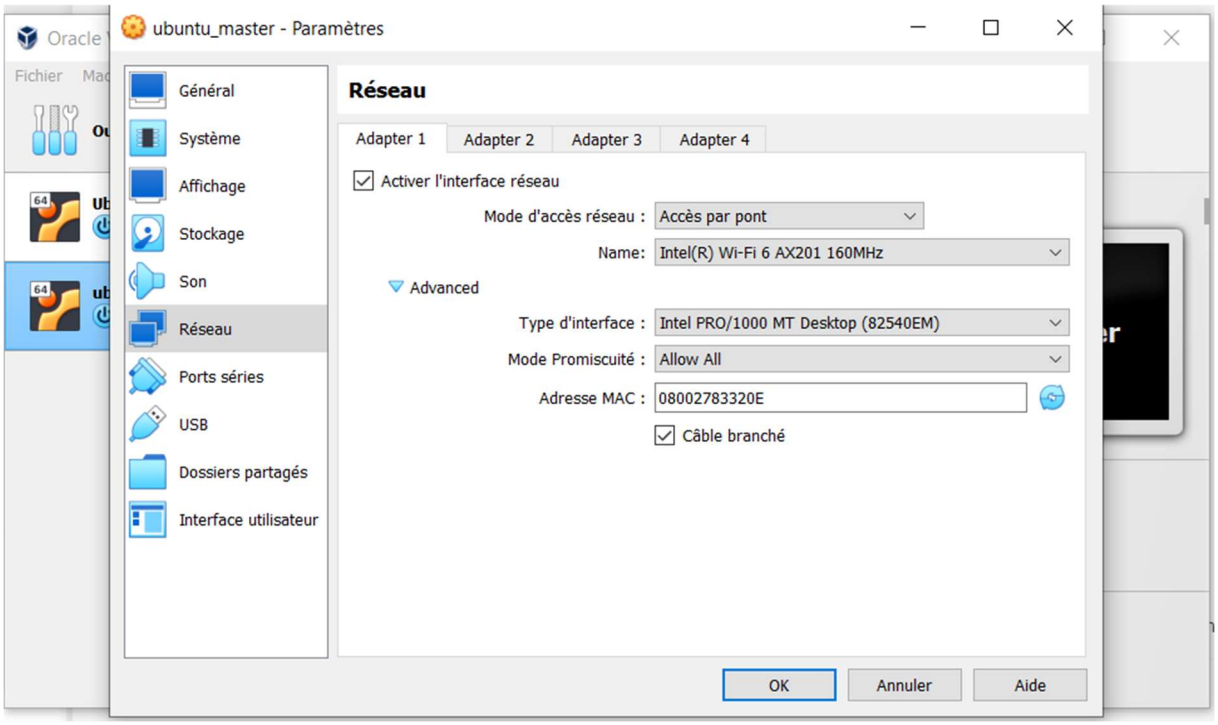
Lab 0 – Creating a Hadoop cluster with multiple VMs

In this lab, we are going to set up a hadoop cluster composed of three nodes (namenode (hadoop-master), datanode1 (hadoop-slave1) and datanode2(hadoop-slave2)) where each node is a separate VM running Ubuntu Linux. Let's start with setting up our main machine.

Note : We use VirtualBox to create our VMs

Step 1: Giving our machine access to our computer's network adapter

- Go into setting then choose Network.
- Set our Adapter to bridge and set Promiscuous mode to Allow all.



Step 2: Installing SSH

```
> sudo apt install ssh
```

Step 3: Installing Java

```
> sudo apt install openjdk-8-jdk
> java -version
```

Step 4: Download (and install) Hadoop

```
> sudo wget -P ~ https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz
> tar xzf hadoop-3.3.6.tar.gz
> mv hadoop-3.3.6 hadoop
```

Step 5: Setting up Hadoop

```
> nano ~/hadoop/etc/hadoop/hadoop-env.sh
```

Add the next line at the end of the file :

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

Step 6: Moving Hadoop to a local user directory

```
> sudo mv hadoop /usr/local/hadoop
```

Step 7: Setting up the Hadoop path and Java Home Path

```
> sudo nano /etc/environment
```

Edit the file as follows :

```
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/jre"
```

Step 8: Creating an Hadoop user

We need to create a user on our virtual machine that will later be used by the other nodes just for Hadoop.

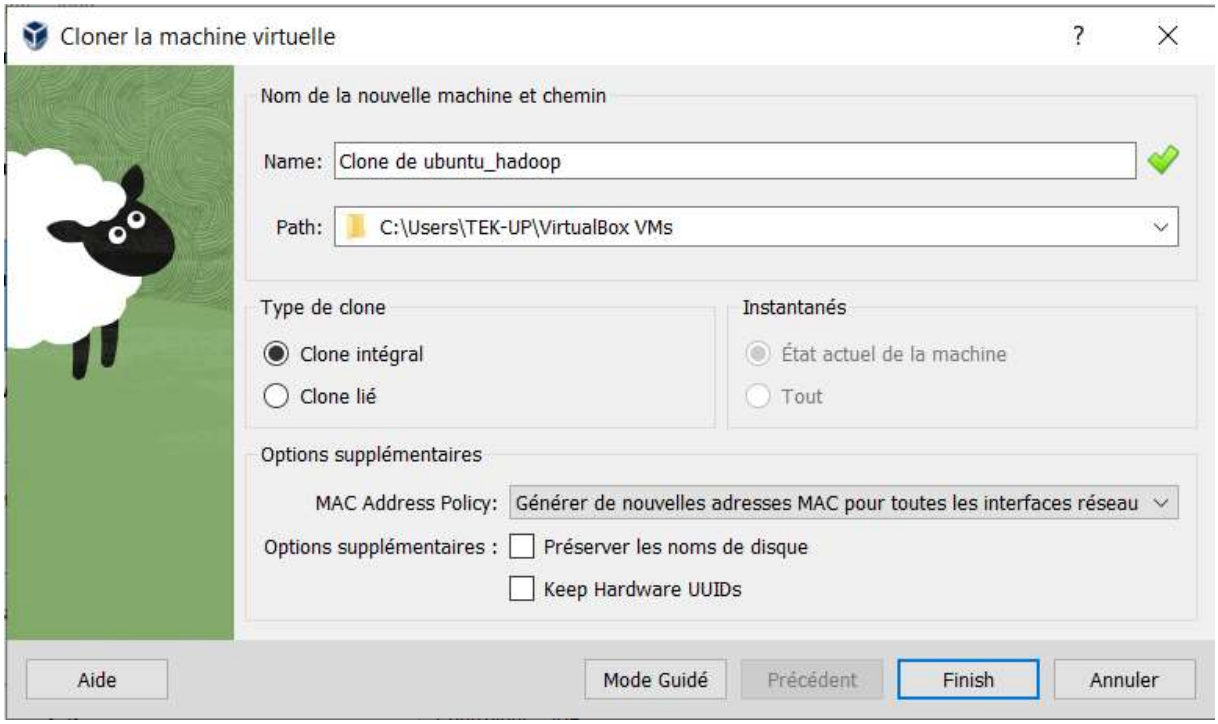
```
> sudo adduser hadoopuser
```

Now, we need to give our new user a bunch of permissions, sudo and also root access to the hadoop folder:

```
> sudo chown hadoopuser:root -R /usr/local/hadoop/
> sudo chmod g+rwx -R /usr/local/hadoop/
> sudo usermod -a -G sudo hadoopuser
```

Step 9: Cloning our machine to create datanodes

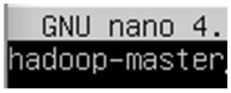
Make sure you choose Generate new MAC addresses for all network adapters.



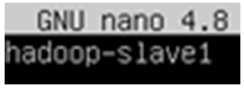
Step 10: Changing the hostname of each machine (for each machine)

```
> sudo nano /etc/hostname
```

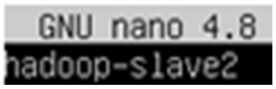
For the main machine, write in the file:



And on Slave1:



And on Slave2:



Then, reboot each machine :

```
> reboot
```

Step 11: Checking IP address of each machine and writing down the hosts

```
> ip addr
```

Write down the IP address of each machine onto the hosts file.

```
> sudo nano /etc/hosts
```

Add in the IP addresses with their corresponding hostnames.

Step 12: Configuring Hadoop ports (main machine only)

We need to configure Hadoop ports and write more configuration files.

```
> sudo nano /usr/local/hadoop/etc/hadoop/core-site.xml
```

And then, in the file, inside the configuration segment, write:

```
<property>
<name>fs.defaultFS</name>
<value>hdfs://hadoop-master:9000</value>
</property>
```

Step 13: Configuring HDFS (main machine only)

```
> sudo nano /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

Once again, in the file in the configuration segment, write:

```
<property>
<name>dfs.namenode.name.dir</name><value>/usr/local/hadoop/data/n
ameNode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name><value>/usr/local/hadoop/data/d
ataNode</value>
</property>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
```

Step 14: Labeling our slave nodes (main machine only)

Let's label our worker (slave) nodes so that Hadoop knows who to put to work. The command is:

```
> sudo nano /usr/local/hadoop/etc/hadoop/workers
```

In the file, write in the name of our slave nodes. Since we only have two, we only need to write two names.

```
hadoop-slave1
hadoop-slave2
```

Step 15: Generating a SSH key

```
> su - hadoopuser
> ssh-keygen -t rsa
```

Now, we need to copy (share) the hadoopuser SSH key in the master node to all the other nodes so that they use the same key, and this way they don't need a password

```
> ssh-copy-id hadoopuser@hadoop-slave1
> ssh-copy-id hadoopuser@hadoop-slave2
```

To make sure everything worked, let's try to access our own virtual machine through SSH :

```
> ssh hadoop-slave1
> exit
```

Step 16: Copying files from the master to the slaves

Copy the entire subdirectory into the other machines, they need to be running.

```
> scp /usr/local/hadoop/etc/hadoop/* hadoop-
slave1:/usr/local/hadoop/etc/hadoop/

> scp /usr/local/hadoop/etc/hadoop/* hadoop-
slave2:/usr/local/hadoop/etc/hadoop/
```

Step 17: Saving configurations and formating/starting HDFS (main machine only)

```
> source /etc/environment
```

Format the HDFS system with the following command:

```
> hdfs namenode -format
```

Start hdfs:

```
> start-dfs.sh
```

To make sure everything is running, write:

```
> jps
```

On the slave node machines, you can run jps as well. It'll look pretty much the same, but only jps and datanode will show up.

Step 18: Accessing the nodes through a management site tool

You can do hadoop-master:9870 on a browser on your main Ubuntu Desktop virtual machine.

Step 19: Exporting paths and files (master only)

Export all the paths and files

```
> export HADOOP_HOME= « /usr/local/hadoop »
```

```
> export HADOOP_COMMON_HOME=$HADOOP_HOME
> export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
> export HADOOP_HDFS_HOME=$HADOOP_HOME
> export HADOOP_MAPRED_HOME=$HADOOP_HOME
> export HADOOP_YARN_HOME=$HADOOP_HOME
```

Step 20 : Configuring YARN tool and files (slave only)

Let's configure the Yarn tool and files, only on the slave/secondary nodes. Be sure to do it on all of them.

```
> sudo nano /usr/local/hadoop/etc/hadoop/yarn-site.xml
```

In the file in the configuration segment, write:

```
<property>
<name>yarn.resourcemanager.hostname</name>
<value>hadoop-master</value>
</property>
```

Step 21 : Starting YARN (master only)

```
> start-yarn.sh
```

Now, if you want to access the yarn service management tool, you can do hadoop-master:8088 on a browser on your master machine.