apache-spark-multi-node-installation / index.md

OS: Ubuntu 16.04 Desktop

Spark Version: 2.1.0 stable

Oracle VM VirtualBox

This tutorial is to install the Spark 2.1.0 stable version on the Ubuntu 16.04.

Install Spark on master

Prerequisites: Install Java and Scala on the Master node

Install Java on master

```
$ sudo apt-get install python-software-properties
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install oracle-java8-installer
```

Install Scala

```
$ sudo apt-get install scala
```

Configure hosts file

To edit the host file, use the following commands,

```
$ sudo vim /etc/hosts
```

Contents of `hosts` file should be as,

```
MASTER-IP master
SLAVE01-IP slave01
SLAVE02-IP slave02
```

Replace the MASTER-IP with the IP address of master and similarly for slaves. And, make sure you have the vim or you can use any editor.

(Note: Comment the localhost already present in the hosts file if it gives problem)

Configure SSH

You need to configure ssh for password-less login from master to slaves and from all the slaves to master.

Install OpenSSH

Install openssh using following command as,

```
$ sudo apt-get install openssh-server openssh-client
```

Generate Key-Pairs

```
$ ssh-keygen -t rsa -P ""
```

Configuring password-less SSH

Copying the content of `~/.ssh/id_rsa.pub` file to authorized_user file of `~/.ssh` folder in master as well as slave nodes.

```
$ ssh-copy-id ubuntu@master
$ ssh-copy-id ubuntu@slave01
$ ssh-copy-id ubuntu@slave02
```

Note: Make sure you have same username in all the machines. It will be easy to configure.

You can check whether password-less login is working or not by using following commands.

```
$ ssh master
$ ssh slave01
$ ssh slave02
```

Install Spark 2.1.0 on Master and Slave (Worker) nodes

You can either download from http://spark.apache.org/downloads.html or can directly download using wget command.

```
$ wget http://d3kbcqa49mib13.cloudfront.net/spark-2.1.0-bin-hadoop2.7.tgz
$ tar xvf spark-2.1.0-bin-hadoop2.7.tgz
```

Configuration of .bashrc file

Add the following lines to configure scala and spark in `.bashrc` file.

```
#scala
export SCALA_HOME=/usr/local/src/scala/scala-2.10.4
export PATH=$SCALA_HOME/bin:$PATH

#Spark
export SPARK_HOME=/home/ubuntu/spark-2.1.0-bin-hadoop2.7
export PATH=$SPARK_HOME/bin:$PATH
```

Restart .bashrc configuration

To restart the `.bashrc` configuration:

```
$ source ~/.bashrc
```

Configuring Spark configuration files:

You need to configure `spark-defaults.conf`, `spark-env.sh` and `slaves` files on master node.

Configure spark-defaults.conf

You need to create the `spark-defaults.conf` file and update as,

```
$ cp spark-defaults.conf.template spark-defaults.conf
$ vim spark-defaults.conf
```

Update the `spark-defaults.conf` file with the following line as,

```
spark.master                     spark://master-pc:7077
```

Configure spark-env.sh

Add the following lines in `spark-env.sh` file as,

```
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
export SPARK_WORKER_CORES=8
export SPARK_MASTER_HOST=MASTER-IP
export SPARK_LOCAL_IP=MASTER-IP
```

Configure slaves file

Create/Update the `slave` file and add the list of slaves in slave file with `slave01`, `slave02` as,

```
slave01
slave02
```

Note: Name should be same as that added in the `/etc/hosts` file.

Installation on Spark Worker nodes (Slaves)

Setup prequisities on all slaves as done in master nodes,

- Entries to hosts file
- Install scala and java

From the master node, you can directly secure copy the downloaded spark file to all the slaves as,

```
$ scp spark-2.1.0-bin-hadoop2.7.tgz slave01:~
$ scp spark-2.1.0-bin-hadoop2.7.tgz slave02:~
```

Or, you could simply copy using your flash-drive or download as shown for the master node in the upper sections.

Now ssh into the `slave01` and `slave02` from the `master` node separately to unzip the spark file by using following command,

```
$ ssh slave01
```

Inside the slave01 node, use the tar command to extract compressed spark file as, $ tar xvf spark-2.1.0-bin-hadoop2.7.tgz

Similarly, repeat the process for slave02.

Now, the configuration is all set.

Running Spark Clusters

Start Spark master (on Master node)

In order to start the `master` node, you should run `start-master.sh` from the `sbin` directory inside your spark folder as,

```
~/spark-2.1.0-bin-hadoop2.7/sbin$ ./start-master.sh
```

Start Spark slaves (on Master node)

In order to start the `slave` nodes, you should run `start-slaves.sh` from the `sbin` directory inside your spark folder as, ~/spark-2.1.0-bin-hadoop2.7/sbin$ ./start-slaves.sh spark://master-pc:7077

Check daemons on Master

In order to check whether the configuration is set or not, you could use `jps` command as,

```
$ jps
Master
```

Check daemons on Slave nodes

In order to check whether the configuration is set or not, you could use `jps` on all the `slave` nodes command as,

```
$ jps
Worker
```

Now, once every looks good as mentioned above, you can verify whether your slaves are connecting properly with the master node, you can go to the following address in your web browser,

```
http://master:8080/
```