

CREATE IBM IDB2 AND CONNECT WITH PYTHON

Team ID	PNT2022TMID32918
Project Name	Project – Skill/Job Recommender

Welcome to Project! Delighted to... IBM IBM-39732-1662630478 Catalog - IBM Cloud

cloud.ibm.com/catalog

IBM Cloud Search resources and products... Catalog Manage Dhivya V's Account

Search the catalog... Sell on IBM Cloud Catalog settings

Recommended products (6)

- Compute (30)
- Containers (9)
- Networking (29)
- Storage (19)
- AI / Machine Learning (17)
- Analytics (10)
- Blockchain (1)
- Databases (28)
- Developer tools (25)
- Logging and monitoring (3)
- Migration (7)
- Integration (10)
- Internet of Things (1)
- Security (25)
- Mobile (1)

Analytics Engine
By IBM
Submit your Apache Spark applications as needed and customize the Spark runtimes to satisfy the requirements of your application.
Lite • Free • HIPAA Enabled • IAM-enabled • Service Endpoint Supported • IBM supported

AnonTech ViziVault Platform
By Anon Technology, Inc.
Manage personal information as-a-service safely, securely, and in compliance with data privacy regulations using ViziVault
Lite • Free • HIPAA Enabled • IAM-enabled • Third party supported

API Connect
By IBM
An enterprise-grade platform for creating, securing, managing, sharing, monetizing, and analyzing custom APIs located on-premises...
Lite • Free • EU Supported • IAM-enabled • IBM supported

App Configuration
By IBM
Centralized, in-flight configuration for web and mobile applications and distributed environments.
Lite • Free • IAM-enabled •

App Connect
By IBM
Connect your applications, automate tasks, and improve productivity

App ID
By IBM
User Authentication and User Profiles for your apps.
Lite • Free • EU Supported • Financial Services Validated • HIPAA Enabled •

Welcome to Project! Delighted to... IBM IBM-39732-1662630478 Catalog - IBM Cloud

cloud.ibm.com/catalog

IBM Cloud Search resources and products... Catalog Manage Dhivya V's Account

Search the catalog... Sell on IBM Cloud Catalog settings

Build mission critical web applications that offer high performance with scalability and security.
EU Supported • HIPAA Enabled • IAM-enabled • Service Endpoint Supported • IBM supported

DataStage
By IBM
Create ETL and data pipeline services for real-time, micro-batch, and batch data orchestration.
Lite • Free • IAM-enabled • IBM supported

Db2
By IBM
A fully managed, highly-performant relational data store running the enterprise-class Db2 database engine.
Lite • Free • EU Supported • HIPAA Enabled • IAM-enabled • IBM supported

Db2 Warehouse
By IBM
A fully managed, elastic, cloud data warehouse built for high-performance analytics and machine learning.
HIPAA Enabled • IAM-enabled • IBM supported

Dedicated Host for VPC
By IBM

Direct Link Connect
By IBM

Direct Link Connect on Classic
By IBM

https://cloud.ibm.com/catalog/services/db2

IBM Cloud Catalog page for Db2 service.

Create | About

Type: Service

Provider: IBM

Last updated: 11/10/2022

Category: Databases

Compliance: EU Supported, HIPAA Enabled, IAM-enabled

Location: Sydney, Frankfurt, London, Dallas, Sao Paulo, Toronto

Select a location: Dallas (us-south)

Select a pricing plan

Displayed prices do not include tax. Monthly prices shown are for country or location: [United States](#)

Plan	Features	Pricing
Lite	200 MB of data storage 5 simultaneous connections Shared multitenant system	Free

The Free plan provides a free Db2 service for development and evaluation. The plan has a set amount of limitations as shown. You can continue using the free plan for as long as needed, however, users are asked to re-extend their free account every 90 days by email. If you do not re-extend, your free account is cleaned out a further 90 days later. This helps provide free resources for everyone.

Lite plan services are deleted after 30 days of inactivity.

Summary

Db2 **Free**

Location: Dallas
Plan: Lite
Service name: Db2-ur
Resource group: Default

☒ I have read and agree to the following license agreements:
[Terms](#)

[Create](#)

[Add to estimate](#)

IBM Cloud Resource list page.

Search resources and products...

Name	Group	Location	Product	Status	Tags
Filter by name or IP address... Filter by group or org... Filter... Filter... Filter... Filter...					
Storage (0)					
AI / Machine Learning (0)					
Analytics (0)					
Blockchain (0)					
Databases (1)					
Db2-ur	Default	Dallas	Db2	Provision in pro...	
Developer tools (0)					
Logging and monitoring (0)					
Migration (0)					
Integration (0)					
Internet of Things (0)					
Security (0)					

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

Find schemas or tables Refresh

Schemas

<input checked="" type="checkbox"/>	Name	Type	Tables
<input checked="" type="checkbox"/>	HKD04084	User	2

Total: 1, selected: 1

Tables

<input type="checkbox"/>	Name	Schema	Properties
<input type="checkbox"/>	JOB	HKD04084	...
<input type="checkbox"/>	USERS	HKD04084	...

Total: 2, selected: 0

76°F Partly cloudy Search the web 10:08 PM 11/17/2022

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

Find schemas or tables Refresh

Tables

<input type="checkbox"/>	Name	Schema	Properties
<input type="checkbox"/>	JOB	HKD04084	...
<input type="checkbox"/>	USERS	HKD04084	...

Total: 2, selected: 0

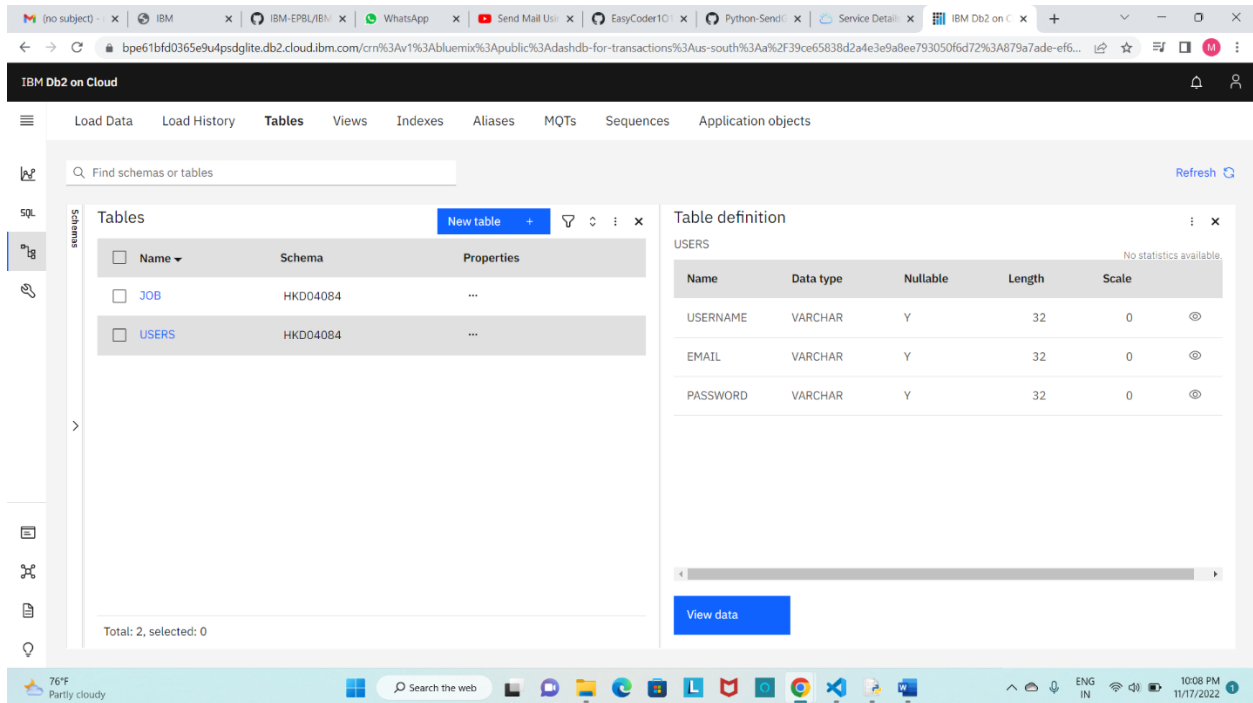
Table definition

JOB No statistics available.

Name	Data type	Nullable	Length	Scale
USERNAME	VARCHAR	Y	32	0
EMAIL	VARCHAR	Y	32	0
QUALIFICATION	VARCHAR	Y	32	0
SKILLS	VARCHAR	Y	32	0
JOBS	VARCHAR	Y	32	0

View data

76°F Partly cloudy Search the web 10:08 PM 11/17/2022



App.py:

```
from flask import Flask, render_template, request, redirect, url_for, session
```

```
import ibm_db
```

```
import re
```

```
app = Flask(__name__)
```

```
app.secret_key = 'a'
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-
a6c3e6b4b907.c3n41cmd0nqn timer 39u98g.databases.appdomain.cloud;PORT=30756;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=hkd04084;PWD=Bwjcq4p5AL1sL0FQ","")
```

```
@app.route('/')
```

```
def homer():
```

```
    return render_template('login.html')
```

```
@app.route('/login', methods=['GET', 'POST'])
```

```
def login():
```

```
    global userid
```

```
    msg = "
```

```

if request.method == 'POST' :
    username = request.form['username']
    password = request.form['password']
    sql = "SELECT * FROM users WHERE username =? AND password=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,username)
    ibm_db.bind_param(stmt,2,password)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print (account)
    if account:
        session['loggedin'] = True
        session['id'] = account['USERNAME']
        userid= account['USERNAME']
        session['username'] = account['USERNAME']
        msg = 'Logged in successfully !'
        msg = 'Logged in successfully !'
        return render_template('dashboard.html', msg = msg)
    else:
        msg = 'Incorrect username / password !'
    return render_template('login.html', msg = msg)
@app.route('/register', methods=['GET', 'POST'])
def registet():
    msg = ""
    if request.method == 'POST' :
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']

```

```

sql = "SELECT * FROM users WHERE username =?"

stmt = ibm_db.prepare(conn, sql)

ibm_db.bind_param(stmt,1,username)

ibm_db.execute(stmt)

account = ibm_db.fetch_assoc(stmt)

print(account)

if account:

    msg = 'Account already exists !'

elif not re.match(r'^@]+@[^@]+\.[^@]+', email):

    msg = 'Invalid email address !'

elif not re.match(r'[A-Za-z0-9]+', username):

    msg = 'name must contain only characters and numbers !'

else:

    insert_sql = "INSERT INTO  users VALUES (?, ?, ?)"

    prep_stmt = ibm_db.prepare(conn, insert_sql)

    ibm_db.bind_param(prepare_stmt, 1, username)

    ibm_db.bind_param(prepare_stmt, 2, email)

    ibm_db.bind_param(prepare_stmt, 3, password)

    ibm_db.execute(prepare_stmt)

    msg = 'You have successfully registered !'

elif request.method == 'POST':

    msg = 'Please fill out the form !'

    return render_template('register.html', msg = msg)

@app.route('/dashboard')

def dash():

    return render_template('dashboard.html')

@app.route('/apply',methods=['GET', 'POST'])

def apply():

    msg = "

```

```

if request.method == 'POST' :

    username = request.form['username']

    email = request.form['email']

    qualification= request.form['qualification']

    skills = request.form['skills']

    jobs = request.form['s']

    sql = "SELECT * FROM users WHERE username =?"

    stmt = ibm_db.prepare(conn, sql)

    ibm_db.bind_param(stmt,1,username)

    ibm_db.execute(stmt)

    account = ibm_db.fetch_assoc(stmt)

    print(account)

    if account:

        msg = 'there is only 1 job position! for you'

        return render_template('apply.html', msg = msg)

    insert_sql = "INSERT INTO job VALUES (?, ?, ?, ?, ?)"

    prep_stmt = ibm_db.prepare(conn, insert_sql)

    ibm_db.bind_param(prepare_stmt, 1, username)

    ibm_db.bind_param(prepare_stmt, 2, email)

    ibm_db.bind_param(prepare_stmt, 3, qualification)

    ibm_db.bind_param(prepare_stmt, 4, skills)

    ibm_db.bind_param(prepare_stmt, 5, jobs)

    ibm_db.execute(prepare_stmt)

    msg = 'You have successfully applied for job !'

    session['loggedin'] = True

    TEXT = "Hello sandeep,a new appliaction for job position" +jobs+"is requested"

    #sendmail(TEXT,"sandeep@thesmartbridge.com")

    sendgridmail("sandeep@thesmartbridge.com",TEXT)

```

```

elif request.method == 'POST':

    msg = 'Please fill out the form !'

    return render_template('apply.html', msg = msg)
@app.route('/display')
def display():

    print(session["username"],session['id'])

    cursor = mysql.connection.cursor()

    cursor.execute('SELECT * FROM job WHERE userid = % s', (session['id'],))

    account = cursor.fetchone()

    print("accountdisplay",account)

    return render_template('display.html',account = account)
@app.route('/logout')
def logout():

    session.pop('loggedin', None)

    session.pop('id', None)

    session.pop('username', None)

    return render_template('home.html')
if __name__ == '__main__':

    app.run(host='0.0.0.0')

```