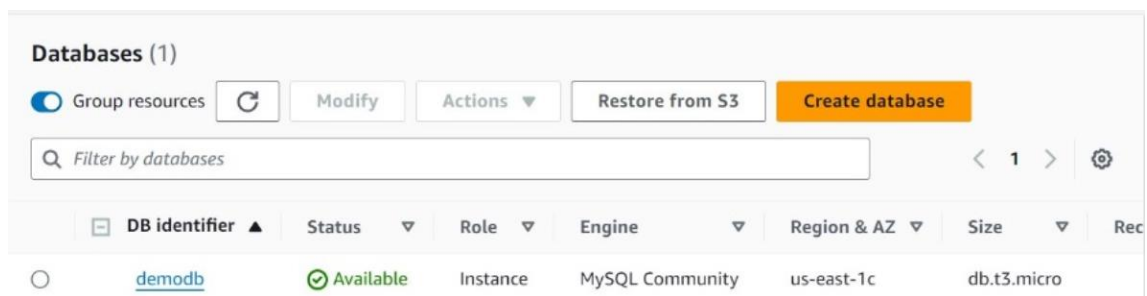


## TASK 2

Use Sqoop command to ingest the data from RDS into the HBase Table.

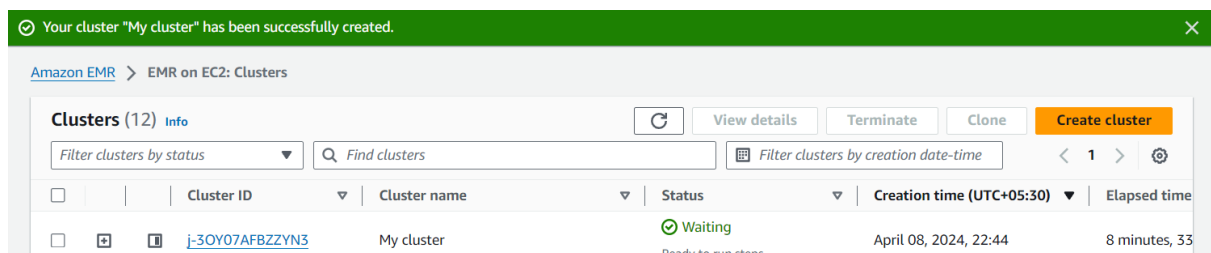
**Step 1:** Creating a RDS instance.

- Search for RDS on Amazon Web Server.
- Click on create database under database menu.
- Further choose “Standard create” from the database creation method.
- Next click on MySQL from Engine Options to run a MySQL engine.
- Click on Free tier under the Templates option to use the engine for free.
- Now, give a name to your database under DB cluster identifier.
- Next, give a public access to your database by clicking on yes under the Public Access option.
- Ensure to make the database password authentication under database authentication for security.
- At last, click on Create Database to create the RDS Database.
- The database will take 5-10 minutes to get active.



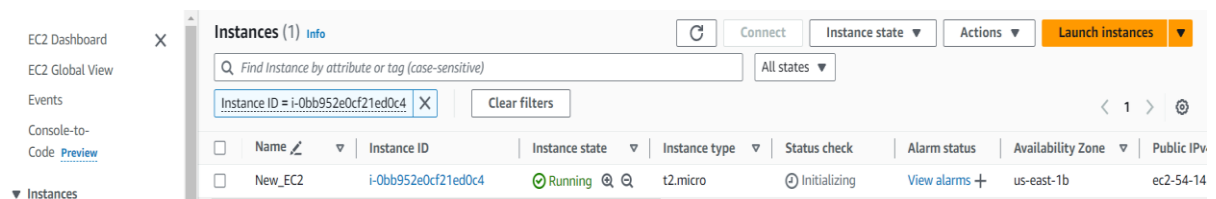
**Step 2:** Creating an EMR cluster.

- Search for EMR on Amazon Web Server.
- Click on “create cluster” to create a EMR cluster
- Now, give name to your cluster under the name column.
- Next, tick the required applications under the application bundle for example Hadoop, HBase, Sqoop etc.
- Further choose an EC2 instance type for Primary node.
- Next enter the Size required (30 GiB) under EBS root volume.
- Now, select the EC2 key pair for SSH to the cluster by creating a new key pair.
- Now at last, select the service role and instance profile as default role
- Finally, click on Create Cluster to create an EMR Cluster
- The cluster will take 5-10 minutes to get active



### Step 3: Creating an EC2 instance

- Search for EC2 on Amazon Web Server
- Click on Launch instances to create a EC2 instance
- Now, give a name to your EC2 instance under the Name and Tags option
- Now, Select a desired Amazon Machine Image for example, Amazon Linux, macOS etc.
- Now select the instance type which is required for example t2.micro with 1 vCPU and 1GiB memory
- Now, select a key pair name from the drop-down menu or create a new key pair
- At last, click on Launch Instance to create a EC2 Instance
- The creation of the instance may take around 5-10 minutes to get active



#### Step 4. Connecting EMR instance with the local computer using Putty.

- Firstly, copy the Primary node public DNS from the EMR cluster and paste it as a host name or IP Address in Putty
- Now, click on credentials under the Auth menu from SSH and browse for private key file for authentication.
- After inserting the private key, click open and the connection between EMR instance with the local computer will be executed.

[illegible]

**Step 5:** Execute below command

```
sudo yum update
```

```
sudo yum install python3 python3-pip
```

```
python3 --version
```

```
pip3 --version
```

```
pip3 install thriftpy2
```

**Step 6:** Execute the below command on Hadoop

```
sqoop import --connect jdbc:mysql://demodb.crog2ckcif6b.us-east-1.rds.amazonaws.com/taxi_records --username admin --password admin123 --table trip_log --hbase-table trip_log_hbase --column-family cf1 --hbase-create-table --hbase-row-key tpep_pickup_datetime,tpep_dropoff_datetime --hbase-bulkload --split-by payment_type
```

**Step 7:** Execute below python script in root

```
vi task3.py
```

```
-----Python script-----
```

```
import happybase
```

```
# create connection
```

```
connection = happybase.Connection('localhost', port=9090, autoconnect=False)
```

```
# open connection to perform operations
```

```
def open_connection():
```

```
    connection.open()
```

```
# close opened connection
```

```
def close_connection():
```

```
    connection.close()
```

```
# get the pointer to a table
```

```
def get_table(name):
```

```
    open_connection()
```

```

table = connection.table(name)

close_connection()

return table

def batch_insert_data(filename, tablename):
    print("starting batch insert of "+filename)
    file = open(filename, 'r')
    table = get_table(tablename)
    open_connection()
    i = 0
    with table.batch(batch_size=50000) as b:
        for line in file:
            if i!=0:
                temp = line.strip().split(",")

                b.put(temp[1]+temp[2], {'cf1:VendorID': str(temp[0]), 'cf1:tpep_pickup_datetime':
str(temp[1]), 'cf1:tpep_dropoff_datetime': str(temp[2]), 'cf1:passenger_count': str(temp[3]),
'cf1:trip_distance': str(temp[4]), 'cf1:RatecodeID': str(temp[5]), 'cf1:store_and_fwd_flag':
str(temp[6]), 'cf1:PULocationID': str(temp[7]), 'cf1:DOLocationID': str(temp[8]), 'cf1:payment_type':
str(temp[9]), 'cf1:fare_amount': str(temp[10]), 'cf1:extra': str(temp[11]), 'cf1:mta_tax': str(temp[12]),
'cf1:tip_amount': str(temp[13]), 'cf1:tolls_amount': str(temp[14]), 'cf1:improvement_surcharge':
str(temp[15]), 'cf1:total_amount': str(temp[16]), 'cf1:congestion_surcharge': str(temp[17]),
'cf1:airport_fee': str(temp[18]) })

            i+=1

    file.close()
    print("batch insert done")
    close_connection()

batch_insert_data('yellow_tripdata_2017-01.csv', 'trip_log')
batch_insert_data('yellow_tripdata_2017-02.csv', 'trip_log')

```

-----Python script-----

python task3.py

**Step 8:** After this, Execute below commands

```
sudo -i
```

```
cd hbase
```

```
hbase shell
```

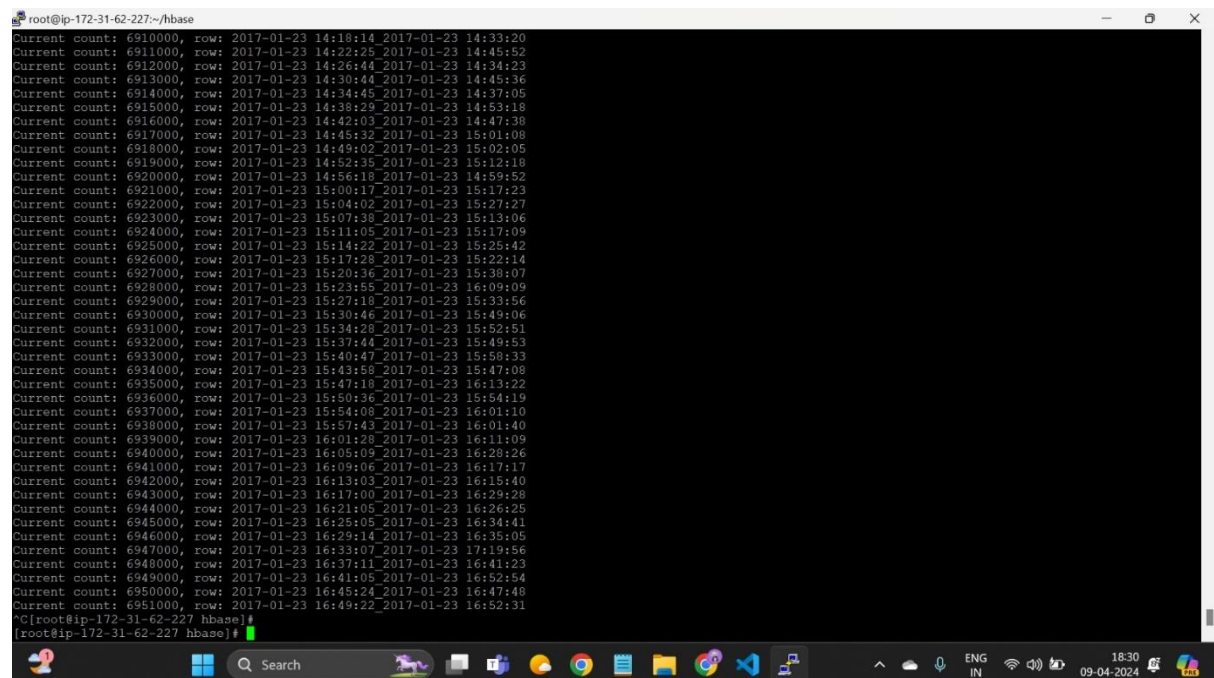
```
list
```

```
describe 'trip_log_hbase'
```

```
count 'trip_log_hbase'
```

Resulting in,

Ingesting the data from RDS into the HBase Table.



The screenshot shows a terminal window with the title bar "root@ip-172-31-62-227:~/hbase". The terminal displays the output of the HBase shell commands. The "list" command shows a single table "trip\_log\_hbase". The "describe 'trip\_log\_hbase'" command shows the table's schema with columns "row" and "time". The "count 'trip\_log\_hbase'" command shows the total count of rows in the table, which is 6951000. The terminal output is as follows:

```
root@ip-172-31-62-227:~/hbase
hbase> list
Current count: 6910000, row: 2017-01-23 14:18:14 2017-01-23 14:33:20
Current count: 6911000, row: 2017-01-23 14:22:25 2017-01-23 14:45:52
Current count: 6912000, row: 2017-01-23 14:26:44 2017-01-23 14:34:23
Current count: 6913000, row: 2017-01-23 14:30:44 2017-01-23 14:45:36
Current count: 6914000, row: 2017-01-23 14:34:45 2017-01-23 14:37:05
Current count: 6915000, row: 2017-01-23 14:38:29 2017-01-23 14:53:18
Current count: 6916000, row: 2017-01-23 14:42:03 2017-01-23 14:47:38
Current count: 6917000, row: 2017-01-23 14:45:32 2017-01-23 15:01:08
Current count: 6918000, row: 2017-01-23 14:49:02 2017-01-23 15:02:05
Current count: 6919000, row: 2017-01-23 14:52:35 2017-01-23 15:12:18
Current count: 6920000, row: 2017-01-23 14:56:18 2017-01-23 14:59:52
Current count: 6921000, row: 2017-01-23 15:00:17 2017-01-23 15:17:23
Current count: 6922000, row: 2017-01-23 15:04:02 2017-01-23 15:27:27
Current count: 6923000, row: 2017-01-23 15:07:38 2017-01-23 15:13:06
Current count: 6924000, row: 2017-01-23 15:11:05 2017-01-23 15:17:09
Current count: 6925000, row: 2017-01-23 15:14:22 2017-01-23 15:25:42
Current count: 6926000, row: 2017-01-23 15:17:28 2017-01-23 15:22:14
Current count: 6927000, row: 2017-01-23 15:20:36 2017-01-23 15:38:07
Current count: 6928000, row: 2017-01-23 15:23:55 2017-01-23 16:09:09
Current count: 6929000, row: 2017-01-23 15:27:18 2017-01-23 15:33:56
Current count: 6930000, row: 2017-01-23 15:30:46 2017-01-23 15:49:06
Current count: 6931000, row: 2017-01-23 15:34:28 2017-01-23 15:52:51
Current count: 6932000, row: 2017-01-23 15:37:44 2017-01-23 15:49:53
Current count: 6933000, row: 2017-01-23 15:40:47 2017-01-23 15:58:33
Current count: 6934000, row: 2017-01-23 15:43:58 2017-01-23 15:47:08
Current count: 6935000, row: 2017-01-23 15:47:18 2017-01-23 16:13:22
Current count: 6936000, row: 2017-01-23 15:50:36 2017-01-23 15:54:19
Current count: 6937000, row: 2017-01-23 15:54:08 2017-01-23 16:01:10
Current count: 6938000, row: 2017-01-23 15:57:43 2017-01-23 16:01:40
Current count: 6939000, row: 2017-01-23 16:01:28 2017-01-23 16:11:09
Current count: 6940000, row: 2017-01-23 16:05:09 2017-01-23 16:28:26
Current count: 6941000, row: 2017-01-23 16:09:06 2017-01-23 16:17:17
Current count: 6942000, row: 2017-01-23 16:13:03 2017-01-23 16:15:40
Current count: 6943000, row: 2017-01-23 16:17:00 2017-01-23 16:29:28
Current count: 6944000, row: 2017-01-23 16:21:05 2017-01-23 16:26:25
Current count: 6945000, row: 2017-01-23 16:25:05 2017-01-23 16:34:41
Current count: 6946000, row: 2017-01-23 16:29:14 2017-01-23 16:35:05
Current count: 6947000, row: 2017-01-23 16:33:07 2017-01-23 17:19:56
Current count: 6948000, row: 2017-01-23 16:37:11 2017-01-23 16:41:23
Current count: 6949000, row: 2017-01-23 16:41:05 2017-01-23 16:52:54
Current count: 6950000, row: 2017-01-23 16:45:24 2017-01-23 16:47:48
Current count: 6951000, row: 2017-01-23 16:49:22 2017-01-23 16:52:31
^C[root@ip-172-31-62-227 hbase]#
[root@ip-172-31-62-227 hbase]#
```