INPT
المعهد الوطني للبريد والمواصلات
ⵓⵔⴰⵡ ⵏⵜⵉⵎⵓⵔⴰⵡⵉⵏ ⵏ ⵜⵙⵙⵓⵡⴰⵏ ⴰⵏⴰⵎⵓⵔ
Institut National des Postes et Télécommunications

anrt
agence nationale de réglementation des télécommunications
الوكالة الوطنية لتقنين المواصلات

Rapport de :

TP 1 : “Installation et configuration d’Apache Hadoop et exécution d’un programme MapReduce dans un cluster hadoop à nœud unique et à nœuds multiples.”

Réalisé par :

→ Riali Mouad

→ Addi Kamal

encadré par :

→ Pr. D.Zaidouni

Table de matières :

Pré-requis techniques :

- ***Oracle AM VirtualBox-6.0 :***

  Oracle VM VirtualBox (anciennement VirtualBox) est un logiciel libre de virtualisation publié par Oracle.

  Lien de Téléchargement :

https://download.virtualbox.org/virtualbox/6.0.12/virtualbox-6.0_6.0.12-133076~Ubuntu~bionic_amd64.ddeb

✗ ***Ubuntu 18.04.3 :***

Ubuntu est un système d'exploitation GNU/Linux basé sur la distribution Linux Debian. Il est développé, commercialisé et maintenu pour les ordinateurs individuels (desktop), les serveurs (Server) et les objets connectés (Core) par la société Canonical.

Lien de Téléchargement de la version Ubuntu 20.04 :

https://ubuntu.com/download/desktop/thank-you?version=20.04.1&architecture=amd64

✗ ***Apache Hadoop version=3.2.1 :***

est un framework libre et open source écrit en Java destiné à faciliter la création d'applications distribuées et échelonnables permettant aux applications de travailler avec des milliers de nœuds et des pétaoctets de données. Ainsi chaque nœud est constitué de machines standard regroupées en grappe.

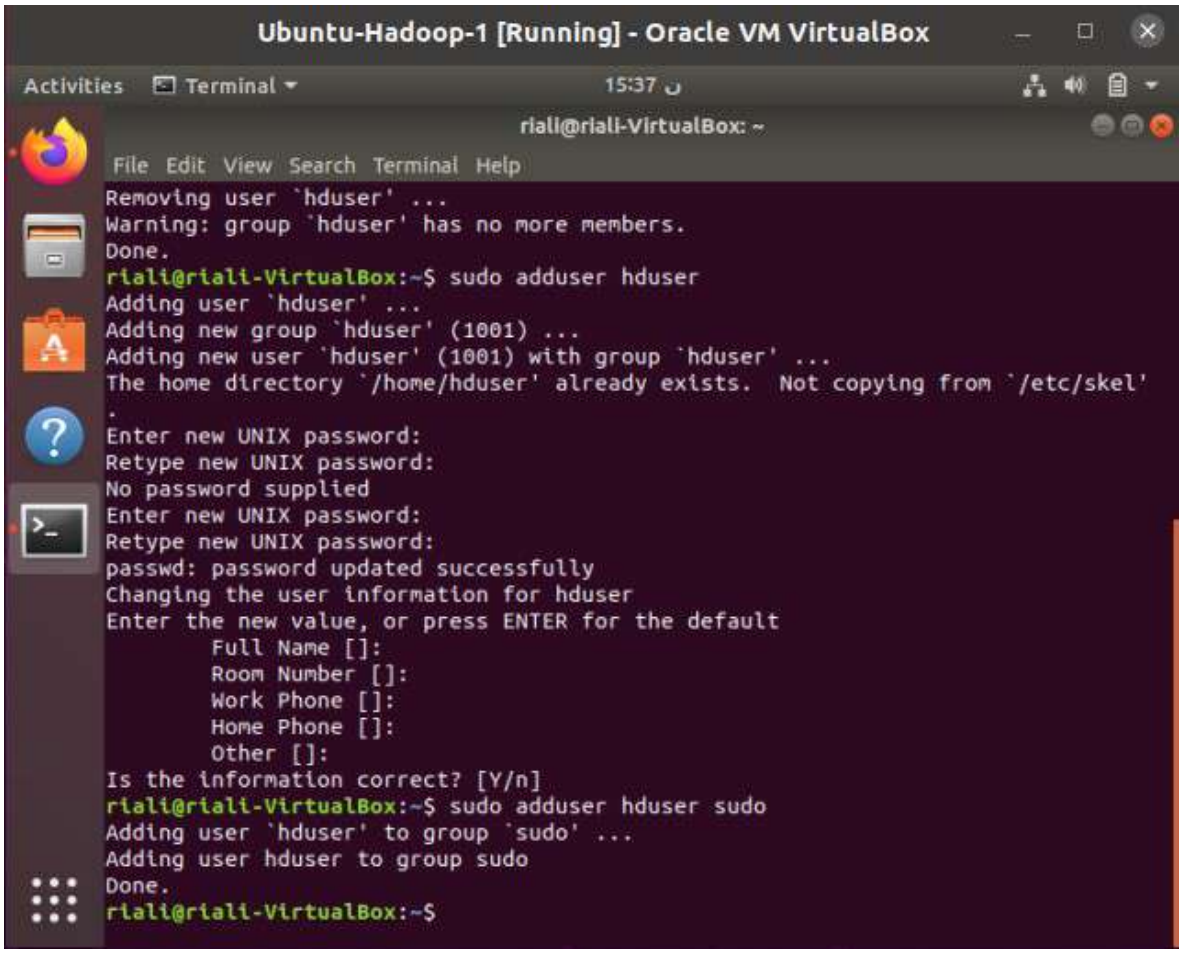
Lien de Téléchargement :

https://downloads.apache.org/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz

pour Java 8 : https://github.com/sanyoushi/java-buildpack.git

I. Installation et configuration d'un nœud unique d'Apache Hadoop 3.2.1

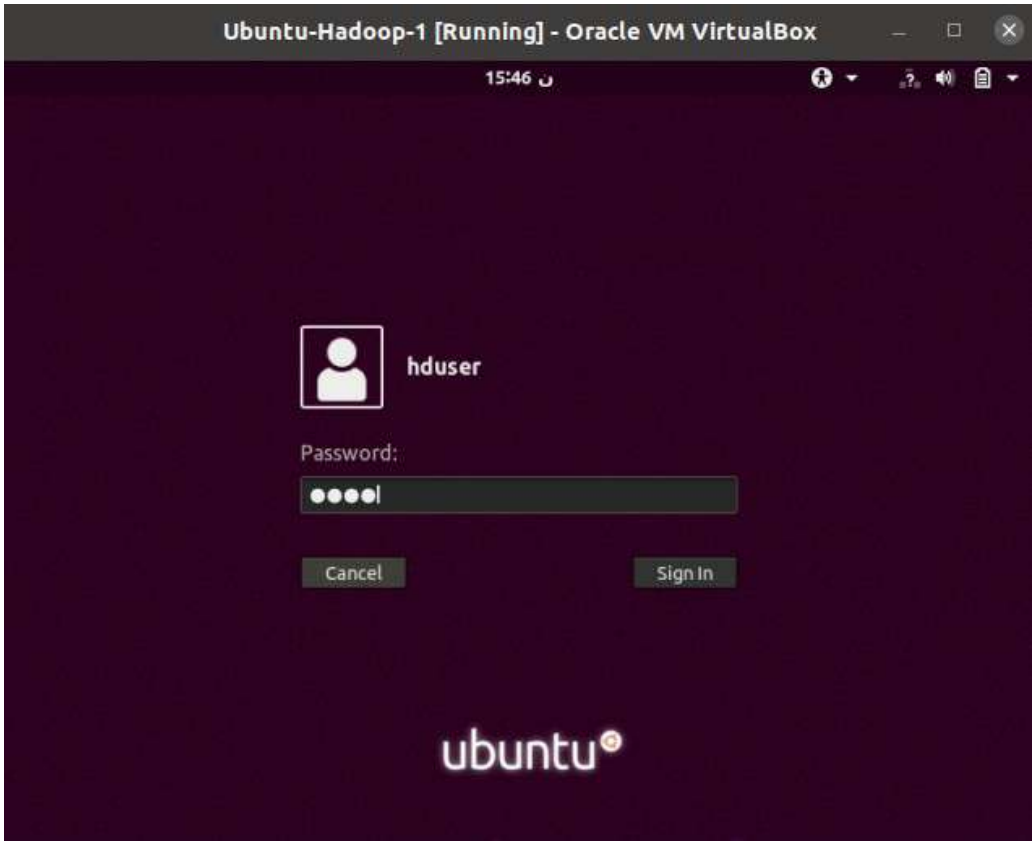
Création d'un utilisateur hduser:



- **Pour ajouter un nouveau "user" qu'on va l'appeler "hduser" , il faut juste appeler la commande suivante : sudo adduser hduser**
- **Ensuite on va appeler la commande : sudo adduser hduser sudo pour inclure le nouveau user dans le groupe sudo pour faire de lui un sudoer dont l'objectif est de lui permettre à exécuter des commandes en tant que super utilisateur .**

Mise en place de la clé ssh :

En premier lieu, il faut qu'on se connecte par l'utilisateur : hduser



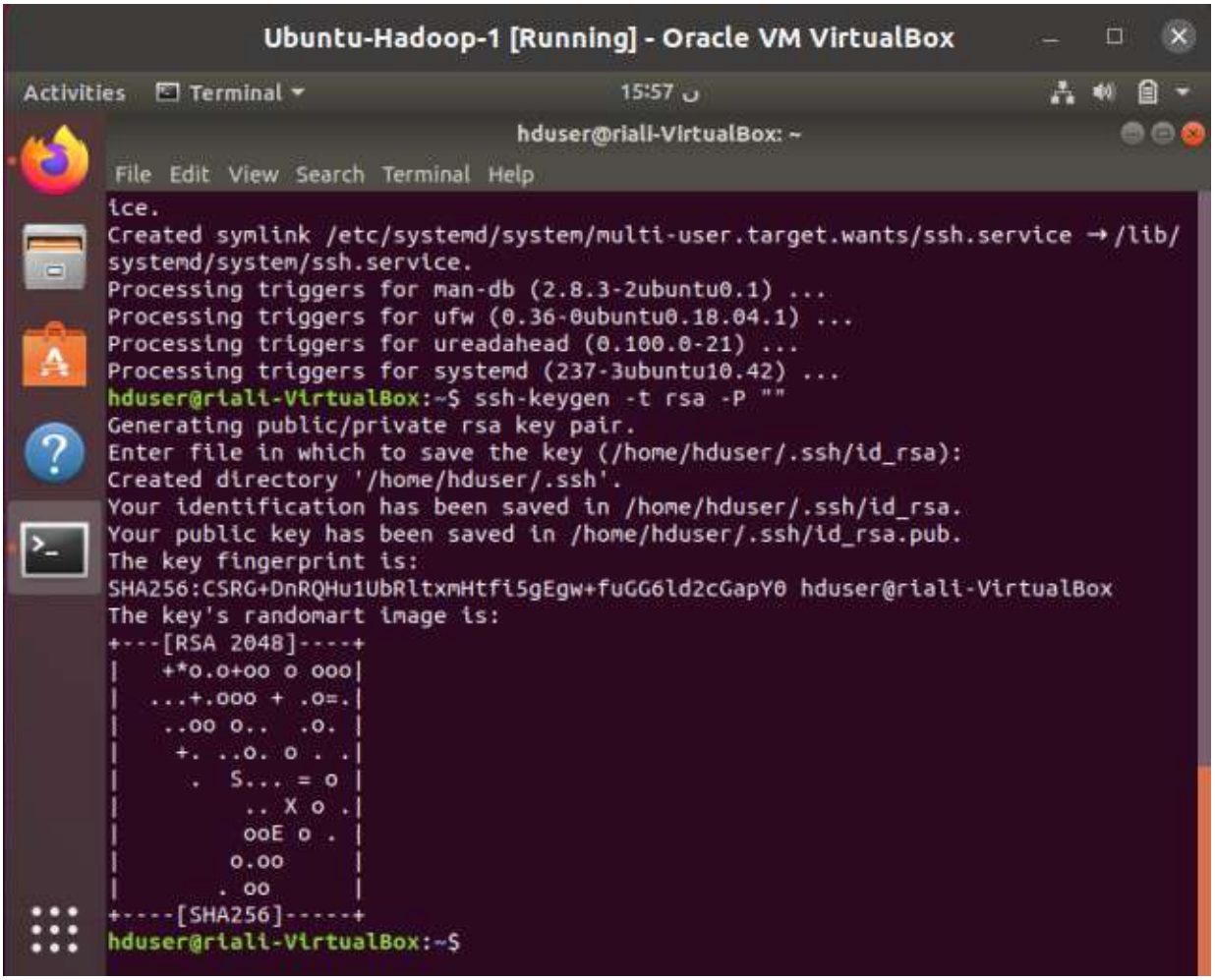
Installer le serveur openssh :

Sur Ubuntu 18.04, pour installer le serveur openssh, il ne faut qu'appeler la commande suivante : sudo apt-get install openssh-server

Mettre en place la clé ssh pour son propre compte :

Dans cette étape, on va comprendre pourquoi on doit se connecter par hduser, tout simplement , car il faut mettre en place une clé ssh pour celui-ci, pour qu'il puisse dans ce qui suit, se connecter aux autres machines de cluster qui vont être clonées à partir de cette machine-là, alors il faut taper ces commandes-là : **ssh-keygen -t rsa - P ""**

- **cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys**
- **chmod 0600 ~/.ssh/authorized_keys**

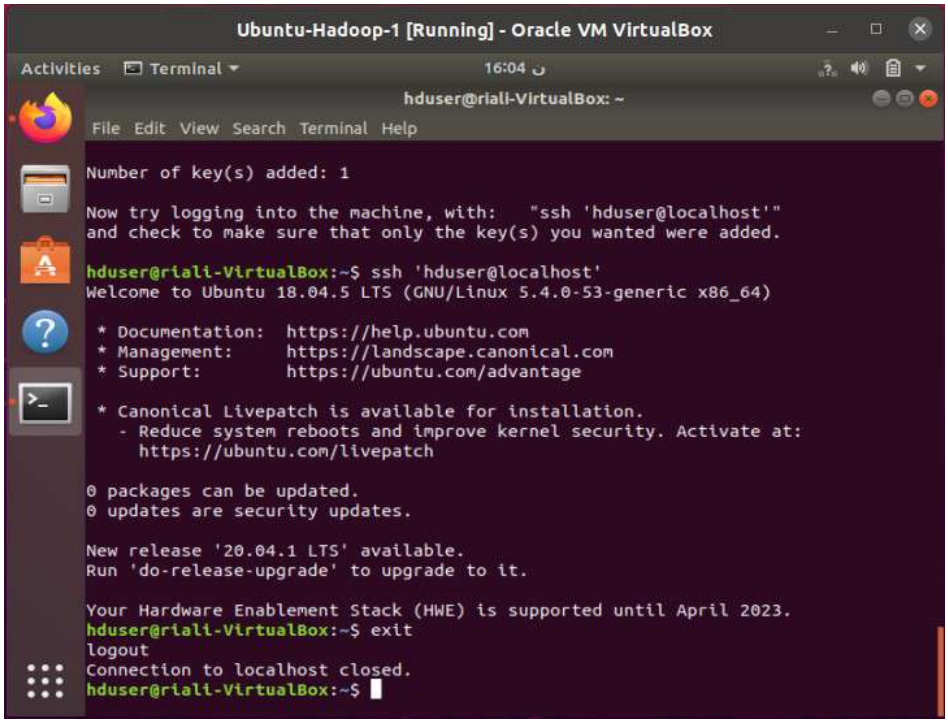


Copier la clé publique sur le serveur localhost :

Après la mise e place de la clé ssh pou le compte hduser, c'est le moment pour copier la clé publique sur le serveur localhost : **ssh-copy-id -i /home/hduser/.ssh/id_rsa.pub hduser@localhost**

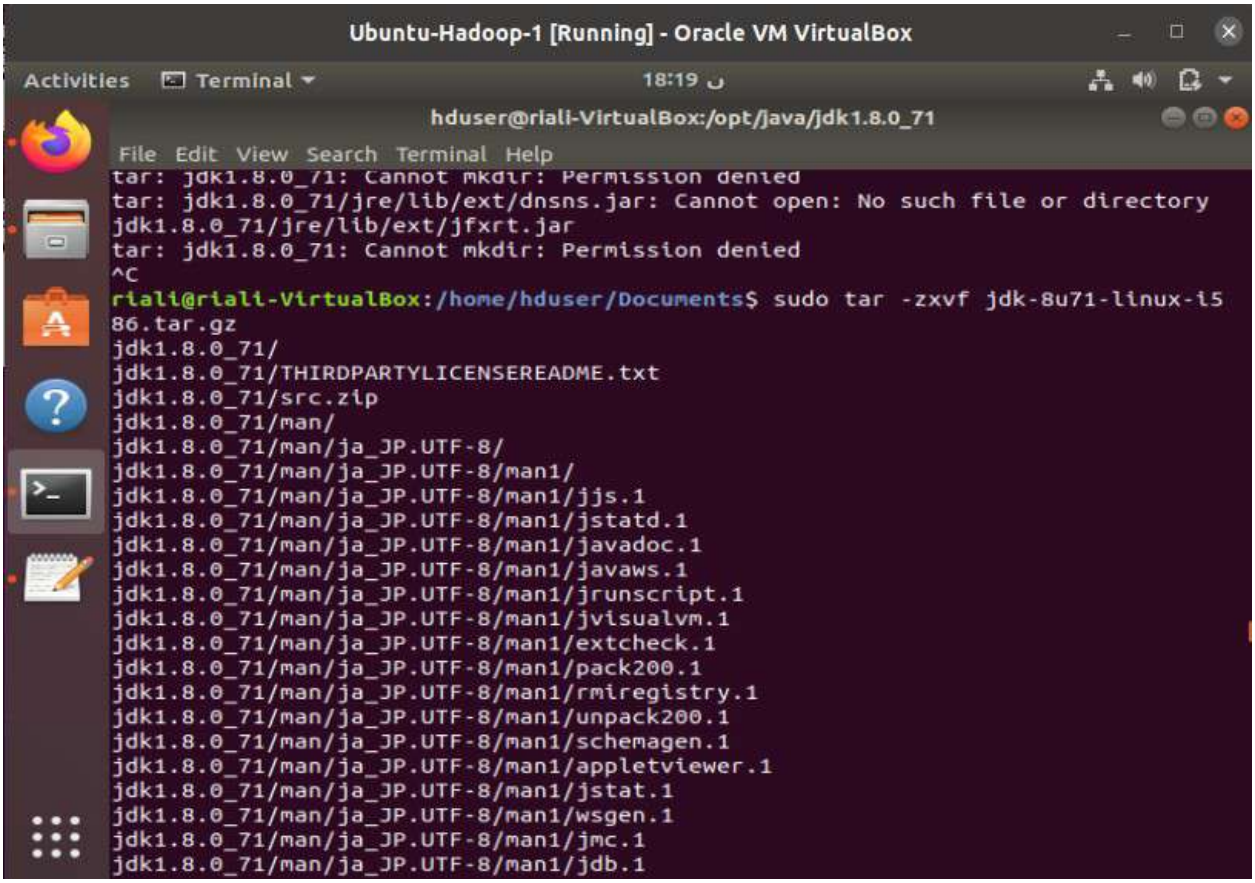
Enfin on va tester la connexion a localhost par la commande suivante : ssh 'hduser@localhost'

Installation de JAVA 8 :



Comme on a déjà mentionné au début, Hadoop est programmé par java, alors il est impérativement d'avoir un JDK pour une bonne configuration et un bon fonctionnement de Hadoop , alors dans ce TP on a choisi d'installer le JAVA 8, D'abord on va télécharger jdk-8u71-linux-x64.tar.gz et après on va suite les étapes suivantes : tar -zxvf jdk-8u71-linux-x64.tar.gz mv jdk1.8.0_71/ /opt/java/

ps : on a déjà créé le rep /opt/java



Après le déplacement du répertoire jdk1.8.0_71 vers opt/java, on va mettre à jour les liens par défaut de jdk, pour cela on va appeler les commandes suivantes dans le nouveau répertoire de java :

- **update-alternatives --install /usr/bin/java java /opt/java/jdk1.8.0_71/bin/java 100**
- **update-alternatives --config java**

et on va refaire la même chose pour javac :

- **update-alternatives --install /usr/bin/javac javac /opt/java/ jdk1.8.0_71/bin/javac 100**
- **update-alternatives –config javac**

- **sudo nano OU sudo vim OU sudo gedit /etc/profile , pour pouvoir modifier le fichier et ajouter ces trois lignes vers sa fin :**

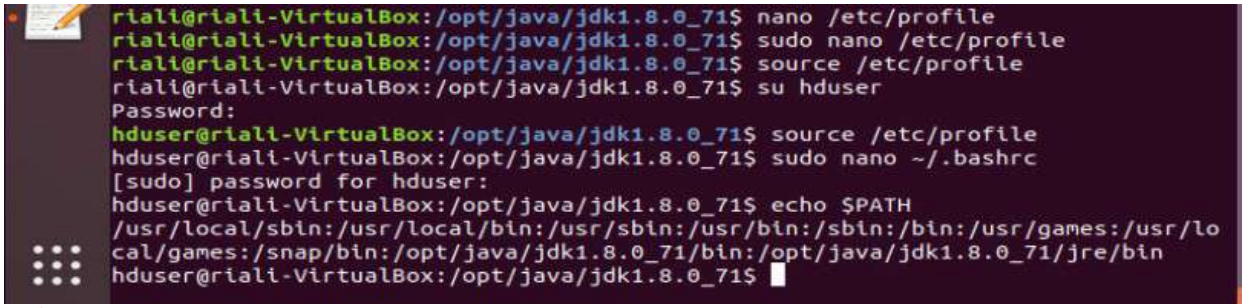
  export JAVA_HOME=/opt/java/jdk1.8.0_71/

  export JRE_HOME=/opt/java/jdk1.8.0._71/jre

  export PATH=$PATH:/opt/java/jdk1.8.0_71/bin:/opt/java/jdk1.8.0_71/jre/bin

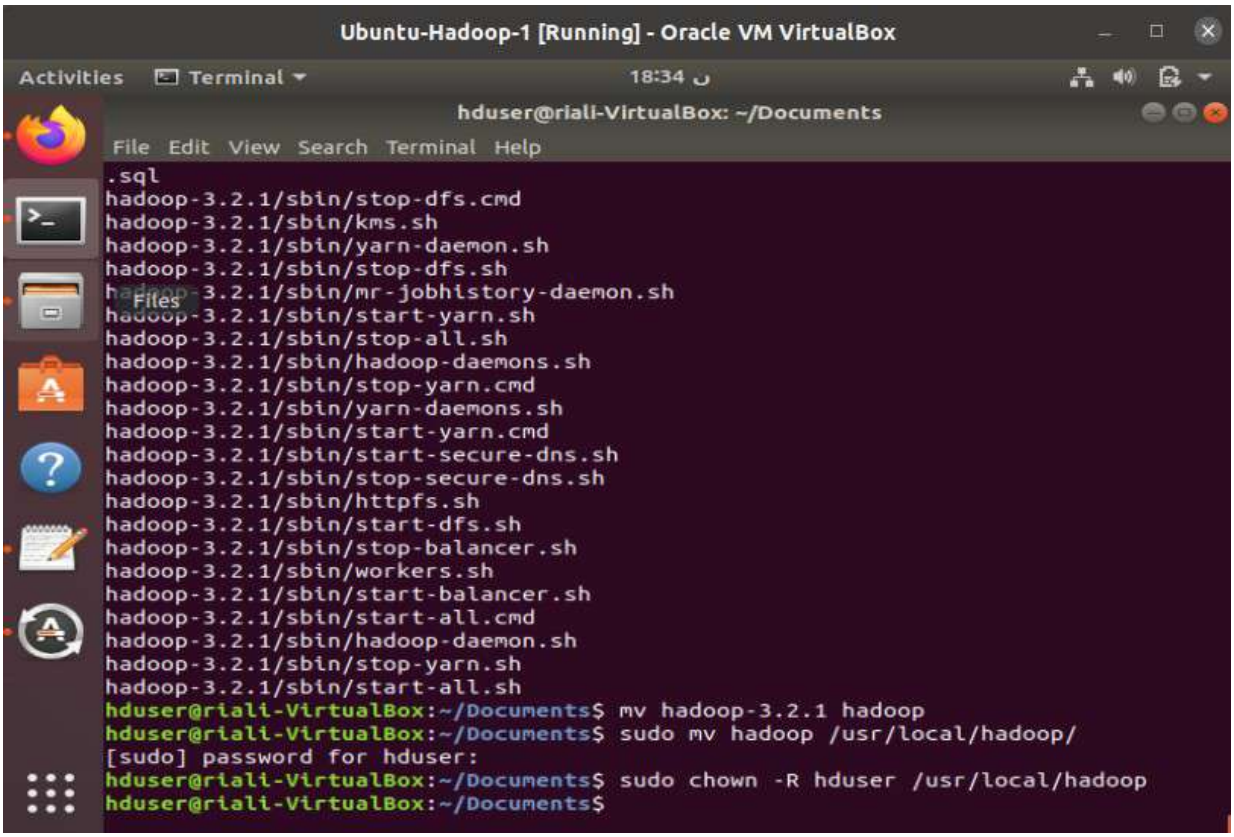
- **refaire les mêmes opérations pour le fichier : ~/.bashrc**
- **Exécuter les 2 commandes suivantes : source /etc/profile ET source ~/.bashrc**

  ET VOILA →on observe que le variable $PATH nous donne le vrai chemin vers JAVA 8

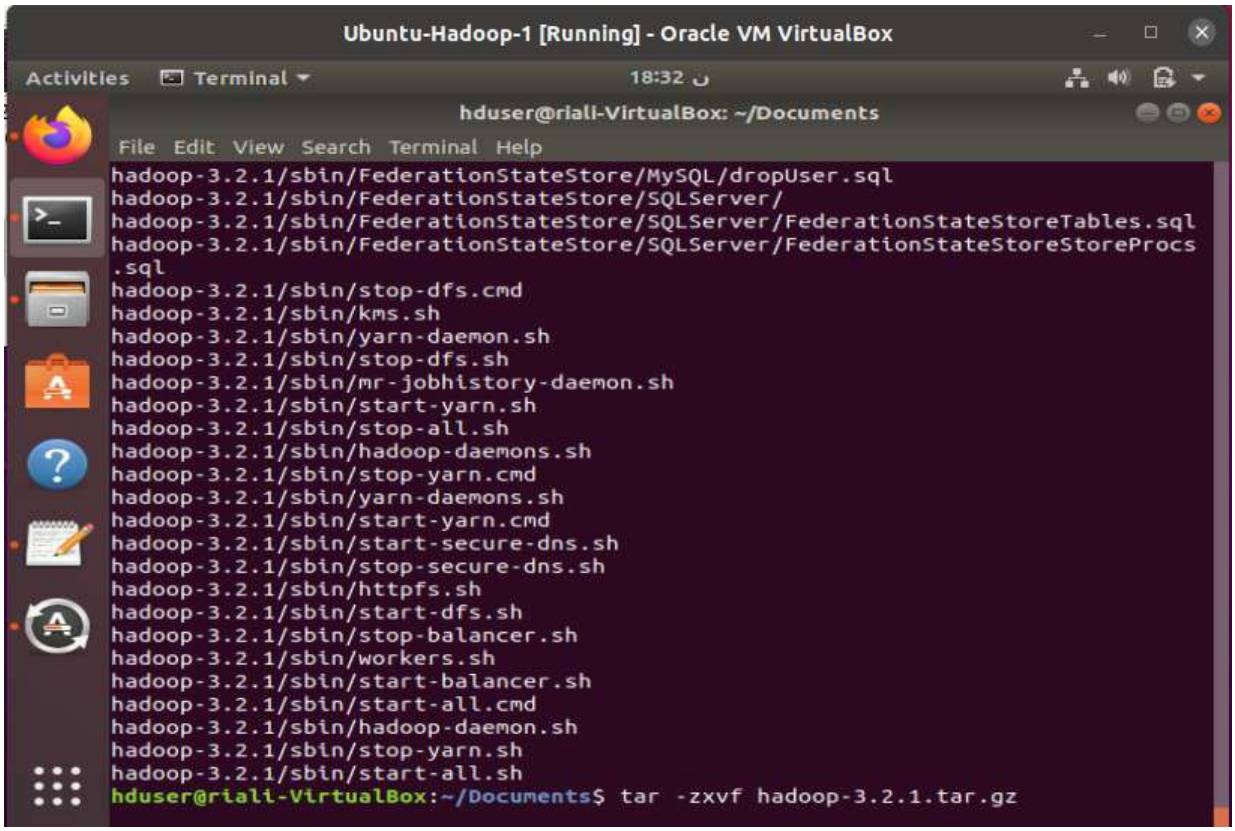


Installation d'Apache Hadoop 3.2.1

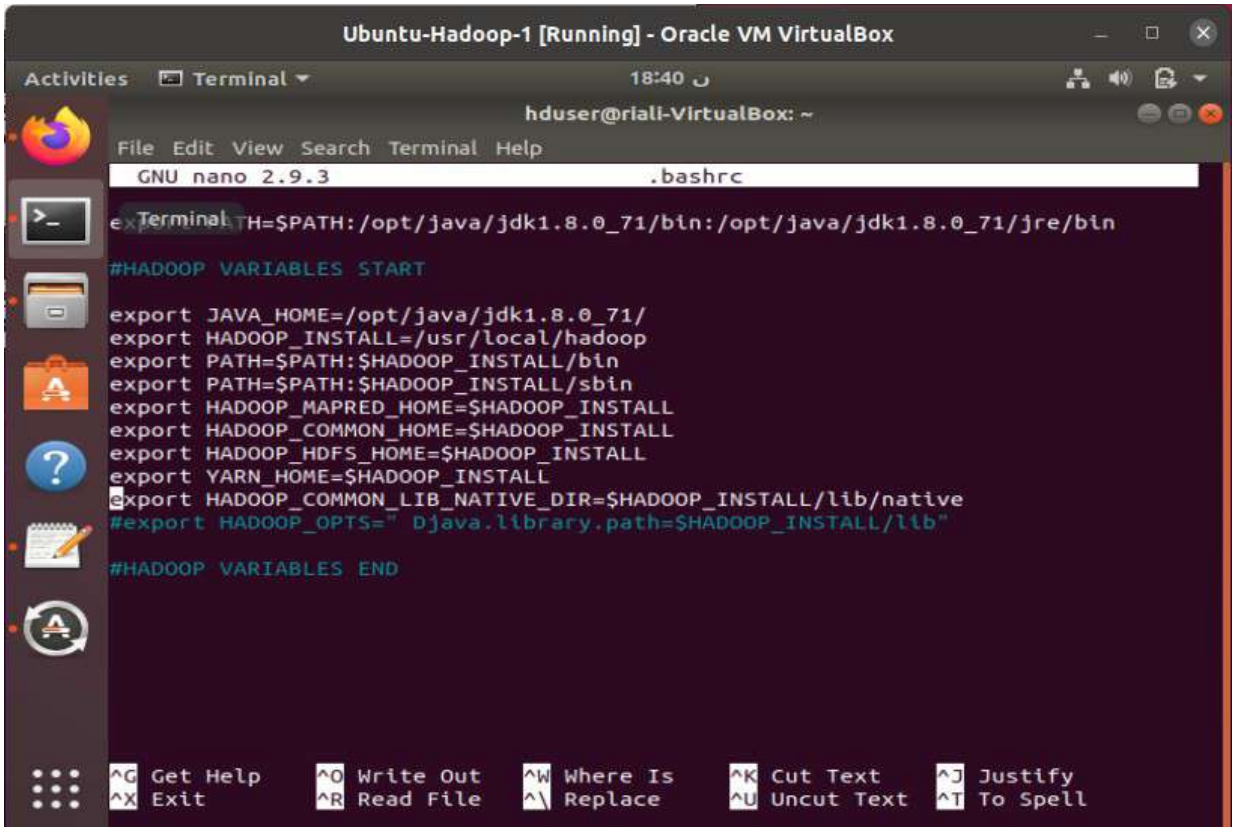
On va imiter les étapes faites pour Java 8 , sauf qu'on va déplacer le répertoire vers : "/usr/local/" **On aura alors :**



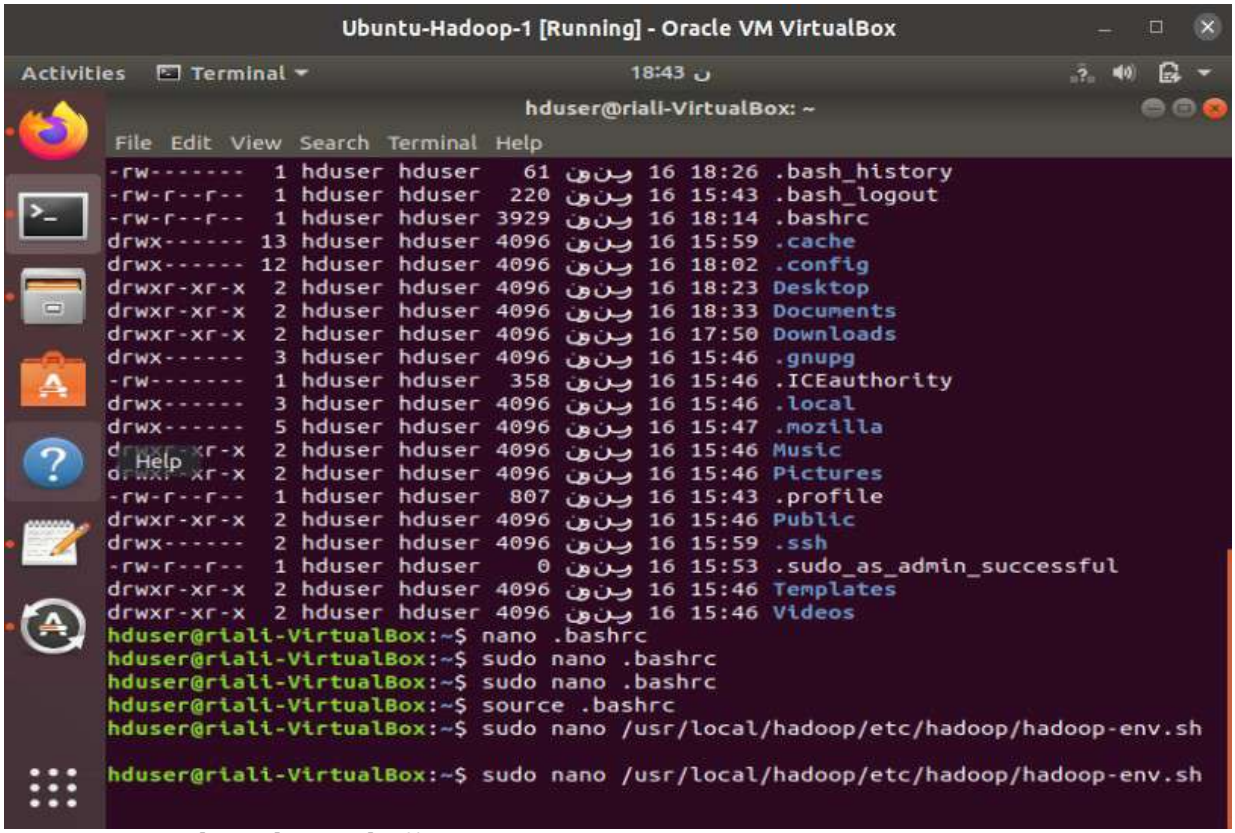
la ligne : “ sudo chown -R hduser /usr/local/hadoop” donne le droit a hduser d’opérer sur les fichiers et répertoires inclus dans /usr/local/hadoop de manière récursive.récursive.



Ainsi, il faut modifier les fichiers “.bashrc” et “*/etc/profile*” et ajouter les chemins vers $HADOOP_HOME ….etc, comme il est clair dans l’image au-dessous :



, on ouvre le fichier /usr/local/hadoop/etc/hadoop/hadoop-env.sh et on modifie la variable d'environnement JAVA_HOME : export

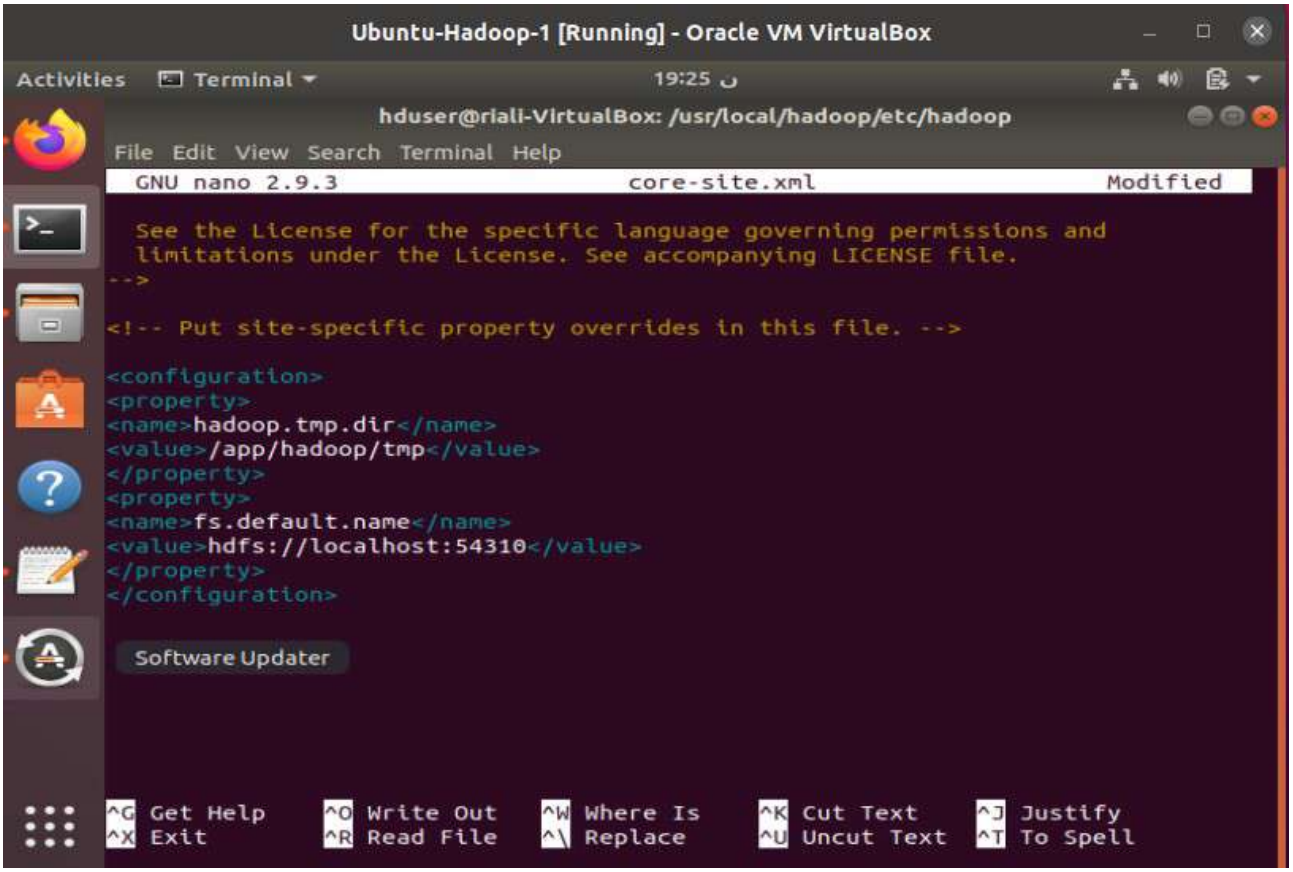


JAVA_HOME=/opt/java/jdk1.8.0_71

On crée le répertoire des fichiers temporaires de hadoop :

- **sudo mkdir -p /app/hadoop/tmp**
- **sudo chown hduser /app/hadoop/tmp**

Modification des fichiers de configuration de Hadoop :



Nous allons maintenant passer au répertoire «/ usr / local / hadoop» Commençons avec “core-site.xml” :

- hadoop.tmp.dir : C’est une base locale pour des répertoires temporaires.
- fs.default.name : Le nom du système de fichiers par défaut.défaut. Un URI dont le schéma et l’autorité déterminent l'implémentation FileSystem

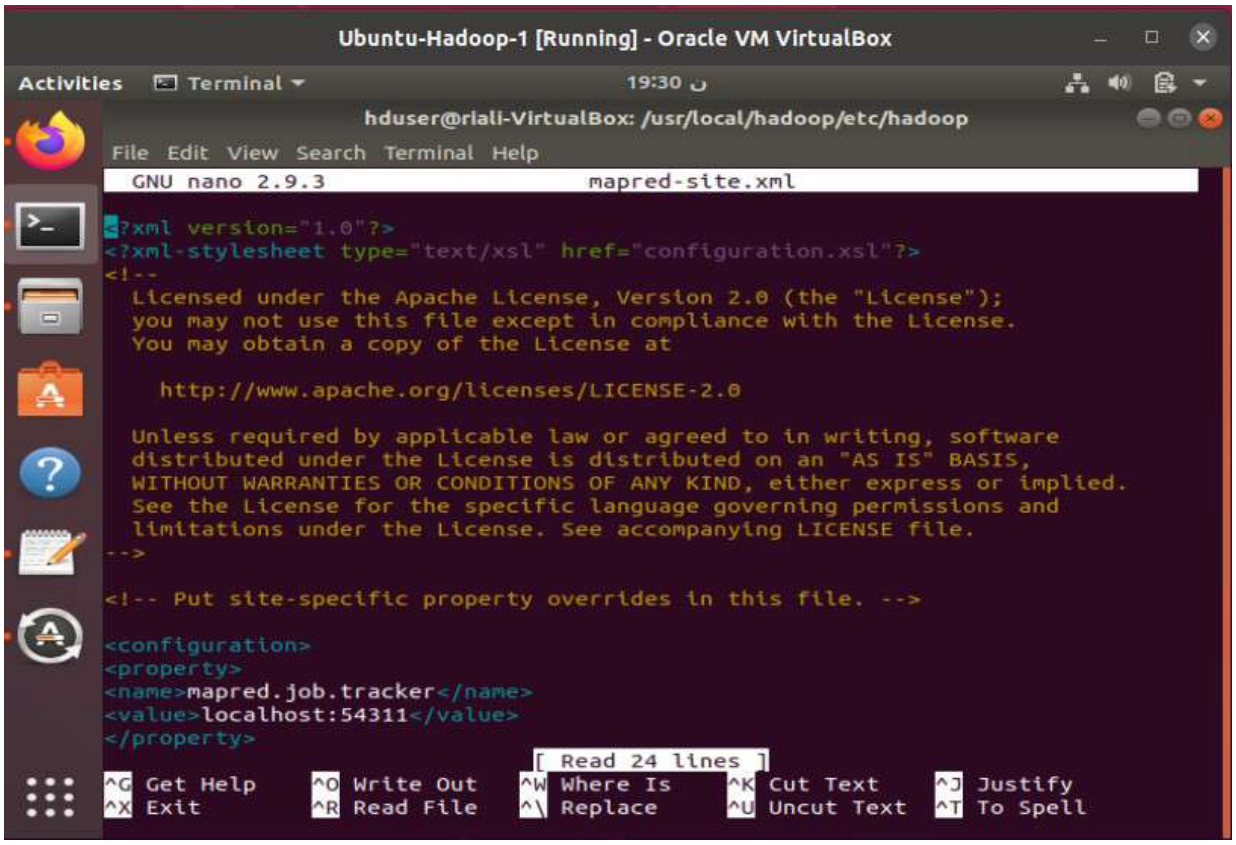
Puis on va passer au “**hdfs-site.xml**” :

- **dfs.replication :** Lorsque nous stockons les fichiers dans HDFS, la structure hadoop divise le fichier en un ensemble de blocs (64 Mo ou 128 Mo), puis ces blocs seront répliqués sur les nœuds du cluster. La configuration dfs. la

réplication consiste à spécifier le nombre de réplications requises.

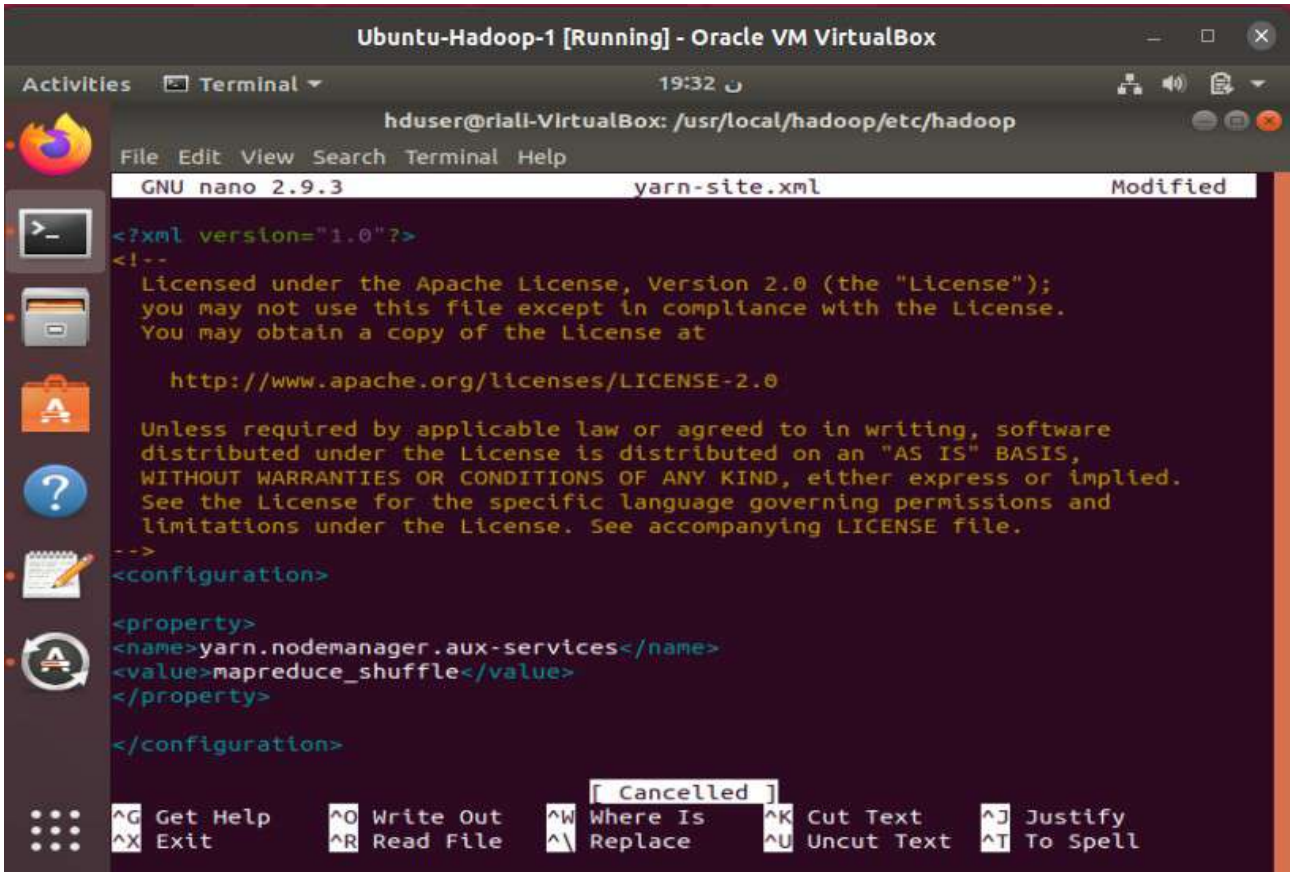
- **dfs.namenode.name.dir :** Détermine où sur le système de fichiers local le nœud de nom DFS doit stocker la table de noms (fsimage). S'il s'agit d'une liste de répertoires séparés par des virgules, la table de noms est répliquée dans tous les répertoires, pour la redondance.
- **Dfs.datanode.data.dir :** Détermine où sur le système de fichiers local un nœud de données DFS doit stocker ses blocs. S'il s'agit d'une liste de répertoires séparés par des virgules, les données seront stockées dans tous les répertoires nommés, généralement sur différents appareils. Les répertoires qui n'existent pas sont ignorés.

→ **mapred-site.xml :**



- **mapred.job.tracker :** L'hôte et le port sur lesquels s'exécute le suivi des travaux MapReduce. Si « local », les travaux sont exécutés en cours de processus comme une seule carte et réduisent la tâche.

→ **yarn-site.xml :**

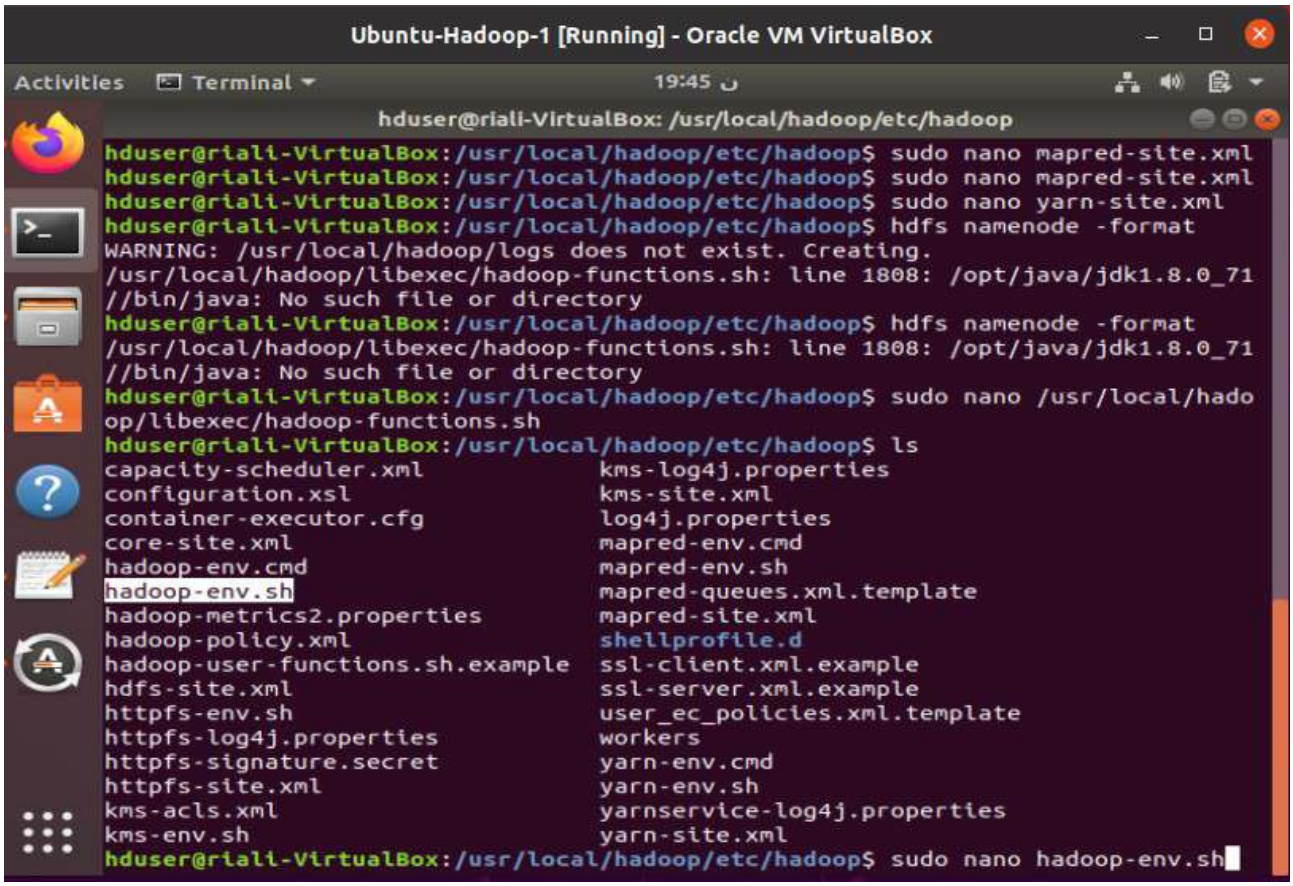


- **yarn.nodemanager.aux-services :** Le nom du service auxiliaire

Formatage du Namenode :

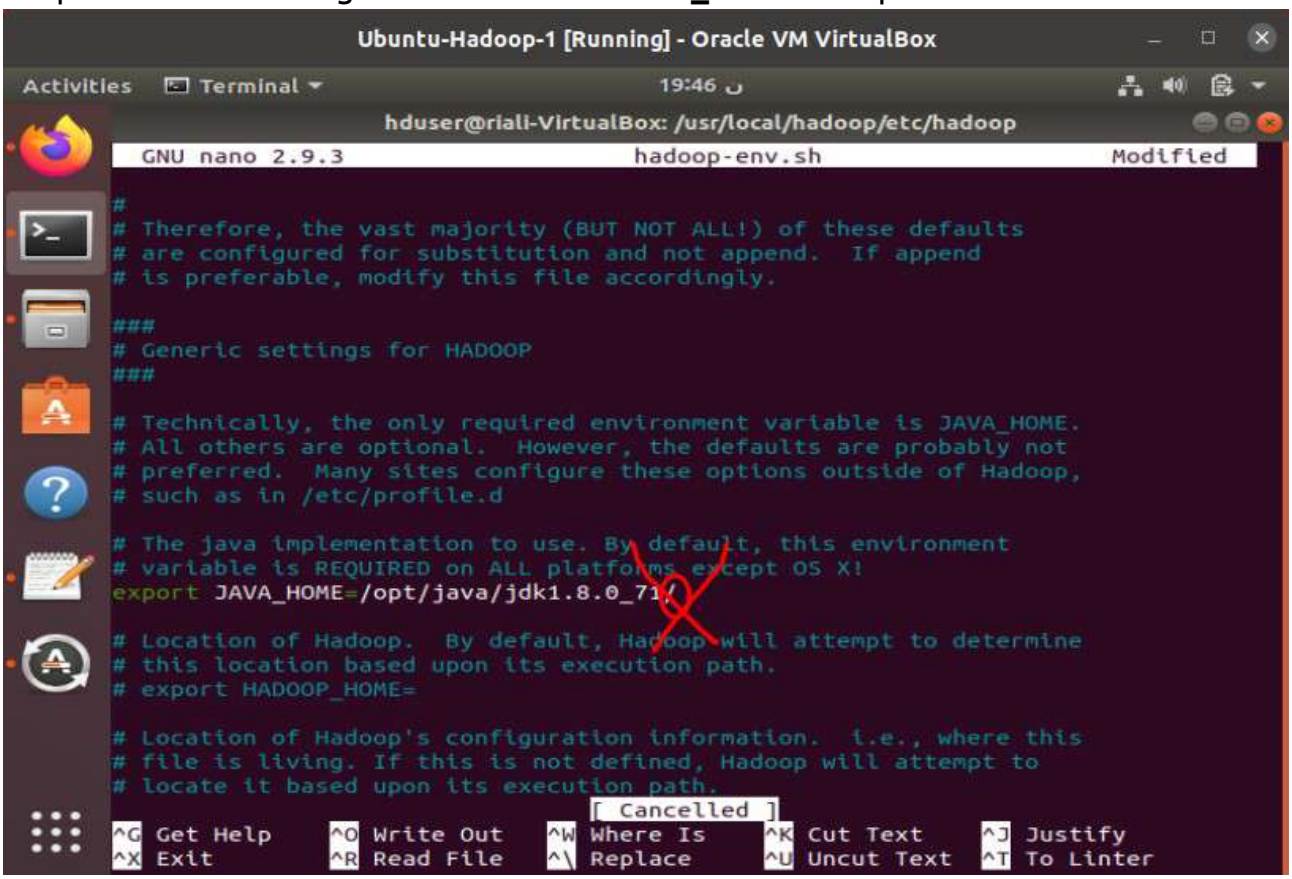
Pour pouvoir lancer le service **Hadoop Distributed File System** , il faut d'abord formatter le **Namenode** par la commande suivante :

"**hdfs namenode -format**" comme il est clair au-dessous :

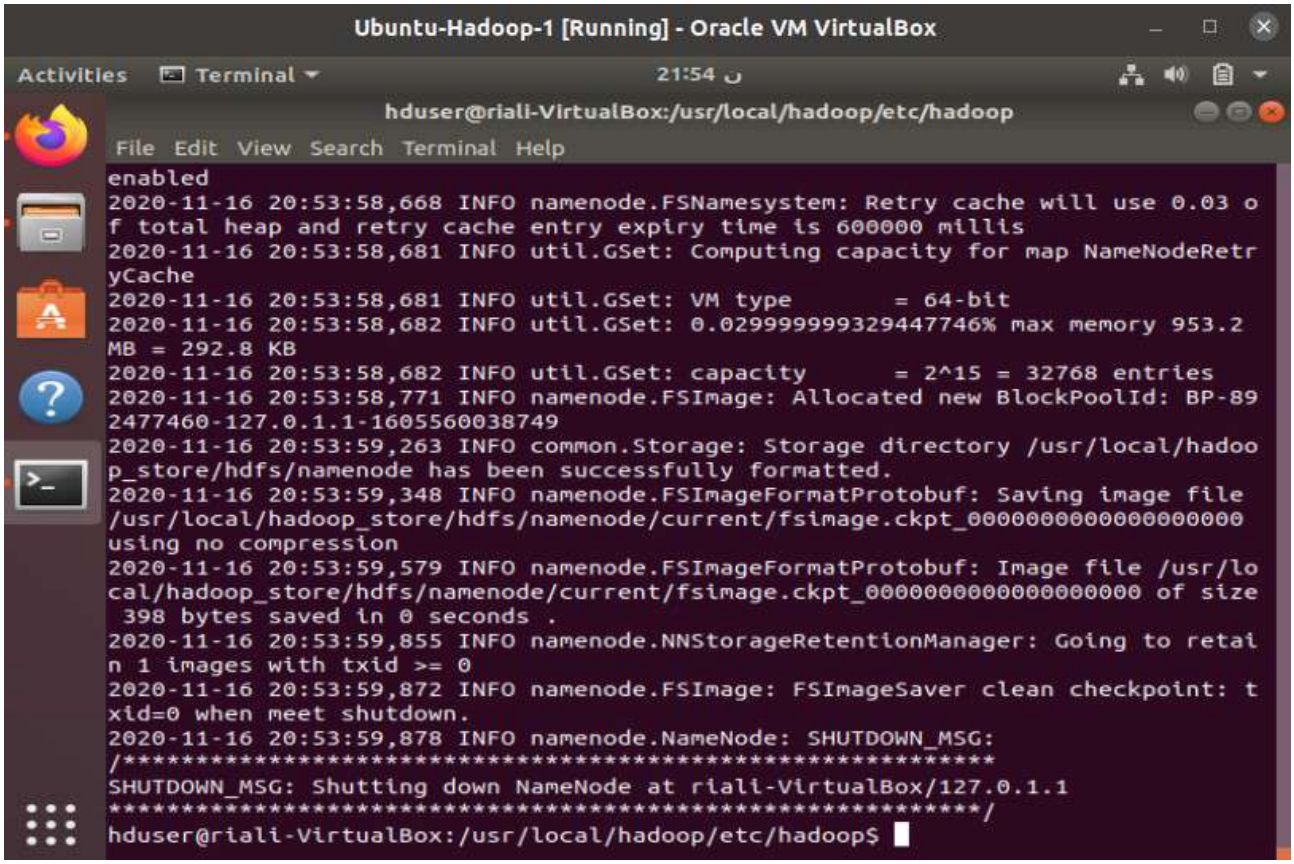


Malheureusement une erreur est survenue et voir le message d'erreur on peut bien estimer qu'il faut modifier le variable **JAVA_HOME** dans le fichier **hadoop-env.sh** :

Après avoir changé le variable **JAVA_HOME** on peut d'abord relancer



notre commande du formatage du **Namenode** :

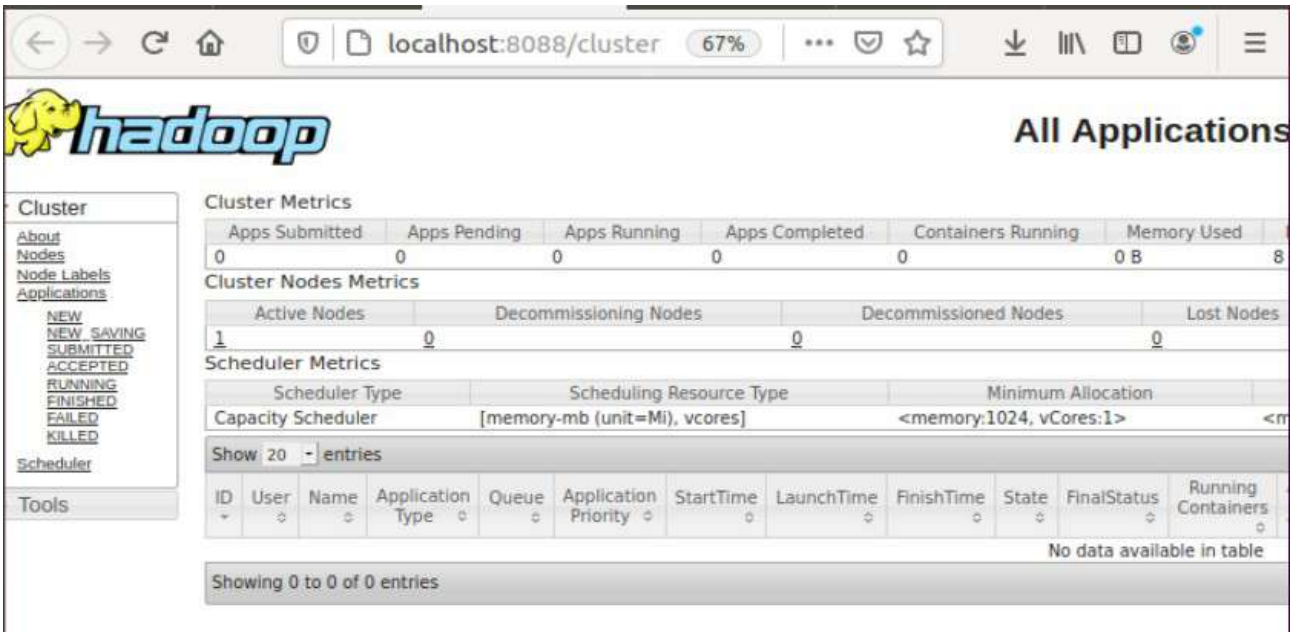


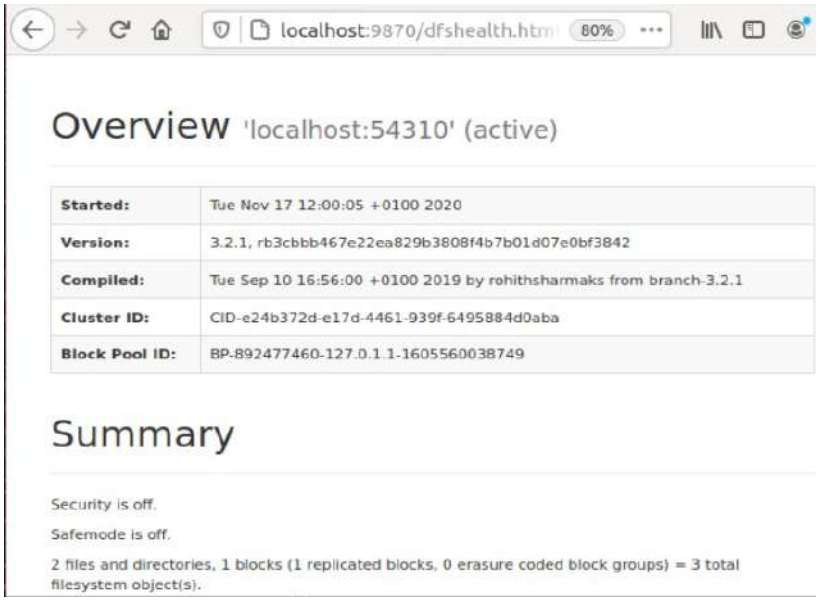
- Ensuite, on va lancer les deux commandes suivantes :
  - **start-dfs.sh :** Démarre les démons Hadoop DFS, le namenode et les datanodes
  - **start-yarn.sh :** démarre le serveur MapReduce
  - **jps :** Pour s'assurer que tout fonctionne, utiliser l'outil jps pour lister les processus Java en cours d'exécution

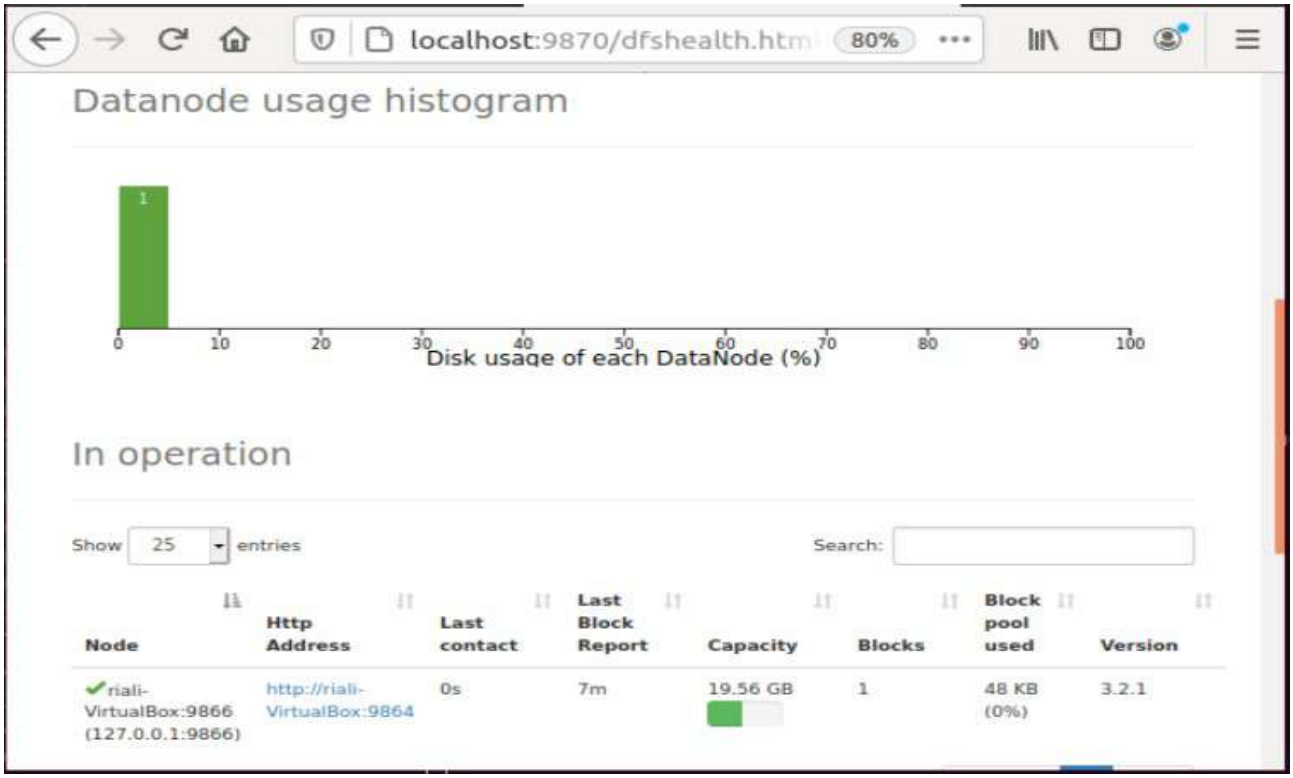
```
hduser@riali-VirtualBox:/usr/local/hadoop/etc/hadoop$ start-dfs.sh
Starting namenodes on [localhost]
localhost: namenode is running as process 2939.  Stop it first.
Starting datanodes
localhost: datanode is running as process 3101.  Stop it first.
Starting secondary namenodes [riali-VirtualBox]
riali-VirtualBox: secondarynamenode is running as process 3300.  Stop it first.
2020-11-16 21:57:02,379 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
hduser@riali-VirtualBox:/usr/local/hadoop/etc/hadoop$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hduser@riali-VirtualBox:/usr/local/hadoop/etc/hadoop$ jps
3300 SecondaryNameNode
5275 Jps
2939 NameNode
4924 NodeManager
3101 DataNode
4767 ResourceManager
```

II. Exécution d'un programme Map/Reduce dans un cluster à nœud unique :

Accéder aux services de Hadoop via le navigateur :



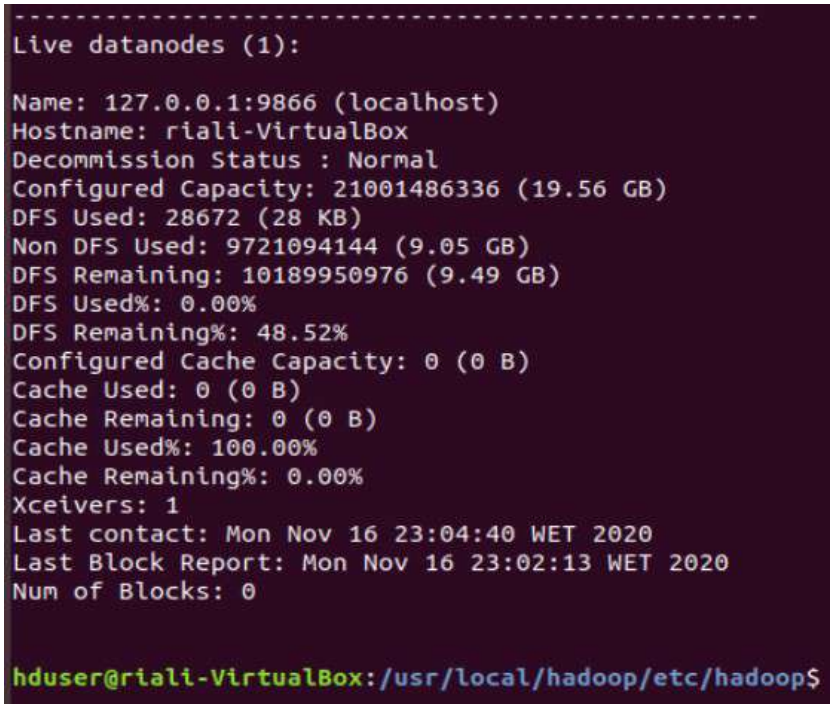




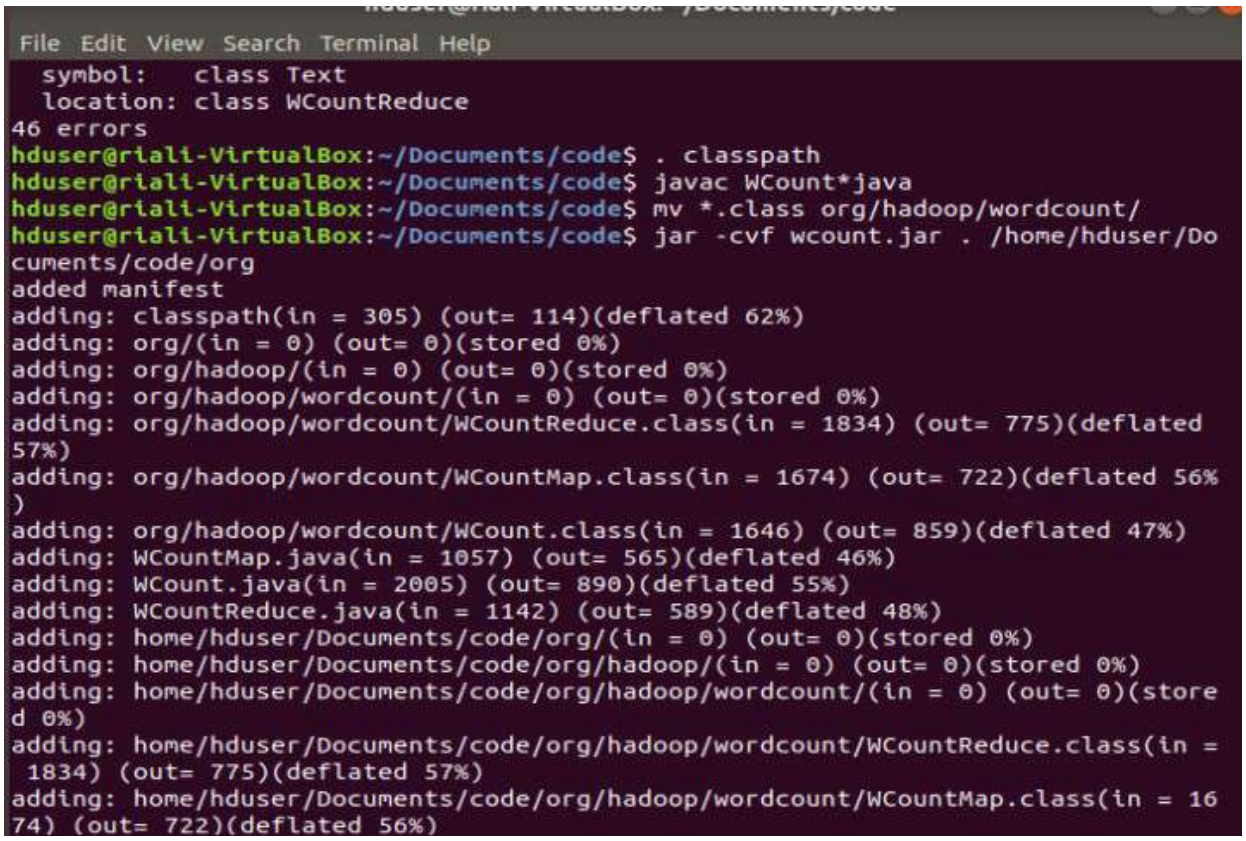
Exécution d'un programme Map/Reduce :

Pour exécuter le programme **Map/Reduce,** on va exécuter les commandes suivantes :

- **hdfs dfsadmin -report :** “pour tester le bon fonctionnement du service **hdfs**”



- **cd /home/hduser/Documents/code/ :** accéder au répertoire contenant les fichiers **wordcount** et le fichier **poeme.txt**
- **mkdir -p org/hadoop/wordcount/ :** créer les répertoires **org ET hadoop ET wordcount**
- **. classpath :** exécuter le fichier “classpath”
- **javac WCount*java :** exécuter tous les fichiers {Wcount*java}
- **mv *.class org/hadoop/wordcount/**
- jar -cvf wcount.jar . /home/hduser/Documents/code/org : Générer le .jar



- **cd /usr/local/hadoop/ :** accéder au répertoire de hadoop
- **bin/hdfs dfs -put /home/hduser/Documents/code/poeme.txt / :** Copier “poeme.txt” de fichiers local vers le système de fichiers de destination. Lit également l’entrée de stdin et écrit dans le système de fichiers de destination.
- **bin/hdfs dfs -ls / :** Pour un répertoire comme notre cas, il renvoie la liste de ses enfants directs comme dans Unix
- **cd /home/hduser/Documents/code/**
- **hadoop jar wcount.jar org.hadoop.wordcount.WCount /poeme.txt /results**

```
hduser@riali-VirtualBox:~/Documents/code$ cd /usr/local/hadoop/
hduser@riali-VirtualBox:/usr/local/hadoop$ bin/hdfs dfs -put /home/hduser/Docum
ents/code/poeme.txt /
2020-11-17 11:39:54,447 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
put: `/home/hduser/Documents/code/poeme.txt': No such file or directory
hduser@riali-VirtualBox:/usr/local/hadoop$ bin/hdfs dfs -put /home/hduser/Docum
ents/code/poeme.txt /
2020-11-17 11:40:51,377 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
put: `/poeme.txt': File exists
hduser@riali-VirtualBox:/usr/local/hadoop$ bin/hdfs dfs -ls /
2020-11-17 11:41:23,131 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--   1 hduser supergroup       1670 2020-11-17 00:11 /poeme.txt
hduser@riali-VirtualBox:/usr/local/hadoop$ cd /home/hduser/Documents/code/
hduser@riali-VirtualBox:~/Documents/code$ hadoop jar wcount.jar org.hadoop.word
count.WCount /poeme.txt /results
2020-11-17 11:42:48,083 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
2020-11-17 11:42:49,654 INFO impl.MetricsConfig: Loaded properties from hadoop-
metrics2.properties
2020-11-17 11:42:50,836 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot
period at 10 second(s).
2020-11-17 11:42:50,836 INFO impl.MetricsSystemImpl: JobTracker metrics system
started
```

- **hadoop fs -ls /results**
- **hadoop fs -cat /results/part-r-00000**

```
hduser@riali-VirtualBox:~/Documents/code$ hadoop fs -ls /results
2020-11-17 11:43:38,675 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2020-11-17 11:42 /results/_SUCCESS
-rw-r--r--   1 hduser supergroup       2823 2020-11-17 11:42 /results/part-r-00
000
hduser@riali-VirtualBox:~/Documents/code$  hadoop fs -cat /results/part-r-00000

2020-11-17 11:44:24,582 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
2020-11-17 11:44:26,078 INFO sasl.SaslDataTransferClient: SASL encryption trust
 check: localHostTrusted = false, remoteHostTrusted = false
a       6 occurences.
adoraient       1 occurences.
ailes   1 occurences.
aima    1 occurences.
amour   1 occurences.
au      11 occurences.
bas     1 occurences.
belle   1 occurences.
bles    1 occurences.
bras    1 occurences.
bretagne        1 occurences.
brula   1 occurences.
celle   1 occurences.
celui   20 occurences.
cette   1 occurences.
chancelle       1 occurences.
```

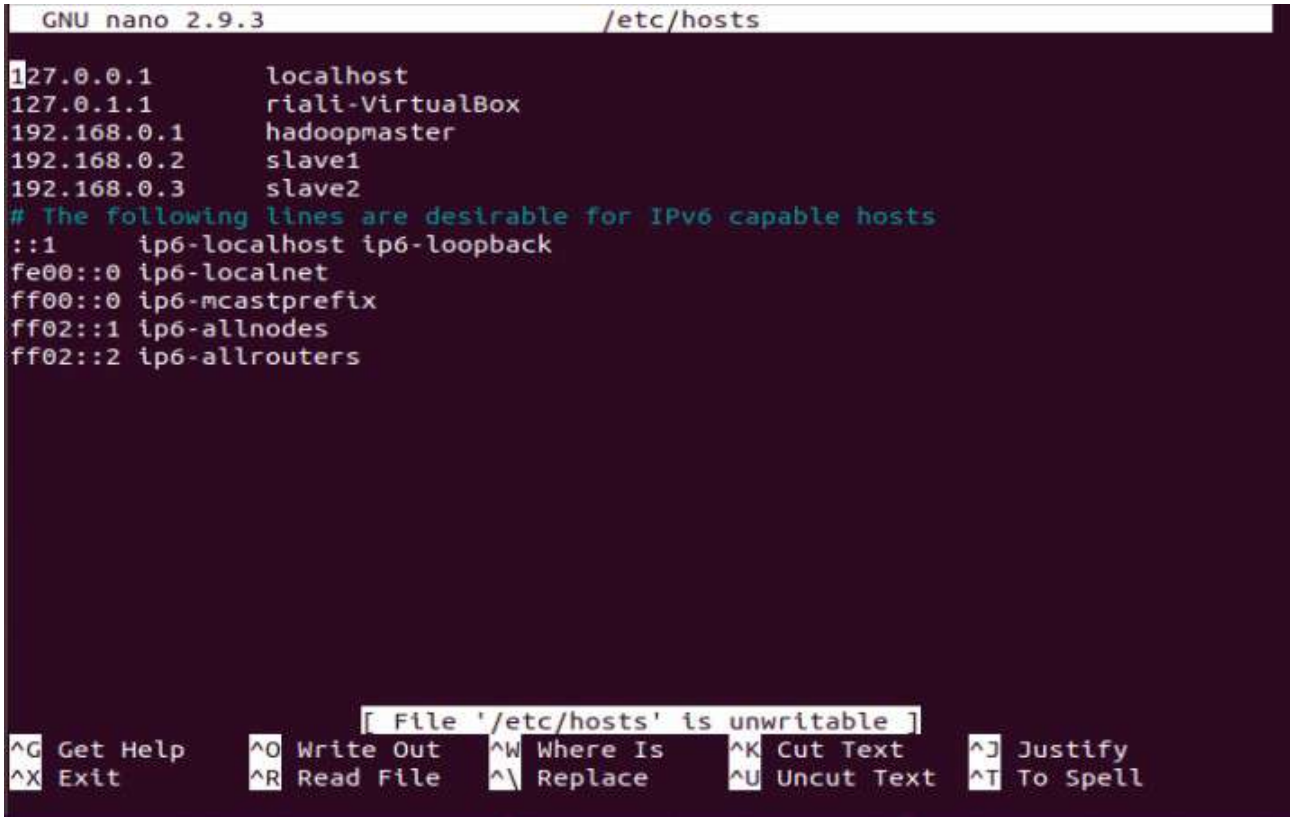
III. Configuration d'un cluster multi-noeuds d'Apache Hadoop .

Attribution statique d'adresse IP à la machine hadoopmaster :

/etc/hosts

Modifier le fichier /etc/hosts de la machine en ajoutant les lignes suivantes qui représentent les IP fixes des différentes machines du cluster :

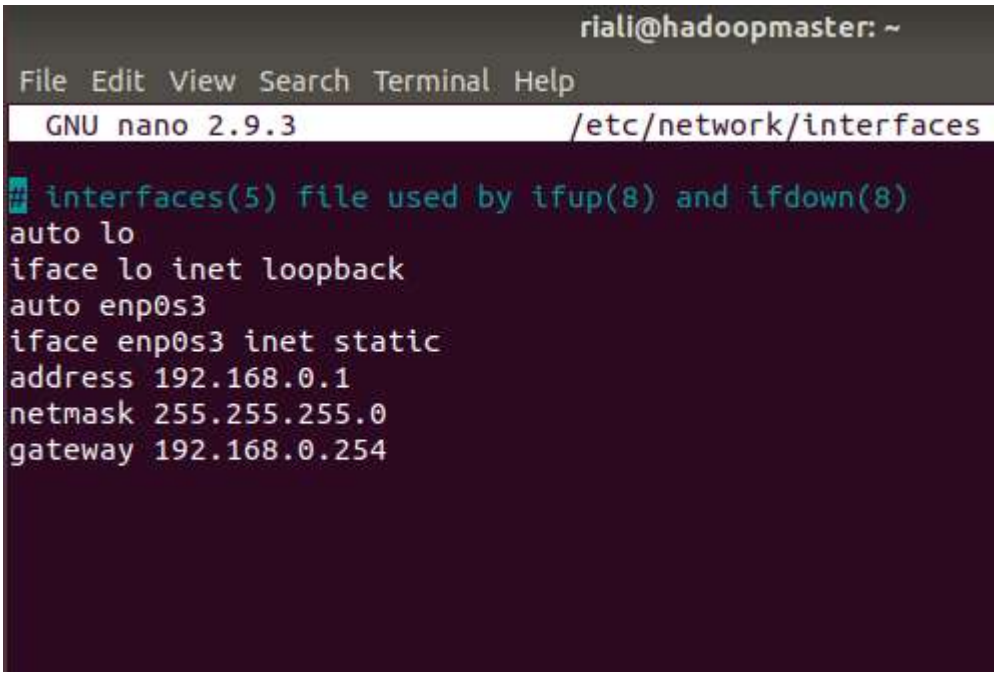
- 192.168.0.1 hadoopmaster
- 192.168.0.2 slave1
- 192.168.0.3 slave2



/etc/network/interfaces :

Afin de configurer de manière permanente l'adresse IP fixe de la machine hadoopmaster, il faut ajouter les lignes suivantes dans /etc/network/interfaces :

```
auto enp0s3
iface enp0s3 inet static
address 192.168.0.1
netmask 255.255.255.0
gateway 192.168.0.254
```

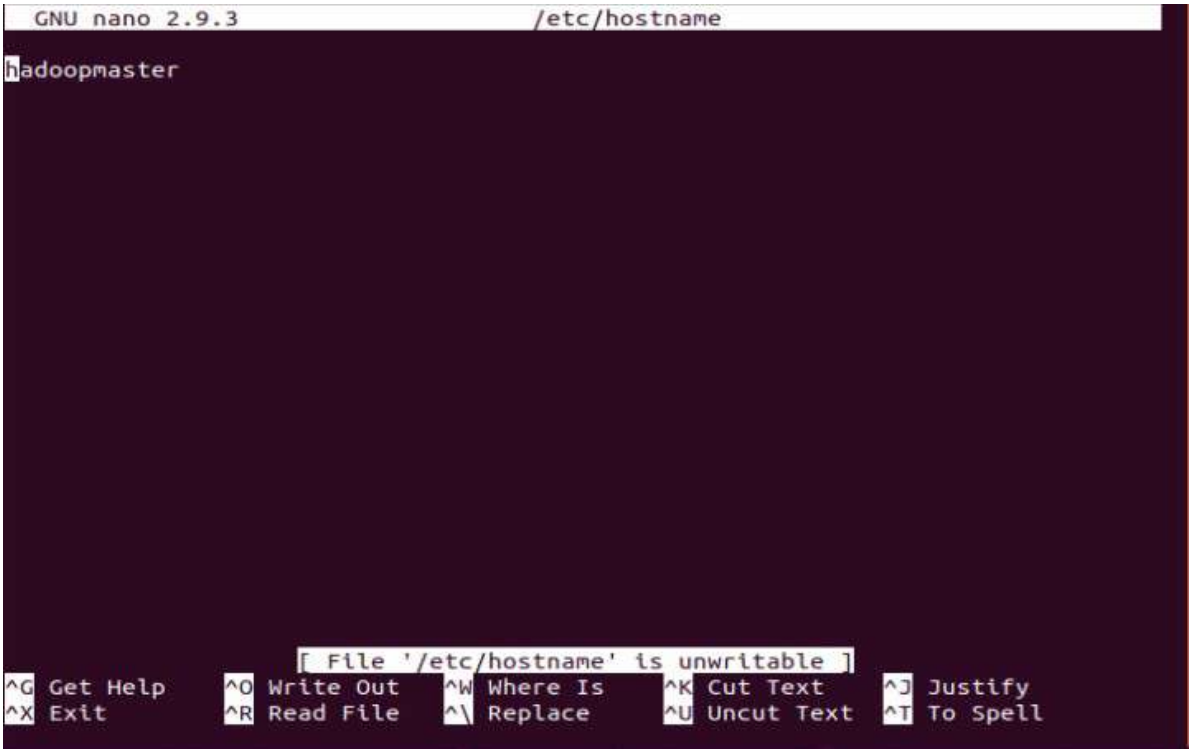


Pour prendre en compte ces configurations on doit redémarrer la machine :

sudo reboot

etc/hostname :

Modifier le contenu de ce fichier par : **hadoopmaster**



Modification des fichiers de configuration de hadoop :

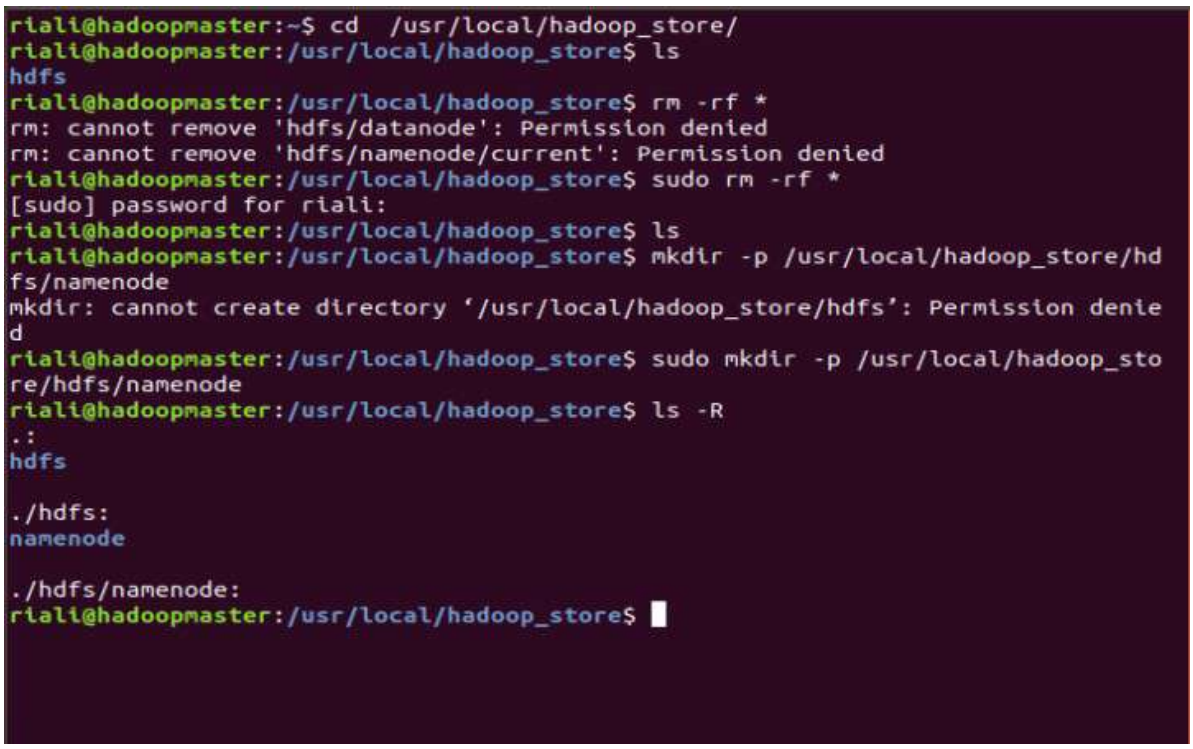
hadoop_master :

Ensuite , il faut supprimer les fichiers dans **hadoop_master** et créer de nouveau le répertoire :

***/usr/local/hadoop_master/*hdfs/namenode** exécuter les commandes suivantes :

- **cd /usr/local/hadoop_store/**
- **sudo rm -rf ***
- **sudo mkdir -p /usr/local/hadoop_store/hdfs/namenode**

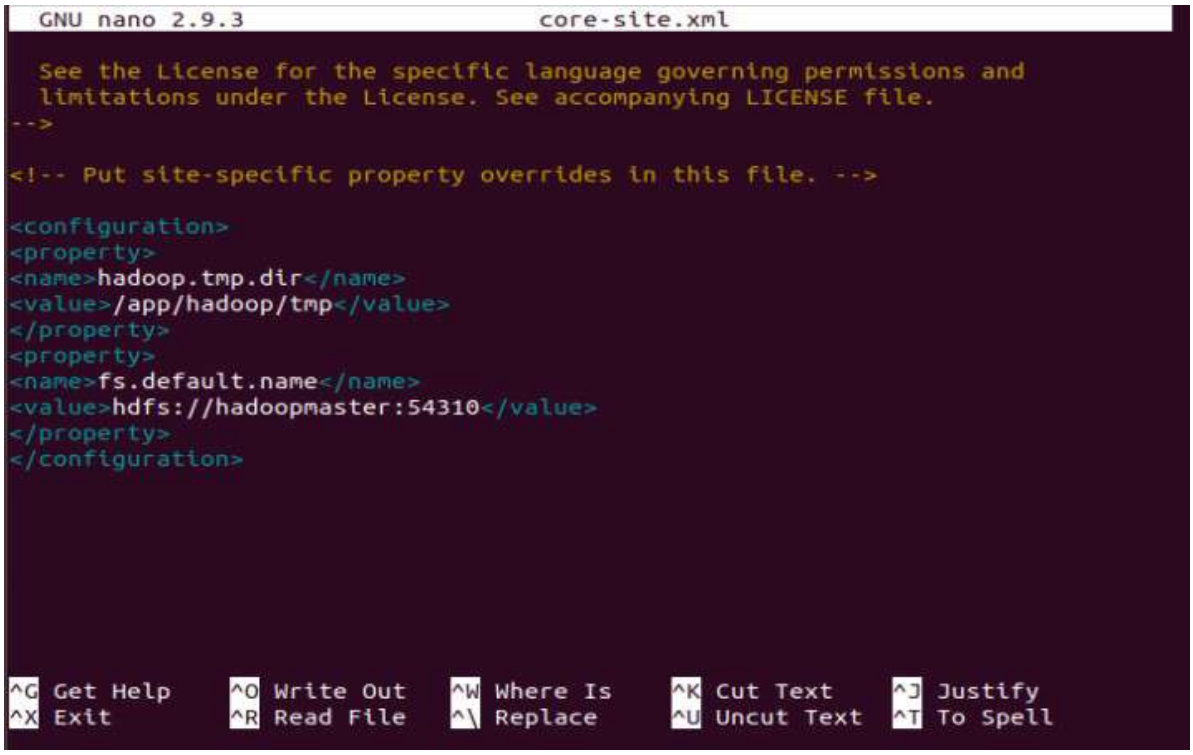
- **chown -R hduser /*usr*/local/hadoop_store/**



core-site.xml : [$ cd /usr/local/hadoop/etc/hadoop/]

Modifier le fichier **core-site.xml,** en modifiant la balise **<configuration></configuration> →**

```
<configuration>
<property>
<name>hadoop.tmp.dir</name>
<value>/app/hadoop/tmp</value>
</property>
<property>
<name>fs.default.name</name>
<value>hdfs://hadoopmaster: 54310</value>
</property>
</configuration>
```



hdfs-site.xml :

Modifier le fichier **hdfs-site.xml,** en modifiant la balise **<configuration></configuration> →**

```
<configuration>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/usr/local/hadoop_store/hdfs/namenode</value>
</property>
</configuration>
```

mapred-site.xml :

On va manipuler ce fichier comme ceux-ci au-dessus :

yarn-site.xml :

masters :

Créer le fichier masters qui contient le hostname de la machine masters :

- **cd /usr/local/hadoop/etc/hadoop**
- **sudo nano masters**

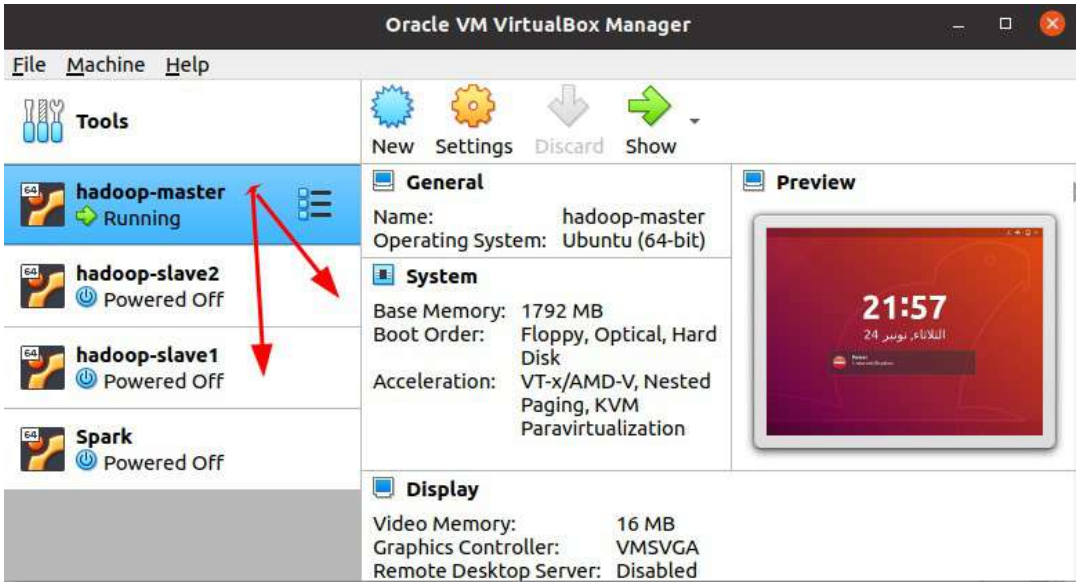
workers :

- **cd /usr/local/hadoop/etc/hadoop**
- **sudo nano workers**

Clonage de la machine hadoopmaster :

Nous allons cloner deux machines slave1 et slave2 à partir de la machine hadoopmaster, pour cela on va effectuer les étapes suivantes :

- Éteindre la machine hadoopmaster
- Bouton droit sur l'icône de la machine dans le gestionnaire virtualBox, puis cloner.
- Entrer le nom de la machine puis suivant.
- Choisir le type de clonage : clone intégral. [full]

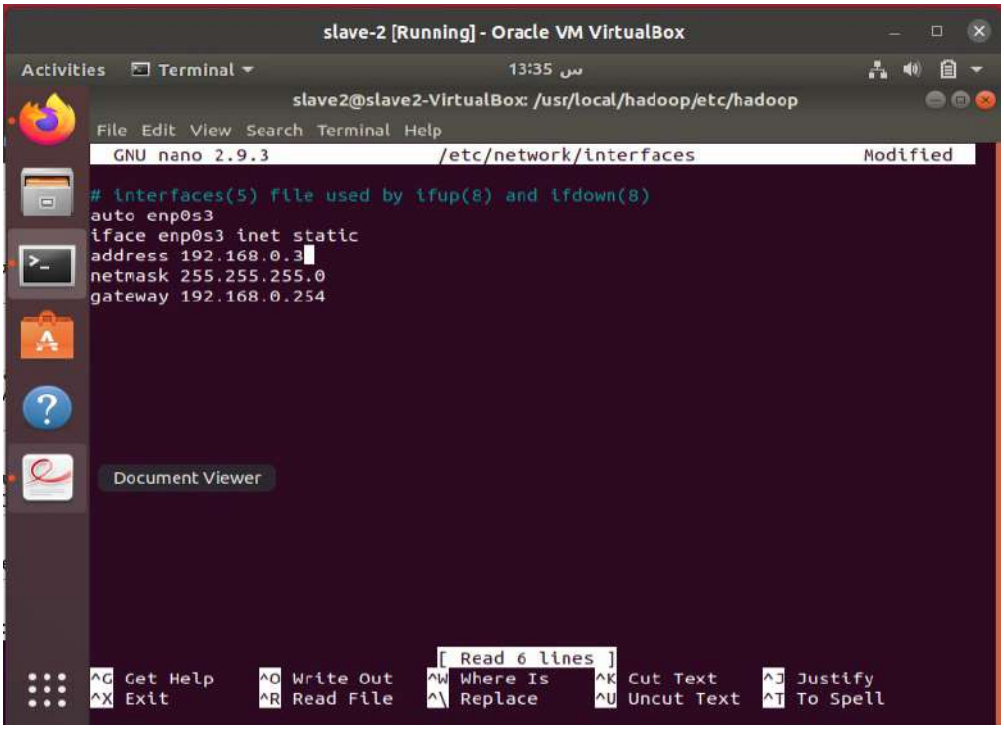


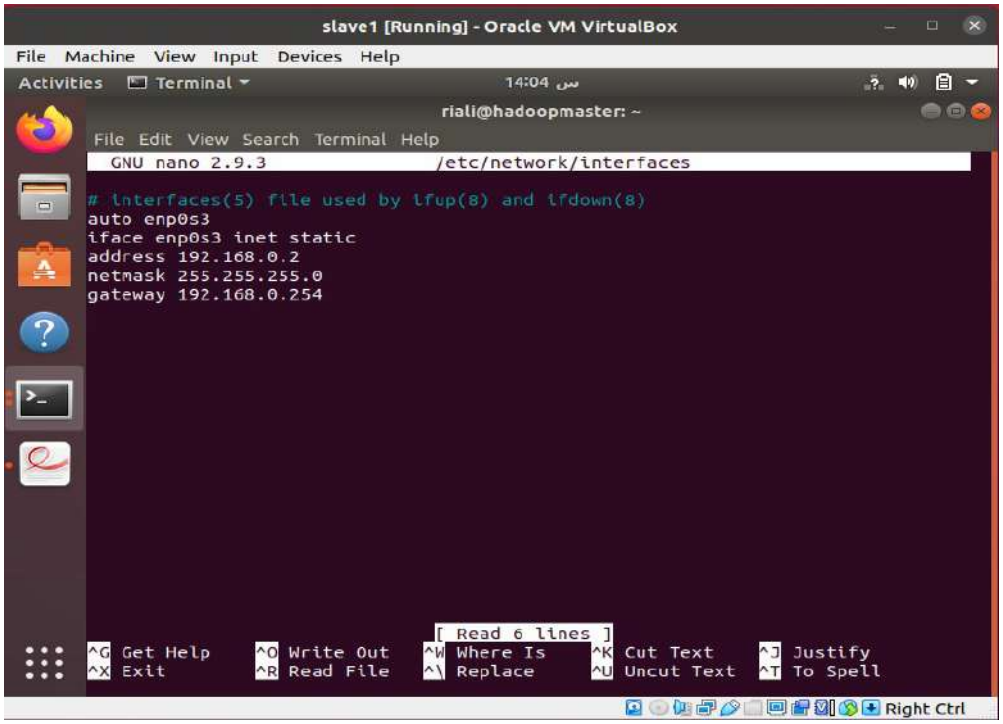
Configuration d'adresse IP fixe et de hostname :

Par modifier le fichier ***/etc/*hostname** pour chacune des 2 machines slaves, en définissant deux nouveaux **hostnames** qui sont : **slave1 → hadoop-slave1 ET slave2 → hadoop-slave2, puis**, on modefie les fichiers/etc/hosts /etc/network/interfaces, pour chacune des machines afin de leurs donner des adresses IP fixes :

```
 GNU nano 2.9.3                    /etc/hosts

127.0.0.1       localhost
127.0.1.1       slave1
auto enp0s3
iface enp0s3 inet static
address 192.168.0.2
netmask 255.255.255.0
gateway 192.168.0.254
# 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
```





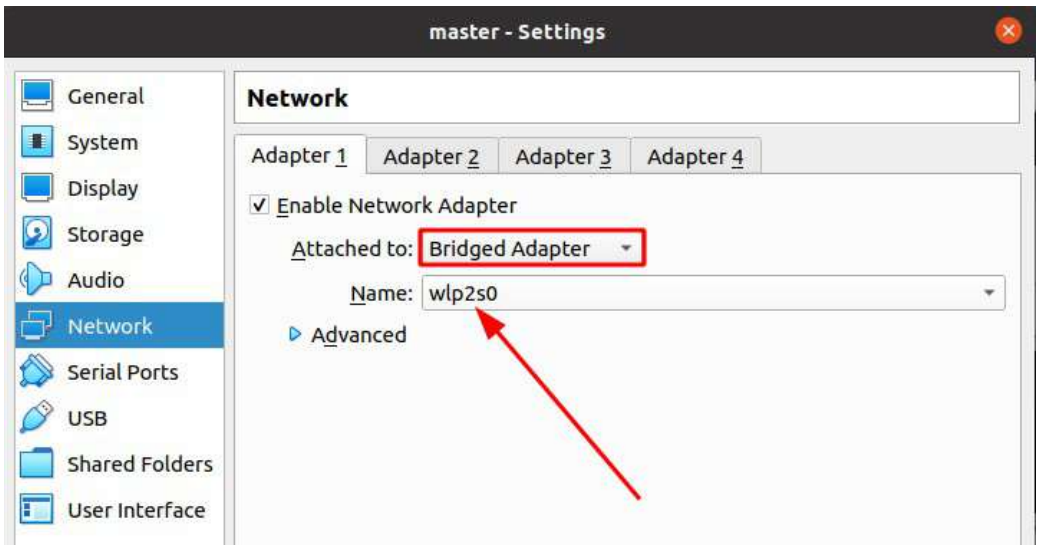
Ps : Pour prendre en compte les modifications, il faut redémarrer les deux machines slave1 et slave2: $ sudo reboot

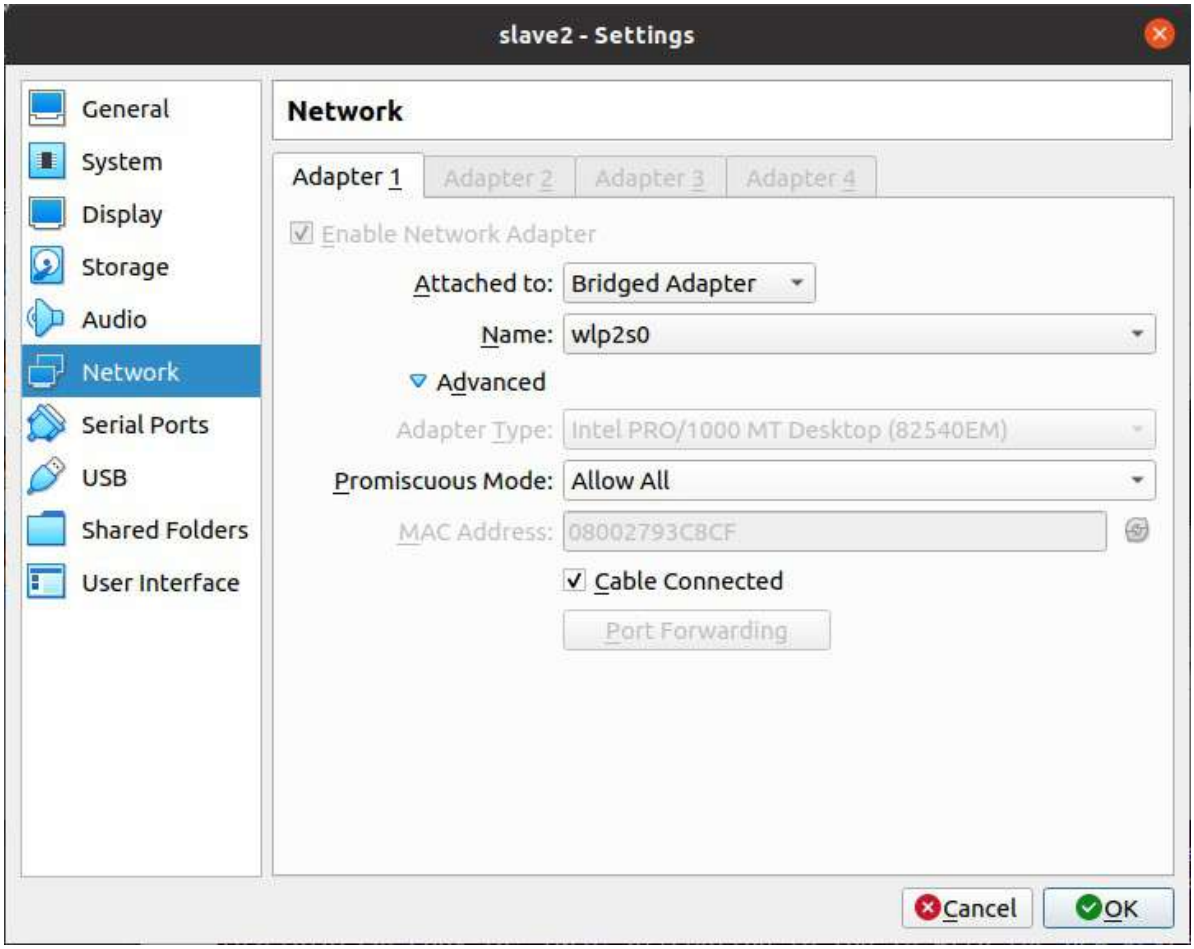
Dans la machine slave1 et slave2, supprimer les fichiers du répertoire de stockage de données créer par l'installation single node de Hadoop :

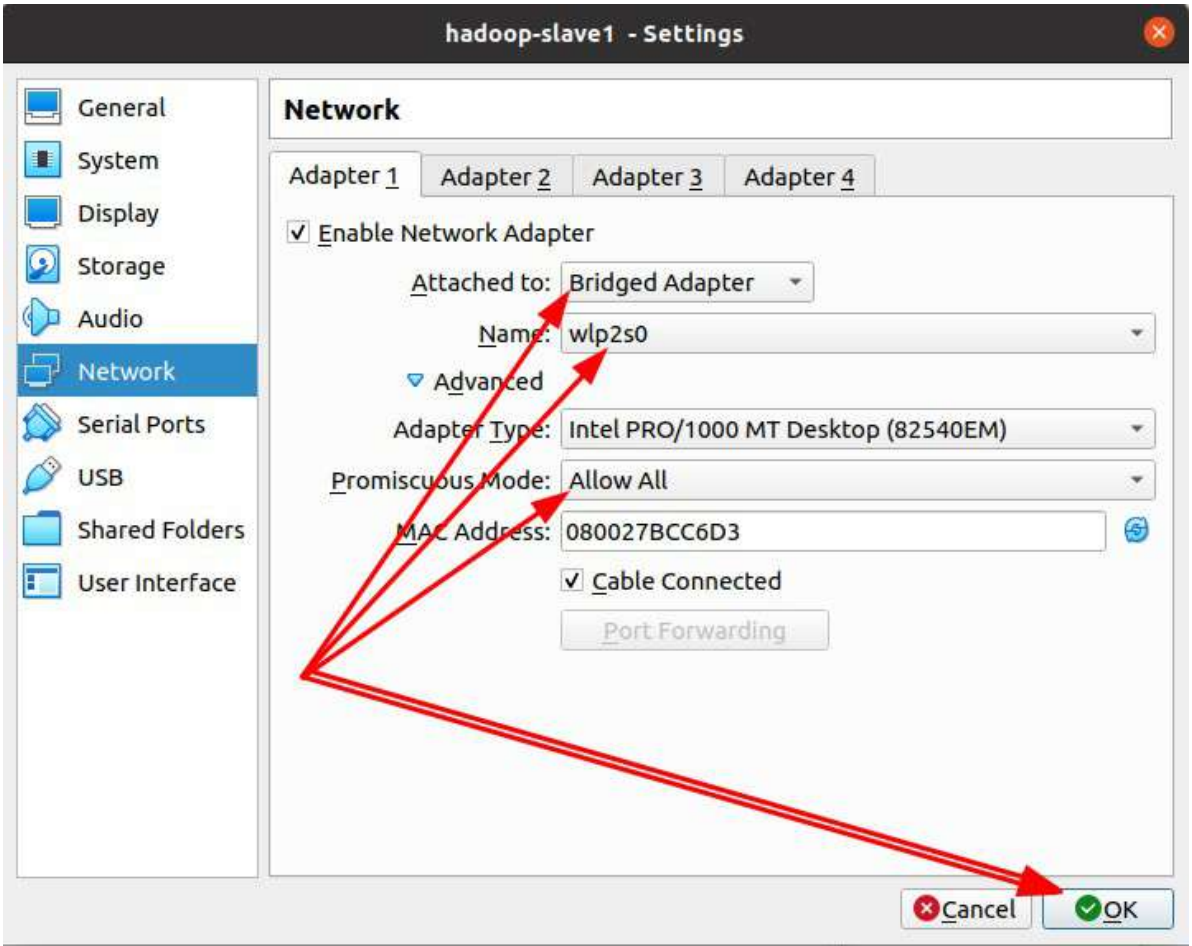
```
riali@slave2: /usr/local/hadoop_store
File  Edit  View  Search  Terminal  Help
riali@slave2:~$ cd
riali@slave2:~$ cd /usr/local/hadoop_store/
riali@slave2:/usr/local/hadoop_store$ rm -rf *
rm: cannot remove 'hdfs/namenode': Permission denied
riali@slave2:/usr/local/hadoop_store$ sudo rm -rf *
[sudo] password for riali:
riali@slave2:/usr/local/hadoop_store$ mkdir -p /usr/local/hadoop_store/hdfs/dat
anode
mkdir: cannot create directory '/usr/local/hadoop_store/hdfs': Permission denie
d
riali@slave2:/usr/local/hadoop_store$ sudo mkdir -p /usr/local/hadoop_store/hdf
s/datanode
riali@slave2:/usr/local/hadoop_store$ chown -R hduser /usr/local/hadoop_store/h
dfs/datanode
chown: changing ownership of '/usr/local/hadoop_store/hdfs/datanode': Operation
 not permitted
riali@slave2:/usr/local/hadoop_store$ sudo chown -R hduser /usr/local/hadoop_st
ore/hdfs/datanode
riali@slave2:/usr/local/hadoop_store$
```

Connexion entre les machines du cluster :

établir un réseau ponté entre les machines virtuelles :

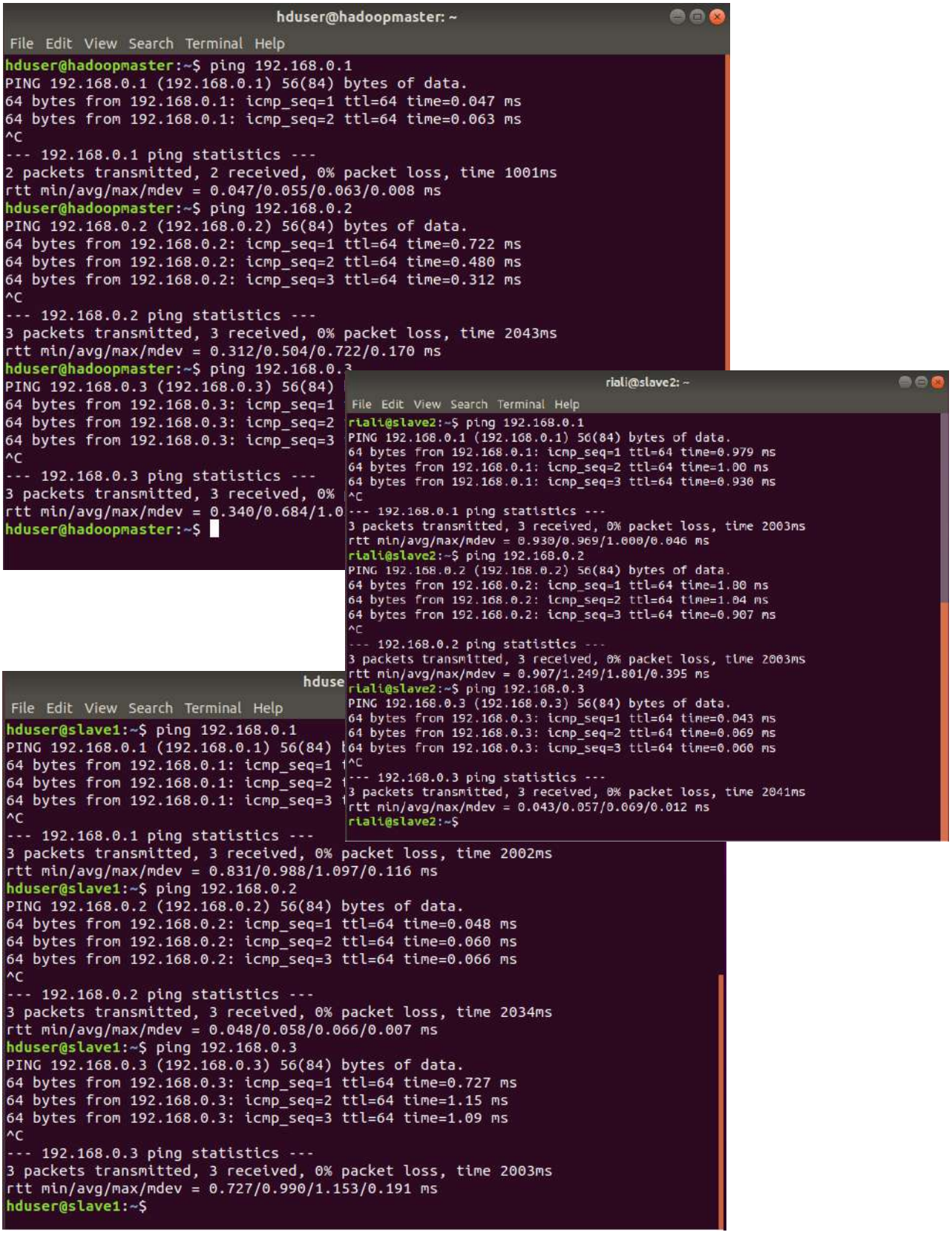



slave2 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
Network
Adapter 1
Adapter 2
Adapter 3
Adapter 4
Enable Network Adapter
Attached to: Bridged Adapter
Name: wlp2s0
Advanced
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)
Promiscuous Mode: Allow All
MAC Address: 08002793C8CF
Cable Connected
Port Forwarding
Cancel
OK


hadoop-slave1 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
Network
Adapter 1
Adapter 2
Adapter 3
Adapter 4
Enable Network Adapter
Attached to: Bridged Adapter
Name: wlp2s0
Advanced
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)
Promiscuous Mode: Allow All
MAC Address: 080027BCC6D3
Cable Connected
Port Forwarding
Cancel
OK

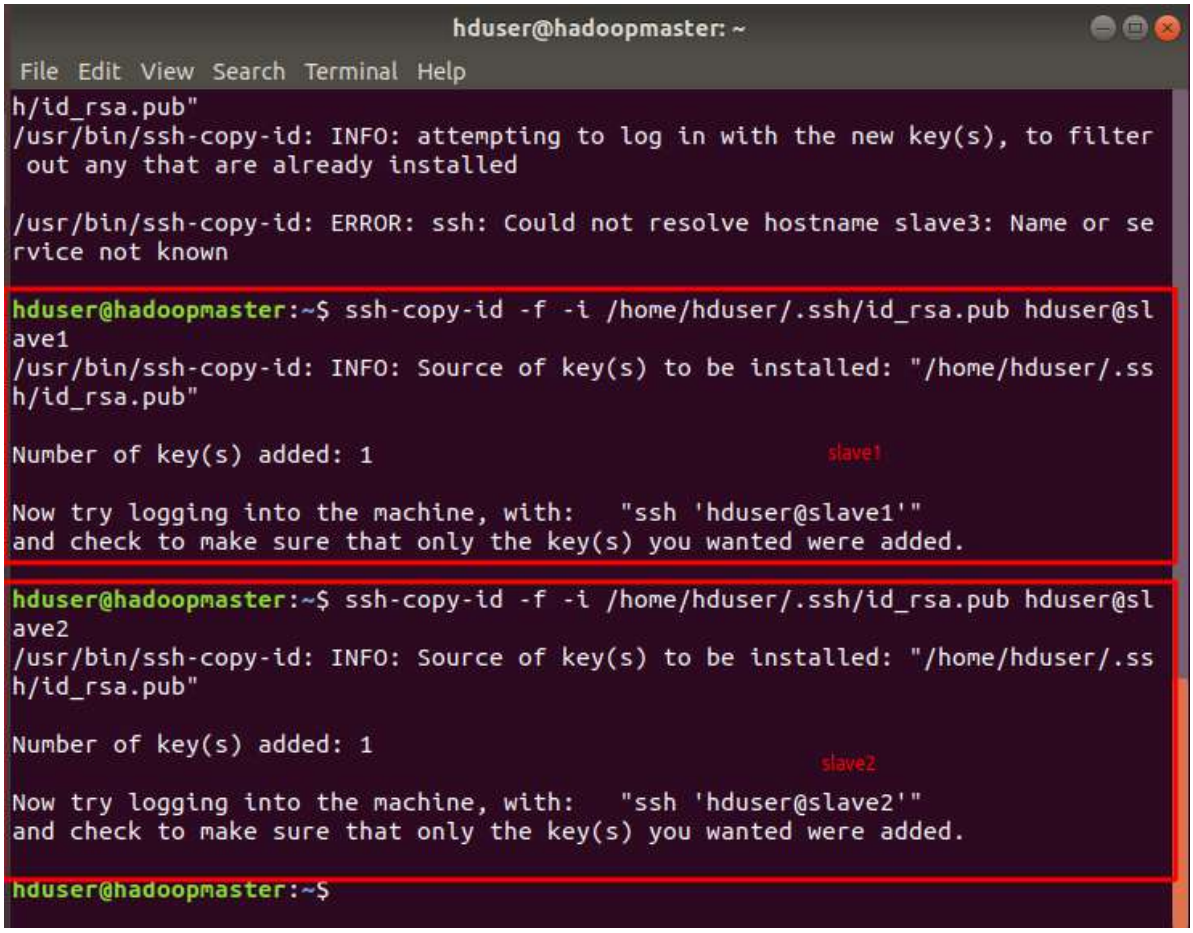
On va tester la connexion entre les machines du cluster en effectuant des pings entre les différentes machines.

Dans notre cas tout va bien !!!

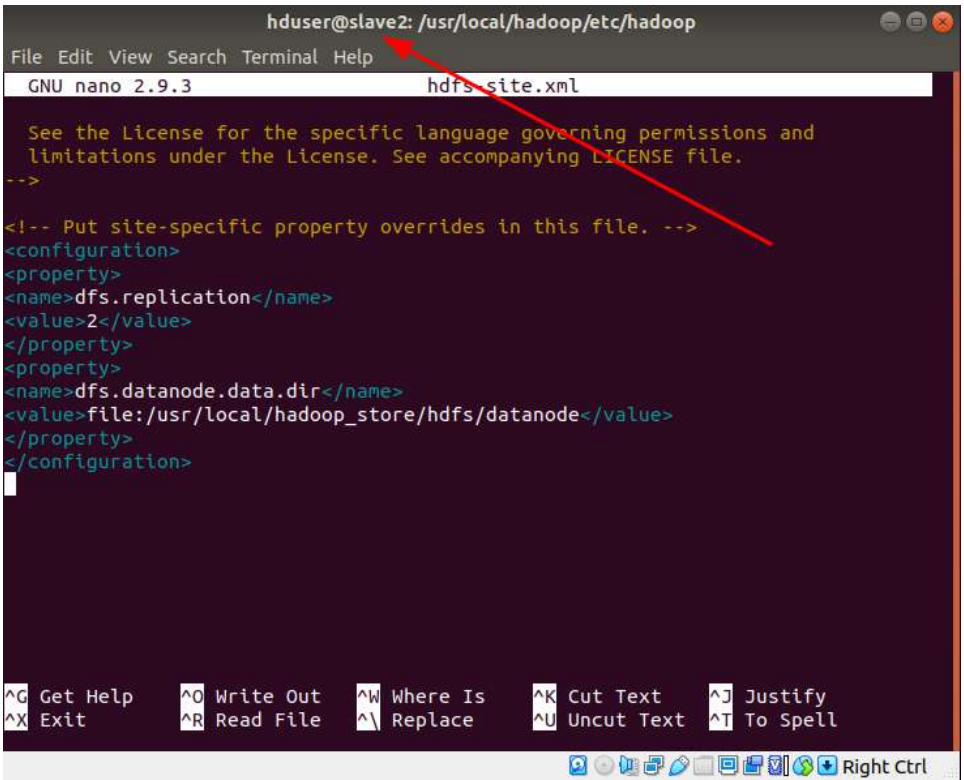


On va copier la clé ssh pour configurer un accès ssh sans mot de passe entre les machines du cluster.

- ssh-copy-id -i /home/hduser/.ssh/id_rsa.pub hduser@slave1
- ssh-copy-id -i /home/hduser/.ssh/id_rsa.pub hduser@slave2

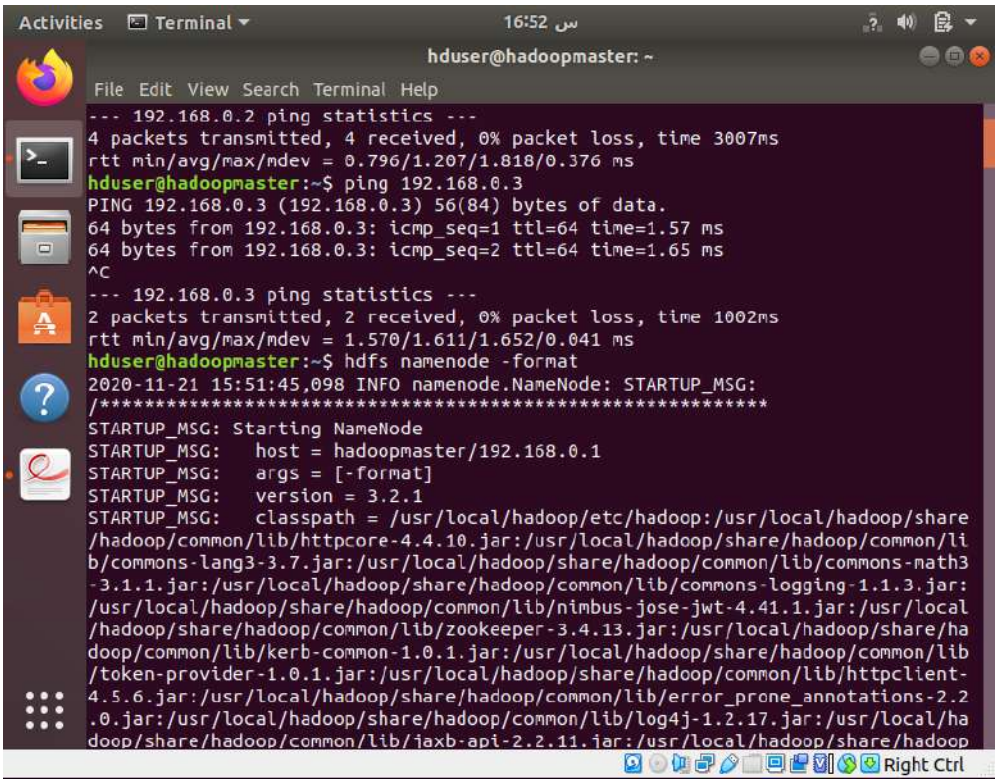


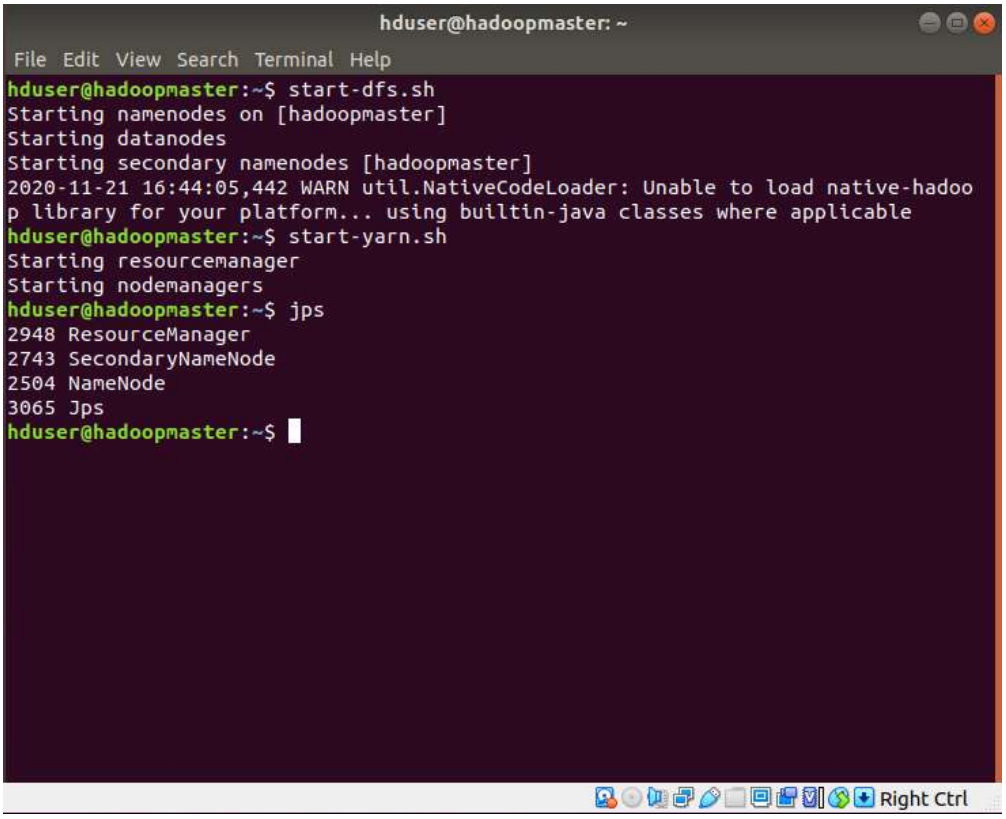
Ainsi, il faut modifier le fichier **hdfs-site.xml** de la machine virtuelle **slave1** et **slave2** comme suit :



Formater le namenode dans hadoopmaster :

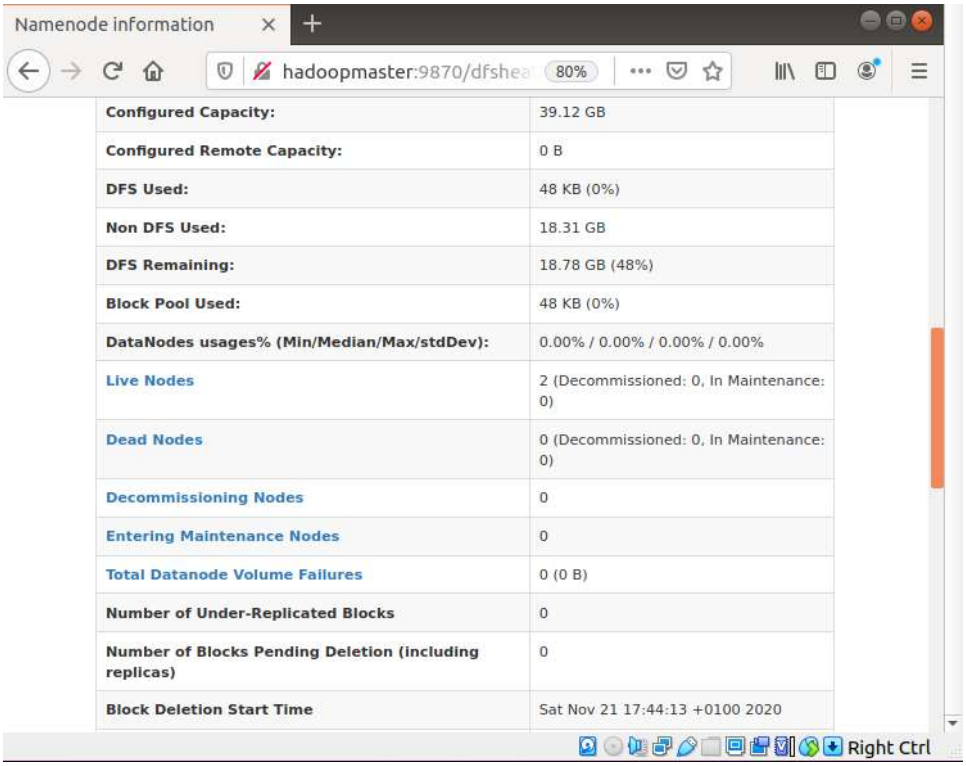
On va suivre les mêmes étapes et les commandes pour un unique nœud :

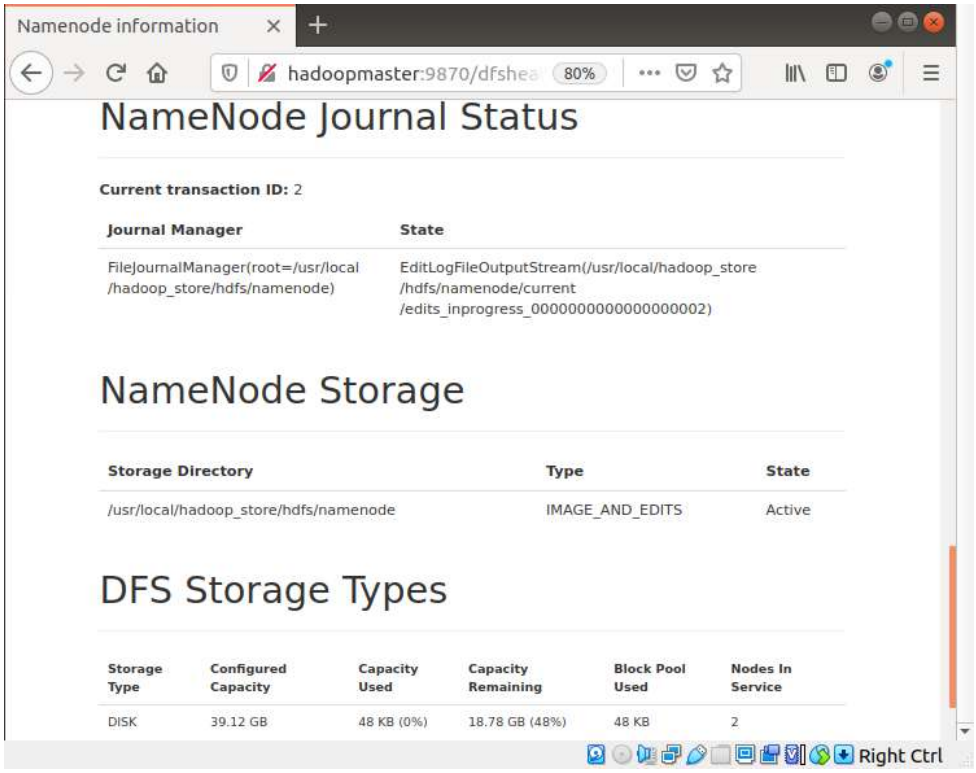


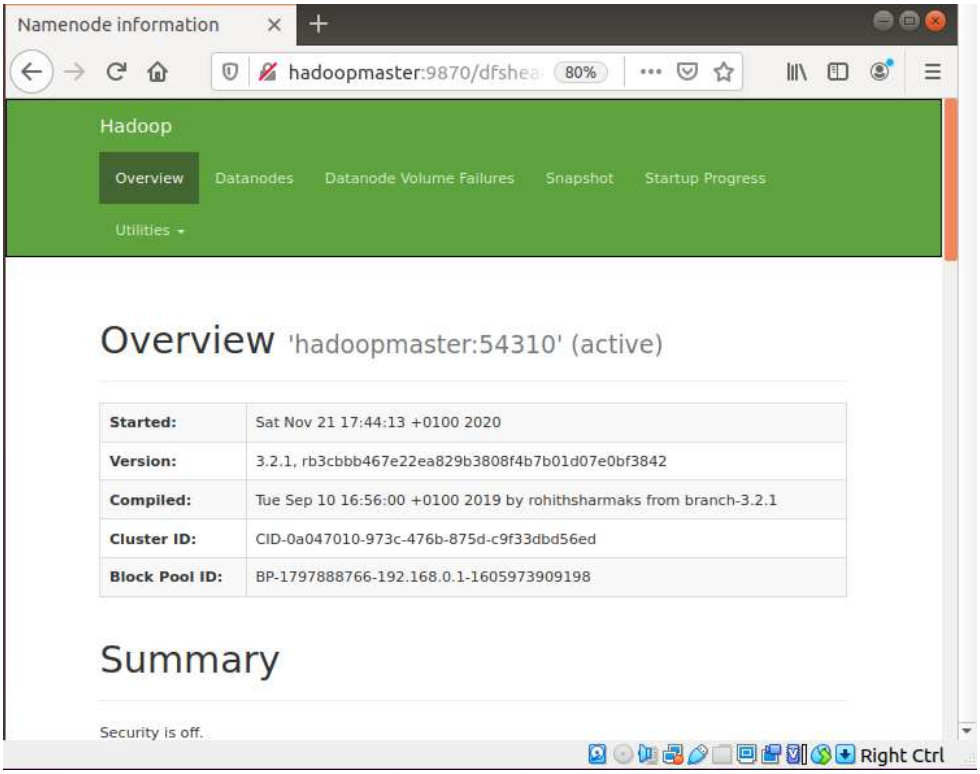


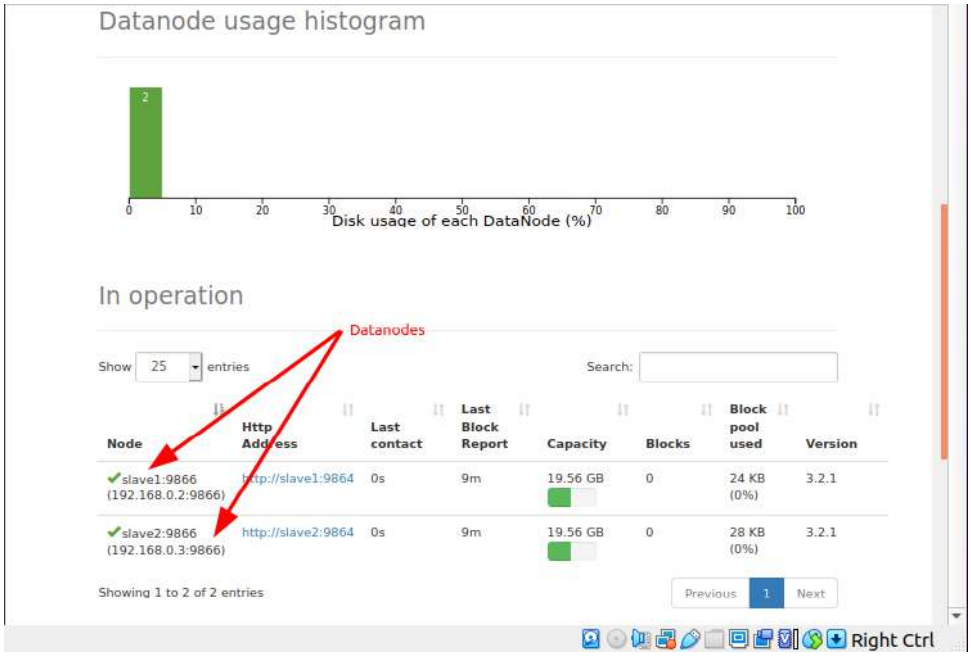
Accéder aux services de Hadoop via le navigateur :

http://hadoopmaster:9870/

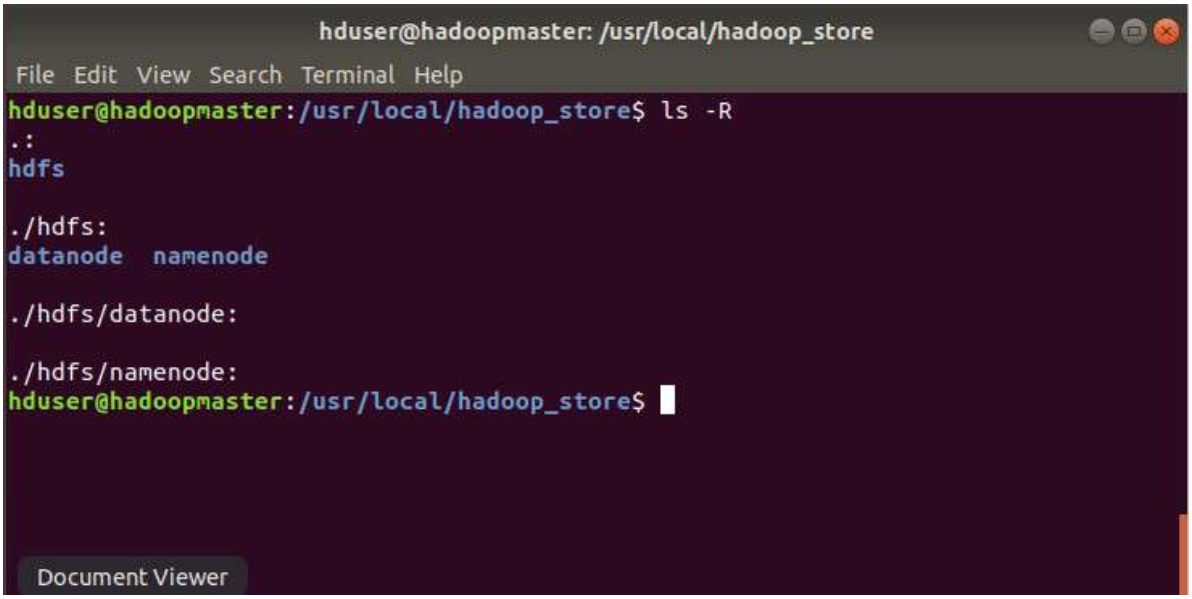





Namenode information
hadoopmaster:9870/dfshea
80%
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Overview 'hadoopmaster:54310' (active)
Started: Sat Nov 21 17:44:13 +0100 2020
Version: 3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842
Compiled: Tue Sep 10 16:56:00 +0100 2019 by rohithsharmaks from branch-3.2.1
Cluster ID: CID-0a047010-973c-476b-875d-c9f33dbd56ed
Block Pool ID: BP-1797888766-192.168.0.1-1605973909198
Summary
Security is off.
Right Ctrl


Datanode usage histogram
2
0
10
20
30
40
50
60
70
80
90
100
Disk usage of each DataNode (%)
In operation
Datanodes
Show 25 entries
Search:
Node
Http Address
Last contact
Last Block Report
Capacity
Blocks
Block pool used
Version
slave1:9866 (192.168.0.2:9866)
http://slave1:9864
0s
9m
19.56 GB
0
24 KB (0%)
3.2.1
slave2:9866 (192.168.0.3:9866)
http://slave2:9864
0s
9m
19.56 GB
0
28 KB (0%)
3.2.1
Showing 1 to 2 of 2 entries
Previous
1
Next
Right Ctrl

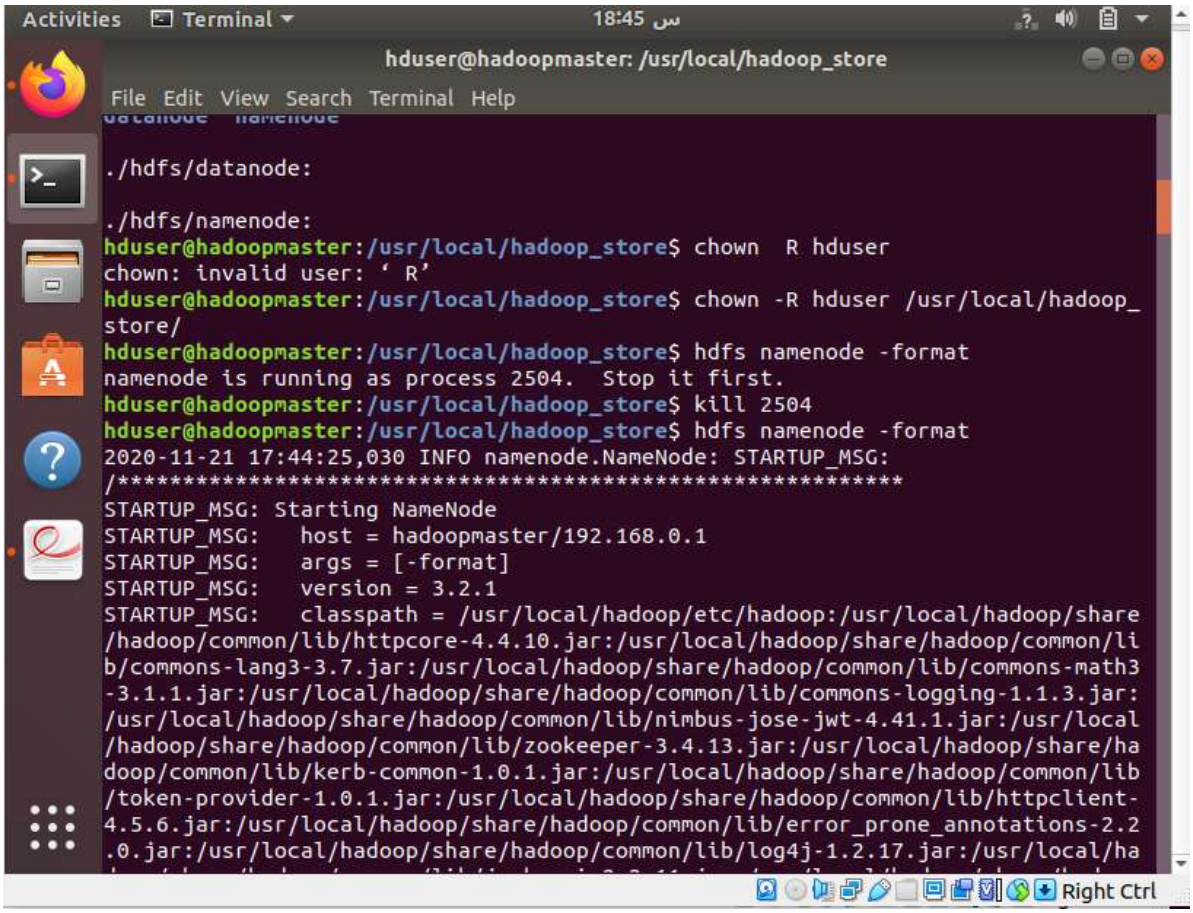
IV. Exécution d'un programme Map/Reduce dans un cluster multi-noeuds :

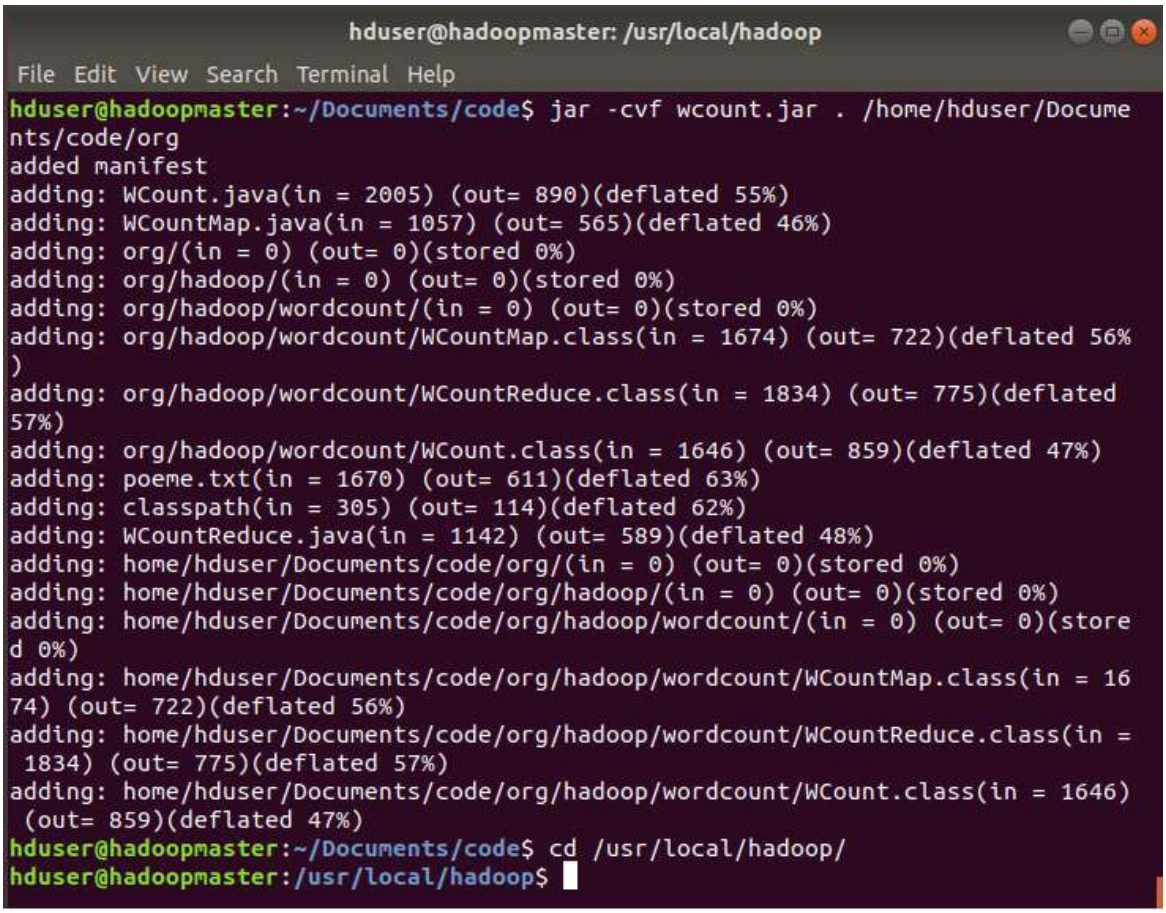


Lancer la commande : « **hdfs dfsadmin -report** » pour vérifier le bon fonctionnement de tous les nœuds du cluster ou bien accéder aux services de Hadoop via le navigateur comme c'est le cas ici :

Configured Capacity:	39.12 GB
Configured Remote Capacity:	0 B
DFS Used:	48 KB (0%)
Non DFS Used:	18.31 GB
DFS Remaining:	18.78 GB (48%)
Block Pool Used:	48 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	2 (Decommissioned: 0, In Maint 0)
Dead Nodes	0 (Decommissioned: 0, In Maint 0)
Decommissioning Nodes	0
Entering Maintenance Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0

et on va suivre les mêmes étapes citées pour l'exécution d'un programme Map/Reduce pour un seul nœud :





```
adding: org/hadoop/(in = 0) (out= 0)(stored 0%)
adding: org/hadoop/wordcount/(in = 0) (out= 0)(stored 0%)
adding: org/hadoop/wordcount/WCountMap.class(in = 1674) (out= 722)(deflated 56%
)
adding: org/hadoop/wordcount/WCountReduce.class(in = 1834) (out= 775)(deflated
57%)
adding: org/hadoop/wordcount/WCount.class(in = 1646) (out= 859)(deflated 47%)
adding: poeme.txt(in = 1670) (out= 611)(deflated 63%)
adding: classpath(in = 305) (out= 114)(deflated 62%)
adding: WCountReduce.java(in = 1142) (out= 589)(deflated 48%)
adding: home/hduser/Documents/code/org/(in = 0) (out= 0)(stored 0%)
adding: home/hduser/Documents/code/org/hadoop/(in = 0) (out= 0)(stored 0%)
adding: home/hduser/Documents/code/org/hadoop/wordcount/(in = 0) (out= 0)(store
d 0%)
adding: home/hduser/Documents/code/org/hadoop/wordcount/WCountMap.class(in = 16
74) (out= 722)(deflated 56%)
adding: home/hduser/Documents/code/org/hadoop/wordcount/WCountReduce.class(in =
 1834) (out= 775)(deflated 57%)
adding: home/hduser/Documents/code/org/hadoop/wordcount/WCount.class(in = 1646)
 (out= 859)(deflated 47%)
hduser@hadoopmaster:~/Documents/code$ cd /usr/local/hadoop/
hduser@hadoopmaster:/usr/local/hadoop$ bin/hdfs dfs -put /home/hduser/Documents
/code/poeme.txt /
2020-11-22 23:02:23,397 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
2020-11-22 23:02:28,235 INFO sasl.SaslDataTransferClient: SASL encryption trust
 check: localHostTrusted = false, remoteHostTrusted = false
hduser@hadoopmaster:/usr/local/hadoop$
```

Document Viewer

```
hduser@hadoopmaster:/usr/local/hadoop$ bin/hdfs dfs -ls /
2020-11-22 23:03:42,261 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--   2 hduser supergroup       1670 2020-11-22 23:02 /poeme.txt
hduser@hadoopmaster:/usr/local/hadoop$
```

```
2020-11-22 23:05:08,880 INFO mapred.LocalJobRunner: Finishing task: attempt_loc
al1315506290_0001_r_000000_0
2020-11-22 23:05:08,880 INFO mapred.LocalJobRunner: reduce task executor comple
te.
2020-11-22 23:05:09,770 INFO mapreduce.Job:  map 100% reduce 100%
2020-11-22 23:05:09,773 INFO mapreduce.Job: Job job_local1315506290_0001 comple
ted successfully
```

hduser@hadoopmaster: ~/Documents/code

File Edit View Search Terminal Help

```
sang     1 occurences.
sanglots         1 occurences.
sappelle         1 occurences.
se       1 occurences.
sentinelle       1 occurences.
ses      1 occurences.
seul     1 occurences.
soldats  1 occurences.
songe    1 occurences.
sont     2 occurences.
sous     1 occurences.
sur      1 occurences.
sy       1 occurences.
terre    1 occurences.
tira     1 occurences.
tombe    1 occurences.
tous     3 occurences.
trepas   1 occurences.
triste   1 occurences.
trompa   1 occurences.
un       4 occurences.
verra    1 occurences.
vie      1 occurences.
vient    1 occurences.
violoncelle      1 occurences.
vive     1 occurences.
vivra    1 occurences.
hduser@hadoopmaster:~/Documents/code$
```

Right Ctrl

FIN