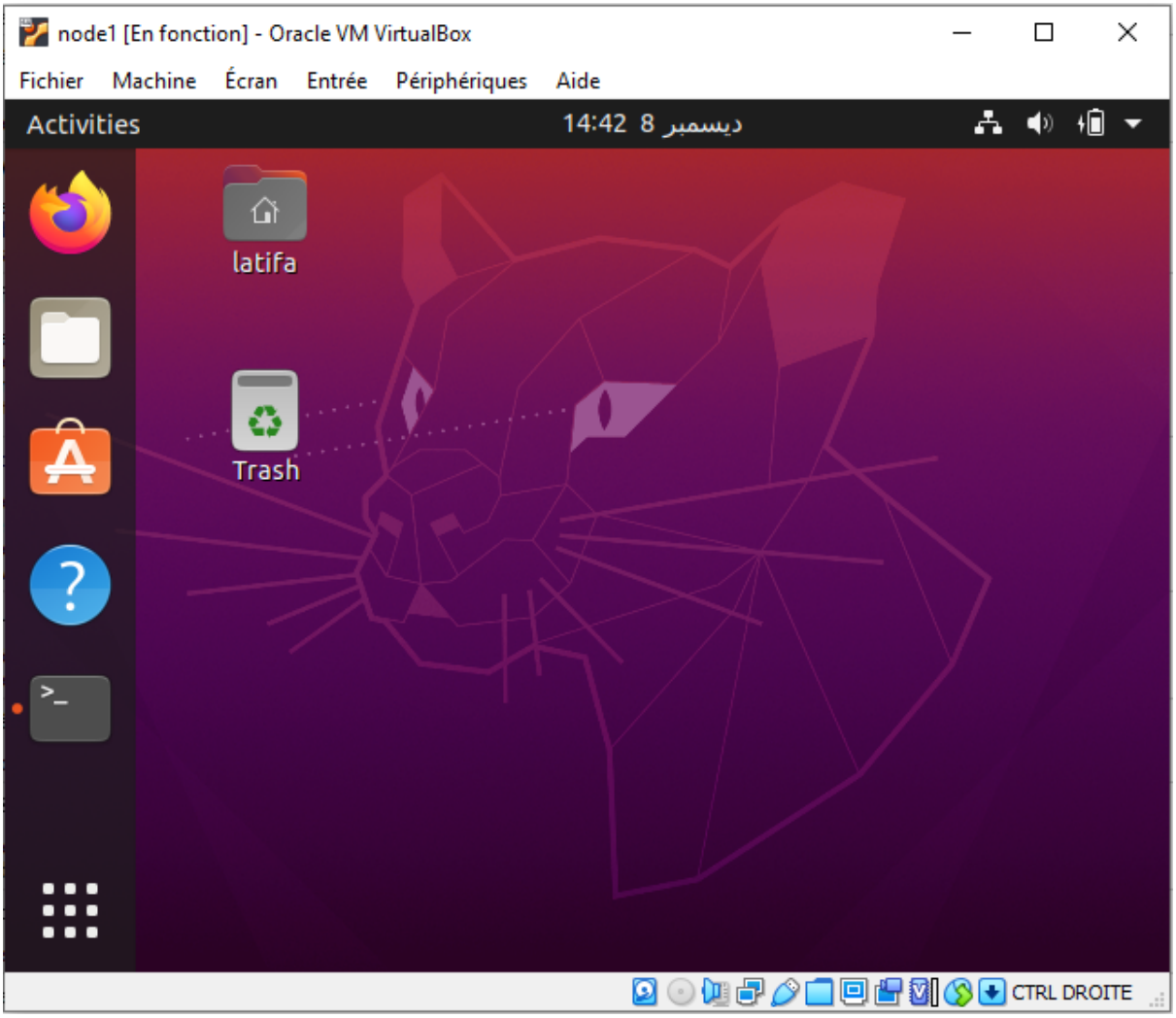
Projet Big Data

1. Création d'un cluster single node

a. Installation hadoop

La première étape pour installer hadoop est l'installation du JDk sur le système d'exploitation(dans notre cas ubuntu).



Tout d'abord il faut mettre à jour les fichiers du dépôt dans le système en tapant la commande:

```
latifa@latifa-VirtualBox:~$ sudo apt-get update
```

```
latifa@latifa-VirtualBox:~$ sudo apt-get update
[sudo] password for latifa:
Hit:1 http://tn.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://tn.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://tn.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://tn.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1,3
91 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [344 k
B]
Get:7 http://tn.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [572
```

apt-get update : commande qui met à jour la liste des fichiers disponibles dans les dépôts APT présents dans le fichier de configuration /etc/apt/sources.list.

Ensuite, on installe le java JDk, dans notre projet, nous avons installé la version par défaut avec la commande:

```
latifa@latifa-VirtualBox:~$ sudo apt-get install default-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
default-jdk is already the newest version (2:1.11-72).
0 upgraded, 0 newly installed, 0 to remove and 11 not upgraded.
```

Pour vérifier la version installée de java, on tape la commande:

```
latifa@latifa-VirtualBox:~$ java --version
openjdk 11.0.11 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.20.04, mixed mode, sha
ring)
latifa@latifa-VirtualBox:~$
```

Il faut maintenant configurer l'environnement java avec l'installation du ssh:

```
latifa@latifa-VirtualBox:~$ sudo apt-get install ssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
ssh is already the newest version (1:8.2p1-4ubuntu0.3).
```

```
latifa@latifa-VirtualBox:~$ sudo apt-get install rsync
```

rsync (pour la synchronisation à distance), est un logiciel de synchronisation de fichiers. Il est fréquemment utilisé pour mettre en place des systèmes de sauvegarde distante.
On génère par la suite la clé du cryptage avec la commande:

```
latifa@latifa-VirtualBox:~$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/latifa/.ssh/id_rsa):
/home/latifa/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Your identification has been saved in /home/latifa/.ssh/id_rsa
Your public key has been saved in /home/latifa/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:QZCgbC5yEijMPA2Y2AS/KhRjUhNfkNkFNkmRn3yUfpQ latifa@latifa-VirtualBox
The key's randomart image is:
+---[RSA 3072]----+
|+*=.oBOOo  . .   |
|O+++oo=o  o E    |
|=O=..  o.+ .     |
|+++.    +.o .    |
|+.+     S. .     |
|o=               |
|o                |
|.                |
|                 |
+----[SHA256]-----+
latifa@latifa-VirtualBox:~$
```

Il faut copier la clé par la suite dans le répertoire Authorised keys avec la commande

```
|o                |
|.                |
|                 |
+----[SHA256]-----+
latifa@latifa-VirtualBox:~$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_k
eys
latifa@latifa-VirtualBox:~$
```

Pour vérifier l'installation, on tape la commande: ssh localhost

```
latifa@latifa-VirtualBox:~$ ssh localhost
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Fri Dec  3 22:16:19 2021 from 127.0.0.1
latifa@latifa-VirtualBox:~$
```

Une fois qu'on a terminé la configuration du jdk, on passe maintenant à la désactivation du protocole IPV6, il faut donc taper la commande et ajouter les lignes ci-dessous dans le fichier sysctl.conf.

```
latifa@latifa-VirtualBox:~$ sudo gedit /etc/sysctl.conf
[sudo] password for latifa:
```

```
69 # disable ipv6
70 net.ipv6.conf.all.disable_ipv6 = 1
71 net.ipv6.conf.default.disable_ipv6 = 1
72 net.ipv6.conf.lo.disable_ipv6 = 1
73
```

Par la suite, on fait le téléchargement de la version d'Hadoop à partir du site officiel d'Apache ou simplement à partir du terminal et extraire le dossier zippé par la suite avec les commandes:

```
latifa@latifa-VirtualBox:~$ cd Downloads
latifa@latifa-VirtualBox:~/Downloads$ wget https://archive.apache.org/dist/hadoo
p/core/hadoop-2.6.0/hadoop-2.6.0.tar.gz
```

```
latifa@latifa-VirtualBox:~/Downloads$ sudo tar -xzvf hadoop-2.6.0.tar.gz
```

On déplace le dossier dans le usr/local/hadoop et on attribue les droits d'accès au user1 déjà créé avec la commande:

```
latifa@latifa-VirtualBox:~/Downloads$ sudo mv hadoop-2.6.0 /usr/local/hadoop
```

```
latifa@latifa-VirtualBox:~/Downloads$ ls /usr/local/hadoop
bin  include  libexec      logs        README.txt  share
etc  lib      LICENSE.txt  NOTICE.txt  sbin
latifa@latifa-VirtualBox:~/Downloads$
```

```
latifa@latifa-VirtualBox:~/Downloads$ sudo chown user1:hadoop -R /usr/local/hado
op
latifa@latifa-VirtualBox:~/Downloads$
```

On crée par la suite les répertoires pour les datanode et le namenode et on applique les droits accès à ces deux répertoires aux user1.

```
user1@latifa-VirtualBox:~$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
```

```
user1@latifa-VirtualBox:~$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
```

```
user1@latifa-VirtualBox:~$ sudo chown user1:hadoop -R /usr/local/hadoop_tmp/
user1@latifa-VirtualBox:~$
```

b. Configuration Hadoop single node

Tout d'abord, il faut mettre à jour le fichier **bashrc avec la commande:**

```
user1@latifa-VirtualBox:~$ sudo gedit .bashrc
```

Et on ajoute les lignes au bas du fichier tout en spécifiant l'emplacement du JDK dans l'alias **JAVA_HOME** et le chemin d'hadoop dans l'alias **HADOOP_HOME**

```
fi
    # -- HADOOP ENVIRONMENT VARIABLES START -- #
    export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
    export HADOOP_HOME=/usr/local/hadoop
    export PATH=$PATH:$HADOOP_HOME/bin
    export PATH=$PATH:$HADOOP_HOME/sbin
    export HADOOP_MAPRED_HOME=$HADOOP_HOME
    export HADOOP_COMMON_HOME=$HADOOP_HOME
    export HADOOP_HDFS_HOME=$HADOOP_HOME
    export YARN_HOME=$HADOOP_HOME
    export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
    export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
    # -- HADOOP ENVIRONMENT VARIABLES END -- #
```

Pour que les alias puissent entrer en vigueur**, il faut redémarrer le terminal** ou taper la commande **source .bashrc.**
On peut maintenant faire la configuration de Hadoop en mode pseudo-distribué,il faut modifier 5 fichiers qui sont dans **/usr/local/hadoop/etc/hadoop/**

- **Mettre à jour le fichier env.sh**

Comme Hadoop est développé en Java, on doit lui spécifier le chemin de Jdk pour qu'il puisse activer ses démons.

```
user1@latifa-VirtualBox:~$ cd /usr/local/hadoop/etc/hadoop
user1@latifa-VirtualBox:/usr/local/hadoop/etc/hadoop$ sudo gedit hadoop-env.sh
[sudo] password for user1:
```

Il faut modifier la variable **JAVA_HOME** comme suit

```
100
101 export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
```

.

- **Mettre à jour le fichier core-site.xml**

Le fichier core-site.xml informe les démons de hadoop qu'un namenode s'exécute sur le cluster en mentionnant son adresse.
Puisqu'on a une seule machine dans le cluster, le namenode va être sur le localhost (127.0.0.1).Le port 9000 est associé au système de fichier HDFS.

```
user1@latifa-VirtualBox:/usr/local/hadoop/etc/hadoop$ sudo gedit core-site.xml
```

Et on ajoute dans la balise configuration:

```
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20 <property>
21 <name>fs.default.name</name>
22 <value>hdfs://localhost:9000</value>
23 </property>
24 </configuration>
```

- **Mettre à jour le fichier hdfs-site.xml**

le fichier hdfs-site.xml informe hadoop et son système hdfs du nombre de réplications (property 1),l'adresse de l'historique des transactions du NameNode (property 2) et l'adresse du stockage des blocs par les DataNode (property 3).
La réplication doit être 1 dans notre cas car on travaille sur un cluster à un seul noeud
On tape alors la commande et on ajoute les 3 propriétés dans la balise de configuration.

```
user1@latifa-VirtualBox:/usr/local/hadoop/etc/hadoop$ sudo gedit hdfs-site.xml
```

```
18
19 <configuration>
20 <property>
21 <name>dfs.replication</name>
22 <value>1</value>
23 </property>
24
25 <property>
26 <name>dfs.namenode.name.dir</name>
27 <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
28 </property>
29
30 <property>
31 <name>dfs.datanode.data.dir</name>
32 <value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
33 </property>
34 </configuration>
                                                    XML ▾  Ta
```

- **Mettre à jour le fichier mapred-site.xml**

Ce fichier mapred-site.xml informe le package MapReduce qu'il va s'exécuter en tant qu'une application yarn (séparation entre la gestion des ressources et la gestion des

traitements). Avant de le modifier, il faut d'abord faire une copie du fichier mapred-site.xml.template sous le nom mapred-site.xml avec la commande :

```
user1@latifa-VirtualBox:/usr/local/hadoop/etc/hadoop$ sudo cp mapred-site.xml.te
mplate mapred-site.xml
```

```
user1@latifa-VirtualBox:/usr/local/hadoop/etc/hadoop$ sudo gedit mapred-site.xml
```

Dans la balise configuration, on ajoute les lignes suivantes :

```
18
19 <configuration>
20 <property>
21 <name>mapreduce.framework.name</name>
22 <value>yarn</value>
23 </property>
24
25 </configuration>
```

- **Mettre à jour le fichier yarn-site.xml**

Le fichier yarn-site.xml indique au node manager qu'il aura un service auxiliaire indiquant au MapReduce comment faire son shuffling.

```
user1@latifa-VirtualBox:/usr/local/hadoop/etc/hadoop$ sudo gedit yarn-site.xml
```

```
15 <configuration>
16 <property>
17 <name>yarn.nodemanager.aux-services</name>
18 <value>mapreduce_shuffle</value>
19 </property>
20
21 <property>
22 <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
23 <value>org.apache.hadoop.mapred.ShuffleHandler</value>
24 </property>
25 <!-- Site specific YARN configuration properties -->
26
27 </configuration>
```

c. Vérifier l'installation

Une fois qu'on a terminé la configuration de hadoop, on va vérifier son installation. Pour ce faire, il faut tout d'abord formater le NameNode avant de démarrer Hadoop et ça se fait à chaque fois qu'on va démarrer les services de Hadoop.

```
user1@latifa-VirtualBox:/usr/local/hadoop/etc/hadoop$ hdfs namenode -format
21/12/08 17:22:32 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = latifa-VirtualBox/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.6.0
STARTUP_MSG:   classpath = /usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/
hadoop/common/lib/curator-client-2.6.0.jar:/usr/local/hadoop/share/hadoop/common
/lib/jackson-jaxrs-1.9.13.jar:/usr/local/hadoop/share/hadoop/common/lib/log4j-1.
2.17.jar:/usr/local/hadoop/share/hadoop/common/lib/jsch-0.1.42.jar:/usr/local/ha
doop/share/hadoop/common/lib/jasper-runtime-5.5.23.jar:/usr/local/hadoop/share/h
adoop/common/lib/jersey-core-1.9.jar:/usr/local/hadoop/share/hadoop/common/lib/j
sp-api-2.1.jar:/usr/local/hadoop/share/hadoop/common/lib/slf4j-api-1.7.5.jar:/us
r/local/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar:/usr/local/hadoop/sha
re/hadoop/common/lib/gson-2.2.4.jar:/usr/local/hadoop/share/hadoop/common/lib/ht
tpcore-4.2.5.jar:/usr/local/hadoop/share/hadoop/common/lib/apacheds-kerberos-cod
ec-2.0.0-M15.jar:/usr/local/hadoop/share/hadoop/common/lib/avro-1.7.4.jar:/usr/l
ocal/hadoop/share/hadoop/common/lib/protobuf-java-2.5.0.jar:/usr/local/hadoop/sh
```

d. Démarrer le processus Hadoop

```
user1@latifa-VirtualBox:/usr/local/hadoop/etc/hadoop$ start-dfs.sh
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.
util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.
6.0.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.
security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflect
```

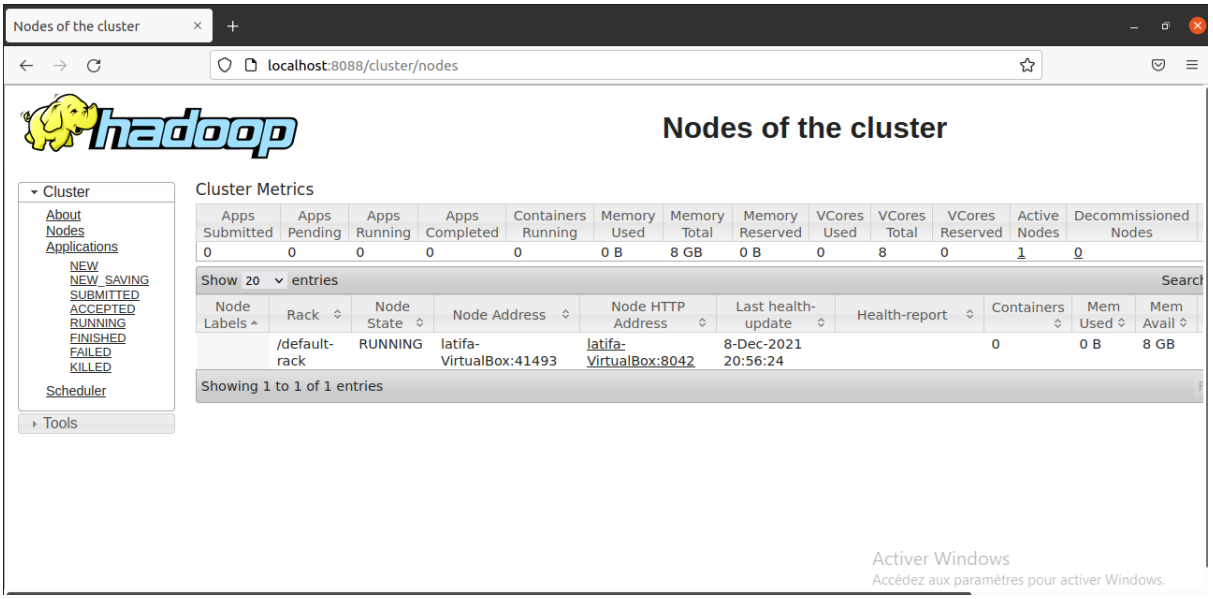
```
user1@latifa-VirtualBox:/usr/local/hadoop/etc/hadoop$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-user1-resourcem
anager-latifa-VirtualBox.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-user1-no
demanager-latifa-VirtualBox.out
user1@latifa-VirtualBox:/usr/local/hadoop/etc/hadoop$
```

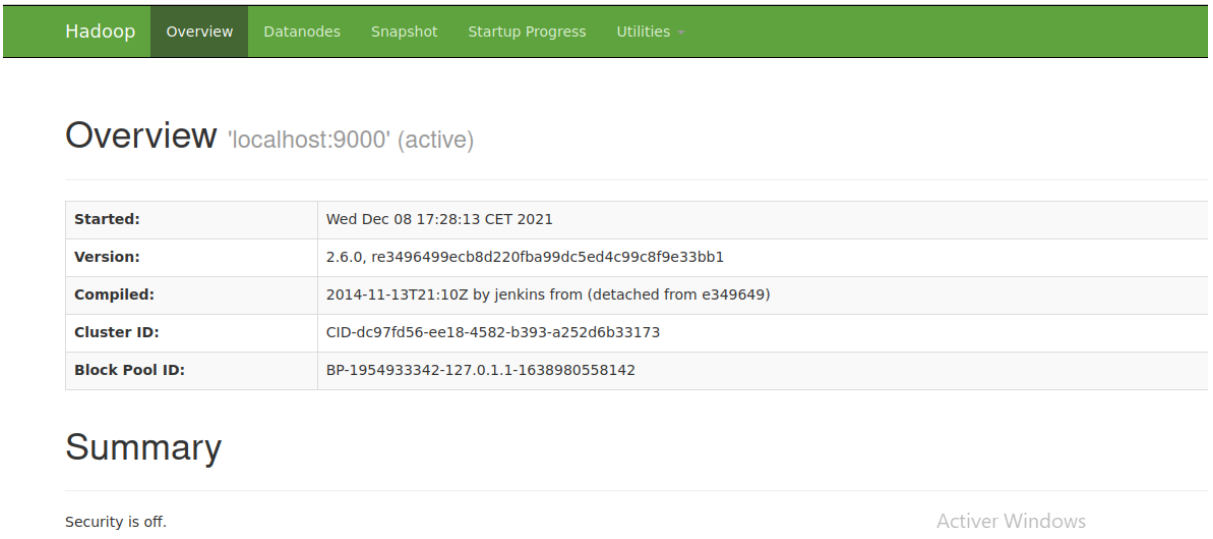
e. Vérifier le service après le démarrage

```
user1@latifa-VirtualBox:/usr/local/hadoop/etc/hadoop$ jps
11649 ResourceManager
11080 NameNode
12073 Jps
11454 SecondaryNameNode
11775 NodeManager
user1@latifa-VirtualBox:/usr/local/hadoop/etc/hadoop$
```

f. Visualisation de l'interface graphique du NameNode

Quand on se connecte sur **http//:localhost:8080** une interface se présente indiquant un seul noeuds actif.





2. Création d'un cluster multi-nodes

On a choisi de travailler avec un cluster à 3 nœuds: Un NameNode et deux DataNodes. Par conséquent, on doit modifier la configuration précédente pour permettre d'exécuter les *jobs* de façon distribuée.

La première chose à faire est de dupliquer la machine qu'on a déjà créé en mode pseudo-distribué et mettre la configuration réseau en mode accès par pont afin d'avoir des adresses IP différentes dans les trois machines.

Après la duplication,il faut obtenir les adresses IP des trois machines virtuelles.

```
latifa@NameNode:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:1d:49:1c brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.131/24 brd 192.168.100.255 scope global dynamic noprefixrou
te enp0s3
       valid_lft 85758sec preferred_lft 85758sec
    inet6 fe80::32c9:c6c8:4478:f686/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

```
latifa@DataNode1:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:12:1e:51 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.168/24 brd 192.168.100.255 scope global dynamic noprefixrou
te enp0s3
       valid_lft 85659sec preferred_lft 85659sec
    inet6 fe80::f0b6:19f3:dd7b:3d6f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

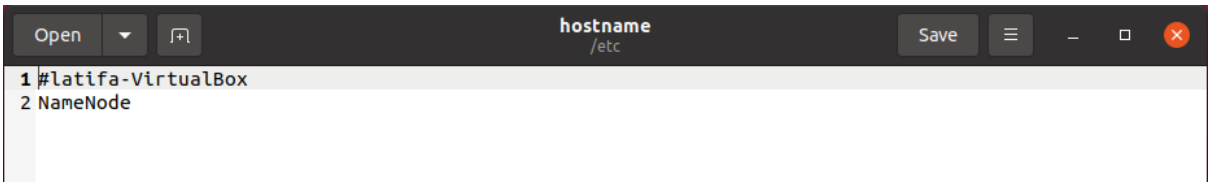
```
latifa@DataNode2:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:41:86:4b brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.183/24 brd 192.168.100.255 scope global dynamic noprefixrou
te enp0s3
       valid_lft 85631sec preferred_lft 85631sec
    inet6 fe80::bfe4:62c0:61a:73e8/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

a. Configuration des alias réseaux:

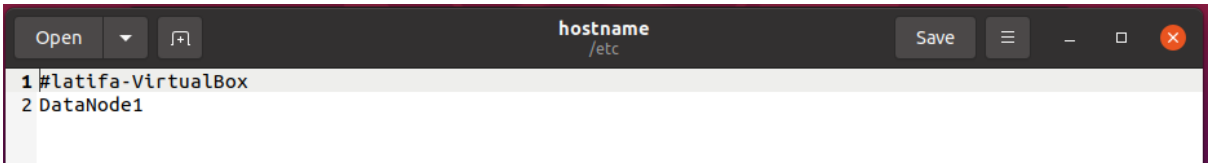
On édite le fichier /etc/hostname en tant qu'utilisateur root dans les trois Vms.

```
latifa@DataNode1:~$ su
Password:
root@DataNode1:/home/latifa# gedit /etc/hostname
```

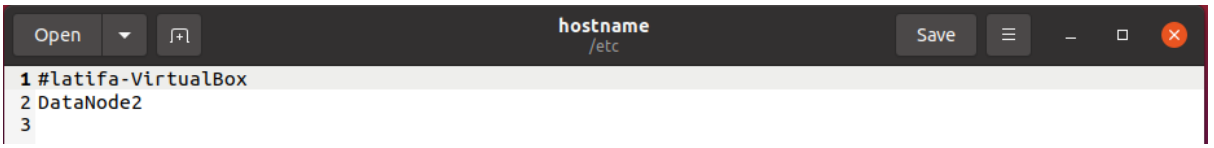
- **Dans le nameNode:**



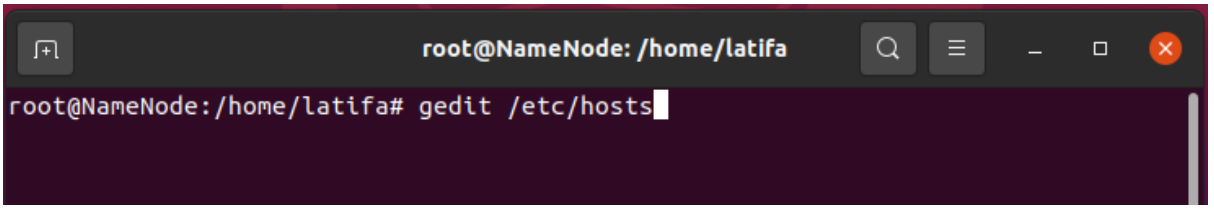
- **Dans le DataNode 1**

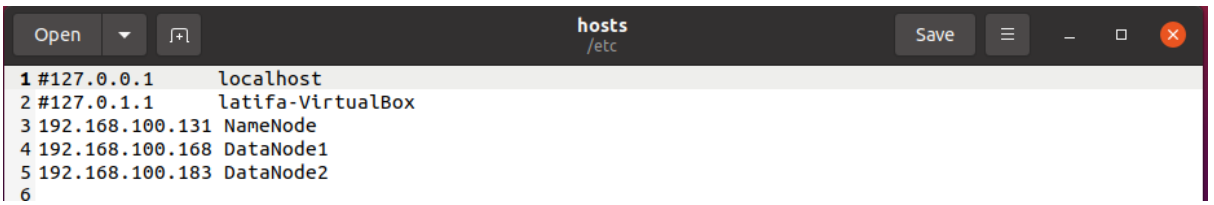


- **Dans le DataNode 2**



On édite aussi le fichier /etc/hosts dans les 3 machines virtuelles comme suit:





Après cette configuration, il faut redémarrer les VMs avec la commande reboot.

b. Configuration des fichiers Xml:

Ces configurations sont communes aux trois nœuds, on doit donc se connecter en tant qu'utilisateur user1 déjà créé.

- **Configuration du fichier core-site.xml**:

```
user1@NameNode:~$ su user1
Password:
user1@NameNode:~$ cd /usr/local/hadoop/etc/hadoop
user1@NameNode:/usr/local/hadoop/etc/hadoop$ sudo gedit core-site.xml
```

On remplace la valeur de la propriété fs.default.name hdfs://localhost:9000 par hdfs://NameNode:9000 puisque l'alias réseau a été modifié dans le fichier /etc/hosts.

```
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20 <property>
21 <name>fs.default.name</name>
22 <value>hdfs://NameNode:9000</value>
23 </property>
24 </configuration>
```

XML Tab Width: 8 Ln 21, Col 29 INS

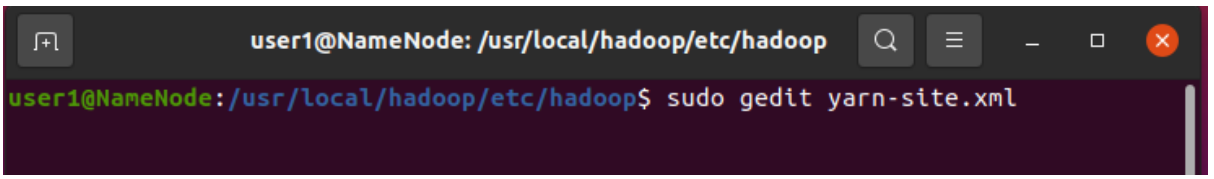
- **Configuration du fichier hdfs-site.xml**:

```
(gedit:2270): Tepl-WARNING **: 00:18:11.342: GVfs metadata is not supported. Fal
lback to TeplMetadataManager. Either GVfs is not correctly installed or GVfs met
adata are not supported on this platform. In the latter case, you should configu
re Tepl with --disable-gvfs-metadata.
user1@NameNode:/usr/local/hadoop/etc/hadoop$ sudo gedit hdfs-site.xml
```

On change le facteur de réplication à 2 dans la propriété dfs.replication.

```
19 <configuration>
20 <property>
21 <name>dfs.replication</name>
22 <value>2</value>
23 </property>
24
25 <property>
26 <name>dfs.namenode.name.dir</name>
27 <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
28 </property>
29
30 <property>
31 <name>dfs.datanode.data.dir</name>
32 <value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
33 </property>
34 </configuration>
```

- **Configuration du fichier yarn-site.xml**:



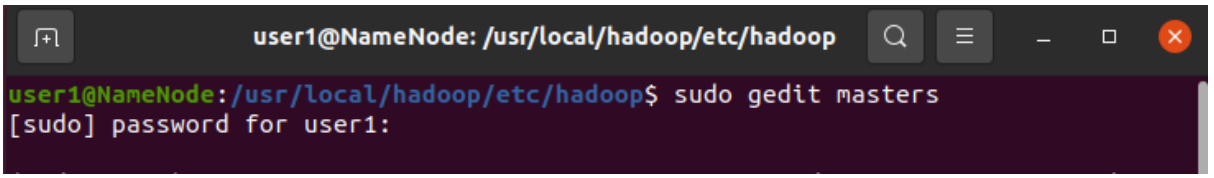

```
14 -->
15 <configuration>
16 <property>
17 <name>yarn.nodemanager.aux-services</name>
18 <value>mapreduce_shuffle</value>
19 </property>
20 <property>
21 <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
22 <value>org.apache.hadoop.mapred.ShuffleHandler</value>
23 </property>
24 <property>
25 <name>yarn.resourcemanager.resource-tracker.address</name>
26 <value>NameNode:8025</value>
27 </property>
28
29 <property>
30 <name>yarn.resourcemanager.scheduler.address</name>
31 <value>NameNode:8035</value>
32 </property>
33
34 <property>
35 <name>yarn.resourcemanager.address</name>
36 <value>NameNode:8050</value>
37 </property>
38
39 </configuration>
```

- **Configuration du fichier mapred-site.xml**:

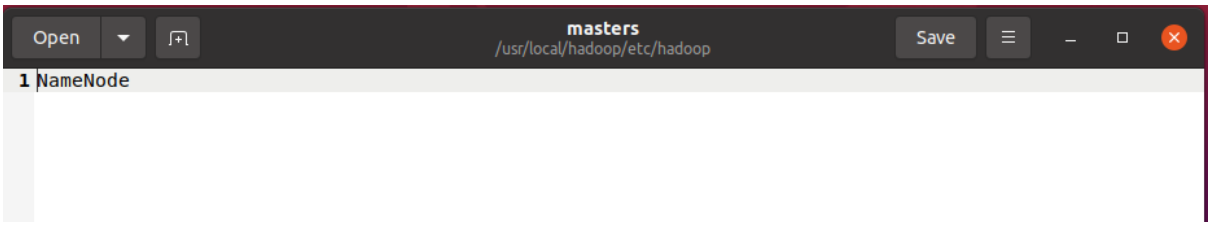
On modifie la valeur du paramètre `mapreduce.framework.name` en la passant à `yarn`.

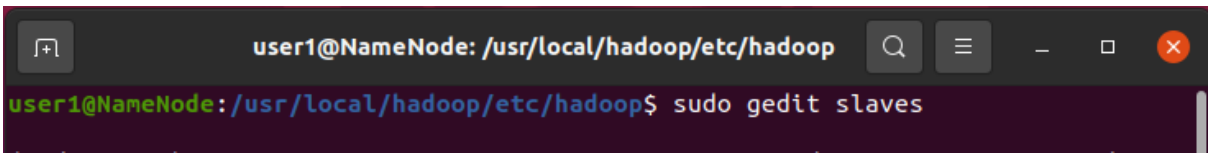
```
18
19 <configuration>
20 <property>
21 <name>mapreduce.job.tracker</name>
22 <value>NameNode:5431</value>
23 </property>
24 <property>
25 <name>mapreduce.framework.name</name>
26 <value>yarn</value>
27 </property>
28
29 </configuration>
```

c. Configuration des fichiers masters et slaves:

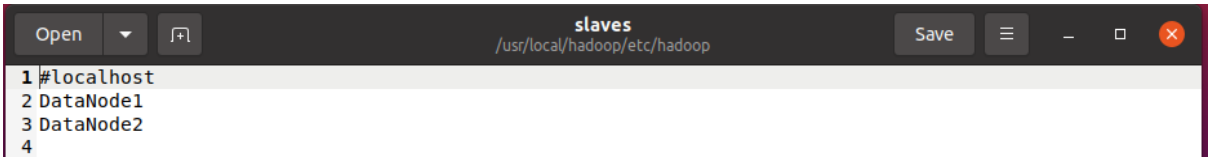


On spécifie dans le fichier le nom du master.



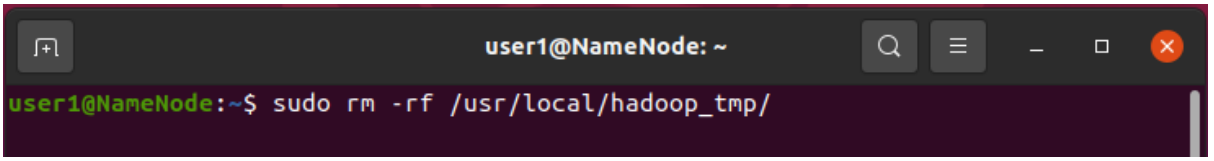


On spécifie dans le fichier le nom des workers.

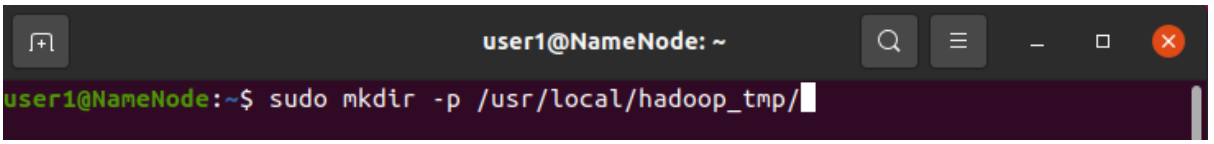


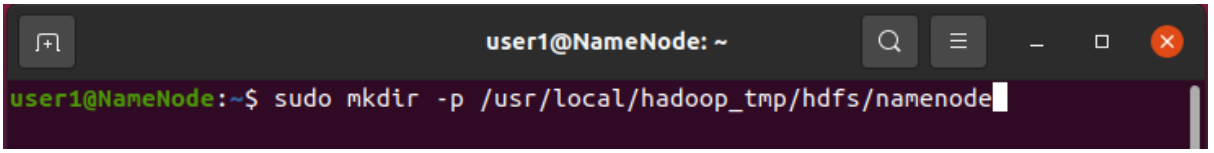
d. Formatage du système HDFS dans le NameNode:

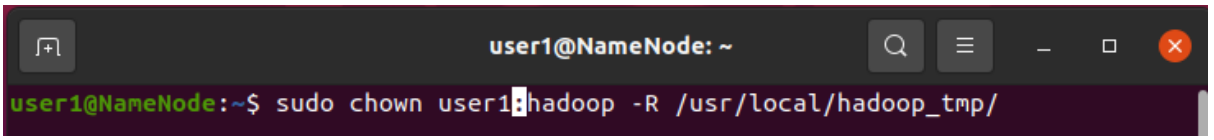
On supprime le dossier **hadoop_tmp** existant et on crée lors de la configuration pseudo-distribué.



Et on recrée par la suite le répertoire NameNode et on donne les droits d'accès de ce répertoire au user1.







e. Démarrage des services:

```
user1@NameNode:~$ hdfs namenode -format
21/12/11 17:41:11 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = NameNode/192.168.100.120
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.6.0
STARTUP_MSG:   classpath = /usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/
hadoop/common/lib/curator-client-2.6.0.jar:/usr/local/hadoop/share/hadoop/common
/lib/jackson-jaxrs-1.9.13.jar:/usr/local/hadoop/share/hadoop/common/lib/log4j-1.
2.17.jar:/usr/local/hadoop/share/hadoop/common/lib/jsch-0.1.42.jar:/usr/local/ha
```

```
user1@NameNode:~$ start-dfs.sh
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.
util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.
6.0.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.
security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflect
ive access operations
WARNING: All illegal access operations will be denied in a future release
```

```
user1@NameNode:~$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-user1-resourcem
anager-NameNode.out
DataNode2: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-user1-no
demanager-DataNode2.out
DataNode1: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-user1-no
demanager-DataNode1.out
```

Résultat dans le Master

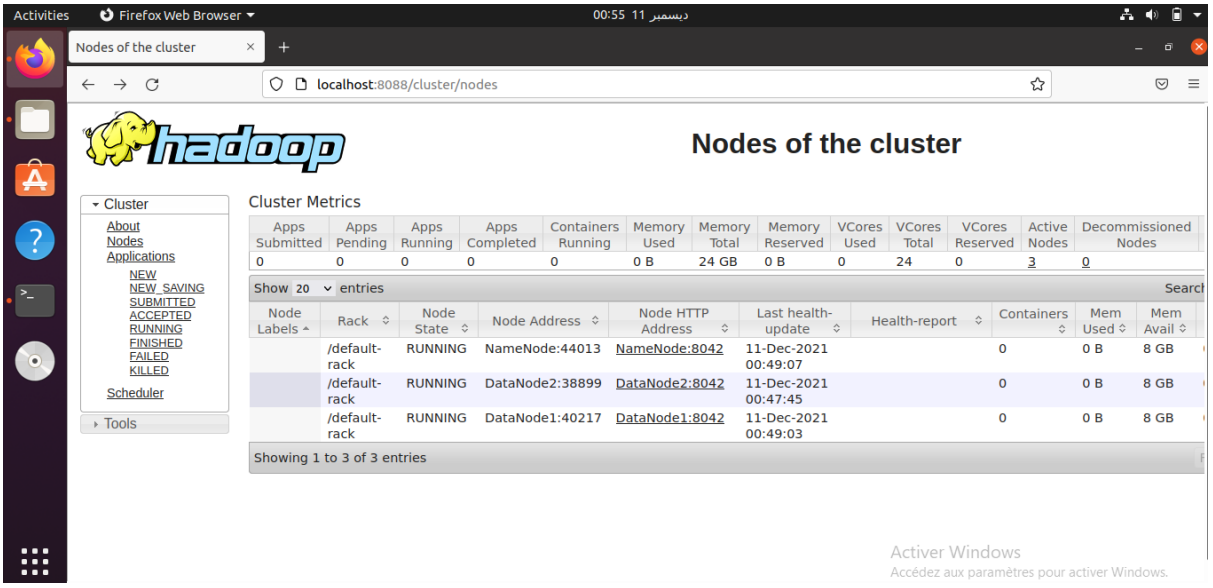
```
user1@NameNode:/usr/local/hadoop/etc/hadoop$ jps
3712 DataNode
4065 NodeManager
2068 SecondaryNameNode
1831 NameNode
4093 Jps
2207 ResourceManager
user1@NameNode:/usr/local/hadoop/etc/hadoop$
```

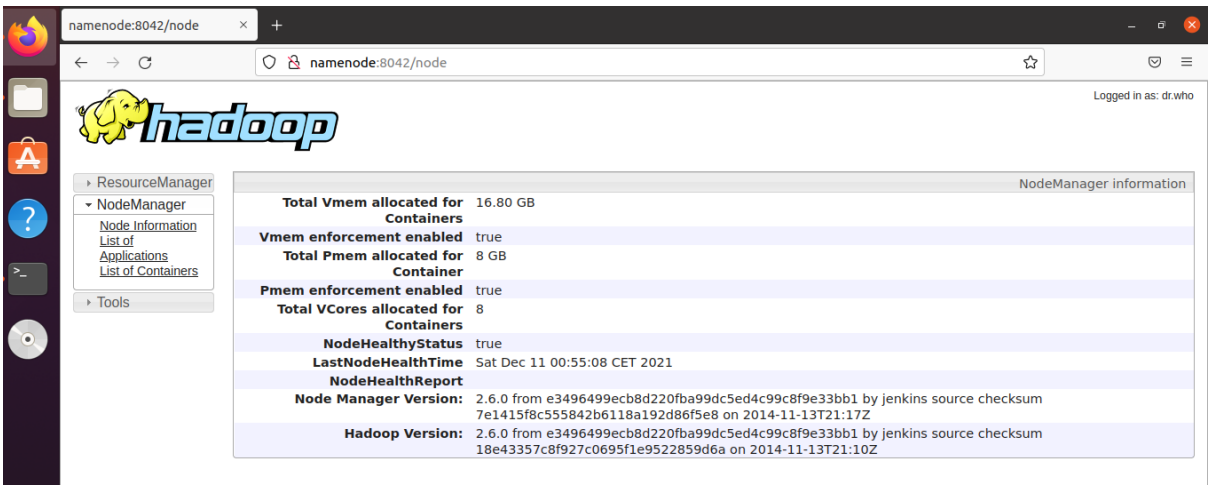
Résultat dans les workers

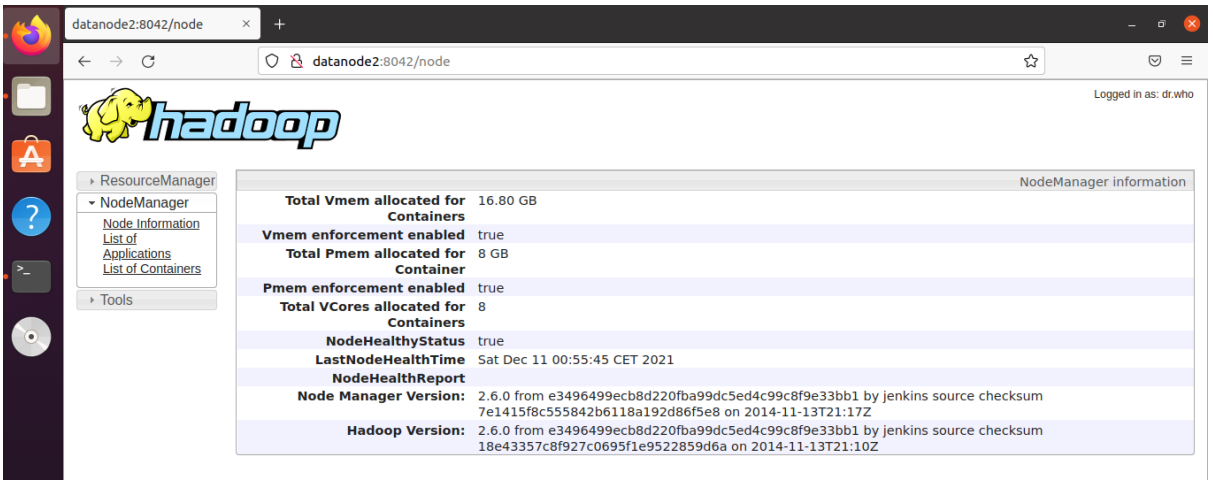
```
user1@DataNode2:~$ jps
2625 NodeManager
2417 DataNode
2852 Jps
user1@DataNode2:~$
```

```
user1@DataNode1:~$ jps
2452 DataNode
2888 Jps
2664 NodeManager
user1@DataNode1:~$
```

f. Visualisation des résultats:







g. Exécution d'un Jar:

```
user1@NameNode:~$ hdfs dfs -mkdir /test/
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.
util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.
6.0.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.
security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflect
ive access operations
WARNING: All illegal access operations will be denied in a future release
21/12/14 00:10:53 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
mkdir: `/test': File exists
```

```
user1@NameNode:~$ hdfs dfs -put /home/latifa/wordCountinput.txt.txt /test/
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.
util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.
6.0.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.
security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflect
ive access operations
WARNING: All illegal access operations will be denied in a future release
21/12/14 00:14:40 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
```

```
user1@NameNode:~$ hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-map
reduce-examples-2.6.0.jar wordcount /test/wordCountinput.txt.txt /test/output/
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.
util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.
```

```
21/12/14 00:32:03 INFO mapreduce.Job: Job job_1639436260463_0002 completed succ
ssfully
21/12/14 00:32:16 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=469
                FILE: Number of bytes written=212627
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=392
                HDFS: Number of bytes written=311
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=32312
                Total time spent by all reduces in occupied slots (ms)=264820
                Total time spent by all map tasks (ms)=32312
                Total time spent by all reduce tasks (ms)=264820
                Total vcore-seconds taken by all map tasks=32312
                Total vcore-seconds taken by all reduce tasks=264820
```