Project Report

Name: Divya Ann Kurien (divya.kurien@gmail.com)
Title: Intelligent Customer Help Desk With Smart Document Understanding
Category: Artificial Intelligence

Internship at smartinternz.com@2020

CONTENTS

Introduction

Overview

We will be able to write an application that leverages multiple WatsonAI Services (Discovery, Assistant, Cloud function, and Node-Red). By the end of the project, we'll learn best practices of combining Watson services, and how they can build interactive information retrieval systems with Discovery + Assistant.

- **Project Requirements :** Python, JavaScript, IBM Cloud, IBM Watson
- **Functional Requirements :** IBM cloud
- **Technical Requirements :** AI, ML, IBM Watson services, Python
- **Software Requirements :** IBM Watson Assistant, IBM Watson Discovery, Node-Red
- **Project Deliverables :** Intelligent Customer Helpdesk Chatbot with SDU
- **Project Team :** Individual project work [Divya Ann Kurien].
- **Project Duration :** 29 days.

Purpose

The typical customer care chatbot can answer simple questions, such as store locations and hours, directions, and maybe even making appointments. When a question falls outside of the scope of the pre-determined question set, the option is typically to tell the customer the question isn't valid or offer to speak to a real person.

In this project, there will be another option. If the customer question is about the operation of a device, the application shall pass the question onto Watson Discovery Service, which has been pre-loaded with the device's owner's manual. So now, instead of "Would you like to speak to a customer representative?" we can return relevant sections of the owner's manual to help solve our customers' problems.

To take it a step further, the project shall use the Smart Document Understanding feature of Watson Discovery to train it on what text in the owner's manual is important and what is not. This will improve the answers returned from the queries.

Scope of Work

- Create a customer care dialog skill in Watson Assistant
- Use Smart Document Understanding to build an enhanced Watson Discovery collection
- Create an IBM Cloud Functions web action that allows Watson Assistant to post queries to Watson Discovery
- Build a web application with integration to all these services & deploy the same on IBM Cloud Platform

LITERATURE SURVEY

Existing Problem

A standard chatbot will provide solutions for basic queries like "location of the store" or "when does the store open". But when the query is something related to the operation of a device, the bot will say like “try again”, “I don’t understand”, “will you repeat again”, and so on… instead of displaying the solution, it will direct the customer to a customer representative.

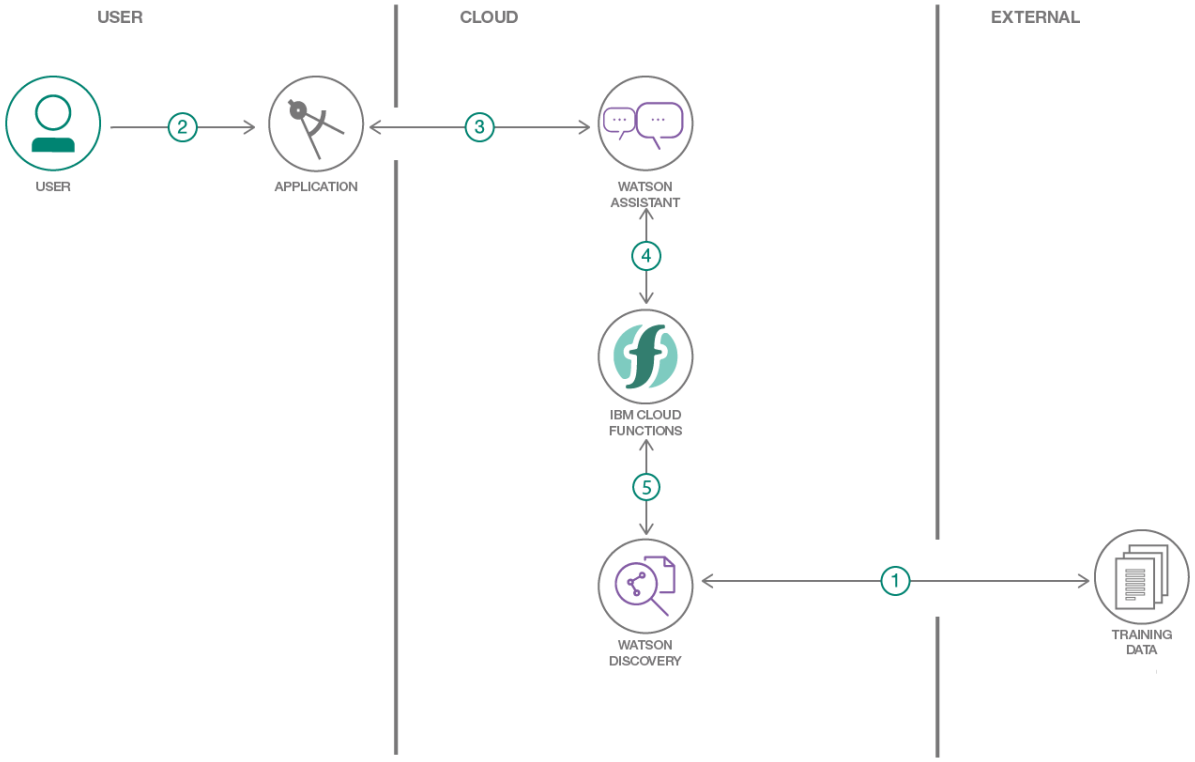
Due to this, the customer has to wait until the representative gets back to them. Sometimes they don't provide useful solutions to the query. Another problem is these chatbots have limited responses for customers. This overall deteriorates the customer experience.

Proposed Solution

For the above problem to get solved we have to build an Intelligent Chatbot with Smart Document Understanding, so it can understand the queries that are posted by customers. The chatbot should be trained from some insight records based company background, user manuals, etc... so it can answer queries based on the product or related to the company. In this project, I have used IBM Watson Discovery along with IBM Watson Assistant to achieve the above solution.

THEORETICAL ANALYSIS

Block Diagram



1. The document is annotated using Watson Discovery SDU
2. The user interacts with the backend server via the app UI. The frontend app UI is a chatbot that engages the user in a conversation.
3. Dialog between the user and backend server is coordinated using a Watson Assistant dialog skill.
4. If the user asks a product operation question, a search query is passed to a predefined IBM Cloud Functions action.
5. The Cloud Functions action will query the Watson Discovery service and return the results.

Hardware/Software Designing.

The software utilized are Watson Discovery, Watson Assistant, and Node-Red.

1. **Watson Discovery**: Discovery makes it possible to rapidly build cognitive, cloud-based exploration applications that unlock actionable insights hidden in unstructured data. It applies the latest breakthroughs in machine learning, including natural language processing capabilities, and is easily trained on the language of your domain. It breaks open the data silos and retrieves specific answers to user's questions while analyzing trends and relationships buried in the uploaded data. With Smart Document Understanding (SDU) feature you can train IBM Watson Discovery to extract custom fields in your documents.

2. **Watson Assistan**t: It is a conservation AI platform that provides customers fast, straightforward and accurate answers to their questions, across any application, device, or channel. By addressing common customer inquiries, Watson Assistant reduces the cost of customer interactions.

It uses Watson's AI machine learning (ML) and natural language understanding (NLU). The user input is received by the assistant and routes it to a dialog skill. The dialog skill interprets the input further, then directs the flow of the conservation. The dialog gathers any information that needs to respond to the user's behalf.

3. **Node-Red**: It is a prototyping tool that builds applications easily. Applications are developed by building data flows through a series of connected nodes. Intricate applications can be built that allow Watson services to interact with a range of capabilities and services exposed as Node-Red nodes.

It helps in wiring together hardware devices, APIs, and online services without writing any code. It provides a web-based flow editor, which can be used to connect these services and also create UI.

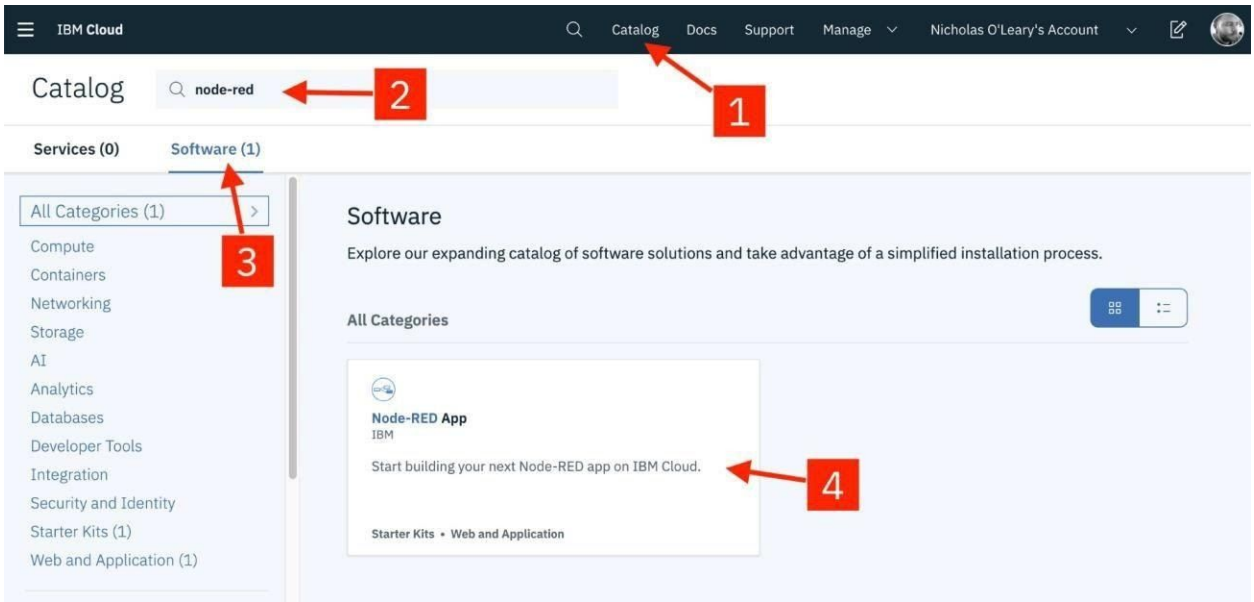
EXPERIMENTAL INVESTIGATION

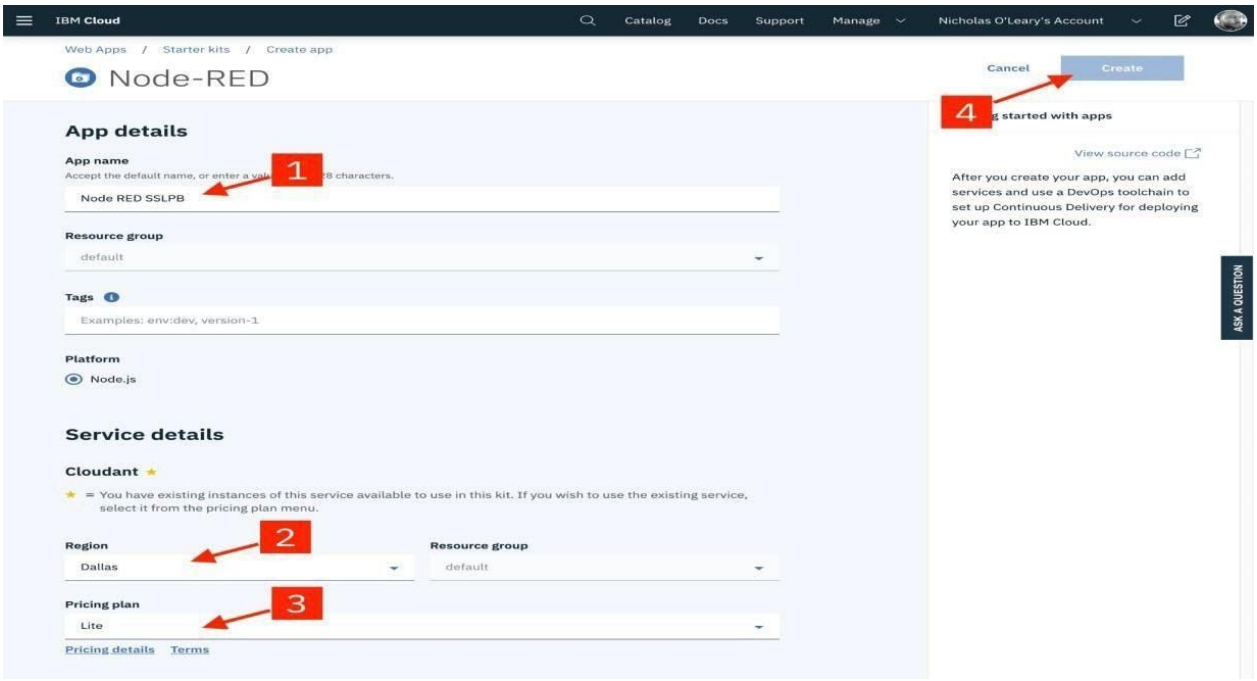
The Following Services are to be created:

1. Node-Red Application
2. Watson Discovery
3. Cloud Functions
4. Watson Assistant

1. Node-Red Application

1. Search for Node-Red in the catalog and create an application by filling in details like the cloudant region and a unique name.

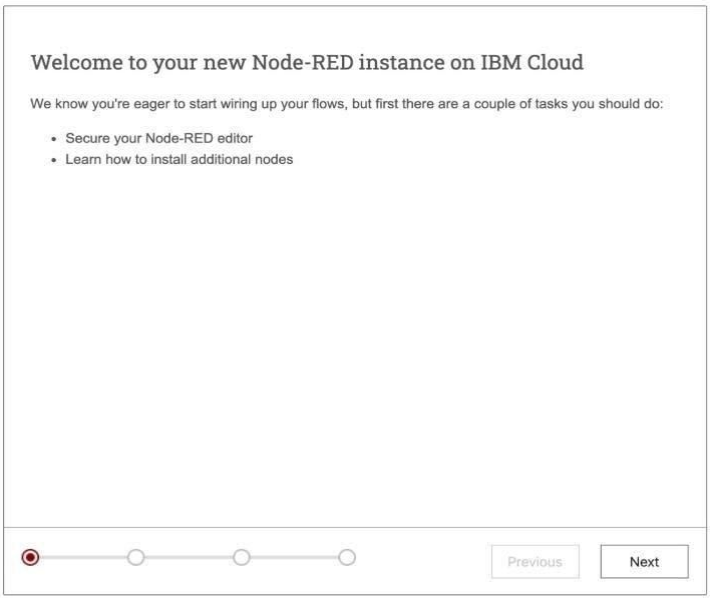




2. Setup Continous Delivery feature to deploy the application created into the Cloud foundry apps in IBM Cloud.

After deploying stage is completed, the application is now running.

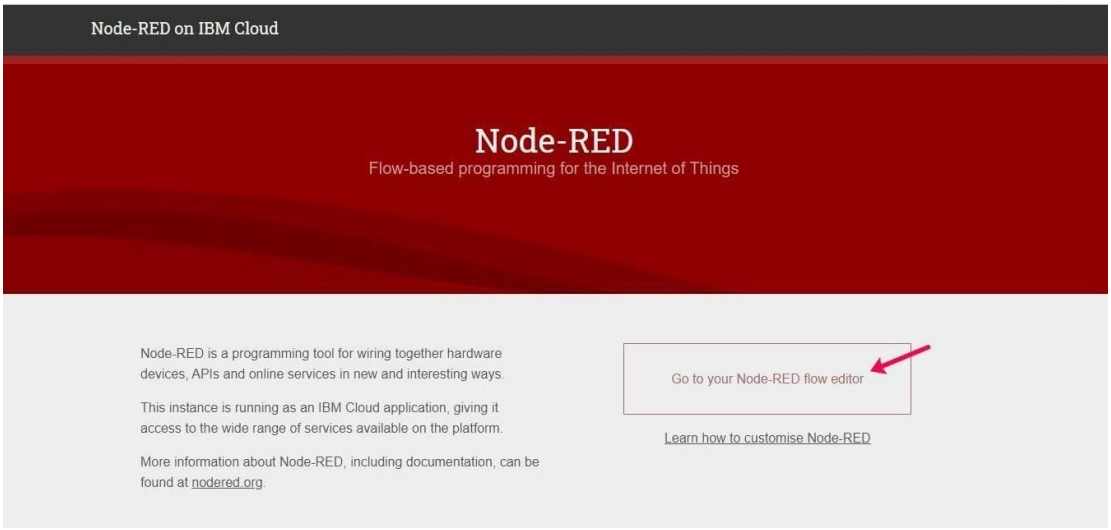
3. In the details page, click on Visit App URL to configure it and set up security.



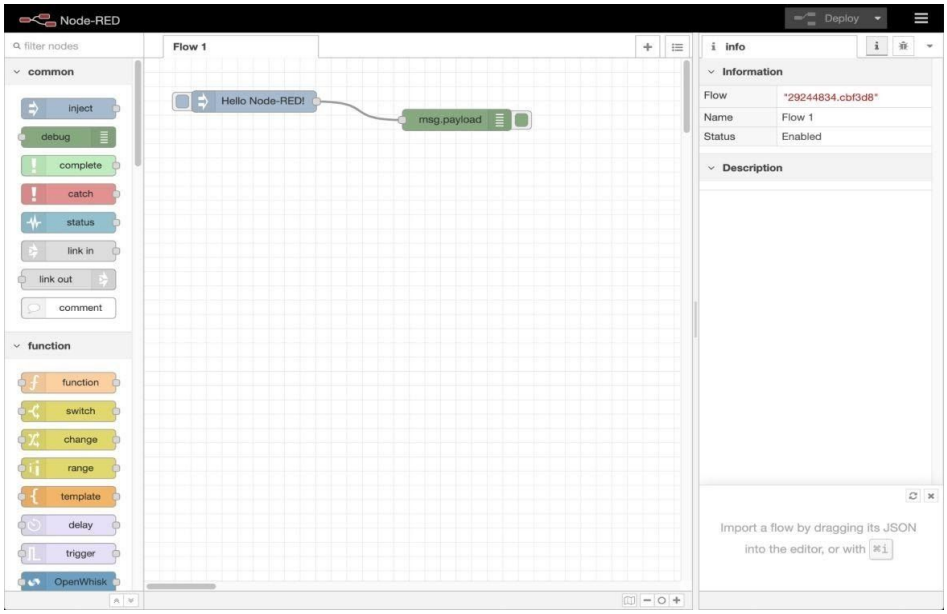
4. Click on Next. Now it will ask for username and password for security reasons. Enter them.

5. The final screen summarizes the options you've made and highlights the environment variables you can use to change the options in the future. Click Finish to proceed.

6. Node-RED will save your changes and then load the main application. From here you can click the Go to your Node-RED flow editor button to open the editor



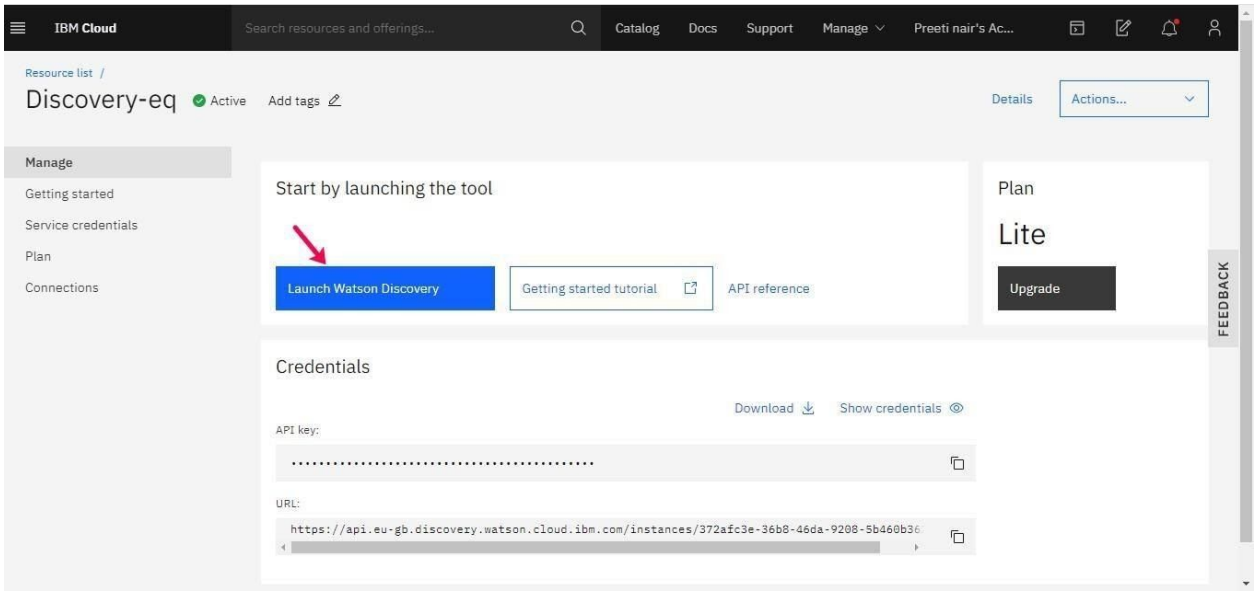
7. This is the place where you will create the flow to wire all the services utilized to function the chatbot.



2. Watson Discovery

1. Search for Discovery from Catalog and create an instance of the service.

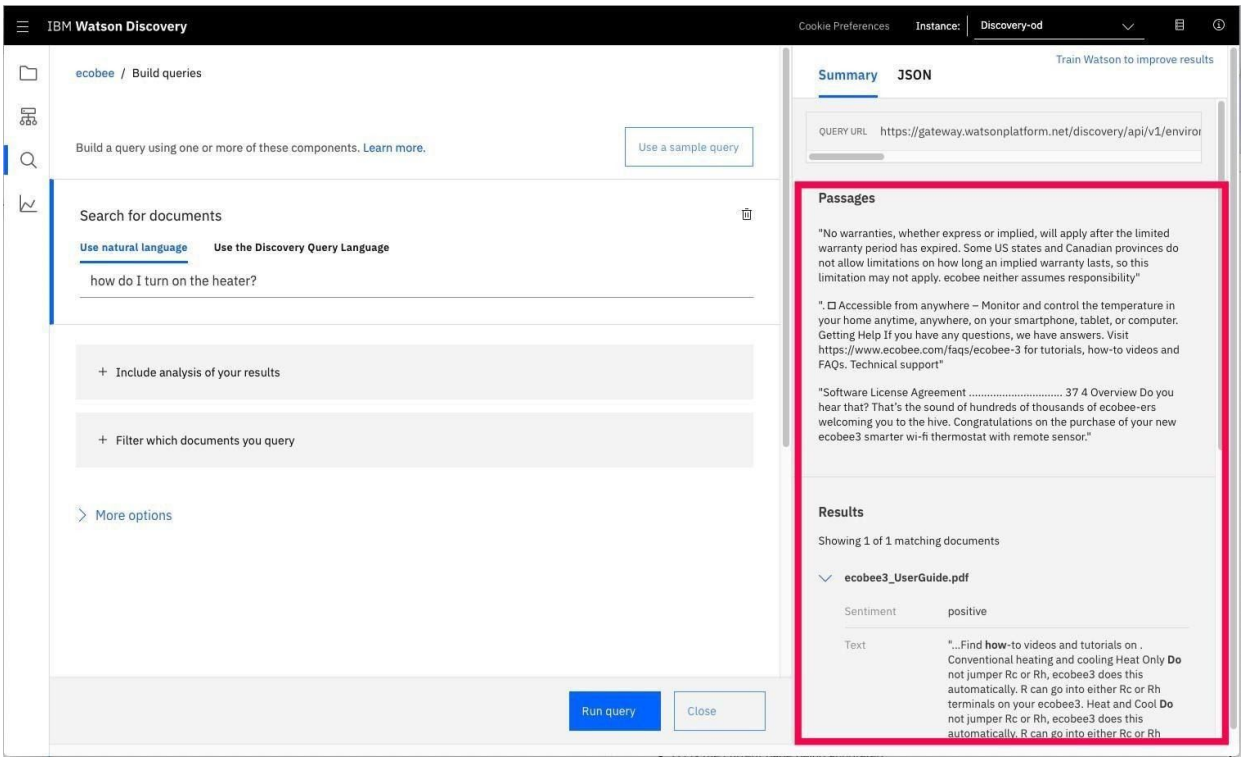
2. After creating, launch the service by clicking on Launch Watson Discovery.



Upload the ecobee3 owner's manual by clicking on Upload your own data.

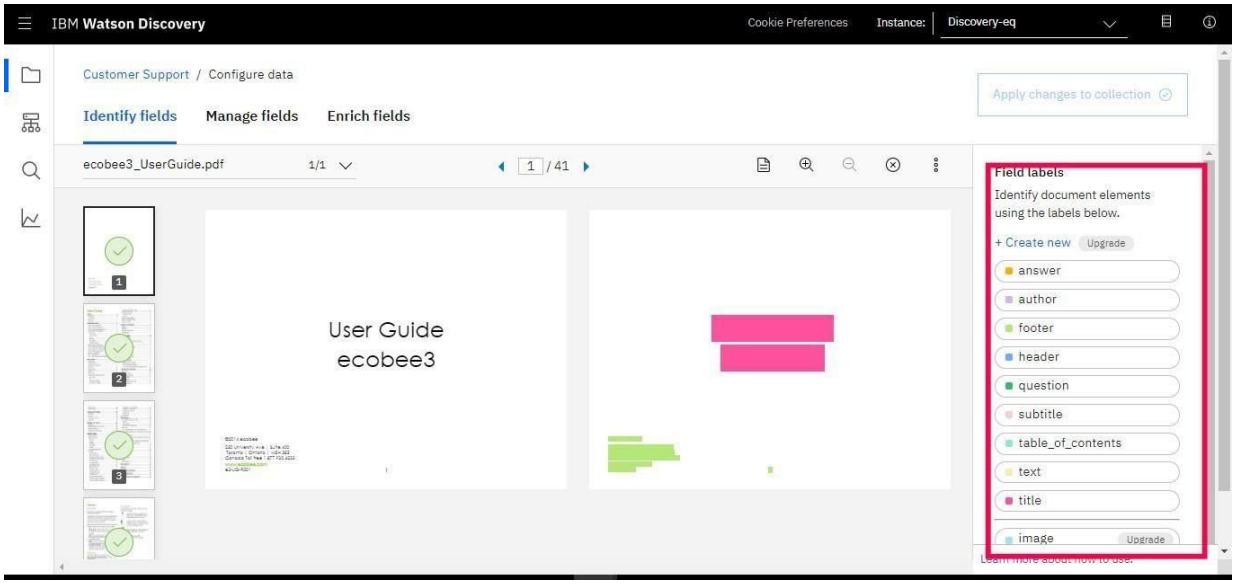
4. Now before setting up the SDU feature, try a few queries. Click on the Build your own query.

5. When asking a query, it will display you many sections from the manual. To improve query result, setup SDU feature.



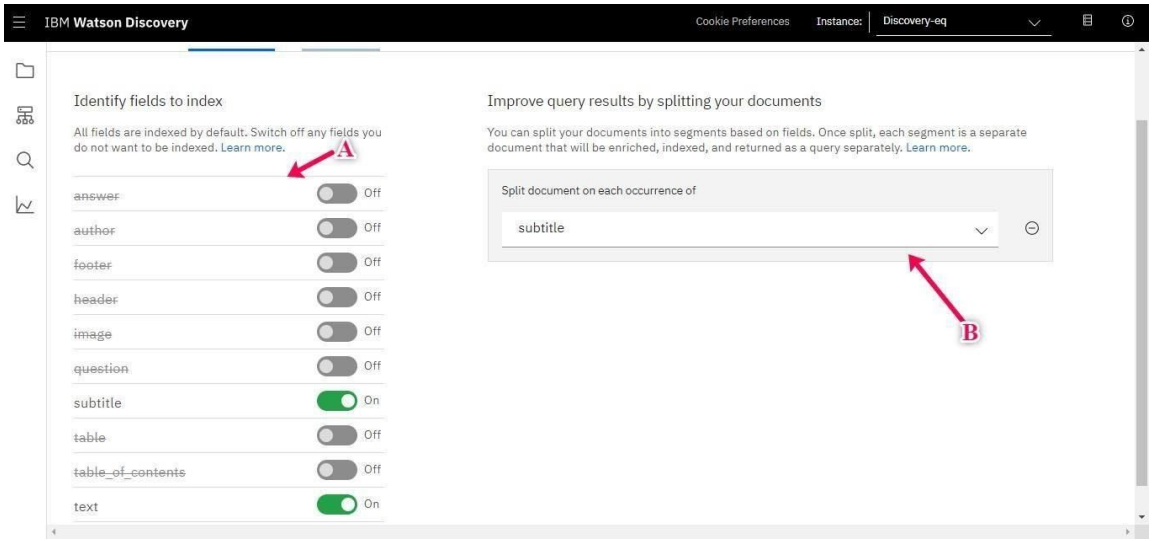
6. Go to Configure data. You will see the SDU layout. There are three fields: Identify fields, Manage fields, and Enrich fields.

7. In Identify field, here you will identify the elements in the document by labeling them as title, subtitle, or text which you will see on the right side of the layout page.



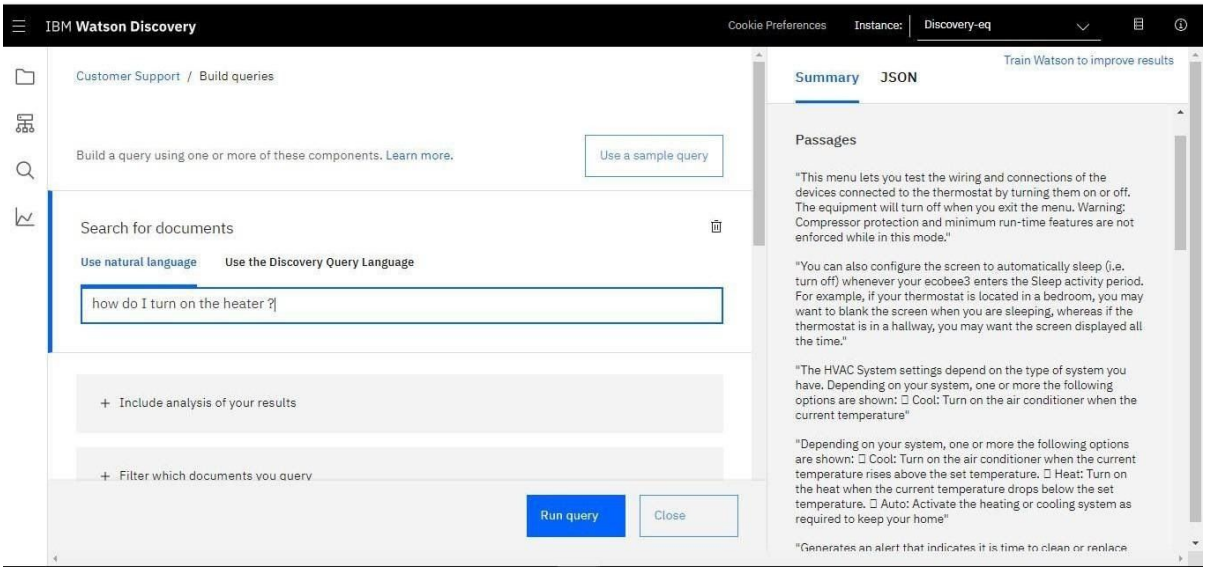
8. Assign relevant labels for 10-11 pages and click on Submit Page accordingly. After a few pages, Discovery starts to recognize the element automatically. This means the service is trained.

9. Now go to Manage Fields tab, select the items you want to be indexed (A). After that, split the document by the subtitle (B). Click the Apply to changes button.



10. It will ask you to upload the document, upload it. You will see all the annotations applied to your index.

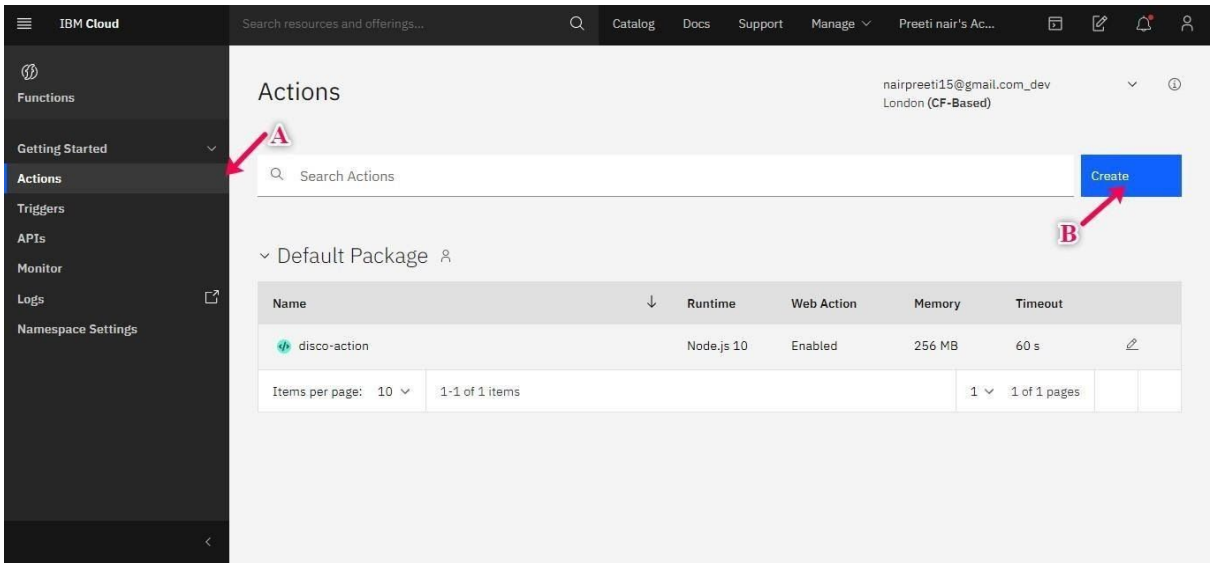
11. Again click on Build your own query. Ask the same query. This time you will see the most relevant answer from the manual being displayed



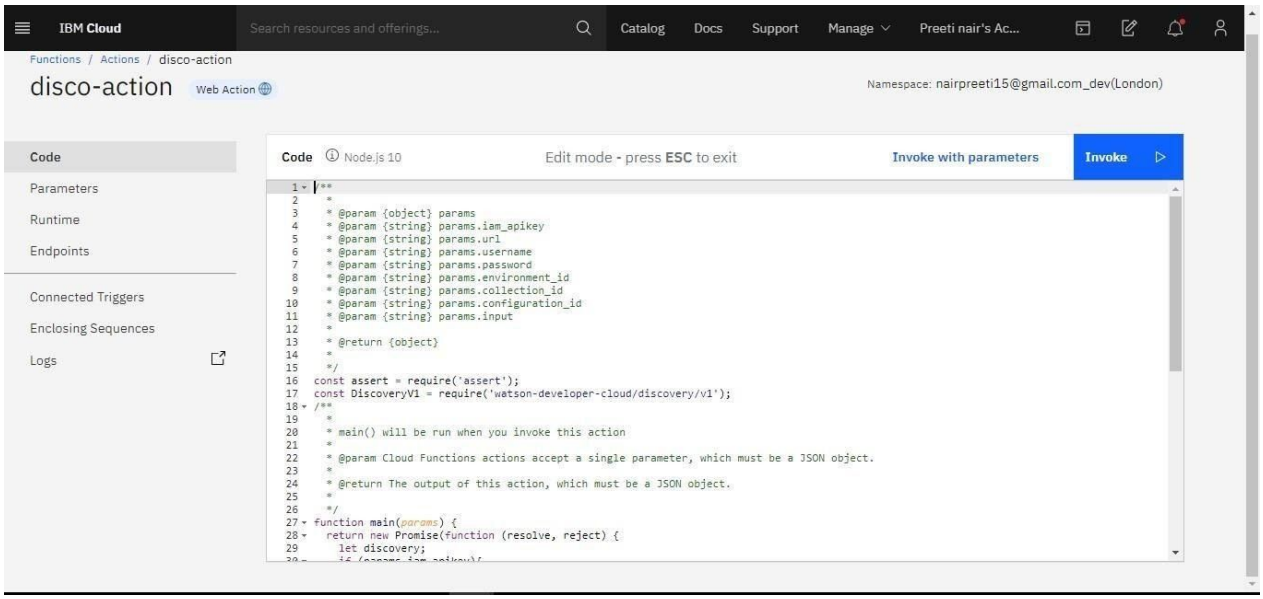
3. Cloud Functions

1. Search for Functions in the catalog and click on Functions.

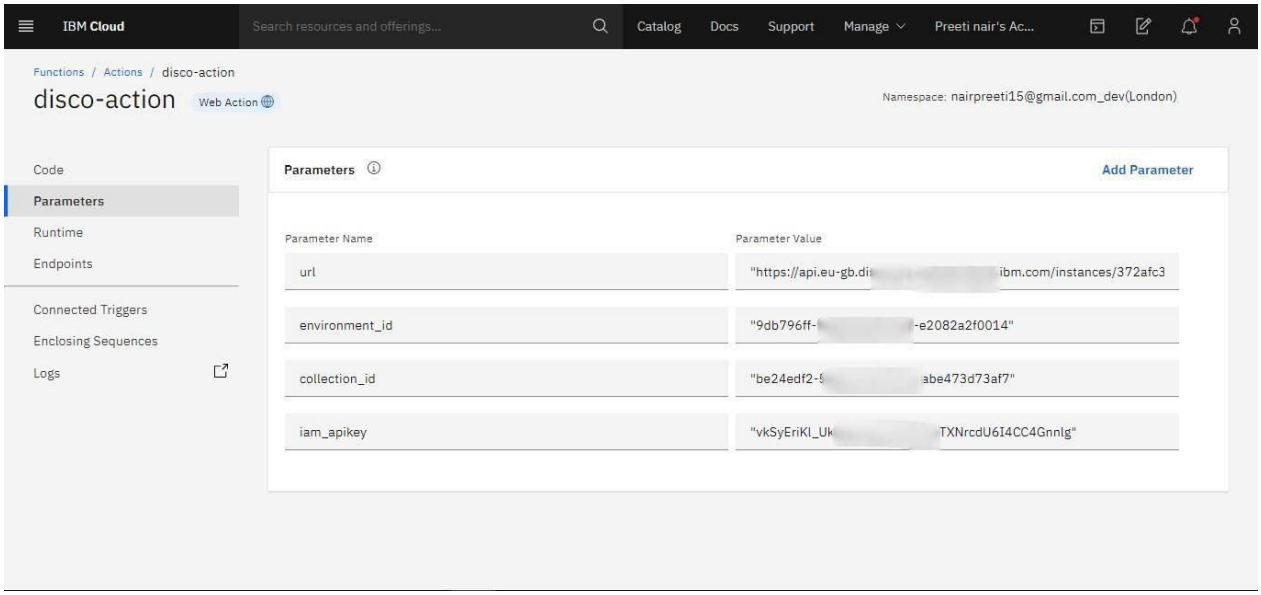
2. Opening Functions, in the left tab, click on Actions (A) and select create (B).

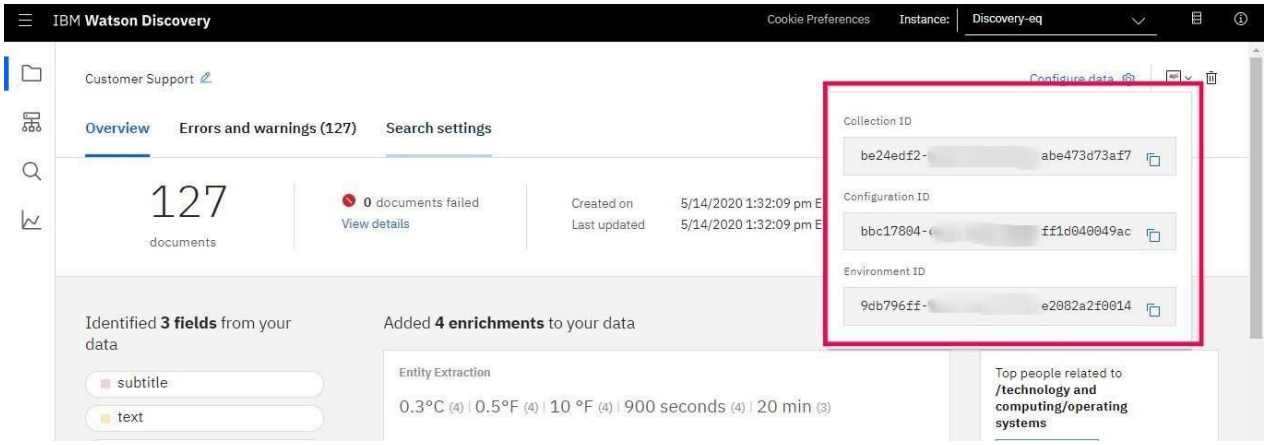


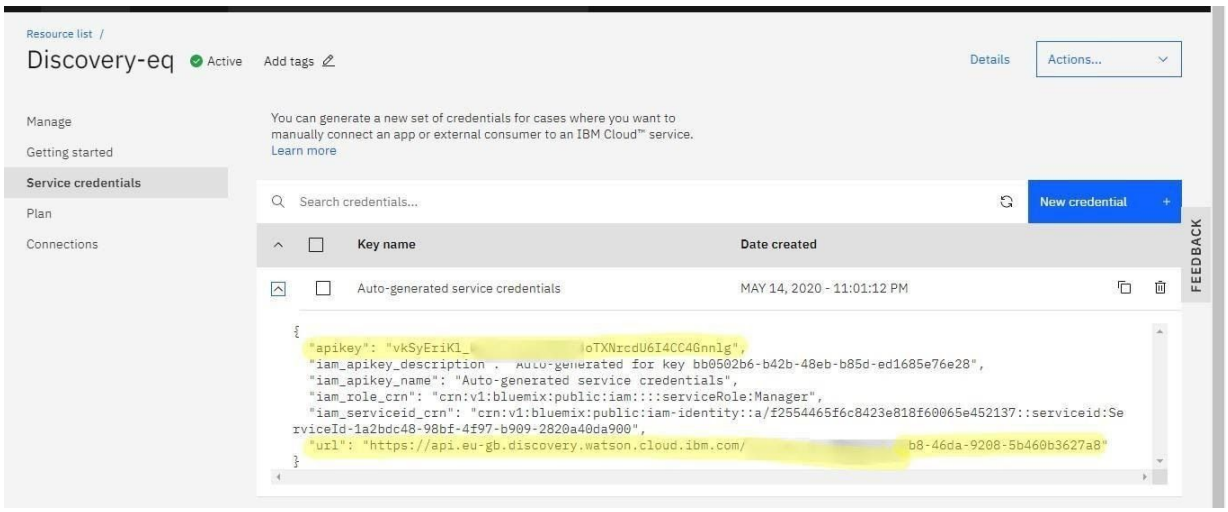
3. Once the action is being created, go to the code tab. Paste the code which will revoke the Discovery service.



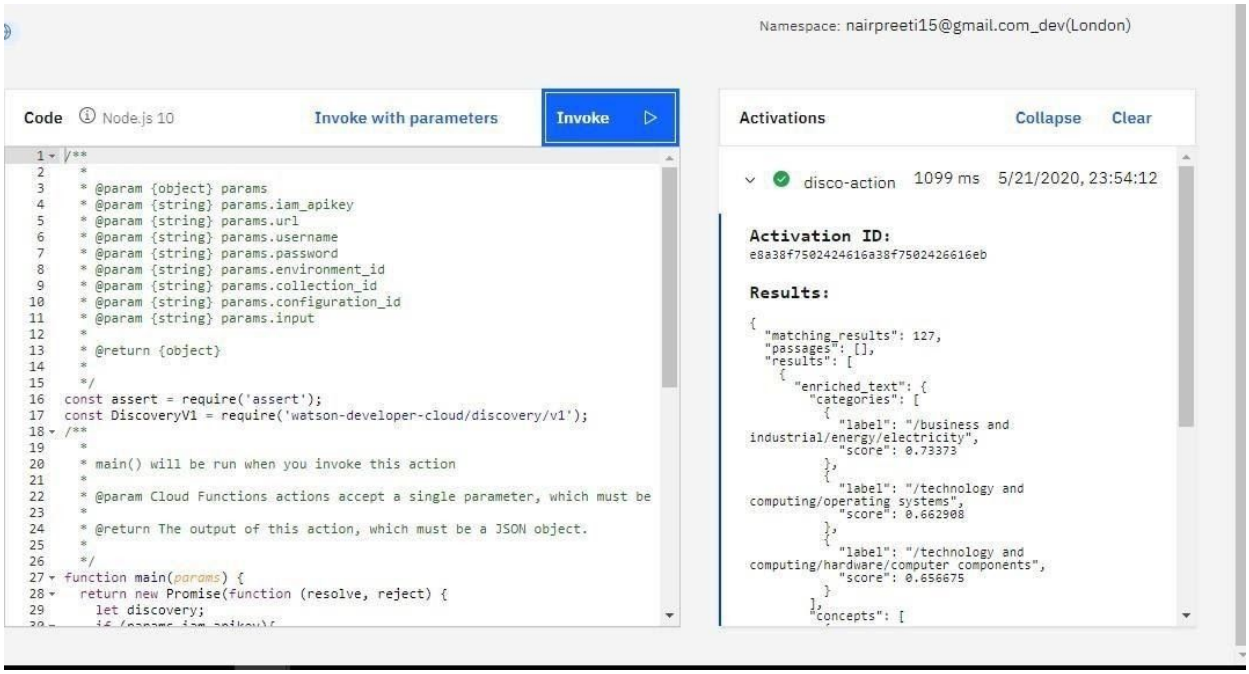
4. Before clicking on Invoke button, click on Parameters tab and add all necessary parameters of the Discovery service which you will find on the following pages.


IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
Preeti nair's Ac...
Functions / Actions / disco-action
disco-action
Web Action
Namespace: nairpreeti15@gmail.com_dev(London)
Code
Parameters
Runtime
Endpoints
Connected Triggers
Enclosing Sequences
Logs
Parameters
Add Parameter
Parameter Name
Parameter Value
url
"https://api.eu-gb.di ibm.com/instances/372afc3
environment_id
"9db796ff- -e2082a2f0014"
collection_id
"be24edf2-5 abe473d73af7"
iam_apikey
"vkSyEriKl_Uk TXNrcdU6I4CC4Gnnlg"

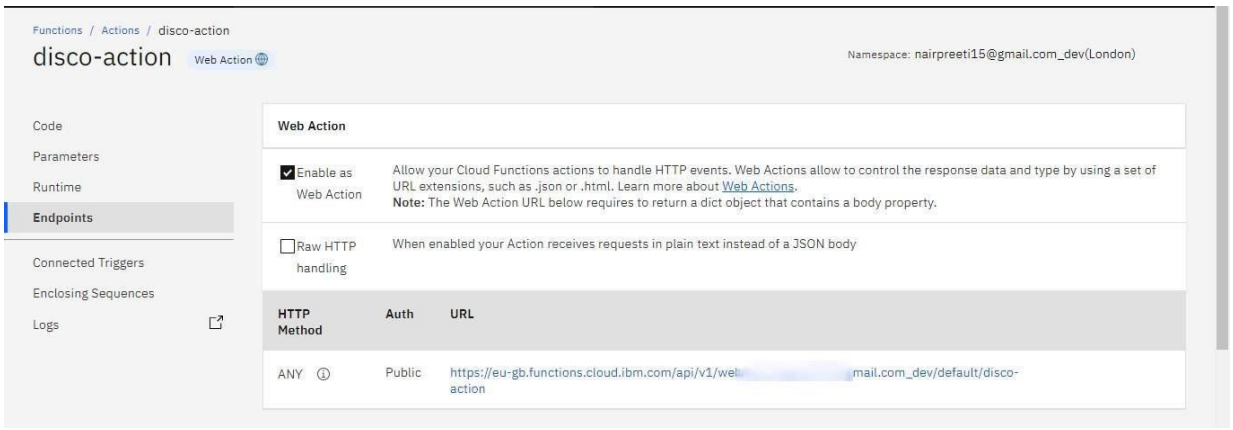

IBM Watson Discovery
Cookie Preferences
Instance:
Discovery-eq
Customer Support
Configure data
Overview
Errors and warnings (127)
Search settings
127
documents
0 documents failed
View details
Created on
5/14/2020 1:32:09 pm E
Last updated
5/14/2020 1:32:09 pm E
Collection ID
be24edf2- abe473d73af7
Configuration ID
bbc17804- ff1d040049ac
Environment ID
9db796ff- e2082a2f0014
Identified 3 fields from your data
subtitle
text
Added 4 enrichments to your data
Entity Extraction
0.3°C (4) | 0.5°F (4) | 10 °F (4) | 900 seconds (4) | 20 min (3)
Top people related to /technology and computing/operating systems


Resource list /
Discovery-eq
Active
Add tags
Details
Actions...
Manage
Getting started
Service credentials
Plan
Connections
You can generate a new set of credentials for cases where you want to manually connect an app or external consumer to an IBM Cloud™ service.
Learn more
Search credentials...
New credential
Key name
Date created
Auto-generated service credentials
MAY 14, 2020 - 11:01:12 PM
"apikey": "vkSyEriKl_ oTXNrcdU6I4CC4Gnnlg",
"iam_apikey_description": "Auto-generated for key bb0502b6-b42b-48eb-b85d-ed1685e76e28",
"iam_apikey_name": "Auto-generated service credentials",
"iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Manager",
"iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/f2554465f6c8423e818f60065e452137::serviceid:ServiceId-1a2bdc48-98bf-4f97-b909-2820a40da900",
"url": "https://api.eu-gb.discovery.watson.cloud.ibm.com/ b8-46da-9208-5b460b3627a8"
FEEDBACK

5. After adding parameters, go back to the code page and click on Invoke. You will see the actual results returned from the Discovery service.



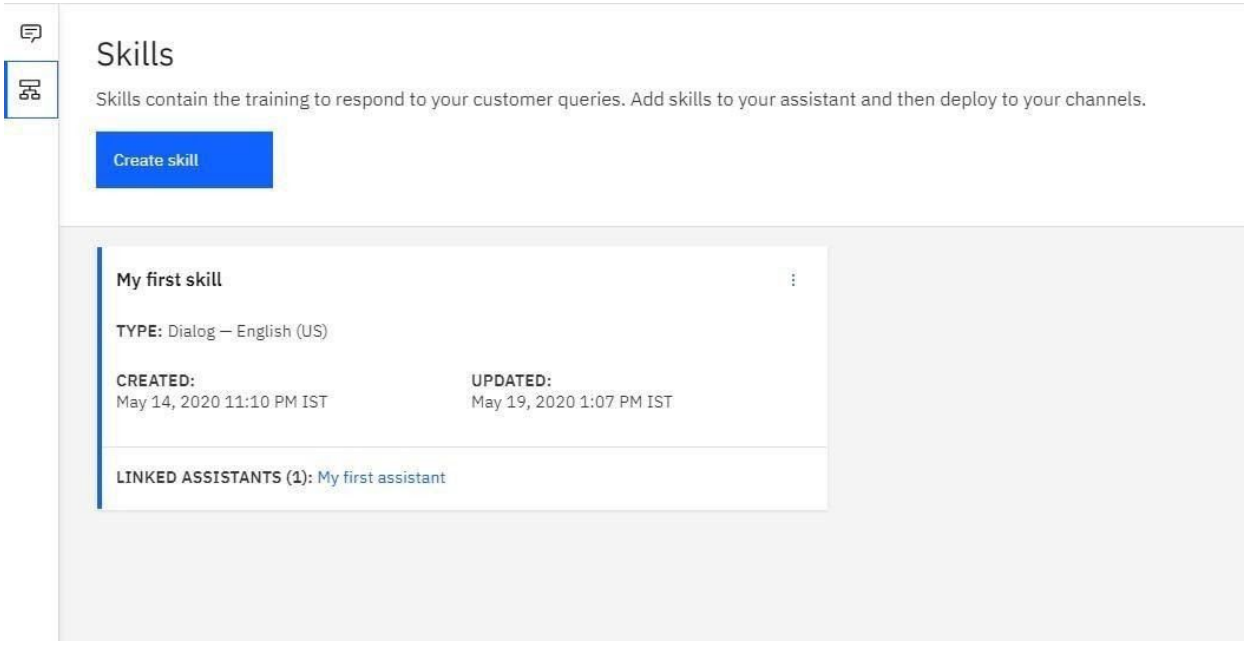
6. In Cloud Functions, click on Endpoints tab, and select Enable as Web Action. This will create a URL that will be used to set up Webhook in Watson Assistant.



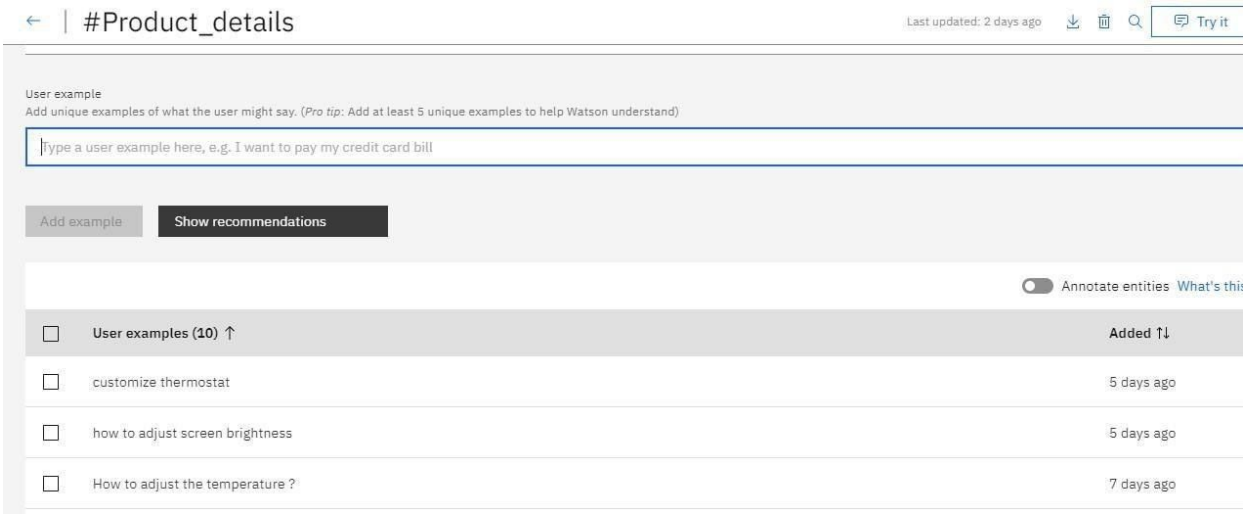
4. Watson Assistant

1. Create Watson Assistant service in a similar fashion as Watson Discovery. Launch the service.

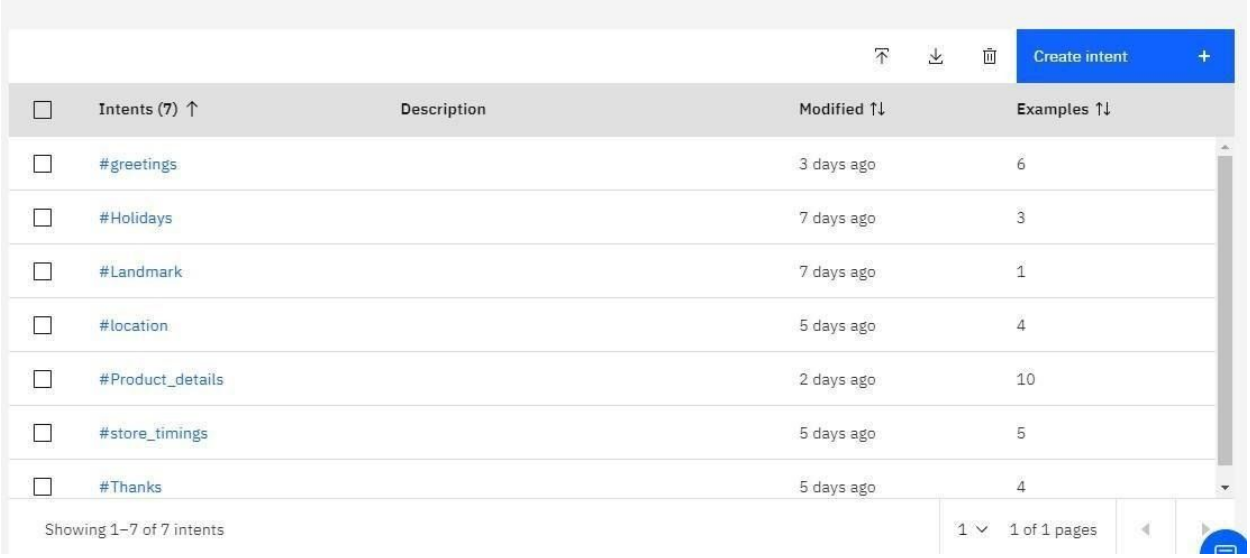
2. When you open the service, you will see a dialog skill named Customer Care skill automatically provided by the service. Either choose that and remodify it or create new skill.



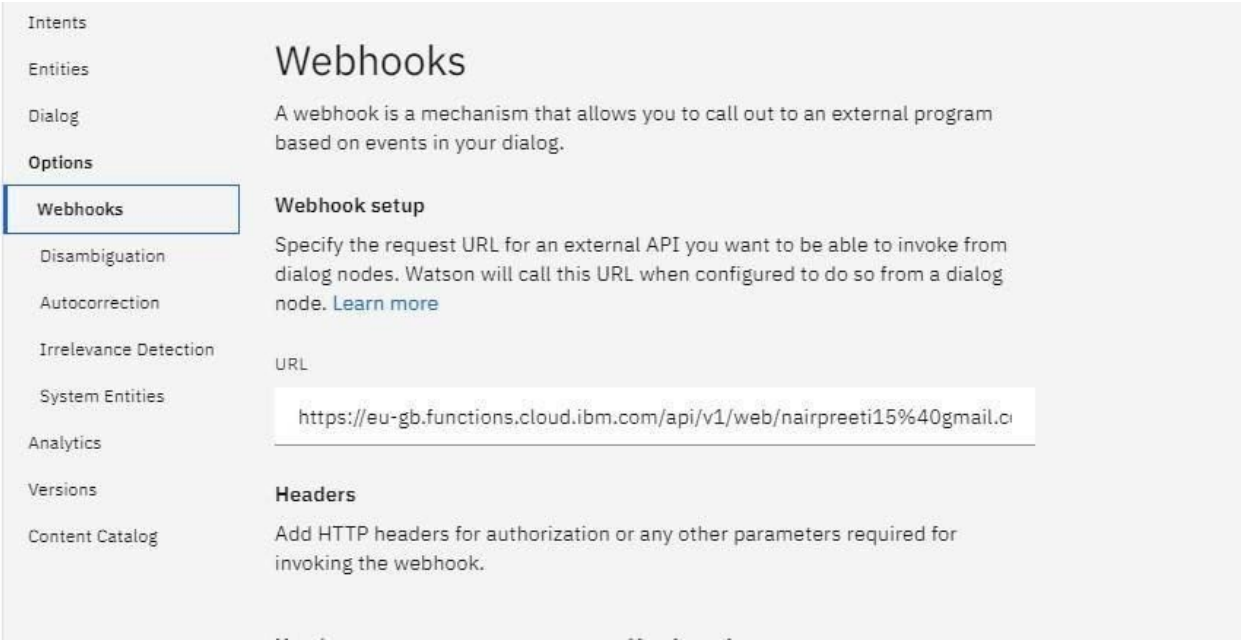
3. Firstly create intent which we will detect if the customer is asking about the operation of the product. Given down below is intent about Product_details. Provide few sample examples of queries for the assistant to detect.



Similarly, you can give intents for location, landmarks or timings.



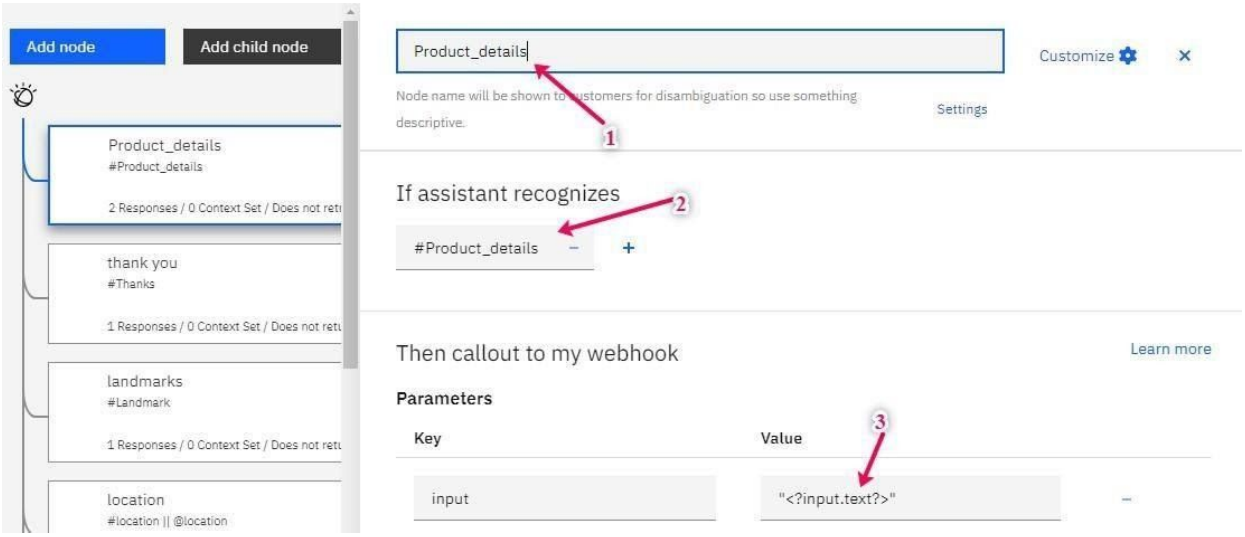
4. Click on Options. Here you will setup Webhook. Use the URL from the Web action of Cloud functions in the Endpoints tab.



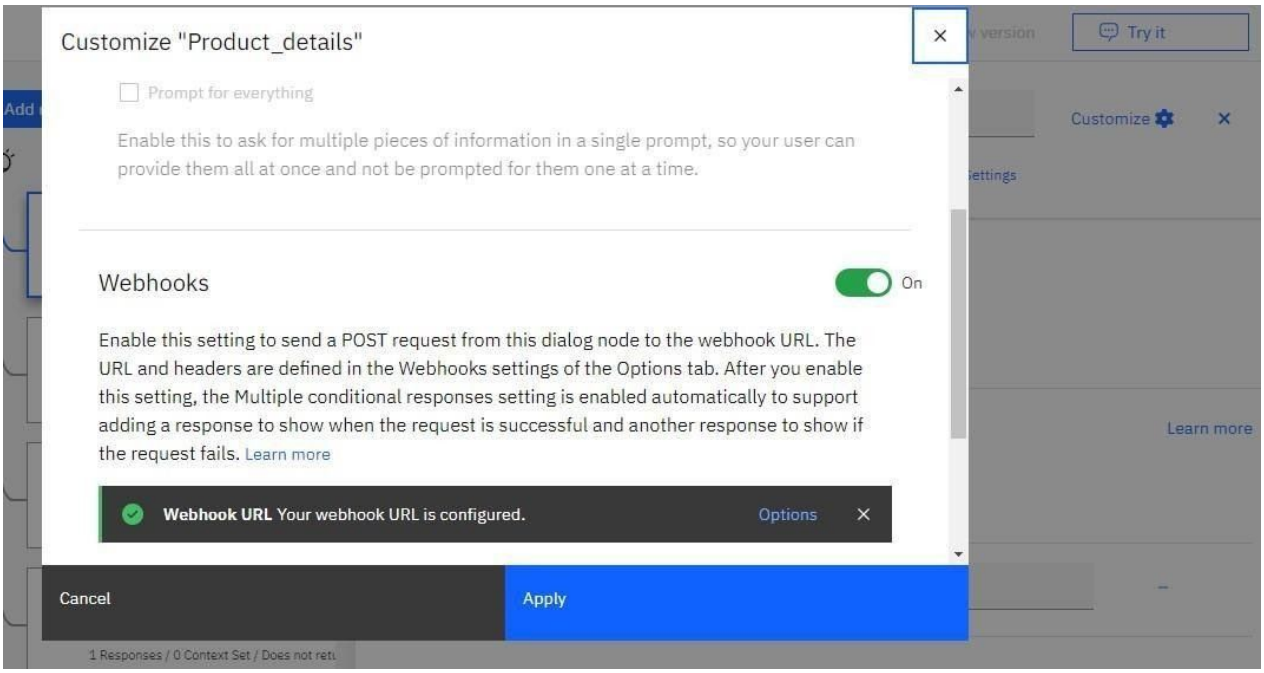
5. Move onto the Dialog tab. Here, add nodes to handle intents.

6. Create a node for Product_details (1) and accordingly assign its intent (2).

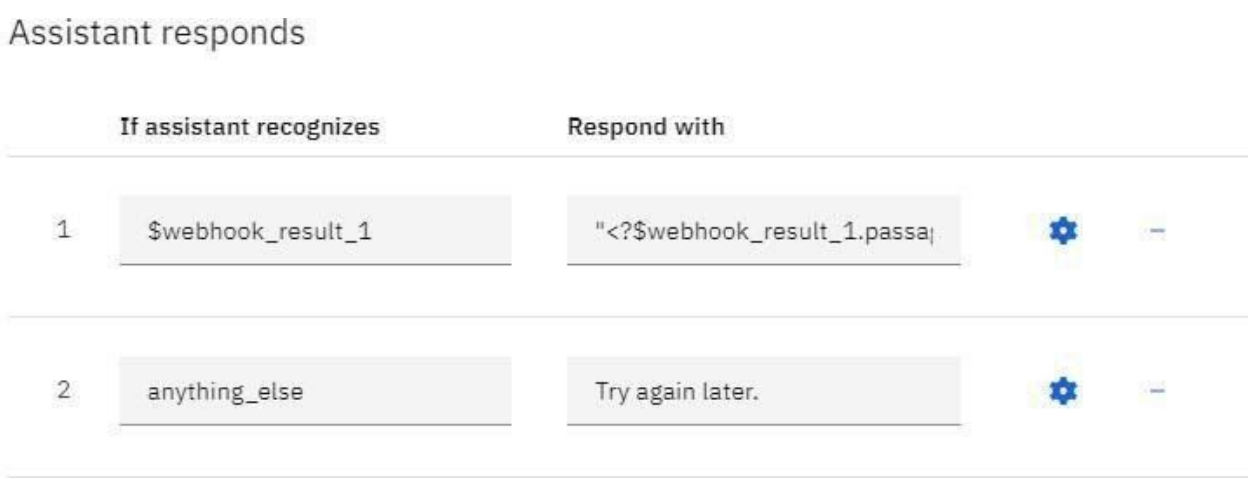
7. Enter syntax (3) which will pass on the query via input parameter to discovery.



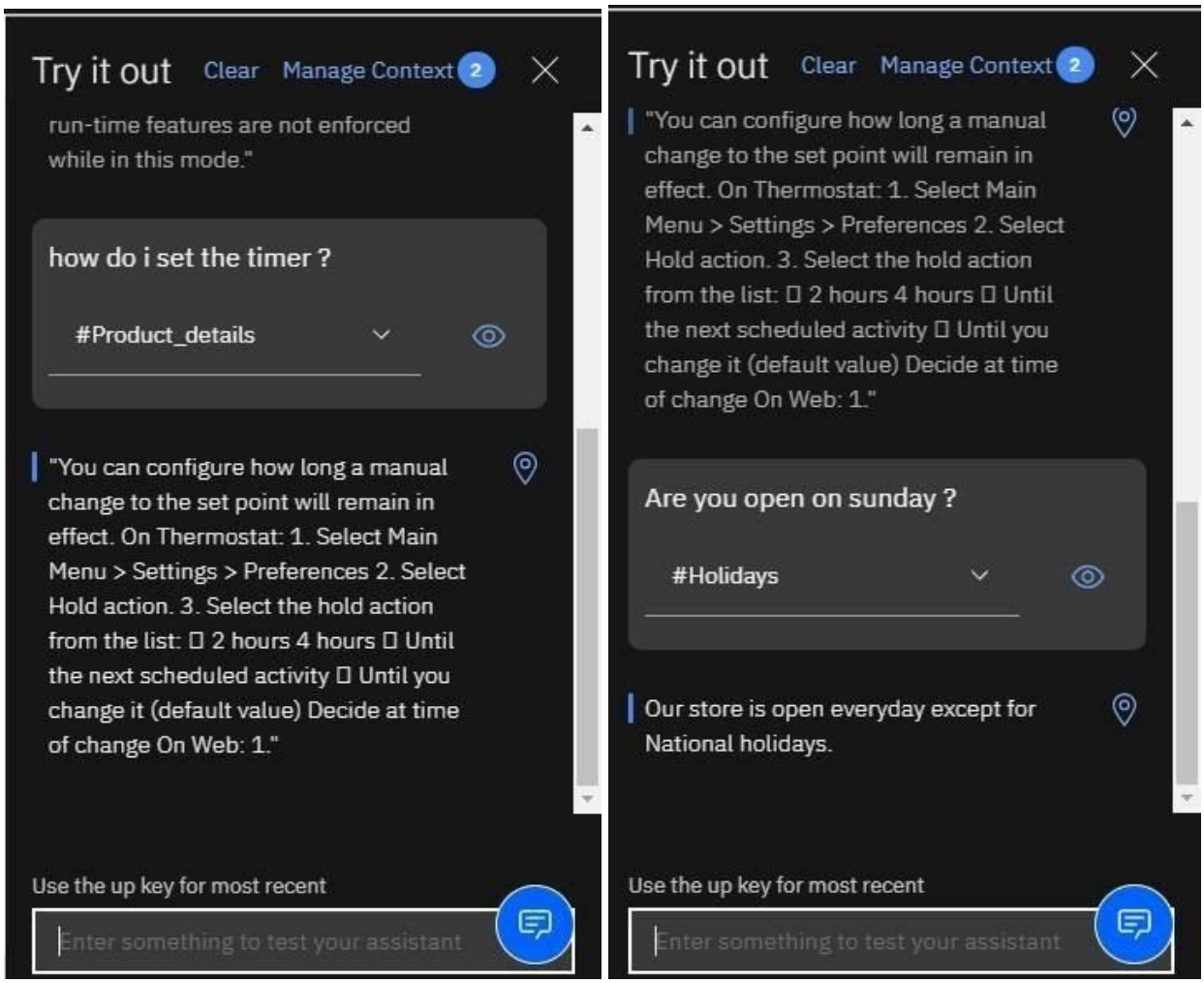
8. Click on Customize and enable the Webhook.



9. Now add the responses to aid in debugging.

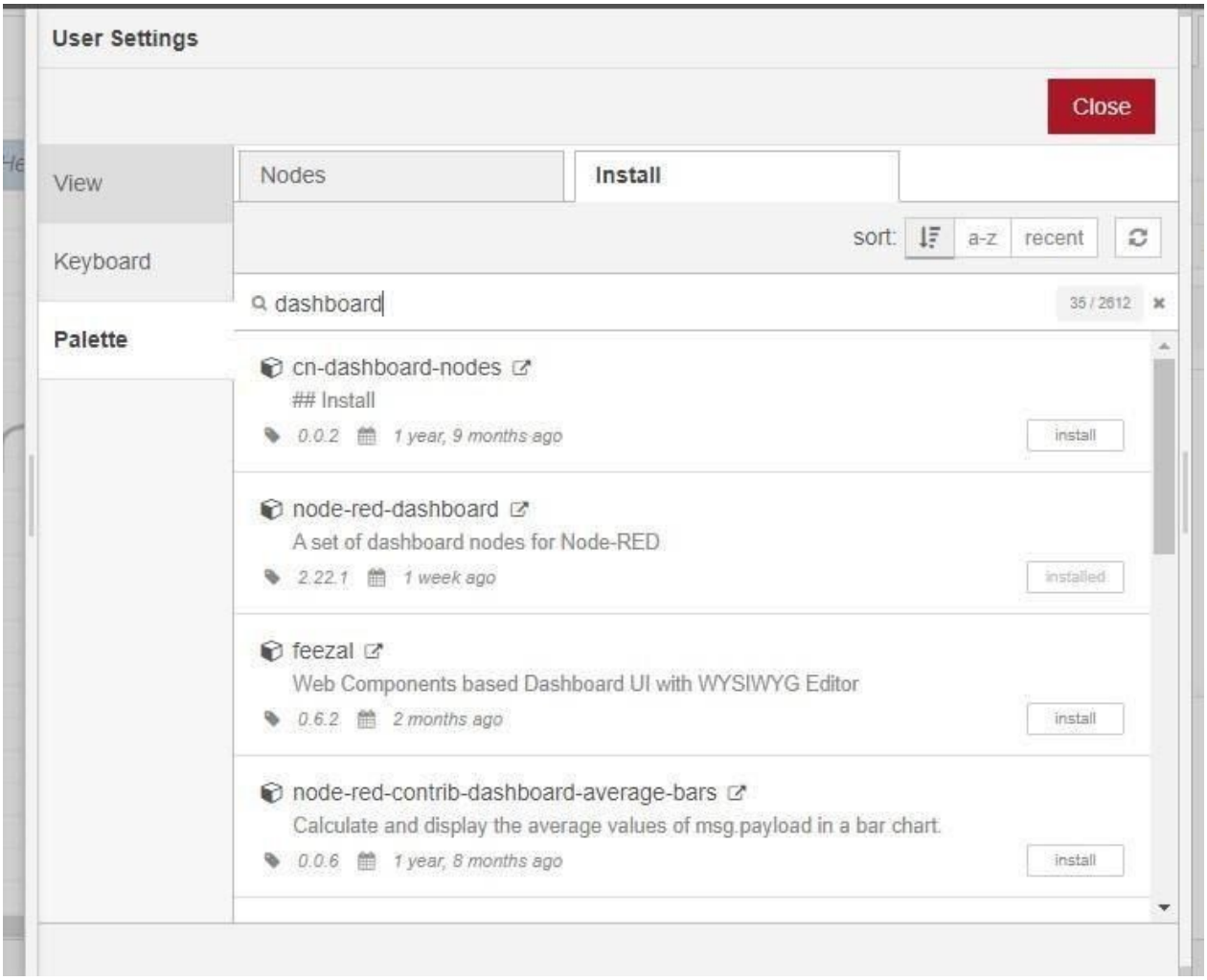


10. Try out the assistant.



Integration of all the above services in NODE-RED

1. Open the Node-Red Flow editor. Before beginning, download dashboard nodes used for creating chatbot UI. Click on Manage Palette. Go to the install tab and search for node-red-dashboard. Then click on install.



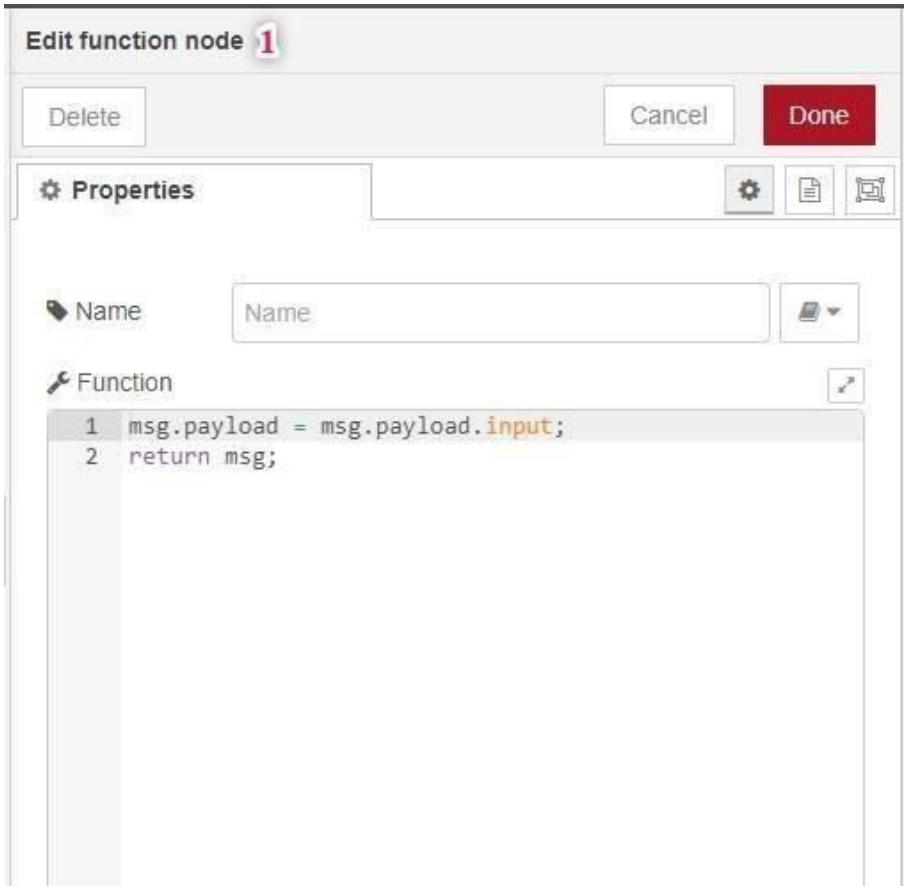
2. Add the Form node. Double click on it and add the Group name and Tab name. Also, select the size of form.

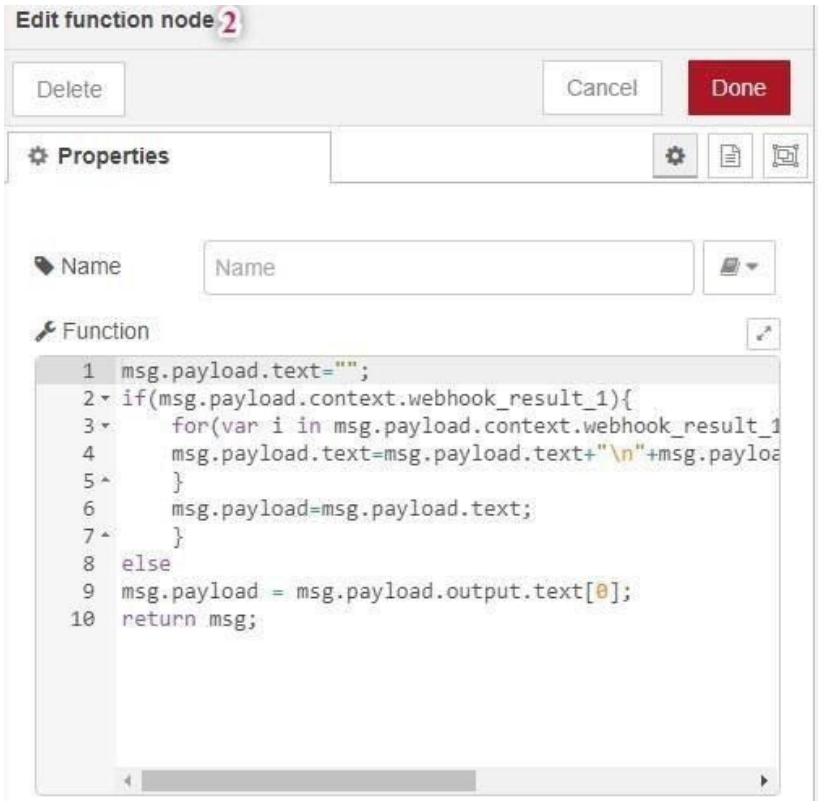
3. Add Inject node. Double click it and name it as Hello.

4. Add two Text nodes. Double click the first one and name it as Query and the second one as the Response.

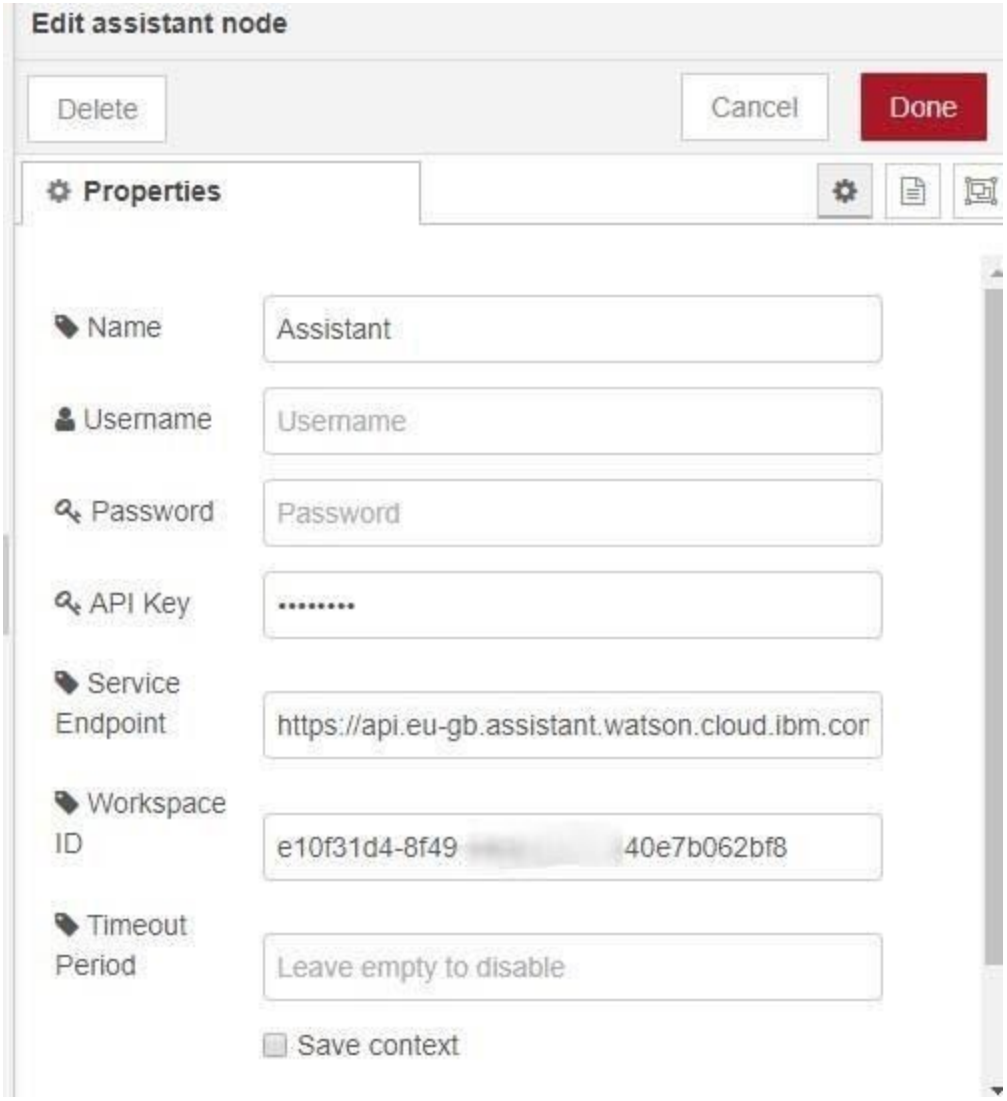
5. Drag two Debug nodes.

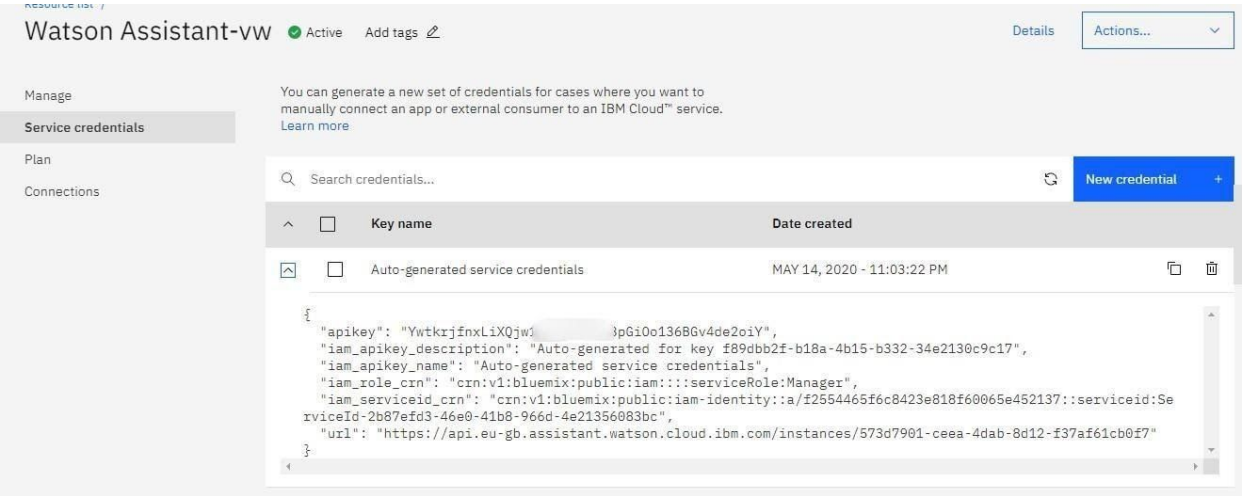
6. Drag two Functions nodes. Add in the codes for both the nodes.


Edit function node
1
Delete
Cancel
Done
Properties
Name
Name
Function
1 msg.payload = msg.payload.input;
2 return msg;


Edit function node
2
Delete
Cancel
Done
Properties
Name
Name
Function
1 msg.payload.text="";
2 if(msg.payload.context.webhook_result_1){
3 for(var i in msg.payload.context.webhook_result_1
4 msg.payload.text=msg.payload.text+"\n"+msg.payloa
5 }
6 msg.payload=msg.payload.text;
7 }
8 else
9 msg.payload = msg.payload.output.text[0];
10 return msg;

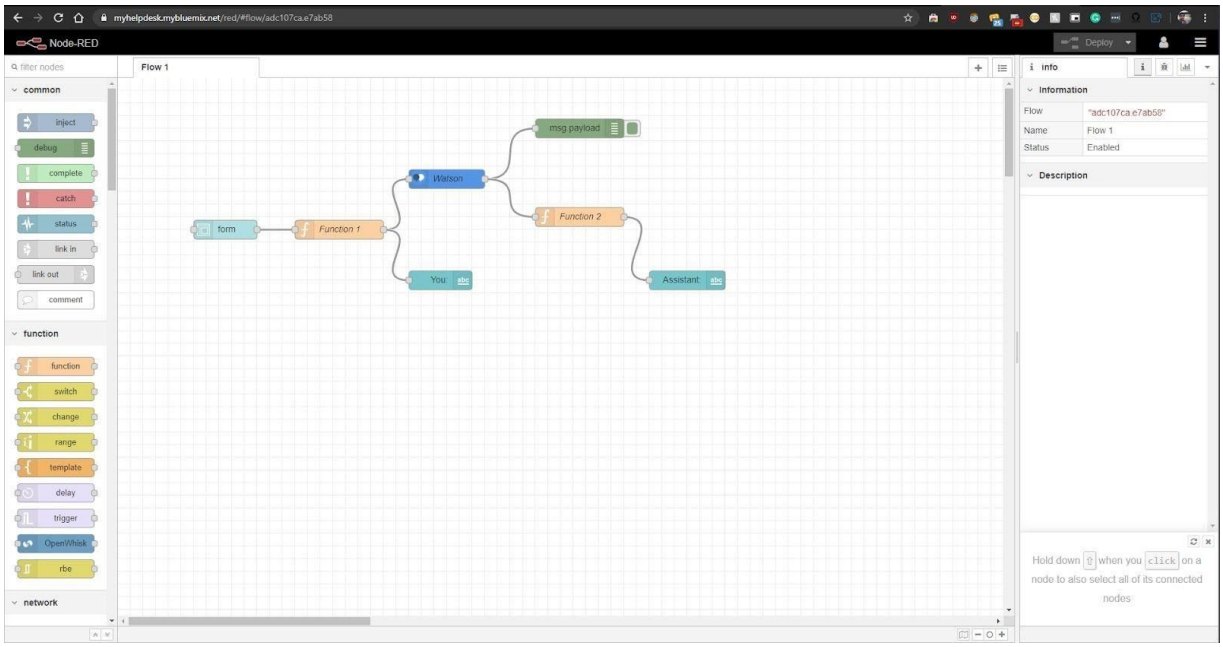
7. Drag Assistant node. Double click on it and it will ask credentials like API key, Workspace ID, and Service Endpoint which you will get from the Watson Assistant service page. Get the details and enter them and click on Done.



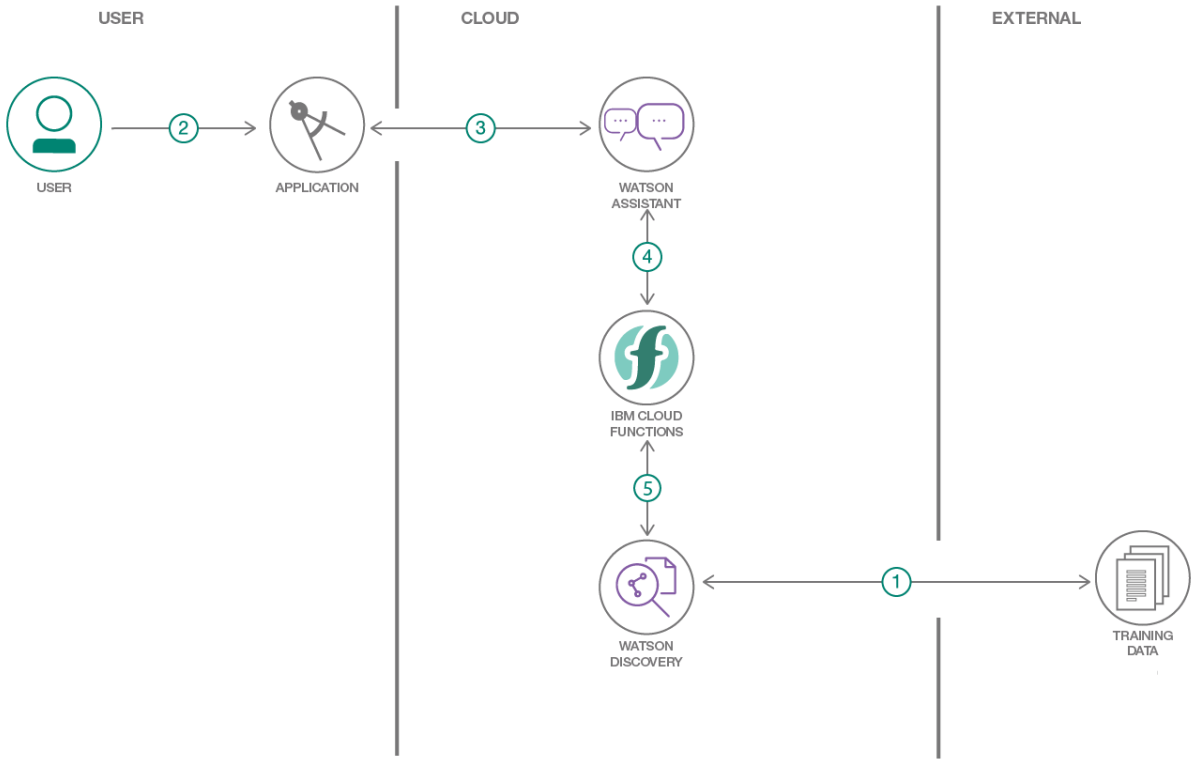


8. Connect the form node to the input of function node 1 and its output to the input of the Assistant node. Connect the output of function node 1 to the "Query" text node. Connect the output of the Assistant node to the input of function node 2 and its output to the "Response" text node.

9. Click on Deploy.



FLOWCHART



The flow summarizes the functionality of the Chatbot.

The document preloaded in Discovery is being annotated by the Smart Document Understanding feature. This trains the Discovery to improve the query result. The user interacts with frontend UI of the chatbot and it keeps them engaged in conversation with small talk. This interaction between the user and the UI is being coordinated by Watson Assistant.

When a user asks a technical query, the Assistant invokes the Cloud Function action. This action then queries Watson Discovery and returns with the relevant results from the pre-loaded owner's manual.

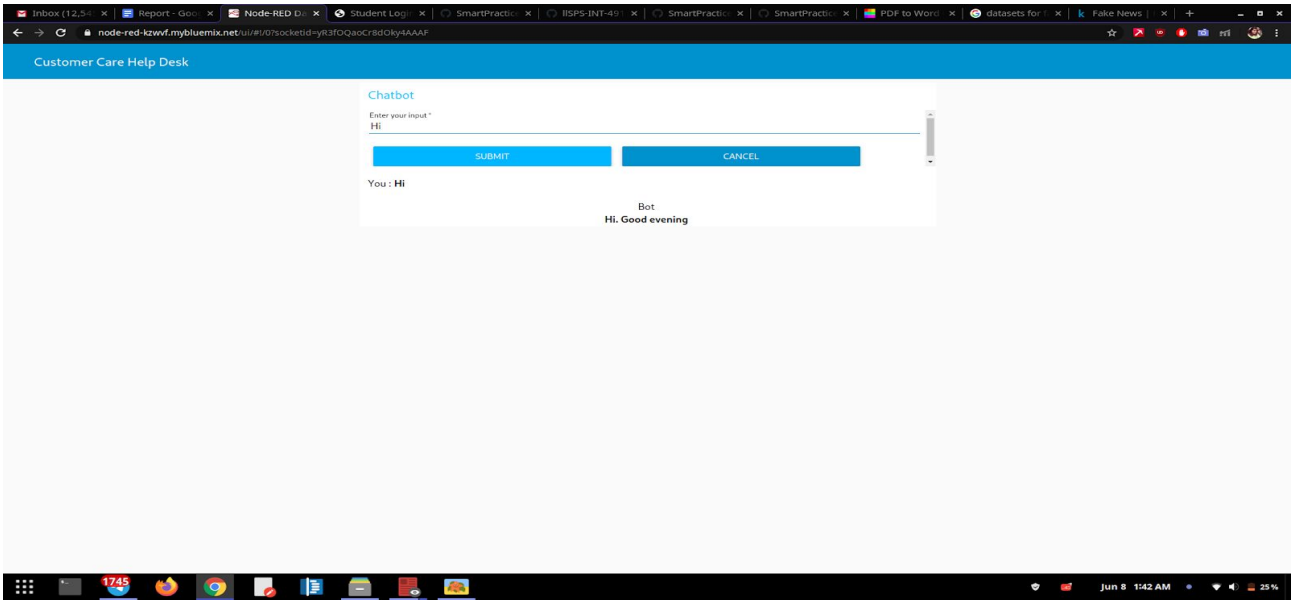
The Watson Assistant, Discovery, and Cloud Functions are IBM services. The Training Data is the owner's manual which is an external section. The user and applications come under

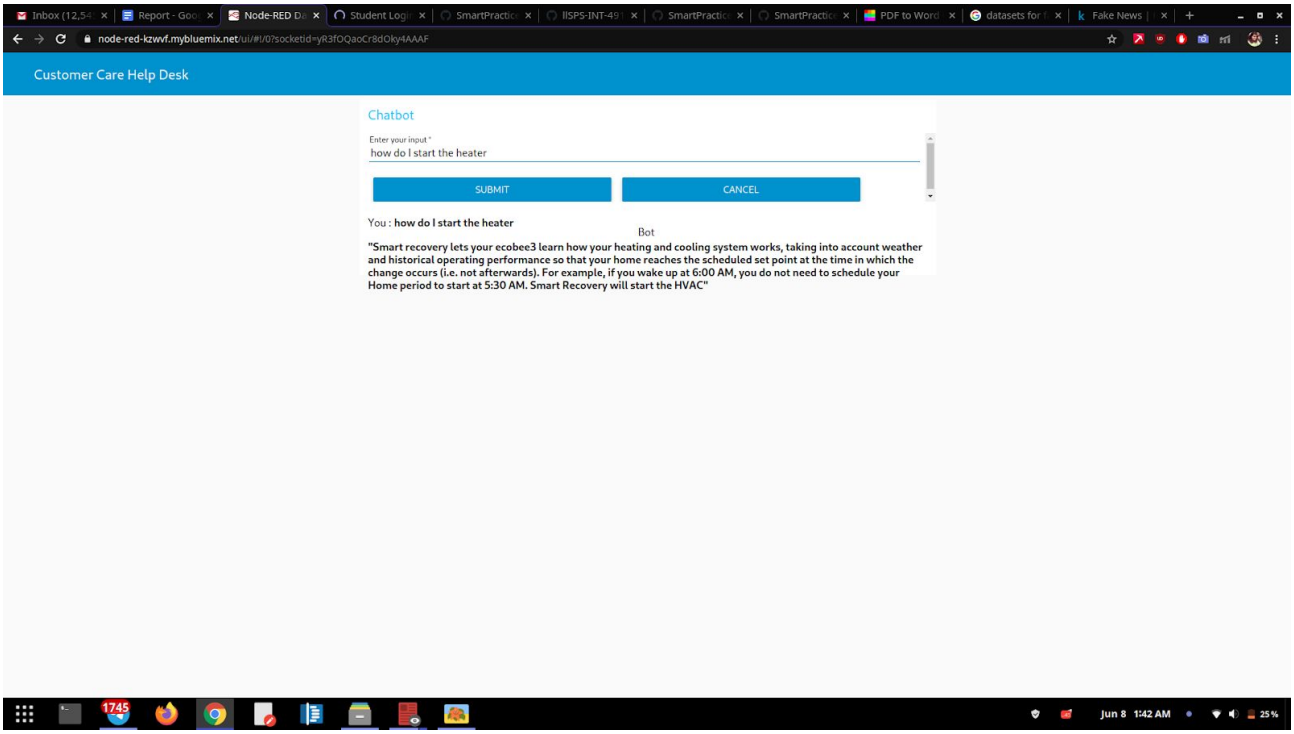
User section because the application is accessed by the user.

RESULT

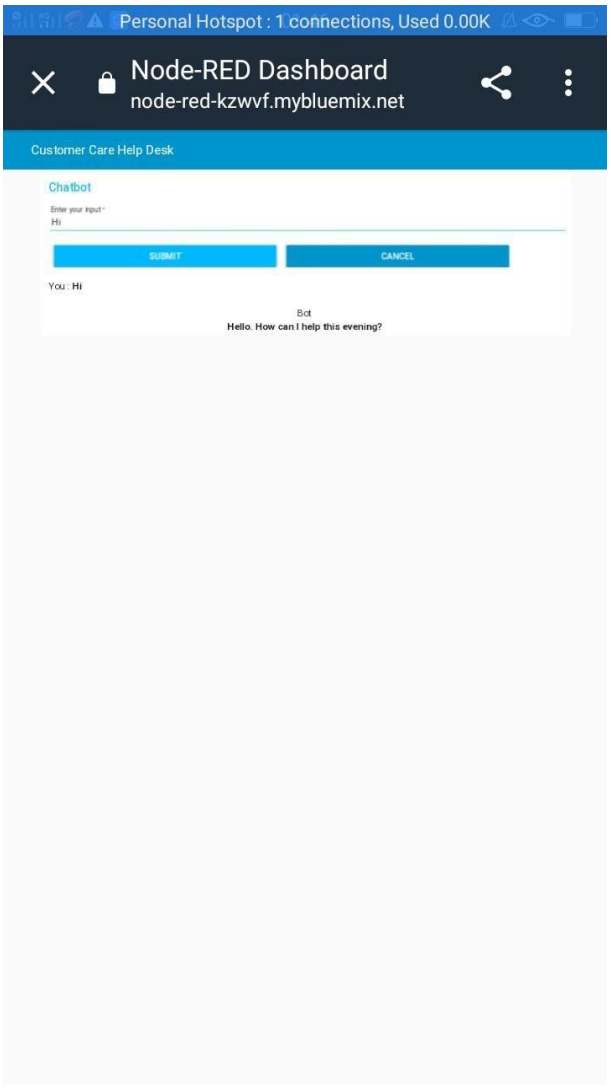
After deploying the flow in Node-Red, open the UI by going to the link https://node-red-kzwvf.mybluemix.net/ui in the browser. You will see the Help Desk created. The layout of the page can also be customized.

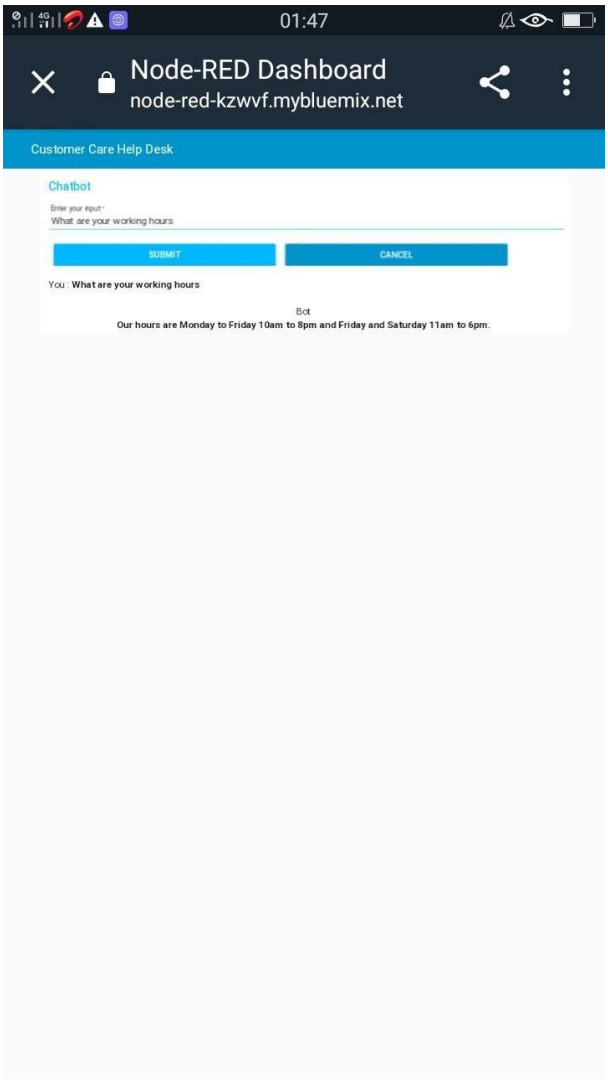
Desktop





Mobile


Personal Hotspot : 1 connections, Used 0.00K
Node-RED Dashboard
node-red-kzwvf.mybluemix.net
Customer Care Help Desk
Chatbot
Enter your input
Hi
SUBMIT
CANCEL
You : Hi
Bot
Hello. How can I help this evening?


01:47
Node-RED Dashboard
node-red-kzwvf.mybluemix.net
Customer Care Help Desk
Chatbot
Enter your input
What are your working hours
SUBMIT
CANCEL
You : What are your working hours
Bot
Our hours are Monday to Friday 10am to 8pm and Friday and Saturday 11am to 6pm.

ADVANTAGES / DISADVANTAGES

Advantages

- They are available 24/7 which maintains continuous communication between the customer and the seller.
- An operator can concentrate on one customer at a time. A chatbot can, however, answer thousands of questions at the same time.
- This reduces the cost of manpower.
- Actions like changing or querying records are almost instantaneous for bots which can significantly improve customer experience.
- Since bots are on digital platforms where people spend the majority of their time, bots can be used to automate common tasks such as providing advanced search functionality.

Disadvantages

- Chatbots are installed with the aim of getting a fast response and improving customer interaction. However, due to the limited availability of data and the time needed for self-updating, the process can be slow and costly.
- For better outcome, more complex designing is required and need of skilled developers.
- At times they can mislead customers if the query doesn't match with the pre-loaded dialog. Bots get confused and respond with the same answer for different questions. This leads to frustrated customers.
- Unlike humans, each bot needs to be programmed differently for each business, which increases the initial installation cost.

APPLICATIONS

The integration of chatbots with social media platforms like Telegram, Facebook Messenger, LINE, WeChat, and WhatsApp makes it easier for businesses to provide 24/7 customer service to the user.

- For the travel industry, such customer service chatbot proved to be a boon. Apart from the benefit of booking and scheduling flights, they help to integrate additional services via social media platforms. Services like Airbnb, Uber can be integrated for the enhancement of customer experience.
- One of the most useful applications of these bots is in the healthcare department. An average patient spends 30 minutes trying to get to the right
- service in a local hospital. But deploying conversational chatbots in the healthcare sector can significantly reduce long waits. From registration to coverage and claims, chatbots are in popular demand in this industry.
- The use of chatbots on any appliance website is a very common thing today. Earlier, chatbots would transfer any function related query onto an agent. But with growing technology, that is also eliminated. The bot will then give accurate results without any human interaction.

CONCLUSION

In conclusion, we have created a chatbot which not only engages the customer with small talk but will also provide the user with an accurate and smart solution. The response will be fast and the customer will be satisfied with the service.

By integrating all the above-mentioned services, a smarter and efficiently working Customer Help Desk has been developed.

FUTURE SCOPE

While most businesses will have a chatbot for customer services, many will develop bots for their internal processing.

One such thing will be Bots handling first stage interviews to find promising candidates. The bots can process resumes to find highly qualified candidates and fasten the otherwise slow recruiting process and managing cost.

Customer service is increasingly an automated one. Many companies are adopting AI tools to build a better way of handling and expanding the service.

Chatbots have a long way to go to realize their full potential. Still, the chatbots will ultimately generate significant future value in both corporate and customer settings.

BIBLIOGRAPHY

https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/

https://nodered.org/

https://github.com/watson-developer-cloud/node-red-la

bs https://www.ibm.com/watson/products-services

https://developer.ibm.com/components/watson-assistant/series/learning-path-watson-assist

ant https://cloud.ibm.com/docs/services/discovery?topic=discovery-getting-started

https://medium.com/@rimaibrahim/node-red-watson-discovery-chatbot-telegram-ce616ddcd0d9

https://developer.ibm.com/technologies/web-development/articles/ws-restful

https://cloud.ibm.com/docs/openwhisk?topic=cloud-functions-getting-started

APPENDIX

Source Code

1. Cloud Functions (cloudfunctions.js)

```
/**
 *
 * @param {object} params
 * @param {string} params.iam_apikey
 * @param {string} params.url
 * @param {string} params.username
 * @param {string} params.password
 * @param {string} params.environment_id
 * @param {string} params.collection_id
 * @param {string}
   params.configuration_id
 * @param {string} params.input
 *
 * @return {object}
 *
 */


const assert = require('assert');
const DiscoveryV1 =
require('watson-developer-cloud/discovery
/v1');



/
*
*
*
 * main() will be run when you invoke
   this action
 *
 *  @param Cloud Functions actions
accept a single parameter, which must
be a JSON object.
 *
 * @return The output of this action,
   which must be a JSON object.
 *
 */
```

```
    function main(params) {
     return new Promise(function (resolve, reject) {


       let discovery;
if (params.iam_apikey){ discovery = new
DiscoveryV1({
                  'iam_apikey': params.iam_apikey,
                  'url': params.url,
                  'version': '2019-03-25'
                });
              }
              else {
                discovery = new DiscoveryV1({
                  'username': params.username,
                  'password': params.password,
                  'url': params.url,
                  'version': '2019-03-25'
                });
              }


              discovery.query({
                'environment_id': params.environment_id,
                'collection_id': params.collection_id,
                'natural_language_query': params.input,
                'passages': true,
                'count': 3,
                'passages_count': 3
              }, function(err, data) {
                if (err) {
                  return reject(err);
                }
                return resolve(data);
              });
            });
          }
```

2. **Watson Assistant Skill (skill-Customer-Care-Sample-Skill.json)**

   The code is too long to paste here, please refer to the GitHub Link.

3. **SDU Dataset (ecobee3_UserGuide.pdf)**

   SDU Dataset is the e cobee3_UserGuide.pdf linked above

4. Node-Red Flow (fl

```
[
  {
    "id": "adc107ca.e7ab58",
    "type": "tab",
    "label": "Flow 1",
    "disabled": false,
    "info": ""
  },
  {
    "id": "298f793d.2bc0d6",
    "type": "ui_tab",
    "z": "",
    "name": "Customer Care Helpdesk",
    "icon": "dashboard",
    "disabled": false,
    "hidden": false
  },
  {
    "id": "f553fce1.2ab25",
    "type": "ui_group",
    "z": "",
    "name": "Assistant",
    "tab": "298f793d.2bc0d6",
    "order": 1,
    "disp": true,
    "width": 8,
    "collapse": false
  },
  {
    "id": "578bdfef.d92ab",
    "type": "ui_base",
    "theme": {
      "name": "theme-light",
      "lightTheme": {
        "default": "#0094CE",
        "baseColor": "#268000",
                    "baseFont":
"-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
        "edited": true,
        "reset": false
      },
      "darkTheme": {
        "default": "#097479",
        "baseColor": "#097479",
```

```
                    "baseFont":
"-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
        "edited": false
      },
      "customTheme": {
        "name": "Untitled Theme 1",
        "default": "#4B7930",
        "baseColor": "#4B7930",
                    "baseFont":
"-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"
      },
      "themeState": {
        "base-color":
        {
          "default": "#0094CE",
          "value": "#268000",
          "edited": true
        },
        "page-titlebar-backgroundColor": {
          "value": "#268000",
          "edited": false
        },
        "page-backgroundColor": {
          "value": "#fafafa",
          "edited": false
        },
        "page-sidebar-backgroundColor": {
          "value": "#000000",
          "edited": false
        },
        "group-textColor": {
          "value": "#3dcc00",
          "edited": false
        },
        "group-borderColor": {
          "value": "#ffffff",
          "edited": false
        },
        "group-backgroundColor": {
          "value": "#ffffff",
          "edited": false
        },
        "widget-textColor": {
          "value": "#111111",
          "edited": false
```

```
        },
        "widget-backgroundColor": {
          "value": "#268000",
          "edited": false
        },
        "widget-borderColor": {
          "value": "#ffffff",
          "edited": false
        },
        "base-font": {
                        "value":  "-apple-system,BlinkMacSystemFont,Segoe
UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"
        }
      },
      "angularTheme": {
        "primary": "indigo",
        "accents": "blue",
        "warn": "red",
        "background": "grey"
      }
    },
    "site": {
      "name": "Helpdesk",
      "hideToolbar": "false",
      "allowSwipe": "false",
      "lockMenu": "false",
      "allowTempTheme": "true",
      "dateFormat": "DD/MM/YYYY",
      "sizes": {
        "sx": 48,
        "sy": 48,
        "gx": 6,
        "gy": 6,
        "cx": 6,
        "cy": 6,
        "px": 0,
        "py": 0
      }
    }
  },
  {
    "id": "8bb033af.10a71",
    "type": "ui_form",
    "z": "adc107ca.e7ab58",
    "name": "",
    "label": "",
    "group": "f553fce1.2ab25",
    "order": 1,
```

```
  "width": 0,
  "height": 0,
  "options": [
    {
      "label": "Enter your query here",
      "value": "text",
      "type": "text",
      "required": true,
      "rows": null
    }
  ],
  "formValue": {
    "text": ""
  },
  "payload": "",
  "submit": "submit",
  "cancel": "cancel",
  "topic": "",
  "x": 170,
  "y": 240,
  "wires": [
    [
      "40b00757.b43958"
    ]
  ]
},
{
  "id": "40b00757.b43958",
  "type": "function",
  "z": "adc107ca.e7ab58",
  "name": "Function 1",
  "func": "msg.payload = msg.payload.text;\nreturn msg;",
  "outputs": 1,
  "noerr": 0,
  "x": 350,
  "y": 240,
  "wires": [
    [
      "76f70b6.28918f4",
      "e2c238eb.574e48"
    ]
  ]
},
{
  "id": "83a199fe.1b6568",
  "type": "function",
```

```
    "z": "adc107ca.e7ab58",
    "name": "Function 2",
    "func": "msg.payload = msg.payload.output.text[0];\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 730,
    "y": 220,
    "wires": [
      [
        "17660cf0.c27da3"
      ]
    ]
  },
  {
    "id": "76f70b6.28918f4",
    "type": "watson-conversation-v1",
    "z": "adc107ca.e7ab58",
    "name": "Watson",
    "workspaceid": "817fca58-7a80-4ae5-a9a5-a45edb82216a",
    "multiuser": false,
    "context": true,
    "empty-payload": false,
                                                          "service-endpoint":
"https://api.eu-gb.assistant.watson.cloud.ibm.com/instances/c36ac805-5bcc-
489b-bbbf-a127d64fa181",
    "timeout": "",
    "optout-learning": false,
    "x": 520,
    "y": 160,
    "wires": [
      [
        "746a3840.fa3498",
        "83a199fe.1b6568"
      ]
    ]
  },
  {
    "id": "e2c238eb.574e48",
    "type": "ui_text",
    "z": "adc107ca.e7ab58",
    "group": "f553fce1.2ab25",
    "order": 2,
    "width": 8,
    "height": "3",
    "name": "",
    "label": "You:  ",
```

```
    "format": "{{msg.payload}}",
    "layout": "row-left",
    "x": 510,
    "y": 320,
    "wires": [

    ]
  },
  {
    "id": "746a3840.fa3498",
    "type": "debug",
    "z": "adc107ca.e7ab58",
    "name": "",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "x": 730,
    "y": 80,
    "wires": [

    ]
  },
  {
    "id": "17660cf0.c27da3",
    "type": "ui_text",
    "z": "adc107ca.e7ab58",
    "group": "f553fce1.2ab25",
    "order": 5,
    "width": 0,
    "height": 0,
    "name": "",
    "label": "Assistant: ",
    "format": "{{msg.payload}}",
    "layout": "col-center",
    "x": 900,
    "y": 320,
    "wires": [

    ]
  }
]
```

THANK YOU