Install Hadoop 2 with YARN in Pseudo-Distributed Mode

13 Jun, 2016 in Hadoop Tutorials by DataFlair Team

1. Install Hadoop 2 with YARN Tutorial: Objective

In this tutorial to install Hadoop 2 with Yarn, we will learn how to setup and run **Apache Hadoop** 2 with yarn on a single-node on Ubuntu **Linux** OS. By Single Node Hadoop cluster we mean to say that all master, as well as slave daemons, will run on the same machine. After the completion to install Hadoop 2 with yarn procedure of single node Hadoop cluster, we will be easily able to perform operations such as **Hadoop Distributed File System (HDFS)** and **Map-Reduce**. Single Node **Hadoop cluster** is also termed as Hadoop in Pseudo-Distributed Mode.

To learn more big data technologies like Apache Spark, follow this comprehensive guide.

Install Hadoop 2 with Yarn

2. Setup and Install Hadoop 2 with Yarn

Follow the steps given below to install Hadoop 2 with Yarn on ubuntu os-

2.1. Required Platforms to Install Hadoop 2 with Yarn on Ubuntu

- **Operating system:** Ubuntu 14.04 or later, we can also use other Linux versions such as CentOS, Redhat, etc.
- **Hadoop:** Cloudera Distribution for Apache Hadoop CDH5.x (we can also use Apache Hadoop 2.x)

Note: If you are using Windows/Mac OS you can create a virtual machine and install Ubuntu using VMWare Player, alternatively you can create a virtual machine and install Ubuntu using Oracle Virtual Box.

2.2. Prerequisites to Install Hadoop 2 with Yarn

Necessary Software's to be installed before starting up to install Hadoop 2 with Yarn.

I. Java 8 Installation

a. Install Python Software Properties

In order to add the Java repositories, we need to download python-software-properties. Download and Installation of Python software properties can be easily executed by using the below command:

```
sudo apt-get install python-software-properties
```

NOTE: After pressing "Enter" key. It will ask for your password because we are using "sudo" command to provide root privileges for the installation. And in order to install or configure any software, we always need root privileges.

b. Add Repository

Now we will add a Repository. A repository is nothing but a collection of software for a Linux distribution on a server from where we can easily download and install software's using command "apt-get" in Linux. So, now we will manually add a java repository so that Ubuntu can easily install the Java. Command to add a Java repository:

```
sudo add-apt-repository ppa:webupd8team/java
```

Press "Enter".

c. Update the source list

The source list is basically a location where Ubuntu download and install the software and also updates the software. So whenever we install a new package/software or update any previously installed software then it is recommended that we should update the source list periodically so that changes made can easily get saved and we can easily use the software. So in order to update source list, we need to use the following command:

```
sudo apt-get update
```

When you run the above command Ubuntu will update its source list.

d. Install Java

Hadoop framework is written in Java. Hence it requires Java to be installed. So to download and install Java we need to use the below command:

```
sudo apt-get install oracle-java8-installer
```

When you will press enter it will start downloading and installing Java.
To check the whether the Java is installed successfully or not and to check the version of your Java type the below command terminal:

```
java -version
```

II. Configure SSH to Install Hadoop with Yarn

SSH means secured shell which is used for the remote login. Means we can login to a remote machine using SSH. Hence SSH setup is required to allow password-less login for the **Hadoop** user from machines in the cluster. Password-less SSH setup is required for remote script invocation. Automatically remotely master will start the demons on slaves.

a. Install Open SSH Server-Client

These SSH tools will be used for the remote login.

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

b. Generate Key Pairs

Now will generate public and private key pair, which is used for authentication.

```
ssh-keygen -t rsa -P ""
```

Note: It will ask "Enter the file in which to save the key (/home/hdadmin/.ssh/id_rsa):" You don't need to make any changes just press "Enter". Now the file will be available in".ssh" path. You can check the default path by "$ls .ssh/" command and there you will see that two files are created "id_rsa" which is a private key and "id_rsa.pub" which is a public key.

c. Configure passwordless SSH

We need to copy the contents of "id_rsa.pub" into the "authorized_keys" file. So this can be done by using the command:

```
cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

d. Check by SSH to localhost

```
ssh localhost
```

As previously we had configured passwordless SSH hence it will not ask for any password and thus we can easily get logged into localhost.

2.3. Getting Started with Hadoop Installation

Step by step procedure for installation of Hadoop-2.6.2 in pseudo distributed mode is given below.

I. Downloading Hadoop

We can download Hadoop from the below link.

```
http://www.eu.apache.org/dist/hadoop/common/hadoop-2.6.2/hadoop-2.6.2.tar.gz
```

After downloading Hadoop, we need to copy it to our home directory.

II. Untar Tar-ball

Now we need to extract all the contents of compressed Hadoop file. For this we will use Tar command:

```
tar xzf hadoop-2.6.2.tar.gz
```

Note: All the libraries, scripts, configuration files, etc. are available in HADOOP_HOME directory (the extracted directory is called as HADOOP_HOME. e.g. hadoop-2.6.0-cdh5.5.1)

III. Setup Configuration to Install Hadoop with Yarn

Making changes in Setup Configuration files for successful installation and running of Hadoop services:

a. Edit .bashrc

Now firstly we need to edit environment file ".bashrc" which is present in our home directory. We can identify and edit this file with the command: $ nano -/.bashrc". In this file we need to add the below contents at the end of this file:

```
1.  export JAVA_HOME=/usr/lib/jvm/java-8-oracle/
2.  export HADOOP_HOME=home/dataflair/hadoop-2.6.2
3.  export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
4.  export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib
5.  export HADOOP_PREFIX="/home/hdadmin/hadoop-2.6.2"
6.  export PATH=$PATH:$HADOOP_PREFIX/bin
7.  export PATH=$PATH:$HADOOP_PREFIX/sbin
8.  export HADOOP_MAPRED_HOME=${HADOOP_PREFIX}
9.  export HADOOP_COMMON_HOME=${HADOOP_PREFIX}
10. export HADOOP_HDFS_HOME=${HADOOP_PREFIX}
11. export YARN_HOME=${HADOOP_PREFIX}
```

Note: Make to enter the correct path. "/home/dataflair/hadoop-2.6.2" this is my home directory path. By "$pwd" command you can get your home directory path.

On adding the above contents in the .bashrc file we need to save it. In order to save it press "Ctrl+X".

Note: After above step restarts the terminal so that all the environment variables will come into effect

b. Edit hadoop-env.sh

Now we need to edit configuration "hadoop-env.sh" file which is present in configuration directory (HADOOP_HOME/etc/hadoop) and here we have to provide root path of Java installation. We have to provide Java installation path in "hadoop-env.sh" file under the heading "JAVA_HOME". This can be done by executing steps:

```
1.  dataflair@ubuntu:~$cd hadoop-2.6.2/
2.  dataflair@ubuntu:~/hadoop-2.6.2$ cd etc/hadoop
3.  dataflair@ubuntu:~/hadoop-2.6.2/etc/hadoop$ nano hadoop-env.sh
```

Here set JAVA_HOME as:

```
export JAVA_HOME=<path-to-the-root-of-your-Java-installation> (eg: /usr/lib/jvm/java-8-oracle/)
```

On adding the above contents in hadoop-env.sh file we need to save it. In order to save it press "Ctrl+X".
Note: "/usr/lib/jvm/java-8-oracle/" is default Java path. If you had made any changes in your java path then enter that path here.

c. Edit core-site.xml

Now we need to edit configuration "core-site.xml" file which is present in configuration directory (HADOOP_HOME/etc/hadoop) with the help of the following command:

```
dataflair@ubuntu:~/ hadoop-2.6.2/etc/hadoop$ nano core-site.xml
```

And here we have to add below entries between present at the end of this file:

```
<property>
<name>fs.defaultFS</name>
<value>hdfs://localhost:9000</value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>/home/dataflair/hdata</value>
</property>
```

Note: "/home/dataflair/hdata" is my location where I have read-write privileges. So kindly enter the location of your path where you have Read Write privileges.

On adding the above contents in core-site.xml file we need to save it. In order to save it press "Ctrl+X".

d. Edit hdfs-site.xml

Now we need to edit configuration "hdfs-site.xml" file which is present in configuration directory (HADOOP_HOME/etc/hadoop) with the help of the following command:

```
dataflair@ubuntu:~/ hadoop-2.6.2/etc/hadoop$ nano hdfs-site.xml
```

And here we have to add below entries between present at the end of this file:

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

On adding the above contents in hdfs-site.xml file we need to save it. In order to save it press "Ctrl+X".

e. Edit mapred-site.xml

Now we need to edit configuration ”mapred-site.xml” file which is present in configuration directory (HADOOP_HOME/etc/hadoop) with the help of the following command:

```
dataflair@ubuntu:~/ hadoop-2.6.2/etc/hadoop$ nano mapred-site.xml
```

After pressing “Enter” we will see that a new blank file named mapred-site.xml gets generate. This happens because there is no such file named mapred-site.xml is present in our home directory. Instead of that, there exists a template file available mapred-site.xml.template. So in order to edit mapred-site.xml file, firstly we need to create a copy of mapred-site.xml.template file.
Command to create a copy of mapred-site.xml.template file is:

```
dataflair@ubuntu:~/ hadoop-2.6.2/etc/hadoop$ cp mapred-site.xml.template mapred-site.xml
```

After creating a copy of mapred-site.xml.template file, we can now be able to edit mapred-site.xml file with the following command:

```
dataflair@ubuntu:~/ hadoop-2.6.2/etc/hadoop$ nano mapred-site.xml
```

And here we need to add below entries between present at the end of this file:

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

On adding the above contents in mapred-site.xml file we need to save it. In order to save it press “Ctrl+X”.

f. Edit yarn-site.xml

Now we need to edit configuration ”yarn-site.xml” file which is present in configuration directory (HADOOP_HOME/etc/hadoop) with the help of the following command:

```
dataflair@ubuntu:~/ hadoop-2.6.2/etc/hadoop$ nano yarn-site.xml
```

And here we need to add below entries between present at the end of this file:

```
<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>
```

Now we are completed with the installation procedure of Hadoop-2.6.2 in pseudo distributed mode. Now we will start Hadoop services by following steps.

For deep dive into Hadoop Yarn, resource manager follow this yarn tutorial.

IV. Start the Hadoop Cluster

a. Format the namenode

Before starting the Hadoop services we need to format namenode. In order to do so, we will use the below command:

```
dataflair@ubuntu:~$ hdfs namenode -format
```

NOTE: This activity should be done only once when you install Hadoop if you repeat it you lose all your data and different namespaceId will be assigned to namenode. If namenode and datanode have different id datanode daemon will not be started when to start the daemons.

b. Start HDFS Daemons:

```
dataflair@ubuntu:~$ start-dfs.sh
```

c. Start YARN Daemons:

```
dataflair@ubuntu:~$ start-yarn.sh
```

d. Check the status of services:

```
dataflair@ubuntu:~$ jps
```

NameNode
DataNode
ResourceManager
NodeManager

If you are getting above output after running jps command, it means Hadoop has been installed successfully. Now you can perform **HDFS** operations (Follow HDFS commands guide) and MapReduce operations.

If you like this tutorial on How to install Hadoop with Yran on Ubuntu or if you have any queries regarding the tutorial, do let us know it in the comment section below and we will get back to you.

See Also-

- Hadoop 2.x vs Hadoop 3.x
- Hadoop vs Spark vs Flink

6 thoughts on “Install Hadoop 2 with YARN in Pseudo-Distributed Mode”

Bhavya

February 18, 2017 at 6:59 am

hdfs namenode -format:- hdfs:command not found

Ashutosh

July 23, 2017 at 9:45 pm

To People getting hdfs command not found error:

The content of .bashrc has some typo mistake, if you look closely. There is a double quote in export HADOOP_OPTS which is not closed. Also path need not be enclosed in double quotes at all. Edit your .bashrc. Use the below version:

export JAVA_HOME=/usr/lib/jvm/java-8-oracle/
export HADOOP_HOME=home/dataflair/hadoop-2.6.2
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS=-Djava.library.path=$HADOOP_HOME/lib
export HADOOP_PREFIX=/home/hdadmin/hadoop-2.6.2
export PATH=$PATH:$HADOOP_PREFIX/bin
export PATH=$PATH:$HADOOP_PREFIX/sbin
export HADOOP_MAPRED_HOME=${HADOOP_PREFIX}
export HADOOP_COMMON_HOME=${HADOOP_PREFIX}
export HADOOP_HDFS_HOME=${HADOOP_PREFIX}
export YARN_HOME=${HADOOP_PREFIX}

Haresh

April 10, 2017 at 1:45 pm

If sudo apt-get not identified, run sudo apt-get update && sudo apt-get upgrade
Above command will get the apt-get libraries and also update to latest

FrankensteiN

April 21, 2017 at 12:10 pm

frankenstein@ubuntu:~$ hdfs namenode -format
No command 'hdfs' found, did you mean:
Command 'hdfls' from package 'hdf4-tools' (universe)
Command 'hfs' from package 'hfsutils-tcltk' (universe)
hdfs: command not found

gurwinder

July 29, 2017 at 7:57 am

tar xzf hadoop-2.6.2.tar.gz
not able to run this.
plz guide.
i nstalled VMPlayer. at this tar xzf hadoop-2.6.2.tar.gz step i stuck

Prasuna

November 4, 2017 at 4:50 am

Hi,

I have followed the same instructions. But while formatting namenode, I am getting the below error

Error: Could not find or load main class org.apache.hadoop.hdfs.server.namenode.NameNode

please help!!

Comments are closed.

Post navigation

← Apache Flink vs Apache Spark – A comparison guide

15 most Frequently used Linux Commands Tutorial →

Search

Search

Categories

- Data Mining Tutorials
- Data Science Interview Questions
- Data Science Tutorials
- Entrepreneurship
- Flume Interview Questions
- Flume Tutorials
- Hadoop Interview Questions
- Hadoop Quiz
- Hadoop Tutorials
- HBase Interview Questions
- HBase Tutorials
- HCatalog Interview Questions
- HCatalog Tutorials
- HDFS Tutorials
- Hive Interview Questions
- Hive Quiz
- Hive Tutorials
- Impala Interview Questions
- Impala Quiz
- Impala Tutorials
- Interview Questions
- IOT Interview Questions
- IOT Tutorials
- Java Interview Questions
- Java Tutorials
- Jobs
- Kafka Interview Questions
- Kafka Quiz
- Linux Tutorials
- Machine Learning Tutorials
- MapReduce Tutorials
- MongoDB Tutorials
- Pig Interview Questions
- Pig Quiz
- Pig Tutorials
- Power BI Interview Questions
- Power BI Tutorials
- Pyspark Interview Questions

- PySpark Tutorials
- Python Interview Questions
- Python Tutorials
- QlikView Tutorials
- Quiz
- R Interview Questions
- R Quiz
- R Tutorials
- Salesforce Interview Questions
- Salesforce Tutorials
- SAS – STAT Interview Questions
- SAS – STAT Tutorials
- SAS Interview Questions
- SAS Quiz
- SAS Tutorials
- Scala Interview Questions
- Scala Tutorials
- Spark Interview Questions
- Spark Quiz
- Spring Tutorials
- SQL Interview Questions
- SQL Tutorials
- Sqoop Interview Questions
- Sqoop Tutorials
- Storm Tutorials
- Tableau Interview Questions
- Tableau Tutorials
- TensorFlow Interview Questions
- Tensorflow Tutorials
- Training
- YARN Tutorials
- ZooKeeper Interview Questions
- Zookeeper Tutorials

Training in Cities: Bangalore | Chennai | Hyderabad | Delhi | NCR | Mumbai | Pune | Kolkata | Chicago | San Francisco | Los Angeles | New York | Boston | London