



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 environmen
# !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]: `import ibm_db`

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 [6]: #Replace the placeholder values with your actual Db2 hostname, username, and pas
dsn_hostname = "1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databa
dsn_uid = "dxc70089"          # e.g. "abc12345"
dsn_pwd = "A9vq58iRSQlvpgZk" # 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 [7]: #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, ds

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

```

```

DRIVER={IBM DB2 ODBC DRIVER};DATABASE=BLUDB;HOSTNAME=1bbf73c5-d84a-4bb0-85b9-ab1a
4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286;PORT=32286;PROTOCOL
=TCPIP;UID=dx70089;PWD=A9vq58iRSQlvpgZk;SECURITY=SSL;

```

Now establish the connection to the database

```

In [8]: #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 ho

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

```

```

Unable to connect: [IBM][CLI Driver] SQL1336N The remote host "1bbf73c5-d84a-4b
b0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.ap" was not found.  SQLSTATE=
08001  SQLCODE=-1336

```

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

```

In [9]: #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)

```

```

-----
NameError                                Traceback (most recent call last)
/tmp/ipykernel_697/4159812593.py in <module>
      1 #Retrieve Metadata for the Database Server
----> 2 server = ibm_db.server_info(conn)
      3
      4 print ("DBMS_NAME: ", server.DBMS_NAME)
      5 print ("DBMS_VER: ", server.DBMS_VER)

NameError: name 'conn' is not defined

```

```

In [10]: #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)

```

```

-----
NameError                                Traceback (most recent call last)
/tmp/ipykernel_697/2038037146.py in <module>
      1 #Retrieve Metadata for the Database Client / Driver
----> 2 client = ibm_db.client_info(conn)
      3
      4 print ("DRIVER_NAME:           ", client.DRIVER_NAME)
      5 print ("DRIVER_VER:           ", client.DRIVER_VER)

NameError: name 'conn' is not defined

```

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 [11]: ibm_db.close(conn)
```

```

-----
NameError                                Traceback (most recent call last)
/tmp/ipykernel_697/3798206860.py in <module>
----> 1 ibm_db.close(conn)

NameError: name 'conn' is not defined

```

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](#)

Change Log

| Date (YYYY-MM-DD) | Version | Changed By | Change Description |
|-------------------|---------|------------|------------------------------------|
| 2021-11-17 | 2.2 | Lakshmi | Updated library |
| 2021-07-09 | 2.1 | Malika | Updated the connection string |
| 2020-08-28 | 2.0 | Lavanya | Moved lab to course repo in GitLab |

© IBM Corporation 2020. All rights reserved.