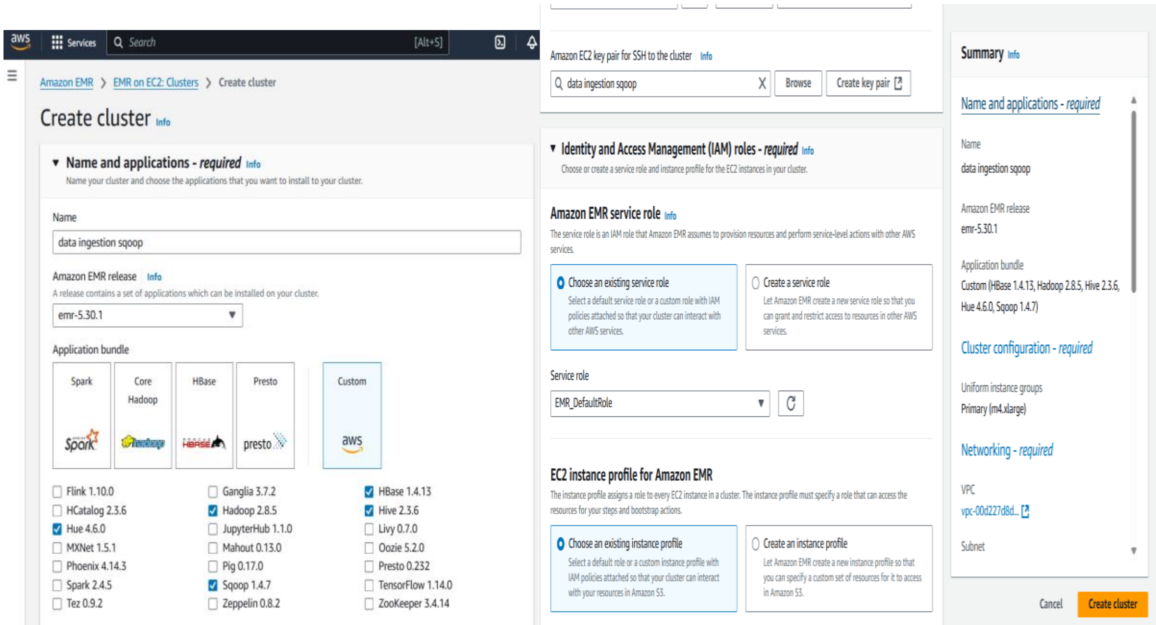
CREATING HBASE INSTANCE AND INGESTING DATA THROUGH RDS

CREATING AN HBASE INSTANCE:

- First of all, we will login to the AWS console and navigate to the EC2 dashboard. There we will click on Create EC2 Key Pair. This will create .ppk key file
- Next, we will navigate to the EMR dashboard and click on Create Cluster.
- We will add the configuration accordingly and click on the Create Cluster again.



Amazon EMR > EMR on EC2: Clusters > Create cluster

Create cluster Info

▼ Name and applications - *required* Info

Name your cluster and choose the applications that you want to install to your cluster.

Name

data ingestion sqoop

Amazon EMR release Info

A release contains a set of applications which can be installed on your cluster.

emr-5.30.1

Application bundle

☐ Flink 1.10.0
☐ HCatalog 2.3.6
☑ Hue 4.6.0
☐ MXNet 1.5.1
☐ Phoenix 4.14.3
☐ Spark 2.4.5
☐ Tez 0.9.2
☐ Ganglia 3.7.2
☑ Hadoop 2.8.5
☐ JupyterHub 1.1.0
☐ Mahout 0.13.0
☐ Pig 0.17.0
☑ Sqoop 1.4.7
☐ Zeppelin 0.8.2
☑ HBase 1.4.13
☑ Hive 2.3.6
☐ Livy 0.7.0
☐ Oozie 5.2.0
☐ Presto 0.232
☐ TensorFlow 1.14.0
☐ ZooKeeper 3.4.14

Amazon EC2 key pair for SSH to the cluster Info

data ingestion sqoop

▼ Identity and Access Management (IAM) roles - *required* Info

Choose or create a service role and instance profile for the EC2 instances in your cluster.

Amazon EMR service role Info

The service role is an IAM role that Amazon EMR assumes to provision resources and perform service-level actions with other AWS services.

● Choose an existing service role
Select a default service role or a custom role with IAM policies attached so that your cluster can interact with other AWS services.

○ Create a service role
Let Amazon EMR create a new service role so that you can grant and restrict access to resources in other AWS services.

Service role

EMR_DefaultRole

EC2 instance profile for Amazon EMR

The instance profile assigns a role to every EC2 instance in a cluster. The instance profile must specify a role that can access the resources for your steps and bootstrap actions.

● Choose an existing instance profile
Select a default role or a custom instance profile with IAM policies attached so that your cluster can interact with your resources in Amazon S3.

○ Create an instance profile
Let Amazon EMR create a new instance profile so that you can specify a custom set of resources for it to access in Amazon S3.

Summary Info

Name and applications - *required*

Name
data ingestion sqoop

Amazon EMR release
emr-5.30.1

Application bundle
Custom (HBase 1.4.13, Hadoop 2.8.5, Hive 2.3.6, Hue 4.6.0, Sqoop 1.4.7)

Cluster configuration - *required*

Uniform instance groups
Primary (m4.xlarge)

Networking - *required*

VPC
vpc-00d227d8d...

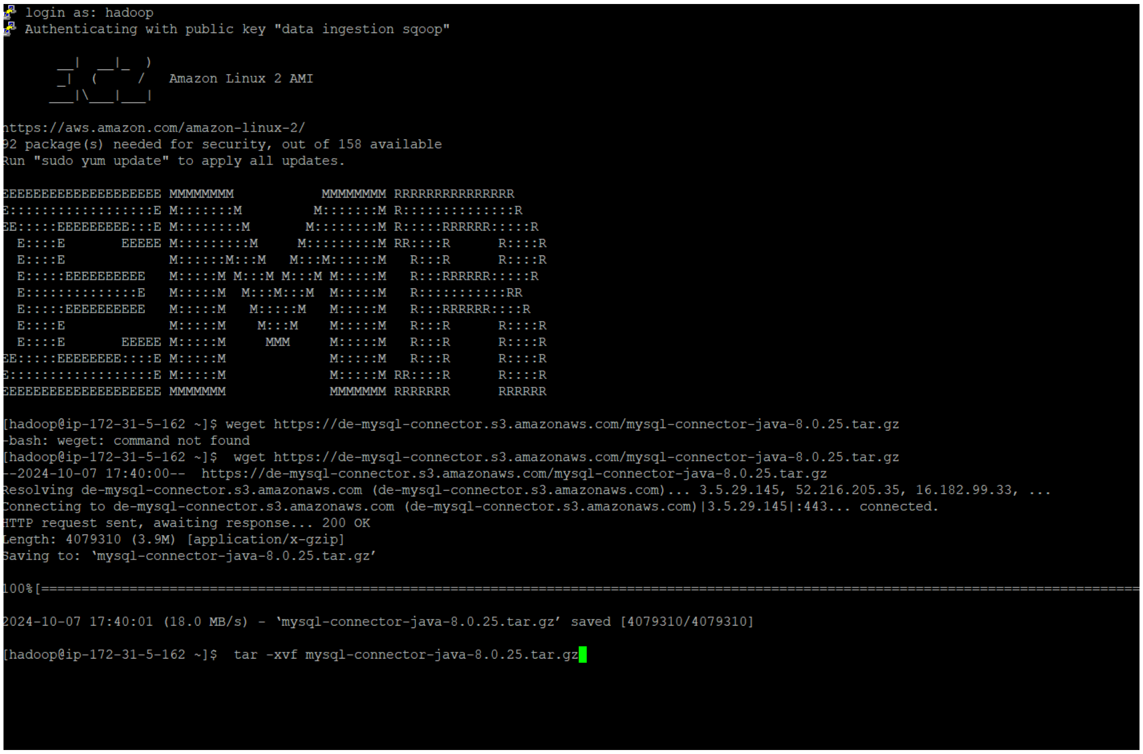
Subnet

- Now we will login into EMR instance and run the following command to install the MySQL connector jar file.

  wget https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz

- After that we will run the following command to extract the MySQL Connector tar file:

  tar -xvf mysql-connector-java-8.0.25.tar.gz



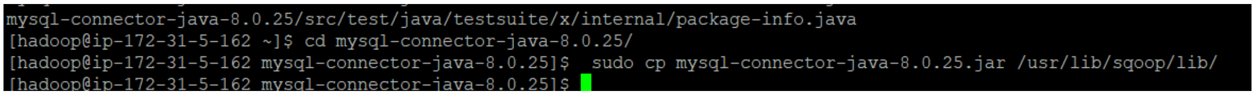
```
login as: hadoop
Authenticating with public key "data ingestion sqoop"

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
92 package(s) needed for security, out of 158 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-5-162 ~]$ weget https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz
-bash: weget: command not found
[hadoop@ip-172-31-5-162 ~]$  wget https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz
--2024-10-07 17:40:00--  https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz
Resolving de-mysql-connector.s3.amazonaws.com (de-mysql-connector.s3.amazonaws.com)... 3.5.29.145, 52.216.205.35, 16.182.99.33, ...
Connecting to de-mysql-connector.s3.amazonaws.com (de-mysql-connector.s3.amazonaws.com)|3.5.29.145|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4079310 (3.9M) [application/x-gzip]
Saving to: ‘mysql-connector-java-8.0.25.tar.gz’

100%[==============================================================================================================================

2024-10-07 17:40:01 (18.0 MB/s) - ‘mysql-connector-java-8.0.25.tar.gz’ saved [4079310/4079310]

[hadoop@ip-172-31-5-162 ~]$  tar -xvf mysql-connector-java-8.0.25.tar.gz
```

- Now, we need to go to the MySQL Connector directory created in the previous step and then copy it to the Sqoop library to complete the installation.

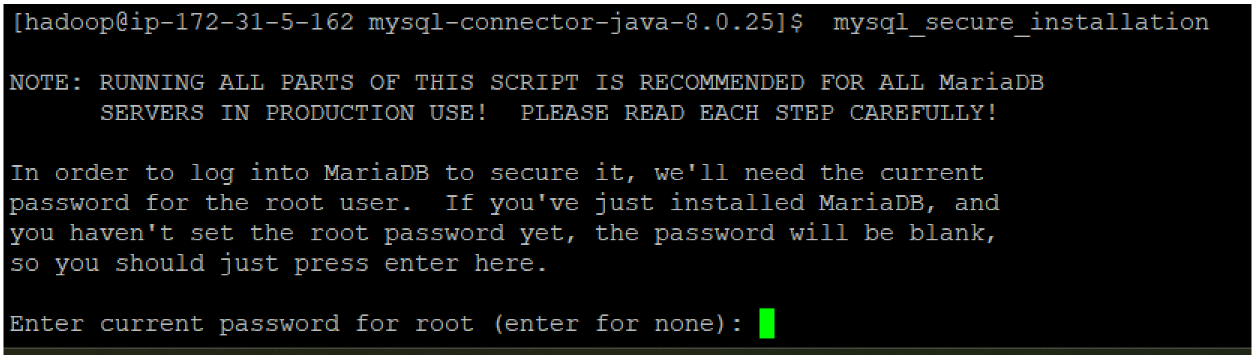
  cd mysql-connector-java-8.0.25/

  sudo cp mysql-connector-java-8.0.25.jar /usr/lib/sqoop/lib/

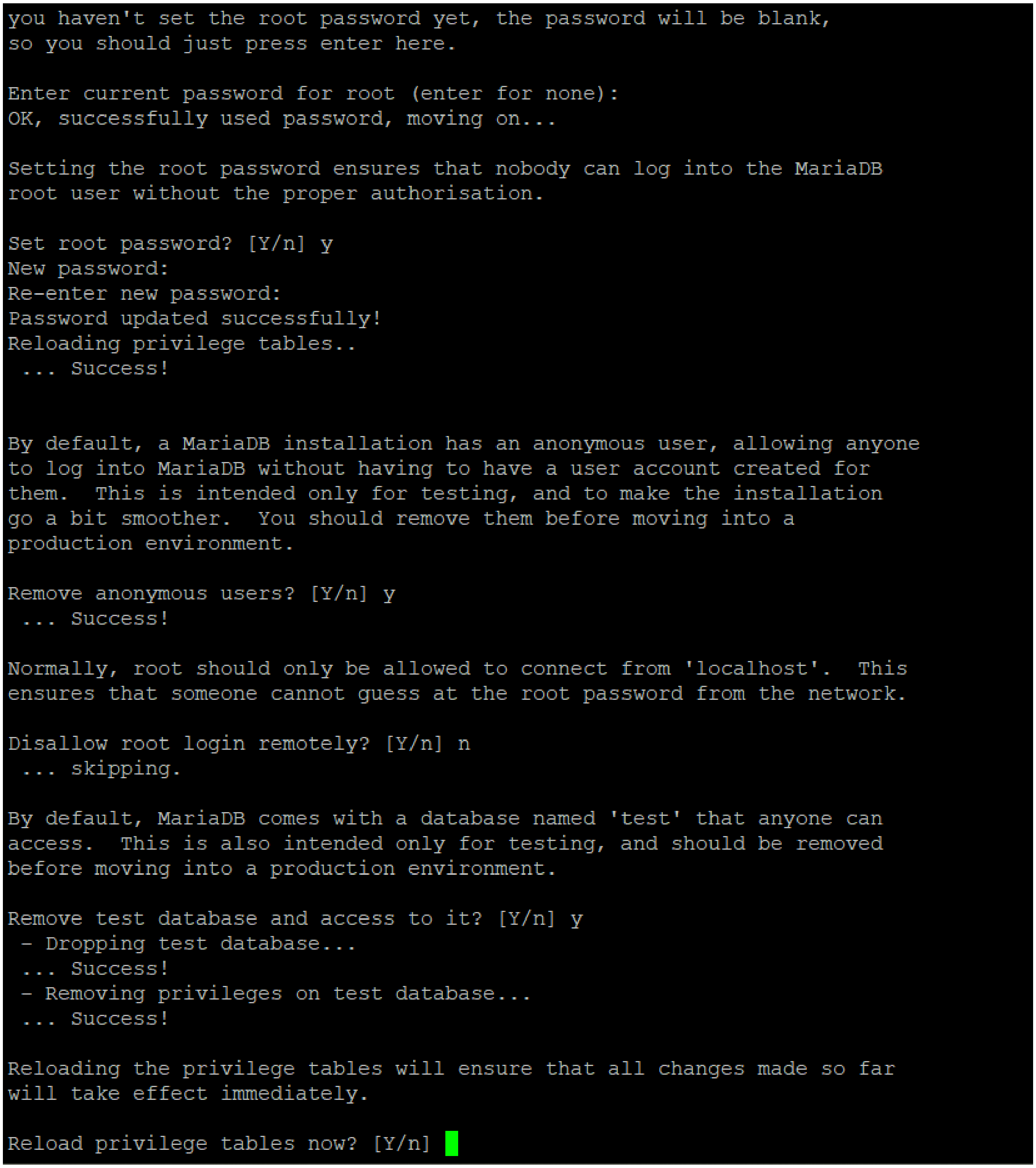
- Now we will run the following command to setup MySQL on EMR.
  mysql_secure_installation
- Now, the command will ask for the current password for root. We need to press Enter as no password is currently set for root.
- Then, we will be asked if we want to set the root password. We need to type Y and then press Enter.
- After pressing Y, we need to type 123 as the password and then press Enter. After that, we will need to confirm the password by typing the password, that is, 123, again.
- Now, we will be asked whether we want to remove anonymous users. Type Y and then press Enter.
- Next, we will be asked if we want to disallow root login remotely. Type n and then press Enter.
- Then after this, we will be asked whether we want to remove the test database along with its access. Type Y and then press Enter
- Now, we will be asked if we want to reload privilege tables. Type Y and then press Enter.



```
mysql-connector-java-8.0.25/src/test/java/testsuite/x/internal/package-info.java
[hadoop@ip-172-31-5-162 ~]$ cd mysql-connector-java-8.0.25/
[hadoop@ip-172-31-5-162 mysql-connector-java-8.0.25]$  sudo cp mysql-connector-java-8.0.25.jar /usr/lib/sqoop/lib/
[hadoop@ip-172-31-5-162 mysql-connector-java-8.0.25]$
```

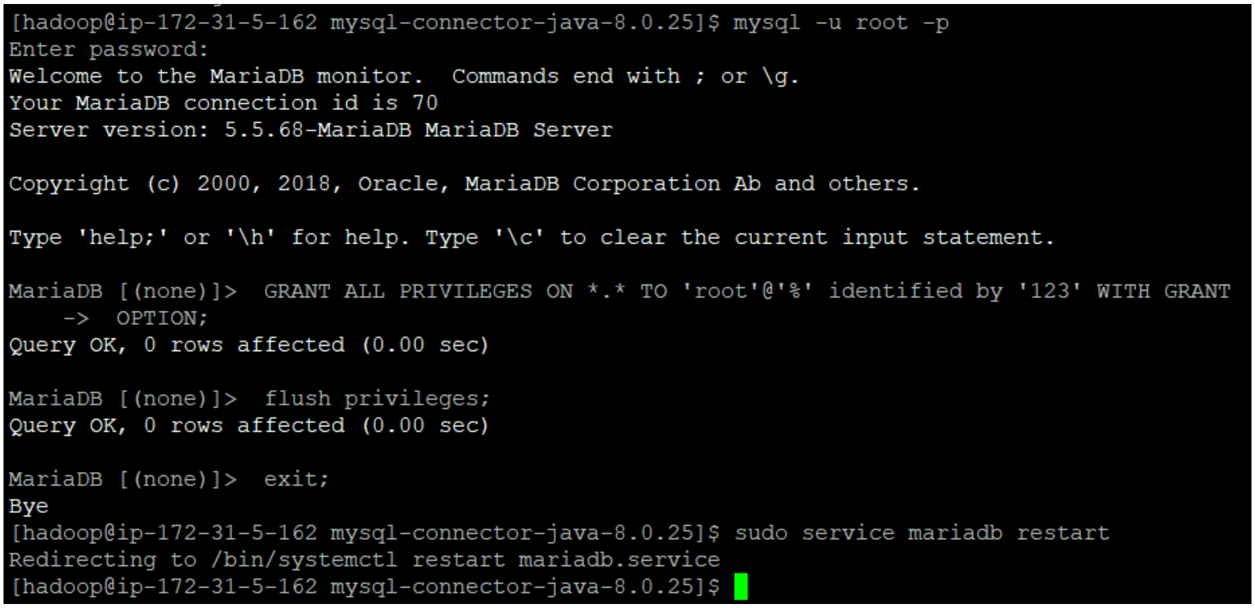


```
[hadoop@ip-172-31-5-162 mysql-connector-java-8.0.25]$  mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
```



```
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!


By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n
 ... skipping.

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n]
```

- With this, we have set up MySQL. we now need to access the MySQL shell. Enter the following command, type 123 when the password prompt comes up, and finally, press Enter.
  mysql –u root –p
- Once we have accessed MySQL, we need to run the following queries to grant all privileges to the root user.
  GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' identified by '123' WITH GRANT OPTION;
  flush privileges;
  exit;

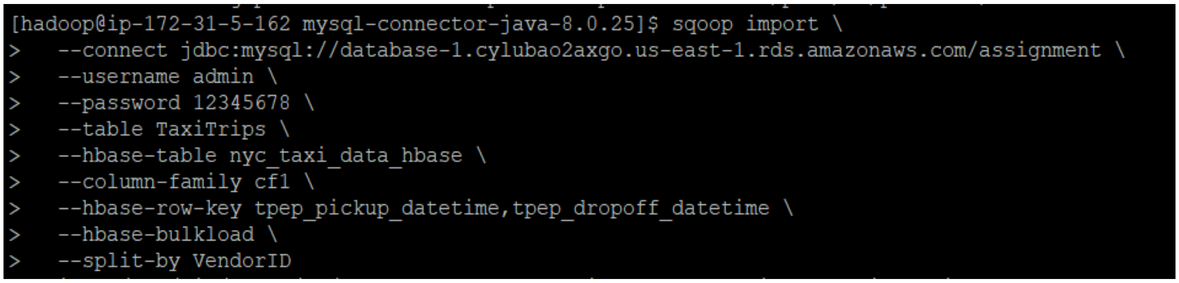


```
[hadoop@ip-172-31-5-162 mysql-connector-java-8.0.25]$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 70
Server version: 5.5.68-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>  GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' identified by '123' WITH GRANT
    ->  OPTION;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]>  flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]>  exit;
Bye
[hadoop@ip-172-31-5-162 mysql-connector-java-8.0.25]$ sudo service mariadb restart
Redirecting to /bin/systemctl restart mariadb.service
[hadoop@ip-172-31-5-162 mysql-connector-java-8.0.25]$
```

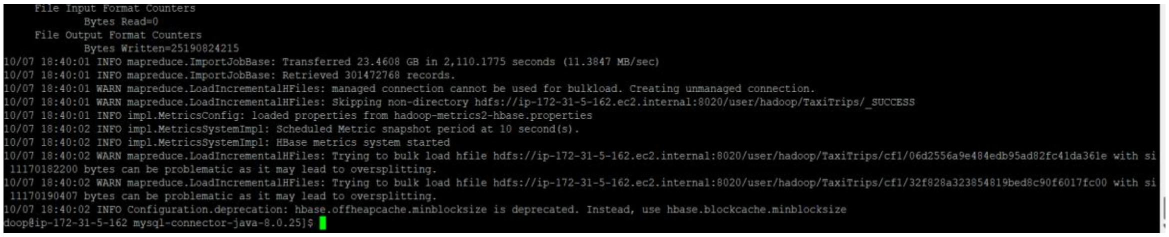
- Finally, we need to run the following command to restart the MySQL service to finish setting up MySQL.

  sudo service mariadb restart

- **Then we will write Sqoop script to import into HBase**
- sqoop import \
- --connect "jdbc:mysql://<RDS Endpoint>/<DB Name>" \
- --username <Username> \
- --password <Password> \
- --table <Tabler Name> \
- --target-dir <Path to Directory> \
- --hbase-table <HBase Table Name> --column-family cf1 --hbase-create-table \
- --hbase-row-key tpep_pickup_datetime,tpep_dropoff_datetime \
- --hbase-bulkload \
- --split-by VendorID



```
[hadoop@ip-172-31-5-162 mysql-connector-java-8.0.25]$ sqoop import \
>   --connect jdbc:mysql://database-1.cylubao2axgo.us-east-1.rds.amazonaws.com/assignment \
>   --username admin \
>   --password 12345678 \
>   --table TaxiTrips \
>   --hbase-table nyc_taxi_data_hbase \
>   --column-family cf1 \
>   --hbase-row-key tpep_pickup_datetime,tpep_dropoff_datetime \
>   --hbase-bulkload \
>   --split-by VendorID
```



```
    File Input Format Counters
            Bytes Read=0
    File Output Format Counters
            Bytes Written=25190824215
10/07 18:40:01 INFO mapreduce.ImportJobBase: Transferred 23.4608 GB in 2,110.1775 seconds (11.3847 MB/sec)
10/07 18:40:01 INFO mapreduce.ImportJobBase: Retrieved 301472768 records.
10/07 18:40:01 WARN mapreduce.LoadIncrementalHFiles: managed connection cannot be used for bulkload. Creating unmanaged connection.
10/07 18:40:01 WARN mapreduce.LoadIncrementalHFiles: Skipping non-directory hdfs://ip-172-31-5-162.ec2.internal:8020/user/hadoop/TaxiTrips/_SUCCESS
10/07 18:40:01 INFO impl.MetricsConfig: loaded properties from hadoop-metrics2-hbase.properties
10/07 18:40:02 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
10/07 18:40:02 INFO impl.MetricsSystemImpl: HBase metrics system started
10/07 18:40:02 WARN mapreduce.LoadIncrementalHFiles: Trying to bulk load hfile hdfs://ip-172-31-5-162.ec2.internal:8020/user/hadoop/TaxiTrips/cf1/06d2556a9e484edb95ad82fc41da361e with si
 11170182200 bytes can be problematic as it may lead to oversplitting.
10/07 18:40:02 WARN mapreduce.LoadIncrementalHFiles: Trying to bulk load hfile hdfs://ip-172-31-5-162.ec2.internal:8020/user/hadoop/TaxiTrips/cf1/32f828a323854819bed8c90f6017fc00 with si
 11170190407 bytes can be problematic as it may lead to oversplitting.
10/07 18:40:02 INFO Configuration.deprecation: hbase.offheapcache.minblocksize is deprecated. Instead, use hbase.blockcache.minblocksize
doop@ip-172-31-5-162 mysql-connector-java-8.0.25]$ 
```