


<b>Modules:</b> Framework et architectures et Big Data / BigData et architectures associées <b>Responsable de cours :</b> Leila Baccour <b>Enseignantes de TP :</b> Hana MALLEK, Maysam Chaari, Samar AKERMI	<b>Auditoire :</b> T-LSI MM /D-ADBD	AU : 2023-2024 
--	--	---

## TP4 : Exécution des jobs MapReduce sur un cluster

### 1. Objectifs

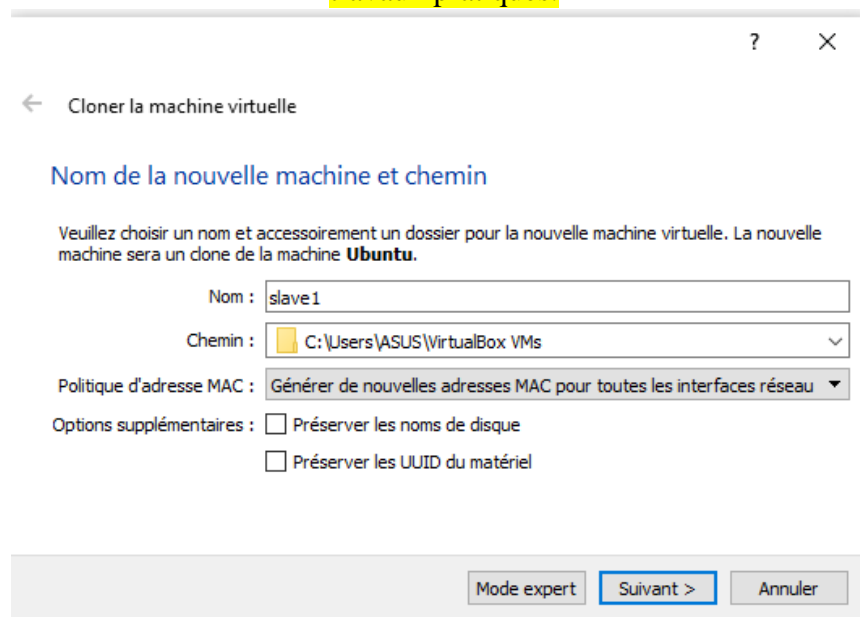
L'objectif de ce TP est de lancer les jobs MapReduce sur un cluster de machines virtuelles.

### 2. Mise en place des machines virtuelle

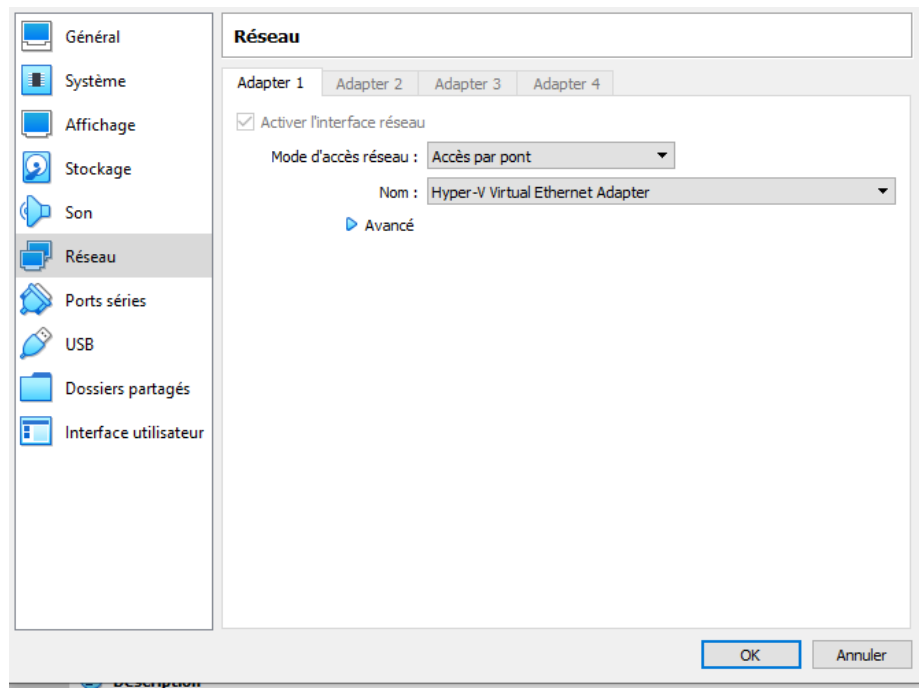
#### 2.1. Cloner notre machine virtuelle :

- Assurer que la machine virtuelle à cloner est arrêtée pour éviter toute corruption de données pendant le processus de clonage.
- Cliquer sur le bouton droit sur la machine virtuelle et choisir « cloner »
- Donner le nom « slave1 » pour la machine à créer et changer la politique d'adresse MAC à « Générer de nouvelles adresses MAC pour toutes les interfaces réseaux » et valider.
- Garder « clone intégral » coché pour créer une copie exacte de la machine virtuelle d'origine puis valider
- Attendre un peu pour avoir une nouvelle machine créée dans notre hyperviseur.
- Répéter les étapes (b,c,d,e) pour construire deux autres machine virtuelles « slave2 » et « master ».

**Remarque :** Il est nécessaire de conserver la machine d'origine pour poursuivre les autres travaux pratiques.



- Modifier le mode d'accès réseau de chaque machines créée, par « accès par pont » Afin d'attribuer des adresses IP différentes pour chaque machine puis valider.



## 2.2. Configuration de la machine Master :

- a) Lancer la machine virtuelle master
- b) Obtenir l'adresse IP de cette machine à travers la commande « **ifconfig** »  
Si la commande « ifconfig » ne fonctionne pas, il faut télécharger le package **net-tools** à travers la commande suivante : **sudo apt-get install net-tools**

```

user1@worker2: ~
Fichier Édition Affichage Rechercher Terminal Aide
user1@worker2:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 172.22.201.180  netmask 255.255.240.0  broadcast 172.22.207.255
    inet6 fe80::9792:c6a8:1330:7e86  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:db:84:16  txqueuelen 1000  (Ethernet)
    RX packets 993  bytes 1299839 (1.2 MB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 623  bytes 56055 (56.0 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Boucle locale)
    RX packets 90  bytes 8842 (8.8 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 90  bytes 8842 (8.8 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

user1@worker2:~$ 

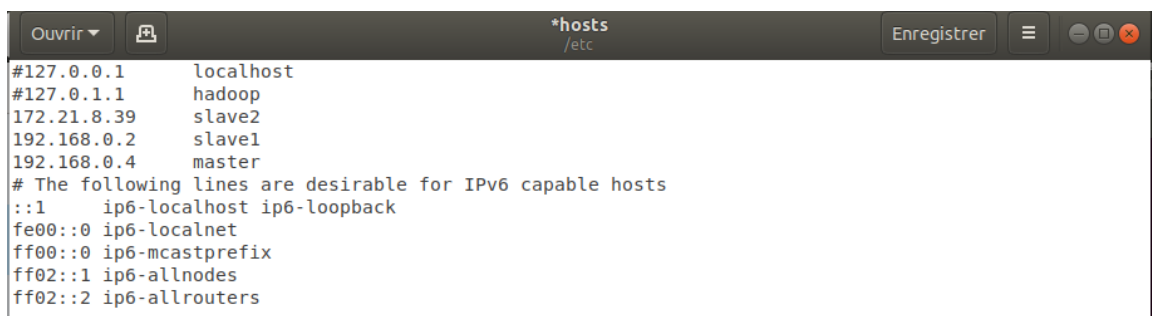
```

- a) Lancer la commande suivante **sudo gedit /etc/hostname** pour ouvrir le fichier hostname

- b) Modifier le contenu de ce fichier en ajoutant le nom « master »



- c) Répéter cette étape pour les autres machines : slave1 pour la machine « slave1 » et slave2 pour la machine « slave2 »
- d) Sur la machine master : lancer la commande suivante **sudo gedit /etc/hosts** pour ouvrir le fichier hosts
- e) Modifier le contenu de ce fichier en ajoutant les différentes adresses IP de chaque machine virtuelle dans le réseau : master, slave1 et slave2

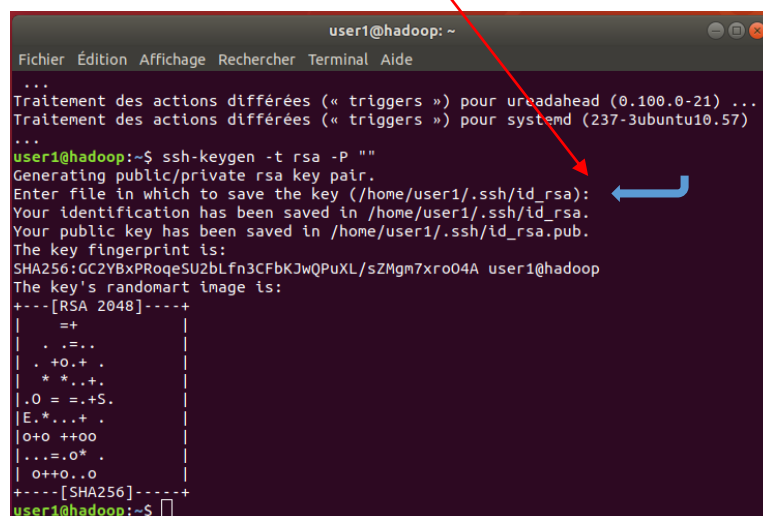


- f) Redémarrer toutes les machines
- g) Vérifier la connectivité réseau entre les différentes machines avec **ping HostName**

### 2.3. Génération des clés RSA

Afin d'établir des communications sécurisées et d'authentifier les différents nœuds dans un cluster il faut créer une paire de clés publique/privée nommée RSA.

- a) Dans chaque machine, générer les clés RSA à travers : **ssh-keygen -t rsa -P ""**
- b) Puis taper sur le bouton **entrer** pour générer une clé privée vide



- c) Copier la clé dans « `authorized_keys` » à travers la commande suivante

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

## 2.4. Echange des clés publiques entre les différentes machines

La commande **ssh-copy-id** permet de copier la clé publique d'un utilisateur sur une machine locale vers le fichier **authorized\_keys** de cet utilisateur sur une machine distante. Cette étape est essentielle pour permettre une connexion SSH sans mot de passe entre les machines.

- a) Lancer la commande suivante entre les différentes machines

```
ssh-copy-id -i /chemin/vers/clé_publicque [NomUtilisateurCible]@[hostNameCible]
```

### Exemple :

```
ssh-copy-id -i /home/user1/.ssh/id_rsa.pub user1@192.168.1.100
```

ou

```
ssh-copy-id -i /home/user1/.ssh/id_rsa.pub user1@slave1
```

- b) Vérifier la connectivité entre les différentes machines à travers la commande :

```
ssh hostName
```

### Exemple :

```
ssh user1@slave1
```

Si vous êtes connecté, vous pouvez quitter la machine distante (slave1) en utilisant la commande **"exit"**.

- c) Donner la permission d'exécution et de lecture pour les clés publiques générées :

```
chmod +r /home/user1/.ssh/authorized_keys
```

```
chmod +r /home/user1/.ssh
```

## 2.5 Editer le fichier **\$HADOOP\_INSTALL/etc/hadoop/slaves** dans toutes les machines comme suit :

**slave1**

**slave2**

- a) Modifier le fichier `core-site.xml` dans toutes les machines comme suit :

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

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

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

- b) Modifier le fichier `hdfs-site.xml` dans la machine master comme suit :

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

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

<configuration>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
<property>
<name>dfs.name.dir</name>
<value>/home/master/name</value>
</property>
</configuration>
```

- c) Modifier le fichier `hdfs-site.xml` dans les machines `slave1` et `slave2` comme suit :

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

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

<configuration>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
<property>
<name>dfs.data.dir</name>
<value>/home/user1/data</value>
</property>
</configuration>
```

## 2.6 Exécuter un job MapReduce

- a) Formater le Namenode dans la machine master : **hdfs namenode -format**
- b) Lancer Hadoop sur la machine master
- c) Vérifier l'exécution de tous les services de Hadoop sur la machine master avec la commande **jps**.
- d) Exécuter le job MapReduce de l'exercice 1 du TP3 sur la machine master à travers la commande **hs** ou la commande suivante :  
**hadoop jar \$HAD\_INS/share/hadoop/tools/lib/hadoop-streaming-2.10.2.jar \**  
**-file /.../.../mapper.py -mapper /.../.../mapper.py \**  
**-file /.../.../reducer.py -reducer /.../.../reducer.py \**  
**-input /<cheminHDFS>/Book.txt -output /<cheminHDFS>/Resultat**
- e) Visualiser liste des nodes actives
- f) Visualiser le résultat final