

Project Development Phase
Delivery of Sprint-3

Date	12 November 2022
Team ID	PNT2022TMID24036
Project Name	Inventory Management System For Retailer

Creating IBM Db2 database and chat-box using IBM Watson Assistant:

Creating IBM Db2 Database and Connecting it:

Step 1:

Go to IBM cloud resource list and click on database.

Step 2:

Create a database and use the service credentials on your python flask to connect to IBM Db2 database services.

Step 3:

Click on 'Go to UI' and click 'Data' on the left side.

Step 4:

Click tables and select the name of your database.

Step 5:

Create new table according to the database you need.

Step 6:

Verify it is working.

Creating Chat-Box using IBM Watson Assistant:

Step 1:

Go to IBM cloud resource list and click on IBM Watson Assistant.

Step 2:

Click on launch Watson Assistant.

Step 3:

Build your virtual assistant.

Step 4:

Add the script to your python file.

Step 5:

Verify on our application page.

IBM Db2 Database Output: IBM Resource List Page:

IBM Cloud

Search resources and products...

Catalog

Manage

Sheik Amanullah's Account

Navigation Menu

Resource list

Create resource

Name	Group	Location	Product	Status	Tags
Filter by name or IP address...	Filter by group or org...	Filter...	Filter...	Filter...	Filter...
Compute (0)					
Containers (2+)					
Networking (0)					
Storage (1)					
AI / Machine Learning (1+)					
Analytics (0)					
Blockchain (0)					
Databases (1)					
Db2-yc	Default	London	Db2	Active	
Developer tools (4+)					
Logging and monitoring (2+)					
Migration (0)					
Integration (0+)					
Internet of Things (0)					
Security (0)					
Mobile (0)					

Db2 Service Credentials:

	IBM Cloud	Search resources and topics...	Catalog	Sheik Amanullah's Account			
	▼	8c151035-82fc-4719-85d2-f7fa4b2560c			2022-11-04 2:56 PM		
	▲	kube-157b5e3aa32748fe9a0451581bcffd5.cdn0jus10eg28bp5jeog.default			2022-11-22 11:40 PM		
		<pre>{ "connection": { "cli": { "arguments": ["-u", "qpm23984", "-p", "4WHC1EebOv9DkvHF", "--ssl", "--sslCAFile", "1dd14d9c-1b52-4f63-a696-53ecba28771d", "--authenticationDatabase", "admin", "--host", "3883e764-1bf5-4afe-be8c-fa31c41761d2.bs2io9l08kqb1odl9g.databases.appdomain.cloud:31498"], "bin": "db2", "certificate": { "certificate_base64": "LS0tLSU1cURudjRlODRvJUSUZjOUFURSotLS0tckJ3SURVbnU0dWxkZmFSc0Y3SUZBVTI3dGVwcy9vcyUENSRjFweHFs3vgHKRWpbDbVDWRpRUWblp3aHzjKTRERuMQWFlD6dghQNNWQH9HQTFRVFUFS3dRTVUpUoSUoc21cVet3RVJoZEEdgaVIYTexjekHfRncweU1EOTNRFF3TnpMvpMnAlphRncwek1EOTNRN13TnpM9lQwmF0JR4SERBYUJnlIZCQ01NRT8Bz01RTokR1ckzxkkNCrVLVMhhZ0Uz6clPYTXdz8vPUtWeR0NTcdUTSMiZRFFFQKFVRUFBNELCRHD6d2dnRUTsbelCVQFeb2FNQGSQdGoEXZUVUtwR3kQ0TB8amRX0nMANVBjJTDNyRSjtW1R3K2d1RuDSUJSJ0UBV23oAvY1G1GRK9ryYfnO6GvSsrOXJ3QeoXadRZOpSmllU21c1f4WTMB8c3BqeGRFVEZKHEScnJhMGU2Vm4MM42T1LJL0ZHsn1lQ3htTG5GMUTFQW9bhHYawDM2CnhDTBFcxwR7LfzFnPMTRGeU8YRkRajckK1AR6K4vy9XMVPvdvhMNgwzZXVLZVCeTruzmh3VkySVC3aUMkgbGP23R1Nh3ZTDVhbvpKOUDsYwtSn31cnpNRFFQLzVUYNR1UUIYdeEl0dTBR SVRFZHIIESFWYZGZRBDBHyZloZAo3Mp29jdngv2UJ3vCP9uRHNS0TJNMCo2SkfZmkPNb1pdfBTN3y2a2d1UVInIbAuVjWkpdpUPvZNOYI9Gr6PyScmY2a2BZ9tCOUfHaVgc0JlJW0JlBHQtFVZER nhdvdc0JP2m2Z2RmP9d038mx0cmMSD012DuB2Z3nT1VPSNRUEQVdrnQ1R2az2Z2RmP9d038mx0cmMSD012DuB2Z3nT1Z1ZUKCoV4RUJUQRBUJyTUExmpwD3PH011MBRRRNDd1VB0TBJOKFR7kdgvEVNUtZ1EtZYMKZnd03duIdtkpva1YSUNkkaTKzBEVFKNUdy1ClVOZ3NmhNmbNldxcHltAtmRkhjchZ1Vmp0vHRHT2ktNUM2wlZsRnyXpc1cu9zdfB5bk3J4b1Nac1Se0dk6dtKKvjbmRUGuxCE1tznZ8SxxkV3c4UEJTzG31Tk1Hgdm4S2lwT0e0SV0dBQ1ZFrxXVgddehDKJTFxBZnpVUX1lDUVdgppc1lpV2smTWXNGks5EZ452dTWUvQWofXVSls1R dXB2R2RPV1UgeAV8ab03aEVRbmZPV2Z1TTThD088xNWFCZLRGQ2s0Q0pDumR4MISU284V3o1Z3Mc2ncyYRKFDQJ1zy8NYEFdZnzTZT2UDNNINXkFRHJHBmpWgASVXsq2d4KT1RxwXROaVBhaHP uTHJrl6rfamZLMx2vQMLTmfMYNUdqajhqalpnBzRiZ84KL58tLS1FTkQG0vSVeIGSNBVUUtLS8tLQo=", "name": "1dd14d9c-1b52-4f63-a696-53ecba28771d" }, "composed": ["db2 -u qpm23984 -p 4WHC1EebOv9DkvHF --ssl --sslCAFile 1dd14d9c-1b52-4f63-a696-53ecba28771d --authenticationDatabase admin --host 3883e764-1bf5-4afe-be8c-fa31c41761d2.bs2io9l08kqb1odl9g.databases.appdomain.cloud:31498"], "environment": {}, "type": "cli" } } }</pre>					

Linking our python flask application with IBM Db2:

```
app.py > delprod
1  from __future__ import print_function
2  import ibm_db
3  from flask import Flask, redirect, render_template, request, session, url_for
4  import re
5  import sqlite3 as sql
6  import os
7
8
9
10 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90108kqb1od81cg.databases.appdomain.cloud;PORT=31498;SECURITY=SSL;SSLServerCertificate=bludb.pem;SSLCompression=DISABLED;UID=apikey;PWD=apikey")
11
12
13 app=Flask(__name__)
14 app.secret_key = 'asdfghjklzxcvbnm'
15
16 @app.route('/main')
17 @app.route('/')
18 def main():
19     return render_template('main.html')
20
21 @app.route('/brands',methods = ['POST', 'GET'])
22 def brands():
23     if request.method == 'POST':
24         name = request.form['name']
```

Going to IBM Db2 database :

IBM Db2 on Cloud

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

Find schemas or tables Refresh

SQL Schemas

Tables

Name	Schema	Properties
BRANDS	QPM23984	...
PRODUCTS	QPM23984	...
REGISTER	QPM23984	...
STORES	QPM23984	...

Total: 4, selected: 0

Table definition

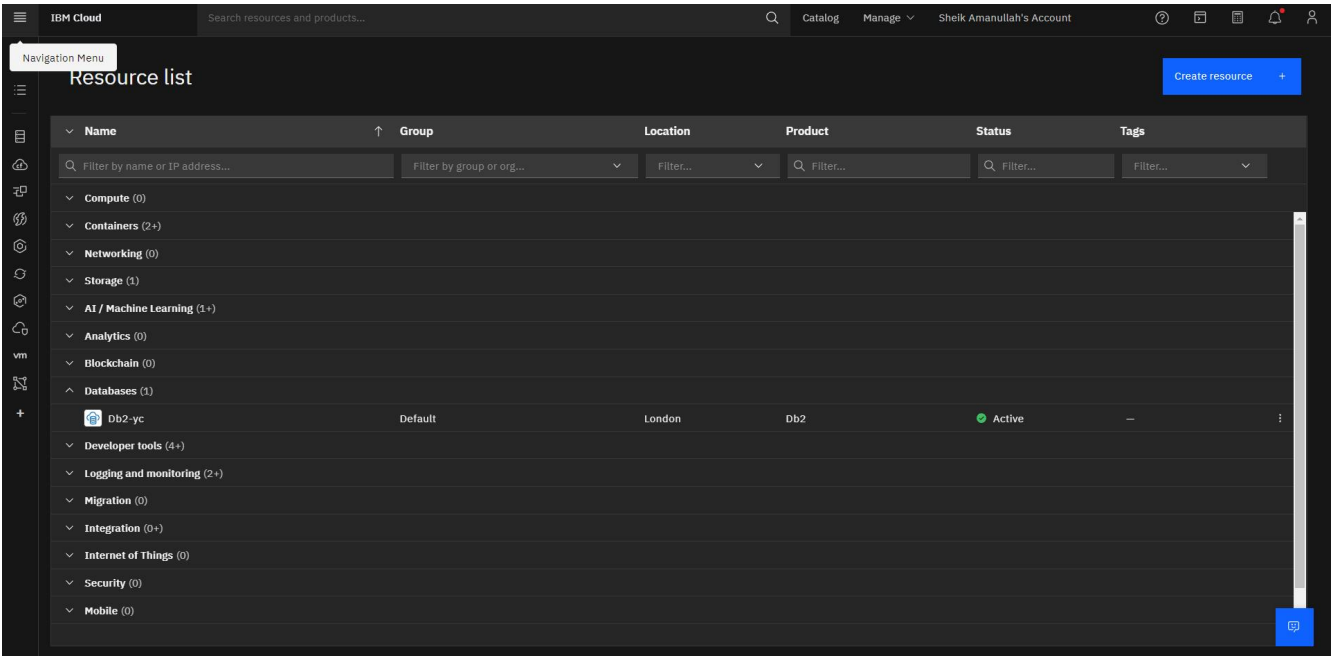
Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	32	0
STATUS	VARCHAR	Y	32	0

Approximate 3 rows (32.0 KB)
Updated on 2022-11-24 10:15:46

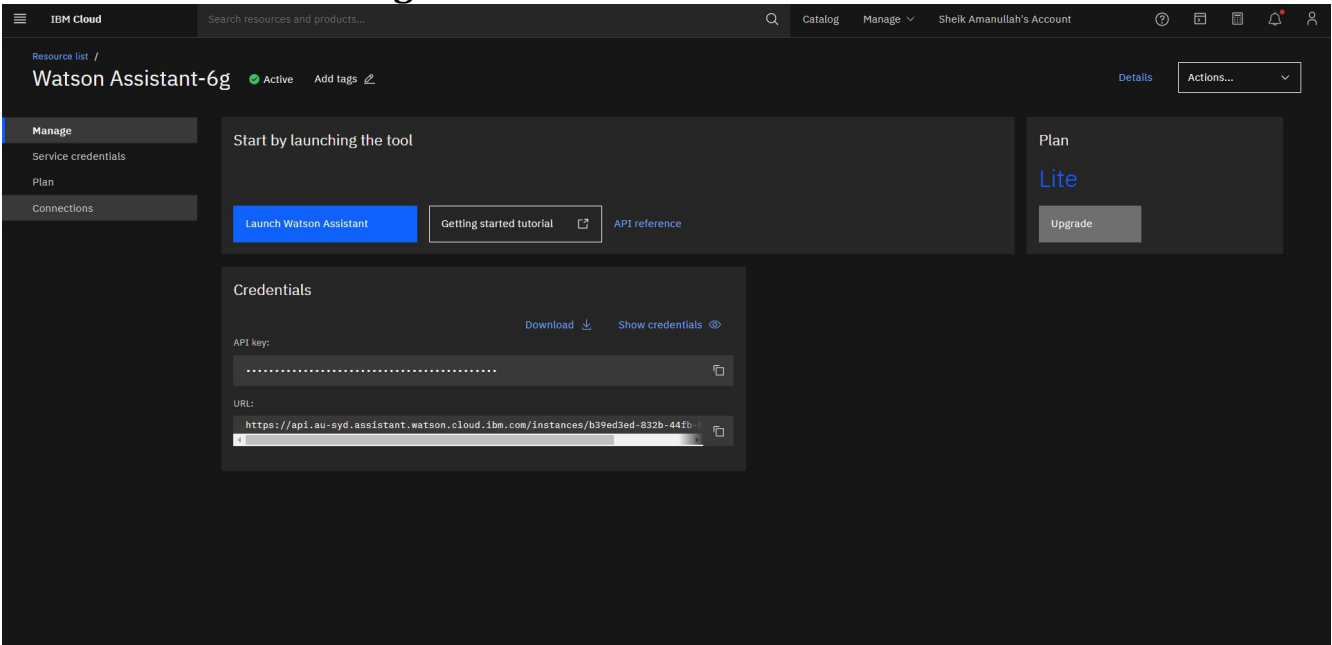
View data

Chat-Box Output:

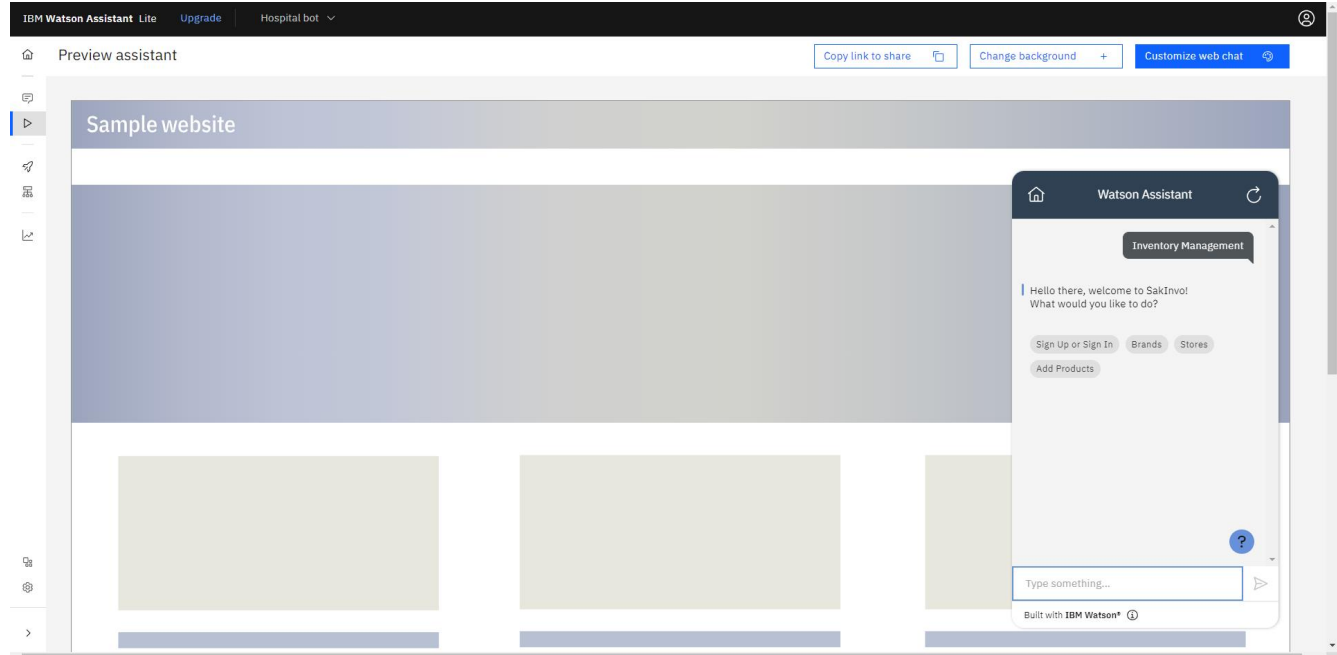
IBM Resource List Page:



IBM Watson Service Page on IBM Cloud:



Launch Watson Assistant and create a virtual chat-box:



Embed it on your python file:

```
<li class="nav-item">
  <a class="nav-link collapsed" href="/main">
    <i class="bi bi-box-arrow-in-left"></i>
    <span>Logout</span>
  </a>
</li><!-- End Login Page Nav -->
</ul>

</aside><!-- End Sidebar-->

<script>
window.watsonAssistantChatOptions = {
  integrationID: "ae5f7070-e140-4fd9-ae54-a71eadcfc36d", // The ID of this integration.
  region: "au-syd", // The region your integration is hosted in.
  serviceInstanceID: "b39ed3ed-832b-44fb-bd79-46948f50472d", // The ID of your service instance.
  onload: function(instance) { instance.render(); }
};
setTimeout(function(){
  const t=document.createElement('script');
  t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssis
  document.head.appendChild(t);
});
</script>

{% block brand %}
{%endblock%}
{% block stores %}
{%endblock%}
```

Verifying on our page:

 SakInvo

[signin](#)

Inventory management software designed for Retailers

Manage orders.Track inventory.Handle GST billing.
One inventory management software to
run all your inventory operations.

[Get started](#)

