VNUHCM-UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY

REPORT

INTRODUCTION TO BIG DATA

< Lab 01 - A Gentle Introduction to Hadoop >

Student: 21127104 - Doan Ngoc Mai
21127129 - Le Nguyen Kieu Oanh
21127229 - Duong Truong Binh
21127616 - Le Phuoc Quang Huy

Lecturer: Le Ngoc Thanh
Nguyen Ngoc Thao

Teaching Assistant: Do Trong Le
Bui Huynh Trung Nam

Class: 21KHDL

Table of Contents

Task Progression

No. of Task	% Completed
1	100%
2	100%
3	100%
4.1	0%
4.2	0%

Team's Result

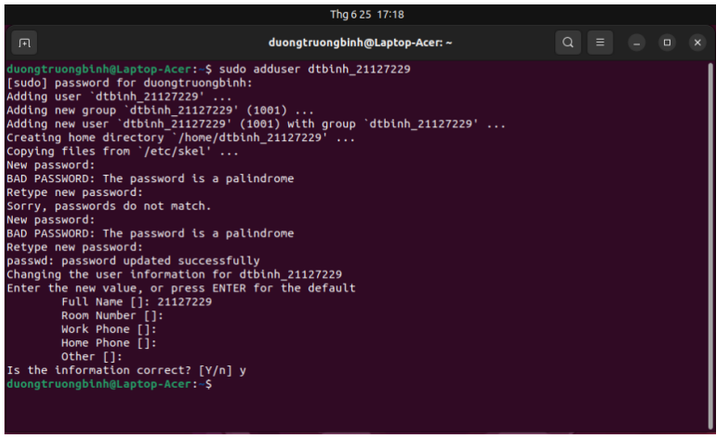
ID	Member	Task	% Completed
21127104	Đoàn Ngọc Mai	• 1. Setting up Single-node Hadoop Cluster • 2. Paper Reading	100%
21127129	Lê Nguyễn Kiều Oanh	• 1. Setting up Single-node Hadoop Cluster • 3 Running a warm-up problem: Word Count	100%
21127229	Dương Trường Bình	• 1. Setting up Single-node Hadoop Cluster • 3 Running a warm-up problem: Word Count	100%
21127616	Lê Phước Quang Huy	• 1. Setting up Single-node Hadoop Cluster • 2. Paper Reading	100%

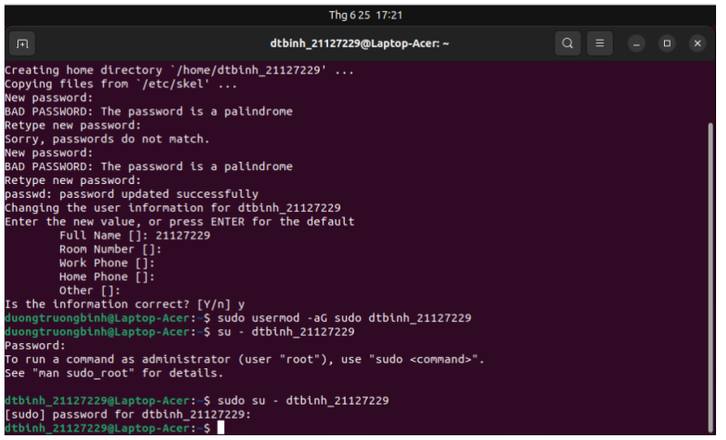
1 Lab Answer

1.1 Setting up Single-node Hadoop Cluster

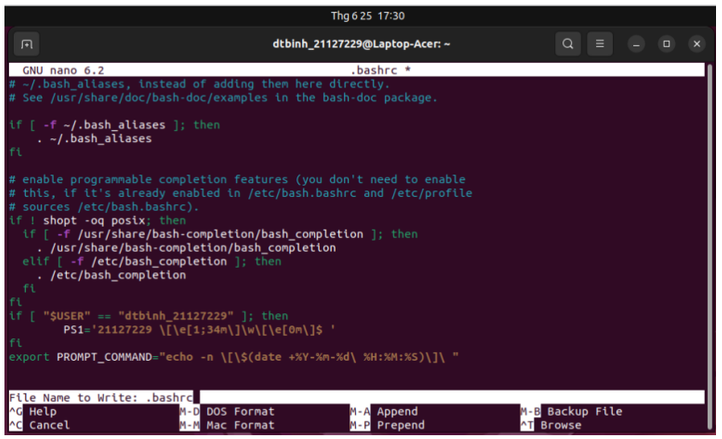
a Binh

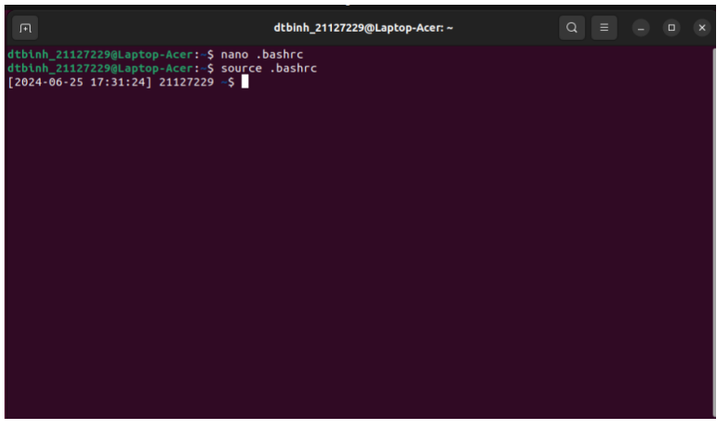
- Create a new user and add it to the *sudo* group



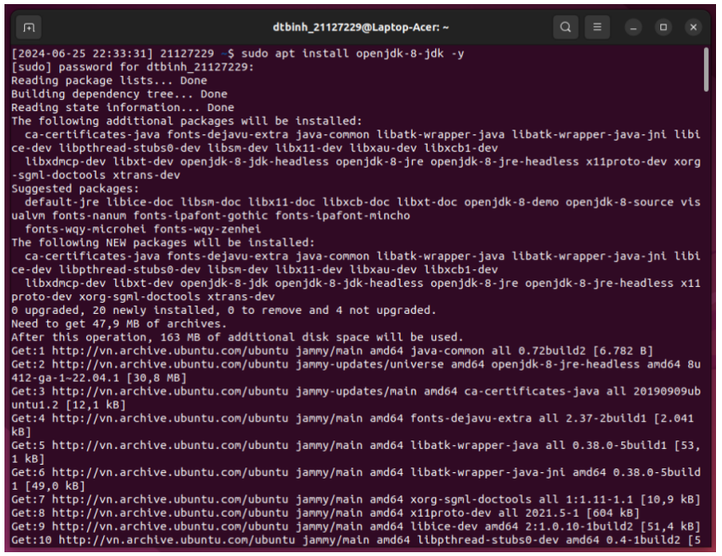


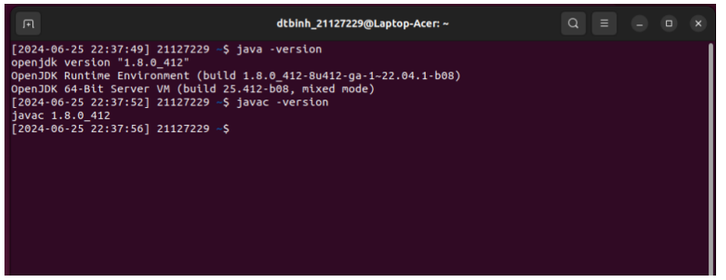
- Change the shell prompt to show student ID and timestamp by modifying *.bashrc* file



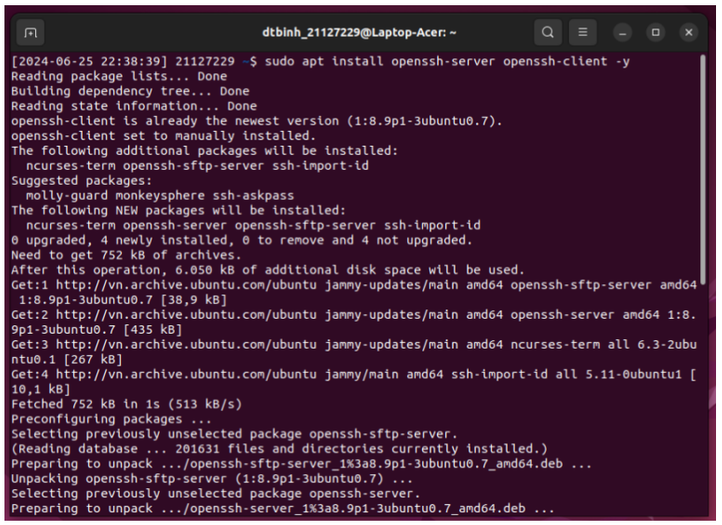


- Install Java and verify the installation.

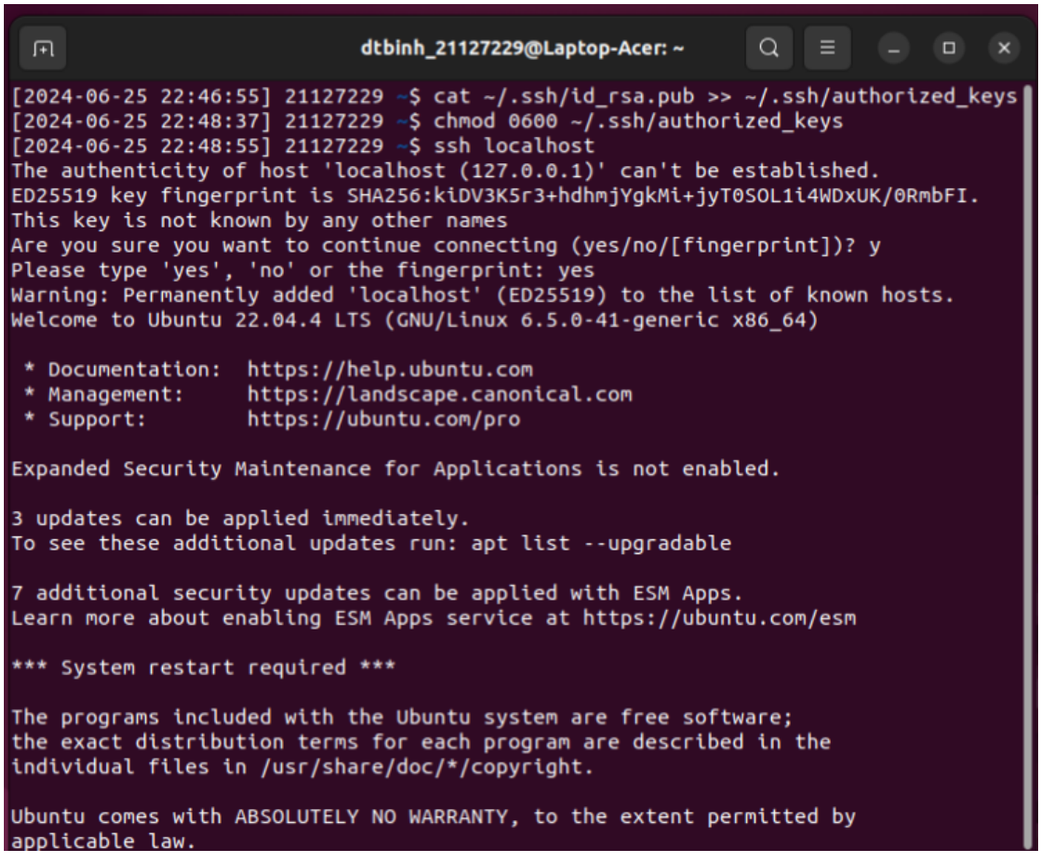




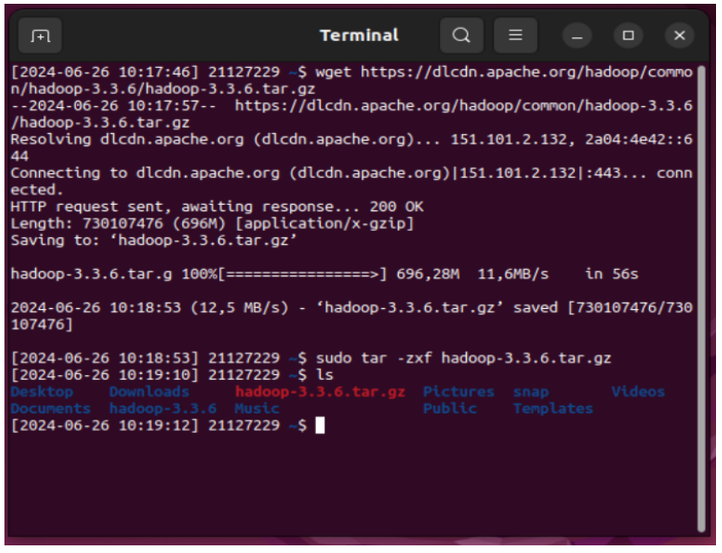
- Install the *OpenSSH* server and client. Next, generate the public and private keys.

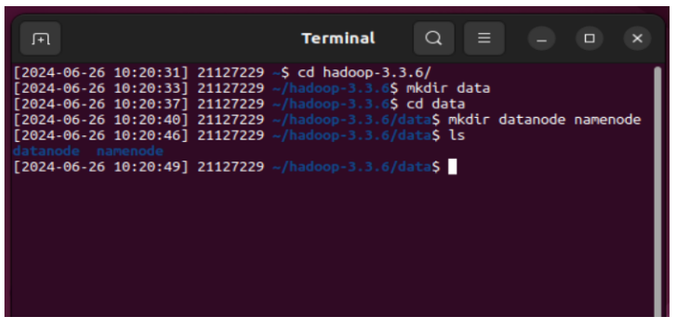


- Add the public key to *authorized_keys*. Use *chmod* to change the file permissions of *authorized_keys*. Finally, verify the SSH configuration.

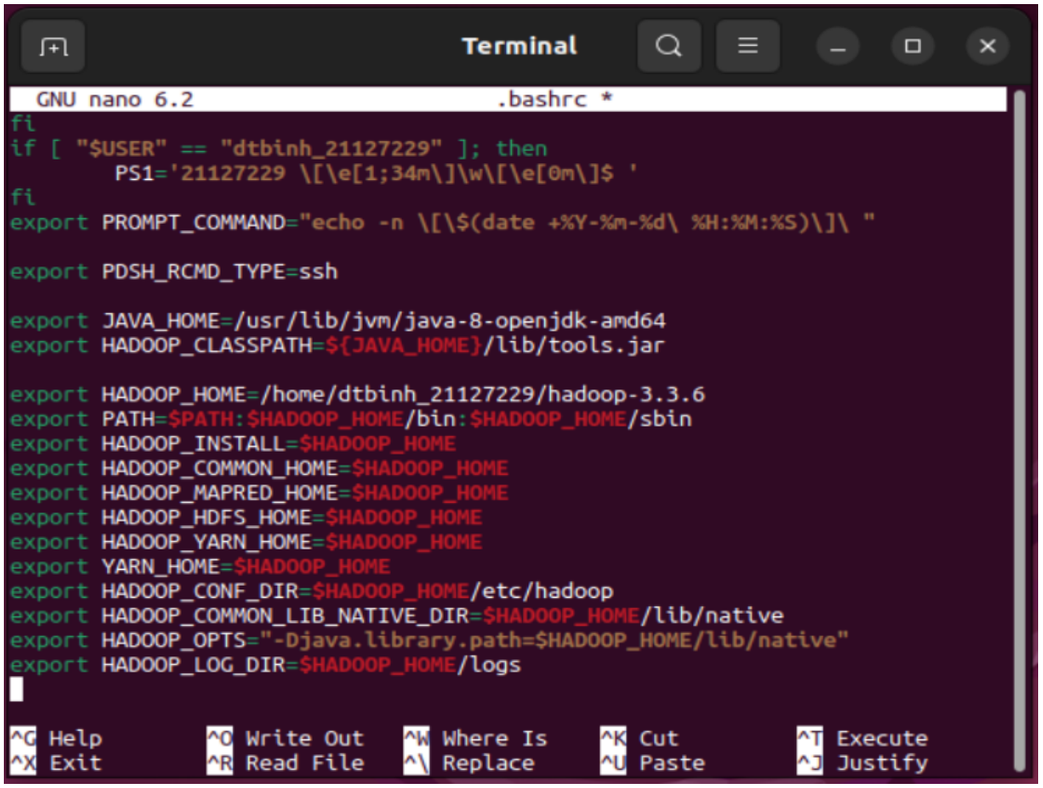


- Install and extract Hadoop version 3.3.6. Create the directories *data/datanode* and *data/namenode*.

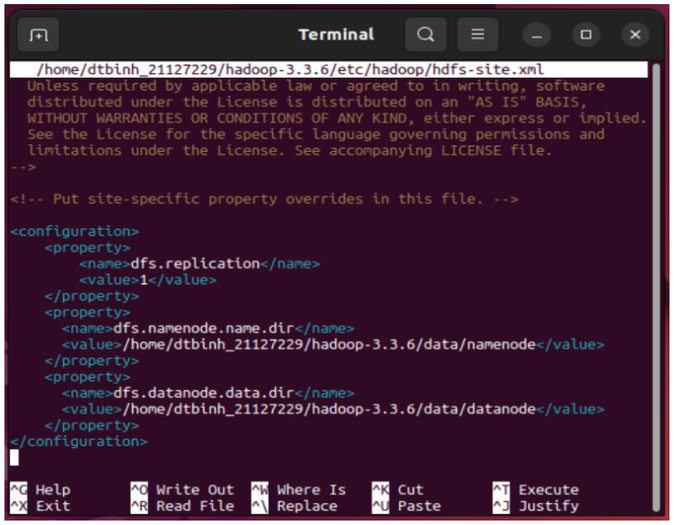


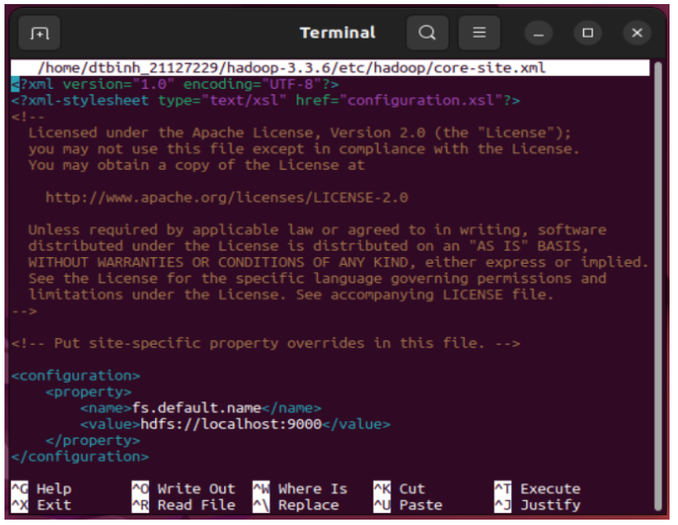


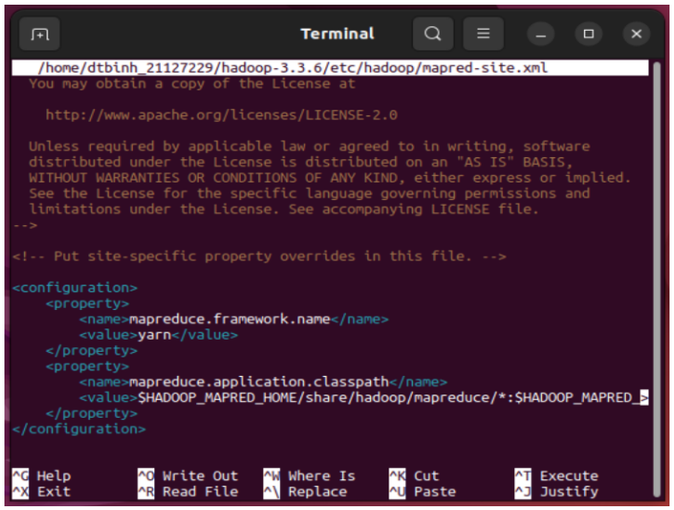
- Modify the *.bashrc* file to configure the Hadoop environment variables and use the command *source /.bashrc* to apply the changes.

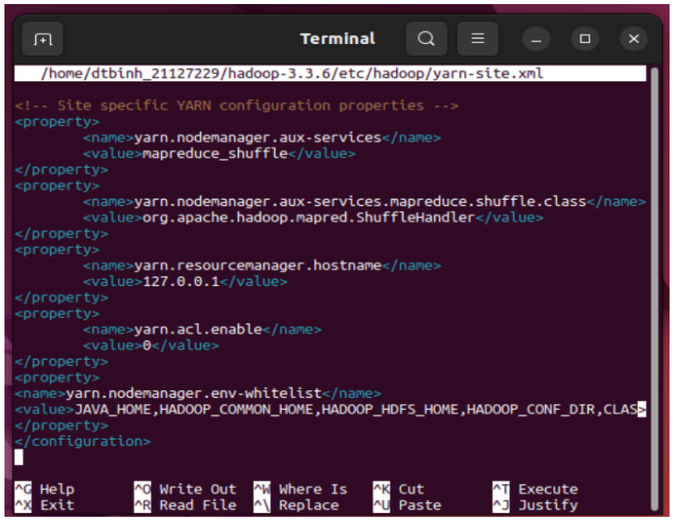


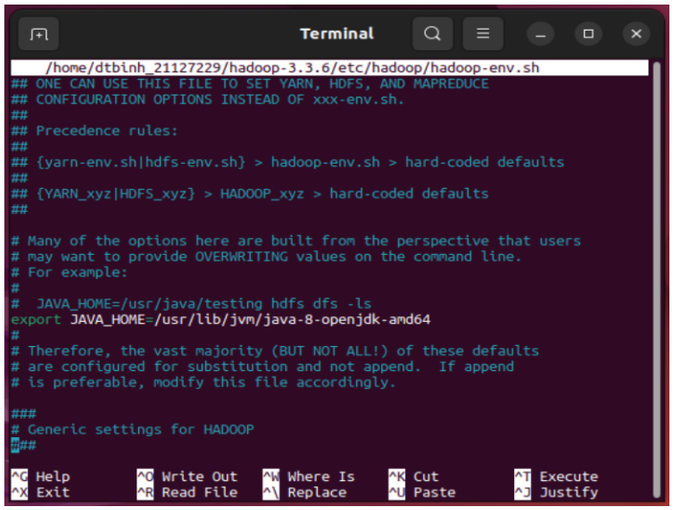
- Configure Java environment variables by modifying the files *hadoop-env.sh*, *core-site.xml*, *hdfs-site.xml*, *mapred-site.xml*, and *yarn-site.xml*.



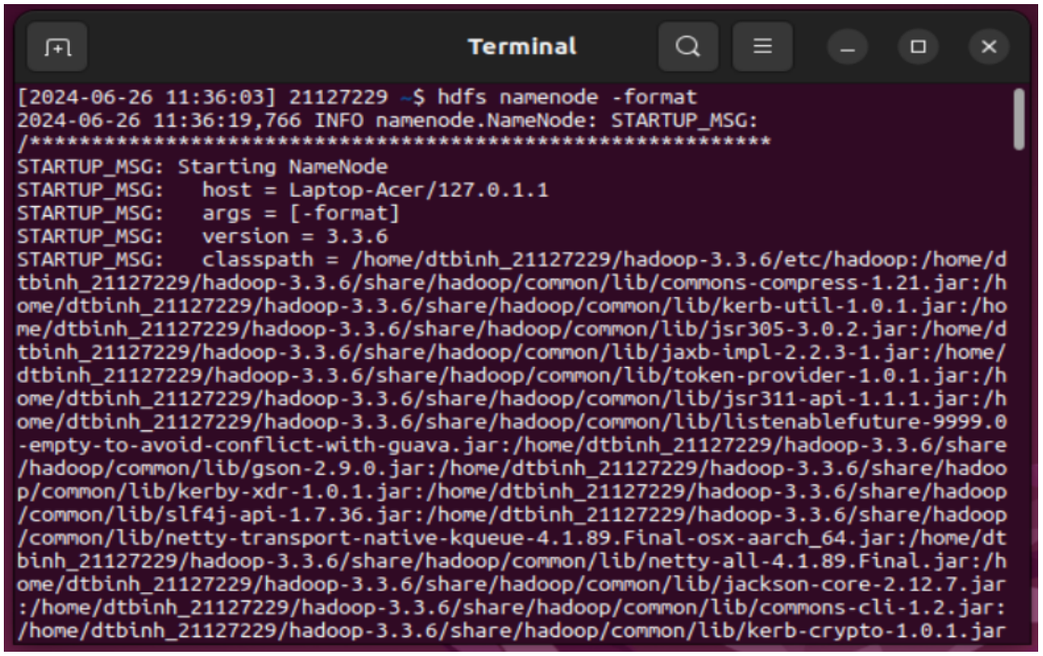




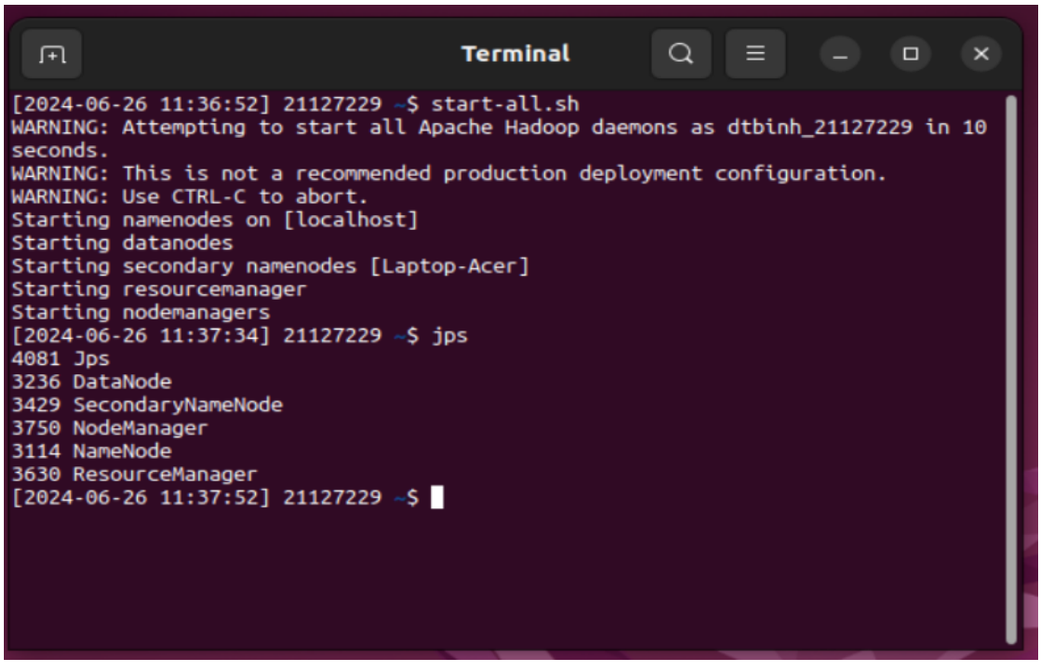




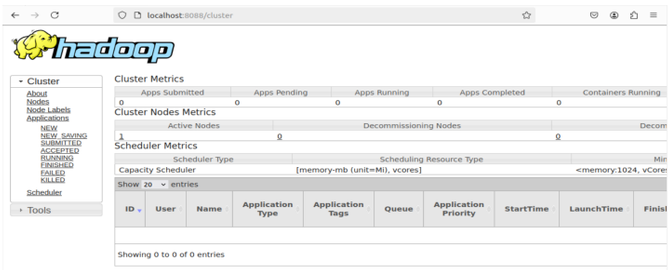
- Validate the Hadoop configuration and format the NameNode.

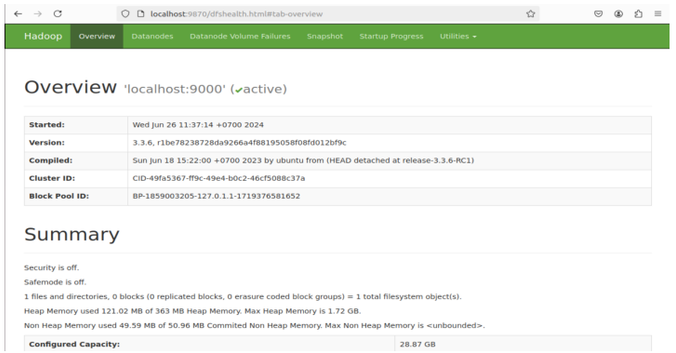


- Start the Hadoop cluster



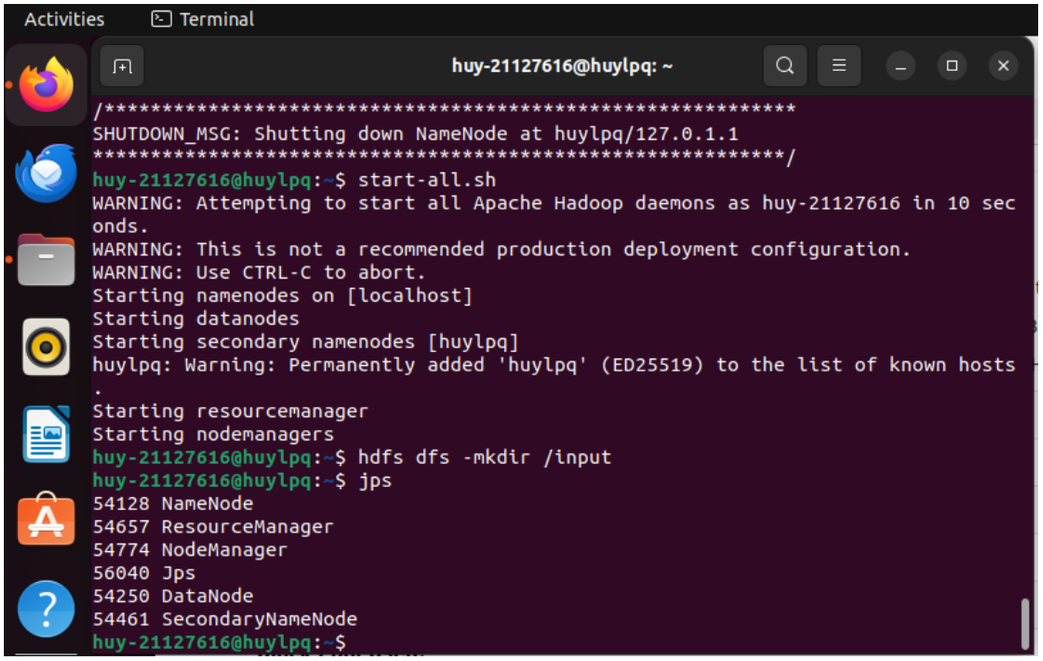
- View the result using the Hadoop Web Interface at *localhost:9870* and *localhost:8088*.



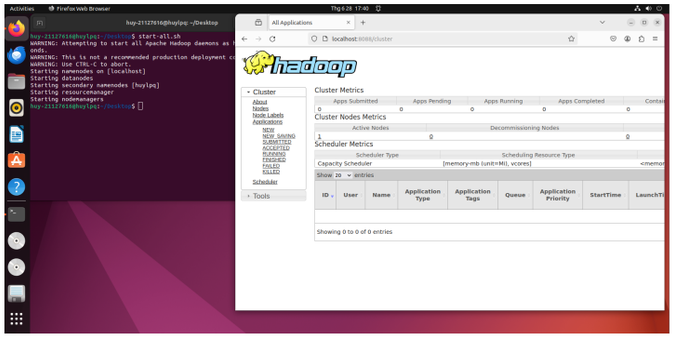


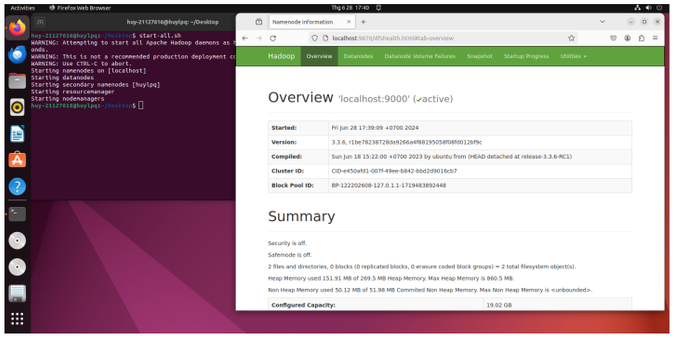
1.2 Huy

- The installation steps are the same for all group members.
- Start Hadoop Cluster



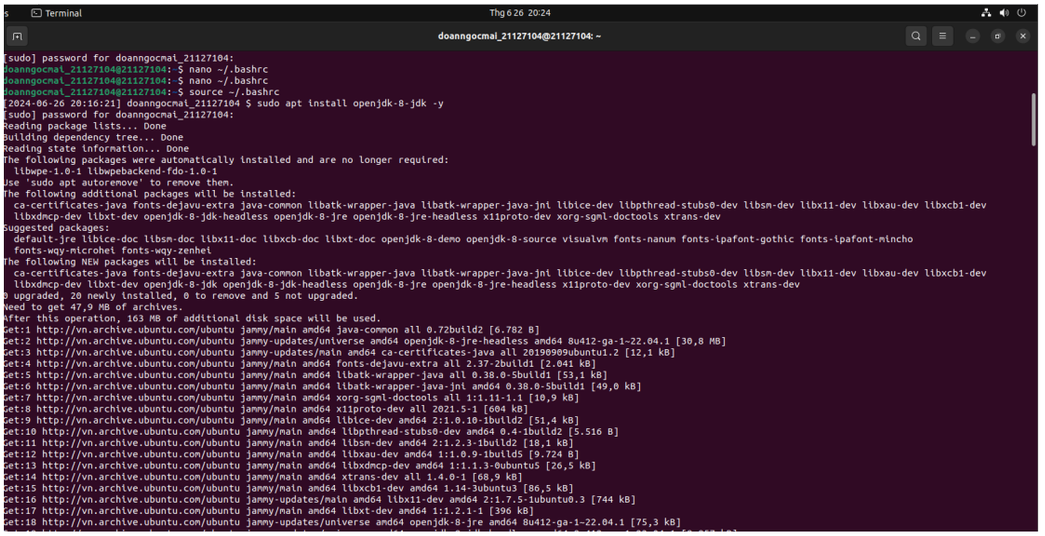
- Finally, checking the result using the Hadoop Web Interface at *localhost:8088* and *localhost:9870*.

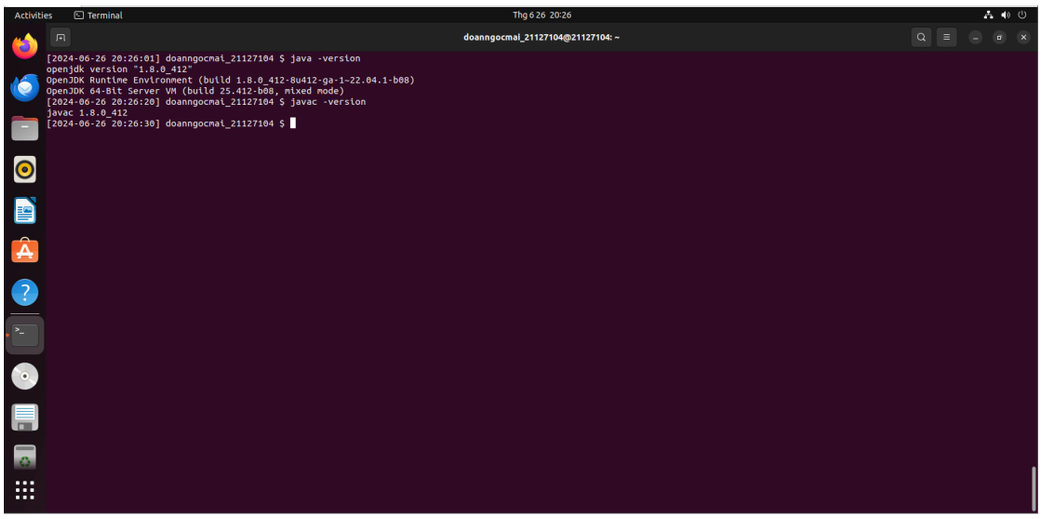




1.3 Mai

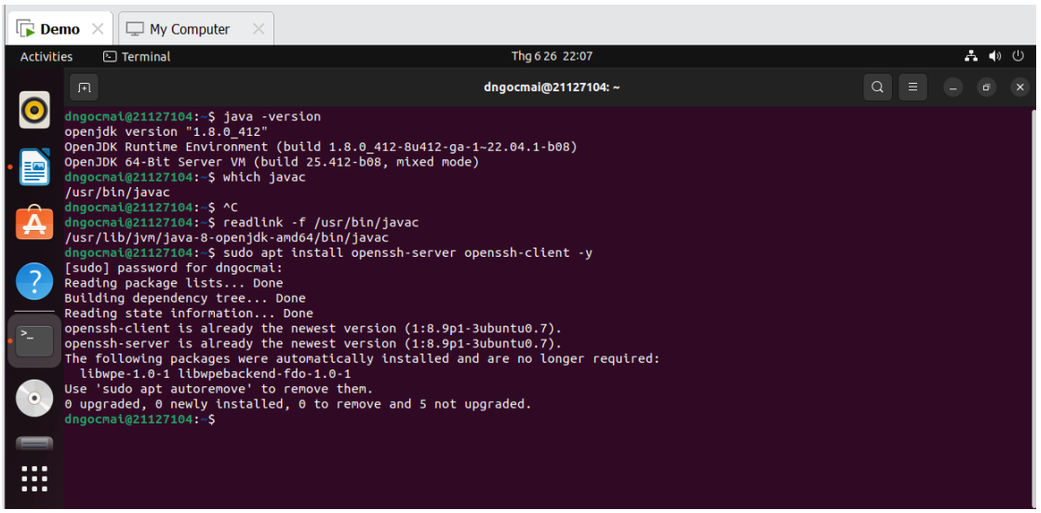
- Install Java and make sure that a suitable version of Java installed.

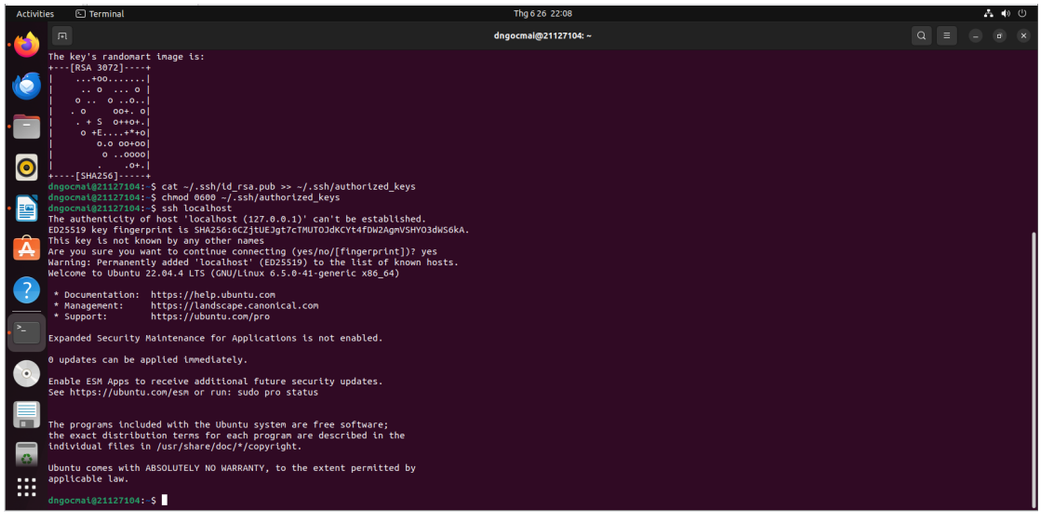




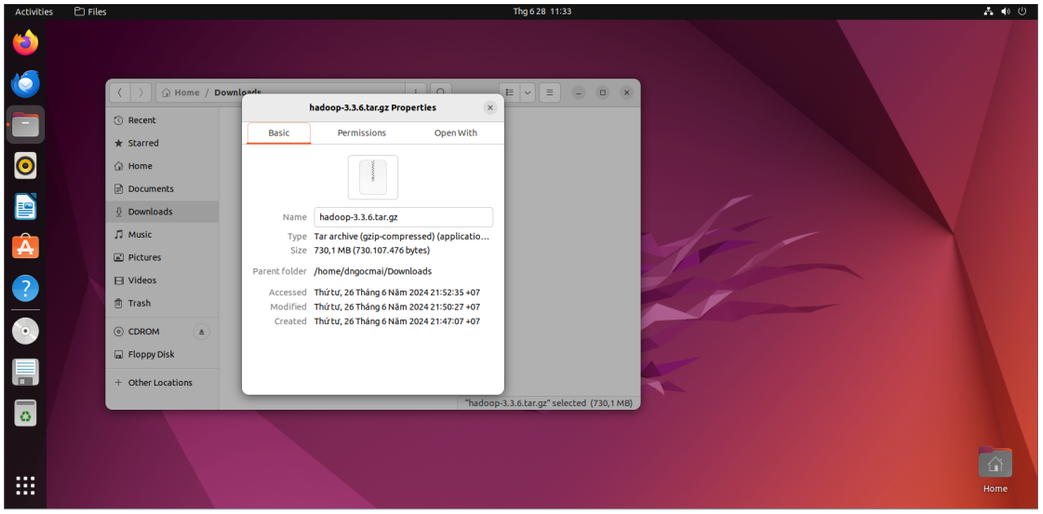
- Install OpenSSH, create an SSH key pair, add the public key to the authorized keys in the

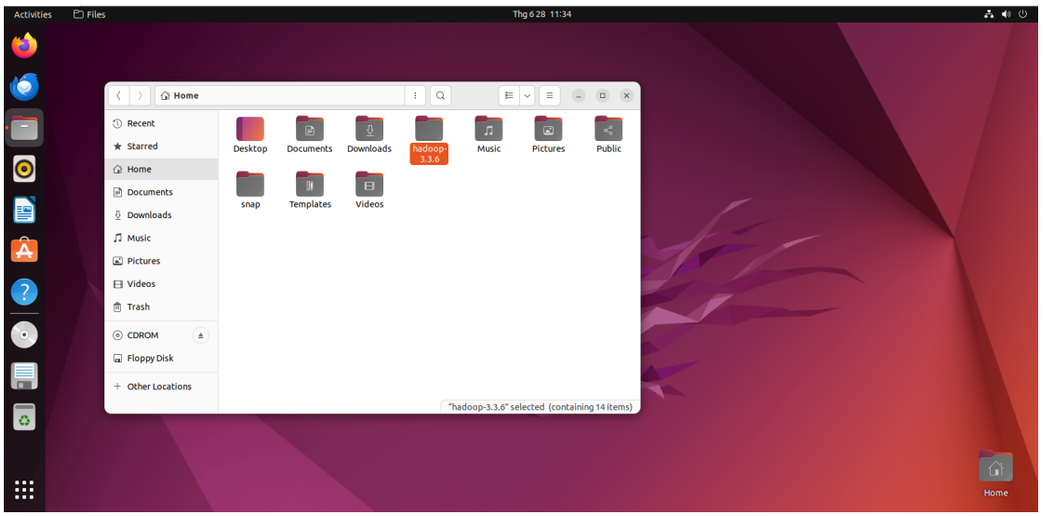
SSH directory, adjust the user permissions using the chmod command, and verify the SSH connection to localhost.



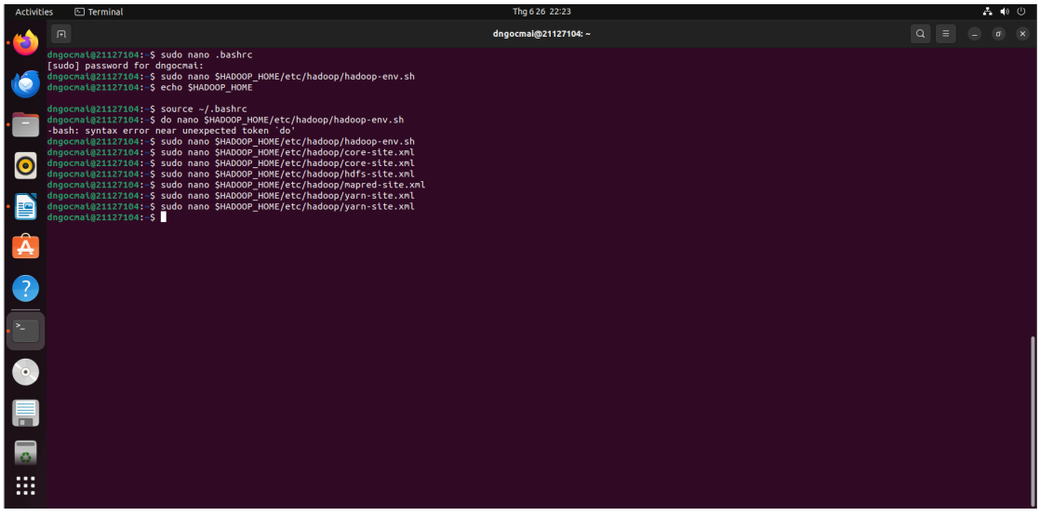


- Download Hadoop 3.3.6 using Firefox and extract it to the home directory.



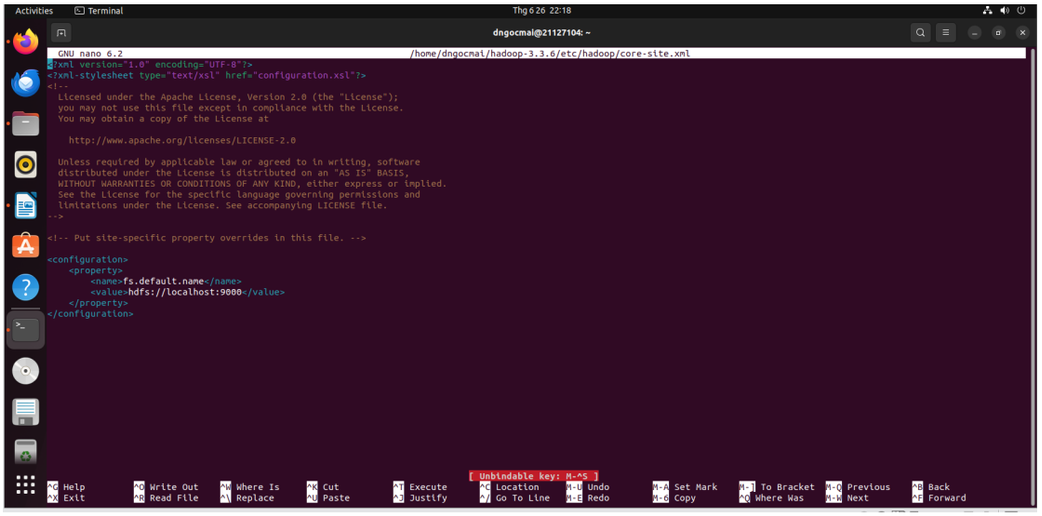


- Command to configure file *bashrc*, *core-site.xml*, *hadoop-env.sh*,*hdfs-site.xml*, *yard-site.xml*

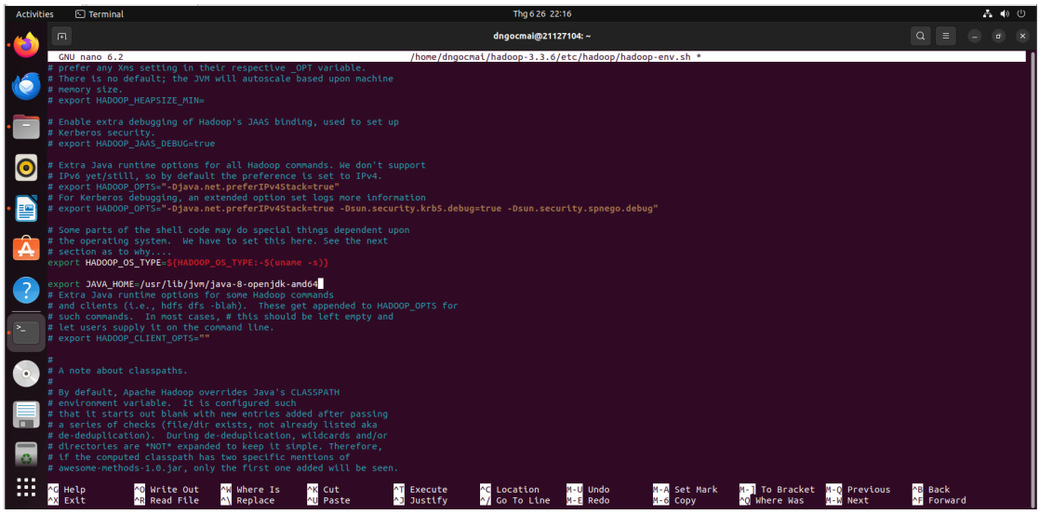


- Configure file *bashrc*

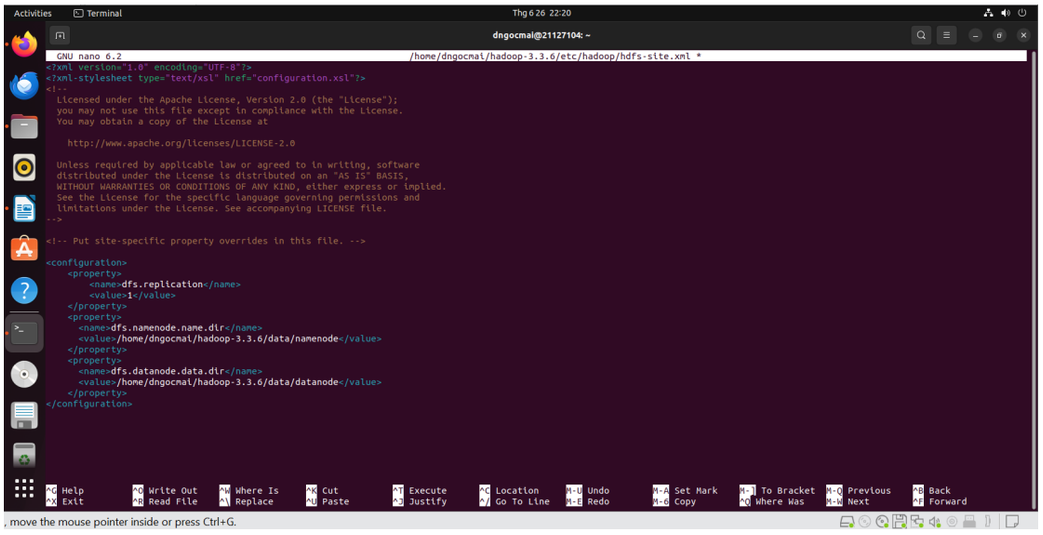
- Configure file *core-site.xml*



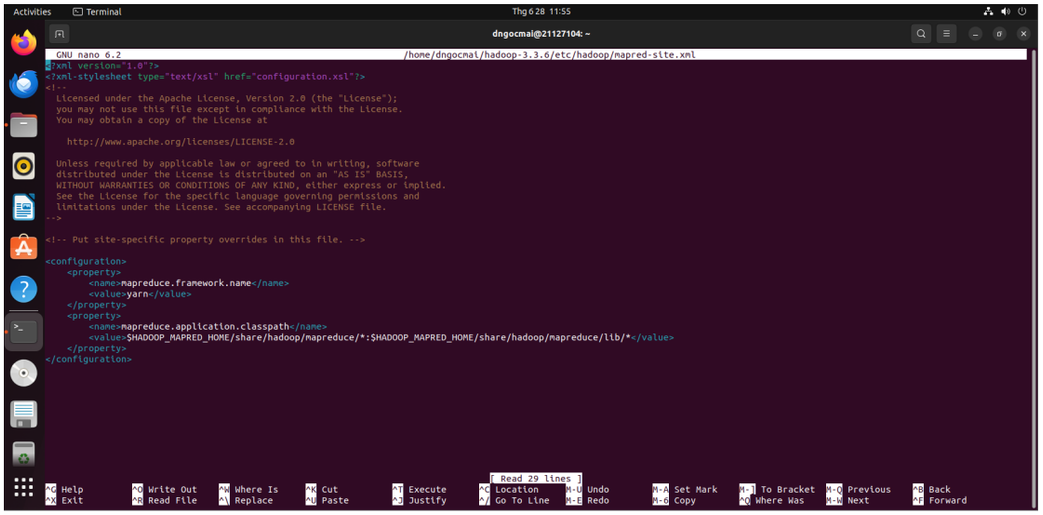
- Configure file *Hadoop-env.sh*



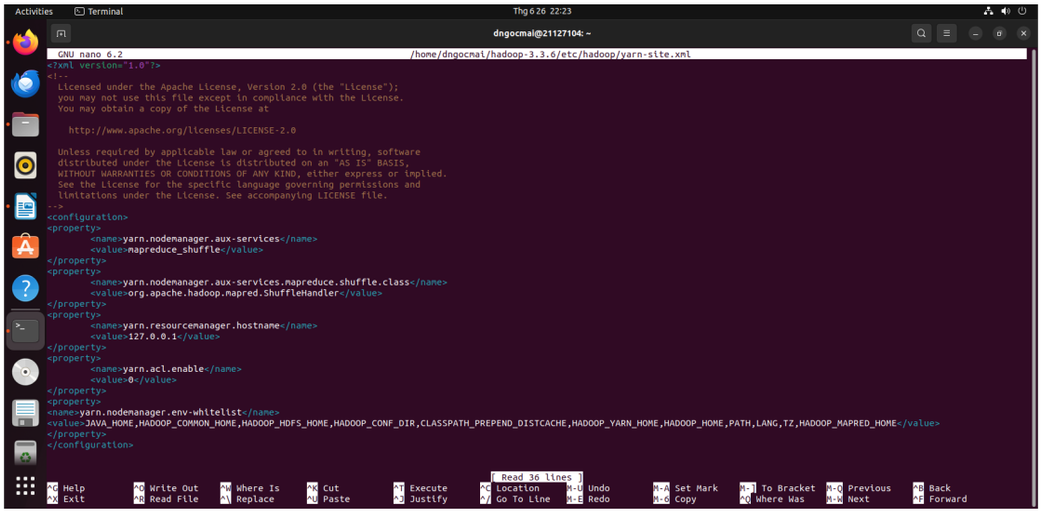
- Configure file *hdfs-site.xml*



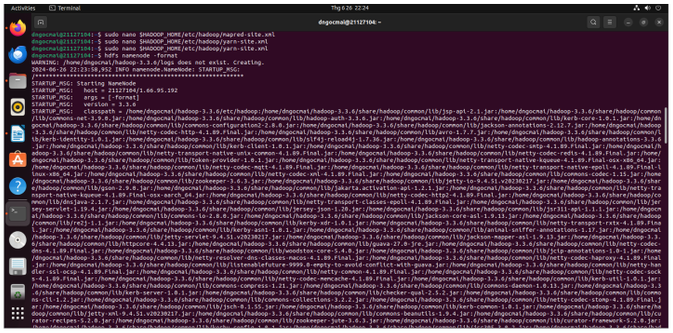
- Configure file *mapred-site.xml*

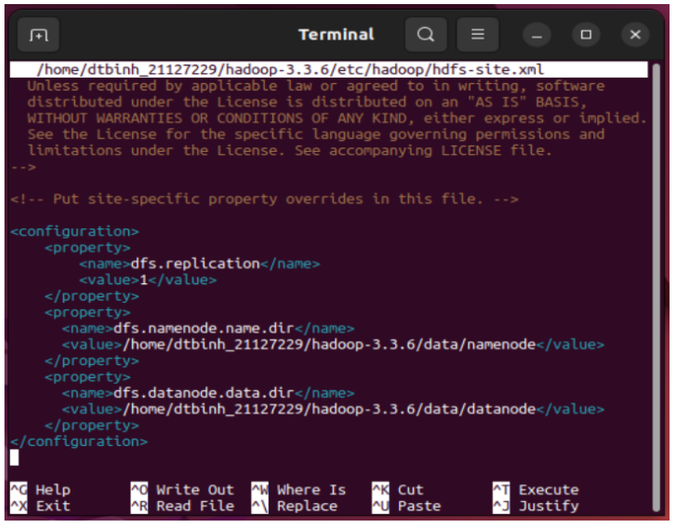


- Configure file *yard-site.xml*

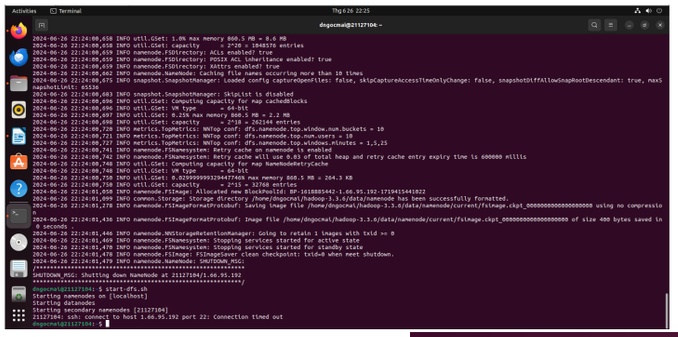


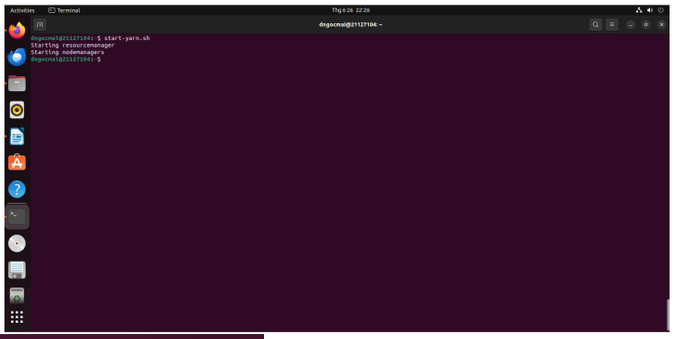
- Format HDFS Name Node

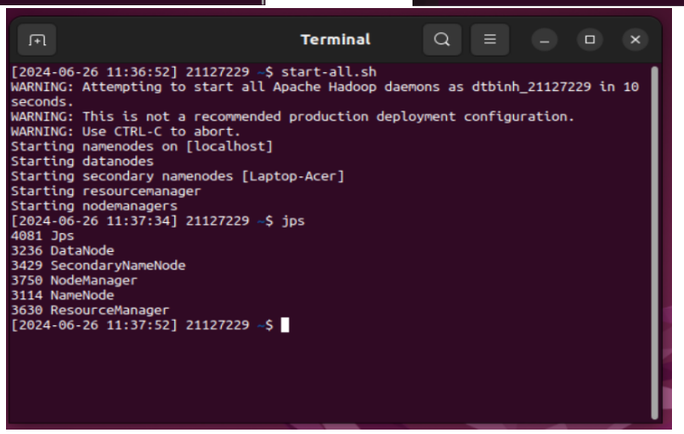




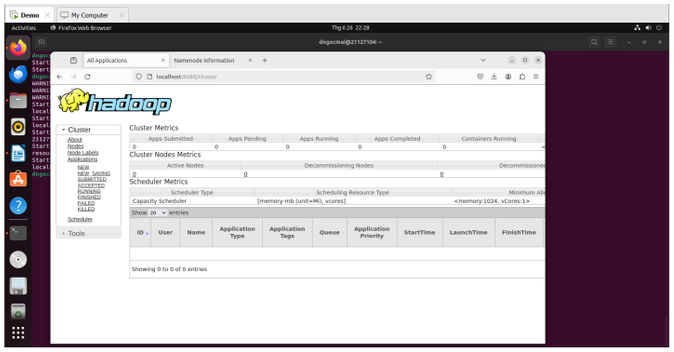
- Start Hadoop Cluster

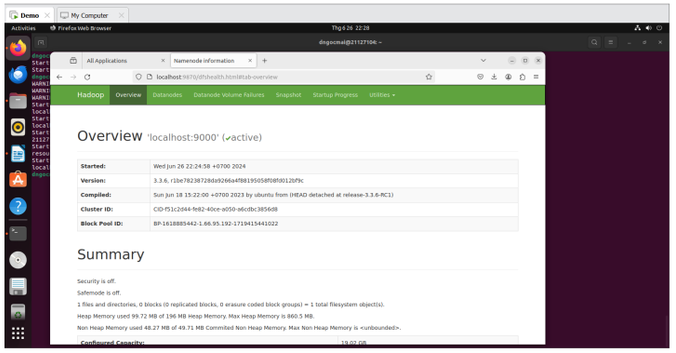




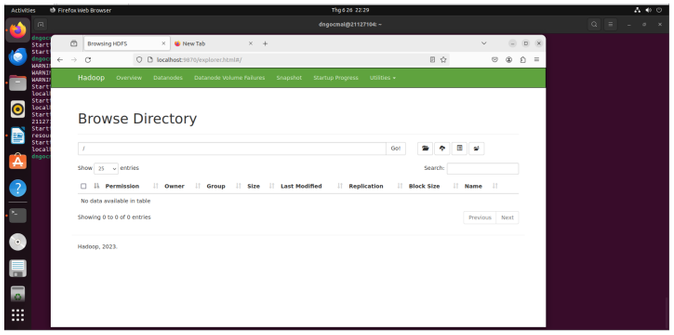


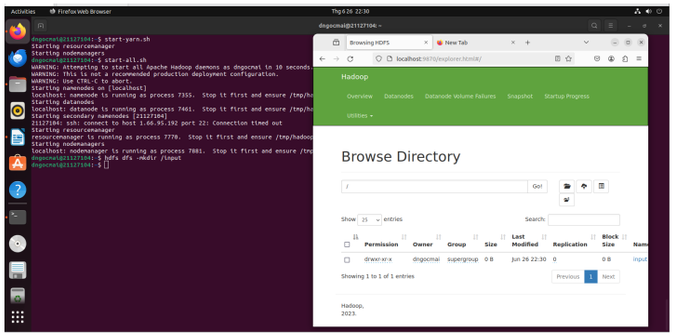
- Verifying the installation by accessing the Hadoop Web Interface at *localhost:8088* and *localhost:9870*.





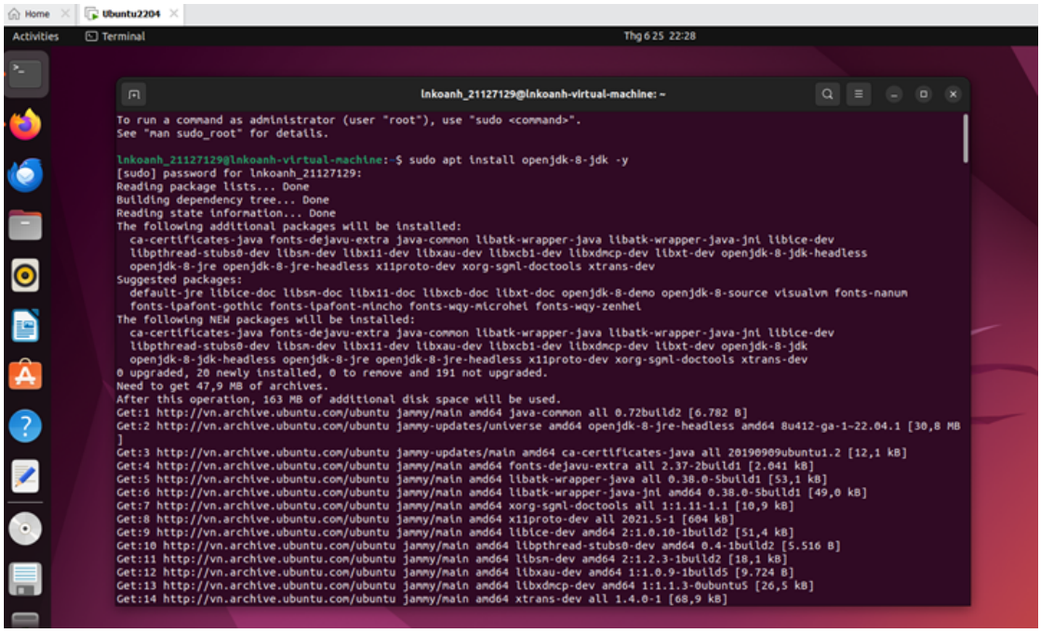
- Run a Hadoop MapReduce Job to search for patterns in HDFS and then retrieve results



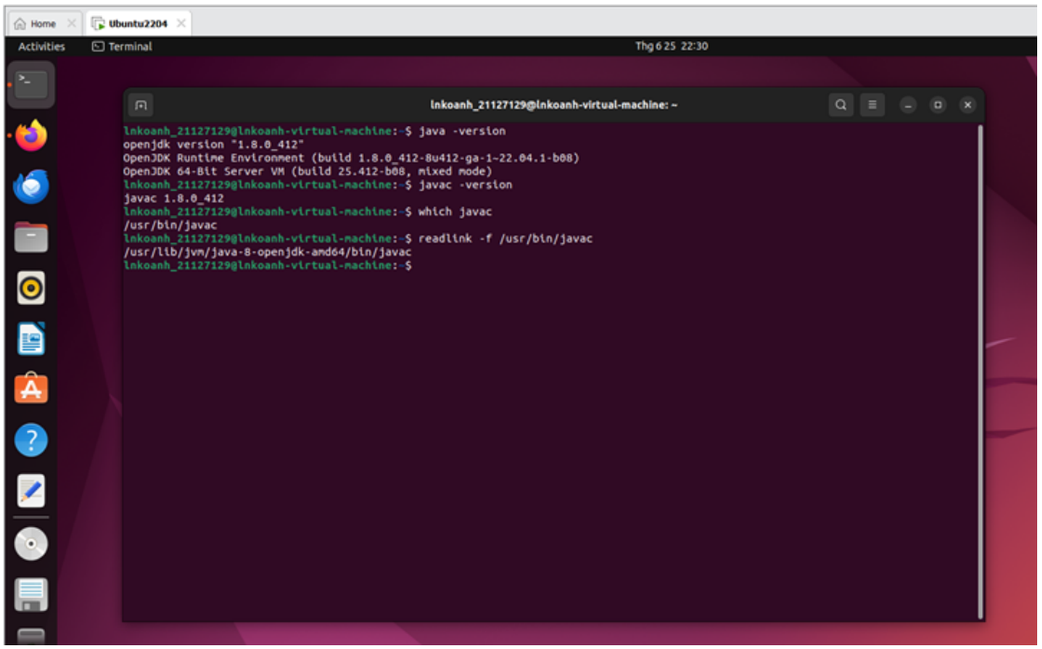


1.4 Oanh

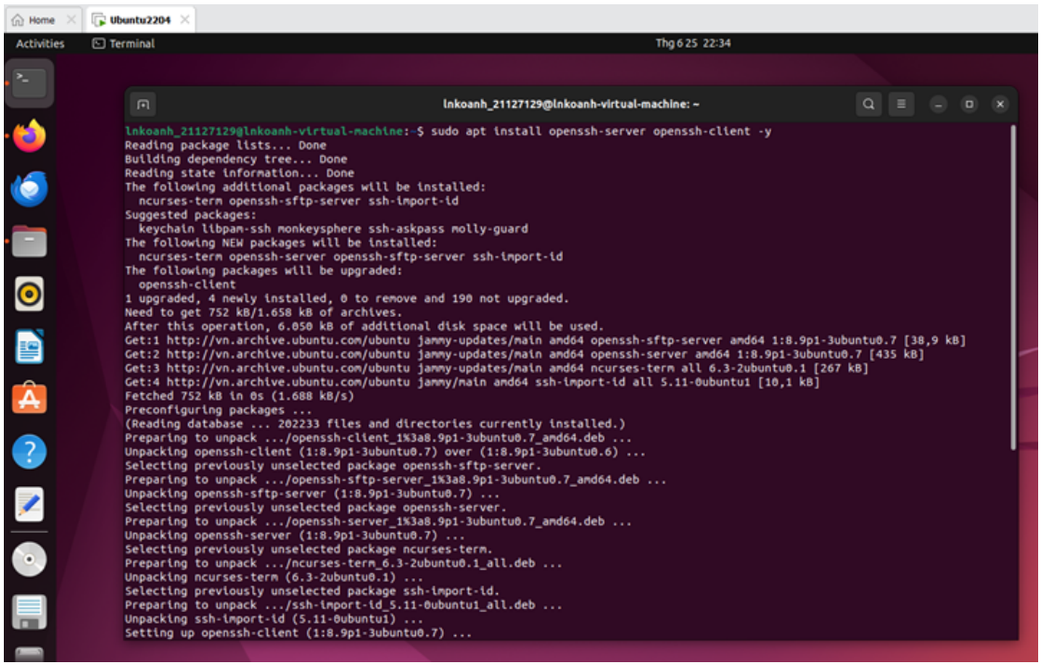
- Firstly, installing Java



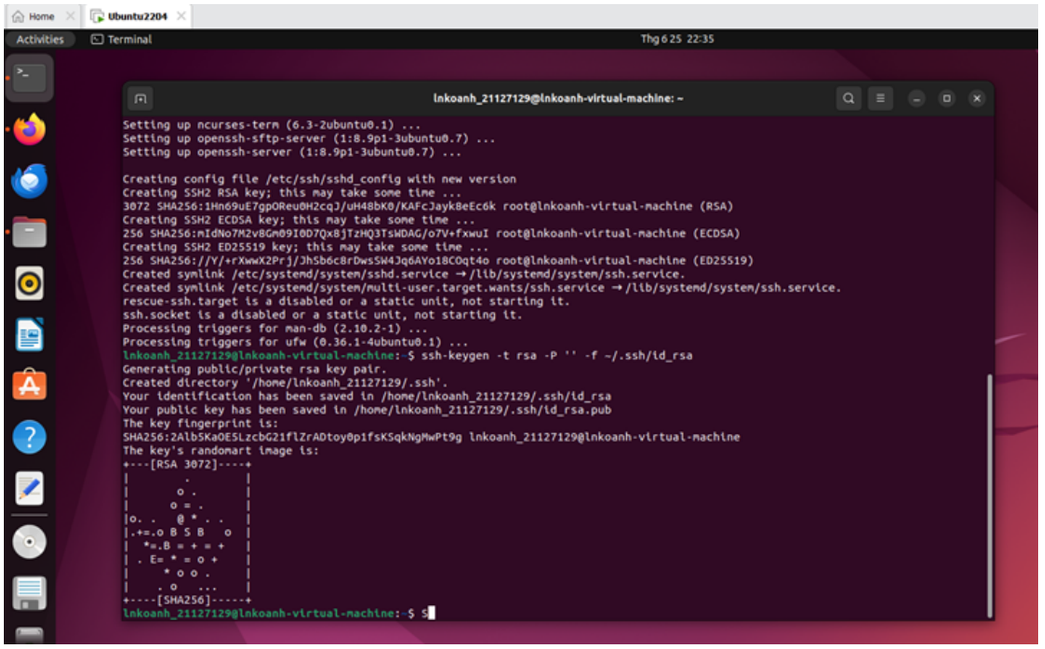
- Checking Java and Javac version



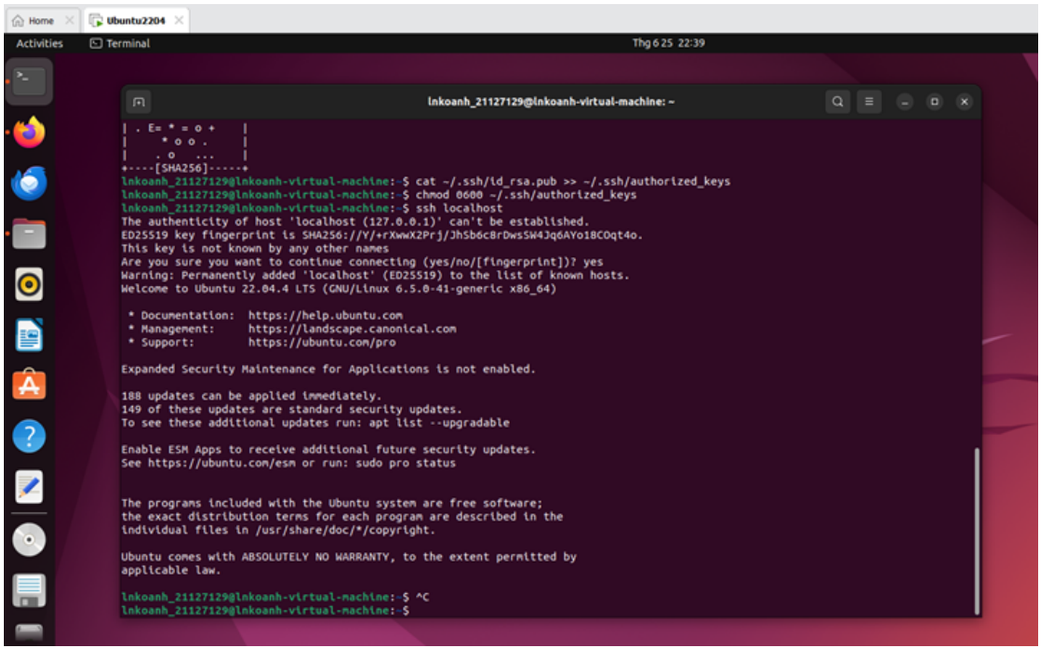
- Installing OpenSSH



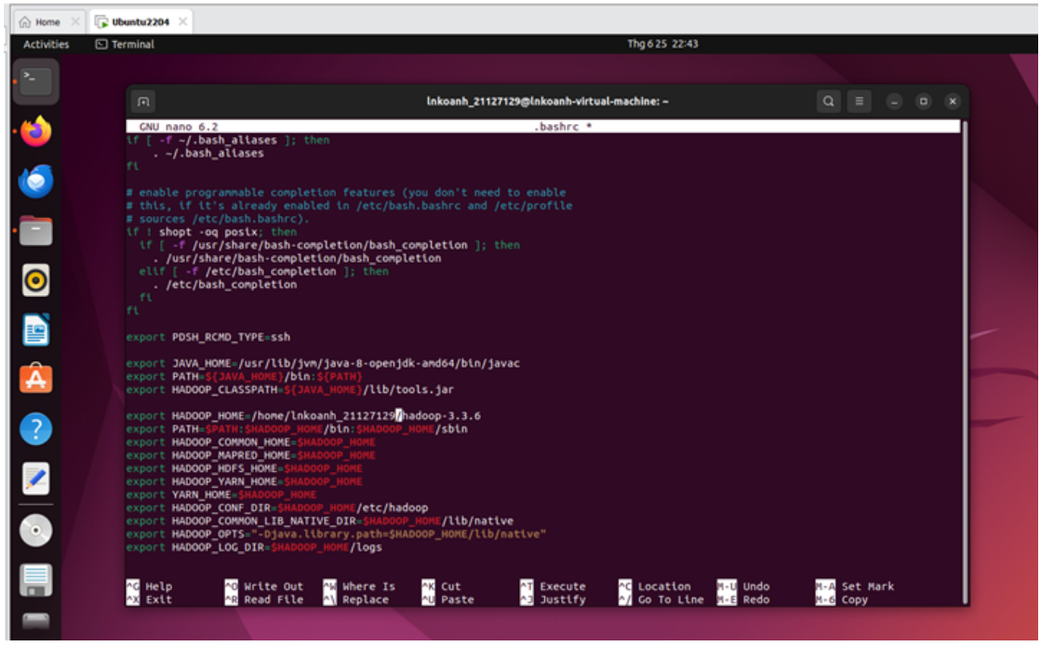
- Generating SSH Key



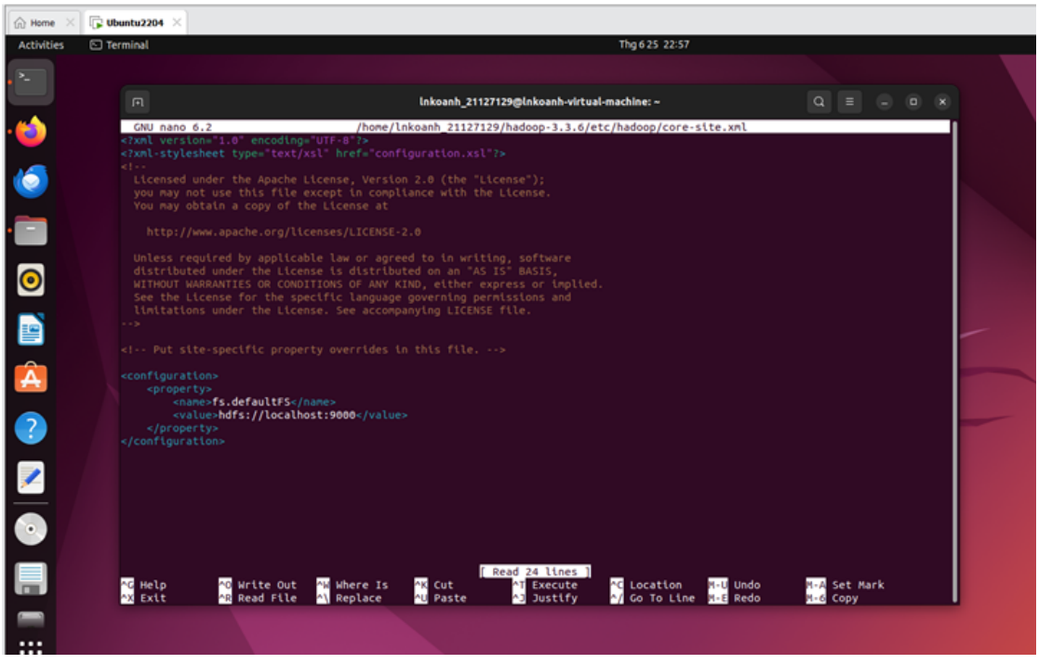
- Store public key as authorized key in SSH directory, set permission for user with chmod command and check SSH localhost



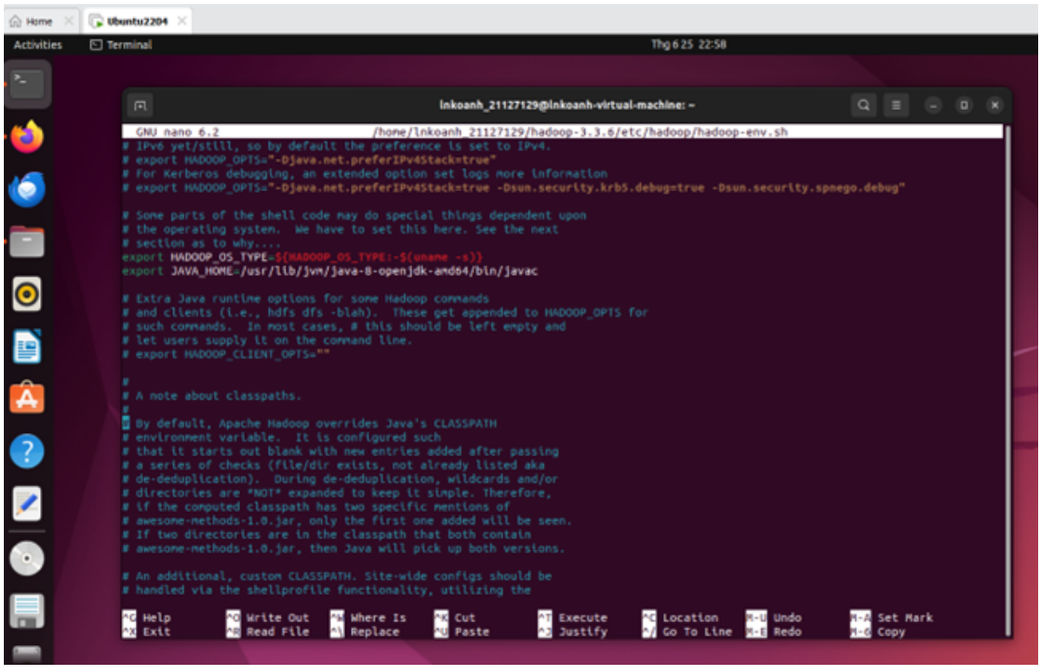
- Download Hadoop 3.3.6 and extract file at Home
- Configure file *bashrc*



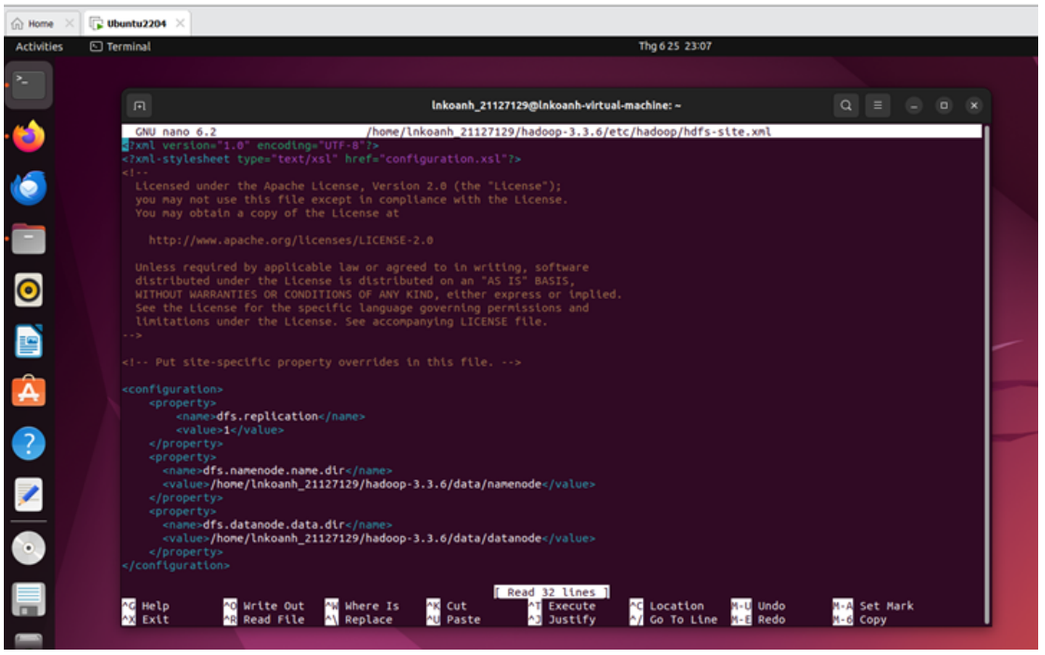
- Configure file *core-site.xml*



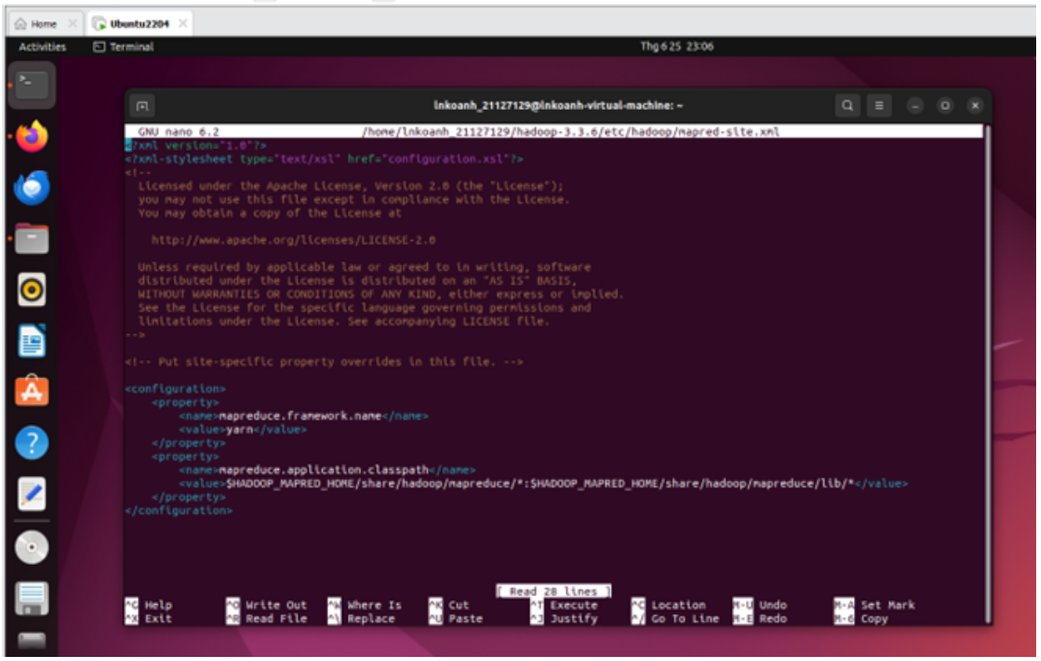
- Configure file *Hadoop-env.sh*



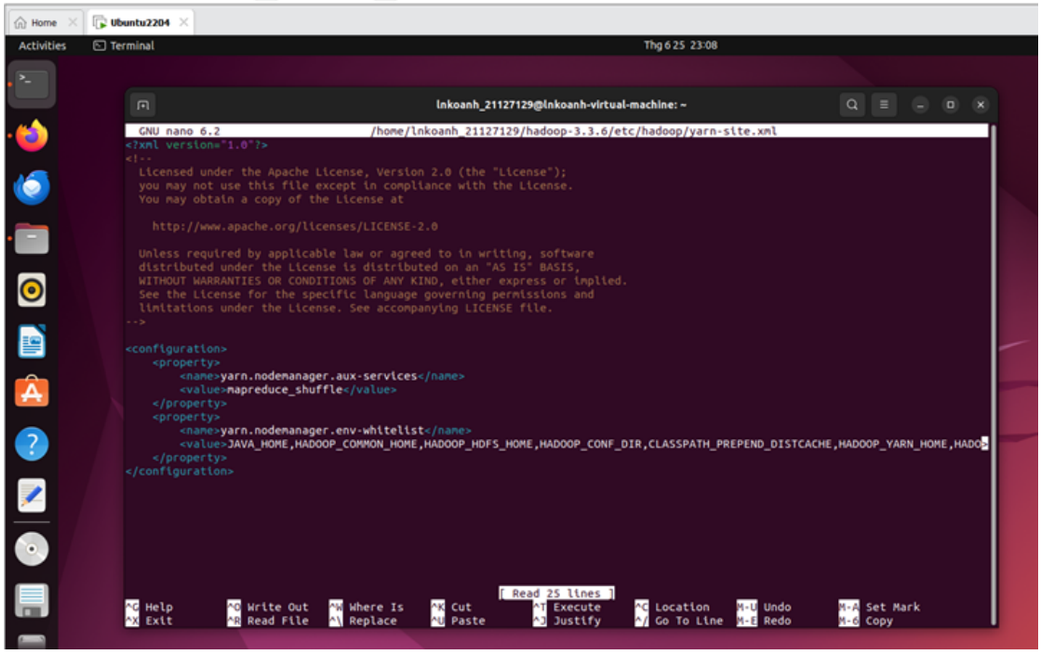
- Configure file *hdfs-site.xml*



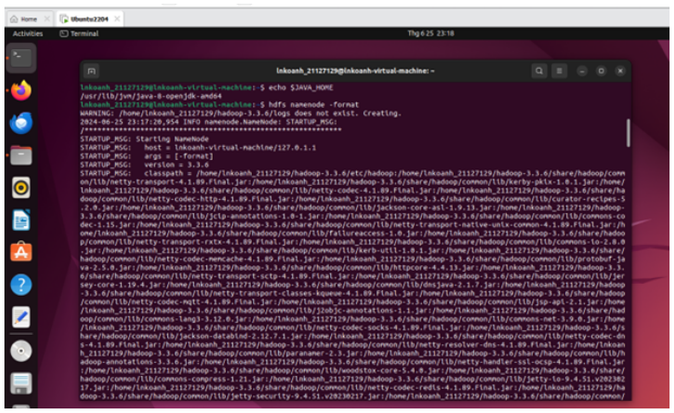
- Configure file *mapred-site.xml*

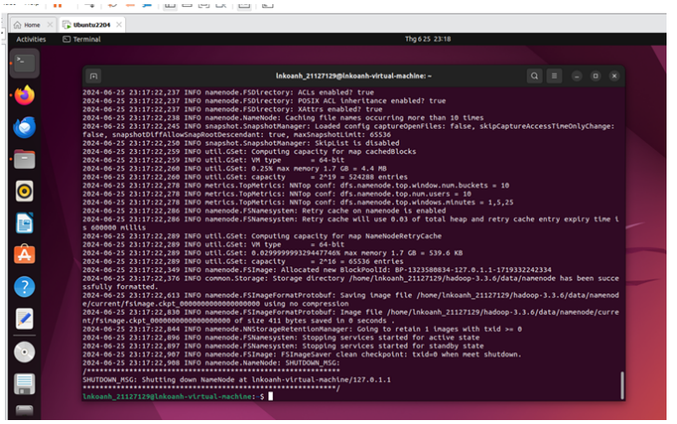


- Configure file *yard-site.xml*

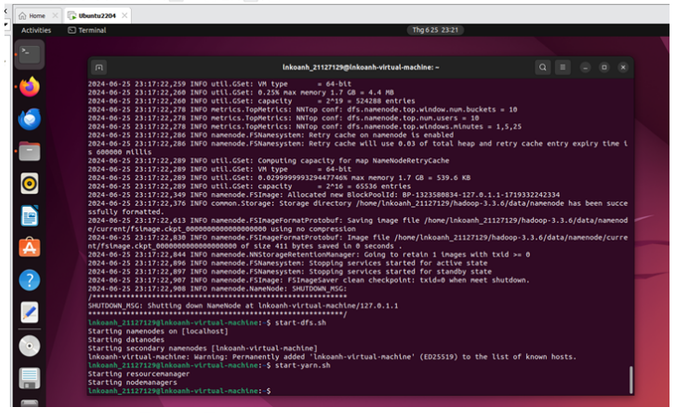


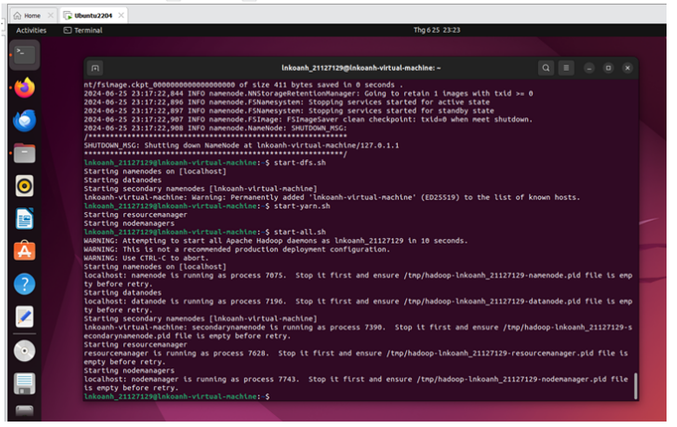
- Format HDFS Name Node

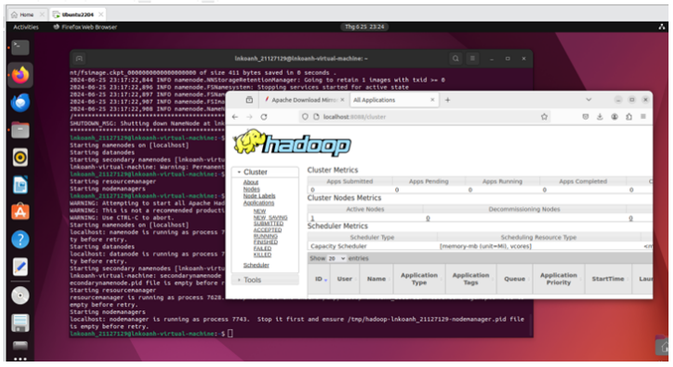


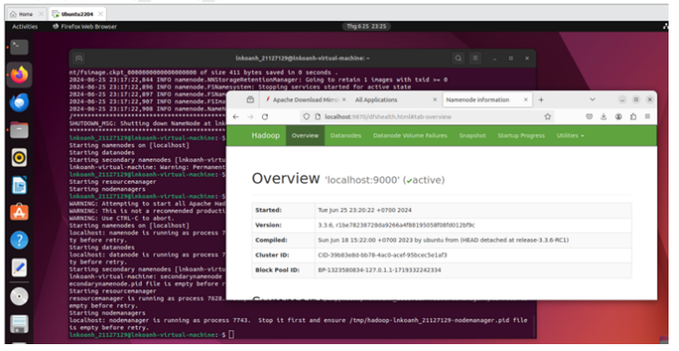


- Starting Hadoop Cluster

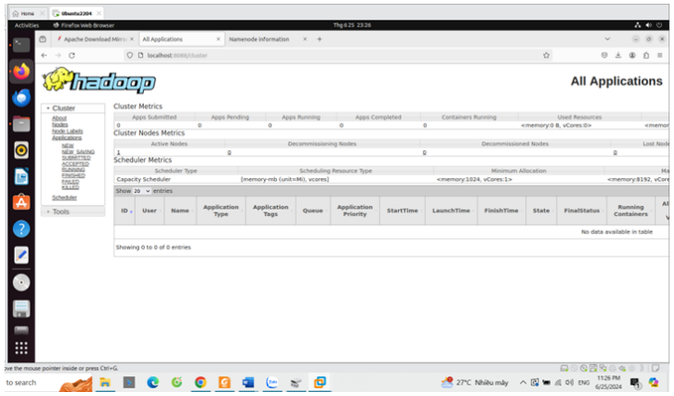


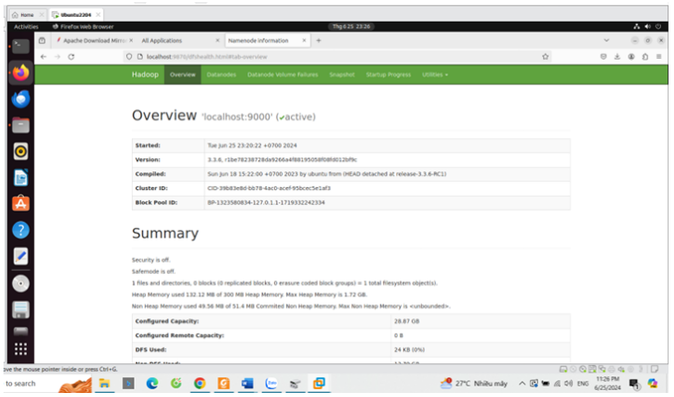






- Finally, checking the result using the Hadoop Web Interface at *localhost:8088* and *localhost:9870*.


All Applications
Cluster Metrics
Cluster Nodes Metrics
Scheduler Metrics
No data available in table


Overview 'localhost:9000' (active)
Summary

2 Paper Reading

2.1 How do the input keys-values, the intermediate keys-values, and the output keys-values relate?

In the MapReduce model, the relationship between the input, intermediate, and output keys-values is as follows:

- **Input Keys-Values:** These are the initial data pairs processed by the Map function. The user writes the Map function to take these input pairs and generate intermediate key-value pairs.
- **Intermediate Keys-Values:** The intermediate key-value pairs generated by the Map function are grouped by key. All values associated with the same intermediate key are then passed to the Reduce function.
- **Output Keys-Values:** The intermediate key-value pairs are then passed to the Reduce phase, where the reduce function processes them to produce the final output key-value pairs. The output keys are typically aggregated or summarized versions of the intermediate keys, and the output values are the final processed data corresponding to these output keys.

2.2 How does MapReduce deal with node failures?

MapReduce deals with node failures through several mechanisms to ensure the reliability and efficiency of its operations.

- **Worker Failure**: The master node periodically pings each worker node to check its status. If a worker node fails to respond within a certain timeframe, it is marked as failed. All map tasks that were completed by the failed worker are reset to their initial idle state, making them eligible for reassignment to other workers. Similarly, any map or reduce tasks in progress on the failed worker are also reset and rescheduled. Completed map tasks must be re-executed because their output is stored on the local disk of the failed machine and becomes inaccessible. In contrast, completed reduce tasks do not need to be re-executed as their output is stored in a global file system.
- **Master Failure**: To handle potential master node failures, the master can periodically save checkpoints of its state. If the master node fails, a new master can be restarted from the last checkpointed state. However, given the low likelihood of master node failure and the

complexity involved, the current implementation of MapReduce opts to abort the computation if the master fails. Clients can detect this condition and choose to retry the MapReduce operation if needed.

- **Backup Tasks**: To mitigate the impact of stragglers—tasks that take unusually long to complete—the master schedules backup executions of the remaining in-progress tasks when the operation is near completion. The task is marked as complete when either the primary or the backup execution finishes. This approach typically increases computational resources by only a small percentage and significantly reduces the time required to complete large MapReduce operations.
- **Semantics in the Presence of Failures**: The system ensures that deterministic map and reduce operations yield the same output as a sequential execution would, even in the presence of failures. This is achieved through atomic commits of map and reduce task outputs.
- **Skipping Bad Records**: MapReduce provides an optional mode of execution to skip records that cause deterministic crashes, allowing the job to progress despite the presence of problematic records. This feature is useful in scenarios where ignoring a few records is acceptable, such as statistical analysis on large datasets.

2.3 What is the meaning and implication of locality? What does it use?

Meaning of Locality: Locality refers to the strategic scheduling of map tasks to run on the machines that already hold the data to be processed, or at least on machines that are close to these data-holding machines. This approach leverages the fact that input data is stored on local disks across a cluster, with each file divided into 64 MB blocks and replicated on multiple machines.

Implication of Locality: The implication of this strategy is a significant reduction in the consumption of network bandwidth. By reading data locally, rather than transferring it over the network, MapReduce operations can run more efficiently, particularly when dealing with large datasets spread across many workers in a cluster. Locality thereby helps in conserving network resources, reducing potential bottlenecks, and improving overall performance of the MapReduce jobs.

Locality uses the structure of the Google File System (GFS) for data storage and replication, and the intelligent scheduling capabilities of the MapReduce master, which takes into account the location of data replicas to optimize task placement.

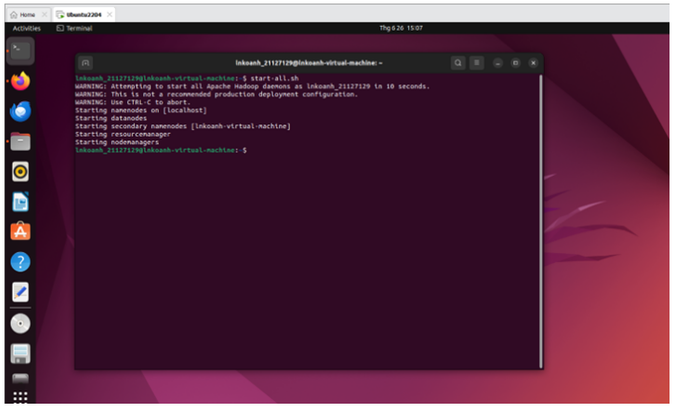
2.4 Which problem is addressed by introducing a combiner function to the MapReduce model?

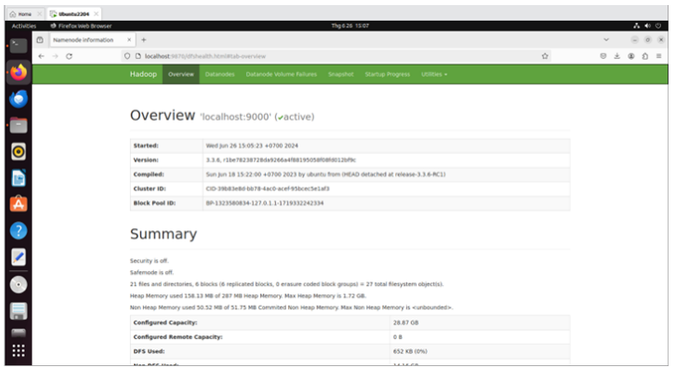
The problem addressed by introducing a combiner function to the MapReduce model is the excessive network traffic caused by the large volume of intermediate data generated during the map phase, particularly when there is significant repetition in the intermediate keys. The combiner function performs partial merging of this data locally on each machine that performs a map task, thereby reducing the amount of data that needs to be transferred over the network. This approach alleviates network congestion and improves the overall performance of MapReduce operations.

3 Running a warm-up problem: Word Count

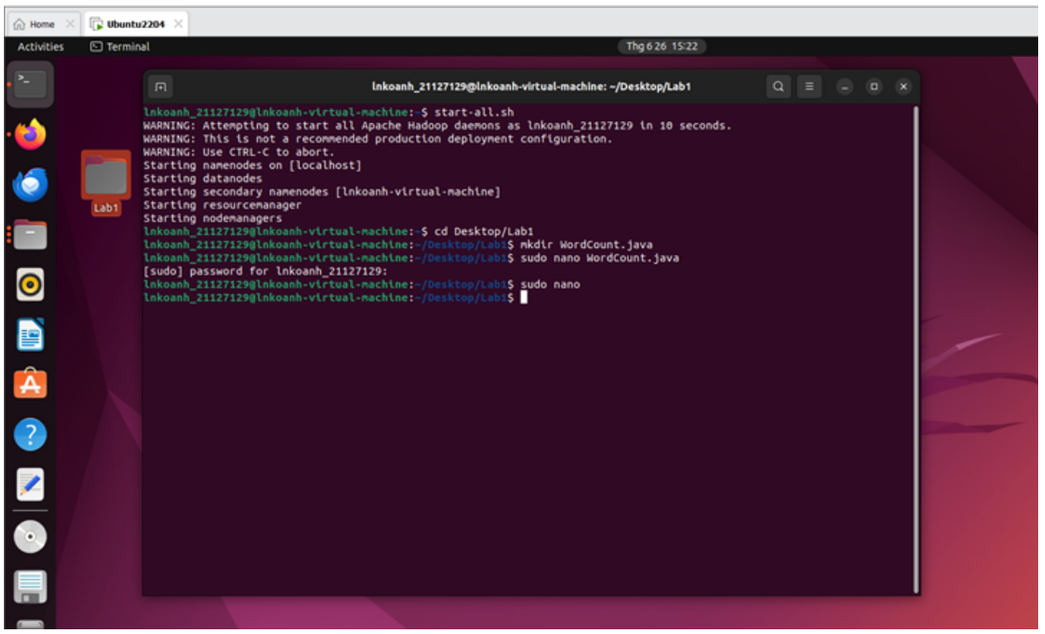
3.1 Word Count

- Firstly, we need to start all the Apache Hadoop Daemons.

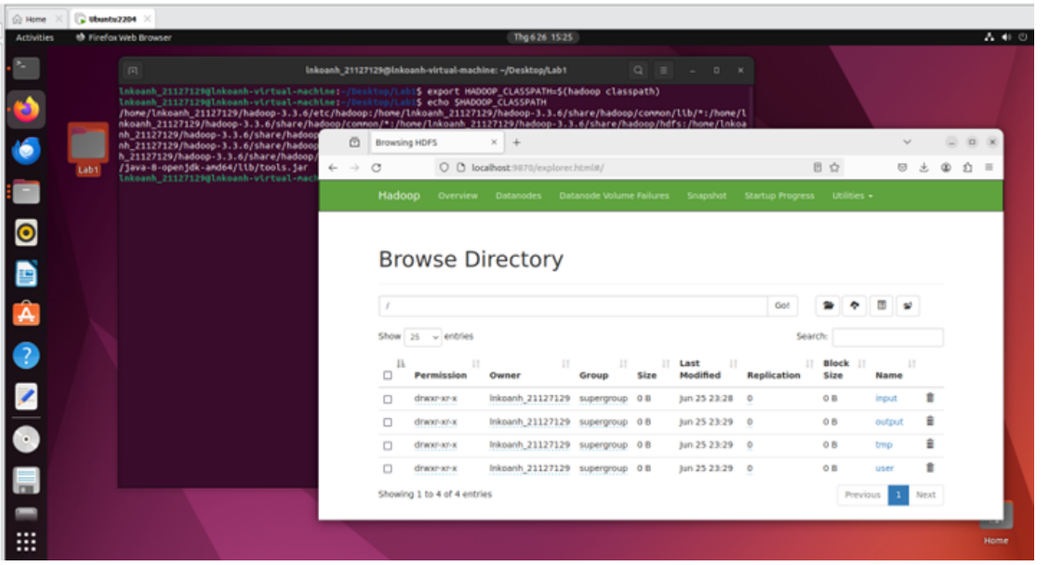




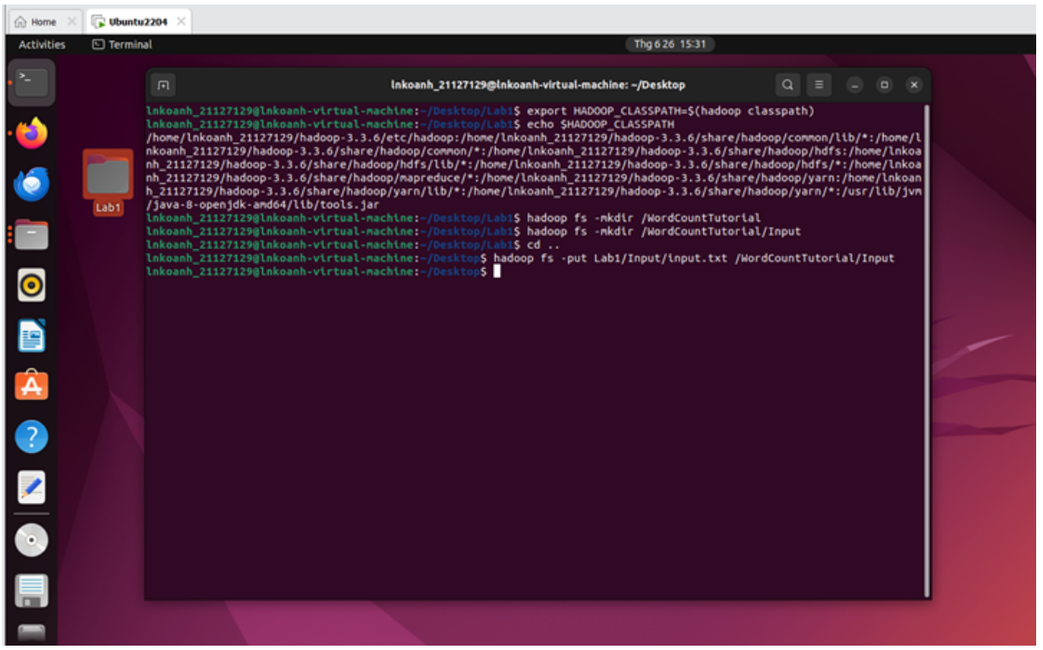
- Create a directory on the Desktop named Lab1 and inside it create two folders, one called *Input* and the other called *tutorial-classes* using GUI normally.



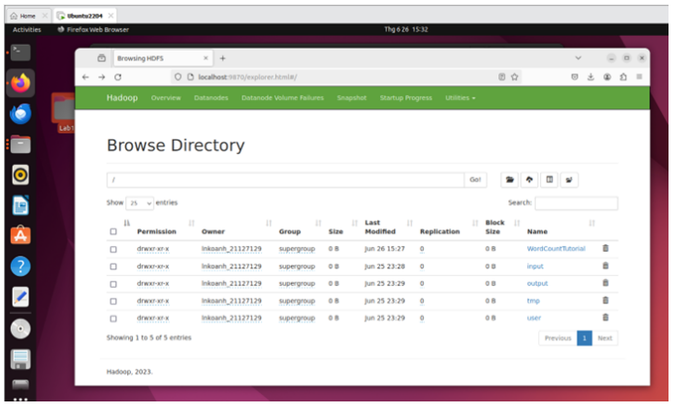
- Let see he Hadoop Web Interface at *localhost:9870* before we upload the file to HDFS.

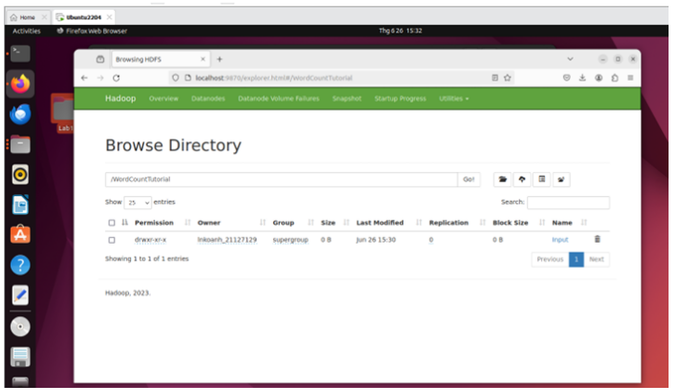


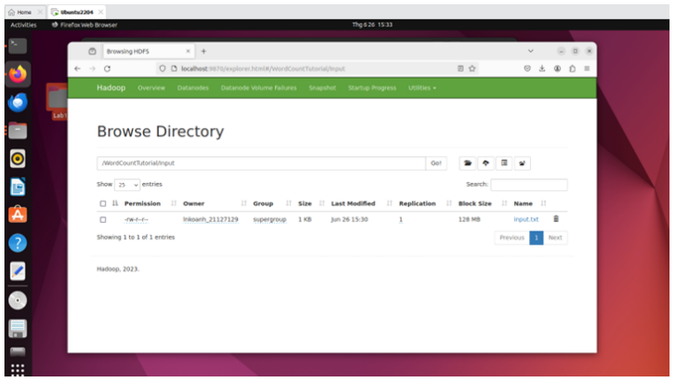
- Add the file attached with this document "WordCount.java" in the directory Lab1 and add the file attached with this document "input.txt" in the directory Lab1/Input using GUI normally.
- Next, we create these directories on HDFS

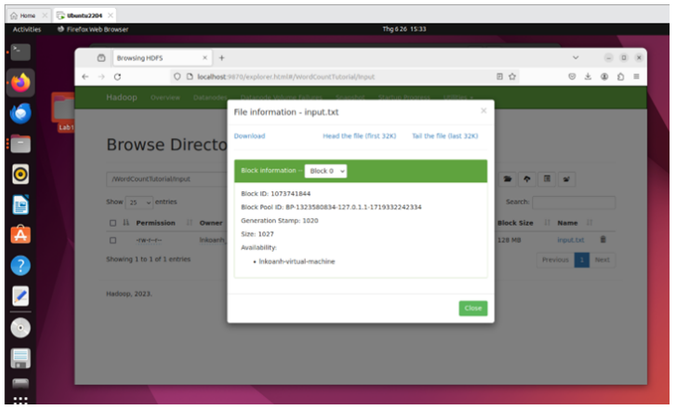


- Now, let see the directories and files we placed in the file system

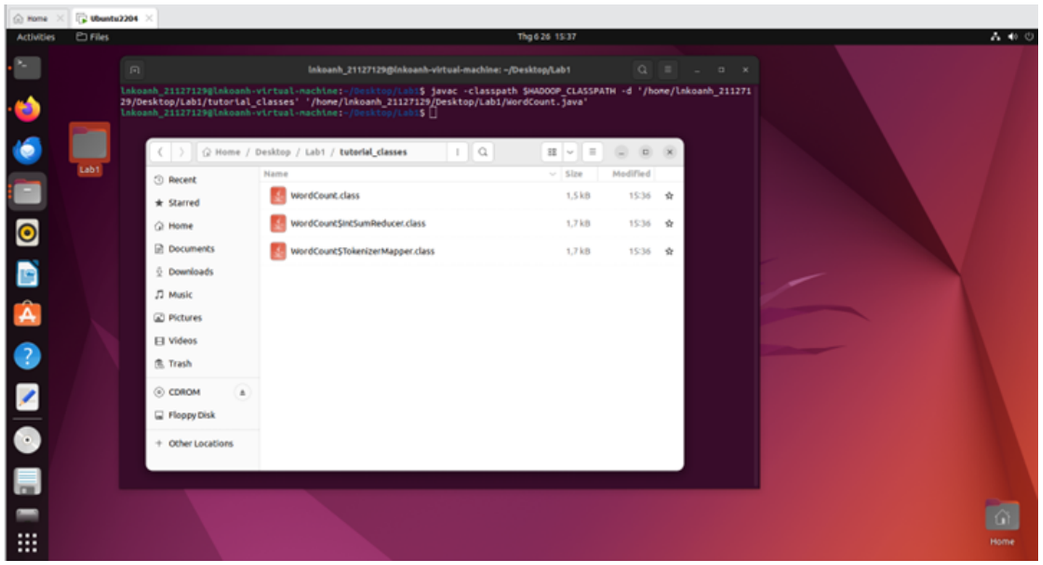




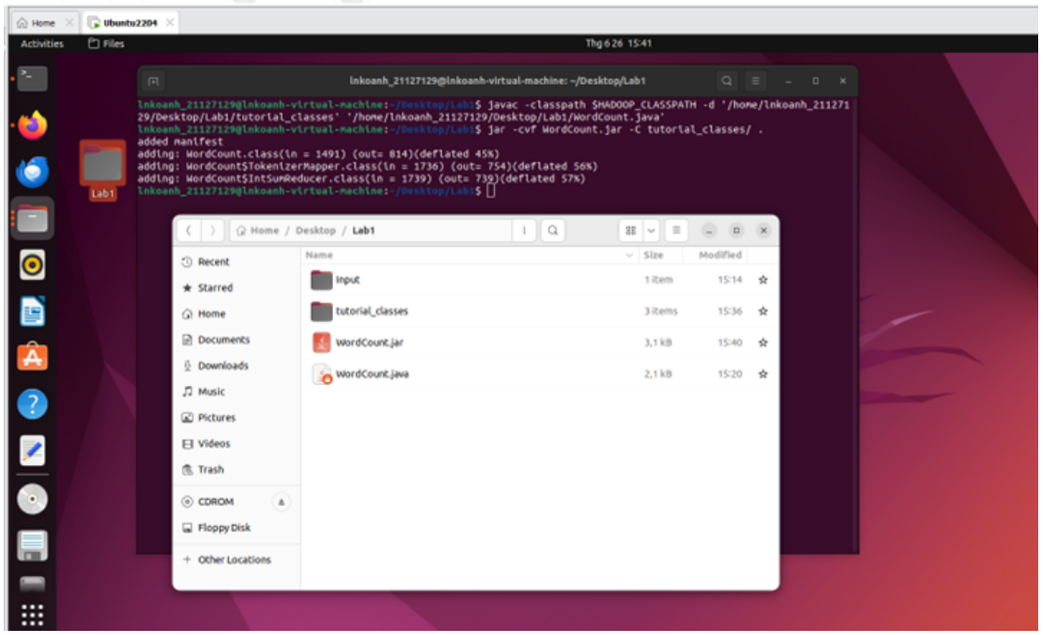




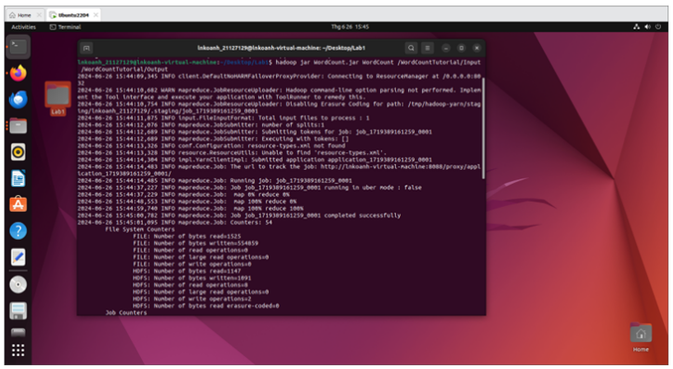
- Compile the "WordCount.java" file

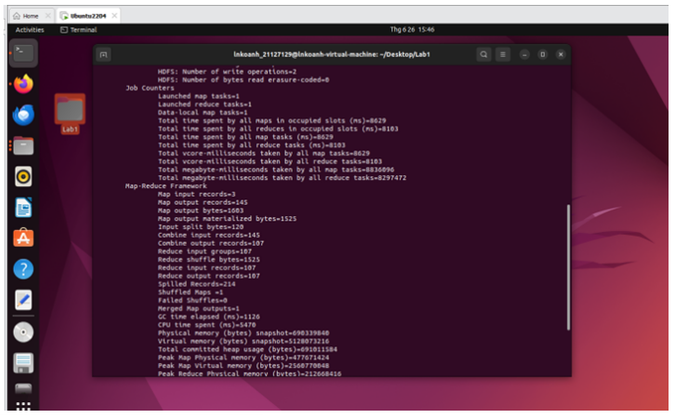


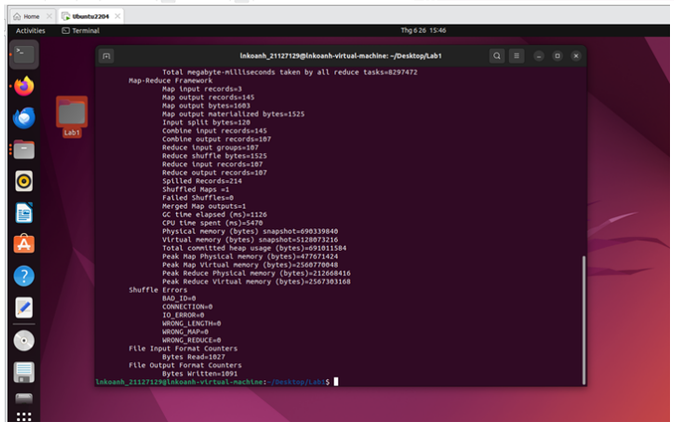
- Putting the output files in one jar file



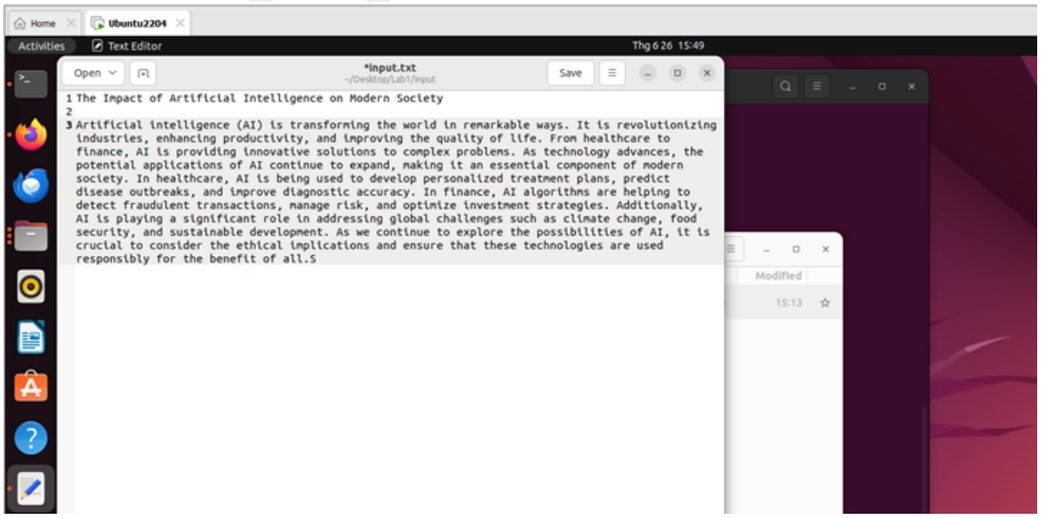
- Running the jar file on Hadoop



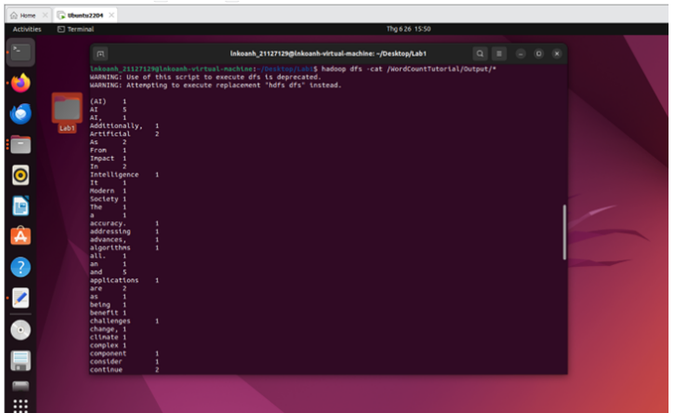


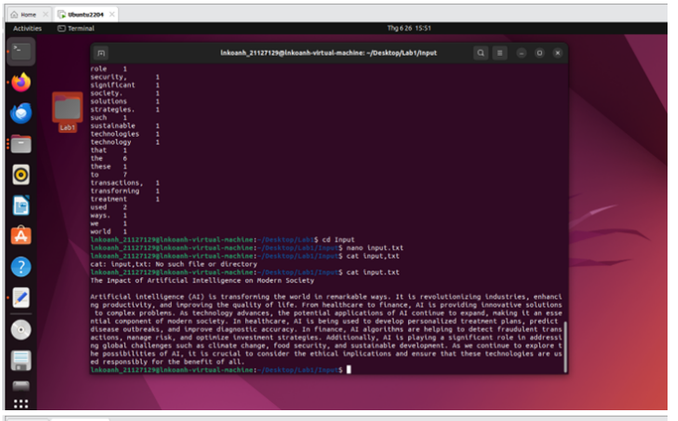


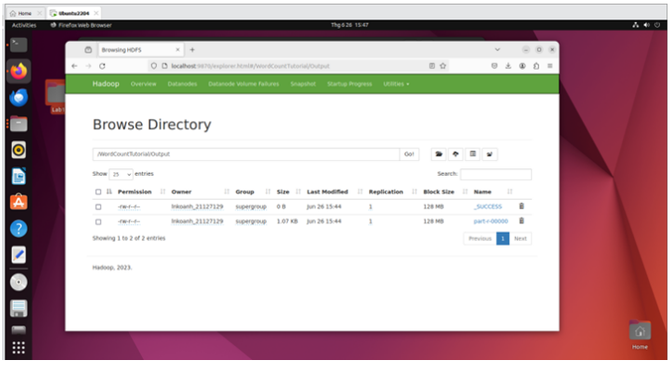
- The input file

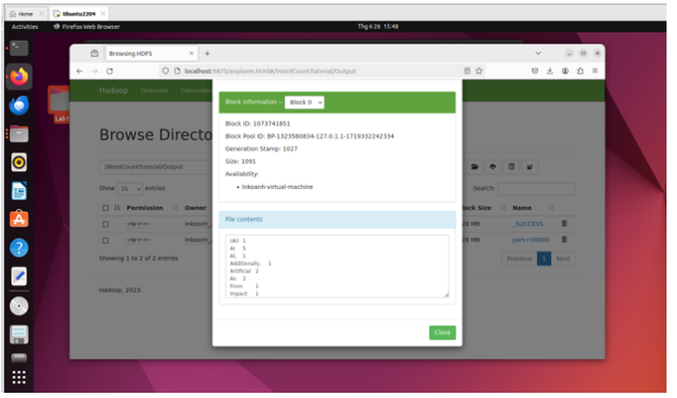


- Finally, we can check the result by using Terminal or Hadoop Web Interface



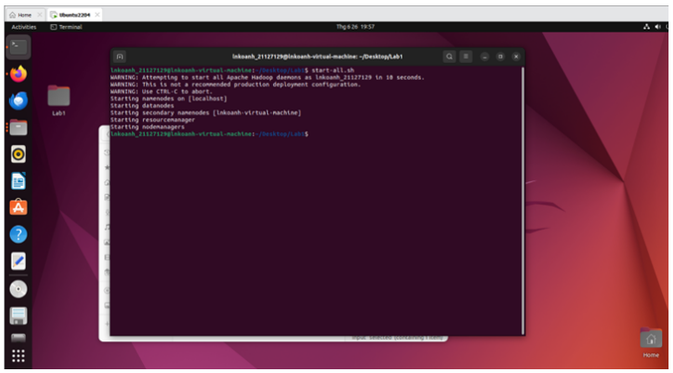


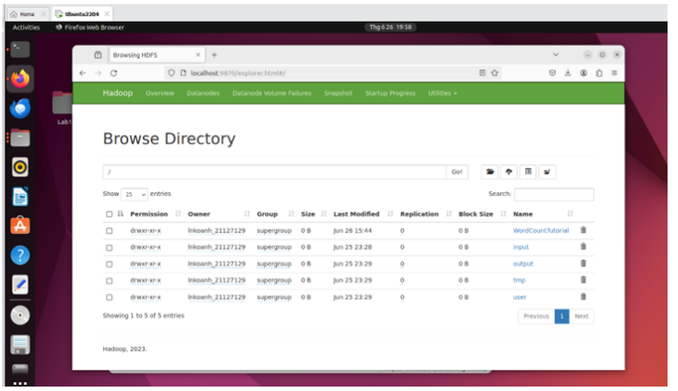




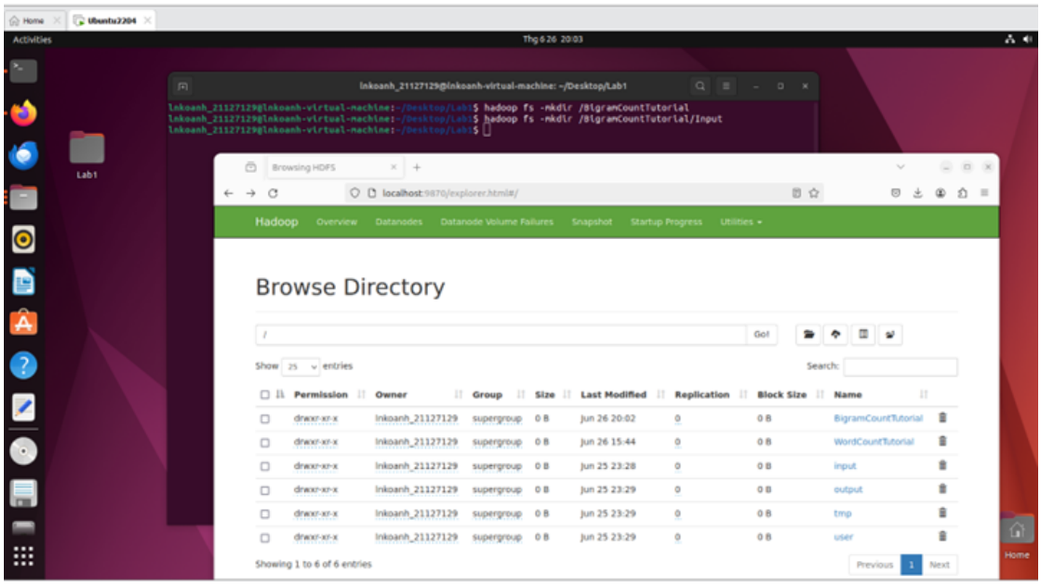
3.2 Bigrams Count

- Firstly, we need to start all the Apache Hadoop Daemons.

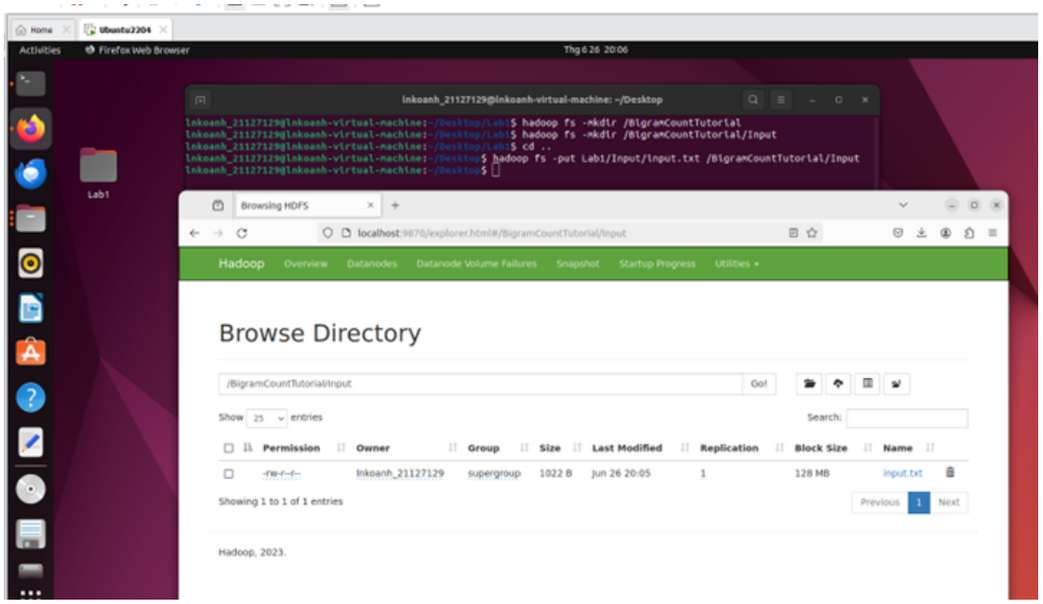




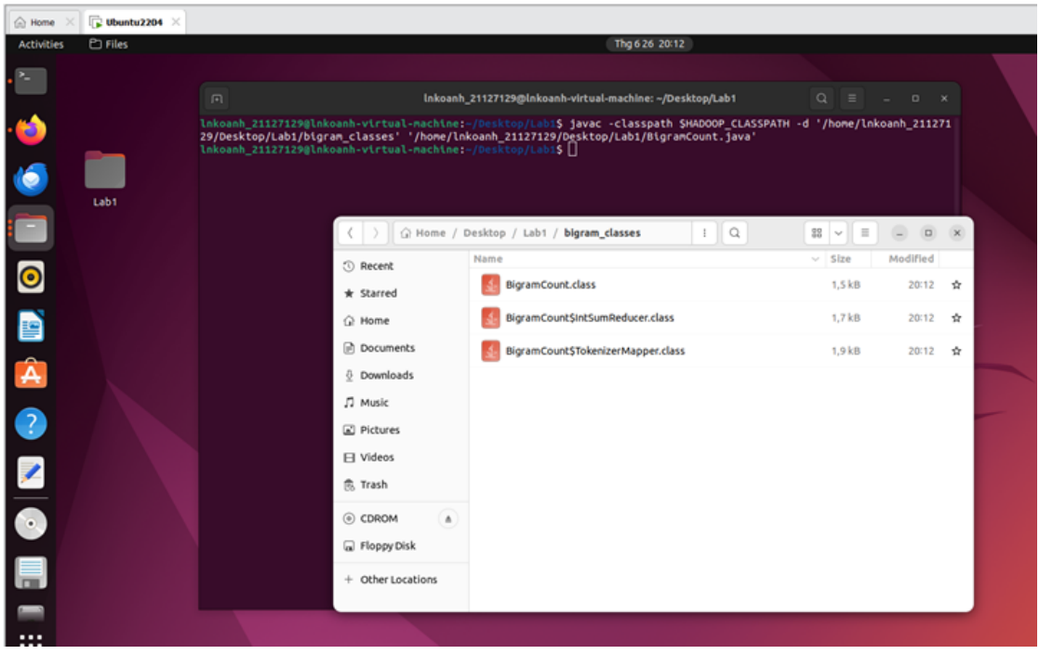
- Create directorys on HDFS



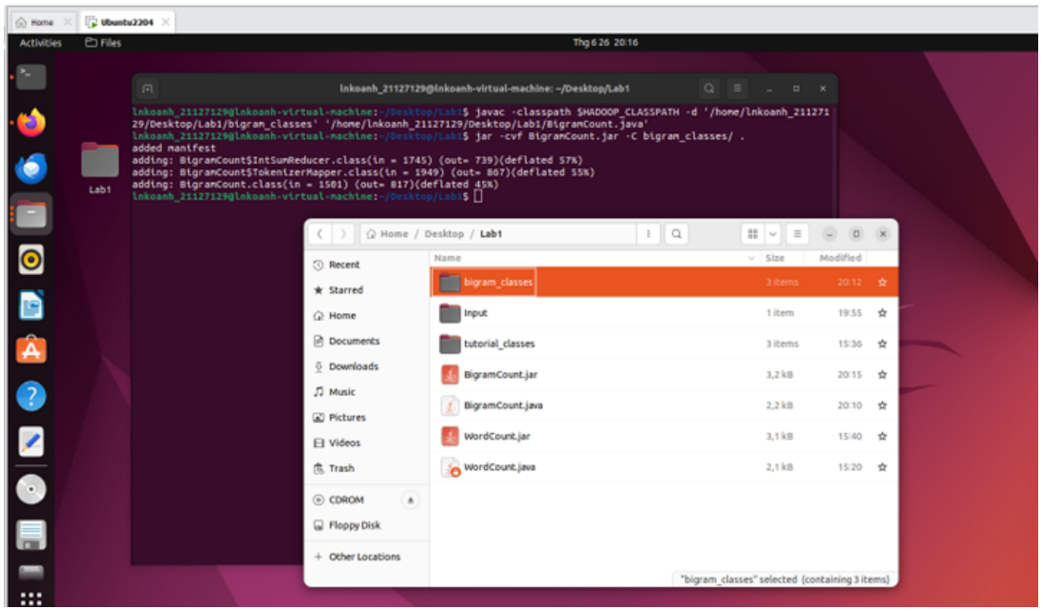
- Upload file "input.txt" on HDFS



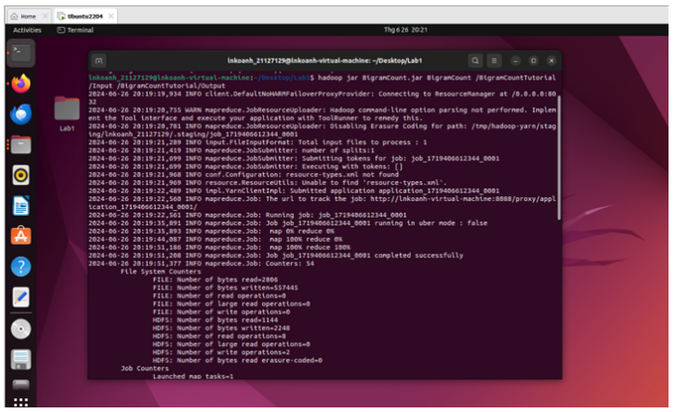
- Compile file "BigramCount.java"

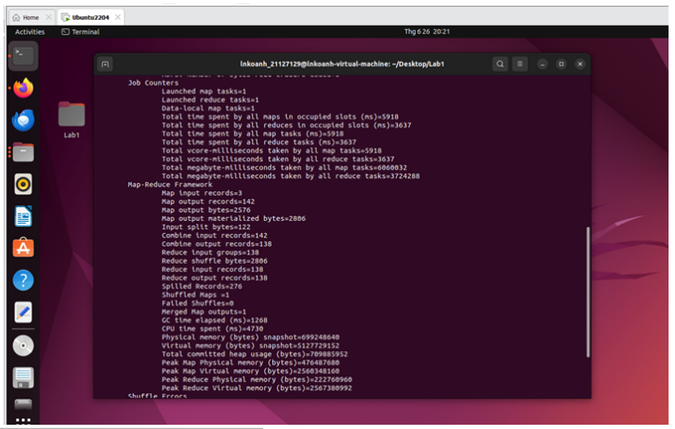


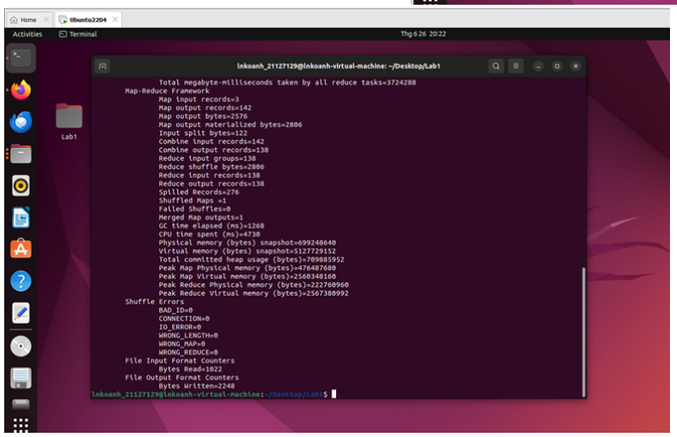
- Putting files in one jar file



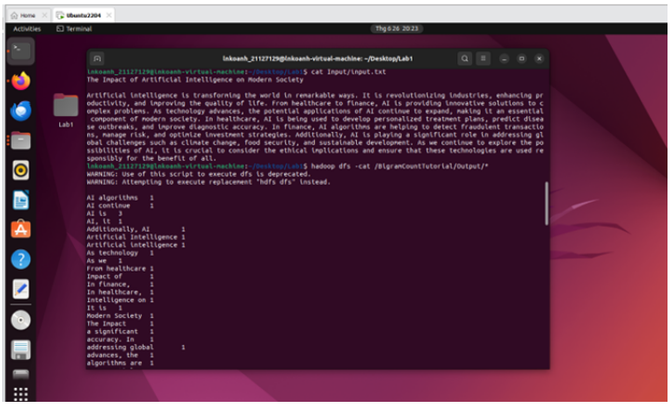
- Running the jar file on Hadoop

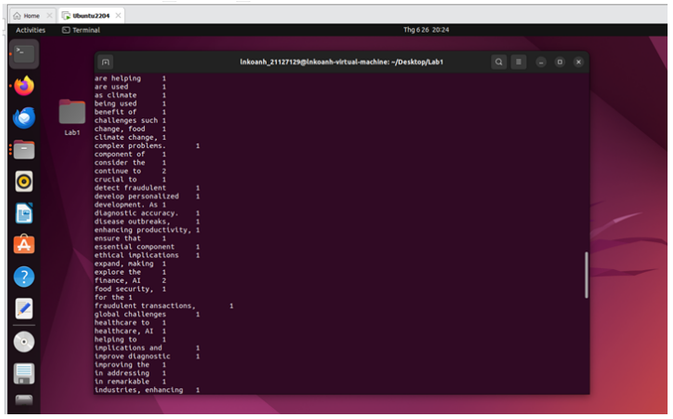


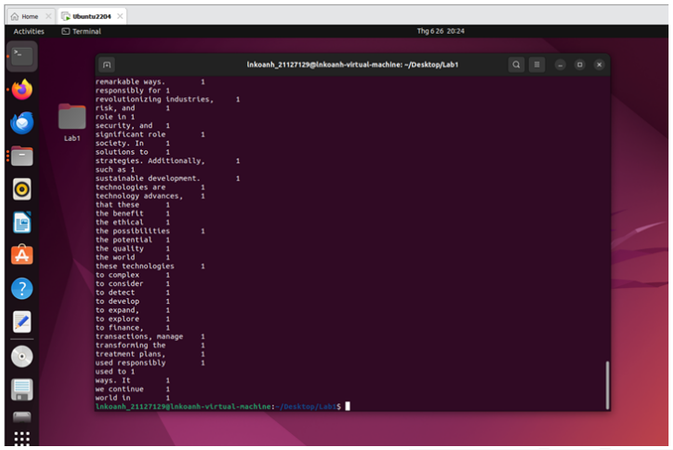


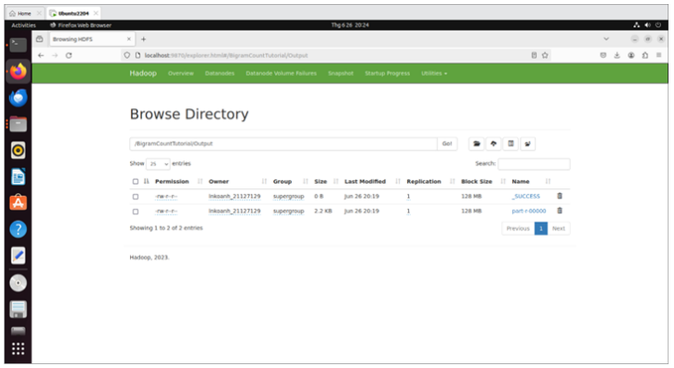


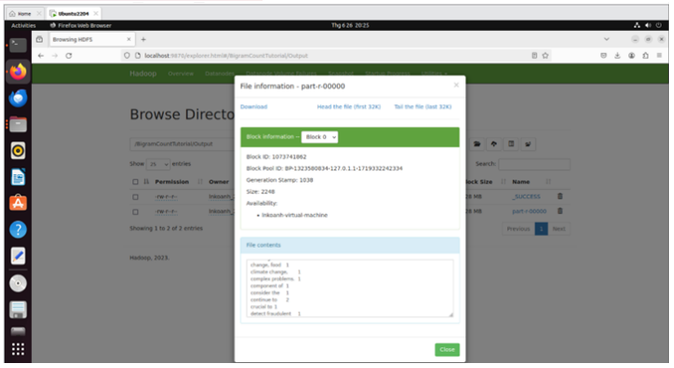
- Finally, we can check the result by using Terminal or Hadoop Web Interface


lnkoanh_21127129@lnkoanh-virtual-machine: ~/Desktop/Lab1
$ cat input/input.txt
The Impact of Artificial Intelligence on Modern Society
Artificial intelligence is transforming the world in remarkable ways. It is revolutionizing industries, enhancing pr
oductivity, and improving the quality of life. From healthcare to finance, AI is providing innovative solutions to c
omplex problems. As technology advances, the potential applications of AI continue to expand, making it an essential
component of modern society. In healthcare, AI is being used to develop personalized treatment plans, predict disea
se outbreaks, and improve diagnostic accuracy. In finance, AI algorithms are helping to detect fraudulent transactio
ns, manage risk, and optimize investment strategies. Additionally, AI is playing a significant role in addressing gl
obal challenges such as climate change, food security, and sustainable development. As we continue to explore the po
ssibilities of AI, it is crucial to consider the ethical implications and ensure that these technologies are used re
sponsibly for the benefit of all.
$ hadoop dfs -cat /BigramCountTutorial/Output/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.
AI algorithms 1
AI continue 1
AI is 3
AI, it 1
Additionally, AI 1
Artificial intelligence 1
Artificial intelligence 1
As technology 1
As we 1
From healthcare 1
Impact of 1
In finance, 1
In healthcare, 1
Intelligence on 1
It is 1
Modern Society 1
The Impact 1
a significant 1
accuracy. In 1
addressing global 1
advances, the 1
algorithms are 1


lnkoanh_21127129@lnkoanh-virtual-machine: ~/Desktop/Lab1
are helping 1
are used 1
as climate 1
being used 1
benefit of 1
challenges such 1
change, food 1
climate change, 1
complex problems. 1
component of 1
consider the 1
continue to 2
crucial to 1
detect fraudulent 1
develop personalized 1
development, AI 1
diagnostic accuracy. 1
disease outbreaks, 1
enhancing productivity, 1
ensure that 1
essential component 1
ethical implications 1
expand, making 1
explore the 1
finance, AI 2
food security, 1
for the 1
fraudulent transactions, 1
global challenges 1
healthcare to 1
healthcare, AI 1
helping to 1
implications and 1
improve diagnostic 1
improving the 1
in addressing 1
in remarkable 1
industries, enhancing 1


lnkoanh_21127129@lnkoanh-virtual-machine: ~/Desktop/Lab1
remarkable ways. 1
responsibly for 1
revolutionizing industries, 1
risk, and 1
role in 1
security, and 1
significant role 1
society. In 1
solutions to 1
strategies. Additionally, 1
such as 1
sustainable development. 1
technologies are 1
technology advances, 1
that these 1
the benefit 1
the ethical 1
the possibilities 1
the potential 1
the quality 1
the world 1
these technologies 1
to complex 1
to consider 1
to detect 1
to develop 1
to expand, 1
to explore 1
to finance, 1
transactions, manage 1
transforming the 1
treatment plans, 1
used responsibly 1
used to 1
ways. It 1
we continue 1
world in 1
lnkoanh_21127129@lnkoanh-virtual-machine: ~/Desktop/Lab1$


Browse Directory
Hadoop, 2023.


File information - part-r-00000
Browse Directo
Hadoop, 2023.

4 Reflection

4.1 Challenges and Bugs Encountered

- The stage of installing the virtual machine and the Ubuntu operating system encountered difficulties in selecting the version and configuration.
- It is not possible to create a folder on HDFS due to the default root directory starting at /user/username.
- In section 4: Setting up Fully Distributed Mode, our team is not located near each other and has not yet found an appropriate solution, so we have not been able to complete the requirement.

4.2 How We Overcame It

- During our setup of the virtual machine and Ubuntu OS, we researched different versions, selected the most suitable for our project needs, and followed a detailed YouTube tutorial for installation guidance. We verified each step against the tutorial, adjusted configurations as necessary, and documented the process for future reference and troubleshooting. This method ensured a smooth setup and optimized our Ubuntu environment effectively.
- It is not possible to create a folder on HDFS due to the default root directory starting at /user/username. To resolve this issue, the folder needs to be created in that directory first, or / should be added at the beginning of the path in any command.

4.3 Lessons Learned

Setting up our Hadoop cluster and running MapReduce programs provided valuable insights into big data processing and distributed computing. We delved deep into distributed systems, mastering key components like NameNode and DataNode, enhancing our cluster architecture knowledge.

The process significantly improved our Linux command skills as we handled tasks such as OS installation, user permissions, network configurations, and software setups. Troubleshooting various issues strengthened our system administration capabilities, particularly in diagnosing configuration errors and optimizing cluster performance.

In summary, our Hadoop experience was educational, boosting our understanding of distributed systems, refining Linux proficiency, and enhancing problem-solving skills crucial for managing

large-scale data processing.

Bibliography

[1] Apache Hadoop 3.3.6 – Hadoop: Setting up a Single Node Cluster https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html

[2] Apache Hadoop 3.3.6 – MapReduce Tutorial https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#Example%3A_WordCount_v1.0

[3] Code With Arjun. (2022, October 2). Install Hadoop on Ubuntu (22.04 / 20.04 LTS) | HDFS | Namenode | DataNode | Big Data Analytics [Video]. YouTube https://www.youtube.com/watch?v=Slbi-uzPtnw

[4] Code With Arjun. (2022b, October 12). MapReduce Word Count Example using Hadoop and Java [Video]. YouTube. https://www.youtube.com/watch?v=qgBu8Go1SyM

[5] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters (research.google) https://research.google/pubs/mapreduce-simplified-data-processing-on-large-clusters/