

Chatbot Deployment with IBM Cloud Watson Assistant

Problem Definition:

The problem we aim to address is the efficient deployment of a chatbot using support capabilities by harnessing the potential of a chatbot through IBM Cloud Watson Assistant. We need to create a robust and user-friendly chatbot that can effectively interact with users, provide accurate responses, and integrate seamlessly with various applications and platforms. We also need to ensure that the chatbot is easily deployable on the IBM Cloud platform and can scale to handle a growing user base. The objective is to institute an interactive and streamlined conversational interface capable of automating routine tasks, furnishing prompt responses, and elevating overall customer satisfaction.

The salient challenges and prerequisites for deploying a chatbot employing IBM Cloud Watson Assistant encompass:

- **Customer Support Automation:** The organization seeks to alleviate the burden on support agents by automating frequently asked questions and providing self-service options for clientele.
- **Enhanced Responsiveness:** Customers demand expeditious and precise responses to their inquiries. The chatbot must possess the ability to discern user intents and proffer timely and pertinent information.
- **Tailored User Experience:** The organization aspires to endow users with personalized interactions by customizing responses predicated on user profiles, historical interactions, and preferences.
- **Backend System Integration:** The chatbot necessitates seamless integration with backend systems, such as customer databases, knowledge repositories, or support ticketing platforms, to access pertinent data and furnish accurate responses.
- **Multi-channel Proficiency:** The chatbot should be accessible via diverse channels, encompassing websites, mobile applications, and social media platforms, to ensure a uniform experience across disparate touchpoints.

Design Thinking Approach:

Empathize: Conduct comprehensive interviews and workshops with stakeholders, customers, and support agents to gain a deep understanding of their pain points, challenges, and expectations. Gather insights into prevalent support requests, frequently asked questions, and customer preferences for autonomous service.

Objective Definition: Articulate clear chatbot deployment objectives, with a focus on optimizing cost-efficiency, enhancing customer satisfaction, and boosting conversion rates. Establish quantifiable success metrics, including response time benchmarks and customer satisfaction assessments.

Generate Innovative Concepts: Brainstorming creative chatbot functionalities based on user needs is essential for crafting a valuable user experience. Exploring various conversation flows, interfaces, and use cases allows for innovative solutions. Strategically consider potential use cases where the chatbot can deliver significant value, such as handling product inquiries, providing FAQ support,

Develop and Refine Prototypes: Creating a robust chatbot prototype is a practical step to visualize and test your ideas. Designing conversational flows aligned with use cases and iteratively refining the prototype based on feedback ensures a user-friendly interface.

Testing and Iteration: Conducting user testing and collecting feedback are critical for validating the chatbot's effectiveness. This iterative process helps iron out any issues and continuously improves the design and functionality.

Deploy and Evaluate: Launching the chatbot on relevant platforms and monitoring performance metrics ensures that it operates effectively. Gathering user feedback post-deployment allows for ongoing enhancements, aligning the chatbot with evolving user needs.

Chatbot Deployment with IBM Watson Assistant

Scope and Requirements Gathering:

- 1. Definition of Use Case and Objectives:** Commencing this journey involves the imperative task of precisely determining the purpose of your chatbot. Is it intended for customer support, sales, or a more specific function? The pivotal aspect is the establishment of unambiguous objectives, serving as benchmarks for assessing accomplishments.
- 2. Understanding the Target Audience:** The subsequent step necessitates the delineation of the chatbot's designated users and their requisites. This entails contemplation of demographics, preferences, and the issues your chatbot should adeptly address.
- 3. Accumulation of Data:** Within this, an essential endeavor involves the aggregation of pertinent data and content, serving as the bedrock upon which your chatbot will draw to disseminate information. This encompassing process may encompass FAQs, product specifications, or support documentation, among other facets.
- 4. Establishing Key Performance Indicators (KPIs):** The underpinning of this involves the formulation of KPIs, instrumental in appraising the chatbot's performance. These benchmarks include response time, user satisfaction, and the efficiency in resolving user queries.

Design:

- 5. Architecting the Conversation Flow:** In this context, you will meticulously craft a conversation flowchart that delineates user interactions. The central objective revolves around planning the chatbot's engagement with users and its ability to steer them through meaningful dialogues.
- 6. Constructing Dialogues:** This encompasses the methodical design of dialogues and responses exhibited by your chatbot, predicated upon user input. The primary aim is to forge responses that resonate with a natural tone while delivering genuine assistance.
- 7. Implementation of Natural Language Understanding (NLU):** In this juncture, the emphasis lies in augmenting your chatbot's capacity to grasp user queries through the infusion of Natural Language Understanding capabilities. This entails the diligent training of the chatbot to discern user intent and entities embedded within the discourse.

Development:

- 8. Creation of the Chatbot:** In this stage, you will leverage IBM Watson Assistant to construct the foundational framework of your chatbot. This will encompass the delineation of intents (user intentions), entities (specific data elements), and dialog nodes (branches of conversation), collectively shaping the capabilities of your chatbot.

9. Definition of Intents and Entities: Within the framework of Watson Assistant, the configuration of intents, entities, and dialog nodes takes center stage. These components are instrumental in empowering the chatbot to adeptly comprehend and responsively address user inquiries.

10. Training and Rigorous Testing: Dedication of time and resources to train the chatbot through exposure to sample conversations, followed by a comprehensive testing regimen to assess its responsiveness. The overarching goal is the perpetual refinement of the chatbot's knowledge and the logic underpinning its dialogues.

Integration:

11. Seamless Platform Integration: Within this, the seamless integration of the chatbot into your preferred platform or application assumes prominence. The primary objective revolves around ensuring the uninterrupted and efficient exchange of information between the chatbot and its users.

12. Testing and Optimization: Subsequent to integration, an exhaustive testing process within a real-world environment comes to the fore, identifying any glitches or areas ripe for improvement. The optimization efforts are channelled towards elevating the chatbot's overall performance and functionality.

Monitoring and Maintenance:

13. Continuous Vigilance: Ongoing, real-time scrutiny of the chatbot's performance stands as an imperative. This entails the collection of user feedback and the meticulous tracking of metrics, serving as catalysts for pinpointing opportunities for enhancement.

14. Regular Iterations: To ensure the alignment of your chatbot with the evolving needs and feedback of users, the implementation of periodic updates to its dialogues and responses assumes paramount significance. This iterative process is pivotal in sustaining its long-term efficacy.

Assessment and Documentation:

15. Comprehensive Documentation: The creation of thorough and exhaustive documentation encompassing all facets of your chatbot's design, development, and deployment process emerges as a fundamental undertaking. This repository of information serves as a valuable resource for future maintenance and expansion endeavors.

16. Inclusion of Performance Metrics: The documentation should include the integration of performance metrics and KPIs, providing a robust framework for measuring the chatbot's efficacy and its return on investment (ROI).

CHATBOT DEPLOYMENT WITH IBM CLOUD

WATSON ASSISTANT

PHASE 3: DEVELOPMENT PART 1

In this phase, we will begin developing a chatbot using IBM Watson Assistant. IBM Watson Assistant is a sophisticated cloud-based platform for building conversational AI solutions. It leverages three fundamental components to create and train chatbots or virtual assistants. Here is a brief description of each component:

Entities:

- Entities identify and categorize specific information within a conversation.
- They enable the chatbot to recognize and extract relevant data.
- Example: In a customer support chatbot, entities like "product names" or "issue types" help understand user inputs accurately, even with varying information.

Intents:

- Intents define the purpose or goal behind a user's message.
- They help the chatbot recognize why a user is reaching out.
- Example: "FAQ" for general questions or "Order Status" for inquiries about an order's delivery.
- Associations of intents with appropriate responses enable contextually relevant conversations.

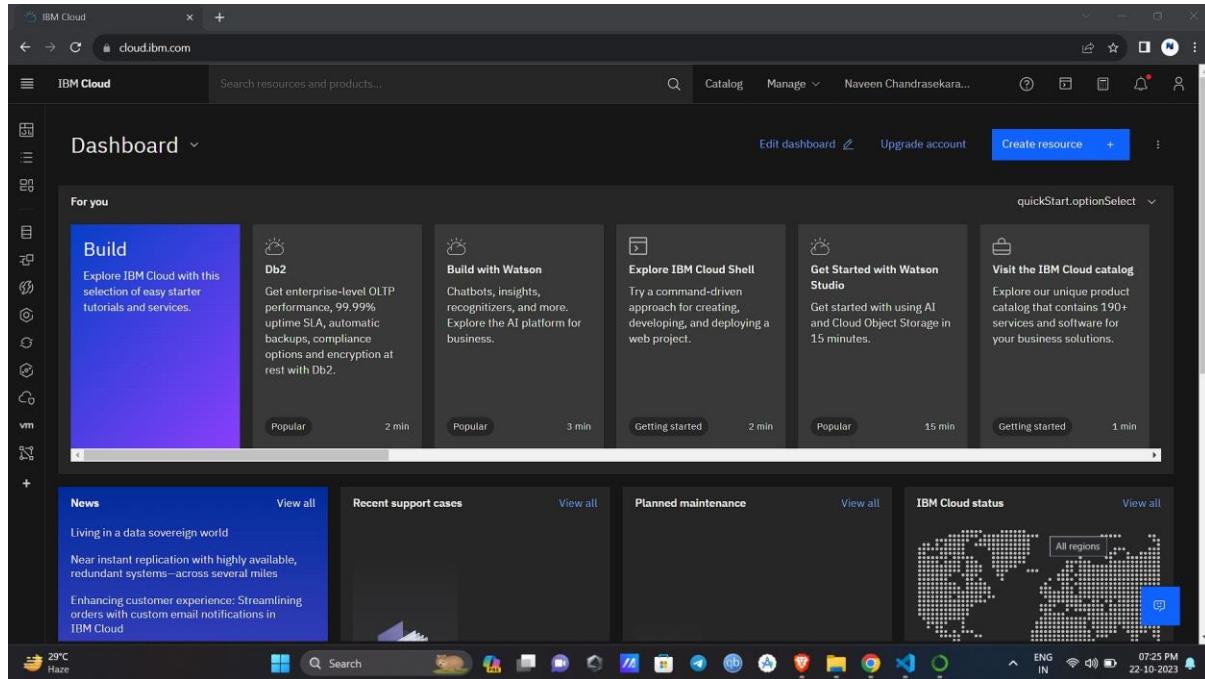
Dialogs:

- Dialogs control the conversational flow of the chatbot.
- They specify how the chatbot responds to user inputs based on identified intents and entities.
- Dialogs consist of nodes, each representing a step in the conversation.
- Nodes can include messages, actions, and conditions, allowing the chatbot to guide users and provide dynamic responses.

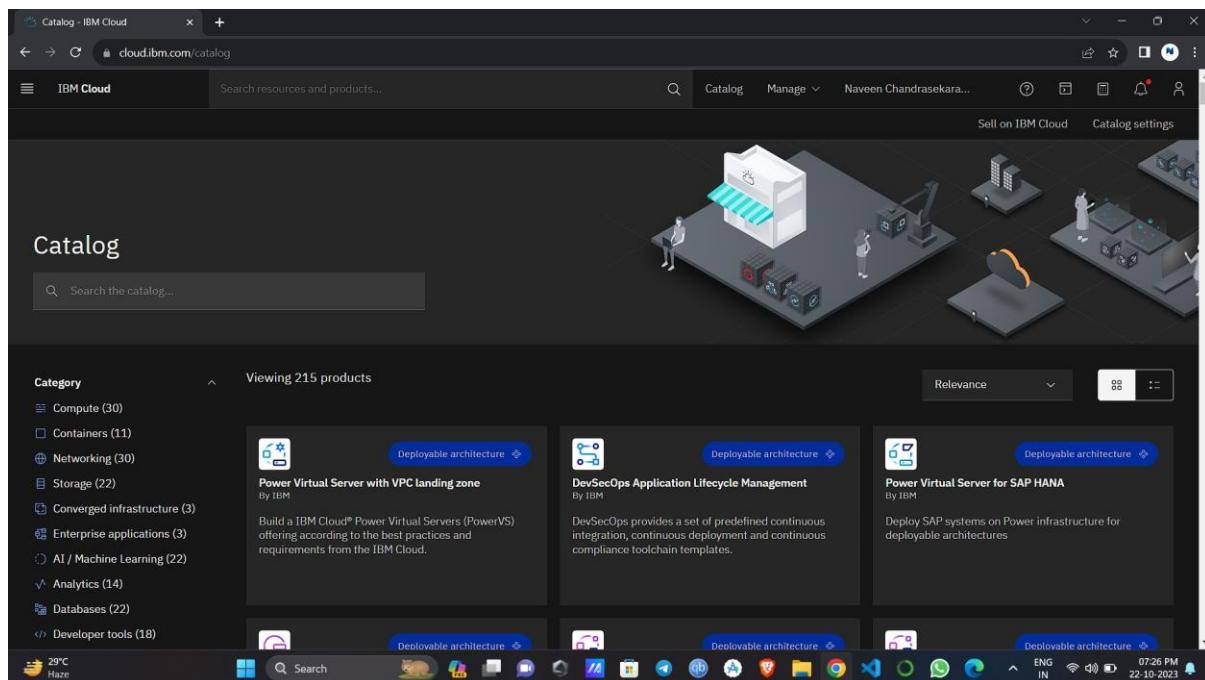
These components collectively form the foundation of a chatbot, ensuring it can understand, engage, and respond effectively to user interactions.

Here's an overview of the steps involved in deploying a chatbot using IBM Watson Assistant:

1. Sign in to your IBM account and navigate to the IBM Cloud Catalog.



2. Use the catalog's search function to find "Watson Assistant," then select it.



The screenshot shows the IBM Cloud Catalog interface. A search bar at the top contains the query "Watson". Below the search bar, there is a sidebar with a search input field containing "Watson" and a dropdown menu listing various Watson services: Watson Assistant, Watson Discovery, Watson Studio, Watson OpenScale, Watson Knowledge Catalog, Watson Machine Learning, Enterprise applications (3), AI / Machine Learning (22), Analytics (14), Databases (22), and Developer tools (18). The main content area displays several service cards. One card for "Watson Assistant" is highlighted, showing its "Deployable architecture" badge. Other visible cards include "DevSecOps Application Lifecycle Management" and "Power Virtual Server for SAP HANA". The interface includes a navigation bar with "Catalog" and "Manage" tabs, and a user profile for "Naveen Chandrasekara...". The bottom of the screen shows a taskbar with various application icons.

3. Create a new instance of Watson Assistant, indicating the desired location and plan (e.g., Lite).

The screenshot shows the "Watson Assistant - IBM Cloud" service creation page. The left sidebar lists service details: Type: Service, Provider: IBM, Last updated: 10/04/2023, Category: AI / Machine Learning, and Compliance: EU Supported, HIPAA Enabled, IAM-enabled. The main area has tabs for "Create" and "About". Under "Create", the "Select a location" dropdown is set to "London (eu-gb)". The "Select a pricing plan" section shows the "Lite" plan selected, which is "Free". The "Features and capabilities" for the Lite plan include: "Everything you need to get started, free for as long as you need it", "Up to 1,000 unique monthly active users (MATUs) chatting with your assistant", "Up to 10,000 messages per month", and a list of features: "World-class conversational AI with Watson", "Make your website assistant your own with Webchat - deploy Webchat in minutes, or use our fully extensible architecture", "Bootstrap your assistant by using some of our prebuilt content", "Connect to any application or database with a prebuilt integration", and "Build your own custom integration on top of API endpoints". On the right, the "Summary" panel shows the service name "Watson Assistant", location "London", plan "Lite", and resource group "Default". It also includes a checkbox for accepting license agreements and buttons for "Create" and "Add to estimate". The bottom of the screen shows a taskbar with various application icons.

Watson Assistant - IBM Cloud

cloud.ibm.com/catalog/services/watson-assistant

IBM Cloud

Search resources and products...

Displayed prices do not include tax. Monthly prices shown are for country or location: United States

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

Plan: Features and capabilities

Lite

Pricing: Free

Features and capabilities:

- Everything you need to get started, free for as long as you need it
- Up to 1,000 unique monthly active users (MAUs) chatting with your assistant
- Up to 10,000 messages per month
- Features ---
 - World-class conversational AI with Watson
 - Make your website assistant your own with Webchat - deploy Webchat in minutes, or use our fully extensible architecture
 - Bootstrap your assistant by using some of our prebuilt content
 - Connect to your application or database with a prebuilt integration, or build your own custom integrations via our API endpoints
 - Create engaging user interactions using images, buttons, and more
 - Analyze and enhance your assistant with our analytics dashboard
 - Take comfort knowing your assistant is reliably hosted on IBM Public Cloud
 - Limits ---
 - Up to 5 Skills (Dialog, Action, Search)
 - 7 days of usage analytics
 - Session inactivity timeout 5 minutes
 - Services are deleted after 30 days of inactivity

Everything you need to get started, free for as long as you need it

Lite plan services are deleted after 30 days of inactivity.

Trial

Pricing: 30 Day trial period (no credit card required)

Features:

Watson Assistant

Location: London

Plan: Lite

Service name: Watson Assistant-rz

Resource group: Default

Summary

I have read and agree to the following license agreements:

Terms

Create

Add to estimate

29°C Haze

Search

07:36 PM 22-10-2023

4. Access the Watson Assistant tool and proceed to create a new assistant, assigning it a name and an optional description.

IBM Watson Service Page

eu-gb.assistant.watson.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Aconversation%3Aeu-gb%3Aa%2F02f47ceb494343cb9089801119fc034%3A9b9d8350-8b75-494c-beec-daa637478fc4...

IBM Watsonx Assistant Lite Upgrade ChatBot with Wa... Learning resources

Home

Welcome, you're in watsonx Assistant!

ChatBot with Watson | English (US)

No description

Get started

3 steps left: 7 min

0%

Learn about watsonx Assistant 5 min

Explore your learning resources New! 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

0%

Create a conversation

3 steps left: 15 min

0%

Create your first action

29°C Haze

Search

07:43 PM 22-10-2023

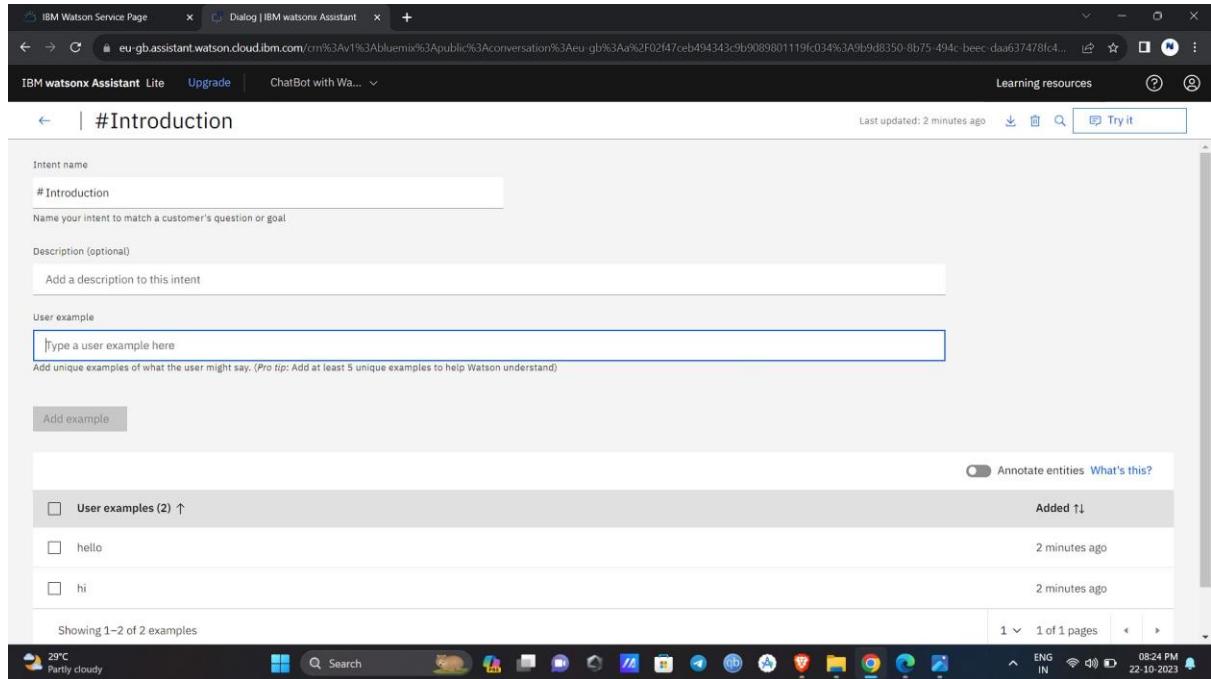
5. Activate the dialog flow for the assistant.

The screenshot shows the 'IBM Watson Service Page' with the URL eu-gb.assistant.watson.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Aconversation%3Aeu_gb%3Aa%2F02f47ceb494343c9b9089801119fc034%3A9b9d8350-8b75-494c-beec-daa637478fc4.... The tab is 'Settings | IBM Watsonx Assistant'. The main content area has a sidebar with 'Assistant settings' and 'Download/Upload' sections. Under 'Download/Upload', there are buttons for 'Download/Upload files' and 'Enable multilingual download'. Below this is the 'Dialog' section, which contains a button labeled 'Activate dialog'. Further down is the 'Delete this assistant' section, which includes a 'Delete assistant' button. The bottom of the screen shows a Windows taskbar with various icons and system status.

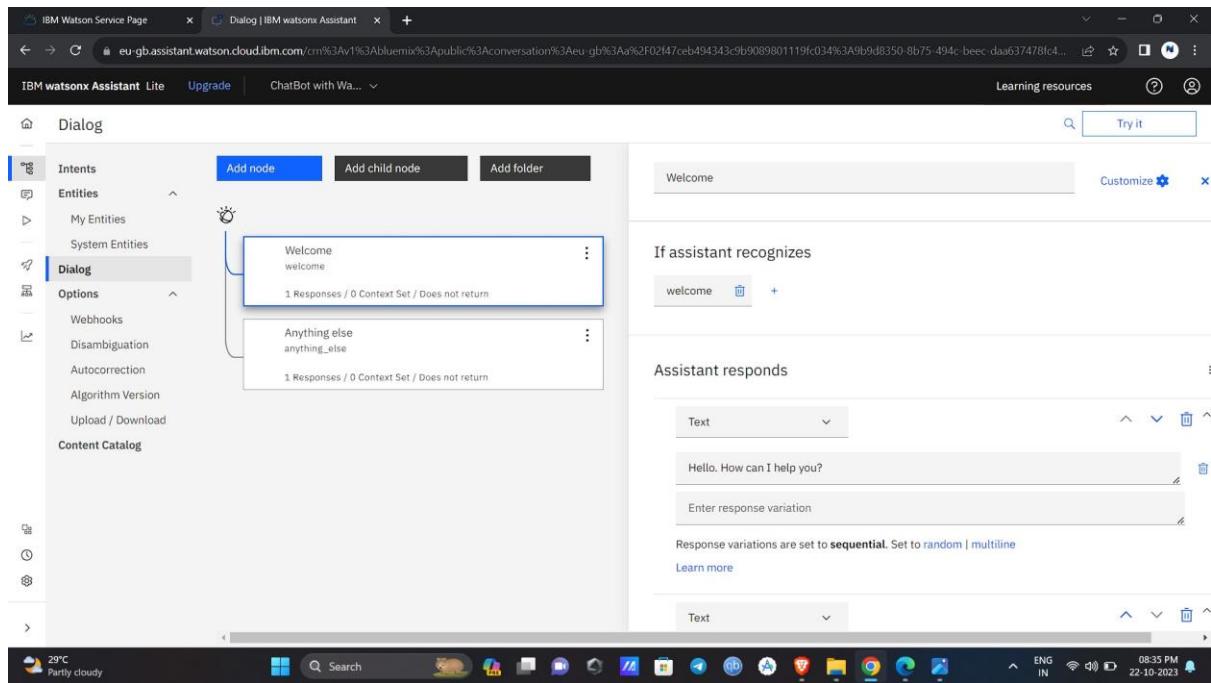
6. Begin by creating entities, defining relevant information (e.g., cuisine types, genres), and assigning variables to them.

The screenshot shows the 'IBM Watson Service Page' with the URL eu-gb.assistant.watson.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Aconversation%3Aeu_gb%3Aa%2F02f47ceb494343c9b9089801119fc034%3A9b9d8350-8b75-494c-beec-daa637478fc4.... The tab is 'Dialog | IBM Watsonx Assistant'. The main content area shows the creation of an entity named '@username'. The 'Entity name' field contains '@username'. The 'Value' field has a placeholder 'Type a value'. There are tabs for 'Dictionary (5)' and 'Annotation (5)'. The 'Dictionary (5)' tab is selected, showing a table with 5 rows of values: Adam, Kishore, and Madheesh, all categorized under 'Synonyms'. The bottom of the screen shows a Windows taskbar with various icons and system status.

7. Generate intents to represent the various purposes or goals expressed in user messages (e.g., check account balance, transfer funds).



8. Assemble a dialog flow by creating nodes for each intent, along with defining appropriate responses for different user inputs.



9. Validate the functionality of the chatbot by testing it with sample queries and verifying the responses within the Watson Assistant interface.

The screenshot shows the IBM Watson Assistant interface. On the left, a configuration panel for an intent named '#General_Greetings' is displayed. It includes fields for Intent name (#General_Greetings), Description (optional) (Greet the bot.), and User example (Type a user example here). Below these are sections for Annotations and User examples, which lists 27 examples including 'Good day', 'Good evening', and 'Good morning', each added 14 hours ago. On the right, a 'Try it out' window is open, showing a conversation between a user (@username:Naveen) and the bot. The user says 'Hi My name is Naveen', and the bot responds with 'Hi! I'm a Chatbot. What's your name?'. The user replies '@username:Naveen', and the bot responds with 'Hi! Naveen . How can I help you?'. A message bar at the bottom of the test window says 'Enter something to test your assistant'.

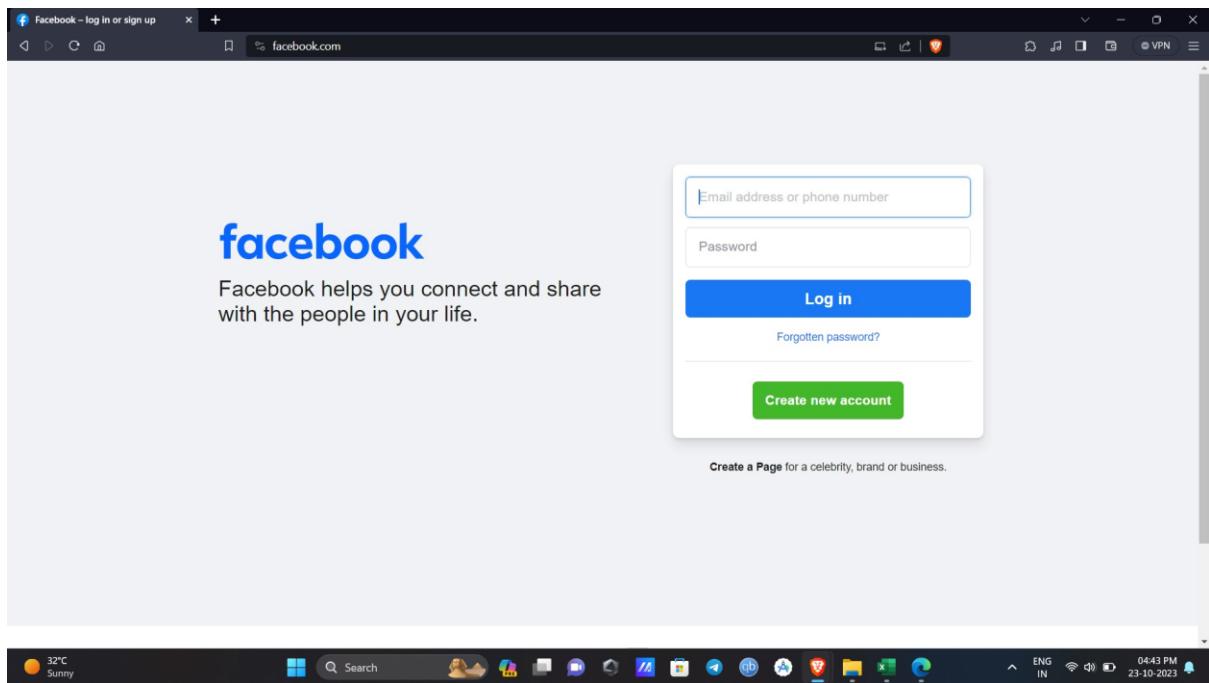
CHATBOT DEPLOYMENT WITH IBM WATSON ASSISTANT

PHASE4: DEVELOPMENT PART 2

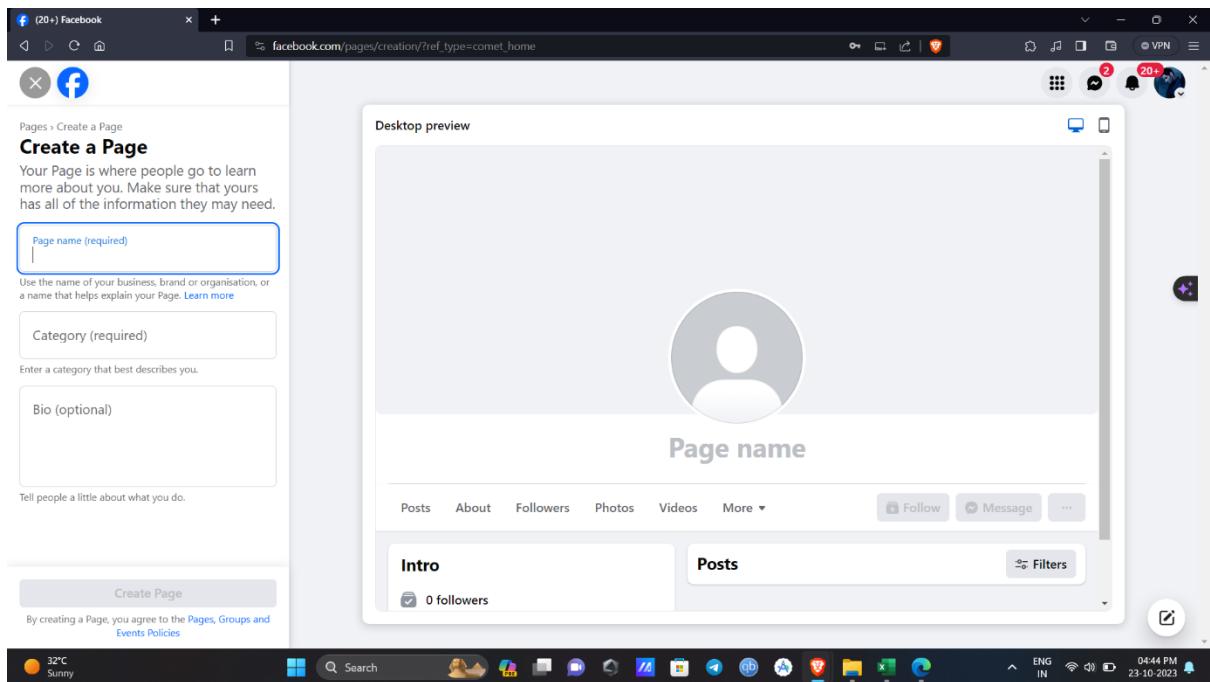
In this phase, we will continue building our project using IBM Cloud Foundry. After finishing the project, we will integrate our chatbot with facebook messenger and slack. Ensuring that the conversation flows naturally and that the chatbot's responses are informative and accurate.

Here's an overview of the steps involved in integrating a chatbot using IBM Watson Assistant into facebook messenger:

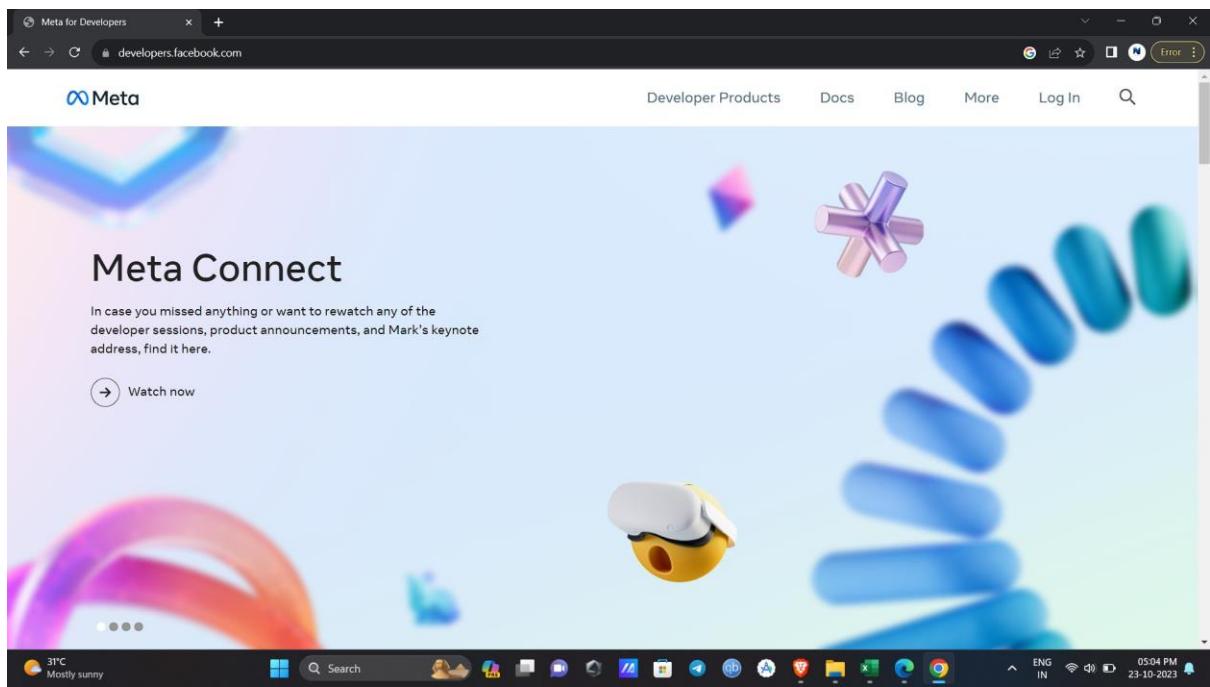
1. Begin by signing up for a Facebook account if you don't have one already.

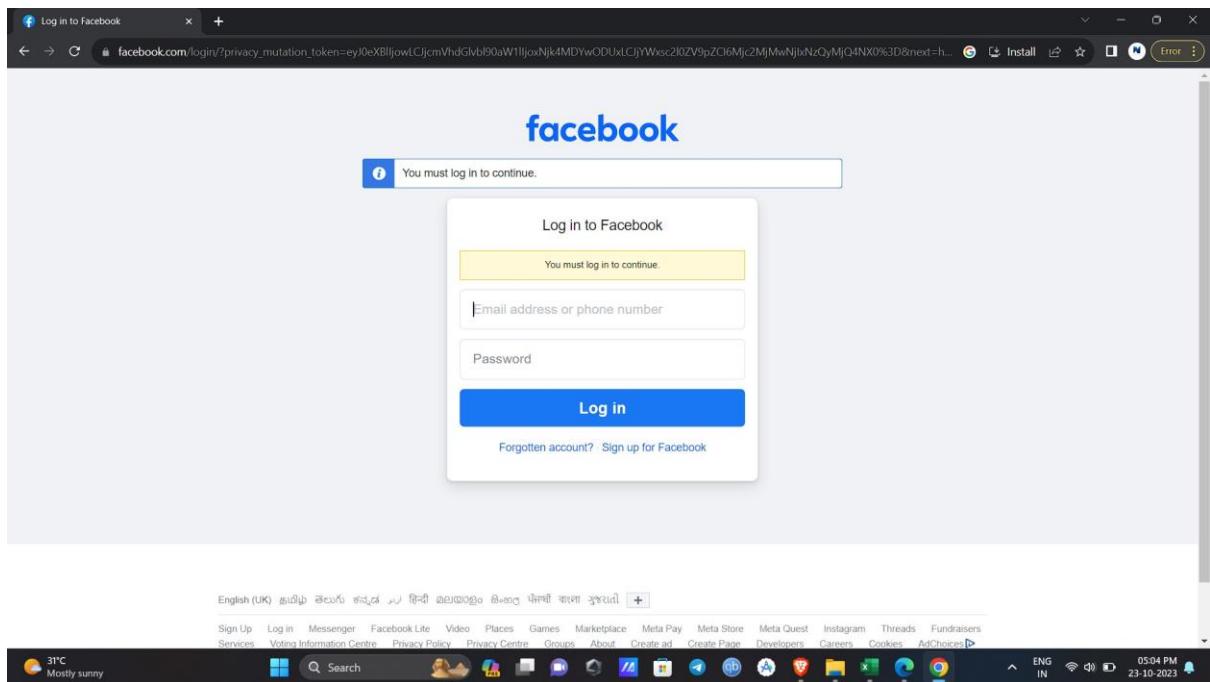


2. After that, create a Facebook page that corresponds to your chatbot.



3. Next, create a Facebook developer account or log in to an existing one.





Create a Facebook app specifically for your chatbot.

Obtain the app secret for your Facebook app from there.

The screenshot shows the Facebook Developers Dashboard for the Chatbot app (App ID: 630254602655368). The dashboard includes sections for Required actions, Application Rate Limit, and User Rate Limit. The Application Rate Limit section shows that 0% of the limit has been used, with 100% remaining. The User Rate Limit section shows 0 users throttled. There is also a link to 'View Details'.

The screenshot shows the Facebook Developers Settings page for the Chatbot app (App ID: 630254602655368). The 'App settings' section is selected, and the 'Basic' tab is active. Other tabs include 'Advanced', 'App roles', 'Alerts', and 'App Review'. The 'Basic' tab displays fields for App ID (630254602655368), App secret (redacted), Display name (Chatbot), Namespace (empty), App domains (empty), Contact email (naveen.chandrasekaran20@gmail.com), Privacy Policy URL (Privacy policy for Login dialog and app details), Terms of Service URL (Terms of Service for Login dialog and App Details), App icon (1024 x 1024 pixels, placeholder image), and Category (empty dropdown).

4. Connect your chatbot to your Facebook page by generating a page access token. Remember to save this token for later use.

The screenshot shows the Facebook Developer Portal interface. On the left, there's a sidebar with various settings like Dashboard, Required actions, App settings, App roles, Alerts, App Review, Products, and Messenger. Under Messenger, 'Settings' is selected. In the main content area, there's a 'Get started' section with a note about resources and a 'Complete Documentation' link. Below it is the 'Access Tokens' section. It has a table with one row for 'Chatbot' (App ID: 144856992047581) and a 'Generate token' button. There's also a 'Pages' section with a 'Add or remove Pages' button. At the bottom, there's a 'Webhooks' section with a 'Add Callback URL' button. The status bar at the bottom shows weather (31°C, Mostly sunny), system icons, and the date/time (23-10-2023, 05:09 PM).

This screenshot is similar to the previous one but shows a modal dialog titled 'Token generated' over the 'Access Tokens' section. The modal displays the generated token: EAAI0... (with several dots). It includes a 'Copy' button and a 'Done' button. The background content remains the same, showing the 'Get started' section and the 'Webhooks' section. The status bar at the bottom is identical to the first screenshot.

5. Now, it's time to integrate your chatbot with Watson Assistant or any other chatbot you prefer.

6. First, launch your chatbot assistant instance and pick the specific chatbot you wish to connect with your web page.

The screenshot shows the IBM Cloud interface for managing a Watson Assistant instance named "Watson Assistant-ox". The "Manage" tab is selected. It displays the "Plan" as "Lite" and provides a "Launch Watson Assistant" button. Below this, there is a "Credentials" section showing an API key and a URL. The URL is <https://api.eu-gb.assistant.watson.cloud.ibm.com/instances/f81...>. The status bar at the bottom indicates the system is running on "ENG - SL" and was last checked "25-10-2023 08:44 PM".

The screenshot shows the "IBM Watson Assistant Lite" interface. The left sidebar includes "Home", "Build", "Dialog" (which is selected), "Actions", "Preview", "Deploy", "Publish", "Environments", "Improve", and "Analyze". The main area displays a table of intents. The columns are "Intents (31) ↑", "Description", "Modified ↑", and "Examples ↑". The table lists 31 intents, such as "#budget", "#buy", "#Customer_Care_Appointments", etc. At the bottom, it says "Showing 1-31 of 31 intents". The status bar at the bottom indicates the system is running on "ENG - IN" and was last checked "25-10-2023 08:41 PM".

7. Add the Facebook Messenger integration and provide the required credentials. Make sure to verify the callback URL and save it.

The screenshot shows the IBM Watson Assistant Lite interface. On the left, there's a sidebar with icons for Home, Channels, and Extensions. Under 'Channels', there are five options: SMS, Facebook Messenger, Microsoft Teams, Slack, and WhatsApp with Twilio. Each option has an 'Add' button and a '+' button. Below the channels, under 'Extensions', there's a search bar and a result for 'Segment'. A note says 'Get a better understanding of your'.

The screenshot shows the 'Facebook Messenger' setup screen. At the top, it says 'Facebook Messenger' and 'Draft'. There are four tabs: 'Facebook app' (selected), 'Facebook page', 'Webhook', and 'Advanced options'. Below the tabs, there's a section titled 'Create a Facebook application'. It instructs users to go to [Facebook for Developers](#) and log in with their Facebook credentials. It also notes that if they already have an app, they can select it from the 'My Apps' menu. It provides instructions for finding the app secret in the settings. A text input field is shown for 'Application secret' with a placeholder of '.....' and a copy icon.

8. Additionally, add a subscription for messages and messaging postbacks to ensure smooth communication.

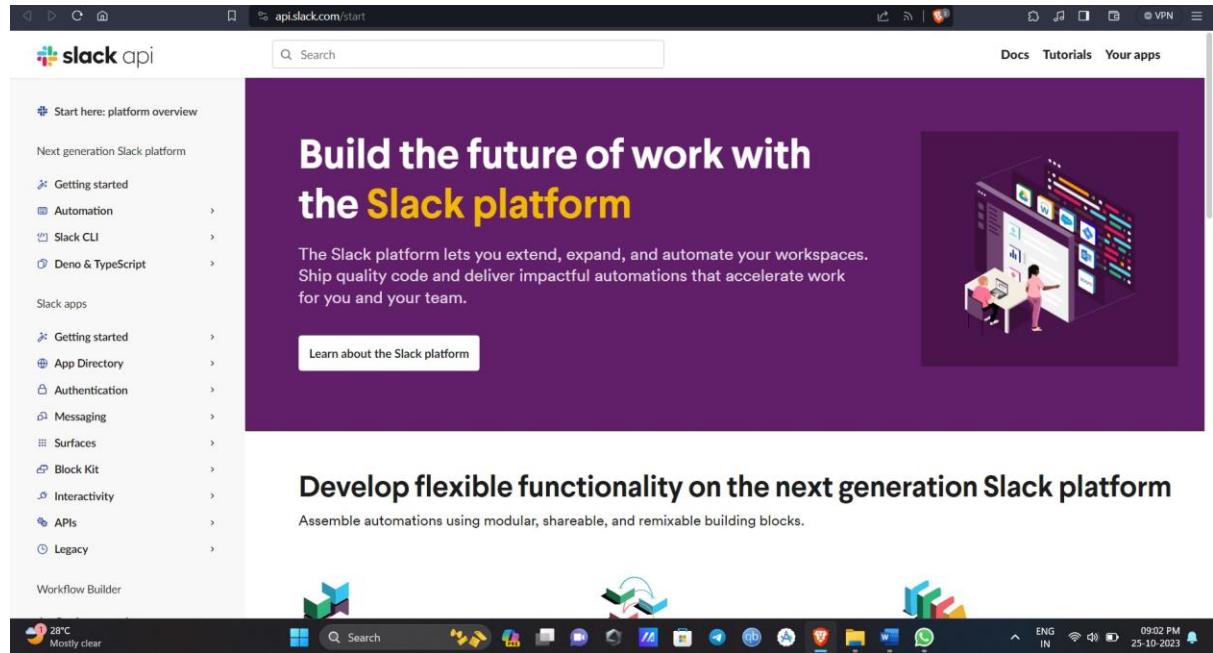
The screenshot shows the Facebook Developers portal with the URL developers.facebook.com/apps/630254602655368/messenger/settings/. The page title is "Meta for Developers". The main navigation bar includes "Docs", "Tools", "Support", "My Apps", and a search bar. The "App Mode" is set to "Development" and "Live", and the "App type" is "None". On the left sidebar, under the "Messenger" section, "Settings" is selected. In the center, a modal window titled "Edit page subscriptions" is open for a page named "Chatbot". The modal lists various subscription fields with checkboxes, some of which are checked (e.g., "messages", "messaging_postbacks"). At the bottom of the modal are "Cancel" and "Save" buttons. Below the modal, there's a warning about new privacy rules in Europe.

Finally, don't forget to test your chatbot app on Messenger to ensure everything is working correctly.

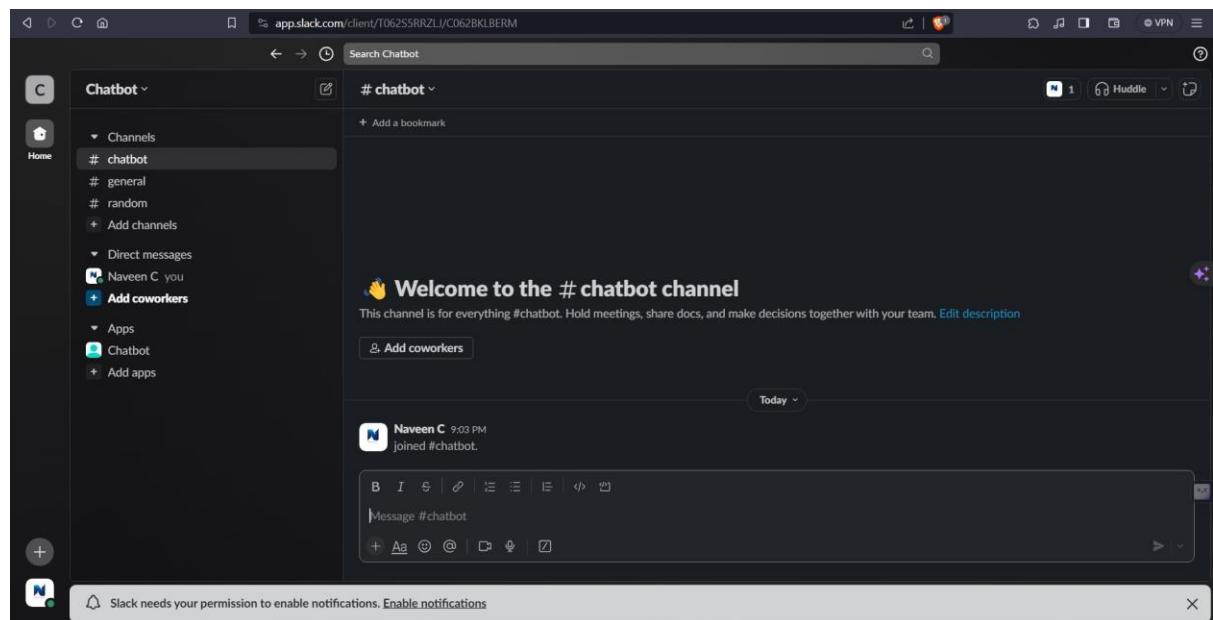
The screenshot shows a Facebook profile page for a page named "Chatbot". The page has a blue circular profile picture with a white letter "C" and 0 likes, 0 followers. The top navigation bar includes "Search Facebook" and other social links. Below the profile picture, there's a message box with a blue arrow pointing right and the text "Switch into Chatbot's Page to start managing it.". The main content area has tabs for "Posts", "About", "Mentions", "Reviews", "Followers", "Photos", and "More". Under the "Posts" tab, it says "No posts available". On the right side, there's a live chat window with a blue arrow pointing down and the text "hi". The chat transcript shows the bot responding with "Hello! How can I assist you today?". A user message "I want to buy a product" is followed by a response from the bot: "Sure, I can help you with that. Great! Please enter the product you want to buy?". Below the chat window, there are icons for "Friendly Greeting", "Create", and other messaging options.

To integrate IBM Watson Assistant chatbot into Slack, please adhere to the following step-by-step instructions:

1. If you do not already have a Slack account, begin by signing up for one.



2. Create a Workspace that corresponds to your chatbot.



3. Proceed to the Slack API website and create a Slack app specifically for your chatbot.

The screenshot shows the 'Your Apps' section of the Slack API documentation. On the left, there's a sidebar with various API categories like Start learning, Automation, Authentication, and APIs. The main area features a 'Build something amazing.' callout with a 'Create an App' button. Below it is a 'Your App Configuration Tokens' section with a 'Generate Token' button. At the bottom, there's a note about switching workspaces and a Windows taskbar at the bottom.

4. Access the Slack app settings page and navigate to the Basic Information tab. Locate the App Credentials section and copy your verification token.

The screenshot shows the 'App Credentials' tab in the Slack app settings. It displays the App ID (A062S39JRJ7), Date of App Creation (October 25, 2023), Client ID (6094195883698.6094111637619), and Client Secret (redacted). The Client Secret has 'Show' and 'Regenerate' buttons. Below the Client Secret, there's a note about sending it with OAuth requests. The Signing Secret is also shown with 'Show' and 'Regenerate' buttons. The Verification Token is displayed as VlDzI8FaVFXrbT4RvicnhMP5 with a 'Regenerate' button. At the bottom, there are 'Discard Changes' and 'Save Changes' buttons.

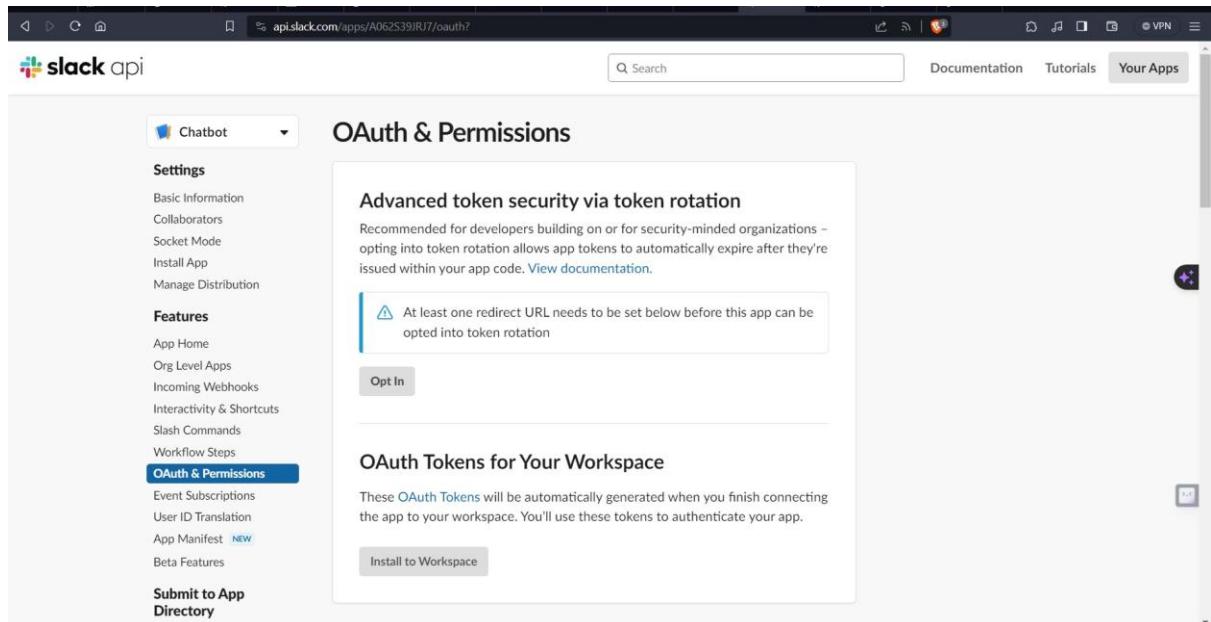
The screenshot shows the IBM Watson Assistant Lite interface with the Slack app selected. The main content area is titled "Connect watsonx Assistant to Slack". It provides instructions to copy the verification token from the Slack app settings page. A text input field contains a redacted verification token. Below the token field, step 2 of 2 is listed: "Go to the OAuth & Permissions tab. In the Bot Token Scopes section click Add an OAuth Scope, and then select the following scopes:" followed by a list of scopes: "app_mentions:read", "chat:write", "im:history", "im:read", and "im:write".

5. Head to the OAuth & Permissions tab. Under the Bot Token Scopes section, select "Add an OAuth Scope" and include the following scopes:

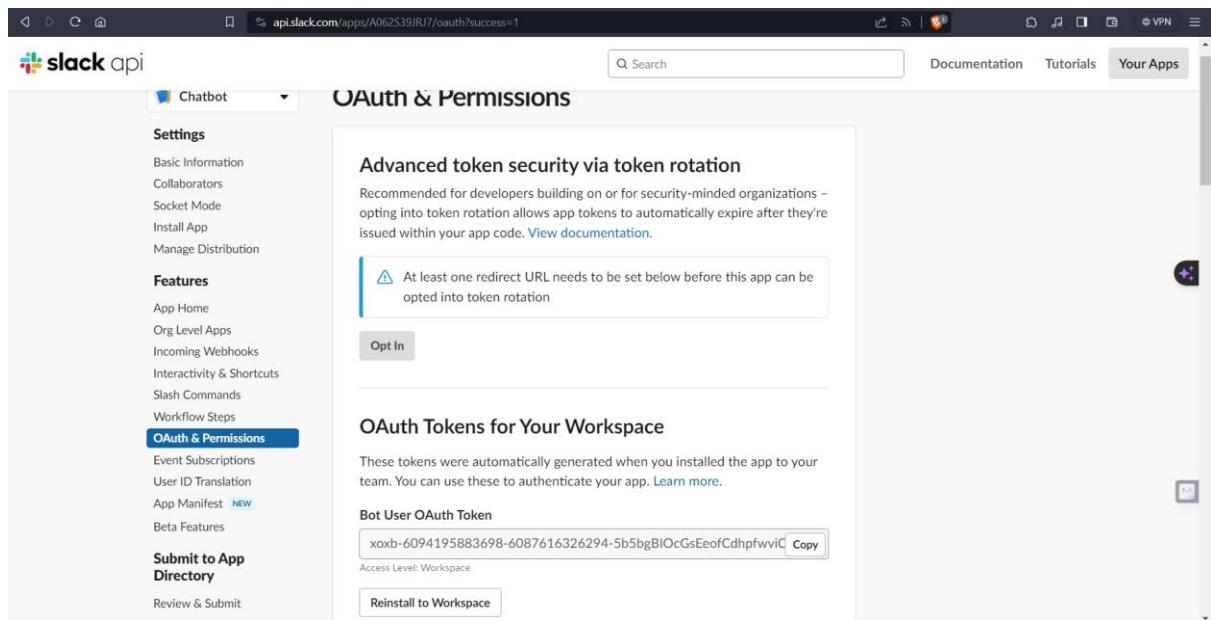
- app_mentions:read
- chat:write
- im:history
- im:read
- im:write

The screenshot shows the Slack OAuth & Permissions configuration page at api.slack.com/apps/A062S39JU7/oauth?. The "Bot Token Scopes" section is expanded, showing five available scopes: "app_mentions:read", "chat:write", "im:history", "im:read", and "im:write". Each scope has a description and a "View" link. At the bottom of the list is a button labeled "Add an OAuth Scope".

6. Visit the OAuth & Permissions tab. Click on "Install App to Workspace" and grant the necessary permissions by selecting "Allow." Afterward, you will be redirected back to the OAuth & Permissions page. Copy and paste your Bot user OAuth access token.



The screenshot shows the Slack API OAuth & Permissions page. On the left, there's a sidebar with sections like Chatbot, Settings, Features, and OAuth & Permissions (which is currently selected). The main content area has a heading "OAuth & Permissions". Below it is a section titled "Advanced token security via token rotation" with a note about redirect URLs. A prominent "Install to Workspace" button is located in the "OAuth Tokens for Your Workspace" section. The URL in the browser bar is `api.slack.com/apps/A062539IRJ7/oauth?`.



This screenshot shows the same OAuth & Permissions page after the "Install to Workspace" button was clicked. The "Bot User OAuth Token" field now contains a long string of characters, which is the copied access token. The URL in the browser bar includes the parameter `?success=1`. Other visible elements include the "Copy" button and the "Access Level: Workspace" label.

The screenshot shows a browser window for the IBM Watson Assistant Lite interface. The URL is eu-gb.assistant.watson.cloud.ibm.com/cn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aeu-gb%3Aa%2F02f47ceb4.... The page title is "IBM Watson Assistant Lite". The main content area is titled "Slack" with a "Draft" status. It contains instructions for step 2 and step 3 of the OAuth & Permissions setup. Step 2 lists scopes: app_mentions:read, chat:write, im:history, im:read, and im:write. Step 3 instructs to click "Install App to Workspace" and "Allow", then copy the "Bot user OAuth access token" from the clipboard icon. A placeholder text field is shown for pasting the token.

7. Return to the Slack app settings page and access the Event Subscriptions tab. Enable the Events toggle by switching it to the On position. Then, paste your request URL in the designated text field.

The screenshot shows a browser window for the Slack API app settings at api.slack.com/apps/A062539JU7/event-subscriptions. The page title is "slack api". The left sidebar shows "Chatbot" selected under "Settings". The main content is the "Event Subscriptions" tab. It has an "Enable Events" section with a toggle switch set to "On". Below it is a "Request URL" input field containing "https://my.app.com/slack/action-endpoint". A note explains that Slack will send HTTP POST requests to this URL for events. There's also a "New event authorization format" section with a note about recent changes to Events API payloads. At the bottom are "Discard Changes" and "Save Changes" buttons.

8. While on the Event Subscriptions tab, locate the "Subscribe to Bot Events" section. Click "Add Bot User Event" and select the desired event types you wish to subscribe to.

It is mandatory to choose at least one of the following types:

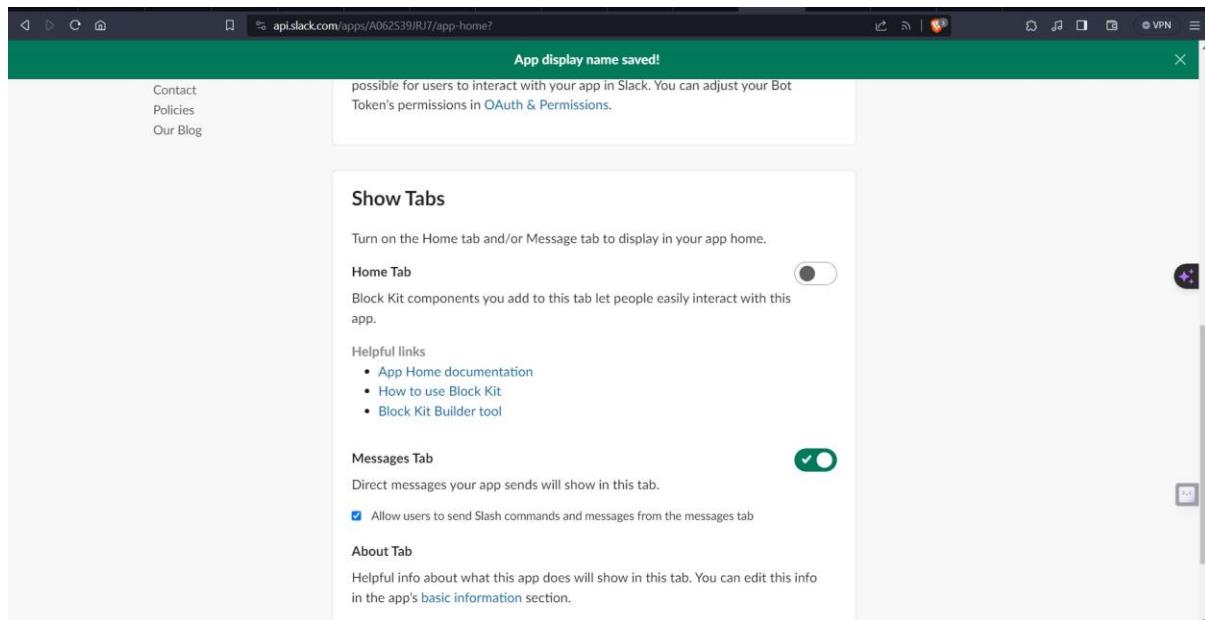
- message.im: Listens for message events within direct message channels.
 - app_mention: Listens exclusively for message events that mention your app or bot.
- Be sure to save your changes.

The screenshot shows the 'Event Subscriptions' section of the Slack App configuration. It includes a sidebar with links like 'Event Subscriptions', 'User ID Translation', 'App Manifest', 'Beta Features', 'Submit to App Directory', 'Review & Submit', 'Give feedback', and 'Slack Help'. The main area has a note about recent changes to Events API payloads. It shows two selected event types: 'message.im' (a message was posted in a direct message channel) and 'app_mention' (Subscribe to only the message events that mention your app or bot). There are 'Add Bot User Event' and 'Save Changes' buttons at the bottom.

9. Proceed to the App Home tab. Click "Edit" and provide a display name and default username for your virtual assistant. Save your changes and activate the "Always Show My Bot as Online" toggle.

The screenshot shows the 'App Home' configuration page. The sidebar includes 'Basic Information', 'Collaborators', 'Socket Mode', 'Install App', 'Manage Distribution', 'Features' (with 'App Home' selected), 'Org Level Apps', 'Incoming Webhooks', 'Interactivity & Shortcuts', 'Slash Commands', 'Workflow Steps', 'OAuth & Permissions', 'Event Subscriptions', 'User ID Translation', 'App Manifest', 'Beta Features', 'Submit to App Directory', 'Review & Submit', 'Give feedback', and 'Slack Help'. The main area shows 'Where people find your app on Slack' (Home, Messages, About tabs) and 'Your App's Presence in Slack' (display name: Chatbot, default name: chatbot). It also features a 'Bot Token Added' section and an 'Always Show My Bot as Online' toggle switch, which is turned on.

10. On the App Home tab, locate the Show Tabs section. Enable the Messages Tab toggle and check the "Allow users to send Slash commands and messages from the messages tab" checkbox.



Finally, it is essential to thoroughly test your chatbot application on Messenger to ensure seamless functionality.

