**PROJECT6:CHATBOT DEPLOYMENT WITH IBM**

**CLOUD WATSON ASSISTANT**

### Phase 1: Problem Definition and Design Thinking

**Designing and deploying a chatbot involves several key steps, and applying a design thinking approach can be highly beneficial to ensure the bot addresses the right problem and meets user needs effectively. Here's a step-by-step guide on how to use design thinking for chatbot deployment**.

**Problem Definition: Chatbots are Artificially-Intelligent Softwares showing responsiveness towards the existing queries. Those queries may either be of a student or an employee, depending upon the requirements in real-times. But this project will promisingly let the beginners prepare a software whose database is cloud-based and let a student get responses for the queries he/she may enter. NLP i.e Natural Language Processing and ML i.e Machine Learning algorithms, a list of appropriate responses will be entered in those chatbots which will answer the questions (these are the input patterns) of a student in a goal-oriented manner.**

**Design Thinking:**

1. **Understand your target audience: Understand your target audience: Begin by empathizing with your users.**
2. **Define the problem**: **Based on your research, clearly define the problem or challenge the chatbot is meant to address.**
3. **Generate ideas: Encourage creativity and consider different approaches, features, and functionalities for the chatbot**.
4. **Create a low-fidelity prototype**: **Develop a basic version of the chatbot, focusing on its conversational flow and user interface.**
5. **Build the chatbot**: **Once you have a refined design, proceed with development.**
6. **Launch the chatbot: Deploy the chatbot on the chosen platform(s), such as a website, messaging app, or dedicated chatbot platform**.

**CHATBOT DEPLOYMENT WITH IBM CLOUD WATSON ASSISTANT**

**PHASE 2: INNOVATION**

consider implementing advanced features such as natural language understanding(NLU)for more accurate user internet recognition.

1. **Data Collection and Annotation:**
   * Collect a diverse dataset of user interactions with your AI system.
   * Annotate the data with information about user intents, entities, and context.
2. **Preprocessing:**
   * Tokenize and preprocess the user input to break it down into words, phrases, and sentences.
   * Remove stop words and apply stemming or lemmatization to normalize text.
3. **Intent Recognition:**
   * Use machine learning models such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), or transformers like BERT to recognize user intents.
   * Train the model on your annotated dataset to classify user inputs into different intent categories.
4. **Entity Recognition:**
   * Implement entity recognition to identify specific pieces of information within the user input (e.g., dates, names, locations, products).
   * You can use named entity recognition (NER) models or custom rules for this purpose**.**
5. **Context Management:**
   * Implement context management to understand the flow of the conversation and maintain context between user turns.
   * Store information from previous interactions to provide more contextually relevant responses**.**
6. **Testing and Evaluation:**
   * Thoroughly test your NLU system to ensure it can accurately recognize user intents and entities in various real-world scenarios.
7. **Privacy and Security:**
   * Ensure that you are handling user data securely and respecting privacy regulations, especially when dealing with sensitive information**.**
8. **Scalability:**
   * Design your NLU system to scale with increased usage and demand**.**
9. **User Experience Design:**
   * Integrate NLU seamlessly into your user interface to provide a smooth and intuitive conversational experience.

# IBM NAAN MUDHALAVAN PROJECT

**CLOUD APPLICATION DEVELOPMENT PHASE-3 DEVELOPMENT PART-1**

Building a chatbot using IBM Cloud Watson Assistant involves several steps, including creating an instance of Watson Assistant, designing your chatbot's dialog flow, integrating it with various channels, and training it to understand and respond to user input effectively.

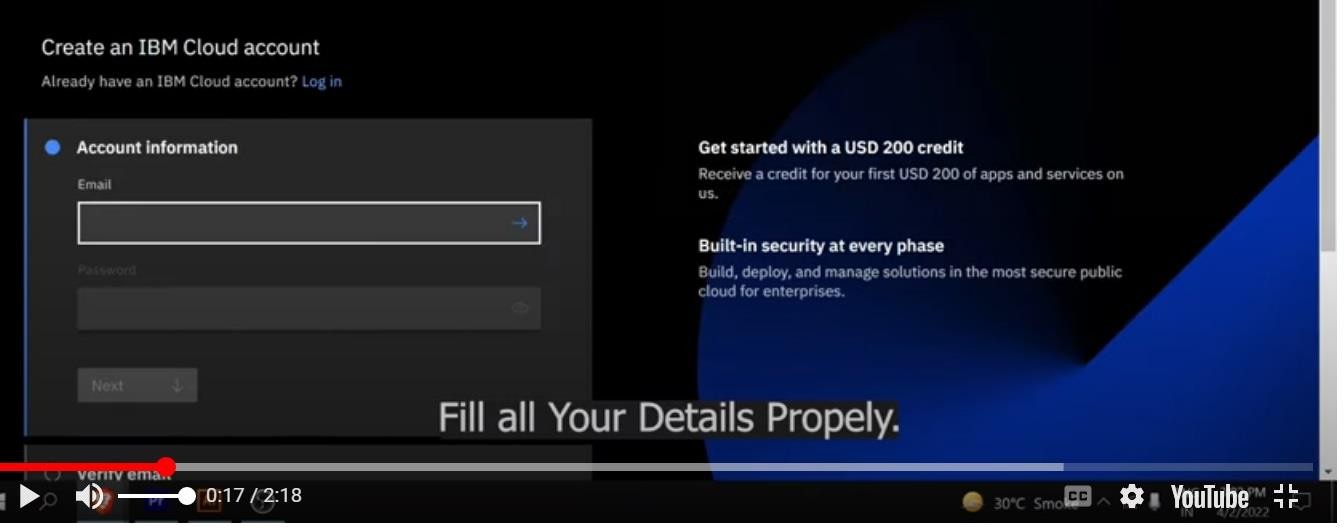
Here's a step-by-step guide to help you get started:

**Step 1: Sign Up for IBM Cloud**

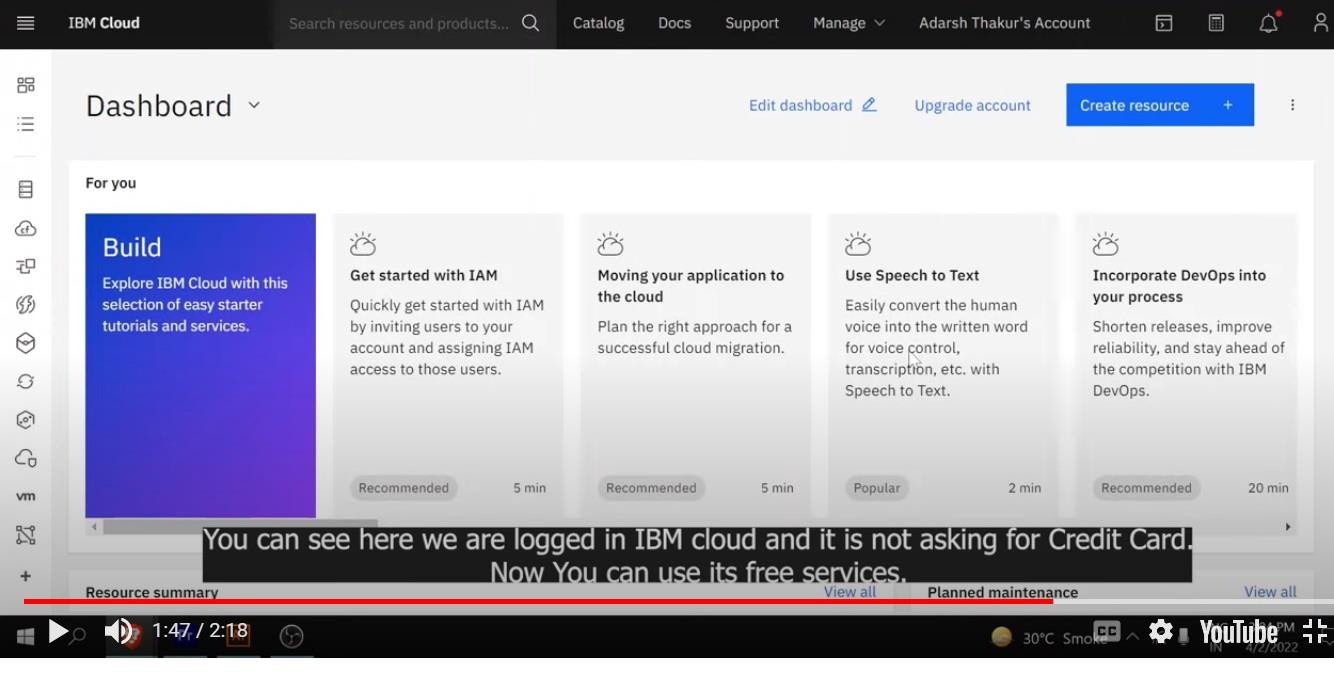
If you don't already have an IBM Cloud account, sign up for one at [https://cloud.ibm.com/registration.](https://cloud.ibm.com/registration)

**Step 2: Create an Instance of Watson Assistant**

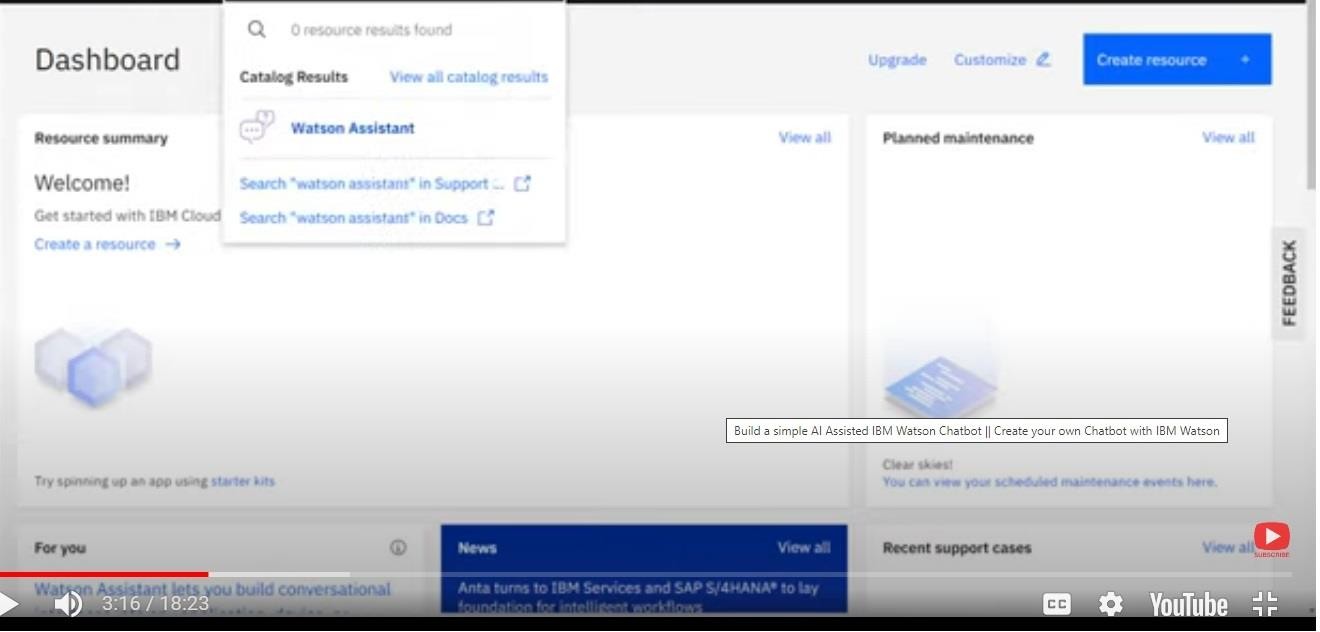
1. Log in to your IBM Cloud account.



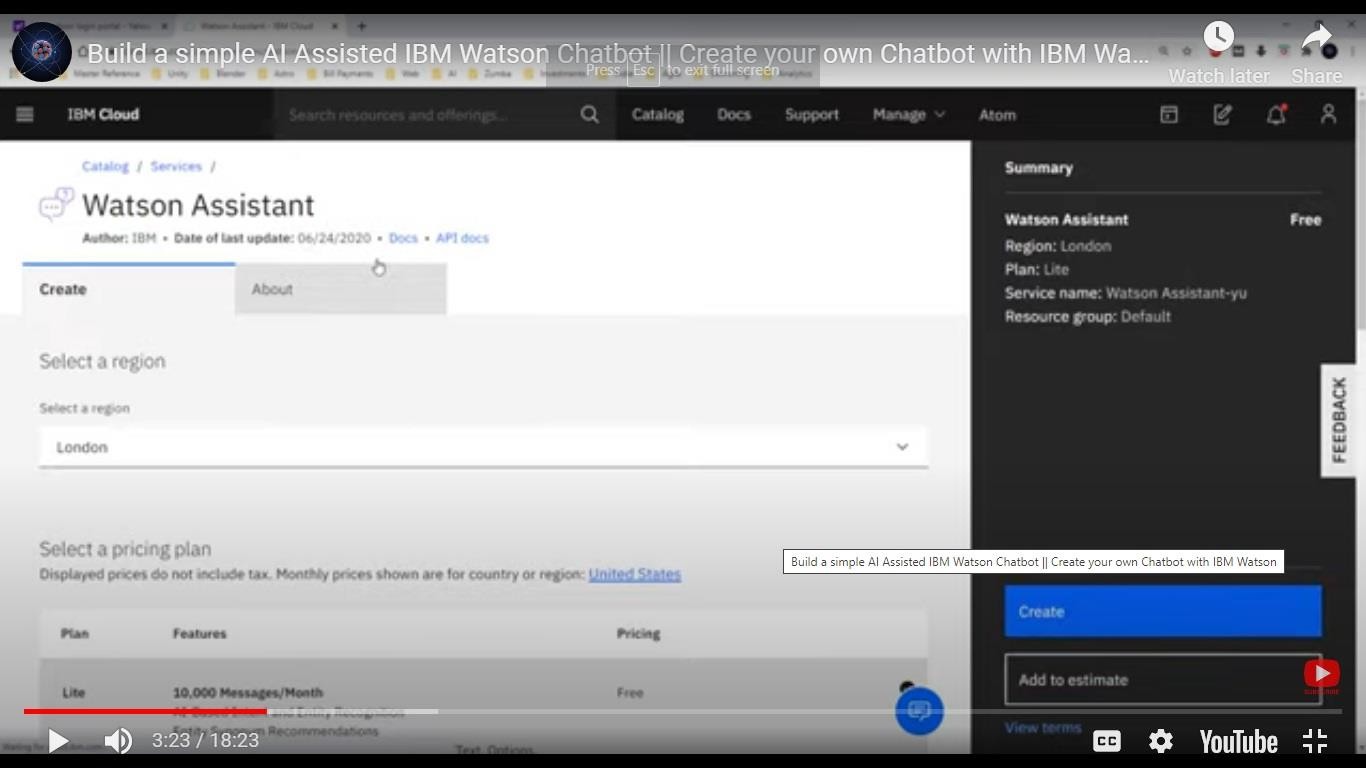
1. Go to the IBM Cloud dashboard and click "Create Resource."



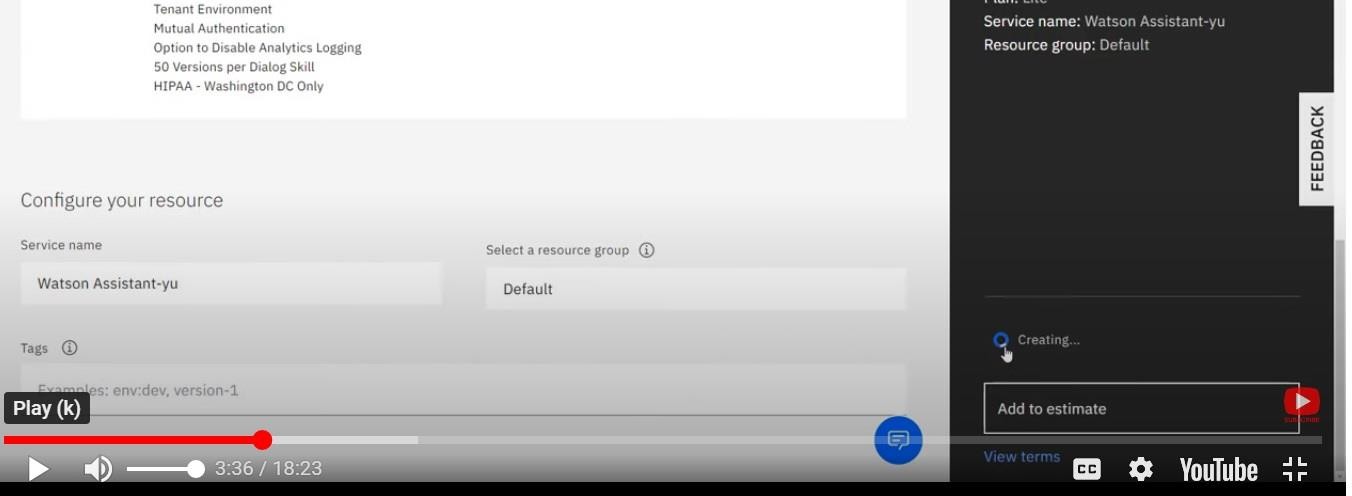
1. In the IBM Cloud Catalog, search for "Watson Assistant."



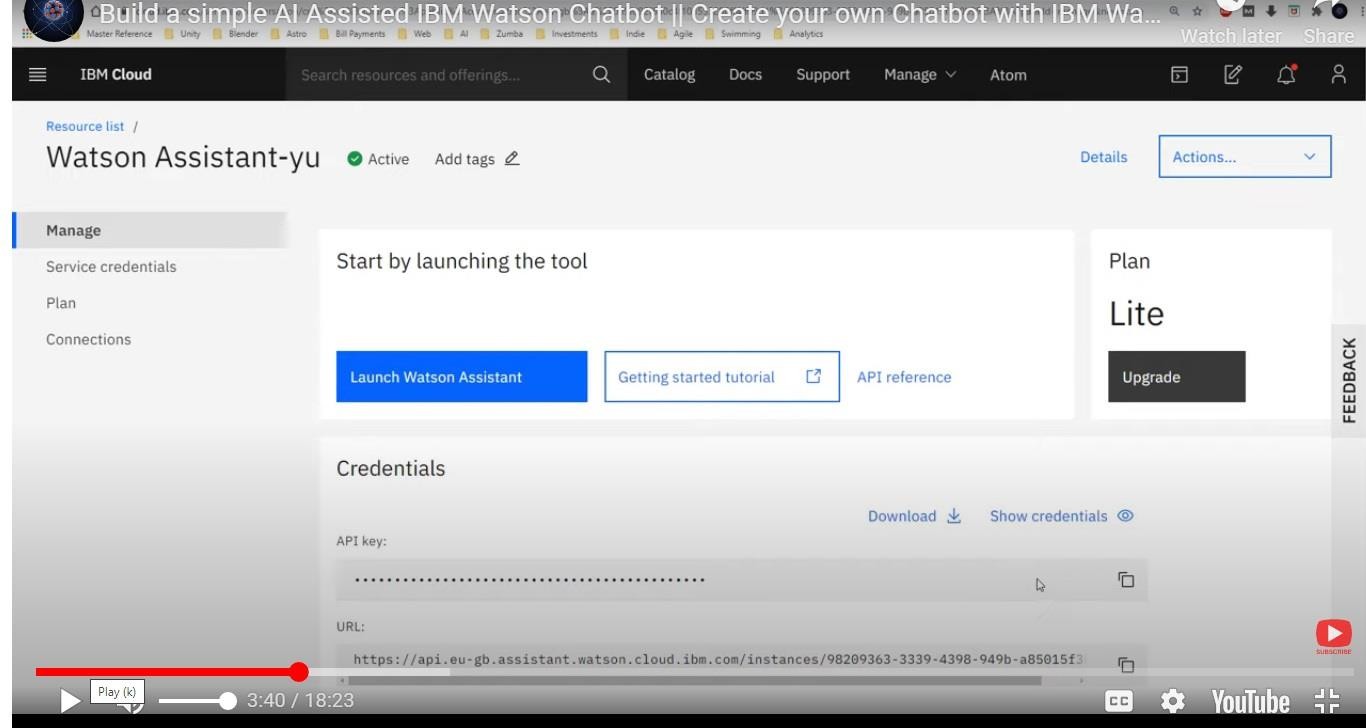
1. Select "Watson Assistant" from the search results.



1. Configure your service by giving it a name, selecting the region, and choosing the appropriate plan (Lite plan is free to use).

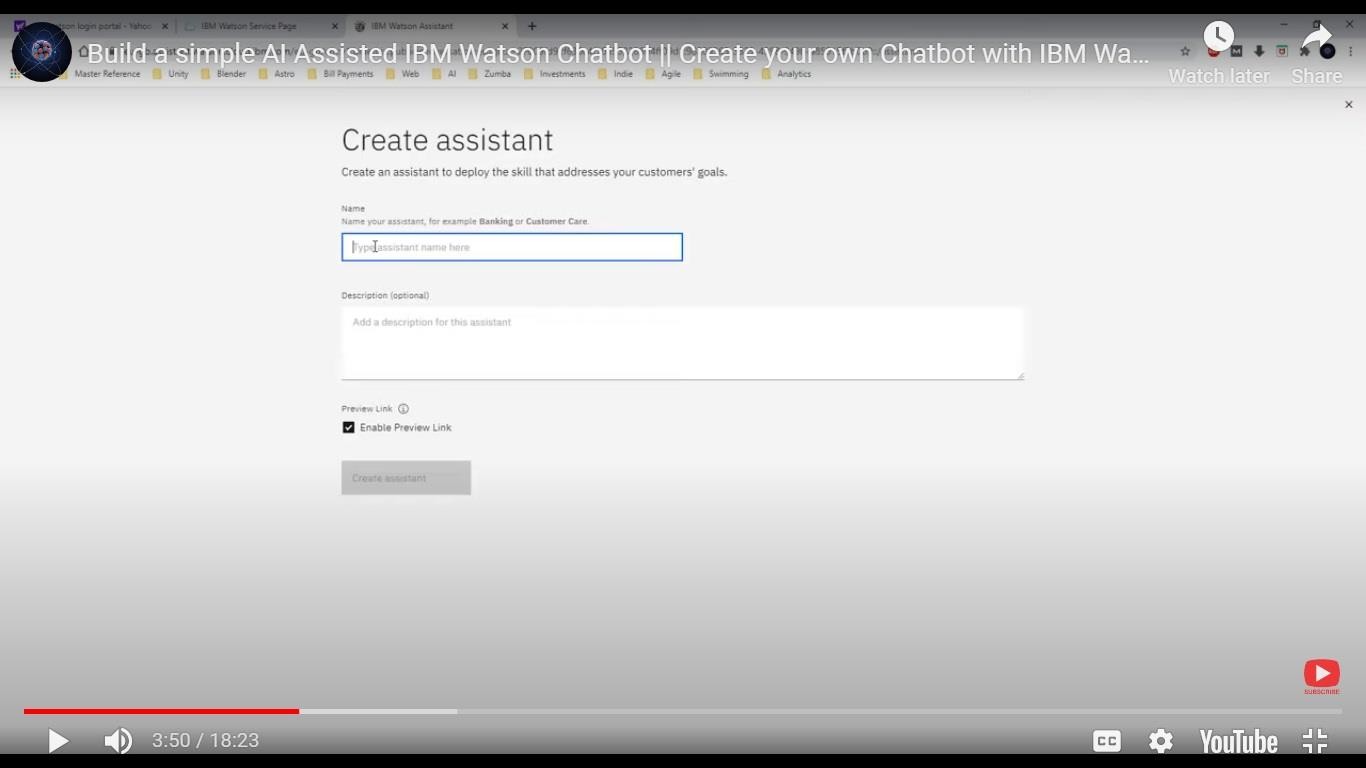


1. Click "Create" to create your Watson Assistant service instance.



### Step 3: Create a Watson Assistant Skill

* 1. Once your Watson Assistant service instance is created, click on it to open the Watson Assistant dashboard.



* 1. Inside the Watson Assistant dashboard, click on "Launch Watson Assistant" to open the Watson Assistant tool.
  2. In the Watson Assistant tool, click on "Skills" in the left menu.
  3. Click on "Create skill" to start building your chatbot's conversation flow.

## Step 4: Build Your Chatbot's Dialog Flow

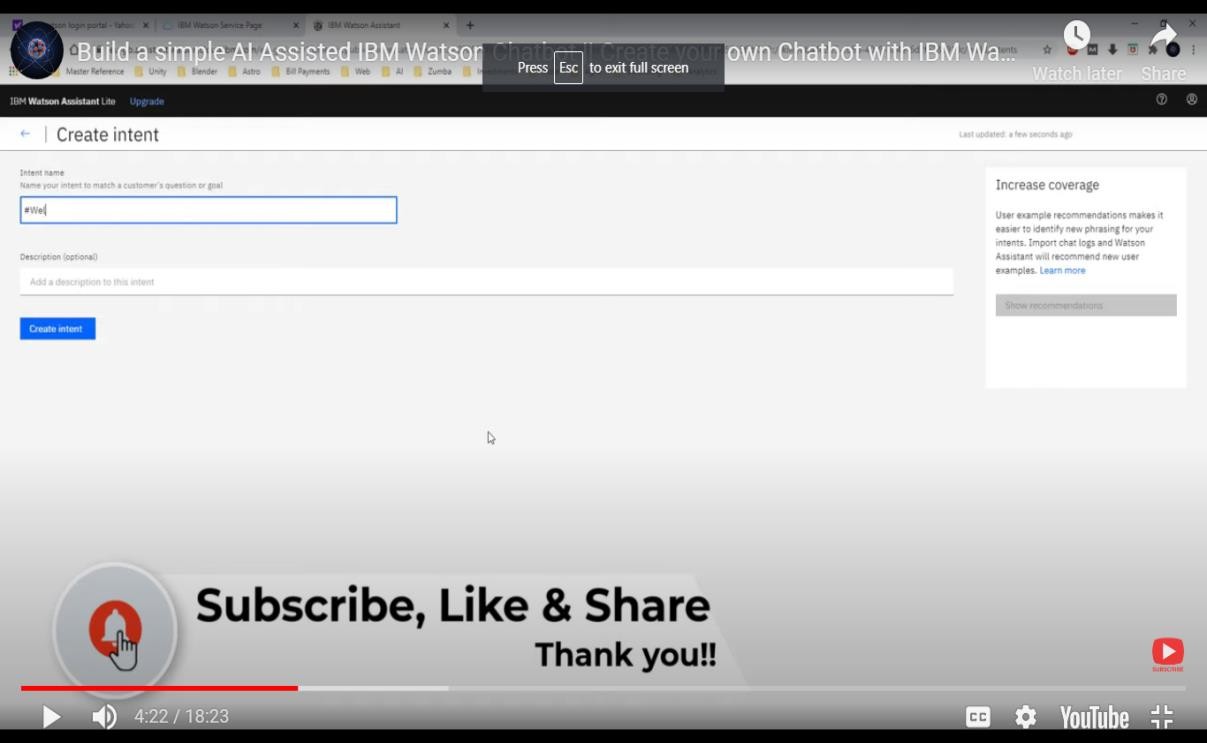
* + 1. In your newly created skill, you can add Intents, Entities, and Dialog nodes to design your chatbot's conversation flow.

\***Intents**: Define the user's intentions or what they might be trying to do.

\***Entities**: Define specific terms or objects the chatbot should recognize.

\***Dialog Nodes**: Create the responses your chatbot will provide based on user input, Intents, and Entities.

* + 1. You can use the built-in Dialog Builder to visually design your chatbot's conversation flow by creating dialog nodes and connecting them.
    2. Make sure to test your chatbot as you build it to ensure it understands and responds correctly.



## Step 5: Integrating Your Chatbot

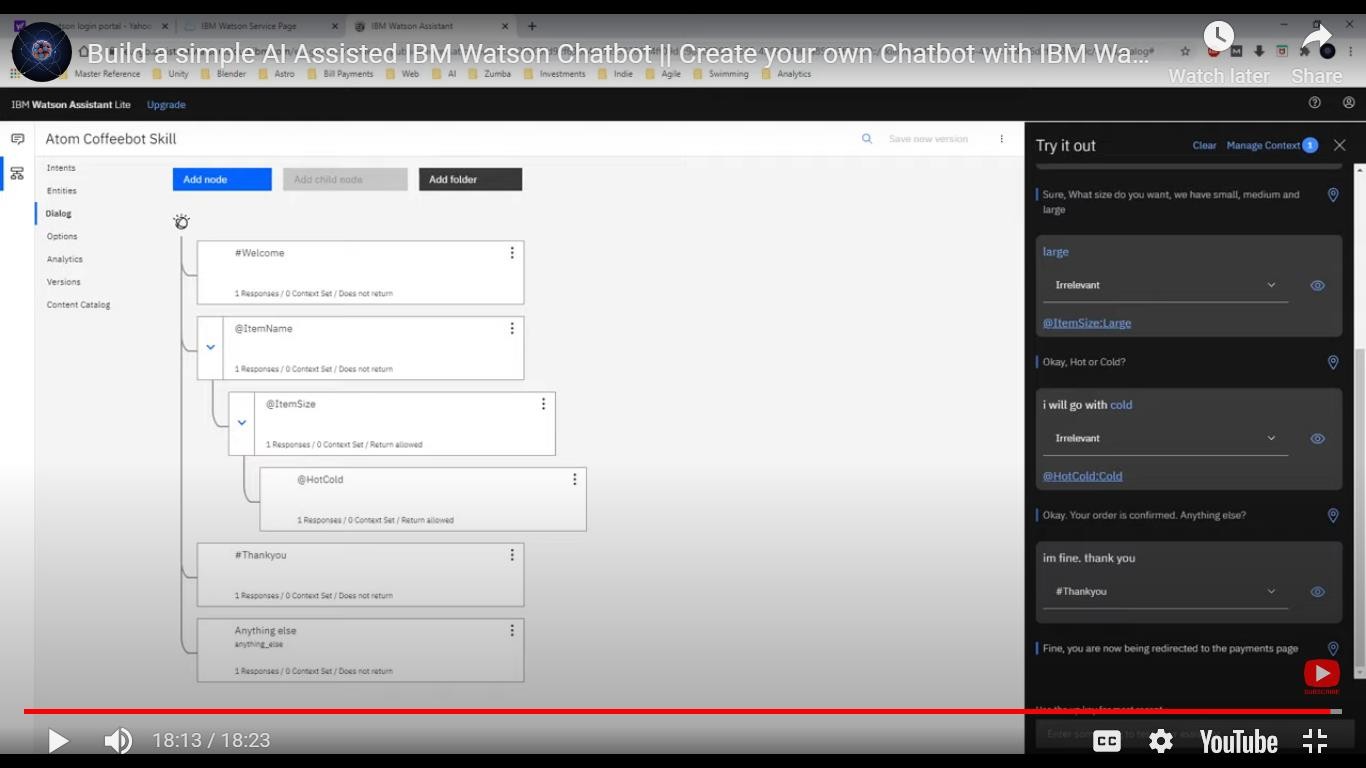
* + - 1. You can integrate your chatbot with various channels, such as a website, mobile app, or messaging platform. Here's how to do it for a website:
      2. In the Watson Assistant tool, click on "Options" in the left menu and select "Integration."
      3. Choose the integration channel you want (e.g., Web Chat).
      4. Configure the settings and customize the chat widget's appearance to match your website's design.
      5. Generate the code and add it to your website to embed the chatbot.

## Step 6: Train and Deploy Your Chatbot

1. After building and testing your chatbot, make sure it understands user input correctly and provides relevant responses.
2. Train your chatbot by providing a variety of user examples for each intent to improve its natural language understanding.
3. Once satisfied with the performance, click "Deploy" to make your chatbot live.

## Step 7: Monitor and Improve

1. Continuously monitor your chatbot's performance, user interactions, and feedback. You can use Watson Assistant's analytics to understand how your chatbot is being used and identify areas for improvement.



1. That's a basic overview of how to build a chatbot using IBM Cloud Watson Assistant. You can explore more advanced features like entity extraction, integration with databases, and using external APIs to enhance your chatbot's capabilities.

## IBM NAAN MUDHALVAN PROJECT

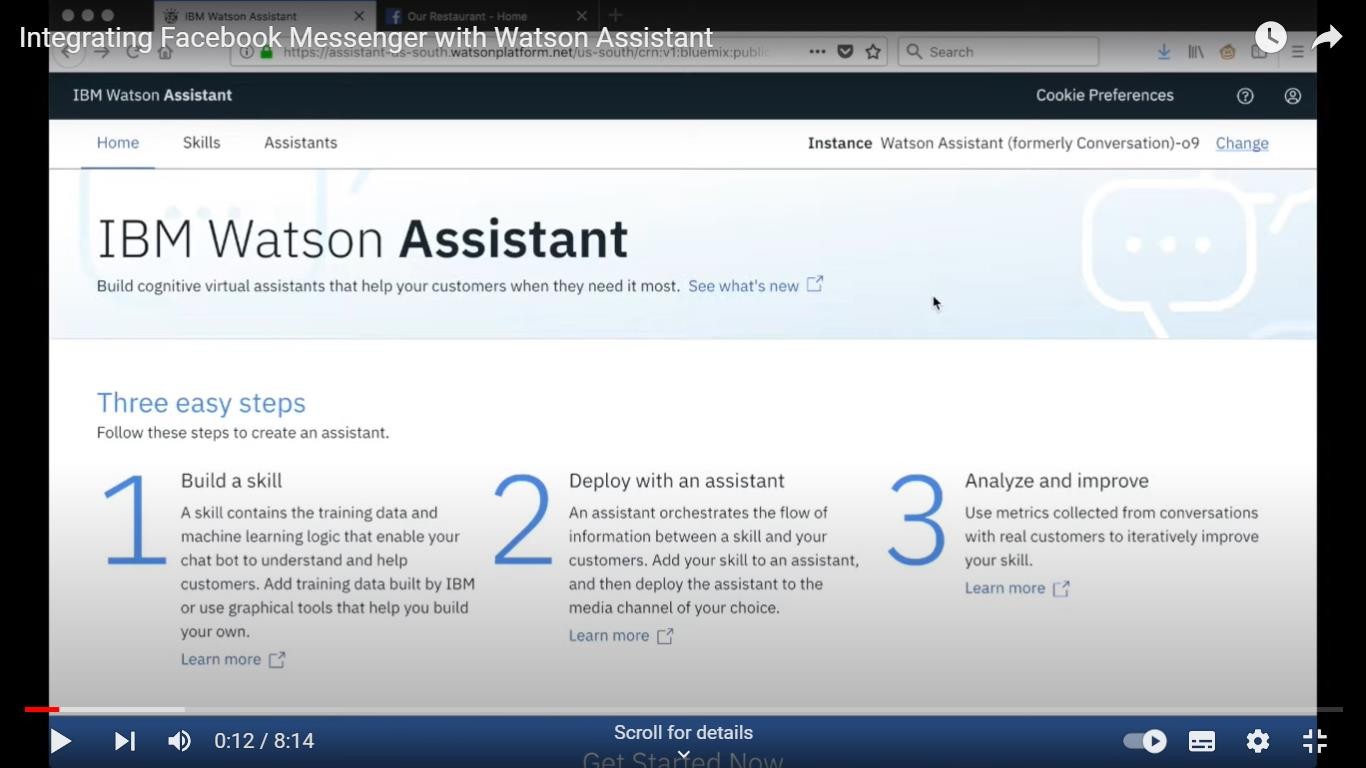
**CLOUD APPLICATION DEVELOPMENT**

**PHASE-4 DEVELOPMENT PART-2**

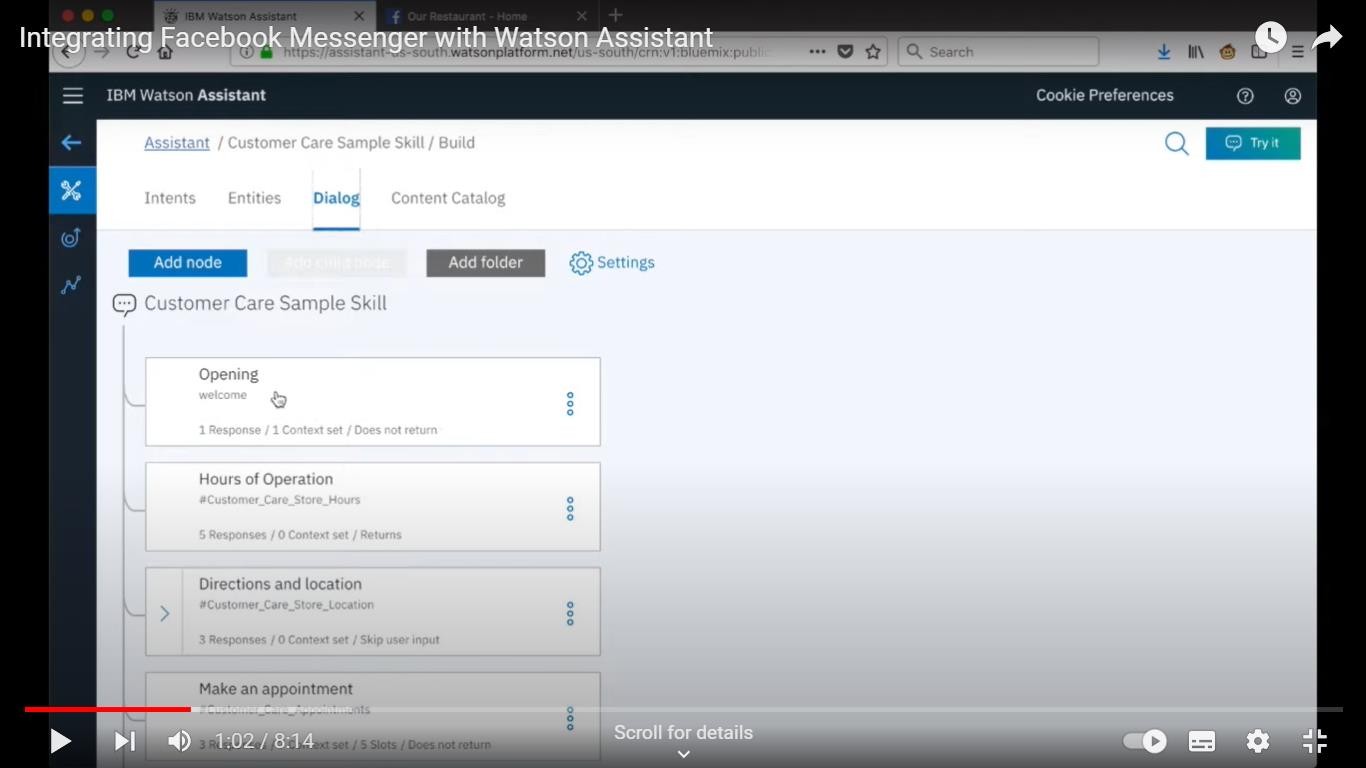
Integrate the chatbot with facebook Messenger and Slack using respective API.

Ensure that the conversation flows naturally and that the chatbot responses are informative and accurate.

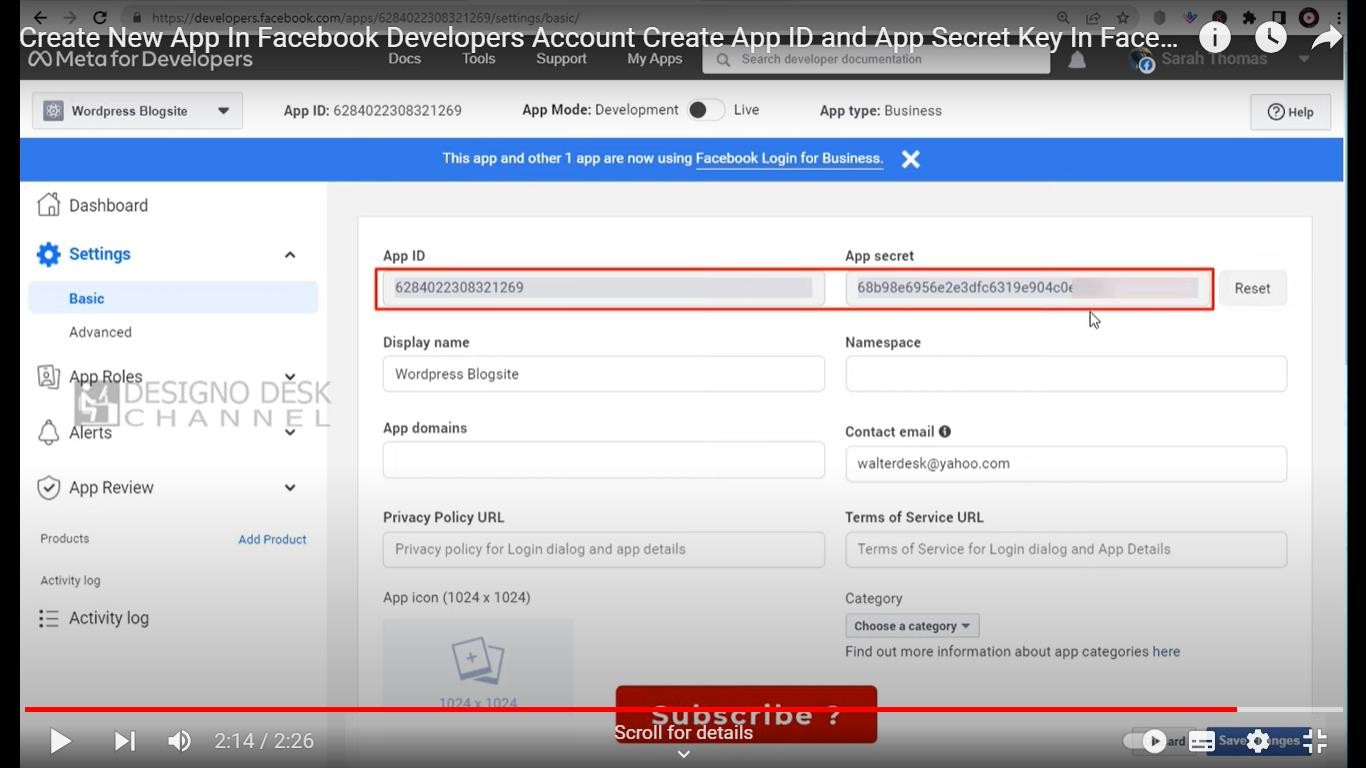
1. **Create a Watson Assistant instance:**
   * If you don't already have a Watson Assistant instance, create one on the IBM Cloud platform.



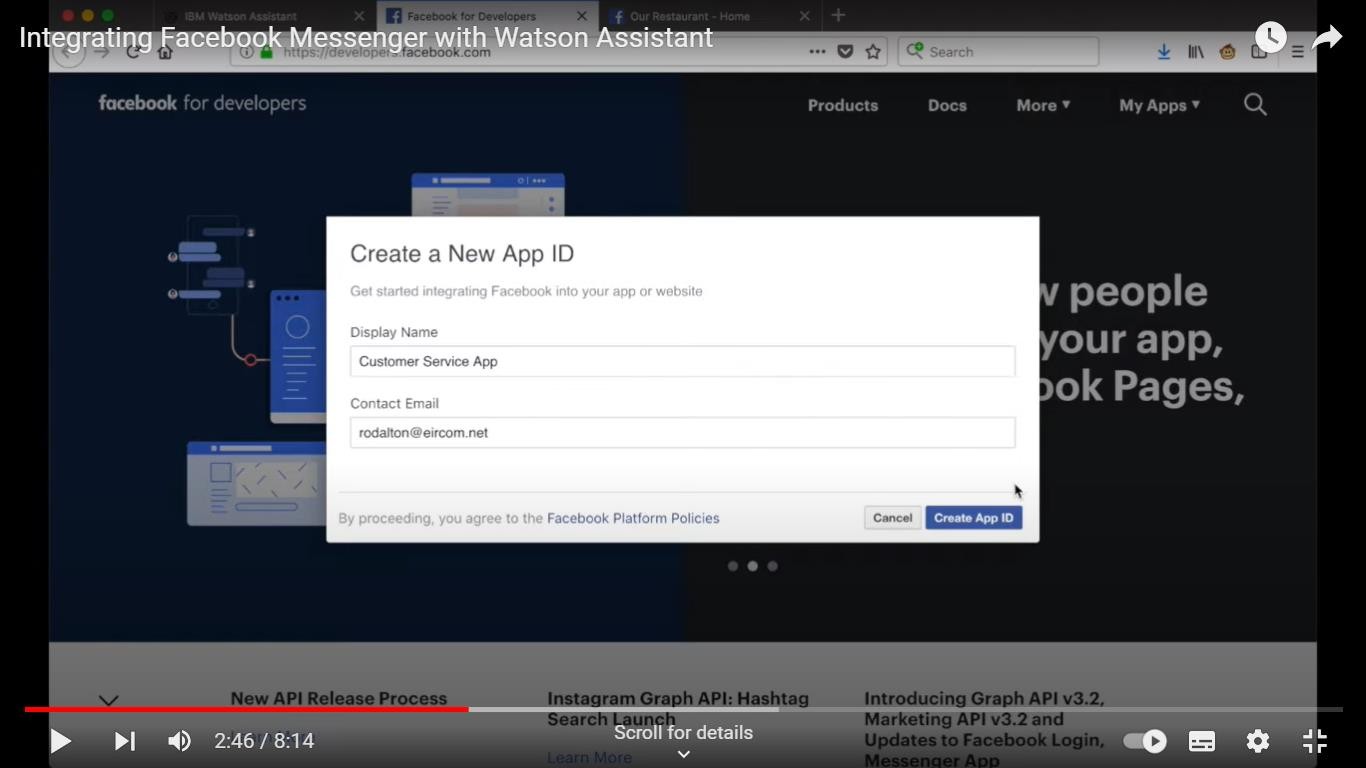
1. **Build and Train Your Chatbot:**
   * Create and train your chatbot using the Watson Assistant interface. Define intents, entities, and dialog flows.



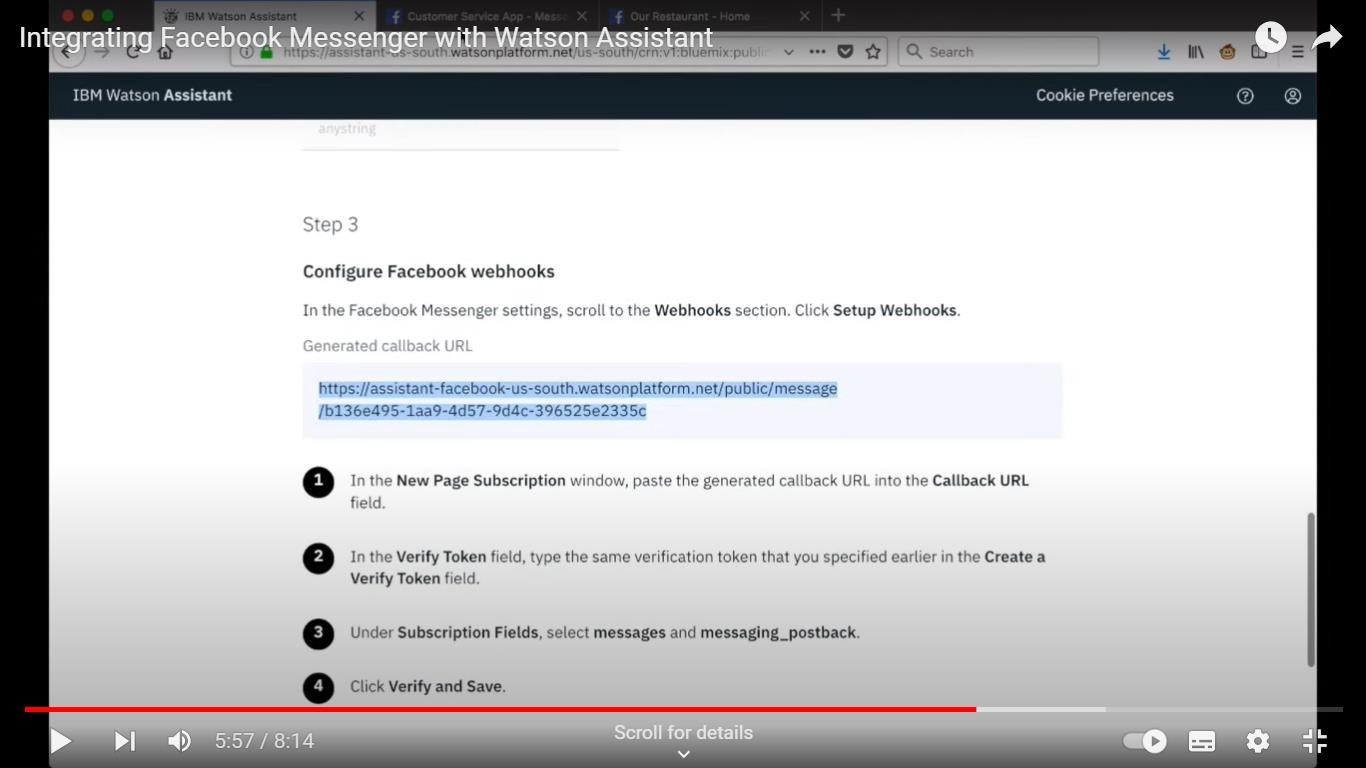
1. **Set Up Facebook Developer Account**:
   * Go to the Facebook for Developers platform (https://developers.facebook.com/), create a new app, and set up a Facebook Page

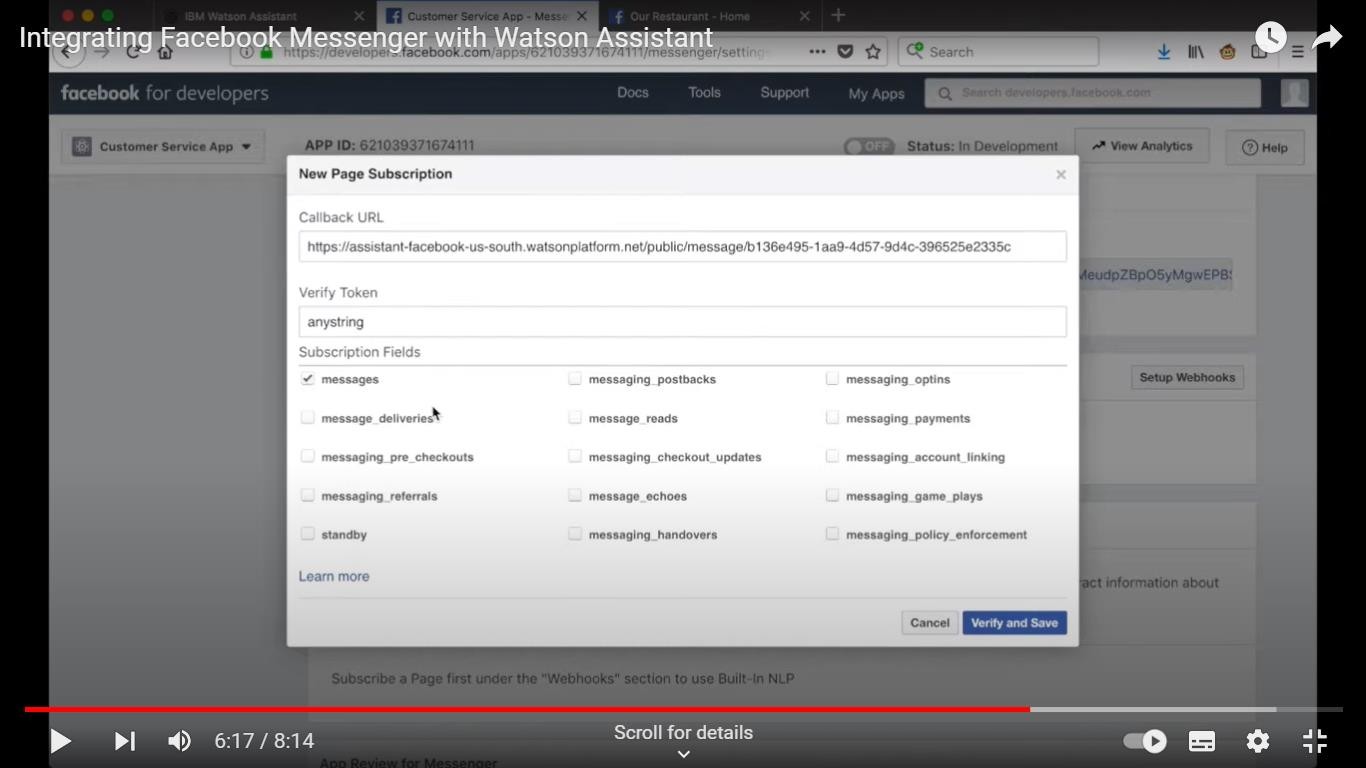


1. **Connect Watson Assistant with Facebook Messenger**
   * In the Watson Assistant dashboard, navigate to the "Integrations" section.
   * Select "Facebook Messenger" as an integration channel.
   * Follow the instructions to link your Watson Assistant instance to your Facebook Page and generate a Facebook Page Access Token.

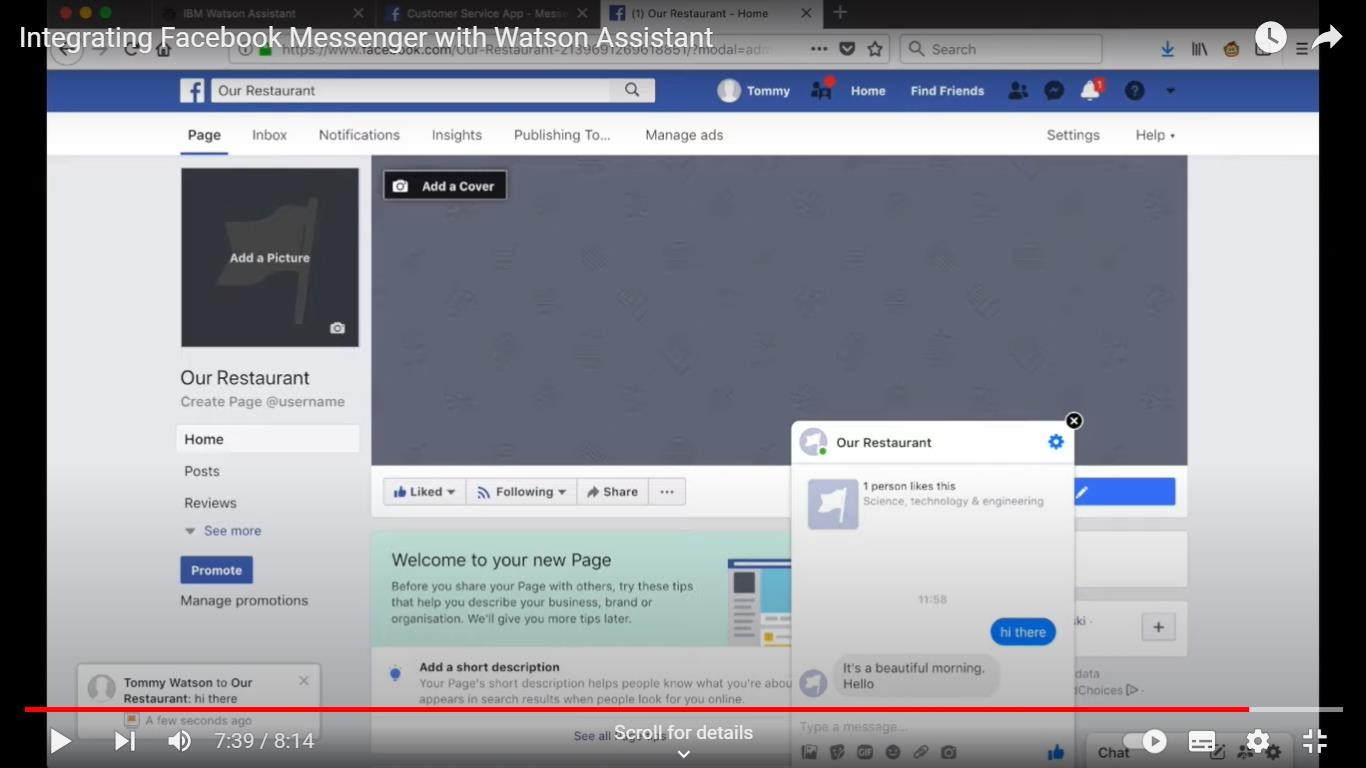


1. **Configure Webhooks:**
   * Configure the webhook in your Facebook App settings to point to the URL provided by Watson Assistant.



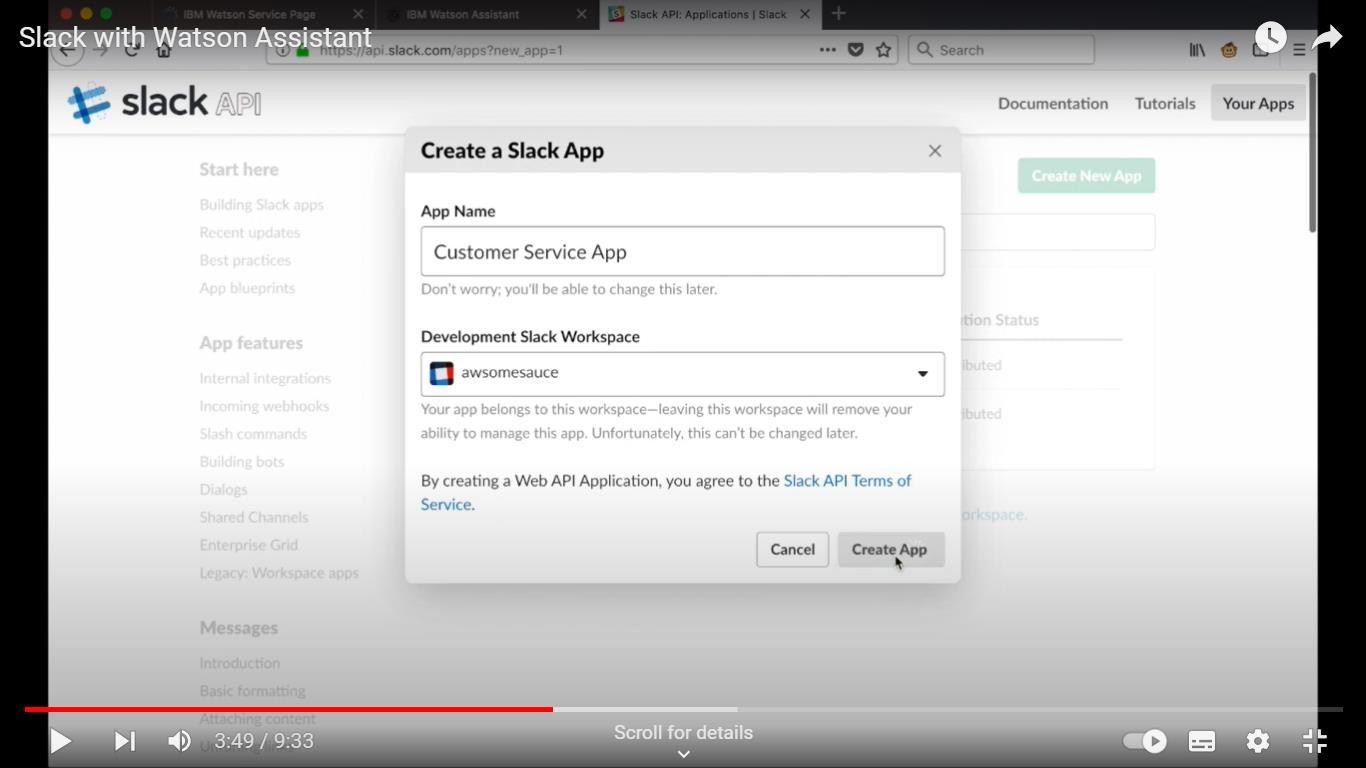


1. **Test and Deploy:**
   * Test your chatbot on Facebook Messenger. Once you're satisfied with the performance, deploy it for public access.

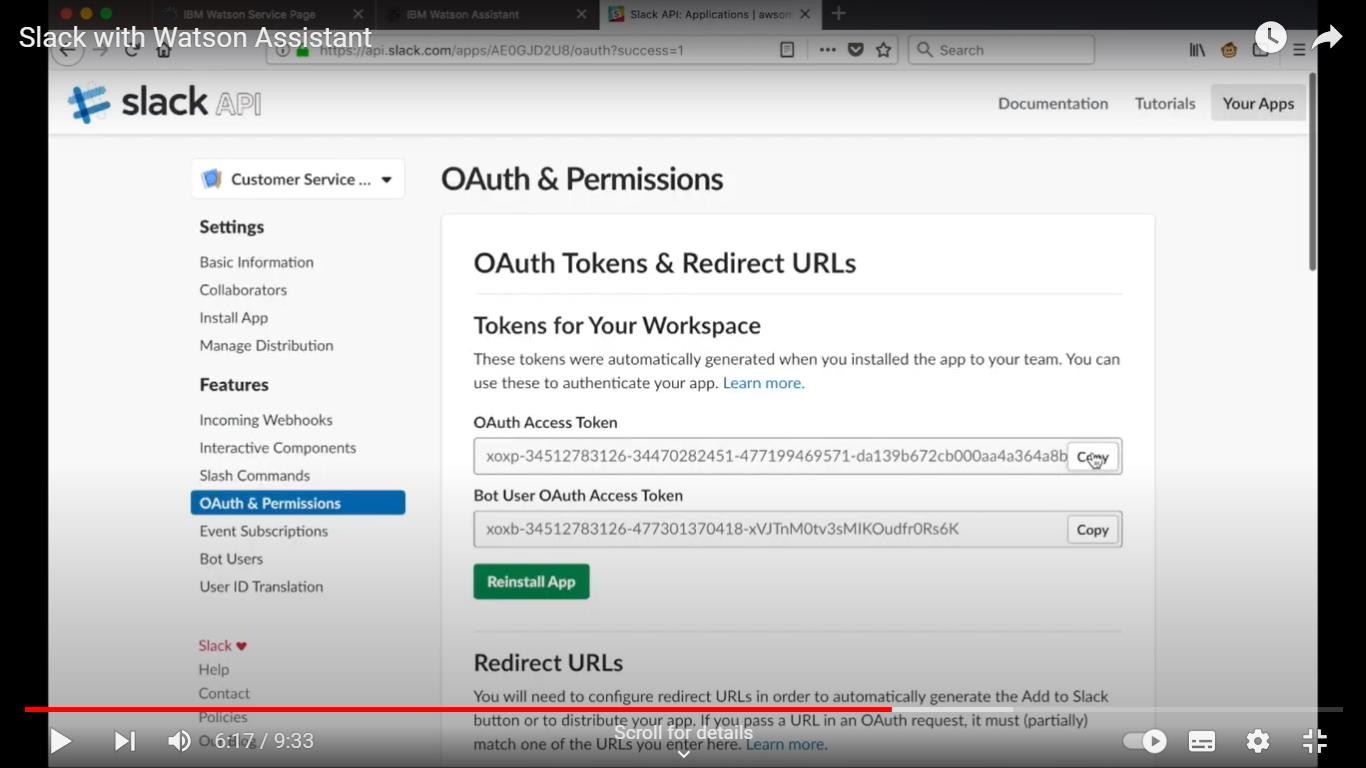


### Integrating Watson Assistant with Slack:

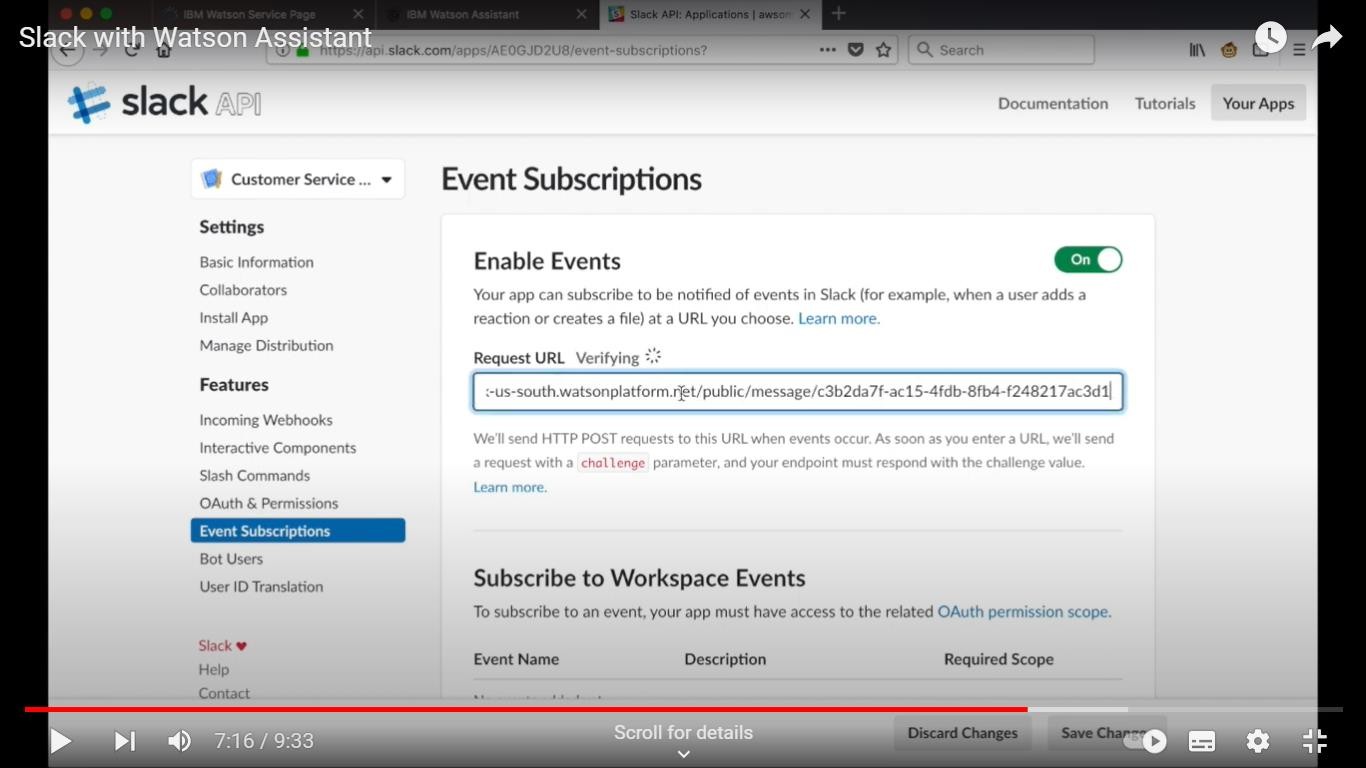
1. **Create a Watson Assistant instance:**
   * If you haven't already, create a Watson Assistant instance on the IBM Cloud platform.
2. **Develop Your Chatbot:**
   * Create your chatbot in Watson Assistant, training it with the necessary intents, entities, and dialog.
3. **Create a Slack App:**
   * Go to the Slack API portal (https://api.slack.com/apps) and create a new app for your chatbot



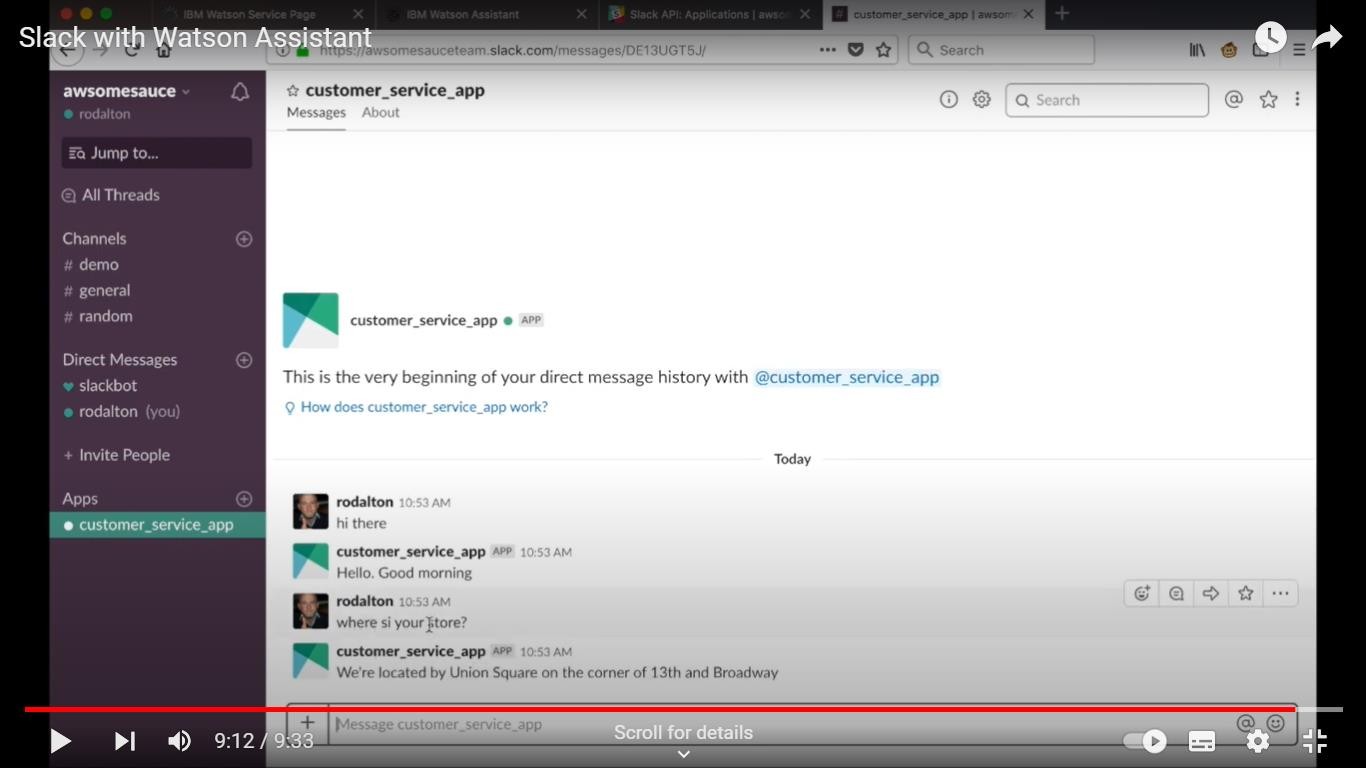
1. **Configure OAuth and Permissions:**
   * In your Slack app's settings, configure OAuth scopes and permissions to allow your app to interact with Slack workspaces.



1. **Install Your App:**
   * Install your app in the Slack workspace where you want to use the chatbot. This will generate an OAuth access token.
2. **Set Up Event Subscriptions:**
   * Configure event subscriptions in your Slack app to listen for messages and other relevant events in the workspace.



1. **Integrate with Watson Assistant:**
   * Develop a middleware or server that connects your Slack app to Watson Assistant using the Watson Assistant API.
2. **Handle Events and Messages:**
   * Handle incoming events and messages from Slack and send them to Watson Assistant. Process responses from Watson Assistant and send them back to Slack.



1. **Deploy Your Chatbot:**
   * Host your middleware or server on a platform that's accessible from the internet.
2. **Test and Distribute:**
   * Test your chatbot in the Slack workspace and distribute it to other users if desired.