

Install Apache Hive 3.1.3 on Windows 10 and Configure 3 Metastore Modes

Author: Sri Adilakshmi M

Table of Contents

1. Overview:	4
2. Prerequisites:	5
3. Download Hive Binaries:	5
4. Set up Environment Variables:	9
5. Verify Hive Installation:	13
6. Start Hadoop Services:	13
7. Configure Embedded Derby Metastore:	16
7.1. Initialize Hive Metastore:	16
7.2. Start Hive CLI:	17
7.3. Common Hive CLI Issues:	18
7.4. Hive Interactive Shell Commands:	19
7.5. Run Queries on Hive CLI:	22
7.6. Beeline CLI:	29
8. Configure Local Derby Metastore:	31
8.1. Install Apache Derby:	31
8.2. Set up Environment Variables:	33
8.3. Start Derby Network Server:	36
8.4. Initialize Local Metastore:	37
8.5. Configure Hive Site:	37
8.6. Copy Derby Libraries:	38
8.7. Start Hive CLI:	39
8.8. Common Hive CLI Issues:	39
8.9. Run Queries on Hive CLI:	42
8.10. Beeline in Embedded Mode:	46
8.11. Start HiveServer2 Service:	48
8.12. Beeline in Remote Mode:	50
9. Configure Remote MySQL Metastore:	52
9.1. Install MySQL Server:	52

9.2.	Create Metastore DB in MySQL:	53
9.3.	Download MySQL JDBC Driver:	54
9.4.	Configure Hive Site File:	55
9.5.	Initialize Metastore DB:	56
9.6.	Verify Metastore in MySQL:	57
9.7.	Start Hive Metastore Service:	58
9.8.	Run Queries in Hive CLI:	59
9.9.	Start HiveServer2 Service:	69
9.10.	Beeline CLI:	72
9.11.	Verify Metadata in MySQL:	73
10.	Hive Web UI:	75
11.	HCatalog and WebHCat:	75
11.1.	Create Symbolic Link for Cygwin:	76
11.2.	Setup Env variables for Cygwn:	76
11.3.	Start HCatalog CLI:	77
11.4.	Start WebHCat Server:	79

This document outlines the steps needed to install Apache Hive on Windows Operating system.

1. Overview:

Apache Hive was developed by Facebook and became an open-source ETL and data warehousing tool which is built on top of Hadoop for analyzing, querying and managing large datasets stored in HDFS. Hive uses **HQL** (Hive Query Language) as a processing engine that processes HDFS datasets such that queries executed from Hive are internally converted into MapReduce tasks for parallel computation and distribution of data.

The key components of Apache Hive include Hive CLI, Beeline CLI, HiveServer2, Hive Web Interface, Hive Driver, Hive Metastore, HCatalog and WebHCat.

Hive Metastore is a critical component of Hive because it is the central schema repository that stores Hive metadata including tables, columns, datatypes, data locations etc. created by Hive and this schema repository can be used by other data processing tools such as Spark, Pig etc.

Hive Metastore works in three different modes:

1. **Embedded Metastore:** In this mode, Hive Metastore service runs in the same JVM where Hive Driver service runs and it uses **Apache Derby** as metastore database that is stored on the local file system. This is the default metastore that comes with Hive installation and is used for testing purposes only. Only one embedded Derby database can access database files at any time so only one Hive session can be opened and if we try to start second Hive session, it errors out. To allow multiple Hive sessions, we can configure Derby to run as Network Server.
2. **Local Metastore:** In this mode, Hive Metastore and Hive Driver still run within the same JVM process but metastore service connects to a JDBC supported database such as MySQL that runs on a different JVM in the same machine or on different machine. Local metastore currently supports **Derby, MySQL, MSSQL, Oracle and Postgres** database systems only.
3. **Remote Metastore:** In this mode, Hive Metastore service runs in a different JVM but not in Hive Driver service JVM and metastore service connects to a remote database which could be MySQL, MSSQL, Oracle or Postgres. In Remote Metastore, Hive Client will make a connection to Hive Metastore using Thrift protocol, and Metastore server in turn communicates with the database and run queries.

This document provides instructions to install Hive 3.1.3 version on top of Hadoop 3.x.

2. Prerequisites:

The following prerequisites need to be installed before running Hive.

1. **Hadoop:** Before installing Hive, Hadoop cluster must have been installed and running. Go through [these steps](#) to install Hadoop on Windows operating system.
2. **File Archiver:** Any file archiver such as **7zip** or **WinRAR** is needed to unzip the downloaded Hive binaries. 7zip can be downloaded from the [7zip Downloads](#) website and WinRAR can be downloaded from the [RAR lab Downloads](#) website.
3. **Cygwin:** Since some Hive utilities are not compatible with Windows, we will need the Cygwin tool to run some Linux commands. You can go through [these steps](#) to install Cygwin.

3. Download Hive Binaries:

After installing prerequisites, download Hive 3.1.3 release from the [Apache Hive Downloads](#) mirror website.

The screenshot shows a web browser displaying the Apache Software Foundation's download page for Hive. The URL in the address bar is <https://www.apache.org/dyn/closer.cgi/hive/>. The page features the Apache Software Foundation logo at the top left. A navigation bar with links for Community, Projects, Downloads, Learn, Resources & Tools, About, and a search icon is visible. The main content area contains instructions for downloading Hive, including the suggested download location (<https://dlcdn.apache.org/hive/>), alternate download locations, and instructions for verifying file integrity using PGP signatures or hashes. Sections for HTTP download, backup sites, and verify integrity are present.

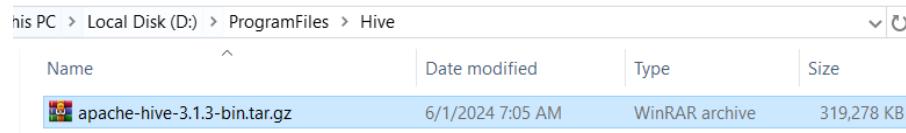
Go to the [suggested location](#) for download and click on [hive-3.1.3/](#) from where you need to download the binary file named apache-hive-3.1.3-bin.tar.gz file which gets downloaded to your **Downloads** folder.

Index of /hive/hive-3.1.3

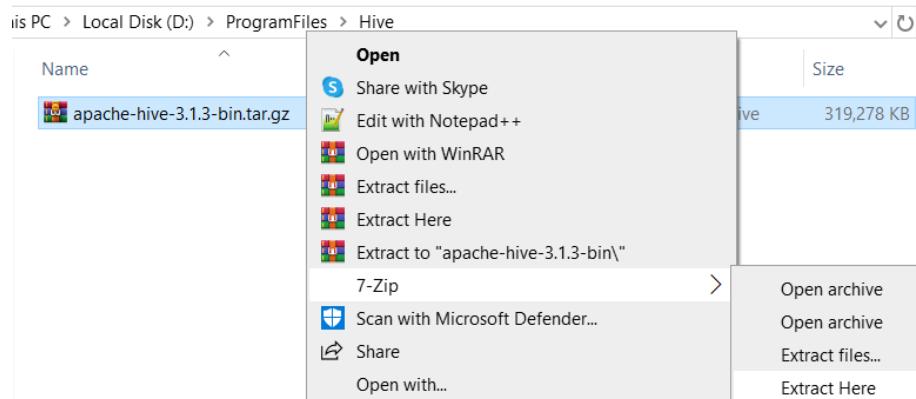
Name	Last modified	Size	Description
Parent Directory	-		
apache-hive-3.1.3-bin.tar.gz	2022-04-08 17:42	312M	
apache-hive-3.1.3-bin.tar.gz.asc	2022-04-08 17:42	488	
apache-hive-3.1.3-bin.tar.gz.sha256	2022-04-08 17:42	95	
apache-hive-3.1.3-src.tar.gz	2022-04-08 17:42	25M	
apache-hive-3.1.3-src.tar.gz.asc	2022-04-08 17:42	488	
apache-hive-3.1.3-src.tar.gz.sha256	2022-04-08 17:42	95	

After the binary file is downloaded, unpack it using any file archiver (7zip or WinRAR) utility as below:

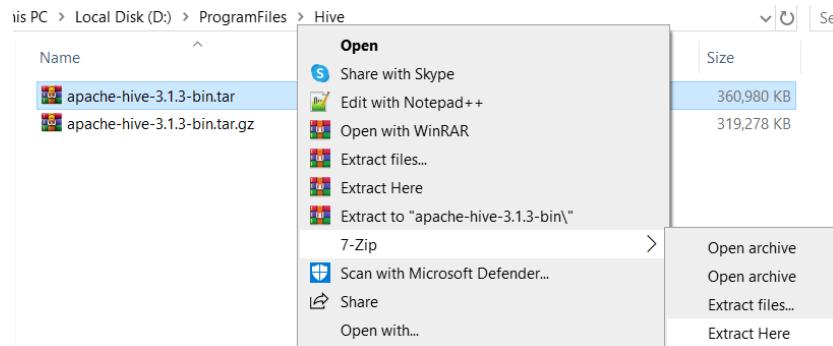
- Choose the installation directory in your machine and copy apache-hive-3.1.3-bin.tar.gz file to that directory. For example, I am choosing my Hive installation directory as D:\ProgramFiles\Hive.



- Right click on the file and choose **7-Zip -> Extract Here** option which extracts a new packed file apache-hive-3.1.3-bin.tar.



- Next, unpack apache-hive-3.1.3-bin.tar file using 7zip utility.



- The tar file extraction may take few minutes to finish. After finishing, you see a folder named apache-hive-3.1.3-bin which consists of Hive binaries and libraries.

Name	Date modified	Type	Size
bin	6/1/2024 7:14 AM	File folder	
binary-package-licenses	6/1/2024 7:12 AM	File folder	
conf	6/1/2024 8:56 AM	File folder	
examples	6/1/2024 7:12 AM	File folder	
hcatalog	6/1/2024 7:12 AM	File folder	
jdbc	6/1/2024 7:12 AM	File folder	
lib	6/1/2024 12:12 PM	File folder	
scripts	6/1/2024 7:12 AM	File folder	
LICENSE	3/29/2022 1:09 AM	File	21 KB
NOTICE	3/29/2022 1:09 AM	File	1 KB
RELEASE_NOTES.txt	3/29/2022 1:15 AM	Text Document	1 KB

Note:

By default, Apache Hive is built to run on Linux Operating system. To make it running on Windows OS, we should use **Cygwin** utility to execute Linux commands from Windows. However, we can run some Hive utilities directly on Windows without Cygwin by downloading .cmd files from the [HadiFadl GitHub repository](#). These .cmd files allow you to start Hive from Windows command line but you will not be able to run some utilities such as schematool, metastore, etc. directly in which case Cygwin is required.

Alternatively, I would suggest you to download .cmd files from [my GitHub repository](#) for the corresponding Hive version. For this installation, download files from [hive-3.1.3](#) and paste under the exact folder structure where ever Hive is installed. This makes you to start Hive as well as run other utilities including schematool, metastore, metatool, etc. from Windows itself.

- Copy .cmd files in [hive-3.1.3/bin](#) folder to D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\bin folder.

Name	Date modified	Type	Size
ext	6/1/2024 7:12 AM	File folder	
beeline	10/24/2019 8:47 AM	File	1 KB
beeline.cmd	6/1/2024 1:43 AM	Windows Comma...	2 KB
derbyserver.cmd	6/1/2024 1:43 AM	Windows Comma...	2 KB
hive	3/29/2022 1:09 AM	File	10 KB
hive.cmd	6/1/2024 1:43 AM	Windows Comma...	8 KB
hive-config.cmd	6/1/2024 1:43 AM	Windows Comma...	2 KB
hive-config.sh	2/28/2022 4:01 AM	SH File	3 KB
hiveserver2	10/24/2019 8:47 AM	File	1 KB
hiveserver2.cmd	6/1/2024 1:43 AM	Windows Comma...	2 KB
hqlsql	10/24/2019 8:47 AM	File	1 KB
hqlsql.cmd	6/1/2024 1:43 AM	Windows Comma...	2 KB
init-hive-dfs.cmd	6/1/2024 1:43 AM	Windows Comma...	3 KB
init-hive-dfs.sh	10/24/2019 8:47 AM	SH File	3 KB
metatool	10/24/2019 8:47 AM	File	1 KB
metatool.cmd	6/1/2024 1:43 AM	Windows Comma...	2 KB
schematool	10/24/2019 8:47 AM	File	1 KB
schematool.cmd	6/1/2024 1:43 AM	Windows Comma...	2 KB

- Copy .cmd files in [hive-3.1.3/bin/ext](#) folder to D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\bin\ext folder.

Name	Date modified	Type	Size
util	10/24/2019 8:47 AM	File folder	
beeline.cmd	6/1/2024 1:43 AM	Windows Comma...	3 KB
beeline.sh	3/29/2022 1:09 AM	SH File	2 KB
cleardanglingscratchdir.cmd	6/1/2024 1:43 AM	Windows Comma...	2 KB
cleardanglingscratchdir.sh	10/24/2019 8:47 AM	SH File	2 KB
cli.cmd	6/1/2024 1:43 AM	Windows Comma...	2 KB
cli.sh	10/24/2019 8:47 AM	SH File	2 KB
debug.cmd	6/1/2024 1:43 AM	Windows Comma...	4 KB
debug.sh	3/29/2022 1:09 AM	SH File	4 KB
fixacidkeyindex.cmd	6/1/2024 1:43 AM	Windows Comma...	2 KB
fixacidkeyindex.sh	9/5/2020 12:39 AM	SH File	2 KB
help.cmd	6/1/2024 1:43 AM	Windows Comma...	2 KB
help.sh	10/24/2019 8:47 AM	SH File	2 KB
hiveburnclient.cmd	6/1/2024 1:43 AM	Windows Comma...	2 KB
hiveburnclient.sh	10/24/2019 8:47 AM	SH File	2 KB
hiveserver2.cmd	6/1/2024 1:43 AM	Windows Comma...	4 KB
hiveserver2.sh	9/5/2020 12:39 AM	SH File	2 KB
hqlsql.cmd	6/1/2024 1:43 AM	Windows Comma...	3 KB
hqlsql.sh	10/24/2019 8:47 AM	SH File	2 KB
jar.cmd	6/1/2024 1:43 AM	Windows Comma...	2 KB

- Copy .cmd files in [hive-3.1.3/bin/ext/util](#) folder to D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\bin\ext folder.

Name	Date modified	Type	Size
execHiveCmd.cmd	6/1/2024 1:43 AM	Windows Comma...	2 KB
execHiveCmd.sh	10/24/2019 8:47 AM	SH File	2 KB

4. Set up Environment Variables:

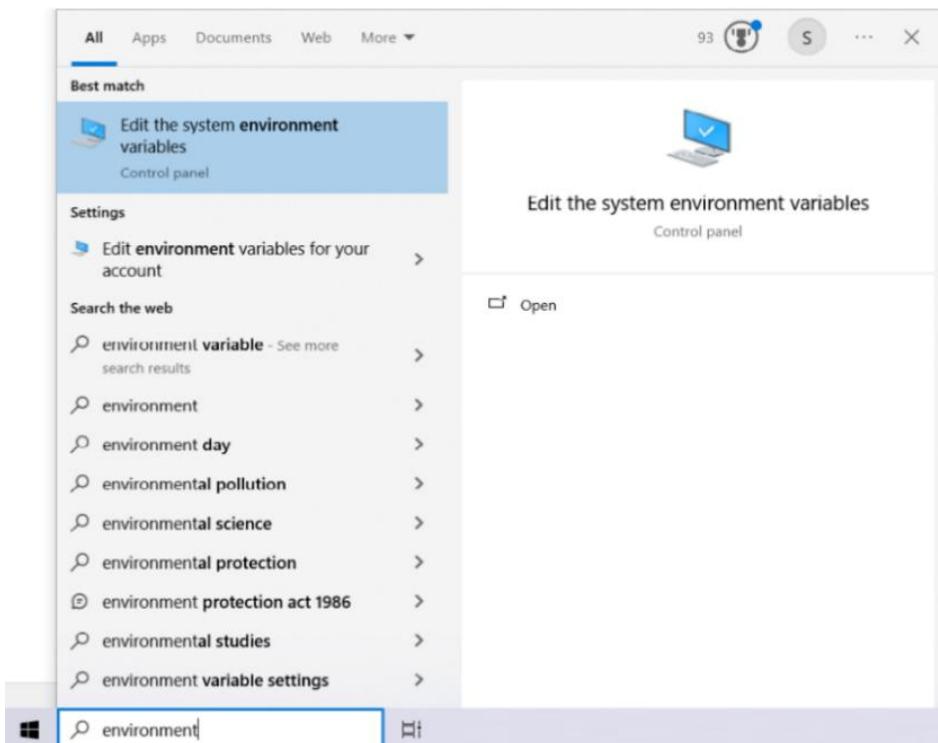
After installing Hadoop prerequisite and Hive binaries, we should configure two environment variables defining Hive installation path.

- **HIVE_HOME**: This is the Hive installation directory path in the machine (*in our case, it is D:\Programs\Hive\apache-hive-3.1.3-bin*)
- **HADOOP_USER_CLASSPATH_FIRST**: Set this variable value to `true` for Hive to use Hadoop user Class path first. This ensures `log4j2.x` and `jline` jars are loaded ahead of the jars pulled by Hadoop.

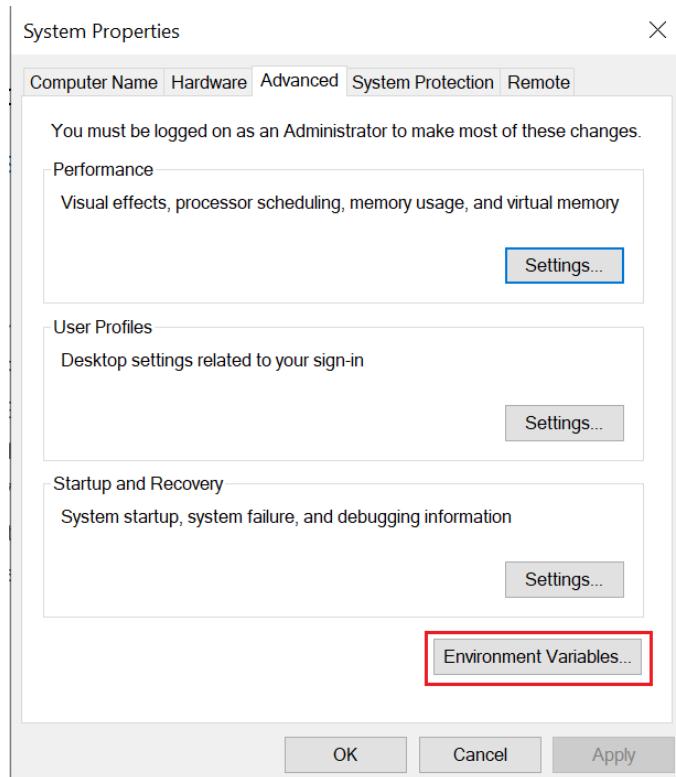
These variables need to be added to either **User environment variables** or **System environment variables** depending on Hive configuration needed **for a single user** or **for multiple users**.

In this tutorial, we will add User environment variables since we are configuring Hive for a single user. If you would like to configure Hive for multiple users, then define System environment variables.

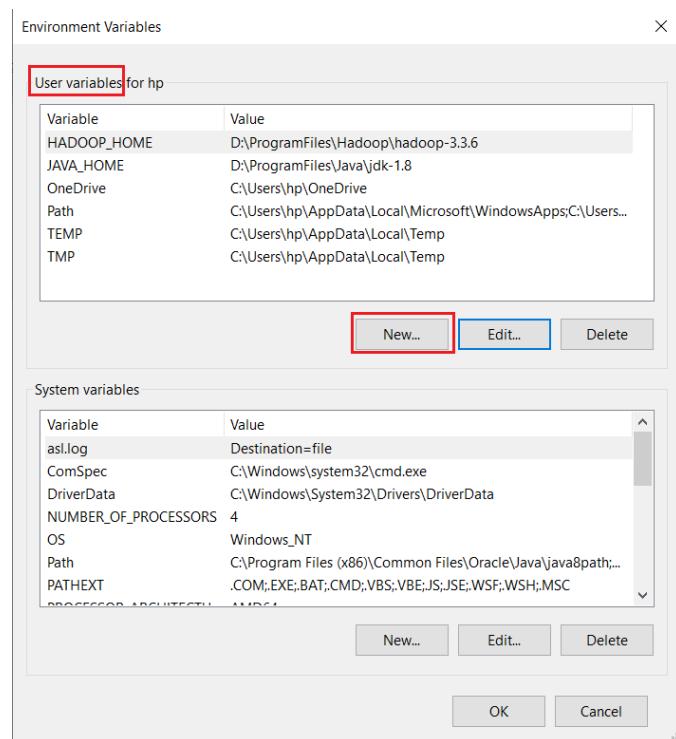
In the Windows search bar, start typing “environment variables” and select the first match which opens up **System Properties** dialog.



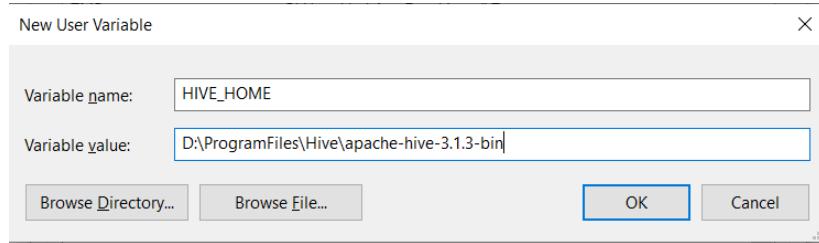
On the **System Properties** window, press **Environment Variables** button.



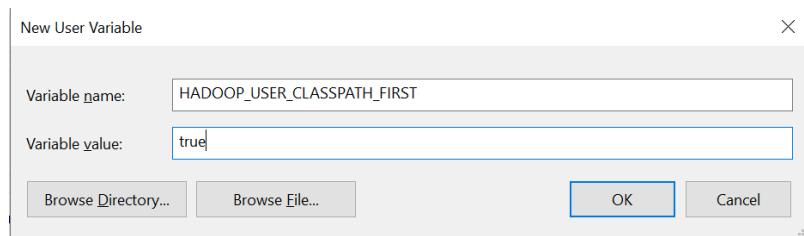
In the **Environment Variables** dialog, press **New** under **User variables** section.



Add **HIVE_HOME** variable and press OK.

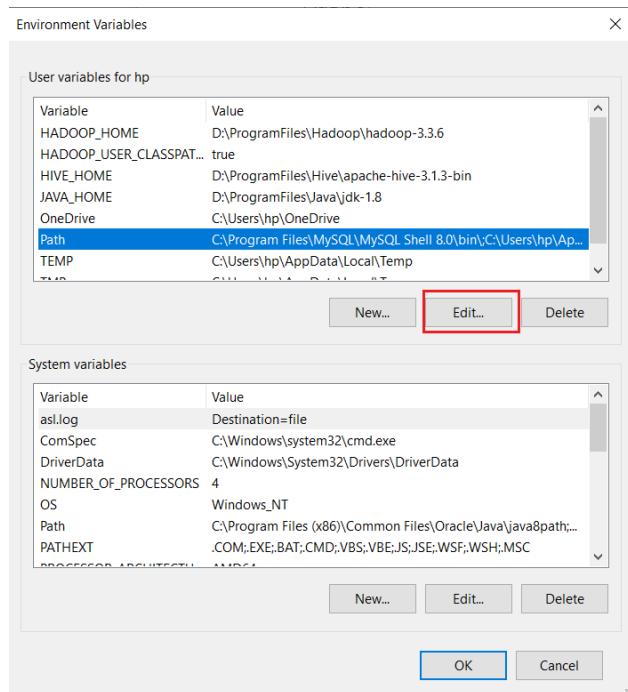


Click on **New** again and add **HADOOP_USER_CLASSPATH_FIRST** variable to **true** and press **OK**.

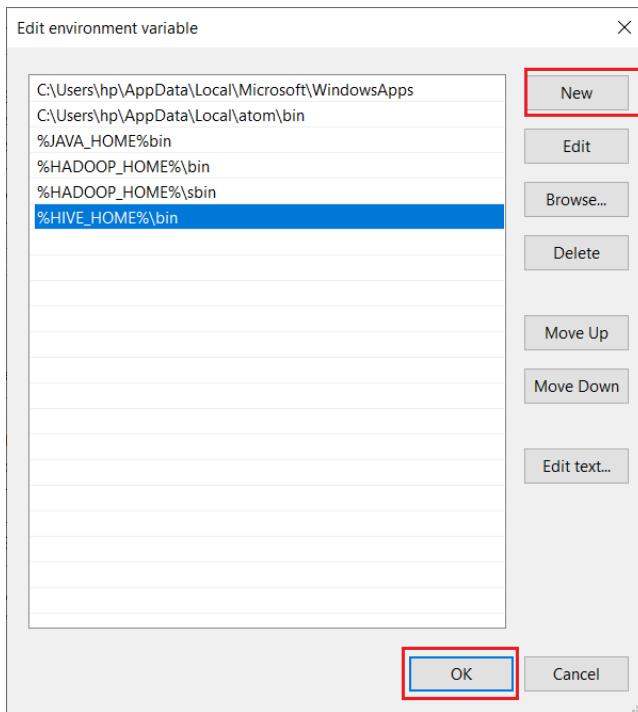


Now, we will update PATH variable to add Hive binary path

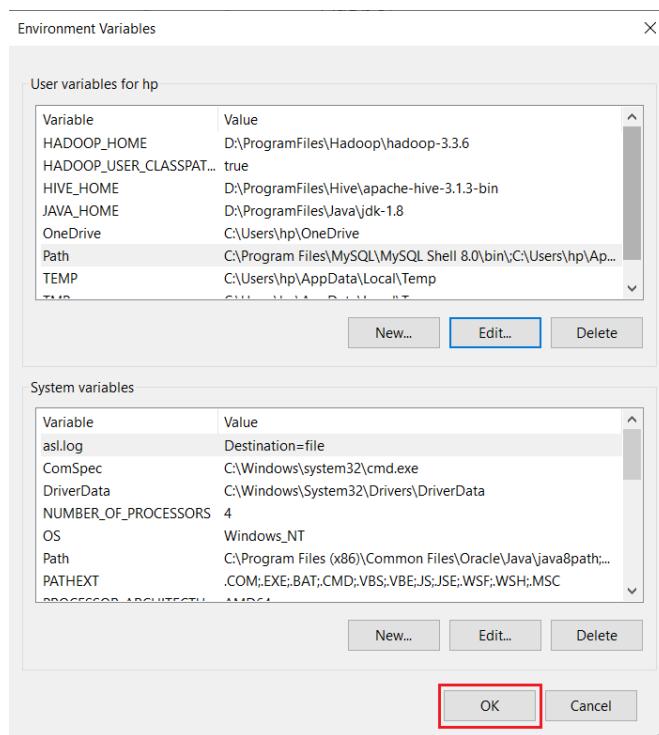
Select PATH variable under **User variables** and press **Edit** button.



Press **New** and add **%HIVE_HOME%\bin** path and press **OK**.



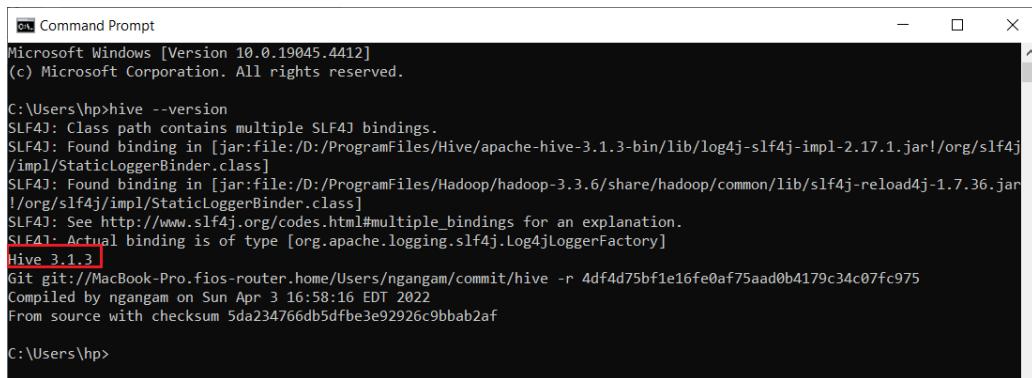
Press **OK** to apply the environment variable changes and close window.



5. Verify Hive Installation:

Open **Windows PowerShell** or **Command Prompt** and verify if Hive is installed properly by running the following command:

```
hive --version
```



```
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>hive --version
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]
Hive 3.1.3
Git git://MacBook-Pro.fios-router.home/Users/ngangam/commit/hive -r 4df4d75bf1e16fe0af75aad0b4179c34c07fc975
Compiled by ngangam on Sun Apr 3 16:58:16 EDT 2022
From source with checksum 5da234766db5dfbe3e92926c9bab2af

C:\Users\hp>
```

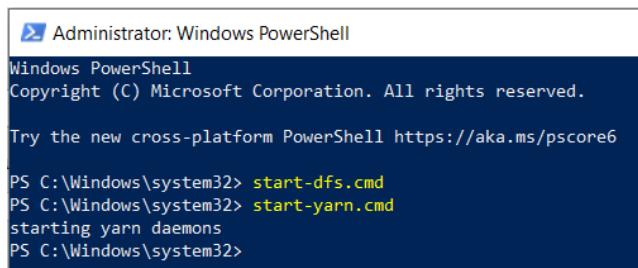
This may take couple of minutes to complete and displays **Hive 3.1.3** version installed.

6. Start Hadoop Services:

Before starting Hive, Hadoop services must be running since Hive runs on top of HDFS.

Open **Windows Command Prompt** or **Windows PowerShell** in **Administrator** mode and run the following commands to start Hadoop services.

```
start-dfs.cmd
start-yarn.cmd
```



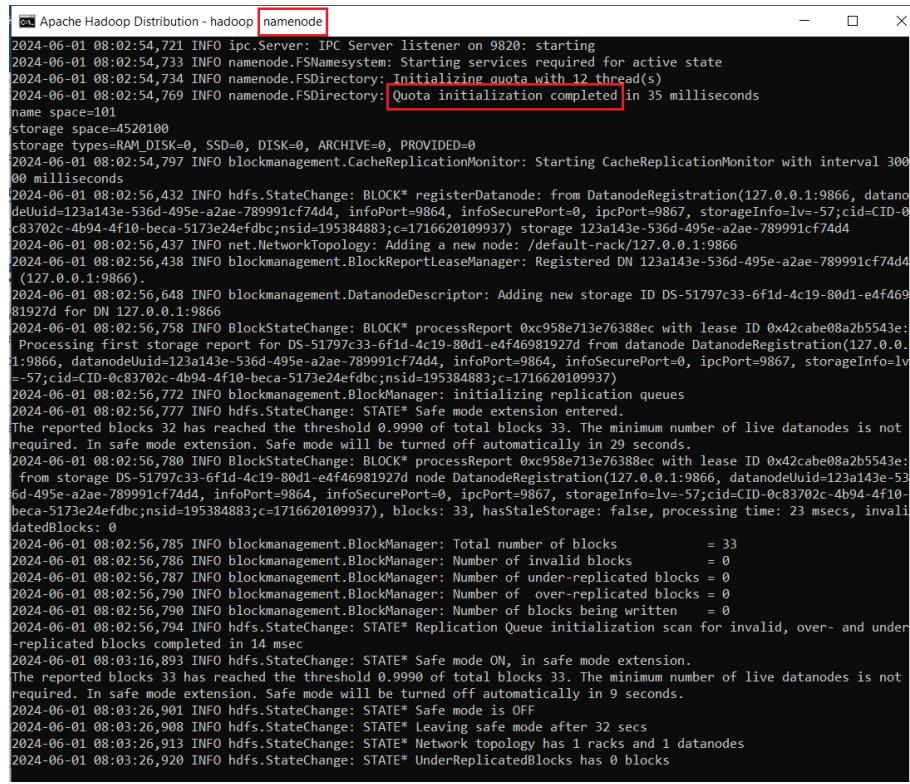
```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> start-dfs.cmd
PS C:\Windows\system32> start-yarn.cmd
starting yarn daemons
PS C:\Windows\system32>
```

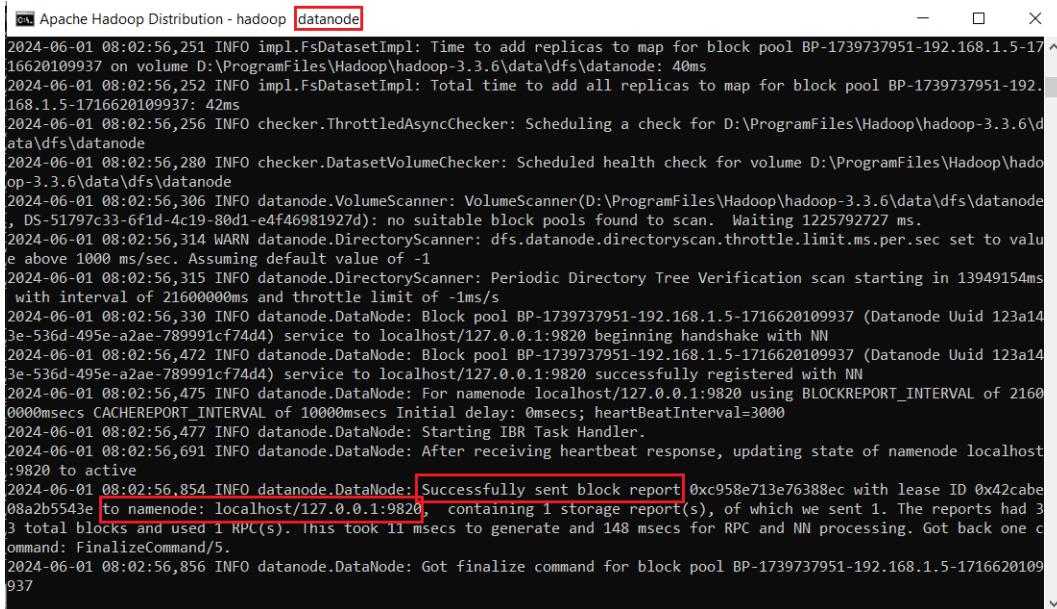
After executing the above commands, we can see four Windows command prompts opened for **namenode**, **datanode**, **resourcemanager** and **nodemanager** as below:

Wait for namenode service to say “Quota initialization completed”.



```
Apache Hadoop Distribution - hadoop namenode
2024-06-01 08:02:54,721 INFO ipc.Server: IPC Server listener on 9820: starting
2024-06-01 08:02:54,733 INFO namenode.FSNamesystem: Starting services required for active state
2024-06-01 08:02:54,734 INFO namenode.FSDirectory: Initializing quota with 12 thread(s)
2024-06-01 08:02:54,769 INFO namenode.FSDirectory: Quota initialization completed in 35 milliseconds
name space=101
storage space=4520100
storage types=RAM_DISK=0, SSD=0, DISK=0, ARCHIVE=0, PROVIDED=0
2024-06-01 08:02:54,797 INFO blockmanagement.CacheReplicationMonitor: Starting CacheReplicationMonitor with interval 300
00 milliseconds
2024-06-01 08:02:56,432 INFO hdfs.StateChange: BLOCK* registerDatanode: from DatanodeRegistration(127.0.0.1:9866, datanodeUuid=123a143e-536d-495e-a2ae-789991cf74d4, infoPort=9864, infoSecurePort=0, ipcPort=9867, storageInfo=lv=-57;cid=CID-0
c83702c-4b94-4f10-beca-5173e24efdbc;nsid=195384883;c=1716620109937) storage 123a143e-536d-495e-a2ae-789991cf74d4
2024-06-01 08:02:56,437 INFO net.NetworkTopology: Adding a new node: /default-rack/127.0.0.1:9866
2024-06-01 08:02:56,438 INFO blockmanagement.BlockReportLeaseManager: Registered DN 123a143e-536d-495e-a2ae-789991cf74d4
(127.0.0.1:9866).
2024-06-01 08:02:56,648 INFO blockmanagement.DatanodeDescriptor: Adding new storage ID DS-51797c33-6f1d-4c19-80d1-e4f469
81927d for DN 127.0.0.1:9866
2024-06-01 08:02:56,758 INFO BlockStateChange: BLOCK* processReport 0xc958e713e76388ec with lease ID 0x42cab08a2b5543e:
Processing first storage report for DS-51797c33-6f1d-4c19-80d1-e4f46981927d from datanodeRegistration(127.0.0.
1:9866, datanodeUuid=123a143e-536d-495e-a2ae-789991cf74d4, infoPort=9864, infoSecurePort=0, ipcPort=9867, storageInfo=lv=
-57;cid=CID-0c83702c-4b94-4f10-beca-5173e24efdbc;nsid=195384883;c=1716620109937)
2024-06-01 08:02:56,772 INFO blockmanagement.BlockManager: initializing replication queues
2024-06-01 08:02:56,777 INFO hdfs.StateChange: STATE* Safe mode extension entered.
The reported blocks 32 has reached the threshold 0.9990 of total blocks 33. The minimum number of live datanodes is not
required. In safe mode extension. Safe mode will be turned off automatically in 29 seconds.
2024-06-01 08:02:56,780 INFO BlockStateChange: BLOCK* processReport 0xc958e713e76388ec with lease ID 0x42cab08a2b5543e:
from storage DS-51797c33-6f1d-4c19-80d1-e4f46981927d node DatanodeRegistration(127.0.0.1:9866, datanodeUuid=123a143e-53
6d-495e-a2ae-789991cf74d4, infoPort=9864, infoSecurePort=0, ipcPort=9867, storageInfo=lv=-57;cid=CID-0c83702c-4b94-4f10-
beca-5173e24efdbc;nsid=195384883;c=1716620109937), blocks: 33, hasStaleStorage: false, processing time: 23 msecs, invalid
atedBlocks: 0
2024-06-01 08:02:56,785 INFO blockmanagement.BlockManager: Total number of blocks = 33
2024-06-01 08:02:56,786 INFO blockmanagement.BlockManager: Number of invalid blocks = 0
2024-06-01 08:02:56,787 INFO blockmanagement.BlockManager: Number of under-replicated blocks = 0
2024-06-01 08:02:56,790 INFO blockmanagement.BlockManager: Number of over-replicated blocks = 0
2024-06-01 08:02:56,793 INFO blockmanagement.BlockManager: Number of blocks being written = 0
2024-06-01 08:02:56,794 INFO hdfs.StateChange: STATE* Replication Queue initialization scan for invalid, over- and under-
replicated blocks completed in 14 msec
2024-06-01 08:03:16,893 INFO hdfs.StateChange: STATE* Safe mode ON, in safe mode extension.
The reported blocks 33 has reached the threshold 0.9990 of total blocks 33. The minimum number of live datanodes is not
required. In safe mode extension. Safe mode will be turned off automatically in 9 seconds.
2024-06-01 08:03:26,901 INFO hdfs.StateChange: STATE* Safe mode is OFF
2024-06-01 08:03:26,908 INFO hdfs.StateChange: STATE* Leaving safe mode after 32 secs
2024-06-01 08:03:26,913 INFO hdfs.StateChange: STATE* Network topology has 1 racks and 1 datanodes
2024-06-01 08:03:26,920 INFO hdfs.StateChange: STATE* UnderReplicatedBlocks has 0 blocks
```

Wait for datanode service to say “Successfully sent block report to namenode: localhost/127.0.0.1:9820”.



```
Apache Hadoop Distribution - hadoop datanode
2024-06-01 08:02:56,251 INFO impl.FsDatasetImpl: Time to add replicas to map for block pool BP-1739737951-192.168.1.5-17
16620109937 on volume D:\ProgramFiles\Hadoop\hadoop-3.3.6\data\dfs\datanode: 40ms
2024-06-01 08:02:56,252 INFO impl.FsDatasetImpl: Total time to add all replicas to map for block pool BP-1739737951-192.
168.1.5-1716620109937: 42ms
2024-06-01 08:02:56,256 INFO checker.ThrottledAsyncChecker: Scheduling a check for D:\ProgramFiles\Hadoop\hadoop-3.3.6\d
ata\dfs\datanode
2024-06-01 08:02:56,280 INFO checker.DatasetVolumeChecker: Scheduled health check for volume D:\ProgramFiles\Hadoop\had
oop-3.3.6\data\dfs\datanode
2024-06-01 08:02:56,306 INFO datanode.VolumeScanner: VolumeScanner(D:\ProgramFiles\Hadoop\hadoop-3.3.6\data\dfs\datanod
e, DS-51797c33-6f1d-4c19-80d1-e4f46981927d): no suitable block pools found to scan. Waiting 1225792727 ms.
2024-06-01 08:02:56,314 WARN datanode.DirectoryScanner: dfs.datanode.directoryscan.throttle.limit.ms.per.sec set to val
ue above 1000 ms/sec. Assuming default value of -1
2024-06-01 08:02:56,315 INFO datanode.DirectoryScanner: Periodic Directory Tree Verification scan starting in 13949154ms
with interval of 2160000ms and throttle limit of -1ms/s
2024-06-01 08:02:56,330 INFO datanode.DataNode: Block pool BP-1739737951-192.168.1.5-1716620109937 (Datanode Uuid 123a14
3e-536d-495e-a2ae-789991cf74d4) service to localhost/127.0.0.1:9820 beginning handshake with NN
2024-06-01 08:02:56,472 INFO datanode.DataNode: Block pool BP-1739737951-192.168.1.5-1716620109937 (Datanode Uuid 123a14
3e-536d-495e-a2ae-789991cf74d4) service to localhost/127.0.0.1:9820 successfully registered with NN
2024-06-01 08:02:56,475 INFO datanode.DataNode: For namenode localhost/127.0.0.1:9820 using BLOCKREPORT_INTERVAL of 2160
000msecs CACHEREPORT_INTERVAL of 10000msecs Initial delay: 0msecs; heartBeatInterval=3000
2024-06-01 08:02:56,477 INFO datanode.DataNode: Starting IBB Task Handler.
2024-06-01 08:02:56,691 INFO datanode.DataNode: After receiving heartbeat response, updating state of namenode localhost
:9820 to active
2024-06-01 08:02:56,854 INFO datanode.DataNode: Successfully sent block report 0xc958e713e76388ec with lease ID 0x42cab0
8a2b5543e to namenode: localhost/127.0.0.1:9820, containing 1 storage report(s), of which we sent 1. The reports had 3
total blocks and used 1 RPC(s). This took 11 msecs to generate and 148 msecs for RPC and NN processing. Got back one c
ommand: FinalizeCommand/5.
2024-06-01 08:02:56,856 INFO datanode.DataNode: Got finalize command for block pool BP-1739737951-192.168.1.5-1716620109
937
```

Wait for `resourcemanager` service to say “*Transitioned to active state*”.

```
Apache Hadoop Distribution - yarn resourcemanager
2024-06-01 08:05:40,229 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.server.api.ResourceTrackerPB to the server
2024-06-01 08:05:40,279 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue, queueCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false.
2024-06-01 08:05:40,284 INFO ipc.Server: IPC Server Responder: starting
2024-06-01 08:05:40,287 INFO ipc.Server: IPC Server listener on 8031: starting
2024-06-01 08:05:40,355 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2024-06-01 08:05:40,371 INFO ipc.Server: Listener at 0.0.0.0:8030
2024-06-01 08:05:40,416 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationMasterProtocolPB to the server
2024-06-01 08:05:40,510 INFO ipc.Server: Starting Socket Reader #1 for port 8030
2024-06-01 08:05:40,575 INFO ipc.Server: IPC Server listener on 8030: starting
2024-06-01 08:05:40,575 INFO ipc.Server: IPC Server Responder: starting
2024-06-01 08:05:40,808 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue, queueCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false.
2024-06-01 08:05:40,810 INFO ipc.Server: Listener at 0.0.0.0:8032
2024-06-01 08:05:40,814 INFO ipc.Server: Starting Socket Reader #1 for port 8032
2024-06-01 08:05:40,826 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationClientProtocolPB to the server
2024-06-01 08:05:40,829 INFO ipc.Server: IPC Server Responder: starting
2024-06-01 08:05:40,830 INFO ipc.Server: IPC Server listener on 8032: starting
2024-06-01 08:05:41,491 INFO webproxy.ProxyCA: Created Certificate for OU=YARN-161431da-0ba0-4968-bf02-1fd97c72acf8
2024-06-01 08:05:41,658 INFO recovery.RMStateStore: Storing CA Certificate and Private Key
2024-06-01 08:05:41,663 INFO resourcemanager.ResourceManager: Transitioned to active state
2024-06-01 08:05:41,792 INFO resourcemanager.ResourceTrackerService: NodeManager from node 192.168.56.1(cmPort: 45959 httpPort: 8042) registered with capability: <memory:8192, vCores:8>, assigned nodeId 192.168.56.1:45959
2024-06-01 08:05:41,798 INFO rmnode.RMNodeImpl: 192.168.56.1:45959 Node Transitioned from NEW to RUNNING
2024-06-01 08:05:41,870 INFO capacity.CapacityScheduler: Added node 192.168.56.1:45959 clusterResource: <memory:8192, vCores:8>
```

Wait for `nodemanager` service to say “*Registered with ResourceManager*”.

```
Apache Hadoop Distribution - yarn nodemanager
Jun 01, 2024 8:05:39 AM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory register
INFO: Registering org.apache.hadoop.yarn.server.nodemanager.webapp.JAXBContextResolver as a provider class
Jun 01, 2024 8:05:39 AM com.sun.jersey.server.impl.application.WebApplicationImpl _initiate
INFO: Initiating Jersey application, version 'Jersey: 1.19.4 05/24/2017 03:20 PM'
Jun 01, 2024 8:05:39 AM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.JAXBContextResolver to GuiceManagedComponentProvider with the scope "Singleton"
Jun 01, 2024 8:05:40 AM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.webapp.GenericExceptionHandler to GuiceManagedComponentProvider with the scope "Singleton"
Jun 01, 2024 8:05:40 AM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.NMWebServices to GuiceManagedComponentProvider with the scope "Singleton"
2024-06-01 08:05:41,093 INFO handler.ContextHandler: Started o.e.j.w.WebAppContext@524270b8{node/,file:///C:/Users/hp/appData/Local/Temp/jetty-0_0_0-8042-hadoop-yarn-common-3_3_6_jar-any-7367246008935846293/webapp/,AVAILABLE}{jar:file:///D:/ProgramFiles/Hadoop/hadoop-3.3.6/share/hadoop/yarn/hadoop-yarn-common-3.3.6.jar!/webapps/node}
2024-06-01 08:05:41,108 INFO server.AbstractConnector: Started ServerConnector@677dbd89{HTTP/1.1}{0.0.0.0:8042}
2024-06-01 08:05:41,109 INFO server.Server: Started @12794ms
2024-06-01 08:05:41,109 INFO webapp.WebApps: Web app node started at 8042
2024-06-01 08:05:41,110 INFO nodemanager.NodeStatusUpdaterImpl: Node ID assigned is : 192.168.56.1:45959
2024-06-01 08:05:41,113 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2024-06-01 08:05:41,126 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8031
2024-06-01 08:05:41,853 INFO security.NMContainerTokenSecretManager: Rolling master-key for container-tokens, got key with id 404372701
2024-06-01 08:05:41,855 INFO security.NMTokenSecretManagerInNM: Rolling master-key for container-tokens, got key with id -1703419895
2024-06-01 08:05:41,856 INFO nodemanager.NodeStatusUpdaterImpl: Registered with ResourceManager as 192.168.56.1:45959 with total resource of <memory:8192, vCores:8>
```

7. Configure Embedded Derby Metastore:

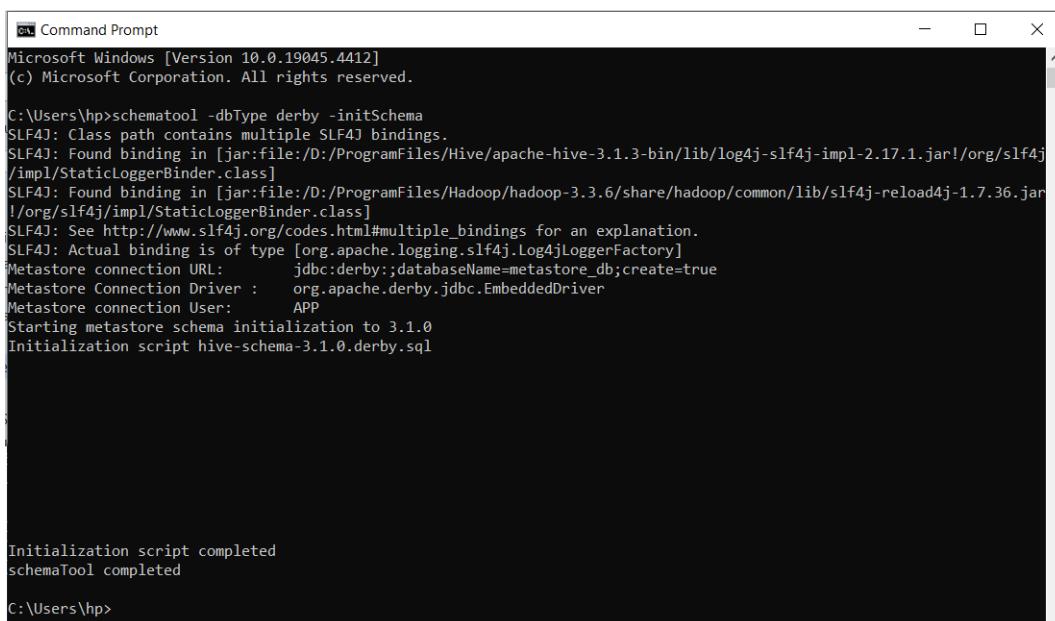
Hive by default runs with Embedded Metastore with Derby database on Hadoop File System.

7.1. Initialize Hive Metastore:

To start Hive, we need to initialize the Hive Metastore (`metastore_db`) using `schematool` utility provided by Hive.

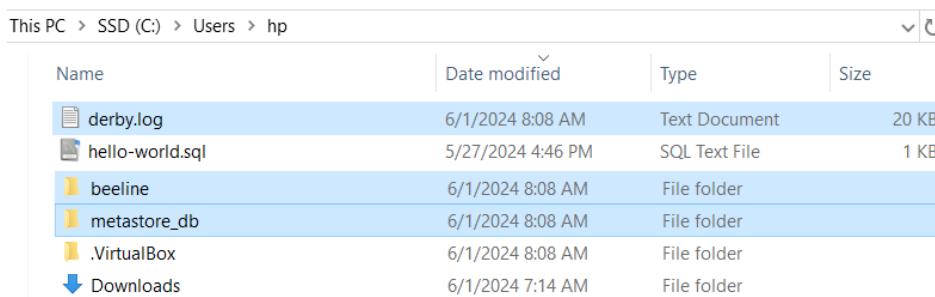
Execute the following command in Command Prompt or Windows PowerShell

```
schematool -dbType derby -initSchema
```



The screenshot shows a Microsoft Windows Command Prompt window titled "Command Prompt". The window displays the output of the `schematool -dbType derby -initSchema` command. The output includes several SLF4J binding logs, the connection URL (`jdbc:derby:;databaseName=metastore_db;create=true`), the connection driver (`org.apache.derby.jdbc.EmbeddedDriver`), and the user (`APP`). It also shows the initialization script being run (`hive-schema-3.1.0.derby.sql`). The process concludes with a message stating "Initialization script completed" and "schemaTool completed". The command prompt prompt is visible at the bottom as "C:\Users\hp>".

After executing the above command, it creates `metastore_db` folder, `beeline` folder and `derby.log` file in the location from where ever `schematool` utility got executed.



Name	Date modified	Type	Size
derby.log	6/1/2024 8:08 AM	Text Document	20 KB
hello-world.sql	5/27/2024 4:46 PM	SQL Text File	1 KB
beeline	6/1/2024 8:08 AM	File folder	
metastore_db	6/1/2024 8:08 AM	File folder	
.VirtualBox	6/1/2024 8:08 AM	File folder	
Downloads	6/1/2024 7:14 AM	File folder	

- metastore_db folder contains the database files of Hive metastore.

Name	Date modified	Type	Size
log	6/1/2024 8:08 AM	File folder	
seg0	6/1/2024 8:08 AM	File folder	
tmp	6/1/2024 8:08 AM	File folder	
db.lck	6/1/2024 8:08 AM	LCK File	1 KB
README_DO_NOT_TOUCH_FILES.txt	6/1/2024 8:08 AM	Text Document	1 KB
service.properties	6/1/2024 8:08 AM	PROPERTIES File	1 KB

- Open derby.log file to see that Hive has booted Apache Derby database of 10.14.1.0 version.

```

Sat Jun 01 08:08:49 IST 2024:
Booting Derby version The Apache Software Foundation - Apache Derby - 10.14.1.0 -
(1808820): instance a816c00e-018f-d1a9-fa61-0000095597f0
on database directory C:\Users\hp\metastore_db with class loader sun.misc.Launcher
$AppClassLoader@3ace8af7
Loaded from file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/derby-10.14.1.0.jar
java.vendor=Oracle Corporation
java.runtime.version=1.8.0_411-b09
user.dir=C:\Users\hp
os.name=Windows 10
os.arch=amd64
os.version=10.0
derby.system.home=null
Database Class Loader started - derby.database.classpath=''
```

7.2. Start Hive CLI:

Now, we can start `hive` command line interface using the below command

```
hive
```

To see the detailed logging of hive CLI, use the following command for starting hive.

```
hive --hiveconf hive.root.logger=console
```

```

C:\Users\hp>hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 2e3551da-e1e2-4411-b84a-3ba029c14567

Logging initialized using configuration in jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
Hive Session ID = d49d598d-f0a1-4f8a-9964-ff1553420b31
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive>

```

As soon as Hive is started, it creates a /tmp/hive folder on Hadoop File System.

Open another command prompt and execute this command to see /tmp/hive folder on HDFS

```
hadoop fs -ls /tmp
```

Hive uses /tmp HDFS directory for storing temp_space database which is available at /tmp/hive/<userid>/<temp_folder>/_temp_space.db

```

C:\Users\hp>hadoop fs -ls /tmp
Found 2 items
drwxr-xr-x - hp supergroup          0 2024-05-25 13:36 /tmp/hadoop-yarn
drwxrwxrwx - hp supergroup          0 2024-06-01 08:15 /tmp/hive

C:\Users\hp>hadoop fs -ls /tmp/hive
Found 1 items
drwx----- - hp supergroup          0 2024-06-01 08:16 /tmp/hive/hp

C:\Users\hp>hadoop fs -ls /tmp/hive/hp
Found 2 items
drwx----- - hp supergroup          0 2024-06-01 08:15 /tmp/hive/hp/2e3551da-e1e2-4411-b84a-3ba029c14567
drwx----- - hp supergroup          0 2024-06-01 08:16 /tmp/hive/hp/d49d598d-f0a1-4f8a-9964-ff1553420b31

C:\Users\hp>hadoop fs -ls /tmp/hive/hp/2e3551da-e1e2-4411-b84a-3ba029c14567
Found 1 items
drwx----- - hp supergroup          0 2024-06-01 08:15 /tmp/hive/hp/2e3551da-e1e2-4411-b84a-3ba029c14567/_temp_space.db

C:\Users\hp>hadoop fs -ls /tmp/hive/hp/2e3551da-e1e2-4411-b84a-3ba029c14567/_temp_space.db
C:\Users\hp>

```

7.3. Common Hive CLI Issues:

1. Unnecessary logging on Console:

You may see unnecessary logging to the console as below while launching `hive`.

```

C:\Users\lenovo\hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/Programs/Hadoop/hadoop-3.3.5/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/Programs/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.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.Reload4jLoggerFactory]
2024-05-07 07:41:04,852 INFO conf.HiveConf: Found configuration file file:/D:/Programs/Hive/apache-hive-3.1.3-bin/conf/hive-site.xml
Hive Session ID = fb8226bd-01cb-46d1-911a-a17ebf9517ab
2024-05-07 07:41:08,534 INFO SessionState: Hive Session ID = fb8226bd-01cb-46d1-911a-a17ebf9517ab

Logging initialized using configuration in jar:file:/D:/Programs/Hive/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
2024-05-07 07:41:09,113 INFO SessionState: Logging initialized using configuration in jar:file:/D:/Programs/Hive/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
2024-05-07 07:41:11,086 INFO session.SessionState: Created HDFS directory: /tmp/hive/lavanya/fb8226bd-01cb-46d1-911a-a17ebf9517ab
2024-05-07 07:41:11,097 INFO session.SessionState: Created local directory: C:/Users/lenovo/AppData/Local/Tmp/lavanya/fb8226bd-01cb-46d1-911a-a17ebf9517ab
2024-05-07 07:41:13,739 INFO conf.MetastoreConf: Unable to find config file metastore-site.xml
2024-05-07 07:41:13,739 INFO conf.MetastoreConf: Found configuration file null
2024-05-07 07:41:14,284 INFO DataNucleus.Persistence: Property datanucleus.cache.level2 unknown - will be ignored
2024-05-07 07:41:14,926 INFO hikari.HikariDataSource: HikariPool-1 - Starting...
2024-05-07 07:41:15,272 INFO pool.PoolBase: HikariPool-1 - Driver does not support get/set network timeout for connecti
ns. (Feature not implemented: getNetworkTimeout.)
2024-05-07 07:41:15,335 INFO hikari.HikariDataSource: HikariPool-1 - Start completed.
2024-05-07 07:41:15,457 INFO hikari.HikariDataSource: HikariPool-2 - Starting...
2024-05-07 07:41:15,463 INFO pool.PoolBase: HikariPool-2 - Starting...
2024-05-07 07:41:15,466 INFO hikari.HikariDataSource: HikariPool-2 - Start completed.
2024-05-07 07:41:15,667 INFO metastore.ObjectStore: Setting Metastore object pin classes with hive.metastore.cache.pinTypes="Table,StorageDescriptor,SerDeInfo,Partition,Database,type,FieldsSchema,Order"
2024-05-07 07:41:15,988 INFO metastore.MetaStoreDirectSql: Using direct SQL, underlying DB is DERBY
2024-05-07 07:41:15,912 INFO metastore.ObjectStore: Initialized ObjectStore
2024-05-07 07:41:16,432 WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
2024-05-07 07:41:16,437 WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
2024-05-07 07:41:16,440 WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
2024-05-07 07:41:16,443 WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
2024-05-07 07:41:16,446 WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
2024-05-07 07:41:16,448 WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
2024-05-07 07:41:20,760 WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored

```

To suppress the above logging, make sure the following are set:

- Set the environment variable `HADOOP_USER_CLASSPATH_FIRST=true`
- Open `hive-config.cmd` file in `HIVE_HOME\bin` directory and add the below line at the end of the file.

```
set HADOOP_CLIENT_OPTS=%HADOOP_CLIENT_OPTS% -Dlog4j.configurationFile=hive-log4j2.properties
```

Then, launch the `hive` CLI from the new command prompt.

Note: It is important to always open new command terminal after environment variables are updated otherwise environment variables will not come into effect.

7.4. Hive Interactive Shell Commands:

On the Hive interactive shell (`hive>` prompt), we can enter commands to perform various operations:

- Set the value of a particular configuration variable

For example:

```
set hive.cli.print.header=true;
```

- Print a list of configuration variables overridden by Hive

```
set;
```

- Print all Hadoop and Hive configuration variables

```
set -v;
```

- Add resources such as files, jars, archives to list of resources in the distributed cache

```
add file <filepath>;
add jar <filepath>;
add archive <filepath>;
```

- List resources added to the distributed cache

```
list files;
list file <filename>;
list jars;
list jar <jarname>;
list archives;
list archive <archivename>;
```

- Delete resources from the distributed cache

```
delete file <filepath>;
delete jar <filepath>;
delete archive <filepath>;
```

- Execute a shell command from Hive interactive shell

```
! <shell command>;
```

Note: This works in Linux operating system only. In Windows, it throws error since it cannot identify a Linux command.

- Execute a `dfs` command from Hive interactive shell

```
dfs <dfs command>;
```

For example:

```
dfs -ls /tmp/hive;
```

Note: This is same as executing Hadoop `dfs` commands on Windows or Linux operating system directly.

```
hive> dfs -ls /tmp/hive;
Found 1 items
drwx-----  - hp supergroup          0 2024-06-01 08:16 /tmp/hive/hp
hive>
```

- Execute a Hive query script file inside interactive shell

```
source <scriptfilepath>;
```

For example:

```
source C:\Users\hp\hello-world.sql;
```

```
hive> source C:\Users\hp\hello-world.sql;
OK
Hello World!    Current Time is 2024-06-01 08:19:24.722 IST
Time taken: 6.63 seconds, Fetched: 1 row(s)
hive>
```

My hello-world.sql file consist of the following query:

```
select "Hello World!", "Current Time is", current_timestamp, "IST";
```

- Execute a Hive query inside interactive shell

For example:

```
select current_timestamp;
```

```
hive> select current_timestamp;
OK
2024-06-01 08:20:09.51
Time taken: 0.369 seconds, Fetched: 1 row(s)
hive>
```

- Exit out of Hive shell

```
exit;
```

or

```
quit;
```

We can also execute Hive queries in batch mode by passing command line arguments to Hive CLI.

- Execute a Hive query outside interactive shell

```
hive -e '<query>'
```

For example:

```
hive -e 'select emp_id, emp_name from employee'
```

- Execute a Hive query script file outside interactive shell

```
hive -f <script_file_path>
```

For example:

```
hive -f C:\Users\hp\hello-world.sql
```

- Dump Hive query output into a file in Silent mode

```
hive -S -e 'query' > output_file_path
```

For example:

```
hive -e 'select emp_id, emp_name from employee'
```

- Dump Hive query script file into a file in Silent mode

```
hive -S -f <script_file_path> > output_file_path
```

For example:

```
hive -S -f C:\Users\hp\hello-world.sql > C:\Users\hp\hive_output.txt
```

7.5. Run Queries on Hive CLI:

Open Hive CLI and run the following queries to create a hive metadata database, create table, load data and select data.

- Create a database in Hive metastore:

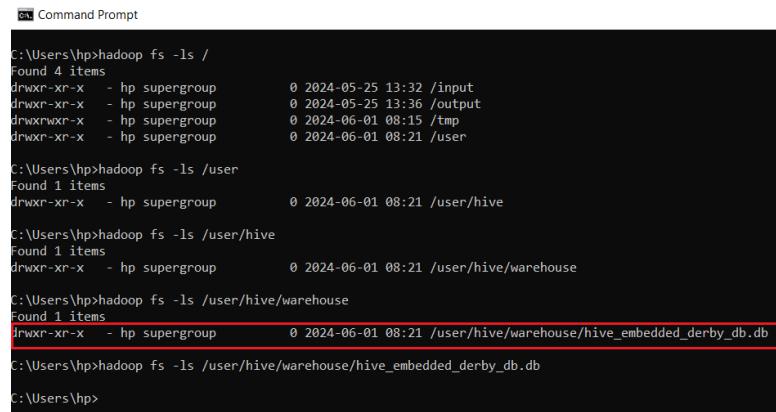
```
create database hive_embedded_derby_db;
show databases;
```

```
hive> create database hive_embedded_derby_db;
OK
Time taken: 0.191 seconds
hive> show databases;
OK
default
hive_embedded_derby_db
Time taken: 0.046 seconds, Fetched: 2 row(s)
hive>
```

As soon as the above query is executed, Hive creates `hive_embedded_derby_db` folder in `/user/hive/warehouse` location in Hadoop File System.

Open another Command Prompt and run this command:

```
hadoop fs -ls /user/hive/warehouse
```



```
C:\Users\hp>hadoop fs -ls /
Found 4 items
drwxr-xr-x  - hp supergroup          0 2024-05-25 13:32 /input
drwxr-xr-x  - hp supergroup          0 2024-05-25 13:36 /output
drwxrwxr-x  - hp supergroup          0 2024-06-01 08:15 /tmp
drwxr-xr-x  - hp supergroup          0 2024-06-01 08:21 /user

C:\Users\hp>hadoop fs -ls /user
Found 1 items
drwxr-xr-x  - hp supergroup          0 2024-06-01 08:21 /user/hive

C:\Users\hp>hadoop fs -ls /user/hive
Found 1 items
drwxr-xr-x  - hp supergroup          0 2024-06-01 08:21 /user/hive/warehouse

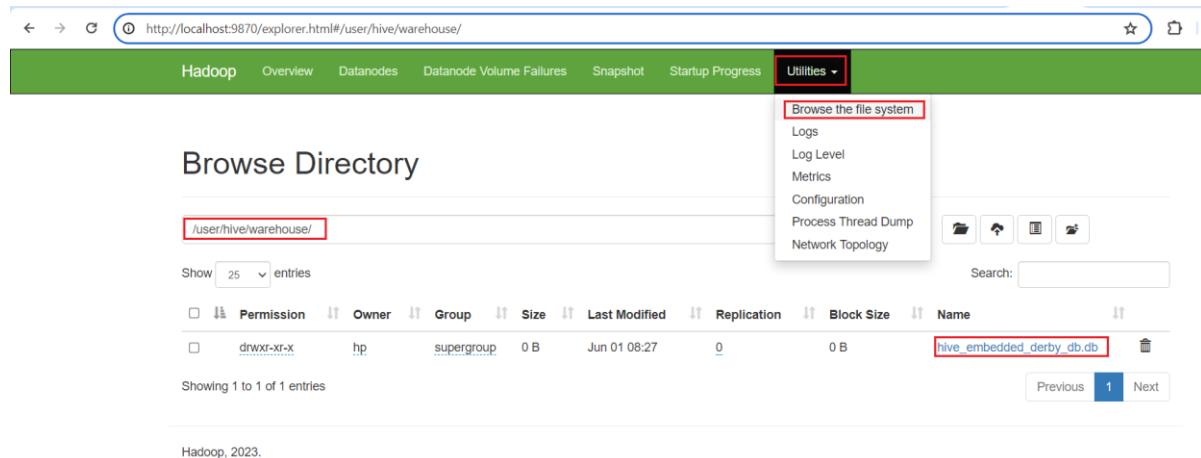
C:\Users\hp>hadoop fs -ls /user/hive/warehouse
Found 1 items
drwxr-xr-x  - hp supergroup          0 2024-06-01 08:21 /user/hive/warehouse/hive_embedded_derby_db.db

C:\Users\hp>hadoop fs -ls /user/hive/warehouse/hive_embedded_derby_db.db

C:\Users\hp>
```

We can verify the same in NameNode UI: <http://localhost:9870/dfshealth.html>

Open NameNode UI, go to **Utilities** tab and select **Browse the file system** option. Enter the directory name `/user/hive/warehouse` and you can see `hive_embedded_derby_db.db` folder available.



http://localhost:9870/explorer.html#/user/hive/warehouse/

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities ▾

Utilities ▾

Browse the file system

Logs Log Level Metrics Configuration Process Thread Dump Network Topology

Search:

Browser Directory

/user/hive/warehouse/

Show 25 entries

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	hp	supergroup	0 B	Jun 01 08:27	0	0 B	hive_embedded_derby_db.db

Showing 1 to 1 of 1 entries

Previous 1 Next

Hadoop, 2023.

- Create a table in Hive metastore:

```
use hive_embedded_derby_db;
create table employees(emp_id int, emp_name string, emp_salary int);
show tables;
```

```

hive> create table employees(emp_id int, emp_name string, emp_salary int);
OK
Time taken: 1.906 seconds
hive> show tables;
OK
employees
Time taken: 0.065 seconds, Fetched: 1 row(s)
hive>

```

We can see the above table is saved under

/user/hive/warehouse/hive_embedded_derby_db.db HDFS location.

Open another Command Prompt and run this command:

```
hadoop fs -ls /user/hive/warehouse/hive_embedded_derby_db.db
```

```

C:\Users\hp>hadoop fs -ls /user/hive/warehouse/hive_embedded_derby_db.db
Found 1 items
drwxr-xr-x  - hp supergroup          0 2024-06-01 08:27 /user/hive/warehouse/hive_embedded_derby_db/
employees

C:\Users\hp>hadoop fs -ls /user/hive/warehouse/hive_embedded_derby_db.db/employees

C:\Users\hp>

```

We can verify the same in NameNode UI. Click on `hive_embedded_derby_db.db` folder in `/user/hive/warehouse` directory and you can see `employees` folder available.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	hp	supergroup	0 B	Jun 01 08:27	0	0 B	employees

Showing 1 to 1 of 1 entries

Hadoop, 2023.

- Insert data into the table:

```
insert into employees values (101, 'johnson', 5000);
```

The above insert command submits the MapReduce job to get the record into table.

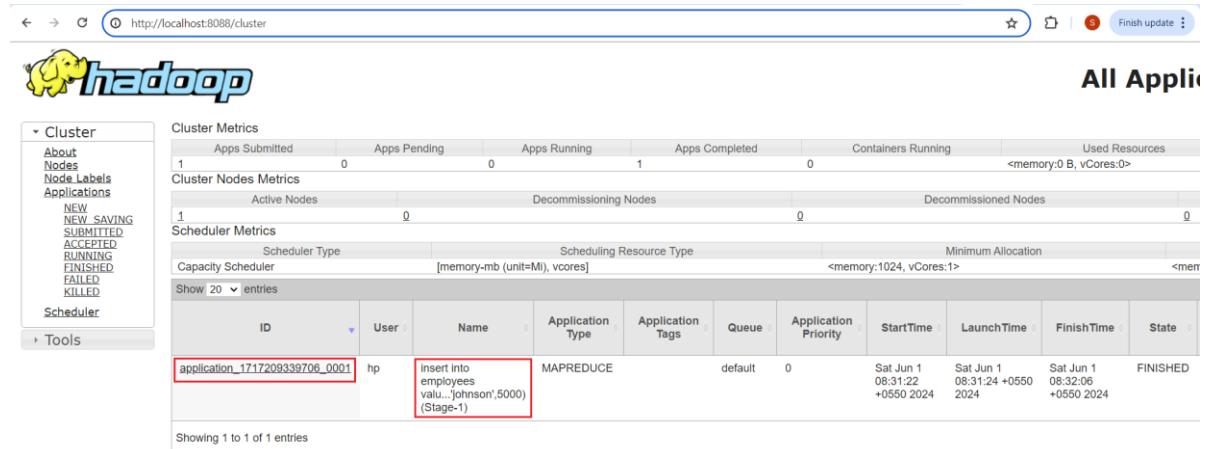
```

hive> insert into employees values (101, 'johnson',5000);
Query ID = hp_20240601083116_1f9d3943-4ecc-46e3-a92c-81f41dbb447a
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1717209339706_0001, Tracking URL = http://DESKTOP-KGH2E2G:8088/proxy/application_1717209339706_0001
Kill Command = D:\Program Files\Hadoop\hadoop-3.3.6\bin\mapred_job -kill job_1717209339706_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-06-01 08:31:43,284 Stage-1 map = 0%, reduce = 0%
2024-06-01 08:31:55,110 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.076 sec
2024-06-01 08:32:06,824 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 10.668 sec
MapReduce Total cumulative CPU time: 10 seconds 668 msec
Ended Job = job_1717209339706_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9820/user/hive/warehouse/hive_embedded_derby_db.db/employees/.hive-staging_hive_2024-06-01_08-31-16_275_2897099383453422591-l/_ext-10000
Loading data to table hive_embedded_derby_db.employees
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 10.668 sec HDFS Read: 16868 HDFS Write: 308 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 668 msec
OK
Time taken: 53.415 seconds
hive>

```

We can track the job status on YARN UI: <http://localhost:8088/cluster>

In YARN UI, you can see an application name with `insert into employees...` that was executed.



The screenshot shows the Apache Hadoop YARN UI interface. On the left, there's a sidebar with navigation links like Cluster, About, Nodes, Node Labels, Applications (with sub-options: NEW, NEW_SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED), Scheduler, and Tools. The main area is titled "All Applications". It displays a table of applications with columns: ID, User, Name, Application Type, Application Tags, Queue, Application Priority, Start Time, Launch Time, Finish Time, and State. One row in the table is highlighted with a red box, corresponding to the application ID "application_1717209339706_0001". The application details shown are: User: hp, Name: "insert into employees values ('johnson',5000) (Stage-1)", Application Type: MAPREDUCE, Application Tags: default, Queue: 0, Start Time: Sat Jun 1 08:31:22 2024, Launch Time: 08:31:24 +0550 2024, Finish Time: Sat Jun 1 08:32:06 +0550 2024, and State: FINISHED.

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	Start Time	Launch Time	Finish Time	State
application_1717209339706_0001	hp	insert into employees values ('johnson',5000) (Stage-1)	MAPREDUCE	default	0		Sat Jun 1 08:31:22 2024	08:31:24 +0550 2024	Sat Jun 1 08:32:06 +0550 2024	FINISHED

Click on the application ID to see the additional job details and logs.

The screenshot shows the Apache Hadoop Cluster Application Overview page. The URL is http://localhost:8088/cluster/app/application_1717209339706_0001. The application name is highlighted in red as "application_1717209339706_0001". The application details include:

- User: hp
- Name: insert into employees val... [Johnson],5000 (Stage-1)
- Application Type: MAPREDUCE
- Application Tags:
- Application Priority: 0 (Higher Integer value indicates higher priority)
- YarnApplicationState: FINISHED
- Queue: default
- FinalStatus Reported by AM: SUCCEEDED
- Started: Sat Jun 01 08:31:22 +0530 2024
- Launched: Sat Jun 01 08:31:24 +0530 2024
- Finished: Sat Jun 01 08:32:06 +0530 2024
- Elapsed: 43sec
- Tracking URL: History
- Log Aggregation Status: DISABLED
- Application Timeout (Remaining Time): Unlimited
- Diagnostics:
- Unmanaged Application: false
- Application Node Label expression: <Not set>
- AM container Node Label expression: <DEFAULT_PARTITION>

Application Metrics

Total Resource Preempted:	<memory:0, vCores:0>
Total Number of Non-AM Containers Preempted:	0
Total Number of AM Containers Preempted:	0
Resource Preempted from Current Attempt:	<memory:0, vCores:0>
Number of Non-AM Containers Preempted from Current Attempt:	0
Aggregate Resource Allocation:	122432 MB-seconds, 70 vcore-seconds
Aggregate Preempted Resource Allocation:	0 MB-seconds, 0 vcore-seconds

Logs

Show: 20 entries	Search:				
Attempt ID	Started	Node	Logs	Nodes blacklisted by the app	Nodes blacklisted by the system
appattempt_1717209339706_0001_000001	Sat Jun 01 08:31:23 +0550 2024	http://192.168.56.1:8042	Logs	0	0

Showing 1 to 1 of 1 entries

After the above job is succeeded, we can see a file created under /user/hive/warehouse/hive_embedded_derby_db.db/employees HDFS location.

Open another Command Prompt and run this command:

```
hadoop fs -ls /user/hive/warehouse/hive_embedded_derby_db.db/employees
```

```
C:\Users\hp>hadoop fs -ls /user/hive/warehouse/hive_embedded_derby_db.db/employees
Found 1 items
-rw-r--r-- 1 hp supergroup          17 2024-06-01 08:31 /user/hive/warehouse/hive_embedded_derby_db/
employees/000000_0
C:\Users\hp>
```

In NameNode UI, click on employees folder in /user/hive/warehouse/hive_embedded_derby_db.db directory and you can see a file 000000_0 available.

The screenshot shows the Hadoop NameNode UI at http://localhost:9870/explorer.html#/user/hive/warehouse/hive_embedded_derby_db.db/employees. The 'Employees' table is listed with one row:

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
	-rw-r--r--	hp	supergroup	17 B	Jun 01 08:31	1	128 MB	000000_0

Browse Directory

/user/hive/warehouse/hive_embedded_derby_db.db/employees									Go!				
Show 25 entries									Search:				
	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name					
	-rw-r--r--	hp	supergroup	17 B	Jun 01 08:31	1	128 MB	000000_0					
Showing 1 to 1 of 1 entries													
Hadoop, 2023.													

- Select data from the Hive table:

```
select * from employees;
```

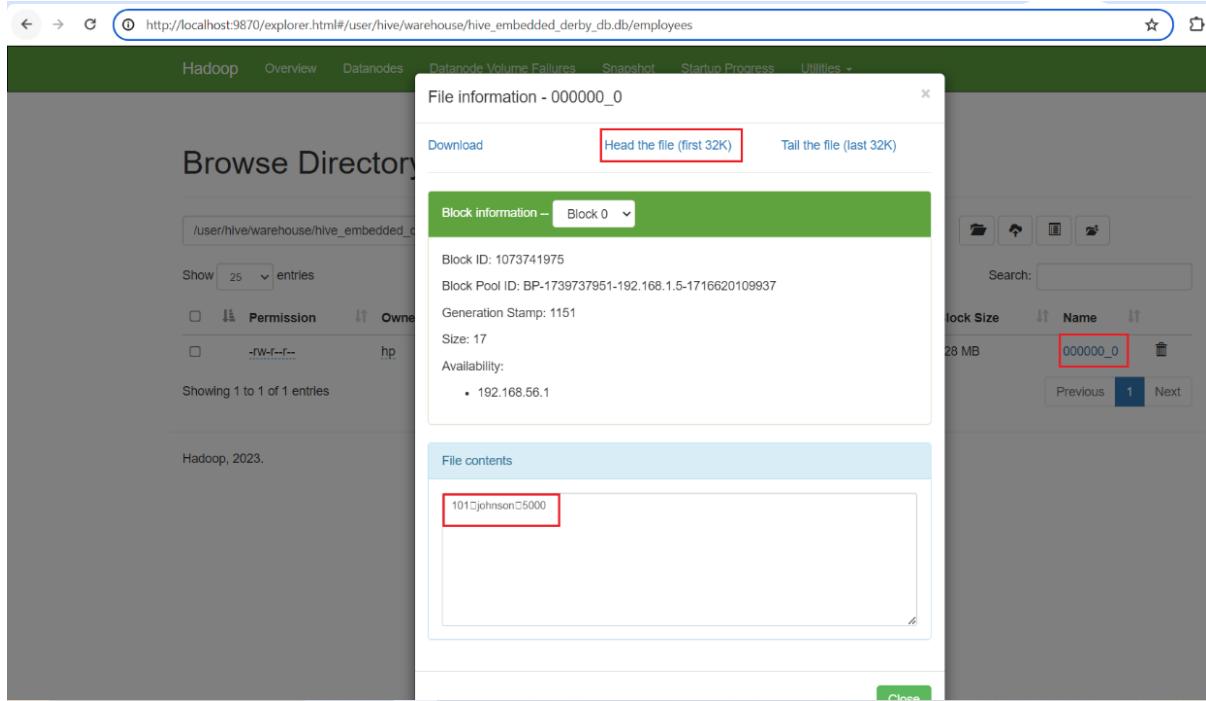
```
hive> select * from employees;
OK
101      johnson 5000
Time taken: 0.292 seconds, Fetched: 1 row(s)
hive>
```

We can verify the above table output in HDFS using the following command

```
hadoop fs -cat /user/hive/warehouse/hive_embedded_derby_db.db/employees/000000_0
```

```
C:\> Command Prompt
C:\Users\hp>hadoop fs -cat /user/hive/warehouse/hive_embedded_derby_db.db/employees/000000_0
101johnson@5000
C:\Users\hp>
```

In NameNode UI, click on 000000_0 file and select **Head the file** or **Tail the file** to see the file contents. We can download this file by clicking on **Download** option.



Note:

In embedded Hive metastore mode, we cannot start a second Hive session while the current one is active because embedded metastore allows only one connection at any time.

To test this, open another Command Prompt and start Hive shell and try to run a sample query (For example, `show databases;`).

```
Command Prompt - hive
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = c213aa40-3c24-4d0c-a885-8726affeb9a3

Logging initialized using configuration in jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> show tables;
FAILED: HiveException java.lang.RuntimeException: Unable to instantiate org.apache.hadoop.hive.ql.metadata.SessionHiveMetaStoreClient
hive>
```

You can see an error “**Unable to instantiate org.apache.hadoop.hive.ql.metadata.SessionHiveMetaStoreClient**” because another Hive

session is still active. Until the previous Hive session is closed, we cannot start another Hive session in Embedded metastore.

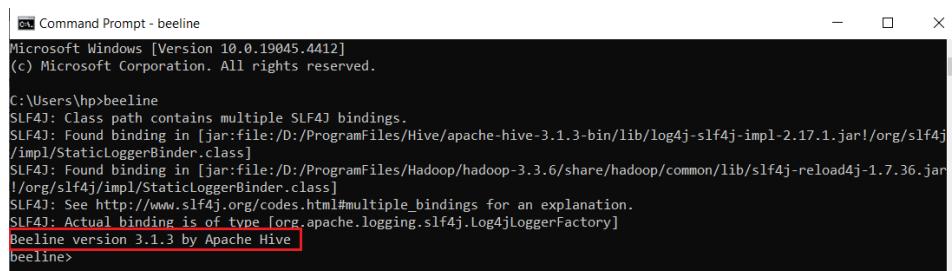
Close or exit out of all Hive CLI and make sure that no one other Hive session is active.

7.6. Beeline CLI:

Beeline is the advanced version of Command Line Interface that comes with HiveServer2 service. Beeline is a JDBC client that is based on SQLLine CLI. In embedded mode, Beeline connects to an embedded HiveServer2 service similar to Hive CLI and run Hive commands.

To start Beeline, open command prompt and run the following command.

```
beeline
```

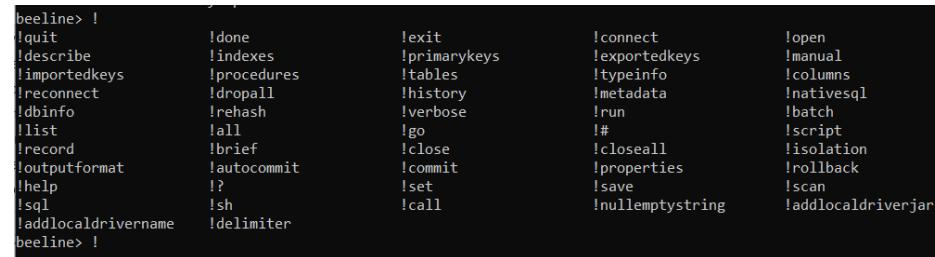


```
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>beeline
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j
./impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]
Beeline version 3.1.3 by Apache Hive
beeline>
```

After executing the above command, it prints the **Beeline version 3.1.3 by Apache Hive** and provides us `beeline>` prompt.

To get a list of Beeline commands, type `!` on `beeline>` prompt and press Tab key. Go through [this Apache Hive documentation](#) to get more understanding on these commands.



```
beeline> !
!quit          !done           !exit          !connect        !open
!describe      !indexes        !primarykeys   !exportedkeys  !manual
!importedkeys  !procedures     !tables         !typeinfo      !columns
!reconnect     !dropall        !history       !metadata      !nativesql
!dbinfo        !rehash          !verbose       !run           !batch
!list          !all             !go            !#              !script
!record        !brief           !close          !closeall      !isolation
!outputformat  !autocommit     !commit         !properties    !rollback
!help          !?               !set            !save          !scan
!sql           !sh              !call           !nullemptystring !addlocaldriverjar
!addlocaldrivername !delimiter
beeline> !
```

To connect to embedded HiveServer2, enter the following command in `beeline>` terminal.

```
!connect jdbc:hive2://
```

It asks for username and password to connect. Enter the default username `scott` and password `tiger` which are provided by HiveServer2.

We can also provide HiveServer2 credentials in the `beeline connect` command itself with this command:

```
!connect jdbc:hive2:// scott tiger
```

To start beeline connecting to HiveServer2 directly, use the following command:

```
beeline -u jdbc:hive2:// -n scott -p tiger  
or  
beeline -u jdbc:hive2:// scott tiger
```

After beeline is connected, we can run interactive shell commands as we do in Hive CLI.

```
show databases;
use hive_embedded_derby_db;
show tables;
select * from employees;
```

To come out of beeline shell, use !quit command.

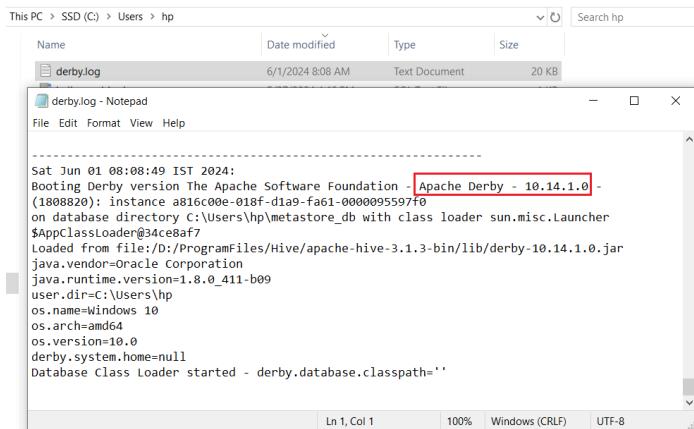
8. Configure Local Derby Metastore:

We can configure Hive metastore in local mode with Derby database by making Derby to run in network mode. For this, we should first download and install Apache Derby.

8.1. Install Apache Derby:

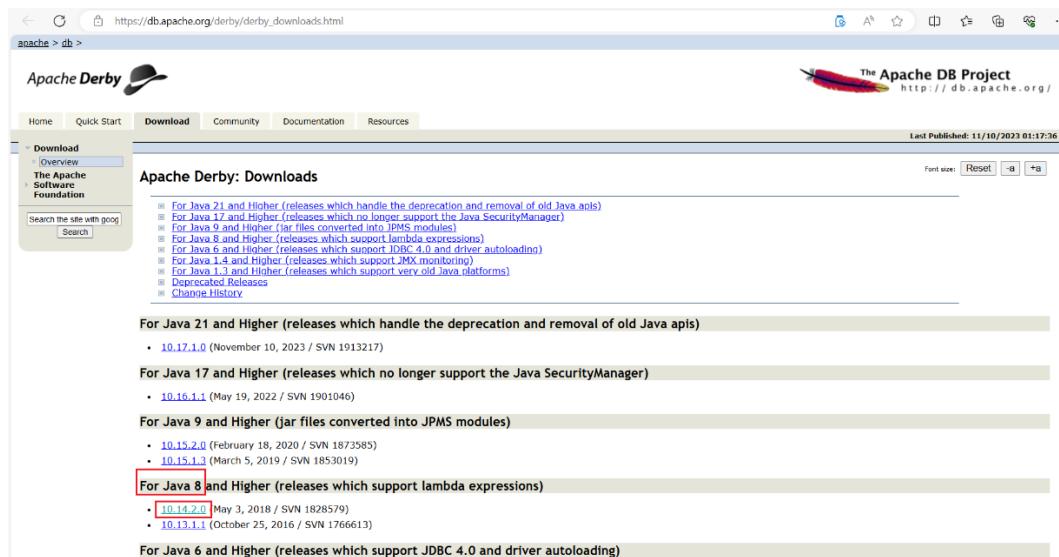
It is recommended to install Derby database of version that is shown for Embedded Derby installed with Hive.

Open `derby.log` file available in the location where `metastore_db` is created to see the embedded Derby database version.



```
Sat Jun 01 08:08:49 IST 2024:  
Booting Derby version The Apache Software Foundation - Apache Derby - 10.14.1.0 -  
(1808820): instance a816c00e-018f-d1a9-fa61-0000095597f0  
on database directory C:\Users\hp\metastore_db with class loader sun.misc.Launcher  
$AppClassLoader@34ce8af7  
Loaded from file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/derby-10.14.1.0.jar  
java.vendor=Oracle Corporation  
java.runtime.version=1.8.0_411-b09  
user.dir=C:\Users\hp  
os.name=Windows 10  
os.arch=amd64  
os.version=10.0  
derby.system.home=null  
Database Class Loader started - derby.database.classpath=""
```

Additionally, we can check the appropriate Derby version in [Apache Derby Downloads](#) page for supported versions of Java releases. Since we have Java 8 to run Hadoop and Hive, the latest Derby release for Java 8 is **Apache Derby 10.14.2.0** version.



The Apache DB Project <http://db.apache.org>

Last Published: 11/10/2023 01:17:36

Font size:

Apache Derby: Downloads

For Java 21 and Higher (releases which handle the deprecation and removal of old Java apis)

- [10.17.1.0](#) (November 10, 2023 / SVN 1913217)

For Java 17 and Higher (releases which no longer support the Java SecurityManager)

- [10.16.1.1](#) (May 19, 2022 / SVN 1901046)

For Java 9 and Higher (jar files converted into JPMS modules)

- [10.15.2.0](#) (February 18, 2020 / SVN 1873985)
- [10.15.1.3](#) (March 5, 2019 / SVN 1853019)

For Java 8 and Higher (releases which support lambda expressions)

- [10.14.2.0](#) (May 3, 2018 / SVN 1828579)
- [10.13.1.1](#) (October 25, 2016 / SVN 1766613)

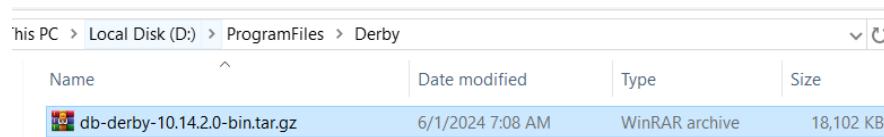
For Java 6 and Higher (releases which support JDBC 4.0 and driver autoloading)

Download db-derby-10.14.2.0-bin.tar.gz file from the [Apache Derby 10.14.2.0 Release](https://db.apache.org/derby/releases/release-10_14_2_0.html) website which gets downloaded to your **Downloads** folder.

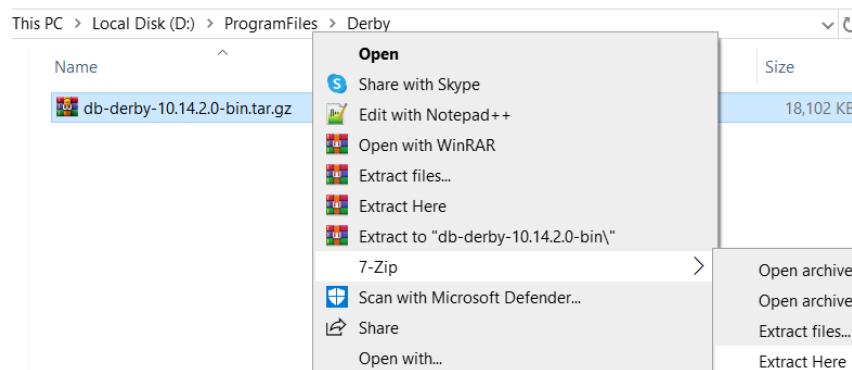
The screenshot shows the Apache Derby 10.14.2.0 Release page. The left sidebar contains a navigation menu with sections like Home, Quick Start, Download, Community, Documentation, and Resources. The main content area is titled "Apache Derby 10.14.2.0 Release". Under "Distributions", there is a link to "Release Notes for Apache Derby 10.14.2.0" which includes an "Overview", "New Features", "Bug Fixes", "Issues", "Build Environment", and "Verifying Releases". Below this, there is a section titled "Distributions" with a note about verifying integrity. It lists four distribution types: bin, lib, lib-debug, and src. For each type, there are two links: one for PGP signature and one for MD5 checksum. The "bin" distribution link is highlighted with a red box. The page footer indicates it was last published on 11/10/2023 at 01:17:36.

After the file is downloaded, unpack it using any file archiver (7zip or WinRAR) utility as below.

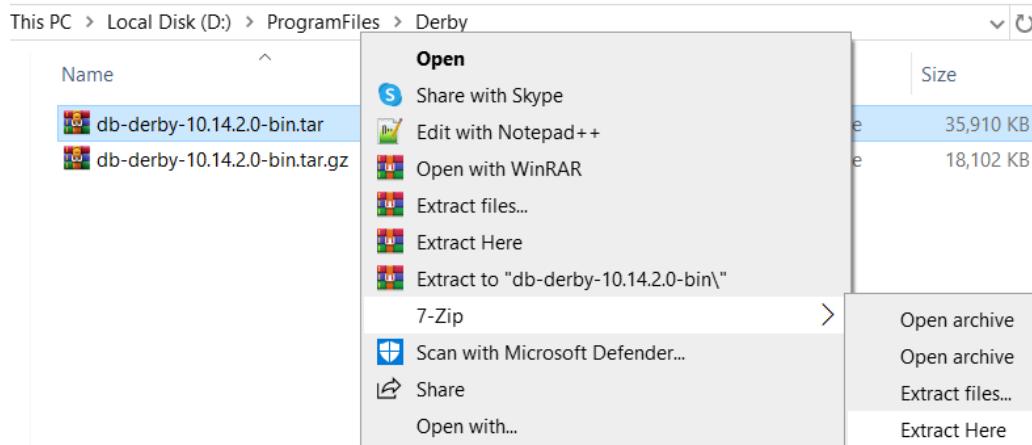
- Choose the installation directory in your machine and copy db-derby-10.14.2.0-bin.tar.gz file to that directory. For example, I am choosing my Derby installation directory as D:\ProgramFiles\Derby.



- Right click on db-derby-10.14.2.0-bin.tar.gz and choose **7-Zip -> Extract Here** option which extracts a new packed file db-derby-10.14.2.0-bin.tar.



- Next, unpack `db-derby-10.14.2.0-bin.tar` file using 7zip utility.



- The tar file extraction may take few seconds to finish. After finishing, you see a folder named `db-derby-10.14.2.0-bin` which consists of Derby binaries and libraries.

Name	Date modified	Type	Size
bin	6/1/2024 8:52 AM	File folder	
demo	6/1/2024 8:52 AM	File folder	
docs	6/1/2024 8:52 AM	File folder	
javadoc	6/1/2024 8:52 AM	File folder	
lib	6/1/2024 8:52 AM	File folder	
test	6/1/2024 8:52 AM	File folder	
index.html	3/10/2018 10:01 PM	Chrome HTML Do...	5 KB
KEYS	4/7/2018 6:44 AM	File	46 KB
LICENSE	4/7/2018 6:44 AM	File	12 KB
NOTICE	4/7/2018 6:44 AM	File	13 KB
RELEASE-NOTES.html	4/7/2018 6:44 AM	Chrome HTML Do...	7 KB

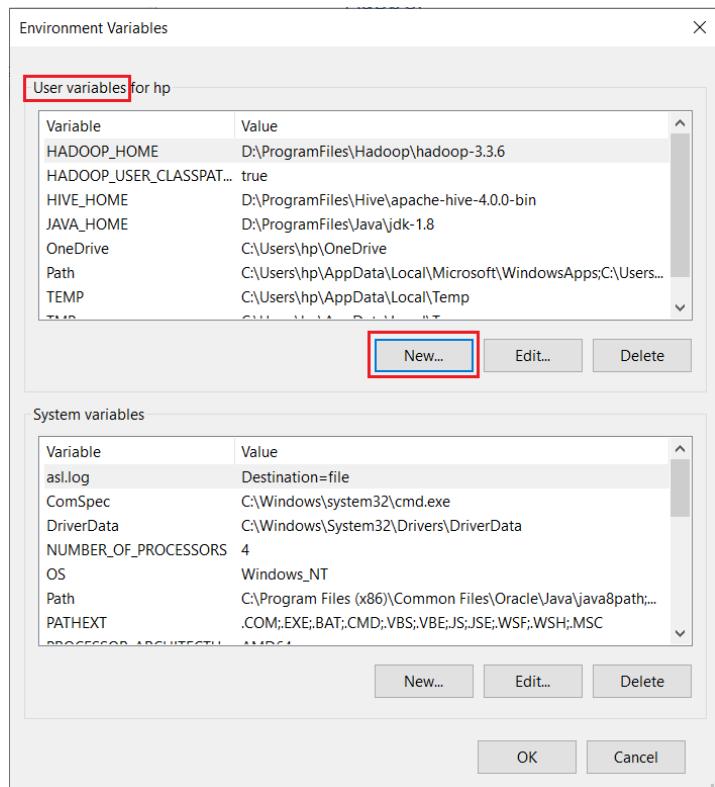
8.2. Set up Environment Variables:

After installing Derby, we should configure two environment variables defining Derby installation path.

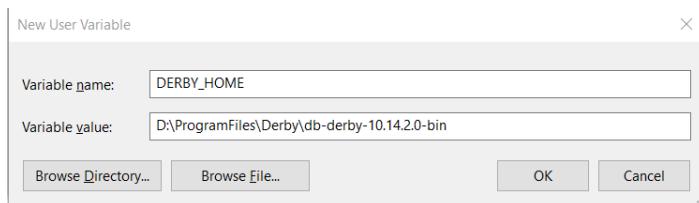
- **DERBY_HOME**: This is the Derby installation directory path in the machine (*in my machine it is D:\ProgramFiles\Derby\db-derby-10.14.2.0-bin*)
- **DERBY_OPTS**: Set it to `-Dderby.system.home=%DERBY_HOME%` location. This variable is optional but required if you want to create Hive metastore database in a custom location other than default location.

In the Windows search bar, start typing “environment variables” and select the first match which opens up **System Properties** dialog. On the **System Properties** window, press **Environment Variables** button.

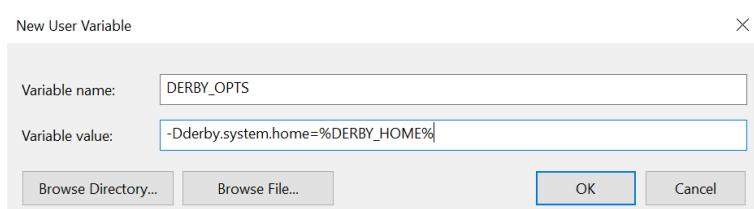
In the **Environment Variables** dialog, click on **New** under **User variables** section



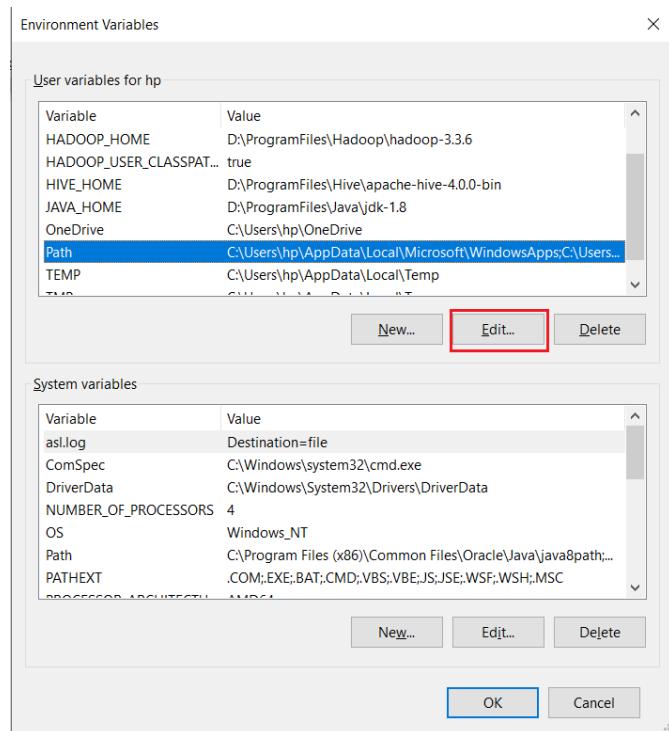
Add **DERBY_HOME** variable and press OK.



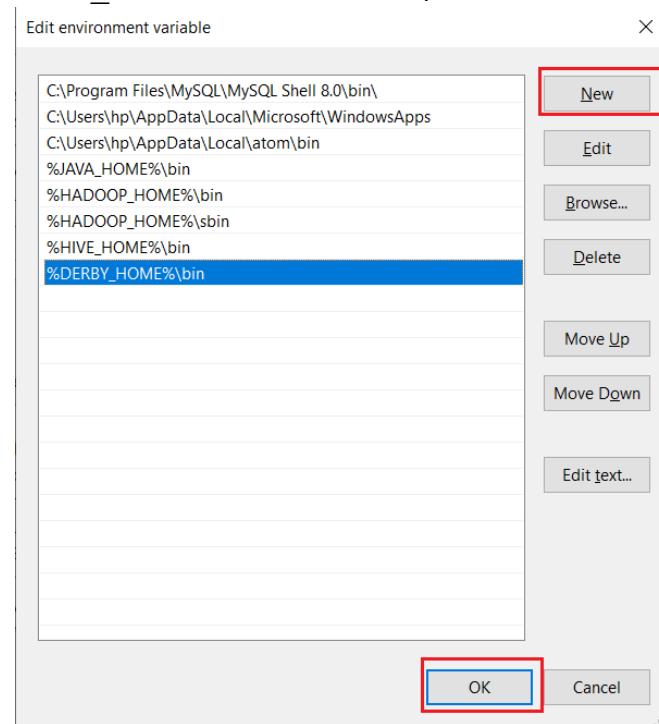
Click on **New** again and add **DERBY_OPTS** variable and press OK.



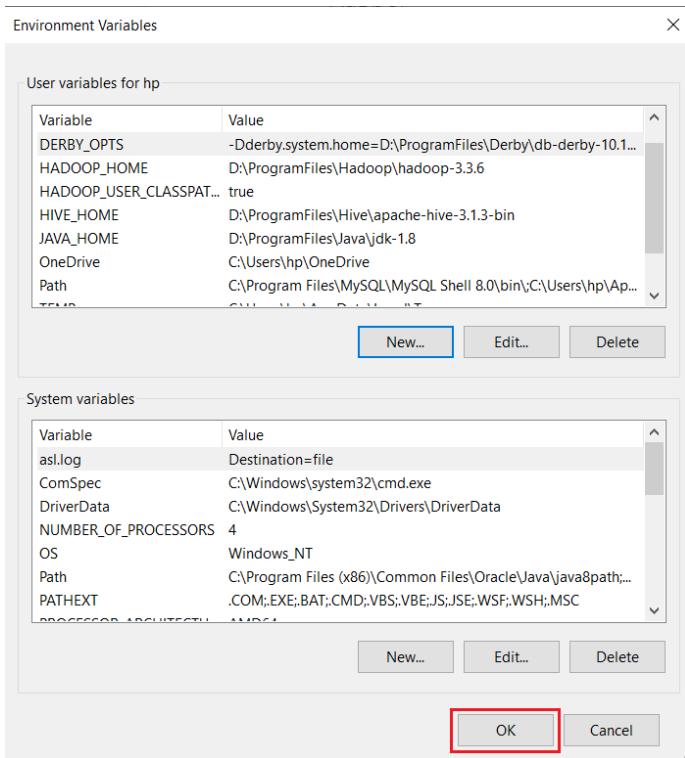
Select PATH variable and press **Edit** button



Press **New** and add `%DERBY_HOME%\bin` value and press **OK**.



Press OK to apply environment variable changes and close window.



8.3. Start Derby Network Server:

Now, start the Derby network server on the local host.

Open **Command Prompt** or Windows PowerShell in **Administrator** mode and run the following command

```
startNetworkServer -h 0.0.0.0
```

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

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> startNetworkServer -h 0.0.0.0
Sat Jun 01 08:54:11 IST 2024 : Security manager installed using the Basic server security policy.
Sat Jun 01 08:54:12 IST 2024 : Apache Derby Network Server - 10.14.2.0 - (1828579) started and ready to accept connections on port 1527
```

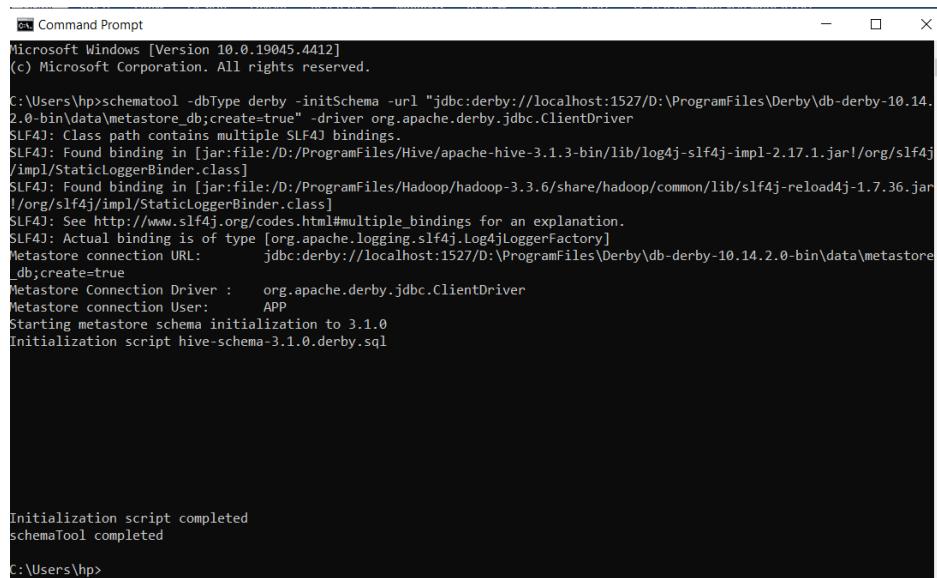
Apache Derby Network server runs on **1527** port by default.

8.4. Initialize Local Metastore:

Before starting Hive, we need to initialize the Hive Metastore using `schematool` utility to create remote metastore database.

Run the following `schematool` command which connects to Derby Network Server and creates `metastore_db` in `DERBY_HOME\data` location (*i.e D:\ProgramFiles\Derby\db-derby-10.14.2.0-bin\data location*)

```
schematool -dbType derby -initSchema -url "jdbc:derby://localhost:1527/D:\ProgramFiles\Derby\db-derby-10.14.2.0-bin\data\metastore_db;create=true" -driver org.apache.derby.jdbc.ClientDriver
```



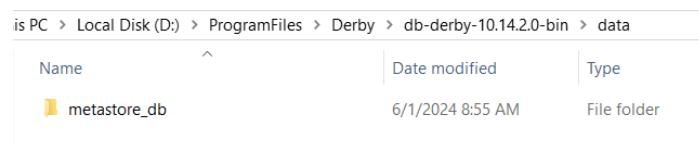
```
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>schematool -dbType derby -initSchema -url "jdbc:derby://localhost:1527/D:\ProgramFiles\Derby\db-derby-10.14.2.0-bin\data\metastore_db;create=true" -driver org.apache.derby.jdbc.ClientDriver
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:      jdbc:derby://localhost:1527/D:\ProgramFiles\Derby\db-derby-10.14.2.0-bin\data\metastore_db;create=true
Metastore Connection Driver :   org.apache.derby.jdbc.ClientDriver
Metastore connection User:     APP
Starting metastore schema initialization to 3.1.0
Initialization script hive-schema-3.1.0.derby.sql

Initialization script completed
schemaTool completed

C:\Users\hp>
```

After executing the above command, we can see that it created `metastore_db` folder in `DERBY_HOME\data` location.



8.5. Configure Hive Site:

Now, add Derby Network database configuration settings in `hive-site.xml` file for Hive to connect.

Go to `%HIVE_HOME%\conf` directory, create a file named `hive-site.xml` and paste the following code in the file. This code is referring to `metastore_db` in `DERBY_HOME\data` location.

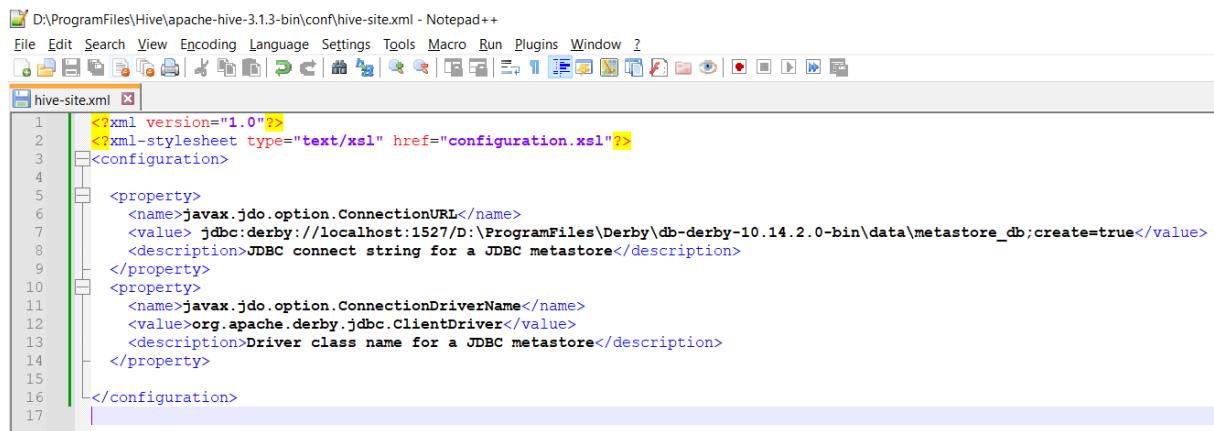
```

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>

    <property>
        <name>javax.jdo.option.ConnectionURL</name>
        <value> jdbc:derby://localhost:1527/D:\ProgramFiles\Derby\db-derby-10.14.2.0-bin\data\metastore_db;create=true</value>
        <description> JDBC connect string for a JDBC metastore</description>
    </property>
    <property>
        <name>javax.jdo.option.ConnectionDriverName</name>
        <value>org.apache.derby.jdbc.ClientDriver</value>
        <description> Driver class name for a JDBC metastore</description>
    </property>

</configuration>

```



8.6. Copy Derby Libraries:

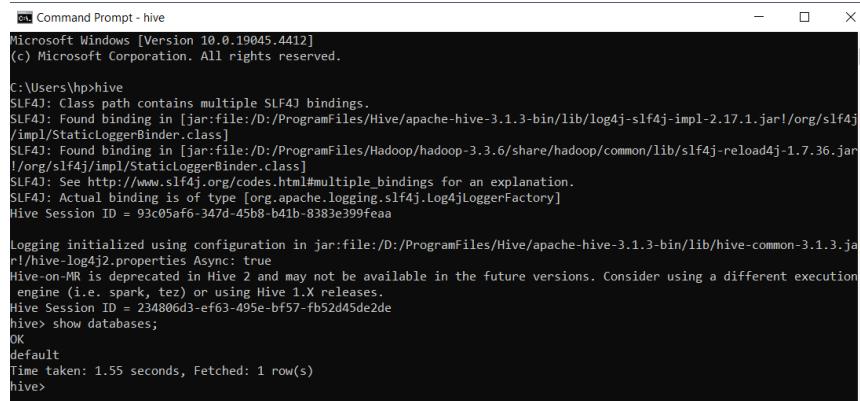
For Hive to communicate with local Derby, we need to copy `derbyclient.jar` and `derbytools.jar` files from `%DERBY_HOME%\lib` directory and paste them into `%HIVE_HOME%\lib` directory.

This PC > Local Disk (D:) > ProgramFiles > Hive > apache-hive-3.1.3-bin > lib			
Name	Date modified	Type	Size
datanucleus-rdbms-4.1.19.jar	10/19/2019 3:10 AM	Executable Jar File	1,864 KB
derby-10.14.1.0.jar	10/19/2019 3:05 AM	Executable Jar File	3,157 KB
derbyclient.jar	4/7/2018 6:40 AM	Executable Jar File	575 KB
derbytools.jar	4/7/2018 6:40 AM	Executable Jar File	226 KB
disruptor-3.3.6.jar	10/19/2019 3:11 AM	Executable Jar File	83 KB

8.7. Start Hive CLI:

Now, we can start the hive command line interface and run a sample query (For example, `show databases;`).

hive



```
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]

Hive Session ID = 93c05af6-347d-45b8-b41b-8383e399feab

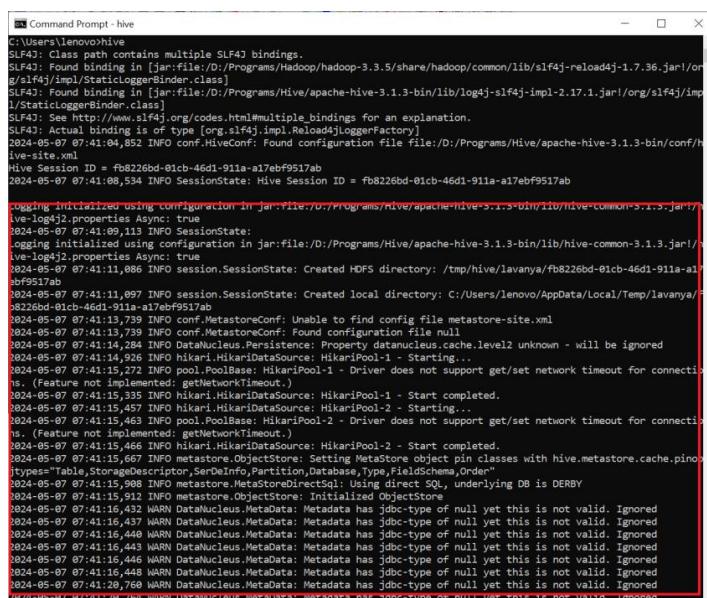
Logging initialized using configuration in jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Hive Session ID = 234806d3-ef63-495e-bf57-fb52d45de2de
hive> show databases;
OK
default
Time taken: 1.55 seconds, Fetched: 1 row(s)
hive>
```

Note: After running a query, if you receive “*FAILED: HiveException java.lang.RuntimeException: Unable to instantiate org.apache.hadoop.hive.ql.metadata.SessionHiveMetaStoreClient*” error then make sure `metastore_db` is initialized properly. If not done properly, stop the Derby Network server, delete the metastore database, reinitialize it using `schematool` utility, start Derby Network server and start hive.

8.8. Common Hive CLI Issues:

1. Unnecessary logging on Console:

You may see unnecessary logging to the console as below while launching `hive`.



```
C:\Users\lenovo>hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/Programs/Hadoop/hadoop-3.3.5/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/Programs/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.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.apache.logging.slf4j.Log4jLoggerFactory]

2024-05-07 07:41:04,652 INFO conf.HiveConf: Found configuration file file:/D:/Programs/Hive/apache-hive-3.1.3-bin/conf/hive-site.xml
Hive Session ID = f08226bd-01cb-46d1-911a-a17ebf9517ab
2024-05-07 07:41:08,534 INFO SessionState: Hive Session ID = f08226bd-01cb-46d1-911a-a17ebf9517ab

Logging initialized using configuration in jar:file:/D:/Programs/Hive/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
2024-05-07 07:41:09,113 INFO SessionState:
Logging initialized using configuration in jar:file:/D:/Programs/Hive/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
2024-05-07 07:41:11,866 INFO session.SessionState: Created HDFS directory: /tmp/hive/lavanya/f08226bd-01cb-46d1-911a-a17ebf9517ab
2024-05-07 07:41:11,997 INFO session.SessionState: Created local directory: C:/Users/lenovo/AppData/Local/Temp/lavanya/f08226bd-01cb-46d1-911a-a17ebf9517ab
2024-05-07 07:41:13,739 INFO conf.MetastoreConf: Unable to find config file metastore-site.xml
2024-05-07 07:41:13,739 INFO conf.MetastoreConf: Found configuration file null
2024-05-07 07:41:14,284 INFO DataNucleus.Persistence: Property datanucleus.cache.level unknown - will be ignored
2024-05-07 07:41:14,926 INFO hikari.HikariDataSource: HikariPool-1 - Starting...
2024-05-07 07:41:15,272 INFO pool.PoolBase: HikariPool-1 - Driver does not support get/set network timeout for connecti...
2024-05-07 07:41:15,335 INFO hikari.HikariDataSource: HikariPool-1 - Start completed.
2024-05-07 07:41:15,457 INFO hikari.HikariDataSource: HikariPool-2 - Starting...
2024-05-07 07:41:15,463 INFO pool.PoolBase: HikariPool-2 - Driver does not support get/set network timeout for connecti...
2024-05-07 07:41:15,466 INFO hikari.HikariDataSource: HikariPool-2 - Start completed.
2024-05-07 07:41:15,667 INFO metastore.ObjectStore: Setting MetaStore object pin classes with hive.metastore.cache.pinTypes="Table,StorageDescriptor,SerDeInfo,Partition,Database,Type,FieldSchema,Order"
2024-05-07 07:41:15,988 INFO metastore.MetaStoreDirectSql: Using direct SQL, underlying DB is DERBY
2024-05-07 07:41:16,132 WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
2024-05-07 07:41:16,432 WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
2024-05-07 07:41:16,437 WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
2024-05-07 07:41:16,440 WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
2024-05-07 07:41:16,443 WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
2024-05-07 07:41:16,446 WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
2024-05-07 07:41:16,448 WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
2024-05-07 07:41:29,760 WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
```

To suppress the above logging, make sure the following are set:

- Set the environment variable `HADOOP_USER_CLASSPATH_FIRST=true`
- Open `hive-config.cmd` file in `HIVE_HOME\bin` directory and add the below line at the end of the file.

```
set HADOOP_CLIENT_OPTS=%HADOOP_CLIENT_OPTS% -Dlog4j.configurationFile=hive-log4j2.properties
```

Then, launch the `hive` CLI from the new command prompt.

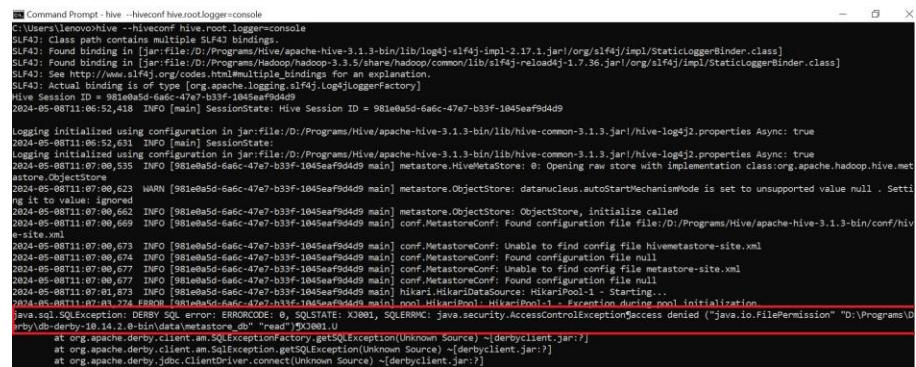
Note: It is important to always open new command terminal after environment variables are updated otherwise environment variables will not come into effect.

2. Failed to start database:

Sometimes, you may see the below errors in `hive`

`ERRORCODE: 0, SQLSTATE: XJ001, SQLERRMC: java.security.AccessControlException@access denied ("java.io.FilePermission" "D:\Programs\Derby\db-derby-10.14.2.0-bin\data\metastore_db" "read")`

`ERRORCODE: 40000, SQLSTATE: XJ040, SQLERRMC: Failed to start database 'D:\Programs\Derby\db-derby-10.14.2.0-bin\data\metastore_db' with class loader sun.misc.Launcher$AppClassLoader@511d50c0`



```
Command Prompt : hive --hiveconf hive.root.logger=console
C:\Users\lenovo\hive --hiveconf hive.root.logger=console
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/Programs/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.7.1.jar!/org/slf4j/jimpl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/Programs/Hadoop/hadoop-3.3.5/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/jimpl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/Programs/Hadoop/hadoop-3.3.5/share/hadoop/common/lib/slf4j-slf4j-1.7.36.jar!/org/slf4j/jimpl/StaticLoggerBinder.class]
SLF4J: Actual binding is of type [org.apache.logging.slf4j.LogFactory]
Hive Session ID = 981e8a5d-6a6c-47e7-b33f-1045ea9d4d9
2024-05-08T11:06:52,418 INFO [main] SessionState: Hive Session ID = 981e8a5d-6a6c-47e7-b33f-1045ea9d4d9

Logging initialized using configuration in jar:file:/D:/Programs/Hive/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
2024-05-08T11:06:52,631 INFO [main] SessionState: Logging initialized using configuration in jar:file:/D:/Programs/Hive/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
2024-05-08T11:07:00,535 INFO [981e8a5d-6a6c-47e7-b33f-1045ea9d4d9 main] metastore.HiveMetaStore: 0: Opening raw store with implementation class:org.apache.hadoop.hive.metastore.ObjectStore
2024-05-08T11:07:00,535 WARN [981e8a5d-6a6c-47e7-b33f-1045ea9d4d9 main] metastore.ObjectStore: datanucleus.autoStartMechanismMode is set to unsupported value null . Setting it to auto
2024-05-08T11:07:00,662 INFO [981e8a5d-6a6c-47e7-b33f-1045ea9d4d9 main] metastore.ObjectStore: ObjectStore_initialize called
2024-05-08T11:07:00,662 INFO [981e8a5d-6a6c-47e7-b33f-1045ea9d4d9 main] conf.MetastoreConf: Found configuration file: /D:/Programs/Hive/apache-hive-3.1.3-bin/conf/hive-site.xml
2024-05-08T11:07:00,673 INFO [981e8a5d-6a6c-47e7-b33f-1045ea9d4d9 main] conf.MetastoreConf: Unable to find config file hivemetastore-site.xml
2024-05-08T11:07:00,673 INFO [981e8a5d-6a6c-47e7-b33f-1045ea9d4d9 main] conf.MetastoreConf: Using default configuration file
2024-05-08T11:07:00,677 INFO [981e8a5d-6a6c-47e7-b33f-1045ea9d4d9 main] conf.MetastoreConf: Unable to find config file metastore-site.xml
2024-05-08T11:07:00,677 INFO [981e8a5d-6a6c-47e7-b33f-1045ea9d4d9 main] conf.MetastoreConf: Found configuration file null
2024-05-08T11:07:01,873 INFO [981e8a5d-6a6c-47e7-b33f-1045ea9d4d9 main] hikari.HikariDataPool: HikariPool-1 - Starting...
2024-05-08T11:07:03,774 INFO [981e8a5d-6a6c-47e7-b33f-1045ea9d4d9 main] pool.HikariPool: HikariPool-1 - Eviction during pool initialization
java.sql.SQLException: DERBY SQL error: ERRORCODE: 0, SQLSTATE: XJ001, SQLERRMC: java.security.AccessControlException@access denied ("java.io.FilePermission" "D:\Programs\Derby\db-derby-10.14.2.0-bin\data\metastore_db" "read")
        at org.apache.derby.client.am.SQLExceptionFactory.getSQLException(Unknown Source) ~[derbyclient.jar:?]
        at org.apache.derby.client.am.SqlException.getSQLException(Unknown Source) ~[derbyclient.jar:?]
        at org.apache.derby.jdbc.ClientDriver.connect(Unknown Source) ~[derbyclient.jar:?]
```

These errors arise because Hive is trying to connect in network mode to Derby database `metastore_db` located in `DERBY_HOME\data` directory which is different to the location from where the Derby's Network server was started and so it throws “*access denied (java.io.FilePermission)*” error.

To fix the above error:

- Set the below environment variable which tells Derby to consider the system home path as `DERBY_HOME` directory where derby database is located.

```
DERBY_OPTS=-Dderby.system.home=%DERBY_HOME%
```

- Open new command prompt and restart the Derby network server using the following command.

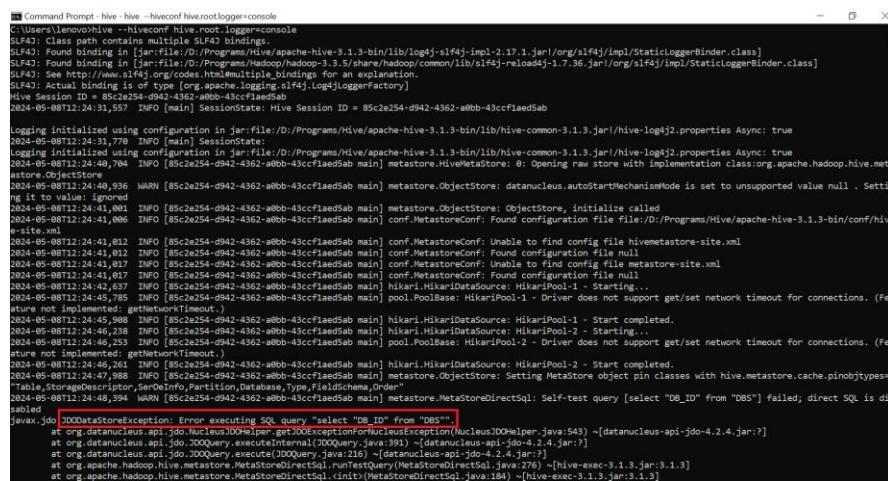
```
D:\Programs\Derby\db-derby-10.14.2.0-bin\bin\startNetworkServer -h 0.0.0.0
```

Then, launch the `hive` CLI from the new command prompt.

3. Table missing error:

Sometimes, you may see the below errors while launching `hive`

ERROR 42X05: Table/View 'DBS' does not exist. Required table missing : "VERSION" in Catalog "" Schema "". DataNucleus requires this table to perform its persistence operations. Either your MetaData is incorrect, or you need to enable "datanucleus.schema.autoCreateTables" ERROR metastore.RetryingHMSHandler: MetaException(message:Version information not found in metastore.)



```
[...]
2024-05-08T12:24:39,988 INFO [main] SessionState: Hive Session ID = 85c2e254-d942-4362-aebb-43ccf1ead5ab
Logging initialized using configuration in jar:file:/D:/Programs/Hive/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
2024-05-08T12:24:39,988 INFO [main] SessionState: Hive Session ID = 85c2e254-d942-4362-aebb-43ccf1ead5ab
Logging initialized using configuration in jar:file:/D:/Programs/Hive/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
2024-05-08T12:24:40,994 INFO [85c2e254-d942-4362-aebb-43ccf1ead5ab main] metastore.HiveMetaStore: 0: Opening raw store with implementation class:org.apache.hadoop.hive.metastore.ObjectStore
2024-05-08T12:24:40,996 WARN [85c2e254-d942-4362-aebb-43ccf1ead5ab main] metastore.ObjectStore: datanucleus.autoStartMechanismMode is set to unsupported value null . Setting it to false.
2024-05-08T12:24:41,001 INFO [85c2e254-d942-4362-aebb-43ccf1ead5ab main] metastore.ObjectStore: initialize called
2024-05-08T12:24:41,006 INFO [85c2e254-d942-4362-aebb-43ccf1ead5ab main] conf.MetastoreConf: Found configuration file file:/D:/Programs/Hive/apache-hive-3.1.3-bin/conf/hive-site.xml
2024-05-08T12:24:41,012 INFO [85c2e254-d942-4362-aebb-43ccf1ead5ab main] conf.MetastoreConf: Unable to find config file hivemetastore-site.xml
2024-05-08T12:24:41,012 INFO [85c2e254-d942-4362-aebb-43ccf1ead5ab main] conf.MetastoreConf: Found configuration file file:/D:/Programs/Hive/apache-hive-3.1.3-bin/conf/hive-site.xml
2024-05-08T12:24:41,017 INFO [85c2e254-d942-4362-aebb-43ccf1ead5ab main] conf.MetastoreConf: Unable to find config file metastore-site.xml
2024-05-08T12:24:41,017 INFO [85c2e254-d942-4362-aebb-43ccf1ead5ab main] conf.MetastoreConf: Found configuration file null
2024-05-08T12:24:42,637 INFO [85c2e254-d942-4362-aebb-43ccf1ead5ab main] hikaricp.HikariDataSource: HikariPool-1 - Starting...
2024-05-08T12:24:45,792 INFO [85c2e254-d942-4362-aebb-43ccf1ead5ab main] pool.PoolBase: HikariPool-1 - Driver does not support get/set network timeout for connections. (Fe
2024-05-08T12:24:45,792 INFO [85c2e254-d942-4362-aebb-43ccf1ead5ab main] hikaricp.HikariDataSource: HikariPool-1 - Start completed.
2024-05-08T12:24:46,238 INFO [85c2e254-d942-4362-aebb-43ccf1ead5ab main] hikaricp.HikariDataSource: HikariPool-2 - Starting...
2024-05-08T12:24:46,238 INFO [85c2e254-d942-4362-aebb-43ccf1ead5ab main] pool.PoolBase: HikariPool-2 - Driver does not support get/set network timeout for connections. (Fe
2024-05-08T12:24:47,988 INFO [85c2e254-d942-4362-aebb-43ccf1ead5ab main] metastore.ObjectStore: Setting MetaStore object pin classes with hive.metastore.cache.pinobjtypes=Table,StorageDescriptor,SerDeInfo,Partition,Database,Type,FieldSchema,Order
2024-05-08T12:24:48,393 WARN [85c2e254-d942-4362-aebb-43ccf1ead5ab main] metastore.MetaStoreDirectSql: Self-test query (select 'DB_ID' from 'DBS') failed; direct SQL is disabled
java.sql.SQLException: Error executing SQL query "select 'DB_ID' from 'DBS'" at org.datanucleus.api.jdo.NucleusJDOHelper.getJDOExceptionOrNucleusException(NucleusJDOHelper.java:549) ~[datanucleus-api-jdo-4.2.4.jar:?]
at org.datanucleus.api.jdo.JDOQuery.executeInternal(JDOQuery.java:391) ~[datanucleus-api-jdo-4.2.4.jar:?]
at org.datanucleus.api.jdo.JDOQuery.execute(JDOQuery.java:216) ~[datanucleus-api-jdo-4.2.4.jar:?]
at org.apache.hadoop.hive.metastore.MetastoreDirectSql.runTestQuery(MetastoreDirectSql.java:176) ~[hive-exec-3.1.3.jar:3.1.3]
at org.apache.hadoop.hive.metastore.MetastoreDirectSql.init(MetastoreDirectSql.java:184) ~[hive-exec-3.1.3.jar:3.1.3]
```

These errors arise because either the metastore database is not properly initialized or `hive-site.xml` file is not referring to the proper metastore database location.

To fix the above errors:

- Execute the following command to create `metastore_db` in a custom location such as `DERBY_HOME/data` folder

```
schematool -dbType derby -initSchema -url
"jdbc:derby:;databaseName=D:\Programs\Derby\db-derby-10.14.2.0-
bin\data\metastore_db;create=true" -driver
org.apache.derby.jdbc.EmbeddedDriver
```

- Open `hive-site.xml` in `HIVE_HOME\conf` directory and verify ConnectionURL is referring to the above metastore_db

```

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>javax.jdo.option.ConnectionURL</name>
<value>jdbc:derby://localhost:1527/D:/Programs/Derby/db-derby-10.14.2.0-bin\data\metastore_db;create=true</value>
<description>JDBC connect string for a JDBC metastore</description>
</property>

```

8.9. Run Queries on Hive CLI:

On `hive>` prompt, let us run the following queries to create a database, table and insert data.

```

create database hive_local_derby_db;
show databases;
use hive_local_derby_db;
create table employees(emp_id int, empp_name string, emp_salary int);
show tables;
insert into employees values (101, 'johnson',5000);

```

The above insert statement submits a MapReduce job to YARN.

```

hive> create database hive_local_derby_db;
OK
Time taken: 1.882 seconds
hive> show databases;
OK
default
hive local derby db
Time taken: 0.427 seconds, Fetched: 2 row(s)
hive> use hive_local_derby_db;
OK
Time taken: 0.067 seconds
hive> create table employees(emp_id int, empp_name string, emp_salary int);
OK
Time taken: 1.872 seconds
hive> show tables;
OK
employees
Time taken: 0.064 seconds, Fetched: 1 row(s)
hive> insert into employees values (101, 'johnson',5000);
Query ID = hp_20240601091135_d5eb3321-d299-4117-96e1-42d51c989c3c
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1717209339706_0002, Tracking URL: http://DESKTOP-KGH2E2G:8088/proxy/application_1717209339706_0002/
Kill Command = D:\Program Files\Hadoop\hadoop-3.3.6\bin\mapred job -kill job_1717209339706_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-06-01 09:12:00,190 Stage-1 map = 0%, reduce = 0%
2024-06-01 09:12:11,919 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.076 sec
2024-06-01 09:12:24,602 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 10.403 sec
MapReduce Total cumulative CPU time: 10 seconds 403 msec
Ended Job = job_1717209339706_0002
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9820/user/hive/warehouse/hive_local_derby_db.db/employees/.hive-staging_hive_2
024-06-01 09:11:35.913 1552783422734741116-1-ext-10000
Loading data to table hive_local_derby_db.employees
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 10.403 sec HDFS Read: 16848 HDFS Write: 305 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 403 msec
OK
Time taken: 51.29 seconds
hive>

```

We can track the job status on YARN UI: <http://localhost:8088/cluster>

In YARN UI, you can see an application name with insert into employees... that was executed.

Note: You may encounter this issue as I encountered it in my different machine.

- In Hive CLI, I noticed that the MapReduce job submitted was not progressing further.

```
hive> insert into employees select employee_id, first_name, last_name, email, phone_number, from_unixtime(unix_timestamp(hire_date,'dd-MMM-yy'), 'yyyy-MM-dd') as hire_date, job_id, salary, commission_pct, manager_id, department_id from employees_tmp;
Query ID = lavalnya_20240508192757_6d6f68f4-f946-4bcf-ac1b-087ba0fbe8d0
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1715176593261_0001, Tracking URL = http://kubernetes.docker.internal:8088/proxy/application_1715176593261_0001/
Kill Command = D:\Programs\Hadoop\hadoop-3.3.5\bin\mapred job -kill job_1715176593261_0001
```

- In YARN UI, the application status was showing “*Application is added to the scheduler and is not yet activated. Skipping AM assignment as cluster resource is empty.*”

- Then, I noticed that the node became unhealthy in YARN UI.

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	
application_17152175853261_0001	lavanya	Insert into employees select employee_id, emp_name, age	MAPREDUCE	default	0	Wed May 8 16:27:59 +0530 2014	N/A	N/A	ACCEPTED	UNDEFINED	0	0	-1	0	0	-1	0.0	0.0

- The health report of node says that *1/1 local-dirs usable space is below configured utilization percentage/no more usable space [/tmp/hadoop-lavanya/nm-local-dir : used space above threshold of 90.0%]*;

Label	rack	node state	node address	node http address	last health update	Health-report	Containers	Allocation steps	Mem Used	Mem Avail	Phys Mem Used %	VCore Used	VCore Avail	Phys VCore Used %	Version
localhost	UNHEALTHY	localhost	localhost:54314	localhost:54314	Wed May 8 19:52:35 +0530 2014	1/1 local-dir usable space is below configured utilization percentage/no more usable space [/tmp/hadoop-lavanya/nm-local-dir : used space above threshold of 90.0%].	0	0	0 B	0 B	0	0	0	3.35	

This error is because that the C: drive in local Windows system is left with just 7 GB out of 85 GB and Hadoop is using C:\tmp as the local temporary directory.

- To get rid of the above health check error, change yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage property to **98.5 %** as below:
 - Open `yarn-site.xml` file located in `HADOOP_HOME\etc\hadoop` directory and add the below property.

```
<property>
  <name>yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage</name>
  <value>98.5</value>
</property>
```

```

15 <configuration>
16   <!-- Site specific YARN configuration properties -->
17
18   <property>
19     <name>yarn.nodemanager.aux-services</name>
20     <value>mapreduce_shuffle</value>
21     <description>Yarn Node Manager Aux Service</description>
22   </property>
23
24   <property>
25     <name>yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage</name>
26     <value>98.5</value>
27   </property>
28

```

- Restart YARN services using the below commands and then rerun the above hive query.

```

stop-yarn.cmd
start-yarn.cmd

```

After the above insert query is completed, we can verify the output in HDFS using the following commands:

```

hadoop fs -ls /user/hive/warehouse
hadoop fs -ls /user/hive/warehouse/hive_local_derby_db.db
hadoop fs -ls /user/hive/warehouse/hive_local_derby_db.db/employees
hadoop fs -cat /user/hive/warehouse/hive_local_derby_db.db/employees/000000_0

```

```

C:\Users\hp>hadoop fs -ls /user/hive/warehouse
Found 2 items
drwxr-xr-x - hp supergroup          0 2024-06-01 08:27 /user/hive/warehouse/hive embedded derby db.db
drwxr-xr-x - hp supergroup          0 2024-06-01 09:11 /user/hive/warehouse/hive_local_derby_db.db

C:\Users\hp>hadoop fs -ls /user/hive/warehouse/hive_local_derby_db.db
Found 1 items
drwxr-xr-x - hp supergroup          0 2024-06-01 09:12 /user/hive/warehouse/hive_local_derby_db.db/emp
loyees

C:\Users\hp>hadoop fs -ls /user/hive/warehouse/hive_local_derby_db.db/employees
Found 1 items
-rw-r--r-- 1 hp supergroup          17 2024-06-01 09:12 /user/hive/warehouse/hive_local_derby_db.db/emp
loyees/000000_0

C:\Users\hp>hadoop fs -cat /user/hive/warehouse/hive_local_derby_db.db/employees/000000_0
1010johnson@5000

C:\Users\hp>

```

The same is visible in NameNode UI: <http://localhost:9870/dfshealth.html> also.

Open NameNode UI, go to **Utilities** tab and select **Browse the file system** option. Enter the directory name /user/hive/warehouse and you can see `hive_local_derby_db.db` folder available. Click on `hive_local_derby_db.db` folder to see `employees` folder in

which `000000_0` file available. Click on the file and select **Head the file** or **Tail the file** to see the file contents. We can download this file by clicking on **Download** option.

Name	Size	Last Modified	Replication	Block Size	Actions
<code>000000_0</code>	17 B	Jun 01 09:12	1	128 MB	

Since we enabled Derby to run as Network server, we should be able to make multiple Hive connections. To test this, open another Command Prompt and start Hive shell and try to run a sample query (For example, `show databases;`).

```
C:\ Command Prompt - hive
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = d20248f5-1e3b-4264-a48b-cc4e262120bf

Logging initialized using configuration in jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Hive Session ID = 9fd2832f-5325-4944-8e71-0378acf2512d
hive> show databases;
OK
default
hive_local_derby_db
Time taken: 1.5 seconds, Fetched: 2 row(s)
hive>
```

We are able to access metastore database without any issues while other Hive session is still active and accessing the same database.

To come out of `hive>` shell, use `exit;` command.

8.10. Beeline in Embedded Mode:

The Beeline works in embedded mode and remote mode. In embedded mode, Beeline connects to an embedded HiveServer2 similar to Hive CLI and in remote mode, it connects to HiveServer2

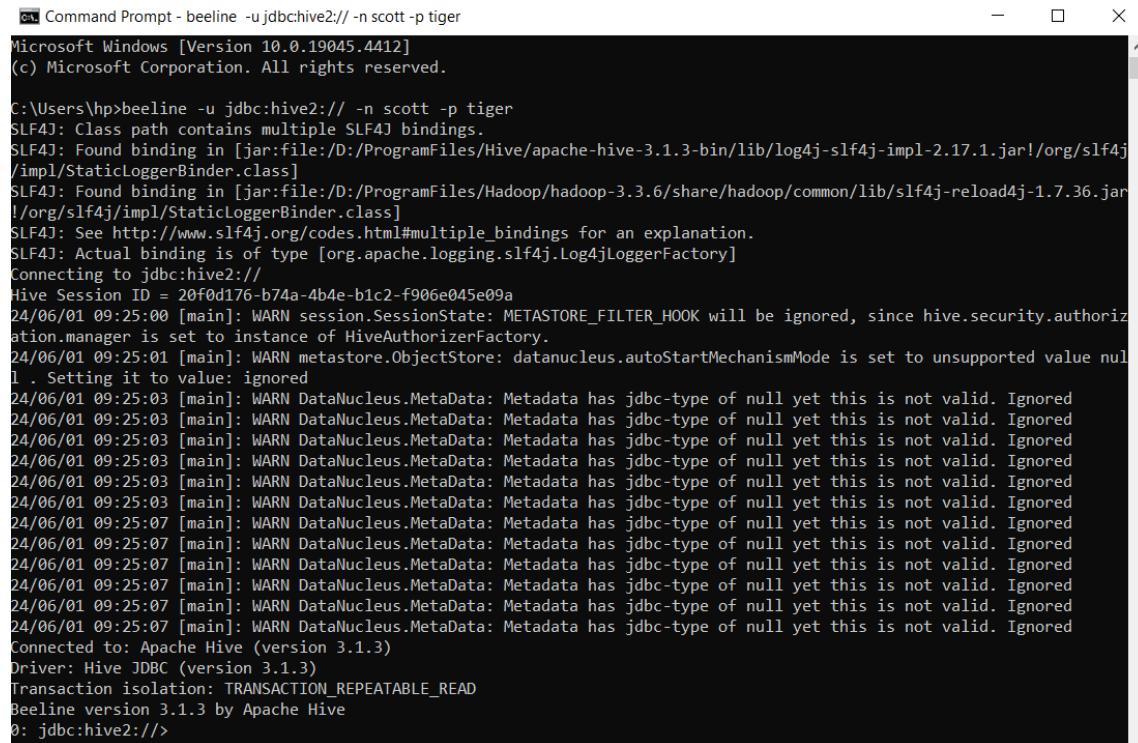
service over Thrift. Using Beeline, we can connect to Hive running on Local or Remote server using hostname and port.

To start Beeline in embedded mode, open command prompt and run the following command.

```
beeline -u jdbc:hive2:// -n scott -p tiger
```

or

```
beeline -u jdbc:hive2:// scott tiger
```



The screenshot shows a Windows Command Prompt window with the title "Command Prompt - beeline -u jdbc:hive2:// -n scott -p tiger". The window displays the following log output:

```
C:\Users\hp>beeline -u jdbc:hive2:// -n scott -p tiger
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>beeline -u jdbc:hive2:// -n scott -p tiger
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://
Hive Session ID = 20f0d176-b74a-4b4e-b1c2-f906e045e09a
24/06/01 09:25:00 [main]: WARN session.SessionState: METASTORE_FILTER_HOOK will be ignored, since hive.security.authorization.manager is set to instance of HiveAuthorizerFactory.
24/06/01 09:25:01 [main]: WARN metastore.ObjectStore: datanucleus.autoStartMechanismMode is set to unsupported value null . Setting it to value: ignored
24/06/01 09:25:03 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/01 09:25:03 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/01 09:25:03 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/01 09:25:03 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/01 09:25:03 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/01 09:25:03 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/01 09:25:03 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/01 09:25:07 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/01 09:25:07 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/01 09:25:07 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/01 09:25:07 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/01 09:25:07 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/01 09:25:07 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
Connected to: Apache Hive (version 3.1.3)
Driver: Hive JDBC (version 3.1.3)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.3 by Apache Hive
0: jdbc:hive2://>
```

After beeline is connected, we can execute queries as we do in Hive CLI.

```
show databases;
use hive_local_derby_db;
show tables;
select * from employees;
```

```
0: jdbc:hive2://> show databases;
OK
+-----+
| database_name |
+-----+
| default      |
| hive_local_derby_db |
+-----+
2 rows selected (3.069 seconds)
0: jdbc:hive2://> use hive_local_derby_db;
OK
No rows affected (0.091 seconds)
0: jdbc:hive2://> show tables;
OK
+-----+
| tab_name |
+-----+
| employees |
+-----+
1 row selected (0.098 seconds)
0: jdbc:hive2://> select * from employees;
OK
+-----+-----+-----+
| employees.emp_id | employees.empp_name | employees.emp_salary |
+-----+-----+-----+
| 101           | johnson        | 5000          |
+-----+-----+-----+
1 row selected (3.304 seconds)
0: jdbc:hive2://> !quit
Closing: 0: jdbc:hive2://

C:\Users\hp>
```

To come out of `beeline> shell`, use the `!quit` command.

8.11. Start HiveServer2 Service:

Before starting Beeline in remote mode that connects to HiveServer2, make sure that HiveServer2 service is running.

Open command prompt and start the HiveServer2 service using the following command:

```
hiveserver2 --hiveconf hive.root.logger=console
```

```
Command Prompt - hiveserver2 --hiveconf hive.root.logger=console
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>hiveserver2 --hiveconf hive.root.logger=console
"Starting Hive Server2"
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = e7885a5c-0e50-4516-88ce-81d88b88d466
2024-06-01T09:38:24,025 INFO [main] SessionState: Hive Session ID = e7885a5c-0e50-4516-88ce-81d88b88d466
2024-06-01T09:38:26,271 INFO [main] session.SessionState: Created HDFS directory: /tmp/hive/hp/e7885a5c-0e50-4516-88ce-81d88b88d466
2024-06-01T09:38:26,280 INFO [main] session.SessionState: Created local directory: C:/Users/hp/AppData/Local/Temp/hp/e7885a5c-0e50-4516-88ce-81d88b88d466
2024-06-01T09:38:26,304 INFO [main] session.SessionState: Created HDFS directory: /tmp/hive/hp/e7885a5c-0e50-4516-88ce-81d88b88d466/_tmp_space.db
2024-06-01T09:38:26,350 INFO [main] sqlstd.SQLStdHiveAccessController: Created SQLStdHiveAccessController for session context : HiveAuthzSessionContext [sessionString=e7885a5c-0e50-4516-88ce-81d88b88d466, clientType=HIVESERVER2]
2024-06-01T09:38:26,356 WARN [main] session.SessionState: METASTORE_FILTER_HOOK will be ignored, since hive.security.authorization.manager is set to instance of HiveAuthorizerFactory.
2024-06-01T09:38:26,820 INFO [main] metastore.HiveMetaStore: 0: Opening raw store with implementation class:org.apache.hadoop.hive.metastore.ObjectStore
2024-06-01T09:38:26,871 WARN [main] metastore.ObjectStore: datanucleus.autoStartMechanismMode is set to unsupported value null . Setting it to value: ignored
2024-06-01T09:38:26,890 INFO [main] metastore.ObjectStore: ObjectStore, initialize called
2024-06-01T09:38:26,892 INFO [main] conf.MetastoreConf: Found configuration file file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/conf/metastore.xml
2024-06-01T09:38:40,731 INFO [pool-10-thread-1] metastore.ObjectStore: ObjectStore, initialize called
2024-06-01T09:38:40,756 INFO [pool-10-thread-1] metastore.MetaStoreDirectSql: Using direct SQL, underlying DB is DERBY
2024-06-01T09:38:40,756 INFO [pool-10-thread-1] metastore.ObjectStore: Initialized ObjectStore
2024-06-01T09:38:40,786 INFO [pool-10-thread-1] metastore.HiveMetaStore: 1: get_tables_by_type: db=@hive#default pat=.* ,type=MATERIALIZED_VIEW
2024-06-01T09:38:40,786 INFO [pool-10-thread-1] metastore.HiveMetaStore: 1: get_tables_by_type: db=@hive#default pat=.* ,type=MATERIALIZED_VIEW
2024-06-01T09:38:40,797 INFO [pool-10-thread-1] metastore.HiveMetaStore: 1: get_multi_table : db=default tbls=
2024-06-01T09:38:40,797 INFO [pool-10-thread-1] metastore.HiveMetaStore: 1: get_multi_table : db=default tbls=
2024-06-01T09:38:40,802 INFO [pool-10-thread-1] metastore.HiveMetaStore: 1: get_tables_by_type: db=@hive_local_derby_db pat=.* ,type=MATERIALIZED_VIEW
2024-06-01T09:38:40,802 INFO [pool-10-thread-1] metastore.HiveMetaStore: 1: get_tables_by_type: db=@hive_local_derby_db pat=.* ,type=MATERIALIZED_VIEW
2024-06-01T09:38:40,806 INFO [pool-10-thread-1] metastore.HiveMetaStore: 1: get_multi_table : db=hive_local_derby_db tbls=
2024-06-01T09:38:40,806 INFO [pool-10-thread-1] metastore.HiveMetaStore: 1: get_multi_table : db=hive_local_derby_db tbls=
2024-06-01T09:38:40,806 INFO [pool-10-thread-1] metastore.HiveMetaStore: 1: get_tables_by_type: db=hive_local_derby_db pat=.* ,type=MATERIALIZED_VIEW
2024-06-01T09:38:40,806 INFO [pool-10-thread-1] metastore.HiveMetaStore: 1: get_tables_by_type: db=hive_local_derby_db pat=.* ,type=MATERIALIZED_VIEW
2024-06-01T09:38:41,747 INFO [main] thrift.ThriftCLIService: Starting ThriftBinaryCLIService on port 10000 with 5...500 worker threads
2024-06-01T09:38:41,761 INFO [main] server.Server: jetty-9.3.20.v20170531
2024-06-01T09:38:42,591 INFO [main] handler.ContextHandler: Started o.e.j.w.WebApplicationContext@601d295{/ ,file:///C:/Users/hp/AppData/Local/Temp/jetty-0.0.0-10002-hiveserver2-_-any-1283630274153401388.dir/webapp/,AVAILABLE}{jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/hive-service-3.1.3.jar!/hive-webapps/hiveserver2}
2024-06-01T09:38:42,592 INFO [main] handler.ContextHandler: Started o.e.j.s.ServletContextHandler@485451d8{/static, jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/hive-service-3.1.3.jar!/hive-webapps/static,AVAILABLE}
2024-06-01T09:38:42,608 INFO [main] server.AbstractConnector: Started ServerConnector@7ff6804a{HTTP/1.1,[http/1.1]}{0.0 .0:10002}
2024-06-01T09:38:42,608 INFO [main] server.Server: Started @28845ms
2024-06-01T09:38:42,608 INFO [main] http.HttpServer: Started HttpServer[hiveserver2] on port 10002
```

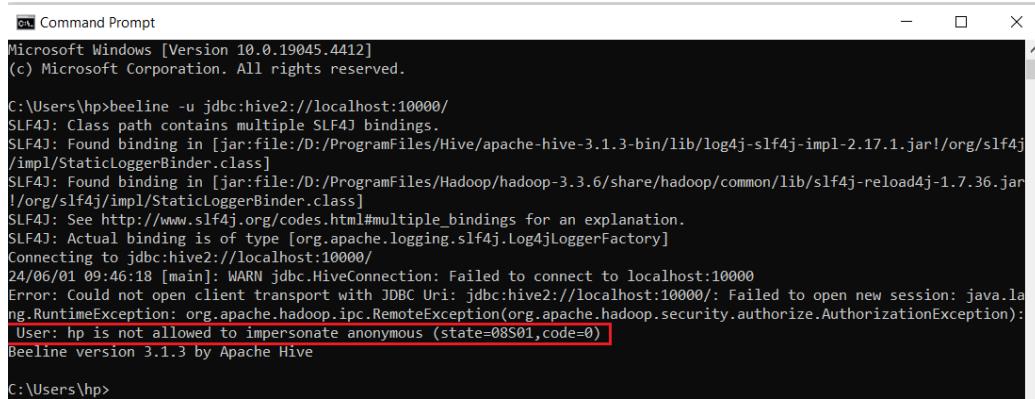
We can see that HiveServer2 service connected to **DERBY** metastore and started on port 10002 by default while ThriftBinaryCLIService started on port 10000.

8.12. Beeline in Remote Mode:

In Remote mode, Beeline connects to HiveServer2 over Thrift CLI server that runs on 10000 port. Beeline remote connection can be made as **anonymous** user or with **specific credentials**.

Run the following command to connect Beeline as anonymous user:

```
beeline -u jdbc:hive2://localhost:10000/
```



```
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

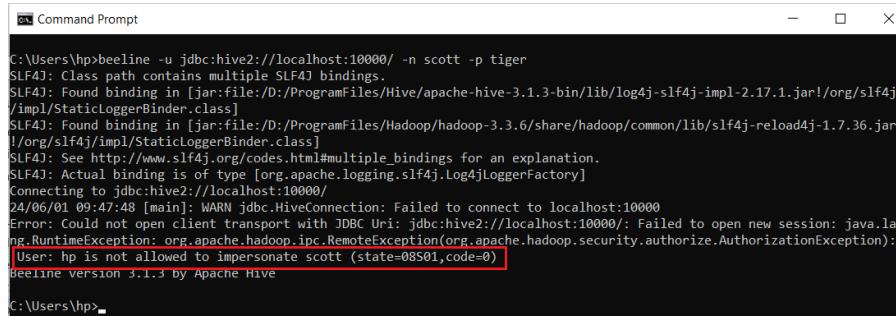
C:\Users\hp>beeline -u jdbc:hive2://localhost:10000/
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/share/hadoop/common/lib/slf4j-reload4j-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.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://localhost:10000
24/06/01 09:46:18 [main]: WARN jdbc.HiveConnection: Failed to connect to localhost:10000
Error: Could not open client transport with JDBC Uri: jdbc:hive2://localhost:10000/: Failed to open new session: java.lang.RuntimeException: org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.security.authorize.AuthorizationException):
User: hp is not allowed to impersonate anonymous (state=08S01,code=0)
Beeline version 3.1.3 by Apache Hive

C:\Users\hp>
```

Here, we see an error that "**AuthorizationException: User xxx is not allowed to impersonate anonymous**". This is because HiveServer2 does not allow impersonation by default.

Run the following command to connect Beeline with default user **scott**:

```
beeline -u jdbc:hive2://localhost:10000/ -n scott -p tiger
```



```
C:\Users\hp>beeline -u jdbc:hive2://localhost:10000/ -n scott -p tiger
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://localhost:10000
24/06/01 09:47:48 [main]: WARN jdbc.HiveConnection: Failed to connect to localhost:10000
Error: Could not open client transport with JDBC Uri: jdbc:hive2://localhost:10000/: Failed to open new session: java.lang.RuntimeException: org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.security.authorize.AuthorizationException):
User: hp is not allowed to impersonate scott (state=08S01,code=0)
Beeline version 3.1.3 by Apache Hive

C:\Users\hp>
```

Here, we see an error that "**AuthorizationException: User xxx is not allowed to impersonate scott**". This is because HiveServer2 does not allow impersonation by default.

To fix the above error, open **hive_site.xml** file in **HIVE_HOME\conf** location and add the following property between **<configuration>** and **</configuration>** tag.

```

<property>
    <name>hive.server2.enable.doAs</name>
    <value>false</value>
    <description>Enable user impersonation for HiveServer2</description>
</property>

```

```

D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\conf\hive-site.xml - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
hive-site.xml
1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xml" href="configuration.xsl"?>
3 <configuration>
4
5     <property>
6         <name>javax.jdo.option.ConnectionURL</name>
7         <value>jdbc:derby://localhost:1527/D:\ProgramFiles\Derby\db-derby-10.14.2.0-bin\data\metastore_db;create=true</value>
8         <description>JDBC connect string for a JDBC metastore</description>
9     </property>
10    <property>
11        <name>javax.jdo.option.ConnectionDriverName</name>
12        <value>org.apache.derby.jdbc.ClientDriver</value>
13        <description>Driver class name for a JDBC metastore</description>
14    </property>
15
16    <property>
17        <name>hive.server2.enable.doAs</name>
18        <value>false</value>
19        <description>Enable user impersonation for HiveServer2</description>
20    </property>
21
22 </configuration>
23

```

Stop and restart HiveServer2 service using the following command:

```
hiveserver2 --hiveconf hive.root.logger=console
```

Now, we are able to connect to Beeline in remote mode and run queries.

```

show databases;
use hive_local_derby_db;
show tables;
select * from employees;

```

```

C:\Users\hp>beeline -u jdbc:hive2://localhost:10000/ -n scott -p tiger
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://localhost:10000/
Connected to: Apache Hive (version 3.1.3)
Driver: Hive JDBC (version 3.1.3)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.3 by Apache Hive
0: jdbc:hive2://localhost:10000/ > show databases;
INFO : Compiling command(queryId=hp_20240601095141_c538099b-b625-4904-b86f-fe3cc17eb3d3): show databases
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldschemas:[fieldSchema(name:database_name, type:string, comment:from deserializer), properties:null])
INFO : Completed compiling command(queryId=hp_20240601095141_c538099b-b625-4904-b86f-fe3cc17eb3d3); Time taken: 1.427 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hp_20240601095141_c538099b-b625-4904-b86f-fe3cc17eb3d3): show databases
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hp_20240601095141_c538099b-b625-4904-b86f-fe3cc17eb3d3); Time taken: 0.142 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
+-----+
| database_name |
+-----+
| default      |
| hive_local_derby_db |
+-----+
2 rows selected (2.385 seconds)
0: jdbc:hive2://localhost:10000/ > !quit
Closing: 0: jdbc:hive2://localhost:10000/
C:\Users\hp>

```

9. Configure Remote MySQL Metastore:

Now, we will see how to configure MySQL as the remote metastore database.

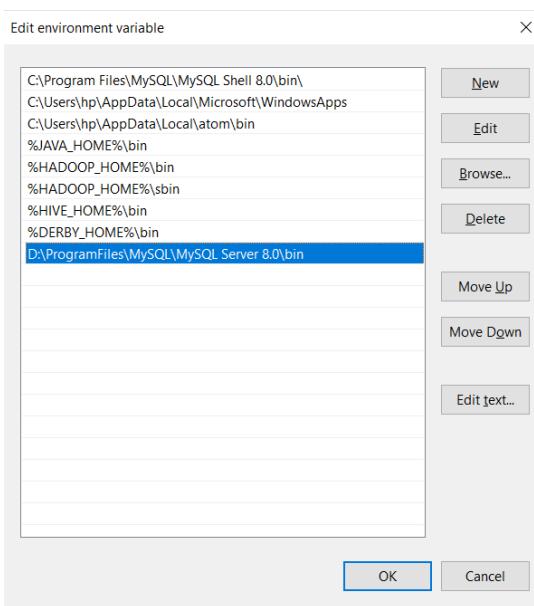
9.1. Install MySQL Server:

First, install MySQL server from the [official MySQL Downloads](#) website in the machine if was not already installed (*I installed the latest MySQL Server 8.0 version to D:\ProgramFiles\MySQL directory*)

Note that MySQL runs on port **3306** by default.

PC > Local Disk (D:) > ProgramFiles > MySQL > MySQL Server 8.0			
Name	Date modified	Type	Size
bin	5/29/2024 12:09 PM	File folder	
docs	5/29/2024 12:09 PM	File folder	
etc	5/29/2024 12:09 PM	File folder	
include	5/29/2024 12:09 PM	File folder	
lib	5/29/2024 12:09 PM	File folder	
share	5/29/2024 12:09 PM	File folder	
LICENSE	3/27/2024 7:22 PM	File	276 KB
LICENSE.router	3/27/2024 7:22 PM	ROUTER File	114 KB
README	3/27/2024 7:22 PM	File	1 KB
README.router	3/27/2024 7:22 PM	ROUTER File	1 KB

Open the **Environment Variables** dialog, select **Path** variable under **User variables** section and press **Edit** button. Press **New** and add MySQL Install Path\bin value (*for example, D:\ProgramFiles\MySQL\MySQL Server 8.0\bin*) and press OK. Then press OK again to apply environment variable changes and close window.



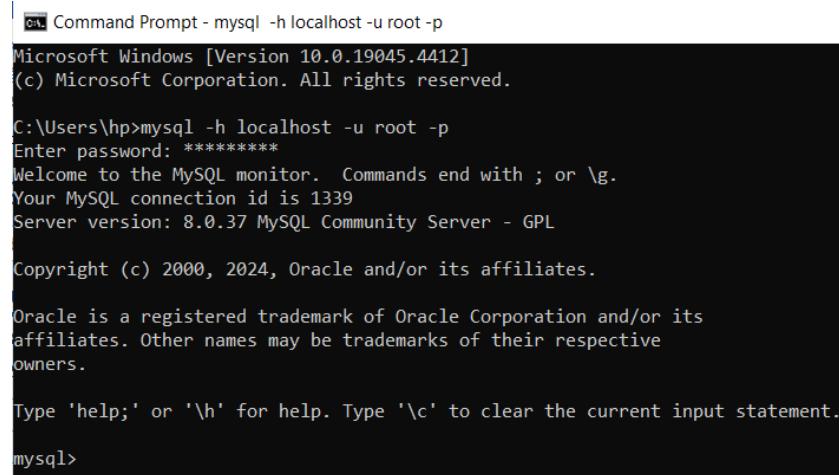
9.2. Create Metastore DB in MySQL:

In MySQL Server, create a database for Hive metastore.

Open MySQL command prompt or Workbench.

To launch mysql> command prompt with root user, execute this command:

```
mysql -h localhost -u root -p
```



```
mysql> Command Prompt - mysql -h localhost -u root -p
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>mysql -h localhost -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1339
Server version: 8.0.37 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Run the following queries to create a database named `hive_metastore` and a new user for Hive named `hive` and password as `HiveAdmin` in MySQL server.

```
CREATE DATABASE hive_metastore;
USE hive_metastore;
CREATE USER 'hive'@'localhost' IDENTIFIED WITH mysql_native_password BY 'HiveAdmin';
GRANT ALL PRIVILEGES ON hive_metastore.* TO 'hive'@'localhost';
FLUSH PRIVILEGES;
```

Note:

Make sure that `hive_metastore` database and `hive` user are not already available in your MySQL server. If they are already available, use a different database name and user name to create. Otherwise, you may encounter below errors:

`ERROR 1007 (HY000): Can't create database 'hive_metastore'; database exists`

`ERROR 1396 (HY000): Operation CREATE USER failed for 'hive'@'localhost'`

```

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
6 rows in set (0.00 sec)

mysql> CREATE DATABASE hive_metastore;
Query OK, 1 row affected (0.10 sec)

mysql> USE hive_metastore;
Database changed
mysql> CREATE USER 'hive'@'localhost' IDENTIFIED WITH mysql_native_password BY 'HiveAdmin';
Query OK, 0 rows affected (0.10 sec)

mysql> GRANT ALL PRIVILEGES ON hive_metastore.* TO 'hive'@'localhost';
Query OK, 0 rows affected (0.24 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.07 sec)

mysql> CREATE DATABASE hive_metastore;
ERROR 1007 (HY000): Can't create database 'hive_metastore'; database exists
mysql> exit;
Bye

C:\Users\hp>

```

9.3. Download MySQL JDBC Driver:

Since Hive does not provide the JDBC driver for MySQL by default, we need to explicitly get the driver and place it in `HIVE_HOME\lib` directory.

Download **JDBC Driver for MySQL (Connector/J)** from the below link:

<https://dev.mysql.com/downloads/connector/j/>

In the above link, choose **Platform Independent** Operating System and download `mysql-connector-j-* .zip` file as shown below:

The screenshot shows the MySQL Community Downloads page for the Connector/J 8.4.0 release. The user has selected "Platform Independent" as the operating system. Two download links are visible:

- Platform Independent (Architecture Independent), Compressed TAR Archive**: MySQL Connector/J 8.4.0, 4.1M, Download button, MD5: 33a74d1a803b504a1141a327bf6ff7c3 | Signature
- Platform Independent (Architecture Independent), ZIP Archive**: MySQL Connector/J 8.4.0, 4.9M, Download button, MD5: 5d426b639f7b4c314492edff07c615dc | Signature

The latest version of zip file at the time of writing this document is **mysql-connector-j-8.4.0.zip**. You can download the file of whichever latest version that you could see.

After downloading the zip file, unzip it which creates mysql-connector-j-* directory. Open the directory and copy mysql-connector-j-* .jar file to HIVE_HOME\lib directory.

is PC > New Volume (D:) > Programs > Hive > apache-hive-3.1.3-bin > lib			
Name	Date modified	Type	Size
metrics-graphite-3.1.0.jar	10/19/2019 3:05 AM	Executable Jar File	16 KB
metrics-json-3.1.0.jar	10/19/2019 3:05 AM	Executable Jar File	36 KB
metrics-jvm-3.1.0.jar	10/19/2019 3:05 AM	Executable Jar File	6 KB
minlog-1.3.0.jar	10/19/2019 3:09 AM	Executable Jar File	2,475 KB
mysql-connector-j-8.4.0.jar	3/13/2024 12:43 AM	Executable Jar File	11 KB
mysql-metadata-storage-0.12.0.jar	12/20/2019 4:21 AM	Executable Jar File	1,300 KB
netty-3.10.5.Final.jar	10/19/2019 3:05 AM	Executable Jar File	3,692 KB
netty-all-4.1.17.Final.jar	10/19/2019 3:05 AM	Executable Jar File	

9.4. Configure Hive Site File:

Now, we need to configure Metastore service to communicate with MySQL database. Open `hive-site.xml` file in `HIVE_HOME\conf` directory and replace the existing properties with the following properties between `<configuration>` and `</configuration>` element.

```
<property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://localhost:3306/hive_metastore</value>
    <description>JDBC connect string for a JDBC metastore</description>
</property>
<property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.jdbc.Driver</value>
    <description>Driver class name for a JDBC metastore</description>
</property>
<property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>hive</value>
    <description>Username to use against metastore database</description>
</property>
<property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>HiveAdmin</value>
    <description>password to use against metastore database</description>
</property>
<property>
    <name>hive.metastore.uris</name>
    <value>thrift://127.0.0.1:9083</value>
    <description>Thrift URI for the remote metastore. Used by metastore client to connect to remote metastore</description>
</property>
<property>
```

```

<name>hive.server2.enable.doAs</name>
<value>false</value>
<description>Enable user impersonation for HiveServer2</description>
</property>

```

```

1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3 <configuration>
4   <property>
5     <name>javax.jdo.option.ConnectionURL</name>
6     <value>jdbc:mysql://localhost:3306/hive_metastore</value>
7     <description>JDBC connect string for a JDBC metastore</description>
8   </property>
9   <property>
10    <name>javax.jdo.option.ConnectionDriverName</name>
11    <value>com.mysql.jdbc.Driver</value>
12    <description>Driver class name for a JDBC metastore</description>
13  </property>
14  <property>
15    <name>javax.jdo.option.ConnectionUserName</name>
16    <value>hive</value>
17    <description>Username to use against metastore database</description>
18  </property>
19  <property>
20    <name>javax.jdo.option.ConnectionPassword</name>
21    <value>HiveAdmin</value>
22    <description>password to use against metastore database</description>
23  </property>
24  <property>
25    <name>hive.metastore.uris</name>
26    <value>thrift://127.0.0.1:9083</value>
27    <description>Thrift URI for the remote metastore. Used by metastore client to connect to remote metastore</description>
28  </property>
29  <property>
30    <name>hive.server2.enable.doAs</name>
31    <value>false</value>
32    <description>Enable user impersonation for HiveServer2</description>
33  </property>
34 </configuration>
35
36

```

9.5. Initialize Metastore DB:

Now, run the schematool utility to create the initial metadata DB structure in MySQL database using the following command:

```
schematool -dbType mysql -initSchema
```

```

C:\Users\hp>schematool -dbType mysql -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:      jdbc:mysql://localhost:3306/hive_metastore
Metastore Connection Driver :  com.mysql.jdbc.Driver
Metastore connection User:     hive
Loading class 'com.mysql.jdbc.Driver'. This is deprecated. The new driver class is 'com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and manual loading of the driver class is generally unnecessary.
Starting metastore schema initialization to 3.1.0
Initialization script hive-schema-3.1.0.mysql.sql

Initialization script completed
schemaTool completed

C:\Users\hp>

```

9.6. Verify Metastore in MySQL:

Let us connect to MySQL server and verify Hive metastore created under `hive_metastore` database.

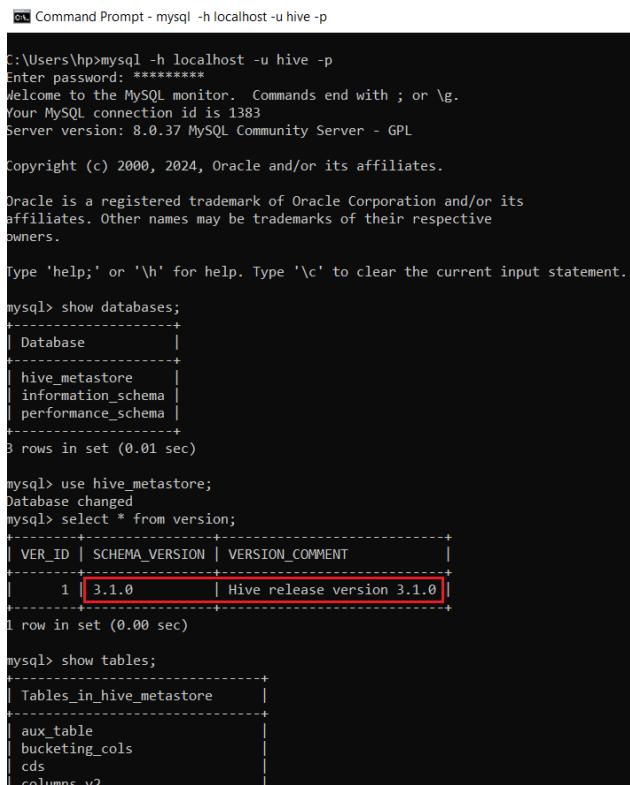
Launch MySQL command prompt using this command. Note that we are connecting with `hive` user credentials that we created earlier.

```
mysql -h localhost -u hive -p
```

Provide the password of `hive` user when asked.

In mysql> prompt, run the following queries:

```
show databases;
use hive_metastore;
select * from version;
show tables;
```



The screenshot shows a Command Prompt window titled "Command Prompt - mysql -h localhost -u hive -p". The MySQL monitor starts with standard welcome messages. Then, the user runs several commands:

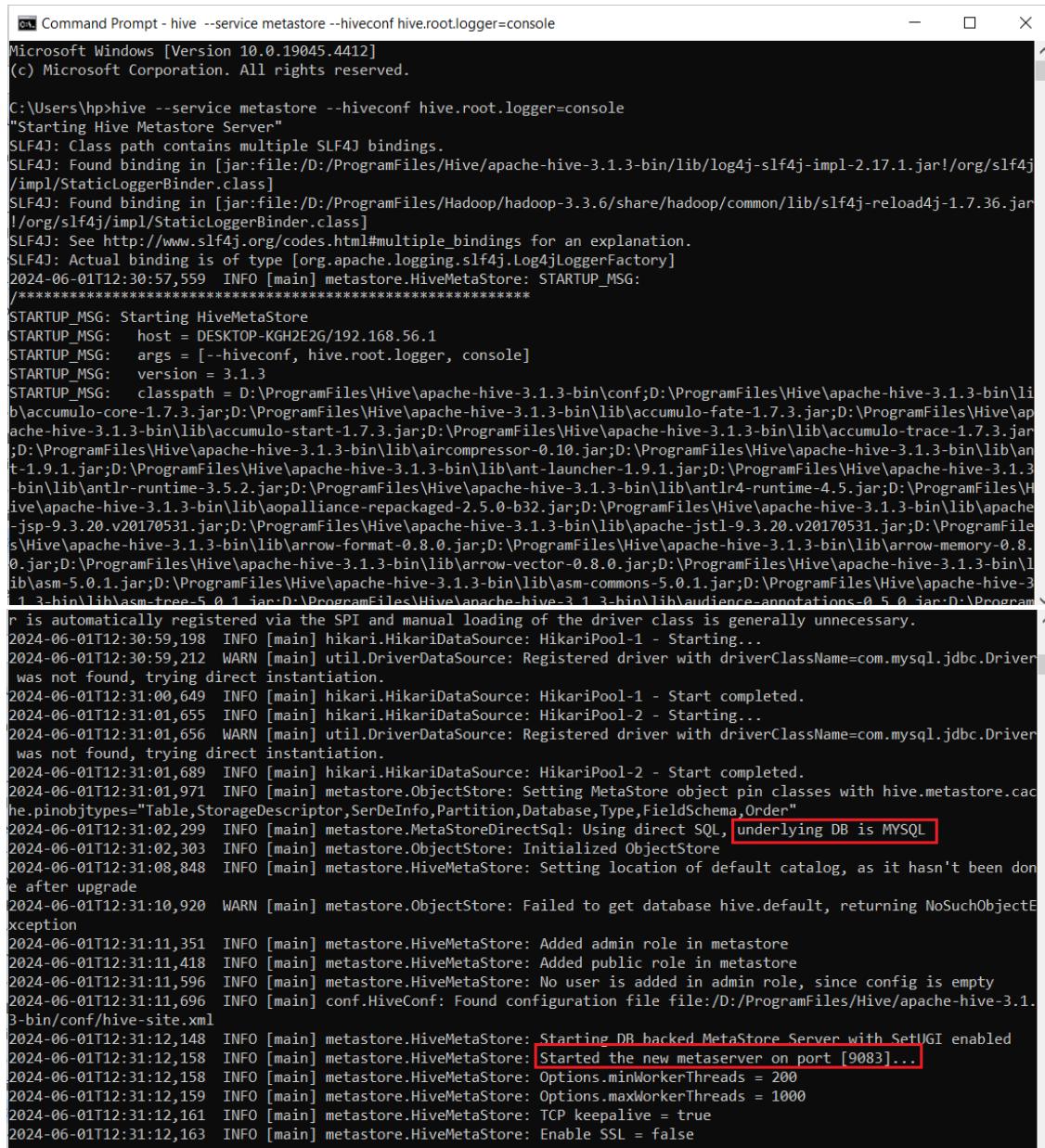
- `show databases;` displays the databases: `hive_metastore`, `information_schema`, and `performance_schema`.
- `use hive_metastore;` changes the database context to `hive_metastore`.
- `select * from version;` retrieves the version information, showing a single row with `VER_ID` as 1, `SCHEMA_VERSION` as 3.1.0, and `VERSION_COMMENT` as "Hive release version 3.1.0".
- `show tables;` lists the tables in the metastore: `Tables_in_hive_metastore`, `aux_table`, `bucketing_cols`, `cds`, and `columns_v2`.

Here, we can see that Hive metastore has created tables and the Hive release version **3.1.0**.

9.7. Start Hive Metastore Service:

Hive will be able to connect to remote metastore in MySQL database using Thrift URIs. So, let us start the metastore service using the below command and make sure it connects to our MySQL server.

```
hive --service metastore --hiveconf hive.root.logger=console
```



```
C:\Users\hp>hive --service metastore --hiveconf hive.root.logger=console
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>hive --service metastore --hiveconf hive.root.logger=console
"Starting Hive Metastore Server"
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]
2024-06-01T12:30:57,559 INFO [main] metastore.HiveMetaStore: STARTUP_MSG:
/*****STARTUP_MSG: Starting HiveMetaStore
STARTUP_MSG: host = DESKTOP-KGH2E2G/192.168.56.1
STARTUP_MSG: args = [--hiveconf, hive.root.logger, console]
STARTUP_MSG: version = 3.1.3
STARTUP_MSG: classpath = D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\conf;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\accumulo-core-1.7.3.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\accumulo-fate-1.7.3.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\accumulo-start-1.7.3.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\accumulo-trace-1.7.3.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\aircompressor-0.10.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\antlr-2.7.1.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\ant-launcher-1.9.1.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\antlr-runtime-3.5.2.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\antlr4-runtime-4.5.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\apoolalliance-repackaged-2.5.0-b32.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\apache-jsp-9.3.20.v20170531.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\apache-jstl-9.3.20.v20170531.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\arrow-format-0.8.0.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\arrow-memory-0.8.0.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\arrow-vector-0.8.0.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\asm-5.0.1.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\asm-commons-5.0.1.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\asm-tree-5.0.1.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\audience-annotations-0.5.0.jar;D:\Program
r is automatically registered via the SPI and manual loading of the driver class is generally unnecessary.
2024-06-01T12:30:59,198 INFO [main] hikari.HikariDataSource: HikariPool-1 - Starting...
2024-06-01T12:30:59,212 WARN [main] util.DriverDataSource: Registered driver with driverClassName=com.mysql.jdbc.Driver was not found, trying direct instantiation.
2024-06-01T12:31:00,649 INFO [main] hikari.HikariDataSource: HikariPool-1 - Start completed.
2024-06-01T12:31:01,655 INFO [main] hikari.HikariDataSource: HikariPool-2 - Starting...
2024-06-01T12:31:01,656 WARN [main] util.DriverDataSource: Registered driver with driverClassName=com.mysql.jdbc.Driver was not found, trying direct instantiation.
2024-06-01T12:31:01,689 INFO [main] hikari.HikariDataSource: HikariPool-2 - Start completed.
2024-06-01T12:31:01,971 INFO [main] metastore.ObjectStore: Setting MetaStore object pin classes with hive.metastore.cache.pinobjtypes="Table,StorageDescriptor,SerDeInfo,Partition,Database,Type,FieldSchema,Order"
2024-06-01T12:31:02,299 INFO [main] metastore.MetaStoreDirectSql: Using direct SQL, underlying DB is MYSQL
2024-06-01T12:31:02,303 INFO [main] metastore.ObjectStore: Initialized ObjectStore
2024-06-01T12:31:08,848 INFO [main] metastore.HiveMetaStore: Setting location of default catalog, as it hasn't been done after upgrade
2024-06-01T12:31:10,920 WARN [main] metastore.ObjectStore: Failed to get database hive.default, returning NoSuchObjectException
2024-06-01T12:31:11,351 INFO [main] metastore.HiveMetaStore: Added admin role in metastore
2024-06-01T12:31:11,418 INFO [main] metastore.HiveMetaStore: Added public role in metastore
2024-06-01T12:31:11,596 INFO [main] metastore.HiveMetaStore: No user is added in admin role, since config is empty
2024-06-01T12:31:11,696 INFO [main] conf.HiveConf: Found configuration file file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/conf/hive-site.xml
2024-06-01T12:31:12,148 INFO [main] metastore.HiveMetaStore: Starting DB backed MetaStore Server with SetUGI enabled
2024-06-01T12:31:12,158 INFO [main] metastore.HiveMetaStore: Started the new metaserver on port [9083]...
2024-06-01T12:31:12,158 INFO [main] metastore.HiveMetaStore: Options.minWorkerThreads = 200
2024-06-01T12:31:12,159 INFO [main] metastore.HiveMetaStore: Options.maxWorkerThreads = 1000
2024-06-01T12:31:12,161 INFO [main] metastore.HiveMetaStore: TCP keepalive = true
2024-06-01T12:31:12,163 INFO [main] metastore.HiveMetaStore: Enable SSL = false
```

On the console, we can see that **metaserver** started on port **9083** and connected to underlying MySQL metastore database.

9.8. Run Queries in Hive CLI:

Now, let us see if we can run queries in `hive` CLI.

Open new command prompt and start Hive CLI using the below command

```
hive
```

Run the following queries in Hive CLI to create a database, table and load data into it.

- Create a database named `hive_mysql_db`

```
show databases;  
create database hive_mysql_db;
```

```
hive> show databases;  
OK  
default  
Time taken: 1.712 seconds, Fetched: 1 row(s)  
hive> create database hive_mysql_db;  
OK  
Time taken: 0.63 seconds  
hive>
```

As soon as we run the above query, Hive has created `hive_mysql_db.db` directory in `/user/hive/warehouse` location in Hadoop File System.

```
hadoop fs -ls /user/hive/warehouse
```

```
C:\ Command Prompt  
Microsoft Windows [Version 10.0.19045.4412]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\hp>hadoop fs -ls /user/hive/warehouse  
Found 3 items  
drwxr-xr-x - hp supergroup 0 2024-06-01 08:27 /user/hive/warehouse/hive_embedded_derby_db.db  
drwxr-xr-x - hp supergroup 0 2024-06-01 09:11 /user/hive/warehouse/hive_local_derby_db.db  
drwxr-xr-x - hp supergroup 0 2024-06-01 12:35 /user/hive/warehouse/hive_mysql_db.db
```

The same is visible in NameNode UI: <http://localhost:9870/dfshealth.html>

In NameNode UI, go to **Utilities** tab and select **Browse the file system** option. Enter the directory name `/user/hive/warehouse` and you can see `hive_mysql_db.db` folder available.

Browse Directory

/user/hive/warehouse/

Show 25 entries

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	hp	supergroup	0 B	Jun 01 08:27	0	0 B	hive_embedded_derby_db.db
drwxr-xr-x	hp	supergroup	0 B	Jun 01 09:11	0	0 B	hive_local_derby_db.db
drwxr-xr-x	hp	supergroup	0 B	Jun 01 12:39	0	0 B	hive_mysql_db.db

Showing 1 to 3 of 3 entries

Hadoop, 2023.

- Create a table named employees_tmp in hive_mysql_db database. This table is created based on the columns data in the CSV file that we are going to load.

```
use hive_mysql_db;

create table employees_tmp(employee_id int, first_name string, last_name
string, email string, phone_number string, hire_date string, job_id string,
salary int, commission_pct int, manager_id int, department_id int) row format
delimited fields terminated by ',' tblproperties
('skip.header.line.count'='1');
```

```
hive> use hive_mysql_db;
OK
Time taken: 0.134 seconds
hive> create table employees_tmp(employee_id int, first_name string, last_name string, email string, phone_number string, hire_date string, job_id string, salary int, commission_pct int, manager_id int, department_id int) row format delimited fields terminated by ',' tblproperties ('skip.header.line.count'='1');
OK
Time taken: 1.34 seconds
hive>
```

On HDFS, employees_tmp folder is created in
 /user/hive/warehouse/hive_mysql_db.db location.

Command Prompt

```
C:\Users\hp>hadoop fs -ls /user/hive/warehouse/hive_mysql_db.db
Found 1 items
drwxr-xr-x  - hp supergroup          0 2024-06-01 12:48 /user/hive/warehouse/hive_mysql_db.db/employees_tmp
C:\Users\hp>
```

On NameNode UI, click on `hive_mysql_db` folder to see `employees_tmp` folder.

- Verify the format of the `employees_tmp` table created.

```
describe formatted employees_tmp;
```

```
hive> describe formatted employees_tmp;
OK
# col_name          data_type            comment
employee_id         int
first_name          string
last_name           string
email               string
phone_number        string
hire_date           string
job_id              string
salary              int
commission_pct      int
manager_id          int
department_id       int

# Detailed Table Information
Database:          hive_mysql_db
OwnerType:          USER
Owner:              hp
CreateTime:         Sat Jun 01 12:39:45 IST 2024
LastAccessTime:     UNKNOWN
Retention:          0
Location:          hdfs://localhost:9820/user/hive/warehouse/hive_mysql_db.db/employees_tmp
Table Type:         MANAGED_TABLE
Table Parameters:
  COLUMN_STATS_ACCURATE  {"BASIC_STATS":true,"COLUMN_STATS":{ "commission_pct":true,"department_id":true,"email":true,"employee_id":true,"first_name":true,"hire_date":true,"job_id":true,"last_name":true,"manager_id":true,"phone_number":true,"salary":true}}
  bucketing_version    2
  numFiles             0
  numRows              0
  rawDataSize         0
  skip.header.line.count 1
  totalSize            0
  transient_lastDdlTime 1717225785

# Storage Information
SerDe Library:      org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:         org.apache.hadoop.mapred.TextInputFormat
OutputFormat:        org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Compressed:          No
Num Buckets:        -1
Bucket Columns:     []
Sort Columns:        []
Storage Desc Params:
  field.delim          ,
  serialization.format ,
Time taken: 0.794 seconds, Fetched: 43 row(s)
hive>
```

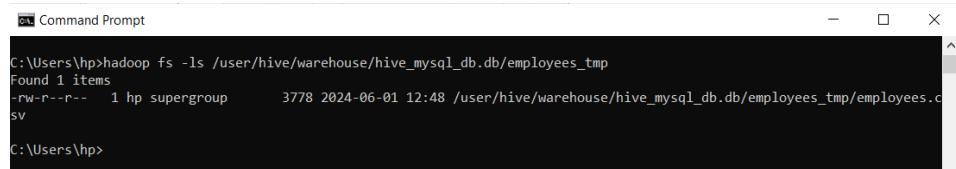
- Load data from `employees.csv` file located in the local directory `D:\Datasets` into the hive table `employees_tmp` in Hive CLI.

Note: This CSV file is available in [this location](#) that you can download and copy to `D:\Datasets` folder in your machine.

```
load data local inpath 'file:///D:\Datasets\employees.csv' into table employees_tmp;
```

```
hive> load data local inpath 'file:///D:\Datasets\employees.csv' into table employees_tmp;
Loading data to table hive_mysql_db.employees_tmp
OK
Time taken: 2.841 seconds
hive>
```

In HDFS, there is a `employees.csv` file created in `/user/hive/warehouse/hive_mysql_db.db/employees_tmp` location.



On NameNode UI, click on `employees_tmp` folder and you can see `employees.csv` file is created.

Name	Size	Last Modified	Replication	Block Size
employees.csv	3.69 KB	Jun 01 12:48	1	128 MB

- Select top 5 records in `employees_tmp` table from Hive CLI.

```
set hive.cli.print.header=true;
select * from employees_tmp limit 5;
```

```

hive> set hive.cli.print.header=true;
hive> select * from employees_tmp limit 5;
OK
+-----+-----+-----+-----+-----+-----+-----+
| employee_id | first_name | last_name | email | salary | commission_pct | phone_number |
+-----+-----+-----+-----+-----+-----+-----+
| 198 | Donald O'Connell | DOCONNEL | 650.507.9833 | 21-JUN-07 | SH_CLERK | 2600 | NULL | 124 | 50 |
| 199 | Douglas Grant DGRANT | 650.507.9844 | 13-JAN-08 | SH_CLERK | 2600 | NULL | 124 | 50 |
| 200 | Jennifer Whalen JWHALEN | 515.123.4444 | 17-SEP-03 | AD_ASST | 4400 | NULL | 101 | 10 |
| 201 | Michael Hartstein MHARTSTE | 515.123.5555 | 17-FEB-04 | MK_MAN | 13000 | NULL | 100 | 20 |
| 202 | Pat Fay PFAY | 603.123.6666 | 17-AUG-05 | MK_REP | 6000 | NULL | 201 | 20 |
+-----+-----+-----+-----+-----+-----+-----+
Time taken: 5.818 seconds, Fetched: 5 row(s)
hive>

```

We can verify the same in HDFS using hadoop cat command.

```

hadoop fs -cat
/usr/hive/warehouse/hive_mysql_db.db/employees_tmp/employees.csv

```

```

C:\Users\hp>hadoop fs -cat /user/hive/warehouse/hive_mysql_db.db/employees_tmp/employees.csv
EMPLOYEE_ID,FIRST_NAME,LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMMISSION_PCT,MANAGER_ID,DEPARTMENT_ID
198,Donald,O'Connell,DOCONNEL,650.507.9833,21-JUN-07,SH_CLERK,2600,-,124,50
199,Douglas,Grant,DGRANT,650.507.9844,13-JAN-08,SH_CLERK,2600,-,124,50
200,Jennifer,Whalen,JWHALEN,515.123.4444,17-SEP-03,AD_ASST,4400,-,101,10
201,Michael,Hartstein,MHARTSTE,515.123.5555,17-FEB-04,MK_MAN,13000,NULL,100,20
202,Pat,Fay,PFAY,603.123.6666,17-AUG-05,MK_REP,6000,-,201,20
203,Susan,Mavris,SMAVRIS,515.123.7777,07-JUN-02,HR_REP,6500,-,101,40
204,Hermann,Baer,HBAER,515.123.8888,07-JUN-02,PR_REP,10000,-,101,70
205,Shelley,Higgins,SHIGGINS,515.123.8080,07-JUN-02,AC_MGR,12008,-,101,110
206,William,Gietz,WGIETZ,515.123.8181,07-JUN-02,AC_ACCOUNT,8300,-,205,110
207,Steven,King,SKING,515.123.4567,17-JUN-03,AD_PRES,24000,-,-,90
208,Neena,Kochhar,NKOCHHAR,515.123.4568,21-SEP-05,AD_VP,17000,-,100,90
209,Lex,De Haan,LDEHAAN,515.123.4569,13-JAN-01,AD_VP,17000,-,100,90
210,Alexander,Hunold,AHUNOLD,590.423.4567,03-JAN-06,IT_PROG,9000,-,102,60

```

In the NameNode UI, click on `employees.csv` file and go to **Head the file** tab to see first few records or **Tail the file** tab to see last few records.

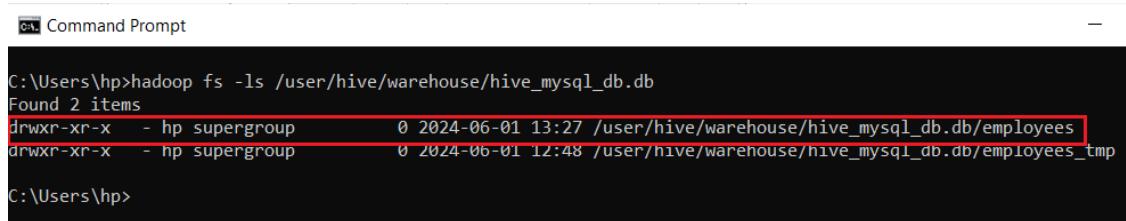
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
198	Donald	O'Connell	DOCONNEL	650.507.9833	21-JUN-07	SH_CLERK	2600	-	124	50
199	Douglas	Grant	DGRANT	650.507.9844	13-JAN-08	SH_CLERK	2600	-	124	50
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-03	AD_ASST	4400	-	101	10
201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-04	MK_MAN	13000	-	100	20
202	Pat	Fay	PFAY	603.123.6666	17-AUG-05	MK_REP	6000	-	201	20
203	Susan	Mavris	SMAVRIS	515.123.7777	07-JUN-02	HR_REP	6500	-	101	40

- Now, let us create a new table named `employees` similar to `employees_tmp` table except that data type of `hire_date` column would be `date` instead of `string`.

```
create table employees(employee_id int, first_name string, last_name string,
email string, phone_number string, hire_date date, job_id string, salary int,
commission_pct int, manager_id int, department_id int);
```

```
hive> create table employees(employee_id int, first_name string, last_name string, email string, phone_number string, hire_date date, job_id string, salary int, commission_pct int, manager_id int, department_id int);
OK
Time taken: 0.399 seconds
hive>
```

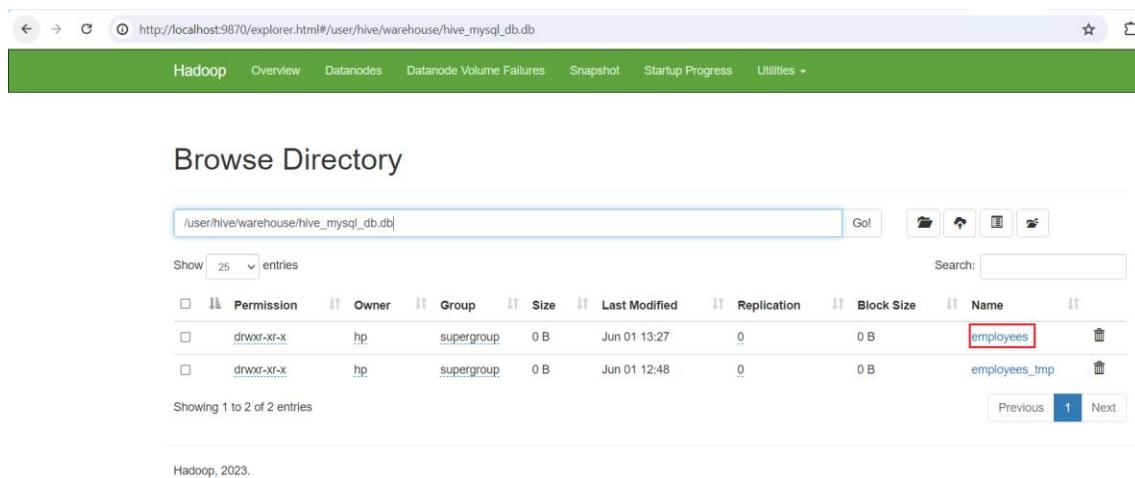
On HDFS, `employees` folder is created in
`/user/hive/warehouse/hive_mysql_db.db` location.



```
C:\Users\hp>hadoop fs -ls /user/hive/warehouse/hive_mysql_db.db
Found 2 items
drwxr-xr-x  - hp supergroup          0 2024-06-01 13:27 /user/hive/warehouse/hive_mysql_db.db/employees
drwxr-xr-x  - hp supergroup          0 2024-06-01 12:48 /user/hive/warehouse/hive_mysql_db.db/employees_tmp

C:\Users\hp>
```

On NameNode UI, go to `/user/hive/warehouse/hive_mysql_db.db` folder to see `employees` folder.



Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
<code>drwxr-xr-x</code>	hp	supergroup	0 B	Jun 01 13:27	0	0 B	<code>employees</code>
<code>drwxr-xr-x</code>	hp	supergroup	0 B	Jun 01 12:48	0	0 B	<code>employees_tmp</code>

- Load data into `employees` table from `employees_tmp` table by transforming `hire_date` column in `employees_tmp` table into the Hive accepted date format.

```
insert into employees select employee_id, first_name, last_name, email,
phone_number, from_unixtime(unix_timestamp(hire_date,'dd-MMM-yy'), 'yyyy-MM-
dd') as hire_date, job_id, salary, commission_pct, manager_id, department_id
from employees_tmp;
```

```
hive> insert into employees select employee_id, first_name, last_name, email, phone_number, from_unixtime(unix_timestamp(hire_date,'d-
d-MMM-yy'), 'yyyy-MM-dd') as hire_date, job_id, salary, commission_pct, manager_id, department_id from employees_tmp;
Query ID = hp_20240601132656_39fb8d1f-0b7b-4879-a67d-2eb335d14388
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1717215281098_0001, Tracking URL = http://DESKTOP-KGH2E2G:8088/proxy/application_1717215281098_0001/
Kill Command = D:\ProgramFiles\Hadoop\hadoop-3.3.6\bin\mapred job -kill job_1717215281098_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2024-06-01 13:27:28,907 Stage-1 map = 0%, reduce = 0%
2024-06-01 13:27:41,073 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 6.467 sec
MapReduce Total cumulative CPU time: 6 seconds 467 msec
Ended Job = job_1717215281098_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9820/user/hive/warehouse/hive_mysql_db.db/employees/.hive-staging_hive_2024-06-01_13-26-57_
011_6165538390635873943-1/-ext-10000
Loading data to table hive_mysql_db.employees
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Cumulative CPU: 6.467 sec HDFS Read: 11731 HDFS Write: 3742 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 467 msec
OK
_time _col0 _col1 _col2 _col3 _col4 _col5 _col6 _col7 _col8 _col9 _col10
Time taken: 47.208 seconds
hive>
```

The above query submitted a MapReduce job to Hadoop YARN which we can track using the application tracking URL provided.

Open Hadoop YARN UI: <http://localhost:8088/cluster> where we can see the application has been submitted which is running the application “*insert into employees .. from employees_tmp (Stage-1)*”

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State
application_1717215281098_0001	hp	insert into employees .. from employees_tmp (Stage-1)	MAPREDUCE		default	0	Sat Jun 1 13:27:06 +0550 2024	Sat Jun 1 13:27:09 +0550 2024	Sat Jun 1 13:27:41 +0550 2024	FINISHED

After the above query is completed, we can see a file named `000000_0` created in `/user/hive/warehouse/hive_mysql_db.db/employees` HDFS location.

```
C:\Users\hp>hadoop fs -ls /user/hive/warehouse/hive_mysql_db.db/employees
Found 1 items
-rw-r--r-- 1 hp supergroup 3661 2024-06-01 13:27 /user/hive/warehouse/hive_mysql_db.db/employees/000000_0
C:\Users\hp>
```

On NameNode UI, click on employees folder to see 000000_0 file.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hp	supergroup	3.58 KB	Jun 01 13:27	1	128 MB	000000_0

- Select top 5 records in employees table in Hive CLI.

```
select * from employees limit 5;
```

```
hive> select * from employees limit 5;
OK
employees.employee_id    employees.first_name    employees.last_name    employees.email    employees.phone_number    employees.hire_date
employees.job_id          employees.salary      employees.commission_pct   employees.manager_id   employees.department_id
198    Donald OConnell    DOCNEL             650.507.9833    2007-06-21    SH_CLERK           2600    NULL    124    50
199    Douglas Grant     DGRANT            650.507.9844    2008-01-13    SH_CLERK           2600    NULL    124    50
200    Jennifer Whalen   JWHALEN           515.123.4444    2003-09-17    AD_ASST            4400    NULL    101    10
201    Michael Hartstein MHARTSTE          515.123.5555    2004-02-17    MK_MAN            13000   NULL    100    20
202    Pat Fay            PFAY              603.123.6666    2005-08-17    MK_REP             6000    NULL    201    20
Time taken: 0.388 seconds, Fetched: 5 row(s)
hive>
```

We can verify the same in HDFS using hadoop cat command.

```
hadoop fs -cat /user/hive/warehouse/hive_mysql_db.db/employees/000000_0
```

```
C:\Users\hp>hadoop fs -ls /user/hive/warehouse/hive_mysql_db.db/employees
Found 1 items
-rw-r--r-- 1 hp supergroup      3661 2024-06-01 13:27 /user/hive/warehouse/hive_mysql_db.db/employees/000000_0

C:\Users\hp>hadoop fs -cat /user/hive/warehouse/hive_mysql_db.db/employees/000000_0
1980Donald@OCONNELL@DOCONNEL@650.507.9833@2007-06-21@SH_CLERK@26000\N@124@50
1990Douglas@Grant@DGRANT@650.507.9844@2008-01-13@SH_CLERK@26000\N@124@50
2000Jennifer@Whalen@JWHALEN@515.123.4444@2003-09-17@AD_ASST@44000\N@101@10
2010Michael@Hartstein@MHARTSTE@515.123.5555@2004-02-17@MK_MAN@13000\N@100@20
2020Pat@Fay@PFAY@603.123.6666@2005-08-17@MK_REP@60000\N@201@20
2030Susan@Mavris@SMAVRIS@515.123.7777@2002-06-07@HR_REP@65000\N@101@40
2040Hermann@Baer@HBAER@515.123.8888@2002-06-07@PR_REP@100000\N@101@70
2050Shelley@Higgins@SHIGGINS@515.123.8080@2002-06-07@AC_MGR@120080\N@101@110
2060William@Gietz@WGIETZ@515.123.8181@2002-06-07@AC_ACCOUNT@83000\N@205@110
1000Steven@King@SKING@515.123.4567@2003-06-17@AD_PRES@240000\N@N@90
```

In the NameNode UI, click on `000000_0` file and go to **Head the file** tab to see first few records.

The screenshot shows the Apache NameNode UI interface. A modal window titled "File information - 000000_0" is open. At the top of the modal, there are three buttons: "Download", "Head the file (first 32K)" (which is highlighted with a red border), and "Tail the file (last 32K)". Below this, there's a "Block information" section for Block 0, which includes details like Block ID, Block Pool ID, Generation Stamp, Size, Availability, and Owner. The "File contents" section displays the first 32KB of the file, showing the 10 records from the terminal output. The main HDFS browser interface is visible in the background, showing the directory structure and other files.

- Verify if the count of records between `employees_tmp` and `employees` tables are matching.

Count records in employees_tmp table:

```
select count(*) from employees_tmp;
```

```

hive> select count(*) from employees_tmp;
Query ID = hp_20240601140142_127e8a2b-a8d9-4dfb-8bfb-52ff41b80985
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1717215281098_0003, Tracking URL = http://DESKTOP-KGH2E2G:8088/proxy/application_1717215281098_0003/
Kill Command = D:\ProgramFiles\Hadoop\hadoop-3.3.6\bin\mapred job -kill job_1717215281098_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-06-01 14:02:00,708 Stage-1 map = 0%, reduce = 0%
2024-06-01 14:02:11,362 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.482 sec
2024-06-01 14:02:22,912 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 11.182 sec
MapReduce Total cumulative CPU time: 11 seconds 182 msec
Ended Job = job_1717215281098_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1  Cumulative CPU: 11.182 sec  HDFS Read: 18041 HDFS Write: 102 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 182 msec
OK
_c0
50
Time taken: 41.338 seconds, Fetched: 1 row(s)
hive>

```

It displayed the count as 50.

The above query submitted a job to Hadoop YARN and provided us a tracking URL which can be tracked in YARN UI: <http://localhost:8088/cluster/>

The screenshot shows the Hadoop YARN UI at <http://localhost:8088/cluster>. The left sidebar has sections for Cluster (About, Nodes, Node Labels, Applications, Scheduler, Tools), and Applications (NEW, NEW_SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED). The main area displays Cluster Metrics and Application Metrics. Under Application Metrics, three applications are listed:

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State
application_1717215281098_0003	hp	select count(*) from employees_tmp (Stage-1)	MAPREDUCE		default	0	Sat Jun 1 14:01:44 2024	Sat Jun 1 14:01:45 +0550 2024	Sat Jun 1 14:02:22 +0550 2024	FINISHED
application_1717215281098_0002	hp	Insert into employees select...employeesTmp (Stage-1)	MAPREDUCE		default	0	Sat Jun 1 13:50:11 2024	Sat Jun 1 13:50:11 +0550 2024	Sat Jun 1 13:50:54 +0550 2024	FINISHED
application_1717215281098_0001	hp	Insert into employees select...employeesTmp (Stage-1)	MAPREDUCE		default	0	Sat Jun 1 13:27:06 2024	Sat Jun 1 13:27:09 +0550 2024	Sat Jun 1 13:27:41 +0550 2024	FINISHED

Showing 1 to 3 of 3 entries

Count records in employees table:

```
select count(*) from employees;
```

```

hive> select count(*) from employees;
OK
_c0
50
Time taken: 0.34 seconds, Fetched: 1 row(s)
hive>

```

It displayed the count as 50.

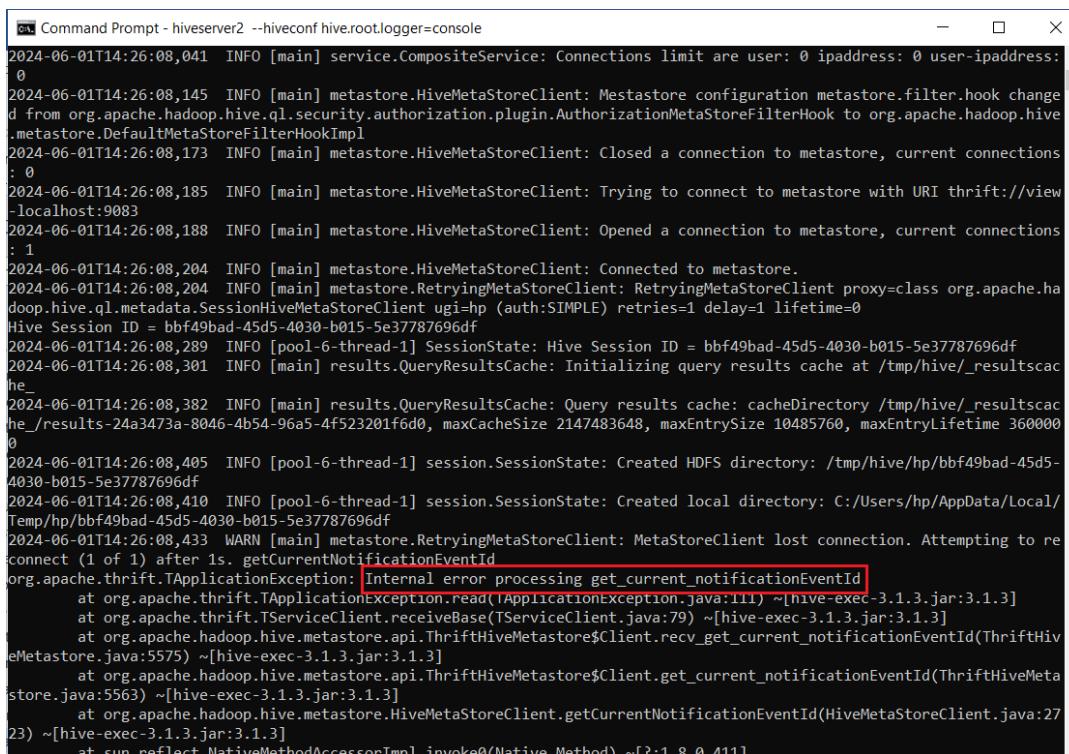
9.9. Start HiveServer2 Service:

To start Beeline in remote mode, HiveServer2 service must be running.

If HiveServer2 is already running, stop it and start it freshly because we made configuration changes in `hive_site.xml` that should be loaded into HiveServer2.

Open command prompt and start the HiveServer2 service with this command:

```
hiveserver2 --hiveconf hive.root.logger=console
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt - hiveserver2 --hiveconf hive.root.logger=console". The window displays a series of log messages from the HiveServer2 service. The messages include information about metastore connections, session IDs, and a warning about a lost connection attempt. A specific error message is highlighted in red: "org.apache.thrift.TApplicationException: Internal error processing get_current_notificationEventId". This error is further detailed with a stack trace pointing to several Java class files and line numbers, such as "TServiceClient.java:79" and "ThriftHiveMetastore\$Client.recv_get_current_notificationEventId(ThriftHiveMetastore.java:5575)".

```
2024-06-01T14:26:08,041 INFO [main] service.CompositeService: Connections limit are user: 0 ipaddress: 0 user-ipaddress: 0
2024-06-01T14:26:08,145 INFO [main] metastore.HiveMetaStoreClient: Metastore configuration metastore.filter.hook changed from org.apache.hadoop.hive.ql.security.authorization.plugin.AuthorizationMetaStoreFilterHook to org.apache.hadoop.hive.metastore.DefaultMetaStoreFilterHookImpl
2024-06-01T14:26:08,173 INFO [main] metastore.HiveMetaStoreClient: Closed a connection to metastore, current connections : 0
2024-06-01T14:26:08,185 INFO [main] metastore.HiveMetaStoreClient: Trying to connect to metastore with URI thrift://view-localhost:9083
2024-06-01T14:26:08,188 INFO [main] metastore.HiveMetaStoreClient: Opened a connection to metastore, current connections : 1
2024-06-01T14:26:08,204 INFO [main] metastore.HiveMetaStoreClient: Connected to metastore.
2024-06-01T14:26:08,204 INFO [main] metastore.RetryingMetaStoreClient: RetryingMetaStoreClient proxy=class org.apache.hadoop.hive.ql.metadata.SessionHiveMetaStoreClient ugi=hp (auth:SIMPLE) retries=1 delay=1 lifetime=0
Hive Session ID = bbf49bad-45d5-4030-b015-5e37787696df
2024-06-01T14:26:08,289 INFO [pool-6-thread-1] SessionState: Hive Session ID = bbf49bad-45d5-4030-b015-5e37787696df
2024-06-01T14:26:08,301 INFO [main] results.QueryResultsCache: Initializing query results cache at /tmp/hive/_resultscache_
2024-06-01T14:26:08,382 INFO [main] results.QueryResultsCache: Query results cache: cacheDirectory /tmp/hive/_resultscache/_results-24a3473a-8046-4b54-96a5-4f523201f6d0, maxCacheSize 2147483648, maxEntrySize 10485760, maxEntryLifetime 3600000
2024-06-01T14:26:08,405 INFO [pool-6-thread-1] session.SessionState: Created HDFS directory: /tmp/hive/hive/bbf49bad-45d5-4030-b015-5e37787696df
2024-06-01T14:26:08,410 INFO [pool-6-thread-1] session.SessionState: Created local directory: C:/Users/hp/AppData/Local/Tmp/hive/bbf49bad-45d5-4030-b015-5e37787696df
2024-06-01T14:26:08,433 WARN [main] metastore.RetryingMetaStoreClient: MetaStoreClient lost connection. Attempting to reconnect (1 of 1) after 1s. get_current_notificationEventId
org.apache.thrift.TApplicationException: Internal error processing get_current_notificationEventId
    at org.apache.thrift.TApplicationException.read(TApplicationException.java:111) ~[hive-exec-3.1.3.jar:3.1.3]
    at org.apache.thrift.TServiceClient.receiveBase(TServiceClient.java:79) ~[hive-exec-3.1.3.jar:3.1.3]
    at org.apache.hadoop.hive.metastore.api.ThriftHiveMetastore$Client.recv_get_current_notificationEventId(ThriftHiveMetastore.java:5575) ~[hive-exec-3.1.3.jar:3.1.3]
    at org.apache.hadoop.hive.metastore.api.ThriftHiveMetastore$Client.get_current_notificationEventId(ThriftHiveMetastore.java:5563) ~[hive-exec-3.1.3.jar:3.1.3]
    at org.apache.hadoop.hive.metastore.HiveMetaStoreClient.getCurrentNotificationEventId(HiveMetaStoreClient.java:273) ~[hive-exec-3.1.3.jar:3.1.3]
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) ~[?:1.8.0_411]
```

During HiveServer2 startup, we will encounter error “**Internal error processing get_current_notificationEventId**” and in the metastore logs (*go to the window where metastore service was started*), we can see an additional error “**User xxx is not allowed to perform this API call**”.

```

Command Prompt - hive --service metastore --hiveconf hive.root.logger=console
2024-06-01T14:20:28,175 INFO [pool-8-thread-30] HiveMetaStore.audit: ugi=hp ip=127.0.0.1 cmd=Done cleaning up thread local RawStore
2024-06-01T14:20:28,187 INFO [pool-8-thread-30] metastore.HiveMetaStore: 31: Done cleaning up thread local RawStore
2024-06-01T14:20:28,204 INFO [pool-8-thread-30] HiveMetaStore.audit: ugi=hp ip=127.0.0.1 cmd=Done cleaning up thread local RawStore
2024-06-01T14:20:28,206 ERROR [pool-8-thread-31] metastore.HiveMetaStore: Not authorized to make the get_current_notificationEventId call. You can try to disable metastore.metastore.event.db.notification.api.auth
org.apache.hadoop.hive.metastore.api.MetaException: User hp is not allowed to perform this API call
    at org.apache.hadoop.hive.metastore.HiveMetaStore$HMSHandler.authorizeProxyPrivilege(HiveMetaStore.java:7528) ~[hive-exec-3.1.3.jar:3.1.3]
    at org.apache.hadoop.hive.metastore.HiveMetaStore$HMSHandler.get_current_notificationEventId(HiveMetaStore.java:7488) ~[hive-exec-3.1.3.jar:3.1.3]
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) ~[:1.8.0_411]
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) ~[:1.8.0_411]
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[:1.8.0_411]
    at java.lang.reflect.Method.invoke(Method.java:498) ~[:1.8.0_411]
    at org.apache.hadoop.hive.metastore.RetryingHMSHandler.invokeInternal(RetryingHMSHandler.java:147) [hive-exec-3.1.3.jar:3.1.3]
    at org.apache.hadoop.hive.metastore.RetryingHMSHandler.invoke(RetryingHMSHandler.java:108) [hive-exec-3.1.3.jar:3.1.3]
    at com.sun.proxy.$Proxy33.get_current_notificationEventId(Unknown Source) [?:?]
    at org.apache.hadoop.hive.metastore.api.ThriftHiveMetastore$Processor$get_current_notificationEventId.getResult(ThriftHiveMetastore.java:18364) [hive-exec-3.1.3.jar:3.1.3]
    at org.apache.hadoop.hive.metastore.api.ThriftHiveMetastore$Processor$get_current_notificationEventId.getResult(ThriftHiveMetastore.java:18349) [hive-exec-3.1.3.jar:3.1.3]
    at org.apache.thrift.ProcessFunction.process(ProcessFunction.java:39) [hive-exec-3.1.3.jar:3.1.3]
    at org.apache.hadoop.hive.metastore.TUGIBasedProcessor$1.run(TUGIBasedProcessor.java:111) [hive-exec-3.1.3.jar:3.1.3]
    at org.apache.hadoop.hive.metastore.TUGIBasedProcessor$1.run(TUGIBasedProcessor.java:107) [hive-exec-3.1.3.jar:3.1.3]
    at java.security.AccessController.doPrivileged(Native Method) ~[:1.8.0_411]
    at javax.security.auth.Subject.doAs(Subject.java:422) [?:1.8.0_411]
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1899) [hadoop-common-3.3.6.jar:?:?]
    at org.apache.hadoop.hive.metastore.TUGIBasedProcessor.process(TUGIBasedProcessor.java:119) [hive-exec-3.1.3.jar:3.1.3]
...

```

This is because the superuser does not have access to impersonate as Hive user. To resolve this issue, we need to configure proxy user in Hadoop's `core-site.xml` file as below:

- Open `core-site.xml` file available in `%HADOOP_HOME%\etc\hadoop` location and add the following additional properties. In this configuration, replace `$superuser` with your user name displayed in the above error (*in my case, I am running Hadoop with hp user and the same is being displayed in the above error*).

```

<property>
    <name>hadoop.proxyuser.$superuser.hosts</name>
    <value>*</value>
</property>
<property>
    <name>hadoop.proxyuser.$superuser.groups</name>
    <value>*</value>
</property>

```

```
D:\ProgramFiles\Hadoop\hadoop-3.3.6\etc\hadoop\core-site.xml - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window .
core-site.xml x
16      <!-- Put site-specific property overrides in this file. -->
17
18      <configuration>
19          <property>
20              <name>fs.default.name</name>
21              <value>hdfs://localhost:9820</value>
22              <description>HDFS NameNode URL</description>
23          </property>
24
25          <!-- Configuration for proxy user -->
26          <property>
27              <name>hadoop.proxyuser.hp.hosts</name>
28              <value>*</value>
29          </property>
30
31          <property>
32              <name>hadoop.proxyuser.hp.groups</name>
33              <value>*</value>
34
35      </configuration>
```

- After the above configuration change is done, restart Hadoop services first. Open **Windows Command Prompt** in **Administrator** mode and run the following commands.

```
stop-dfs.cmd
stop-yarn.cmd
start-dfs.cmd
start-yarn.cmd
```

- When Hadoop services are up and running, close and start Hive Metastore and HiveServer2 using these commands.

```
hive --service metastore --hiveconf hive.root.logger=console
hiveserver2 --hiveconf hive.root.logger=console
```

```

: 0
2024-06-01T14:39:07,734 INFO [main] metastore.HiveMetaStoreClient: Trying to connect to metastore with URI thrift://view-localhost:9083
2024-06-01T14:39:07,734 INFO [main] metastore.HiveMetaStoreClient: Opened a connection to metastore, current connections: 1
2024-06-01T14:39:07,740 INFO [main] metastore.HiveMetaStoreClient: Connected to metastore.
2024-06-01T14:39:07,740 INFO [main] metastore.RetryingMetaStoreClient: RetryingMetaStoreClient proxy=class org.apache.hadoop.hive.ql.metadata.SessionHiveMetaStoreClient ugi=hp (auth:SIMPLE) retries=1 delay=1 lifetime=0
Hive Session ID = 76097d25-8778-4c58-ac00-c82c93cdac4e
2024-06-01T14:39:07,971 INFO [pool-6-thread-1] SessionState: Hive Session ID = 76097d25-8778-4c58-ac00-c82c93cdac4e
2024-06-01T14:39:07,982 INFO [main] results.QueryResultsCache: Initializing query results cache at /tmp/hive/_resultscache_
2024-06-01T14:39:08,022 INFO [main] results.QueryResultsCache: Query results cache: cacheDirectory /tmp/hive/_resultscache/_results-0b54d762-cd03-4004-9b1a-ecef7acaac7d, maxCacheSize 2147483648, maxEntrySize 10485760, maxEntryLifetime 3600000
2024-06-01T14:39:08,038 INFO [pool-6-thread-1] session.SessionState: Created HDFS directory: /tmp/hive/hp/76097d25-8778-4c58-ac00-c82c93cdac4e
2024-06-01T14:39:08,050 INFO [pool-6-thread-1] session.SessionState: Created local directory: C:/Users/hp/AppData/Local/Temp/hp/76097d25-8778-4c58-ac00-c82c93cdac4e
2024-06-01T14:39:08,091 INFO [pool-6-thread-1] session.SessionState: Created HDFS directory: /tmp/hive/hp/76097d25-8778-4c58-ac00-c82c93cdac4e/_tmp_space.db
2024-06-01T14:39:08,133 INFO [main] events.NotificationEventPoll: Initializing lastCheckedEventId to 0
2024-06-01T14:39:08,244 INFO [main] util.log: Logging initialized @14969ms
2024-06-01T14:39:08,261 INFO [pool-6-thread-1] metadata.HiveMaterializedViewsRegistry: Materialized views registry has been initialized
2024-06-01T14:39:08,624 INFO [main] thrift.ThriftCLIService: Starting ThriftBinaryCLIService on port 10000 with 5...500 worker threads
2024-06-01T14:39:08,628 INFO [main] server.Server: jetty-9.3.20.v20170531
2024-06-01T14:39:08,904 INFO [main] handler.ContextHandler: Started o.e.j.w.WebApplicationContext@110b7837{/file:///C:/Users/hp/AppData/Local/Temp/jetty-0.0.0.0-10002-hiveserver2-_any-6632097667708750539.dir/webapp/,AVAILABLE}{jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/hive-service-3.1.3.jar!/hive-webapps/hiveserver2}
2024-06-01T14:39:08,905 INFO [main] handler.ContextHandler: Started o.e.j.s.ServletContextHandler@6ee88e21{/static,jar:/file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/hive-service-3.1.3.jar!/hive-webapps/static,AVAILABLE}
2024-06-01T14:39:08,918 INFO [main] server.AbstractConnector: Started ServerConnector@740a0d5e[HTTP/1.1,[http/1.1]}{0.0.0.1:10002}
2024-06-01T14:39:08,918 INFO [main] server.Server: Started @15646ms
2024-06-01T14:39:08,918 INFO [main] http.HttpServer: Started HttpServer[hiveserver2] on port 10002
2024-06-01T14:40:08,232 INFO [NotificationEventPoll 0] metastore.HiveMetaStoreClient: Trying to connect to metastore with URI thrift://view-localhost:9083

```

Here, we can see that HiveServer2 service connected to metastore running over Thrift URI `thrift://localhost:9083` and started on port 10002 by default and `ThriftBinaryCLIService` started on port 10000.

9.10. Beeline CLI:

Now, let us connect to Beeline in Remote mode as anonymous user.

```
beeline -u jdbc:hive2://localhost:10000/
```

```

: 0
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>beeline -u jdbc:hive2://localhost:10000/
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://localhost:10000/
Connected to: Apache Hive (version 3.1.3)
Driver: Hive JDBC (version 3.1.3)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.3 by Apache Hive
0: jdbc:hive2://localhost:10000/>

```

This connected to HiveServer2 running over Thrift URI on 10000 port and connected to MySQL database internally.

After beeline is connected, we should be able to execute queries as we do in Hive CLI.

```
show databases;
use hive_mysql_db;
set hive.cli.print.header=true;
select * from employees limit 5;
```

```
0: jdbc:hive2://localhost:10000/> select * from employees limit 5;
INFO : Compiling command(queryId=hp_20240601144719_684189c8-2936-4b25-9525-ccf8c8e85014): select * from employees limit 5
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema{fieldschemas:[FieldSchema(name:employees.employee_id, type:int, comment:null), FieldSchema(name:employees.first_name, type:string, comment:null), FieldSchema(name:employees.last_name, type:string, comment:null), FieldSchema(name:employees.email, type:string, comment:null), FieldSchema(name:employees.phone_number, type:string, comment:null), FieldSchema(name:employees.hire_date, type:date, comment:null), FieldSchema(name:employees.job_id, type:string, comment:null), FieldSchema(name:employees.salary, type:int, comment:null), FieldSchema(name:employees.commission_pct, type:int, comment:null), FieldSchema(name:employees.manager_id, type:int, comment:null), FieldSchema(name:employees.department_id, type:int, comment:null)], properties=null}
INFO : Completed compiling command(queryId=hp_20240601144719_684189c8-2936-4b25-9525-ccf8c8e85014); Time taken: 5.019 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hp_20240601144719_684189c8-2936-4b25-9525-ccf8c8e85014): select * from employees limit 5
INFO : Completed executing command(queryId=hp_20240601144719_684189c8-2936-4b25-9525-ccf8c8e85014); Time taken: 0.001 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager
+-----+-----+-----+-----+-----+-----+
| employees.employee_id | employees.first_name | employees.last_name | employees.email | employees.phone_number | e
| employees.hire_date | employees.job_id | employees.salary | employees.commission_pct | employees.manager_id | employ
| employees.department_id |
+-----+-----+-----+-----+-----+-----+
| 198 | Donald | O'Connell | NULL | DOCONNEL | 650.507.9833 | 2
| 007-06-21 | SH_CLERK | 2600 | NULL | 124 | 50 |
| 199 | Douglas | Grant | NULL | DGRANT | 650.507.9844 | 2
| 008-01-13 | SH_CLERK | 2600 | NULL | 124 | 50 |
| 200 | Jennifer | Whalen | NULL | JWHALEN | 515.123.4444 | 2
| 003-09-17 | AD_ASST | 4400 | NULL | 101 | 10 |
| 201 | Michael | Hartstein | NULL | MHARTSTE | 515.123.5555 | 2
| 004-02-17 | MK_MAN | 13000 | NULL | 100 | 20 |
| 202 | Pat | Fay | NULL | PFAY | 603.123.6666 | 2
| 005-08-17 | MK_REP | 6000 | NULL | 201 | 20 |
```

9.11. Verify Metadata in MySQL:

Let us connect to MySQL server and verify Hive metadata of database and tables created in MySQL.

Launch MySQL command prompt using this command. Note that we are connecting with `hive` user credentials that we created earlier.

```
mysql -h localhost -u hive -p
```

Provide the password of `hive` user when asked.

In `mysql>` prompt, run the following queries:

- Get the metadata of `hive_mysql_db` database.

```
use hive_metastore;
select * from dbs;
```

```

mysql> use hive_metastore;
Database changed
mysql> select * from dbs;
+-----+-----+-----+-----+-----+
| DB_ID | DESC          | DB_LOCATION_URI           | NAME      | OWNER_NAME |
+-----+-----+-----+-----+
| 1    | Default Hive database | hdfs://localhost:9820/user/hive/warehouse | default   | public     |
| 2    | NULL             | hdfs://localhost:9820/user/hive/warehouse/hive_mysql_db.db | hive_mysql_db | hp         |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>

```

It shows the DB location URI as

`hdfs://localhost:9820/user/hive/warehouse/hive_mysql_db.db`.

- Get the metadata of `employees_tmp` and `employees` tables and its columns.

```

select * from tbls;
select * from columns_v2;

```

```

mysql> select * from tbls;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| TBL_ID | CREATE_TIME | DB_ID | LAST_ACCESS_TIME | OWNER | OWNER_TYPE | RETENTION | SD_ID | TBL_NAME | TBL_TYPE |
| VIEW_EXPANDED_TEXT | VIEW_ORIGINAL_TEXT | IS_REWRITE_ENABLED |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 1717225785 | 2 | 0 | hp | USER | 0 | 1 | employees_tmp | MANAGED_TABLE |
| 2 | 1717228589 | 2 | 0 | hp | USER | 0 | 2 | employees | MANAGED_TABLE |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from columns_v2;
+-----+-----+-----+-----+-----+
| CD_ID | COMMENT | COLUMN_NAME | TYPE_NAME | INTEGER_IDX |
+-----+-----+-----+-----+-----+
| 1 | NULL | commission_pct | int | 8 |
| 1 | NULL | department_id | int | 10 |
| 1 | NULL | email | string | 3 |
| 1 | NULL | employee_id | int | 0 |
| 1 | NULL | first_name | string | 1 |
| 1 | NULL | hire_date | string | 5 |
| 1 | NULL | job_id | string | 6 |
| 1 | NULL | last_name | string | 2 |
| 1 | NULL | manager_id | int | 9 |
| 1 | NULL | phone_number | string | 4 |
| 1 | NULL | salary | int | 7 |
| 2 | NULL | commission_pct | int | 8 |
| 2 | NULL | department_id | int | 10 |
| 2 | NULL | email | string | 3 |
| 2 | NULL | employee_id | int | 0 |
| 2 | NULL | first_name | string | 1 |
| 2 | NULL | hire_date | date | 5 |
| 2 | NULL | job_id | string | 6 |
| 2 | NULL | last_name | string | 2 |
| 2 | NULL | manager_id | int | 9 |
| 2 | NULL | phone_number | string | 4 |
| 2 | NULL | salary | int | 7 |
+-----+-----+-----+-----+-----+
22 rows in set (0.00 sec)

mysql>

```

It shows `employees_tmp` and `employees` tables and their respective columns.

10. Hive Web UI:

Hives provides the following web interface to monitor HiveServer2 service.

HiveServer2 UI: <http://localhost:10002/>

The screenshot shows the Apache Hive Web UI at <http://localhost:10002/>. The top navigation bar includes links for Home, Local logs, Metrics Dump, Hive Configuration (which is highlighted with a red border), Stack Trace, and Log Daemons. The main content area is titled "HiveServer2".

Active Sessions

User Name	IP Address	Operation Count	Active Time (s)	Idle Time (s)
anonymous	127.0.0.1	0	791	570

Total number of sessions: 1

Open Queries

User Name	Query	Execution Engine	State	Opened Timestamp	Opened (s)	Latency (s)	Drilldown Link
anonymous	select * from employees limit 5	mr	FINISHED	6	Sat Jun 01 14:47:25 IST 2024	5	Drilldown
anonymous	use hive_mysql_db	mr	FINISHED	0	Sat Jun 01 14:46:59 IST 2024	0	Drilldown
anonymous	show databases	mr	FINISHED	2	Sat Jun 01 14:46:41 IST 2024	2	Drilldown

Total number of queries: 0

Last Max 25 Closed Queries

User Name	Query	Execution Engine	State	Opened (s)	Closed Timestamp	Latency (s)	Drilldown Link
anonymous	select * from employees limit 5	mr	FINISHED	6	Sat Jun 01 14:47:25 IST 2024	5	Drilldown
anonymous	use hive_mysql_db	mr	FINISHED	0	Sat Jun 01 14:46:59 IST 2024	0	Drilldown
anonymous	show databases	mr	FINISHED	2	Sat Jun 01 14:46:41 IST 2024	2	Drilldown

Total number of queries: 3

Software Attributes

Attribute Name	Value	Description
Hive Version	3.1.3, r4df4d75bf1e16fe0af75aad0b4179c34c07fc975	Hive version and revision
Hive Compiled	Sun Apr 3 16:58:16 EDT 2022, ngangam	When Hive was compiled and by whom
HiveServer2 Start Time	Sat Jun 01 14:56:55 IST 2024	Date stamp of when this HiveServer2 was started

Since we last connected to Beeline as anonymous user, it shows the **User Name** as anonymous .

It also displays the last 25 queries executed on HiveServer2 after the latest restart. We can also view Hive configuration, stack trace and much more in this UI.

11. HCatalog and WebHCat:

HCatalog is a table and storage management layer for Hadoop that allows **MapReduce**, **Pig** and **Hive** users to easily read and write data on HDFS. HCatalog provides a relational view of data stored in HDFS. It is built on top of the Hive metastore and incorporates Hive's DDL. It provides read and write interfaces for Pig and MapReduce and uses Hive's command line interface for issuing data definition and metadata exploration commands.

WebHCat (previously called as **Templeton**) is the REST API provided to access HCatalog service. Unlike HCatalog, which executed the command directly, WebHCat keeps the Hive, PIG, and MapReduce jobs in queues. The jobs can then be monitored and stopped as needed.

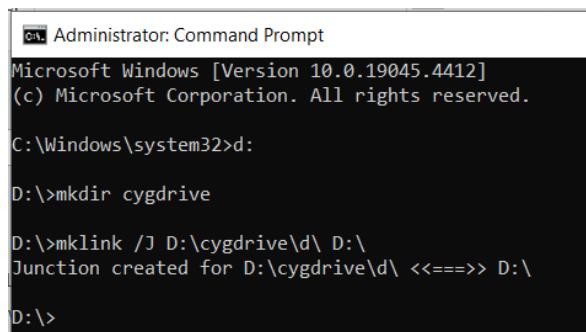
Note: We cannot start **HCatalog** and **WebHCat** server from Windows since they are not compatible for Windows and so, we will use Cygwin utility which allows executing Linux commands from Windows.

11.1. Create Symbolic Link for Cygwin:

Since Java cannot understand Cygwin paths properly, we will first create symbolic links for cygdrive to use Cygwin utility.

Open command prompt in Administrator mode and run the following commands:

```
d:  
mkdir cygdrive  
mklink /J D:\cygdrive\d\ D:\
```



The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The window displays the following text:

```
Administrator: Command Prompt  
Microsoft Windows [Version 10.0.19045.4412]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Windows\system32>d:  
  
D:>mkdir cygdrive  
  
D:>mklink /J D:\cygdrive\d\ D:\  
Junction created for D:\cygdrive\d\ <<==>> D:\  
  
D:>
```

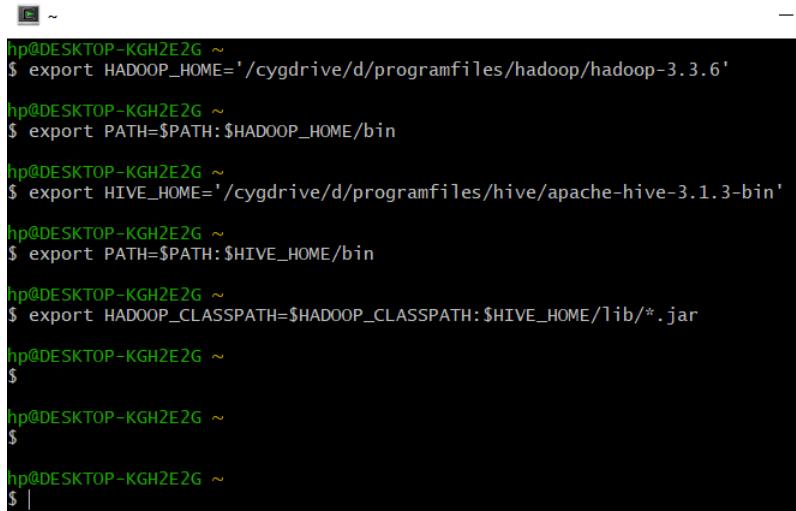
Note that we created `cygdrive` directory in `D` drive since we installed Hive in this drive. If you have installed Hive in a different drive, create symbolic link to that drive.

11.2. Setup Env variables for Cygwin:

Now, open **Cygwin64 Terminal** and run the following commands to define environment variables. We can add the below lines to `~/.bashrc` file so we don't need to execute every time we open Cygwin.

```
export HADOOP_HOME='/cygdrive/d/programfiles/hadoop/hadoop-3.3.6'  
export PATH=$PATH:$HADOOP_HOME/bin  
export HIVE_HOME='/cygdrive/d/programfiles/hive/apache-hive-3.1.3-bin'  
export PATH=$PATH:$HIVE_HOME/bin  
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$HIVE_HOME/lib/*.jar
```

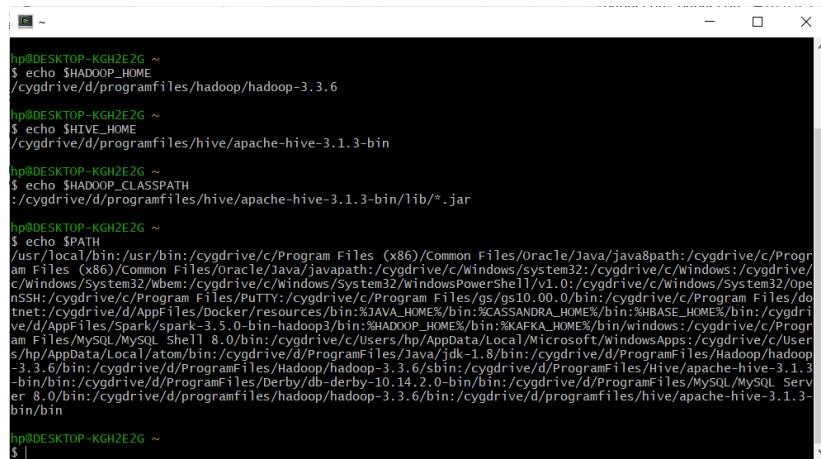
Make sure the above HADOOP_HOME and HIVE_HOME variables refer to the exact location where Hadoop and Hive were installed.



```
hp@DESKTOP-KGH2E2G ~
$ export HADOOP_HOME='/cygdrive/d/programfiles/hadoop/hadoop-3.3.6'
hp@DESKTOP-KGH2E2G ~
$ export PATH=$PATH:$HADOOP_HOME/bin
hp@DESKTOP-KGH2E2G ~
$ export HIVE_HOME='/cygdrive/d/programfiles/hive/apache-hive-3.1.3-bin'
hp@DESKTOP-KGH2E2G ~
$ export PATH=$PATH:$HIVE_HOME/bin
hp@DESKTOP-KGH2E2G ~
$ export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$HIVE_HOME/lib/*.jar
hp@DESKTOP-KGH2E2G ~
$
hp@DESKTOP-KGH2E2G ~
$
```

Verify if the above variables are properly set with these commands

```
echo $HADOOP_HOME
echo $HIVE_HOME
echo $HADOOP_CLASSPATH
echo $PATH
```



```
hp@DESKTOP-KGH2E2G ~
$ echo $HADOOP_HOME
/cygdrive/d/programfiles/hadoop/hadoop-3.3.6
hp@DESKTOP-KGH2E2G ~
$ echo $HIVE_HOME
/cygdrive/d/programfiles/hive/apache-hive-3.1.3-bin
hp@DESKTOP-KGH2E2G ~
$ echo $HADOOP_CLASSPATH
:/cygdrive/d/programfiles/hive/apache-hive-3.1.3-bin/lib/*.jar
hp@DESKTOP-KGH2E2G ~
$ echo $PATH
/usr/local/bin:/usr/bin:/cygdrive/c/Program Files (x86)/Common Files/Oracle/Java/java8path:/cygdrive/c/Program Files (x86)/Common Files/Oracle/Java/javapath:/cygdrive/c/Windows/system32:/cygdrive/c/windows:/cygdrive/c/windows/system32/wbem:/cygdrive/c/windows/system32/windowspowershell/v1.0:/cygdrive/c/windows/system32/openSSH:/cygdrive/c/Program Files/PuTTY:/cygdrive/c/Program Files/gs/gs10.00.0/bin:/cygdrive/c/Program Files/dotnet:/cygdrive/c/Program Files/dotnet/resources/bin:$JAVA_HOME/bin:$CASSANDRA_HOME/bin:$BASE_HOME/bin:/cygdrive/c/Program Files/Spark/spark-3.5.0-bin-hadoop3/bin:$HADOOP_HOME/bin:$AFKA_HOME/bin/windows:/cygdrive/c/Program Files/MySQL/MySQL Shell 8.0/bin:/Cygdrive/c/Users/hp/AppData/Local/Microsoft/WindowsApps:/cygdrive/c/users/hp/AppData/Local/atom/bin:/cygdrive/d/ProgramFiles/Java/jdk-1.8/bin:/cygdrive/d/ProgramFiles/Hadoop/hadoop-3.3.6/bin:/cygdrive/d/ProgramFiles/Hadoop/hadoop-3.3.6/sbin:/cygdrive/d/ProgramFiles/Hive/apache-hive-3.1.3-bin/bin:/cygdrive/d/ProgramFiles/derby/db-derby-10.14.2.0-bin/bin:/cygdrive/d/ProgramFiles/MySQL/MySQL Server 8.0/bin:/cygdrive/d/programfiles/hadoop/hadoop-3.3.6/bin:/cygdrive/d/programfiles/hive/apache-hive-3.1.3-bin/bin
```

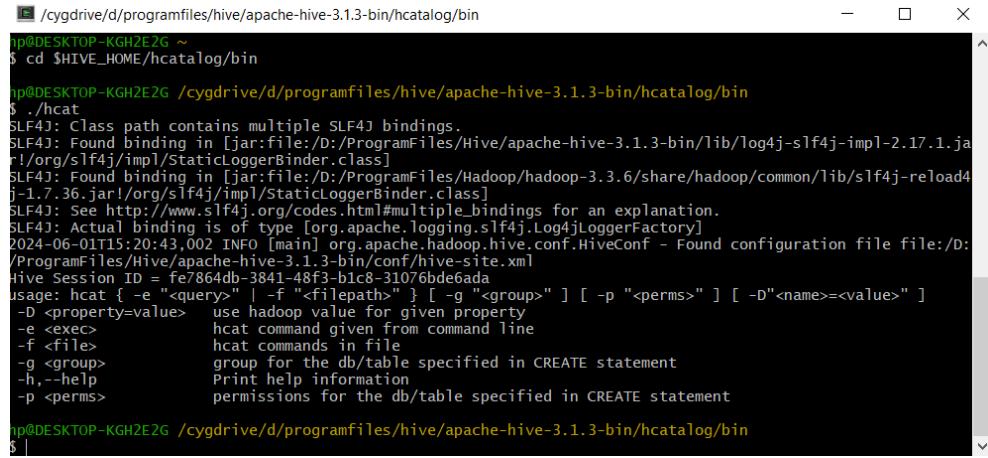
11.3. Start HCatalog CLI:

We can start the HCatalog command line interface by using hcat file available in HIVE_HOME\hcatalog\bin

In Cygwin, run the following commands to start HCatalog:

```
cd $HIVE_HOME/hcatalog/bin
```

```
./hcat
```



```
[~/cygdrive/d/programfiles/hive/apache-hive-3.1.3-bin/hcatalog/bin] hp@DESKTOP-KGH2E2G ~
$ cd $HIVE_HOME/hcatalog/bin
hp@DESKTOP-KGH2E2G ~/cygdrive/d/programfiles/hive/apache-hive-3.1.3-bin/hcatalog/bin
$ ./hcat
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]
2024-06-01T15:20:43,002 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Found configuration file file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/conf/hive-site.xml
Hive Session ID = fe7864db-3841-48f3-b1c8-31076bde6ada
Usage: hcat { -e "<query>" | -f "<filepath>" } [ -g "<group>" ] [ -p "<perms>" ] [ -D"<name>=<value>" ]
-D <property=value>      use hadoop value for given property
-e <exec>                 hcat command given from command line
-f <file>                  hcat commands in file
-g <group>                group for the db/table specified in CREATE statement
-h,--help                  Print help information
-p <perms>                permissions for the db/table specified in CREATE statement
hp@DESKTOP-KGH2E2G ~/cygdrive/d/programfiles/hive/apache-hive-3.1.3-bin/hcatalog/bin
$ |
```

Note that `hcat` commands can be issued as `hive` commands. HCatalog CLI supports the following command line options:

-g : Tells HCatalog that the table which needs to be created must have group `mygroup`.

For example:

```
hcat -g mygroup ...
```

-p : Tells HCatalog that the table which needs to be created must have permissions

"`rwxr-xr-x`".

For example:

```
hcat -p rwxr-xr-x ...
```

-f : Tells HCatalog that `myscript.hcatalog` is a file containing DDL commands to execute.

For example:

```
hcat -f myscript.hcatalog ...
```

-e : Tells HCatalog to treat the following string as a DDL command and execute it

For example:

```
hcat -e 'create table mytable(a int);' ...
```

-D : Passes the key-value pair to HCatalog as a Java System Property.

For example:

```
hcat -Dkey=value ...
```

hcat: Prints a usage message.

For example:

Run the following hcat command to create employee_sample table in hive_mysql_db database:

```
./hcat -e 'create table hive_mysql_db.employee_sample(emp_id int, emp_name string)'
```

```
hp@DESKTOP-KGH2E2G /cygdrive/d/programfiles/hive/apache-hive-3.1.3-bin/hcatalog/bin
$ ./hcat -e 'create table hive_mysql_db.employee_sample(emp_id int, emp_name string)'

SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/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.apache.logging.slf4j.Log4jLoggerFactory]
2024-06-01T15:23:25,546 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Found configuration file file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/conf/hive-site.xml
Hive Session ID = c1630590-5d59-4448-957f-fd22794b2967
OK
Time taken: 7.108 seconds

hp@DESKTOP-KGH2E2G /cygdrive/d/programfiles/hive/apache-hive-3.1.3-bin/hcatalog/bin
$ |
```

Run these queries in Hive CLI, to see the newly created table.

```
use hive_mysql_db;
show tables
```

```
hive> use hive_mysql_db;
OK
Time taken: 1.257 seconds
hive> show tables;
OK
tab_name
employee_sample
employees
employees_tmp
Time taken: 0.124 seconds, Fetched: 3 row(s)
hive>
```

11.4. Start WebHCat Server:

Now, start the **WebHCat** server using the following command in Cygwin:

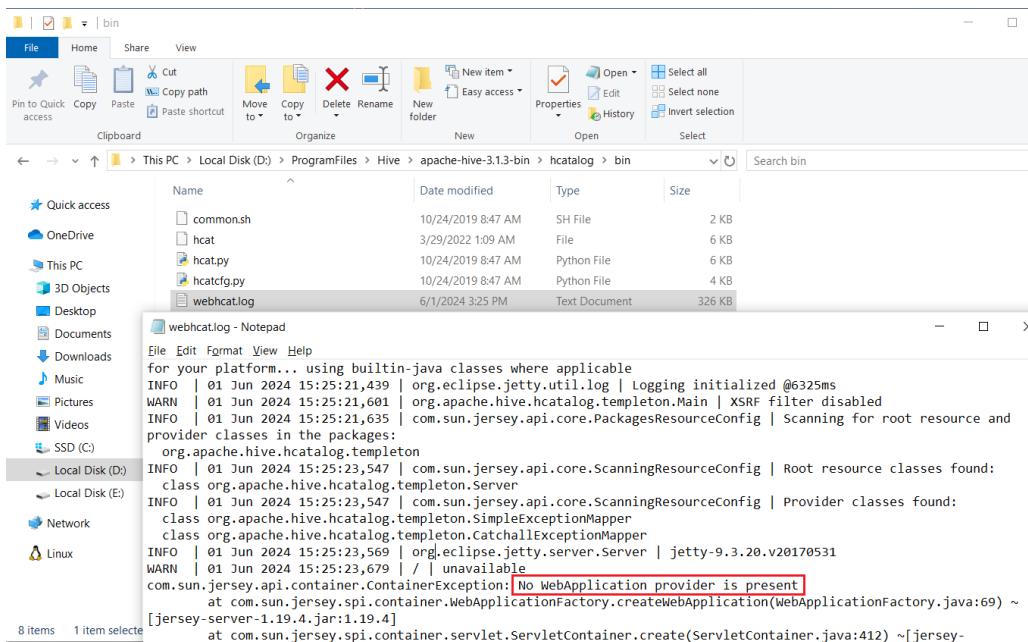
```
$HIVE_HOME/hcatalog/sbin/webhcatt_server.sh start
```

```

/cydrive/d/programfiles/hive/apache-hive-3.1.3-bin/hcatalog/bin
time taken: 7.108 seconds
$HIVE_HOME/hcatalog/sbin/webhcatalog_server.sh start
Length of string is non zero
webhcatalog: starting ...
webhcatalog: /cydrive/d/programfiles/hadoop-3.3.6/bin/hadoop jar /cydrive/d/programfiles/hive/apache-hive-3.1.3-bin/hcatalog/sbin/../share/webhcatalog/srv/lib/hive-webhcatalog-3.1.3.jar org.apache.hive.hcatalog.templeton.Main
webhcatalog: starting ... started.
webhcatalog: done

```

Though it says “webchat started” on console, but in `webchat.log` (available in the location from where ever `webhcatalog_server.sh` script was executed), there is an error “**Server failed to start: No WebApplication provider is present**” that you can see.



This is because some configuration is needed before starting WebHCat server. Follow the [Apache Hive documentation](#) on how to setup WebHCat.

Once WebHCat service is successfully running, HCatlog resources can be accessed by REST APIs using the URI format: <http://localhost:50111/templeton/v1/<resource>> (For example: <http://localhost:50111/templeton/v1/status>)

Congratulations!! You have now successfully installed and configured Hive with 3 metastore modes in Windows operating system. You also got a glimpse of HCatlog and WebHCat components.