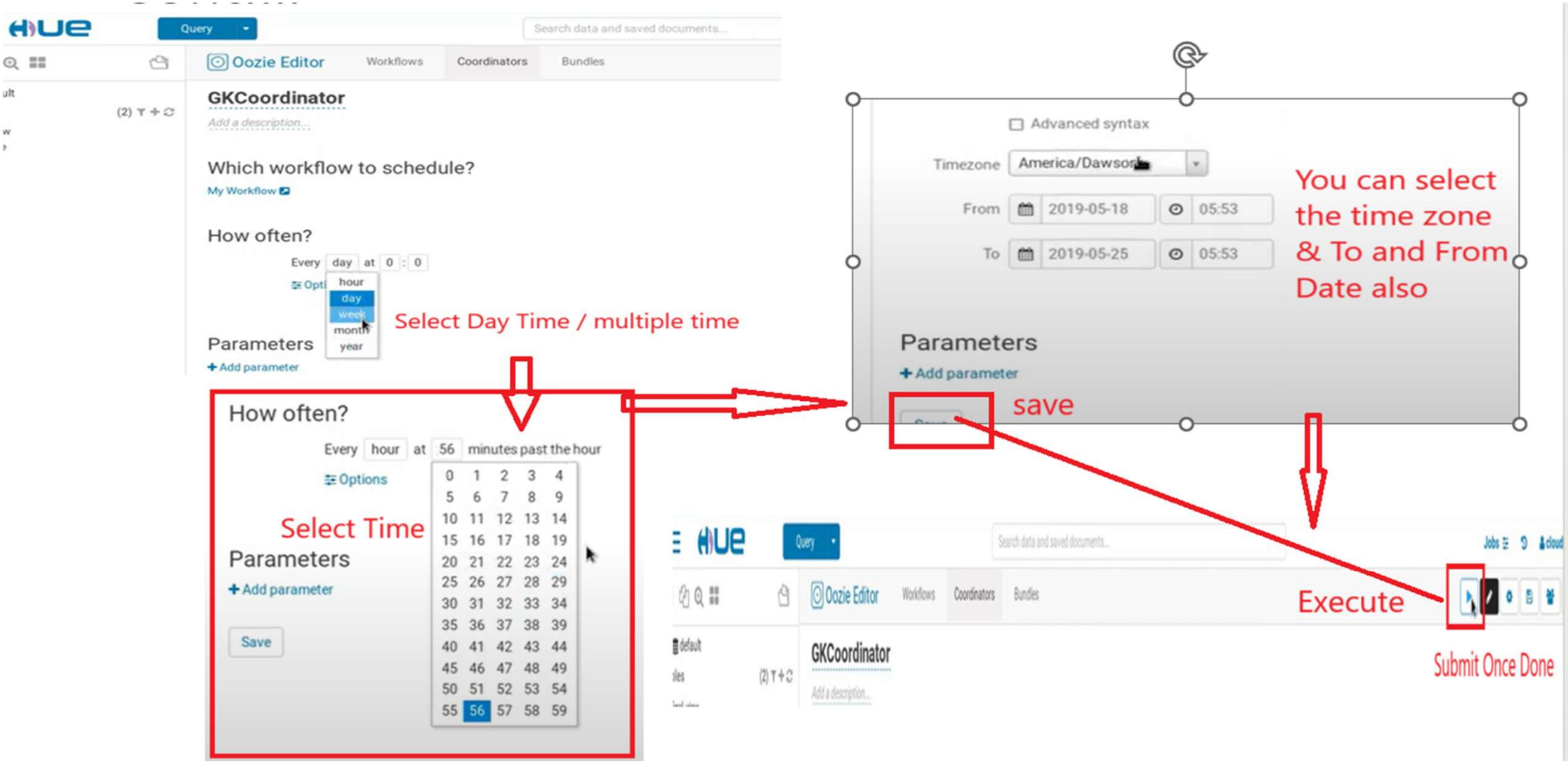
Part file created successfully

Data in Output Part File

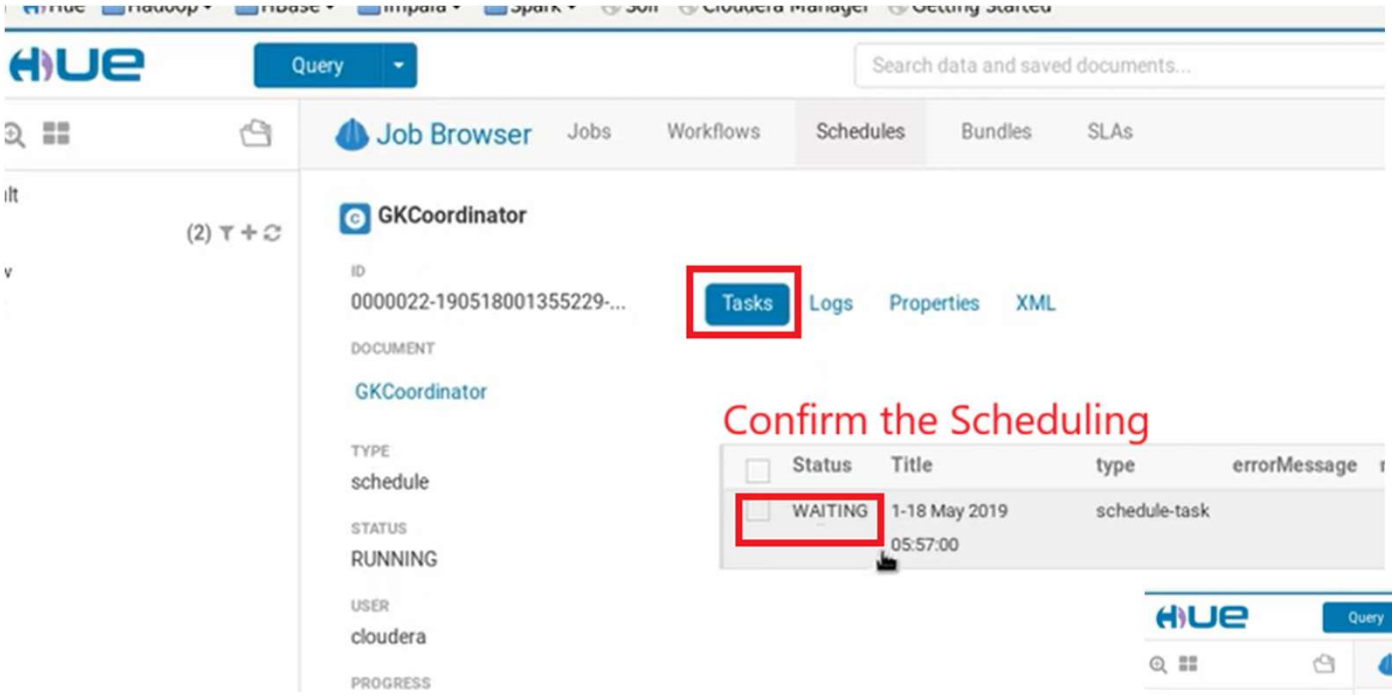
Scheduling using Start & End Time through Coordinator

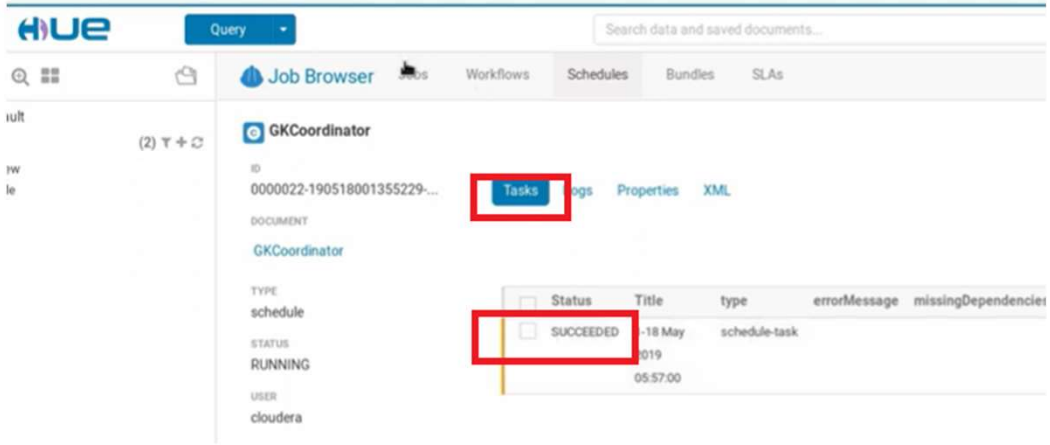


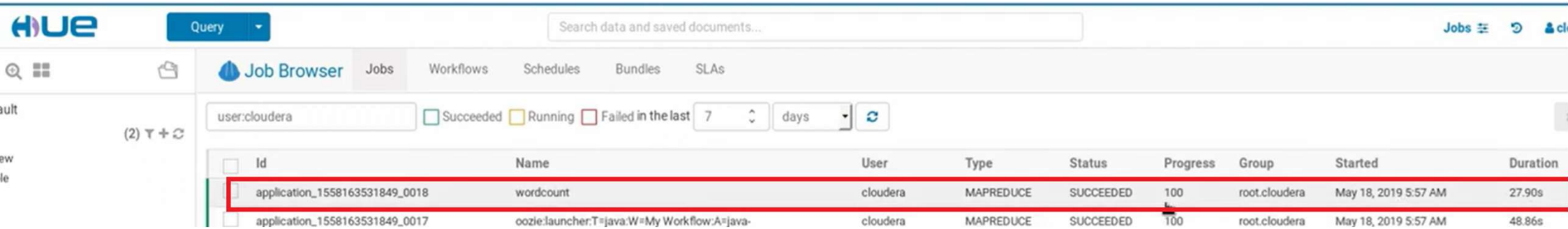
Cont....


Query
Search data and saved documents...
Oozie Editor
Workflows
Coordinators
Bundles
GKCoordinator
Add a description...
Which workflow to schedule?
My Workflow
How often?
Every day at 0 : 0
hour
day
week
month
year
Select Day Time / multiple time
Parameters
+ Add parameter
How often?
Every hour at 56 minutes past the hour
Options
Select Time
Parameters
+ Add parameter
Save
Advanced syntax
Timezone America/Dawson
From 2019-05-18 05:53
To 2019-05-25 05:53
You can select the time zone & To and From Date also
Parameters
+ Add parameter
save
Execute
Submit Once Done

Cont….


Query
Search data and saved documents...
Job Browser
Jobs
Workflows
Schedules
Bundles
SLAs
GKCoordinator
ID
0000022-190518001355229-...
Tasks
Logs
Properties
XML
DOCUMENT
GKCoordinator
TYPE
schedule
STATUS
RUNNING
USER
cloudera
PROGRESS
Confirm the Scheduling
Status
Title
type
errorMessage
WAITING
1-18 May 2019
05:57:00
schedule-task


Query
Job Browser
Workflows
Schedules
Bundles
SLAs
GKCoordinator
0000022-190518001355229-...
Tasks
Properties
XML
DOCUMENT
GKCoordinator
TYPE
schedule
STATUS
RUNNING
USER
cloudera
Status
Title
type
errorMessage
missingDependencies
SUCCEEDED
schedule-task
05:57:00




```
[cloudera@quickstart WordCount3]$ hadoop fs -rm -r /wctest/output
Deleted /wctest/output
[cloudera@quickstart WordCount3]$ hadoop fs -ls /wctest/
Found 3 items
drwxr-xr-x   - cloudera supergroup          0 2019-05-18 03:09 /wctest/WordCount3
drwxr-xr-x   - cloudera supergroup          0 2019-05-18 03:12 /wctest/input
drwxr-xr-x   - cloudera supergroup          0 2019-05-18 05:57 /wctest/output
[cloudera@quickstart WordCount3]$ hadoop fs -ls /wctest/output Output Directory created
Found 2 items
-rw-r--r--   1 cloudera supergroup          0 2019-05-18 05:57 /wctest/output/_SUCCESS
-rw-r--r--   1 cloudera supergroup         26 2019-05-18 05:57 /wctest/output/part-r-00000
[cloudera@quickstart WordCount3]$                                Output Part File Created
```