

# MSDS 7346

## Cloud Computing

### Mini Project 5 – MongoDB & End-to-end AWS

Name: Mooyoung Lee

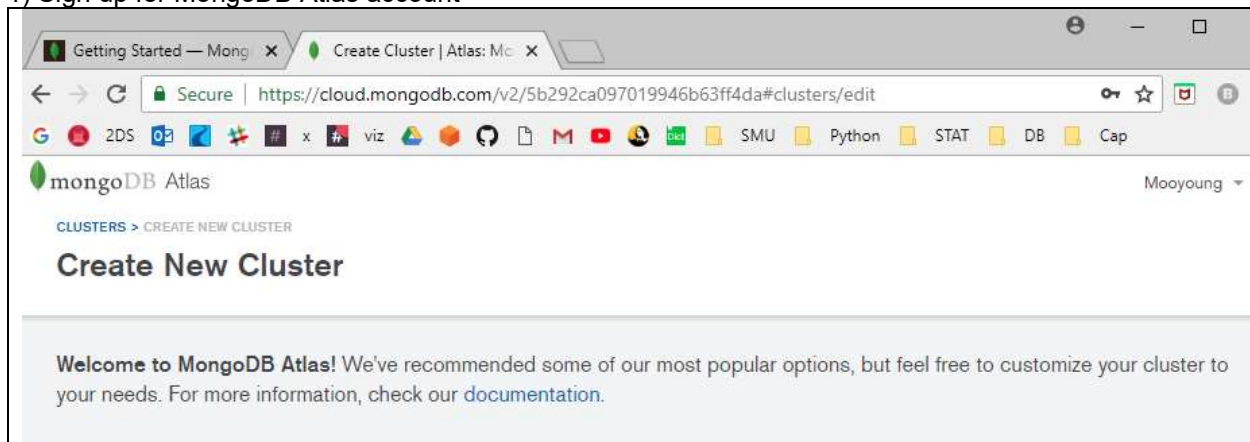
#### Part 1 – MongoDB

The purpose of this part is to gain familiarity with MongoDB Atlas DB Managed Cloud Service. Please sign up for MongoDB account on [cloud.mongodb.com](https://cloud.mongodb.com). Once you are signed up to MongoDB Atlas, build and deploy a cluster using M0 instance (Free). Download MongoDB Compass on your local machine and connect to the MongoDB Atlas cluster that you just created. Instructions for connecting to the cluster can be found on MongoDB Atlas. You have to make sure that IPs are added in the white list similar to what you did in AWS.

I have included a csv file that contains **postcode** data. Use `mongoimport` command to import the csv file into the MongoDB Atlas. You can find detailed instruction for `mongoimport` in <https://docs.atlas.mongodb.com/importmongoimport>

Using MongoDB Compass analyze data, click under "Document" and experiment with "Filters". Similarly go into "Schema" and "Explain Plan". Submit screenshots of each of these activities.

#### 1) Sign up for MongoDB Atlas account



#### 2) Create a cluster in MongoDB Atlas

A. [Select Cloud provider, and select a free tier available region](#)

Cloud Provider & Region

AWS, N. Virginia (us-east-1)

Create a **free tier cluster** by selecting a region with **FREE TIER AVAILABLE** and choosing the **M0** cluster tier below.

★ recommended region

NORTH AMERICA

N. Virginia (us-east-1) ★  
**FREE TIER AVAILABLE**

Ohio (us-east-2) ★

N. California (us-west-1)

Oregon (us-west-2) ★

Montreal (ca-central-1)

EUROPE

Ireland (eu-west-1) ★

London (eu-west-2) ★

Paris (eu-west-3) ★

Frankfurt (eu-central-1) ★  
**FREE TIER AVAILABLE**

SOUTH AMERICA

Sao Paulo (sa-east-1)

AUSTRALIA

Sydney (ap-southeast-2) ★

ASIA

Tokyo (ap-northeast-1) ★

Seoul (ap-northeast-2)

Singapore (ap-southeast-1) ★  
**FREE TIER AVAILABLE**

Mumbai (ap-south-1)  
**FREE TIER AVAILABLE**

Configure clusters across multiple regions (M10 and up)

☐ NO

Better withstand data center outages and optimize for reads in selected service areas.

PREVIOUS: CLOUD PROVIDER & REGION

NEXT: CLUSTER TIER

B. Select cluster tier, and it comes with MongoDB 3.6 automatically

Cluster Tier

M0 (Shared RAM, 512 MB Storage)

Encrypted

Base hourly rate is for a MongoDB replica set with 3 data bearing servers.

Shared Clusters

M0	Shared RAM	512 MB Storage	Shared VCPUs	FREE
M2	Shared RAM	2 GB Storage	Shared VCPUs	from \$0.012/hr
M5	Shared RAM	5 GB Storage	Shared VCPUs	from \$0.035/hr

Dedicated Development Clusters

M10	2 GB RAM	10 GB Storage	0.2 vCPUs	from \$0.08/hr
M20	4 GB RAM	20 GB Storage	0.4 vCPUs	from \$0.20/hr

Dedicated Production Clusters

M30	8 GB RAM	40 GB Storage	2 vCPUs	from \$0.54/hr
> M40	16 GB RAM	80 GB Storage	4 vCPUs	from \$1.04/hr
> M50	32 GB RAM	160 GB Storage	8 vCPUs	from \$2.00/hr
> M60	64 GB RAM	320 GB Storage	16 vCPUs	from \$3.95/hr
M80 <small>LOW CPU</small>	122 GB RAM	750 GB Storage	16 vCPUs	from \$5.61/hr
M100	160 GB RAM	1000 GB Storage	40 vCPUs	from \$9.16/hr
> M200	256 GB RAM	1500 GB Storage	64 vCPUs	from \$14.59/hr
M400 <small>LOW CPU</small>	488 GB RAM	3000 GB Storage	64 vCPUs	from \$22.40/hr

PREVIOUS: CLOUD PROVIDER & REGION

NEXT: ADDITIONAL SETTINGS

Additional Settings

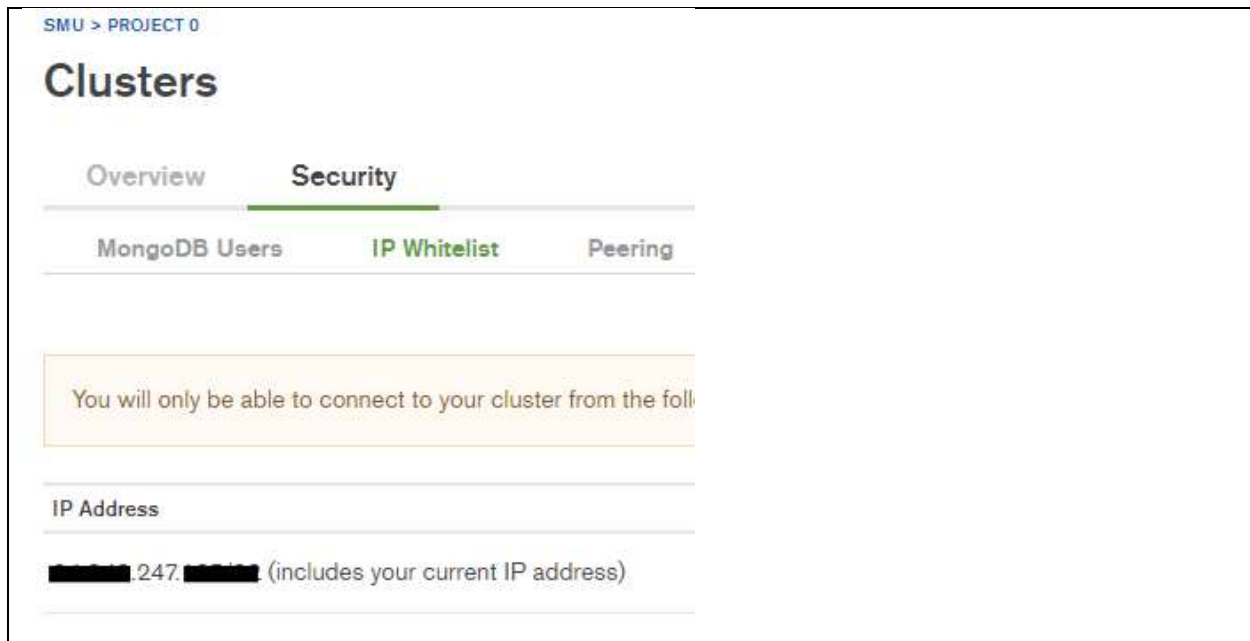
MongoDB 3.6, No Backup

Select a version

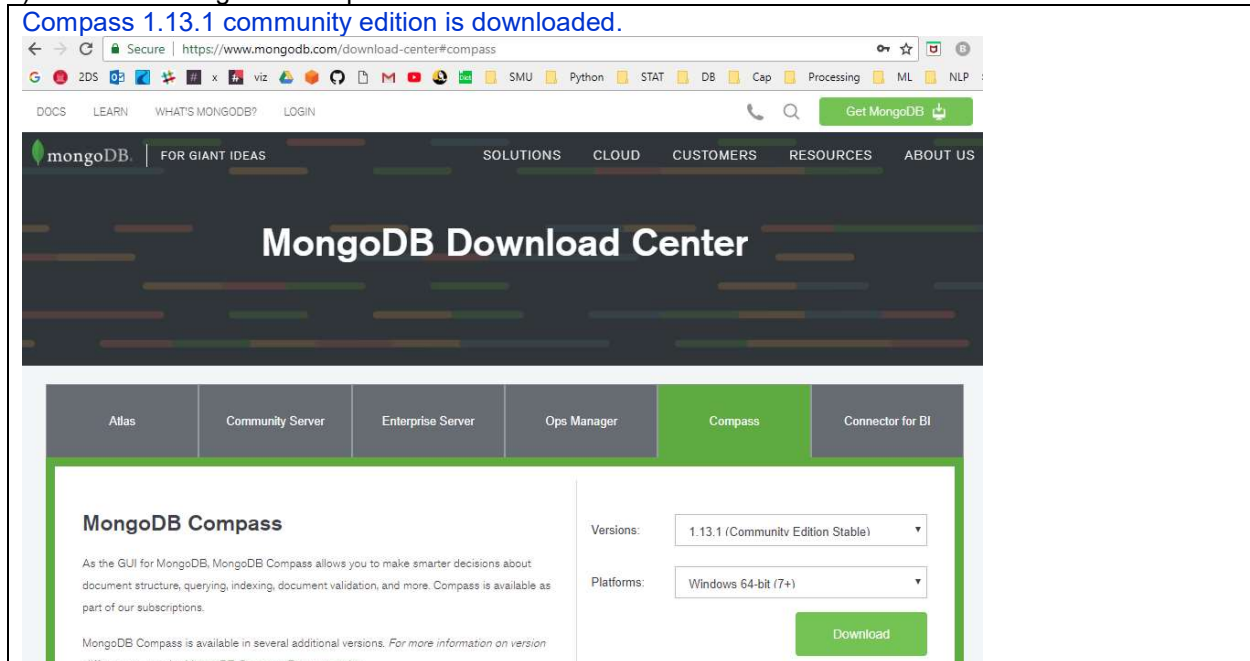
3.6 with WiredTiger™

All clusters launch with the WiredTiger™ storage engine.

C. IP Whitelist is modified to allow incoming connection. One easy way to add current IP address is going to Clusters/Sandbox/Connect/Add Current IP address.

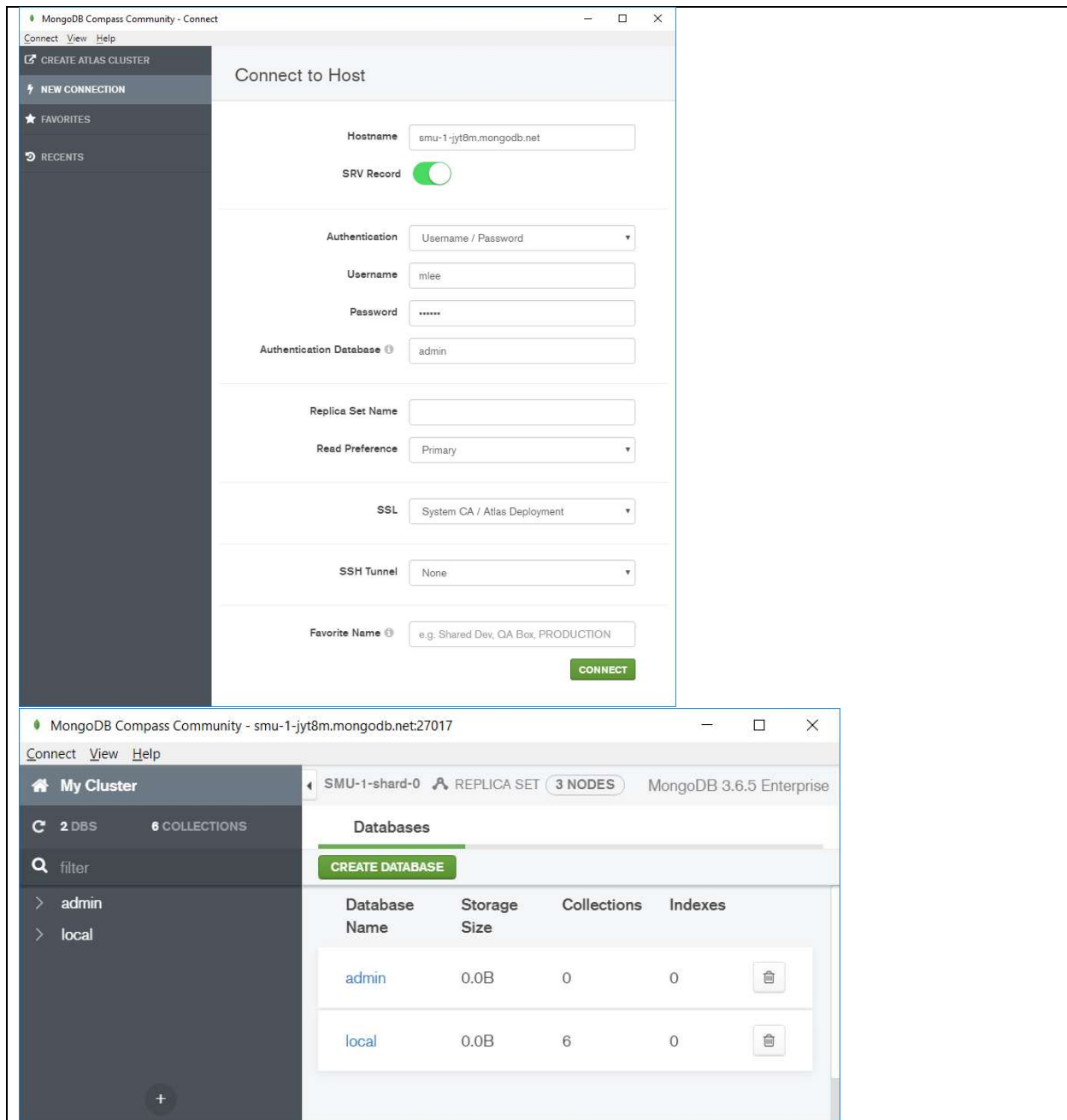


### 3) Download MongoDB Compass



### 4) Connect MongoDB Compass to Atlas's cluster

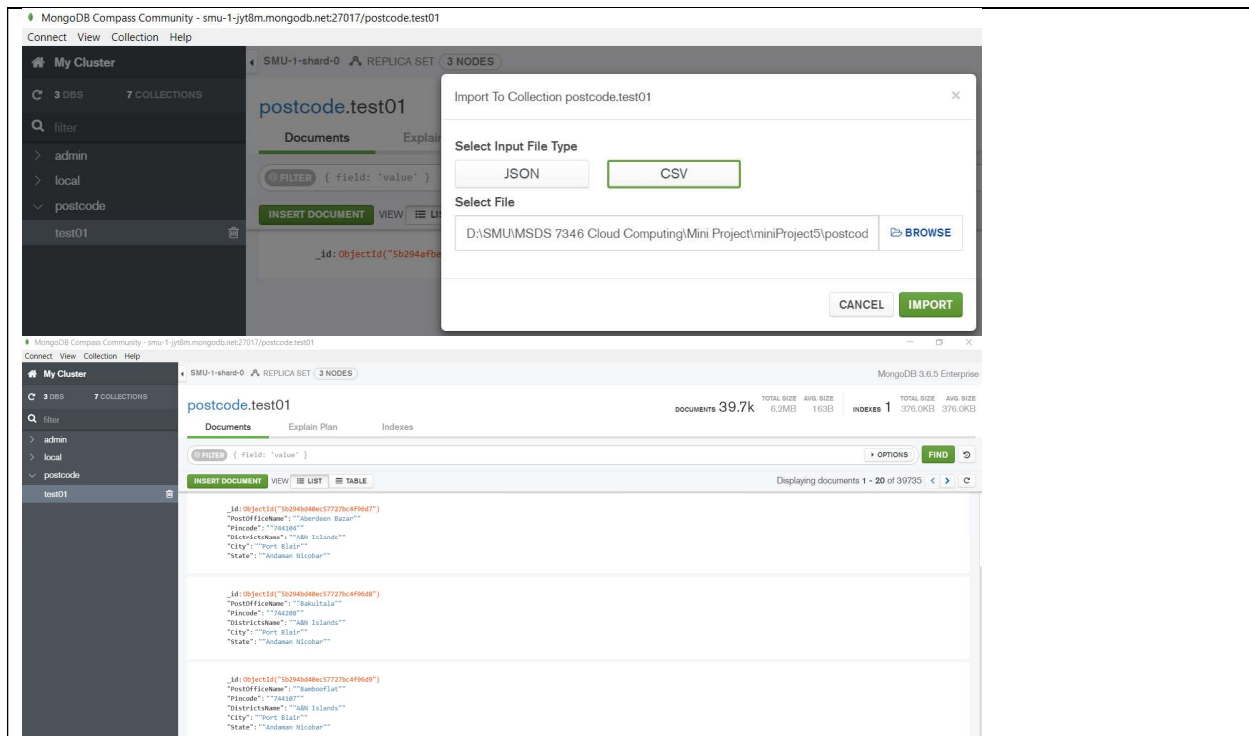
- Go to 'Clusters/ Sandbox/ Connect/ Connect with MongoDB Compass/ I am using Compass 1.12 or later/ copy the URI connection string
  - Open MongoDB Compass, and go to 'New Connection'
  - The Compass will automatically fill the host name using the clipboard data.
- Source: <https://docs.mongodb.com/compass/current/connect/>



##### 5) Experiment with Compass as stated above

###### Import CSV file

- A database(postcode) and a collection(test01) is created from home/create database
- A postcode.csv file is imported using a Compass import feature from Collection/Import



MongoDB Compass Community - smu-1-jy8m.mongodb.net:27017/postcode.test01

Connect View Collection Help

My Cluster

3 DBS 7 COLLECTIONS

filter

admin

local

postcode

test01

postcode.test01

Documents Explain

FILTER { field: 'value' }

INSERT DOCUMENT VIEW LIST

\_id: ObjectId("5b294a1b...")

Import To Collection postcode.test01

Select Input File Type

JSON CSV

Select File

D:\SMU\MSDS 7346 Cloud Computing\Mini Project\miniProject5\postcod BROWSE

CANCEL IMPORT

MongoDB 3.6.5 Enterprise

postcode.test01

DOCUMENTS 39.7k TOTAL SIZE 6.2MB AVG. SIZE 163B INDEXES 1 TOTAL SIZE 376.0KB AVG. SIZE 376.0KB

Documents Explain Plan Indexes

FILTER { field: 'value' } OPTIONS FIND

INSERT DOCUMENT VIEW LIST TABLE

Displaying documents 1 - 20 of 39735

\_id: ObjectId("5b294a1b...")

PostOfficeName: "Dongdeon Baza"

Pincode: "744398"

DistrictName: "Abi Islands"

City: "Port Blau"

State: "Andaman Nicobar"

\_id: ObjectId("5b294a1b...")

PostOfficeName: "Bakulata"

Pincode: "744398"

DistrictName: "Abi Islands"

City: "Port Blau"

State: "Andaman Nicobar"

\_id: ObjectId("5b294a1b...")

PostOfficeName: "Bamoorlat"

Pincode: "744398"

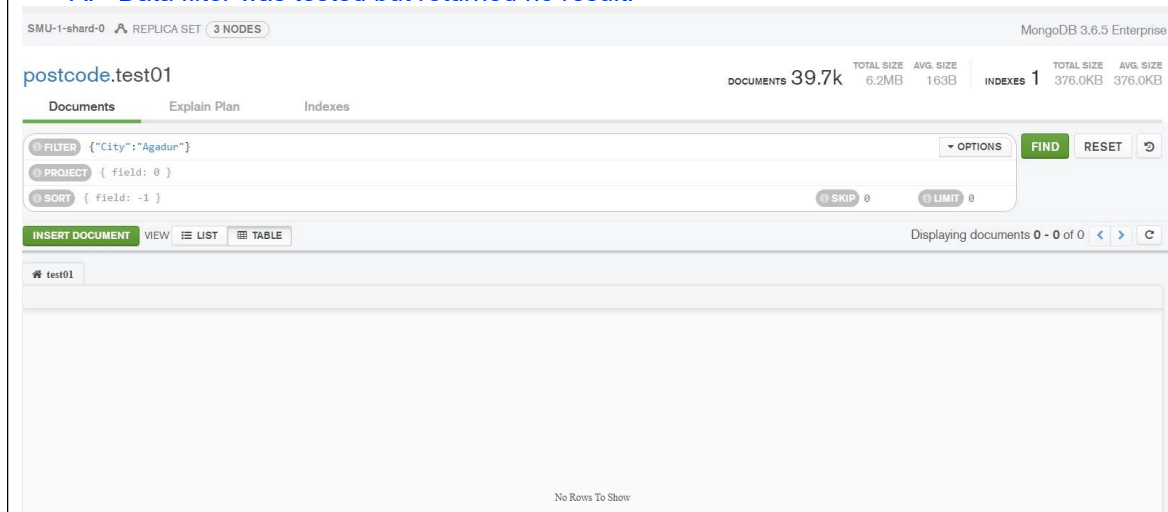
DistrictName: "Abi Islands"

City: "Port Blau"

State: "Andaman Nicobar"

## Explore Compass

### A. Data filter was tested but returned no result.



SMU-1-shard-0 REPLICATION SET 3 NODES

MongoDB 3.6.5 Enterprise

postcode.test01

DOCUMENTS 39.7k TOTAL SIZE 6.2MB AVG. SIZE 163B INDEXES 1 TOTAL SIZE 376.0KB AVG. SIZE 376.0KB

Documents Explain Plan Indexes

FILTER { City: "Agadun" } OPTIONS FIND RESET

PROJECT { field: 0 }

SORT { field: -1 } SKIP 0 LIMIT 0

INSERT DOCUMENT VIEW LIST TABLE

Displaying documents 0 - 0 of 0

test01

No Rows To Show

### B. Explain Plan tab shows some query summaries.

The screenshot displays the MongoDB Compass interface for a database named 'SMU-1-shard-0' with a replica set of 3 nodes. The selected collection is 'postcode.test01'. The 'Explain Plan' tab is active, showing a query with a filter on 'field: 'value'', a project on 'field: 0', and a sort on 'field: -1'. The 'Query Performance Summary' indicates 39735 documents returned, 0 index keys examined, and 39735 documents examined. The actual query execution time is 11 ms, and it is sorted in memory. A warning states 'No index available for this query.' A callout box highlights the 'COLLSCAN' operation with a returned count of 39735, an execution time of 10 ms, and documents examined of 39735. Below this, a blue text label reads 'C. Indexes tab shows the existing indexes.'

The second screenshot shows the 'Indexes' tab for the same collection. It features a 'CREATE INDEX' button and a table of existing indexes. The table has columns for 'Name and Definition', 'Type', 'Size', and 'Usage'. One index is listed: '\_id\_' with a definition of '\_id\_'. Its type is 'REGULAR', its size is 376.0 KB, and it has been used 7 times since Tuesday, June 19, 2018. The overall document count is 39.7k and the total size is 6.2MB.

Name and Definition	Type	Size	Usage
_id_ _id_	REGULAR	376.0 KB	7 since Tue Jun 19 2018

#### 6) Submit screenshots

Submission: Submit different screen shots to show completion of each steps

Collaborators: [None](#)

Resources:

[MongoDB/Compass Import Data](#)

< <https://docs.mongodb.com/compass/master/import-export/> >

[MongoDB connection](#)

<<https://docs.mongodb.com/compass/current/connect/>>

[Connection String URI Format](#)

<<https://docs.mongodb.com/manual/reference/connection-string/>>

[Load File with mongoimport](#)

<<https://docs.atlas.mongodb.com/import/mongoimport/>>

## Part 2 – End-to-end AWS

The purpose of this exercise is to get hands-on experience working with AWS, more specifically I am looking for you to create a fully functional configuration of Application Servers, Load Balancer, RDS, and S3. We have worked with all of these pieces during our class. This lab brings all of the pieces of the puzzle together.

**Application Servers** – are used to host application such as your website and provide service to external users.

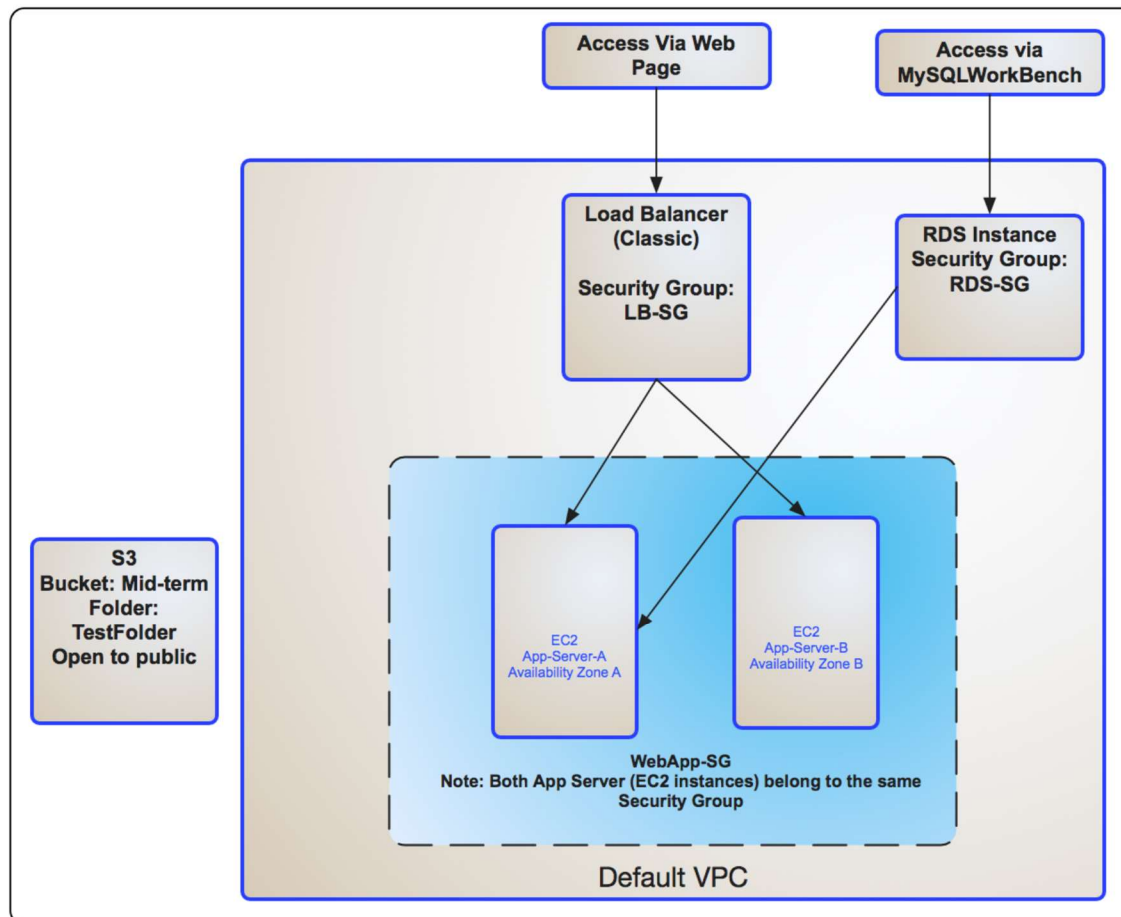
**Load Balancer** – is used to distribute traffic between application server nodes. In this exercise you will configure classic load balancer that will direct user request in a round robin fashion (alternate between different EC2 instances.)

**RDS** – is used to store data. In this exercise we will not be storing or retrieving any data as we will not host any application.

The objective is connectivity of these building blocks.

**S3** – object storage is created for the purpose of backup.

Fig 1 – Architecture of Configuration in AWS



### Description

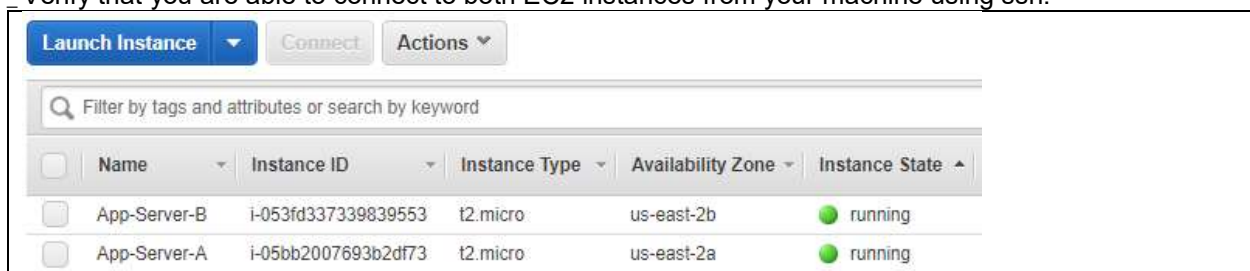
The following are some of the steps that you need to do to complete this lab. This in no way intended to be a complete list, but I tried to provide as much explanation as possible. You are expected to have this complete system working. Towards the end of this document, you will find screenshots from my AWS account to give you some idea of different configurations.



In this mini project, not only you would setup AWS, but will also install a "Hello World" HTML file, simulating a web application on EC2 instances. This web app will be accessed via browser using the DNS address of the load balancer. This will demonstrate how user gets access to an web application running on EC2 instance and in addition provides insight into how the load balancer works. You also see a MySQLWorkbench on your local machine and this is simply there to test connection to the RDS instance and also to validate if your database command on EC2. Figure 1 below should give you an architecture of the final configuration would look like.

### 1) EC2

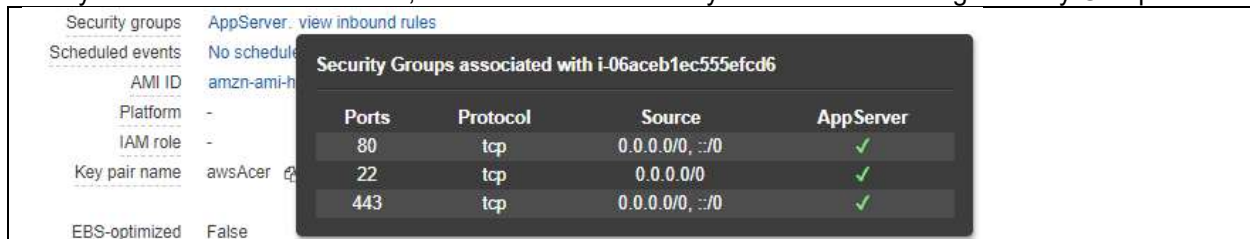
- \_ Create two EC2 instances (use free tier) for each of these two use the default VPC (Virtual Private Cloud).
- \_ Create a new Security Group which will allow all traffic from any source. You will need to only create this when you launch the first instance, for the second instance you will select existing Security Group.
- \_ Create these instances in different availability zones. To make it easy, name them App-Server-A for instance in availability zone A and App-Server-B for availability zone B.
- \_ Verify that you are able to connect to both EC2 instances from your machine using ssh.



	Name	Instance ID	Instance Type	Availability Zone	Instance State
<input type="checkbox"/>	App-Server-B	i-053fd337339839553	t2.micro	us-east-2b	running
<input type="checkbox"/>	App-Server-A	i-05bb2007693b2df73	t2.micro	us-east-2a	running

### 2) Security Group

- \_ Create a new Security Group which will allow all traffic from any source. You will need to only create this when you launch the first instance, for the second instance you will select existing Security Group.



Ports	Protocol	Source	AppServer
80	tcp	0.0.0.0/0, :::/0	✓
22	tcp	0.0.0.0/0	✓
443	tcp	0.0.0.0/0, :::/0	✓

### 3) Availability Zone

- \_ Each region, as we discussed in the class, by default has three zones.
- \_ For example if you logged into US West 2 region then you will have us-west-2a, us-west-2b, us-west-2c availability zones available to you.
- \_ Create both of the EC2 instances in two different availability zones.
- \_ To make it easy to remember, name each of these EC2 instances as App-Server-A for instance in availability zone A and App-Server-B for availability zone B.

### 4) Web Application

- \_ Now that you have verified these instances. Install httpd on your EC2 instances.
- \_ Ye need this to run a basic html file (simulating a Web Application).
- \_ Use following instructions as a guideline:
  - Log into EC2 instances one at a time via SSH

Followed instruction shown on the below website for the rest of Web Application process.

[https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP\\_Tutorials.WebServerDB.CreateWebServer.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateWebServer.html)

# Install an Apache Web Server with PHP

```
# This install server and change permissions on the 'www' directory
```

```
sudo yum update -y
sudo yum install -y httpd24 php56 php56-mysqlnd
sudo service httpd start
sudo chkconfig httpd on
sudo groupadd www
sudo usermod -a -G www ec2-user
exit
groups
sudo chown -R root:www /var/www
sudo chmod 2775 /var/www
find /var/www -type d -exec sudo chmod 2775 {} +
find /var/www -type f -exec sudo chmod 0664 {} +
```

```
# Create index.html file
```

```
cd /var/www/html
```

```
>index.html
```

```
nano index.html
```

```
# Type html code as shown below, and close with saving changes.
```

```
<!DOCTYPE html>
```

```
<html>
```

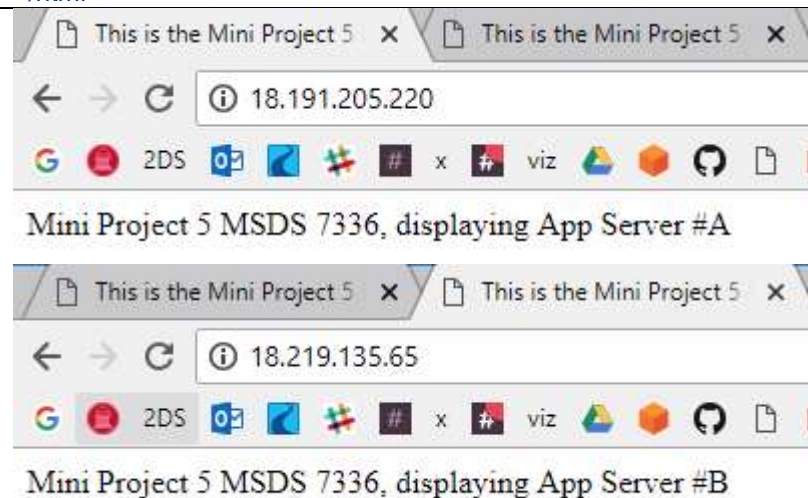
```
<header><title>This is the Mini Project 5 HTML File</title></header>
```

```
<body>
```

```
Mini Project 5 MSDS 7336, displaying App Server #A
```

```
</body>
```

```
</html>
```



- Once you are in EC2 instance use "yum" to install httpd:  

```
sudo yum install -y httpd
```
- install the index.html (provided separately) in `/var/www/html` directory ← 'install' is wrong, create/copy file.
- Modify the # sign in the file to reflect the instance number For example "Mini Project MSDS 7336, displaying App Server #" should be either A for App-Server-A and B for App-Server-B
- Now you start the server:  

```
sudo service httpd start
```

\_ At this time if you take the dns address of your EC2 instance and put that in the browser you would see the "MiniProject ..." message.

– NOTE: If you do not see this message then make sure httpd is running and your file is in the correct directory.

### 5) Load Balancer

\_ Configure Classic Load Balancer using the default VPC. You should allow all http traffic

\_ Add both nodes (instances) to the Load Balancer

\_ Make sure health check is setup.

\_ You should see the status of both instances go from OutOfService to InService. It takes couple of minutes for the status to change.

\_ Put the DNS name that you find under "Description" tab of the Load Balancer in a browser you will see the above message from html file display.

\_ Refresh your browser and you will see it go to the next EC2 instance. Keep refreshing it to see different messages.

#### A. Load balancer is working.

The screenshot shows the AWS Management Console interface for a Classic Load Balancer. At the top, there's a 'Create Load Balancer' button and an 'Actions' dropdown. Below is a search bar and a table listing the load balancer. The table has columns: Name, DNS name, State, VPC ID, and Availability Zones. The entry 'LB-APP' is listed with DNS name 'LB-APP-743055387.us-east-2.elb.amazonaws.com', VPC ID 'vpc-9fd9a7f7', and Availability Zones 'us-east-2a, us-east-2b, ...'. Below the table, the 'Load balancer: LB-APP' section is visible with tabs for Description, Instances (selected), Health Check, Listeners, Monitoring, Tags, and Migration. Under the 'Instances' tab, it shows 'Connection Draining: Enabled, 300 seconds (Edit)' and an 'Edit Instances' button. A table lists the instances attached to the load balancer:

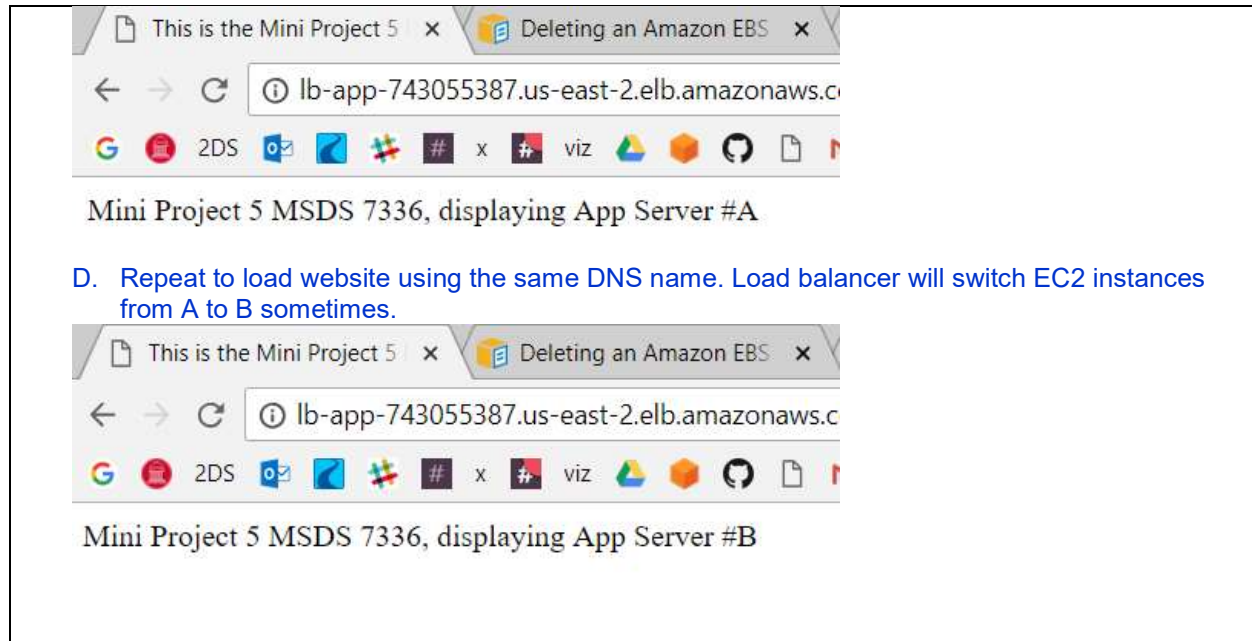
Instance ID	Name	Availability Zone	Status
i-053fd337339839553	App-Server-B	us-east-2b	InService ⓘ
i-05bb2007693b2df73	App-Server-A	us-east-2a	InService ⓘ

#### B. Locate DNS name from load balancer

The screenshot shows the 'Description' tab of the 'LB-APP' Classic Load Balancer. It displays the 'Basic Configuration' section with the following details:

- Name:** LB-APP
- \* DNS name:** [LB-APP-743055387.us-east-2.elb.amazonaws.com](https://LB-APP-743055387.us-east-2.elb.amazonaws.com) (A Record)
- Type:** Classic (Migrate Now)
- Scheme:** internet-facing
- Availability Zones:** subnet-148de77c - us-east-2a, subnet-1c43e066 - us-east-2b, subnet-725c4d3f - us-east-2c

#### C. Load website using web browser



## 6) RDS

\_ Configure RDS instance

\_ Once RDS is configured, go to one of the EC2 instance and install MySQL client as it does not come installed by default. (Note: Only do this on one EC2 instance)

\_ Use "yum" command to do that. Exact syntax as I showed earlier for httpd except now it would be MySQL.

Below code is used to install mysql server, generate a password, and create database.

```
sudo yum install -y mysql-server
Sudo service mysqld start
mysqladmin -u root password 'new_password'
mysqladmin -u root -p create 'db_app'
mysql -u root -p
```

```
ec2-user@ip-172-31-15-118:~
Verifying : 1:perl-Compress-Raw-Zlib-2.061-4.1.amzn1.x86_64 4/10
Verifying : perl-DBD-MySQL55-4.023-5.23.amzn1.x86_64 5/10
Verifying : perl-Net-Daemon-0.48-5.5.amzn1.noarch 6/10
Verifying : perl-Compress-Raw-Bzip2-2.061-3.11.amzn1.x86_64 7/10
Verifying : mysql55-server-5.5.60-1.21.amzn1.x86_64 8/10
Verifying : perl-Data-Dumper-2.145-3.5.amzn1.x86_64 9/10
Verifying : mysql-server-5.5-1.6.amzn1.noarch 10/10

Installed:
mysql-server.noarch 0:5.5-1.6.amzn1

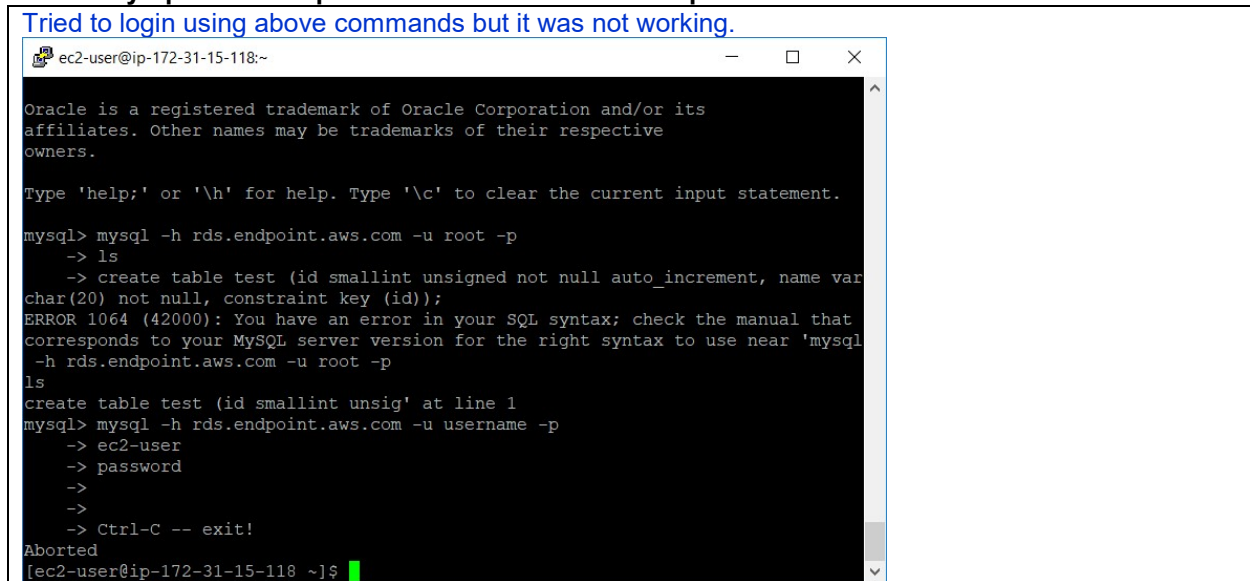
Dependency Installed:
mysql55-server.x86_64 0:5.5.60-1.21.amzn1
perl-Compress-Raw-Bzip2.x86_64 0:2.061-3.11.amzn1
perl-Compress-Raw-Zlib.x86_64 1:2.061-4.1.amzn1
perl-DBD-MySQL55.x86_64 0:4.023-5.23.amzn1
perl-DBI.x86_64 0:1.627-4.8.amzn1
perl-Data-Dumper.x86_64 0:2.145-3.5.amzn1
perl-IO-Compress.noarch 0:2.061-2.12.amzn1
perl-Net-Daemon.noarch 0:0.48-5.5.amzn1
perl-PlRPC.noarch 0:0.2020-14.7.amzn1

Complete!
[ec2-user@ip-172-31-15-118 ~]$
```

\_ Login to the MySQL using the following command:

**mysql -h rds.endpoint.aws.com -u username -p**

Tried to login using above commands but it was not working.

A terminal window titled 'ec2-user@ip-172-31-15-118:~' with standard window controls. The terminal shows the following text:

```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> mysql -h rds.endpoint.aws.com -u root -p
-> ls
-> create table test (id smallint unsigned not null auto_increment, name var
char(20) not null, constraint key (id));
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'mysql
-h rds.endpoint.aws.com -u root -p
ls
create table test (id smallint unsig' at line 1
mysql> mysql -h rds.endpoint.aws.com -u username -p
-> ec2-user
-> password
->
->
-> Ctrl-C -- exit!
Aborted
[ec2-user@ip-172-31-15-118 ~]$
```

- \_ One you are there create a database using the create database command
- \_ Now go to the MySQLWorkbench connect to the RDS and you should see that database.

Collaborators: Albert Asuncion

Resources:

Classic Load Balancer <<https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-getting-started.html>>

## Submission

The following is what you need to submit.

\_ Provide few **bullet points or a short paragraph** of what you were able to work, if you had problems or couldn't get some things to work please provide brief explanation here.

→ Please see above screen shots and notes.

\_ Submit screenshots of all of the different configurations depicting the architecture picture in Figure 1. I also want to see the screenshot your web application (html page) showing different application server names. Arrange these screenshots with proper labels and explanation for each screenshot to make is clear for the grader.

\_ If you still run into any problems when it is time to submit, please explain what steps you have taken and provide screenshots for all of those steps.

\_ Good Luck and Enjoy!!