**FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING**

**Department of Computer Engineering**

**Academic Year 2025-26**

# Rubrics for Lab Experiments

**Class : B*.E. Computer*  Subject Name :*BDA Lab Engineering***

**Semester : VII Subject Code :CSL702**

|  |  |
| --- | --- |
| **Practical No:** | 2 |
| **Title:** | Hands on Hadoop HDFS |
| **Date of Performance:** | 15/07/2025 |
| **Roll No:** | 9913 |
| **Name of the Student:** | Mark Lopes |

**Evaluation:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Performance**  **Indicator** | **Below average** | **Average** | **Good** | **Excellent** | **Marks** |
| **On time**  **Submission (2)** | Not submitted  (0) | Submitted after deadline (1) | Early or on time submission(2) | --- |  |
| **Test cases and output**  **(4)** | Incorr ect output  (1) | The expected output is verified only a for few test cases (2) | The expected  output is Verified for all test cases but is not  presentable (3) | Expected output is obtained for all test cases. Presentable and easy to follow  (4) |  |
| **Coding efficiency (2)** | The code is not  structured at all (0) | The code is structured but not efficient (1) | The code is Structured and efficient.  (2) | - |  |
| **Knowledge(2)** | Basic concepts not clear  (0) | Understood the basic concepts  (1) | Could explain the concept with suitable example (1.5) | Could relate the theory with real world application(2) |  |
| **Total** |  |  |  |  |  |

**Experiment** **No 2**

Aim: Hands on Hadoop HDFS

Objective:

The objective of this lab experiment is to provide hands-on experience with Hadoop

Distributed File System (HDFS). Students will learn how to interact with HDFS, manage

files and directories, understand replication and fault tolerance, and perform basic

administrative tasks.

Tools and Technologies:

● Hadoop: Specifically focusing on HDFS (Hadoop Distributed File System).

● Virtual or physical machines capable of running a Linux distribution (e.g., Ubuntu, CentOS).

Pre-requisites:

● Basic understanding of Linux/Unix commands.

● Familiarity with Java programming (helpful but not mandatory).

Equipment Required:

● Virtual or physical machines capable of running a Linux distribution (e.g., Ubuntu, CentOS).

● Sufficient memory and disk space to accommodate Hadoop&#39;s requirements (minimum of 4GB

RAM recommended per node).

Experiment Steps:

1. Setting Up the Environment:

o Prepare the environment by setting up virtual machines (VMs) or physical machines

with a Linux distribution (e.g., Ubuntu Server).

o Ensure that each machine has a static IP address and can communicate with each

other over the network.

2. Installing Java Development Kit (JDK):

o Hadoop requires Java, so install JDK on all machines that will be part of the Hadoop

cluster.

o Example command to install OpenJDK:

bash

Copy code

sudo apt-get update

sudo apt-get install openjdk-8-jdk

3. Downloading and Extracting Hadoop:

o Download the desired version of Hadoop from the Apache Hadoop website

(https://hadoop.apache.org/releases.html).

o Extract the downloaded Hadoop tarball to a suitable directory on each machine in

your cluster.

bash

Copy code

tar -xzvf hadoop-3.x.x.tar.gz -C /opt

4. Configuring Hadoop Environment Variables:

o Set up Hadoop environment variables in the .bashrc or .bash\_profile file for each user:

bash

Copy code

export HADOOP\_HOME=/opt/hadoop-3.x.x

export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin

5. Configuring HDFS:

o Navigate to the Hadoop configuration directory ($HADOOP\_HOME/etc/hadoop)

and edit core-site.xml and hdfs-site.xml files.

o core-site.xml:

xml

Copy code

&lt;configuration&gt;

&lt;property&gt;

&lt;name&gt;fs.defaultFS&lt;/name&gt;

&lt;value&gt;hdfs://namenode\_host:9000&lt;/value&gt;

&lt;/property&gt;

&lt;/configuration&gt;

o hdfs-site.xml:

xml

Copy code

&lt;configuration&gt;

&lt;property&gt;

&lt;name&gt;dfs.replication&lt;/name&gt;

&lt;value&gt;3&lt;/value&gt; &lt;!-- Adjust replication factor as needed --&gt;

&lt;/property&gt;

&lt;/configuration&gt;

6. Formatting HDFS Namenode:

o Before starting HDFS, format the namenode to initialize the filesystem metadata:

bash

Copy code

hdfs namenode -format

7. Starting HDFS Services:

o Start HDFS services using the provided scripts:

bash

Copy code

start-dfs.sh

8. Interacting with HDFS:

o Use Hadoop commands (hdfs dfs) to interact with HDFS:

▪ Creating a directory in HDFS:

bash

Copy code

hdfs dfs -mkdir /user

▪ Copying files from local filesystem to HDFS:

bash

Copy code

hdfs dfs -put /local/path/to/file /hdfs/path/

▪ Listing files in a directory in HDFS:

bash

Copy code

hdfs dfs -ls /hdfs/path/

▪ Reading files from HDFS:

bash

Copy code

hdfs dfs -cat /hdfs/path/to/file

9. Understanding Replication and Fault Tolerance:

o Discuss the concept of replication factor and how it ensures fault tolerance.

o Simulate a failure scenario (e.g., shutdown a datanode) and observe how HDFS

maintains data availability.

10. Stopping HDFS Services:

o Stop HDFS services when done experimenting:

bash

Copy code

stop-dfs.sh

11. Observations and Conclusion:

o Document any issues encountered during setup and how they were resolved.

o Discuss the benefits of using HDFS for storing and managing large datasets.

o Reflect on the role of HDFS in the Hadoop ecosystem and its importance in big data

processing.

Expected Outcome:

By the end of this experiment, students should have a solid understanding of how HDFS

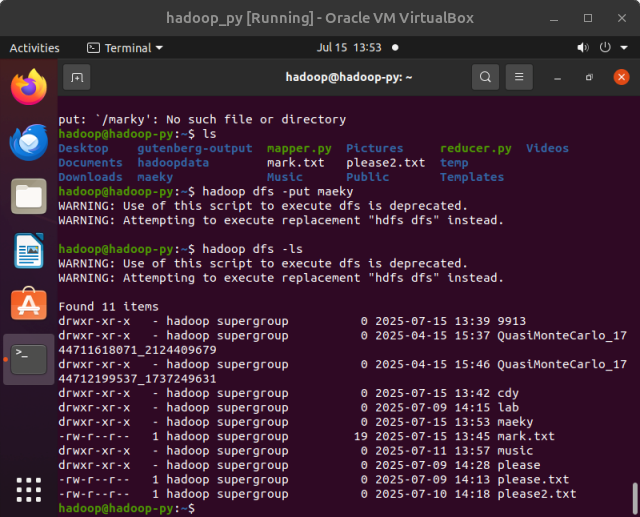
operates within a Hadoop cluster. They should be able to perform basic administrative tasks

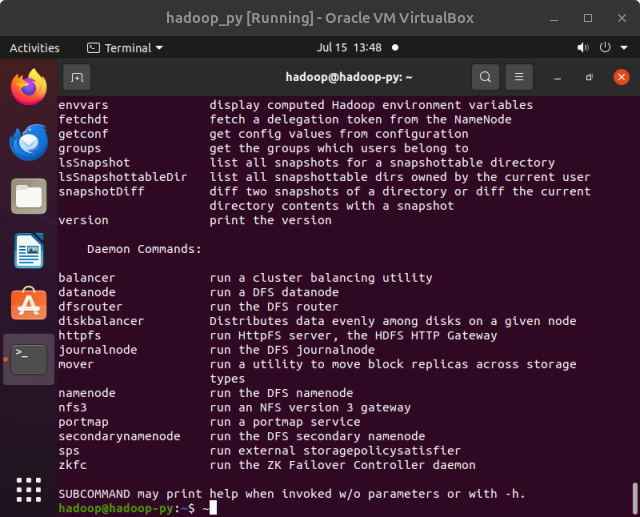
such as creating directories, copying files, and understanding replication strategies. Students

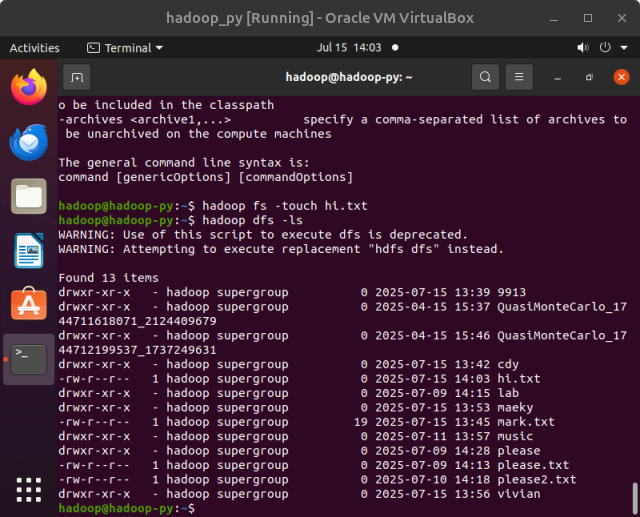
should also gain insights into HDFS&#39;s fault tolerance mechanisms and its role in supporting

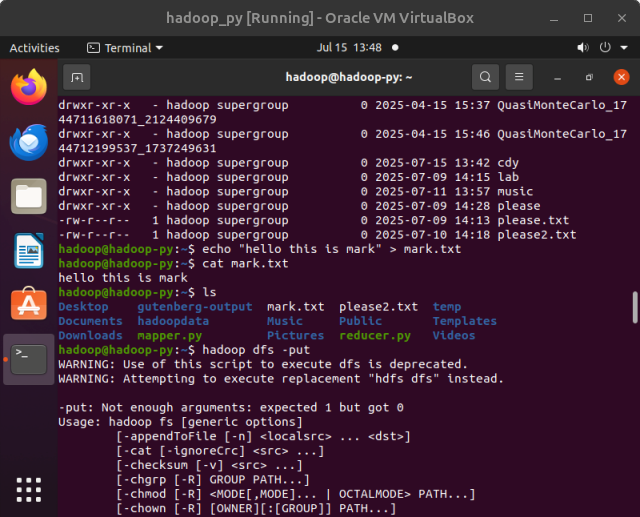
distributed data storage.

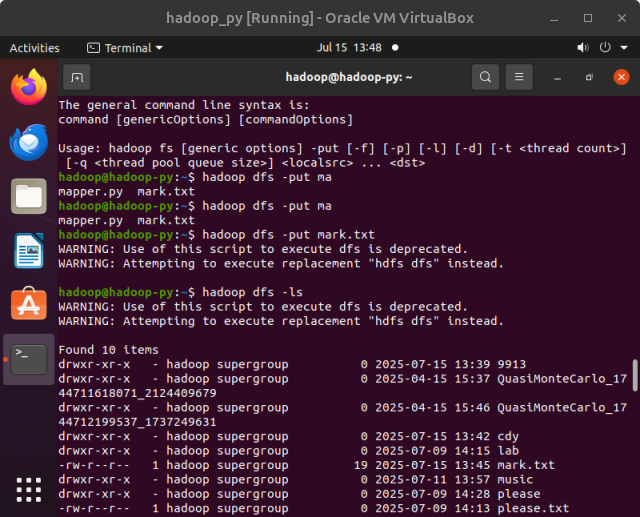
Conclusion:

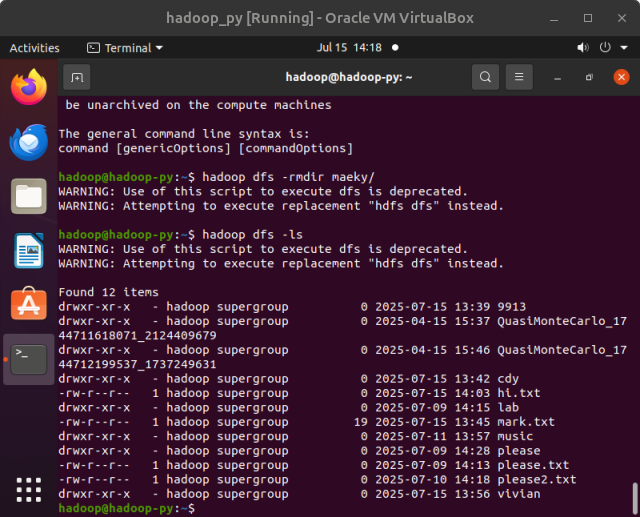


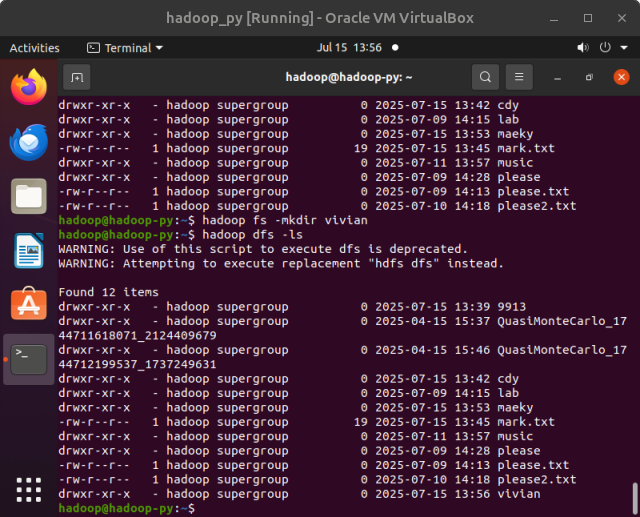


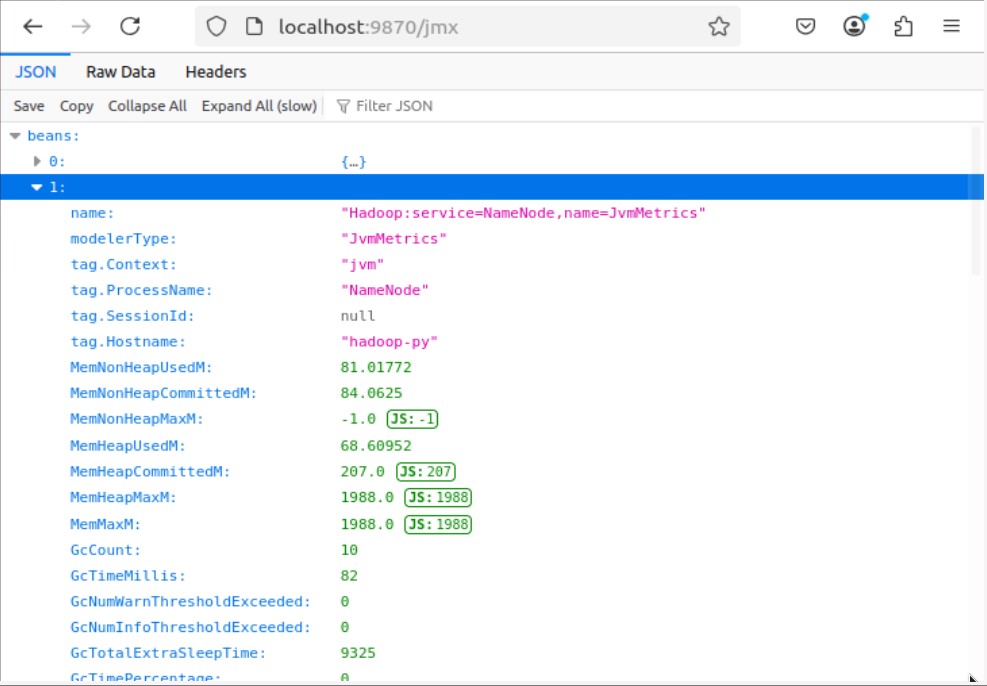
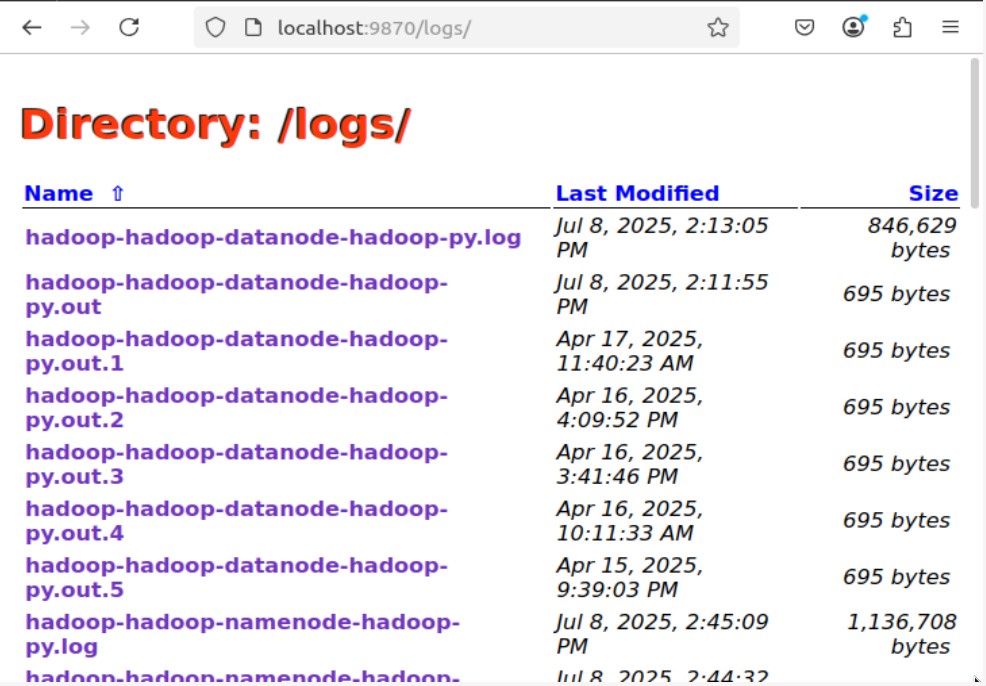
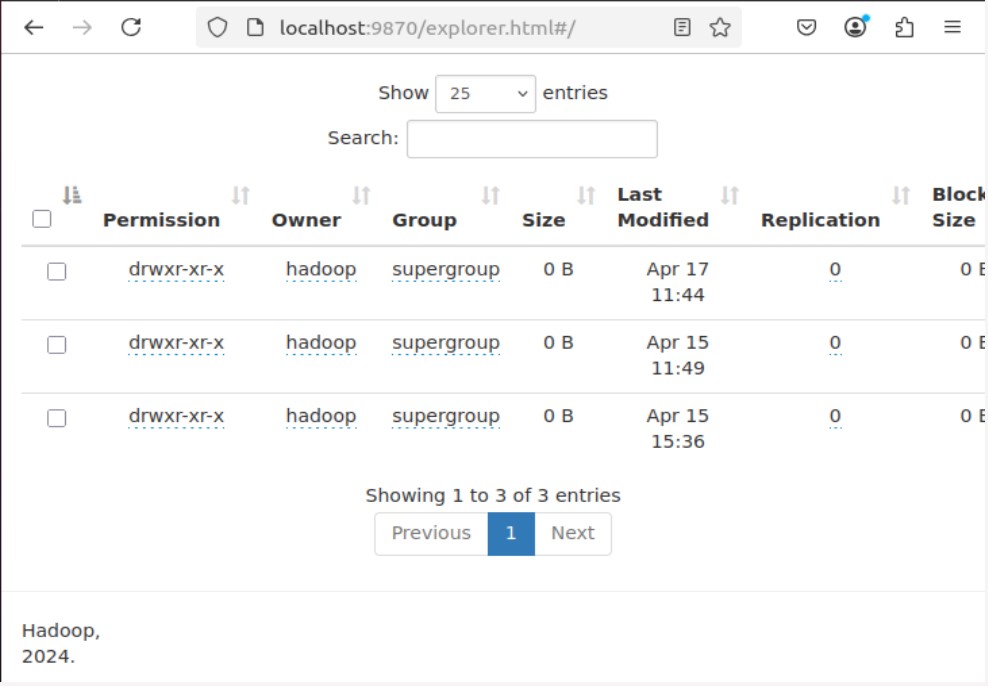
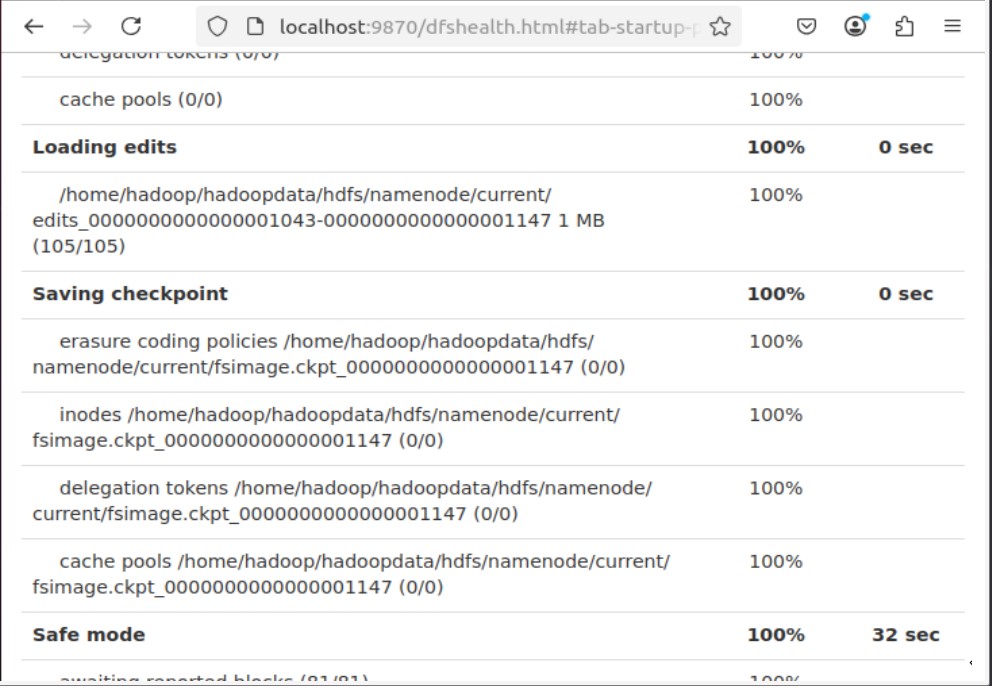
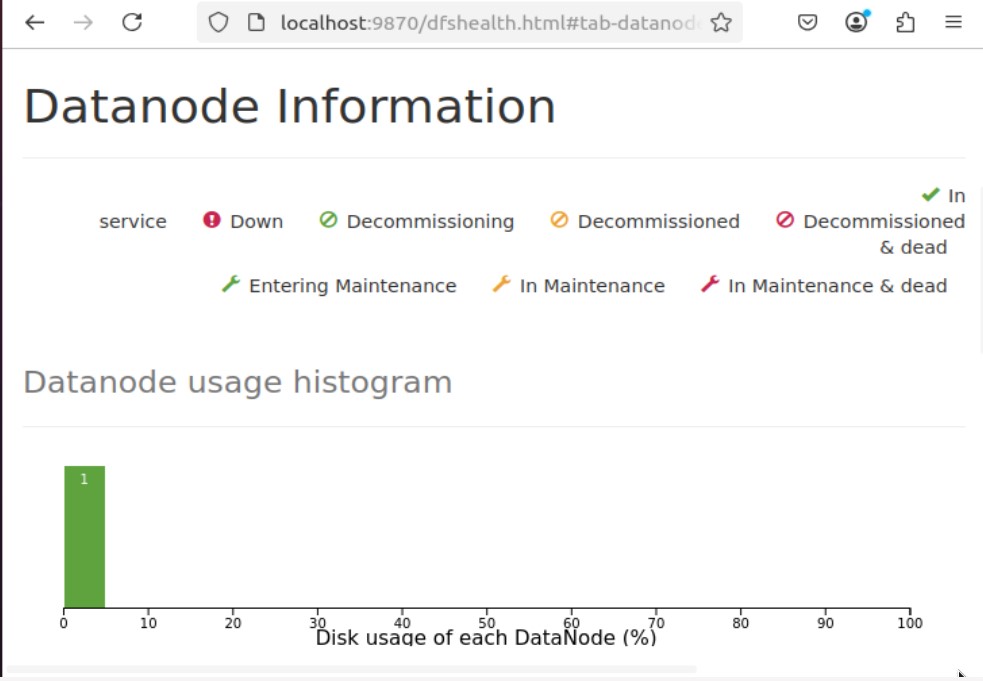
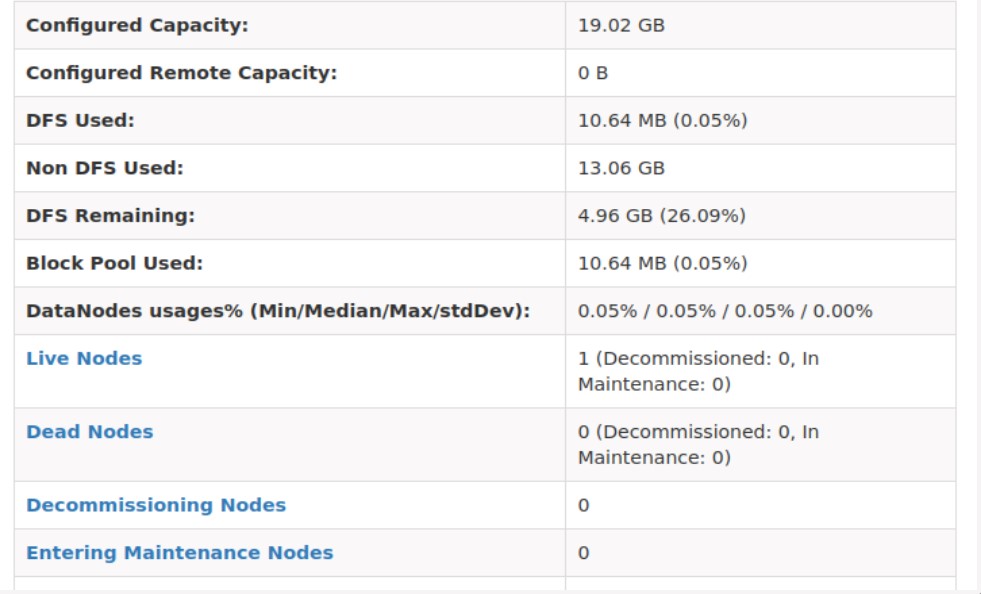
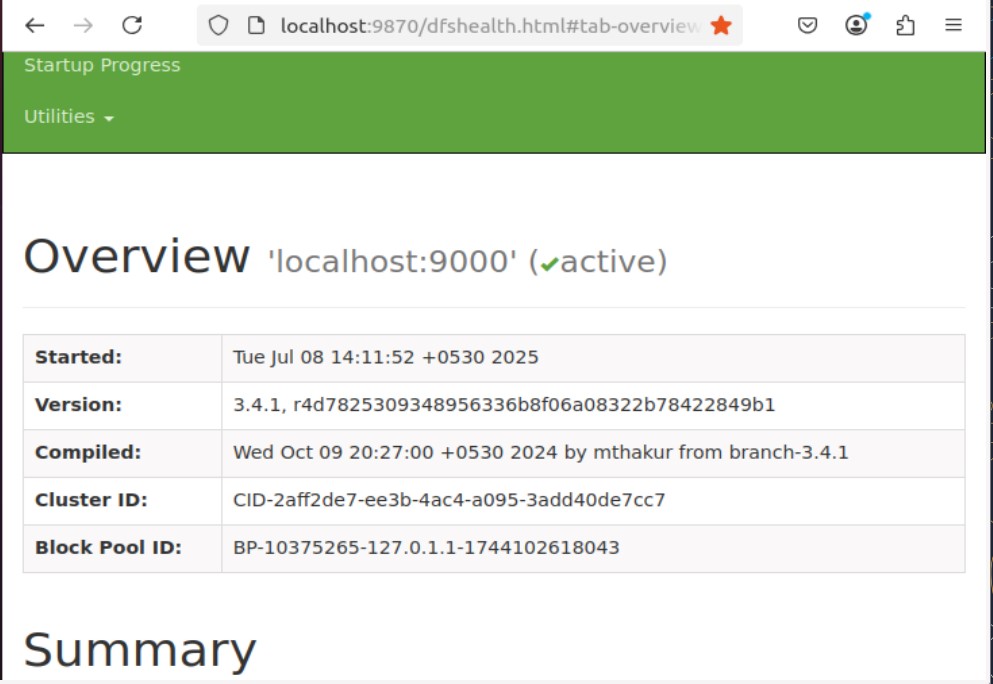
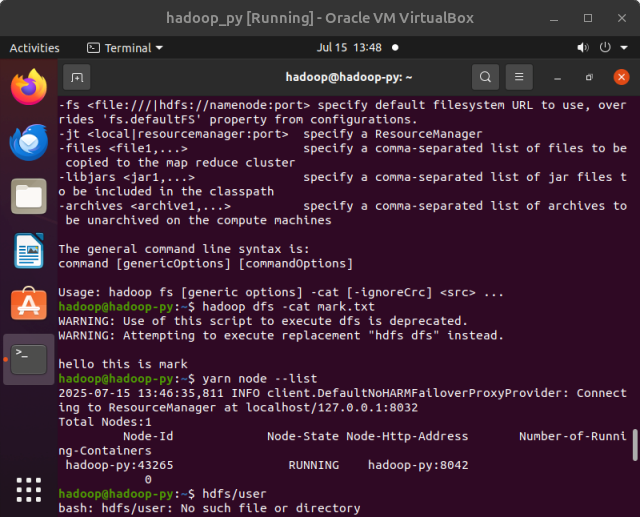












# Postlab:-

1. What are the main components of a Hadoop application?

HDFS (Hadoop Distributed File System):

Stores large files across multiple machines with fault tolerance using replication.

YARN (Yet Another Resource Negotiator):

Manages cluster resources and job scheduling.

MapReduce:

A programming model used for distributed data processing (map = split, reduce = aggregate).

Hadoop Common:

Provides essential Java libraries and utilities used by other modules.

1. Difference between NameNode, Backup Node, and Checkpoint Node:

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Function | Real-Time  Sync | Failure Recovery  Role |
| NameNode | Manages file system metadata like file names, directories, and block locations. | Yes | Acts as the master; essential for HDFS operation. |
| Backup Node | Maintains an in-memory, up-to-date copy of metadata from the NameNode. | Yes | Can immediately take over if  NameNode fails. |
| Checkpoint  Node | Periodically downloads and merges fsimage and edits, then sends a new fsimage to NameNode. | No | Reduces NameNode  startup time, not used for failover. |

1. Explain the use of cat, du, du -s:
   * cat (concatenate):

Used to view the contents of files in the terminal. Example: cat file.txt

* + du (disk usage):

Shows the space used by files and directories. Example: du myfolder/

* + du -s (summary):

Displays the total size of a folder, instead of listing all subdirectories. Example: du

-s myfolder/