



Connect to Db2 database on Cloud using Python

Estimated time needed: **15** minutes

Objectives

After completing this lab you will be able to:

- Import the `ibm_db` Python library
- Enter the database connection credentials
- Create the database connection
- Close the database connection

Note: Please follow the instructions given in the first Lab of this course to Create a database service instance of Db2 on Cloud and retrieve your database Service Credentials.

Import the `ibm_db` Python library

The `ibm_db` API provides a variety of useful Python functions for accessing and manipulating data in an IBM® data server database, including functions for connecting to a database, preparing and issuing SQL statements, fetching rows from result sets, calling stored procedures, committing and rolling back transactions, handling errors, and retrieving metadata.

We first import the `ibm_db` library into our Python Application

Execute the following cells by clicking within it and then press `Shift` and `Enter` keys simultaneously

The following required modules are pre-installed in the Skills Network Labs environment. However if you run this notebook commands in a different Jupyter environment (e.g. Watson Studio or Ananconda) you may need to install these libraries by removing the `#` sign before `!pip` in the code cell below.

```
In [1]: # These libraries are pre-installed in SN Labs. If running in another environment p
# !pip install --force-reinstall ibm_db==3.1.0 ibm_db_sa==0.3.3
# Ensure we don't load_ext with sqlalchemy>=1.4 (incompadible)
# !pip uninstall sqlalchemy==1.4 -y && pip install sqlalchemy==1.3.24
# !pip install ipython-sql
```

```
In [2]: !pip install --force-reinstall ibm_db ibm_db_sa
import ibm_db
```

```
Collecting ibm_db
  Downloading ibm_db-3.2.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (1.4 kB)
Collecting ibm_db_sa
  Downloading ibm_db_sa-0.4.1-py3-none-any.whl.metadata (5.3 kB)
Collecting sqlalchemy>=0.7.3 (from ibm_db_sa)
  Downloading SQLAlchemy-2.0.36-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (9.7 kB)
Collecting typing-extensions>=4.6.0 (from sqlalchemy>=0.7.3->ibm_db_sa)
  Downloading typing_extensions-4.12.2-py3-none-any.whl.metadata (3.0 kB)
Collecting greenlet!=0.4.17 (from sqlalchemy>=0.7.3->ibm_db_sa)
  Downloading greenlet-3.1.1-cp311-cp311-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl.metadata (3.8 kB)
Downloading ibm_db-3.2.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (43.6 MB)
_____ 43.6/43.6 MB 62.4 MB/s eta 0:00:00:00:01
00:01
Downloading ibm_db_sa-0.4.1-py3-none-any.whl (31 kB)
Downloading SQLAlchemy-2.0.36-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.2 MB)
_____ 3.2/3.2 MB 104.2 MB/s eta 0:00:00
Downloading greenlet-3.1.1-cp311-cp311-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl (602 kB)
_____ 602.4/602.4 kB 48.1 MB/s eta 0:00:00
Downloading typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Installing collected packages: ibm_db, typing-extensions, greenlet, sqlalchemy, ibm_db_sa
  Attempting uninstall: typing-extensions
    Found existing installation: typing_extensions 4.11.0
    Uninstalling typing_extensions-4.11.0:
      Successfully uninstalled typing_extensions-4.11.0
  Attempting uninstall: greenlet
    Found existing installation: greenlet 3.0.3
    Uninstalling greenlet-3.0.3:
      Successfully uninstalled greenlet-3.0.3
  Attempting uninstall: sqlalchemy
    Found existing installation: SQLAlchemy 2.0.30
    Uninstalling SQLAlchemy-2.0.30:
      Successfully uninstalled SQLAlchemy-2.0.30
Successfully installed greenlet-3.1.1 ibm_db-3.2.3 ibm_db_sa-0.4.1 sqlalchemy-2.0.36 typing-extensions-4.12.2
```

When the command above completes, the `ibm_db` library is loaded in your notebook.

Identify the database connection credentials

Connecting to dashDB or DB2 database requires the following information:

- Driver Name
- Database name
- Host DNS name or IP address
- Host port
- Connection protocol
- User ID (or username)
- User Password

Notice: To obtain credentials please refer to the instructions given in the first Lab of this course

Now enter your database credentials below and execute the cell with **Shift + Enter**

```
In [12]: #Replace the placeholder values with your actual Db2 hostname, username, and password
dsn_hostname = "1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases"
dsn_uid = "jwg19774"          # e.g. "abc12345"
dsn_pwd = "RsgH6FXWZ30srUD9" # e.g. "7dBZ3wWt9XN6$o0J"

dsn_driver = "{IBM DB2 ODBC DRIVER}"
dsn_database = "bludb"          # e.g. "BLUDB"
dsn_port = "32286"             # e.g. "32733"
dsn_protocol = "TCPIP"         # i.e. "TCPIP"
dsn_security = "SSL"           #i.e. "SSL"
```

Create the DB2 database connection

Ibm_db API uses the IBM Data Server Driver for ODBC and CLI APIs to connect to IBM DB2 and Informix.

Lets build the dsn connection string using the credentials you entered above

```
In [13]: #DO NOT MODIFY THIS CELL. Just RUN it with Shift + Enter
#Create the dsn connection string
dsn = (
    "DRIVER={0};"
    "DATABASE={1};"
    "HOSTNAME={2};"
    "PORT={3};"
    "PROTOCOL={4};"
    "UID={5};"
    "PWD={6};"
    "SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname, dsn_port, dsn_p

#print the connection string to check correct values are specified
print(dsn)
```

```
DRIVER={IBM DB2 ODBC DRIVER};DATABASE=bludb;HOSTNAME=1bbf73c5-d84a-4bb0-85b9-ab1a434
8f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=32286;PROTOCOL=TCPIP;UID=j
wg19774;PWD=RsgH6FXWZ30srUD9;SECURITY=SSL;
```

Now establish the connection to the database

```
In [14]: #DO NOT MODIFY THIS CELL. Just RUN it with Shift + Enter
#Create database connection

try:
    conn = ibm_db.connect(dsn, "", "")
    print ("Connected to database: ", dsn_database, "as user: ", dsn_uid, "on host: ")

except:
    print ("Unable to connect: ", ibm_db.conn_errormsg() )
```

```
Connected to database: bludb as user: jwg19774 on host: 1bbf73c5-d84a-4bb0-85b9-a
b1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud
```

Congratulations if you were able to connect successfully. Otherwise check the error and try again.

```
In [15]: #Retrieve Metadata for the Database Server
server = ibm_db.server_info(conn)

print ("DBMS_NAME: ", server.DBMS_NAME)
print ("DBMS_VER:  ", server.DBMS_VER)
print ("DB_NAME:   ", server.DB_NAME)
```

```
DBMS_NAME:  DB2/LINUX8664
DBMS_VER:   11.05.0900
DB_NAME:    BLUDB
```

```
In [16]: #Retrieve Metadata for the Database Client / Driver
client = ibm_db.client_info(conn)

print ("DRIVER_NAME:      ", client.DRIVER_NAME)
print ("DRIVER_VER:       ", client.DRIVER_VER)
print ("DATA_SOURCE_NAME:  ", client.DATA_SOURCE_NAME)
print ("DRIVER_ODBC_VER:    ", client.DRIVER_ODBC_VER)
print ("ODBC_VER:           ", client.ODBC_VER)
print ("ODBC_SQL_CONFORMANCE: ", client.ODBC_SQL_CONFORMANCE)
print ("APPL_CODEPAGE:      ", client.APPL_CODEPAGE)
print ("CONN_CODEPAGE:      ", client.CONN_CODEPAGE)
```

```
DRIVER_NAME:      libdb2.a
DRIVER_VER:       11.05.0900
DATA_SOURCE_NAME: BLUDB
DRIVER_ODBC_VER:  03.51
ODBC_VER:         03.01.0000
ODBC_SQL_CONFORMANCE: EXTENDED
APPL_CODEPAGE:    1208
CONN_CODEPAGE:    1208
```

Close the Connection

We free all resources by closing the connection. Remember that it is always important to close connections so that we can avoid unused connections taking up resources.

```
In [17]: ibm_db.close(conn)
```

```
Out[17]: True
```

Summary

In this tutorial you established a connection to a DB2 database on Cloud database from a Python notebook using ibm_db API.

Author

[Rav Ahuja](#)

© IBM Corporation 2020. All rights reserved.

```
{toggle}
{toggle}|
{toggle}|---|---|---|---|
{toggle}|
{toggle}|
{toggle}|
```