Big Data, Techniques, and Platforms

Document Databases & Structured Query Language

The purpose of this tutorial is to work with MongoDB which is a NoSQL document database [1] server installed in your system (part 1) or on the cloud (part 2).
Notice that :
— **Part 1** of the tutorial is mandatory
— **Part 2** is an extra

Studio 3T [2] will be used as a client to connect to a MongoDB server instance. Notice that the trial license will expire within 30 days but you can ask for an academic license at this link using your student address :

Studio3T request for academic license.

We will use a dataset that consists of a collection of movies. All necessary files are :
— The subject file EX-MongoDB-2021.pdf
— One data file : moviesEmbedded.json
— One data file : moviesReferences.json
— One data file : artists.json.

Download the files and store them in a directory of your computer. I suggest you to create a directory called data.

1 MONGODB AND NOSQL DOCUMENT MODEL

MongoDB is a document datastore. A MongoDB server can manage several databases. A *database* is composed of a set of *collections*. A collection is a set of documents and each *document* has a unique identifier, the _id. Unlike the relational databases, MongoDB allows a flexible schema, meaning that it is not necessary to define a schema before adding data. A document can contain another document (i.e., an *embedded* document) or reference another document by using its identifier.

Data are represented as JSON (Javascipt Object) documents, a lightweight data-interchange format, where a document is an ordered set of key-value pairs.
— Here is an extract of a collection of movies, where the documents describing the artists (directors and actors) are embedded in the documents.

---

1. https://www.mongodb.com/fr/what-is-mongodb
2. https://studio3t.com/features/

```
[{
"_id": "movie:1",
"title": "Vertigo",
"year": 1958,
"genre": "drama",
"summary": "Scottie Ferguson, ancien inspecteur de police, ... ",
"country": "DE",
"director":  {
"_id": "artist:3",
"last_name": "Hitchcock",
"first_name": "Alfred",
"birth_date": "1899"
},
"actors": [{
"_id": "artist:15",
"first_name": "James",
"last_name": "Stewart",
"birth_date": "1908",
"role": "John Ferguson"
}, ...
{...
}]
}, ...]
```
— Here is an extract of a collection of movies, where a document describing a movie contains the references to artists (director and actors) described in external documents.
```
[{
"_id": "movie:1",
"title": "Vertigo",
"year": 1958,
"genre": "drama",
"summary": "Scottie Ferguson, ancien inspecteur de police, ... ",
"country": "DE",
"director":  {
"_id": "artist:3"
},
"actors": [{
"_id": "artist:15",
"role": "John Ferguson"
},
...
{
"_id": "artist:282",
"role": null
}]
}, ...]
```
Here is a document that describes an artist.
```
{
"_id": "artist:15",
"last_name": "Stewart",
"first_name": "James",
"birth_date": "1908"
}
```
— List the differences between the two representations of the data. Compare with the relational model.
— Describe when the second data representation is preferred with respect to the first one.
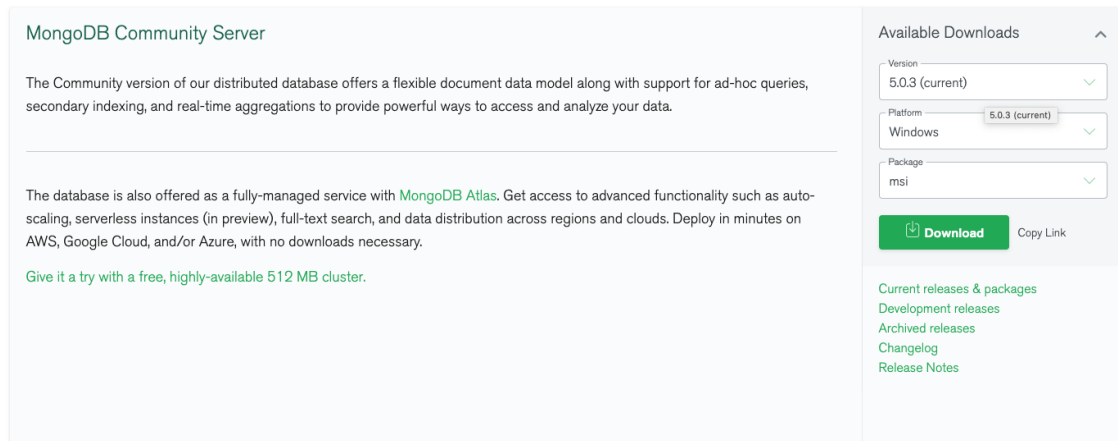
2 Install MongoDB server

First, you need to install MongoDB Community Edition. The instructions are available at :
— Linux Install
— MacOS Install
— Windows Install.

2.1 Windows installation

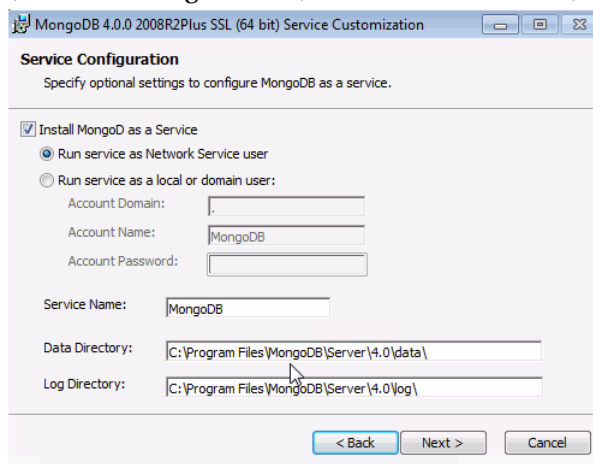Here is a small guide from the documentation thanks to the contribution of Yuting Feng (thanks a lot).

1) Choose the correct version to download :

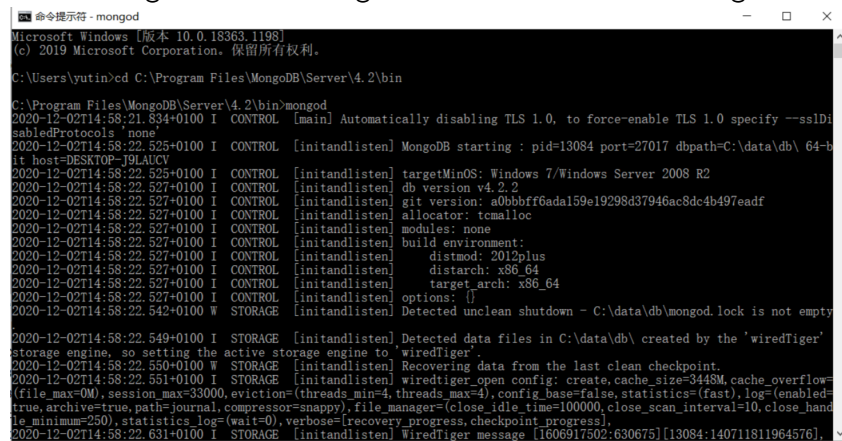2) Double click .msi documents
3) Complete setup type (recommended)
4) Service configuration (default recommended)

5) Don't install MongoDB compass (recommended) (since we are going to use Studio 3T)

6) Open a cmd terminal using the wollowing command

```
cd C:\Program Files\MongoDB\Server\5.0.3\bin mongod
```


8) Open another new cmd terminal

```
cd C:\Program Files\MongoDB\Server\5.0.3\bin mongo
```


9)Now the mongo server is on, and the database is open, you can jump to Section 5.

3 HOW TO START WITH MONGODB SERVER

Launch an instance of MongoDB server, following the procedure needed by your machine [3]
— If not already done when you dowloaded the files of the exercise, create a directory having the name `"data"` on your machine. This directory has a unique location in the file system of your machine. This location is called `PATH`.
— Open a terminal to run an instance of MongoDB server by typing the command required by your system. REMEMBER : keep this terminal open otherwise the launched instance will be stopped. For example in unix-like system the command to run in order to start the MongoDB server is :
```
mongod --dbpath /PATH/data --port 27018
```
The option `port` is used to specify the port the server listens to. In this example MongoDB server will listen at the port 27018. The option `dbpath` is used to specify the directory where the data files will be stored. Each MongoDB instance manages its own data directory. For each MongoDB instance you must specify a new data directory.

4 HOW TO START WITH MONGODB CLIENT

Launch an instance of MongoDB client, following again the procedure needed by your machine [4]
— Open a second terminal. To run an instance of MongoDB client by typing the command required by your system. REMEMBER : In unix-like system the command to run in order to start the MongoDB client is :
```
mongo --port 27018
```

5 INTERACT WITH MONGODB USING THE CLIENT

You can create/access a database in MongoDB using the `use` command. To create a database that is called `cities` the command to run is :

```
use cities
```

If you want to check which database are you using you must type the `db` command :

```
db
```

To delete a database and its content the command to use is `db.dropDatabase()`. The `show dbs` command will show you the updated database list.
You can create a collection using the `createCollection()` command. For example to create the collection `people` you can run the following command :

```
db.createCollection('people')
```

You also have the command that shows you the already defined collections :

```
show collections
```

To create a document inside a collection the command `insert` is available.

---

3. Linux, MacOS, and Windows.
4. Linux, MacOS, and Windows.

```
db.people.insert({name: "kelly", age: 34, email: "kelly@kelly.com"})
```

The ObjectID will be automatically created. In order to check the element inside a collection the `find` command is available. In this case in order to check the content of the `people` collection you can type :

```
db.people.find()
```

```
(function() {
  var names = [
    'Dave',
    'Alex',
    'Jenny',
    'Frank',
    'Patty',
    'Mario',
    'Jack',
    'Tom',
    'Ralf',
    'Rick'
  ]
  var randName = function() {
    var n = names.length;
    return names[Math.floor(Math.random() * n)];
  }
  var randAge = function(n) {
    return Math.floor(Math.random() * n);
  }
  for (var i = 0; i < 1000; ++i) {
    var person = {
      name: randName(),
      age: randAge(100)
    }
    if (Math.random() > 0.8) {
      person.cat = {
        name: randName(),
        age: randAge(18)
      }
    }
    db.people.insert(person);
  };
})();
```

Copy/paste the code in your Mongodb client console. This code will populate the collection `people`.

Using the manual reference `http://docs.mongodb.org/manual/reference/operator/query/` :
  — Provide the MongoDB query that finds the list of people whose name is "Jenny".
  — Provide the MongoDB query that finds the list of people older than 45.
  — Provide the MongoDB query that finds the list of people older than 45 whose name is `Jenny`.
  — Provide the MongoDB query that checks if there is a person who has a `nickname`.
Run the following command :

```
db.people.insert({name: "kelly", nickname:"Penny", age: 34, email: "kelly@kelly.com"})
```

Run again the query that returns the elements of `people` collection and verify that the new document has been correctly stored.

Notice that in this run the port is 27017 :

```
Last login: Tue Nov  2 10:38:48 on ttys000
[(base) funny-ecran:~ francy$ mongo —version
MongoDB shell version v5.0.3
connecting to: mongodb://127.0.0.1:27017/%E2%80%93version?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("9af3e7de-e0f4-41db-8ac1-9d15b6f81f29") }
MongoDB server version: 5.0.3
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
```

— Provide the MongoDB query that lists just the name of people older than 98.
— Provide the MongoDB query that counts how many people are older than 56.

6 IMPORT DATA

Open a third terminal to import the data `moviesEmbedded.json`.
This terminal is an empty terminal of your system (no databases, no servers running in this terminal).
In linux-like systems you can use the following procedure. Pay attention : before running this command verify that your mongodb server is listening at port 27018, if it is not the case you must specify the correct port number in the command :

```
mongoimport --db movie --collection moviesEmbedded --drop  --jsonArray --file  /PATH/moviesEm
```

Where `PATH` is the path to access to `moviesEmbedded.json` file you downloaded.
The option `db` creates a database named `movie`, the option `collection` creates the collection named `moviesEmbedded` which will store the documents imported from file `moviesEmbedded.json`.
The option `drop` removes the collection if it already exists.

7 INTERACT WITH MONGODB USING STUDIO3T

To query MongoDB database, it is also possible to use a client. We use Studio 3T's through IntelliShell. This is a client that makes easier to interact with MongoDB, to show in a better format the result of the query, to inspect the database.

— Launch Studio 3T and create a new connection to MongoDB server by specifying its location as `localhost` and the port number (normally it is 27018). Then open Intellishell to edit/execute queries.
— You can open the `IntelliShell` and run the same commands you run using the `mongo shell`:
    — to get the list of the databases `show databases` command,
    — to access to the target database `use movie` command,
    — to show the collections located in the current database `show collections` command,
    — to count the number of documents in the target collection `db.moviesEmbedded.count()`,
    — you can also shutdown the server (as admin) using: `use admin` and `db.shutdownServer()`.

8 QUERYING DATA

Now that you are familiar with the environment and the Mongodb query syntax find the queries to retrieve the following data :

1. The movies titled `Gladiator`.

2. The distinct genre values of movies.

3. The movies of `crime` or `drama` genre.

4. The movies released between 1967 and 1995.

5. The list of the movies released between 1967 and 1995, by displaying only title, year, director's last name sorted by year.

6. The same query by adding actor's last name.

7. The number of movies by country.

8. The number of movies by country and by year.

9. The number of movie of genre `drama` by country.

10. The number of movies then by country and director.

11. The number of movies by actor.

12. The movies in which the artist `John Travolta` has played a role.

13. The number of movies in which the artist `John Travolta` has played a role.

9 USING MANUAL REFERENCES

Import a second dataset that codify the same data using references :
  — Open a terminal to import the data `artists.json`. In linux-like systems you can use the following procedure :

```
mongoimport --db movie --collection artists --drop
--jsonArray --file  /PATH/artists.json --port 27018
```

Where `PATH` is the path to access to `artists.json` file you downloaded. The option `db` creates a database named `movie`, the option `collection` creates the collection named `artists` which will store the documents imported from file `artists.json`. The option `drop` removes the collection if it already exists.
  — Import in the same terminal the data `moviesReferences.json`. In linux-like systems you can use the following procedure :

```
mongoimport --db movie --collection moviesReferences --drop
--jsonArray --file  /PATH/moviesReferences.json --port 27018
```

Where `PATH` is the path to access to `moviesReferences.json` file you downloaded. The option `db` creates a database named `movie`, the option `collection` creates the collection named `moviesReferences` which will store the documents imported from file `moviesReferences.json`. The option `drop` removes the collection if it already exists.

Refresh now the IntelliShell view. You will see that you have two new available collections.
Now we try to interact with the data. Explore them running the `find` operation.

1. In the collection `moviesReferences` how are the directors stored?

2. In the collection `moviesReferences` how are the actors stored?

3. Write the query that returns all the movies directed by `Hitchcock` as new documents containing all the data of the director in a separate field.

10 MONGODB IN THE CLOUD

Now that you are familiar with MongoDB and how NoSQL databases works you can try to use MongoDB on a cloud environment and put together what we have seen about data distribution and what we have seen about Databases.

10.1 MAIN OBJECTIVE

In this part of the exercise class you can try to set-up your own cluster and see how to use MongoDB instance in the cluster from your machine.

10.2 ATLAS

MongoDB Atlas is a multi-cloud database service provided by the same people that build MongoDB. MongoDB Atlas makes it easy to deploy and manage databases on-demand with MongoDB on AWS, Azure, and Google Cloud. You can access the service and having a more deep look to what is provided using the following link :

Atlas

10.3 IMPORTANT

In the set-up of the machines and of the cluster you ALWAYS have to select the free options. For running this class IS NOT required to buy any subscription or set-up any payment in the connection to the cluster.
— you will lunch a free cluster
— you will choose your provider and region
— you will run your database
— you will query from your machine.

10.4 CREATE AN ATLAS ACCOUNT OR LOGIN

As first step you must log-in to Atlas and for this step you can use the same credentials you used for downloading your MongoDB community Version.

```
https://cloud.mongodb.com
```

The fist step will be to login to the system as shown in the following picture.
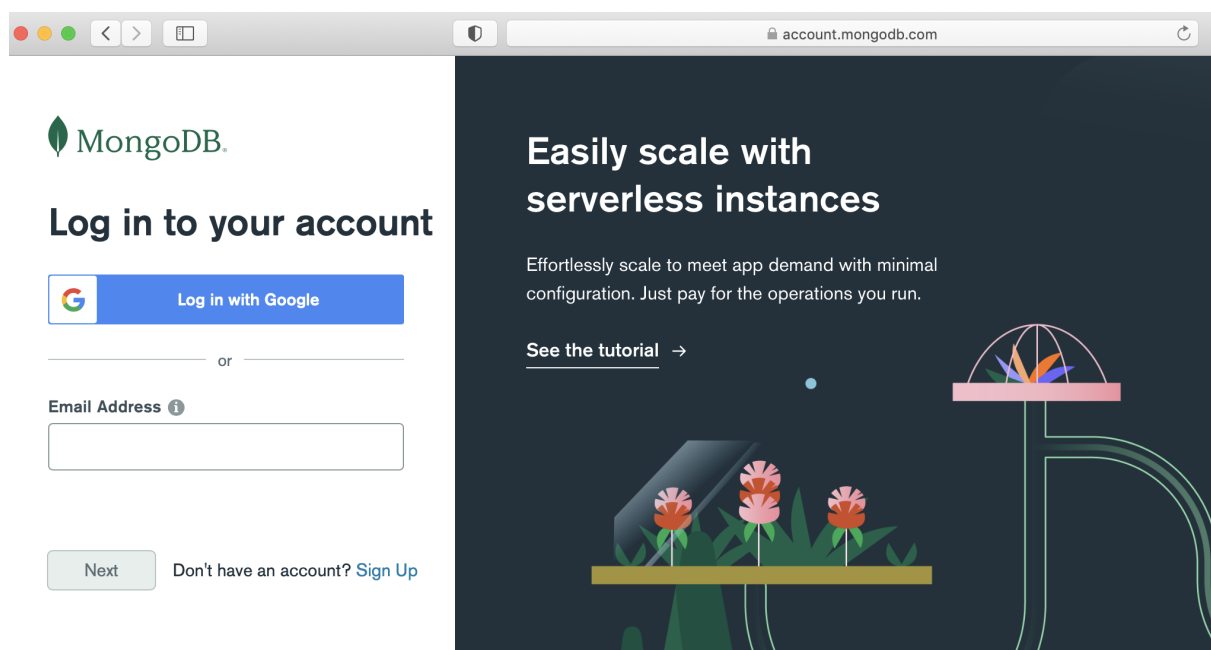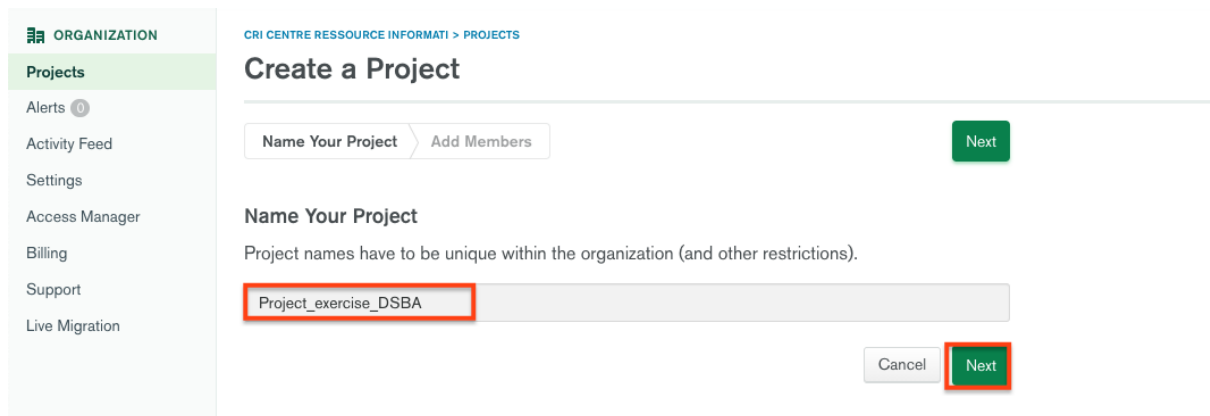
10.5 DEPLOY A **FREE** CLUSTER

Atlas free clusters provide a development environment to host your data and provide access to a subset of Atlas features that are sufficient for this class.
In order to deploy your cluster you must apply the following steps.
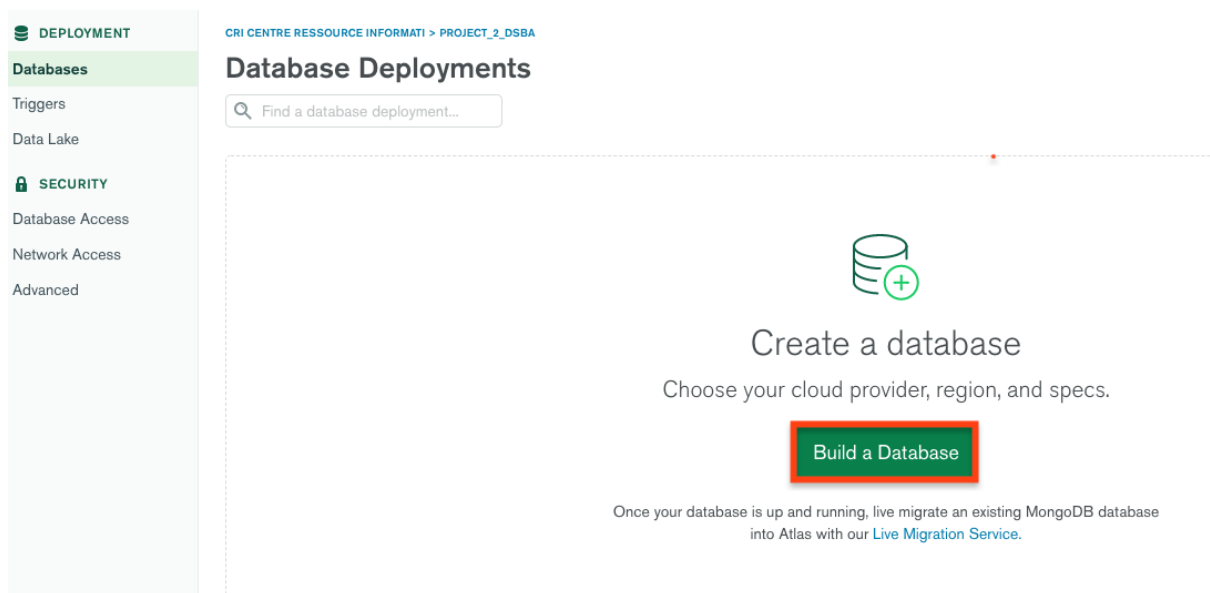
1) Create a new project.
Include the **Name**, in this example **Project_Exercise_DSBA** :

2) Select to deploy a cluster for MongoDB in the following window :

3) Select **Shared** Clusters and click **Create a Cluster** :

4) Select your preferred Cloud Provider & Region and clic Create Cluster :

Do not change the default options for the Cluster Tier (MO Sandbox) and do not select additional settings.

5) Optionally change the cluster name :

6) Finally click on Create Cluster :
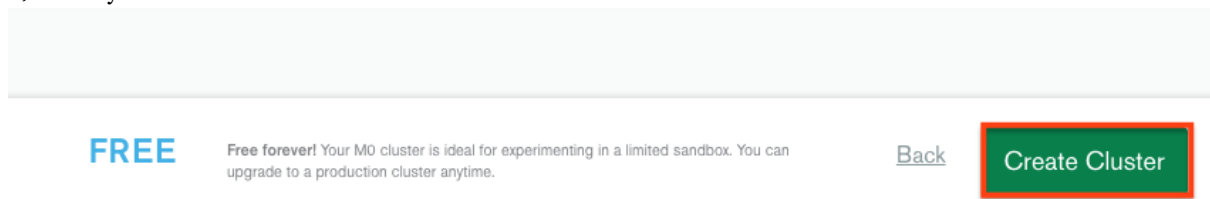
The deployment will start :
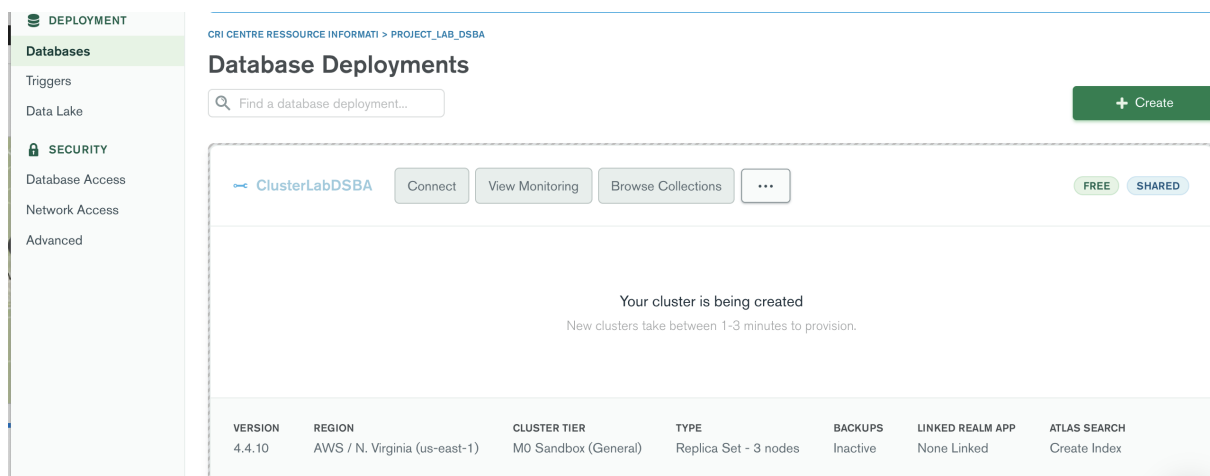
And finally you will have your cluster running as shown in this picture :
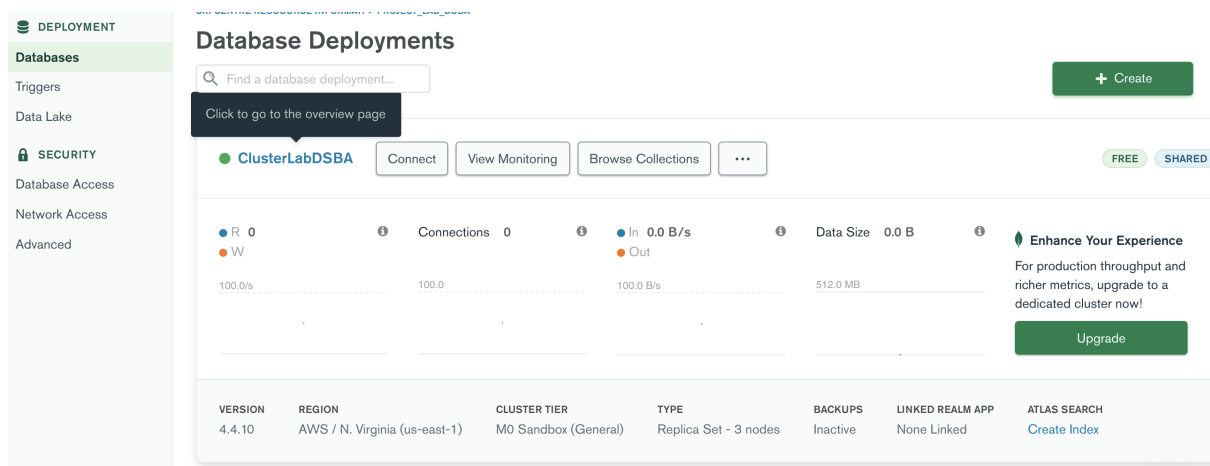
You can start looking at this interface and see the concepts we have already seen at lesson : the region, the replica set (equal 3 by default).

10.6 To configure an IP address

At this point you must configure how and who can access the cluster from the outside. In Atlas, you can only connect to a cluster only from a trusted IP address. Then you must create a list of trusted IP addresses, referred to as a IP access list, that can be used to connect to your cluster and access your data.

1) Click **Connect** in the top-left corner of Atlas :
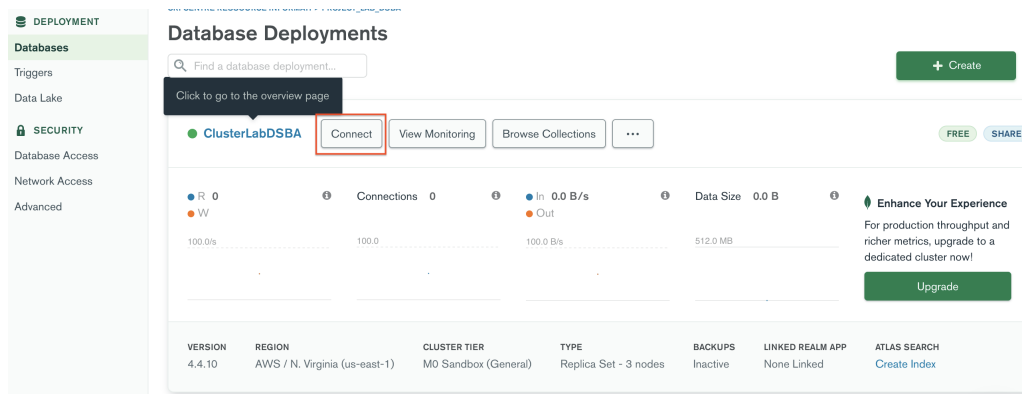
2) Click **Add Your Current IP Address** :
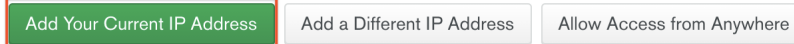
Your IP address will be automatically discovered and you can add it clicking on **Add IP address**.

After you confirm clicking on **Add IP Address** you will have the following feedback :

10.7 ADD A DATABASE USER

You must create a database user to access your cluster. For security purposes, Atlas requires clients to authenticate as MongoDB database users to access clusters. In this step you must create a database user with username and password.

Add you data into the form and confirm by clicking on **Create database user** :

After you confirm clicking on **Create database user** you will have the following feedback :

10.8 DATABASE CONNECTION

You can create a database user to access your Atlas database deployment in the Connect dialog or you can also add the database user from the Database Deployment view following the previous steps and clicking on the option Choose a connection method.

In the code of your application shows :

— The `<username>`) of the user that will connect to the database
— The string `<password>` to be replaced **in the code** with the password specified when you created your database user.
— Replace `myFirstDatabase` with the name of the database that connections will use by default. Notice that if you omit the database, the test database is used by default.

1) Select your driver and version from the dropdown menus. The code sample containing a connection string will display.

In the following we selected `Python` as language and `3.6 or later` as version :

×

Connect to ClusterLabDSBA

✔ Setup connection security ⟩ ✔ Choose a connection method ⟩ Connect

1 Select your driver and version

DRIVER
| Python ▼ |

VERSION
| 3.6 or later ▼ |

2 Add your connection string into your application code

☐ Include full driver code example

```
mongodb+srv://franbugiotti:
<password>@clusterlabdsba.jgafb.mongodb.net/myFirstDatabase?
retryWrites=true&w=majority
```

Replace **<password>** with the password for the **franbugiotti** user. Replace **myFirstDatabase** with the name of the database that connections will use by default. Ensure any option params are URL encoded.

Having trouble connecting? View our troubleshooting documentation

The <username to be changed in your code :

Connect to ClusterLabDSBA

✔ Setup connection security ✔ Choose a connection method Connect

1 Select your driver and version

DRIVER   VERSION
Python   ▼   3.6 or later   ▼

2 Add your connection string into your application code

☐ Include full driver code example

```
mongodb+srv://<username>:
<password>@clusterlabdsba.jgafb.mongodb.net/myFirstDatabase?
retryWrites=true&w=majority
```

Replace **<password>** with the password for the **<username>** user. Replace **myFirstDatabase** with the name of the database that connections will use by default. Ensure any option params are URL encoded.

Having trouble connecting? View our troubleshooting documentation

The <password> string to be replaced in your code :

Connect to ClusterLabDSBA

✔ Setup connection security ✔ Choose a connection method Connect

1 Select your driver and version

DRIVER   VERSION
Python   ▼   3.6 or later   ▼

2 Add your connection string into your application code

☐ Include full driver code example

```
mongodb+srv://<username>:
<password>@clusterlabdsba.jgafb.mongodb.net/myFirstDatabase?
retryWrites=true&w=majority
```

Replace **<password>** with the password for the **<username>** user. Replace **myFirstDatabase** with the name of the database that connections will use by default. Ensure any option params are URL encoded.

Having trouble connecting? View our troubleshooting documentation

Go Back   Close

The database to be connected to (to be replaced with your target) :

Connect to ClusterLabDSBA   ✕

✔ Setup connection security   ✔ Choose a connection method   Connect

1 Select your driver and version

DRIVER   VERSION
Python   ▼   3.6 or later   ▼

2 Add your connection string into your application code

☐ Include full driver code example

```
mongodb+srv://<username>:
<password>@clusterlabdsba.jgafb.mongodb.net/myFirstDatabase?
retryWrites=true&w=majority
```

Replace **<password>** with the password for the **<username>** user. Replace **myFirstDatabase** with the name of the database that connections will use by default. Ensure any option params are URL encoded.

Having trouble connecting? View our troubleshooting documentation
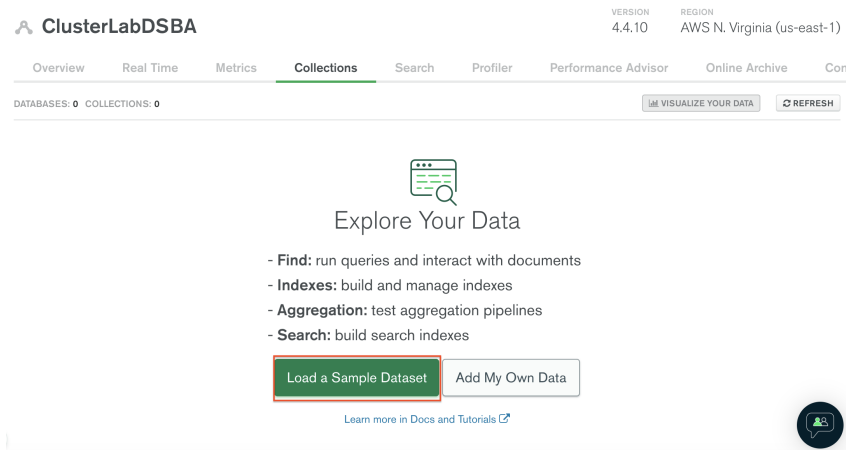
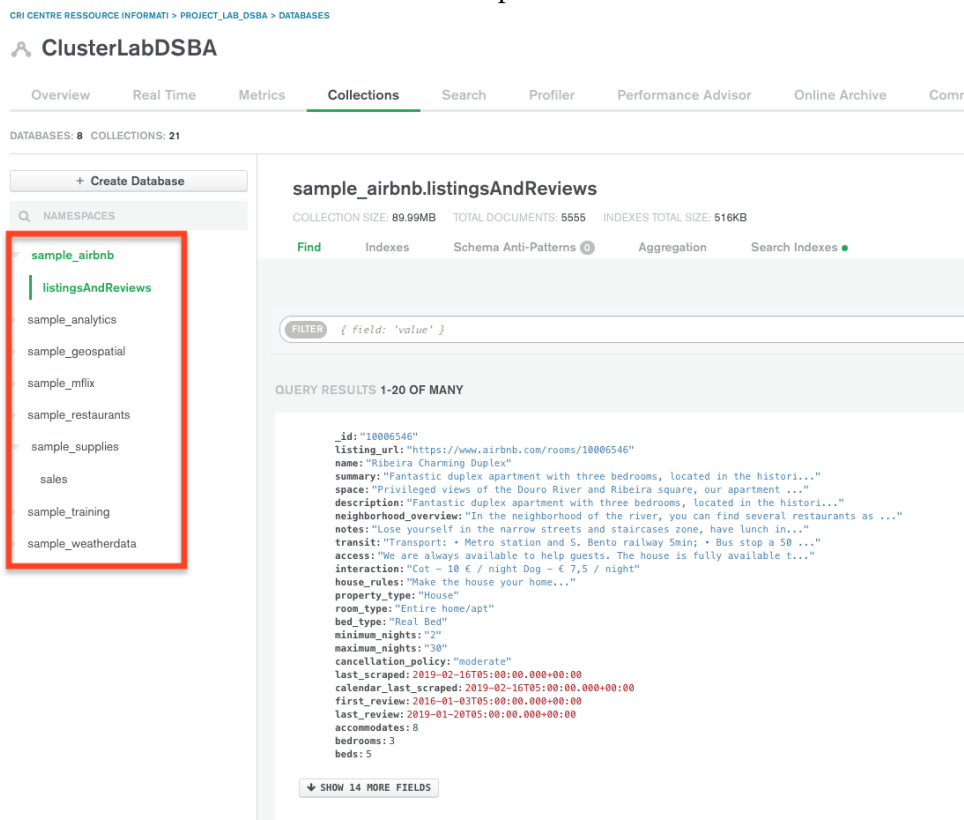Go Back   Close

10.9 LOAD SAMPLE DATA

Atlas provides sample data you can load into your Atlas clusters. Otherwise you can use the **Migrate or Import Data** option.

We suggest you to start with sample data and you can perform this import clicking on **Load Sample Dataset** :

Here the default database list after the import :

And the **sample_supplies** to which we connect with the provided notebook in the following :

⋏ ClusterLabDSBA

Overview	Real Time	Metrics	**Collections**	Search	Profiler

DATABASES: **8** COLLECTIONS: **21**

+ Create Database

🔍 NAMESPACES

▷ sample_airbnb

▷ sample_analytics

▷ sample_geospatial

▷ sample_mflix

▷ sample_restaurants

sample_supplies   ⊕ 🗑

   sales

▷ sample_training

▷ sample_weatherdata

sample_supplies.sales

COLLECTION SIZE: **4.13MB**   TOTAL DOCUMENTS: **5000**   INI

Find   Indexes   Schema Anti-Patterns ⓪

(FILTER)   { field: 'value' }

QUERY RESULTS **1-20 OF MANY**

```
_id: ObjectId("5bd761dcae323e45a93ccfe8")
saleDate: 2015-03-23T21:06:49.506+00:00
> items: Array
storeLocation: "Denver"
> customer: Object
couponUsed: true
purchaseMethod: "Online"
```

```
_id: ObjectId("5bd761dcae323e45a93ccfe9")
saleDate: 2015-08-25T10:01:02.918+00:00
> items: Array
storeLocation: "Seattle"
> customer: Object
couponUsed: false
purchaseMethod: "Phone"
```

Notice that you can run simple queries using filters in this form :

Find	Indexes	Schema Anti-Patterns ⓪	Aggregation	Search Indexes ●

(FILTER)   { field: 'value' }

QUERY RESULTS **1-20 OF MANY**
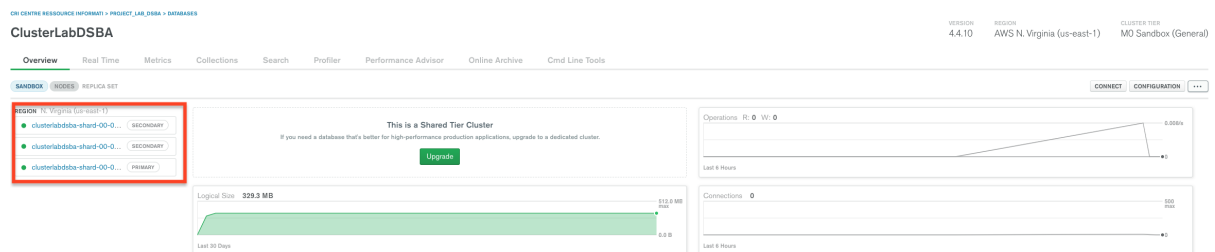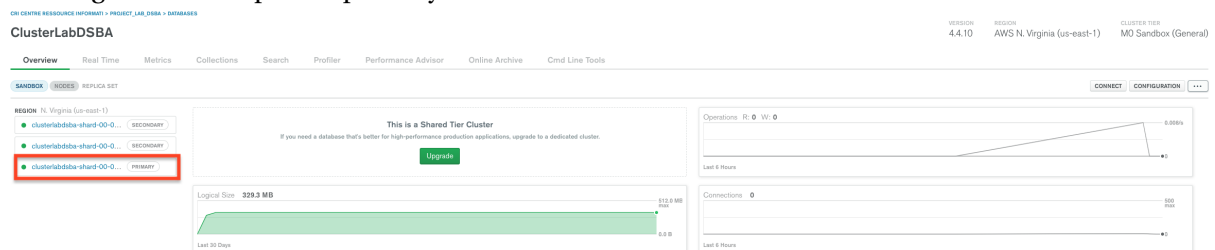
10.10 LOOK AT THE DASHBOARD

You can have a look at the dash board and set-up some considerations about the cluster usage. A strong suggestion is to check the dashboard right now and verify how the cluster react to write and read queries. A write query is performed when you import your dataset for example.

Check the cluster shards :

Investigate for example the primary :

And check how write and/or read queries affect the cluster :

⊙ **clusterlabdsba-shard-00-02.jgafb.mongodb.net:27017**

VERS
4.4.

Status

GRANULARITY Auto ∨ ZOOM 1 hour ∨ CURRENT DISPLAY 🗓 11/02/2021 🕐 08:33am to 🗓 11/02/2021 🕐 09:33am AT **1 MINUTE** GRANULARITY

EXPORT

▦ ≡   ADD CHART ∨   ☐ DISPLAY OPCOUNTERS ON SEPARATE CHARTS ☑ DISPLAY TIMELINE ANNOTATIONS

Opcounters	Connections	Network

TOGGLE CHARTS

| – Opcounters | – Connections | – Network | + Logical Size |
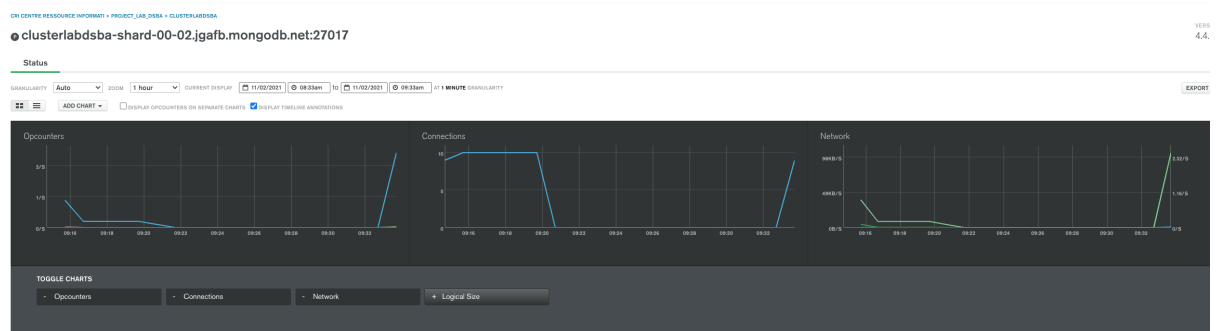
10.11 Notebook to connect to the cluster

In this part you will interact with your MongoDB database in the cluster. We provided you a first example of connection and access to the **sample_supplies** database but in the following the objective is to repeat the queries you wrote in the first part of the assignment and to check the cluster analysis. The python notebook :

— `advanced_spark_notebook_mongodb.ipynb`

While you write your queries check how the cluster is interacting.

10.12 From the documentation

You can also check the documentation and find concepts we have seen at lesson. For example how to check that :

— "Atlas does not guarantee that host names remain consistent with respect to node types during topology changes."