



# Build your First Chatbot with Watson and WhatsApp

Open Lab | Digital Summit 2019



# Build your First Chatbot with Watson and WhatsApp

## Introduction

This document contains a step-by-step process of creating a Chatbot with IBM Watson Assistant NLP and will teach you how to create a chatbot with Watson Assistant and how to integrate with WhatsApp using Node JS.

This guide was prepared by [Miracle's Innovation Labs](#)

## Pre-Requisites

All attendees must have their workstation (with Internet) to participate in the lab (Both PC and MAC are compatible). The following pre-requisites will help you to make the Hands-on Lab experience easier.

- Active email ID for registering with IBM Bluemix to get access for Watson Assistant
- Download and install Node JS and ngrok
- Access for WhatsApp Web
- Test Editor such as Sublime Text (or) Notepad++

## Technology Involved

- IBM Bluemix (PaaS)
- NLP - IBM Watson Assistant
- Server Side - Node JS

## Labs Steps

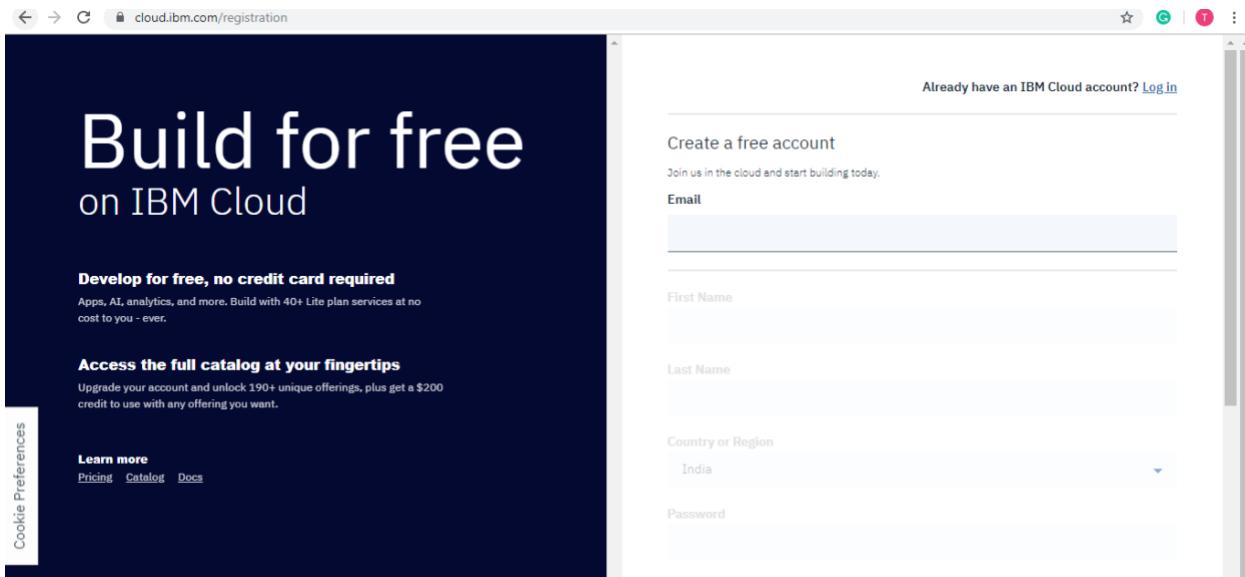
So, let us get started with the bot!

The following steps will outline creating a chatbot using Watson Assistant and integrate it with WhatsApp using Twilio. Users will be able to directly message your bot through WhatsApp to know about Miracle Software Systems, AP Cloud and Digital Summit.

### Step #1 | Create Watson Assistant Service in IBM Bluemix

The first step will be to make sure that we have access to the IBM Bluemix console with either free trial option (or) the paid subscription option.

Login to Bluemix at <https://cloud.ibm.com/> (or) Register today at, <https://cloud.ibm.com/registration>. If you are a new user, provide the email ID.



Now, you need to provide the details to the fields that are marked as required (\*) and click on **Create Account**.

[cloud.ibm.com/registration](https://cloud.ibm.com/registration)

# Build for free on IBM Cloud

**Develop for free, no credit card required**  
Apps, AI, analytics, and more. Build with 40+ Lite plan services at no cost to you - ever.

**Access the full catalog at your fingertips**  
Upgrade your account and unlock 190+ unique offerings, plus get a \$200 credit to use with any offering you want.

[Learn more](#)

Pricing Catalog Docs

Already have an IBM Cloud account? [Log in](#)

---

Create a free account  
Join us in the cloud and start building today.

Email ✓  
DSIabdemos@gmail.com

First Name ✓  
Mani

Last Name ✓  
Nirjogi

Country or Region ✓  
India

Password ✓  
\*\*\*\*\*

IBM may use my contact data to keep me informed of products, services and offerings:

by email  by telephone

You can withdraw your marketing consent at any time by sending an email to [netsupp@us.ibm.com](mailto:netsupp@us.ibm.com). Also you may unsubscribe from receiving marketing emails by clicking the unsubscribe link in each such email.

More information on our processing can be found in the [IBM Privacy Statement](#). By submitting this form, I acknowledge that I have read and understand the IBM Privacy Statement.

I accept the product [Terms and Conditions](#) of this registration form.

[Create Account](#)

© Copyright IBM Corp. 2014, 2019. All rights reserved.

[cloud.ibm.com/registration](https://cloud.ibm.com/registration)

# Build for free on IBM Cloud

**Develop for free, no credit card required**  
Apps, AI, analytics, and more. Build with 40+ Lite plan services at no cost to you - ever.

**Access the full catalog at your fingertips**  
Upgrade your account and unlock 190+ unique offerings, plus get a \$200 credit to use with any offering you want.

[Learn more](#)

Pricing Catalog Docs

Nirjogi

Country or Region ✓  
India

Password ✓  
\*\*\*\*\*

IBM may use my contact data to keep me informed of products, services and offerings:

by email  by telephone

You can withdraw your marketing consent at any time by sending an email to [netsupp@us.ibm.com](mailto:netsupp@us.ibm.com). Also you may unsubscribe from receiving marketing emails by clicking the unsubscribe link in each such email.

More information on our processing can be found in the [IBM Privacy Statement](#). By submitting this form, I acknowledge that I have read and understand the IBM Privacy Statement.

I accept the product [Terms and Conditions](#) of this registration form.

[Create Account](#)

© Copyright IBM Corp. 2014, 2019. All rights reserved.

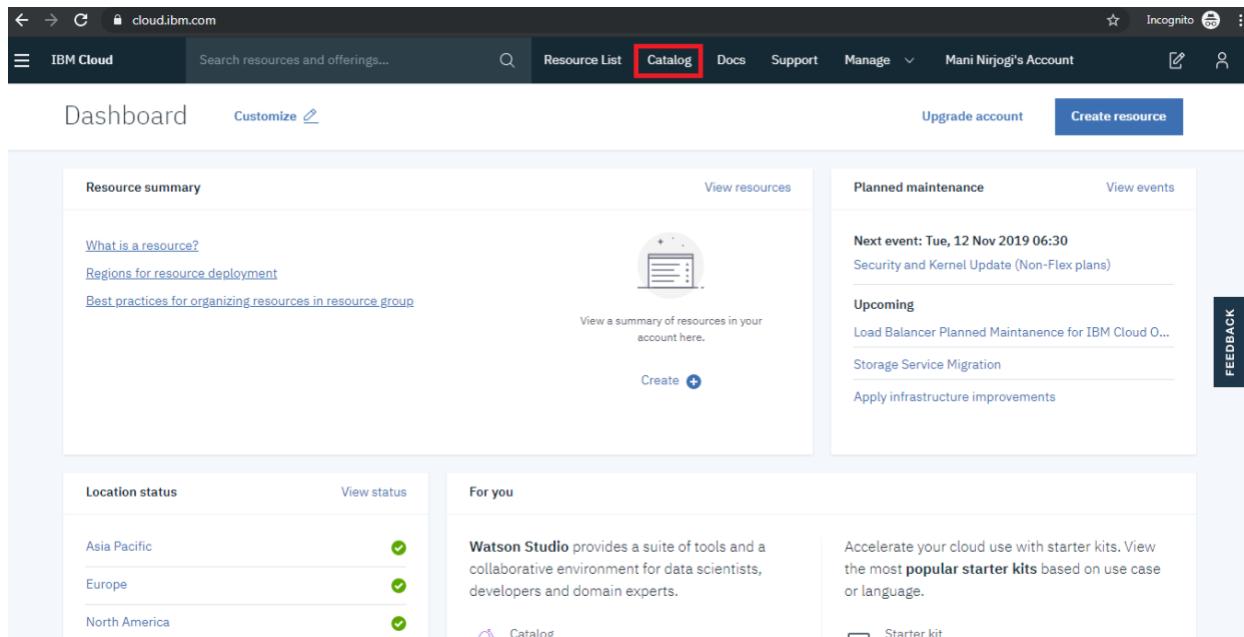
After successful creation of the account, a confirmation mail will be sent to the registered mail id. Click on **Confirm Account**.

The screenshot shows a Gmail inbox with one unread email from 'IBM Cloud <no-reply@cloud.ibm.com>' to 'DSlabdemos'. The subject of the email is 'Action required: Confirm your IBM Cloud account'. The email body starts with 'Hello Mani,' and includes a message: 'Thank you for signing up for IBM Cloud! Confirm your account to get started.' Below this is a red-bordered 'Confirm account' button. The text continues: 'By confirming your account, you accept the [Terms of Use](#). Welcome and happy building!' It ends with 'Thank you, IBM Cloud'. On the left sidebar, the 'Inbox' tab is selected, showing 1 new message. Other tabs include 'Starred', 'Snoozed', 'Sent', 'Drafts', and 'More'. The bottom left of the sidebar shows a profile picture for 'Mani'.

Now, Log in to your Bluemix account by providing your credentials.

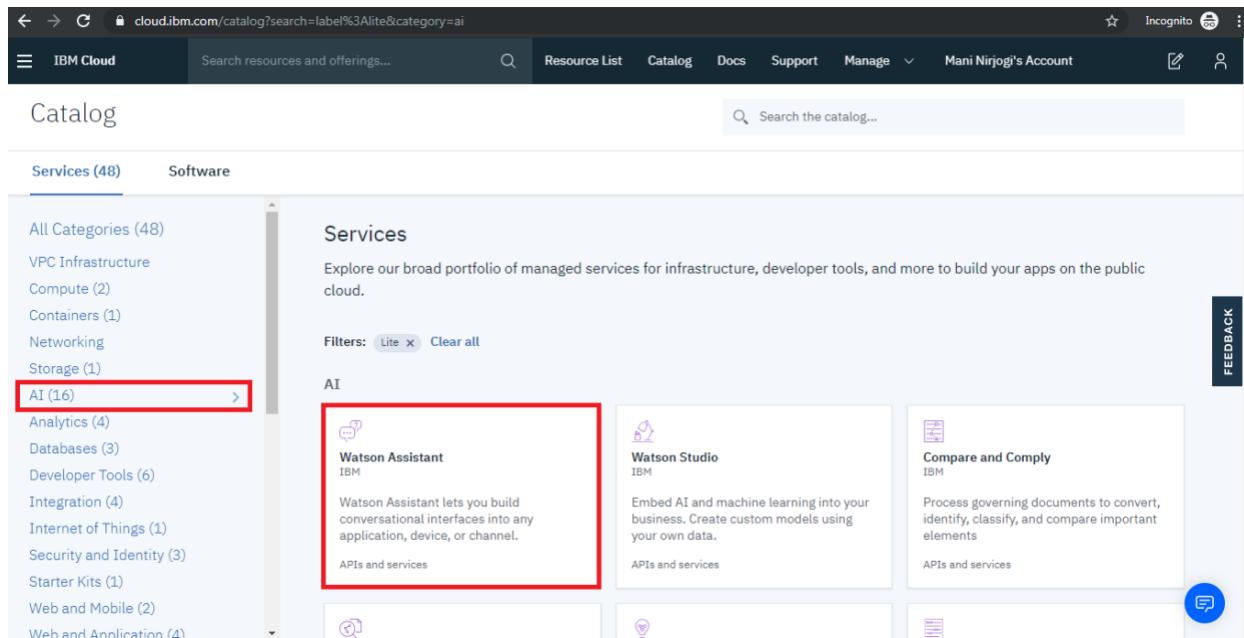
The screenshot shows the 'Log in to IBM Cloud' page. The left side features a dark blue header with the text 'Welcome to IBM Cloud' and 'Start building immediately using 190+ unique services.' Below this is a 'Create an IBM Cloud account' button and a 'Get a \$200 credit when you upgrade' section. The right side has a 'Log in to IBM Cloud' heading, an 'ID' field containing 'IBMid' and 'DSlabdemos@gmail.com', a 'Remember me' checkbox, a 'Password' field with a red border, and a 'Log in' button. Below the password field are links for 'Forgot ID?' and 'Forgot password?'

Once you are logged in, you can see the dashboard as shown below. Now, click on **Catalog**.



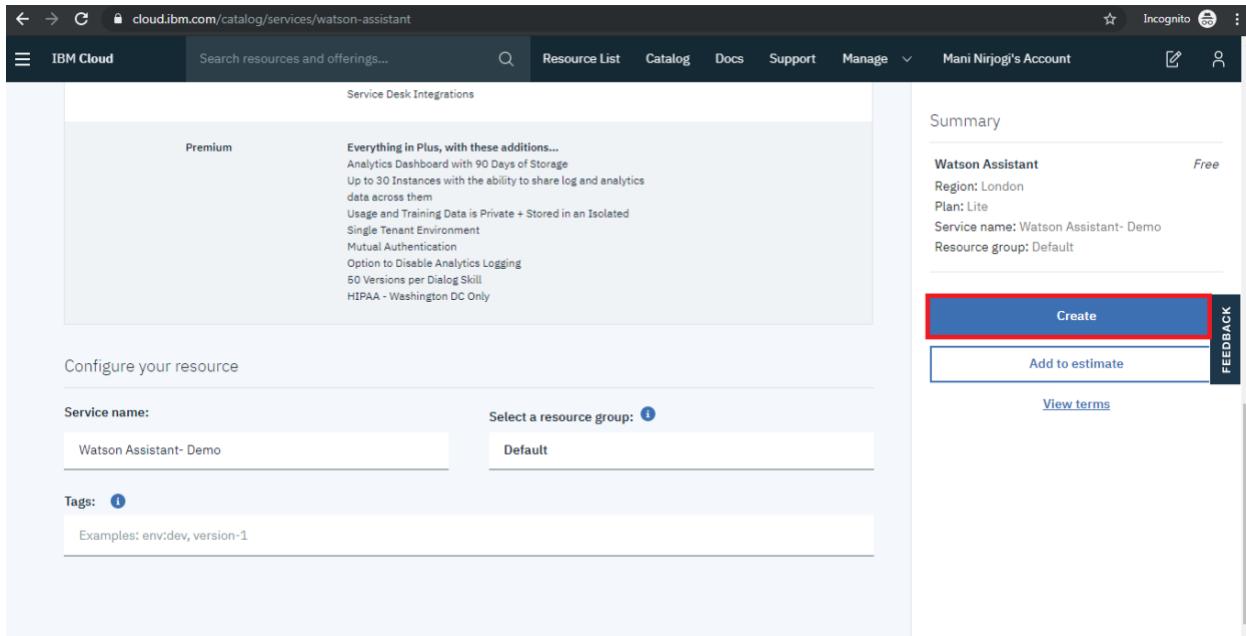
The screenshot shows the IBM Cloud dashboard. At the top, there's a navigation bar with links for Resource List, Catalog (which is highlighted with a red box), Docs, Support, Manage, and Mani Nirjogi's Account. Below the navigation bar is a search bar and a "Dashboard" section with a "Customize" link. To the right of the dashboard is a "Planned maintenance" section listing an event for Tuesday, November 12, 2019, at 06:30. Further down are sections for "Location status" (listing Asia Pacific, Europe, and North America) and "For you" (listing Watson Studio and Starter kit). A vertical "FEEDBACK" button is on the far right.

In Catalog, under the **AI** category you can see all the Watson Services where you can find **Watson Assistant**. Click on **Watson Assistant** tile for training the chatbot.



The screenshot shows the IBM Cloud Catalog page. The left sidebar has a "Services (48)" section with a "AI (16)" category highlighted with a red box. Other categories include All Categories (48), VPC Infrastructure, Compute (2), Containers (1), Networking, Storage (1), Analytics (4), Databases (3), Developer Tools (6), Integration (4), Internet of Things (1), Security and Identity (3), Starter Kits (1), Web and Mobile (2), and Web and Application (4). The main content area shows a "Services" section with a brief description and a "Filters" section with "Lite" and "Clear all" buttons. Below this is an "AI" section containing three tiles: "Watson Assistant" (highlighted with a red box), "Watson Studio", and "Compare and Comply". Each tile has a small icon, a title, a brief description, and a "APIs and services" link.

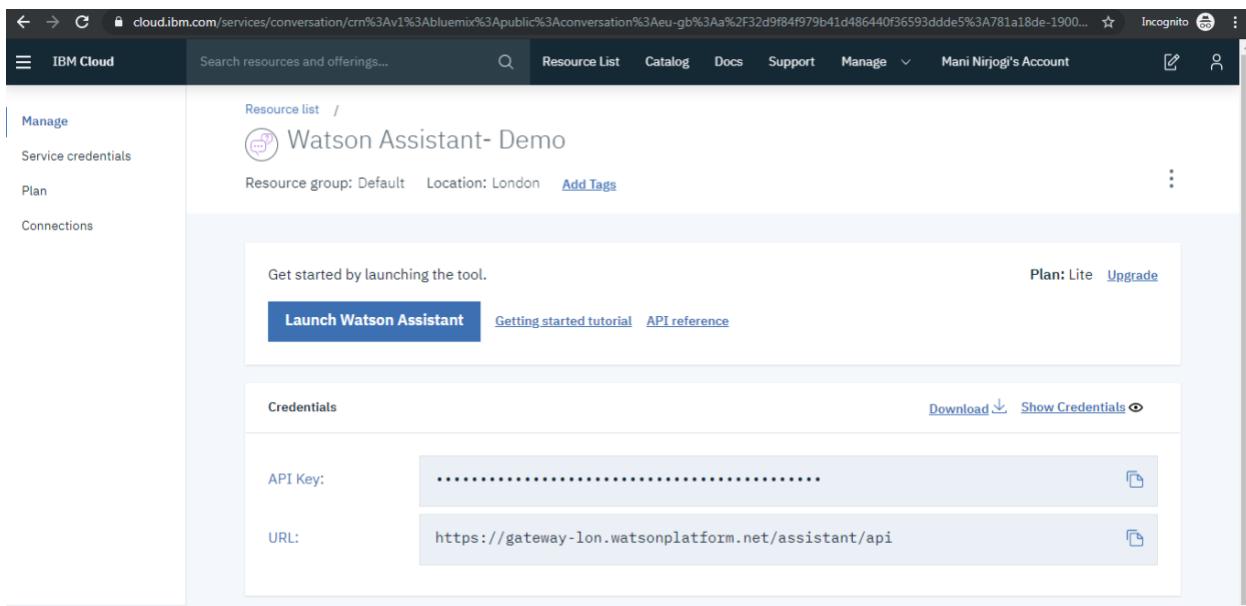
Now, type a unique name for the service instance in the **Service name** field. For example, type **my-conversation-demo**. Leave the default values for other options and click on **Create**.



The screenshot shows the IBM Cloud catalog interface. On the left, there's a sidebar with 'IBM Cloud' and a search bar. The main area displays a 'Service Desk Integrations' card for the 'Premium' plan. To the right, a 'Summary' panel shows the service details: 'Watson Assistant' (Free), Region: London, Plan: Lite, Service name: Watson Assistant- Demo, and Resource group: Default. Below this is a large red-bordered 'Create' button. At the bottom of the summary panel are links for 'View terms', 'Add to estimate', and 'FEEDBACK'.

**Note** - Creation of service may take up to a minute or two.

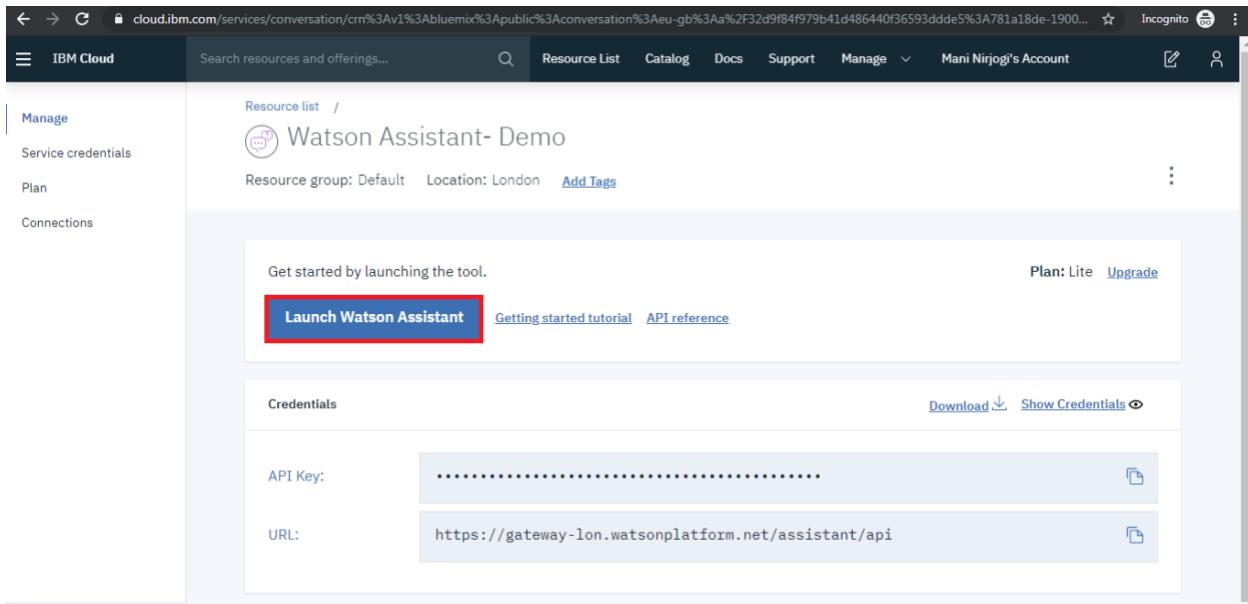
You will land on the **Manage** page of the service dashboard.



The screenshot shows the Watson Assistant service dashboard. The left sidebar has 'Manage' selected. The main area shows the service name 'Watson Assistant- Demo' with a status of 'Up and running'. It includes fields for 'Resource group: Default', 'Location: London', and a 'Add Tags' link. Below this is a 'Get started by launching the tool.' section with a 'Launch Watson Assistant' button and links for 'Getting started tutorial' and 'API reference'. Further down are sections for 'Credentials' (with download and show links) and 'API Key' and 'URL' fields (each with a copy icon).

Copy the **API key** and **URL**. Navigate to **app.js** file which you downloaded in your local drive and paste the API key in the place of <**Your-Watson-Assistant-APIKey**> and URL in place of <**Your-Watson-Assistant-URL**>

Now, go back to Watson Assistant dashboard and click on **Launch Watson Assistant**. It will redirect you to the Assistants page.

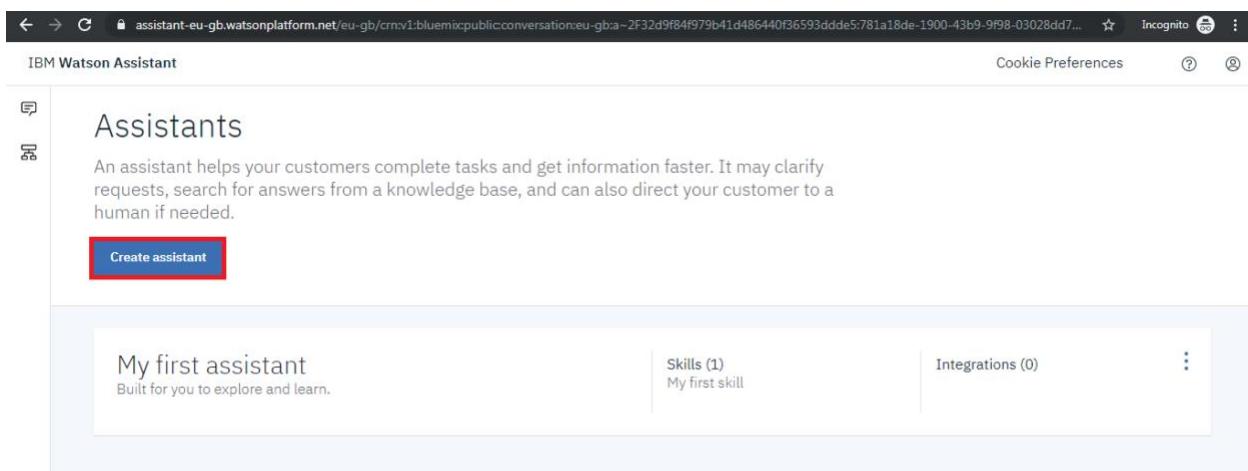


The screenshot shows the IBM Cloud interface for a Watson Assistant resource named "Watson Assistant- Demo". The "Launch Watson Assistant" button is highlighted with a red box. Below it, the "Credentials" section displays the API Key and URL.

Credentials
API Key: ..... (redacted)
URL: <a href="https://gateway-lon.watsonplatform.net/assistant/api">https://gateway-lon.watsonplatform.net/assistant/api</a>

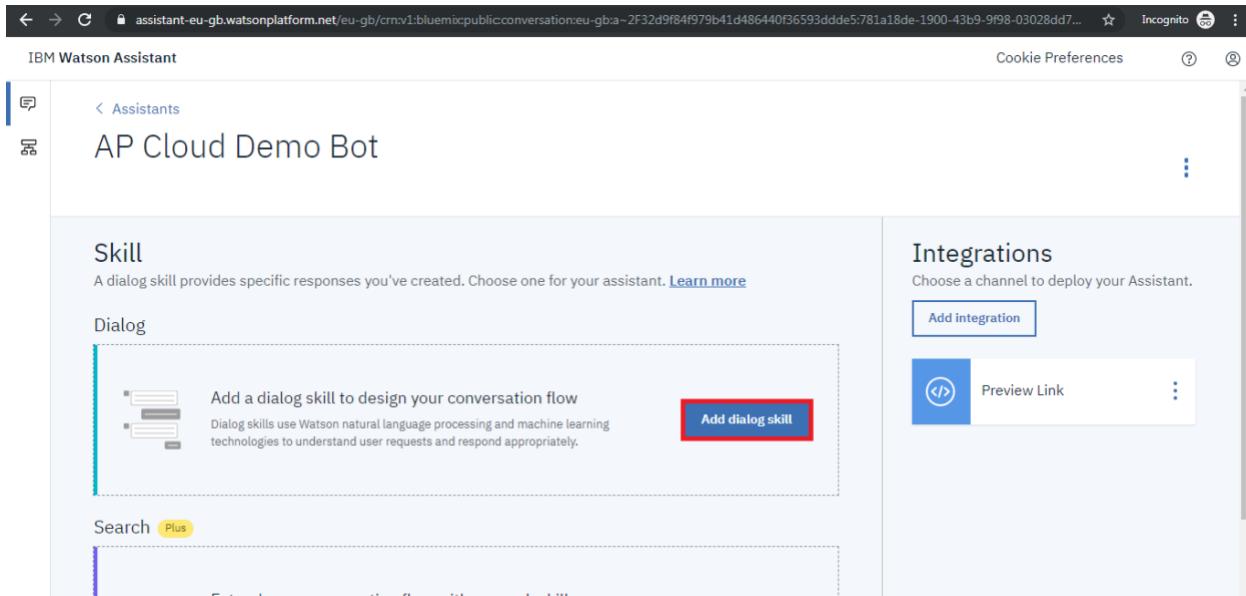
## Step #2 | Create Skill in Watson Assistant Service

Your first step in the Watson Assistant tool is to create an Assistant. Once you redirect to the Assistants page, click on the **Create assistant** button to build your chatbot as shown below.



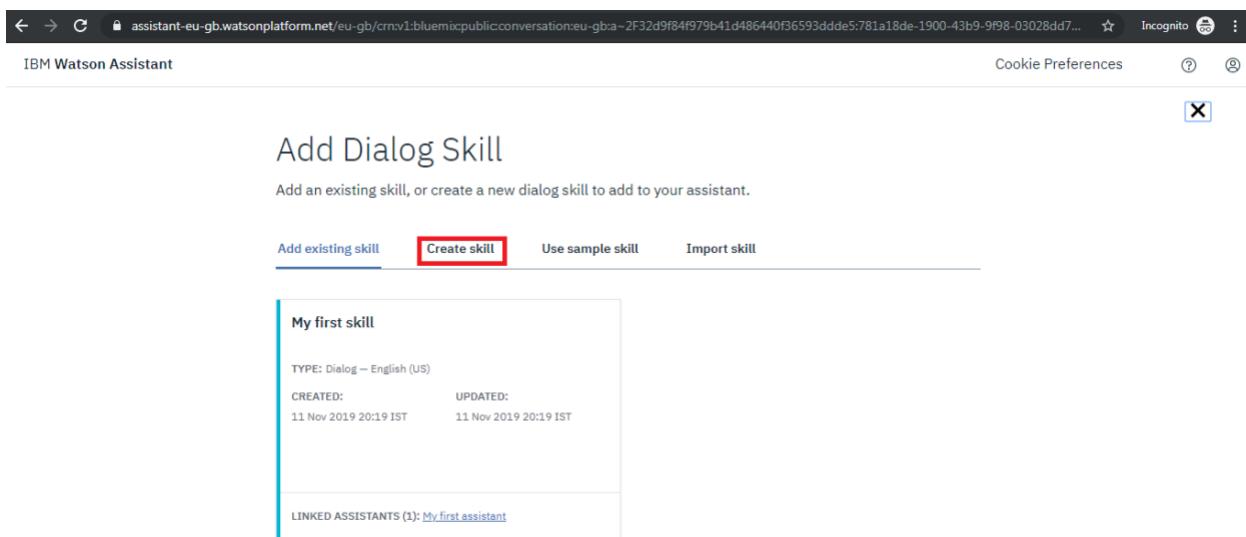
The screenshot shows the Watson Assistant "Assistants" page. The "Create assistant" button is highlighted with a red box. Below it, a card displays "My first assistant" with the status "Built for you to explore and learn." To its right, there are sections for "Skills (1)" and "Integrations (0)".

Provide the Assistant name and click on **Create assistant**, it will redirect you to the skills page. To add combine the skill to Assistant click on **Add dialog skill** as shown below.



The screenshot shows the 'Skills' section of the IBM Watson Assistant interface. On the left, there's a sidebar with icons for Assistant, Conversation, and Data. The main area has a title 'AP Cloud Demo Bot'. Below it, there's a 'Skill' section with a sub-section 'Dialog'. Inside the 'Dialog' section, there's a description: 'Add a dialog skill to design your conversation flow. Dialog skills use Watson natural language processing and machine learning technologies to understand user requests and respond appropriately.' To the right of this description is a red-bordered 'Add dialog skill' button. Below the 'Dialog' section is a 'Search' section with a 'Plus' icon. To the right of the 'Skill' section is an 'Integrations' section with a 'Preview Link' button.

Click on **Create skill** and then you will need to provide a Name and Description to your skill. After entering all details, click on Create button.



The screenshot shows the 'Add Dialog Skill' creation form. At the top, there's a header 'Add Dialog Skill' with a close button. Below it, a sub-header says 'Add an existing skill, or create a new dialog skill to add to your assistant.' There are four tabs at the top: 'Add existing skill' (disabled), 'Create skill' (selected and highlighted with a red border), 'Use sample skill', and 'Import skill'. The main area contains a card for a skill named 'My first skill'. The card shows the skill type as 'Dialog — English (US)', created and updated on '11 Nov 2019 20:19 IST'. At the bottom of the card, it says 'LINKED ASSISTANTS (1): My first assistant'.

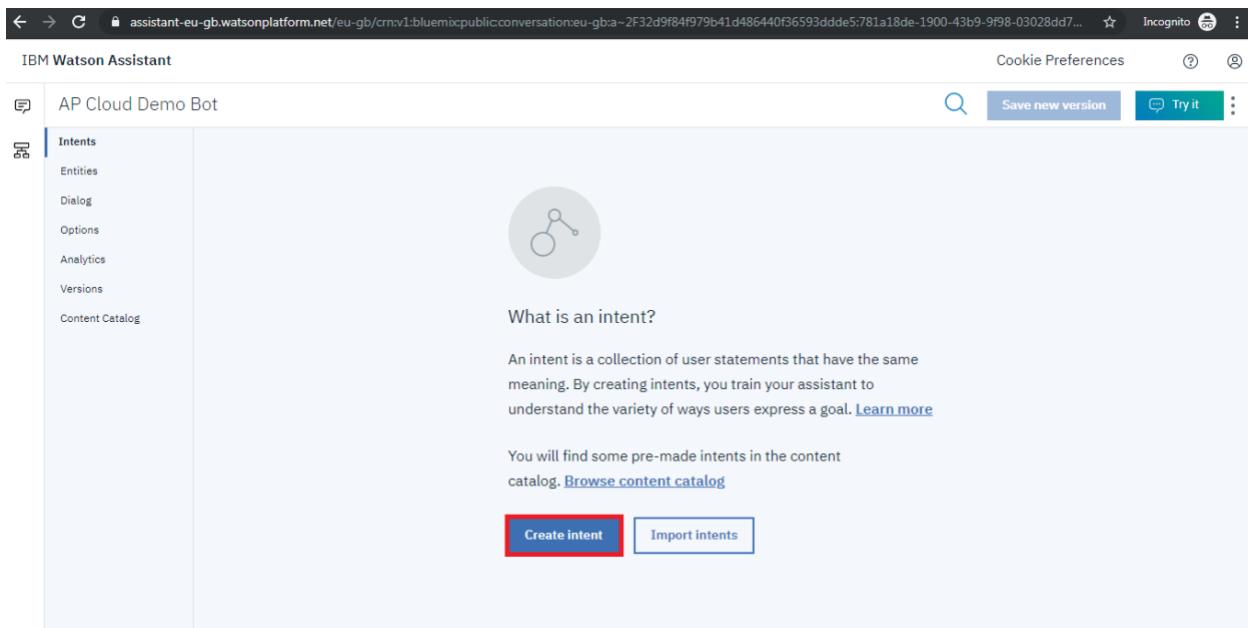
If you want to import the existing skill, click on the **Import Skill** tab. Choose any of the existing Watson Assistant Skill which is a **JSON** file to upload and click on **Import**.

**Note** - You can find the sample JSON file with the name **dialog\_model.json** in the example folder which you downloaded from GitHub repository.

## Step #3 | Training the Skill using Intents, Entities and Dialogs

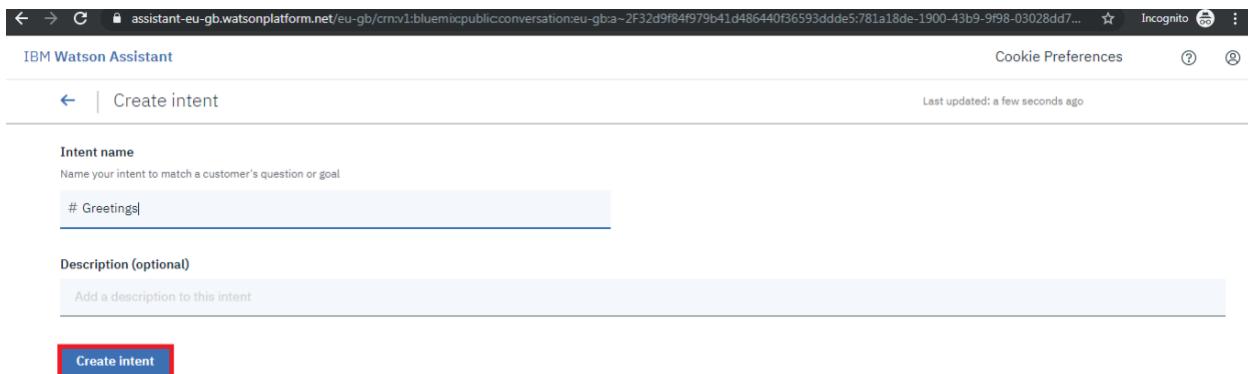
### a) Intents

Now, you will land on the Intents page of your new skill. Intents are purpose or goals expressed in a user's input. Click on **Create intent** to define new Intents for your chatbot to get trained.



The screenshot shows the IBM Watson Assistant interface. The top navigation bar includes links for Assistant, Conversation, and Assistant API. The main header says "IBM Watson Assistant" and "AP Cloud Demo Bot". On the left, a sidebar menu lists "Intents" (which is selected and highlighted in blue), "Entities", "Dialog", "Options", "Analytics", "Versions", and "Content Catalog". The main content area features a large circular icon with a network graph. Below it, the text "What is an intent?" is followed by a detailed explanation: "An intent is a collection of user statements that have the same meaning. By creating intents, you train your assistant to understand the variety of ways users express a goal." A link to "Learn more" is provided. Further down, it says "You will find some pre-made intents in the content catalog." with a link to "Browse content catalog". At the bottom of the content area are two buttons: a red-bordered "Create intent" button and a blue "Import intents" button.

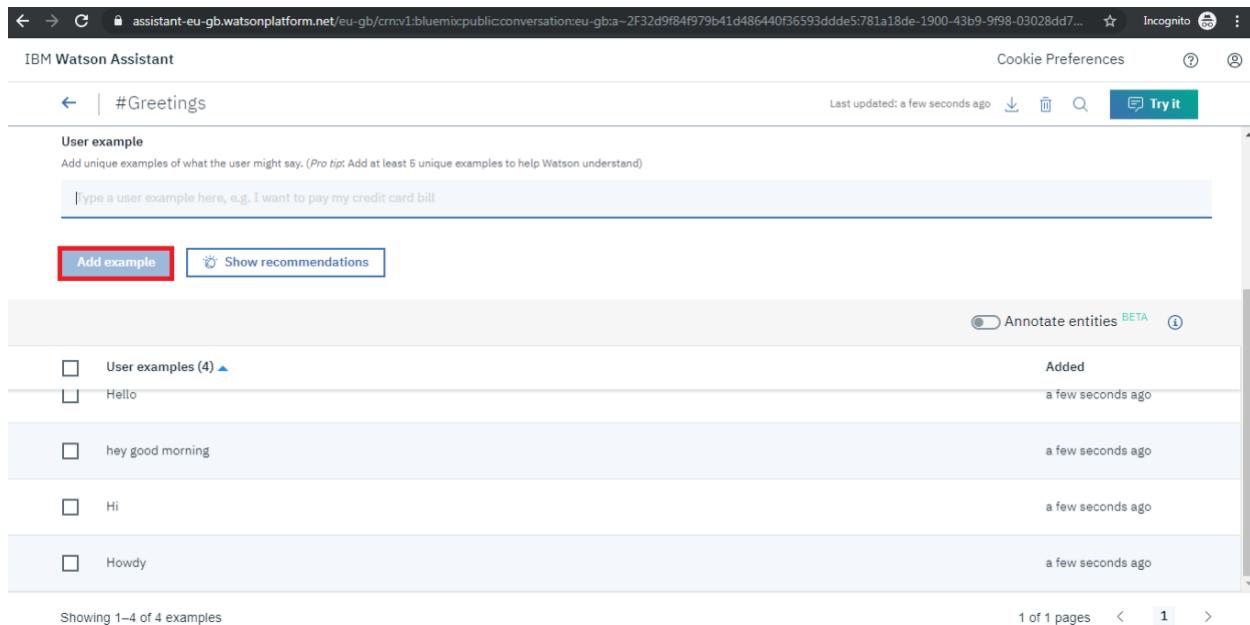
After clicking on **Create intent** button, you need to provide the Intent name, Description, and then click on **Create intent**.



The screenshot shows the "Create intent" form. The top navigation bar and header are identical to the previous screenshot. The main form has a title "Create intent". It contains two main sections: "Intent name" and "Description (optional)". The "Intent name" section has a placeholder "Name your intent to match a customer's question or goal" and a text input field containing "# Greetings". The "Description (optional)" section has a placeholder "Add a description to this intent" and a text input field. At the bottom of the form is a red-bordered "Create intent" button.

Now, you need to add user examples for that intent as shown below,

**Note** - After each example, you can click on **Add example** button or hit the enter button.



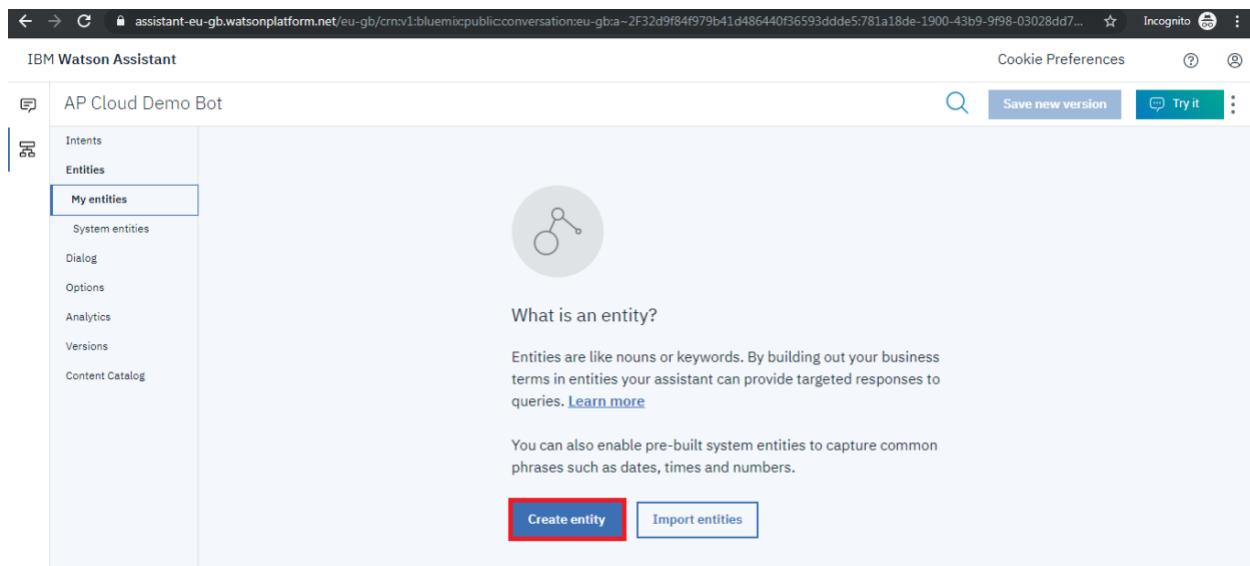
The screenshot shows the 'User example' section of the IBM Watson Assistant interface. It includes a text input field with placeholder text 'Type a user example here, e.g. I want to pay my credit card bill', a red-bordered 'Add example' button, and a 'Show recommendations' button. Below this is a list of four examples:

User example	Added
Hello	a few seconds ago
hey good morning	a few seconds ago
Hi	a few seconds ago
Howdy	a few seconds ago

At the bottom, it says 'Showing 1–4 of 4 examples' and '1 of 1 pages'.

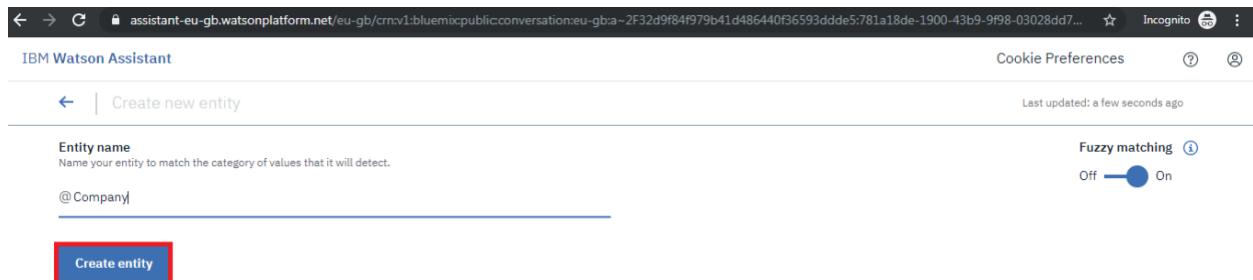
## b) Entities

An entity represents information in the user input that is relevant to the user's purpose. To create an entity, go to **Entities** tab, under **My entities** tab click on **Create entity**.



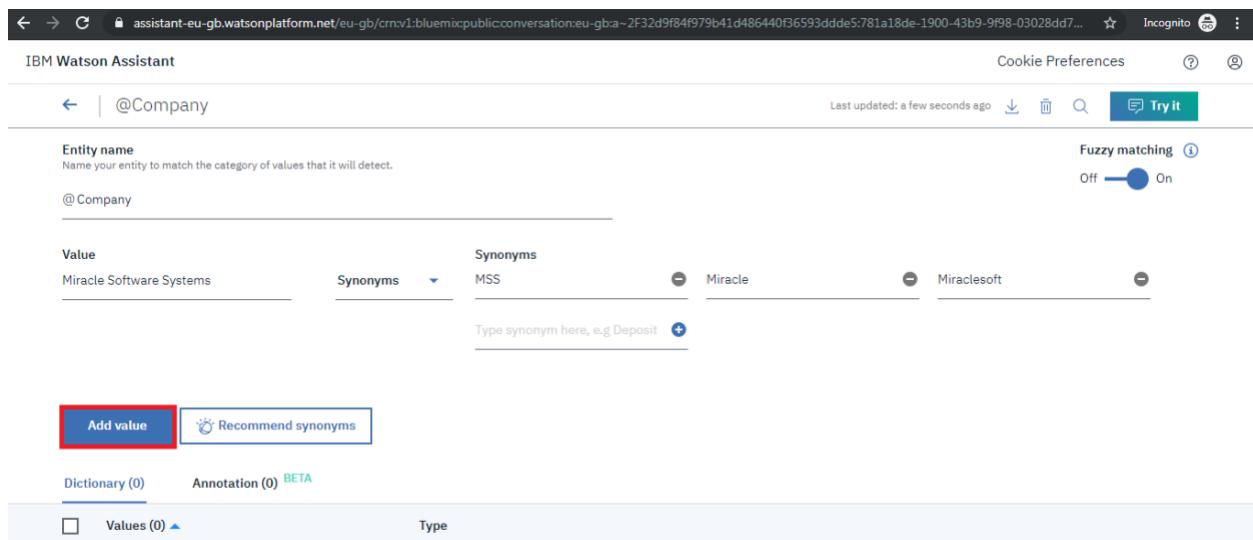
The screenshot shows the 'My entities' section of the IBM Watson Assistant interface. On the left, there is a sidebar with tabs for 'Intents', 'Entities' (which is selected), and 'My entities'. The main area features a circular icon with a network graph. Below it, the text 'What is an entity?' is followed by a description: 'Entities are like nouns or keywords. By building out your business terms in entities your assistant can provide targeted responses to queries.' A link 'Learn more' is provided. Further down, it says 'You can also enable pre-built system entities to capture common phrases such as dates, times and numbers.' At the bottom are two buttons: a red-bordered 'Create entity' button and a 'Import entities' button.

In the Entity name field, provide a name for the entity and then click on **Create entity**.



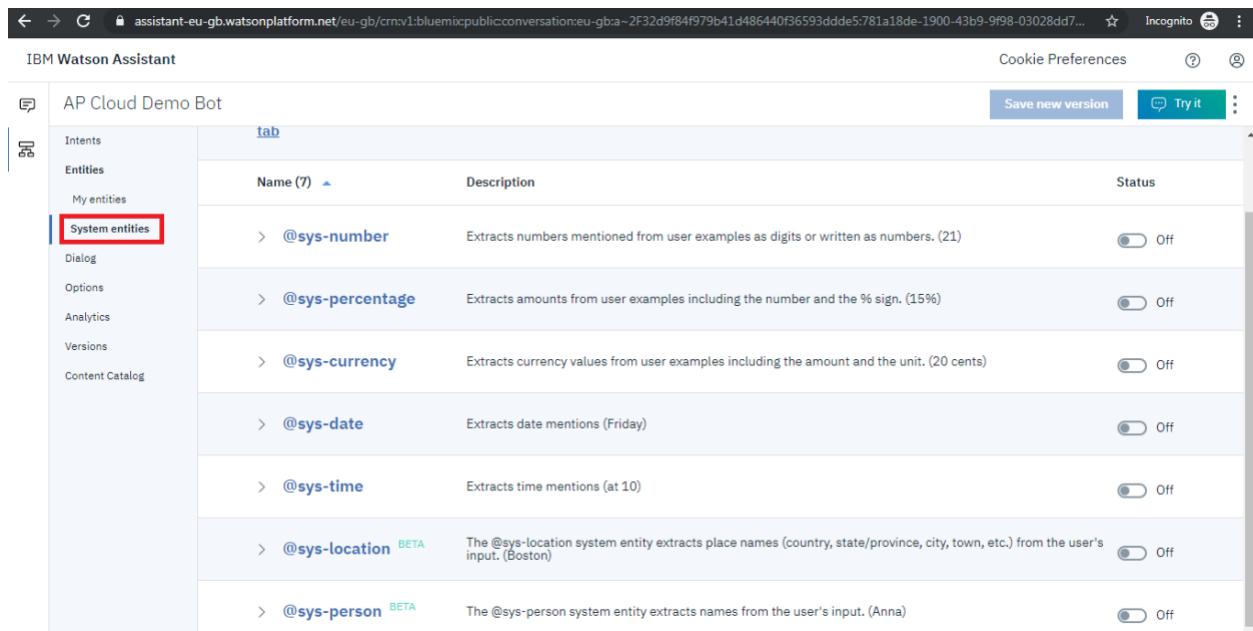
The screenshot shows the 'Create new entity' page. In the 'Entity name' field, the value '@ Company' is entered. A red box highlights the 'Create entity' button at the bottom left.

After entity creation, you need to enter a Value name for the entity, click on **Add value** and also you can add any synonyms which are based on contextual similarity for specific values as shown below.



The screenshot shows the entity details page for '@ Company'. Under the 'Value' section, 'Miracle Software Systems' is listed. Under the 'Synonyms' section, 'MSS' is listed. A red box highlights the 'Add value' button at the bottom left.

You can also use predefined entities called **System Entities**. Click on **System entities** tab to select from a list of common entities that are provided by IBM.

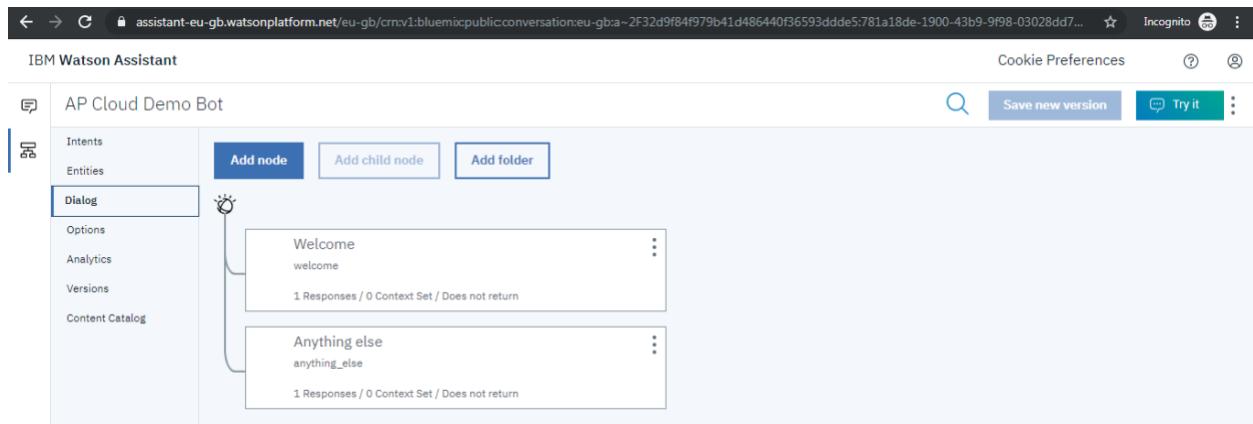


The screenshot shows the IBM Watson Assistant interface for the AP Cloud Demo Bot. The left sidebar has a 'System entities' option highlighted with a red box. The main area displays a table of system entities:

Name	Description	Status
@sys-number	Extracts numbers mentioned from user examples as digits or written as numbers. (21)	Off
@sys-percentage	Extracts amounts from user examples including the number and the % sign. (15%)	Off
@sys-currency	Extracts currency values from user examples including the amount and the unit. (20 cents)	Off
@sys-date	Extracts date mentions (Friday)	Off
@sys-time	Extracts time mentions (at 10)	Off
@sys-location <small>BETA</small>	The @sys-location system entity extracts place names (country, state/province, city, town, etc.) from the user's input. (Boston)	Off
@sys-person <small>BETA</small>	The @sys-person system entity extracts names from the user's input. (Anna)	Off

### c) Dialog

Dialog uses the intents and entities together that are identified in the user's input and respond back to the user based on the matched intent. In Dialog, you will get two nodes initially.



The screenshot shows the IBM Watson Assistant interface for the AP Cloud Demo Bot. The left sidebar has a 'Dialog' option highlighted with a red box. The main area shows two nodes in a tree structure:

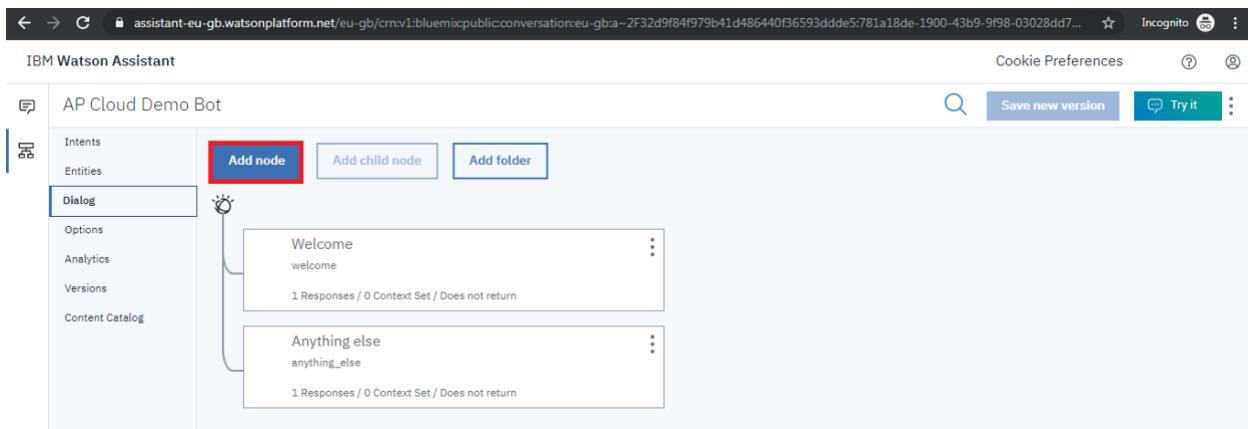
- Welcome**: welcome  
1 Responses / 0 Context Set / Does not return
- Anything else**: anything\_else  
1 Responses / 0 Context Set / Does not return

At the top of the main area, there are buttons for 'Add node', 'Add child node', and 'Add folder'.

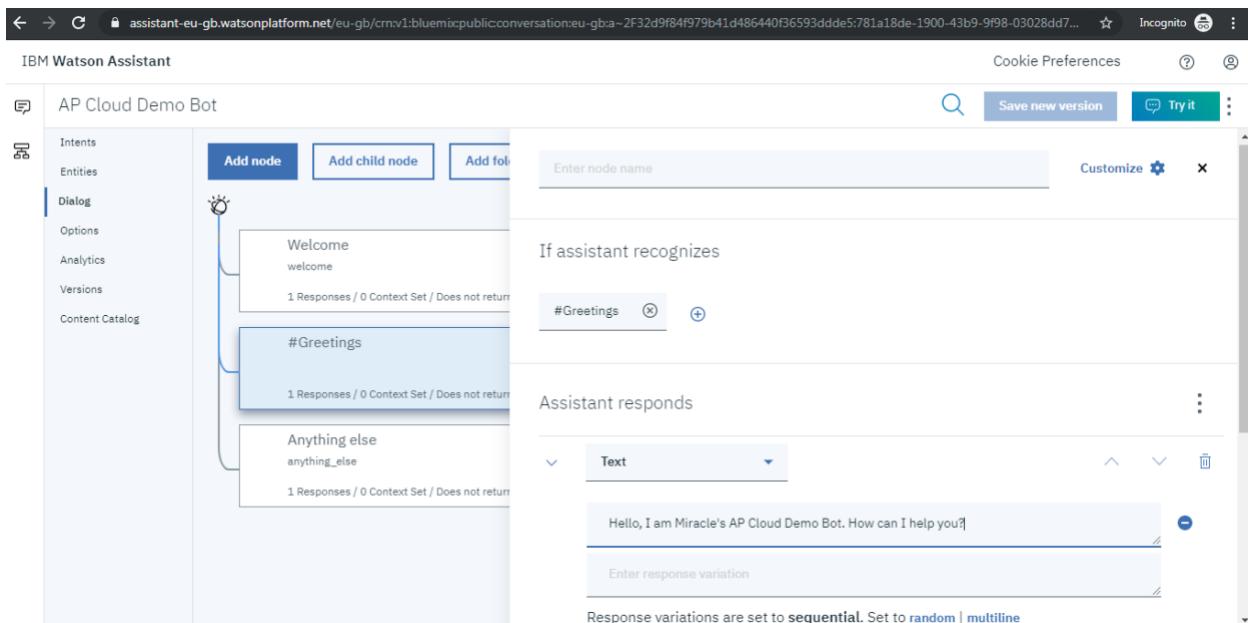
**Welcome** - This is the first node that contains the initial greetings when users engage with the service.

**Anything else** - This is the final node that contains phrases which are used to respond back to the user when input is not recognized.

If you want to add more nodes to your dialog tree, click on **Add node** button on the top as shown in the below image.



Once you add a node, you will see one new node added into your dialog tree as shown below.



Add the text or multimedia elements that you want the service to display to the user as a response. For that, click on the node and provide response to user's request.

If you want to test your skill in the Watson Assistant tool itself, click on **Try it**.

Now, you are all set to test your chatbot in Watson Assistant tool.

To access the skill through Node JS, you need to provide the Workspace ID in **app.js file**. For that, go back to Assistants and click on the created assistant.

Then it will be redirected to the skill which was added to that particular assistant. Click on three dots (⋮) at the right corner, and the below options are displayed.

IBM Watson Assistant

Cookie Preferences [?](#) [@](#)

**Skill**

A dialog skill provides specific responses you've created. Choose one for your assistant. [Learn more](#)

**Dialog**

**AP Cloud Demo Bot**

LANGUAGE: English (US) TRAINED DATA: 1 Intents | 1 Entities | 3 Dialog Nodes VERSION: Development CREATED: 11 Nov 2019 20:25 IST

UPDATED: 11 Nov 2019 20:38 IST

LINKED ASSISTANTS (1): AP Cloud Demo Bot

**Integrations**

Choose a channel to deploy your Assistant.

[Add integration](#)

 Preview Link [:](#)

Choose **View API Details** option and click on it. You will get all the API details.

From those API details, you will get the Workspace ID. Copy that ID and navigate to **.env** file, paste the ID in the place of <**Your-Watson-Assistant-WorkspaceID**>

IBM Watson Assistant

[?](#) [@](#)

X

**Skill Details**

**Skill Name:** APCloud Demo Bot

**Skill ID:** c64106ba-b772-4430-9c2f-092652ce95f1

**Workspace ID:** [REDACTED]

**Legacy v1 Workspace URL:** <https://gateway-syd.watsonplatform.net/assistant/api/v1/workspaces/cf5f9527-527c-43cf-a5df-4b7a4125fe52/message>

---

**Service Credentials**

**Service Credentials Name:** auto-generated-apikey-6e7a6d68-6098-42b0-8f7a-a6129c1a534e

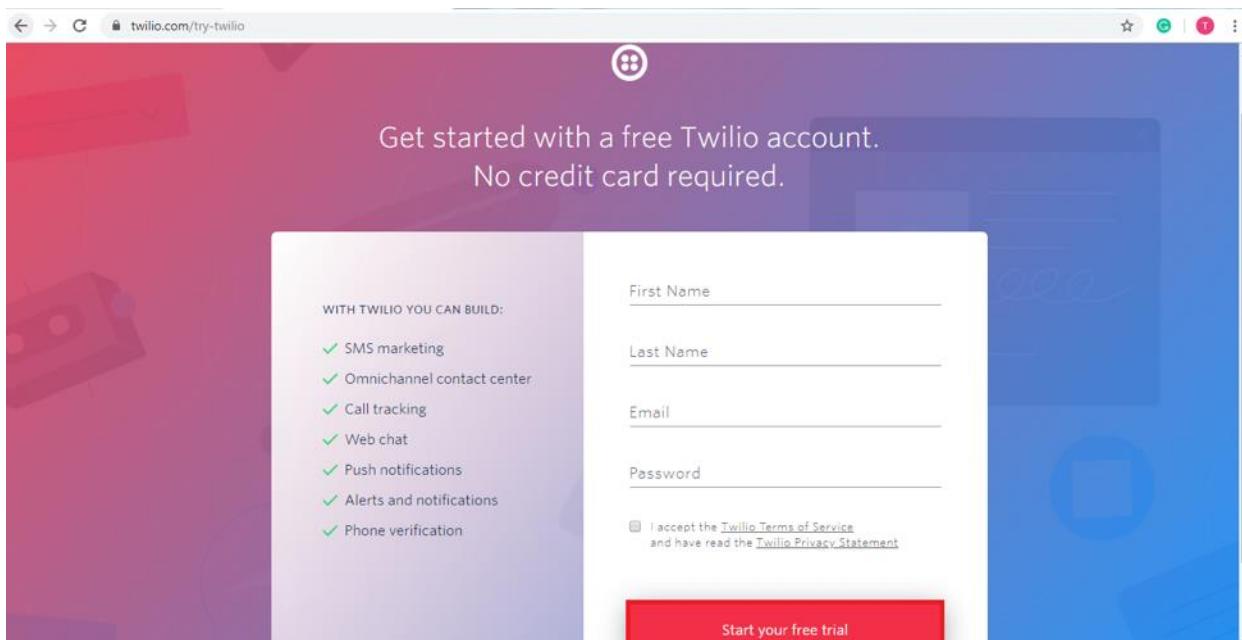
**Username:** apikey

**Password:** ZNrByXG3xHjzJb8nXFbSQw5dAlw1yvyzZ0wj-iYqmaN

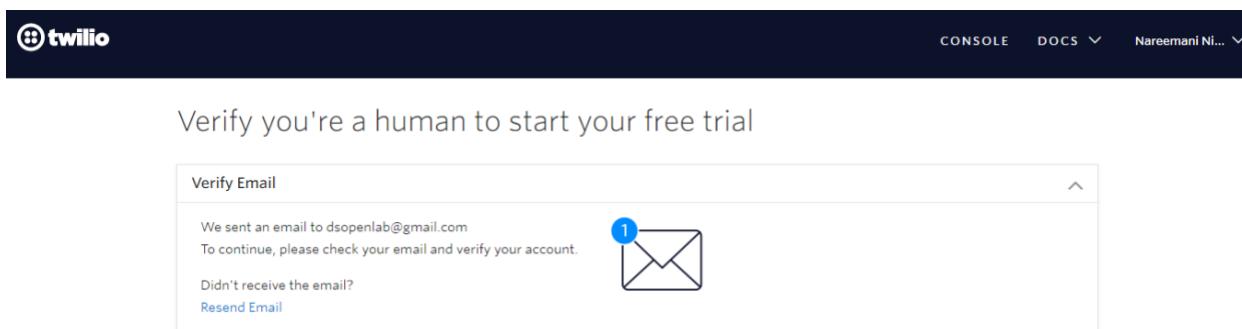
## Step #4 | Creating Twilio Account

Initially, you need to create a Twilio account to get the WhatsApp number for the bot. Here, is the link to create free Twilio account <http://www.twilio.com/try-twilio>.

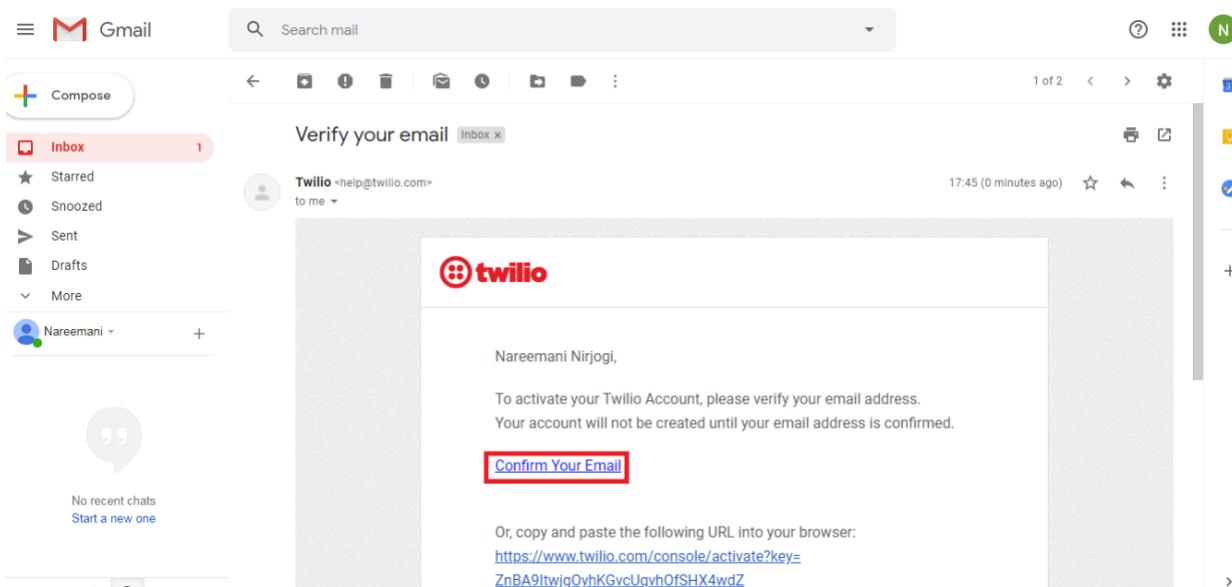
Enter the required details and click on **Start your free trial**.



Once we click on the **Start your free trial** it sends the confirmation email to the registered mail.



Once open your gmail and confirm it by clicking the **Confirm Your Email** as shown in the below image.



After verifying the account, it will redirect you to the below page, here provide your mobile number and click on **Verify**.

Verify you're a human to start your free trial

Verify Email

Verify Phone Number

NUMBER  +91 Phone Number

Why verify a phone number?

We will contact you at the number above with a verification code.

The phone number you provide will be used for authentication when you login to Twilio Console. A Twilio onboarding specialist may also use this number to reach out with free onboarding support. If you do not want to be contacted at this phone number, please check this box.

You will get the verification code to the number provided, enter the code and click on **Submit**.



Verify you're a human to start your free trial

This is a continuation of the Twilio verification page from the previous screenshot. It shows the 'Verify Phone Number' section in more detail. It includes a note about entering the verification code sent to the phone number 6303815600, a 'Verification Code' input field, and a 'Submit' button. Below this, there are links for 'Want to verify with a Call instead of SMS?' and 'Didn't receive a code? Resend Code (51)'.

It will display the Project Dashboard with the Project Name and Account SID. By using Account SID we can connect the Twilio project to Watson Assistant using Node JS. Copy the ACCOUNT SID and AUTH TOKEN for later use.

In this dashboard click on 3 horizontal dots icon (Products and Services option) as shown below.

The screenshot shows the Twilio Project Dashboard. The left sidebar has icons for Home, Billing, Usage, Settings, and Upgrade. The '... (3 dots)' icon is highlighted with a red box. The main area displays the 'My first Twilio project Dashboard'. It includes sections for 'Project Info' (Trial Balance: \$15.50, Get a Trial Number button), 'Referral Program' (Refer your network to Twilio — give \$10, get \$10, Referral Dashboard button), 'ACCOUNT SID' (AC32173693cb02f23d5ea1cf879d01e7cb), 'AUTH TOKEN' (Show button), 'PROJECT NAME' (My first Twilio project), and 'PROGRAMMING LANGUAGE' (Node.js). The top navigation bar includes 'Upgrades Project', 'Go to...', and other account-related links.

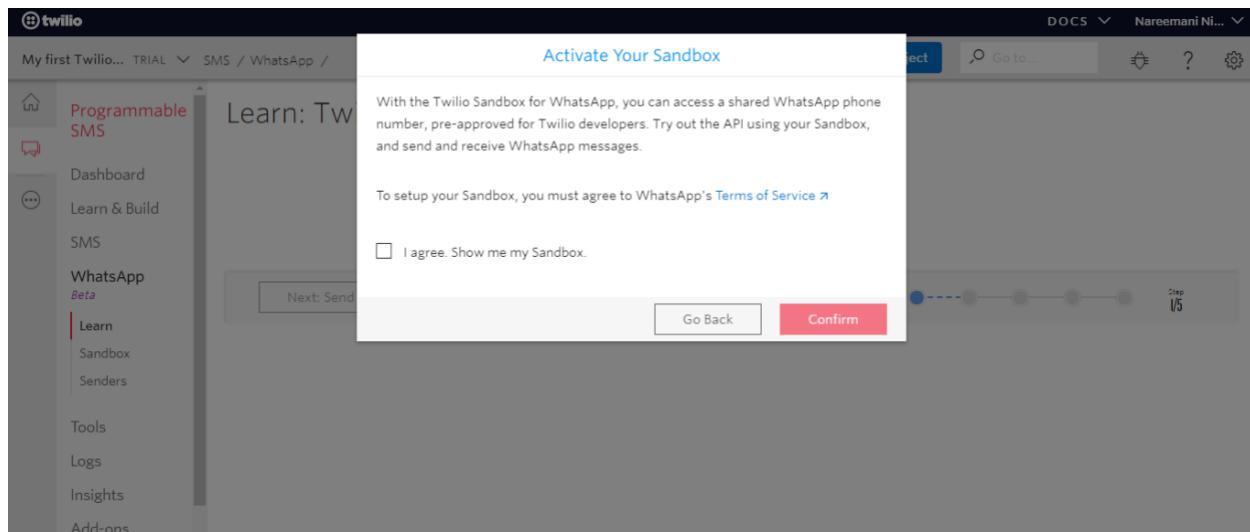
In Products and Services select **Programmable SMS** option.

The screenshot shows the Twilio project Dashboard. On the left sidebar under 'COMMUNICATIONS CLOUD', the 'Programmable SMS' option is highlighted with a red box. The main dashboard area displays a 'Twilio project Dashboard' with sections for 'Get a Trial Number', 'PROGRAM', and 'ID'. A table lists a single entry: 'PROJECT NAME' is 'My first Twilio project' and 'PROGRAMMING LANGUAGE' is 'Node.js'. The top right corner shows 'DOCS', 'Nareemani Ni...', and other navigation icons.

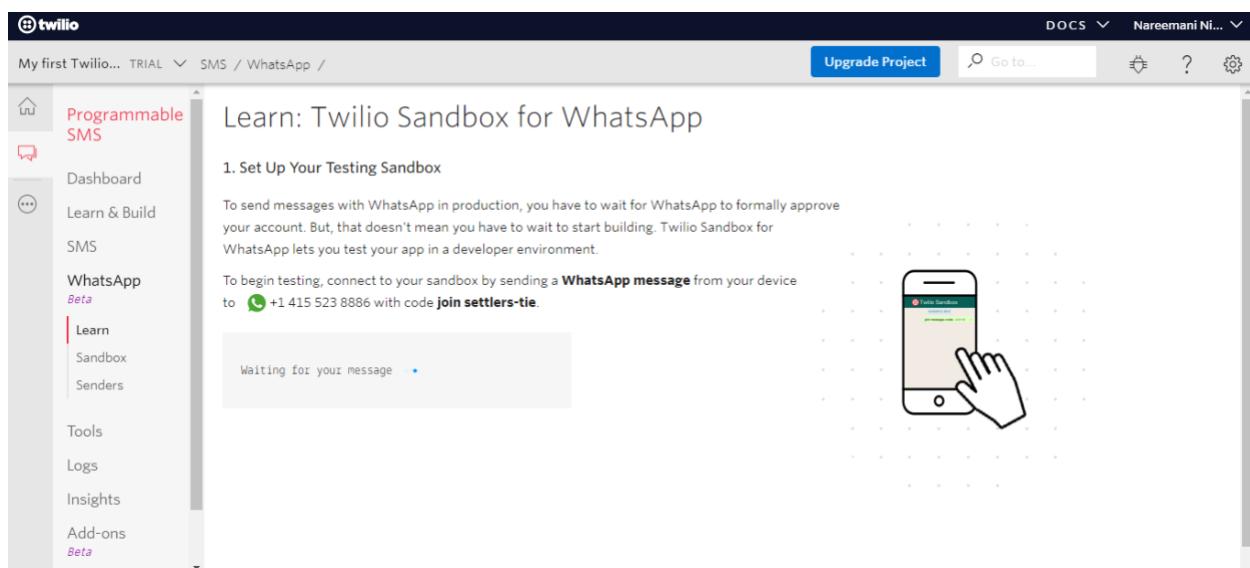
Under Programmable SMS, select **WhatsApp** which you can find as Beta.

The screenshot shows the Programmable SMS Dashboard. On the left sidebar under 'SMS', the 'WhatsApp' option is highlighted with a red box and labeled 'Beta'. The main dashboard area displays a 'Programmable SMS Dashboard' with a 'Building with WhatsApp? Get started here' banner. Below it, there are links for 'Get Started', 'Read the Tutorial Docs', and 'Features & Pricing'. A note says 'Heads up! You have a Trial Account. Here's what that means:' followed by a list of four items. The top right corner shows 'DOCS', 'Nareemani Ni...', and other navigation icons.

After selecting WhatsApp, it will display **Activate your Sandbox** as shown below. Select the dialog box **I agree. Show me my Sandbox** and click on **Confirm** button.



After activating the Sandbox it will provide a WhatsApp contact number to the bot for testing and gives the code for connecting.



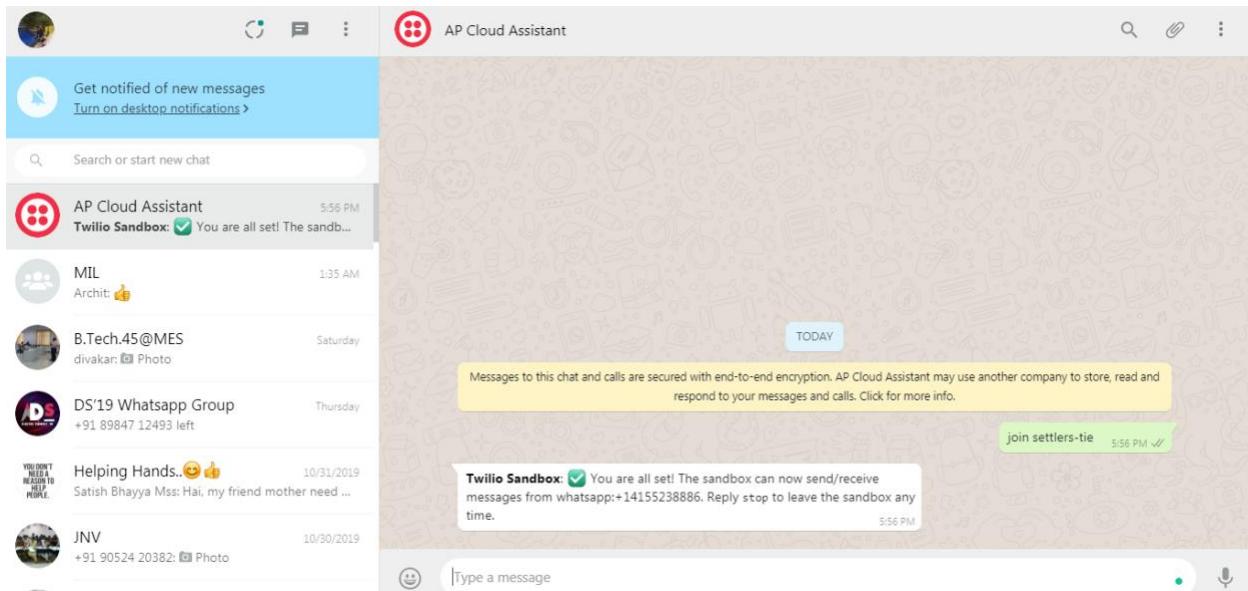
After adding the number that you received in the above step, save it to your contact list. Now, open WhatsApp in your mobile. If you don't have the WhatsApp application, download it.

Open your WhatsApp, search for your bot number or with the name you saved.

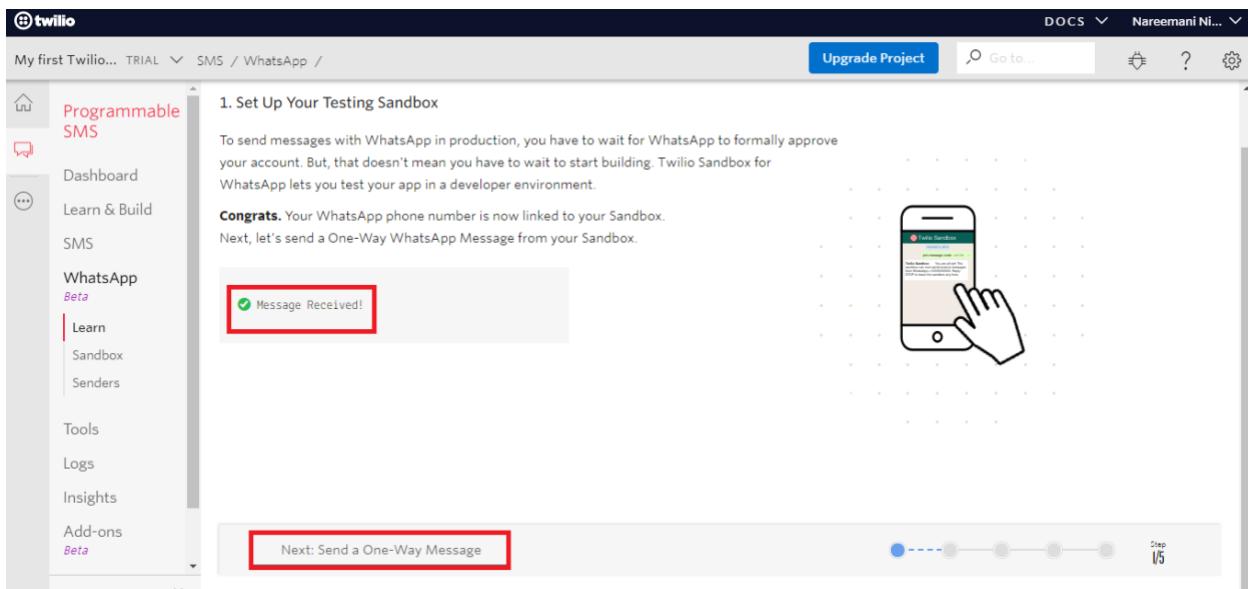
You need to send the code to the bot number to link your number with Sandbox.

**Note** - You can also open your WhatsApp in browser using,

<https://web.whatsapp.com>



After giving the code, your WhatsApp number will be linked to the Sandbox and will receive the message from Twilio. Then in your Twilio account it displays like **Message Received** as below and then click on **Next: Send a One-Way Message**.



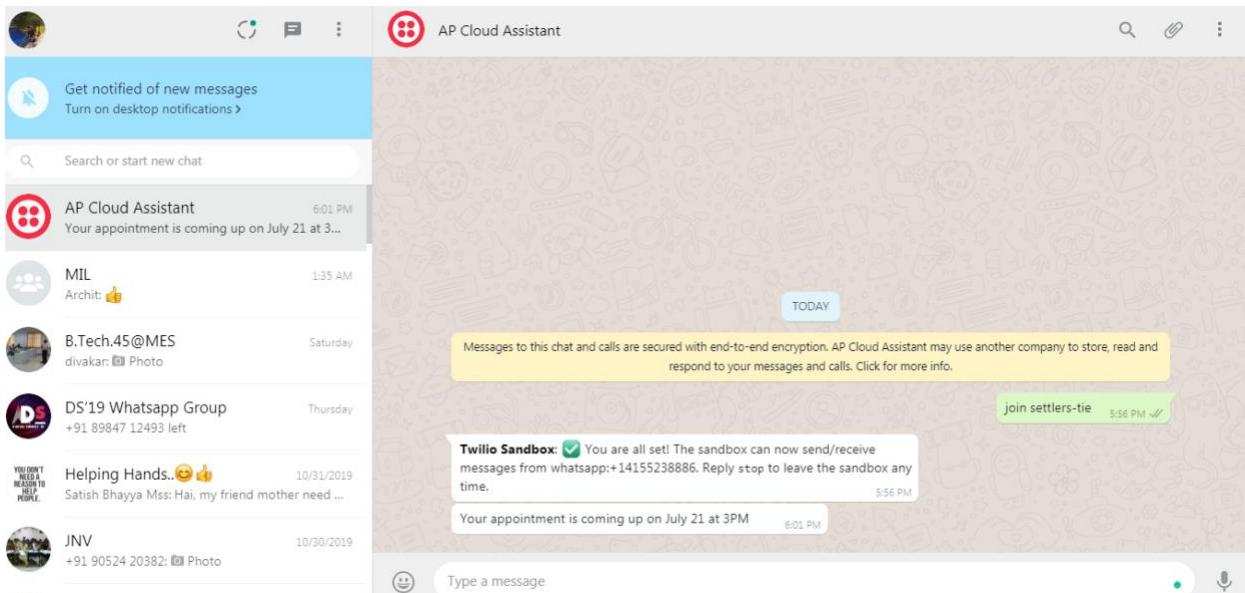
This One-Way Message is used for sending Outbound messages and Notifications. So, once you are done with One-Way Message click on any one of the options as shown below.

The screenshot shows the Twilio interface for WhatsApp. On the left sidebar, under the WhatsApp section, 'Appointment Reminders' is highlighted with a red box. The main content area is titled 'Learn: Twilio Sandbox for WhatsApp' and '2. Send a One-Way WhatsApp Message'. It explains that messaging with WhatsApp requires a pre-approved template. Below this, there are three buttons: 'Appointment Reminders' (selected), 'Order Notifications', and 'Verification Codes'. A template message is shown: 'Template: Your appointment is coming up on [[1]] at [[2]]'. The 'Parameters' section shows 'TO: whatsapp:+917093534594' and 'FROM: whatsapp:+14155238886'. To the right, there's a 'Request' section with a 'curl' dropdown containing a command to send the message using the specified parameters.

Then click on **Make Request** option so that user will get the appointment message.

This screenshot shows the Twilio interface after clicking the 'Make Request' button. The 'Request' section now displays the curl command used to send the message. The 'Response' section is a placeholder stating 'Response will appear here after you make request'. At the bottom, there are navigation links for 'Previous' and 'Next: Two-Way Messaging'.

We will get the message as shown below.



Then click on **Two-Way Messaging** in dashboard.

The screenshot shows the Twilio Programmable SMS dashboard. The left sidebar has links for 'Programmable SMS', 'Dashboard', 'Learn & Build', 'SMS', 'WhatsApp Beta', 'Learn', 'Sandbox', 'Senders', 'Tools', 'Logs', 'Insights', and 'Add-ons Beta'. The 'WhatsApp Beta' link is currently selected. The main area shows a 'BODY' input field containing 'Your appointment is coming up on July 21 at 3PM' and a red 'Make Request' button. To the right, there's a 'Response' section with the placeholder 'Response will appear here after you make request'. At the bottom, there are navigation buttons: '< Previous' (disabled), 'Next: Two-Way Messaging' (highlighted with a red box), and 'Step 2/5'.

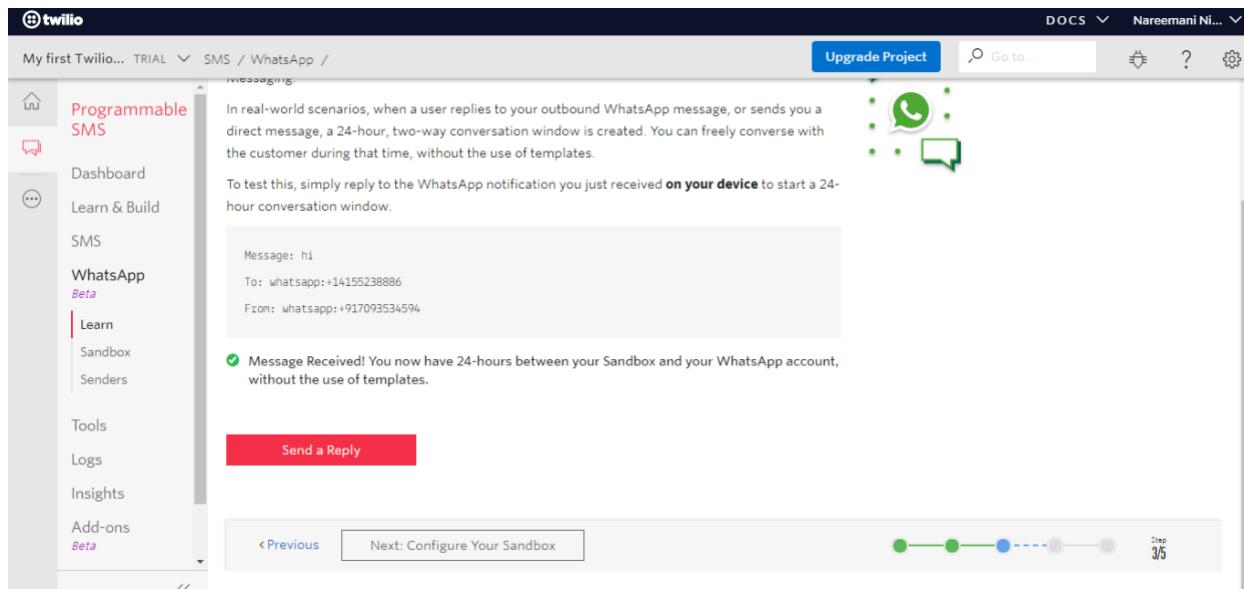
By enabling the Two-Way Messaging it creates two-way 24 hour conversation window.

The screenshot shows the Twilio web interface under the 'Programmable SMS' section. On the left, a sidebar lists options like Dashboard, Learn & Build, SMS, WhatsApp (Beta), Tools, Logs, Insights, and Add-ons. The main area is titled 'Messaging' and contains instructions about starting a 24-hour conversation window by replying to a WhatsApp message. It includes fields for 'Message:', 'To:', and 'From:', and a red 'Send a Reply' button. Below this is a progress bar labeled 'Step 35'.

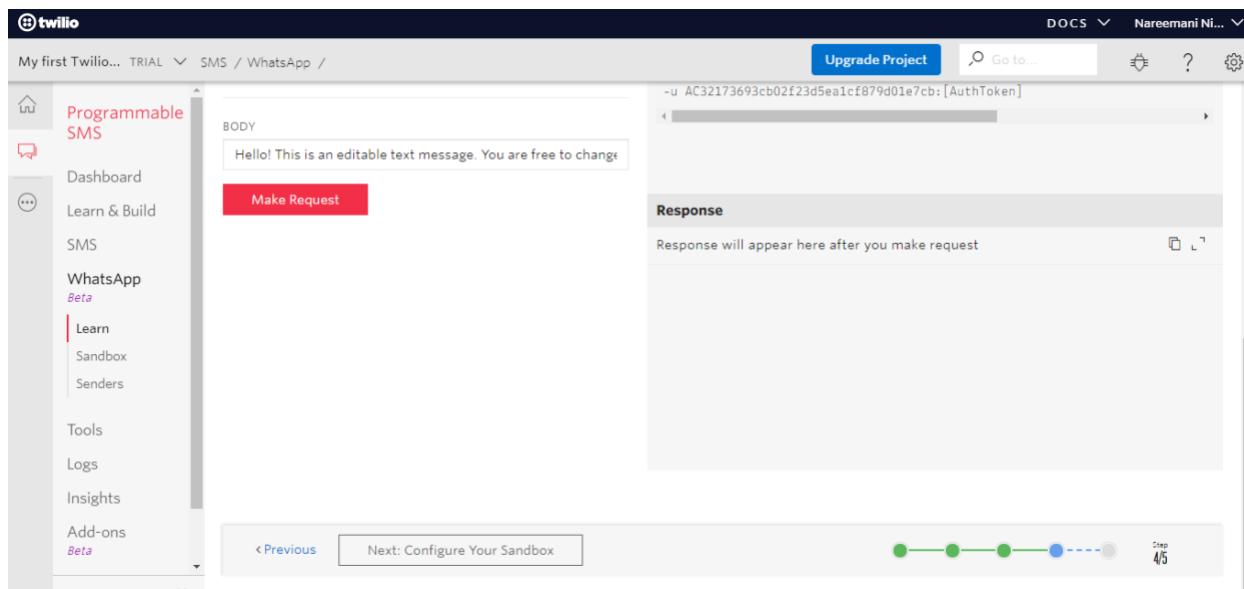
To test this, simply reply to the WhatsApp notification you received on your device to start a 24-hour conversation window. It responds back as shown below.

The screenshot shows a WhatsApp desktop application window. On the left, a sidebar lists recent chats: AP Cloud Assistant, MIL, B.Tech.45@MES, DS'19 Whatsapp Group, Helping Hands., and JNV. The main chat window is with 'AP Cloud Assistant'. It shows a message from the bot inviting notifications, a search bar, and a list of messages. One message from 'Twilio Sandbox' says: 'You are all set! The sandbox can now send/receive messages from whatsapp:+14155238886. Reply stop to leave the sandbox any time.' Another message from the bot says: 'Your appointment is coming up on July 21 at 3PM'. A message from 'Helping Hands.' says: 'You said :hi. Configure your WhatsApp Sandbox's Inbound URL to change this message.' A message from 'JNV' says: 'hi'. At the bottom, there's a message input field with a smiley face icon and a microphone icon.

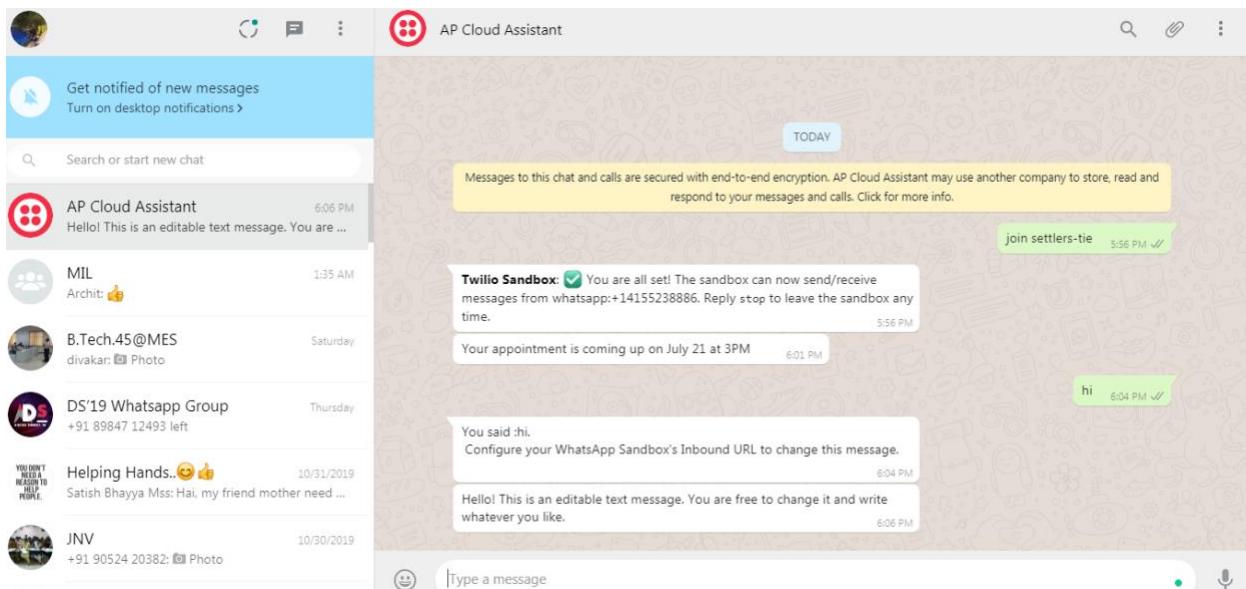
After receiving message in WhatsApp, the **To** number and **From** number will be updated automatically in the Two-Way messaging as shown below.



Then click on **Make Request** for checking the API's.



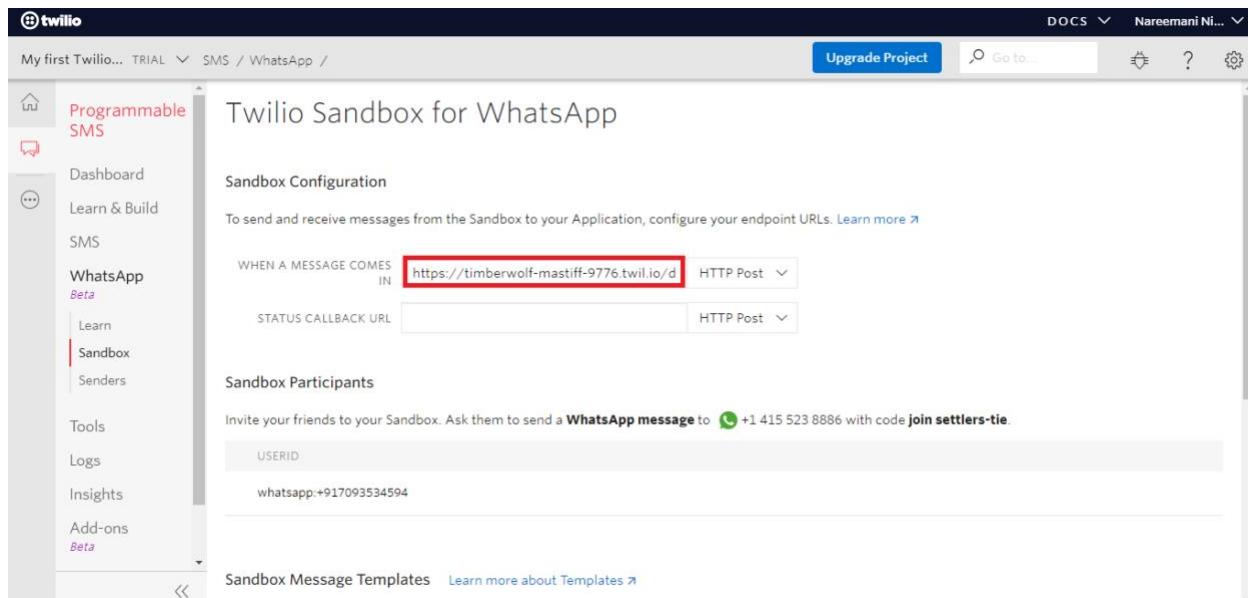
Now will get the configuration message to the WhatsApp as shown below.



After getting the configuration message, click on **Configure Your Sandbox**.

The screenshot shows the Twilio developer console under the 'Programmable SMS' section. On the left, there's a sidebar with options like Dashboard, Learn & Build, SMS, WhatsApp (Beta), Tools, Logs, Insights, and Add-ons. The main area shows a 'BODY' input field containing a test message and a 'Make Request' button. To the right, there's a 'Response' section showing a successful '201 - CREATED' response with a JSON representation of the message details. Below the response, there's a navigation bar with 'Next: Configure Your Sandbox' highlighted.

After selecting **Configure Your Sandbox**, it will provide a default endpoint for the WhatsApp service like from where the messages are coming in.



The screenshot shows the Twilio Sandbox for WhatsApp configuration page. On the left, there's a sidebar with options like Dashboard, Learn & Build, SMS, WhatsApp (Beta), and Tools. The main area is titled "Twilio Sandbox for WhatsApp" and has a section for "Sandbox Configuration". It says "To send and receive messages from the Sandbox to your Application, configure your endpoint URLs." Below this, there are two input fields: "WHEN A MESSAGE COMES IN" with the value "https://timberwolf-mastiff-9776.twilio.com/d" and "HTTP Post", and "STATUS CALLBACK URL" with "HTTP Post" dropdown. There's also a section for "Sandbox Participants" with a placeholder "USERID" and a value "whatsapp:+917093534594". At the bottom, there's a link "Sandbox Message Templates" and a note "Learn more about Templates".

Here it provides the default endpoint for messages, now you need to give your bot endpoint in place of default so, that it will be connected to your bot which is built in IBM Watson.

## Step #5 | Integration with WhatsApp using Node JS App

Go to **app.js** file, and place the Account SID and AUTH token which was copied previously from Twilio for authorizing the Twilio project. Now you connected Watson Conversation service with Twilio and finally you have to place the number of your Bot in **app.js** file, which was provided by Twilio in line **#49** in place of <WhatsApp-Number>. So, that it will receive the messages from the user and send the messages.

Navigate to the workspace folder where the code exists, and open command prompt. Run **node app.js**

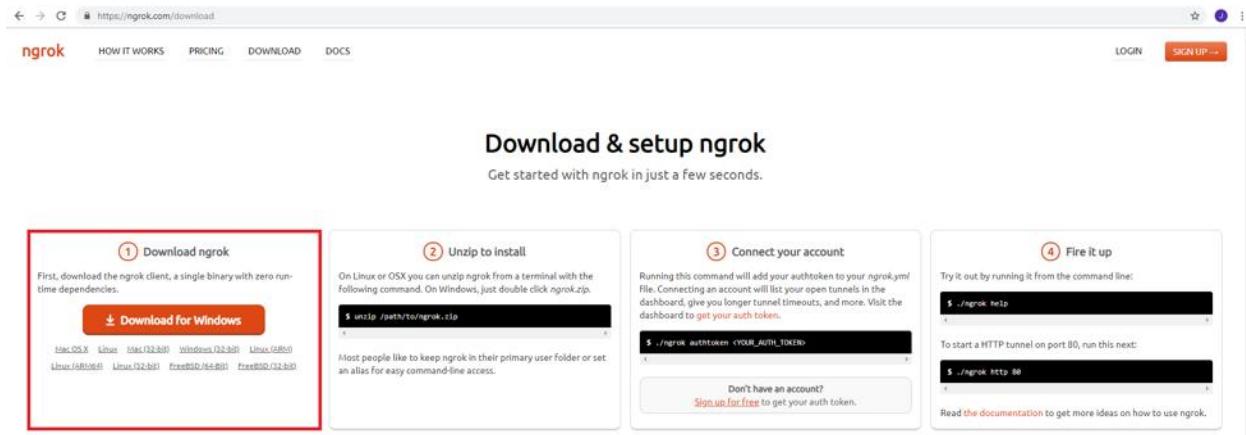


```
C:\Windows\System32\cmd.exe - node app.js
C:\Users\avennela\Desktop\APcloud lab\ node app.js
running on 8000
```

Application is **running on 8000** port. Remember this for later use.

Open the browser and download ngrok by using the below link, <https://ngrok.com/download> (ngrok helps your app to get secure URL to your localhost server).

You need to install the ngrok based on the system requirement as provided below.



**Download & setup ngrok**

Get started with ngrok in just a few seconds.

- 1 Download ngrok**  
First, download the ngrok client, a single binary with zero runtime dependencies.  
[Download for Windows](#)  
Mac OS X Linux Mac OS X 64-bit Windows Q3 64-bit Linux (32-bit)  
Linux (64-bit) Linux (32-bit) FreeBSD 64-bit FreeBSD 32-bit
- 2 Unzip to install**  
On Linux or OSX you can unzip ngrok from a terminal with the following command. On Windows, just double click ngrok.zip.  
`$ unzip /path/to/ngrok.zip`  
Most people like to keep ngrok in their primary user folder or set an alias for easy command-line access.
- 3 Connect your account**  
Running this command will add your auth token to your `ngrok.yml` file. Connecting an account will list your open tunnels in the dashboard, give you longer tunnel timeouts, and more. Visit the dashboard to [get your auth token](#).  
`$ ./ngrok auth token YOUR_AUTH_TOKEN`  
Don't have an account?  
[Sign up for free](#) to get your auth token.
- 4 Fire it up**  
Try it out by running it from the command line:  
`$ ./ngrok http 80`  
To start a HTTP tunnel on port 80, run this next:  
`$ ./ngrok http 80`  
Read the documentation to get more ideas on how to use ngrok.

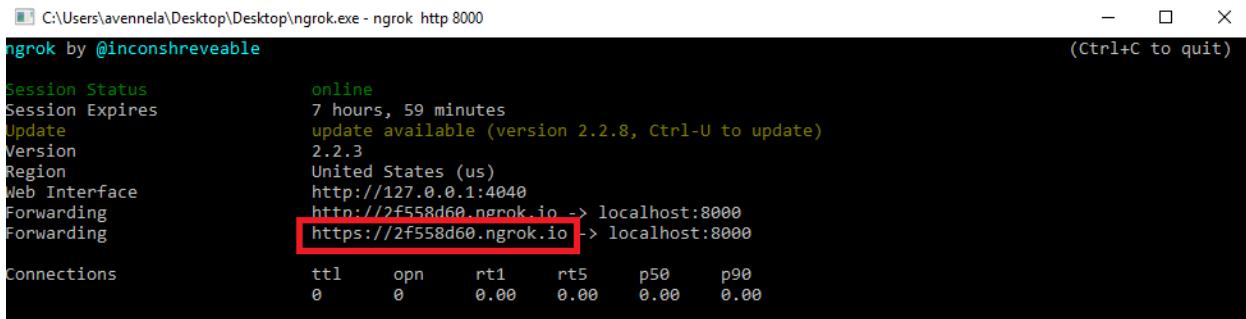
After downloading the ngrok, extract the downloaded folder and run the **ngrok.exe** file. Now, give the following command and click on the Enter button.

**ngrok http <Your-Application-Port>**

**Example:** `ngrok http 8000`

Once you run the application successfully with the above command, you will get the URL as [https://<random\\_code>.ngrok.io](https://<random_code>.ngrok.io). Copy this for later use.

**Note** - You should not close this window until you stop running your Node JS application.



```
C:\Users\avennela\Desktop\Desktop\ngrok.exe - ngrok http 8000
ngrok by @inconshreveable                                         (Ctrl+C to quit)

Session Status          online
Session Expires        7 hours, 59 minutes
Update                 update available (version 2.2.8, Ctrl-U to update)
Version                2.2.3
Region                 United States (us)
Web Interface          http://127.0.0.1:4040
Forwarding             http://2f558d60.ngrok.io -> localhost:8000
Forwarding             https://2f558d60.ngrok.io -> localhost:8000

Connections            ttl     opn     rt1     rt5     p50     p90
                        0       0      0.00    0.00    0.00    0.00
```

Now, navigate to the app that you created in your Twilio account and place the URL in the place of **WHEN MESSAGE COMES IN** as shown below.

## Step #6 | Testing AP Cloud Bot in WhatsApp

Open your WhatsApp, search for your Bot number and start a conversation with the bot.

Hurray!! With this lab you are able to create your WhatsApp using **Watson Assistant** and **Twilio**.

For any questions regarding this lab please feel free to reach out to [innovation@miraclesoft.com](mailto:innovation@miraclesoft.com).