MongoDB Tutorial

MSBA 7024/MACC 7020

What will you learn in this tutorial?

- Create a MongoDB Atlas account and install MongoDB Compass in Windows 10 X64 / Mac
- Compare the concepts in MongoDB with MySQL
- Use MongoDB Compass to import and extract data into/from MongoDB
- Translate basic SQL to codes in MongoDB
- Use Python to import and extract data into/from MongoDB

When to use MongoDB?

- Unstructured data: Not all columns are known before database design. Adding column in SQL is not easy.
- Performance: Real-time analytics and high-speed logging
- Large data volumes: The economic cost of storing large data, especially long strings is much lower.
- Agile development and collaboration: Allow one team to control one part of a document and another team to control another part.

When not to use MongoDB?

- If you're building a simple application
- If you don't have scalability issues with traditional RDBMS
- If you don't have a specific use case to which a NoSQL database might offer a solution to.

Concepts in MongoDB and MySQL

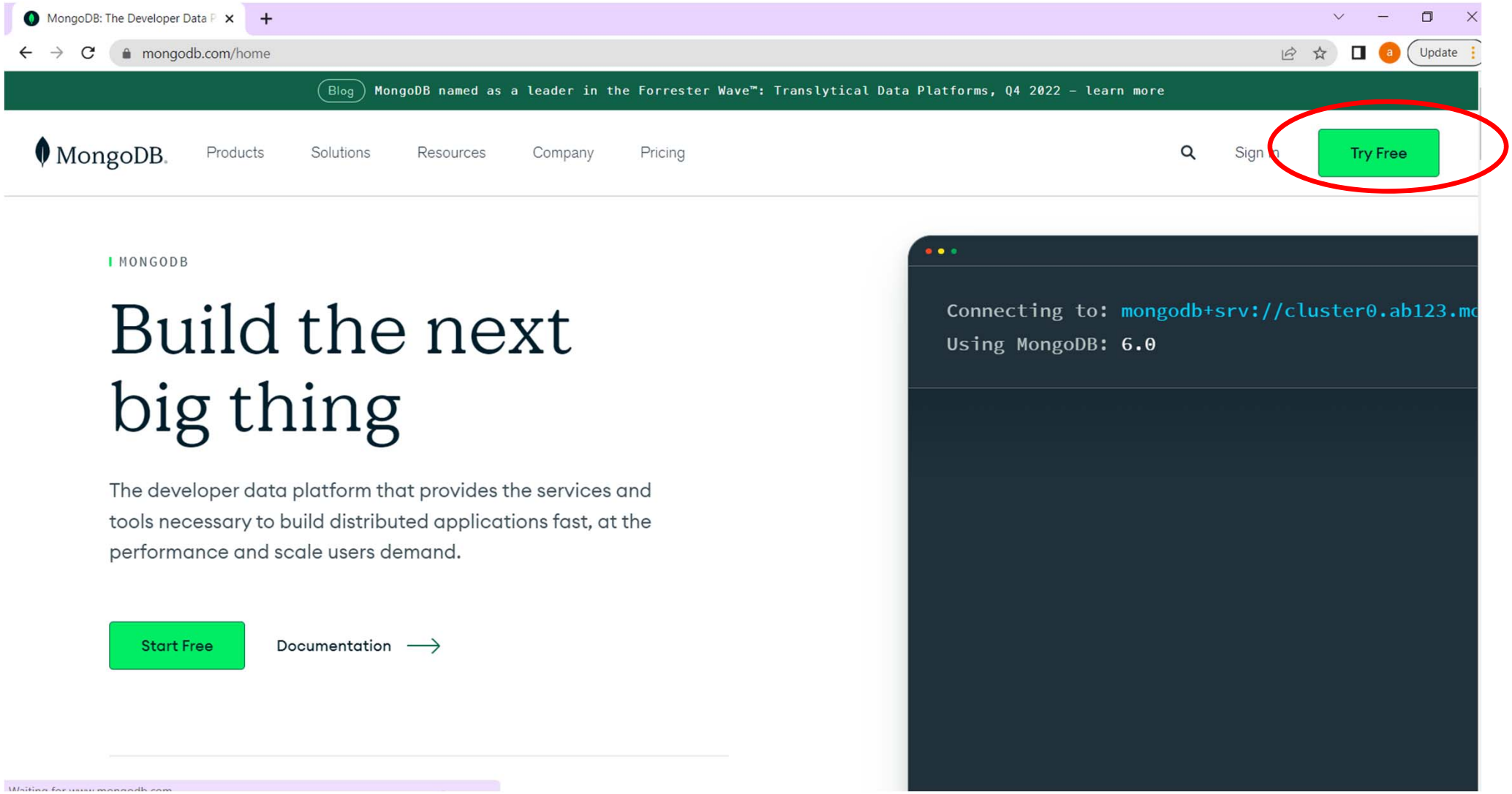
MySQL	MongoDB
Database	Database
Table	Collection
Row	Document
Column	Field
Primary Key	_id field: Auto-generated or user defined

MongoDB Setup and Installation

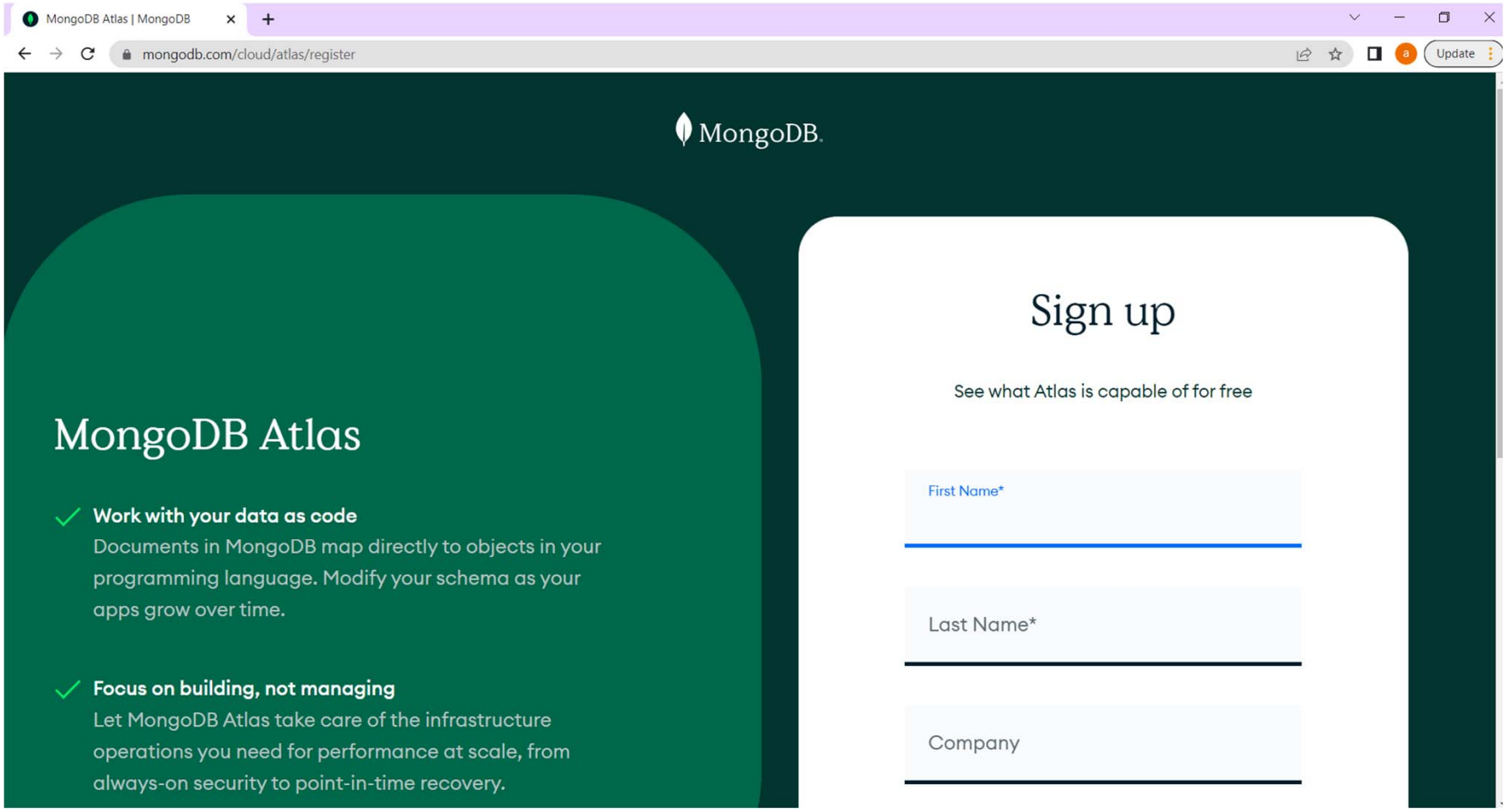
MongoDB Setup

- We will setup
  - A MongoDB account
  - A database cluster on MongoDB Atlas on the cloud
  - MongoDB Compass (the graphical user interface to connect to the database from your computer)

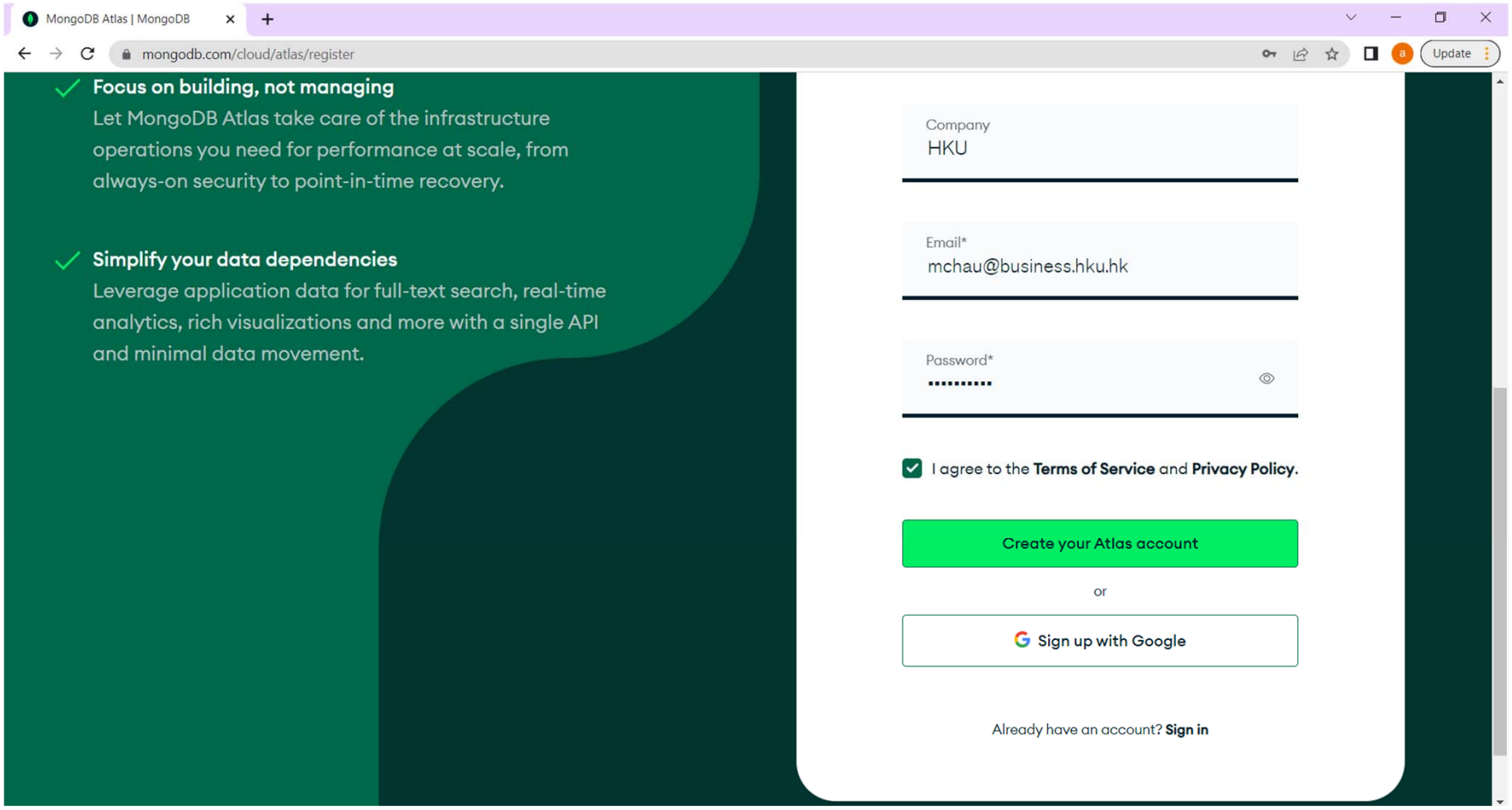
1. Open a web browser, go to mongodb.com Click “Try free” to register an account



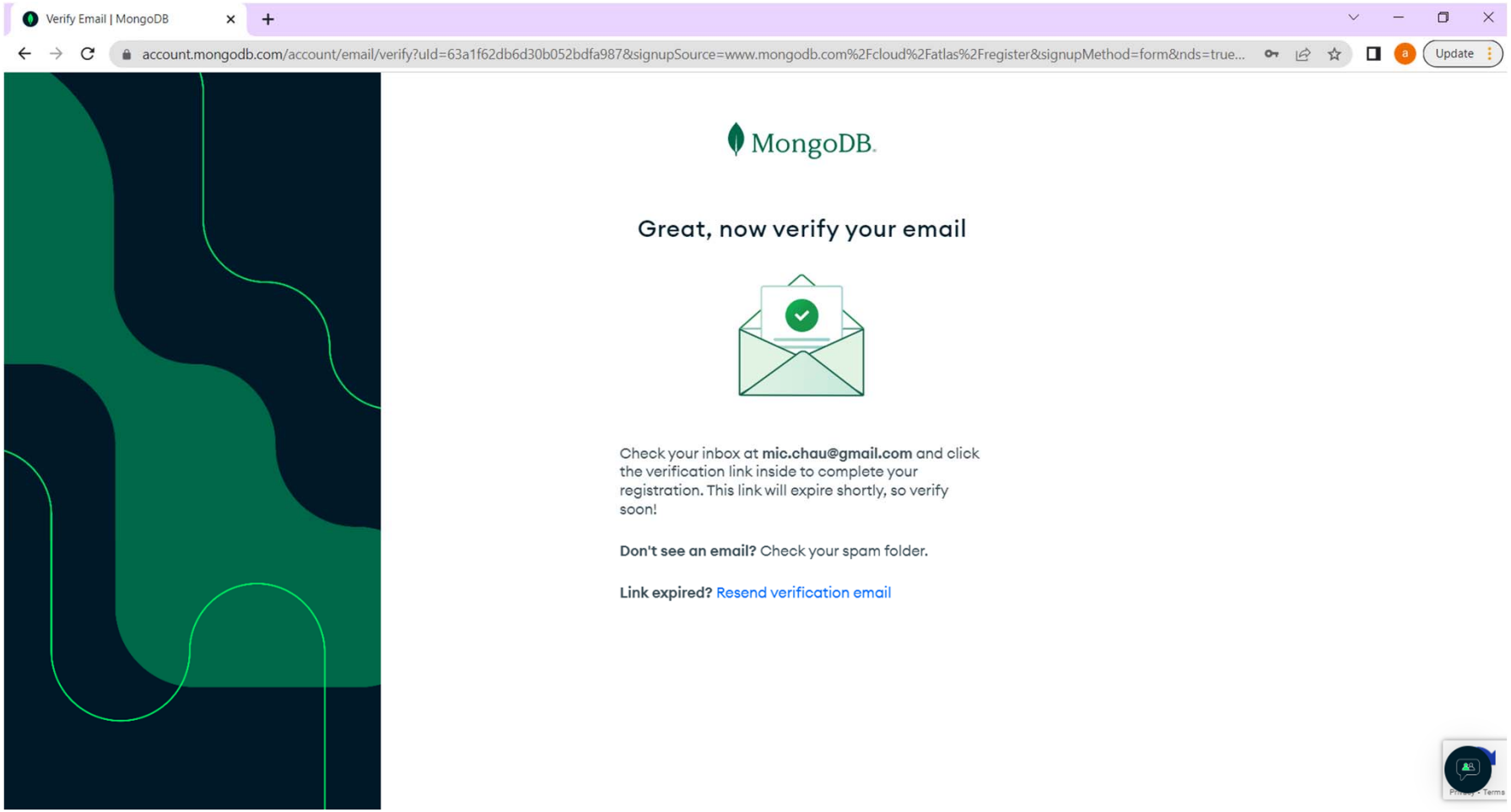
2. Fill in the information to sign up



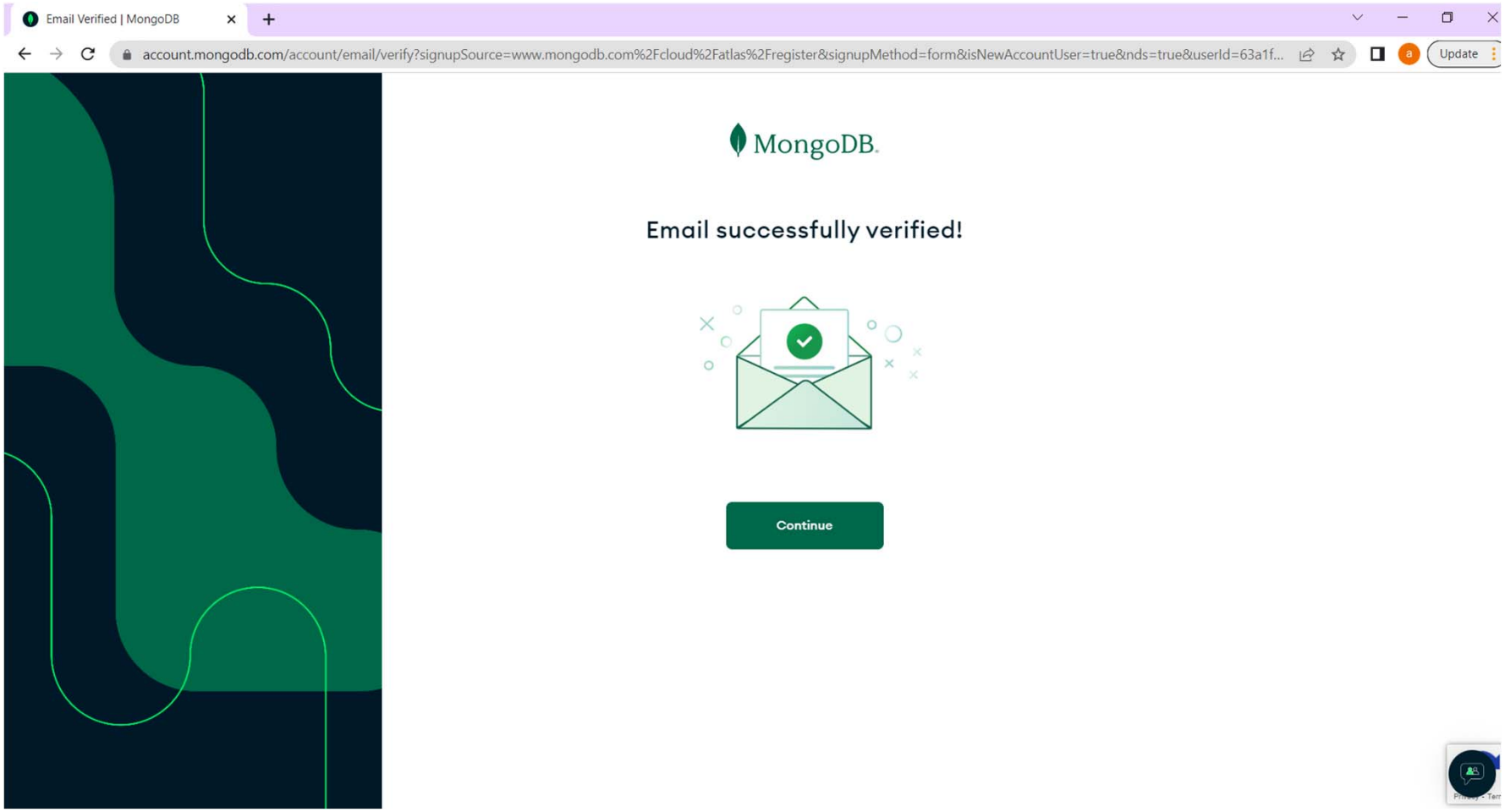
2. Fill in the information to sign up



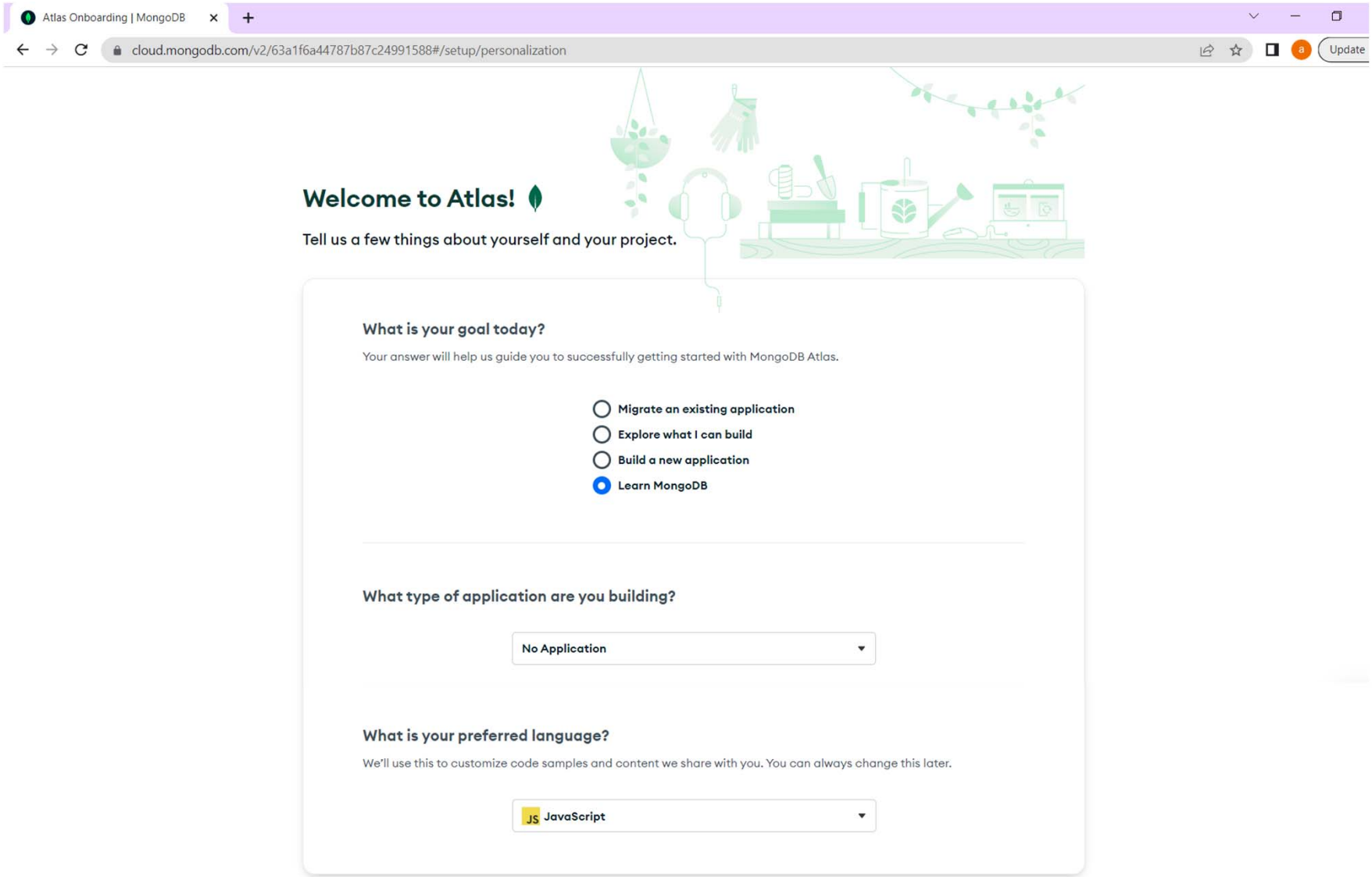
3. Login to your email account and look for the verification email



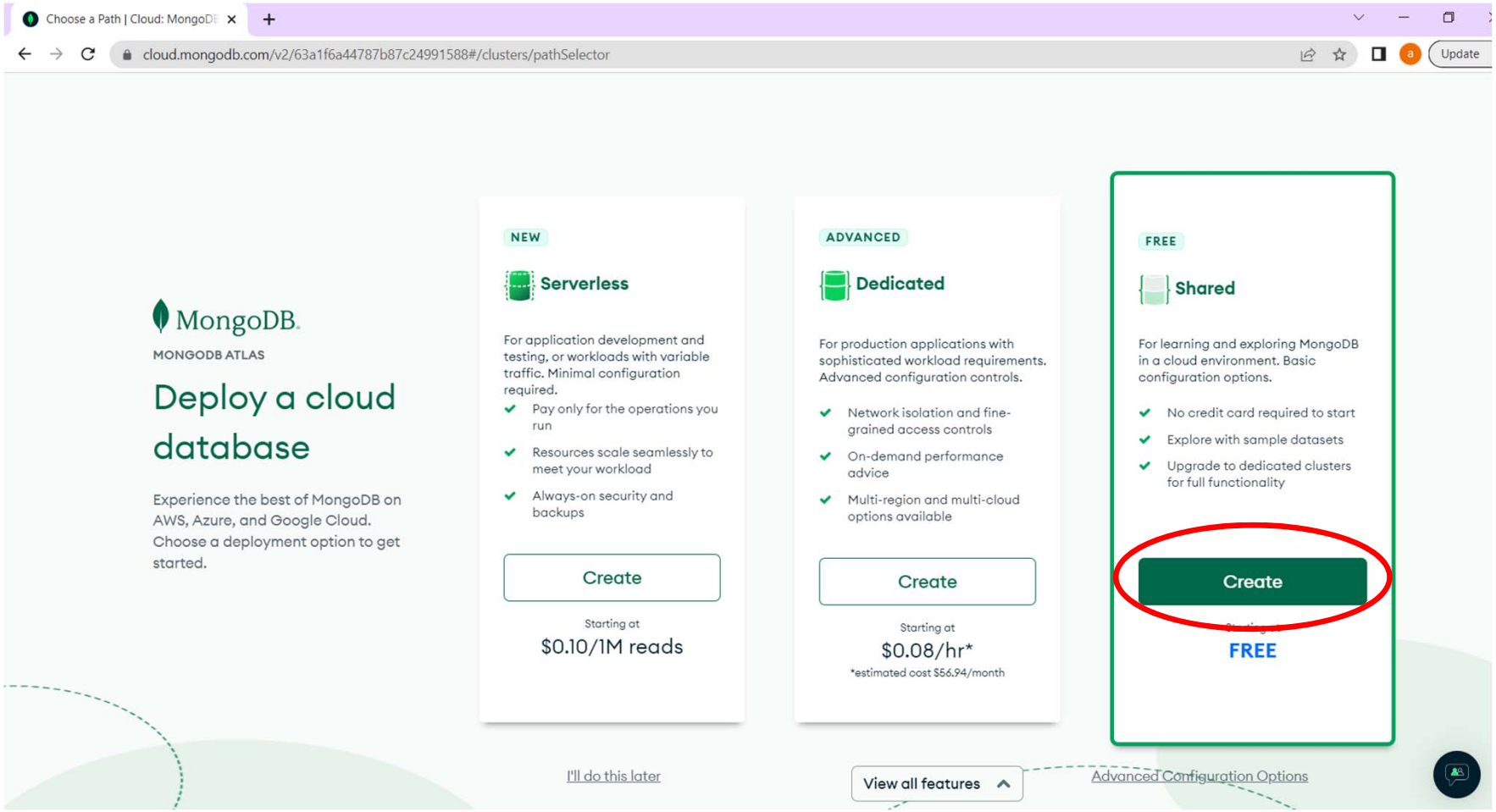
In the verification email, click “Verify Email”, then click “Continue”



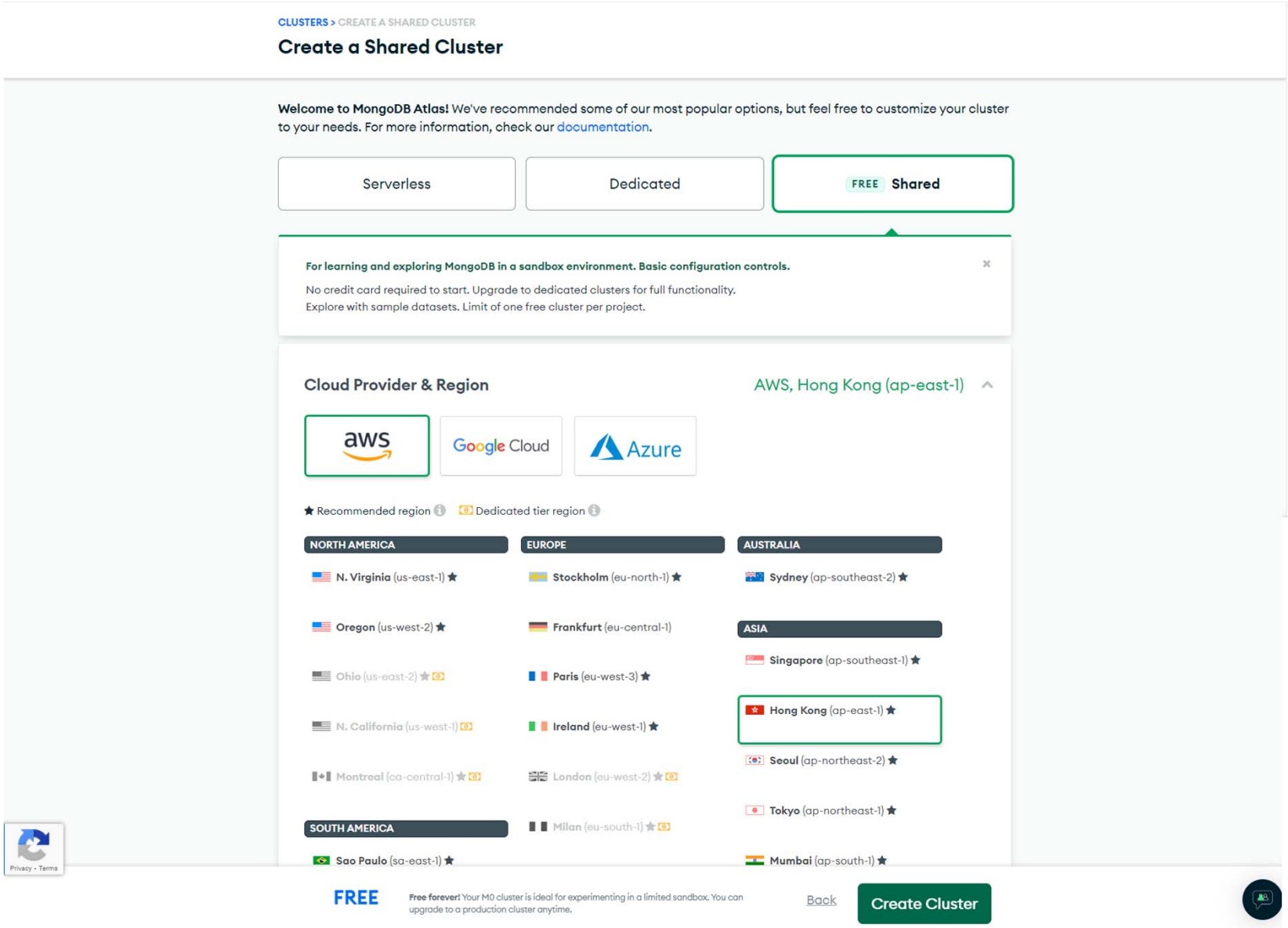
4. In the survey, you may choose “Learn MongoDB”, “No Application” and “JavaScript” Then click “Finish”



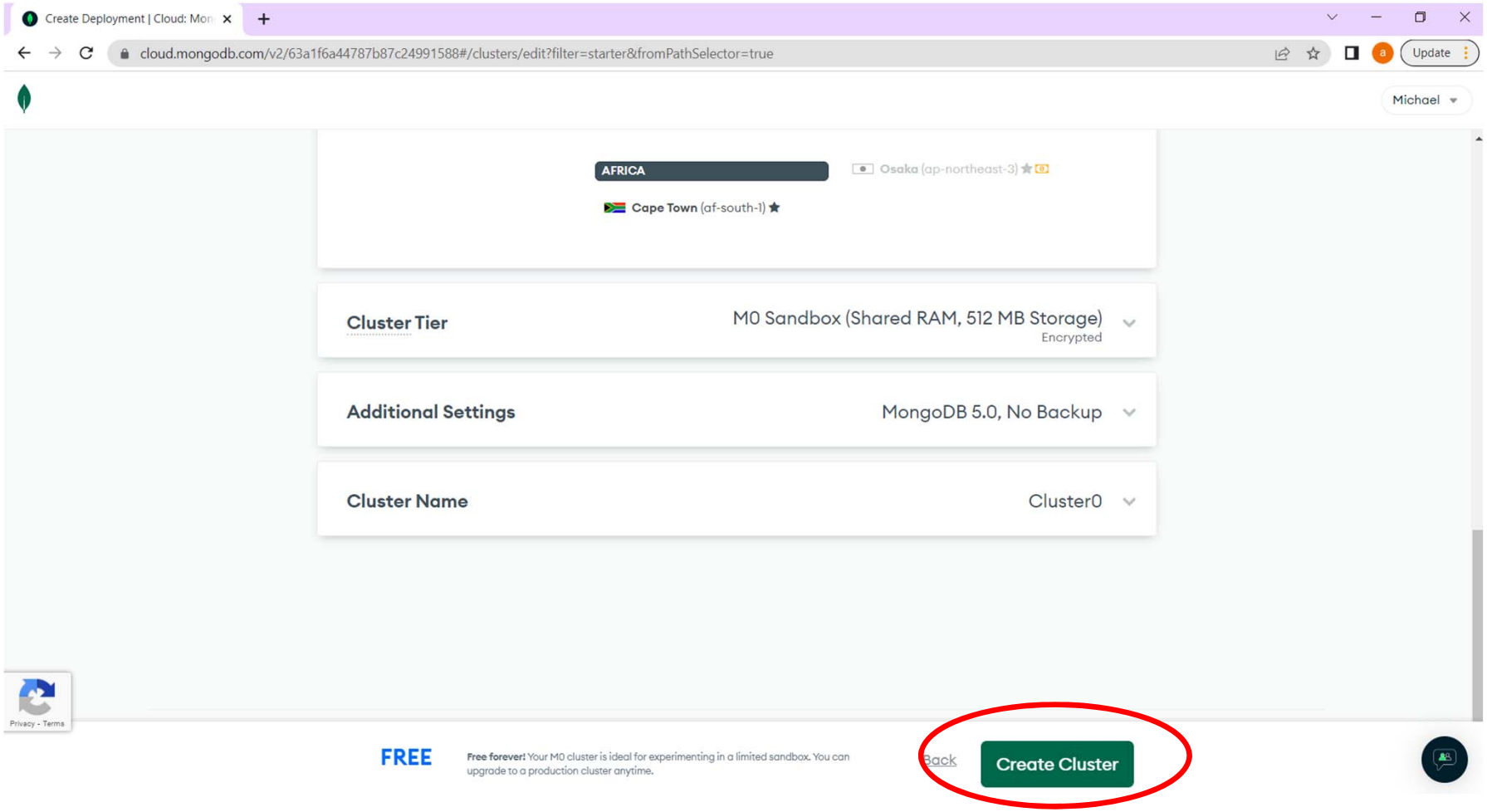
5. Click “Create” in the “Shared” section



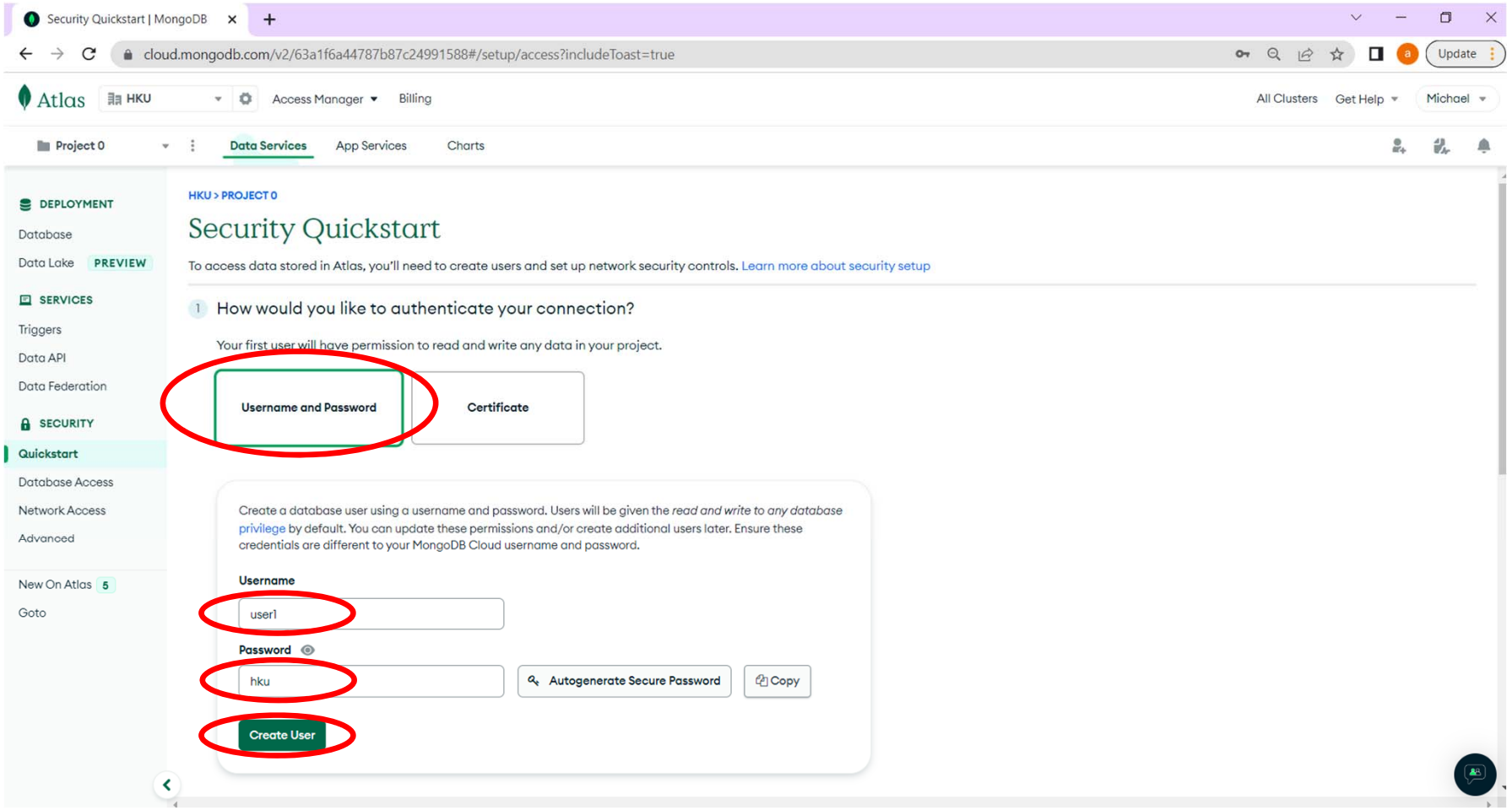
6. Choose a Cloud Provider available in your area



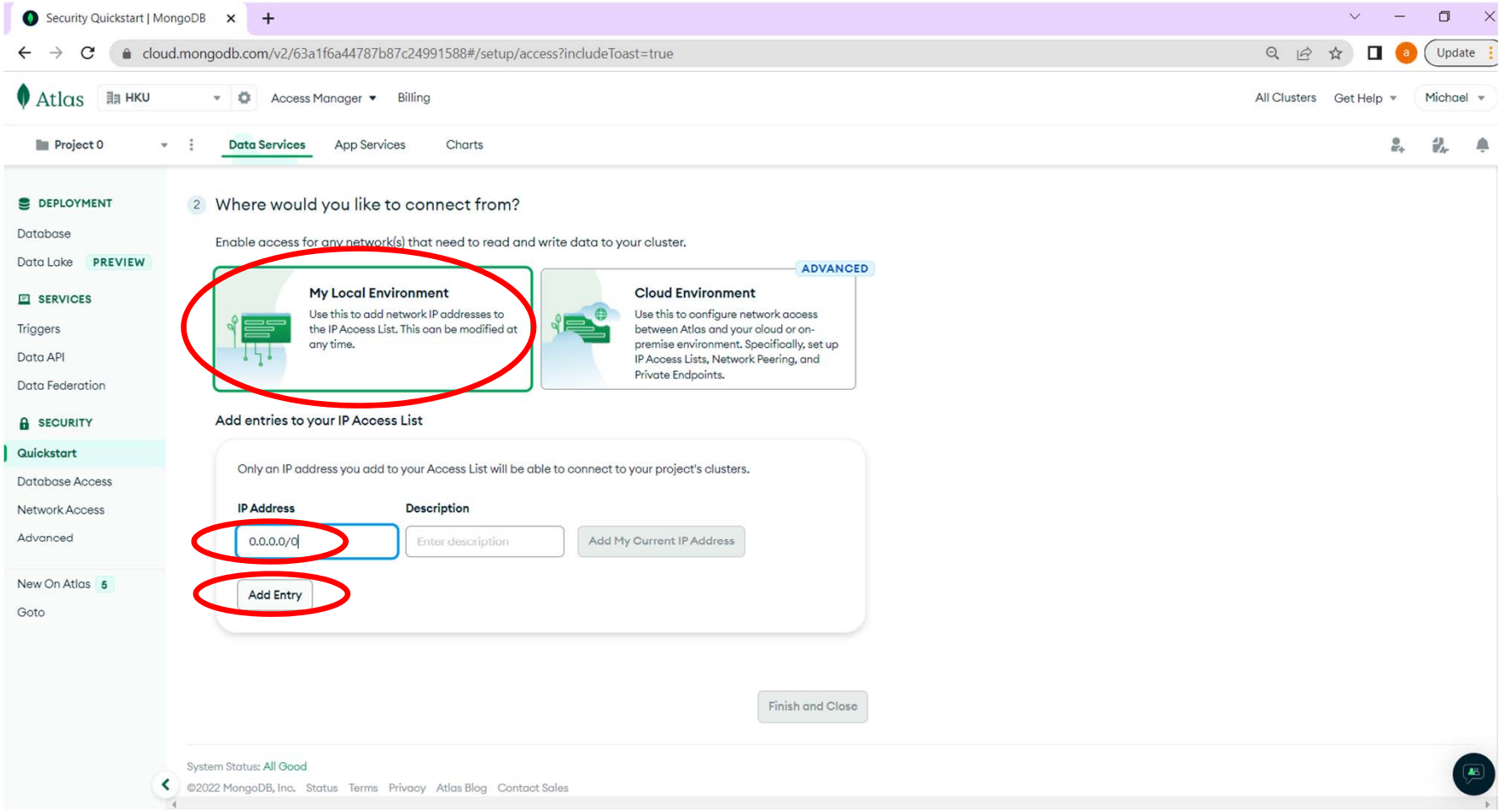
Use the default settings and click “Create Cluster” at the bottom



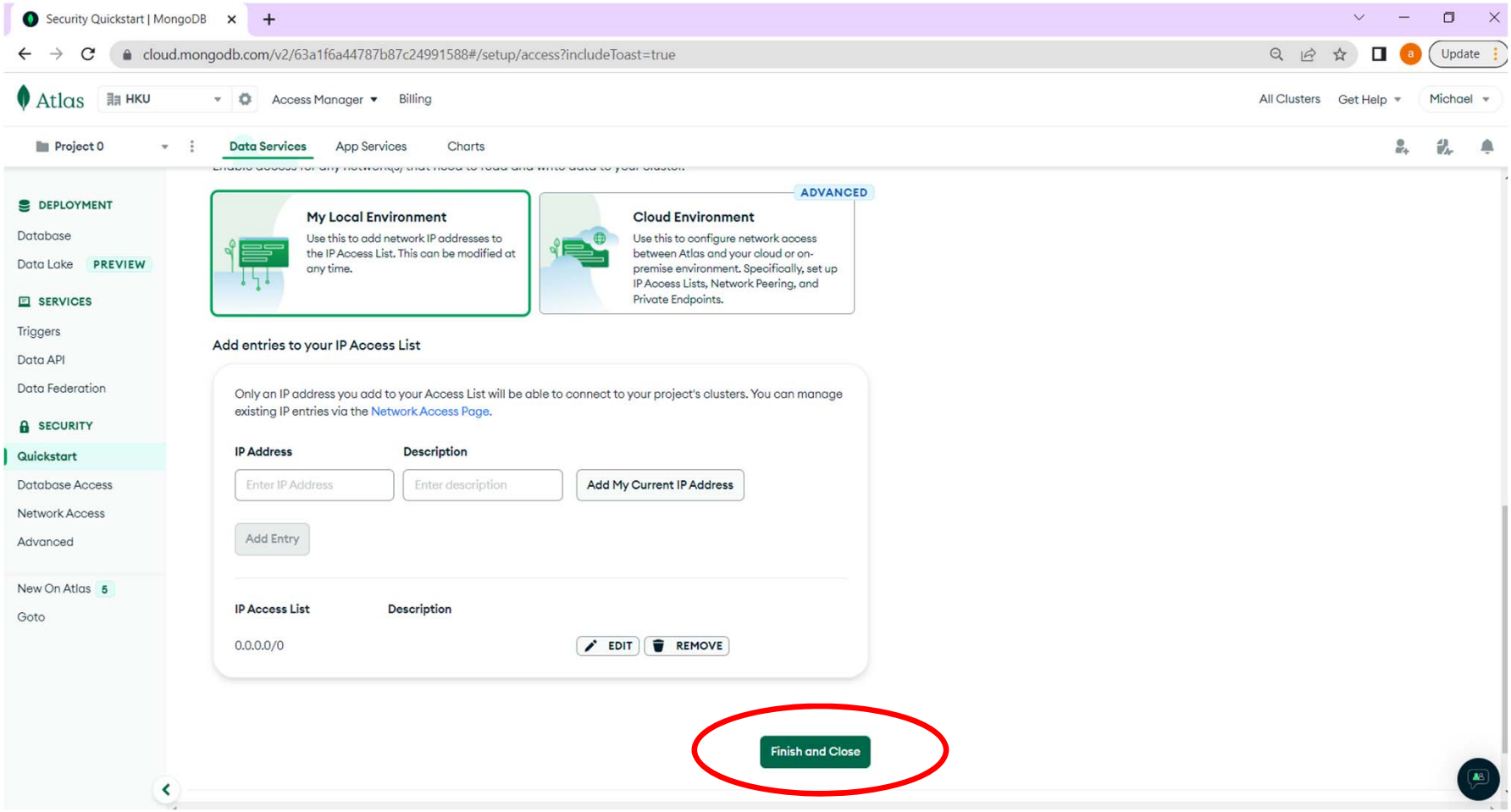
7. Select “Username and Password”
Input “**user1**” in Username
Input “**hku**” in Password
Then click “Create User”



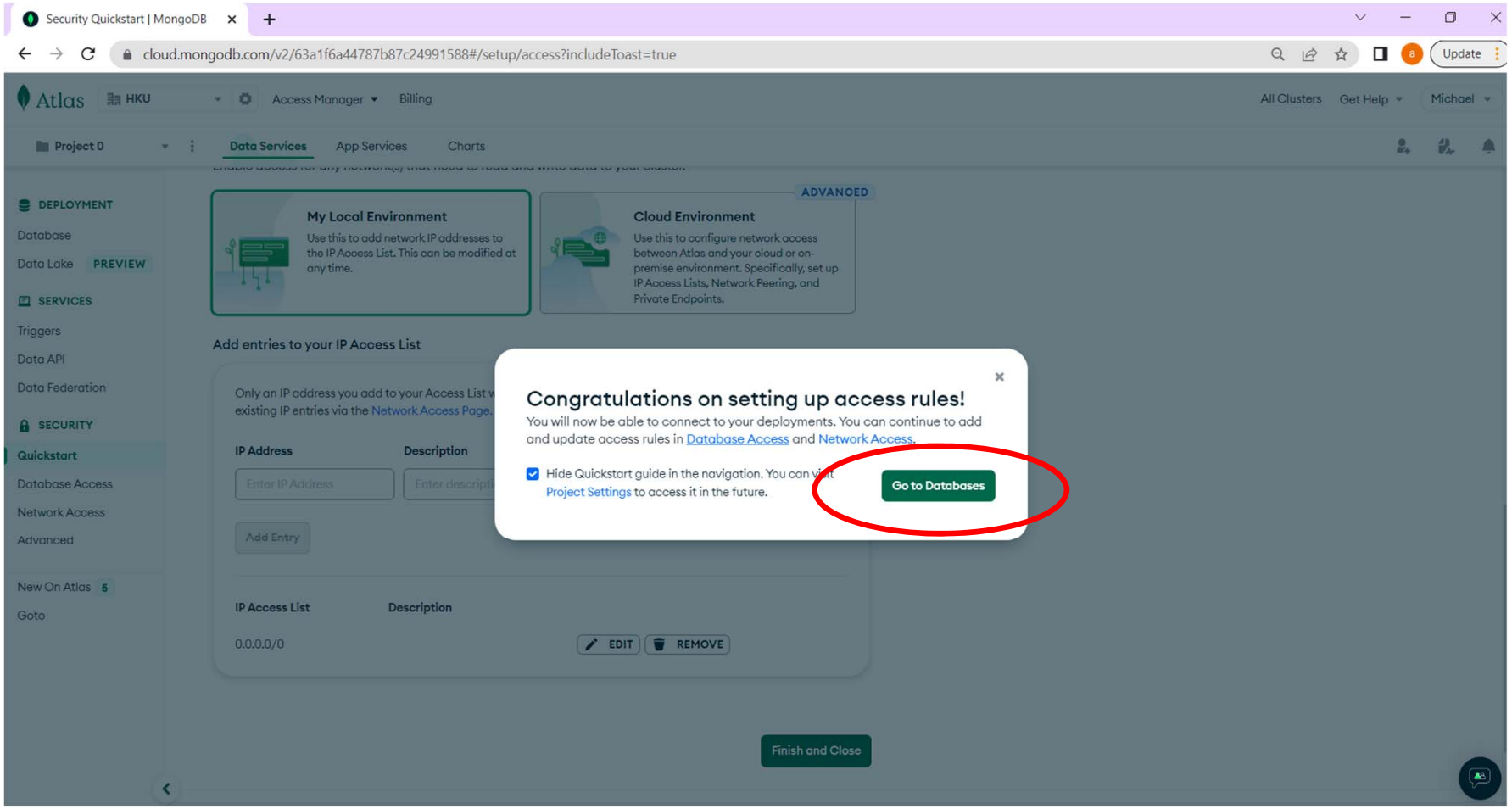
8. Select “My Local Environment” Input “0.0.0.0/0” in IP Address Then click “Add Entry”



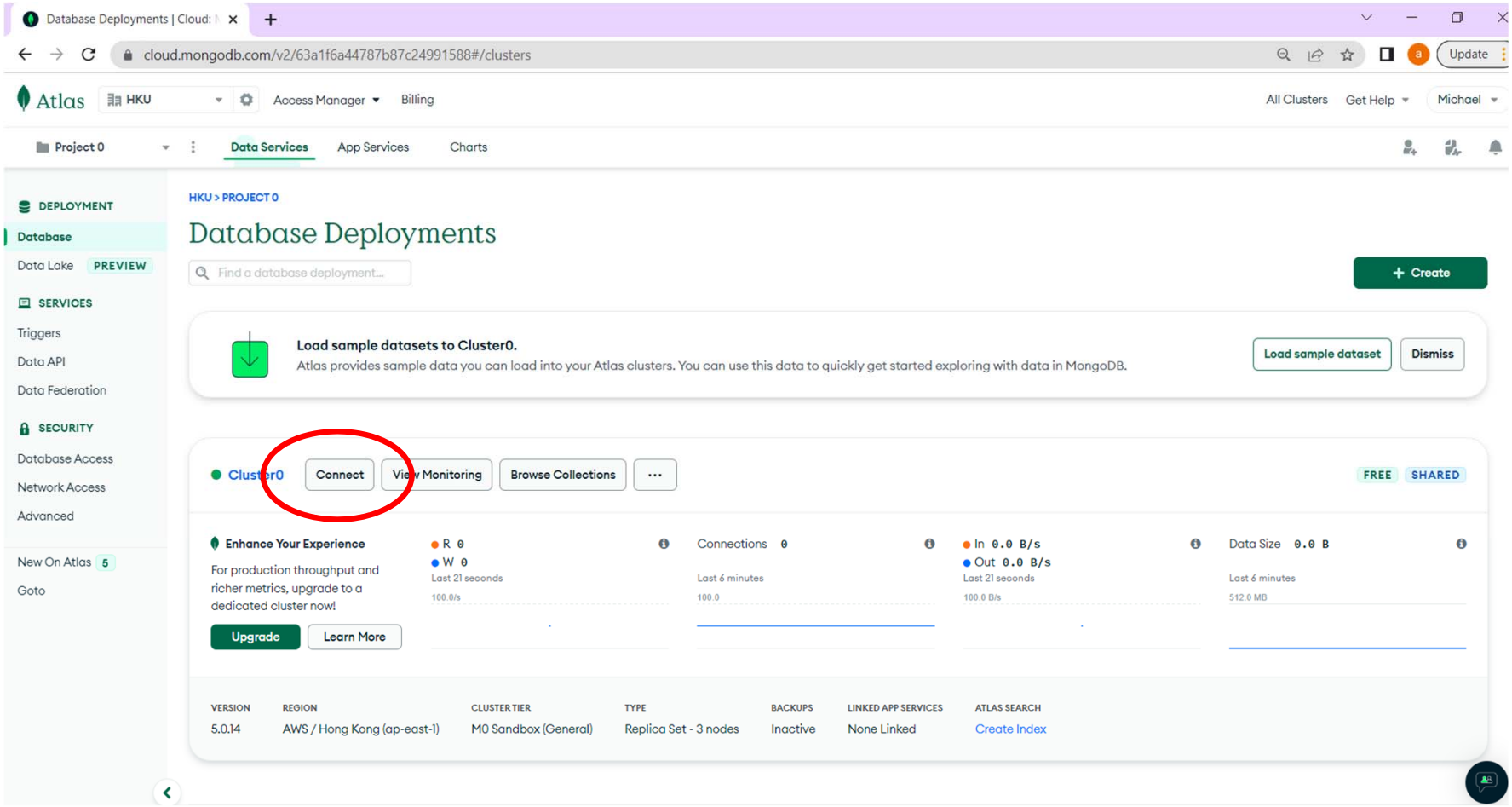
Click “Finish and Close”



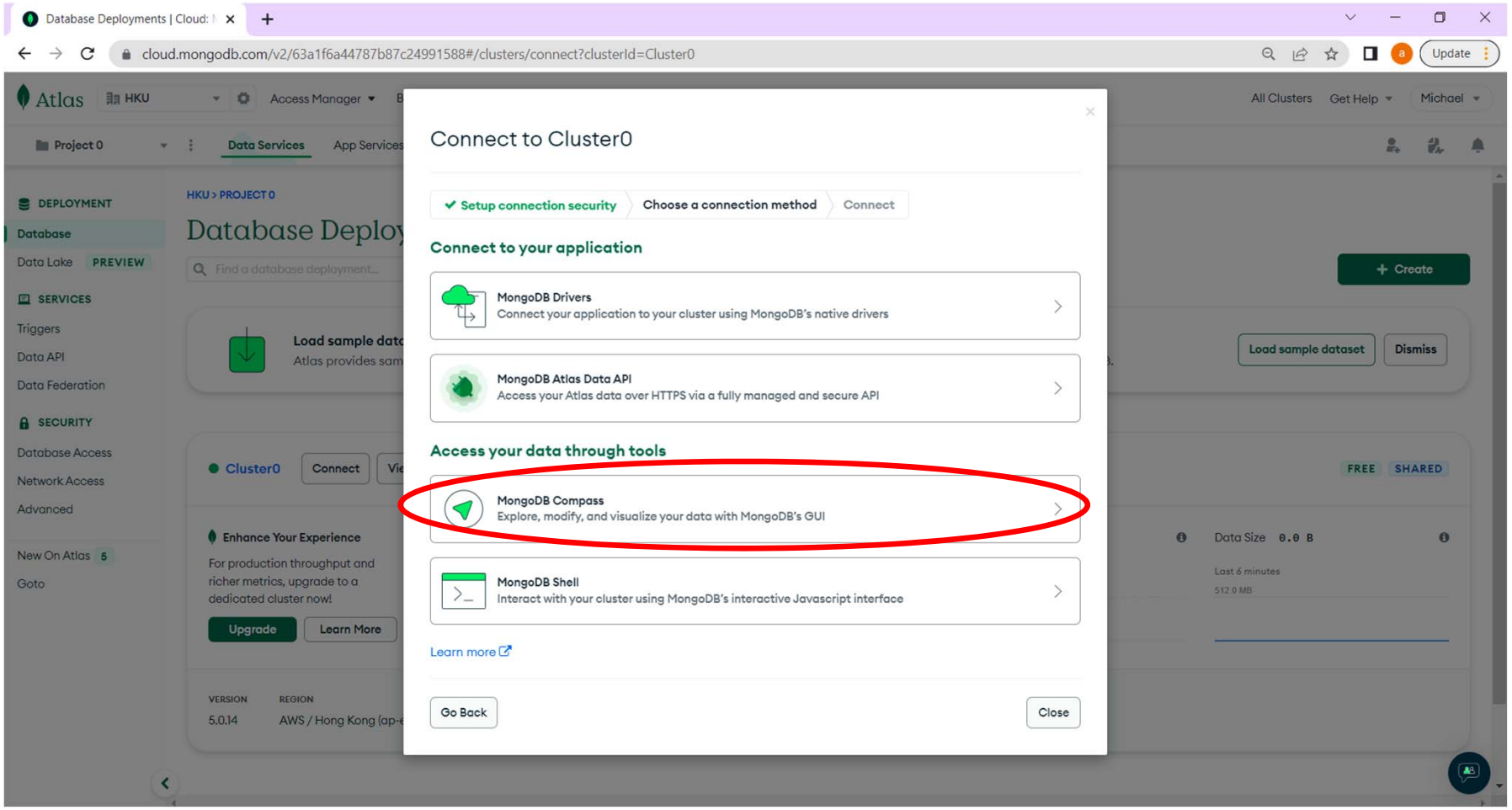
Click “Go to Databases”



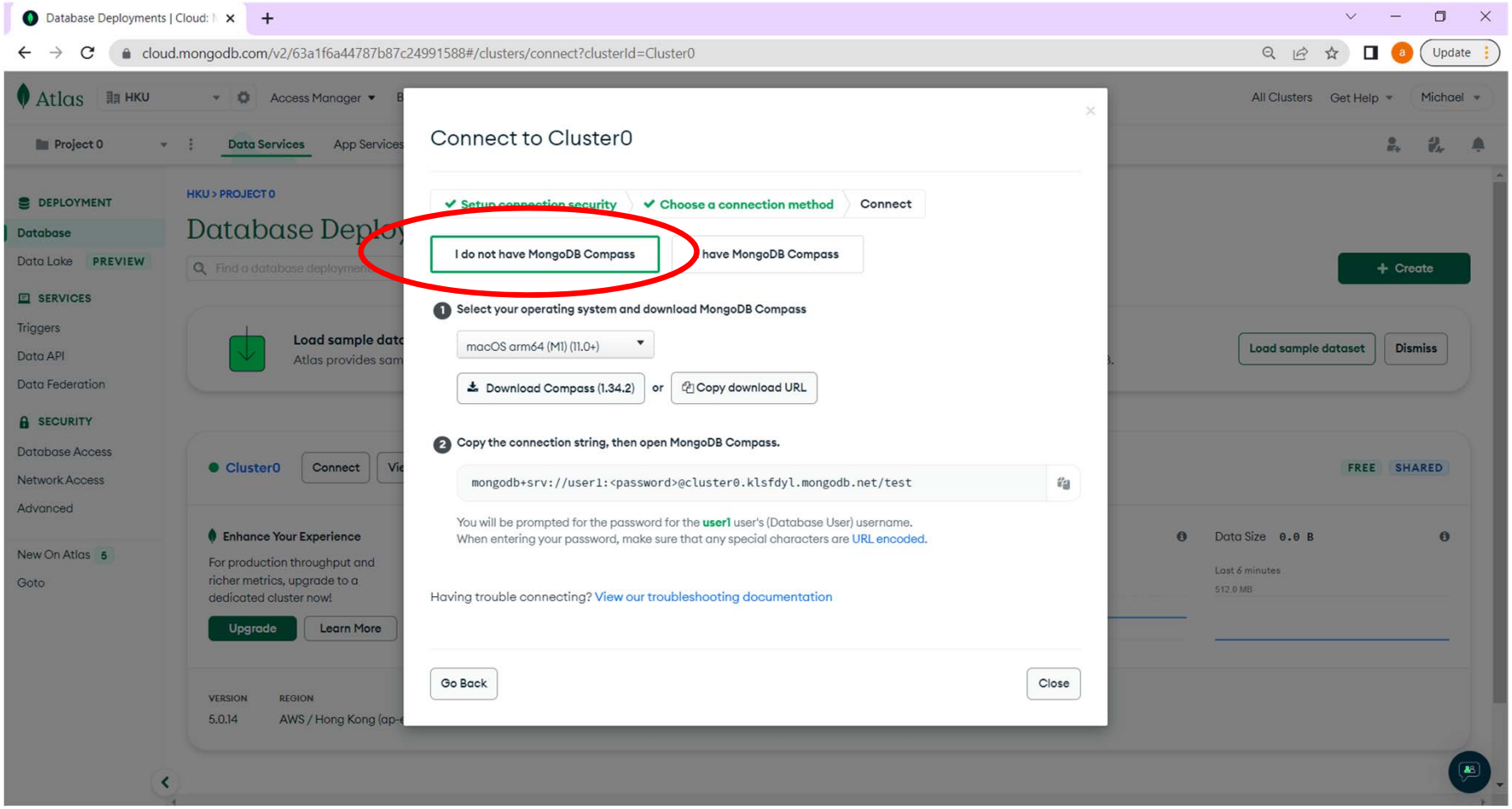
11. Click “Connect”



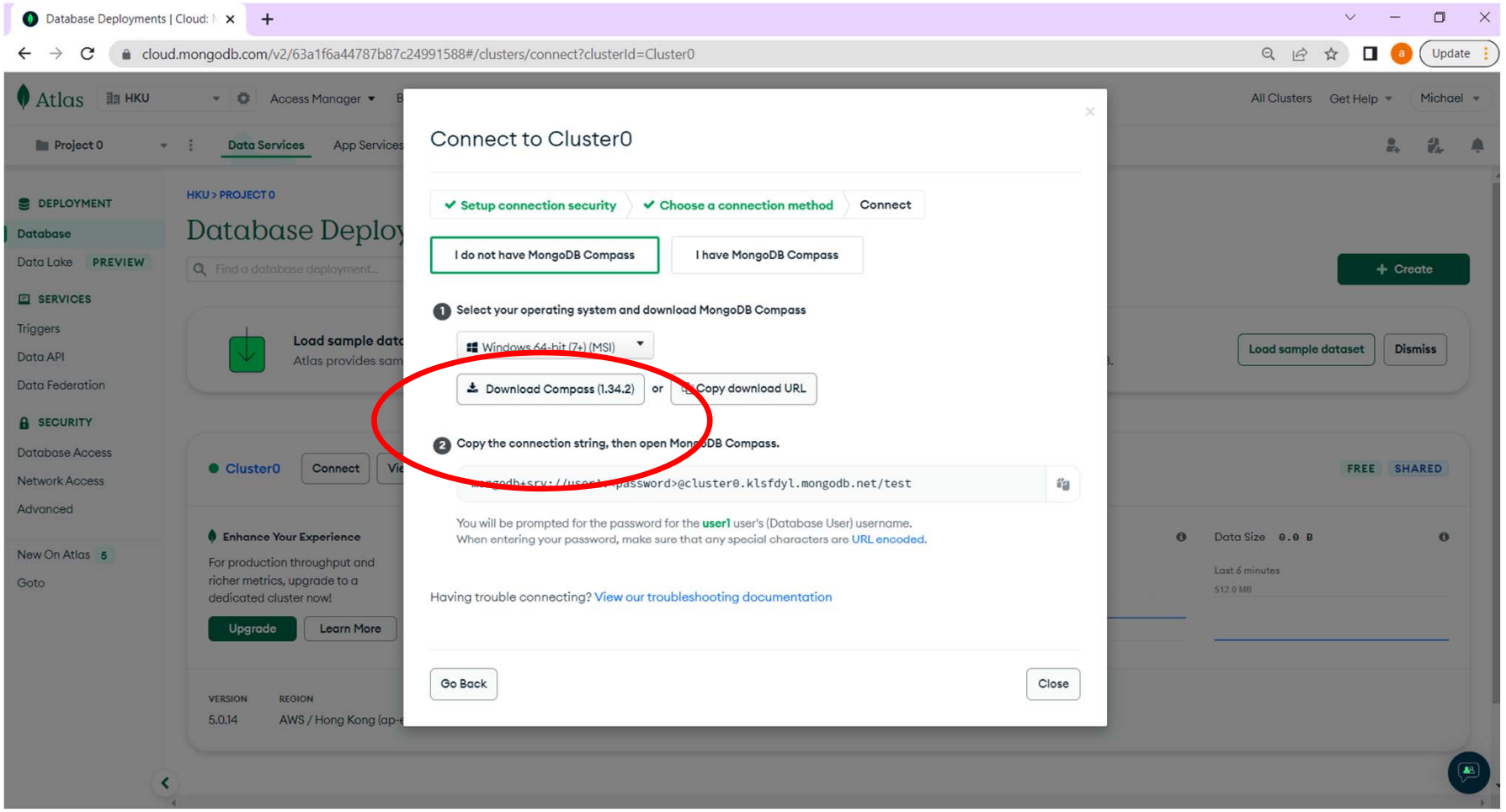
Choose “MongoDB Compass”



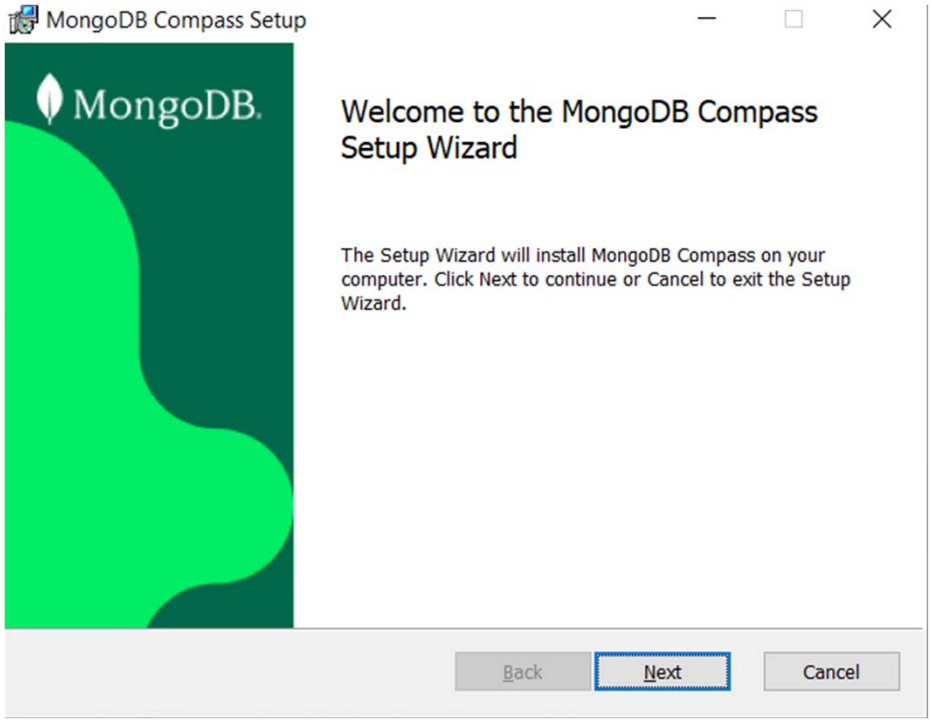
Choose “I do not have MongoDB Compass”



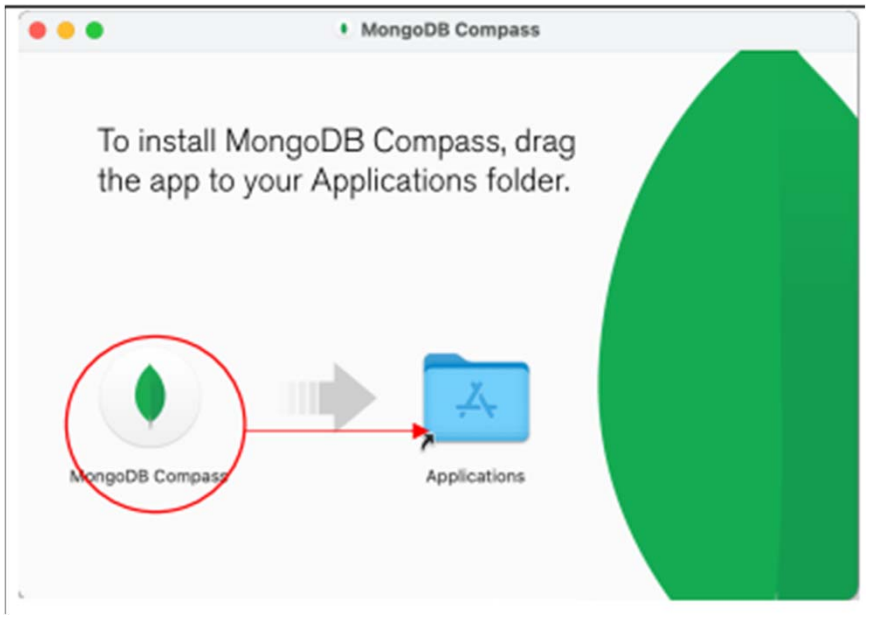
12. Just for the first time, select your operating system and click “Download Compass”



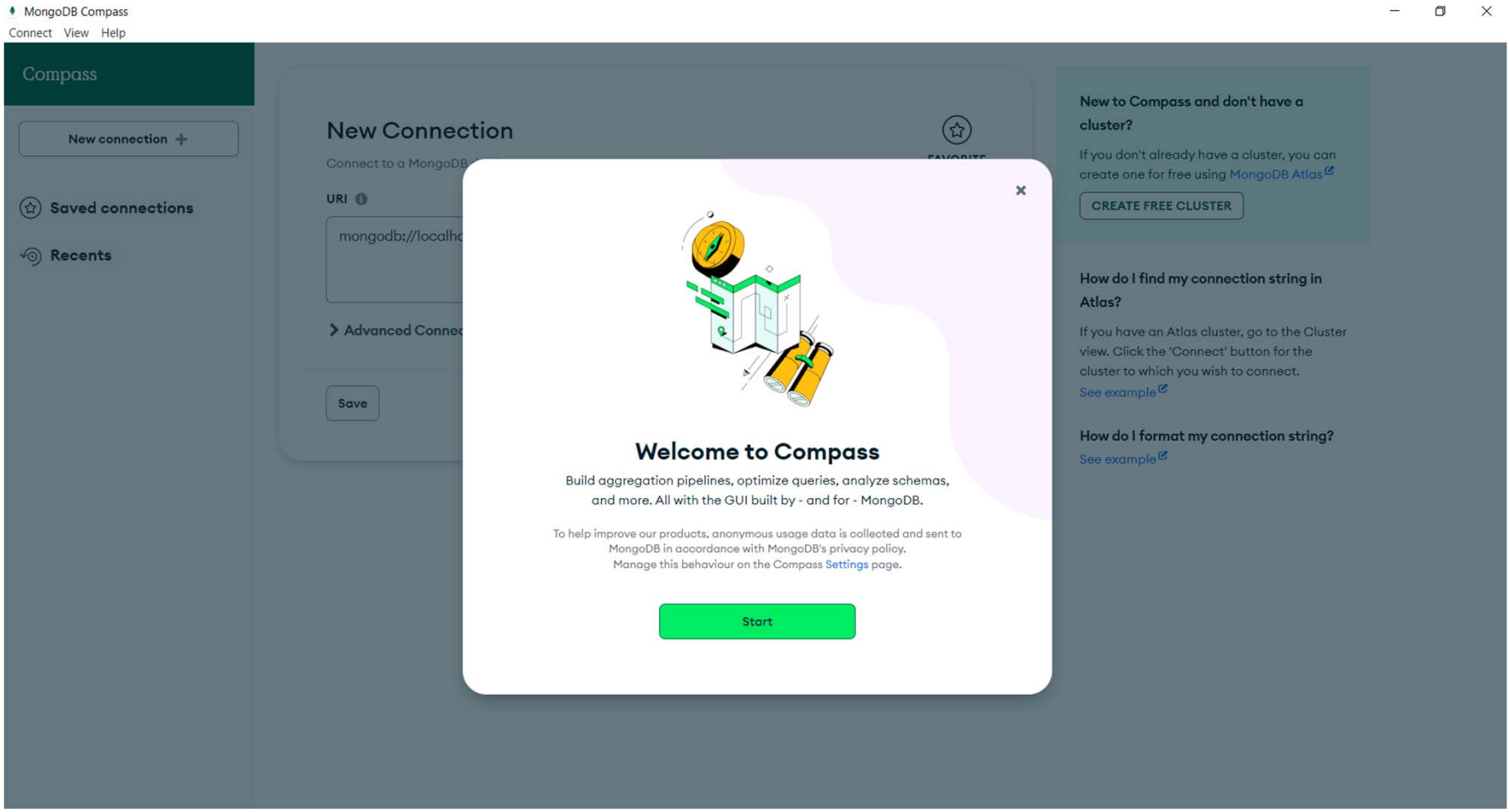
For Windows users, run the downloaded file and choose the default settings during installation



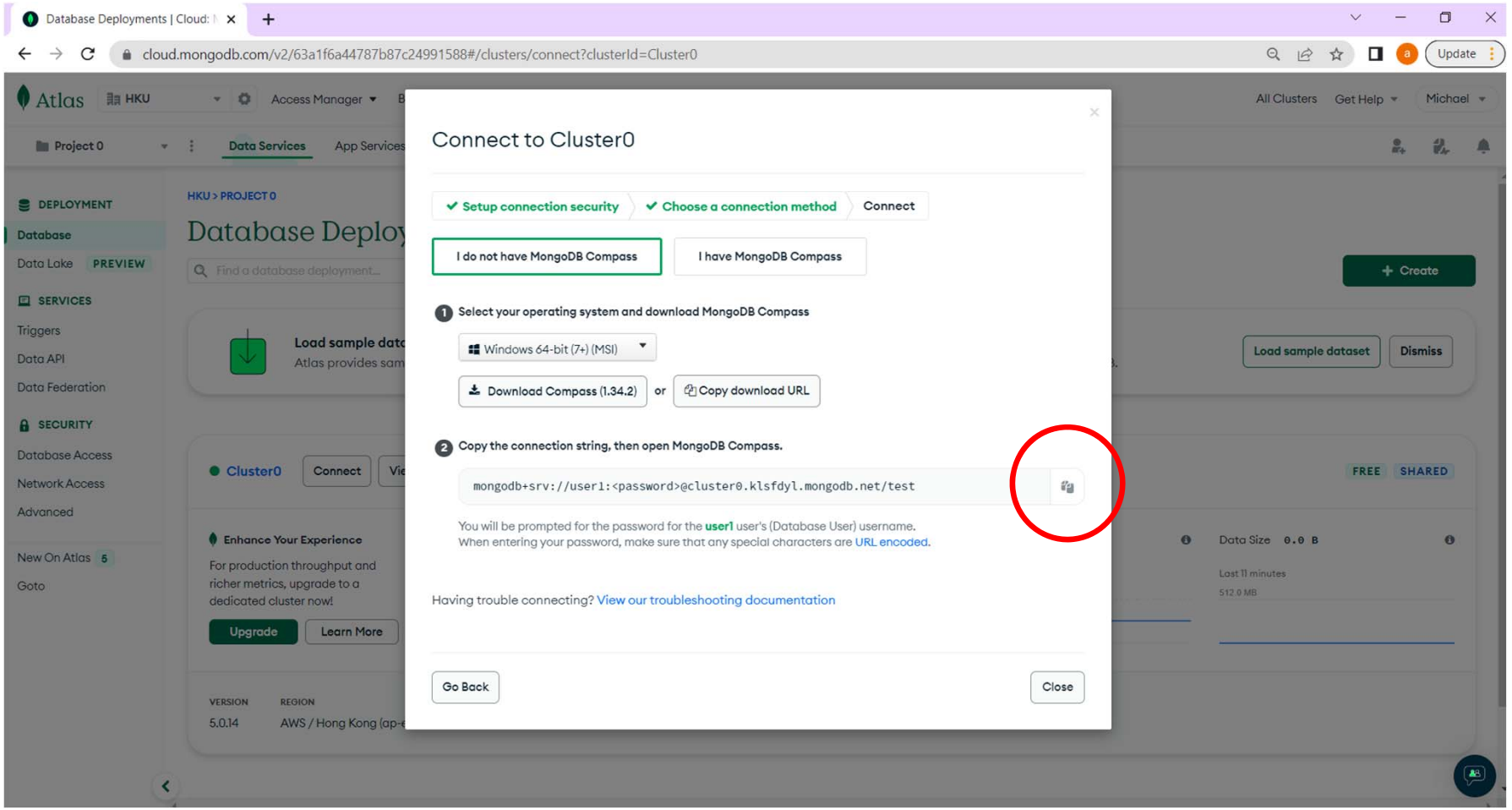
For Mac users, double click the file to extract it
Then drag the icon to the shortcut of Applications
Then start the MongoDB Compass in Applications



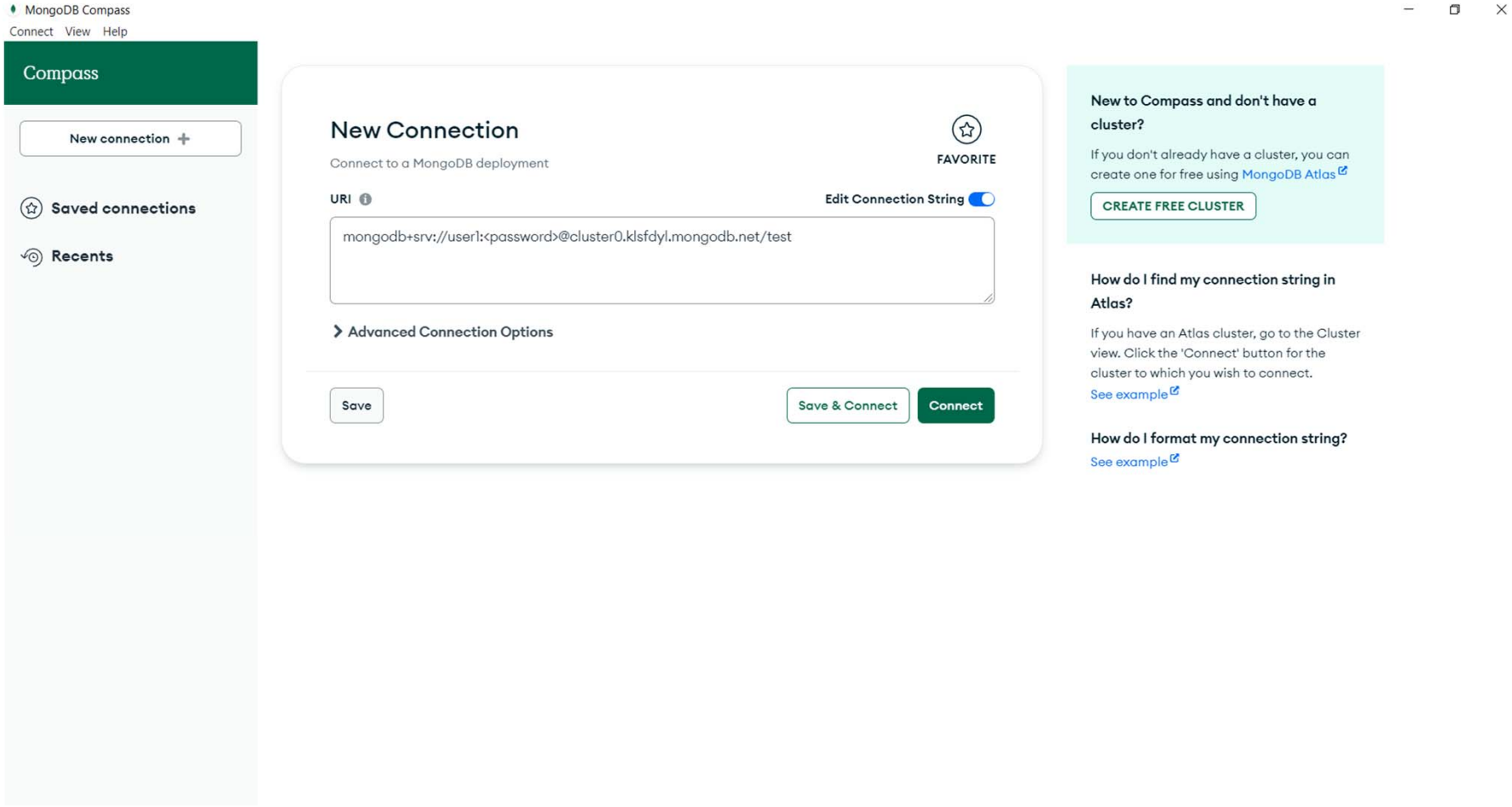
13. Start MongoDB Compass



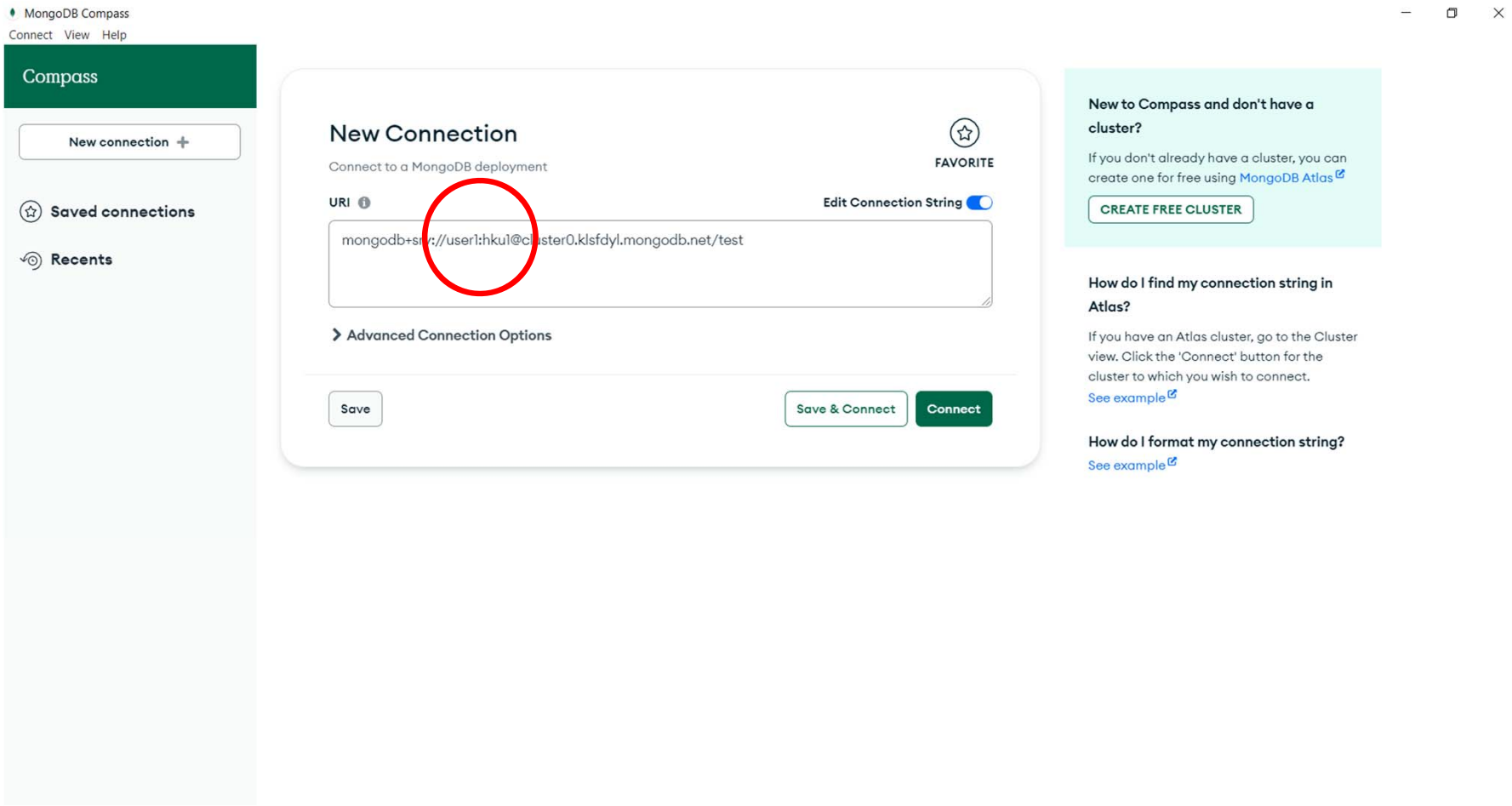
14. Go back to the web browser Copy the connection string by clicking the copy button



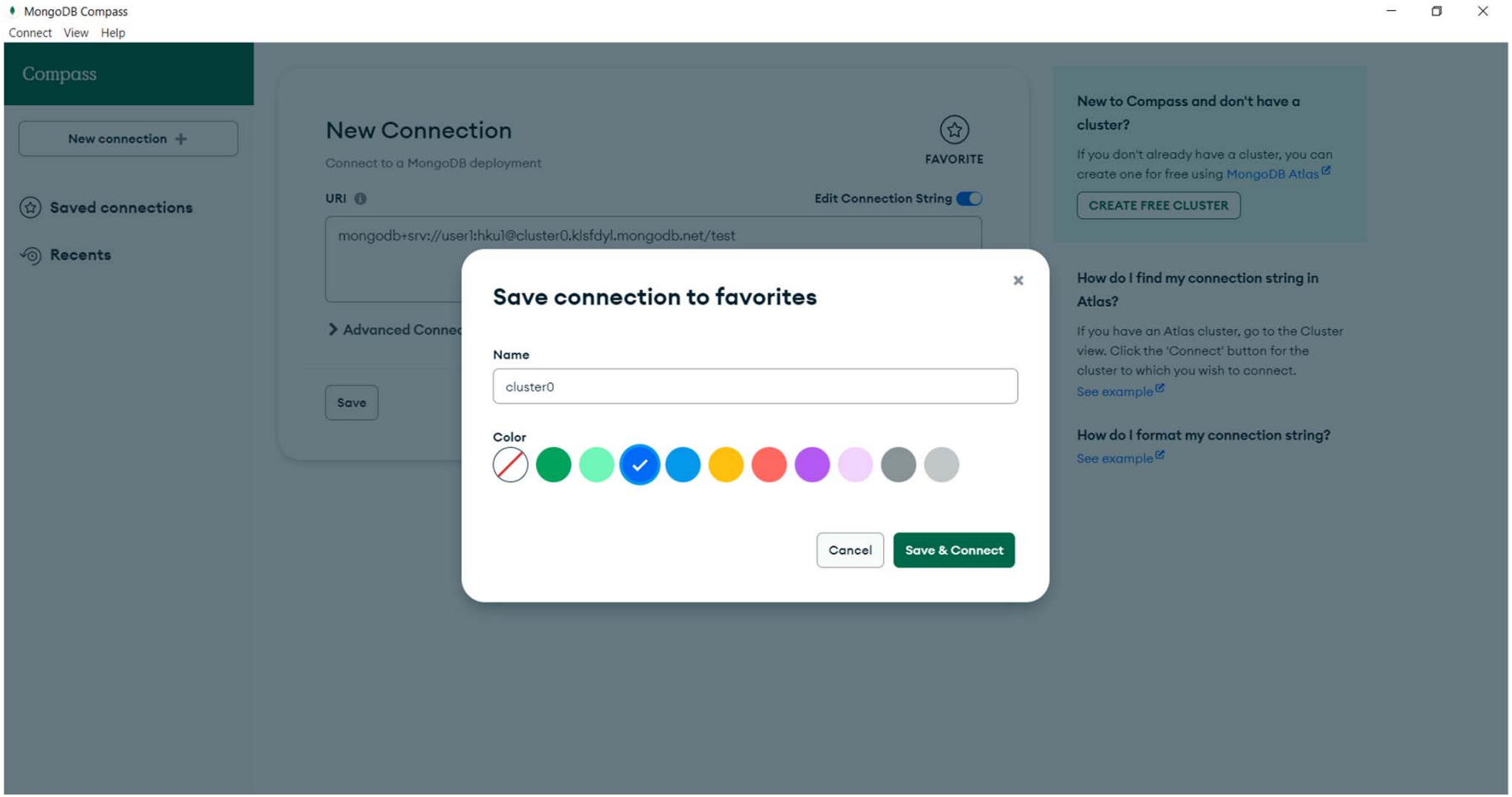
15. Go back to MongoDB Compass Paste the just copied connection string to URI



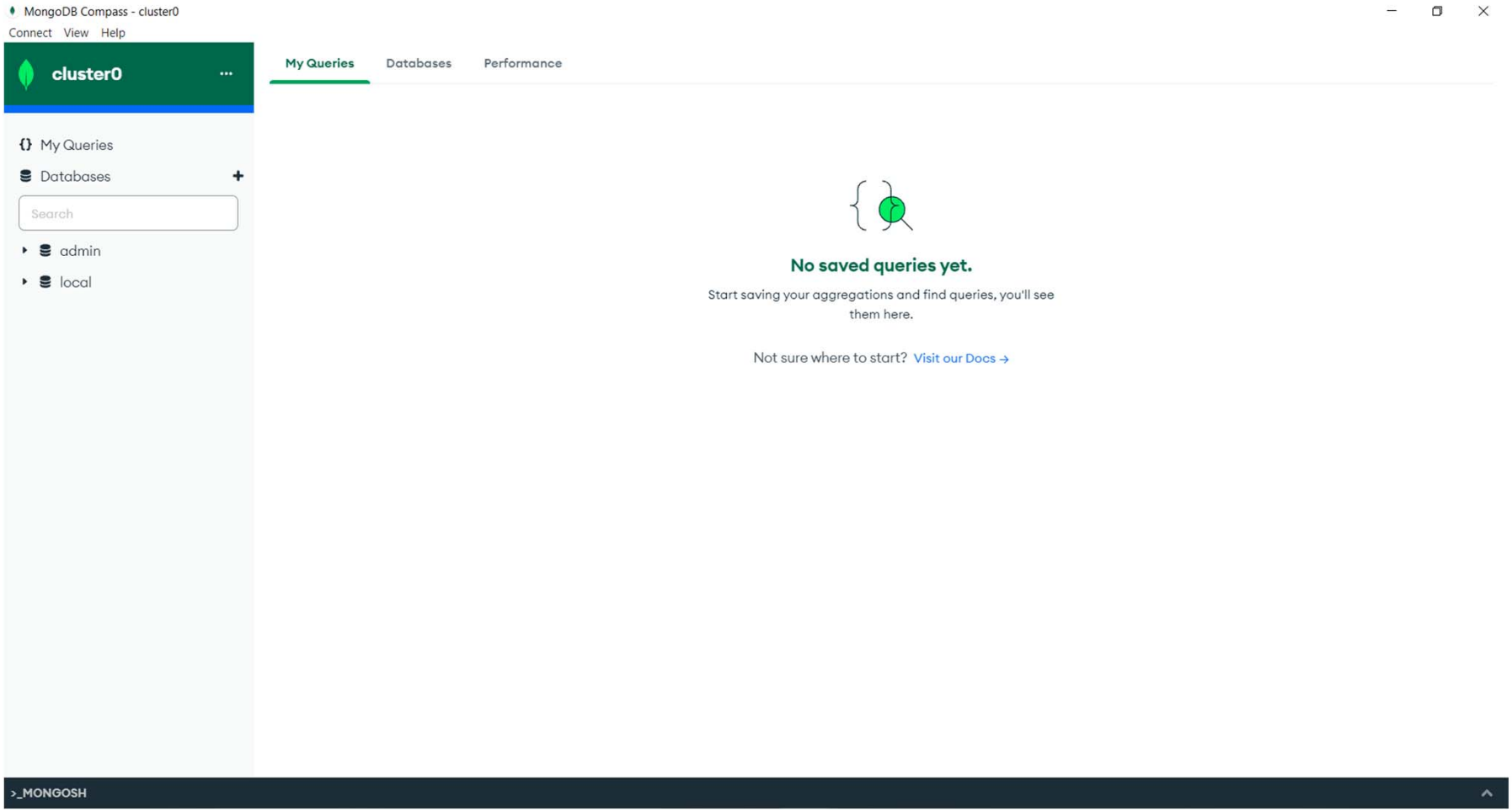
Change “<password>” to “hku”
Then click “Connect”



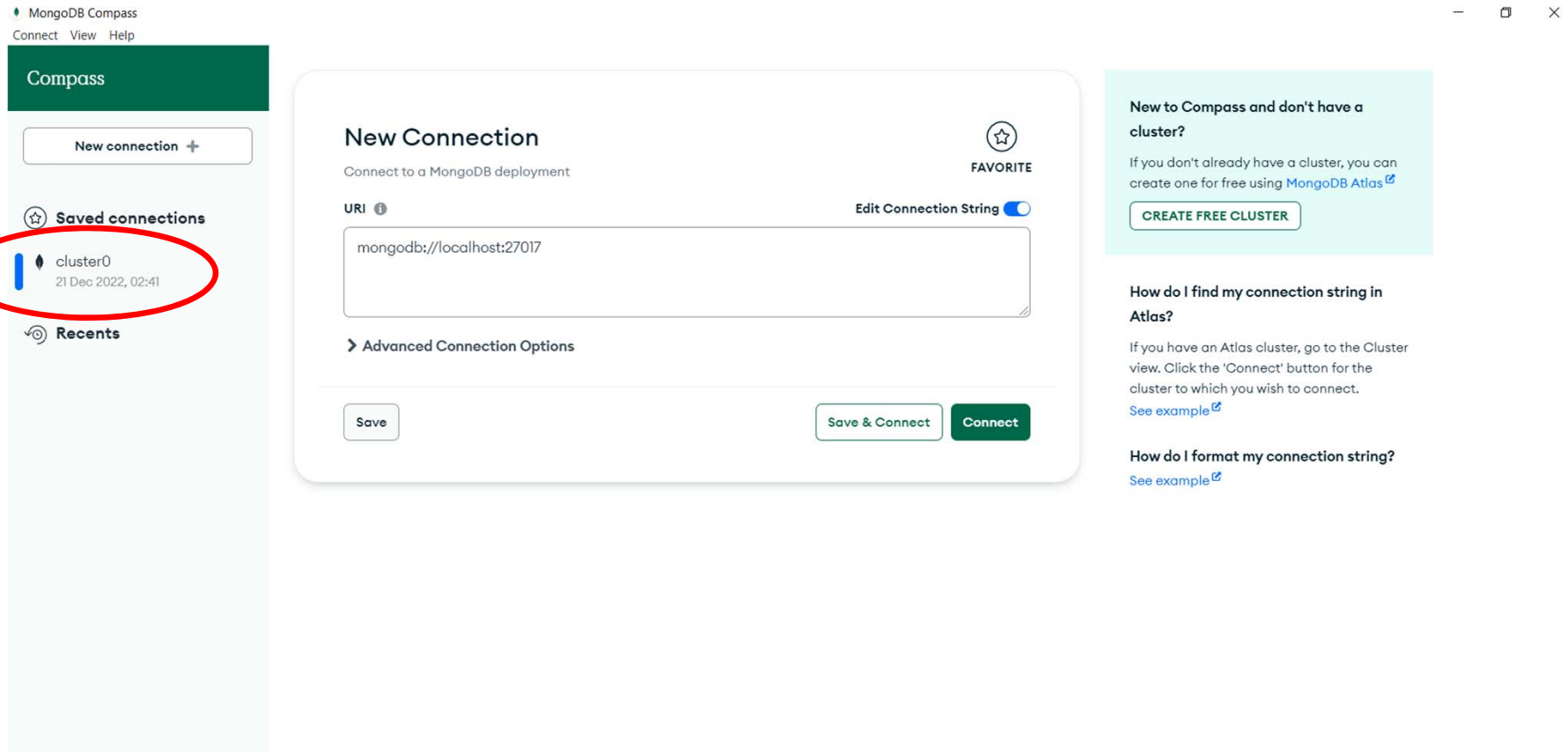
Click “Save & Connect”



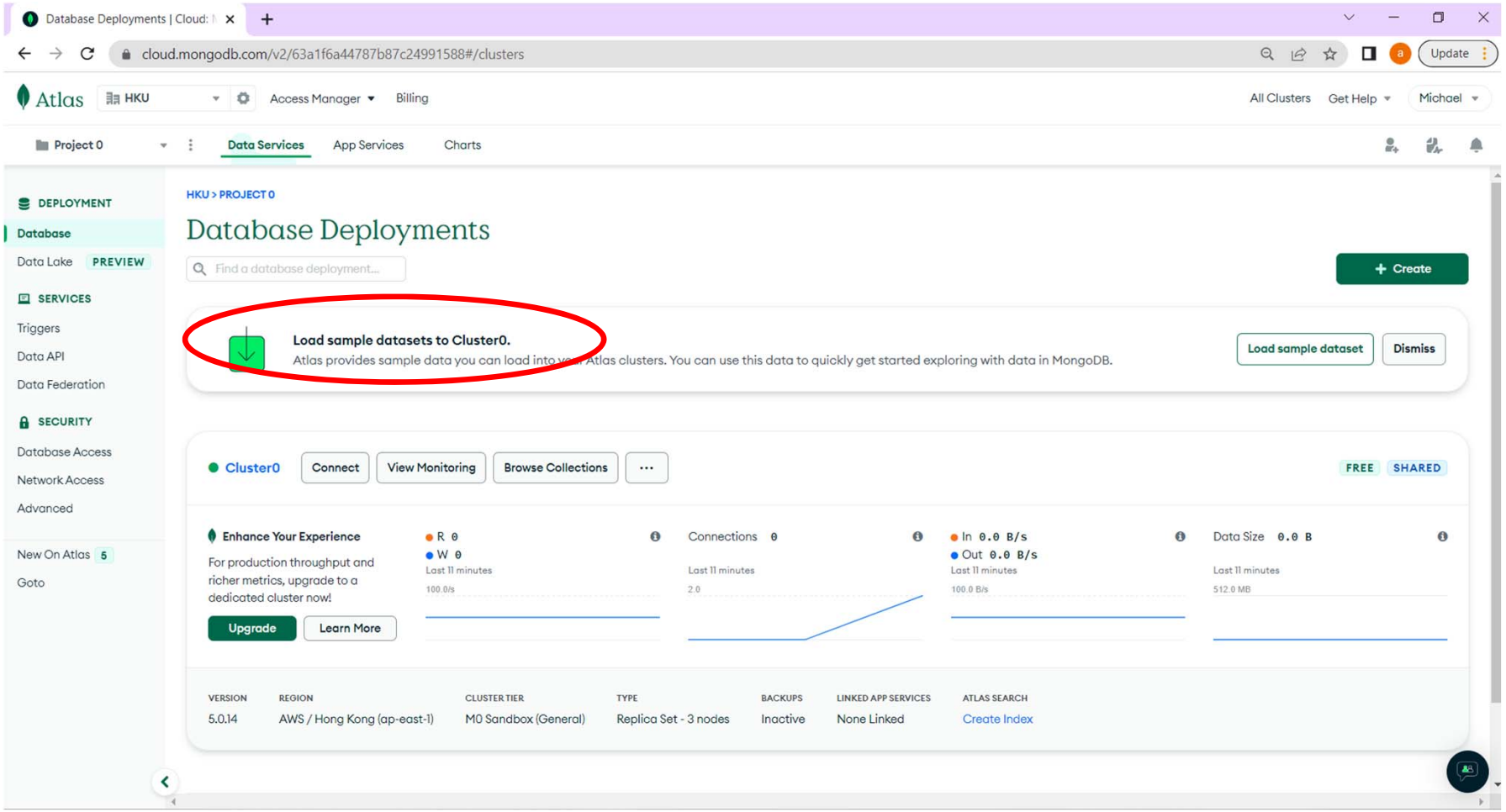
Successfully connected!



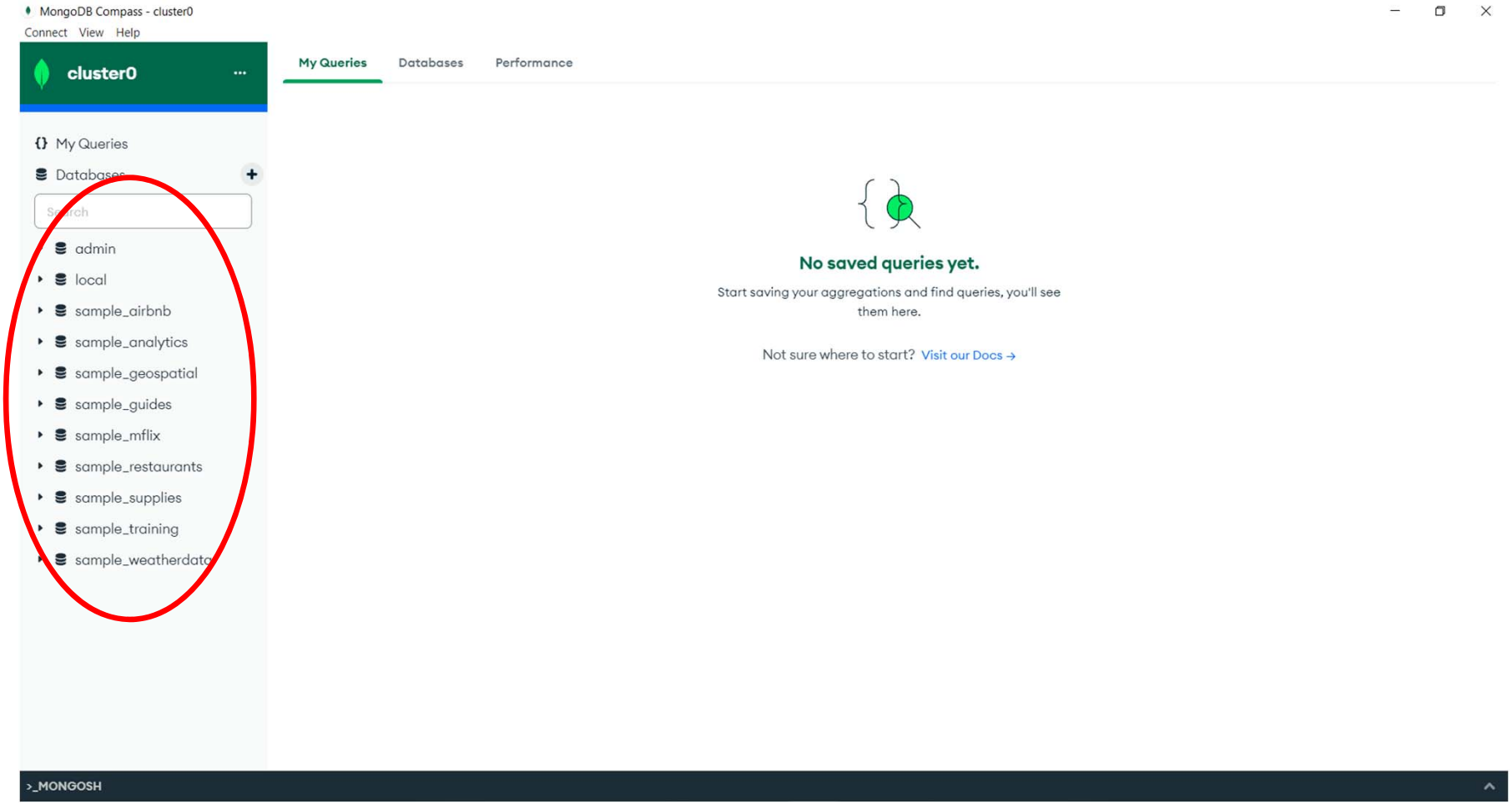
You can reuse the connection to “cluster0” when you open MongoDB Compass next time



You may also load some sample data to your database (this button may not be available)

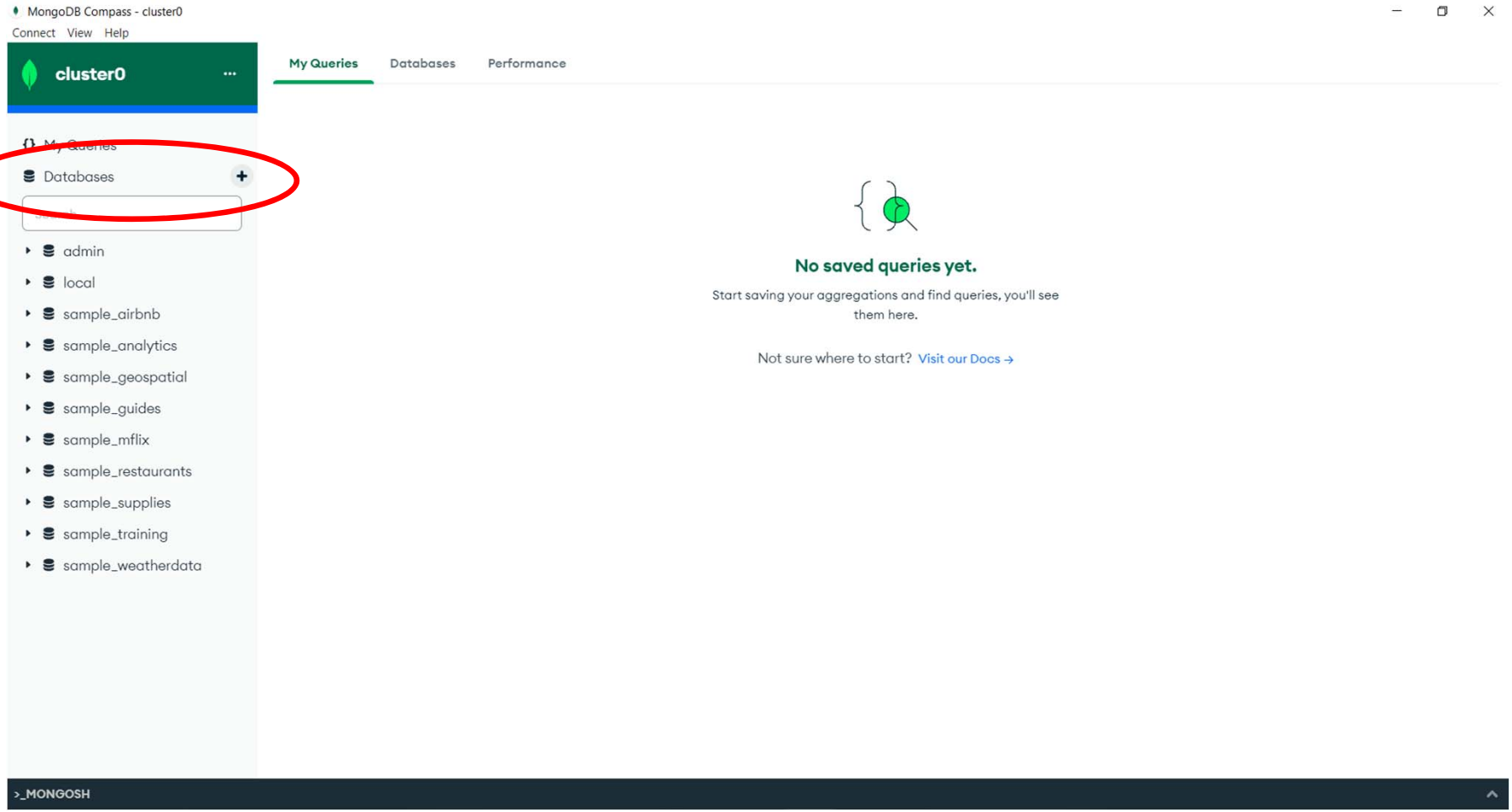


List of sample data will then be visible in MongoDB (may take some time to load)

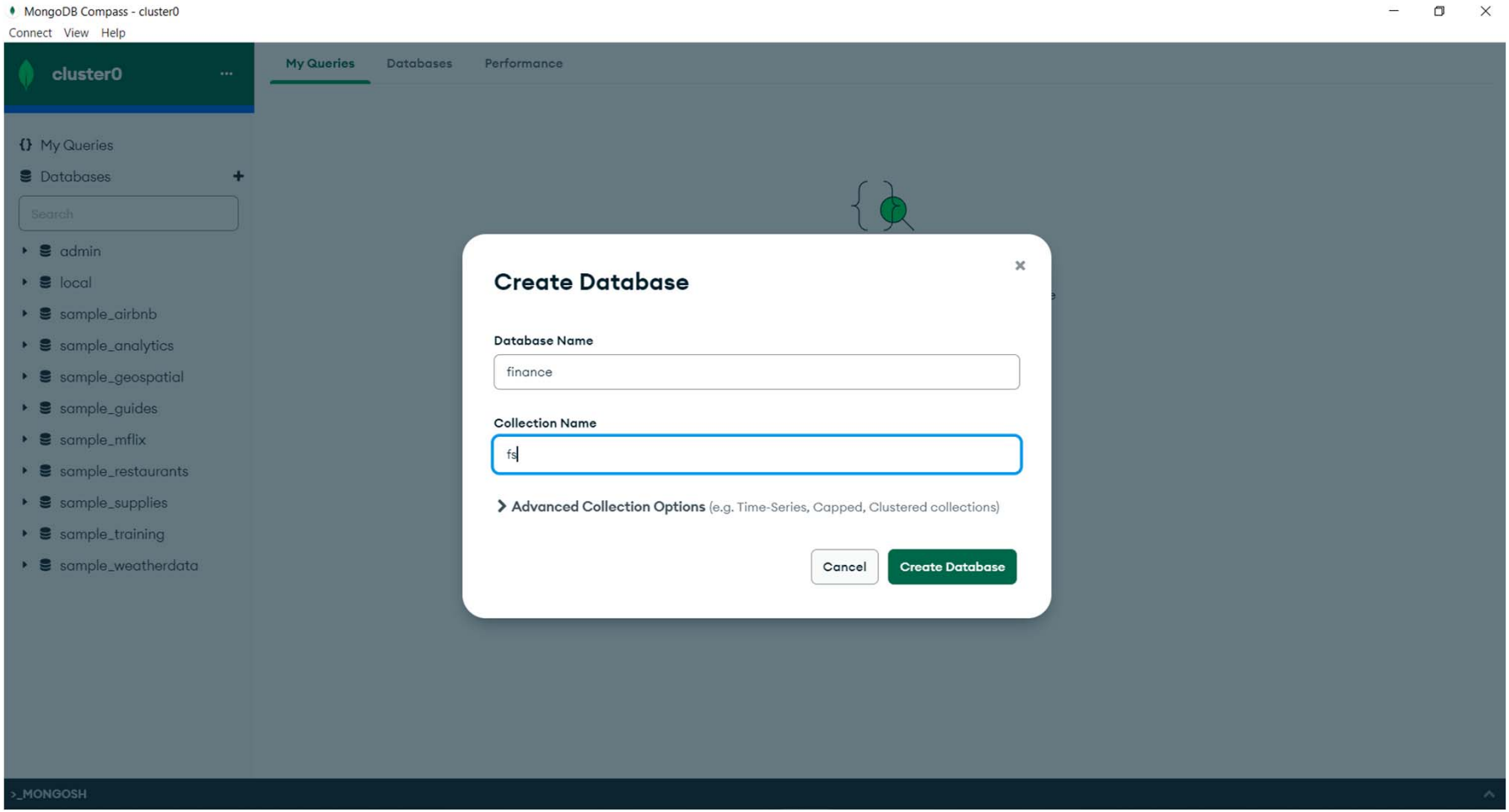


Creating Databases and Tables Through MongoDB Compass

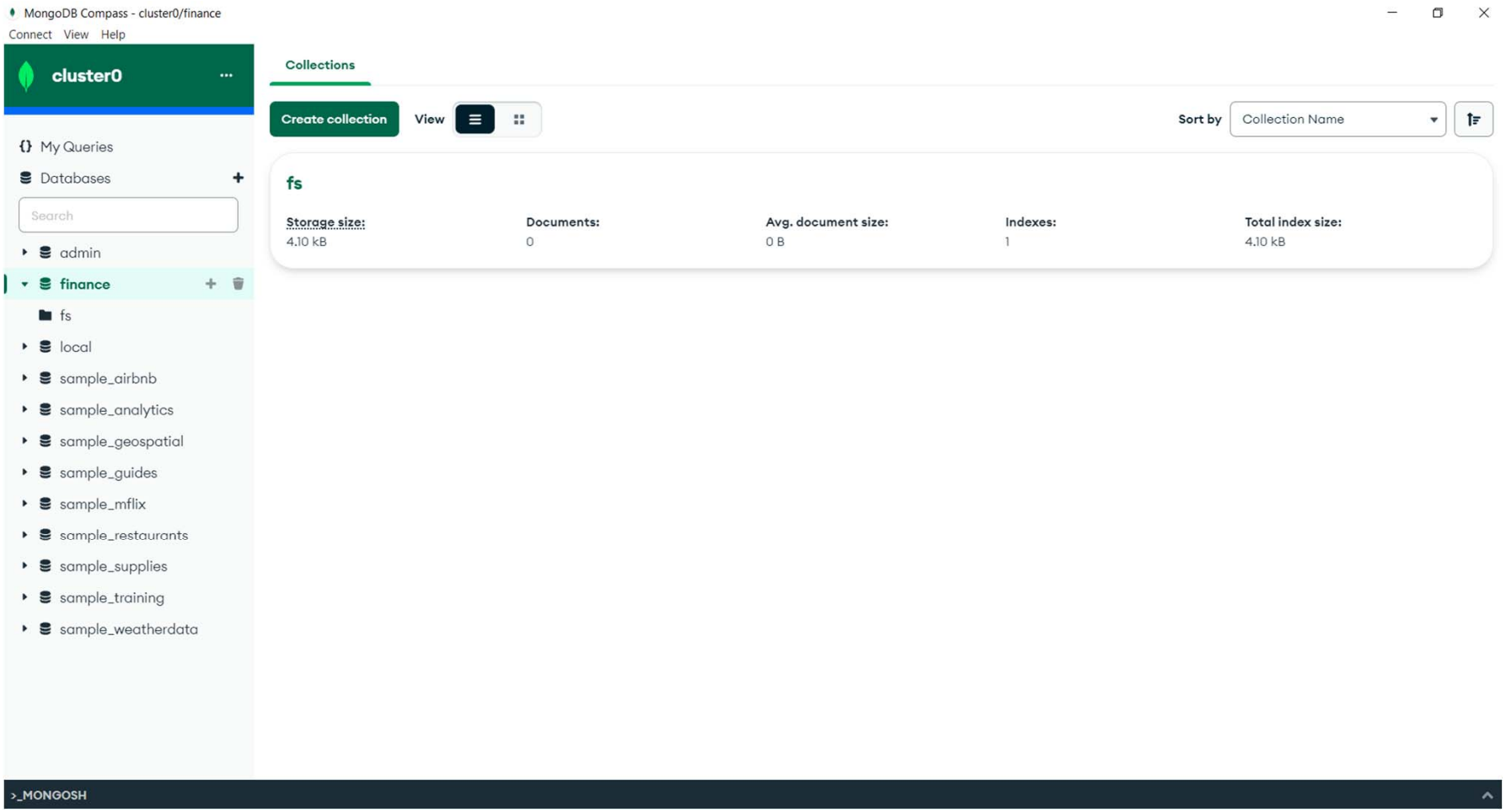
Click the “+” sign next to “Databases” on the left panel



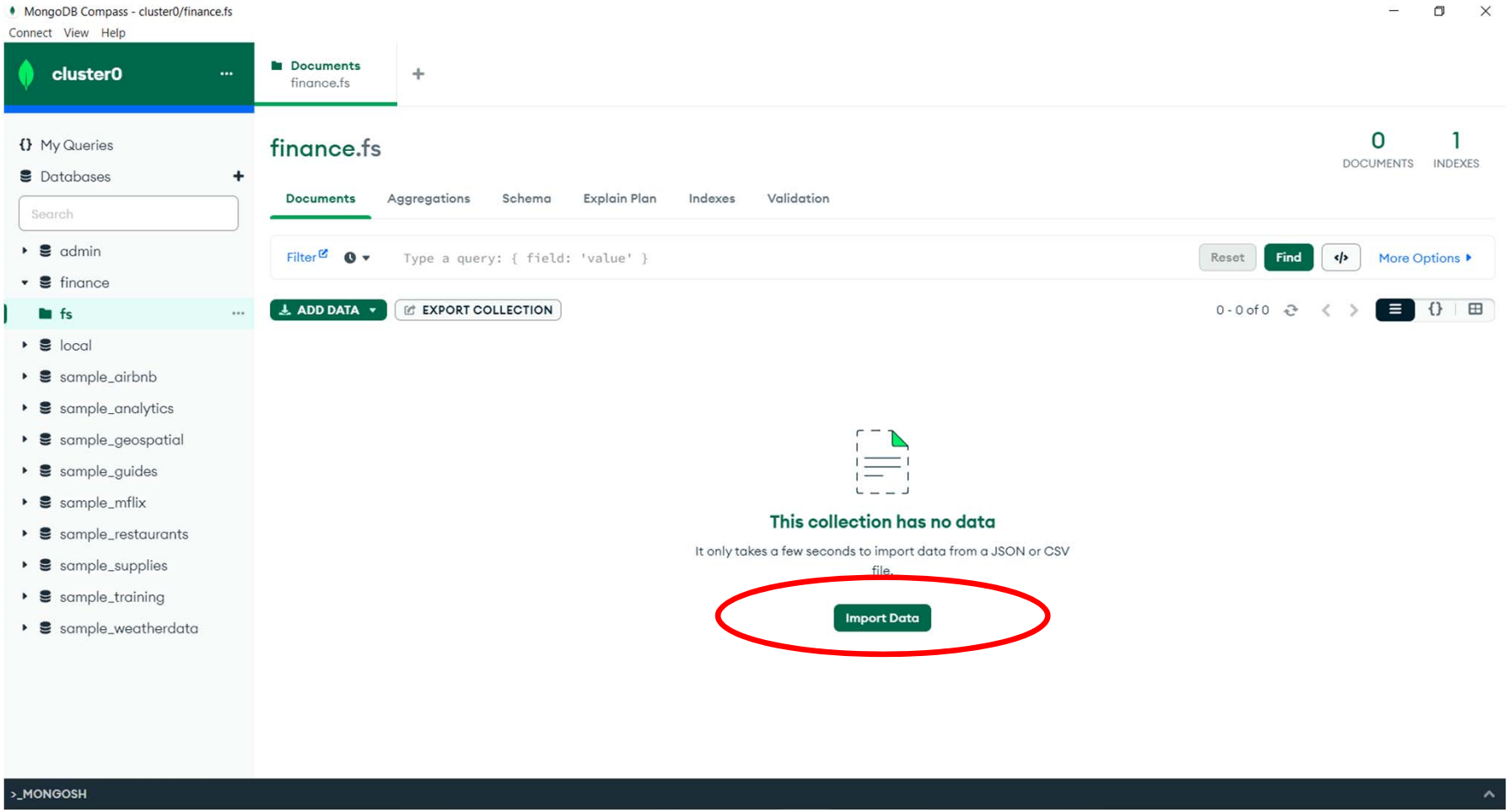
Put “finance” as Database Name “fs” as Collection Name



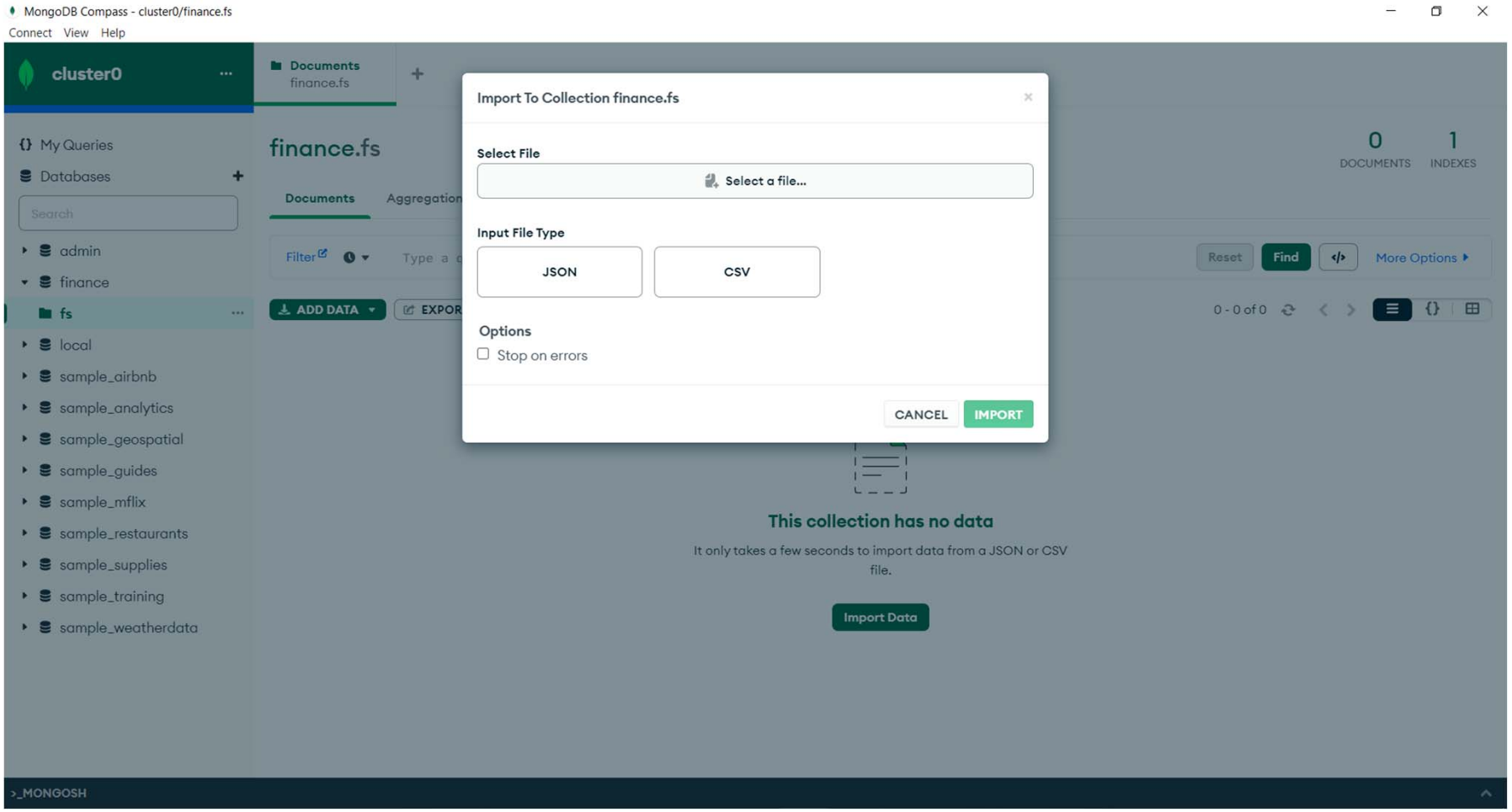
Import FS into MongoDB



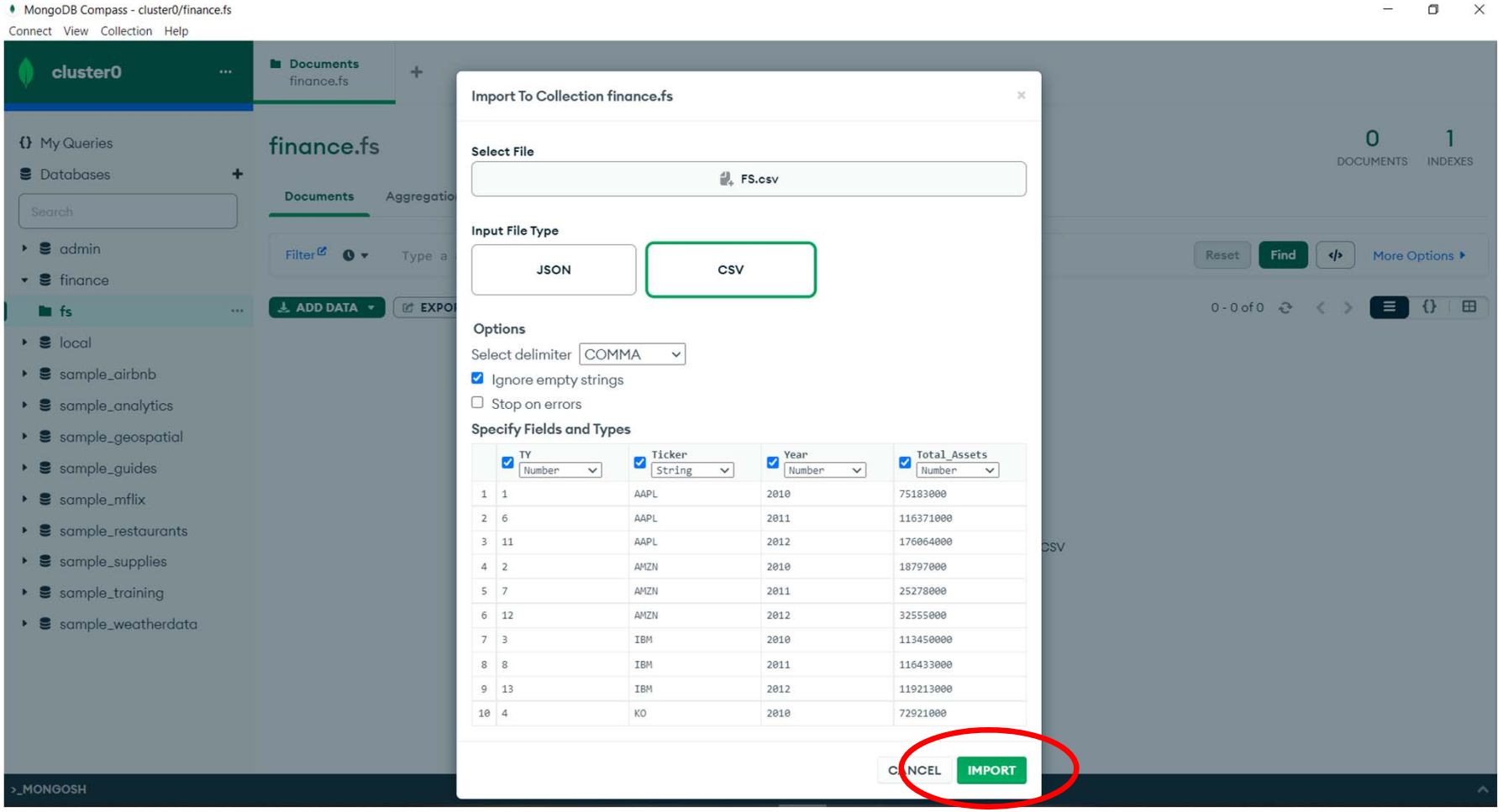
Click “Import Data”

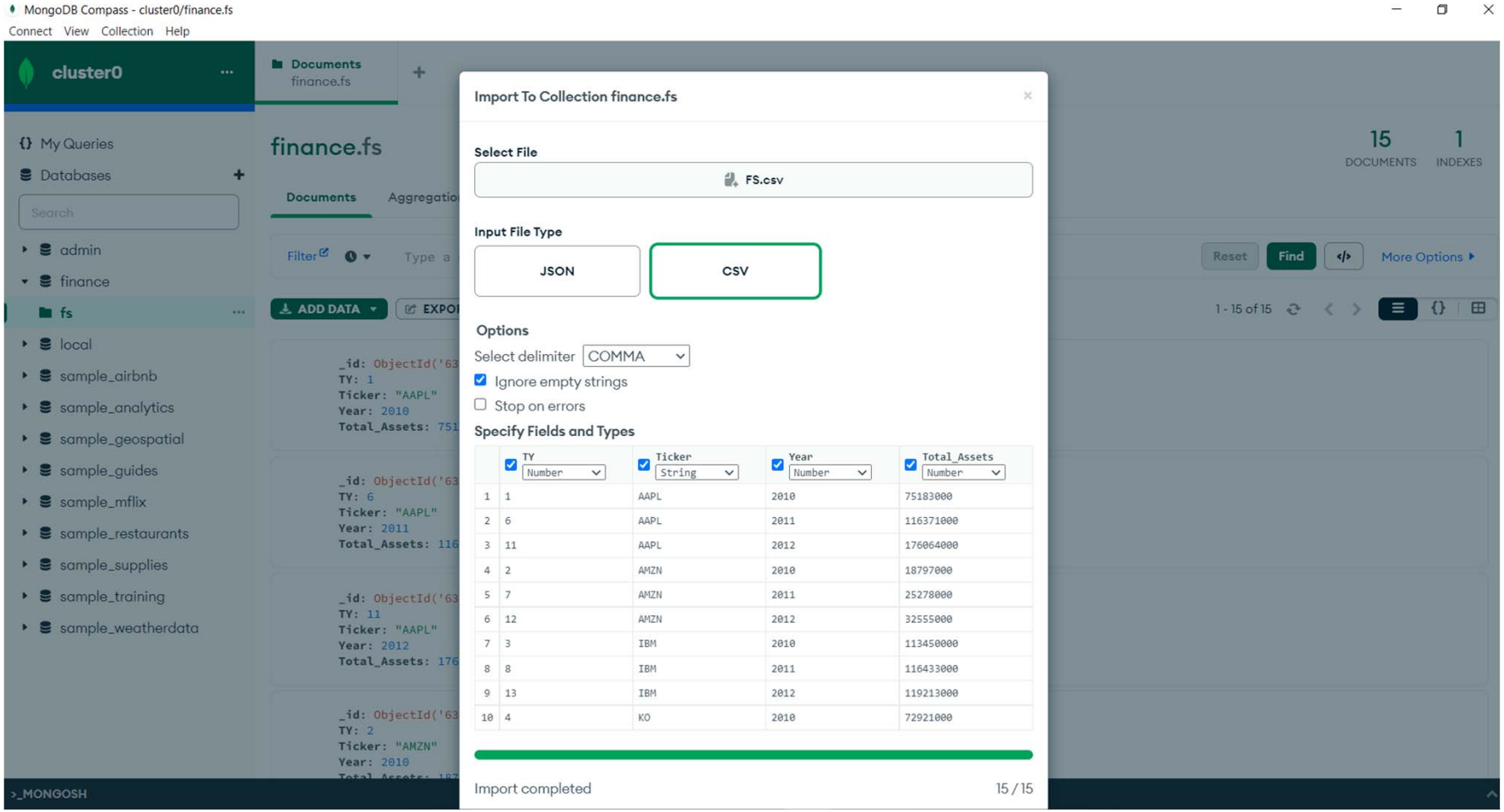


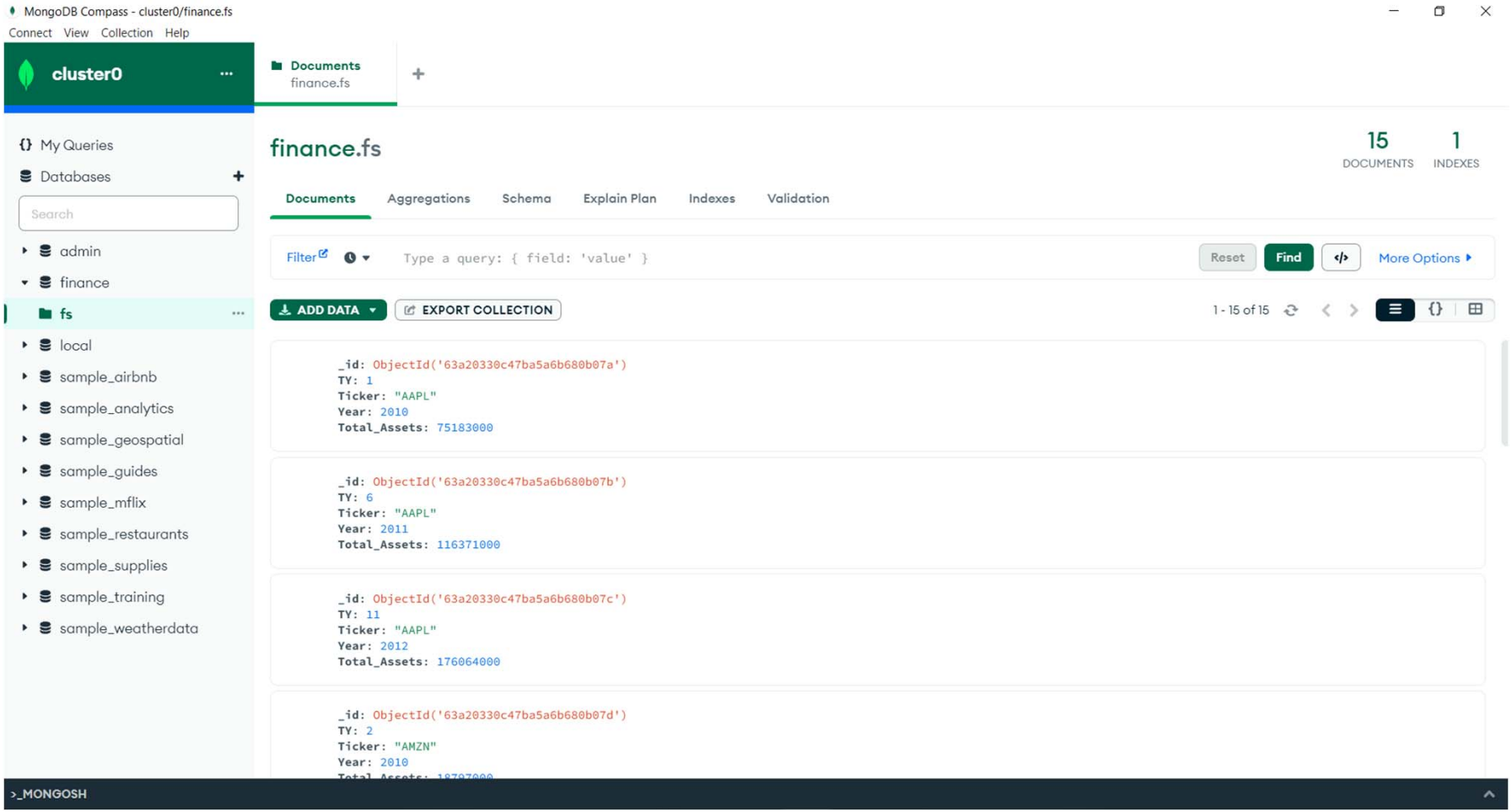
Choose the FS file from your computer Select “CSV”



Choose the correct data type for each field
Click Import




MongoDB Compass - cluster0/finance.fs
Connect View Collection Help
cluster0
My Queries
Databases
admin
finance
fs
local
sample_airbnb
sample_analytics
sample_geospatial
sample_guides
sample_mflix
sample_restaurants
sample_supplies
sample_training
sample_weatherdata
Documents
finance.fs
finance.fs
Documents
Aggregatio
15
DOCUMENTS
1
INDEXES
Filter
Type a
Reset
Find
More Options
ADD DATA
1 - 15 of 15
Import To Collection finance.fs
Select File
FS.csv
Input File Type
JSON
CSV
Options
Select delimiter COMMA
Ignore empty strings
Stop on errors
Specify Fields and Types
TY Number
Ticker String
Year Number
Total_Assets Number
1 1 AAPL 2010 75183000
2 6 AAPL 2011 116371000
3 11 AAPL 2012 176064000
4 2 AMZN 2010 18797000
5 7 AMZN 2011 25278000
6 12 AMZN 2012 32555000
7 3 IBM 2010 113450000
8 8 IBM 2011 116433000
9 13 IBM 2012 119213000
10 4 KO 2010 72921000
Import completed
15 / 15
>_MONGOSH

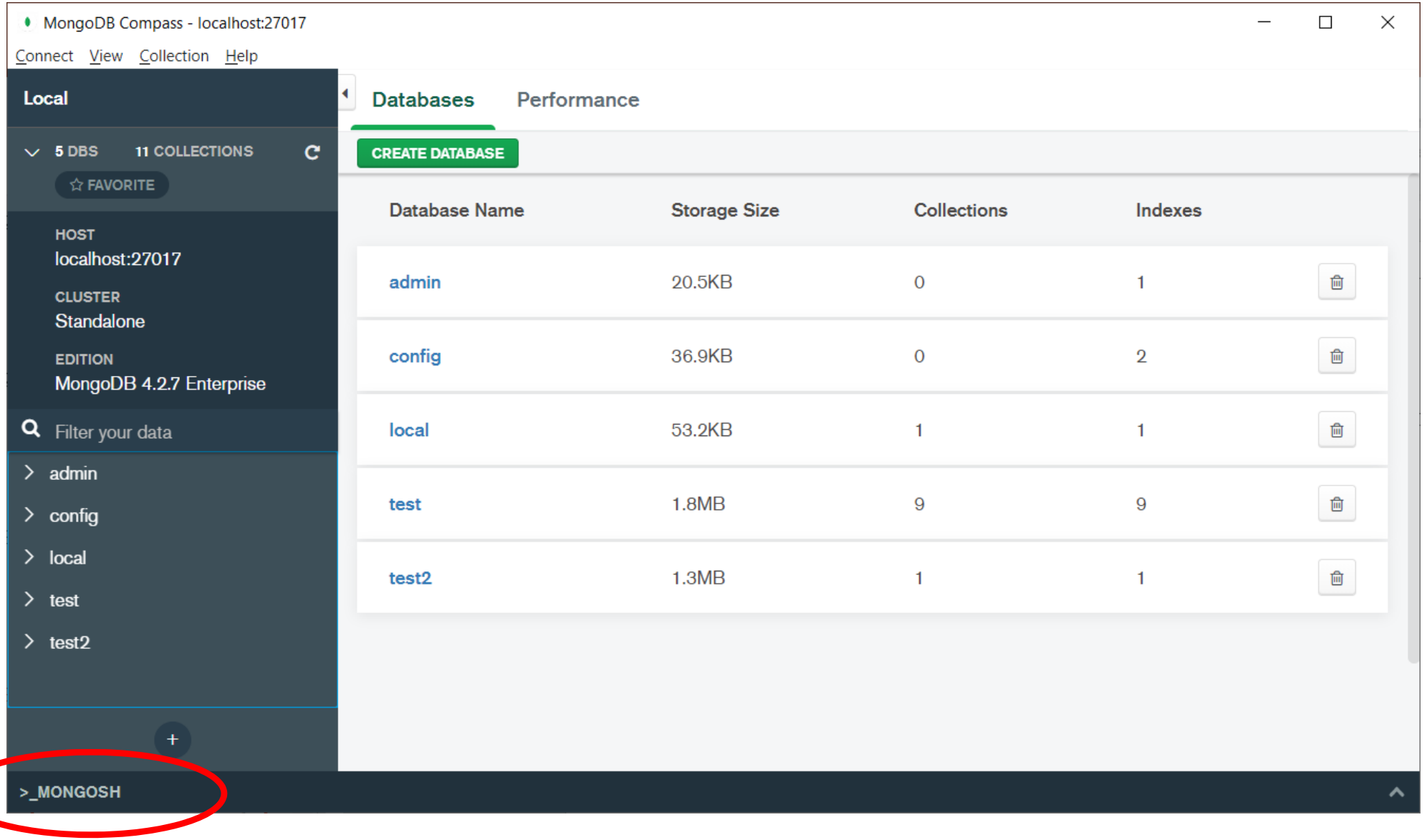

MongoDB Compass - cluster0/finance.fs
Connect View Collection Help
cluster0
Documents
finance.fs
My Queries
Databases
Search
admin
finance
fs
local
sample_airbnb
sample_analytics
sample_geospatial
sample_guides
sample_mflix
sample_restaurants
sample_supplies
sample_training
sample_weatherdata
finance.fs
15
DOCUMENTS
1
INDEXES
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
Filter
Type a query: { field: 'value' }
Reset
Find
More Options
ADD DATA
EXPORT COLLECTION
1 - 15 of 15
_id: ObjectId('63a20330c47ba5a6b680b07a')
TY: 1
Ticker: "AAPL"
Year: 2010
Total_Assets: 75183000
_id: ObjectId('63a20330c47ba5a6b680b07b')
TY: 6
Ticker: "AAPL"
Year: 2011
Total_Assets: 116371000
_id: ObjectId('63a20330c47ba5a6b680b07c')
TY: 11
Ticker: "AAPL"
Year: 2012
Total_Assets: 176064000
_id: ObjectId('63a20330c47ba5a6b680b07d')
TY: 2
Ticker: "AMZN"
Year: 2010
>_MONGOSH

Exercise

- Import DP and Articles into MongoDB and check the results

Writing MongoDB Queries

Click Mongosh at the lower left of Mongo Compass to run MongoDB queries



Show Databases and Tables/Collections

	MongoDB	MySQL
Show database	`show dbs/databases`	`SHOW DATABASES;`
Use database	`use database_name`	`USE database_name`
Show tables/collections	`show collections/tables`	`SHOW TABLES;`
Show all values in a table	`db.collectionName.find()`	`SELECT * FROM tableName`

Create/Drop Table/Collection

- Create Table/Collection:
- MySQL:

```
CREATE TABLE TableName (column1, datatype1, …)
```

- MongoDB:

```
db.createCollection("CollectionName")
```

- Drop Table/Collection:
- MySQL:

```
DROP TABLE TableName
```

- MongoDB:

```
db.collectionName.drop()
```

Create/Drop Table/Collection

- Exercise 1. Create a collection in MongoDB with the name of “t4” and show collections.

```
db.createCollection("t4")
show collections
```

- Exercise 2. Drop the collection “t4” in MongoDB and show collections.

```
db.t4.drop()
show collections
```

Add/Drop Column/Field

- MySQL Add Column:

```
ALTER TABLE TableName ADD new_column_name new_datatype
```

- MySQL Drop Column:

```
ALTER TABLE TableName DROP column_name
```

- MongoDB Add Field:

```
db.collectionName.updateMany({},{$set:{new_field:""}})
```

- MongoDB Drop Field:

```
db.collectionName.updateMany({},{$unset:{"new_field": ""}})
```

Insert Values

- MySQL insert values:

```
INSERT INTO TableName (Column1, Column2,
Column3)
VALUES (Value1, Value2, Value3)
```

- MongoDB insert values:

```
db.collectionName.insertOne({ Field1: Value1,
Field2: Value2, Field3: Value3})
```

Exercise

- Create collection `olympics` in MongoDB with the following data.

Year	Location
2016	Rio de Janeiro
2020	Tokyo

```
db.olympics.insertOne({Year: 2016, Location: 'Rio de Janeiro'})
db.olympics.insertOne({Year: 2020, Location: 'Tokyo'})
```

OR

```
db.olympics.insertMany([{Year: 2016, Location: 'Rio de Janeiro'},
{Year: 2020, Location: 'Tokyo'}])
```

Select Statements

- MySQL:

```
SELECT Column1, Column2 FROM TableName
```

- MongoDB:

```
db.collectionName.find({ }, { Field1: 1, Field2: 1})
```

- Exercise: select fields Ticker and Year in the collection "fs"

```
db.fs.find({ },{ Ticker: 1, Year: 1 })
```

- Hide id:

```
db.fs.find({ },{ Ticker: 1, Year: 1, _id:0 })
```

Select Statements – Limit Row

- MySQL:

```
SELECT * FROM TableName LIMIT 10
```

- MongoDB:

```
db.collectionName.find().limit(10)
```

- Exercise: Show the first 5 records in collection “fs”

```
db.fs.find().limit(5)
```

Select Statements - WHERE

- MySQL:

```
SELECT Column1, Column2 FROM TableName
WHERE Column1 = Value1
```

- MongoDB:

```
db.collectionName.find({Field1: Value1}, {Field1:
1, Field2: 1})
```

- Exercise: Select field Ticker and Year of AAPL in the collection “fs”

```
db.fs.find({ Ticker: "AAPL" }, { Ticker: 1,
Year: 1})
```

Select Statements - Comparison

- MySQL:
  ```
  SELECT * FROM TableName WHERE Column1 > Value1
  ```
- MongoDB:
  ```
  db.collectionName.find({ Field1: { $gt: Value1 } })
  ```
  - `$gt:  >` `$lt: <`
  - `$gte: >=` `$lte: <=`
- Exercise: Select Ticker, Year, and Total_Assets of all records before 2012 (2012 is not included) in the collection "fs".

```
db.fs.find({Year:{$lt: 2012}}, {Ticker: 1,
Year: 1, Total_Assets: 1})
```

If it does not work, it may be related to data type (String vs Number). Try:

```
db.fs.update( {},
[{ $set: { Year: { $toDouble: "$Year" } } }],
{ multi: true })
```

Select Statements – And/Or

- MySQL:

```
SELECT * FROM TableName WHERE Column1 >= Value1
AND Column1 < Value2
```

- MongoDB:

```
db.collectionName.find({ Field1: { $gte: Value1,
$lt: Value2 } })
```

- MySQL:

```
SELECT * FROM TableName WHERE Column1=Value1 OR
Column2=Value2
```

- MongoDB:

```
db.collectionName.find({ $or: [{ Field1:
Value1 } , { Field2: Value2 }] })
```

Select Statements – And/Or

- Exercise: Translate the SQL to MongoDB query and hide id
- ```
  SELECT Ticker, Year, Total_Assets
  FROM fs
  WHERE Year>2011 OR Total_Assets<=80000000
  ```

```
db.fs.find({ $or:[ {Year: { $gt: 2011 } },
{ Total_Assets : { $lte: 80000000}} ] },
{ Ticker:1, Year:1, Total_Assets:1, _id:0})
```
```

Select Statements – Like

- MySQL:

```
SELECT * FROM TableName WHERE Column1 LIKE "%Value%"
```

- MongoDB:

```
db.collectionName.find( { Field1: /Value/ } )
```

- MySQL:

```
SELECT * FROM TableName WHERE Column1 LIKE "Value%"
```

- MongoDB:

```
db.collectionName.find( { Field1: /^Value/ } )
```

Select Statements – Like

- Exercise: Translate the SQL to MongoDB query and hide id

```
SELECT Ticker, Year, Total_Assets
FROM fs
WHERE Year>2011 OR Total_Assets<=80000000
AND Ticker LIKE "AA%"
```

```
db.fs.find({ $or:[ {Year: { $gt: 2011 } } ,
{ Total_Assets : { $lte: 80000000}} ],
Ticker: /^AA/ },
{ Ticker:1, Year:1, Total_Assets:1, _id:0})
```

Select Statements – Count Record Number

- MySQL:

```
SELECT COUNT(*) FROM TableName
SELECT COUNT(*) FROM TableName WHERE
Field1=Value1
```

- MongoDB:

```
db.collectionName.countDocuments({})
db.collectionName.countDocuments({Field1: Value1})
```

- Exercise: Count the number of records in collection “fs”

```
db.fs.countDocuments()
```

Using MongoDB with Python
(for reference only)

Import Data Into MongoDB Using Python

- In Anaconda Prompt: `pip install pymongo`

  or

  In Jupyter Notebook: `conda install pymongo`

-
```
import pandas as pd
from pymongo import MongoClient
connection_str = "mongodb..."
client = MongoClient(connection_str)
df=pd.read_csv(r'D:\MSBA7024\fs.csv')
data = df.to_dict(orient='records')
collection = client['finance']['fs1']
collection.insert_many(data)
```

Extract Data From MongoDB Using Python

Extract all records from the fs1 collection:

```
import pandas as pd
from pymongo import MongoClient
connection_str = "mongodb..."
client = MongoClient(connection_str)
db = client.finance
collection = db.fs1
data = pd.DataFrame(list(collection.find()))
print(data.head())
```

Extract Data From MongoDB Using Python

Extract only records of AAPL:

```
import pandas as pd
from pymongo import MongoClient
connection_str = "mongodb..."
client = MongoClient(connection_str)
db = client.finance
collection = db.fs1
data = pd.DataFrame(list(collection.find({"Ticker":"AAPL"})))
print(data.head())
```

Exercise

- Import DP and Articles into MongoDB through Python and check the results

Q&A