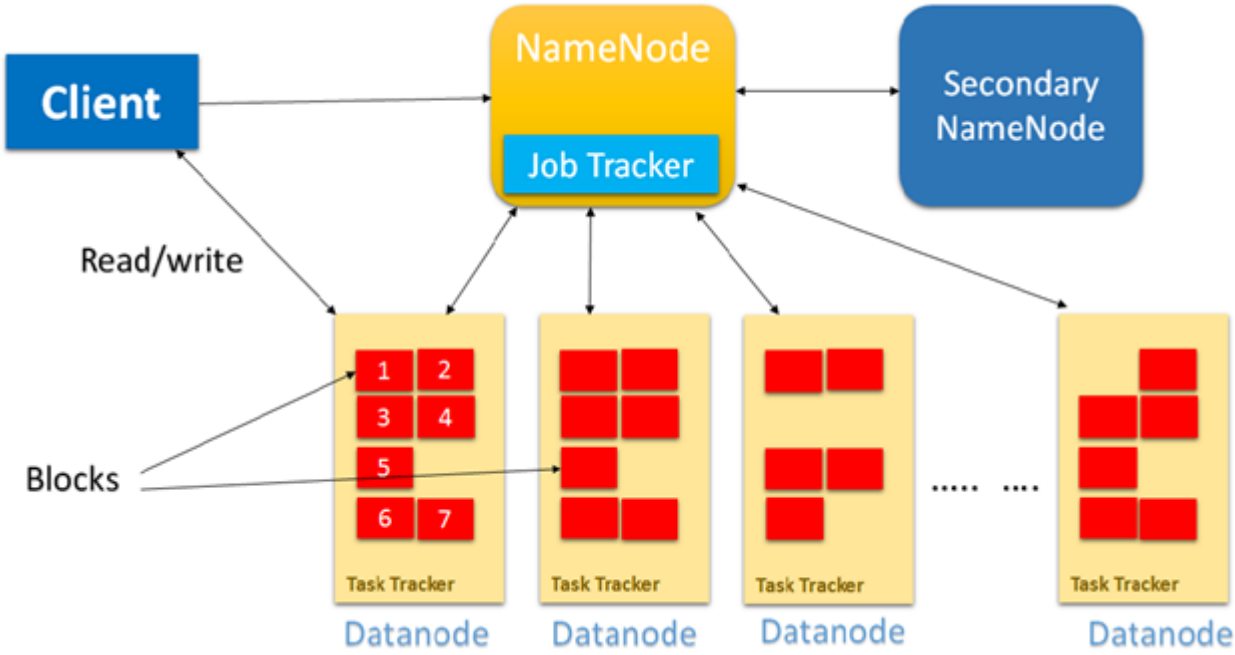
Hadoop Cluster Deployment Guide for Ubuntu

1 NameNode and 3 DataNodes



Preparation Phase

Step 1: System Requirements

Minimum Specifications:

- Ubuntu Server 24.04 LTS
- 4 Servers (1 NameNode, 3 DataNodes)
- Minimum Specs per Node:
  - CPU: 4 cores
  - RAM: 16 GB
  - Storage:
    - NameNode: 500 GB SSD
    - DataNodes: 1-2 TB HDD

Step 2: Pre-Installation Preparation

1. Update All Servers

```
sudo apt-get update && sudo apt-get upgrade -y
```

2. Install Basic Dependencies

```
sudo apt-get install -y openssh-server ssh net-tools wget curl
```

3. Configure Hostname (Do this on EACH server)

```
# On NameNode
sudo hostnamectl set-hostname hadoop-namenode
# On Datanodes
sudo hostnamectl set-hostname hadoop-datanode-1
sudo hostnamectl set-hostname hadoop-datanode-2
sudo hostnamectl set-hostname hadoop-datanode-3
```

Step 3: Network Configuration

```
#Edit `/etc/hosts` on ALL servers:

sudo nano /etc/hosts

# Add these lines (use actual IP addresses)
192.168.18.1 hadoop-namenode
192.168.18.11 hadoop-datanode-1
192.168.18.12 hadoop-datanode-2
192.168.18.13 hadoop-datanode-3
```

Step 4: Java Installation

```
# Install Java 8 or Java 11
sudo apt-get install -y openjdk-8-jdk

# Verify Java installation
java -version
javac -version
```

Step 5: Create Hadoop User (Optional but Recommended)

```
# Create hadoop group
sudo groupadd hadoop

# Create hadoop user
sudo useradd -m -s /bin/bash -g hadoop hdadmin

# Add to sudoers
sudo usermod -aG sudo hdadmin

# Set password
sudo passwd hdadmin
```

Step 6: SSH Passwordless Configuration

On NameNode:

```
# Switch to hdadmin user
sudo su - hdadmin

# Generate SSH Key
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa

# Copy public key to all nodes
ssh-copy-id hdadmin@hadoop-namenode
ssh-copy-id hdadmin@hadoop-datanode-1
ssh-copy-id hdadmin@hadoop-datanode-2
ssh-copy-id hdadmin@hadoop-datanode-3

# Test SSH connectivity
ssh hadoop-datanode-1 hostname
```

Step 7: Hadoop Download and Installation

```
# Download Hadoop (use latest stable version)
wget https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz

# Extract Hadoop
tar -xzvf hadoop-3.4.0.tar.gz

# Move to installation directory
sudo mv hadoop-3.4.0 /opt/hadoop

# Set ownership
sudo chown -R hdadmin:hadoop /opt/hadoop
```

Step 8: Environment Configuration

Edit `~/.bashrc` for hdadmin user:

```
# Hadoop Environment Variables
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/opt/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/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin

# Source the file
source ~/.bashrc
```

Step 9: Hadoop Configuration Files

```
# /opt/hadoop/etc/hadoop/core-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://hadoop-namenode:9000</value>
    </property>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>/opt/hadoop/tmp</value>
    </property>
</configuration>

# /opt/hadoop/etc/hadoop/hdfs-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:///opt/hadoop/hdfs/name</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:///opt/hadoop/hdfs/data</value>
    </property>
    <property>
        <name>dfs.replication</name>
        <value>3</value>
    </property>
    <property>
        <name>dfs.namenode.datanode.registration.ip-hostname-check</name>
        <value>false</value>
    </property>
</configuration>

# /opt/hadoop/etc/hadoop/yarn-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
    <property>
        <name>yarn.resourcemanager.hostname</name>
        <value>hadoop-namenode</value>
    </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>
</configuration>

# /opt/hadoop/etc/hadoop/mapred-site.xml
<?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>yarn.app.mapreduce.am.env</name>
        <value>HADOOP_MAPRED_HOME=/opt/hadoop</value>
    </property>
    <property>
        <name>mapreduce.map.env</name>
        <value>HADOOP_MAPRED_HOME=/opt/hadoop</value>
    </property>
    <property>
        <name>mapreduce.reduce.env</name>
        <value>HADOOP_MAPRED_HOME=/opt/hadoop</value>
    </property>
</configuration>

# /opt/hadoop/etc/hadoop/workers
hadoop-datanode-1
hadoop-datanode-2
hadoop-datanode-3


# /opt/hadoop/etc/hadoop/masters
hadoop-namenode
```

Step 10: Directory Preparation

```
# Create necessary directories
sudo mkdir -p /opt/hadoop/hdfs/name
sudo mkdir -p /opt/hadoop/hdfs/data
sudo mkdir -p /opt/hadoop/tmp

# Set correct permissions
sudo chown -R hdadmin:hadoop /opt/hadoop
```

Step 11: Distributed File Copy

On NameNode:

```
# Copy configuration to all nodes
for host in hadoop-namenode hadoop-datanode-1 hadoop-datanode-2 hadoop-
datanode-3; do
    scp /opt/hadoop/etc/hadoop/core-site.xml $host:/opt/hadoop/etc/hadoop/
    scp /opt/hadoop/etc/hadoop/hdfs-site.xml $host:/opt/hadoop/etc/hadoop/
    scp /opt/hadoop/etc/hadoop/yarn-site.xml $host:/opt/hadoop/etc/hadoop/
    scp /opt/hadoop/etc/hadoop/mapred-site.xml $host:/opt/hadoop/etc/hadoop/
    scp /opt/hadoop/etc/hadoop/workers $host:/opt/hadoop/etc/hadoop/
done
```

Step 12: Firewall Configuration

```
# Allow Hadoop ports
sudo ufw allow 9000   # NameNode RPC
sudo ufw allow 9870   # NameNode Web UI
sudo ufw allow 8088   # YARN ResourceManager
sudo ufw allow 22     # SSH
```

Step 13: Cluster Initialization

On NameNode:

```
# Format NameNode (ONLY ONCE)
hdfs namenode -format

# Start HDFS
start-dfs.sh

# Start YARN
start-yarn.sh
```

Step 14: Verification

```
# Check running services
jps

# HDFS Cluster Report
hdfs dfsadmin -report

# List DataNodes
hdfs dfsadmin -listDatanodeReport

# Test HDFS
hdfs dfs -mkdir /test
hdfs dfs -touchz /test/testfile.txt
```

```
administrator@hadoop-namenode:~$ hdfs dfsadmin -report
Configured Capacity: 3092533276672 (2.81 TB)
Present Capacity: 2916335852354 (2.65 TB)
DFS Remaining: 2916012126208 (2.65 TB)
DFS Used: 323726146 (308.73 MB)
DFS Used%: 0.01%
Replicated Blocks:
        Under replicated blocks: 6
        Blocks with corrupt replicas: 0
        Missing blocks: 0
        Missing blocks (with replication factor 1): 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0
Erasure Coded Block Groups:
        Low redundancy block groups: 0
        Block groups with corrupt internal blocks: 0
        Missing block groups: 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0

-------------------------------------------------
Live datanodes (3):
```

Step 15: Basic MapReduce Test

```
# Run Pi Calculation
yarn jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar pi
10 100
```

```
administrator@hadoop-namenode:~$ yarn jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar pi 10 100
Number of Maps  = 10
Samples per Map = 100
Wrote input for Map #0
Wrote input for Map #1
Wrote input for Map #2
Wrote input for Map #3
Wrote input for Map #4
Wrote input for Map #5
Wrote input for Map #6
Wrote input for Map #7
Wrote input for Map #8
Wrote input for Map #9
Starting Job
```

```
                Total megabyte-milliseconds taken by all reduce tasks=4495360
        Map-Reduce Framework
                Map input records=10
                Map output records=20
                Map output bytes=180
                Map output materialized bytes=280
                Input split bytes=1600
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=280
                Reduce input records=20
                Reduce output records=0
                Spilled Records=40
                Shuffled Maps =10
                Failed Shuffles=0
                Merged Map outputs=10
                GC time elapsed (ms)=1421
                CPU time spent (ms)=6820
                Physical memory (bytes) snapshot=3317080064
                Virtual memory (bytes) snapshot=28564611072
                Total committed heap usage (bytes)=5027397632
                Peak Map Physical memory (bytes)=302833664
                Peak Map Virtual memory (bytes)=2599448576
                Peak Reduce Physical memory (bytes)=338194432
                Peak Reduce Virtual memory (bytes)=2603483136
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=1180
        File Output Format Counters
                Bytes Written=97
Job Finished in 22.777 seconds
Estimated value of Pi is 3.14800000000000000000
```

Post-Deployment Recommendations

Monitoring

View the details on NameNode IP with browser

http://hadoop-namenode:9870

Summary

Security is off.

Safemode is off.

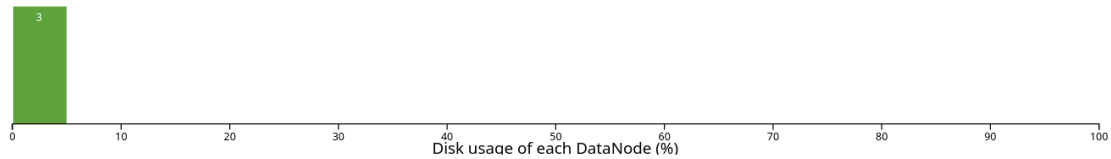
58 files and directories, 37 blocks (37 replicated blocks, 0 erasure coded block groups) = 95 total filesystem object(s).

Heap Memory used 326.6 MB of 867.5 MB Heap Memory. Max Heap Memory is 6.98 GB.

Non Heap Memory used 68.65 MB of 69.97 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	2.81 TB
Configured Remote Capacity:	0 B
DFS Used:	308.73 MB (0.01%)
Non DFS Used:	39.26 GB
DFS Remaining:	2.65 TB (94.29%)
Block Pool Used:	308.73 MB (0.01%)
DataNodes usages% (Min/Median/Max/stdDev):	0.01% / 0.01% / 0.01% / 0.00%
Live Nodes	3 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)
Decommissioning Nodes	0
Entering Maintenance Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	6

Datanode usage histogram



In operation

DataNode State: All Show 25 entries Search:

Node	Http Address	Last contact	Last Block Report	Used	Non DFS Used	Capacity	Blocks	Block pool used	Block pool usage StdDev	Version
/default-rack/hadoop-datanode-3:9866 (9866)	http://hadoop-datanode-3:9864	0s	9m	102.99 MB	13.09 GB	1008.82 GB	37	102.99 MB (0.01%)	0%	3.4.0
/default-rack/hadoop-datanode-1:9866 (:9866)	http://hadoop-datanode-1:9864	0s	9m	102.88 MB	13.09 GB	935.66 GB	37	102.88 MB (0.01%)	0%	3.4.0
/default-rack/hadoop-datanode-2:9866 (9866)	http://hadoop-datanode-2:9864	0s	9m	102.88 MB	13.09 GB	935.66 GB	37	102.88 MB (0.01%)	0%	3.4.0

1. Install monitoring tools

```
# Optional: Install Ganglia for monitoring
sudo apt-get install -y ganglia-monitor
```

Backup Strategy

1. Regular NameNode Metadata Backup

```
# Backup NameNode metadata
hdfs dfsadmin -metasave backup_filename
```

Performance Tuning

Edit `hadoop-env.sh`:

```
# Adjust JVM Heap Size
export HADOOP_HEAPSIZE=8192  # 8 GB
export YARN_HEAPSIZE=4096    # 4 GB
```

Troubleshooting Common Issues

1. Check Logs

```
# NameNode Logs
tail -f $HADOOP_HOME/logs/hadoop-*-namenode-*.log
#DataNode Logs
tail -f $HADOOP_HOME/logs/hadoop-*-datanode-*.log
```

2. Common Startup Issues

- Verify all nodes can resolve hostnames
- Check SSH passwordless configuration
- Ensure consistent Java versions
- Verify file permissions

Security Considerations

- Implement Kerberos Authentication
- Use encrypted communication
- Regular security patches
- Restrict file system permissions

Scaling Considerations

- Add more DataNodes for horizontal scaling
- Increase node resources for vertical scaling

Documentation

- Document cluster configuration
- Track hardware specifications
- Record version and patch levels

Recommended Reading

1. Apache Hadoop Documentation
2. Hadoop: The Definitive Guide (O'Reilly)
3. Professional Hadoop (Wrox)

Maintenance Checklist

- Monthly security updates
- Quarterly performance review
- Regular backup verification
- Monitor cluster health
- Plan for capacity expansion