# **Installation of HDP clients on Greenplum Database system**

Background:

To copy data from Greenplum to Hadoop cluster we need gphdfs protocal to be configured correctly. The gphdfs protocol will make use of the Greenplum parallelism to transfer the data from Greenplum system to Hadoop filesystem using external table.

Prerequisites:

Request the Hadoop team to copy the HDP client rpms to cerl400001 to /u/users/greenplum/ directory.

Implementation steps:

Important Note:

All Implementation steps can be done without restarting the database.

Once all steps are done, stop batch jobs on GPPROD2 by calling\email control room.

This is required to set new JAVA\_HOME in $GPHOME/greenplum\_path.sh

1. Copy HDP rpms to Greenplum system from cerl400001.

scp $USER@cerl400001:/u/users/greenplum/hadoop-client-rpms.zip /root

2. Push rpms to all segment hosts of Greenplum System as root -

gpscp -f ~gpadmin/gpconfigs/hostfile\_segments /root/hadoop-client-rpms.zip =:/root/

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments "unzip /root/hadoop-client-rpms.zip"

3. Install rpms in following order as root -

🡪Make sure krb5-workstation.x86\_64 is installed (which kinit and which klist).

[gpadmin@mdw ~]$ which kinit

/usr/bin/kinit

[gpadmin@mdw ~]$ which klist

/usr/bin/klist

🡪Uninstall if there are any older version of hadoop clients installed

rpm -ev hadoop\*

🡪From gpprod2 (or gpprod2-mdw) run below command as root -

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "cd /root/hadoop-client-rpms/;

rpm -ivh hdp-select-2.4.2.0-258.el6.noarch.rpm;

rpm -ivh bigtop-jsvc-1.0.10.2.4.2.0-258.el6.x86\_64.rpm;

rpm -ivh ranger\_2\_4\_2\_0\_258-hdfs-plugin-0.5.0.2.4.2.0-258.el6.x86\_64.rpm;

rpm -ivh ranger\_2\_4\_2\_0\_258-yarn-plugin-0.5.0.2.4.2.0-258.el6.x86\_64.rpm;

rpm -ivh hadoop\_2\_4\_2\_0\_258-2.7.1.2.4.2.0-258.el6.x86\_64.rpm;

rpm -ivh hadoop\_2\_4\_2\_0\_258-hdfs-2.7.1.2.4.2.0-258.el6.x86\_64.rpm;

rpm -ivh hadoop\_2\_4\_2\_0\_258-mapreduce-2.7.1.2.4.2.0-258.el6.x86\_64.rpm;

rpm -ivh hadoop\_2\_4\_2\_0\_258-yarn-2.7.1.2.4.2.0-258.el6.x86\_64.rpm;

rpm -ivh zookeeper\_2\_4\_2\_0\_258-3.4.6.2.4.2.0-258.el6.noarch.rpm;

rpm -ivh spark\_2\_4\_2\_0\_258-yarn-shuffle-1.6.1.2.4.2.0-258.el6.noarch.rpm;

rpm -ivh hadoop\_2\_4\_2\_0\_258-client-2.7.1.2.4.2.0-258.el6.x86\_64.rpm;"

4. On Hadoop Cluster as root copy following configuration files

krb5.conf (/usr/java/default/jre/lib/security/krb5.conf),

hadoop conf (/etc/hadoop/conf),

.gpadmin.keytab : (u/users/gpadmin/.gpadmin.keytab) to Greenplum DB system from Hadoop system.

Note : Request Hadoop admin to copy the file to gpadmin server

🡪 logon to Hadoop Cluster and run below commands

scp /usr/java/default/jre/lib/security/krb5.conf gpadmin@gpprod2:/usr/java/default/jre/lib/security/

scp -r /etc/hadoop/conf/\* gpadmin@gpprod2:/etc/hadoop/conf/

scp u/users/gpadmin/.gpadmin.keytab gpadmin@gpprod2:/home/gpadmin

5. As root copy krb5.conf to all segments using below command

gpscp -f ~gpadmin/gpconfigs/hostfile\_segments -v /usr/java/default/jre/lib/security/krb5.conf =:/usr/java/default/jre/lib/security/

6. As root take a backup of bashrc file and remove any HADOOP related environmental variables from gpadmin's .bashrc file

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "cp -p /home/gpadmin/.bashrc /home/gpadmin/.bashrc.20170201"

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "sed -i \"s|export HADOOP|#export HADOOP|g\" /home/gpadmin/.bashrc"

7. As gpadmin create a .profile as below in Greenplum Database system-

export KRB5\_CONFIG=/usr/java/default/jre/lib/security/krb5.conf

export CLUSTER=`cat /usr/java/default/jre/lib/security/krb5.conf | grep default\_realm | awk '{print $3}'`

export JAVA\_HOME=/usr/java/default/jre/

kinit -kt .$USER.keytab $USER@${CLUSTER}

🡪 Copy keytab and .profile files to all segments and change permissions

gpscp -f ~gpadmin/gpconfigs/hostfile\_segments -v /home/gpadmin/.profile =:~gpadmin/

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "chown gpadmin:gpadmin /home/gpadmin/.\*profile"

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "chmod 700 /home/gpadmin/.\*profile"

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "chown gpadmin:gpadmin /home/gpadmin/.gpadmin.keytab"

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "chmod 400 /home/gpadmin/.gpadmin.keytab"

8. As gpadmin take a backup of .bash\_profile, make below changes -

🡪set below properties in .bash\_profile file of gpadmin

export JAVA\_HOME=/usr/java/default

./.profile

🡪copy bash\_profile to all segments and change permissions.

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "cp -p /home/gpadmin/.bash\_profile /home/gpadmin/.bash\_profile.20170201"

gpscp -f ~gpadmin/gpconfigs/hostfile\_segments -v /home/gpadmin/.bash\_profile =:~gpadmin/

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "chmod 700 /home/gpadmin/.bash\_profile"

9. As root make below changes in hadoop conf files - Add below properties to hdfs-site.xml

<property>

<name>com.emc.greenplum.gpdb.hdfsconnector.security.user.keytab.file</name>

<value>/home/gpadmin/.gpadmin.keytab</value>

</property>

<property>

<name>com.emc.greenplum.gpdb.hdfsconnector.security.user.name</name>

<value>gpadmin@HADOOP\_PAS11.WAL-MART.COM</value>

</property>

Add KRB5\_CONFIG, GPHDFS\_JARS env variables in hadoop-env.sh and also modify HADOOP\_CLASSPATH and CLASSPATH

export KRB5\_CONFIG=/usr/java/default/jre/lib/security/krb5.conf

export GPHDFS\_JARS=/usr/hdp/2.4.2.0-258/\*/\*.jar

export HADOOP\_CLASSPATH=${HADOOP\_CLASSPATH}:${JAVA\_JDBC\_LIBS}:${GPHDFS\_JARS}

export CLASSPATH=${HADOOP\_CLASSPATH}

10. As root, make a directory called "/etc/hadoop/conf" on all segments if not already exists

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "mkdir /etc/hadoop/conf"

11. As root, copy all hadoop conf file from mdw to all segment hosts

gpscp -r -f ~gpadmin/gpconfigs/hostfile\_segments -v /etc/hadoop/conf/\* =:/etc/hadoop/conf/

12. As root, create .profile with kinit commands and copy keytab. .profile files to all segments and change permissions

gpscp -f ~gpadmin/gpconfigs/hostfile\_segments -v /home/gpadmin/.profile =:~gpadmin/

gpscp -f ~gpadmin/gpconfigs/hostfile\_segments -v /home/gpadmin/.bashrc =:~gpadmin/

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "chown gpadmin:gpadmin /home/gpadmin/.\*profile"

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "chmod 700 /home/gpadmin/.\*profile"

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "chown gpadmin:gpadmin /home/gpadmin/.gpadmin.keytab"

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "chmod 400 /home/gpadmin/.gpadmin.keytab"

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "chown gpadmin:gpadmin /home/gpadmin/.bashrc"

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "chmod 700 /home/gpadmin/.bashrc"

13. As root, create /etc/security/keytabs directory on gpprod2-mdw and all other segment hosts.

mkdir -p /etc/security/keytabs

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "mkdir -p /etc/security/keytabs"

🡪From PAS11 Ambari (Hadoop Cluster ) as root, Copy /etc/security/keytabs/hdfs.headless.keytab to /etc/security/ directory of Greenplum system

🡪 Create a soft link as below

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "ln -s /etc/security/keytabs/hdfs.headless.keytab /etc/security/keytabs/hdfs.service.keytab"

🡪Validate if both hdfs.headless.keytab and hdfs.service.keytab are there under /etc/security directory

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v "ls -l /etc/security/keytabs/"

14. In Greenplum system as gpadmin greenplum gphdfs properties as below

gpconfig -c gp\_hadoop\_home -v "'/usr/hdp/2.4.2.0-258'"

gpconfig -c gp\_hadoop\_target\_version -v "'hdp2'"

🡪Reload the Greenplum Configuration file as below

gpstop -u

15. Add new JAVA home in $GPHOME/greenplum\_path.sh and push to all segment hosts

#JAVA\_HOME=$GPHOME/ext/jre-1.6.0\_32/jre1.6.0\_32

JAVA\_HOME=/usr/java/default/jre/

gpscp -f ~gpadmin/gpconfigs/hostfile\_segments -v $GPHOME/greenplum\_path.sh =:$GPHOME/

16. Restart database as gpadmin.

gpstop -M fast

Start database as gpadmin

gpstart

As gpadmin, run below command to validate hdfs clients installed and working fine -

gpssh -f ~gpadmin/gpconfigs/hostfile\_segments -v " hdfs dfs -ls /"

Issue :

db\_test3=# insert into wm\_ad\_hoc.ext\_repl\_sku\_wkly\_inv select \* from us\_wm\_tables.repl\_sku\_wkly\_inv limit 100;

ERROR: external table gphdfs protocol command ended with error. 17/02/28 15:15:44 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable (seg58 sdw8:1027 pid=234911)

DETAIL:

17/02/28 15:15:44 INFO security.UserGroupInformation: Login successful for user gpadmin@HADOOP\_PROD14.WAL-MART.COM using keytab file /home/gpadmin/.gpadmin.keytab

17/02/28 15:15:45 WARN shortcircuit.DomainSocketFactory: The short-circuit local reads feature cannot be used because libhadoop cannot be loaded.

Exception in thread "main" java.net.ConnectExce

Command: 'gphdfs://prod14/user/vicho1/us\_wm\_tables'

**Resolution**

here is the code snippet of $GPHOME/lib/hadoop/hadoop\_env.sh

Correct the native-hadoop library path as shown below:

if [ -d "${HADOOP\_HOME}/build/native" -o -d "${HADOOP\_HOME}/hadoop/lib/native" ]; then  
 JAVA\_PLATFORM=`CLASSPATH=${CLASSPATH} ${JAVA} -Xmx32m ${HADOOP\_JAVA\_PLATFORM\_OPTS} org.apache.hadoop.util.PlatformName | sed -e "s/ /\_/g"`

if [ -d "$HADOOP\_HOME/build/native" ]; then  
 JAVA\_LIBRARY\_PATH=${HADOOP\_HOME}/build/native/${JAVA\_PLATFORM}/lib  
 fi

if [ -d "${HADOOP\_HOME}/hadoop/lib/native" ]; then  
 if [ "x$JAVA\_LIBRARY\_PATH" != "x" ]; then  
 JAVA\_LIBRARY\_PATH=${JAVA\_LIBRARY\_PATH}:${HADOOP\_HOME}/hadoop/lib/native/  
 else  
 JAVA\_LIBRARY\_PATH=${HADOOP\_HOME}/hadoop/lib/native/  
 fi  
 fi  
fi

Issue: Java version issue

Possible Cause: When segments are completely reimaged from its mirrors

Fix:   Reinstall the Hadoop client on the faulty segment.

        Transfer hadoop configure files from master host to segment.

        Set the Hadoop related parameter and restart the database.

Note: For installation and parameter and path setup please refer the HDP client install plan.

Issue: If your unable to retrieve data from hdfs file system using Greenplum external table with gphdfs protocol because of the following error.

db\_prod1=# select \* from ext\_test\_hdfs ;

ERROR:  external table gphdfs protocol command ended with error. 17/02/27 04:04:51 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  (seg229 slice1 sdw157:40001 pid=8558)

DETAIL:

Exception in thread "main" java.io.IOException: Login failure for [gpadmin@HADOOP\_PAS11.WAL-MART.COM](mailto:gpadmin@HADOOP_PAS11.WAL-MART.COM) from keytab /home/gpadmin/.gpadmin.keytab: javax.security.auth.login.LoginException: Do not have keys of types listed in default\_tkt\_enctypes available; only have keys of following type: AES128 CTS mode with HMAC SHA1-96

       at org.apache.hadoop.security.Use

Command: 'gphdfs://pas11ha/tmp/test1.txt'

External table ext\_test\_hdfs, file gphdfs://pas11ha/tmp/test1.txt

Possible Cause: Change in the encryption type being used at Hadoop system

Fix: Get new keytab for gpadmin user (.gpadmin.keytab) and krb5.conf from Hadoop. Copy the new keytab and krb5.conf file to all segments.

Validation of HDP client and gphdfs protocol

1. Create below external tables to validate all hadoop commands works fine -

DROP EXTERNAL TABLE ext\_hdfs\_cmd\_test ;

CREATE EXTERNAL WEB TABLE ext\_hdfs\_cmd\_test (hadoop text)

EXECUTE '/usr/bin/hdfs dfs -ls /' ON HOST

FORMAT 'TEXT'

(DELIMITER '|') ;

CREATE EXTERNAL WEB TABLE ext\_java\_home (java text, hadoop text)

EXECUTE 'echo "$JAVA\_HOME|$HADOOP\_HOME"'

FORMAT 'TEXT'

(DELIMITER '|') ;

1. Copy a sample file to hdfs as below

$ hdfs dfs -cat /tmp/test1.txt

15,west

25,east

25,east

1. Create a sample external and access data from sample file created above

create external table ext\_test\_hdfs (age int, name text)

location('gphdfs://pas11ha/tmp/test1.txt')

format 'text'

(delimiter ',');

db\_test3=# select \* from ext\_test\_hdfs;

age | name

-----+------

15 | west

25 | east

25 | east