Ex.No : 6	MongoDB Atlas
06.04.2021	

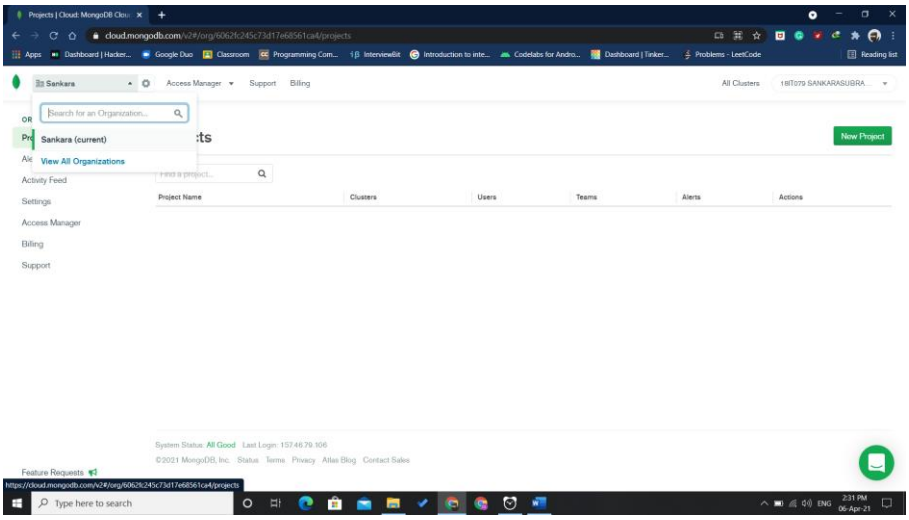
Aim

To create a cluster in mongodb atlas and access database via node.js.

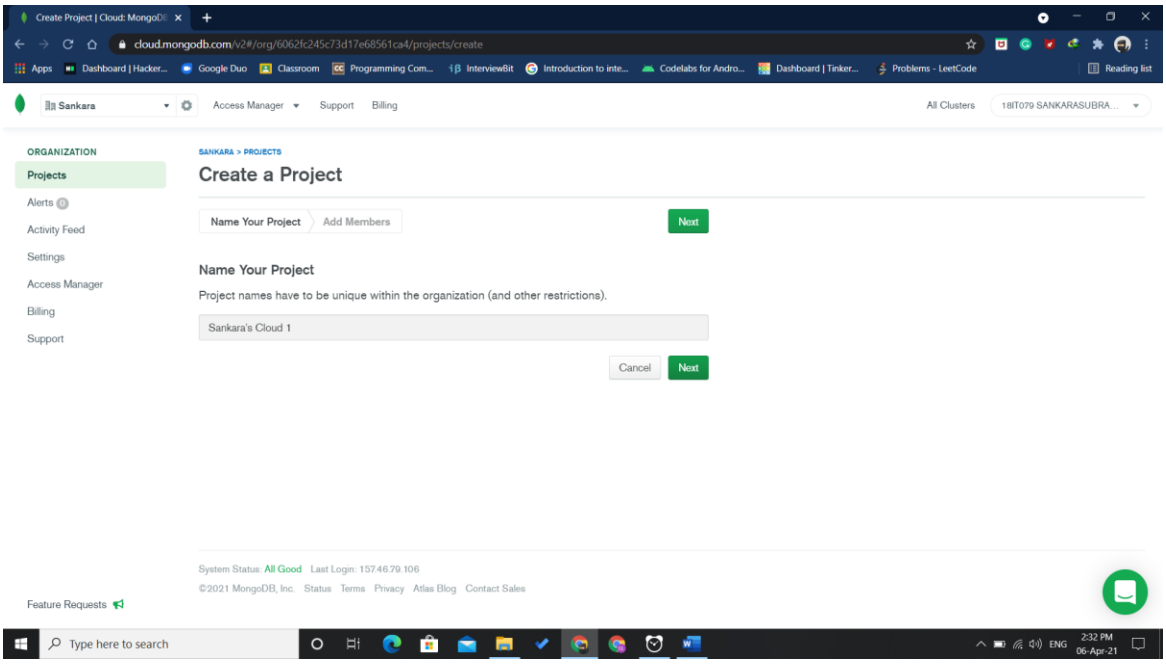
Procedure

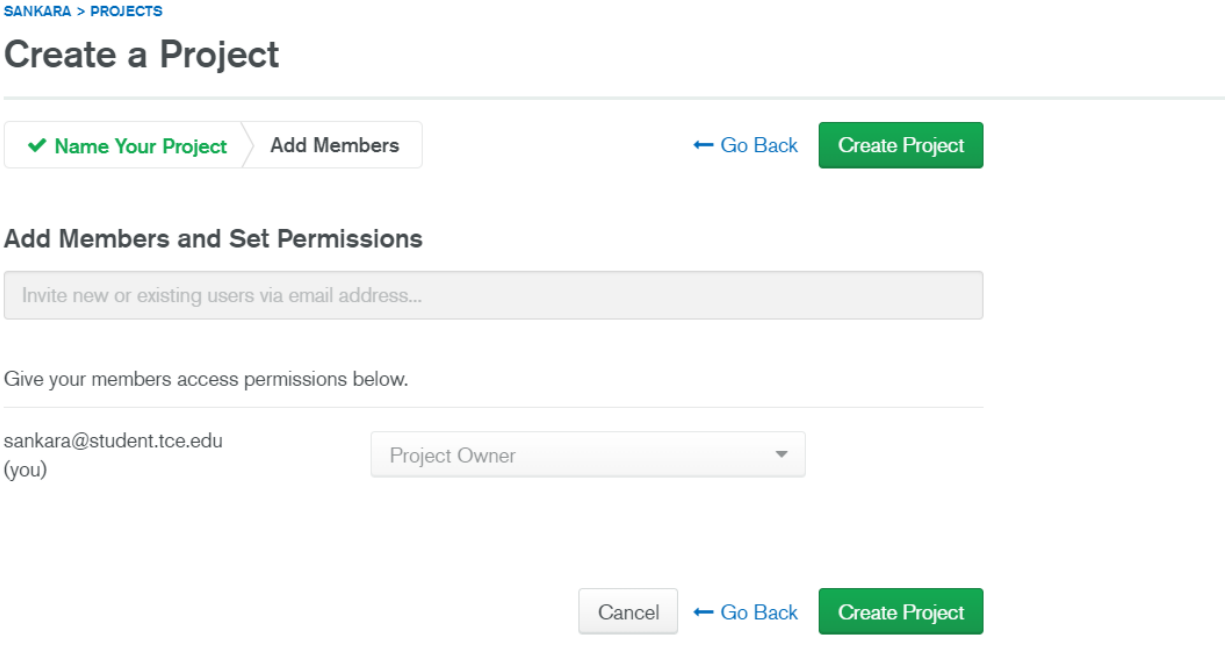
1. Create an account on https://cloud.mongodb.com/ and login.

2. Create an organization.

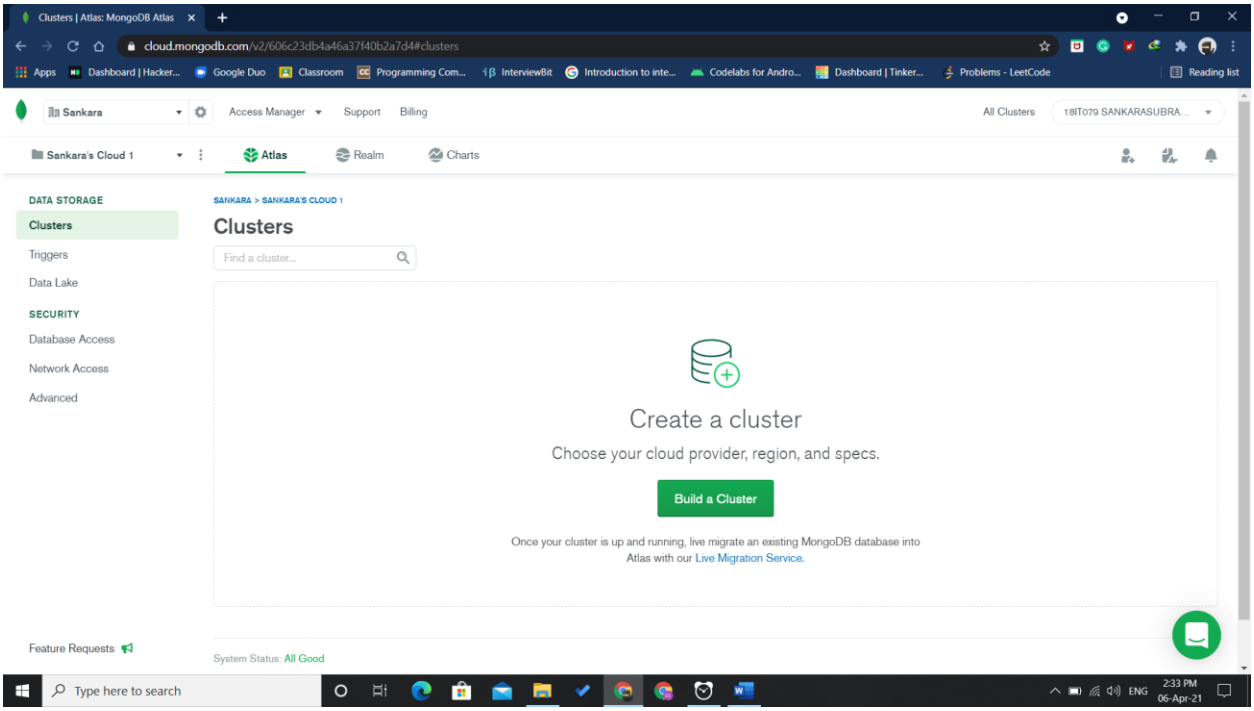


3. Create a new project.

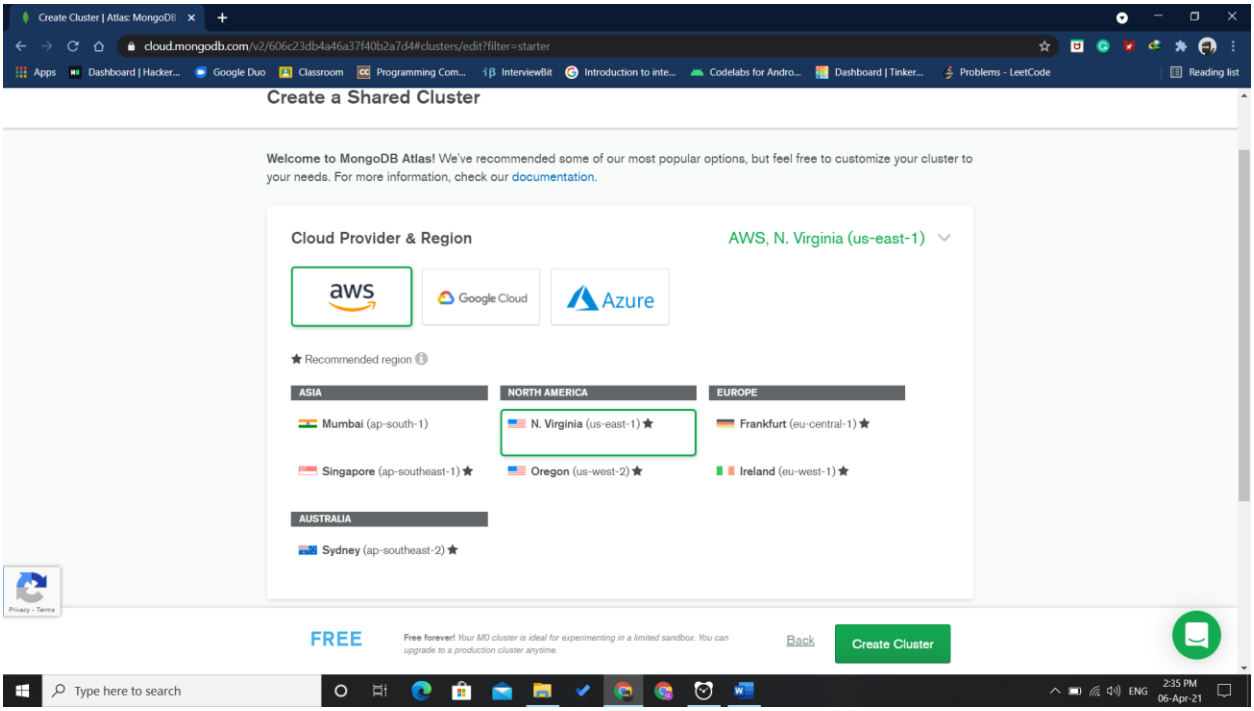


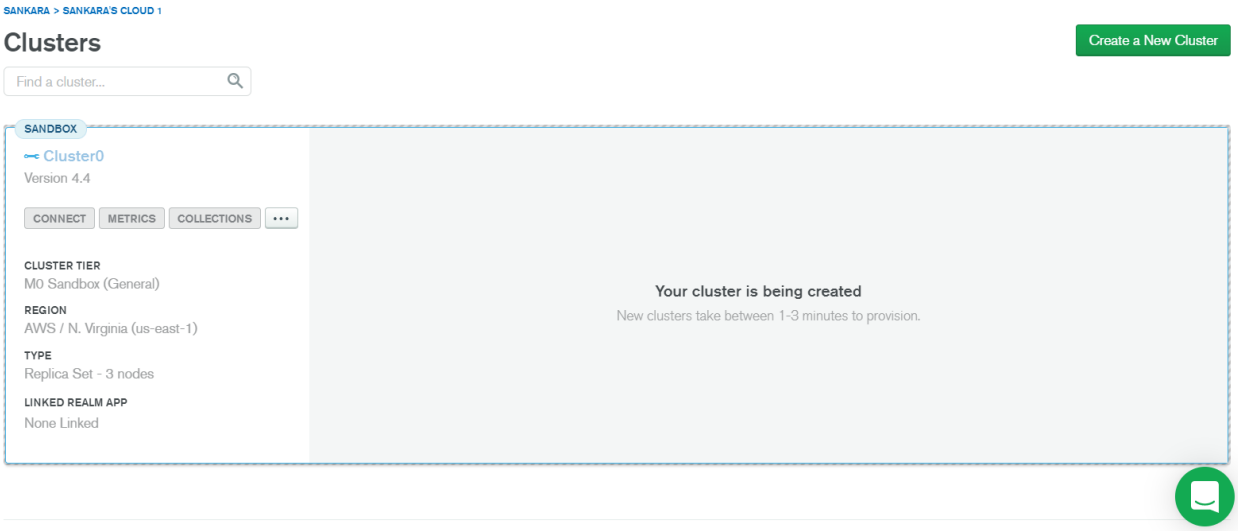


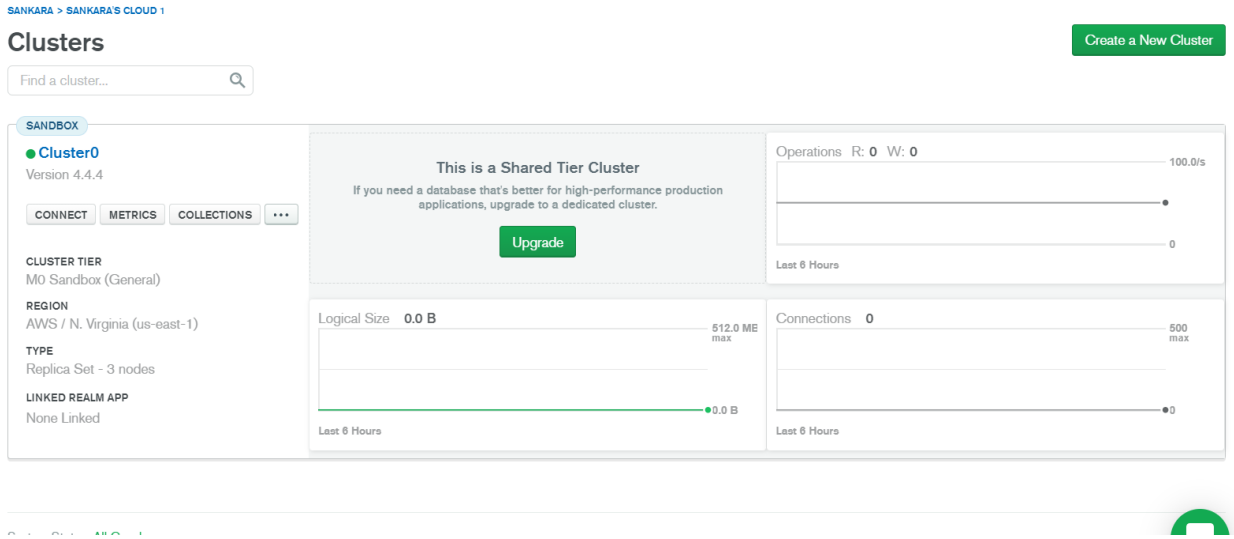
4. Create a new cluster(Shared M0) after the project is created.



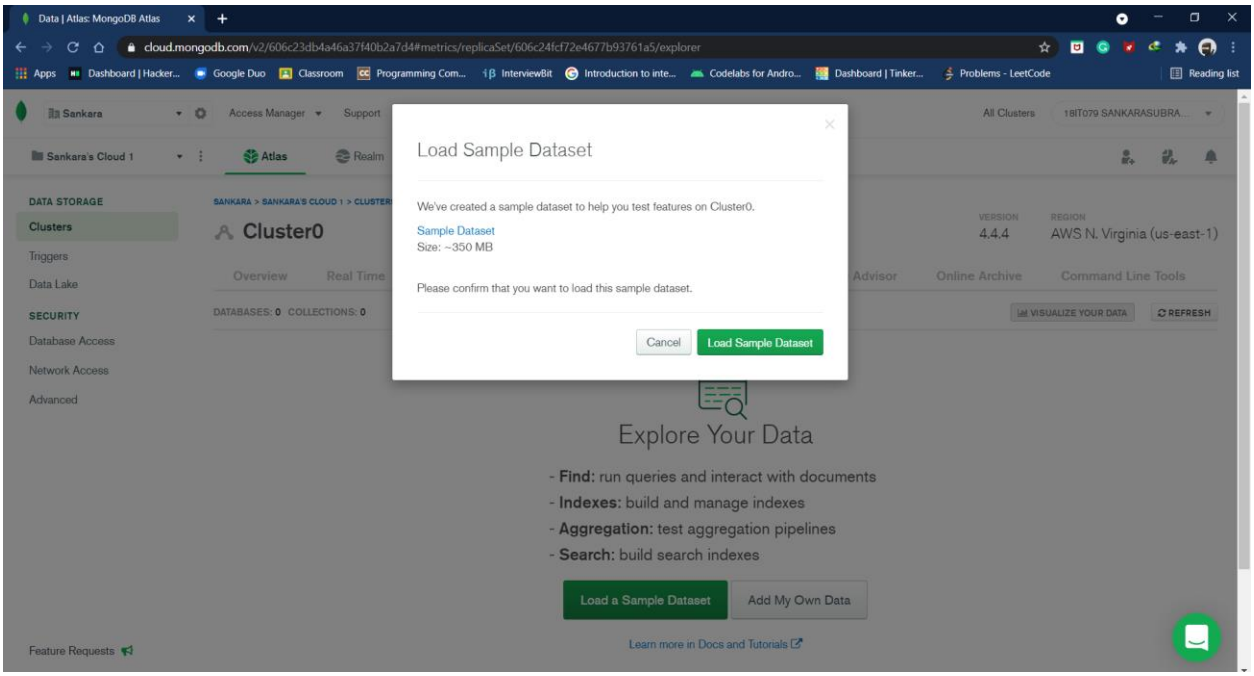
5. Select the required fields and create cluster.

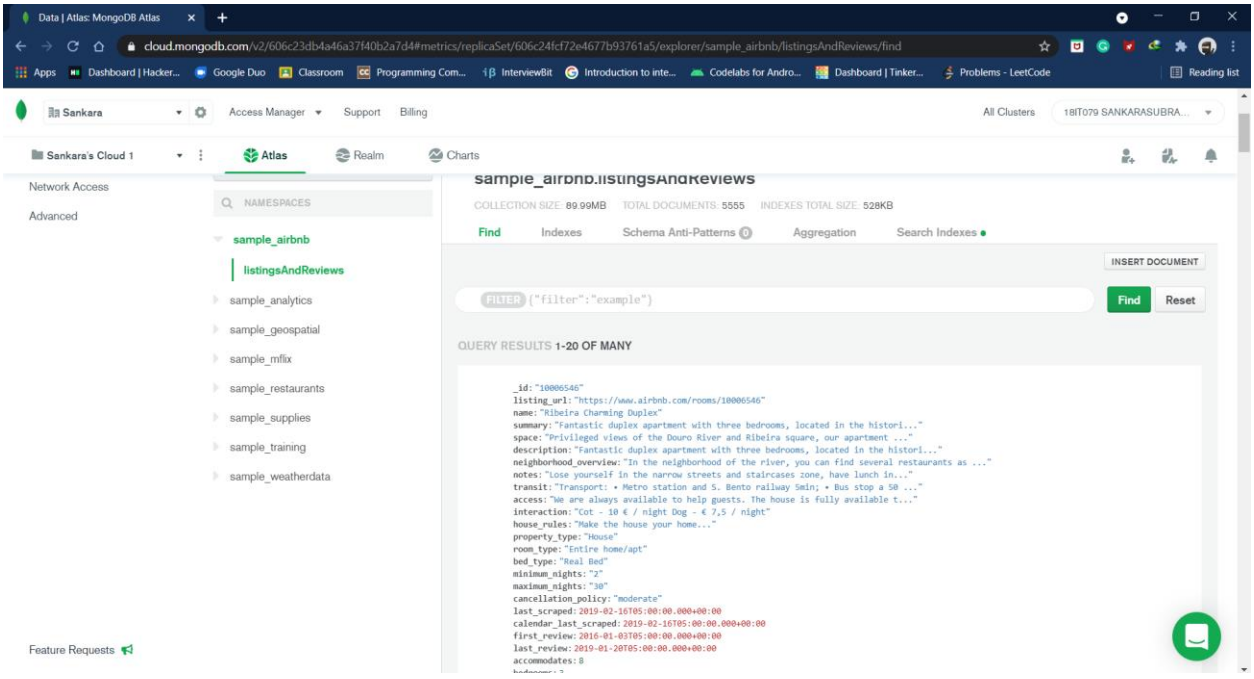



SANKARA > SANKARA'S CLOUD 1
Clusters
Create a New Cluster
Find a cluster...
SANDBOX
Cluster0
Version 4.4
CONNECT
METRICS
COLLECTIONS
CLUSTER TIER
M0 Sandbox (General)
REGION
AWS / N. Virginia (us-east-1)
TYPE
Replica Set - 3 nodes
LINKED REALM APP
None Linked
Your cluster is being created
New clusters take between 1-3 minutes to provision.

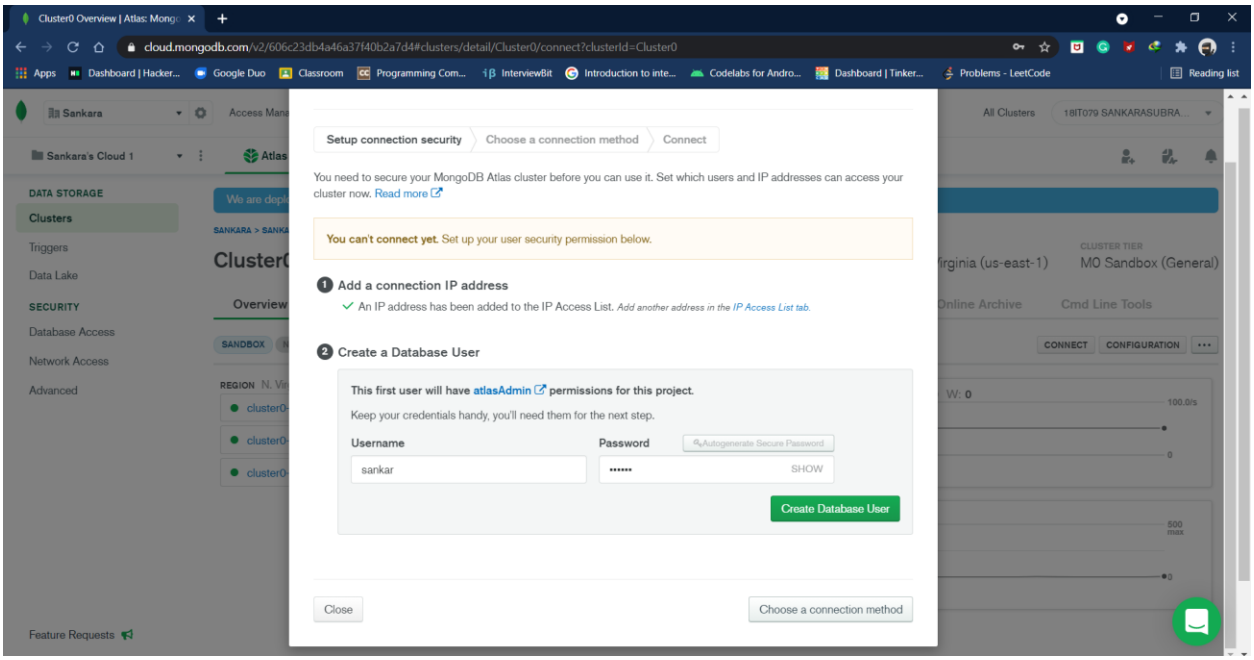

SANKARA > SANKARA'S CLOUD 1
Clusters
Create a New Cluster
Find a cluster...
SANDBOX
Cluster0
Version 4.4.4
CONNECT
METRICS
COLLECTIONS
CLUSTER TIER
M0 Sandbox (General)
REGION
AWS / N. Virginia (us-east-1)
TYPE
Replica Set - 3 nodes
LINKED REALM APP
None Linked
This is a Shared Tier Cluster
If you need a database that's better for high-performance production applications, upgrade to a dedicated cluster.
Upgrade
Operations R: 0 W: 0
100.0/s
0
Last 6 Hours
Logical Size 0.0 B
512.0 MB max
0.0 B
Last 6 Hours
Connections 0
500 max
0
Last 6 Hours

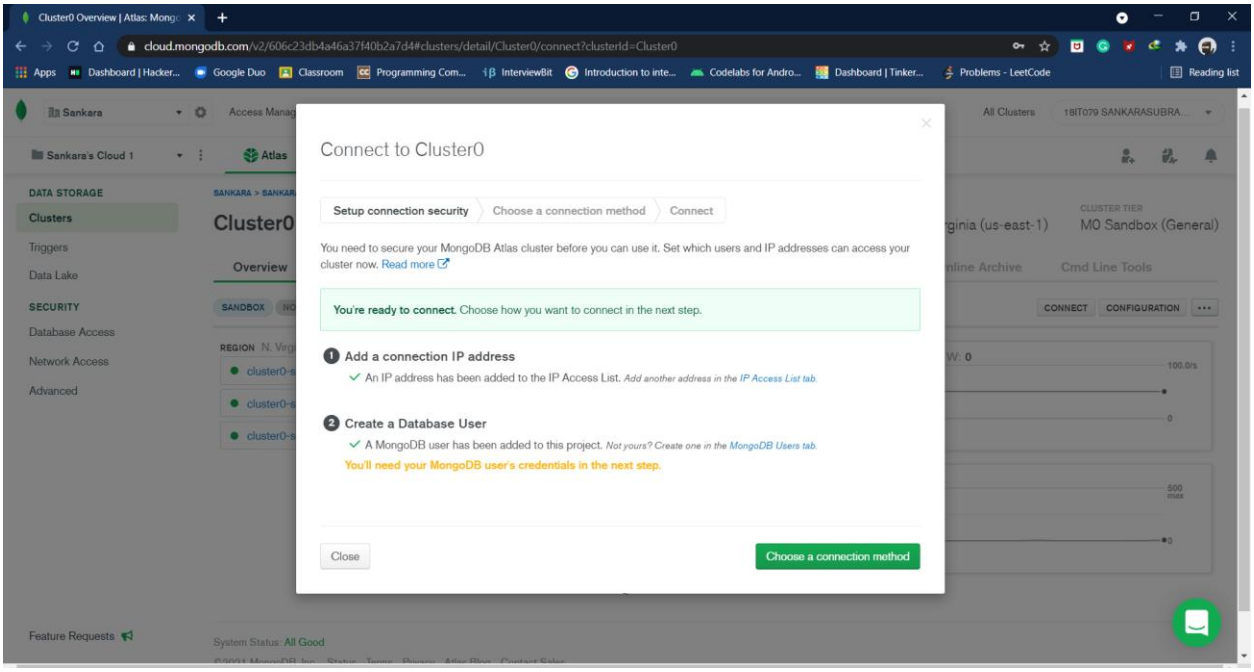
6. Load the sample dataset into the collection



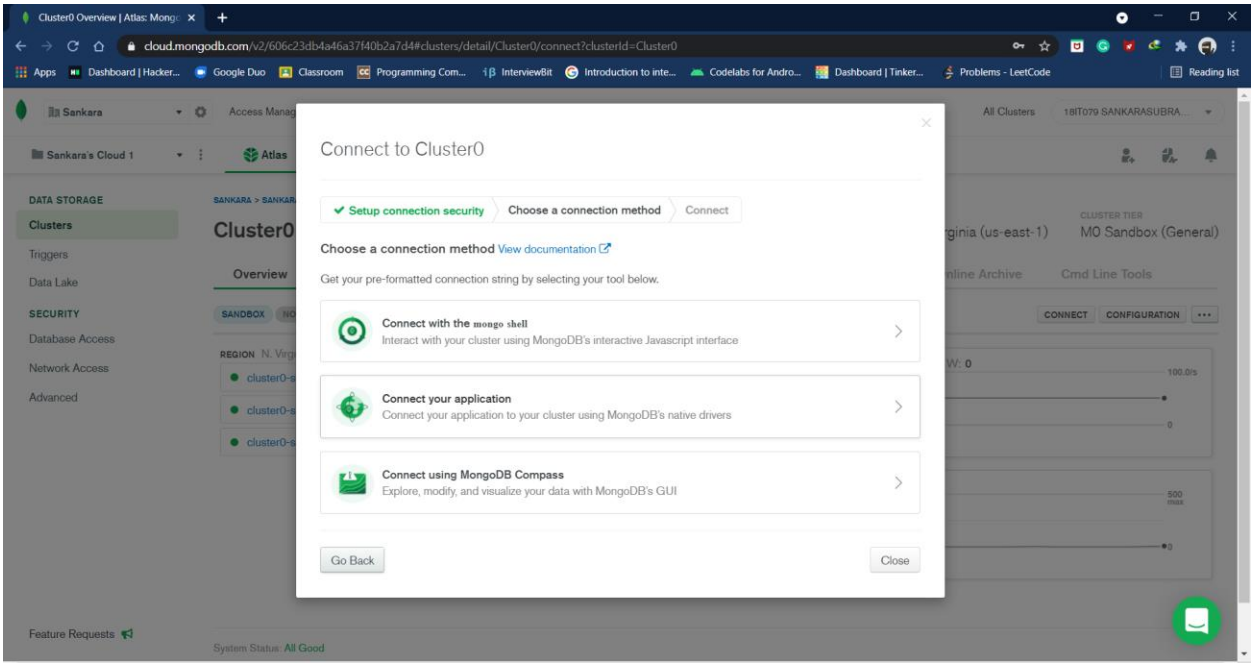


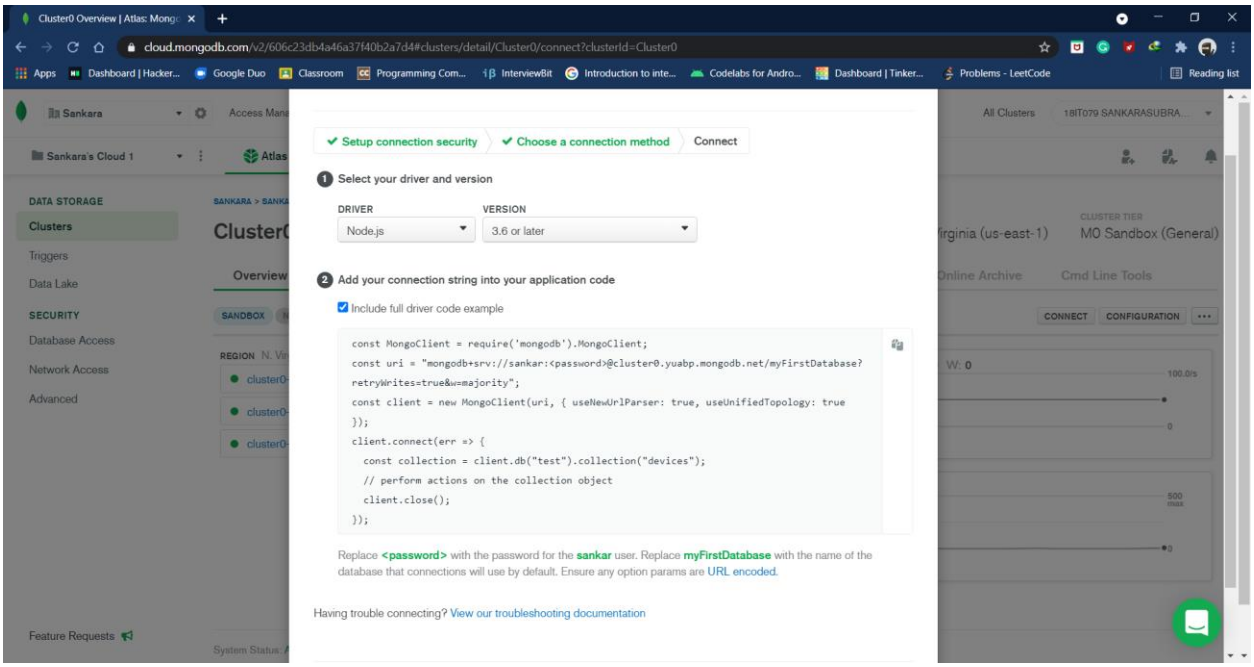
7. Add IP address and Create database user.



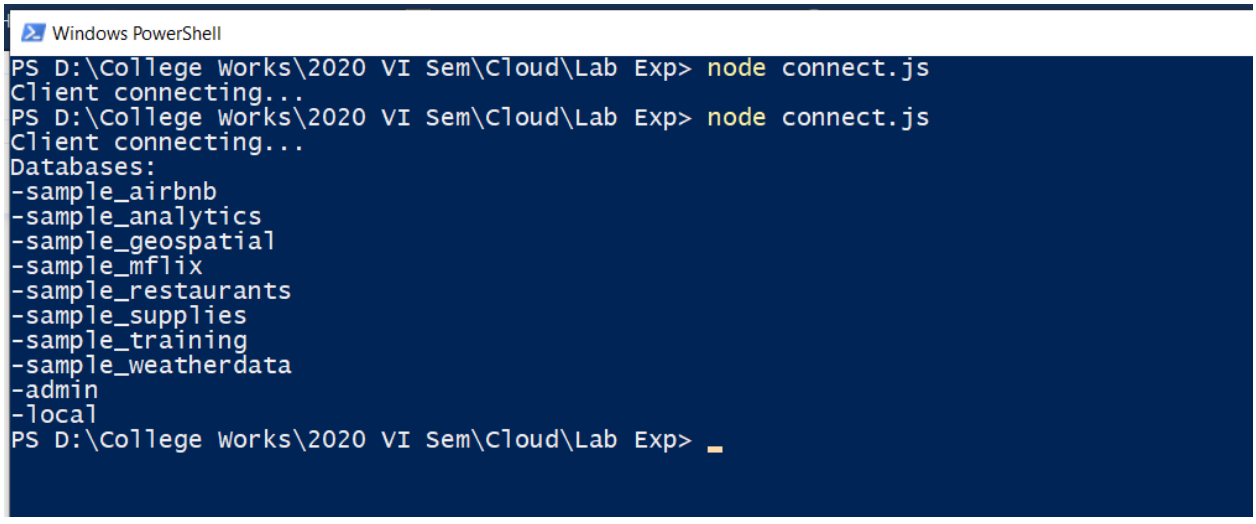


8. Connect your application to the cluster.

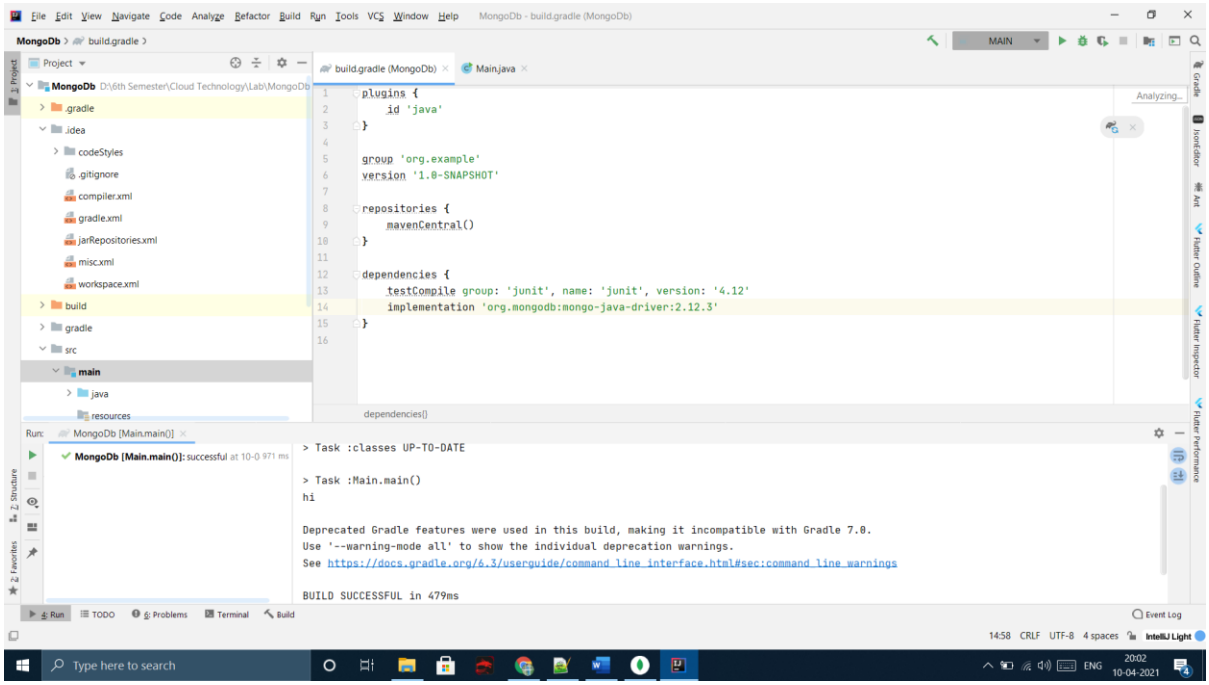




9. Perform some operation from node js and run to see the result.



10. Import the necessary jar files for mongodb java driver.



Code for CRUD Operations(Select All):

```
import com.mongodb.*;
import org.json.JSONObject;

import java.util.Iterator;

public class Main {
   public static void main(String[] arg)
   {
      try {
         MongoClientURI uri = new MongoClientURI(
```

```
"mongodb+srv://YYYY:XXXX@crop.ry3da.mongodb.net/myFirstDatabase?
retryWrites=true&w=majority");
        MongoClient mongoClient = new MongoClient(uri);
        DB db=mongoClient.getDB("Scriptons");
        DBCollection crop=db.getCollection("Water");
        DBCursor cursor=crop.find();
        JSONObject jsonObject=new JSONObject(cursor.next().toString());
        JSONObject
water=jsonObject.getJSONObject("yajith").getJSONObject("water_level");
        Iterator<String> iterator=water.keys();
        while (iterator.hasNext())
        {
           String keys=iterator.next();
           System.out.println(keys+" "+water.getString(keys));
        }
      } catch (Exception e) {
        e.printStackTrace();
      }
   }
}
```

Code for CRUD Operations(Insert):

```
import com.mongodb.*;
```

```
import com.mongodb.client.MongoCollection;

import org.bson.Document;

import org.bson.types.ObjectId;

import org.json.JSONObject;



import java.util.ArrayList;

import java.util.HashMap;

import java.util.Iterator;

import java.util.List;
public class Main {

   public static void main(String[] arg)

   {
```

```
        try {

            MongoClientURI uri = new MongoClientURI(

"mongodb+srv://yajith:vishwa@crop.ry3da.mongodb.net/myFirstDatabase?retryWrites=true&w=majority");

            MongoClient mongoClient = new MongoClient(uri);

            DB db=mongoClient.getDB("Scriptons");

            DBCollection crop=db.getCollection("Water");

            HashMap<String,String> map=new HashMap<>();

            map.put("field1","20");

            map.put("field2","100");

            map.put("field3","10");
```

```
            map.put("field4","50");

            DBObject dbObject=new BasicDBObject("sankar",map);

            crop.insert(dbObject);



        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}
```

Code for CRUD Operations(Delete):

```
import com.mongodb.*;
import com.mongodb.client.MongoCollection;
import org.bson.Document;
```

```
import org.bson.types.ObjectId;

import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
public class Main {


   public static void main(String[] arg)


   {


      try {


         MongoClientURI uri = new MongoClientURI(



"mongodb+srv://yajith:vishwa@crop.ry3da.mongodb.net/myFirstDatabas
e?retryWrites=true&w=majority");


         MongoClient mongoClient = new MongoClient(uri);
```

```
        DB db=mongoClient.getDB("Scriptons");

        DBCollection crop=db.getCollection("Water");

        HashMap<String,String> map=new HashMap<>();

        map.put("field1","20");

        map.put("field2","100");

        map.put("field3","10");

        map.put("field4","50");

        DBObject dbObject=new BasicDBObject("sankar",map);

        crop.remove(dbObject);



    } catch (Exception e) {
```

```
            e.printStackTrace();

        }

    }

}
```

Code for CRUD Operations(Create):

```
import com.mongodb.*;

import com.mongodb.client.MongoCollection;

import org.bson.Document;

import org.bson.types.ObjectId;

import org.json.JSONObject;
```

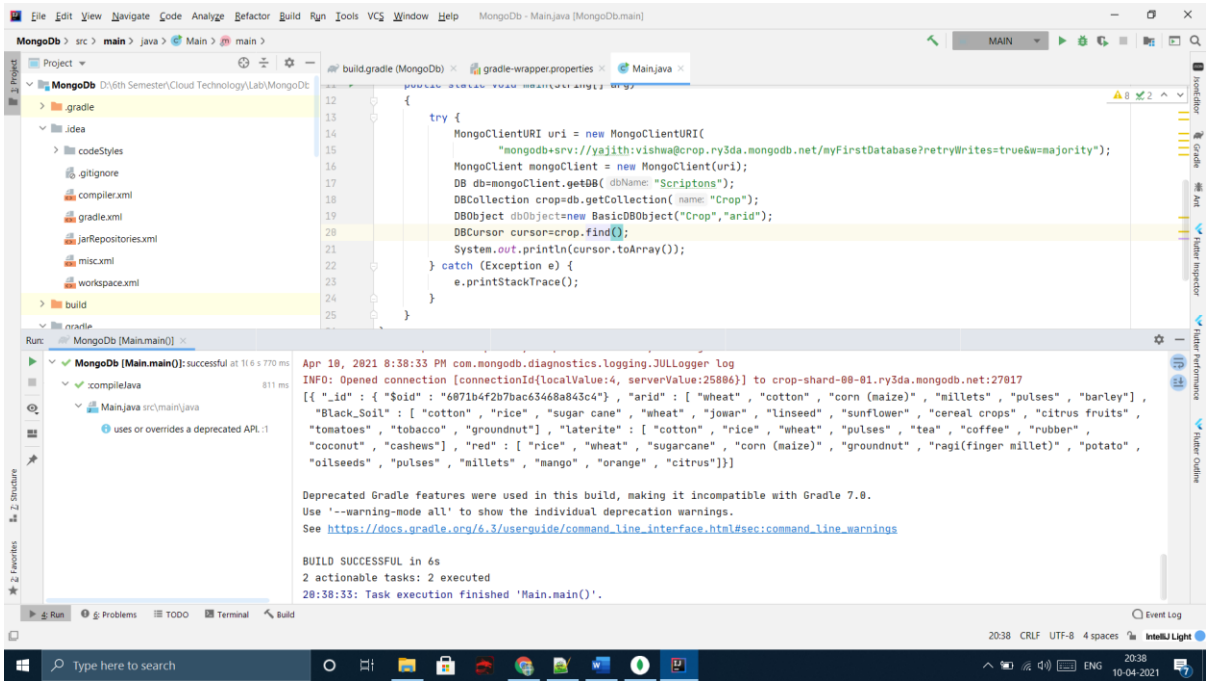
```
import java.util.ArrayList;

import java.util.HashMap;

import java.util.Iterator;

import java.util.List;



public class Main {

   public static void main(String[] arg)

   {

      try {

         MongoClientURI uri = new MongoClientURI(

"mongodb+srv://yajith:vishwa@crop.ry3da.mongodb.net/myFirstDatabas
```

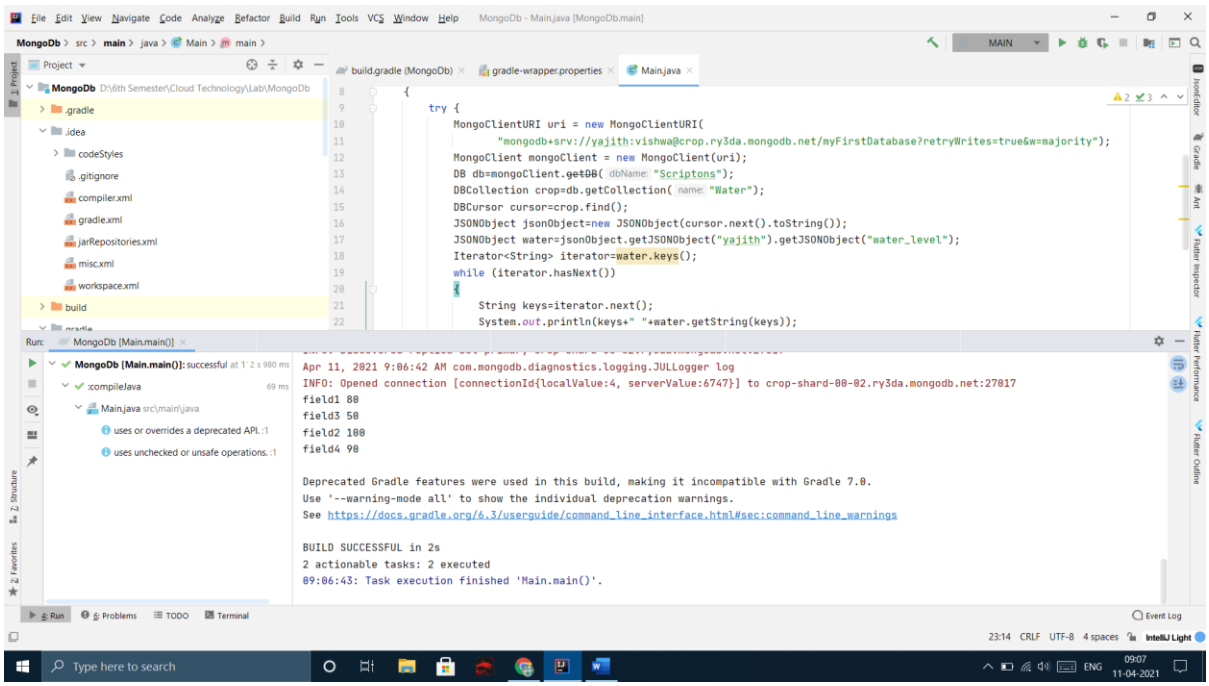
```
e?retryWrites=true&w=majority");

        MongoClient mongoClient = new MongoClient(uri);

        DB db=mongoClient.getDB("Scriptons");

        DBCollection crop=db.getCollection("Water");

        HashMap<String,String> map=new HashMap<>();

        map.put("field1","20");

        map.put("field2","100");

        map.put("field3","10");

        map.put("field4","50");

        DBObject dbObject=new BasicDBObject("sankar",map);

        db.createCollection("Dummy",dbObject);
```

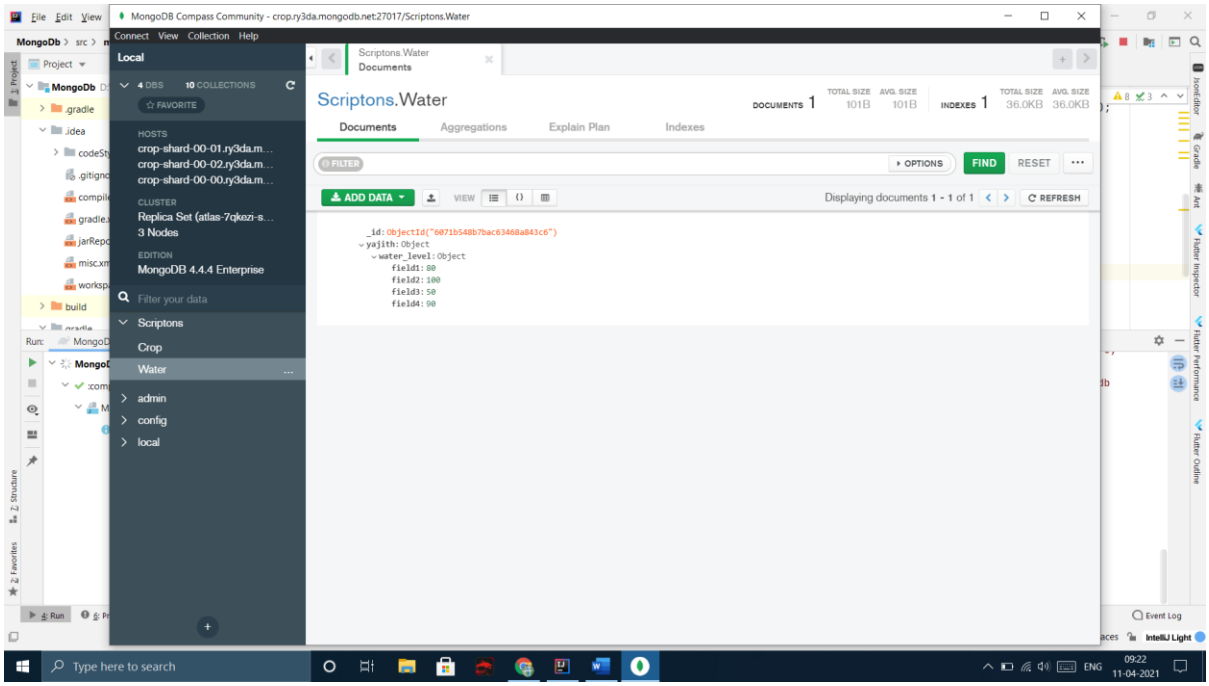
```
        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}
```

Screenshots:

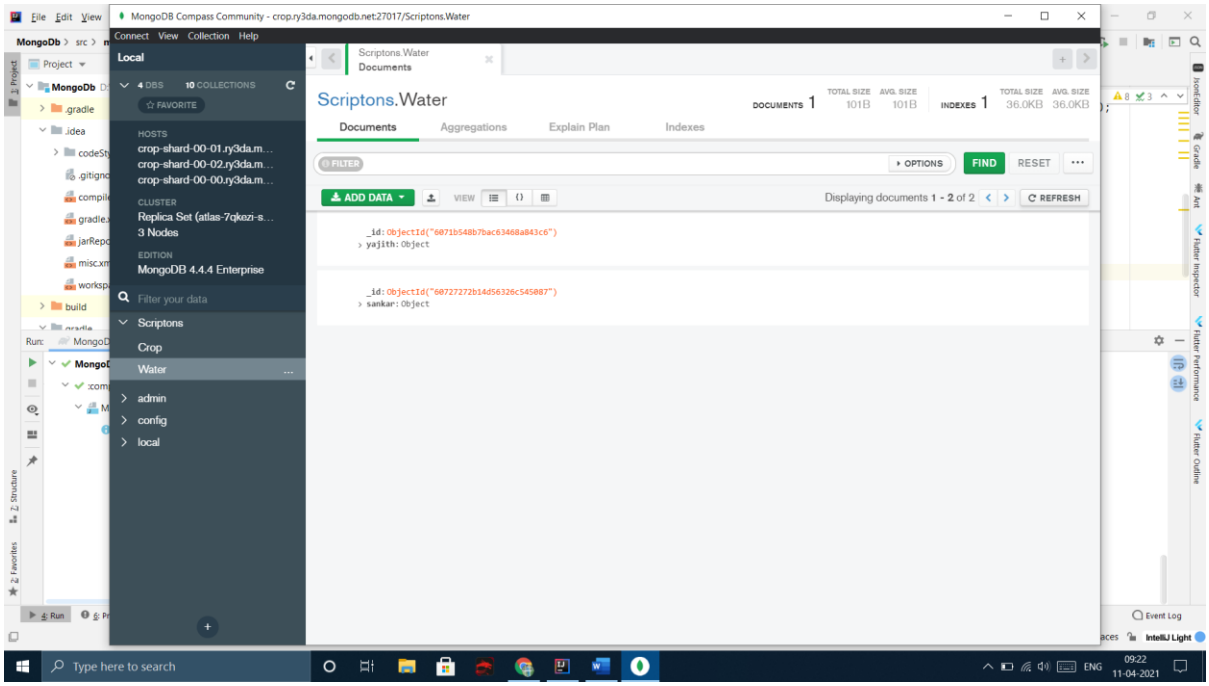
Select All

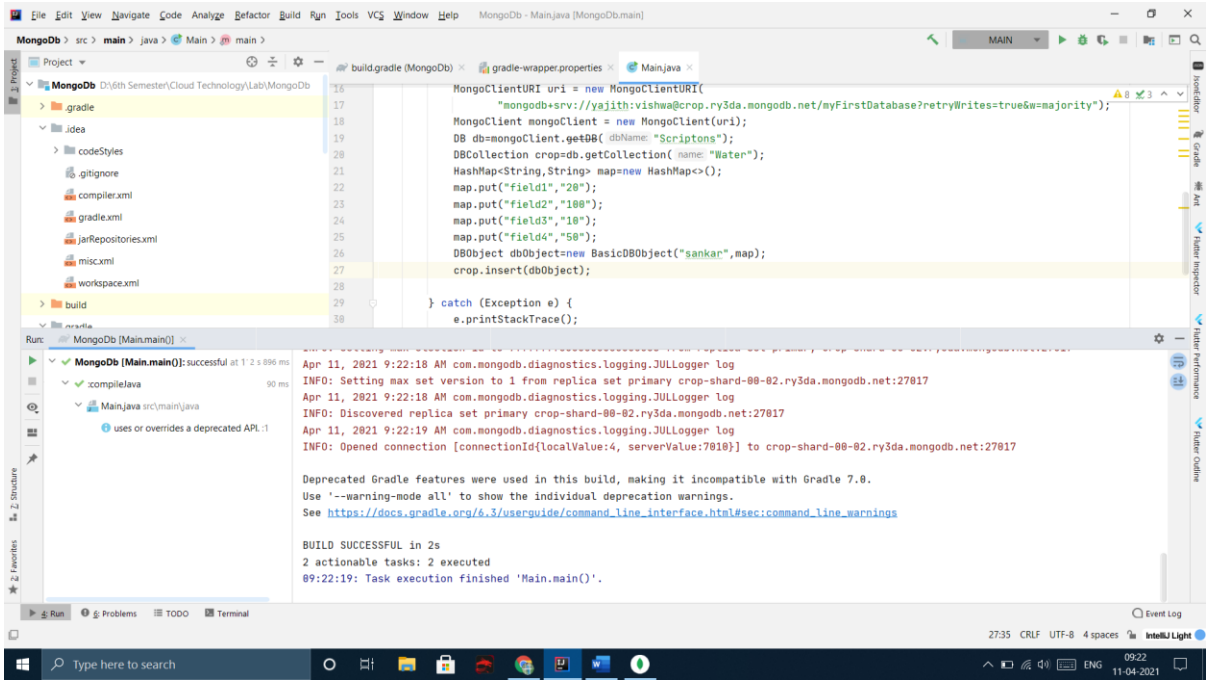




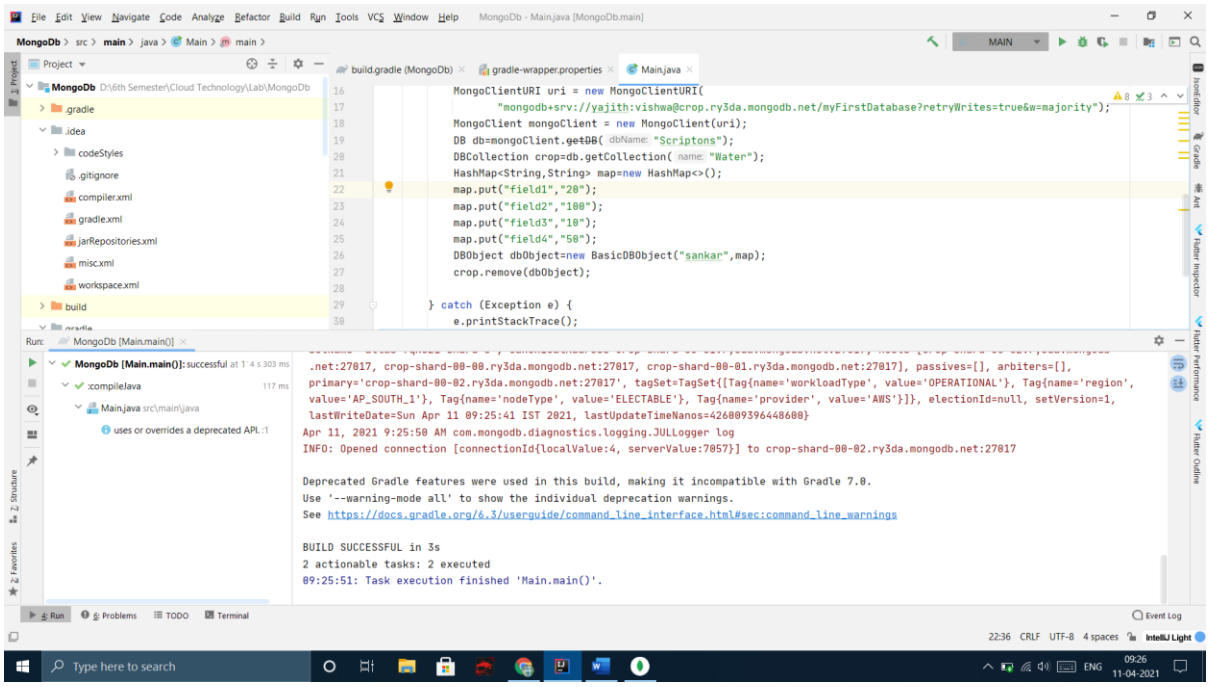


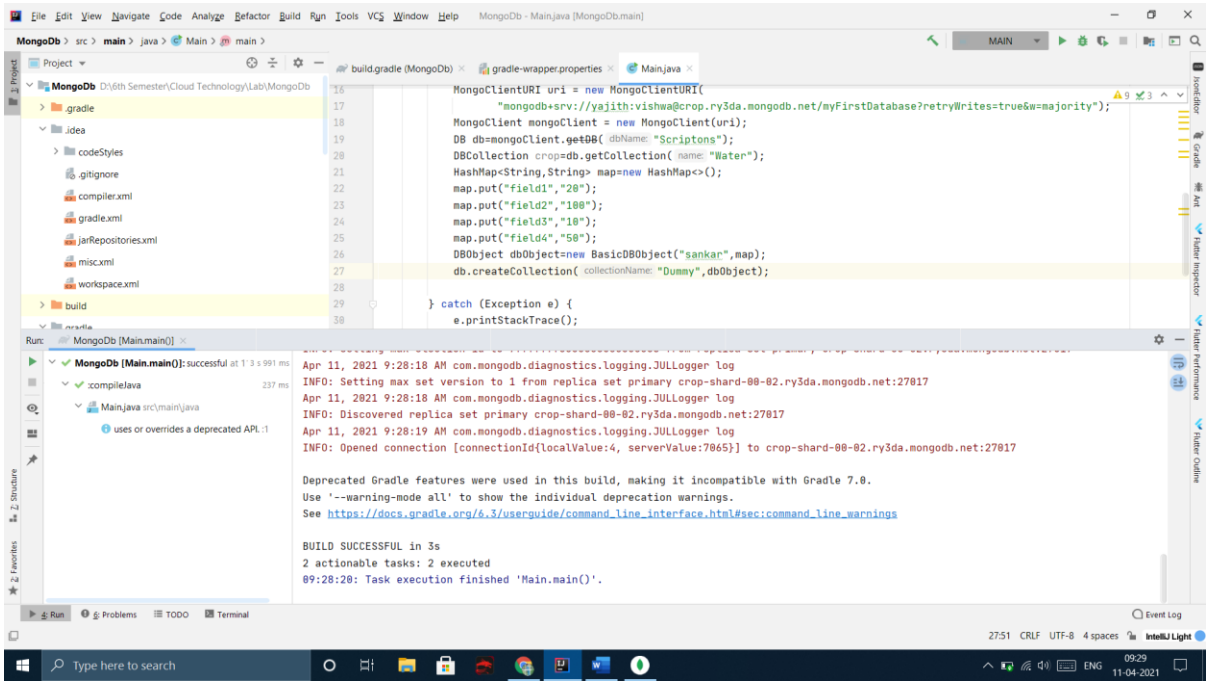
Insert Command


MongoDB Compass Community - crop.ry3da.mongodb.net:27017/Scriptons.Water
Connect View Collection Help
Local
4 DBS 10 COLLECTIONS
FAVORITE
HOSTS
crop-shard-00-01.ry3da.m...
crop-shard-00-02.ry3da.m...
crop-shard-00-00.ry3da.m...
CLUSTER
Replica Set (atlas-7qkezi-s...
3 Nodes
EDITION
MongoDB 4.4.4 Enterprise
Filter your data
Scriptons
Crop
Water
admin
config
local
Scriptons.Water
Documents
Aggregations
Explain Plan
Indexes
DOCUMENTS 1
INDEXES 1
FILTER
OPTIONS
FIND
RESET
ADD DATA
Displaying documents 1 - 2 of 2
REFRESH
_id: ObjectId("6071b548b7bac63468a843c6")
yajith: Object
_id: ObjectId("60727272b14d56326c545887")
sankar: Object
Type here to search
ENG
09:22
11-04-2021

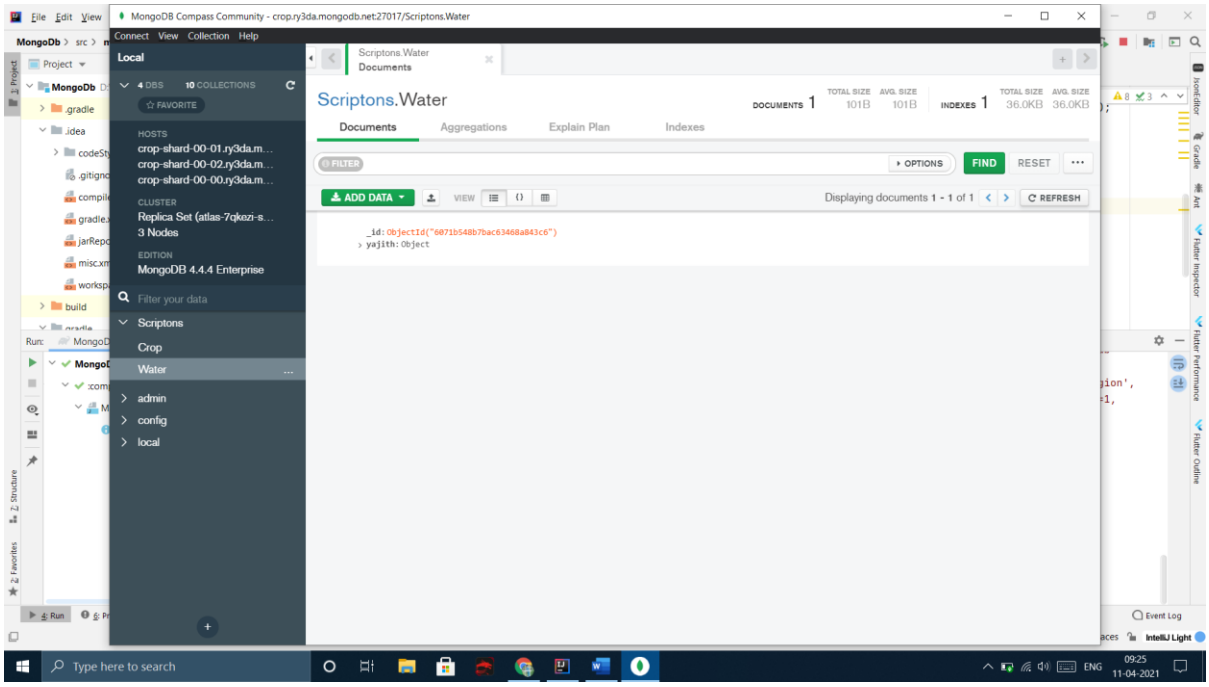

MongoDb - Main.java [MongoDb.main]
MongoDb > src > main > java > Main > main
MAIN
build.gradle (MongoDb)
gradle-wrapper.properties
Main.java
MongoClientURI uri = new MongoClientURI(
"mongodb+srv://yajith:vishwa@crop.ry3da.mongodb.net/myFirstDatabase?retryWrites=true&w=majority");
MongoClient mongoClient = new MongoClient(uri);
DB db=mongoClient.getDB(dbName: "Scriptons");
DBCollection crop=db.getCollection(name: "Water");
HashMap<String,String> map=new HashMap<>();
map.put("field1","20");
map.put("field2","100");
map.put("field3","10");
map.put("field4","50");
DBObject dbObject=new BasicDBObject("sankar",map);
crop.insert(dbObject);
} catch (Exception e) {
e.printStackTrace();
MongoDb [Main.main()]
MongoDb [Main.main()]: successful at 1' 2 s 896 ms
:compileJava 90 ms
Main.java src\main\java
uses or overrides a deprecated API. :1
Apr 11, 2021 9:22:18 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Setting max set version to 1 from replica set primary crop-shard-00-02.ry3da.mongodb.net:27017
Apr 11, 2021 9:22:18 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Discovered replica set primary crop-shard-00-02.ry3da.mongodb.net:27017
Apr 11, 2021 9:22:19 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:4, serverValue:7010}] to crop-shard-00-02.ry3da.mongodb.net:27017
Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.3/userguide/command_line_interface.html#sec:command_line_warnings
BUILD SUCCESSFUL in 2s
2 actionable tasks: 2 executed
09:22:19: Task execution finished 'Main.main()'.
Run
Problems
TODO
Terminal
Event Log
27:35 CRLF UTF-8 4 spaces
Type here to search
ENG
09:22
11-04-2021

Create


MongoClientURI uri = new MongoClientURI(
"mongodb+srv://yajith:vishwa@crop.ry3da.mongodb.net/myFirstDatabase?retryWrites=true&w=majority");
MongoClient mongoClient = new MongoClient(uri);
DB db=mongoClient.getDB(dbName: "Scriptons");
DBCollection crop=db.getCollection(name: "Water");
HashMap<String,String> map=new HashMap<>();
map.put("field1","20");
map.put("field2","100");
map.put("field3","10");
map.put("field4","50");
DBObject dbObject=new BasicDBObject("sankar",map);
crop.remove(dbObject);
} catch (Exception e) {
e.printStackTrace();
Apr 11, 2021 9:25:50 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:4, serverValue:7057}] to crop-shard-00-02.ry3da.mongodb.net:27017
Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.3/userguide/command_line_interface.html#sec:command_line_warnings
BUILD SUCCESSFUL in 3s
2 actionable tasks: 2 executed
09:25:51: Task execution finished 'Main.main()'.


MongoClientURI uri = new MongoClientURI(
"mongodb+srv://yajith:vishwa@crop.ry3da.mongodb.net/myFirstDatabase?retryWrites=true&w=majority");
MongoClient mongoClient = new MongoClient(uri);
DB db=mongoClient.getDB(dbName: "Scriptons");
DBCollection crop=db.getCollection(name: "Water");
HashMap<String,String> map=new HashMap<>();
map.put("field1","20");
map.put("field2","100");
map.put("field3","10");
map.put("field4","50");
DBObject dbObject=new BasicDBObject("sankar",map);
db.createCollection(collectionName: "Dummy",dbObject);
} catch (Exception e) {
e.printStackTrace();
Apr 11, 2021 9:28:18 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Setting max set version to 1 from replica set primary crop-shard-00-02.ry3da.mongodb.net:27017
Apr 11, 2021 9:28:18 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Discovered replica set primary crop-shard-00-02.ry3da.mongodb.net:27017
Apr 11, 2021 9:28:19 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:4, serverValue:7065}] to crop-shard-00-02.ry3da.mongodb.net:27017
Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.3/userguide/command_line_interface.html#sec:command_line_warnings
BUILD SUCCESSFUL in 3s
2 actionable tasks: 2 executed
09:28:20: Task execution finished 'Main.main()'.

Delete



Result

Thus, application is connected to mongoDb atlas and CRUD operations is performed and the result is obtained.