Installing and Managing HIVE

<https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-centos-7>

With remote metastore set up on Centos 7 distributed Hadoop on four nodes

1. First log in as username from which you can access Hadoop

su Hadoop

1. Update your operating system : yum update
2. Download mysql rpm wget <https://dev.mysql.com/get/mysql80-community-release-el7-2.noarch.rpm>
3. Check sum to see if the download is fine.

md5sum mysql80-community-release-el7-2.noarch.rpm

1. On start installing mysql server

sudo rpm -ivh mysql57-community-release-el7-9.noarch.rpm

1. sudo yum install mysql-server
2. start mysql

sudo systemctl status mysqld

sudo systemctl enable mysqld

1. get the temporary password
2. sudo mysql\_secure\_installation

copy password and set new password

1. check version mysqladmin -u root -p version

update /etc/my.cfg with bind-address=YOUR-SERVER-IP

1. restart mysqld
2. while creating new user if error comes then update

SET GLOBAL validate\_password.length = 6;

SET GLOBAL validate\_password.policy=LOW;

mysql> CREATE user 'hadoop'@'c1' IDENTIFIED BY 'password1$';

Query OK, 0 rows affected (0.00 sec)

mysql> CREATE user 'hadoop'@'c2' IDENTIFIED BY 'password1$';

GRANT ALL ON \*.\* TO 'hadoop'@'c3';

mysql -u 'hadoop'@'c2' -p -h c2

1. install client on machines from where we need to access mysql

sudo yum install mysql

sudo firewall-cmd --zone=public --add-port=3036/tcp –permanent

On all servers open port 3306

[hadoop@c3 ~]$ sudo firewall-cmd --zone=public --add-port=3306/tcp --permanent

[sudo] password for hadoop:

success

[hadoop@c3 ~]$ systemctl restart firewalld

[hadoop@c3 ~]$ systemctl restart network

Create database

Now start installing hive

1. from mirror download binary tar.

wget <http://mirrors.estointernet.in/apache/hive/stable-2/apache-hive-2.3.4-bin.tar.gz>

1. untar

tar -xvf apache-hive-2.3.4-bin.tar.gz

1. move the untared firectory to /usr/local
2. now create a link hive for hive

sudo ln -s apche-hive-x.y.z hive

1. Now make changes in hive configuration

Vi hive-env.sh

Mv hive-env.sh.template hive-env.sh

1. Provide Hadoop\_HOME in Hadoop-env.sh
2. Download mysql connector

wget <https://dev.mysql.com/get/mysql-connector-java-8.0.16-1.el7.noarch.rpm>

1. Install mysql connector

sudo rpm -ivh mysql-connector-java-8.0.16-1.el7.noarch.rpm

1. You will find mysql-coonector-java.jar in /usr/share/java
2. Copy this jar in /usr/local/hive/lib
3. Start bin/hive
4. Try creating a demo table

Error

<https://stackoverflow.com/questions/35449274/java-lang-runtimeexception-unable-to-instantiate-org-apache-hadoop-hive-ql-meta>

<value>jdbc:mysql://c2:3306/hive?createDatabaseIfNotExist=true&amp;useSSL=false</value>

hive --service metastore

schematool -initSchema -dbType mysql

<https://stackoverflow.com/questions/51564627/ssl-warning-in-hive>

<https://stackoverflow.com/questions/49576496/why-does-hive-return-failed-semanticexception-unable-to-instantiate>

programming hive

1. Create database

create database classicmodels;

1. Create table

CREATE TABLE IF NOT EXISTS products(product\_id int, product\_category\_id int, product\_name String, product\_description String, product\_price float, product\_image String) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' stored as TEXTFILE;

1. Two type of tables

External and internal

Internal tables :: Internal table are like normal database table where data can be stored and queried on. On dropping these tables the data stored in them also gets deleted and data is lost forever. So one should be careful while using internal tables as one drop command can destroy the whole data. Open new terminal and fire up hive by just typing hive. Create table on weather data.

> CREATE TABLE IF NOT EXISTS weather(wban INT, date\_w STRING, precip INT)

> ROW FORMAT DELIMITED

> FIELDS TERMINATED BY ','

> LINES TERMINATED BY '\n';

Check key contraints in mysql

show INDEX from products where key\_name = "PRIMARY";

SELECT k.COLUMN\_NAME

FROM information\_schema.table\_constraints t

LEFT JOIN information\_schema.key\_column\_usage k

USING(constraint\_name,table\_schema,table\_name)

WHERE t.constraint\_type='PRIMARY KEY'

AND t.table\_schema=DATABASE()

AND t.table\_name='owalog';

1. DESCRIBE products; check table
2. Directly create table and insert data in hive through sqoop

./sqoop-import --connect "jdbc:mysql://c2:3306/classicmodels?allowPublicKeyRetrieval=true&useSSL=false" --username hadoop --password password1$ --table customers --warehouse-dir /user/hive/warehouse/ --hive-import --hive-home /usr/local/hive --create-hive-table --hive-table classicmodels.products -m 1

1. Starting hverserver2

nohup hiveserver2 &

1. ./sqoop-import --connect "jdbc:mysql://c2:3306/classicmodels?allowPublicKeyRetrieval=true&useSSL=false" --username hadoop --password password1$ --table customers --warehouse-dir /user/hive/wahouse/ --hive-import --connect "jdbc:hive2://c1:10001/custommodels" --hive-home /usr/local/hive --create-hive-table --hive-table classicmodels.products --m 1

./sqoop-import --connect "jdbc:mysql://c2:3306/classicmodels?allowPublicKeyRetrieval=true&useSSL=false" --username hadoop --password password1$ --table customers --warehouse-dir /user/hive/warehouse/ --hive-import --connect "jdbc:hive://c1:10000/classicmodels" --hive-home /usr/local/hive --create-hive-table --hive-table classicmodels.customers --m 1

User: server is not allowed to impersonate anonymous

Here server is the user which is attempting to impersonate anonymous user.

Add these properties to core-site.xml and restart the services.

<property>

<name>hadoop.proxyuser.server.hosts</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.server.groups</name>

<value>\*</value>

</property>

SLF4J Warning: Class Path Contains Multiple SLF4J Bindings

rm -r log4j-slf4j-impl-2.6.2.jar

flush hosts mysql

in mysql

FLUSH HOSTS;

For public key retrival error

useSSL=false&allowPublicKeyRetrieval=true