Name : P.ASHIK JERSHAN

Register Number : 210621205007

Phase : 3

Project Title : Chatbot Deployment with IBM Cloud Watson Assistant

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');
                    window.watsonAssistantChatOptions.input =
document.getElementById('user-input');
                    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/prelo
ad-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.