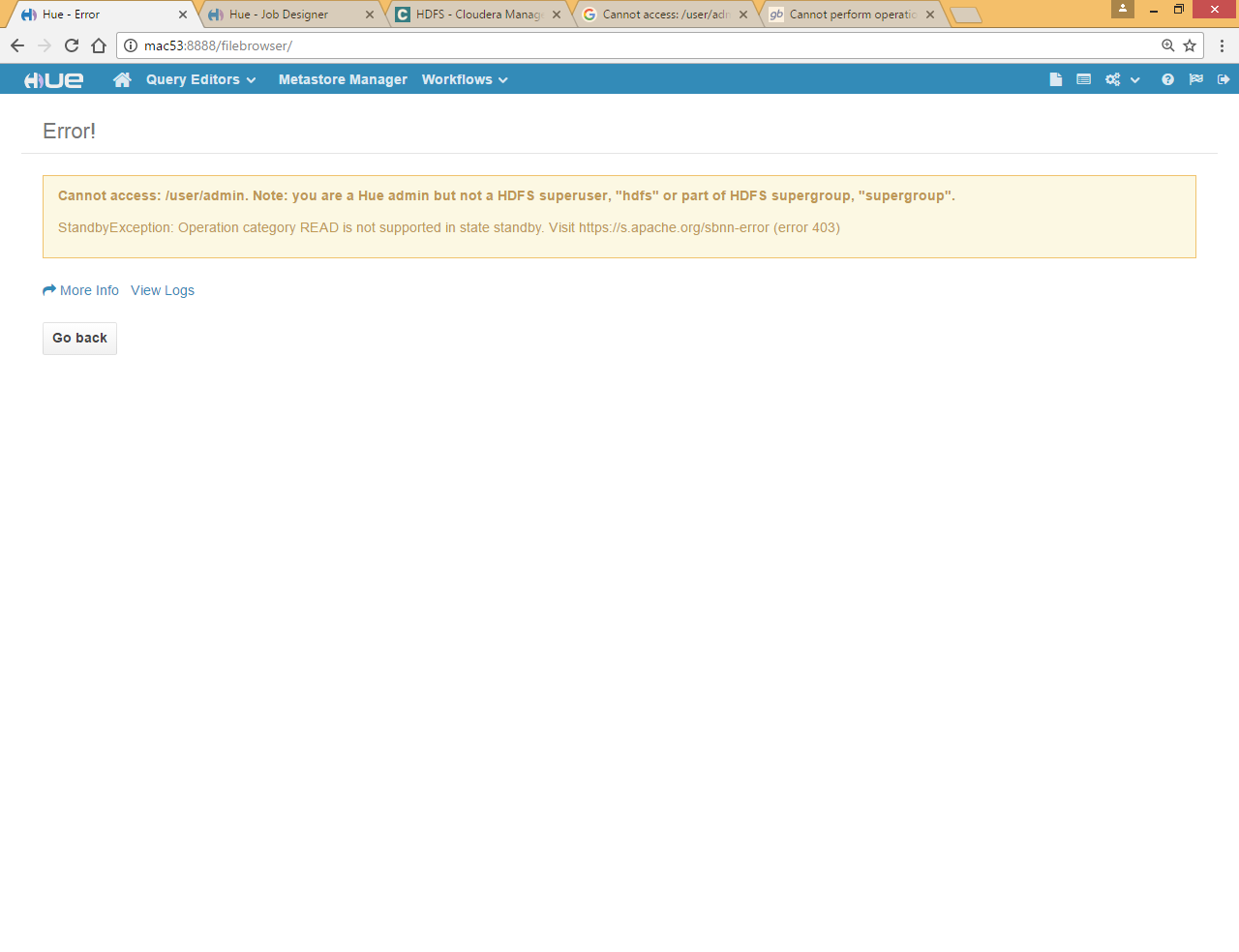
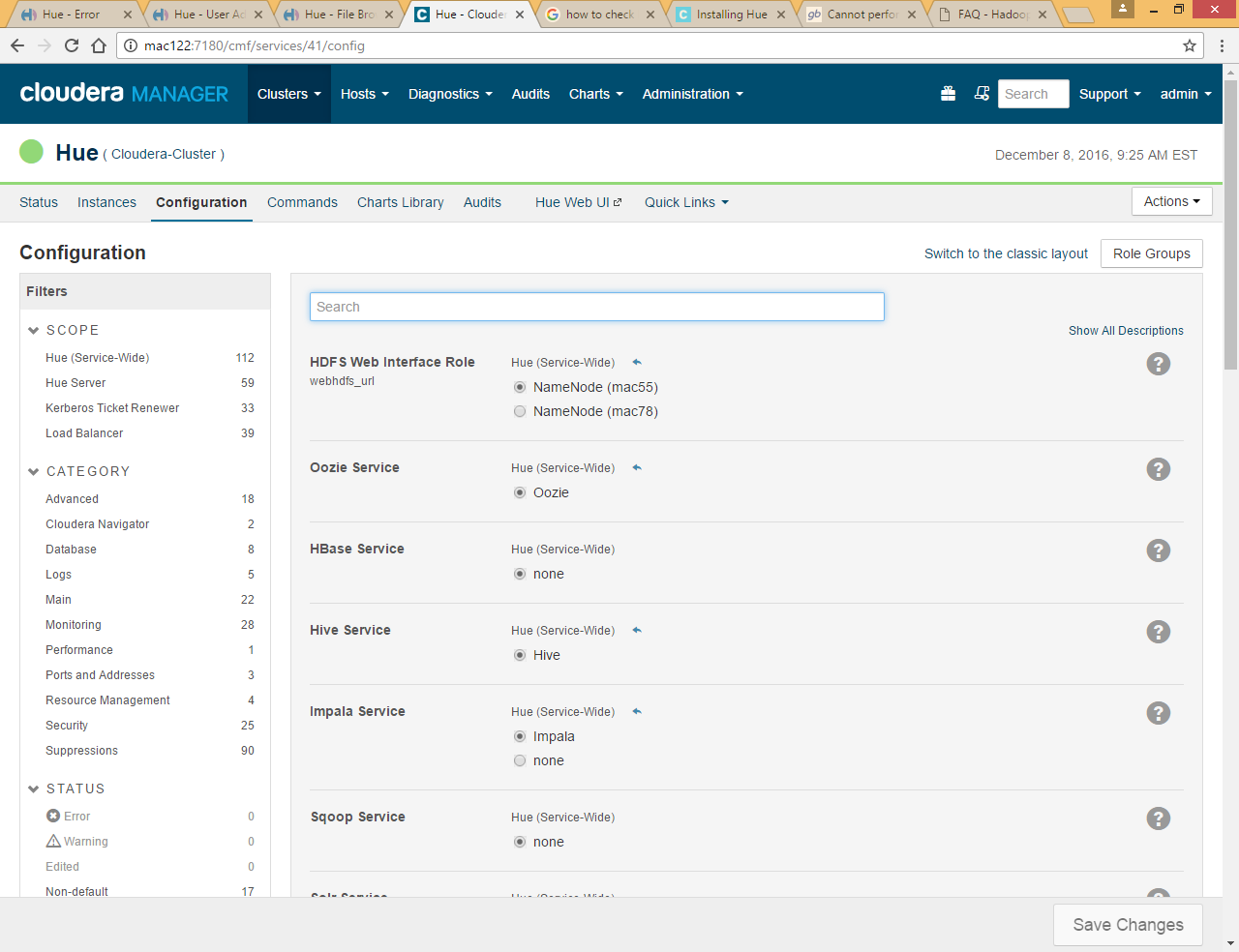
Login to HUE and check file browser

Getting issue in file browser

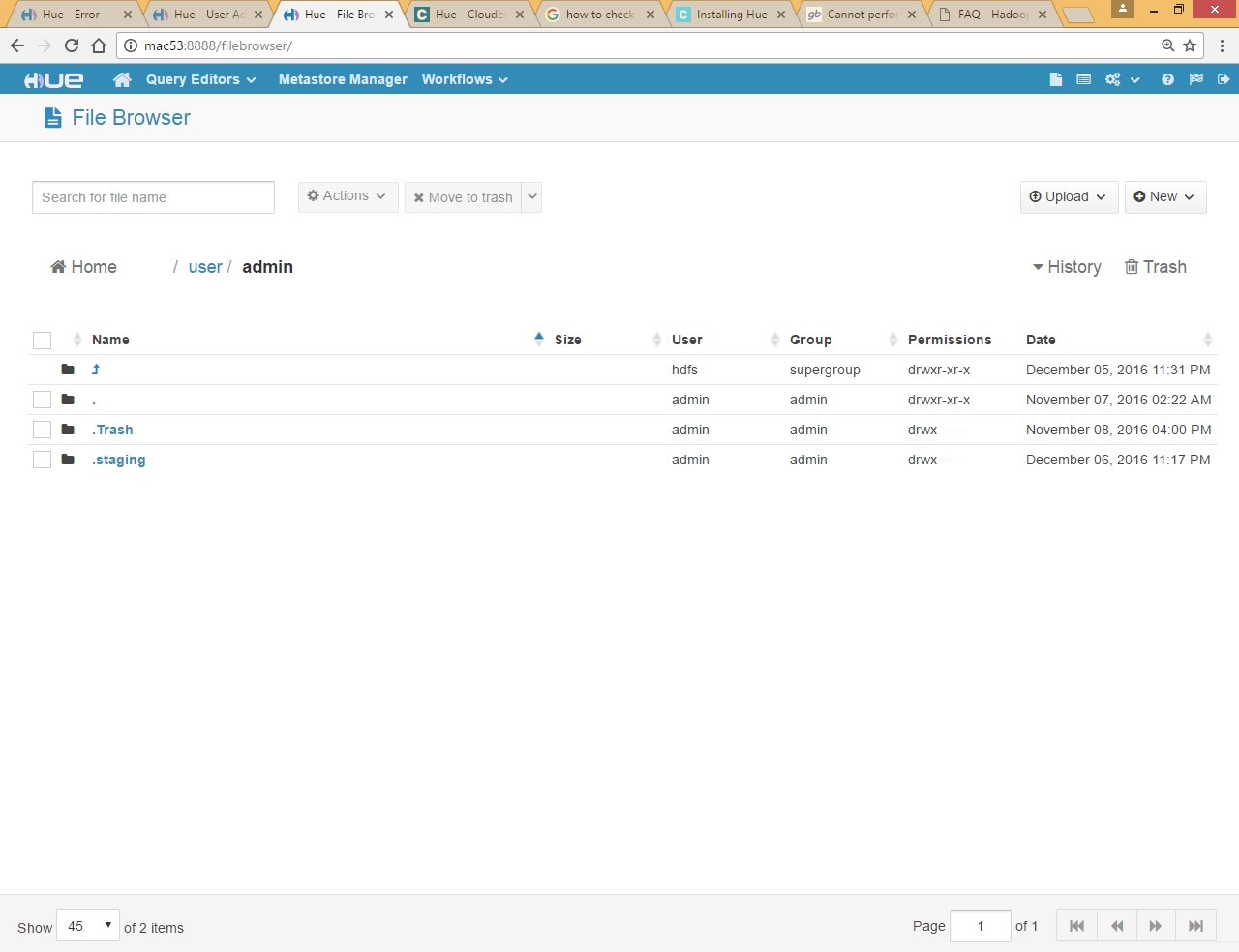


This issue occurs when we enable HA on namenode , HUE is not pointing to Active name node instead pointing to stand by name node

Go to Cloudera-manager >> hue>>configuration and change Namenode

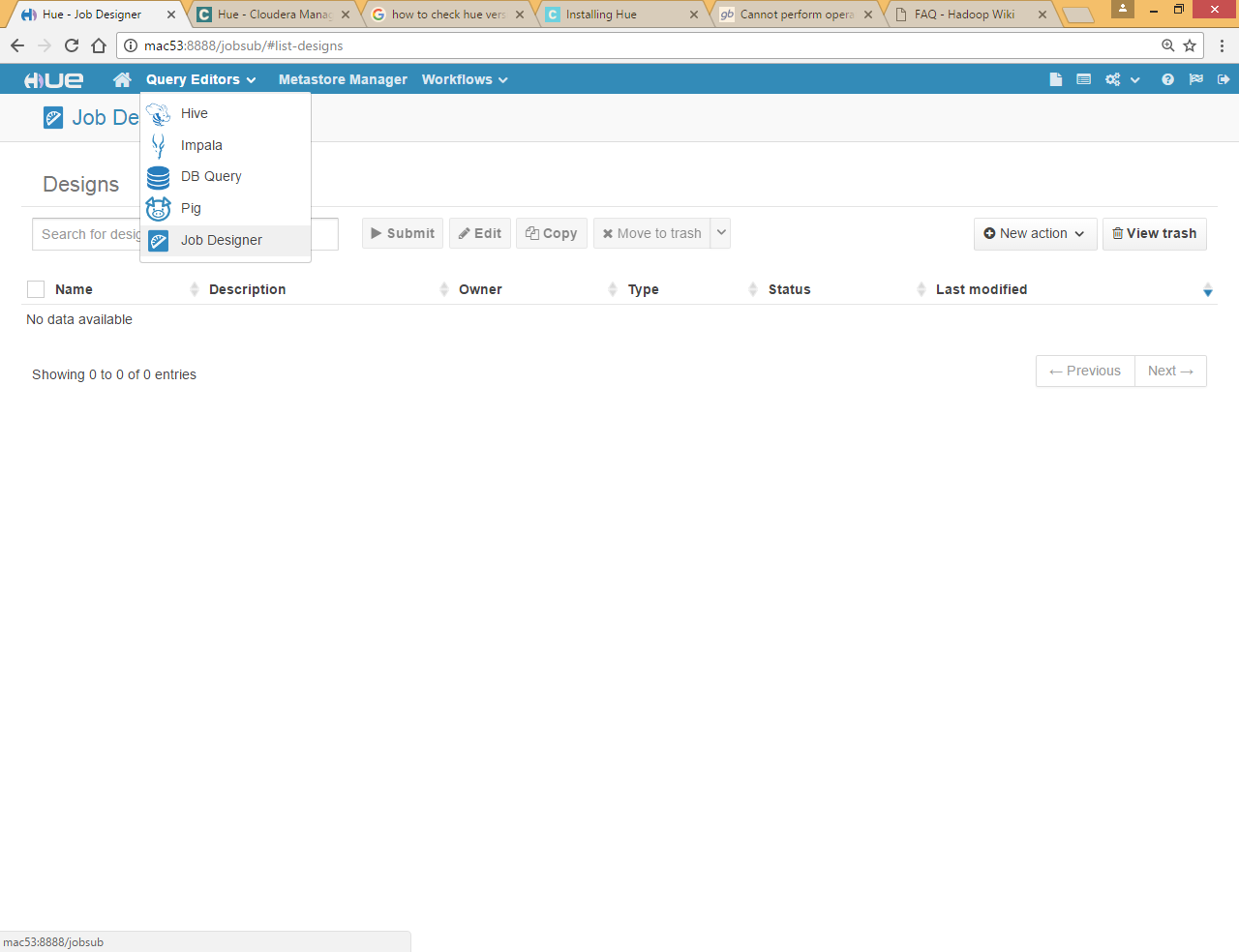


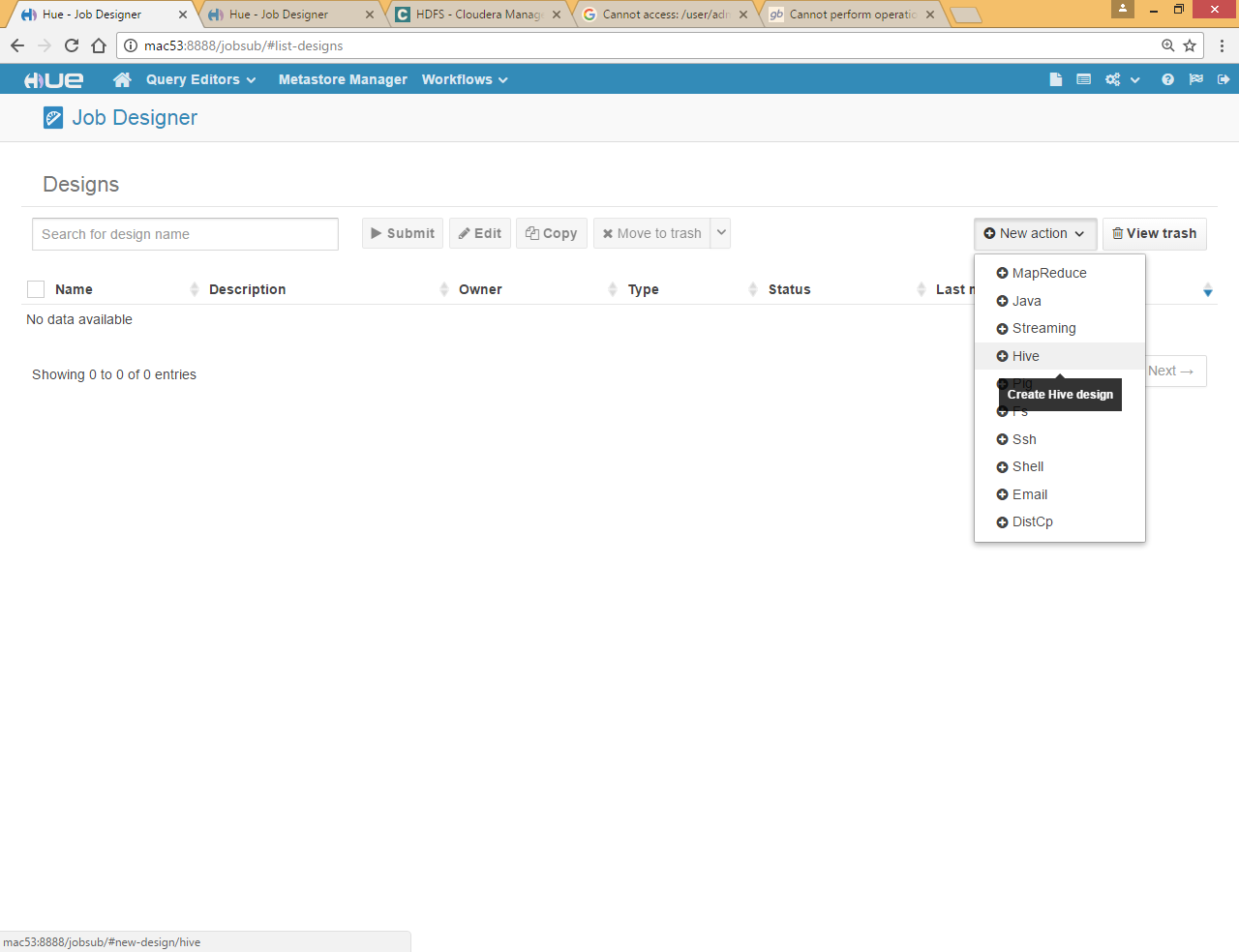
Restart hue

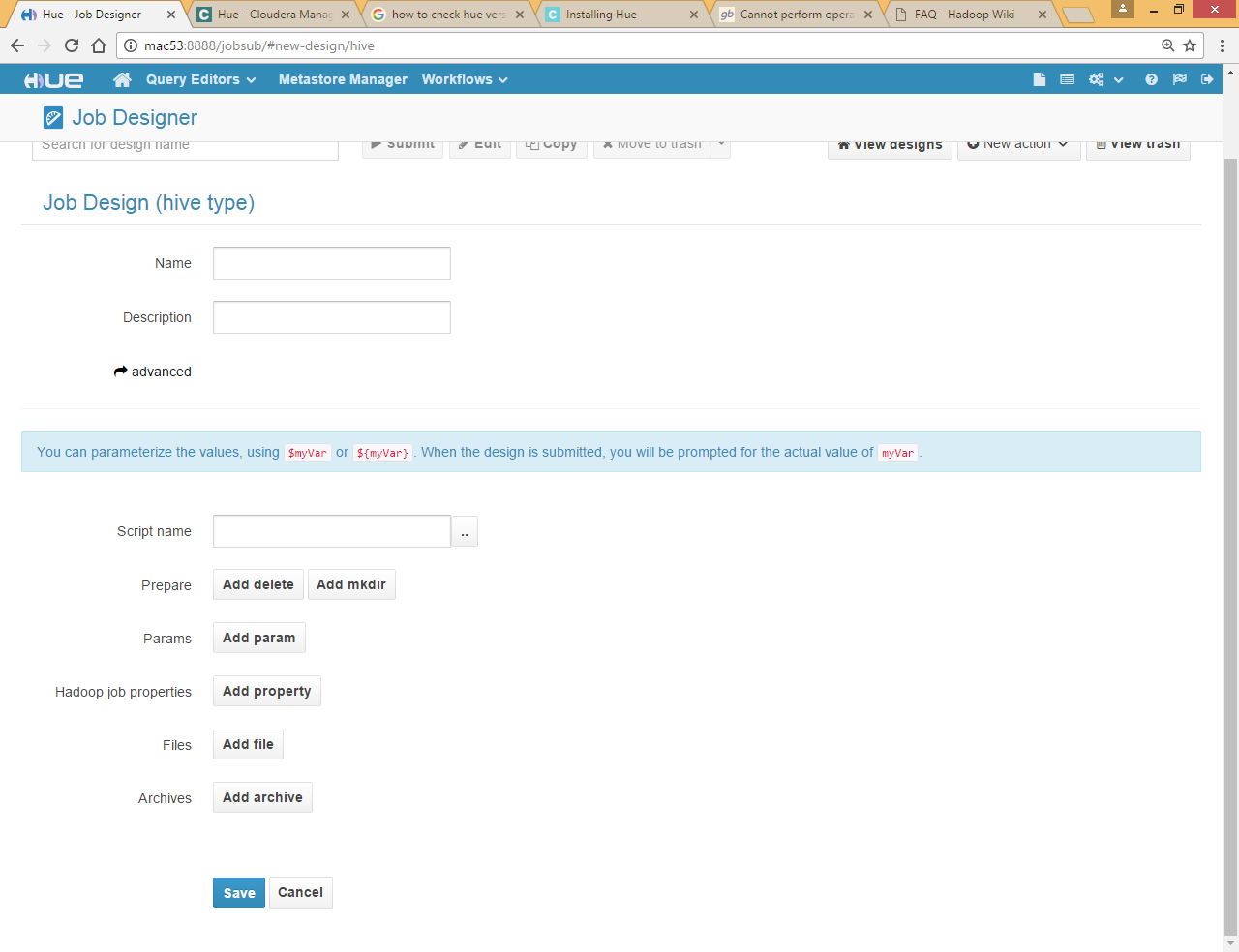


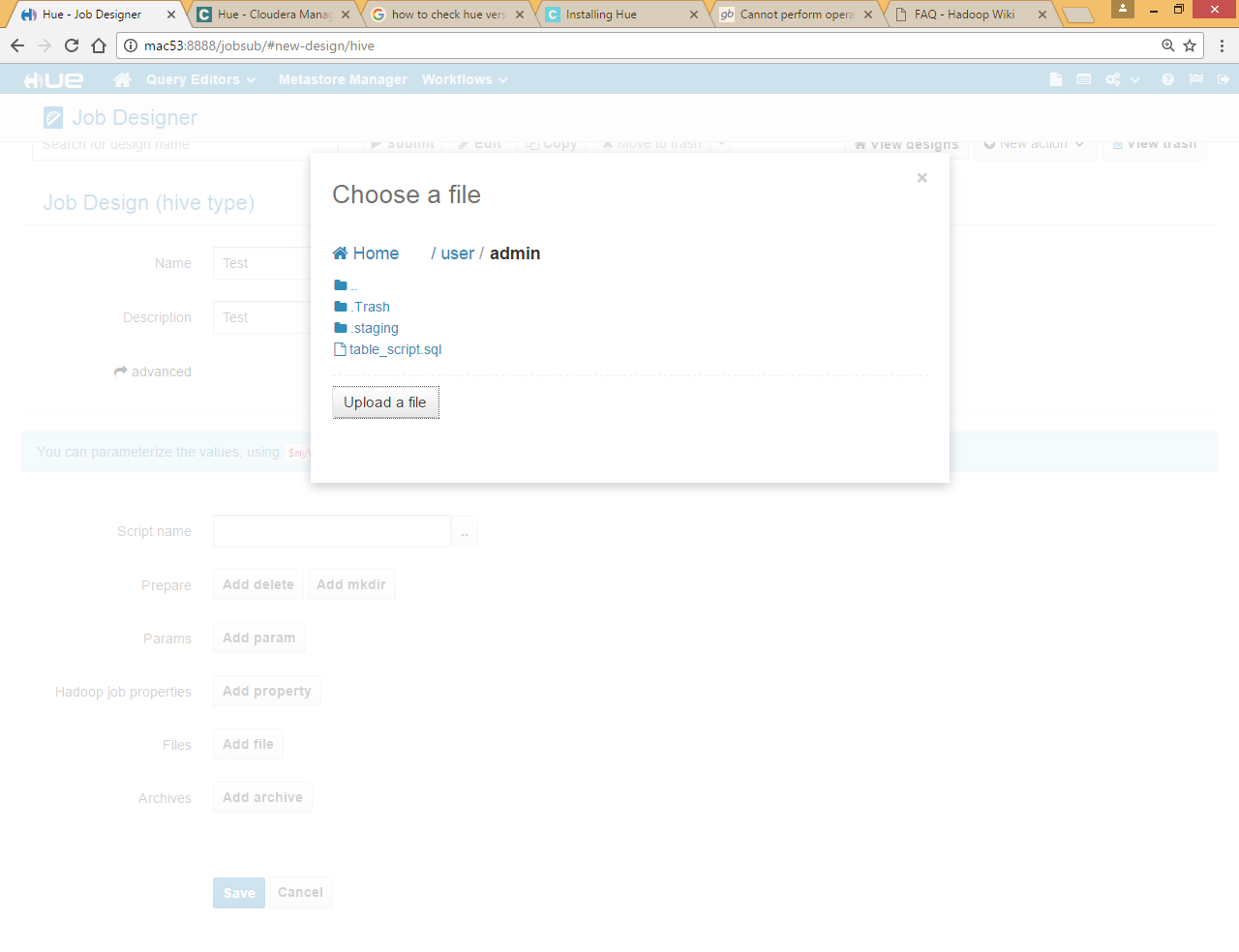
Now you can access file browser

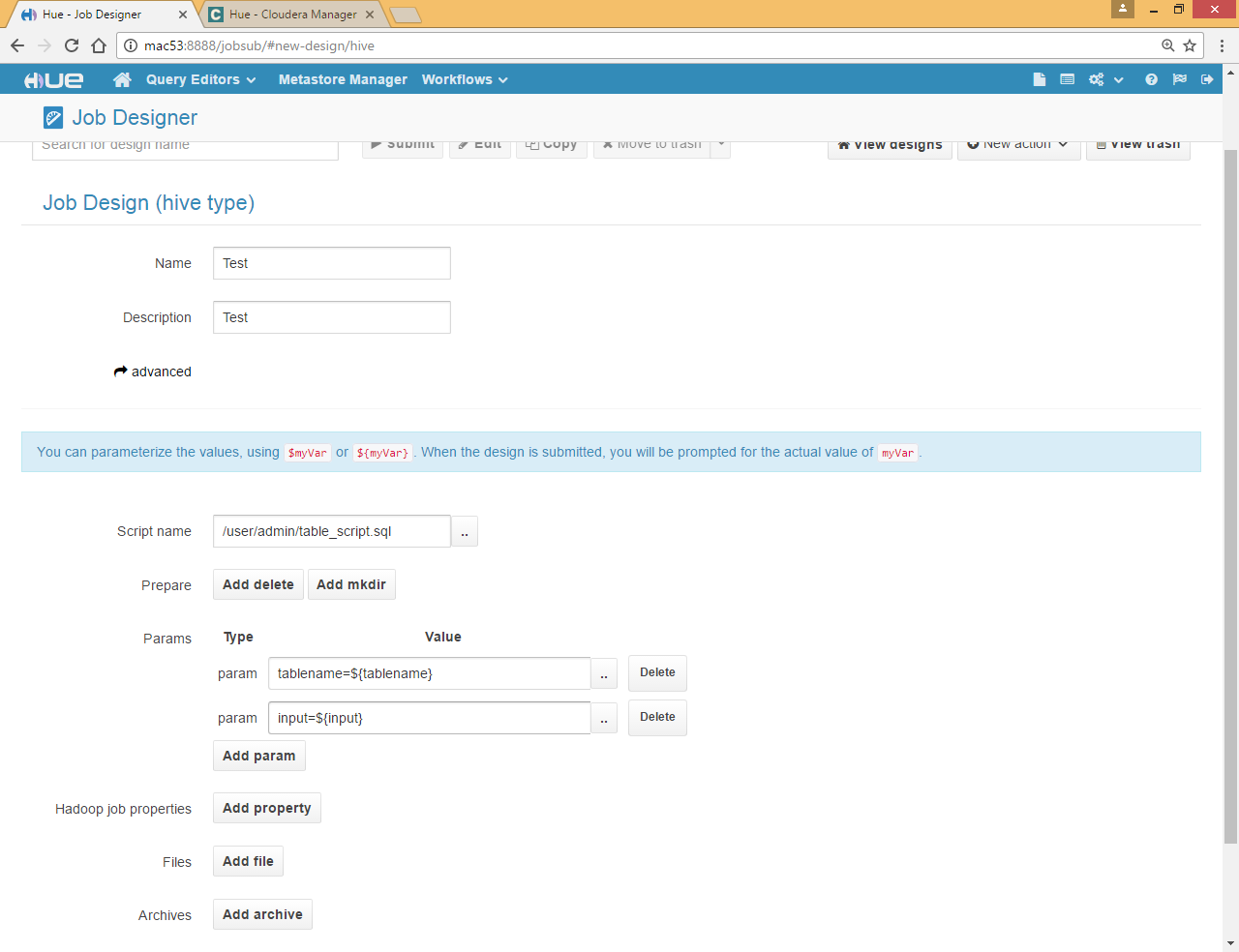
# Create workflow by Job Designer

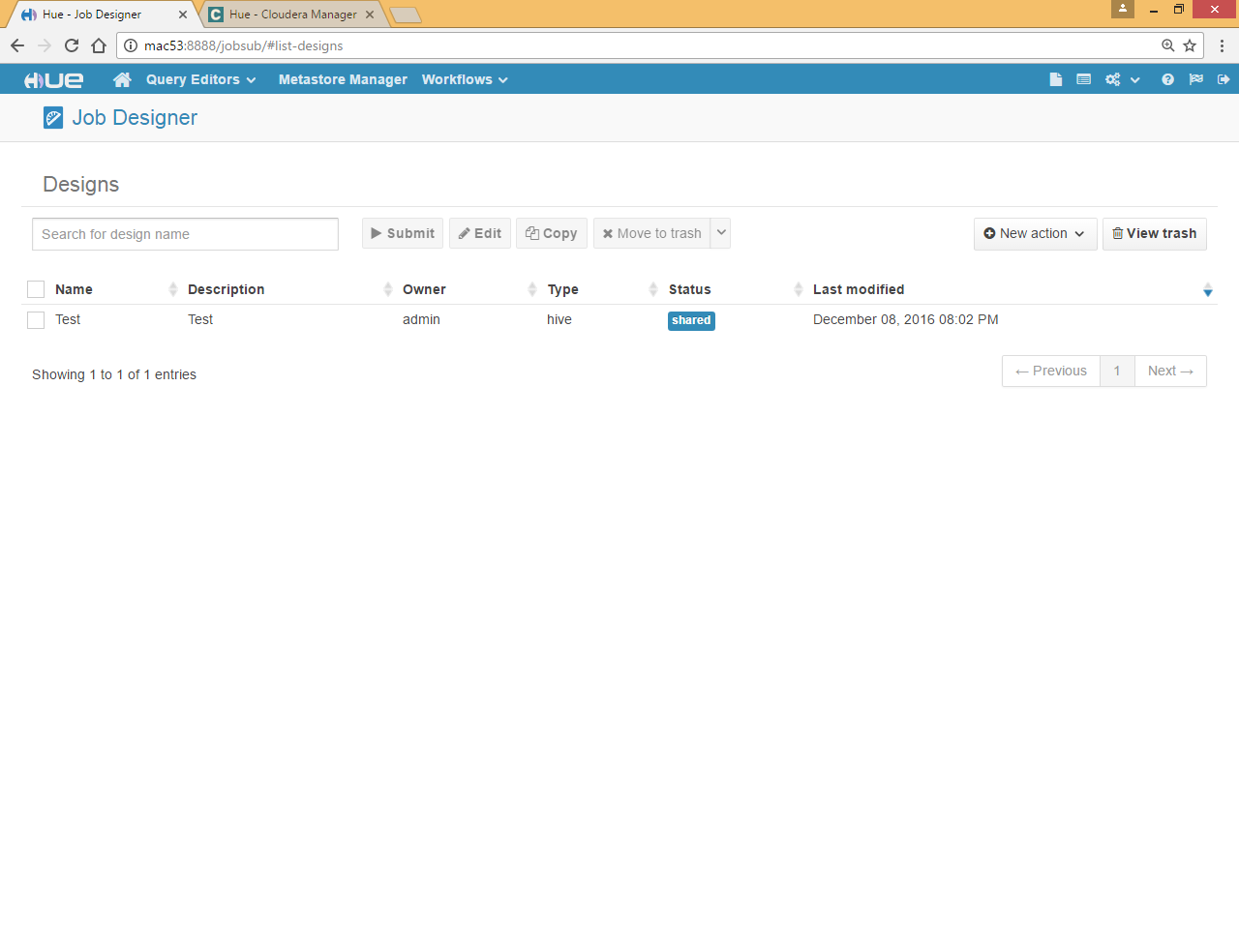


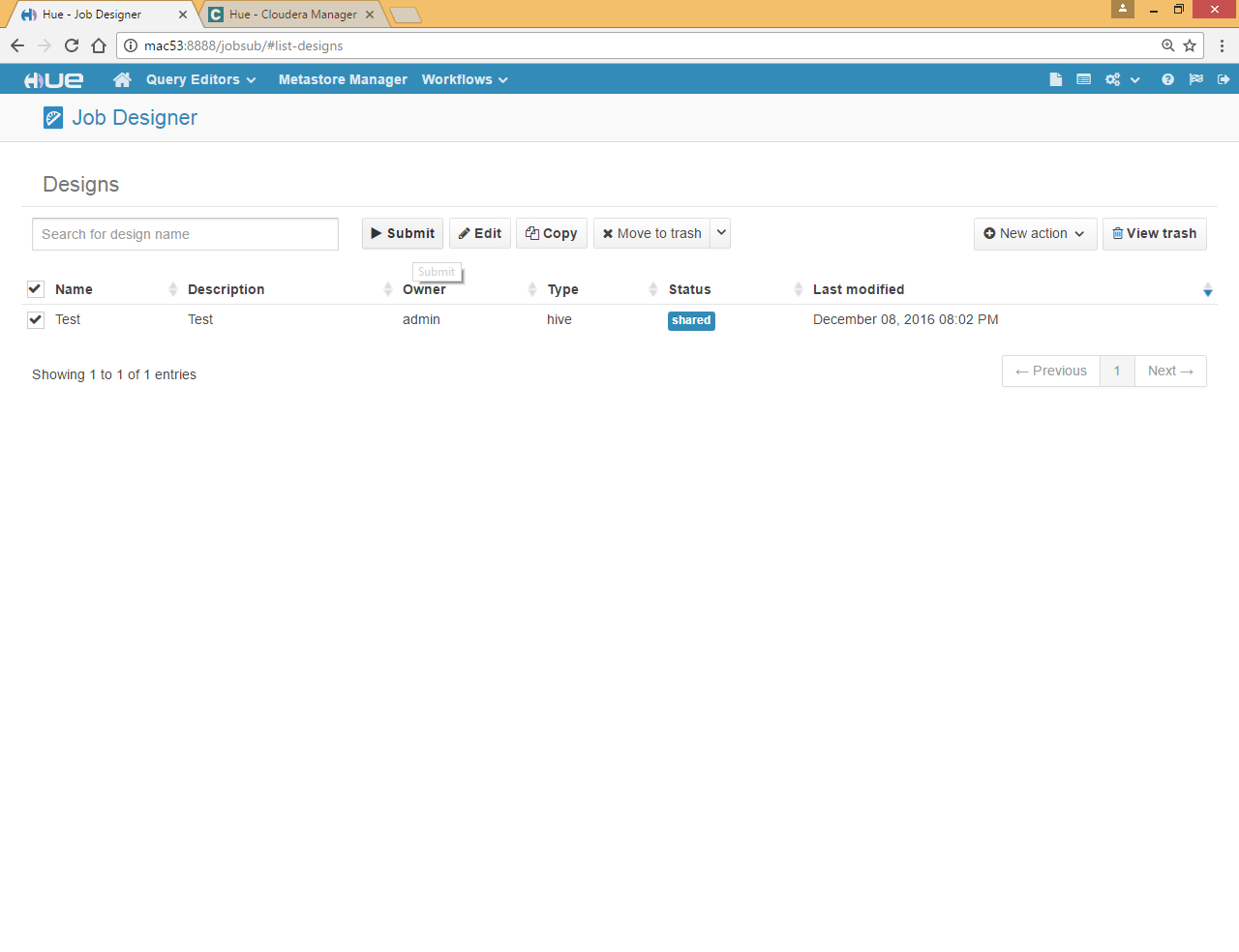




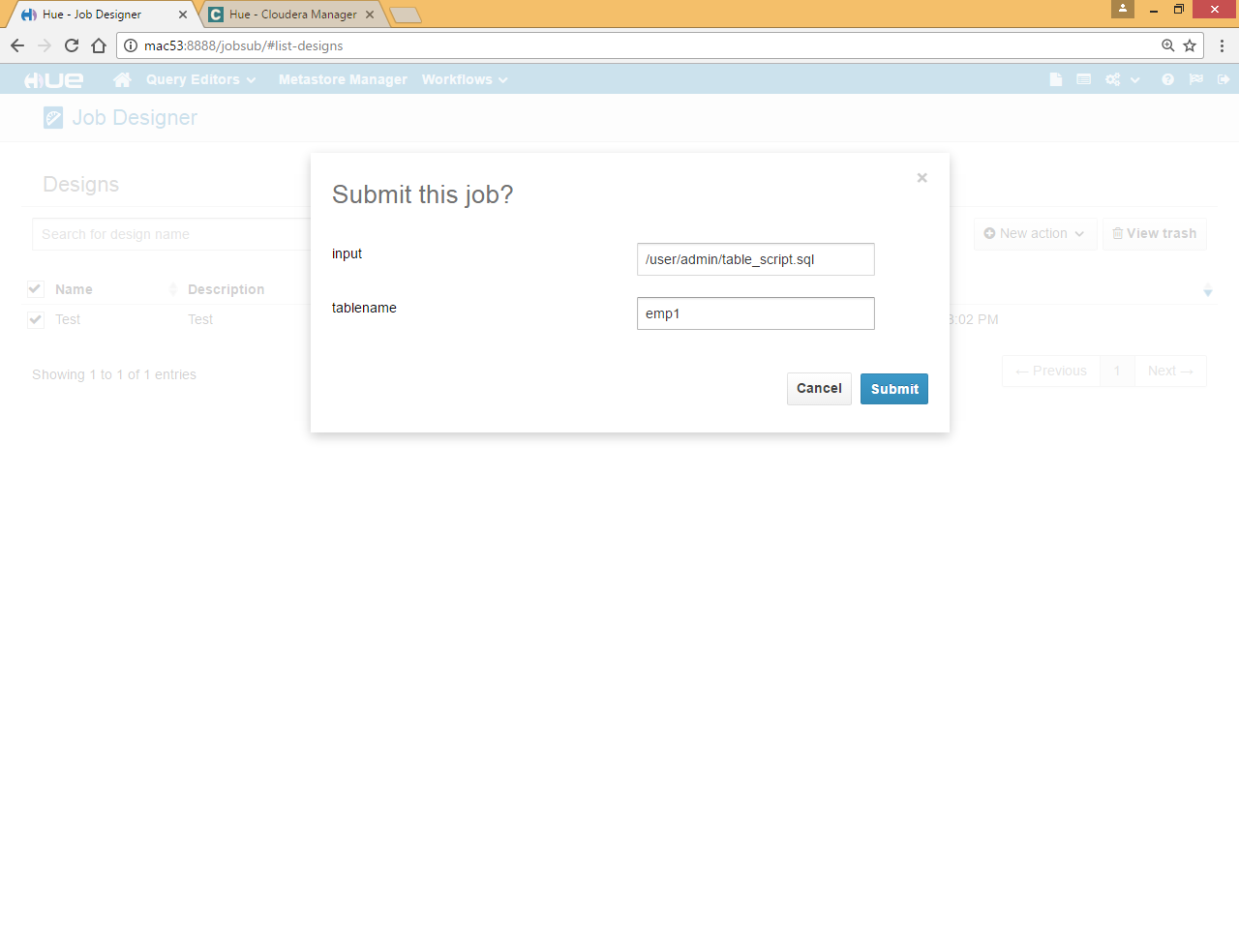


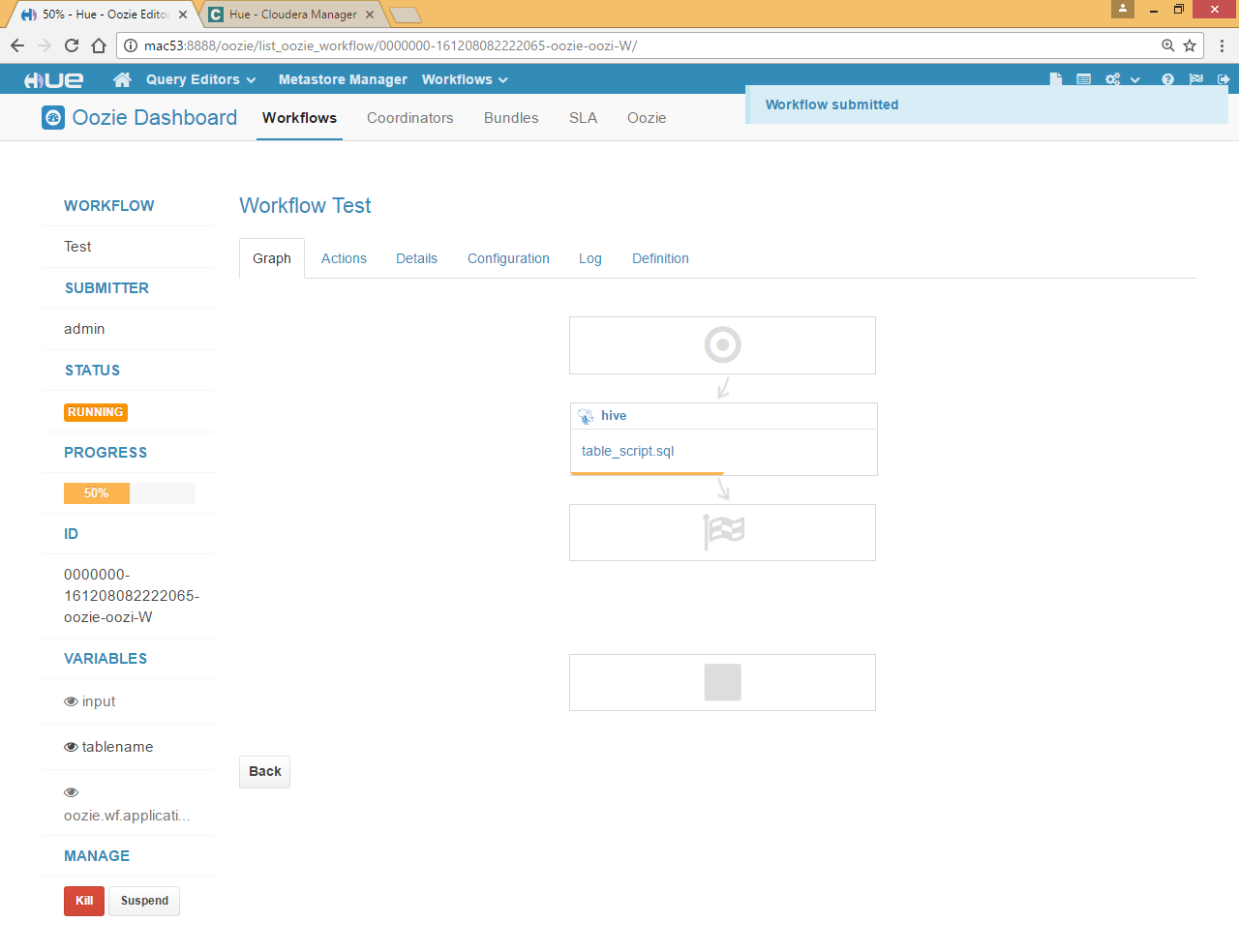


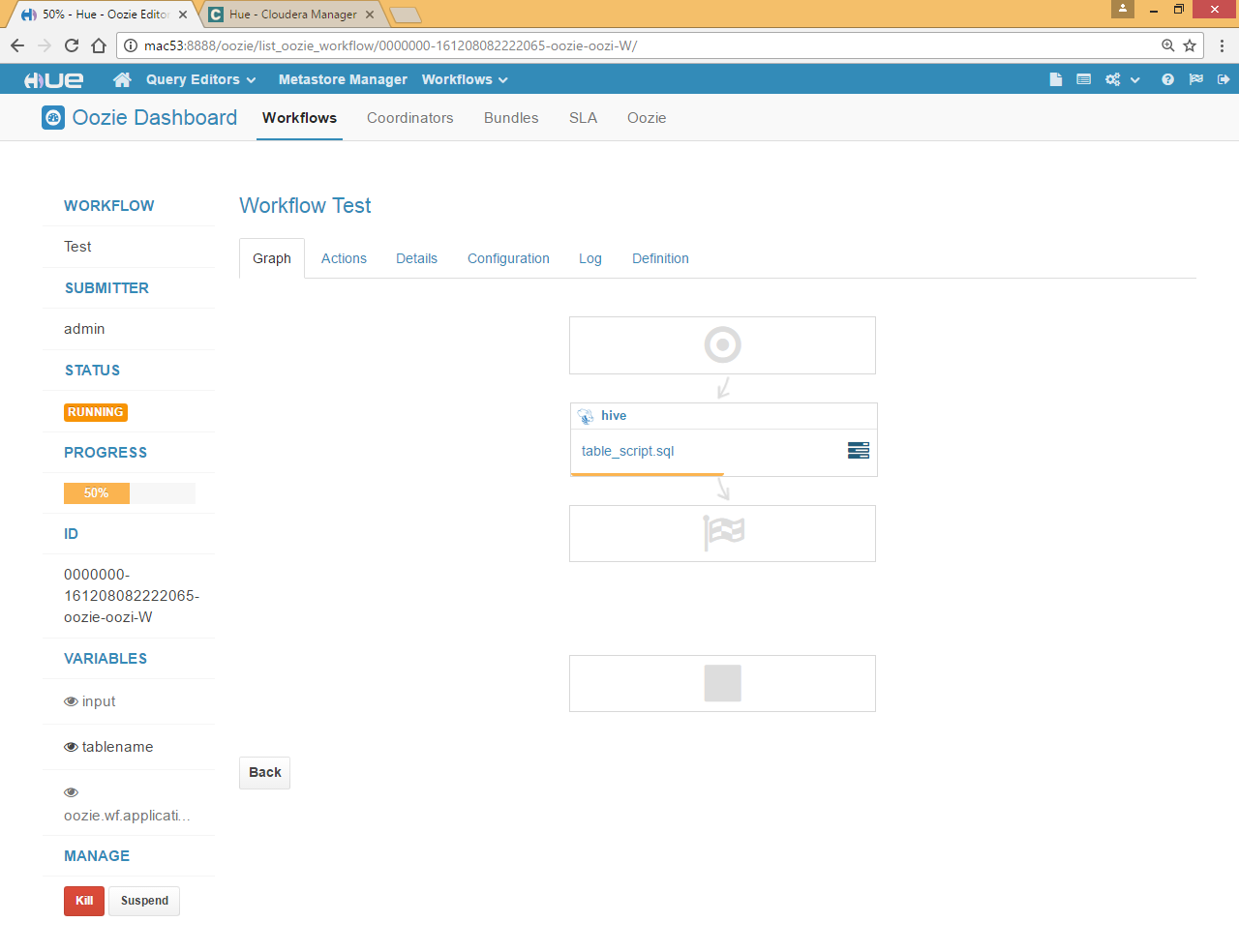




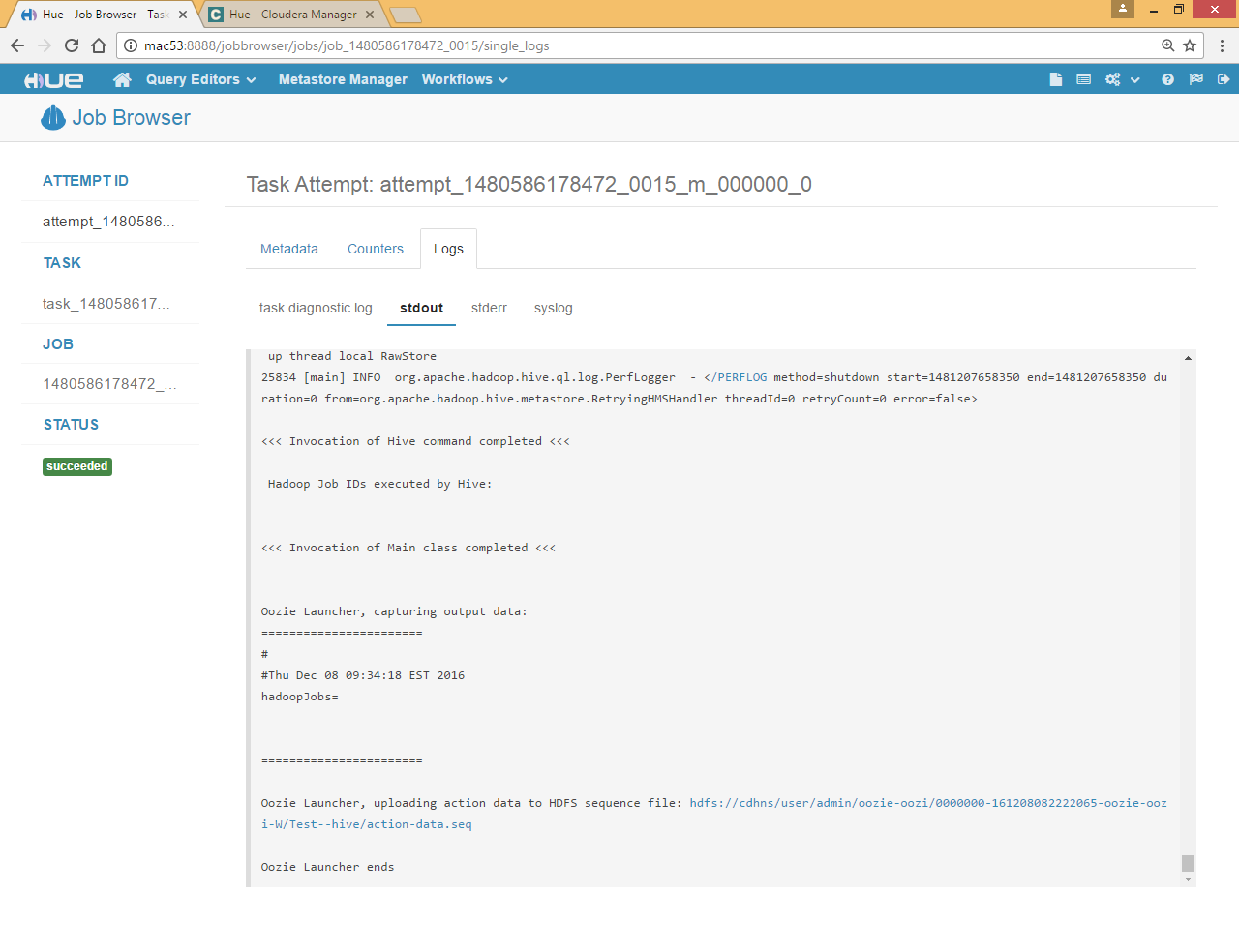
Select and submit

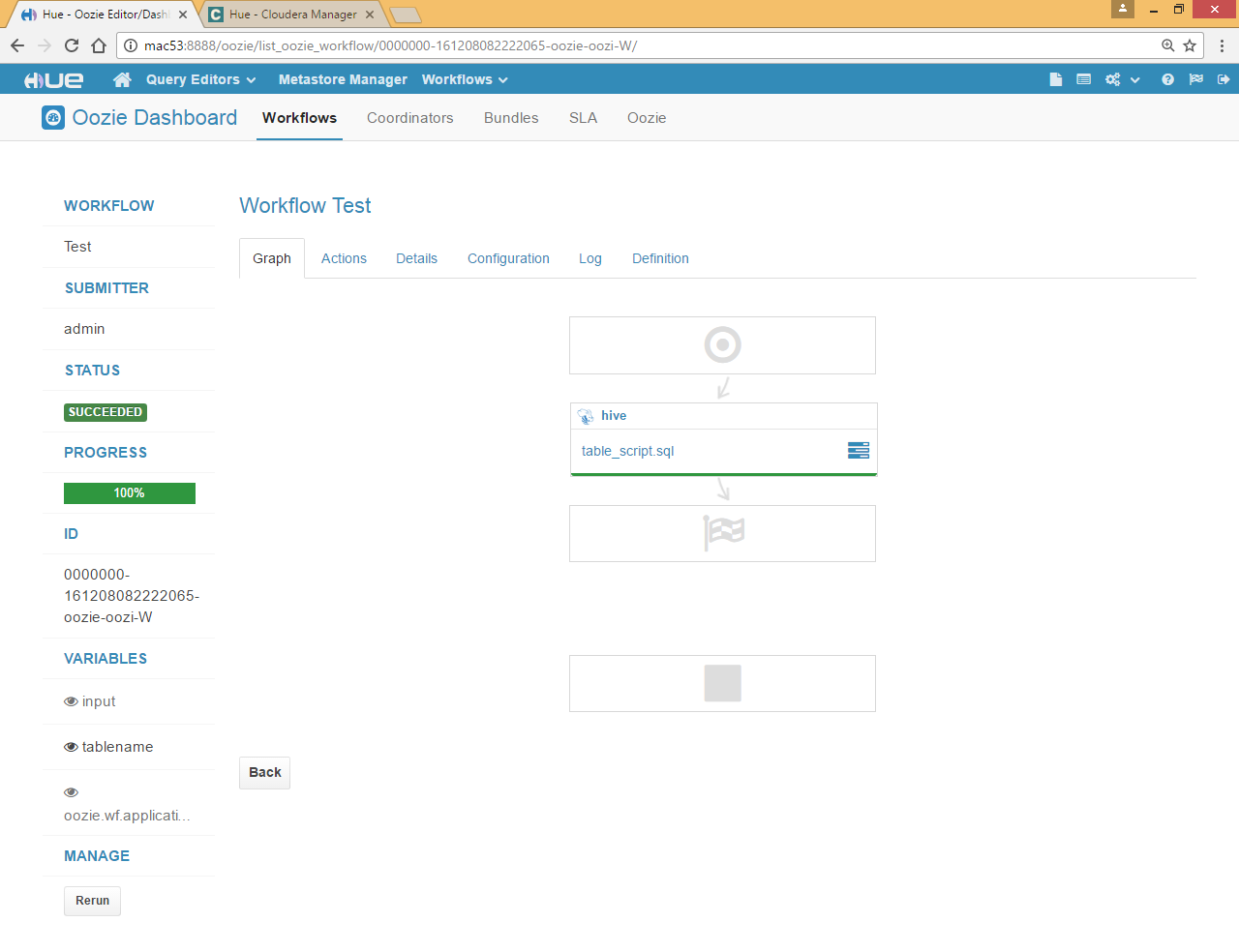




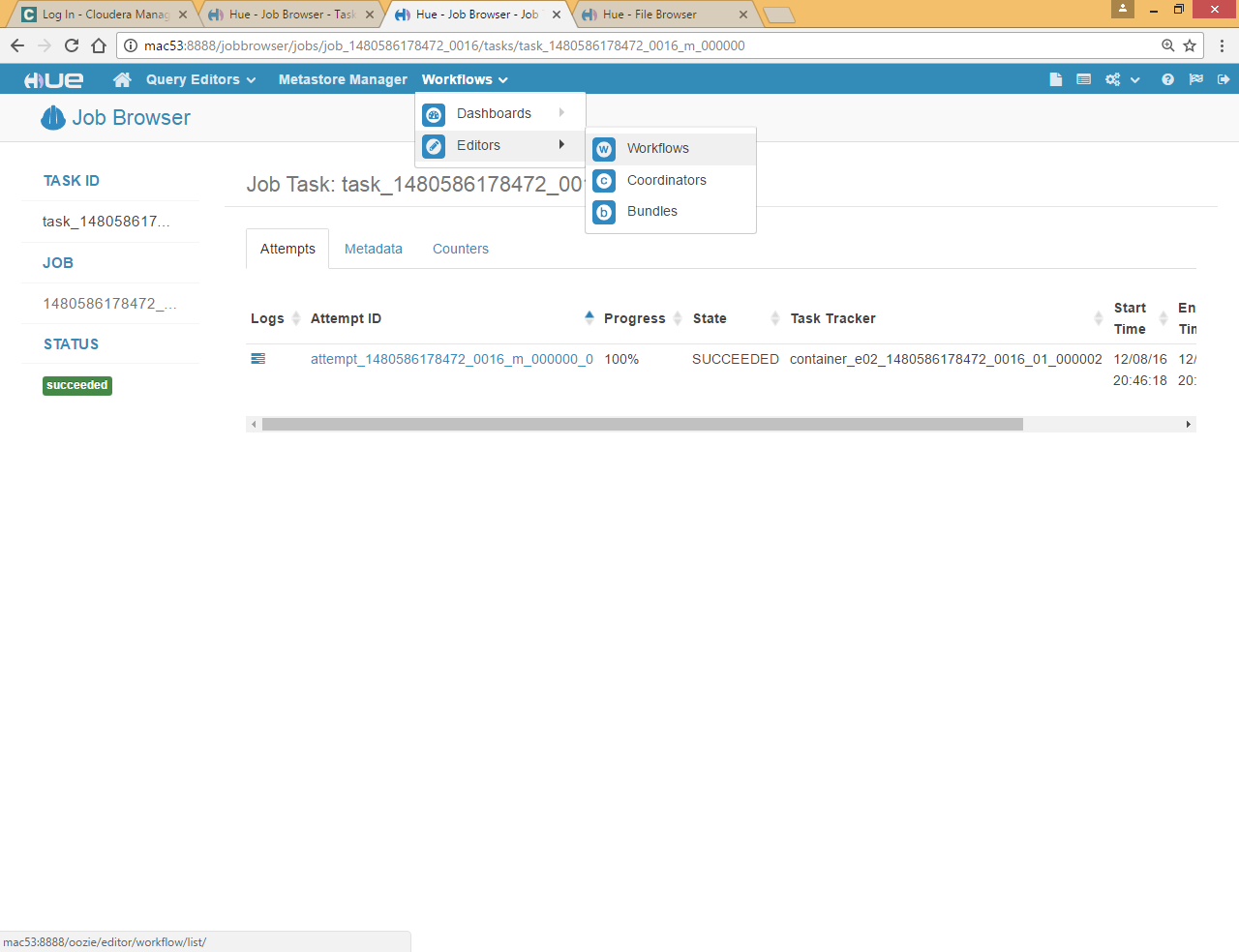


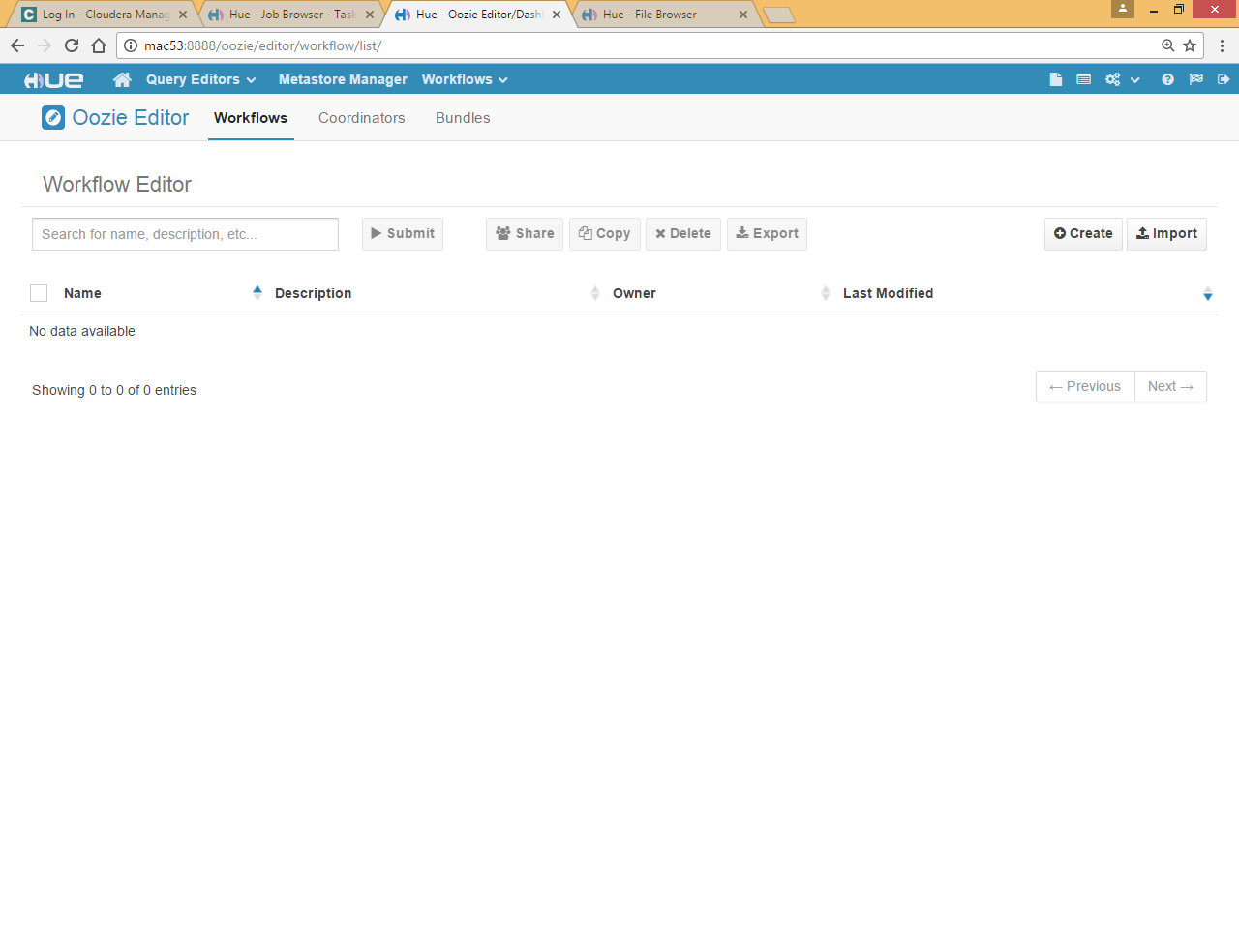
Click on icon log



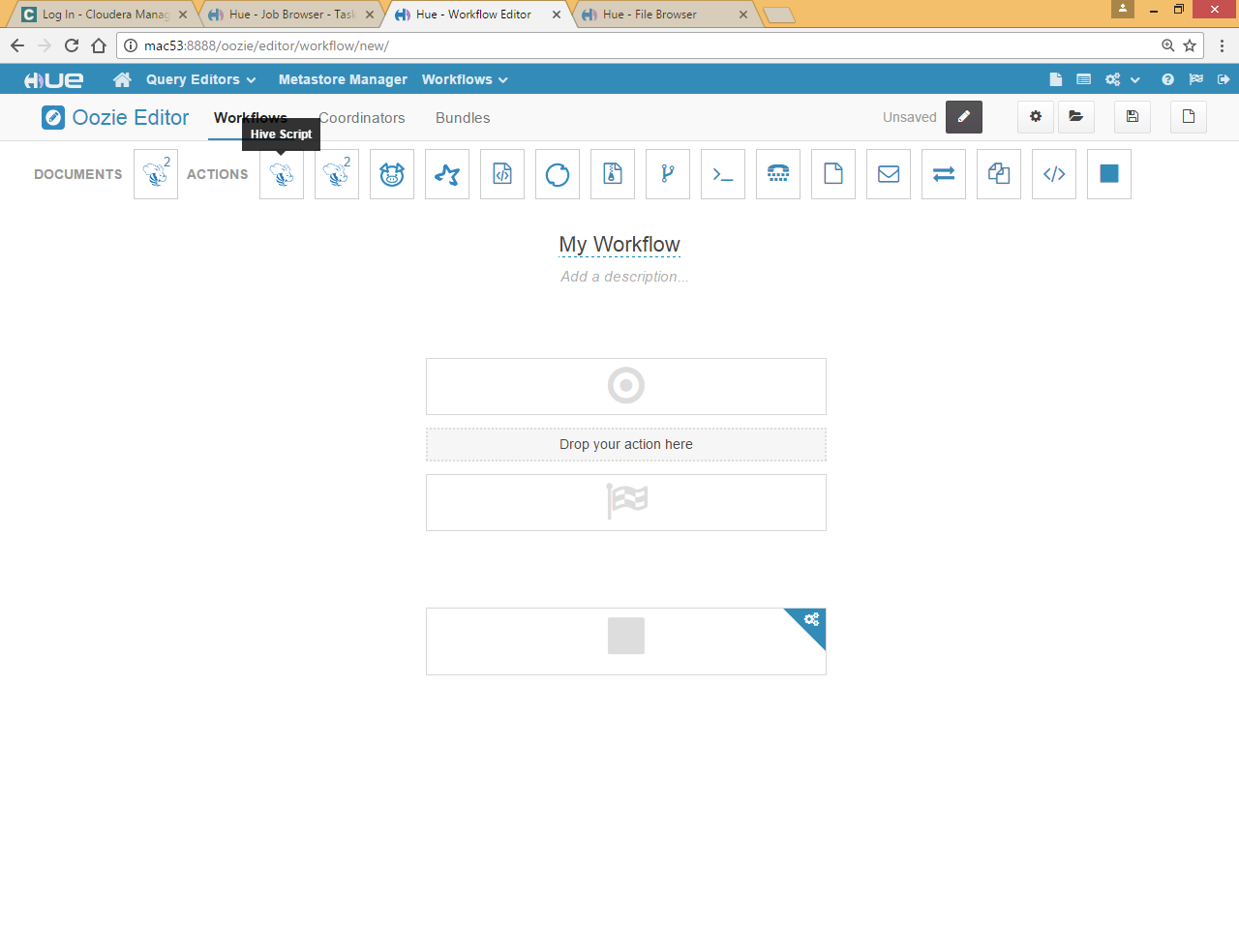


# Oozie workflow by drag and drop

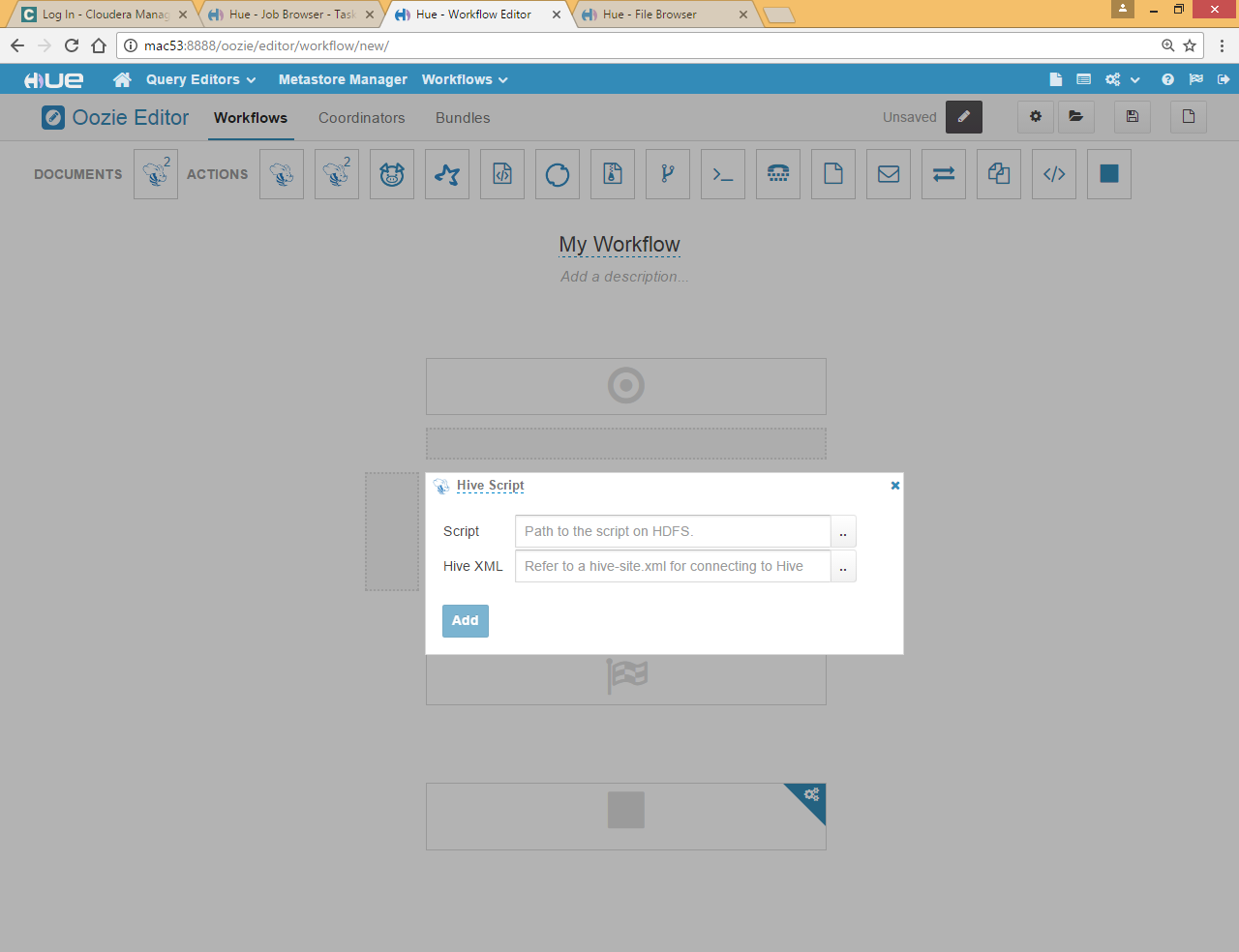




Click Create

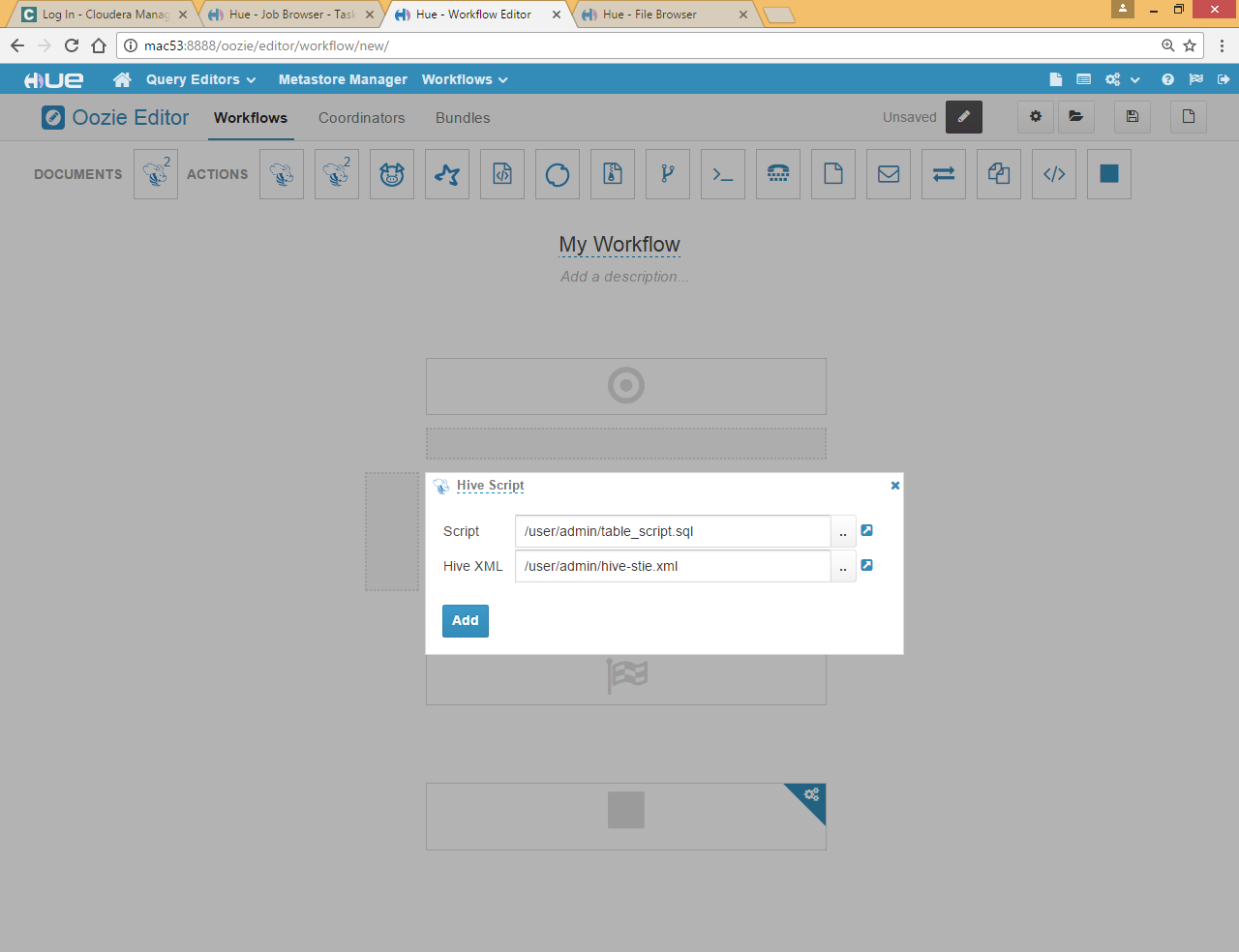


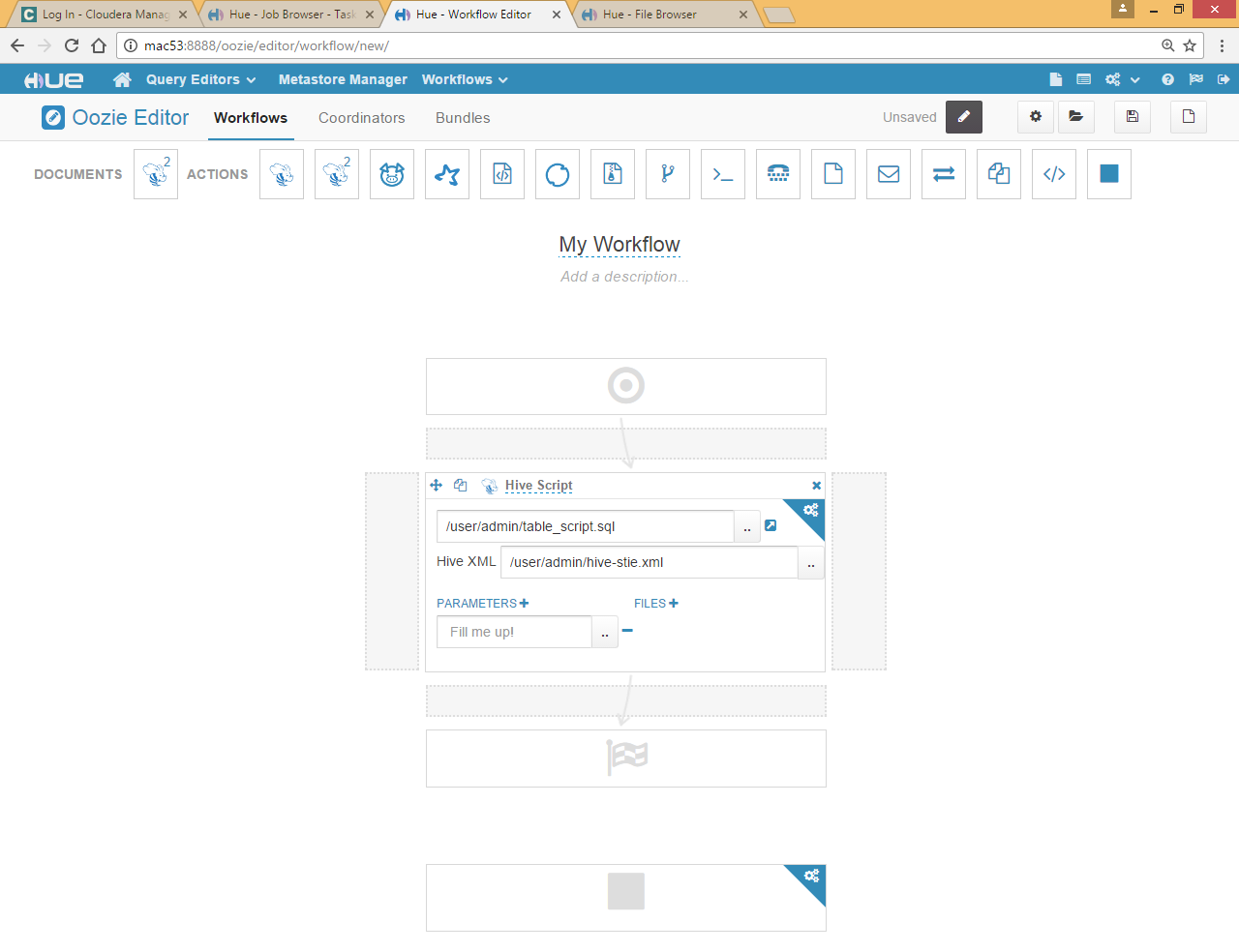
Drag hive script and drop in workflow



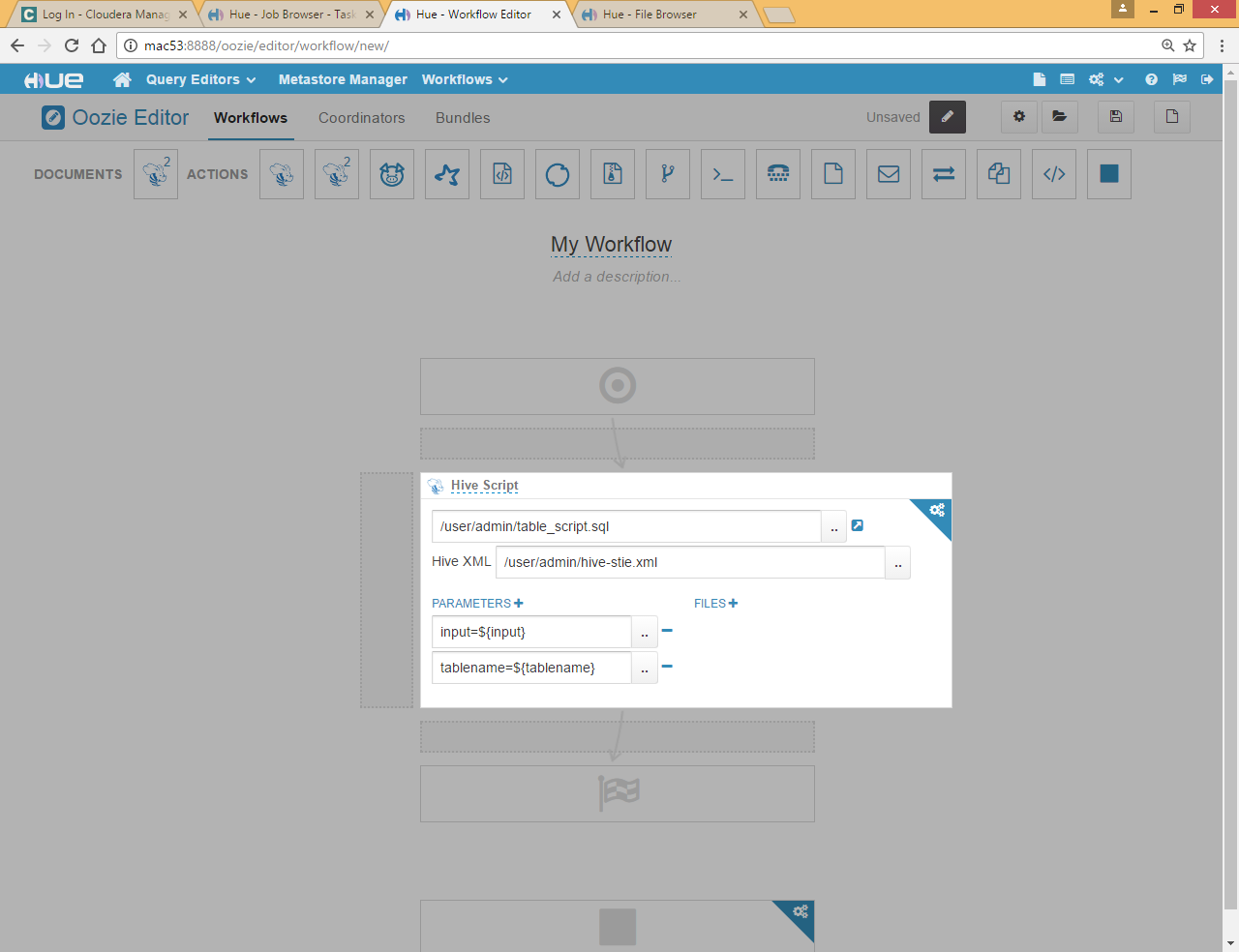
Specify script and hive xml path , you need to upload these files if not present in hdfs



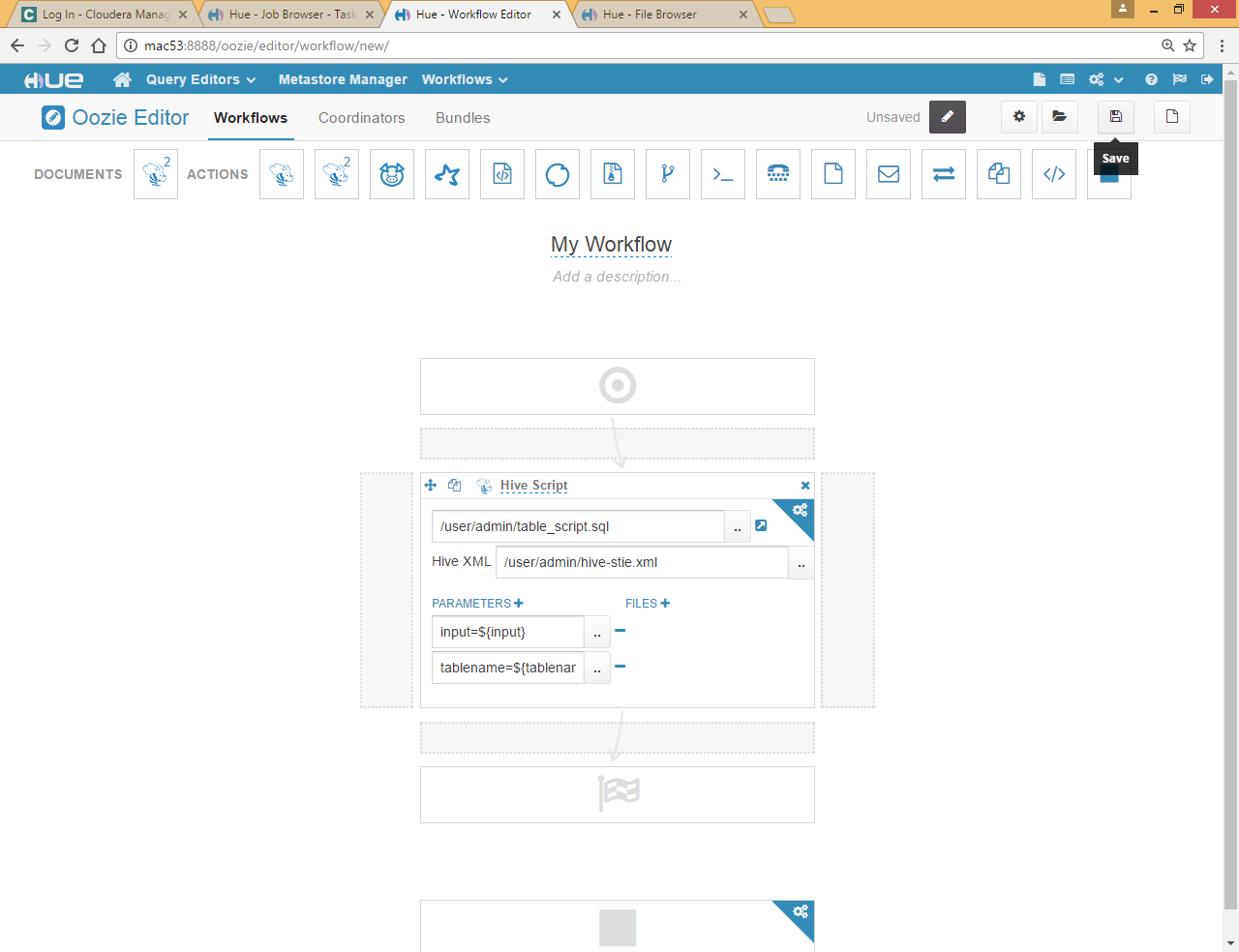




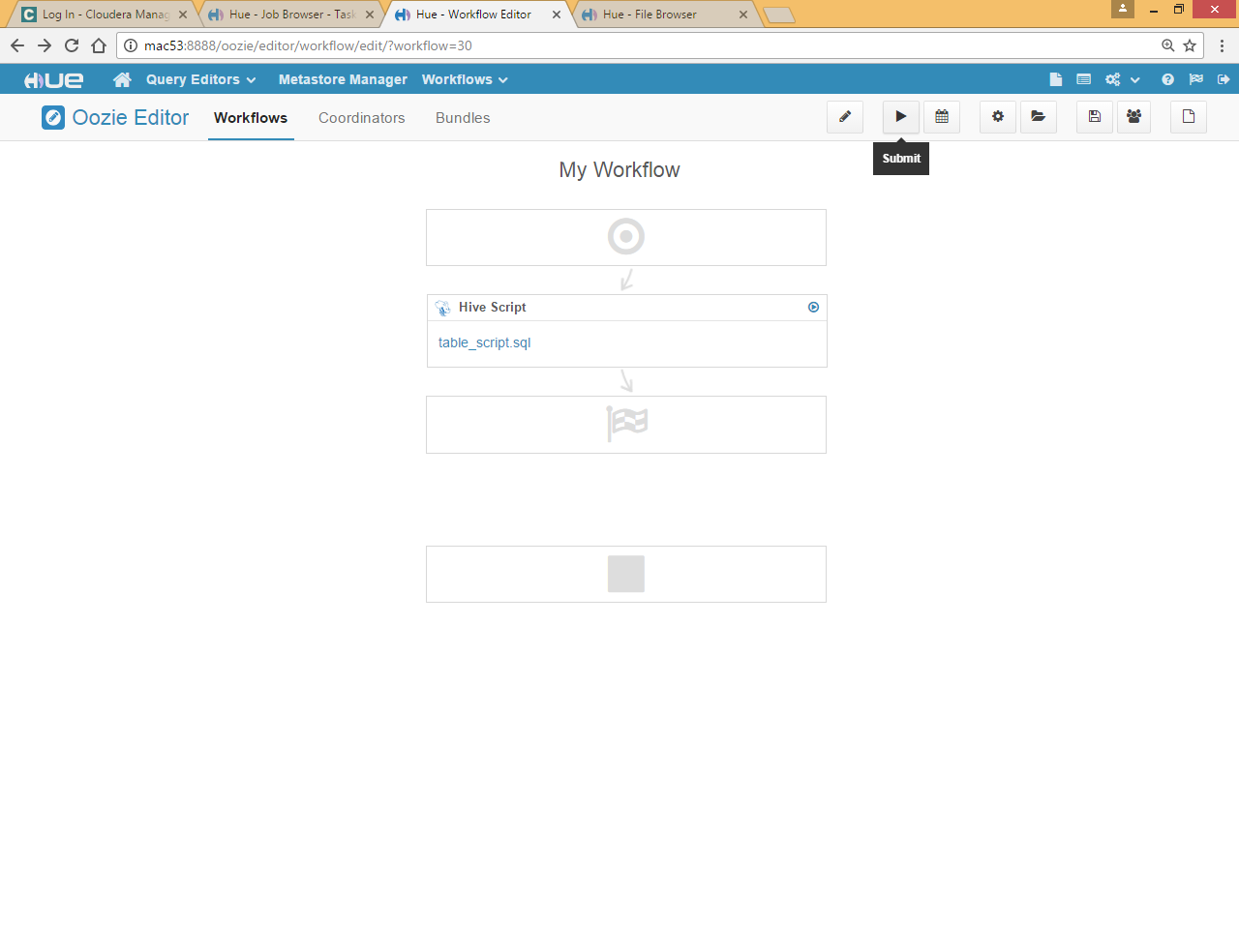
Add parameter



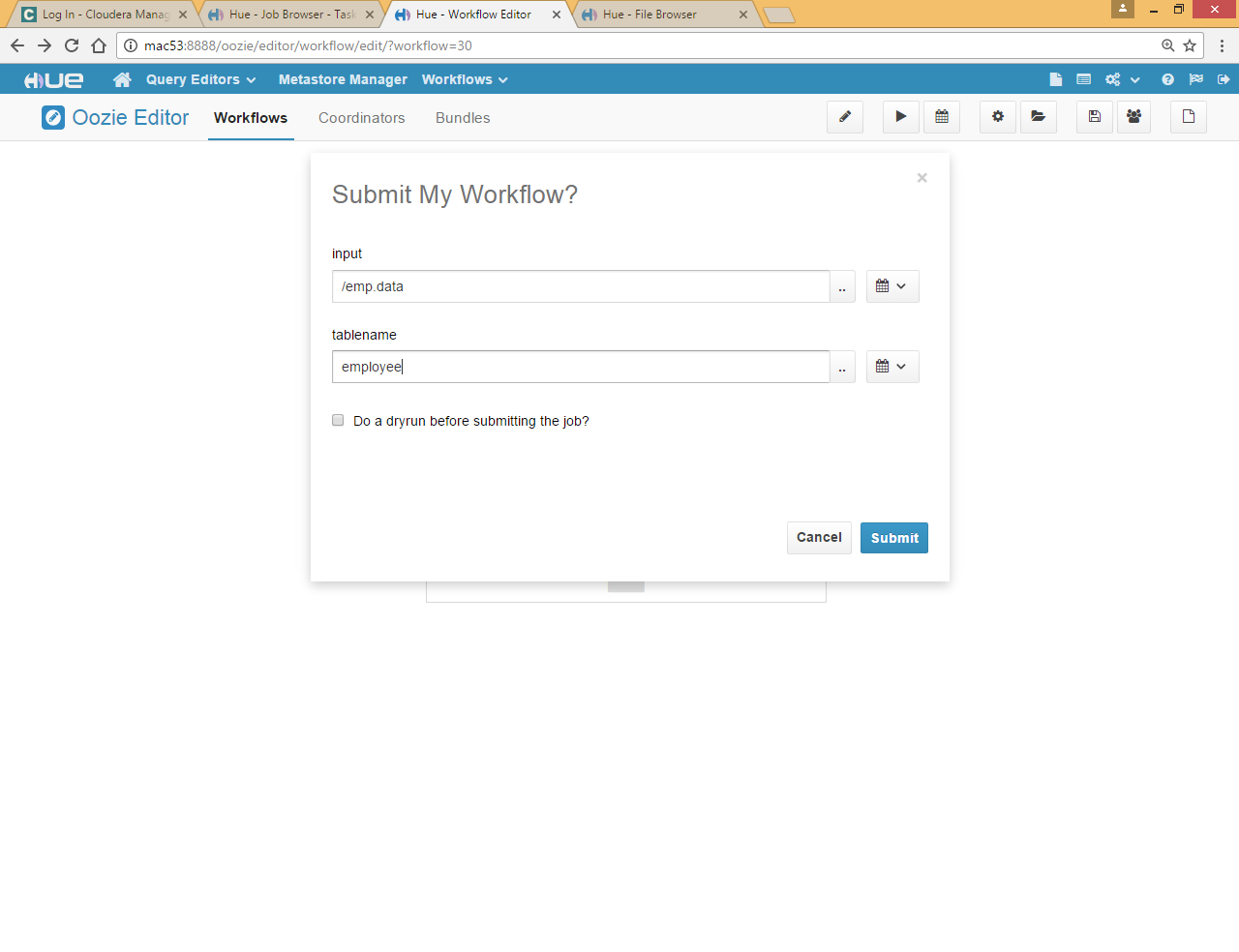
Click on Save

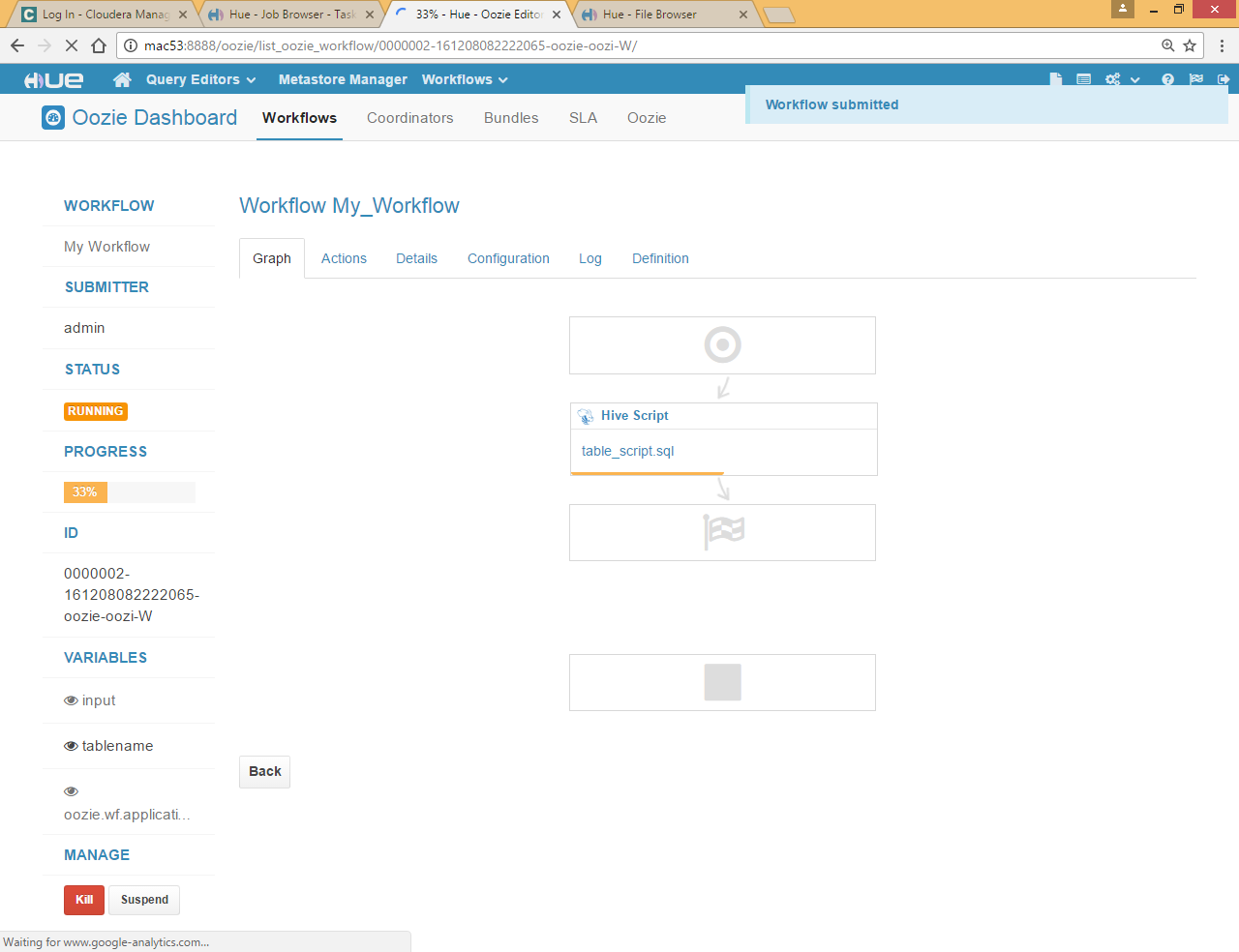


Click on Submit

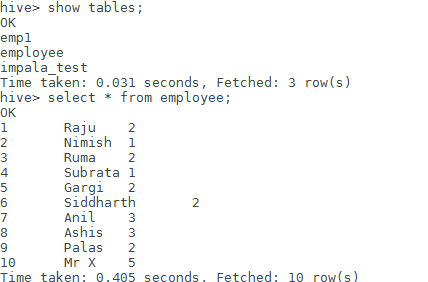


Give input file path and table name

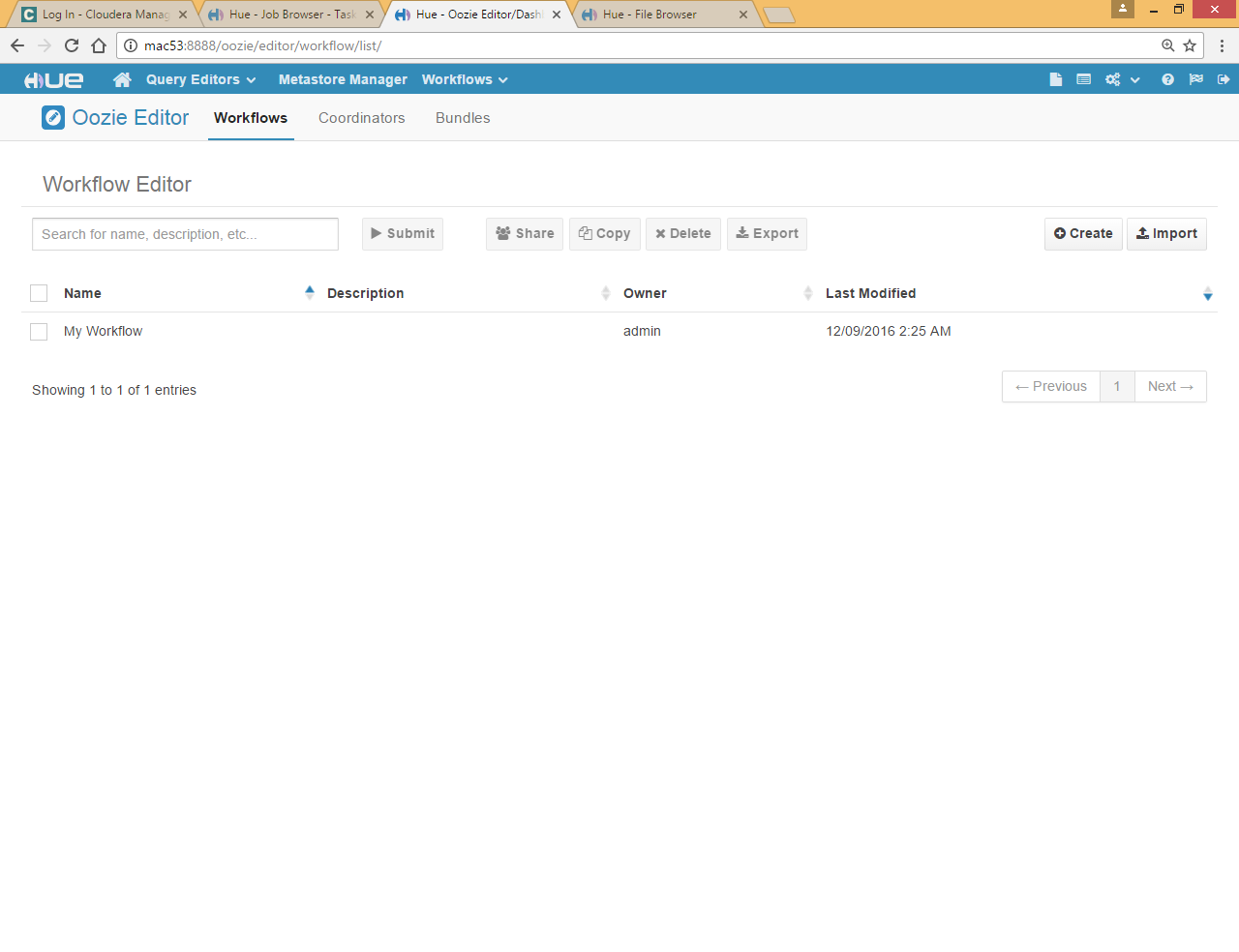




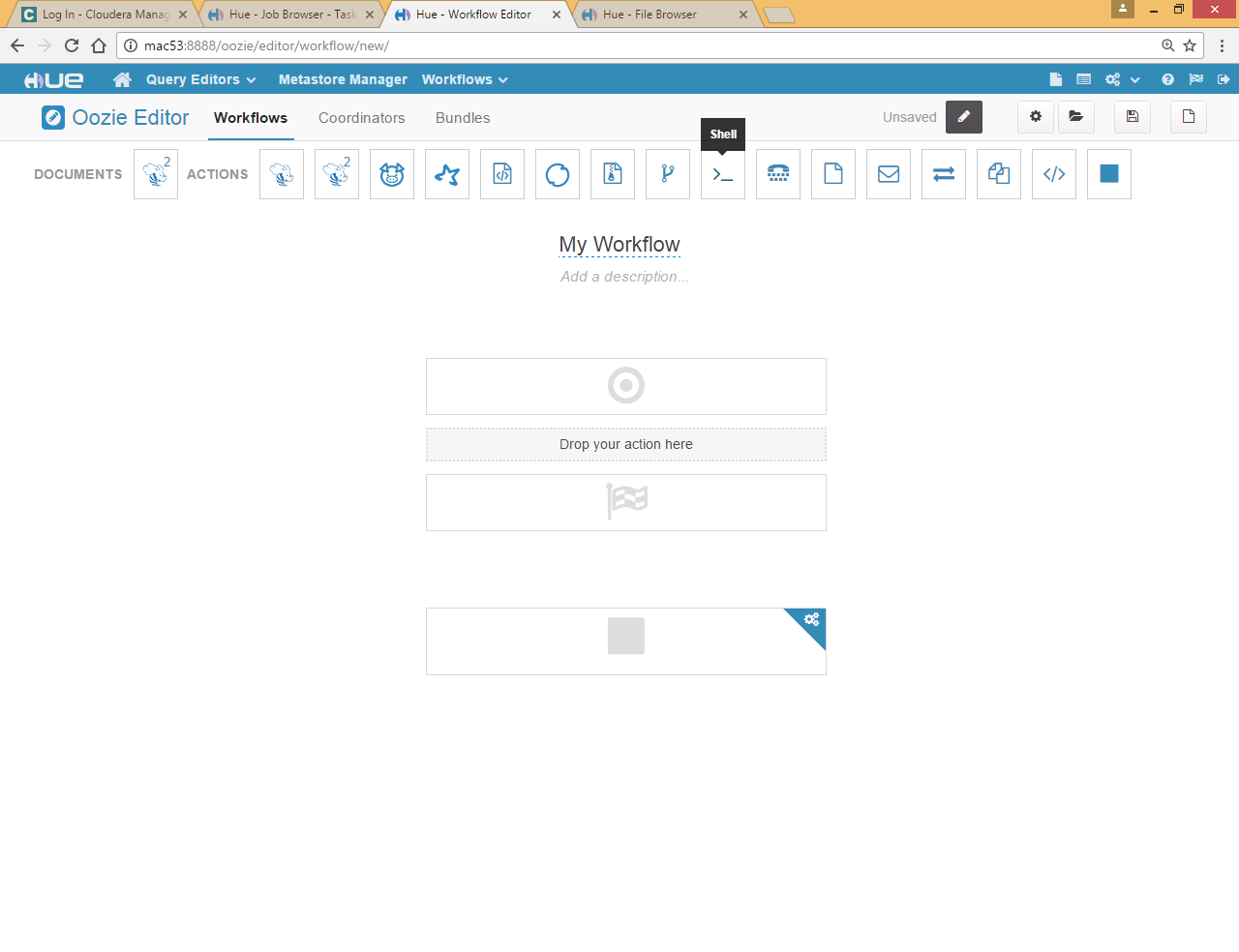
Go to hive shell and check if employee table created successfully



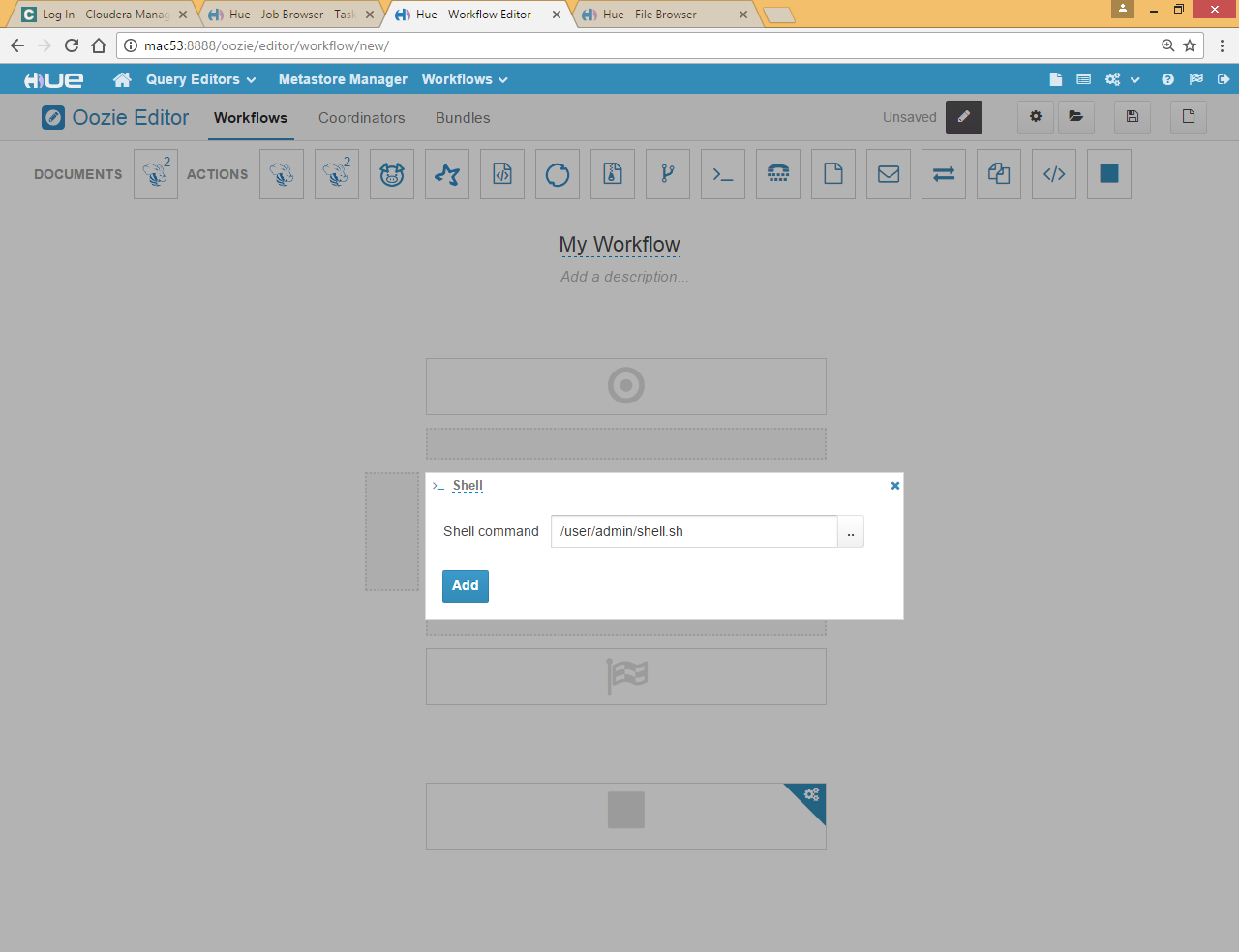
# Oozie workflow for shell script



Create



Drag shell

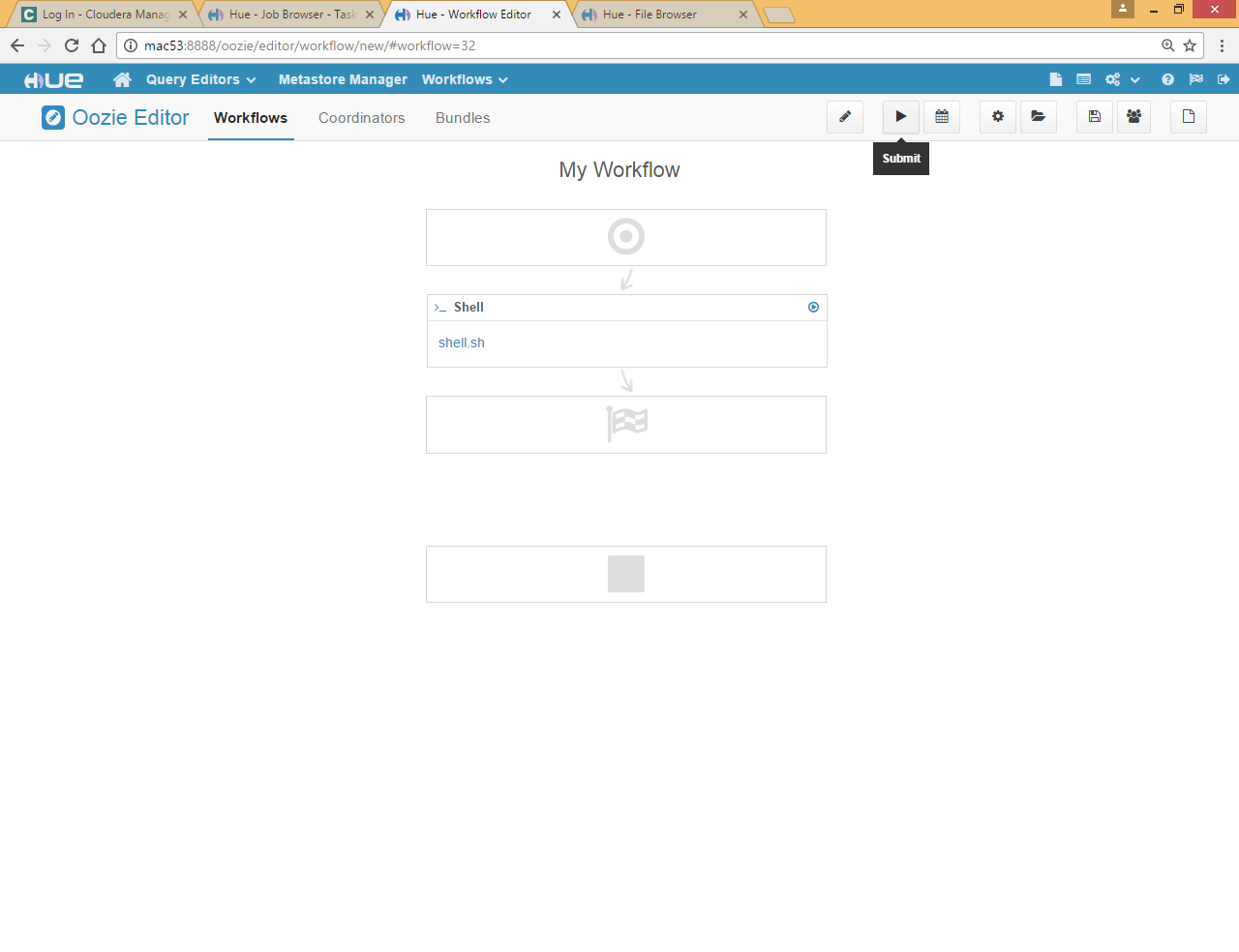


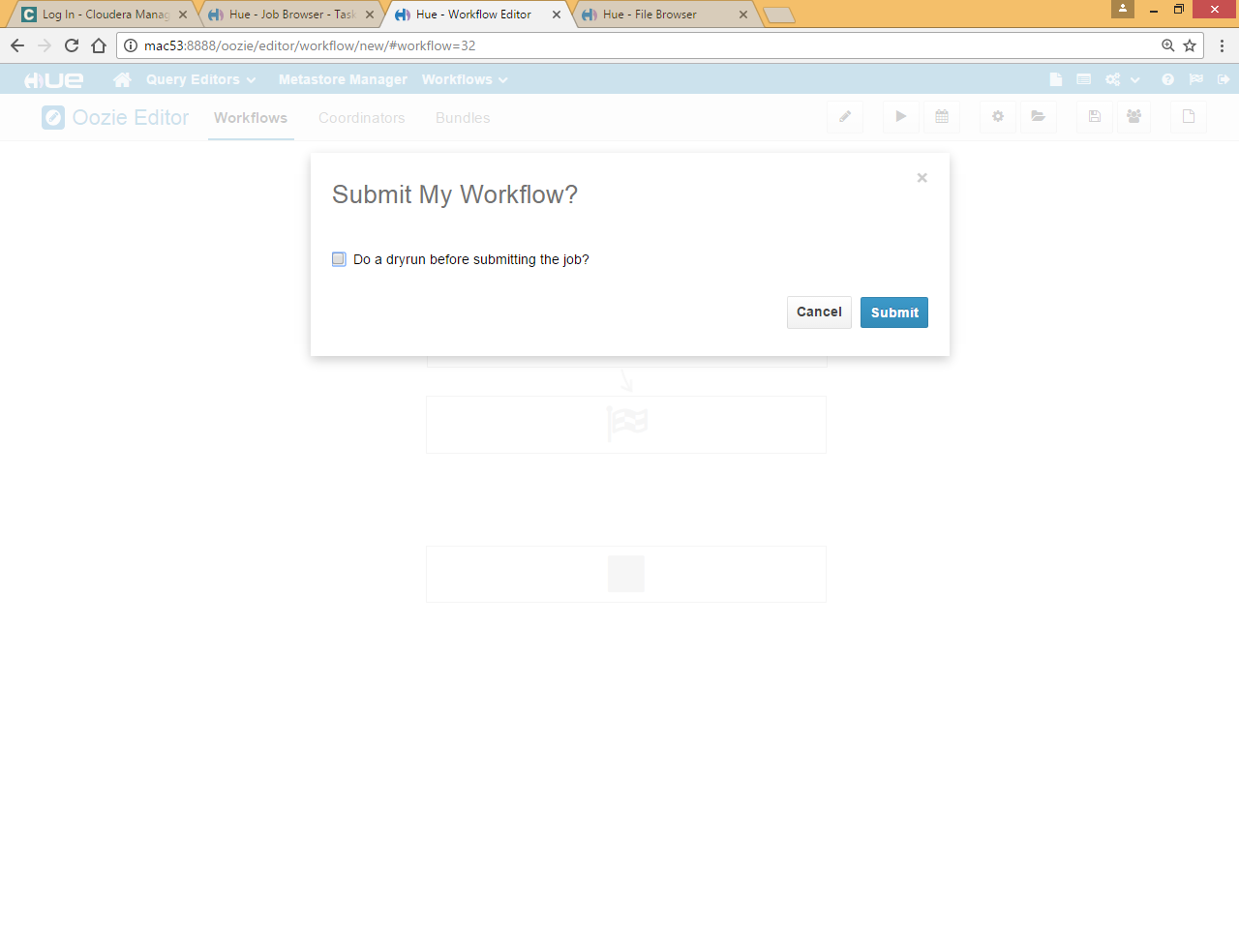
Choose below shell file

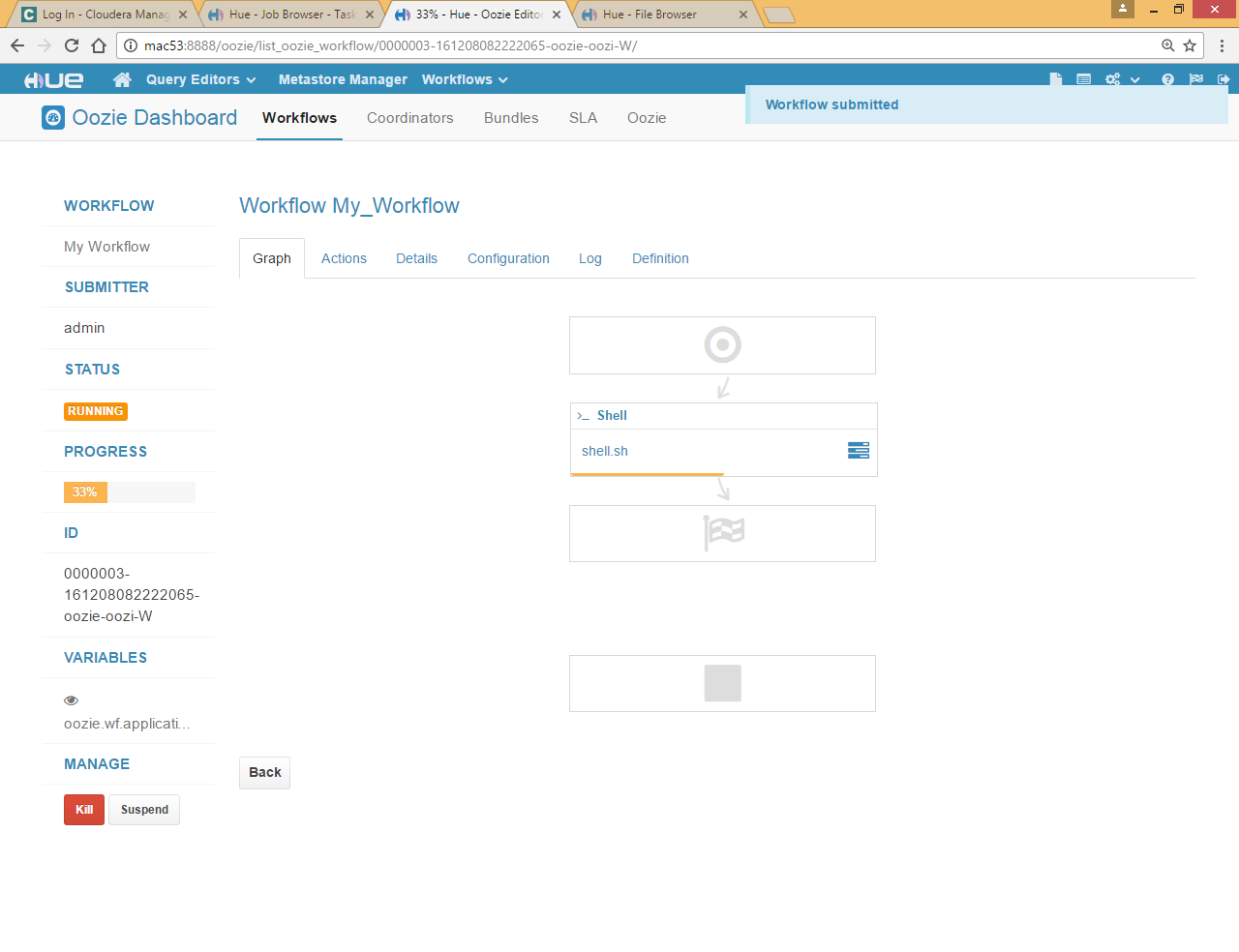


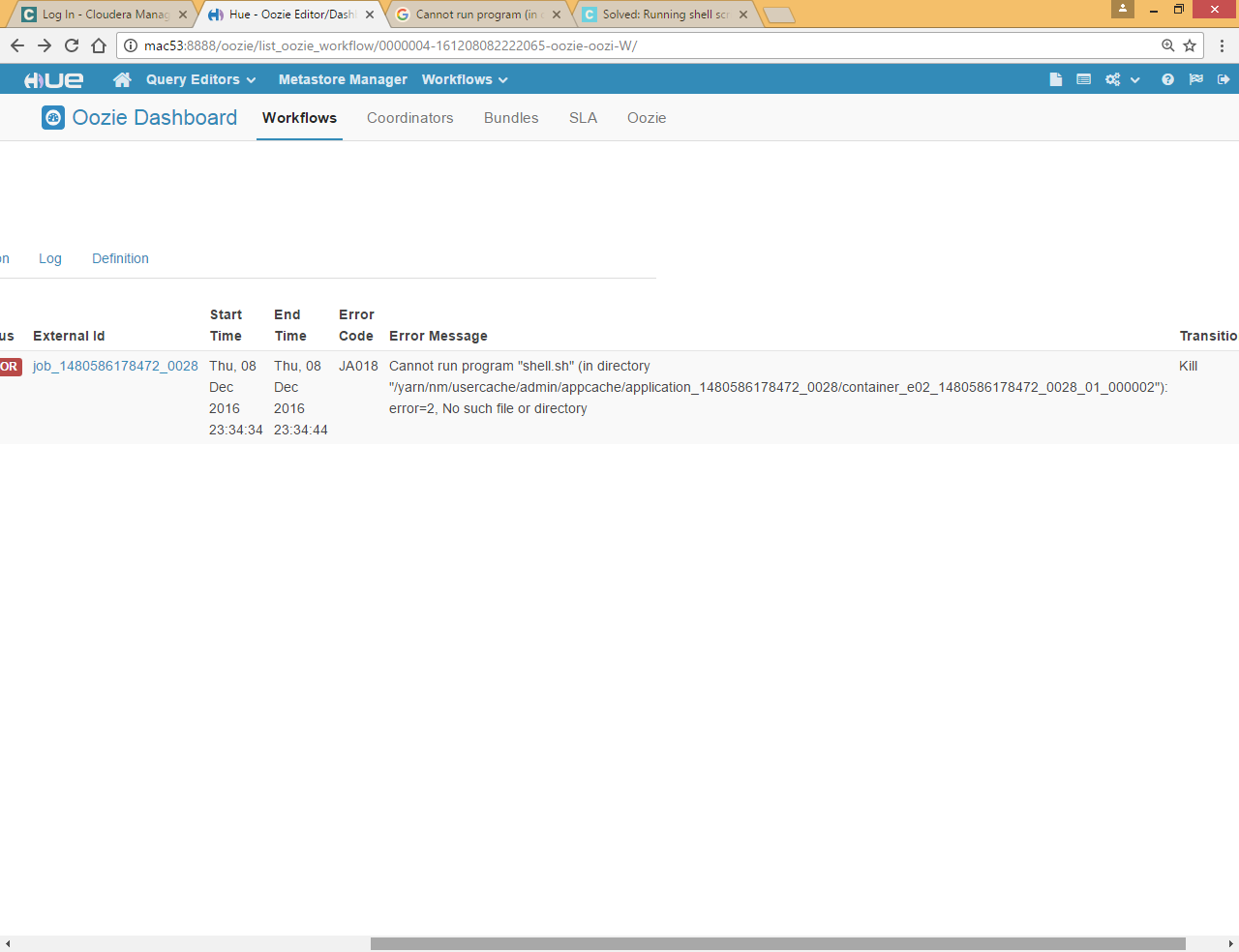


Click save and submit

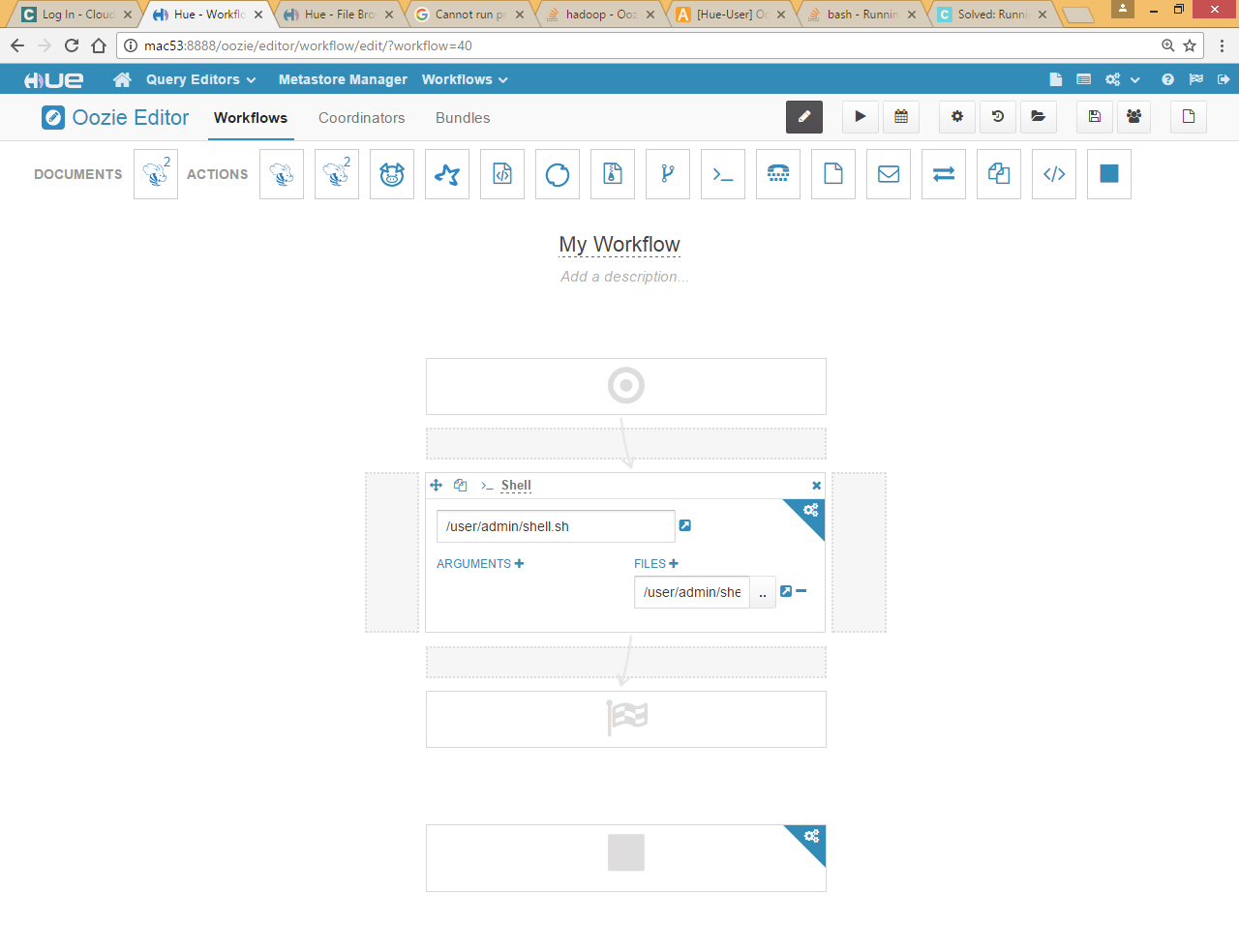




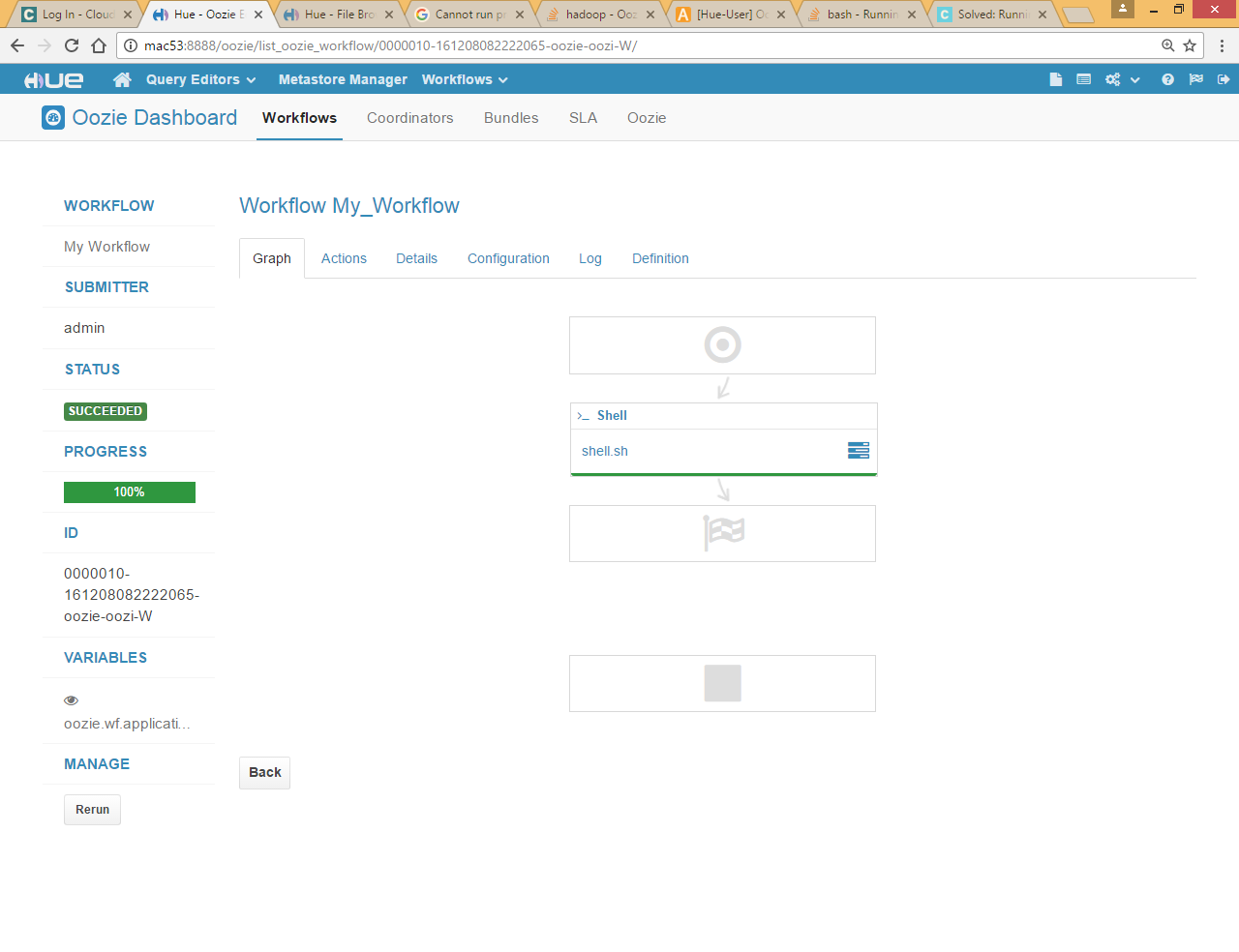




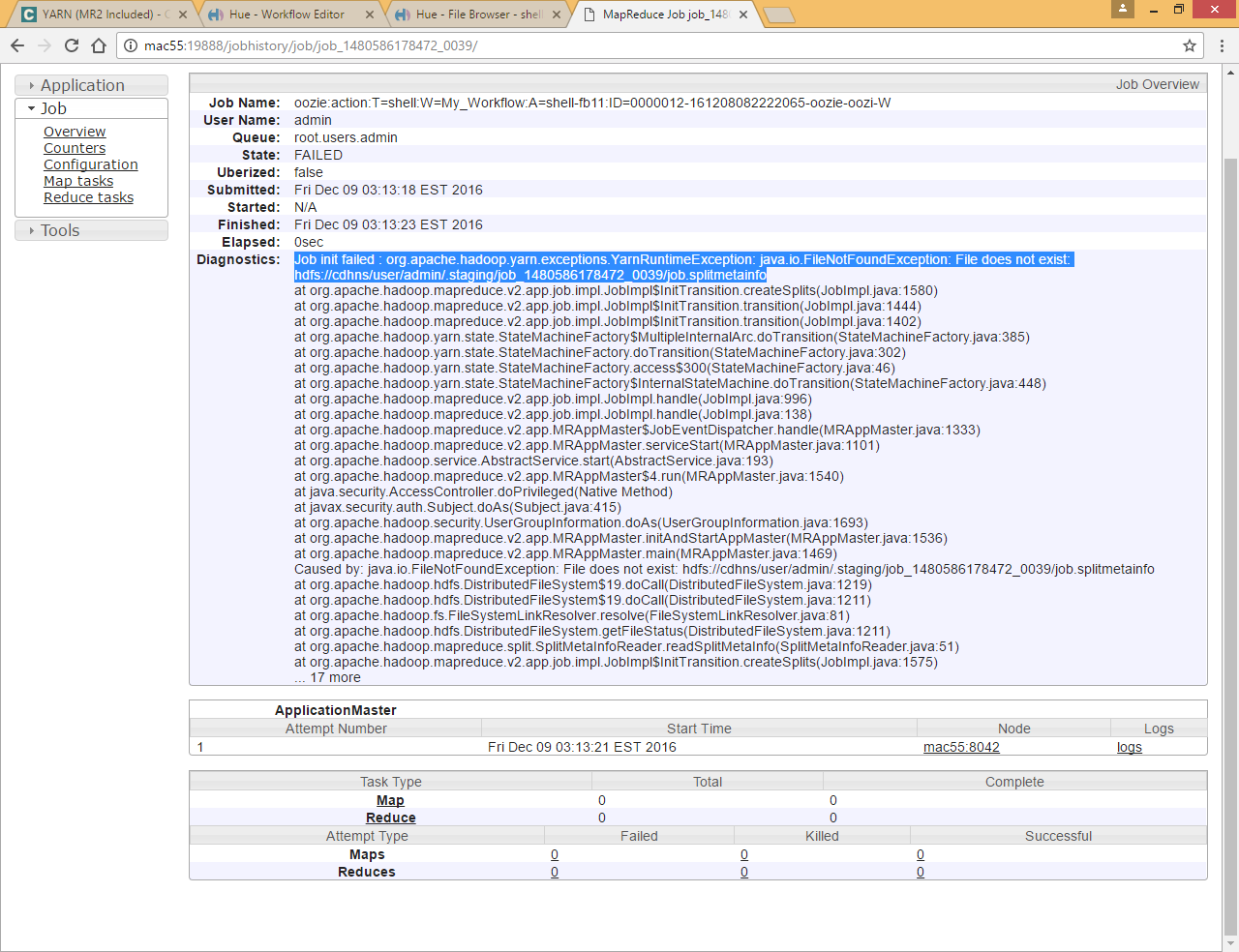
To resolve this issue we need to specify shell script file at two places **\_shell and Files ,** see below screen shot



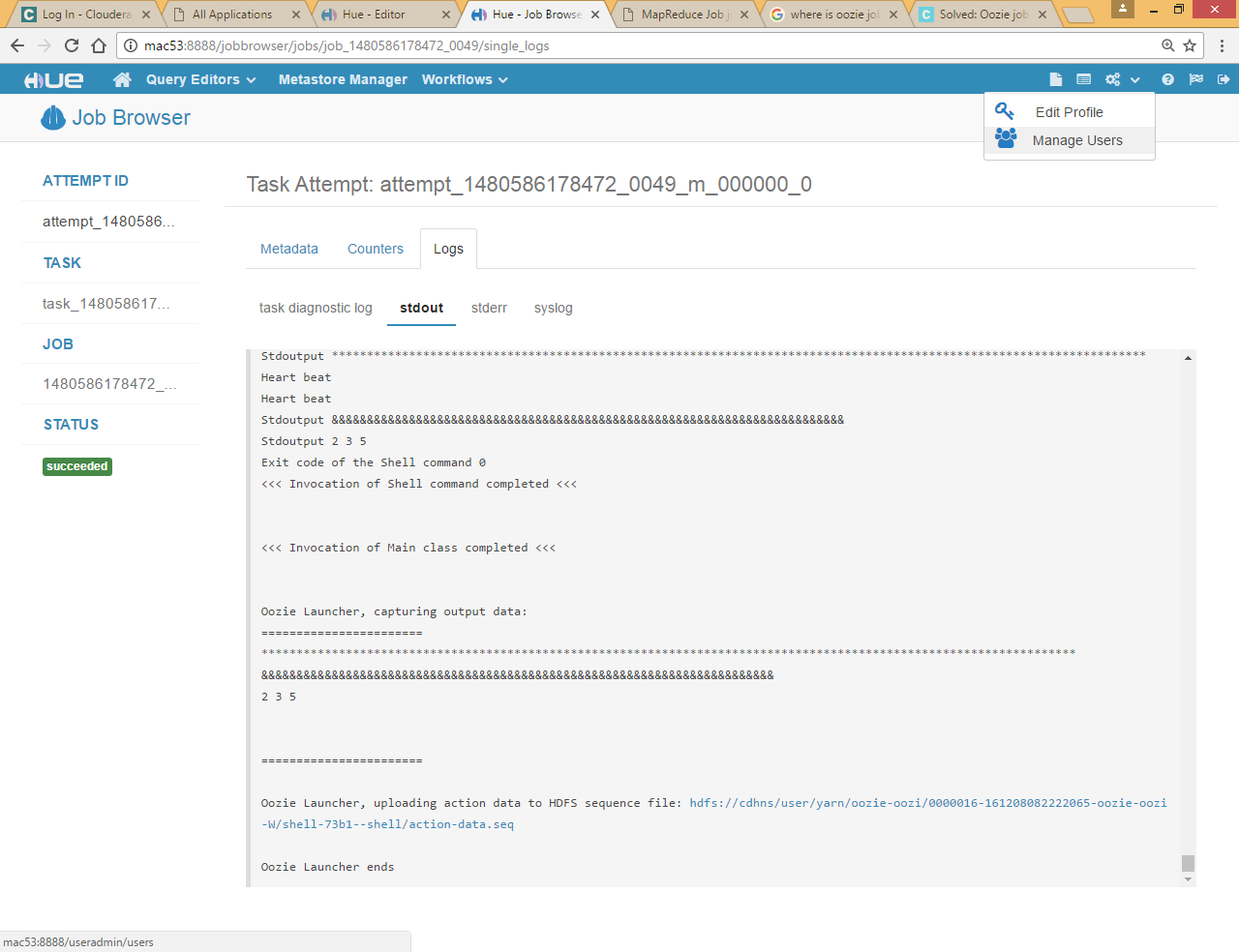
Now save it and submit it



Though script ran with exit code 0 but map red job fail

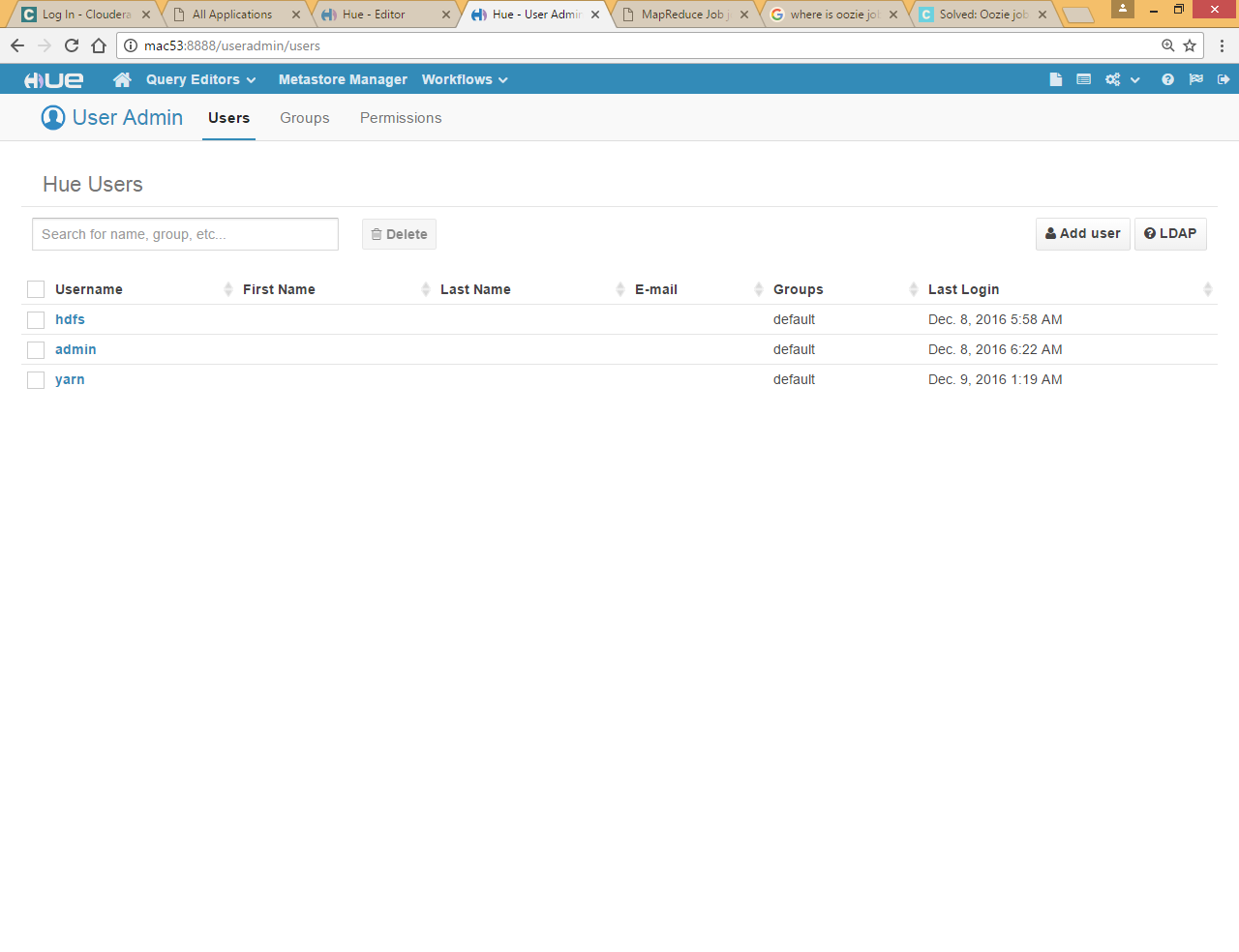


To solve this issue we need to create a yarn user in HUE



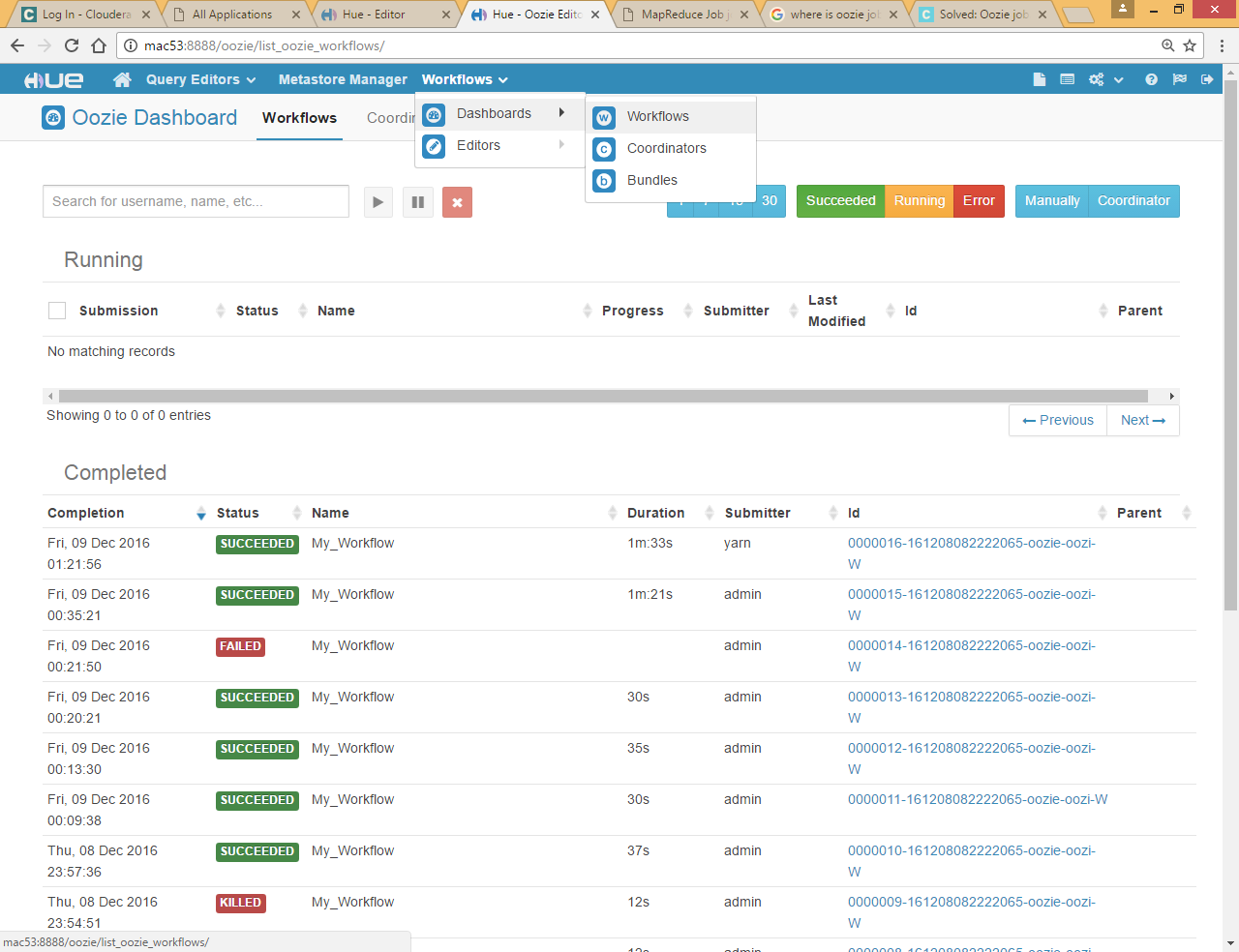
Click on manage users

And create yarn user

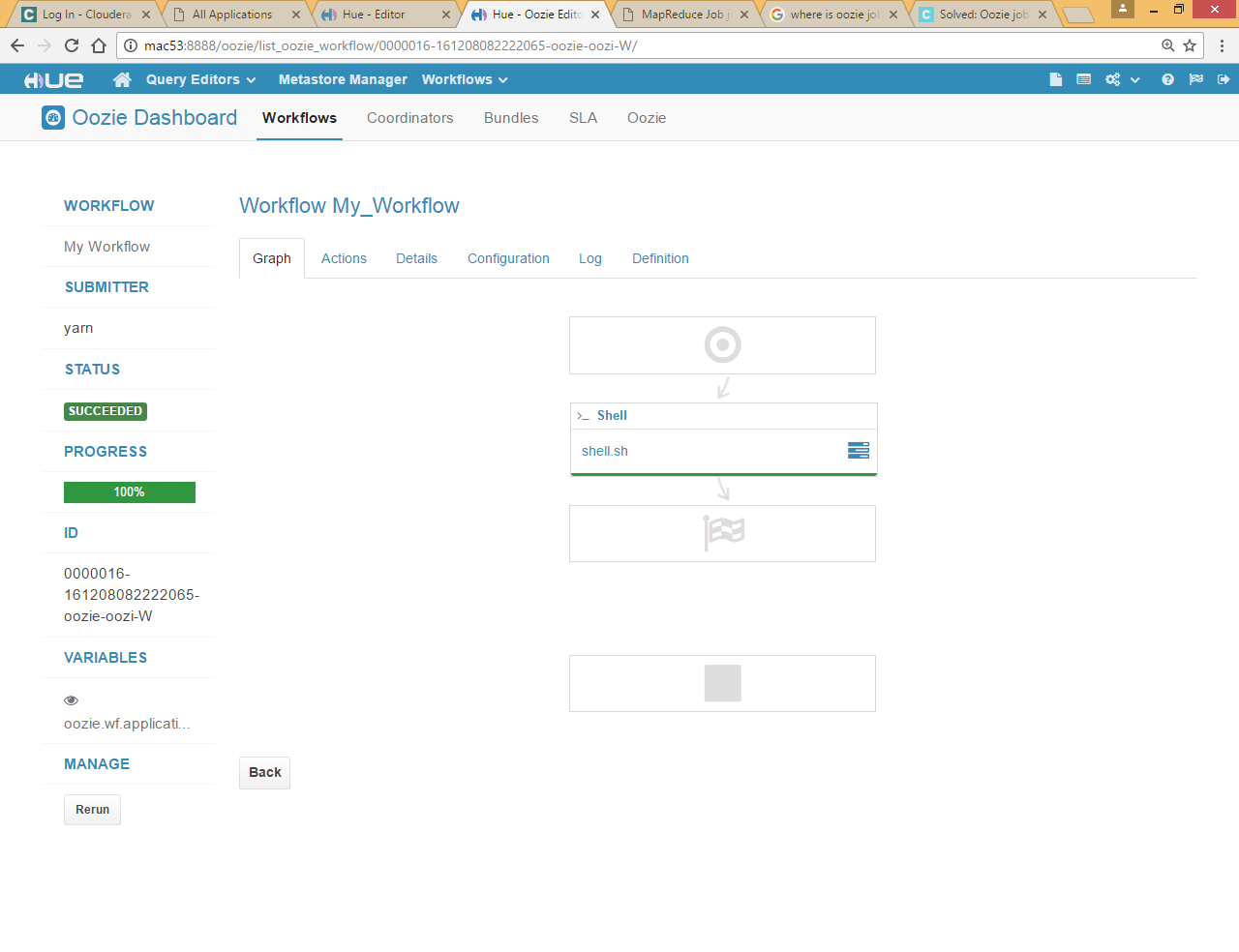


Now login with yarn user and run the workflow again it will successfully executed

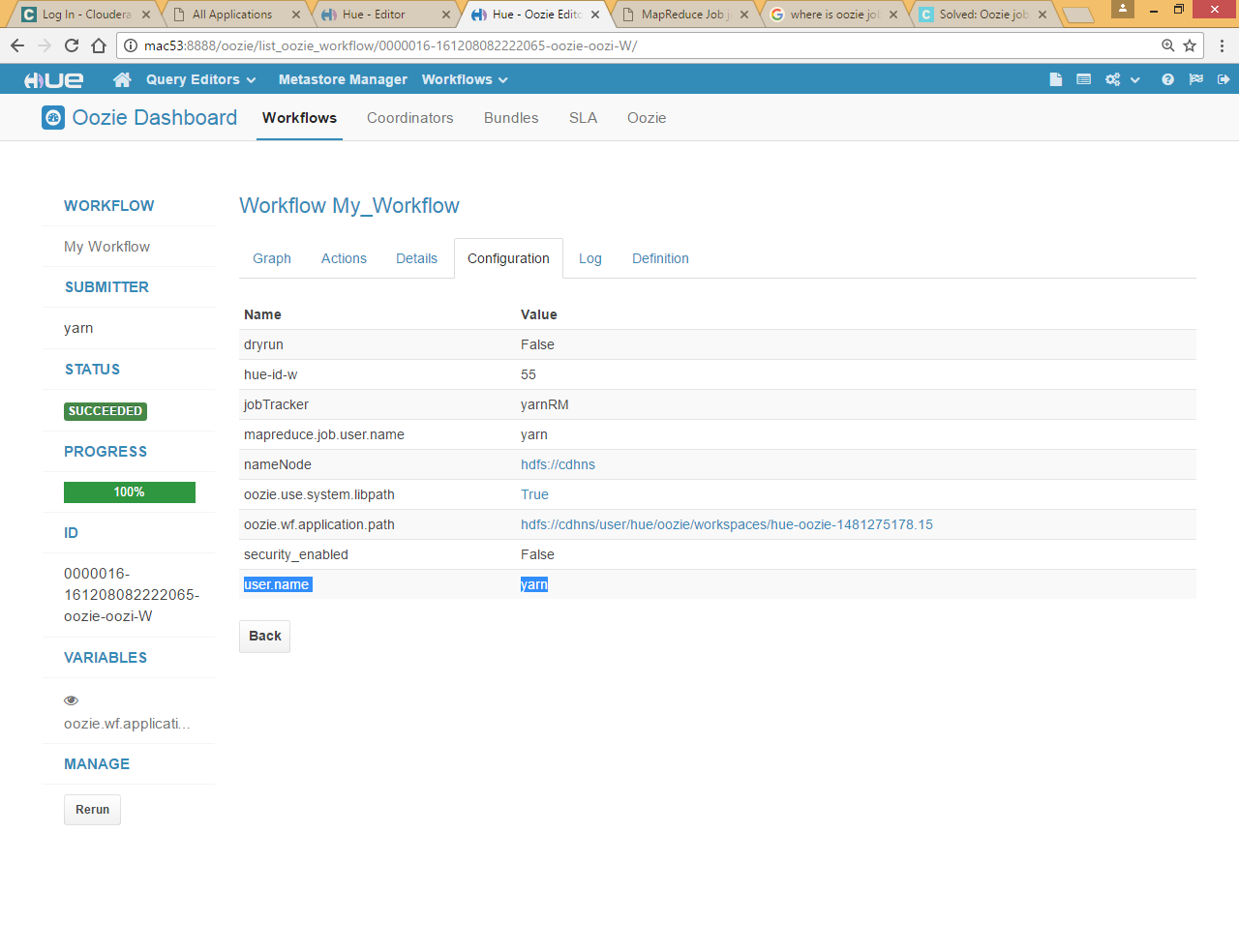
To check the user name click on workflow>>Dashboard>>Workflow



Now click on any link



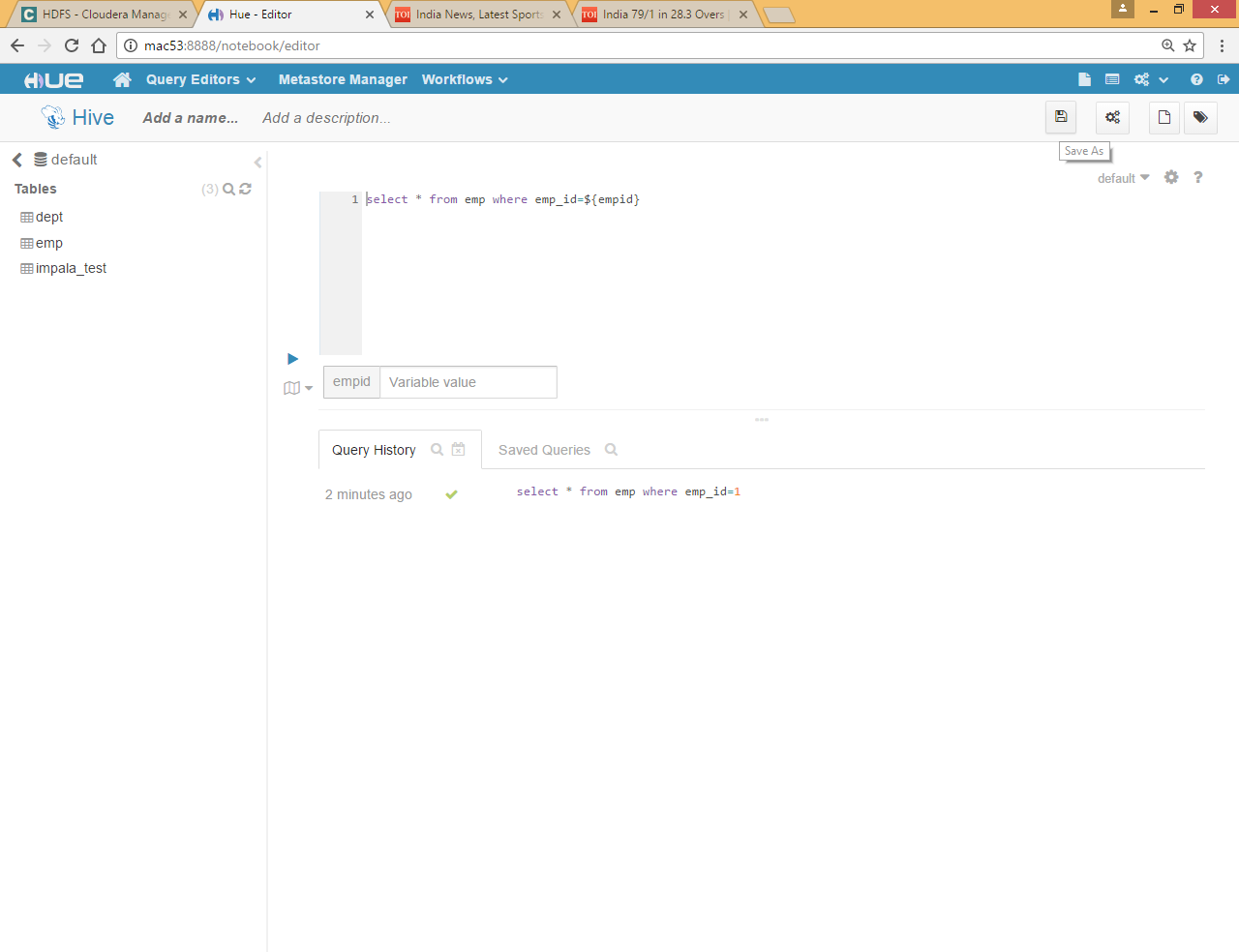
click on Configuration tab to see the configuration



Here you can use that user.name is yarn earlier it was admin as we were login via admin user

# Oozie workflow for hive saved queries

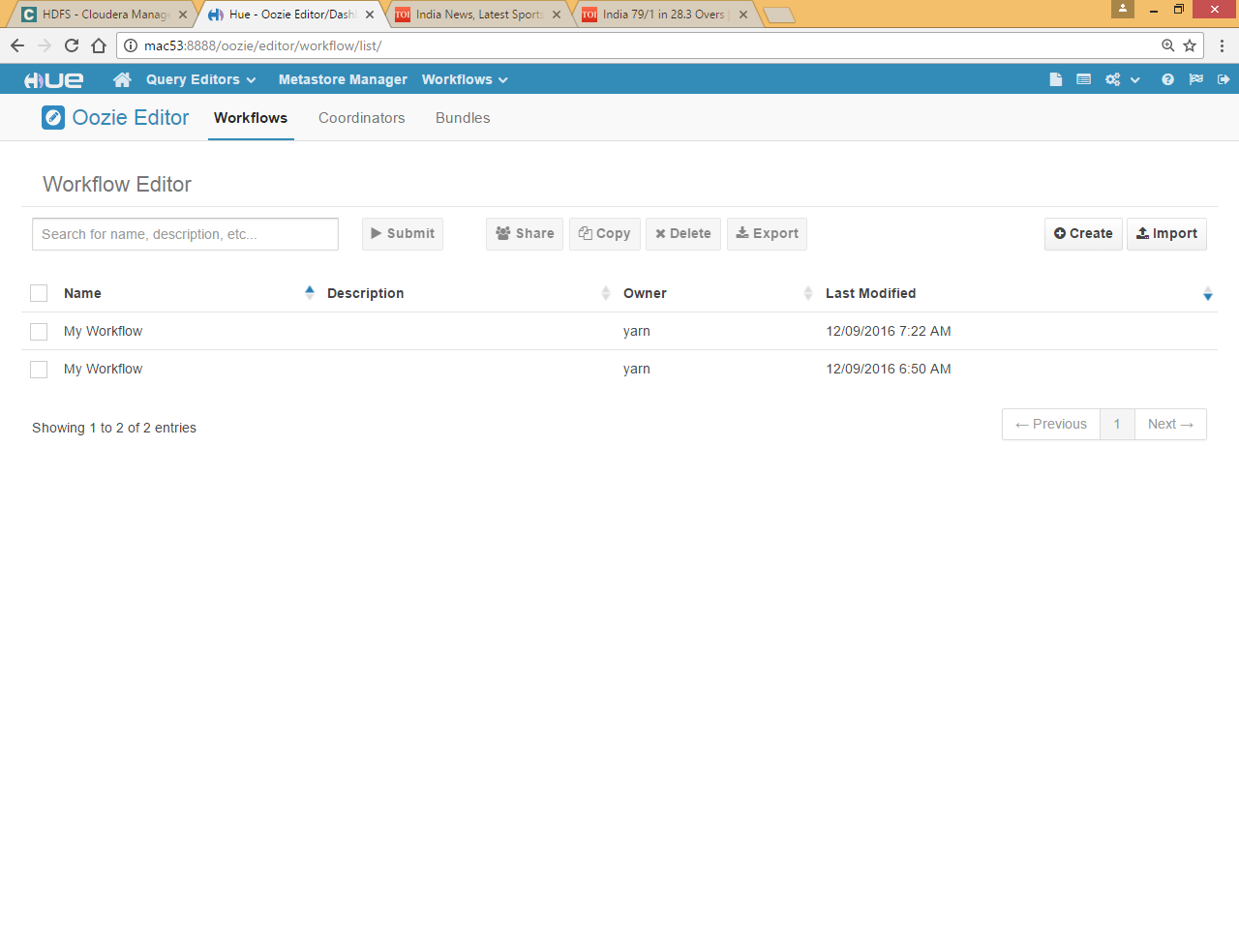
Go to hive editor and save a query



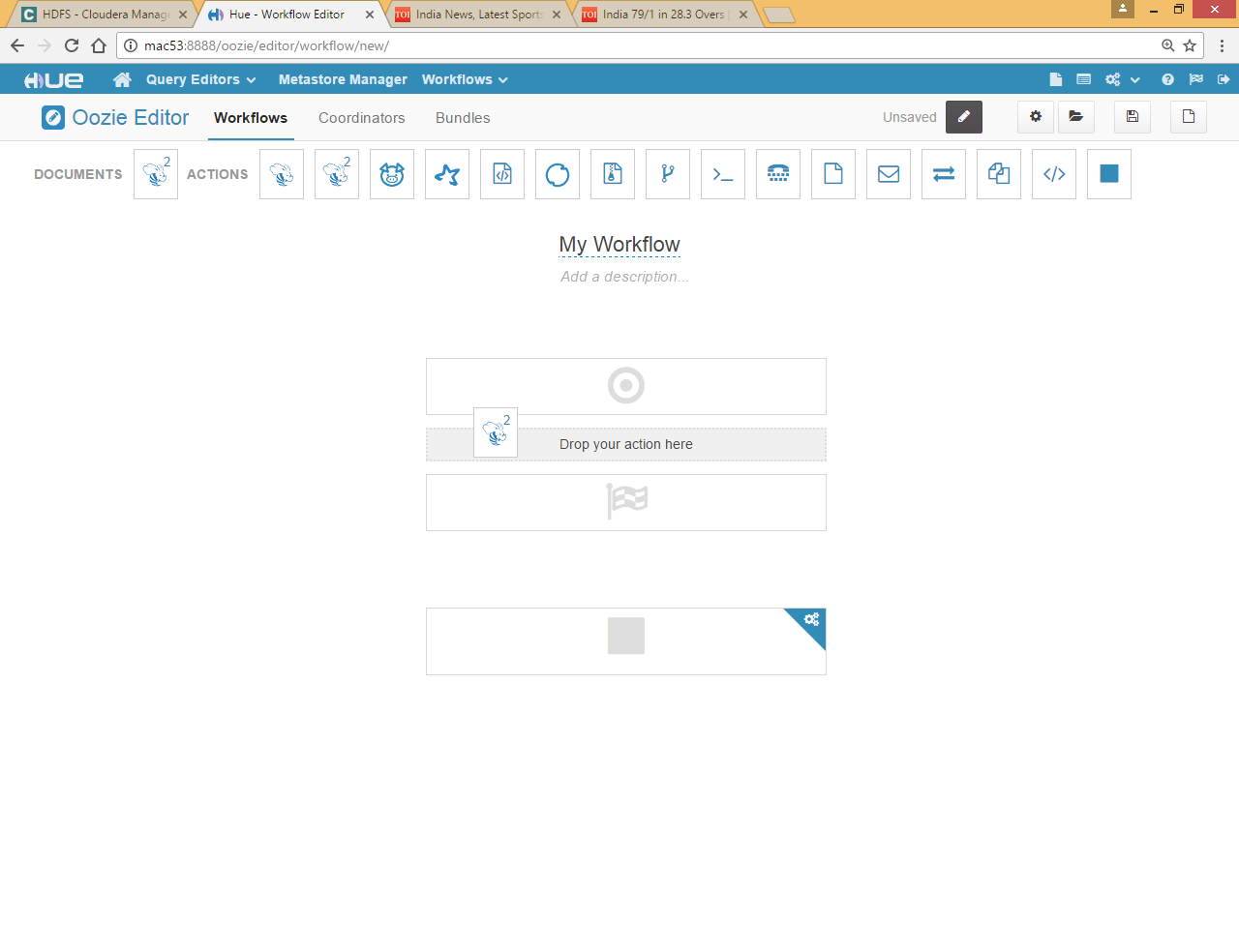
Give any name



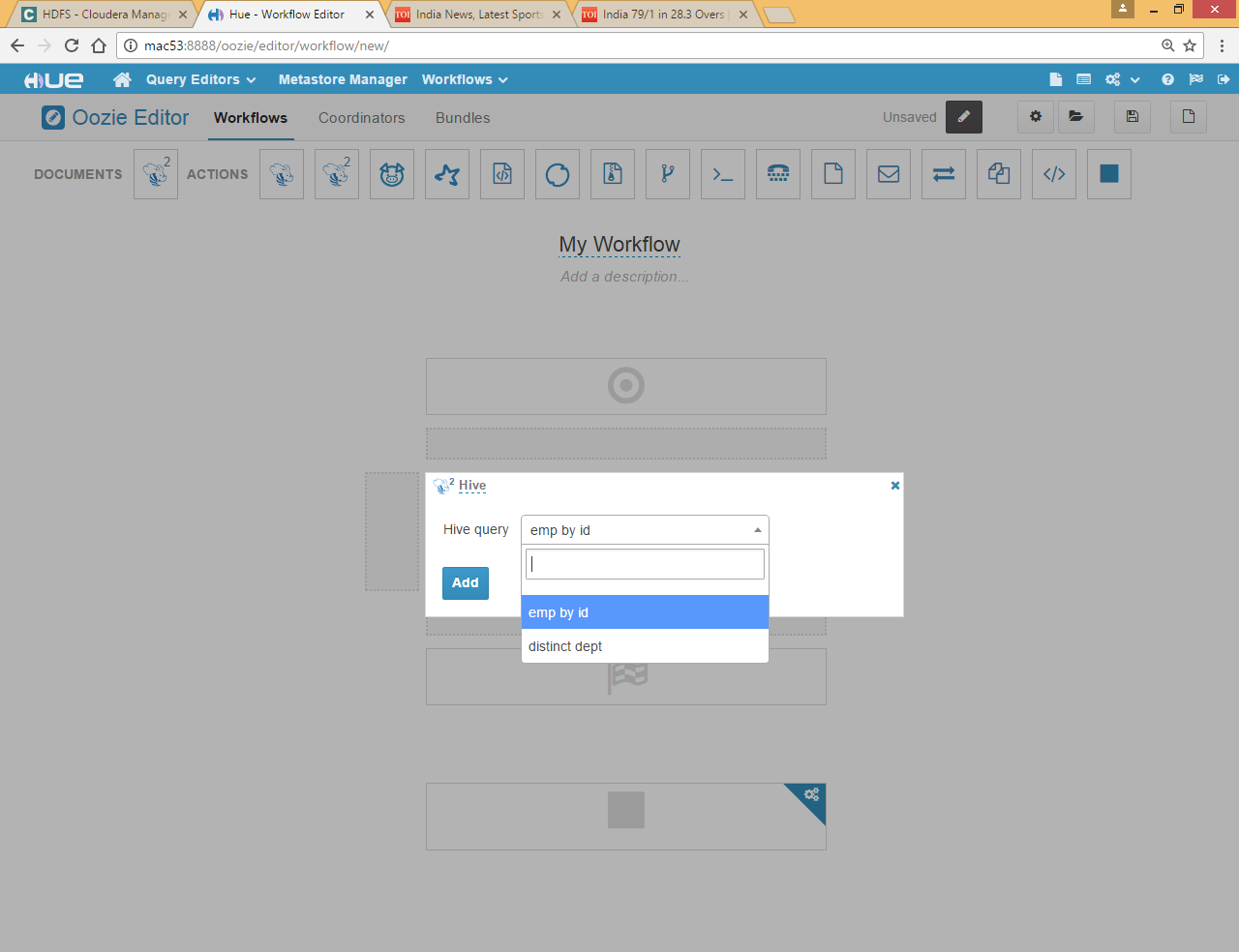
Now come to Oozie workflow and click create



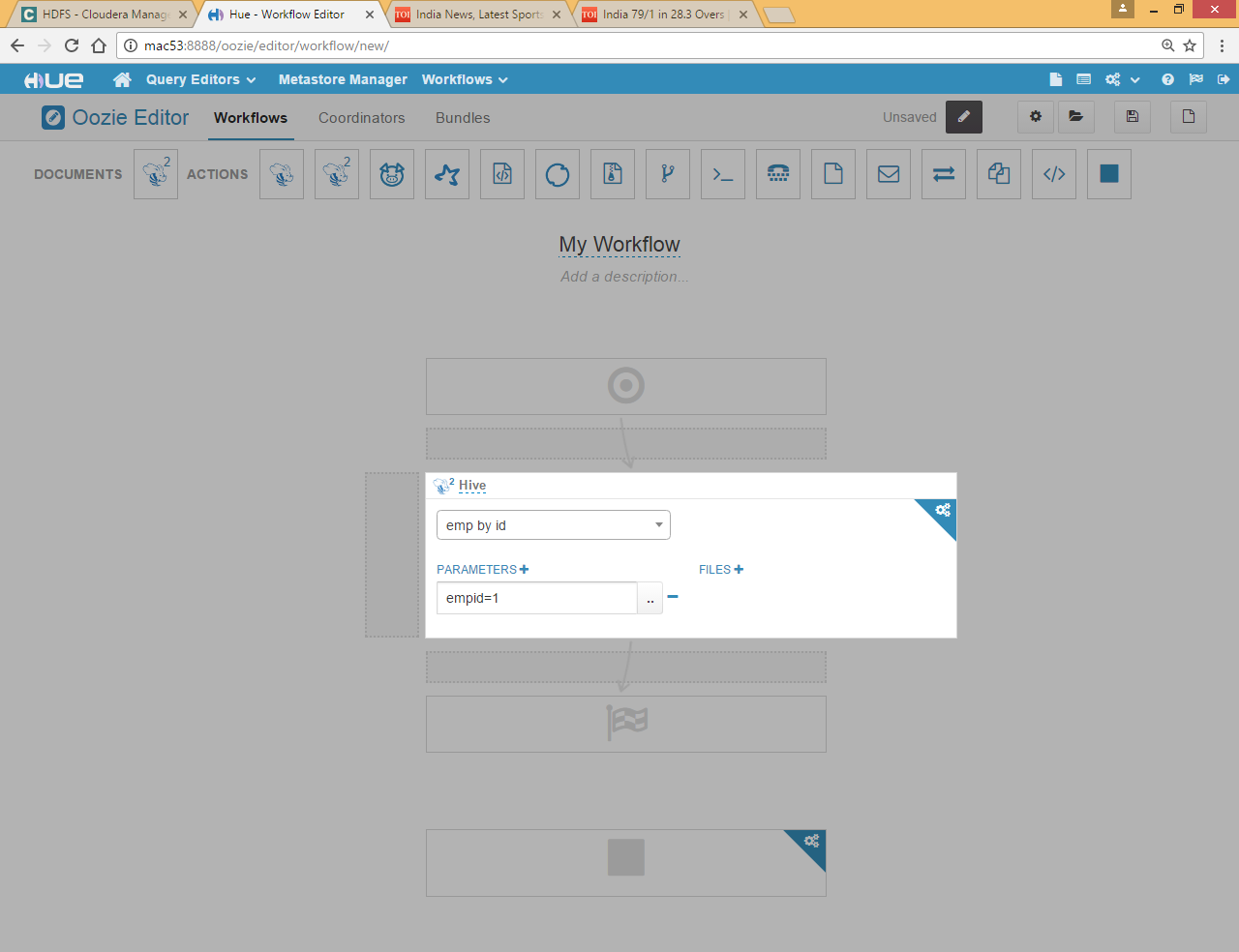
Drag drop saved hive query



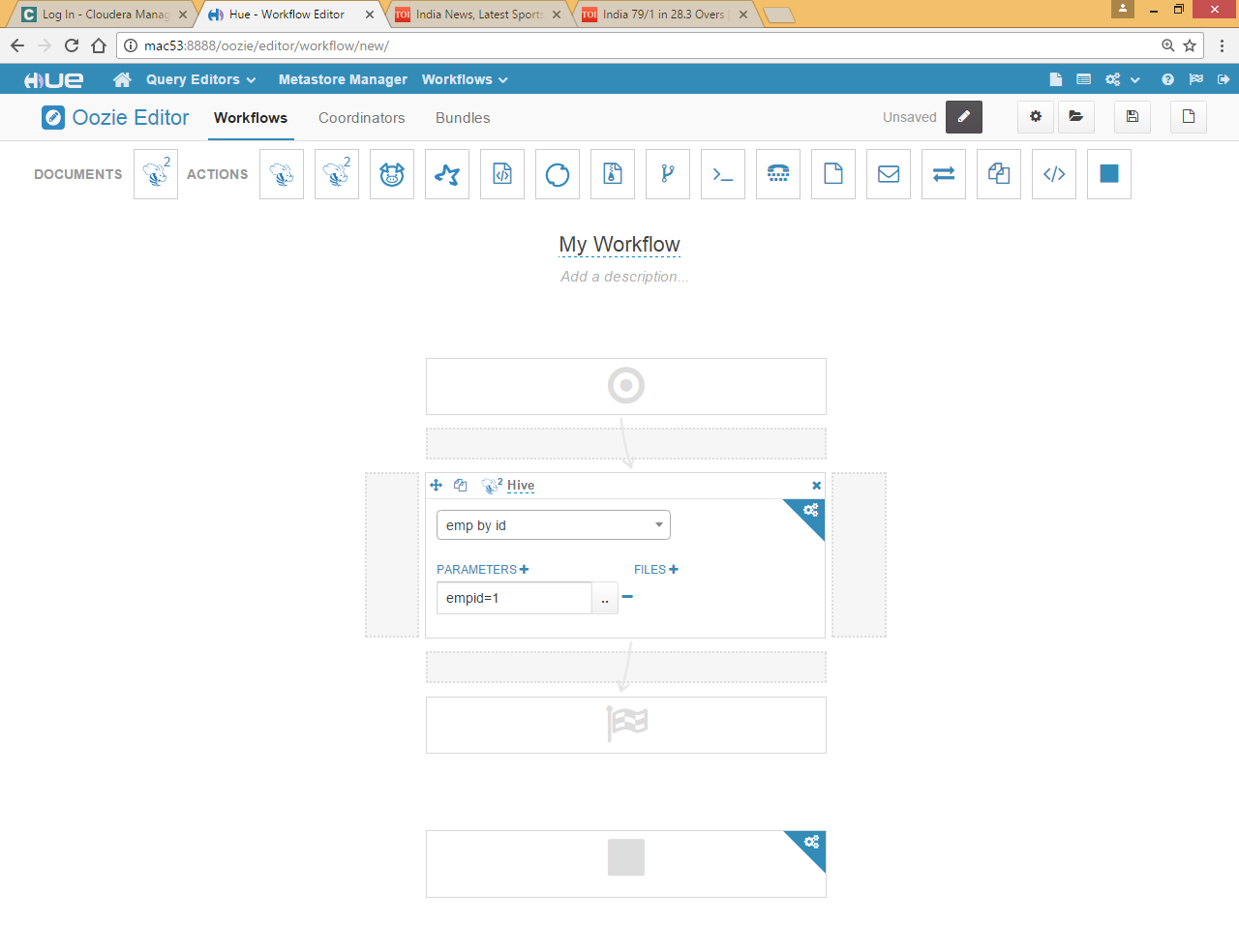
Choose save query



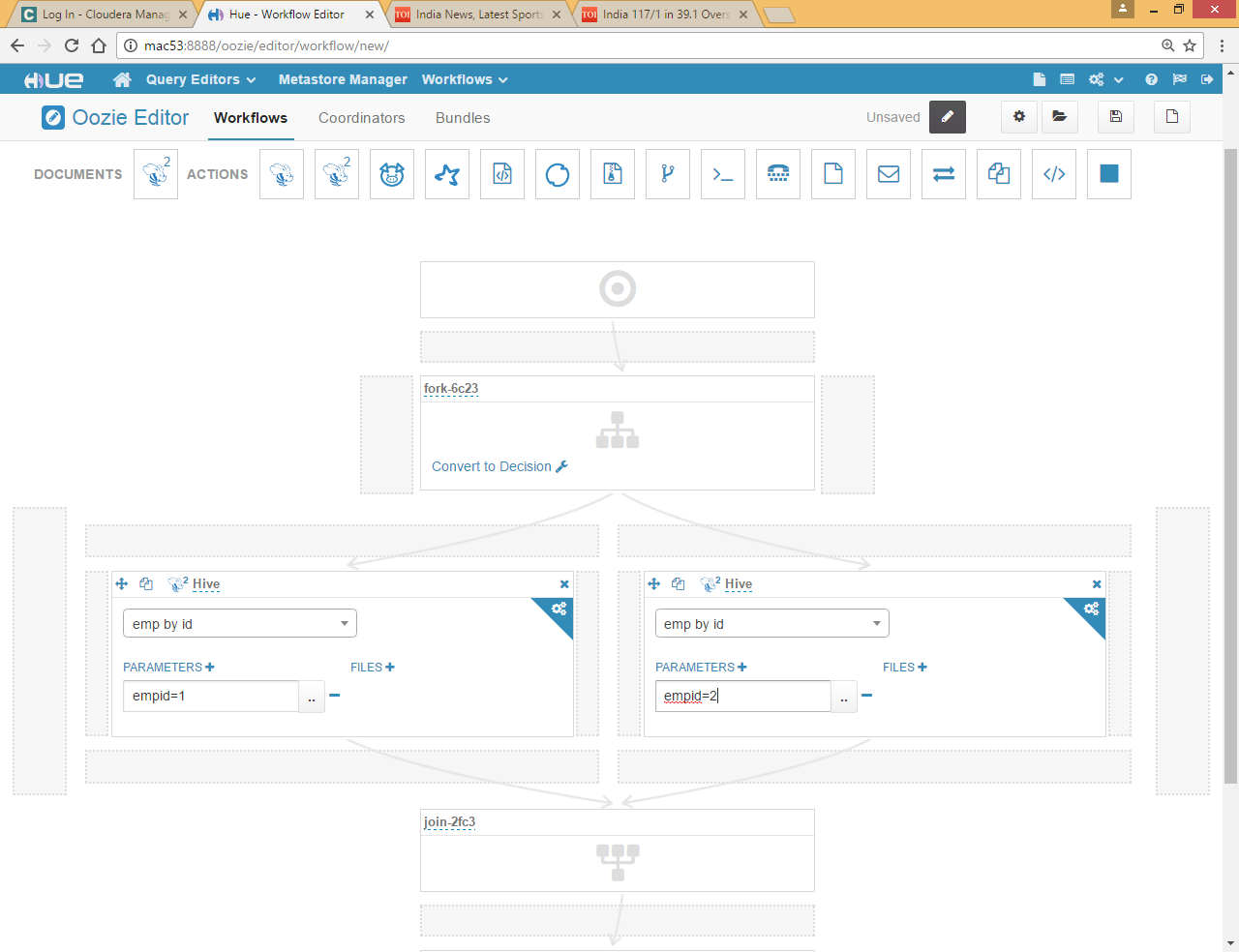
Select empid as empid is a variable there



We can copy the same node

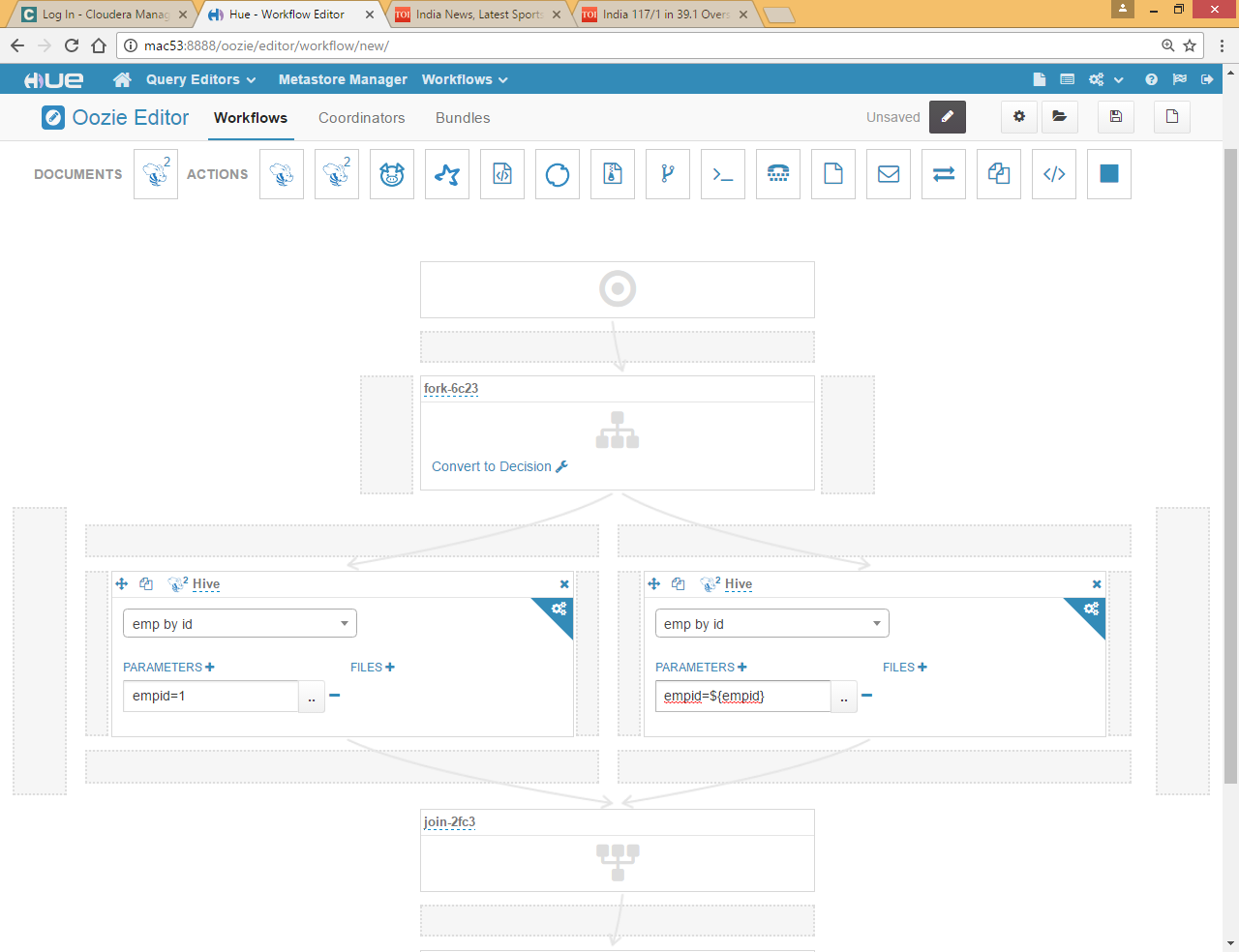


drag  icon and drop it to right box

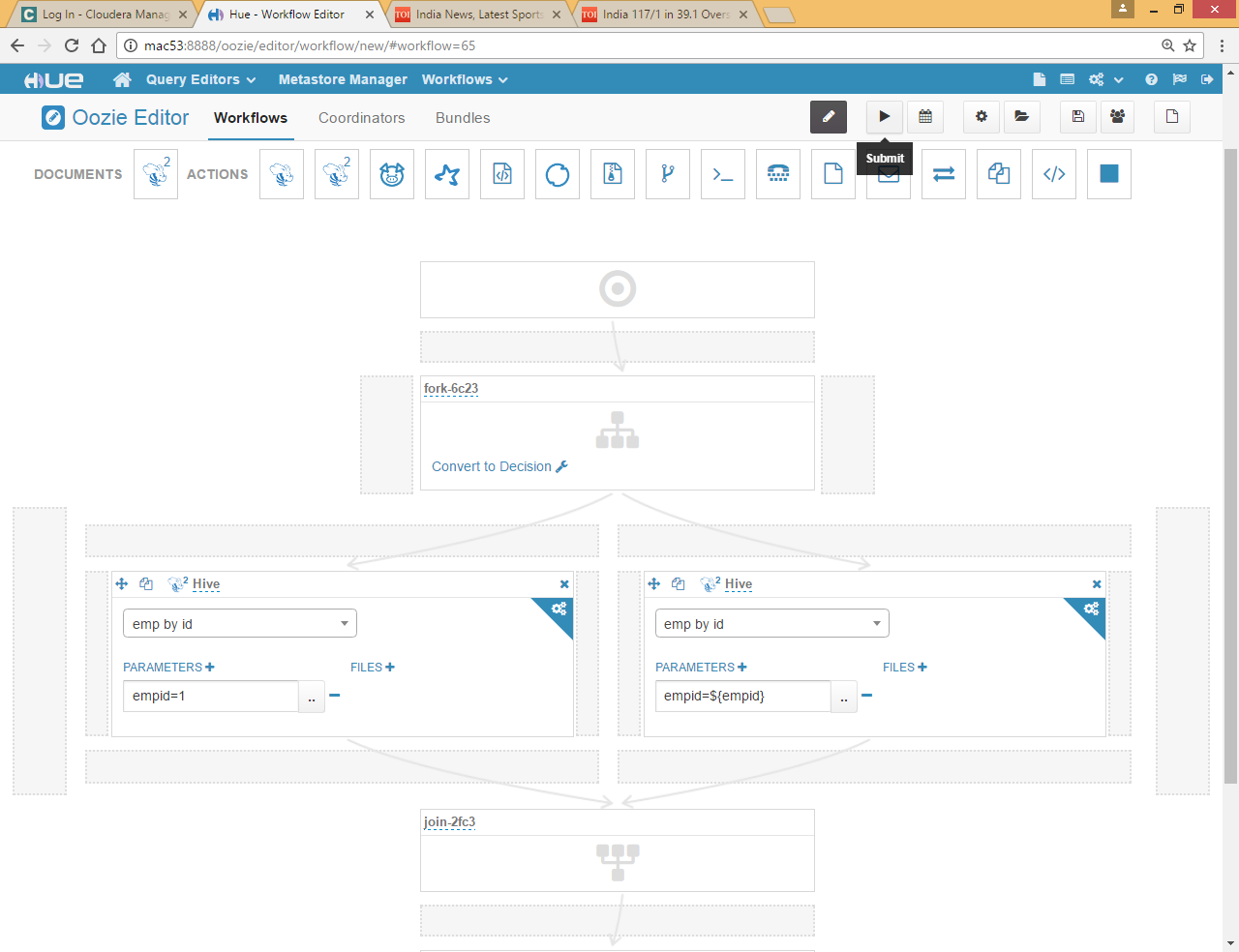


Give some different value of empid

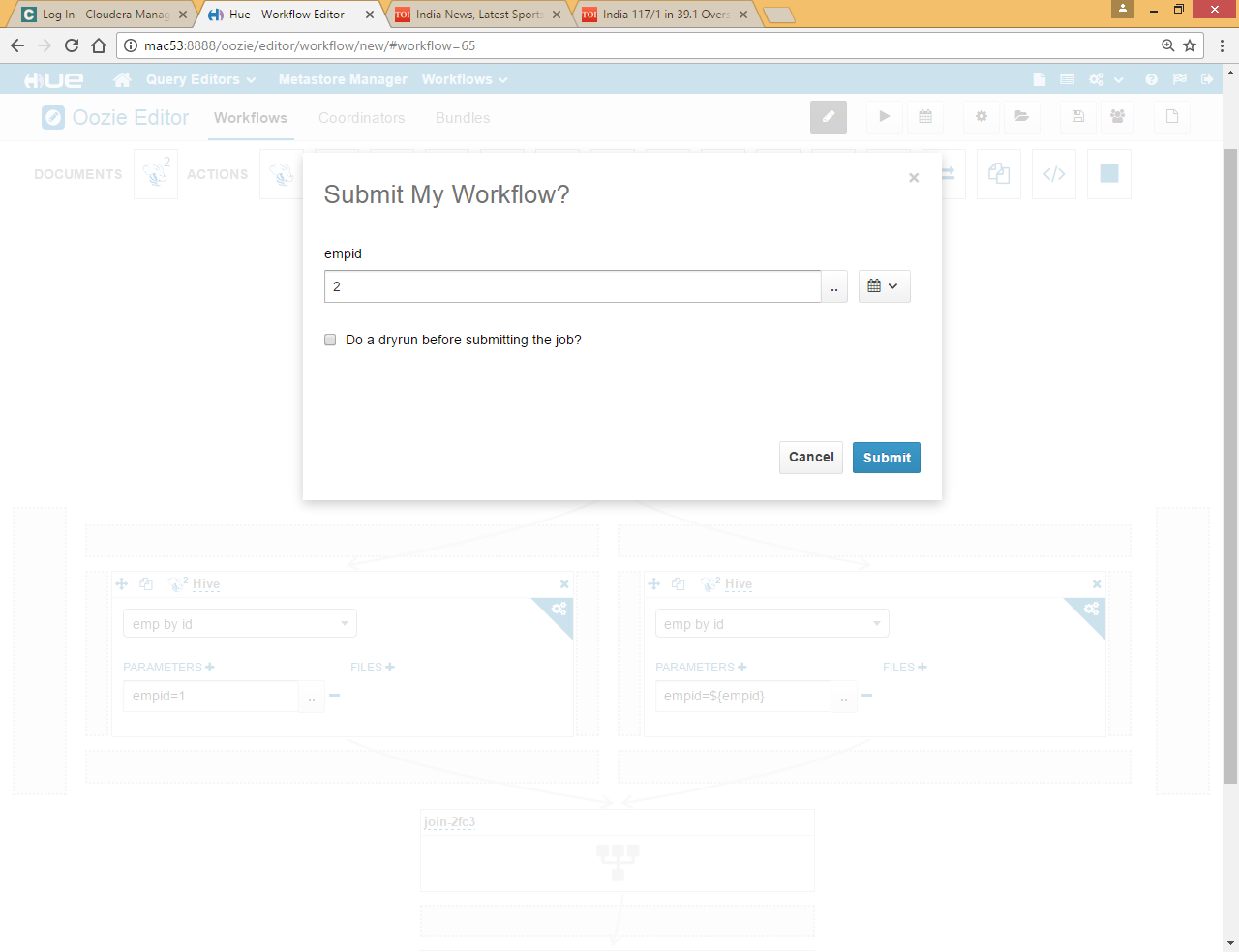
Or you can also parameterized it instead of hard coding here



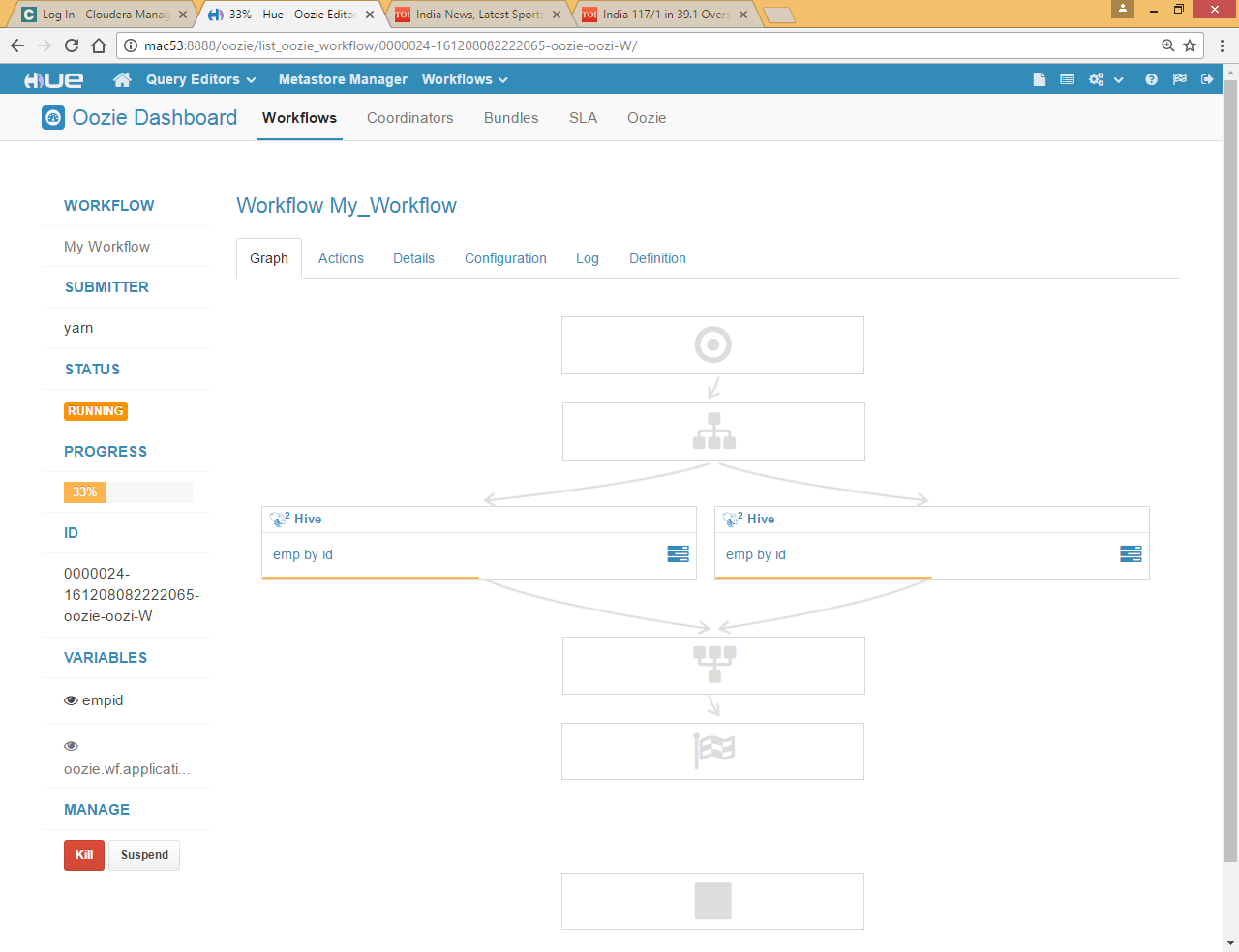
Save it and submit



Here it will ask you for parameter

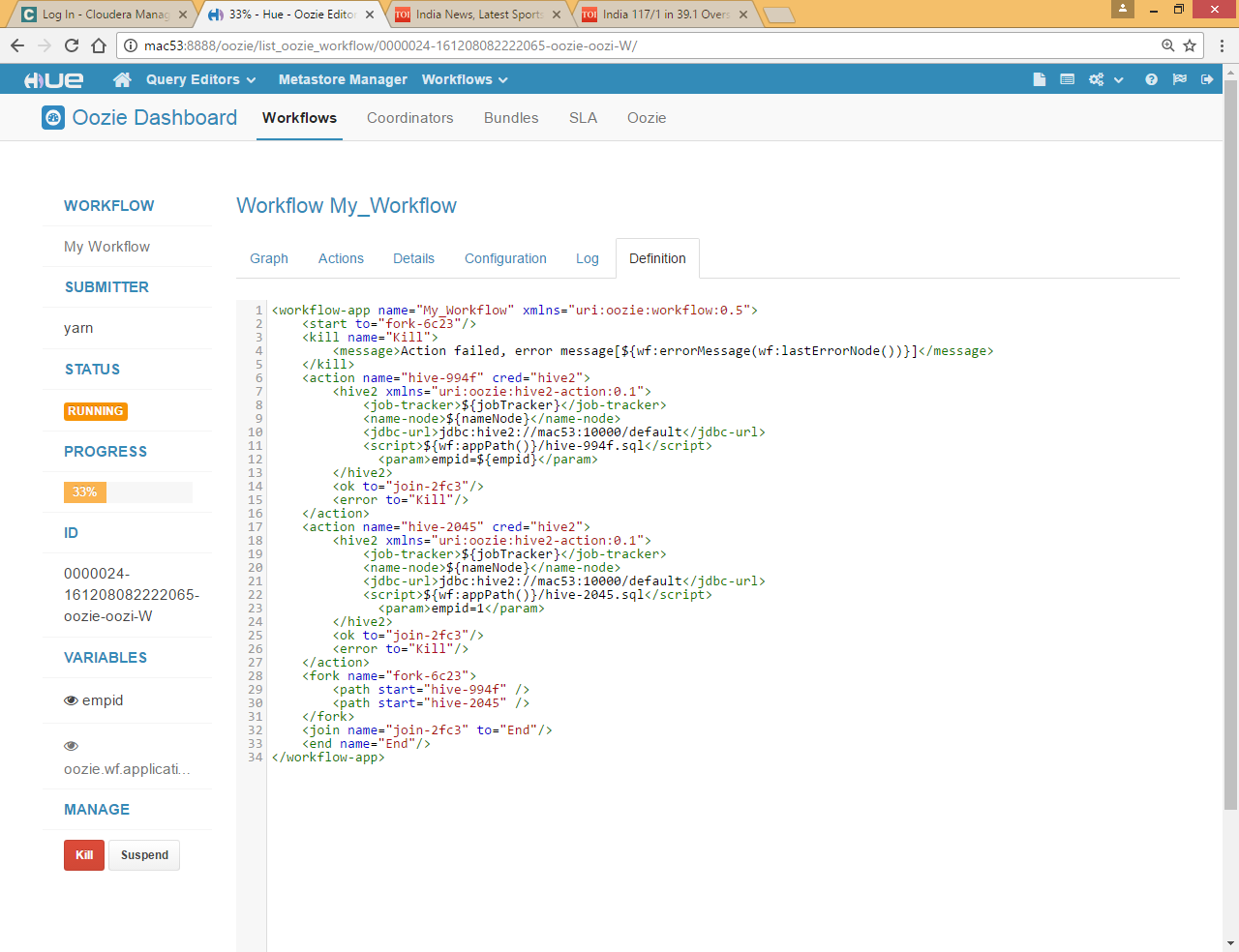


Click submit



Both queries are running parallel

You can check Definition tab to generated xml



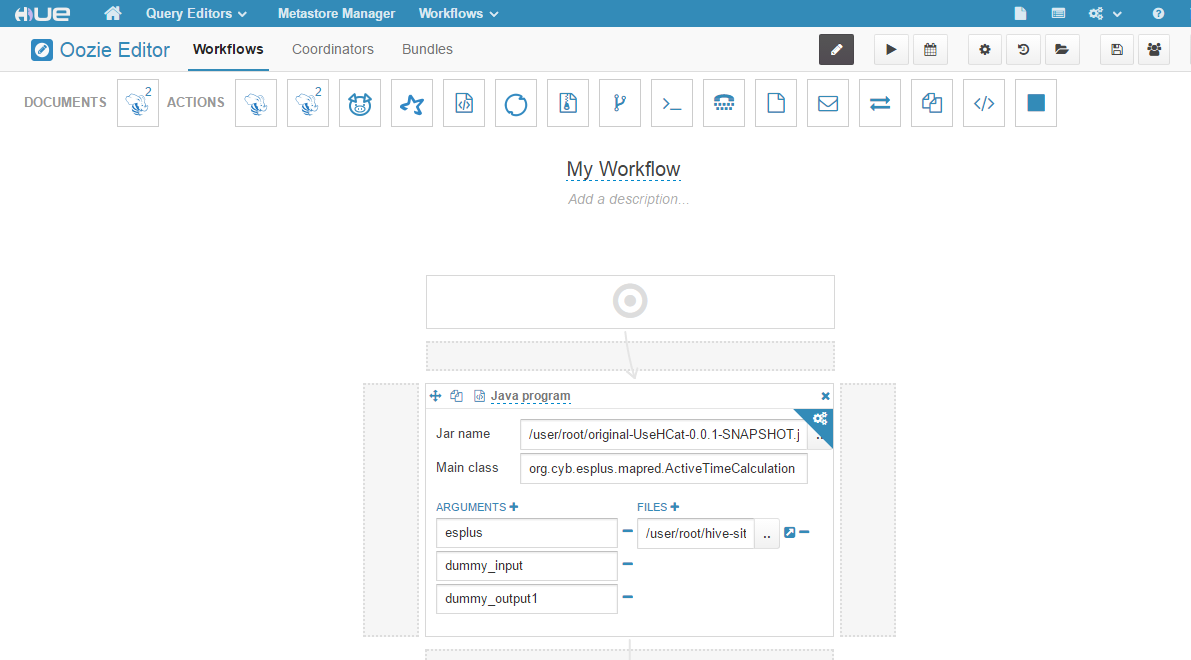
# Oozie workflow for java Action

Go to WorkFlow>>Editors>>Workflows and click on Create Button

Now just drag and drop java Program 

Now give the jar name and main class

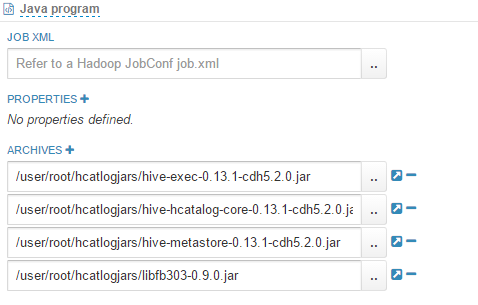
If java class requires some arguments select it using arguments tag, in case of hcatlog java program for map reduce we need to select hive-site.xml file for this you can use Files tag to choose the file



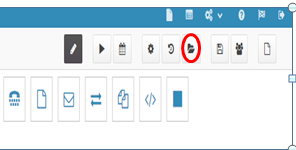
We can define third party jars in class path for that you need to click on jar name … button

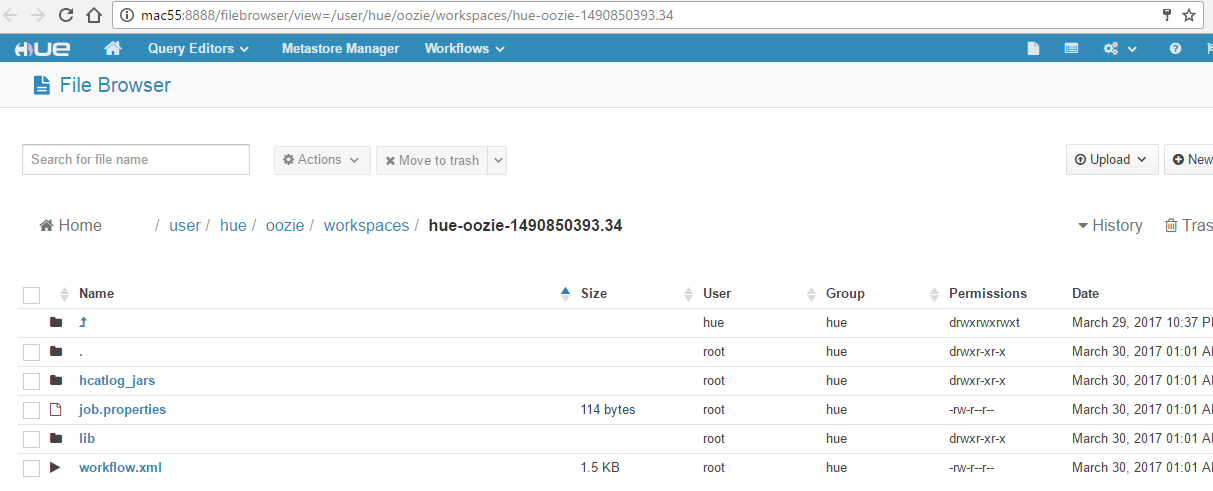


Now in Achieve tag select individual jar files



By clicking on workspace button  you can jump to actual workspace





Here you will find below folder list

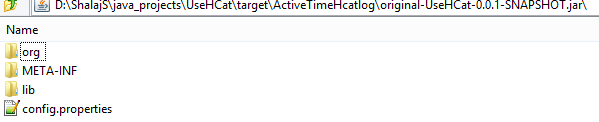
**lib** folder is folder where your main application jar will be uploaded

**Workflow.xml** contain actual workflow

**Job.properties** contains client side properties

Here **hcatlog\_jars** is manually created, you can ignore this folder

**Note : to include third party jars we can directly upload the jars inside lib folder , if these jars conflicts with shared lib jars then we need to include these jar files in lib folder under main application jar as well inside Archives tag otherwise oozie throws class not found exception**



Now you can save and submit the workflow

# Oozie Coordinator

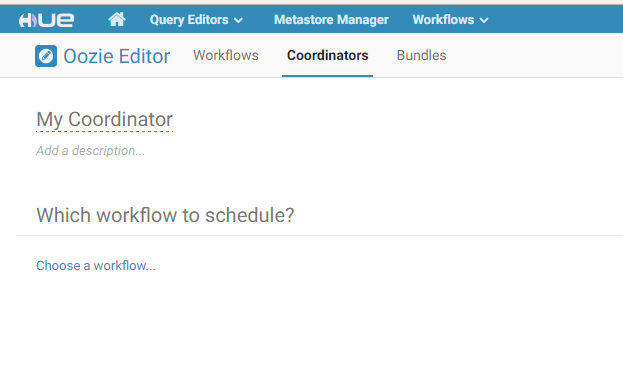
The Oozie Coordinator system allows the user to define and execute recurrent and interdependent workflow jobs

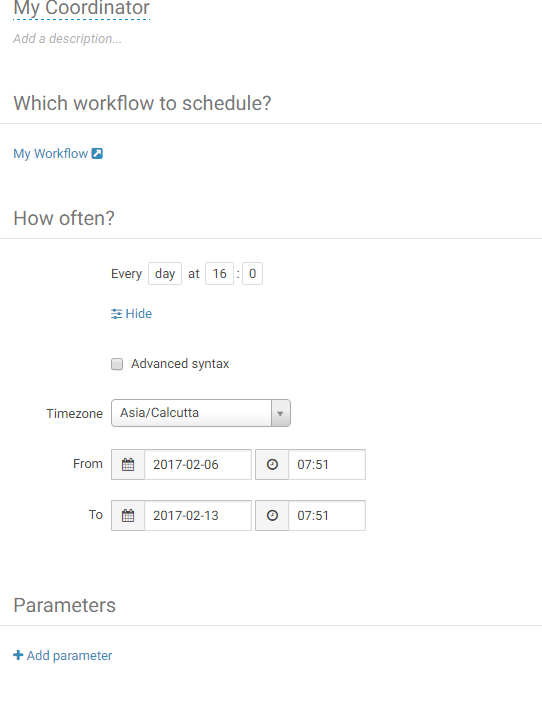
Define some workflow first and if you want to schedule this job we need to create coordinator

Go to Workflow >> Editors >> Coordinator

Click on Create button

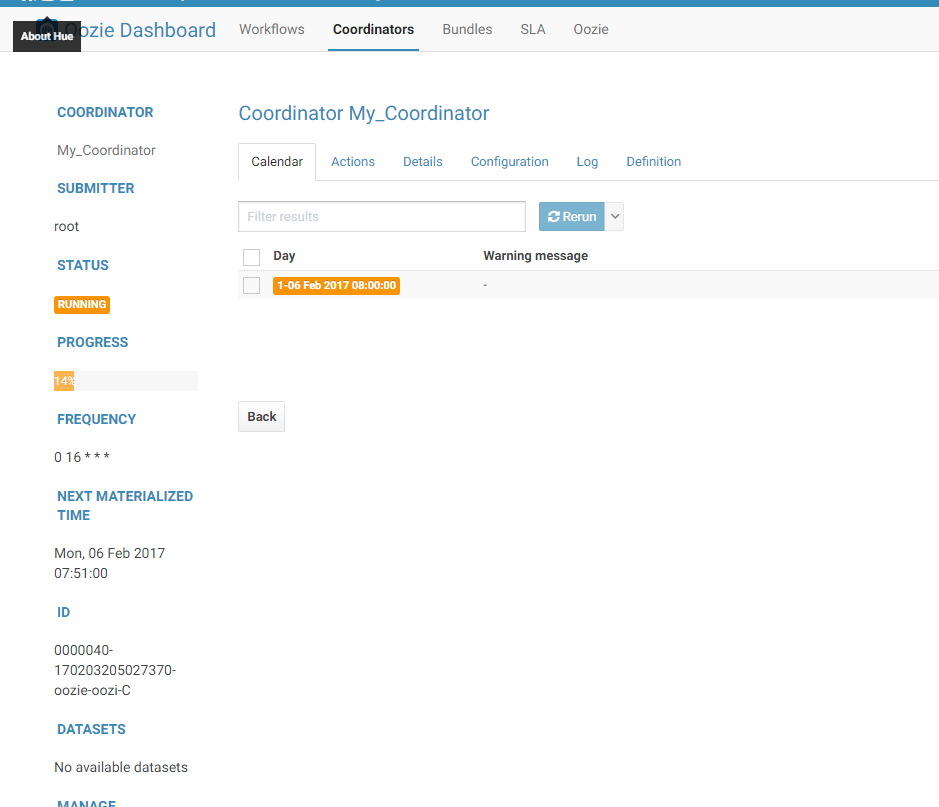
Click choose a workflow and select workflow



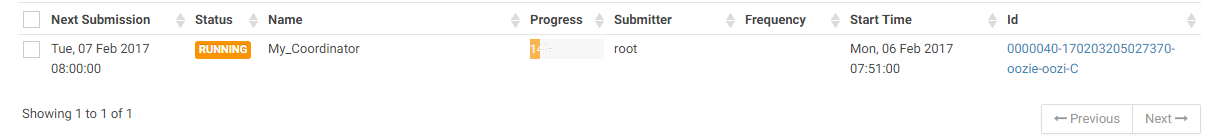


Now specify frequency from and to date to schedule this job, All datetime values are always in UTC so set it accordingly

If you have any parameters in workflow add it using Add parameter link save it and submit it



You can check the log in dashboard click Workflow >> Dashboard>>Coordinator



You can check the log from here