Shalaj Shukla

This document described how to copy the data between two hive cluster using hive IMPORT and Export commands

Copy Data between hive Cluster

Contents

[Copy Data from one hive instance to other 2](#_Toc467152118)

[Export Table 2](#_Toc467152119)

[Distcp to copy data 2](#_Toc467152120)

[Import Table 3](#_Toc467152121)

[Automate Script to move all Tables 6](#_Toc467152122)

[Password less Connection 6](#_Toc467152123)

[Shell Script to move tables 9](#_Toc467152124)

[Issue log 12](#_Toc467152125)

# Copy Data from one hive instance to other

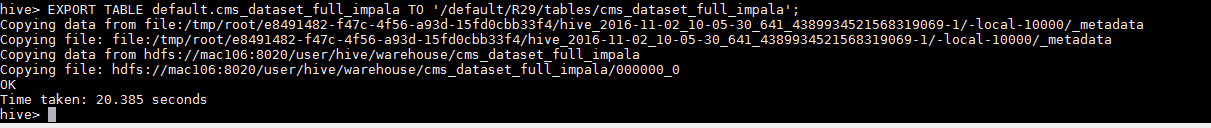
Since version 0.8, Hive supports EXPORT and IMPORT features that allows you to export the metadata as well as the data for the corresponding table to a directory in HDFS, which can then be imported back to another database or Hive instance

Here we are trying to copying data from 172.27.155.106 to 172.27.155.78

## Export Table

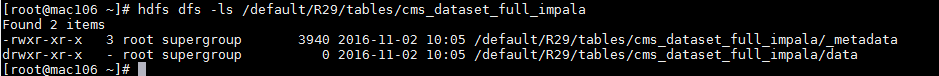
First enter into hive prompt in 172.27.155.106 and issue following command

|  |
| --- |
| EXPORT TABLE default.cms\_dataset\_full\_impala TO '/default/R29/tables/cms\_dataset\_full\_impala'; |



This command will load data as well as metadata into specified directory

Check whether folder structure created on hdfs



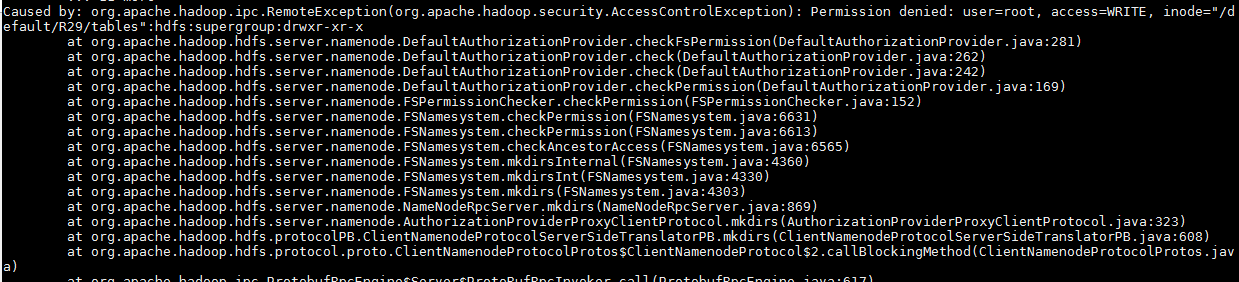
To export partitioned table you need to specify partition values

|  |
| --- |
| EXPORT TABLE default.dar\_site\_wise partition (sampled="unsampled", dt="2016-06-30",path="pagepath",exclude\_paid\_traffic="1",us\_desk\_tablet="1",site="US") TO '/default/R29/tables/dar\_site\_wise'; |

## Distcp to copy data

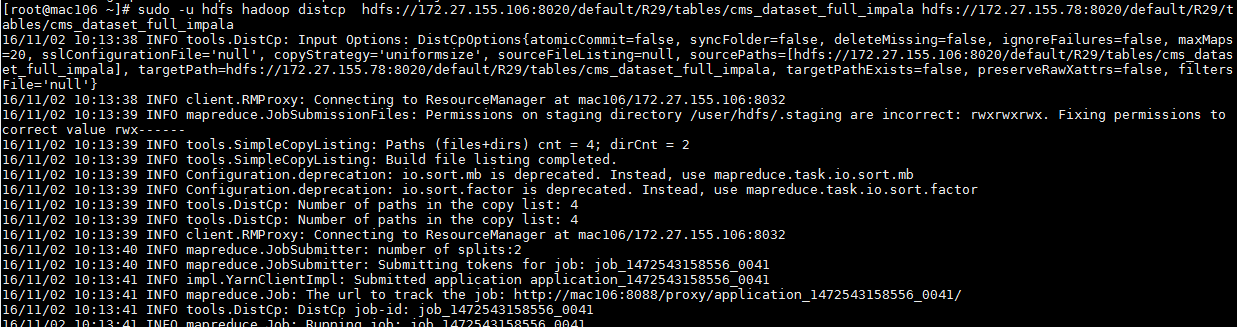
The next step is to copy the data from 172.27.155.106 to to 172.27.155.78 you can use the “distcp” command from Hadoop:

Here we are executing this command with user hdfs as we are getting below permission issue while executing it with root user

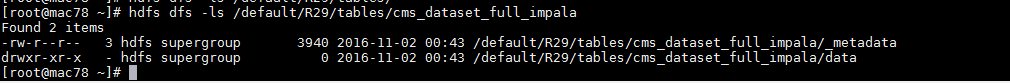


|  |
| --- |
| sudo -u hdfs hadoop distcp hdfs://172.27.155.106:8020/default/R29/tables/cms\_dataset\_full\_impala hdfs://172.27.155.78:8020/default/R29/tables/cms\_dataset\_full\_impala |

*Note: you can run command without sudo if there is no permission issue*



Now check in 172.27.155.78, if files are successfully copied



## Import Table

Now enter into hive prompt in 172.27.155.78 and issue import command

|  |
| --- |
| IMPORT EXTERNAL TABLE R29.cms\_dataset\_full\_impala FROM '/default/R29/tables/cms\_dataset\_full\_impala'; |

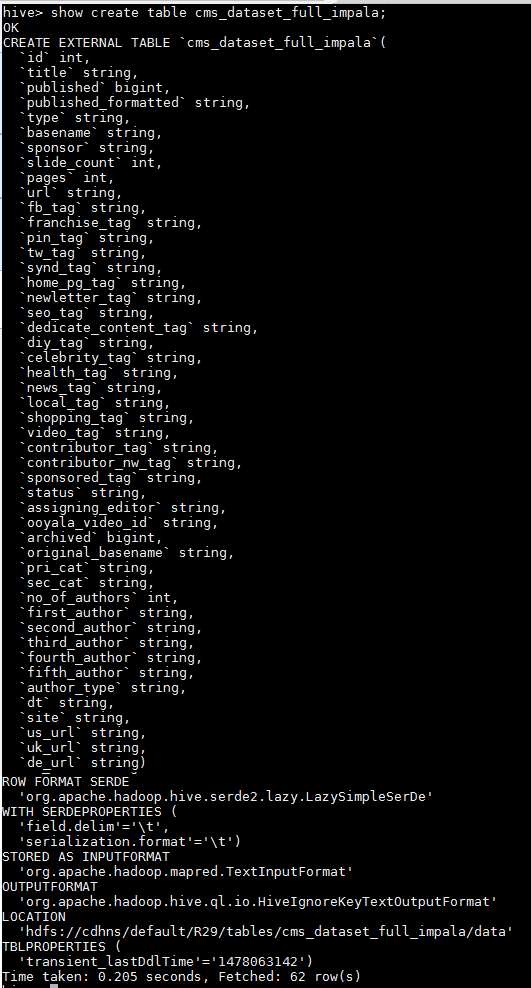


Since it is an External Table we are using External keyword else you need not to use it.

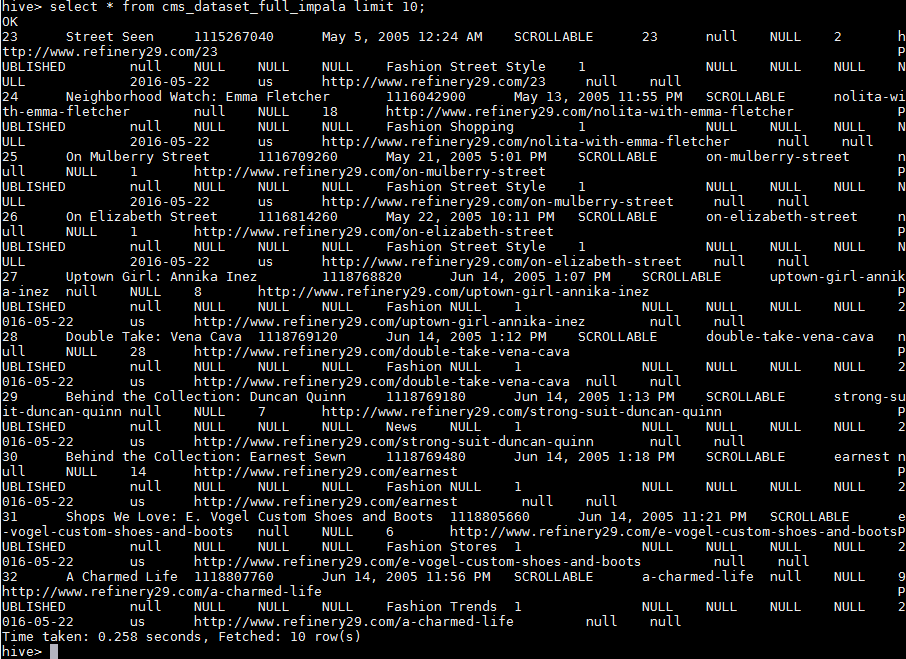
*Note: please drop table if already exist*

You can check the create table script using below command

|  |
| --- |
| show create table cms\_dataset\_full\_impala; |



Now check Data



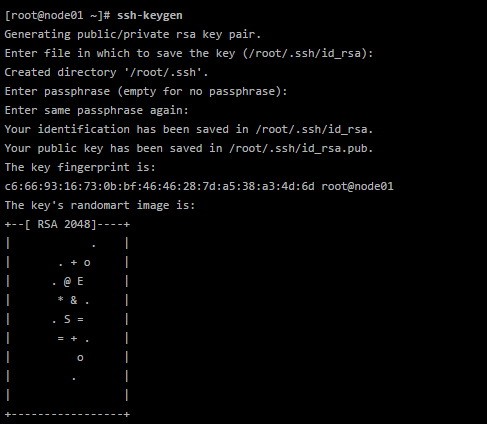
## Automate Script to move all Tables

First you need to setup the password less connection between source (client) machine and destination (remote) machine.

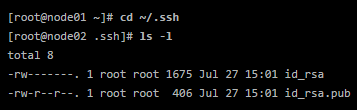
### Password less Connection

On the client machine (ie. the one you are SSH'ing from) you will need to create an SSH RSA key. So run the following command - ensure you don't supply a password:

|  |
| --- |
| ssh-keygen |

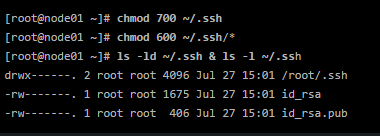


It will create .ssh directory in root and also create private and public key file namely id\_rsa and id\_rsa.pub respectively



Now change the permission by executing below command

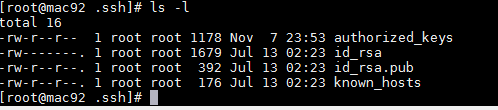
|  |
| --- |
| [root@node01 ~]# **chmod 700 ~/.ssh**  [root@node01 ~]# **chmod 600 ~/.ssh/\*** |



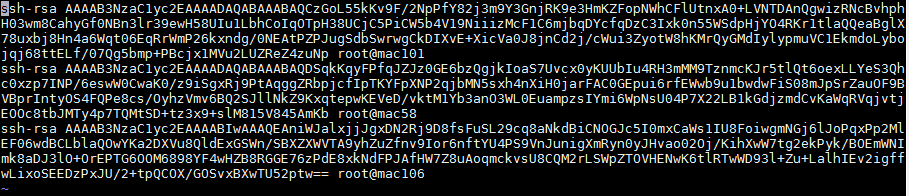
Now copy the public key (id\_rsa.pub file content) to the machine you want to SSH

Go to remote machine

Check if .ssh folder exist at root or not if not just create it and also give the permission same like client machine, also create authorized\_keys file and give the permission 600 to this file



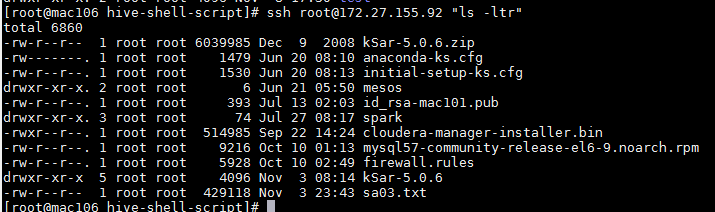
Now open the authorized\_keys file and append the content of **id\_rsa.pub** file that we copied



Now from client machine try to execute below command

In this case mac106 is client and mac92 is remote machine

|  |
| --- |
| ssh root@172.27.155.92 "ls -ltr" |



If it runs without prompting for mac92 password it means password less connection has been made successfully and you can execute any command from client machine.

### Shell Script to move tables

Before running the shell script please create a user in hdfs by which you are going to run this script in both source and destination machine using below command

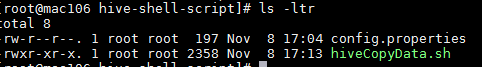
|  |
| --- |
| [root@mac78 ~]# sudo -u hdfs hadoop fs -mkdir /user/root  [root@mac78 ~]# sudo -u hdfs hadoop fs -chown root:root /user/root |

Note: here I am using root user to running the shell script

If root doesn’t have permission you will get below error

|  |
| --- |
| Caused by: org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.security.AccessControlException): Permission denied: user=root, access=WRITE, inode="/user":hdfs:supergroup:drwxr-xr-x  at |

Basically there are two files one is configuration file which you need to modify and the other is shell script that will copy all the data from source schema to destination schema



First we need to update config.properties file

Here we have defined some properties for Source and destination schema you need to change these properties accordingly

|  |
| --- |
| # source schema that we want to copy  SOURCE\_SCHEMA=test  # destination schema  DEST\_SCHEMA=testdb  # destination user name that we use in ssh command  DEST\_USER\_NAME=root  # destination machine IP  DEST\_IP=172.27.155.92  # source hdfs path where we want to export data  SOURCE\_HDFS\_PATH=hdfs://172.27.155.106:8020/test.db/tables  # destination hdfs path where we want to copy the data  DEST\_HDFS\_PATH=hdfs://172.27.155.92:8020/test.db/tables |

After setting up the properties you need to make sure to give executable permission to hiveCopyData.sh file

|  |
| --- |
| chmod +x hiveCopyData.sh |

Before running the shell script please drop tables from destination schema if already exist

To drop all the tables of schema you use below command

|  |
| --- |
| hive -e 'use testdb;show tables' | xargs -I '{}' hive -e 'use testdb;drop table {}' |

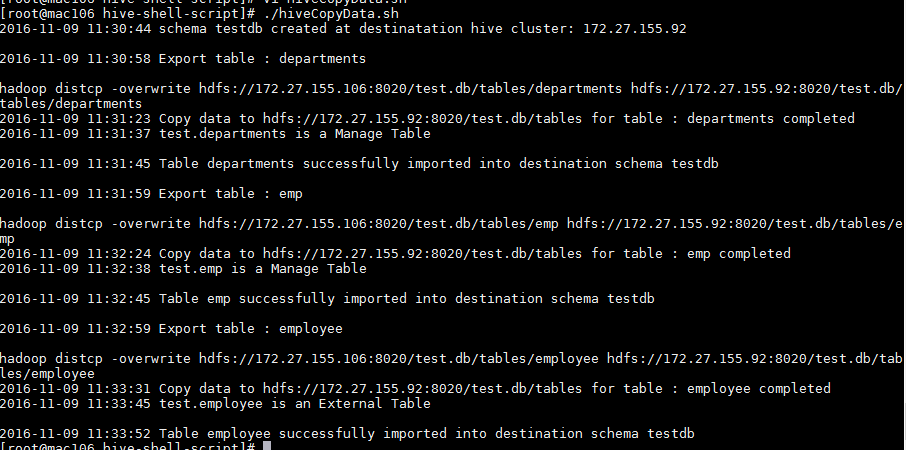
Here testdb is name of schema, you can change it accordingly

Or you can drop the whole database also using below command

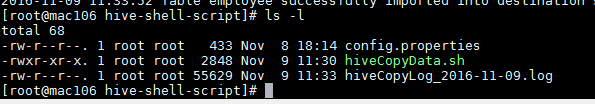
|  |
| --- |
| DROP DATABASE IF EXISTS testdb CASCADE; |

Now run the shell script

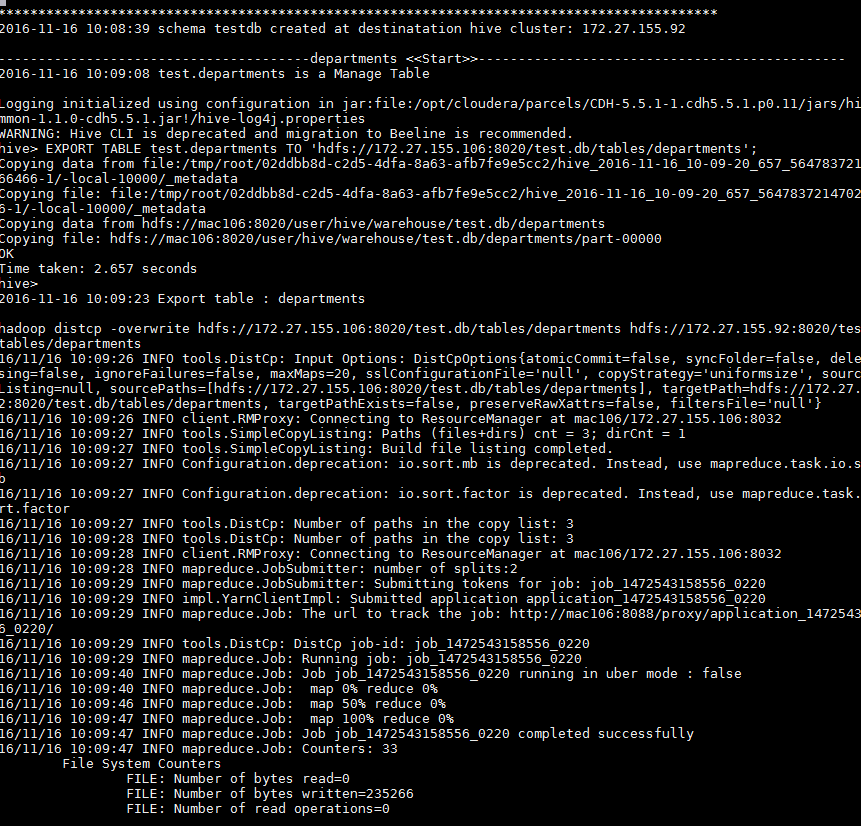
|  |
| --- |
| ./hiveCopyData.sh |



It will create a log file **hiveCopyLog\_<data>.log** in the same directory



You can check the detailed log here



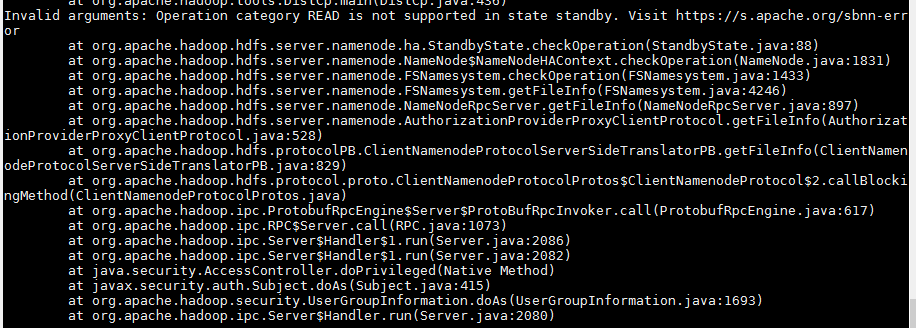
Required files are attached herewith



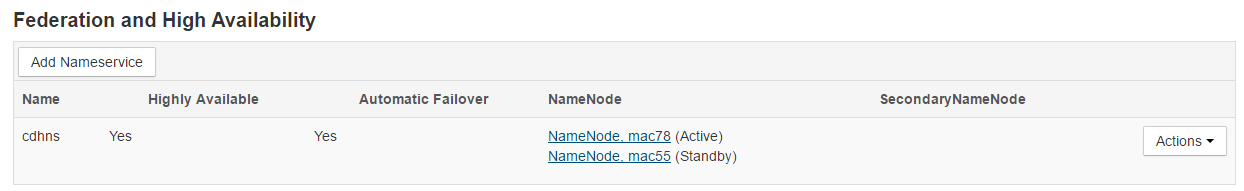
## Issue log

1. While issuing distcp command we are getting below issue

|  |
| --- |
| sudo -u hdfs hadoop distcp -overwrite hdfs://172.27.155.106:8020/default/R29/tables/cms\_dataset\_full\_impala hdfs://172.27.155.55:8020/default/R29/tables/cms\_dataset\_full\_impala |



This issue occurs because we are trying to copying data over standby node, this issue can occur if you configured high availability, so check the Active and Standby Node in Cloudera Manager UI and change the IP accordingly



2. During Import you can get following issue



This issue occurs because we are importing External table without using External command

So change command accordingly

|  |
| --- |
| IMPORT EXTERNAL TABLE R29.cms\_dataset\_full\_impala FROM '/default/R29/tables/cms\_dataset\_full\_impala'; |

3. During executing ***select count(\*) from ga\_dm\_impala*** getting below issue

|  |
| --- |
| Caused by: org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.security.AccessControlException): Permission denied: user=root, access=WRITE, inode="/user":hdfs:supergroup:drwxr-xr-x  at org.apache.hadoop.hdfs.server.namenode.DefaultAuthorizationProvider.checkFsPermission(DefaultAuthorizationProvider.java:281)  at org.apache.hadoop.hdfs.server.namenode.DefaultAuthorizationProvider.check(DefaultAuthorizationProvider.java:262)  at org.apache.hadoop.hdfs.server.namenode.DefaultAuthorizationProvider.check(DefaultAuthorizationProvider.java:242)  at org.apache.hadoop.hdfs.server.namenode.DefaultAuthorizationProvider.checkPermission(DefaultAuthorizationProvider.java:169)  at org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkPermission(FSPermissionChecker.java:152)  at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkPermission(FSNamesystem.java:6631)  at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkPermission(FSNamesystem.java:6613)  at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkAncestorAccess(FSNamesystem.java:6565)  at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.mkdirsInternal(FSNamesystem.java:4360)  at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.mkdirsInt(FSNamesystem.java:4330)  at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.mkdirs(FSNamesystem.java:4303)  at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.mkdirs(NameNodeRpcServer.java:869)  at org.apache.hadoop.hdfs.server.namenode.AuthorizationProviderProxyClientProtocol.mkdirs(AuthorizationProviderProxyClientProtocol.java:323) |

After debugging into the details, found out the to read the count() the user needs to read the blocks on the HDFS. Since the user didn’t have needed privileges, we are getting the error. The issue was resolved using below

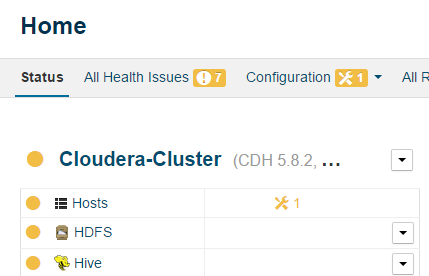
|  |
| --- |
| [root@mac78 ~]# sudo -u hdfs hadoop fs -mkdir /user/root  [root@mac78 ~]# sudo -u hdfs hadoop fs -chown root:root /user/root |

OR

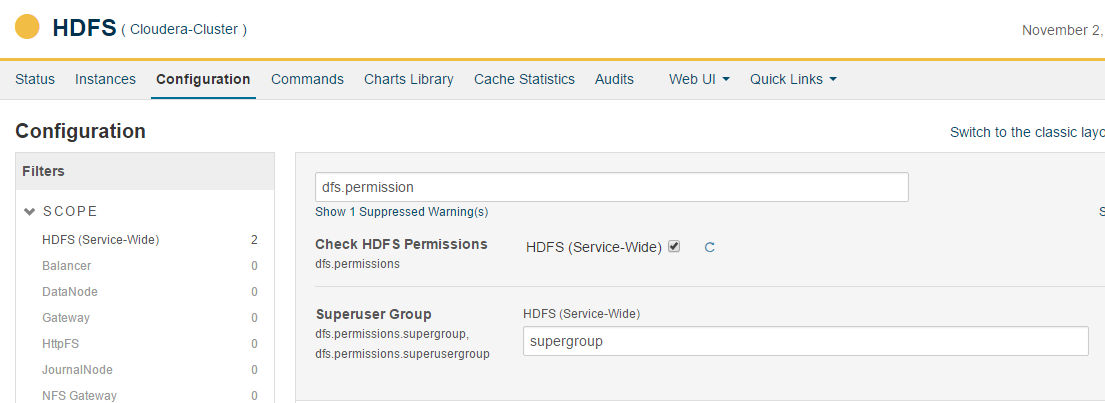
There is one more way to solve this issue, you can update/add below property in /etc/Hadoop/conf/hdfs-site.xml file

|  |
| --- |
| <property> <name>dfs.permissions</name> <value>false</value> </property> |

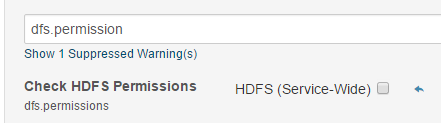
If you are using Cloudera-manager click on HDFS on home page



Click on Configuration



Search **dfs.permission** and deselect check box



Click on **Save Changes** and redeploy it