

Big Data Analytics Journal

Submitted By

Name: Vatsal Parekh

Class: MSC CS (SEM-III)

Roll No: 31031523022

MSc CS – Part II

**Department Of Computer Science
Somaiya Vidyavihar University
SK Somaiya College**

Index:

Practical No	Title
1.	Installation of Hadoop on Windows
2.	Installation of Scala and Apache Spark
3.	Spark GraphX
4.	Working with PySpark
5.	Installation of HBase
6.	
7.	

Practical 1: Installation of Hadoop on Windows.

1. Download the latest Hadoop Binary version from [here](#).

The screenshot shows the Apache Hadoop Download page. At the top, there's a navigation bar with links for Apache Hadoop, Download, Documentation, Community, Development, Help, and Apache Software Foundation. Below the navigation is a section titled "Download" which states: "Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mirror sites and should be checked for tampering using GPG or SHA-512." A table lists three binary releases:

Version	Release date	Source download	Binary download	Release notes
3.4.0	2024 Mar 17	source (checksum signature)	binary (checksum signature) binary-aarch64 (checksum signature)	Announcement
3.3.6	2023 Jun 23	source (checksum signature)	binary (checksum signature) binary-aarch64 (checksum signature)	Announcement
2.10.2	2022 May 31	source (checksum signature)	binary (checksum signature)	Announcement

Below the table, instructions for verifying releases using GPG are provided:

1. Download the release hadoop-X.Y.Z-src.tar.gz from a mirror site.
2. Download the signature file hadoop-X.Y.Z-src.tar.gz.asc from Apache.
3. Download the Hadoop KEYS file.
4. gpg --import KEYS
5. gpg --verify hadoop-X.Y.Z-src.tar.gz.asc

Instructions for performing a quick check using SHA-512 are also present:

1. Download the release hadoop-X.Y.Z-src.tar.gz from a mirror site.
2. Download the checksum hadoop-X.Y.Z-src.tar.gz.sha512 or hadoop-X.Y.Z-src.tar.gz.md5 from Apache.
3. shasum -a 512 hadoop-X.Y.Z-src.tar.gz

Links for "License" and "The software licensed under Apache License 2.0" are at the bottom.

2. Download Java version 11.0.23 from [here](#).

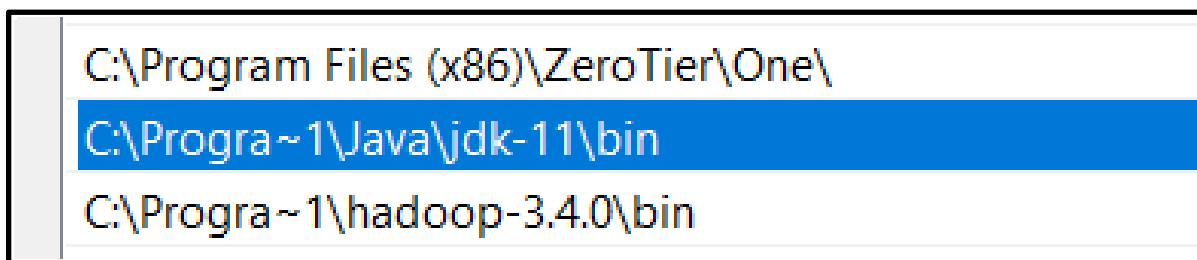
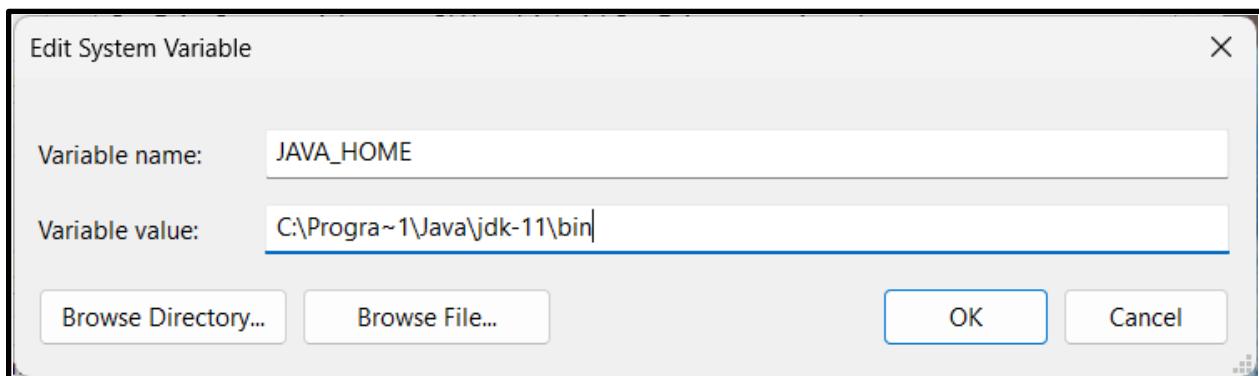
The screenshot shows the Oracle Java SE Development Kit 11.0.23 download page. At the top, there's a navigation bar with links for Products, Industries, Resources, Customers, Partners, Developers, Company, a search bar, and buttons for View Accounts and Contact Sales. Below the navigation is a section for "Java SE Development Kit 11.0.23". It states: "This software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE". A link "JDK 11.0.23 checksum" is present. A table lists various Java package options:

Product / File Description	File Size	Download
Linux ARM64 RPM Package	159.58 MB	jdk-11.0.23_linux-aarch64_bin.rpm
Linux ARM64 Compressed Archive	159.69 MB	jdk-11.0.23_linux-aarch64_bin.tar.gz
Linux x64 Debian Package	138.64 MB	jdk-11.0.23_linux-x64_bin.deb
Linux x64 RPM Package	160.99 MB	jdk-11.0.23_linux-x64_bin.rpm
Linux x64 Compressed Archive	161.10 MB	jdk-11.0.23_linux-x64_bin.tar.gz
macOS ARM64 Compressed Archive	154.27 MB	jdk-11.0.23_macos-aarch64_bin.tar.gz
macOS ARM64 DMG Installer	153.75 MB	jdk-11.0.23_macos-aarch64_bin.dmg

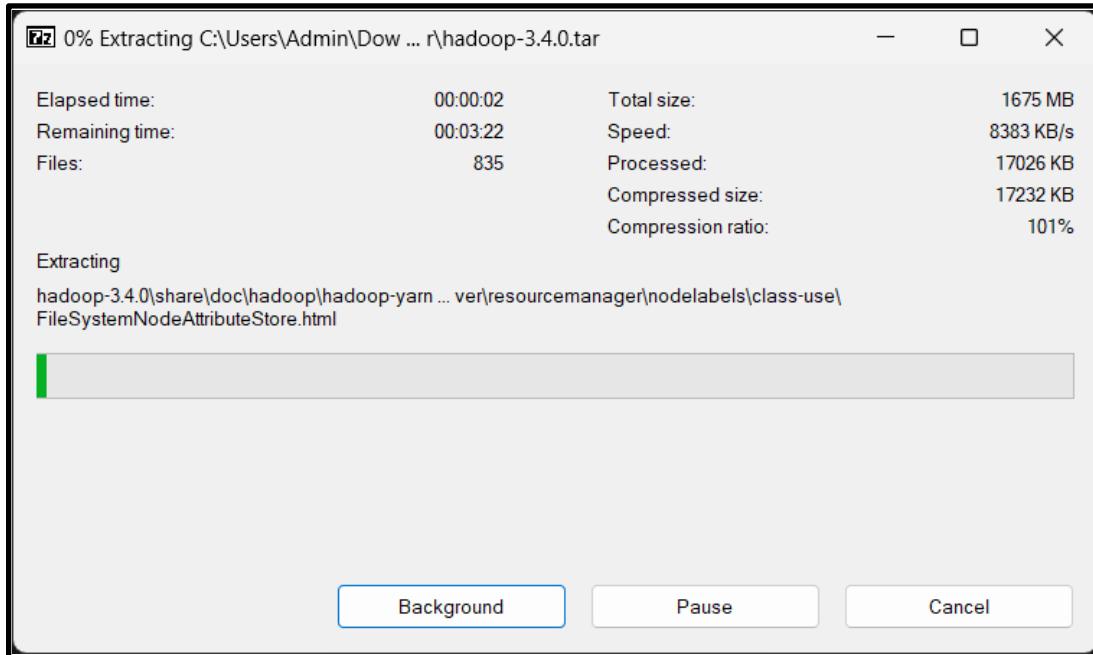
3. Run the installer and follow the steps to install Java.



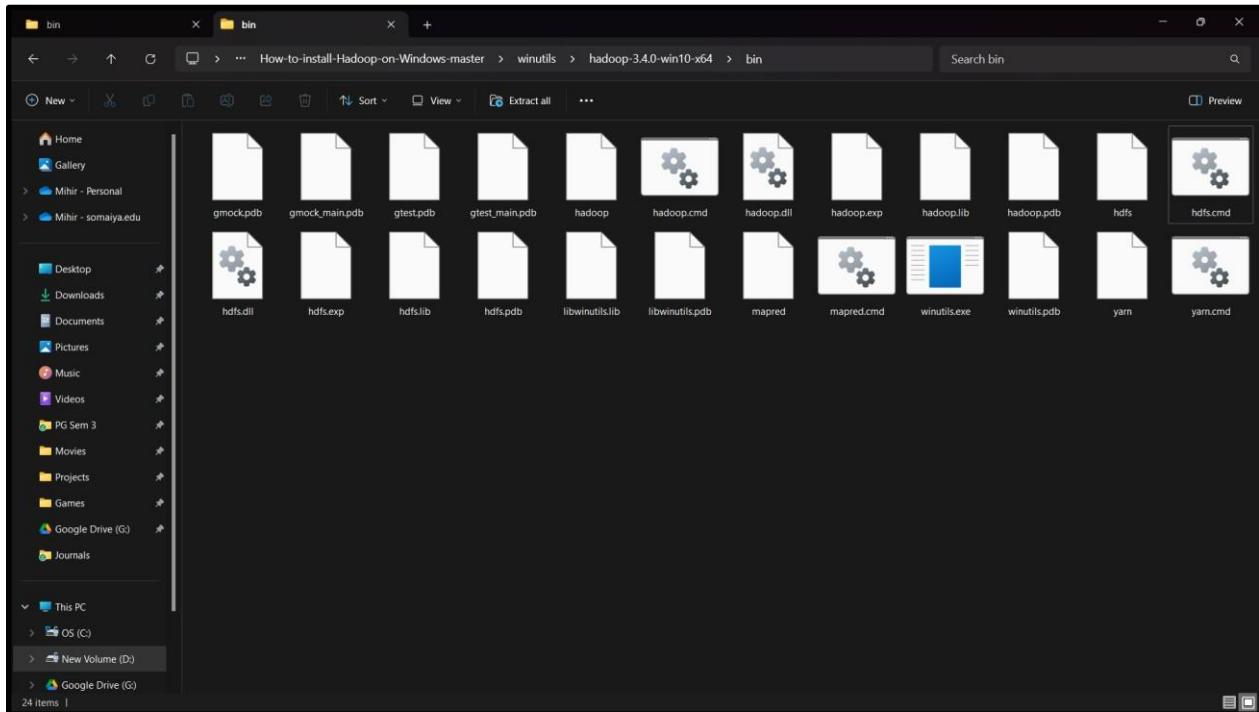
4: Make sure to set the Java Environment Variables by creating a new variable and also adding it to the path variable.



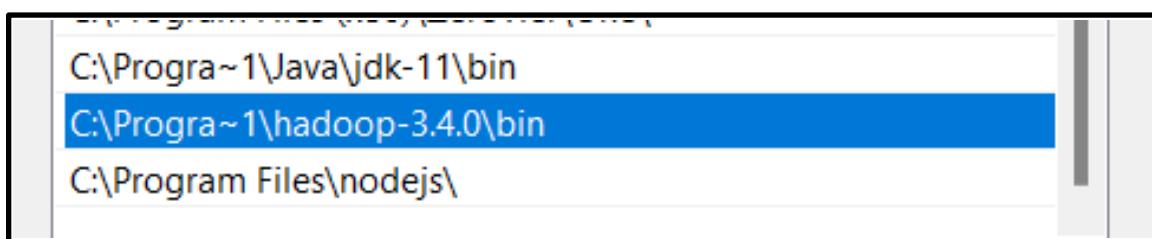
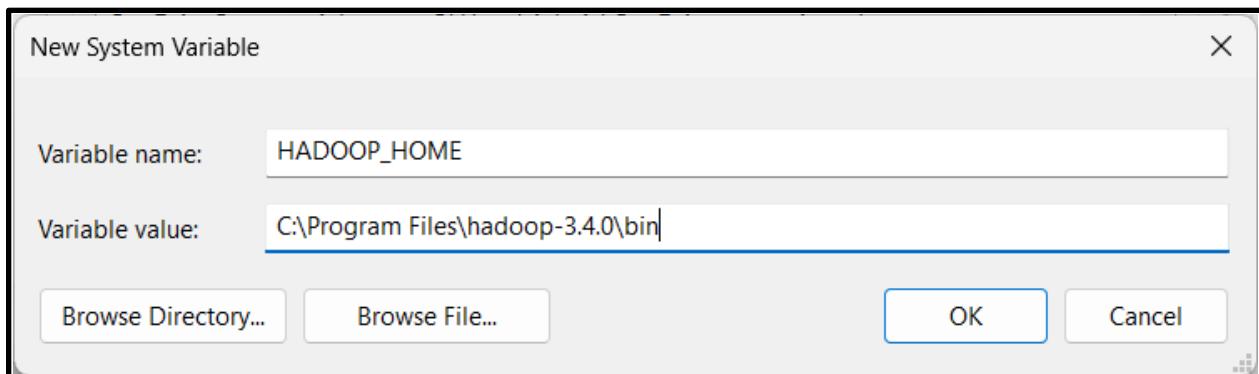
5. Extract the previously downloaded Hadoop archive and copy the contents to your desired location.



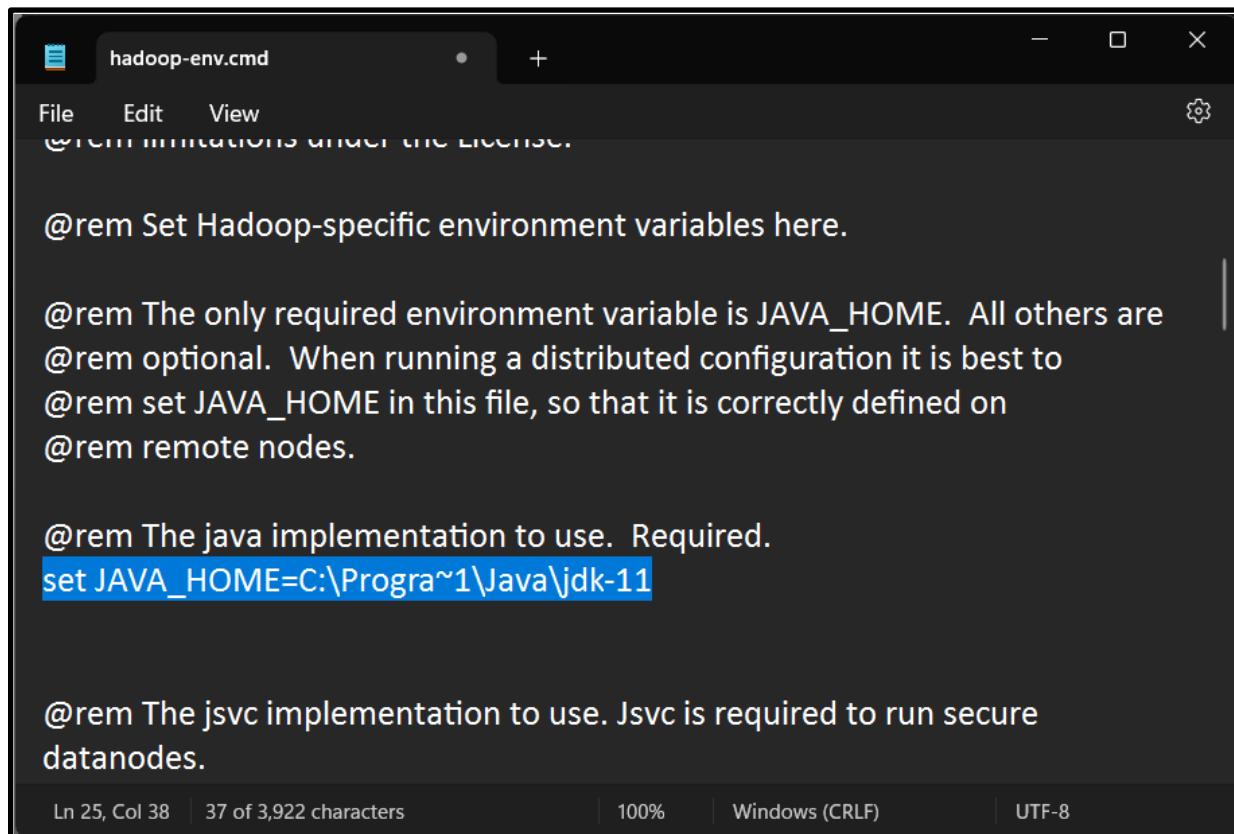
6. Download and Install Hadoop native IO binary from [here](#). To install copy all the contents of the archive to the bin folder of your Hadoop.



7. Set up the Hadoop Environment Variables.



8. Set the JAVA_HOME in the hadoop_env.cmd file located inside the etc folder.



```
@rem Set Hadoop-specific environment variables here.

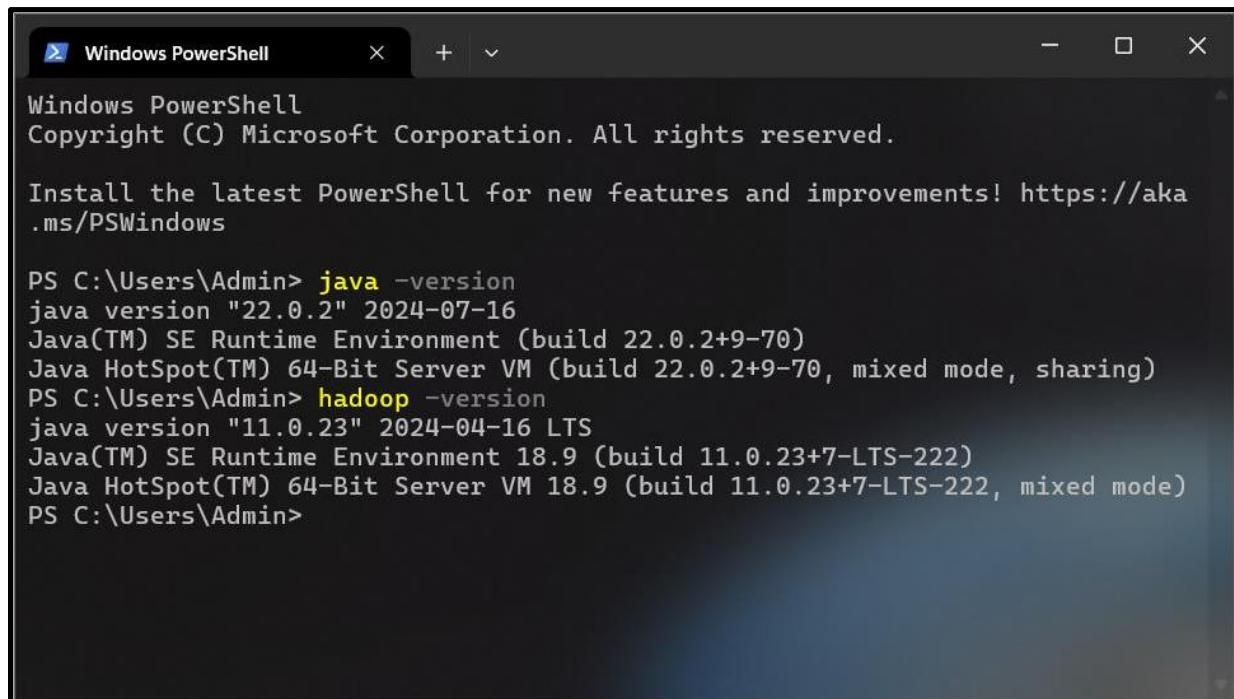
@rem The only required environment variable is JAVA_HOME. All others are
@rem optional. When running a distributed configuration it is best to
@rem set JAVA_HOME in this file, so that it is correctly defined on
@rem remote nodes.

@rem The java implementation to use. Required.
set JAVA_HOME=C:\Program Files\Java\jdk-11

@rem The jsvc implementation to use. Jsvc is required to run secure
datanodes.
```

Ln 25, Col 38 | 37 of 3,922 characters | 100% | Windows (CRLF) | UTF-8

9. Check the versions of both Java and Hadoop.

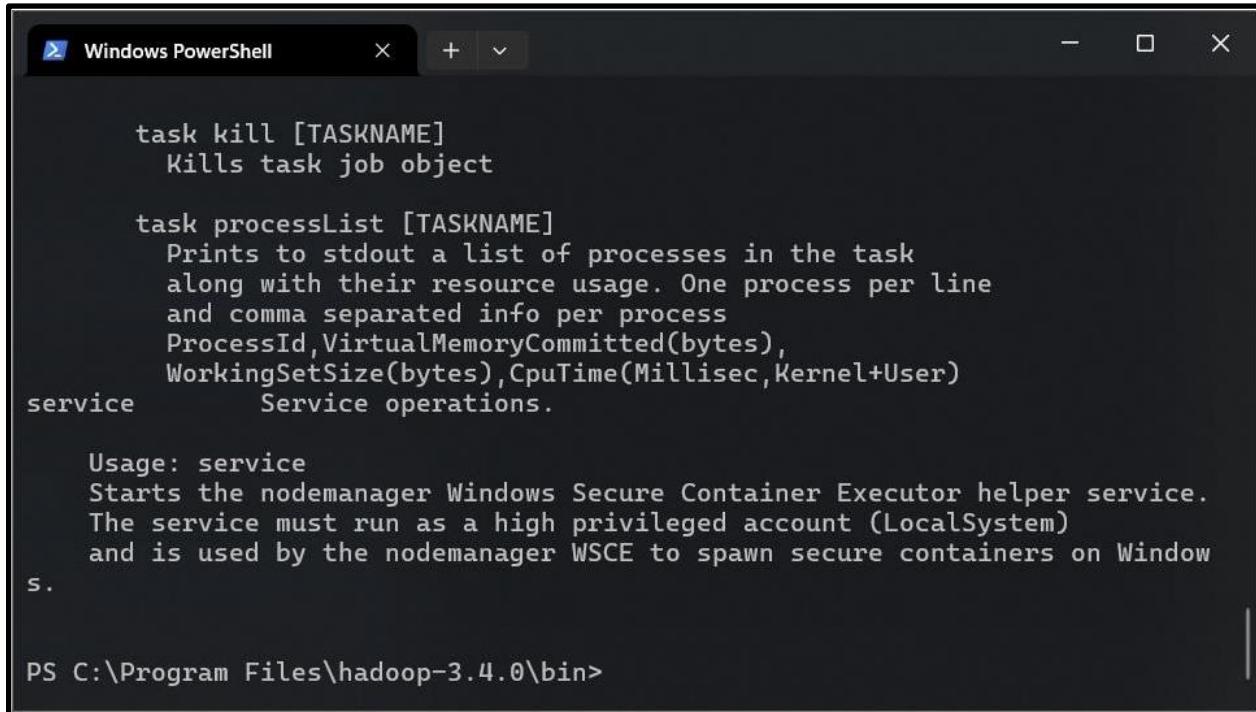


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Admin> java -version
java version "22.0.2" 2024-07-16
Java(TM) SE Runtime Environment (build 22.0.2+9-70)
Java HotSpot(TM) 64-Bit Server VM (build 22.0.2+9-70, mixed mode, sharing)
PS C:\Users\Admin> hadoop -version
java version "11.0.23" 2024-04-16 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.23+7-LTS-222)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.23+7-LTS-222, mixed mode)
PS C:\Users\Admin>
```

10. To verify all the above steps are completed successfully, open the bin folder in terminal and run the command `winutils.exe`



```
task kill [TASKNAME]
Kills task job object

task processList [TASKNAME]
Prints to stdout a list of processes in the task
along with their resource usage. One process per line
and comma separated info per process
ProcessId,VirtualMemoryCommitted(bytes),
WorkingSetSize(bytes),CpuTime(MilliseC,Kernel+User)

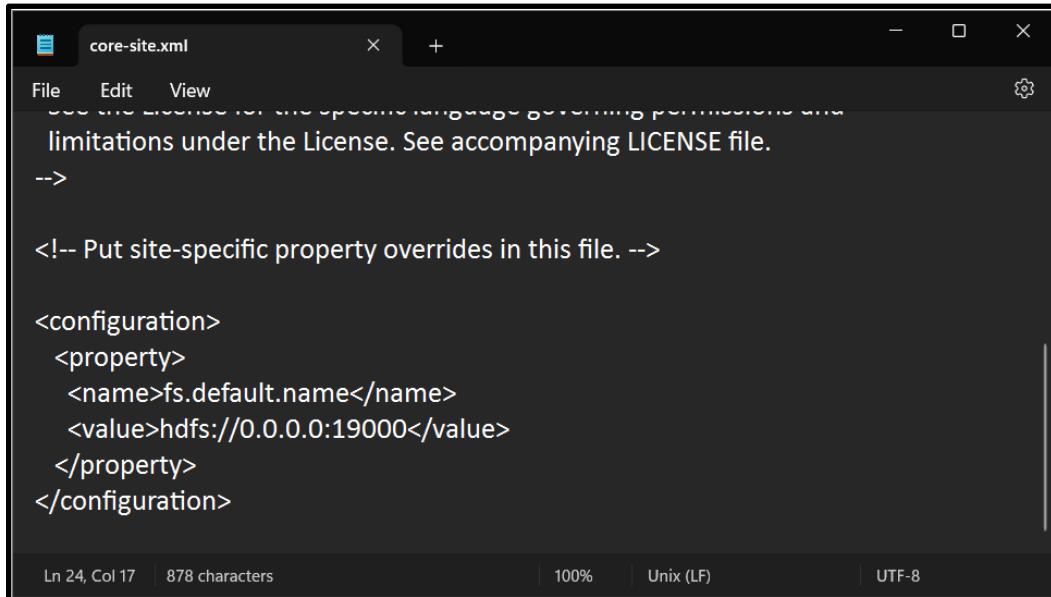
service      Service operations.

Usage: service
Starts the nodemanager Windows Secure Container Executor helper service.
The service must run as a high privileged account (LocalSystem)
and is used by the nodemanager WSCE to spawn secure containers on Window
s.

PS C:\Program Files\hadoop-3.4.0\bin>
```

11. Hadoop Configurations. All the files will be located in the etc folder of your Hadoop installation.

11a. Configure core-site (core-site.xml)



```
core-site.xml

File Edit View
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.

-->

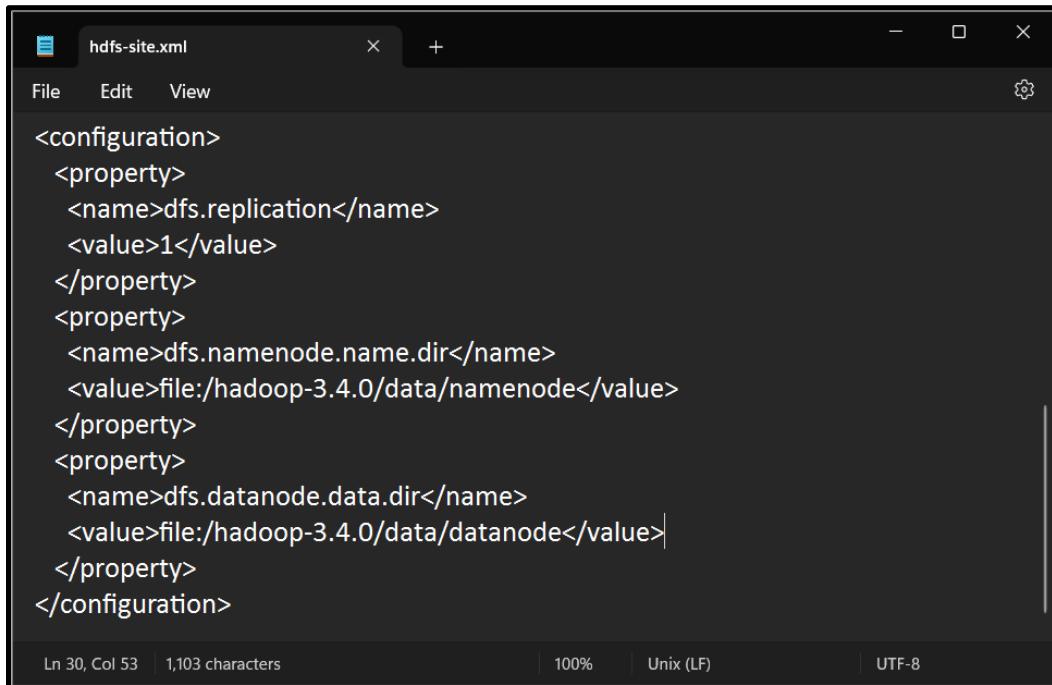
<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://0.0.0.0:19000</value>
</property>
</configuration>

Ln 24, Col 17  878 characters | 100% | Unix (LF) | UTF-8
```

11b. Configure HDFS (hdfs-site.xml)

Before configuring HDFS, make a folder called as `data` in your Hadoop installation. Make two subfolders named as `namenode` and `datanode`.

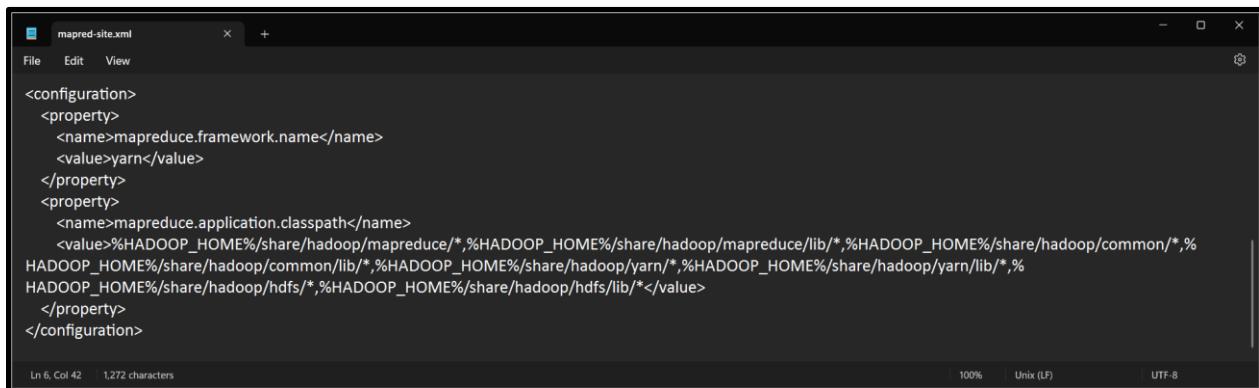


The screenshot shows a code editor window titled "hdfs-site.xml". The file contains XML configuration for HDFS. It includes properties for replication, namenode name directory, and datanode data directory. The code is as follows:

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/hadoop-3.4.0/data/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/hadoop-3.4.0/data/datanode</value>
</property>
</configuration>
```

At the bottom of the editor, status information is displayed: Ln 30, Col 53 | 1,103 characters | 100% | Unix (LF) | UTF-8.

11c. Configure MapReduce (mapred-site.xml)

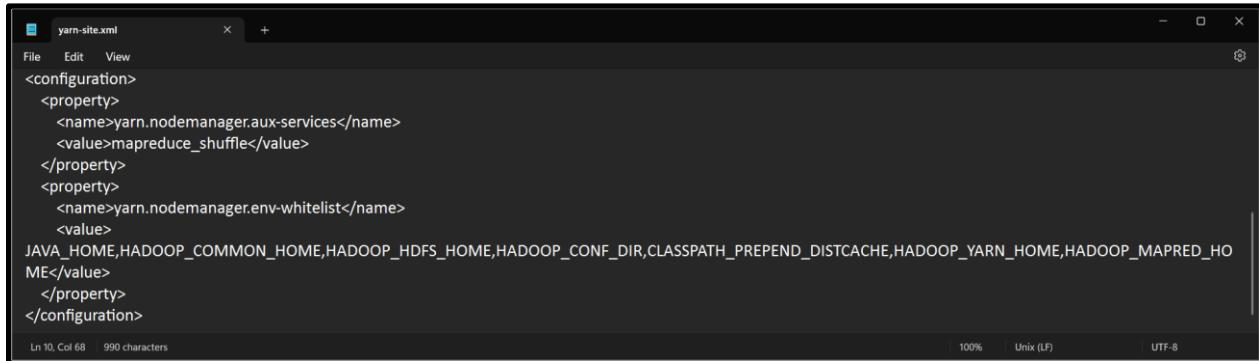


The screenshot shows a code editor window titled "mapred-site.xml". The file contains XML configuration for MapReduce. It includes a property for the framework name set to "yarn". The code is as follows:

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
<property>
<name>mapreduce.application.classpath</name>
<value>%HADOOP_HOME%/share/hadoop/mapreduce/*,%HADOOP_HOME%/share/hadoop/mapreduce/lib/*,%HADOOP_HOME%/share/hadoop/common/*,%HADOOP_HOME%/share/hadoop/common/lib/*,%HADOOP_HOME%/share/hadoop/yarn/*,%HADOOP_HOME%/share/hadoop/yarn/lib/*,%HADOOP_HOME%/share/hadoop/hdfs/*,%HADOOP_HOME%/share/hadoop/hdfs/lib/*</value>
</property>
</configuration>
```

At the bottom of the editor, status information is displayed: Ln 6, Col 42 | 1,272 characters | 100% | Unix (LF) | UTF-8.

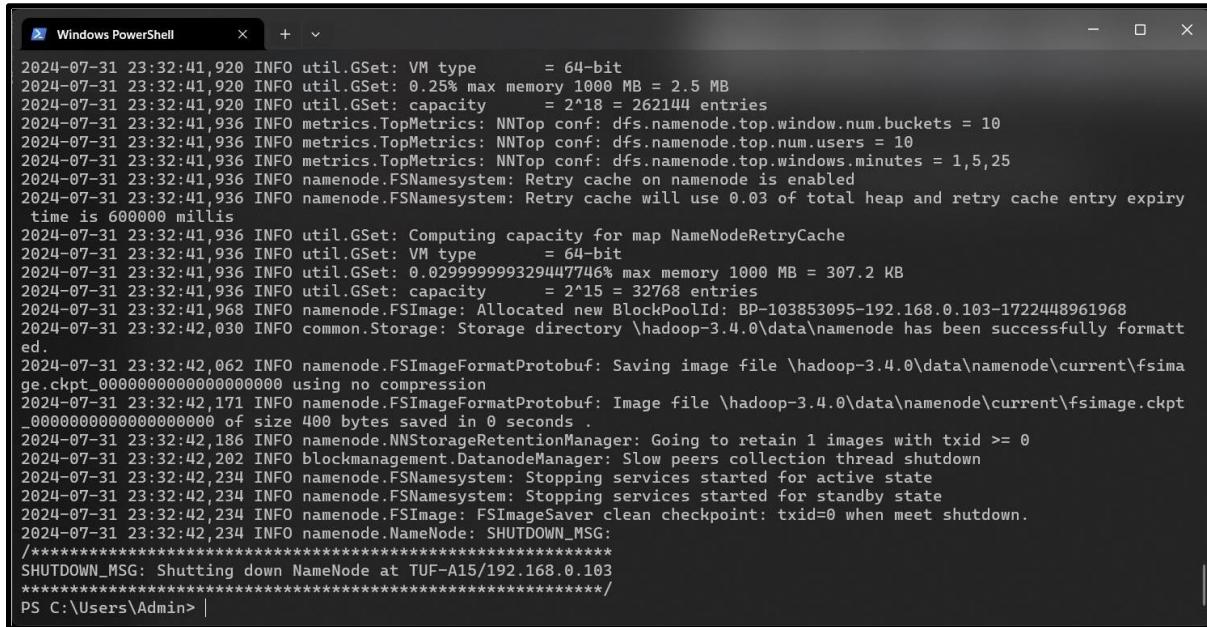
11d. Configure YARN (yarn-site.xml)



A screenshot of a code editor window titled "yarn-site.xml". The code contains XML configuration for YARN, specifically setting properties for nodemanagers. The properties include "yarn.nodemanager.aux-services" set to "mapreduce_shuffle" and "yarn.nodemanager.env-whitelist" containing "JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME". The code editor interface shows standard file operations like File, Edit, View, and tabs for configuration, property, and configuration sections.

```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.env-whitelist</name>
<value>
JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME
ME</value>
</property>
</configuration>
```

12. Initialise HDFS using the following command hdfs namenode -format



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The output shows the process of formatting the HDFS NameNode. It includes log entries from July 31, 2024, at 23:32:41,920 to 23:32:42,234. The logs detail the creation of a new BlockPoolId, the successful formatting of the storage directory, and the saving of the FSImage checkpoint. The process concludes with a shutdown message indicating the NameNode is shutting down at TUF-A15/192.168.0.103.

```
2024-07-31 23:32:41,920 INFO util.GSet: VM type      = 64-bit
2024-07-31 23:32:41,920 INFO util.GSet: 0.25% max memory 1000 MB = 2.5 MB
2024-07-31 23:32:41,920 INFO util.GSet: capacity      = 2^18 = 262144 entries
2024-07-31 23:32:41,936 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
2024-07-31 23:32:41,936 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
2024-07-31 23:32:41,936 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
2024-07-31 23:32:41,936 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
2024-07-31 23:32:41,936 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
2024-07-31 23:32:41,936 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2024-07-31 23:32:41,936 INFO util.GSet: VM type      = 64-bit
2024-07-31 23:32:41,936 INFO util.GSet: 0.029999999329447746% max memory 1000 MB = 307.2 KB
2024-07-31 23:32:41,936 INFO util.GSet: capacity      = 2^15 = 32768 entries
2024-07-31 23:32:41,968 INFO namenode.FSImage: Allocated new BlockPoolId: BP-103853095-192.168.0.103-1722448961968
2024-07-31 23:32:42,030 INFO common.Storage: Storage directory \hadoop-3.4.0\data\namenode has been successfully formatted.
2024-07-31 23:32:42,062 INFO namenode.FSImageFormatProtobuf: Saving image file \hadoop-3.4.0\data\namenode\current\fsimage.ckpt_00000000000000000000 using no compression
2024-07-31 23:32:42,171 INFO namenode.FSImageFormatProtobuf: Image file \hadoop-3.4.0\data\namenode\current\fsimage.ckpt_0000000000000000 of size 400 bytes saved in 0 seconds .
2024-07-31 23:32:42,186 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2024-07-31 23:32:42,202 INFO blockmanagement.DatanodeManager: Slow peers collection thread shutdown
2024-07-31 23:32:42,234 INFO namenode.FSNamesystem: Stopping services started for active state
2024-07-31 23:32:42,234 INFO namenode.FSNamesystem: Stopping services started for standby state
2024-07-31 23:32:42,234 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2024-07-31 23:32:42,234 INFO namenode.NameNode: SHUTDOWN_MSG:
*****SHUTDOWN_MSG: Shutting down NameNode at TUF-A15/192.168.0.103*****
PS C:\Users\Admin> |
```

13. Start HDFS Daemons by going to the `sbin` folder of Hadoop installation and running the command `start-dfs.cmd`. Processes will start running.

```
Apache Hadoop Distribution X + - □ X
8:06:53 +0000] "GET /jmx?qry=Hadoop:service=NameNode,name=ECBlockGroupsState
HTTP/1.1" 200 422 "http://localhost:9870/dfshealth.html" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0
.0.0 Safari/537.36 Edg/127.0.0.0"
2024-07-31 23:36:53,242 INFO requests.namenode: 127.0.0.1 - - [31/Jul/2024:1
8:06:53 +0000] "GET /jmx?qry=Hadoop:service=NameNode,name=FSNamesystem HTTP/
1.1" 200 2799 "http://localhost:9870/dfshealth.html" "Mozilla/5.0 (Windows N
T 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0
Safari/537.36 Edg/127.0.0.0"
2024-07-31 23:36:53,258 INFO requests.namenode: 127.0.0.1 - - [31/Jul/2024:1
8:06:53 +0000] "GET /jmx?qry=java.lang:type=Memory HTTP/1.1" 200 511 "http:
//localhost:9870/dfshealth.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) A
ppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36 Edg/127
.0.0"
2024-07-31 23:36:53,261 INFO requests.namenode: 127.0.0.1 - - [31/Jul/2024:1
8:06:53 +0000] "GET /jmx?qry=Hadoop:service=NameNode,name=BlockStats HTTP/1.
1" 200 526 "http://localhost:9870/dfshealth.html" "Mozilla/5.0 (Windows NT 1
0.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Sa
fari/537.36 Edg/127.0.0.0"
2024-07-31 23:36:53,267 INFO requests.namenode: 127.0.0.1 - - [31/Jul/2024:1
8:06:53 +0000] "GET /jmx?qry=Hadoop:service=NameNode,name=NameNodeInfo HTTP/
1.1" 200 2924 "http://localhost:9870/dfshealth.html" "Mozilla/5.0 (Windows N
T 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0
Safari/537.36 Edg/127.0.0.0"
2024-07-31 23:36:53,270 INFO requests.namenode: 127.0.0.1 - - [31/Jul/2024:1
8:06:53 +0000] "GET /favicon.ico HTTP/1.1" 404 409 "http://localhost:9870/d
fshealth.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36 Edg/127.0.0.0"
2024-07-31 23:36:53,278 INFO requests.namenode: 127.0.0.1 - - [31/Jul/2024:1
8:06:53 +0000] "GET /jmx?qry=Hadoop:service=NameNode,name=FSNamesystemState
HTTP/1.1" 200 1850 "http://localhost:9870/dfshealth.html" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0
.0.0 Safari/537.36 Edg/127.0.0.0"
2024-07-31 23:36:53,304 INFO requests.namenode: 127.0.0.1 - - [31/Jul/2024:1
8:06:53 +0000] "GET /jmx?qry=Hadoop:service=NameNode,name=FSNamesystemState
HTTP/1.1" 200 404 "http://localhost:9870/dfshealth.html" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0
.0.0 Safari/537.36 Edg/127.0.0.0"
2024-07-31 23:36:53,305 INFO requests.namenode: 127.0.0.1 - - [31/Jul/2024:1
8:06:53 +0000] "GET /conf/HTTP/1.1" 200 264778 "http://localhost:9870/dfshe
alth.html" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHT
ML, like Gecko) Chrome/127.0.0.0 Safari/537.36 Edg/127.0.0.0"
2024-07-31 23:36:53,438 INFO requests.namenode: 127.0.0.1 - - [31/Jul/2024:1
8:06:53 +0000] "GET /static/bootstrap-3.4.1/fonts/glyphicons-halflings-regul
ar.woff2 HTTP/1.1" 200 18028 "http://localhost:9870/static/bootstrap-3.4.1/c
ss/bootstrap.min.css" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/5
37.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36 Edg/127.0.0.0"

```

14. Verify HDFS web portal UI through this [link](#).

Overview '0.0.0.0:19000' (✓active)	
<hr/>	
Started:	Wed Jul 31 23:36:23 +0530 2024
Version:	3.4.0, rbd8b77f398f626bb7791783192ee7a5dfaeecc760
Compiled:	Mon Mar 04 12:05:00 +0530 2024 by root from (HEAD detached at release-3.4.0-RC3)
Cluster ID:	CID-660c4b28-518d-4345-b3f5-04010463f330
Block Pool ID:	BP-103853095-192.168.0.103-1722448961968
<hr/>	
Summary	
<hr/>	
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 61.4 MB of 85 MB Heap Memory. Max Heap Memory is 1000 MB.	
Non Heap Memory used 54.22 MB of 56.19 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.	
<hr/>	
Configured Capacity:	552.67 GB
Configured Remote Capacity:	0 B
DFS Used:	149 B (0%)
Non DFS Used:	245.09 GB

15. Start YARN Daemons by using the following command `start-yarn.cmd`.

16. Verify YARN resource manager through this [link](#).

The screenshot shows the Hadoop 'All Applications' interface. On the left, there's a sidebar with navigation links: Cluster (selected), About Nodes, Node Labels, Applications, Scheduler, and Tools. The main area displays 'Cluster Metrics' with counts for Apps Submitted (0), Apps Pending (0), Apps Running (0), Apps Completed (0), Containers Running (0), Used Resources (<memory:0 B, vCores:0>), and a warning for memory usage. Below it is 'Cluster Nodes Metrics' showing Active Nodes (0), Decommissioning Nodes (0), Decommissioned Nodes (0), and Lost Nodes (0). Under 'Scheduler Metrics', it lists 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 Cluster Application F. A table below shows application details with columns: ID, User, Name, Application Type, Application Tags, Queue, Application Priority, StartTime, LaunchTime, FinishTime, State, FinalStatus, Running Containers, and Allocatable CPU VCore. The table currently shows 'No data available in table'. At the bottom, it says 'Showing 0 to 0 of 0 entries'.

Practical 2: Installation of Scala and Apache Spark

1. Go to this [link](#) and download Scala msi installer.

The screenshot shows the 'Other resources' section of the Scala 2.13.14 release page. It lists download links for different file formats and systems:

Archive	System	Size
scala-2.13.14.tgz	Mac OS X, Unix, Cygwin	23.31M
scala-2.13.14.msi	Windows (msi installer)	137.83M
scala-2.13.14.zip	Windows	23.35M
scala-2.13.14.deb	Debian	663.46M
scala-2.13.14.rpm	RPM package	138.07M
scala-docs-2.13.14.txz	API docs	61.61M
scala-docs-2.13.14.zip	API docs	117.91M
scala-sources-2.13.14.tar.gz	Sources	8.0M

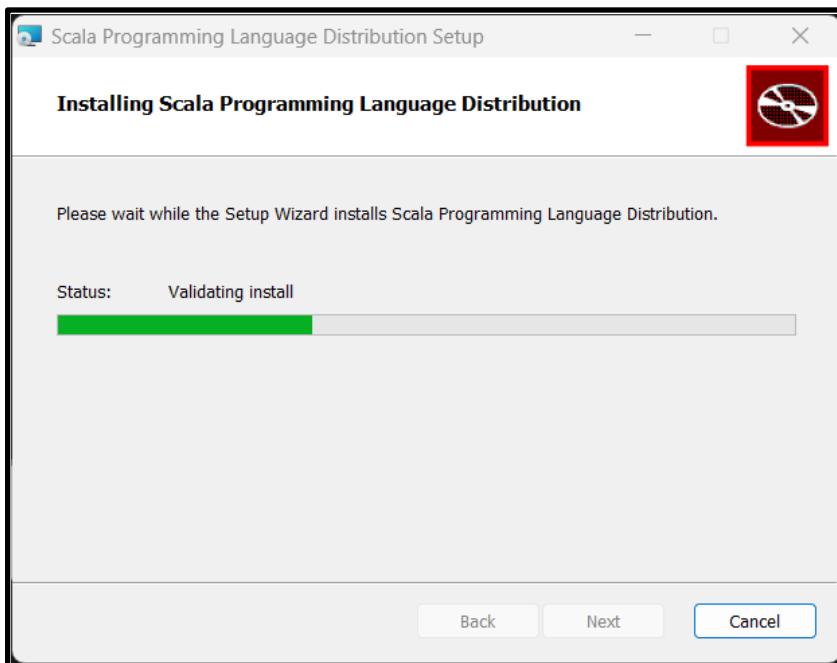
Other Releases

You can find the links to prior versions or the latest development version below. To see a detailed list of changes for each version of Scala please refer to the [changelog](#).

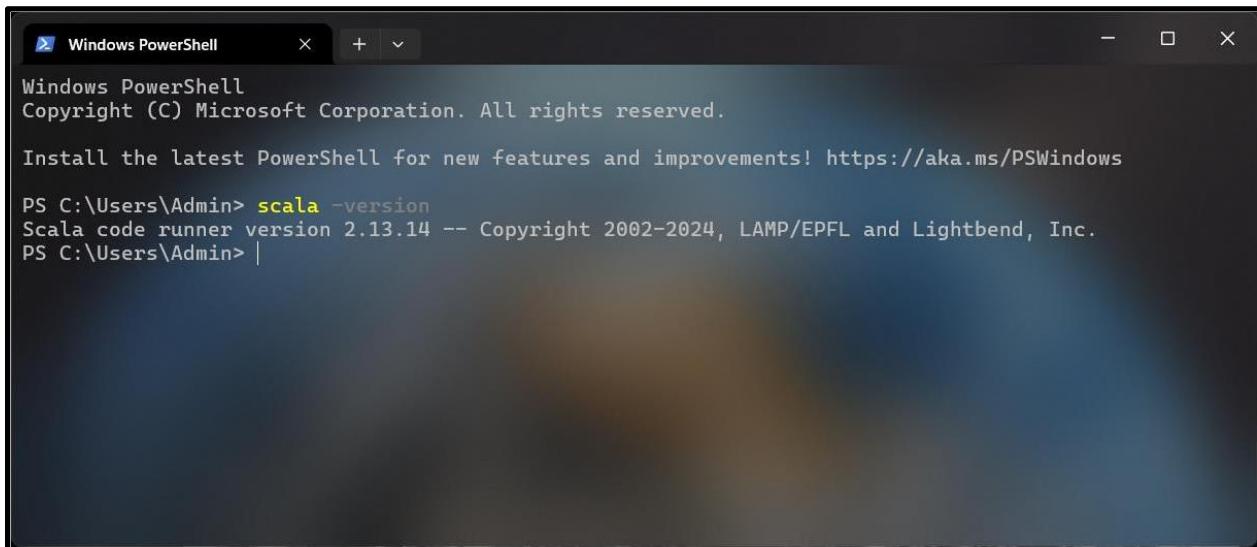
Note that different major releases of Scala 2 (e.g. Scala 2.11.x and Scala 2.12.x) are not [binary compatible](#) with each other. Scala 3 minor releases (e.g. 3.0.x and 3.1.x) follow a [different compatibility model](#).

- Current 3.5.x release:

2. Run the installer.



3. Using the command `scala -version` check if scala is installed.

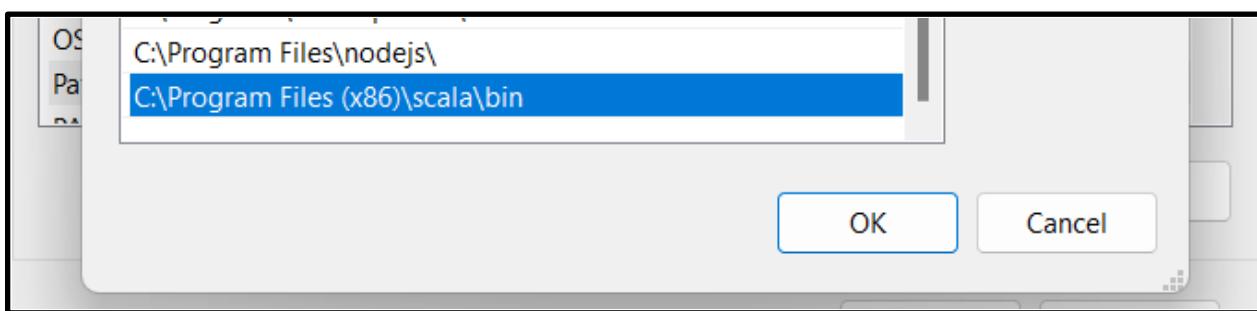
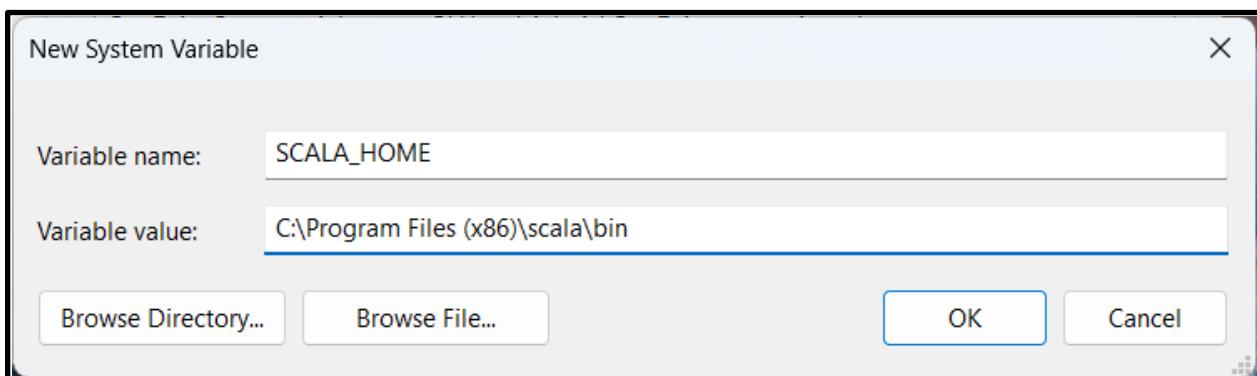


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

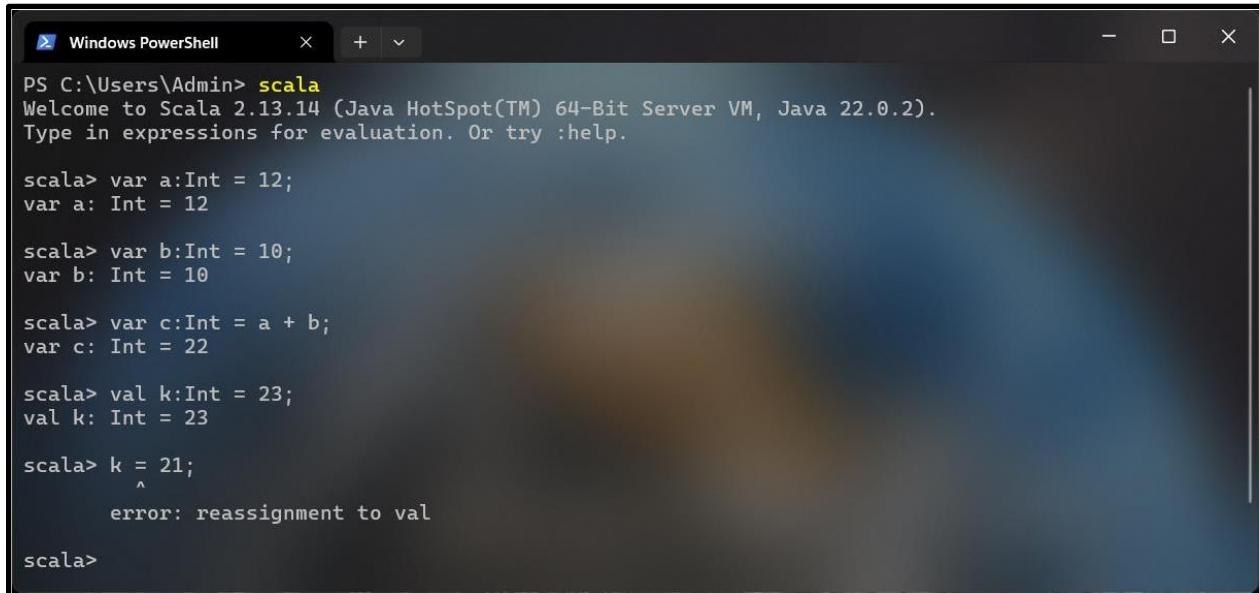
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Admin> scala -version
Scala code runner version 2.13.14 -- Copyright 2002-2024, LAMP/EPFL and Lightbend, Inc.
PS C:\Users\Admin>
```

4. Configure the System Environment Variable.



5. For mutable variables we use keyword ‘var’, whereas for immutable variables we use the keyword ‘val’. Error will be thrown if values are reassigned to an immutable variable.



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The command PS C:\Users\Admin> scala is run, followed by several Scala expressions. The output shows the evaluation of variables a, b, and c, and then an attempt to reassign the immutable variable k, which results in an error message: "error: reassignment to val".

```
PS C:\Users\Admin> scala
Welcome to Scala 2.13.14 (Java HotSpot(TM) 64-Bit Server VM, Java 22.0.2).
Type in expressions for evaluation. Or try :help.

scala> var a:Int = 12;
var a: Int = 12

scala> var b:Int = 10;
var b: Int = 10

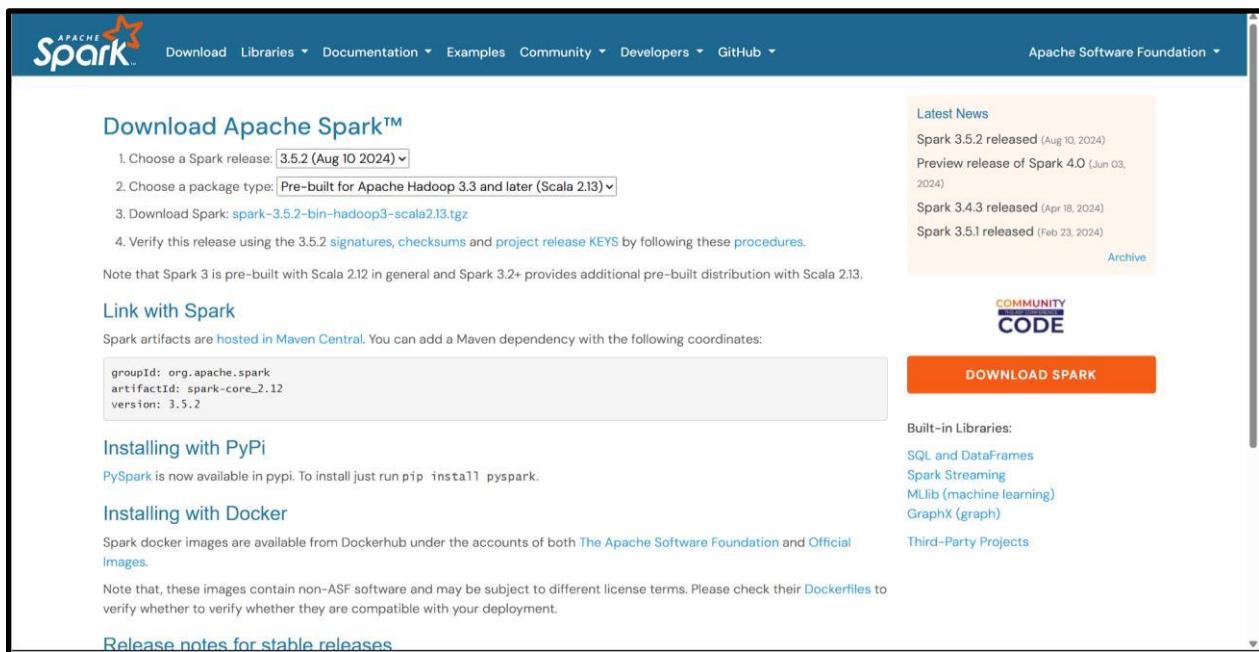
scala> var c:Int = a + b;
var c: Int = 22

scala> val k:Int = 23;
val k: Int = 23

scala> k = 21;
           ^
      error: reassignment to val

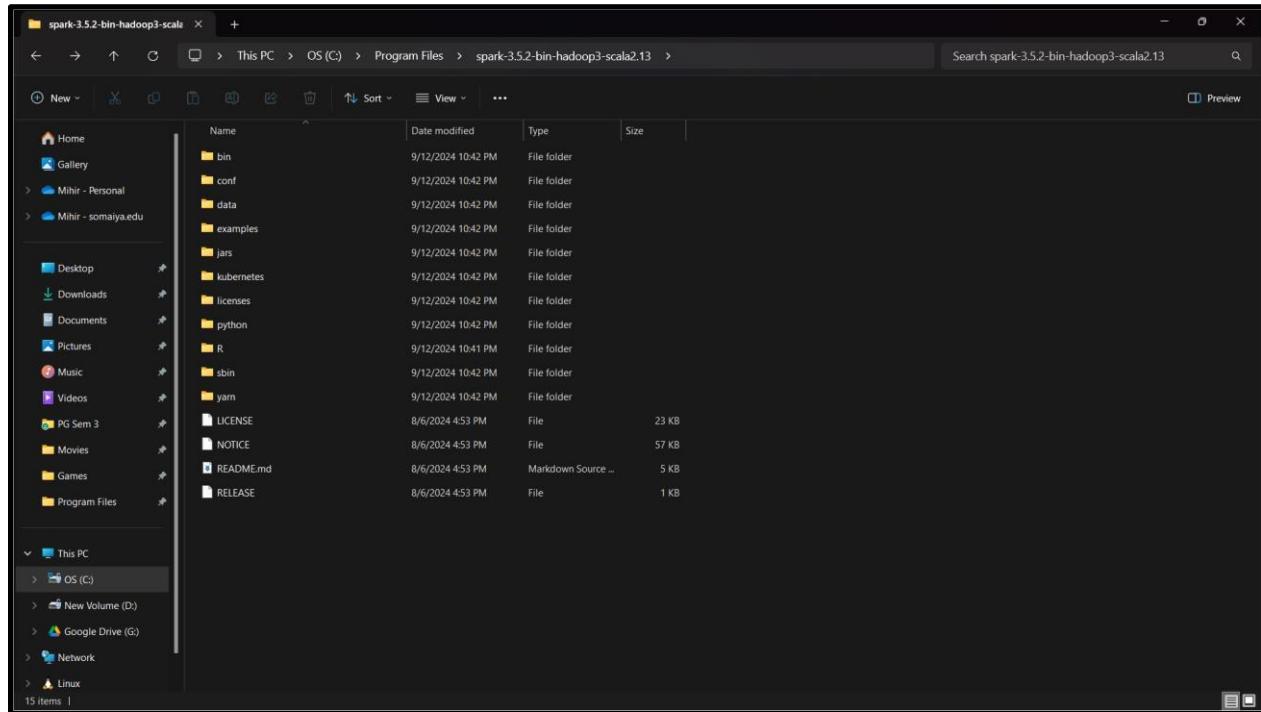
scala>
```

6. Download Apache Spark from [here](#).

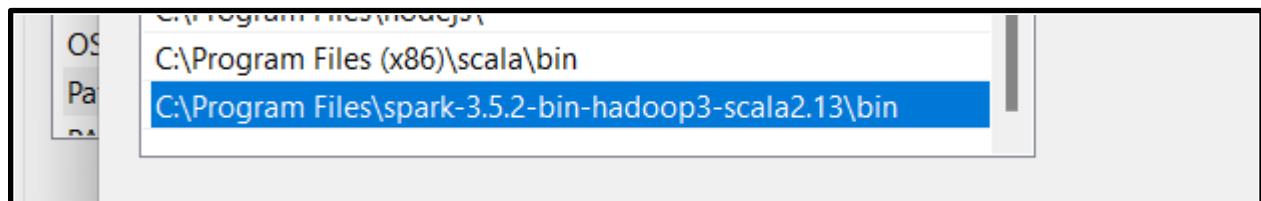
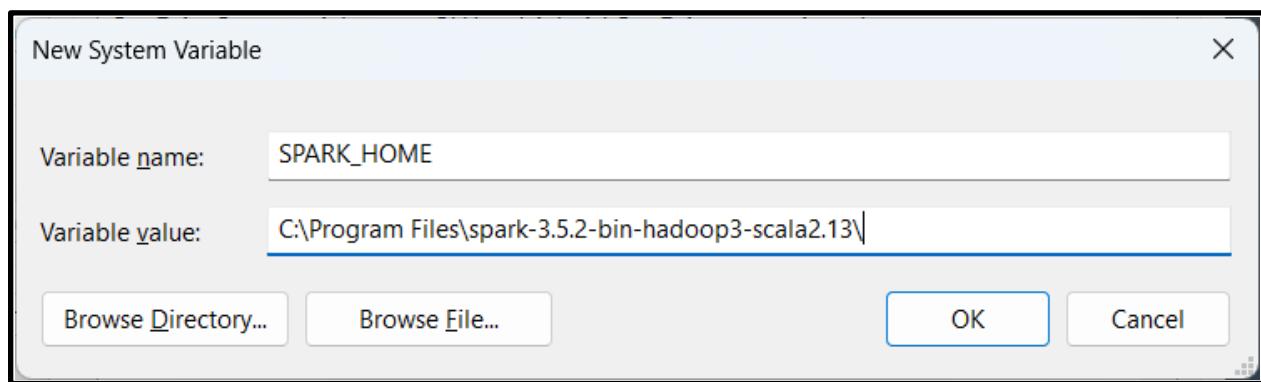


The screenshot shows the Apache Spark download page. At the top, there's a navigation bar with links for Download, Libraries, Documentation, Examples, Community, Developers, GitHub, and Apache Software Foundation. The main content area has a heading "Download Apache Spark™" and a numbered list of steps to choose a release, package type, and download file. Below this is a note about pre-built distributions. To the right, there's a "Latest News" sidebar with links to recent releases and an "Archive" link. Further down, there's a "Link with Spark" section for Maven Central dependencies, a "DOWNLOAD SPARK" button, and sections for "Installing with PyPi", "Installing with Docker", and "Release notes for stable releases". On the far right, there's a "COMMUNITY CODE" section and a "Built-in Libraries" list including SQL and DataFrames, Spark Streaming, MLlib, and GraphX.

7. Extract the zip and paste it in Program Files.



8. Configure the user and system environment variables.



9. Run spark-shell in the bin folder of spark.

10. Run the following commands

```
val x = spark.read.json("C:\\Program Files\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.json");
```

x.show()

```
Windows PowerShell
/_/
Using Scala version 2.13.8 (Java HotSpot(TM) 64-Bit Server VM, Java 11.0.23)
Type in expressions to have them evaluated.
Type :help for more information.
Spark context Web UI available at http://TUF-A15:4040
Spark context available as 'sc' (master = local[*], app id = local-1726163575780).
Spark session available as 'spark'.

scala> val x = spark.read.json("C:\\Program Files\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.json");
val x: org.apache.spark.sql.DataFrame = [age: bigint, name: string]

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

scala> |
```

```
x.printSchema()
```

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

```
x.select($"name",$"age").show()
```

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

```
x.filter($"age">>20).show()
```

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

```
x.select($"age"+1).show()
```

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

```
x.createOrReplaceTempView("people")
```

```
val sqlDF = spark.sql("Select * from people")
```

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

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

```
sqlDF.show()
```

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

```
df.createGlobalTempView("people")
```

```
scala> df.createGlobalTempView("people")
24/09/19 10:40:27 WARN HiveConf: HiveConf of name hive.stats.jdbc.timeout does not exist
24/09/19 10:40:27 WARN HiveConf: HiveConf of name hive.stats.retries.wait does not exist
24/09/19 10:40:29 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 10:40:29 WARN ObjectStore: setMetaStoreSchemaVersion called but recording version is disabled: version = 2.3.0, comment = Set by MetaStore UNKNOWN@172.23.1.71
24/09/19 10:40:29 WARN ObjectStore: Failed to get database global_temp, returning NoSuchObjectException
```

```
spark.sql("SELECT * FROM global_temp.people").show()
```

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

```
spark.newSession().sql("SELECT * FROM global_temp.people").show()
```

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

```
case class Person(name: String, age: Long)
```

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

```
val caseClassDS = Seq(Person("Andy", 32)).toDS()
```

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

```
caseClassDS.show()
```

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

```
val primitiveDS = Seq(1, 2, 3).toDS()
```

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

```
primitiveDS.map(_ + 1).collect()
```

```
scala> primitiveDS.map(_ + 1).collect()
val res8: Array[Int] = Array(2, 3, 4)
```

```
val path = "C:\\Program Files\\spark-3.5.2-bin-hadoop3-
scala2.13\\examples\\src\\main\\resources\\people.json"
```

```
scala> val path = "C:\\Program Files\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.json"
val path: String = C:\\Program Files\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.json
```

```
val peopleDS = spark.read.json(path).as[Person]
```

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

```
peopleDS.show()
```

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

```
import spark.implicits._
```

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

```
val peopleDF = spark.sparkContext.textFile("C:\\Program Files\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.txt").map(_.split(",")).map(attributes => Person(attributes(0), attributes(1).trim.toInt)).toDF()
```

```
scala> val peopleDF = spark.sparkContext.textFile("C:\\Program Files\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.txt").map(_.split(",")).map(attributes => Person(attributes(0), attributes(1).trim.toInt)).toDF()
val peopleDF: org.apache.spark.sql.DataFrame = [name: string, age: bigint]
```

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

```
val teenagersDF = spark.sql("SELECT name, age FROM people WHERE age BETWEEN 13 AND 19")
```

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

```
teenagersDF.map(teenager => "Name: " + teenager(0)).show()
```

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

```
teenagersDF.map(teenager => "Name: " + teenager.getAs[String]("name")).show()
```

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

```
implicit val mapEncoder = org.apache.spark.sql.Encoders.kryo[Map[String, Any]]
```

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

```
teenagersDF.map(teenager => teenager.getValuesMap[Any](List("name",
"age"))).collect()
```

```
scala> teenagersDF.map(teenager => teenager.getValuesMap[Any](List("name", "age"))).collect()
val res29: Array[Map[String, Any]] = Array(Map(name -> Justin, age -> 19))
```

11. Perform the same operations on people.csv.

```
val a = spark.read.option("header", "true").csv("C:\\Program Files\\spark-3.5.2-bin-
hadoop3-scala2.13\\examples\\src\\main\\resources\\people.csv");
```

a.show()

```
scala> val a = spark.read.option("header", "true").csv("C:\\Program Files\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\people.csv");
val a: org.apache.spark.sql.DataFrame = [name:age:job: string]

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

a.printSchema()

```
scala> a.printSchema()
root
|-- name:age;job: string (nullable = true)
```

12. Performing operations on custom data.

```
val mydata = spark.read.format("csv").option("inferschema",
"true").option("header", "true").load("C:\\Program Files\\spark-3.5.2-bin-hadoop3-
scala2.13\\examples\\src\\main\\resources\\banking.csv")
```

```
scala> val mydata = spark.read.format("csv").option("inferschema", "true").option("header", "true").load("C:\\Program Fi
les\\spark-3.5.2-bin-hadoop3-scala2.13\\examples\\src\\main\\resources\\banking.csv")
val mydata: org.apache.spark.sql.DataFrame = [age: int, job: string ... 19 more fields]
```

mydata.show()

Name: Vatsal Parekh

Roll No: 31031523022

```
scala> mydata.show()
+---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|age| job | marital| education|default|housing|loan| contact|month|day_of_week|duration|campaign|pdays|previous| poutcome|emp_var_rate|cons_price_idx|cons_conf_idx|euribor3m|nr_employed| y|
+---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 44|blue-collar| married| basic.4y| unknown| yes| no| cellular| aug| thu| 210| 1| 999| 0| nonexistent| 1.4| 93.44uu| -36.1| 4.963| 5228.1| 0|
| 53|technician| married| unknown| no| no| no| cellular| nov| fri| 138| 1| 999| 0| nonexistent| -0.1| 93.2| -42.0| 4.621| 5195.8| 0|
| 28|management| single| university.degree| no| yes| no| cellular| jun| thu| 339| 3| 6| 2| success| -1.7| 94.055| -39.8| 0.729| 4991.6| 1|
| 39| services| married| high.school| no| no| no| cellular| apr| fri| 185| 2| 999| 0| nonexistent| -1.8| 93.075| -47.1| 1.4085| 5099.1| 0|
| 55| retired| married| basic.4y| no| yes| no| cellular| aug| fri| 177| 1| 3| 1| success| -2.9| 92.281| -31.4| 0.869| 5076.2| 1|
| 30|management| divorced| basic.4y| no| yes| no| cellular| jul| tue| 68| 8| 999| 0| nonexistent| 1.4| 93.918| -42.7| 4.961| 5228.1| 0|
| 37|blue-collar| married| basic.4y| no| yes| no| cellular| may| thu| 204| 1| 999| 0| nonexistent| -1.8| 92.893| -46.2| 1.327| 5099.1| 0|
| 39|blue-collar| divorced| basic.9y| no| yes| no| cellular| may| fri| 191| 1| 999| 0| nonexistent| -1.8| 92.893| -46.2| 1.313| 5099.1| 0|
| 36| admin.| married| university.degree| no| no| no| cellular| jun| mon| 174| 1| 3| 1| success| -2.9| 92.963| -48.8| 1.266| 5076.2| 1|
| 27|blue-collar| single| basic.4y| no| yes| no| cellular| apr| thu| 191| 2| 999| 1| fail| -1.8| 93.096| -47.1| 1.311| 5099.1| 0|
| 30| housemaid| single| university.degree| no| no| no| telephone| apr| fri| 62| 2| 999| 0| nonexistent| 1.1| 93.99u| -36.1| 4.860| 5191.0| 0|
| 41|management| married| university.degree| no| yes| no| cellular| aug| thu| 789| 1| 999| 0| nonexistent| 1.4| 93.44uu| -36.1| 4.960| 5228.1| 0|
| 55| management| married| university.degree| no| no| no| cellular| aug| mon| 372| 3| 999| 0| nonexistent| 1.4| 93.44uu| -36.1| 4.965| 5228.1| 1|
| 33| services| divorced| high.school| no| yes| no| cellular| may| tue| 75| 5| 999| 0| nonexistent| -1.8| 92.893| -46.2| 1.291| 5099.1| 0|
| 26| admin.| married| high.school| no| no| yes| telephone| jun| mon| 1021| 1| 999| 0| nonexistent| 1.4| 94.465| -41.8| 4.96| 5228.1| 0|
| 52| services| married| high.school| unknown| yes| no| cellular| jul| thu| 117| 2| 999| 0| nonexistent| 1.4| 93.918| -42.7| 4.962| 5228.1| 0|
| 35| services| married| high.school| unknown| yes| no| cellular| may| thu| 1034| 2| 999| 0| nonexistent| -1.8| 93.075| -47.1| 1.365| 5099.1| 1|
| 27| admin.| single| university.degree| no| no| no| telephone| oct| tue| 548| 1| 999| 0| nonexistent| 1.1| 93.798| -48.4| 4.86| 5195.8| 1|
| 28|blue-collar| married| basic.4y| unknown| yes| no| cellular| may| thu| 108| 1| 999| 0| nonexistent| 1.1| 93.99u| -36.4| 4.86| 5191.0| 0|
| 26|blue-collar| married| basic.9y| no| yes| no| cellular| jul| mon| 108| 1| 999| 0| nonexistent| 1.1| 93.99u| -36.4| 4.86| 5191.0| 0|
| 26|unemployed| single| basic.9y| no| yes| yes| cellular| jul| mon| 108| 4| 999| 0| nonexistent| 1.4| 93.918| -42.7| 4.96| 5228.1| 0|
only showing top 20 rows
```

mydata.show(50)

```
scala> mydata.show(50)
+---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|age| job | marital| education|default|housing|loan| contact|month|day_of_week|duration|campaign|pdays|previous| poutcome|emp_var_rate|cons_price_idx|cons_conf_idx|euribor3m|nr_employed| y|
+---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 44|blue-collar| married| basic.4y| unknown| yes| no| cellular| aug| thu| 210| 1| 999| 0| nonexistent| 1.4| 93.44uu| -36.1| 4.963| 5228.1| 0| |
| 53|technician| married| unknown| no| no| no| cellular| nov| fri| 138| 1| 999| 0| nonexistent| -0.1| 93.2| -42.0| 4.621| 5195.8| 0|
| 28|management| single| university.degree| no| yes| no| cellular| jun| thu| 339| 3| 6| 2| success| -1.7| 94.055| -39.8| 0.729| 4991.6| 1|
| 39| services| married| high.school| no| no| no| cellular| apr| fri| 185| 2| 999| 0| nonexistent| -1.8| 93.075| -47.1| 1.4085| 5099.1| 0|
| 55| retired| married| basic.4y| no| yes| no| cellular| aug| fri| 177| 1| 3| 1| success| -2.9| 92.281| -31.4| 0.869| 5076.2| 1|
| 30|management| divorced| basic.4y| no| yes| no| cellular| jul| tue| 68| 8| 999| 0| nonexistent| 1.4| 93.918| -42.7| 4.961| 5228.1| 0|
| 37|blue-collar| married| basic.4y| no| yes| no| cellular| may| thu| 204| 1| 999| 0| nonexistent| -1.8| 92.893| -46.2| 1.327| 5099.1| 0|
| 39|blue-collar| divorced| basic.9y| no| yes| no| cellular| may| fri| 191| 1| 999| 0| nonexistent| -1.8| 92.893| -46.2| 1.313| 5099.1| 0|
| 36| admin.| married| university.degree| no| no| no| cellular| jun| mon| 174| 1| 3| 1| success| -2.9| 92.963| -48.8| 1.266| 5076.2| 1|
| 27|blue-collar| single| basic.4y| no| yes| no| cellular| apr| thu| 191| 2| 999| 1| fail| -1.8| 93.096| -47.1| 1.311| 5099.1| 0|
| 30| housemaid| single| university.degree| no| no| no| telephone| apr| fri| 62| 2| 999| 0| nonexistent| 1.1| 93.99u| -36.1| 4.860| 5191.0| 0|
| 41|management| married| university.degree| no| yes| no| cellular| aug| thu| 789| 1| 999| 0| nonexistent| 1.4| 93.44uu| -36.1| 4.960| 5228.1| 0|
| 55| management| married| university.degree| no| no| no| cellular| aug| mon| 372| 3| 999| 0| nonexistent| 1.4| 93.44uu| -36.1| 4.965| 5228.1| 1|
| 33| services| divorced| high.school| no| yes| no| cellular| may| tue| 75| 5| 999| 0| nonexistent| -1.8| 92.893| -46.2| 1.291| 5099.1| 0|
| 26| admin.| married| high.school| no| no| yes| telephone| jun| mon| 1021| 1| 999| 0| nonexistent| 1.4| 94.465| -41.8| 4.96| 5228.1| 0|
| 52| services| married| high.school| unknown| yes| no| cellular| jul| thu| 117| 2| 999| 0| nonexistent| 1.4| 93.918| -42.7| 4.962| 5228.1| 0|
| 35| services| married| high.school| unknown| yes| no| cellular| may| thu| 1034| 2| 999| 0| nonexistent| -1.8| 93.075| -47.1| 1.365| 5099.1| 1|
| 27| admin.| single| university.degree| no| no| no| telephone| oct| tue| 548| 1| 999| 0| nonexistent| 1.1| 93.798| -48.4| 4.86| 5195.8| 1|
| 28|blue-collar| married| basic.4y| unknown| yes| no| cellular| may| thu| 108| 1| 999| 0| nonexistent| 1.1| 93.99u| -36.4| 4.86| 5191.0| 0|
| 26|blue-collar| married| basic.9y| no| yes| no| cellular| jul| mon| 108| 1| 999| 0| nonexistent| 1.1| 93.99u| -36.4| 4.86| 5191.0| 0|
| 26|unemployed| single| basic.9y| no| yes| yes| cellular| jul| mon| 108| 4| 999| 0| nonexistent| 1.4| 93.918| -42.7| 4.96| 5228.1| 0|
| 24| technician| single| professional.course| no| no| no| cellular| may| thu| 204| 1| 999| 0| nonexistent| 1.4| 93.918| -42.7| 4.962| 5228.1| 0|
| 41| blue-collar| married| high.school| unknown| yes| no| cellular| may| thu| 241| 3| 999| 0| nonexistent| 1.4| 93.918| -42.7| 4.962| 5228.1| 0|
| 34|entrepreneur| single| university.degree| no| yes| no| cellular| may| tue| 168| 2| 999| 0| nonexistent| -1.8| 92.893| -46.2| 1.34u| 5099.1| 0|
| 49| technician| divorced| unknown| no| yes| yes| cellular| oct| thu| 81| 2| 999| 0| nonexistent| -3.4| 90.431| -28.9| 0.781| 5017.0| 0|
| 37| services| married| high.school| no| no| no| cellular| may| thu| 204| 1| 999| 0| nonexistent| 1.1| 93.795| -47.1| 1.365| 5099.1| 0|
| 35| blue-collar| married| basic.4y| unknown| yes| no| cellular| may| fri| 746| 2| 999| 0| nonexistent| 1.8| 92.893| -46.2| 1.313| 5099.1| 0|
| 38|blue-collar| single| basic.4y| unknown| yes| no| telephone| jul| tue| 41| 5| 999| 0| nonexistent| 1.4| 93.918| -42.7| 4.961| 5228.1| 0|
| 41| services| divorced| high.school| no| no| no| telephone| jun| tue| 115| 1| 999| 0| nonexistent| 1.4| 94.465| -41.8| 4.961| 5228.1| 0|
| 46| services| married| university.degree| no| no| no| cellular| may| thu| 227| 2| 999| 0| nonexistent| -1.8| 92.893| -46.2| 1.327| 5099.1| 0|
| 39| technician| married| professional.course| no| yes| no| cellular| may| thu| 275| 1| 999| 0| nonexistent| 1.8| 92.893| -46.2| 1.327| 5099.1| 0|
| 29| technician| single| professional.course| no| no| no| cellular| may| thu| 369| 1| 999| 0| nonexistent| 1.4| 93.918| -42.7| 4.963| 5099.1| 0|
| 32| services| divorced| basic.4y| unknown| no| no| no| cellular| may| thu| 35| 1| 999| 0| nonexistent| 1.4| 93.918| -42.7| 4.962| 5228.1| 0|
| 41| blue-collar| married| high.school| unknown| yes| no| cellular| may| thu| 241| 3| 999| 0| nonexistent| 1.4| 93.918| -42.7| 4.962| 5228.1| 0|
| 34| entrepreneur| single| university.degree| no| yes| no| cellular| may| tue| 168| 2| 999| 0| nonexistent| -1.8| 92.893| -46.2| 1.34u| 5099.1| 0|
| 49| technician| divorced| unknown| no| yes| yes| cellular| oct| thu| 81| 2| 999| 0| nonexistent| -3.4| 90.431| -28.9| 0.781| 5017.0| 0|
| 37| services| married| high.school| no| no| no| cellular| may| thu| 204| 1| 999| 0| nonexistent| 1.1| 93.795| -47.1| 1.365| 5099.1| 0|
| 35| blue-collar| married| basic.4y| unknown| yes| no| cellular| may| fri| 746| 2| 999| 0| nonexistent| 1.8| 92.893| -46.2| 1.313| 5099.1| 0|
| 38|blue-collar| single| basic.4y| unknown| yes| no| telephone| jul| tue| 41| 5| 999| 0| nonexistent| 1.4| 93.918| -42.7| 4.961| 5228.1| 0|
| 41| services| divorced| high.school| no| no| no| telephone| jun| tue| 115| 1| 999| 0| nonexistent| 1.4| 94.465| -41.8| 4.961| 5228.1| 0|
| 46| services| married| university.degree| no| no| no| cellular| may| thu| 227| 2| 999| 0| nonexistent| -1.8| 92.893| -46.2| 1.327| 5099.1| 0|
| 39| technician| married| professional.course| no| yes| no| cellular| may| thu| 275| 1| 999| 0| nonexistent| 1.8| 92.893| -46.2| 1.327| 5099.1| 0|
| 29| technician| single| professional.course| no| no| no| cellular| may| thu| 369| 1| 999| 0| nonexistent| 1.4| 93.918| -42.7| 4.963| 5099.1| 0|
| 32| services| divorced| basic.4y| unknown| no| no| no| cellular| may| thu| 35| 1| 999| 0| nonexistent| 1.4| 93.918| -42.7| 4.962| 5228.1| 0|
| 48| blue-collar| married| basic.9y| unknown| no| no| cellular| aug| tue| 213| 3| 999| 0| nonexistent| 1.4| 93.44uu| -36.1| 4.966| 5228.1| 0|
| 36| management| married| high.school| unknown| yes| no| cellular| nov| thu| 371| 2| 999| 0| nonexistent| -0.1| 93.2| -42.0| 4.621| 5195.8| 0|
| 31| housemaid| married| basic.4y| no| yes| no| cellular| mon| 87| 1| 999| 0| nonexistent| 1.0| 93.918| -42.1| 4.965| 5228.1| 0|
| 43|entrepreneur| married| basic.4y| no| yes| no| cellular| may| mon| 195| 3| 999| 1| failure| -1.8| 92.893| -46.2| 1.35u| 5099.1| 0|
| 56| retired| married| basic.4y| unknown| yes| no| cellular| aug| wed| 162| 1| 999| 0| nonexistent| 1.4| 93.44uu| -36.1| 4.967| 5228.1| 0|
| 42| blue-collar| married| basic.9y| unknown| yes| no| telephone| jun| fri| 93| 2| 999| 0| nonexistent| 1.4| 94.465| -41.8| 4.959| 5228.1| 0|
only showing top 50 rows
```

mydata.select("age", "\$y").show()

```
scala> mydata.select($"age", $"y").show()
+---+---+
|age| y|
+---+---+
| 44| 0|
| 53| 0|
| 28| 1|
| 39| 0|
| 55| 1|
| 30| 0|
| 37| 0|
| 39| 0|
| 36| 1|
| 27| 0|
| 34| 0|
| 41| 0|
| 55| 1|
| 33| 0|
| 26| 0|
| 52| 0|
| 35| 1|
| 27| 1|
| 28| 0|
| 26| 0|
+---+---+
only showing top 20 rows
```

mydata.count()

```
scala> mydata.count()
val res39: Long = 41188
```

mydata.count.toDouble

```
scala> mydata.count.toDouble
warning: 1 deprecation (since 2.13.3); for details, enable `:setting -deprecation` or `:replay -deprecation`
val res40: Double = 41188.0
```

Practical 3: Spark GraphX

```
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._
```

```
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))
```

```
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 <consol  
e:>:1
```

```
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))
```

```
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))
```

```
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
```

```
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))
```

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

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

```
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
```

```
#To check number of Airports
```

```
val numairports = graph.numVertices
```

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

```
#To check routes
```

```
val numairports = graph.numEdges
```

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

```
#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)
```

```
#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)
```

```
#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))
```

#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))
```

#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: Working with PySpark

```
!pip install pyspark
```

```
import pyspark
```

```
import pandas as pd
```

```
pd.read_csv('data.csv')
```

	Name	Age	Experience	Salary
0	Willetta	27.0	52.0	62860.0
1	Merrie	45.0	34.0	57591.0
2	Joleen	43.0	42.0	61343.0
3	Nananne	25.0	33.0	75478.0
4	Jennica	56.0	10.0	71299.0
5	Lizzie	55.0	27.0	62050.0
6	Kaja	28.0	18.0	52610.0
7	Aubrie	55.0	18.0	77702.0
8	Ginnie	71.0	26.0	88700.0
9	Correy	47.0	25.0	83288.0
10	Rochette	24.0	12.0	56123.0
11	Hayley	55.0	51.0	72758.0
12	Mathilda	42.0	44.0	62825.0
13	Veda	61.0	31.0	68511.0
14	Jessy	59.0	41.0	85482.0
15	John	30.0	5.0	50000.0
16	Mike	25.0	NaN	NaN
17	NaN	40.0	NaN	NaN
18	Sarah	NaN	3.0	40000.0
19	Sarah	NaN	NaN	NaN

```
from pyspark.sql import SparkSession
```

```
spark = SparkSession.builder.appName('Practise').getOrCreate()
```

```
# inferSchema determines the datatype of each column.
```

```
df = spark.read.csv('data.csv', header=True, inferSchema=True)
```

```
df.show()
```

Name	Age	Experience	Salary
Willetta	27	52	62860
Merrie	45	34	57591
Joleen	43	42	61343
Nananne	25	33	75478
Jennica	56	10	71299
Lizzie	55	27	62050
Kaja	28	18	52610
Aubrie	55	18	77702
Ginnie	71	26	88700
Correy	47	25	83288
Rochette	24	12	56123
Hayley	55	51	72758
Mathilda	42	44	62825
Veda	61	31	68511
Jessy	59	41	85482
John	30	5	50000
Mike	25	NULL	NULL
NULL	40	NULL	NULL
Sarah	NULL	3	40000
Sarah	NULL	NULL	NULL

```
# Type of the data
```

```
print(type(df))
```

```
<class 'pyspark.sql.dataframe.DataFrame'>
```

```
# Check Schema
```

```
df.printSchema()
```

```
root
 |-- Name: string (nullable = true)
 |-- Age: integer (nullable = true)
 |-- Experience: integer (nullable = true)
 |-- Salary: integer (nullable = true)
```

```
# Get Column Names
```

```
df.columns
```

```
['Name', 'Age', 'Experience', 'Salary']
```

```
# First 3 Data Values
```

```
df.head(3)
```

```
[Row(Name='Willetta', Age=27, Experience=52, Salary=62860),  
 Row(Name='Merrie', Age=45, Experience=34, Salary=57591),  
 Row(Name='Joleen', Age=43, Experience=42, Salary=61343)]
```

```
# Selecting Specific Columns
```

```
df.select(['Name', 'Age']).show()
```

```
+-----+-----+  
|      Name|  Age|  
+-----+-----+  
| Willetta|  27|  
| Merrie|  45|  
| Joleen|  43|  
| Nananne|  25|  
| Jennica|  56|  
| Lizzie|  55|  
| Kaja|  28|  
| Aubrie|  55|  
| Ginnie|  71|  
| Correy|  47|  
| Rochette| 24|  
| Hayley|  55|  
| Mathilda| 42|  
| Veda|  61|  
| Jessy|  59|  
| John|  30|  
| Mike|  25|  
| NULL|  40|  
| Sarah|NULL|  
| Sarah|NULL|  
+-----+-----+
```

```
# Check Datatypes
```

```
df.dtypes
```

```
[('Name', 'string'), ('Age', 'int'), ('Experience', 'int'), ('Salary', 'int')]
```

```
# Describe dataset
```

```
df.describe().show()
```

```
+-----+-----+-----+-----+-----+  
| summary|      Name|        Age|   Experience|       Salary|  
+-----+-----+-----+-----+-----+  
|  count|      19|      18|        17|       17|  
|  mean|    NULL|43.77777777777778|27.764705882352942|66389.41176470589|  
| stddev|    NULL|14.63901135988258| 15.29513571272214|13310.58934673266|  
|   min|    Aubrie|      24|          3|     40000|  
|   max|    Willetta|      71|         52|    88700|  
+-----+-----+-----+-----+-----+
```

```
# Adding columns in data frame

df = df.withColumn('Experience After 2 Years', df['Experience'] + 2)

df.show()
```

Name	Age	Experience	Salary	Experience After 2 Years
Willetta	27	52	62860	54
Merrie	45	34	57591	36
Joleen	43	42	61343	44
Nananne	25	33	75478	35
Jennica	56	10	71299	12
Lizzie	55	27	62050	29
Kaja	28	18	52610	20
Aubrie	55	18	77702	20
Ginnie	71	26	88700	28
Correy	47	25	83288	27
Rochette	24	12	56123	14
Hayley	55	51	72758	53
Mathilda	42	44	62825	46
Veda	61	31	68511	33
Jessy	59	41	85482	43
John	30	5	50000	7
Mike	25	NULL	NULL	NULL
NULL	40	NULL	NULL	NULL
Sarah	NULL	3	40000	5
Sarah	NULL	NULL	NULL	NULL

```
# Drop the column

df = df.drop('Experience After 2 Years')

df.show()
```

Name	Age	Experience	Salary
Willetta	27	52	62860
Merrie	45	34	57591
Joleen	43	42	61343
Nananne	25	33	75478
Jennica	56	10	71299
Lizzie	55	27	62050
Kaja	28	18	52610
Aubrie	55	18	77702
Ginnie	71	26	88700
Correy	47	25	83288
Rochette	24	12	56123
Hayley	55	51	72758
Mathilda	42	44	62825
Veda	61	31	68511
Jessy	59	41	85482
John	30	5	50000
Mike	25	NULL	NULL
NULL	40	NULL	NULL
Sarah	NULL	3	40000
Sarah	NULL	NULL	NULL

```
# Rename the columns
```

```
df.withColumnRenamed('Name', 'New Name').show()
```

```
+-----+-----+-----+
| New Name | Age | Experience | Salary |
+-----+-----+-----+
| Willetta | 27 | 52 | 62860 |
| Merrie | 45 | 34 | 57591 |
| Joleen | 43 | 42 | 61343 |
| Nananne | 25 | 33 | 75478 |
| Jennica | 56 | 10 | 71299 |
| Lizzie | 55 | 27 | 62050 |
| Kaja | 28 | 18 | 52610 |
| Aubrie | 55 | 18 | 77702 |
| Ginnie | 71 | 26 | 88700 |
| Correy | 47 | 25 | 83288 |
| Rochette | 24 | 12 | 56123 |
| Hayley | 55 | 51 | 72758 |
| Mathilda | 42 | 44 | 62825 |
| Veda | 61 | 31 | 68511 |
| Jessy | 59 | 41 | 85482 |
| John | 30 | 5 | 50000 |
| Mike | 25 | NULL | NULL |
| NULL | 40 | NULL | NULL |
| Sarah | NULL | 3 | 40000 |
| Sarah | NULL | NULL | NULL |
+-----+-----+-----+
```

```
df.na.drop().show()
```

```
+-----+-----+-----+
| Name | Age | Experience | Salary |
+-----+-----+-----+
| Willetta | 27 | 52 | 62860 |
| Merrie | 45 | 34 | 57591 |
| Joleen | 43 | 42 | 61343 |
| Nananne | 25 | 33 | 75478 |
| Jennica | 56 | 10 | 71299 |
| Lizzie | 55 | 27 | 62050 |
| Kaja | 28 | 18 | 52610 |
| Aubrie | 55 | 18 | 77702 |
| Ginnie | 71 | 26 | 88700 |
| Correy | 47 | 25 | 83288 |
| Rochette | 24 | 12 | 56123 |
| Hayley | 55 | 51 | 72758 |
| Mathilda | 42 | 44 | 62825 |
| Veda | 61 | 31 | 68511 |
| Jessy | 59 | 41 | 85482 |
| John | 30 | 5 | 50000 |
+-----+-----+-----+
```

```
df.na.drop(how="all").show()
```

Name	Age	Experience	Salary
Willetta	27	52	62860
Merrie	45	34	57591
Joleen	43	42	61343
Nananne	25	33	75478
Jennica	56	10	71299
Lizzie	55	27	62050
Kaja	28	18	52610
Aubrie	55	18	77702
Ginnie	71	26	88700
Correy	47	25	83288
Rochette	24	12	56123
Hayley	55	51	72758
Mathilda	42	44	62825
Veda	61	31	68511
Jessy	59	41	85482
John	30	5	50000
Mike	25	NULL	NULL
NULL	40	NULL	NULL
Sarah	NULL	3	40000
Sarah	NULL	NULL	NULL

```
df.na.drop(how="any").show()
```

Name	Age	Experience	Salary
Willetta	27	52	62860
Merrie	45	34	57591
Joleen	43	42	61343
Nananne	25	33	75478
Jennica	56	10	71299
Lizzie	55	27	62050
Kaja	28	18	52610
Aubrie	55	18	77702
Ginnie	71	26	88700
Correy	47	25	83288
Rochette	24	12	56123
Hayley	55	51	72758
Mathilda	42	44	62825
Veda	61	31	68511
Jessy	59	41	85482
John	30	5	50000

```
df.na.drop(how="any", thresh=2).show()
```

Name	Age	Experience	Salary
Willetta	27	52	62860
Merrie	45	34	57591
Joleen	43	42	61343
Nananne	25	33	75478
Jennica	56	10	71299
Lizzie	55	27	62050
Kaja	28	18	52610
Aubrie	55	18	77702
Ginnie	71	26	88700
Correy	47	25	83288
Rochette	24	12	56123
Hayley	55	51	72758
Mathilda	42	44	62825
Veda	61	31	68511
Jessy	59	41	85482
John	30	5	50000
Mike	25	NULL	NULL
Sarah	NULL	3	40000

```
df.na.drop(how="any", thresh=3).show()
```

Name	Age	Experience	Salary
Willetta	27	52	62860
Merrie	45	34	57591
Joleen	43	42	61343
Nananne	25	33	75478
Jennica	56	10	71299
Lizzie	55	27	62050
Kaja	28	18	52610
Aubrie	55	18	77702
Ginnie	71	26	88700
Correy	47	25	83288
Rochette	24	12	56123
Hayley	55	51	72758
Mathilda	42	44	62825
Veda	61	31	68511
Jessy	59	41	85482
John	30	5	50000
Sarah	NULL	3	40000

```
df.na.drop(how="any", subset=['Experience']).show()
```

Name	Age	Experience	Salary
Willetta	27	52	62860
Merrie	45	34	57591
Joleen	43	42	61343
Nananne	25	33	75478
Jennica	56	10	71299
Lizzie	55	27	62050
Kaja	28	18	52610
Aubrie	55	18	77702
Ginnie	71	26	88700
Correy	47	25	83288
Rochette	24	12	56123
Hayley	55	51	72758
Mathilda	42	44	62825
Veda	61	31	68511
Jessy	59	41	85482
John	30	5	50000
Sarah	NULL	3	40000

```
df.na.fill({'Name' : 'Missing', 'Salary': 0, 'Experience' : 0, 'Age' : 0}).show()
```

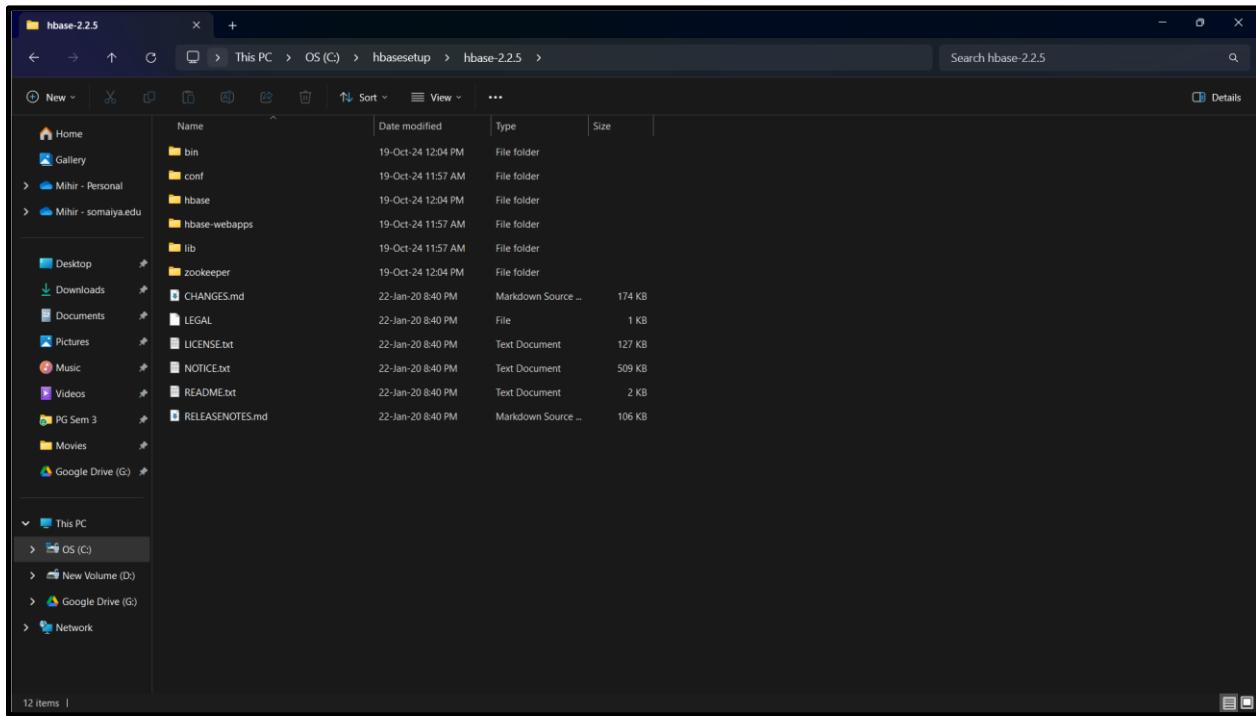
Name	Age	Experience	Salary
Willetta	27	52	62860
Merrie	45	34	57591
Joleen	43	42	61343
Nananne	25	33	75478
Jennica	56	10	71299
Lizzie	55	27	62050
Kaja	28	18	52610
Aubrie	55	18	77702
Ginnie	71	26	88700
Correy	47	25	83288
Rochette	24	12	56123
Hayley	55	51	72758
Mathilda	42	44	62825
Veda	61	31	68511
Jessy	59	41	85482
John	30	5	50000
Mike	25	0	0
Missing	40	0	0
Sarah	0	3	40000
Sarah	0	0	0

Practical 5: Installation of HBase

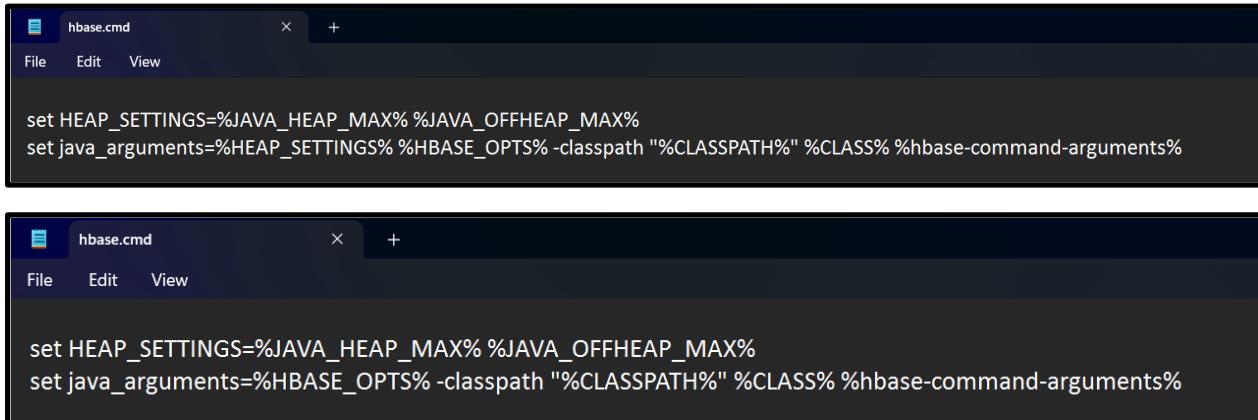
1. Download HBase bin file from [here](#).

Index of /dist/hbase/2.2.5				
Name	Last modified	Size	Description	
Parent Directory	-			
CHANGESE.md	2020-07-03 04:11	174K		
RELEASENOTES.md	2020-07-03 04:11	106K		
api_compare_2.2.5RC0_to_2.2.6.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%.



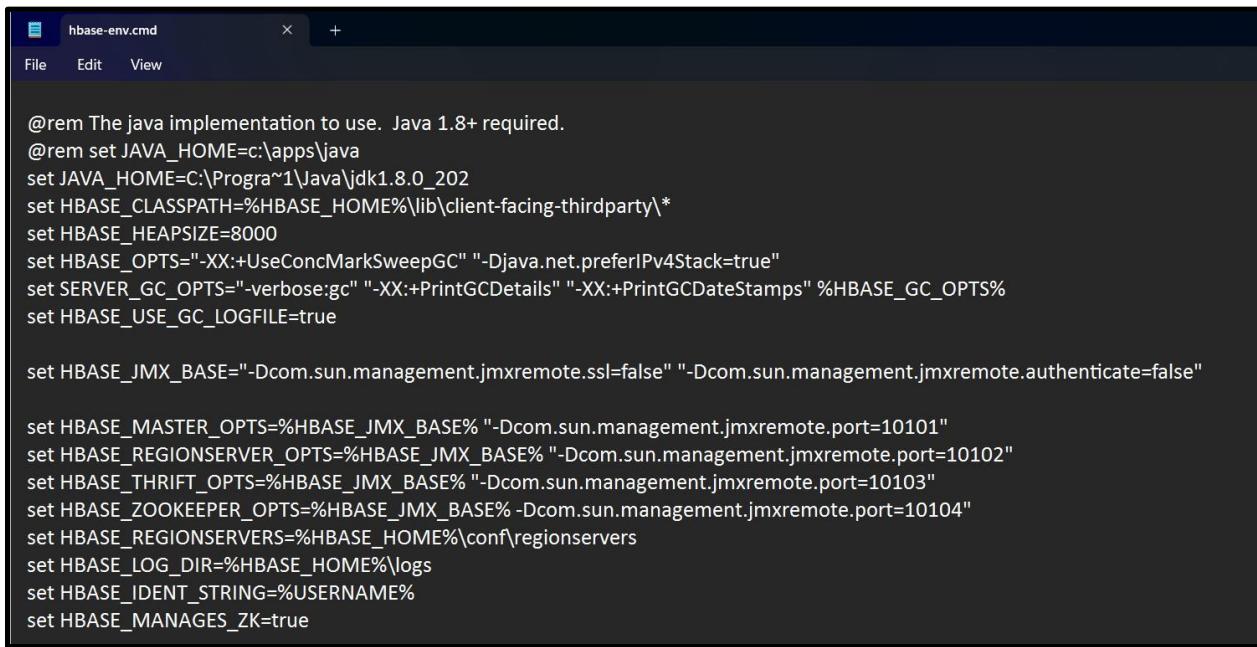
```
hbase.cmd
File Edit View
set HEAP_SETTINGS=%JAVA_HEAP_MAX% %JAVA_OFFHEAP_MAX%
set java_arguments=%HEAP_SETTINGS% %HBASE_OPTS% -classpath "%CLASSPATH%" %CLASS% %hbase-command-arguments%
```



```
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_REGIONSERVERS=%HBASE_HOME%\conf\regionservers
set HBASE_LOG_DIR=%HBASE_HOME%\logs
set HBASE_IDENT_STRING=%USERNAME%
set HBASE_MANAGES_ZK=true

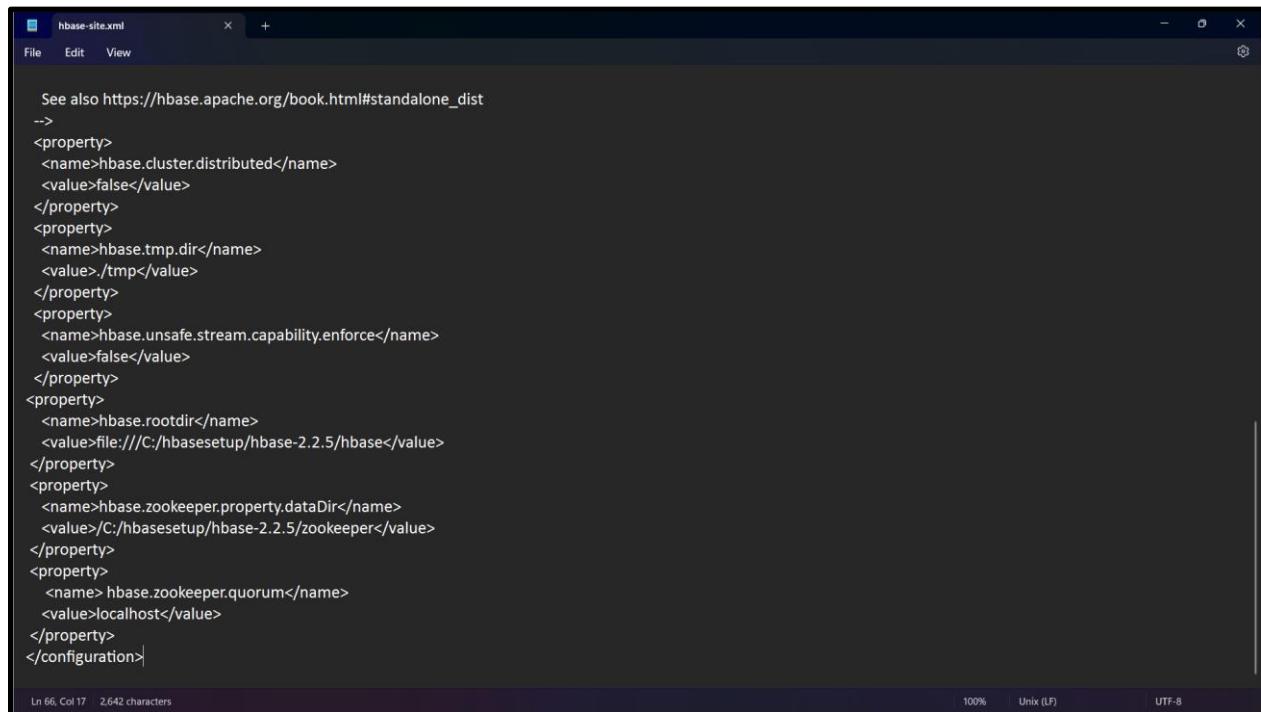
```

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>

```



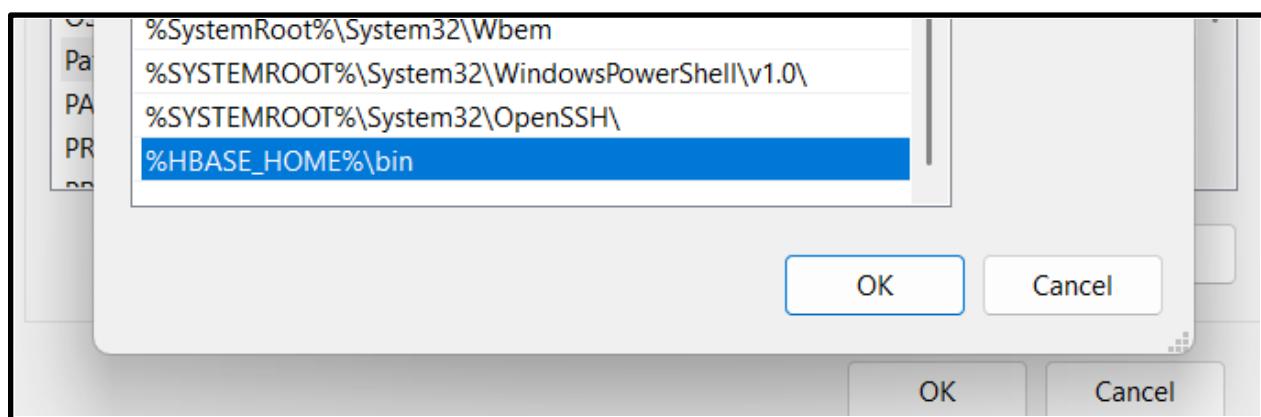
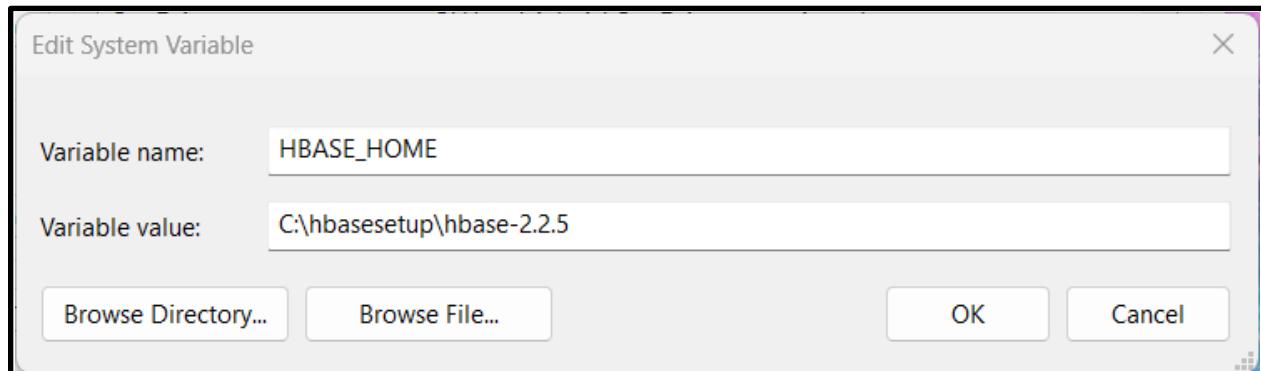
The screenshot shows a code editor window with the title "hbase-site.xml". The content of the file is an XML configuration for HBase. It includes properties for distributed clusters, temporary directories, unsafe stream capabilities, root directories, zookeeper data directories, and quorum settings. The file ends with a closing tag.

```

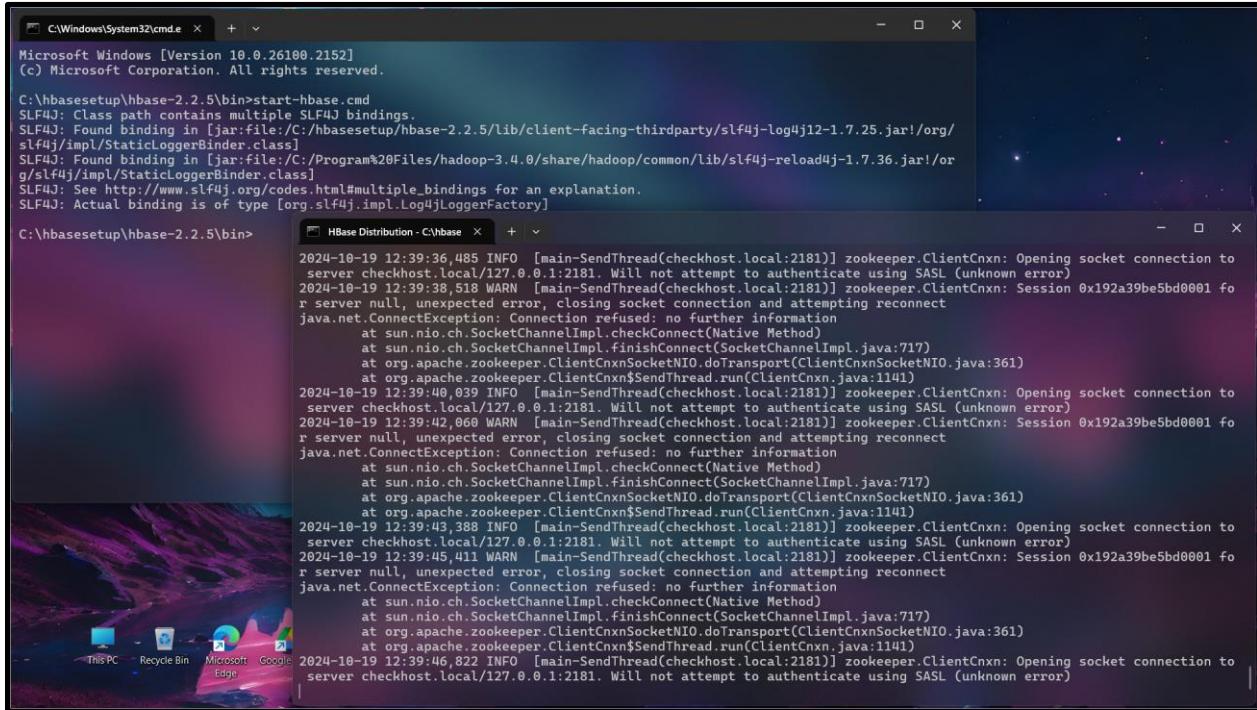
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <!-- See also https://hbase.apache.org/book.html#standalone_dist -->
    <property>
        <name>hbase.cluster.distributed</name>
        <value>false</value>
    </property>
    <property>
        <name>hbase.tmp.dir</name>
        <value>./tmp</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>

```

6. Set the HBase environment variables and its path as well.



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

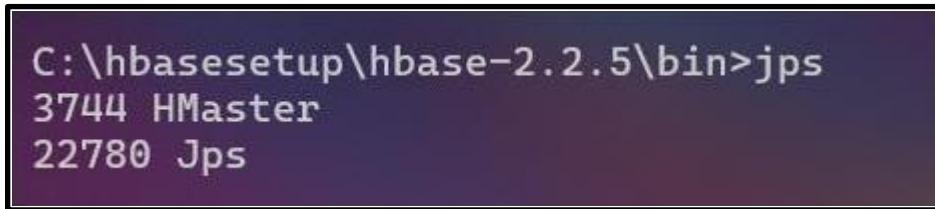


```
C:\Windows\System32\cmd.exe + 
Microsoft Windows [Version 10.0.26100.2152]
(c) Microsoft Corporation. All rights reserved.

C:\hbasesetup\hbase-2.2.5\bin>start-hbase.cmd
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/jimpl/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/jimpl/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>
[Output continues with several lines of Java stack traces and logs from ZooKeeper and HBase clients.]
```

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



```
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.

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

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, rf76a601273e834267b55c0cd12474590283fd4c, 2020? 05? 21? ??? 18:34:48 CST
[ERROR] Terminal initialization failed; falling back to unsupported
java.lang.NoClassDefFoundError: Could not initialize class org.fusesource.jansi.internal.Kernel32
        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:\hbasesetup\hbase-2.2.5\bin>hbase shell
at org.jruby.runtime.callsite.CachingCallSite.call(CachingCallSite.java:131)
at C_3a_.hbasesetup.hbase_minus_2_dot_2_dot_5_dot_hbase.invokeOther172:print_banner(C:\hbasesetup\hbase-2.2.5\bin\hirb.rb:190)
at C_3a_.hbasesetup.hbase_minus_2_dot_2_dot_5_dot_hbase.RUBY$script(C:\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:747)
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
'stty' is not recognized as an internal or external command,
operable program or batch file.
hbase(main):001:0> |
```