

Apache HBase 2.6 Installation on Windows OS

Table of Contents

1. Overview:	4
2. Prerequisites:	4
3. Install Standalone HBase:	5
3.1. Download HBase Binaries:	5
3.2. Download Hadoop Binaries:	7
3.3. Set up Environment variables:	8
3.4. Configure HBase:	12
3.4.1. HBase Site Configuration:	12
3.4.2. HBase Env Configuration:	13
3.5. Verify HBase Installation:	14
3.6. Start HBase:	15
3.7. HBase Shell:	18
4. HBase Shell Commands:	21
4.1. HBase Status:	24
4.2. HBase Version:	24
4.3. Process List:	25
4.4. List Namespace:	25
4.5. Create Namespace:	25
4.6. Describe Namespace:	26
4.7. Create Table:	26
4.8. List Table:	27
4.9. Describe Table:	27
4.10. Insert Data into Table:	28
4.11. Read Data from Table:	29
4.12. Disable Table:	32
4.13. Enable Table:	33
4.14. Delete Rows from Table:	34
4.15. Drop Table:	36
5. HBase User Interface:	37

5.1.	Master Web UI:	37
5.2.	Region Web UI:	38
6.	Access HBase from Hive:	39
6.1.	Verify Hadoop Installation:	39
6.2.	Verify Hive Installation:	39
6.3.	Verify Hive HBase Handler:	40
6.4.	Start Hadoop Services:	40
6.5.	Start Hive Metastore Service:	43
6.6.	Launch Hive Shell:	44
6.7.	Create HBase Table:	46
6.8.	Insert Data into HBase Table:	47
6.9.	Validate Newly Created Table in HBase:	53
6.10.	Create External HBase Table:	54
6.11.	Fetch HBase Data in Hive:	55
7.	Shutdown Standalone HBase:	56

This document outlines the steps needed to install **Apache HBase 2.6** on Windows Operating system and access HBase database from **Apache Hive**.

1. Overview:

Apache HBase is the Hadoop database which is a distributed, scalable storage system for Big Data. It is an open source non-relational database modeled after Google's Bigtable to store and fast access the large amount of Big Data on top of HDFS.

Google's **Bigtable** is a distributed storage system provided by Google File System for managing structured data of petabytes size across thousands of commodity servers whereas **Apache HBase** is a distributed storage system for managing non-relational data (structured, semi-structured and unstructured data) on top of Hadoop Distributed File System.

HBase can be installed in three different modes:

1. **Standalone Mode:** In Standalone Mode, HBase daemons – HMaster, HRegionServer and ZooKeeper – runs within a single JVM (Java Virtual Machine) process /instance on a single host with HBase data stored on local file system. This is the most basic deployment mode.
2. **Pseudo-Distributed Mode:** In this mode, HBase still runs completely on a single host but each HBase daemon such as HMaster, HRegionServer and ZooKeeper runs as a separate process. This mode allows us to configure HBase data storage either on local file system or on HDFS. This deployment mode is for testing purposes only but not for Production.
3. **Fully-Distributed Mode:** In this mode, HBase cluster contains multiple nodes or hosts and each node can run one or more HBase daemons including primary and backup Master instances, multiple RegionServer nodes and multiple ZooKeeper nodes. This mode can only run on top of HDFS which makes a fully distributed configuration for Production and is used in real-time scenarios.

2. Prerequisites:

The following prerequisites need to be installed before running HBase.

1. **File Archiver:** Any file archiver such as **7zip** or **WinRAR** is needed to unzip the downloaded Spark binaries. 7zip can be downloaded from the [7zip Downloads](#) website and WinRAR can be downloaded from the [RAR lab Downloads](#) website.

2. **JRE 8:** HBase 3.x requires Java 8 runtime environment (*HBase 2.3+ supports Java 8 mainly and Java 11 with limited capability. Other Java versions are not supported*). See the [HBase Java support page](#) for more details.

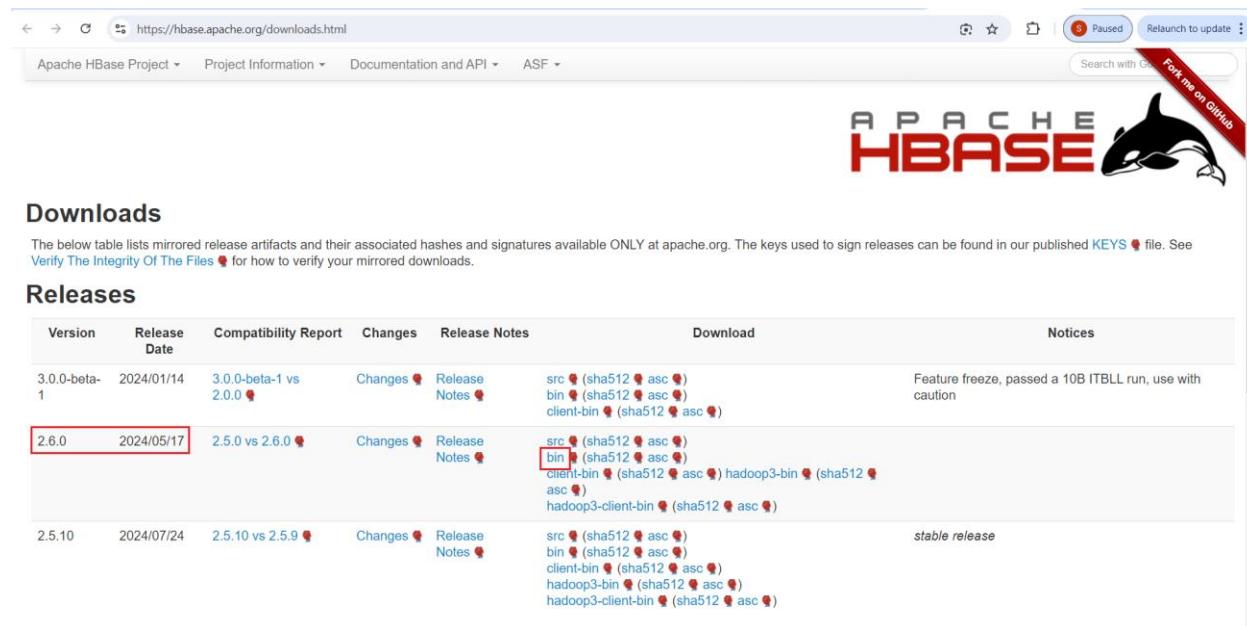
We can either download just JRE 8 (Java Runtime Environment) for Windows offline installation from the official [Java Download for Windows Offline](#) website or download the whole JDK 8 (Java Development Kit) directly from [Oracle Java Downloads](#) website. For the complete JDK installation steps, look at [here](#).

3. Install Standalone HBase:

Let us see how HBase can be installed in standalone mode without Hadoop requirement.

3.1. Download HBase Binaries:

Download the stable version of HBase from the official [Apache HBase Downloads](#) website. At the time of this document preparation, the most recent stable release is 2.6.0. Select the latest version and click on **bin** link as shown below (Check [here](#) for previous versions of HBase).



The screenshot shows the Apache HBase Downloads page. The header features the Apache logo and the HBase logo with an orca. Below the header, there's a search bar and a GitHub fork button. The main content area has sections for 'Downloads' and 'Releases'.

Downloads

The below table lists mirrored release artifacts and their associated hashes and signatures available ONLY at apache.org. The keys used to sign releases can be found in our published KEYS file. See [Verify The Integrity Of The Files](#) for how to verify your mirrored downloads.

Releases

Version	Release Date	Compatibility Report	Changes	Release Notes	Download	Notices
3.0.0-beta-1	2024/01/14	3.0.0-beta-1 vs 2.0.0	Changes	Release Notes	src (sha512 asc) bin (sha512 asc) client-bin (sha512 asc)	Feature freeze, passed a 10B ITBLL run, use with caution
2.6.0	2024/05/17	2.5.0 vs 2.6.0	Changes	Release Notes	src (sha512 asc) bin (sha512 asc) client-bin (sha512 asc) hadoop3-bin (sha512 asc) hadoop3-client-bin (sha512 asc)	
2.5.10	2024/07/24	2.5.10 vs 2.5.9	Changes	Release Notes	src (sha512 asc) bin (sha512 asc) client-bin (sha512 asc) hadoop3-bin (sha512 asc) hadoop3-client-bin (sha512 asc)	stable release

You will be navigated to [Hbase 2.6.0 mirror website](#) where click on the suggested location for [hbase-2.6.0-bin.tar.gz](#) file that gets downloaded to your **Downloads** folder in your machine.

The screenshot shows the Apache Software Foundation website at https://www.apache.org/dyn/closer.lua/hbase-2.6.0/hbase-2.6.0-bin.tar.gz. The page features the Apache logo and navigation links for Community, Projects, Downloads, Learn, Resources & Tools, About, and Search. A red box highlights the download URL: <https://dlcdn.apache.org/hbase/2.6.0/hbase-2.6.0-bin.tar.gz>.



We suggest the following location for your download:

<https://dlcdn.apache.org/hbase/2.6.0/hbase-2.6.0-bin.tar.gz>

Alternate download locations are suggested below.

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

HTTP

<https://dlcdn.apache.org/hbase/2.6.0/hbase-2.6.0-bin.tar.gz>

BACKUP SITES

<https://dlcdn.apache.org/hbase/2.6.0/hbase-2.6.0-bin.tar.gz>

VERIFY THE INTEGRITY OF THE FILES

It is essential that you verify the integrity of the downloaded file using the PGP signature ([.asc](#) file) or a hash ([.md5](#) or [.sha*](#) file). Please read [Verifying Apache Software Foundation Releases](#) for more information on why you should verify our releases.

The PGP signature can be verified using PGP or GPG. First download the [KEYS](#) as well as the [.asc](#) signature file for the relevant distribution. Make sure you get these files from the main distribution site, rather than from a mirror. Then verify the signatures using

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

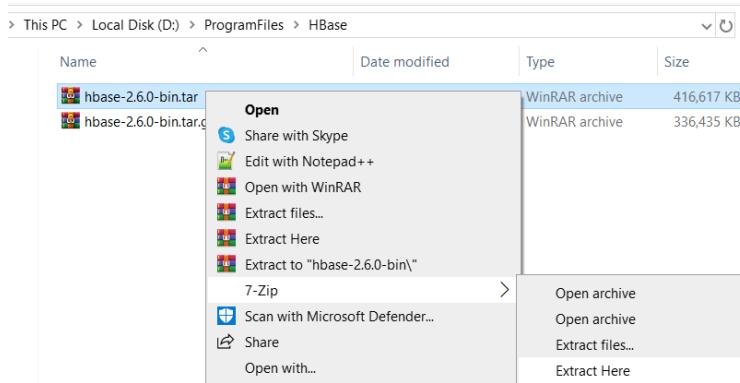
- Choose the installation directory in your machine and copy `hbase-2.6.0-bin.tar.gz` file to that directory. Here, we are choosing HBase installation directory as `D:\ProgramFiles\HBase`.

The screenshot shows a Windows File Explorer window with the path `This PC > Local Disk (D:) > ProgramFiles > HBase`. A single file, `hbase-2.6.0-bin.tar.gz`, is listed. The file is a WinRAR archive (Type) with a size of 336,435 KB. It was modified on 8/30/2024 at 10:17 AM.

- Right click on `hbase-2.6.0-bin.tar.gz` and choose **7-Zip -> Extract Here** option which extracts a new packed file `hbase-2.6.0-bin.tar`.

The screenshot shows a context menu for the `hbase-2.6.0-bin.tar.gz` file. The **Extract Here** option is highlighted. A secondary context menu is open to the right, showing options like **Open archive**, **Open archive**, **Extract files...**, and **Extract Here**.

- Next, unpack `hbase-2.6.0-bin.tar` file using **7zip** utility.



- The tar file extraction may take few minutes to finish. After finishing, you see a folder named `hbase-2.6.0` which consists of HBase binaries and libraries.

Name	Date modified	Type	Size
bin	1/22/2020 8:40 PM	File folder	
conf	1/22/2020 8:40 PM	File folder	
docs	1/22/2020 8:40 PM	File folder	
hbase-webapps	1/22/2020 8:40 PM	File folder	
lib	8/30/2024 10:19 AM	File folder	
CHANGES.md	1/22/2020 8:40 PM	MD File	986 KB
LEGAL	1/22/2020 8:40 PM	File	1 KB
LICENSE.txt	1/22/2020 8:40 PM	Text Document	144 KB
NOTICE.txt	1/22/2020 8:40 PM	Text Document	606 KB
RELEASENOTES.md	1/22/2020 8:40 PM	MD File	577 KB

3.2. Download Hadoop Binaries:

Since HBase is not configured to run on Windows system by default, it is necessary to get Hadoop's native IO utilities for Windows from [cdarlint GitHub repository](#).

Note that HBase 2.6.x version supports Hadoop 3.3.5 or more. Look at [HBase Hadoop Support Matrix](#) to know more about the supported versions of Hadoop for HBase.

- Since we installed the HBase 2.6.0 version, download utilities such as `winutils.exe` and `hadoop.dll` of the latest Hadoop 3.3.6 version from [here](#).
- Create a folder named `hadoop-winutils` in **D drive** and create a sub-folder named `bin` inside it.
- Move `winutils.exe` and `hadoop.dll` files from your downloaded location to `D:\hadoop-winutils\bin` directory.

Name	Date modified	Type	Size
winutils.exe	7/6/2024 2:57 PM	Application	117 KB
hadoop.dll	7/6/2024 2:57 PM	Application extens...	77 KB

3.3. Set up Environment variables:

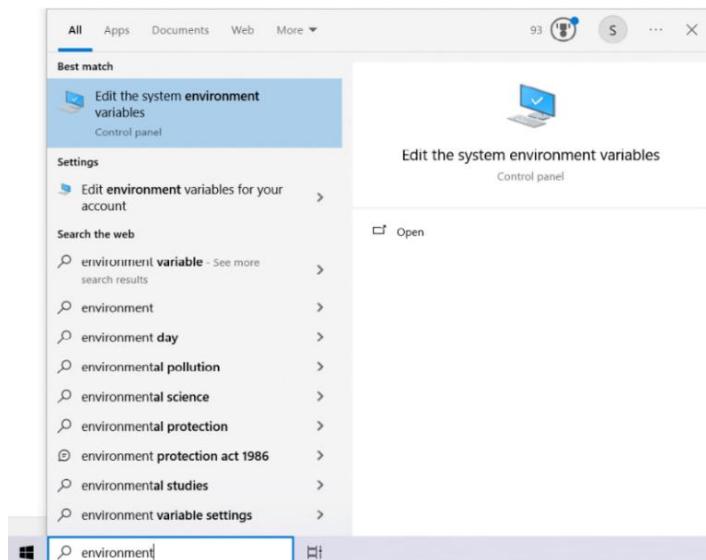
After installing pre-requisites and HBase binaries, we should configure the below environment variables defining Java and HBase default paths.

- **JAVA_HOME**: This is the JDK installation directory path in the machine (*in my machine, it is D:\ProgramFiles\Java\jdk-1.8*). Ignore it if this is already done.
- **HADOOP_HOME**: This is the Hadoop's WinUtils path in the machine (*in our case, it is D:\hadoop-winutils*)
- **HBASE_HOME**: This is the HBase installation directory path in the machine (*in our case, it is D:\ProgramFiles\HBase\hbase-2.6.0*)

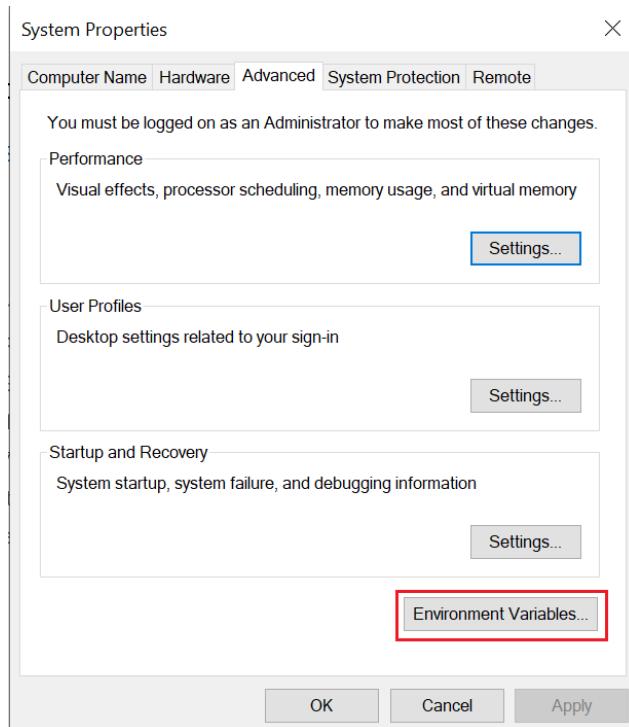
These variables need to be added to either **User environment variables** or **System environment variables** depending on HBase configuration needed **for a single user or for multiple users**.

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

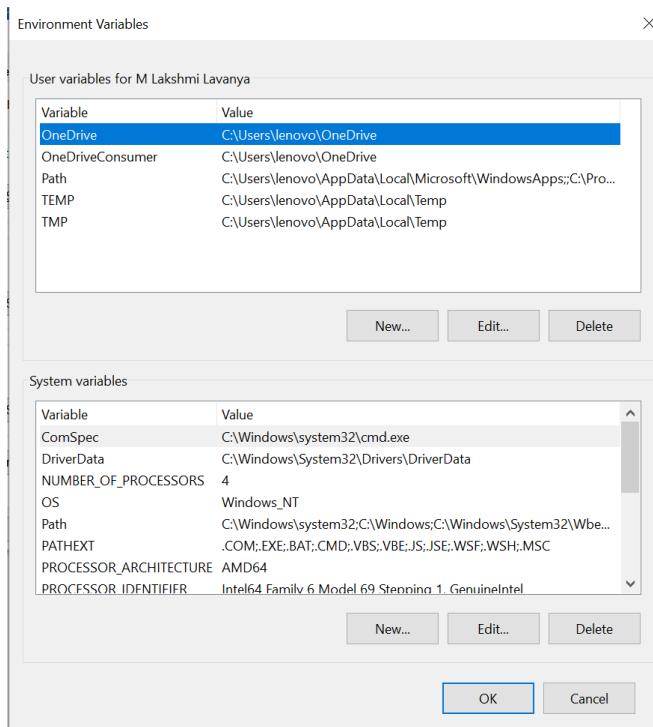
In the Windows search bar, start typing “environment variables” and select the first match which opens up **System Properties** dialog.



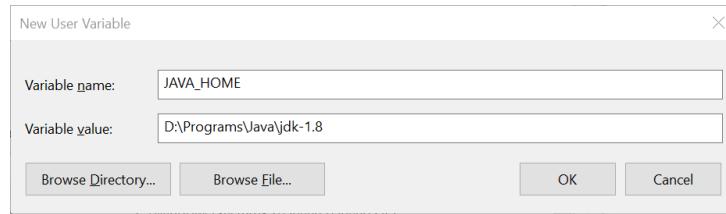
On the **System Properties** window, press **Environment Variables** button.



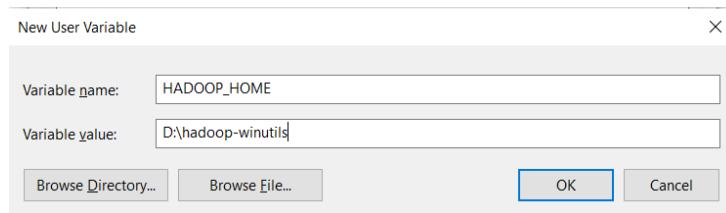
In the **Environment Variables** dialog, click on **New** under **User variables** section.



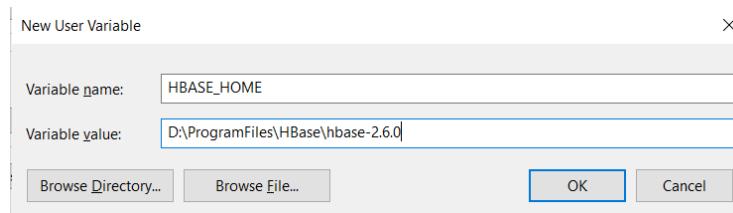
Add JAVA_HOME variable and press OK.



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

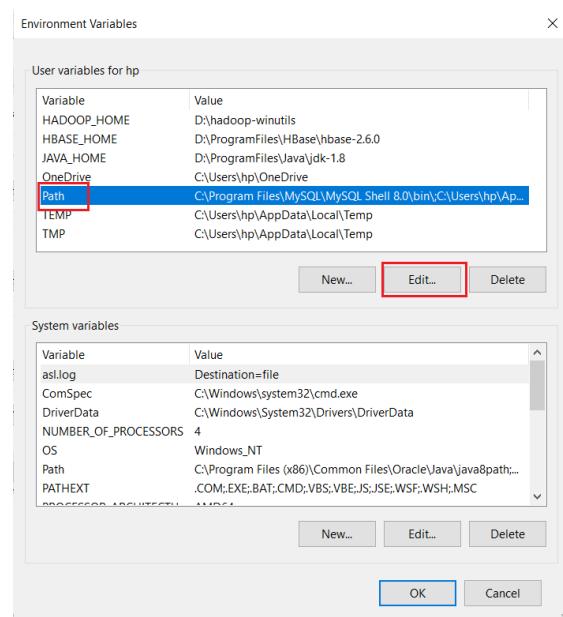


Click on **New** again and add HBASE_HOME variable and press OK.



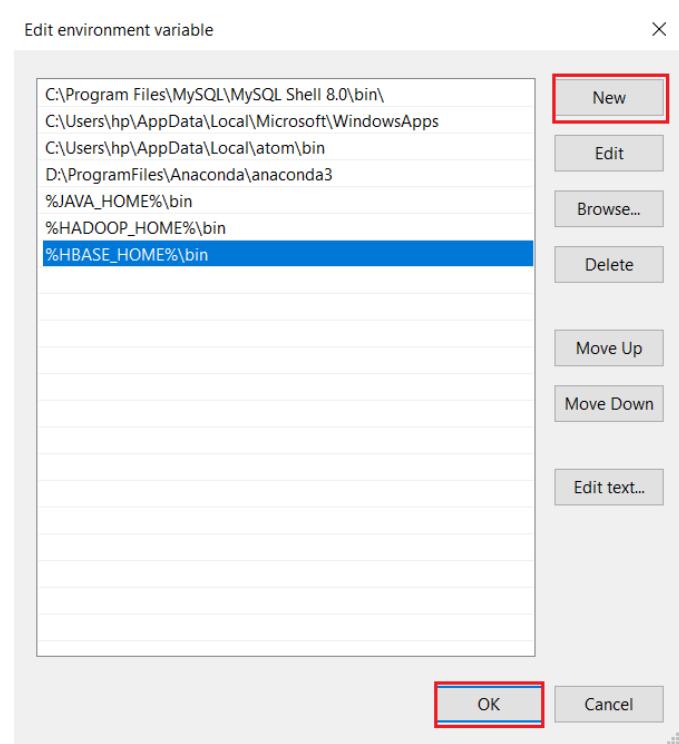
Now, we will update PATH variable to add Java, Hadoop and HBase binary paths.

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

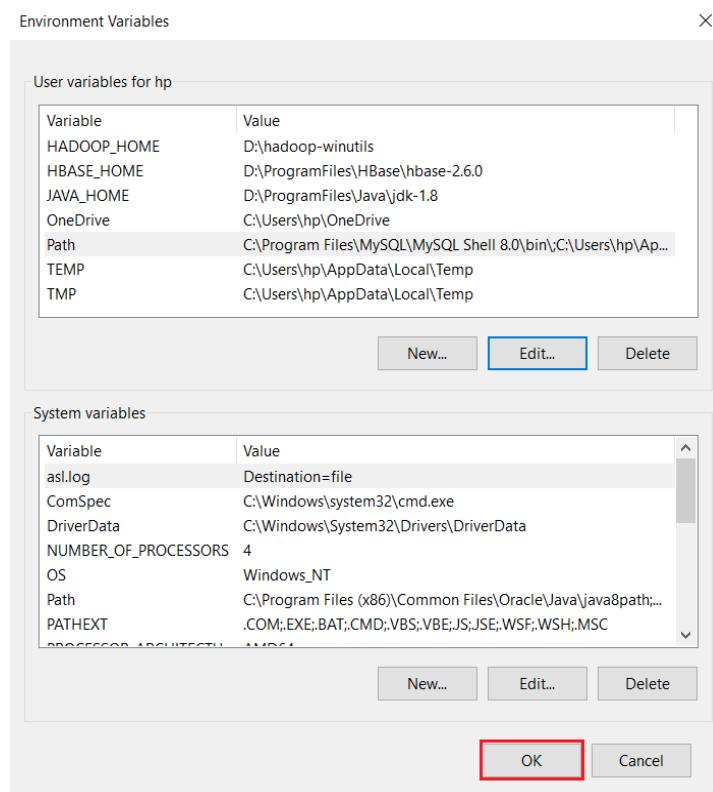


Press **New** and add the following values and press **OK**.

%JAVA_HOME%\bin
%HADOOP_HOME%\bin
%HBASE_HOME%\bin



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



3.4. Configure HBase:

Next, we should modify the following files to configure the standalone HBase:

```
%HBASE_HOME%\conf\hbase-site.xml  
%HBASE_HOME%\conf\hbase-env.cmd
```

3.4.1. HBase Site Configuration:

HBase needs a directory location to store the HBase and Zookeeper files which should be specified in `hbase-site.xml` file. If not specified, it automatically creates directories under `%TEMP%` directory.

Open `hbase-site.xml` file in `HBASE_HOME\conf` folder and replace the existing properties with the below lines inside the `<configuration>` tag.

```
<property>  
    <name>hbase.cluster.distributed</name>  
    <value>false</value>  
</property>  
<property>  
    <name>hbase.unsafe.stream.capability.enforce</name>  
    <value>false</value>  
</property>  
<property>  
    <name>hbase.rootdir</name>  
    <value>file:///D:/ProgramFiles/HBase/hbase-2.6.0/hbase</value>  
</property>  
<property>  
    <name>hbase.zookeeper.property.dataDir</name>  
    <value>/D:/ProgramFiles/HBase/hbase-2.6.0/zookeeper</value>  
</property>  
<property>  
    <name>hbase.zookeeper.quorum</name>  
    <value>localhost</value>  
</property>  
<property>  
    <name>hbase.tmp.dir</name>  
    <value>/D:/ProgramFiles/HBase/hbase-2.6.0/tmp</value>  
</property>
```

```

D:\ProgramFiles\HBase\hbase-2.6.0\conf\hbase-site.xml - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
hbbase-site.xml x
28     local filesystem, in a path under the value configured for `hbase.tmp.dir`.
29     This value is overridden from its default value of `/tmp` because many
30     systems clean `/tmp` on a regular basis. Instead, it points to a path within
31     this HBase installation directory.
32
33     Running against the `LocalFileSystem`, as opposed to a distributed
34     filesystem, runs the risk of data integrity issues and data loss. Normally
35     HBase will refuse to run in such an environment. Setting
36     `hbase.unsafe.stream.capability.enforce` to `false` overrides this behavior,
37     permitting operation. This configuration is for the developer workstation
38     only and should not be used in production!
39
40     See also https://hbase.apache.org/book.html#standalone\_dist
41   -->
42   <property>
43     <name>hbase.cluster.distributed</name>
44     <value>false</value>
45   </property>
46   <property>
47     <name>hbase.unsafe.stream.capability.enforce</name>
48     <value>false</value>
49   </property>
50   <property>
51     <name>hbase.rootdir</name>
52     <value>file:///D:/ProgramFiles/HBase/hbase-2.6.0/hbase</value>
53   </property>
54   <property>
55     <name>hbase.zookeeper.property.dataDir</name>
56     <value>D:/ProgramFiles/HBase/hbase-2.6.0/zookeeper</value>
57   </property>
58   <property>
59     <name>hbase.zookeeper.quorum</name>
60     <value>localhost</value>
61   </property>
62   <property>
63     <name>hbase.tmp.dir</name>
64     <value>/D:/ProgramFiles/HBase/hbase-2.6.0/tmp</value>
65   </property>
66 </configuration>
67

```

Note that HBase creates `hbase`, `zookeeper` and logs `folders` in `%HBASE_HOME%` location automatically when it is started.

3.4.2. HBase Env Configuration:

Open `hbase-env.cmd` file in `%HBASE_HOME%\conf` folder and add the following lines:

```

set JAVA_HOME=%JAVA_HOME%
set HBASE_CLASSPATH=%HBASE_HOME%\lib\client-facing-thirdparty\*
set HBASE_HEAPSIZE=8000
set HBASE_OPTS="-XX:+UseConcMarkSweepGC" "-Djava.net.preferIPv4Stack=true"
set SERVER_GC_OPTS="-verbose:gc" "-XX:+PrintGCDetails" "-XX:+PrintGCDateStamps"
%HBASE_GC_OPTS%
set HBASE_USE_GC_LOGFILE=true
set HBASE_JMX_BASE="-Dcom.sun.management.jmxremote.ssl=false" "-
Dcom.sun.management.jmxremote.authenticate=false"
set HBASE_MASTER_OPTS=%HBASE_JMX_BASE% "-
Dcom.sun.management.jmxremote.port=10101"
set HBASE_REGIONSERVER_OPTS=%HBASE_JMX_BASE% "-
Dcom.sun.management.jmxremote.port=10102"

```

```

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

```

```

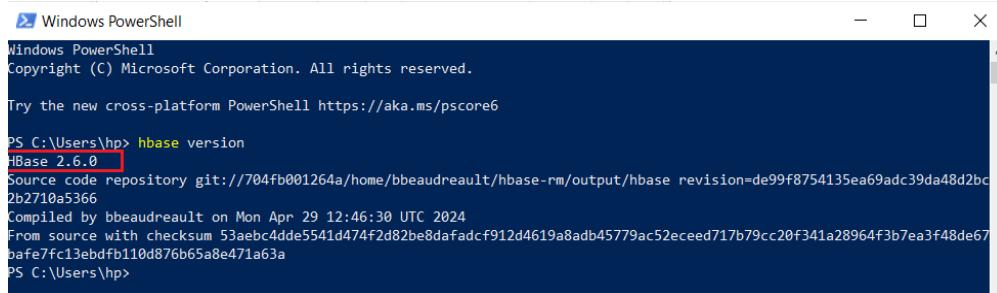
70
71  @rem Where log files are stored. $HBASE_HOME/logs by default.
72  @rem set HBASE_LOG_DIR=%HBASE_HOME%\logs
73
74  @rem A string representing this instance of hbase. $USER by default.
75  @rem set HBASE_IDENT_STRING=%USERNAME%
76
77  @rem Seconds to sleep between slave commands. Unset by default. This
78  @rem can be useful in large clusters, where, e.g., slave rsyncs can
79  @rem otherwise arrive faster than the master can service them.
80  @rem set HBASE_SLAVE_SLEEP=0.1
81
82  @rem Tell HBase whether it should manage its own instance of ZooKeeper or not.
83  @rem set HBASE_MANAGES_ZK=true
84
85  @rem Tell HBase the logger level and appenders
86  @rem set HBASE_ROOT_LOGGER=INFO,DRFA
87
88  @rem Uncomment to enable trace, you can change the options to use other exporters such as jaeger or
89  @rem zipkin. See https://github.com/open-telemetry/opentelemetry-java-instrumentation on how to
90  @rem configure exporters and other components through system properties.
91  @rem set HBASE_TRACE_OPTS="-Dotel.resource.attributes=service.name=HBase -Dotel.traces.exporter=logging otel.metrics.e
92
93  set JAVA_HOME=%JAVA_HOME%
94  set HBASE_CLASSPATH=%HBASE_HOME%\lib\client-facing-thirdparty\*
95  set HBASE_HEAPSIZE=8000
96  set HBASE_OPTS="-XX:+UseConcMarkSweepGC" "-Djava.net.preferIPv4Stack=true"
97  set SERVER_GC_OPTS="-verbose:gc" "-XX:+PrintGCDetails" "-XX:+PrintGCDetails" "%HBASE_GC_OPTS%"
98  set HBASE_USE_GC_LOGFILE=true
99  set HBASE_JMX_BASE="-Dcom.sun.management.jmxremote.ssl=false" "-Dcom.sun.management.jmxremote.authenticate=false"
100  set HBASE_MASTER_OPTS=%HBASE_JMX_BASE% "-Dcom.sun.management.jmxremote.port=10101"
101  set HBASE_REGIONSERVER_OPTS=%HBASE_JMX_BASE% "-Dcom.sun.management.jmxremote.port=10102"
102  set HBASE_THRIFT_OPTS=%HBASE_JMX_BASE% "-Dcom.sun.management.jmxremote.port=10103"
103  set HBASE_ZOOKEEPER_OPTS=%HBASE_JMX_BASE% "-Dcom.sun.management.jmxremote.port=10104"
104  set HBASE_REGIONSERS=%HBASE_HOME%\conf\regionservers
105  set HBASE_LOG_DIR=%HBASE_HOME%\logs
106  set HBASE_IDENT_STRING=%USERNAME%
107  set HBASE_MANAGES_ZK=true
108

```

3.5. Verify HBase Installation:

Open **Windows PowerShell** and run the following command to verify if HBase is installed properly:

```
hbase version
```



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

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

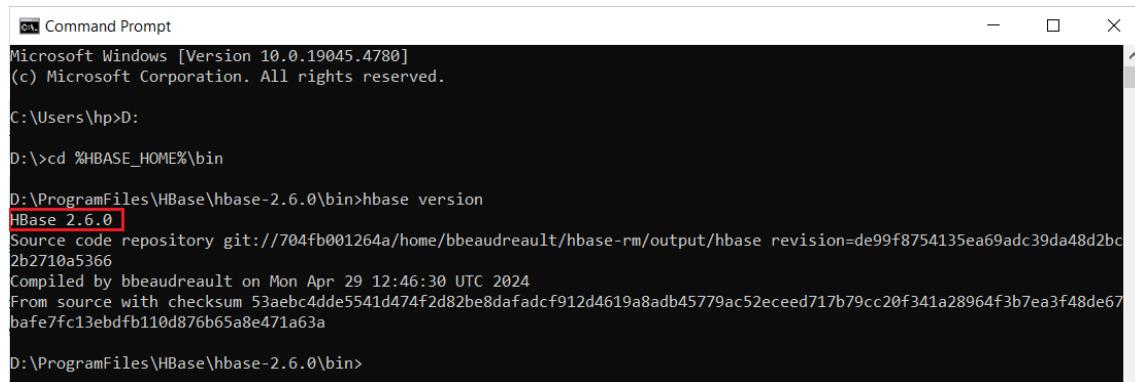
PS C:\Users\hp> hbase version
HBase 2.6.0
Source code repository git://704fb001264a/home/bbeaudreault/hbase-rm/output/hbase revision=de99f8754135ea69adc39da48d2bc
2b2710a5366
Compiled by bbeaudreault on Mon Apr 29 12:46:30 UTC 2024
From source with checksum 53aeabc4dde5541d474f2d82be8dafadcf912d4619a8adb45779ac52eceed717b79cc20f341a28964f3b7ea3f48de67
bafe7fc13ebdfb110d876b65a8e471a63a
PS C:\Users\hp>
```

Here, it shows **HBase 2.6.0** which indicates that HBase has been installed successfully.

Note:

If you would like to check the hbase version using **Command Prompt**, then you must first navigate to the location where HBase is installed as below.

```
D:
cd %HBASE_HOME%\bin
hbase version
```



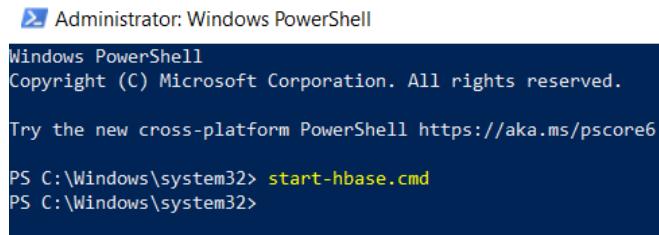
```
Windows Command Prompt
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>D:
D:\>cd %HBASE_HOME%\bin
D:\ProgramFiles\HBase\hbase-2.6.0\bin>hbase version
HBase 2.6.0
Source code repository git://704fb001264a/home/bbeaudreault/hbase-rm/output/hbase revision=de99f8754135ea69adc39da48d2bc
2b2710a5366
Compiled by bbeaudreault on Mon Apr 29 12:46:30 UTC 2024
From source with checksum 53aeabc4dde5541d474f2d82be8dafadcf912d4619a8adb45779ac52eceed717b79cc20f341a28964f3b7ea3f48de67
bafe7fc13ebdfb110d876b65a8e471a63a
D:\ProgramFiles\HBase\hbase-2.6.0\bin>
```

3.6. Start HBase:

Open **Windows PowerShell** as **Administrator** and start HBase using the below command

```
start-hbase.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-hbase.cmd
PS C:\Windows\system32>
```

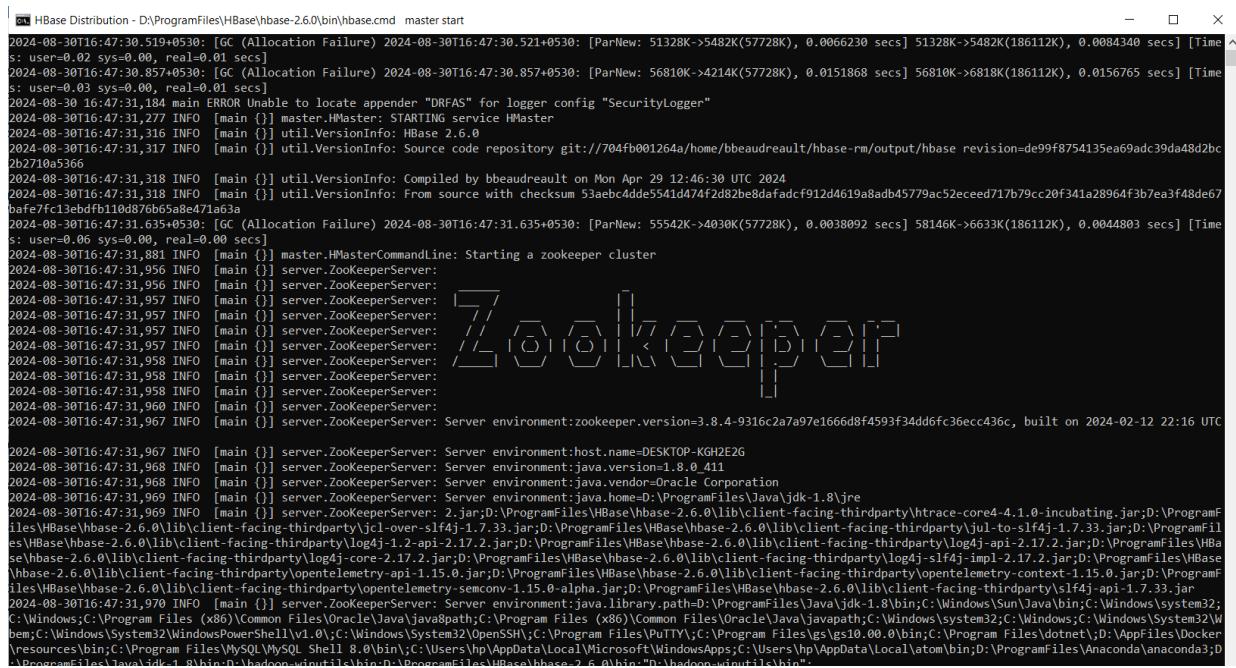
Note:

If you would like to start hbase using **Command Prompt**, then you must first navigate to the location where HBase is installed as below.

D:

```
cd %HBASE_HOME%\bin  
start-hbase.cmd
```

It opens up a new command prompt where we can see three daemon services Zookeeper, HMaster and HRegion server are started. Wait until the HMaster initialization is complete.



```
[  ] HBase Distribution - D:\ProgramFiles\HBase\hbase-2.6.0\bin\hbase.cmd master start
2024-08-30T16:47:30.519+0530: [GC (Allocation Failure) 2024-08-30T16:47:30.521+0530: [ParNew: 51328K->5482K(57728K), 0.0066230 secs] 51328K->5482K(186112K), 0.0084340 secs] [Time
s: user=0.02 sys=0.00, real=0.01 secs]
2024-08-30T16:47:30.857+0530: [GC (Allocation Failure) 2024-08-30T16:47:30.857+0530: [ParNew: 56810K->4214K(57728K), 0.0151868 secs] 56810K->6818K(186112K), 0.0156765 secs] [Time
s: user=0.03 sys=0.00, real=0.01 secs]
2024-08-30T16:47:31.184 INFO [main {}] main ERROR Unable to locate appender "DRFAS" for logger config "SecurityLogger"
2024-08-30T16:47:31.277 INFO [main {}] master.HMaster: STARTING service HMaster
2024-08-30T16:47:31.316 INFO [main {}] util.VersionInfo: HBase 2.6.0
2024-08-30T16:47:31.317 INFO [main {}] util.VersionInfo: Source code repository git://704fb001264a/home/bbeaudreault/hbase-rm/output/hbase revision=de99f8754135ea69adc39da48d2bc
2b2710a5366
2024-08-30T16:47:31.318 INFO [main {}] util.VersionInfo: Compiled by bbeaudreault on Mon Apr 29 12:46:30 UTC 2024
2024-08-30T16:47:31.318 INFO [main {}] util.VersionInfo: From source with checksum 53aeabc4dde5541d474f2d82be8dafadcf912d4619a8adb45779ac5eceed717b79cc20f341a28964f3b7ea3f48de67
bafe7fc13ebdfb110d876b65a8e471a63a
2024-08-30T16:47:31.635+0530: [GC (Allocation Failure) 2024-08-30T16:47:31.635+0530: [ParNew: 55542K->4030K(57728K), 0.0038092 secs] 58146K->6633K(186112K), 0.0044803 secs] [Time
s: user=0.06 sys=0.00, real=0.00 secs]
2024-08-30T16:47:31.881 INFO [main {}] master.HMasterCommandLine: Starting a zookeeper cluster
2024-08-30T16:47:31.950 INFO [main {}] server.ZooKeeperServer:
2024-08-30T16:47:31.950 INFO [main {}] server.ZooKeeperServer:
2024-08-30T16:47:31.957 INFO [main {}] server.ZooKeeperServer:
2024-08-30T16:47:31.957 INFO [main {}] server.ZooKeeperServer:
2024-08-30T16:47:31.957 INFO [main {}] server.ZooKeeperServer:
2024-08-30T16:47:31.958 INFO [main {}] server.ZooKeeperServer:
2024-08-30T16:47:31.958 INFO [main {}] server.ZooKeeperServer:
2024-08-30T16:47:31.958 INFO [main {}] server.ZooKeeperServer:
2024-08-30T16:47:31.960 INFO [main {}] server.ZooKeeperServer:
2024-08-30T16:47:31.967 INFO [main {}] server.ZooKeeperServer: Server environment:zookeeper.version=3.8.4-9316c2a7a97e1666d8f4593f34dd6fc36ecc436c, built on 2024-02-12 22:16 UTC
2024-08-30T16:47:31.967 INFO [main {}] server.ZooKeeperServer: Server environment:host.name=DESKTOP-KGH2E2G
2024-08-30T16:47:31.967 INFO [main {}] server.ZooKeeperServer: Server environment:java.version=1.8.0_411
2024-08-30T16:47:31.967 INFO [main {}] server.ZooKeeperServer: Server environment:java.vendor=Oracle Corporation
2024-08-30T16:47:31.967 INFO [main {}] server.ZooKeeperServer: Server environment:java.home=D:\ProgramFiles\Java\jdk-1.8.jre
2024-08-30T16:47:31.967 INFO [main {}] server.ZooKeeperServer: 2.jar:D:\ProgramFiles\HBase\hbase-2.6.0\lib\client-facing-thirdparty\htrace-core-4.1.0-incubating.jar;D:\ProgramF
iles\HBase\hbase-2.6.0\lib\client-facing-thirdparty\jcl-over-slf4j-1.7.33.jar;D:\Programfiles\HBase\hbase-2.6.0\lib\client-facing-thirdparty\jul-to-slf4j-1.7.33.jar;D:\ProgramFil
es\HBase\hbase-2.6.0\lib\client-facing-thirdparty\log4j-1.2-api-2.17.2.jar;D:\Programfiles\HBase\hbase-2.6.0\lib\client-facing-thirdparty\log4j-api-2.17.2.jar;D:\ProgramFiles\HBase
\hbase-2.6.0\lib\client-facing-thirdparty\log4j-core-2.17.2.jar;D:\Programfiles\HBase\hbase-2.6.0\lib\client-facing-thirdparty\log4j-impl-2.17.2.jar;D:\ProgramFiles\HBase
\hbase-2.6.0\lib\client-facing-thirdparty\opentelemetry-api-1.15.0.jar;D:\Programfiles\HBase\hbase-2.6.0\lib\client-facing-thirdparty\opentelemetry-context-1.15.0.jar;D:\ProgramF
iles\HBase\hbase-2.6.0\lib\client-facing-thirdparty\opentelemetry-semconv-1.15.0-alpha.jar;D:\Programfiles\HBase\hbase-2.6.0\lib\client-facing-thirdparty\slf4j-api-1.7.33.jar
2024-08-30T16:47:31.970 INFO [main {}] server.ZooKeeperServer: Server environment:java.library.path=D:\ProgramFiles\Java\jdk-1.8\bin;C:\Windows\Sun\Java\bin;C:\Windows\system32;
C:\Windows;C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Windows\system32;C:\Windows;C:\Windows\System32\W
bem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Windows\System32\OpenSSH;C:\Program Files\PutTY\;C:\Program Files\gs\gs10.00\bin;C:\Program Files\dotnet\;D:\Appfiles\docker
\resources\bin;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\hp\AppData\Local\Microsoft\WindowsApps;C:\Users\hp\AppData\Local\atom\bin;D:\ProgramFiles\Anaconda3;D
:\ProgramFiles\Java\jdk-1.8\bin;D:\hadoop\winutils\bin;D:\ProgramFiles\HBase\hbase-2.6.0\bin;"D:\hadoop\winutils\bin".
```

```

[HBase Distribution - D:\ProgramFiles\HBase\hbase-2.6.0\bin\hbase.cmd master start]
2024-08-30T16:47:32,634 INFO [ProcessThread($id:0 cport:2181: {})] server.PreRequestProcessor: PreRequestProcessor (sid:0) started, reconfigEnabled=false
2024-08-30T16:47:32,635 INFO [main ()] server.RequestThrottler: zookeeper.request_throttler.shutdownTimeout = 10000 ms
2024-08-30T16:47:32,661 INFO [main ()] zookeeper.MinizooKeeperCluster: Started connectionTimeout=30000, dir=D:\ProgramFiles\HBase\hbase-2.6.0\zookeeper\zookeeper_0, clientPort=2181, secureClientPort=-1, dataDir=D:\ProgramFiles\HBase\hbase-2.6.0\zookeeper\zookeeper_0\version-2, dataDirSize=104012
dataLogDir=D:\ProgramFiles\HBase\hbase-2.6.0\zookeeper\zookeeper_0\version-2, datalogSize=104012
tickTime=2000, maxClientCnxns=300, minSessionTimeout=40000, clientPortListenBacklog=1, serverId=0
2024-08-30T16:47:32,703 INFO [main ()] client.FourLetterWordMain: connecting to localhost 2181
2024-08-30T16:47:32,750 INFO [NIOWorkerThread-1 {}] command.FourLetterCommands: The list of known four letter word commands is : [(1936881266=srvr, 1937006964=stat, 20030034
91=wchc, 1685417328=dump, 1668425044=crst, 1936880590=rst, 1701738890=envi, 1668247142=conf, -720899=telnet close, 1751217000=hash, 2003003507=wchs, 2003003504=wchp, 1684632
179=dirs, 1668247155=cons, 1835955314=mrtr, 1769173615=isro, 1735683435=gtmk, 1937010027=stmk)]
2024-08-30T16:47:32,750 INFO [NIOWorkerThread-1 {}] command.FourLetterCommands: The list of enabled four letter word commands is : [[wchs, stat, wchp, dirs, stmk, conf, ruok
, mtr, srvc, wchc, envi, srst, isro, dump, gtmk, telnet close, crst, hash, cons]]
2024-08-30T16:47:32,752 INFO [NIOWorkerThread-1 {}] server.NIOServerCnxn: Processing stat command from /127.0.0.1:55841
2024-08-30T16:47:32,760 INFO [NIOWorkerThread-1 {}] command.AbstractFourLetterCommand: Stat command output
2024-08-30T16:47:32,761 INFO [main ()] zookeeper.MinizooKeeperCluster: Started MinizooKeeperCluster and ran 'stat' on client port=2181
2024-08-30T16:47:32,762 INFO [main ()] master.MasterCommandline: Starting up instance of LocalHBaseCluster; master=1, regionserverCount=1
2024-08-30T16:47:33,159+0530: [GC (Allocation Failure) 2024-08-30T16:47:33.168+0530: [ParNew: 5538K->5670K(57728K), 0.0078116 secs] 57961K->8760K(186112K), 0.0086486 secs] [
Times: user=0.00 sys=0.00 real=0.01 secs]
2024-08-30T16:47:33,169+0530: [GC (CMS Initial Mark) [CMS-initial-mark: 3089K(128384K)] 8760K(186112K), 0.0031888 secs] [Times: user=0.00 sys=0.00, real=0.00 secs]
2024-08-30T16:47:33,173+0530: [CMS-concurrent-mark-start]
2024-08-30T16:47:33,267+0530: [CMS-concurrent-mark: 0.092/0.092 secs] [Times: user=0.03 sys=0.09, real=0.09 secs]
2024-08-30T16:47:33,267+0530: [CMS-concurrent-preclean-start]
2024-08-30T16:47:33,297+0530: [CMS-concurrent-preclean: 0.029/0.029 secs] [Times: user=0.02 sys=0.00, real=0.03 secs]
2024-08-30T16:47:33,298+0530: [CMS-concurrent-clean-start]
2024-08-30T16:47:34,589+0530: [GC (Allocation Failure) 2024-08-30T16:47:34.589+0530: [ParNew: 56998K->6020K(57728K), 0.0086866 secs] 60088K->11943K(186112K), 0.0097990 secs] [
Times: user=0.06 sys=0.00, real=0.01 secs]
2024-08-30T16:47:34,902+0530: [GC (CMS Final Remark) [YG occupancy: 34883 K (57728 K)] 2024-08-30T16:47:34.904+0530: [Rescan (parallel), 0.0032091 secs] 2024-08-30T16:47:34.90
7+0530: [weak refs processing, 0.0165421 secs] 2024-08-30T16:47:34.924+0530: [class unloading, 0.0043375 secs] 2024-08-30T16:47:34.928+0530: [scrub symbol table, 0.0050797 secs]
2024-08-30T16:47:34.933+0530: [scrub string table, 0.0010343 secs] [1 CMS-remark: 5923K(128384K)] 40807K(186112K), 0.0454174 secs] [Times: user=0.00 sys=0.02, real=0.03 secs]
2024-08-30T16:47:34,951+0530: [CMS-concurrent-sweep-start]
2024-08-30T16:47:34,966+0530: [CMS-concurrent-sweep: 0.014/0.014 secs] [Times: user=0.00 sys=0.00, real=0.02 secs]
2024-08-30T16:47:34,967+0530: [CMS-concurrent-reset-start]
2024-08-30T16:47:34,979+0530: [CMS-concurrent-reset: 0.011/0.011 secs] [Times: user=0.02 sys=0.00, real=0.01 secs]
2024-08-30T16:47:35,459 INFO [main ()] regionserver.RSRpcServices: master@DESKTOP-KGH2E2G:160000 server-side Connection retries=45
2024-08-30T16:47:35,476+0530: [GC (Allocation Failure) 2024-08-30T16:47:35.477+0530: [ParNew: 57348K->5191K(57728K), 0.0066103 secs] 63255K->12299K(186112K), 0.0074365 secs] [
Times: user=0.06 sys=0.00, real=0.01 secs]
2024-08-30T16:47:35,647 INFO [main ()] ipc.RpcExecutor: Instantiated default.FPBQ.Fifo with queueClass=class java.util.concurrent.LinkedBlockingQueue; numCallQueues=3, maxQu
ueLength=300, handlerCount=30
2024-08-30T16:47:35,649 INFO [main ()] ipc.RpcExecutor: Instantiated priority.RWQ.Fifo with queueClass=class java.util.concurrent.LinkedBlockingQueue; numCallQueues=2, maxQu
ueLength=200, handlerCount=20
2024-08-30T16:47:35,660 INFO [main ()] ipc.RWQExecutor: priority.RWQ.Fifo writeQueues=1 writeHandlers=2 readQueues=18 scanHandlers=0
2024-08-30T16:47:35,660 INFO [main ()] ipc.RpcExecutor: Instantiated replication.FPBQ.Fifo with queueClass=class java.util.concurrent.LinkedBlockingQueue; numCallQueues=1, m
axQueueLength=30, handlerCount=3

```

```

[HBase Distribution - D:\ProgramFiles\HBase\hbase-2.6.0\bin\hbase.cmd master start]
scripts (Longer MTR)
2024-08-30T16:47:52,030 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] wal.WALFactory: Instantiating WALProvider of type class org.apache.hadoop.hbase.wal.AsyncFSWALProvider
2024-08-30T16:47:52,037 INFO [RegionServerTracker-0 {}] master.RegionServerTracker: RegionServer ephemeral node created, adding [desktop-kgh2e2g,16020,1725016664539]
2024-08-30T16:47:52,125 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] regionserver.MetricsRegionServerRapperImpl: Computing regionserver metrics every 5000 milliseconds
2024-08-30T16:47:52,151 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] regionserver.MemStoreFlusher: globalMemStoreLimit=3.1 G, globalMemStoreLimitLowMark=3.0 G, Offheap=false
2024-08-30T16:47:52,162 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] throttle.PressureAwareCompactionThroughputController: Compaction throughput configurations, higher bound: 100.00
MB/second, lower bound 50.00 MB/second, off peak: unlimited, tuning period: 60000 ms
2024-08-30T16:47:52,163 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] hbase.ChoreService: Chore ScheduledChore name=CompactionThroughputTuner, period=60000, unit=MILLISECONDS is enable
d.
2024-08-30T16:47:52,165 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] regionserver.HRegionServer: CompactionChecker runs every PT10S
2024-08-30T16:47:52,180 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] hbase.ChoreService: Chore ScheduledChore name=CompactedHFileCleaner, period=120000, unit=MILLISECONDS is enable
d.
2024-08-30T16:47:52,183 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] hbase.ChoreService: Chore ScheduledChore name=CompactionChecker, period=10000, unit=MILLISECONDS is enabled.
2024-08-30T16:47:52,185 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] hbase.ChoreService: Chore ScheduledChore name=MemstoreFlusherChore, period=10000, unit=MILLISECONDS is enabled.
2024-08-30T16:47:52,185 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] hbase.ChoreService: Chore ScheduledChore name=nonceCleaner, period=360000, unit=MILLISECONDS is enabled.
2024-08-30T16:47:52,186 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] hbase.ChoreService: Chore ScheduledChore name=BrokenStorefileCleaner, period=21600000, unit=MILLISECONDS is enable
d.
2024-08-30T16:47:52,186 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] hbase.ChoreService: Chore ScheduledChore name=deskstop-kgh2e2g,16020,1725016664539-MobFileCleanerChore, period=86
400, unit=SECONDS is enabled.
2024-08-30T16:47:52,228 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] regionserver.HeapMemoryManager: Starting, tuneOn=false
2024-08-30T16:47:52,233 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] hbase.ChoreService: Chore ScheduledChore name=kgh2e2g,16020,1725016664539-HeapMemoryTunerChore, period=6
0000, unit=MILLISECONDS is enabled.
2024-08-30T16:47:52,258 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] regionserver.Replication: desktop-kgh2e2g,16020,1725016664539 started
2024-08-30T16:47:52,259 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] regionserver.HRegionServer: Serving as desktop-kgh2e2g,16020,1725016664539, RpcServer on DESKTOP-KGH2E2G/192.168
.56.1:16020, sessionid=0x10028F0F050001
2024-08-30T16:47:52,267 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] quotas.RegionServerRpcQuotaManager: Quota support disabled
2024-08-30T16:47:52,267 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] quotas.RegionServerSpaceQuotaManager: Quota support disabled, not starting space quota manager.
2024-08-30T16:47:53,003 WARN [DESKTOP-KGH2E2G:160000 ()] assignment.AssignmentManager: No servers available; cannot place 1 unassigned regions.
2024-08-30T16:47:54,032 WARN [DESKTOP-KGH2E2G:160000 ()] assignment.AssignmentManager: No servers available; cannot place 1 unassigned regions.
2024-08-30T16:47:55,034 WARN [DESKTOP-KGH2E2G:160000 ()] assignment.AssignmentManager: No servers available; cannot place 1 unassigned regions.
2024-08-30T16:47:55,416 INFO [RS:0;DESKTOP-KGH2E2G:16020 {}] wal.AbstractFSWAL: WAL configuration: blocksize=64 MB, rollsize=32 MB, prefix=desktop-kgh2e2g%2C16020%2C1725016664539, s
uffix=, logDir=file:/D:/ProgramFiles/HBase/hbase-2.6.0/hbase/WALS/desktop-kgh2e2g,16020,1725016664539, archiveDir=file:/D:/ProgramFiles/HBase/hbase/oldWALs, maxLogs=199
2024-08-30T16:47:55,653 INFO [PEWorker-4 {}] procedure2.ProcedureExecutor: Initialized subprocedures=[[pid=3, ppid=2, state=RUNNABLE; OpenRegionProcedure 1588230740, server=
desktop-kgh2e2g,16020,1725016664539]]
2024-08-30T16:47:55,691+0530: [GC (Allocation Failure) 2024-08-30T16:47:55.692+0530: [ParNew: 57728K->6400K(57728K), 0.0182590 secs] 77926K->32370K(186112K), 0.0204321 secs] [
Times: user=0.05 sys=0.02, real=0.02 secs]
2024-08-30T16:47:55,982 INFO [RS_OPEN_META-regionserver/DESKTOP-KGH2E2G:16020-0 {event_type=M_RS_OPEN_META, pid=3}] handler.AssignRegionHandler: Open hbase:meta,,1.158823074
0
2024-08-30T16:47:55,984 INFO [RS_OPEN_META-regionserver/DESKTOP-KGH2E2G:16020-0 {event_type=M_RS_OPEN_META, pid=3}] wal.WALFactory: Instantiating WALProvider of type class o

```

```

[HBase Distribution - D:\ProgramFiles\HBase\hbase-2.6.0\bin\hbase.cmd master start]
- □ ×

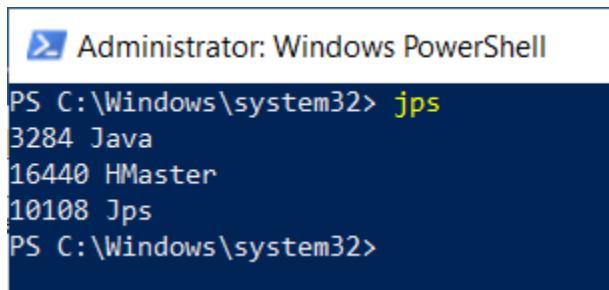
2024-08-30T16:48:01,992 INFO [PEWorker-9 ()] assignment.RegionStateStore: pid=5 updating hbase:meta row=53fbdbab28269f64baeb71b2553afe7d1, regionState=OPENING, regi ^ 
onLocation=desktop-kgh2e2g,16020,1725016664539
2024-08-30T16:48:01,993 INFO [PEWorker-9 ()] procedure2.ProcedureExecutor: Initialized subprocedures=[{pid=6, ppid=5, state=RUNNABLE; OpenRegionProcedure 53fbdbab28269f64baeb71b2553afe7d1, server=desktop-kgh2e2g,16020,1725016664539}]
2024-08-30T16:48:02,166 INFO [RS_OPEN_PRIORITY_REGION-regionserver/DESKTOP-KGH2E2G:16020-0 {event_type=M_RS_OPEN_PRIORITY_REGION, pid=6}] handler.AssignRegionHandler: Open hbase:namespace,,1725016681591.53fbdbab28269f64baeb71b2553afe7d1-1 []
2024-08-30T16:48:02,176 INFO [StoreOpener-53fbdbab28269f64baeb71b2553afe7d1-1 {}] regionserver.HStore: Created cacheConfig: cacheDataOnRead=true, cacheDataOnWrite=false, cacheBloomOnWrite=false, cacheEvictOnClose=false, cacheDataCompressed=false, prefetchOnOpen=false, for column family info of region 53fbdbab28269f64baeb71b2553afe7d1
2024-08-30T16:48:02,183 INFO [StoreOpener-53fbdbab28269f64baeb71b2553afe7d1-1 {}] compactions.CompactionConfiguration: size [minCompactSize:128 MB, maxCompactSize:8.00 EB, offPeakMaxCompactSize:8.00 EB]; files [minFilesToCompact:3, maxFilesToCompact:10]; ratio: 1.200000; off-peak ratio 5.000000; throttle point 2684354560; major period 604800000, major jitter 0.500000, min window to compact 0.000000; tiered compaction: max_age 9223372036854775807, incoming window min 6, compaction policy for tiered window org.apache.hadoop.hbase.regionserver.compactions.ExploringCompactionPolicy, single output for minor true, compaction window factory org.apache.hadoop.hbase.regionserver.compactions.ExponentialCompactionWindowFactory, region 53fbdbab28269f64baeb71b2553afe7d1 columnFamilyName info
2024-08-30T16:48:02,183 INFO [StoreOpener-53fbdbab28269f64baeb71b2553afe7d1-1 {}] regionserver.HStore: Store=53fbdbab28269f64baeb71b2553afe7d1/info, memstore type=DefaultMemStore, storagePolicy=None, verifyBulkLoads=false, parallelPutCountInThreshold=50, encoding=None, compression=None
2024-08-30T16:48:02,197 INFO [RS_OPEN_PRIORITY_REGION-regionserver/DESKTOP-KGH2E2G:16020-0 {event_type=M_RS_OPEN_PRIORITY_REGION, pid=6}] regionserver.HRegion: Opened 53fbdbab28269f64baeb71b2553afe7d1 next sequenceId=2; SteppingSplitPolicy.super{IncreasingToUpperBoundRegionSplitPolicy{initialSize=268435456, ConstantSizeRegionSplitPolicy{desiredMaxFileSize=10686330720, jitterRate=-0.00475789065711975}}}, FlushLargeStoresPolicy{flushSizeLowerBound=1}
2024-08-30T16:48:02,204 INFO [RS_OPEN_PRIORITY_REGION-regionserver/DESKTOP-KGH2E2G:16020-0 {event_type=M_RS_OPEN_PRIORITY_REGION, pid=6}] regionserver.HRegionServer: Post open deploy tasks for hbase:namespace,,1725016681591.53fbdbab28269f64baeb71b2553afe7d1, pid=6, mastersystemTime=1725016682155
2024-08-30T16:48:02,214 INFO [RS_OPEN_PRIORITY_REGION-regionserver/DESKTOP-KGH2E2G:16020-0 {event_type=M_RS_OPEN_PRIORITY_REGION, pid=6}] handler.AssignRegionHandler: Opened hbase:namespace,,1725016681591.53fbdbab28269f64baeb71b2553afe7d1.
2024-08-30T16:48:02,215 INFO [PEWorker-7 ()] assignment.RegionStateStore: pid=5 updating hbase:meta row=53fbdbab28269f64baeb71b2553afe7d1, regionState=OPEN, openSeqNum=2, regionLocation=desktop-kgh2e2g,16020,1725016664539
2024-08-30T16:48:02,226 INFO [PEWorker-7 ()] procedure2.ProcedureExecutor: Finished subprocedure pid=6, resume processing ppid=5
2024-08-30T16:48:02,231 INFO [PEWorker-7 ()] procedure2.ProcedureExecutor: Finished pid=6, ppid=5, state=SUCCESS; OpenRegionProcedure 53fbdbab28269f64baeb71b2553afe7d1, server=desktop-kgh2e2g,16020,1725016664539 in 227 msec
2024-08-30T16:48:02,234 INFO [PEWorker-6 ()] procedure2.ProcedureExecutor: Finished subprocedure pid=5, resume processing ppid=4
2024-08-30T16:48:02,234 INFO [PEWorker-6 ()] procedure2.ProcedureExecutor: Finished pid=5, ppid=4, state=SUCCESS; TransitRegionStateProcedure table=hbase:namespace, region=hbase:namespace execute state=CREATE_TABLE_UPDATE_DESC_CACHE
2024-08-30T16:48:02,236 INFO [PEWorker-8 ()] procedure.CreateTableProcedure: pid=4, state=RUNNABLE:CREATE_TABLE_UPDATE_DESC_CACHE, locked=true; CreateTableProcedure table=hbase:namespace execute state=CREATE_TABLE_UPDATE_DESC_CACHE
2024-08-30T16:48:02,244 INFO [PEWorker-8 ()] hbase.MetaTableAccessor: Updated tableName=hbase:namespace, state=ENABLED in hbase:meta
2024-08-30T16:48:02,300 INFO [PEWorker-8 ()] procedure.CreateTableProcedure: pid=4, state=RUNNABLE:CREATE_TABLE_POST_OPERATION, locked=true; CreateTableProcedure table=hbase:namespace execute state=CREATE_TABLE_POST_OPERATION
2024-08-30T16:48:02,304 INFO [PEWorker-8 ()] procedure2.ProcedureExecutor: Finished pid=4, state=SUCCESS; CreateTableProcedure table=hbase:namespace in 705 msec
2024-08-30T16:48:02,557 INFO [PEWorker-5 ()] procedure2.ProcedureExecutor: Finished pid=7, state=SUCCESS; CreateNamespaceProcedure, namespace=default in 121 msec
2024-08-30T16:48:02,635 INFO [PEWorker-2 ()] procedure2.ProcedureExecutor: Finished pid=8, state=SUCCESS; CreateNamespaceProcedure, namespace=hbase in 66 msec
2024-08-30T16:48:02,740 INFO [master/DESKTOP-KGH2E2G:16000:becomeActiveMaster {}] master.HMaster: Master has completed initialization 17.703sec
2024-08-30T16:48:02,747 INFO [master/DESKTOP-KGH2E2G:16000:becomeActiveMaster {}] quotas.MasterQuotaManager: Quota support disabled
2024-08-30T16:48:02,753 INFO [master/DESKTOP-KGH2E2G:16000:becomeActiveMaster {}] slowlog.SlowLogMasterService: Slow/Large requests logging to system table hbase:slowlog is disabled. Quitting.
2024-08-30T16:48:02,760 INFO [master/DESKTOP-KGH2E2G:16000:becomeActiveMaster {}] waleventtracker.WALEventTrackerTableCreator: wal event tracker requests logging to table REPLICATION.WALEVENTTRACKER is disabled. Quitting.
2024-08-30T16:48:02,763 INFO [master/DESKTOP-KGH2E2G:16000:becomeActiveMaster {}] master.ReplicationSinkTrackerTableCreator: replication sink tracker requests logging to table REPLICATION.SINK_TRACKER is disabled. Quitting.
2024-08-30T16:48:02,764 INFO [master/DESKTOP-KGH2E2G:16000:becomeActiveMaster {}] zookeeper.ZKWatcher: not a secure deployment, proceeding
2024-08-30T16:48:02,770 INFO [master/DESKTOP-KGH2E2G:16000:becomeActiveMaster {}] hbase.ChoreService: Chore ScheduledChore name=desktop-kgh2e2g,16000,1725016652956

```

To verify the HBase has been started successfully, use the `jps` command as below

```
jps
```

It should display **HMaster** process which denotes **HMaster**, single **HRegionServer** and **Zookeeper** services are running in a single JVM.



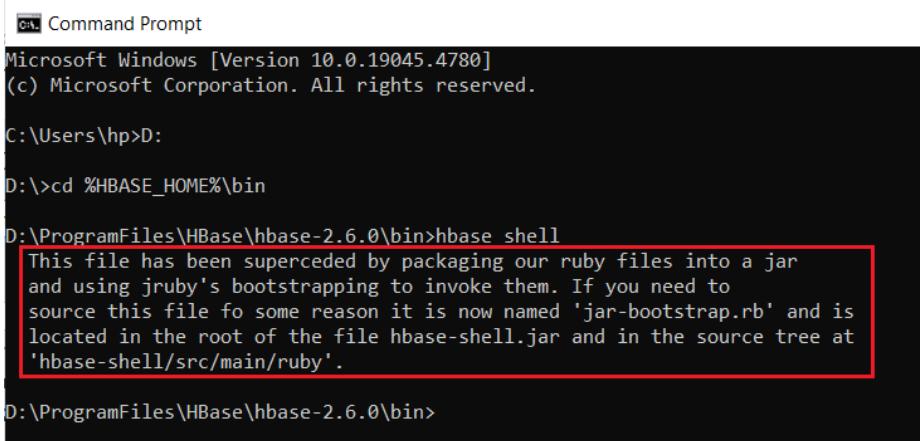
3.7. HBase Shell:

Open **Windows PowerShell** and run the below command to launch HBase shell.

```
hbase shell
```

If you would like to launch hbase from **Command Prompt**, then you must first navigate to the location where HBase is installed as below.

```
D:  
cd %HBASE_HOME%\bin  
hbase shell
```



```
Command Prompt  
Microsoft Windows [Version 10.0.19045.4780]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\hp>D:  
  
D:>cd %HBASE_HOME%\bin  
  
D:\ProgramFiles\HBase\hbase-2.6.0\bin>hbase shell  
This file has been superceded by packaging our ruby files into a jar  
and using jruby's bootstrapping to invoke them. If you need to  
source this file for some reason it is now named 'jar-bootstrap.rb' and is  
located in the root of the file hbase-shell.jar and in the source tree at  
'hbase-shell/src/main/ruby'.  
  
D:\ProgramFiles\HBase\hbase-2.6.0\bin>
```

While launching the hbase shell, you may encounter error *"This file has been superceded by packaging our ruby files into a jar and using jruby's bootstrapping to invoke them."*

To resolve this error, follow the below steps:

- Edit hbase.cmd file in %HBASE_HOME%\bin location and replace the following lines of code under :shell label.

Before:

```
:shell  
rem find the hbase ruby sources  
if exist "%HBASE_HOME%\lib\ruby" (  
    set HBASE_OPTS=%HBASE_OPTS% -Dhbase.ruby.sources="%HBASE_HOME%\lib\ruby"  
) else (  
    set HBASE_OPTS=%HBASE_OPTS% -Dhbase.ruby.sources="%HBASE_HOME%\hbase-  
shell\src\main\ruby"  
)  
set HBASE_OPTS=%HBASE_OPTS% %HBASE_SHELL_OPTS%  
  
set CLASS=org.jruby.Main -X+O %JRUBY_OPTS% "%HBASE_HOME%\bin\hirb.rb"  
goto :eof
```

After:

```
:shell  
rem find the hbase ruby sources  
if exist "%HBASE_HOME%\lib\ruby" (  
    @rem set HBASE_OPTS=%HBASE_OPTS% -  
    Dbase.ruby.sources="%HBASE_HOME%\lib\ruby"  
    set JRUBY_OPTS="${JRUBY_OPTS} -X+O"
```

```

set CLASS="org.jruby.JarBootstrapMain"
) else (
    set HBASE_OPTS=%HBASE_OPTS% -Dhbase.ruby.sources="%HBASE_HOME%\hbase-
shell\src\main\ruby"
    set CLASS="org.jruby.Main -X+O %JRUBY_OPTS% %HBASE_HOME%/hbase-
shell/src/main/ruby/jar-bootstrap.rb"
)
set HBASE_OPTS=%HBASE_OPTS% %HBASE_SHELL_OPTS%

@rem set CLASS=org.jruby.Main -X+O %JRUBY_OPTS% "%HBASE_HOME%\bin\hirb.rb"
goto :eof

```

```

360 ) else (
361     call %JAVA% %java_arguments%
362 )
363
endlocal
365 goto :eof
366
367 :shell
368     rem find the hbase ruby sources
369 if exist "%HBASE_HOME%\lib\ruby" (
370     @rem set HBASE_OPTS=%HBASE_OPTS% -Dhbase.ruby.sources="%HBASE_HOME%\lib\ruby"
371     set JRUBY_OPTS=%JRUBY_OPTS% -X+O%
372     set CLASS="org.jruby.JarBootstrapMain"
373 )
374     set HBASE_OPTS=%HBASE_OPTS% -Dhbase.ruby.sources="%HBASE_HOME%\hbase-shell\src\main\ruby"
375     set CLASS="org.jruby.Main -X+O %JRUBY_OPTS% %HBASE_HOME%\hbase-shell\src\main\ruby\jar-bootstrap.rb"
376 )
377     set HBASE_OPTS=%HBASE_OPTS% %HBASE_SHELL_OPTS%
378
379     @rem set CLASS=org.jruby.Main -X+O %JRUBY_OPTS% "%HBASE_HOME%\bin\hirb.rb"
380     goto :eof
381
382 :master
383     set CLASS=org.apache.hadoop.hbase.master.HMaster
384

```

- Then download jansi-1.18.jar file from [this Maven repository](#) and place the file under %HBASE_HOME%\lib location.

Name	Date modified	Type	Size
jakarta.validation-api-2.0.2.jar	1/22/2020 8:40 PM	Executable Jar File	90 KB
jamon-runtime-2.4.1.jar	1/22/2020 8:40 PM	Executable Jar File	24 KB
jansi-1.18.jar	8/29/2024 7:18 PM	Executable Jar File	281 KB
javassist-3.30.2-GAjar	1/22/2020 8:40 PM	Executable Jar File	777 KB
java-util-1.9.0jar	1/22/2020 8:40 PM	Executable Jar File	58 KB
javax.activation-api-1.2.0.jar	1/22/2020 8:40 PM	Executable Jar File	56 KB

- Now, you should be able to launch the hbase shell

```
hbase shell
```

```
cmd Command Prompt
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>D:

D:\>cd %HBASE_HOME%\bin

D:\ProgramFiles\HBase\hbase-2.6.0\bin>hbase shell
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.6.0, rde99f8754135ea69adc39da48d2bc2b2710a5366, Mon Apr 29 12:46:30 UTC 2024
Took 0.0030 seconds

'stty' is not recognized as an internal or external command,
operable program or batch file.
hbase:001:> quit

D:\ProgramFiles\HBase\hbase-2.6.0\bin>
```

Use `quit` or `exit` to come out of HBase interactive shell.

Note:

HBase 2.x version does not support installing Pseudo Distributed mode of HBase in Windows Operating System.

4. HBase Shell Commands:

HBase provides an interactive tool called “`hbase shell`” where you can execute commands to interact with HBase server.

To interact with the database, HBase Shell commands are divided into 14 groups:

1. General Commands
2. Namespace Commands
3. Data Definition Commands
4. Data Manipulation Commands
5. Tool Commands
6. Cluster Replication Commands
7. Snapshot Commands
8. Configuration Commands
9. Quota Commands
10. Security Commands
11. Procedure Commands
12. Visibility Label Commands
13. RS Group Commands
14. Store File Tracker Commands

Once you entered into HBase shell, just run `help` to get the list of all shell commands.

```
hbase:001:0> help
HBase Shell, version 2.6.0, rde99f8754135ea69adc39da48d2bc2b2710a5366, Mon Apr 29 12:46:30 UTC 2024
Type 'help "COMMAND"', (e.g. 'help "get"' -- the quotes are necessary) for help on a specific command.
Commands are grouped. Type 'help "COMMAND_GROUP"', (e.g. 'help "general"') for help on a command group.

COMMAND GROUPS:
  Group name: general
    Commands: processlist, status, table_help, version, whoami

  Group name: ddl
    Commands: alter, alter_async, alter_status, clone_table_schema, create, describe, disable, disable_all, drop, drop_all
, enable, enable_all, exists, get_table, is_disabled, is_enabled, list, list_disabled_tables, list_enabled_tables, list_
regions, locate_region, show_filters

  Group name: namespace
    Commands: alter_namespace, create_namespace, describe_namespace, drop_namespace, list_namespace, list_namespace_tables

  Group name: dml
    Commands: append, count, delete, deleteall, get, get_counter, get_splits, incr, put, scan, truncate, truncate_preserve

  Group name: tools
    Commands: assign, balance_switch, balancer, balancer_enabled, catalogjanitor_enabled, catalogjanitor_run, catalogjanit
or_switch, cleaner_chore_enabled, cleaner_chore_run, cleaner_chore_switch, clear_block_cache, clear_compaction_queues,
clear_deadservers, clear_slowlog_responses, close_region, compact, compact_rs, compaction_state, compaction_switch, decom
mission_regionservers, flush, flush_master_store, get_balancer_decisions, get_balancer_rejections, get_largelog_respons
es, get_slowlog_responses, hbck_chore_run, is_in_maintenance_mode, list_deadservers, list_decommissioned_regionservers, l
ist_liveservers, list_unknownservers, major_compact, merge_region, move, normalize, normalizer_enabled, normalizer_switc
h, recommission_regionserver, regioninfo, rit, snapshot_cleanup_enabled, snapshot_cleanup_switch, split, splitormerge_en
abled, splitormerge_switch, stop_master, stop_regionserver, trace, truncate_region, unassign, wal_roll, zk_dump

  Group name: replication
    Commands: add_peer, append_peer_exclude_namespaces, append_peer_exclude_tableCFs, append_peer_namespaces, append_peer_
tableCFs, disable_peer, disable_table_replication, enable_peer, enable_table_replication, get_peer_config, list_peer_con
figs, list_peers, list_replicated_tables, peer_modification_enabled, peer_modification_switch, remove_peer, remove_peer_
exclude_namespaces, remove_peer_exclude_tableCFs, remove_peer_namespaces, remove_peer_tableCFs, set_peer_bandwidth, set_
peer_exclude_namespaces, set_peer_exclude_tableCFs, set_peer_namespaces, set_peer_replicate_all, set_peer_serial, set_pe
er_tableCFs, show_peer_tableCFs, update_peer_config
```

To know the usage of each shell command, run `help '<command>'` (*i.e after help, give a space and enter the command name in single quotes*).

For example, run the following to get more help of `status` command

```
help 'status'
```

```
hbase:004:0> help 'status'
Show cluster status. Can be 'summary', 'simple', 'detailed', 'tasks', or 'replication'. The
default is 'summary'. Examples:

  hbase> status
  hbase> status 'summary'
  hbase> status 'simple'
  hbase> status 'detailed'
  hbase> status 'tasks'
  hbase> status 'replication'
  hbase> status 'replication', 'source'
  hbase> status 'replication', 'sink'
hbase:005:0>
```

Some of the HBase general commands include:

- **status**: It provides the HBase system status such as number of servers present in the cluster, active servers count and average load value.
- **version**: It displays the current HBase version value.
- **table_help**: It provides what and how to use table-reference commands such as `create`, `put`, `get`, `scan`, etc.
- **whoami**: It provides information such as user information and groups present in the HBase.
- **processlist**: It provides the total number of tasks executed on the HBase cluster, active running tasks etc.

Some of the HBase namespace commands include:

- **create_namespace**: This command is used to create a namespace in HBase database.
- **list_namespace**: It displays all namespaces present in HBase.
- **describe_namespace**: It provides more information of the given namespace.
- **alter_namespace**: It alters the namespace properties such as adding a new property, modifying the existing property and deleting the existing property.
- **drop_namespace**: It drops the given namespace from HBase.

Some of the HBase table management (Data Definition) commands include:

- **create**: This command is used to create a table in HBase.
- **list**: It displays all tables present in HBase.
- **describe**: It provides more information about column families, associated filters, versions, etc. of the given table.
- **disable**: It disables the given table. It is necessary to disable the table before deleting or dropping it.
- **drop**: It drops the given table from HBase. Make sure that the table is already disabled before dropping it.
- **enable**: It starts enabling the given table in HBase. This command retrieves the disabled table to its previous state.
- **alter**: It alters the column family schema such as altering single or multiple column family names, deleting column families, etc.
- **show_filters**: It displays all filters such as `RowFilter`, `ColumnPrefixFilter`, `TimestampsFilter`, `PageFilter`, `FamilyFilter`, etc that are present in HBase.

Some of the HBase data manipulation commands include:

- **count**: It retrieves the count of rows in a HBase table.

- put: It inserts a cell value at the defined table or row or column and / or timestamp coordinates.
- Append: It appends a cell value at specified table or row or column coordinates.
- get: It retrieves a row or cell contents present in a HBase table. We can also add additional parameters such as TIMERANGE, TIMESTAMP, VERSIONS and FILTERS.
- scan: It scans the entire HBase table and displays the table contents. We can specify optional parameters such as TIMERANGE, TIMESTAMP, FILTER, LIMIT, STARTROW, STOPROW, MAXLENGTH, COLUMNS, CACHE, VERSIONS, etc.
- delete: It deletes a cell value at the defined table of row or column coordinates.
- truncate: It deletes the entire contents in a HBase table. This command internally performs 3 operations - disabling the table, dropping the table and then recreating the table.

Now, we will see how to use the above commands to get the HBase status, version, and create a namespace and a table, insert data into table and retrieve contents of it.

4.1. HBase Status:

On the `hbase>` prompt, run the following command to get the status of HBase cluster.

```
status
```

```
hbase:001:0> status
1 active master, 0 backup masters, 1 servers, 0 dead, 2.0000 average load
Took 1.9250 seconds

hbase:002:0>
```

Here, we can see that there is **1 active master** and **0 backup masters** available out of which **1 server is active** and **0 servers are dead**. The **average load value is 2.00**.

4.2. HBase Version:

Run the following command to get the installed version of HBase.

```
version
```

```
hbase:002:0> version
2.6.0, rde99f8754135ea69adc39da48d2bc2b2710a5366, Mon Apr 29 12:46:30 UTC 2024
Took 0.0010 seconds

hbase:003:0>
```

Here, we can see that HBase version as **2.6.0**.

4.3. Process List:

Run the following command to get the list of process running or completed in HBase.

```
processlist
```

Host	Start Time	State	Description	Status
N...	2024-09-03 09:26:01	COMPLETE	Flushing master:store,,1.1595...	Flush successful flush result:CA

Took 0.7670 seconds
hbase:004:0>

Here, we can see that one process of flushing the master has been completed.

4.4. List Namespace:

On hbase> prompt, use `list_namespace` command to get the list of existing namespaces available on the HBase database as below

```
list_namespace
```

```
hbase:019:0> list_namespace
NAMESPACE
default
hbase
2 row(s)
Took 0.0190 seconds
hbase:020:0>
```

4.5. Create Namespace:

Create a namespace using `create_namespace` command in HBase database. It takes a namespace name as mandatory argument.

Run the following command to create a namespace called 'test' in our HBase

```
create_namespace 'test'
```

```
hbase:012:0> create_namespace 'test'  
Took 0.3490 seconds  
  
hbase:013:0>
```

4.6. Describe Namespace:

Get the existing namespace details using `describe_namespace` command.

Run the following command to get the properties of `test` namespace:

```
describe_namespace 'test'
```

```
hbase:021:0> describe_namespace 'test'  
DESCRIPTION  
  
{NAME => 'test'}  
  
Quota is disabled  
Took 0.0270 seconds  
  
hbase:022:0>
```

4.7. Create Table:

Use `create` command to create a HBase table. It takes a table name and column family as mandatory arguments. The syntax to create a table is as follows:

```
create 'namespace:table_name', 'column_family1', 'column_family2'  
or  
create 'namespace:table_name', {NAME => 'column_family1'}, {NAME =>  
'column_family2'}
```

Note that namespace is optional here and when it is not specified, the table name is created in the default namespace. You can specify the column family as either a simple string or in the form of dictionaries using NAME attribute.

Run the following command to create an `employee` table with `emp` column family and `office` column family in `test` namespace.

```
create 'test:employee', 'emp', 'office'
```

```
hbase:024:0> create 'test:employee', 'emp', 'office'
Created table test:employee
Took 1.3180 seconds
:
=> Hbase::Table - test:employee
hbase:025:0>
```

4.8. List Table:

Use `list` command to show all user tables in HBase

```
list
```

```
hbase:025:0> list
TABLE
test:employee
1 row(s)
Took 0.0480 seconds
=> ["test:employee"]
hbase:026:0>
```

You can also filter the output using regular expression in the `list` command as below

```
list '.*emp.*'
```

```
hbase:016:0> list '.*emp.*'
TABLE
test:employee
1 row(s)
Took 0.0160 seconds
:
=> ["test:employee"]
hbase:017:0>
```

It displays the list of table names containing `emp` string.

4.9. Describe Table:

Use `describe` command to describe details such as version, compression, blocksize, replication, etc. of HBase table.

Run the following command to get the details of `employee` table available in `test` namespace.

```
describe 'test:employee'
```

```

hbase:019:0> describe 'test:employee'
Table test:employee is ENABLED

test:employee, {TABLE_ATTRIBUTES => {METADATA => {'hbase.store.file-tracker.impl' => 'DEFAULT'}}}

COLUMN FAMILIES DESCRIPTION

{NAME => 'emp', INDEX_BLOCK_ENCODING => 'NONE', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSIONS => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', COMPRESSION => 'NONE', BLOCKCACHE => 'true', BLOCKSIZE => '65536 B (64KB)'}

{NAME => 'office', INDEX_BLOCK_ENCODING => 'NONE', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSIONS => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', COMPRESSION => 'NONE', BLOCKCACHE => 'true', BLOCKSIZE => '65536 B (64KB)'}

2 row(s)
Quota is disabled
Took 0.2410 seconds

hbase:020:0>

```

4.10. Insert Data into Table:

Use `put` command to insert data to rows and columns in a HBase table. This is similar to `INSERT` statement in RDBMS but the syntax is different as follows:

```

put 'namespace:table_name', 'row_key', 'column_family:column_name',
'value'

```

Run the following commands to create 3 employee records using their employee IDs as row key and into columns such as name, gender under `emp` column family and address, city, state, country under `office` column family.

```

put 'test:employee', '101', 'emp:name', 'Scott'
put 'test:employee', '101', 'emp:gender', 'M'
put 'test:employee', '102', 'emp:name', 'Mark'
put 'test:employee', '103', 'emp:name', 'Linda'
put 'test:employee', '103', 'emp:gender', 'F'
put 'test:employee', '102', 'office:address', '101 BayHill Drive'
put 'test:employee', '102', 'office:city', 'Bentonville'
put 'test:employee', '102', 'office:state', 'Arkansas'
put 'test:employee', '102', 'office:country', 'USA'
put 'test:employee', '103', 'office:city', 'Tampa'
put 'test:employee', '103', 'office:state', 'Florida'
put 'test:employee', '103', 'office:country', 'USA'
put 'test:employee', '103', 'office:city', 'Jacksonville'

```

```

hbase:039:0> put 'test:employee', '101', 'emp:name', 'Scott'
Took 0.1380 seconds

hbase:040:0> put 'test:employee', '101', 'emp:gender', 'M'
Took 0.0030 seconds

hbase:041:0> put 'test:employee', '102', 'emp:name', 'Mark'
Took 0.0040 seconds

hbase:042:0> put 'test:employee', '103', 'emp:name', 'Linda'
Took 0.0030 seconds

hbase:043:0> put 'test:employee', '103', 'emp:gender', 'F'
Took 0.0030 seconds

hbase:044:0> put 'test:employee', '102', 'office:address', '101 BayHill Drive'
Took 0.0030 seconds

hbase:045:0> put 'test:employee', '102', 'office:city', 'Bentonville'
Took 0.0030 seconds

hbase:046:0> put 'test:employee', '102', 'office:state', 'Arkansas'
Took 0.0020 seconds

hbase:047:0> put 'test:employee', '102', 'office:country', 'USA'
Took 0.0040 seconds

hbase:048:0> put 'test:employee', '103', 'office:city', 'Tampa'
Took 0.0030 seconds

hbase:049:0> put 'test:employee', '103', 'office:state', 'Florida'
Took 0.0050 seconds

hbase:050:0> put 'test:employee', '103', 'office:country', 'USA'
Took 0.0030 seconds

hbase:051:0> put 'test:employee', '103', 'office:city', 'Jacksonville'
Took 0.0110 seconds

hbase:052:0>

```

Note that the last command inserts a new value Jacksonville at city column in office column family for the 103 row key. When you put a new value at the same column for the same row key, then HBase does not do any update but it assigns the same column with a new timestamp so that when you retrieve the data, it fetches the latest data from columns.

4.11. Read Data from Table:

We can use get and scan commands to retrieve data from HBase table depending on whether to retrieve single row or multiple rows of data.

- Use get command to get single row data and its columns from a HBase table.

For example, run the following command to get the row key 102 from employee table:

```
get 'test:employee', '102'
```

```
hbase:021:0> get 'test:employee', '102'
COLUMN                                CELL
                                         timestamp=2024-09-03T11:19:05.937, value=Mark
                                         timestamp=2024-09-03T11:19:37.189, value=101 BayHill Drive
                                         timestamp=2024-09-03T11:19:43.373, value=Bentonville
                                         timestamp=2024-09-03T11:19:52.748, value=USA
                                         timestamp=2024-09-03T11:19:48.119, value=Arkansas
1 row(s)
Took 0.0720 seconds

hbase:022:0>
```

Run the following command to get the row key 103 with emp:name and emp:gender columns only from employee table:

```
get 'test:employee', '103', {COLUMNS =>
  ['emp:name', 'emp:gender']}
```

```
hbase:001:0> get 'test:employee', '103', {COLUMNS => ['emp:name', 'emp:gender']}
COLUMN                                CELL
                                         timestamp=2024-09-03T11:19:15.729, value=F
                                         timestamp=2024-09-03T11:19:11.129, value=Linda
1 row(s)
Took 1.4350 seconds

hbase:002:0>
```

- Use scan command to fetch multiple rows of data from a HBase table. By default, it fetches all data from the table.

Run the following command to fetch all records from employee table available in test namespace.

```
scan 'test:employee'
```

```

hbase:002:0> scan 'test:employee'
ROW                                COLUMN+CELL
101                               column=emp:gender, timestamp=2024-09-03T11:18:59.202, value=M
101                               column=emp:name, timestamp=2024-09-03T11:18:54.252, value=Scott
102                               column=emp:name, timestamp=2024-09-03T11:19:05.937, value=Mark
102                               column=office:address, timestamp=2024-09-03T11:19:37.189, value=101 BayHill Drive
102                               column=office:city, timestamp=2024-09-03T11:19:43.373, value=Bentonville
102                               column=office:country, timestamp=2024-09-03T11:19:52.748, value=USA
102                               column=office:state, timestamp=2024-09-03T11:19:48.119, value=Arkansas
103                               column=emp:gender, timestamp=2024-09-03T11:19:15.729, value=F
103                               column=emp:name, timestamp=2024-09-03T11:19:11.129, value=Linda
103                               column=office:city, timestamp=2024-09-03T11:27:13.116, value=Jacksonville
103                               column=office:country, timestamp=2024-09-03T11:20:17.180, value=USA
103                               column=office:state, timestamp=2024-09-03T11:20:09.770, value=Florida

3 row(s)
Took 0.3960 seconds

hbase:003:0>

```

The `scan` command can take additional parameters in dictionary format to retrieve specific columns with specific rows of data.

Run the below command to retrieve the first 2 rows with `emp:name` and `office:country` columns:

```

scan 'test:employee',{COLUMNS => ['emp:name','office:country'],
LIMIT => 2}

```

```

hbase:019:0> scan 'test:employee',{COLUMNS => ['emp:name','office:country'], LIMIT => 2}
ROW                                COLUMN+CELL
101                               column=emp:name, timestamp=2024-09-03T11:18:54.252, value=Scott
102                               column=emp:name, timestamp=2024-09-03T11:19:05.937, value=Mark
102                               column=office:country, timestamp=2024-09-03T11:19:52.748, value=USA

2 row(s)
Took 0.0230 seconds

hbase:020:0>

```

Run the below command to retrieve `emp:name` and `emp:gender` columns from starting row key 102 and ending row key 103:

```

scan 'test:employee',{COLUMNS => ['emp:name','emp:gender'],
STARTROW => '102', STOPROW => '103'}

```

```
hbase:020:0> scan 'test:employee',{COLUMNS => ['emp:name','emp:gender'], STARTROW => '102', STOPROW => '103'}
ROW                                     COLUMN+CELL
102                               column=emp:name, timestamp=2024-09-03T11:19:05.937, value=Mark
1 row(s)
Took 0.0090 seconds
hbase:021:0>
```

4.12. Disable Table:

Use `disable` command to disable a table which is needed before deleting a table or changing its setting in HBase.

Run the following command to disable `employee` table available in `test` namespace.

```
disable 'test:employee'
```

```
hbase:002:0> disable 'test:employee'
Took 1.5640 seconds

hbase:003:0>
```

Use `is_disabled` command as below to check the disabled status of a table

```
is_disabled 'test:employee'
```

```
hbase:005:0> is_disabled 'test:employee'
true
Took 0.0150 seconds
=> true
hbase:006:0>
```

Here, it displays the status as true which means that our table is disabled already

When you describe the disabled table, it shows the table status as disabled:

```
describe 'test:employee'
```

```

hbase:004:0> describe 'test:employee'
Table test:employee is DISABLED
test:employee, {TABLE_ATTRIBUTES => {METADATA => {'hbase.store.file-tracker.impl' => 'DEFAULT'}}}
COLUMN FAMILIES DESCRIPTION
{NAME => 'emp', INDEX_BLOCK_ENCODING => 'NONE', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSIONS => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', COMPRESSION => 'NONE', BLOCKCACHE => 'true', BLOCKSIZE => '65536 B (64KB)'}

{NAME => 'office', INDEX_BLOCK_ENCODING => 'NONE', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSIONS => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', COMPRESSION => 'NONE', BLOCKCACHE => 'true', BLOCKSIZE => '65536 B (64KB)'}

2 row(s)
Quota is disabled
Took 0.1160 seconds
hbase:005:0>

```

When you try to fetch data from the disabled table, it throws an error:

```

hbase:006:0> scan 'test:employee'
ROW                                     COLUMN+CELL
org.apache.hadoop.hbase.TableNotEnabledException: test:employee is disabled.
    at org.apache.hadoop.hbase.client.ScannerCallable.prepare(ScannerCallable.java:162)
    at org.apache.hadoop.hbase.client.ScannerCallableWithReplicas$RetryingRPC.prepare(ScannerCallableWithReplicas.java:449)
    at org.apache.hadoop.hbase.client.RpcRetryingCallerImpl.callWithRetries(RpcRetryingCallerImpl.java:102)
    at org.apache.hadoop.hbase.client.ResultBoundedCompletionService$QueueingFuture.run(ResultBoundedCompletionService.java:74)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:750)

ERROR: Table test:employee is disabled!

For usage try 'help "scan"'

Took 0.9530 seconds
hbase:007:0>

```

4.13. Enable Table:

Use `enable` command to enable a table which is needed to perform any regular commands

Run the following command to enable `employee` table available in `test` namespace.

```
enable 'test:employee'
```

```

hbase:007:0> enable 'test:employee'
Took 0.7210 seconds

hbase:008:0>

```

Use `is_enabled` command as below to check the enabled status of a table

```
is_enabled 'test:employee'
```

```

hbase:008:0> is_enabled 'test:employee'
true

Took 0.0110 seconds

=> true
hbase:009:0>

```

Here, it displays the status as true which means that our table is enabled already

When you describe the enabled table, it shows the table status as enabled:

```
describe 'test:employee'
```

```

hbase:009:0> describe 'test:employee'
Table test:employee is ENABLED

test:employee, {TABLE_ATTRIBUTES => {METADATA => {'hbase.store.file-tracker.impl' => 'DEFAULT'}}}

COLUMN FAMILIES DESCRIPTION

{NAME => 'emp', INDEX_BLOCK_ENCODING => 'NONE', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSIONS => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', COMPRESSION => 'NONE', BLOCKCACHE => 'true', BLOCKSIZE => '65536 B (64KB)'}

{NAME => 'office', INDEX_BLOCK_ENCODING => 'NONE', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSIONS => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', COMPRESSION => 'NONE', BLOCKCACHE => 'true', BLOCKSIZE => '65536 B (64KB)'}

2 row(s)
Quota is disabled
Took 0.0610 seconds

hbase:010:0>

```

4.14. Delete Rows from Table:

HBase provides `delete` and `deleteall` commands to delete a single cell or multiple cells at a given row in a table.

- Use `delete` command to delete a cell at specified row in a table.
For example, run the following command to remove `office:state` column at 102 row key from `employee` table:

```
delete 'test:employee', '102', 'office:state'
```

```

hbase:015:0> delete 'test:employee', '102', 'office:state'
Took 0.0050 seconds
hbase:016:0>

```

When you scan the `employee` table, you will notice that `office:state` column has been removed at 102 row key only while this column data is present for other rows.

```
scan 'test:employee'
```

```
hbase:020:0> scan 'test:employee'
ROW                                COLUMN+CELL
101                               column=emp:gender, timestamp=2024-09-03T11:18:59.202, value=M
101                               column=emp:name, timestamp=2024-09-03T11:18:54.252, value=Scott
102                               column=emp:name, timestamp=2024-09-03T11:19:05.937, value=Mark
102                               column=office:address, timestamp=2024-09-04T10:41:59.221, value=101 BayHill Drive
102                               column=office:city, timestamp=2024-09-03T11:19:43.373, value=Bentonville
102                               column=office:country, timestamp=2024-09-03T11:19:52.748, value=USA
103                               column=emp:gender, timestamp=2024-09-03T11:19:15.729, value=F
103                               column=emp:name, timestamp=2024-09-03T11:19:11.129, value=Linda
103                               column=office:city, timestamp=2024-09-03T11:27:13.116, value=Jacksonville
103                               column=office:country, timestamp=2024-09-03T11:20:17.180, value=USA
103                               column=office:state, timestamp=2024-09-03T11:20:09.770, value=Florida

3 row(s)
Took 0.0750 seconds

hbase:021:0>
```

- Use `deleteall` command to delete all cells at specified row in a table.

For example, run the following command to remove all columns data at 103 row key from `employee` table:

```
deleteall 'test:employee', '103'
```

```
hbase:021:0> deleteall 'test:employee', '103'
Took 0.0040 seconds

hbase:022:0>
```

When you scan the `employee` table, you will notice that 103 row key has been removed completely.

```
scan 'test:employee'
```

```
hbase:022:0> scan 'test:employee'
ROW                                COLUMN+CELL
101                               column=emp:gender, timestamp=2024-09-03T11:18:59.202, value=M
101                               column=emp:name, timestamp=2024-09-03T11:18:54.252, value=Scott
102                               column=emp:name, timestamp=2024-09-03T11:19:05.937, value=Mark
102                               column=office:address, timestamp=2024-09-04T10:41:59.221, value=101 BayHill Drive
102                               column=office:city, timestamp=2024-09-03T11:19:43.373, value=Bentonville
102                               column=office:country, timestamp=2024-09-03T11:19:52.748, value=USA
2 row(s)
Took 0.0260 seconds
hbase:023:0>
```

4.15. Drop Table:

Use drop command to delete a table from HBase database. Note that you must disable the table before dropping it.

Run the following commands to disable employee table and then drop it in test namespace.

```
disable 'test:employee'
drop 'test:employee'
```

```
hbase:023:0> disable 'test:employee'
Took 1.0810 seconds
hbase:024:0> drop 'test:employee'
Took 0.4580 seconds
hbase:025:0>
```

HBase also provides drop_all command to delete multiple tables using a regular expression as below:

```
drop_all '.*emp.*'
```

```
hbase:025:0> drop_all '.*emp.*'
No tables matched the regex .*emp.*
Took 0.0250 seconds

hbase:026:0>
```

5. HBase User Interface:

Each HBase service provides a simple Web UI, which gives us the basic information about the current cluster state, change log level, access logs and many more. These Web UIs are useful for troubleshooting and diagnostics. The Web UIs are embedded using the Jetty Web Server (which has a low memory footprint and fast response times), so they start when we run the relevant service.

5.1. Master Web UI:

The Master server web UI helps to check the overall health of HBase system and displays the currently running Region servers, backup masters and HBase tables. By default, the Web UI runs on 16010 port and is accessible at <http://localhost:16010/>. If multiple masters are running, only the active master shows any information.

The screenshot shows the Apache HBase Master Web UI interface. At the top, there's a navigation bar with links like Home, Table Details, Procedures & Locks, HBCK Report, Operation Details, Process Metrics, Local Logs, Log Level, Debug Dump, Metrics, Profiler, HBase Configuration, and Startup Progress. Below the navigation bar, it says "Master [REDACTED].1".

Region Servers

Base Stats

ServerName	State	Start time	Last contact	Version	Requests Per Second	Num. Regions
[REDACTED],16020,1725422178900	Normal	Wed Sep 04 09:26:18 IST 2024	0 s	2.6.0	0	2

Total:1

Backup Masters

ServerName	Port	Start Time

Total:0

Tables

User Tables

Region Visualizer

Total 'storefileSize' per Region Server

Table

- hbase meta
- hbase namespace

Region Server

The Master web UI shows the created tables and their definition (*such as ColumnFamilies, blocksize, etc.*). It also lists the summary of available RegionServers in the cluster stating how many regions each RegionServer is serving, how many requests per second they are handling, usedHeap, maxheap, etc.

5.2. Region Web UI:

HBase also provides a web UI for Region Server runs on 16030 port by default and is accessible at <http://localhost:16030/>. This web interface provides the detailed information about the performance of each Region server and data it is providing.

The screenshot displays the Apache HBase Region Server status page at <http://localhost:16030/rs-status>. The top navigation bar includes links for Home, Process Metrics, Local Logs, Operation Details, Log Level, Debug Dump, Metrics, Profiler, and HBase Configuration. The main content is organized into several sections:

- RegionServer** [REDACTED] 16020,1725422178900
- Server Metrics**: A table showing Request Per Second (0), Num. Regions (2), Block locality (100.000%), and Block locality (Secondary replicas) (0.000%). A note indicates "Slow WAL Append Count" with a value of 0.
- Block Cache**: A table showing Attribute (Implementation), Value (LruBlockCache), and Description (Block cache implementing class). A note says "See block cache in the HBase Reference Guide for help."
- Tasks**: Buttons for Show All Monitored Tasks, Show non-RPC Tasks (which is selected), Show All RPC Handler Tasks, Show Active RPC Calls, and Show Client Operations. A note says "View as JSON" and "No tasks currently running on this node."
- Regions**: A table showing Region Name (hbase:meta,,1 and hbase:namespace,,1725016681591.53fbdb28269f64baeb71b2553afe7d1), Start Key, End Key, ReplicID, and a note indicating the number of regions (0).

A detailed note at the bottom of the Regions section explains region names and their relationship to the table `domains,apache.org,5464829424211263407`.

The **Server Metrics** section tells us how each Region server is working by showing how many regions it is hosting and how many requests per second it is serving. The **Tasks** sections displays the long-running tasks and their status.

6. Access HBase from Hive:

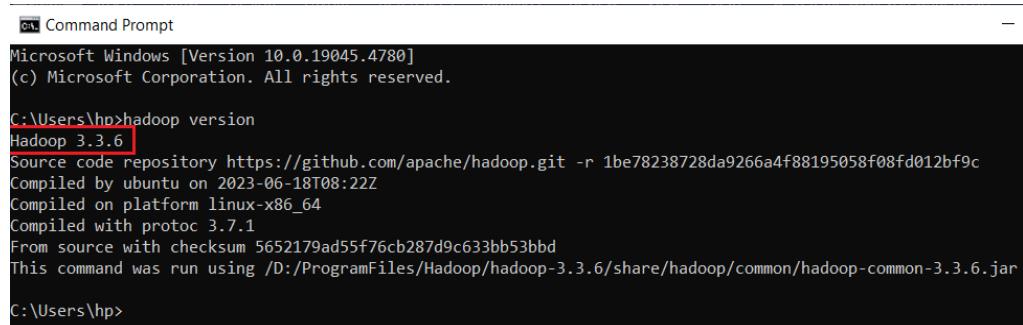
Since we are more comfortable on running SQL like queries on any database, we can leverage Apache Hive to access HBase and get the benefit of SQL syntax and run queries on HBase tables. To do so, we must use **HBaseStorageHandler** java class from `hive-hbase-handler.jar` file to register HBase tables with the Hive metastore.

First, make sure that Apache Hive is installed in your machine. If not already installed, follow [this guide](#) and install it.

6.1. Verify Hadoop Installation:

Open **Windows PowerShell** or **Command Prompt** and run the below command to verify if the Hadoop version:

```
hadoop version
```



```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>hadoop version
Hadoop 3.3.6
Source code repository https://github.com/apache/hadoop.git -r 1be78238728da9266a4f88195058f08fd012bf9c
Compiled by ubuntu on 2023-06-18T08:22Z
Compiled on platform linux-x86_64
Compiled with protoc 3.7.1
From source with checksum 5652179ad55f76cb287d9c633bb53bbd
This command was run using /D:/ProgramFiles/Hadoop/hadoop-3.3.6/share/hadoop/common/hadoop-common-3.3.6.jar

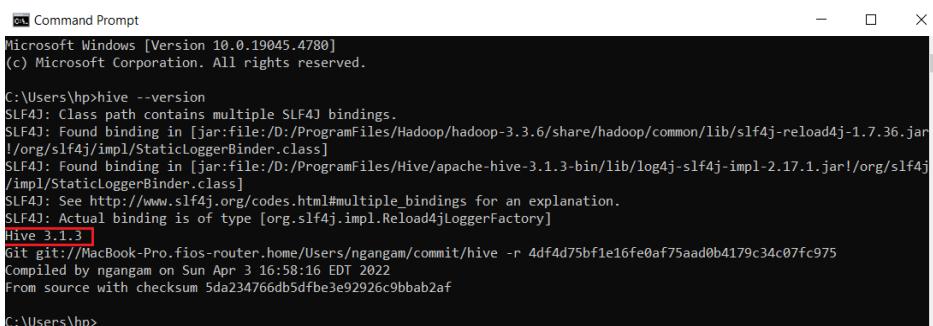
C:\Users\hp>
```

It displays **Hadoop 3.3.6** version that has been installed.

6.2. Verify Hive Installation:

Open **Windows PowerShell** or **Command Prompt** and run the below command to verify if the Hive version:

```
hive --version
```



```
Microsoft Windows [Version 10.0.19045.4780]
(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/Hadoop/hadoop-3.3.6/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
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: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
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 5da234766db5dfbe3e92926c9bbab2af

C:\Users\hp>
```

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

6.3. Verify Hive HBase Handler:

Go to %HIVE_HOME%\lib directory and verify `hive-hbase-handler-x.x.x.jar` file is available.

Name	Date modified	Type	Size
<code>hive-common-3.1.3.jar</code>	4/4/2022 2:30 AM	Executable Jar File	127 KB
<code>hive-contrib-3.1.3.jar</code>	4/4/2022 2:31 AM	Executable Jar File	50,815 KB
<code>hive-druid-handler-3.1.3.jar</code>	4/4/2022 2:30 AM	Executable Jar File	40,892 KB
<code>hive-hbase-handler-3.1.3.jar</code>	4/4/2022 2:31 AM	Executable Jar File	118 KB
<code>hive-hcatalog-core-3.1.3.jar</code>	4/4/2022 2:31 AM	Executable Jar File	265 KB
<code>hive-hcatalog-server-extensions-3.1.3.jar</code>	4/4/2022 2:31 AM	Executable Jar File	77 KB

Note: If you can't find `hive-hbase-handler-x.x.x.jar` file in your Hive installation path, you can download it for the corresponding Hive version from [Maven Repository](#) and place it in %HIVE_HOME%\lib location.

6.4. 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.

```
D:  
cd %HADOOP_HOME%\sbin  
start-dfs.cmd  
start-yarn.cmd
```

```
Administrator: Command Prompt  
Microsoft Windows [Version 10.0.19045.4780]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Windows\system32>d:  
  
D:\>cd %HADOOP_HOME%\sbin  
  
D:\ProgramFiles\Hadoop\hadoop-3.3.6\sbin>start-dfs.cmd  
  
D:\ProgramFiles\Hadoop\hadoop-3.3.6\sbin>start-yarn.cmd  
starting yarn daemons  
  
D:\ProgramFiles\Hadoop\hadoop-3.3.6\sbin>
```

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
e been reached.
2024-09-10 23:18:48,680 INFO ipc.Server: IPC Server Responder: starting
2024-09-10 23:18:48,680 INFO ipc.Server: IPC Server listener on 9820: starting
2024-09-10 23:18:48,686 INFO namenode.NameNode: NameNode RPC up at: localhost/127.0.0.1:9820
2024-09-10 23:18:48,695 INFO namenode.FSNamesystem: Starting services required for active state
2024-09-10 23:18:48,696 INFO namenode.FSDirectory: Initializing quota with 12 thread(s)
2024-09-10 23:18:48,715 INFO namenode.FSDirectory: Quota initialization completed in 19 milliseconds
name space=445
storage space=308051118
storage types=RAM_DISK=0, SSD=0, DISK=0, ARCHIVE=0, PROVIDED=0
2024-09-10 23:18:48,741 INFO blockmanagement.CacheReplicationMonitor: Starting CacheReplicationMonitor with interval 300
00 milliseconds
2024-09-10 23:18:50,729 INFO hdfs.StateChange: BLOCK* registerDatanode: from DatanodeRegistration(127.0.0.1:9866, datano
deUuid=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
2024-09-10 23:18:50,732 INFO net.NetworkTopology: Adding a new node: /default-rack/127.0.0.1:9866
2024-09-10 23:18:50,733 INFO blockmanagement.BlockReportLeaseManager: Registered DN 123a143e-536d-495e-a2ae-789991cf74d4
(127.0.0.1:9866).
2024-09-10 23:18:50,858 INFO blockmanagement.DatanodeDescriptor: Adding new storage ID DS-51797c33-6f1d-4c19-80d1-e4f469
81927d for DN 127.0.0.1:9866
2024-09-10 23:18:50,970 INFO BlockStateChange: BLOCK* processReport 0xa684def46d04a5d7 with lease ID 0x1aa96cb0cc0586df:
Processing first storage report for DS-51797c33-6f1d-4c19-80d1-e4f46981927d from datanode 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)
2024-09-10 23:18:50,983 INFO blockmanagement.BlockManager: initializing replication queues
2024-09-10 23:18:50,984 INFO hdfs.StateChange: STATE* Safe mode extension entered.
The reported blocks 302 has reached the threshold 0.9990 of total blocks 303. The minimum number of live datanodes is no
t required. In safe mode extension. Safe mode will be turned off automatically in 29 seconds.
2024-09-10 23:18:50,987 INFO BlockStateChange: BLOCK* processReport 0xa684def46d04a5d7 with lease ID 0x1aa96cb0cc0586df:
from storage DS-51797c33-6f1d-4c19-80d1-e4f46981927d node DatanodeRegistration(127.0.0.1:9866, datanodeU
```

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

```
Apache Hadoop Distribution - hadoop datanode
2024-09-10 23:18:50,588 INFO impl.FsDatasetImpl: Time to add replicas to map for block pool BP-1739737951-192.16620109937 on volume D:\ProgramFiles\Hadoop\hadoop-3.3.6\data\dfs\datanode: 75ms
2024-09-10 23:18:50,589 INFO impl.FsDatasetImpl: Total time to add all replicas to map for block pool BP-1739737951-192.16620109937: 78ms
2024-09-10 23:18:50,591 INFO checker.ThrottledAsyncChecker: Scheduling a check for D:\ProgramFiles\Hadoop\hadoop-3.3.6\data\dfs\datanode
2024-09-10 23:18:50,604 INFO checker.DatasetVolumeChecker: Scheduled health check for volume D:\ProgramFiles\Hadoop\hadoop-3.3.6\data\dfs\datanode
2024-09-10 23:18:50,627 INFO datanode.VolumeScanner: VolumeScanner(D:\ProgramFiles\Hadoop\hadoop-3.3.6\data\dfs\datanode, DS-51797c33-6f1d-4c19-80d1-e4f46981927d): no suitable block pools found to scan. Waiting 779084883 ms.
2024-09-10 23:18:50,631 WARN datanode.DirectoryScanner: dfs.datanode.directoryscan.throttle.limit.ms.per.sec set to value above 1000 ms/sec. Assuming default value of -1
2024-09-10 23:18:50,632 INFO datanode.DirectoryScanner: Periodic Directory Tree Verification scan starting in 10936830ms with interval of 21600000ms and throttle limit of -1ms/s
2024-09-10 23:18:50,641 INFO datanode.DataNode: Block pool BP-1739737951-192.168.1.5-1716620109937 (Datanode Uuid 123a143e-536d-495e-a2ae-789991cf74d4) service to localhost/127.0.0.1:9820 beginning handshake with NN
2024-09-10 23:18:50,755 INFO datanode.DataNode: Block pool BP-1739737951-192.168.1.5-1716620109937 (Datanode Uuid 123a143e-536d-495e-a2ae-789991cf74d4) service to localhost/127.0.0.1:9820 successfully registered with NN
2024-09-10 23:18:50,757 INFO datanode.DataNode: For namenode localhost/127.0.0.1:9820 using BLOCKREPORT_INTERVAL of 2160000msecs CACHEREPORT_INTERVAL of 10000msecs Initial delay: 0msecs; heartBeatInterval=3000
2024-09-10 23:18:50,757 INFO datanode.DataNode: Starting IBR Task Handler.
2024-09-10 23:18:50,915 INFO datanode.DataNode: After receiving heartbeat response, updating state of namenode localhost:9820 to active
2024-09-10 23:18:51.052 INFO datanode.DataNode: Successfully sent block report 0xa684def46d04a5d7 with lease ID 0x1aa96cb0cc0586df to namenode: localhost/127.0.0.1:9820, containing 1 storage report(s), of which we sent 1. The reports had 303 total blocks and used 1 RPC(s). This took 16 msecs to generate and 120 msecs for RPC and NN processing. Got back one command: FinalizeCommand/5.
2024-09-10 23:18:51.054 INFO datanode.DataNode: Got finalize command for block pool BP-1739737951-192.16620109937
```

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

```
Apache Hadoop Distribution - yarn resourcemanager
2024-09-10 23:23:31,363 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.server.api.ResourceTrackerPB to the server
2024-09-10 23:23:31,364 INFO ipc.Server: IPC Server listener on 8031: starting
2024-09-10 23:23:31,364 INFO ipc.Server: IPC Server Responder: starting
2024-09-10 23:23:31,386 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2024-09-10 23:23:31,424 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue, queueCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false.
2024-09-10 23:23:31,431 INFO ipc.Server: Listener at 0.0.0.0:8030
2024-09-10 23:23:31,434 INFO ipc.Server: Starting Socket Reader #1 for port 8030
2024-09-10 23:23:31,444 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationMasterProtocolPB to the server
2024-09-10 23:23:31,445 INFO ipc.Server: IPC Server listener on 8030: starting
2024-09-10 23:23:31,447 INFO ipc.Server: IPC Server Responder: starting
2024-09-10 23:23:31,689 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue, queueCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false.
2024-09-10 23:23:31,690 INFO ipc.Server: Listener at 0.0.0.0:8032
2024-09-10 23:23:31,693 INFO ipc.Server: Starting Socket Reader #1 for port 8032
2024-09-10 23:23:31,697 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationClientProtocolPB to the server
2024-09-10 23:23:31,698 INFO ipc.Server: IPC Server listener on 8032: starting
2024-09-10 23:23:31,700 INFO ipc.Server: IPC Server Responder: starting
2024-09-10 23:23:31,996 INFO resourcemanager.ResourceTrackerService: NodeManager from node [REDACTED] (cmPort: 4815 httpPort: 8042) registered with capability: <memory:8192, vCores:8>, assigned nodeId [REDACTED]:4815
2024-09-10 23:23:31,999 INFO rmnode.RMNodeImpl: [REDACTED]:4815 Node Transitioned from NEW to RUNNING
2024-09-10 23:23:32,096 INFO capacity.CapacityScheduler: Added node [REDACTED]:4815 clusterResource: <memory:8192, vCores:8>
2024-09-10 23:23:32,675 INFO webproxy.ProxyCA: Created Certificate for OU=YARN-7ae8f4d7-9aff-4573-a0fc-0307725e58f7
2024-09-10 23:23:32,797 INFO recovery.RMStateStore: Storing CA Certificate and Private Key
2024-09-10 23:23:32,798 INFO resourcemanager.ResourceManager: Transitioned to active state
```

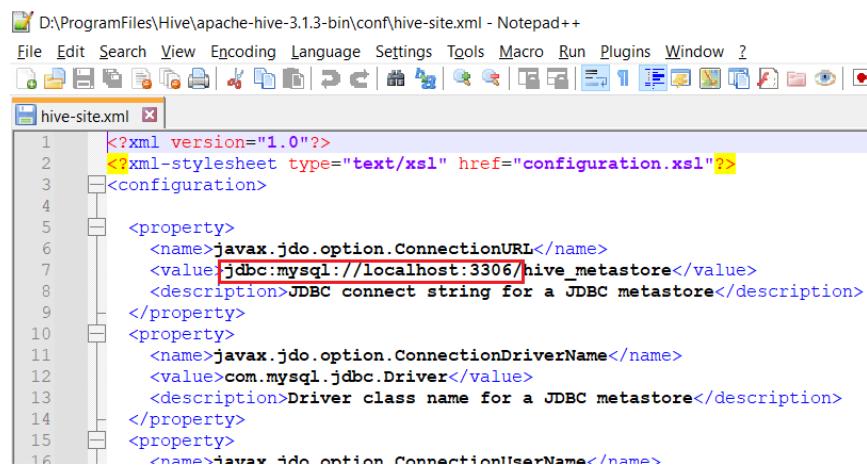
Wait for nodemanager service to say “*Registered with ResourceManager*”.

```
Apache Hadoop Distribution - yarn nodemanager
Sep 10, 2024 11:23:28 PM com.sun.jersey.spi.container.GuiceComponentProviderFactory register
INFO: Registering org.apache.hadoop.yarn.server.nodemanager.webapp.JAXBContextResolver as a provider class
Sep 10, 2024 11:23:28 PM com.sun.jersey.server.impl.application.WebApplicationImpl _initiate
INFO: Initiating Jersey application, version 'Jersey: 1.19.4 05/24/2017 03:20 PM'
Sep 10, 2024 11:23:28 PM com.sun.jersey.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.JAXBContextResolver to GuiceManagedComponentProvider with the scope "Singleton"
Sep 10, 2024 11:23:29 PM com.sun.jersey.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.webapp.GenericExceptionHandler to GuiceManagedComponentProvider with the scope "Singleton"
Sep 10, 2024 11:23:30 PM com.sun.jersey.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.NMWebServices to GuiceManagedComponentProvider with the scope "Singleton"
2024-09-10 23:23:30,630 INFO handler.ContextHandler: Started o.e.j.w.WebAppContext@2a4f5433{node/,file:///C:/Users/hp/AppData/Local/Temp/jetty-0_0_0-8042-hadoop-yarn-common-3_3_6_jar-_any-1986172007106471223/webapp/,AVAILABLE}{jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/share/hadoop/yarn/hadoop-yarn-common-3.3.6.jar!/webapps/node}
2024-09-10 23:23:30,649 INFO server.AbstractConnector: Started ServerConnector@263f04ca[HTTP/1.1, (http/1.1)]{0.0.0.0:8042}
2024-09-10 23:23:30,650 INFO server.Server: Started @17763ms
2024-09-10 23:23:30,650 INFO webapp.WebApps: Web app node started at 8042
2024-09-10 23:23:30,654 INFO nodemanager.NodeStatusUpdaterImpl: Node ID assigned is : [REDACTED] 4815
2024-09-10 23:23:30,661 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2024-09-10 23:23:30,675 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8031
2024-09-10 23:23:32,048 INFO security.NMContainerTokenSecretManager: Rolling master-key for container-tokens, got key with id 1254362717
2024-09-10 23:23:32,057 INFO security.NMTokenSecretManagerInNM: Rolling master-key for container-tokens, got key with id -1338438860
2024-09-10 23:23:32,095 INFO nodemanager.NodeStatusUpdaterImpl: Registered with ResourceManager as [REDACTED] :4815 with total resource of <memory:8192, vCores:8>
```

6.5. Start Hive Metastore Service:

It is necessary to start Hive metastore service if your Hive metastore is configured in Remote database.

Verify your Hive metastore configuration in %HIVE_HOME%\conf\hive-site.xml file in which you can look for javax.jdo.option.ConnectionURL property which tells the metastore connectivity. In my case, Hive metastore is connected to MySQL running locally.



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

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

Since Hive is able to connect to remote metastore using Thrift URIs, start the metastore service using the below command:

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

```

Command Prompt - hive --service metastore --hiveconf hive.root.logger=console
Microsoft Windows [Version 10.0.19045.4780]
(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]
2024-09-11T00:29:32,124 INFO [main] metastore.HiveMetaStore: STARTUP_MSG:
/*****
STARTUP_MSG: Starting Hive Metastore...
STARTUP_MSG: host = D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\accumulo-core-1.7.3.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\accumulo-fate-1.7.3.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\accumulo-start-1.7.3.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\accumulo-trace-1.7.3.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\aircompressor-0.10.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\antlr-runtime-3.5.2.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\antlr4-runtime-4.5.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\apoolalliance-repackaged-2.5.0-b32.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\apache-jsp-9.3.20.v20170531.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\apache-jstl-9.3.20.v20170531.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\arrow-format-0.8.0.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\arrow-memory-0.8.0.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\arrow-vector-0.8.0.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\asm-5.0.1.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\asm-commons-5.0.1.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\asm-tree-5.0.1.jar;D:\Program Files\Hive\apache-hive-3.1.3-bin\lib\audience-annotations-0.5.0.jar;D:\Program
2024-09-11T00:29:32,674 INFO [main] conf.MetastoreConf: Unable to find config file hivemetastore-site.xml
2024-09-11T00:29:32,674 INFO [main] conf.MetastoreConf: Found configuration file null
2024-09-11T00:29:32,677 INFO [main] conf.MetastoreConf: Unable to find config file metastore-site.xml
2024-09-11T00:29:32,677 INFO [main] conf.MetastoreConf: Found configuration file null
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.
2024-09-11T00:29:33,899 INFO [main] hikari.HikariDataSource: HikariPool-1 - Starting...
2024-09-11T00:29:33,908 WARN [main] util.DriverDataSource: Registered driver with driverClassName=com.mysql.jdbc.Driver was not found, trying direct instantiation.
2024-09-11T00:29:37,325 INFO [main] hikari.HikariDataSource: HikariPool-1 - Start completed.
2024-09-11T00:29:37,446 INFO [main] hikari.HikariDataSource: HikariPool-2 - Starting...
2024-09-11T00:29:37,448 WARN [main] util.DriverDataSource: Registered driver with driverClassName=com.mysql.jdbc.Driver was not found, trying direct instantiation.
2024-09-11T00:29:37,510 INFO [main] hikari.HikariDataSource: HikariPool-2 - Start completed.
2024-09-11T00:29:38,297 INFO [main] metastore.ObjectStore: Setting MetaStore object pin classes with hive.metastore.cache.pinobjtypes="Table,StorageDescriptor,SerDeInfo,Partition,Database,Type,FieldSchema,Order"
2024-09-11T00:29:39,002 INFO [main] metastore.MetaStoreDirectSql: Using direct SQL, underlying DB is MySQL
2024-09-11T00:29:39,005 INFO [main] metastore.ObjectStore: Initialized ObjectStore
2024-09-11T00:29:46,413 INFO [main] metastore.HiveMetaStore: Added admin role in metastore
2024-09-11T00:29:46,422 INFO [main] metastore.HiveMetaStore: Added public role in metastore
2024-09-11T00:29:46,545 INFO [main] metastore.HiveMetaStore: No user is added in admin role, since config is empty
2024-09-11T00:29:46,714 INFO [main] conf.HiveConf: Found configuration file file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/conf/hive-site.xml
2024-09-11T00:29:47,147 INFO [main] metastore.HiveMetaStore: Starting DB backed MetaStore Server with SetUGI enabled
2024-09-11T00:29:47,157 INFO [main] metastore.HiveMetaStore: Started the new metaserver on port [9083]...
2024-09-11T00:29:47,157 INFO [main] metastore.HiveMetaStore: Options.minWorkerThreads = 200
2024-09-11T00:29:47,158 INFO [main] metastore.HiveMetaStore: Options.maxWorkerThreads = 1000
2024-09-11T00:29:47,159 INFO [main] metastore.HiveMetaStore: TCP keepalive = true
2024-09-11T00:29:47,160 INFO [main] metastore.HiveMetaStore: Enable SSL = false

```

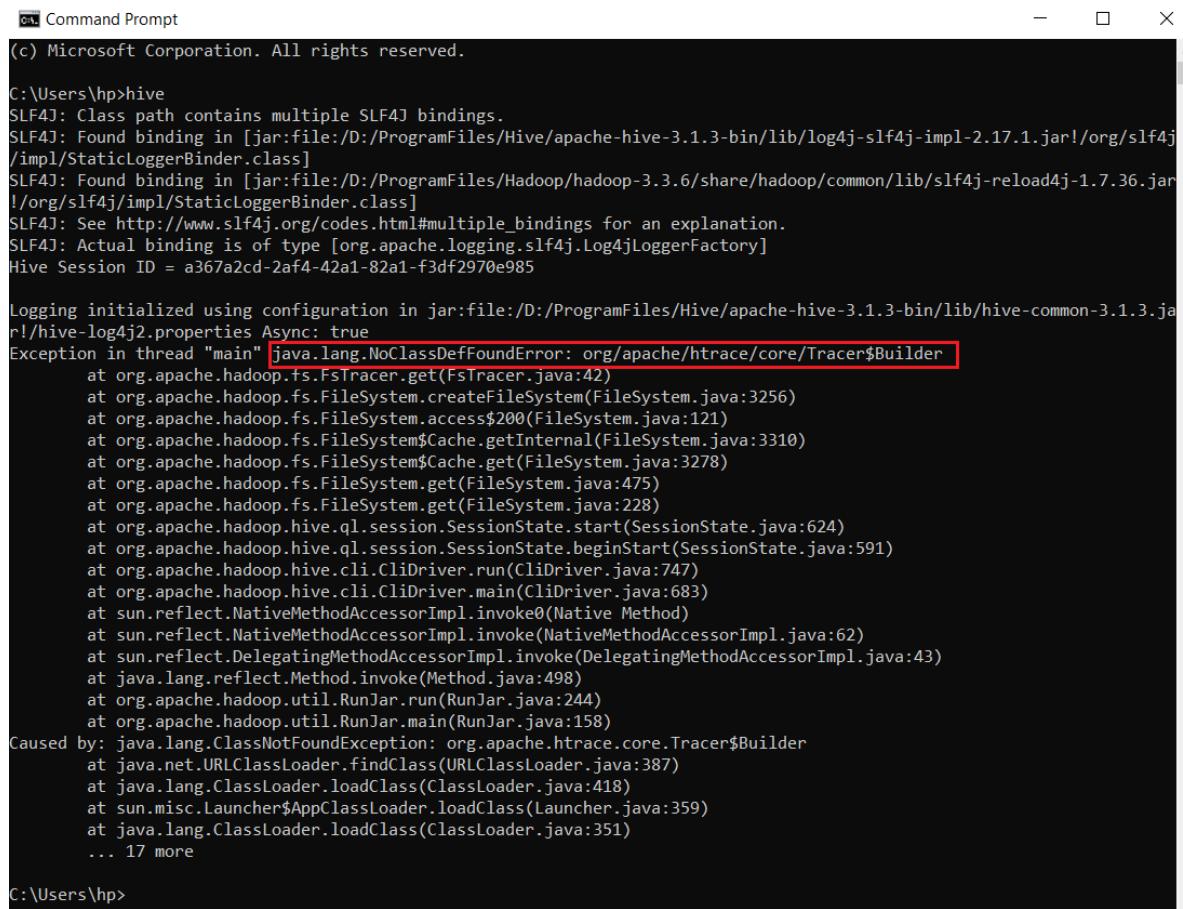
On the console, we can see that **metaserver** started on port **9083** and connected to underlying MySQL database.

6.6. Launch Hive Shell:

Now, open **Command Prompt** or **Windows PowerShell** and run the following command to launch Hive shell.

hive

Note: Sometimes, you might face an error `java.lang.NoClassDefFoundError: org/apache/htrace/core/Tracer$Builder` as below:



```
(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 = a367a2cd-2af4-42a1-f3df2970e985

Logging initialized using configuration in jar:file:/D:/ProgramFiles/Hive/apache-hive-3.1.3-bin/lib/hive-common-3.1.3.ja
r!/hive-log4j2.properties Async: true
Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/htrace/core/Tracer$Builder
    at org.apache.hadoop.fs.FsTracer.get(FsTracer.java:42)
    at org.apache.hadoop.fs.FileSystem.createFileSystem(FileSystem.java:3256)
    at org.apache.hadoop.fs.FileSystem.access$200(FileSystem.java:121)
    at org.apache.hadoop.fs.FileSystem$Cache.getInternal(FileSystem.java:3310)
    at org.apache.hadoop.fs.FileSystem$Cache.get(FileSystem.java:3278)
    at org.apache.hadoop.fs.FileSystem.get(FileSystem.java:475)
    at org.apache.hadoop.fs.FileSystem.get(FileSystem.java:228)
    at org.apache.hadoop.hive.ql.session.SessionState.start(SessionState.java:624)
    at org.apache.hadoop.hive.ql.session.SessionState.beginStart(SessionState.java:591)
    at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:747)
    at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:683)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at org.apache.hadoop.util.RunJar.run(RunJar.java:244)
    at org.apache.hadoop.util.RunJar.main(RunJar.java:158)
Caused by: java.lang.ClassNotFoundException: org.apache.htrace.core.Tracer$Builder
    at java.net.URLClassLoader.findClass(URLClassLoader.java:387)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:418)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:359)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:351)
    ... 17 more
C:\Users\hp>
```

This happens when `htrace-core.jar` file is missing in `%HBASE_HOME%\lib` directory. To resolve this, go to `%HBASE_HOME%\lib\client-facing-thirdparty` location and copy `htrace-core4-x.x.x-incubating.jar` file to `%HBASE_HOME%\lib` directory.

Name	Date modified	Type	Size
hbase-testing-util-2.6.0.jar	1/22/2020 8:40 PM	Executable Jar File	25 KB
hbase-thrift-2.6.0.jar	1/22/2020 8:40 PM	Executable Jar File	3,950 KB
hbase-unsafe-4.1.7.jar	1/22/2020 8:40 PM	Executable Jar File	20 KB
hbase-zookeeper-2.6.0.jar	1/22/2020 8:40 PM	Executable Jar File	107 KB
hbase-zookeeper-2.6.0-testsjar	1/22/2020 8:40 PM	Executable Jar File	71 KB
HikariCP-java7-2.4.12.jar	1/22/2020 8:40 PM	Executable Jar File	132 KB
htrace-core4-4.1.0-incubating.jar	1/22/2020 8:40 PM	Executable Jar File	1,468 KB
httpclient-4.5.13.jar	1/22/2020 8:40 PM	Executable Jar File	763 KB
httpcore-4.4.13.jar	1/22/2020 8:40 PM	Executable Jar File	321 KB

Additionally, I would suggest you to set the following properties which are helpful to disable unnecessary logging on the Hive console:

- Set environment variable `HADOOP_USER_CLASSPATH_FIRST` to 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
```

Now, we should be able to launch the hive shell successfully

```
C:\Users\hp>hive
Microsoft Windows [Version 10.0.19045.4780]
(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 = e321f555-6a7e-4aa9-b25d-6efad1ba41f2

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>
```

6.7. Create HBase Table:

Use the following syntax to create a new HBase table managed by Hive using the `STORED BY` clause pointed to `HBaseStorageHandler` class.

```
CREATE TABLE hive_table_name (key datatype, hive_column datatype)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key",
hbase_column_family:hbase_column")
TBLPROPERTIES ("hbase.table.name" = "hbase_table_name");
```

Here, the `hbase.columns.mapping` property is required and is used to map the column names between HBase and Hive tables. The `hbase.table.name` property is optional and it controls the name of the table as known by HBase, and allows the Hive table to have a different name. If not specified, then the Hive and HBase table names will be identical.

On the Hive shell, run the following command to create `hbase_departments` table in Hive with HBase table name as `departments`:

```
CREATE TABLE hbase_departments (dept_id int, name string, manager
string, city string, state string, country string, pincode int)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH
SERDEPROPERTIES ("hbase.columns.mapping" = ":key, dept:name,
```

```
dept:manager, location:city, location:state, location:country,  
location:pincode") TBLPROPERTIES ("hbase.table.name" =  
"departments");
```

```
hive> CREATE TABLE hbase_departments (dept_id int, name string, manager string, city string, state string, country string, pincode int) STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key, dept:name, dept:manager, location:city, location:state, location:country, location:pincode") TBLPROPERTIES ("hbase.table.name" = "departments");  
OK  
Time taken: 7.618 seconds  
hive>
```

Describe the newly created table with this hive command:

```
describe formatted hbase_departments;
```

```
hive> describe formatted hbase_departments;  
OK  
# col_name          data_type          comment  
dept_id            int  
name               string  
manager             string  
city               string  
state              string  
country             string  
pincode            int  
  
# Detailed Table Information  
Database:          default  
OwnerType:         USER  
Owner:              hp  
CreateTime:        Wed Sep 11 19:54:06 IST 2024  
LastAccessTime:    UNKNOWN  
Retention:         0  
Location:          hdfs://localhost:9820/user/hive/warehouse/hbase_departments  
Table Type:        MANAGED_TABLE  
Table Parameters:  
COLUMN_STATS_ACCURATE  {"BASIC_STATS":"true","COLUMN_STATS":{ "city":"true", "country":"true", "dept_id":"true", "manager":"true", "name":"true", "pincode":"true", "state":"true"} }  
bucketing_version     2  
hbase.table.name      departments  
numFiles              0  
 numRows              0  
 rawDataSize          0  
 storage_handler       org.apache.hadoop.hive.hbase.HBaseStorageHandler  
 totalSize             0
```

6.8. Insert Data into HBase Table:

Run the following command to insert record into `hbase_departments` table.

```
INSERT INTO hbase_departments VALUES (10, 'Finance', 'Steve',  
'Houston', 'Texas', 'USA', 633001);
```

This Insert statement triggers a MapReduce job that will get executed on the Hadoop cluster. Here, you might get an **Execution error** as shown below

```

hive> INSERT INTO hbase_departments VALUES (10, 'Finance', 'Steve', 'Houston', 'Texas', 'USA', 633001);
Query ID = hp_20240911222452_f2fceb02-40fe-4b59-9f82-3ee69a01ff97
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1726072508090_0001, Tracking URL = http://DESKTOP-KGH2E2G:8088/proxy/application_1726072508090_0001/
Kill Command = D:\ProgramFiles\Hadoop\hadoop-3.3.6\bin\mapred job -kill job_1726072508090_0001
Hadoop job information for Stage-2: number of mappers: 0; number of reducers: 0
2024-09-11 22:25:43,431 Stage-2 map = 0%, reduce = 0%
Ended Job = job_1726072508090_0001 with errors
Error during job, obtaining debugging information...
FAILED: Execution Error, return code 2 from org.apache.hadoop.hive.ql.exec.mr.MapRedTask
MapReduce Jobs Launched:
Stage-Stage-2: HDFS Read: 0 HDFS Write: 0 FAIL
Total MapReduce CPU Time Spent: 0 msec
hive>

```

The details of the error can be seen on the Hadoop YARN UI at <http://localhost:8088/cluster>

- On the YARN UI, click on the Application ID.

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State
application_1726072508090_0001	hp	INSERT INTO hbase_departments VALUES (10, 'Finance', 'Steve', 'Houston', 'Texas', 'USA', 633001) (Stage-2)	MAPREDUCE		default	0	Wed Sep 11 22:25:18 +0550 2024	Wed Sep 11 22:25:21 +0550 2024	N/A	ACCEPTED

- Click on Logs link available for one of the attempts.

Attempt ID	Started	Node	Logs	Nodes blacklisted by the app	Nodes blacklisted by the system
appattempt_1726072508090_0001_000002	Wed Sep 11 22:25:35 +0550 2024	http://192.168.56.1:8042	Logs	0	0
appattempt_1726072508090_0001_000001	Wed Sep 11 22:25:18 +0550 2024	http://192.168.56.1:8042	Logs	0	0

- Select **syslog** link.

The screenshot shows a web interface for Hadoop logs. On the left, there's a sidebar with links for 'ResourceManager', 'NodeManager', and 'Tools'. The main area is titled 'Logs for container_1726072508090_0001_02_000001'. Under this title, it says 'Local Logs:' followed by several log entries. One entry, 'syslog', is highlighted with a red border.

```

directory.info : Total file length is 8893.bytes.
launch_container.cmd : Total file length is 17356.bytes.
stderr : Total file length is 243.bytes.
stdout : Total file length is 0.bytes.
syslog : Total file length is 11295.bytes.

```

- You can see a detailed error message here

```

java.lang.ClassCastException:
org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$GetFileInfoRequestProto cannot be cast to com.google.protobuf.Message

```

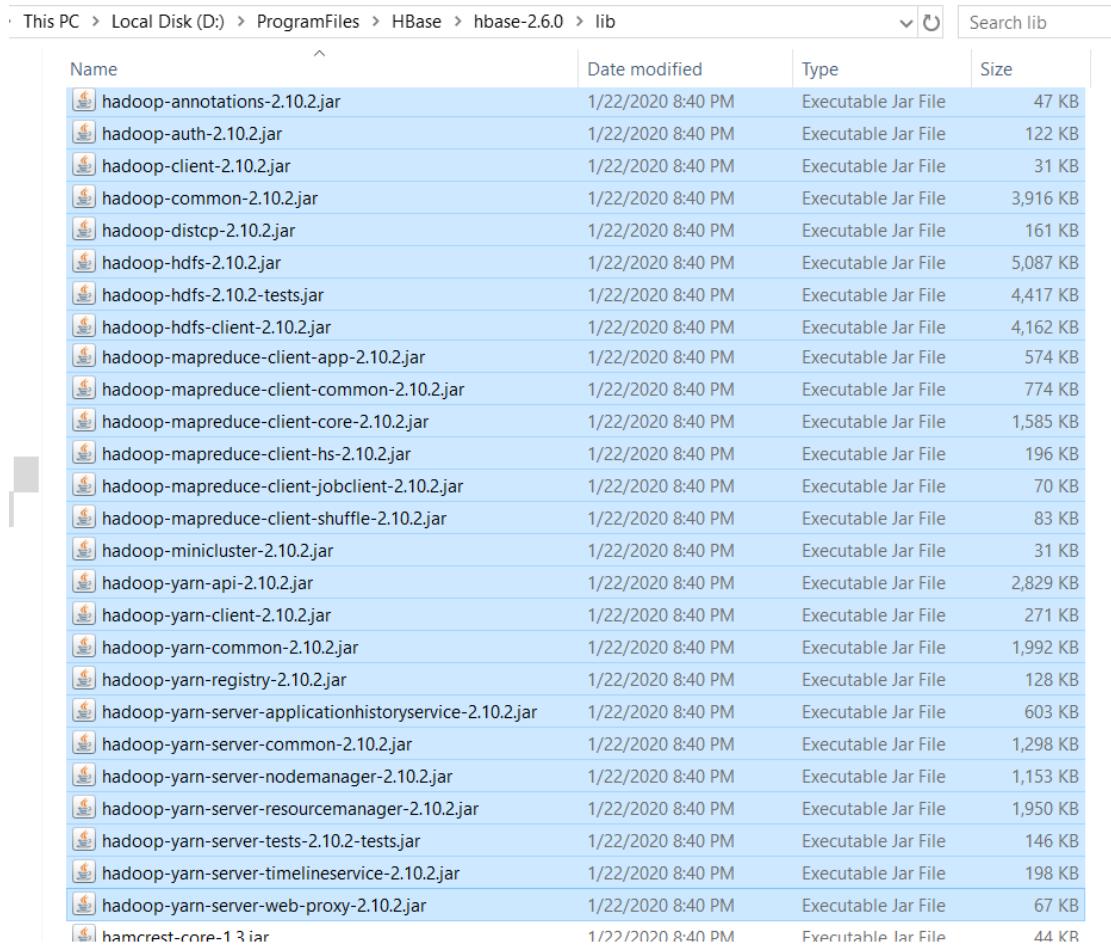
This screenshot shows a detailed error stack trace from the logs. The error message 'java.lang.ClassCastException' is at the top, followed by a long stack trace involving multiple classes from org.apache.hadoop.hdfs.protocol.proto, org.apache.hadoop.mapreduce.v2.app, and org.apache.hadoop.security. The stack trace continues through various methods like 'getFileInfo', 'NativeMethodInvocationHandler', and 'FileLinkResolver'.

```

java.lang.ClassCastException:
org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$GetFileInfoRequestProto cannot be cast to com.google.protobuf.Message
at com.sun.prism.impl.SPIImplementation.getMethodInfo(Unknown Source)
at org.apache.hadoop.hdfs.DFSClient.getFileInfo(DFSClient.java:1739)
at org.apache.hadoop.hdfs.DistributedFileSystem$29.doCall(DistributedFileSystem.java:1829)
at org.apache.hadoop.hdfs.DistributedFileSystem$29.doCall(DistributedFileSystem.java:1826)
at org.apache.hadoop.fs.FileSystemLinkResolver.resolve(FileSystemLinkResolver.java:81)
at org.apache.hadoop.hdfs.DistributedFileSystem.getFileStatus(DistributedFileSystem.java:1841)
at org.apache.hadoop.fs.FileSystem.exists(FileSystem.java:1862)
at org.apache.hadoop.mapreduce.v2.app.MRAppMaster.serviceInit(MRAppMaster.java:317)
at org.apache.hadoop.service.AbstractService.startInternal(AbstractService.java:164)
at org.apache.hadoop.service.AbstractService.access$000(AbstractService.java:160)
at org.apache.hadoop.service.AbstractService$2.run(MRAppMaster.java:1760)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Subject.java:422)
at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1899)
at org.apache.hadoop.mapreduce.v2.app.MRAppMaster.initAndStartAppMaster(MRAppMaster.java:1757)
at org.apache.hadoop.mapreduce.v2.app.MRAppMaster.main(MRAppMaster.java:1691)
2014-07-11 12:25:41,118 ERROR [main] org.apache.hadoop.mapreduce.v2.app.MRAppMaster: Error: testing #MRAppMaster
at java.lang.Class.getDeclaredMethods0(Native Method)
at java.lang.Class.privateGetDeclaredMethods(Unknown Source)
at org.apache.hadoop.hdfs.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcEngine.java:247)
at org.apache.hadoop.hdfs.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcEngine.java:132)
at com.sun.proxy.$Proxy9.getFileInfo(Unknown Source)
at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolTranslatorPB.getFileInfo(ClientNamenodeProtocolTranslatorPB.java:966)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.apache.hadoop.ipc.RpcServer$Handler$Call.invoke(RpcServer.java:831)
at org.apache.hadoop.ipc.RetryInvocationHandler$Call.invoke(RetryInvocationHandler.java:166)
at org.apache.hadoop.ipc.RetryInvocationHandler$Call.invoke(RetryInvocationHandler.java:158)
at org.apache.hadoop.ipc.RetryInvocationHandler$Call.invokeOnce(RetryInvocationHandler.java:96)
at org.apache.hadoop.ipc.RpcServer$Handler$Call.invoke(RpcServer.java:362)
at com.sun.proxy.$Proxy10.getFileInfo(Unknown Source)
at org.apache.hadoop.hdfs.DFSClient.getFileInfo(DFSClient.java:1739)
at org.apache.hadoop.hdfs.DistributedFileSystem$29.doCall(DistributedFileSystem.java:1829)
at org.apache.hadoop.hdfs.DistributedFileSystem$29.doCall(DistributedFileSystem.java:1826)
at org.apache.hadoop.fs.FileSystemLinkResolver.resolve(FileSystemLinkResolver.java:81)
at org.apache.hadoop.fs.DistributedFileSystem.getFileStatus(DistributedFileSystem.java:1841)
at org.apache.hadoop.fs.FileSystem.exists(FileSystem.java:1862)
at org.apache.hadoop.mapreduce.v2.app.MRAppMaster.access$160(MRAppMaster.java:317)
at org.apache.hadoop.mapreduce.service.AbstractService.int$(AbstractService.java:164)
at org.apache.hadoop.mapreduce.v2.app.MRAppMaster$6.run(MRAppMaster.java:1760)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Subject.java:422)
at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1899)
at org.apache.hadoop.mapreduce.v2.app.MRAppMaster.initAndStartAppMaster(MRAppMaster.java:1757)

```

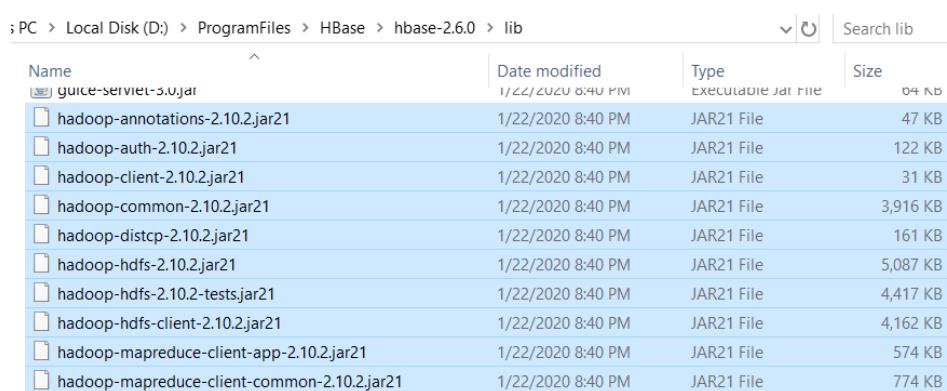
The above ClassCastException occurs when HBase is referring to different version of Hadoop libraries compared to libraries available on the existing Hadoop cluster. Go to %HBASE_HOME%\lib location where you can see all hadoop jars are of **2.10.2** version while our Hadoop is actually running on **3.3.6** version.



Name	Date modified	Type	Size
hadoop-annotations-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	47 KB
hadoop-auth-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	122 KB
hadoop-client-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	31 KB
hadoop-common-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	3,916 KB
hadoop-distcp-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	161 KB
hadoop-hdfs-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	5,087 KB
hadoop-hdfs-2.10.2-tests.jar	1/22/2020 8:40 PM	Executable Jar File	4,417 KB
hadoop-hdfs-client-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	4,162 KB
hadoop-mapreduce-client-app-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	574 KB
hadoop-mapreduce-client-common-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	774 KB
hadoop-mapreduce-client-core-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	1,585 KB
hadoop-mapreduce-client-hs-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	196 KB
hadoop-mapreduce-client-jobclient-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	70 KB
hadoop-mapreduce-client-shuffle-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	83 KB
hadoop-minicluster-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	31 KB
hadoop-yarn-api-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	2,829 KB
hadoop-yarn-client-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	271 KB
hadoop-yarn-common-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	1,992 KB
hadoop-yarn-registry-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	128 KB
hadoop-yarn-server-applicationhistoryservice-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	603 KB
hadoop-yarn-server-common-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	1,298 KB
hadoop-yarn-server-nodemanager-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	1,153 KB
hadoop-yarn-server-resourcemanager-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	1,950 KB
hadoop-yarn-server-tests-2.10.2-tests.jar	1/22/2020 8:40 PM	Executable Jar File	146 KB
hadoop-yarn-server-timelineservice-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	198 KB
hadoop-yarn-server-web-proxy-2.10.2.jar	1/22/2020 8:40 PM	Executable Jar File	67 KB
hamcrest-core-1.3.jar	1/22/2020 8:40 PM	Executable Jar File	44 KB

Follow the below steps to resolve the ClassCastException issue:

- Close **hive terminal** window, **Hive metastore service** window and **hbase-master start** window.
- Remove or rename all `hadoop*.jar` files in `%HBASE_HOME%\lib` location (*here, I renamed all `hadoop*.jar` files instead of deleting them*)



Name	Date modified	Type	Size
guice-servlet-3.0.jar	1/22/2020 8:40 PM	Executable Jar File	04 KB
hadoop-annotations-2.10.2.jar21	1/22/2020 8:40 PM	JAR21 File	47 KB
hadoop-auth-2.10.2.jar21	1/22/2020 8:40 PM	JAR21 File	122 KB
hadoop-client-2.10.2.jar21	1/22/2020 8:40 PM	JAR21 File	31 KB
hadoop-common-2.10.2.jar21	1/22/2020 8:40 PM	JAR21 File	3,916 KB
hadoop-distcp-2.10.2.jar21	1/22/2020 8:40 PM	JAR21 File	161 KB
hadoop-hdfs-2.10.2.jar21	1/22/2020 8:40 PM	JAR21 File	5,087 KB
hadoop-hdfs-2.10.2-tests.jar21	1/22/2020 8:40 PM	JAR21 File	4,417 KB
hadoop-hdfs-client-2.10.2.jar21	1/22/2020 8:40 PM	JAR21 File	4,162 KB
hadoop-mapreduce-client-app-2.10.2.jar21	1/22/2020 8:40 PM	JAR21 File	574 KB
hadoop-mapreduce-client-common-2.10.2.jar21	1/22/2020 8:40 PM	JAR21 File	774 KB

- Open **Windows Powershell as Administrator** and run the following command to start HBase cluster.

```
start-hbase.cmd
```

```
PS C:\Windows\system32> start-hbase.cmd
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/HBase/hbase-2.6.0/lib/client-facing-thirdparty/log4j-slf4j-impl-2.17.2.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]
PS C:\Windows\system32>
```

- Open a new Command Prompt and start the metastore service using the below command:

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

```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>hive --service metastore --hiveconf hive.root.logger=console
"Starting Hive Metastore"
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]
2024-09-11T23:06:07,810 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-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\apacheds-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\avro-5.0.1.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\hadoop-mapreduce-client-app-3.1.3-bin.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\hadoop-mapreduce-client-common-3.1.3-bin.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\hadoop-mapreduce-client-core-3.1.3-bin.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\hadoop-mapreduce-client-jobclient-3.1.3-bin.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\hadoop-mapreduce-client-shuffle-3.1.3-bin.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\hadoop-mapreduce-client-tasktrackers-3.1.3-bin.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\hadoop-mapreduce-client-trackers-3.1.3-bin.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\hadoop-mapreduce-client-tokentrackers-3.1.3-bin.jar;D:\ProgramFiles\Hive\apache-hive-3.1.3-bin\lib\hadoop-mapreduce-client-龥
```

- Open a new Command Prompt, launch Hive shell and run the following insert statement:

```
INSERT INTO hbase_departments VALUES (10, 'Finance', 'Steve', 'Houston', 'Texas', 'USA', 633001);
```

It submitted a new application to the Hadoop YARN and provided application URL which is successful and inserted record into table.

```

Command Prompt - hive
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 = 0226860d-c88e-4041-8839-acb6747a94bd

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 = 5626c5e6-c515-4e3f-9774-1628b876a408
hive> INSERT INTO hbase_departments VALUES (10, 'Finance', 'Steve', 'Houston', 'Texas', 'USA', 633001);
Query ID = np_20240911231022_a04390eb-0e44-435f-ab83-0d93efdfc3b0t
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1726072508090_0002, Tracking URL = http://DESKTOP-KGH2E2G:8088/proxy/application_1726072508090_0002/
Kill Command = D:\ProgramFiles\Hadoop\hadoop-3.3.6\bin\mapred job -kill job_1726072508090_0002
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 0
2024-09-11 23:11:00,193 Stage-2 map = 0%, reduce = 0%
2024-09-11 23:11:15,134 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 8.108 sec
MapReduce Total cumulative CPU time: 8 seconds 108 msec
Ended Job = job_1726072508090_0002
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1 Cumulative CPU: 8.108 sec HDFS Read: 14040 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 8 seconds 108 msec
OK
Time taken: 54.608 seconds
hive>

```

You can notice the same on the YARN UI at <http://localhost:8088/cluster> where the latest application status is FINISHED successfully.

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State
application_1726072508090_0002	hp	INSERT INTO hbase_departments VALUES(10, 'Finance', 'Steve', 'Houston', 'Texas', 'USA', 633001)	MAPREDUCE		default	0	Wed Sep 11 23:10:41 +0550 2024	Wed Sep 11 23:10:41 +0550 2024	Wed Sep 11 23:11:15 +0550 2024	FINISHED
application_1726072508090_0001	hp	INSERT INTO hbase_departments VALUES(10, 'Finance', 'Steve', 'Houston', 'Texas', 'USA', 633001)	MAPREDUCE		default	0	Wed Sep 11 22:26:18 +0550 2024	Wed Sep 11 22:26:21 +0550 2024	Wed Sep 11 22:26:42 +0550 2024	FAILED

Let us quickly verify the data in the table by running the following `select` statement.

```
select * from hbase_departments;
```

```

hive> select * from hbase_departments;
OK
10      Finance Steve    Houston Texas    USA      633001
Time taken: 1.09 seconds, Fetched: 1 row(s)
hive>

```

Note: To view the column names in the above result, set `hive.cli.print.header` property to `true` and run the `select` statement.

```
set hive.cli.print.header=true;
select * from hbase_departments;
```

6.9. Validate Newly Created Table in HBase:

Open new Command Prompt and start hbase shell.

```
D:
cd %HBASE_HOME%\bin
hbase shell
```

On `hbase>` prompt, list the available tables

```
list
```

```
hbase:001:0> list
TABLE
departments
test:employee
2 row(s)
Took 1.1280 seconds
=> ["departments", "test:employee"]
hbase:002:0>
```

Here, you can see that `departments` table is available which got created from Hive.

Describe the `departments` table:

```
describe "departments"
```

```

hbase:002:0> describe "departments"
Table departments is ENABLED

departments, {TABLE_ATTRIBUTES => {METADATA => {'hbase.store.file-tracker.impl' => 'DEFAULT'}}}

COLUMN FAMILIES DESCRIPTION

{NAME => 'dept', INDEX_BLOCK_ENCODING => 'NONE', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSIONS => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', COMPRESSION => 'NONE', BLOCKCACHE => 'true', BLOCKSIZE => '65536 B (64KB)', METADATA => {'EVICT_BLOCKS_ON_CLOSE' => 'false', 'NEW_VERSION_BEHAVIOR' => 'false', 'CACHE_DATA_ON_WRITE' => 'false', 'CACHE_INDEX_ON_WRITE' => 'false', 'CACHE_BLOOMS_ON_WRITE' => 'false', 'PREFETCH_BLOCKS_ON_OPEN' => 'false', 'CACHE_DATA_IN_L1' => 'false'}}}

{NAME => 'location', INDEX_BLOCK_ENCODING => 'NONE', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSIONS => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', COMPRESSION => 'NONE', BLOCKCACHE => 'true', BLOCKSIZE => '65536 B (64KB)', METADATA => {'EVICT_BLOCKS_ON_CLOSE' => 'false', 'NEW_VERSION_BEHAVIOR' => 'false', 'CACHE_DATA_ON_WRITE' => 'false', 'CACHE_INDEX_ON_WRITE' => 'false', 'CACHE_BLOOMS_ON_WRITE' => 'false', 'PREFETCH_BLOCKS_ON_OPEN' => 'false', 'CACHE_DATA_IN_L1' => 'false'}}}

2 row(s)
Quota is disabled
Took 0.6860 seconds
hbase:003:0>

```

Scan the `departments` table to see its contents:

```
scan "departments"
```

```

hbase:003:0> scan "departments"
ROW          COLUMN+CELL
10           column=dept:manager, timestamp=2024-09-11T23:11:14.519, value=Steve
10           column=dept:name, timestamp=2024-09-11T23:11:14.519, value=Finance
10           column=location:city, timestamp=2024-09-11T23:11:14.519, value=Houston
10           column=location:country, timestamp=2024-09-11T23:11:14.519, value=USA
10           column=location:pincode, timestamp=2024-09-11T23:11:14.519, value=633001
10           column=location:state, timestamp=2024-09-11T23:11:14.519, value=Texas

1 row(s)
Took 0.1750 seconds
hbase:004:0>

```

Here, you can see that one record of **row key 10** is available in `departments` table.

6.10. Create External HBase Table:

If you want to give Hive access to an existing HBase table, then use `CREATE EXTERNAL TABLE` command on Hive. The syntax is similar to `CREATE TABLE` command:

```
CREATE EXTERNAL TABLE hive_table_name (key datatype, hive_column
datatype)
```

```
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'  
WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key,  
hbase_column_family:hbase_column")  
TBLPROPERTIES ("hbase.table.name" = "hbase_table_name");
```

Go to hive terminal and run the following command to create hbase_external_employees table in Hive:

```
CREATE EXTERNAL TABLE hbase_external_employees (emp_id int, name  
string, gender string, address string, city string, state string,  
country string) STORED BY  
'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH  
SERDEPROPERTIES ("hbase.columns.mapping" = ":key, emp:name,  
emp:gender, office:address, office:city, office:state,  
office:country") TBLPROPERTIES ("hbase.table.name" =  
"test:employee");
```

```
hive> CREATE EXTERNAL TABLE hbase_external_employees (emp_id int, name string, gender string, address string, city string, state string, country string) STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key, emp:name, emp:gender, office:address, office:city, office:state, office:country") TBLPROPERTIES ("hbase.table.name" = "test:employee");  
OK  
Time taken: 0.43 seconds  
hive>
```

External table has been created successfully which means this table is pointed to the existing HBase table “test:employee”

6.11. Fetch HBase Data in Hive:

Now, let us run fetch records from hbase_external_employees table:

```
select * from hbase_external_employees;
```

```
hive> select * from hbase_external_employees;  
OK  
hbase_external_employees.emp_id hbase_external_employees.name hbase_external_employees.gender hbase_external_employees.  
.address hbase_external_employees.city hbase_external_employees.state hbase_external_employees.country  
101 Scott M NULL NULL NULL  
102 Mark NULL 101 BayHill Drive Bentonville NULL USA  
103 Linda F NULL Jacksonville Florida USA  
Time taken: 0.48 seconds, Fetched: 3 row(s)  
hive>
```

It displays 3 records which are available in the existing HBase “test:employee” table.

Similarly, we can run SQL queries on Hive which pulls data from HBase.

```
select count(emp_id) from hbase_external_employees;
```

```

hive> select count(emp_id) from hbase_external_employees;
2024-09-12 00:02:20,526 main ERROR Unable to delete file C:\Users\hp\AppData\Local\Temp\hp\hive.log: java.nio.file.FileSystemException C:\Users\hp\AppData\Local\Temp\hp\hive.log: The process cannot access the file because it is being used by another process.

Query ID = hp_20240912000220_41a5eb33-2320-4264-9996-f9340250ddb4
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_1726072508090_0003, Tracking URL = http://DESKTOP-KGH2E2G:8088/proxy/application_1726072508090_0003/
Kill Command = D:\ProgramFiles\Hadoop\hadoop-3.3.6\bin\mapred job -kill job_1726072508090_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-09-12 00:02:47,531 Stage-1 map = 0%, reduce = 0%
2024-09-12 00:03:04,416 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 9.186 sec
2024-09-12 00:03:17,440 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 15.012 sec
MapReduce Total cumulative CPU time: 15 seconds 12 msec
Ended Job = job_1726072508090_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 15.012 sec HDFS Read: 22377 HDFS Write: 101 SUCCESS
Total MapReduce CPU Time Spent: 15 seconds 12 msec
OK
_c0
B
Time taken: 59.257 seconds, Fetched: 1 row(s)
hive> Display all 634 possibilities? (y or n)

```

It submits a MapReduce job and upon successful, it displays the count of records in the table.

7. Shutdown Standalone HBase:

We can shutdown the standalone Hbase cluster in the Windows environment.

Open **Windows PowerShell as Administrator** and shutdown the entire HBase cluster using the below command:

```
hbase master stop -shutDownCluster
```

Note:

If you would like to stop hbase using **Command Prompt**, then you must first navigate to the location where HBase is installed as below.

```
D:
cd %HBASE_HOME%\bin
hbase master stop -shutDownCluster
```

```

C:\ Command Prompt
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>d:
D:\>cd %HBASE_HOME%\bin

D:\ProgramFiles\HBase\hbase-2.6.0\bin>hbase master stop -shutDownCluster
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/ProgramFiles/HBase/hbase-2.6.0/lib/client-facing-thirdparty/log4j-slf4j-impl-2.17.2.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]
2024-08-29T20:06:57.777+0530: [GC (Allocation Failure) 2024-08-29T20:06:57.777+0530: [ParNew: 51328K->6034K(57728K), 0.0080291 secs] 51328K->6034K(186112K), 0.0085397 secs] [Times: user=0.06 sys=0.00, real=0.01 secs]
2024-08-29T20:06:58.102+0530: [GC (Allocation Failure) 2024-08-29T20:06:58.102+0530: [ParNew: 57362K->4266K(57728K), 0.0183610 secs] 57362K->6881K(186112K), 0.0191050 secs] [Times: user=0.06 sys=0.00, real=0.02 secs]
2024-08-29 20:06:58,515 main ERROR Unable to locate appender "DRFAS" for logger config "SecurityLogger"
2024-08-29T20:06:58,610 INFO [main {}] master.HMaster: STARTING service HMaster
2024-08-29T20:06:58,621 INFO [main {}] util.VersionInfo: HBase 2.6.0
2024-08-29T20:06:58,621 INFO [main {}] util.VersionInfo: Source code repository git://704fb001264a/home/bbeaudreault/hbase-rm/output/hbase revision=de9f8754135ea69adc39da48d2bc2b2710a5366
2024-08-29T20:06:58,622 INFO [main {}] util.VersionInfo: Compiled by bbeaudreault on Mon Apr 29 12:46:30 UTC 2024
2024-08-29T20:06:58,622 INFO [main {}] util.VersionInfo: From source with checksum 53aebc4dde5541d474f2d82be8dafadcf912d4619a8adb45779ac52ceeed717b79cc20f341a28964f3b7ea3f48de67bafe7fc13ebdfb110d876b65a8e471a63a
2024-08-29T20:06:58.637+0530: [GC (Allocation Failure) 2024-08-29T20:06:58.637+0530: [ParNew: 55594K->3343K(57728K), 0.0038885 secs] 58209K->5958K(186112K), 0.0042304 secs] [Times: user=0.00 sys=0.00, real=0.00 secs]

```

This will shut down three daemon services HRegion, HMaster and Zookeeper (*you will see the same in the another Command Prompt where HBase Distribution was running*)

```

Administrator: HBase Distribution
2024-08-29T20:07:11,815 INFO [M:0;DESKTOP-KGH2E2G:16000 {}] server.SyncRequestProcessor: Shutting down
2024-08-29T20:07:11,815 INFO [ProcessThread(sid:0 cport:2181): {}] server.PrepRequestProcessor: PrepRequestProcessor exited loop!
2024-08-29T20:07:11,818 INFO [SyncThread:0 {}] server.SyncRequestProcessor: SyncRequestProcessor exited!
2024-08-29T20:07:11,818 INFO [M:0;DESKTOP-KGH2E2G:16000 {}] server.FinalRequestProcessor: shutdown of request processor complete
2024-08-29T20:07:11,841 INFO [M:0;DESKTOP-KGH2E2G:16000 {}] client.FourLetterWordMain: connecting to localhost 2181
2024-08-29T20:07:13,327 INFO [SessionTracker {}] server.SessionTrackerImpl: SessionTrackerImpl exited loop!
2024-08-29T20:07:13,863 INFO [M:0;DESKTOP-KGH2E2G:16000 {}] zookeeper.MiniZooKeeperCluster: Shutdown MiniZK cluster with all ZK servers
2024-08-29T20:07:13,881 INFO [shutdown-hook-0 {}] regionserver.ShutdownHook: Shutdown hook starting; hbase.shutdown.hook=true; fsShutdownHook=org.apache.hadoop.fs.FileSystem$Cache$ClientFinalizer@56afdf9a
2024-08-29T20:07:13,881 INFO [shutdown-hook-0 {}] regionserver.ShutdownHook: Shutdown hook finished.
2024-08-29T20:07:13,885 INFO [shutdown-hook-0 {}] regionserver.ShutdownHook: Shutdown hook starting; hbase.shutdown.hook=true; fsShutdownHook=org.apache.hadoop.fs.FileSystem$Cache$ClientFinalizer@56afdf9a
2024-08-29T20:07:13,886 INFO [shutdown-hook-0 {}] regionserver.ShutdownHook: Starting fs shutdown hook thread.
2024-08-29T20:07:13,889 INFO [shutdown-hook-0 {}] regionserver.ShutdownHook: Shutdown hook finished.

Heap
par new generation total 57728K, used 39241K [0x00000005c000000, 0x00000005cfea0000, 0x00000005e0cc0000)
eden space 51328K, 63% used [0x00000005c000000, 0x00000005ce012530, 0x00000005cf220000)
from space 6400K, 100% used [0x00000005cf220000, 0x00000005cf860000, 0x00000005cf860000)
to space 6400K, 0% used [0x00000005cf860000, 0x00000005cf860000, 0x00000005cfea0000)
concurrent mark-sweep generation total 128384K, used 44105K [0x00000005e0cc0000, 0x00000005e8a20000, 0x00000007c000000)

Metaspace      used 75567K, capacity 77964K, committed 78132K, reserved 1118208K
  class space   used 9114K, capacity 9587K, committed 9592K, reserved 1048576K

```

Congratulations!! You have successfully installed Standalone HBase 2.6 in Windows operating system and executed few HBase commands. You also understood how to access HBase tables from Hive and run SQL type queries on HBase.