Hadoop 2.6.5 Installing on Ubuntu 16.04

(Single-Node Cluster)

Hadoop 2.6.5 on Ubuntu 16.04

Dans ce chapitre, nous allons installer un cluster Hadoop dans un seul noeud, basé sur le système de fichiers distribués Hadoop (HDFS) sur Ubuntu 16.04 (mode pseudo-distribué).

- **Installing Java**

```
userhadoop@Hadoop:~$ java -version
userhadoop@Hadoop:~$ sudo apt-get update
userhadoop@Hadoop:~$ sudo apt-get install default-jdk

userhadoop@Hadoop:~$ java -version
```

- **Ajout d'un utilisateur Hadoop**

Pour voir la liste des groupes

```
userhadoop@Hadoop:~$ cat /etc/group ou userhadoop@Hadoop:~$ cat /etc/passwd

userhadoop@Hadoop:~$ sudo addgroup hadoop

userhadoop@Hadoop:~$ sudo adduser --ingroup hadoop hduser
```

Mot de passe : **hduser**

```
Nous pouvons vérifier si nous avons bien crée le groupe hadoop et
l'utilisateur hduser avec
userhadoop@Hadoop:~$ groups hduser
```

- **Installing SSH**

```
SSH a deux composants principaux:
ssh: La commande utilisée pour se connecter aux machines distantes - le
client.
sshd: le démon qui s'exécute sur le serveur et permet aux clients de se
connecter au serveur.
```

```
ssh est pré-activé sur Linux, mais pour démarrer le démon sshd, nous devons
d'abord l'installer. Utilisez cette commande pour l'installer :
userhadoop@Hadoop:~$ sudo apt-get install ssh
```

```
Pour vérifier si l'installation s'est bien déroulée, il faut avoir le
résultat des commandes which comme suit :
userhadoop@Hadoop:~$ which ssh
/usr/bin/ssh
```

```
userhadoop@Hadoop:~$ which sshd
/usr/sbin/sshd
```

- **Create and Setup SSH Certificates**

Hadoop a besoin d'un accès SSH pour gérer ses nœuds, c'est-à-dire les ordinateurs distants et notre ordinateur local. Pour notre configuration à nœud unique de Hadoop, nous devons donc configurer l'accès SSH à localhost.

Donc, nous devons avoir SSH opérationnel sur notre machine et le configurer pour permettre l'authentification par **clé publique SSH.**

Hadoop utilise SSH (pour accéder à ses nœuds), ce qui obligerait normalement l'utilisateur à saisir un mot de passe. Toutefois, cette exigence peut être éliminée en créant et en configurant des **certificats SSH** à l'aide des commandes suivantes.
(Si vous êtes invité à entrer un nom de fichier, laissez-le vide et appuyez sur la touche Entrée pour continuer.)

```
userhadoop@Hadoop:~$ su hduser
hduser@Hadoop:/home/userhadoop$ mkdir ~/.ssh
hduser@Hadoop:/home/userhadoop$ chmod 700 ~/.ssh
hduser@Hadoop:/home/userhadoop$ ssh-keygen -t rsa
```

Mot de passe passphrase : **~~had~~oop**

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hduser/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hduser/.ssh/id_rsa.
Your public key has been saved in /home/hduser/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:86rstj/o/VOwxAyL/Ux1wFj9yzJpMOENXC+ccajcVZg hduser@Hadoop
The key's randomart image is:
+---[RSA 2048]----+
|          .++=.+o|
|         . .++.E. |
|        o =o.*=.o |
|       . o ** o. .|
|         S= oo o .|
|          o+ .= o |
|         .  ... o  |
|      .o....      |
|      +*++o..     |
+----[SHA256]-----+
```

La commande ci-dessous ajoute la clé nouvellement créée à la liste des clés autorisées afin que Hadoop puisse utiliser ssh sans demander de mot de passe.

```
hduser@Hadoop:/home/userhadoop$ cat $HOME/.ssh/id_rsa.pub >>
$HOME/.ssh/authorized_keys
```

Nous pouvons vérifier si ssh fonctionne:

```
hduser@Hadoop:/home/userhadoop$ ssh localhost
```

- **Install Hadoop**

```
hduser@Hadoop:~$ wget https://archive.apache.org/dist/hadoop/core/hadoop-2.6.5/hadoop-2.6.5.tar.gz

hduser@Hadoop:~$ tar xvzf hadoop-2.6.5.tar.gz
```

Déplacer l'installation Hadoop dans le répertoire /usr/local/hadoop

```
hduser@Hadoop:~$ sudo mv hadoop-2.6.5/ /usr/local/hadoop
```

Nous pouvons vérifier à nouveau si hduser n'est pas dans le groupe sudo:

```
hduser@Hadoop:~$ sudo -v
```

Cela peut être résolu en vous connectant en tant qu'utilisateur root, puis ajoutez hduser au groupe sudo:

```
hduser@Hadoop:~$ su userhadoop
Mot de passe :

userhadoop@Hadoop:/home/hduser$ sudo adduser hduser sudo
```

Maintenant, l'utilisateur hduser a le privilège root, nous pouvons déplacer l'installation de Hadoop dans le répertoire /usr/local/hadoop sans problème:

```
userhadoop@Hadoop:/home/hduser$ exit
hduser@Hadoop:~$ sudo mv hadoop-2.6.5/ /usr/local/hadoop
hduser@Hadoop:~$ sudo chown -R hduser:hadoop /usr/local/hadoop
```

- **Configuration des fichiers de configuration**

Les fichiers suivants doivent être modifiés pour compléter la configuration de Hadoop:

1. ~/.bashrc
2. /usr/local/hadoop/etc/hadoop/hadoop-env.sh
3. /usr/local/hadoop/etc/hadoop/core-site.xml
4. /usr/local/hadoop/etc/hadoop/mapred-site.xml.template
5. /usr/local/hadoop/etc/hadoop/hdfs-site.xml

1. ~/.bashrc:

Avant de modifier le fichier .bashrc dans le répertoire de base de hduser, vous devez rechercher le chemin d'installation de Java pour définir la variable d'environnement JAVA_HOME à l'aide de la commande suivante:

```
update-alternatives --config java
There is only one alternative in link group java (providing /usr/bin/java): 
/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
Nothing to configure.
```

Notez que JAVA_HOME doit être défini comme chemin juste avant '... / bin /':

```
javac -version
javac 1.8.0_111

which javac
/usr/bin/javac

readlink -f /usr/bin/javac
/usr/lib/jvm/java-8-openjdk-amd64/bin/javac
```

Nous pouvons maintenant ajouter ce qui suit à la fin de ~ / .bashrc:

```
hduser@Hadoop:~$ vi ~/.bashrc

#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
#HADOOP VARIABLES END

Sourcer le fichier ~/.bashrc
hduser@Hadoop:~$ source ~/.bashrc
```

2. /usr/local/hadoop/etc/hadoop/hadoop-env.sh

Nous devons définir JAVA_HOME en modifiant le fichier hadoop-env.sh.

```
hduser@Hadoop:~$ vi /usr/local/hadoop/etc/hadoop/hadoop-env.sh
```

Ajouter la ligne suivante :

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

L'ajout de l'instruction ci-dessus dans le fichier hadoop-env.sh garantit que la valeur de la variable JAVA_HOME sera disponible pour Hadoop à chaque démarrage.

3. /usr/local/hadoop/etc/hadoop/core-site.xml

Le fichier /usr/local/hadoop/etc/hadoop/core-site.xml contient les propriétés de configuration utilisées par Hadoop lors du démarrage.
Ce fichier peut être utilisé pour remplacer les paramètres par défaut définis par Hadoop.

```
hduser@Hadoop:~$ sudo mkdir -p /app/hadoop/tmp
hduser@Hadoop:~$ sudo chown hduser:hadoop /app/hadoop/tmp
```

Ouvrez le fichier et entrez les informations suivantes entre les balises <configuration> </ configuration>:

```
hduser@Hadoop:~$ vi /usr/local/hadoop/etc/hadoop/core-site.xml

<configuration>
 <property>
  <name>hadoop.tmp.dir</name>
  <value>/app/hadoop/tmp</value>
  <description>A base for other temporary directories.</description>
 </property>

 <property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:54310</value>
  <description>The name of the default file system.  A URI whose
  scheme and authority determine the FileSystem implementation.  The
  uri's scheme determines the config property (fs.SCHEME.impl) naming
  the FileSystem implementation class.  The uri's authority is used to
  determine the host, port, etc. for a filesystem.</description>
 </property>
</configuration>
```

4. /usr/local/hadoop/etc/hadoop/mapred-site.xml

Par défaut, le dossier /usr/local/hadoop/etc/hadoop/ contient
/usr/local/hadoop/etc/hadoop/mapred-site.xml.template
fichier qui doit être renommé et copié avec le nom mapred-site.xml:

```
hduser@Hadoop:~$ cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template
/usr/local/hadoop/etc/hadoop/mapred-site.xml
```

Le fichier /usr/local/hadoop/etc/hadoop/mapred-site.xml est utilisé pour spécifier la structure utilisée pour MapReduce.
Nous devons entrer le contenu suivant entre les balises <configuration> </ configuration>:

```
<configuration>
 <property>
  <name>mapred.job.tracker</name>
  <value>localhost:54311</value>
  <description>The host and port that the MapReduce job tracker runs
  at.  If "local", then jobs are run in-process as a single map
  and reduce task.
  </description>
 </property>
</configuration>
```

6. /usr/local/hadoop/etc/hadoop/hdfs-site.xml

Le fichier /usr/local/hadoop/etc/hadoop/hdfs-site.xml doit être configuré pour chaque hôte du cluster utilisé (ici on a qu'un).
Il spécifie les répertoires qui seront utilisés comme nom et code de données sur cet hôte.

Avant de modifier ce fichier, nous devons créer deux répertoires qui contiendront le namenode et le datanode pour cette installation Hadoop.

Cela peut être fait en utilisant les commandes suivantes:

```
hduser@Hadoop:~$ sudo mkdir -p /usr/local/hadoop_store/hdfs/namenode
hduser@Hadoop:~$ sudo mkdir -p /usr/local/hadoop_store/hdfs/datanode
hduser@Hadoop:~$ sudo chown -R hduser:hadoop /usr/local/hadoop_store
```

Ouvrez le fichier et entrez le contenu suivant entre la balise <configuration> </ configuration>:

```
hduser@Hadoop:~$ vi /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

```
<configuration>
 <property>
  <name>dfs.replication</name>
  <value>1</value>
  <description>Default block replication.
  The actual number of replications can be specified when the file is
created.
  The default is used if replication is not specified in create time.
  </description>
 </property>
 <property>
   <name>dfs.namenode.name.dir</name>
   <value>file:/usr/local/hadoop_store/hdfs/namenode</value>
 </property>
 <property>
   <name>dfs.datanode.data.dir</name>
   <value>file:/usr/local/hadoop_store/hdfs/datanode</value>
 </property>
</configuration>
```

- **Format the New Hadoop Filesystem**

Maintenant, le système de fichiers Hadoop doit être formaté pour que nous puissions commencer à l'utiliser. La commande format doit être émise avec une autorisation en écriture car elle crée un répertoire en cours. dans le dossier /usr/local/hadoop_store/hdfs/namenode:

```
hduser@Hadoop:~$ hadoop namenode -format
```

ATTENTION ! Notez que la commande hadoop namenode -format doit être exécutée une fois avant de commencer à utiliser Hadoop.
Si cette commande est exécutée à nouveau après l'utilisation de Hadoop, toutes les données du système de fichiers Hadoop seront détruites.

- **Starting Hadoop**

Il est maintenant temps de démarrer le cluster avec un seul noeud.
Nous pouvons utiliser start-all.sh ou (start-dfs.sh et start-yarn.sh)

```
userhadoop@Hadoop:~$ cd /usr/local/hadoop/sbin
userhadoop@Hadoop:~$ ls
userhadoop@Hadoop:~$ sudo su hduser
```

Démarrez les démons NameNode et DataNode:
hduser@Hadoop:/usr/local/hadoop/sbin$ start-dfs.sh

Démarrez le démon ResourceManager et le démon NodeManager:
hduser@Hadoop:/usr/local/hadoop/sbin$ start-yarn.sh

Nous pouvons vérifier si c'est vraiment opérationnel:
hduser@Hadoop:/usr/local/hadoop/sbin$ jps
14306 DataNode
14660 ResourceManager
14505 SecondaryNameNode
14205 NameNode
14765 NodeManager
15166 Jps

Pour stoper Hadooop

stop-dfs.sh

- **Hadoop Web Interfaces**

Relançons Hadoop et voyons son interface Web:
start-dfs.sh
start-yarn.sh

NameNode et DataNodes

Tapez http://localhost:50070/
dans le navigateur, nous verrons ensuite l'interface utilisateur Web du démon **NameNode**

SecondaryNameNode

Tapez
http://localhost:50090/status.jsp comme URL, nous obtenons SecondaryNameNode:

Le numéro de port par défaut pour accéder à toutes les applications du cluster est 8088. Utilisez l'URL suivante pour visiter Resource Manager:
http://localhost:8088/

Nous aurons peut-être besoin que les configurations suivantes soient correctement définies.

/usr/local/hadoop/etc/hadoop/yarn-site.xml:

```
<configuration>
   <property>
      <name>yarn.nodemanager.aux-services</name>
      <value>mapreduce_shuffle</value>
   </property>
</configuration>
```

/usr/local/hadoop/etc/hadoop/mapred-site.xml:

```
<configuration>
 <property>
  <name>mapred.job.tracker</name>
  <value>localhost:54311</value>
 </property>

 <property>
   <name>mapreduce.framework.name</name>
   <value>yarn</value>
 </property>
</configuration>
```