# Hive

https://downloads.apache.org//db/derby/db-derby-10.14.2.0/db-derby-10.14.2.0-bin.tar.gz

https://www.7-zip.org/download.html
https://www.cygwin.com/
https://downloads.apache.org/hive/hive-3.1.2/

While working on a project, we were asked to install Apache Hive on a Windows 10 operating system. Many guides were found online but unfortunately, they didn’t work. For this reason, I decided to write a step-by-step guide to help others.
The starting point of this guide was from a great video I found on Youtube which provides a working scenario for Hive 2.x without much detail.
This article is a part of a series that we are publishing on TowardsDataScience.com

that aims to illustrate how to install Big Data technologies on Windows operating system.

**Other published articles in this series:**

- Installing Hadoop 3.2.1 Single node cluster on Windows 10
- Installing Apache Pig 0.17.0 on Windows 10

## 1. Prerequisites

### 1.1. 7zip

In order to extract tar.gz archives, you should install the 7zip tool.

### 1.2. Installing Hadoop

To install Apache Hive, you must have a Hadoop Cluster installed and running: You can refer to our previously published step-by-step guide to install Hadoop 3.2.1 on Windows 10.

### 1.3. Apache Derby

In addition, Apache Hive requires a relational database to create its Metastore (where all metadata will be stored). In this guide, we will use the Apache Derby database 4.

Since we have Java 8 installed, we must

install Apache Derby 10.14.2.0 version (check downloads page) which can be downloaded from the following link.

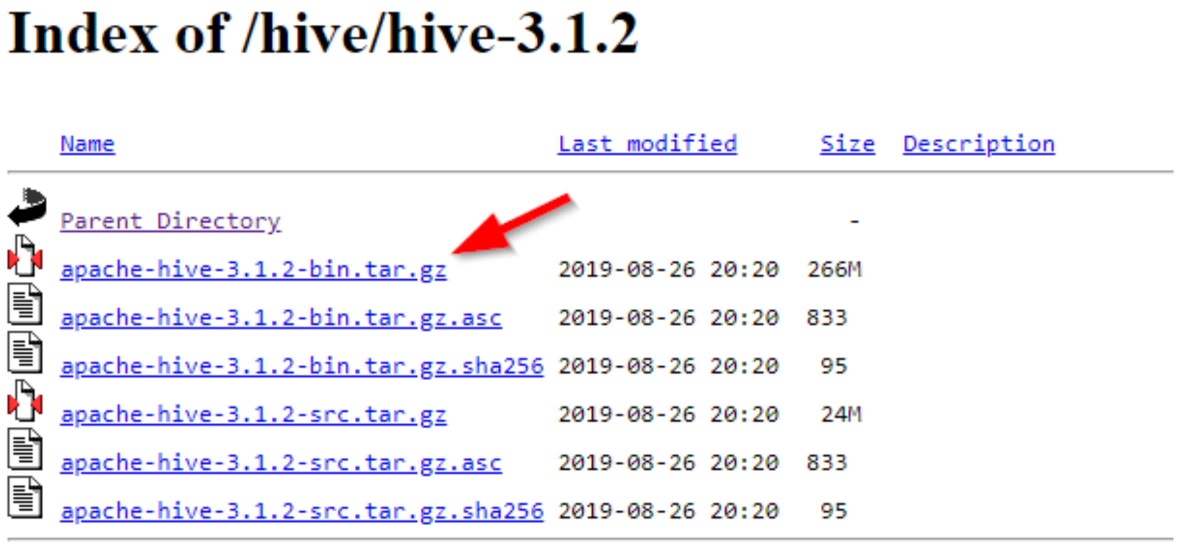
Once downloaded, we must extract twice *(using 7zip: the first time we extract the .tar.gz file, the second time we extract the .tar file)* the content of the db-derby-10.14.2.0-bin.tar.gz archive into the desired installation directory. Since in the previous guide we have installed Hadoop within "E:\hadoop-env\hadoop-3.2.1\" directory, we will extract Derby into "E:\hadoop-env\db-derby-10.14.2.0\" directory.

### 1.4. Cygwin

Since there are some Hive 3.1.2 tools that aren't compatible with Windows (such as schematool). We will need the Cygwin tool to run some Linux commands.

## 2. Downloading Apache Hive binaries

In order to download Apache Hive binaries, you should go to the following website: https://downloads.apache.org/hive/hive-3.1.2/. Then, download the apache-hive-3.1.2.-bin.tar.gz file.



# Index of /hive/hive-3.1.2

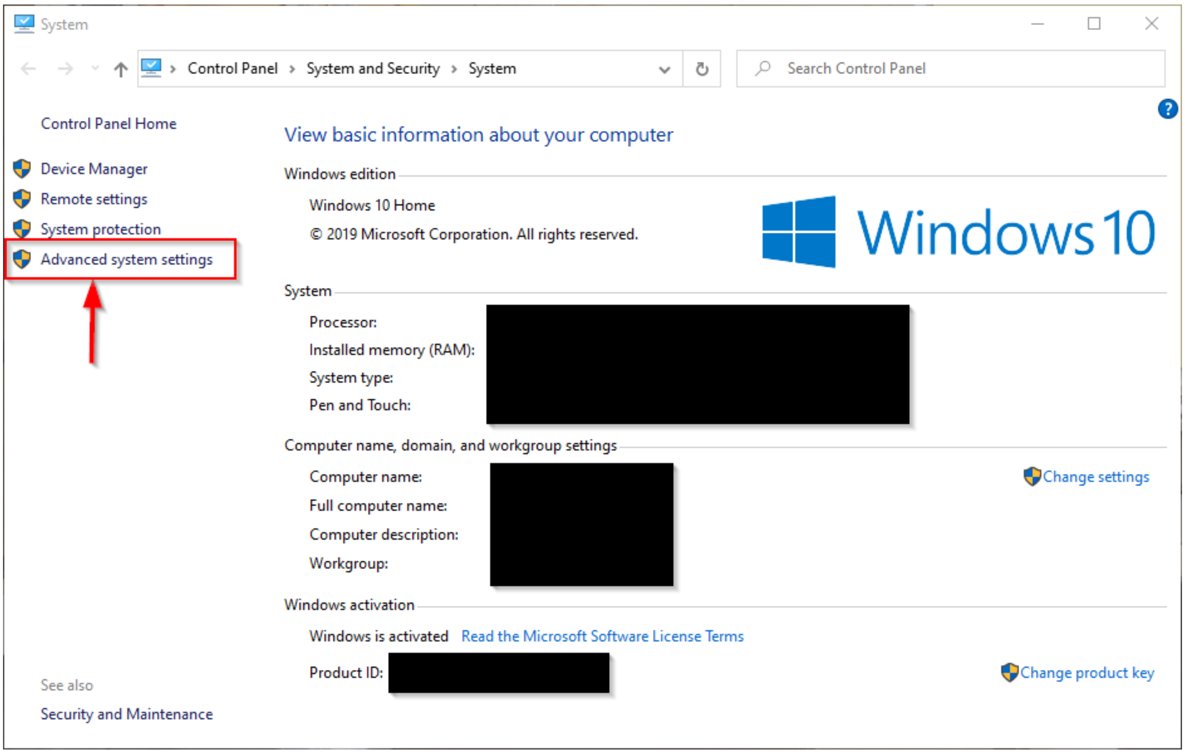
| Name | Last modified | Size | Description |
|---|---|---|---|
| Parent Directory | | - | |
| apache-hive-3.1.2-bin.tar.gz | 2019-08-26 20:20 | 266M | |
| apache-hive-3.1.2-bin.tar.gz.asc | 2019-08-26 20:20 | 833 | |
| apache-hive-3.1.2-bin.tar.gz.sha256 | 2019-08-26 20:20 | 95 | |
| apache-hive-3.1.2-src.tar.gz | 2019-08-26 20:20 | 24M | |
| apache-hive-3.1.2-src.tar.gz.asc | 2019-08-26 20:20 | 833 | |
| apache-hive-3.1.2-src.tar.gz.sha256 | 2019-08-26 20:20 | 95 | |

Figure 1 — apache-hive.3.1.2-bin.tar.gz file

When the file download is complete, we should extract twice *(as mentioned above)* the apache-hive.3.1.2-bin.tar.gz archive into "E:\hadoop-env\apache-hive-3.1.2" directory (Since we decided to use E:\hadoop-env\" as the installation directory for all technologies used in the previous guide.

3. Setting environment variables

After extracting Derby and Hive archives, we should go to Control Panel > System and Security > System. Then Click on "Advanced system settings".



View basic information about your computer

Windows edition

Windows 10 Home

© 2019 Microsoft Corporation. All rights reserved.

System

Processor: [redacted]
Installed memory (RAM):
System type:
Pen and Touch:

Computer name, domain, and workgroup settings

Computer name: [redacted]
Full computer name:
Computer description:
Workgroup:

Change settings

Windows activation

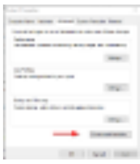
Windows is activated Read the Microsoft Software License Terms

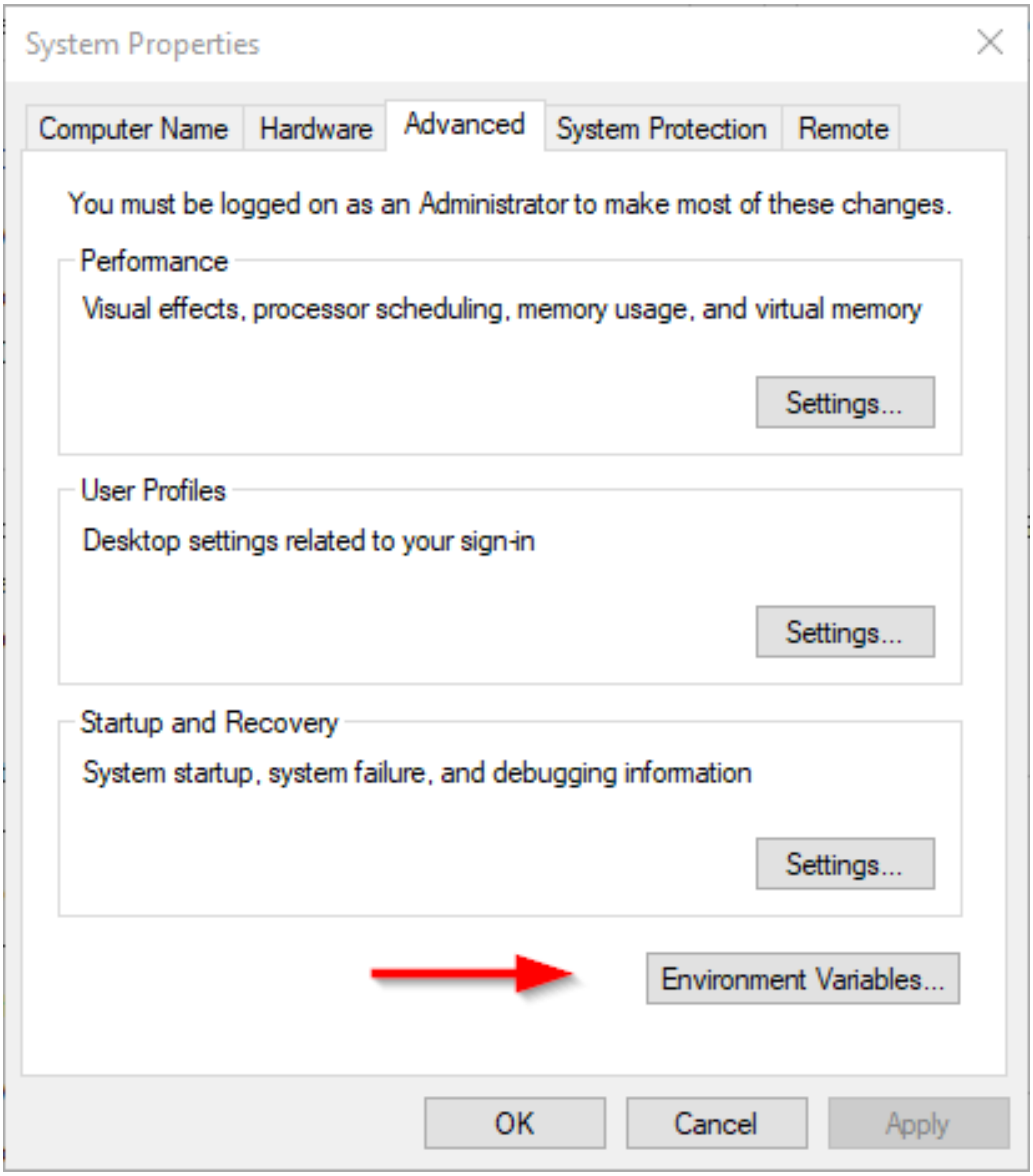
Product ID: [redacted]

Change product key

Figure 2 — Advanced system settings

In the advanced system settings dialog, click on “Environment variables” button.





You must be logged on as an Administrator to make most of these changes.

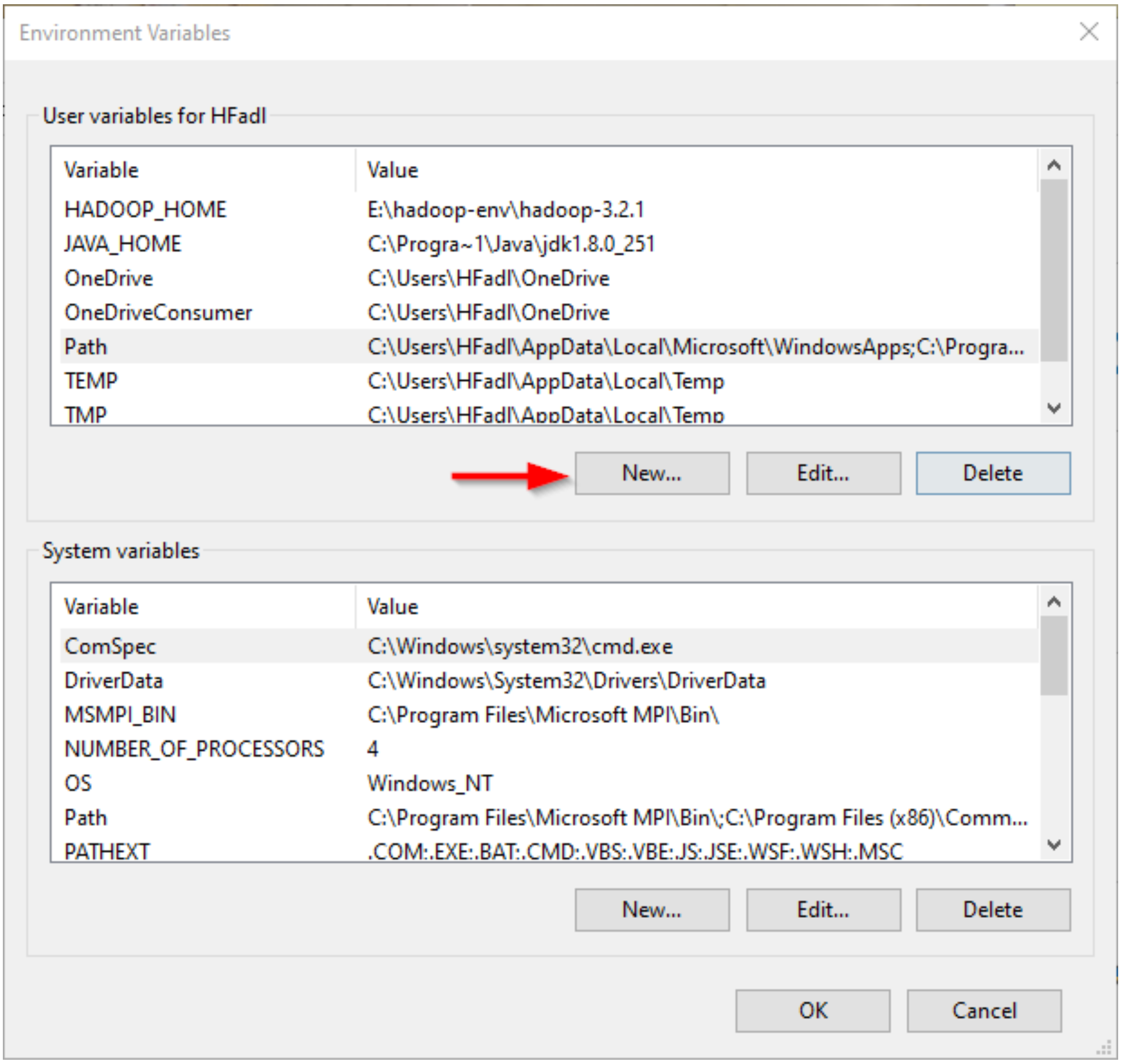
Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your sign-in

System startup, system failure, and debugging information

Figure 3 — Opening environment variables editor

Now we should add the following user variables:



User variables for HFadl

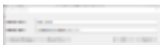
| Variable | Value |
|---|---|
| HADOOP_HOME | E:\hadoop-env\hadoop-3.2.1 |
| JAVA_HOME | C:\Progra~1\Java\jdk1.8.0_251 |
| OneDrive | C:\Users\HFadl\OneDrive |
| OneDriveConsumer | C:\Users\HFadl\OneDrive |
| Path | C:\Users\HFadl\AppData\Local\Microsoft\WindowsApps;C:\Progra... |
| TEMP | C:\Users\HFadl\AppData\Local\Temp |
| TMP | C:\Users\HFadl\AppData\Local\Temp |

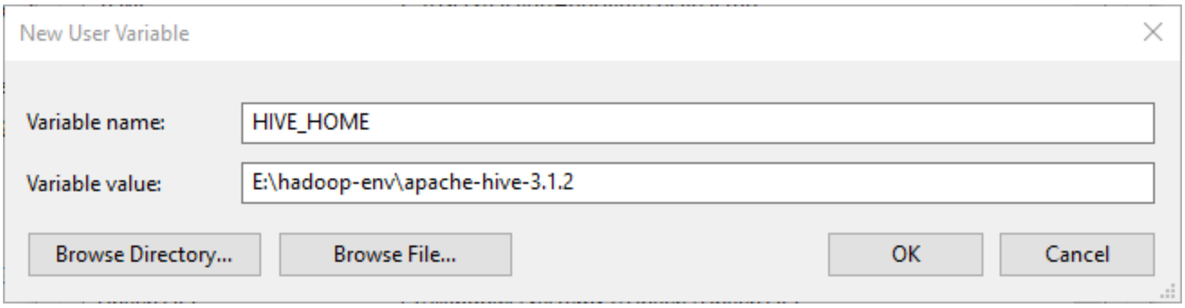
System variables

| Variable | Value |
|---|---|
| ComSpec | C:\Windows\system32\cmd.exe |
| DriverData | C:\Windows\System32\Drivers\DriverData |
| MSMPI_BIN | C:\Program Files\Microsoft MPI\Bin\ |
| NUMBER_OF_PROCESSORS | 4 |
| OS | Windows_NT |
| Path | C:\Program Files\Microsoft MPI\Bin\;C:\Program Files (x86)\Comm... |
| PATHEXT | .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC |

Figure 4 — Adding User variables

- HIVE_HOME: “E:\hadoop-env\apache-hive-3.1.2\”
- DERBY_HOME: “E:\hadoop-env\db-derby-10.14.2.0\”
- HIVE_LIB: “%HIVE_HOME%\lib”
- HIVE_BIN: “%HIVE_HOME%\bin”
- HADOOP_USER_CLASSPATH_FIRST: “true”





New User Variable

Variable name: HIVE_HOME

Variable value: E:\hadoop-env\apache-hive-3.1.2

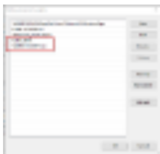
Figure 5 — Adding HIVE_HOME user variable

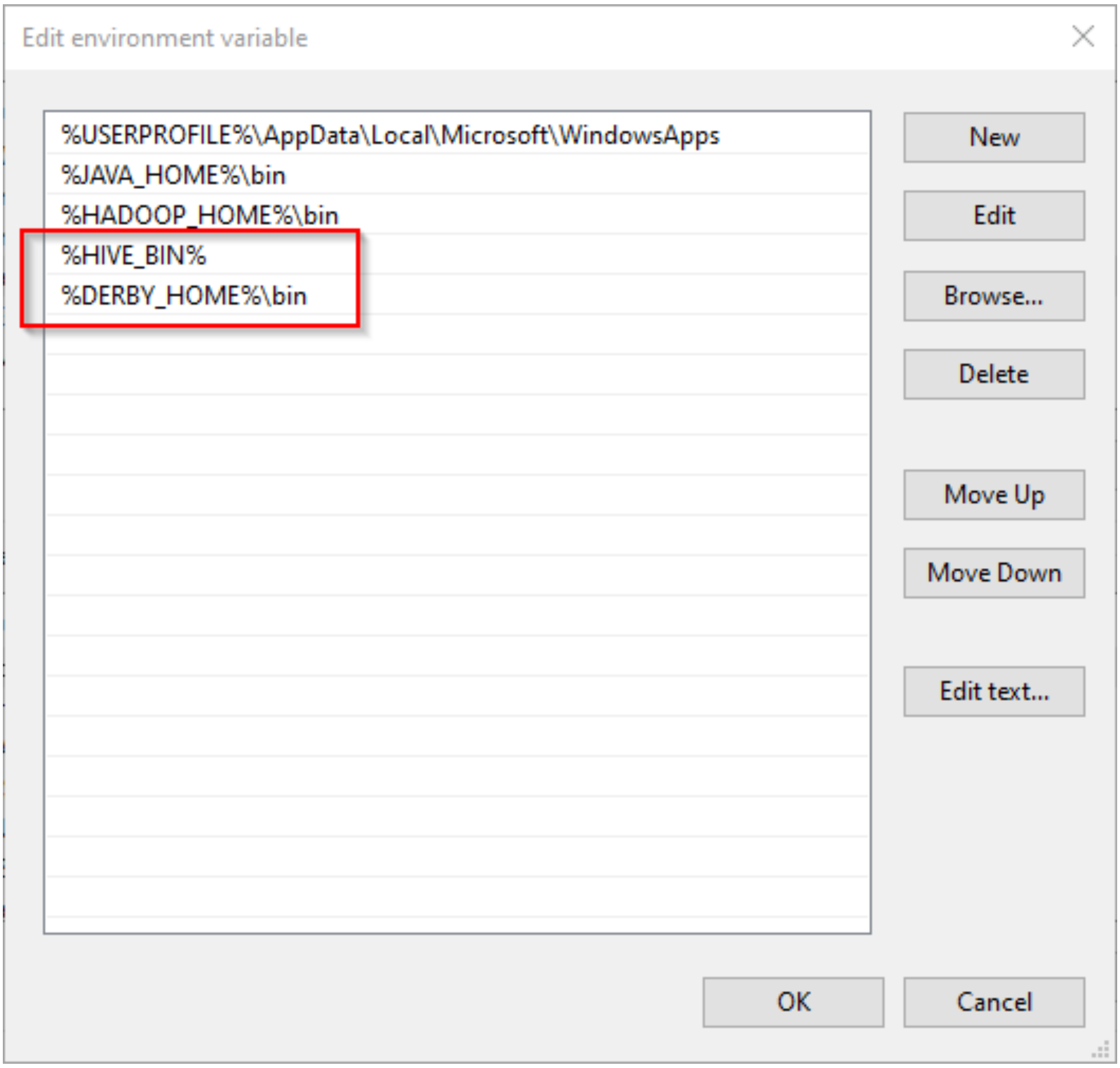
Besides, we should add the following system variable:

- HADOOP_USER_CLASSPATH_FIRST: “true”

Now, we should edit the Path user variable to add the following paths:

- %HIVE_BIN%
- %DERBY_HOME%\bin





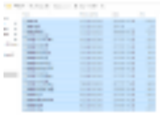
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
%JAVA_HOME%\bin
%HADOOP_HOME%\bin
%HIVE_BIN%
%DERBY_HOME%\bin

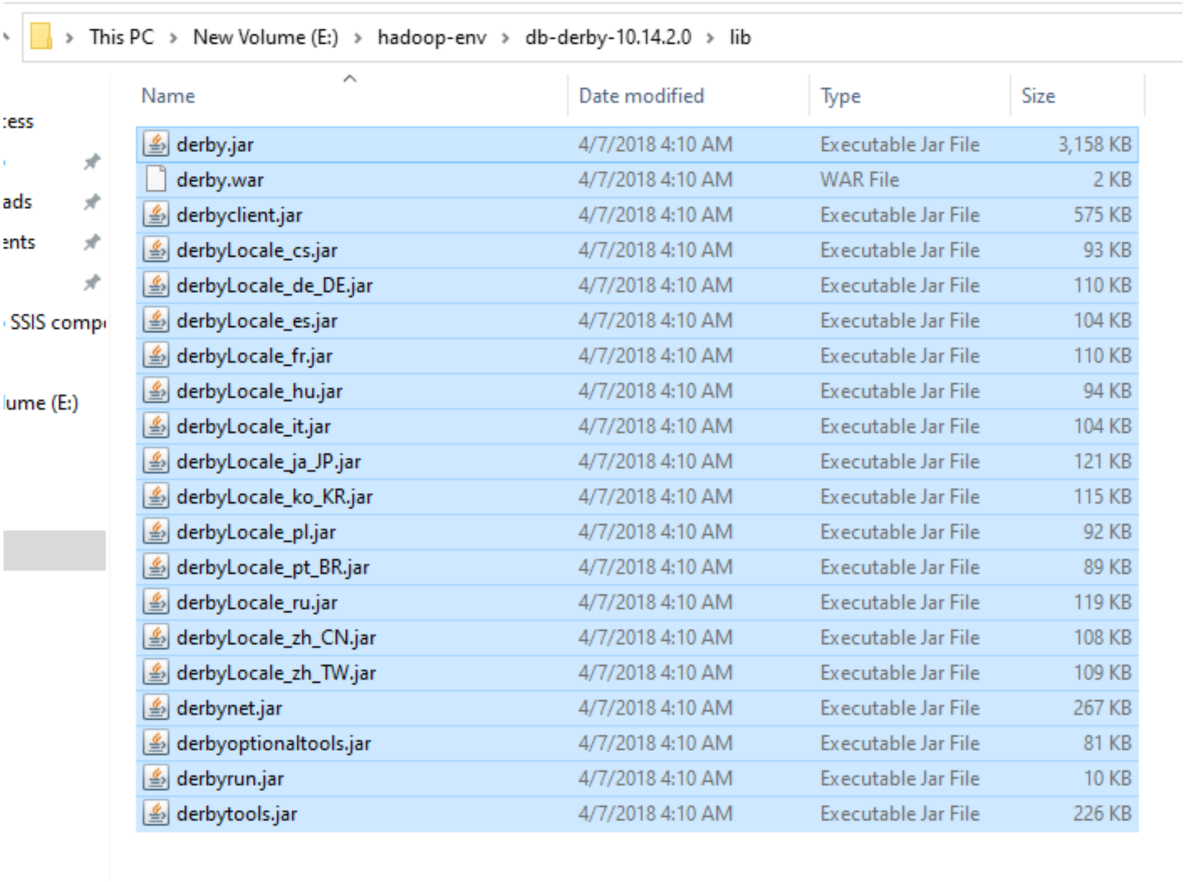
Figure 6 — Editing path environment variable

## 4. Configuring Hive

### 4.1. Copy Derby libraries

Now, we should go to the Derby libraries directory (E:\hadoop-env\db-derby-10.14.2.0\lib) and copy all *.jar files.



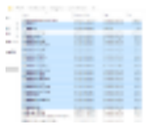


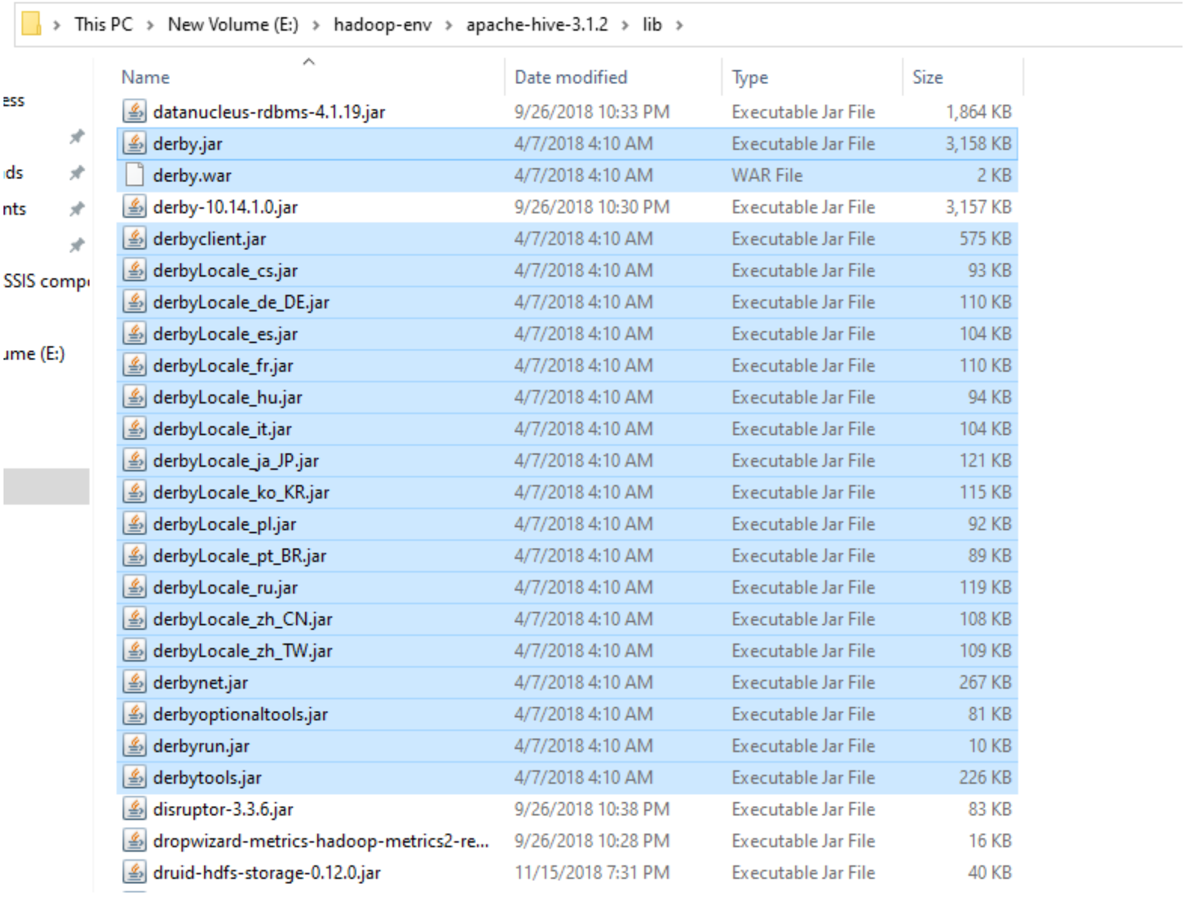
This PC › New Volume (E:) › hadoop-env › db-derby-10.14.2.0 › lib
:ess
ads
ents
SSIS comp
lume (E:)

| Name | Date modified | Type | Size |
|---|---|---|---|
| derby.jar | 4/7/2018 4:10 AM | Executable Jar File | 3,158 KB |
| derby.war | 4/7/2018 4:10 AM | WAR File | 2 KB |
| derbyclient.jar | 4/7/2018 4:10 AM | Executable Jar File | 575 KB |
| derbyLocale_cs.jar | 4/7/2018 4:10 AM | Executable Jar File | 93 KB |
| derbyLocale_de_DE.jar | 4/7/2018 4:10 AM | Executable Jar File | 110 KB |
| derbyLocale_es.jar | 4/7/2018 4:10 AM | Executable Jar File | 104 KB |
| derbyLocale_fr.jar | 4/7/2018 4:10 AM | Executable Jar File | 110 KB |
| derbyLocale_hu.jar | 4/7/2018 4:10 AM | Executable Jar File | 94 KB |
| derbyLocale_it.jar | 4/7/2018 4:10 AM | Executable Jar File | 104 KB |
| derbyLocale_ja_JP.jar | 4/7/2018 4:10 AM | Executable Jar File | 121 KB |
| derbyLocale_ko_KR.jar | 4/7/2018 4:10 AM | Executable Jar File | 115 KB |
| derbyLocale_pl.jar | 4/7/2018 4:10 AM | Executable Jar File | 92 KB |
| derbyLocale_pt_BR.jar | 4/7/2018 4:10 AM | Executable Jar File | 89 KB |
| derbyLocale_ru.jar | 4/7/2018 4:10 AM | Executable Jar File | 119 KB |
| derbyLocale_zh_CN.jar | 4/7/2018 4:10 AM | Executable Jar File | 108 KB |
| derbyLocale_zh_TW.jar | 4/7/2018 4:10 AM | Executable Jar File | 109 KB |
| derbynet.jar | 4/7/2018 4:10 AM | Executable Jar File | 267 KB |
| derbyoptionaltools.jar | 4/7/2018 4:10 AM | Executable Jar File | 81 KB |
| derbyrun.jar | 4/7/2018 4:10 AM | Executable Jar File | 10 KB |
| derbytools.jar | 4/7/2018 4:10 AM | Executable Jar File | 226 KB |

Figure 7 — Copy Derby libraries

Then, we should paste them within the Hive libraries directory (E:\hadoop-env\apache-hive-3.1.2\lib).





This PC › New Volume (E:) › hadoop-env › apache-hive-3.1.2 › lib ›

| Name | Date modified | Type | Size |
|---|---|---|---|
| datanucleus-rdbms-4.1.19.jar | 9/26/2018 10:33 PM | Executable Jar File | 1,864 KB |
| derby.jar | 4/7/2018 4:10 AM | Executable Jar File | 3,158 KB |
| derby.war | 4/7/2018 4:10 AM | WAR File | 2 KB |
| derby-10.14.1.0.jar | 9/26/2018 10:30 PM | Executable Jar File | 3,157 KB |
| derbyclient.jar | 4/7/2018 4:10 AM | Executable Jar File | 575 KB |
| derbyLocale_cs.jar | 4/7/2018 4:10 AM | Executable Jar File | 93 KB |
| derbyLocale_de_DE.jar | 4/7/2018 4:10 AM | Executable Jar File | 110 KB |
| derbyLocale_es.jar | 4/7/2018 4:10 AM | Executable Jar File | 104 KB |
| derbyLocale_fr.jar | 4/7/2018 4:10 AM | Executable Jar File | 110 KB |
| derbyLocale_hu.jar | 4/7/2018 4:10 AM | Executable Jar File | 94 KB |
| derbyLocale_it.jar | 4/7/2018 4:10 AM | Executable Jar File | 104 KB |
| derbyLocale_ja_JP.jar | 4/7/2018 4:10 AM | Executable Jar File | 121 KB |
| derbyLocale_ko_KR.jar | 4/7/2018 4:10 AM | Executable Jar File | 115 KB |
| derbyLocale_pl.jar | 4/7/2018 4:10 AM | Executable Jar File | 92 KB |
| derbyLocale_pt_BR.jar | 4/7/2018 4:10 AM | Executable Jar File | 89 KB |
| derbyLocale_ru.jar | 4/7/2018 4:10 AM | Executable Jar File | 119 KB |
| derbyLocale_zh_CN.jar | 4/7/2018 4:10 AM | Executable Jar File | 108 KB |
| derbyLocale_zh_TW.jar | 4/7/2018 4:10 AM | Executable Jar File | 109 KB |
| derbynet.jar | 4/7/2018 4:10 AM | Executable Jar File | 267 KB |
| derbyoptionaltools.jar | 4/7/2018 4:10 AM | Executable Jar File | 81 KB |
| derbyrun.jar | 4/7/2018 4:10 AM | Executable Jar File | 10 KB |
| derbytools.jar | 4/7/2018 4:10 AM | Executable Jar File | 226 KB |
| disruptor-3.3.6.jar | 9/26/2018 10:38 PM | Executable Jar File | 83 KB |
| dropwizard-metrics-hadoop-metrics2-re... | 9/26/2018 10:28 PM | Executable Jar File | 16 KB |
| druid-hdfs-storage-0.12.0.jar | 11/15/2018 7:31 PM | Executable Jar File | 40 KB |

Figure 8 — Paste Derby libraries within Hive libraries directory

## 4.2. Configuring hive-site.xml

Now, we should go to the Apache Hive configuration directory (E:\hadoop-env\apache-hive-3.1.2\conf) create a new file "hive-site.xml". We should paste the following XML code within this file:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
href="configuration.xsl"?>
<configuration><property>
<name>javax.jdo.option.ConnectionURL</
```

```
name>
<value>jdbc:derby://localhost:1527/
metastore_db;create=true</value>
<description>JDBC connect string for a
JDBC metastore</description>
</property><property>
<name>javax.jdo.option.ConnectionDriver
Name</name>
<value>org.apache.derby.jdbc.ClientDriver
</value>
<description>Driver class name for a JDBC
metastore</description>
</property>
<property>
<name>hive.server2.enable.doAs</name>
<description>Enable user impersonation
for HiveServer2</description>
<value>true</value>
</property>
<property>
<name>hive.server2.authentication</
name>
<value>NONE</value>
<description> Client authentication types.
NONE: no authentication check LDAP:
```

```
LDAP/AD based authentication KERBEROS:
Kerberos/GSSAPI authentication CUSTOM:
Custom authentication provider (Use with
property
hive.server2.custom.authentication.class)
</description>
</property>
<property>
<name>datanucleus.autoCreateTables</name>
<value>True</value>
</property>
</configuration>
```

## 5. Starting Services

### 5.1. Hadoop Services

To start Apache Hive, open the command prompt utility as administrator. Then, start the Hadoop services using start-dfs and start-yarn commands (as illustrated in the Hadoop installation guide).

### 5.2. Derby Network Server

Then, we should start the Derby network server on the localhost using the following command:

```
E:\hadoop-env\db-
```

```
derby-10.14.2.0\bin\StartNetworkServer -h 0.0.0.0
```

6. Starting Apache Hive

Now, let try to open a command prompt tool and go to the Hive binaries directory (E:\hadoop-env\apache-hive-3.1.2\bin) and execute the following command:

```
hive
```

We will receive the following error:

```
'hive' is not recognized as an internal or external command, operable program or batch file.
```

This error is thrown since the Hive 3.x version is not built for Windows (only in some Hive 2.x versions). To get things working, we should download the necessary *.cmd files from the following link: https://svn.apache.org/repos/asf/hive/trunk/bin/. Note that, you should keep the folder hierarchy (bin\ext\util).

You can download all *.cmd files from the following GitHub repository

- https://github.com/HadiFadl/Hive-cmd

Now if we try to execute the “hive”

command, we will receive the following error:

```
Exception in thread "main"
java.lang.NoSuchMethodError:
com.google.common.base.Preconditions.checkArgument(ZLjava/lang/String;Ljava/lang/Object;)V
at
org.apache.hadoop.conf.Configuration.set(Configuration.java:1357)
at
org.apache.hadoop.conf.Configuration.set(Configuration.java:1338)
at
org.apache.hadoop.mapred.JobConf.setJar(JobConf.java:518)
at
org.apache.hadoop.mapred.JobConf.setJarByClass(JobConf.java:536)
at
org.apache.hadoop.mapred.JobConf.<init>(JobConf.java:430)
at
org.apache.hadoop.hive.conf.HiveConf.initialize(HiveConf.java:5141)
```

```
at
org.apache.hadoop.hive.conf.HiveConf.<ini
t>(HiveConf.java:5104)
at
org.apache.hive.beeline.HiveSchemaTool.<
init>(HiveSchemaTool.java:96)
at
org.apache.hive.beeline.HiveSchemaTool.
main(HiveSchemaTool.java:1473)
at
sun.reflect.NativeMethodAccessorImpl.inv
oke0(Native Method)
at
sun.reflect.NativeMethodAccessorImpl.inv
oke(NativeMethodAccessorImpl.java:62)
at
sun.reflect.DelegatingMethodAccessorImpl
.invoke(DelegatingMethodAccessorImpl.ja
va:43)
at
java.lang.reflect.Method.invoke(Method.jav
a:498)
at
org.apache.hadoop.util.RunJar.run(RunJar.j
ava:318)
```

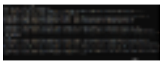
at org.apache.hadoop.util.RunJar.main(RunJar.java:232)

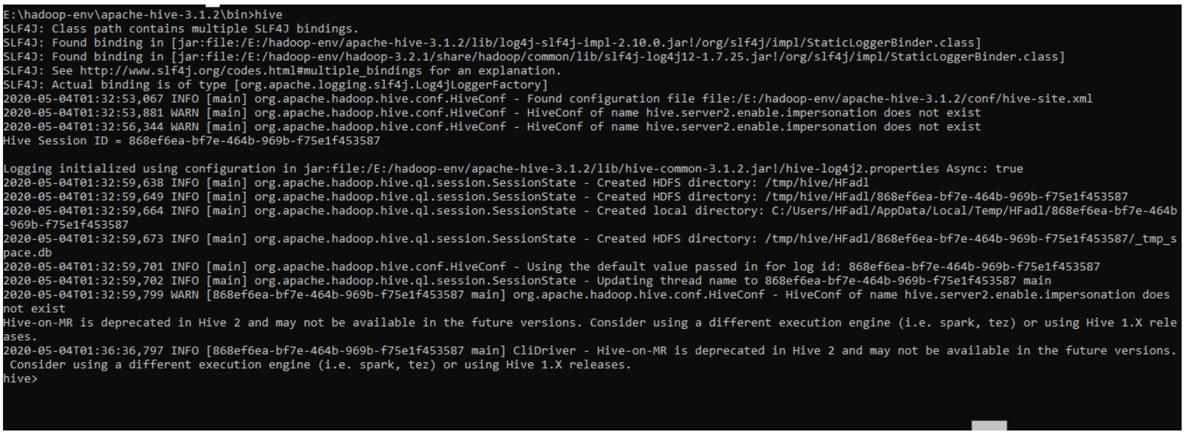
This error is thrown due to a Bug mentioned in the following Hive issue link: HIVE-22718.

As mentioned in the comments, this issue can be solved by replacing the guava-19.0.jar stored in “E:\hadoop-env\apache-hive-3.1.2\lib” with Hadoop’s guava-27.0-jre.jar found in “E:\hadoop-env\hadoop-3.2.1\share\hadoop\hdfs\lib”.

*Note: This file is also uploaded to the GitHub repository mentioned above.*

Now, if we run hive command again, then Apache Hive will start successfully.





```
E:\hadoop-env\apache-hive-3.1.2\bin>hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/E:/hadoop-env/apache-hive-3.1.2/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/E:/hadoop-env/hadoop-3.2.1/share/hadoop/common/lib/slf4j-log4j12-1.7.25.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]
2020-05-04T01:32:53,067 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Found configuration file file:/E:/hadoop-env/apache-hive-3.1.2/conf/hive-site.xml
2020-05-04T01:32:53,881 WARN [main] org.apache.hadoop.hive.conf.HiveConf - HiveConf of name hive.server2.enable.impersonation does not exist
2020-05-04T01:32:56,344 WARN [main] org.apache.hadoop.hive.conf.HiveConf - HiveConf of name hive.server2.enable.impersonation does not exist
Hive Session ID = 868ef6ea-bf7e-464b-969b-f75e1f453587

Logging initialized using configuration in jar:file:/E:/hadoop-env/apache-hive-3.1.2/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async: true
2020-05-04T01:32:59,638 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Created HDFS directory: /tmp/hive/HFadl
2020-05-04T01:32:59,649 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Created HDFS directory: /tmp/hive/HFadl/868ef6ea-bf7e-464b-969b-f75e1f453587
2020-05-04T01:32:59,664 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Created local directory: C:/Users/HFadl/AppData/Local/Temp/HFadl/868ef6ea-bf7e-464b
-969b-f75e1f453587
2020-05-04T01:32:59,673 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Created HDFS directory: /tmp/hive/HFadl/868ef6ea-bf7e-464b-969b-f75e1f453587/_tmp_s
pace.db
2020-05-04T01:32:59,701 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: 868ef6ea-bf7e-464b-969b-f75e1f453587
2020-05-04T01:32:59,702 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to 868ef6ea-bf7e-464b-969b-f75e1f453587 main
2020-05-04T01:32:59,799 WARN [868ef6ea-bf7e-464b-969b-f75e1f453587 main] org.apache.hadoop.hive.conf.HiveConf - HiveConf of name hive.server2.enable.impersonation does
not exist
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 rele
ases.
2020-05-04T01:36:36,797 INFO [868ef6ea-bf7e-464b-969b-f75e1f453587 main] CliDriver - 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>
```

Figure 9 — Starting Apache Hive

## 7. Initializing Hive

After ensuring that the Apache Hive started successfully. We may not be able to run any HiveQL command. This is because the Metastore is not initialized yet. Besides HiveServer2 service must be running.

To initialize Metastore, we need to use schematool utility which is not compatible with windows. To solve this problem, we will use Cygwin utility which allows executing Linux command from windows.

### 7.1. Creating symbolic links

First, we need to create the following directories:

- E:\cygdrive
- C:\cygdrive

Now, open the command prompt as administrator and execute the following commands:

```
mklink /J  E:\cygdrive\e\ E:\
mklink /J  C:\cygdrive\c\ C:\
```

These symbolic links are needed to work with Cygwin utility properly since Java may cause some problems.

### 7.2. Initializing Hive Metastore

Open Cygwin utility and execute the

following commands to define the environment variables:

```
export HADOOP_HOME='/cygdrive/e/hadoop-env/hadoop-3.2.1'
export PATH=$PATH:$HADOOP_HOME/bin
export HIVE_HOME='/cygdrive/e/hadoop-env/apache-hive-3.1.2'
export PATH=$PATH:$HIVE_HOME/bin
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$HIVE_HOME/lib/*.jar
```

We can add these lines to the “~/.bashrc” file then you don’t need to write them each time you open Cygwin.

Now, we should use the schematool utility to initialize the Metastore:

```
$HIVE_HOME/bin/schematool -dbType derby -initSchema
```

### 7.3. Starting HiveServer2 service

Now, open a command prompt and run the following command:

```
hive --service hiveserver2 start
```

We should leave this command prompt open, and open a new one where we should start Apache Hive using the following command:

```
hive
```

## 7.4. Starting WebHCat Service (Optional)

In the project we are working on, we need to execute HiveQL statement from SQL Server Integration Services which can access Hive from the WebHCat server.

To start the WebHCat server, we should open the Cygwin utility and execute the following command:

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