

Introducción

Apache Hadoop es un marco para procesar y almacenar grandes cantidades de datos. Está diseñado para ser implementado en una red de cientos o incluso miles de servidores dedicados. Todas estas máquinas trabajan juntas para hacer frente al enorme volumen y variedad de conjuntos de datos entrantes.

La implementación de servicios de Hadoop en un solo nodo es una excelente manera de familiarizarse con los comandos y conceptos básicos.

Esta guía fácil de seguir explica cómo instalar Hadoop en Ubuntu y configurar una implementación de un solo nodo.

Prerrequisitos

- Acceso a una ventana de terminal.
- Privilegios de sudo o **root** en máquinas locales/remotas.
- Ubuntu versión **24.04** instalada.

Instalar JDK en Ubuntu

El marco de Hadoop está escrito en **Java** y sus servicios requieren un entorno de ejecución de Java (JRE) y un kit de desarrollo de Java (JDK) compatibles. Utilice el siguiente comando para actualizar el sistema antes de iniciar una nueva instalación:

```
sudo apt update
```

Por el momento, **Apache Hadoop 3.x es totalmente compatible con Java 8 y 11.**

Escribe uno de los siguientes comandos en su terminal para instalar OpenJDK dependiendo de la versión elegida :

```
sudo apt install openjdk-8-jdk -y  
ó  
sudo apt install openjdk-11-jdk -y
```

Nota: La versión de OpenJDK u Oracle Java puede afectar a la forma en que interactúan los elementos de un ecosistema de Hadoop.

Una vez completado el proceso de instalación, **verifica la versión actual de Java:**

```
java -version; javac -version
```

```
pnep@phoenixnap:~$ java -version; javac -version
openjdk version "1.8.0_422"
OpenJDK Runtime Environment (build 1.8.0_422-8u422-b05-1~24.04-b05)
OpenJDK 64-Bit Server VM (build 25.422-b05, mixed mode)
javac 1.8.0_422
```

El resultado le informa de qué versión de Java está en uso.

Configurar el usuario de Hadoop y configurar SSH

Es recomendable crear un usuario no root, específicamente para el entorno Hadoop. Un usuario distinto mejora la seguridad y le ayuda a administrar el clúster de forma más eficiente. Para garantizar el buen funcionamiento de los servicios de Hadoop, el usuario debe tener la capacidad de establecer una conexión SSH sin contraseña con el host local.

Instalar OpenSSH en Ubuntu

Instala el servidor y el cliente OpenSSH mediante el siguiente comando:

```
sudo apt install openssh-server openssh-client -y
```

Si ya está instalado, el resultado confirma que ya está instalada la última versión.

```
pnep@phoenixnap:~$ sudo apt install openssh-server openssh-client -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:9.6p1-3ubuntu13.5).
openssh-client is already the newest version (1:9.6p1-3ubuntu13.5).
openssh-client set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 166 not upgraded.
```

Crear usuario de Hadoop

Utilice el comando adduser para crear un nuevo usuario de Hadoop:

```
sudo adduser hdoop
```

El nombre de usuario, en este ejemplo, es **hdoop**. Puedes usar cualquier otro nombre de usuario y contraseña.

Cambia a este usuario creado e introduce la contraseña correspondiente:

```
su - hdoop
```

El usuario ahora debe poder conectarse mediante SSH al host local sin que se le solicite una contraseña.

Habilitar SSH sin contraseña para el usuario de Hadoop

Genera un par de claves SSH y defina la ubicación en la que se almacenará:

```
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
```

El sistema procede a generar y guardar el par de claves SSH.

Utiliza el comando cat para almacenar la clave pública tal como **authorized_keys** en el directorio ssh:

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Establezca los permisos de archivo para su usuario con el comando `chmod`:

```
chmod 0600 ~/.ssh/authorized_keys
```

```
hdoop@phoenixnap:~$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
Created directory '/home/hdoop/.ssh'.
Your identification has been saved in /home/hdoop/.ssh/id_rsa
Your public key has been saved in /home/hdoop/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:DFtcZg3wmo56IKQKdGTWSG8/+YePol1UvGWpVPpoy34 hdoop@phoenixnap
The key's randomart image is:
+---[RSA 3072]---+
| ..O ..=O . |
| =.. . =. + . |
| + O. O . = + |
| .... . =.oo B |
| .O. . +S. = . |
| O . . O+ + . |
| O . . . + + |
| . .O.. = E |
| .O.... O. |
+---[SHA256]-----+
hdoop@phoenixnap:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
hdoop@phoenixnap:~$ chmod 0600 ~/.ssh/authorized_keys
```

El nuevo usuario ahora puede conectar con SSH sin ingresar una contraseña cada vez. Verifica que todo esté configurado correctamente mediante el uso del **usuario hadoop** para SSH a localhost:

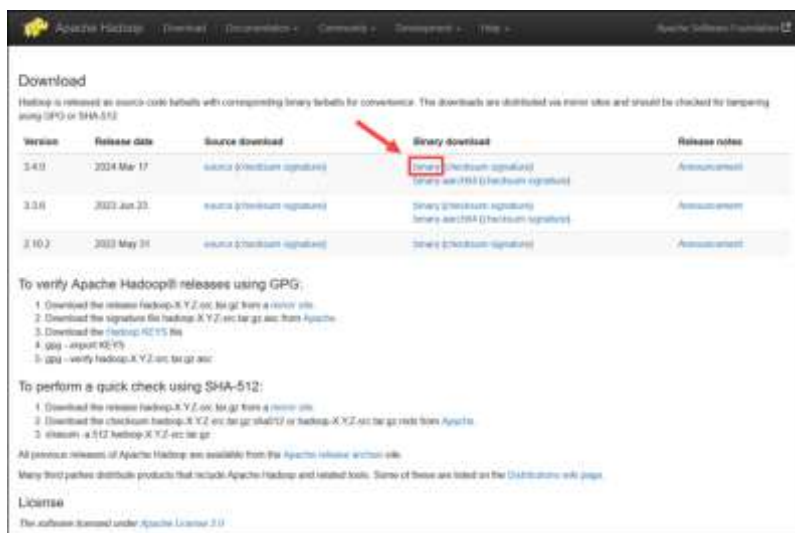
```
ssh localhost
```

Después de un mensaje inicial, el usuario de Hadoop puede establecer sin problemas una conexión SSH con el host local.

Descargar e instalar Hadoop en Ubuntu

Después de configurar el usuario de Hadoop, está listo para instalar Hadoop en su sistema. Siga los pasos a continuación:

1. Visite la [página oficial del proyecto Apache Hadoop](#) y seleccione la versión de Hadoop que desea implementar.



Los pasos descritos en este tutorial utilizan la [descarga binaria](#) para la **versión 3.4.0 de Hadoop**. Seleccione su opción preferida y se le presentará un enlace espejo para descargar el **paquete tar** de Hadoop.



2. Utiliza el enlace de espejo proporcionado y descargue el paquete de Hadoop mediante el comando wget:

```
wget https://dldcn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz
```



```
hadoop@phoenixnap:~$ wget https://dldcn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz
--2024-09-09 11:53:23-- https://dldcn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz
Resolving dldcn.apache.org (dldcn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dldcn.apache.org (dldcn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 965537117 (921M) [application/x-gzip]
Saving to: 'hadoop-3.4.0.tar.gz'

hadoop-3.4.0.tar.gz      100%[=====]
=====>] 920.81M  3.29MB/s   in 4m 31s

2024-09-09 11:57:55 (3.39 MB/s) - 'hadoop-3.4.0.tar.gz' saved [965537117/965537117]
```

3. Una vez completada la descarga, utiliza el comando tar para extraer el archivo.tar.gz e iniciar la instalación de Hadoop:

```
tar xzf hadoop-3.4.0.tar.gz
```

Los archivos binarios de Hadoop ahora se encuentran dentro del *directorio* `hadoop-3.4.0`. Se pueden mover a otro directorio, pero entonces más abajo, la variable `$HADOOP_HOME` contendría otro camino

```
# por ejemplo
sudo mv hadoop-3.4.0 /usr/local/hadoop
```

Implementación de Hadoop de un solo nodo (modo pseudodistribuido)

Hadoop sobresale cuando se implementa en un **modo totalmente distribuido** en un gran clúster de servidores en red. Sin embargo, si no está familiarizado con Hadoop y desea explorar comandos básicos o probar aplicaciones, puede configurar Hadoop en un solo nodo.

Esta configuración, también llamada **modo pseudodistribuido**, permite que cada demonio de Hadoop se ejecute como un único proceso Java. Configure un entorno de Hadoop editando un conjunto de archivos de configuración:

- `.bashrc`
- `hadoop-env.sh`
- `core-site.xml`

- *hdfs-site.xml*
- *mapred-site.xml*
- *yarn-site.xml*

Configurar variables de entorno de Hadoop (bashrc)

El archivo de configuración *.bashrc* es un script de shell que inicializa la configuración específica del usuario, como las variables de entorno, los alias y las funciones, cada vez que se inicia una nueva sesión de shell de Bash. Sigue los pasos que se indican a continuación para configurar las variables de entorno de Hadoop:

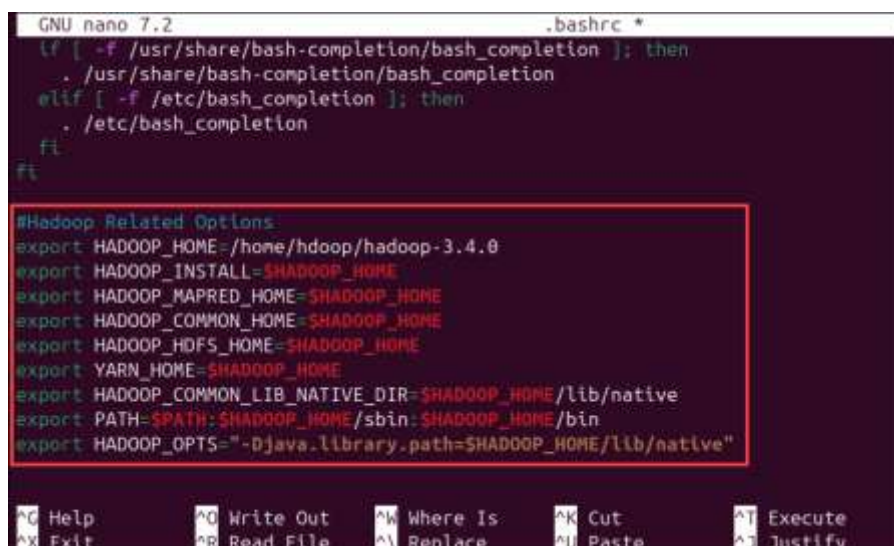
1. Edita el archivo de configuración del *shell .bashrc* usando un editor de texto de su elección (usaremos nano):

```
nano .bashrc
```

2. Define las variables de entorno de Hadoop agregando el siguiente contenido al final del archivo:

```
#Hadoop Related Options
export HADOOP_HOME=/home/hadoop/hadoop-3.4.0
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

3. Una vez que agregues las variables, guarda y sal del archivo *.bashrc*.



```
GNU nano 7.2 .bashrc *
if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
fi

#Hadoop Related Options
export HADOOP_HOME=/home/hadoop/hadoop-3.4.0
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

4. Ejecuta el siguiente comando para aplicar los cambios al entorno de ejecución actual:

```
source ~/.bashrc
```

Editar archivo `hadoop-env.sh`

El archivo `hadoop-env.sh` sirve como archivo maestro para configurar los ajustes de proyecto relacionados con YARN (encargado de administrar los recursos almacenados en HDFS), **HDFS** (responsable del mantenimiento de datos), **MapReduce** (responsable de ejecutar tareas) y Hadoop. Como se quiere configurar un **clúster de Hadoop de un solo nodo**, debes definir qué implementación de Java se utilizará. Pasos:

1. Utiliza la variable `$HADOOP_HOME` creada anteriormente para acceder al *archivo `hadoop-env.sh`*:

```
nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

2. Descomenta la variable `$JAVA_HOME` (es decir, elimina al comienzo el signo `#`) y agrega la ruta completa a la instalación de OpenJDK en su sistema. Dependiendo de la versión instalada agrega:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
ó
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
```

La ruta debe coincidir con la ubicación de la instalación de Java en su sistema.


```
GNU nano 7.2 /home/hdoop/hadoop-3.4.0/etc/hadoop/hadoop-env.sh *
# Technically, the only required environment variable is JAVA_HOME.
# All others are optional. However, the defaults are probably not
# preferred. Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d

# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

# The language environment in which Hadoop runs. Use the English
# environment to ensure that logs are printed as expected.
export LANG=en_US.UTF-8

# Location of Hadoop. By default, Hadoop will attempt to determine
# this location based upon its execution path.
# export HADOOP_HOME=

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

Si necesitas localizar la ruta de Java correcta, ejecuta el siguiente comando en la ventana de su terminal:

```
which javac
```

La salida resultante proporciona la ruta al directorio binario de Java.

```
hdoop@phoenixnap:~$ which javac
/usr/bin/javac
```

3. Utiliza la ruta proporcionada para encontrar el directorio OpenJDK con el siguiente comando:

```
readlink -f /usr/bin/javac
```

La sección de la ruta justo antes del directorio `/bin/javac` debe asignarse a la variable `$JAVA_HOME`.

```
hdoop@phoenixnap:~$ readlink -f /usr/bin/javac
/usr/lib/jvm/java-8-openjdk-amd64/bin/javac
```

Editar core-site.xml archivo

El archivo `core-site.xml` define las propiedades principales de HDFS y Hadoop. Para configurar Hadoop en un modo pseudodistribuido, debe especificar la **dirección URL** de su NameNode y el directorio temporal que Hadoop utiliza para el proceso de asignación y reducción.

Los pasos a continuación muestran cómo configurar el archivo.

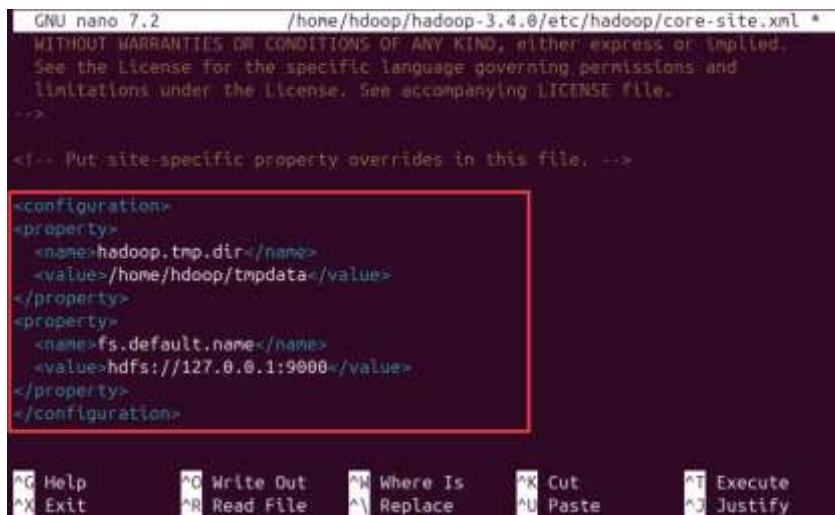
1. Abre el archivo *core-site.xml* en un editor de texto:

```
nano $HADOOP_HOME/etc/hadoop/core-site.xml
```

2. Agrega la siguiente configuración para anular los valores predeterminados del directorio temporal y agregue la dirección URL de HDFS para reemplazar la configuración predeterminada del sistema de archivos local:

```
<configuration>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/home/hadoop/tmpdata</value>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://127.0.0.1:9000</value>
</property>
</configuration>
```

En este ejemplo se utilizan valores específicos del sistema local. Debe utilizar valores que coincidan con los requisitos del sistema. Los datos deben ser coherentes durante todo el proceso de configuración.



```
GNU nano 7.2 /home/hadoop/hadoop-3.4.0/etc/hadoop/core-site.xml *
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the license for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/home/hadoop/tmpdata</value>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://127.0.0.1:9000</value>
</property>
</configuration>

^G Help      ^O Write Out  ^M Where Is    ^K Cut        ^T Execute
^X Exit      ^R Read File  ^I Replace    ^U Paste      ^J Justify
```

No olvides crear un directorio en la ubicación que especificó para sus datos temporales.

Editar archivo *hdfs-site.xml*

El archivo de *hdfs-site.xml* rige los parámetros críticos, como las rutas de almacenamiento de datos, la configuración de replicación y los tamaños de bloque, que

rigen el comportamiento y el rendimiento del clúster de HDFS. Configura el archivo definiendo los **directorios de almacenamiento NameNode y DataNode**. Además, el valor predeterminado de `dfs.replication` de 3 debes cambiarlo a 1 para que coincida con la configuración de un solo nodo. Pasos:

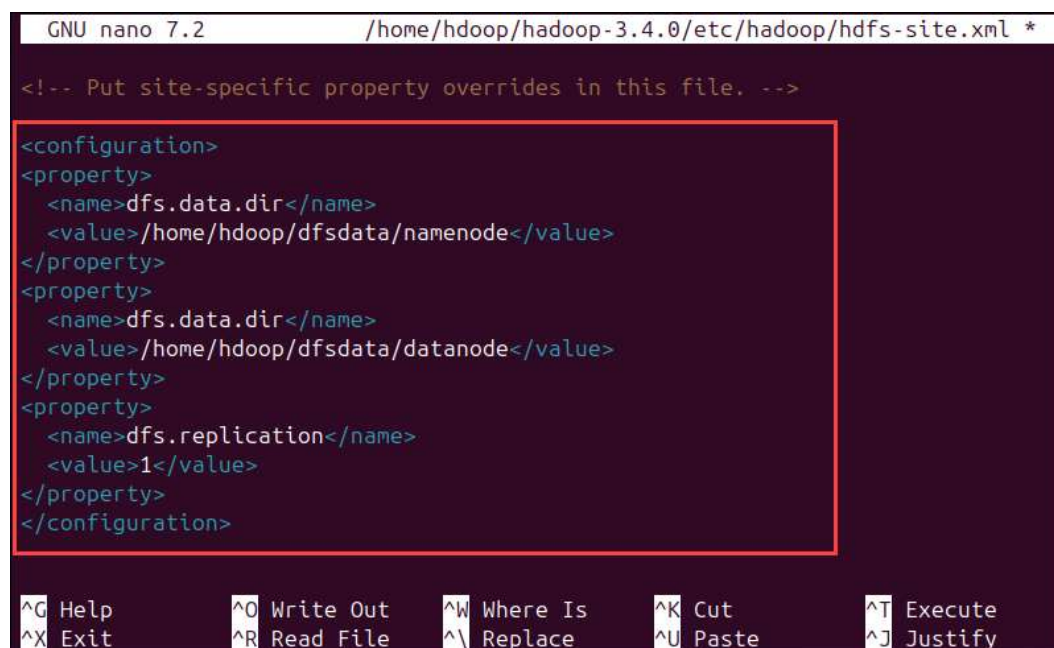
1. Utilice el siguiente comando para abrir el archivo `hdfs-site.xml` para editarlo:

```
sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

2. Agrega la siguiente configuración al archivo y, si es necesario, ajuste los directorios NameNode y DataNode a sus ubicaciones personalizadas:

```
<configuration>
<property>
  <name>dfs.data.dir</name>
  <value>/home/hadoop/dfsdata/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>/home/hadoop/dfsdata/datanode</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
</configuration>
```

Si es necesario, crea los directorios específicos que definió para el valor `dfs.data.dir`.



```
GNU nano 7.2 /home/hadoop/hadoop-3.4.0/etc/hadoop/hdfs-site.xml *
<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>dfs.data.dir</name>
  <value>/home/hadoop/dfsdata/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>/home/hadoop/dfsdata/datanode</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
</configuration>
```

Editar archivo mapred-site.xml

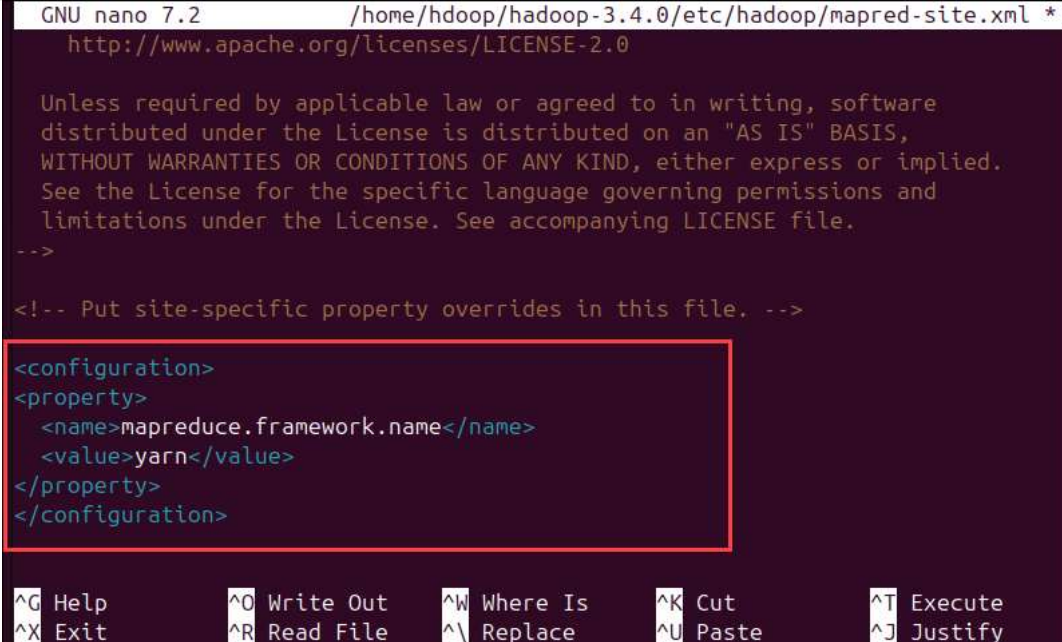
El archivo `mapred-site.xml` es un archivo de configuración que define la configuración del marco de trabajo de MapReduce, incluidos parámetros como la dirección del rastreador de trabajos, el número de tareas de asignación y reducción, y la administración de recursos, para controlar cómo se ejecutan los trabajos de MapReduce en el clúster. Pasos:

1. Utiliza el siguiente comando para acceder al *archivo mapred-site.xml* y **definir los valores de MapReduce**:

```
sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

2. Agrega la siguiente configuración para cambiar el valor predeterminado del nombre del marco de MapReduce a `yarn`:

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



```
GNU nano 7.2 /home/hadoop/hadoop-3.4.0/etc/hadoop/mapred-site.xml *
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

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

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

Editar archivo yarn-site.xml

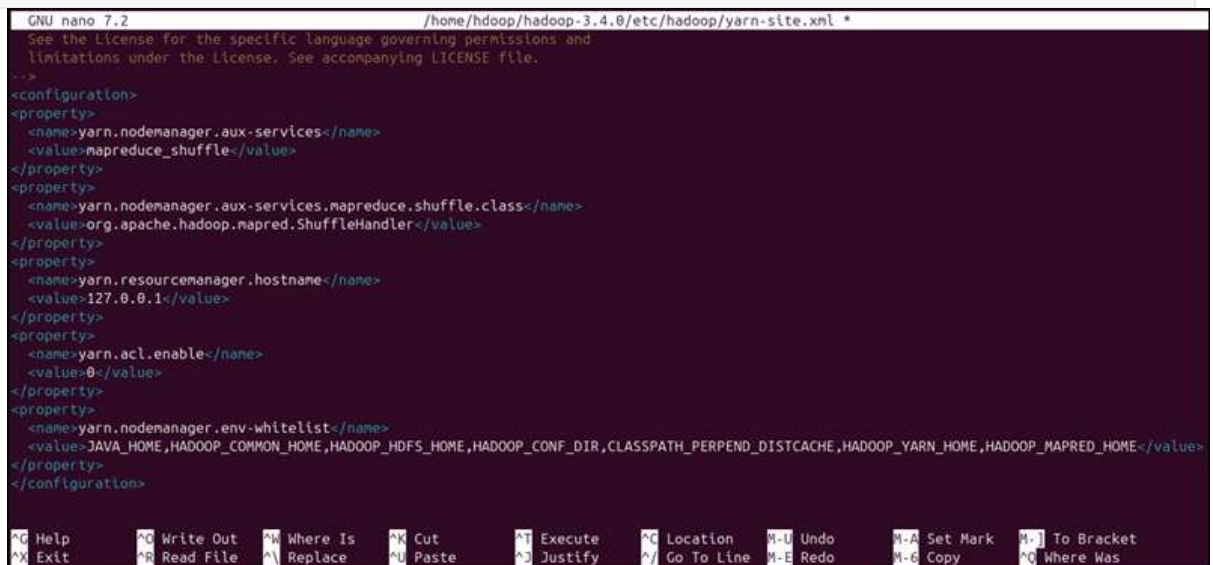
El archivo `yarn-site.xml` define la configuración de YARN. Contiene configuraciones para el **Administrador de nodos**, el **Administrador de recursos**, los **contenedores** y el **maestro de aplicaciones**. Pasos:

1. Abre el archivo *yarn-site.xml* en un editor de texto:

```
nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

2. Anexa la siguiente configuración al archivo:

```
<configuration>
<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>
<property>
  <name>yarn.resourcemanager.hostname</name>
  <value>127.0.0.1</value>
</property>
<property>
  <name>yarn.acl.enable</name>
  <value>0</value>
</property>
<property>
  <name>yarn.nodemanager.env-whitelist</name>
  <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PERPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
</property>
</configuration>
```



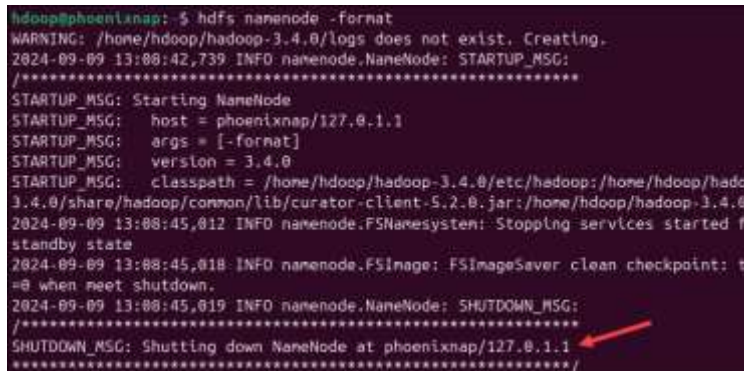
```
GNU nano 7.2 /home/hdoop/hadoop-3.4.0/etc/hadoop/yarn-site.xml *
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<configuration>
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>
  <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
  <name>yarn.resourcemanager.hostname</name>
  <value>127.0.0.1</value>
</property>
<property>
  <name>yarn.acl.enable</name>
  <value>0</value>
</property>
<property>
  <name>yarn.nodemanager.env-whitelist</name>
  <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PERPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
</property>
</configuration>
^G Help      ^O Write Out ^W Where Is  ^R Cut       ^T Execute  ^Q Location  ^U Undo      ^A Set Mark  ^] To Bracket
^X Exit      ^R Read File ^M Replace   ^U Paste     ^J Justify  ^_ Go To Line ^E Redo      ^G Copy      ^Q Where Was
```

Formatear HDFS NameNode

Es importante formatear el **NameNode** antes de iniciar los servicios de Hadoop por primera vez:

```
hdfs namenode -format
```

La notificación de apagado significa el final del proceso de formato NameNode.



```
hadoop@phoenixnap:~$ hdfs namenode -format
WARNING: /home/hadoop/hadoop-3.4.0/logs does not exist. Creating.
2024-09-09 13:08:42,739 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = phoenixnap/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.4.0
STARTUP_MSG:   classpath = /home/hadoop/hadoop-3.4.0/etc/hadoop:/home/hadoop/hadoop-3.4.0/share/hadoop/common/lib/curator-client-5.2.0.jar:/home/hadoop/hadoop-3.4.0
2024-09-09 13:08:45,012 INFO namenode.FSNamesystem: Stopping services started for standby state
2024-09-09 13:08:45,018 INFO namenode.FSImage: FSImageSaver clean checkpoint: t
=>0 when next shutdown.
2024-09-09 13:08:45,019 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at phoenixnap/127.0.1.1
*****/
```

Iniciar el clúster de Hadoop

El inicio de un clúster de Hadoop implica la inicialización de los servicios clave: HDFS para el almacenamiento distribuido y YARN para la administración de recursos. Esto permite que el sistema procese y almacene datos a gran escala en varios nodos. Pasos:

1. Navega hasta el *directorio* `hadoop-3.4.0/sbin` y ejecute el siguiente comando para iniciar NameNode y DataNode:

```
./start-dfs.sh
```

El sistema tarda algo en iniciar los nodos necesarios.



```
hadoop@phoenixnap:~/hadoop-3.4.0/sbin$ ./start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [phoenixnap]
```

2. Una vez que el *namenode*, los *datanodes* y el *namenode secundario* estén en funcionamiento, inicie el recurso YARN y los *administradores de nodos* escribiendo:

```
./start-yarn.sh
```

Al igual que con el comando anterior, la salida le informa de que los procesos se están iniciando.

```
hadoop@phoenixnap:~/hadoop-3.4.0/sbin$ ./start-yarn.sh
Starting resourcemanager
Starting nodemanagers
```

3. Ejecuta el siguiente comando para comprobar si todos los demonios están activos y ejecutándose como procesos Java:

```
jps
```

Si todo funciona según lo previsto, la lista resultante de procesos Java en ejecución contiene todos los demonios HDFS y YARN.

```
hadoop@phoenixnap:~/hadoop-3.4.0/sbin$ jps
45169 DataNode
46355 ResourceManager
45033 NameNode
46476 NodeManager
45373 SecondaryNameNode
47390 Jps
```

Si se desea detener el recurso YARN:

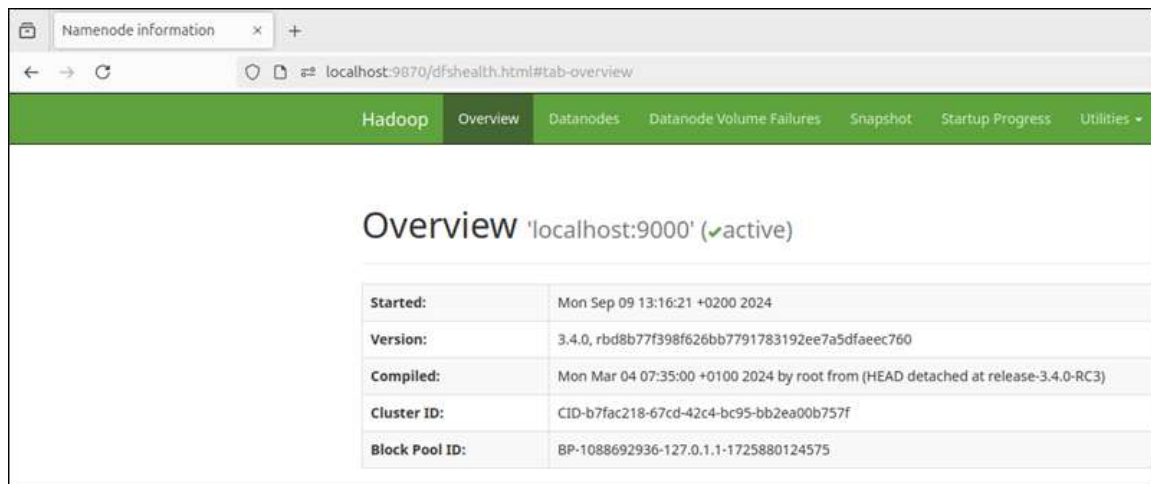
```
./stop-yarn.sh
```

Acceder a Hadoop desde el navegador

Utiliza un navegador y pon la URL o IP de su localhost. El número de puerto predeterminado 9870 da acceso a la interfaz de usuario de Hadoop NameNode:

```
http://localhost:9870
```

La interfaz de usuario de NameNode proporciona una descripción general completa de todo el clúster.



El **puerto predeterminado 9864** se utiliza para acceder a DataNodes individuales directamente desde su navegador:

<http://localhost:9864>



Se puede acceder al administrador de recursos de YARN en el **puerto 8088**:

<http://localhost:8088>

Resource Manager es una herramienta invaluable que le permite monitorear todos los procesos en ejecución en su clúster de Hadoop.



El **puerto 9870** es la versión actualizada de 50070 para acceder a la interfaz gráfica