

Team ID	PNT2022TMID03825
Project Name	Inventory Management For Retailers

Resource list

[Create resource](#)

Name	Group	Location	Product	Status	Tags
<input type="text" value="Filter by name or IP address..."/>	<input style="background-color: #f0f0f0;" type="text" value="Filter by group or org..."/>	<input style="background-color: #f0f0f0;" type="text" value="Filter..."/>	<input style="background-color: #f0f0f0;" type="text" value="Q Filter..."/>	<input style="background-color: #f0f0f0;" type="text" value="Q Filter..."/>	<input style="background-color: #f0f0f0;" type="text" value="Filter..."/>
AI / Machine Learning (0)					
Analytics (0)					
Blockchain (0)					
Databases (1)					
Db2-u3	Default	Dallas	Db2	Active	-
Developer tools (0)					
Logging and monitoring (0)					
Migration (0)					
Integration (0+)					
Internet of Things (0)					
Security (0)					
Mobile (0)					
Other (0+)					

Resource list /

Db2-u3

Active
Add tags

Details
Actions...

Manage
Getting started
Service credentials
Connections

Service credentials

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](#)

Search credentials...

New credential +

Key name	Date created
<div>Service credentials-1</div> <pre> { "connection": { "cli": { "arguments": ["-u", "vpz73631", "-p", "D7IrnX3aSj2gV8wo", "--ssl", "--sslCAFile", "1cbb1b6-3a1a-4d49-9262-3102a8f7a7c8", "--authenticationDatabase", "admin", "--host", "fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgut0lqde00.databases.appdomain.cloud:32731"] }, "bin": "db2", "certificate": { </pre>	2022-11-13 7:40 PM

Created Tables:

The screenshot shows the IBM Db2 on Cloud web interface. The 'Tables' tab is selected in the top navigation bar. On the left, under 'Schemas', a table lists the schema 'VPZ73631' of type 'User' with 6 tables. On the right, under 'Tables', a list of tables is shown, all belonging to the 'VPZ73631' schema. The tables are: PRODUCTS, REGISTER, REGISTER_RETAILER, THRESHOLD_VALUE, USERPRODUCTS, and USERS. Each table has a checkbox for selection and a 'Properties' link.

Name	Type	Tables
VPZ73631	User	6

Name	Schema	Properties
PRODUCTS	VPZ73631	...
REGISTER	VPZ73631	...
REGISTER_RETAILER	VPZ73631	...
THRESHOLD_VALUE	VPZ73631	...
USERPRODUCTS	VPZ73631	...
USERS	VPZ73631	...

Connecting IBM DB2 with Python:

The screenshot shows a Python script in a code editor. The script imports necessary modules (flask, flask_cors, ibm_db, urllib, sendgrid) and sets up a Flask application. It connects to an IBM DB2 database using the ibm_db module. The connection string is: "DATABASE=bludb;HOSTNAME=fbd88901-cbdb-4a4f-a32e-9822b9fb237b.clogj3sd0gtu0lqde00.databases.appdomain.cloud;PORT=32731;SECURITY=SSL;SSLServerCertificate=DigicertGlobalRootCA.crt;UID=vpz73631;PWD=D71rmXJaSj2gV8wo";. The script defines a home route and a signin route. The signin route handles POST requests, queries the database for a user, and either logs them in or returns an error.

```
1 from flask import request, redirect, url_for, render_template, session, json
2 from flask import Flask
3 from flask_cors import CORS
4 import ibm_db,urllib.parse
5 from sendgrid import mailto, checkstatus, getProductsBelowThValue
6 app=flask(__name__)
7 app.secret_key = "121212"
8 cors=CORS(app)
9 # =====
10 conn =ibm_db.connect("DATABASE=bludb;HOSTNAME=fbd88901-cbdb-4a4f-a32e-9822b9fb237b.clogj3sd0gtu0lqde00.databases.appdomain.cloud;PORT=32731;SECURITY=SSL;
11 SSLServerCertificate=DigicertGlobalRootCA.crt;UID=vpz73631;PWD=D71rmXJaSj2gV8wo";, '')
12
13 @app.route("/")
14 def home():
15     return render_template("home.html", title="Home")
16
17 @app.route("/signin", methods=('POST', 'GET'))
18 def signin():
19     error = None
20     if request.method == 'POST':
21         email = request.form['email']
22         password = request.form['password']
23
24         sql = "SELECT username FROM users WHERE password = '{}' AND email = '{}'.format(password, email)"
25         stmt = ibm_db.exec_immediate(conn, sql)
26         fetchUser = ibm_db.fetch_assoc(stmt)
27         if fetchUser == False:
28             error = "Incorrect Username/Password."
29
30         if error is None:
31             user = fetchUser["USERNAME"]
32             session['loggedIn'] = True
33             session['id'] = user
34             session['email'] = email
35             return redirect(url_for('.dashboard', username=user))
36         return render_template("signin.html", title='Sign In', error=error)
```