

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

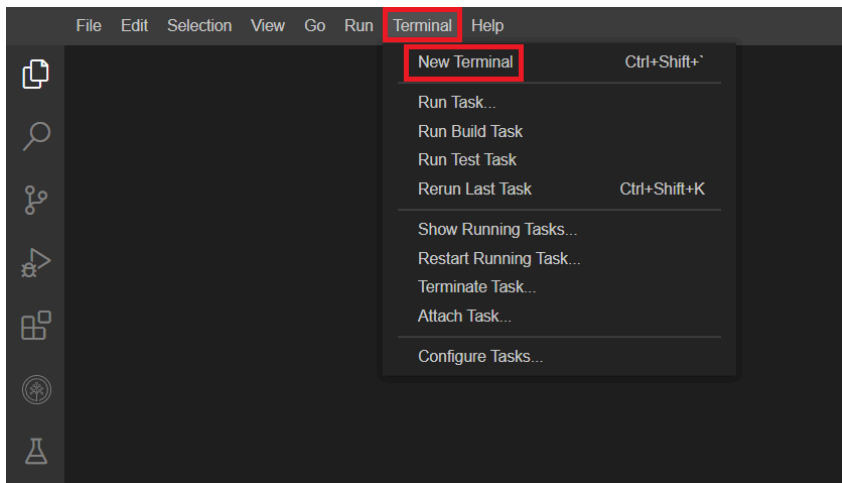
This Skills Network Labs Cloud IDE provides a hands-on environment for course and project related labs. It utilizes Theia, an open-source IDE (Integrated Development Environment) platform, that can be run on desktop or on the cloud. To complete this lab, we will be using the Cloud IDE based on Theia and Cassandra and MongoDB provided by Skills Network. 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.

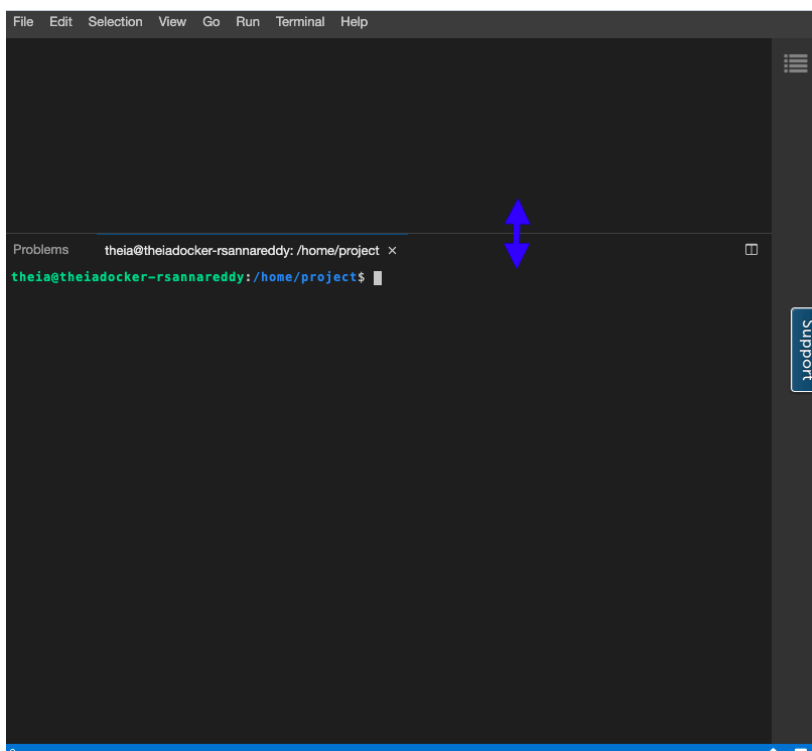
Useful tips about the lab IDE

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.

```
sudo npm install -g couchimport@1.4.0
```

Here is the expected output.

```
theia@theiadocker:~/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.

```
couchimport --version
```

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.

```
wget https://fastdl.mongodb.org/tools/db/mongodb-database-tools-ubuntu1804-x86_64-100.3.1.tgz
tar -xzf mongodb-database-tools-ubuntu1804-x86_64-100.3.1.tgz
export PATH=$PATH:/home/project/mongodb-database-tools-ubuntu1804-x86_64-100.3.1/bin
echo "done"
```

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

```
mongoimport --version
```

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 variable CLOUDANTURL to your actual cloudant url from your service credentials.

▼ Click here for Hint

[Using HTTP API to create and query Cloudant databases](#) lab has instructions to setup the CLOUDANTURL variable.

▼ Click here for Solution

```
export CLOUDANTURL="YOUR_URL_HERE"
```

Where the YOUR_URL_HERE is the url found under your instance's **Service credentials**.

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.)

```
couchexport --url $CLOUDANTURL --db diamonds --delimiter ","
```

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).

```
couchexport --url $CLOUDANTURL --db diamonds --type jsonl
```

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’.

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

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

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

Exercise 3 - MongoDB import/export data

Open the MongoDB database page by clicking the button below:

Open MongoDB Page in IDE

On that page, click the Create button to create a MongoDB database.

Note down the user name and password from the Connection Information tab.

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.

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

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’

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

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’

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

Exercise 4 - Cassandra import/export data

Open the Cassandra database page by clicking the button below:

Open Cassandra Page in IDE

On that page, click the Create button to create a Cassandra database.

After creation has finished, click the Cassandra CLI button (below the Create button from the previous step) to connect to the database using the cqlsh CLI.

Import csv into cassandra.

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

Step - 1: Create a keyspace named ‘training’.

▼ Click here for Hint

Refer to the [Keyspace Operations Hands on Lab](#)

▼ Click here for Solution

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

Step - 2: 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

Refer to the [Table Operations Hands on Lab](#)

▼ Click here for Solution

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

Step 3: Run the below commands on cqlsh.

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

Export the 'diamonds' table into a csv file.

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

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.

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

Exercise 6 - Disconnect from the Cassandra server

```
exit
```

Authors

Ramesh Sannareddy

Other Contributors

Rav Ahuja

© IBM Corporation. All rights reserved.