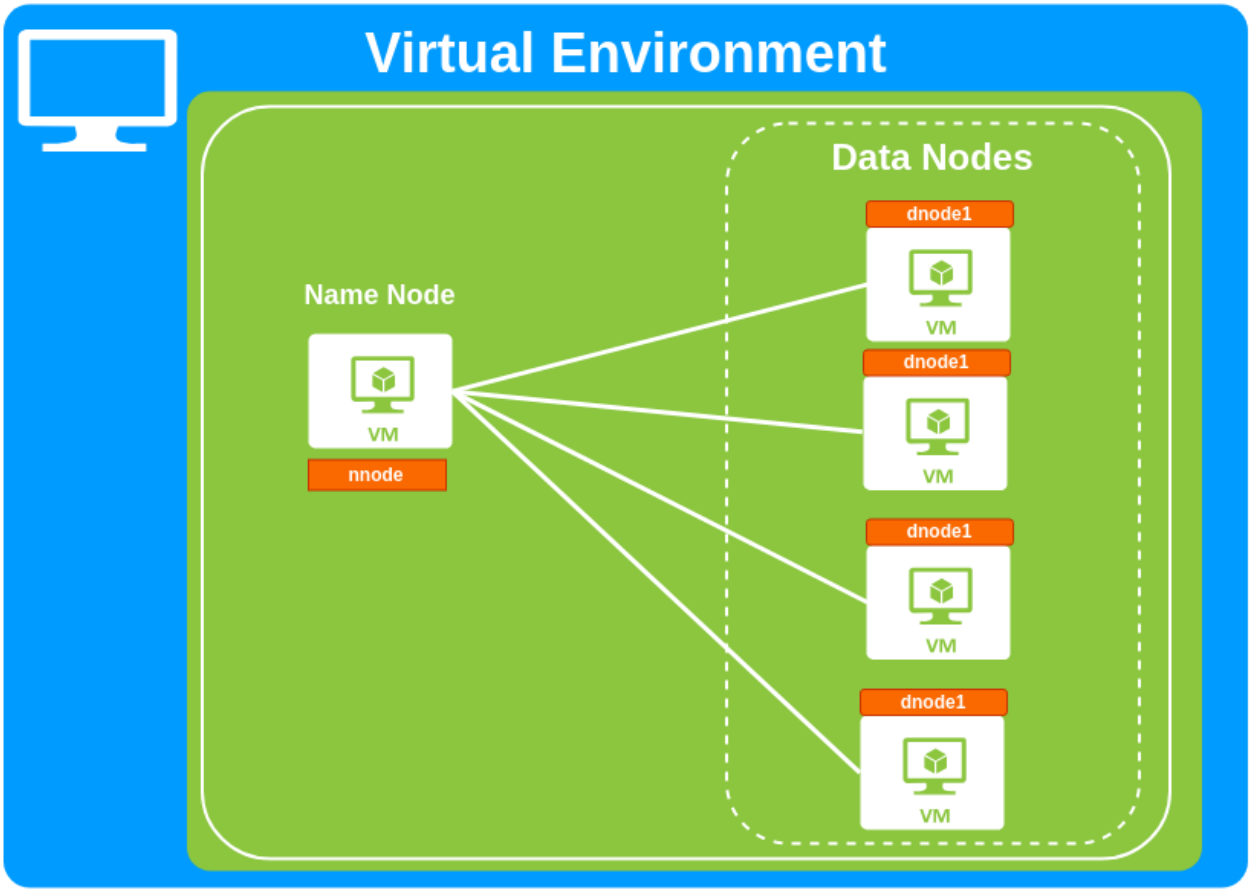
Hadoop Cluster Setup

This is what our environment should look like



Requirements

- Ubuntu Server 18.04, download from: here
- Install SSH during installation time or after (worst case)
- 5 Virtual Machines, or at least 3 of them. Details could be found in table below.

System	Hostname	IP Address	Role
VM#1	nnode.yourname.local	172.16.150.10	Name Node
VM#2	dnode1.yourname.local	172.16.150.11	Data Node
VM#3	dnode2.yourname.local	172.16.150.12	Data Node
VM#4	dnode3.yourname.local	172.16.150.13	Data Node
VM#5	dnode4.yourname.local	172.16.150.14	Data Node

The Setup

On your main Ubuntu system do the following:

1. Install first and name it “ubuntu-18.04-BASE”.
2. **REMINDER:** Install SSH during installation time or after (worst case)
3. Login and update it:
   $ sudo apt update
   $ sudo apt upgrade

4. Install Java OpenJDK 8
   $ sudo add-apt-repository ppa:openjdk-r/ppa
   $ sudo apt-get update
   $ sudo apt-get install openjdk-8-jdk

   Verify that it is working:
   $ java -version

```
user1@node:~$ java -version
openjdk version "1.8.0_191"
OpenJDK Runtime Environment (build 1.8.0_191-8u191-b12-2ubuntu0.18.04.1-b12)
OpenJDK 64-Bit Server VM (build 25.191-b12, mixed mode)
```

5. Create a directory for downloading and using for temp stuff.
   $ mkdir files

   Now download hadoop inside the new directory
   $ cd files
   $ wget -c
   http://apache.cbox.biz/hadoop/common/hadoop-3.1.2/hadoop-3.1.2.tar.gz

   Extract the files:
   $ tar xvfz hadoop-3.1.2.tar.gz

6. Create a Virtual Network using the VMWare Network Editor

7. Configure the IP Address of the BASE
   $ vim /etc/cloud/cloud.cfg

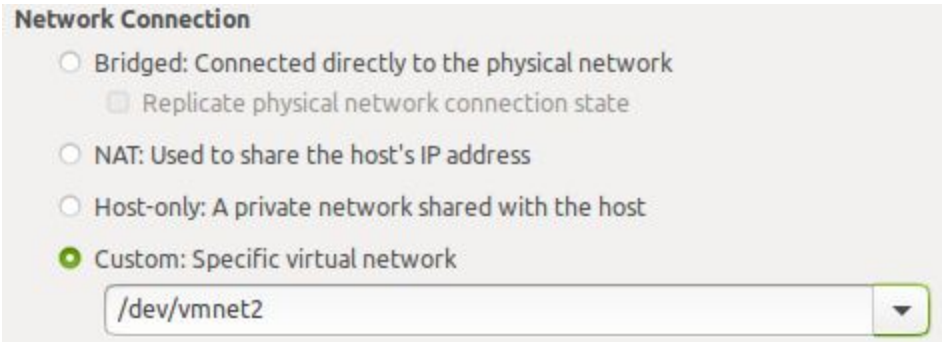
```
# This file is generated from information provided by
# the datasource.  Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
    ethernets:
        ens33:
            addresses: [172.16.150.10/24]
            gateway4: 172.16.150.1
            nameservers:
                    addresses: [172.16.150.1]
            dhcp4: no
            optional: true
    version: 2
```

   Then apply your settings:
   $ sudo netplan apply

   Verify your results:
   $ ifconfig

   If you want to speed things up, you might want to check check Note (1) in cyan below.

8. Power off the system and move on to making copies
9. Now switch the network interface of your BASE from **NAT** → **Custom** and choose vmet2 (assuming you created vmnet2, otherwise, adjust it to something else).



Creating Clones

10. Within VMWare, go to **Manage** → **Clone**
11. Then choose to create a link clone, repeat the process (4 more times). Name them as following:
    a. Ubuntu-18.04-NameNode
    b. Ubuntu-18.04-DataNode1
    c. Ubuntu-18.04-DataNode2
    d. Ubuntu-18.04-DataNode3
    e. Ubuntu-18.04-DataNode4

12. Start the first one (NameNode) and change the IP Address and hostname. It should be as seen in table #1.
    $ sudo hostnamectl set-hostname nnode.ali.local --static

    Then edit the /etc/cloud/cloud.cfg file in order to preserve the hostname across reboots and change "preserve_hostname" from **false** to true as seen below.

```
user1@nnode:~$ grep preserve /etc/cloud/cloud.cfg
preserve_hostname: true
```

    Reboot your system to double check.

Note (1): the reason why we are adding the hostnames to the hosts file of each system is we do not have a DNS Server, if you do, then no need for this step. If you did this step when installing BASE, then you will not need to repeat it.

Example of Name Node: nnode.ali.local

```
user1@nnode:~$ hostname
nnode.ali.local
user1@nnode:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.150.10  netmask 255.255.255.0  broadcast 172.16.150.255
```

Data Node1: dnode1.ali.local

```
user1@dnode1:~$ hostname
dnode1.ali.local
user1@dnode1:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.150.11  netmask 255.255.255.0  broadcast 172.16.150.255
```

Data Node2: dnode2.ali.local

```
user1@dnode2:~$ hostname
dnode2.ali.local
user1@dnode2:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.150.12  netmask 255.255.255.0  broadcast 172.16.150.255
```

/etc/hosts file

```
127.0.0.1       localhost.localdomain   localhost
::1             localhost6.localdomain6 localhost6

172.16.150.10   nnode.ali.local         nnode
172.16.150.11   dnode1.ali.local        dnode1
172.16.150.12   dnode2.ali.local        dnode2
172.16.150.13   dnode3.ali.local        dnode3
172.16.150.14   dnode4.ali.local        dnode4
```

13. Check that all devices can ping each other as seen in one example below:

```
user1@dnode1:~$ ping nnode -c1
PING nnode.ali.local (172.16.150.10) 56(84) bytes of data.
64 bytes from nnode.ali.local (172.16.150.10): icmp_seq=1 ttl=64 time=0.251 ms

--- nnode.ali.local ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.251/0.251/0.251/0.000 ms
user1@dnode1:~$ ping dnode2.ali.local -c1
PING dnode2.ali.local (172.16.150.12) 56(84) bytes of data.
64 bytes from dnode2.ali.local (172.16.150.12): icmp_seq=1 ttl=64 time=0.522 ms

--- dnode2.ali.local ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.522/0.522/0.522/0.000 ms
```

Passwordless Authentication using SSH Keys

Time to setup our boxes to connect to each other using a passwordless authentication method, which is achieved using SSH keys.

14. On the name node use the ssh-keygen to create the keys as seen below.

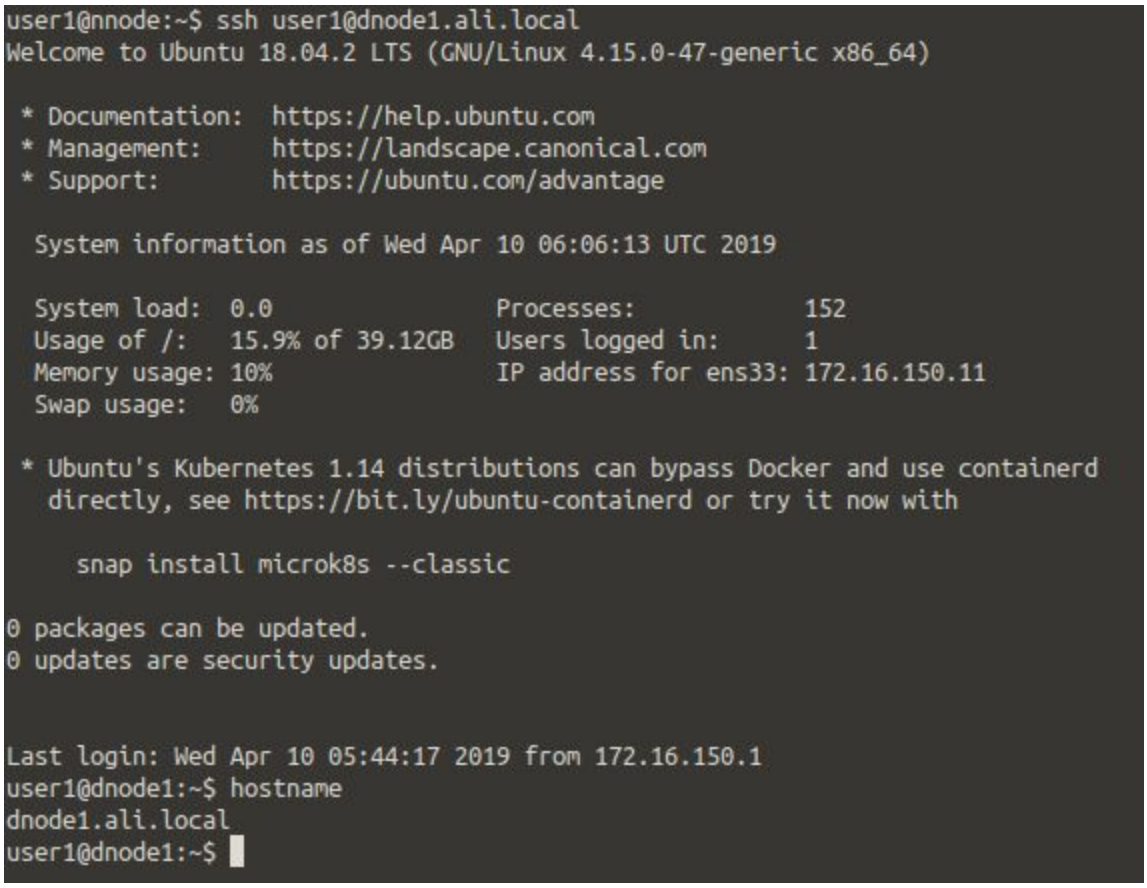
```
user1@nnode:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user1/.ssh/id_rsa):
Created directory '/home/user1/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user1/.ssh/id_rsa.
Your public key has been saved in /home/user1/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:ScmY9k6nyNYttNBpq93AxJDz3DcO4ZcANyyLeHIIpZ0 user1@nnode.ali.local
The key's randomart image is:
+---[RSA 2048]----+
|    ..   ..o     |
|  .o . =.+..     |
|  ..EoB.+oo      |
|    +.+X.= o .   |
|     +. S = =    |
|     . @ * = .   |
|      + X . .    |
|     . o +       |
|      . . .      |
+----[SHA256]-----+
user1@nnode:~$
```

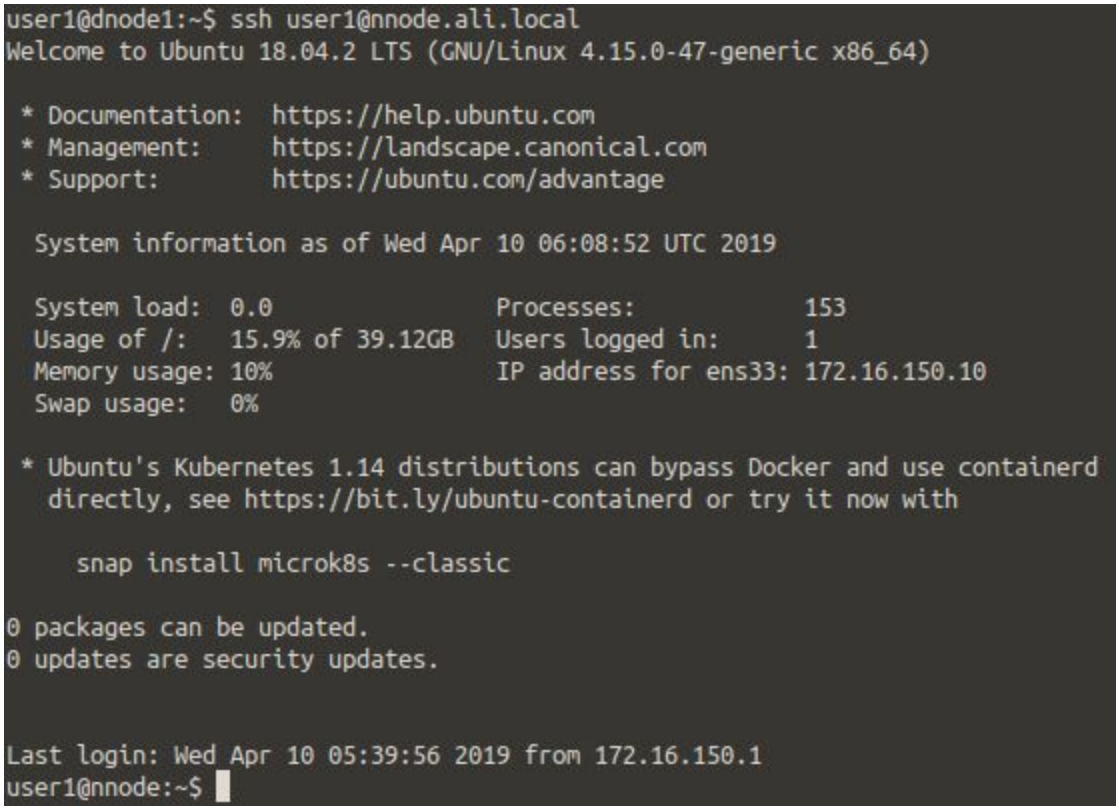
Repeat the process on all of your systems.

15. Now copy the file over to all of the systems, including the name node itself using the ssh-copy-id command.

```
user1@nnode:~$ ssh-copy-id user1@dnode1.ali.local
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/user1/.ssh/id_rsa.pub"
The authenticity of host 'dnode1.ali.local (172.16.150.11)' can't be established.
ECDSA key fingerprint is SHA256:EME3m+DVN7+kbtbLuwZ4sLly2KTcFQxyMJzTum+vAwU.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
user1@dnode1.ali.local's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'user1@dnode1.ali.local'"
and check to make sure that only the key(s) you wanted were added.

user1@nnode:~$
```

16. To test that you are able to login without password and using key based authentication, try to ssh into the system you just copied the key to. It is very important for the rest of the work to be successful that your namenode is able to ssh into the datanodes without a password. An example could be seen below.





17. Move the extracted copy of hadoop from your files directory to the main home directory. We will be using this path for hadoop on any system required: /home/user1/hadoop

18. Open your .bashrc file and add Hadoop to the running path, as seen in the example below. If you do not prefer “vim” use “nano”.
    $ vim .bashrc

```
PATH=/home/user1/hadoop/bin:/home/user1/hadoop/sbin:$PATH
export PATH
```

    Then to update your current running shell with the new value, do the following:
    $ source .bashrc

    To validate, just type hdfs with double tabs. Do you see anything?

HDFS Configuration

Now let us move on to configuring Hadoop.

19. Open the **/home/user1/hadoop/etc/hadoop/core-site.xml** file and add the lines below. Don't forget to change the name of the host to nnode.yourname.local.

```
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://nnode.ali.local:9000</value>
    </property>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>/home/user1/hadoop/tempdata</value>
    </property>
</configuration>
```

20. Create the tempdata directory inside the hadoop directory.
21. Now open the **/home/user1/hadoop/etc/hadoop/hdfs-site.xml** file and make sure you have your settings similar to the following:

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>2</value>
        <name>dfs.name.dir</name>
        <value>file:///home/user1/hadoop/hdfs/namenode</value>
        <name>dfs.data.dir</name>
        <value>file:///home/user1/hadoop/hdfs/datanode</value>
        <name>dfs.namenode.num.checkpoints.retained</name>
        <description>No. of edits/fsimages to retain</description>
        <value>10</value>
    </property>
</configuration>
```

Create both of those directories:

```
$ mkdir -p hadoop/hdfs/{namenode,datanode}
```

22. Finally, open the hadoop/etc/hadoop/workers file and make sure you add your datanodes there, like this:

```
dnode1.ali.local
dnode2.ali.local
```

23. Adding the location of JAVA to your hadoop-env.sh file, so hadoop know where to look for JAVA.This could be done by opening the **/home/user1/hadoop/etc/hadoop/hadoop-env.sh** file and adding the line under the line "**# export JAVA_HOME=**"
    export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

Before we move on to starting our hadoop cluster, we need to copy all of the files to the datanodes we have. We can simply do that using scp.\

24. Now, copy the files from your name node to the datanodes using the command below. Don't forget to adjust your command to your hostname.
    $ scp -r hadoop user1@dnode1.ali.local

    Do the same for the other datanodes.

Starting our Hadoop Cluster

Now it is time to start our hadoop cluster for the first time. But, before we do that, we need to format the cluster.

25. To format our hadoop cluster, on the namenode, do the following:
    $ hdfs namenode -format

```
user1@nnode:~$ hdfs namenode -format
2019-04-10 07:01:17,879 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = nnode.ali.local/172.16.150.10
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.1.2
```

    And:

```
2019-04-10 07:01:29,063 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at nnode.ali.local/172.16.150.10
************************************************************/
```

26. Check the hadoop version, similar to the screenshot below.
    $ hadoop version

    Or use the hdfs command instead, both should work.

27. Now to start your cluster, use the start-dfs.sh command.
28. Now run the jps command to check the status.
29. Go to your datanode (any of them) and run the jps command.

```
user1@nnode:~$ start-dfs.sh
Starting namenodes on [nnode.ali.local]
Starting datanodes
Starting secondary namenodes [nnode.ali.local]
user1@nnode:~$ jps
5650 SecondaryNameNode
5363 NameNode
5764 Jps
user1@nnode:~$
```

    On datanode1:

```
user1@dnode1:~$ jps
3081 Jps
3019 DataNode
```

30. Run a report check on the namenode using:
    $ hdfs dfsadmin -report

```
Configured Capacity: 84008173568 (78.24 GB)
Present Capacity: 64499810304 (60.07 GB)
DFS Remaining: 64499761152 (60.07 GB)
DFS Used: 49152 (48 KB)
DFS Used%: 0.00%
Replicated Blocks:
        Under replicated blocks: 0
        Blocks with corrupt replicas: 0
        Missing blocks: 0
        Missing blocks (with replication factor 1): 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0
Erasure Coded Block Groups:
        Low redundancy block groups: 0
        Block groups with corrupt internal blocks: 0
        Missing block groups: 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0

-------------------------------------------------
Live datanodes (2):

Name: 172.16.150.11:9866 (dnode1.ali.local)
Hostname: dnode1.ali.local
Decommission Status : Normal
Configured Capacity: 42004086784 (39.12 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 7590084608 (7.07 GB)
DFS Remaining: 32249876480 (30.04 GB)
DFS Used%: 0.00%
DFS Remaining%: 76.78%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Wed Apr 10 07:26:19 UTC 2019
Last Block Report: Wed Apr 10 07:22:20 UTC 2019
Num of Blocks: 0


Name: 172.16.150.12:9866 (dnode2.ali.local)
Hostname: dnode2.ali.local
Decommission Status : Normal
Configured Capacity: 42004086784 (39.12 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 7590076416 (7.07 GB)
DFS Remaining: 32249884672 (30.04 GB)
DFS Used%: 0.00%
DFS Remaining%: 76.78%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
```

31. Let us test the cluster by creating a for420 directory. This could be done using the command below.
    $ hdfs dfs -mkdir /for420

32. Let us check if the directory was created using the dfs ls command as seen below.
    $ hdfs dfs -ls /

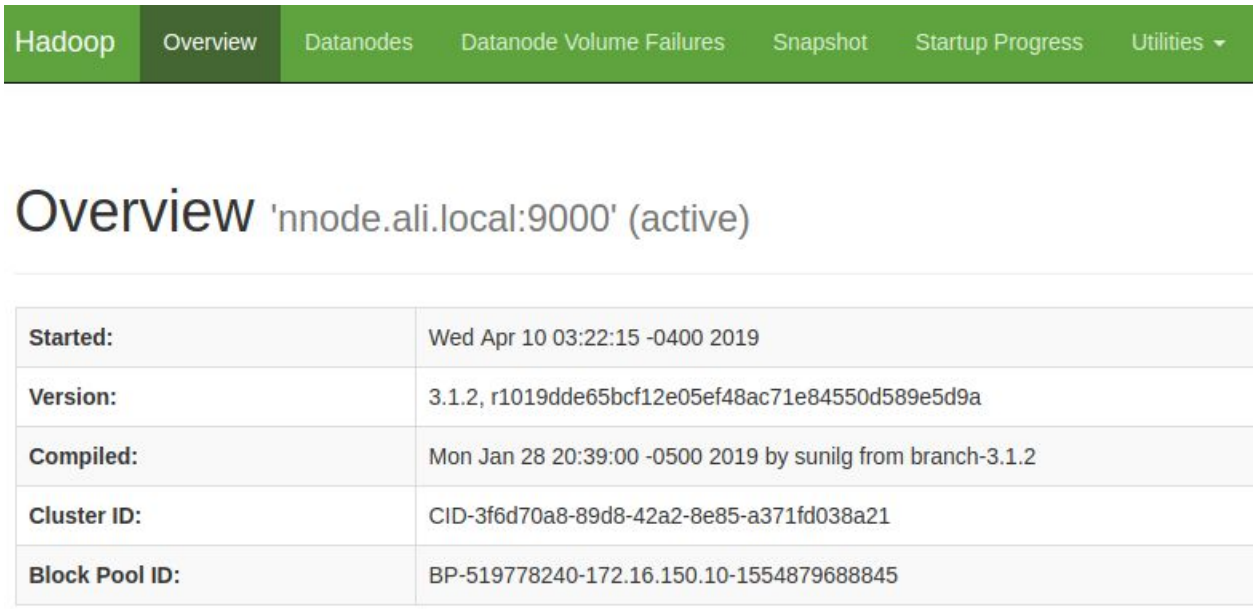
33. You will see results similar to the screenshot below.

Accessing the Web Interface

We have configured hadoop to also be accessed by the web interface. Check it out.

34. On your system with your browser, go to the IP Address (or hostname if your system is configured properly) and navigate to the following: http://172.16.150.10:9870/

35. You should receive an overview page, similar to the one below.



36. Go to the Datanodes tab and check that your datanodes are there.