Hands-on Lab: Setup and Practice Assignment

Estimated time needed: **30** minutes

Objectives

In this assignment you will:

- install the couchimport and couchexport tools.
- install the mongoimport and mongoexport tools.
- export data from a Cloudant database.
- import data into a Cloudant database.
- 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

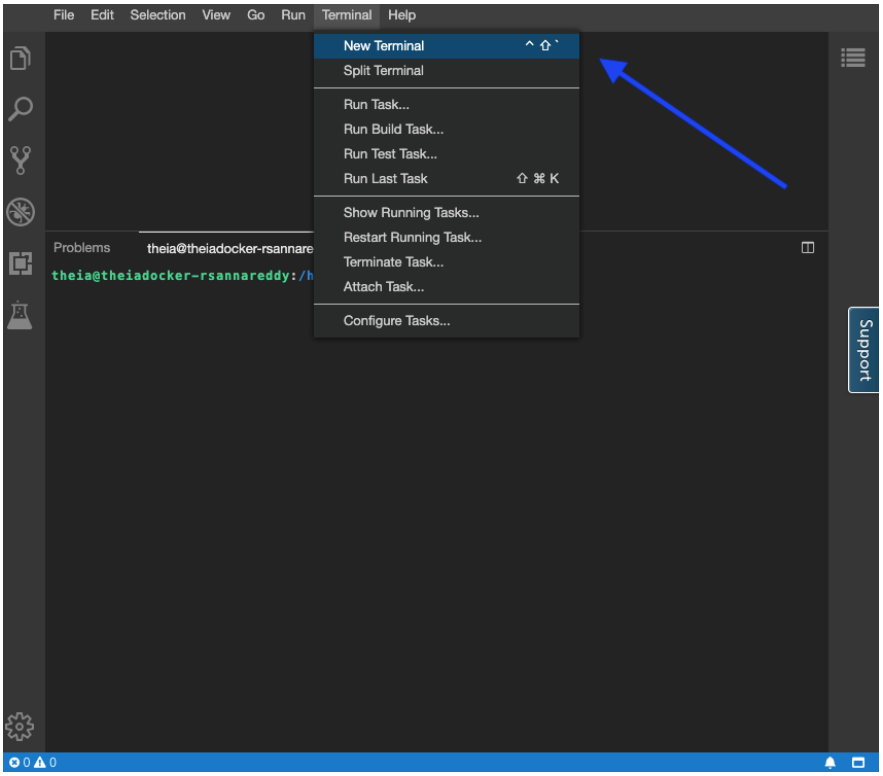
This Skills Network Labs Cloud IDE provides a hands-on environment for course and project related labs. It utilizes Theia, an open-source IDE (Integrated Development Environment) platform, that can be run on desktop or on the cloud. To complete this lab, we will be using the Cloud IDE based on Theia and Cassandra and Mongodb running in a Docker container. You will also need an instance of Cloudant running in IBM Cloud.

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.

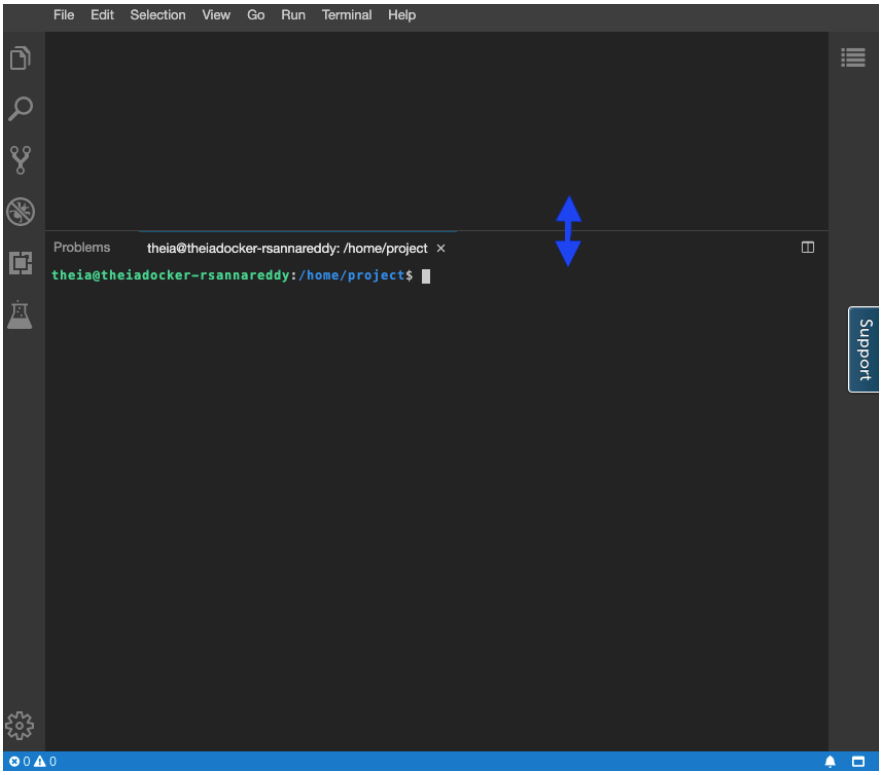
Useful tips about the lab IDE

The lab environment allows you to open multiple terminals should you need them.



Click on `New Terminal` from the Terminal menu.

You can resize the terminal window. You may wish to increase the terminal size to view longer outputs.



Exercise 1 - Getting the environment ready

You need the 'couchimport' and 'couchexport' tools to move data in and out of the Cloudant database.

To install these tools run the below commands on the terminal.

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

```
1.
2. npm install -g couchimport
```

Copied!

Here is the expected output.

```
theia@theiadocker-[redacted]:/home/project$ npm install -g couchimport

added 30 packages, and audited 31 packages in 3s

3 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Verify that the tool got installed, by running the below command on the terminal.

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

```
1.
2. couchimport --version
```

Copied!

If you do not get an error and get a version number, you are good to go ahead.

You need the 'mongoimport' and 'mongoexport' tools to move data in and out of the mongodb database.

To install these tools run the below commands on the terminal.

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
```

```
1.
2. wget https://fastdl.mongodb.org/tools/db/mongodb-database-tools-ubuntu1804-x86_64-100.3.1.tgz
3. tar -xf mongodb-database-tools-ubuntu1804-x86_64-100.3.1.tgz
4. export PATH=$PATH:/home/project/mongodb-database-tools-ubuntu1804-x86_64-100.3.1/bin
5. echo "done"
6.
```

Copied!

Verify that the tools got installed, by running the below command on the terminal.

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

```
1.
2. mongoimport --version
```

Copied!

If you do not get an error and get a version number, you are good to go ahead.

Exercise 2 - Cloudant import/export data

Before going ahead set the environment varible CLOUDANTURL to your actual cloudant url from your service credentials.

▸ Click here for Hint
▸ Click here for Solution

Export data from the 'diamonds' database into csv format. (You have created this database in the Cloudant practice labs. If you do not have this database, use any database you have on your Cloudant instance.)

```
1. 1
2. 2

1.
2. couchexport --url $CLOUDANTURL --db diamonds --delimiter ","
```

Copied!

You should see all the documents in the 'diamonds' database printed in csv format.

Export data from the 'diamonds' database into json format (one document per line).

```
1. 1
2. 2

1.
2. couchexport --url $CLOUDANTURL --db diamonds --type jsonl
```

Copied!

You should see all the documents in the 'diamonds' database printed in json format.

Export data from the 'diamonds' database into json format and save to a file named 'diamonds.json'.

```
1. 1
2. 2

1.
2. couchexport --url $CLOUDANTURL --db diamonds --type jsonl > diamonds.json
```

Copied!

Export data from the 'diamonds' database into csv format and save to a file named 'diamonds.csv'.

```
1. 1
2. 2

1.
2. couchexport --url $CLOUDANTURL --db diamonds --delimiter "," > diamonds.csv
```

Copied!

Exercise 3 - MongoDB import/export data

Start the mongodb server.

```
1. 1
2. 2

1.
2. start_mongo
```

Copied!

Make a note of the username and password based on the output of the previous command.

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 own password.

```
1. 1
2. 2

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

Copied!

Login to mongodb and check if the 'training' database and the 'diamonds' collection are created 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
2. 2

1.
2. mongoexport -u root -p MzA2NDAtcnNhbm5h --authenticationDatabase admin --db training --collection diamonds --out mongodb_exported_data.json
```

Copied!

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
2. 2

1.
2. mongoexport -u root -p MzA2NDAtcnNhbm5h --authenticationDatabase admin --db training --collection diamonds --out mongodb_exported_data.csv --type=csv --fields _id,clarity,cut,price
```

Copied!

Exercise 4 - Cassandra import/export data

Start the Cassandra server.

```
1. 1
2. 2

1.
2. start_cassandra
```

Copied!

Import csv into cassandra.

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

Step - 1: Login to cqlsh.

Step - 2: Create a keyspace named 'training'.

▸ Click here for Hint

▾ Click here for Solution

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

```
1.
2. CREATE KEYSPACE training
3. WITH replication = {'class':'SimpleStrategy', 'replication_factor' : 3};
```

Copied!

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

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

```
1.
2. use training;
3. CREATE TABLE diamonds(
4. id int PRIMARY KEY,
5. clarity text,
6. cut text,
7. price int
8. );
```

Copied!

Step 4: Run the below commands on cqlsh.

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

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

Copied!

Export the 'diamonds' table into a csv file.

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

```
1.
2. COPY diamonds TO 'cassandra-diamonds.csv';
```

Copied!

Exercise 5 - Creating an index on Cassandra

The command below creates an index named 'price_index' for the 'price' column which is in the 'diamonds' table.

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

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

Copied!

Exercise 6 - Disconnect from the Cassandra server

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

```
1.
2. exit
```

Copied!

Authors

Ramesh Sannareddy

Other Contributors

Rav Ahuja

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2021-11-22	0.3	Kathy An	Updated lab instructions
2021-04-23	0.2	Steve Ryan	Review pass
2021-04-16	0.1	Ramesh Sannareddy	Created initial version