

# Capstone: Instant Health Alert System – Final Submission

## *A script to extract patient info using Sqoop into hive table*

### **Sqoop Setup**

Following steps are followed to setup Sqoop on EMR Cluster

1. To install the MySQL connector jar file.

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

```
[hadoop@ip-172-31-83-130 ~]$ wget https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz
--2023-03-25 07:08:33-- 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)... 54.231.130.1, 3.5.16.186, 3.5.20.112, ...
Connecting to de-mysql-connector.s3.amazonaws.com (de-mysql-connector.s3.amazonaws.com) |54.231.130.1|: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%[=====>] 4,079,310 20.6MB/s in 0.2s

2023-03-25 07:08:34 (20.6 MB/s) - 'mysql-connector-java-8.0.25.tar.gz' saved [4079310/4079310]
```

2. Extract the MySQL connector tar file

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

```
[hadoop@ip-172-31-83-130 ~]$ tar -xvf mysql-connector-java-8.0.25.tar.gz
mysql-connector-java-8.0.25/
mysql-connector-java-8.0.25/src/
mysql-connector-java-8.0.25/src/build/
mysql-connector-java-8.0.25/src/build/java/
mysql-connector-java-8.0.25/src/build/java/documentation/
mysql-connector-java-8.0.25/src/build/java/instrumentation/
mysql-connector-java-8.0.25/src/build/misc/
mysql-connector-java-8.0.25/src/build/misc/debian.in/
mysql-connector-java-8.0.25/src/build/misc/debian.in/source/
mysql-connector-java-8.0.25/src/demo/
mysql-connector-java-8.0.25/src/demo/java/
mysql-connector-java-8.0.25/src/demo/java/demo/
mysql-connector-java-8.0.25/src/demo/java/demo/x/
mysql-connector-java-8.0.25/src/demo/java/demo/x/devapi/
mysql-connector-java-8.0.25/src/generated/
mysql-connector-java-8.0.25/src/generated/java/
mysql-connector-java-8.0.25/src/generated/java/com/
mysql-connector-java-8.0.25/src/generated/java/com/mysql/
mysql-connector-java-8.0.25/src/generated/java/com/mysql/cj/
mysql-connector-java-8.0.25/src/generated/java/com/mysql/cj/x/
mysql-connector-java-8.0.25/src/generated/java/com/mysql/cj/x/protobuf/
mysql-connector-java-8.0.25/src/legacy/
mysql-connector-java-8.0.25/src/legacy/java/
mysql-connector-java-8.0.25/src/legacy/java/com/
mysql-connector-java-8.0.25/src/legacy/java/com/mysql/
mysql-connector-java-8.0.25/src/legacy/java/com/mysql/jdbc/
```

3. Go to the MySQL Connector directory created in the previous step and 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/
```

```
[hadoop@ip-172-31-83-130 ~]$ cd mysql-connector-java-8.0.25/
[hadoop@ip-172-31-83-130 mysql-connector-java-8.0.25]$ sudo cp mysql-connector-j
ava-8.0.25.jar /usr/lib/sqoop/lib/
```

#### 4. Set up MySQL on your EMR cluster (Inside this folder mysql-connector-java-8.0.25)

mysql\_secure\_installation

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

Set root password [Y/n] Y

New password: 123

Re-enter password: 123

Remove anonymous users [Y/n] Y

Disallow root login remotely [Y/n] n

Remove test database and access to it [Y/n] Y

Reload privilege tables now [Y/n] Y

```
[hadoop@ip-172-31-83-130 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):

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

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] Y
... Success!
```

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

Thanks for using MariaDB!

6. With this, MySQL setup is done. Now, we can access the MySQL shell. Enter the following command, type 123 when the password prompt comes up, and finally, press Enter.

```
mysql -u root -p
```

```
[hadoop@ip-172-31-83-130 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 66
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'
```

7. Inside MariaDB (MariaDB > )

Following queries need to be run for granting all privileges to the root user.

```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' identified by '123' WITH GRANT OPTION;
flush privileges;
exit;
```

```
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
```

8. Restart the MySQL service to finish setting up MySQL. (Inside this folder mysql-connector-java-8.0.25 )

```
sudo service mariadb restart
```

```
[hadoop@ip-172-31-83-130 mysql-connector-java-8.0.25]$ sudo service mariadb rest
art
Redirecting to /bin/systemctl restart mariadb.service
```

9. Change the directory (come outside mysql-connector-java-8.0.25 folder)

```
cd ..
```

## Sqoop Commands

1. Import data to HDFS

```
sqoop import --connect jdbc:mysql://upgraddetest.cyaielc9bmnf.us-east-1.rds.amazonaws.com/testdatabase --table patients_information --username student --password STUDENT123 --target-dir /user/livy/patient_contact_info -m 1
```

```
[root@ip-172-31-83-130 ~]# sqoop import --connect jdbc:mysql://upgraddetest.cyaielc9bmnf.us-east-1.rds.amazonaws.com/testdatabase --table patients_information --username student --password STUDENT123 --target-dir /user/livy/patient_contact_info -m 1
```

```
Other local map tasks=1
Total time spent by all maps in occupied slots (ms)=161328
Total time spent by all reduces in occupied slots (ms)=0
Total time spent by all map tasks (ms)=3361
Total vcore-milliseconds taken by all map tasks=3361
Total megabyte-milliseconds taken by all map tasks=5162496
Map-Reduce Framework
  Map input records=5
  Map output records=5
  Input split bytes=87
  Spilled Records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=67
  CPU time spent (ms)=1890
  Physical memory (bytes) snapshot=261730304
  Virtual memory (bytes) snapshot=3281002496
  Total committed heap usage (bytes)=247463936
File Input Format Counters
  Bytes Read=0
File Output Format Counters
  Bytes Written=230
23/03/25 07:11:39 INFO mapreduce.ImportJobBase: Transferred 230 bytes in 20.9571 seconds (10.9748 bytes/sec)
23/03/25 07:11:39 INFO mapreduce.ImportJobBase: Retrieved 5 records.
```

2. View the list of files in HDFS target directory

```
hadoop fs -ls /user/livy/patient_contact_info
```

```
[root@ip-172-31-83-130 ~]# hadoop fs -ls /user/livy/patient_contact_info
Found 2 items
-rw-r--r--  1 root livy          0 2023-03-25 07:11 /user/livy/patient_contact_info/_SUCCESS
-rw-r--r--  1 root livy      230 2023-03-25 07:11 /user/livy/patient_contact_info/part-m-00000
```

3. View the imported contents in HDFS file

```
hadoop fs -cat /user/livy/patient_contact_info/part-m-00000
```

```
[root@ip-172-31-83-130 ~]# hadoop fs -cat /user/livy/patient_contact_info/part-m-00000
1,Alex S,XDC test Address,8982739282,1,23,null
2,Sammy A,New Building Address,2382739282,2,45,null
3,Karan C,Aws Address,8923739282,3,56,null
4,Dara M,India Address,2182739282,4,67,null
5,Pam,ABC test Address,4982739282,5,72,null
```

Set up for the EMR Cluster

Screenshot of EMR cluster (with Spark, Hive, Sqoop)

Cluster: SparkwithHiveSqoopcluster Starting

SummaryApplication user interfacesMonitoringHardwareConfigurationsEventsStepsBootstrap actions

Summary

ID: j-1RGRZKURC1L6Q

Creation date: 2023-03-28 14:00 (UTC+5:30)

Elapsed time: 0 seconds

After last step completes: Cluster waits

Termination protection: Off [Change](#)

Tags: -- [View All](#) / [Edit](#)

Master public DNS: --

Configuration details

Release label: emr-5.30.1

Hadoop distribution: Amazon 2.8.5

Applications: Spark 2.4.5, JupyterHub 1.1.0, Zeppelin 0.8.2, Livy 0.7.0, Hive 2.3.6, HCatalog 2.3.6, Sqoop 1.4.7

Log URI: s3://aws-logs-545120555452-us-east-1/elasticmapreduce/

EMRFS consistent view: Disabled

Custom AMI ID: --

Application user interfaces

Persistent user interfaces : --

On-cluster user interfaces : --

Security and access

Key name: MyNew\_KeyValue

EC2 instance profile: EMR\_EC2\_DefaultRole

EMR role: EMR\_DefaultRole

Network and hardware

Availability zone: --

Subnet ID: [subnet-007e940c0eb58c44b](#)

Master: Provisioning 1 m4.xlarge

Core: --

Task: --

Cluster scaling: Not enabled

Auto-termination: Not enabled

Screenshot of EMR Hardware Configuration (with 1 master node and 2 core nodes)

Cluster: SparkwithHiveSqoopcluster Terminated Terminated by user request

SummaryApplication user interfacesMonitoringHardwareConfigurationsEventsStepsBootstrap actions

Add task instance group

Instance groups

Filter:  2 instance groups (all loaded)

ID	Status	Node type & name	Instance type	Instance count	Purchasing option
<a href="#">ig-14KVR95NI797O</a>	Terminated (1 Requested)	MASTER Master - 1	m4.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 70 GiB	0 Instances	On-demand ⓘ
<a href="#">ig-H78HMFXPML5O</a>	Terminated	CORE Core - 2	m4.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB	0 Instances	On-demand ⓘ

Hive table (for Patients\_Contact\_Info)

- Open Hive shell.

```
^C[hadoop@ip-172-31-82-68 ~]$ hive
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: true
```

- Create a database *patient\_health\_care*

create database if not exists patient\_health\_care;

```
hive> create database if not exists patient_health_care;
OK
Time taken: 0.855 seconds
```

- Use database patient\_health\_care

```
use patient_health_care;
```

```
hive> use patient_health_care;
OK
Time taken: 0.046 seconds
```

- Create external table named *Patients\_Contact\_Info*

```
CREATE EXTERNAL TABLE IF NOT EXISTS Patients_Contact_Info (
  patientid int,
  patientname string,
  patientaddress string,
  phone_number string,
  admitted_ward int,
  age int,
  other_details string
)
row format delimited
fields terminated by ','
lines terminated by '\n'
location '/user/livy/patient_contact_info';
```

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS Patients_Contact_Info (
>   patientid int,
>   patientname string,
>   patientaddress string,
>   phone_number string,
>   admitted_ward int,
>   age int,
>   other_details string
> )
> row format delimited
> fields terminated by ','
> lines terminated by '\n'
> location '/user/livy/patient_contact_info';
OK
Time taken: 0.318 seconds
```

- View the records in *Patients\_Contact\_Info* table

```
select * from Patients_Contact_Info;
```

```
hive> select * from Patients_Contact_Info;
OK
patients_contact_info.patientid patients_contact_info.patientname      patients
_contact_info.patientaddress  patients_contact_info.phone_number  patients
_contact_info.admitted_ward   patients_contact_info.age          patients_contact
_info.other_details
1      Alex S   XDC test Address      8982739282      1      23      null
2      Sammy A New Building Address  2382739282      2      45      null
3      Karan C  Aws Address      8923739282      3      56      null
4      Dara M   India Address   2182739282      4      67      null
5      Pam     ABC test Address 4982739282      5      72      null
Time taken: 1.541 seconds, Fetched: 5 row(s)
```