

Homework: Run a YARN Job

Files and Data Used in this Homework

Exercise directory: `$DEV1/exercises/yarn`

Dataset (HDFS): `/loudacre/kb`

In this assignment you will submit an application to the YARN cluster, and monitor the application using both the Hue Job Browser and the YARN Web UI.

The application you will run is provided for you. It is a simple Spark application written in Python that counts the occurrence of words in Loudacre’s customer service Knowledge Base (which you uploaded in the last exercise). The focus of this exercise is not on what the application does, but on how YARN distributes tasks in a job across a cluster, and how to monitor an application and view its log files.

Explore the YARN cluster

1. Visit the YARN Resource Manager (RM) UI in Firefox using the provided bookmark, or by going to URL `http://localhost:8088`

No jobs are currently running so the current view shows the cluster “at rest”.

2. Take note of the values in the Cluster Metrics section, which displays information about applications, memory utilization, and the status of worker nodes in the cluster.

Cluster Metrics														
Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes
0	0	0	0	0	0 B	2 GB	0 B	0	2	0	1	0	0	0

User Metrics for dr.wino

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved	VCores Used	VCores Pending	VCores Reserved
0	0	0	0	0	0	0	0 B	0 B	0 B	0	0	0

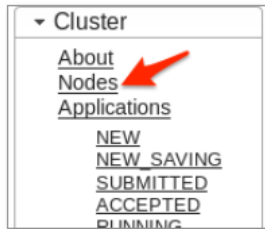
Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
No data available in table										

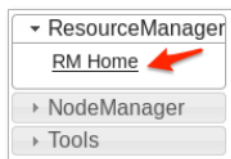
Showing 0 to 0 of 0 entries

First Previous Next Last

3. Click on the **Nodes** link in the Cluster menu on the left. The bottom section will display a list of worker nodes in the cluster. The pseudo-distributed cluster used for training has only a single node, which is running on the local machine. In the real world, this list would show multiple worker nodes.



4. Click on the **Node HTTP Address** to open the Node Manager UI on that specific node. This displays statistics about the selected node, including amount of available memory, currently running applications (none, currently) and so on.
5. To return to the Resource Manager, expand ResourceManager → RM Home on the left.



Submit an application to the YARN cluster

6. In a terminal window, change to the exercise directory:

```
$ cd $DEV1/exercises/yarn
```

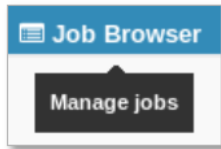
7. Run the example wordcount.py program on the YARN cluster to count the frequency of words in the knowledge dataset:

```
$ spark-submit --master yarn-cluster \  
wordcount.py /loudacre/kb/*
```

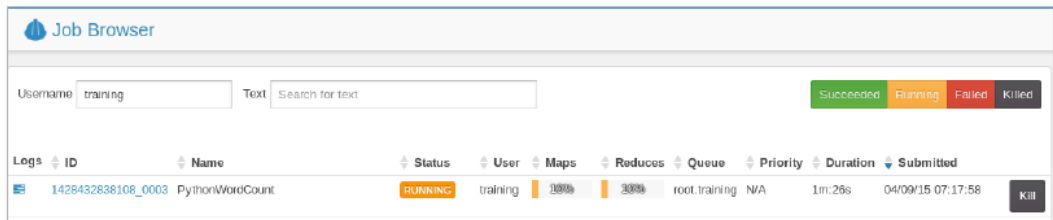
Later in the course you will learn how to write and submit your own Spark applications, as you are doing here, including what the various options mean. For now, focus on learning about the YARN UI.

View the application in the Hue Job Browser

8. Go to Hue in Firefox, and select the Job Browser. (Depending on the width of your browser, you may see the whole label, or just the icon.)



9. The Job Browser displays a list of currently running and recently completed applications. (If you don't see the application you just started, wait a few seconds, the page will automatically reload; it can take some time for the application to be accepted and start running.)



10. This page allows you to click the application ID to see details of the running application, or to kill a running job. (Don't do that now though!)

View the application in the YARN UI

To get a more detailed view of the cluster, use the YARN UI.

11. Reload the YARN RM page in Firefox. You will now see the application you just started in the bottom section of the RM home page.

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
1	0	1	0	2	2 GB	2 GB	1 GB	2	2	1	1	0	0	0	0

User Metrics for dr.who

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved	VCores Used	VCores Pending	VCores Reserved
0	0	1	0	0	0	0	0 B	0 B	0 B	0	0	0

Show 20 entries

Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
application_1428590029950_0001	training	PythonWordCount	SPARK	root.training	Thu Apr 9 07:53:57 -0700 2015	N/A	RUNNING	UNDEFINED	<div></div>	ApplicationMaster

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

12. As you did in the first exercise section, select **Nodes**.

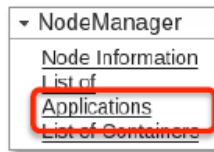
13. Select the **Node HTTP Address** to open the Node Manager UI.

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update
	/default-rack	RUNNING	localhost:59233	localhost:8042	Mon Aug 24 10:41:13 -0700 2015

14. Now that an application is running, you can click on **List of Applications** to see the application you submitted.

▼ NodeManager
Node Information
List of Applications
List of Containers

15. If your application is still running, try clicking on **List of Containers**.



This will display the containers the Resource Manager has allocated on the selected node for the current application. (No containers will show if no applications are running; if you missed it because the application completed, you can run the application again.)



Show 20 entries			Search:		
Containerid	ContainerState	logs			
container_1428590029950_0001_01_000001	RUNNING	logs			
container_1428590029950_0001_01_000002	RUNNING	logs			
Showing 1 to 2 of 2 entries			First	Previous	1 Next Last