Getting Started with MongoDB

Estimated time needed: **30** minutes

Objectives

After completing this lab, you will be able to:

- Access the MongoDB server using the command-line interface
- Describe the process of listing and creating collections, which contain documents and databases, which contain one or more collections
- Perform basic operations on a collection, such as inserting, counting, and listing documents

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 you can run on a desktop or the cloud. To complete this lab, you will use 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 do not persist. You will see a new environment every time you connect to this lab. 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.
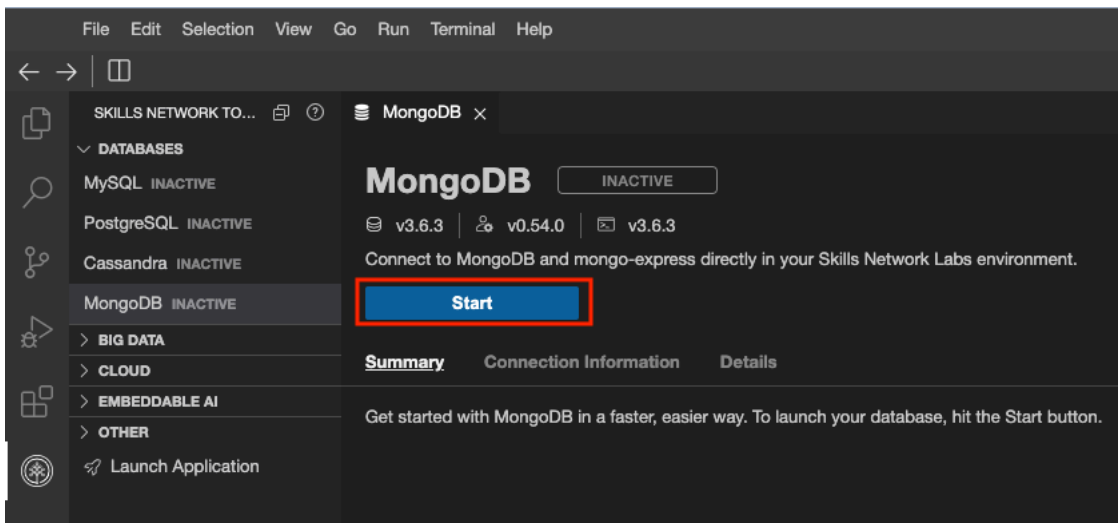
Set-up: Start MongoDB

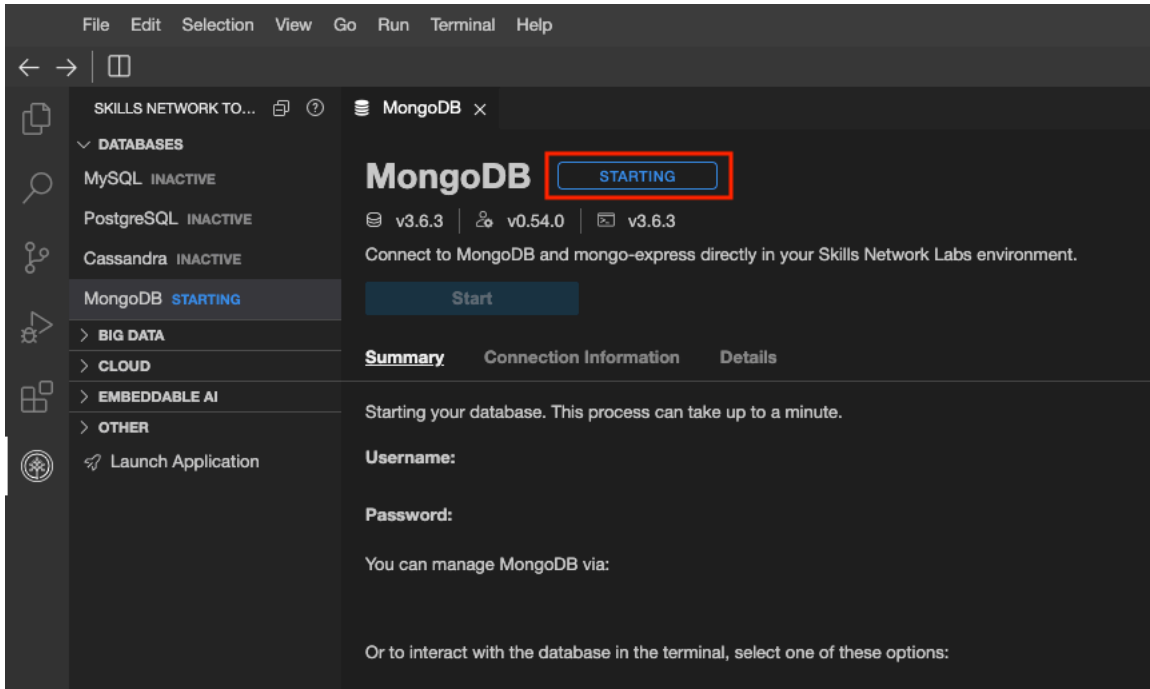Navigate to Skills Network Toolbox.
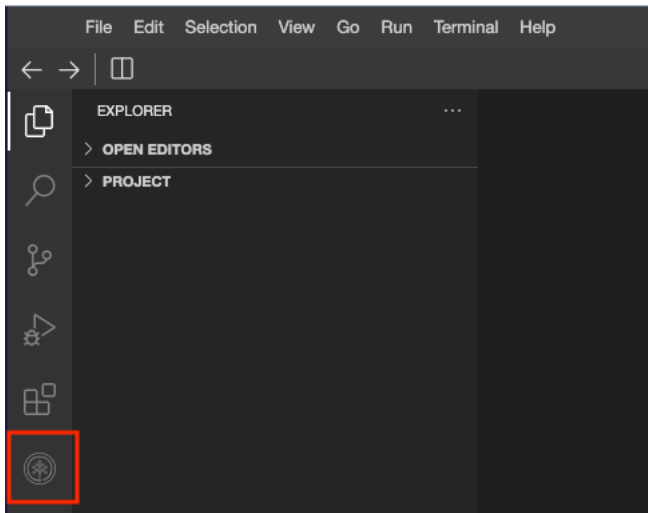
You will notice MongoDB is listed there but inactive. Which means the database is not available for use.
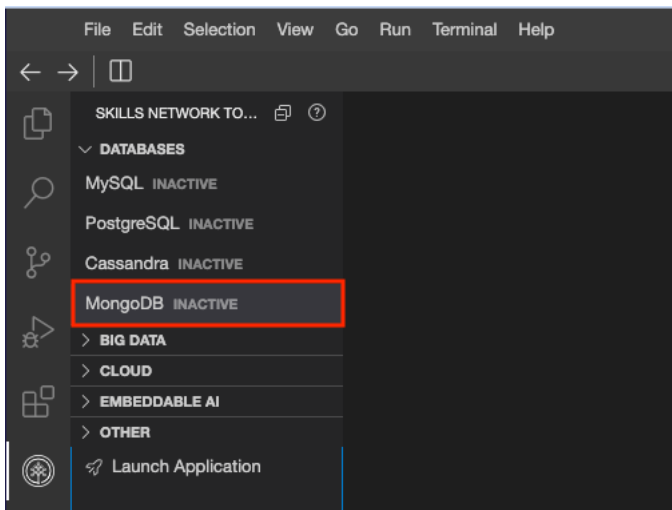
Once you select MongoDB, you will see more details and a place to start it.

Clicking Start 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 log in as the `root` user.

You can now open the terminal and enter details yourself.

This action will open a new terminal at the end of the screen, as in the following image.
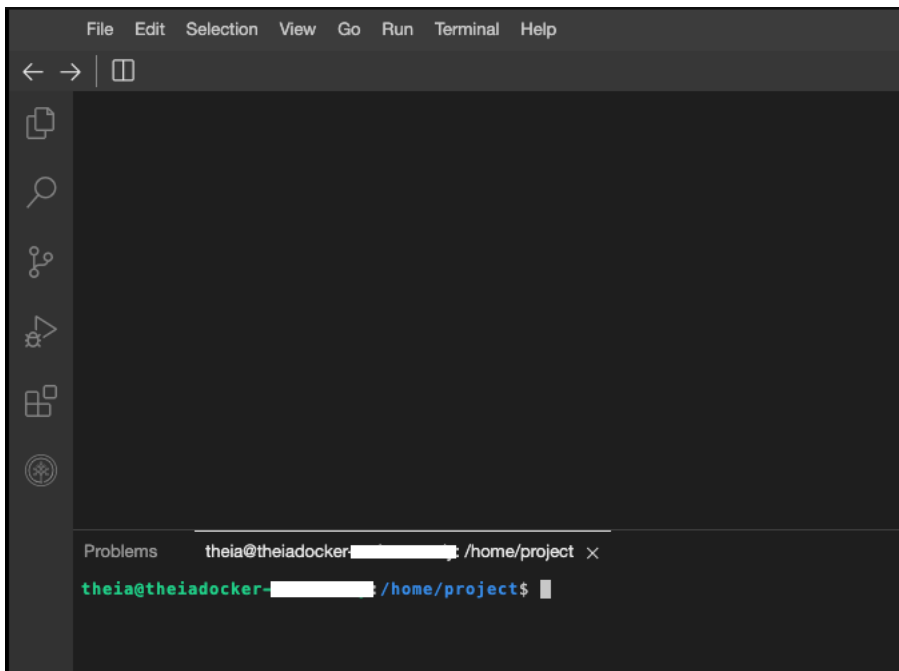
Run the following command on the newly opened terminal. (Copy the code by selecting copy on the right of the code block and then paste it wherever you wish.)

1. 1

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

Copied! Executed!

The command contains the username and password to connect to the MongoDB server (the text after the -p option is the password). Your output will be different from the one shown here. Copy the command given to you, or click MongoDB CLI. You will need this command in the next step.

Exercise 1: Find the version of the server

On the Mongo client, run the following command.

```
1. 1
1. db.version()
```

Copied!

This code will show the version of the MongoDB server.

Exercise 2: List databases

On the Mongo client, run the following command.

```
1. 1
1. show dbs
```

Copied!

This command will print a list of the databases present on the server, including default and user-defined.

Exercise 3: Create database

On the Mongo client, run the following command.

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

Copied!

This command will switch the context to the database named `training`. If the database `training` doesn't exist, MongoDB will create it for you when you insert data.

Exercise 4: Create collection

On the Mongo client, run the following command.

```
1. 1
1. db.createCollection("mycollection")
```

Copied!

This command will create a collection name `mycollection` inside the `training` database.

Exercise 5: List collections

On the Mongo client, run the following command.

```
1. 1
```

```
1. show collections
```

Copied!

This command will print the list of collections in your current database.

Exercise 6: Insert documents into a collection

On the Mongo client, run the following command.

```
1. 1
```

```
1. db.mycollection.insert({"color":"white","example":"milk"})
```

Copied!

The above command inserts the json document {"color":"white","example":"milk"} into the collection.

Let us insert one more document.

```
1. 1
```

```
1. db.mycollection.insert({"color":"blue","example":"sky"})
```

Copied!

The previous command inserts the JSON document {"color":"blue","example":"sky"} into the collection.

Insert three more documents of your choice.

Exercise 7: Count the number of documents in a collection

On the Mongo client, run the following command.

```
1. 1
```

```
1. db.mycollection.countDocuments()
```

Copied!

This command gives you the number of documents in the collection.

Exercise 8: List all documents in a collection

On the Mongo client, run the following command.

```
1. 1
```

```
1. db.mycollection.find()
```

Copied!

This command lists all the documents in the collection **mycollection**

Notice that MongoDB automatically adds an `_id` field to every document to identify the document.

Exercise 9: Disconnect from MongoDB server

On the Mongo client, run the following command.

```
1. 1
```

```
1. exit
```

Copied!

Practice exercises

1. Problem:

   Connect to mongodb server.

▼ Click here for hint

   Use the `mongosh` command with correct username and password.

▼ Click here for solution
Use the below command with the user name `root` and password generated earlier.

```
1. 1
```

```
1. mongosh -u root -p NTc0My1yc2FubmFy --authenticationDatabase admin local
```

Copied!

2. Problem:

List databases.

▼ Click here for hint

Use the `show` command with dbs option.

▼ Click here for solution

1. 1

1. show dbs

Copied!

3. Problem:

Create a database named `mydatabase`.

▼ Click here for hint

Use the `use` command with the database name.

▼ Click here for solution

1. 1

1. use mydatabase

Copied!

4. Problem:

Create a collection named `Landmarks` in the database `mydatabase`.

▼ Click here for hint

Use the `createCollection` command.

▼ Click here for solution

1. 1

1. db.createCollection("landmarks")

Copied!

5. Problem:

List collections

▼ Click here for hint

Use the `show` command with `collections` option.

▼ Click here for solution

1. 1

1. show collections

Copied!

6. Problem:

Insert details of five landmarks, including name, city, and country. Example: Eiffel Tower, Paris, France.

▼ Click here for hint

Use the `db.collection.insert()` command with the correct options.

▼ Click here for solution

1. 1
2. 2
3. 3
4. 4
5. 5

1. db.landmarks.insert({"name":"Statue of Liberty","city":"New York","country":"USA"})
2. db.landmarks.insert({"name":"Big Ben","city":"London","country":"UK"})
3. db.landmarks.insert({"name":"Taj Mahal","city":"Agra","country":"India"})
4. db.landmarks.insert({"name":"Pyramids","country":"Egypt"})
5. db.landmarks.insert({"name":"Great Wall of China","country":"China"})

Copied!

7. Problem:

Count the number of documents you have inserted.

▼ Click here for hint

Use the `countDocuments` command on your collections.

▼ Click here for solution

1. 1

1. `db.landmarks.countDocuments()`

Copied!

8. Problem:

List the documents.

▼ Click here for hint

Use the `db.collection.find()` command.

▼ Click here for solution

1. 1

1. `db.landmarks.find()`

Copied!

9. Problem:

Disconnect from the server.

▼ Click here for hint

Use the `exit` command.

▼ Click here for solution

1. 1

1. `exit`

Copied!

Summary

In this lab, you have gained an understanding of basic commands to interact with MongoDB by performing Insert and Read operations.

Author(s)

[Muhammad Yahya](#)