SPARK Installation

Download spark source from spark.apache.org.

Extract at appropriate location.
Assuming it is extracted at /opt/spark

Add $SPARK_HOME in .bashrc file.

export SPARK_HOME=/opt/spark

export PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:
$HBASE_HOME/bin:$HIVE_HOME/bin:$MAVEN_HOME/bin:$SPARK_HOME/bin:
$SPARK_HOME/sbin:

Spark Deamons:

Spark-shell is available for python, R and scala.

To start python shell, type “pyspark”

To start scala shell, type “spark-shell”

Spark can be deployed with different options:

1. Local mode :

- Single machine
- Used for learning purpose, testing…
- Operating system becomes resource manager in this mode
- Command to start spark shell: spark-shell (for scala), pyspark (for python)
- Spark shell GUI will be available at ‘localhost:4040’.
- Logs cannot be seen in spark’s history manager, instead everything is available at localhost:4040. For multiple shells, it takes port number 4041, 4042 so on.
- If master and slave is not started by following below steps, by default shells get started in local mode only and its logs can be seen on history server.

2. Standalone mode:

- Spark’s own cluster
- Spark’s own cluster manager is used as a resource manager
- Spark’s master has to started using ‘start-master.sh’. Master deamon can be seen using ‘jps’. Spark Master GUI is available at ‘localhost:8081’. It will have no workers initially.
- Spark’s slave has to be started using ‘start-slave.sh localhost:7077’. Worker deamon can be seen using ‘jps’. This worker gets added and can be seen in spark master GUI too.
- Command to start spark shell: spark-shell --master spark://localhost:7077 (for scala). This spark-shell can be seen in the list of running applications on spark master GUI.
- Spark shell GUI will be available at ‘localhost:4040’. If 4040 port number is already in use, it take 4041 and so on.

3. YARN mode:

- Hadoop's YARN used as a resource manager. So YARN and HDFS deamons must be running to start spark in this mode(start-dfs.sh, start-yarn.sh). HDFS deamons are needed because all logs are written on HDFS(if master is running). Spark history manager will not store logs for currently running applications. They can be seen on HDFS. Once completed, these logs can be seen on spark history server.
- To launch spark application in c luster mode, we have to use spark-submit command. We cannot run yarn-cluster mode via spark-shell because when we run spark application, driver program will be running as part application master container/process. So it is not possible to run cluster mode via spark-shell.
  (Note: If master is running, logs can be seen on HDFS for currently running application. Once completed they can be seen on spark history manager. If master is not running, logs can be seen at GUI of spark-shell which is localhost:4040)

To start history server,

1. go to conf directory of $SPARK_HOME
2. cp spark-defaults.conf.template spark-defaults.conf
3. uncomment following line in this file.
   spark.eventLog.enabled true

This is a configuration file where all properties are specified. Properties can also be specified at command line.

4. Create directory spark-events in /tmp directory.
Cd /tmp
mkdir spark-events

5. start-history-server.sh

Web GUI for history server available at localhost:18080

To launch a Spark application in `cluster` mode:

```
$ ./bin/spark-submit --class path.to.your.Class --master yarn --deploy-mode cluster
[options] <app jar> [app options]
```

Example: (Note: removing --queue option works)

```
$ ./bin/spark-submit --class org.apache.spark.examples.SparkPi \
    --master yarn \
    --deploy-mode cluster \
    --driver-memory 4g \
    --executor-memory 2g \
    --executor-cores 1 \
    --queue thequeue \
    examples/jars/spark-examples*.jar \
    10
```