Accessing MongoDB using Python

Estimated time needed: **30** minutes

Objectives

After completing this lab, you will be able to:

- Access the MongoDB database from Python with the Pymongo driver
- Perform basic operations such as selecting, inserting, and listing using Python
- Create a Python program to run the MongoDB operations

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) for a desktop or cloud. To complete this lab, you will use the Cloud IDE based on Theia and MongoDB running in a Docker container.

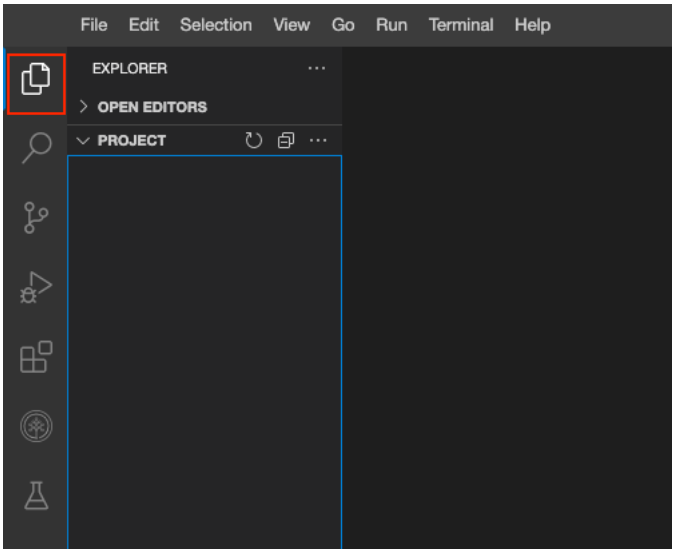
Important notice about this lab environment

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.

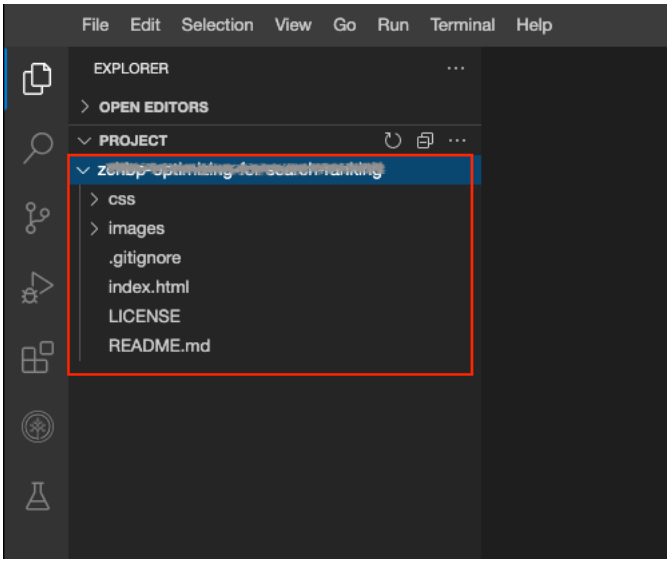
Working with files in Cloud IDE

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

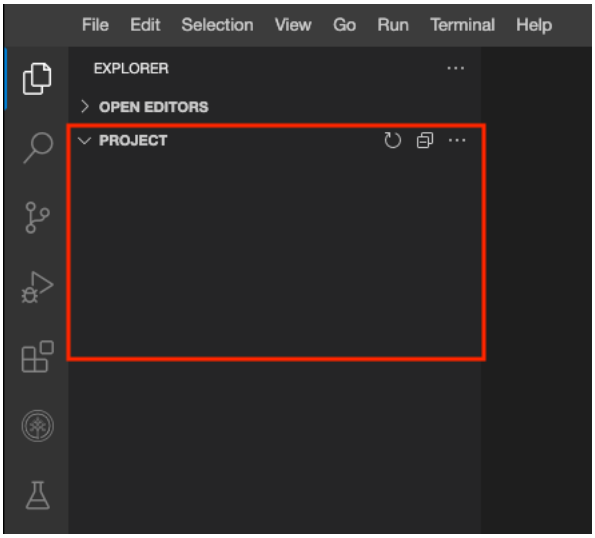
To view your files and directories inside Cloud IDE, click the file's icon.



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

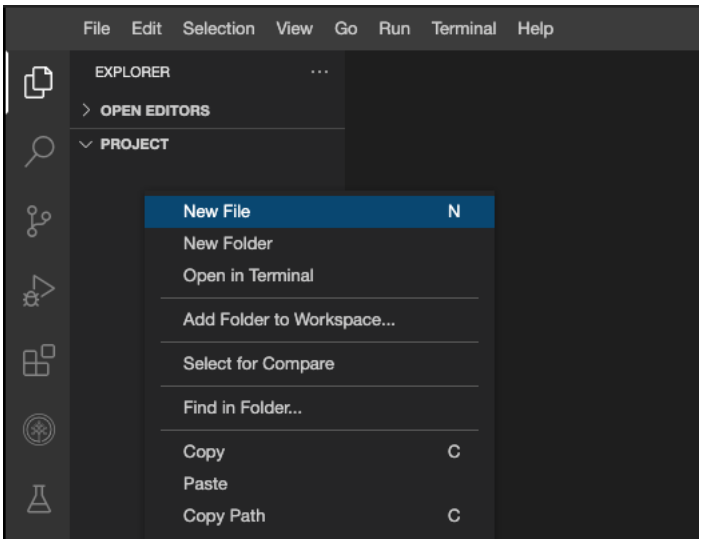


Otherwise, a blank project looks like this:



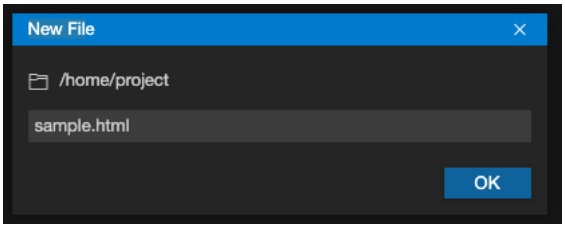
Create a new file

You can right-click and select New File to create a file in your project.

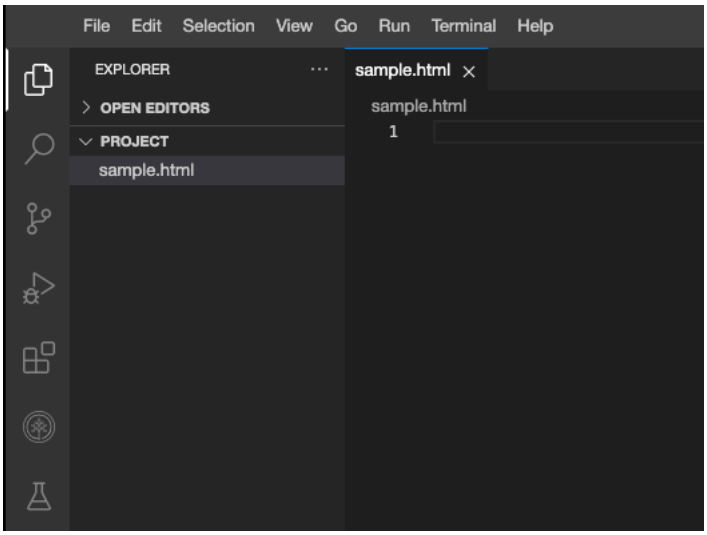


You can also choose `File -> New File` to do the same.

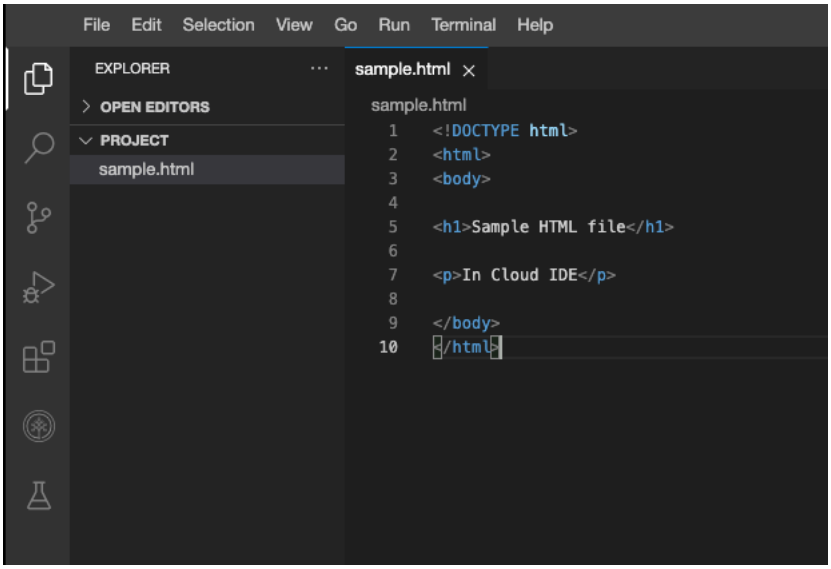
It will then prompt you to enter the name of this new file. In the example below, the new file name is `sample.html`.



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

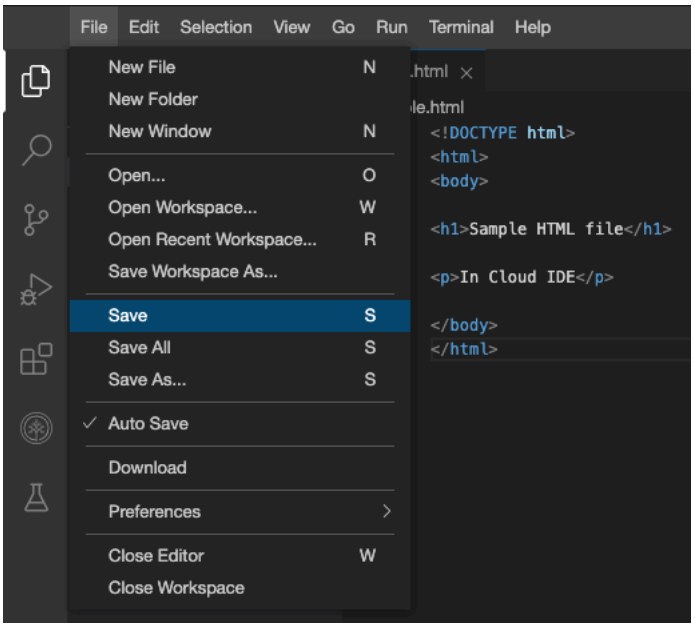


In the following image, we pasted some basic HTML code and then saved the file.



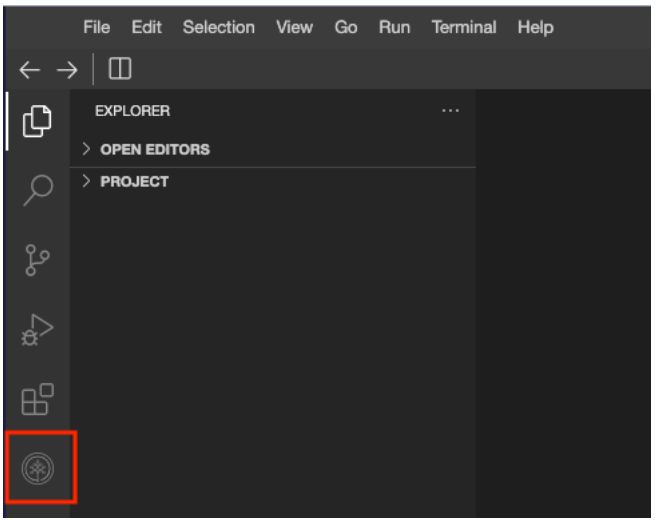
You can save a file with one of the following methods:

- Using the menu
- Press ⌘ + S on Mac or CTRL + S on Windows
- Or it can Autosave it for you, too

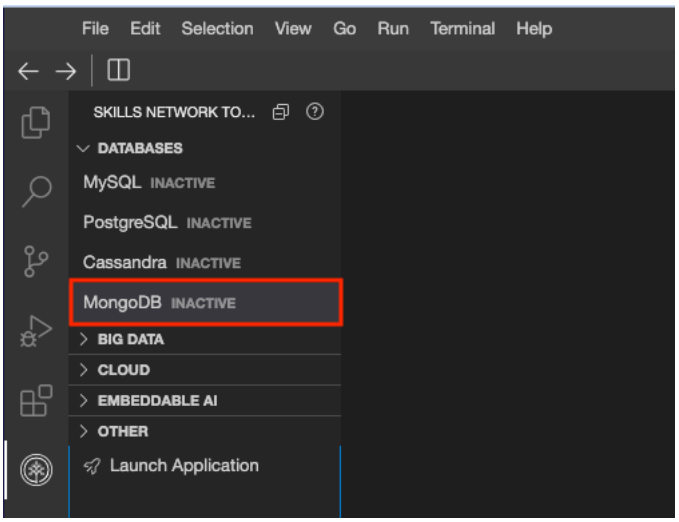


Set-up: Start MongoDB

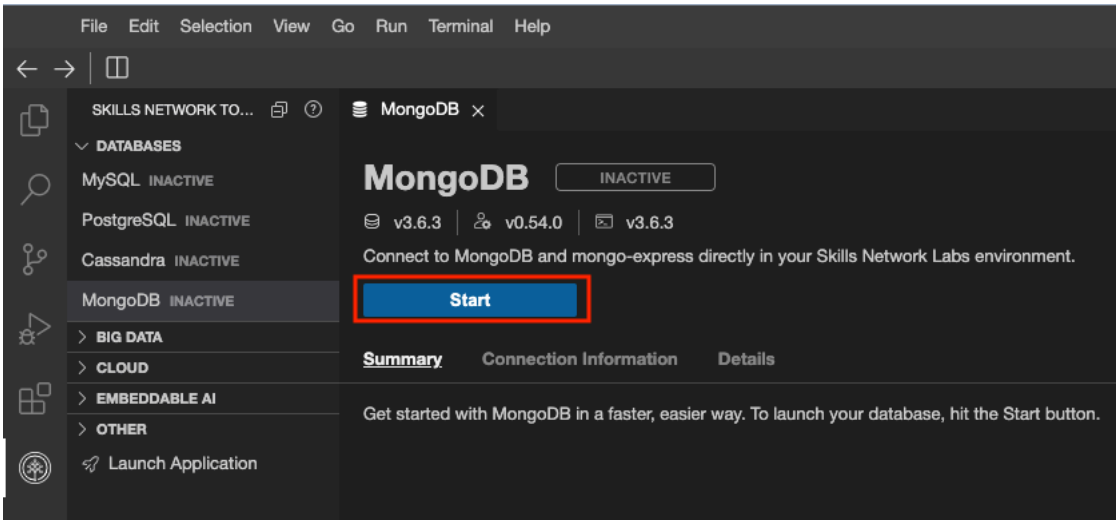
Navigate to Skills Network Toolbox.



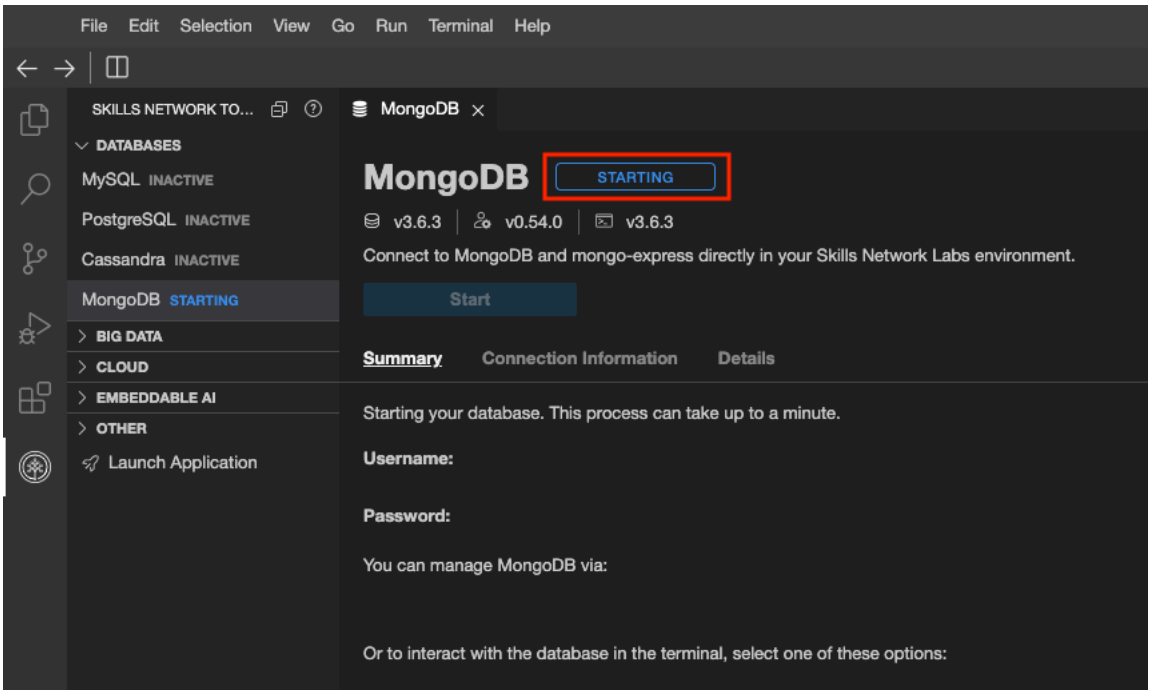
You will notice MongoDB is listed there but inactive.



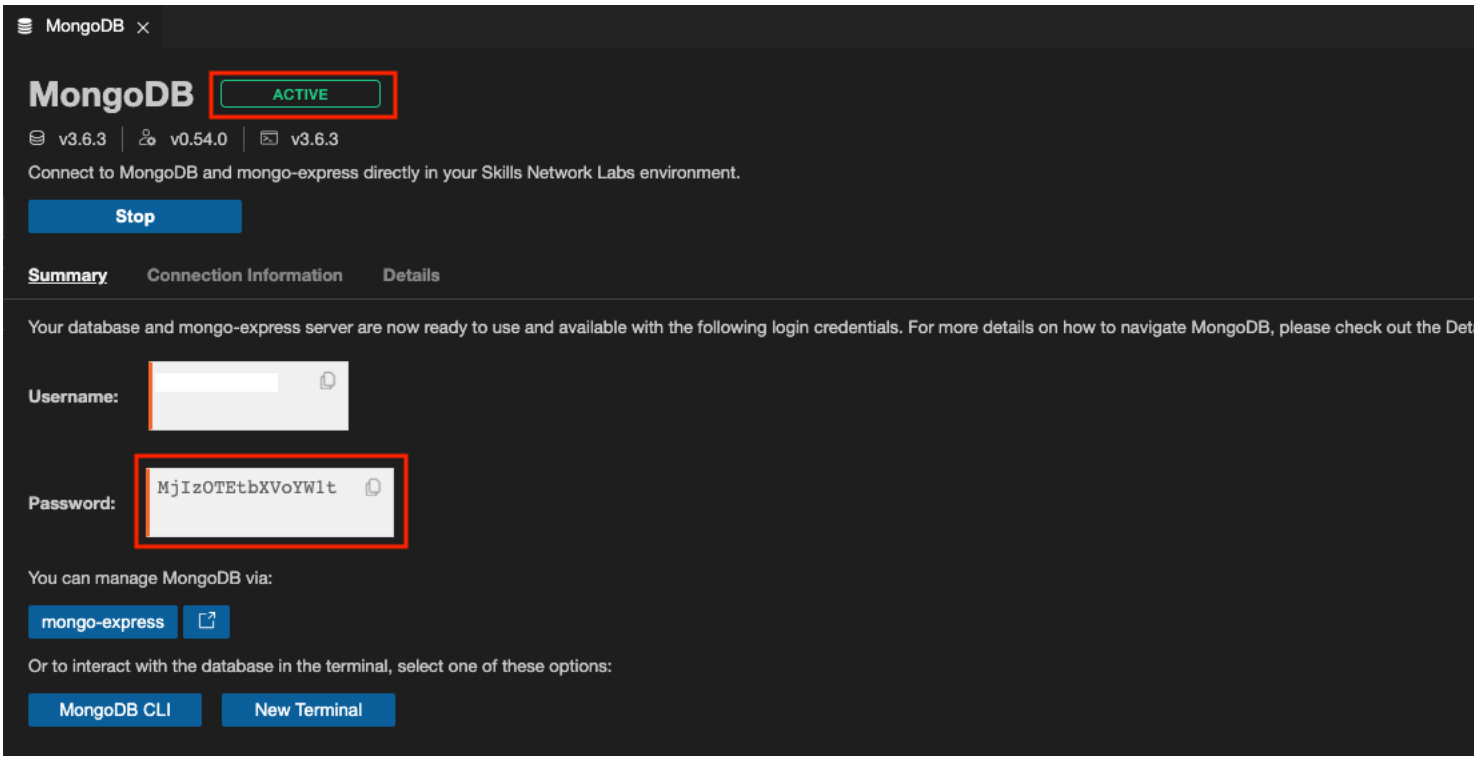
Once you click on it, you will see more details and an icon to start it.



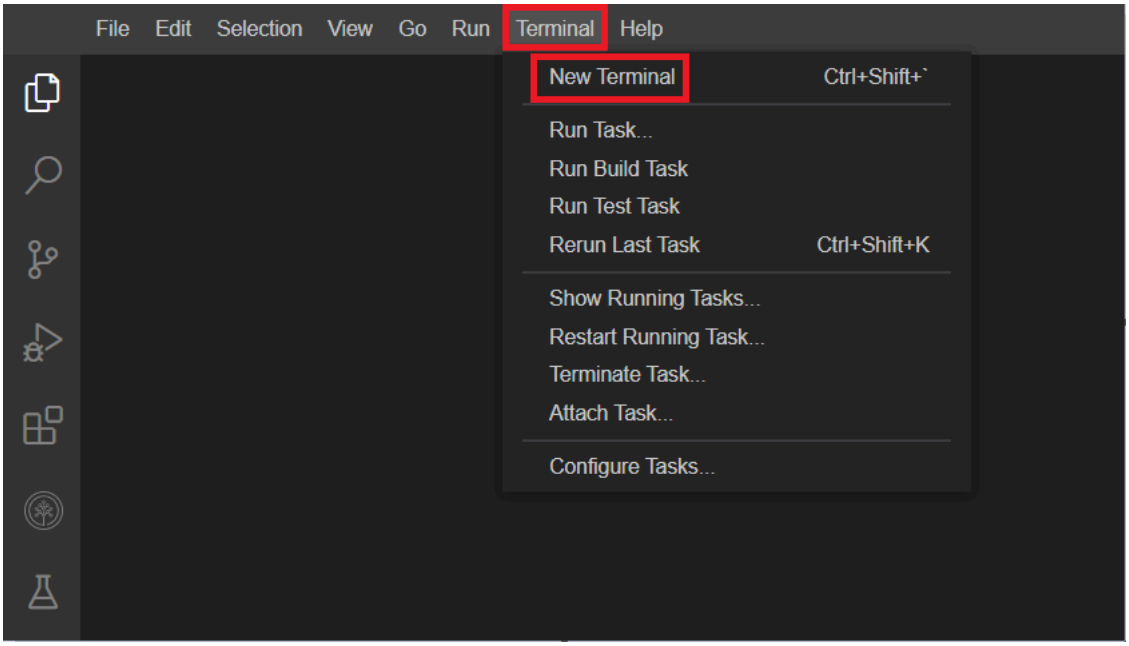
Click Start to run a background process to configure and start your MongoDB server.



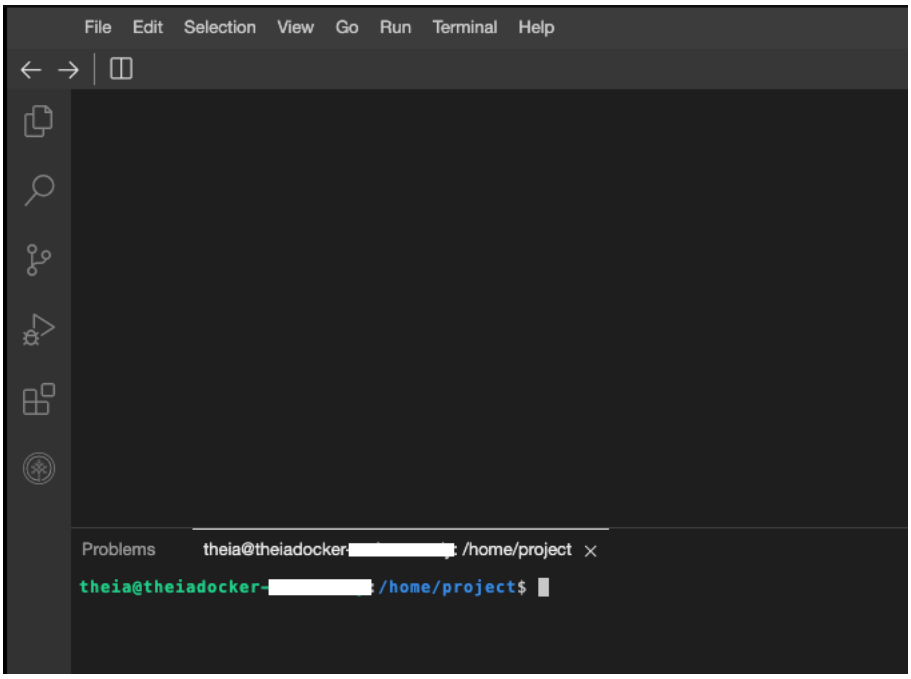
Now, your server is ready for use. This deployment has access control enabled, and MongoDB enforces authentication. So, take note of the password, as you will need it to log in as the `root` user.



You can now open the terminal and enter details yourself.

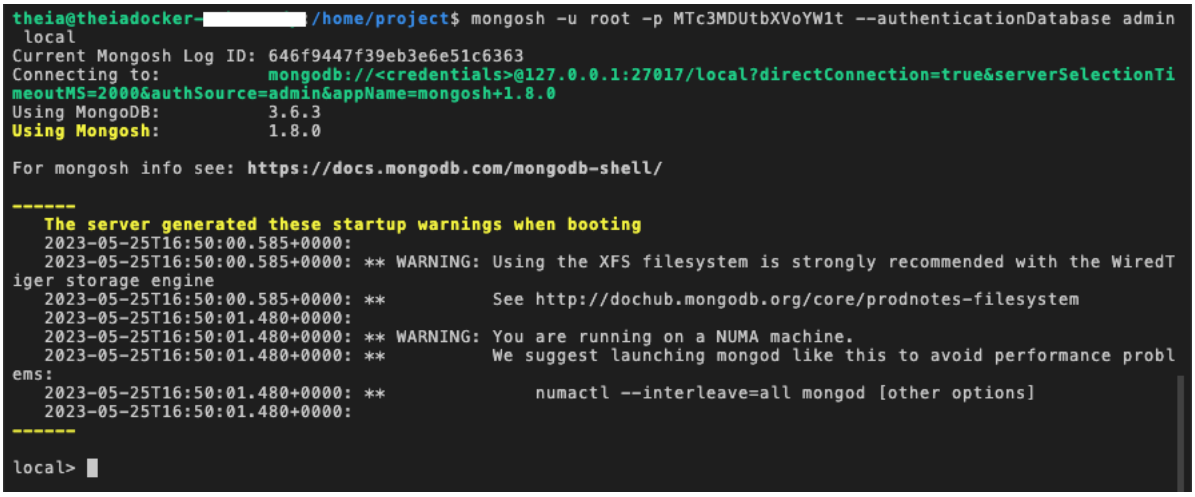


This action will open a new terminal at the end of the screen, as in the image below.



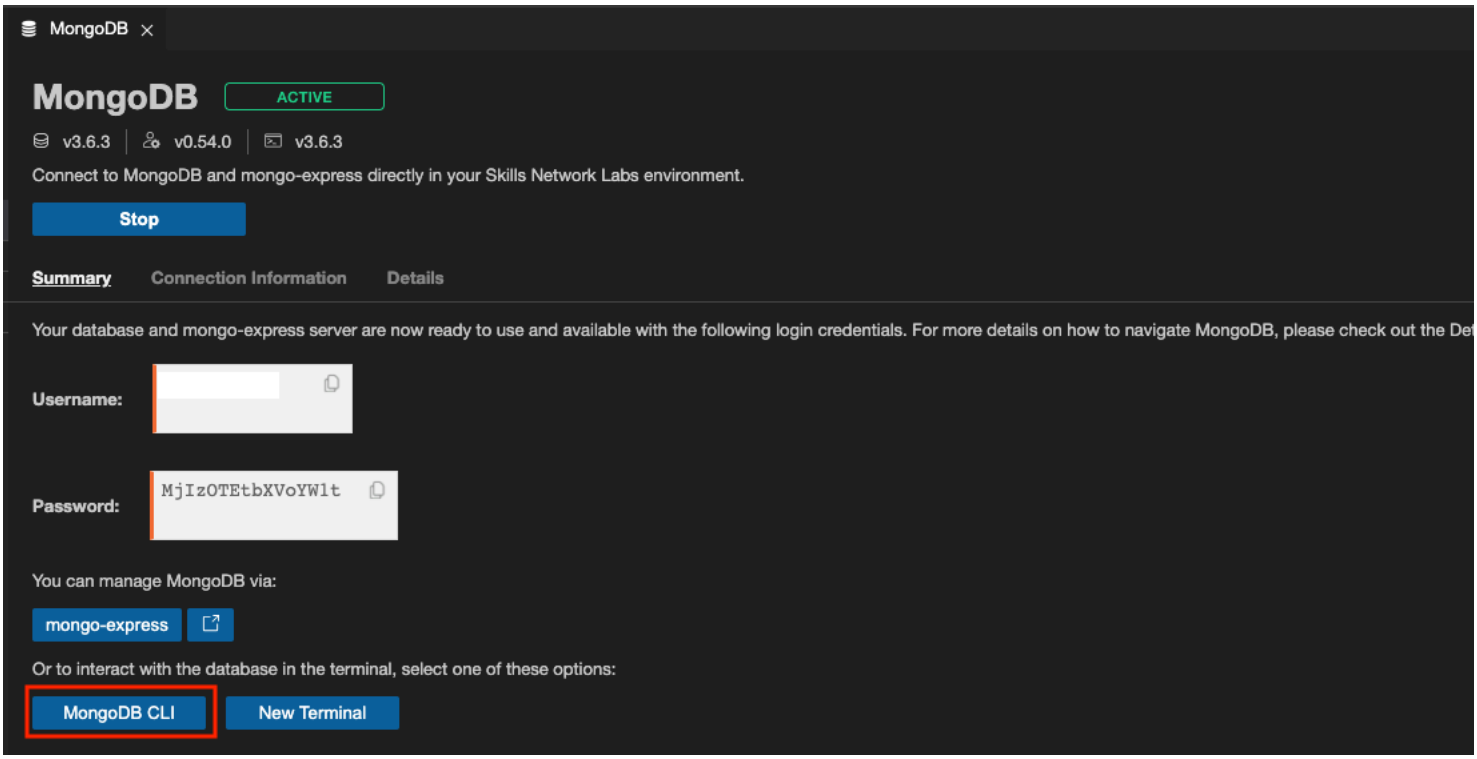
Run the following command on the newly opened terminal. (Copy the code by selecting the copy button on the lower right of the code block and then paste it wherever you wish.)

```
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 handy. You will need it in the next step.

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



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

```
use training
```

And create a collection called `mycollection`

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

Exercise 1: Install the Pymongo driver

You need the Pymongo driver installed to access the MongoDB database from Python.

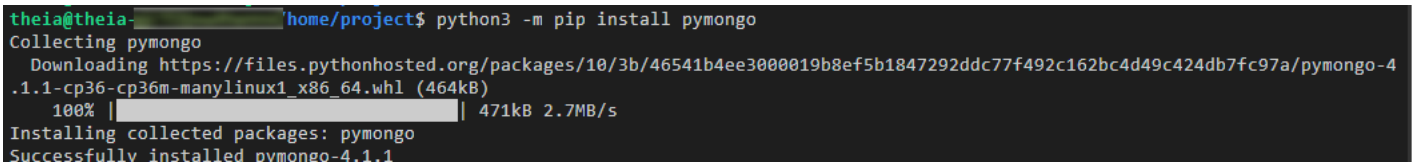
Open a new terminal and run the following command:

```
python3 -m pip install pymongo
```

If the above command results in error `/usr/bin/python3: No module named pip`, you need to install pip (PIP is a package manager for Python packages or modules) and then install `pymongo`.

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python3 get-pip.py
python3 -m pip install pymongo
```

This action installs the Python MongoDB driver like in the following image.



Exercise 2: Connect to mongodb server using Python

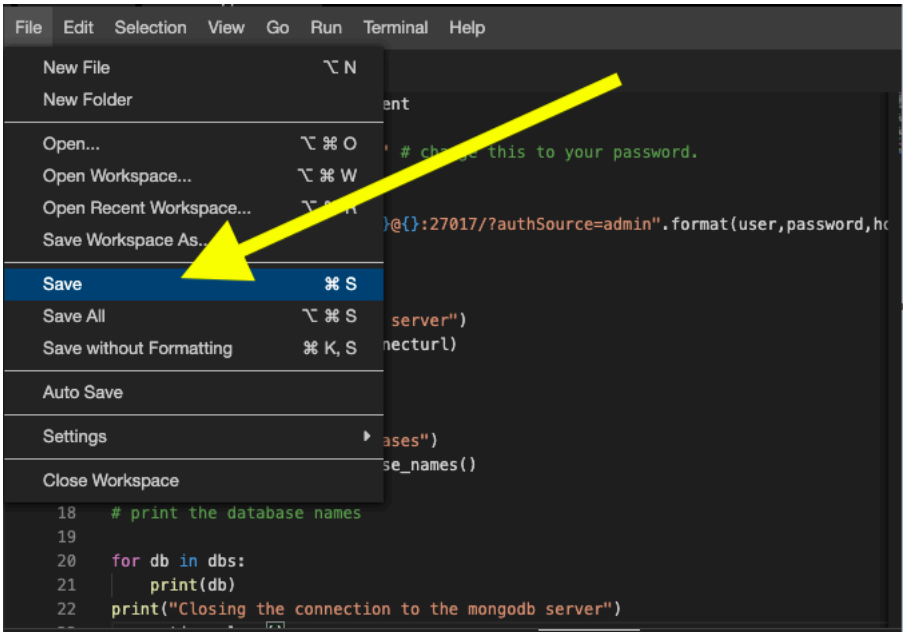
Open **mongo_connect.py** in IDE

Copy and paste the below code into this file.

```
from pymongo import MongoClient
user = 'root'
password = 'MjQwOTgtcnNhbm5h' # CHANGE THIS TO THE PASSWORD YOU NOTED IN THE EARLIER EXCERCISE - 2
host='localhost'
#create the connection url
connecturl = "mongodb://{}:{}@{}:27017/?authSource=admin".format(user,password,host)
# connect to mongodb server
print("Connecting to mongodb server")
connection = MongoClient(connecturl)
# get database list
print("Getting list of databases")
dbs = connection.list_database_names()
# print the database names
for db in dbs:
    print(db)
print("Closing the connection to the mongodb server")
# close the server connecton
connection.close()
```

Note: Please ensure that you have replaced the password value in the file above with the password for your MongoDB server.

Save the code file using the `File->Save` menu option, like in the following image.



Copy and paste the following code on the terminal to run this file.

```
python3 mongo_connect.py
```

You should see an output like the one in the following image.

Exercise 4 - Working with documents

In this exercise, you will make the Python program do the following tasks:

- Connect to the MongoDB server
- Select a database named `training`
- Select a collection named `python`
- Insert a sample document
- Query all the documents in the `training` database and `python` collection
- Close the connection to the server

Open **mongo_connect.py** in IDE

Copy and paste the following code into `mongo_query.py`.

```
from pymongo import MongoClient
user = 'root'
password = 'MjQwOTgtcnNhbm5h' # CHANGE THIS TO THE PASSWORD YOU NOTED IN THE EARLIER EXCERCISE - 2
host='localhost'
#create the connection url
connecturl = "mongodb://{}:{}@{}:27017/?authSource=admin".format(user,password,host)
# connect to mongodb server
print("Connecting to mongodb server")
connection = MongoClient(connecturl)
# select the 'training' database
db = connection.training
# select the 'python' collection
collection = db.python
# create a sample document
doc = {"lab":"Accessing mongodb using python", "Subject":"No SQL Databases"}
# insert a sample document
print("Inserting a document into collection.")
db.collection.insert_one(doc)
# query for all documents in 'training' database and 'python' collection
docs = db.collection.find()
print("Printing the documents in the collection.")
for document in docs:
    print(document)
# close the server connecton
print("Closing the connection.")
connection.close()
```

Note: Please ensure you have replaced the password value in the above file with the password for the MongoDB server you copied.

Save the file.

Run the file using the following command.

```
python3 mongo_query.py
```

You should see an output like the one in the following below.

```
theia@theiadocker-[redacted]:/home/project$ python3 mongo_query.py
Connecting to mongodb server
Inserting a document into collection.
Printing the documents in the collection.
{'_id': ObjectId('64b5b63efe8fc494a6a999bb'), 'lab': 'Accessing mongodb using python', 'Subject': 'No SQL Databases'}
Closing the connection.
```

Practice exercise

Write a Python program that can:

- Connect to the MongoDB server
- Select a database named `training`
- Select a collection named `mongodb_glossary`
- Insert the following documents into the collection `mongodb_glossary`

```
{"database":"a database contains collections"}
{"collection":"a collection stores the documents"}
{"document":"a document contains the data in the form of key value pairs."}
```

- Query and print all the documents in the `training` database and `mongodb_glossary` collection
- Close the connection to the server

Solution to practice exercise

```
from pymongo import MongoClient
user = 'root'
password = 'MjQwOTgtcnNhbm5h' # CHANGE THIS TO THE PASSWORD YOU NOTED IN THE EARLIER EXCERCISE - 2
host='localhost'
#create the connection url
connecturl = "mongodb://{}:{}@{}:27017/?authSource=admin".format(user,password,host)
# connect to mongodb server
print("Connecting to mongodb server")
connection = MongoClient(connecturl)
# select the 'training' database
db = connection.training
# select the 'python' collection
collection = db.mongodb_glossary
# create documents
doc1 = {"database":"a database contains collections"}
doc2 = {"collection":"a collection stores the documents"}
doc3 = {"document":"a document contains the data in the form or key value pairs."}
# insert documents
print("Inserting documents into collection.")
db.collection.insert_one(doc1)
db.collection.insert_one(doc2)
db.collection.insert_one(doc3)
# query for all documents in 'training' database and 'python' collection
docs = db.collection.find()
print("Printing the documents in the collection.")
for document in docs:
    print(document)
# close the server connecton
print("Closing the connection.")
connection.close()
```

Summary

In this lab, you have gained an understanding of working with MongoDB in Python.

Author(s)

Muhammad Yahya