Install and Configure Apache Hadoop on Ubuntu 20.04

Last Updated: Tue, Aug 3, 2021 Server Apps Ubuntu

Using a Different System?

Introduction

Apache Hadoop is an open-source software framework used to store, manage and process large datasets for various big data computing applications running under clustered systems. It is Java-based and uses Hadoop Distributed File System (HDFS) to store its data and process data using MapReduce. In this article, you will learn how ro install and configure Apache Hadoop on Ubuntu 20.04.

Prerequisites

Deploy a fully updated Vultr Ubuntu 20.04 Server.

Create a non-root user with sudo access.

1. Install Java

Install the latest version of Java.

```
$ sudo apt install default-jdk default-jre -y
```

Verify the installed version of Java.

```
$ java -version
```

2. Create Hadoop User and Configure Password-less SSH

Add a new user `hadoop` .

```
$ sudo adduser hadoop
```

Add the `hadoop` user to the sudo group.

```
$ sudo usermod -aG sudo hadoop
```

Switch to the created user.

```
$ sudo su - hadoop
```

Install the OpenSSH server and client.

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

When you get a prompt, respond with:

```
keep the local version currently installed
```

Switch to the created user.

```
$ sudo su - hadoop
```

Generate public and private key pairs.

```
$ ssh-keygen -t rsa
```

Add the generated public key from `id_rsa.pub` to `authorized_keys` .

```
$ sudo cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_k
```

Change the permissions of the `authorized_keys` file.

```
$ sudo chmod 640 ~/.ssh/authorized_keys
```

Verify if the password-less SSH is functional.

```
$ ssh localhost
```

3. Install Apache Hadoop

Log in with `hadoop` user.

```
$ sudo su - hadoop
```

Download the latest stable version of Hadoop. To get the latest version, go to Apache Hadoop official download page.

```
$ wget https://downloads.apache.org/hadoop/common/h
```

Extract the downloaded file.

```
$ tar -xvzf hadoop-3.3.1.tar.gz
```

Move the extracted directory to the `/usr/local/` directory.

```
$ sudo mv hadoop-3.3.1 /usr/local/hadoop
```

Create directory to store system logs.

```
$ sudo mkdir /usr/local/hadoop/logs
```

Change the ownership of the hadoop directory.

```
$ sudo chown -R hadoop:hadoop /usr/local/hadoop
```

4. Configure Hadoop

Edit file `~/.bashrc` to configure the Hadoop environment variables.

```
$ sudo nano ~/.bashrc
```

Add the following lines to the file. Save and close the file.

```
export HADOOP_HOME=/usr/local/hadoop

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/li

export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bi

export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOM
```

Activate the environment variables.

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

5. Configure Java Environment Variables

Hadoop has a lot of components that enable it to perform its core functions. To configure these components such as YARN, HDFS, MapReduce, and Hadoop-related project settings, you need to define Java environment variables in `hadoop-env.sh` configuration file.

Find the Java path.

```
$ which javac
```

Find the OpenJDK directory.

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

Edit the `hadoop-env.sh` file.

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

Add the following lines to the file. Then, close and save the file.

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

```
export HADOOP_CLASSPATH+=" $HADOOP_HOME/lib/*.jar"
```

Browse to the hadoop `lib` directory.

```
$ cd /usr/local/hadoop/lib
```

Download the Javax activation file.

```
$ sudo wget https://jcenter.bintray.com/javax/activ
```

Verify the Hadoop version.

```
$ hadoop version
```

Edit the `core-site.xml` configuration file to specify the URL for your NameNode.

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

Add the following lines. Save and close the file.

```
<configuration>

   <property>

      <name>fs.default.name</name>

      <value>hdfs://0.0.0.0:9000</value>

      <description>The default file system URI</des

   </property>

</configuration>
```

Create a directory for storing node metadata and change the ownership to `hadoop` .

```
$ sudo mkdir -p /home/hadoop/hdfs/{namenode,datanoc



$ sudo chown -R hadoop:hadoop /home/hadoop/hdfs
```

Edit `hdfs-site.xml` configuration file to define the location for storing node metadata, fs-image file.

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

Add the following lines. Close and save the file.

```
<configuration>

   <property>

      <name>dfs.replication</name>
```

```
        <value>1</value>
    </property>

    <property>
        <name>dfs.name.dir</name>
        <value>file:///home/hadoop/hdfs/namenode</val
    </property>

    <property>
        <name>dfs.data.dir</name>
        <value>file:///home/hadoop/hdfs/datanode</val
    </property>
</configuration>
```

Edit `mapred-site.xml` configuration file to define MapReduce values.

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

Add the following lines. Save and close the file.

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

Edit the `yarn-site.xml` configuration file and define YARN-related settings.

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

Add the following lines. Save and close the file.

```
<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
```

```
    </property>

</configuration>
```

Log in with `hadoop` user.

```
$ sudo su - hadoop
```

Validate the Hadoop configuration and format the HDFS NameNode.

```
$ hdfs namenode -format
```

6. Start the Apache Hadoop Cluster

Start the NameNode and DataNode.

```
$ start-dfs.sh
```

Start the YARN resource and node managers.

```
$ start-yarn.sh
```

Verify all the running components.

```
$ jps
```

7. Access Apache Hadoop Web Interface

You can access the Hadoop NameNode on your browser via `http://server-IP:9870` . For example:

```
http://192.0.2.11:9870
```

Conclusion

You have successfully installed Apache Hadoop on your server. You can now access the dashboard and configure your preferences.

More Information

For more information on Apache Hadoop, please see the official documentation.

Request an article

Products

Features

Use Cases

Marketplace

Resources

Company

Over 45,000,000 Cloud Servers Launched