

# MongoDB Indexing



Estimated time needed: **30** minutes

## Objectives

After completing this lab, you will be able to:

- Measure the time it takes to execute a query with the explain function
- Describe the process of creating, listing and deleting indexes
- Evaluate the effectiveness of an index

## About Skills Network Cloud IDE

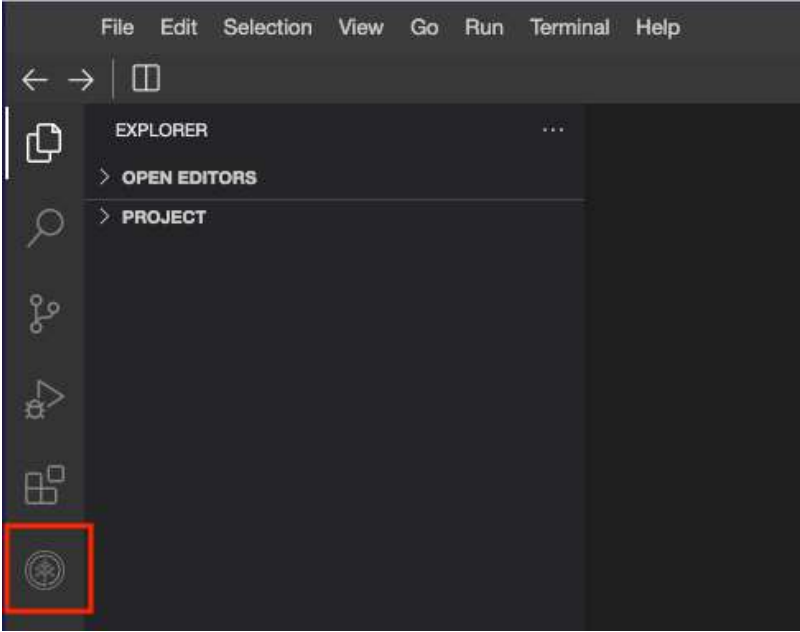
Skills Network Cloud IDE (based on Theia and Docker) provides an environment for hands on labs for course and project related labs. Theia is an open source IDE (Integrated Development Environment), that can be run on desktop or on the cloud. To complete this lab, we will be using the Cloud IDE based on Theia and MongoDB running in a Docker container.

## Important Notice about this lab environment

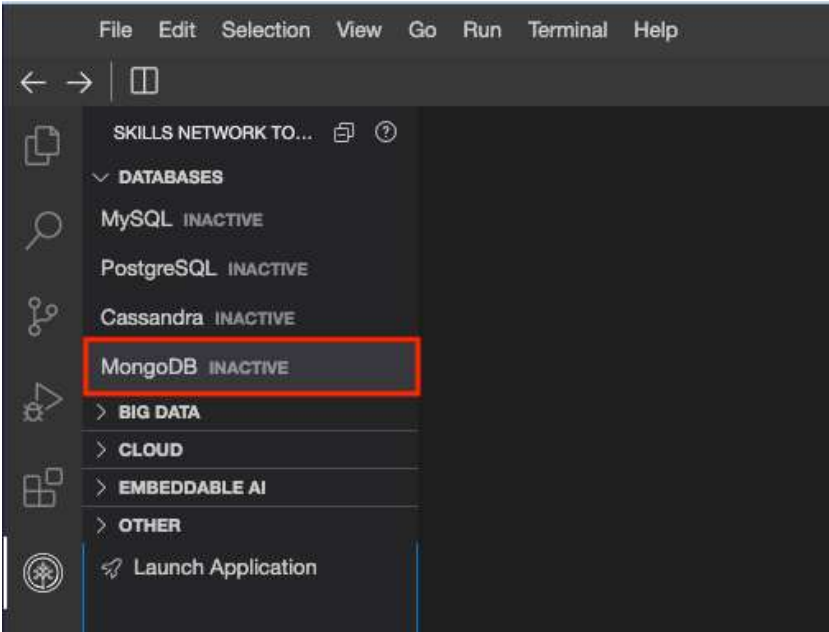
Please be aware that sessions for this lab environment are not persisted. Every time you connect to this lab, a new environment is created for you. Any data you may have saved in the earlier session would get lost. Plan to complete these labs in a single session, to avoid losing your data.

## Set-up: Start MongoDB

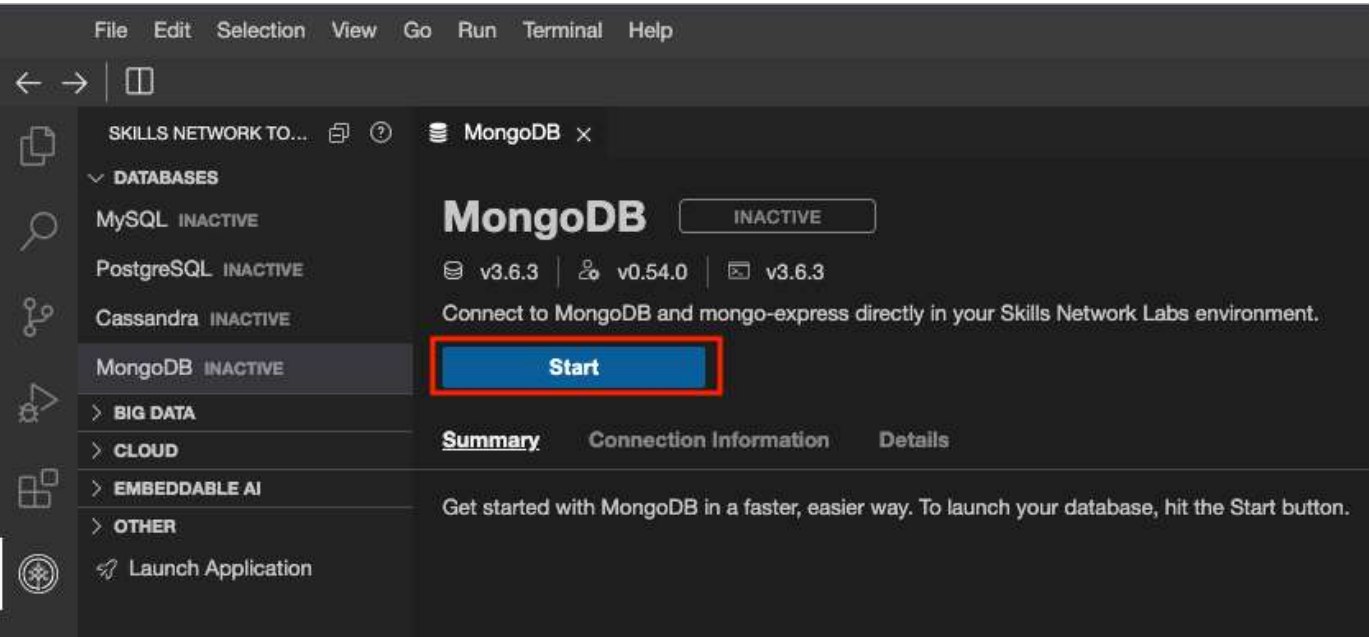
Navigate to Skills Network Toolbox.



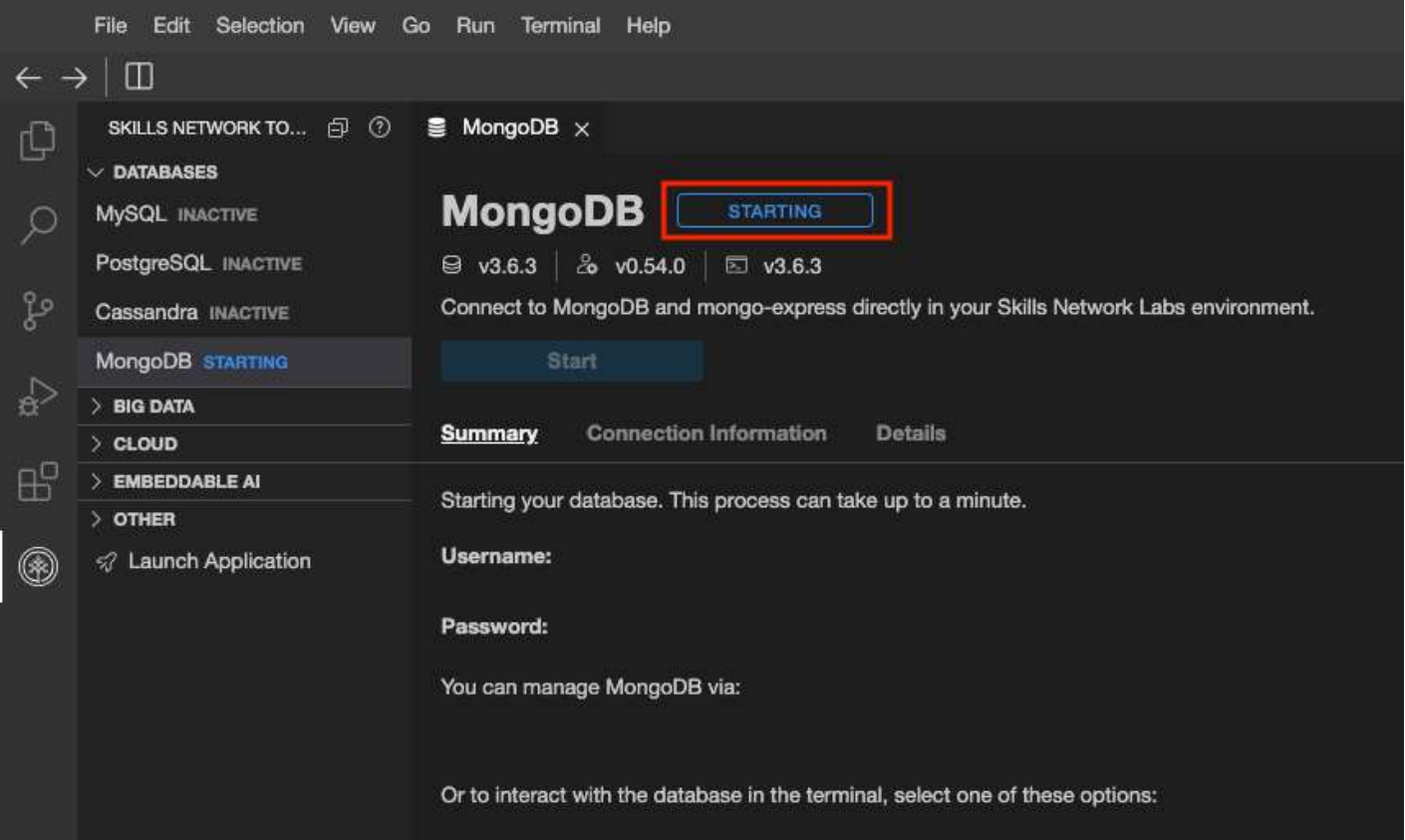
You will notice MongoDB listed there, but inactive. Which means the database is not available to use.



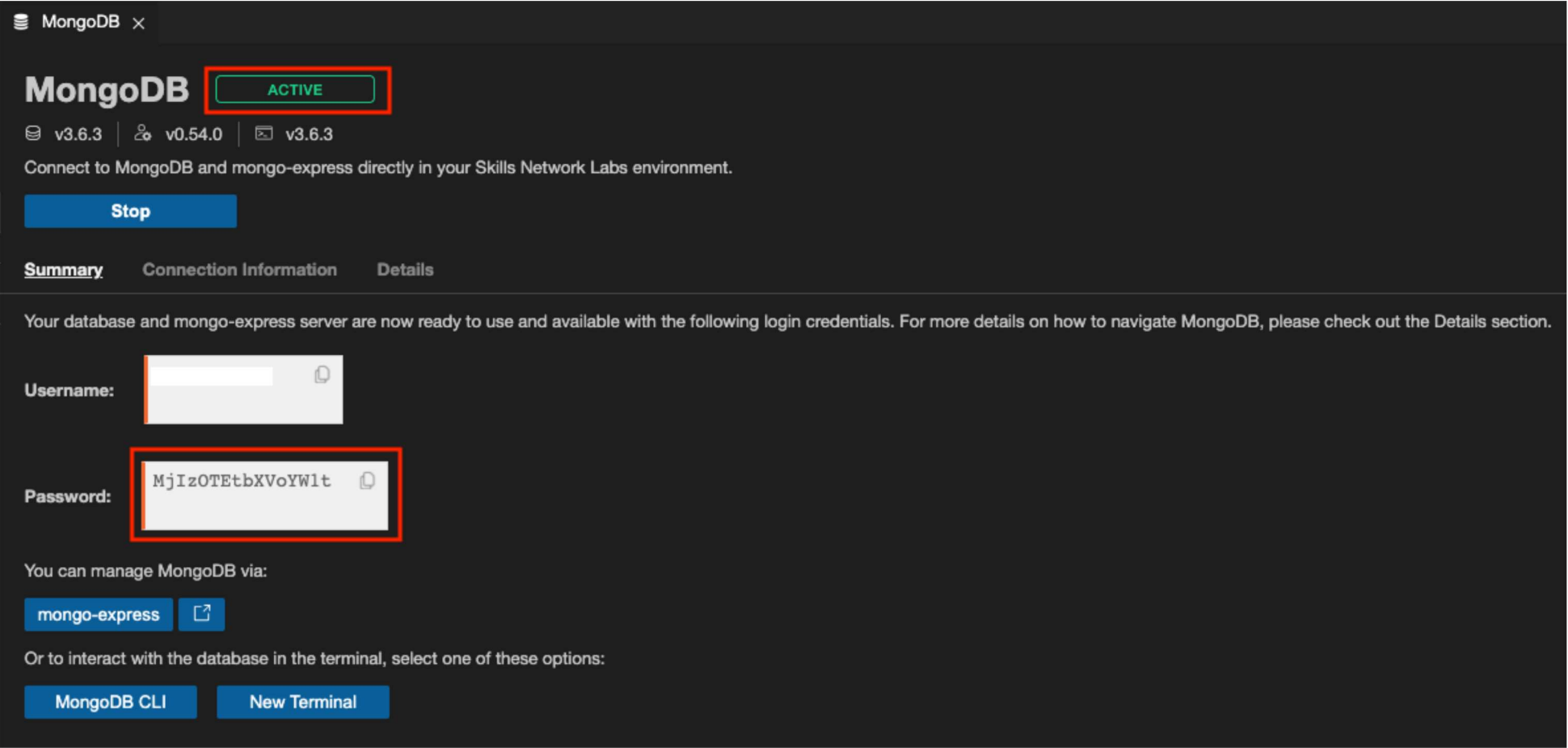
Once you click on it, you will see more details about it and a button to start it.



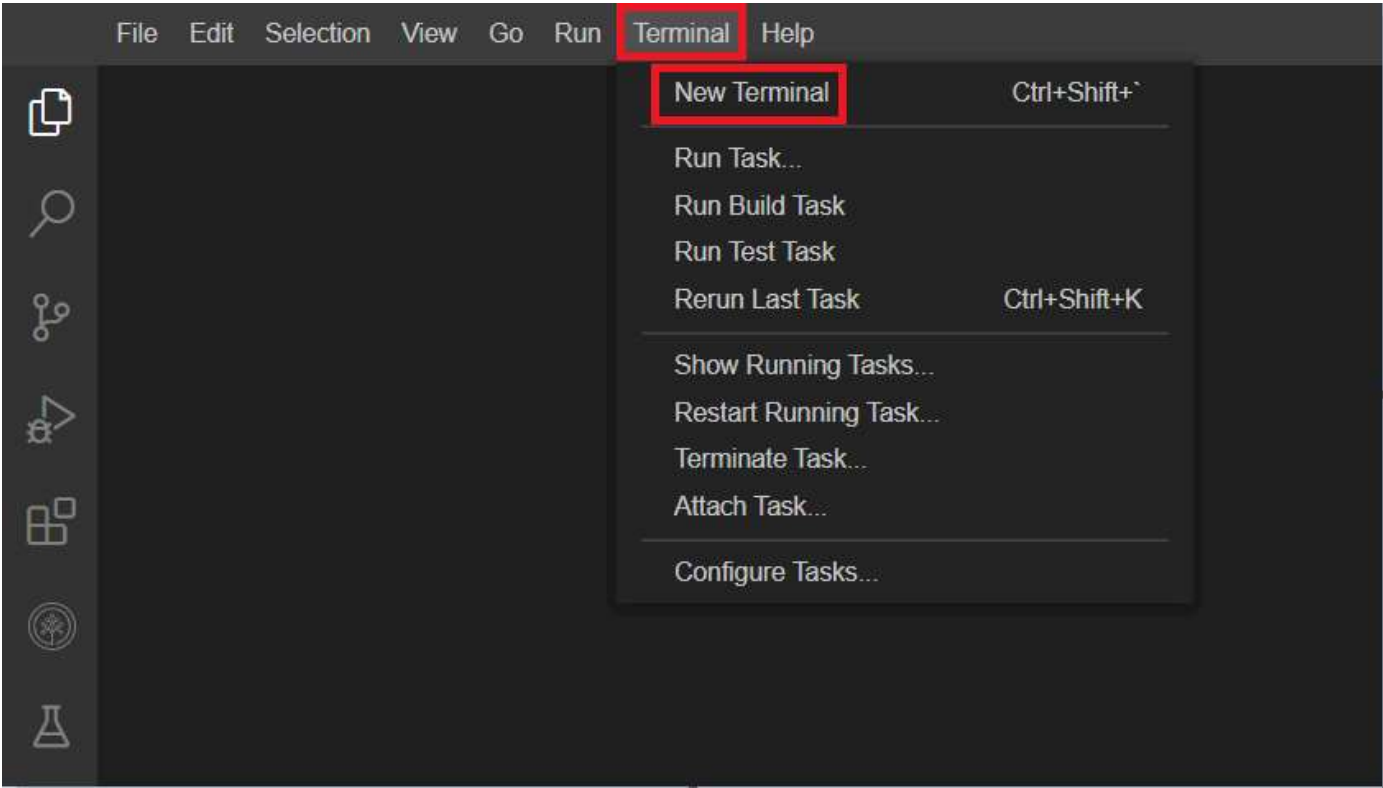
Clicking on the start button will run a background process to configure and start your MongoDB server.



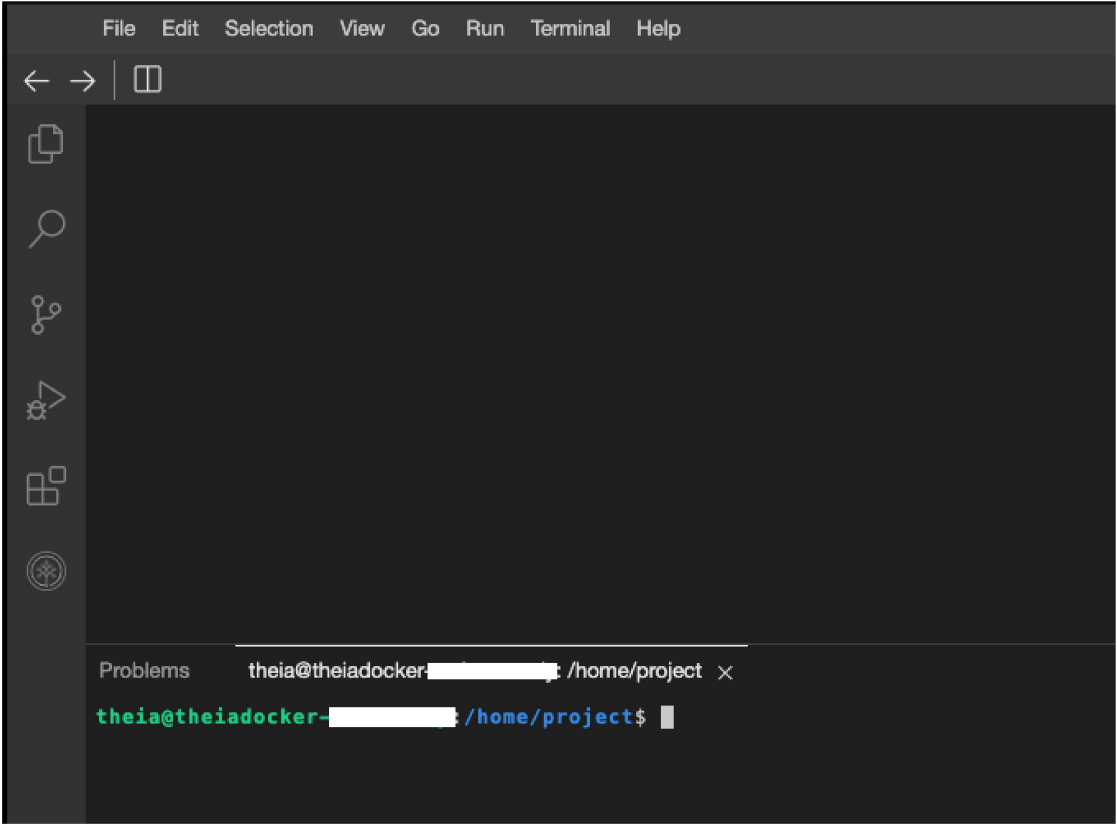
Shortly after that, your server is ready for use. This deployment has access control enabled and MongoDB enforces authentication. So, take note of the password as you will need it to login as root user.



You can now open terminal and enter details yourself.



This will open a new terminal at the bottom of the screen as in the image below.

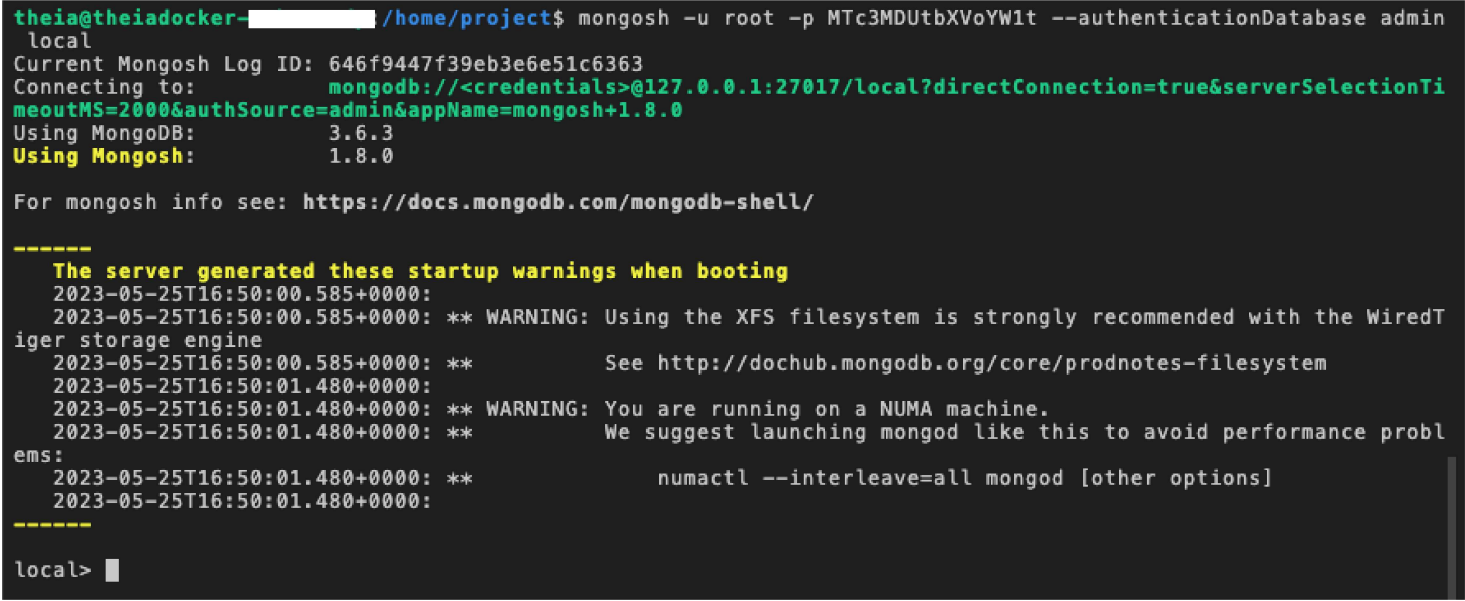


Run the below command on the newly opened terminal. (You can copy the code by clicking on the little copy button on the bottom right of the codeblock below and then paste it, wherever you wish.)

1. 1
1. mongosh -u root -p PASSWORD --authenticationDatabase admin local

Copied!

Executed!



The command contains the username and password to connect to mongodb server (the text after the -p option is the password). Your output would be different from the one shown above. Copy the command given to you, and keep it handy. You will need it in the next step.

Or you can simply click on MongoDB CLI which does that for you.

MongoDB

×

MongoDB

ACTIVE

v3.6.3

v0.54.0

v3.6.3

Connect to MongoDB and mongo-express directly in your Skills Network Labs environment.

Stop

Summary

Connection Information

Details

Your database and mongo-express server are now ready to use and available with the following login credentials. For more details on how to navigate MongoDB, please check out the Details section.

Username:

Password:

MjIzOTEtbXVoYW1t

You can manage MongoDB via:

mongo-express

Or to interact with the database in the terminal, select one of these options:

MongoDB CLI

New Terminal

In MongoDB CLI (for example, mongo shell), switch the context to training database.

- ```
1. 1
1. use training
```

Copied!

And create a collection called bigdata

- ```
1. 1
1. db.createCollection("bigdata")
```

Copied!

## Exercise 1 - Insert documents

- Let's insert 200k documents into the newly created collection.
- This should take a few seconds to complete.
- The code given below will insert these documents into the bigdata collection.
- Each document has a field named account\_no which is assigned to incrementing variable i.
- Another field balance contains a randomly generated number, to simulate the bank balance for the account.

Copy the below code and paste it on the mongo client.

- ```
1. 1
2. 2
3. 3

1. docsToInsert = []
2. for (i = 1; i <= 200000; i++) { docsToInsert[i-1] = { "account_no": i, "balance": Math.round(Math.random() * 20000) } }
3. db.bigdata.insertMany(docsToInsert);
```

Copied!

Verify that 200000 documents got inserted by running the below command.

- ```
1. 1
1. db.bigdata.countDocuments()
```

Copied!

## Exercise 2 - Measure the time taken by a query

Let's run a query and find out how much time it takes to complete. You will query for the details of account number 58982.

We will make use of the explain function to find the time taken to run the query in milliseconds.

The db.collection.explain(“executionStats”) method provides statistics about the performance of a query. These statistics can be useful in measuring if and how a query uses an index. See db.collection.explain() for details.

Run the below command.

- ```
1. 1
1. db.bigdata.find({"account_no":58982}).explain("executionStats").executionStats.executionTimeMillis
```

Copied!

Make a note of the milliseconds it took to run the query. We will need this at a later stage.

## Exercise 3 - Working with indexes

Before you create an index, choose the field you wish to create an index on. It is usually the field that you query most.

Run the below command to create an index on the field account\_no.

- ```
1. 1
1. db.bigdata.createIndex({"account_no":1})
```

Copied!

Where 1 means ascending order.

Run the below command to get a list of indexes on the bigdata collection.

```
1. 1

1. db.bigdata.getIndexes()
```

Copied!

You should see an index named account\_no\_1

## Exercise 4 - Find out how effective an index is

You will now run the same query for account 58982 and compare the execution time from previous run. This way, you can compare the improvement.

Run the below command.

```
1. 1

1. db.bigdata.find({"account_no":58982}).explain("executionStats").executionStats.executionTimeMillis
```

Copied!

This time, the execution time should be a lot less than previously. If you see 0, it means the query completed under 1 millisecond.

## Exercise 6 - Delete an index

Use the below command to delete the index we created earlier. Here you can provide index definition or name.

```
1. 1

1. db.bigdata.dropIndex({"account_no":1})
```

Copied!

## Bonus information

MongoDB creates a unique index on the \_id field during the creation of a collection. The \_id index prevents clients from inserting two documents with the same value for the \_id field. You cannot drop this index on the \_id field.

## Practice exercises

1. Problem:
- Create an index on the balance field.

▼ Click here for Hint

```
use the command db.collection.createIndex()
```

▼ Click here for Solution

On the mongo client run the below commands.

```
1. 1

1. db.bigdata.createIndex({"balance":1})
```

Copied!

2. Problem:
- Query for documents with a balance of 10000 and record the time taken.

▼ Click here for Hint

```
use the command db.collection.find().explain()
```

▼ Click here for Solution

```
1. 1

1. db.bigdata.find({"balance":10000}).explain("executionStats").executionStats.executionTimeMillis
```

Copied!

3. Problem:
- Drop the index you have created.

▼ Click here for Hint

```
use the command db.collection.dropIndex()
```

▼ Click here for Solution

```
1. 1

1. db.bigdata.dropIndex({"balance":1})
```

Copied!

4. Problem:
- Query for documents with a balance of 10000 and record the time taken, and compare it with the previously recorded time.

▼ Click here for Hint

```
use the command db.collection.find().explain()
```

▼ Click here for Solution

```
1. 1

1. db.bigdata.find({"balance": 10000}).explain("executionStats").executionStats.executionTimeMillis
```

Copied!

# Summary

In this lab, you have gained an understanding of indexing in MongoDB.

## Author(s)

[Muhammad Yahya](#)

(C) IBM Corporation. All rights reserved.