

Project Development Phase
Delivery of Sprint-3

Date	12 November 2022
Team ID	PNT2022TMID24001
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...

Q

Catalog

Manage

Mohamed Yaseen's Account

?

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...
Containers (2)					
Networking (0)					
Storage (1)					
AI / Machine Learning (1)					
Analytics (0)					
Blockchain (0)					
Databases (1)					
ibm-project-db2oncloud-1669141780805	Default	London	Db2	Active	
Developer tools (3+)					
Logging and monitoring (2+)					
Migration (0)					
Integration (0+)					
Internet of Things (0)					
Security (0)					
Mobile (0)					
Other (0+)					

Db2 Service Credentials:

IBM Cloud

Search resources and products...

Q

Catalog

Manage

Mohamed Yaseen's Account

?

Resource list /

ibm-project-db2oncloud-1669141780805

Active

Add tags

Details

Actions...

Manage

Getting started

Service credentials

Connections

Service credentials

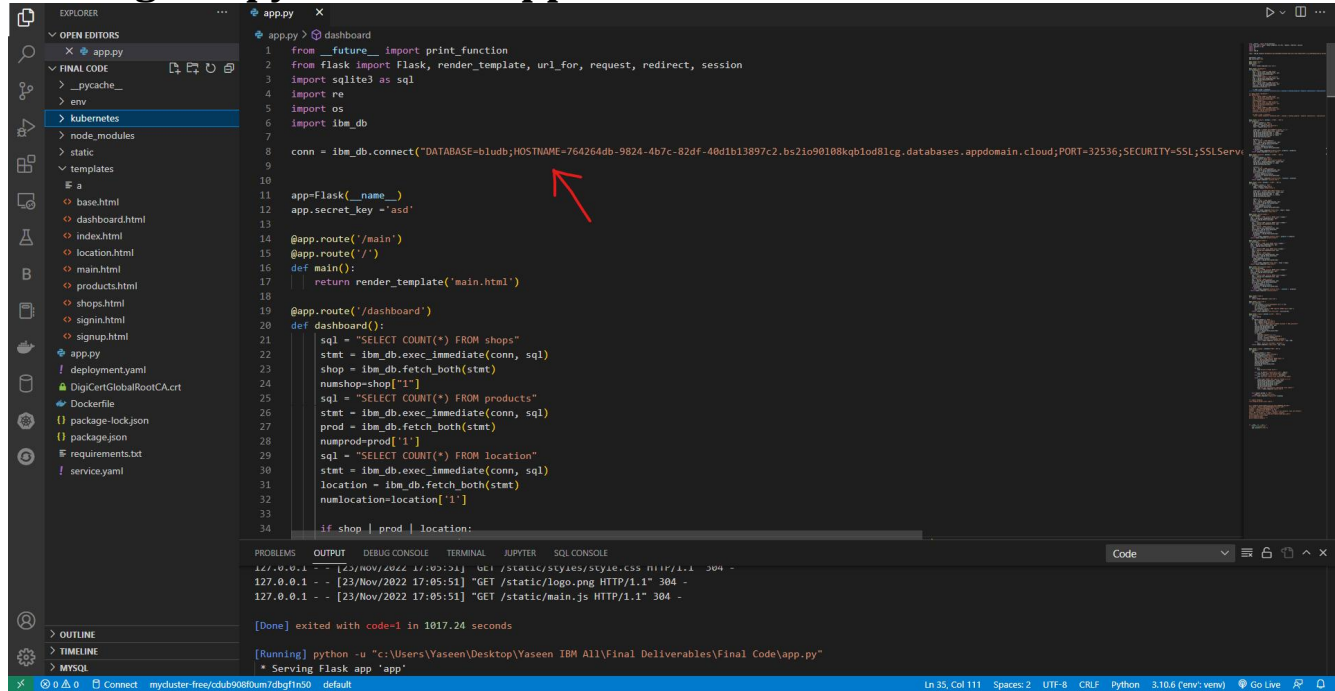
You can generate a new set of credentials for cases where you want to manually connect an app or external consumer to an IBM Cloud service. [Learn more](#)

Search credentials...

New credential +

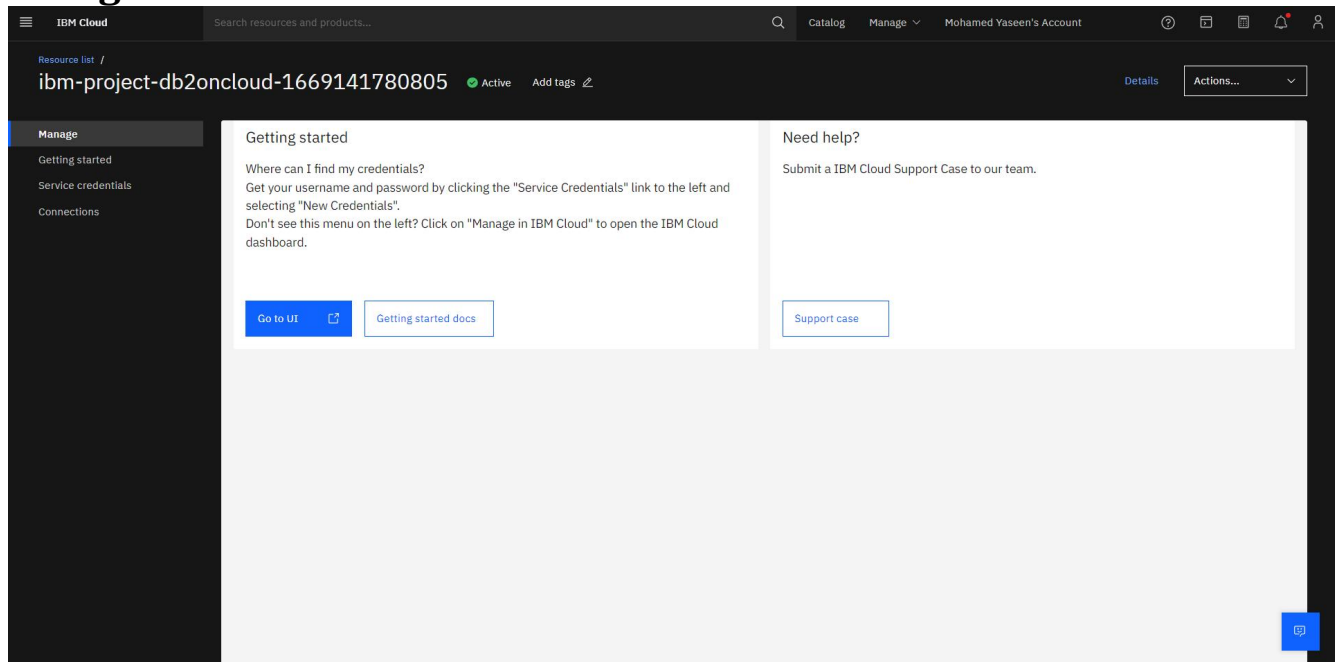
Key name	Date created
dc2e89f4-bae6-487a-bbe7-1d855e286bac	2022-11-23 12:00 AM

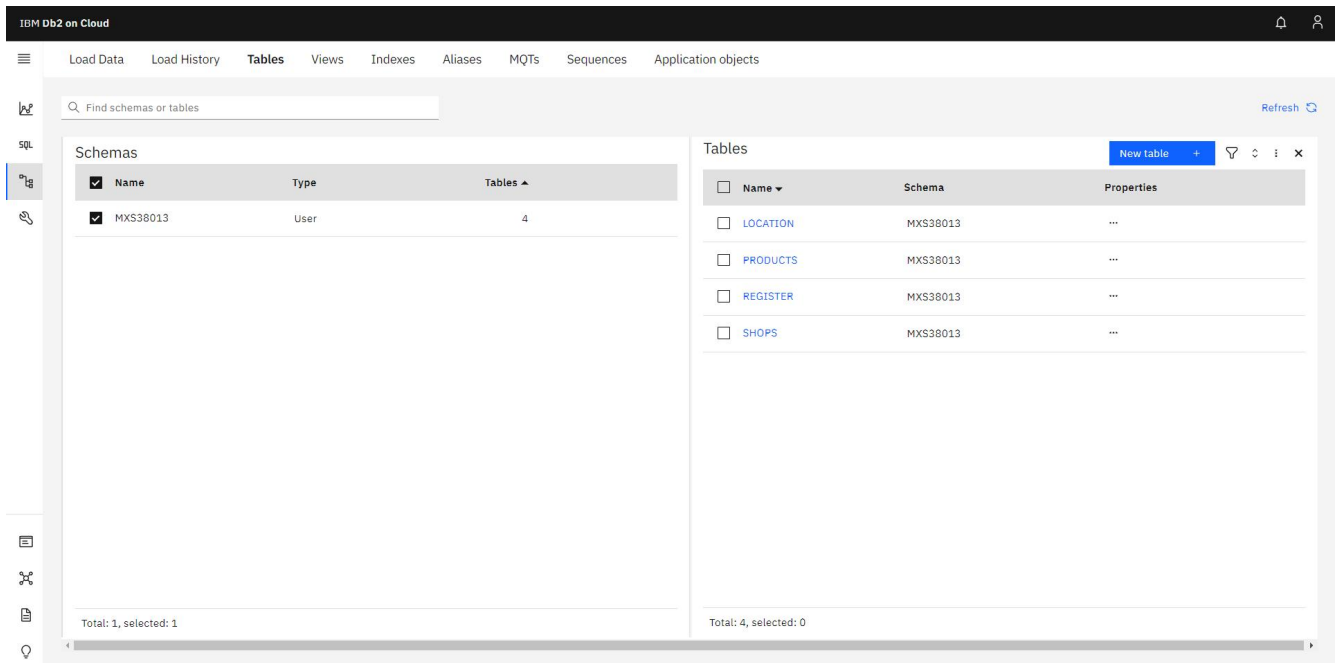
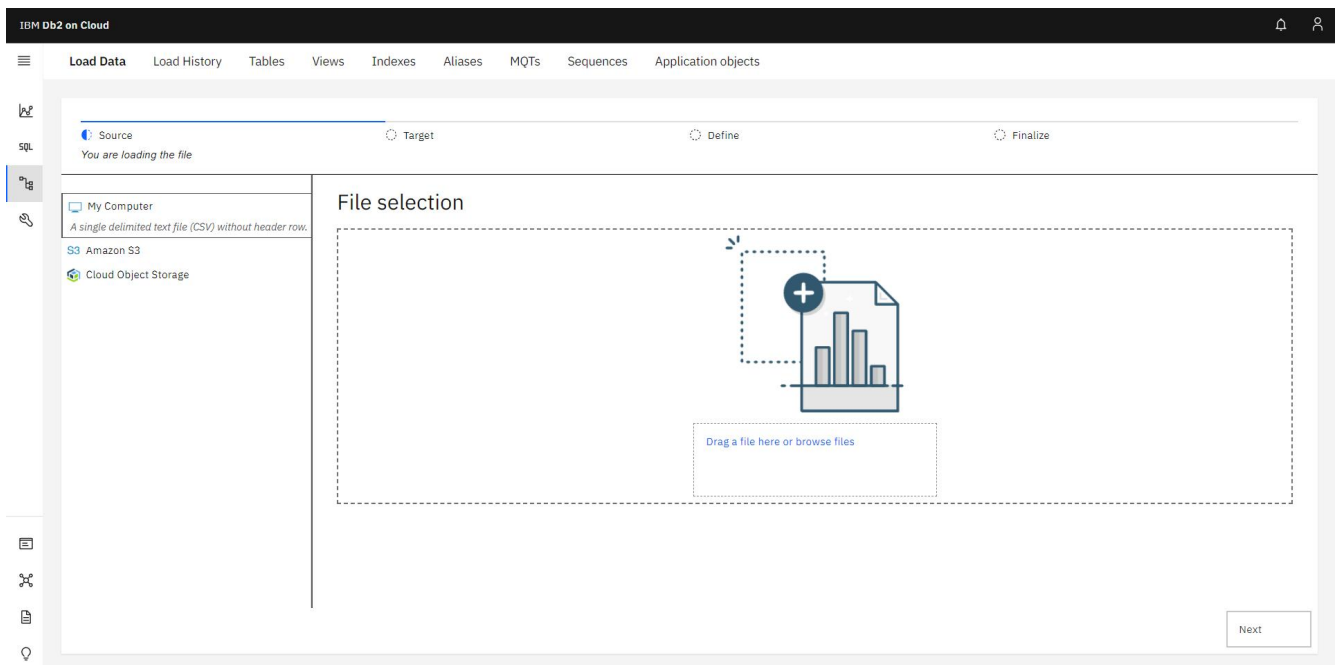
Linking our python flask application with IBM Db2:



```
1 from __future__ import print_function
2 from flask import Flask, render_template, url_for, request, redirect, session
3 import sqlite3 as sql
4 import re
5 import os
6 import ibm_db
7
8 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=764264db-9824-4b7c-82df-48d1b13897c2.bs2io90108kqblod8lcg_databases.appdomain.cloud;PORT=32536;SECURITY=SSL;SSLServerCertificate=ca.crt", "", "")
9
10 app=Flask(__name__)
11 app.secret_key = 'asd'
12
13 @app.route('/')
14 @app.route('/main')
15 @app.route('/login')
16 def main():
17     return render_template('main.html')
18
19 @app.route('/dashboard')
20 def dashboard():
21     sql = "SELECT COUNT(*) FROM shops"
22     stmt = ibm_db.exec_immediate(conn, sql)
23     shop = ibm_db.fetch_both(stmt)
24     numshop=shop[1]
25     sql = "SELECT COUNT(*) FROM products"
26     stmt = ibm_db.exec_immediate(conn, sql)
27     prod = ibm_db.fetch_both(stmt)
28     numprod=prod[1]
29     sql = "SELECT COUNT(*) FROM location"
30     stmt = ibm_db.exec_immediate(conn, sql)
31     location = ibm_db.fetch_both(stmt)
32     numlocation=location[1]
33
34     if shop | prod | location:
```

Going to IBM Db2 database UI:





IBM Db2 on Cloud

Load Data

Load History

Tables

Views

Indexes

Aliases

MQTs

Sequences

Application objects

SQL

MXS38013.PRODUCTS

Back

Export to CSV

NAME	QUANTITY	COST
Shirt	1	500

IBM Db2 on Cloud

Load Data

Load History

Tables

Views

Indexes

Aliases

MQTs

Sequences

Application objects

Find schemas or tables

Refresh

Schemas

Tables

New table

Name	Schema	Properties
<input type="checkbox"/> LOCATION	MXS38013	...
<input checked="" type="checkbox"/> PRODUCTS	MXS38013	...
<input type="checkbox"/> REGISTER	MXS38013	...
<input type="checkbox"/> SHOPS	MXS38013	...

Total: 4, selected: 1

Table definition

PRODUCTS

No statistics available.

Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	32	0
QUANTITY	INTEGER	Y		0
COST	INTEGER	Y		0

View data

Chat-Box Output:

IBM Resource List Page:

The screenshot shows the IBM Cloud Resource List page. The top navigation bar includes the IBM Cloud logo, a search bar, and links to Catalog, Manage, and Mohamed Yaseen's Account. The main content area is titled "Resource list" and features a "Create resource" button. Below the title is a table with columns: Name, Group, Location, Product, Status, and Tags. The table lists various resource categories such as Compute, Containers, Networking, Storage, AI / Machine Learning, Analytics, Blockchain, Databases, Developer tools, Logging and monitoring, Migration, Integration, Internet of Things, Security, and Mobile. The "AI / Machine Learning" category is expanded, showing a single resource: "Watson Assistant-ro". This resource is located in Tokyo, has a status of "Active", and is associated with the "Watson Assistant" product. The left sidebar contains icons for different resource types, and the bottom right corner has a chat icon.

Name	Group	Location	Product	Status	Tags
▼ Compute (0)					
▼ Containers (2+)					
▼ Networking (0)					
▼ Storage (1+)					
▲ AI / Machine Learning (1)					
Watson Assistant-ro	Default	Tokyo	Watson Assistant	Active	—
▼ Analytics (0)					
▼ Blockchain (0)					
▼ Databases (1)					
▼ Developer tools (3+)					
▼ Logging and monitoring (2+)					
▼ Migration (0)					
▼ Integration (0+)					
▼ Internet of Things (0)					
▼ Security (0)					
▼ Mobile (0)					

IBM Watson Service Page on IBM Cloud:

The screenshot shows the IBM Watson Assistant service page. The top navigation bar is identical to the previous page. The main content area is titled "Watson Assistant-ro" and shows the resource is "Active". Below the title is a "Manage" section with tabs for Service credentials, Plan, and Connections. The "Service credentials" tab is selected, displaying a "Start by launching the tool" section with a "Launch Watson Assistant" button, a "Getting started tutorial" link, and an "API reference" link. The "Plan" section shows the current plan is "Lite" with an "Upgrade" button. The "Credentials" section displays the API key and URL, with a "Download" button and a "Show credentials" link. The API key is masked with dots, and the URL is "https://api.jp-tok.assistant.watson.cloud.ibm.com/instances/9a508ac7-1766-402a-b00d-22feef879...". A note at the bottom states "View all credentials in the Service credentials tab."

Resource list / **Watson Assistant-ro** Active [Add tags](#) [Details](#) [Actions...](#)

Manage

- Service credentials
- Plan
- Connections

Start by launching the tool

[Launch Watson Assistant](#) [Getting started tutorial](#) [API reference](#)

Plan

Lite

[Upgrade](#)

Credentials

[Download](#) [Show credentials](#)

API key:

URL:

View all credentials in the **Service credentials** tab.

Launch Watson Assistant and create a virtual chat-box:

The screenshot displays the IBM Watson Assistant interface for configuring a chatbot named 'Inventory Management'. The top navigation bar includes 'IBM Watson Assistant', 'Lite', 'Upgrade', and 'Hospital Bot'. The main content area is divided into three sections:

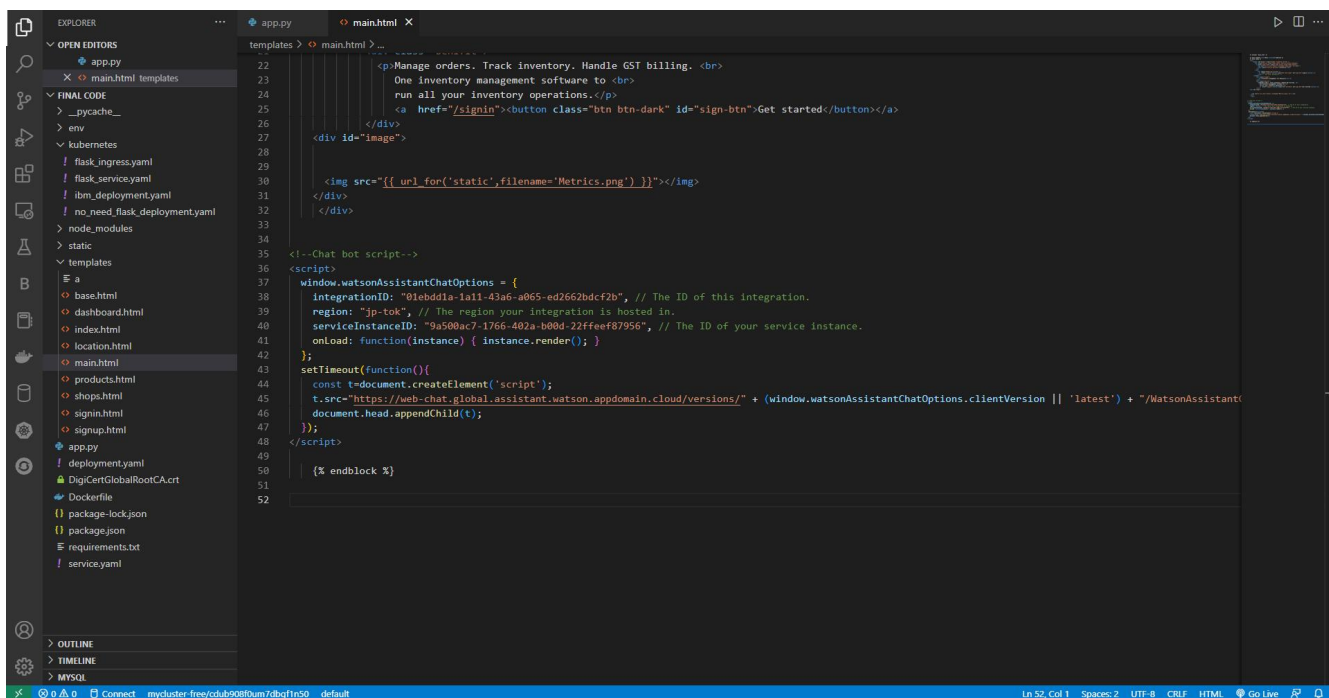
- Conversation steps:** A list of steps for the chatbot's conversation flow. Step 1 is 'Hello there! Welcome to Inventory Management System. What can I do to assist you today?' with buttons for 'Shops', 'Location', and '+ 2'. Subsequent steps include 'Sign up/in', 'Welcome to our inventory management system...', 'Shops', 'Products', and 'How to use'.
- Customer starts with:** A section for defining phrases that trigger specific actions. It includes a text input field for 'Enter a phrase' and a list of phrases: 'Store', 'How to use', 'Help', 'I need help', and 'Inventory Management'.
- Preview:** A section showing a simulated chat conversation. It starts with 'Welcome, how can I assist you?' and 'inventory management' as input. The chatbot responds with 'Hello there! Welcome to Inventory Management System. What can I do to assist you today?' and provides instructions for registration.

Embed it on your python file:

The screenshot displays the IBM Watson Assistant interface for embedding the chatbot into a website. The top navigation bar includes 'IBM Watson Assistant', 'Lite', 'Upgrade', and 'Hospital Bot'. The main content area is divided into two sections:

- Web chat:** A section for configuring the chatbot's appearance and behavior. It includes a 'Draft' button and a 'Close' button.
- Embed:** A section for integrating the chatbot into a website. It includes a code block for embedding the chatbot and a 'Show more' button.

```
<script>
window.watsonAssistantChatOptions = {
  integrationID: "01ebdd1a-1a11-43a6-a065-ed2662bdcf2b", // The ID of this integration.
  region: "jp-tok", // The region your integration is hosted in.
  serviceInstanceID: "9a500ac7-1766-402a-b00d-22f1ee187956", // 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
  document.head.appendChild(t);
});
</script>
```



Verifying on our page:

