Lab: Designing a non-relational database application

Objectives:

- Design and develop an application using a non-relational database.

This is a 1-week lab (due in week 9). You will use MongoDB for the two week project (due end of week 10), unless you get another project pre-approved with your professor.

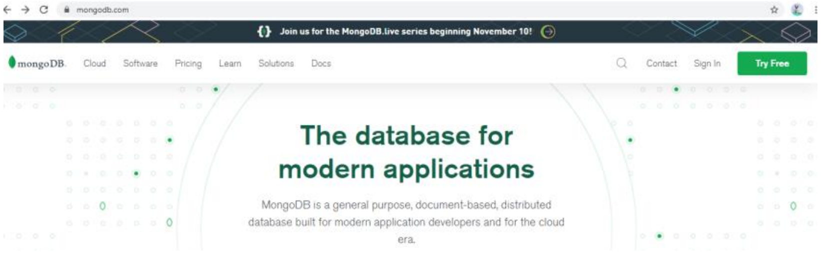
Summary:

1. Set up a free MongoDB account and load in sample data.
2. Complete ***Part 2,*** which has you working through many of the steps in a MongoDB/java tutorial. If you would rather use a different programming language, you can find the relevant tutorial and ask you professor to approve the switch. ***You'll be turning in a very short report that includes a few screenshots to show were able to work through the tutorial.***

Note: Consult Canvas for due dates.

Part 1: Set up a free MongoDB account and load in sample data (the only observation/screenshot needed for part 1 is in step 12).

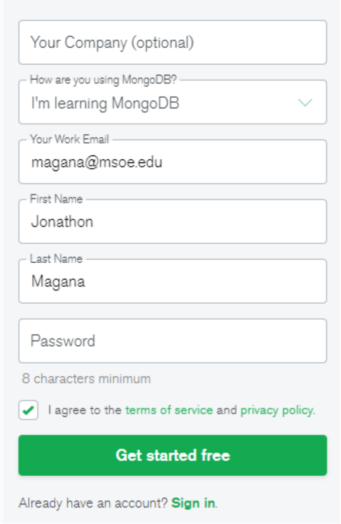
- **There is a video that walks through this process available, however it appears it uses a slightly older version of MongoDB, *and ends with installing the MongoDB Compass application, which we don't need*. You can view the view here (https://developer.mongodb.com/quickstart/free-atlas-cluster), but there are also step by step screenshots included below.**
  - ○ **Step 1: go to www.mongodb.com and click "Try Free" in the upper right corner. Note that if you have a MongoDB account already from a previous class, you can login and use that (skip to step 6). You can login with a google account by clicking "Sign In" if you want (skip to step 3).**



The database for modern applications

MongoDB is a general purpose, document-based, distributed database built for modern application developers and for the cloud era.

- **Step 2: Fill in your info ---------------------------→**



Your Company (optional)

How are you using MongoDB? I'm learning MongoDB

Your Work Email: magana@msoe.edu

First Name: Jonathon

Last Name: Magana

Password

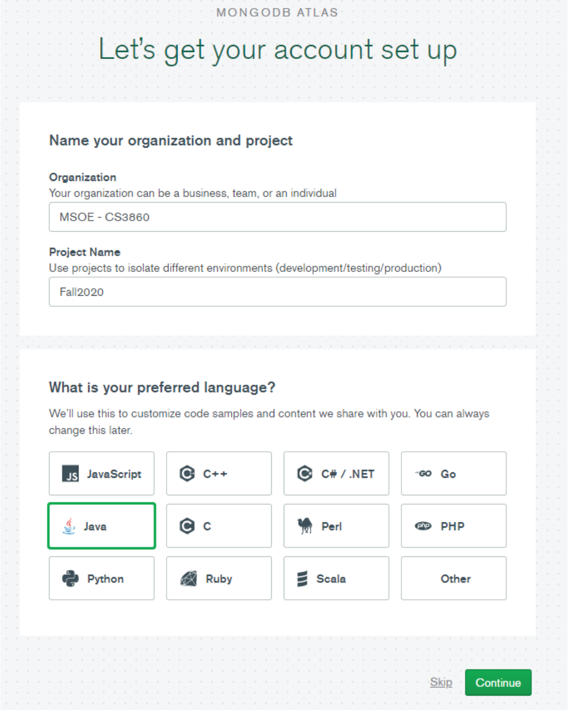
8 characters minimum

☑ I agree to the terms of service and privacy policy.

Get started free

Already have an account? Sign in.

- **Step 3: Fill in additional info (you can choose what to put here, it doesn't really matter for our class).**



MONGODB ATLAS

Let's get your account set up

Name your organization and project

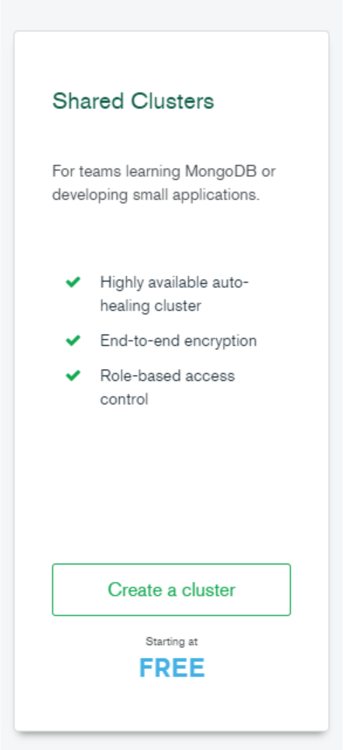
Organization
Your organization can be a business, team, or an individual
MSOE - CS3860

Project Name
Use projects to isolate different environments (development/testing/production)
Fall2020

What is your preferred language?

We'll use this to customize code samples and content we share with you. You can always change this later.

- **Step 4: Choose the "Free" Cluster option -------→**



Shared Clusters

For teams learning MongoDB or developing small applications.

✓ Highly available auto-healing cluster

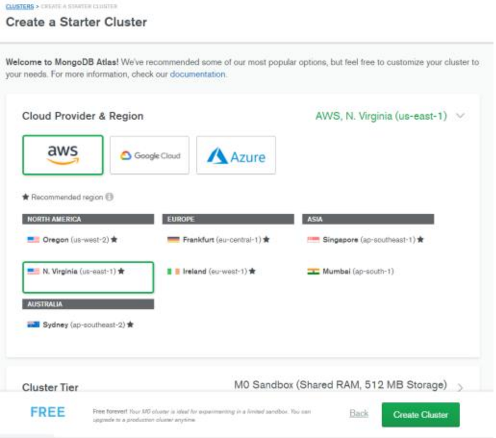
✓ End-to-end encryption

✓ Role-based access control

Create a cluster

Starting at FREE

- **Step 5: Leave defaults for Cluster:**



CLUSTERS > CREATE A STARTER CLUSTER

Create a Starter Cluster

Welcome to MongoDB Atlas! We've recommended some of our most popular options, but feel free to customize your cluster to your needs. For more information, check our documentation.

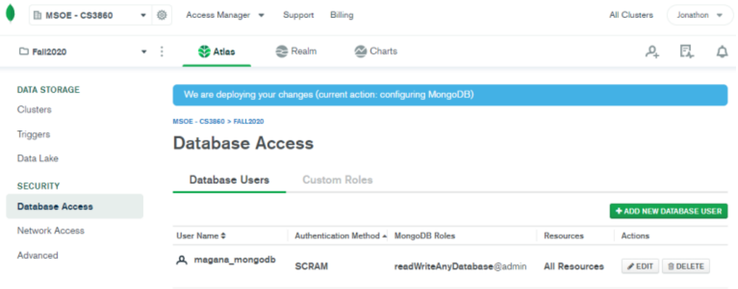
Cloud Provider & Region: AWS, N. Virginia (us-east-1)

Cluster Tier: M0 Sandbox (Shared RAM, 512 MB Storage)

FREE

Free forever! Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

- **Step 6: Add a user (magana_mongodb user is already created here, you click “add new database user”)**



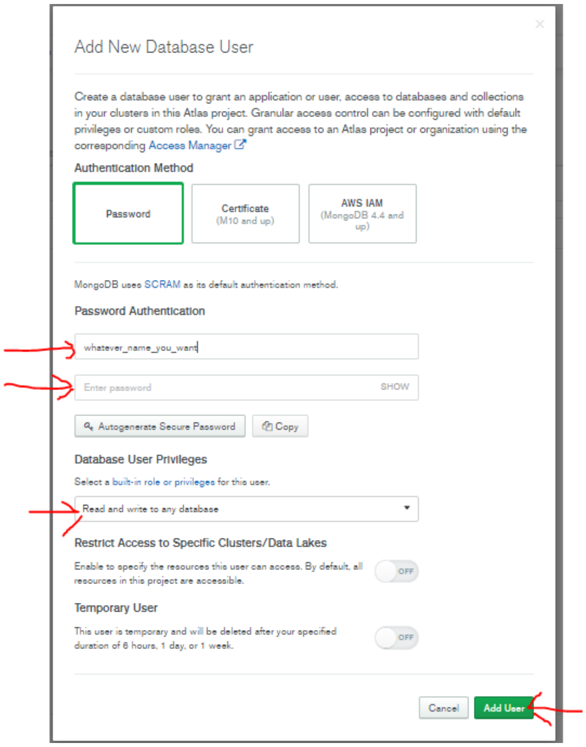
We are deploying your changes (current action: configuring MongoDB)

MSOE - CS3860 > FALL2020

Database Access

User Name	Authentication Method	MongoDB Roles	Resources	Actions
magana_mongodb	SCRAM	readWriteAnyDatabase@admin	All Resources	EDIT DELETE

- **Step 7: Fill in create user details**



Add New Database User

Create a database user to grant an application or user, access to databases and collections in your clusters in this Atlas project. Granular access control can be configured with default privileges or custom roles. You can grant access to an Atlas project or organization using the corresponding Access Manager

Authentication Method

MongoDB uses SCRAM as its default authentication method.

Password Authentication

whatever_name_you_want

Enter password SHOW

Database User Privileges

Select a built-in role or privileges for this user.

Read and write to any database

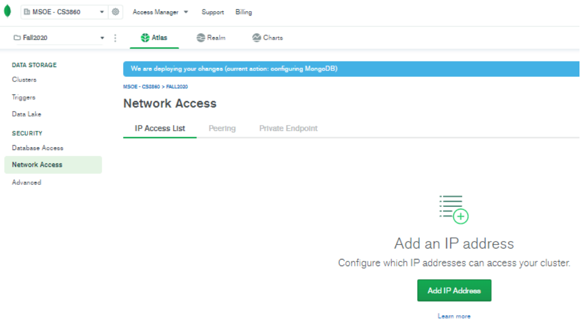
Restrict Access to Specific Clusters/Data Lakes

Enable to specify the resources this user can access. By default, all resources in this project are accessible. OFF

Temporary User

This user is temporary and will be deleted after your specified duration of 6 hours, 1 day, or 1 week. OFF

- **Step 8: Add network access**



We are deploying your changes (current action: configuring MongoDB)

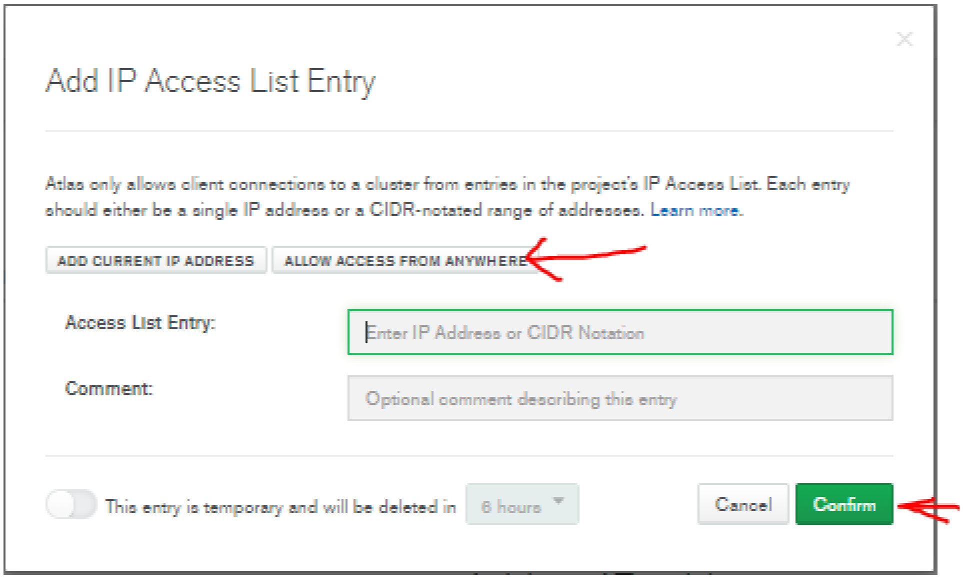
MSOE - CS3860 > FALL2020

Network Access

Add an IP address

Configure which IP addresses can access your cluster.

- **Step 9: Adding network access details (click “allow access from anywhere” and then “Confirm”)**



Add IP Access List Entry

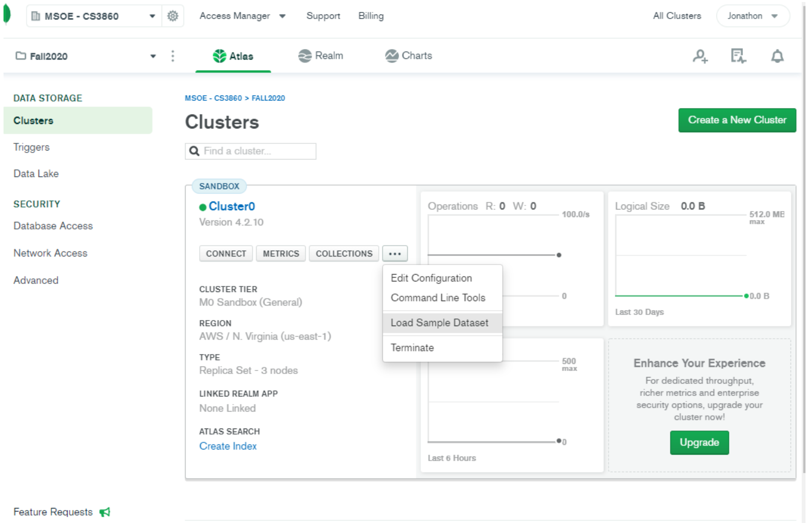
Atlas only allows client connections to a cluster from entries in the project's IP Access List. Each entry should either be a single IP address or a CIDR-notated range of addresses. Learn more.

Access List Entry: Enter IP Address or CIDR Notation

Comment: Optional comment describing this entry

This entry is temporary and will be deleted in 6 hours

- **Step 10: Load in the sample database (click Clusters to see your Cluster0), and then click the “…” next to connections button.**



MSOE - CS3860 > FALL2020

Clusters

SANDBOX

Cluster0
Version 4.2.10

CLUSTER TIER
M0 Sandbox (General)

REGION
AWS / N. Virginia (us-east-1)

TYPE
Replica Set - 3 nodes

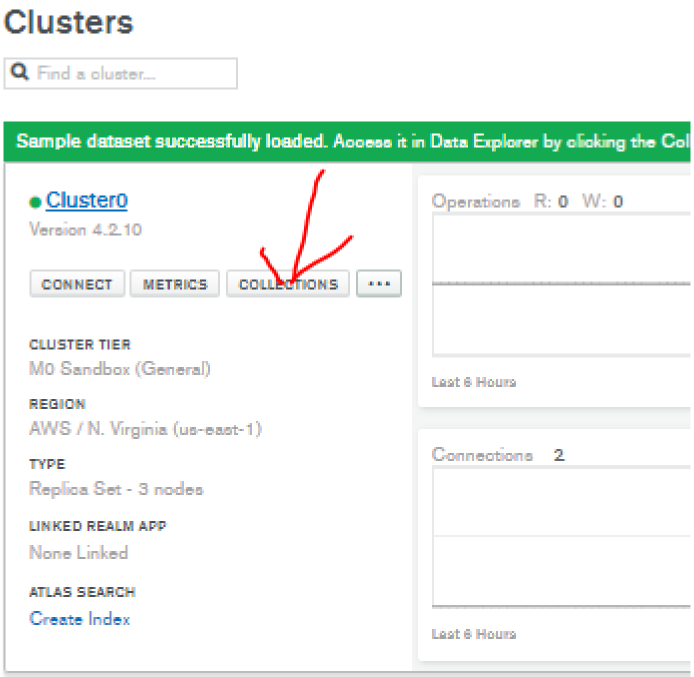
LINKED REALM APP
None Linked

ATLAS SEARCH
Create Index

Enhance Your Experience

For dedicated throughput, richer metrics and enterprise security options, upgrade your cluster now!

- **Step 11: Now click “Collections” and you should see the sample data**



Clusters

Sample dataset successfully loaded. Access it in Data Explorer by clicking the Col

Cluster0
Version 4.2.10

CLUSTER TIER
M0 Sandbox (General)

REGION
AWS / N. Virginia (us-east-1)

TYPE
Replica Set - 3 nodes

LINKED REALM APP
None Linked

ATLAS SEARCH
Create Index

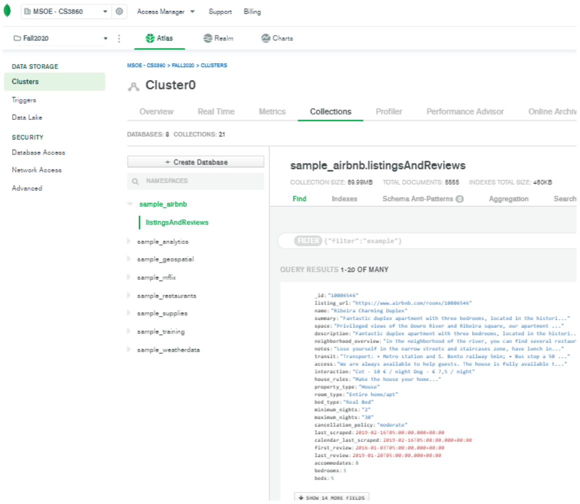
Operations R: 0 W: 0

Last 6 Hours

Connections 2

Last 6 Hours

- **Step 12: Here is a view of the sample data. As part of your lab report, you will need to include a screenshot like the one below to show you were able to load in the sample data. This is the “first” observation you need to capture for the lab report.**



MSOE - CS3860 > FALL2020 > CLUSTERS

Cluster0

DATABASES: 8 COLLECTIONS: 21

sample_airbnb.listingsAndReviews

COLLECTION SIZE: 89.99MB TOTAL DOCUMENTS: 5555 INDEXES TOTAL SIZE: 480KB

FILTER {"filter":"example"}

QUERY RESULTS 1-20 OF MANY

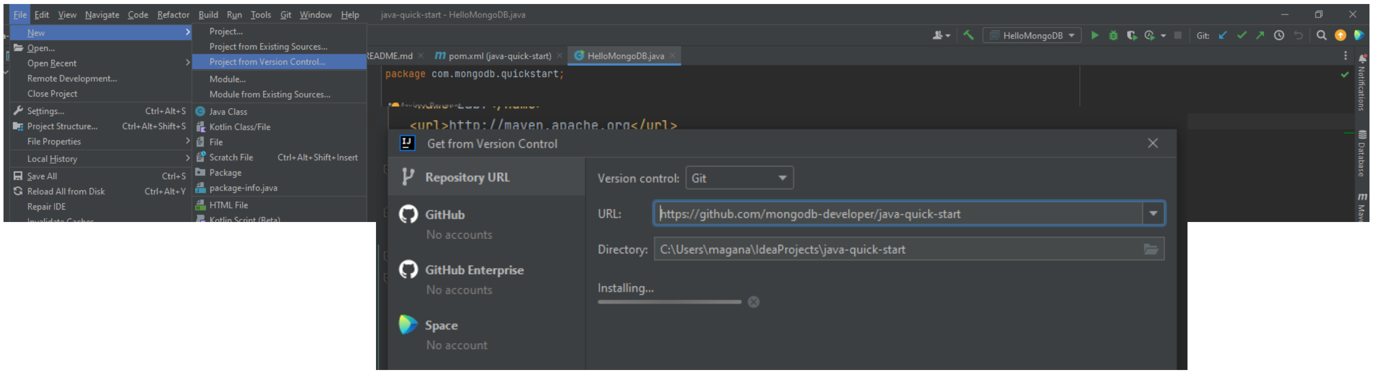
```
_id: "10006546"
listing_url: "https://www.airbnb.com/rooms/10006546"
name: "Ribeira Charming Duplex"
summary: "Fantastic duplex apartment with three bedrooms, located in the histori..."
space: "Privileged views of the Douro River and Ribeira square, our apartment ..."
description: "Fantastic duplex apartment with three bedrooms, located in the histori..
neighborhood_overview: "In the neighborhood of the river, you can find several restaur
notes: "Lose yourself in the narrow streets and staircases zone, have lunch in..."
transit: "Transport: • Metro station and S. Bento railway 5min; • Bus stop a 50 ..."
access: "We are always available to help guests. The house is fully available t..."
interaction: "Cot - 10 € / night Dog - € 7,5 / night"
house_rules: "Make the house your home..."
property_type: "House"
room_type: "Entire home/apt"
bed_type: "Real Bed"
minimum_nights: "2"
maximum_nights: "30"
cancellation_policy: "moderate"
last_scraped: 2019-02-16T05:00:00.000+00:00
calendar_last_scraped: 2019-02-16T05:00:00.000+00:00
first_review: 2016-01-03T05:00:00.000+00:00
last_review: 2019-01-20T05:00:00.000+00:00
accommodates: 8
bedrooms: 3
beds: 5
```

Part 2: Our goal with part 2 is to work through many of the steps listed in this tutorial: https://developer.mongodb.com/quickstart/java-setup-crud-operations You may also find this page helpful in understanding what is happening in the tutorial: https://docs.mongodb.com/manual/crud/

Note that you can choose to work through a tutorial in a different language, but make sure to get that pre-approved with your professor first. You can search through this site to find tutorials for other languages: https://developer.mongodb.com/learn

- **Step 1:** Open the tutorial (https://developer.mongodb.com/quickstart/java-setup-crud-operations).

- **Step 2:** Clone git repository to create a new project



```
package com.mongodb.quickstart;
```

```
<url>http://maven.apache.org</url>
```

Version control: Git

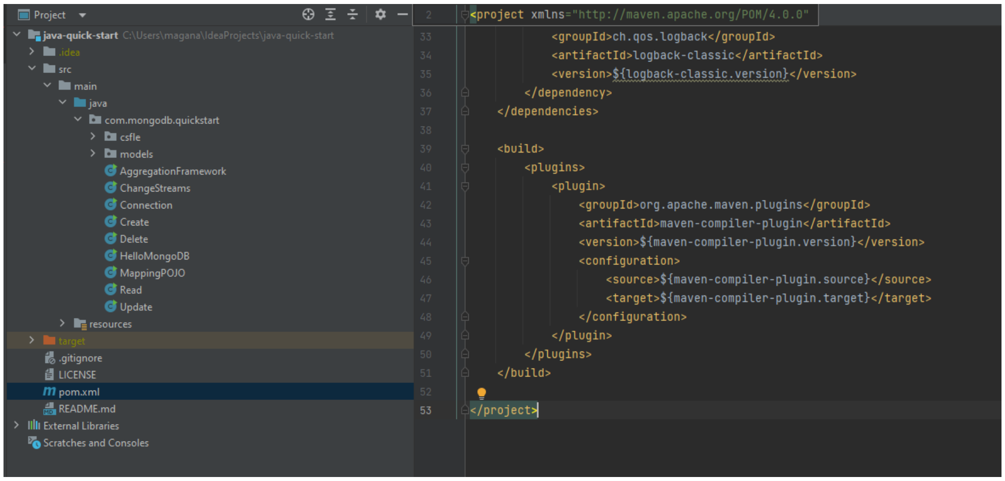
URL: https://github.com/mongodb-developer/java-quick-start

Directory: C:\Users\magana\IdeaProjects\java-quick-start

Installing...

You may need to click on this screen to install git, then use URL: https://github.com/mongodb-developer/java-quick-start

- **Step 3:** Verify that the files, including pom.xml, show up correctly

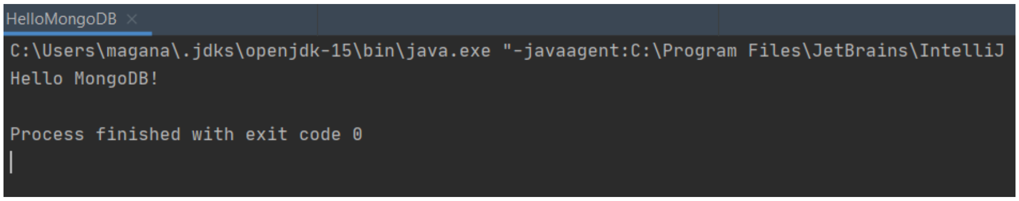


```
<project xmlns="http://maven.apache.org/POM/4.0.0"
            <groupId>ch.qos.logback</groupId>
            <artifactId>logback-classic</artifactId>
            <version>${logback-classic.version}</version>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>${maven-compiler-plugin.version}</version>
                <configuration>
                    <source>${maven-compiler-plugin.source}</source>
                    <target>${maven-compiler-plugin.target}</target>
                </configuration>
            </plugin>
        </plugins>
    </build>

</project>
```

- **Step 4:** Verify your pom.xml and directory structure match the code specified in the tutorial. You should not need to make any changes.

- **Step 5:** Run the HelloMongoDB java class

  I didn't get the fancier output mentioned in the tutorial, just the standard:

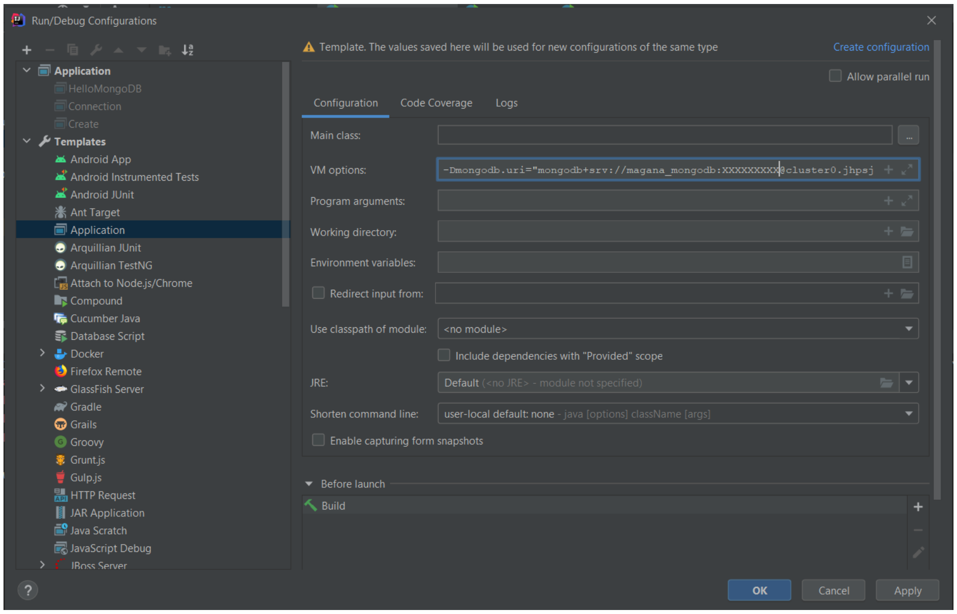
  ```
  C:\Users\magana\.jdks\openjdk-15\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ
  Hello MongoDB!

  Process finished with exit code 0
  ```

- **Step 6: Connect to MongoDB – we will be using the Connection class mentioned in the tutorial (**https://developer.mongodb.com/quickstart/java-setup-crud-operations#putting-the-driver-to-work) **BUT FIRST:** edit the connection information (in IntelliJ Run Menu -> Edit Configurations -> Templates -> Application and then put the modified connection string into the VM options field (so it will be used for all classes created from now on in this project). You will need to copy your connection String from your MongoDB cluster page. Your connection string will be something like this, and you will need to enter your password where it says <password>. This is the password created in Part1, step7 – not the password from Part1, step2.

  mongodb+srv://magana_mongodb:<password>@cluster0.jhpsj.mongodb.net/<dbname>?retryWrites=true&w=majority

  And you paste it into VM Options, adding the -Dmongodb.uri="…"*:*

  -Dmongodb.uri="mongodb+srv://
  magana_mongodb:<password>@cluster0.jhps j.mongodb.net/<dbname>?
  retryWrites=true&w=majority"



Template. The values saved here will be used for new configurations of the same type

□ Allow parallel run

Main class:

VM options: -Dmongodb.uri="mongodb+srv://magana_mongodb:XXXXXXXXX@cluster0.jhpsj

Program arguments:

Working directory:

Environment variables:

□ Redirect input from:

Use classpath of module: <no module>

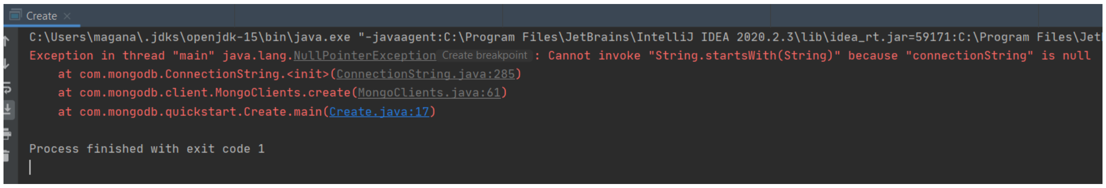
□ Include dependencies with "Provided" scope

JRE: Default (<no JRE> - module not specified)

Shorten command line: user-local default: none - java [options] className [args]

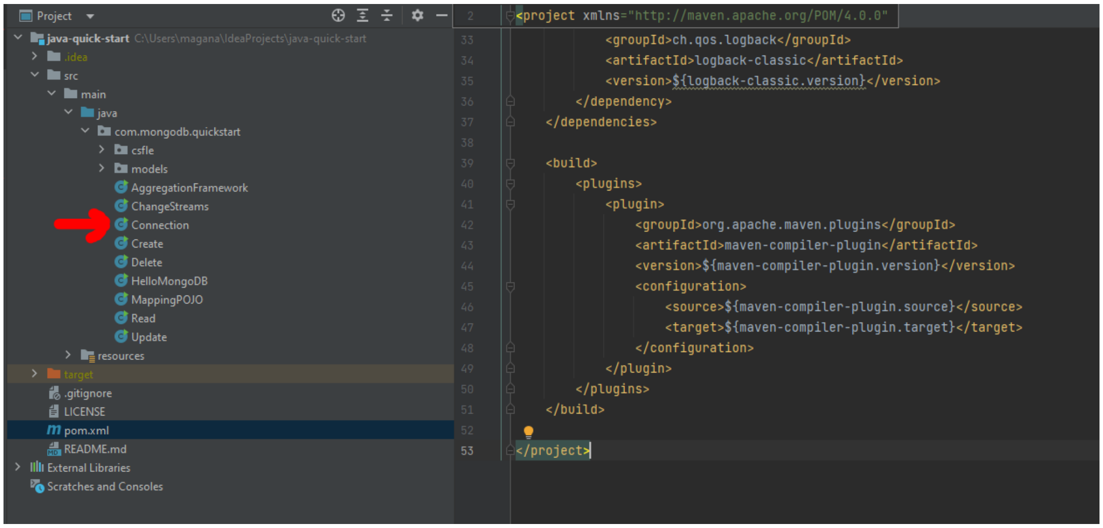
□ Enable capturing form snapshots

If you get this error in the future, double check that the VM Options under build configuration for the specific class you are running has the connection string:



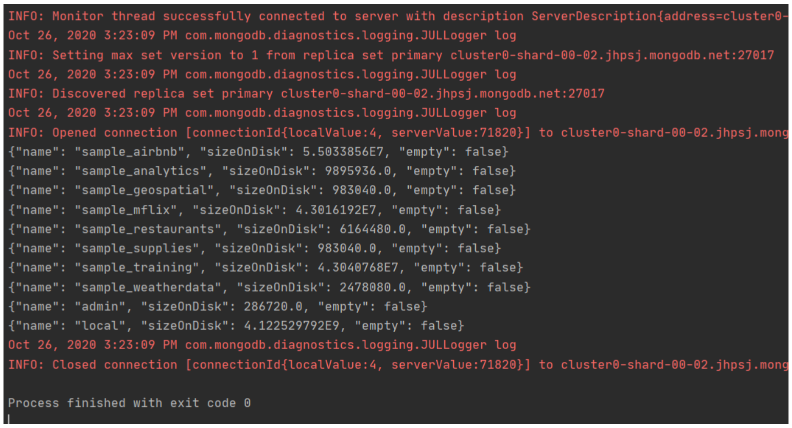
```
C:\Users\magana\.jdks\openjdk-15\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2020.2.3\lib\idea_rt.jar=59171:C:\Program Files\Jet
Exception in thread "main" java.lang.NullPointerException Create breakpoint : Cannot invoke "String.startsWith(String)" because "connectionString" is null
	at com.mongodb.ConnectionString.<init>(ConnectionString.java:285)
	at com.mongodb.client.MongoClients.create(MongoClients.java:61)
	at com.mongodb.quickstart.Create.main(Create.java:17)

Process finished with exit code 1
```

Now create verify the Connection class is available in the com.mongodb.quickstart package.



```
<project xmlns="http://maven.apache.org/POM/4.0.0"
                <groupId>ch.qos.logback</groupId>
                <artifactId>logback-classic</artifactId>
                <version>${logback-classic.version}</version>
            </dependency>
        </dependencies>

        <build>
            <plugins>
                <plugin>
                    <groupId>org.apache.maven.plugins</groupId>
                    <artifactId>maven-compiler-plugin</artifactId>
                    <version>${maven-compiler-plugin.version}</version>
                    <configuration>
                        <source>${maven-compiler-plugin.source}</source>
                        <target>${maven-compiler-plugin.target}</target>
                    </configuration>
                </plugin>
            </plugins>
        </build>

</project>
```

- **Step 7:** Run Connection class. You will see some red text and possibly errors (the tutorial mentions this), but you should also see some output. **Take a screenshot of this output and include it in the report!**



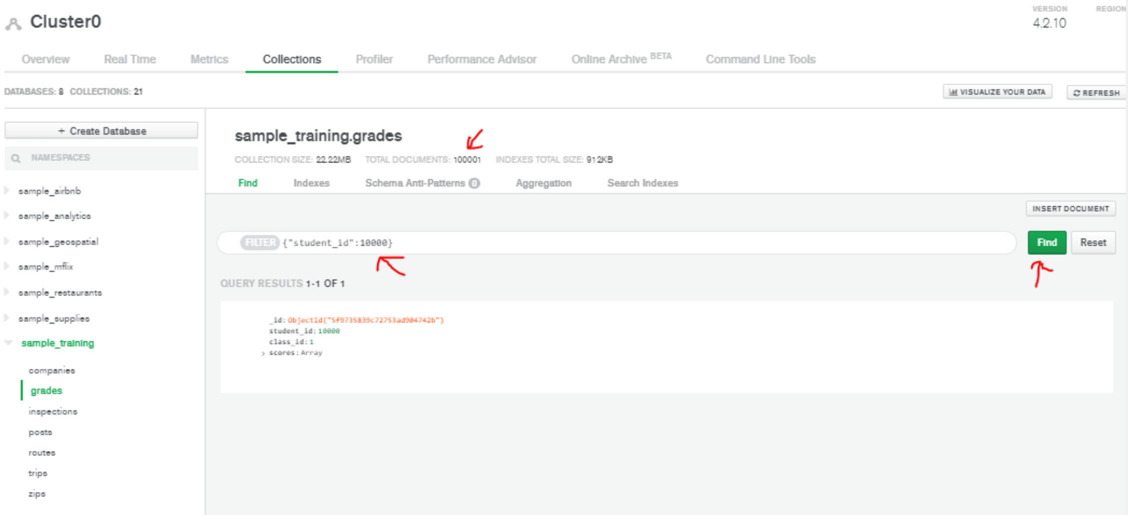
```
INFO: Monitor thread successfully connected to server with description ServerDescription{address=cluster0-
Oct 26, 2020 3:23:09 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Setting max set version to 1 from replica set primary cluster0-shard-00-02.jhpsj.mongodb.net:27017
Oct 26, 2020 3:23:09 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Discovered replica set primary cluster0-shard-00-02.jhpsj.mongodb.net:27017
Oct 26, 2020 3:23:09 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:4, serverValue:71820}] to cluster0-shard-00-02.jhpsj.mong
{"name": "sample_airbnb", "sizeOnDisk": 5.5033856E7, "empty": false}
{"name": "sample_analytics", "sizeOnDisk": 9895936.0, "empty": false}
{"name": "sample_geospatial", "sizeOnDisk": 983040.0, "empty": false}
{"name": "sample_mflix", "sizeOnDisk": 4.3016192E7, "empty": false}
{"name": "sample_restaurants", "sizeOnDisk": 6164480.0, "empty": false}
{"name": "sample_supplies", "sizeOnDisk": 983040.0, "empty": false}
{"name": "sample_training", "sizeOnDisk": 4.3040768E7, "empty": false}
{"name": "sample_weatherdata", "sizeOnDisk": 2478080.0, "empty": false}
{"name": "admin", "sizeOnDisk": 286720.0, "empty": false}
{"name": "local", "sizeOnDisk": 4.122529792E9, "empty": false}
Oct 26, 2020 3:23:09 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Closed connection [connectionId{localValue:4, serverValue:71820}] to cluster0-shard-00-02.jhpsj.mong

Process finished with exit code 0
```

- **Step 8:** Read about the data already loaded in the sample data set in the Insert Operations section (https://developer.mongodb.com/quickstart/java-setup-crud-operations#insert-operations)

- **Step 9:** Find the "Create" class based on the (Final Code to Insert One Document) and run it (https://developer.mongodb.com/quickstart/java-setup-crud-operations#final-code-to-insert-one-document)

  You may get several lines of red output messages, but should also get messages about grades being inserted.

- **Step 10:** Verify that your code create an addition student using the mongodb website, filter for {"student_id":10000}



sample_training.grades

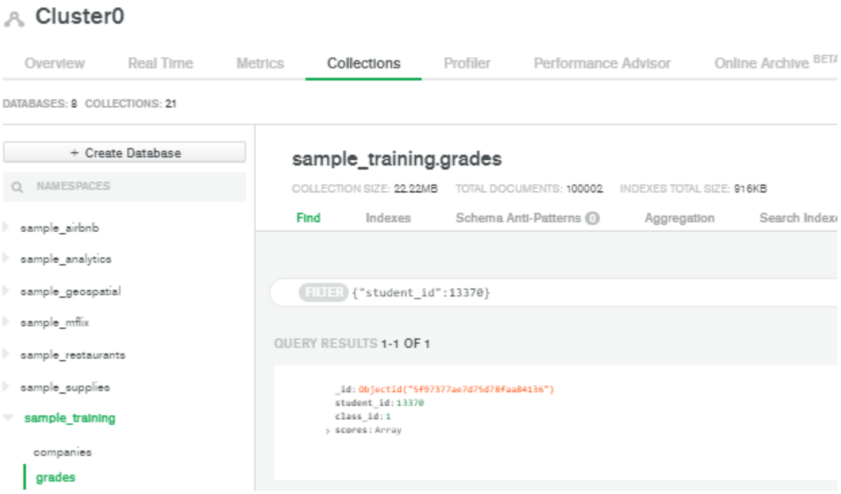
COLLECTION SIZE: 22.22MB TOTAL DOCUMENTS: 100001 INDEXES TOTAL SIZE: 912KB

FILTER {"student_id":10000}

QUERY RESULTS 1-1 OF 1

```
_id: ObjectId("5f9735839c72753ad904742b")
student_id: 10000
class_id: 1
> scores: Array
```

- **Step 11:** **THIS IS NOT PART OF THE TUTORIAL!** Modify the Create class to add an additional student with an id > 10000, run it, and verify it was added (you pick the student_id)



DATABASES: 8 COLLECTIONS: 21

sample_training.grades

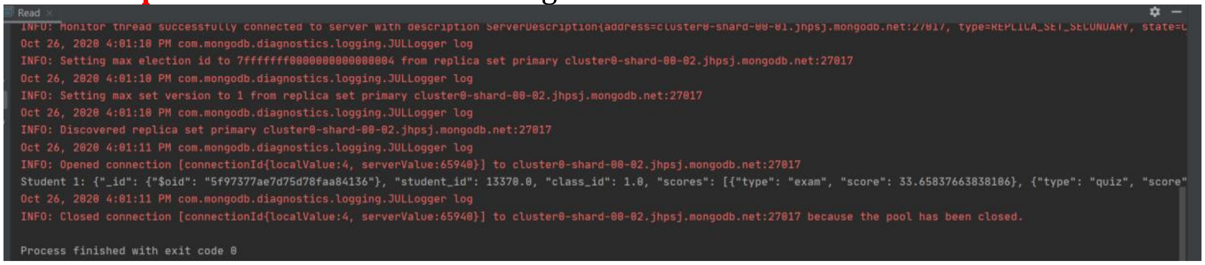
COLLECTION SIZE: 22.22MB TOTAL DOCUMENTS: 100002 INDEXES TOTAL SIZE: 916KB

```
{"student_id":13370}
```

QUERY RESULTS 1-1 OF 1

```
_id: ObjectId("5f97377ae7d75d78faa84136")
student_id: 13370
class_id: 1
> scores: Array
```

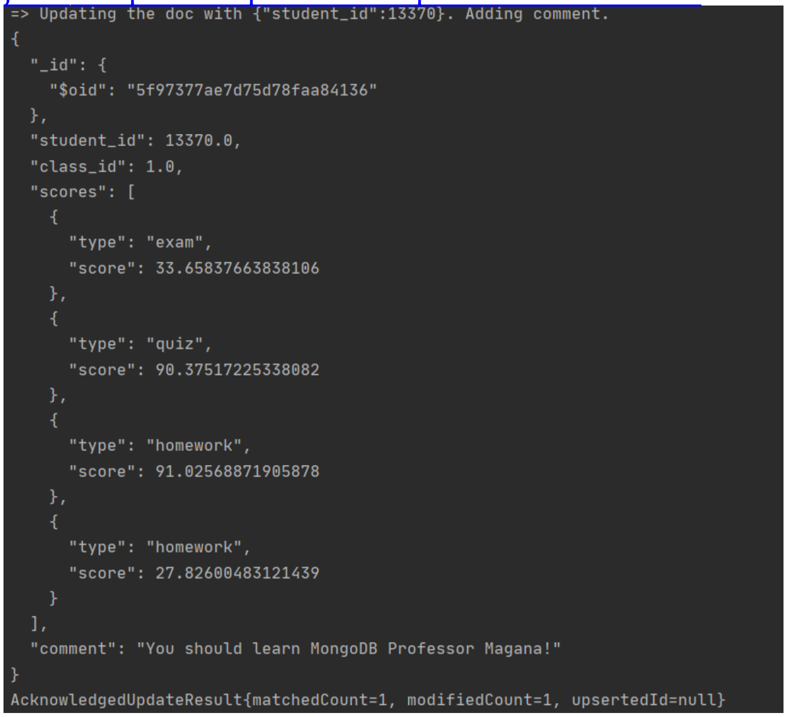
- **Step 12:** Skip down to https://developer.mongodb.com/quickstart/java-setup-crud-operations#read-operations and **use the "Read With Filter" Read. java class.** Modify it so it reads just your 1 additional student with the student_id you chose (in step 11, I chose student_id of 13370). **Take a screenshot of your output for the report.** It should look something like this:



```
INFO: Monitor thread successfully connected to server with description ServerDescription{address=cluster0-shard-00-01.jhpsj.mongodb.net:27017, type=REPLICA_SET_SECONDARY, state=C
Oct 26, 2020 4:01:10 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Setting max election id to 7fffffff0000000000000004 from replica set primary cluster0-shard-00-02.jhpsj.mongodb.net:27017
Oct 26, 2020 4:01:10 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Setting max set version to 1 from replica set primary cluster0-shard-00-02.jhpsj.mongodb.net:27017
Oct 26, 2020 4:01:10 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Discovered replica set primary cluster0-shard-00-02.jhpsj.mongodb.net:27017
Oct 26, 2020 4:01:11 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:4, serverValue:65940}] to cluster0-shard-00-02.jhpsj.mongodb.net:27017
Student 1: {"_id": {"$oid": "5f97377ae7d75d78faa84136"}, "student_id": 13370.0, "class_id": 1.0, "scores": [{"type": "exam", "score": 33.65837663838106}, {"type": "quiz", "score"
Oct 26, 2020 4:01:11 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Closed connection [connectionId{localValue:4, serverValue:65940}] to cluster0-shard-00-02.jhpsj.mongodb.net:27017 because the pool has been closed.

Process finished with exit code 0
```

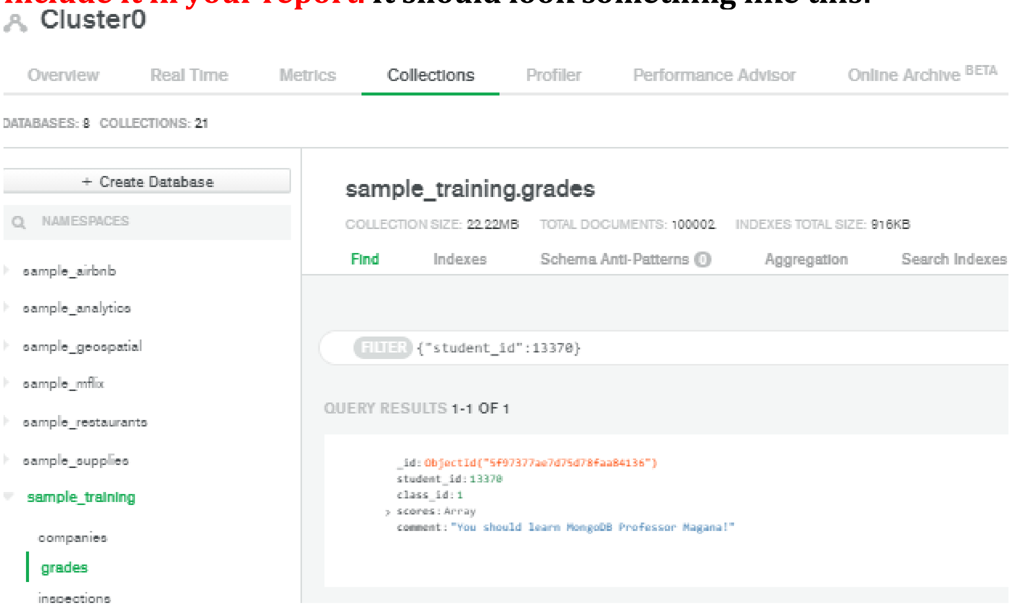
- **Step 13:** Note that you can create indexes in MongoDB as explain here: https://developer.mongodb.com/quickstart/java-setup-crud-operations#indexes and can modify you read class as noted here: https://developer.mongodb.com/quickstart/java-setup-crud-operations#the-final-code-to-read-documents

- **Step 14:** Use the Update class to modify your student created in Step 11 (not student_id 10000 as in the example, use the id you chose in step 11). Be sure to change to comment a bit as well: https://developer.mongodb.com/quickstart/java-setup-crud-operations#update-one-document



```
=> Updating the doc with {"student_id":13370}. Adding comment.
{
  "_id": {
    "$oid": "5f97377ae7d75d78faa84136"
  },
  "student_id": 13370.0,
  "class_id": 1.0,
  "scores": [
    {
      "type": "exam",
      "score": 33.65837663838106
    },
    {
      "type": "quiz",
      "score": 90.37517225338082
    },
    {
      "type": "homework",
      "score": 91.02568871905878
    },
    {
      "type": "homework",
      "score": 27.82600483121439
    }
  ],
  "comment": "You should learn MongoDB Professor Magana!"
}
AcknowledgedUpdateResult{matchedCount=1, modifiedCount=1, upsertedId=null}
```

- **Step 15**: Verify the change on the MongoDB website**. Take a screenshot and include it in your report. It should look something like this:**

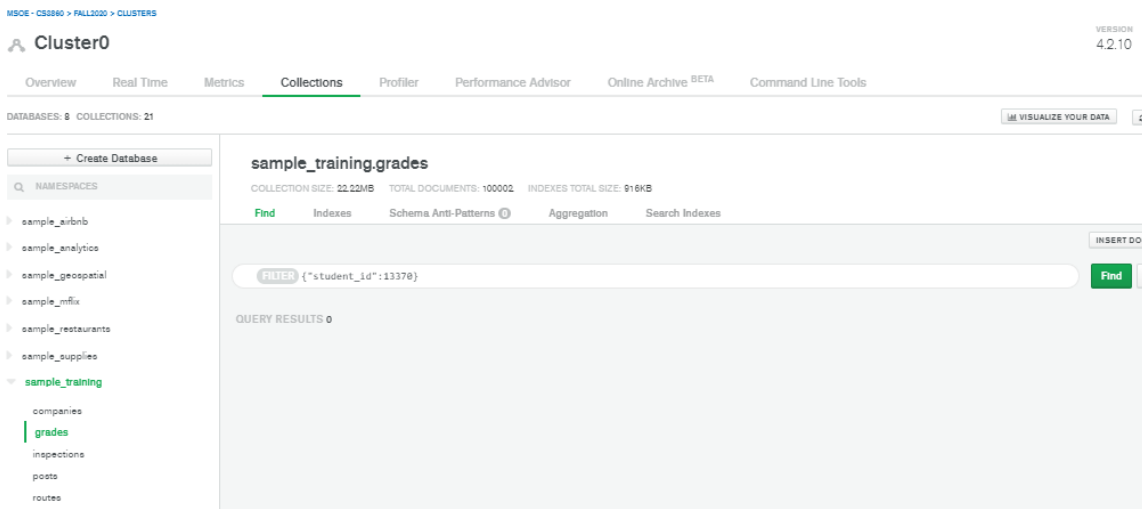


```
{"student_id":13370}
```

QUERY RESULTS 1-1 OF 1

```
_id: ObjectId("5f97377ae7d75d78faa84136")
student_id: 13370
class_id: 1
> scores: Array
comment: "You should learn MongoDB Professor Magana!"
```

- **Step 16:** Skim the rest of the update section, then move on the Delete section. **Use the delete class based on the "Find and Delete One" code** (https://developer.mongodb.com/quickstart/java-setup-crud-operations#delete-one-document), modify it to find the student with the id you created in Step 11, run it, and verify on mongodb.com that the student is deleted from your cluster**. Take a Screenshot for your report (and that is the last screenshot of the report).**



DATABASES: 8 COLLECTIONS: 21

sample_training.grades

COLLECTION SIZE: 22.22MB TOTAL DOCUMENTS: 100002 INDEXES TOTAL SIZE: 916KB

FILTER `{"student_id":13370}`

QUERY RESULTS 0

- **Step 17: Include the screenshots noted above (both the one from Part 1 and the 4 in Part 2) along with a brief description of what each screenshot shows. Submit your lab in pdf form on Canvas.**