**INSTALLATION DE APACHE HIVE**

sudo -i

#verifier jdk version

java -version

**#installation de hadoop :**

mkdir -p /opt/

tar xvzf /tmp/hadoop-2.10.1.tar.gz -C /opt

**#Définir les variables d’environnement JAVA\_HOME and HADOOP\_HOME**

**# Ajouter java.sh dans profile.d**

echo 'export JAVA\_HOME=/usr/lib/jvm/java-1.7.0\_80'> /etc/profile.d/java.sh

**# Ajouter hadoop.sh dans profile.d**

echo 'export HADOOP\_HOME=/opt/hadoop-2.10.1;export PATH=$HADOOP\_HOME/bin:$PATH' >/etc/profile.d/hadoop.sh

**# pour valider que JAVA\_HOME and HADOOP\_HOME est deja define dans cette session**

source /etc/profile.d/java.sh

source /etc/profile.d/hadoop.sh

**#pour checker la source**

env | grep HADOOP

env | grep JAVA

**#On a deja copier les 2 dossiers files et scripts dans /home/centos/ comme indique dans le video**

cd /home/centos/scripts/

sh hadoop-setup-script.sh

**#Format HDFS**

su - hdfs

cd /opt/hadoop-2.10.1/bin

./hdfs namenode -format

**#SI c’est bon tu vas voir à la fin de la longues listes des messages :**

**#INFO common.Storage: Storage directory /var/data/hadoop/hdfs/nn has been successfully #formatted.**

**# Avec le user hdfs**

cd /opt/hadoop-2.10.1/sbin

./hadoop-daemon.sh start namenode

./hadoop-daemon.sh start secondarynamenode

./hadoop-daemon.sh start datanode

jps

**#Affichage**

#$ jps

#15140 SecondaryNameNode

#15015 NameNode

#15335 Jps

#15214 DataNode

**# Creation /mr-history pour le job history server directory dans hdfs**

**# Aussi un bon test pour verifier que HDFS up**

hdfs dfs -mkdir -p /mr-history/tmp

hdfs dfs -mkdir -p /mr-history/done

hdfs dfs -chown -R yarn:hadoop /mr-history

**# Avec le user yarn:**

su - yarn

cd /opt/hadoop-2.10.1/sbin

./yarn-daemon.sh start resourcemanager

./yarn-daemon.sh start nodemanager

./mr-jobhistory-daemon.sh start historyserver

jps

**#Affichage**

#$ jps

#23090 Jps

#20546 JobHistoryServer

#19987 ResourceManager

#20080 NodeManager

# Avec le user hdfs

Su – hdfs

# Exécuter un exemple MapReduce

export YARN\_EXAMPLES=/opt/hadoop-2.10.1/share/hadoop/mapreduce/

yarn jar $YARN\_EXAMPLES/hadoop-mapreduce-examples-2.10.1.jar pi 8 100000

**#si le test est bon hadoop est installe correctement**

**#Installation Apache Hive**

**#As root**

wget -P /tmp https://archive.apache.org/dist/hive/hive-2.3.2/apache-hive-2.3.2-bin.tar.gz

tar xvzf /tmp/apache-hive-2.3.2-bin.tar.gz -C /opt

echo 'export PATH=$PATH:/opt/apache-hive-2.3.2-bin/bin; export HIVE\_HOME=/opt/apache-hive-2.3.2-bin' >/etc/profile.d/hive.sh

**#Ajouter les repertoires necessaires**

su - hdfs -c "hdfs dfs -mkdir -p /user/hive/warehouse"

su - hdfs -c "hdfs dfs -chmod g+w /user/hive/warehouse"

**# Copier le fichier hive-site.xml**

cp files/hive-site.xml /opt/apache-hive-2.3.2-bin/conf

**# Suppimer extra log4j-slf4j (included in Hadoop install)**

mv /opt/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2.jar /opt/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2.jar.extra

**#As root**

useradd -g hadoop hive

chown -R hive:hadoop /opt/apache-hive-2.3.2-bin

**#Install Apache Derby**

wget -P /tmp https://archive.apache.org/dist/db/derby/db-derby-10.12.1.1/db-derby-10.12.1.1-bin.tar.gz

tar xvzf /tmp/db-derby-10.12.1.1-bin.tar.gz -C /opt

**#Setting environment**

echo 'export DERBY\_HOME=/opt/db-derby-10.12.1.1-bin; export PATH=$DERBY\_HOME/bin:$PATH; export DERBY\_OPTS="-Dderby.system.home=$DERBY\_HOME/data"'>/etc/profile.d/derby.sh

source /etc/profile.d/derby.sh

source /etc/profile.d/hive.sh

cp $DERBY\_HOME/lib/derbyclient.jar $HIVE\_HOME/lib

cp $DERBY\_HOME/lib/derbytools.jar $HIVE\_HOME/lib

**#Start derby**

cd /home/centos/

nohup startNetworkServer -h 0.0.0.0 &

**#Change derby to hive user**

chown -R hive:hadoop /opt/db-derby-10.12.1.1-bin

**# Config Hive schema**

schematool -initSchema -dbType derby

**#Start/Test Hive**

**#Valider d’abord que les services hadoop sont tous en marche**

su – hdfs

$ hive

hive> show tables;

OK

Time taken: 2.867 seconds

hive> quit;