Title : **Mini-Project**

Setup your own cloud for Software as a Service (SaaS) over the existing LAN in your laboratory. In this assignment you have to write your own code for cloud controller using open-source technologies to implement with HDFS. Implement the basic operations may be like to divide the file in segments/blocks and upload/ download file on/from cloud in encrypted form.

Name : - Mehatab Mahibub Sanadi

Roll No. : - CO3056

Class : - TE Computer

Subject : - Cloud Computing

Date : -

****

- **HDFS (Hadoop Distributed File System)**
  - A distributed file system that stores large datasets across multiple machines.
  - Provides high fault tolerance and scalability.
  - Ideal for storing structured, semi-structured, and unstructured data.

- **YARN (Yet Another Resource Negotiator)**
  - Manages and schedules resources in a Hadoop cluster.
  - Allocates CPU, memory, and handles job execution.
  - Supports multiple data processing engines like MapReduce and Spark.

- **MapReduce**
  - A programming model for batch processing of large datasets.
  - Consists of Map (data filtering and sorting) and Reduce (aggregation).
  - Best for processing historical or static data.

- **Spark**
  - An in-memory data processing engine, faster than MapReduce.
  - Supports real-time data processing, machine learning, and stream processing.
  - Offers APIs in Python, Java, Scala, and R.

🔹 **PIG & HIVE**

- **PIG:** A high-level scripting platform for data transformation (uses Pig Latin).
- **HIVE:** A SQL-like interface to query and manage large datasets.
- Both simplify querying and processing data stored in HDFS.

🔹 **HBase**

- A NoSQL column-oriented database built on top of HDFS.
- Supports real-time read/write access to large datasets.
- Ideal for random, fast access to data (e.g., medical or financial records).

🔹 **Mahout & Spark MLlib**

- **Mahout:** Provides scalable machine learning algorithms (older, MapReduce-based).
- **Spark MLlib:** Modern, in-memory ML library with classification, clustering, etc.
- Used for predictive analytics on big data.

---

s 🔹 **Solr & Lucene**

- **Lucene:** A text search engine library.
- **Solr:** A full-text search platform built on Lucene.
- Used for indexing, searching, and real-time querying over big data (e.g., logs, documents).

Here's a **step-by-step guide** to set up and run a **Single Node Hadoop Cluster** (pseudo-distributed mode) on **Ubuntu**. This will include installing Java, Hadoop 1.2.1, SSH configuration, environment variables, and starting Hadoop daemons.

1. Update & Install Required Packages

Open terminal and run:

```
sudo apt update
sudo apt install ssh rsync curl wget -y
```

2. Configure Hostname and DNS

Edit your /etc/hosts file:

```
sudo nano /etc/hosts
```

Add or modify:

```
127.0.0.1       localhost
127.0.1.1       prasad.com pib

# For IPv6:
::1             ip6-localhost ip6-loopback
fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
ff02::3         ip6-allhosts
```

3. Set up Passwordless SSH

Generate SSH key:

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

Just press Enter through the prompts (no passphrase needed).

Then:

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
```

Test SSH:

```
ssh localhost
```

(If it logs in without asking a password, it's working.)

4. Download & Extract Hadoop

```
wget https://archive.apache.org/dist/hadoop/common/hadoop-1.2.1/hadoop-1.2.1.tar.gz
tar -xzvf hadoop-1.2.1.tar.gz
sudo mv hadoop-1.2.1 /usr/local/hadoop
```

5. Install Java 1.8

```
sudo apt install openjdk-8-jdk -y
```

Verify:

```
java -version
```

6. Configure Environment Variables

Edit .bashrc:

nano ~/.bashrc

Add to bottom:

```
export HADOOP_PREFIX=/usr/local/hadoop
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export PATH=$PATH:$HADOOP_PREFIX/bin:$JAVA_HOME/bin
export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true
```

Apply changes:

source ~/.bashrc

7. Configure Hadoop Environment

Edit Hadoop's hadoop-env.sh:

nano /usr/local/hadoop/conf/hadoop-env.sh

Set:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true
```

8. Create Hadoop Required Directories

```
sudo mkdir -p /usr/local/hadoop/tmp
sudo chown -R $USER:$USER /usr/local/hadoop
```

9. Configure Hadoop XML Files

- **core-site.xml**

nano /usr/local/hadoop/conf/core-site.xml

Paste:

```
<configuration>
 <property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:9000</value>
 </property>
 <property>
  <name>hadoop.tmp.dir</name>
```

```
 <value>/usr/local/hadoop/tmp</value>
 </property>
</configuration>
```

- **hdfs-site.xml**

nano /usr/local/hadoop/conf/hdfs-site.xml

Paste:

```
<configuration>
 <property>
  <name>dfs.replication</name>
  <value>1</value>
 </property>
</configuration>
```

- **mapred-site.xml**

Create file (if not exists):

```
cp /usr/local/hadoop/conf/mapred-site.xml.template /usr/local/hadoop/conf/mapred-site.xml
nano /usr/local/hadoop/conf/mapred-site.xml
```

Paste:

```
<configuration>
 <property>
  <name>mapred.job.tracker</name>
  <value>localhost:9001</value>
 </property>
</configuration>
```

- **masters**

nano /usr/local/hadoop/conf/masters

Add:

pib

- **slaves**

nano /usr/local/hadoop/conf/slaves

Add:

pib

10. Format the NameNode

hadoop namenode -format

You should see Storage directory ... has been successfully formatted.

11. Start Hadoop Daemons

All at once

start-all.sh

12. Verify Daemons Running

Use jps (requires Java):

jps

Expected Output:

NameNode
DataNode
SecondaryNameNode
JobTracker
TaskTracker

13. Stop Hadoop

stop-all.sh

Terminal Output :-

anuj@anuj-GF63-Thin-11UC:~$ nano ~/.bashrc

anuj@anuj-GF63-Thin-11UC:~$ source ~/.bashrc

anuj@anuj-GF63-Thin-11UC:~$ nano /usr/local/hadoop/conf/hadoop-env.sh

anuj@anuj-GF63-Thin-11UC:~$ sudo mkdir -p /usr/local/hadoop/tmp

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

anuj@anuj-GF63-Thin-11UC:~$ nano /usr/local/hadoop/conf/core-site.xml

anuj@anuj-GF63-Thin-11UC:~$ nano /usr/local/hadoop/conf/hdfs-site.xml

anuj@anuj-GF63-Thin-11UC:~$ cp /usr/local/hadoop/conf/mapred-site.xml.template /usr/local/hadoop/conf/mapred-site.xml

```
nano /usr/local/hadoop/conf/mapred-site.xml

cp: cannot stat '/usr/local/hadoop/conf/mapred-site.xml.template': No such file or directory

anuj@anuj-GF63-Thin-11UC:~$ ^[[200~nano /usr/local/hadoop/conf/masters

nano: command not found

anuj@anuj-GF63-Thin-11UC:~$ nano /usr/local/hadoop/conf/masters

anuj@anuj-GF63-Thin-11UC:~$ nano /usr/local/hadoop/conf/slaves

anuj@anuj-GF63-Thin-11UC:~$ hadoop namenode -format

25/04/11 18:19:47 INFO namenode.NameNode: STARTUP_MSG:

/************************************************************

STARTUP_MSG: Starting NameNode

STARTUP_MSG:   host = anuj-GF63-Thin-11UC/127.0.1.1

STARTUP_MSG:   args = [-format]

STARTUP_MSG:   version = 1.2.1

STARTUP_MSG:   build = https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.2 -r
1503152; compiled by 'mattf' on Mon Jul 22 15:23:09 PDT 2013

STARTUP_MSG:   java = 1.8.0_392

************************************************************/

25/04/11 18:19:47 INFO util.GSet: Computing capacity for map BlocksMap

25/04/11 18:19:47 INFO util.GSet: VM type       = 64-bit

25/04/11 18:19:47 INFO util.GSet: 2.0% max memory = 932184064

25/04/11 18:19:47 INFO util.GSet: capacity      = 2^21 = 2097152 entries

25/04/11 18:19:47 INFO util.GSet: recommended=2097152, actual=2097152

25/04/11 18:19:47 INFO namenode.FSNamesystem: fsOwner=anuj

25/04/11 18:19:47 INFO namenode.FSNamesystem: supergroup=supergroup

25/04/11 18:19:47 INFO namenode.FSNamesystem: isPermissionEnabled=true

25/04/11 18:19:47 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100

25/04/11 18:19:47 INFO namenode.FSNamesystem: isAccessTokenEnabled=false
accessKeyUpdateInterval=0 min(s), accessTokenLifetime=0 min(s)

25/04/11 18:19:47 INFO namenode.FSEditLog: dfs.namenode.edits.toleration.length = 0

25/04/11 18:19:47 INFO namenode.NameNode: Caching file names occuring more than 10 times

25/04/11 18:19:47 INFO common.Storage: Image file
/usr/local/hadoop/tmp/dfs/name/current/fsimage of size 110 bytes saved in 0 seconds.

25/04/11 18:19:47 INFO namenode.FSEditLog: closing edit log: position=4,
editlog=/usr/local/hadoop/tmp/dfs/name/current/edits
```

25/04/11 18:19:47 INFO namenode.FSEditLog: close success: truncate to 4, editlog=/usr/local/hadoop/tmp/dfs/name/current/edits

25/04/11 18:19:47 INFO common.Storage: Storage directory /usr/local/hadoop/tmp/dfs/name has been successfully formatted.

25/04/11 18:19:47 INFO namenode.NameNode: SHUTDOWN_MSG:

/**

SHUTDOWN_MSG: Shutting down NameNode at anuj-GF63-Thin-11UC/127.0.1.1

**/

anuj@anuj-GF63-Thin-11UC:~$ start-dfs.sh

start-mapred.sh

starting namenode, logging to /usr/local/hadoop/libexec/../logs/hadoop-anuj-namenode-anuj-GF63-Thin-11UC.out

pib: ssh: Could not resolve hostname pib: Temporary failure in name resolution

localhost: starting datanode, logging to /usr/local/hadoop/libexec/../logs/hadoop-anuj-datanode-anuj-GF63-Thin-11UC.out

pib: ssh: Could not resolve hostname pib: Temporary failure in name resolution

localhost: starting secondarynamenode, logging to /usr/local/hadoop/libexec/../logs/hadoop-anuj-secondarynamenode-anuj-GF63-Thin-11UC.out

starting jobtracker, logging to /usr/local/hadoop/libexec/../logs/hadoop-anuj-jobtracker-anuj-GF63-Thin-11UC.out

pib: ssh: Could not resolve hostname pib: Temporary failure in name resolution

localhost: starting tasktracker, logging to /usr/local/hadoop/libexec/../logs/hadoop-anuj-tasktracker-anuj-GF63-Thin-11UC.out

anuj@anuj-GF63-Thin-11UC:~$ start-all.sh

namenode running as process 7894. Stop it first.

pib: ssh: Could not resolve hostname pib: Temporary failure in name resolution

localhost: datanode running as process 8077. Stop it first.

pib: ssh: Could not resolve hostname pib: Temporary failure in name resolution

localhost: secondarynamenode running as process 8257. Stop it first.

jobtracker running as process 8367. Stop it first.

pib: ssh: Could not resolve hostname pib: Temporary failure in name resolution

localhost: tasktracker running as process 8560. Stop it first.

anuj@anuj-GF63-Thin-11UC:~$ jps

8560 TaskTracker

8977 Jps

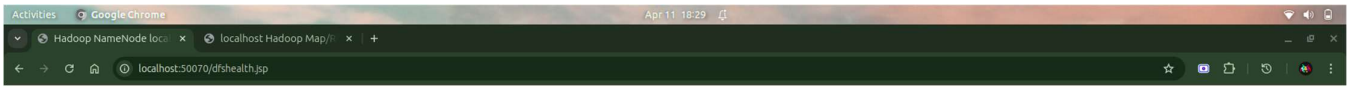
8257 SecondaryNameNode

7894 NameNode

8077 DataNode

8367 JobTracker

Output:-



NameNode 'localhost:9000'

Started: Fri Apr 11 18:21:16 IST 2025
Version: 1.2.1, r1503152
Compiled: Mon Jul 22 15:23:09 PDT 2013 by mattf
Upgrades: There are no upgrades in progress.

Browse the filesystem
Namenode Logs

Cluster Summary

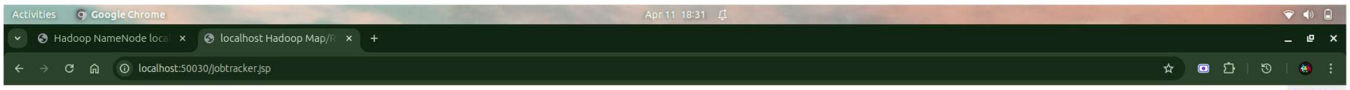
8 files and directories, 1 blocks = 9 total. Heap Size is 236 MB / 889 MB (26%)

Configured Capacity	:	52.77 GB
DFS Used	:	28.01 KB
Non DFS Used	:	38.23 GB
DFS Remaining	:	14.54 GB
DFS Used%	:	0 %
DFS Remaining%	:	27.55 %
Live Nodes	:	1
Dead Nodes	:	0
Decommissioning Nodes	:	0
Number of Under-Replicated Blocks	:	0

NameNode Storage:

Storage Directory	Type	State
/usr/local/hadoop/tmp/dfs/name	IMAGE_AND_EDITS	Active

This is Apache Hadoop release 1.2.1



Quick Links

localhost Hadoop Map/Reduce Administration

State: RUNNING
Started: Fri Apr 11 18:21:20 IST 2025
Version: 1.2.1, r1503152
Compiled: Mon Jul 22 15:23:09 PDT 2013 by mattf
Identifier: 202504111821
SafeMode: OFF

Cluster Summary (Heap Size is 236 MB/889 MB)

Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes	Graylisted Nodes	Excluded Nodes
0	0	0	1	0	0	0	0	2	2	4.00	0	0	0

Scheduling Information

Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name)
Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

none

Retired Jobs

none

Local Logs

Log directory, Job Tracker History

This is Apache Hadoop release 1.2.1