INVENTORY MANAGEMENT SYSTEM FOR RETAILERS

Domain: Cloud Application Development

Team Id: PNT2022TMID14519

Create IBM DB2 And connect with python

In order to use the IBM DB2 DBMS, you have to make a IBM cloud account. There is an IBM DB2 Lite plan that is free to use.

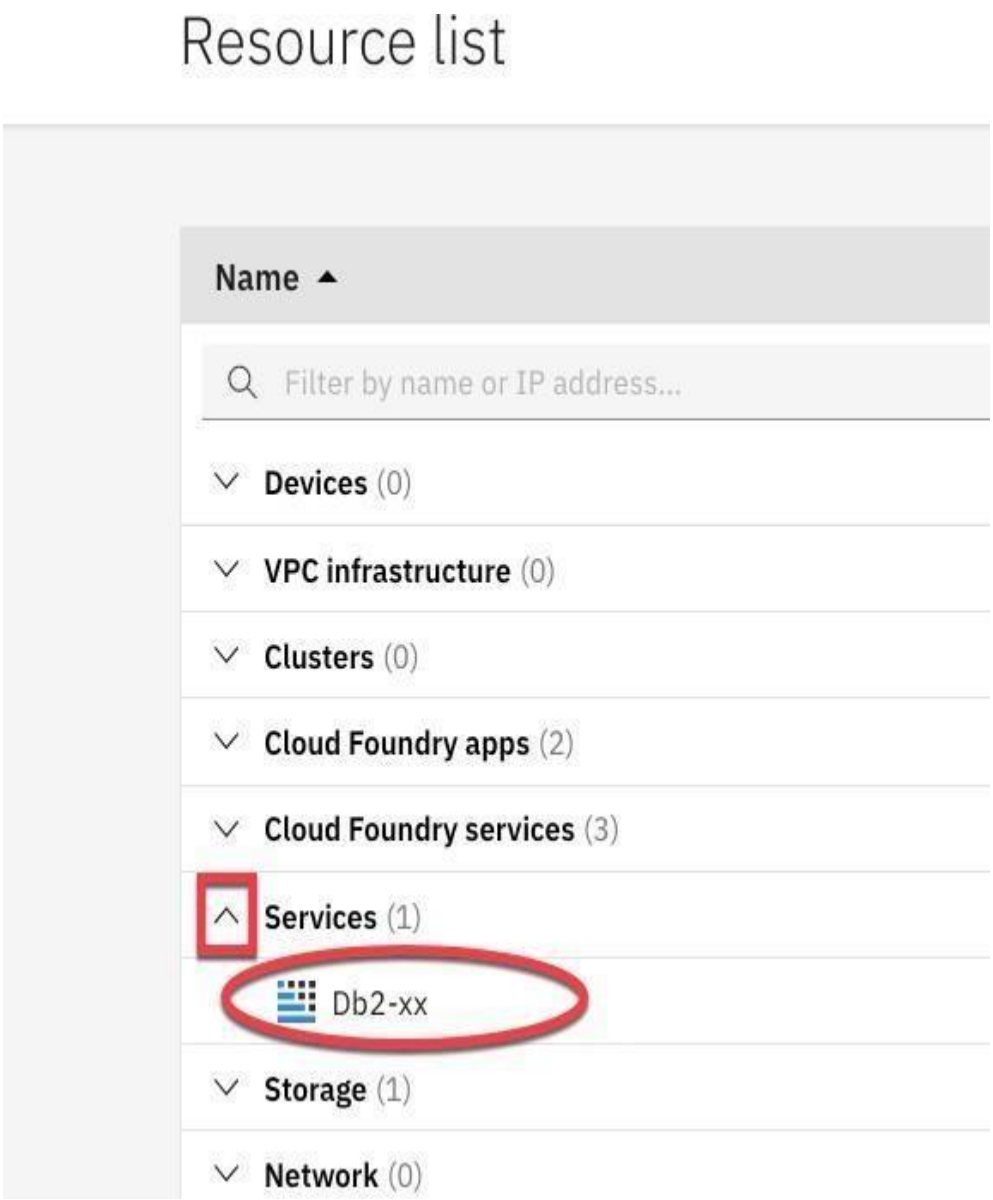
Go to this link : https://cloud.ibm.com/registration to make an IBM cloud account.

After logging in your IBM cloud account. You will notice a catalog option on the top just to the left of the search bar available on the webpage.

After going in the catalog you will see the webpage given below.

Make sure you choose DB2 and nothing else such as DB2 Warehouse, DB2 Hosted or SQL Query.

After that everything will be pre — selected, you just have to move down to Pricing Plans and select the Lite plan as it is a free plan.



Then click on the Create at the bottom right of the page.

After that open your dashboard in the IBM cloud.

After that click on the open console button. This will open a new tab on your web browser and then choose the 3rd option from the top left drop down menu, now if you want to run SQL queries. You can do it from here

We need to have Service Credentials in order to access the database from Python. So, you have to go to the web page where there is an Open Console button.

On the left side you can see the Service Credentials option. Click on that button and then click on the New Credentials button to generate Service Credentials for your IBM DB2 Database.

We will need the credentials later on.

Starting with the Python Code

First of all you need to download the python library ibm_db.

You can see how to download ibm_db library here : *https://pypi.org/project/ibm-db/*

After that you have to import ibm_db in the jupyter notebook.

```
In [21]: import ibm_db
```

The credentials that you will be needing to connect to the database are as follows :

1. Driver Name
2. Database Name
3. Host DNS name or IP Address
4. Host Port
5. Connection Protocol
6. Username
7. Password

```
In [22]: dsn_hostname = "YourDb2Hostname" # "dashdb-txn-sbox-yp-dal09-04.services.dal.bluemix.net"
         dsn_uid = "YourDb2Username"        # "abc12345"
         dsn_pwd = "YoueDb2Password"      # "7dBZ3wWt9XN6$o0J"

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

The dsn_driver will remain the same. Others may change, so you have to refer to the credentials in order to replace the values.

The ibm_db API uses the IBM Data Server Driver for Open Database Connectivity and Command Line Interface API's to connect to the IBM DB2 database.

```
In [23]: 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(dsn)
```

```
In [24]: 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() )
```

You have to print the results in order to check if the details are correct

```
In [30]: insertQuery = "insert into CARTOON_CHARACTERS values (1, 'Mickey', 'Mouse', '123 Fantasy Way', 'Anaheim', 18), (
         insert_table = ibm_db.exec_immediate(conn, insertQuery)
```

```
In [25]: 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:   11.01.0404
DB_NAME:    BLUDB
```

First Name	Last Name	Address	City	Age
Mickey	Mouse	123 Fantasy Way	Anaheim	73
Bat	Man	321 Cavern Ave	Gotham	54
Wonder	Woman	987 Truth Way	Paradise	39

First of all we will give the query to create the above table.

```
In [14]: createQuery = "create Table CARTOON_CHARACTERS(ID INTEGER PRIMARY KEY NOT NULL, First_Name VARCHAR(20) NOT NULL, Las
         create_table = ibm_db.exec_immediate(conn, createQuery)
```

Here, ibm_db.exec_immediate() is the function that will send the query to your IBM database and create changes in the database.

Now we will insert all the data into the database.

```
In [25]: 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:   11.01.0404
DB_NAME:    BLUDB
```

After inserting the data into the database we will check if the table in the data has been modified or not, so we will run the following command

```
stmt = ibm_db.exec_immediate(conn, "select * from CARTOON_CHARACTERS")

while ibm_db.fetch_row(stmt) != False:
    print(" ID:", ibm_db.result(stmt, 0), "First Name - ", ibm_db.result(stmt, 1))
```

```
 ID: 1 First Name -  Mickey
 ID: 2 First Name -  Bat
 ID: 3 First Name -  Wonder
```

```
In [51]: pd_conn = ibm_db_dbi.Connection(conn)
```

Now, the best thing about accessing databases through python is that you can load the database into Pandas data frame and you can use all the data science tools on the database using all the Python data science libraries.

```
import pandas
import ibm_db_dbi
...
```

```
In [54]: selectQuery = "select * from CARTOON_CHARACTERS"

         dataframe = pandas.read_sql(selectQuery, pd_conn)

         dataframe
```

Out[54]:

	ID	FIRST_NAME	LAST_NAME	ADDRESS	CITY	AGE
0	1	Mickey	Mouse	123 Fantasy Way	Anaheim	18
1	2	Bat	Man	321 Cavern Ave	Gotham	32
2	3	Wonder	Woman	987 Truth Way	Paradise	28

Now we have to establish a connection for the pandas.

After establishing the connection, now we can load the database into the pandas data frame
I

Now you can do the typical pandas operations on the database.

For example, you can use the shape function of the pandas.

```
In [55]: dataframe.shape
Out[55]: (3, 6)
```

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

```
In [56]: ibm_db.close(conn)
Out[56]: True
```