

## Technologies « Big Data »

### TP #1 – Mise en place d'un cluster (virtuel) Hadoop

**Objectif :** Ce TP est destiné à mettre en place un cluster Hadoop « virtuel » composé de 2 machines Linux. Pour atteindre cet objectif trois grandes étapes sont à réaliser :

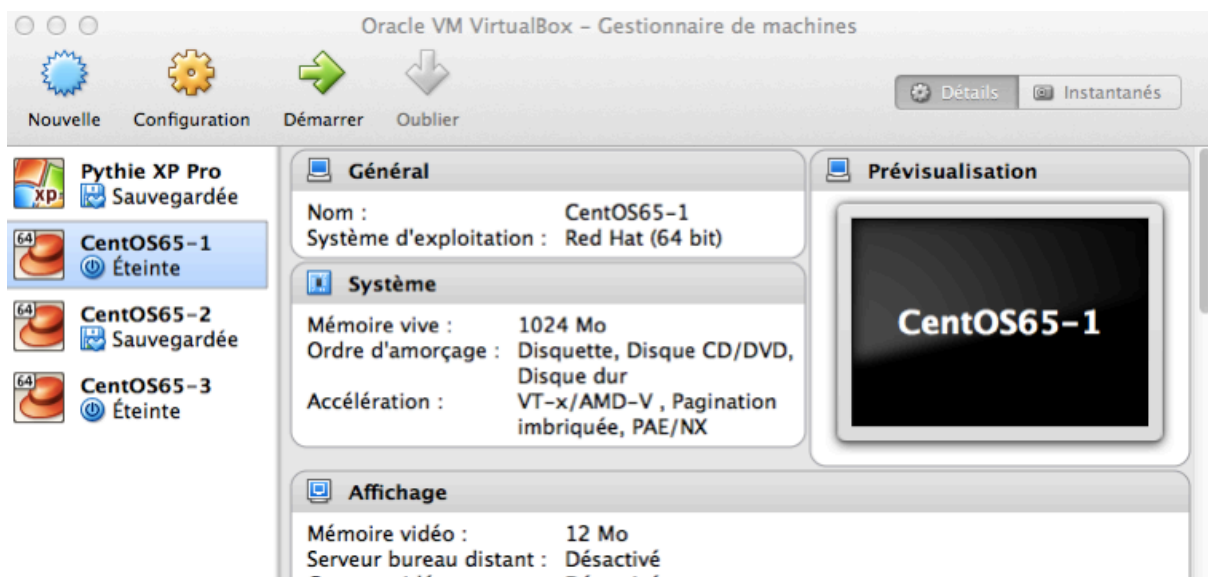
- L'importation d'une machine « virtuelle » Linux CentOS,
- le clonage de la machine « virtuelle » et la configuration des deux machines obtenues pour former un cluster Hadoop virtuel.

**Pré-requis :** Le logiciel Oracle VM VirtualBox installé

#### Pré-requis : Outil de gestion de machines « virtuelles » Oracle VM VirtualBox

---

Oracle VM VirtualBox est un logiciel de virtualisation disponible en tant qu'hôte sur plusieurs systèmes d'exploitation, notamment Windows, Linux 32 et 64 bits et Mac OS X. Il supporte en tant qu'invité entre autres plusieurs systèmes Windows, Linux et MacOS X. Pour certains systèmes, tels que Windows, il est bien entendu nécessaire de posséder une licence pour installer une machine virtuelle invitée fonctionnant sous ce système, comme pour une machine réelle.



#### Étape 1 : Importation d'une machine « virtuelle » Linux CentOS

---

L'installation d'une machine virtuelle s'effectue généralement à partir d'une image ISO d'un DVD d'installation du système d'exploitation choisi. Dans notre cas, pour simplifier cette partie nous nous contenterons de récupérer une machine virtuelle toute prête.

- 1) Récupérer le fichier d'importation CentOS7.ova contenant la machine virtuelle.
- 2) Lancer VirtualBox et effectuer une importation du fichier CentOS7.ova (menu Fichier/importer un appareil virtuel)
- 3) Démarrer la nouvelle machine virtuelle créée, se connecter sous le compte hduser
- 4) Démarrer ensuite les processus de l'environnement Hadoop :

- a. Exécuter les commandes :

```
start-dfs.sh
start-yarn.sh
```

(et éventuellement `mr-jobhistory-daemon.sh start historyserver`)

- b. Vérifier que les processus ont bien démarré en exécutant la commande : `jps`

La liste des processus est : `NodeManager, NameNode, DataNode, SecondaryNameNode, ResourceManager, Jps`

Remarque : pour arrêter les processus Hadoop, il faut exécuter les commandes `stop-dfs.sh` et `stop-yarn.sh` (et éventuellement `mr-jobhistory-daemon.sh stop historyserver`). La commande `jps` doit alors seulement afficher comme liste le processus `Jps`

- 5) Essayer d'exécuter un programme qui fait seulement un traitement parallèle sans manipulation de fichier : `hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.X.Y.jar pi 2 5`

- 6) Une fois ce premier exemple validé, il reste à passer à la manipulation de fichier dans le système hdfs. Il faut distinguer la manipulation dans le système de fichier classique au travers des commandes Linux habituelles (`ls`, `cp`, `mv...`) et la manipulation de fichiers dans le systèmes hdfs au travers de commandes qui commencent systématiquement par le mot-clé 'hdfs dfs'.

- a. Créer les répertoires où seront manipulés les fichiers hdfs en exécutant les commandes :

```
hdfs dfs -mkdir /user
hdfs dfs -mkdir /user/hduser
```

- b. Lister le contenu du répertoire hdfs hduser en exécutant la commande :

```
hdfs dfs -ls /user/hduser
```

- c. Copier des fichiers textes dans le répertoire utilisateur hdfs créé en exécutant la commande :

```
hdfs dfs -put /usr/local/hadoop/*.txt /user/hduser
```

Vérifier que les (trois) fichiers ont bien été copiés dans le répertoire hdfs

- d. Exécuter un programme qui fait un traitement parallèle avec manipulation de fichier :

```
hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.X.Y.jar wordcount *.txt testoutput
```

Vérifier que le résultat généré par ce programme avec la commande :

```
hdfs dfs -cat testoutput/*
```

Supprimer le résultat généré avec la commande : `hdfs dfs -rm -R testoutput`

Remarque : Il est possible de surveiller le cluster et l'exécution des programmes (jobs) à l'aide d'un navigateur, au travers d'interfaces Web : <http://centos7-1:8088/cluster>, <http://centos7-1:50070/dfshealth.jsp>, et <http://centos7-1:19888/jobhistory>.

- 7) Si tout s'est bien passé jusqu'ici, il reste à arrêter les services hadoop (`stop-dfs.sh` et `stop-yarn.sh` (et éventuellement `mr-jobhistory-daemon.sh stop historyserver`) et vérifier avec `jps`) et tenter la prochaine étape de mise en place d'un cluster hadoop basée sur la clonage de cette première machine virtuelle fonctionnelle. Pour cela, éteindre cette machine virtuelle centos7-1.

## Étape 2 : Clonage de machines virtuelles et configuration du cluster Hadoop

---

- 1) Pour créer un cluster Hadoop, il faut au préalable faire en sorte que les machines virtuelles puissent communiquer. Pour cela, cliquer sur la machine virtuelle centos7-1 dans la liste des VM du logiciel VirtualBox, aller dans le menu Machine et sélectionner l'option Configuration, onglet Réseau, onglet Carte 2, cocher la case 'Activer la carte réseau', choisir le mode d'accès réseau 'Accès par pont' (nom de réseau : bridge0). Dans la configuration 'Avancé', sélectionner l'option 'Autoriser les VMs' pour le mode 'Promiscuité' puis OK.
- 2) Pour cloner la machine virtuelle qui vient d'être créée, cliquer sur la machine virtuelle centos7-1 dans la liste des VM du logiciel Virtual Box, aller dans le menu Machine et sélectionner l'option Cloner. Cocher la réinitialisation des adresses MAC et nommer la nouvelle machine virtuelle centos7-2 (clone intégral).
- 3) Démarrer les deux machines virtuelles centos7-1 et centos7-2. Se connecter en 'root' chacune des machines. Modifier le nom de la machine centos7-2 afin qu'elle s'appelle réellement centos7-2 (i.e., modifier le fichier hostname). Vérifier dans le fichier /etc/sysconfig/network-scripts/ifcfg-enp0s3 que l'adresse matérielle réseau (HWADDR) correspond à celle donnée au niveau de la VM pour la carte du réseau à accès par pont (Adresse MAC). Par défaut, les machines sont configurées avec attribution dynamique des adresses IP (i.e., DHCP) ce qui pose problème pour notre réseau à accès par pont.

Une solution est d'utiliser des adresses IP fixes (par exemple, 10.0.3.20 pour centos7-1 et 10.0.3.21 pour centos7-2). Pour cela, sur chaque machine il faut éditer le fichier /etc/sysconfig/network-script/ifcfg-enp0s3 pour avoir les lignes suivantes, par exemple pour centos7-1 :

```
BOOTPROTO="static"
IPADDR=10.0.3.20
BROADCAST=10.0.3.255
NETMASK=255.255.255.0
NETWORK=10.0.3.0
```

Sur chaque machine, éditer le fichier /etc/hosts et avoir les lignes :

```
10.0.3.20 centos7-1
10.0.3.21 centos7-2
```

- 4) Redémarrer les deux machines virtuelles. Vérifier les adresses IP des deux machines (qui doivent être différentes, normalement 10.0.3.20 pour centos7-1 et 10.0.3.21 pour centos7-2) avec la commande : *ifconfig* (regarder carte enp0s3).
- 5) Pour que l'utilisateur hduser puisse se connecter à chaque machine du cluster virtuel sans mot de passe, il faut configurer les connexions ssh sur les machines de manière adéquate. Pour cela, se connecter sur chaque machine virtuelle en tant qu'utilisateur hduser. Sur la machine centos7-1, exécuter les commandes suivantes :

```
cp ~/.ssh/id_rsa.pub ~/.ssh/id_rsa_centos7-1.pub
scp ~/.ssh/id_rsa_centos7-1.pub hduser@centos7-2 ~/.ssh
```

Exécuter ces commandes de manière analogue sur centos7-2.

Sur chaque machine virtuelle, exécuter les commandes suivantes :

```
cat ~/.ssh/id_rsa_centos7-*.pub >> ~/.ssh/authorized_keys
chmod 640 ~/.ssh/authorized_keys
chmod 700 ~/.ssh
```

Vérifier sur chaque machine que la connexion ne nécessite plus de question, en exécutant simplement, par exemple sur centos7-1, `ssh centos7-2`.

- 6) Il reste maintenant à configurer le cluster virtuel Hadoop qui sera constitué dans notre cas de deux machines. Un cluster Hadoop est composé d'une machine Maître (master) et de machines Esclave (slave). Une des deux machines de notre cluster doit donc être choisie comme Maître (master). Prenons par exemple, centos7-1 comme 'master', centos7-1 sera également esclave et centos7-2 sera uniquement esclave.

Sur le master (centos7-1), éditer le fichier `/usr/local/hadoop/etc/hadoop/slaves` afin d'avoir les deux lignes suivantes :

```
centos7-1
centos7-2
```

Désactiver le pare-feu (firewall) sur chaque machine, à l'aide des commandes : `sudo systemctl stop firewalld` (arrêt du pare-feu) puis `sudo systemctl disable firewalld` (désactivation du pare-feu). Les pare-feux peuvent bloquer les communications de certains services hadoop/yarn et entraîner un dysfonctionnement du cluster. Arrêter les pare-feu est une solution un peu brutale, une solution plus élégante serait d'ajouter des règles aux pare-feu pour ne pas bloquer les ports nécessaires à la communication entre les services hadoop/yarn.

Sur toutes les machines, les fichiers `/usr/local/hadoop/etc/hadoop/*-site.xml` définis pour centos7-1 doivent être identiques et indiquer comme machine le master (centos7-1) ce qui est le cas pour notre slave centos7-2 puisqu'il a été créé en clonant le 'futur' master centos7-1.

Sur toutes les machines, exécuter les commandes suivantes pour effacer des vestiges du cluster mono-nœud (single node) :

```
rm -r /app/hadoop/tmp/*
rm -r /home/hduser/mydata/hdfs/namenode/*
rm -r /home/hduser/mydata/hdfs/datanode/*
```

Sur le master (centos7-1), formater le namenode (comme précédemment). Démarrer les services hadoop (dfs et yarn, comme précédemment). Vérifier que les services ResourceManager, DataNode, NameNode, NodeManager ont bien démarré sur la machine master et vérifier que les services NodeManager et DataNode ont bien démarré sur la machine esclave.

Remarque : Il est possible d'étendre le cluster en ajoutant de nouvelles machines esclaves, en suivant les mêmes principes (fichier 'slaves' sur le master, fichiers \*-site.xml...).

- 7) Pour vérifier, le bon fonctionnement du nouveau cluster, ré-exécuter les deux exemples précédemment utilisés avec le cluster mono-nœud (cf. étape 1).