

Hands-on Lab: Setup and Practice Assignment



Estimated time needed: 30 minutes

Objectives

In this assignment, you will:

- Export data from a MongoDB database.
- Import data into a MongoDB database.
- Export data from a Cassandra database.
- Import data into a Cassandra database.
- Create indexes on a Cassandra database.

About this SN Labs Cloud IDE

This Skills Network Labs Cloud IDE provides a hands-on environment for course and project-related labs. It utilizes Theia, an open-source IDE platform that runs on a desktop or the cloud. To complete this lab, you will use the Cloud IDE based on Theia Cassandra and MongoDB running in a Docker container.

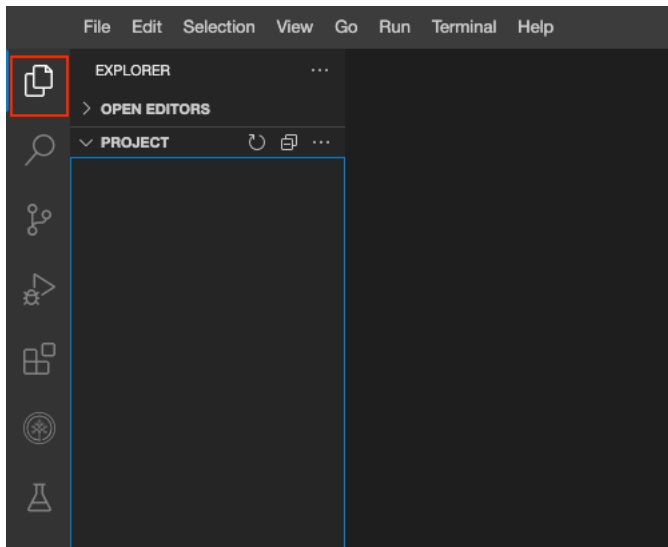
Important notice about this lab environment

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.

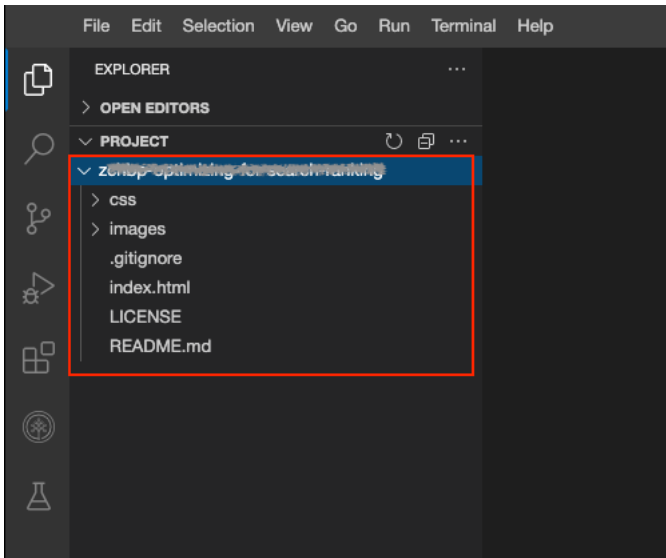
Working with Files in Cloud IDE

If you are new to Cloud IDE, this section will show you how to create and edit files in Cloud IDE.

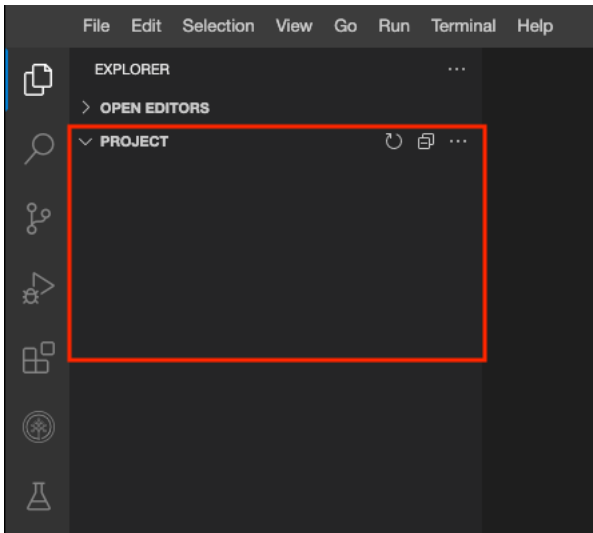
To view your files and directories in Cloud IDE, click the file icon to reveal it.



If you have cloned (using the `git clone` command) boilerplate or starting code, then it will look like the following image:

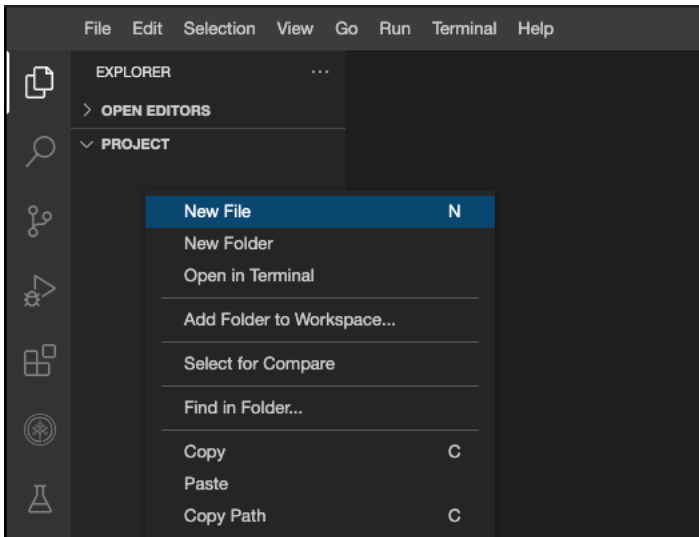


If you have not cloned and are starting with a blank project, it will look like this:

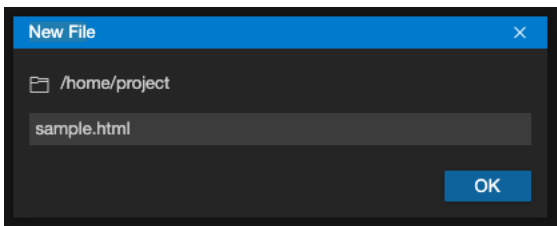


Create a new file

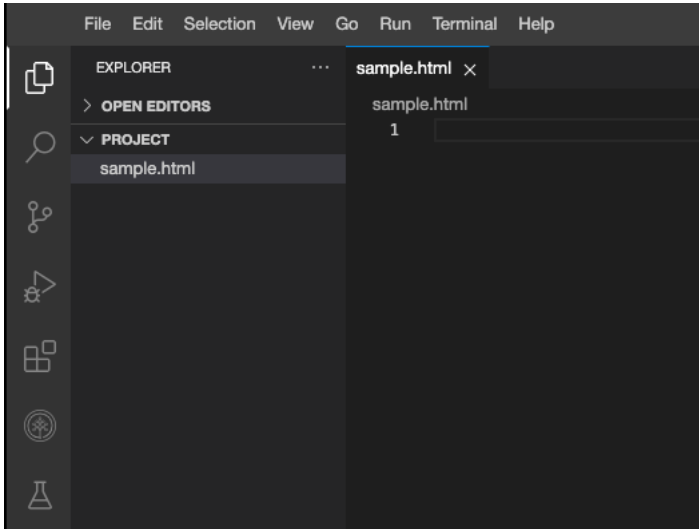
To create a new file in your project, right-click and select the New File option. You can also choose File -> New File to do the same.



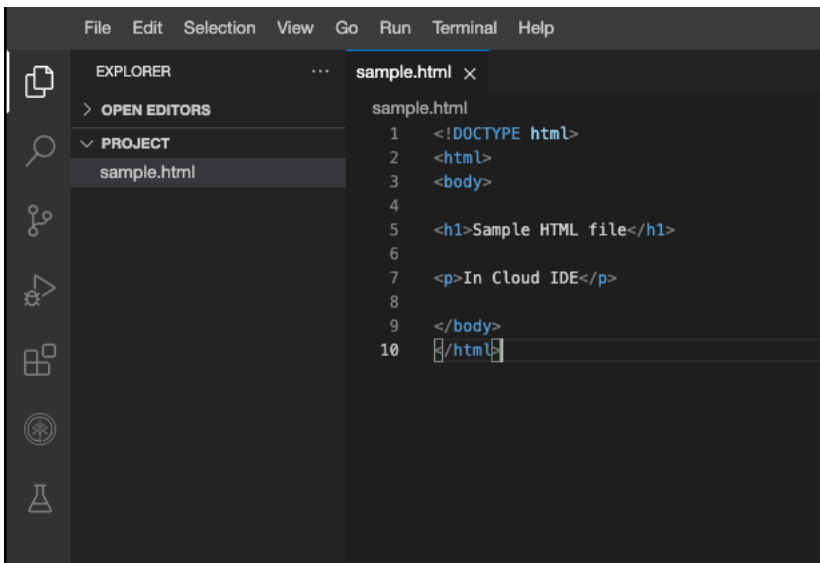
You will then need to name the new file. In this scenario, let's name it sample.html.



Clicking the file name `sample.html` in the directory structure will open the file on the right pane. You can create all different types of files, for example, `FILE_NAME.js` for JavaScript files.

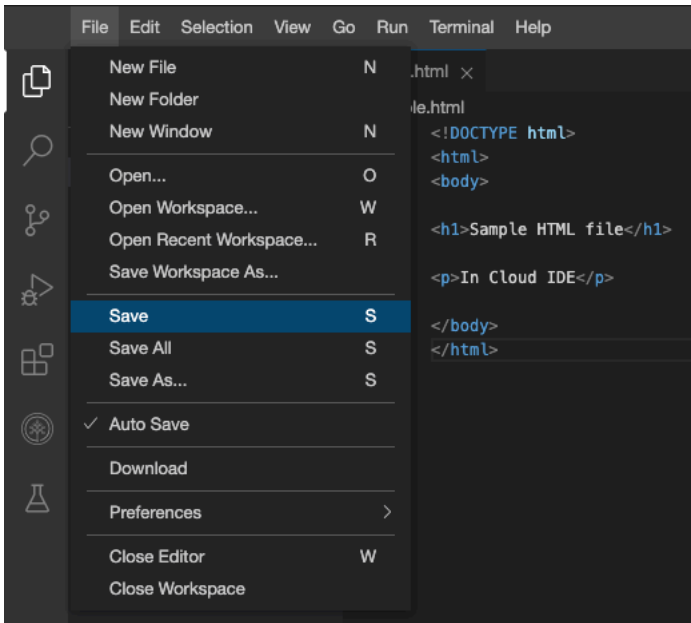


In the example below, we pasted some basic HTML code and then saved the file.



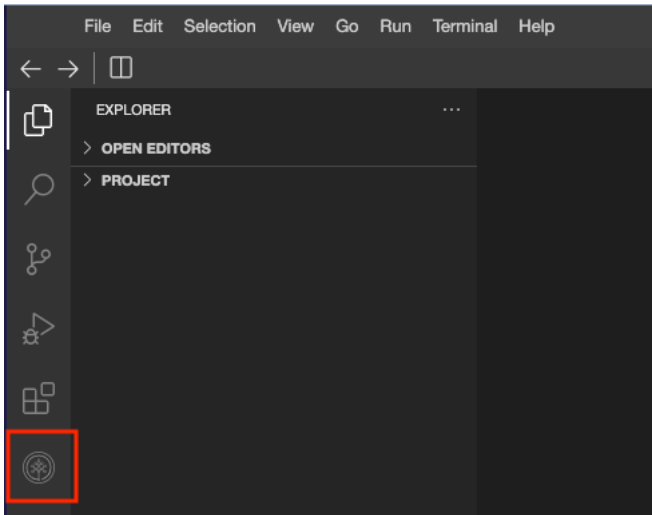
You can save this file in three different ways:

- Select File > Save
- Select Command + S on Mac or CTRL + S on Windows
- Alternatively, it will Autosave your work as well

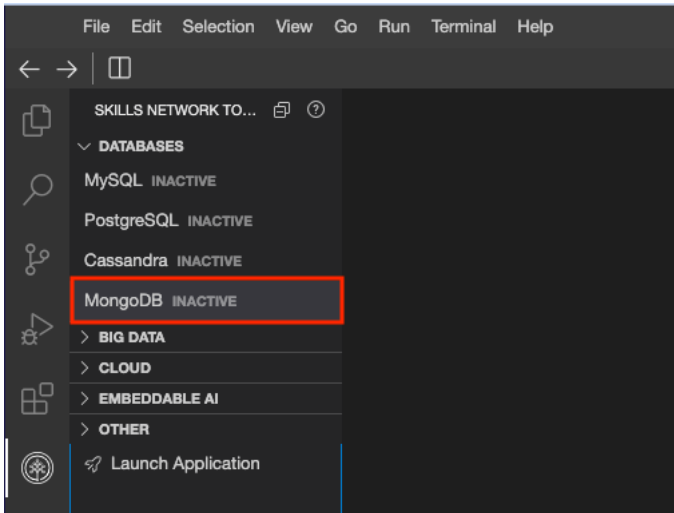


Setup: Start MongoDB

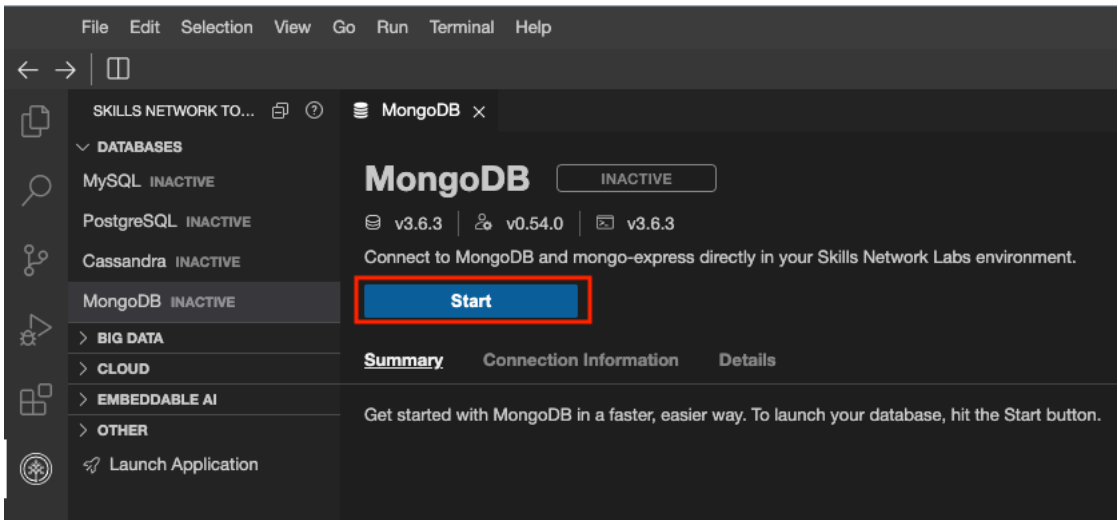
Navigate to Skills Network Toolbox.



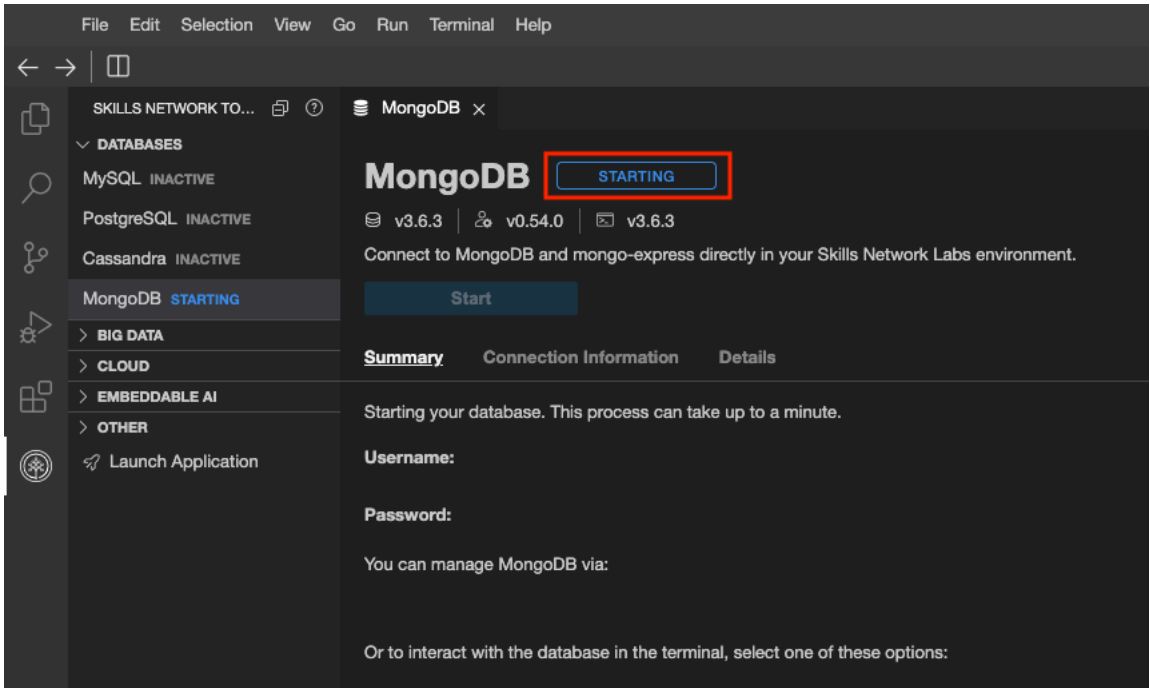
You will notice MongoDB is listed there but inactive. Which means the database is not available for use.



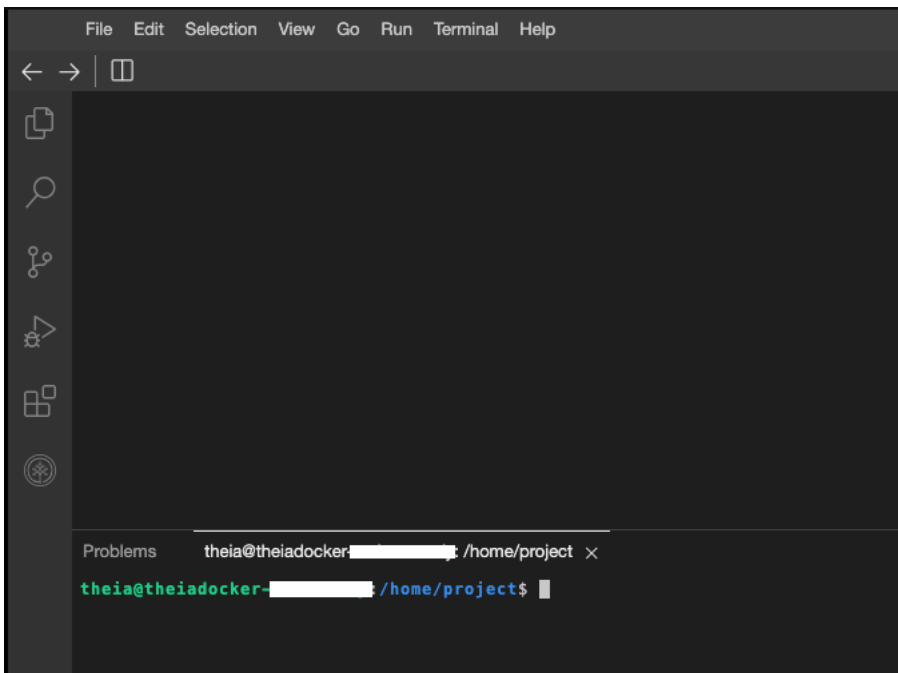
Once you click on it, you will see more details and a button to start it.



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

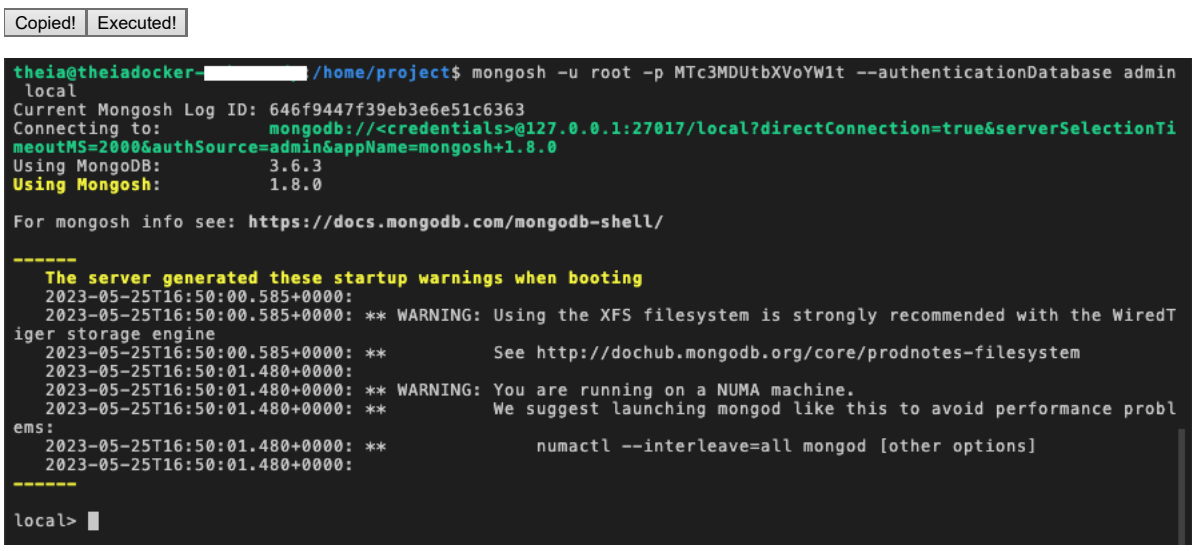


Shortly after that, your server is ready for use. This deployment has access control enabled, and MongoDB enforces authentication. So, take note of the password, as you will need it to log in as the root user.



Run the below command on the newly opened terminal. Copy the code by selecting copy on the right of the code block below and then paste it wherever you wish.

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



The command contains the username and password to connect to the 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 near you. You will need it in the next step.

Or you can click MongoDB CLI, which does that for you.

MongoDB x

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 Det

Username:

Password: MjIzOTETtbXVoYW1t

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 (or Mongo shell), switch the context to the training database.

- 1
1. use training

Copied!

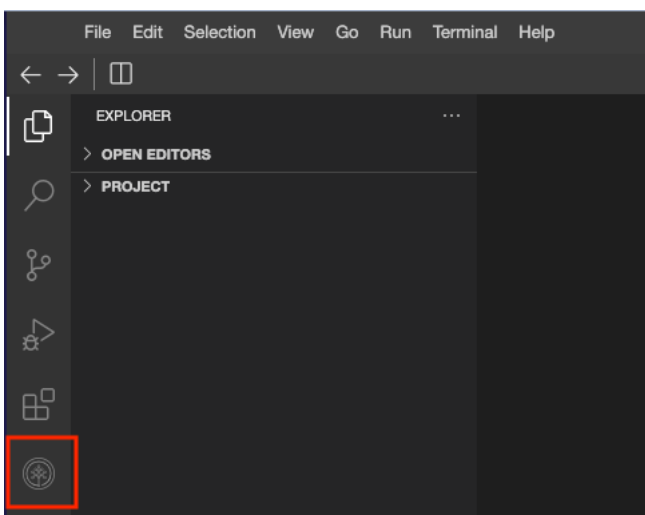
And create a collection called bigdata

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

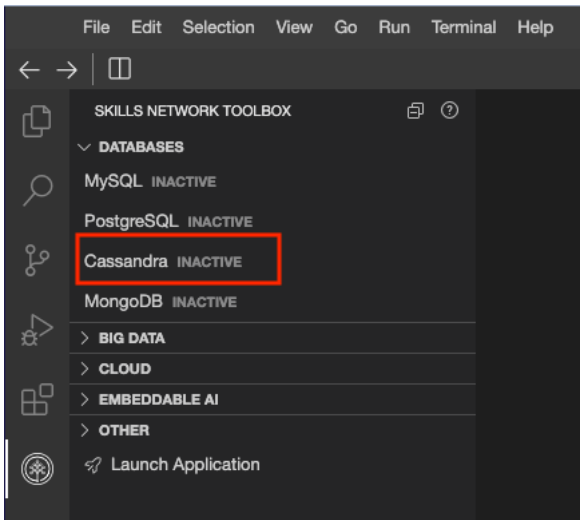
Copied!

Setup: Start Cassandra

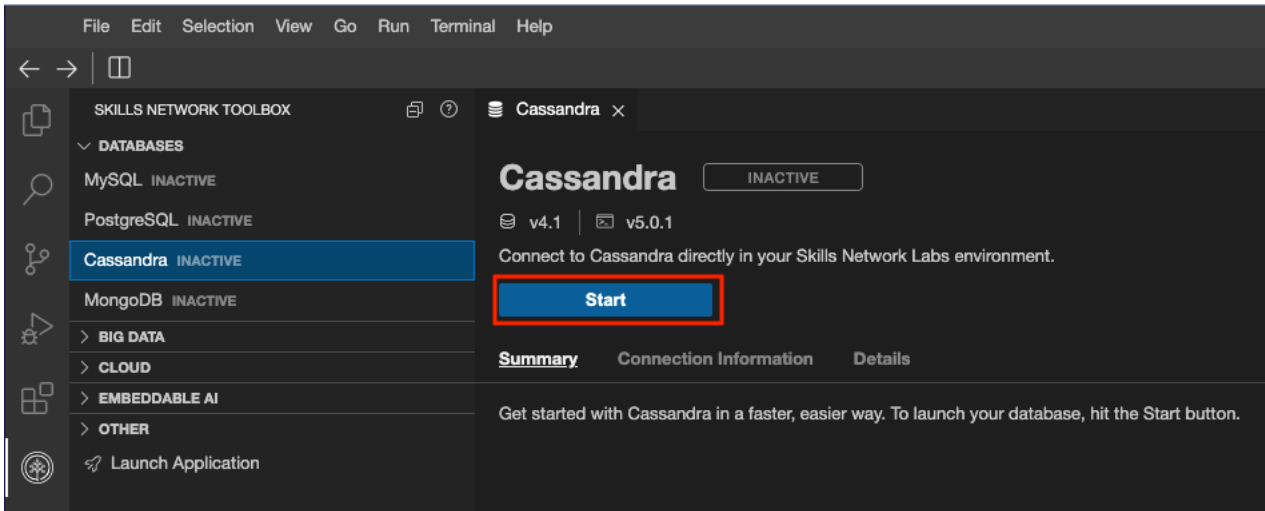
Navigate to Skills Network Toolbox.



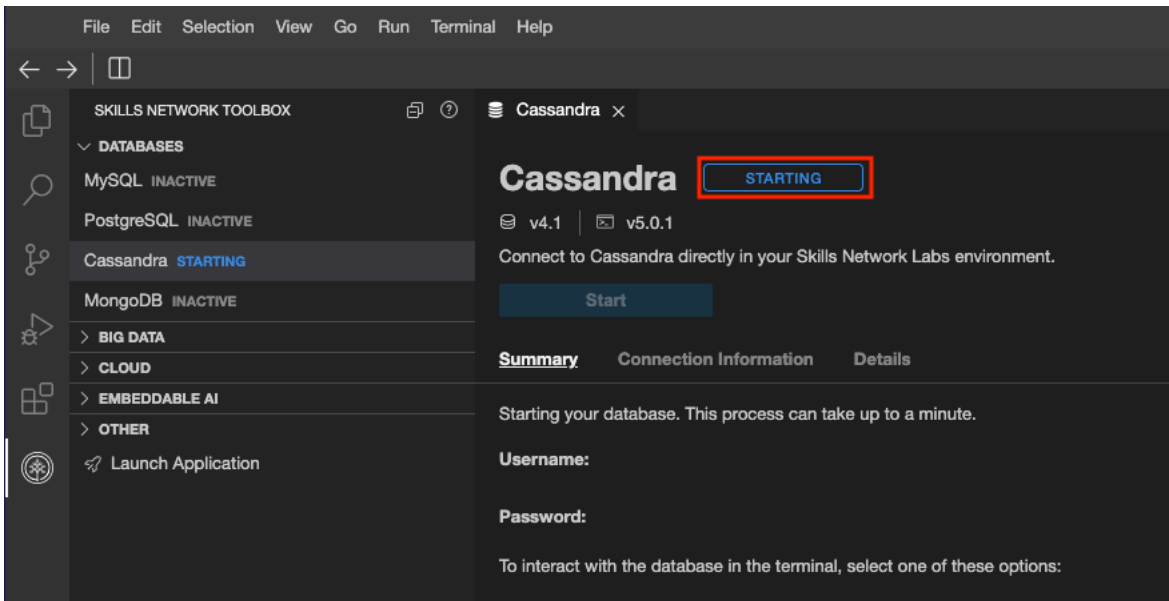
You will notice Cassandra is listed there but inactive. Which means the database is not available for use.



You will see more details once you click on it, and then select **Start**.



Selecting **Start** will run a background process to configure and start your Cassandra server.



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

Cassandra

Cassandra

ACTIVE

v4.1 | v5.0.1

Connect to Cassandra directly in your Skills Network Labs environment.

Stop

Summary | Connection Information | Details

Your database server is now ready to use and available with the following login credentials. For more details on how to navigate Cassandra, please check out the Details section.

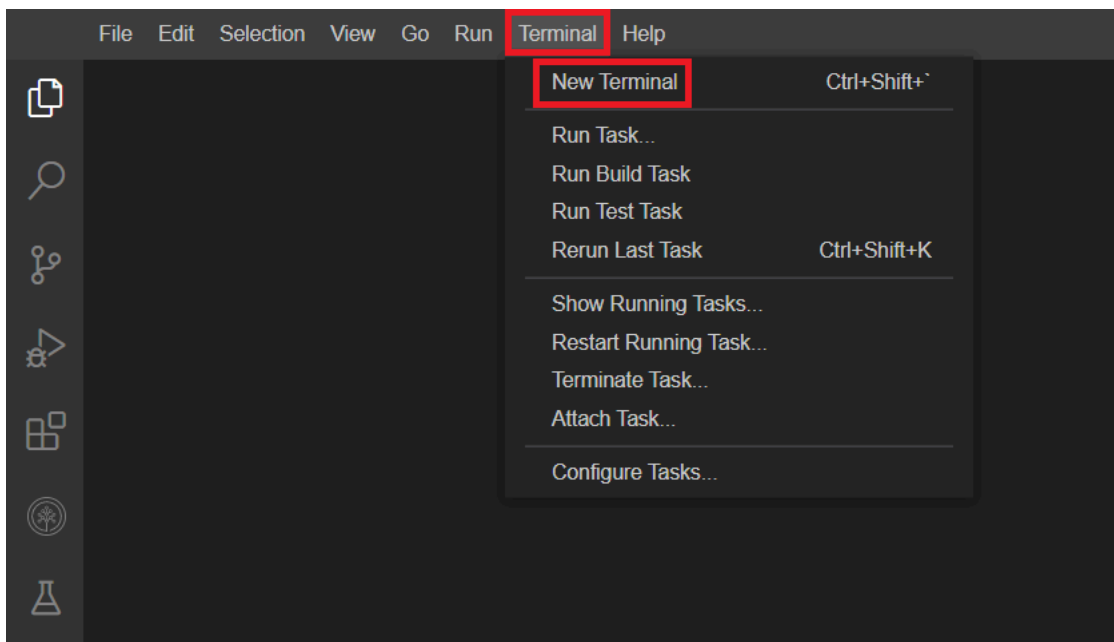
Username:

Password: NTc2Ny1tdWhhbW1h

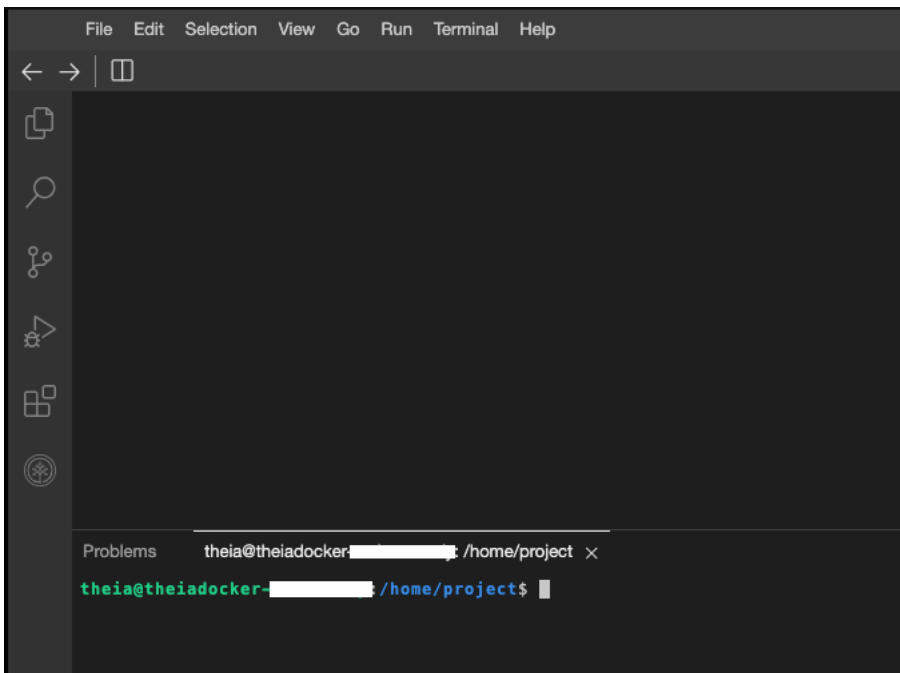
To interact with the database in the terminal, select one of these options:

Cassandra CLI | New Terminal

You can now open the terminal and enter details yourself.



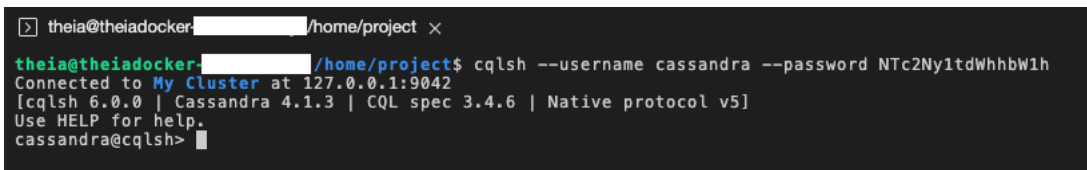
This action will open a new terminal at the bottom of the screen, as in the image below.



Run the following command on the newly opened terminal. Copy the code by selecting copy on the right of the code block below and then paste it wherever you wish.

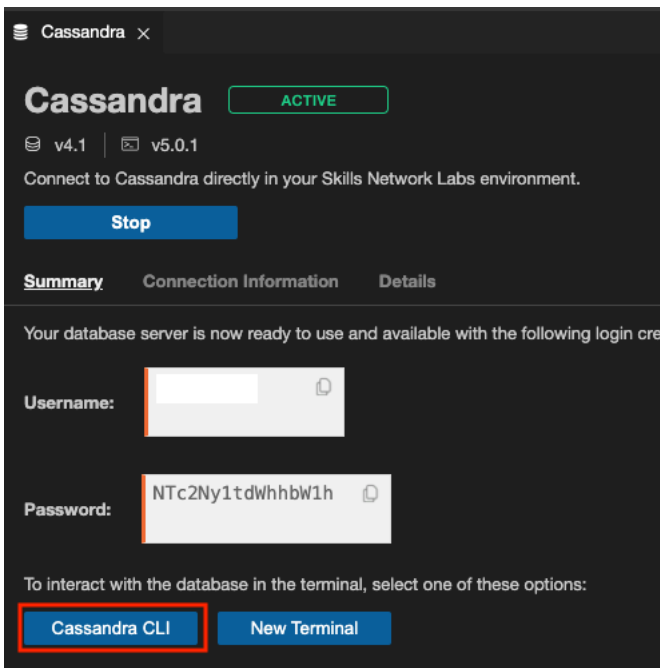
- 1
1. `cqlsh --username cassandra --password PASSWORD`

Copied! Executed!



The command contains the username and password to connect to the Cassandra server. Your output could 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 Cassandra CLI, which does that for you.



Exercise 1: Create sample data

Create diamonds.json

First, You will create a diamonds.json file with the following content to import in later exercises.

Open diamonds.json in IDE

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10

1. { "carat": 0.31, "cut": "Ideal", "color": "J", "clarity": "SI2", "depth": 62.2, "table": 54, "price": 339 }
2. { "carat": 0.2, "cut": "Premium", "color": "E", "clarity": "SI2", "depth": 60.2, "table": 62, "price": 351 }
3. { "carat": 0.32, "cut": "Premium", "color": "E", "clarity": "I1", "depth": 60.9, "table": 58, "price": 342 }
4. { "carat": 0.3, "cut": "Good", "color": "J", "clarity": "SI1", "depth": 63.4, "table": 54, "price": 349 }
5. { "carat": 0.3, "cut": "Good", "color": "J", "clarity": "SI1", "depth": 63.8, "table": 56, "price": 347 }
6. { "carat": 0.3, "cut": "Very Good", "color": "J", "clarity": "SI1", "depth": 62.7, "table": 59, "price": 349 }
7. { "carat": 0.3, "cut": "Good", "color": "I", "clarity": "SI2", "depth": 63.3, "table": 56, "price": 343 }
8. { "carat": 0.23, "cut": "Very Good", "color": "E", "clarity": "VS2", "depth": 63.8, "table": 55, "price": 339 }
9. { "carat": 0.23, "cut": "Very Good", "color": "H", "clarity": "VS1", "depth": 61, "table": 57, "price": 323 }
10. { "carat": 0.31, "cut": "Very Good", "color": "J", "clarity": "SI1", "depth": 59.4, "table": 62, "price": 346 }
```

Copied!

Exercise 2: MongoDB import/export data

JSON

Import data in diamonds.json into a collection named diamonds and a database named training, replacing the password (the characters following -p: MzA2NDAtcnNhbm5h) with your password.

```
1. 1

1. mongoimport -u root -p MzA2NDAtcnNhbm5h --authenticationDatabase admin --db training --collection diamonds --file /home/project/diamonds.json
```

Copied! Executed!

Log in to mongoDB and checkfor the creation of the training database and the diamonds collection and the collection has the imported documents.

Export data into json format.

Export data from the training database, diamonds collection into a file named mongodb_exported_data.json

```
1. 1

1. mongoexport -u root -p MzA2NDAtcnNhbm5h --authenticationDatabase admin --db training --collection diamonds --out /home/project/mongodb_exported_data.
```

Copied! Executed!

Verify the output by opening the file.

Open mongodb_exported_data.json in IDE

CSV

Export data into CSV format.

Export the fields _id,clarity,cut,price from the training database, diamonds collection into a file named mongodb_exported_data.csv

```
1. 1

1. mongoexport -u root -p MzA2NDAtcnNhbm5h --authenticationDatabase admin --db training --collection diamonds --out /home/project/mongodb_exported_data.
```

Copied! Executed!

Exercise 3: Cassandra import/export data

Import CSV into Cassandra

Import diamonds.csv into the training keyspace and the diamonds table/column family.

Step 1: Log in to cqlsh.

Step 2: Create a keyspace named training.

- ▶ Click here for hint
- ▶ Click here for solution

Step 3: In the training keyspace, create a table named `diamonds` with the below schema.

- `id` - primary key (use `'id'` as the primary key; Cassandra does not allow you to create a column starting with underscore(`_`))
- `clarity` - text
- `cut` - text
- `price` - integer

► [Click here for hint](#)

► [Click here for solution](#)

Step 4: Run the below commands on `cqlsh`.

```
1. 1
2. 2
```

```
1. use training;
2. COPY training.diamonds(id,clarity,cut,price) FROM '/home/project/mongodb_exported_data.csv' WITH DELIMITER=';' AND HEADER=TRUE;
```

Copied!

Export the `diamonds` table into a CSV file.

```
1. 1
```

```
1. COPY diamonds TO '/home/project/cassandra-diamonds.csv';
```

Copied!

Verify the output by opening the file

Open `cassandra-diamonds.csv` in IDE

Exercise 4: Creating an index on Cassandra

The following command creates a `price_index` for the `price` column in the `diamonds` table.

```
1. 1
```

```
1. create index price_index on diamonds(price);
```

Copied!

Summary

In this lab, you have practiced working on importing and exporting data from MongoDB and Cassandra. You also created an index in Cassandra using `cqlsh`.

Author(s)

- [Muhammad Yahya](#)

© IBM Corporation. All rights reserved.