Hadoop deployment and use HDFS

School of Science and Engineering

Meriem Lmoubariki

Dr. Tajjeeddine Rachidi

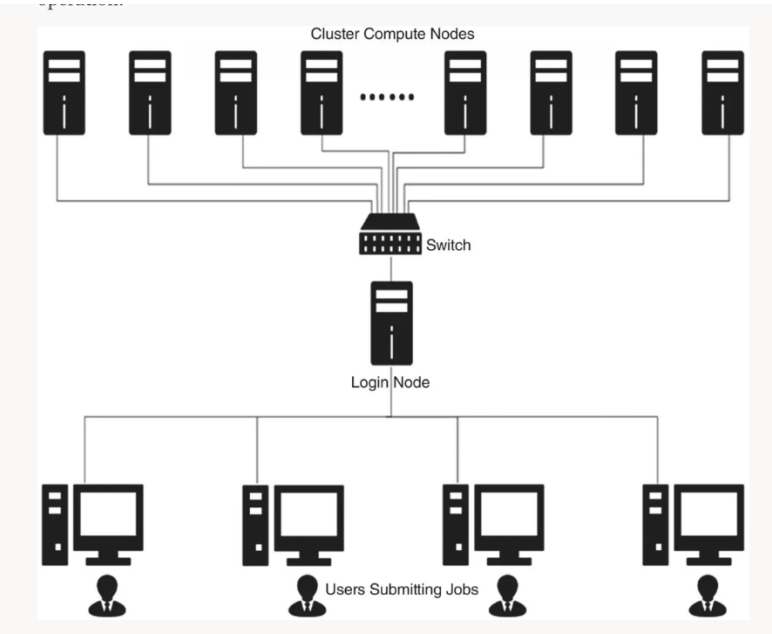
Al Akhawayn University in Ifrane

September 20, 2025

Table of Contents

Introduction

Our textbook describes cluster computing as a system involving multiple standalone PCs linked together to function as a single, integrated resource, offering heightened availability. These computing resources are connected via local area networks (LANs), creating a powerful virtual computer where each resource runs its own instance of an operating system.(1)



Given the exponential growth of data and the need for effective processing capabilities, Hadoop emerges as a vital tool in the big data ecosystem. Hadoop is designed to store and analyze vast amounts of data across a distributed environment using simple programming models. In this assignment , we will see the basics tips on how to manipulate it.

Task 1: Comparison of Hadoop Cluster Deployment Strategies

We have several ways to deploy the cluster: (1) (2) (3) (4)

- Standalone Mode: Hadoop runs on a single computer where the entire system is confined to one physical location.
- Pseudo-Distributed Mode: Each part of Hadoop that normally runs on separate machines runs on one machine.
- Fully Distributed Mode: Hadoop runs across many machines, splitting tasks among them to handle very large data sets efficiently.
- Cloud-Based Deployment: Hadoop runs on virtual machines in a data center managed by a cloud provider.
- Hybrid Deployment: Aa mix of on-premises and cloud-based resources, providing flexibility, especially for sensitive data that require more secure handling, while still offering scalability for less critical tasks

Strategy	Pros	Cons	Best use case
Standalone Mode	- Easy to set up and use. - Costs less as it uses existing computers.	- Only uses one computer. - Not good for big tasks or real-world use	Ideal for development and testing small datasets.
Pseudo-Distributed Mode	- Acts like a cluster but on one machine. - Good for catching problems early.	- Still limited to one computer. - Can't handle large data sets.	Suitable for developers simulating a cluster environment.

Fully Distributed Mode	- Uses many computers, so it's powerful and reliable. - Can grow as needed by adding more machines.	- Costs a lot to set up and keep running. - Needs a lot of technical skill to manage.	Large-scale data processing in enterprise environments.
Cloud-Based Deployment	- Can adjust to needs quickly, scaling up or down. - Less work to maintain as the service provider handles it.	- Security concerns since data is online. - Needs constant internet.	Businesses needing flexible computing resources without major upfront investment.
Hybrid Deployment	- Best of both worlds: control over critical data and cloud flexibility. - Can plan for disasters better	- Hard to manage because it combines two systems. - Can get expensive depending on setup.	Organizations with fluctuating workloads and strict data governance.

Real life examples

1. Standalone Mode

Example: A software developer working from home uses standalone mode to prototype a new data parsing algorithm. The initial development and testing phases are done on his local machine to debug and optimize the code before considering deployment on a larger scale.
Reference: "Using Hadoop in Standalone Mode," an article on DataFlair, DataFlair - Standalone Mode Hadoop, 2022.

2. Pseudo-Distributed Mode

Example: A tech startup utilizes pseudo-distributed mode to test their new recommendation system which integrates with their existing online retail platform. This setup allows them to simulate a distributed environment while using limited resources. Reference: "Configuring Hadoop in a Pseudo-Distributed Mode," a tutorial on Tutorialspoint, Tutorialspoint - Pseudo-Distributed Hadoop, 2021.

3. Fully Distributed Mode

Example: A healthcare analytics company processes large datasets of patient records across a fully distributed Hadoop cluster to identify trends and improve care outcomes. This deployment is crucial for handling the vast amount of data efficiently and in real time. Reference: "Benefits of Fully Distributed Mode in Hadoop," an analysis on GeeksforGeeks, GeeksforGeeks - Fully Distributed Hadoop, 2022.

4. Cloud-Based Deployment

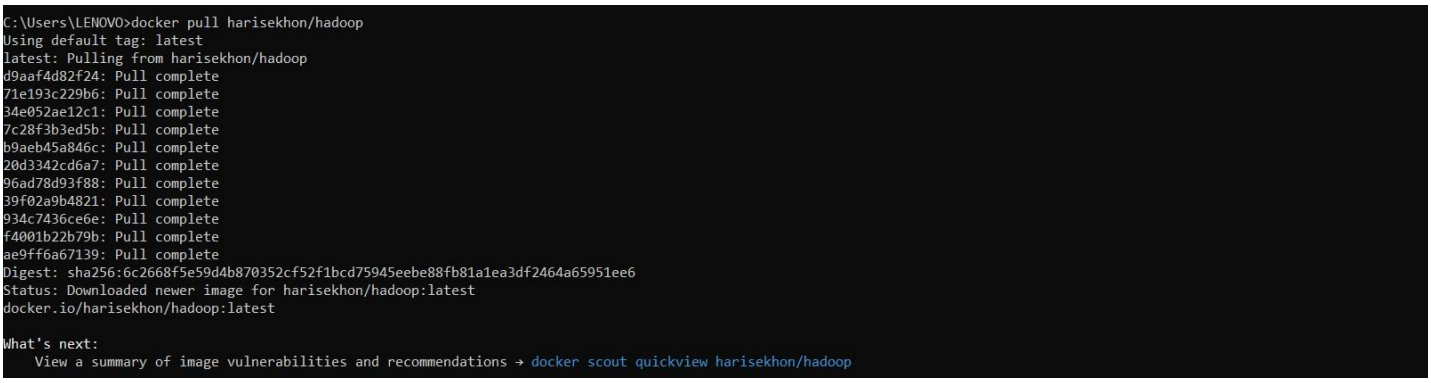
Example: An online media streaming service experiences variable viewer traffic and uses a cloud-based Hadoop deployment to manage this variability. They leverage Amazon EMR to quickly scale their processing capabilities during peak hours. Reference: "Scaling Big Data Processing with Cloud-Based Hadoop," an insight on InfoQ, InfoQ - Cloud-Based Hadoop, 2021.

5. Hybrid Deployment

Example: A multinational bank employs a hybrid Hadoop deployment where sensitive financial data is processed and stored on-premises while less sensitive data analytics tasks are performed in the cloud. This approach helps them maintain stringent security standards while still capitalizing on the scalability of cloud resources. Reference: "Navigating Hybrid Hadoop Deployments in Financial Services," an article on Hortonworks, Hortonworks - Hybrid Hadoop Deployment, 2020.

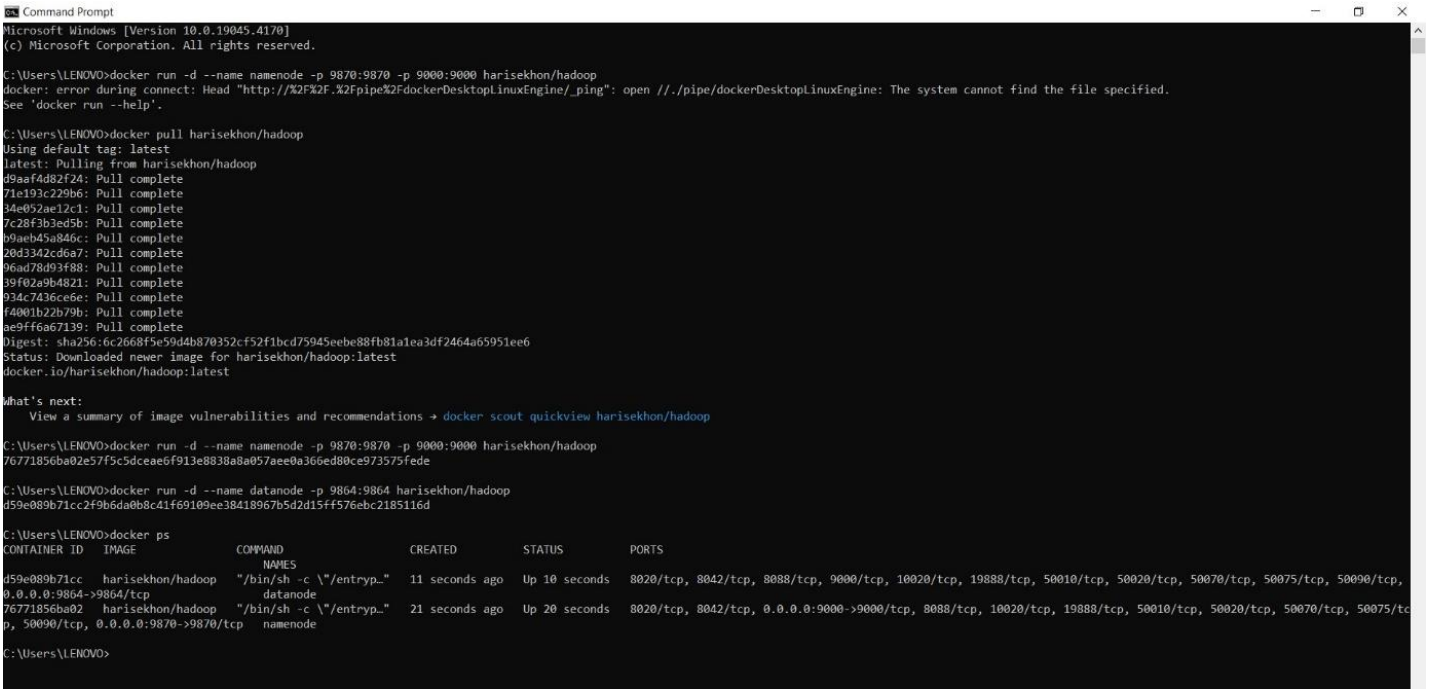
Task 2: Docker Integration and Hadoop Image Deployment

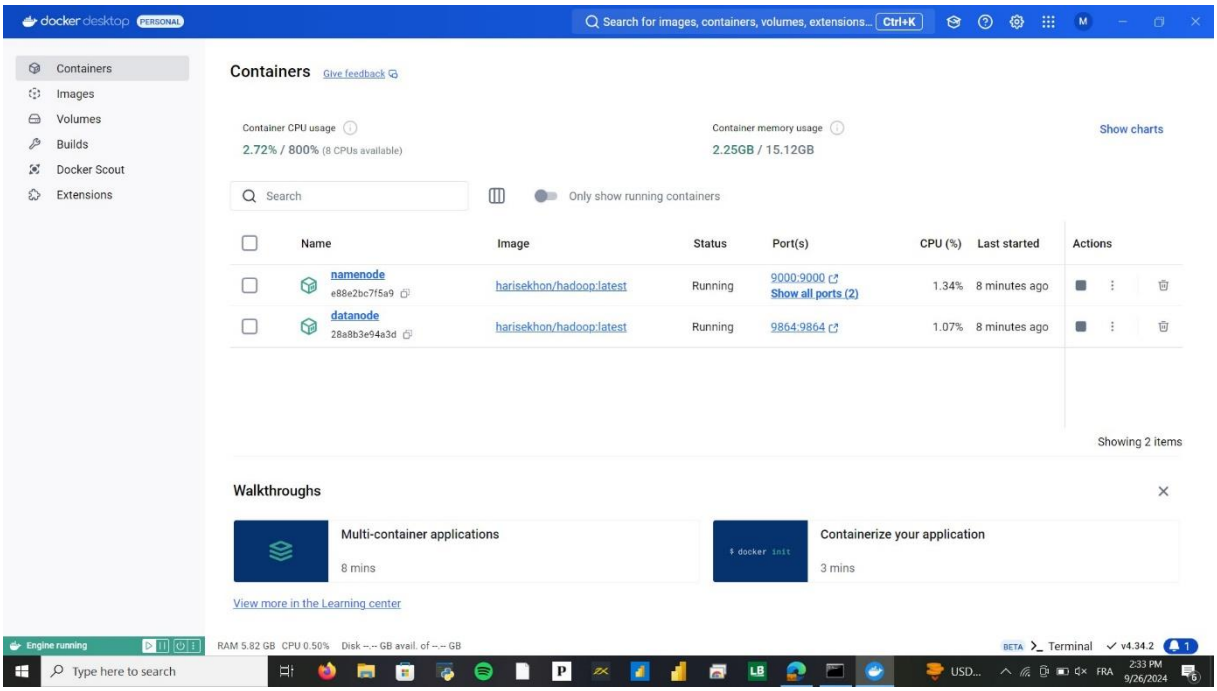
I successfully downloaded the Harisekhon Hadoop Docker image, which makes setting up Hadoop straightforward. This Docker image includes everything needed to run Hadoop without additional configuration.



Task 3: Configuration Insights of Namenode and Datanode

I have created two different containers.





For the Namenode, I demonstrated how to locate the fsimage and edit logs, which are crucial for understanding the state of the HDFS filesystem. For the Datanode, I showed where the data blocks, which store the actual data, are located.

```
@76771856ba02:/
[root@76771856ba02 /]#  cat /hadoop-2.8.2/etc/hadoop/hdfs-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
</configuration>
[root@76771856ba02 /]# find / -name 'dfs'
/tmp/hadoop-root/dfs
[root@76771856ba02 /]# ls /tmp/hadoop-root/dfs/name/current
VERSION                             edits_inprogress_0000000000000000003  fsimage_0000000000000000000.md5  fsimage_0000000000000000002.md5
edits_0000000000000000001-0000000000000000002  fsimage_0000000000000000000           fsimage_0000000000000000002      seen_txid
[root@76771856ba02 /]# ls /tmp/hadoop-root/dfs/data/current
BP-1947034320-172.17.0.2-1510154331033  VERSION
[root@76771856ba02 /]#
```

On the name node I showed where the fsimage and edit logs are located.

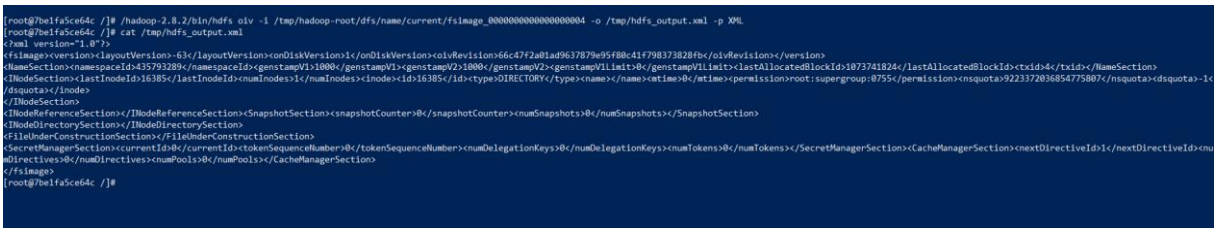
Data blocks

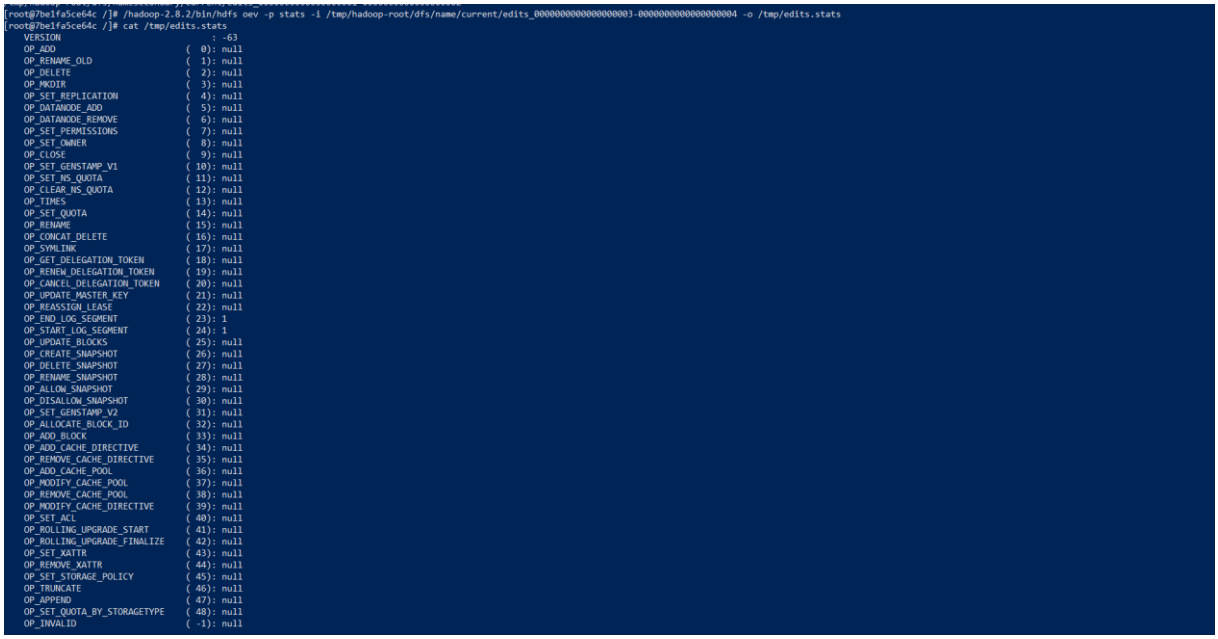
```
Command Prompt
C:\Users\LENOVO>docker exec -it datanode /bin/bash
[root@c450a54a7c0e /]# ls /tmp/hadoop-root/dfs/data/current
BP-1947034320-172.17.0.2-1510154331033  VERSION
[root@c450a54a7c0e /]# exit
exit

C:\Users\LENOVO>
```

```
exit
PS C:\Users\LENOVO> docker cp "C:\Users\LENOVO\OneDrive\Bureau\nouveau 1.txt" datanode:/hadoop/dfs/data/
Successfully copied 2.05kB to datanode:/hadoop/dfs/data/
PS C:\Users\LENOVO>
```

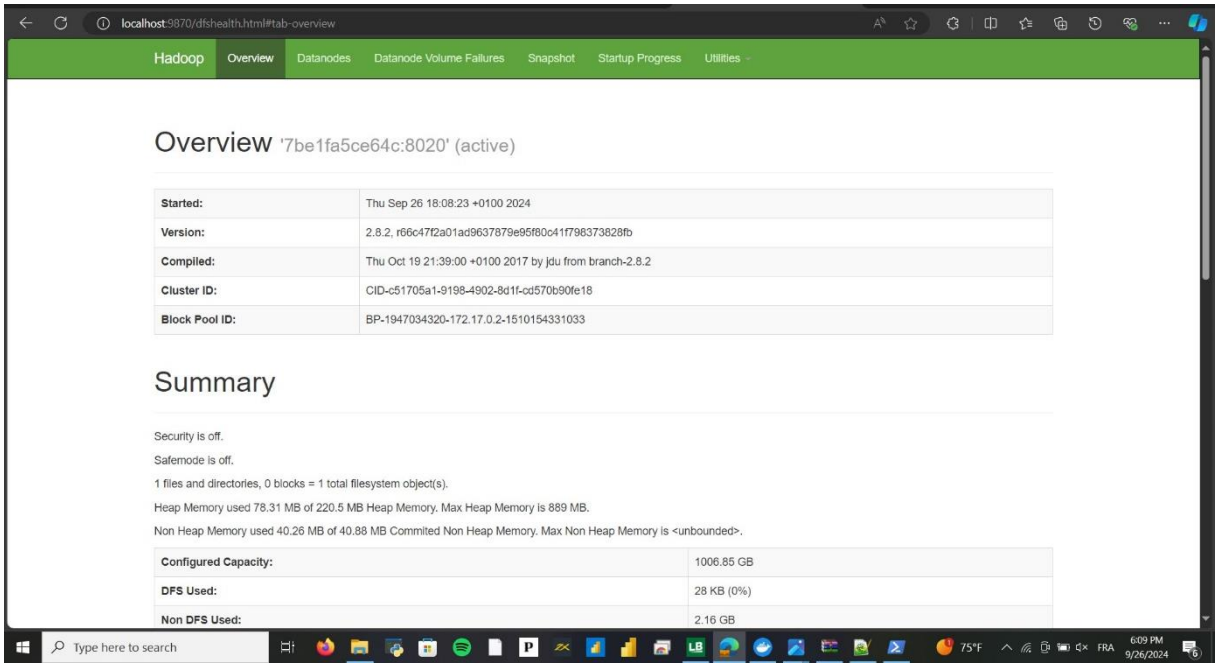
```
The filesystem under path '/user/root/mydata/nouveau1.txt' is HEALTHY
[root@c450a54a7c0e /]# hdfs fsck /user/root/mydata/nouveau1.txt -files -blocks -locations
Connecting to namenode via http://c450a54a7c0e:50070/fsck?ugi=root&files=1&blocks=1&locations=1&path=%2Fuser%2Froot%2Fmy
data%2Fnouveau1.txt
FSCK started by root (auth:SIMPLE) from /172.17.0.3 for path /user/root/mydata/nouveau1.txt at Thu Sep 26 16:14:57 UTC 2
024
/user/root/mydata/nouveau1.txt 17 bytes, 1 block(s):  OK
0. BP-1947034320-172.17.0.2-1510154331033:blk_1073741825_1001 len=17 Live_repl=1 [DatanodeInfoWithStorage[172.17.0.3:500
10,DS-a8a8726a-57df-4bbb-826c-78a81bd27dc1,DISK]]

Status: HEALTHY
 Total size:    17 B
 Total dirs:    0
 Total files:   1
 Total symlinks:                0
 Total blocks (validated):      1 (avg. block size 17 B)
 Minimally replicated blocks:   1 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
 Number of data-nodes:          1
 Number of racks:               1
FSCK ended at Thu Sep 26 16:14:57 UTC 2024 in 0 milliseconds


The filesystem under path '/user/root/mydata/nouveau1.txt' is HEALTHY
```

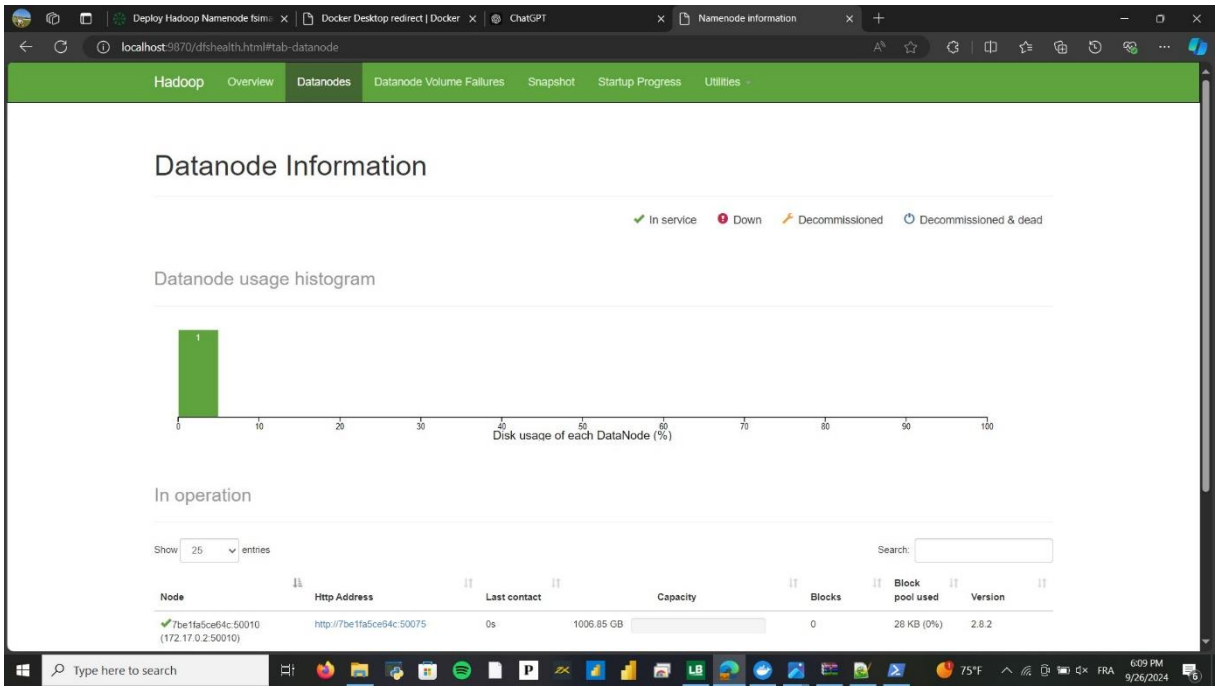
Task 4: Command Line Mastery for HDFS Information Extraction

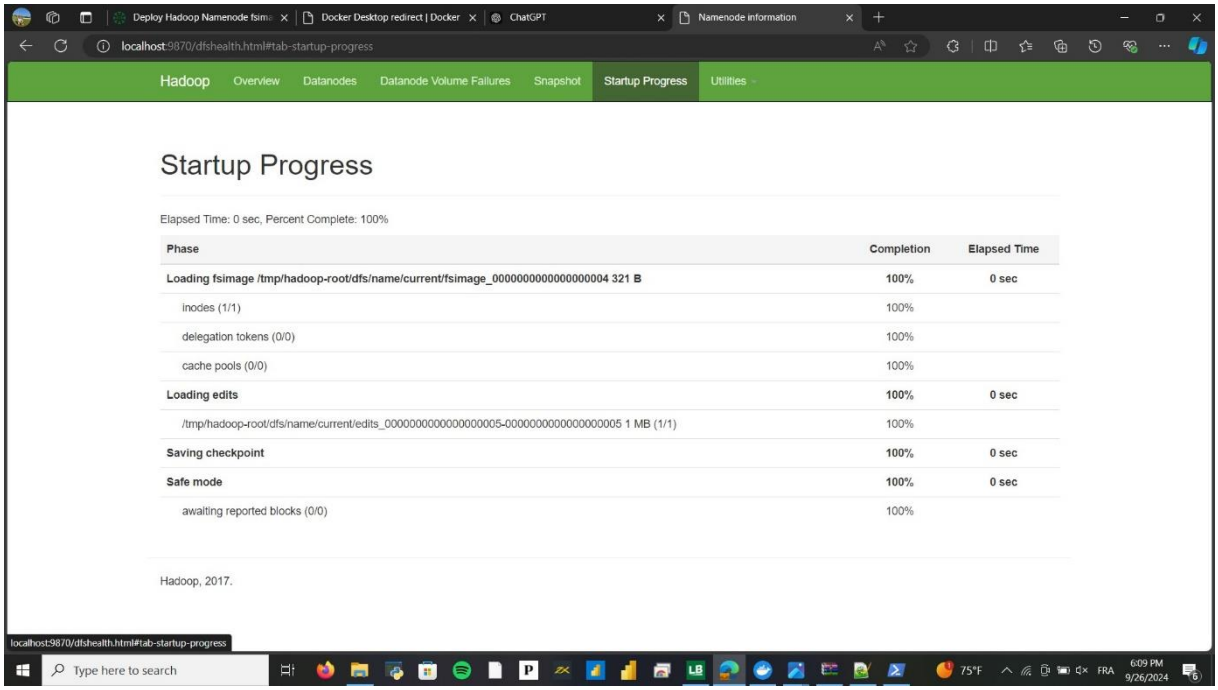




Task 5: Web Interface Utilization for Detailed HDFS Insights

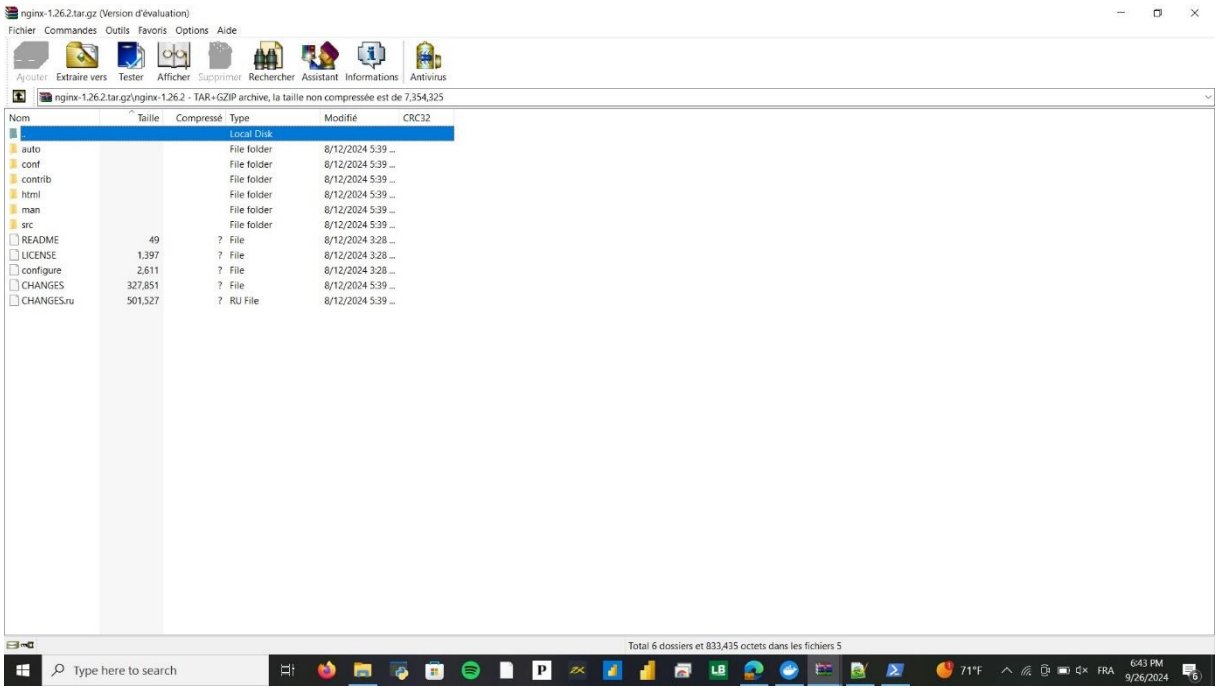


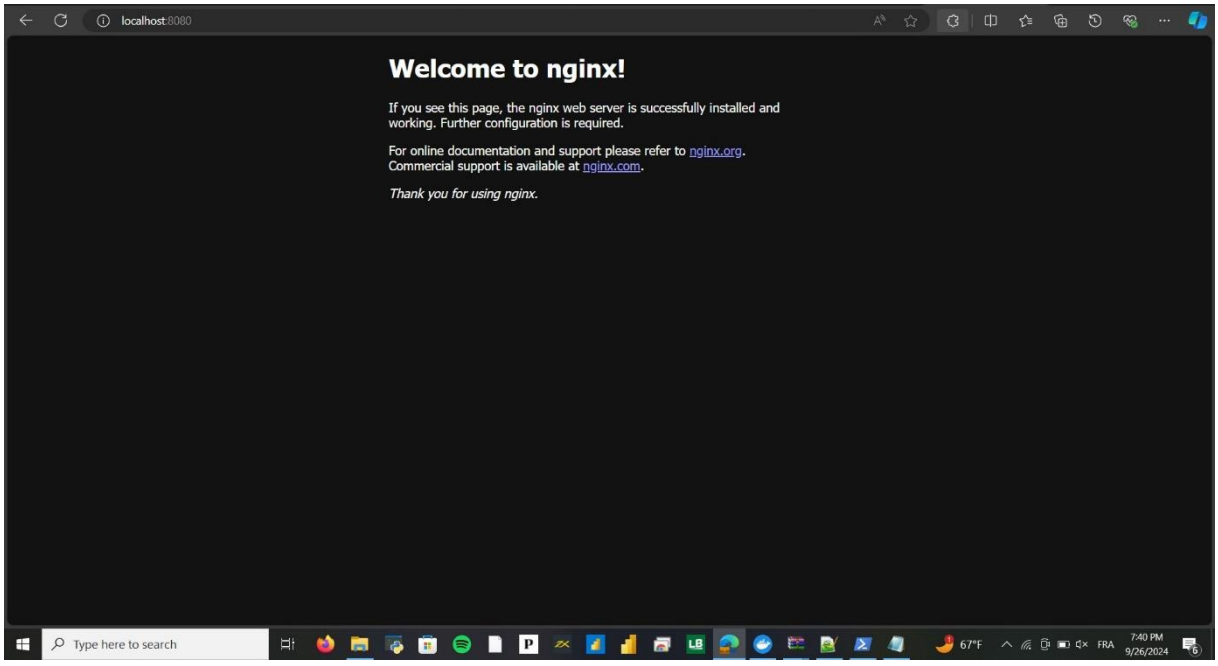

Deploy Hadoop Namenode fsima
Docker Desktop redirect | Docker
ChatGPT
Namenode information
localhost:9870/dfshealth.html#tab-datanode
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Datanode Information
In service
Down
Decommissioned
Decommissioned & dead
Datanode usage histogram
1
0
10
20
30
40
50
60
70
80
90
100
Disk usage of each DataNode (%)
In operation
Show 25 entries
Search:
Node
Http Address
Last contact
Capacity
Blocks
Block pool used
Version
7be1fa5ce64c:50010 (172.17.0.2:50010)
http://7be1fa5ce64c:50075
0s
1006.85 GB
0
28 KB (0%)
2.8.2
Type here to search
75°F
FRA
6:09 PM
9/26/2024

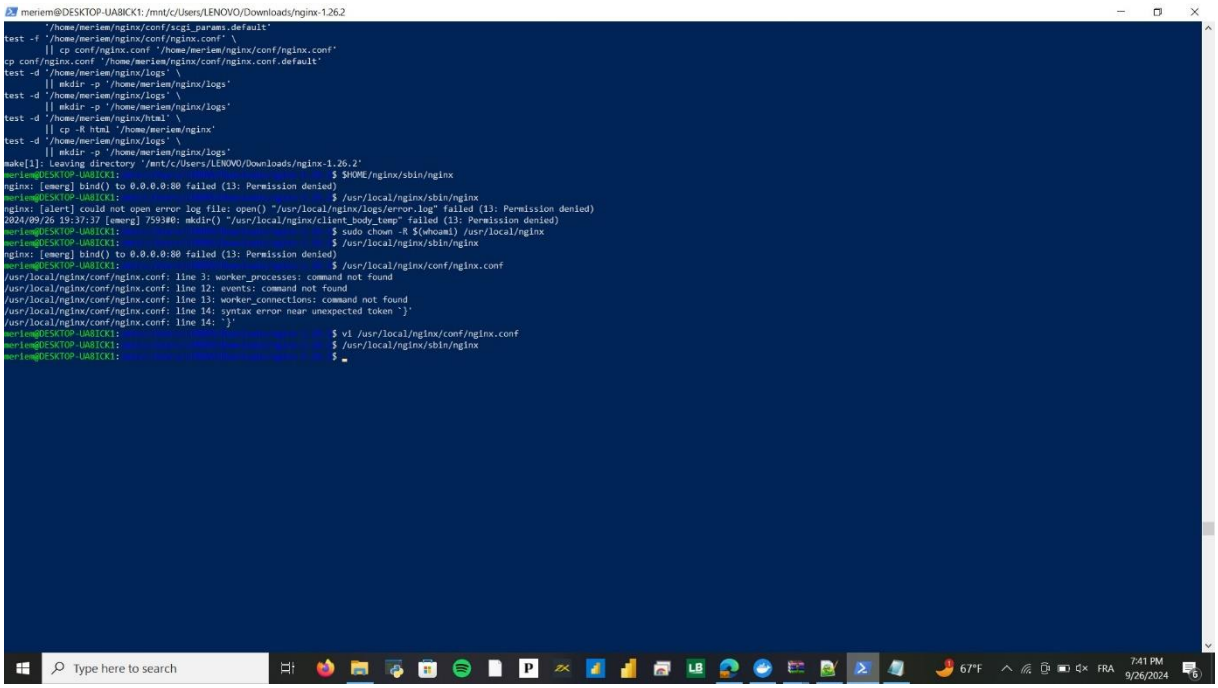

Deploy Hadoop Namenode fsima
Docker Desktop redirect | Docker
ChatGPT
Namenode information
localhost:9870/dfshealth.html#tab-startup-progress
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Startup Progress
Elapsed Time: 0 sec, Percent Complete: 100%
Phase
Completion
Elapsed Time
Loading fsimage /tmp/hadoop-root/dfs/name/current/fsimage_0000000000000000004 321 B
100%
0 sec
inodes (1/1)
100%
delegation tokens (0/0)
100%
cache pools (0/0)
100%
Loading edits
100%
0 sec
/tmp/hadoop-root/dfs/name/current/edits_0000000000000000005-0000000000000000005 1 MB (1/1)
100%
Saving checkpoint
100%
0 sec
Safe mode
100%
0 sec
awaiting reported blocks (0/0)
100%
Hadoop, 2017.
localhost:9870/dfshealth.html#tab-startup-progress
Type here to search
75°F
FRA
6:09 PM
9/26/2024

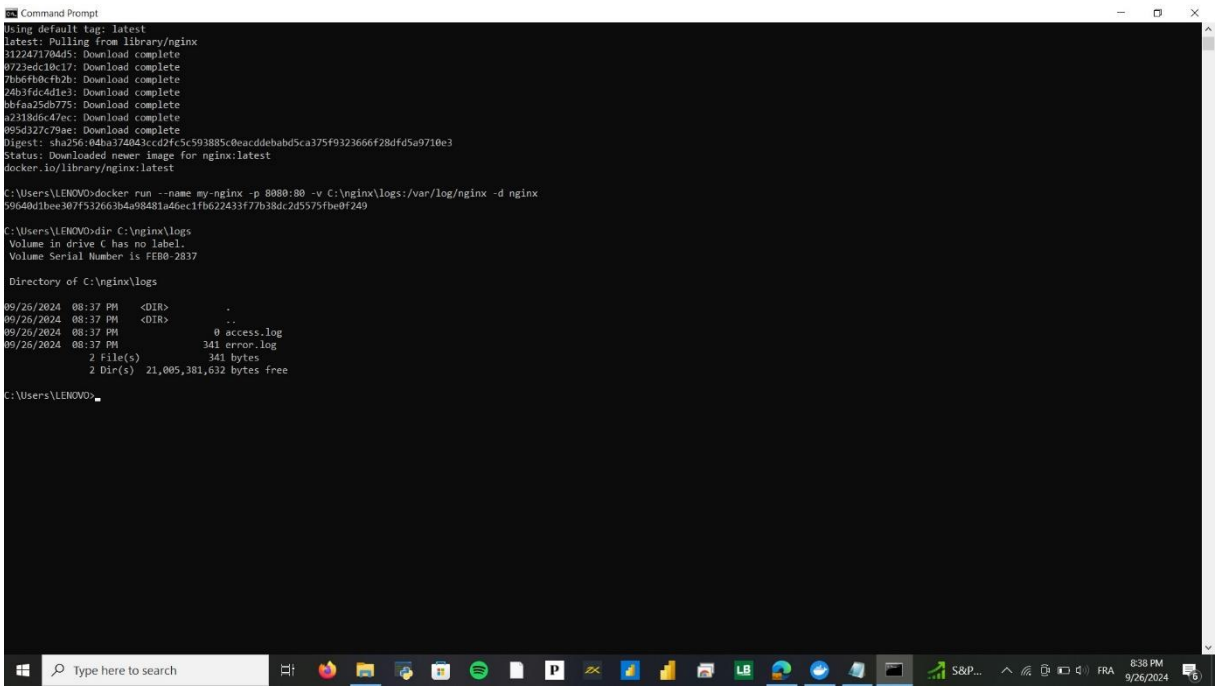
Task 6: Real-World Data Manipulation with Nginx Logs

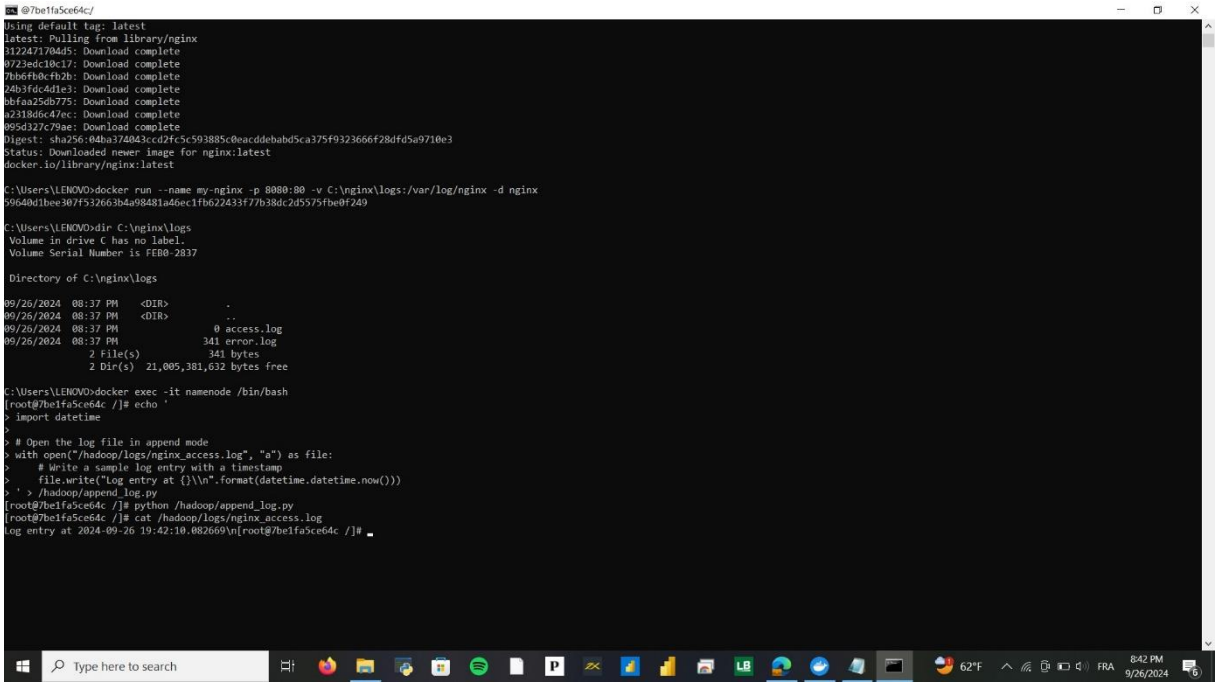
I used a combination of the echo command in the command prompt and Python scripting to interact with my system. First, I ran the echo command to quickly display or pass data to my Python script.






meriem@DESKTOP-UA8ICK1: /mnt/c/Users/LENOVO/Downloads/nginx-1.26.2
'/home/meriem/nginx/conf/scgi_params.default'
test -f '/home/meriem/nginx/conf/nginx.conf' \
|| cp conf/nginx.conf '/home/meriem/nginx/conf/nginx.conf'
cp conf/nginx.conf '/home/meriem/nginx/conf/nginx.conf.default'
test -d '/home/meriem/nginx/logs' \
|| mkdir -p '/home/meriem/nginx/logs'
test -d '/home/meriem/nginx/logs' \
|| mkdir -p '/home/meriem/nginx/logs'
test -d '/home/meriem/nginx/html' \
|| cp -R html '/home/meriem/nginx'
test -d '/home/meriem/nginx/logs' \
|| mkdir -p '/home/meriem/nginx/logs'
make[1]: Leaving directory '/mnt/c/Users/LENOVO/Downloads/nginx-1.26.2'
$ $HOME/nginx/sbin/nginx
nginx: [emerg] bind() to 0.0.0.0:80 failed (13: Permission denied)
$ /usr/local/nginx/sbin/nginx
nginx: [alert] could not open error log file: open() "/usr/local/nginx/logs/error.log" failed (13: Permission denied)
2024/09/26 19:37:37 [emerg] 7593#0: mkdir() "/usr/local/nginx/client_body_temp" failed (13: Permission denied)
$ sudo chown -R ${whoami} /usr/local/nginx
$ /usr/local/nginx/sbin/nginx
nginx: [emerg] bind() to 0.0.0.0:80 failed (13: Permission denied)
$ /usr/local/nginx/conf/nginx.conf
/usr/local/nginx/conf/nginx.conf: line 3: worker_processes: command not found
/usr/local/nginx/conf/nginx.conf: line 12: events: command not found
/usr/local/nginx/conf/nginx.conf: line 13: worker_connections: command not found
/usr/local/nginx/conf/nginx.conf: line 14: syntax error near unexpected token `}'
/usr/local/nginx/conf/nginx.conf: line 14: `}'
$ vi /usr/local/nginx/conf/nginx.conf
$ /usr/local/nginx/sbin/nginx
$
Type here to search
67°F
FRA
7:41 PM
9/26/2024

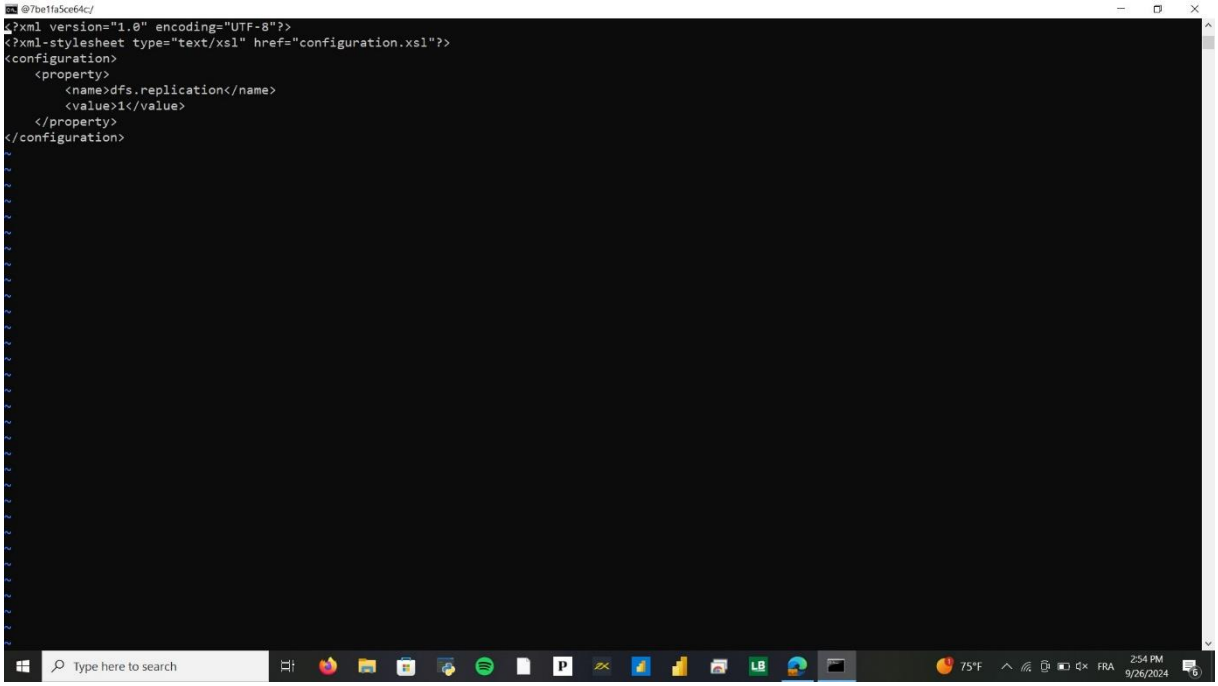

Command Prompt
Using default tag: latest
latest: Pulling from library/nginx
3122471704d5: Download complete
0723edc10c17: Download complete
7bb6fb0cfb2b: Download complete
24b3fdc4d1e3: Download complete
bbfaa25db775: Download complete
a2318d6c47ec: Download complete
005d327c79ae: Download complete
Digest: sha256:04ba374043ccd2fc5c593885c0eacddebabd5ca375f9323666f28dfd5a9710e3
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
C:\Users\LENOVO>docker run --name my-nginx -p 8080:80 -v C:\nginx\logs:/var/log/nginx -d nginx
59640d1bee307f532663b4a98481a46ec1fb622433f77b38dc2d5575fbe0f249
C:\Users\LENOVO>dir C:\nginx\logs
Volume in drive C has no label.
Volume Serial Number is FEB0-2837
Directory of C:\nginx\logs
09/26/2024 08:37 PM <DIR> .
09/26/2024 08:37 PM <DIR> ..
09/26/2024 08:37 PM 0 access.log
09/26/2024 08:37 PM 341 error.log
2 File(s) 341 bytes
2 Dir(s) 21,005,381,632 bytes free
C:\Users\LENOVO>
Type here to search
S&P...
FRA
8:38 PM
9/26/2024


@7be1fa5ce64c:/
Using default tag: latest
latest: Pulling from library/nginx
3122471704d5: Download complete
0723edc10c17: Download complete
7bb6fb0cfb2b: Download complete
24b3fdc4d1e3: Download complete
bbfaa25db775: Download complete
a2318d6c47ec: Download complete
095d327c79ae: Download complete
Digest: sha256:04ba374043ccd2fc5c593885c0eacddebabd5ca375f9323666f28dfd5a9710e3
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
C:\Users\LENOVO>docker run --name my-nginx -p 8080:80 -v C:\nginx\logs:/var/log/nginx -d nginx
59640d1bee307f532663b4a98481a46ec1fb622433f77b38dc2d5575fbe0f249
C:\Users\LENOVO>dir C:\nginx\logs
Volume in drive C has no label.
Volume Serial Number is FEB0-2837
Directory of C:\nginx\logs
09/26/2024 08:37 PM <DIR> .
09/26/2024 08:37 PM <DIR> ..
09/26/2024 08:37 PM 0 access.log
09/26/2024 08:37 PM 341 error.log
2 File(s) 341 bytes
2 Dir(s) 21,005,381,632 bytes free
C:\Users\LENOVO>docker exec -it namenode /bin/bash
[root@7be1fa5ce64c /]# echo '
> import datetime
>
> # Open the log file in append mode
> with open("/hadoop/logs/nginx_access.log", "a") as file:
> # Write a sample log entry with a timestamp
> file.write("Log entry at {}\\n".format(datetime.datetime.now()))
> ' > /hadoop/append_log.py
[root@7be1fa5ce64c /]# python /hadoop/append_log.py
[root@7be1fa5ce64c /]# cat /hadoop/logs/nginx_access.log
Log entry at 2024-09-26 19:42:10.082669\n[root@7be1fa5ce64c /]#
Type here to search
62°F
FRA
8:42 PM
9/26/2024

Challenges and conclusion

Throughout this project, I encountered several challenges, particularly in locating the necessary log files. The process involved installing and uninstalling Nginx multiple times, which was compounded by technical difficulties that necessitated a switch back to using the terminal. I couldn’t find the log files easily; I had to install and uninstall nginx and then switch again to terminal.

The following picture is a bug of terminal that I have faced, for hits reason I switched to power shell, where I could past eeasily command in case I made mistakes



Despite the obstacles, this assignment has been instrumental in enhancing my understanding of Hadoop's operational environment and its deployment using Docker. I successfully navigated through various configurations of Namenodes and Datanodes, utilized the command line and web interfaces to extract detailed file system information, and managed real-world data with Nginx logs.

References:

1. Etextbook: Big Data: Concepts, Technology, and Architecture by Balamurugan Balusamy, Nandhini Abirami R, Seifedine Kadry, Amir H. Gandomi Links to an external site.

   "Using Hadoop in Standalone Mode," DataFlair - Standalone Mode Hadoop, 2022.

2. https://data-flair.training/blogs/hadoop-tutorials-home/
3. Creating a Hadoop Pseudo-Distributed Environment | by District Data Labs | District Insights | Medium
4. Hadoop - Different Modes of Operation - GeeksforGeeks
5. **Hortonworks unveils roadmap to make Hadoop cloud-native | ZDNET**
6. **youtube.com/watch?v=hLnB0uzGvDI**
7. **Hadoop&cie - 02 - Installation de Hadoop avec Docker (youtube.com)**