Práctica Vagrant-Ansible

El objetivo de esta práctica es aprovisionar un box de Vagrant, usando Ansible, con el software Cloudera Distribution Hadoop, DH5.

Para comprobar el correcto funcionamiento de *Hadoop* en la máquina virtual creada con *Vagrant* y aprovisionada con *Ansible* se ha seguido el apartado *"Starting Hadoop and Verifying it is Working Properly"* del manual explicado por *Cloudera*, que podemos encontrar en el siguiente enlace:

https://docs.cloudera.com/documentation/cdh/5-1-x/CDH5-Quick-Start/CDH5-Quick-Start.html

Los pasos que se han seguido son los siguientes:

- Creación de un directorio llamado /tmp y configuración de sus permisos:

  $ sudo -u hdfs hadoop fs -mkdir -p /tmp
  $ sudo -u hdfs hadoop fs -chmod -R 1777 /tmp

- Creación de los directorios de sistema de MapReduce:

  $ sudo -u hdfs hadoop fs -mkdir -p /var/lib/hadoop-hdfs/cache/mapred/mapred/staging
  $ sudo -u hdfs hadoop fs -chmod 1777 /var/lib/hadoop-hdfs/cache/mapred/mapred/staging
  $ sudo -u hdfs hadoop fs -chown -R mapred /var/lib/hadoop-hdfs/cache/mapred

- A continuación, se ha verificado la estructura del archivo HDFS:

  $ sudo -u hdfs hadoop fs -ls -R /

```
vagrant@precise64:~$ sudo -u hdfs hadoop fs -ls -R /
drwxrwxrwt   - hdfs supergroup          0 2020-03-26 16:44 /prueba
-rw-r--r--   1 vagrant supergroup    2198903 2020-03-26 16:44 /prueba/donquijote.txt
drwxrwxrwt   - hdfs    supergroup          0 2020-03-26 16:39 /tmp
drwxr-xr-x   - hdfs    supergroup          0 2020-03-26 18:03 /var
drwxr-xr-x   - hdfs    supergroup          0 2020-03-26 18:03 /var/lib
drwxr-xr-x   - hdfs    supergroup          0 2020-03-26 18:03 /var/lib/hadoop-hdfs
drwxr-xr-x   - hdfs    supergroup          0 2020-03-26 18:03 /var/lib/hadoop-hdfs/cache
drwxr-xr-x   - mapred  supergroup          0 2020-03-26 18:03 /var/lib/hadoop-hdfs/cache/mapred
drwxr-xr-x   - mapred  supergroup          0 2020-03-26 18:03 /var/lib/hadoop-hdfs/cache/mapred/mapred
drwxrwxrwt   - mapred  supergroup          0 2020-03-26 18:03 /var/lib/hadoop-hdfs/cache/mapred/mapred/staging
```

- Se ha ejecutado MapReduce para comprobar que los pasos anteriores se han realizado correctamente y ver que la instalación funciona adecuadamente:

  $ for x in 'cd /etc/init.d; ls hadoop-0.20-mapreduce-*'; do sudo service $x start; done

```
vagrant@precise64:~$ for x in `cd /etc/init.d ; ls hadoop-0.20-mapreduce-*` ; do sudo service $x start ; done
/usr/lib/hadoop-0.20-mapreduce/hadoop-core-2.6.0-mr1-cdh5.15.1.jar  /usr/lib/hadoop-0.20-mapreduce/hadoop-core-mr1.jar
starting jobtracker, logging to /var/log/hadoop-0.20-mapreduce/hadoop-hadoop-jobtracker-precise64.out
/usr/lib/hadoop-0.20-mapreduce/hadoop-core-2.6.0-mr1-cdh5.15.1.jar  /usr/lib/hadoop-0.20-mapreduce/hadoop-core-mr1.jar
 * Started Hadoop jobtracker:
/usr/lib/hadoop-0.20-mapreduce/hadoop-core-2.6.0-mr1-cdh5.15.1.jar  /usr/lib/hadoop-0.20-mapreduce/hadoop-core-mr1.jar
tasktracker running as process 1340. Stop it first.
 * Started Hadoop tasktracker:
```

- Una vez ejecutado *MapReduce*, se ha realizado una aplicación de ejemplo para comprobar que funciona. Primero, se ha creado un directorio en *HDFS* para el usuario que ejecutará el trabajo (vagrant en nuestro caso):

  $ sudo -u hdfs hadoop fs -mkdir -p /user/vagrant
  $ sudo -u hdfs hadoop fs -chown vagrant /user/vagrant

- Dentro de *HDFS* se ha creado otro directorio llamado *input* y se han copiado algunos archivos XML

  $ hadoop fs -mkdir input
  $ hadoop fs -put /etc/hadoop/conf/*.xml input
  $ hadoop fs -ls input

```
vagrant@precise64:~$ hadoop fs -mkdir input
vagrant@precise64:~$ hadoop fs -put /etc/hadoop/conf/*.xml input
vagrant@precise64:~$ hadoop fs -ls input
Found 4 items
-rw-r--r--   1 vagrant supergroup       2133 2020-03-26 18:09 input/core-site.xml
-rw-r--r--   1 vagrant supergroup       3032 2020-03-26 18:09 input/fair-scheduler.xml
-rw-r--r--   1 vagrant supergroup       1875 2020-03-26 18:09 input/hdfs-site.xml
-rw-r--r--   1 vagrant supergroup        582 2020-03-26 18:09 input/mapred-site.xml
```

- A continuación, se ha ejecutado un trabajo de ejemplo de *Hadoop* para añadir con una expresión regular con los datos de entrada:

  $ /usr/bin/hadoop jar /usr/lib/hadoop-0.20-mapreduce/hadoop-examples.jar grep input output 'dfs[a-z.]+'

- Después de haber ejecutado el comando del paso anterior, podemos ver que se ha creado el directorio *output* que hemos especificado:

  $ hadoop fs -ls

```
vagrant@precise64:~$ hadoop fs -ls
Found 2 items
drwxr-xr-x   - vagrant supergroup          0 2020-03-26 18:09 input
drwxr-xr-x   - vagrant supergroup          0 2020-03-26 18:11 output
```

- Por último, podemos ver los ficheros que se generan como resultado en el directorio *output:*

  $ hadoop fs -ls output

```
vagrant@precise64:~$ hadoop fs -ls output
Found 3 items
-rw-r--r--   1 vagrant supergroup          0 2020-03-26 18:11 output/_SUCCESS
drwxr-xr-x   - vagrant supergroup          0 2020-03-26 18:10 output/_logs
-rw-r--r--   1 vagrant supergroup        150 2020-03-26 18:11 output/part-00000
```

- Ejecutando el siguiente comando veremos los resultados de los archivos generados:

  $ hadoop fs -cat output/part-00000 | head

```
vagrant@precise64:~$ hadoop fs -cat output/part-00000 | head
1       dfs.datanode.data.dir
1       dfs.namenode.checkpoint.dir
1       dfs.namenode.name.dir
1       dfs.replication
1       dfs.safemode.extension
1       dfs.safemode.min.datanodes
```

Como conclusión de los pasos explicados y realizados en este documento, se puede decir que *Hadoop*, además de la instalación de *MapReduce*, funcionan correctamente.