

Practical Journal
Big Data Analytics

Submitted By: Mohammad Tabish Mukaddam

Submitted To: Prof. Swati Maurya

Seat: 31031523015

MSc CS-II

A.Y. 2024-25

Department of Computer Science
Somaiya Vidyavihar University
SK Somaiya College

| Sr. No. | Title |
|---------|------------------------------------|
| 1 | Installation of Hadoop |
| 2 | Spark SQL |
| 3 | Spark GraphX |
| 4 | PySpark |
| 5 | Download and installation of HBase |

Practical 1:

Aim: Installation of Hadoop

Step 1: Download Binary File for Windows <https://hadoop.apache.org/releases.html>



We suggest the following location for your download:

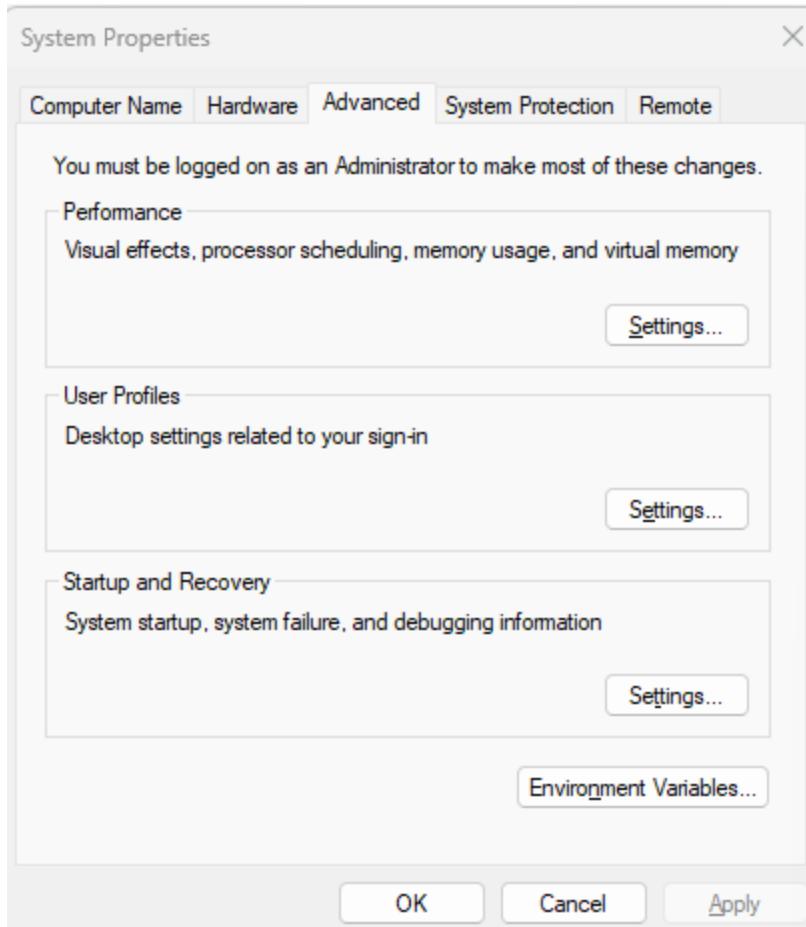
<https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz>

Alternate download locations are suggested below.

It is essential that you [verify the integrity](#) of the downloaded file using the PGP signature ([.asc](#) file) or a hash ([.md5](#) or [.sha*](#) file).

Step 2: Extract the files in C drive .

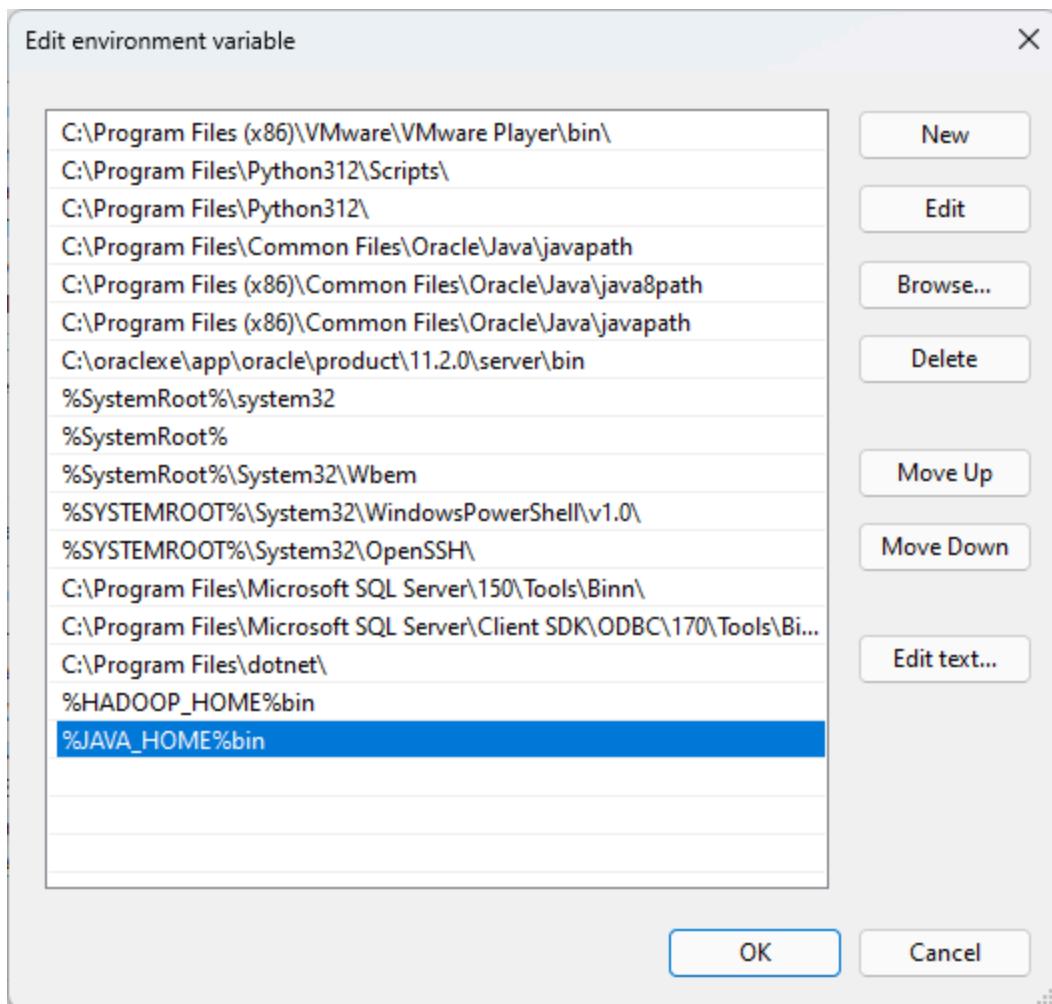
Step 3: Edit Environment Variables.



Step 4: Under System Variables click “New” and set “Variable name” as JAVA_HOME and “Variable value” as the path of your JAVA JDK.

Step 5: Similarly add “HADOOP_HOME” variable and download the bin folder from the below link

<https://drive.google.com/drive/folders/1iURNbow2IglhAhSy3sfY5xxVfAg33NBW>



Step 6: Extract the bin archive and replace the bin folder in Hadoop folder with the bin folder in this archive.

| Name | Date modified | Type | Size |
|-------------------------|------------------|----------------------|----------|
| hadoop | 07-07-2020 00:16 | File | 9 KB |
| hadoop | 07-07-2020 00:16 | Windows Comma... | 12 KB |
| hadoop.dll | 01-08-2020 17:58 | Application exten... | 85 KB |
| hadoop.exp | 01-08-2020 17:58 | Exports Library File | 20 KB |
| hadoop.lib | 01-08-2020 17:58 | Object File Library | 33 KB |
| hadoop | 01-08-2020 17:58 | PDB File | 684 KB |
| hdfs | 14-08-2023 21:39 | File | 12 KB |
| hdfs | 14-08-2023 21:39 | Windows Comma... | 8 KB |
| libwinutils.lib | 01-08-2020 17:58 | Object File Library | 1,283 KB |
| mapred | 14-08-2023 21:39 | File | 7 KB |
| mapred | 14-08-2023 21:39 | Windows Comma... | 7 KB |
| oom-listener | 14-08-2023 21:39 | File | 29 KB |
| test-container-executor | 07-07-2020 01:03 | File | 819 KB |
| winutils | 14-08-2023 21:39 | Application | 110 KB |
| winutils | 01-08-2020 17:58 | PDB File | 1,156 KB |
| yarn | 14-08-2023 21:39 | File | 13 KB |
| yarn | 14-08-2023 21:39 | Windows Comma... | 13 KB |

Step 7: Check if “winutils” is working. If you get any dll error then download that dll and paste in the Windows -> System32 folder.

Step 8: Create a data folder in the hadoop home directory and add the folders datanode and namenode to it.

| Name | Date modified | Type | Size |
|----------|------------------|-------------|------|
| namenode | 31-07-2024 19:29 | File folder | |
| datanode | 31-07-2024 19:29 | File folder | |

9: Add the following path to “Path” under “System Variables” in “Edit Environment Variables”
C:\hadoop-3.4.0\sbin

10: Make the changes to the following files as given, in “etc/hadoop” folder of hadoop home.

| | | | |
|----------------------------------|------------------|-----------------------|-------|
| core-site.xml | 31-07-2024 08:44 | xmlfile | 1 KB |
| hadoop-env | 04-03-2024 12:06 | Windows Comma... | 4 KB |
| hadoop-env | 04-03-2024 13:35 | SH Source File | 17 KB |
| hadoop-metrics2 | 04-03-2024 12:06 | Properties Source ... | 4 KB |
| hadoop-policy.xml | 04-03-2024 12:06 | xmlfile | 14 KB |
| hadoop-user-functions.sh.example | 04-03-2024 12:06 | EXAMPLE File | 4 KB |
| hdfs-rbf-site.xml | 04-03-2024 12:37 | xmlfile | 1 KB |
| hdfs-site.xml | 04-03-2024 12:13 | xmlfile | 1 KB |
| httpfs-env | 04-03-2024 12:22 | SH Source File | 2 KB |
| httpfs-log4j | 04-03-2024 12:22 | Properties Source ... | 2 KB |
| httpfs-site.xml | 04-03-2024 12:22 | xmlfile | 1 KB |
| kms-acls.xml | 04-03-2024 12:08 | xmlfile | 4 KB |
| kms-env | 04-03-2024 12:08 | SH Source File | 2 KB |
| kms-log4j | 04-03-2024 12:08 | Properties Source ... | 2 KB |
| kms-site.xml | 04-03-2024 12:08 | xmlfile | 1 KB |
| log4j | 04-03-2024 12:06 | Properties Source ... | 15 KB |
| mapred-env | 04-03-2024 13:00 | Windows Comma... | 1 KB |
| mapred-env | 04-03-2024 13:00 | SH Source File | 2 KB |
| mapred-queues.xml.template | 04-03-2024 13:00 | TEMPLATE File | 5 KB |
| mapred-site.xml | 04-03-2024 13:00 | xmlfile | 1 KB |

core-site.xml

```

<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>

```

mapred-site.xml

```

<configuration>
  <property>
    <name>mapred.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>

```

```

        </property>
</configuration>

hdfs-site.xml

<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>

    <property>
        <name>dfs.namenode.name.dir</name>
        <value>C:\hadoop-3.4.0\data\namenode</value>
    </property>

    <property>
        <name>dfs.datanode.data.dir</name>
        <value>C:\hadoop-3.4.0\data\datanode</value>
    </property>
</configuration>
```

```

yarn-site.xml
<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
    <property>
        <name>yarn.nodemanager.auxservice.mapreduce.shuffle.class</name>
        <value>org.apache.hadoop.mapred.shuffleHandler</value>
    </property>
</configuration>
```

Step 11: Go to hadoop-env.cmd file in /etc/hadoop folder and replace the set JAVA_HOME=%JAVA_HOME% line with the following:

```
set JAVA_HOME=C:\Program~1\Java\jdk-21
```

Step 12: Restart your PC for the changes to take effect.

Step 13: Go to Admin Command prompt and type “hadoop” to see if the server is recognized.

```
Administrator: Command Pro X + ▾  
Microsoft Windows [Version 10.0.22631.3880]  
(c) Microsoft Corporation. All rights reserved.  
C:\Users\admin>cd C:\Program Files\hadoop-3.4.0  
C:\Program Files\hadoop-3.4.0>cd bin  
C:\Program Files\hadoop-3.4.0\bin>
```

```
Administrator: Command Pro X + ▾ - □ ×  
C:\Users\admin>hadoop  
Usage: hadoop [--config confdir] [--loglevel loglevel] COMMAND  
where COMMAND is one of:  
  fs          run a generic filesystem user client  
  version     print the version  
  jar <jar>    run a jar file  
              note: please use "yarn jar" to launch  
              YARN applications, not this command.  
  checknative [-a|-h]  check native hadoop and compression libraries availability  
  conftest    validate configuration XML files  
  distch path:owner:group:permisson  
              distributed metadata changer  
  distcp <srcurl> <desturl> copy file or directories recursively  
  archive -archiveName NAME -p <parent path> <src> <dest> create a hadoop archive  
  classpath   prints the class path needed to get the  
              Hadoop jar and the required libraries  
  credential  interact with credential providers  
  jnopath     prints the java.library.path  
  kerbname    show auth_to_local principal conversion  
  kdiag       diagnose kerberos problems  
  key         manage keys via the KeyProvider  
  trace       view and modify Hadoop tracing settings  
  daemonlog   get/set the log level for each daemon  
  or  
  CLASSNAME   run the class named CLASSNAME  
  
Most commands print help when invoked w/o parameters.  
C:\Users\admin>
```

Step 14: Type “hdfs namenode -format” to format the namenode.

```
C:\Users\admin>hdfs namenode -format
```

```
Administrator: Command Pro + 
2024-07-31 10:17:41,894 INFO util.GSet: 0.25% max memory 1000 MB = 2.5 MB
2024-07-31 10:17:41,894 INFO util.GSet: capacity      = 2^18 = 262144 entries
2024-07-31 10:17:41,898 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
2024-07-31 10:17:41,898 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
2024-07-31 10:17:41,899 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
2024-07-31 10:17:41,901 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
2024-07-31 10:17:41,901 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
2024-07-31 10:17:41,902 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2024-07-31 10:17:41,902 INFO util.GSet: VM type      = 64-bit
2024-07-31 10:17:41,902 INFO util.GSet: 0.029999999329447746% max memory 1000 MB = 307.2 KB
2024-07-31 10:17:41,902 INFO util.GSet: capacity      = 2^15 = 32768 entries
2024-07-31 10:17:46,837 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1936360632-192.168.56.1-1722401266837
2024-07-31 10:17:46,865 INFO common.Storage: Storage directory C:\hadoop-3.4.0\data\namenode has been successfully formatted.
2024-07-31 10:17:46,878 INFO namenode.FSImageFormatProtobuf: Saving image file C:\hadoop-3.4.0\data\namenode\current\fsimage.ckpt_00000000000000000000 using no compression
2024-07-31 10:17:46,920 INFO namenode.FSImageFormatProtobuf: Image file C:\hadoop-3.4.0\data\namenode\current\fsimage.ckpt_00000000000000000000 of size 400 bytes saved in 0 seconds .
2024-07-31 10:17:46,935 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2024-07-31 10:17:46,938 INFO blockmanagement.DatanodeManager: Slow peers collection thread shutdown
2024-07-31 10:17:46,947 INFO namenode.FSNamesystem: Stopping services started for active state
2024-07-31 10:17:46,948 INFO namenode.FSNamesystem: Stopping services started for standby state
2024-07-31 10:17:46,950 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2024-07-31 10:17:46,950 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at 31D-LAB4-11/192.168.56.1
*****/
```

Step 15: Type start-all.cmd to start all hadoop processes (make sure you have added set JAVA_HOME=C:\Progra~1\Java\jdk-22)

```
Apache Hadoop Distribution - yarn resourcemanager
2024-08-08 11:07:30,879 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2024-08-08 11:07:30,926 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue, queueCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false, ipcFailOver: false.
2024-08-08 11:07:30,948 INFO ipc.Server: Listener at 0.0.0.0:8030
2024-08-08 11:07:30,950 INFO ipc.Server: Starting Socket Reader #1 for port 8030
2024-08-08 11:07:30,960 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationMasterProtocolPB to the server
2024-08-08 11:07:31,023 INFO ipc.Server: IPC Server Responder: starting
2024-08-08 11:07:31,023 INFO ipc.Server: IPC Server listener on 8030: starting
2024-08-08 11:07:31,243 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue, queueCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false, ipcFailOver: false.
2024-08-08 11:07:31,244 INFO ipc.Server: Listener at 0.0.0.0:8032
2024-08-08 11:07:31,254 INFO ipc.Server: Starting Socket Reader #1 for port 8032
2024-08-08 11:07:31,256 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationClientProtocolPB to the server
2024-08-08 11:07:31,258 INFO ipc.Server: IPC Server Responder: starting
2024-08-08 11:07:31,259 INFO ipc.Server: IPC Server listener on 8032: starting
2024-08-08 11:07:32,117 INFO webproxy.ProxyCA: Created Certificate for OU=YARN-9bffb8a3-6efe-4c00-b2cd-e0d105ef686c
2024-08-08 11:07:32,323 INFO recovery.RMStateStore: Storing CA Certificate and Private Key
2024-08-08 11:07:32,356 INFO resourcemanager.ResourceManager: Transitioned to active state
2024-08-08 11:07:33,889 INFO resourcemanager.ResourceTrackerService: NodeManager from node 31D-LAB5-26.SVV.local(cmPort: 62661 httpPort: 8042) registered with capability: <memory:8192, vCores:8>, assigned nodeId 31D-LAB5-26.SVV.local:62661
2024-08-08 11:07:33,905 INFO rmnode.RMNodeImpl: 31D-LAB5-26.SVV.local:62661 Node Transitioned from NEW to RUNNING
2024-08-08 11:07:33,934 INFO capacity.AbstractLeafQueue: LeafQueue: root.default update max app related, maxApplications =10000, maxApplicationsPerUser=10000, Abs Cap:1.0, Cap: 1.0, MaxCap : 1.0
2024-08-08 11:07:33,937 INFO capacity.CapacityScheduler: Added node 31D-LAB5-26.SVV.local:62661 clusterResource: <memory:8192, vCores:8>
2024-08-08 11:07:33,938 INFO capacity.AbstractLeafQueue: LeafQueue: root.default update max app related, maxApplications =10000, maxApplicationsPerUser=10000, Abs Cap:1.0, Cap: 1.0, MaxCap : 1.0
```

```
Apache Hadoop Distribution - yarn resourcemanager
2024-08-08 11:07:30,879 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2024-08-08 11:07:30,926 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue, queueCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false, ipcFailOver: false.
2024-08-08 11:07:30,948 INFO ipc.Server: Listener at 0.0.0.0:8030
2024-08-08 11:07:30,950 INFO ipc.Server: Starting Socket Reader #1 for port 8030
2024-08-08 11:07:30,960 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationMasterProtocolPB to the server
2024-08-08 11:07:31,023 INFO ipc.Server: IPC Server Responder: starting
2024-08-08 11:07:31,023 INFO ipc.Server: IPC Server listener on 8030: starting
2024-08-08 11:07:31,243 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue, queueCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false, ipcFailOver: false.
2024-08-08 11:07:31,244 INFO ipc.Server: Listener at 0.0.0.0:8032
2024-08-08 11:07:31,254 INFO ipc.Server: Starting Socket Reader #1 for port 8032
2024-08-08 11:07:31,256 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationClientProtocolPB to the server
2024-08-08 11:07:31,258 INFO ipc.Server: IPC Server Responder: starting
2024-08-08 11:07:31,259 INFO ipc.Server: IPC Server listener on 8032: starting
2024-08-08 11:07:32,117 INFO webproxy.ProxyCA: Created Certificate for OU=YARN-9bffb8a3-6efe-4c00-b2cd-e0d105ef686c
2024-08-08 11:07:32,323 INFO recovery.RMStateStore: Storing CA Certificate and Private Key
2024-08-08 11:07:32,356 INFO resourcemanager.ResourceManager: Transitioned to active state
2024-08-08 11:07:33,889 INFO resourcemanager.ResourceTrackerService: NodeManager from node 31D-LAB5-26.SVV.local(cmPort: 62661 httpPort: 8042) registered with capability: <memory:8192, vCores:8>, assigned nodeId 31D-LAB5-26.SVV.local:62661
2024-08-08 11:07:33,905 INFO rmnode.RMNodeImpl: 31D-LAB5-26.SVV.local:62661 Node Transitioned from NEW to RUNNING
2024-08-08 11:07:33,934 INFO capacity.AbstractLeafQueue: LeafQueue: root.default update max app related, maxApplications =10000, maxApplicationsPerUser=10000, Abs Cap:1.0, Cap: 1.0, MaxCap : 1.0
2024-08-08 11:07:33,937 INFO capacity.CapacityScheduler: Added node 31D-LAB5-26.SVV.local:62661 clusterResource: <memory:8192, vCores:8>
2024-08-08 11:07:33,938 INFO capacity.AbstractLeafQueue: LeafQueue: root.default update max app related, maxApplications =10000, maxApplicationsPerUser=10000, Abs Cap:1.0, Cap: 1.0, MaxCap : 1.0
```

Step 16: Go to your browser and type localhost:9870 to view Hadoop Page.

Overview 'localhost:9000' (active)

| | |
|----------------|--|
| Started: | Wed Jul 31 10:18:41 +0530 2024 |
| Version: | 3.4.0, rbd8b77f9398626bd779178319ee7a5dfaeecc760 |
| Compiled: | Mon Mar 04 12:05:00 +0530 2024 by root from (HEAD detached at release-3.4.0-RC3) |
| Cluster ID: | CID-669245c9-e417-4ab6-95ab-94f13245bd70 |
| Block Pool ID: | BP-1936360632-192.168.56.1-172240126687 |

Summary

Security is off.
Safemode is off.

1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).

Heap Memory used 62.28 MB of 81 MB Heap Memory. Max Heap Memory is 1000 MB.

Non Heap Memory used 54.78 MB of 56.88 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

| | |
|--|--|
| Configured Capacity: | 276.42 GB |
| Configured Remote Capacity: | 0 B |
| DFS Used: | 149 B (0%) |
| Non DFS Used: | 129.64 GB |
| DFS Remaining: | 146.79 GB (53.1%) |
| Block Pool Used: | 149 B (0%) |
| DataNodes Usages% (Min/Median/Max/stdDev): | 0.00% / 0.00% / 0.00% / 0.00% |
| Live Nodes | 1 (Decommissioned: 0, In Maintenance: 0) |

Step 17: Now, go to cmd and type start-yarn.cmd:

```
C:\hadoop-3.4.0\bin>start-dfs.cmd

C:\hadoop-3.4.0\bin>start-yarn.cmd
starting yarn daemons

C:\hadoop-3.4.0\bin>_
```

Step 18: Now, go to localhost:8088 and observe accordingly:

The screenshot shows the Hadoop Cluster Metrics page at localhost:8088/cluster. The left sidebar has a 'Cluster' section with links for About, Nodes, Node Labels, Applications, Scheduler, and Tools. The main area displays various metrics tables:

- Cluster Metrics:** Shows Apps Submitted (0), Apps Pending (0), Apps Running (0), Apps Completed (0), Containers Running (0), and Used Resources (<memory:0 B, vCores:0>).
- Cluster Nodes Metrics:** Shows Active Nodes (1), Decommissioning Nodes (0), and Decommissioned Nodes (0).
- Scheduler Metrics:** Shows Scheduler Type (Capacity Scheduler), Scheduling Resource Type ([memory-mb (unit=Mi), vcores]), Minimum Allocation (<memory:1024, vCores:1>), Maximum Allocation (<memory:8192, vCores:4>), and Maximum Clus.

A message at the bottom states "Showing 0 to 0 of 0 entries".

Practical 2:

Aim: Spark SQL

Steps:

A-Installation of Scala and Spark:

1-Install Scala and perform the following steps:

The screenshot shows the Scala website's 'INSTALL' page. At the top, there is a navigation bar with links for LEARN, INSTALL (which is highlighted in red), PLAYGROUND, FIND A LIBRARY, COMMUNITY, and BLOG. Below the navigation bar, the word 'INSTALL' is prominently displayed in large white letters. The main content area has a heading 'Install Scala with `cs setup` (recommended)'. Below this, a paragraph explains that it is recommended to use `cs setup`, the Scala installer powered by Coursier. It installs everything necessary to use the latest Scala release from a command line. There are tabs for macOS, Linux, Windows (which is selected and highlighted in red), and Other. A note below the tabs says to download and execute the Scala installer for Windows based on Coursier and follow the on-screen instructions. A blue button labeled 'Testing your setup' with a progress bar is shown. Below the button, a note encourages users to read the getting started guide if they are just beginning their journey with Scala. A pink button at the bottom right says 'GET STARTED WITH SCALA' with a download icon.

2-

The screenshot shows a terminal window with a black background and white text. The window title is 'C:\Users\admin\Downloads\c'. The terminal output shows the following steps:

```
Checking if a JVM is installed
Found a JVM installed under C:\Program Files\Java\jdk1.8.0_191.

Checking if ~\AppData\Local\Coursier\data\bin is in PATH
Should we add ~\AppData\Local\Coursier\data\bin to your PATH? [Y/n] $
```

3-

The screenshot shows a terminal window with the title bar "C:\Users\admin\Downloads\c". The window contains the following text:

```
Checking if a JVM is installed
Found a JVM installed under C:\Program Files\Java\jdk1.8.0_191.

Checking if ~\AppData\Local\Coursier\data\bin is in PATH
Should we add ~\AppData\Local\Coursier\data\bin to your PATH? [Y/n] y

Checking if the standard Scala applications are installed
```

4-

The screenshot shows a terminal window with the title bar "C:\Users\admin\Downloads\c". The window contains the following text:

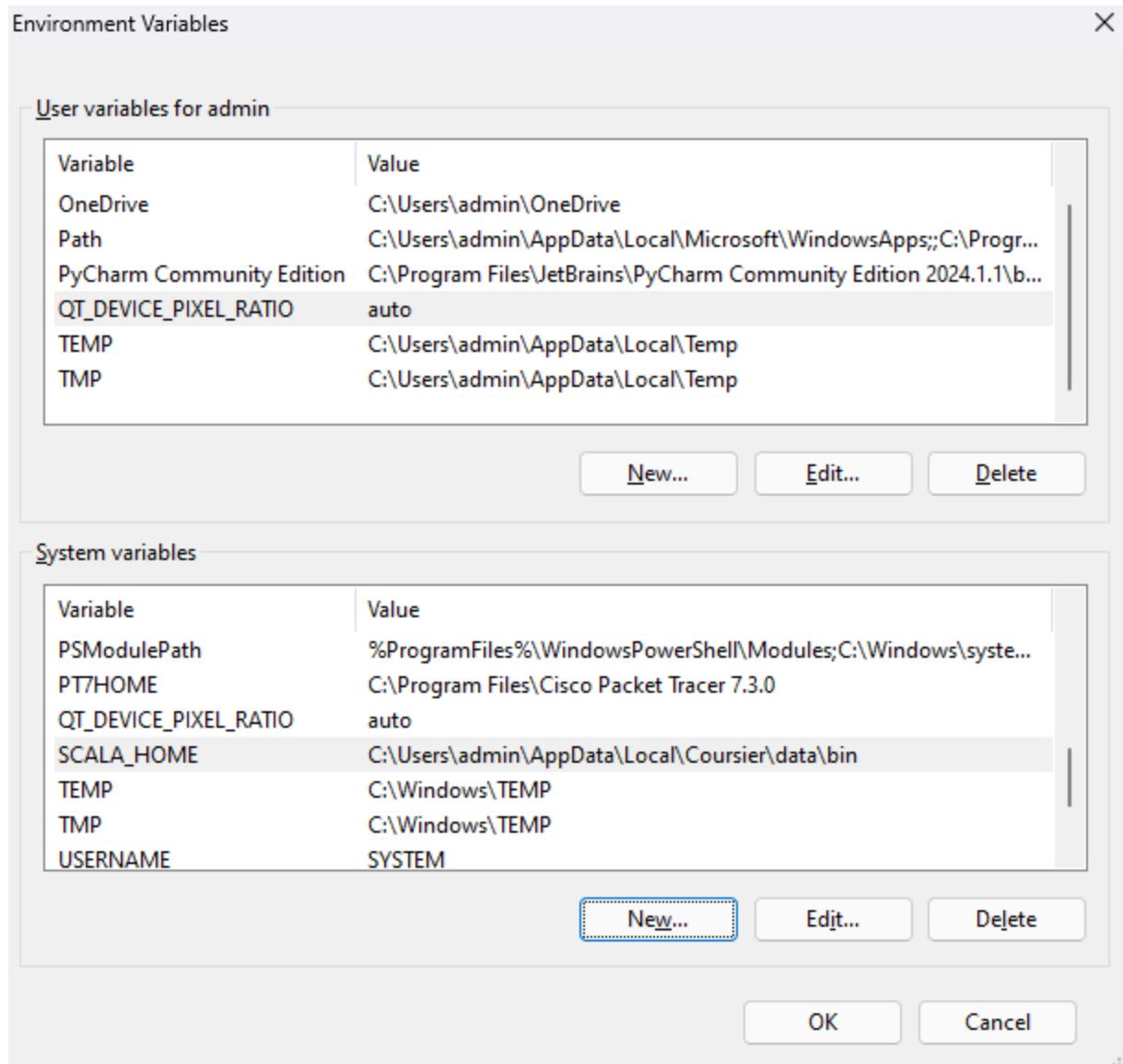
```
Checking if a JVM is installed
Found a JVM installed under C:\Program Files\Java\jdk1.8.0_191.

Checking if ~\AppData\Local\Coursier\data\bin is in PATH
Should we add ~\AppData\Local\Coursier\data\bin to your PATH? [Y/n] y

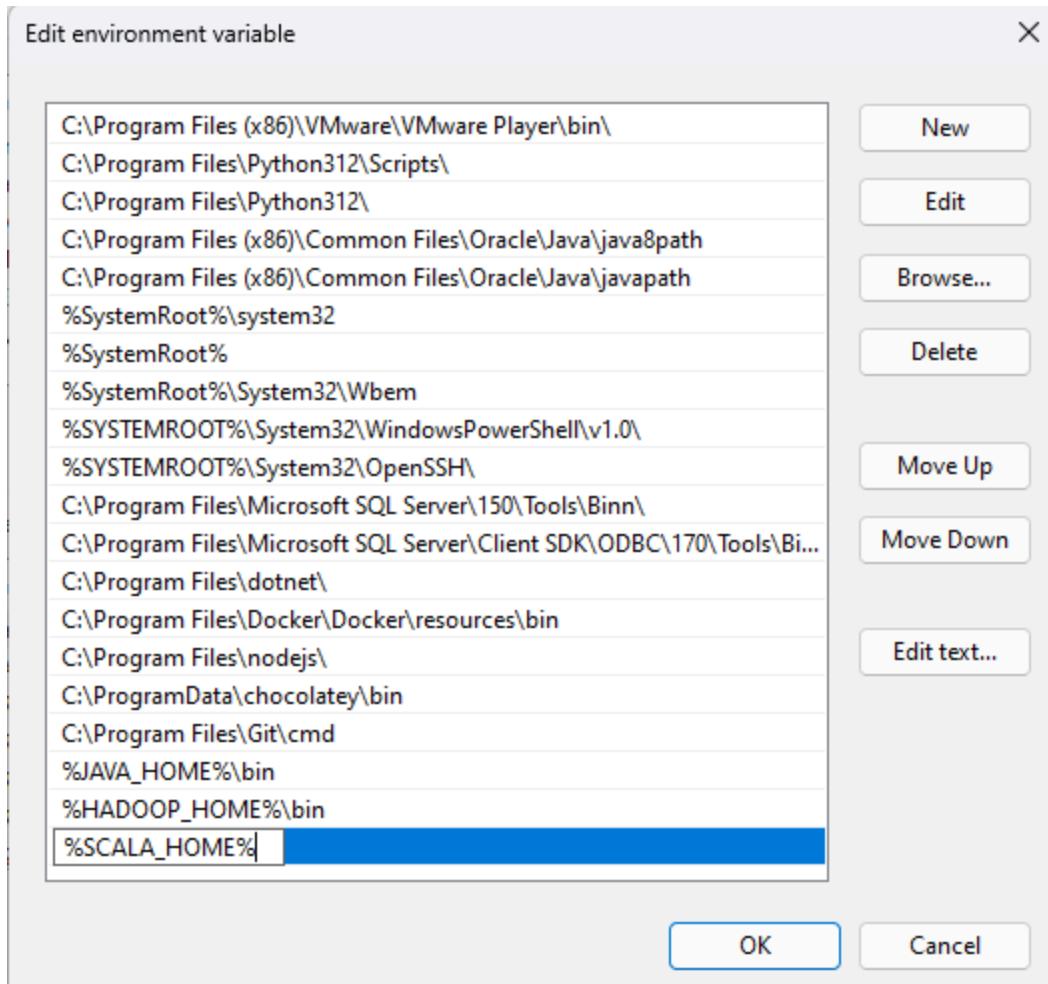
Checking if the standard Scala applications are installed
Installed ammonite
Installed cs
Installed coursier
Installed scala
Installed scalac
Installed scala-cli
Installed sbt
Installed sbtn
Installed scalafmt

Press "ENTER" to continue...
```

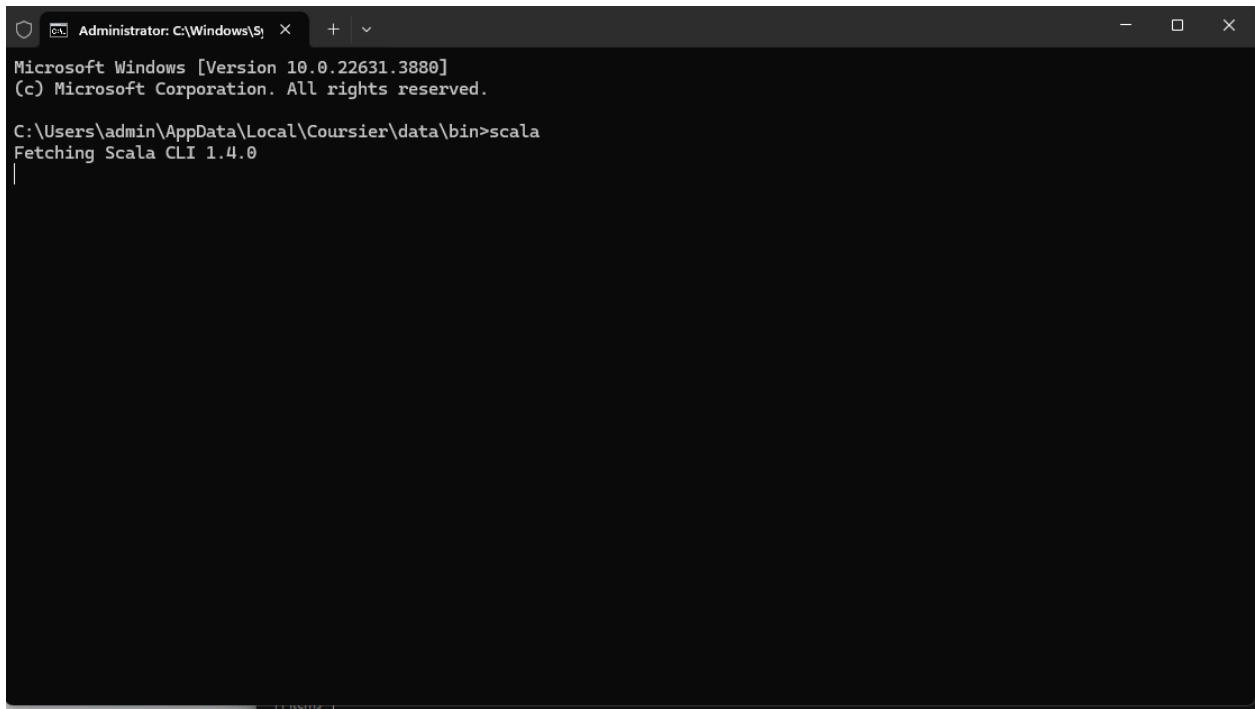
5-Now, make sure to set the environmental path variables



6-

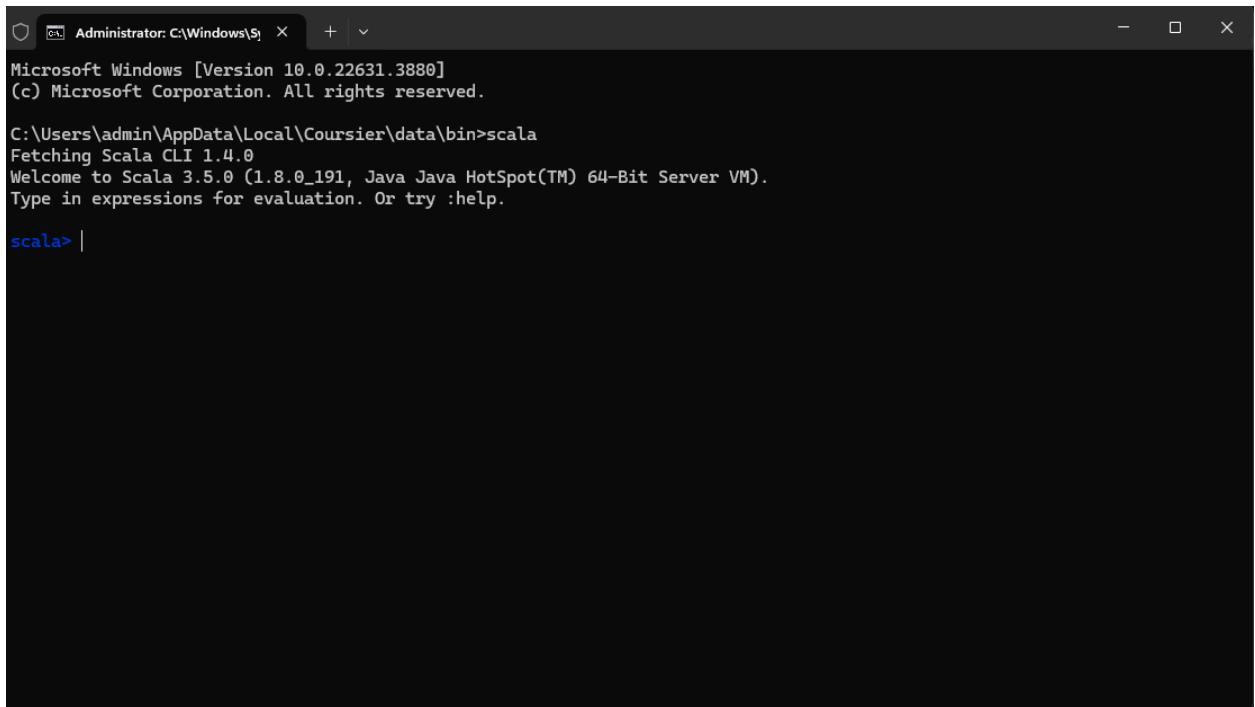


7-Now, open the following with the help of cmd (enable hidden files if unable to find it as desired) and the following path C:\Users\admin\AppData\Local\Coursier\data\bin to access scala



Administrator: C:\Windows\\$ Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.
C:\Users\admin\AppData\Local\Coursier\data\bin>scala
Fetching Scala CLI 1.4.0
|

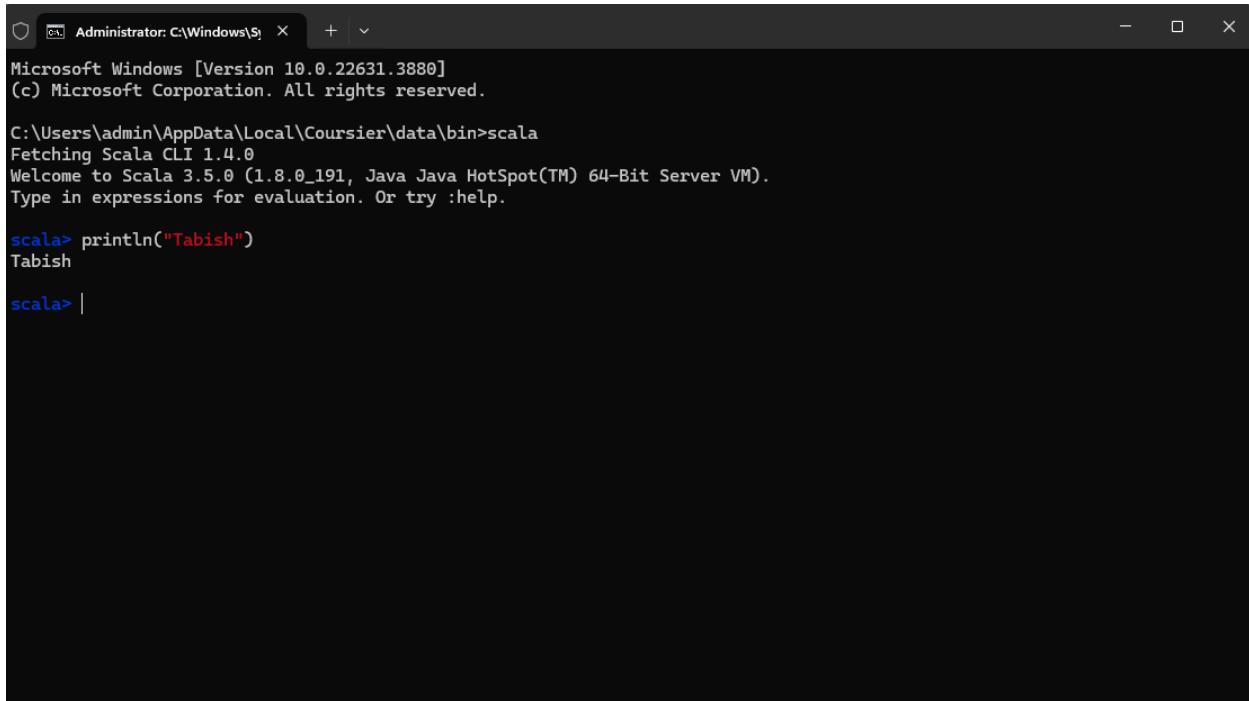
8-Successfully accessed the desired interface



Administrator: C:\Windows\\$ Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.
C:\Users\admin\AppData\Local\Coursier\data\bin>scala
Fetching Scala CLI 1.4.0
Welcome to Scala 3.5.0 (1.8.0_191, Java Java HotSpot(TM) 64-Bit Server VM).
Type in expressions for evaluation. Or try :help.
scala> |

9-Perform the following programs:

A-

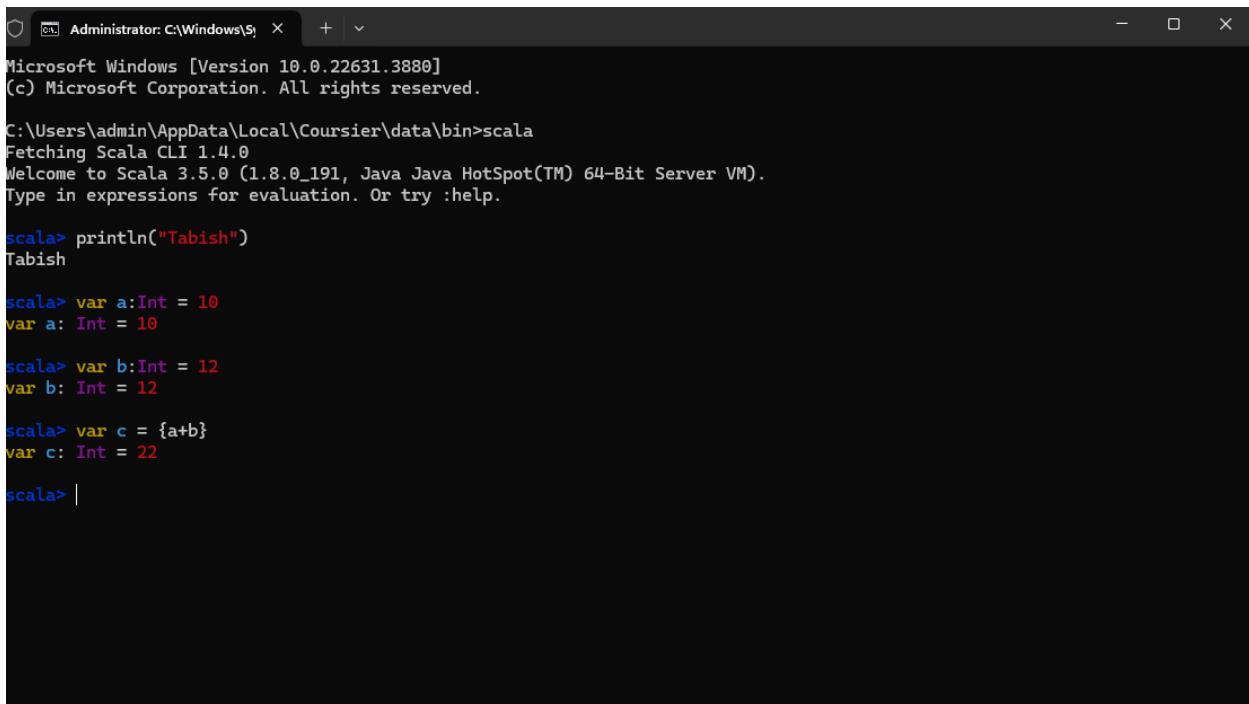


```
Administrator: C:\Windows\S... + | 
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin\AppData\Local\Coursier\data\bin>scala
Fetching Scala CLI 1.4.0
Welcome to Scala 3.5.0 (1.8.0_191, Java Java HotSpot(TM) 64-Bit Server VM).
Type in expressions for evaluation. Or try :help.

scala> println("Tabish")
Tabish
scala> |
```

B-



```
Administrator: C:\Windows\S... + | 
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin\AppData\Local\Coursier\data\bin>scala
Fetching Scala CLI 1.4.0
Welcome to Scala 3.5.0 (1.8.0_191, Java Java HotSpot(TM) 64-Bit Server VM).
Type in expressions for evaluation. Or try :help.

scala> println("Tabish")
Tabish
scala> var a:Int = 10
var a: Int = 10
scala> var b:Int = 12
var b: Int = 12
scala> var c = {a+b}
var c: Int = 22
scala> |
```

10-Install Spark



Download Libraries Documentation Examples Community Developers GitHub Apache Software Foundation

Download Apache Spark™

1. Choose a Spark release: **3.4.3 (Apr 18 2024)**
2. Choose a package type: **Pre-built for Apache Hadoop 3.3 and later**
3. Download Spark: [spark-3.4.3-bin-hadoop3.tgz](#)
4. Verify this release using the 3.4.3 signatures, checksums and project release KEYS by following these [procedures](#).

Note that Spark 3 is pre-built with Scala 2.12 in general and Spark 3.2+ provides additional pre-built distribution with Scala 2.13.

Link with Spark

Spark artifacts are hosted in [Maven Central](#). You can add a Maven dependency with the following coordinates:

```
groupId: org.apache.spark  
artifactId: spark-core_2.12  
version: 3.5.2
```

Installing with PyPI

PySpark is now available in pypi. To install just run `pip install pyspark`.

Installing with Docker

Spark docker images are available from Dockerhub under the accounts of both [The Apache Software Foundation](#) and [Official Images](#).

Note that, these images contain non-ASF software and may be subject to different license terms. Please check their [Dockerfiles](#) to verify whether to verify whether they are compatible with your deployment.

Release notes for stable releases

- [Spark 3.5.2 \(Aug 10 2024\)](#)
- [Spark 3.4.3 \(Apr 18 2024\)](#)

Archived releases

As new Spark releases come out for each development stream, previous ones will be archived, but they are still available at [Spark](#)

Latest News

- Spark 3.5.2 released (Aug 10, 2024)
Preview release of Spark 4.0 (Jun 03, 2024)
Spark 3.4.3 released (Apr 18, 2024)
Spark 3.5.1 released (Feb 23, 2024)

[Archive](#)



[DOWNLOAD SPARK](#)

Built-in Libraries

- SQL and DataFrames
Spark Streaming
MLlib (machine learning)
GraphX (graph)

Third-Party Projects

11-Set the environmental variables accordingly (using C:\spark\spark-3.5.2-bin-hadoop3\bin)

Environment Variables

X

User variables for admin

| Variable | Value |
|---------------------------|--|
| OneDrive | C:\Users\admin\OneDrive |
| Path | C:\Users\admin\AppData\Local\Microsoft\WindowsApps;;C:\Progr... |
| PyCharm Community Edition | C:\Program Files\JetBrains\PyCharm Community Edition 2024.1.1\b... |
| QT_DEVICE_PIXEL_RATIO | auto |
| TEMP | C:\Users\admin\AppData\Local\Temp |
| TMP | C:\Users\admin\AppData\Local\Temp |

New...

[Edit...](#)

Delete

System variables

| Variable | Value |
|-----------------------|--|
| QT_DEVICE_PIXEL_RATIO | auto |
| SCALA_HOME | C:\Users\admin\AppData\Local\Coursier\data\bin |
| SPARK_HOME | C:\spark\spark-3.5.2-bin-hadoop3 |
| TEMP | C:\Windows\TEMP |
| TMP | C:\Windows\TEMP |
| USERNAME | SYSTEM |
| windir | C:\Windows |

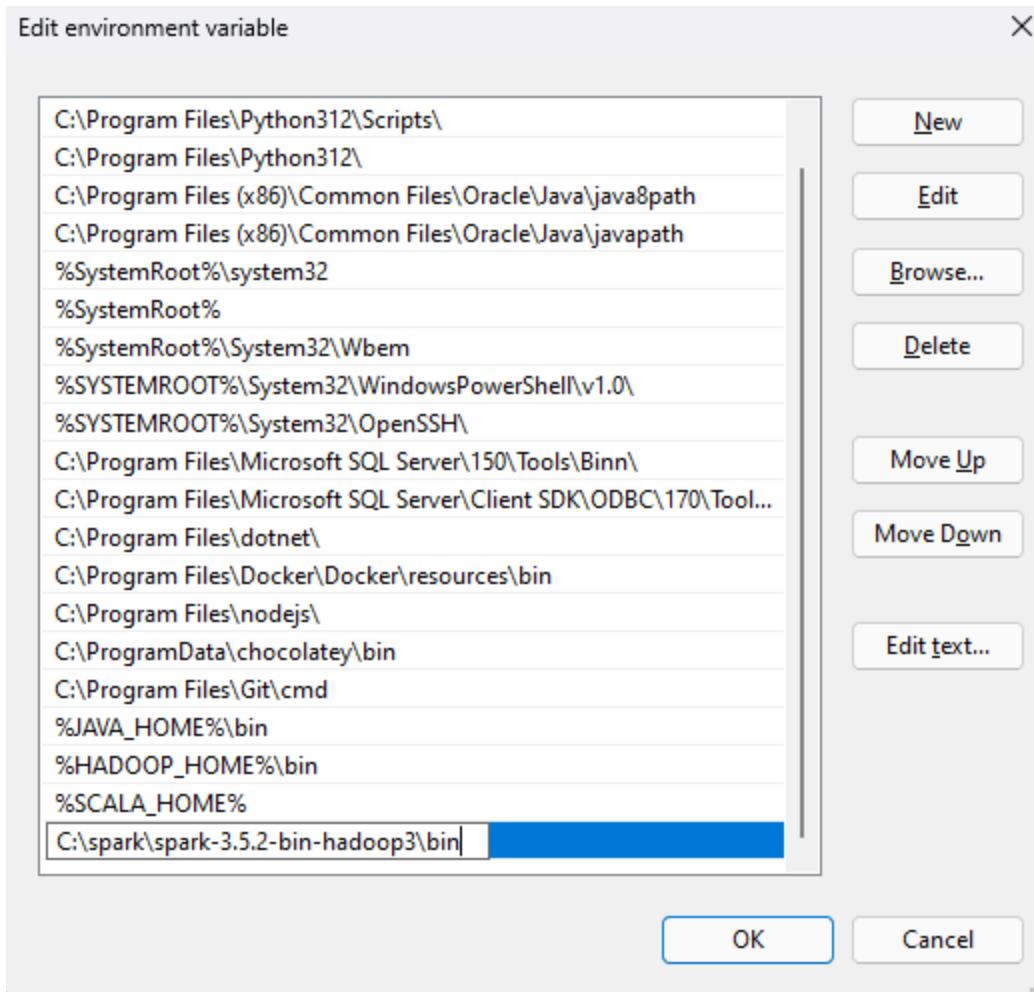
New...

[Edit...](#)

Delete

ok

[Cancel](#)



12-Run spark (spark-shell) and perform the following:

A-

B-

```
scala> val Data = spark.read.json("C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.json")
Data: org.apache.spark.sql.DataFrame = [age: bigint, name: string]
```

C-

```
scala> Data.show()
+---+-----+
| age|    name|
+---+-----+
| NULL|Michael|
|   30|    Andy|
|   19| Justin|
+---+-----+
```

D-

```
scala> Data.printSchema()
root
|-- age: long (nullable = true)
|-- name: string (nullable = true)

scala> |
```

E-

```
scala> Data.select($"name",$"age").show()
+-----+---+
|    name| age|
+-----+---+
|Michael|NULL|
|   Andy|  30|
| Justin|  19|
+-----+---+
```

```
scala> |
```

F-

```
scala> Data.filter($"age">>20).show()
+---+---+
|age|name|
+---+---+
| 30|Andy|
+---+---+
```

```
scala> |
```

G-

```
scala> Data.select($"age"+1).show()
+-----+
|(age + 1)|
+-----+
|      NULL|
|        31|
|        20|
+-----+
```

```
scala> |
```

H-

```
scala> val Data2 = spark.read.csv("C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.csv")
Data2: org.apache.spark.sql.DataFrame = [_c0: string]

scala> Data2.show()
+-----+
|      _c0|
+-----+
| name;age;job|
| Jorge;30;Developer|
| Bob;32;Developer|
+-----+


scala> |
```

I-

```
scala> val Data2 = spark.read.option("header", "true").csv("C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.csv")
Data2: org.apache.spark.sql.DataFrame = [name;age;job: string]

scala> Data2.show()
+-----+
|      name;age;job|
+-----+
| Jorge;30;Developer|
| Bob;32;Developer|
+-----+


scala> |
```

B-Now, further perform the following (refer: [Getting Started - Spark 3.5.2 Documentation \(apache.org\)](#)):

A-

```
scala> val Data=spark.read.json("C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.json")
Data: org.apache.spark.sql.DataFrame = [age: bigint, name: string]

scala> Data.createOrReplaceTempView("people")

scala> val sqlDF=spark.sql("SELECT * FROM people")
sqlDF: org.apache.spark.sql.DataFrame = [age: bigint, name: string]

scala> sqlDF.show()
+---+---+
| age| name|
+---+---+
|NULL|Michael|
| 30| Andy|
| 19| Justin|
+---+---+


scala> |
```

B-

```

scala> Data.createGlobalTempView("people")
24/09/19 10:34:09 WARN HiveConf: HiveConf of name hive.stats.jdbc.timeout does not exist
24/09/19 10:34:09 WARN HiveConf: HiveConf of name hive.stats.retries.wait does not exist
24/09/19 10:34:11 WARN ObjectStore: Version information not found in metastore. hive.metastore.schema.verification is not enabled so recording the schema version 2.3.0
24/09/19 10:34:11 WARN ObjectStore: setMetaStoreSchemaVersion called but recording version is disabled: version = 2.3.0, comment = Set by MetaStore UNKNOWN@172.23.1.154
24/09/19 10:34:11 WARN ObjectStore: Failed to get database default, returning NoSuchObjectException
24/09/19 10:34:12 WARN ObjectStore: Failed to get database global_temp, returning NoSuchObjectException

scala> spark.sql("SELECT * FROM global_temp.people").show()
+---+---+
| age| name|
+---+---+
|NULL|Michael|
| 30| Andy|
| 19| Justin|
+---+---+

scala> spark.newSession().sql("SELECT * FROM global_temp.people").show()
+---+---+
| age| name|
+---+---+
|NULL|Michael|
| 30| Andy|
| 19| Justin|
+---+---+

scala> |

```

C-

```

scala> case class Person(name: String, age: Long)
defined class Person

scala> val caseClassDS = Seq(Person("Andy", 32)).toDS()
caseClassDS: org.apache.spark.sql.Dataset[Person] = [name: string, age: bigint]

scala> caseClassDS.show()
+---+---+
|name|age|
+---+---+
|Andy| 32|
+---+---+

scala> val primitiveDS = Seq(1, 2, 3).toDS()
primitiveDS: org.apache.spark.sql.Dataset[Int] = [value: int]

scala> primitiveDS.map(_ + 1).collect() // Returns: Array(2, 3, 4)
res6: Array[Int] = Array(2, 3, 4)

scala> val path = "C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.json"
path: String = C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.json

scala> val peopleDS = spark.read.json(path).as[Person]
peopleDS: org.apache.spark.sql.Dataset[Person] = [age: bigint, name: string]

scala> peopleDS.show()
+---+---+
| age| name|
+---+---+
|NULL|Michael|
| 30| Andy|
| 19| Justin|
+---+---+

scala> |

```

D-

```
scala> import spark.implicits._
import spark.implicits._

scala> val peopleDF = spark.sparkContext
peopleDF: org.apache.spark.SparkContext = org.apache.spark.SparkContext@4599f1e0

scala> .textFile("C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.txt")
res0: org.apache.spark.rdd.RDD[String] = C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.txt MapPartitionsRDD[28] at textFile at <console>:27

scala> .map(_.split(","))
res1: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[29] at map at <console>:27

scala> .map(attributes => Person(attributes(0), attributes(1).trim.toInt))
res10: org.apache.spark.rdd.RDD[Person] = MapPartitionsRDD[30] at map at <console>:29

scala> .toDF()
res11: org.apache.spark.sql.DataFrame = [name: string, age: bigint]

scala> peopleDF.createOrReplaceTempView("people")
```

```
scala> val teenagersDF = spark.sql("SELECT name, age FROM people WHERE age BETWEEN 13 AND 19")
teenagersDF: org.apache.spark.sql.DataFrame = [name: string, age: bigint]

scala> teenagersDF.map(teenager => "Name: " + teenager(0)).show()
+-----+
|    value|
+-----+
|Name: Justin|
+-----+


scala> teenagersDF.map(teenager => "Name: " + teenager.getAs[String]("name")).show()
+-----+
|    value|
+-----+
|Name: Justin|
+-----+


scala> implicit val mapEncoder = org.apache.spark.sql.Encoders.kryo[Map[String, Any]]
mapEncoder: org.apache.spark.sql.Encoder[Map[String,Any]] = class[value[0]: binary]

scala> teenagersDF.map(teenager => teenager.getValuesMap[Any](List("name", "age"))).collect()
res16: Array[Map[String,Any]] = Array(Map(name -> Justin, age -> 19))

scala> |
```

E-

```
scala> import org.apache.spark.sql.Row
import org.apache.spark.sql.Row

scala> import org.apache.spark.sql.types._
import org.apache.spark.sql.types._

scala> val peopleRDD = spark.sparkContext.textFile("C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.txt")
peopleRDD: org.apache.spark.rdd.RDD[String] = C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.txt MapPartitionsRDD[44] at textFile at <console>:30

scala> val schemaString = "name age"
schemaString: String = name age

scala> val fields = schemaString.split(" ")
fields: Array[String] = Array(name, age)

scala> .map(fieldName => StructField(fieldName, StringType, nullable = true))
res17: Array[org.apache.spark.sql.types.StructField] = Array(StructField(name,StringType,true), StructField(age,StringType,true))

scala> val schema = StructType(fields)

scala> val rowRDD = peopleRDD
rowRDD: org.apache.spark.rdd.RDD[String] = C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.txt MapPartitionsRDD[44] at textFile at <console>:30
```

```

scala> .map(_.split(","))
res18: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[45] at map at <console>:32

scala> .map(attributes => Row(attributes(0), attributes(1).trim))
res19: org.apache.spark.rdd.RDD[org.apache.spark.sql.Row] = MapPartitionsRDD[46] at map at <console>:32

scala> val peopleDF = spark.createDataFrame(rowRDD, schema)

```

```
scala> val schema = StructType(fields)
```

```
scala> val peopleDF = spark.createDataFrame(rowRDD, schema)
```

```
scala> val results = spark.sql("SELECT name FROM people")
results: org.apache.spark.sql.DataFrame = [name: string]
```

```
scala> results.map(attributes => "Name: " + attributes(0)).show()
+-----+
|      value|
+-----+
|Name: Michael|
|  Name: Andy|
| Name: Justin|
+-----+
```

```
scala> |
```

F-

```

scala> val Data = spark.createDataFrame(rowRDD, schema)
<console>:31: error: package schema is not a value
      val Data = spark.createDataFrame(rowRDD, schema)
                           ^

scala> val results = spark.sql("SELECT name FROM people")
results: org.apache.spark.sql.DataFrame = [name: string]
scala> results.map(attributes => "Name: " + attributes(0)).show()
+-----+
|      value|
+-----+
|Name: Michael|
|  Name: Andy|
| Name: Justin|
+-----+
```

scala> val mydata=spark.read.format("csv").option("inferSchema", "true").option("header", "true").load("C:/spark/spark-3.5.2-bin-hadoop3/examples/src/main/resources/people.csv")
mydata: org.apache.spark.sql.DataFrame = [name;age;job: string]

```

scala> mydata.show()
+-----+
|      name;age;job|
+-----+
|Jorge;30;Developer|
|  Bob;32;Developer|
+-----+
```

scala> mydata.show(50)
+-----+
| name;age;job|
+-----+
|Jorge;30;Developer|
| Bob;32;Developer|
+-----+

```
scala> mydata.select($"name;age;job").show()
+-----+
|    name;age;job|
+-----+
|Jorge;30;Developer|
|  Bob;32;Developer|
+-----+  
  
scala> mydata.select($"name;age;job")
res36: org.apache.spark.sql.DataFrame = [name;age;job: string]  
  
scala> |
```

```
scala> mydata.count()
res28: Long = 2  
  
scala> mydata.count.toDouble
res29: Double = 2.0  
  
scala> val totalcount=mydata.count.toDouble
totalcount: Double = 2.0
```

G-

```
scala> val res=spark.sql("select* from people where age>20")
res: org.apache.spark.sql.DataFrame = [age: bigint, name: string]  
  
scala> res.w
<console>:32: error: value w is not a member of org.apache.spark.sql.DataFrame
          res.w
                  ^  
  
scala> res.write.save("mynewdf")  
  
scala> res.show()
+---+---+
|age|name|
+---+---+
| 30|Andy|
+---+---+  
  
scala> |
```

Practical 3:

Aim: Spark GraphX

Steps: Perform the following

1-

```
import org.apache.spark._  
import org.apache.spark.rdd.RDD  
import org.apache.spark.graphx._
```

```
scala> import org.apache.spark._  
import org.apache.spark._  
  
scala> import org.apache.spark.rdd.RDD  
import org.apache.spark.rdd.RDD  
  
scala> import org.apache.spark.graphx._  
import org.apache.spark.graphx._
```

2-

```
val vertices = Array((1L,"A"),(2L,"B"),(3L,"C"))
```

```
scala> val vertices = Array((1L,"A"),(2L,"B"),(3L,"C"))  
val vertices: Array[(Long, String)] = Array((1,A), (2,B), (3,C))
```

3-

```
val vRDD = sc.parallelize(vertices)
```

```
scala> val vRDD = sc.parallelize(vertices)  
warning: 1 deprecation (since 2.13.0); for details, enable `:setting -deprecation` or `:replay -deprec  
ation`  
val vRDD: org.apache.spark.rdd.RDD[(Long, String)] = ParallelCollectionRDD[0] at parallelize at <conso  
le>:1
```

4-

```
vRDD.take(1)  
vRDD.take(2)
```

```
scala> vRDD.take(1)
val res0: Array[(Long, String)] = Array((1,A))

scala> vRDD.take(2)
val res1: Array[(Long, String)] = Array((1,A), (2,B))
```

5-

```
val edges = Array(Edge(1L,2L,1800),Edge(2L,3L,800),Edge(3L,1L,1400))
```

```
scala> val edges = Array(Edge(1L,2L,1800),Edge(2L,3L,800),Edge(3L,1L,1400))
val edges: Array[org.apache.spark.graphx.Edge[Int]] = Array(Edge(1,2,1800), Edge(2,3,800), Edge(3,1,1400))
```

6-

```
val eRDD = sc.parallelize(edges)
```

```
scala> val eRDD = sc.parallelize(edges)
warning: 1 deprecation (since 2.13.0); for details, enable `:setting -deprecation` or `:replay -deprecation`
val eRDD: org.apache.spark.rdd.RDD[org.apache.spark.graphx.Edge[Int]] = ParallelCollectionRDD[1] at parallelize at <console>:1
```

7-

```
eRDD.take(2)
```

```
scala> eRDD.take(2)
val res2: Array[org.apache.spark.graphx.Edge[Int]] = Array(Edge(1,2,1800), Edge(2,3,800))
```

8-

```
val nowhere = "nowhere"
```

```
scala> val nowhere = "nowhere"
val nowhere: String = nowhere
```

9-

```
val graph = Graph(vRDD,eRDD,nowhere)
```

```
scala> val graph = Graph(vRDD,eRDD,nowhere)
val graph: org.apache.spark.graphx.Graph[String,Int] = org.apache.spark.graphx.impl.GraphImpl@2e008502
```

10-

```
#To check number of Airports
val numairports = graph.numVertices
```

```
scala> val numairports = graph.numVertices
val numairports: Long = 3
```

11-

```
#To check routes
val numairports = graph.numEdges
```

```
scala> val numairports = graph.numEdges
val numairports: Long = 3
```

12-

```
#Route having distance > 1000
(graph.edges.filter{case Edge(src,dst,prop)=>prop>1000}.collect.foreach(println))
```

```
scala> (graph.edges.filter{case Edge(src,dst,prop)=>prop>1000}.collect.foreach(println))
warning: 1 deprecation (since 2.13.3); for details, enable `:setting -deprecation` or `:replay -deprecation`
Edge(1,2,1800)
Edge(3,1,1400)
```

13-

```
#Triplet Information
graph.triplets.take(3).foreach(println)
```

```
scala> graph.triplets.take(3).foreach(println)
((1,A),(2,B),1800)
((2,B),(3,C),800)
((3,C),(1,A),1400)
```

14-

```
#Indegree
val i = graph.inDegrees
i.collect()
```

```
scala> val i = graph.inDegrees
val i: org.apache.spark.graphx.VertexRDD[Int] = VertexRDDImpl[25] at RDD at VertexRDD.scala:57

scala> i.collect()
val res5: Array[(org.apache.spark.graphx.VertexId, Int)] = Array((1,1), (2,1), (3,1))
```

15-

```
#Outdegrees
val o = graph.outDegrees
o.collect()
```

```
scala> val o = graph.outDegrees
val o: org.apache.spark.graphx.VertexRDD[Int] = VertexRDDImpl[29] at RDD at VertexRDD.scala:57

scala> o.collect()
val res6: Array[(org.apache.spark.graphx.VertexId, Int)] = Array((1,1), (2,1), (3,1))
```

16-

```
#Total Degree
val t = graph.degrees
t.collect()
```

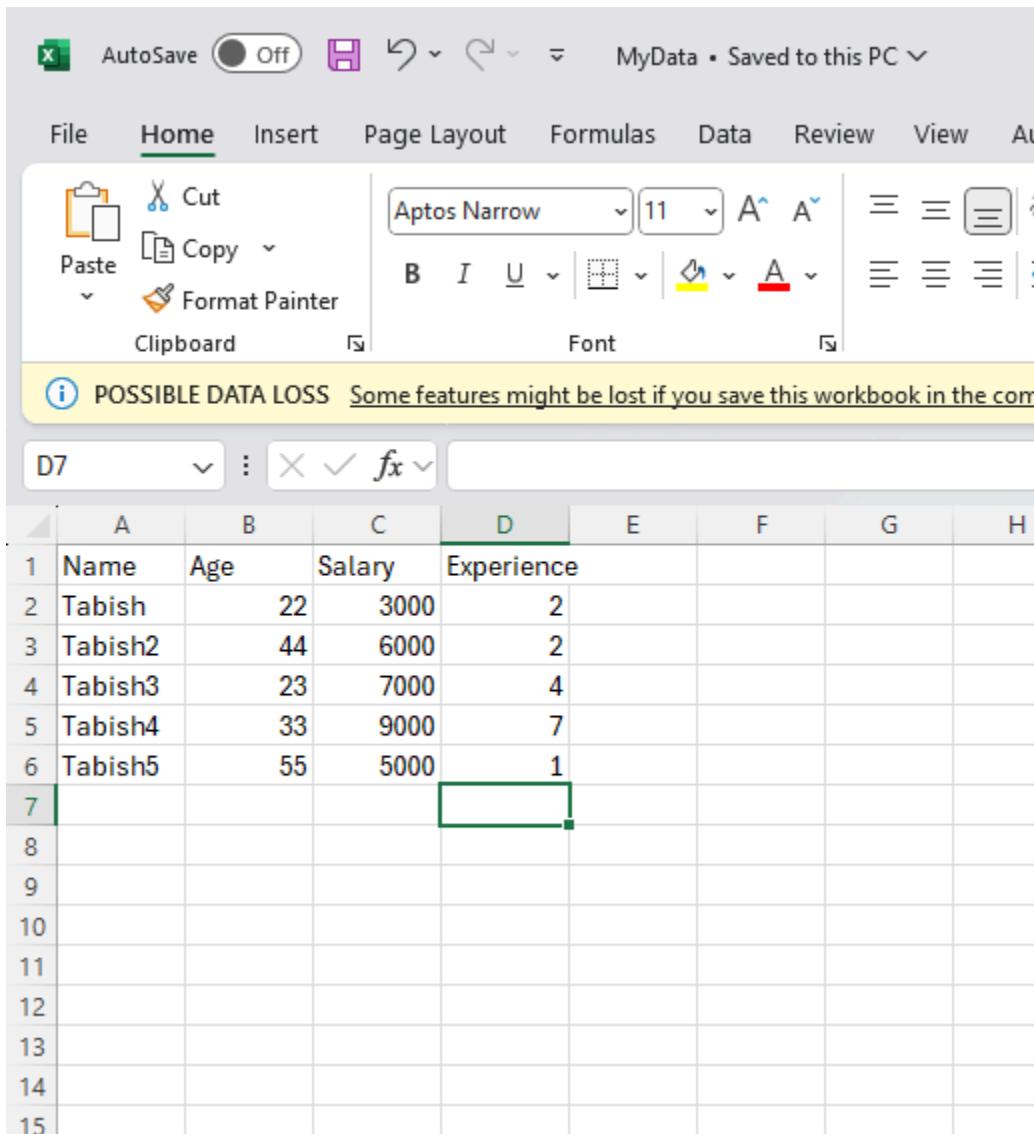
```
scala> val t = graph.degrees
val t: org.apache.spark.graphx.VertexRDD[Int] = VertexRDDImpl[33] at RDD at VertexRDD.scala:57

scala> t.collect()
val res8: Array[(org.apache.spark.graphx.VertexId, Int)] = Array((1,2), (2,2), (3,2))
```

Practical 4:

Aim: PySpark

Step 1:



The screenshot shows a Microsoft Excel spreadsheet titled "MyData". The ribbon at the top is visible with tabs like File, Home, Insert, etc. The Home tab is selected. The clipboard ribbon on the left shows options for Paste, Cut, Copy, and Format Painter. The Font ribbon below it shows settings for Aptos Narrow font, size 11, bold, italic, underline, and various color and style options. The main area displays a table with columns A through H and rows 1 through 15. The first six rows contain data: Row 1 has headers Name, Age, Salary, and Experience; Rows 2 through 6 have values Tabish, 22, 3000, 2; Tabish2, 44, 6000, 2; Tabish3, 23, 7000, 4; Tabish4, 33, 9000, 7; and Tabish5, 55, 5000, 1 respectively. Row 7 is currently selected, indicated by a green border around the entire row.

| | A | B | C | D | E | F | G | H |
|----|---------|-----|--------|------------|---|---|---|---|
| 1 | Name | Age | Salary | Experience | | | | |
| 2 | Tabish | 22 | 3000 | 2 | | | | |
| 3 | Tabish2 | 44 | 6000 | 2 | | | | |
| 4 | Tabish3 | 23 | 7000 | 4 | | | | |
| 5 | Tabish4 | 33 | 9000 | 7 | | | | |
| 6 | Tabish5 | 55 | 5000 | 1 | | | | |
| 7 | | | | | | | | |
| 8 | | | | | | | | |
| 9 | | | | | | | | |
| 10 | | | | | | | | |
| 11 | | | | | | | | |
| 12 | | | | | | | | |
| 13 | | | | | | | | |
| 14 | | | | | | | | |
| 15 | | | | | | | | |

Step 2: Install PySpark

+ Code + Text

Install PySpark

```
✓ 0s [1] pip install pyspark
Collecting pyspark
  Downloading pyspark-3.5.3.tar.gz (317.3 MB)
    Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
    Created wheel for pyspark: filename=pyspark-3.5.3-py2.py3-none-any.whl size=317840625 sha256=c8bc4f14d106284f10a1ba434e3c45a9d3570de44faf5295bd4c6efbc1cb223c
    Stored in directory: /root/.cache/pip/wheels/1b/3a/92/28b93e2fbfdbb7509ca4d6f50c5e407f48dce4ddbda69a4ab
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.5.3
```

Step 2:

```
✓ 0s [3] import pyspark
import pandas as pd
pd.read_csv('MyData.csv')



|   | Name    | Age | Salary | Experience | grid icon | refresh icon |
|---|---------|-----|--------|------------|-----------|--------------|
| 0 | Tabish  | 22  | 3000   | 2          |           |              |
| 1 | Tabish2 | 44  | 6000   | 2          |           |              |
| 2 | Tabish3 | 23  | 7000   | 4          |           |              |
| 3 | Tabish4 | 33  | 9000   | 7          |           |              |
| 4 | Tabish5 | 55  | 5000   | 1          |           |              |


```

Step 3:

create sparksession

```
✓ 0s [4] from pyspark.sql import SparkSession
```

Start Spark Session

```
✓ 0s [5] spark=SparkSession.builder.appName('Practice').getOrCreate()
```

Step 4:

Read dataset and store in variable

```
✓ 1s  df_pyspark=spark.read.csv('MyData.csv', header=True, inferSchema=True)
    df_pyspark.show()
```

```
→ +---+---+---+
|   Name|Age|Salary|Experience|
+---+---+---+
| Tabish| 22| 3000|      2|
|Tabish2| 44| 6000|      2|
|Tabish3| 23| 7000|      4|
|Tabish4| 33| 9000|      7|
|Tabish5| 55| 5000|      1|
+---+---+---+
```

Step 5:

Type of the data

```
✓ 0s  type(df_pyspark)
→ pyspark.sql.dataframe.DataFrame
def __init__(jdf: JavaObject, sql_ctx: Union['SQLContext', 'SparkSession']):

/usr/local/lib/python3.10/dist-packages/pyspark/sql/dataframe.py
A distributed collection of data grouped into named columns.

.. versionadded:: 1.3.0
.. versionchanged:: 3.4.0
```

Step 6:

Check Schema

```
[14] df_pyspark.printSchema()  
root  
| -- Name: string (nullable = true)  
| -- Age: integer (nullable = true)  
| -- Salary: integer (nullable = true)  
| -- Experience: integer (nullable = true)
```

First 3 data value

```
[15] df_pyspark.head(3)  
[Row(Name='Tabish', Age=22, Salary=3000, Experience=2),  
 Row(Name='Tabish2', Age=44, Salary=6000, Experience=2),  
 Row(Name='Tabish3', Age=23, Salary=7000, Experience=4)]
```

To get names of columns

```
[16] df_pyspark.columns  
['Name', 'Age', 'Salary', 'Experience']
```

Step 7:

Selecting Specific Columns

0s

```
df_pyspark.select('Name').show()
df_pyspark.select(['Name','Experience']).show()
```

```
+-----+
|    Name|
+-----+
| Tabish|
|Tabish2|
|Tabish3|
|Tabish4|
|Tabish5|
+-----+
```

```
+-----+-----+
|    Name|Experience|
+-----+-----+
| Tabish|      2|
|Tabish2|      2|
|Tabish3|      4|
|Tabish4|      7|
|Tabish5|      1|
+-----+-----+
```

Step 8:

To check datatypes

```
✓ [23] df_pyspark.dtypes  
0s ➔ [('Name', 'string'), ('Age', 'int'), ('Salary', 'int'), ('Experience', 'int')]
```

To describe the dataset

```
✓ 2s ⏴ df_pyspark.describe().show()  
➔ +-----+-----+-----+-----+  
| summary | Name | Age | Salary | Experience |  
+-----+-----+-----+-----+  
| count | 5 | 5 | 5 | 5 |  
| mean | NULL | 35.4 | 6000.0 | 3.2 |  
| stddev | NULL | 14.117365193264641 | 2236.06797749979 | 2.387467277262665 |  
| min | Tabish | 22 | 3000 | 1 |  
| max | Tabish5 | 55 | 9000 | 7 |  
+-----+-----+-----+-----+
```

Step 9:

Adding columns in data frame

```
[25] df_pyspark=df_pyspark.withColumn('Experience after 2 years', df_pyspark['Experience']+2)
```

Display the info

```
[26] df_pyspark.show()
```

```
+---+---+---+-----+
| Name|Age|Salary|Experience|Experience after 2 years|
+---+---+---+-----+
| Tabish| 22| 3000|     2|          4|
| Tabish2| 44| 6000|     2|          4|
| Tabish3| 23| 7000|     4|          6|
| Tabish4| 33| 9000|     7|          9|
| Tabish5| 55| 5000|     1|          3|
+---+---+---+-----+
```

Drop columns

```
[27] df_pyspark=df_pyspark.drop('Experience after 2 years').show()
```

```
+---+---+---+-----+
| Name|Age|Salary|Experience|
+---+---+---+-----+
| Tabish| 22| 3000|     2|
| Tabish2| 44| 6000|     2|
| Tabish3| 23| 7000|     4|
| Tabish4| 33| 9000|     7|
| Tabish5| 55| 5000|     1|
+---+---+---+-----+
```

Step 10:

```
Renaming columns

[55] df_pyspark.withColumnRenamed('Name','New Name')
df_pyspark.show()

+---+---+---+
| Name|Age|Salary|Experience|
+---+---+---+
| Tabish| 22| 3000|      2|
|Tabish2| 44| 6000|      2|
|Tabish3| 23| 7000|      4|
|Tabish4| 33| 9000|      7|
|Tabish5| 55| 5000|      1|
+---+---+---+


[56] df_pyspark.na.drop().show()

+---+---+---+
| Name|Age|Salary|Experience|
+---+---+---+
| Tabish| 22| 3000|      2|
|Tabish2| 44| 6000|      2|
|Tabish3| 23| 7000|      4|
|Tabish4| 33| 9000|      7|
|Tabish5| 55| 5000|      1|
+---+---+---+


[57] df_pyspark.na.drop(how='all').show()

+---+---+---+
| Name|Age|Salary|Experience|
+---+---+---+
| Tabish| 22| 3000|      2|
|Tabish2| 44| 6000|      2|
|Tabish3| 23| 7000|      4|
|Tabish4| 33| 9000|      7|
|Tabish5| 55| 5000|      1|
+---+---+---+
```

Step 11:

```
Renaming columns

[58] df_pyspark=df_pyspark.withColumnRenamed('Name','New Name')
df_pyspark.show()

+---+---+---+
|New Name|Age|Salary|Experience|
+---+---+---+
| Tabish| 22| 3000|      2|
| Tabish2| 44| 6000|      2|
| Tabish3| 23| 7000|      4|
| Tabish4| 33| 9000|      7|
| Tabish5| 55| 5000|      1|
| Tabish| 33| NULL|      |
+---+---+---+
```

Step 12:

```
✓ [129] df_pyspark=df_pyspark.na.drop(how='all')
df_pyspark.show()
```

```
平淡无奇的输出
```

| New Name | Age | Salary | Experience |
|----------|-----|--------|------------|
| Tabish | 22 | 3000 | 2 |
| Tabish2 | 44 | 6000 | 2 |
| Tabish3 | 23 | 7000 | 4 |
| Tabish4 | 33 | 9000 | 7 |
| Tabish5 | 55 | 5000 | 1 |
| Tabish6 | 33 | NULL | |
| Tabish7 | 35 | NULL | NULL |

```
✓ [130] df_pyspark.na.drop(how='any')
df_pyspark.show()
```

```
平淡无奇的输出
```

| New Name | Age | Salary | Experience |
|----------|-----|--------|------------|
| Tabish | 22 | 3000 | 2 |
| Tabish2 | 44 | 6000 | 2 |
| Tabish3 | 23 | 7000 | 4 |
| Tabish4 | 33 | 9000 | 7 |
| Tabish5 | 55 | 5000 | 1 |
| Tabish6 | 33 | NULL | |
| Tabish7 | 35 | NULL | NULL |

Step 13:

threshold

```
✓ 0s [183] df_pyspark.na.drop(how='any', thresh=2)  
df_pyspark.show()
```

```
→ +---+---+---+  
| Name|Age|Salary|Experience|  
+---+---+---+  
| Tabish| 22| 3000| 2|  
| Tabish2| 44| 6000| 2|  
| Tabish3| 23| 7000| 4|  
| Tabish4| 33| 9000| 7|  
| Tabish5| 55| 5000| 1|  
| Tabish6| 33| NULL| |  
| Tabish7| 35| NULL| NULL|  
+---+---+---+
```

subset

```
✓ 0s [184] df_pyspark=df_pyspark.na.drop(how='any', subset=['Experience']).show()
```

```
→ +---+---+---+  
| Name|Age|Salary|Experience|  
+---+---+---+  
| Tabish| 22| 3000| 2|  
| Tabish2| 44| 6000| 2|  
| Tabish3| 23| 7000| 4|  
| Tabish4| 33| 9000| 7|  
| Tabish5| 55| 5000| 1|  
| Tabish6| 33| NULL| |  
+---+---+---+
```

Step 14:

Filling the missing value

```
✓ 0s df_pyspark.na.fill('Missing Values').show()
```

| Name | Age | Salary | Experience |
|---------|-----|--------|----------------|
| Tabish | 22 | 3000 | 2 |
| Tabish2 | 44 | 6000 | 2 |
| Tabish3 | 23 | 7000 | 4 |
| Tabish4 | 33 | 9000 | 7 |
| Tabish5 | 55 | 5000 | 1 |
| Tabish6 | 33 | NULL | |
| Tabish7 | 35 | NULL | Missing Values |

Step 15:

Writing the function to calculate the value at the place of Null

```
✓ 0s [208] from pyspark.ml.feature import Imputer  
  
imputer= Imputer(  
    inputCols=['Age','Salary'],  
    outputCols=["{}_imputed".format(c) for c in ['Age','Salary']]).setStrategy('mean')
```

Add imputation cols to df

```
✓ 1s df_pyspark = imputer.fit(df_pyspark).transform(df_pyspark).show()
```

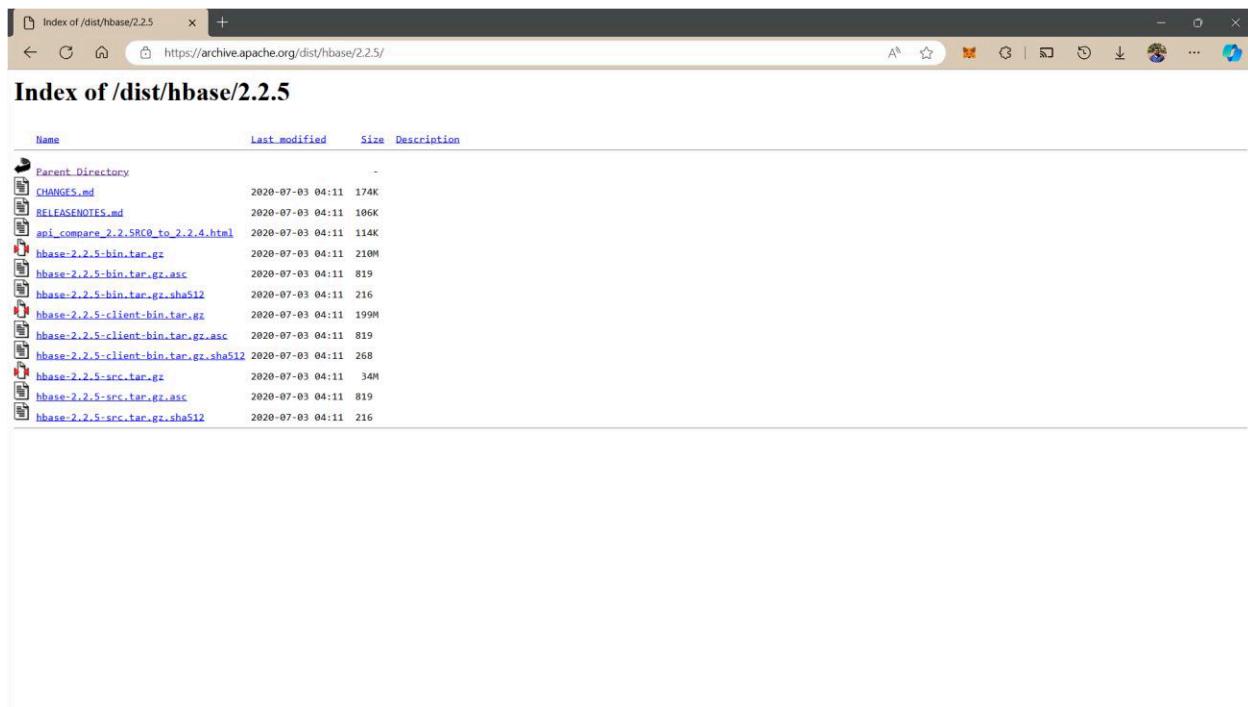
| Name | Age | Salary | Experience | Age_imputed | Salary_imputed |
|---------|-----|--------|------------|-------------|----------------|
| Tabish | 22 | 3000 | 2 | 22 | 3000 |
| Tabish2 | 44 | 6000 | 2 | 44 | 6000 |
| Tabish3 | 23 | 7000 | 4 | 23 | 7000 |
| Tabish4 | 33 | 9000 | 7 | 33 | 9000 |
| Tabish5 | 55 | 5000 | 1 | 55 | 5000 |
| Tabish6 | 33 | NULL | | 33 | 6000 |
| Tabish7 | 35 | NULL | NULL | 35 | 6000 |

Practical 5:

Aim: Download and installation of HBase

Steps:

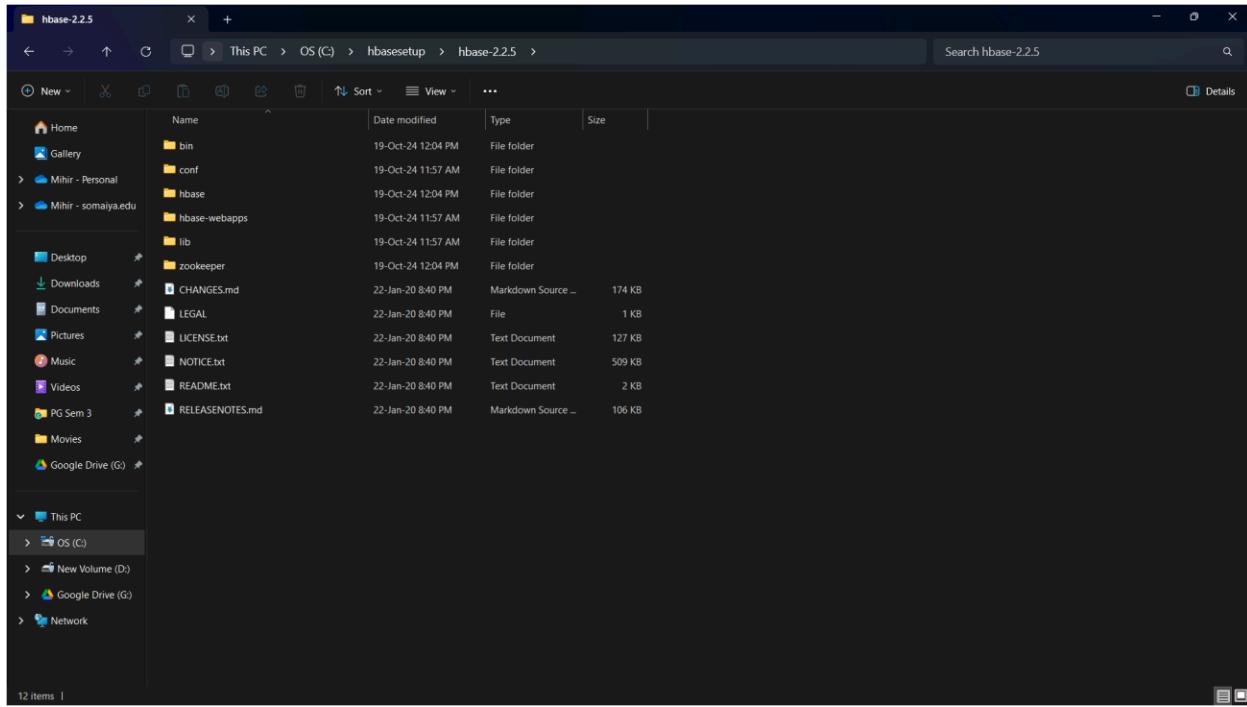
- 1- Download HBase bin file from [here](#).



The screenshot shows a Windows 10 desktop with a Microsoft Edge browser window open. The address bar displays the URL: <https://archive.apache.org/dist/hbase/2.2.5/>. The page title is "Index of /dist/hbase/2.2.5". Below the title is a table listing files and their details. The columns are: Name, Last_modified, Size, and Description. The table includes entries for CHANGE5.md, RELEASENOTES.md, api_compare_2.2_SRC0_to_2.2.4.html, hbase-2.2.5-bin.tar.gz, hbase-2.2.5-bin.tar.gz.asc, hbase-2.2.5-bin.tar.gz.sha512, hbase-2.2.5-client-bin.tar.gz, hbase-2.2.5-client-bin.tar.gz.asc, hbase-2.2.5-client-bin.tar.gz.sha512, hbase-2.2.5-src.tar.gz, hbase-2.2.5-src.tar.gz.asc, and hbase-2.2.5-src.tar.gz.sha512. All files were last modified on 2020-07-03 at 04:11, with sizes ranging from 819 to 210M.

| Name | Last_modified | Size | Description |
|--|------------------|------|-------------|
| Parent Directory | - | | |
| CHANGE5.md | 2020-07-03 04:11 | 174K | |
| RELEASENOTES.md | 2020-07-03 04:11 | 106K | |
| api_compare_2.2_SRC0_to_2.2.4.html | 2020-07-03 04:11 | 114K | |
| hbase-2.2.5-bin.tar.gz | 2020-07-03 04:11 | 210M | |
| hbase-2.2.5-bin.tar.gz.asc | 2020-07-03 04:11 | 819 | |
| hbase-2.2.5-bin.tar.gz.sha512 | 2020-07-03 04:11 | 216 | |
| hbase-2.2.5-client-bin.tar.gz | 2020-07-03 04:11 | 199M | |
| hbase-2.2.5-client-bin.tar.gz.asc | 2020-07-03 04:11 | 819 | |
| hbase-2.2.5-client-bin.tar.gz.sha512 | 2020-07-03 04:11 | 268 | |
| hbase-2.2.5-src.tar.gz | 2020-07-03 04:11 | 34M | |
| hbase-2.2.5-src.tar.gz.asc | 2020-07-03 04:11 | 819 | |
| hbase-2.2.5-src.tar.gz.sha512 | 2020-07-03 04:11 | 216 | |

- 2- Create a new folder in C drive named hbasesetup. Extract the files and paste it in the hbasesetup. Create 2 new folders hbase and zookeeper inside.



3-Open the bin folder. Search for the file hbase.cmd and edit it in notepad. Search for java_arguments and remove %HEAP_SETTINGS%.

```
hbase.cmd
File Edit View
set HEAP_SETTINGS=%JAVA_HEAP_MAX% %JAVA_OFFHEAP_MAX%
set java_arguments=%HEAP_SETTINGS% %HBASE_OPTS% -classpath "%CLASSPATH%" %CLASS% %hbase-command-arguments%
```



```
hbase.cmd
File Edit View
set HEAP_SETTINGS=%JAVA_HEAP_MAX% %JAVA_OFFHEAP_MAX%
set java_arguments=%HBASE_OPTS% -classpath "%CLASSPATH%" %CLASS% %hbase-command-arguments%
```

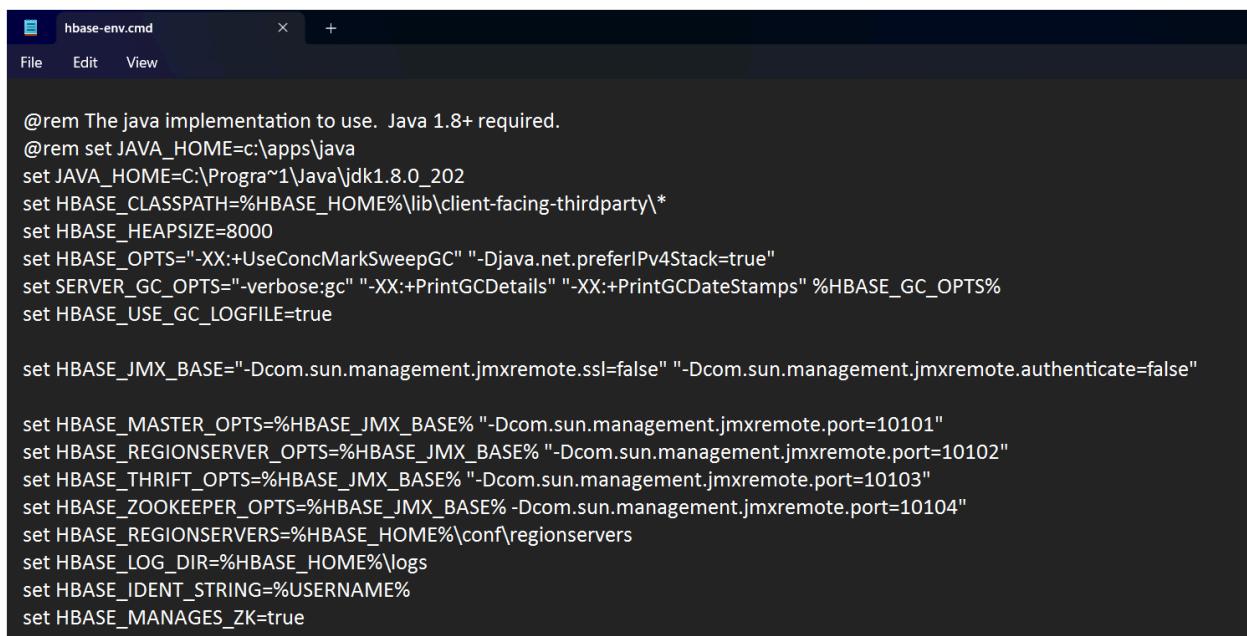
4-Open the conf folder and edit the file hbase-env.cmd in notepad. Add the following lines inside.

```
set JAVA_HOME=C:\Program~1\Java\jdk1.8.0_202
set HBASE_CLASSPATH=%HBASE_HOME%\lib\client-facing-thirdparty\*
set HBASE_HEAPSIZE=8000
set HBASE_OPTS="-XX:+UseConcMarkSweepGC" "-Djava.net.preferIPv4Stack=true"
set SERVER_GC_OPTS="-verbose:gc" "-XX:+PrintGCDetails" "-XX:+PrintGCDateStamps"
%HBASE_GC_OPTS%
set HBASE_USE_GC_LOGFILE=true
```

```

set HBASE_JMX_BASE="-Dcom.sun.management.jmxremote.ssl=false"
"-Dcom.sun.management.jmxremote.authenticate=false"
set HBASE_MASTER_OPTS=%HBASE_JMX_BASE% "-Dcom.sun.management.jmxremote.port=10101"
set HBASE_REGIONSERVER_OPTS=%HBASE_JMX_BASE%
"-Dcom.sun.management.jmxremote.port=10102"
set HBASE_THRIFT_OPTS=%HBASE_JMX_BASE% "-Dcom.sun.management.jmxremote.port=10103"
set HBASE_ZOOKEEPER_OPTS=%HBASE_JMX_BASE%
-Dcom.sun.management.jmxremote.port=10104"
set HBASE_REGIONSERS=%HBASE_HOME%\conf\regionservers
set HBASE_LOG_DIR=%HBASE_HOME%\logs
set HBASE_IDENT_STRING=%USERNAME%
set HBASE_MANAGES_ZK=true

```



```

@rem The java implementation to use. Java 1.8+ required.
@rem set JAVA_HOME=c:\apps\java
set JAVA_HOME=C:\Program Files\Java\jdk1.8.0_202
set HBASE_CLASSPATH=%HBASE_HOME%\lib\client-facing-thirdparty\*
set HBASE_HEAPSIZE=8000
set HBASE_OPTS="-XX:+UseConcMarkSweepGC" "-Djava.net.preferIPv4Stack=true"
set SERVER_GC_OPTS="-verbose:gc" "-XX:+PrintGCDetails" "-XX:+PrintGCDateStamps" %HBASE_GC_OPTS%
set HBASE_USE_GC_LOGFILE=true

set HBASE_JMX_BASE="-Dcom.sun.management.jmxremote.ssl=false" "-Dcom.sun.management.jmxremote.authenticate=false"

set HBASE_MASTER_OPTS=%HBASE_JMX_BASE% "-Dcom.sun.management.jmxremote.port=10101"
set HBASE_REGIONSERVER_OPTS=%HBASE_JMX_BASE% "-Dcom.sun.management.jmxremote.port=10102"
set HBASE_THRIFT_OPTS=%HBASE_JMX_BASE% "-Dcom.sun.management.jmxremote.port=10103"
set HBASE_ZOOKEEPER_OPTS=%HBASE_JMX_BASE% -Dcom.sun.management.jmxremote.port=10104"
set HBASE_REGIONSERS=%HBASE_HOME%\conf\regionservers
set HBASE_LOG_DIR=%HBASE_HOME%\logs
set HBASE_IDENT_STRING=%USERNAME%
set HBASE_MANAGES_ZK=true

```

5-Open the conf folder and edit the file hbase-site.xml in notepad. Add the following lines after the last property tag.

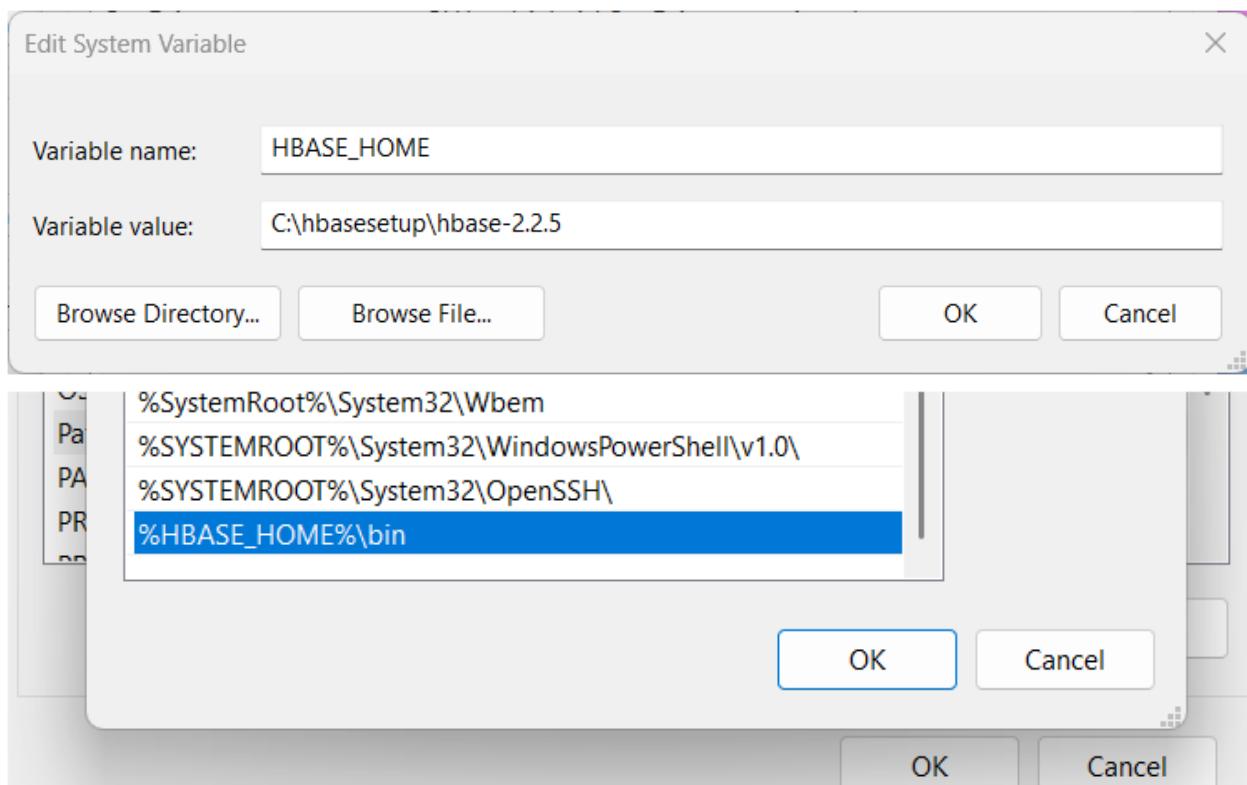
```

<property>
<name>hbase.rootdir</name>
<value>file:///C:/hbasesetup/hbase-2.2.5/hbase</value>
</property>
<property>
<name>hbase.zookeeper.property.dataDir</name>
<value>/C:/hbasesetup/hbase-2.2.5/zookeeper</value>
</property>
<property>
<name> hbase.zookeeper.quorum</name>
<value>localhost</value>
</property>

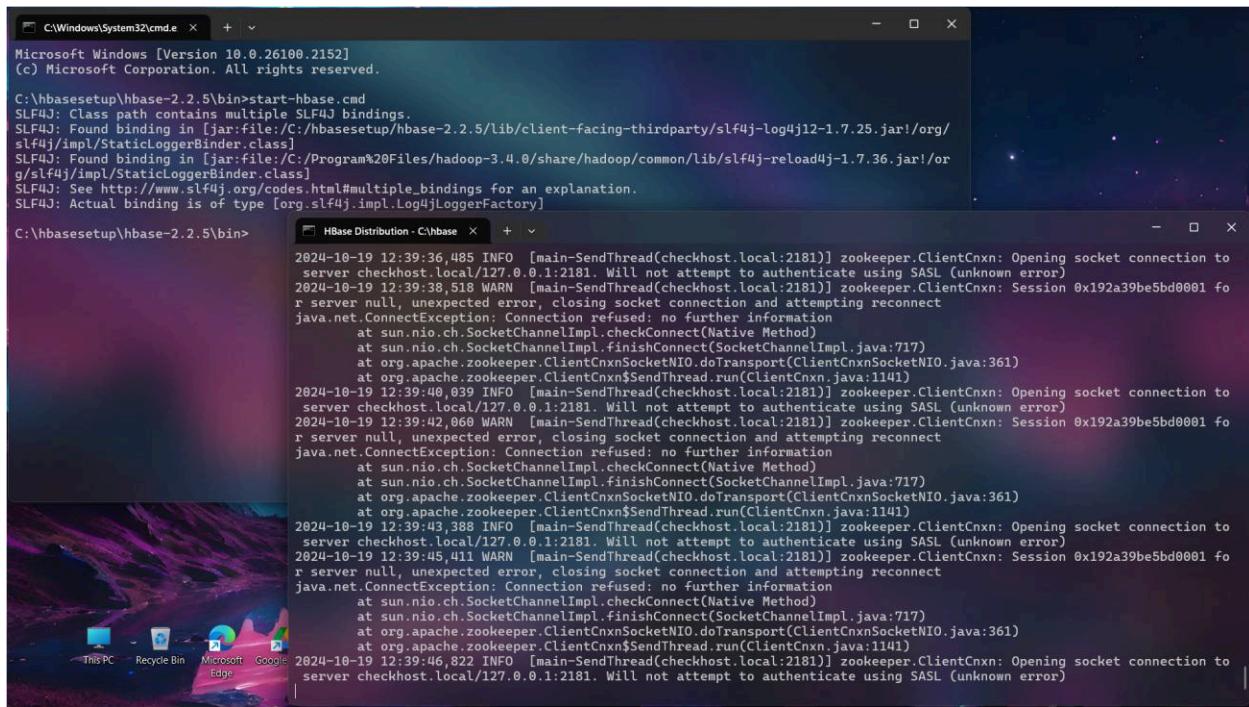
```

```
hbase-site.xml +  
File Edit View  
See also https://hbase.apache.org/book.html#standalone_dist  
-->  
<property>  
<name>hbase.cluster.distributed</name>  
<value>false</value>  
</property>  
<property>  
<name>hbase.tmp.dir</name>  
<value>./tmp</value>  
</property>  
<property>  
<name>hbase.unsafe.stream.capability.enforce</name>  
<value>false</value>  
</property>  
<property>  
<name>hbase.rootdir</name>  
<value>file:///C:/hbasesetup/hbase-2.2.5/hbase</value>  
</property>  
<property>  
<name>hbase.zookeeper.property.dataDir</name>  
<value>C:/hbasesetup/hbase-2.2.5/zookeeper</value>  
</property>  
<property>  
<name> hbase.zookeeper.quorum</name>  
<value>localhost</value>  
</property>  
</configuration>  
  
Ln 66, Col 17 2,642 characters 100% Unix (LF) UTF-8
```

6-Set the HBase environment variables and it too path as well.



7-Open command prompt and navigate to the hbase bin folder. Run the command start-hbase.cmd to start hbase



The screenshot shows two windows. The top window is titled 'C:\Windows\System32\cmd.e' and contains the command 'start-hbase.cmd'. It displays SLF4J binding logs and ends with the command 'C:\hbasesetup\hbase-2.2.5\bin>'. The bottom window is titled 'HBase Distribution - C:\hbase' and shows the output of the 'jps' command. It lists two processes: '3744 HMaster' and '22780 Jps'.

```
C:\hbasesetup\hbase-2.2.5\bin>start-hbase.cmd
Microsoft Windows [Version 10.0.26100.2152]
(c) Microsoft Corporation. All rights reserved.

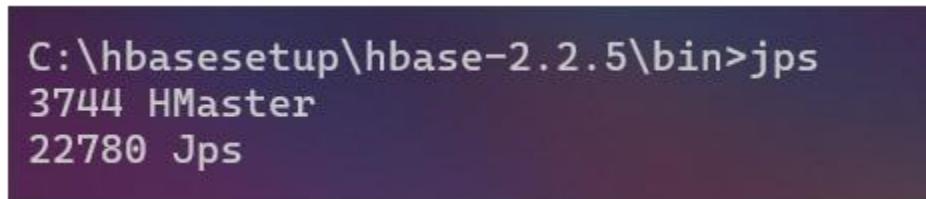
C:\hbasesetup\hbase-2.2.5\bin>SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/hbasesetup/hbase-2.2.5/lib/client-facing-thirdparty/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/C:/Program%20Files/hadoop-3.4.0/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

C:\hbasesetup\hbase-2.2.5\bin>jps
2024-10-19 12:39:36,485 INFO  [main-SendThread(checkhost.local:2181)] zookeeper.ClientCnxn: Opening socket connection to server checkhost.local/127.0.0.1:2181. Will not attempt to authenticate using SASL (unknown error)
2024-10-19 12:39:38,518 WARN  [main-SendThread(checkhost.local:2181)] zookeeper.ClientCnxn: Session 0x192a39be5bd0001 for server null, unexpected error, closing socket connection and attempting reconnect
java.net.ConnectException: Connection refused: no further information
    at sun.nio.ch.SocketChannelImpl.checkConnect(Native Method)
    at sun.nio.ch.SocketChannelImpl.finishConnect(SocketChannelImpl.java:717)
    at org.apache.zookeeper.ClientCnxn$SendThread.run(ClientCnxn.java:1141)
2024-10-19 12:39:42,039 INFO  [main-SendThread(checkhost.local:2181)] zookeeper.ClientCnxn: Opening socket connection to server checkhost.local/127.0.0.1:2181. Will not attempt to authenticate using SASL (unknown error)
2024-10-19 12:39:42,060 WARN  [main-SendThread(checkhost.local:2181)] zookeeper.ClientCnxn: Session 0x192a39be5bd0001 for server null, unexpected error, closing socket connection and attempting reconnect
java.net.ConnectException: Connection refused: no further information
    at sun.nio.ch.SocketChannelImpl.checkConnect(Native Method)
    at sun.nio.ch.SocketChannelImpl.finishConnect(SocketChannelImpl.java:717)
    at org.apache.zookeeper.ClientCnxn$SendThread.run(ClientCnxn.java:1141)
2024-10-19 12:39:43,388 INFO  [main-SendThread(checkhost.local:2181)] zookeeper.ClientCnxn: Opening socket connection to server checkhost.local/127.0.0.1:2181. Will not attempt to authenticate using SASL (unknown error)
2024-10-19 12:39:45,411 WARN  [main-SendThread(checkhost.local:2181)] zookeeper.ClientCnxn: Session 0x192a39be5bd0001 for server null, unexpected error, closing socket connection and attempting reconnect
java.net.ConnectException: Connection refused: no further information
    at sun.nio.ch.SocketChannelImpl.checkConnect(Native Method)
    at sun.nio.ch.SocketChannelImpl.finishConnect(SocketChannelImpl.java:717)
    at org.apache.zookeeper.ClientCnxn$SendThread.run(ClientCnxn.java:1141)
2024-10-19 12:39:46,822 INFO  [main-SendThread(checkhost.local:2181)] zookeeper.ClientCnxn: Opening socket connection to server checkhost.local/127.0.0.1:2181. Will not attempt to authenticate using SASL (unknown error)

C:\hbasesetup\hbase-2.2.5\bin>
```

```
C:\hbasesetup\hbase-2.2.5\bin>jps
3744 HMaster
22780 Jps
```

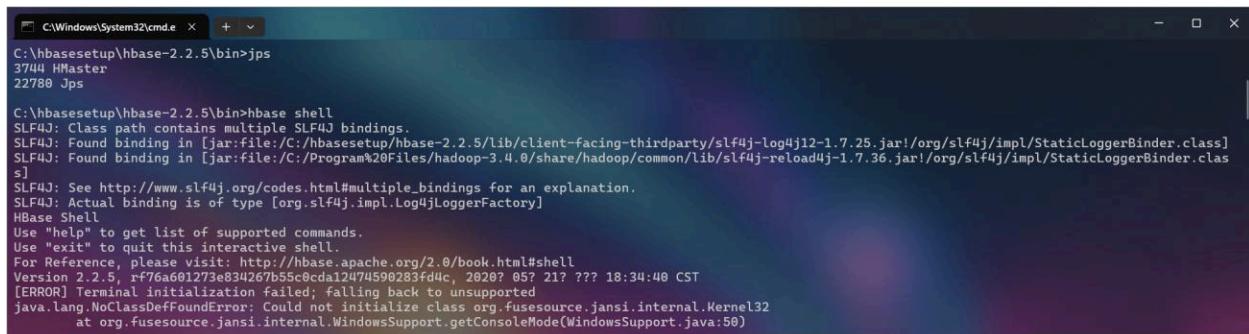
8-Using the command jps you can check that our HMaster is running.



The screenshot shows a single command prompt window with the title 'C:\Windows\System32\cmd.e'. It displays the output of the 'jps' command, which shows two processes: '3744 HMaster' and '22780 Jps'.

```
C:\hbasesetup\hbase-2.2.5\bin>jps
3744 HMaster
22780 Jps
```

9-Start the HBase Shell now with the command hbase shell. Initial startup may take some time.



The screenshot shows a command prompt window with the title 'C:\Windows\System32\cmd.e'. It displays the command 'hbase shell' being run. The output includes SLF4J binding logs, a warning about terminal initialization failing, and a Java exception related to the 'org.fusesource.jansi.internal.WindowsSupport.getConsoleMode' method.

```
C:\hbasesetup\hbase-2.2.5\bin>hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/hbasesetup/hbase-2.2.5/lib/client-facing-thirdparty/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/C:/Program%20Files/hadoop-3.4.0/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]

HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.2.5, rf76a001273e034267b55c0cd124740590283fd4c, 2020-05-21T18:34:40 CST
[ERROR] Terminal initialization failed; falling back to unsupported
java.lang.NoClassDefFoundError: Could not initialize class org.fusesource.jansi.internal.WindowsSupport
    at org.fusesource.jansi.internal.WindowsSupport.getConsoleMode(WindowsSupport.java:50)
```

10-Your HBase Shell has been started. Ignore the warnings received while starting the shell.

```
C:\Windows\System32\cmd.e + - x
at org.jruby.runtime.callsite.CachingCallSite.call(CachingCallSite.java:131)
at C_3a_.hbasesetup.hbase_minus_2_dot_2_dot_5.bin.hirb.invokeOther172:print_banner(C:\hbasesetup\hbase-2.2.5\bin\hirb.rb:190)
at C_3a_.hbasesetup.hbase_minus_2_dot_2_dot_5.bin.hirb.RUBY$scriptC:\hbasesetup\hbase-2.2.5\bin\hirb.rb:190)
at java.lang.invoke.MethodHandle.invokeWithArguments(MethodHandle.java:627)
at org.jruby.ir.Compiler$1.load(Compiler.java:95)
at org.jruby.Ruby.runScript(Ruby.java:828)
at org.jruby.Ruby.runNormally(Ruby.java:765)
at org.jruby.Ruby.runFromMain(Ruby.java:578)
at org.jruby.Main.doRunFromMain(Main.java:417)
at org.jruby.Main.internalRun(Main.java:305)
at org.jruby.Main.run(Main.java:232)
at org.jruby.Main.main(Main.java:204)

Took 0.0040 seconds
'sty' is not recognized as an internal or external command,
operable program or batch file.
hbase(main):001:0> |
```