

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.

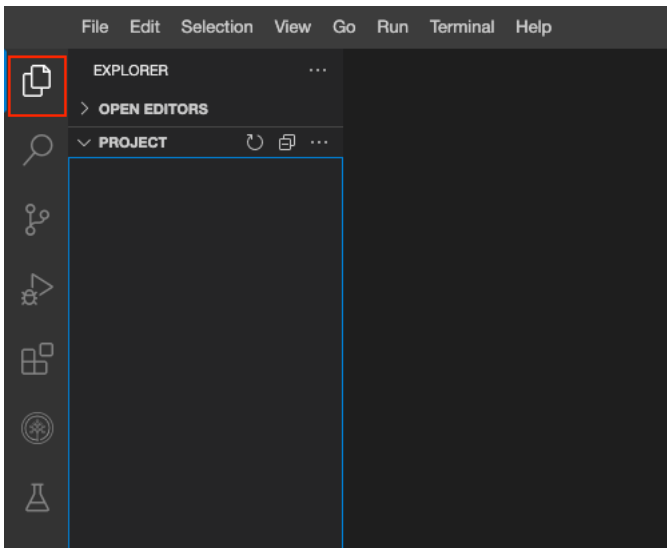
Important notice about this lab environment

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.

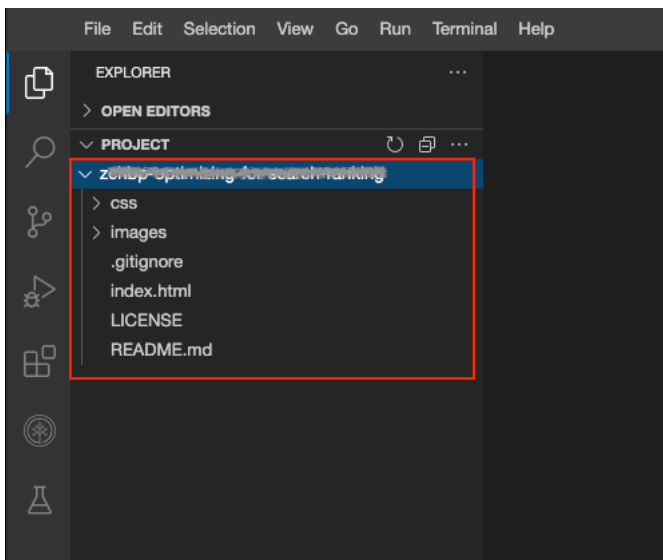
Working with Files in Cloud IDE

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

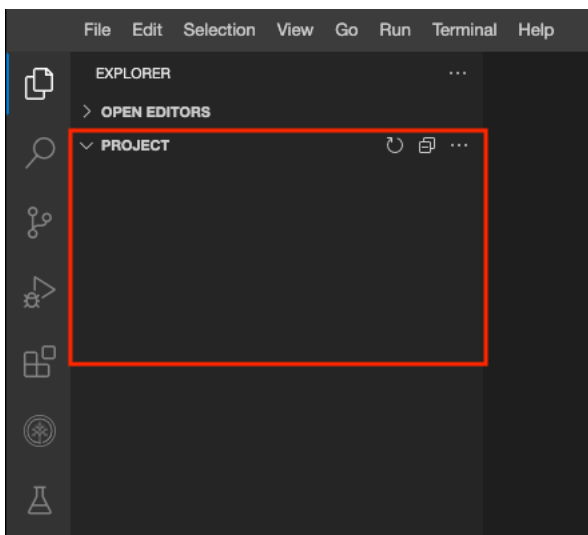
To view your files and directories in Cloud IDE, click the file icon to reveal it.



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

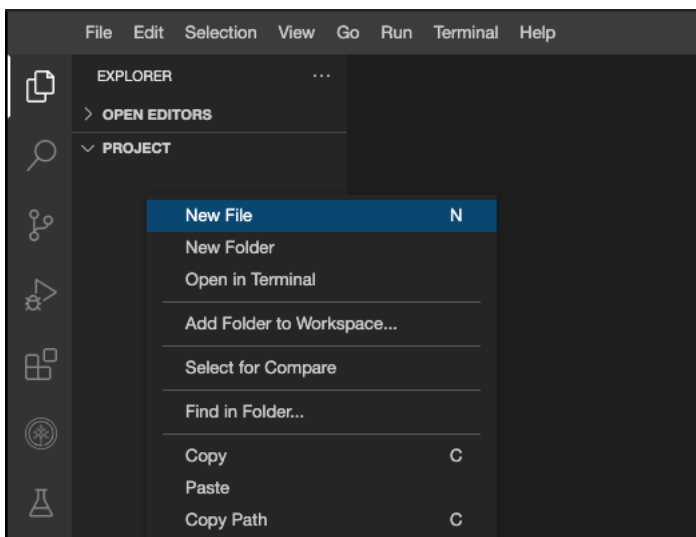


If you have not cloned and are starting with a blank project, it will look like this:

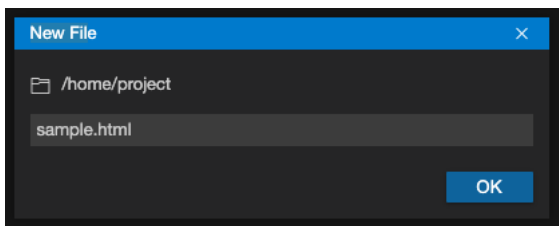


Create a new file

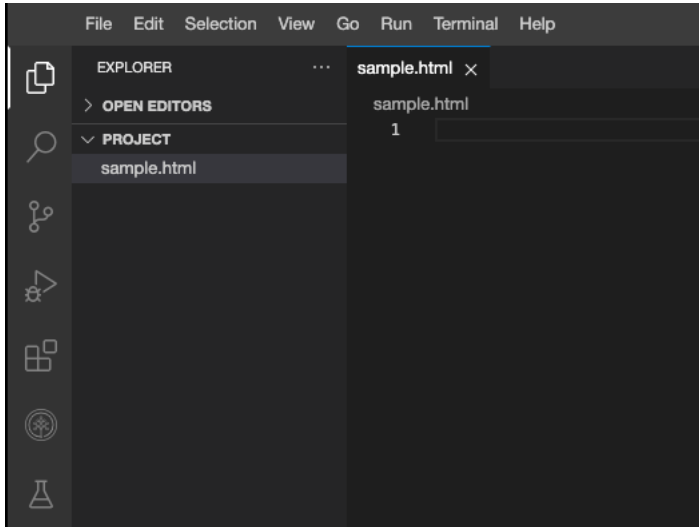
To create a new file in your project, right-click and select the New File option. You can also choose File -> New File to do the same.



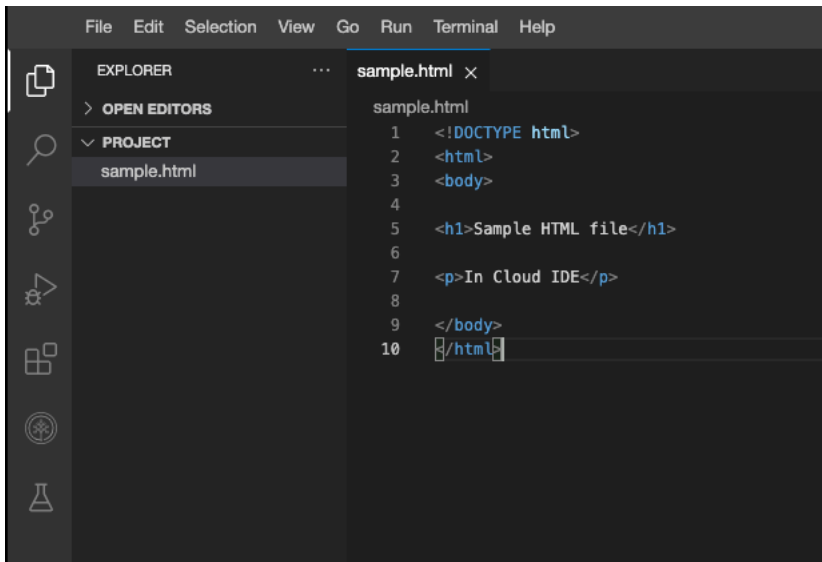
You will then need to name the new file. In this scenario, let's name it `sample.html`.



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.

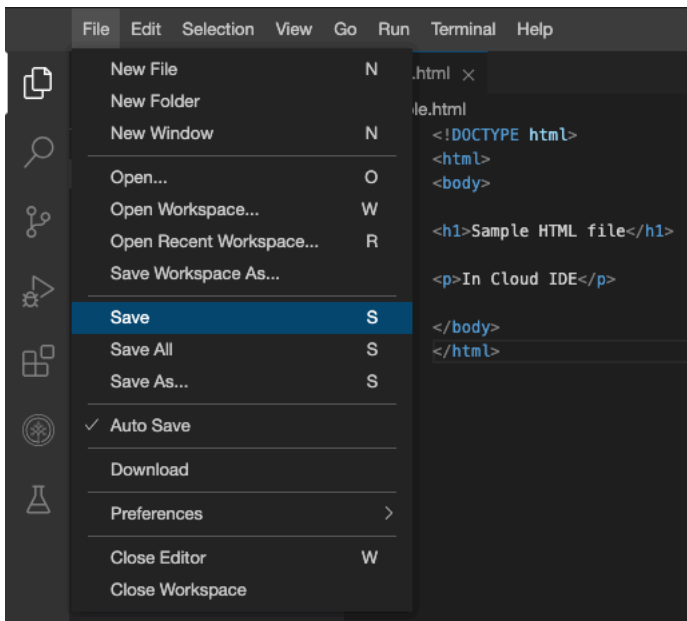


In the example below, we pasted some basic HTML code and then saved the file.



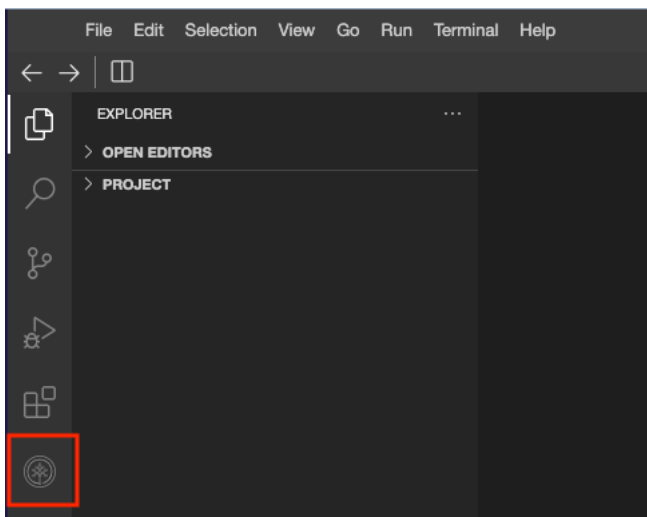
You can save this file in three different ways:

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

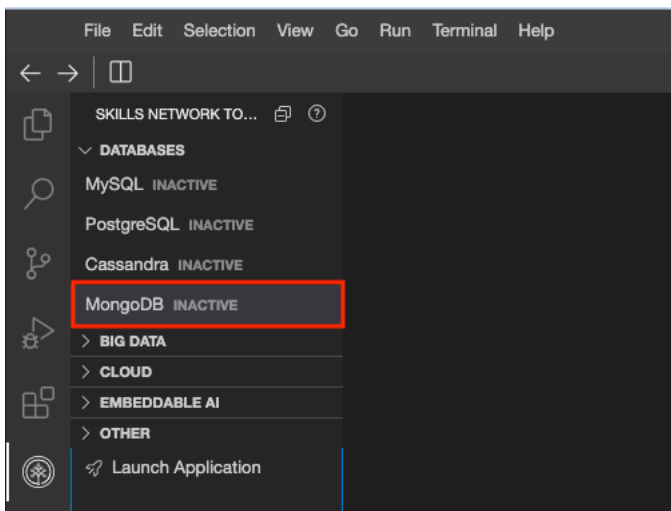


Setup: Start MongoDB

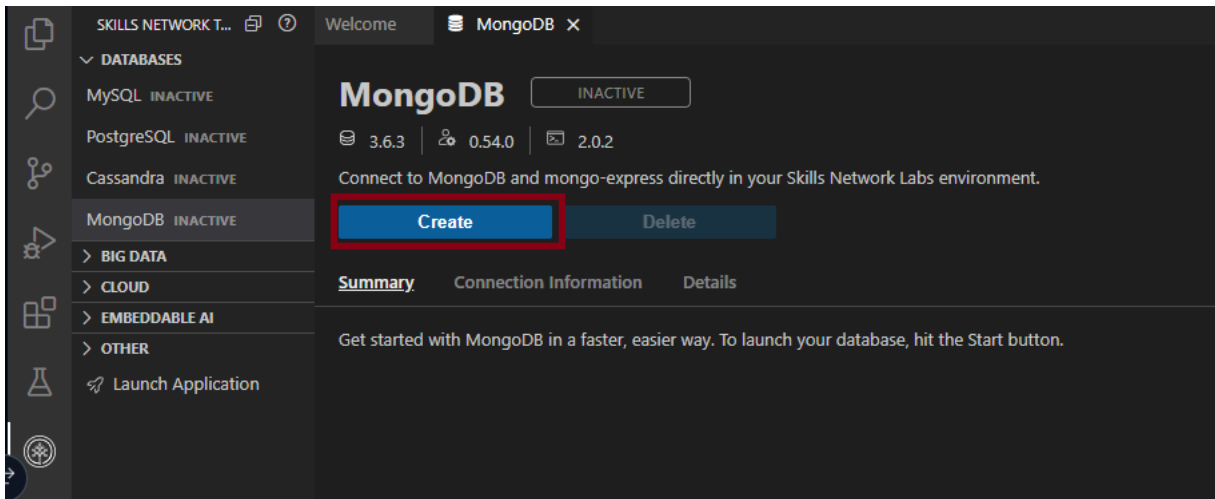
Navigate to Skills Network Toolbox.



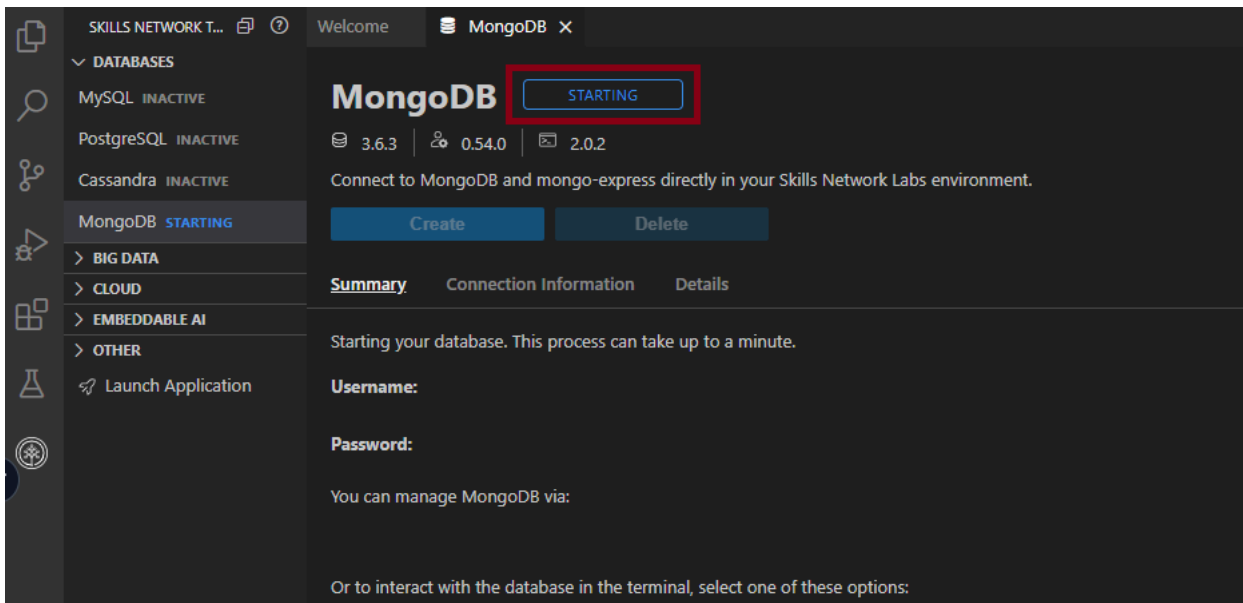
You will notice MongoDB is listed there but inactive. Which means the database is not available for use.



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



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



Soon, your server is ready for use. This deployment has access control enabled and MongoDB enforces authentication. So, take note of the password. You will need this password to login as root user.

Note: For Password and other information click on Connection Information

MongoDB

ACTIVE

3.6.3 | 0.54.0 | 2.0.2

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

CreateDelete

SummaryConnection InformationDetails

MONGO_USERNAME:root

MONGO_HOST:172.21.41.238

MONGO_PORT:27017

URL:https://labs-mongo-thundering-most-fall.mongo.databases.labs.skills.network

MONGO_URL:https://labs-mongo-thundering-most-fall.mongo.databases.labs.skills.network

MongoDB CLI Command: mongosh mongodb://root:aPmca0TKv1vxBt1ANcooACHH@172.21.41.238:27017

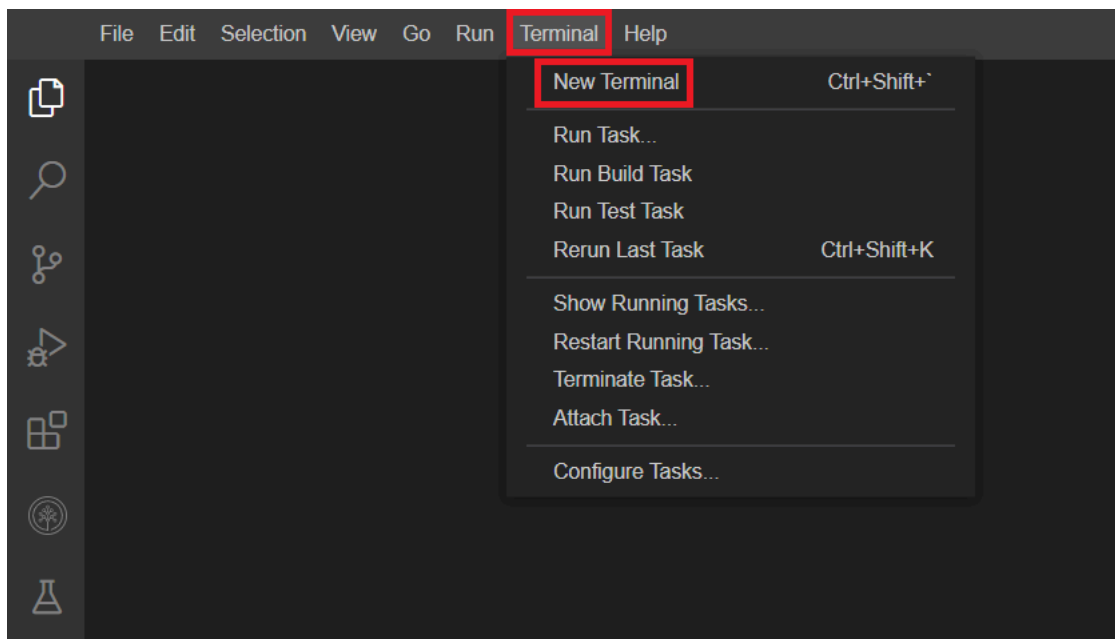
MONGO_COMMAND:mongosh mongodb://root:aPmca0TKv1vxBt1ANcooACHH@172.21.41.238:27017

MONGO_PASSWORD:aPmca0TKv1vxBt1ANcooACHH

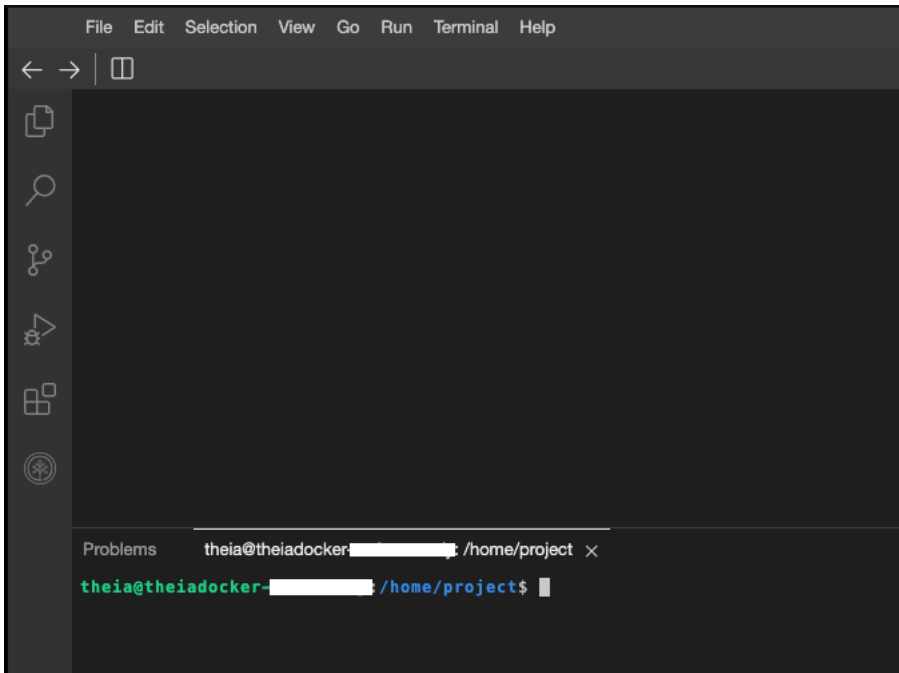
MONGO_TITLE:MongoDB Database

MONGO_ID:labs-mongo-thundering-most-fall

You can now open the terminal



This action will open a new terminal at the end of the screen, like the following image.



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.

Note: Replace the PASSWORD with the MONGO_PASSWORD

```
mongosh -u root -p PASSWORD --authenticationDatabase admin local --host mongo
```

```
theia@theiadocker-nikeshkr:/home/project$ mongosh -u root -p aPmca0TKv1vxBt1ANcooACHH --
authenticationDatabase admin local --host mongo
Current Mongosh Log ID: 66de8a0693cc32c77c5e739b
Connecting to:      mongodb://<credentials>@mongo:27017/local?directConnection=true&
authSource=admin&appName=mongosh+2.3.0
Using MongoDB:      3.6.3
Using Mongosh:      2.3.0

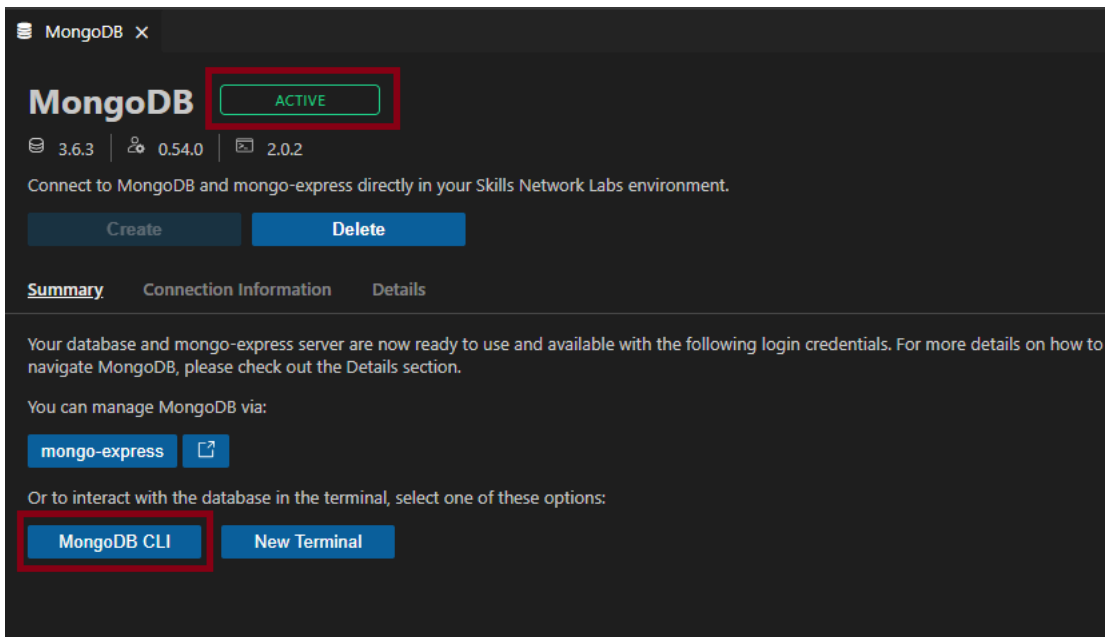
For mongosh info see: https://www.mongodb.com/docs/mongosh-shell/

-----
The server generated these startup warnings when booting
2024-09-09T04:08:06.675+0000:
2024-09-09T04:08:06.675+0000: ** WARNING: Using the XFS filesystem is strongly recomm
ended with the WiredTiger storage engine
2024-09-09T04:08:06.675+0000: **           See http://dochub.mongodb.org/core/prodnote
s-filesystem
-----

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.



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

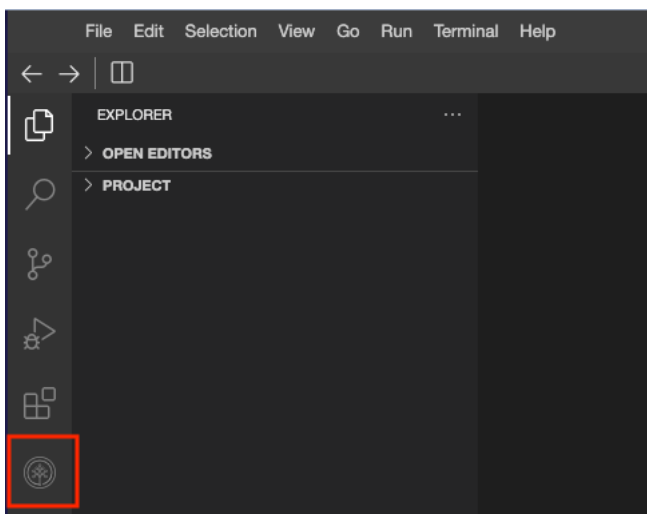
```
use training
```

And create a collection called bigdata

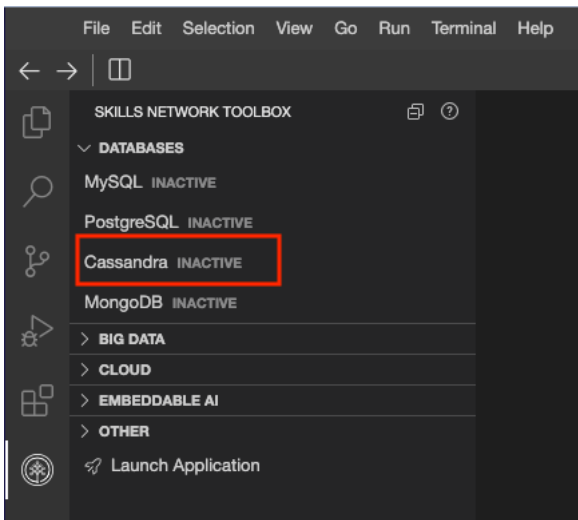
```
db.createCollection("bigdata")
```

Setup: Start Cassandra

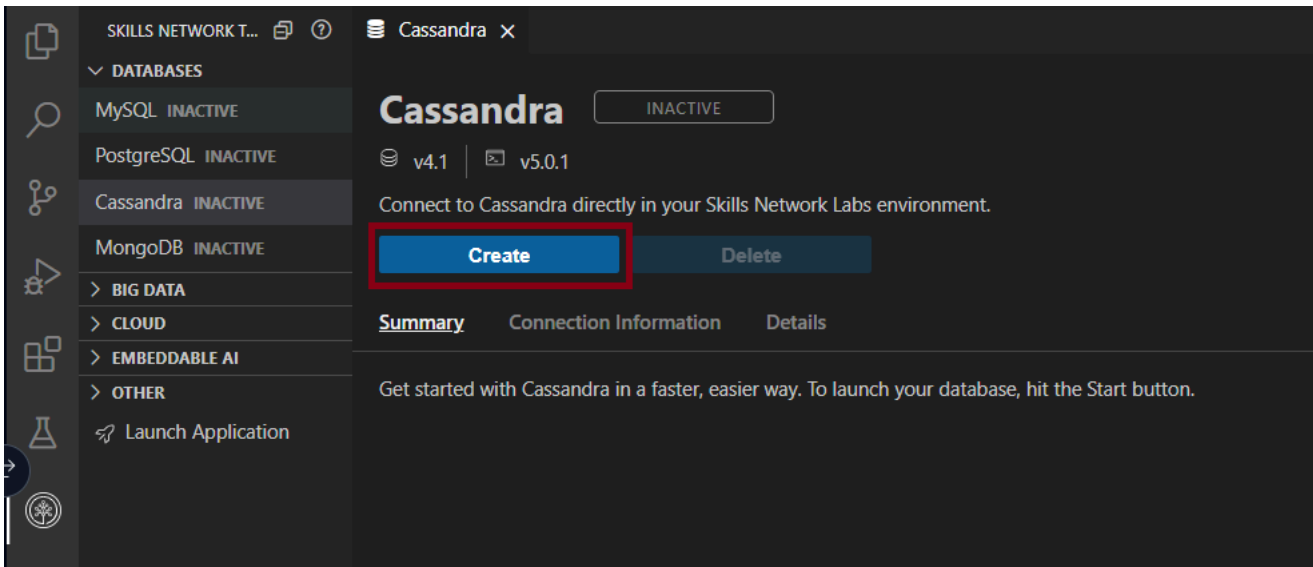
Navigate to Skills Network Toolbox.



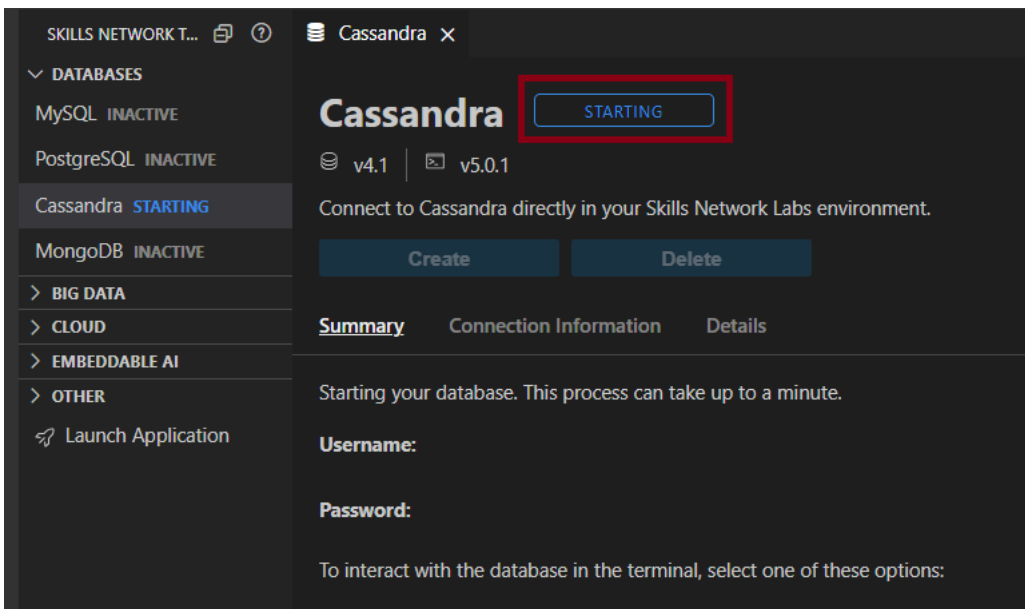
You will notice Cassandra is listed there but inactive. Which means the database is not available for use.



Once you click on Cassandra, you will see more details and the button to Create.



Clicking the Create button runs 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. Click on the Connection Information tab take note of the Cassandra CLI Command as you will need to login as a root user.

Cassandra ACTIVE

v4.1 | v5.0.1

Connect to Cassandra directly in your Skills Network Labs environment.

Create Delete

Summary **Connection Information** Details

Username: nikeshkr

Password: MTkyMS1uaWt1c2hr

Host: 127.0.0.1

Port: 9042

Cassandra CLI Command:

```
cqlsh 172.21.151.104 9042 --username root --password 8Vo0KNCYFggUh59uv1Lnz1VD
```

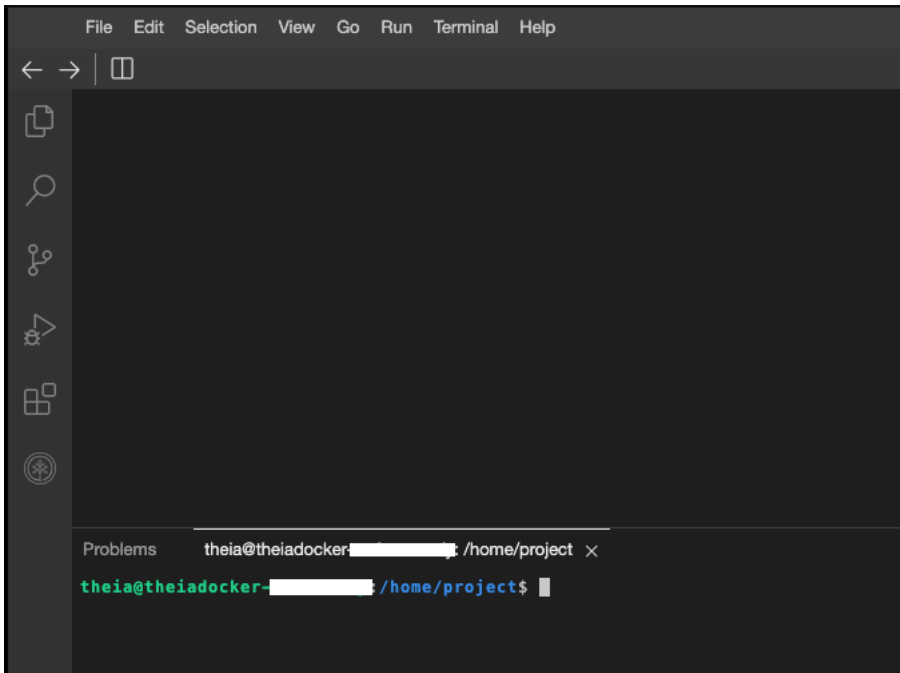
You can now open the terminal.

File Edit Selection View Go Run **Terminal** Help

New Terminal Ctrl+Shift+`

- Run Task...
- Run Build Task
- Run Test Task
- Rerun Last Task Ctrl+Shift+K
- Show Running Tasks...
- Restart Running Task...
- Terminate Task...
- Attach Task...
- Configure Tasks...

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

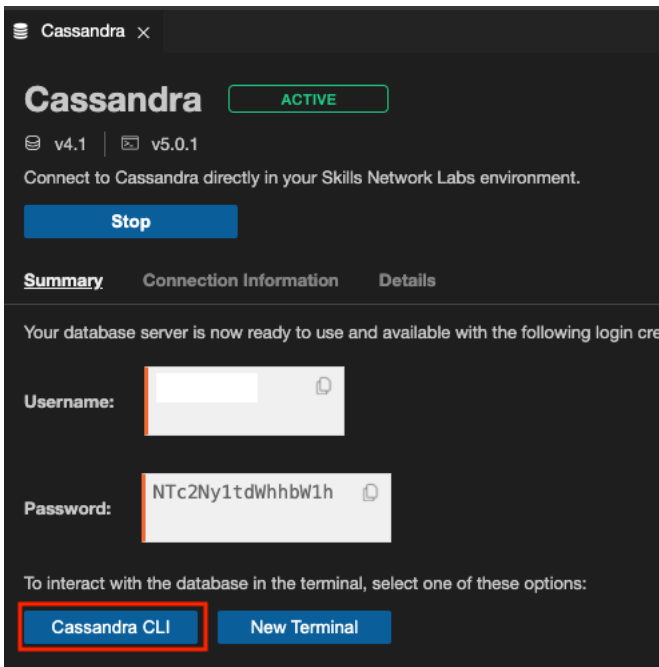


Run the Cassandra CLI Command on the newly opened terminal.

```
theia@theiadocker-nikeshkr:/home/project$ cqlsh 172.21.151.104 9042 --username root --password 8Vo0KNCYFggUh59uvlLnz1VD
WARNING: cqlsh was built against 4.0.13, but this server is 5.0. All features may not work!
Connected to Test Cluster at 172.21.151.104:9042
[cqlsh 6.0.0 | Cassandra 5.0-beta1 | CQL spec 3.4.7 | Native protocol v5]
Use HELP for help.
root@cqlsh>
```

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](#)

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

Exercise 2: MongoDB import/export data

JSON

Import data in diamonds.json into a collection named diamonds and a database named training.

Replace the password (the characters following **-p**: MzA2NDAtcnNhbm5h) with your Mongo password.

Open a new terminal and run the following commands:

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

Log in to mongoDB and check for 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

Replace the password (the characters following **-p**: MzA2NDAtcnNhbm5h) with your Mongo password.

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

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

Replace the password (the characters following **-p**: MzA2NDAtcnNhbm5h) with your Mongo password.

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

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

```
CREATE KEYSPACE training
WITH replication = {'class':'SimpleStrategy', 'replication_factor' : 3};
```

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

- id - primary key (use 'id' as the primary key (type-varchar); 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

```
use training;
CREATE TABLE diamonds(
    id varchar PRIMARY KEY,
    clarity text,
    cut text,
    price int
);
```

Step 4: Run the below commands on cqlsh.

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

Export the diamonds table into a CSV file.

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

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.

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

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.