Lab 1 — Deploying Classic Hadoop (HDFS + YARN)

Course theme: Evolution of Big Data Platforms
Lab objective: Understand *why* Hadoop exists by deploying and operating it manually.

This lab introduces **Apache Hadoop** as a *foundational big-data system*, focusing on **storage (HDFS)** and **resource management (YARN)**.

---

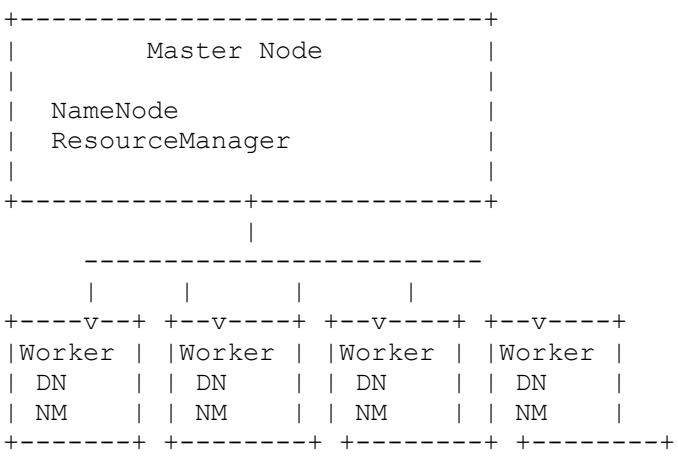
1. Learning objectives (for students)

After completing this lab, you will be able to:

- Explain the roles of **NameNode, DataNode, ResourceManager, and NodeManager**
- Deploy a distributed HDFS cluster across multiple machines
- Upload and manage datasets in HDFS
- Observe fault tolerance and replication behavior
- Explain why MapReduce jobs have high latency and operational overhead

---

2. Target architecture



---

3. Prerequisites checklist (before starting)

Students must verify:

- 5 Linux VMs running
- All VMs reachable via hostname
- Private network connectivity
- At least 2 GB RAM per VM
- Open ports: 9870, 8088, 9000

---

4. Step-by-step deployment tutorial

4.1 Hostnames and name resolution (all nodes)

Set hostname:

```
sudo hostnamectl set-hostname hadoop-master     # master
sudo hostnamectl set-hostname hadoop-worker1    # workers
```

Edit `/etc/hosts` on **all nodes (Important: use the real IP addresses of your nodes)**:

```
10.0.0.1  hadoop-master
10.0.0.2  hadoop-worker1
10.0.0.3  hadoop-worker2
10.0.0.4  hadoop-worker3
10.0.0.5  hadoop-worker4
```

✅ **Checkpoint:**

```
ping hadoop-worker3
```

4.2 Java installation (all nodes)

```
sudo apt update
sudo apt install -y openjdk-11-jdk
```

Verify:

```
java -version
```

4.3 Hadoop user (recommended practice)

```
sudo adduser hadoop
sudo usermod -aG sudo hadoop
su - hadoop
```

All remaining steps are done as user **hadoop**.

4.4 Passwordless SSH (critical)

On **master**:

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

Copy key to all nodes:

```
ssh-copy-id hadoop@hadoop-master
ssh-copy-id hadoop@hadoop-worker1
ssh-copy-id hadoop@hadoop-worker2
ssh-copy-id hadoop@hadoop-worker3
ssh-copy-id hadoop@hadoop-worker4
```

If the above doesn't work, then copy the key manually:

On the **master node**:

```
cat ~/.ssh/id_rsa.pub
```

Copy the output of this somewhere.

One **all nodes**:

```
mkdir -p /home/hadoop/.ssh
nano /home/hadoop/.ssh/authorized_keys
```

Paste the key **on a single line**.

Fix permissions **exactly as below**:

```
chmod 700 /home/hadoop/.ssh
chmod 600 /home/hadoop/.ssh/authorized_keys
chown -R hadoop:hadoop /home/hadoop/.ssh
```

✅ **Checkpoint:**

```
ssh hadoop-worker2
```

4.5 Hadoop installation (all nodes)

```
wget https://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-
3.3.6.tar.gz
tar -xzf hadoop-3.3.6.tar.gz
mv hadoop-3.3.6 ~/hadoop
```

4.6 Environment variables (all nodes)

Append to `~/.bashrc`:

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export HADOOP_HOME=$HOME/hadoop
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
source ~/.bashrc
```

---

5. Hadoop configuration (master)

The following files are all in the folder hadoop/etc/hadoop

5.1 core-site.xml

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://hadoop-master:9000</value>
  </property>
</configuration>
```

5.2 hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/home/hadoop/hdfs/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/home/hadoop/hdfs/datanode</value>
  </property>
</configuration>
```

Create directories (all nodes):

```
mkdir -p ~/hdfs/namenode ~/hdfs/datanode
```

5.3 yarn-site.xml

```
<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>hadoop-master</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

5.4 mapred-site.xml

```
cp mapred-site.xml.template mapred-site.xml
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

5.5 Workers file (hadoop/etc/hadoop/workers)

```
hadoop-worker1
hadoop-worker2
hadoop-worker3
hadoop-worker4
```

5.6 Set the Java path

You must set `JAVA_HOME` **inside Hadoop's own environment file**.

Step 1: Open **`hadoop-env.sh`**

Step 2: Set `JAVA_HOME` explicitly - Find this line:

export JAVA_HOME=

Replace it with (example for Ubuntu OpenJDK 11):

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

5.7 Distribute configuration

```
scp -r ~/hadoop hadoop@hadoop-worker1:~
scp -r ~/hadoop hadoop@hadoop-worker2:~
scp -r ~/hadoop hadoop@hadoop-worker3:~
scp -r ~/hadoop hadoop@hadoop-worker4:~
```

---

6. Initialize and start Hadoop

6.1 Format HDFS (Only ONCE on all nodes)

```
hdfs namenode -format
```

⚠ **Never repeat this step.**

6.2 Start services (only on the Master)

```
start-dfs.sh
start-yarn.sh
```

Verify:

```
jps
```

Expected:

- Master: NameNode, ResourceManager
- Workers: DataNode, NodeManager

---

7. Dataset ingestion

```
hdfs dfs -mkdir /datasets
hdfs dfs -put hadoop-3.3.6.tar.gz /datasets/
hdfs dfs -ls /datasets
```

7.1 Add my public key so that I can copy the large file to your VM:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCq5ElL3yW+thiyBKfsu7eXMhpFDpnBOs9FCHdoE/pb8msBp
vrBJJ5fu7CGQY5yeqdFsAlTVAynDDFkaRDco901hkYjDObonJvh3peWalqr6QkY6zgPALaTvtGH2z
MPWmWxzYdgt25y8LyARfS5w6hGetZs5AuEiyv37OncXy7NDxCarvqCImmnEpaPFviOTm61eOzTDEb
s06zQRRXYVpcMAM0uKeWl8Pfxp7wp+uQHg/Ga7YAOkOcHxjvaKoQdWQx8zvr8Z+v9oeVQZoJtxryd
LR8+j5G1sZOucIZkooq7Ou7gRkXjJEbS9bDJRboRUat35QP3NQHo4bXluARJggez Generated-
by-Nova
```

---

8. Web interfaces

First, open the ports on the security group on Strato

- NameNode UI: `http://hadoop-master:9870`
- ResourceManager UI: `http://hadoop-master:8088`

---

9. Debugging & failure-injection exercises (mandatory)

Exercise 1 — Kill a DataNode

```
kill -9 <DataNode PID>
```

Observe:

- Block under-replication
- Automatic re-replication

Exercise 2 — Break SSH

Remove a key and restart DFS.

Question:
Why does Hadoop fail even though ports are open?

Exercise 3 — Inspect block placement

```
hdfs fsck /datasets/hadoop-3.3.6.tar.gz -files -blocks -locations
```

---

10. Teaching checkpoints (discussion prompts)

Ask students:

1. Why is the NameNode a scalability concern?
2. Why does MapReduce require disk between stages?
3. Why does job startup take seconds?
4. Why does YARN predate containers?

These questions **set up Spark naturally**.

---

11. Common errors & fixes

Error	Likely cause
DataNodes not showing	Hostname mismatch
start-dfs.sh hangs	SSH broken
YARN apps fail	JAVA_HOME mismatch
Cluster restarts break HDFS	HDFS re-formatted