Project Scope Document

Project Title:
Intelligent Customer Help Desk with Smart Document Understanding - SB13620

Project Summery :
We are interested to build a customer care chatbot which can answer simple questions like store location , opening hours, direction and many realated things. When a question falls outside of the scope of t he pre-determined question set, the option is typically to tell the c ustomer 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 a bout the operation of a device, the application shall pass the q uestion onto Watson Discovery Service, which has been pre-loaded w ith the device's owner's manual

Project Requirement:
In order to develope this project, we will be using multiple IBM cloud services like:
Watson Assistant, Discovery, Node Red, Cloud Functions,etc.

Functional Requirement:
IBM Cloud

Technical Requirement:
AI,ML,Watson

Software Requirement:
Watson Assistant,Watson Discovery.

Project Deliverables:
Smartinternz Internship

Project Team:
Vivek Auti

Project Schedule:
project duration : 19 days
start date : 05/05/2020
End date :

Project Report

Title :
Intelligent Customer Help Desk with Smart Document Understanding - SB13620

Name :
Vivek Ramesh Auti

Email ID :
vrauti13@gmail.com

Category :
Artificial Inteliigence

CONTENTS

1. INTRODUCTION

1.1 Overview

We will be able to write an application that leverages multiple Watson AI services like Discovery, Assistant, Cloud Function and Node Red. By the end of the project we will learn best practices of combining Watson Services, and how they can build interactive retrieval system with discovery + assistant.

- Project Requirements : Python, IBM Watson,IBM cloud.
- Functional Requirement : IBM Cloud
- Technical Requirements : Watson, AI, ML
- Software Requirements : Watson Discovery, Watson Assistant.
- Project Deliverbles : Smart Internz Internship
- Project duration : 1 Month

1.2 Purpose

The chatbot is designed to answer simple questions like store locations and hours, directions and maybe even making appointments. When any user input (question) falls out of the scope of the predetermined question set of chatbot, it typically tells us to rephrase our sentence or mention that this question is not valid. In t his project, there will be another option. If the customer question is a bout the operation of a device, the application shall pass the q uestion onto Watson Discovery Service, which has been pre-loaded w ith the device's owner's manual. So now, instead of "Would you like t o speak to a customer representative?" we can return relevant s ections of the owner's manual to help solve our customers' problems.To take it a step further, the project shall use the Smart D ocument 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

2. LITERATURE SURVEY

2.1 Existing Problem :

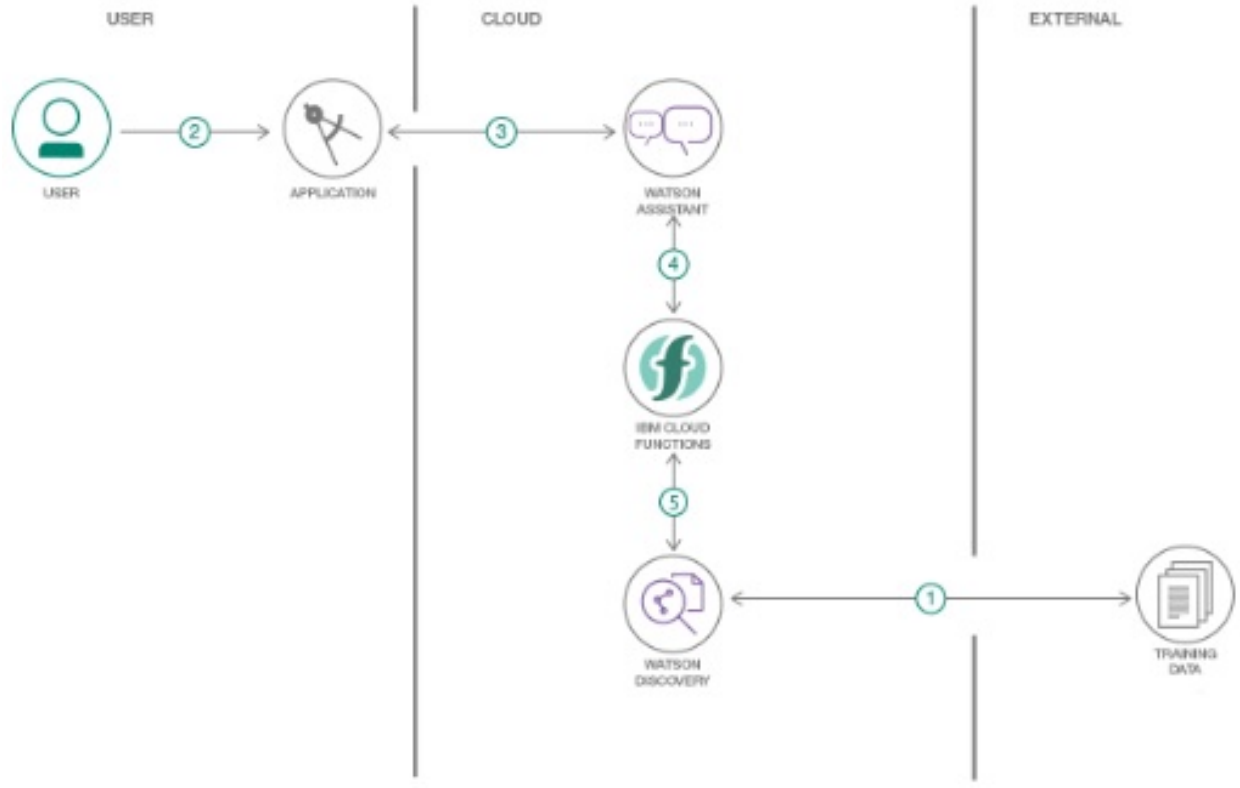
a computer program designed to simulate conversation with human users, especially over the Internet. "chatbots often treat conversations like they're a game of tennis: talk, reply, talk, reply".
Generally chatbot are loaded with a certain set of questions that is more like if and else flow, the question or the user input which lies out of the scope of the chatbot is not answered and rather a message like “Try again after rephrasing” & “I am unable to understand, Please Rephrase” are displayed and it directs the user to the customer agent or the representative

2.2 Proposed Solution :

to achieve solution to the existing problem, we include a Smart chatbot so that it can answer the queries of the customer effectively and efficiently. For the above-mentioned problem, we have to put a virtual agent in chatbot so it can understand the queries that are posted by customers. The virtual agent should be trained from some insight based on company backgrounds, working hours, store locations and product related information. Watson Discovery will help to improve this situations.

3. Theoretical Analysis

3.1 Block / Flow Diagram :



1. The document is understood smartly using Watson Assistant.
2. User Interacts with server via app UI. The frontend UI is a chatbot that engages the user in a conversation.
3. Dialog between the user and backend server is established via Watson Assistant dialog Skill.
4. The dialog skill will reply for the basic questions asked by the user.
5. If the consumer or user asks about a product operations, a search query is passed to predefined IBM cloud function Action.
6. The cloud Function will query the watson Discovery service and return the results.

3.2 Hardware / Software designing :

- Create IBM Cloud Services
- configure Watson Discovery
- Create IBM cloud Functions action
- Configure Watson Assistant
- Add necessary skills to Assistant
- Create flow and configure node
- Deploy and run Node Red app.

4. Experimental Investigation

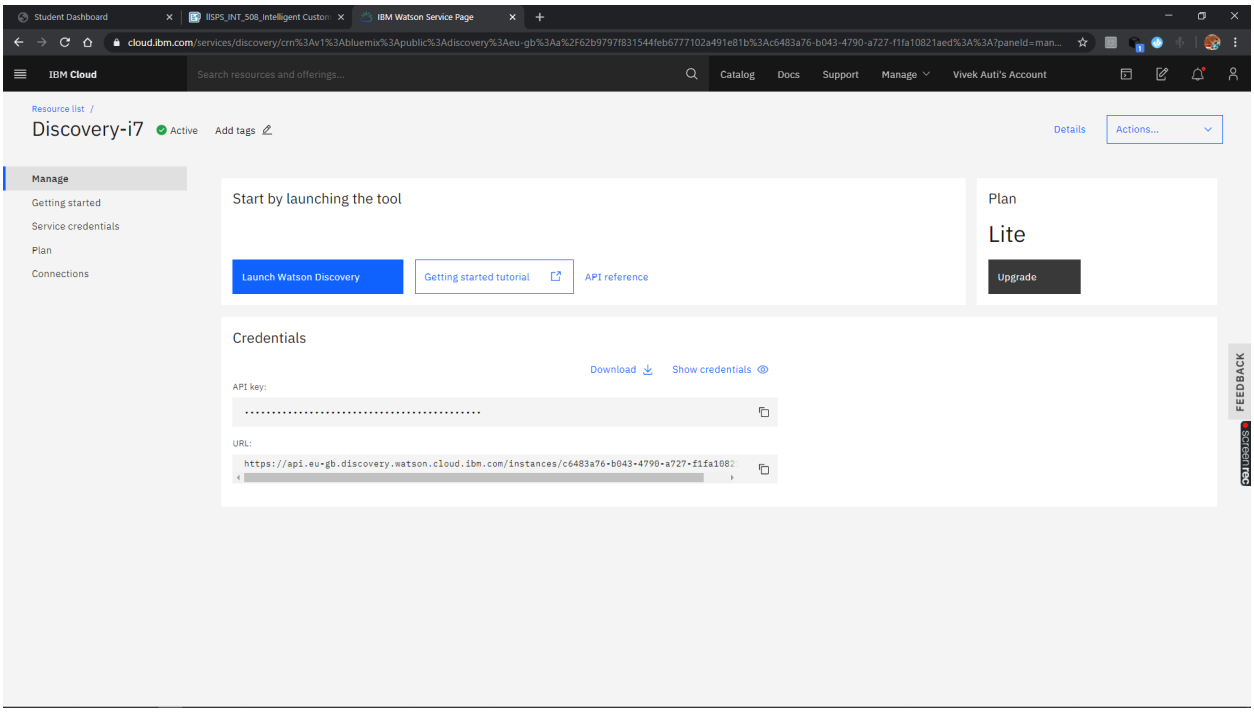
1. Create IBM Cloud services

- Watson Discovery
- Watson Assistant
- Node-Red Application.

2. Configure Watson Discovery

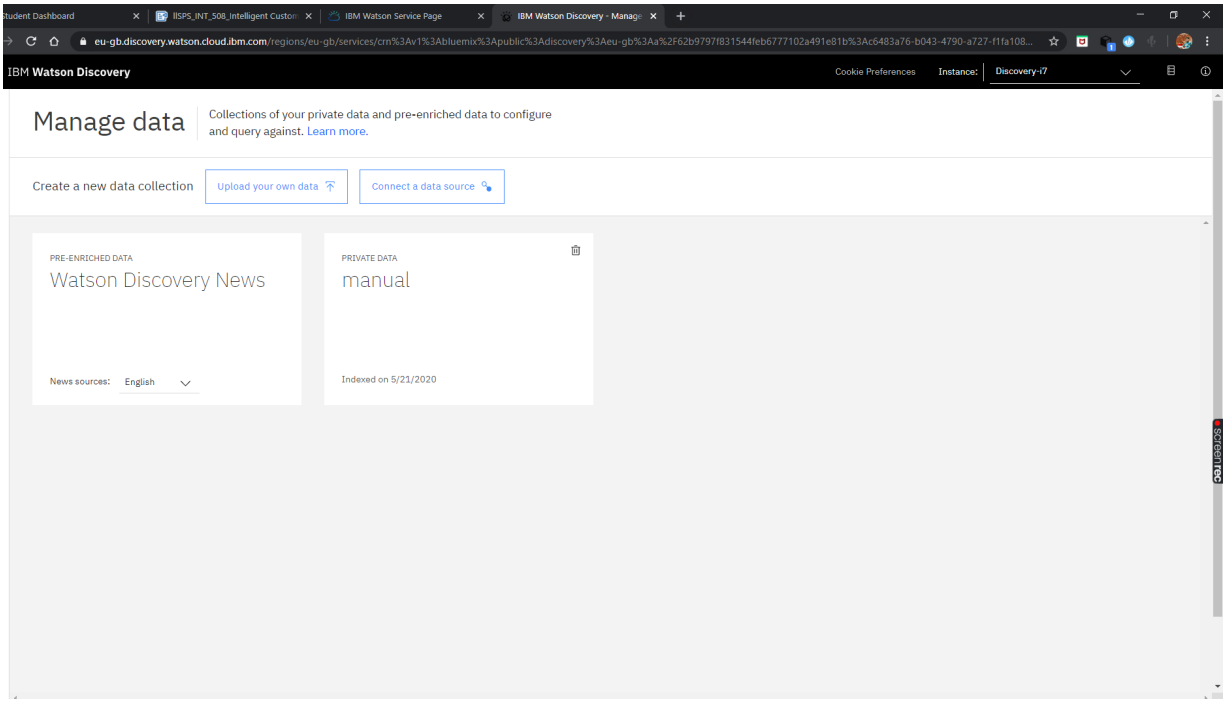
I. LAUNCH THE TOOLING :

On the Manage page, click Launch Watson Discovery. If you're prompted to log in to the tooling, provide your IBM Cloud credentials.

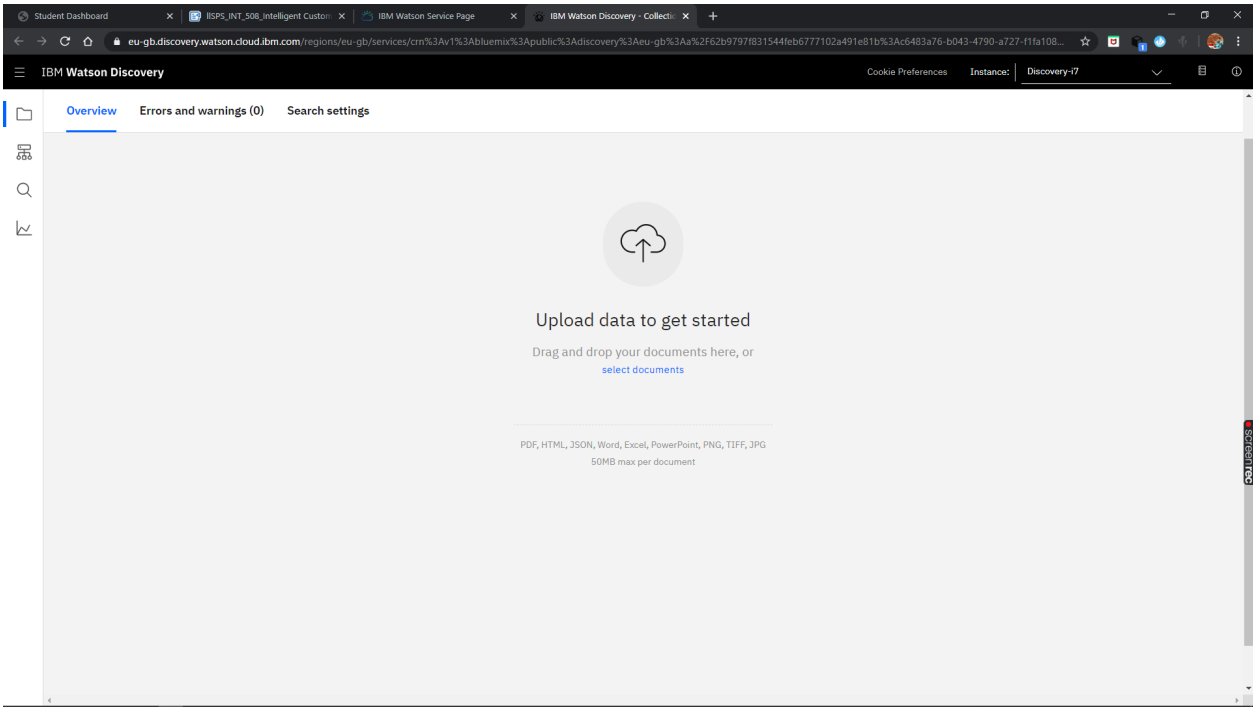


II. CREATE A COLLECTION

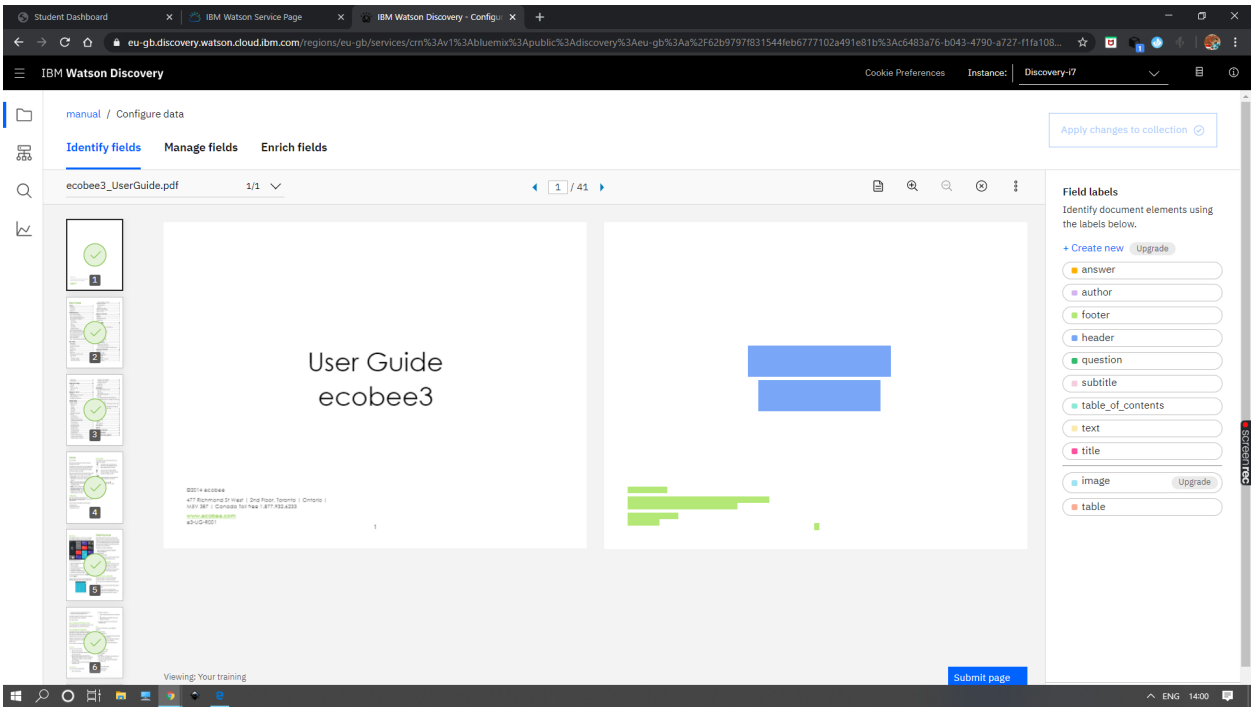
A collection is a set of your documents.The public, pre-enriched Discovery News data collection is also available for your use. It is ready to query, and you can begin to create queries on it immediately.Based on your need, you can create your own collection.



III. UPLOAD THE DOCUMENT TO YOUR COLLECTION



IV. ANNOTATE YOUR DOCUMENT

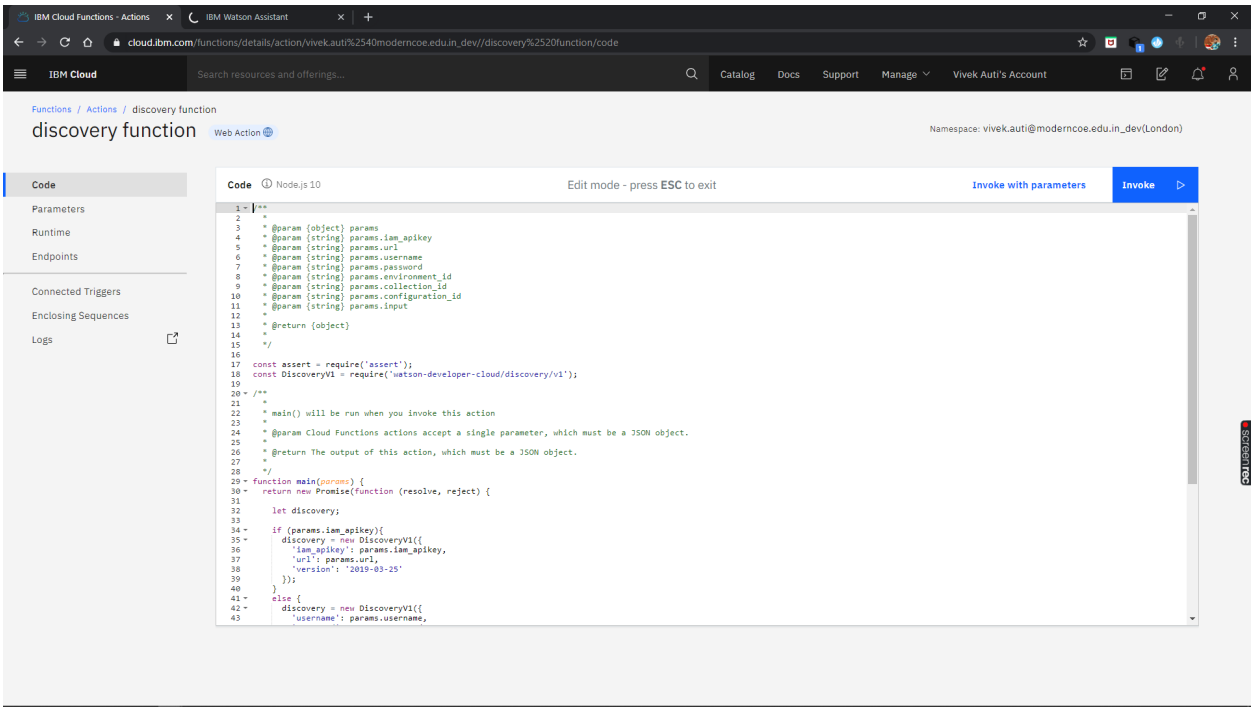


The Watson Explorer Installation Guide is displayed and ready for annotation on the Identify fields tab. All available fields (answer, author, footer, header, question, subtitle, table_of_contents, text, and title) are displayed in the Field labels list on the right.

V. LET'S QUERY

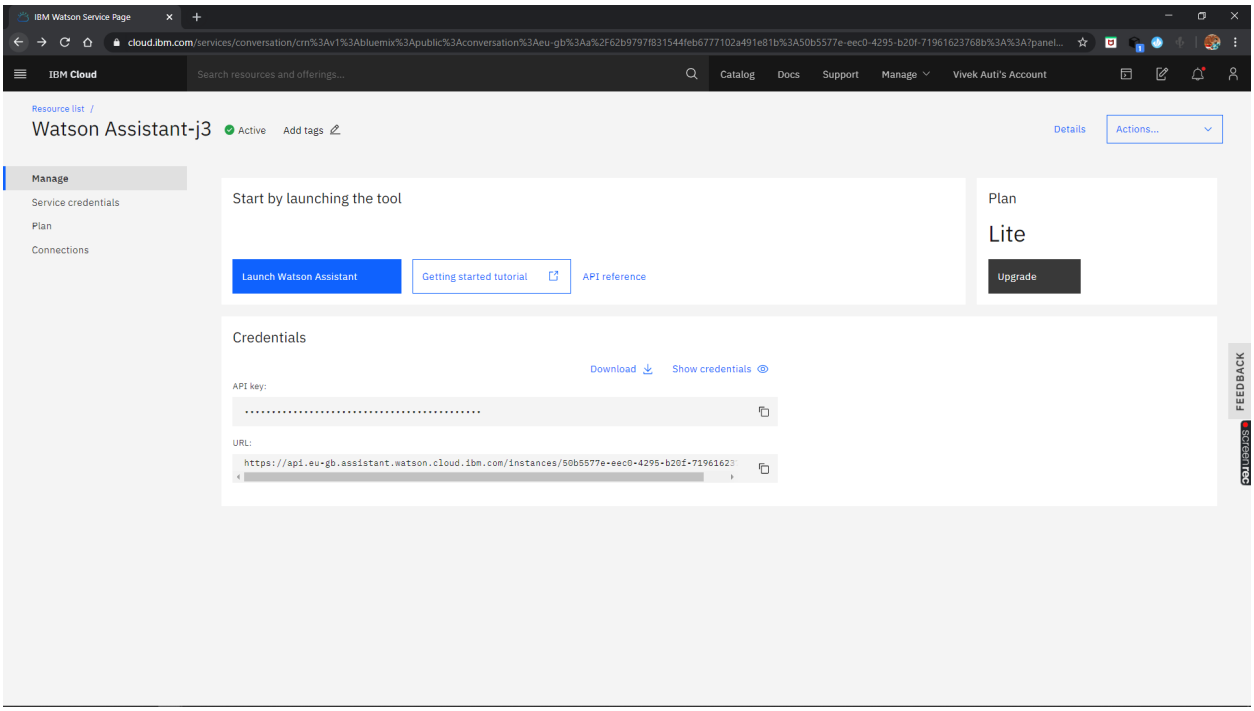
3. Create the IBM Cloud Function Action :

It is used to link the discovery with assistant, so that our queries can be answered by the discovery. After selecting the action from the IBM catalog, we have to click on the action tab as shown on the left menu. Here we made the Information function. Then we can post the code which will help us to link the discovery.

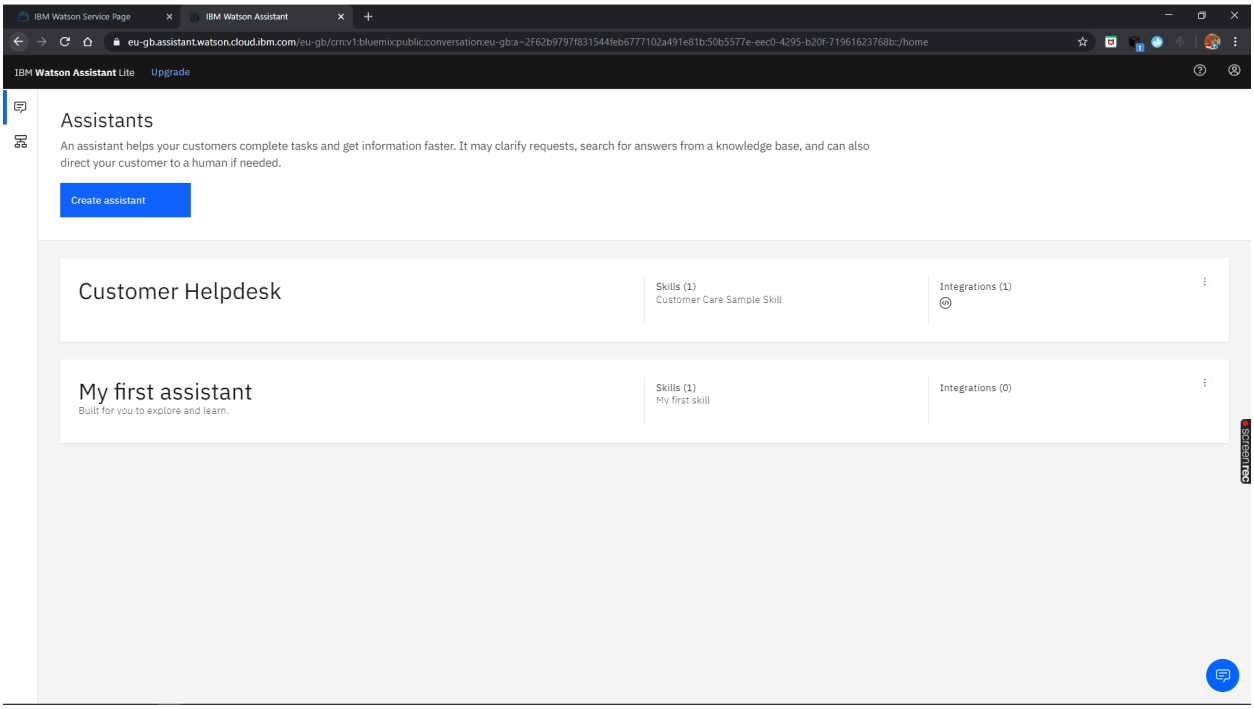


4. Create Watson Assistant

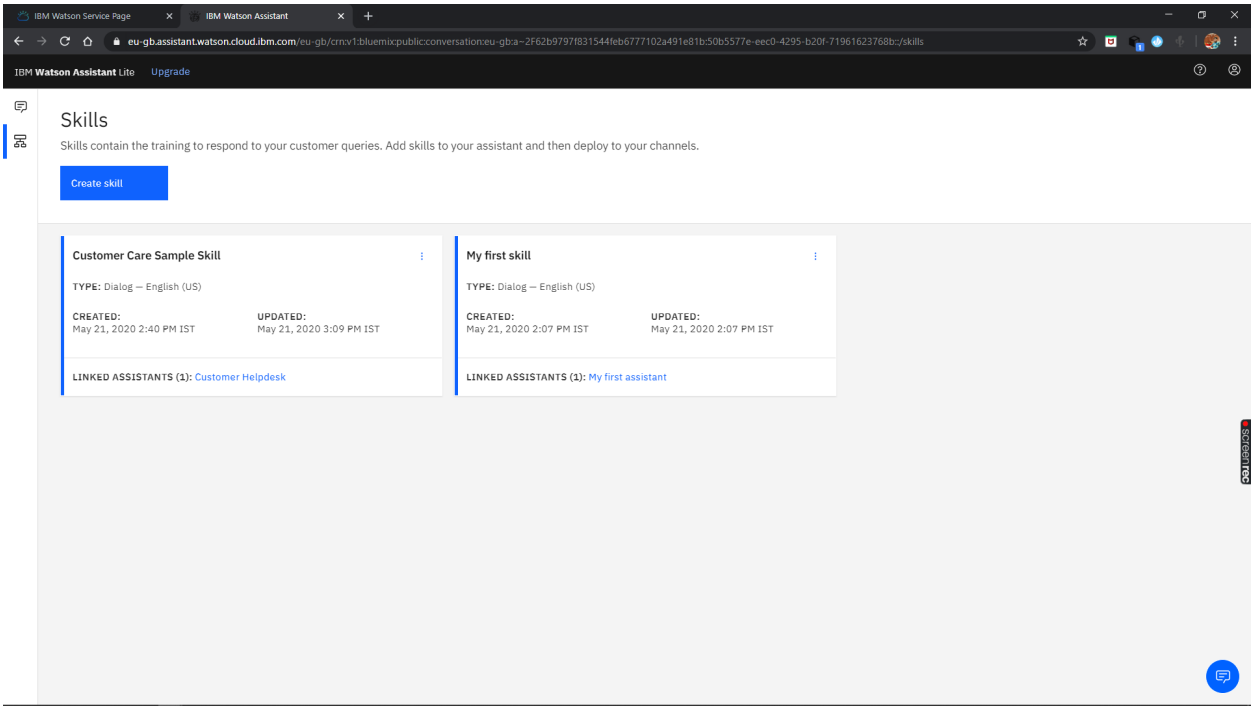
I. OPEN WATSON ASSISTANT



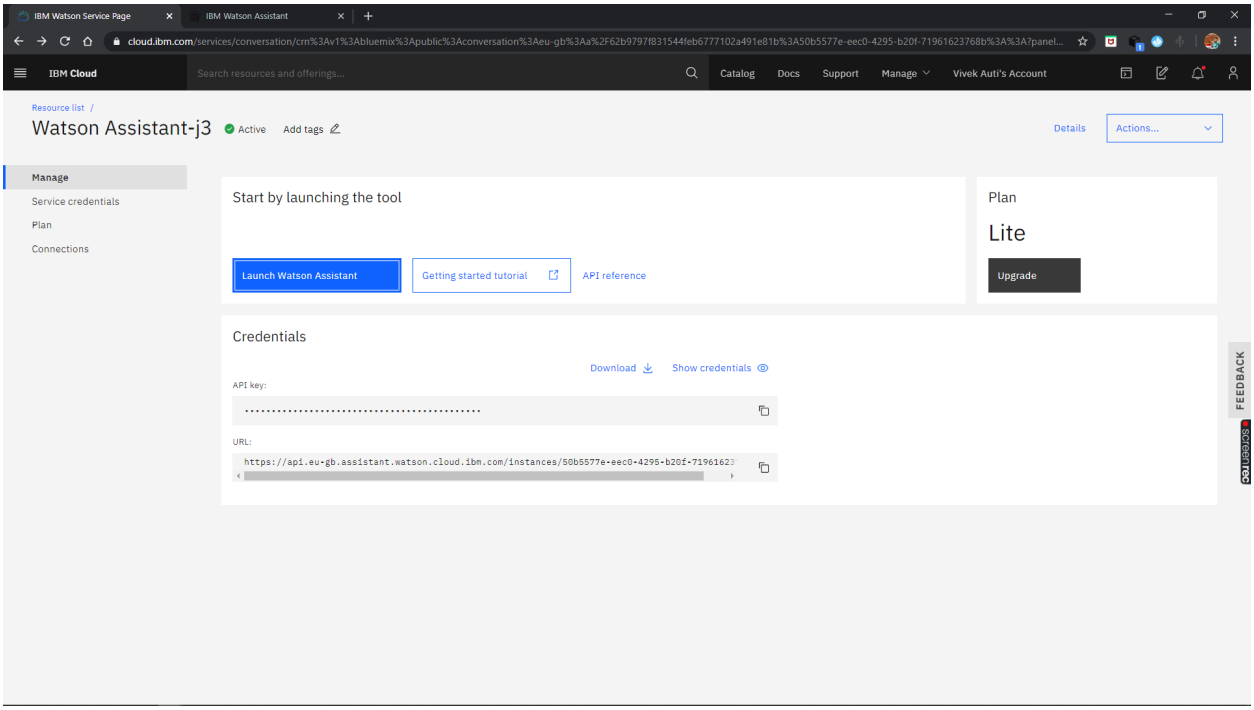
II. CREATE AN ASSISTANT



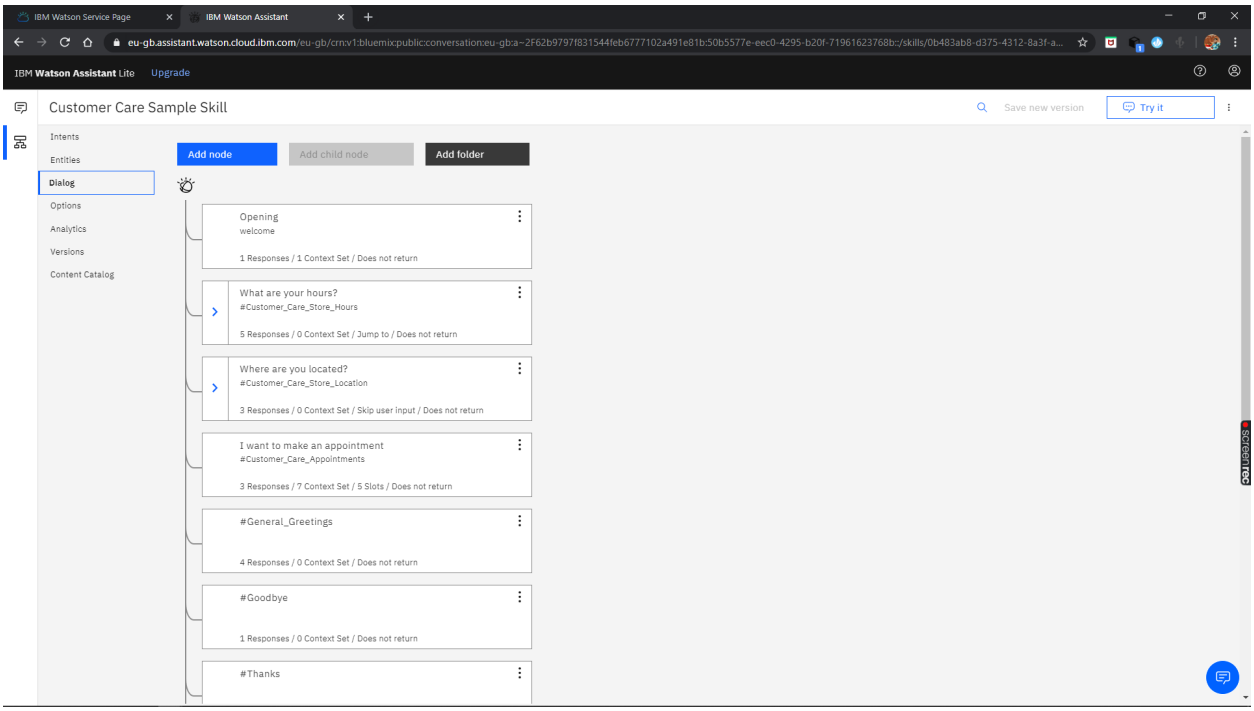
III. CREATE A DIALOG SKILL



IV. ADD INTENTS FROM A CONTENT CATALOG



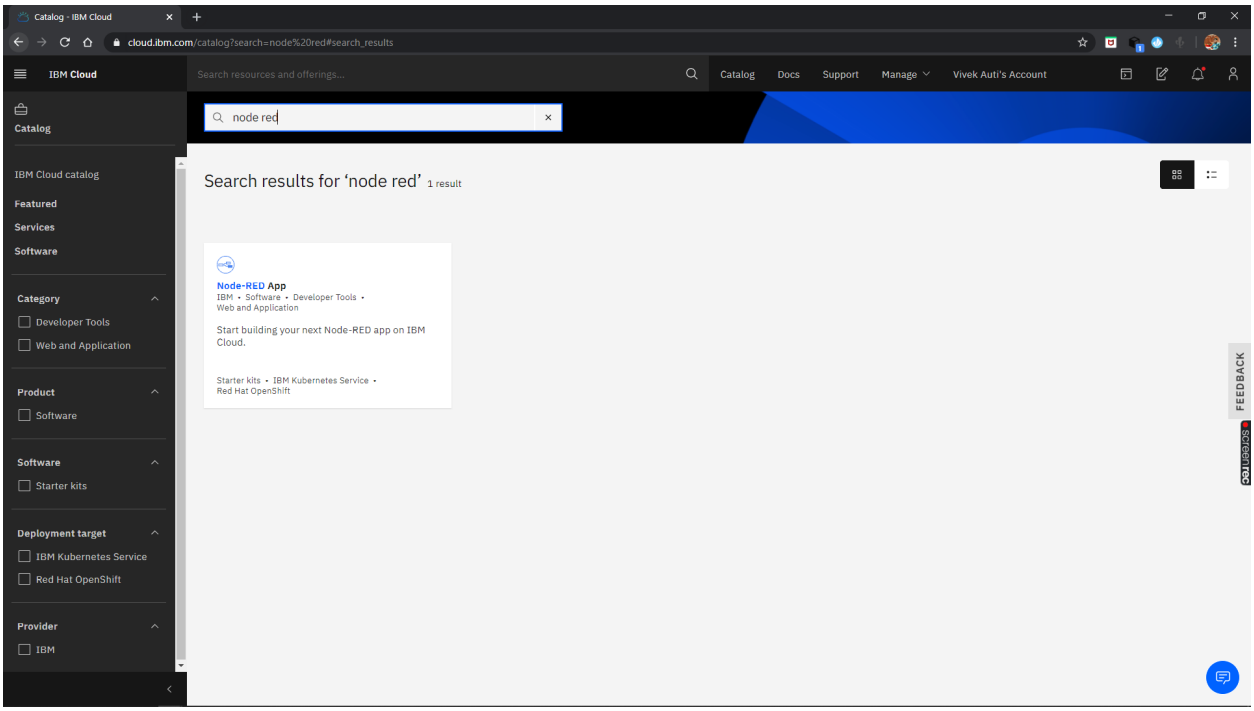
V. BUILD A DIALOG



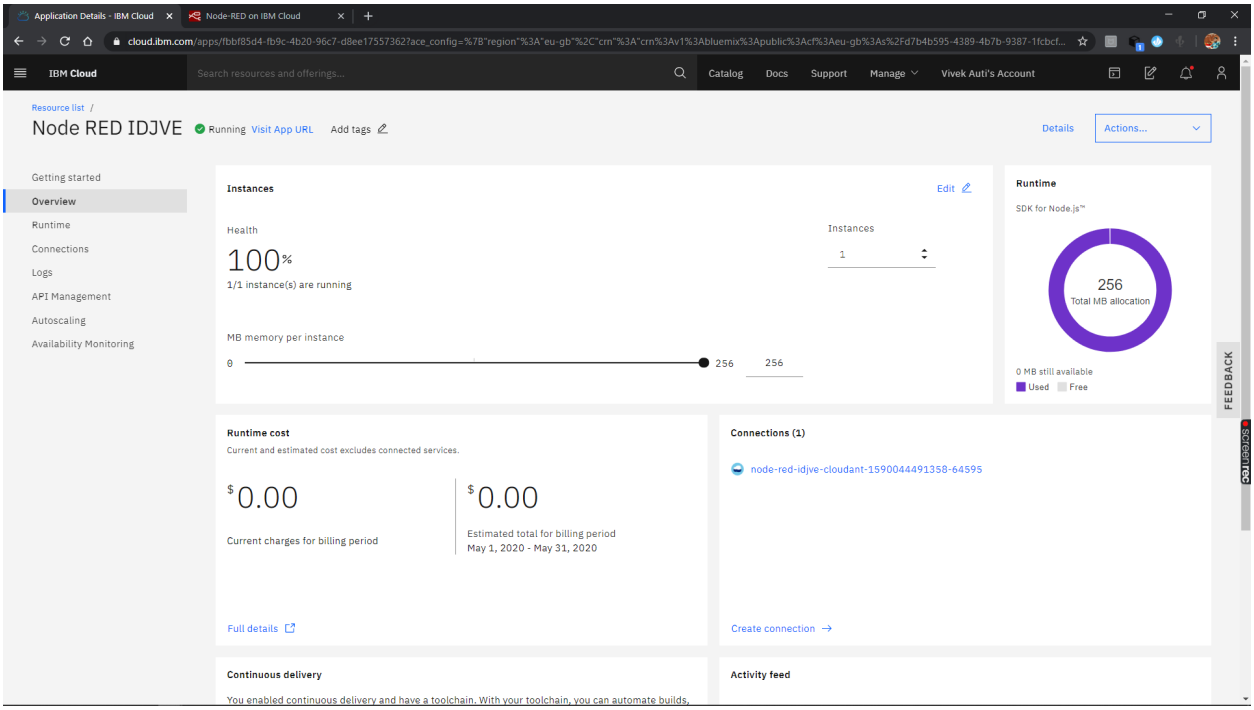
VI. INTEGRATE THE ASSISTANT

5.Build Node Red App

I. FIND THE NODE-RED STARTER IN THE IBM CLOUD CATALOG

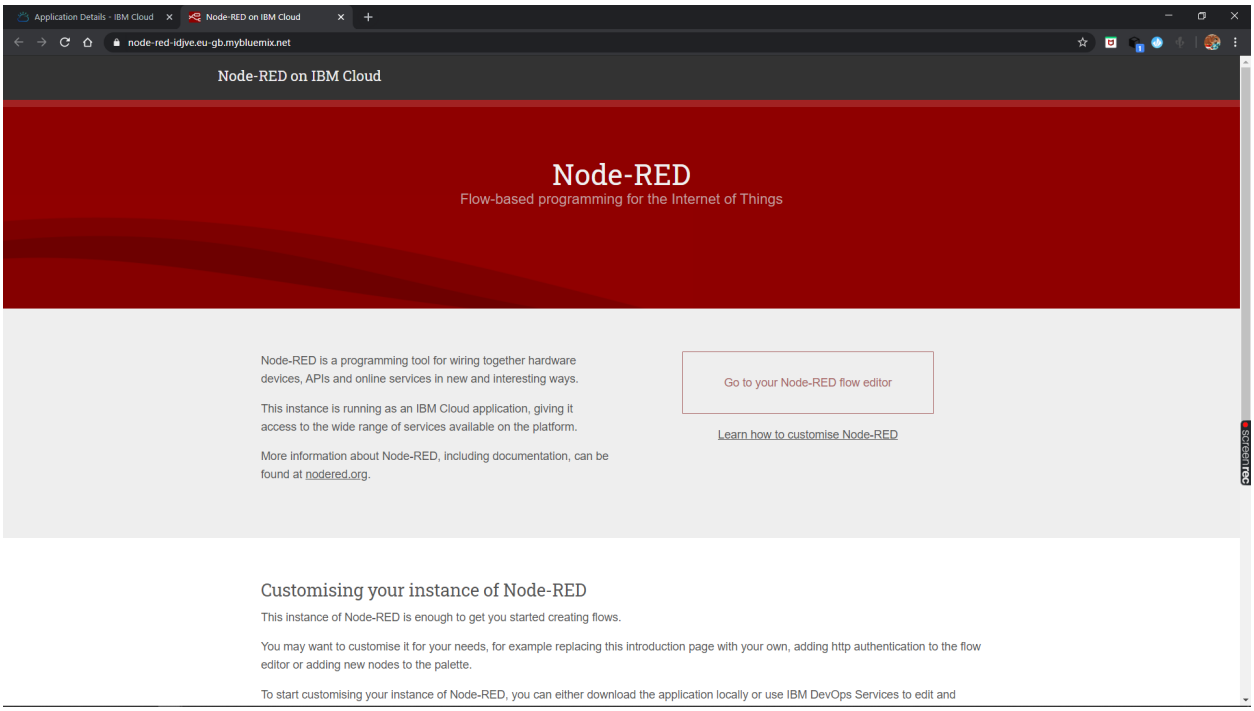


II. CONFIGURE YOUR APPLICATION



III. ENABLE THE CONTINUOUS DELIVERY FEATURE

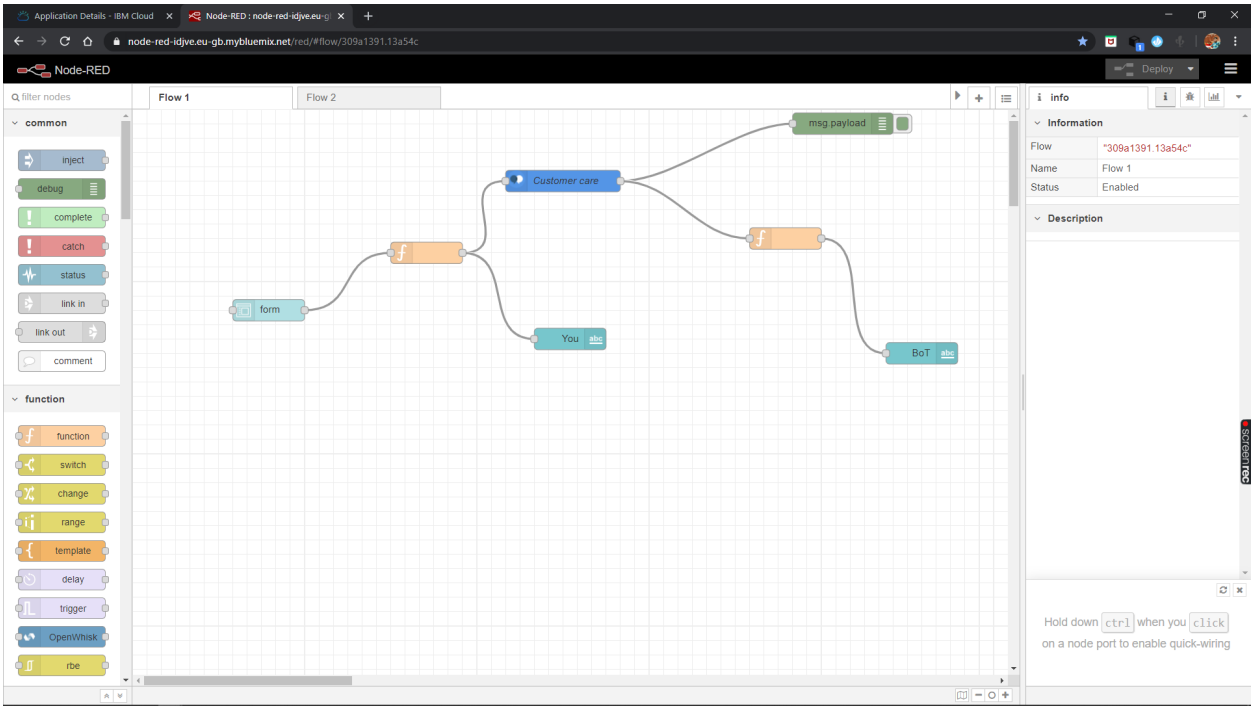
IV. OPEN THE NODE-RED APPLICATION



V. CONFIGURE YOUR NODE-RED APPLICATION

VI. ADD EXTRA NODES TO YOUR NODE-RED PALETTE

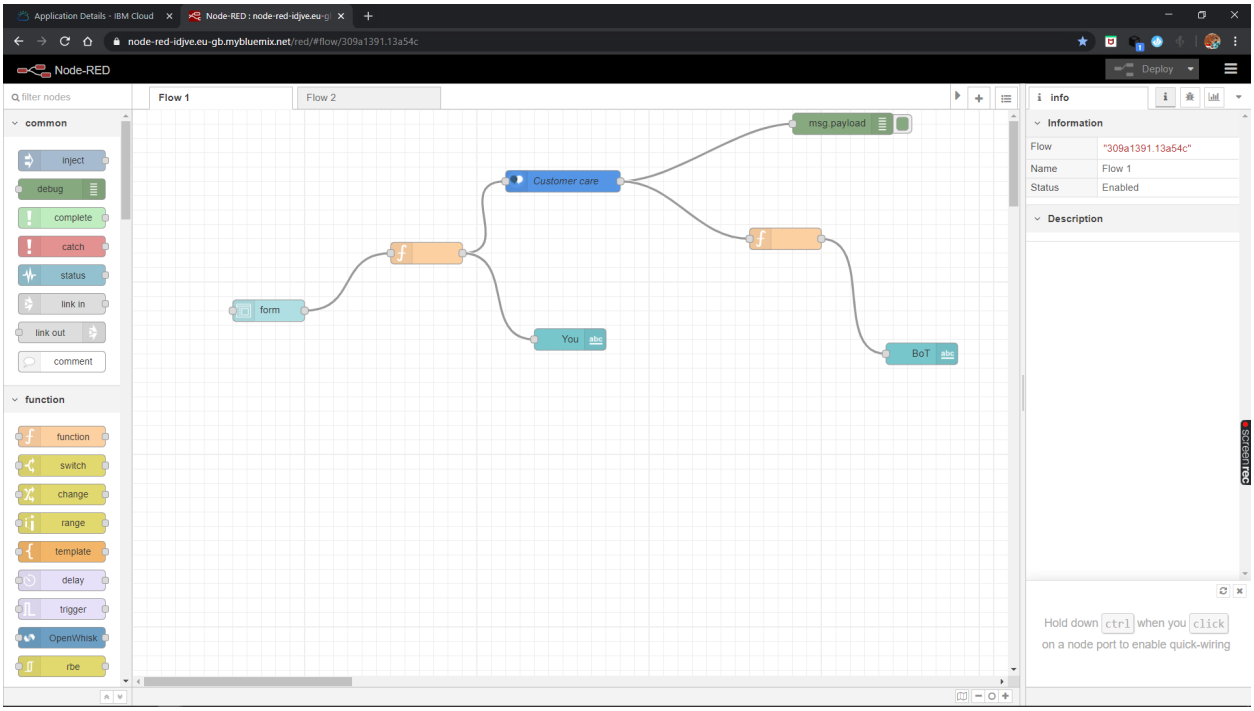
VII. INTEGRATION OF WATSON ASSISTANT IN NODE-RED



5. FLOWCHART

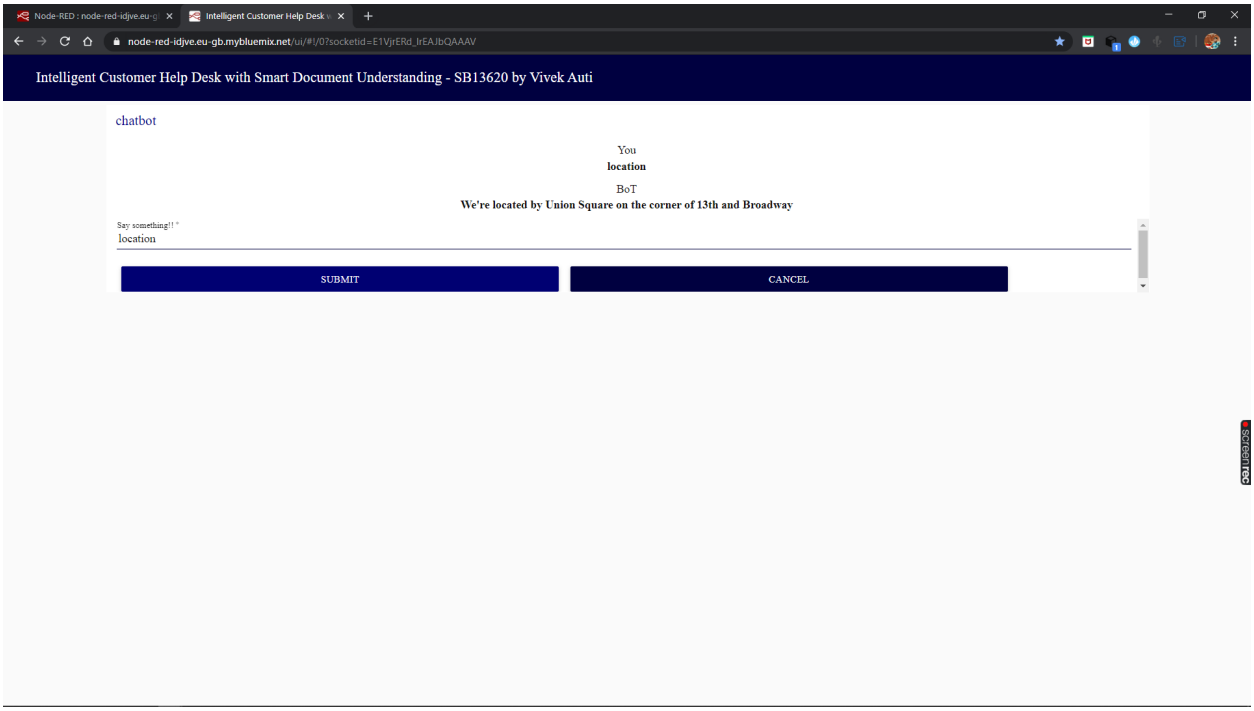
First, go to manage pallete and install dashboard. Now, Create the flow with the help of following nodes:

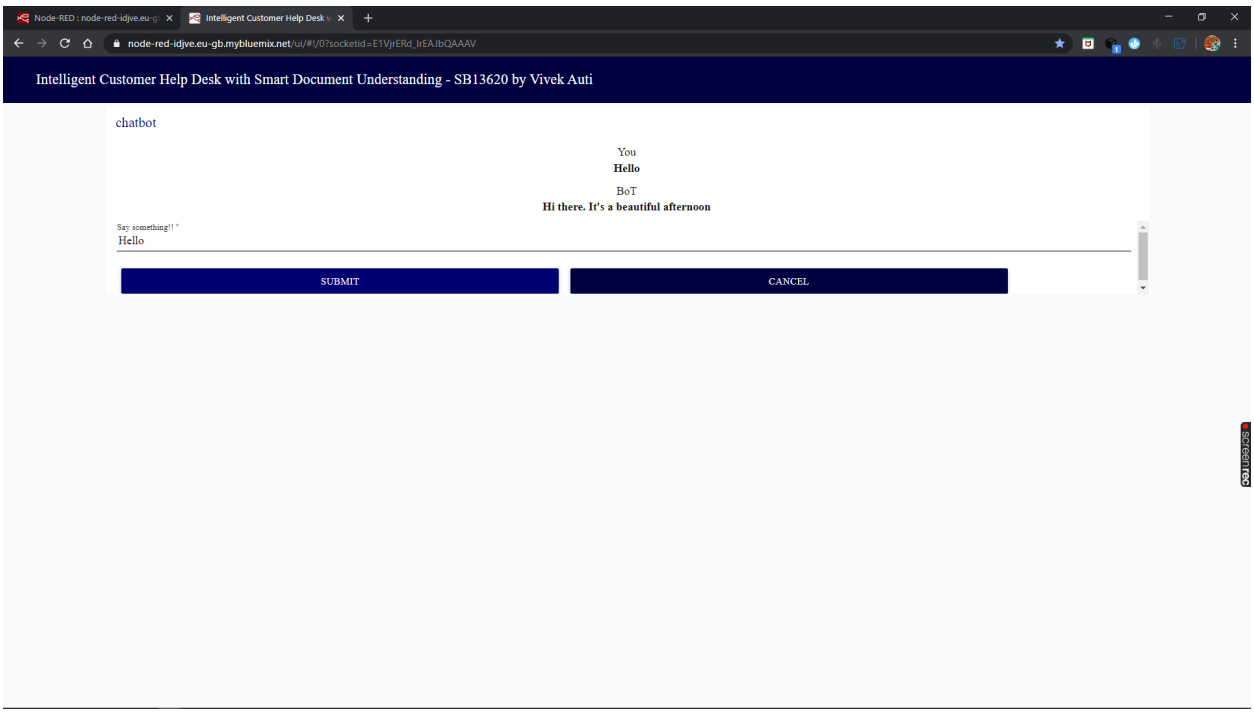
- Inject
- Assistant
- Debug
- Function Ui_Form
- Ui



6. RESULTS

● Finally our Node-RED dash board integrates all the components and displayed in the Dashboard UI by typing
URL- https://node-red-idjve.eu-gb.mybluemix.net/ui/





7. ADVANTAGES & DISADVANTAGES

Advandtages :

● 24X7 Support
● Reduce operational and service expense
● Get a new age platform to wow your customers
● Increase engagement with customers and touchpoints
● to rectifiy simple and general human queries .
● Reduces man power
● Cost efficient
● No need to divert calls to customer agent and customer agent can look on other works
● Multiply reach, increase breadth and depth of engagement
● Rich analytics and customer interaction

Disadvantages :

● Lack Emotions
● Some times chatbot can mislead customers
● They have been designed to handle first-level questions only. They may not be able to solve complex queries.
● Chatbots require ongoing review, maintenance, and optimization in terms of their knowledge base and the way they are supposed to communicate with our customers.

8. Applications

1. It can be deployed in popular social media applications like Facebook, Slack and Telegram.
2. Chatbot can be deployed at any website to clear the basic doubts of the customer.

9. Conclusion

By following the above-mentioned steps, we can create a basic chatbot which can help us to answer the basic questions of the customer or user related to location of the office, working hours and the information about the product. We successfully create the intelligent helpdesk smart chatbot using Watson Assistant, Watson Cloud Function, Watson Discovery and Node-Red.

10. Future Scope

We can import the pre-built node-red flow and can improve our UI, moreover we can make a data base and use it to show the recent chats to the customer. We can also improve the results of discovery by enriching it with more fields and doing the Smart Data Annotation more accurately. We can get the premium version to increase the scope of our chatbot in terms of the calla and requests.
We can also include Watson text to audio and Speech to text services to access the chatbot handsfree. These are few of the future scopes which are possible.

11. Appendix

cloud function code :

```
/**
 *
 * @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);
    });
  });
}
```

12. References

1. https://www.ibm.com/cloud/architecture/tutorials/cognitive_discovery
2. https://cloud.ibm.com/docs/assistant?topic=assistant-getting-started
3. https://developer.ibm.com/recipes/tutorials/how-to-create-a-watson-chatbot-on-nodered