



# Hadoop deployment and use HDFS

## **School of Science and Engineering**

Meriem Lmoubariki

Dr. Tajjeeddine Rachidi

Al Akhawayn University in Ifrane

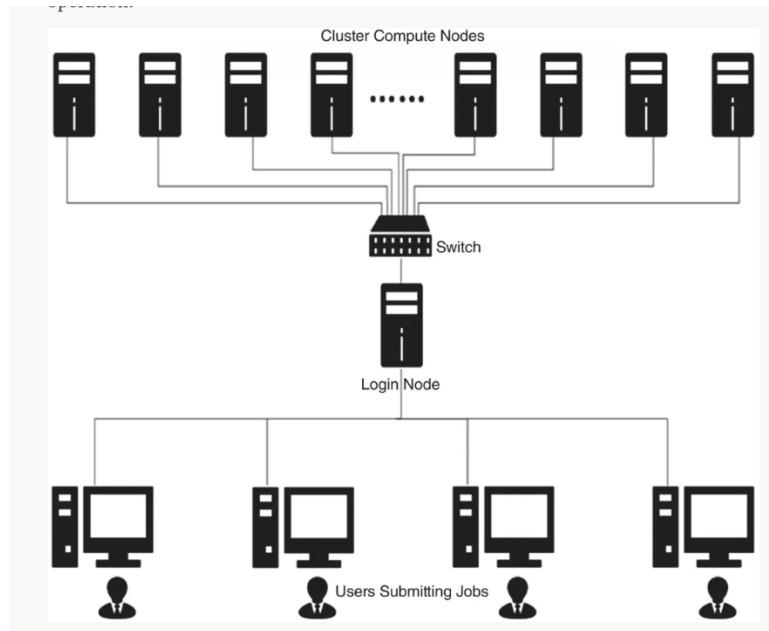
September 20, 2025

# Table of Contents

<b>1. Introduction to Hadoop Deployment -</b>	p. 1
<b>2. Task 1: Deployment Strategies for Hadoop Clusters -</b>	p. 2
○ a. Overview of the different ways to deploy a Hadoop cluster. -	p. 2
○ b. Pros and cons in a comparative table format. -	p. 3
<b>3. Task 2: Docker Integration and Hadoop Image Deployment -</b>	p. 4
○ a. Steps to get acquainted with Docker. -	p. 4
○ b. Selection and deployment of a Docker Hadoop image. -	p. 5
<b>4. Task 3: Configuration Insights of Namenode and Datanode -</b>	p. 6
○ a. Locating fsimage and edit logs on the Namenode. -	p. 6
○ b. Identifying where the blocks are stored on the Datanode. -	p. 7
<b>5. Task 4: Command Line Mastery for HDFS Information Extraction -</b>	p. 8
○ a. Using Hadoop command line tools to extract information about HDFS. -	p. 8
<b>6. Task 5: Web Interface Utilization for Detailed HDFS Insights -</b>	p. 9
○ a. Extracting information via Hadoop's web interface. -	p. 9
<b>7. Task 6: Real-World Data Manipulation with Nginx Logs -</b>	p. 10
○ a. Writing and appending Nginx log data into HDFS. -	p. 10
<b>8. Conclusion -</b>	p. 11
<b>9. References -</b>	p. 12

## 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:** A 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
<b>Standalone Mode</b>	- 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.
<b>Pseudo-Distributed Mode</b>	- 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.

<b>Fully Distributed Mode</b>	<ul style="list-style-type: none"> <li>- Uses many computers, so it's powerful and reliable.</li> <li>- Can grow as needed by adding more machines.</li> </ul>	<ul style="list-style-type: none"> <li>- Costs a lot to set up and keep running.</li> <li>- Needs a lot of technical skill to manage.</li> </ul>	Large-scale data processing in enterprise environments.
<b>Cloud-Based Deployment</b>	<ul style="list-style-type: none"> <li>- Can adjust to needs quickly, scaling up or down.</li> <li>- Less work to maintain as the service provider handles it.</li> </ul>	<ul style="list-style-type: none"> <li>- Security concerns since data is online.</li> <li>- Needs constant internet.</li> </ul>	Businesses needing flexible computing resources without major upfront investment.
<b>Hybrid Deployment</b>	<ul style="list-style-type: none"> <li>- Best of both worlds: control over critical data and cloud flexibility.</li> <li>- Can plan for disasters better</li> </ul>	<ul style="list-style-type: none"> <li>- Hard to manage because it combines two systems.</li> <li>- Can get expensive depending on setup.</li> </ul>	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.

```
C:\Users\LENOVO>docker pull harisekhon/hadoop
Using default tag: latest
latest: Pulling from harisekhon/hadoop
d9aaaf4d82f24: Pull complete
71e193c229b6: Pull complete
34e052ae12c1: Pull complete
7c28f3b3ed5b: Pull complete
b9aeb45a846c: Pull complete
20d3342cd6a7: Pull complete
96ad78d93ff88: Pull complete
39f02a9b4821: Pull complete
934c7436ce6e: Pull complete
f4001b2b79b: Pull complete
ae9fffa67139: Pull complete
Digest: sha256:6c2668f5e59d4b870352cf52f1bcd75945eebe88fb81a1ea3df2464a65951ee6
Status: Downloaded newer image for harisekhon/hadoop:latest
docker.io/harisekhon/hadoop:latest

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview harisekhon/hadoop
```

## Task 3: Configuration Insights of Namenode and Datanode

I have created two different containers.

```
Command Prompt
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LENOVO>docker run -d --name namenode -p 9870:9870 -p 9000:9000 harisekhon/hadoop
docker: error during connect: Head "http://%2F%2Fpipe%2FdockerDesktopLinuxEngine/_ping": open //./pipe/dockerDesktopLinuxEngine: The system cannot find the file specified.
See 'docker run --help'.

C:\Users\LENOVO>docker pull harisekhon/hadoop
Using default tag: latest
latest: Pulling from harisekhon/hadoop
d9aaf7d82f2d4: Pull complete
71e191c229b6: Pull complete
34e052ae12c1: Pull complete
7c28fb3b3ed5b: Pull complete
b9aeb5a5a846c: Pull complete
29d3342cda07: Pull complete
96ad78d3f88: Pull complete
39ff02a9b4821: Pull complete
934c7436c66: Pull complete
f4001b22b79b: Pull complete
ae9ff6a67139: Pull complete
Digest: sha256:6c266875e594d870352cf52f1bcd75945eebe88fb81alea3df2464a65951ee6
Status: Downloaded newer image for harisekhon/hadoop:latest
docker.io/harisekhon/hadoop:latest

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview harisekhon/hadoop

C:\Users\LENOVO>docker run -d --name namenode -p 9870:9870 -p 9000:9000 harisekhon/hadoop
76771856ba02e57f5c5dceae6f913e8838a8a057aee0a365ed800ce973575fede

C:\Users\LENOVO>docker run -d --name datanode -p 9864:9864 harisekhon/hadoop
d59e089b71cc2f9b6da0b8c41f69109ee38418967b5d2d15ff576ebc2185116d

C:\Users\LENOVO>docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
d59e089b71cc harisekhon/hadoop "/bin/sh -c '/entryp..." 11 seconds ago Up 10 seconds 8020/tcp, 8042/tcp, 8088/tcp, 9000/tcp, 10020/tcp, 19888/tcp, 50010/tcp, 50020/tcp, 50070/tcp, 50075/tcp, 50090/tcp, 0.0.0.0:9864->9864/tcp
76771856ba02 harisekhon/hadoop "/bin/sh -c '/entryp..." 21 seconds ago Up 20 seconds 8020/tcp, 8042/tcp, 0.0.0.0:9000->9000/tcp, 8088/tcp, 10020/tcp, 19888/tcp, 50010/tcp, 50020/tcp, 50070/tcp, 50075/tcp, 50090/tcp, 0.0.0.0:9870->9870/tcp
namenode

C:\Users\LENOVO>
```

The screenshot shows the Docker Desktop application window. On the left, there is a sidebar with icons for Containers, Images, Volumes, Builds, Docker Scout, and Extensions. The main area is titled 'Containers' and shows a table of running containers. The table has columns for Name, Image, Status, Port(s), CPU (%), Last started, and Actions. Two containers are listed:

Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
namenode	harisekhon/hadoop:latest	Running	9000:9000	1.34%	8 minutes ago	[Stop, Edit, Delete]
datanode	harisekhon/hadoop:latest	Running	9864:9864	1.07%	8 minutes ago	[Stop, Edit, Delete]

Below the table, it says 'Showing 2 items'. At the bottom of the interface, there are sections for 'Walkthroughs' featuring 'Multi-container applications' and 'Containerize your application', both with a duration of 8 mins. The taskbar at the bottom shows various system icons and the date/time as 9/26/2024 2:33 PM.

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.

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

## Data blocks

```
C:\Users\LENOVO>docker exec -it datanode /bin/bash
[root@4590a54a7c0e ~]# ls /tmp/hadoop-root/dfs/data/current
ip-194-70-34-28-172.17.0.2-1$10154331033 VERSION
[root@4590a54a7c0e ~]# 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%2Fmydata%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 2024
/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[172.17.0.3:50010,DS-a8a8726a-57df-4bb8-826c-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

```
[root@7be1fa5ce64c ~]# /hadoop-2.8.2/bin/hdfs oiv -l /tmp/hadoop-root/dfs/name/current/fsimage_00000000000000000004 -o /tmp/hdfs_output.xml -p XML
[root@7be1fa5ce64c ~]# cat /tmp/hdfs_output.xml
<xml version="1.0">
<fsimage><version>63</version><layoutVersion>1</layoutVersion><onDiskVersion>1</onDiskVersion><oivRevision>66c47f2a01ad9637879e95f80c41f798373828fb</oivRevision></version>
<namespacedId>43579329</namespacedId><genstampV1>10800</genstampV1><genstampV2>10800</genstampV2><genstampV1limit>0</genstampV1limit><lastAllocatedBlockId>1073741824</lastAllocatedBlockId><txid>4</txid></NameSection>
<inodeSection><lastNodeID>16385</lastNodeID><numInodes>1</numInodes><inode><id>16385</id><type>DIRECTORY</type><name></name><mtime>0</mtime><perm>0755</perm><supergroup>0</supergroup><nsQuota>9223372036854775807</nsQuota><dsQuota>-1</dsQuota></inode></inodeSection>
<INodeReferenceSection><INodeReferenceSection><SnapshotSection><snapshotCounter>0</snapshotCounter><numSnapshots>0</numSnapshots></SnapshotSection>
<INodeDirectorySection><INodeDirectorySection>
<FileHeaderConstructionSection><FileHeaderConstructionSection>
<delegationSection><currentId>4</currentId><tokenSequenceNumber>0</tokenSequenceNumber><numDelegationKeys>0</numDelegationKeys><numTokens>0</numTokens><SecretManagerSection><CacheManagerSection><nextDirectiveId>1</nextDirectiveId></nextDirectiveId></CacheManagerSection></SecretManagerSection></delegationSection>
<numPools>0</numPools><numPoolS>0</numPoolS></CacheManagerSection>
</fsimage>
[root@7be1fa5ce64c ~]#
```

```
[root@7be1fa5ce64c ~]# /hadoop-2.8.2/bin/hdfs oev -p stats -l /tmp/hadoop-root/dfs/name/current/edits_00000000000000000003-00000000000000000004 -o /tmp/edits.stats
[root@7be1fa5ce64c ~]# cat /tmp/edits.stats
VERSION : -63
OP_ADD : ( 0): null
OP_REMOVE_OLD : ( 1): null
OP_DELETE : ( 2): null
OP_MKDIR : ( 3): null
OP_SET_REPLICATION : ( 4): null
OP_DATANODE_ADD : ( 5): null
OP_DATANODE_REMOVE : ( 6): null
OP_SET_PERMISSIONS : ( 7): null
OP_SET_OWNER : ( 8): null
OP_CLOSE : ( 9): null
OP_SET_GENSTAMP_V1 : ( 10): null
OP_SET_NS_QUOTA : ( 11): null
OP_ERASE_NS_QUOTA : ( 12): null
OP_TIMES : ( 13): null
OP_SET_QUOTA : ( 14): null
OP_RENAME : ( 15): null
OP_CONCAT_DELETE : ( 16): null
OP_SYMBOLIC : ( 17): null
OP_RENEW_DELEGATION_TOKEN : ( 18): null
OP_RENEW_DELEGATION_TOKEN : ( 19): null
OP_CANCEL_DELEGATION_TOKEN : ( 20): null
OP_UPDATE_MASTER_KEY : ( 21): null
OP_RELASSIGNLEASE : ( 22): null
OP_END_LOG_SEGMENT : ( 23): 1
OP_ALLOCATE_BLOCK : ( 24): null
OP_UPDATE_BLOCKS : ( 25): null
OP_CREATE_SNAPSHOT : ( 26): null
OP_DELETE_SNAPSHOT : ( 27): null
OP_RENAME_SNAPSHOT : ( 28): null
OP_ALLOW_SNAPSHOT : ( 29): null
OP_SET_BLOCK_SNAPSHOT : ( 30): null
OP_SET_GENSTAMP_V2 : ( 31): null
OP_ALLOCATE_BLOCK_ID : ( 32): null
OP_ADD_BLOCK : ( 33): null
OP_ADD_CACHE_DIRECTIVE : ( 34): null
OP_REMOVE_CACHE_DIRECTIVE : ( 35): null
OP_SET_BLOCK_POOL : ( 36): null
OP_MOOTIFY_CACHE_POOL : ( 37): null
OP_REMOVE_CACHE_POOL : ( 38): null
OP_MODIFY_CACHE_DIRECTIVE : ( 39): null
OP_SET_ACL : ( 40): null
OP_ROLLING_UPGRADE_START : ( 41): null
OP_ROLLING_UPGRADE_FINALIZE : ( 42): null
OP_SET_XATTR : ( 43): null
OP_REMOVE_XATTR : ( 44): null
OP_SET_STORAGE_POLICY : ( 45): null
OP_TRUNCATE : ( 46): null
OP_APPEND : ( 47): null
OP_SET_BLOCK_BY_STORAGETYPE : ( 48): null
OP_INVALID : (-1): null
```

## Task 5: Web Interface Utilization for Detailed HDFS Insights

The screenshot shows the HDFS Health Overview page from the HDFS web interface. The top navigation bar includes links for Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities.

**Overview** '7be1fa5ce64c:8020' (active)

Started:	Thu Sep 26 18:08:23 +0100 2024
Version:	2.8.2, r66c47f2a01ad9637879e95f80c41f798373828fb
Compiled:	Thu Oct 19 21:39:00 +0100 2017 by jdu from branch-2.8.2
Cluster ID:	CID-c51705a1-9198-4902-8d1f-cd570b90fe18
Block Pool ID:	BP-1947034320-172.17.0.2-1510154331033

**Summary**

Security is off.  
Safemode is off.  
1 files and directories, 0 blocks = 1 total filesystem object(s).  
Heap Memory used 78.31 MB of 220.5 MB Heap Memory, Max Heap Memory is 889 MB.  
Non Heap Memory used 40.26 MB of 40.88 MB Committed Non Heap Memory, Max Non Heap Memory is <unbounded>.

Configured Capacity:	1006.85 GB
DFS Used:	28 KB (0%)
Non DFS Used:	2.16 GB

Type here to search

localhost:9870/dfshealth.html#tab-datanode

Namenode information

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

## Datanode Information

In service: 1 Down: 0 Decommissioned: 0 Decommissioned & dead: 0

### Datanode usage histogram

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 6:09 PM 9/26/2024

localhost:9870/dfshealth.html#tab-startup-progress

Namenode information

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_00000000000000000004 321 B	100%	0 sec
inodes (1/1)	100%	
delegation tokens (0/0)	100%	
cache pools (0/0)	100%	
Loading edits /tmp/hadoop-root/dfs/name/current/edits_0000000000000005-0000000000000005 1 MB (1/1)	100%	0 sec
Saving checkpoint	100%	0 sec
Safe mode	100%	0 sec
awaiting reported blocks (0/0)	100%	

Hadoop, 2017.

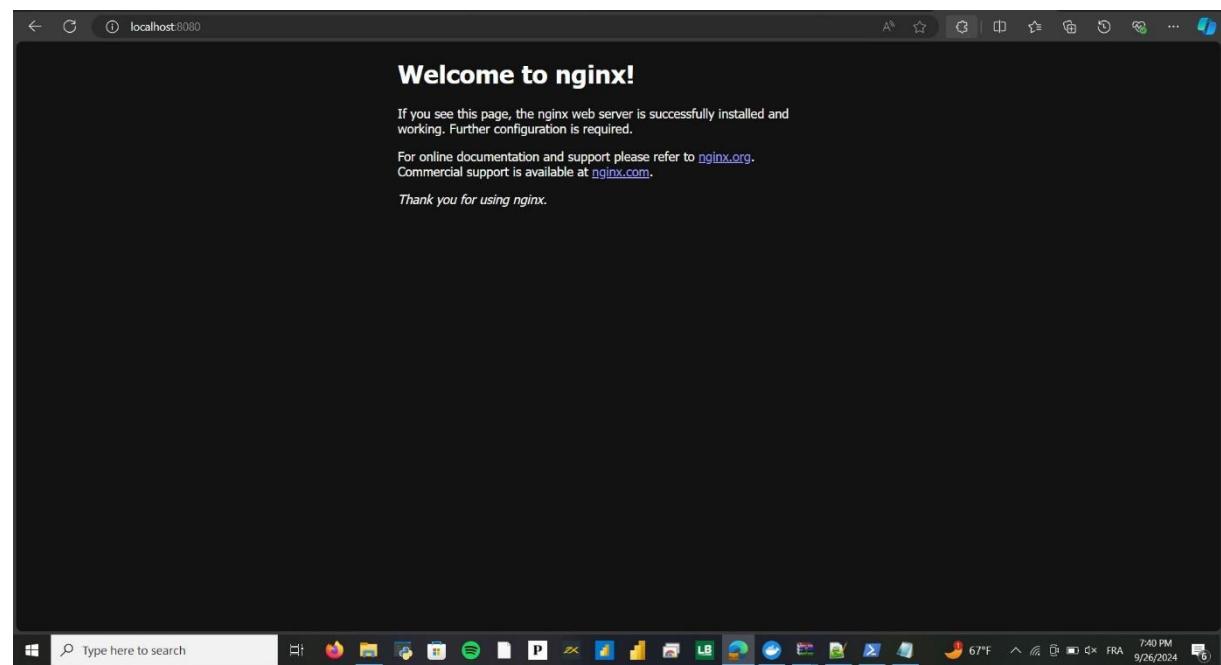
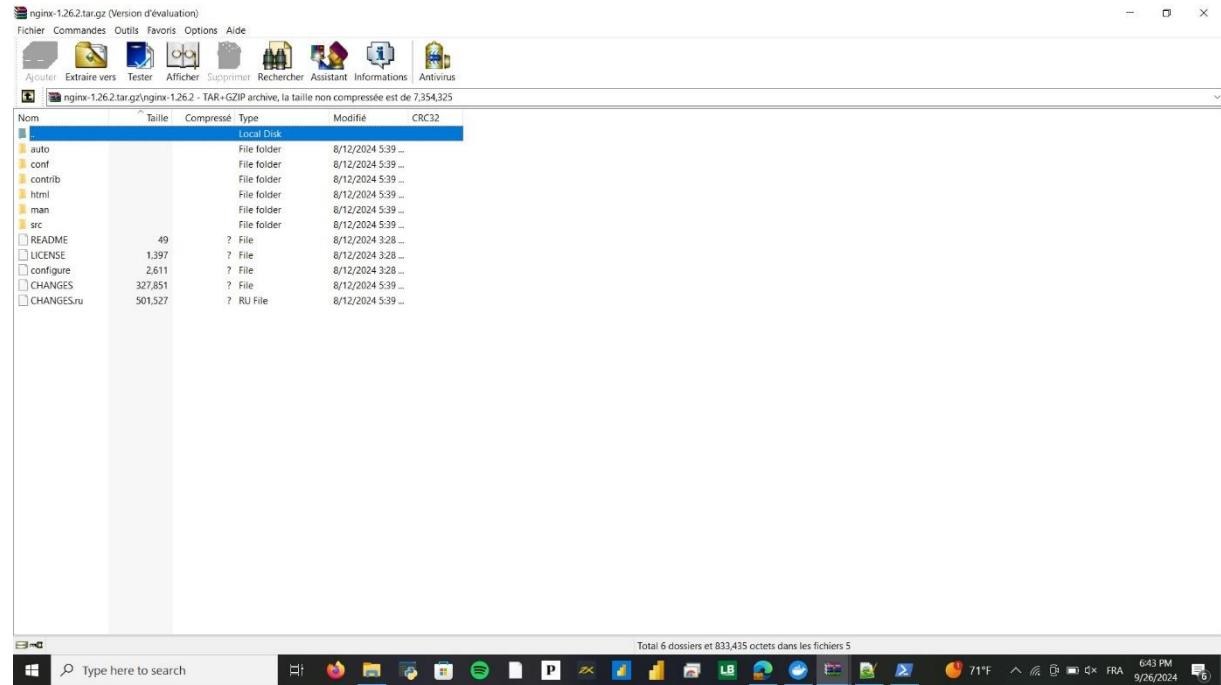
localhost:9870/dfshealth.html#tab-startup-progress

Type here to search

75°F 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-UABICK1: /mnt/c/Users/LENOVO/Downloads/nginx-1.26.2
+ ./configure --prefix=/usr/local/nginx --with-pcre --with-pcre-jit
[...]
|| 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/html' \
|| mkdir -p '/home/meriem/nginx/html'
test -d '/home/meriem/nginx' \
|| cp -rhtml '/home/meriem/nginx'
test -d '/home/meriem/nginx/logs' \
|| cp -rlogs '/home/meriem/nginx/logs'
test -d '/home/meriem/nginx/html' \
|| cp -rhtml '/home/meriem/nginx/html'
make[1]: Leaving directory '/mnt/c/Users/LENOVO/Downloads/nginx-1.26.2'
meriem@DESKTOP-UABICK1: ~ $ SHOME/nginx/sbin/nginx
nginx: [emerg] bind() to 0.0.0.0:80 failed (13: Permission denied)
meriem@DESKTOP-UABICK1: ~ $ /usr/local/nginx/sbin/nginx
nginx: [emerg] open() failed (2: No such file or directory) open() failed (13: Permission denied)
2024/09/26 19:17:37 [emerg] 55946: mkdir() "/usr/local/nginx/client_body_temp" failed (13: Permission denied)
meriem@DESKTOP-UABICK1: ~ $ sudo chown -R $wwwuser /usr/local/nginx
meriem@DESKTOP-UABICK1: ~ $ /usr/local/nginx/sbin/nginx
nginx: [emerg] bind() to 0.0.0.0:80 failed (13: Permission denied)
meriem@DESKTOP-UABICK1: ~ $ /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: '
meriem@DESKTOP-UABICK1: ~ $ vi /usr/local/nginx/conf/nginx.conf
meriem@DESKTOP-UABICK1: ~ $ /usr/local/nginx/sbin/nginx
meriem@DESKTOP-UABICK1: ~ $
```

```
Command Prompt
Using default tag: latest
latest: Pulling from library/nginx
3122471704d5: Download complete
0723edc19c17: Download complete
7fb0f3e0a030: Download complete
2db5fc461e3: Download complete
bf5fa25d0775: Download complete
a2318d6c79ae: Download complete
095d327c79ae: Download complete
Digest: sha256:04ba374043cd2f5c593885ceacddebabd5ca375f9323666f28df5a9710e3
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
59640bd1beef307f53265b3a498481a4dec1f622433f77b38dc2d5575fe0e92837
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\
```

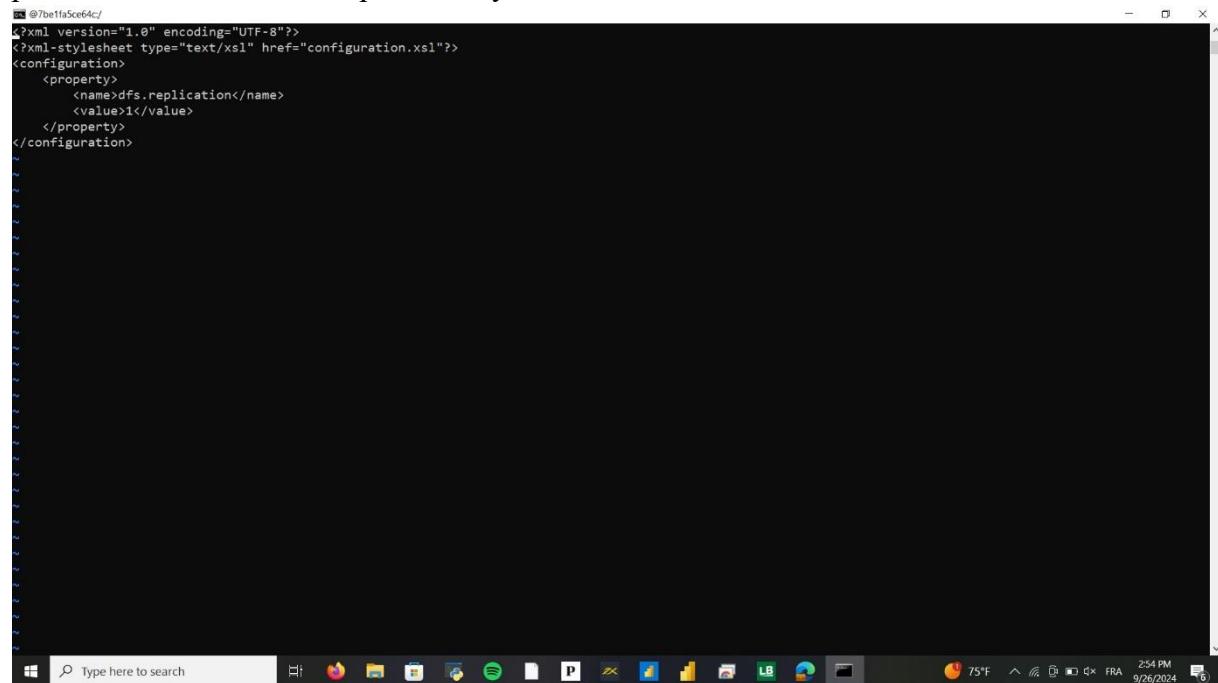
```
git: @7be1fa5ce64c:/  
Using default tag: latest  
1: Pulling repository/.../library/nginx  
1:2247f1704d5: Download complete  
9723edc10c17: Download complete  
7hbefbf0cfb2b: Download complete  
2db5fd4d1e3: Download complete  
bbfaa25d8775: Download complete  
a2318d6c47ec: Download complete  
095d327c79ae: Download complete  
Digest: sha256:d4ba374043cd2fc5c593885c0eaddebabd5ca375f9323666f28df5a9710e3  
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  
59640d1bee307f532663b4a98481a40ec1fb622433f77b38dc2d575fbe0f249  
  
C:\Users\LENOVO\dir C:\nginx\logs  
Volume in drive C has no label.  
Volume Serial Number is FFB0-2837  
  
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 /]#
```

Windows Taskbar: Type here to search, File Explorer, Firefox, File Manager, Task View, Taskbar icons, 62°F, 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

A screenshot of a Windows desktop environment showing a terminal window. The terminal window has a black background and contains a partially visible XML configuration file. The file starts with `<?xml version="1.0" encoding="UTF-8"?>` and includes a `configuration` section with a single property named `dfs.replication` set to the value `1`. Below the XML, there is a series of vertical blue tick marks, likely indicating a stuck cursor or a very slow response from the system. The taskbar at the bottom shows various icons for common applications like File Explorer, Edge, and Spotify. The system tray indicates the date as 9/26/2024, the time as 2:54 PM, and the location as FRA. The temperature is listed as 75°F. A battery icon shows it is currently charging.

```
<?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>
```

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](https://www.youtube.com/watch?v=hLnB0uzGvDI)
7. [Hadoop&cie - 02 - Installation de Hadoop avec Docker \(youtube.com\)](#)