

Lesson 01 Demo 03

Performing CRUD Operations with Mongo Shell

Objective: To perform CRUD operations with Mongo Shell by installing mongosh and performing update operations

Tools Required: VS Code

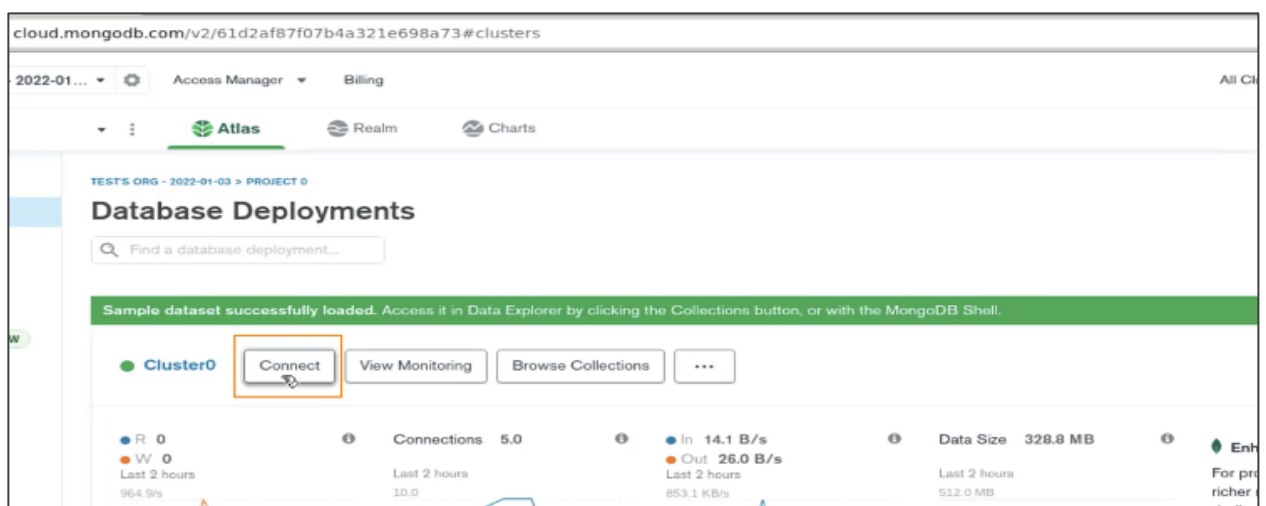
Prerequisites: None

Steps to be followed:

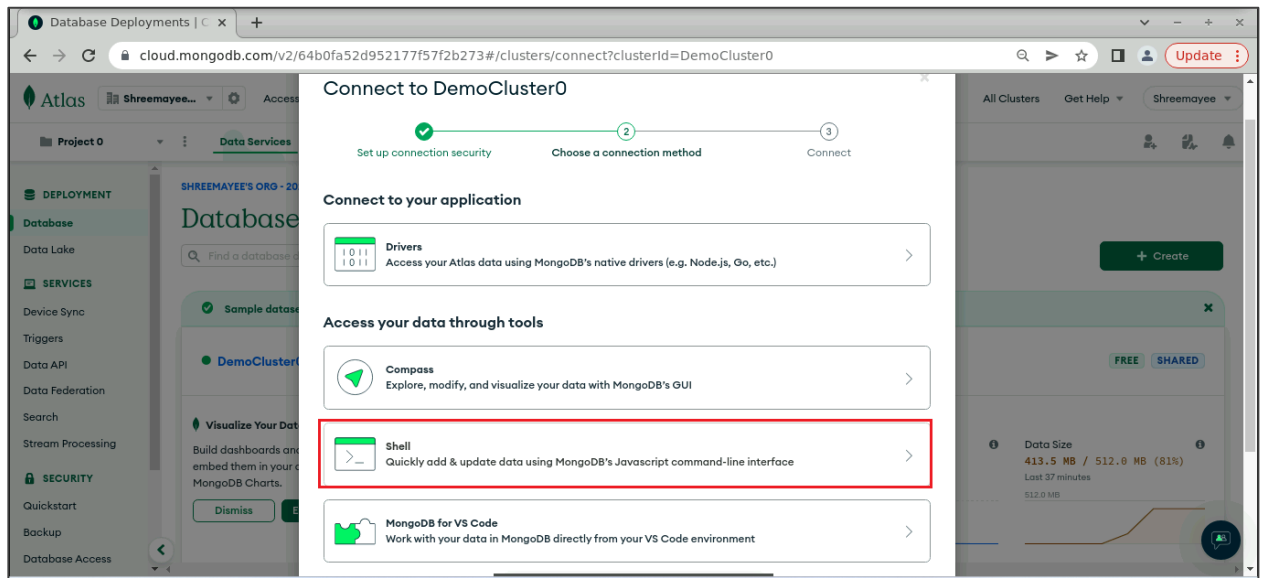
1. Install mongosh
2. Install Mongo Shell
3. Connect to MongoDB
4. Execute commands
5. Create a document
6. Insert more documents
7. Perform update operations

Step 1: Install mongosh

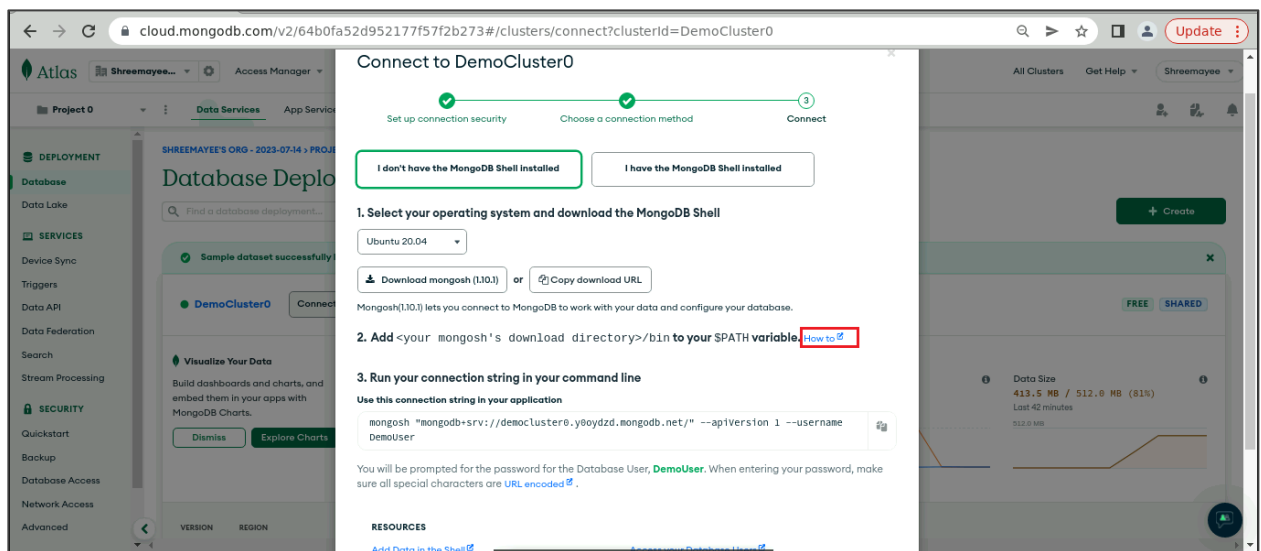
- 1.1 On the main interface of MongoDB Atlas, click on **Connect**



1.2 Select the **Shell** option to connect with MongoDB Shell



1.3 Click on **How to** and set the path variable for the MongoDB shell



1.4 Follow the instructions to import the public key used by the package management system

1

Import the public key used by the package management system.

From a terminal, issue the following command to import the MongoDB public GPG Key from <https://www.mongodb.org/static/pgp/server-5.0.asc>:

```
wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -
```

The operation should respond with an `OK`.

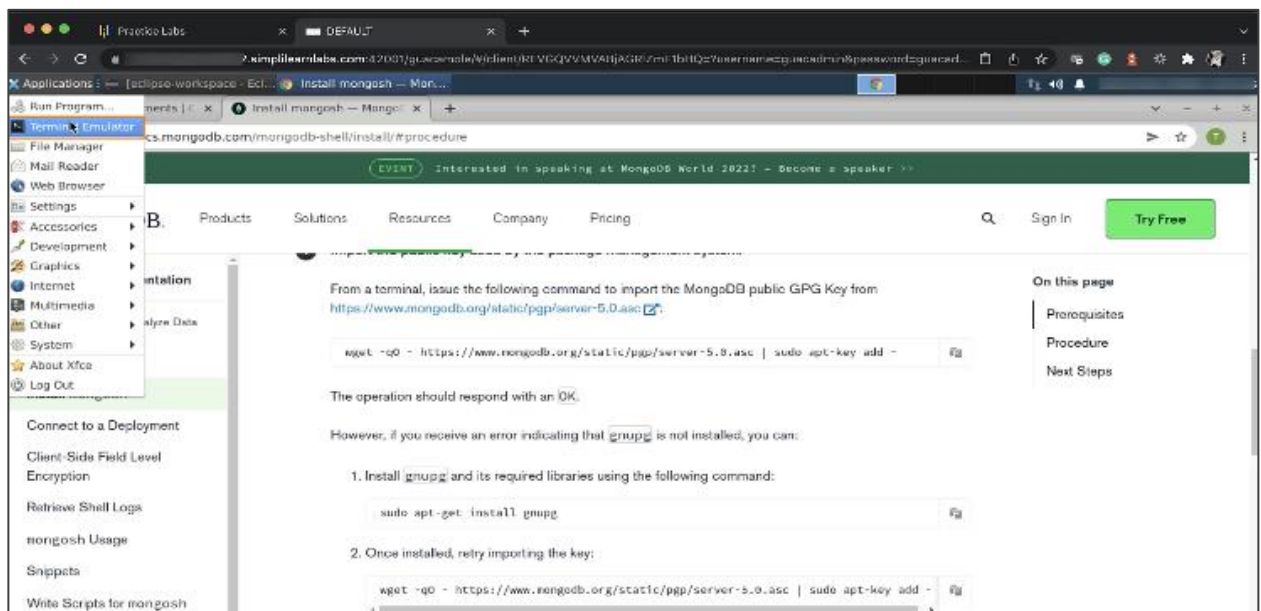
However, if you receive an error indicating that `gnupg` is not installed, you can:

1. Install `gnupg` and its required libraries using the following command:

```
sudo apt-get install gnupg
```

2. Once installed, retry importing the key:

1.5 Open a Terminal Emulator



- 1.6 Copy the command below and paste it into the terminal to import the public GPG key:
sudo apt-get install gnupg

1

Import the public key used by the package management system.

From a terminal, issue the following command to import the MongoDB public GPG Key from <https://www.mongodb.org/static/pgp/server-5.0.asc>:

```
wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -
```

The operation should respond with an `OK`.

However, if you receive an error indicating that `gnupg` is not installed, you can:

1. Install `gnupg` and its required libraries using the following command:

```
sudo apt-get install gnupg
```
2. Once installed, retry importing the key:

```
wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -
```

- 1.7 Paste the command in the terminal and press **Enter**

The screenshot shows a terminal window with the following commands and output:

```

l@ip-172-31-17-157:~$ wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -
OK
l@ip-172-31-17-157:~$ sudo apt-get install gnupg
Reading package lists... Done
Building dependency tree
Reading state information... Done
gnupg is already the newest version (2.2.19-3ubuntu2.1).
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
l@ip-172-31-17-157:~$
  
```

1.8 Execute the release command to check the version

lsb_release -dc

```

@ip-172-31-17-157:~$ wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -
OK
@ip-172-31-17-157:~$ sudo apt-get install gnupg
Reading package lists... Done
Building dependency tree
Reading state information... Done
gnupg is already the newest version (2.2.19-3ubuntu2.1).
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
@ip-172-31-17-157:~$ wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -
OK
@ip-172-31-17-157:~$ lsb_release -dc.
Usage: lsb_release [options]

lsb_release: error: no such option: -.
@ip-172-31-17-157:~$ lsb_release -dc
Description:    Ubuntu 20.04.3 LTS
Codename:       focal
@ip-172-31-17-157:~$
  
```

1.9 Follow the Ubuntu 20.04 instructions. Create a file for Ubuntu 20.04 by copying the command from the screenshot below:

Ubuntu 20.04 (Focal) Ubuntu 18.04 (Bionic)

The following instruction is for **Ubuntu 20.04 (Focal)**. For Ubuntu 18.04 (Bionic) click on the appropriate tab.

Create the `/etc/apt/sources.list.d/mongodb-org-5.0.list` file for Ubuntu 20.04 (Focal):

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0" | sudo tee /etc/apt/sources.list.d/mongodb-org-5.0.list
```

3 Reload local package database.

Issue the following command to reload the local package database:

```
sudo apt-get update
```

1.10 Paste it into the terminal

```

server-5.0.asc | sudo apt-key add -
OK
gip-172-31-17-157:~$ sudo apt-get install gnupg
Reading package lists... Done
Building dependency tree
Reading state information... Done
gnupg is already the newest version (2.2.19-3ubuntu2.1).
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
gip-172-31-17-157:~$ wget -qO - https://www.mongodb.org/static/pgp/
server-5.0.asc | sudo apt-key add -
OK
gip-172-31-17-157:~$ lsb_release -dc.
Usage: lsb_release [options]
lsb_release: error: no such option: -.
gip-172-31-17-157:~$ lsb_release -dc
Description: Ubuntu 20.04.3 LTS
Codename: focal
gip-172-31-17-157:~$ echo "deb [ arch=amd64,arm64 ] https://repo.m
mongodb.org/apt/ubuntu focal/mongodb-org/5.0 multiverse" | sudo tee /etc/apt/sourc
es.list.d/mongodb-org-5.0.list
deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5
.0 multiverse
gip-172-31-17-157:~$

```

Retrieve Shell Logs

mongosh Usage

Snippets

Write Scripts for mongosh

Reference

Access mongosh Help

Compatibility Changes with

3 Reload local package database.

Issue the following command to reload the local package database:

sudo apt-get update

Give Feedback

1.11 Once the file is created, reload the local package using the command below: **sudo apt-get update**

```

erishantgmail@gip-172-31-17-157:~$ lsb_release -dc.
lsb_release: error: no such option: -.
erishantgmail@gip-172-31-17-157:~$ lsb_release -dc
Description: Ubuntu 20.04.3 LTS
Codename: focal
erishantgmail@gip-172-31-17-157:~$ echo "deb [ arch=amd64,arm64 ] https://repo.m
mongodb.org/apt/ubuntu focal/mongodb-org/5.0 multiverse" | sudo tee /etc/apt/sourc
es.list.d/mongodb-org-5.0.list
deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5
.0 multiverse
erishantgmail@gip-172-31-17-157:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 http://cli.github.com/packages stable InRelease
Hit:6 http://packages.microsoft.com/repos/code stable InRelease
Ign:7 https://pkg.jenkins.io/debian binary/ InRelease
Hit:8 https://pkg.jenkins.io/debian binary/ Release
Hit:9 https://dl.google.com/linux/chrome/deb stable InRelease
Get:10 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 multiverse InRelease
Err:10 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 multiverse InRelease
  404 Not Found [IP: 157.140.2.2]

```

Retrieve Shell Logs

sudo apt-get update

Step 2: Install Mongo Shell

2.1 Install the Mongo Shell using the command below:

sudo apt-get install -y mongodb-mongosh

3 Reload local package database.

Issue the following command to reload the local package database:

```
sudo apt-get update
```

4 Install the **mongosh** package.

To install the latest stable version of **mongosh**, issue the following command:

```
sudo apt-get install -y mongodb-mongosh
```

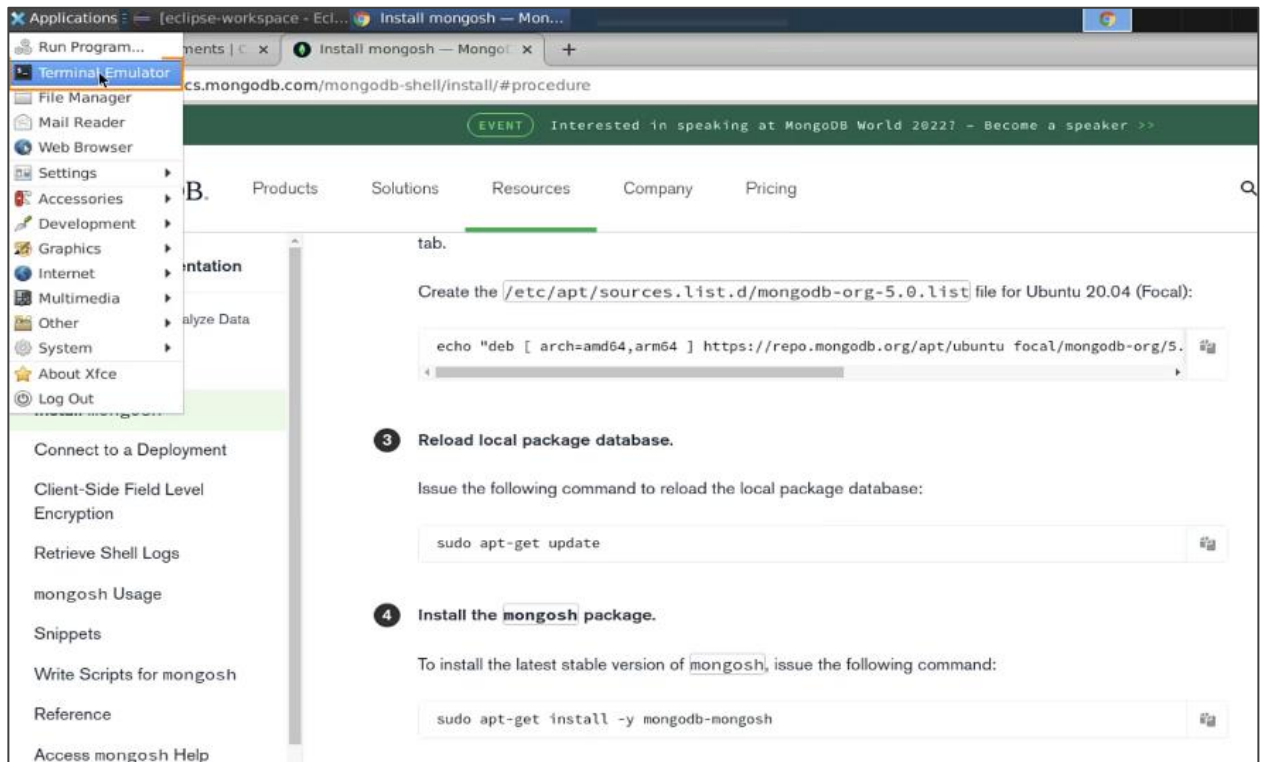
Next Steps

2.2 Paste the command in the terminal

Terminal output:

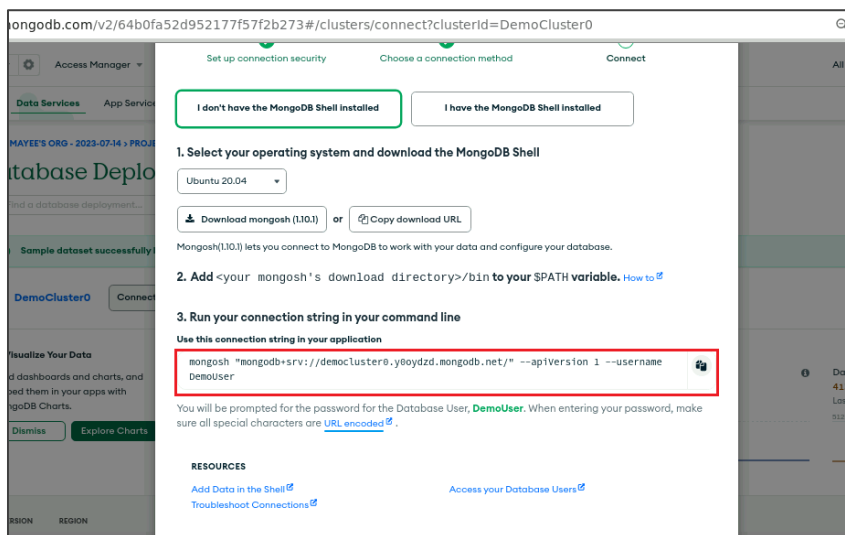
```
gip-172-31-17-157:~$ sudo apt-get update
deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 multiverse
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 https://cli.github.com/packages stable InRelease
Hit:6 http://packages.microsoft.com/repos/code stable InRelease
Ign:7 https://pkg.jenkins.io/debian binary/ InRelease
Hit:8 https://pkg.jenkins.io/debian binary/ Release
Hit:9 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:10 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 InRelease
Hit:11 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 Release
Hit:12 https://repo.zabbix.com/zabbix/5.0/ubuntu focal InRelease
Hit:13 https://deb.nodesource.com/node_17.x focal InRelease
Reading package lists... Done
gip-172-31-17-157:~$ sudo apt-get install -y mongodb-mongosh
Reading package lists... Done
Building dependency tree
Reading state information... Done
mongodb-mongosh is already the newest version (1.1.7).
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
gip-172-31-17-157:~$
```


2.3 Close the terminal and reopen it by clicking on **Terminal Emulator**



Step 3: Connect to MongoDB

3.1 Connect to MongoDB by using the connection string in the application



3.2 Paste it into the terminal and enter the password provided during account creation

```

mongosh mongodb+srv://<credentials>@demoCluster0.y0oydzd.mongodb.net/
File Edit View Search Terminal Help
shreemayeebhatt@ip-172-31-17-182:~$ mongosh "mongodb+srv://demoCluster0.y0oydzd.
mongodb.net/" --apiVersion 1 --username DemoUser
Enter password: *****
Current Mongosh Log ID: 64b11c267d9950f0b68557e6
Connecting to:      mongodb+srv://<credentials>@demoCluster0.y0oydzd.mongodb
.net/?appName=mongosh+1.10.1
Using MongoDB:      6.0.6 (API Version 1)
Using Mongosh:      1.10.1

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

To help improve our products, anonymous usage data is collected and sent to Mong
oDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

Atlas atlas-919l1k-shard-0 [primary] test>

```

It takes some time to establish the connection; once it is established, you will enter the MongoDB shell.

```

mongosh mongodb+srv://<credentials>@demoCluster0.y0oydzd.mongodb.net/
File Edit View Search Terminal Help
shreemayeebhatt@ip-172-31-17-182:~$ mongosh "mongodb+srv://demoCluster0.y0oydzd.
mongodb.net/" --apiVersion 1 --username DemoUser
Enter password: *****
Current Mongosh Log ID: 64b11c267d9950f0b68557e6
Connecting to:      mongodb+srv://<credentials>@demoCluster0.y0oydzd.mongodb
.net/?appName=mongosh+1.10.1
Using MongoDB:      6.0.6 (API Version 1)
Using Mongosh:      1.10.1

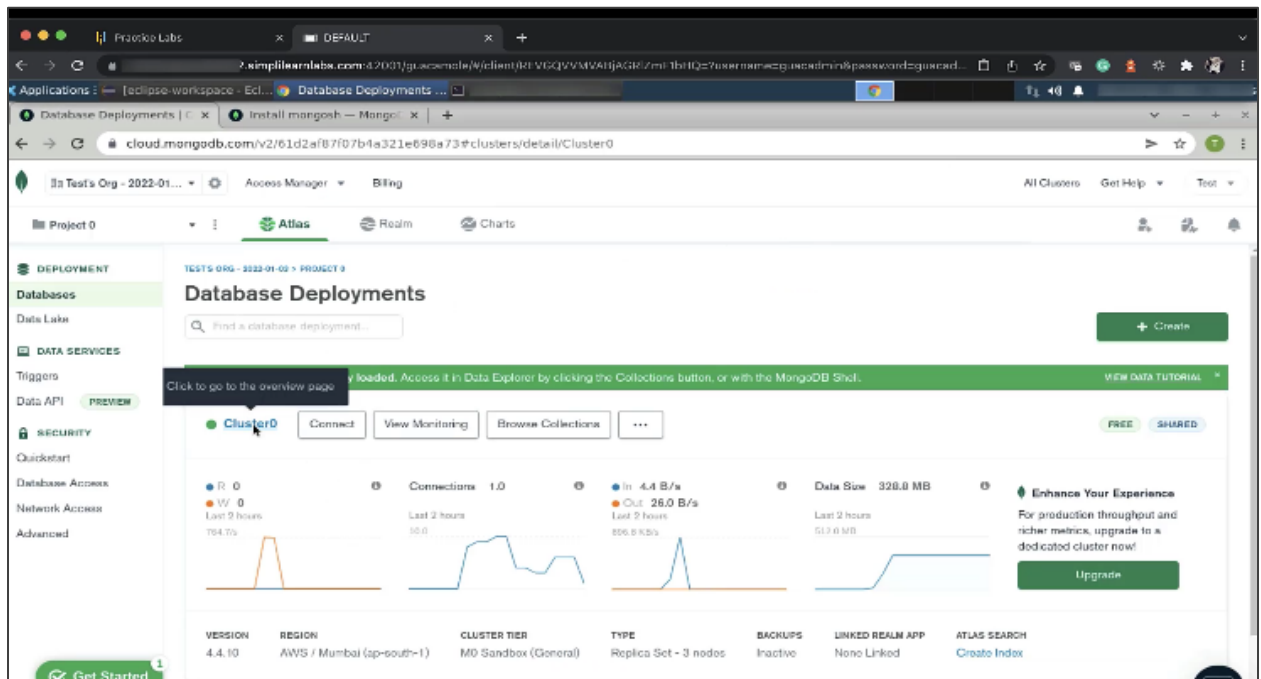
For mongosh info see: https://docs.mongodb.com/mongosh-shell/

To help improve our products, anonymous usage data is collected and sent to Mong
oDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

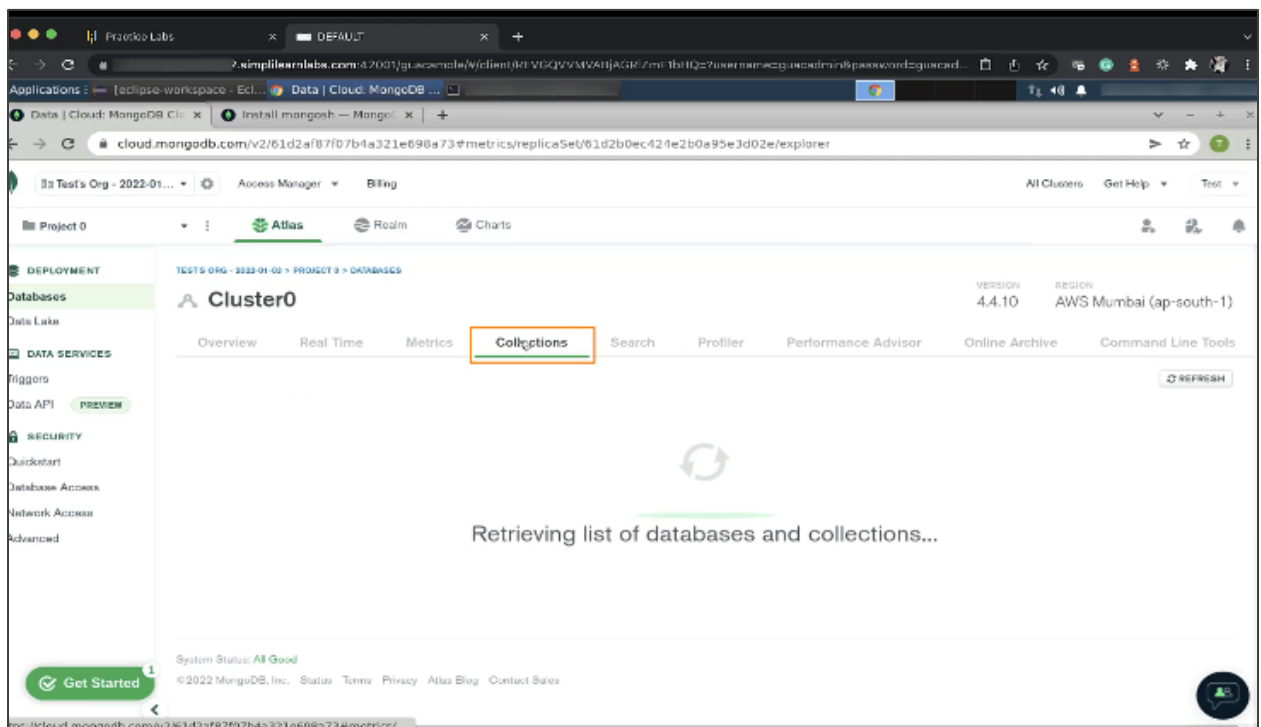
Atlas atlas-919l1k-shard-0 [primary] test>

```

3.3 Select Cluster0



3.4 Navigate to the Collections tab



3.5 Enter **use estore** in the terminal window

```

mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
@ip-172-31-17-157:~$ mongosh "mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase" --username john
Enter password: *****
Current Mongosh Log ID: 61d2c6c8db075dcf4facca44
Connecting to: mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
Using MongoDB: 4.4.10
Using Mongosh: 1.1.7

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

Atlas atlas-z742sk-shard-0 [primary] myFirstDatabase> use estore
switched to db estore
Atlas atlas-z742sk-shard-0 [primary] estore>

```

3.6 Enter the **db** command to understand which database you are working on

```

mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
@ip-172-31-17-157:~$ mongosh "mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase" --username john
Enter password: *****
Current Mongosh Log ID: 61d2c6c8db075dcf4facca44
Connecting to: mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
Using MongoDB: 4.4.10
Using Mongosh: 1.1.7

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

Atlas atlas-z742sk-shard-0 [primary] myFirstDatabase> use estore
switched to db estore
Atlas atlas-z742sk-shard-0 [primary] estore> db
estore
Atlas atlas-z742sk-shard-0 [primary] estore>

```

3.7 Get the collection name using the `db.getCollectionNames()` command

```

mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
@ip-172-31-17-157:~$ mongosh "mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase" --username john
Enter password: *****
Current Mongosh Log ID: 61d2c6c8db075dcf4facca44
Connecting to: mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
Using MongoDB: 4.4.10
Using Mongosh: 1.1.7

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

Atlas atlas-z742sk-shard-0 [primary] myFirstDatabase> use estore
switched to db estore
Atlas atlas-z742sk-shard-0 [primary] estore> db
estore
Atlas atlas-z742sk-shard-0 [primary] estore> db.getCollectionNames()
[ 'products', 'users' ]
Atlas atlas-z742sk-shard-0 [primary] estore>
  
```

3.8 Open the **estore** collection. You will see the products and users listed.

Cluster0 4.4.10 AWS Mumbai (ap-south-1)

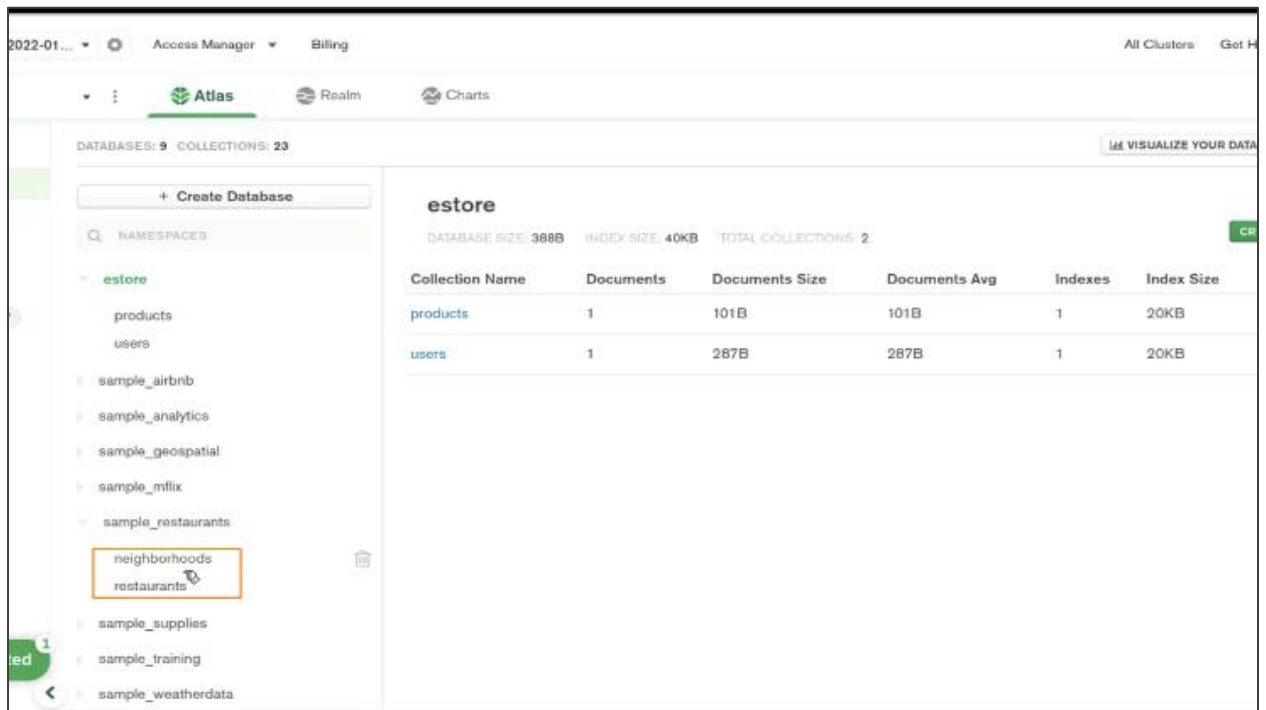
Overview Real Time Metrics **Collections** Search Profiler Performance Advisor Online Archive Command Line Tools

DATABASES: 9 COLLECTIONS: 23 VISUALIZE YOUR DATA REFRESH

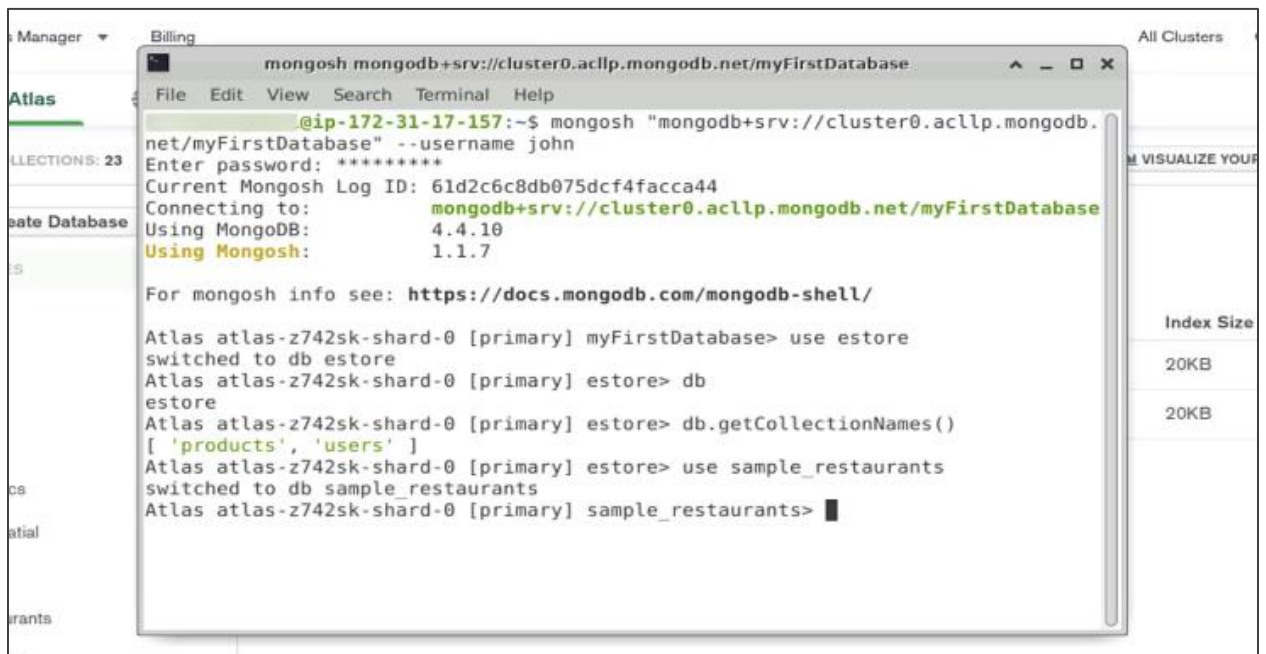
estore DATABASE SIZE: 388B INDEX SIZE: 40KB TOTAL COLLECTIONS: 2 CREATE COLLECTION

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
products	1	101B	101B	1	20KB	20KB
users	1	287B	287B	1	20KB	20KB

3.9 Select **Sample_restaurants** from the collections



3.10 Type **sample_restaurants** in the terminal to access the **sample_restaurants** database



Step 4: Execute commands

4.1 Execute the command below to get the collection names:

Db.getCollectionNames()

```

61d2af87f07b4a321e698a73#metrics/replicaSet/61d2b0ec424e2b0a95e3d02e/explorer/estore

Atlas Manager
Billing
Atlas
COLLECTIONS: 23
Create Database
Atlas atlas-z742sk-shard-0 [primary] myFirstDatabase> use estore
switched to db estore
Atlas atlas-z742sk-shard-0 [primary] estore> db
estore
Atlas atlas-z742sk-shard-0 [primary] estore> db.getCollectionNames()
[ 'products', 'users' ]
Atlas atlas-z742sk-shard-0 [primary] estore> use sample_restaurants
switched to db sample_restaurants
Atlas atlas-z742sk-shard-0 [primary] sample_restaurants> db
sample_restaurants
Atlas atlas-z742sk-shard-0 [primary] sample_restaurants> db.getCollectionNames()
[ 'neighborhoods', 'restaurants' ]
Atlas atlas-z742sk-shard-0 [primary] sample_restaurants>
  
```

4.2 Execute the **db.restaurants.find()** command to determine the documents inside the collection

```

Atlas Manager
Billing
Atlas
COLLECTIONS: 23
Create Database
Atlas atlas-z742sk-shard-0 [primary] sample_restaurants> db.restaurants.find()
{
  score: 12
},
{
  date: ISODate("2013-01-04T00:00:00.000Z"),
  grade: 'A',
  score: 11
},
{
  date: ISODate("2012-06-07T00:00:00.000Z"),
  grade: 'A',
  score: 6
},
{
  date: ISODate("2012-01-17T00:00:00.000Z"),
  grade: 'A',
  score: 8
}
],
name: "Bully'S Deli",
restaurant_id: '40361708'
}
Type "it" for more
Atlas atlas-z742sk-shard-0 [primary] sample_restaurants>
  
```


4.3 Enter the **estore** command once again and the database will switch to estore

```

mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
{
  date: ISODate("2013-01-04T00:00:00.000Z"),
  grade: 'A',
  score: 11
},
{
  date: ISODate("2012-06-07T00:00:00.000Z"),
  grade: 'A',
  score: 6
},
{
  date: ISODate("2012-01-17T00:00:00.000Z"),
  grade: 'A',
  score: 8
}
],
name: "Bully'S Deli",
restaurant_id: '40361708'
}
]
Type "it" for more
Atlas atlas-z742sk-shard-0 [primary] sample_restaurants> use estore
switched to db estore
Atlas atlas-z742sk-shard-0 [primary] estore>

```

4.4 Use **db.products.find()** to list the products

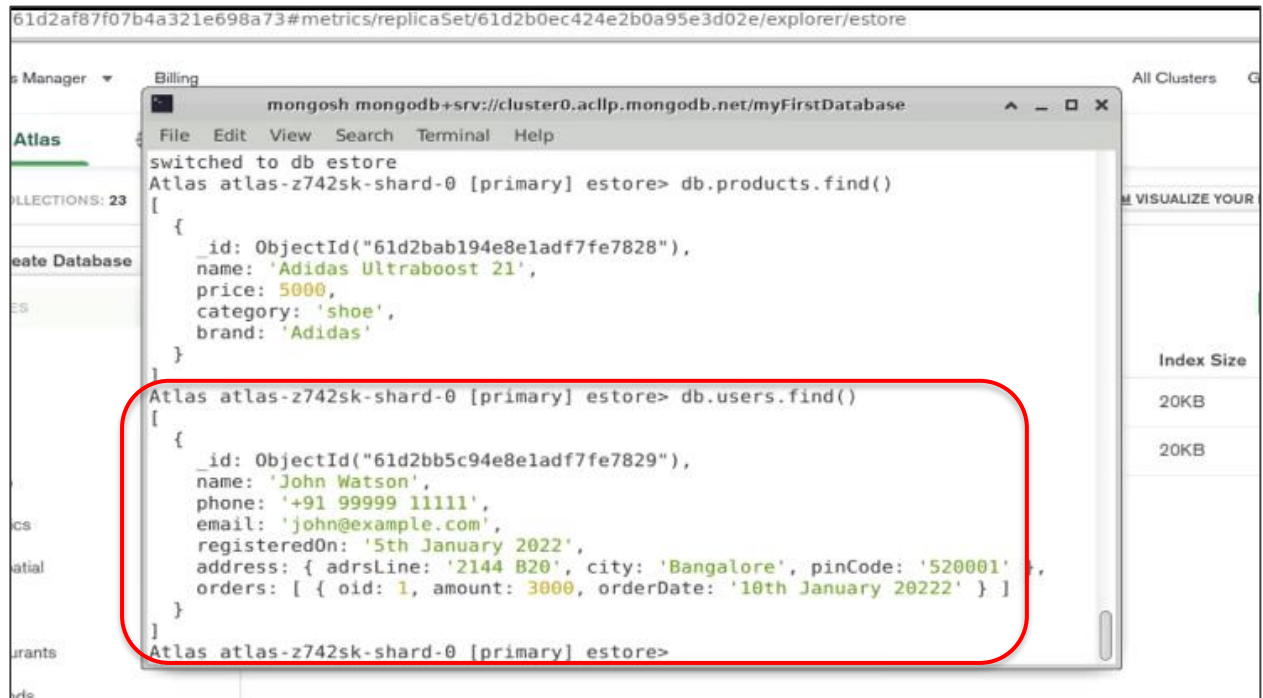
```

mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
{
  date: ISODate("2012-01-17T00:00:00.000Z"),
  grade: 'A',
  score: 8
},
{
  name: "Bully'S Deli",
  restaurant_id: '40361708'
}
]
Type "it" for more
Atlas atlas-z742sk-shard-0 [primary] sample_restaurants> use estore
switched to db estore
Atlas atlas-z742sk-shard-0 [primary] estore> db.products.find()
[
  {
    _id: ObjectId("61d2bab194e8eladf7fe7828"),
    name: 'Adidas Ultraboost 21',
    price: 5000,
    category: 'shoe',
    brand: 'Adidas'
  }
]
Atlas atlas-z742sk-shard-0 [primary] estore>

```


You can see that there is only one product listed in the terminal.

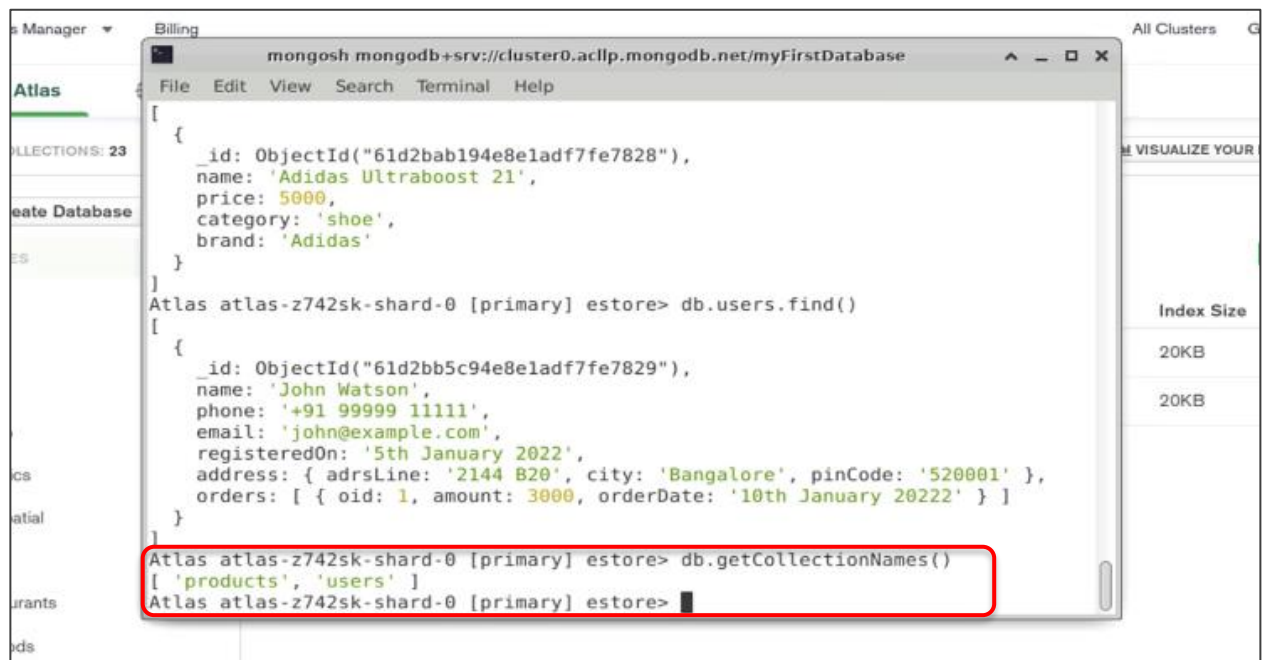
4.5 Similarly, use **db.users.find()** to determine the data



```

61d2af87f07b4a321e698a73#metrics/replicaSet/61d2b0ec424e2b0a95e3d02e/explorer/estore:
Atlas
COLLECTIONS: 23
Create Database
Atlas atlas-z742sk-shard-0 [primary] estore> db.products.find()
[
  {
    _id: ObjectId("61d2bab194e8e1adf7fe7828"),
    name: 'Adidas Ultraboost 21',
    price: 5000,
    category: 'shoe',
    brand: 'Adidas'
  }
]
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.find()
[
  {
    _id: ObjectId("61d2bb5c94e8e1adf7fe7829"),
    name: 'John Watson',
    phone: '+91 99999 11111',
    email: 'john@example.com',
    registeredOn: '5th January 2022',
    address: { adrslne: '2144 B20', city: 'Bangalore', pinCode: '520001' },
    orders: [ { oid: 1, amount: 3000, orderDate: '10th January 2022' } ]
  }
]
Atlas atlas-z742sk-shard-0 [primary] estore>
  
```

4.6 Get the collection names again using the **db.getCollectionNames()** command



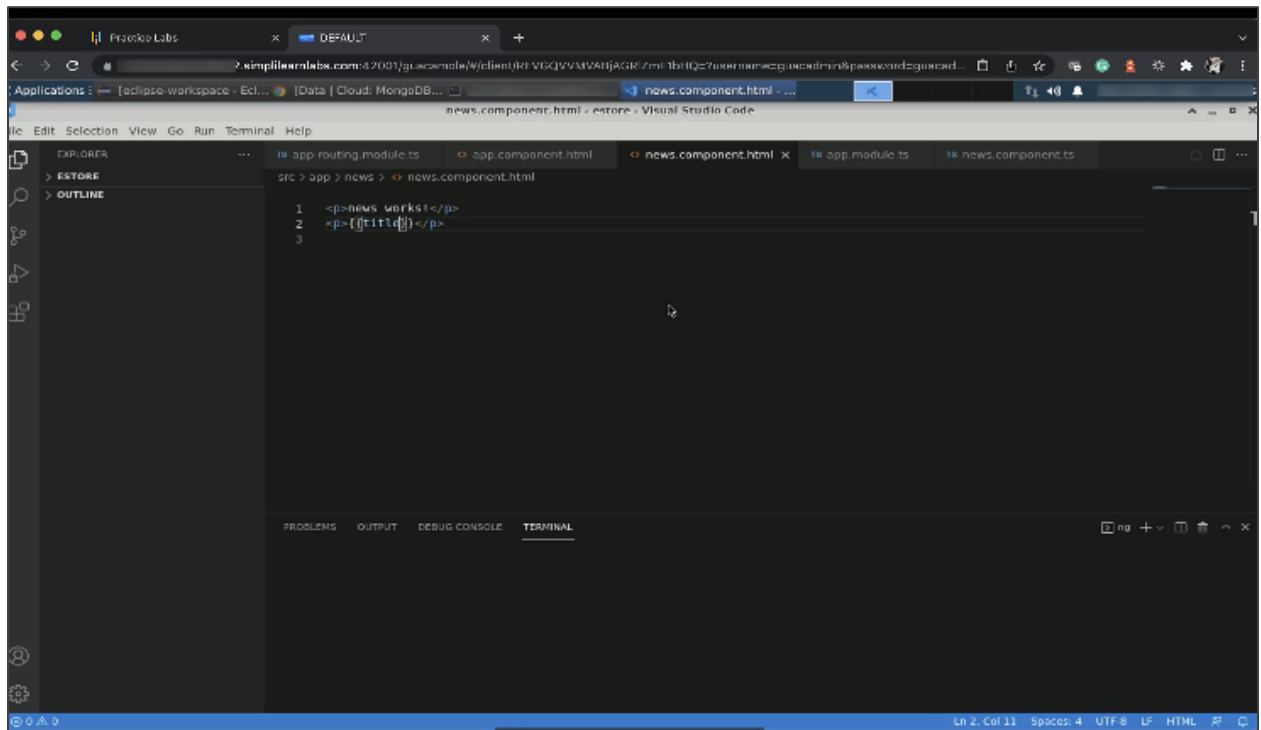
The screenshot shows a MongoDB terminal window with the following content:

```
mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
Atlas atlas-z742sk-shard-0 [primary] estore> db.getCollectionNames()
[ 'products', 'users' ]
```

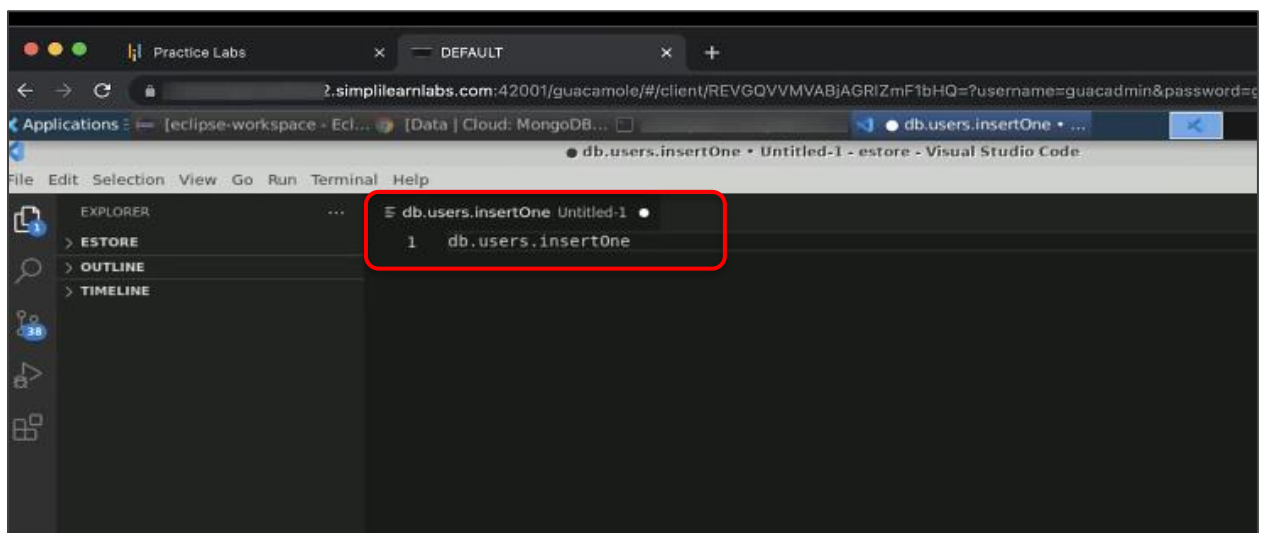
The command `db.getCollectionNames()` is highlighted with a red box, and its output is `['products', 'users']`.

Step 5: Create a document

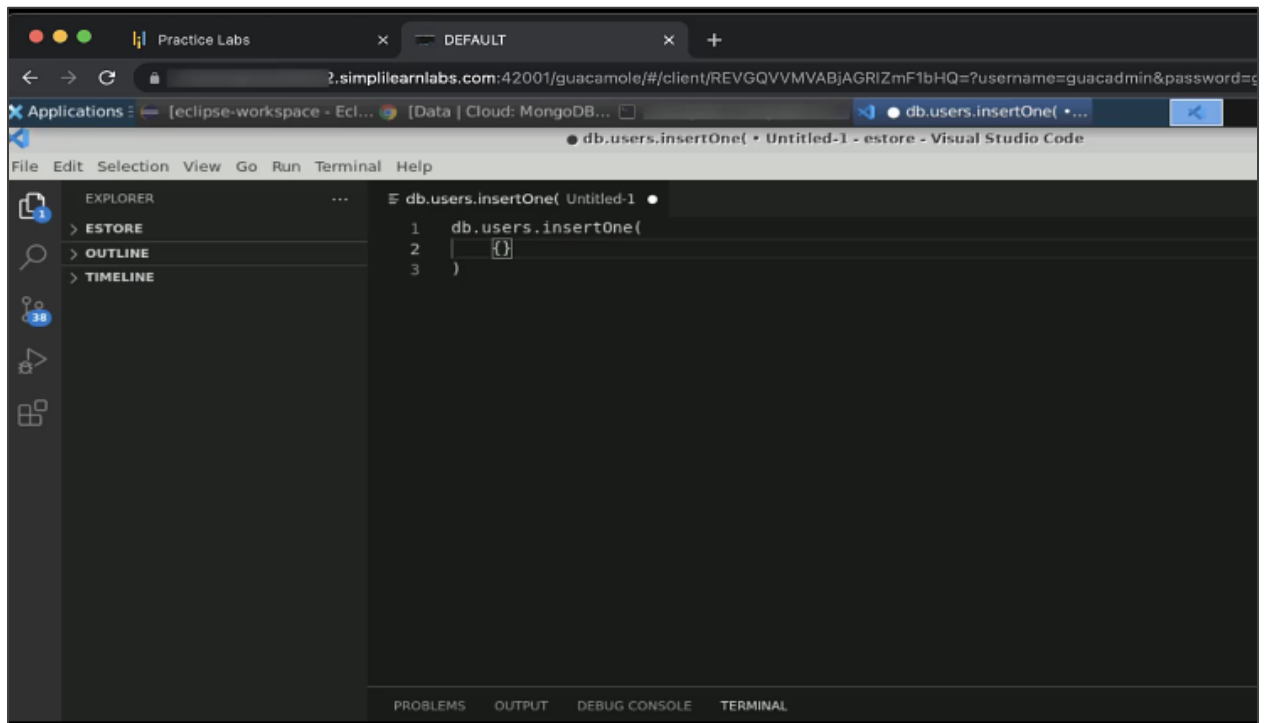
5.1 Open the Visual Studio Code editor



5.2 Create a new file and use the **insertOne** method to insert a document inside the database



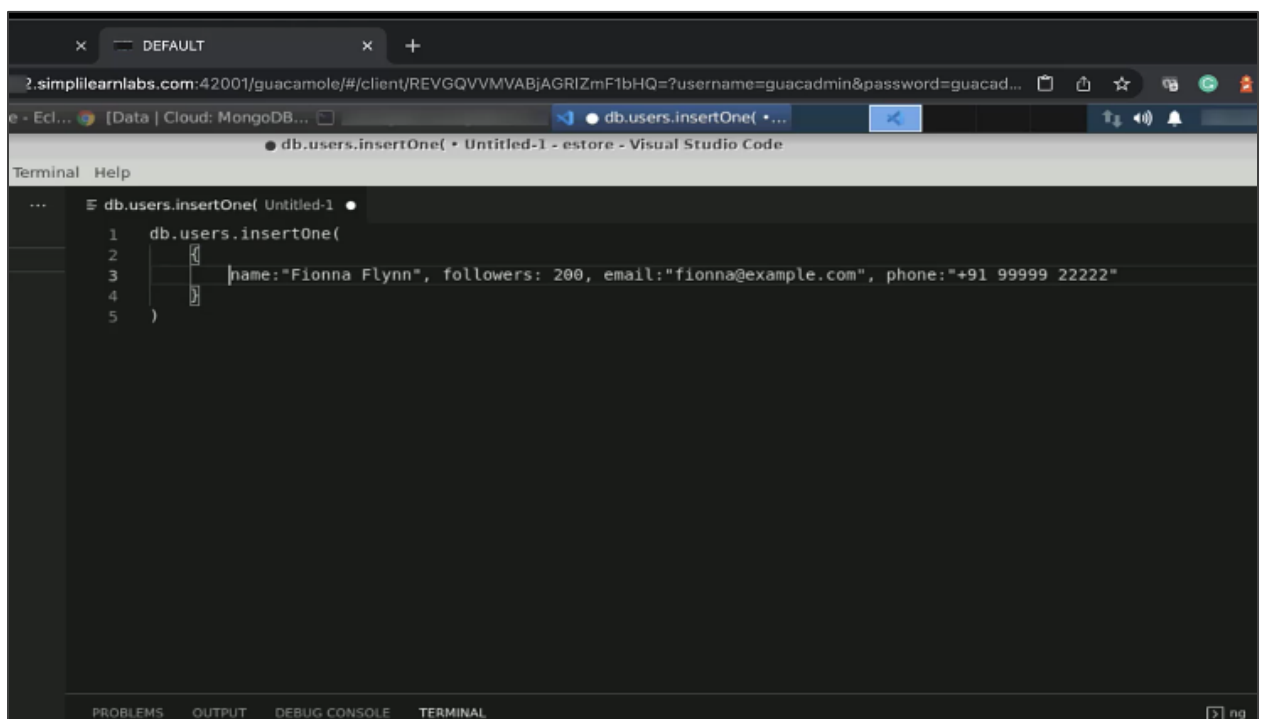
5.3 Create a document as per the JSON standards



The screenshot shows the Visual Studio Code interface with a MongoDB connection established. The Explorer panel on the left shows the 'ESTORE' collection. The main editor area displays the following code:

```
1 db.users.insertOne(
2   {}
3 )
```

5.4 Give the attributes in key-value pairs, consisting of **name**, **followers**, **email**, and **phone**



The screenshot shows the Visual Studio Code interface with a MongoDB connection established. The Explorer panel on the left shows the 'ESTORE' collection. The main editor area displays the following code:

```
1 db.users.insertOne(
2   {
3     name: "Fionna Flynn", followers: 200, email: "fionna@example.com", phone: "+91 99999 22222"
4   }
5 )
```

5.5 Add an **address** attribute to represent another object

The screenshot shows a web browser at `simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guacad...` and a Visual Studio Code editor. The editor has a file named `db.users.insertOne(` with the following code:

```
1 db.users.insertOne(
2   {
3     name:"Fionna Flynn", followers: 200, email:"fionna@example.com", phone:"+91 99999 22222",
4     address: {adrsLine:"77 E Country Homes North", city:"Bangalore", pincode: 520001}
5   }
6 )
```

5.6 Copy the snippet

The screenshot shows the same Visual Studio Code editor with the same code snippet. The terminal window at the bottom shows the following output:

```
main.js      main      50.86 kB
runtime.js   runtime   6.85 kB

3 unchanged chunks

Build at: 2021-12-20T10:15:06.878Z   Hash: 05bfa806acf8f8f0   Time: 460ms
✓ Compiled successfully.

Session contents restored from 12/20/2021 at 10:50:05 AM
-172-31-37-157:~/Downloads/estore$
```

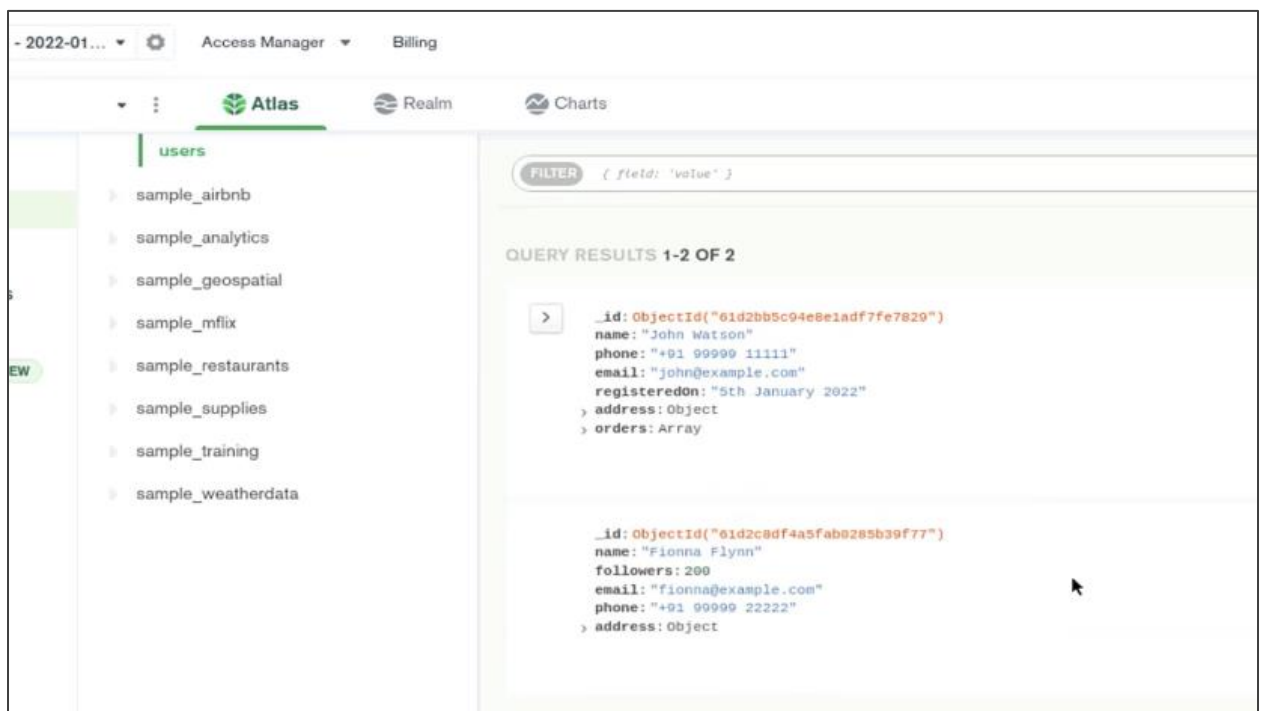
5.7 Paste it into the terminal in **MongoDB Shell** and press **Enter**

```

mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
phone: '+91 99999 11111',
email: 'john@example.com',
registeredOn: '5th January 2022',
address: { adrsLine: '2144 B20', city: 'Bangalore', pinCode: '520001' },
orders: [ { oid: 1, amount: 3000, orderDate: '10th January 2022' } ]
}
]
Atlas atlas-z742sk-shard-0 [primary] estore> db.getCollectionNames()
[ 'products', 'users' ]
Atlas atlas-z742sk-shard-0 [primary] estore> db
estore
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.insertOne(
...   {
...     name:"Fionna Flynn", followers: 200, email:"fionna@example.com", p
phone:"+91 99999 22222",
...     address: {adrsLine:"77 E Country Homes North", city:"Bangalore", p
pincode: 520001}
...   }
... )
{
  acknowledged: true,
  insertedId: ObjectId("61d2c8df4a5fab0285b39f77")
}
Atlas atlas-z742sk-shard-0 [primary] estore>

```

5.8 Refresh the **users** collection. You can see that another document was created.



5.9 Paste the same snippet in Visual Studio Code

```

1 db.users.insertOne(
2   {
3     name: "Fionna Flynn", followers: 280, email: "fionna@example.com", phone: "+91 99999 22222",
4     address: {adrsLine: "77 E Country Homes North", city: "Bangalore", pincode: 528001}
5   }
6 )
7
8 db.users.insertOne(
9   {
10    name: "Fionna Flynn", followers: 280, email: "fionna@example.com", phone: "+91 99999 22222",
11    address: {adrsLine: "77 E Country Homes North", city: "Bangalore", pincode: 528001}
12  }
13 )
  
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

main.js	runtime.js	main	runtime	50.88 KB	6.85 KB
3 unchanged chunks					

Build at: 2021-12-20T10:18:06.878Z Hash: 05bfa800acf8f8f0 Time: 460ms

✓ Compiled successfully.

Session contents restored from 12/20/2021 at 10:50:05 AM

-172-31-17-157:~/Downloads/estore\$

5.10 Use the **db.users.InsertMany()** method to pass an array of documents

```

1 db.users.insertOne(
2   {
3     name: "Fionna Flynn", followers: 280, email: "fionna@example.com", phone: "+91 99999 22222",
4     address: {adrsLine: "77 E Country Homes North", city: "Bangalore", pincode: 528001}
5   }
6 )
7
8 db.users.insertMany(
9   [
10    {
11      name: "Fionna Flynn", followers: 280, email: "fionna@example.com", phone: "+91 99999 22222",
12      address: {adrsLine: "77 E Country Homes North", city: "Bangalore", pincode: 528001}
13    }
14  ]
15 )
  
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

main.js	runtime.js	main	runtime	50.88 KB	6.85 KB
3 unchanged chunks					

Build at: 2021-12-20T10:18:06.878Z Hash: 05bfa800acf8f8f0 Time: 460ms

✓ Compiled successfully.

Session contents restored from 12/20/2021 at 10:50:05 AM

-172-31-17-157:~/Downloads/estore\$

Step 6: Insert more documents

6.1 Insert some more documents with different values

```

3      name:"Fionna Flynn", followers: 200, email:"fionna@example.com", phone:"+91 99999 22222",
4      address: {adrsLine:"77 E Country Homes North", city:"Bangalore", pincode: 520001}
5    }
6  }
7
8  db.users.insertMany(
9    [
10     {
11       name:"Dave", followers: 300, email:"dave@example.com", phone:"+91 99999 33333",
12       address: {adrsLine:"80 E Country Homes North", city:"Delhi", pincode: 110001}
13     },
14     {
15       name:"Kim", followers: 400, email:"kim@example.com", phone:"+91 99999 44444",
16       address: {adrsLine:"90 E Country Homes North", city:"Bangalore", pincode: 520001}
17     },
18     {
19       name:"Leo", followers: 500, email:"leo@example.com", phone:"+91 99999 55555",
20       address: {adrsLine:"100 E Country Homes North", city:"Chennai", pincode: 320001}
21     }
22   ]
23 )
24
25

```

6.2 Copy the code

```

5    }
6  }
7
8  db.users.insertMany(
9    [
10     {
11       name:"Dave", followers: 300, email:"dave@example.com", phone:"+91 99999 33333",
12       address: {adrsLine:"80 E Country Homes North", city:"Delhi", pincode: 110001}
13     },
14     {
15       name:"Kim", followers: 400, email:"kim@example.com", phone:"+91 99999 44444",
16       address: {adrsLine:"90 E Country Homes North", city:"Bangalore", pincode: 520001}
17     },
18     {
19       name:"Leo", followers: 500, email:"leo@example.com", phone:"+91 99999 55555",
20       address: {adrsLine:"100 E Country Homes North", city:"Chennai", pincode: 320001}
21     }
22   ]
23 )
24
25

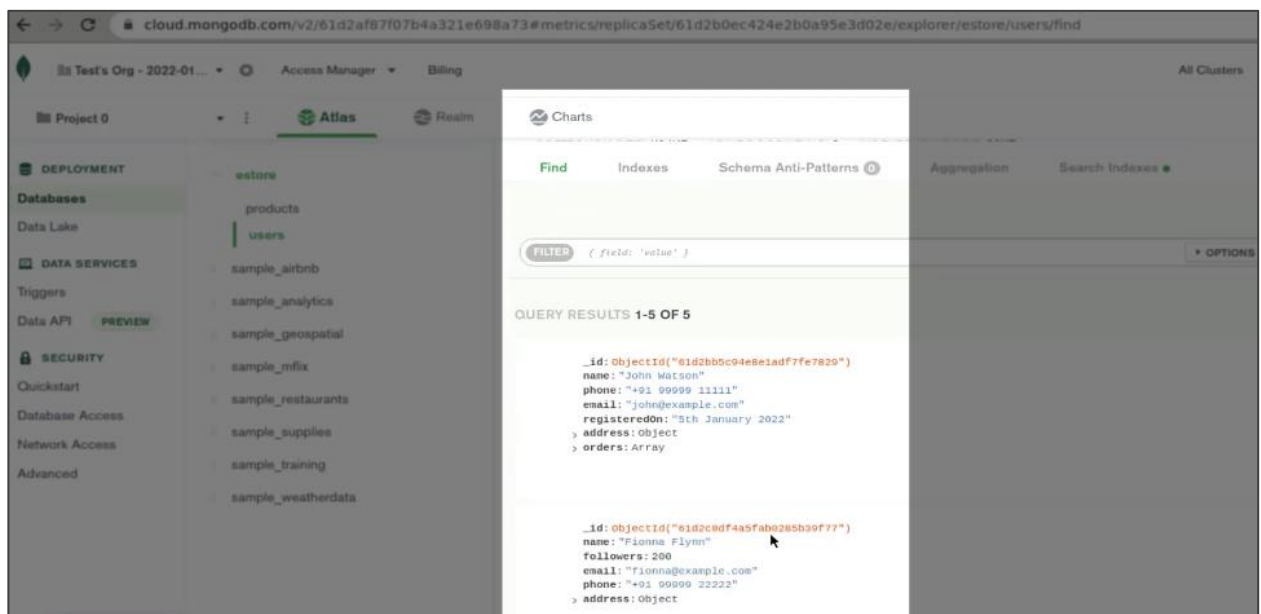
```

6.3 Paste it into the terminal and execute it

```

mongosh mongodb+srv://cluster0.aclp.mongodb.net/myFirstDatabase
...
name:"Kim", followers: 400, email:"kim@example.com", phone:"+91
99999 44444",
...
address: {adrsLine:"90 E Country Homes North", city:"Bangalore",
pincode: 520001}
...
{
name:"Leo", followers: 500, email:"leo@example.com", phone:"+91
99999 55555",
...
address: {adrsLine:"100 E Country Homes North", city:"Chennai",
pincode: 320001}
...
}
...
}
acknowledged: true,
insertedIds: {
'0': ObjectId("61d2c99d4a5fab0285b39f78"),
'1': ObjectId("61d2c99d4a5fab0285b39f79"),
'2': ObjectId("61d2c99d4a5fab0285b39f7a")
}
}
Atlas atlas-z742sk-shard-0 [primary] estore>
3Z - Hash: 05bfa800acf8f8f0 - Time: 460ms
  
```

6.4 Refresh the **users** collection page. You can see more documents updated in the database.



6.5 Now, execute the find function on users using the **db.users.find()** function

```

mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
> use users
switched to db users
> find()
{
  city: 'Bangalore',
  pincode: 520001
},
{
  _id: ObjectId("61d2c99d4a5fab0285b39f78"),
  name: 'Dave',
  followers: 300,
  email: 'dave@example.com',
  phone: '+91 99999 33333',
  address: {
    adrLine: '80 E Country Homes North',
    city: 'Delhi',
    pincode: 110001
  }
},
{
  _id: ObjectId("61d2c99d4a5fab0285b39f79"),
  name: 'Kim',
  followers: 400,
  email: 'kim@example.com',
  phone: '+91 99999 44444',
  address: {
    adrLine: '90 E Country Homes North',
    city: 'Bangalore',
    pincode: 520001
  }
}

```

6.6 Use the **db.users.find(city: "Bangalore")** query to find users with the city as Bengaluru

```

db.users.insertMany([
  {
    name: 'Dave', followers: 300, email: 'dave@example.com', phone: '+91 99999 33333',
    address: {adrLine: '80 E Country Homes North', city: 'Delhi', pincode: 110001}
  },
  {
    name: 'Kim', followers: 400, email: 'kim@example.com', phone: '+91 99999 44444',
    address: {adrLine: '90 E Country Homes North', city: 'Bangalore', pincode: 520001}
  },
  {
    name: 'Leo', followers: 500, email: 'leo@example.com', phone: '+91 99999 55555',
    address: {adrLine: '100 E Country Homes North', city: 'Chennai', pincode: 520001}
  }
])

```

```

db.users.find({city: 'Bangalore'})

```

6.7 Execute the query in the terminal

The screenshot shows a MongoDB terminal window with the title 'mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase'. The terminal displays the following JSON data for users in Bangalore:

```

{
  email: 'kim@example.com',
  phone: '+91 99999 44444',
  address: {
    adrsLine: '90 E Country Homes North',
    city: 'Bangalore',
    pincode: 520001
  }
},
{
  _id: ObjectId("61d2c99d4a5fab0285b39f7a"),
  name: 'Leo',
  followers: 500,
  email: 'leo@example.com',
  phone: '+91 99999 55555',
  address: {
    adrsLine: '100 E Country Homes North',
    city: 'Chennai',
    pincode: 320001
  }
}

```

The terminal prompt is 'Atlas atlas-z742sk-shard-0 [primary] estore> db.users.find({city: "Bangalore"})'. The output shows the first user (Kim) with city 'Bangalore'.

6.8 Change the key value to address.city

The screenshot shows a Visual Studio Code editor window with the title 'db.users.insertOne() - estore - Visual Studio Code'. The editor displays the following JSON data for users in Bangalore:

```

{
  name: 'Dave', followers: 300, email: 'dave@example.com', phone: '+91 99999 33333',
  address: {adrsLine: '88 E Country Homes North', city: 'Delhi', pincode: 110001}
},
{
  name: 'Kim', followers: 400, email: 'kim@example.com', phone: '+91 99999 44444',
  address: {adrsLine: '90 E Country Homes North', city: 'Bangalore', pincode: 520001}
},
{
  name: 'Leo', followers: 500, email: 'leo@example.com', phone: '+91 99999 55555',
  address: {adrsLine: '100 E Country Homes North', c: 'Chennai', pincode: 320001}
}

```

The terminal prompt is 'db.users.find({address.city: 'Bangalore'})'. The output shows the first user (Kim) with city 'Bangalore'.

6.9 Copy the query

```

1  db.users.insertMany([
2    {
3      name: "Dave", followers: 300, email: "dave@example.com", phone: "+91 99999 33333",
4      address: { adrLine: "88 E Country Homes North", city: "Delhi", pincode: 110001 }
5    },
6    {
7      name: "Kim", followers: 400, email: "kim@example.com", phone: "+91 99999 44444",
8      address: { adrLine: "90 E Country Homes North", city: "Bangalore", pincode: 520001 }
9    },
10   {
11     name: "Leo", followers: 500, email: "leo@example.com", phone: "+91 99999 55555",
12     address: { adrLine: "180 E Country Homes North", city: "Chennai", pincode: 320001 }
13   }
14 ])
15
16 db.users.find({address.city: "Bangalore"})

```

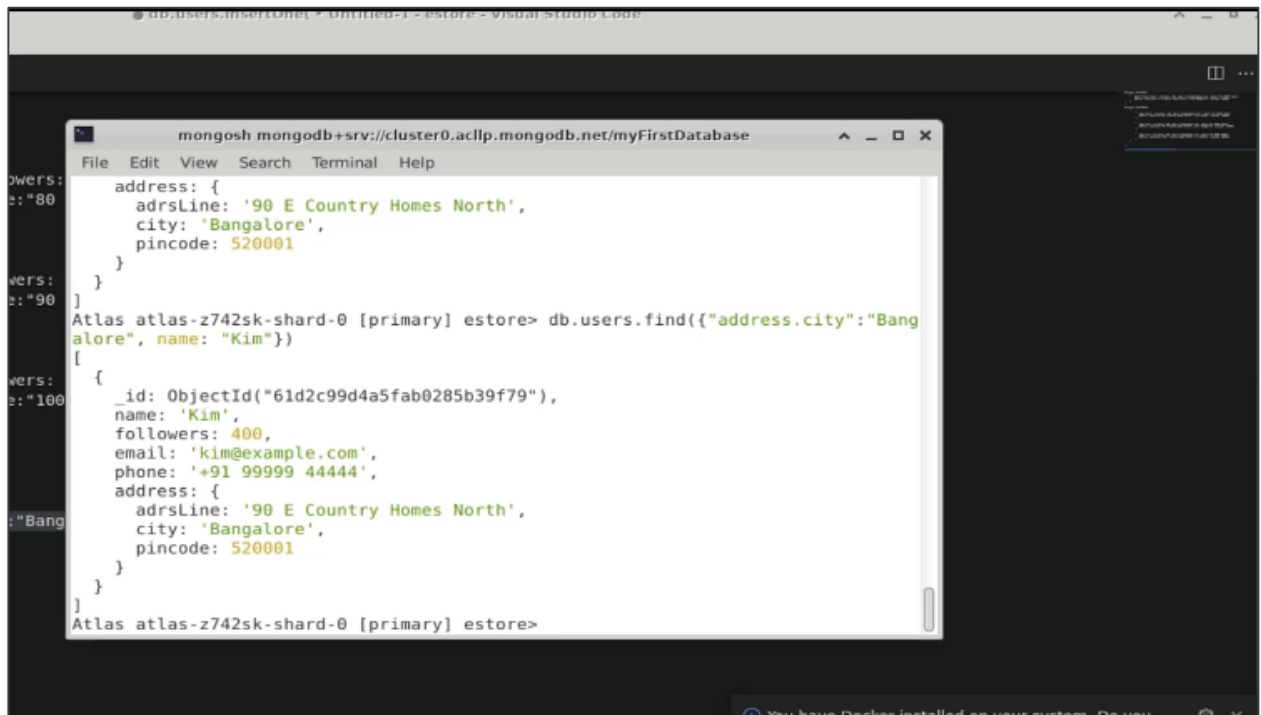
6.10 Paste it into the terminal

```

name: 'Fionna Flynn',
followers: 200,
email: 'fionna@example.com',
phone: '+91 99999 22222',
address: {
  adrLine: '77 E Country Homes North',
  city: 'Bangalore',
  pincode: 520001
},
_id: ObjectId("61d2c99d4a5fab0285b39f79"),
name: 'Kim',
followers: 400,
email: 'kim@example.com',
phone: '+91 99999 44444',
address: {
  adrLine: '90 E Country Homes North',
  city: 'Bangalore',
  pincode: 520001
}
}
Atlas atlas-z742sk-shard-0 [primary] estore>

```

6.11 Add a find statement to list the results of user Kim in Bengaluru



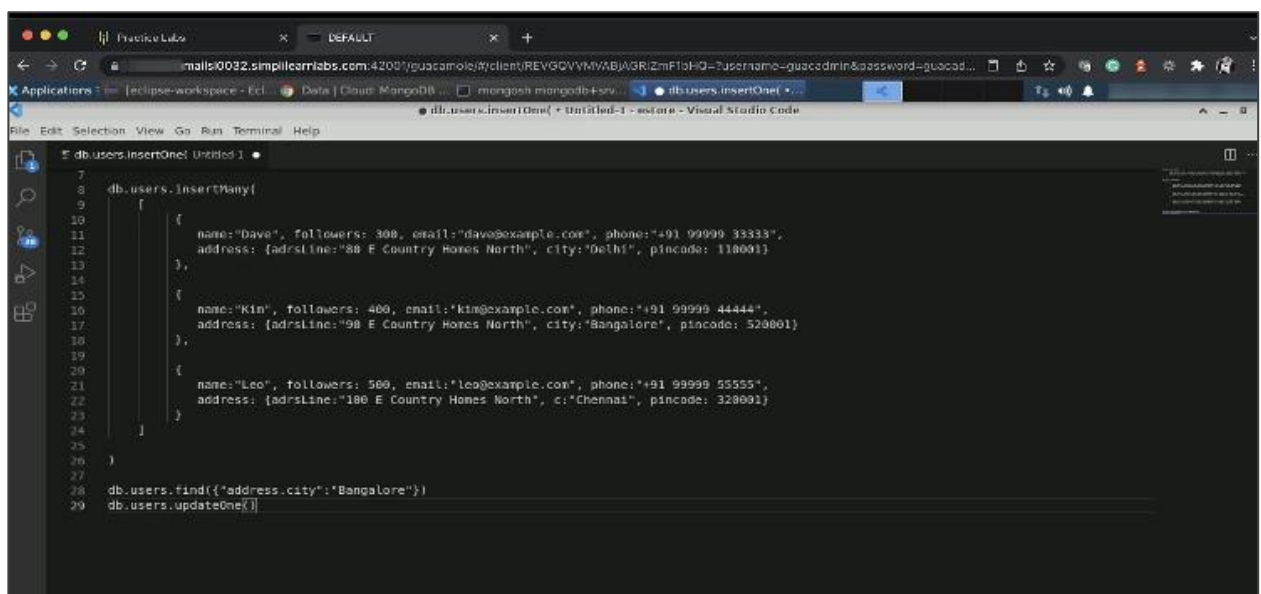
```

mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.find({"address.city":"Bangalore", "name": "Kim"})
[
  {
    _id: ObjectId("61d2c99d4a5fab0285b39f79"),
    name: 'Kim',
    followers: 400,
    email: 'kim@example.com',
    phone: '+91 99999 44444',
    address: {
      adrLine: '90 E Country Homes North',
      city: 'Bangalore',
      pincode: 520001
    }
  }
]
Atlas atlas-z742sk-shard-0 [primary] estore>

```

Step 7: Perform update operations

7.1 Perform the update operation using the `updateOne()` function



```

db.users.insertMany([
  {
    name: "Dave", followers: 300, email: "dave@example.com", phone: "+91 99999 33333",
    address: {adrLine: "80 E Country Homes North", city: "Delhi", pincode: 110001}
  },
  {
    name: "Kim", followers: 400, email: "kim@example.com", phone: "+91 99999 44444",
    address: {adrLine: "90 E Country Homes North", city: "Bangalore", pincode: 520001}
  },
  {
    name: "Leo", followers: 500, email: "leo@example.com", phone: "+91 99999 55555",
    address: {adrLine: "100 E Country Homes North", city: "Chennai", pincode: 320001}
  }
])

db.users.find({'address.city': 'Bangalore'})
db.users.updateOne({})

```


7.2 Enter conditions to update, such as **email**

```

7
8 db.users.insertMany(
9   [
10     {
11       name:"Dave", followers: 300, email:"dave@example.com", phone:"+91 99999 33333",
12       address: {adrsLine:"80 E Country Homes North", city:"Delhi", pincode: 110001}
13     },
14     {
15       name:"Kim", followers: 400, email:"kim@example.com", phone:"+91 99999 44444",
16       address: {adrsLine:"90 E Country Homes North", city:"Bangalore", pincode: 520001}
17     },
18     {
19       name:"Leo", followers: 500, email:"leo@example.com", phone:"+91 99999 55555",
20       address: {adrsLine:"100 E Country Homes North", c:"Chennai", pincode: 320001}
21     }
22   ]
23 )
24
25
26
27
28 db.users.find({"address.city":"Bangalore"})
29 db.users.updateOne(
30   [{"email": "leo@example.com"}]
31 )

```

7.3 Set the details to update using the **set** variable that specifies the followers and pincode (zip code)

```

7
8 db.users.insertMany(
9   [
10     {
11       name:"Dave", followers: 300, email:"dave@example.com", phone:"+91 99999 33333",
12       address: {adrsLine:"80 E Country Homes North", city:"Delhi", pincode: 110001}
13     },
14     {
15       name:"Kim", followers: 400, email:"kim@example.com", phone:"+91 99999 44444",
16       address: {adrsLine:"90 E Country Homes North", city:"Bangalore", pincode: 520001}
17     },
18     {
19       name:"Leo", followers: 500, email:"leo@example.com", phone:"+91 99999 55555",
20       address: {adrsLine:"100 E Country Homes North", c:"Chennai", pincode: 320001}
21     }
22   ]
23 )
24
25
26
27
28 db.users.find({"address.city":"Bangalore"})
29 db.users.updateOne(
30   [{"email": "leo@example.com"}]
31 )

```


7.4 Copy the snippet

```

7
8 db.users.insertMany([
9
10   {
11     name: "Dave", followers: 300, email: "dave@example.com", phone: "+91 99999 33333",
12     address: {adrLine: "88 E Country Homes North", city: "Delhi", pincode: 110001}
13   },
14   {
15     name: "Kim", followers: 400, email: "kim@example.com", phone: "+91 99999 44444",
16     address: {adrLine: "90 E Country Homes North", city: "Bangalore", pincode: 520001}
17   },
18   {
19     name: "Leo", followers: 500, email: "leo@example.com", phone: "+91 99999 55555",
20     address: {adrLine: "100 E Country Homes North", city: "Chennai", pincode: 320001}
21   }
22 ])
23
24
25
26
27
28 db.users.find({"address.city": "Bangalore"})
29 db.users.updateOne(
30   {email: "leo@example.com"},
31   {
32     $set: {followers: 3700, "address.pincode": 520001}
33   }
34 )

```

7.5 Paste it and press **Enter**. You can see that the acknowledgment is **true**.

```

mongosh mongodb+srv://cluster0.ac1p.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
followers: 400,
line: "88 email: 'kim@example.com',
phone: '+91 99999 44444',
line: "90 address: {
adrLine: '90 E Country Homes North',
city: 'Bangalore',
pincode: 520001
}
}
}
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.updateOne(
... {email: "leo@example.com"},
... {
... $set: {followers: 3700, "address.pincode": 520001}
... }
... }
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-z742sk-shard-0 [primary] estore>

```

7.6 Write **db.deleteOne()** and add a condition to delete one of the documents

```

mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
> 1 | db.users.find({name: "Leo"})
  2 |
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.find({name: "Leo"})
[
  {
    _id: ObjectId("61d2c99d4a5fab0285b39f7a"),
    name: 'Leo',
    followers: 3700,
    email: 'leo@example.com',
    phone: '+91 99999 55555',
    address: {
      adrsLine: '100 E Country Homes North',
      city: 'Chennai',
      pincode: 520001
    }
  }
]
Atlas atlas-z742sk-shard-0 [primary] estore> db.deleteOne({email: "leo@example.com"})
TypeError: db.deleteOne is not a function
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.deleteOne({email: "leo@example.com"})

```

7.7 Update the query as **db.users.deleteOne()** and press Enter

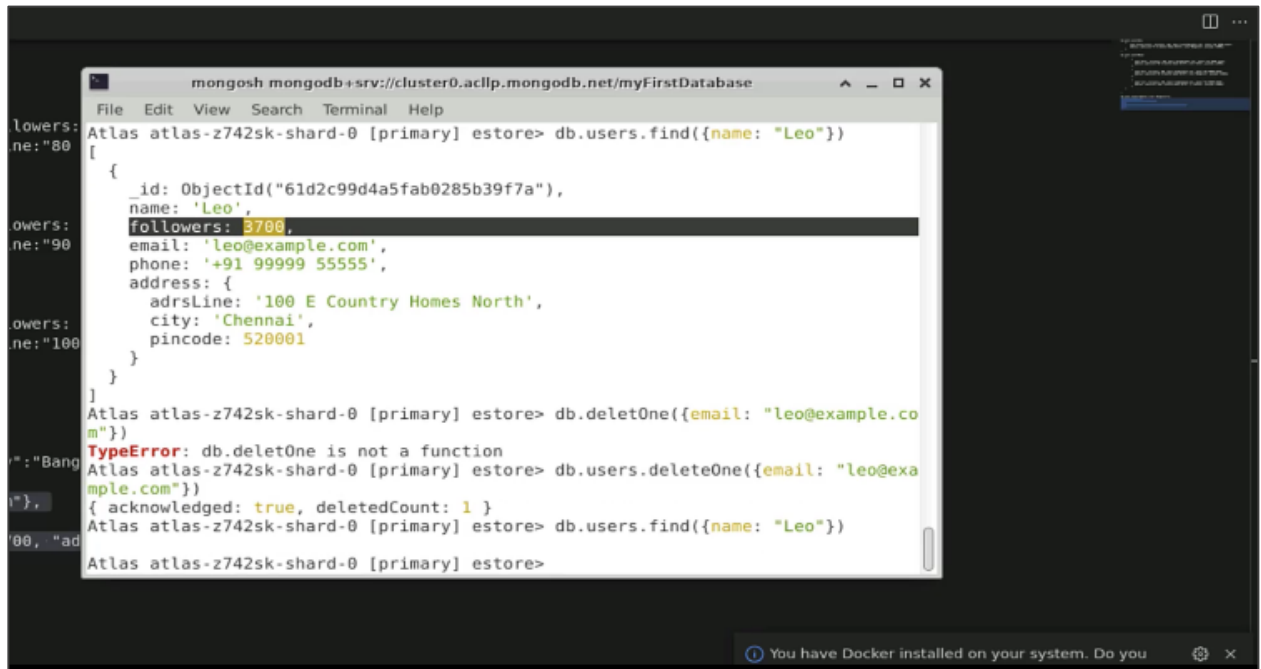
```

mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
2 |
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.find({name: "Leo"})
[
  {
    _id: ObjectId("61d2c99d4a5fab0285b39f7a"),
    name: 'Leo',
    followers: 3700,
    email: 'leo@example.com',
    phone: '+91 99999 55555',
    address: {
      adrsLine: '100 E Country Homes North',
      city: 'Chennai',
      pincode: 520001
    }
  }
]
Atlas atlas-z742sk-shard-0 [primary] estore> db.deleteOne({email: "leo@example.com"})
TypeError: db.deleteOne is not a function
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.deleteOne({email: "leo@example.com"})
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-z742sk-shard-0 [primary] estore>

```

You can see that the acknowledgment is **true** and one of the documents is deleted.

7.8 Execute another command named **Leo**



```

mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.find({name: "Leo"})
[
  {
    _id: ObjectId("61d2c99d4a5fab0285b39f7a"),
    name: 'Leo',
    followers: 3700,
    email: 'leo@example.com',
    phone: '+91 99999 55555',
    address: {
      adrsLine: '100 E Country Homes North',
      city: 'Chennai',
      pincode: 520001
    }
  }
]
Atlas atlas-z742sk-shard-0 [primary] estore> db.deleteOne({email: "leo@example.com"})
TypeError: db.deleteOne is not a function
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.deleteOne({email: "leo@example.com"})
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.find({name: "Leo"})
Atlas atlas-z742sk-shard-0 [primary] estore>
  
```

No such document is available.

These CRUD operations can be executed and will remain the same for any API. This applies to programming languages as well, for which these methods and functions remain the same.

By following these steps, you have successfully performed CRUD operations with Mongo Shell by installing mongosh and performing update operations.