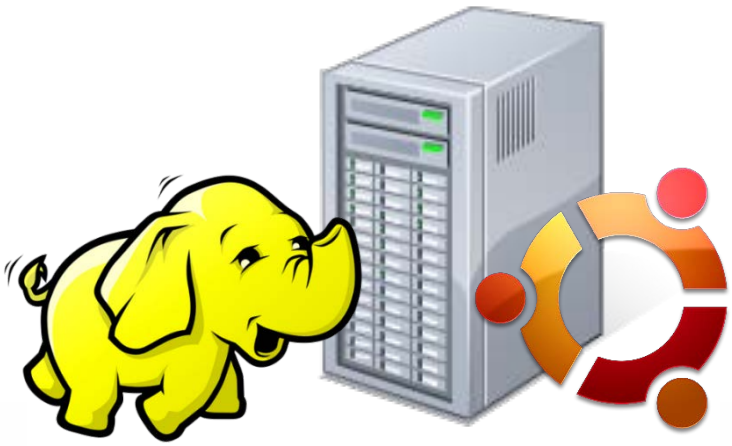
edureka!

15-Minute

Guide to Install

Apache Hadoop Cluster 2.0

With single Data Node Configuration on Ubuntu…

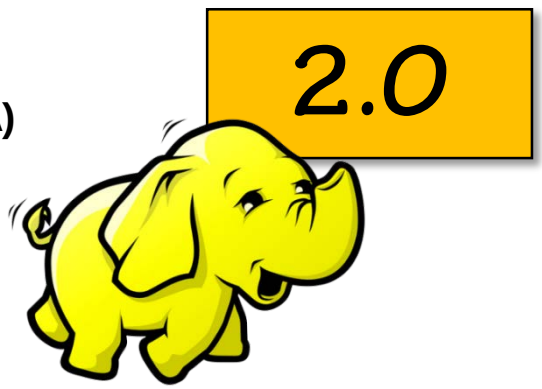


Let's know what is Apache Hadoop 2.0 all about?

Apache Hadoop 2.0 is now generally available! Thanks to the long awaited announcement done by **Apache Software Foundation (ASF).** The elephant has gone bigger with great features, all set to manage Big Data even better than before!

What's new in Hadoop 2.0?

- ✓ **YARN Framework (MapReduce 2.0)**
- ✓ **HDFS High Availability (NameNode HA)**
- ✓ **HDFS Federation**
- ✓ **Data Snapshot**
- ✓ **Support for Windows**
- ✓ **NFS Access**
- ✓ **Binary Compatibility**
- ✓ **Extensive Testing**



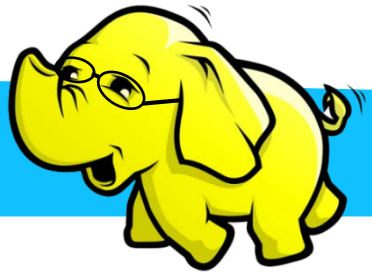
All these remarkable attributes and many more will increase Hadoop adoption tremendously in the industry to solve Big Data problems. Hadoop is now very much enterprise-ready with crucial security abilities!

According to Chris Douglas, Vice President of Apache Hadoop,

"With the release of stable Hadoop 2, the community celebrates not only an iteration of the software, but an inflection point in the project's development. We believe this platform is capable of supporting new applications and research in large-scale, commodity computing.

The Apache Software Foundation creates the conditions for innovative, community-driven technology like Hadoop to evolve. When that process converges, the result is inspiring."

An Intro to this Guide…



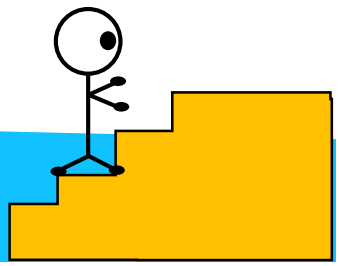
This setup and configuration document is a guide to setup a Single-Node Apache Hadoop 2.0 cluster on an **Ubuntu Virtual Machine (VM)** on your PC. If you are new to both **Ubuntu** and **Hadoop**, this guide comes handy to quickly setup a Single-Node Apache Hadoop 2.0 Cluster on Ubuntu and start your Big Data and Hadoop learning journey!

The Guide describes the whole process in two steps:

Step 1: Setting up the Ubuntu OS for Hadoop 2.0
This section describes step by step guide to download, configure an Ubuntu Virtual Machine image in VMPlayer, and provides steps to install pre-requisites for Hadoop Installation on Ubuntu.

Step 2: Installing Apache Hadoop 2.0 and Setting up the Single Node Cluster
This section explains primary Hadoop 2.0 configuration files, Single-Node cluster configuration and Hadoop daemons start and stop process in detail.

Note:
The configuration described here is intended for learning purposes only.



Follow these simple steps…

Step 1:

Setting up the Ubuntu Server

This section describes the steps to download and create an Ubuntu image on VMPlayer.

1.1 Creating an Ubuntu VMPlayer instance

The first step is to download an Ubuntu image and create an Ubuntu VMPlayer instance.

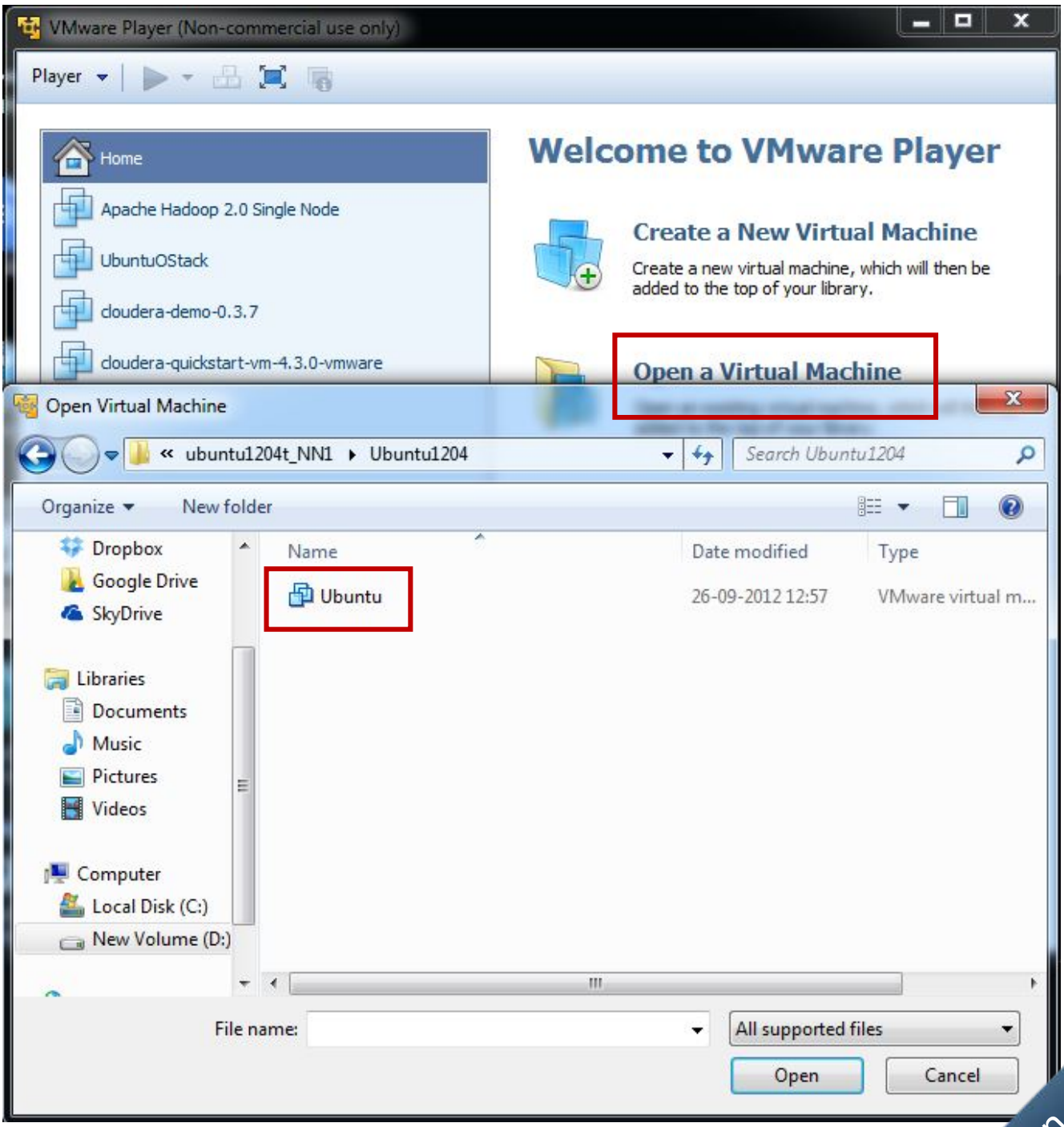
1.1.1 Download the VMware image

Access the following link and download the 12.0.4 Ubuntu image:
http://www.traffictool.net/vmware/ubuntu1204t.html

1.1.2 Open the image file

Extract the Ubuntu VM image and Open it in VMware Player.
Click open virtual machine and select path where you have extracted the image. Select the **'.vmx'** file and click **'ok'**.

The VM:



1.1.3 Play the Virtual Machine

You would see the below screen in VMware Player after the VM image creation completes. Double click on the link to run the machine. You will get the home screen of Ubuntu.

The user details for the Virtual instance are:

Username : user
Password : password

Open the Terminal to access the File System:

1.1.4 Update the OS packages and their dependencies

The first task is to run 'apt-get update' to download the package lists from the repositories and "update" them to get information on the newest versions of packages and their dependencies.

$sudo apt-get update

1.1.5 Install Java for Hadoop 2.2.0

Use apt-get to install JDK 6 on the server.
$sudo apt-get install openjdk-6-jdk

Check Java Version: **$java -version**

1.2 Download the Apache Hadoop 2.0 Binaries:

1.2.1 Download the Hadoop Package:

Download the binaries to your home directory. Use the default user 'user' for the installation.

In Live production instances a dedicated Hadoop user account for running Hadoop is used. Though, it's not mandatory to use a dedicated Hadoop user account but is recommended because this helps to separate the Hadoop installation from other software applications and user accounts running on the same machine (separating for security, permissions, backups, etc.).

$wget

http://apache.mirrors.lucidnetworks.net/hadoop/common/stable2/hadoop-2.2.0.tar.gz

Unzip the files and review the package content and configuration files.

$tar -xvf hadoop-2.2.0.tar.gz

Hadoop Package Content:

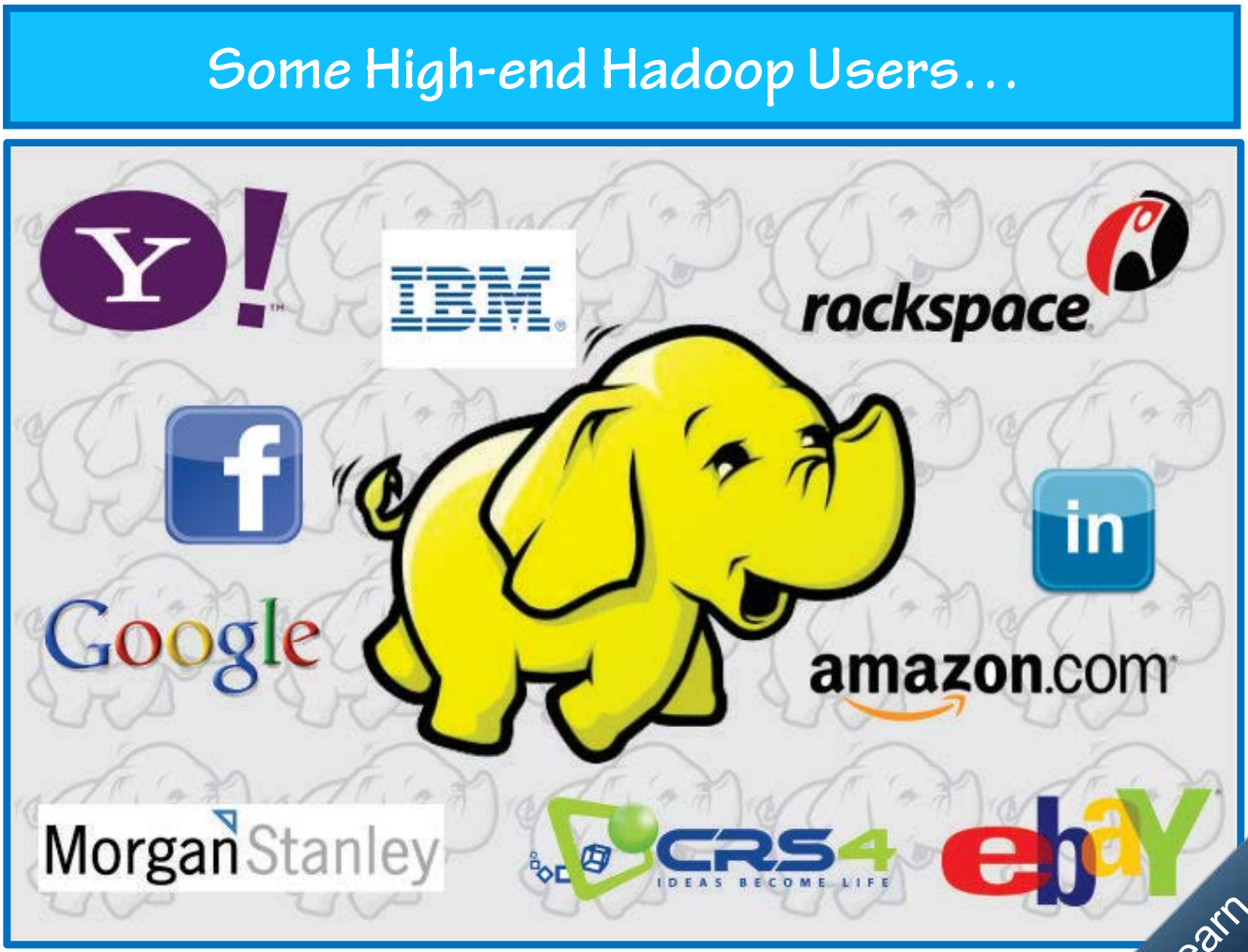
```
user@ubuntu:~$ ls
Desktop     examples.desktop     Music     temp
Documents   hadoop-2.2.0         Pictures  Templates
Downloads   hadoop-2.2.0.tar.gz  Public    Videos
user@ubuntu:~$
```

```
user@ubuntu:~$ cd hadoop-2.2.0/
user@ubuntu:~/hadoop-2.2.0$ ls
bin  include  libexec      NOTICE.txt  sbin
etc  lib      LICENSE.txt  README.txt  share
user@ubuntu:~/hadoop-2.2.0$ cd etc
user@ubuntu:~/hadoop-2.2.0/etc$ ls
hadoop
```

```
user@ubuntu:~/hadoop-2.2.0/etc$ cd hadoop/
user@ubuntu:~/hadoop-2.2.0/etc/hadoop$ ls
capacity-scheduler.xml
configuration.xsl
container-executor.cfg
core-site.xml
hadoop-env.cmd
hadoop-env.sh
hadoop-metrics2.properties
hadoop-metrics.properties
hadoop-policy.xml
hdfs-site.xml
httpfs-env.sh
httpfs-log4j.properties
httpfs-signature.secret
httpfs-site.xml
log4j.properties
mapred-env.cmd
mapred-env.sh
mapred-queues.xml.template
mapred-site.xml.template
slaves
ssl-client.xml.example
ssl-server.xml.example
yarn-env.cmd
yarn-env.sh
yarn-site.xml
user@ubuntu:~/hadoop-2.2.0/etc/hadoop$
```

Review the Hadoop Configuration Files:

After creating and configuring your virtual servers, the Ubuntu instance is now ready to start installation and configuration of **Apache Hadoop 2.0 Single Node Cluster.** This section describes the steps in detail to install Apache Hadoop 2.0 and configure a Single-Node Apache Hadoop cluster.



Step 2:

Configure the Apache Hadoop 2.0 Single Node Server

This section explains the steps to configure Single Node Apache Hadoop 2.0 Server on Ubuntu.

2.1 Update the Configuration Files

2.1.1 Update ".bashrc" file for user 'ubuntu'.
Move to 'user' $HOME directory and edit '.bashrc' file.

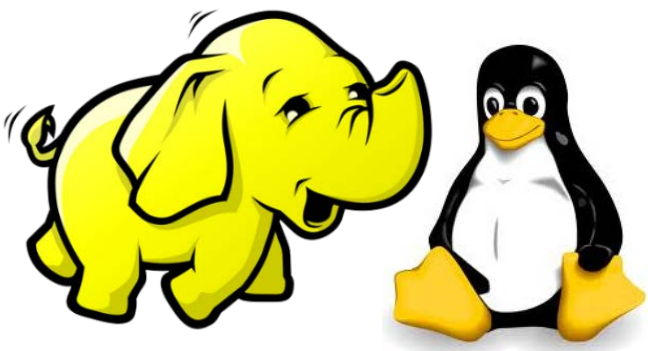
FILE '.BACHRC' LOCATION

```
user@ubuntu:~$ cd
user@ubuntu:~$ ls -al .b*
-rw------- 1 user user 1519 Nov 14 09:39 .bash_history
-rw-r--r-- 1 user user  220 Apr 28  2012 .bash_logout
-rw-r--r-- 1 user user 3486 Apr 28  2012 .bashrc
user@ubuntu:~$
```

Update the **'.bashrc'** file to add important Apache Hadoop environment variables for user.

Change directory to home - $ cd

Edit the file - $ vi .bashrc



Set Hadoop Environment Variables - Begin

```
# Set Hadoop-related environment variables
export HADOOP_HOME=$HOME/hadoop-2.2.0
export HADOOP_CONF_DIR=$HOME/hadoop-2.2.0/etc/hadoop
export HADOOP_MAPRED_HOME=$HOME/hadoop-2.2.0
export HADOOP_COMMON_HOME=$HOME/hadoop-2.2.0
export HADOOP_HDFS_HOME=$HOME/hadoop-2.2.0
export YARN_HOME=$HOME/hadoop-2.2.0
# Set JAVA_HOME (we will also configure JAVA_HOME for Hadoop
execution later on)
export JAVA_HOME=/usr/lib/jvm/java-6-openjdk-amd6

# Add Hadoop bin/ directory to PATH
export PATH=$PATH:$HOME/hadoop-2.2.0/bin
```

Set Hadoop Environment Variables - End

EDIT .BASHRC

```
#set Hadoop-related environment variable

 export HADOOP_HOME=$HOME/hadoop-2.2.0
 export HADOOP_MAPRED_HOME=$HOME/hadoop-2.2.0
 export HADOOP_COMMON_HOME=$HOME/hadoop-2.2.0
 export HADOOP_HDFS_HOME=$HOME/hadoop-2.2.0
 export YARN_HOME=$HOME/hadoop-2.2.0
 export HADOOP_CONF_DIR=$HOME/hadoop-2.2.0/etc/hadoop

# Set JAVA_HOME (we will also configure JAVA_HOME for Hadoop execution later on)

 export JAVA_HOME=/usr/lib/jvm/java-6-openjdk-amd6

# Add Hadoop bin/ directory to PATH

 export PATH=$PATH:$HOME/hadoop-2.2.0/bin
```

c) Source the .bashrc file to set the Hadoop environment variables without having to invoke a new shell:

```
$. ~/.bashrc
```

Execute all the steps of this section on all the remaining cluster servers.

2.2 Setup the Hadoop Cluster:

This section describes the detail steps needed for setting up the Hadoop Cluster and configuring the core Hadoop configuration files.

2.2.1 Configure JAVA_HOME

Configure JAVA_HOME in 'hadoop-env.sh'. This file specifies environment variables that affect the JDK used by Apache Hadoop 2.0 daemons started by the Hadoop start-up scripts.

```
$cd $HADOOP_CONF_DIR
$vi hadoop-env.sh
```

Update the JAVA_HOME to:

```
export JAVA_HOME=/usr/lib/jvm/java-6-openjdk-amd64
```

Java Home Set-up

```
# The java implementation to use.
#export JAVA_HOME=${JAVA_HOME}
export JAVA_HOME=/usr/lib/jvm/java-6-openjdk-i386
```

2.2.2 Create NameNode and DataNode directory

Create DataNode and NameNode directories to store HDFS data.

$ mkdir -p $HOME/hadoop2_data/hdfs/namenode
$ mkdir -p $HOME/hadoop2_data/hdfs/datanode

2.2.3 Configure the Default File system

The 'core-site.xml' file contains the configuration settings for Apache Hadoop Core such as I/O settings that are common to HDFS, YARN and MapReduce. Configure default files-system (Parameter: fs.default.name) used by clients in core-site.xml

$cd $HADOOP_CONF_DIR
$vi core-site.xml

Add the following lines in between the configuration tag:

Configuring the Default File System:

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

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

Here the **Hostname** and **Port** are the machine and port on which Name Node daemon runs and listens. It also informs the Name Node as to which IP and port it should bind. The commonly used port is **9000** and you can also specify IP address rather than hostname.

Note:
For the simplicity of understanding the cluster setup, we have updated only the necessary parameters to start a cluster. You can research more on Apache Hadoop 2.0 page and experiment the configuration for different features.

2.2.4 Configure the HDFS:

This file contains the cconfiguration settings for HDFS daemons; the Name Node and the data nodes.

Configure **hdfs-site.xml** and specify default block replication, and **NameNode** and **DataNode** directories for HDFS. The actual number of replications can be specified when the file is created. The default is used if replication is not specified in create time.

$cd $HADOOP_CONF_DIR
$vi hdfs-site.xml

Add the following lines in between the configuration tag:

Configuring the Default File System:

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

<configuration>
<property>
 <name>dfs.replication</name>
 <value>1</value>
</property>
<property>
 <name>dfs.namenode.name.dir</name>
 <value>file:/home/user/hadoop-2.2.0/hadoop2_data/hdfs/namenode</value>
</property>
<property>
 <name>dfs.datanode.data.dir</name>
 <value>file:/home/user/hadoop-2.2.0/hadoop2_data/hdfs/datanode</value>
</property>
</configuration>
```

2.2.5 Configure YARN framework:

This file contains the configuration settings for YARN; the NodeManager.

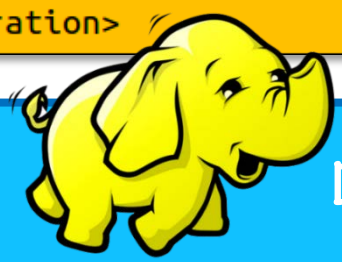
$cd $HADOOP_CONF_DIR
$vi yarn-site.xml

Add the following lines in between the configuration tag:

Configuring the Default Filesystem

```
-->
<configuration>

<!-- Site specific YARN configuration properties -->
<property>
 <name>yarn.nodemanager.aux-service</name>
 <value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-service.mapreduce.shuffle.class</name>
 <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
</configuration>
```



Do You Know?

- ✓ **HDFS is a File system, not a database management system (DBMS), as commonly perceived!**
- ✓ **Hadoop is an ecosystem consisting of multiple products, not a single product!**
- ✓ **Hadoop enables several kinds of analytics, apart from Web analytics!**

2.2.6 Configure MapReduce framework

This file contains the configuration settings for MapReduce.
So, Configure mapred-site.xml and specify framework details.

$cd $HADOOP_CONF_DIR

You need to copy the **mapred-site.xml** template.

$cp mapred-site.xml template mapred-site.xml
$vi mapred-site.xml

Add the following line in between the configuration tag:

Configuring the JobTracker Details:

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

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

2.2.7 Start the DFS services:

The first step in starting up your Hadoop installation is formatting the Hadoop file-system, which is implemented on top of the local file-systems of your cluster. This is required on the first time Hadoop installation.
Do not format a running Hadoop file-system, this will cause all your data to be erased.

To format the file-system, run the command:

$hadoop namenode –format

You are now all set to start the HDFS services i.e. Name Node, Resource Manager, Node Manager and Data Nodes on your Apache Hadoop Cluster!

Starting the Services:

```
user@ubuntu:~/hadoop-2.2.0/sbin$ ./hadoop-daemon.sh start datanode
starting datanode, logging to /home/user/hadoop-2.2.0/logs/hadoop-user-datanode-
ubuntu.out
user@ubuntu:~/hadoop-2.2.0/sbin$ jps
11750 Jps
11716 DataNode
user@ubuntu:~/hadoop-2.2.0/sbin$ ./hadoop-daemon.sh start namenode
starting namenode, logging to /home/user/hadoop-2.2.0/logs/hadoop-user-namenode-
ubuntu.out
user@ubuntu:~/hadoop-2.2.0/sbin$ jps
11716 DataNode
11828 Jps
11795 NameNode
user@ubuntu:~/hadoop-2.2.0/sbin$
```

Start the YARN Daemons i.e. **Resource Manager** and **Node Manager**. Cross check the service start-up using **JPS** (Java Process Monitoring Tool).

Starting the YARN Daemons:

```
user@ubuntu:~/hadoop-2.2.0/sbin$ ./yarn-daemon.sh start resourcemanager
starting resourcemanager, logging to /home/user/hadoop-2.2.0/logs/yarn-user-resc
urcemanager-ubuntu.out
user@ubuntu:~/hadoop-2.2.0/sbin$ jps
13228 DataNode
13402 Jps
13357 ResourceManager
12885 NameNode
user@ubuntu:~/hadoop-2.2.0/sbin$
```

```
user@ubuntu:~/hadoop-2.2.0/sbin$ ./yarn-daemon.sh start nodemanager
starting nodemanager, logging to /home/user/hadoop-2.2.0/logs/yarn-user-nodemana
ger-ubuntu.out
user@ubuntu:~/hadoop-2.2.0/sbin$ jps
13592 NodeManager
13228 DataNode
13357 ResourceManager
12885 NameNode
13622 Jps
user@ubuntu:~/hadoop-2.2.0/sbin$
```

Starting the History Server:

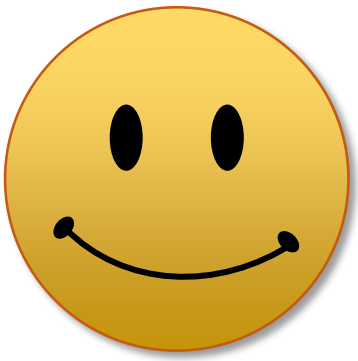
```
user@ubuntu:~/hadoop-2.2.0/sbin$ ./mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /home/user/hadoop-2.2.0/logs/mapred-user-hist
oryserver-ubuntu.out
user@ubuntu:~/hadoop-2.2.0/sbin$ jps
13592 NodeManager
13228 DataNode
13357 ResourceManager
12885 NameNode
13743 Jps
13711 JobHistoryServer
user@ubuntu:~/hadoop-2.2.0/sbin$
```

2.2.8 Finally, perform the Health Check!

a) **Check the NameNode status:**
http://localhost:50070/dfshealth.jsp

b) **JobHistory status:**
http://localhost:19888/jobhistory.jsp

And…You are DONE!



edureka!

Click Here to learn more about
Big Data & Hadoop…

Contact us at: learn@edureka.in