

## LAB ASSIGNMENT 01

IMAAD IMRAN HAJWANE

202101132 / 23

### TOPIC: INSTALLATION OF HADOOP

#### Question:

What is Hadoop? Describe its Architecture.

#### Answer:

Hadoop is an open-source software framework used for storing and processing large datasets in a distributed computing environment. It is designed to scale up from a single server to thousands of machines, each offering local computation and storage.

#### Key Features of Hadoop

1. **Scalability:** Hadoop can store and process petabytes of data efficiently.
  2. **Fault Tolerance:** Automatically handles hardware failures.
  3. **Cost-Effective:** Utilizes commodity hardware to store large datasets.
  4. **Flexibility:** Can handle structured and unstructured data.
  5. **Speed:** Distributes data across clusters, allowing for faster data processing.
- 

#### Hadoop Architecture Overview

Hadoop architecture consists of the following main components:

1. **Hadoop Common**
2. **Hadoop Distributed File System (HDFS)**
3. **Yet Another Resource Negotiator (YARN)**
4. **MapReduce**

## 1. Hadoop Common

**Hadoop Common** is the set of shared utilities and libraries that support the other Hadoop modules. It includes essential Java libraries, scripts, and commands used throughout the Hadoop ecosystem. These components provide file system abstraction, IO utilities, and a platform for Hadoop modules to interact with the underlying operating system.

## 2. Hadoop Distributed File System (HDFS)

**HDFS** is the storage system of Hadoop. It is designed to store large datasets across a distributed cluster of machines. It provides high-throughput access to application data and is highly fault-tolerant.

### Key Components of HDFS:

- **NameNode:**
  - **Role:** The NameNode is the master server that manages the file system namespace and controls access to files. It maintains the metadata of all the files and directories in the HDFS, such as the location of data blocks, permissions, and hierarchy.
  - **Functionality:** It keeps track of where data is stored across the cluster and manages operations such as opening, closing, and renaming files and directories.
  - **Fault Tolerance:** The NameNode is a single point of failure, so to improve fault tolerance, Hadoop provides a Secondary NameNode, which takes periodic snapshots of the NameNode's metadata. In newer versions, high availability is achieved using a standby NameNode.

## Installation of Hadoop:

x

```
| sudo apt update && sudo apt install openjdk-8-jdk
```

```
| java -version
```

```
| sudo apt install ssh
```

```
| sudo adduser hadoop
```

```
| su — hadoop
```

```
| ssh-keygen -t rsa
```

```
| cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

```
| chmod 640 ~/.ssh/authorized_keys
```

```
| ssh localhost
```

```
| su — hadoop
```

```
| wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz
```

```
| tar -xvzf hadoop-3.3.6.tar.gz
```

```
| mv hadoop-3.3.6 hadoop
```

```
| nano ~/.bashrc
```

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export HADOOP_HOME=/home/hadoop/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 HADOOP_YARN_HOME=$HADOOP_HOME

export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native

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

export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

```
source ~/.bashrc
```

Additionally, configure the JAVA\_HOME variable in the hadoop-env.sh file.  
Open the Hadoop environment configuration file with a text editor:

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

```
cd hadoop/
```

```
mkdir -p ~/hadoopdata/hdfs/{namenode,datanode}
```

**Edit core-site.xml:**

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

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

**Format the Namenode:**

Format the Hadoop Namenode by running the following command:

```
hdfs namenode -format
```

```
start-all.sh
```

| jps

## Screenshots:

```
hadoop@imaad: ~  
imaad@imaad:~$ su - hadoop  
Password:  
hadoop@imaad:~$ start-all.sh  
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.  
WARNING: This is not a recommended production deployment configuration.  
WARNING: Use CTRL-C to abort.  
Starting namenodes on [localhost]  
Starting datanodes  
Starting secondary namenodes [imaad]  
Starting resourcemanager  
Starting nodemanagers  
hadoop@imaad:~$ jps  
4419 DataNode  
4230 NameNode  
4616 SecondaryNameNode  
4825 ResourceManager  
4953 NodeManager  
5163 Jps  
hadoop@imaad:~$
```

## Resource Manager:

The screenshot displays the Hadoop Resource Manager web interface. The top navigation bar includes tabs for 'All Applications', 'DataNode Information', and 'NameNode Information'. The main content area is titled 'All Applications' and features a sidebar with a 'Cluster' dropdown menu. The 'Cluster' menu is expanded, showing options like 'About', 'Nodes', 'Node Labels', 'Applications', and 'Scheduler'. The main panel displays 'Cluster Metrics' with a table showing 'Apps Submitted', 'Apps Pending', 'Apps Running', 'Apps Completed', 'Containers Running', 'Used Resources', and 'Total Resources'. Below this, 'Cluster Nodes Metrics' shows 'Active Nodes', 'Decommissioning Nodes', 'Decommissioned Nodes', and 'Lost Nodes'. The 'Scheduler Metrics' section includes 'Scheduler Type', 'Scheduling Resource Type', 'Minimum Allocation', and 'Maximum Allocation'. A table at the bottom lists application details with columns: ID, User, Name, Application Type, Application Tags, Queue, Application Priority, StartTime, LaunchTime, FinishTime, State, FinalStatus, Running Containers, Allocated CPU VCores, and Allocated Memory MB. The table currently shows 'No data available in table'.

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB
No data available in table														

Name Node:

All ApplicationsDataNode InformationNamenode information+  
localhost:9870/dfshealth.html#tab-overview

HadoopOverviewDatanodesDatanode Volume FailuresSnapshotStartup ProgressUtilities

Overview 'localhost:9000' (active)

Started:Mon Jul 29 18:47:38 +0530 2024

Version:3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c

Compiled:Sun Jun 18 13:52:00 +0530 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)

Cluster ID:CID-8cf395dc-244e-4322-aab6-9fa86cf65d9

Block Pool ID:BP-1271918429-127.0.1.1-1722258952135

Summary

Security is off.  
Safemode is off.  
1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).  
Heap Memory used 42.83 MB of 60.48 MB Heap Memory. Max Heap Memory is 595.5 MB.  
Non Heap Memory used 52.3 MB of 53.4 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:23.94 GB

Configured Remote Capacity:0 B

DFS Used:28 KB (0%)

Non DFS Used:19.84 GB

DFS Remaining:2.86 GB (11.96%)

Data Node:

All ApplicationsDataNode InformationNamenode information+  
localhost:9864/datanode.html

HadoopOverviewUtilities

DataNode on imaad:9866

Cluster ID:CID-8cf395dc-244e-4322-aab6-9fa86cf65d9

Started:Mon Jul 29 18:47:43 +0530 2024

Version:3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c

Block Pools

Namenode Address	Namenode HA State	Block Pool ID	Actor State	Last Heartbeat Sent	Last Heartbeat Response	Last Block Report	Last Block Report Size (Max Size)
localhost:9000	active	BP-1271918429-127.0.1.1-1722258952135	RUNNING	1s	1s	11 minutes	0 B (128 MB)

Volume Information

Directory	StorageType	Capacity Used	Capacity Left	Capacity Reserved	Reserved Space for Replicas	Blocks
/home/hadoop/hadoopdata/hdfs/datanode	DISK	28 KB	2.86 GB	0 B	0 B	0