Install JDK on Ubuntu

The Hadoop framework is written in Java, and its services require a compatible Java Runtime Environment (JRE) and Java Development Kit (JDK). Use the following command to update your system before initiating a new installation:

sudo apt updateCopy

At the moment, **Apache Hadoop 3.x fully supports Java 8** and **11**. The OpenJDK 8 package in Ubuntu contains both the runtime environment and development kit.

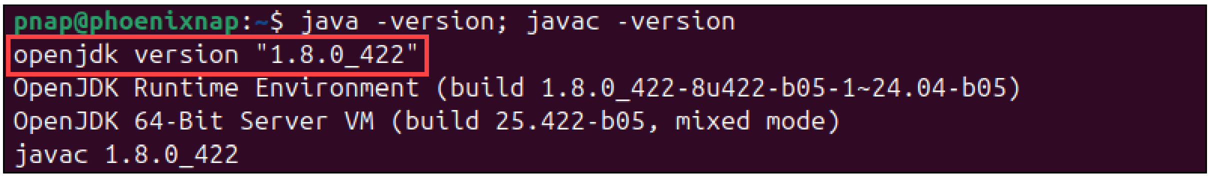
Type the following command in your terminal to install OpenJDK 8:

sudo apt install openjdk-8-jdk -yCopy

Note: The OpenJDK or Oracle Java version can affect how elements of a Hadoop ecosystem interact. To install a specific Java version, check out our detailed guide on how to install Java on Ubuntu.

Once the installation process is complete, verify the current Java version:

java -version; javac -versionCopy



```
pnap@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
```

The output informs you which Java version is in use.

Set Up Hadoop User and Configure SSH

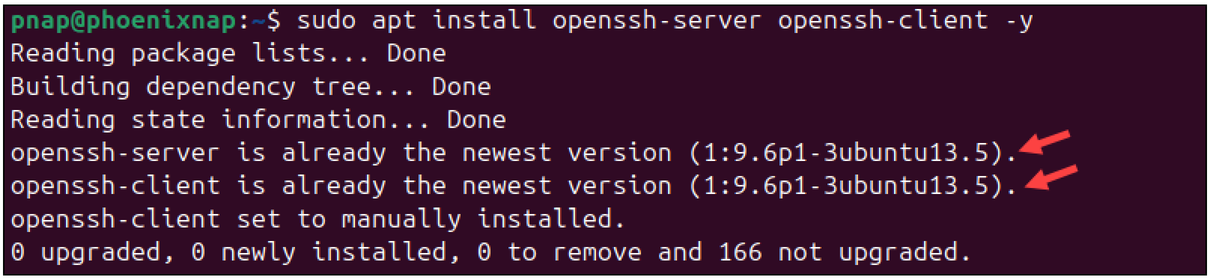
It is advisable to create a non-root user, specifically for the Hadoop environment. A distinct user improves security and helps you manage your cluster more efficiently. To ensure the smooth functioning of Hadoop services, the user should have the ability to establish a passwordless SSH connection with the localhost.

Install OpenSSH on Ubuntu

Install the OpenSSH server and client using the following command:

sudo apt install openssh-server openssh-client -yCopy

In the example below, the output confirms that the latest version is already installed.



```
pnap@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.
```

Note: If you have installed OpenSSH for the first time, use this opportunity to learn about Linux SSH security best practices.

Create Hadoop User

Utilize the adduser command to create a new Hadoop user:

sudo adduser hdoopCopy

The username, in this example, is **hdoop**. You are free to use any username and password you see fit.

Tip: Check out our strong password ideas or try our free password generator.

Switch to the newly created user and enter the corresponding password:

su - hdoopCopy

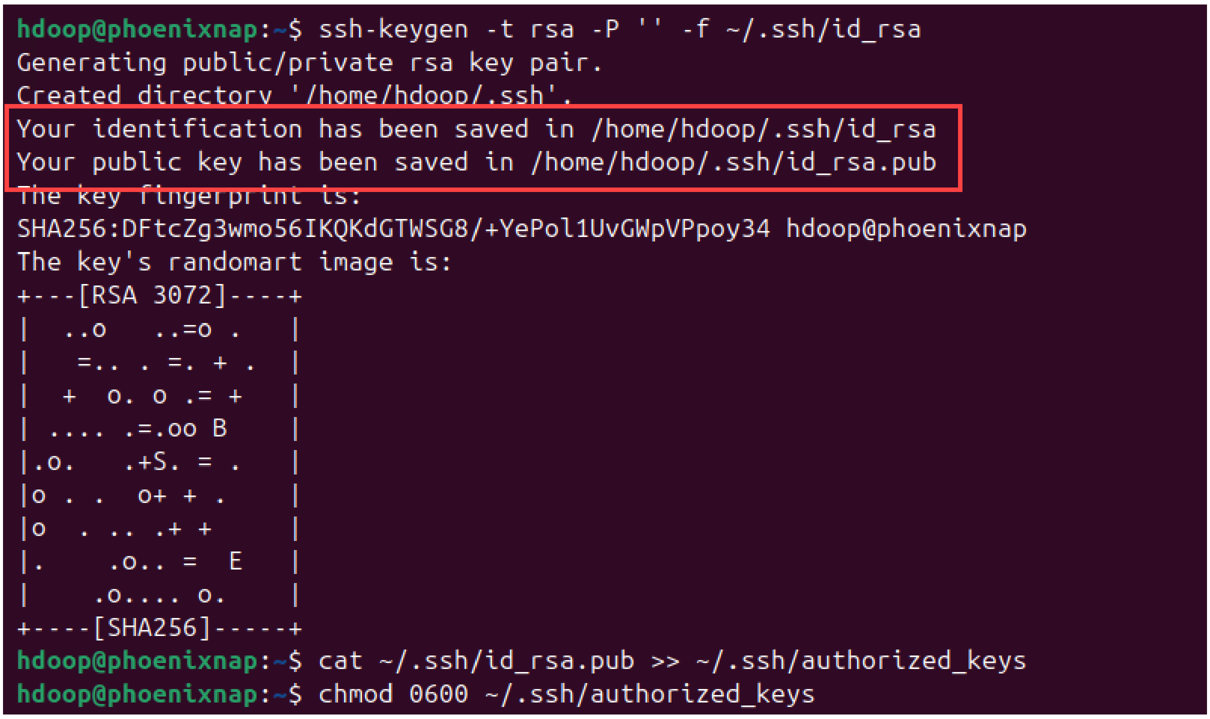
The user now needs to be able to SSH to the localhost without being prompted for a password.

Enable Passwordless SSH for Hadoop User

Generate an SSH key pair and define the location it is to be stored in:

ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsaCopy

The system proceeds to generate and save the SSH key pair.



```
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
```

Use the cat command to store the public key as **authorized_keys** in the *ssh* directory:

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keysCopy

Set the file permissions for your user with the **chmod** command:

chmod 0600 ~/.ssh/authorized_keysCopy

The new user can now SSH without entering a password every time. Verify everything is set up correctly by using the **hdoop** user to SSH to localhost:

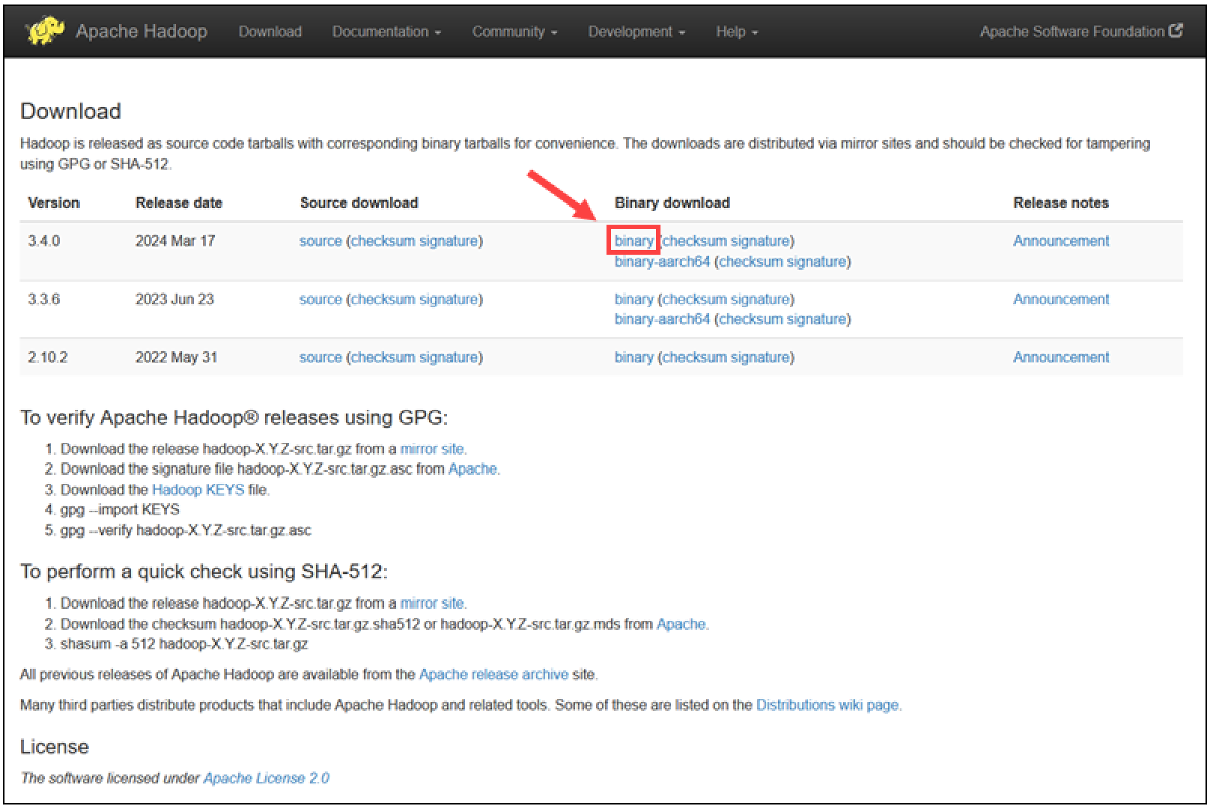
ssh localhostCopy

After an initial prompt, the Hadoop user can seamlessly establish an SSH connection to the localhost.

Download and Install Hadoop on Ubuntu

After configuring the Hadoop user, you are ready to install Hadoop on your system. Follow the steps below:

1. Visit the [official Apache Hadoop project page]() and select the version of Hadoop you want to implement.



Download

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mirror sites and should be checked for tampering using GPG or SHA-512.

Version	Release date	Source download	Binary download	Release notes
3.4.0	2024 Mar 17	source (checksum signature)	binary (checksum signature) binary-aarch64 (checksum signature)	Announcement
3.3.6	2023 Jun 23	source (checksum signature)	binary (checksum signature) binary-aarch64 (checksum signature)	Announcement
2.10.2	2022 May 31	source (checksum signature)	binary (checksum signature)	Announcement

To verify Apache Hadoop® releases using GPG:

1. Download the release hadoop-X.Y.Z-src.tar.gz from a mirror site.
2. Download the signature file hadoop-X.Y.Z-src.tar.gz.asc from Apache.
3. Download the Hadoop KEYS file.
4. gpg --import KEYS
5. gpg --verify hadoop-X.Y.Z-src.tar.gz.asc

To perform a quick check using SHA-512:

1. Download the release hadoop-X.Y.Z-src.tar.gz from a mirror site.
2. Download the checksum hadoop-X.Y.Z-src.tar.gz.sha512 or hadoop-X.Y.Z-src.tar.gz.mds from Apache.
3. shasum -a 512 hadoop-X.Y.Z-src.tar.gz

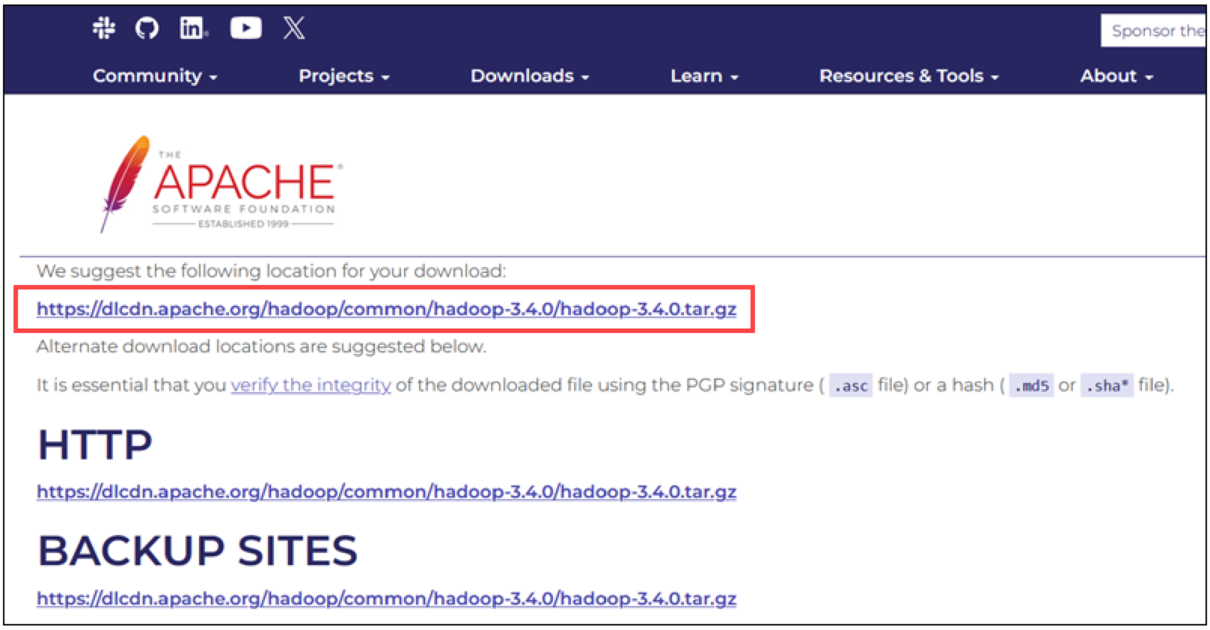
All previous releases of Apache Hadoop are available from the Apache release archive site.

Many third parties distribute products that include Apache Hadoop and related tools. Some of these are listed on the Distributions wiki page.

License

The software licensed under Apache License 2.0

The steps outlined in this tutorial use the [binary]() download for **Hadoop Version 3.4.0**. Select your preferred option, and you will be presented with a mirror link to download the **Hadoop** tar package.



We suggest the following location for your download:

https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz

Alternate download locations are suggested below.

It is essential that you verify the integrity of the downloaded file using the PGP signature (`.asc` file) or a hash (`.md5` or `.sha*` file).

HTTP

https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz

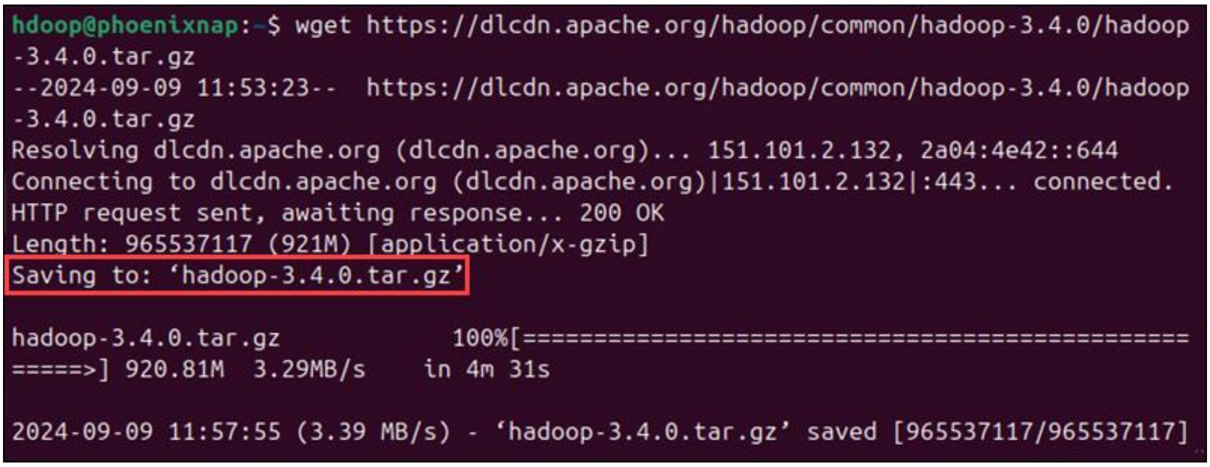
BACKUP SITES

https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz

Note: It is sound practice to verify Hadoop downloads originating from mirror sites. The instructions for using GPG or SHA-512 for verification are provided on the official download page.

2. Use the provided mirror link and download the Hadoop package using the wget command:

wget https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gzCopy



```
hdoop@phoenixnap:~$ wget https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop
-3.4.0.tar.gz
--2024-09-09 11:53:23--  https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop
-3.4.0.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.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. Once the download completes, use the tar command to extract the .tar.gz file and initiate the Hadoop installation:

tar xzf hadoop-3.4.0.tar.gzCopy

The Hadoop binary files are now located within the *hadoop-3.4.0* directory.

Single Node Hadoop Deployment (Pseudo-Distributed Mode)

Hadoop excels when deployed in a **fully distributed mode** on a large cluster of networked servers. However, if you are new to Hadoop and want to explore basic commands or test applications, you can configure Hadoop on a single node.

This setup, also called **pseudo-distributed mode**, allows each Hadoop daemon to run as a single Java process. Configure a Hadoop environment by editing a set of configuration files:

- .bashrc
- *hadoop-env.sh*
- *core-site.xml*
- *hdfs-site.xml*
- *mapred-site-xml*
- *yarn-site.xml*

Configure Hadoop Environment Variables (bashrc)

The *.bashrc* config file is a shell script that initializes user-specific settings, such as environment variables, aliases, and functions, every time a new Bash shell session is started. Follow the steps below to configure Hadoop environment variables:

1. Edit the *.bashrc* shell configuration file using a text editor of your choice (we will use nano):

nano .bashrcCopy

2. Define the Hadoop environment variables by adding the following content to the end of the file:

#Hadoop Related Options

export HADOOP_HOME=/home/hdoop/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

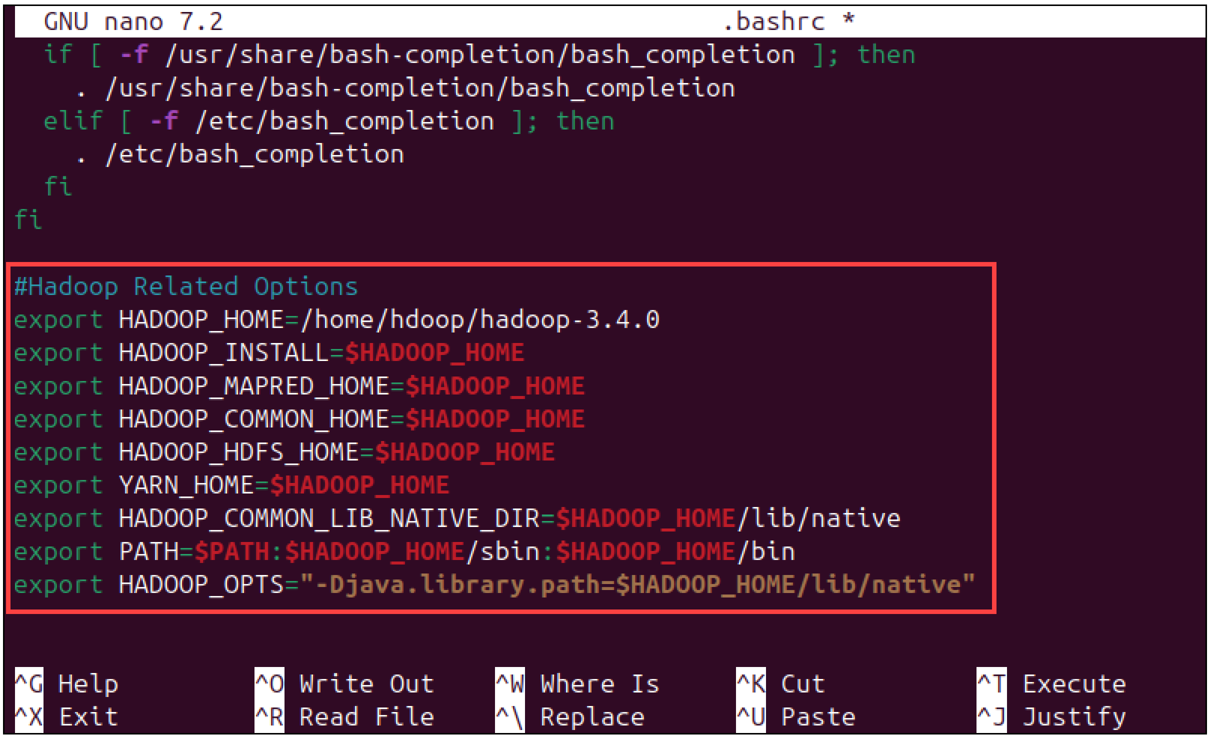
export YARN_HOME=$HADOOP_HOME

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"Copy

3. Once you add the variables, save and exit the *.bashrc* file.



```
  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
fi

#Hadoop Related Options
export HADOOP_HOME=/home/hdoop/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. Run the command below to apply the changes to the current running environment:

source ~/.bashrcCopy

Edit hadoop-env.sh File

The *hadoop-env.sh* file serves as a master file to configure YARN, HDFS, MapReduce, and Hadoop-related project settings. When setting up a **single-node Hadoop cluster**, you need to define which Java implementation will be utilized.

Follow the steps below:

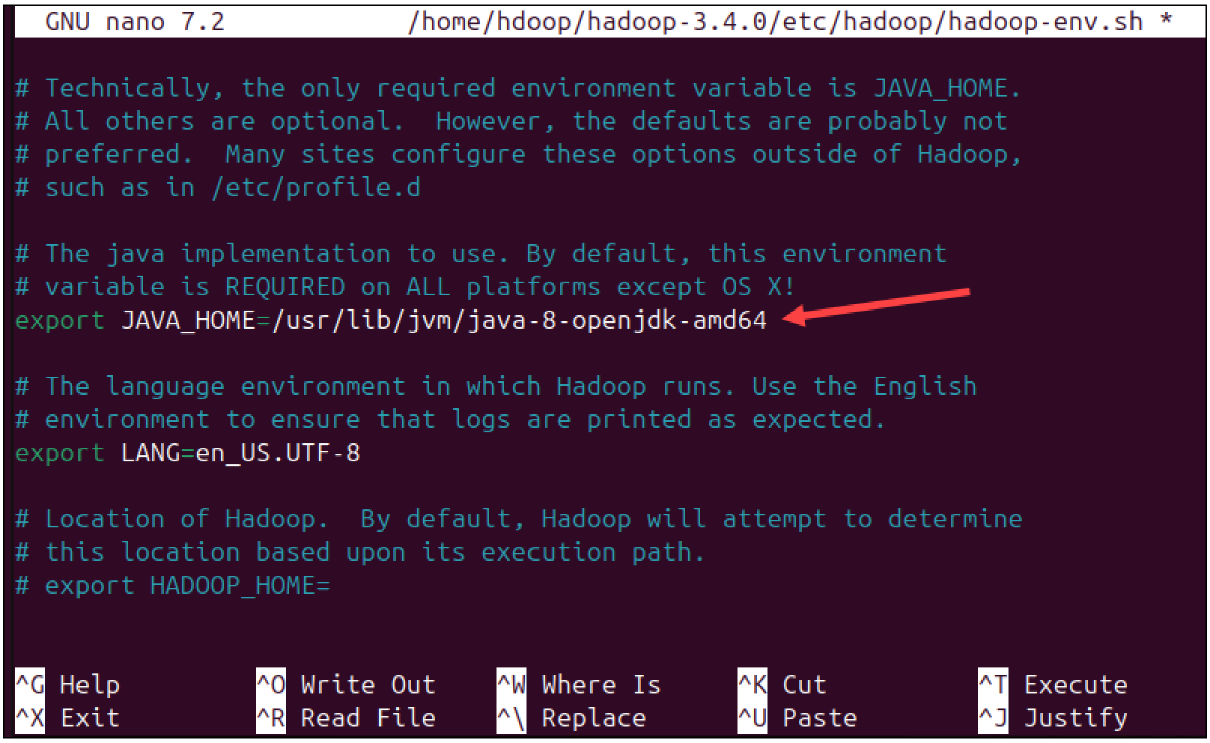
1. Use the previously created **$HADOOP_HOME** variable to access the *hadoop-env.sh* file:

nano $HADOOP_HOME/etc/hadoop/hadoop-env.shCopy

2. Uncomment the **$JAVA_HOME** variable (i.e., remove the **#** sign) and add the full path to the OpenJDK installation on your system. If you have installed the same version as presented in the first part of this tutorial, add the following line:

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64Copy

The path needs to match the location of the Java installation on your system.

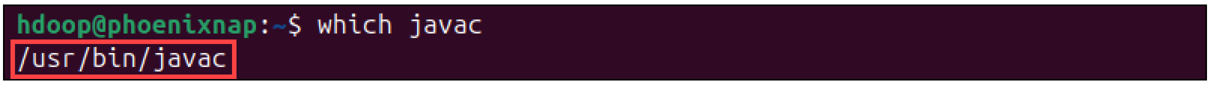


```
  GNU nano 7.2                /home/hdoop/hadoop-3.4.0/etc/hadoop/hadoop-env.sh *

# 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=
```

If you need help to locate the correct Java path, run the following command in your terminal window:

which javacCopy

The resulting output provides the path to the Java binary directory.

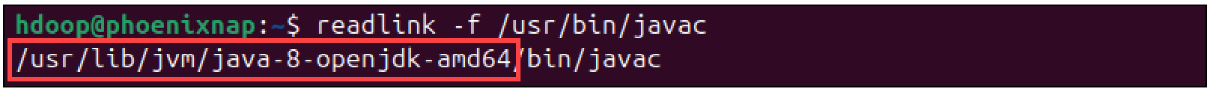


```
hdoop@phoenixnap:~$ which javac
/usr/bin/javac
```

3. Use the provided path to find the OpenJDK directory with the following command:

readlink -f /usr/bin/javacCopy

The section of the path just before the */bin/javac* directory needs to be assigned to the **$JAVA_HOME** variable.



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

Edit core-site.xml File

The *core-site.xml* file defines HDFS and Hadoop core properties. To set up Hadoop in a pseudo-distributed mode, you need to specify the **URL** for your NameNode, and the temporary directory Hadoop uses for the map and reduce process.

The steps below show how to configure the file.

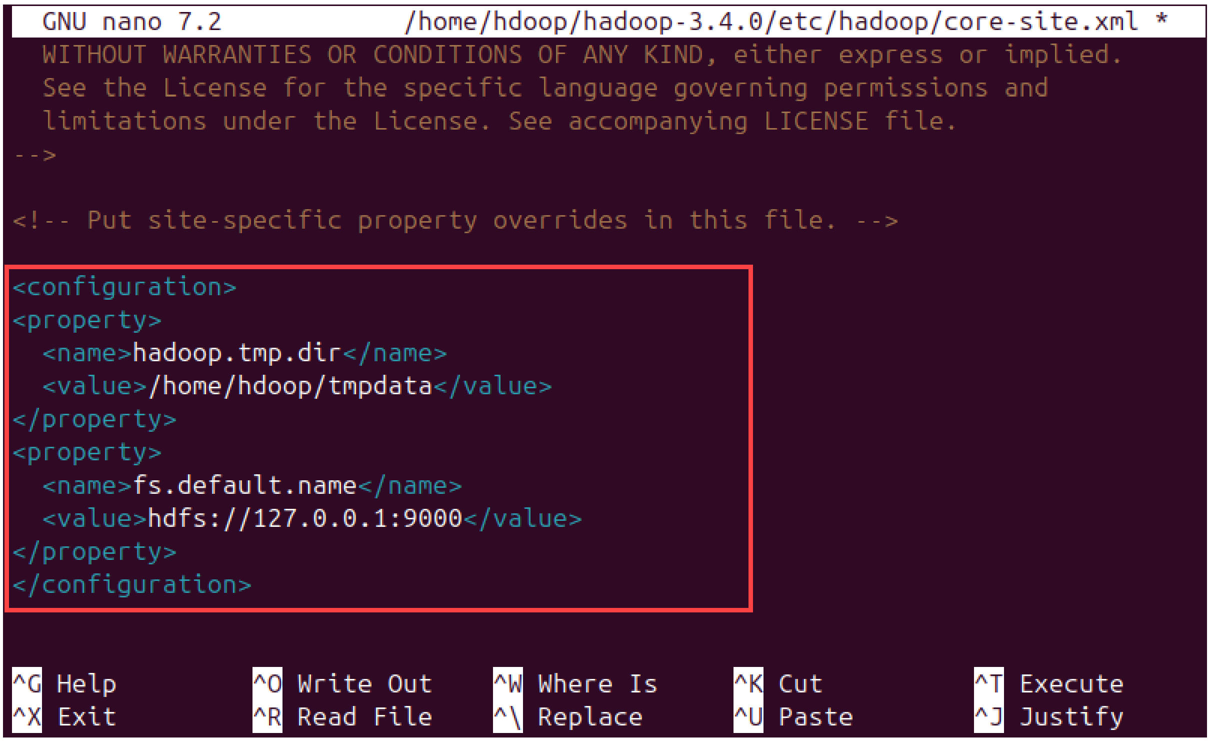
1. Open the *core-site.xml* file in a text editor:

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

2. Add the following configuration to override the default values for the temporary directory and add your HDFS URL to replace the default local file system setting:

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

This example uses values specific to the local system. You should use values that match your system's requirements. The data needs to be consistent throughout the configuration process.



```
  GNU nano 7.2                /home/hdoop/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/hdoop/tmpdata</value>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://127.0.0.1:9000</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
```

Do not forget to create a directory in the location you specified for your temporary data.

Edit hdfs-site.xml File

The *hdfs-site.xml* file governs specifies critical parameters, such as data storage paths, replication settings, and block sizes, which govern the behavior and performance of the HDFS cluster. Configure the file by defining the **NameNode** and **DataNode storage directories**. Additionally, the default **dfs.replication** value of **3** needs to be changed to **1** to match the single-node setup.

Follow the steps below:

1. Use the following command to open the *hdfs-site.xml* file for editing:

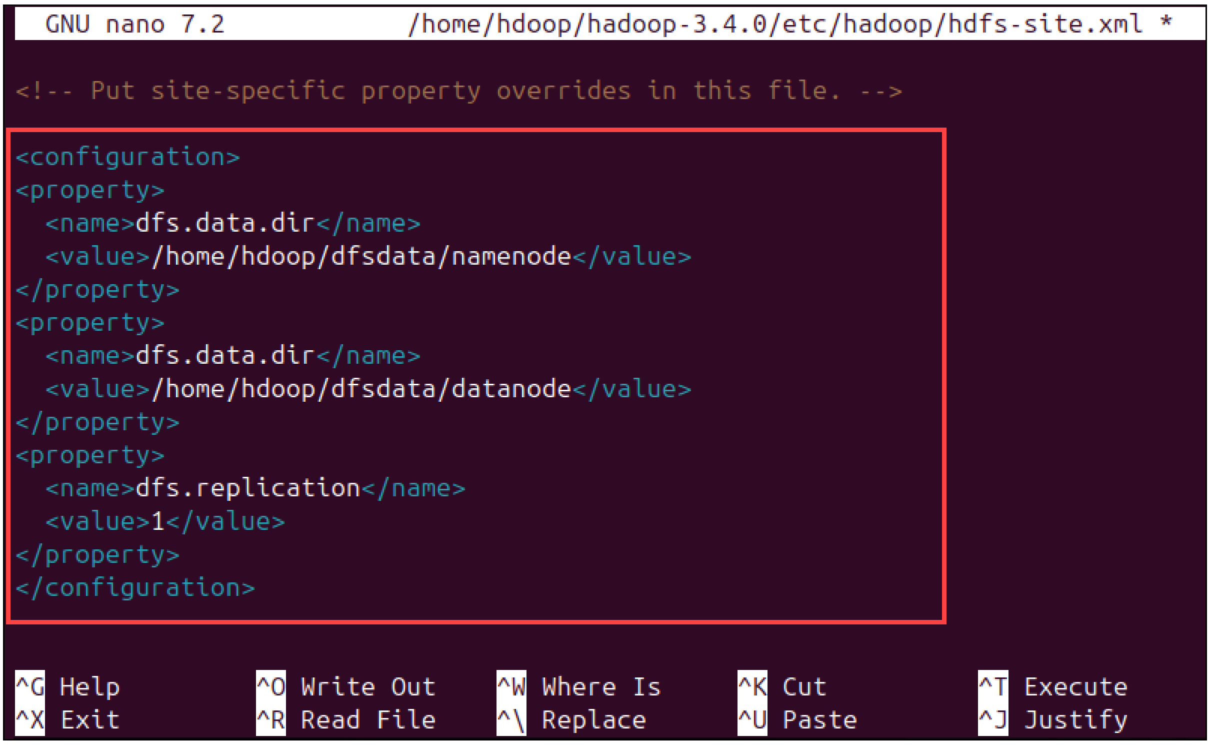
```
sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xmlCopy
```

2. Add the following configuration to the file and, if needed, adjust the NameNode and DataNode directories to your custom locations:

```
<configuration>

<property>

  <name>dfs.data.dir</name>

  <value>/home/hdoop/dfsdata/namenode</value>

</property>

<property>

  <name>dfs.data.dir</name>

  <value>/home/hdoop/dfsdata/datanode</value>

</property>
```

```
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
</configuration>Copy
```

If necessary, create the specific directories you defined for the **dfs.data.dir** value.



```
  GNU nano 7.2             /home/hdoop/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/hdoop/dfsdata/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>/home/hdoop/dfsdata/datanode</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
</configuration>
```

Edit mapred-site.xml File

The *mapred-site.xml* file is a configuration file that defines settings for the MapReduce framework, including parameters such as the job tracker address, the number of map and reduce tasks, and resource management, to control how MapReduce jobs are executed across the cluster.

Follow the steps below to configure the *mapred-site.xml* file:

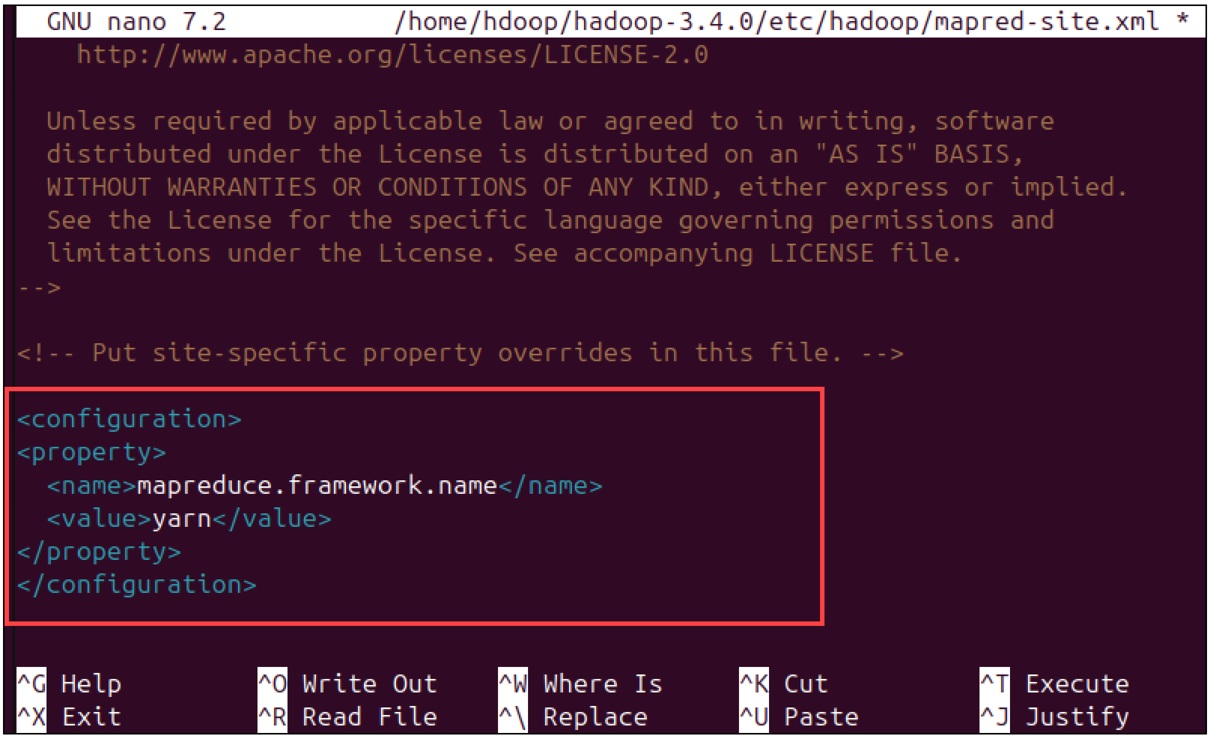
1. Use the following command to access the *mapred-site.xml* file and **define MapReduce values**:

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

2. Add the following configuration to change the default MapReduce framework name value to **yarn**:

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

</configuration>Copy



```
  GNU nano 7.2              /home/hdoop/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>
```

Edit yarn-site.xml File

The *yarn-site.xml* file defines YARN settings. It contains configurations for the **Node Manager, Resource Manager, Containers,** and **Application Master**.

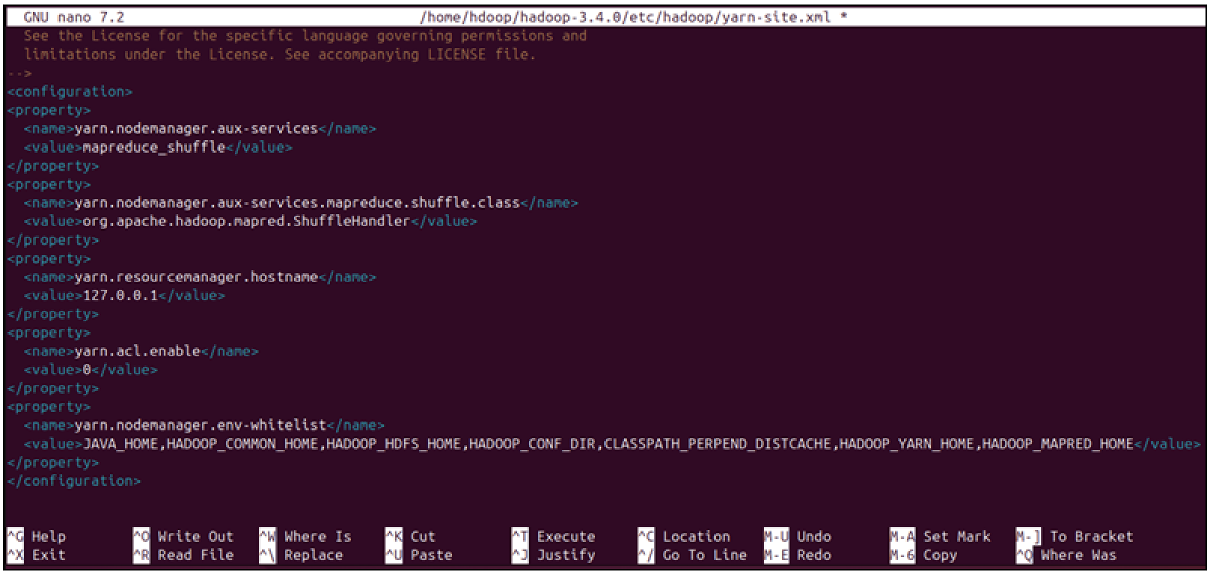
1. Open the *yarn-site.xml* file in a text editor:

nano $HADOOP_HOME/etc/hadoop/yarn-site.xmlCopy

2. Append the following configuration to the 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>Copy
```



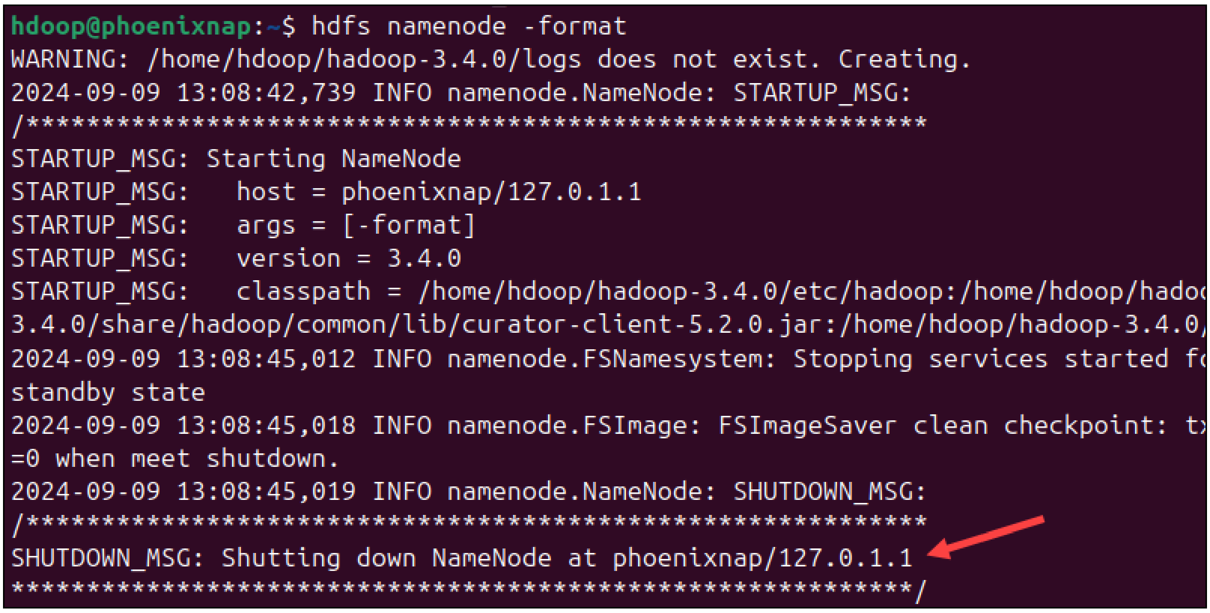
```
  GNU nano 7.2                          /home/hdoop/hadoop-3.4.0/etc/hadoop/yarn-site.xml *
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>
<property>
  <name>yarn.nodemanager.aux-services</name>
  <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>
```

Format HDFS NameNode

It is important to **format the NameNode** before starting Hadoop services for the first time:

```
hdfs namenode -formatCopy
```

The shutdown notification signifies the end of the NameNode format process.



```
hdoop@phoenixnap:~$ hdfs namenode -format
WARNING: /home/hdoop/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/hdoop/hadoop-3.4.0/etc/hadoop:/home/hdoop/hado
3.4.0/share/hadoop/common/lib/curator-client-5.2.0.jar:/home/hdoop/hadoop-3.4.0/
2024-09-09 13:08:45,012 INFO namenode.FSNamesystem: Stopping services started fo
standby state
2024-09-09 13:08:45,018 INFO namenode.FSImage: FSImageSaver clean checkpoint: tx
=0 when meet shutdown.
2024-09-09 13:08:45,019 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at phoenixnap/127.0.1.1
************************************************************/
```

Start Hadoop Cluster

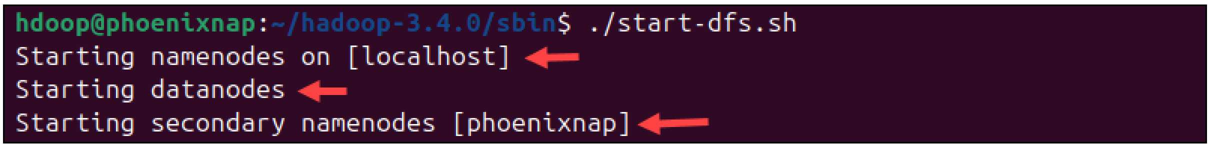
Starting a Hadoop cluster involves initializing the key services - HDFS for distributed storage and YARN for resource management. This enables the system to process and store large-scale data across multiple nodes.

Follow the steps below:

1. Navigate to the *hadoop-3.4.0/sbin* directory and execute the following command to start the NameNode and DataNode:

./start-dfs.shCopy

The system takes a few moments to initiate the necessary nodes.

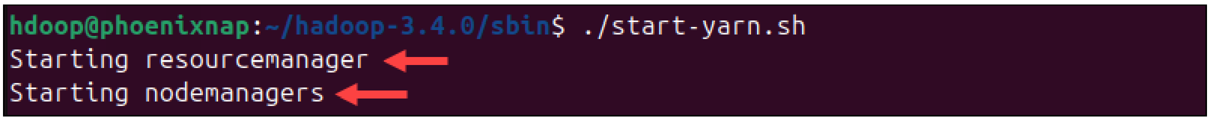


```
hdoop@phoenixnap:~/hadoop-3.4.0/sbin$ ./start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [phoenixnap]
```

2. Once the *namenode*, *datanodes*, and *secondary namenode* are up and running, start the YARN resource and *nodemanagers* by typing:

./start-yarn.shCopy

As with the previous command, the output informs you that the processes are starting.

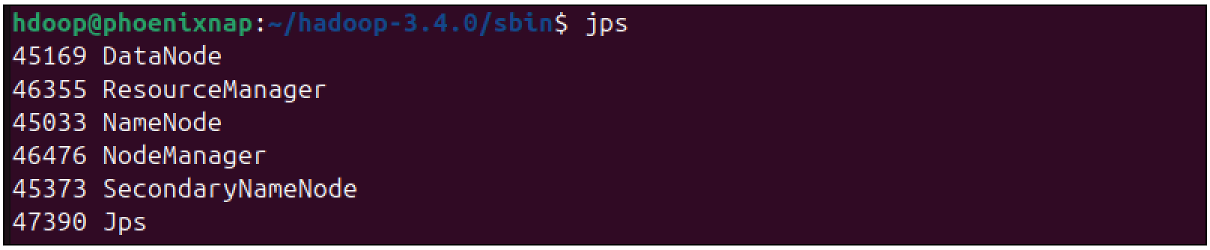


```
hdoop@phoenixnap:~/hadoop-3.4.0/sbin$ ./start-yarn.sh
Starting resourcemanager
Starting nodemanagers
```

3. Run the following command to check if all the daemons are active and running as Java processes:

jpsCopy

If everything works as intended, the resulting list of running Java processes contains all the HDFS and YARN daemons.



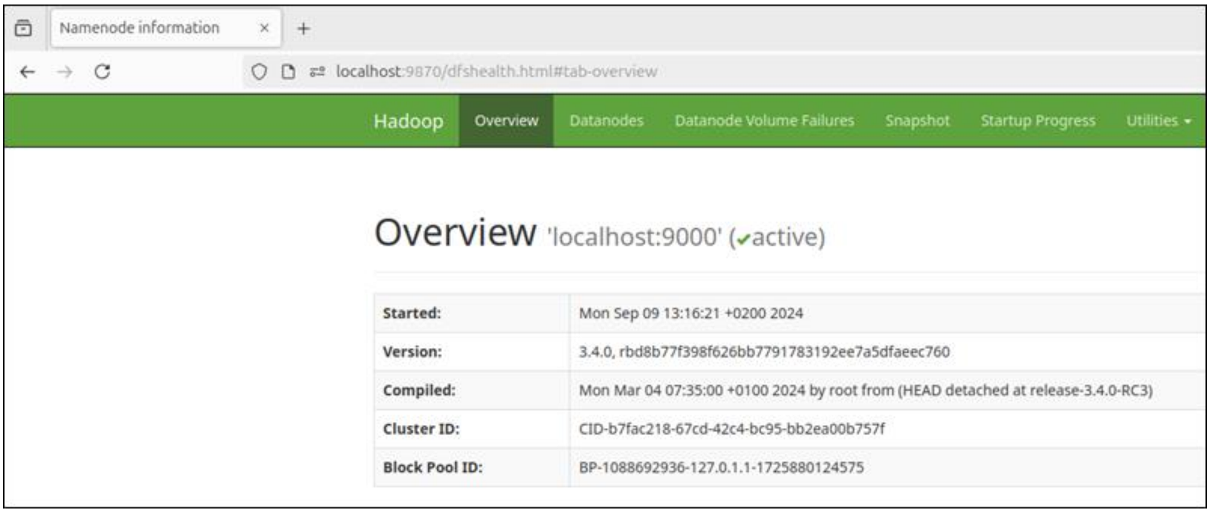
```
hdoop@phoenixnap:~/hadoop-3.4.0/sbin$ jps
45169 DataNode
46355 ResourceManager
45033 NameNode
46476 NodeManager
45373 SecondaryNameNode
47390 Jps
```

Access Hadoop from Browser

Use your preferred browser and navigate to your localhost URL or IP. The default port number **9870** gives you access to the Hadoop NameNode UI:

http://localhost:9870Copy

The NameNode user interface provides a comprehensive overview of the entire cluster.

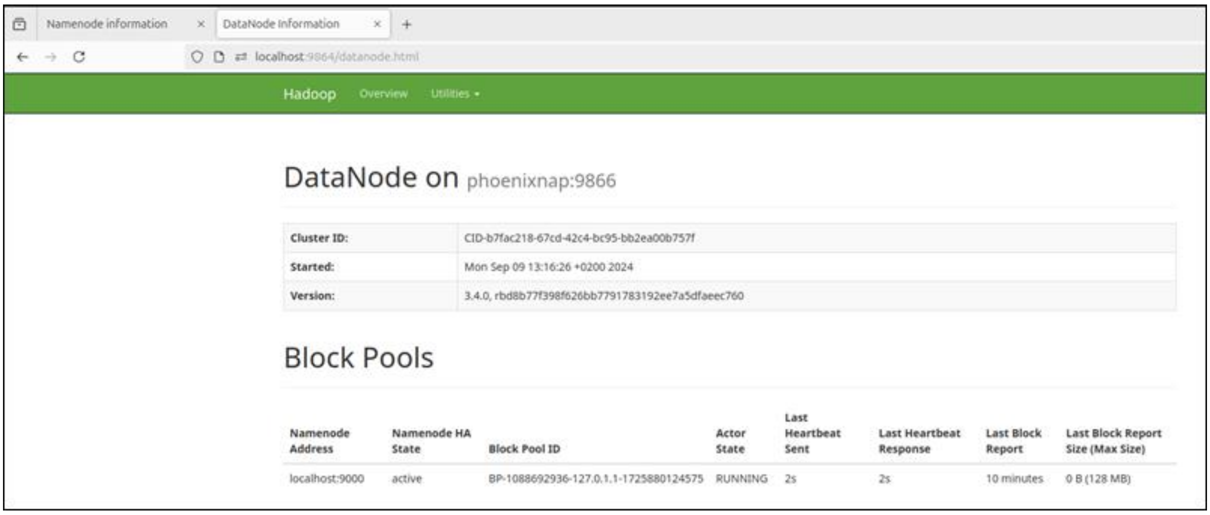


Overview 'localhost:9000' (✓active)

Started:	Mon Sep 09 13:16:21 +0200 2024
Version:	3.4.0, rbd8b77f398f626bb7791783192ee7a5dfaeec760
Compiled:	Mon Mar 04 07:35:00 +0100 2024 by root from (HEAD detached at release-3.4.0-RC3)
Cluster ID:	CID-b7fac218-67cd-42c4-bc95-bb2ea00b757f
Block Pool ID:	BP-1088692936-127.0.1.1-1725880124575

The default port **9864** is used to access individual DataNodes directly from your browser:

http://localhost:9864Copy



DataNode on phoenixnap:9866

Cluster ID:	CID-b7fac218-67cd-42c4-bc95-bb2ea00b757f
Started:	Mon Sep 09 13:16:26 +0200 2024
Version:	3.4.0, rbd8b77f398f626bb7791783192ee7a5dfaeec760

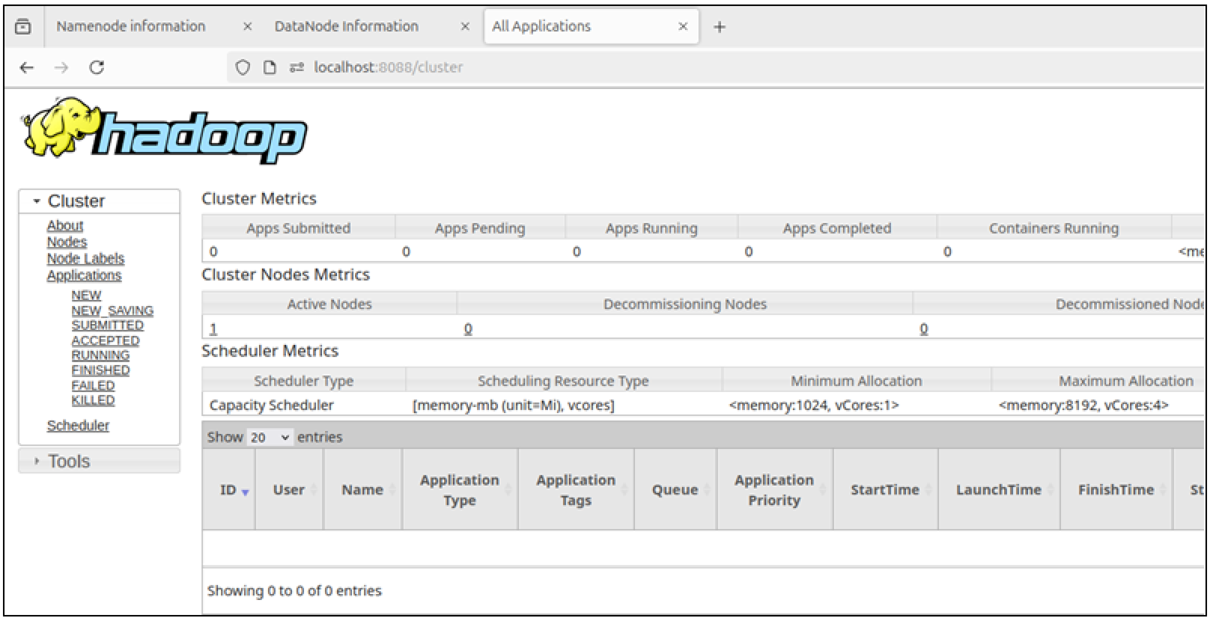
Block Pools

Namenode Address	Namenode HA State	Block Pool ID	Actor State	Last Heartbeat Sent	Last Heartbeat Response	Last Block Report	Last Block Report Size (Max Size)
localhost:9000	active	BP-1088692936-127.0.1.1-1725880124575	RUNNING	2s	2s	10 minutes	0 B (128 MB)

The YARN Resource Manager is accessible on port **8088**:

http://localhost:8088Copy

The Resource Manager is an invaluable tool that allows you to monitor all running processes in your Hadoop cluster.



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	
0	0	0	0	0	<me

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Node
1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	St

Showing 0 to 0 of 0 entries

```
pip install notebook
jupyter notebook
pip install hdfs
```