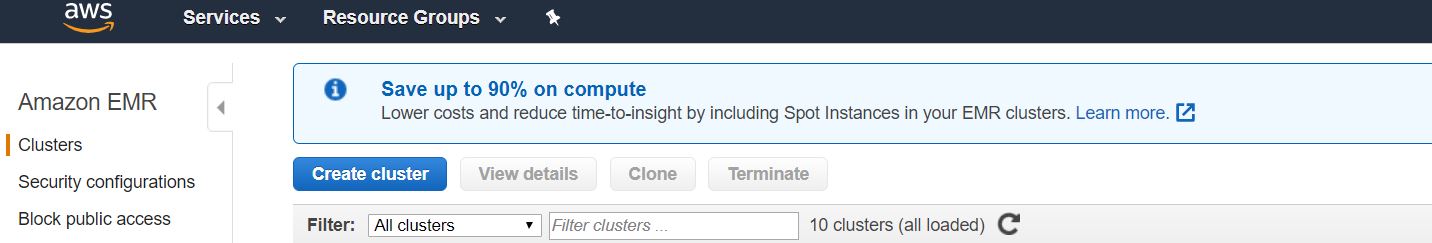
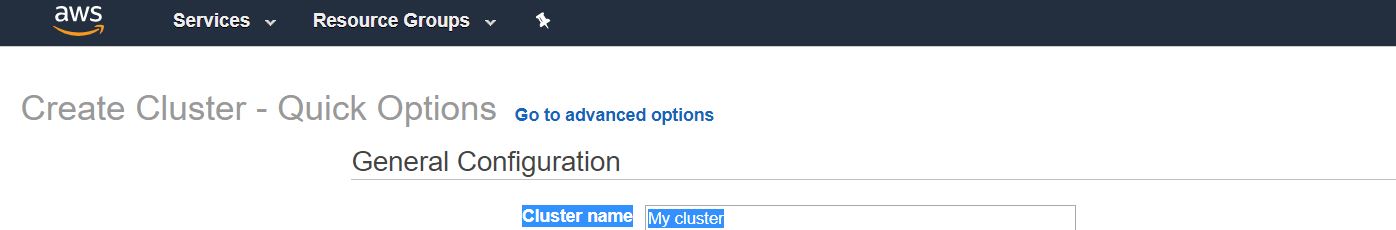
**EMR Cluster and Load CSV file data into Hive tables**

EMR Cluster Creation:

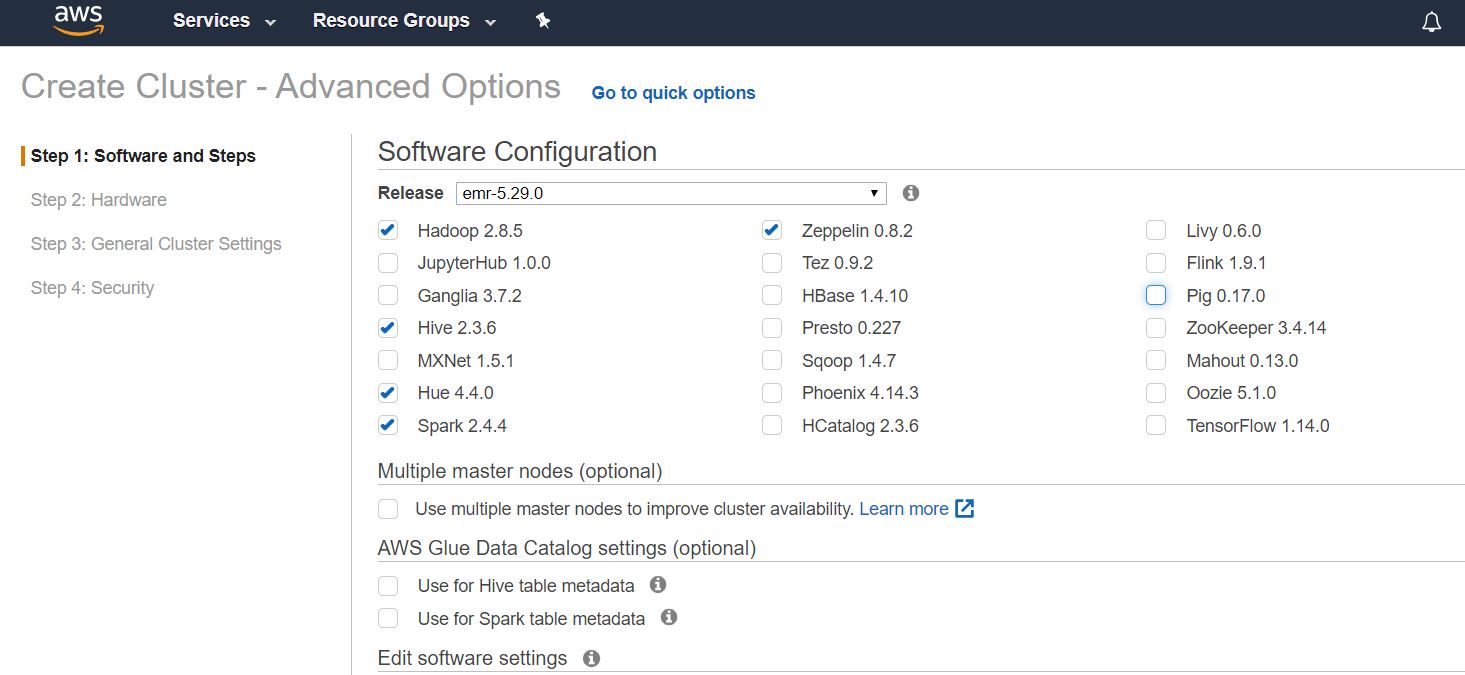
Open EMR Cluster UI in AWS and click on **create cluster**

****

Click on goto advance options and click on Next

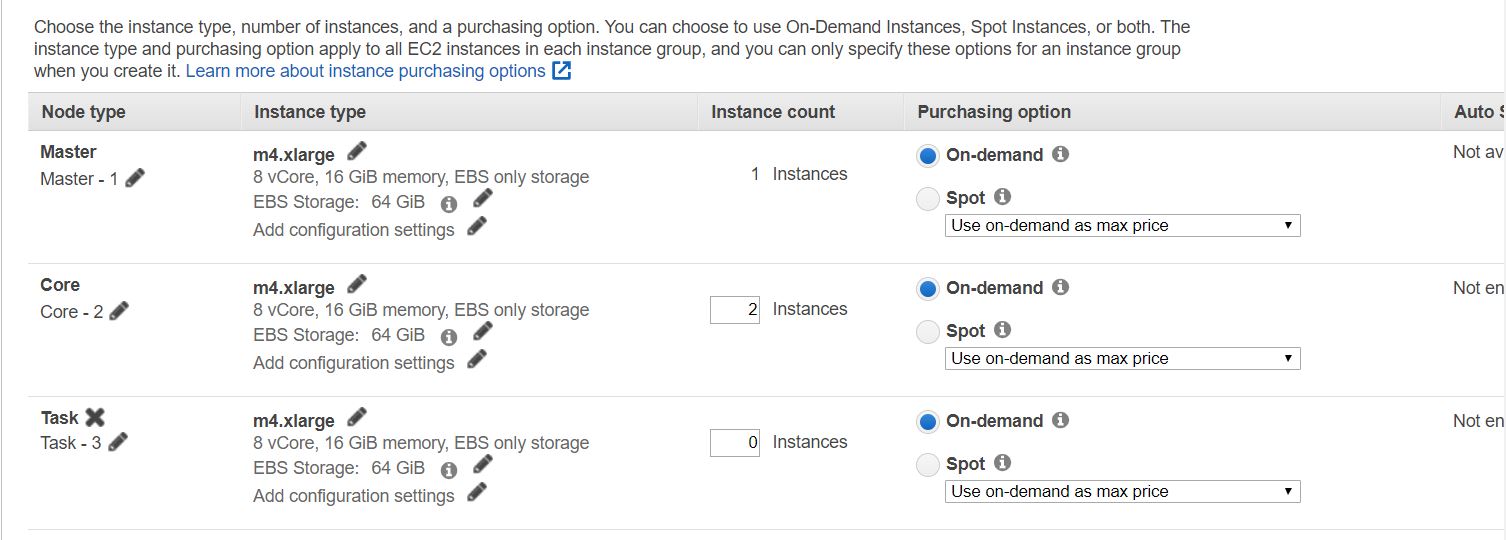
****

Select which I selected in below Image and Click on Next

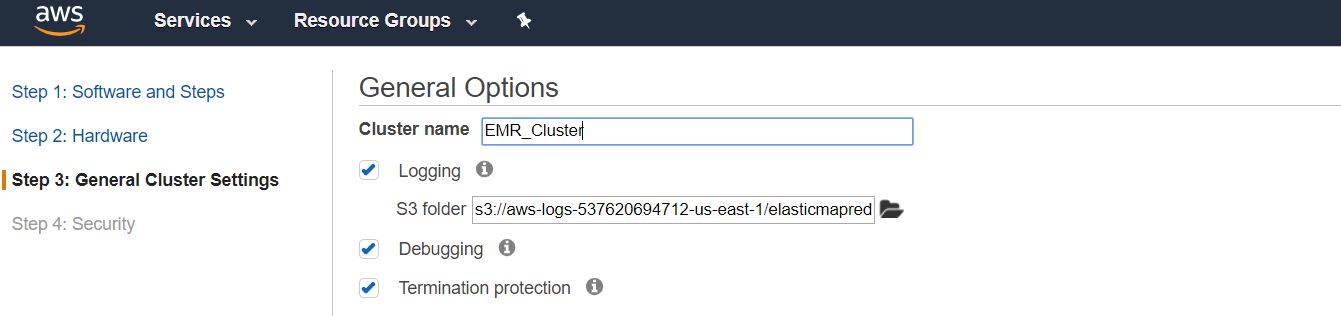
****

Here Instance type gives as m5.xlarge as default

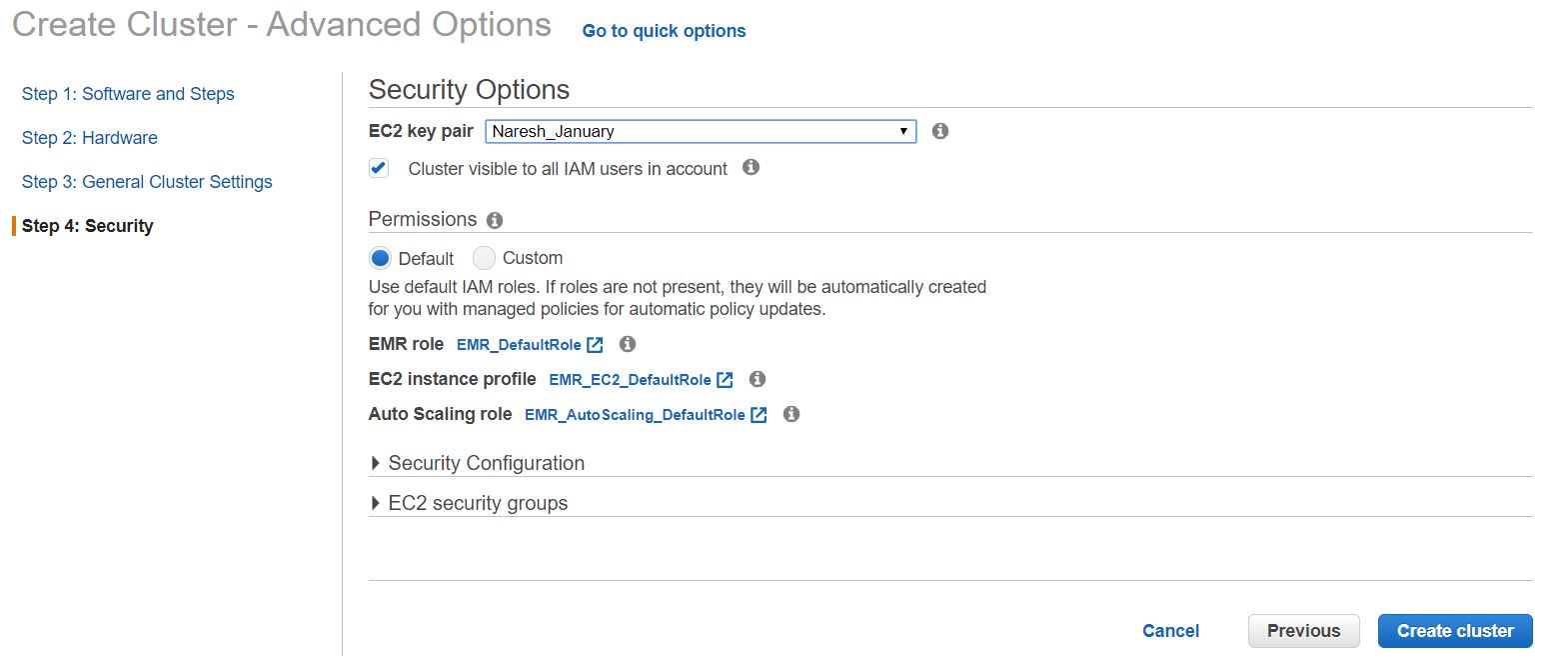
Try to create cluster with m5.xlarge, if you get any error you just try to create with m4.xlarge

****

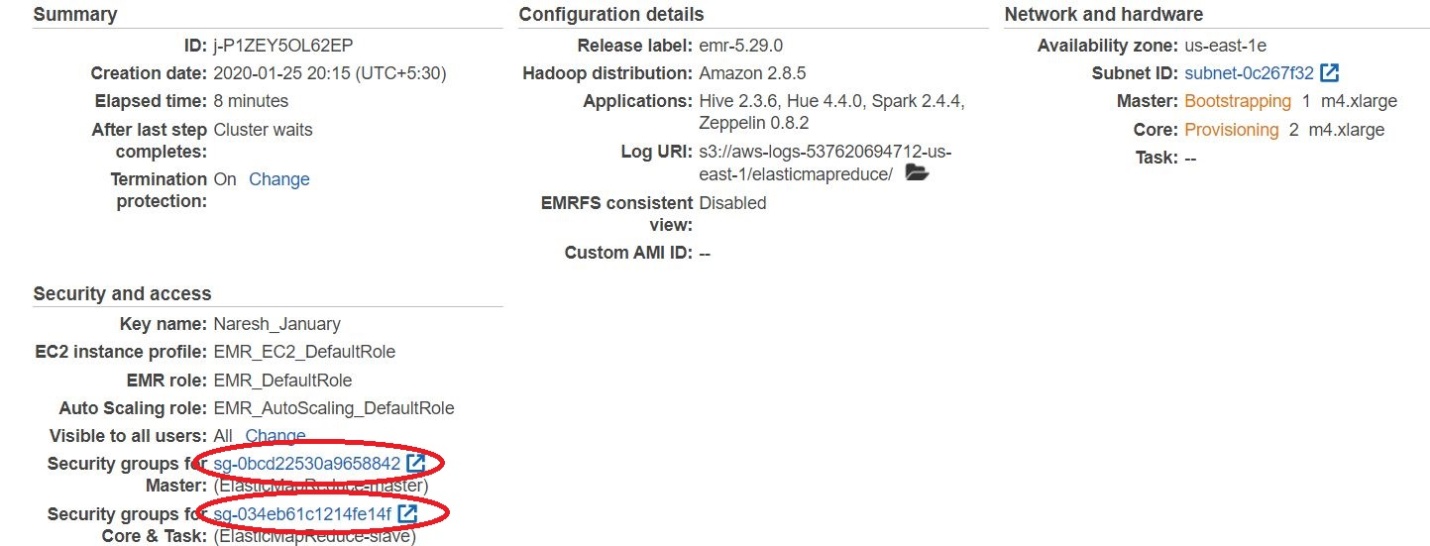
Give name for Cluster and click on Next

****

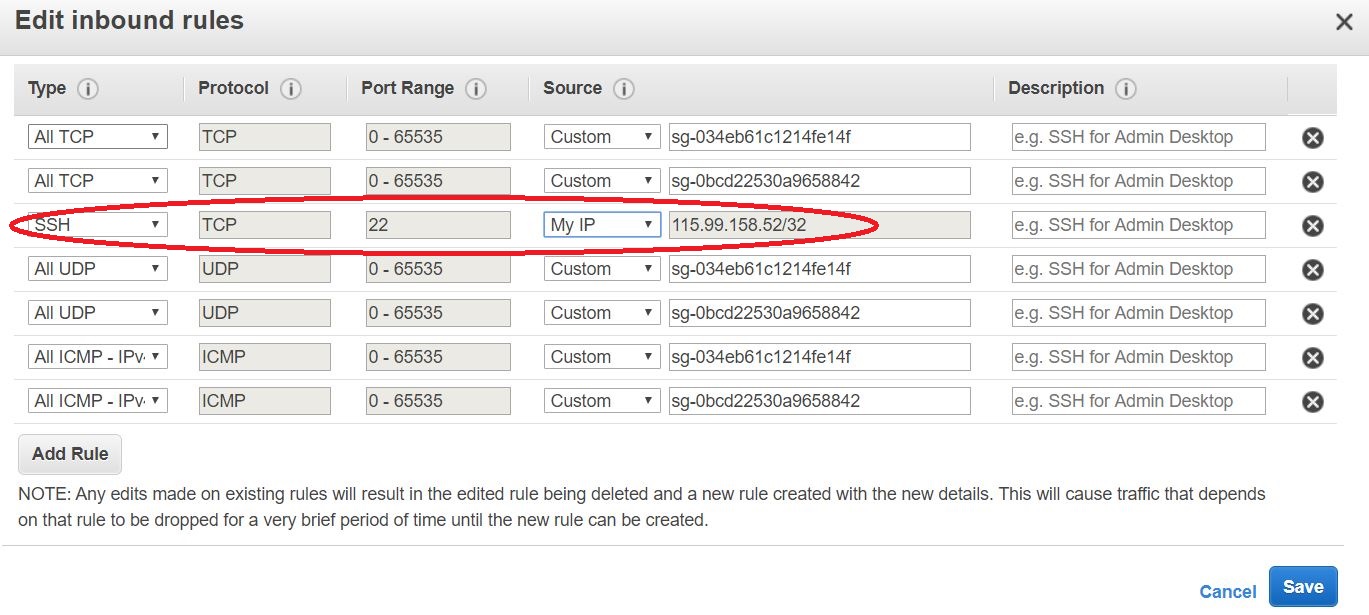
Give EC2 key pair which you in you system and click on Create cluster

****

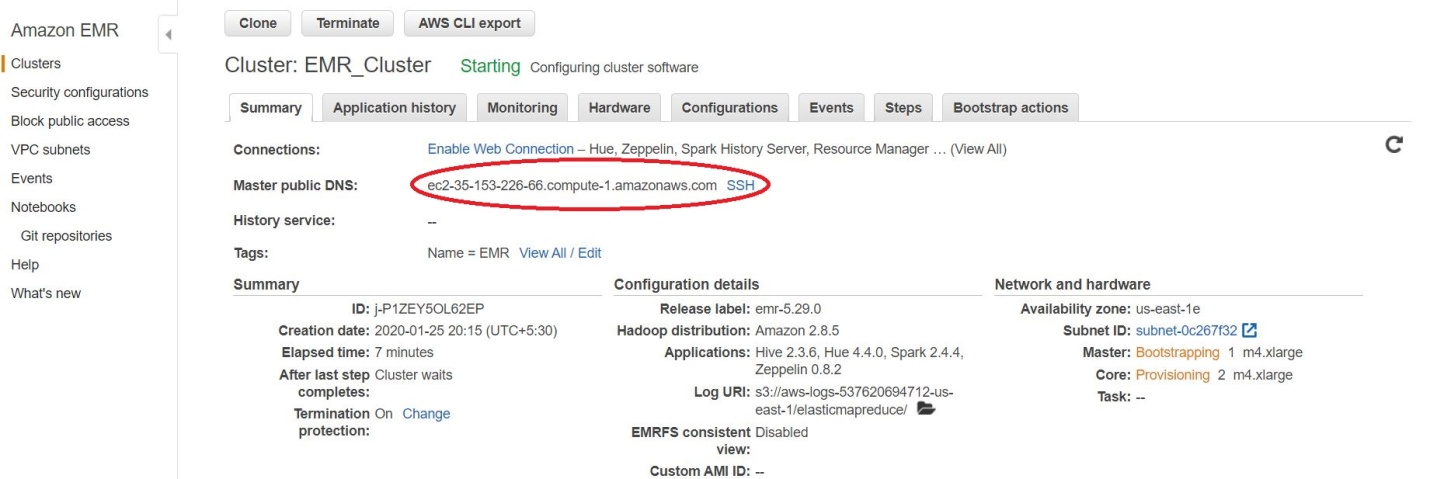
Wait until to create EMR Cluster, Mean while just open security group which we used for our cluster and add SSH with our IP address



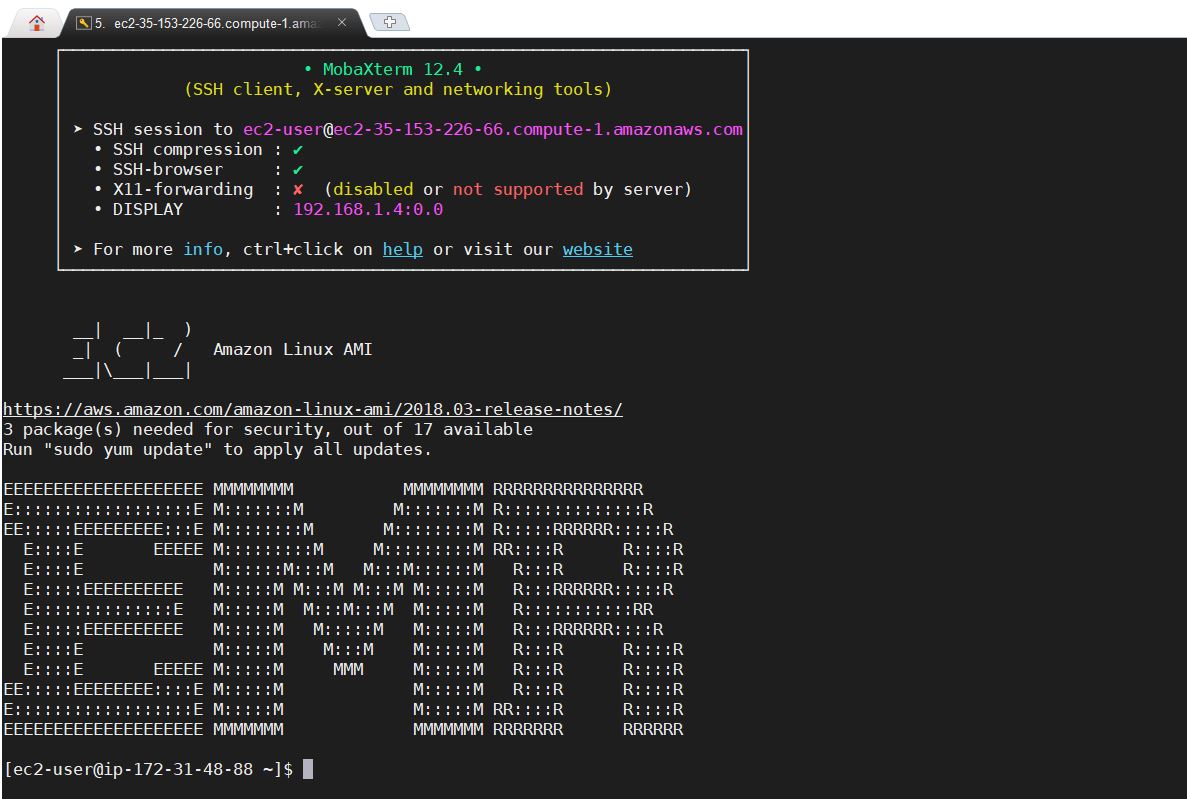
Click on Security groups and allow SSH



Go and Check whether our cluster is ready or not



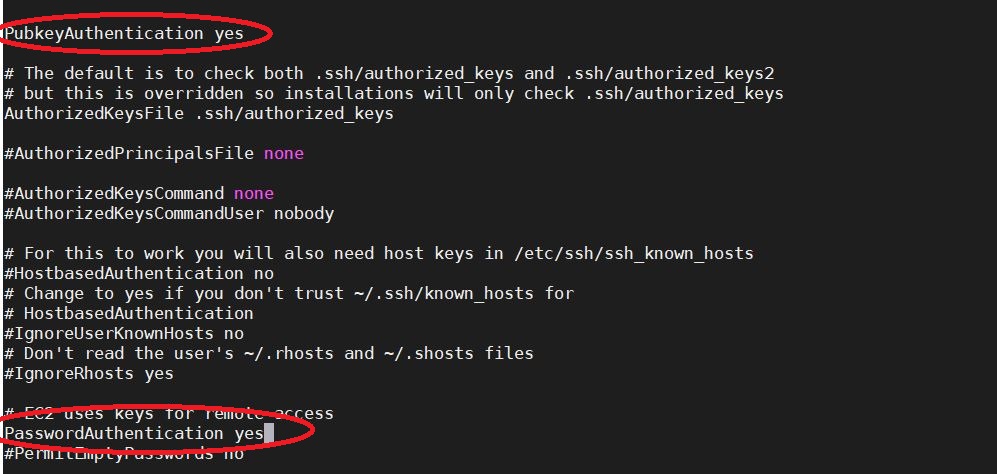
Open MobaXterm and connect with Master public DNS



Goto root user and set password for hadoop user

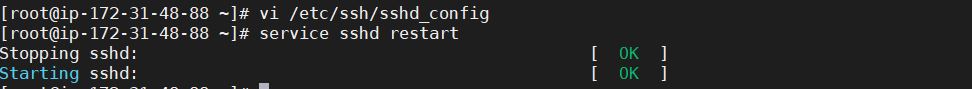
Step1: Open sshd\_config file and uncomment PubkeyAuthentication, enable password authentication

vi /etc/ssh/sshd\_config



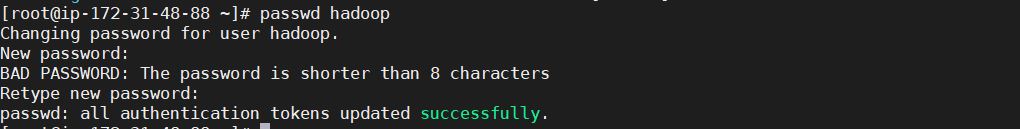
Step2: Now restart sshd service

service sshd restart

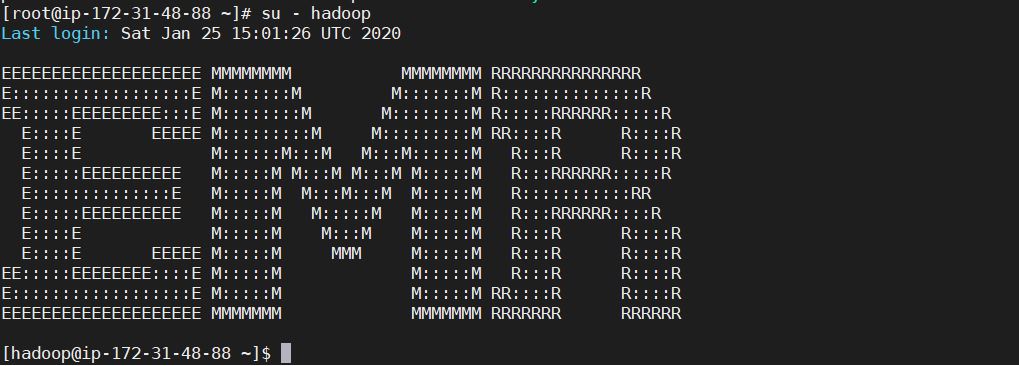


Set password for hadoop:

passwd hadoop



Switch to hadoop user



**Load CSV file into HIVE:**

Create employee.csv with some data

vi employee.csv

Create Directory in hadoop with the name of **sparkinputdata**

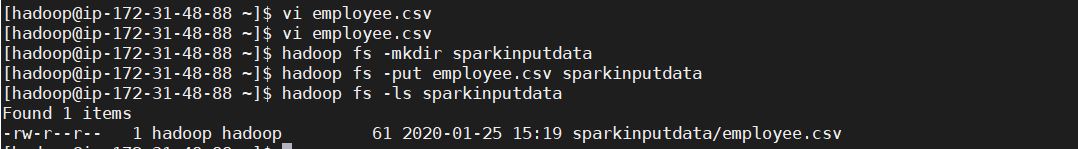
hadoop fs -mkdir sparkinputdata

Copy employee.csv file into sparkinputdata

hadoop fs -put employee.csv sparkinputdata

Check whether the file copied into sparkinputdata or not

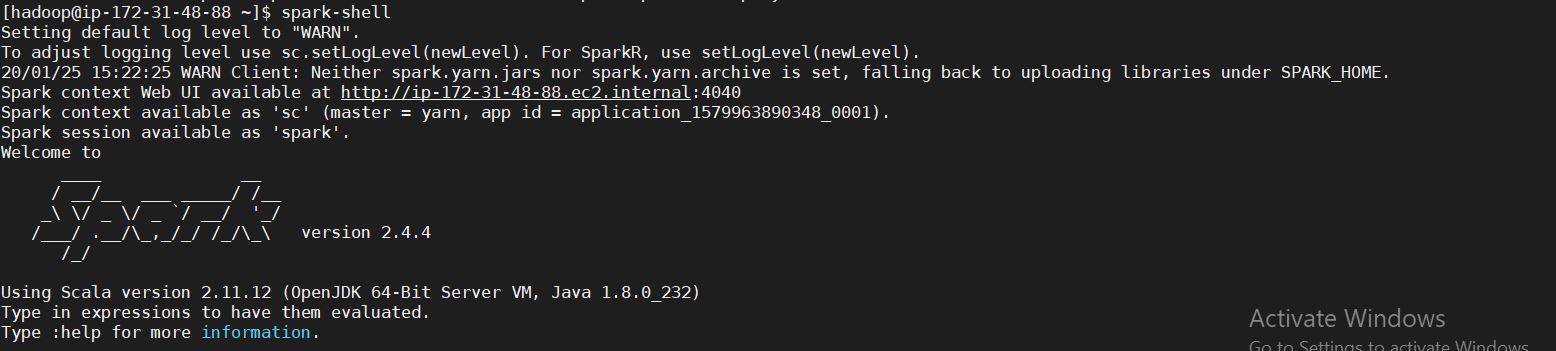
hadoop fs -ls sparkinputdata



To Convert CSV file into parquet file:

Connect to Spark

spark-shell



To read from CSV file:

val employeedata=spark.read.option("header", "true").csv("sparkinputdata/employee.csv")

To check spark variable data:

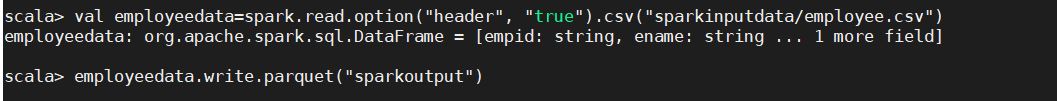
employeedata.show()

employeedata.printSchema



To write data into parquet:

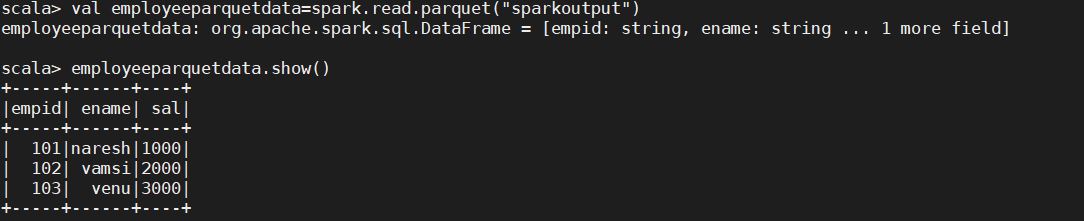
employeedata.write.parquet("sparkoutput")



To check parquet file data:

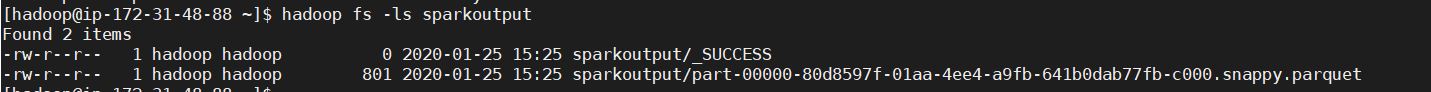
val employeeparquetdata=spark.read.parquet("sparkoutput")

employeeparquetdata.show()



Connect to hadoop user and check whether the file changed into parquet:

hadoop fs -ls sparkoutput



Connect to HIVE to load parquet data into HIVE tables:

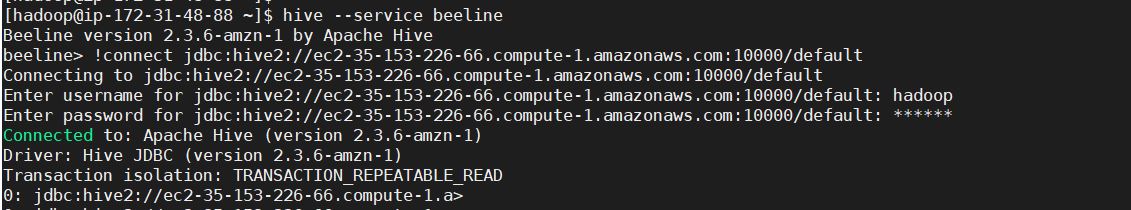
Connect to HIVE with **hadoop** user:

hive --service beeline

!connect jdbc:hive2://ec2-35-153-226-66.compute-1.amazonaws.com:10000/default

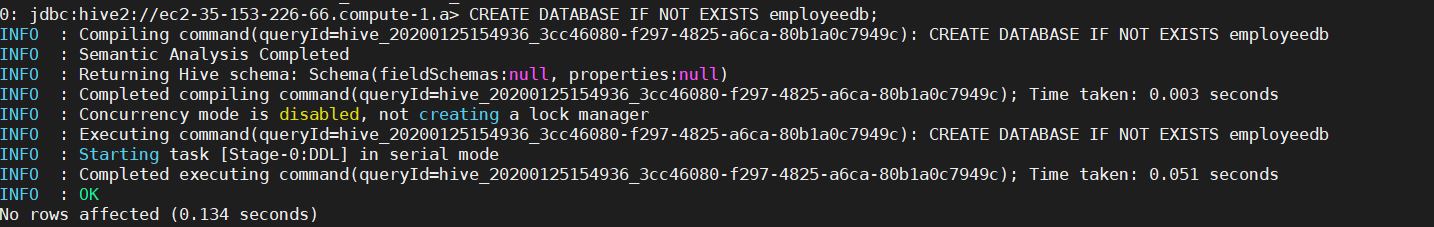
Syntax:

!conncet jdbc:hive2://<Master public DNS of the EMR cluster>:10000/default



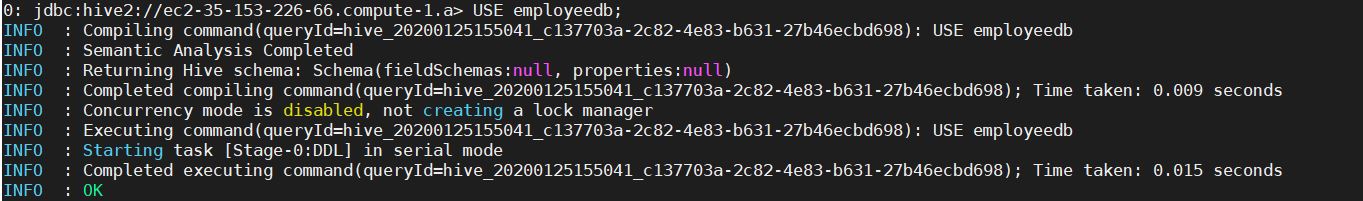
Create database to create table:

CREATE DATABASE IF NOT EXISTS employeedb;



Connect **employeedb**:

USE employeedb;

****

Create table with the name of **employee**:

CREATE EXTERNAL TABLE IF NOT EXISTS employee(

empid string,

ename string,

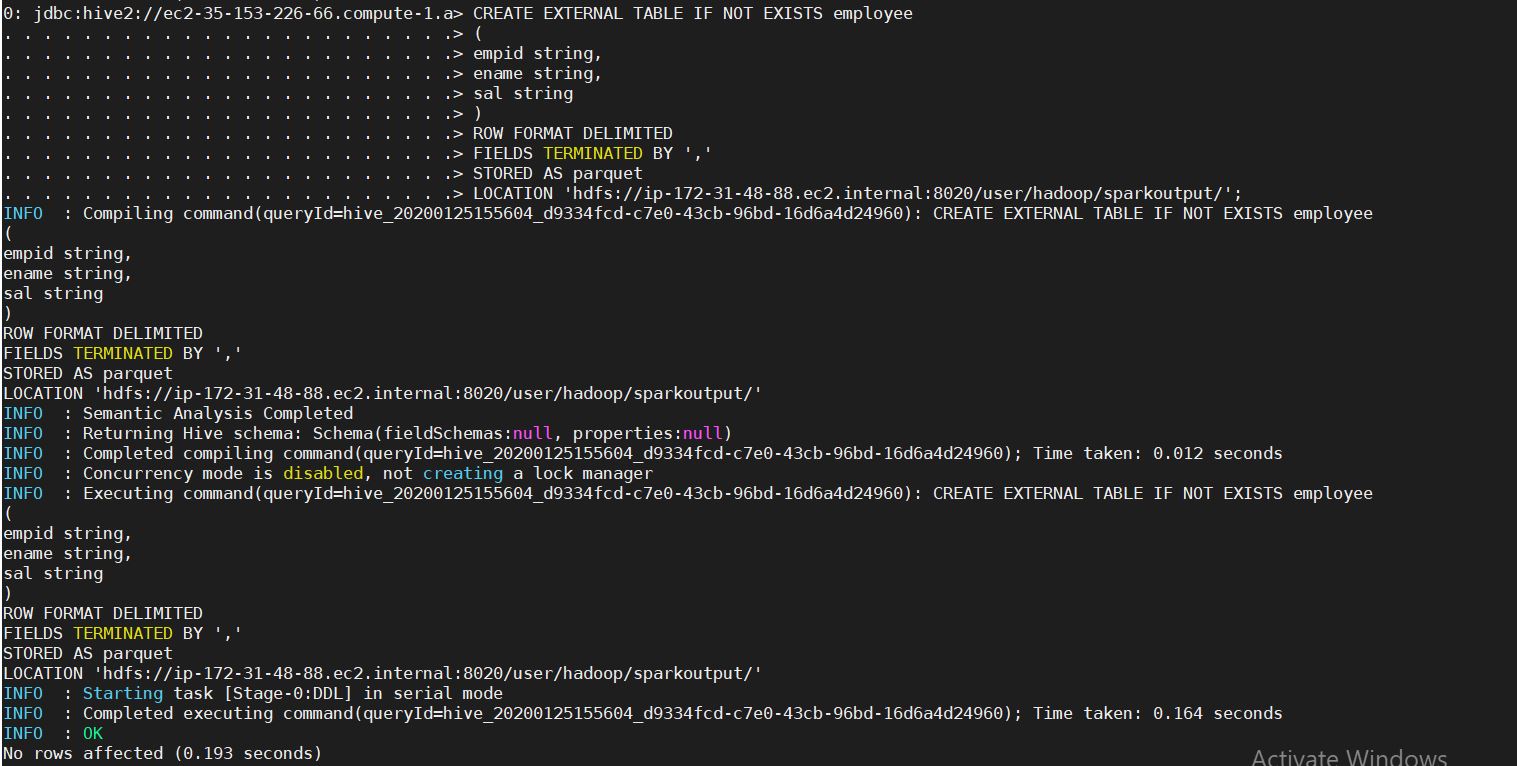
sal string)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

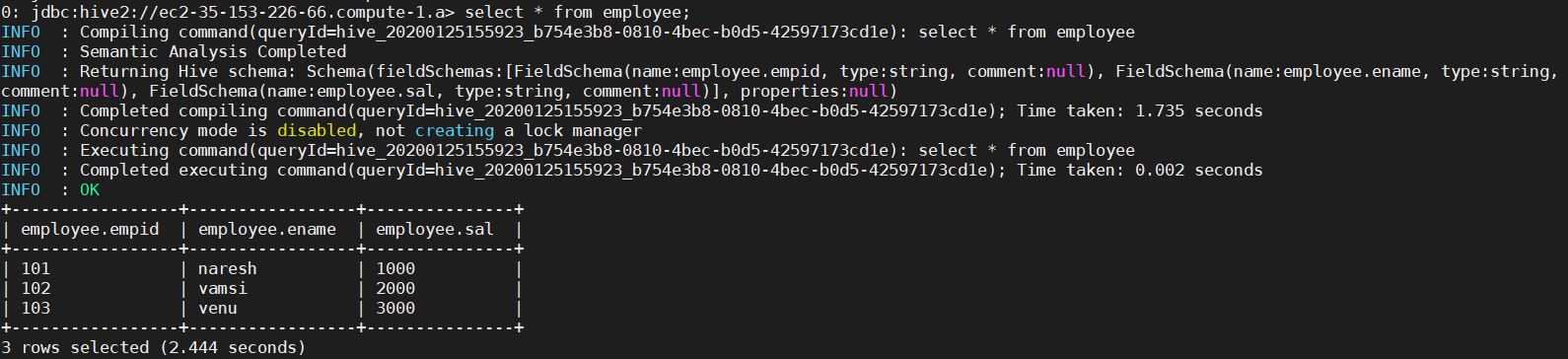
STORED AS parquet

LOCATION 'hdfs://ip-172-31-48-88.ec2.internal:8020/user/hadoop/sparkoutput/';



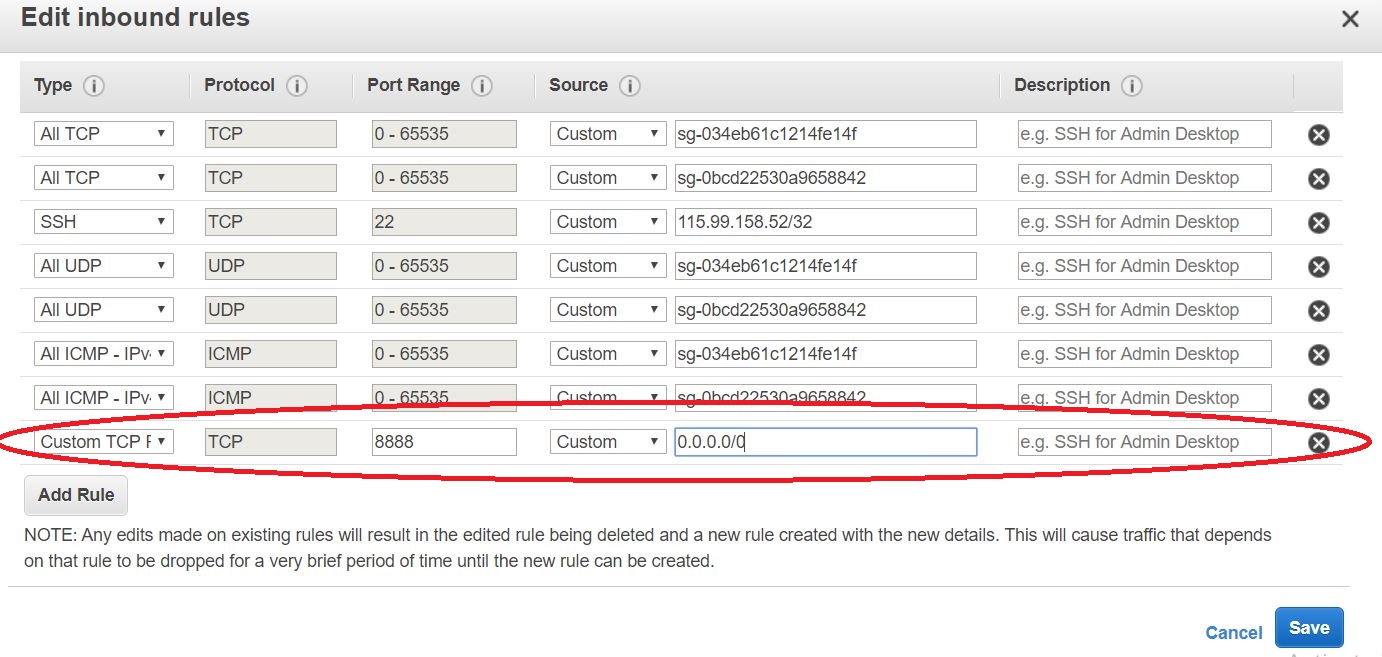
Check Data whether data loaded into employee table or not:

select \* from employee;



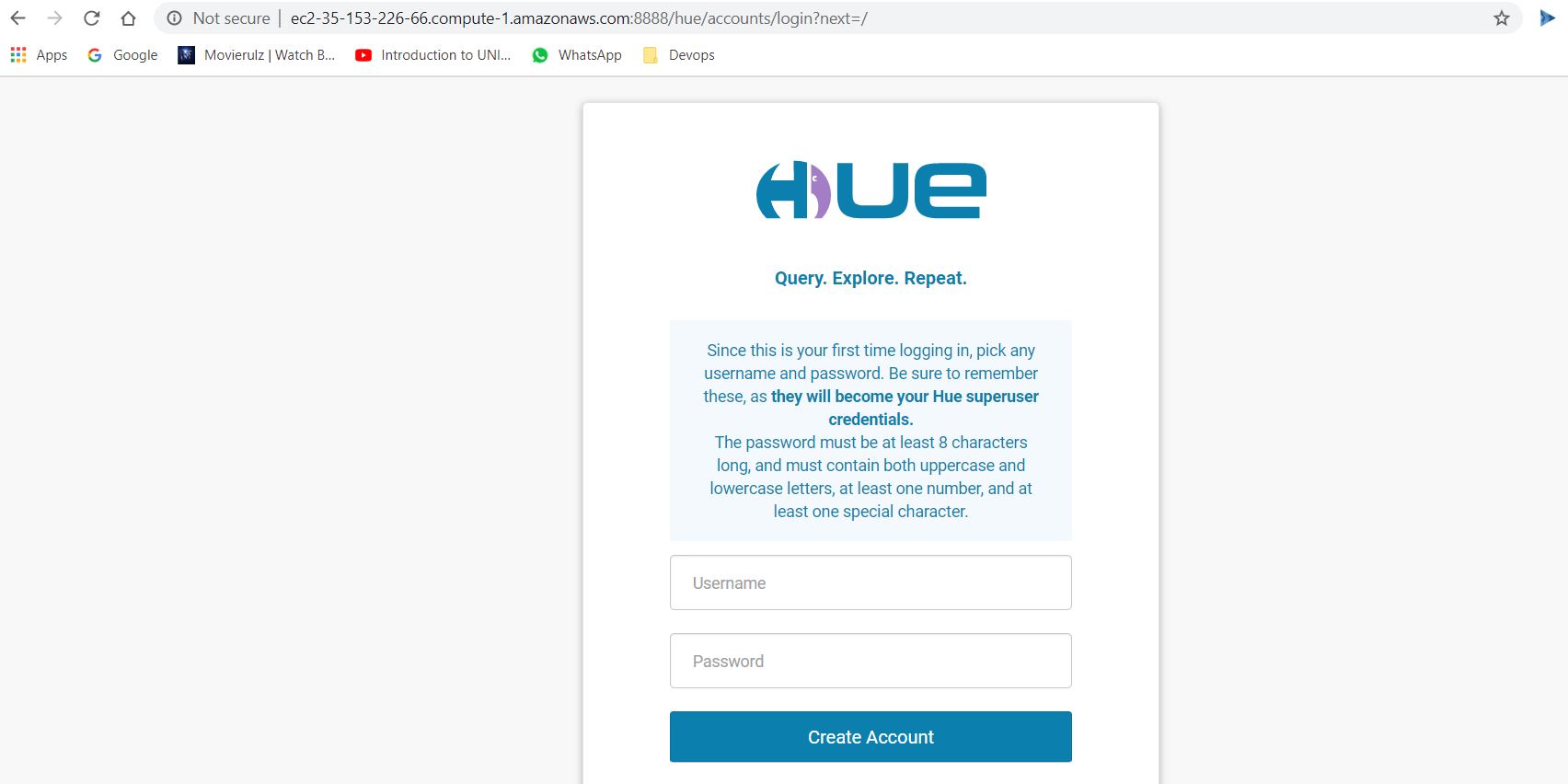
Connect to HUE in UI:

To Conncet HUE in UI we need allow 8888 port number in security group which we used in cluster

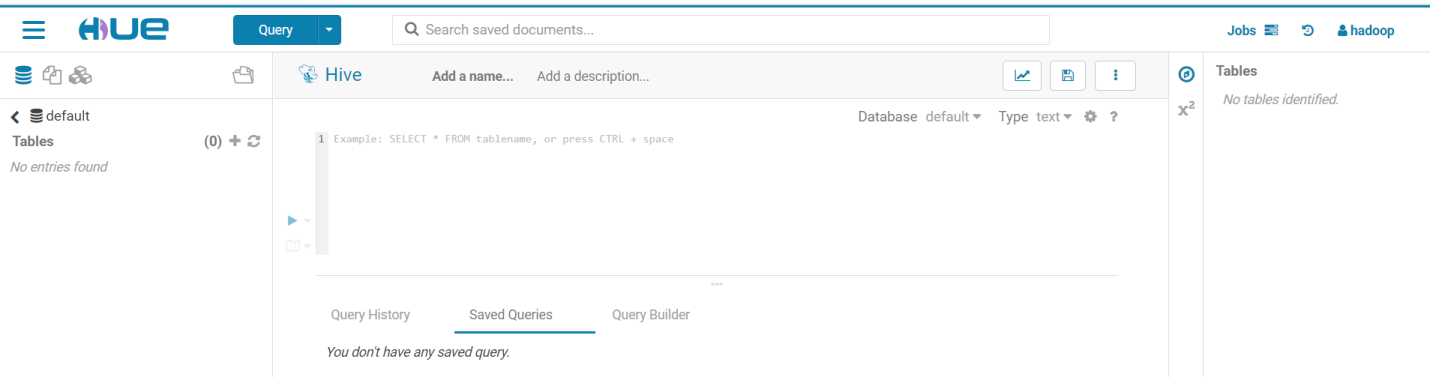


Give DNS name of the cluster in UI with port number:8888

ec2-35-153-226-66.compute-1.amazonaws.com:8888



Create Account with user name and password



Here we can do all operations which we did in HIVE. Its user friendly and easy to do all.