Fully Distributed Hadoop Cluster Setup*

Montasser AKERMI

Last Update: February, 2025

*The course material is hosted at `https://akermi.org/c1/`.

Learning Objectives

In addition to the default Hadoop Local/Standalone mode, there are two other ways to run Hadoop: pseudo-distributed and fully distributed modes. This practical session goes through the steps of setting up a three-node Hadoop cluster on the VirtualBox hypervisor: one master node and two worker nodes.

Pre-Lab Questions

- Why use hypervisors and not containers?
- Why use Bridged and not NAT?
- Why use Java when setting up Hadoop?
- Master and workers, what is a master-slave architecture?
- What are the main components of Hadoop?

Frameworks, Tools, OS, and Versions

- Apache Hadoop, version 3.4.0
- Java, version 8
- Lubuntu, version 24.04
- VirtualBox, version 7.1.4

Glossary

- Apache Hadoop
- Hypervisor and virtual machines
- Linux distribution

Checklist

1 Installing Linux on VirtualBox

A hypervisor is required to install Linux virtual machines. Oracle VirtualBox and VMware are leaders in providing virtualization solutions in the modern IT industry.

Prerequisites

- Download Oracle VirtualBox from `https://www.virtualbox.org/wiki/Downloads`.
- Download Lubuntu .iso file from `https://lubuntu.me/downloads`.

Once finished, run the executable file and continue with the default configurations.

1.1 Creating a New Linux Virtual Machine

1. Now, open VirtualBox and create a new virtual machine. Choose a name for your virtual machine, for example, *Master*. Select Linux as the type. Ubuntu (64-bit) is chosen for this version.
2. The next dialog should bring the Hard disk settings. Accept the default Create a virtual machine hard disk option and click on Create. Continue to accept the VDI option and click Next. Choose Dynamically allocated and click Next. For the hard drive size, the default 20 GB is sufficient for this Hadoop tutorial.
3. Change the network adapter mode to Bridged.
4. When you first start the virtual machine, VirtualBox will ask you to select the start-up disk. We selected the previously downloaded Lubuntu. iso file.
5. Choose the preferred language and press Enter. Once Lubuntu has started up, double-click on the Install Lubuntu 24.04 icon. You should be inside the wizard installation.
6. Again, choose your preferred language.
7. Make sure the default region and zone are correct.
8. Choose your keyboard layout.
9. Make sure the Erase disk option is selected.
   Tip: If you prefer to resize the partitions yourself, then choose Manual partitioning.
   Tip: To encrypt the operation system, you can check the Encrypt system option.
10. Enter your name, username, the name of the computer, and choose a password.
11. Now sit back and wait for the installation until it is finished.
12. After the installation is complete, check the Restart now option and click on Done.
13. When you see a screen with the Lubuntu logo saying Please remove installation medium, then press ENTER, just press ENTER.
14. Enter your password and login to the system. Now, Lubuntu is ready to be used.

2 Setting Up Hadoop in Fully Distributed Mode

In this part of the tutorial, we went through all steps to install Hadoop.

Prerequisites

1. Update and upgrade all the installed packages by running the following commands:
   ```
   sudo apt update
   sudo apt -y upgrade
   ```
2. Install networking utilities for Linux
   sudo apt install -y net-tools
3. Install secure shell protocol
   sudo apt install -y openssh-server

2.1 Installing JVM

Hadoop was developed using Java language; therefore, a java virtual machine must be installed on the system. It is recommended to install Java 8, as Hadoop 2.7.x and Upper support this version. There are two ways to do this: add a repository and install it directly with the terminal. Alternatively, download it and manually install it. Since we want a very specific version of Java, we will go with the second option.

1. Go to home directory and download java development kit
   `wget https://github.com/frekele/oracle-java/releases/download/8u92-b14/jdk-8u92-linux-x64.tar.gz`
2. Once finished, extract the tar archive
   `tar -xzvf jdk-8u92-linux-x64.tar.gz`
3. Edit ∼/.bashrc to include the Java home variable and the Java binary folder. You may use nano or vi. The following lines have been added.

   ```
   export JAVA_HOME=/home/huser/jdk1.8.0_92
   export PATH=$JAVA_HOME/bin:$PATH
   ```

4. Now, all we have to do is to refresh the .bashrc file:
   `source ∼/.bashrc`
5. Make sure that java was properly installed by running:
   `java -version`

2.2 Installing Hadoop

The latest stable version is recommended when installing a package.

1. Navigate to home directory and download Hadoop:
   `wget https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz`
2. Untar the Hadoop archive to extract all the files inside and place them inside a folder:
   `tar -xzvf hadoop-3.4.0.tar.gz`
3. Add Hadoop binaries to the PATH and set Hadoop environment variables by adding the following lines to the end of .bashrc file:

   ```
   export HADOOP_HOME=$HOME/hadoop-3.4.0
   export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
   export HADOOP_MAPRED_HOME=$HADOOP_HOME
   export HADOOP_COMMON_HOME=$HADOOP_HOME
   export HADOOP_HDFS_HOME=$HADOOP_HOME
   export YARN_HOME=$HADOOP_HOME
   export PATH=$PATH:$HADOOP_HOME/bin
   export PATH=$PATH:$HADOOP_HOME/sbin
   ```

4. Refresh .bashrc file
   `source ∼/.bashrc`
5. Check that Hadoop binary files are added to bash environment:
   `hadoop version`

2.3 Editing Hadoop Configuration Files

The minimum properties were added to the Hadoop configuration files. There exists a great range of properties to customize Hadoop-related frameworks.

1. Navigate to the Hadoop configuration directory and type ls to see all config files:
   `cd ~/hadoop-3.4.0/etc/hadoop`

2. Let's start with core-site.xml. This file contains information on common settings, such as the port number used for Hadoop instance, memory allocated for the file system, memory limit for storing data, and the size of read/write buffers. The content is replaced with the following configuration:

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

   Here, the property fs.default.name is the name of the default file system.

3. The file hdfs-site.xml contains information such as the name node path, secondary name node, and data node path of your local file systems, where you want to store the distributed data. Replace its content with the following configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/home/huser/hdata/hdfs/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/home/huser/hdata/hdfs/datanode</value>
  </property>
  <property>
    <name>dfs.namenode.checkpoint.dir</name>
    <value>file:/home/huser/hdata/hdfs/namesecondary</value>
  </property>
</configuration>
```

4. mapred-site.xml is used to specify which MapReduce framework we are using. Change its content with the following:

```
<?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>
  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>Master:10020</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.webapp.address</name>
    <value>Master:19888</value>
  </property>
  <property>
    <name>yarn.app.mapreduce.am.staging-dir</name>
    <value>/user</value>
  </property>
  <property>
    <name>mapreduce.jobtracker.address</name>
    <value>Master:9001</value>
  </property>
  <property>
    <name>mapred.local.dir</name>
    <value>${hadoop.tmp.dir}/mapred/local</value>
  </property>
  <property>
    <name>mapred.map.tasks</name>
    <value>20</value>
  </property>
  <property>
    <name>mapred.reduce.tasks</name>
    <value>2</value>
  </property>
</configuration>
```

5. yarn-site.xml configures settings for YARN daemons: the resource manager, the web app proxy server, and the node managers..., etc. The content was changed as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>Master</value>
  </property>
  <property>
    <name>yarn.nodemanager.local-dirs</name>
    <value>file:/home/huser/hdata/yarn/local</value>
  </property>
  <property>
    <name>yarn.nodemanager.log-dirs</name>
    <value>file:/home/huser/hdata/yarn/log</value>
```

```
  </property>
  <property>
    <name>yarn.log.server.url</name>
  </property>
  <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.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage</name
    <value>100</value>
  </property>
</configuration>
```

You are encouraged to read about all the available configurations on the Hadoop official website.

2.4 Cloning the Virtual Machine

The master–slave architecture was installed, including one master and two workers (slaves). The *Master* machine maintains the namespace of the distributed file system. Thus, it hosts four daemons:

- NameNode
- SecondaryNameNode
- ResourceManager
- JobHistoryServer

Worker1 and *Worker2* store the actual data and provide processing power to run the jobs. Thus, each will host two daemons:

- DataNode
- NodeManager

Usually, we need to repeat all of the above steps for every single machine. However, VirtualBox offers a practical feature called cloning. We will create two exact copies of the virtual machine that we created. The first is for Worker1 and the second is for Worker2.

1. Right-click on the Master virtual machine and click Clone. A new dialog will appear.
2. Write the name of the new machine, e.g., Worker1 and change the path if necessary.
3. Select Generate New MAC Addresses For All Network Adapters for the MAC Address Policy
4. Uncheck Keep Disk Names and Keep Hardware UUIDs and click Next.
5. Check Full clone and click Clone.

Wait until cloning is finished and repeat the process for Worker2. Now, we have three nodes for the Hadoop cluster.

2.5 Editing Hostname and Hosts Files on Each Machine

The names of the nodes can be set by changing the hostname file located at `/etc/hostname`. For each virtual machine to communicate with each other by name, we need to edit the hosts file and add the private IP addresses of the three virtual machines.

1. Get the IP address of each node:
   `ifconfig`
2. Edit the hosts file:
   `sudo nano /etc/hosts`
3. Add the IP addresses of the three virtual machines:

```
10.15.15.11 Master
10.15.15.12 Worker1
10.15.15.13 Worker2
```

Next, we set up an SSH and generate an RSA key pair.

2.6 SSH setup and key generation

The Master will use an Secure Shell (SSH) protocol to connect Worker1 and Worker2 with key-pair authentication. This will allow the master node to actively manage the cluster.

1. Check if SSH server is running:
   `service ssh status`
2. Generate a public/private RSA (Rivest Shamir Adleman) key-pair for SSH :
   `ssh-keygen −t rsa −P ""`
3. Now that the RSA key pair is generated, the key file is copied into the authorized key store:
   `cat ∼/.ssh/id_rsa.pub >> ∼/.ssh/authorized_keys`
4. Now we need to copy the public RSA key to Worker1 and Worker2:
   `ssh-copy-id -i ∼/.ssh/id_rsa.pub huser@Worker1`
   `ssh-copy-id -i ∼/.ssh/id_rsa.pub huser@Worker2`
5. Now, the Master can login to Workers, also to localhost, without the need to enter any password:
   `ssh Worker1`

This step should only be performed on the master.

2.7 Editing Workers file

List the hostnames of the datanodes in ∼/hadoop-3.4.0/etc/hadoop/workers file:

```
Worker1
Worker2
```

Copy the file to other nodes:

```
scp ∼/hadoop-3.4.0/etc/hadoop/workers huser@Worker1:∼/hadoop-3.4.0/etc/hadoop/
scp ∼/hadoop-3.4.0/etc/hadoop/workers huser@Worker2:∼/hadoop-3.4.0/etc/hadoop/
```

2.8 Formatting the new distributed filesystem as HDFS

This step should be done only the first time you startup HDFS:
`hdfs namenode -format`

Now everything was set up correctly, and we are ready to fire up the Hadoop daemons.

2.9 Operating the Hadoop Cluster

Once all necessary configurations are completed, managing the cluster should be easy and error-free.

2.9.1 Hadoop Cluster Startup

You need to start both the HDFS cluster and the YARN cluster.

- Start HDFS Namenode daemon on the Master virtual machine:
  `hdfs --daemon start namenode`
- Start HDFS DataNode daemons on the Datanode nodes:
  `hdfs --daemon start datanode`
- Start YARN resourcemanager daemon on Master node:
  `yarn --daemon start resourcemanager`
- Start YARN nodemanager daemons on Datanode nodes:
  `yarn --daemon start nodemanager`
- Start MapReduce JobHistory Server:
  `mapred --daemon start historyserver`

2.9.2 Hadoop Cluster Shutdown

Both the HDFS and YARN clusters can be stopped using the following commands:

- Stop HDFS Namenode daemons:
  `hdfs --daemon stop namenode`
- Stop HDFS Datanode daemons:
  `hdfs --daemon stop datanode`
- Stop YARN ResourceManager daemons:
  `yarn --daemon stop resourcemanager`
- Stop YARN NodeManager daemons:
  `yarn --daemon stop nodemanager`
- Stop MapReduce JobHistory Server:
  `mapred --daemon stop historyserver`

2.9.3 Accessing Hadoop Web User Interfaces

Once the cluster is up and running, Hadoop web interfaces can be accessed on

- Namenode: `http://10.15.15.11:9870/`
- ResourceManager: `http://10.15.15.11:8088/`
- MapReduce JobHistoryServer: `http://10.15.15.11:19888/`

Post-Lab Questions

- Why format HDFS filesystem only once during the initial setup?
- What is SPOF?
- How does Hadoop ensure fault tolerance?
- How to deal with passwordless SSH in a production environment?
- What to do when editing configuration files in the future?