HADOOP CLUSTER PROJECT REPORT ON UBUNTU SERVER 18.04

2019/2020

Prepared by:
EL MAGUI Fatiha
ESSOUSSY Ilham
JARMOUNE Mohamed

Introduction :

Apache Hadoop is a collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model.

The aim of this project is to install Hadoop cluster on ubuntu server 18.04.

First of all, we will need to create three virtual machines for our cluster : master, slave and slave1.

For this project, we installed Ubuntu Server 18.04 on all three servers, installed all updates and rebooted with making sure that the configuration of each server is with a static IP address.

Preparing the Hadoop Servers

Firstly, we installed Oracle java 8 inside of the three virtual machines:

```
@master:~$ java -version
jdk version "1.8.0_222"
JDK Runtime Environment (build 1.8.0_222-8u222-b10-1ubuntu1~18.04.1-b10
JDK 64-Bit Server VM (build 25.222-b10, mixed mode)
@master:~$
```

Next, we download the hadoop Binary 3.2.1 from the Apache SOFTWARE FOUNDATION archive :

https://www-us.apache.org/dist/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz

we download Hadoop using wget command.

```
drwxr-xr-x 5 jarmoune jarmoune      4096 Nov 18 22:57 .
drwxr-xr-x 3 root     root          4096 Nov 18 17:01 ..
-rw-r--r-- 1 jarmoune jarmoune       220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 jarmoune jarmoune      3771 Apr  4  2018 .bashrc
drwx------ 2 jarmoune jarmoune      4096 Nov 18 17:04 .cache
drwx------ 3 jarmoune jarmoune      4096 Nov 18 17:04 .gnupg
drwxr-xr-x 9     1001     1001      4096 Sep 10 16:51 hadoop-3.2.1
-rw-r--r-- 1 root     root     359196911 Sep 23 05:16 hadoop-3.2.1.tar.gz
-rw-r--r-- 1 jarmoune jarmoune       807 Apr  4  2018 .profile
-rw-r--r-- 1 jarmoune jarmoune         0 Nov 18 17:04 .sudo_as_admin_successful
```

After this step, we extract the archive and move it to /usr/local/ in each machine. Then , we copy

```
root@jarmoune:/home/jarmoune# ls -al /usr/local/
total 44
drwxr-xr-x 11 root   root  4096 Nov 18 23:54 .
drwxr-xr-x 10 root   root  4096 Nov 19 00:43 ..
drwxr-xr-x  2 root   root  4096 Aug  5 19:22 bin
drwxr-xr-x  2 root   root  4096 Aug  5 19:22 etc
drwxr-xr-x  2 root   root  4096 Aug  5 19:22 games
drwxrwxr-x 11 hadoop root  4096 Nov 19 11:55 hadoop
drwxr-xr-x  2 root   root  4096 Aug  5 19:22 include
drwxr-xr-x  3 root   root  4096 Aug  5 19:23 lib
lrwxrwxrwx  1 root   root     9 Aug  5 19:22 man -> share/man
drwxr-xr-x  2 root   root  4096 Aug  5 19:22 sbin
drwxr-xr-x  5 root   root  4096 Nov 18 21:16 share
drwxr-xr-x  2 root   root  4096 Aug  5 19:22 src
root@jarmoune:/home/jarmoune#
```

Hadoop-3.2.1 folder in each slave using scp command.

WE ensure that Hadoop folder is moved using the command ls.

In each machine, we add the hostname using the command : sudo vi /etc/hosts

```
127.0.0.1 localhost
#127.0.1.1 jarmoune
192.168.1.16 jarmoune
192.168.1.17 jarmoune1
192.168.1.20 jarmoune2
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

We need to know where Java is installed to. To find out, we run the following command:

Update -alternatives –display java

```
root@jarmoune:/home/jarmoune# update-alternatives --display java
java - auto mode
  link best version is /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
  link currently points to /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
  link java is /usr/bin/java
  slave java.1.gz is /usr/share/man/man1/java.1.gz
/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java - priority 1081
  slave java.1.gz: /usr/lib/jvm/java-8-openjdk-amd64/jre/man/man1/java.1.gz
root@jarmoune:/home/jarmoune# _
```

We Open /etc/environment and update the PATH line to include the Hadoop binary directories. We also add a line for the JAVA_HOME variable.

This command is also applied to the slaves.

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/
local/hadoop/bin:/usr/local/hadoop/sbin"

JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64"
```

Next, we will add a hadoop user and give them the correct permissions:

```
root@jarmoune1:/home/jarmoune1# adduser hadoop
Adding user `hadoop' ...
Adding new group `hadoop' (1001) ...
Adding new user `hadoop' (1001) with group `hadoop' ...
Creating home directory `/home/hadoop' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for hadoop
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@jarmoune1:/home/jarmoune1# usermod -aG hadoop hadoop
root@jarmoune1:/home/jarmoune1# chown hadoop:root -R /usr/local/hadoop/
root@jarmoune1:/home/jarmoune1# chmod g+rwx -R /usr/local/hadoop/
root@jarmoune1:/home/jarmoune1#
```

After, we Login as the hadoop user and generate the SSH Key. We only need to complete this step on the Hadoop Master.

```
root@jarmoune:/home/jarmoune# su - hadoop
hadoop@jarmoune:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Created directory '/home/hadoop/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match.  Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:+sXk8gpwhUzupGNLzN1yKj1ZwT8umPGhVcjETMWHvBs hadoop@jarmoune
The key's randomart image is:
+---[RSA 2048]----+
|      .+o+..     |
|     + =o.+ .    |
|      = * .o     |
|   o = o +E      |
|    O * S +o     |
|   o * # *..     |
|    o @ + =      |
|     . + =       |
|        o..      |
+----[SHA256]-----+
hadoop@jarmoune:~$
```

Then, we copy the SSH key to the slaves server(this step is done just in hadoop master):

```
hadoop@jarmoune:~$ ssh-copy-id hadoop@jarmoune2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/hadoop/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys
hadoop@jarmoune2's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'hadoop@jarmoune2'"
and check to make sure that only the key(s) you wanted were added.
```

Configuring the Hadoop Master

First, we Open the **/usr/local/hadoop/etc/hadoop/core-site.xml** file and enter the following:

```
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
        <property>
                <name>fs.default.name</name>
                <value>hdfs://jarmoune:9000</value>
        </property>
</configuration>
```

Second, we open the /usr/local/hadoop/etc/hadoop/hdfs-site.xml file and add the following:

```
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
        <property>
                <name>dfs.namenode.name.dir</name>
                <value>/usr/local/hadoop/data/nameNode</value>
        </property>
        <property>
                <name>dfs.datanode.data.dir</name>
                <value>/usr/local/hadoop/data/dataNode</value>
        </property>
        <property>
                <name>dfs.replication</name>
                <value>2</value>
        </property>
</configuration>
~
~
~
~
"/usr/local/hadoop/etc/hadoop/hdfs-site.xml" 32L, 1071C written
hadoop@jarmoune:~$
```

Then, we Open the /usr/local/hadoop/etc/hadoop/workers file and add (one for each of our Hadoop Nodes):

```
jarmoune1
jarmoune2
```

After that, we Copy the configuration files to each of our Hadoop Nodes from Hadoop Master.

```
"/usr/local/hadoop/etc/hadoop/workers" 2L, 20C written
hadoop@jarmoune:~$ scp /usr/local/hadoop/etc/hadoop/* jarmoune1:/usr/local/hadoop/etc/hadoop/
Enter passphrase for key '/home/hadoop/.ssh/id_rsa':
capacity-scheduler.xml                                      100% 8260   102.4KB/s   00:00
configuration.xsl                                           100% 1335    63.5KB/s   00:00
container-executor.cfg                                      100% 1940    61.7KB/s   00:00
core-site.xml                                               100%  868   543.7KB/s   00:00
hadoop-env.cmd                                              100% 3999     1.9MB/s   00:00
hadoop-env.sh                                               100%   16KB   7.8MB/s   00:00
hadoop-metrics2.properties                                  100% 3321   471.0KB/s   00:00
hadoop-policy.xml                                           100%   11KB   5.7MB/s   00:00
hadoop-user-functions.sh.example                            100% 3414     1.5MB/s   00:00
hdfs-site.xml                                               100% 1071     1.0MB/s   00:00
httpfs-env.sh                                               100% 1484   915.0KB/s   00:00
httpfs-log4j.properties                                     100% 1657     1.0MB/s   00:00
httpfs-signature.secret                                     100%   21     2.5KB/s   00:00
httpfs-site.xml                                             100%  620   216.6KB/s   00:00
kms-acls.xml                                                100% 3518     2.0MB/s   00:00
kms-env.sh                                                  100% 1351   694.3KB/s   00:00
kms-log4j.properties                                        100% 1860   812.9KB/s   00:00
kms-site.xml                                                100%  682   340.2KB/s   00:00
log4j.properties                                            100%   13KB 628.2KB/s   00:00
mapred-env.cmd                                              100%  951    28.5KB/s   00:00
mapred-env.sh                                               100% 1764    82.9KB/s   00:00
mapred-queues.xml.template                                  100% 4113     2.3MB/s   00:00
mapred-site.xml                                             100%  758   494.7KB/s   00:00
/usr/local/hadoop/etc/hadoop/shellprofile.d: not a regular file
ssl-client.xml.example                                      100% 2316     1.3MB/s   00:00
ssl-server.xml.example                                      100% 2697     1.0MB/s   00:00
user_ec_policies.xml.template                               100% 2642   903.2KB/s   00:00
workers                                                     100%   20    16.5KB/s   00:00
yarn-env.cmd                                                100% 2250     1.3MB/s   00:00
yarn-env.sh                                                 100% 6056     3.4MB/s   00:00
yarnservice-log4j.properties                                100% 2591     1.4MB/s   00:00
yarn-site.xml                                               100%  690   282.4KB/s   00:00
hadoop@jarmoune:~$
```

To activate the namenode and format the HDFS file system on master server, we tap the two following commands :

```
source /etc/environmnet
hdfs namenode -format
```

```
2019-11-19 13:32:31,823 INFO util.GSet: 0.029999999329447746% max memory 239.8 MB = 73.7 KB
2019-11-19 13:32:31,824 INFO util.GSet: capacity      = 2^13 = 8192 entries
Re-format filesystem in Storage Directory root= /usr/local/hadoop/data/nameNode; location= null ? (Y
 or N) Y
2019-11-19 13:32:43,632 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1277508204-127.0.1.1-15
74170363615
2019-11-19 13:32:43,633 INFO common.Storage: Will remove files: [/usr/local/hadoop/data/nameNode/cur
rent/fsimage_0000000000000000000, /usr/local/hadoop/data/nameNode/current/fsimage_0000000000000000000
0.md5, /usr/local/hadoop/data/nameNode/current/seen_txid, /usr/local/hadoop/data/nameNode/current/VE
RSION]
2019-11-19 13:32:43,690 INFO common.Storage: Storage directory /usr/local/hadoop/data/nameNode has b
een successfully formatted.
2019-11-19 13:32:43,773 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/local/hadoop/dat
a/nameNode/current/fsimage.ckpt_0000000000000000000 using no compression
2019-11-19 13:32:43,971 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/hadoop/data/nameN
ode/current/fsimage.ckpt_0000000000000000000 of size 401 bytes saved in 0 seconds .
2019-11-19 13:32:44,025 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid
>= 0
2019-11-19 13:32:44,036 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutd
own.
2019-11-19 13:32:44,039 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at jarmoune/127.0.1.1
************************************************************/
```

We start HDFS :

We Validate that everything started right by running the **`jps`** command as the Hadoop user on all our Hadoop servers.

On the Hadoop Master:

```
hadoop@jarmoune:~$ start-dfs.sh
Starting namenodes on [jarmoune]
Starting datanodes
jarmoune1: WARNING: /usr/local/hadoop/logs does not exist. Creating.
jarmoune2: WARNING: /usr/local/hadoop/logs does not exist. Creating.
Starting secondary namenodes [jarmoune]
hadoop@jarmoune:~$ jps
6550 SecondaryNameNode
4220 ResourceManager
6717 Jps
6303 NameNode
```

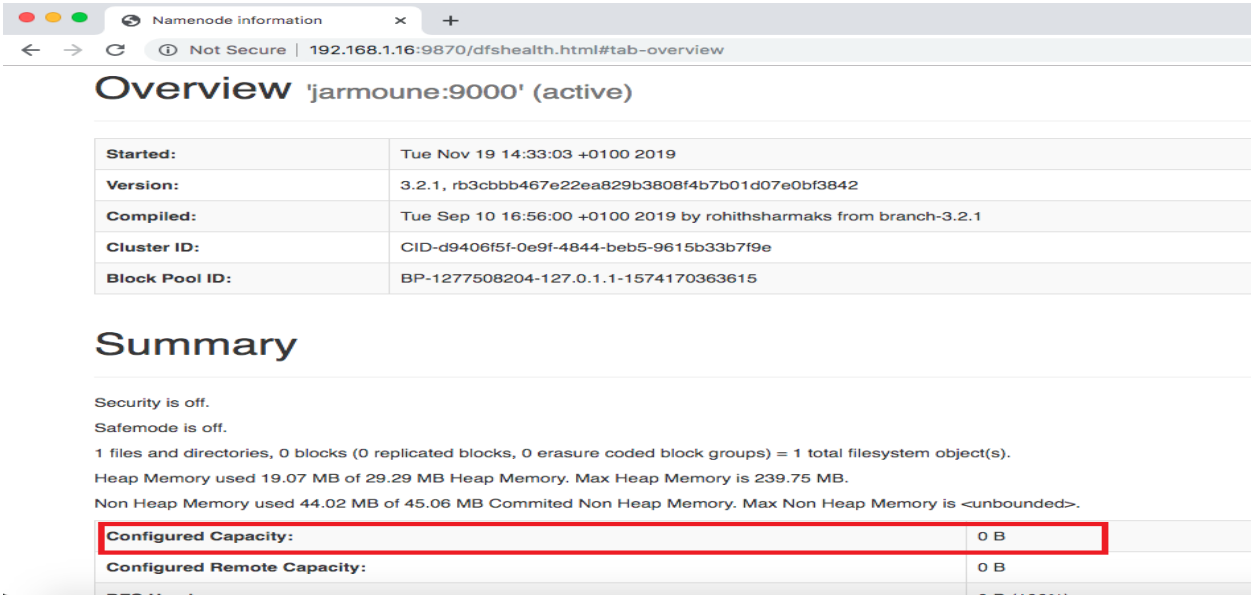
On each of our Hadoop Nodes:

```
hadoop@jarmoune2:~$ jps
1890 Jps
1787 DataNode
```

```
hadoop@jarmoune1:~$ jps
1819 DataNode
1918 Jps
```

Now, we access the HDFS web UI by browsing to our Hadoop Master Server port 9870. But we found that a problem in the configured Capacity: we found 0Bit and that's an error:

To resolve the problem, the fisrt thing to do is to stop Hadoop daemons on master:



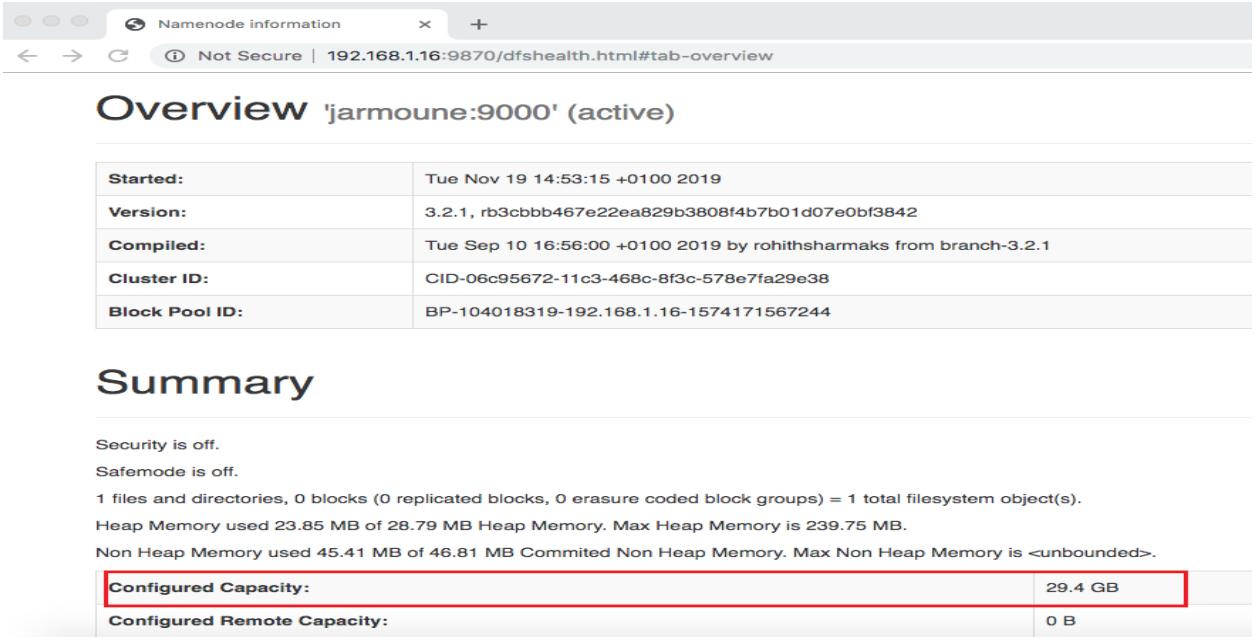
```
hadoop@jarmoune:~$ stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [jarmoune]
Stopping datanodes
jarmoune1: ssh: connect to host jarmoune1 port 22: Connection timed out
jarmoune2: ssh: connect to host jarmoune2 port 22: Connection timed out
Stopping secondary namenodes [jarmoune]
Stopping nodemanagers
jarmoune2: ssh: connect to host jarmoune2 port 22: No route to host
jarmoune1: ssh: connect to host jarmoune1 port 22: No route to host
Stopping resourcemanager
hadoop@jarmoune:~$ stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [jarmoune]
Stopping datanodes
Stopping secondary namenodes [jarmoune]
Stopping nodemanagers
Stopping resourcemanager
```

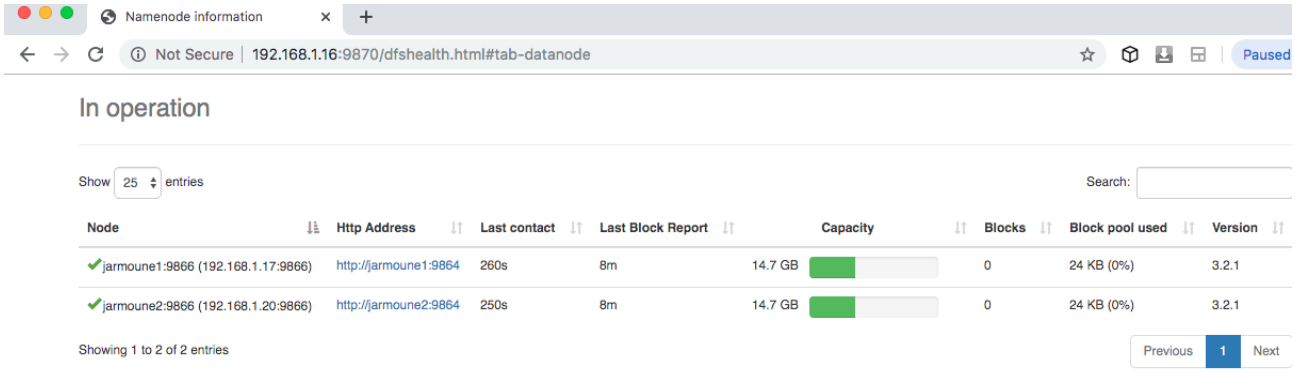
After, we go to /etc/hosts/ and remark $jarmoune:

```
127.0.0.1 localhost
#127.0.1.1 jarmoune ------>this is the problem
192.168.1.16 jarmoune
192.168.1.17 jarmoune1
192.168.1.20 jarmoune2
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

On restart HDFS format using the command: **start-dfs.sh**

Then, we enter again to the web UI:





Now that HDFS is running we are ready to start the Yarn scheduler. Hadoop, on its own, can schedule any jobs so we need to run Yarn so we can schedule jobs on our Hadoop cluster.

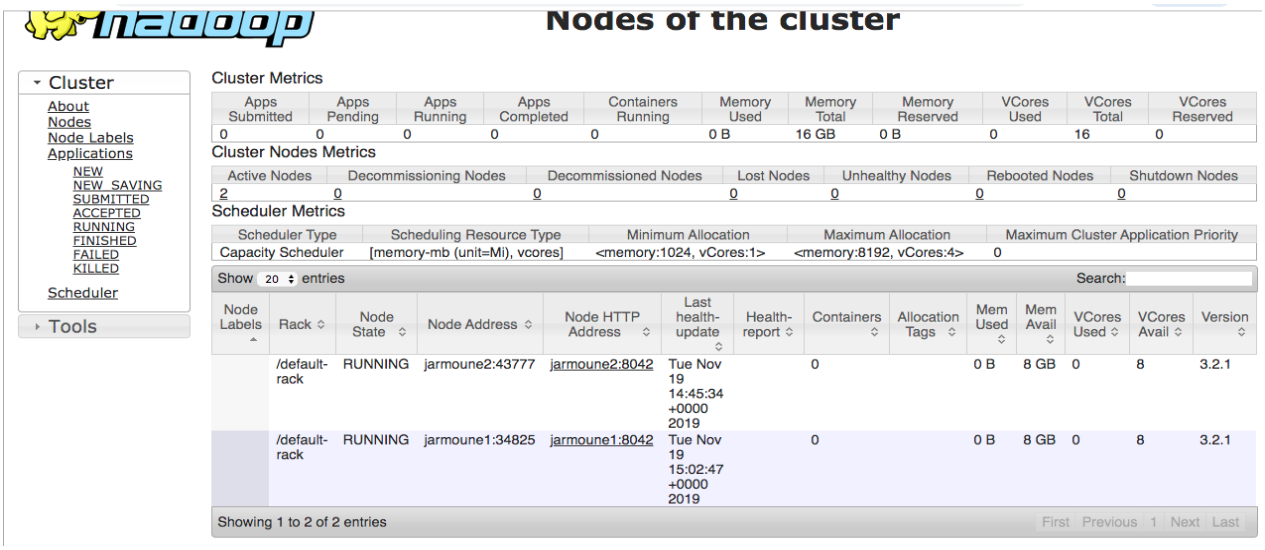
On each of our hadoop slaves (jarmoune1 and jarmoune2) we added these lines

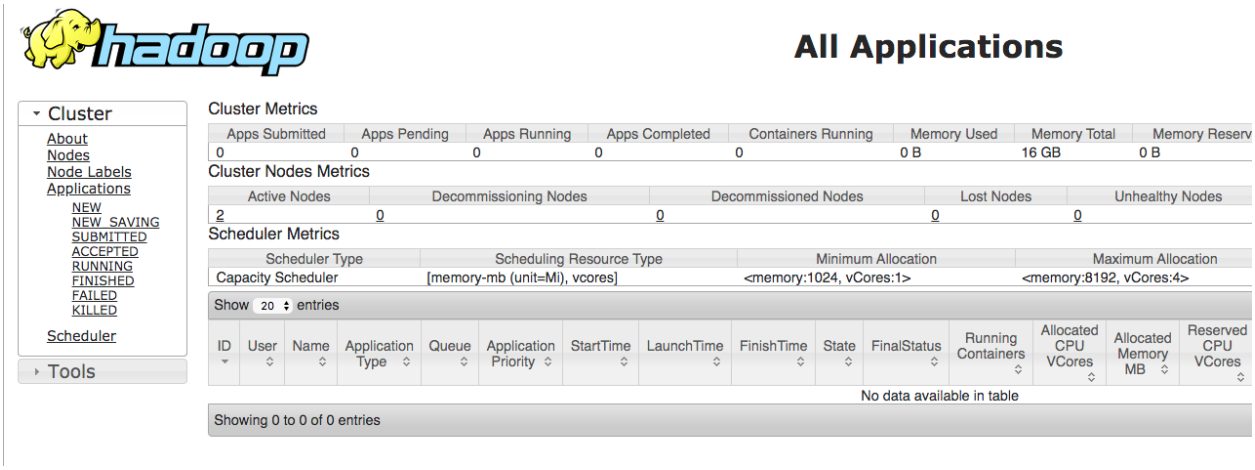
```
hadoop@jarmoune:~$ export HADOOP_HOME="/usr/local/hadoop"
hadoop@jarmoune:~$ export HADOOP_COMMON_HOME=$HADOOP_HOME
hadoop@jarmoune:~$ export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
hadoop@jarmoune:~$ export HADOOP_HDFS_HOME=$HADOOP_HOME
hadoop@jarmoune:~$ export HADOOP_MAPRED_HOME=$HADOOP_HOME
hadoop@jarmoune:~$ export HADOOP_YARN_HOME=$HADOOP_HOME
hadoop@jarmoune:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hadoop@jarmoune:~$ yarn node -list
2019-11-19 14:59:09,727 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0
Total Nodes:2
         Node-Id             Node-State Node-Http-Address       Number-of-Running-Co
 jarmoune2:43777                RUNNING     jarmoune2:8042
 jarmoune1:34825                RUNNING     jarmoune1:8042
```

```
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>

<!-- Site specific YARN configuration properties -->
        <property>
                <name>yarn.resourcemanager.hostname</name>
                <value>jarmoune</value>
        </property>
</configuration>
```

to **/usr/local/hadoop/etc/hadoop/yarn-site.xml**:

To start yarn, we run this command:


Nodes of the cluster
Cluster
About
Nodes
Node Labels
Applications
NEW
NEW_SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED
Scheduler
Tools
Cluster Metrics
Apps Submitted | Apps Pending | Apps Running | Apps Completed | Containers Running | Memory Used | Memory Total | Memory Reserved | VCores Used | VCores Total | VCores Reserved
0 | 0 | 0 | 0 | 0 | 0 B | 16 GB | 0 B | 0 | 16 | 0
Cluster Nodes Metrics
Active Nodes | Decommissioning Nodes | Decommissioned Nodes | Lost Nodes | Unhealthy Nodes | Rebooted Nodes | Shutdown Nodes
2 | 0 | 0 | 0 | 0 | 0 | 0
Scheduler Metrics
Scheduler Type | Scheduling Resource Type | Minimum Allocation | Maximum Allocation | Maximum Cluster Application Priority
Capacity Scheduler | [memory-mb (unit=Mi), vcores] | <memory:1024, vCores:1> | <memory:8192, vCores:4> | 0
Show 20 entries
Search:
Node Labels | Rack | Node State | Node Address | Node HTTP Address | Last health-update | Health-report | Containers | Allocation Tags | Mem Used | Mem Avail | VCores Used | VCores Avail | Version
/default-rack | RUNNING | jarmoune2:43777 | jarmoune2:8042 | Tue Nov 19 14:45:34 +0000 2019 | | 0 | | 0 B | 8 GB | 0 | 8 | 3.2.1
/default-rack | RUNNING | jarmoune1:34825 | jarmoune1:8042 | Tue Nov 19 15:02:47 +0000 2019 | | 0 | | 0 B | 8 GB | 0 | 8 | 3.2.1
Showing 1 to 2 of 2 entries
First Previous 1 Next Last


All Applications
Cluster
About
Nodes
Node Labels
Applications
NEW
NEW_SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED
Scheduler
Tools
Cluster Metrics
Apps Submitted | Apps Pending | Apps Running | Apps Completed | Containers Running | Memory Used | Memory Total | Memory Reserv
0 | 0 | 0 | 0 | 0 | 0 B | 16 GB | 0 B
Cluster Nodes Metrics
Active Nodes | Decommissioning Nodes | Decommissioned Nodes | Lost Nodes | Unhealthy Nodes
2 | 0 | 0 | 0 | 0
Scheduler Metrics
Scheduler Type | Scheduling Resource Type | Minimum Allocation | Maximum Allocation
Capacity Scheduler | [memory-mb (unit=Mi), vcores] | <memory:1024, vCores:1> | <memory:8192, vCores:4>
Show 20 entries
ID | User | Name | Application Type | Queue | Application Priority | StartTime | LaunchTime | FinishTime | State | FinalStatus | Running Containers | Allocated CPU VCores | Allocated Memory MB | Reserved CPU VCores
No data available in table
Showing 0 to 0 of 0 entries