Install Hadoop 3.2.1 on Windows 10 Step by Step Guide

This detailed step-by-step guide shows you how to install the latest Hadoop (v3.2.1) on Windows 10. It's based on the previous articles I published with some updates to reflect the feedback collected from readers to make it easier for everyone to install.

Please follow all the instructions carefully. Once you complete the steps, you will have a shiny pseudo-distributed single node Hadoop to work with.

Step 1 - Download Hadoop binary package

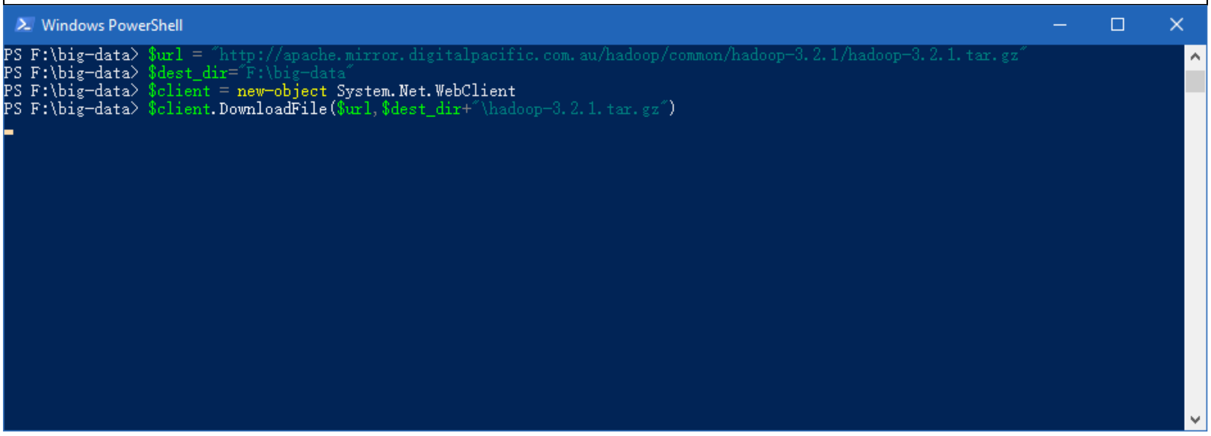
Select download mirror link

Go to download page of the official website:

[Apache Download Mirrors - Hadoop 3.2.1]

Open PowerShell and then run the following command lines one by one:

```
$dest_dir="F:\big-data"
$url = " https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz"
$client = new-object System.Net.WebClient
$client.DownloadFile($url,$dest_dir+"\hadoop-3.2.1.tar.gz")
```



```
PS F:\big-data> $url = "http://apache.mirror.digitalpacific.com.au/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz"
PS F:\big-data> $dest_dir="F:\big-data"
PS F:\big-data> $client = new-object System.Net.WebClient
PS F:\big-data> $client.DownloadFile($url,$dest_dir+"\hadoop-3.2.1.tar.gz")
```

It may take a few minutes to download.
Once the download completes, you can verify it:

```
PS F:\big-data> cd $dest_dir
PS F:\big-data> ls


    Directory: F:\big-data
```

```
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----       18/01/2020  11:01 AM      359196911 hadoop-3.2.1.tar.gz


PS F:\big-data>
```

Step 2 - Unpack the package

Now we need to unpack the downloaded package using GUI tool (like 7 Zip) or command line. For me, I will use git bash to unpack it.

Open git bash and change the directory to the destination folder:

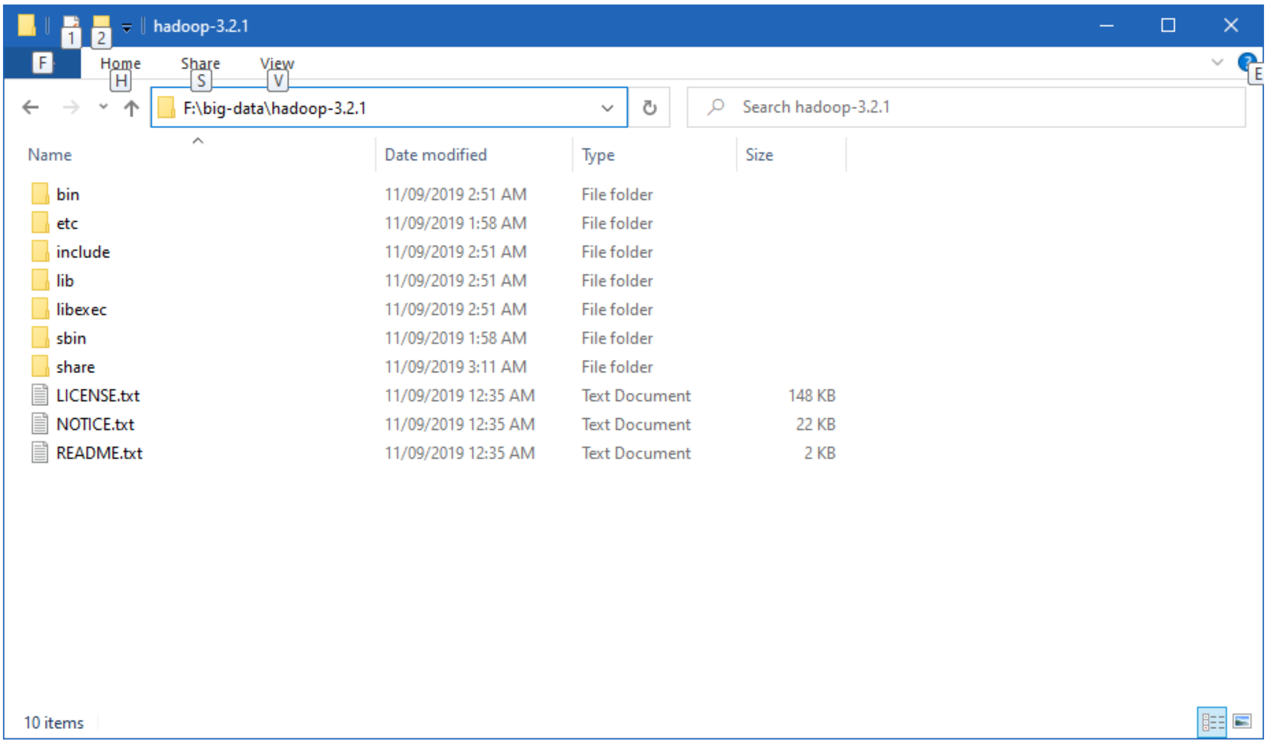
```
cd F:/big-data
```

And then run the following command to unzip:

```
tar -xvzf  hadoop-3.2.1.tar.gz
```

The command will take quite a few minutes as there are numerous files included and the latest version introduced many new features.

After the unzip command is completed, a new folder **hadoop-3.2.1** is created under the destination
folder.



Name	Date modified	Type	Size
bin	11/09/2019 2:51 AM	File folder	
etc	11/09/2019 1:58 AM	File folder	
include	11/09/2019 2:51 AM	File folder	
lib	11/09/2019 2:51 AM	File folder	
libexec	11/09/2019 2:51 AM	File folder	
sbin	11/09/2019 1:58 AM	File folder	
share	11/09/2019 3:11 AM	File folder	
LICENSE.txt	11/09/2019 12:35 AM	Text Document	148 KB
NOTICE.txt	11/09/2019 12:35 AM	Text Document	22 KB
README.txt	11/09/2019 12:35 AM	Text Document	2 KB

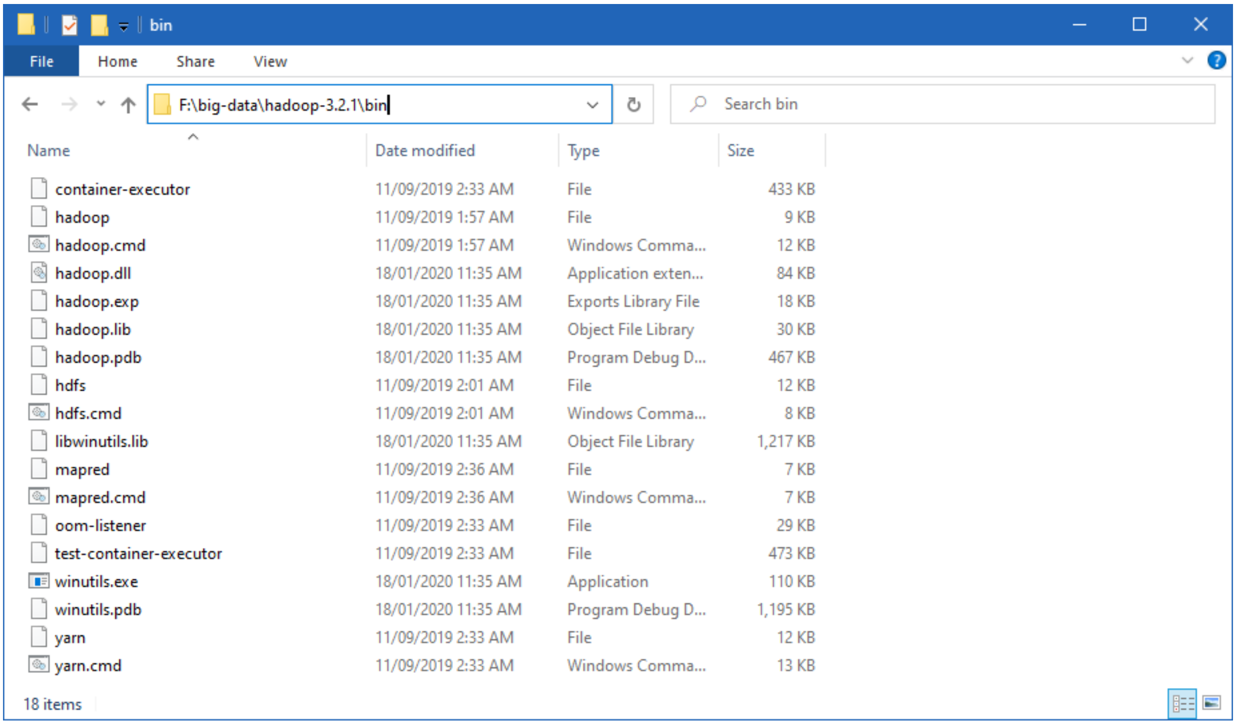
Step 3 - Install Hadoop native IO binary

Hadoop on Linux includes optional Native IO support. However Native IO is mandatory on Windows and without it you will not be able to get your installation working. The Windows native IO libraries are not included as part of Apache Hadoop release. Thus we need to build and install it.

The build may take about one hourand to save our time, we can just download the binary package from github or LMS and copy at F:/big-data/Hadoop-3.2.1/bin/

- hadoop.dll
- hadoop.exp
- hadoop.lib
- hadoop.pdb
- libwinutils.lib
- winutils.exe
- winutils.pdb

After this, the **bin** folder looks like the following:



F:\big-data\hadoop-3.2.1\bin

Name	Date modified	Type	Size
container-executor	11/09/2019 2:33 AM	File	433 KB
hadoop	11/09/2019 1:57 AM	File	9 KB
hadoop.cmd	11/09/2019 1:57 AM	Windows Comma...	12 KB
hadoop.dll	18/01/2020 11:35 AM	Application exten...	84 KB
hadoop.exp	18/01/2020 11:35 AM	Exports Library File	18 KB
hadoop.lib	18/01/2020 11:35 AM	Object File Library	30 KB
hadoop.pdb	18/01/2020 11:35 AM	Program Debug D...	467 KB
hdfs	11/09/2019 2:01 AM	File	12 KB
hdfs.cmd	11/09/2019 2:01 AM	Windows Comma...	8 KB
libwinutils.lib	18/01/2020 11:35 AM	Object File Library	1,217 KB
mapred	11/09/2019 2:36 AM	File	7 KB
mapred.cmd	11/09/2019 2:36 AM	Windows Comma...	7 KB
oom-listener	11/09/2019 2:33 AM	File	29 KB
test-container-executor	11/09/2019 2:33 AM	File	473 KB
winutils.exe	18/01/2020 11:35 AM	Application	110 KB
winutils.pdb	18/01/2020 11:35 AM	Program Debug D...	1,195 KB
yarn	11/09/2019 2:33 AM	File	12 KB
yarn.cmd	11/09/2019 2:33 AM	Windows Comma...	13 KB

18 items

Step 4 - (Optional) Java JDK installation

ava JDK is required to run Hadoop. If you have not installed Java JDK please install it.

You can install JDK 8 from the following page:

https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Once you complete the installation, please run the following command in PowerShell or Git Bash to verify:

```
$ java -version
java version "1.8.0_161"
Java(TM) SE Runtime Environment (build 1.8.0_161-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)
```

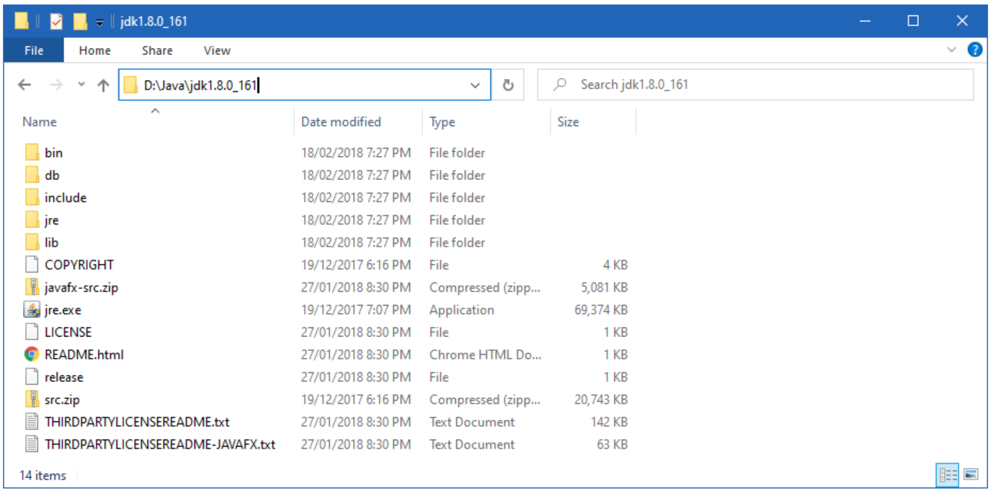
If you got error about 'cannot find java command or executable'. Don't worry we will resolve this in the following step.

Step 5 - Configure environment variables

Configure JAVA_HOME environment variable

As mentioned earlier, Hadoop requires Java and we need to configure **JAVA_HOME** environment variable (though it is not mandatory but I recommend it).

First, we need to find out the location of Java SDK. In my system, the path is: **D:\Java\jdk1.8.0_161**.



D:\Java\jdk1.8.0_161

Name	Date modified	Type	Size
bin	18/02/2018 7:27 PM	File folder	
db	18/02/2018 7:27 PM	File folder	
include	18/02/2018 7:27 PM	File folder	
jre	18/02/2018 7:27 PM	File folder	
lib	18/02/2018 7:27 PM	File folder	
COPYRIGHT	19/12/2017 6:16 PM	File	4 KB
javafx-src.zip	27/01/2018 8:30 PM	Compressed (zipp...	5,081 KB
jre.exe	19/12/2017 7:07 PM	Application	69,374 KB
LICENSE	27/01/2018 8:30 PM	File	1 KB
README.html	27/01/2018 8:30 PM	Chrome HTML Do...	1 KB
release	27/01/2018 8:30 PM	File	1 KB
src.zip	19/12/2017 6:16 PM	Compressed (zipp...	20,743 KB
THIRDPARTYLICENSEREADME.txt	27/01/2018 8:30 PM	Text Document	142 KB
THIRDPARTYLICENSEREADME-JAVAFX.txt	27/01/2018 8:30 PM	Text Document	63 KB

14 items

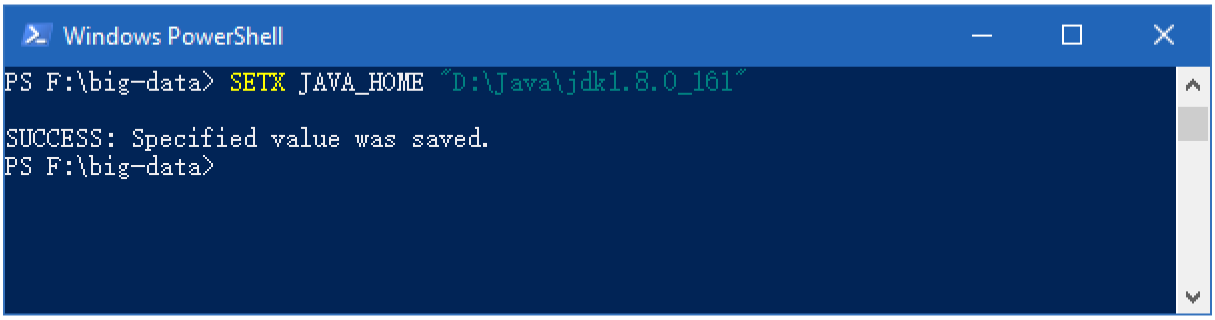
Your location can be different depends on where you install your JDK.

And then run the following command in the previous PowerShell window:

```
SETX JAVA_HOME "D:\Java\jdk1.8.0_161"
```

Remember to quote the path especially if you have spaces in your JDK path.

The output looks like the following:



```
PS F:\big-data> SETX JAVA_HOME "D:\Java\jdk1.8.0_161"

SUCCESS: Specified value was saved.
PS F:\big-data>
```

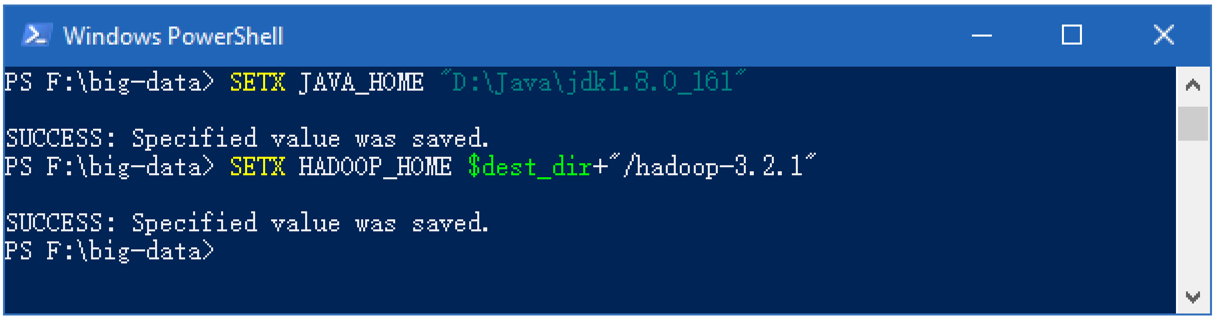
Configure HADOOP_HOME environment variable

Similarly we need to create a new environment variable for **HADOOP_HOME** using the following command. The path should be your extracted Hadoop folder. For my environment it is: **F:\big-data\hadoop-3.2.1**.

If you used PowerShell to download and if the window is still open, you can simply run the following command:

```
SETX HADOOP_HOME $dest_dir+"/hadoop-3.2.1"
```

The output looks like the following screenshot:

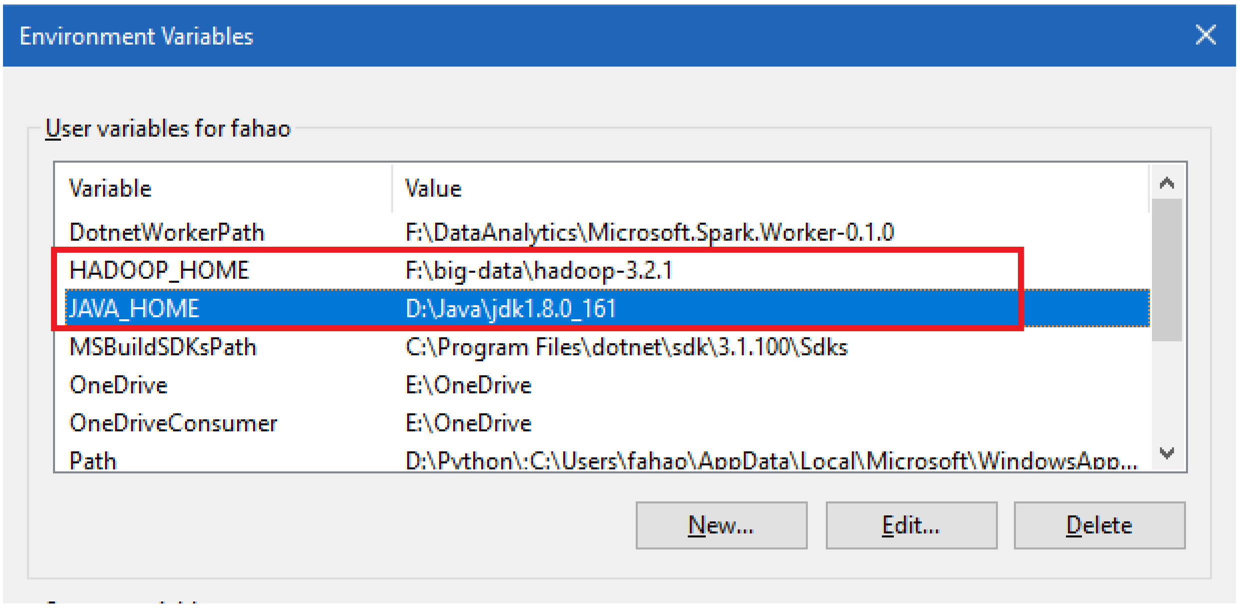


```
PS F:\big-data> SETX JAVA_HOME "D:\Java\jdk1.8.0_161"

SUCCESS: Specified value was saved.
PS F:\big-data> SETX HADOOP_HOME $dest_dir+"/hadoop-3.2.1"

SUCCESS: Specified value was saved.
PS F:\big-data>
```

Alternatively, you can specify the full path:

```
SETX HADOOP_HOME "F:\big-data\hadoop-3.2.1"
```

Now you can also verify the two environment variables in the system:



Environment Variables

User variables for fahao

Variable	Value
DotnetWorkerPath	F:\DataAnalytics\Microsoft.Spark.Worker-0.1.0
HADOOP_HOME	F:\big-data\hadoop-3.2.1
JAVA_HOME	D:\Java\jdk1.8.0_161
MSBuildSDKsPath	C:\Program Files\dotnet\sdk\3.1.100\Sdks
OneDrive	E:\OneDrive
OneDriveConsumer	E:\OneDrive
Path	D:\Python\;C:\Users\fahao\AppData\Local\Microsoft\WindowsApp...

Configure PATH environment variable

Once we finish setting up the above two environment variables, we need to add the **bin** folders to the **PATH** environment variable.

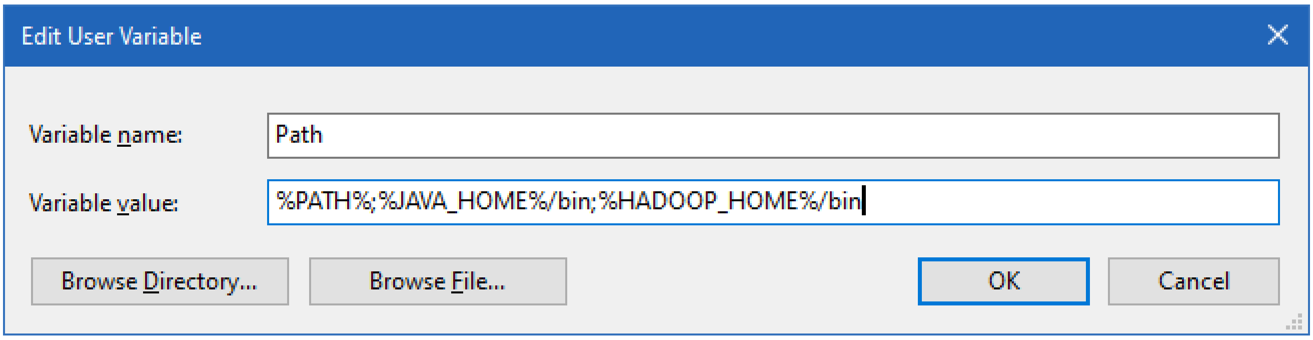
If **PATH** environment exists in your system, you can also manually add the following two paths to it:

- %JAVA_HOME%/bin
- %HADOOP_HOME%/bin

Alternatively, you can run the following command to add them:

```
setx PATH "$env:PATH;$env:JAVA_HOME/bin;$env:HADOOP_HOME/bin"
```

If you don't have other user variables setup in the system, you can also directly add a **Path** environment variable that references others to make it short:

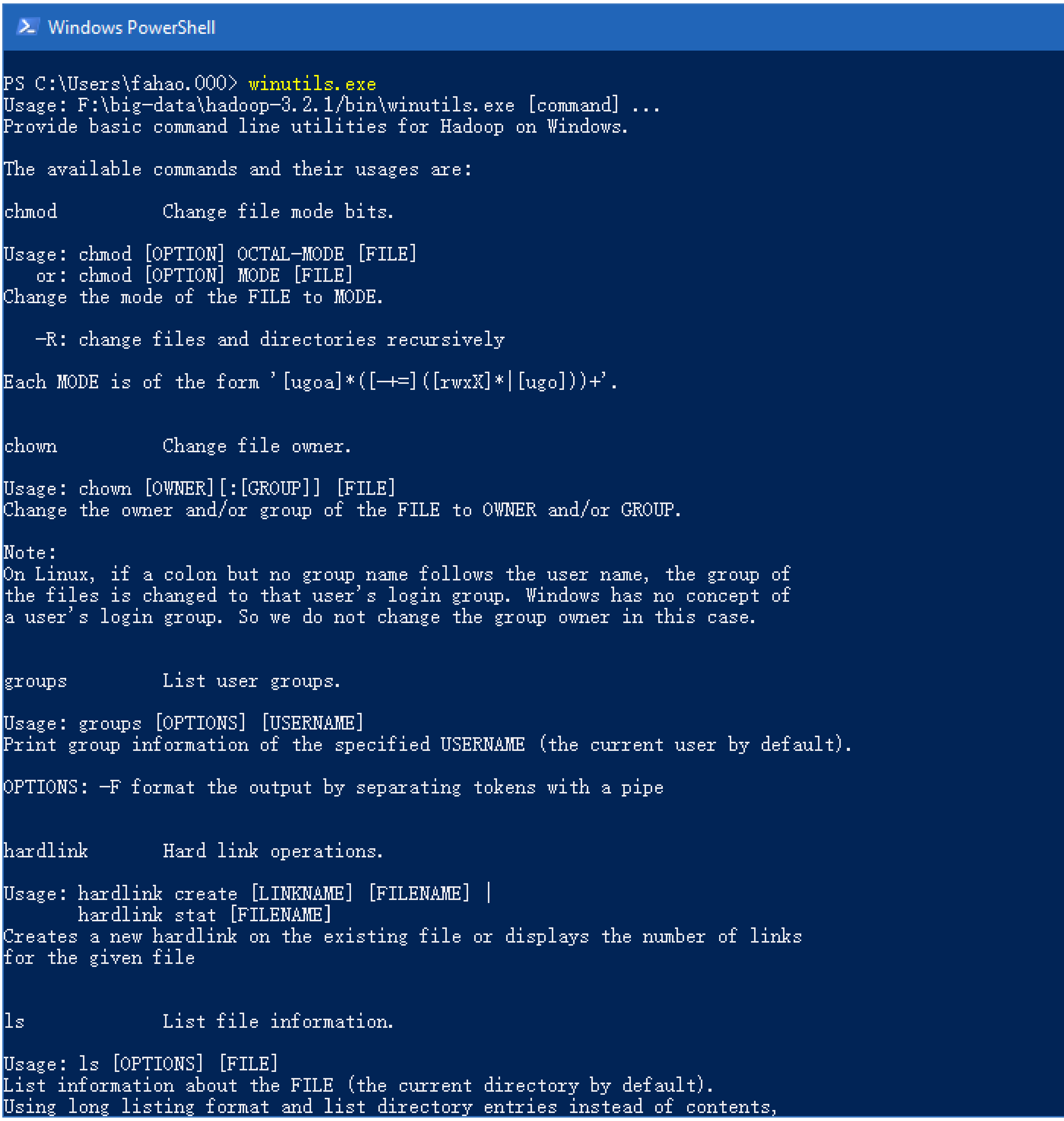


Edit User Variable

Variable name: Path

Variable value: %PATH%;%JAVA_HOME%/bin;%HADOOP_HOME%/bin

Close PowerShell window and open a new one and type **winutils.exe** directly to verify that our above steps are completed successfully:



```
PS C:\Users\fahao.000> winutils.exe
Usage: F:\big-data\hadoop-3.2.1/bin\winutils.exe [command] ...
Provide basic command line utilities for Hadoop on Windows.

The available commands and their usages are:

chmod          Change file mode bits.

Usage: chmod [OPTION] OCTAL-MODE [FILE]
   or: chmod [OPTION] MODE [FILE]
Change the mode of the FILE to MODE.

   -R: change files and directories recursively

Each MODE is of the form '[ugoa]*([-+=]([rwxX]*|[ugo]))+'.


chown          Change file owner.

Usage: chown [OWNER][:[GROUP]] [FILE]
Change the owner and/or group of the FILE to OWNER and/or GROUP.

Note:
On Linux, if a colon but no group name follows the user name, the group of
the files is changed to that user's login group. Windows has no concept of
a user's login group. So we do not change the group owner in this case.


groups         List user groups.

Usage: groups [OPTIONS] [USERNAME]
Print group information of the specified USERNAME (the current user by default).

OPTIONS: -F format the output by separating tokens with a pipe


hardlink       Hard link operations.

Usage: hardlink create [LINKNAME] [FILENAME] |
       hardlink stat [FILENAME]
Creates a new hardlink on the existing file or displays the number of links
for the given file


ls             List file information.

Usage: ls [OPTIONS] [FILE]
List information about the FILE (the current directory by default).
Using long listing format and list directory entries instead of contents,
```

You should also be able to run the following command:

```
hadoop -version
java version "1.8.0_161"
Java(TM) SE Runtime Environment (build 1.8.0_161-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)
```

Troubleshoot 1 :

If get error MSVCR100.dll during execution of winutils.exe following task needs to be done (file available on LMS)

- MSVCR100.dll : Windows 10: winutils.exe doesn't workFolder
- Edit
-
- Download MSVCR100.dll : in case Windows 10: winutils.exe doesn't work and copy the file at C:/Windows/system32

Step 6 - Configure Hadoop

Now we are ready to configure the most important part - Hadoop configurations which involves Core, YARN, MapReduce, HDFS configurations. So, download five configuration files from LMS folder link and copy into the : **F:\big-data\hadoop-3.2.1\etc\hadoop**. (my environment, the actual path)

In hadoop-env.cmd file needs to set JAVA_HOME as per your java path.

```
set JAVA_HOME=C:\Progra~1\Java\jdk1.8.0_131\
```

Notion to set env variable if it contains white spaces:

```
Progra~1 = 'Program Files'
Progra~2 = 'Program Files(x86)'
```

Step 7 - Initialise HDFS & bug fix

Run the following command in Command Prompt

```
hdfs namenode -format
```

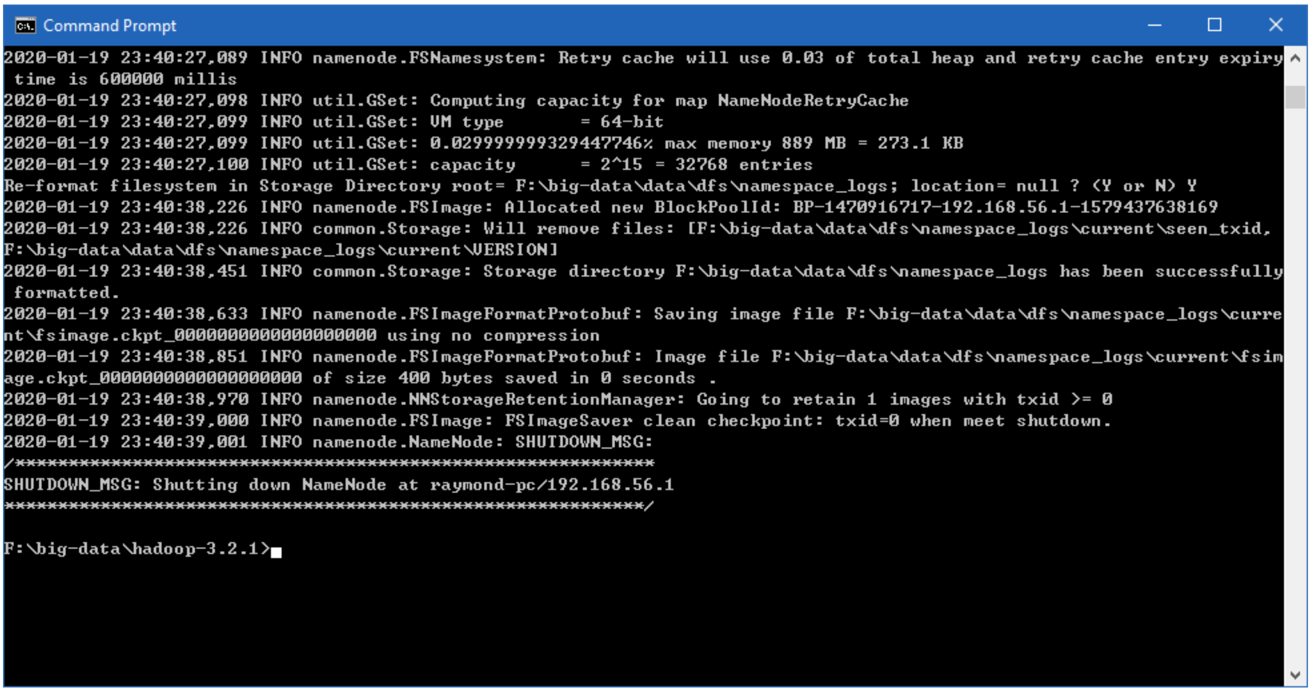
This command failed with the following error and we need to fix it:

```
2020-01-18 13:36:03,021 ERROR namenode.NameNode: Failed to start namenode.
java.lang.UnsupportedOperationException
        at java.nio.file.Files.setPosixFilePermissions(Files.java:2044)
        at
org.apache.hadoop.hdfs.server.common.Storage$StorageDirectory.clearDirectory(Storage.java:452)
        at org.apache.hadoop.hdfs.server.namenode.NNStorage.format(NNStorage.java:591)
        at org.apache.hadoop.hdfs.server.namenode.NNStorage.format(NNStorage.java:613)
        at org.apache.hadoop.hdfs.server.namenode.FSImage.format(FSImage.java:188)
        at org.apache.hadoop.hdfs.server.namenode.NameNode.format(NameNode.java:1206)
        at org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNode.java:1649)
        at org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:1759)
```

```
2020-01-18 13:36:03,025 INFO util.ExitUtil: Exiting with status 1:
java.lang.UnsupportedOperationException
```

Refer to the following sub section (About 3.2.1 HDFS bug on Windows) about the details of fixing this problem.

Once this is fixed, the format command (hdfs namenode -format) will show something like the following:



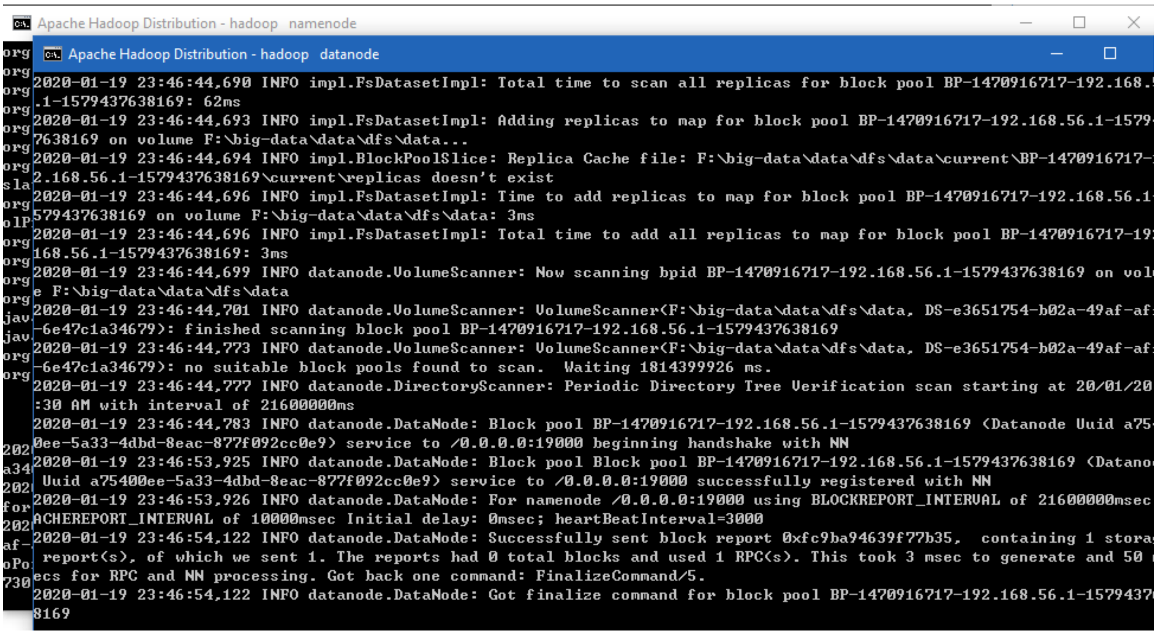
```
2020-01-19 23:40:27,089 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry
 time is 600000 millis
2020-01-19 23:40:27,098 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2020-01-19 23:40:27,099 INFO util.GSet: VM type       = 64-bit
2020-01-19 23:40:27,099 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
2020-01-19 23:40:27,100 INFO util.GSet: capacity      = 2^15 = 32768 entries
Re-format filesystem in Storage Directory root= F:\big-data\data\dfs\namespace_logs; location= null ? (Y or N) Y
2020-01-19 23:40:38,226 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1470916717-192.168.56.1-1579437638169
2020-01-19 23:40:38,226 INFO common.Storage: Will remove files: [F:\big-data\data\dfs\namespace_logs\current\seen_txid,
F:\big-data\data\dfs\namespace_logs\current\VERSION]
2020-01-19 23:40:38,451 INFO common.Storage: Storage directory F:\big-data\data\dfs\namespace_logs has been successfully
 formatted.
2020-01-19 23:40:38,633 INFO namenode.FSImageFormatProtobuf: Saving image file F:\big-data\data\dfs\namespace_logs\curre
nt\fsimage.ckpt_0000000000000000000 using no compression
2020-01-19 23:40:38,851 INFO namenode.FSImageFormatProtobuf: Image file F:\big-data\data\dfs\namespace_logs\current\fsim
age.ckpt_0000000000000000000 of size 400 bytes saved in 0 seconds .
2020-01-19 23:40:38,970 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2020-01-19 23:40:39,000 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2020-01-19 23:40:39,001 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at raymond-pc/192.168.56.1
************************************************************/

F:\big-data\hadoop-3.2.1>
```

Step 8 - Start HDFS daemons

Run the following command to start HDFS daemons in Command Prompt:

```
%HADOOP_HOME%\sbin\start-dfs.cmd
```

Two Command Prompt windows will open: one for datanode and another for namenode as the following screenshot shows:


Apache Hadoop Distribution - hadoop   namenode
Apache Hadoop Distribution - hadoop   datanode

```
2020-01-19 23:46:44,690 INFO impl.FsDatasetImpl: Total time to scan all replicas for block pool BP-1470916717-192.168.5
.1-1579437638169: 62ms
2020-01-19 23:46:44,693 INFO impl.FsDatasetImpl: Adding replicas to map for block pool BP-1470916717-192.168.56.1-1579
7638169 on volume F:\big-data\data\dfs\data...
2020-01-19 23:46:44,694 INFO impl.BlockPoolSlice: Replica Cache file: F:\big-data\data\dfs\data\current\BP-1470916717-1
2.168.56.1-1579437638169\current\replicas doesn't exist
2020-01-19 23:46:44,696 INFO impl.FsDatasetImpl: Time to add replicas to map for block pool BP-1470916717-192.168.56.1-
579437638169 on volume F:\big-data\data\dfs\data: 3ms
2020-01-19 23:46:44,696 INFO impl.FsDatasetImpl: Total time to add all replicas to map for block pool BP-1470916717-19
168.56.1-1579437638169: 3ms
2020-01-19 23:46:44,699 INFO datanode.VolumeScanner: Now scanning bpid BP-1470916717-192.168.56.1-1579437638169 on vol
e F:\big-data\data\dfs\data
2020-01-19 23:46:44,701 INFO datanode.VolumeScanner: VolumeScanner(F:\big-data\data\dfs\data, DS-e3651754-b02a-49af-af
-6e47c1a34679): finished scanning block pool BP-1470916717-192.168.56.1-1579437638169
2020-01-19 23:46:44,773 INFO datanode.VolumeScanner: VolumeScanner(F:\big-data\data\dfs\data, DS-e3651754-b02a-49af-af
-6e47c1a34679): no suitable block pools found to scan.  Waiting 1814399926 ms.
2020-01-19 23:46:44,777 INFO datanode.DirectoryScanner: Periodic Directory Tree Verification scan starting at 20/01/20
:30 AM with interval of 21600000ms
2020-01-19 23:46:44,783 INFO datanode.DataNode: Block pool BP-1470916717-192.168.56.1-1579437638169 (Datanode Uuid a75
0ee-5a33-4dbd-8eac-877f092cc0e9) service to /0.0.0.0:19000 beginning handshake with NN
2020-01-19 23:46:53,925 INFO datanode.DataNode: Block pool Block pool BP-1470916717-192.168.56.1-1579437638169 (Datano
 Uuid a75400ee-5a33-4dbd-8eac-877f092cc0e9) service to /0.0.0.0:19000 successfully registered with NN
2020-01-19 23:46:53,926 INFO datanode.DataNode: For namenode /0.0.0.0:19000 using BLOCKREPORT_INTERVAL of 21600000msec
ACHEREPORT_INTERVAL of 10000msec Initial delay: 0msec; heartBeatInterval=3000
2020-01-19 23:46:54,122 INFO datanode.DataNode: Successfully sent block report 0xfc9ba94639f77b35,  containing 1 stora
 report(s), of which we sent 1. The reports had 0 total blocks and used 1 RPC(s). This took 3 msec to generate and 50
ecs for RPC and NN processing. Got back one command: FinalizeCommand/5.
2020-01-19 23:46:54,122 INFO datanode.DataNode: Got finalize command for block pool BP-1470916717-192.168.56.1-157943
8169
```

Troubleshoot 2: if Namenode failure

Replace Jar file (hadoop-hdfs-3.2.1.jar) on location hadoop-3.2.1\share\hadoop\hdfs

The said file uploaded on LMS for easy access

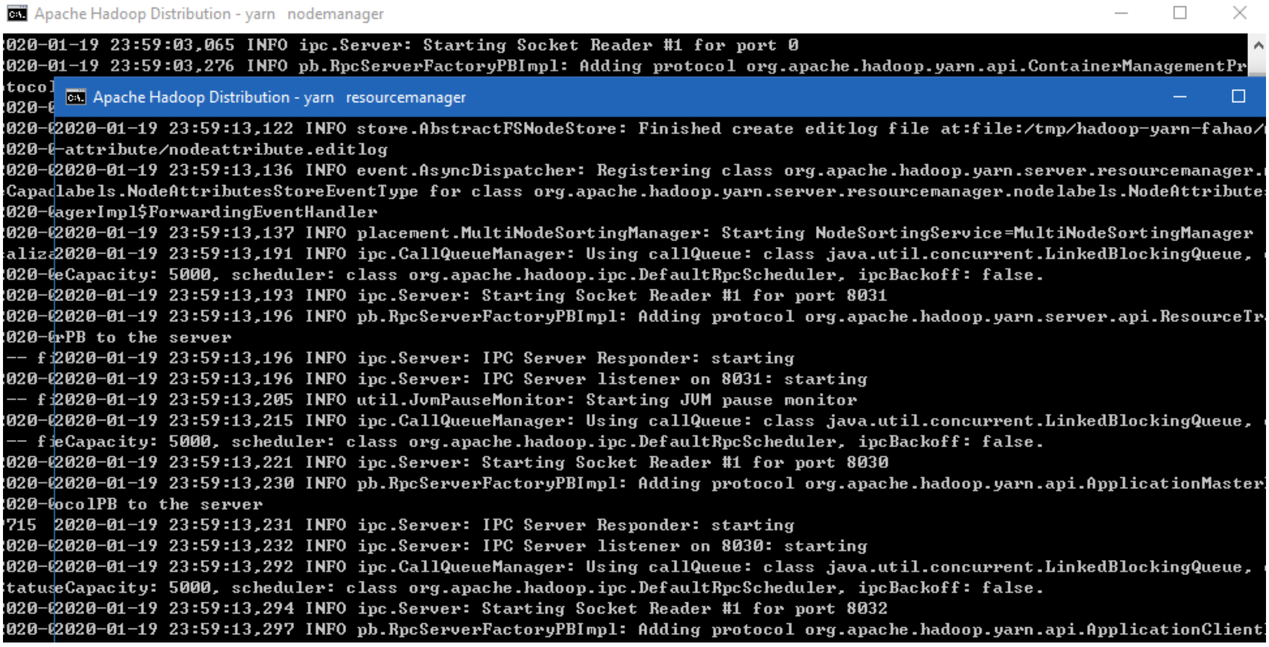
Step 9 - Start YARN daemons

warning You may encounter permission issues if you start YARN daemons using normal user. To ensure you don't encounter any issues. Please open a Command Prompt window using **Run as administrator**.
Alternatively, you can follow this comment on this page which doesn't require Administrator permission using a local Windows account:
https://kontext.tech/column/hadoop/377/latest-hadoop-321-installation-on-windows-10-step-by-step-guide#comment314

Run the following command in an elevated Command Prompt window (Run as administrator) to start YARN daemons:

```
%HADOOP_HOME%\sbin\start-yarn.cmd
```

Similarly two Command Prompt windows will open: one for resource manager and another for node manager as the following screenshot shows:


Apache Hadoop Distribution - yarn nodemanager
Apache Hadoop Distribution - yarn resourcemanager

```
020-01-19 23:59:03,065 INFO ipc.Server: Starting Socket Reader #1 for port 0
020-01-19 23:59:03,276 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ContainerManagementPr
tocol
020-0
020-02020-01-19 23:59:13,122 INFO store.AbstractFSNodeStore: Finished create editlog file at:file:/tmp/hadoop-yarn-fahao/
020-0-attribute/nodeattribute.editlog
020-02020-01-19 23:59:13,136 INFO event.AsyncDispatcher: Registering class org.apache.hadoop.yarn.server.resourcemanager.
Capadlabels.NodeAttributesStoreEventType for class org.apache.hadoop.yarn.server.resourcemanager.nodelabels.NodeAttribute
020-0agerImpl$ForwardingEventHandler
020-02020-01-19 23:59:13,137 INFO placement.MultiNodeSortingManager: Starting NodeSortingService=MultiNodeSortingManager
aliz2020-01-19 23:59:13,191 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue,
020-0eCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false.
020-02020-01-19 23:59:13,193 INFO ipc.Server: Starting Socket Reader #1 for port 8031
020-02020-01-19 23:59:13,196 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.server.api.ResourceTr
020-0rPB to the server
-- f2020-01-19 23:59:13,196 INFO ipc.Server: IPC Server Responder: starting
020-02020-01-19 23:59:13,196 INFO ipc.Server: IPC Server listener on 8031: starting
-- f2020-01-19 23:59:13,205 INFO util.JvmPauseMonitor: Starting JVM pause monitor
020-02020-01-19 23:59:13,215 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue,
-- feCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false.
020-02020-01-19 23:59:13,221 INFO ipc.Server: Starting Socket Reader #1 for port 8030
020-02020-01-19 23:59:13,230 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationMaster
020-0ocolPB to the server
715 2020-01-19 23:59:13,231 INFO ipc.Server: IPC Server Responder: starting
020-02020-01-19 23:59:13,232 INFO ipc.Server: IPC Server listener on 8030: starting
020-02020-01-19 23:59:13,292 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue,
tatuseCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false.
020-02020-01-19 23:59:13,294 INFO ipc.Server: Starting Socket Reader #1 for port 8032
020-02020-01-19 23:59:13,297 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationClient
```

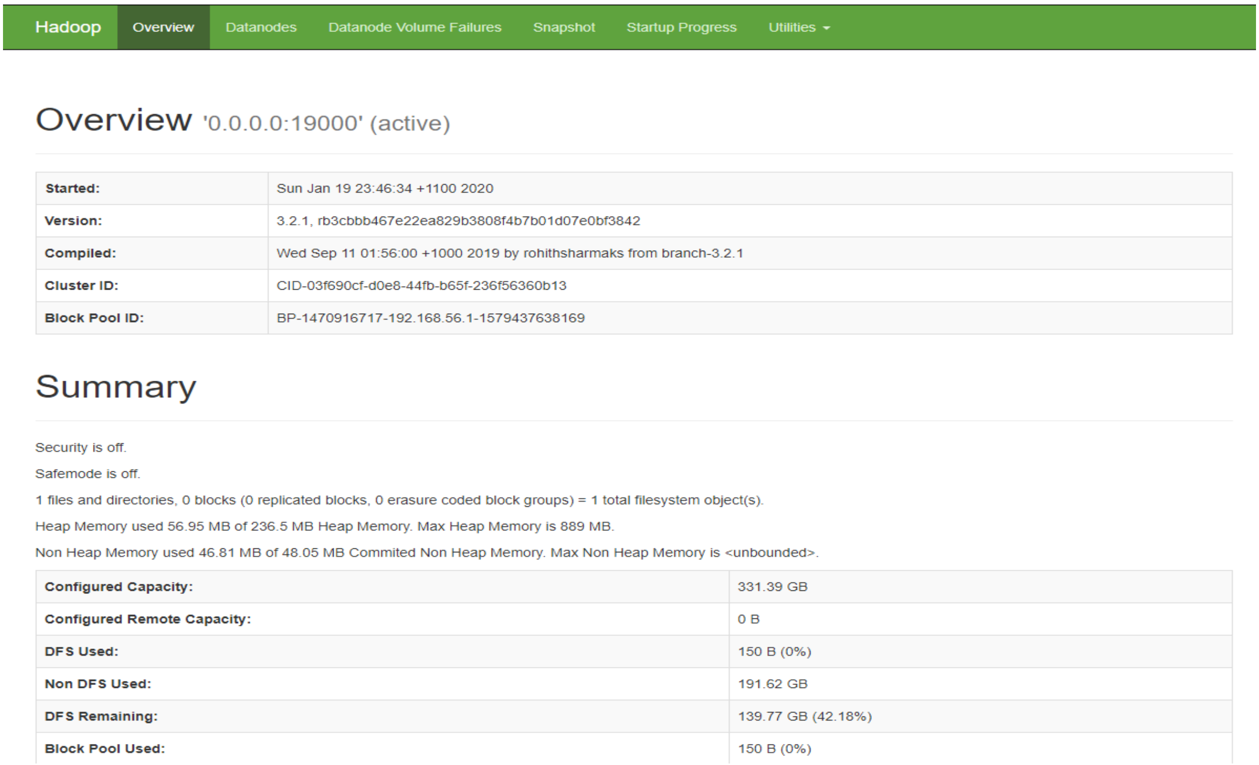
Step 10 - Useful Web portals exploration

The daemons also host websites that provide useful information about the cluster.

HDFS Namenode information UI

http://localhost:9870/dfshealth.html#tab-overview

The website looks like the following screenshot:



Overview '0.0.0.0:19000' (active)

Started:	Sun Jan 19 23:46:34 +1100 2020
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842
Compiled:	Wed Sep 11 01:56:00 +1000 2019 by rohithsharmaks from branch-3.2.1
Cluster ID:	CID-03f690cf-d0e8-44fb-b65f-236f56360b13
Block Pool ID:	BP-1470916717-192.168.56.1-1579437638169

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 56.95 MB of 236.5 MB Heap Memory. Max Heap Memory is 889 MB.

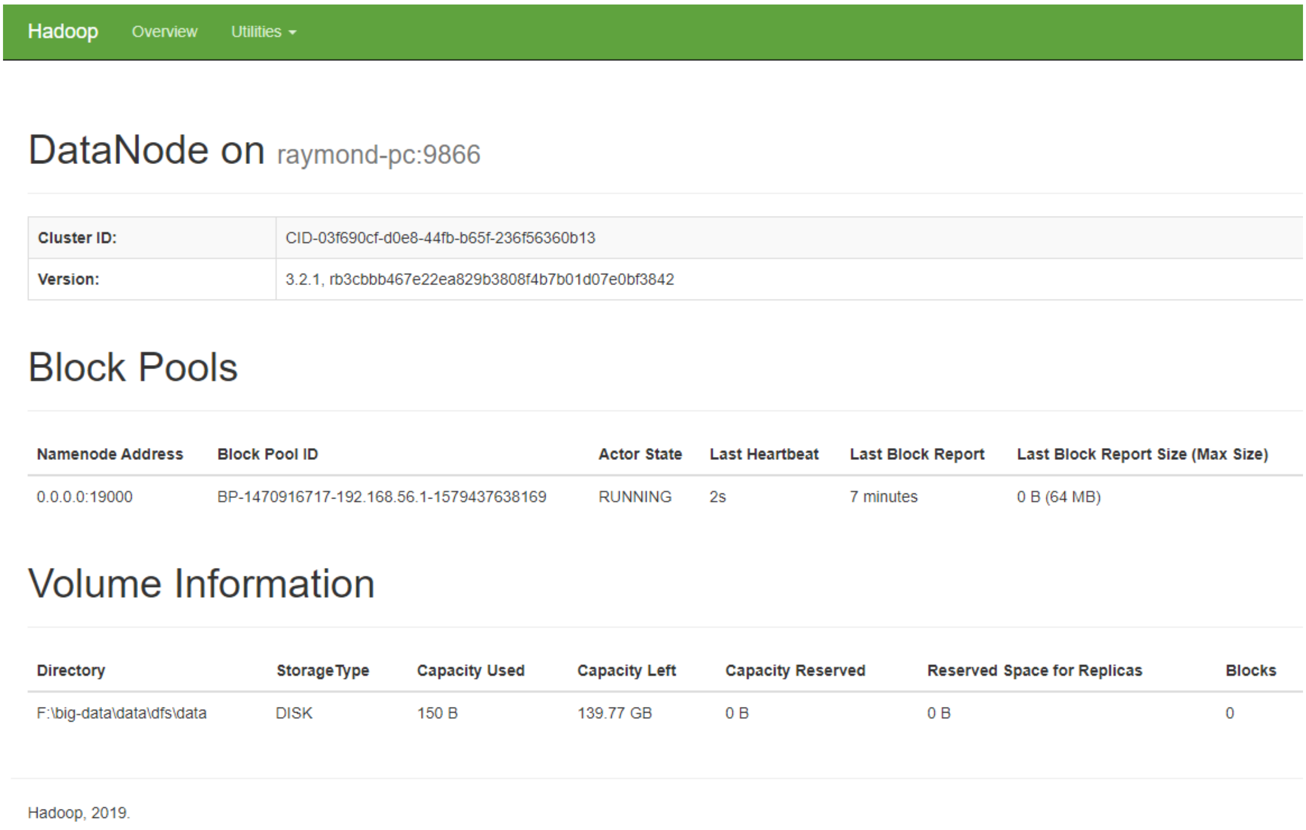
Non Heap Memory used 46.81 MB of 48.05 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	331.39 GB
Configured Remote Capacity:	0 B
DFS Used:	150 B (0%)
Non DFS Used:	191.62 GB
DFS Remaining:	139.77 GB (42.18%)
Block Pool Used:	150 B (0%)

HDFS Datanode information UI

http://localhost:9864/datanode.html

The website looks like the following screenshot:



DataNode on raymond-pc:9866

Cluster ID:	CID-03f690cf-d0e8-44fb-b65f-236f56360b13
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842

Block Pools

Namenode Address	Block Pool ID	Actor State	Last Heartbeat	Last Block Report	Last Block Report Size (Max Size)
0.0.0.0:19000	BP-1470916717-192.168.56.1-1579437638169	RUNNING	2s	7 minutes	0 B (64 MB)

Volume Information

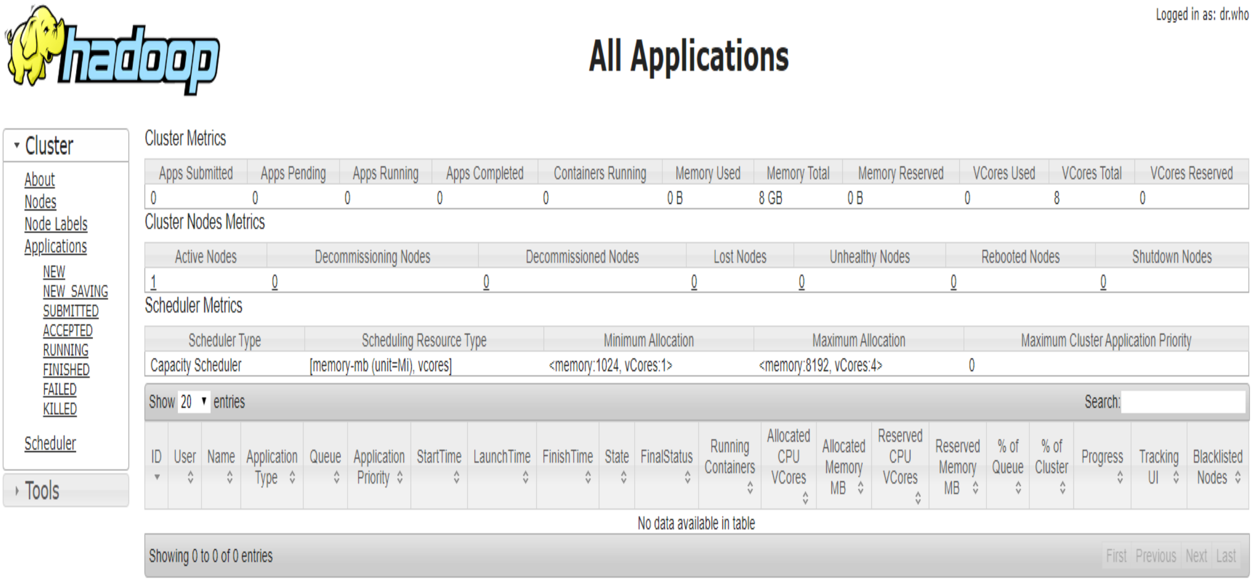
Directory	StorageType	Capacity Used	Capacity Left	Capacity Reserved	Reserved Space for Replicas	Blocks
F:\big-data\data\dfs\data	DISK	150 B	139.77 GB	0 B	0 B	0

Hadoop, 2019.

YARN resource manager UI

http://localhost:8088

The website looks like the following screenshot:



Logged in as: dr.who

All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
0	0	0	0	0	0 B	8 GB	0 B	0	8	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
1	0	0	0	0	0	0

Scheduler Metrics

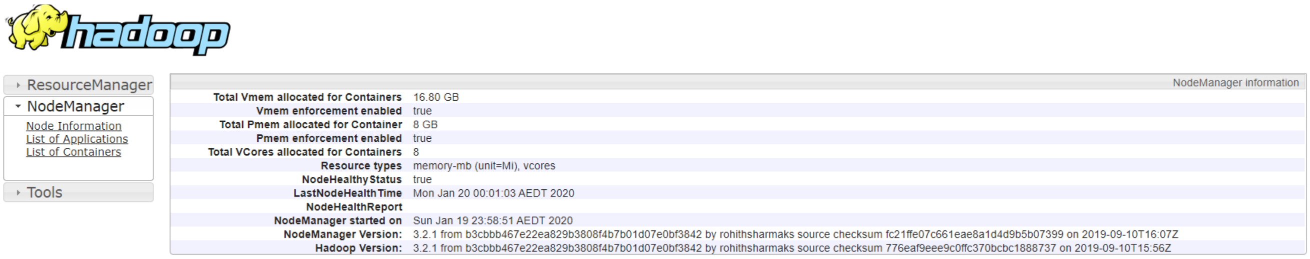
Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries Search:

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Reserved CPU VCores	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
No data available in table																				

Showing 0 to 0 of 0 entries First Previous Next Last

Through Resource Manager, you can also navigate to any Node Manager:



NodeManager information

Total Vmem allocated for Containers	16.80 GB
Vmem enforcement enabled	true
Total Pmem allocated for Container	8 GB
Pmem enforcement enabled	true
Total VCores allocated for Containers	8
Resource types	memory-mb (unit=Mi), vcores
NodeHealthyStatus	true
LastNodeHealthTime	Mon Jan 20 00:01:03 AEDT 2020
NodeHealthReport	
NodeManager started on	Sun Jan 19 23:58:51 AEDT 2020
NodeManager Version:	3.2.1 from b3cbbb467e22ea829b3808f4b7b01d07e0bf3842 by rohithsharmaks source checksum fc21ffe07c661eae8a1d4d9b5b07399 on 2019-09-10T16:07Z
Hadoop Version:	3.2.1 from b3cbbb467e22ea829b3808f4b7b01d07e0bf3842 by rohithsharmaks source checksum 776eaf9eee9c0ffc370bcbc1888737 on 2019-09-10T15:56Z

Step 11 - Shutdown YARN & HDFS daemons

You don't need to keep the services running all the time. You can stop them by running the following commands one by one:

```
%HADOOP_HOME%\sbin\stop-yarn.cmd
%HADOOP_HOME%\sbin\stop-dfs.cmd
```

Congratulations! You've successfully completed the installation of Hadoop 3.2.1 on Windows 10.