

IMPLEMENTING WEB APPLICATIONS

CREATE IBM DB2 AND CONNECTION WITH PYTHON CODE

TEAM ID	PNT2022TMID41646
PROJECT NAME	SMART FASHION RECOMMENDER APPLICATION
TEAM MEMBERS	VELMANI J, MUKESH M, MURALIDHARAN M, ANAND K

1. Navigate to www.cloud.ibm.com
2. Go to -> catalog -> find for DB2.
3. Create the DB2 instance :

The screenshot displays the IBM Cloud console interface for the Db2 service. The main content area shows the 'Create' tab with options to 'Select a location' (Sydney) and 'Select a pricing plan'. A table lists the available plans, including the 'Standard' plan with its features and pricing details. The right sidebar provides a 'Summary' of the selected configuration, including location, plan, service name, and resource group. A blue 'Upgrade' button is visible at the bottom right of the main content area.

Plan	Features	Pricing
Standard	Instance with flexible scaling of compute and storage Base instance starts at 8 GB RAM x 20 GB Storage	\$0.136 USD/Instance-Hour \$0.00027 USD/Gigabyte-Hours \$0.097 USD/Virtual Processor Core-Hour \$0.00003 USD/BACKUP.GIGABYTE_HOURS \$0.0959 USD/SERVICEENDPOINT_INSTANCE_HOURS

4. Access the resources :

The screenshot shows the IBM Cloud console interface. The top navigation bar includes the IBM Cloud logo, a search bar, and a 'Create resource' button. The main content area is titled 'Resource list' and displays a table of resources. The table has columns for Name, Group, Location, Product, Status, and Tags. A resource named 'Db2-manoj' is listed under the 'Databases' group, located in 'Dallas', with a status of 'Active'. The left sidebar contains various icons for navigating through the console. The bottom of the screen shows a Windows taskbar with the system clock at 08:40 PM on 17-11-22.

Name	Group	Location	Product	Status	Tags
Db2-manoj	Default	Dallas	Db2	Active	

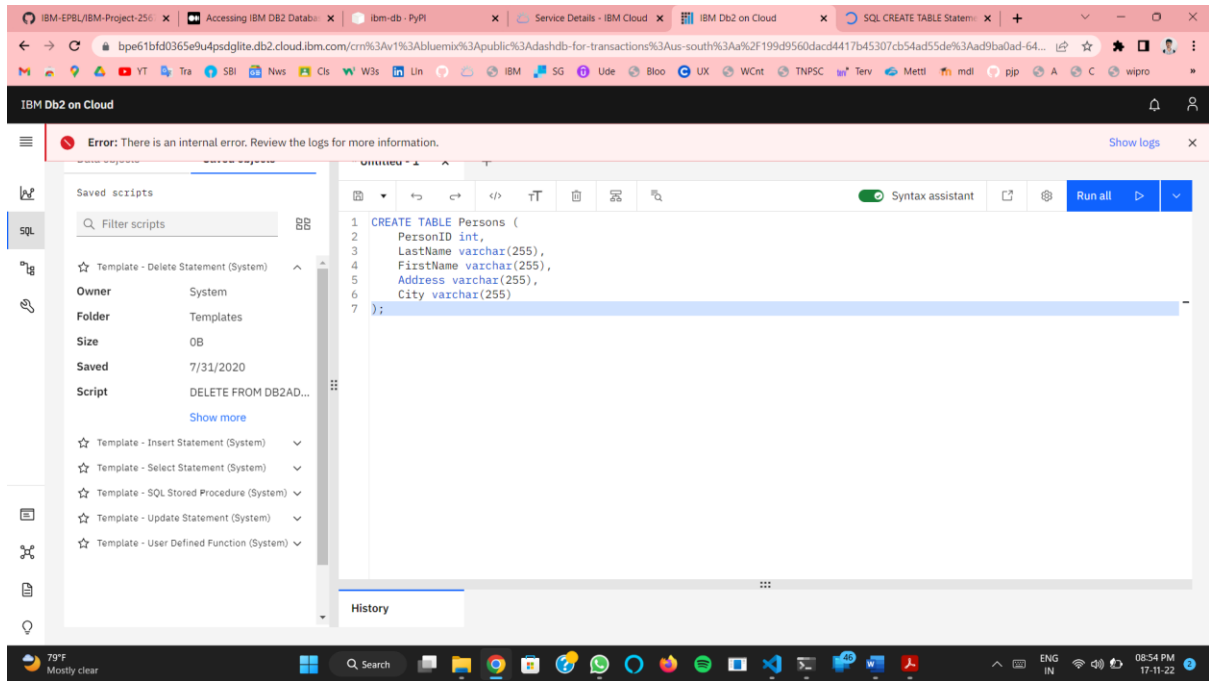
5. Accessing service credentials :

The screenshot shows the 'Service credentials' page for the 'Db2-manoj' resource. The page title is 'Service credentials' and it includes a description: 'You can generate a new set of credentials for cases where you want to manually connect an app or external consumer to an IBM Cloud service. Learn more'. A 'New credential' button is visible in the top right. Below this, a table lists the generated credentials. The table has columns for 'Key name' and 'Date created'. A single credential named 'Service credentials-1' is listed, created on '2022-10-10 7:01 PM'. The details of the credential are shown in a code block, including a 'connection' object with 'cli' and 'arguments' fields. The bottom of the screen shows a Windows taskbar with the system clock at 08:43 PM on 17-11-22.

Key name	Date created
Service credentials-1	2022-10-10 7:01 PM

```
{
  "connection": {
    "cli": {
      "arguments": [
        "-u",
        "tft98988",
        "-p",
        "W0h8dKXKR1568R1U",
        "--ssl",
        "--sslCAFile",
        "1cbbb1b6-3a1a-4d49-9262-3102a8f7a7c8",
        "--authenticationDatabase",
        "admin",
        "--host",
        "6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.clogj3ad8tgtu0lqde00.databases.appdomain.cloud:30376"
      ]
    }
  }
}
```

6. IBM DB2 Console :



7. Installation of IBM DB library in python :

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22h21.6193]
(c) Microsoft Corporation. All rights reserved.

C:\Users\vjman>pip install ibm-db
Collecting ibm-db
  Downloading ibm_db-3.1.3.tar.gz (1.4 MB)
    ----- 1.4/1.4 MB 422.7 KB/s eta 0:00:00
    Installing build dependencies ... done
    Getting requirements to build wheel ... done
    Installing backend dependencies ... done
    Preparing metadata (pyproject.toml) ... done
    Building wheels for collected packages: ibm-db
    Building wheel for ibm-db (pyproject.toml) ... done
    Created wheel for ibm-db: filename=ibm_db-3.1.3-py3-none-any.whl size=27735624 sha256=1ffffb2a2d63d44b7e5b61d5a7ab55b0fa2be75b25dc762a4ab8303bdcbb4d2f
    Stored in directory: c:\users\vjman\appdata\local\pip\cache\wheels\1f\4c\eb\6b\5f\b7\ed\65cc0777e23ca274708b503f23daf30fe156579d4
    Successfully built ibm-db
    Installing collected packages: ibm-db
    Successfully installed ibm-db-3.1.3

C:\Users\vjman>pip install wheel
Collecting wheel
  Using cached wheel-0.36.4-py3-none-any.whl (36 kB)
    Installing collected packages: wheel
    Successfully installed wheel-0.36.4

C:\Users\vjman>pip install build
Collecting build
  Downloading build-0.9.0-py3-none-any.whl (17 kB)
Collecting pep517-0.9.1
  Downloading pep517-0.13.0-py3-none-any.whl (16 kB)
Collecting tomli>=1.0.0
  Downloading tomli-2.0.1-py3-none-any.whl (12 kB)
Collecting colorama
  Downloading colorama-0.4.4-py2.py3-none-any.whl (25 kB)
Collecting packaging>=14.0
  Downloading packaging-21.3-py3-none-any.whl (40 kB)
    ----- 40.0/40.0 kB 277.4 KB/s eta 0:00:00
Collecting pyparsing!>3.0.5, <=2.0.2
  Downloading pyparsing-3.0.9-py3-none-any.whl (98 kB)
    ----- 98.3/98.3 kB 471.1 KB/s eta 0:00:00
Installing collected packages: tomli, pyparsing, colorama, pep517, packaging, build
Successfully installed build-0.9.0 colorama-0.4.4 packaging-21.3 pep517-0.13.0 pyparsing-3.0.9 tomli-2.0.1

C:\Users\vjman>
```

8. Connection between the IBM DB2 and Python :

```
app.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
app.py 9 X
D:\> IBM > app.py {} Flask
1 from flask import Flask,render_template,request,session
2 import ibm_db
3 import re
4
5 app=Flask(__name__)
6 app.secret_key='a'
7
8 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgu1qde00.databases.appdomain.cloud;PORT=30376;Security=SSL;SSLServerCertif
9 @app.route('/')
10 def home():
11     return render_template('home.html')
12
13 @app.route('/login',methods=['GET','POST'])
14 def login():
15     global userid
16     msg=''
17
18     if request.method=='POST':
19         username=request.form['username']
20         password=request.form['password']
21         sql="SELECT * FROM users WHERE username=? AND password=?"
22         stmt = ibm_db.prepre(conn,sql)
23         ibm_db.bind_param(stmt,1,username)
24         ibm_db.bind_param(stmt,2,password)
25         ibm_db.execute(stmt)
26         account=ibm_db.fetch_assoc(stmt)
27         print(account)
28         if account:
29             session['loggedin']=True
30             session['id']=account['USERNAME']
31             userid=account['USERNAME']
32             session['username']=account['USERNAME']
33             msg='logged in successfully'
34             return render_template('dashboard.html',msg=msg)
35         else:
36             msg='Incorrect username/password'
37     return render_template('login.html',msg=msg)
```