Prácticas BigData

1. Instalación de Hadoop. Modo StandAlone

- Arrancamos la máquina virtual facilitada por el profesor

1.1. Crear usuario hadoop

- Si no usamos la máquina del curso debemos crear un usuario para hadoop.
- Accedemos como ROOT al sistema.
- Ejecutamos el siguiente comando para crear el usuario

```
useradd hadoop
```

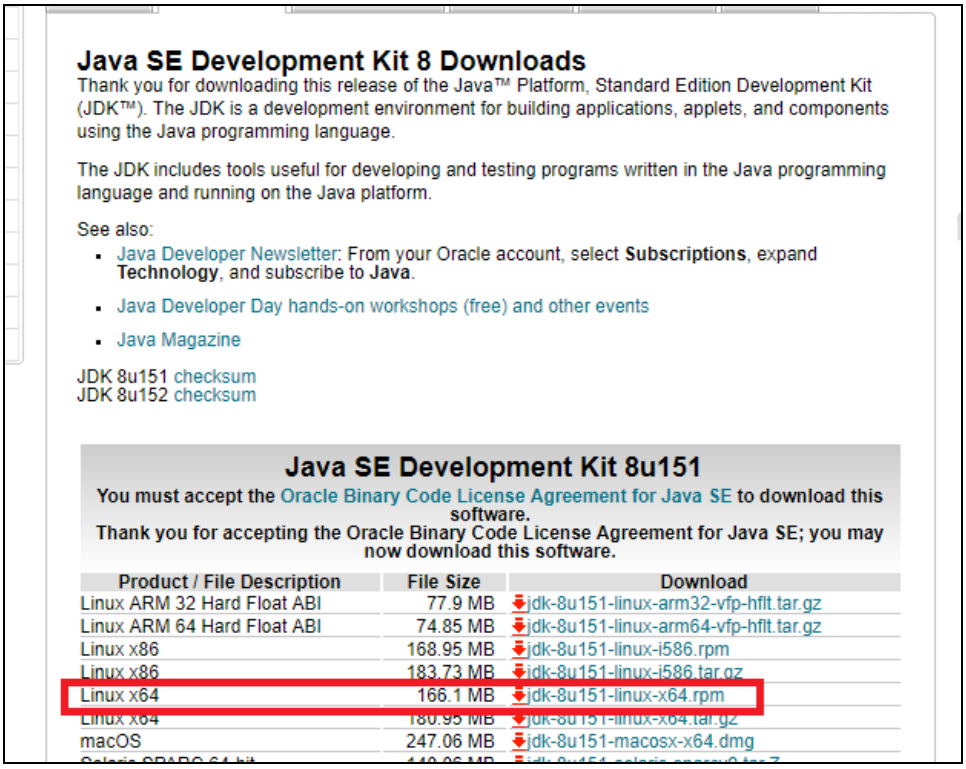
- Le ponemos contraseña

```
passwd hadoop
```

- Debe haber creado un directorio denominado /home/hadoop

1.2. Instalar JDK de Oracle

- Seguimos como ROOT
- Descargamos el RPM de Java de la página de Oracle. Siempre es más fácil de instalar.
- NOTA: la versión puede no cuadrar con la existente en el momento de hacer este manual.



Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u151 checksum
JDK 8u152 checksum

Java SE Development Kit 8u151

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.9 MB	jdk-8u151-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.85 MB	jdk-8u151-linux-arm64-vfp-hflt.tar.gz
Linux x86	168.95 MB	jdk-8u151-linux-i586.rpm
Linux x86	183.73 MB	jdk-8u151-linux-i586.tar.gz
Linux x64	166.1 MB	jdk-8u151-linux-x64.rpm
Linux x64	180.95 MB	jdk-8u151-linux-x64.tar.gz
macOS	247.06 MB	jdk-8u151-macosx-x64.dmg

- Instalamos JDK mediante RPM o cualquier otro de los mecanismos del Sistema operativo con el que estemos trabajando. El que usamos durante el curso es un CENTOS.

```
rpm –ivh jdkXXXXXX.rpm
```

- Debemos asegurarnos de que usa el JDK que hemos descargado para que no tengamos problemas.
- Si disponemos de distintas versiones podemos utilizar el siguiente comando. Debemos seleccionar la versión que hemos descargado. Seguramente existen otras que vienen con el propio CENTOS.

```
alternatives --config java

Hay 3 programas que proporcionan 'java'.

  Selección    Comando
-----------------------------------------------
*+ 1           /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.75-2.5.4.2.el7_0.x86_64/jre/bin/java
   2           /usr/java/jdk1.8.0_45/jre/bin/java
   3           /usr/java/jdk1.8.0_45/bin/java
```

```
Presione Intro para mantener la selección actual[+], o escriba el número de la selección:  3
```

- Comprobamos que podemos acceder a JAVA y a sus comandos

```
javac -version
javac 1.8.0_45

java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)
```

1.3. Configurar las variables de entorno para JAVA

- Accedemos como usuario HADOOP
- Para que no haya problemas durante el curso debemos configurar el entorno.
- Configuramos las variables de entorno dentro del usuario que estemos utilizando.
- Debemos poner esas variables en algún fichero que las cargue cuando el usuario "hadoop" acceda al sistema.
- En este caso vamos a usar el fichero **".bashrc"** (Con punto al principio. Recordemos que los ficheros que empiezan con punto en Linux son ocultos y no se ven con un simple "ls". Hay que hacer un "ls -a")
- Se encuentra en el directorio del usuario **"/home/hadoop"**, pero podemos usar cualquier otro fichero que permita cargar las variables al inicio.
- Incorporamos el acceso a JAVA. (En realidad, ponerlo en el PATH seguramente no hace falta porque lo ha hecho el instalador, pero siempre es mejor indicarlo)

```
export JAVA_HOME=/usr/java/jdkXXXXX
export PATH=$PATH:$JAVA_HOME/bin
```

1.4. Descargar e instalar hadoop

- Accedemos como usuario ROOT
- Con el Firefox, vamos a la página de Hadoop y descargamos el software. En el momento de hacer esta documentación la última versión estable es la 2.9
- **NOTA IMPORTANTE: el 15 de Diciembre se ha liberado la versión 3, pero este curso está basado en la 2. La instalación y otros componentes cambian. Al menos durante un tiempo y hasta que se estabilice es preferible seguir usando la 2. Estamos trabajando para añadir al curso los cambios de la 3 y hacer un anexo con los mismos.**
- Lo debe haber dejado en /root/Descargas (o Downloads si tenéis el entorno en inglés).
- Vamos a hacer la instalación en el directorio /opt
- Copiamos el software de hadoop a /opt.

```
cp hadoop-XXXX.tar /opt
```

- Accedemos a /opt

```
cd /opt
```

- Desempaquetamos el software

```
tar xvf hadoopXXX-bin.tar
```

- Esto debe haber creado un directorio denominado hadoop-XXXXXX.
- Para hacer más sencillo el trabajo lo cambiamos de nombre a “hadoop”

```
mv    hadoop-XXXX      hadoop
```

- Comprobamos si existen los ficheros desempaquetados en el directorio

```
ls -l /opt/hadoop
total 341504
drwxr-xr-x.  2 hadoop hadoop       194 nov 14 00:28 bin
drwxr-xr-x.  3 hadoop hadoop        20 nov 14 00:28 etc
drwxr-xr-x.  2 hadoop hadoop       106 nov 14 00:28 include
drwxr-xr-x.  3 hadoop hadoop        20 nov 14 00:28 lib
drwxr-xr-x.  2 hadoop hadoop       239 nov 14 00:28 libexec
-rw-r--r--.  1 hadoop hadoop    106210 nov 14 00:28 LICENSE.txt
drwxrwxr-x.  2 hadoop hadoop      4096 ene  4 18:28 logs
-rw-r--r--.  1 hadoop hadoop     15915 nov 14 00:28 NOTICE.txt
-rw-r--r--.  1 hadoop hadoop      1366 nov 14 00:28 README.txt
drwxr-xr-x.  3 hadoop hadoop      4096 dic 27 17:03 sbin
drwxr-xr-x.  4 hadoop hadoop        31 nov 14 00:28 share
```

- Cambiamos los permisos para que pertenezcan al usuario “hadoop”, que es con el que vamos a trabajar.

```
cd /opt
chown -R   hadoop:hadoop    hadoop
```

1.5. Configurar las variables de HADOOP y comprobar que todo funciona

- Salimos como usuario “root” y accedemos como usuario HADOOP
- Configuramos en el fichero “/home/hadoop/.bashrc” para las variables de acceso a Hadoop. Incluimos las siguientes

```
export HADOOP_HOME=/opt/hadoop
export PATH=$PATH:/$HADOOP_HOME/bin
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
```

- Para probarlo, salimos de la sesión y volvemos a entrar, o bien ejecutamos el siguiente comando (debemos dejar un espacio en blanco entre los dos puntos.

```
. ./.bashrc
```

- Ejecutar “hadoop –h” para comprobar si accedemos correctamente

```
hadoop -h
Usage: hadoop [--config confdir] COMMAND
      where COMMAND is one of:
  fs                   run a generic filesystem user client
  version              print the version
  jar <jar>            run a jar file
  checknative [-a|-h]  check native hadoop and compression libraries availability
  distcp <srcurl> <desturl> copy file or directories recursively
  archive -archiveName NAME -p <parent path> <src>* <dest> create a hadoop archive
  classpath            prints the class path needed to get the
  credential           interact with credential providers
                     Hadoop jar and the required libraries
  daemonlog            get/set the log level for each daemon
  trace                view and modify Hadoop tracing settings
 or
  CLASSNAME            run the class named CLASSNAME


Most commands print help when invoked w/o parameters.
```

- Comprobamos la versión utilizada (seguramente no será ya igual a la vuestra, en este caso usamos la última de la 2, que es la 2.9)

```
hadoop version
Hadoop 2.9.0
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r 756ebc8394e473ac25feac05fa493f6d612e6c50
Compiled by arsuresh on 2017-11-13T23:15Z
Compiled with protoc 2.5.0
From source with checksum 0a76a9a32a5257331741f8d5932f183
```

```
This command was run using /opt/hadoop/share/hadoop/common/hadoop-common-2.9.0.jar
```

- Vamos a realizar un pequeño ejemplo en modo standalone. Esto nos permite comprobar que todo funciona correctamente.
- **NOTA IMPORTANTE: recordad que tenéis que cambiar la versión en los ficheros. En este ejemplo estamos usando la 2.9. Debéis adaptarlo a lo que tengáis vosotros.**
- Nos situamos en /opt/hadoop

```
cd /opt/hadoop
```

- Creamos un directorio en /tmp

```
mkdir /tmp/input
```

- Copiamos todos los ficheros XML que hay en el diretorio /opt/hadoop/etc/hadoop

```
cp    etc/hadoop/*.xml    /tmp/input/
```

- Ejecutamos el siguiente comando que busca todos los ficheros de /tmp/input que tengan el texto "dfs" y luego tenga un carácter de la "a" a la "z" y deja el resultadoen el directorio /tmp/output". Funciona de forma parecida al grep de linux
- **NOTA**: en siguientes capítulos veremos con más detalle el comando hadoop. Por ahora solo es necesario saber que lanza un proceso de tipo MapReduce de Hadoop

```
hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.jar grep /tmp/input /tmp/output 'dfs[a-z.]+'
```

- Vemos lo que ha dejado en /tmp/output

```
ls -l /tmp/output/
total 4
-rw-r--r--. 1 hadoop hadoop 404 ene  6 15:00 part-r-00000
-rw-r--r--. 1 hadoop hadoop   0 ene  6 15:00 _SUCCESS
```

- Este programa genera un fichero denominado "part-r-0000" con el resultado del comando.
- Debe contener algo parecido a lo siguiente

```
cat /tmp/output/part-r-00000
2       dfs.namenode.http
2       dfs.namenode.rpc
1       dfsadmin
1       dfs.server.namenode.ha.
1       dfs.replication
1       dfs.permissions
1       dfs.nameservices
1       dfs.namenode.shared.edits.dir
```

1	dfs.namenode.name.dir
1	dfs.namenode.checkpoint.dir
1	dfs.journalnode.edits.dir
1	dfs.ha.namenodes.ha
1	dfs.ha.fencing.ssh.private
1	dfs.ha.fencing.methods
1	dfs.ha.automatic
1	dfs.datanode.data.dir
1	dfs.client.failover.proxy.provider.ha