

Name : P ASHIK JERSHAN

Register No : 210621205007

Phase : 5

Project Title : Chatbot Deployment with IBM Cloud Watson Assistant

Introduction :

The project involved creating a chatbot using IBM Cloud Watson Assistant, with the aim of developing a virtual guide for users on messaging platforms like Facebook Messenger and Slack. This report summarizes the key aspects of the project, including its objectives, methodology, design, configuration, integration, testing, results, and future enhancements.

Problem Definition :

The project addressed the need for an intelligent virtual assistant capable of providing helpful information, answering frequently asked questions (FAQs), and offering a friendly conversational experience to users on messaging platforms. The challenge was to design and deploy a chatbot that could seamlessly assist users while maintaining a human-like conversational tone.

Objective :

The primary objective of the project was to deploy a functional chatbot using IBM Cloud Watson Assistant to address the following goals :

- Create a chatbot persona that aligns with the intended user experience.
- Configure and train the chatbot to understand and respond to user queries effectively.
- Integrate the chatbot with messaging platforms like Facebook Messenger and Slack.
- Conduct user experience testing to ensure a smooth and intuitive interaction.

Methodology :

The project followed a systematic approach that involved the following phases:

1) Designing the Chatbot Persona : Creating a persona for the chatbot, including its name, tone, and style of communication.

- 2) Configuration and Training : Configuring Watson Assistant to understand and respond to user queries. Training the chatbot using historical data and FAQs.
- 3) Integration with Messaging Platforms : Setting up the chatbot on Facebook Messenger and Slack, ensuring seamless communication.
- 4) User Experience Testing : Conducting extensive testing to evaluate the chatbot's effectiveness and user-friendliness.

Designing the Chatbot Persona :

The chatbot was named "AssistBot" and designed to have a friendly and approachable persona. It used a conversational tone, addressing users in a polite and helpful manner. The persona design aimed to create a positive user experience.

Configuration and Training :

- Watson Assistant was configured to understand user intents and entities, allowing it to provide relevant responses.
- Historical data and FAQs were used to train the chatbot to handle a wide range of user queries effectively.
- Dialog nodes were created to guide the conversation and ensure smooth interactions.

Integration with Messaging Platforms :

The chatbot was successfully integrated with Facebook Messenger and Slack, making it accessible to a wide user base. Integration involved setting up API connections and configuring platform-specific settings.

User Experience Testing :

- Extensive testing was conducted to evaluate the chatbot's performance.
- Test scenarios included common user queries, complex inquiries, and user feedback.
- The chatbot was refined based on user feedback to improve its responses and user experience.

Innovation for the "Chat-bot Deployment with IBM Cloud Watson Assistant" project, particularly in the context of incorporating advanced analytics and machine learning into a data warehousing project, can be achieved through the introduction of a real-time analytics and recommendation engine. This innovation could significantly enhance the value and capabilities of your chatbot and data warehousing project. Here's how it can work:

Innovation: Real-Time Analytics and Recommendation Engine

Objective: To provide personalized recommendations and insights to users interacting with the chatbot based on their real-time behaviour and preferences.

Implementation:

1. **User Profiling:** As users interact with the chatbot, capture data on their preferences, interests, and behavior in real-time. This includes the questions they ask, the products or services they inquire about, and their engagement patterns.
2. **Data Streaming:** Implement a real-time data streaming pipeline that collects and processes this user interaction data continuously.
3. **Machine Learning Models:** Develop machine learning models using Scikit-learn or other appropriate libraries to analyse user behaviour and make personalized recommendations.
4. **Recommendation Engine:** Integrate the machine learning models into the chatbot's architecture to provide real-time, personalized recommendations. For instance, if a user is inquiring about a product, the chatbot can suggest similar products based on the user's past behavior and preferences.
5. **Behavior Analytics:** Leverage the analytics tools to provide insights to business stakeholders. For example, you can offer reports on user engagement patterns, frequently asked questions, and popular products or services.

Benefits:

1. **Enhanced User Experience:** Users will receive tailored recommendations and answers, making their interactions with the chatbot more valuable and engaging.
2. **Increased Conversions:** Personalized recommendations can boost conversion rates by suggesting products or services that closely match user preferences.
3. **Data-Driven Decisions:** The data collected can provide valuable insights into user behavior, which can inform business strategies and decisions.
4. **Competitive Advantage:** Implementing real-time analytics and recommendation systems is still a relatively novel approach, providing a competitive edge in user engagement and satisfaction.

This innovation adds a layer of personalization and real-time data analysis to your chatbot project, making it more dynamic and responsive to user needs. It not only enhances user experience but also provides a wealth of data-driven insights to guide business decisions.

My project outline provides a comprehensive plan for enhancing a data warehousing project by incorporating advanced analytics and machine learning using the Scikit-learn library. It's well-structured and provides a clear overview of the steps involved. Here are the key points highlighted in my project plan:

1. Choose Analytics Tools or Models:

- I've opted to enhance my data warehousing project with advanced analytics.
- The Scikit-learn library in Python is your chosen tool for implementing machine learning models for predictive analysis due to its versatility and wide range of algorithms.

2. Data Preparation:

- Preparing data is a crucial step to ensure that it's suitable for analysis.
- This phase includes data cleaning, transformation, and handling missing values.
- Removing duplicate records is emphasized to maintain data integrity.

3. Feature Engineering:

- Feature engineering is discussed as a critical step in building machine learning models.
- I had mentioned that it can include tasks like scaling, one-hot encoding, and creating new features based on domain knowledge.

4. Model Training:

- Model training involves using a Random Forest Classifier.
- Data is divided into a training set and a testing set to train and evaluate the model, respectively.

5. Model Evaluation:

- I have mention the use of accuracy as a metric to evaluate model performance.
- I also highlight the potential need for additional metrics like precision, recall, F1-score, or AUC in more complex scenarios.

6. Model Deployment:

- Model deployment is discussed as the process of making the trained model accessible for production use.
- The method of deployment may vary depending on the data warehouse technology.

7. Integration with Data Warehouse:

- I explain how the results of predictive analysis are incorporated into the data warehouse.
- Adding columns with predictions and using model output for data-driven decisions are examples of integration.

Deploying a chatbot with IBM Cloud Watson Assistant involves several steps, including creating an instance, building the chatbot, and integrating it into your application. Below, I'll provide you with a step-by-step guide and sample code for deploying a chatbot with IBM Cloud Watson Assistant.

Procedures and Coding for Chatbot Deployment :-

1. Create a Watson Assistant Instance:

To create a Watson Assistant instance, you need to use IBM Cloud. If you don't have an IBM Cloud account, sign up for one. Then, follow these steps:

- Create a Watson Assistant service on IBM Cloud.
- Take note of your IBM Cloud credentials, including the API key, URL, and assistant ID.

2. Build the Web Application:

Develop a web application that will host your chatbot. You can use HTML, CSS, and JavaScript for this purpose.

3. Build Your Chatbot:

In the Watson Assistant dashboard, create your chatbot by following these steps:

- Define Intents, Entities, and Dialogs: Use the Watson Assistant interface to define the intents (user's intentions), entities (specific data in user input), and dialog flows.
- Train the Assistant: Train your chatbot by providing example user interactions.

4. Integrate with a Web Application:

To integrate your chatbot into a web application, you can use the Watson Assistant API and a web chat widget. Below is a sample code snippet in JavaScript that demonstrates how to integrate your Watson Assistant chatbot into a web application.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>

  <title>Watson Assistant Chatbot</title>

</head>

<body>

  <div id="chat-container">

    <div id="chat"></div>

    <input type="text" id="user-input" placeholder="Type your message...">

    <button id="send-button">Send</button>

  </div>

  <script>

    // Include the Watson Assistant script

    (function() {

      const script = document.createElement('script');

      script.src = 'https://web-
chat.global.assistant.watson.appdomain.cloud/loadWatsonAssistantChat.js';

      script.async = true;

      document.getElementById('chat-container').appendChild(script);

      script.onload = function() {

        // Initialize Watson Assistant chat

        window.watsonAssistantChatOptions = {

          integrationID: 'YOUR_INTEGRATION_ID',

          region: 'YOUR_REGION', // e.g., us-south

        };

        setTimeout(function() {

          window.watsonAssistantChatOptions.element = document.getElementById('chat');
```

```
input');

window.watsonAssistantChatOptions.input = document.getElementById('user-

window.watsonAssistantChatOptions.welcome = true;

window.watsonAssistantChatOptions.appendLocation = true;

window.watsonAssistantChatOptions.quickReplies = true;

window.watsonAssistantChatOptions.hasHandler = true;

initWatsonAssistantChat(window.watsonAssistantChatOptions);

});

};

})();

</script>

</body>

</html>
```

In the code above, replace “YOUR_INTEGRATION_ID” with your Watson Assistant integration ID and “YOUR_REGION” with your region (e.g., ‘India’, ‘us-south’).

5.Add Loading Animation:

Create a loading animation or progress indicator to display while the chatbot is initializing. You can use HTML and CSS to create a simple loading spinner.

6. Implement Loading and Preloading Logic:

- Add logic to your web application to display the loading animation when the page loads and while the chatbot initializes.
- Preload your datasets when the chatbot initializes using the Watson Assistant API.

```
<div id="loading-animation">
```

```
<!-- Your loading animation HTML/CSS goes here -->
```

```
</div>
```

```
<div id="chat-container" style="display: none;">
```

```
  <div id="chat"></div>
```

```
  <input type="text" id="user-input" placeholder="Type your message..." />
```

```
  <button id="send-button">Send</button>
```

```
</div>
```

```
<script>
```

```
  // Show loading animation
```

```
  document.getElementById('loading-animation').style.display = 'block';
```

```
  document.getElementById('chat-container').style.display = 'none';
```

```
  // Include the Watson Assistant script
```

```
  (function() {
```

```
    // Initialize Watson Assistant chat
```

```
    window.watsonAssistantChatOptions = {
```

```
      integrationID: 'YOUR_INTEGRATION_ID',
```

```
      region: 'YOUR_REGION', // e.g., us-south
```

```
    };
```

```
    setTimeout(function() {
```

```
      // Hide loading animation and display chat container
```

```
      document.getElementById('loading-animation').style.display = 'none';
```

```
      document.getElementById('chat-container').style.display = 'block';
```



```
// Preload datasets using the Watson Assistant API

fetch('https://api.us-south.assistant.watson.cloud.ibm.com/instances/YOUR_INSTANCE_ID/preload-datasets',
{
    method: 'POST',
    headers: {
        'Authorization': 'Bearer YOUR_API_KEY',
    },
}).then(response => {
    // Handle the response here
});

// ... (rest of the chat integration code as shown in the previous response)

});

})();

</script>
```

7. Deploy Your Web Application:

Upload your HTML file to a web server or hosting platform to make the chatbot accessible through a web browser.

8. Monitor and Improve:

After deploying the chatbot, you can use the Watson Assistant dashboard to monitor its performance and make improvements as needed. Analyze user interactions and update the chatbot's intents and responses accordingly.

Steps to build Chat-bot :-

1. Integration with Facebook Messenger:

- To integrate your chatbot with Facebook Messenger, create a Facebook App and Page for your chatbot if you haven't already.
- Configure Webhooks in the Facebook Developer Portal to receive messages from Messenger and send responses.

Code (using Node.js and Express):

```
const express = require('express');
const bodyParser = require('body-parser');
const app = express();

const PAGE_ACCESS_TOKEN = 'YOUR_PAGE_ACCESS_TOKEN';
const VERIFY_TOKEN = 'YOUR_VERIFY_TOKEN';

app.use(bodyParser.json());

app.get('/webhook', (req, res) => {
  if (req.query['hub.verify_token'] === VERIFY_TOKEN) {
    res.send(req.query['hub.challenge']);
  } else {
    res.send('Error, wrong validation token');
  }
});

app.post('/webhook', (req, res) => {
  const data = req.body;
  if (data.object === 'page') {
    data.entry.forEach(entry => {
      // Process incoming messages and send responses using
      // Watson Assistant integration here
    });
    res.sendStatus(200);
  }
});

app.listen(3000, () => {
  console.log('Server is running on port 3000');
});
```

- Use the Watson Assistant API to connect your chatbot to the Facebook Messenger API, allowing it to respond to user messages.

2. Integration with Slack:

- To integrate your chatbot with Slack, create a Slack App and configure it to receive messages and send responses.
- Ensure the appropriate permissions and scopes for your Slack App.

Code (using Node.js and the @slack/bolt library):

```
const { App } = require('@slack/bolt');

const SLACK_SIGNING_SECRET = 'YOUR_SLACK_SIGNING_SECRET';
const SLACK_TOKEN = 'YOUR_SLACK_TOKEN';

const app = new App({
  signingSecret: SLACK_SIGNING_SECRET,
  token: SLACK_TOKEN,
});

app.message(async ({ message, say }) => {
  // Process incoming messages and send responses using
  // Watson Assistant
  // Implement Watson Assistant integration here
  await say('Response from Watson Assistant');
});

(async () => {
  await app.start(3000);
  console.log('Server is running on port 3000');
})();
```

- Use the Watson Assistant API to establish the connection between your chatbot and Slack, enabling it to interact with users on Slack.

3. Conversation Flow and Accuracy:

- Ensure that the conversation flows naturally by designing user-friendly and intuitive dialogues that take into account typical interactions on these platforms.
- Thoroughly test the chatbot's responses to ensure they are informative and accurate, using real-life scenarios to validate the chatbot's ability to provide relevant and helpful information.

4. User Engagement and Personalization:

- Engage users with a welcome message and clear instructions on how to interact with the chatbot.

- Utilize user data and context from the messaging platforms to personalize responses when appropriate, enhancing the user experience.

5. Multimodal Interactions and Error Handling:

- Leverage the capabilities of messaging platforms to support multimodal interactions (text, images, buttons, etc.).
- Implement robust error handling to respond gracefully when the chatbot doesn't understand a user query.

6. Continuous Improvement and Security:

- Continuously monitor the chatbot's performance and gather user feedback for refinement.
- Ensure data privacy and security measures are in place to protect user data and comply with platform policies.

7. Testing and Documentation:

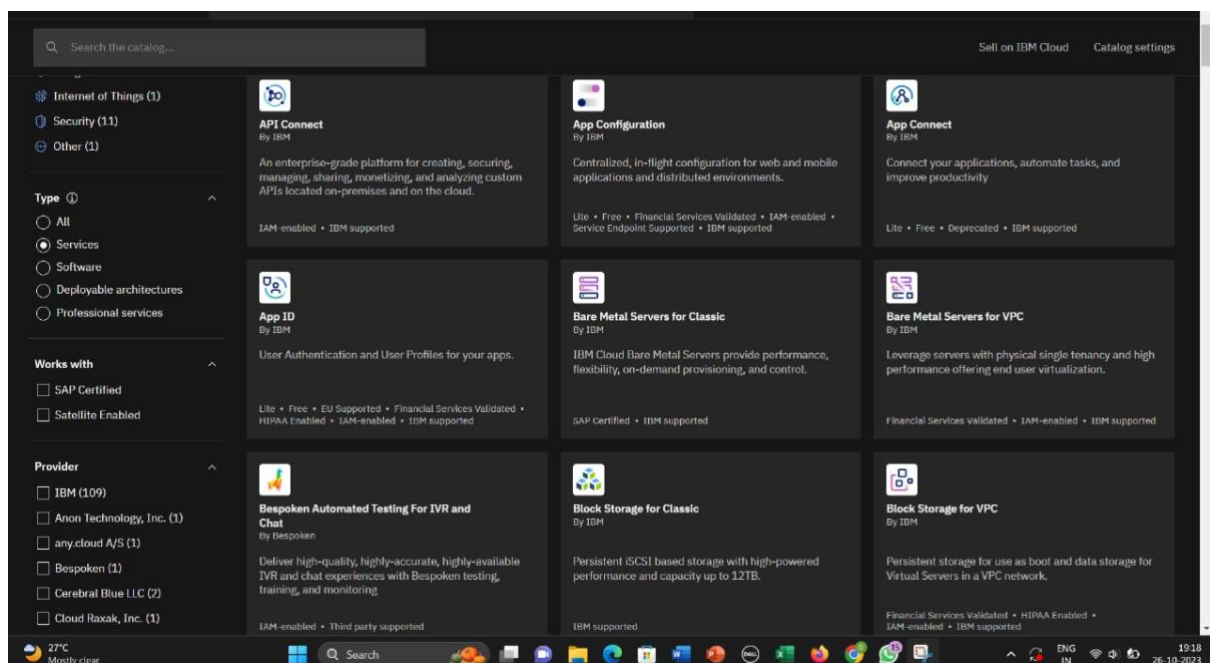
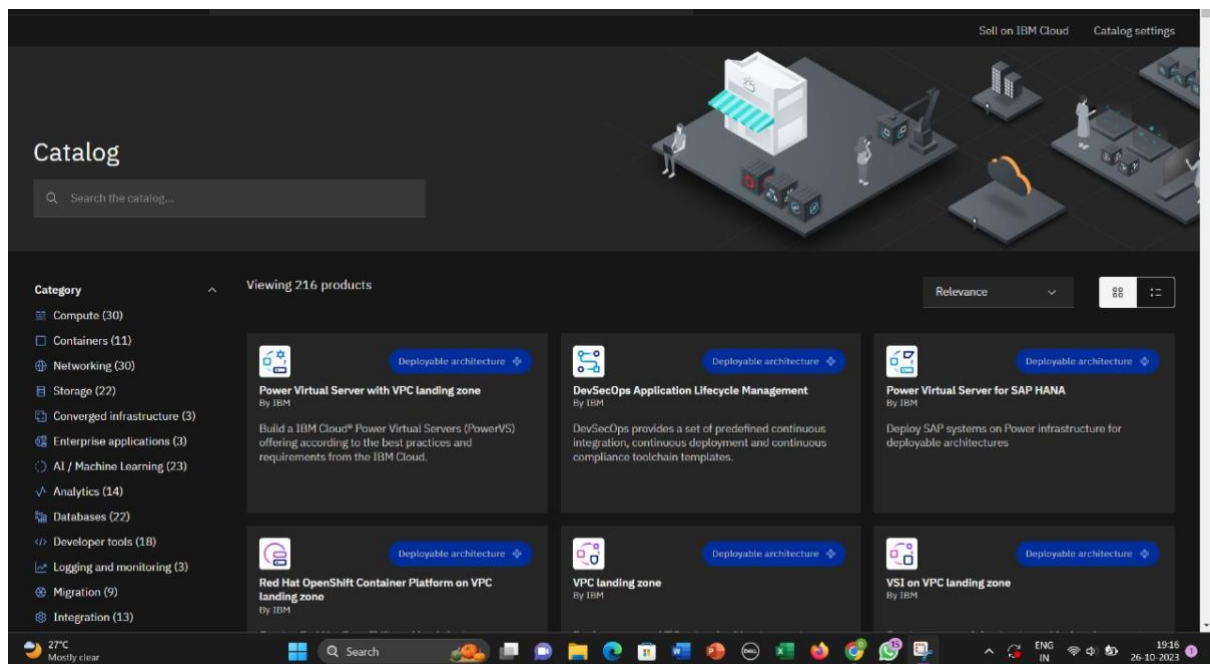
- Rigorously test the chatbot on both Facebook Messenger and Slack to ensure it works as expected and provides a consistent experience.
- Prepare documentation for end users on how to use the chatbot within the messaging platforms.


These steps had helped me to successfully integrate my chatbot with Facebook Messenger and Slack, ensuring a natural conversation flow and accurate responses. Regularly maintain and update the chatbot to enhance its effectiveness.

Conclusion :

The project achieved its objectives by deploying AssistBot, a chatbot built using IBM Cloud Watson Assistant. It demonstrated the effectiveness of using AI-powered chatbots for user assistance on messaging platforms. The chatbot's friendly persona and seamless integration contributed to a positive user experience.

Working Samples :-





Catalog /

Watson Assistant

Watson Assistant lets you build conversational interfaces into any application, device, or channel.

Create

About

Type

Service

Provider

IBM

Last updated

10/04/2023

Category

AI / Machine Learning

Compliance

EU Supported

HIPAA Enabled

IAM-enabled

Location

Sydney

Frankfurt

London

Tokyo

Washington DC

Dallas

Related links

API docs

Docs

Terms

Select a location

Sydney (au-syd)

Select a pricing plan

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

Plan	Features and capabilities	Pricing
Lite	<p>Everything you need to get started, free for as long as you need it</p> <p>Up to 1,000 unique monthly active users (MAUs) chatting with your assistant</p> <p>Up to 10,000 messages per month</p> <p>--- Features ---</p> <ul style="list-style-type: none">World-class conversational AI with WatsonMake your website assistant your own with Webchat - deploy Webchat in minutes, or use our fully extensible architectureBootstrap your assistant by using some of our prebuilt contentConnect to any application or database with a prebuilt integration, or build your own custom integration on top of API endpointsCreate engaging user interactions using images, buttons, and moreAnalyze and enhance your assistant with our analytics dashboardTake comfort knowing your assistant is reliably hosted on IBM Public Cloud <p>--- Limits ---</p> <p>Use to learn more: Watson Assistant</p>	Free

Summary

Watson Assistant

Free

Location: Sydney

Plan: Lite

Service name: Watson Assistant-va

Resource group: Default

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

Create

Add to estimate

27°C

Mostly clear

Search

ENG

IN

19:20

26-10-2023

IBM Cloud

Search resources and products...

Catalog

Manage

DHANUSH MURUGESA...

new

12 month commitment required

Enterprise with data isolation

Contact us for pricing: <http://ibm.biz/contact-wa-enterprise>

--- Features ---

Everything available in Enterprise

Up to 30 instances included

Share log and analytics data across instances

Keep usage and training data private in an isolated single tenant environment

Batch intent classification for regression testing your assistant's dialog, without increasing MAU usage

HIPAA ready instances located in Washington DC

--- Limits ---

Up to 100 Skills per instance

Up to 50 Versions per Dialog Skill

90 days of usage analytics

Assistant session inactivity timeout 7 days

1,000 concurrent calls using phone integration, contact us for higher limits

12 month commitment required

Configure your resource

Service name

ITSupportConversation

Select a resource group

Default

Tags

Examples: env:dev, version-1

Access management tags

Examples: access:dev, proj:version-1

Summary

Watson Assistant

Free

Location: Sydney

Plan: Lite

Service name: ITSupportConversation

Resource group: Default

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

Create

Add to estimate

27°C

Mostly clear

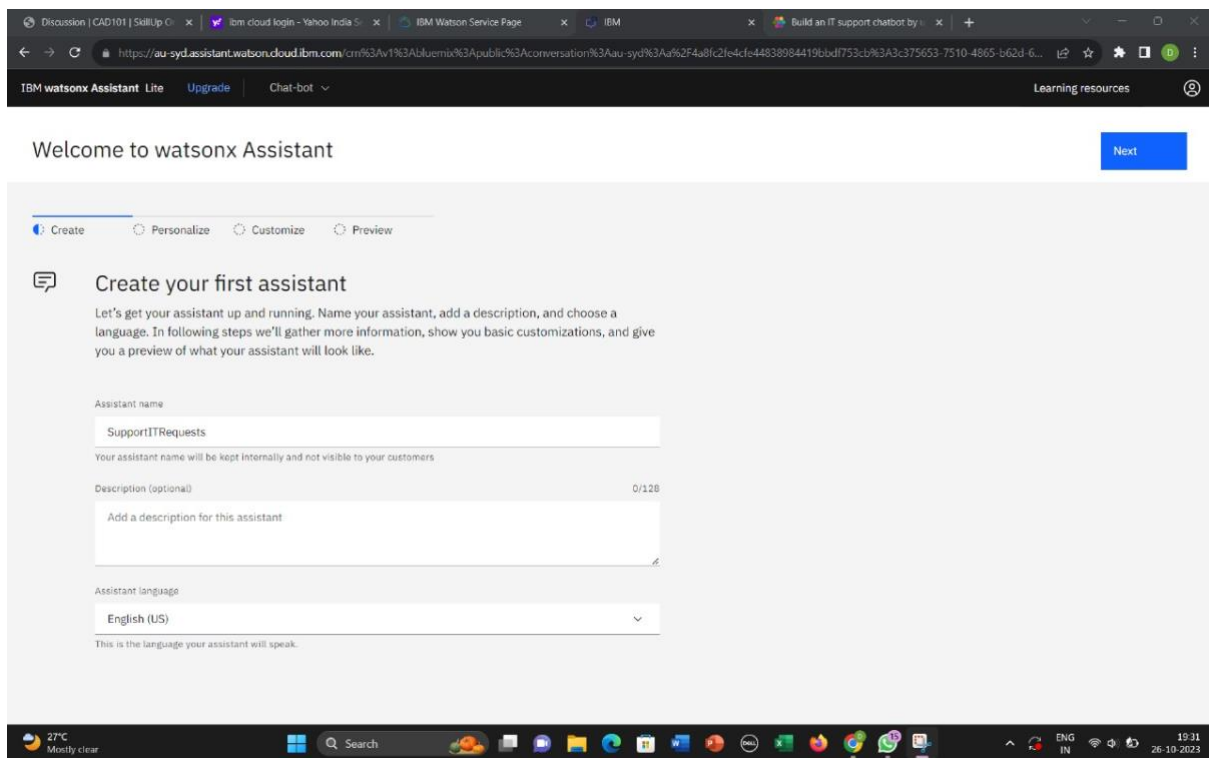
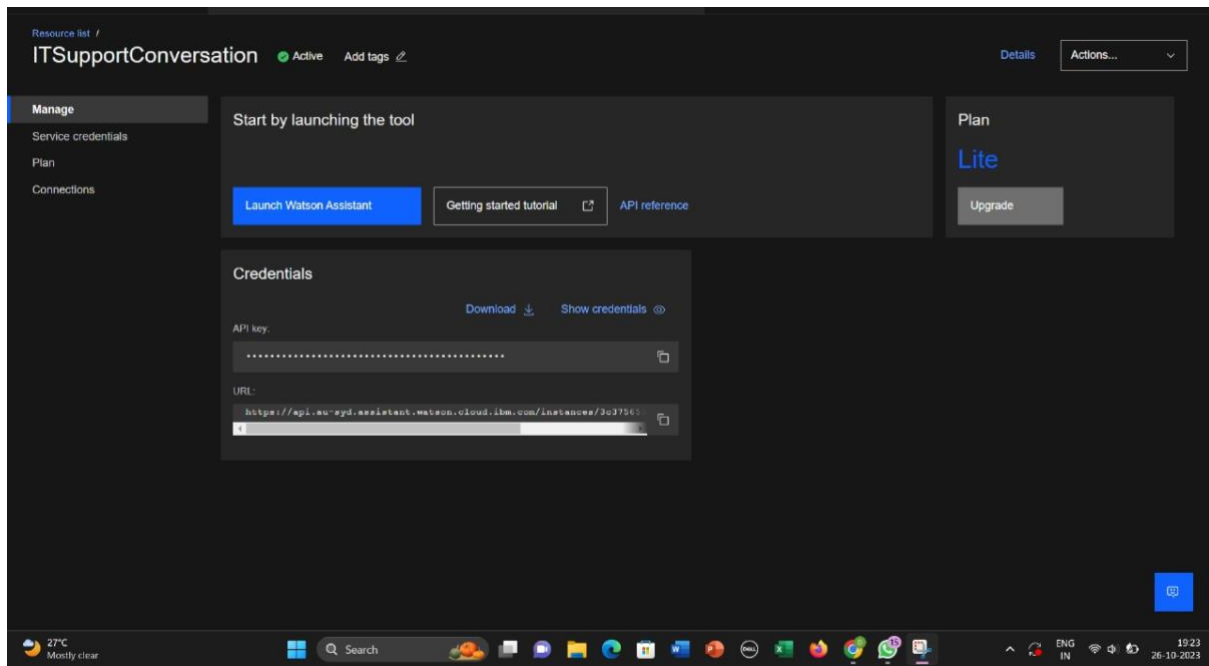
Search

ENG

IN

19:21

26-10-2023



Discussion | CAD101 | SkillUp | IBM cloud login - Yahoo India | IBM Watson Service Page | IBM | Build an IT support chatbot by |

https://au-syd.assistant.watson.cloud.ibm.com/cm/63Av1%3Abluemix%3Apublic%3Aconversation%3Aau-syd%3A%2F4a8fc21e4cfe44838984419b9bd753cb%3A3-375653-7510-4865-b62d-6...

IBM watsonx Assistant Lite Upgrade SupportITReques... Learning resources

Welcome to watsonx Assistant

Back Create

Personalize your assistant

Tell us where your assistant will live
We will create your first channel integration for you, which will be visible on your dashboard. You can always add more or change later.

Where do you plan on deploying your assistant?

SMS

Tell us about yourself
This information will be used to personalize your onboarding experience.

Which industry do you work in?

N/A (I am a student)

What is your role on the team building the assistant?

Developer

Which statement describes your needs best?

I want to provide confident answers to common questions

This is what your customers will experience

4:19

Running World

Do you have the Speed Demons in stock?

The Speed Demons are in stock at our Downtown and Northgate locations, which are both within five miles of you.

What size and color do you need?

I need a size 9 in white

Great news! The Speed Demons are available in white in a size 9. You can purchase them for curbside pick up or we can ship them to you. Which would you prefer?

I'd rather curbside

Wonderful. I can process that payment with the card you have on file. Does that work?

Yes

Text Message

27°C Mostly clear Search 19:34 26-10-2023

Discussion | CAD101 | SkillUp | IBM cloud login - Yahoo India | IBM Watson Service Page | Home | IBM watsonx Assistant | Build an IT support chatbot by |

https://au-syd.assistant.watson.cloud.ibm.com/cm/63Av1%3Abluemix%3Apublic%3Aconversation%3Aau-syd%3A%2F4a8fc21e4cfe44838984419b9bd753cb%3A3-375653-7510-4865-b62d-6...

IBM watsonx Assistant Lite Upgrade SupportITReques... Learning resources

Home

Welcome, you're in watsonx Assistant! Learn more

SupportITRequests | English (US)
No description

Get started

2 steps left 7 min

0%

- ☐ Learn about watsonx Assistant 5 min
- ☐ Explore your learning resources **Now!**
Check out our new set of learning resources, and mark complete when you're ready to continue.
- ☐ Explore watsonx Assistant features on our interactive demo site 2 min

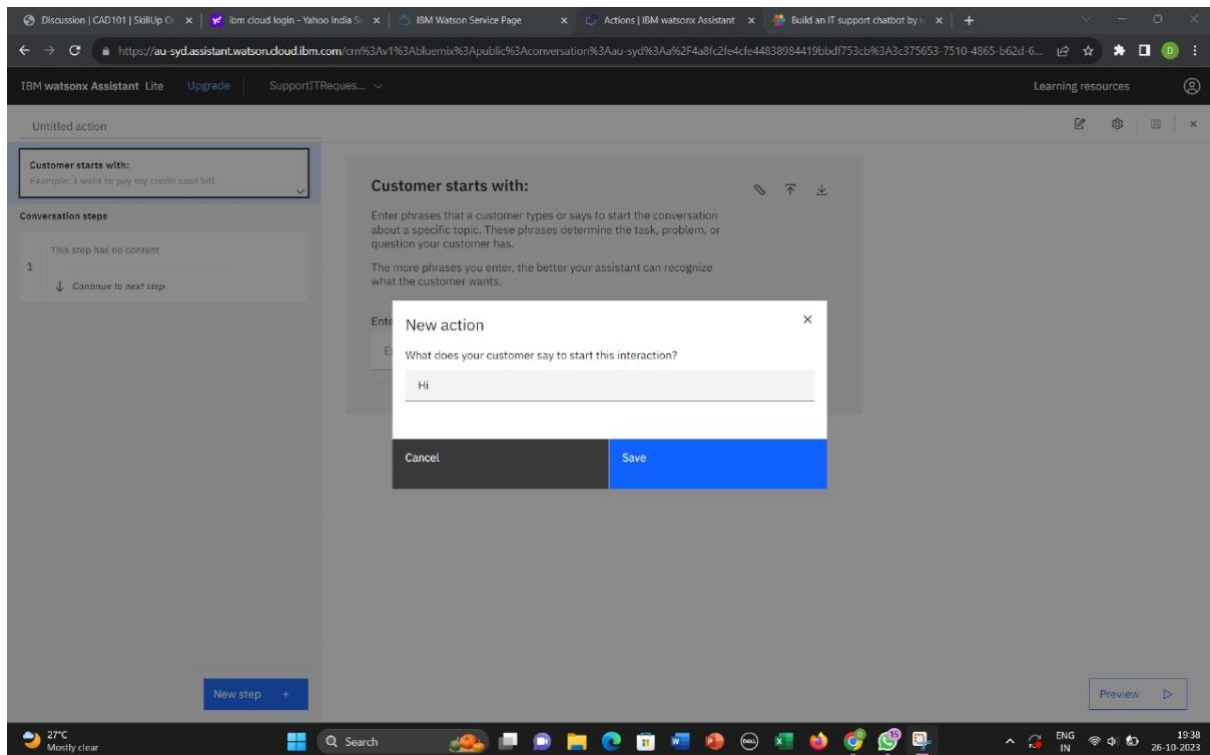
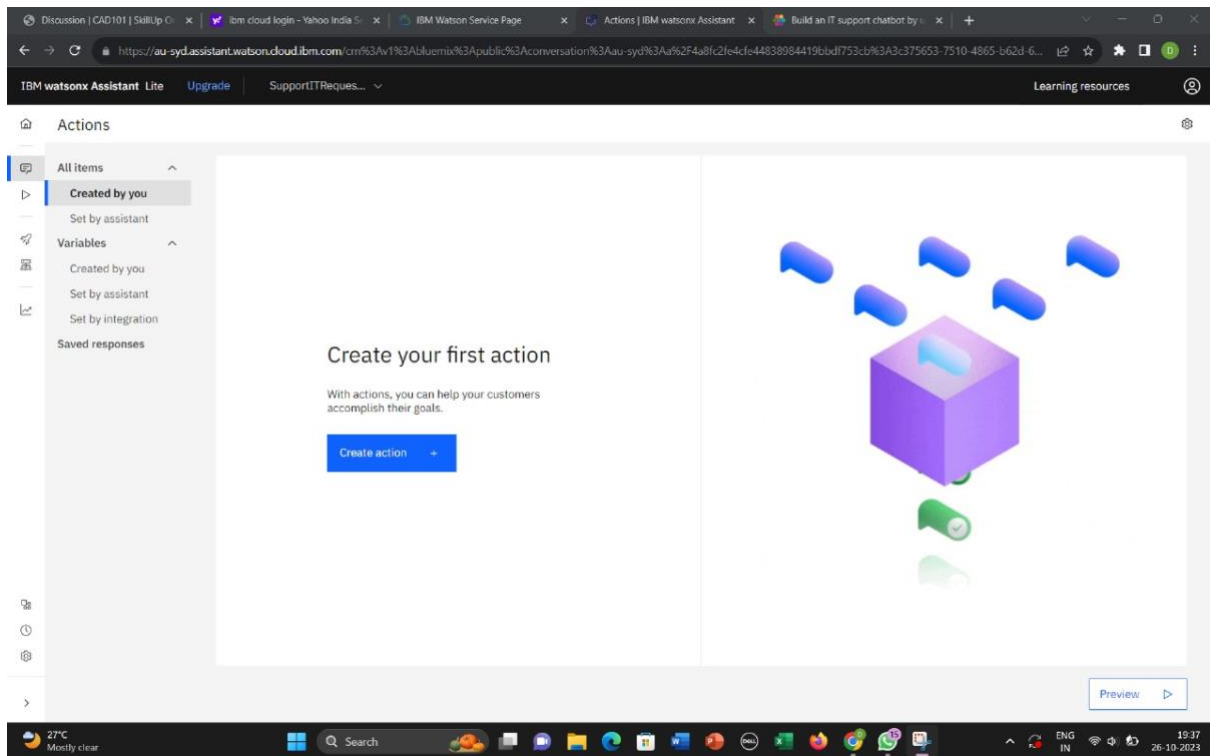
Create a conversation

2 steps left 15 min

0%

- ☐ Create your first action 5 min
- ☐ Create steps 5 min
- ☐ Add 5 examples

27°C Mostly clear Search 19:35 26-10-2023



Discussion | CAD101 | SkillUp C... x | ibm cloud login - Yahoo India S... x | IBM Watson Service Page x | Actions | IBM watsonx Assistant x | Build an IT support chatbot by... x | +

← → ↻ https://au-syd.assistant.watson.cloud.ibm.com/cn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aau-syd%3Aa%2F4a8fc21e4cf44838984419bbd753cb%3A3-375653-7510-4865-b62d-6... u ☆ 🛠️ 📄 🟢 ⋮

IBM watsonx Assistant Lite Upgrade SupportITReques... Learning resources ⓘ

🔄 Your system is being trained... Hide

Customer starts with:
Hi

Conversation steps

1	This step has no content ↓ Continue to next step
---	---

Step 1 [🔗](#)

Is taken without conditions [Set variable values f...](#)

Assistant says

B I ⌨️ f... 🗑️ 📄 📁 📧 📧 📧 📧 📧

For example: Please select from the following options:

👤 Define customer response

And then

↓ Continue to next step

New step + Preview ▶️

27°C Mostly clear Search 19:39 26-10-2023

Discussion | CAD101 | SkillUp C... x | ibm cloud login - Yahoo India S... x | IBM Watson Service Page x | Actions | IBM watsonx Assistant x | Build an IT support chatbot by... x | +

← → ↻ https://au-syd.assistant.watson.cloud.ibm.com/cn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aau-syd%3Aa%2F4a8fc21e4cf44838984419bbd753cb%3A3-375653-7510-4865-b62d-6... u ☆ 🛠️ 📄 🟢 ⋮

IBM watsonx Assistant Lite Upgrade SupportITReques... Learning resources ⓘ

Greet customer

Assistant starts with step 1

Conversation steps

1	Welcome, how can I assist you? 🕒 Action complete
---	---

Action starts

When your assistant is first opened or engaged with. ⓘ

No training examples necessary.

Use the default greeting or customize your own.

New step +

Preview

Greet customer {default}

Welcome, how can I assist you?

Hi

Hi recognized

Welcome, how can I help you?

⚡ There are no additional steps for this action. Add a new step or end the action.

Type something...

27°C Mostly clear Search 19:41 26-10-2023