/ labs / DB0201EN /

| Name | Last Modified |
| --- | --- |
| DB0201EN-Week3-1-1-Conne... | seconds ago |

Launcher | DB0201EN-Week3-1-1-Conn

Markdown | Python


# Lab: Connect to Db2 database on Cloud using Python

## Introduction

This notebook illustrates how to access a DB2 database on Cloud using Python by following the steps below:

1. Import the `ibm_db` Python library
2. Enter the database connection credentials
3. Create the database connection
4. 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 cell by clicking within it and then press `Shift` and `Enter` keys simultaneously

[1]:
```
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`

[2]:
```
#Replace the placeholder values with your actual Db2 hostname, username, and password:
dsn_hostname = "dashdb-txn-sbox-yp-lon02-01.services.eu-gb.bluemix.net" # e.g.: "dashdb-txn-sbox-yp-dal09-04.services.dal.bluemix.net"
dsn_uid = "gkm89241"        # e.g. "abc12345"
dsn_pwd = "w13fqqz1cdc^pqqr"      # e.g. "7dBZ3wWt9XN6$o0J"

dsn_driver = "{IBM DB2 ODBC DRIVER}"
dsn_database = "BLUDB"            # e.g. "BLUDB"
dsn_port = "50000"                # e.g. "50000"
dsn_protocol = "TCPIP"            # i.e. "TCPIP"
```

## 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

[3]:
```
#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};").format(dsn_driver, dsn_database, dsn_hostname, dsn_port, dsn_protocol, dsn_uid, dsn_pwd)

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

```
DRIVER={IBM DB2 ODBC DRIVER};DATABASE=BLUDB;HOSTNAME=dashdb-txn-sbox-yp-lon02-01.services.eu-gb.bluemix.net;PORT=50000;PROTOCOL=TCPIP;UID=gkm89241;PWD=w13fqqz1cdc^pqqr;
```

Now establish the connection to the database

[4]:
```
#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: ", dsn_hostname)

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

```
Connected to database:  BLUDB as user:  gkm89241 on host:  dashdb-txn-sbox-yp-lon02-01.services.eu-gb.bluemix.net
```

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

[5]:
```
#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/LINUXX8664
DBMS_VER:   11.01.0303
DB_NAME:    BLUDB
```

[6]:
```
#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:           11.01.0404
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.



[7]:
```
ibm_db.close(conn)
```

[7]:
```
True
```

## Summary

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