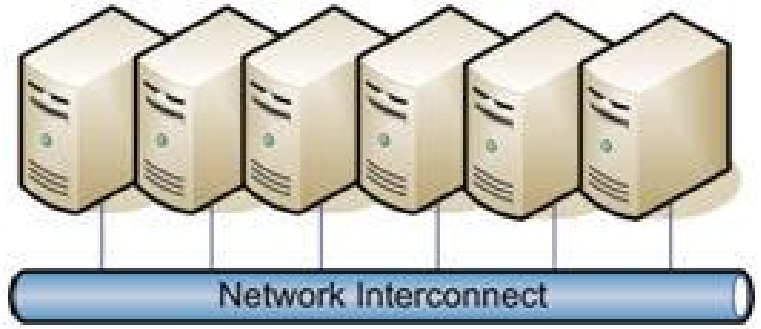
Hadoop Tutorial

Introduction to Cloud Computing, Dr. Ming Zhao

Author: Saman Biookaghazadeh

Setting up cluster

- Here we would tell you how to setting up a cluster, using virtual machines.
- In this tutorial we are going to use virtual machines, because we don't have access to real machines.
- All these instructions are applicable for a real environment, which consists of multiple machines connected together.



Setting up cluster

- In this tutorial we have assumed to have 3 slaves and one master.
- As a result we are going to have four virtual machines.
- 3 virtual machines acting as slaves.
- One virtual machine acting as master.

Setting up KVM

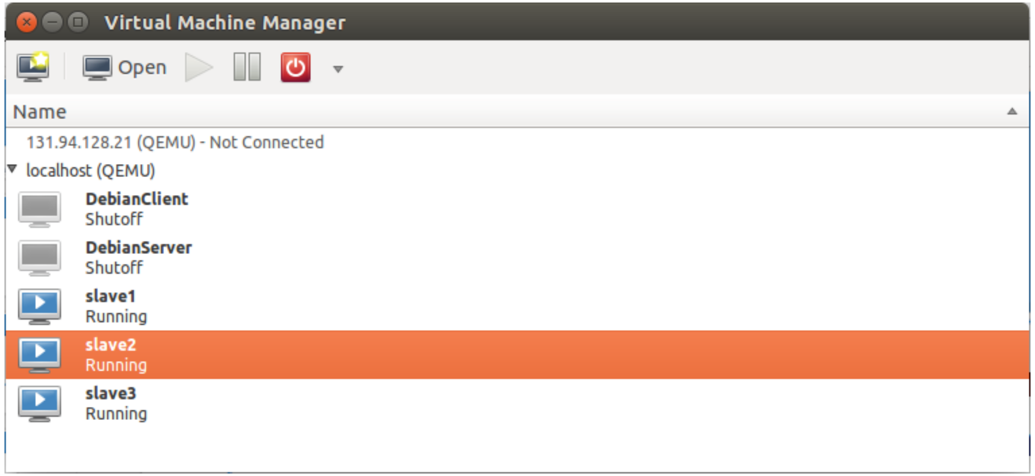
- In order to create multiple virtual machines, we need a **Hypervisor**.
- We have chosen **KVM** to run our virtual machines.
- You can Install **KVM** on a linux machine using below instructions:
    - First verify if your machine's CPU supports virtualization using this command: `egrep -c '(vmx|svm)' /proc/cpuinfo`
    - If it returned 0, it means your machine does not support virtualization, and you need to switch to another machine
    - If it returned 1 or above, then you can continue setting up KVM on this machine.

Setting up KVM

- Now It’s time to install KVM using ***apt-get*** package manager using command below:
  - `sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils`
- Here are some extra information:
  - **qemu-kvm** is the backend
  - **libvirt-bin** is responsible to administer qemu and kvm
  - **ubuntu-vm-builder** command line tool for building virtual machines
  - **bridge-utils** provides a bridge from your network to the virtual machines

Setting up KVM

- For your own convenience, its better to install virtual machine manager on your linux, using command below:
    - sudo apt-get install virt-manager
- At the end you would see an application like below:



131.94.128.21 (QEMU) - Not Connected
localhost (QEMU)
DebianClient
Shutoff
DebianServer
Shutoff
slave1
Running
slave2
Running
slave3
Running

FIU | VISA

Installing Virtual Machines

- Now it's time to install virtual machines using Virtual Machine Manager.
- In this tutorial we intend to have 3 slaves. As a result we are going to create 3 virtual machines.
- First of all download and image of **Linux Debian** from below link:
    - `http://cdimage.debian.org/debian-cd/7.8.0/amd64/iso-cd/debian-7.8.0-amd64-netinst.iso`
- We are also going to give you disk images, which everything is pre installed.

Installing Virtual Machines

- In the ***Virtual Machine Manager*** application click on the ***Create a new Virtual machine***.
- Set your virtual machine name as ***“slave1”*** in first step.
- In next step choose the downloaded debian iso image file.
- Allocate **1 GB** of memory and **1 CPU** for each virtual machine.
- Set virtual disk image size to 80 GB, and uncheck ***“Allocate entire disk now”*** option.
- Continue installing your virtual machine.

Installing other virtual machines

- Do the same instructions as previous slide to install two more virtual machines.
- The only difference is the name of the new virtual machines.
  - Choose ***slave2*** and ***slave3*** for new virtual machines(If the virtual machine is master choose name ***master*** for it).
- At the end we would have four virtual machines, which we have created using KVM.

Setting up SSH

- Hadoop master node is deploying tasks on other slave machines using ssh protocol.
- As a result, all machines should have ssh server installed on them.
- Some linux distributions by default has ssh server.
- For some other you need to install ssh server manually.
- In case your virtual machines do not have ssh server, you can use command below to install ssh on them:

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

Setting up password-less SSH

- Every time we are ssh-ing to one machine, we need to provide username and password.
- Hadoop master node needs to access slaves over SSH, without getting prompted for password.
- For instance, we need to set up password-less SSH.
- Here are the steps to provide it:
  - On the virtual machine which acts as a master, we need to generate a pair of identification keys using: `ssh-keygen -t rsa`

Setting up password-less SSH

- *Continue:*
  - After executing above command, it may asks you bunch of questions. You only need to provide the default values.
  - Now you need to create a folder called ***“.ssh”*** on slave machines using command: `ssh {username}@slave{1,2,3} mkdir -p .ssh`
  - For example you would say: `ssh saman@slave1 mkdir -p .ssh`
  - Now you need to append master’s new public key to slaves using command below:

```
cat ~/.ssh/id_rsa.pub | ssh {username}@slave{1,2,3} ‘cat >> .
ssh/authorized_keys’
```

FIU | VISA

Setting up password-less SSH

- *Continue:*
  - For example we are gonna say: `cat ~/.ssh/id_rsa.pub | ssh saman@slave1 'cat >> .ssh/authorized_keys'`
  - Try ssh to one of the slave machines.
  - This time you shouldn't be prompted for password.
  - We need to do send the public rsa to the master also, like below:

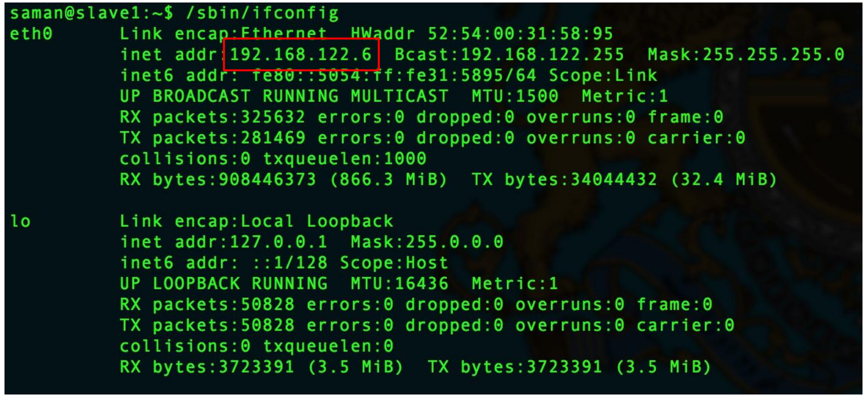
```
cat ~/.ssh/id_rsa.pub | ssh saman@master 'cat >> .
ssh/authorized_keys'
```

Setting VMs and Host Alias Names

- In order to tell the hadoop who is slave and who is master, we need to give the IP address of slaves and masters to it.
- It’s better to set an alias for each machine.
- In this example, we considered we have three slaves, named `slave1, slave2, slave3`; and we have one master (which is the workstation) called `master`.
- First of all resolve slaves and master nodes IPs, using below command:
    - `/sbin/ifconfig`

Setting VMs and Host Alias Names

- Here you would see something like below:



```
saman@slave1:~$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 52:54:00:31:58:95
          inet addr:192.168.122.6  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fe31:5895/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:325632 errors:0 dropped:0 overruns:0 frame:0
          TX packets:281469 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:908446373 (866.3 MiB)  TX bytes:34044432 (32.4 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:50828 errors:0 dropped:0 overruns:0 frame:0
          TX packets:50828 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3723391 (3.5 MiB)  TX bytes:3723391 (3.5 MiB)
```

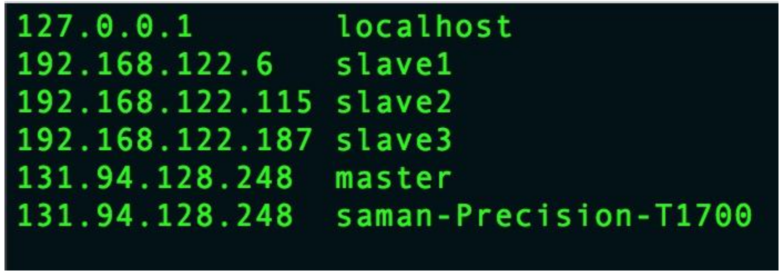
- You can see the IP address assigned to this machine(Slave or Master)

Setting VMs and Host Alias Names

- Now after realizing the master and slaves IPs, it's time set aliases.
- You need to open up a specific file using this command:

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

- Now you need to add node name aliases with their associated IP address to this line. For example we would add: " `slave1 192.168.122.6` "
- Here you can see a sample of this file:



```
127.0.0.1       localhost
192.168.122.6   slave1
192.168.122.115 slave2
192.168.122.187 slave3
131.94.128.248  master
131.94.128.248  saman-Precision-T1700
```

FIU|VISA

Setting VMs and Host Alias Names

- Now you need to do the same thing for all slaves(slave1, slave2, and slave3), and also the master node.

Downloading Hadoop

- In this tutorial we are working on hadoop version 1.2.1
- You can download hadoop-1.2.1 source code from link below:
  - http://supergsego.com/apache/hadoop/common/hadoop-1.2.1/hadoop-1.2.1.tar.gz
- Untar the downloaded file to a convenient folder(home folder is preferred), using below command:

  ```
  tar -zxvf hadoop-1.2.1.tar.gz
  ```
- Now you need to do all above, on all nodes including slaves and masters.

Before Compiling Hadoop

- There are specific packages you need to install on master and slaves virtual machines, before you can compile Hadoop.
- Here there are three packages needs to be install as below:
  a. ***autoconf-2.69***
     - First download *autoconf-2.69*
     - Execute `./configure`
     - Execute `sudo make install`

  b. ***automake-1.14***
     - First download *automake-1.14*
     - Execute `./configure`
     - Execute `sudo make install`

Before Compiling Hadoop

- *Continue:*
  a. ***libtool-2.4***
     - First download libtool-2.4
     - Execute `./configure`
     - Execute `sudo make install`

Compiling Hadoop

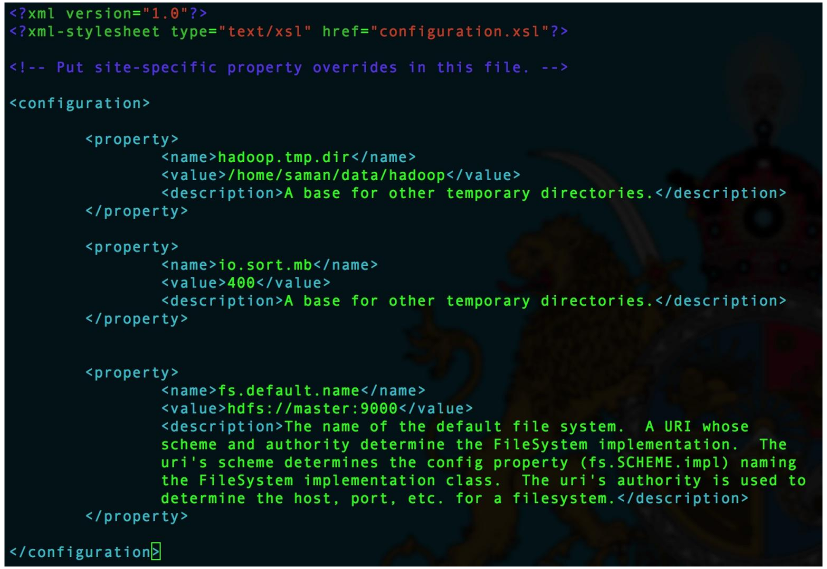
- You first need to install ***ant*** tool on your machine. Type below command to install it for each slave or master:
    - `sudo apt-get install ant`
- After installing ant, you can proceed to compile Hadoop itself.
- Go to the Hadoop folder and type `ant.`
- If everything goes right you need to see a `SUCCESSFUL` message at the end.

Configuring Hadoop

- All configurations regarding Hadoop reside in `conf` folder.
- First of all we need to tell Hadoop who is master and who are slaves. We do this as below:
  - In order to define master, open file `conf/masters` and type `master` inside it. (here we know that master points to the IP of the master node).
  - In order to define slaves, open file `conf/slaves` and type slave names(here are `slave1, slave2, slave3`) in it.

Configuring Hadoop

- now it's time to modify `core-site.xml` file. Our file after modification should be something like below:



```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

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

<configuration>

        <property>
                <name>hadoop.tmp.dir</name>
                <value>/home/saman/data/hadoop</value>
                <description>A base for other temporary directories.</description>
        </property>

        <property>
                <name>io.sort.mb</name>
                <value>400</value>
                <description>A base for other temporary directories.</description>
        </property>


        <property>
                <name>fs.default.name</name>
                <value>hdfs://master:9000</value>
                <description>The name of the default file system.  A URI whose
                scheme and authority determine the FileSystem implementation.  The
                uri's scheme determines the config property (fs.SCHEME.impl) naming
                the FileSystem implementation class.  The uri's authority is used to
                determine the host, port, etc. for a filesystem.</description>
        </property>

</configuration>
```

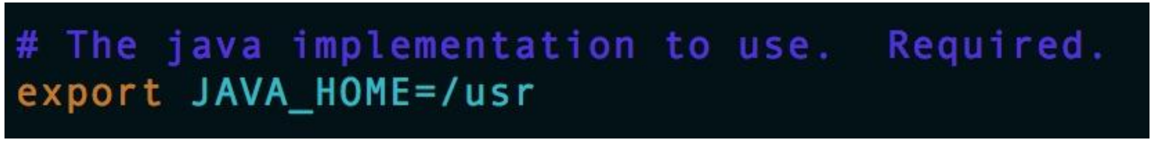
Configuring Hadoop

- `hadoop.tmp.dir` is the path for where exactly HDFS is going to store or retrieve it’s data.
- You need to create a folder (line *“/home/saman/data/hadoop”*), and consider it to be *hadoop dir*.
- `fs.default.name` is the address of the namenode(HDFS namenode). Here we ask it to run on port number 9000.

FIU | VISA

Configuring Hadoop

- Next, it's time to modify *hadoop-env.sh* file. You only need to uncomment one specific line, which tells hadoop where exactly java implementation is located. I have uncommented this line and shown below:

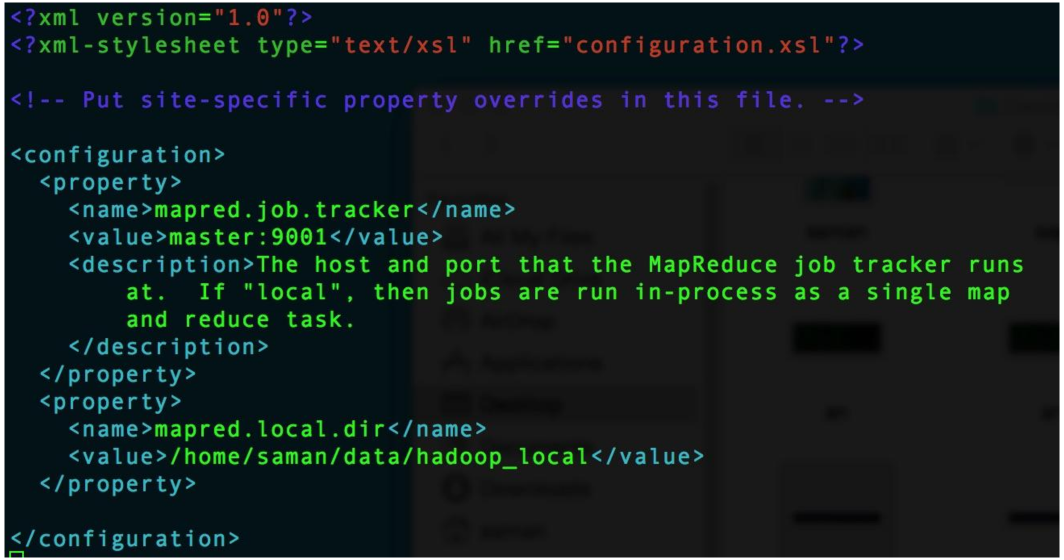


```
# The java implementation to use.  Required.
export JAVA_HOME=/usr
```

- You need to assign */usr* to *JAVA_HOME*.

Configuring Hadoop

- Now we are going to modify *mapred-site.xml* and it we be like below after modification:



```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

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

<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>master:9001</value>
    <description>The host and port that the MapReduce job tracker runs
        at.  If "local", then jobs are run in-process as a single map
        and reduce task.
    </description>
  </property>
  <property>
    <name>mapred.local.dir</name>
    <value>/home/saman/data/hadoop_local</value>
  </property>

</configuration>
```

FIU|VISA

Configuring Hadoop

- `mapred.job.tracker` defines the address that jobtracker would be binded to. For example here we have told it to be binded to 9001.
- `mapred.local.dir` defines the folder where intermediate results would be stored on. For example here we have folder called `/home/saman/data/hadoop_local` and we have assigned it to the `mapred.local.dir`.

FIU | VISA

Configuring Hadoop

- You need to do all configuration steps on all involved nodes, includes slaves and master.

Formatting Namenode

- Before being able to start hadoop, you need to tell your namenode to format the available space. In order to to do that on the master node, you need to execute below command inside the hadoop folder:

```
./bin/hadoop namenode -format
```

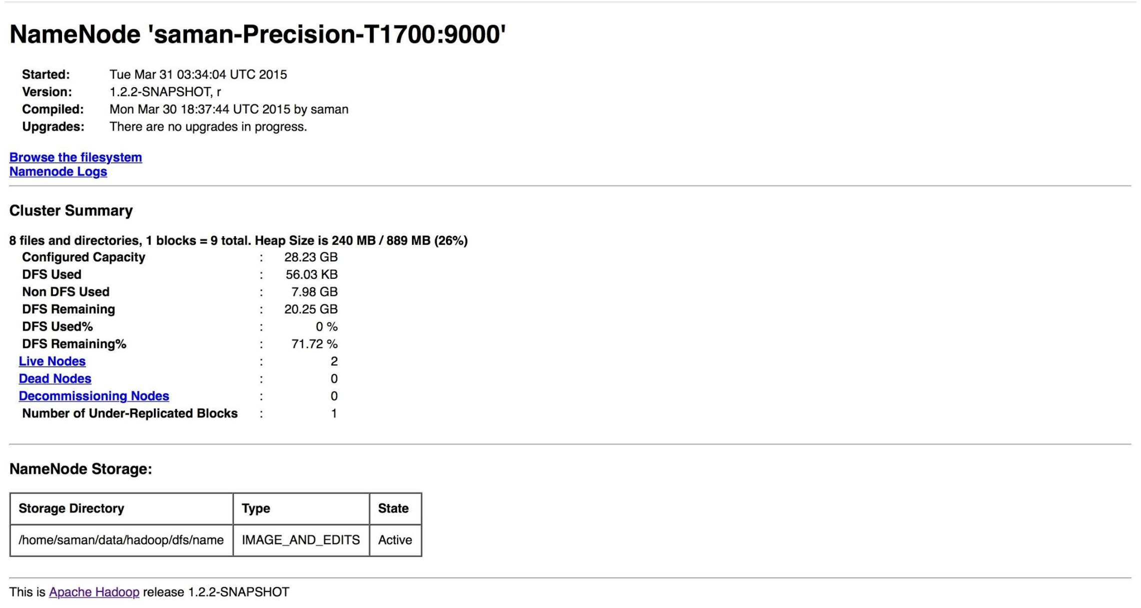
- If there is no error, it means formatting was successful.

Starting hadoop

- Now It’s time to start the Hadoop.
- In the hadoop folder execute: `/bin/start-all.sh`
- After this, you need to check logs in the `logs` folder to see if any error happened or not.

Monitoring Hadoop

- If starting hadoop is successful, we need to see hdfs status as below in address: `http://master:50070/dfshealth.jsp`



NameNode 'saman-Precision-T1700:9000'

Started:	Tue Mar 31 03:34:04 UTC 2015
Version:	1.2.2-SNAPSHOT, r
Compiled:	Mon Mar 30 18:37:44 UTC 2015 by saman
Upgrades:	There are no upgrades in progress.

Browse the filesystem
Namenode Logs

Cluster Summary

8 files and directories, 1 blocks = 9 total. Heap Size is 240 MB / 889 MB (26%)

Configured Capacity	:	28.23 GB
DFS Used	:	56.03 KB
Non DFS Used	:	7.98 GB
DFS Remaining	:	20.25 GB
DFS Used%	:	0 %
DFS Remaining%	:	71.72 %
Live Nodes	:	2
Dead Nodes	:	0
Decommissioning Nodes	:	0
Number of Under-Replicated Blocks	:	1

NameNode Storage:

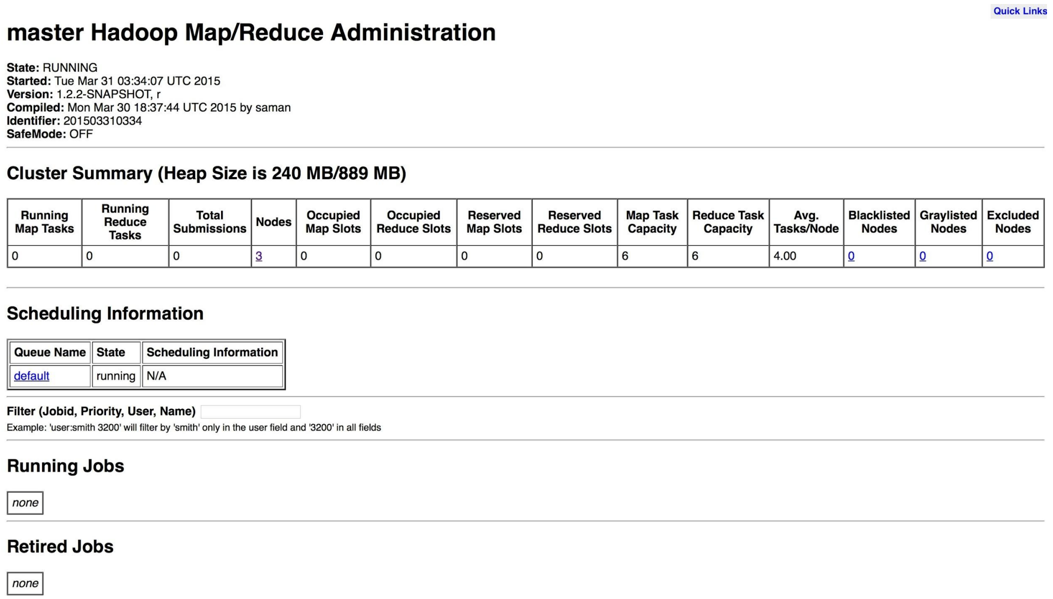
Storage Directory	Type	State
/home/saman/data/hadoop/dfs/name	IMAGE_AND_EDITS	Active

This is Apache Hadoop release 1.2.2-SNAPSHOT

FIU | VISA

Monitoring Hadoop

- We also can see the status of job tracker in address: `http://master:50030/jobtracker.jsp` as below



Quick Links

master Hadoop Map/Reduce Administration

State: RUNNING
Started: Tue Mar 31 03:34:07 UTC 2015
Version: 1.2.2-SNAPSHOT, r
Compiled: Mon Mar 30 18:37:44 UTC 2015 by saman
Identifier: 201503310334
SafeMode: OFF

Cluster Summary (Heap Size is 240 MB/889 MB)

Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes	Graylisted Nodes	Excluded Nodes
0	0	0	3	0	0	0	0	6	6	4.00	0	0	0

Scheduling Information

Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name)
Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

none

Retired Jobs

none

Monitoring Hadoop

- In the `jobtracker` web page you need to be able to see all your slaves are up.
- If there is anything wrong about it, then you need to check logs again and see what’s going on.
- You can stop your hadoop cluster using *`./bin/stop-all.`*

FIU VISA