Hadoop Installation

There are two ways to install Hadoop, i.e. **Single node** and **Multi-node**.

A single node cluster means only one DataNode running and setting up all the NameNode, DataNode, ResourceManager, and NodeManager on a single machine. This is used for studying and testing purposes. For example, let us consider a sample data set inside the healthcare industry. So, for testing whether the Oozie jobs have scheduled all the processes like collecting, aggregating, storing, and processing the data in a proper sequence, we use a single node cluster. It can easily and efficiently test the sequential workflow in a smaller environment as compared to large environments which contain terabytes of data distributed across hundreds of machines.

While in a **Multi-node cluster**, there are more than one DataNode running and each DataNode is running on different machines. The multi-node cluster is practically used in organizations for analyzing Big Data. Considering the above example, in real-time when we deal with petabytes of data, it needs to be distributed across hundreds of machines to be processed. Thus, here we use a multi-node cluster.

Prerequisites

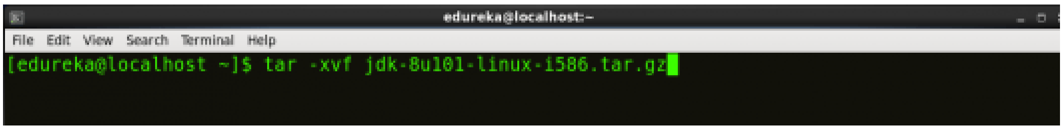
- *VIRTUAL BOX*: it is used for installing the operating system on it.
- *OPERATING SYSTEM*: You can install Hadoop on Linux-based operating systems. Ubuntu and CentOS are very commonly used. In this tutorial, we are using CentOS.
- *JAVA*: You need to install the Java 8 package on your system.
- *HADOOP*: You require Hadoop 2.7.3 package.

Install Hadoop

Single Node cluster

Step 1: **Click here to download the Java 8 Package. Save this file in your home directory.**
Step 2: **Extract the Java Tar File.**
***Command*:** tar -xvf jdk-8u101-linux-i586.tar.gz

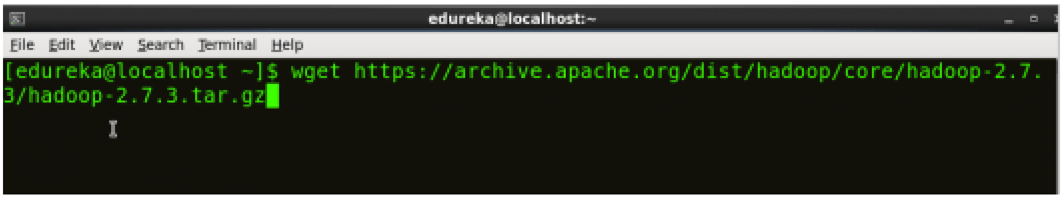
Hadoop Installation



```
[edureka@localhost ~]$ tar -xvf jdk-8u101-linux-i586.tar.gz
```

Fig: Hadoop Installation – Extracting Java Files

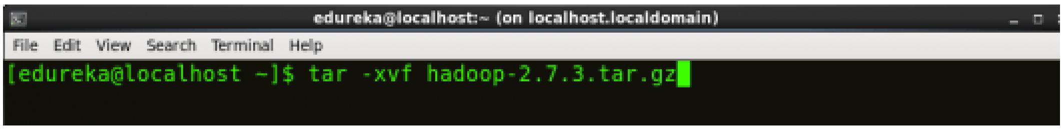
Step 3: **Download the Hadoop 2.7.3 Package.**
***Command*:** wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz



```
[edureka@localhost ~]$ wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.
3/hadoop-2.7.3.tar.gz
```

Fig: Hadoop Installation – Downloading Hadoop

Step 4: **Extract the Hadoop tar File.**
Command: tar -xvf hadoop-2.7.3.tar.gz


edureka@localhost:~ (on localhost.localdomain)
File Edit View Search Terminal Help

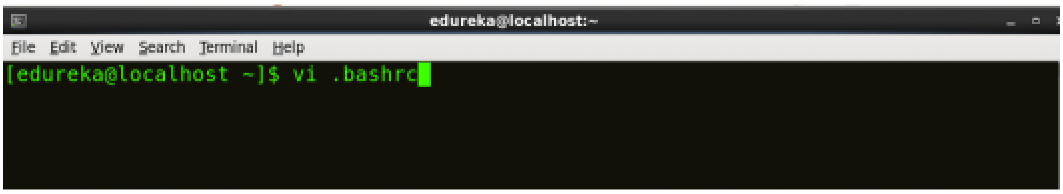
```
[edureka@localhost ~]$ tar -xvf hadoop-2.7.3.tar.gz
```

Fig: Hadoop Installation – Extracting Hadoop Files

Step 5: **Add the Hadoop and Java paths in the bash file (.bashrc).**
Open**. bashrc** file. Now, add Hadoop and Java Path as shown below.

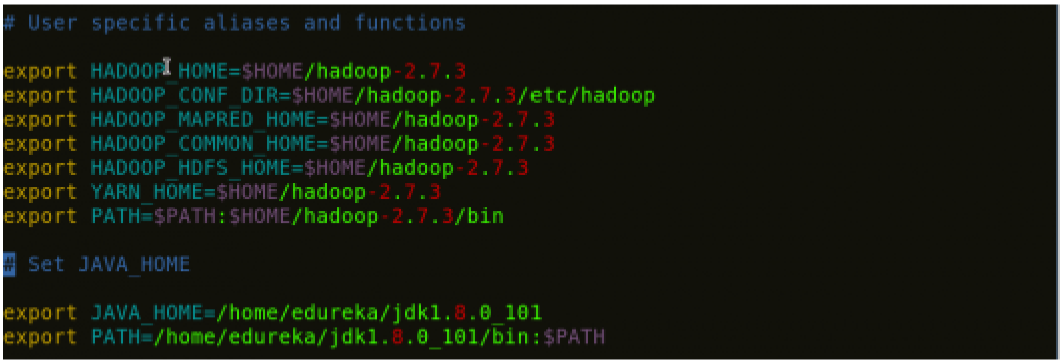
Learn more about the Hadoop Ecosystem and its tools with the Hadoop Certification.

***Command*:** vi .bashrc



```
[edureka@localhost ~]$ vi .bashrc
```

Hadoop Installation



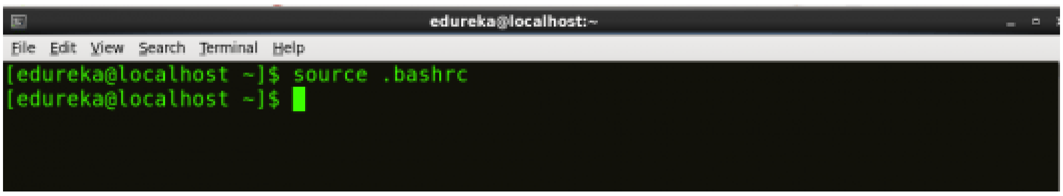
```
# User specific aliases and functions

export HADOOP_HOME=$HOME/hadoop-2.7.3
export HADOOP_CONF_DIR=$HOME/hadoop-2.7.3/etc/hadoop
export HADOOP_MAPRED_HOME=$HOME/hadoop-2.7.3
export HADOOP_COMMON_HOME=$HOME/hadoop-2.7.3
export HADOOP_HDFS_HOME=$HOME/hadoop-2.7.3
export YARN_HOME=$HOME/hadoop-2.7.3
export PATH=$PATH:$HOME/hadoop-2.7.3/bin

# Set JAVA_HOME

export JAVA_HOME=/home/edureka/jdk1.8.0_101
export PATH=/home/edureka/jdk1.8.0_101/bin:$PATH
```

Fig: Hadoop Installation – Setting Environment Variable

Then, save the bash file and close it.

For applying all these changes to the current Terminal, execute the source command.

Command: source .bashrc

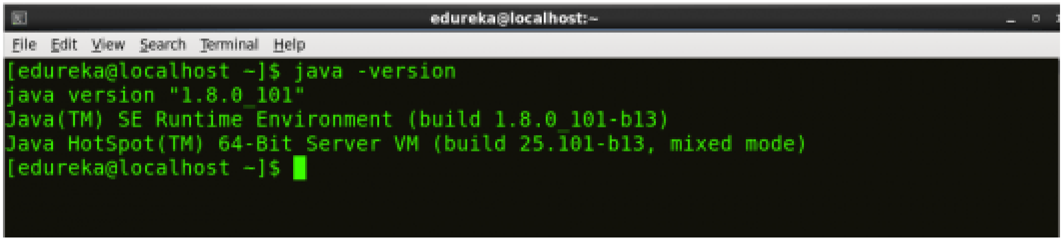


```
[edureka@localhost ~]$ source .bashrc
[edureka@localhost ~]$ 
```

Fig: Hadoop Installation – Refreshing environment variables

To make sure that Java and Hadoop have been properly installed on your system and can be accessed through the Terminal, execute the java -version and hadoop version commands.

Command: java -version

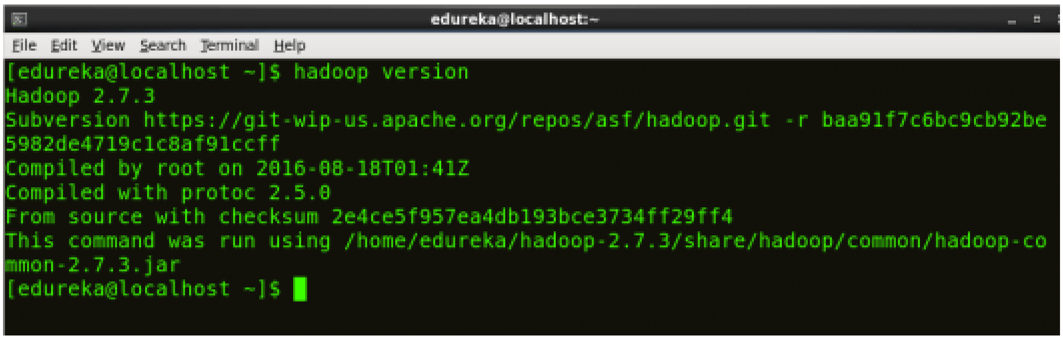


```
[edureka@localhost ~]$ java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)
[edureka@localhost ~]$ 
```

Fig: Hadoop Installation – Checking Java Version

Command: hadoop version

Hadoop Installation



```
[edureka@localhost ~]$ hadoop version
Hadoop 2.7.3
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r baa91f7c6bc9cb92be
5982de4719c1c8af91ccff
Compiled by root on 2016-08-18T01:41Z
Compiled with protoc 2.5.0
From source with checksum 2e4ce5f957ea4db193bce3734ff29ff4
This command was run using /home/edureka/hadoop-2.7.3/share/hadoop/common/hadoop-co
mmon-2.7.3.jar
[edureka@localhost ~]$
```

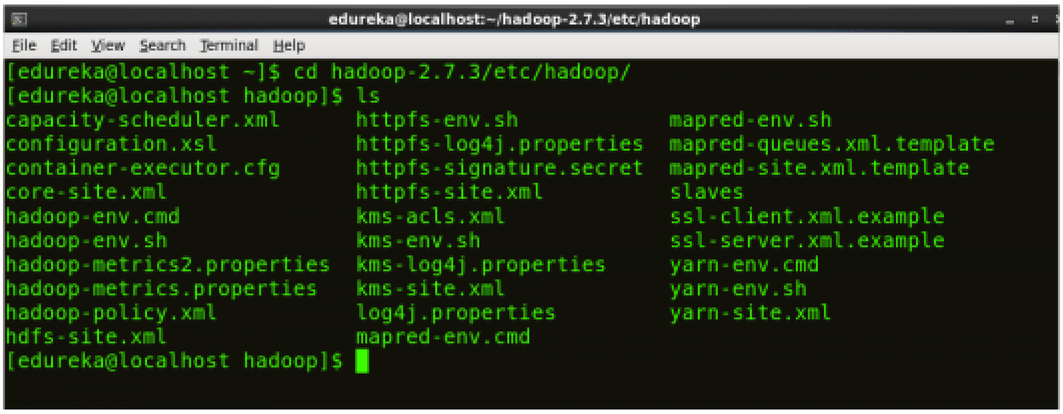
Fig: Hadoop Installation – Checking Hadoop Version

Step 6: **Edit the** Hadoop Configuration files**.**

Command: cd hadoop-2.7.3/etc/hadoop/

Command: ls

All the Hadoop configuration files are located in **hadoop-2.7.3/etc/hadoop** directory as you can see in the snapshot below:


edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help

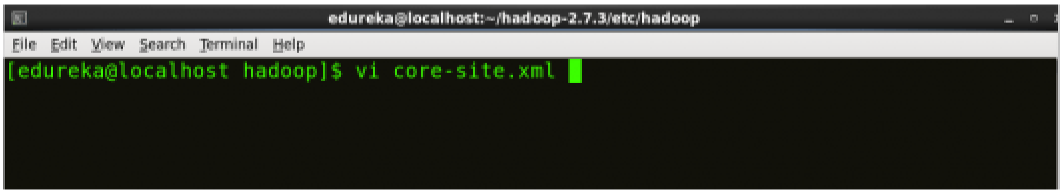
```
[edureka@localhost ~]$ cd hadoop-2.7.3/etc/hadoop/
[edureka@localhost hadoop]$ ls
capacity-scheduler.xml      httpfs-env.sh              mapred-env.sh
configuration.xsl           httpfs-log4j.properties    mapred-queues.xml.template
container-executor.cfg      httpfs-signature.secret    mapred-site.xml.template
core-site.xml               httpfs-site.xml            slaves
hadoop-env.cmd              kms-acls.xml               ssl-client.xml.example
hadoop-env.sh               kms-env.sh                 ssl-server.xml.example
hadoop-metrics2.properties  kms-log4j.properties       yarn-env.cmd
hadoop-metrics.properties   kms-site.xml               yarn-env.sh
hadoop-policy.xml           log4j.properties           yarn-site.xml
hdfs-site.xml               mapred-env.cmd
[edureka@localhost hadoop]$
```

Fig: Hadoop Installation – Hadoop Configuration Files

Step 7: **Open *core-site.xml* and edit the property mentioned below inside configuration tag:**

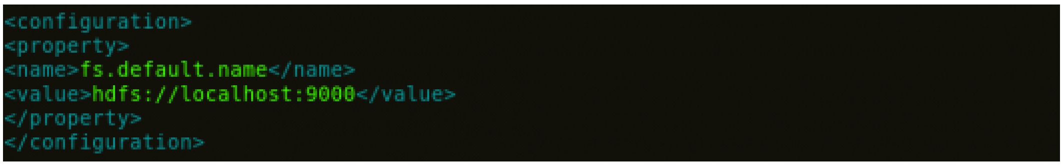
core-site.xml informs Hadoop daemon where NameNode runs in the cluster. It contains configuration settings of Hadoop core such as I/O settings that are common to HDFS & MapReduce.

***Command*:** vi core-site.xml


edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help

```
[edureka@localhost hadoop]$ vi core-site.xml
```

Hadoop Installation



```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

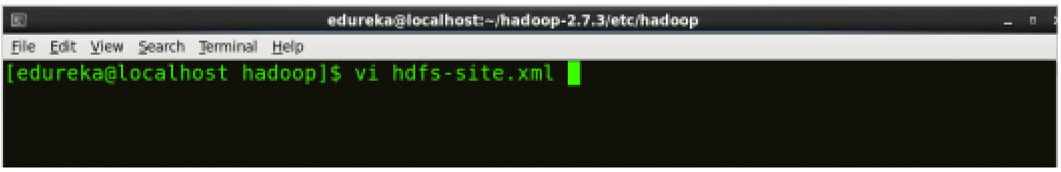
Fig: Hadoop Installation – Configuring core-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

Step 8: **Edit *hdfs-site.xml* and edit the property mentioned below inside configuration tag:**

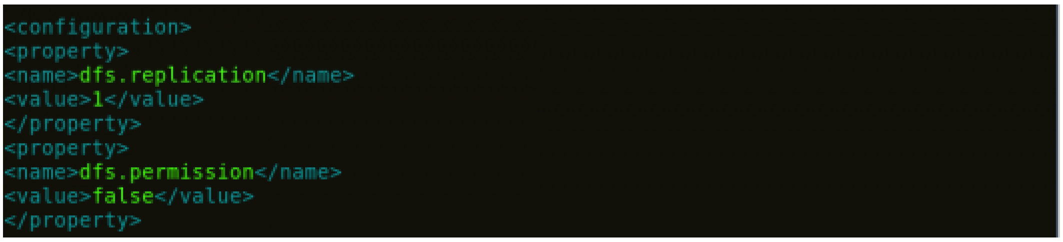
hdfs-site.xml contains configuration settings of HDFS daemons (i.e. NameNode, DataNode, Secondary NameNode). It also includes the replication factor and block size of HDFS.

***Command*:** vi hdfs-site.xml



edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help

```
[edureka@localhost hadoop]$ vi hdfs-site.xml
```



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

Fig: Hadoop Installation – Configuring hdfs-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.permission</name>
<value>false</value>
</property>
</configuration>
```

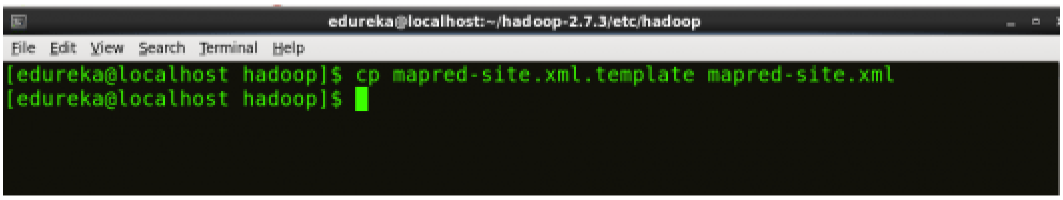
Step 9: **Edit the *mapred-site.xml* file and edit the property mentioned below inside configuration tag:**

mapred-site.xml contains configuration settings of MapReduce application like number of JVM that can run in parallel, the size of the mapper and the reducer process, CPU cores available for a process, etc.

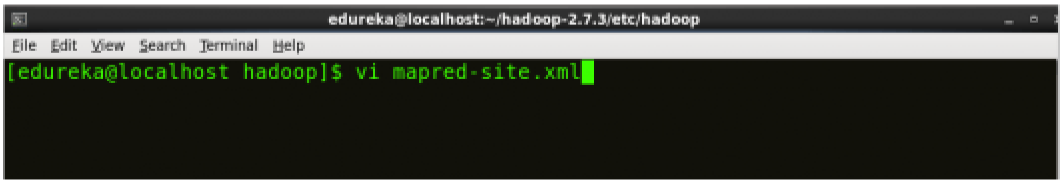
In some cases, mapred-site.xml file is not available. So, we have to create the mapred-site.xml file using mapred-site.xml template.

***Command*:** cp mapred-site.xml.template mapred-site.xml

***Command*:** vi mapred-site.xml.

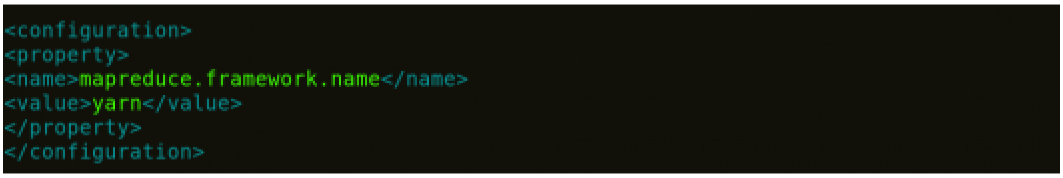

edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help

```
[edureka@localhost hadoop]$ cp mapred-site.xml.template mapred-site.xml
[edureka@localhost hadoop]$ 
```



edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help

```
[edureka@localhost hadoop]$ vi mapred-site.xml
```



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

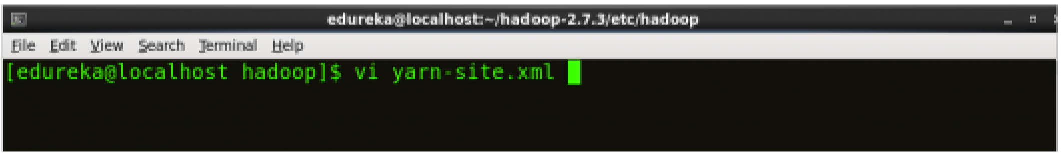
Fig: Hadoop Installation – Configuring mapred-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

Step 10: **Edit *yarn-site.xml* and edit the property mentioned below inside configuration tag:**

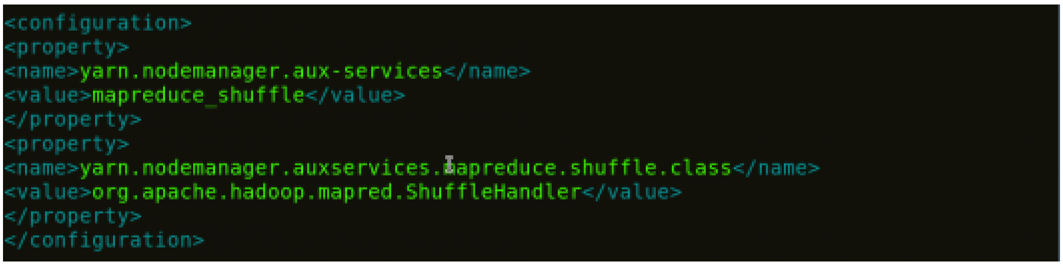
yarn-site.xml contains configuration settings of ResourceManager and NodeManager like application memory management size, the operation needed on program & algorithm, etc.

***Command*:** vi yarn-site.xml



edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help

```
[edureka@localhost hadoop]$ vi yarn-site.xml
```



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

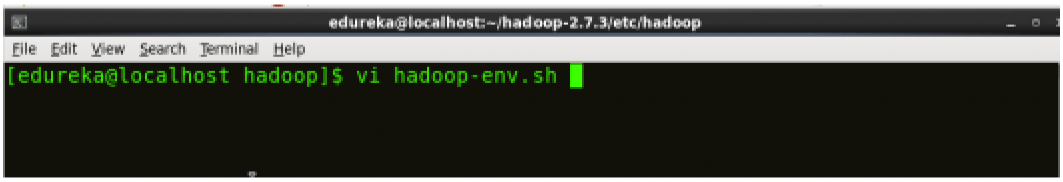
Fig: Hadoop Installation – Configuring yarn-site.xml

```
<?xml version="1.0">
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
</configuration>
```

Step 11: **Edit *hadoop-env.sh* and add the Java Path as mentioned below:**

hadoop-env.sh contains the environment variables that are used in the script to run Hadoop like Java home path, etc.

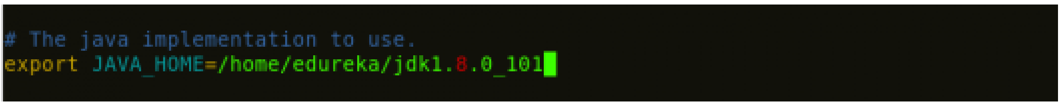
***Command*:** vi hadoop–env.sh



edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help

```
[edureka@localhost hadoop]$ vi hadoop-env.sh
```

Hadoop Installation



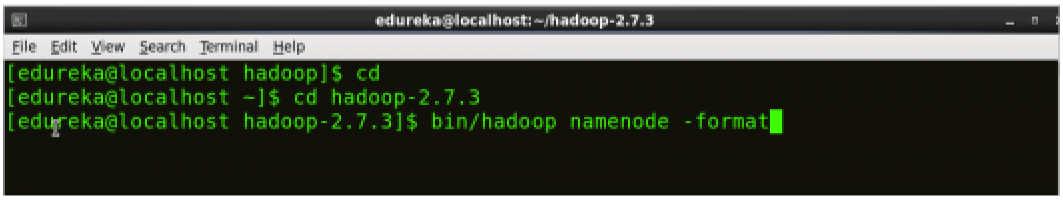
```
# The java implementation to use.
export JAVA_HOME=/home/edureka/jdk1.8.0_101
```

Fig: Hadoop Installation – Configuring hadoop-env.sh

Step 12: **Go to Hadoop home directory and format the NameNode.**
Command**:** cd

Command**:** cd hadoop-2.7.3

Command**:** bin/hadoop namenode -format


edureka@localhost:~/hadoop-2.7.3
File Edit View Search Terminal Help

```
[edureka@localhost hadoop]$ cd
[edureka@localhost ~]$ cd hadoop-2.7.3
[edureka@localhost hadoop-2.7.3]$ bin/hadoop namenode -format
```

Fig: Hadoop Installation – Formatting NameNode

This formats the HDFS via NameNode. This command is only executed for the first time. Formatting the file system means initializing the directory specified by the dfs.name.dir variable.

Never format, up and running Hadoop filesystem. You will lose all your data stored in the HDFS.

Step 13: **Once the NameNode is formatted, go to hadoop-2.7.3/sbin directory and start all the daemons.**
Command: cd hadoop-2.7.3/sbin

Either you can start all daemons with a single command or do it individually.

Command: ./start-all.sh

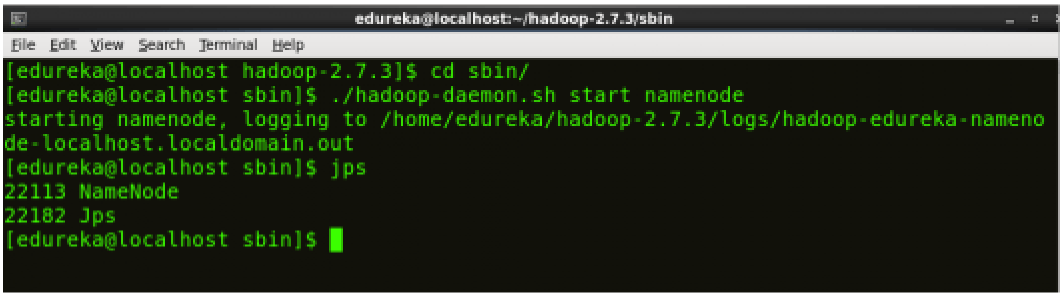
The above command is a combination of ***start-dfs.sh***, ***start-yarn.sh*** & ***mr-jobhistory-daemon.sh***

Or you can run all the services individually as below:

Start NameNode:
The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files stored in the HDFS and tracks all the file stored across the cluster.

Hadoop Installation

Command: ./hadoop-daemon.sh start namenode



edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help

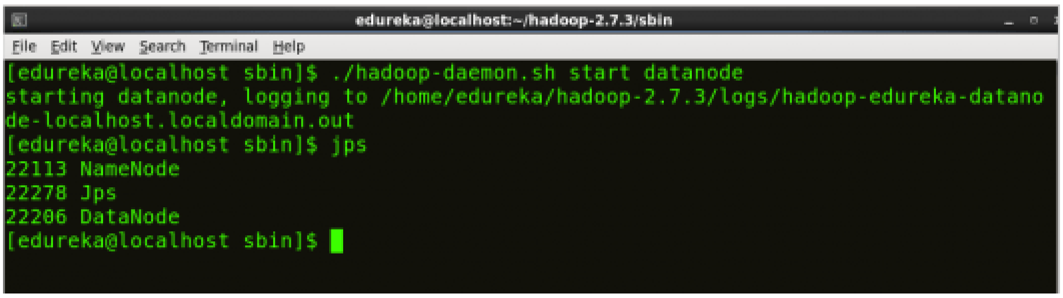
```
[edureka@localhost hadoop-2.7.3]$ cd sbin/
[edureka@localhost sbin]$ ./hadoop-daemon.sh start namenode
starting namenode, logging to /home/edureka/hadoop-2.7.3/logs/hadoop-edureka-nameno
de-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22182 Jps
[edureka@localhost sbin]$
```

Fig: Hadoop Installation – Starting NameNode

Start DataNode:

On startup, a DataNode connects to the Namenode and it responds to the requests from the Namenode for different operations.

Command: ./hadoop-daemon.sh start datanode



edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help

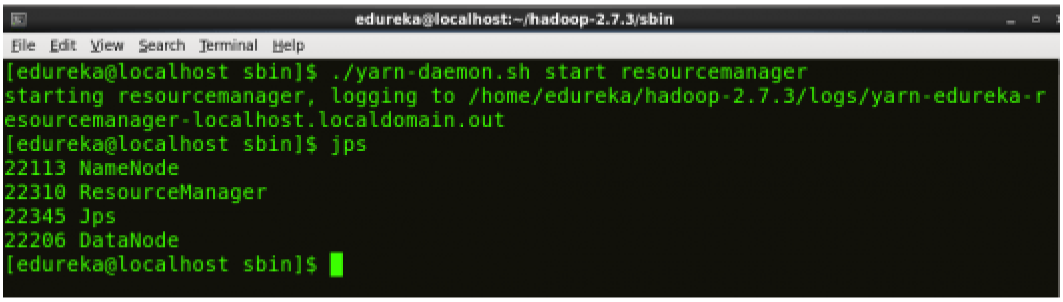
```
[edureka@localhost sbin]$ ./hadoop-daemon.sh start datanode
starting datanode, logging to /home/edureka/hadoop-2.7.3/logs/hadoop-edureka-datano
de-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22278 Jps
22206 DataNode
[edureka@localhost sbin]$
```

Fig: Hadoop Installation – Starting DataNode

Start ResourceManager:

ResourceManager is the master that arbitrates all the available cluster resources and thus helps in managing the distributed applications running on the YARN system. Its work is to manage each NodeManagers and the each application's ApplicationMaster.

Command: ./yarn-daemon.sh start resourcemanager



edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help

```
[edureka@localhost sbin]$ ./yarn-daemon.sh start resourcemanager
starting resourcemanager, logging to /home/edureka/hadoop-2.7.3/logs/yarn-edureka-r
esourcemanager-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22310 ResourceManager
22345 Jps
22206 DataNode
[edureka@localhost sbin]$
```

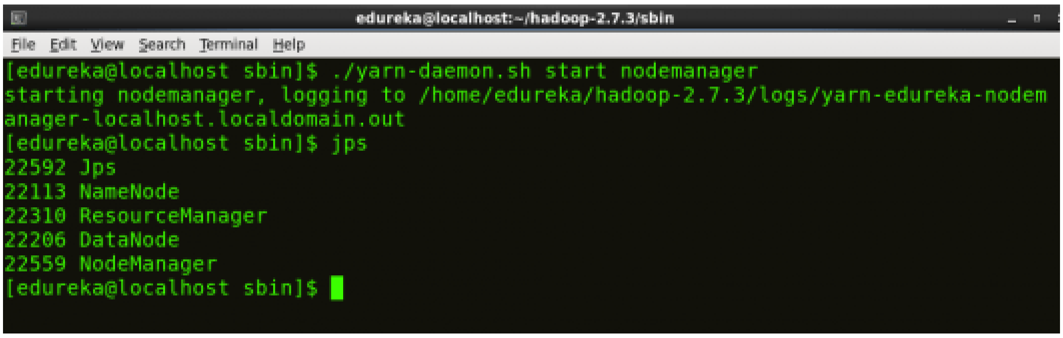
Fig: Hadoop Installation – Starting ResourceManager

Hadoop Installation

Start NodeManager:

The NodeManager in each machine framework is the agent which is responsible for managing containers, monitoring their resource usage and reporting the same to the ResourceManager.

Command: ./yarn-daemon.sh start nodemanager



edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help

```
[edureka@localhost sbin]$ ./yarn-daemon.sh start nodemanager
starting nodemanager, logging to /home/edureka/hadoop-2.7.3/logs/yarn-edureka-nodem
anager-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22592 Jps
22113 NameNode
22310 ResourceManager
22206 DataNode
22559 NodeManager
[edureka@localhost sbin]$ 
```

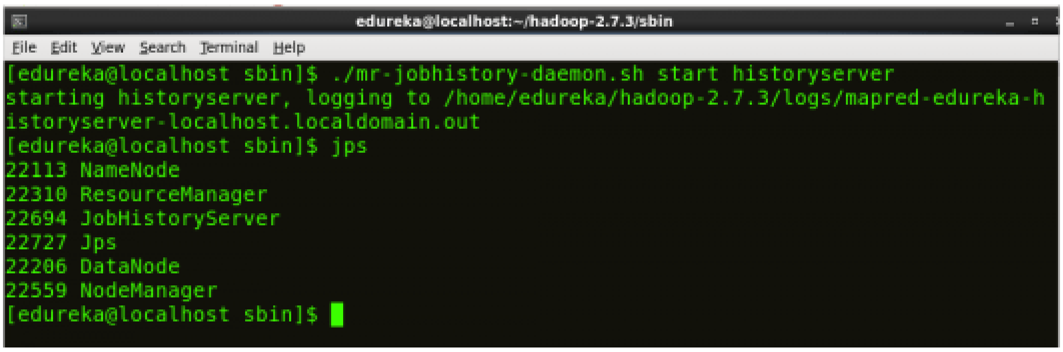
Start JobHistoryServer:

JobHistoryServer is responsible for servicing all job history related requests from client.

***Command*:** ./mr-jobhistory-daemon.sh start historyserver

Step 14: **To check that all the Hadoop services are up and running, run the below command.**

Command: jps



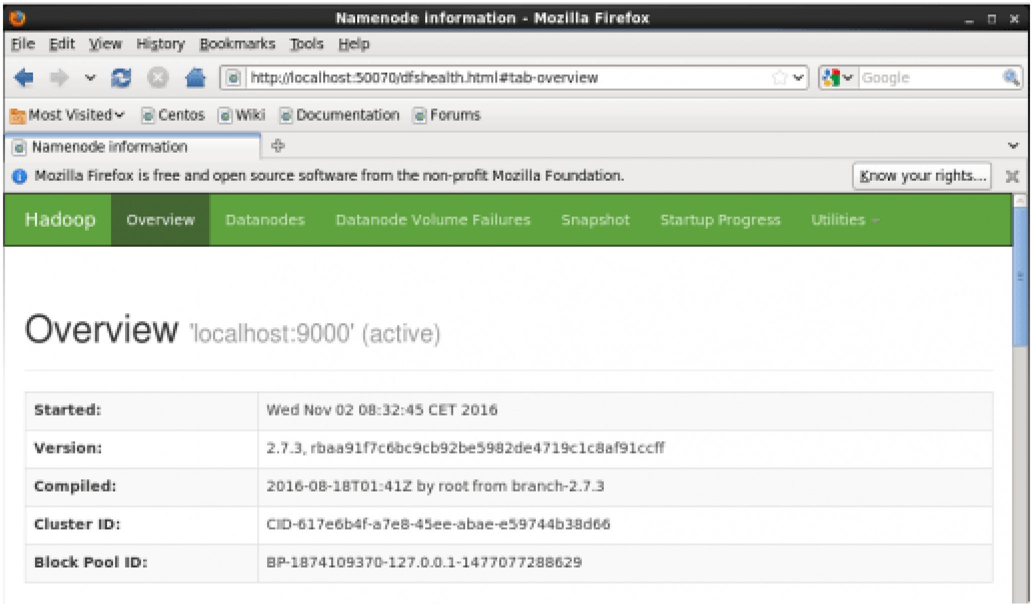
edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help

```
[edureka@localhost sbin]$ ./mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /home/edureka/hadoop-2.7.3/logs/mapred-edureka-h
istoryserver-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22310 ResourceManager
22694 JobHistoryServer
22727 Jps
22206 DataNode
22559 NodeManager
[edureka@localhost sbin]$ 
```

Fig: Hadoop Installation – Checking Daemons

Hadoop Installation

Step 15: **Now open the Mozilla browser and go to** localhost**:**50070/dfshealth.html **to check the NameNode interface.**



http://localhost:50070/dfshealth.html#tab-overview

Mozilla Firefox is free and open source software from the non-profit Mozilla Foundation.

Overview 'localhost:9000' (active)

Started:	Wed Nov 02 08:32:45 CET 2016
Version:	2.7.3, rbaa91f7c6bc9cb92be5982de4719c1c8af91ccff
Compiled:	2016-08-18T01:41Z by root from branch-2.7.3
Cluster ID:	CID-617e6b4f-a7e8-45ee-abae-e59744b38d66
Block Pool ID:	BP-1874109370-127.0.0.1-1477077288629

Fig: Hadoop Installation – Starting WebUI

Congratulations, you have successfully installed a single-node Hadoop cluster