1- Install MongoDB on your EC2 instance

Follow the instruction on
https://docs.aws.amazon.com/zh_cn/dms/latest/sbs/CHAP_MongoDB2DocumentDB.02.html
https://docs.mongodb.com/manual/tutorial/install-mongodb-on-amazon/

Verify Linux Distribution:

```
(base) [ec2-user@ip-172-31-94-236 ~]$ grep ^NAME  /etc/*release
/etc/os-release:NAME="Amazon Linux AMI"
```

Configure the package management system (yum).:

```
(base) [ec2-user@ip-172-31-94-236 ~]$ sudo vi /etc/yum.repos.d/mongodb-org-4.4.repo
```

Input:

copy

```
[mongodb-org-4.4]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/amazon/2/mongodb-org/4.4/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-4.4.asc
```

Install the MongoDB packages.:

Update **glibc**: if you glibc is old version <= 2.17

```
(base) [ec2-user@ip-172-31-94-236 ~]$ yum check dependencies
Loaded plugins: priorities, update-motd, upgrade-helper
check ['dependencies']
(base) [ec2-user@ip-172-31-94-236 ~]$ sudo yum clean all
Loaded plugins: priorities, update-motd, upgrade-helper
Cleaning repos: amzn-main amzn-updates mongodb-org-4.4
Cleaning up everything
(base) [ec2-user@ip-172-31-94-236 ~]$ sudo yum update
Loaded plugins: priorities, update-motd, upgrade-helper
```

```
(base) [ec2-user@ip-172-31-94-236 ~]$ yum whatprovides */libc.so.6
Loaded plugins: priorities, update-motd, upgrade-helper
```

copy

```
sudo yum install -y mongodb-org
```

Follow the instructions in the remaining part: **Run MongoDB Community Edition**
And remember to **ulimit** the **Resource Utilization**

```
(base) [ec2-user@ip-172-31-94-236 ~]$ ulimit -a
```

Configure host name to public DNS:

```
sudo vi /etc/mongod.conf
```

2. By default, the MongoDB server (`mongod`) only allows loopback connections from IP address 127.0.0.1 (localhost). To allow connections from elsewhere in your Amazon VPC, do the following:

   a. Edit the `/etc/mongod.conf` file and look for the following lines.

   ```
   # network interfaces
   net:
     port: 27017
     bindIp: 127.0.0.1  # Enter 0.0.0.0,:: to bind to all IPv4 and IPv6 addresses or, alternatively, use the net.bindIpAll
   ```

   b. Modify the `bindIp` line so that it looks like the following.

   ```
   bindIp: public-dns-name
   ```

   c. Replace *public-dns-name* with the actual public DNS name for your instance, for example `ec2-11-22-33-44.us-west-2.compute.amazonaws.com`.

   d. Save the `/etc/mongod.conf` file, and then restart `mongod`.

   ```
   sudo service mongod restart
   ```

Run MongoDB:

```
mongo --host ec2-18-211-73-183.compute-1.amazonaws.com:27017
```

Next step please see the next page

2- Download GSE13355.zip (attached)

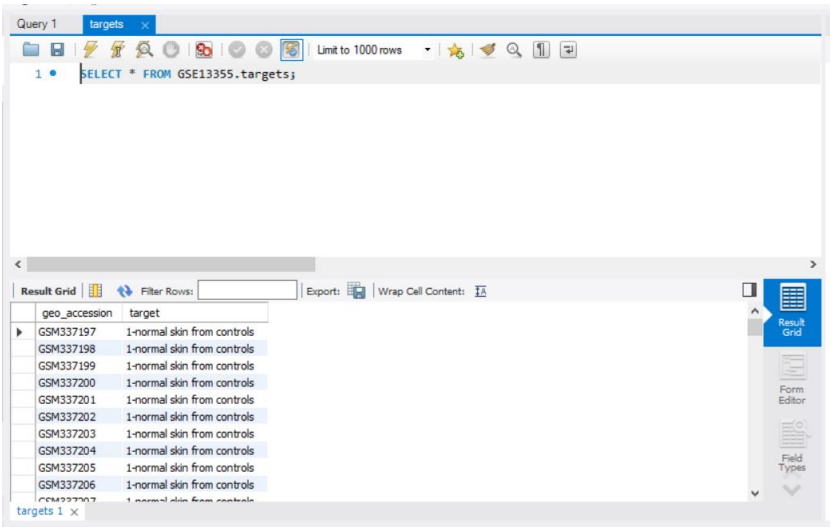
3- Create GSE13355 database on MySQL and MongoDB

4- Upload GSE13355_targets.csv to MySQL and MongoDB

MySQL:

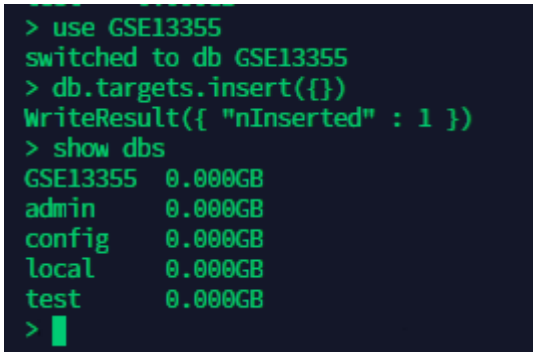
```
CREATE SCHEMA `GSE13355` ;
```

Use MySQL Workbench to create table and import data. Result:



MongoDB:

```
use GSE13355
db.targets.insert({})
```



Import data by shell:

```
mongoimport --db GSE13355 --collection targets --type csv --
headerline --file /home/ec2-user/GSE13355_targets.csv --host ec2-18-
211-73-183.compute-1.amazonaws.com:27017
```

Reference: https://blog.csdn.net/qq_32447321/article/details/79223332

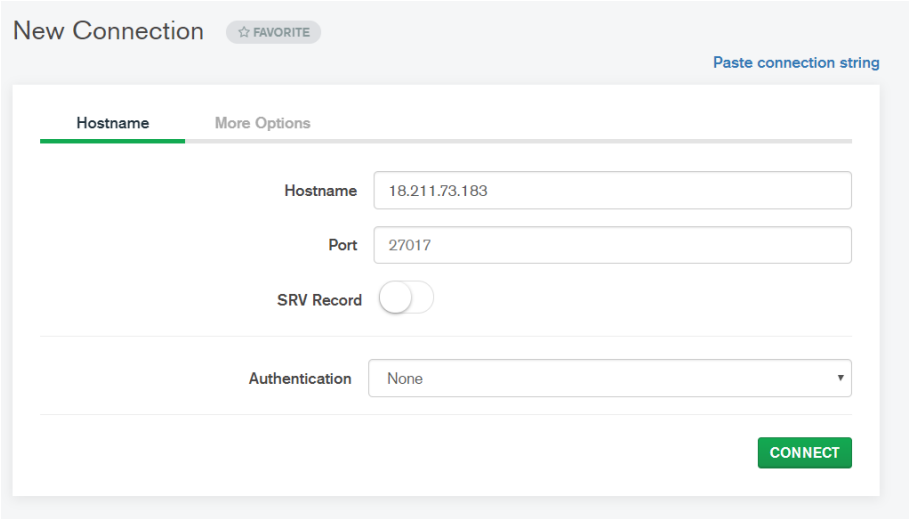
Valid import data using UI interface: **MongoDB Compass**

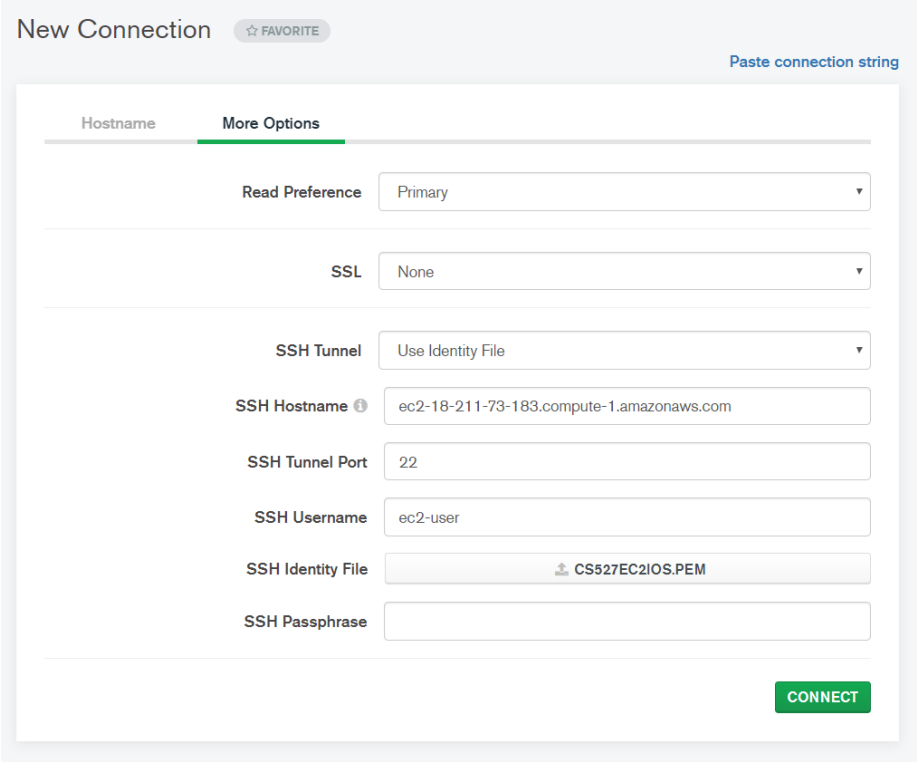
Firstly config the MongoDB server:

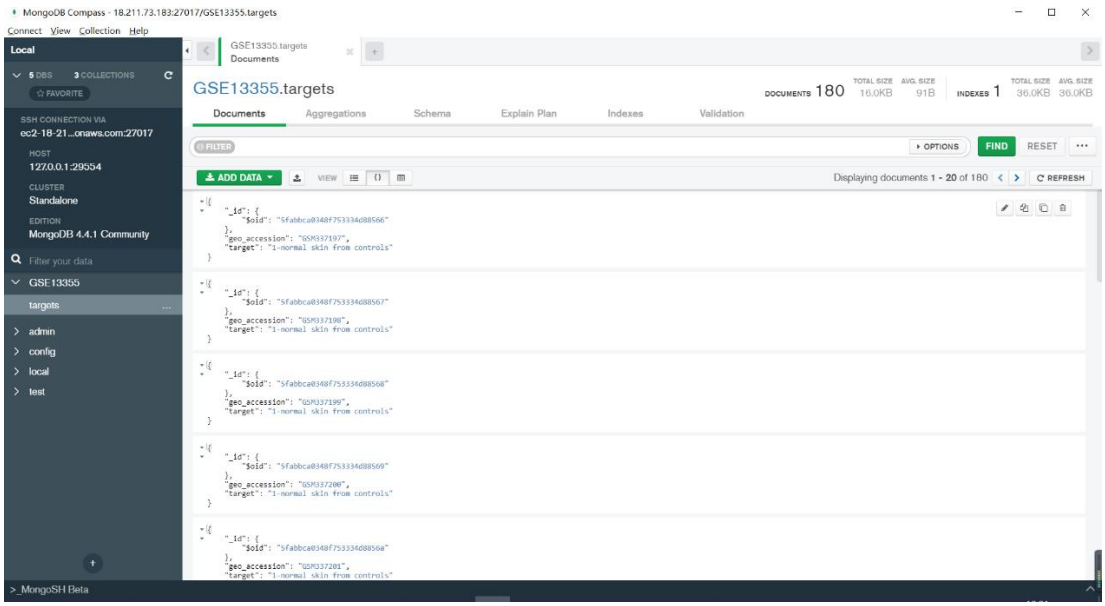
```
sudo vi /etc/mongod.conf
```

```
# network interfaces
net:
  port: 27017
  bindIp: ec2-18-211-73-183.compute-1.amazonaws.com  # Enter 0.0.0.0,:: to bind to all IPv4 and IPv6 addresses or, alternatively, use the net.bindIpAll setting.
```

Then create connection in MongoDB Compass:






MongoDB Compass - 18.211.73.183:27017/GSE13355.targets
Connect View Collection Help
Local
5 DBS 3 COLLECTIONS
FAVORITE
SSH CONNECTION VIA
ec2-18-21...onaws.com:27017
HOST
127.0.0.1:29554
CLUSTER
Standalone
EDITION
MongoDB 4.4.1 Community
Filter your data
GSE13355
targets
admin
config
local
test
GSE13355.targets
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
DOCUMENTS 180
INDEXES 1
FILTER
OPTIONS
FIND
RESET
ADD DATA
VIEW
Displaying documents 1 - 20 of 180
REFRESH
"_id": { "$oid": "5fabbca0348f753334d88566" },
"geo_accession": "GSM337197",
"target": "1-normal skin from controls"
"_id": { "$oid": "5fabbca0348f753334d88567" },
"geo_accession": "GSM337198",
"target": "1-normal skin from controls"
"_id": { "$oid": "5fabbca0348f753334d88568" },
"geo_accession": "GSM337199",
"target": "1-normal skin from controls"
"_id": { "$oid": "5fabbca0348f753334d88569" },
"geo_accession": "GSM337200",
"target": "1-normal skin from controls"
"_id": { "$oid": "5fabbca0348f753334d8856a" },
"geo_accession": "GSM337201",
"target": "1-normal skin from controls"
>_MongoSH Beta

5- Transpose GSE13355_expr.csv and upload the transposed file to MySQL and MongoDB

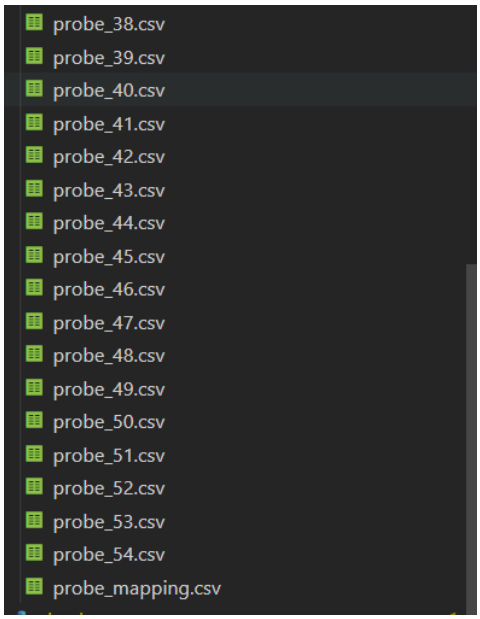
MySQL:

Use **transpose.py** to transpose the csv file and get raw data.

transpose.py

```python
import pandas as pd

file=open('C:\\Users\\95236\\Desktop\\MongoDB_assignment_Group5\\GSE13355_expr_transpose.csv','w')

df = pd.read_csv('C:\\Users\\95236\\Desktop\\MongoDB_assignment_Group5\\GSE13355_expr.csv',header= None)
data = df.values
# data = df.as_matrix()
data = list(map(list,zip(*data)))
data = pd.DataFrame(data)
data.to_csv(file, header=0, index=0)
```

Split the csv into multiple tables by **split_table.py** each has 1001 columns, since the max number of columns in MySQL is 4096, that one is the id related to the **geo_info** like GSM337197 and others are **probe_name** and the mapping between the **probe_name** and **table_name** is stored in table **probe_mapping**. And write the **geo_info** into the table **expr** with **geo_id** and **geo_name**

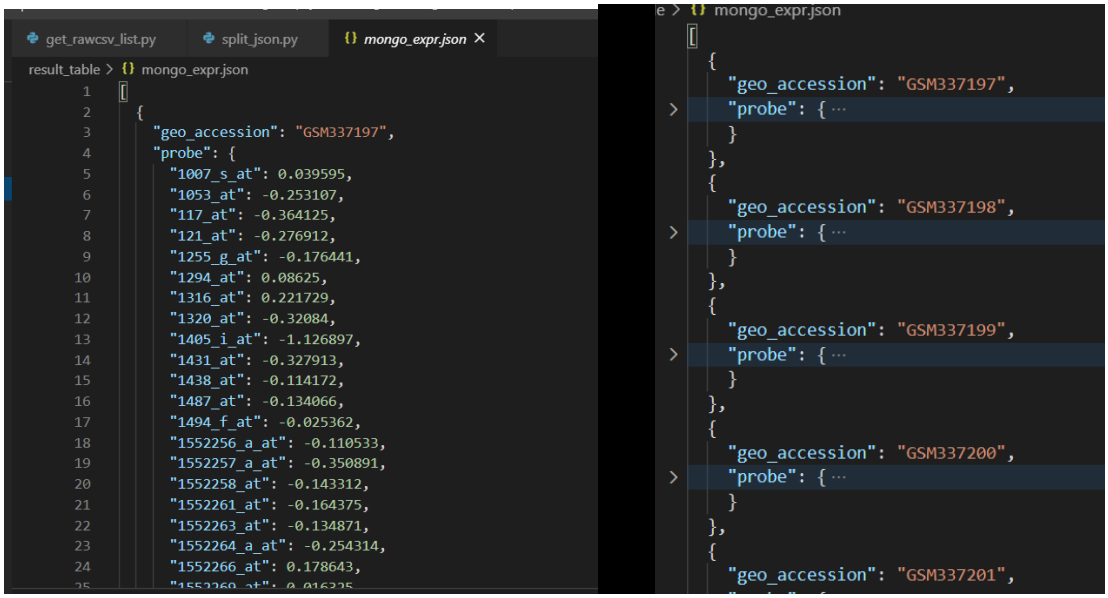


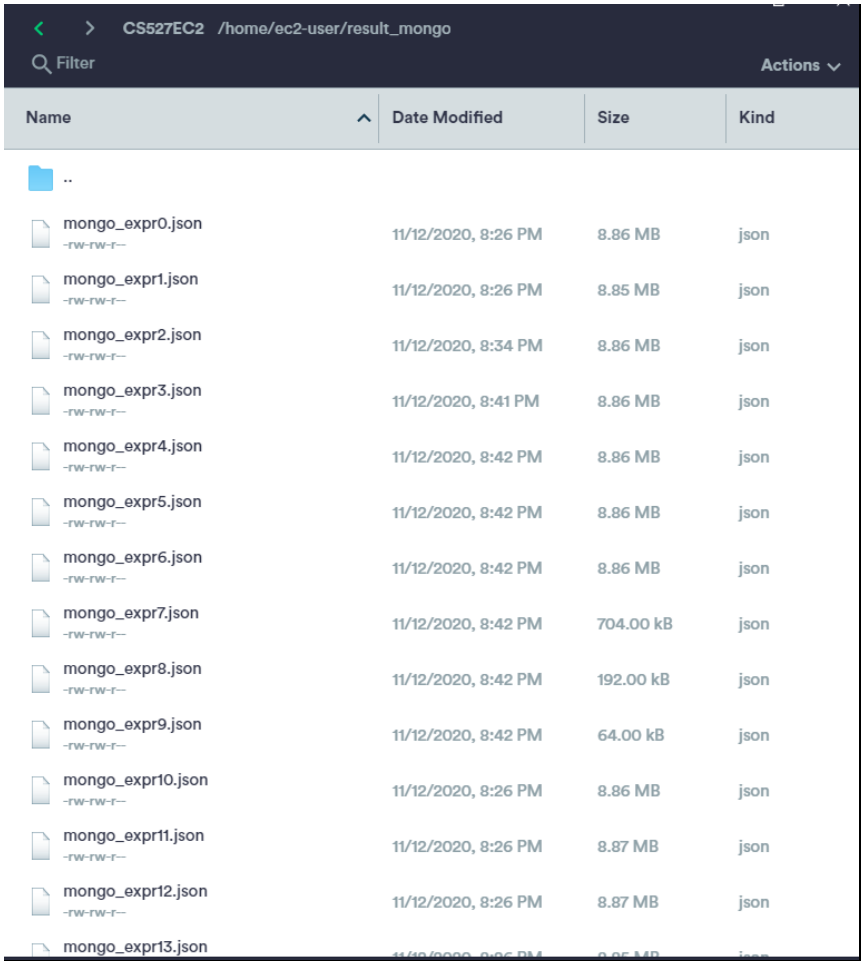
then use the **load** method to import the data from ec2 to rds:

```
mysql> CREATE TABLE IF NOT EXISTS `GSE13355`.`probe_mapping`(
    ->     `probe_name` VARCHAR(500) NOT NULL COMMENT 'column name',
    ->     `table_name` VARCHAR(200) NOT NULL,
    ->     PRIMARY KEY (`probe_name`),
    ->     UNIQUE INDEX (`probe_name`)
    ->     )ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Query OK, 0 rows affected (0.05 sec)

mysql>
mysql> LOAD DATA LOCAL INFILE '/home/ec2-user/result_table/probe_mapping.csv' INTO TABLE `GSE13355`.`probe_mapping` FIELDS TERMINATED BY ',' ENCLOSED BY '|';
Query OK, 54675 rows affected (1.33 sec)
Records: 54675  Deleted: 0  Skipped: 0  Warnings: 0

mysql>
```

MongoDB:

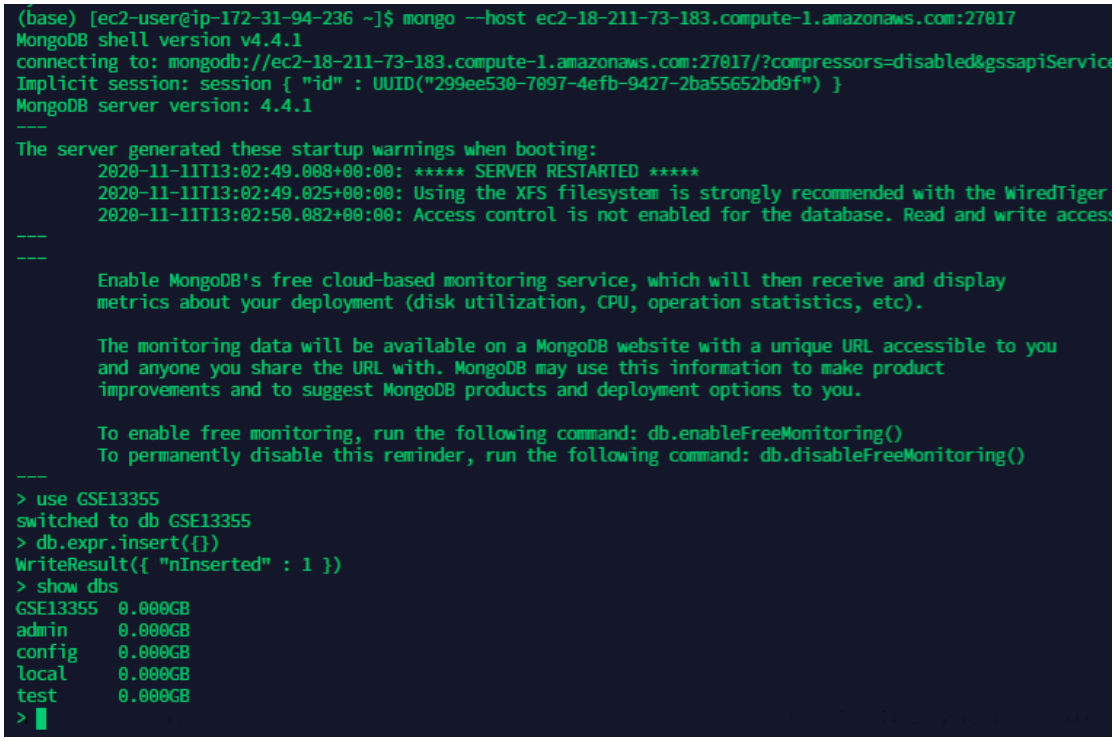
We use **split_json.py** to split the data to the json file as the following format, because the json result is too large and in order to import more convenient, we split the json file to 36 json files and each of it is less than 10 MB so that we can use **mongoimport –jsonArray** to import the data:





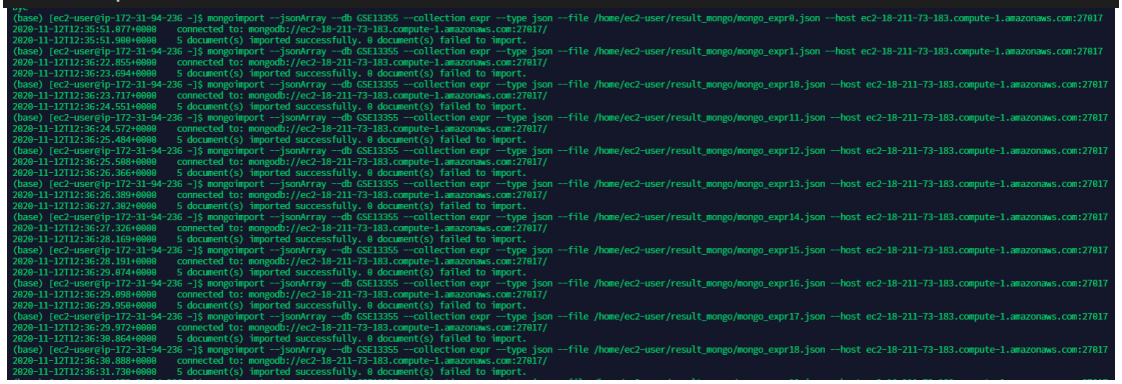
And then as the above method, use the **mongoimport** to import the json data into MongoDB:

```
mongo --host ec2-18-211-73-183.compute-1.amazonaws.com:27017
use GSE13355
db.targets.insert({})
```



Example:

```
mongoimport --jsonArray --db GSE13355 --collection expr --type json --
file /home/ec2-user/result_mongo/mongo_expr1.json --host ec2-18-211-73-
183.compute-1.amazonaws.com:27017
```



6- Up to 5 teams will be selected to present their assignment