**Convering Mysql data to Hive Data**

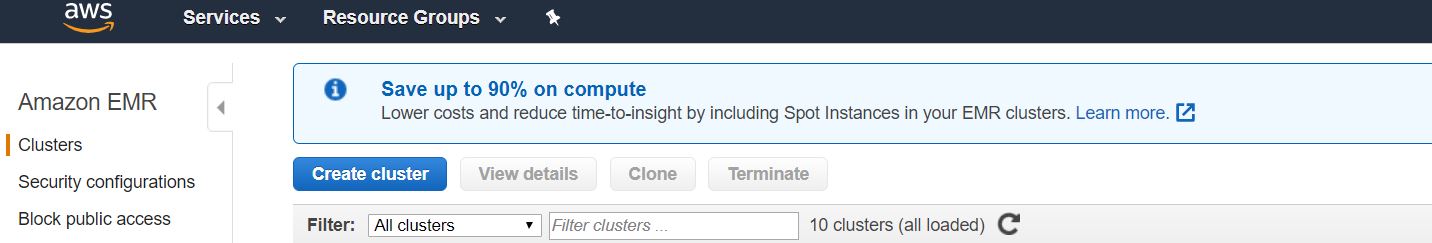
Pre-requisites:

* EMR Cluster
* Mysql in EC2 instance
* mysql-connector-java-8.0.18.jar file
* Pyspark Code

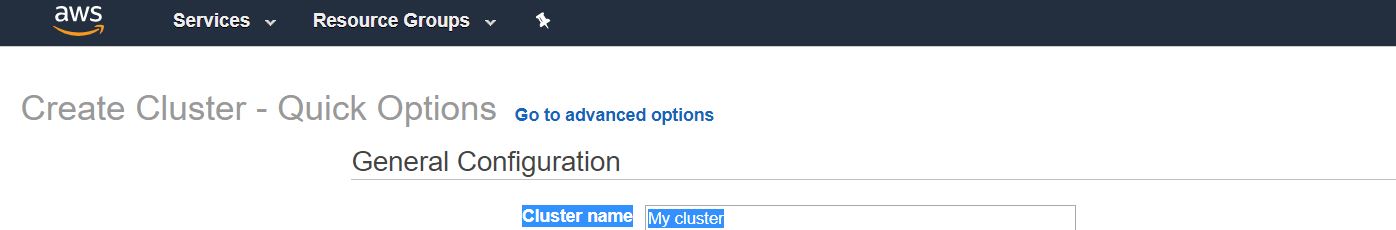
Step1:

EMR Cluster Creation:

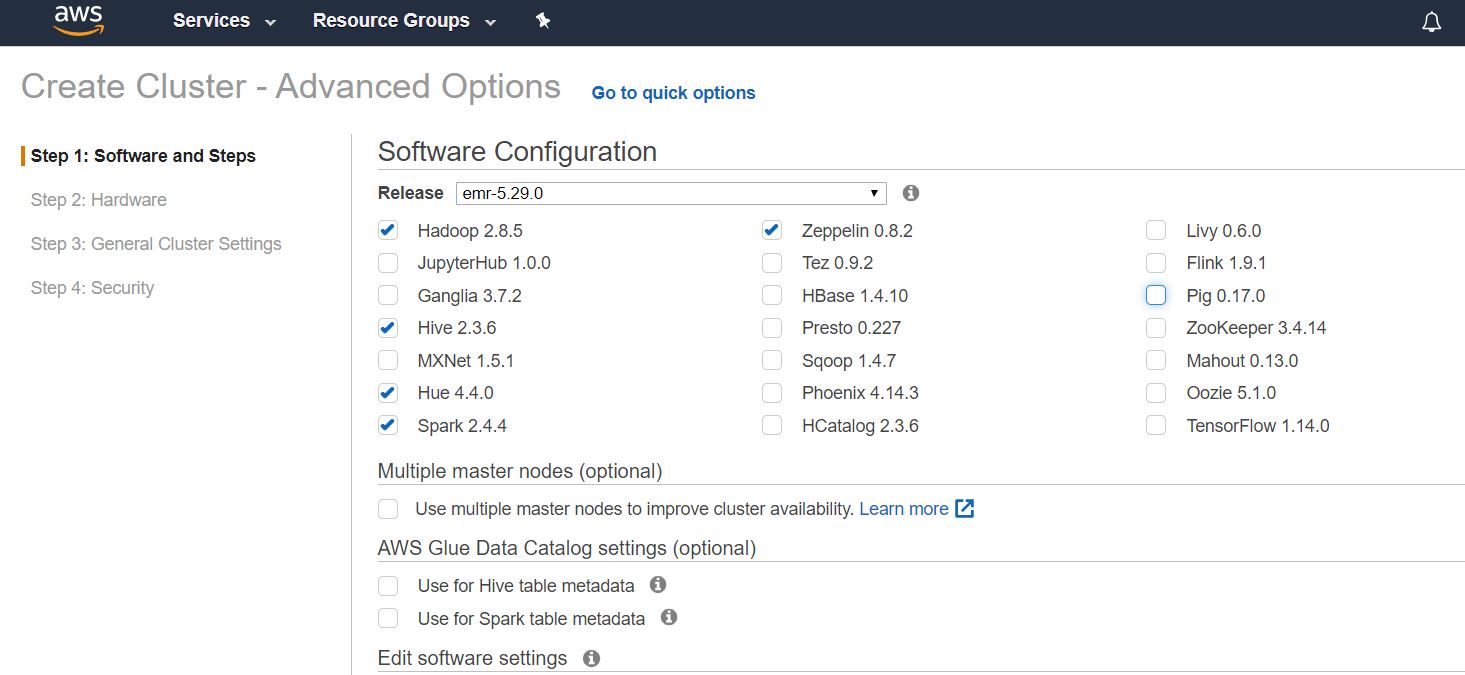
Open EMR Cluster UI in AWS and click on **create cluster**

****

Click on goto advance options and click on Next

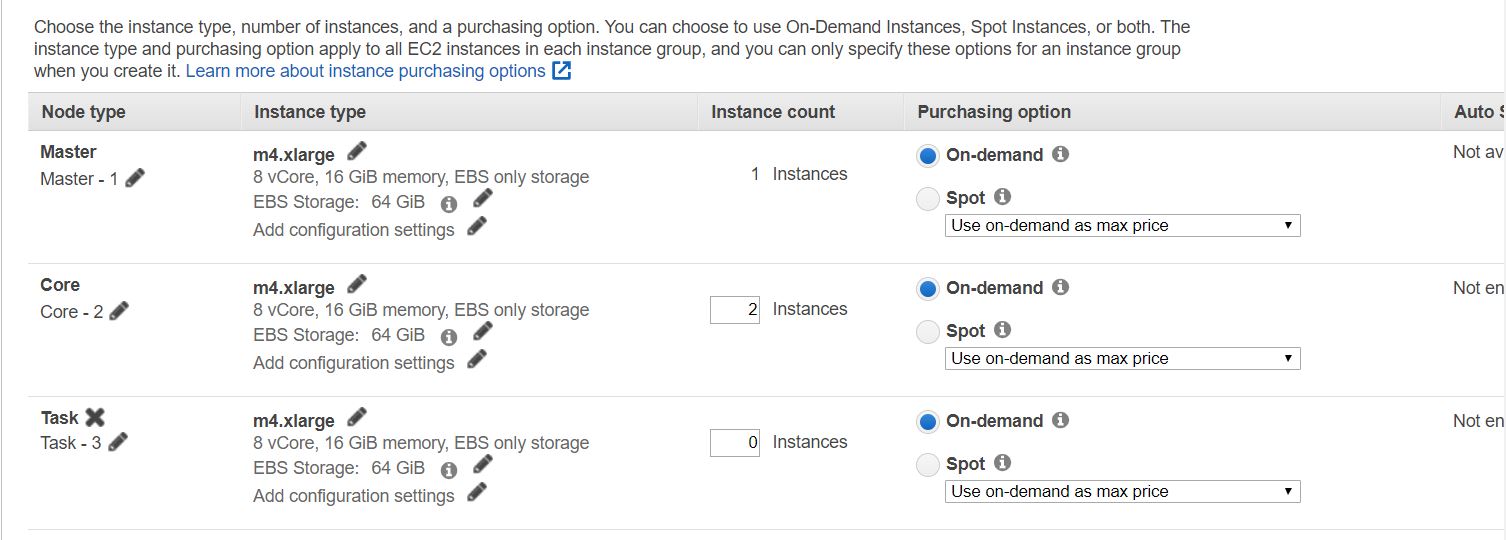
****

Select which I selected in below Image and Click on Next

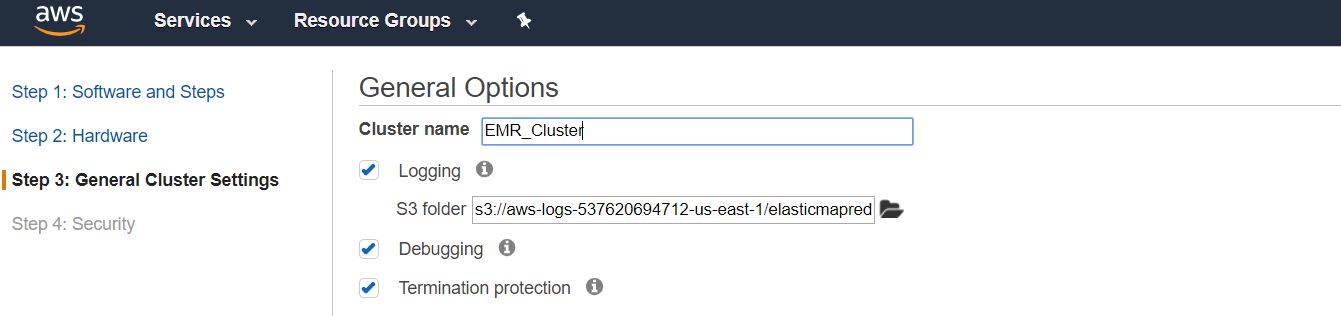
****

Here Instance type gives as m5.xlarge as default

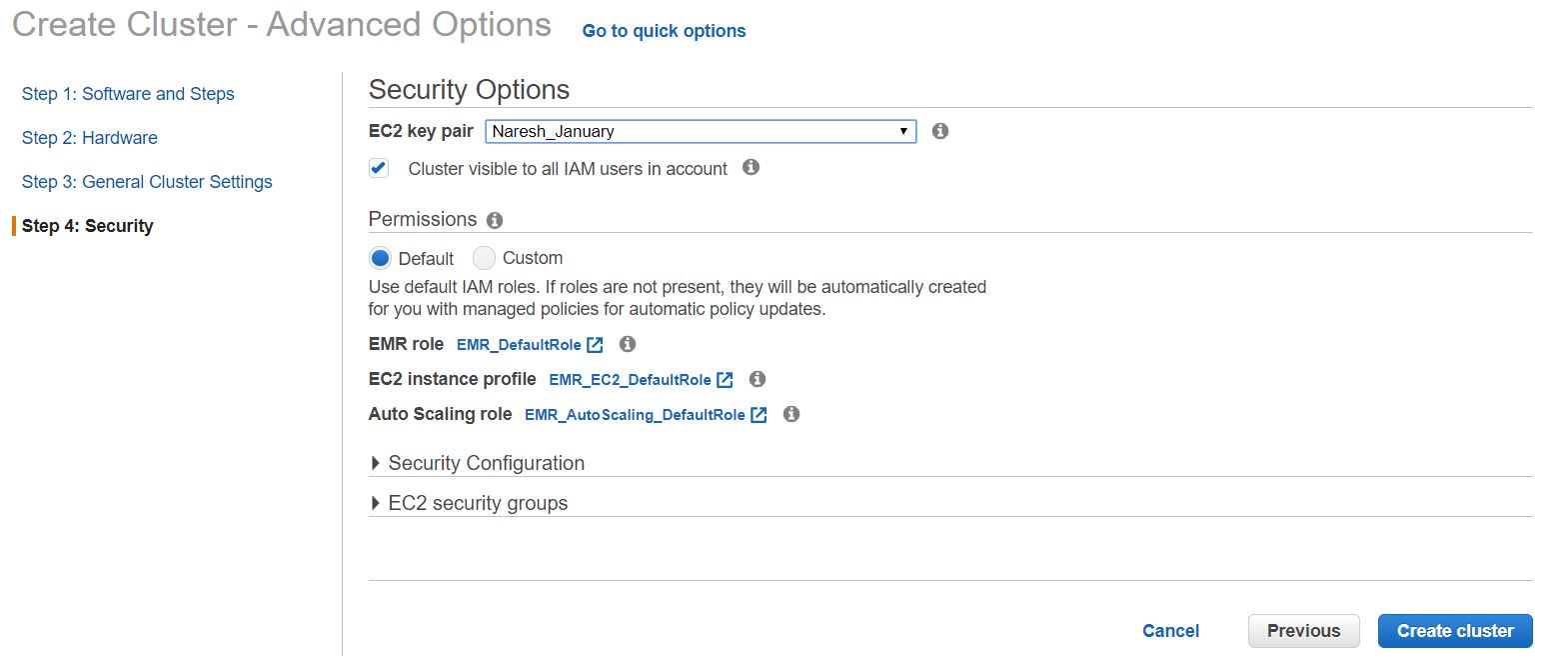
Try to create cluster with m5.xlarge, if you get any error you just try to create with m4.xlarge

****

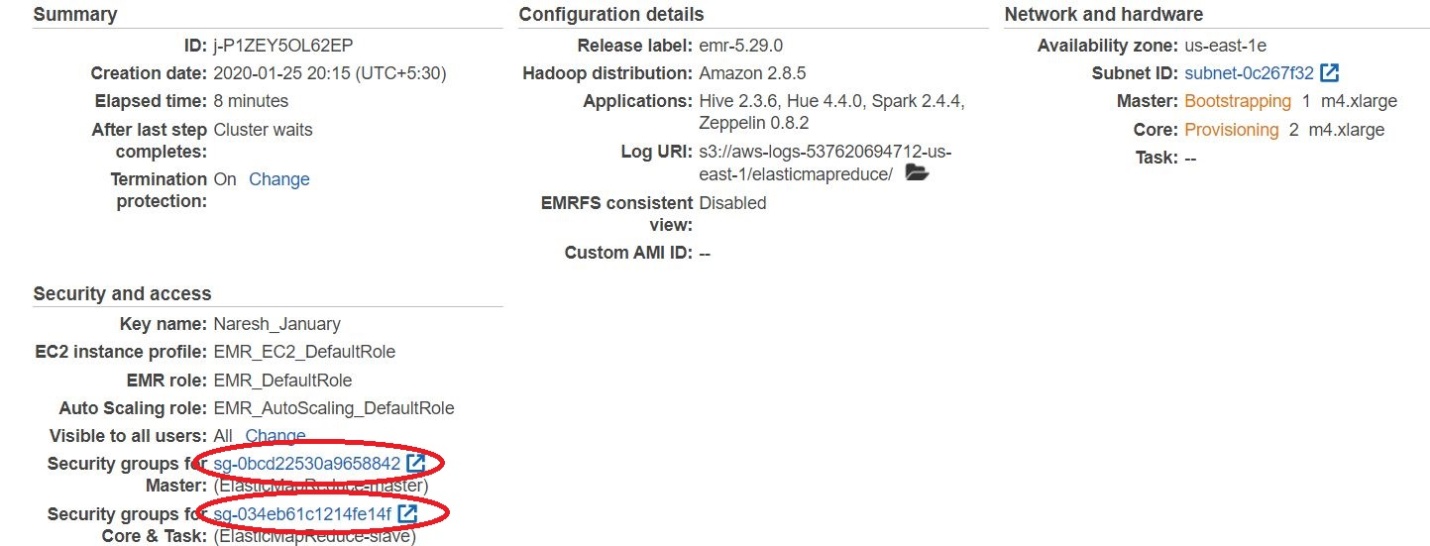
Give name for Cluster and click on Next

****

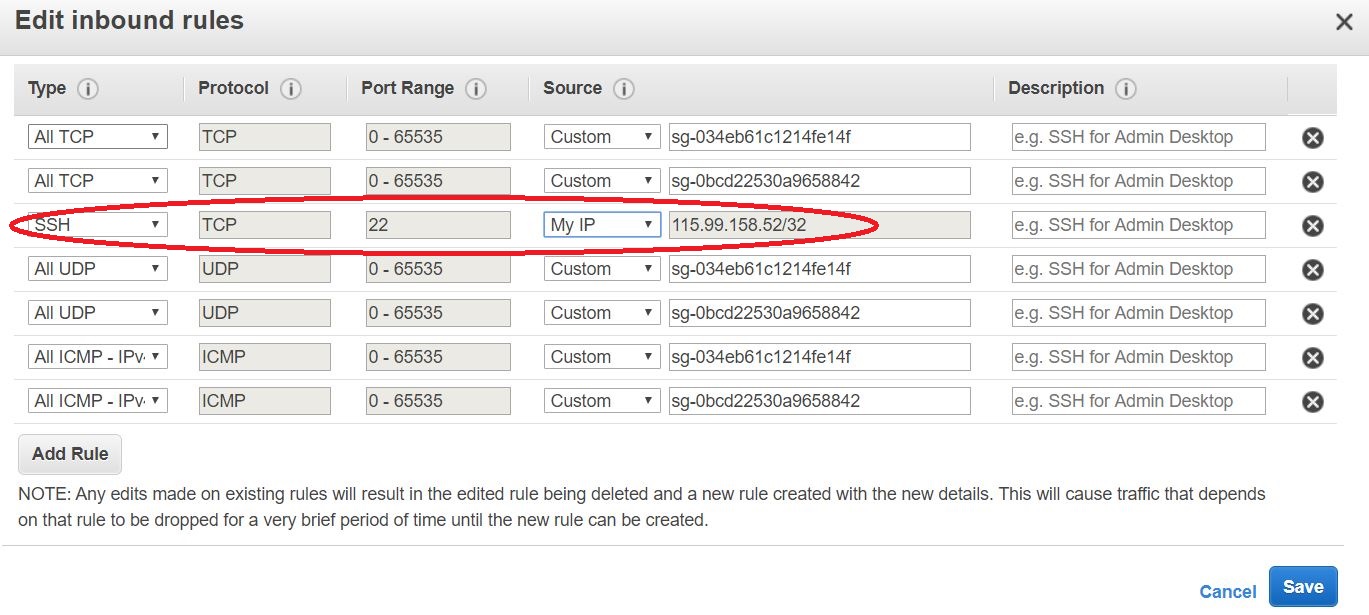
Give EC2 key pair which you in you system and click on Create cluster

****

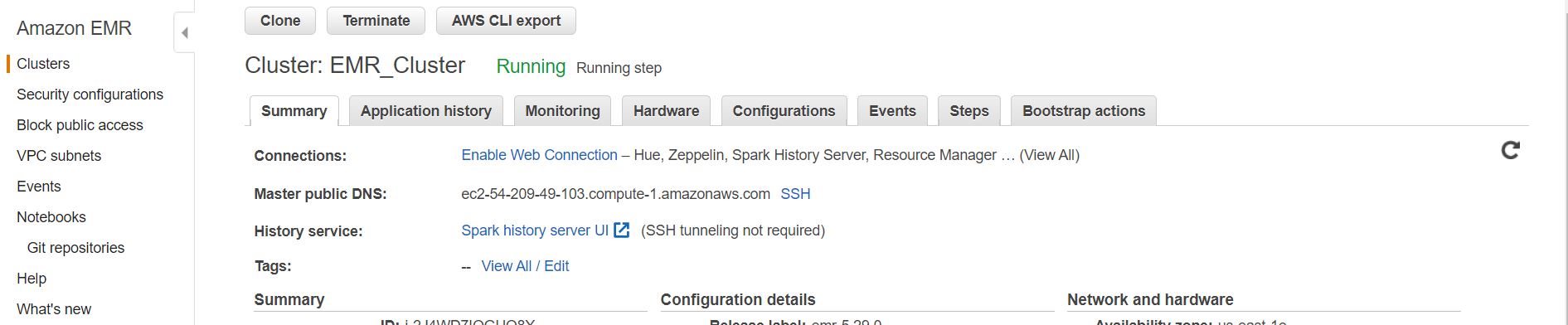
Wait until to create EMR Cluster, Mean while just open security group which we used for our cluster and add SSH with our IP address



Click on Security groups and allow SSH



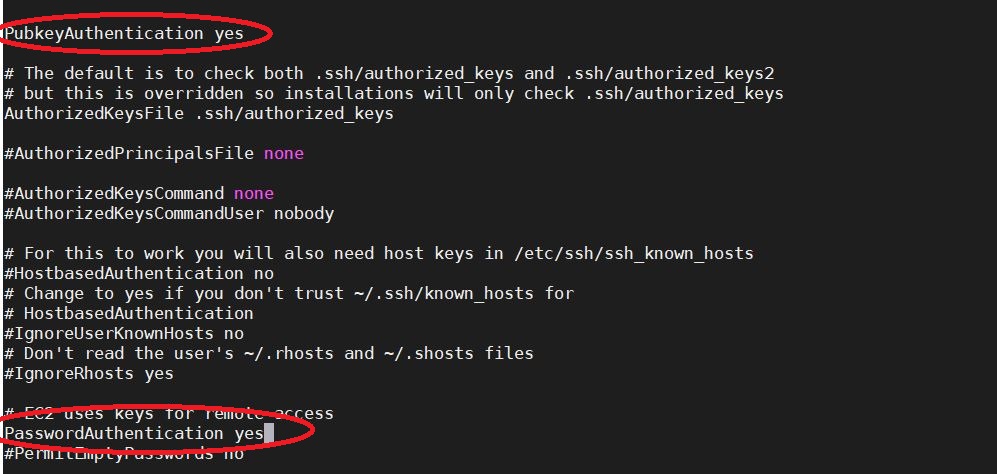
Go and Check whether our cluster is ready or not



Goto root user and set password for hadoop user

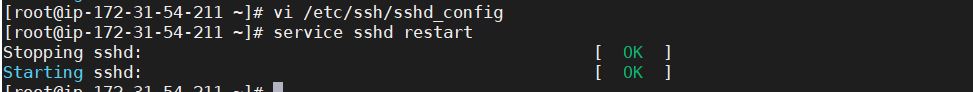
Step1: Open sshd\_config file and uncomment PubkeyAuthentication, enable password authentication

vi /etc/ssh/sshd\_config



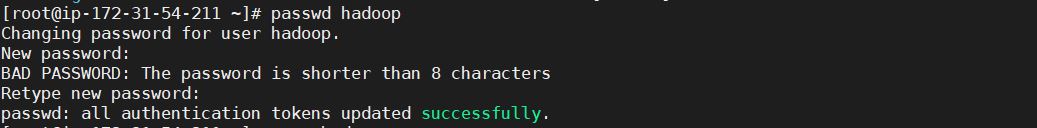
Step2: Now restart sshd service

service sshd restart

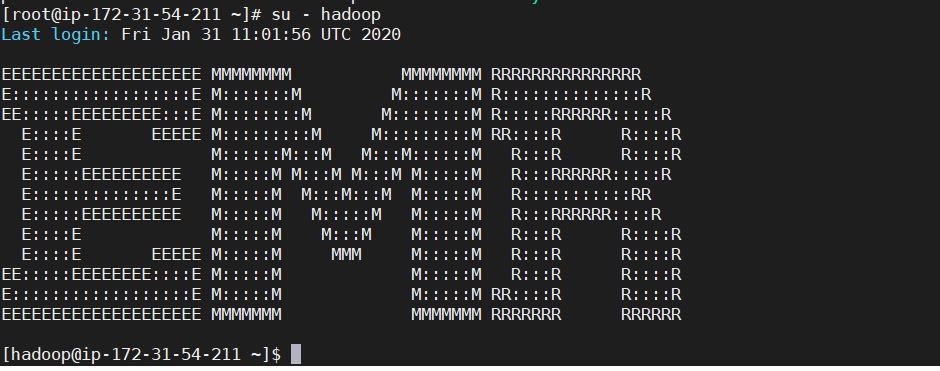


Set password for hadoop:

passwd hadoop



Switch to hadoop user



Step2:

Install Mysql on EC2 Instance using below steps

wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm

yum localinstall mysql57-community-release-el7-11.noarch.rpm -y

yum install mysql-community-server -y

systemctl start mysqld.service

We get password from below file:

cat /var/log/mysqld.log

ALTER USER 'root'@'localhost' IDENTIFIED BY 'Naresh#240';

To provide remote permissions:

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'localhost' IDENTIFIED BY 'Naresh#240' WITH GRANT OPTION;

FLUSH PRIVILEGES;

Create database:

USE mysql;

Create Table:

create table employee( empId int, empName varchar(40), sal int);

Insert Data into Table:

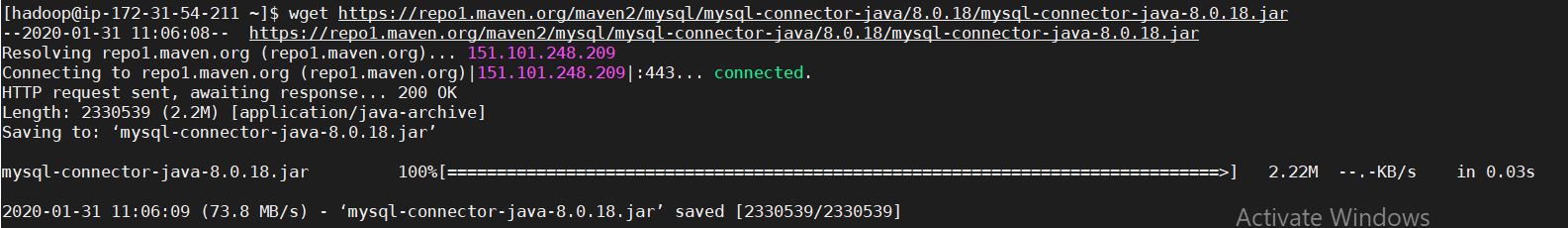
insert into employee values(101, "Naresh", 1000);

insert into employee values(102, "Suresh", 2000);

Step3:

Download mysql-connector-java-8.0.18.jar file using below link

wget <https://repo1.maven.org/maven2/mysql/mysql-connector-java/8.0.18/mysql-connector-java-8.0.18.jar>



Step4:

Create a Pyspark File to convert Mysql data to Parquet

vi SparkReadFromMysql.py

from pyspark.sql import SparkSession

spark = SparkSession\

.builder\

.appName("Write Mysql Data as Parquet")\

.getOrCreate()

dataframe\_mysql = spark.read\

.format("jdbc")\

.option("url", "jdbc:mysql:// 18.234.48.21:3306/mysql")\

.option("driver", "com.mysql.cj.jdbc.Driver")\

.option("dbtable", "employee").option("user", "naresh")\

.option("password", "Naresh#240").load()

print(dataframe\_mysql.columns)

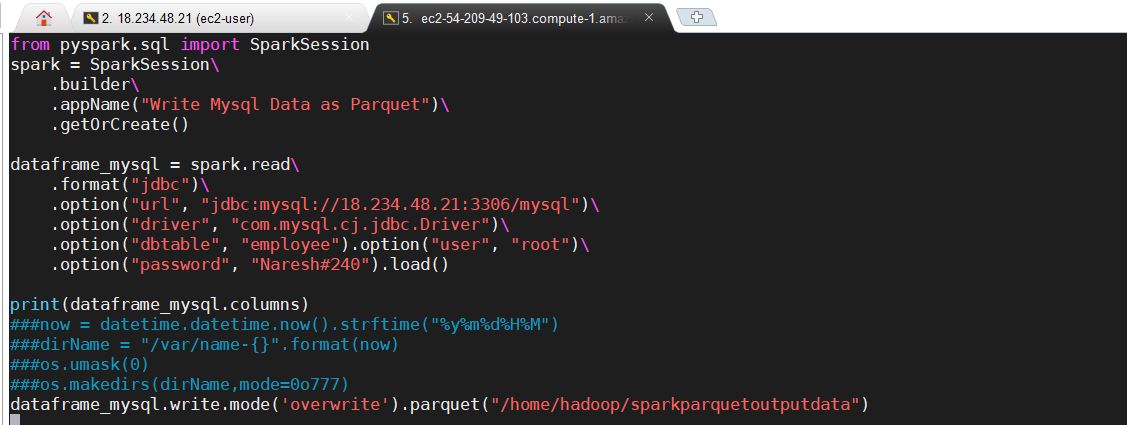
###now = datetime.datetime.now().strftime("%y%m%d%H%M")

###dirName = "/var/name-{}".format(now)

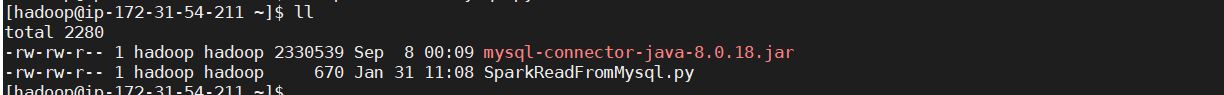
###os.umask(0)

###os.makedirs(dirName,mode=0o777)

dataframe\_mysql.write.mode('overwrite').parquet("/home/hadoop/sparkparquetoutputdata")

****

Check created files:



Open EMR Cluster and Run below commands:

spark-submit --master yarn --jars=/home/hadoop/mysql-connector-java-8.0.18.jar --conf spark.driver.extraClassPath=/home/hadoop/mysql-connector-java-8.0.18.jar --conf spark.executor.extraClassPath=/home/hadoop/mysql-connector-java-8.0.18.jar --conf spark.executor.memory=4g --conf spark.executor.cores=2 --conf spark.executor.instances=12 --conf spark.sql.warehouse.dir=/home/hadoop/ SparkReadFromMysql.py

Now check whether the file Saved or not:

hadoop fs -ls /home/hadoop/sparkparquetoutputdata

Open Hive and create table with Parquet file:

hive --service beeline

!connect jdbc:hive2://ec2-52-91-118-214.compute-1.amazonaws.com:10000/default

CREATE EXTERNAL TABLE IF NOT EXISTS employee(

empId integer,

empName string,

sal integer)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS parquet

LOCATION 'hdfs://ip-172-31-60-218.ec2.internal:8020/home/hadoop/sparkparquetoutputdata/';

select \* from employee;