This document is for insisting on Installing HDFS and Spark on Windows

Our Group:
https://github.com/Di-Pan
https://github.com/Yuwei-D
https://github.com/YingqiL6
https://github.com/xiongshenping
https://github.com/HanyanY2

For installing Hadoop components, it is recommended to install and run on Linux like Ubuntu or Unix-like systems (You can use Virtual Machine).
Otherwise, it is very likely to face compatibility issues.
Also, most tutorials on YouTube for HDFS or Spark will teach within Linux or Unix-like systems.

It is possible I made a mistake or missed something, if you are debugging GPT is your(Ours) best friend.
It is also possible, that the information I provided now will be outdated in the future.

Warning

- Before starting, make sure you confirm the version capabilities from Spark, HDFS Java, and Python.
- Debugging for the environment I faced is Java version with HDFS version not compatible.
- In Windows better put all these into the folder that is without any “space” in the folder name, or you will enjoy debugging.
- You may also face the authority issue if you put in disk C (need administrator).

Capability of Windows, Environments

Install Spark from https://spark.apache.org/downloads.html.
For Windows, You have to search from Google, **what is the most recent update of winutils.exe** of hadoop support you can find from Github. Based on that supported version then download the Spark that from https://spark.apache.org/downloads.html or Spark Archive https://archive.apache.org/dist/spark/ .

Winutils links:
https://github.com/robguilarr/spark-winutils-3.3.1/tree/master
https://github.com/steveloughran/winutils

For instance in my case,
I downloaded from https://github.com/steveloughran/winutils for Hadoop 3.0.0 and downloaded https://archive.apache.org/dist/spark/ for Spark 3.0.0.
The main spark components are from https://archive.apache.org/dist/spark/ or https://spark.apache.org/downloads.html.

After downloading, copy the bin folder from the folder that has winutils.exe and copy it to cover the original bin folder that is from Apache Spark.

We need **Python** (better 3.9 or 3.11) so we can run Pyspark in the future.
We need **Java**, it is really important for the Java versions. Better Download for **JDK 8**
(If your Hadoop is lower than 3.0.0) otherwise, JDK 11 may be fine too.
I recommend you confirm the Versions capability with GPT first.

If above all is done set the Environment variables:
Trying not to put Components into the program files folder as I did below, cause I debug them and it took some time.
It is just a reference, no need to copy the same as I did.

System Variables:

PYSPARK_PYTHON	C:\Users\dpan\AppData\Local\Programs\Python\Python39\python.exe
SPARK_HOME	C:\Spark\spark-3.5.3-bin-hadoop3
HADOOP_CONF_DIR	C:\Spark\Hadoop\etc\hadoop
HADOOP_HOME	C:\Spark\Hadoop
JAVA_HOME	C:\Program Files\Java\jdk-1.8

In Path:

```
C:\Spark\spark-3.5.3-bin-hadoop3\bin
%SPARK_HOME%\bin
%HADOOP_HOME%\bin
%JAVA_HOME%\bin
%HADOOP_HOME%\sbin
```

These are the files I have in Hadoop and bin (after copying the winutils.exe):

PC > Local Disk (C:) > Spark > Hadoop > bin

Name	Date modified	Type	Size
container-executor	2017-12-08 2:30 PM	File	348 KB
hadoop	2024-11-12 12:51 AM	File	9 KB
hadoop.cmd	2024-11-12 12:51 AM	Windows Comma...	11 KB
hadoop.dll	2024-11-12 12:51 AM	Application extens...	91 KB
hadoop.exp	2024-11-12 12:51 AM	EXP File	23 KB
hadoop.lib	2024-11-12 12:51 AM	LIB File	37 KB
hadoop.pdb	2024-11-12 12:51 AM	PDB File	491 KB
hdfs	2024-11-12 12:51 AM	File	11 KB
hdfs.cmd	2024-11-12 12:51 AM	Windows Comma...	8 KB
hdfs.dll	2024-11-12 12:51 AM	Application extens...	62 KB
hdfs.exp	2024-11-12 12:51 AM	EXP File	11 KB
hdfs.lib	2024-11-12 12:51 AM	LIB File	353 KB
hdfs.pdb	2024-11-12 12:51 AM	PDB File	355 KB
libwinutils.lib	2024-11-12 12:51 AM	LIB File	1,199 KB
mapred	2024-11-12 12:51 AM	File	6 KB
mapred.cmd	2024-11-12 12:51 AM	Windows Comma...	6 KB
rcc	2024-11-12 12:51 AM	File	2 KB
test-container-executor	2017-12-08 2:30 PM	File	383 KB
winutils.exe	2024-11-12 12:51 AM	Application	110 KB
winutils.pdb	2024-11-12 12:51 AM	PDB File	875 KB
yarn	2024-11-12 12:51 AM	File	11 KB
yarn.cmd	2024-11-12 12:51 AM	Windows Comma...	12 KB

is PC > Local Disk (C:) > Spark > Hadoop >

Name
bin
data
etc
include
lib
libexec
logs
sbin
share
LICENSE.txt
NOTICE.txt
README.txt

After all is set, you need to set the environment file in etc\hadoop (better ask GPT for help with debugging)
File: (If you don't have them you can create one)
Hadoop-env.cmd (right-click edit it)
Yarn-env.cmd (If you are going to use Yarn)

How to confirm they are all Set:

Go to this path and run cmd with
cd C:\Spark\Hadoop\bin
Hadoop version

```
Command Prompt
Microsoft Windows [Version 10.0.19045.5131]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dpan>cd C:\Spark\Hadoop\bin

C:\Spark\Hadoop\bin>hadoop version
Hadoop 3.0.0
Source code repository https://git-wip-us.apache.org/repos/asf/hadoop.git -r c25427ceca461ee979d30edd7a4b0f50718e6533
Compiled by andrew on 2017-12-08T19:16Z
Compiled with protoc 2.5.0
From source with checksum 397832cb5529187dc8cd74ad54ff22
This command was run using /C:/Spark/Hadoop/share/hadoop/common/hadoop-common-3.0.0.jar

C:\Spark\Hadoop\bin>
```

Then your HDFS should be all set, if not most likely it is related to the JAVA version or environmental variables.

Pyspark and Spark installed.

```
C:\Users\dpan>pyspark
Python 3.9.2rc1 (tags/v3.9.2rc1:4064156, Feb 17 2021, 11:25:18) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.5.3
      /_/

Using Python version 3.9.2rc1 (tags/v3.9.2rc1:4064156, Feb 17 2021 11:25:18)
Spark context Web UI available at http://192.168.250.33:4040
Spark context available as 'sc' (master = local[*], app id = local-1733081089596).
SparkSession available as 'spark'.
>>>
```

Java

```
Command Prompt
Microsoft Windows [Version 10.0.19045.5131]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dpan>Java -version
java version "1.8.0_431"
Java(TM) SE Runtime Environment (build 1.8.0_431-b26)
Java HotSpot(TM) 64-Bit Server VM (build 25.431-b26, mixed mode)
```

How to connect with other workers in Spark?

As a Master, you need to wake up your master machine first.

Master:

Spark-class org.apache.spark.deploy.master.Master

```
Select Command Prompt - spark-class  org.apache.spark.deploy.master.Master
Microsoft Windows [Version 10.0.19045.5131]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dpan>spark-class org.apache.spark.deploy.master.Master
Using Spark's default log4j profile: org/apache/spark/log4j2-defaults.properties
24/12/01 14:31:24 INFO Master: Started daemon with process name: 6916@DESKTOP-NJ6TN44
24/12/01 14:31:24 INFO SecurityManager: Changing view acls to: dpan
24/12/01 14:31:24 INFO SecurityManager: Changing modify acls to: dpan
24/12/01 14:31:24 INFO SecurityManager: Changing view acls groups to:
24/12/01 14:31:24 INFO SecurityManager: Changing modify acls groups to:
24/12/01 14:31:24 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view perm
issions: dpan; groups with view permissions: EMPTY; users with modify permissions: dpan; groups with modify permissions:
 EMPTY
24/12/01 14:31:24 INFO Utils: Successfully started service 'sparkMaster' on port 7077.
24/12/01 14:31:24 INFO Master: Starting Spark master at spark://192.168.250.33:7077
24/12/01 14:31:24 INFO Master: Running Spark version 3.5.3
24/12/01 14:31:25 INFO JettyUtils: Start Jetty 0.0.0.0:8080 for MasterUI
24/12/01 14:31:25 INFO Utils: Successfully started service 'MasterUI' on port 8080.
24/12/01 14:31:25 INFO MasterWebUI: Bound MasterWebUI to 0.0.0.0, and started at http://192.168.250.33:8080
24/12/01 14:31:25 INFO Master: I have been elected leader! New state: ALIVE
```

Not secure 192.168.250.33:8080

Spark 3.5.3 **Spark Master at spark://192.168.250.33:7077**

URL: spark://192.168.250.33:7077
Alive Workers: 0
Cores in use: 0 Total, 0 Used
Memory in use: 0.0 B Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (0)

Worker Id	Address	State	Cores	Memory

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time

As a worker connect to the master machine:

You need to **turn off the public network firewall on Windows so others can ping you**

ping other computer's IP to verify the connection(IPv4 address)

cd C:\spark\spark\bin

spark-class org.apache.spark.deploy.worker.Worker spark://<<Master's IP address>>

```
Administrator: Command Pro

C:\Users\Administrator>cd C:\Program Files\Spark\spark-3.5.3-bin-hadoop3\bin

C:\Program Files\Spark\spark-3.5.3-bin-hadoop3\bin>spark-class org.apache.spark.deploy.worker.Worker spark://192.168.250.33:7077
Using Spark's default log4j profile: org/apache/spark/log4j2-defaults.properties
24/12/01 14:43:11 INFO Worker: Started daemon with process name: 33568@DANIEL
24/12/01 14:43:16 INFO SecurityManager: Changing view acls to: Administrator
24/12/01 14:43:16 INFO SecurityManager: Changing modify acls to: Administrator
24/12/01 14:43:16 INFO SecurityManager: Changing view acls groups to:
24/12/01 14:43:16 INFO SecurityManager: Changing modify acls groups to:
24/12/01 14:43:16 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Administrator; groups with view
 permissions: EMPTY; users with modify permissions: Administrator; groups with modify permissions: EMPTY
24/12/01 14:43:17 INFO Utils: Successfully started service 'sparkWorker' on port 7802.
24/12/01 14:43:17 INFO Worker: Worker decommissioning not enabled.
24/12/01 14:43:18 INFO Worker: Starting Spark worker 192.168.216.1:7802 with 12 cores, 30.9 GiB RAM
24/12/01 14:43:18 INFO Worker: Running Spark version 3.5.3
24/12/01 14:43:18 INFO Worker: Spark home: C:\Program Files\Spark\spark-3.5.3-bin-hadoop3
24/12/01 14:43:18 INFO ResourceUtils: ==============================================================
24/12/01 14:43:18 INFO ResourceUtils: No custom resources configured for spark.worker.
24/12/01 14:43:18 INFO ResourceUtils: ==============================================================
24/12/01 14:43:18 INFO JettyUtils: Start Jetty 0.0.0.0:8081 for WorkerUI
24/12/01 14:43:18 INFO Utils: Successfully started service 'WorkerUI' on port 8081.
24/12/01 14:43:18 INFO WorkerWebUI: Bound WorkerWebUI to 0.0.0.0, and started at http://DANIEL:8081
24/12/01 14:43:18 INFO Worker: Connecting to master 192.168.250.33:7077...
24/12/01 14:43:18 INFO TransportClientFactory: Successfully created connection to /192.168.250.33:7077 after 71 ms (0 ms spent in bootstraps)
24/12/01 14:43:18 INFO Worker: Successfully registered with master spark://192.168.250.33:7077
```

Spark Master at spark://192.168.250.33:7077

URL: spark://192.168.250.33:7077
Alive Workers: 1
Cores in use: 12 Total, 0 Used
Memory in use: 30.9 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (1)

Worker Id	Address	State	Cores	Memory
worker-20241201144317-192.168.216.1-7802	192.168.216.1:7802	ALIVE	12 (0 Used)	30.9 GiB (0.0 B Used)

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User

We have a worker now.

Don’t forget to turn on your firewall if you are not working on the project for a while.

Spark Master at spark://192.168.250.33:7077

URL: spark://192.168.250.33:7077
Alive Workers: 1
Cores in use: 12 Total, 0 Used
Memory in use: 30.9 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (1)

Worker Id	Address	State	Cores	Memory
worker-20241201144317-192.168.216.1-7802	192.168.216.1:7802	ALIVE	12 (0 Used)	30.9 GiB (0.0 B Used)

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User

Start HDFS:

Cmd:

```
C:\Users\dpan>start-dfs.cmd
```

You can see the namenode and datanode are up. (2 windows pop up)

```
Apache Hadoop Distribution - hadoop  namenode
DEPRECATED: Use of this script to execute hdfs co
Instead use the hdfs command for it.
2024-12-01 14:54:50,830 INFO namenode.NameNode: S
/************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = DESKTOP-NJ6TN44/192.168.56.
STARTUP_MSG:   args = []
STARTUP_MSG:   version = 3.0.0
STARTUP_MSG:   classpath = C:\Spark\Hadoop\etc\ha
ommon\lib\accessors-smart-1.2.jar;C:\Spark\Hadoop
mon\lib\avro-1.7.7.jar;C:\Spark\Hadoop\share\hado
common\lib\commons-cli-1.2.jar;C:\Spark\Hadoop\sh
p\common\lib\commons-collections-3.2.2.jar;C:\Spa
adoop\share\hadoop\common\lib\commons-configurati
r;C:\Spark\Hadoop\share\hadoop\common\lib\commons
.jar;C:\Spark\Hadoop\share\hadoop\common\lib\comm
ath3-3.1.1.jar;C:\Spark\Hadoop\share\hadoop\commo
or-client-2.12.0.jar;C:\Spark\Hadoop\share\hadoop
ommon\lib\curator-recipes-2.12.0.jar;C:\Spark\Had
\common\lib\guava-11.0.2.jar;C:\Spark\Hadoop\shar
```

```
Apache Hadoop Distribution - hadoop  datanode
DEPRECATED: Use of this script to execute hdfs c
Instead use the hdfs command for it.
2024-12-01 14:54:50,830 INFO datanode.DataNode:
/************************************************
STARTUP_MSG: Starting DataNode
STARTUP_MSG:   host = DESKTOP-NJ6TN44/192.168.56
STARTUP_MSG:   args = []
STARTUP_MSG:   version = 3.0.0
STARTUP_MSG:   classpath = C:\Spark\Hadoop\etc\h
ommon\lib\accessors-smart-1.2.jar;C:\Spark\Hadoo
mon\lib\avro-1.7.7.jar;C:\Spark\Hadoop\share\had
common\lib\commons-cli-1.2.jar;C:\Spark\Hadoop\s
p\common\lib\commons-collections-3.2.2.jar;C:\Sp
adoop\share\hadoop\common\lib\commons-configurat
r;C:\Spark\Hadoop\share\hadoop\common\lib\commor
.jar;C:\Spark\Hadoop\share\hadoop\common\lib\com
ath3-3.1.1.jar;C:\Spark\Hadoop\share\hadoop\comm
or-client-2.12.0.jar;C:\Spark\Hadoop\share\hadoo
ommon\lib\curator-recipes-2.12.0.jar;C:\Spark\Ha
\common\lib\guava-11.0.2.jar;C:\Spark\Hadoop\sha
```

HDFS (NameNode UI):

- Open a browser and visit: http://localhost:9870

localhost:9870/dfshealth.html#tab-overview

Hadoop | Overview | Datanodes | Datanode Volume Failures | Snapshot | Startup Progress | Utilities

Overview 'localhost:9000' (active)

Started:	Sun Dec 01 14:54:56 -0500 2024
Version:	3.0.0, rc25427ceca461ee979d30ed[redacted]
Compiled:	Fri Dec 08 14:16:00 -0500 2017 by andrew from branch-3.0.0
Cluster ID:	CID-42abb245-d47a[redacted]
Block Pool ID:	BP-118932[redacted]-192.168[redacted]

Summary

Security is off.

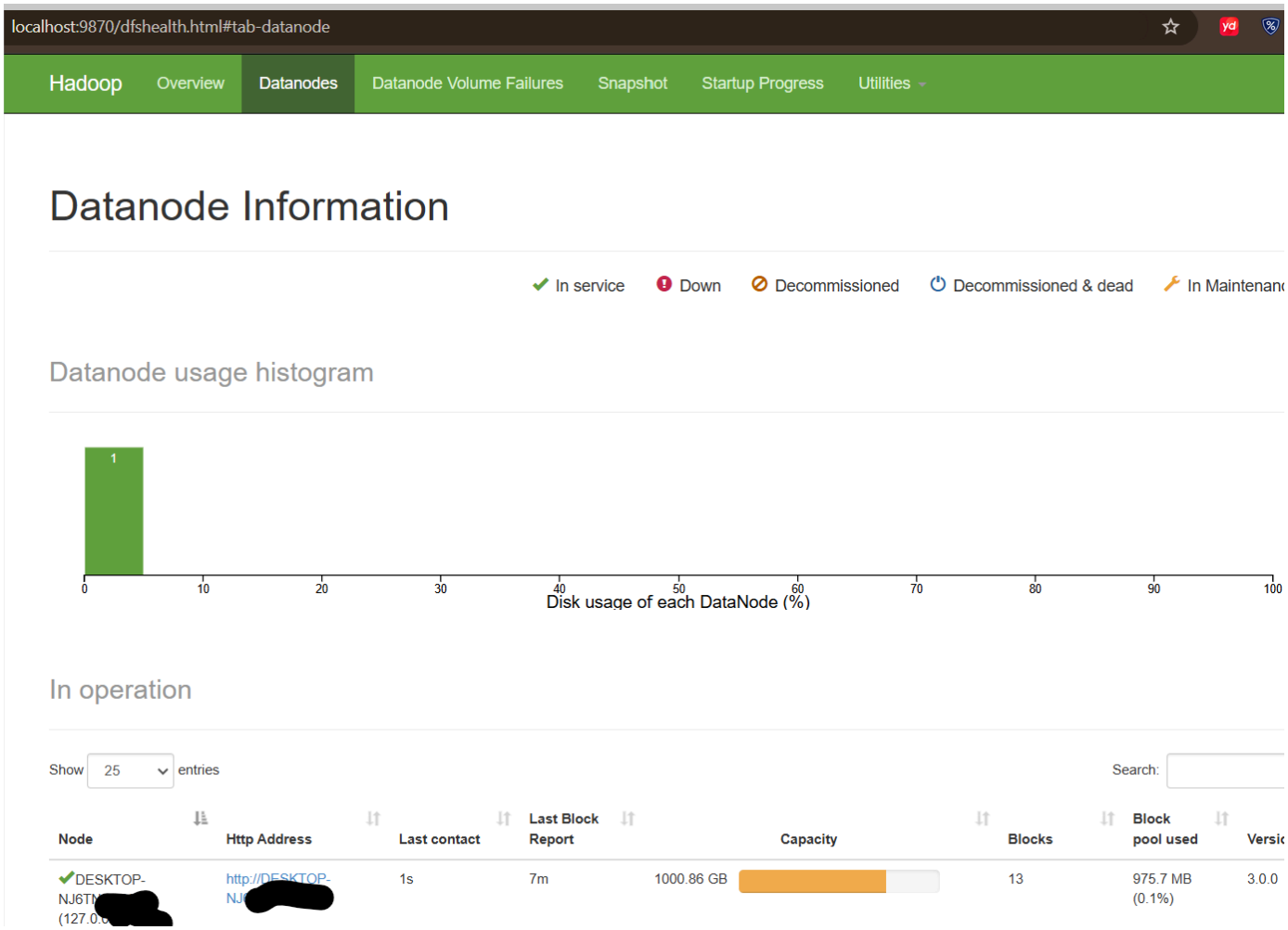
Safemode is off.

21 files and directories, 13 blocks = 34 total filesystem object(s).

Heap Memory used 86.43 MB of 244 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 49.33 MB of 50.17 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	1000.86 GB
DFS Used:	975.7 MB (0.1%)
Non DFS Used:	734.65 GB
DFS Remaining:	265.26 GB (26.5%)
Block Pool Used:	975.7 MB (0.1%)

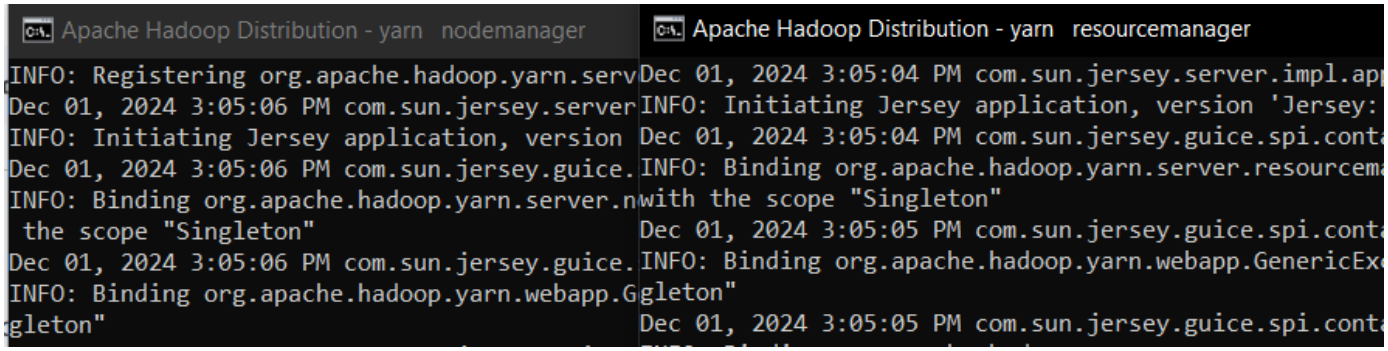


Start Yarn:

Cmd:

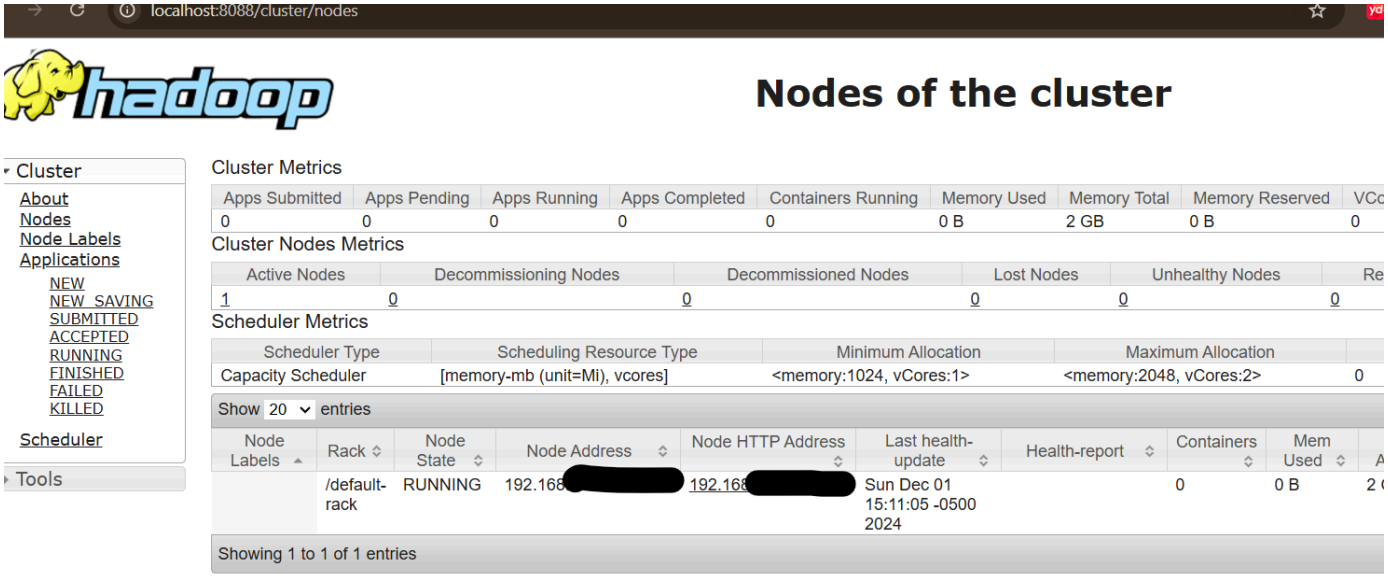
```
C:\Users\dpan>start-yarn.cmd
starting yarn daemons
```

You can see the nodemanager and resource manager are up. (2 windows pop up)



YARN (ResourceManager UI):

- Visit: http://localhost:8088



If not surprisingly you should see around 5,6 cmd (or more) are opening **1 Spark, 2 HDFS, 2 Yarn, and 1 is for calling HDFS and Yarn cmd that you opened**.