

SETTING UP A SPARK CLUSTER:

Assuming you already have a Hadoop cluster set up, setting up Spark to run on top of it is fairly straight-forward.

These were the steps I took to install Spark on the CS machines on campus

- 1) Download Spark from <http://spark.apache.org/downloads.html> (I'm using 1.6.1, but any version should work)
- 2) Export the location of the Spark directory as a shell variable (i.e, edit your bashrc file and add the line "export SPARK_HOME=<path to Spark installation>")
- 3) Within SPARK_HOME, navigate into conf and open "spark-defaults.template". You will need to edit the following lines by uncommenting them and changing the values:

spark.master	spark://HDFSNameNode:anyPortNum
spark.eventLog.enabled	true
spark.eventLog.dir	hdfs://HDFSNameNode:NameNodeListeningPort/history
spark.serializer	org.apache.spark.serializer.KryoSerializer
spark.executor.memory	4g

- 4) Save "spark-defaults.template" as "spark-defaults.conf"
- 5) Copy your "slaves" file from your hadoop configuration into the conf directory of SPARK_HOME.
- 6) Now you should be ready to launch your cluster.
 - 6a) First start up HDFS (\$HADOOP_HOME/sbin/start-dfs.sh)
 - 6b) Start Yarn (\$HADOOP_HOME/sbin/start-yarn.sh)
 - 6c) Start Spark (\$SPARK_HOME/sbin/start-all.sh)

Hopefully there are no errors at this point. If there are, consult log files.

- 7) Check that Spark is up and running by going to your Yarn manager's web portal (the port number can be found in yarn-site.xml, property name: yarn.resourcemanager.webapp.address). It should look something like this:

All Applications

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
9	0	0	9	0	0 B	152 GB	0 B	0	152	0	13	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024,VCores:1>	<memory:8192,VCores:8>

Show 20 entries

application_id	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI	Blacklisted Nodes
application_1492722926822_0009	mroseliu	Test	SPARK	default	Thu Apr 20 17:17:34 -0600 2017	Thu Apr 20 17:29:45 -0600 2017	FINISHED	FAILED		History	N/A
application_1492722926822_0008	mroseliu	Test	SPARK	default	Thu Apr 20 17:10:35 -0600 2017	Thu Apr 20 17:12:43 -0600 2017	FINISHED	FAILED		History	N/A
application_1492722926822_0007	mroseliu	Test	SPARK	default	Thu Apr 20 17:04:07 -0600 2017	Thu Apr 20 17:06:26 -0600 2017	FINISHED	FAILED		History	N/A
application_1492722926822_0006	mroseliu	Test	SPARK	default	Thu Apr 20 16:38:16 -0600 2017	Thu Apr 20 16:39:07 -0600 2017	FINISHED	SUCCEEDED		History	N/A
application_1492722926822_0005	mroseliu	Test	SPARK	default	Thu Apr 20 16:36:49 -0600 2017	Thu Apr 20 16:37:19 -0600 2017	FINISHED	FAILED		History	N/A
application_1492722926822_0004	mroseliu	Test	SPARK	default	Thu Apr 20 16:25:06 -0600 2017	Thu Apr 20 16:32:42 -0600 2017	FINISHED	FAILED		History	N/A
application_1492722926822_0003	mroseliu	Test	SPARK	default	Thu Apr 20 15:26:42 -0600 2017	Thu Apr 20 15:31:39 -0600 2017	FINISHED	FAILED		History	N/A
application_1492722926822_0002	mroseliu	Test	SPARK	default	Thu Apr 20 15:22:20 -0600 2017	Thu Apr 20 15:23:42 -0600 2017	FINISHED	FAILED		History	N/A
application_1492722926822_0001	mroseliu	Test	SPARK	default	Thu Apr 20 15:16:35 -0600 2017	Thu Apr 20 15:17:38 -0600 2017	FINISHED	FAILED		History	N/A

Showing 1 to 9 of 9 entries

You can view the status of any job by clicking the application number. You can then see the logs associate with that task to resolve errors.

8) To submit a job, create your class and also create a jar file. Make sure that HDFS, Yarn, and Spark are all up and running. The command that I have been using is:

`$SPARK_HOME/bin/spark-submit --class <name of your main class> --master yarn --deploy-mode cluster <path to jar> <arguments to your main program>`