



Instalación Hadoop

Guía para Debian y derivados

Índice

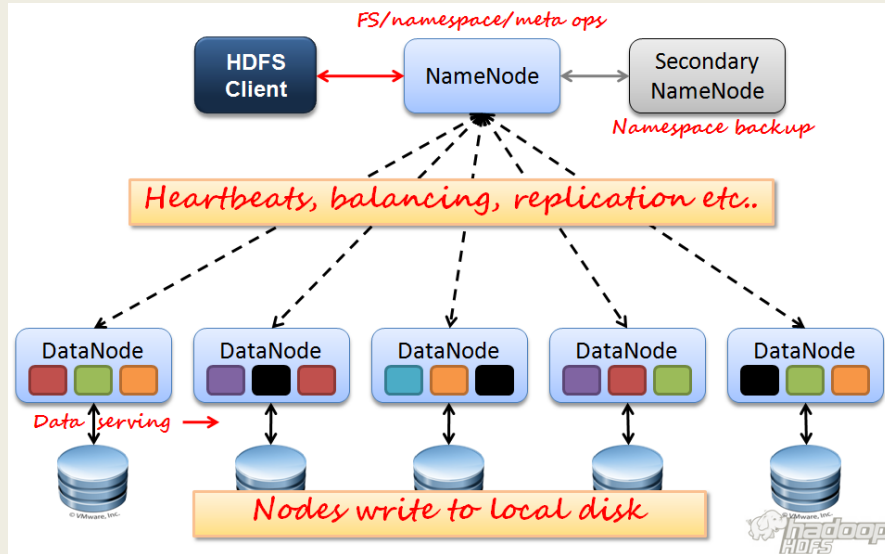
Instalación Hadoop

- [Hadoop Distributed File System](#)
 - a. [NameNode](#)
 - b. [DataNode](#)
- [Requisitos](#)
- [Diferentes modos de configuración](#)
- [Instalación Java](#)
- [Instalación de Hadoop](#)
- [Creación del usuario Hadoop](#)
- [Configuración de red y SSH](#)
- [Configuración HDFS](#)
 - a. [Archivo core-site.xml](#)
 - b. [Archivo hdfs-site.xml](#)
 - c. [Creación de carpetas HDFS](#)
 - d. [Archivo hadoop-env.sh](#)
 - e. [Archivo mapred-site.xml](#)
 - f. [Archivo yarn-site.xml](#)

Hadoop Distributed File System

HDFS es una implementación del GFS (Google File System). HDFS utiliza un bloque de archivos de 64mb o 128mb.

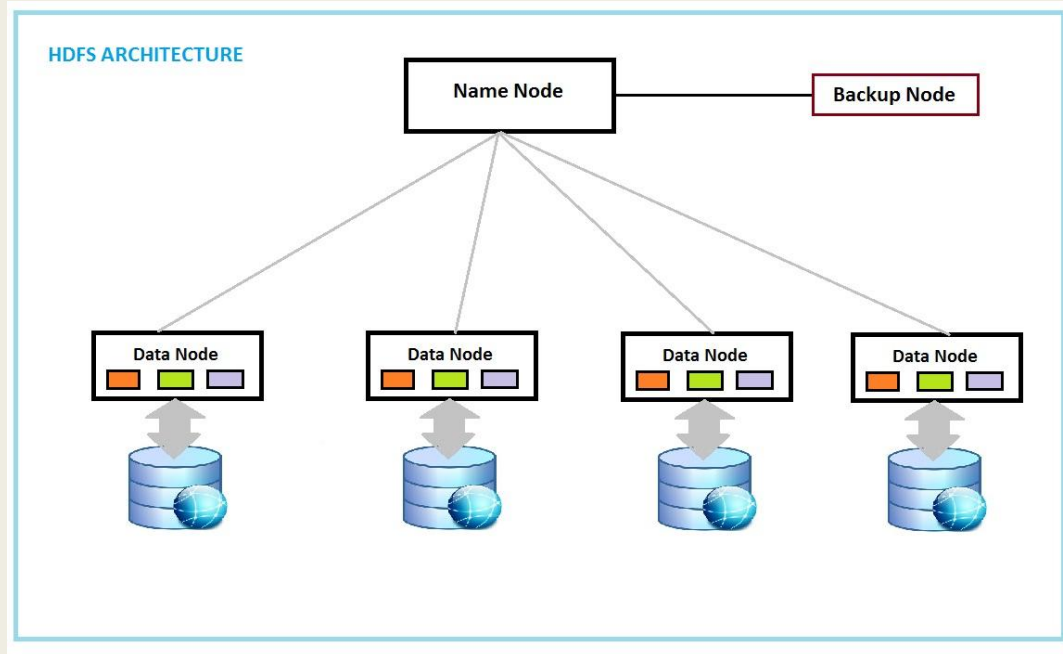
HDFS está diseñado a prueba de fallos, normalmente tiene por defecto un valor de replicación de 3.



DataNode

El NameNode es el nodo principal y los DataNode son los nodos esclavos.

Desde el NameNode se coordinan todos los trabajos a ejecutarse en el HDFS y en los DataNode se encargan de ejecutar las órdenes del NameNode e informar su estado al nodo principal.




NameNode

Realiza la gestión del cluster, los metadatos de los ficheros y de los directorios. Coordina el envío de los bloques de información a cada DataNode, verificando la disponibilidad de los diferentes nodos.

Los metadatos son almacenados en memoria RAM para que el acceso a estos sea mucho más rápido

NameNode Metadata



- Meta-data in Memory
 - » The entire metadata is in main memory
 - » No demand paging of FS meta-data
- Types of Metadata
 - » List of Files
 - » List of Blocks for each file
 - » List of DataNode for each block
 - » File attributes, e.g. access time, replication factor
- A Transaction Log
 - » Records file creations, file deletions. etc

NameNode
(Stores metadata only)

METADATA:
/user/doug/hinfo -> 1 3 5
/user/doug/pdetail -> 4 2

NameNode:
Keeps track of overall file directory structure and the placement of Data Block

Slide 20 Twitter @edurekaIN, Facebook /edurekaIN, use #askEdureka for Questions www.edureka.co/hadoop-admin

Requisitos

- Tener una instalación de alguna distribución de Linux o un Windows de 32 bits.
- Para el ejemplo se tiene un sistema operativo Xubuntu 14.04.

xubuntu 

Diferentes modos de configuración

Un único nodo en local (single node), utilizado para hacer pruebas de concepto corriendo Hadoop en una misma máquina

Un cluster pseudo-distribuido para simular un cluster de varios nodos pero corriendo en una misma máquina.

Montar un cluster entre distintas máquinas (multi node) totalmente distribuido que sería el modo que utilizamos para montar un sistema Big Data en producción.

- Para este tutorial se tendrá una configuración de cluster con único nodo.

Instalación de Hadoop

Verificar que Java esté instalado:

```
hadoop@hadoop-VirtualBox:~$ java -version
java version "1.8.0_51"
Java(TM) SE Runtime Environment (build 1.8.0_51-b16)
Java HotSpot(TM) 64-Bit Server VM (build 25.51-b03, mixed mode)
```

Descargar la última versión de Hadoop desde la página oficial:

<https://hadoop.apache.org/releases.html>

Ejecutar los siguientes comandos, según el nombre del archivo descargado.

```
sudo tar xzf hadoop-2.7.1.tar.gz
sudo mv hadoop-2.7.1 /usr/local/
sudo mv /usr/local/hadoop-2.7.1 /usr/local/hadoop
```


Creación del usuario Hadoop

El usuario Hadoop se utiliza para administrar los servicios del HDFS. Ejecutar el siguiente comando para crear el usuario y darle permisos administrativos:

```
useradd -d /home/hadoop -m hadoop  
passwd hadoop  
usermod -a -G sudo hadoop  
usermod -s /bin/bash hadoop
```

Luego se ingresa al sistema con este usuario:

```
su - hadoop
```

Creación del usuario Hadoop

Agregar al final del archivo `$HOME/.bashrc` del usuario donde se instala Hadoop las siguientes líneas:

```
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=${HADOOP_HOME}
export HADOOP_COMMON_HOME=${HADOOP_HOME}
export HADOOP_HDFS_HOME=${HADOOP_HOME}
export YARN_HOME=${HADOOP_HOME}
```

Estas líneas son las variables de entorno del HDFS, le especifican al sistema dónde encontrar los archivos que necesita para la ejecución y configuración del HDFS.

Configuración de red y SSH

Los nodos de Hadoop deberán poder conectarse entre sí mediante una conexión ssh (sin contraseña o con una misma contraseña para todos). Para esto, desde el usuario donde se está instalando hadoop se crea una clave pública ssh que se compartirá con los demás nodos.

```
sudo apt-get install ssh  
ssh-keygen -t rsa -f ~/.ssh/id_rsa  
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Configuración de red y SSH

Le permisos de lectura al archivo de autorización de conexión por ssh:

```
sudo chmod go-w $HOME $HOME/.ssh  
sudo chmod 600 $HOME/.ssh/authorized_keys  
sudo chown `whoami` $HOME/.ssh/authorized_keys
```

Para probar la configuración, se realiza una conexión por ssh con el localhost:

```
ssh localhost
```

Si todo sale bien, estaremos en una sesión por SSH sin haber ingresado una contraseña. Para salir de la sesión SSH ejecutar:

```
exit
```

Configuración de red y SSH

Ahora deshabilitamos ipv6 desde los archivos de configuración ya que Hadoop advierte en su documentación oficial que no tiene compatibilidad con el uso de ipv6. Para esto agregaremos lo siguiente al archivo `/etc/sysctl.conf`

```
net.ipv6.conf.all.disable_ipv6 = 1  
net.ipv6.conf.default.disable_ipv6 = 1  
net.ipv6.conf.lo.disable_ipv6 = 1
```

Configuración HDFS

(Hadoop Distributed File System)

Los archivos principales de configuración del HDFS se encuentran en el directorio `/usr/local/hadoop/etc/hadoop`. Allí se modificarán varios archivos.

Los archivos más relevantes en la configuración son:

- `core-site.xml`
- `hdfs-site.xml`
- `hadoop-env.sh`
- `mapred-site.xml`
- `yarn-site.xml`

Archivo core-site.xml

Primero editaremos el archivo core-site.xml con lo siguiente:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:8020</value>
    <description>Nombre del filesystem por defecto.</description>
  </property>
</configuration>
```

Archivo hdfs-site.xml

Luego en el archivo hdfs-site.xml especificaremos que se utilizará un factor de replicación igual a 1. Nos cercioramos de que quede escrito lo siguiente:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/home/hadoop/workspace/dfs/name</value>
    <description>Path del filesystem donde el namenode almacenará los metadatos.</description>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/home/hadoop/workspace/dfs/data</value>
    <description>Path del filesystem donde el datanode almacenará los bloques.</description>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
    <description>Factor de replicación. Lo ponemos a 1 porque sólo tenemos 1 máquina.</description>
  </property>
</configuration>
```


Creación de carpetas HDFS

Creamos los directorios `/home/hadoop/workspace/dfs/name` y `/home/hadoop/workspace/dfs/data`:

```
mkdir -p /home/hadoop/workspace/dfs/name
```

```
mkdir -p /home/hadoop/workspace/dfs/data
```

Archivo hadoop-env.sh

Cambiamos el valor de JAVA_HOME en el archivo hadoop-env.sh

```
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
```

Antes

```
GNU nano 2.2.6 File: hadoop-env.sh
# The only required environment variable is JAVA_HOME$
# optional. When running a distributed configuration$
# set JAVA_HOME in this file, so that it is correctly$
# remote nodes.

# The java implementation to use.
export JAVA_HOME=${JAVA_HOME}

# The jsvc implementation to use. Jsvc is required to$
# that bind to privileged ports to provide authentica$

^G Get He^O WriteO^R Read F^Y Prev P^K Cut Te^C Cur Po
^X Exit ^J Justif^W Where ^V Next P^U UnCut ^T To Spe
```

Después

```
GNU nano 2.2.6 File: hadoop-env.sh
# The only required environment variable is JAVA_HOME$
# optional. When running a distributed configuration$
# set JAVA HOME in this file, so that it is correctly$
# remote nodes.

# The java implementation to use.
export JAVA_HOME=/usr/lib/jvm/java-8-oracle

# The jsvc implementation to use. Jsvc is required to$
# that bind to privileged ports to provide authentica$

[ Wrote 98 lines ]
^G Get He^O WriteO^R Read F^Y Prev P^K Cut Te^C Cur Po
^X Exit ^J Justif^W Where ^V Next P^U UnCut ^T To Spe
```

Archivo mapred-site.xml

Hacemos una copia del archivo por defecto mapred-site.xml.template hacia mapred-site.xml

```
cp mapred-site.xml.template mapred-site.xml
```

Creamos los directorios /home/hadoop/workspace/mapred/system y /home/hadoop/workspace/mapred/local

```
mkdir -p /home/hadoop/workspace/mapred/system
```

```
mkdir -p /home/hadoop/workspace/mapred/local
```

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

Archivo yarn-site.xml

Entre las etiquetas `<configuration></configuration>` dentro del archivo `yarn-site.xml` se agrega lo siguiente:

```
<property>  
  <name>yarn.nodemanager.aux-services</name>  
  <value>mapreduce_shuffle</value>  
</property>  
<property>  
  <name>yarn.nodemanager.aux-services.mapreduce_shuffle.class</name>  
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>  
</property>
```

Formatear el NameNode

Formatea el sistema de ficheros HDFS.

```
hdfs namenode -format
```

Nota:

'hadoop namenode -format' es un comando obsoleto



Comandos importantes

Iniciar los servicios de hadoop

Se ejecuta los siguientes comandos uno después del otro.

```
start-dfs.sh
```

```
start-yarn.sh
```

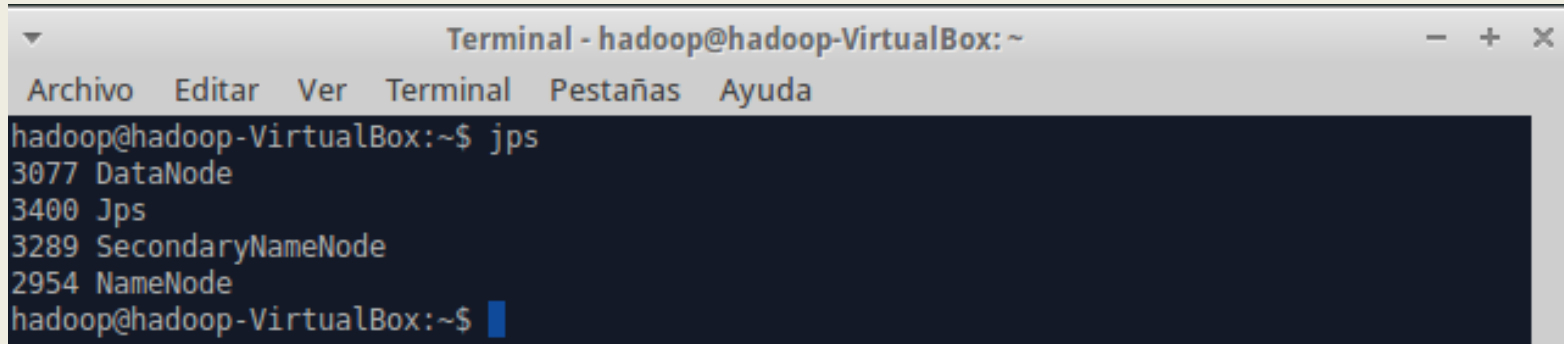
Nota:

'start-all.sh' es un comando obsoleto.

jps

Verificar todos los procesos Java que estén en ejecución:

jps

A terminal window titled "Terminal - hadoop@hadoop-VirtualBox: ~" with standard window controls. The menu bar includes "Archivo", "Editar", "Ver", "Terminal", "Pestañas", and "Ayuda". The terminal shows the command "jps" being executed, resulting in the following output:

```
hadoop@hadoop-VirtualBox:~$ jps
3077 DataNode
3400 Jps
3289 SecondaryNameNode
2954 NameNode
hadoop@hadoop-VirtualBox:~$
```


Crear un directorio en HDFS

Crear un directorio o carpeta dentro del sistema de archivos HDFS de manera análoga al comando `mkdir` de UNIX.

```
hadoop fs -mkdir carpeta
```

Ejemplo

Crear la carpeta `/user/example`:

```
hadoop fs -mkdir /user/example
```

Dar permisos a un archivo

Dar permisos a un archivo (O carpeta) de manera análoga al comando `chmod` de UNIX.

```
hadoop fs -chmod permisos archivo
```

Donde *archivo* será la ruta al archivo que se le aplicará el cambio.

Ejemplo

Dar permisos de lectura y escritura en `/user/example`.

```
hadoop fs -chmod +rw /user/example
```

Listar archivos de un directorio

Listar los archivos y subdirectorios de un directorio de forma análoga al comando ls de UNIX.

```
hadoop fs -ls archivo
```

Donde archivo es la ruta al directorio que se desea listar.

Ejemplo

```
hadoop fs -ls /user/example
```

Copiar archivos en el HDFS

Para enviar un archivo local hacia el sistema de archivos HDFS se utiliza el siguiente comando.

```
hadoop fs -copyFromLocal archOrigen archDestino
```

Donde “archOrigen” es la ruta del archivo que se desea subir y “archDestino” es la ruta donde se desea almacenar el archivo en el HDFS.

Ejemplo

```
hadoop fs -copyFromLocal archivo.txt /user/example
```


Ingresar al administrador web

Si todo lo anterior está correctamente instalado y configurado (Y se han iniciado todos los servicios de Hadoop), se puede ingresar al administrador web de Hadoop desde la dirección <http://localhost:8088/>

All Applications - Mozilla Firefox

All Applications x +

localhost:8088/cluster Search



Cluster

- About
- Nodes
- Node Labels
- Applications
 - NEW
 - NEW_SAVING
 - SUBMITTED
 - ACCEPTED
 - RUNNING
 - FINISHED
 - FAILED
 - KILLED
- Scheduler

Cluster Metrics

| Apps Submitted | Apps Pending | Apps Running | Apps Completed | Containers Running |
|----------------|--------------|--------------|----------------|--------------------|
| 0 | 0 | 0 | 0 | 0 |

Scheduler Metrics

| Scheduler Type | Scheduling |
|--------------------|------------|
| Capacity Scheduler | [MEMORY] |

Show 20 entries

| ID | User | Name | Application Type |
|----|------|------|------------------|
|----|------|------|------------------|

HDFS Web

También se puede acceder a un administrador web del sistema de archivos HDFS a través del url <http://localhost:50070>

Overview 'localhost:9000' (active)

| | |
|-----------------------|---|
| Started: | Wed Aug 26 11:33:07 COT 2015 |
| Version: | 2.6.0, re3496499ecb8d220fba99dc5ed4c99c8f9e33bb1 |
| Compiled: | 2014-11-13T21:10Z by jenkins from (detached from e349649) |
| Cluster ID: | CID-c7b645fb-3638-4242-a97e-f3aa27cdd179 |
| Block Pool ID: | BP-185426378-127.0.0.1-1434645368452 |

Summary

Security is off.

Safemode is off.

8 files and directories, 5 blocks = 13 total filesystem object(s).

Heap Memory used 61.73 MB of 177.5 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 38.27 MB of 40.13 MB Committed Non Heap Memory. Max Non Heap Memory is -1 B.

| | |
|-----------------------------|-----------|
| Configured Capacity: | 85.92 GB |
| DFS Used: | 378.61 MB |
| Non DFS Used: | 31.2 GB |
| DFS Remaining: | 54.35 GB |

Referencias

- <http://www.adictosaltrabajo.com/tutoriales/hadoop-first-steps/>
- <http://tecadmin.net/install-oracle-java-8-jdk-8-ubuntu-via-ppa/>
- <http://www.webupd8.org/2012/09/install-oracle-java-8-in-ubuntu-via-ppa.html>
- <https://hadoop.apache.org/releases.html>
- https://hadoop.apache.org/docs/r1.0.4/file_system_shell.html
- https://hadoop.apache.org/docs/r1.2.1/single_node_setup.html
- <https://www.youtube.com/watch?v=CobVqNMiqww>
- <http://www.apache.org/dyn/closer.cgi/spark/spark-1.4.1/spark-1.4.1-bin-hadoop2.6.tgz>