

PROJECT REPORT

Intelligent Customer Help Desk with Smart Document Understanding

Category: Artificial Intelligence Developer

Application ID: SPS_APL_20200003539
Project ID: SPS_PRO_99

Internship at SmartInternz

Mo Hassan Raza
mhr.mhr.hr@gmail.com

1	INTRODUCTION
	1.1 Overview
	1.2 Purpose
2	LITERATURE SURVEY
	2.1 Existing problem
	2.2 Proposed solution
3	THEORITICAL ANALYSIS
	3.1 Block diagram
	3.2 Hardware / Software designing
4	EXPERIMENTAL INVESTIGATIONS
5	FLOWCHART
6	RESULT
7	ADVANTAGES & DISADVANTAGES
8	APPLICATIONS
9	CONCLUSION
10	FUTURE SCOPE
11	BIBILOGRAPHY
	APPENDIX
	A. Source code
	B. Reference

INTRODUCTION

1.1 Overview

We use the typical customer care chatbot experience but instead of relying on predefined responses, our dialog will provide a hook that can call out to other IBM Watson services for additional sources of information. In our case, it will be an owner's manual that has been uploaded into Watson Discovery.

1.2 Purpose

The purpose is to Enhance the customer helpdesks with Smart Document Understanding using webhooks in Watson Assistant.

2. LITERATURE SURVEY

2.1 Existing Problem

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.

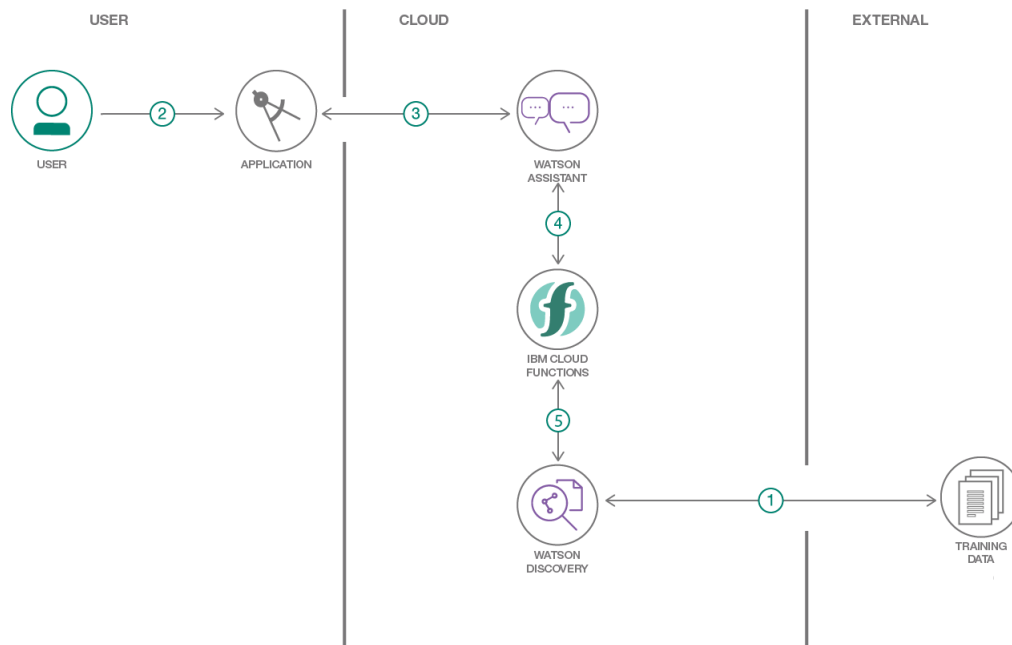
2.2 Proposed solution

In this project, 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.

3. THEORITICAL ANALYSIS

3.1 Block Diagram



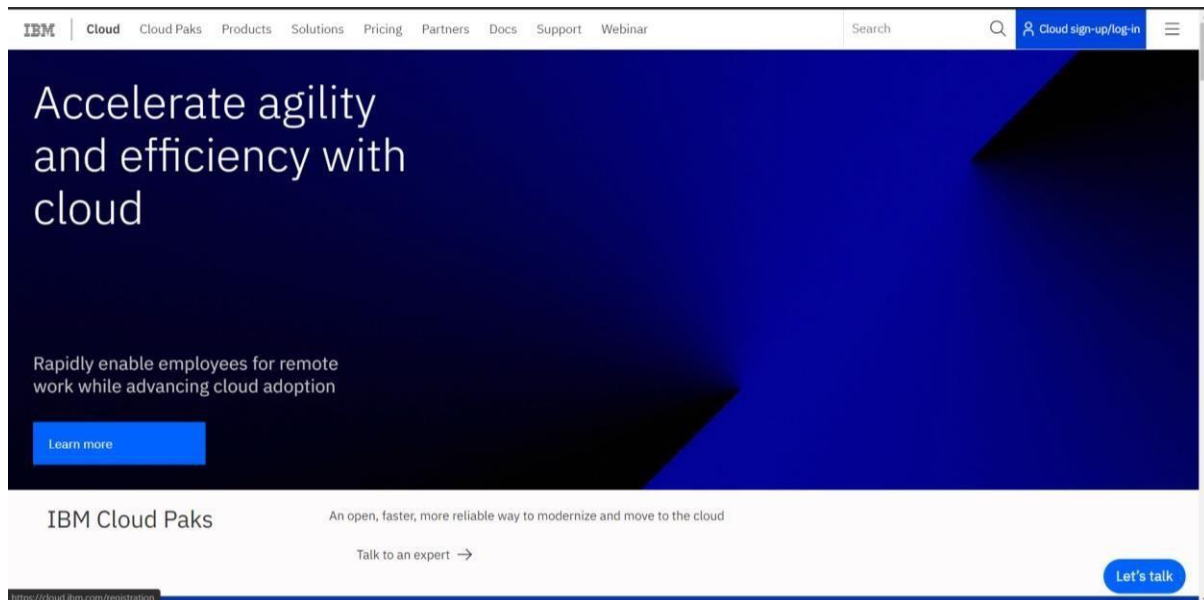
3.2 Hardware / Software designing

1. Create IBM Cloud Services
2. Configure Watson Discovery
3. Create IBM Cloud Functions action
4. Configure Watson Assistant
5. Build Node-RED Flow to Integrate All Services
6. Configure the nodes and Build A Web Dashboard in Node-RED
7. Deploy and Run the application

4. EXPERIMENTAL INVESTIGATIONS

1. Create IBM Cloud Services

To Create IBM Cloud, go to <https://www.ibm.com/cloud>

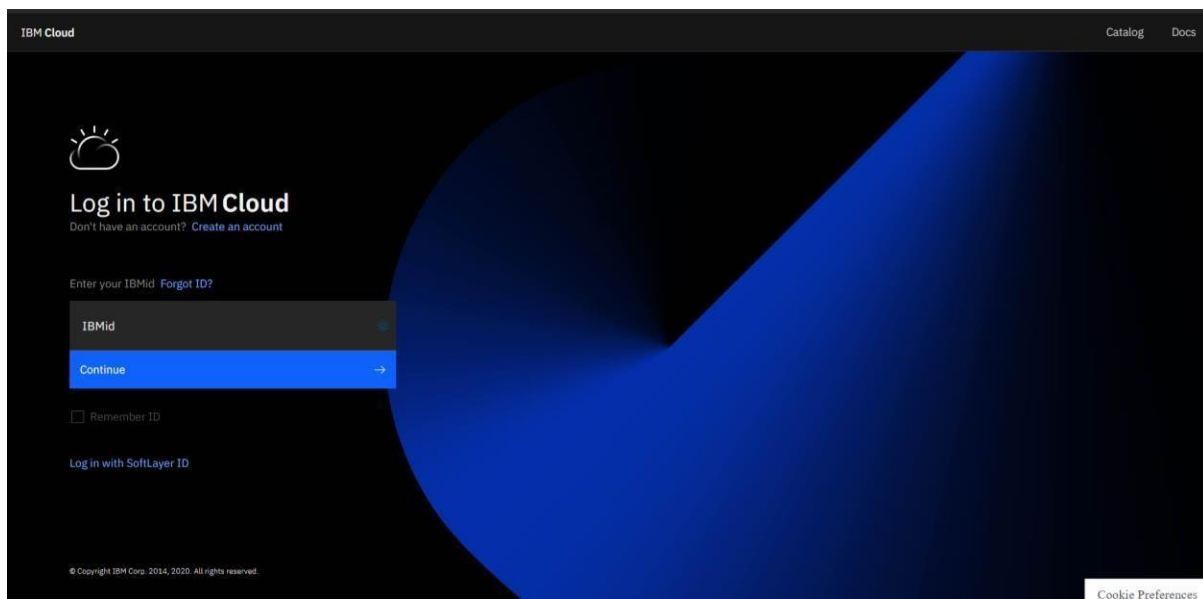


Click on [Cloud sign-up/login](#) to sign up or login to IBM Cloud.

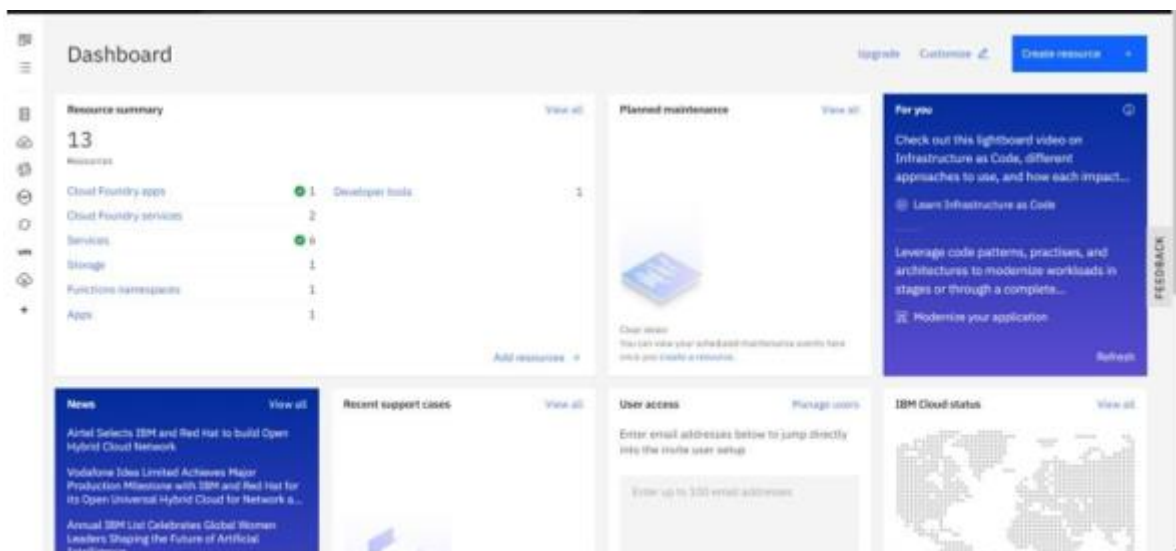
The screenshot shows the IBM Cloud account creation page. On the left, there's a sidebar with the IBM Cloud logo and links for 'Already have an IBM Cloud account? Log in', 'Create an account', and a progress indicator for '1. Account information', '2. Verify email', and '3. Personal information'. The main content area has a heading 'Build for free on IBM Cloud' and two sections: 'Develop for free, no credit card required' (Apps, AI, analytics, and more. Build with 40+ Lite plan services at no cost to you - ever.) and '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.). The 'Create an account' form includes fields for 'Email' and 'Password', a 'Next' button, and a 'Cookie Preferences' link on the right.

For Cloud Sign-Up: Follow the steps on the screen and fill in all the required details to create a new cloud account

For Cloud Log In, click on Already have an IBM Cloud account? [Log in](#) and Log in to your cloud account.

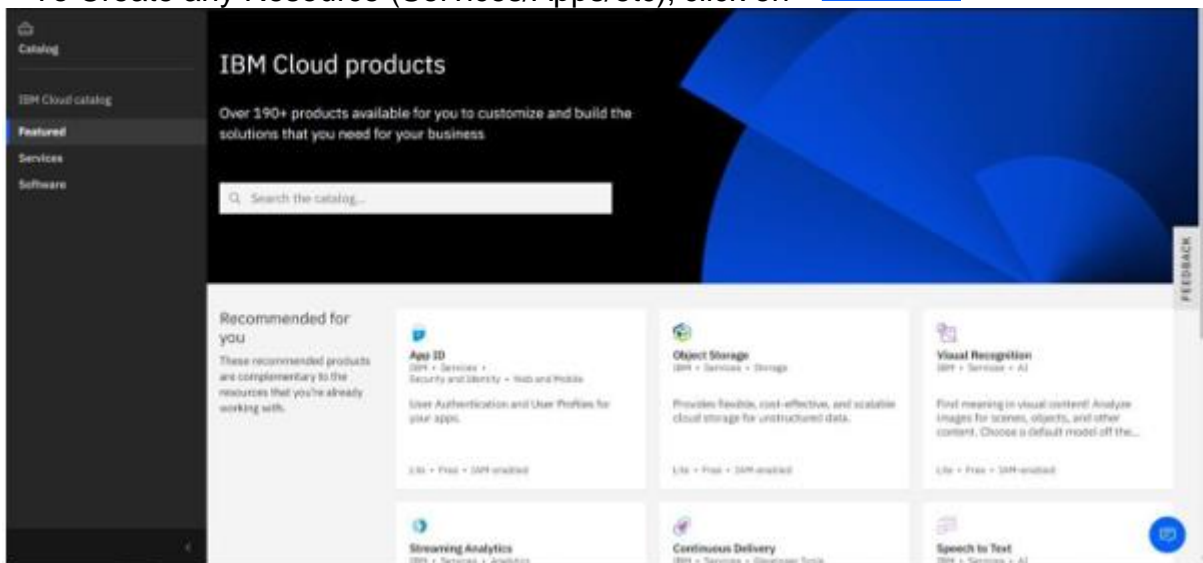


After Logging in, you can see the IBM Cloud Dashboard.



To Create any Resource (Services/Apps/etc), click on

Create resource +



Using the search box, we can find the service we want.

For this project, we need to Create the following services:

1. Watson Discovery
2. Watson Assistant

1. To create a Watson Discovery Service, search for Discovery in the search box



Click on

The screenshot shows the "Create" page for the IBM Watson Discovery service. The interface includes a "Create" button and an "About" button. Below these, there's a "Select a region" section with a dropdown menu showing "London". The "Select a pricing plan" section displays a table with pricing plans. The "Lite" plan is selected, showing features like "0-1,000 documents per month" and "200 query queries per month". A "Create" button is visible at the bottom right of the main content area. A sidebar on the right contains a "Summary" section with details like "Region: London", "Plan: Lite", and "Service name: Discovery-16". It also includes a "FEEDBACK" button and a "View terms" link.

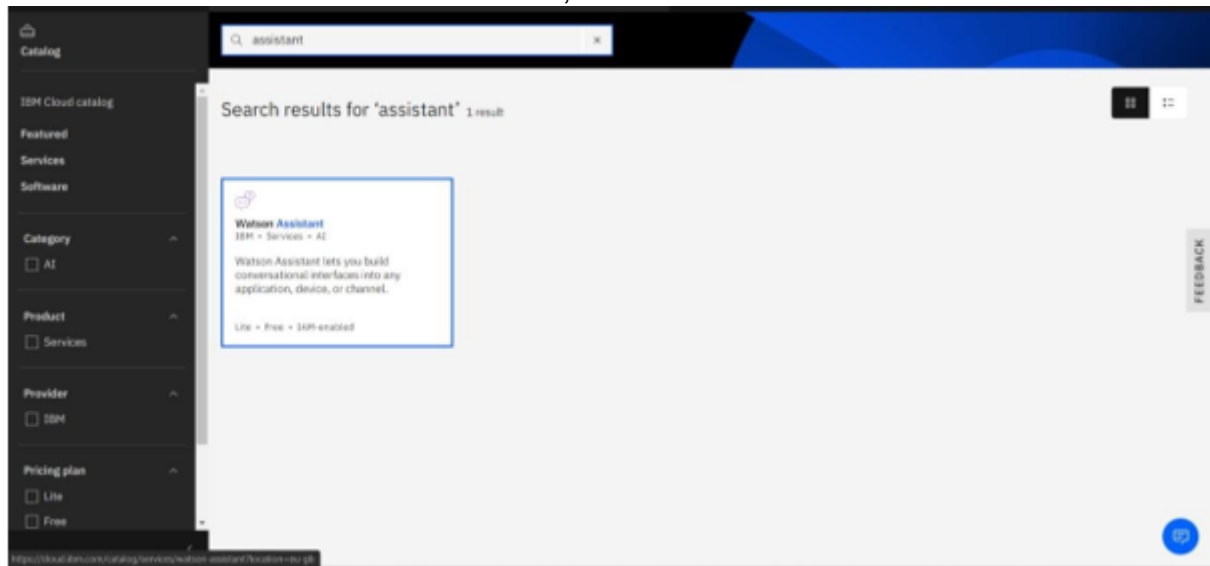
Plan	Features	Pricing
Lite	0-1,000 documents per month 200 query queries per month 3 custom models See documentation for plan details	Free

Select a region, select a plan, configure your service (Service name, etc) and click Create.

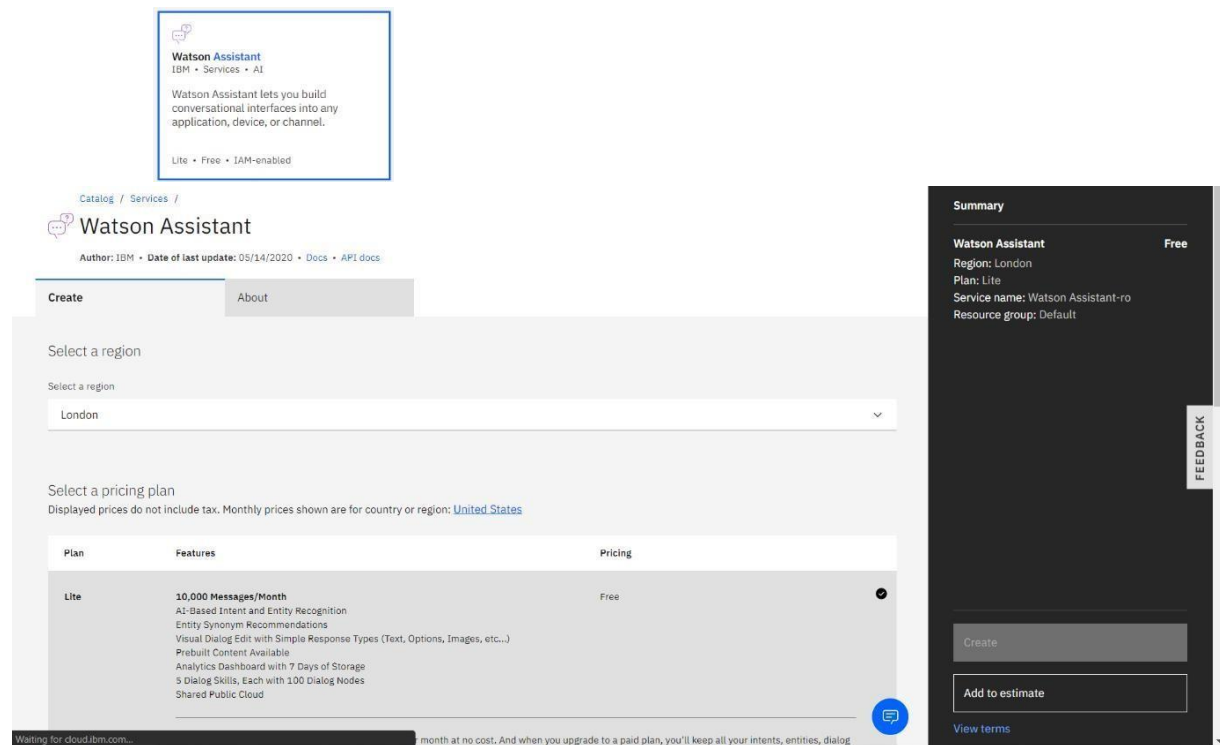
Your Watson Discovery service is created successfully.

(If you are on Lite Plan, you can have only one instance per service)

2. To create a Watson Assistant Service, search for Assistant in the search box



Click on



Select a region, select a plan, configure your service (Service name, etc) and click Create.

Your Watson Assistant service is created successfully.

(If you are on Lite Plan, you can have only one instance per service)

To check whether you have correctly configured the services, go back to the IBM Dashboard and click on View All from the Resource Summary Tab.

Resource summary		View all
13		
Resources		
Cloud Foundry apps	1	Developer tools 1
Cloud Foundry services	2	
Services	6	
Storage	1	
Functions namespaces	1	
Apps	1	
Add resources +		

All of your existing Resource list will be shown here, click on Services to unveil the list of services you have.


Resource list

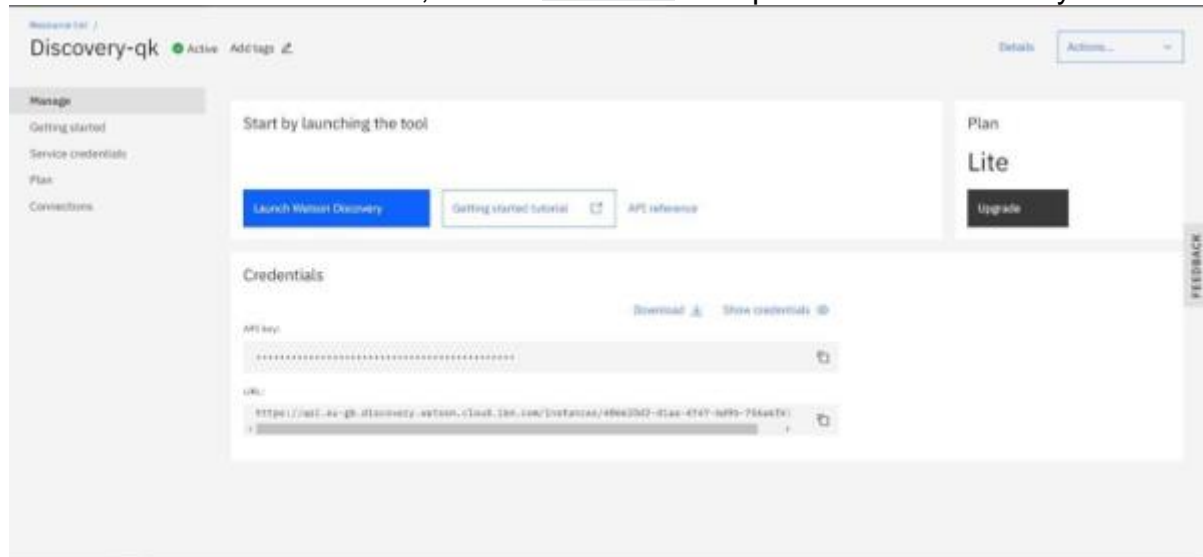
Create resource

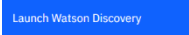
Name	Group	Location	Offering	Status	Tags
Filter by name or IP address...	Filter by group or org...	Filter...	Filter...	Filter...	Filter...
Devices (0)					
VPC infrastructure (0)					
Clusters (0)					
Cloud Foundry apps (1)					
Cloud Foundry services (2)					
Services (6)					
Continuous Delivery	Default	London	Continuous Delivery	Active	—
Discovery-uk	Default	London	Discovery	Active	—
Watson Assistant-61	Default	London	Watson Assistant	Active	—
Watson Studio-4h	Default	London	Watson Studio	Active	—
node-red-qeyad-cloudant-1589268036...	Default	Chennai 01	Cloudant	Active	—
watson-vision-combined-eg	Default	Dallas	Visual Recognition	Active	cpes...
Storage (1)					

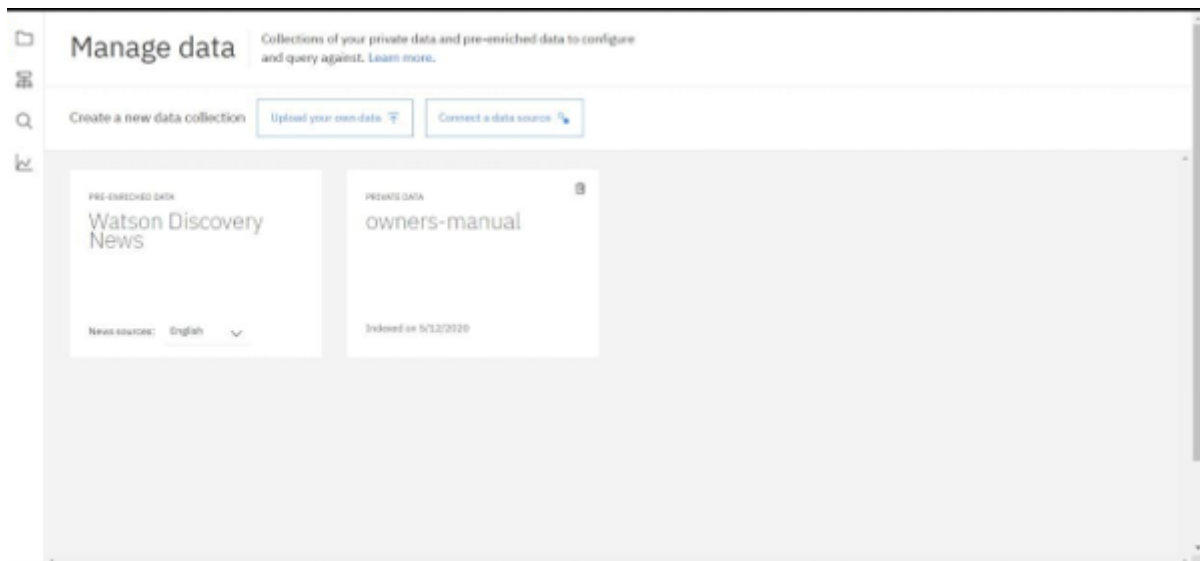
Here we can find that the status of Watson Discovery and Watson Assistant as Active which means we have configured the services correctly.

2. Configure Watson Discovery

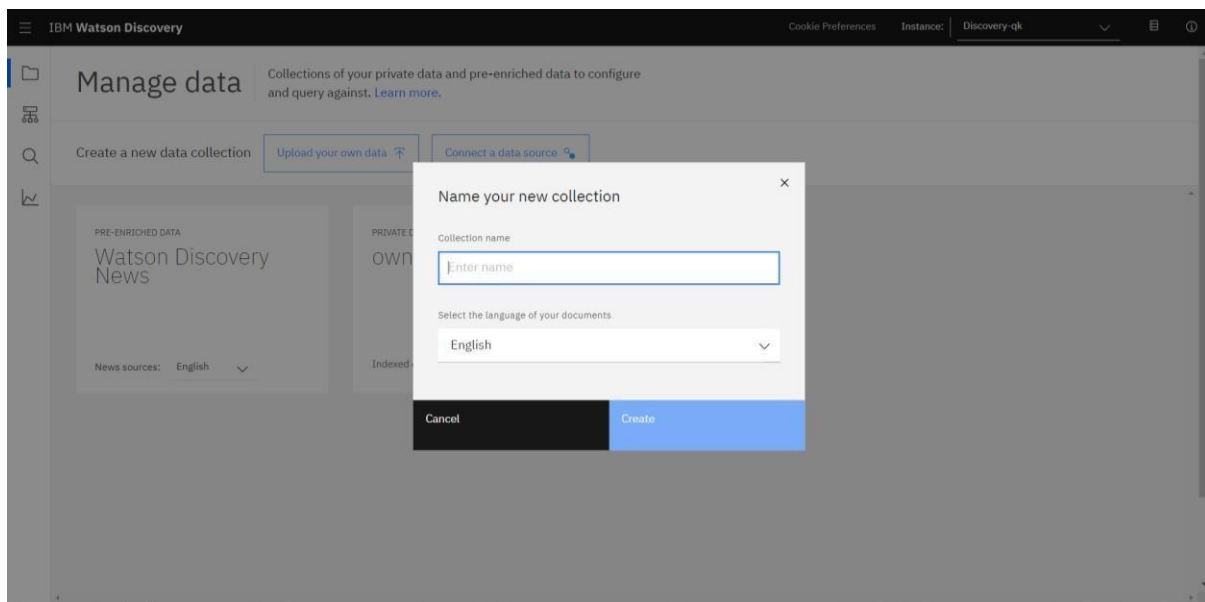
From the resource list screen, click  Discovery-qk to open Watson Discovery service.



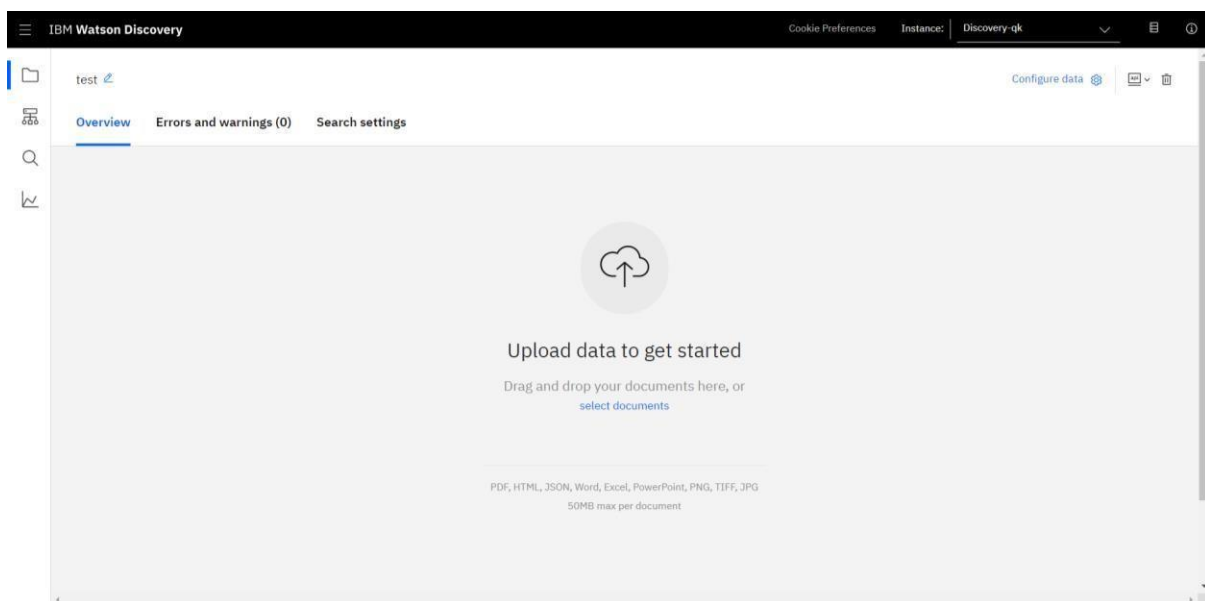
Click on  to launch Watson Discovery Service.

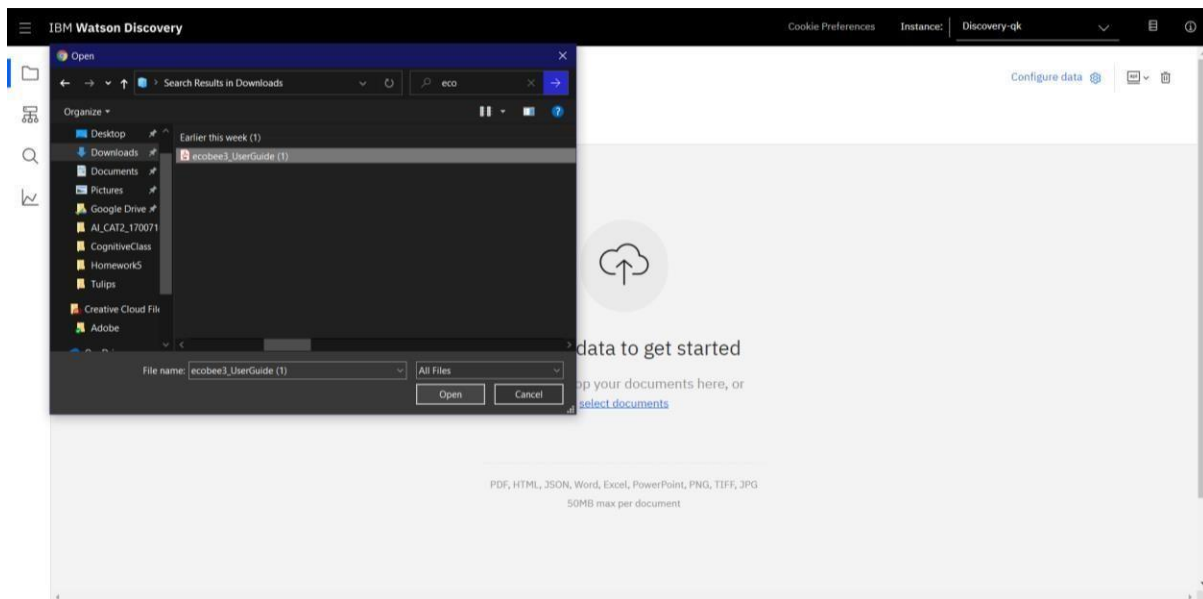


Click on [Upload your own data](#) to create a new data collection.

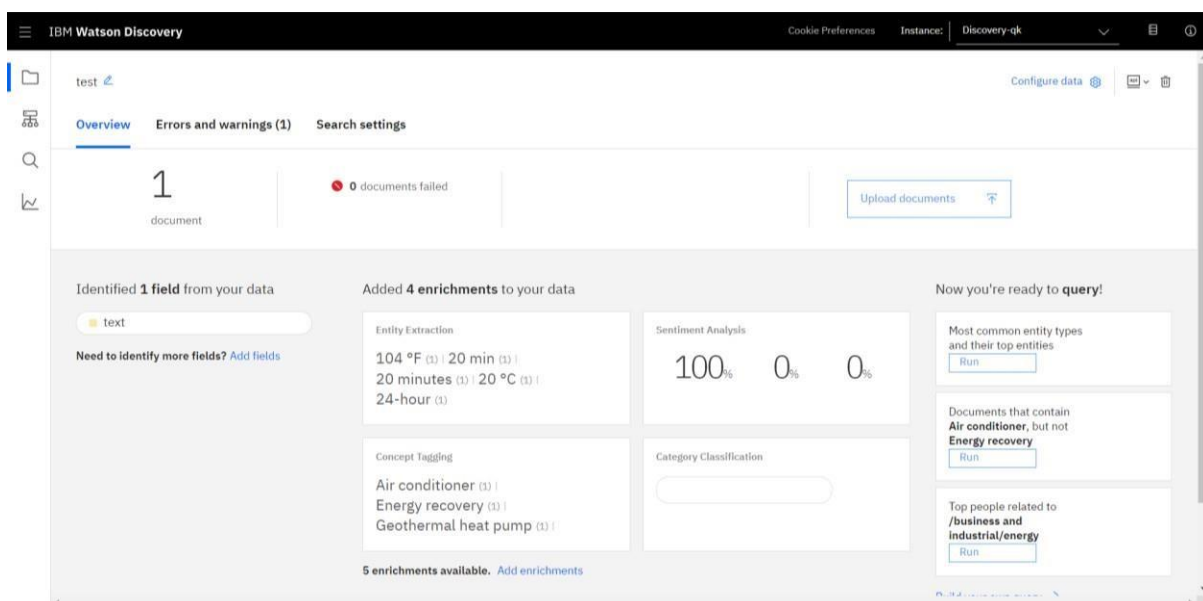


Give the data collection a unique name.





When prompted, select and upload the ecobee3_UserGuide.pdf file located in the data directory of your local repo.



Before proceeding further, let's learn about Smart Document Understanding(SDU)

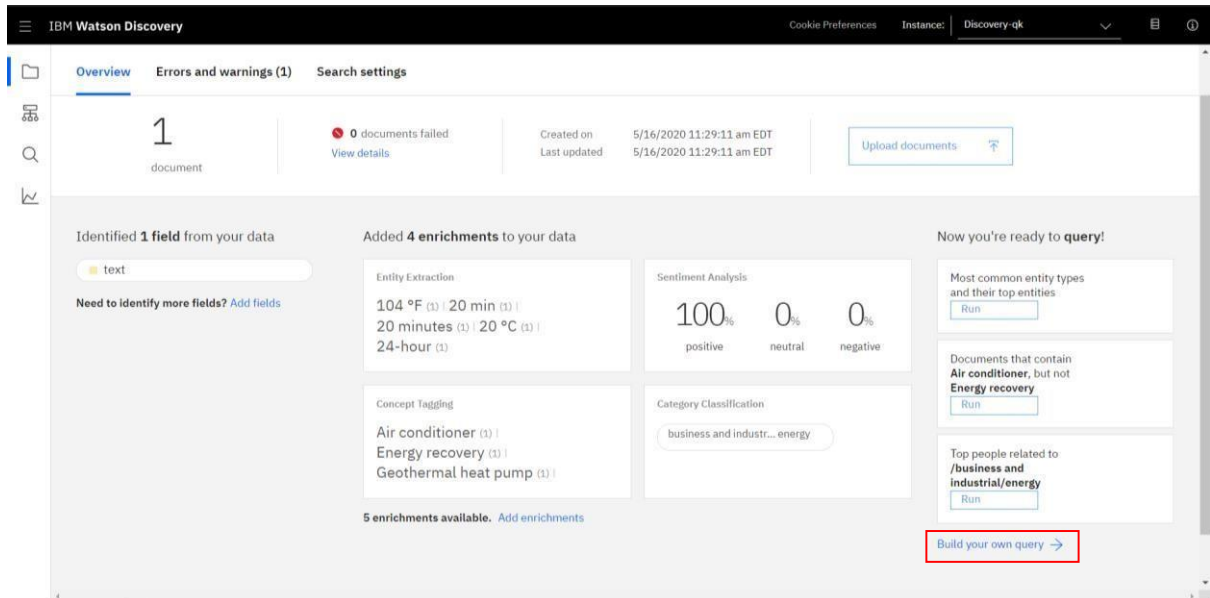
SDU trains Watson Discovery to extract custom fields in your documents. Customizing how your documents are indexed into Discovery will improve the answers returned from queries. With SDU, you annotate fields within your documents to train custom conversion models. As you annotate, Watson is learning and will start predicting annotations. SDU models can also be exported and used on other collections.

Current document type support for SDU is based on your plan:

- Lite plans: PDF, Word, PowerPoint, Excel, JSON, HTML

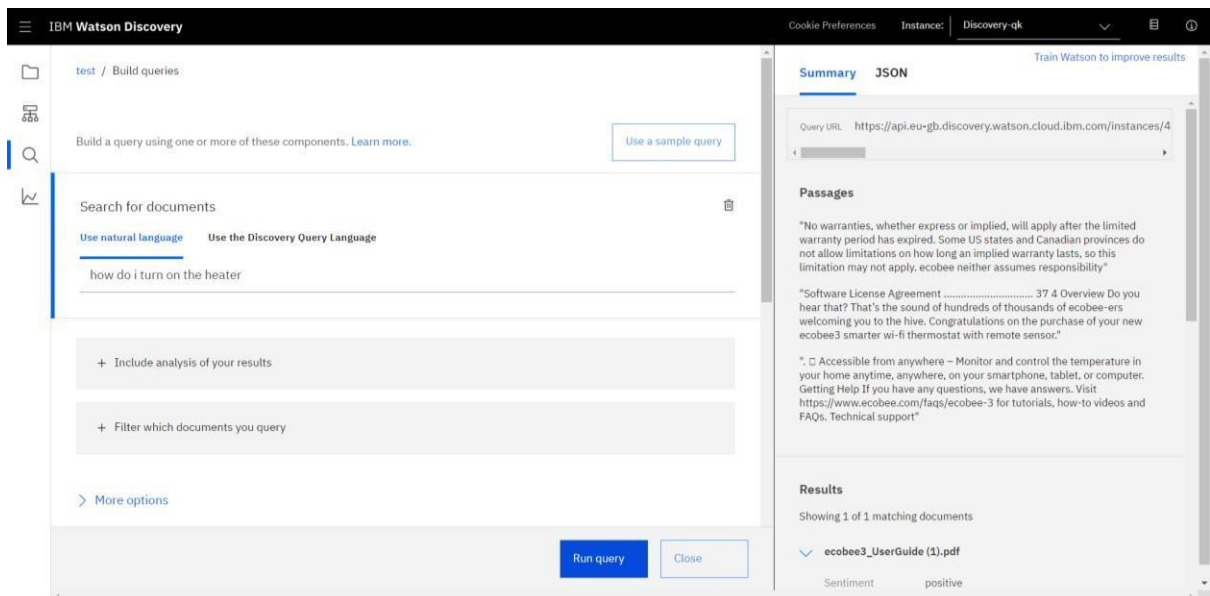
- Advanced plans: PDF, Word, PowerPoint, Excel, PNG, TIFF, JPG, JSON, HTML

Before applying SDU to our document, let's do some simple queries on the data so that we can compare it to results found after applying SDU.



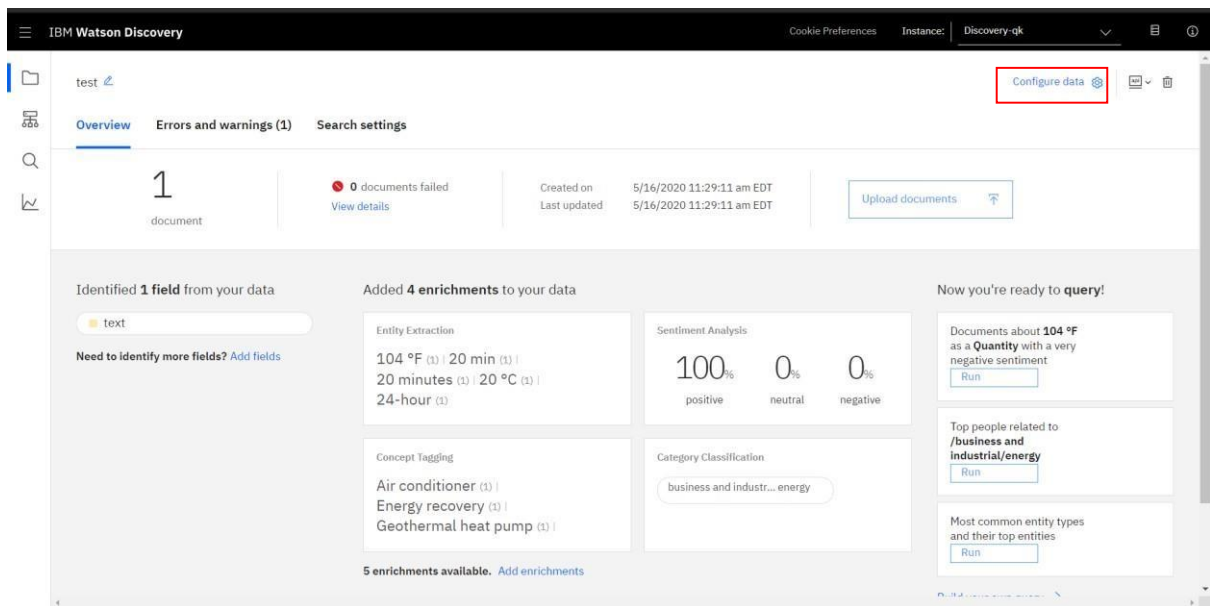
Click on Build your own query.

Now, enter queries related to the operation of the thermostat and view the results. As you will see, the results are not very useful, and in some cases, not even related to the question.



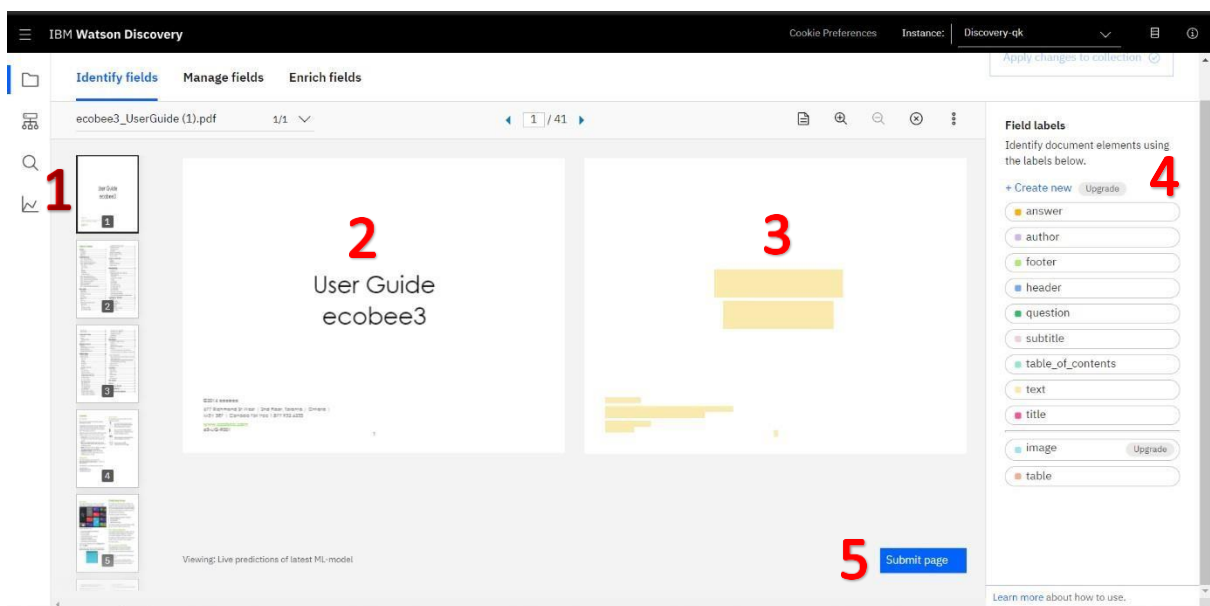
Now let's apply SDU to our document to see if we can generate some better query responses.

Go back to the Discovery collection panel (previous screen)



Click the Configure data button (located in the top right corner) to start the SDU process.

Here is the layout of the Identify fields tab of the SDU annotation panel



The goal is to annotate all of the pages in the document so Discovery can learn what text is important, and what text can be ignored.

[1] is the list of pages in the manual. As each is processed, a green check mark will appear on the page.

[2] is the current page being annotated.

[3] is where you select text and assign it a label.

[4] is the list of labels you can assign to the page text.

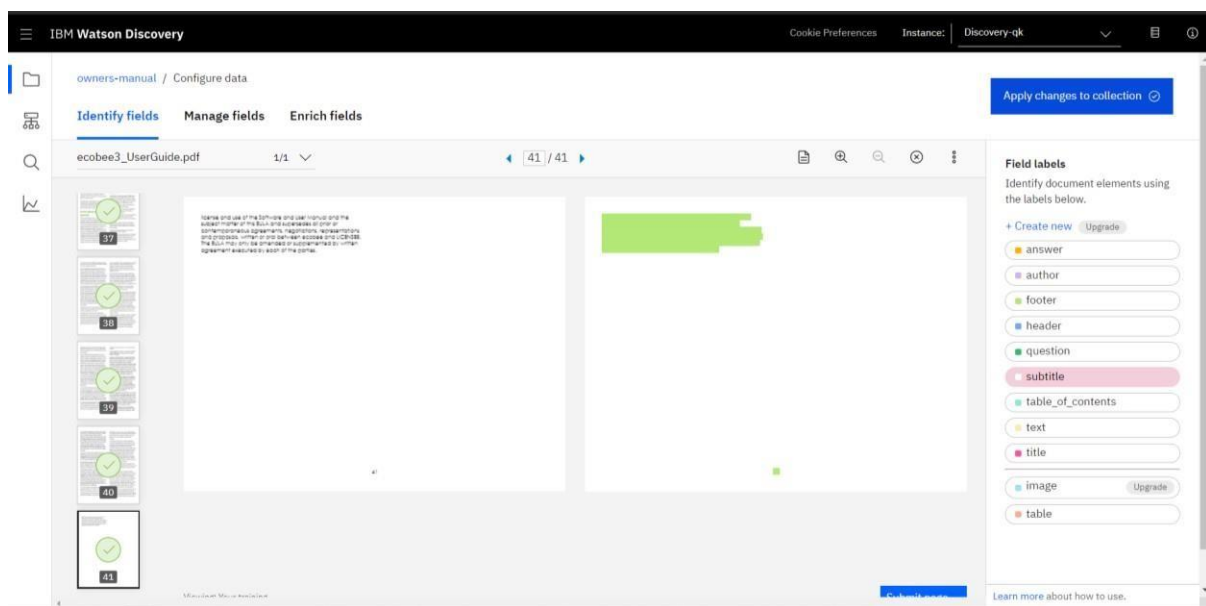
Click [5] to submit the page to Discovery.

As you go through the annotations one page at a time, Discovery is learning and should start automatically updating the upcoming pages. Once you get to a page that is already correctly annotated, you can stop, or simply click Submit [5] to acknowledge it is correct. The more pages you annotate, the better the model will be trained.

For this specific owner's manual, at a minimum, it is suggested to mark the following:

- The main title page as title
- The table of contents (shown in the first few pages) as table_of_contents
- All headers and sub-headers (typed in light green text) as a subtitle
- All page numbers as footers
- All warranty and licensing information (located in the last few pages) as a footer
- All other text should be marked as

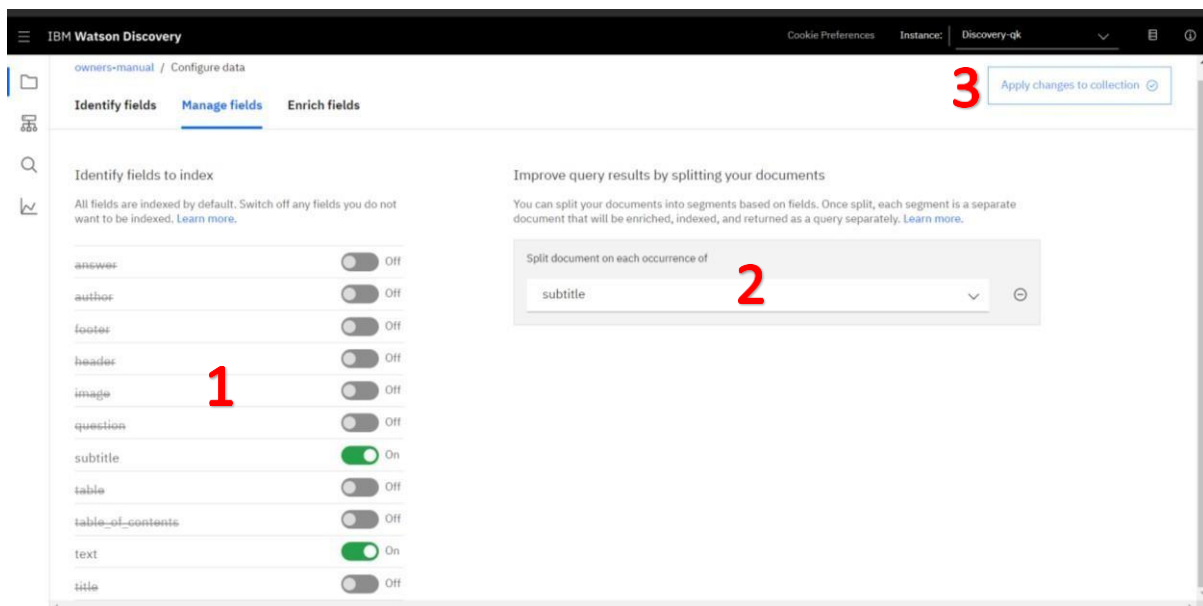
text. After completing the process for all pages,



Click the [Apply changes to collection](#) button.

You will be asked to reload the document. Choose the same ecobee3_UserGuide.pdf document as before.

Now, click on Manage fields tab on the Configure data panel



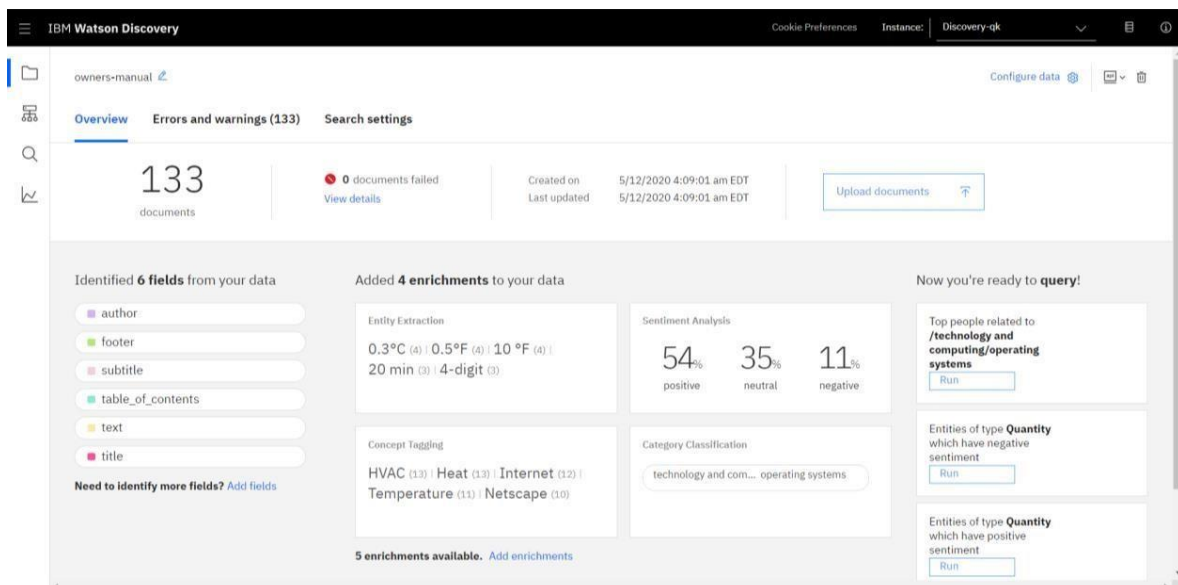
[1] Here is where you tell Discovery which fields to ignore. Using the on/off buttons, turn off all labels except subtitles and text.

[2] is telling Discovery to split the document apart, based on subtitle.

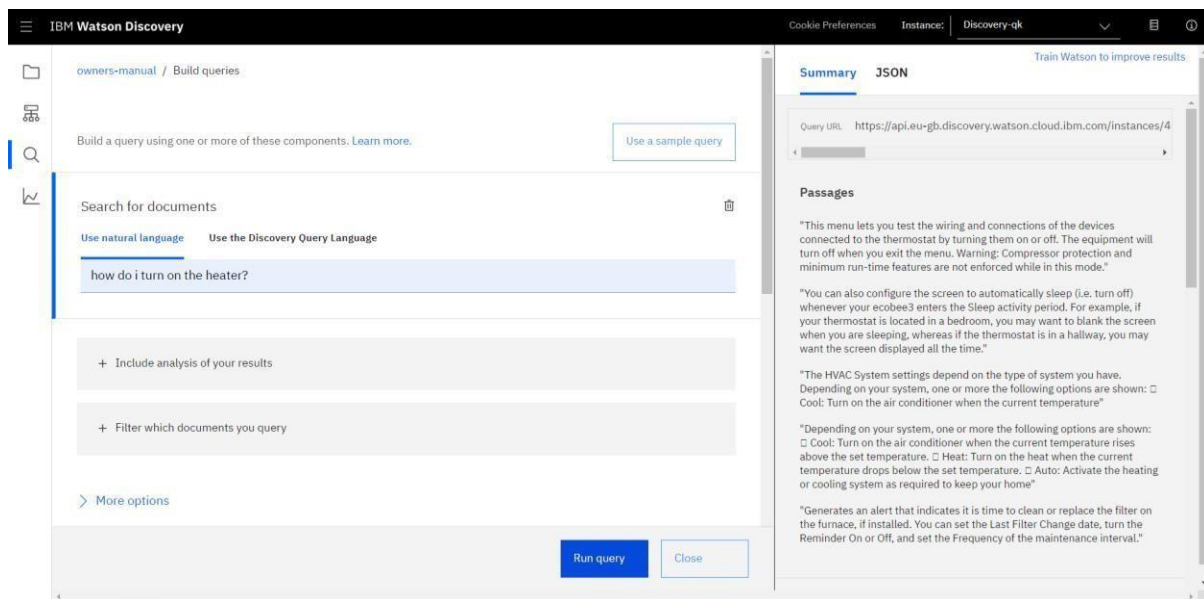
Click [3] to submit your changes.

Once again, you will be asked to reload the document. Choose the same ecobee3_UserGuide .pdf document as before.


Now, as a result of splitting the document apart, your collection will look different

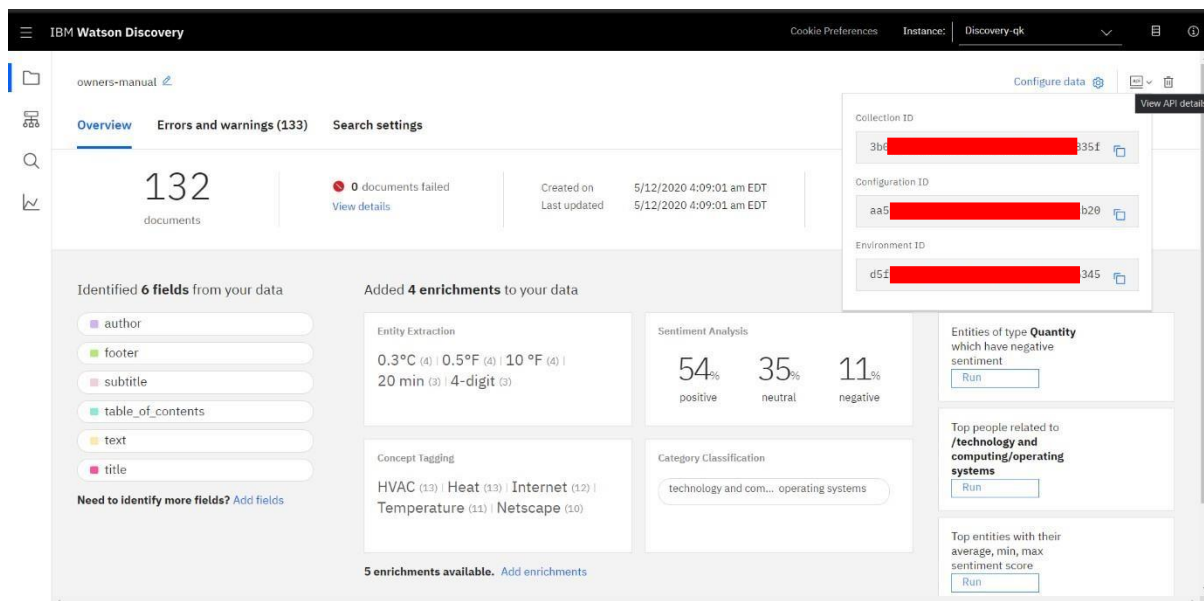


Now click the Build your own query and see how much better the results are.



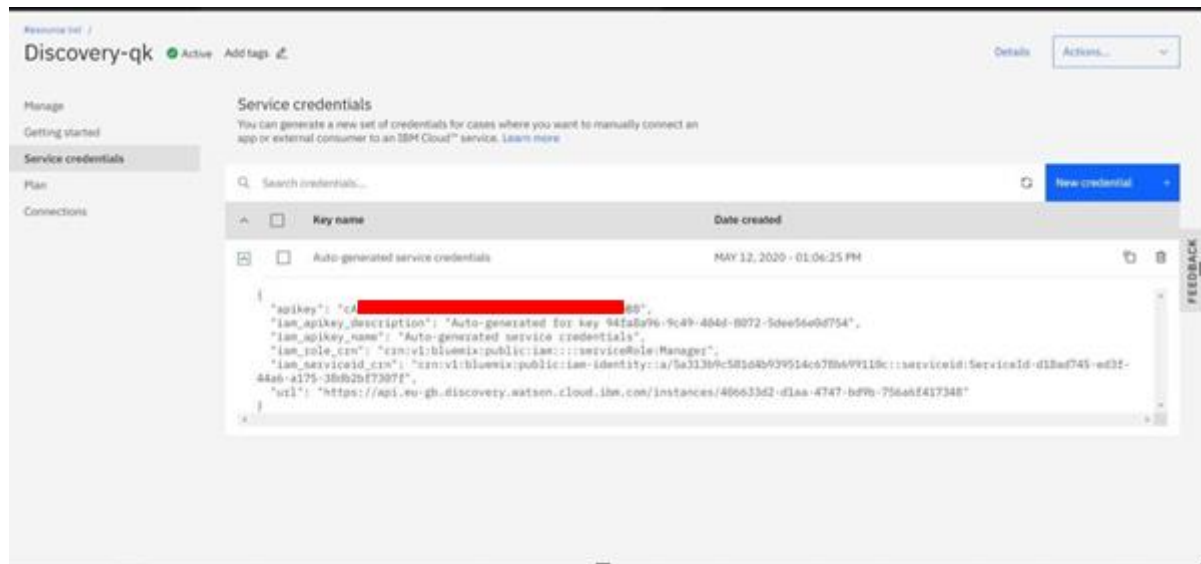
In upcoming steps, you will need to provide some credentials to access your Discovery collection so to Store credentials for future use follow the below steps.

The Collection ID and Environment ID values can be found by clicking the  located at the top right side of your collection panel



Now go to the Watson Discovery Resource List Screen. Here, select service credentials.

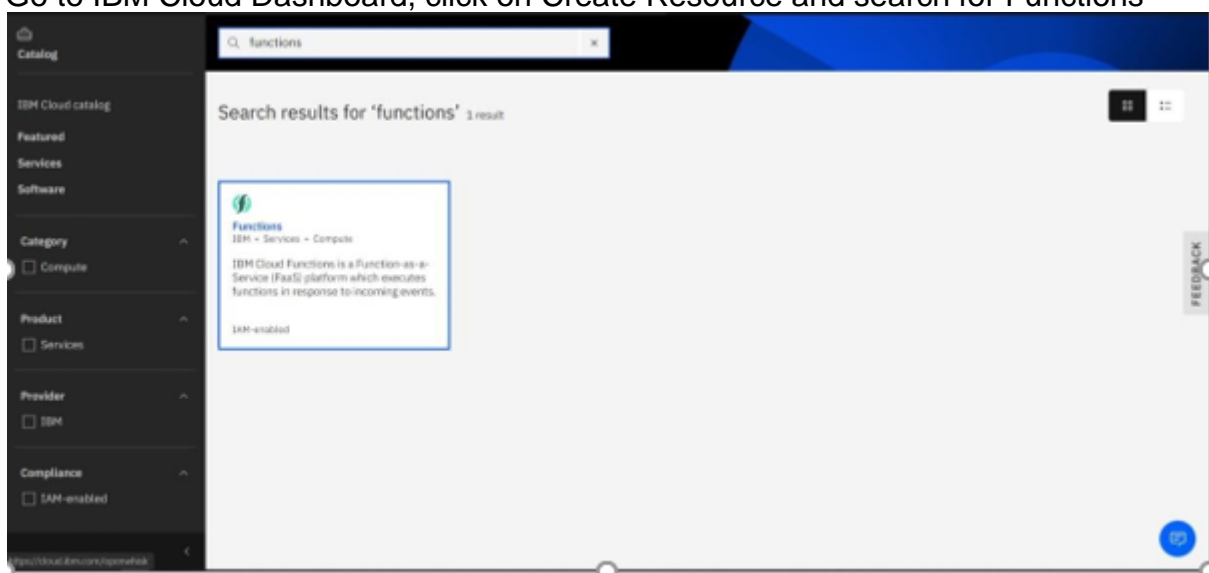
The apikey and URL endpoint for your service can be found here.



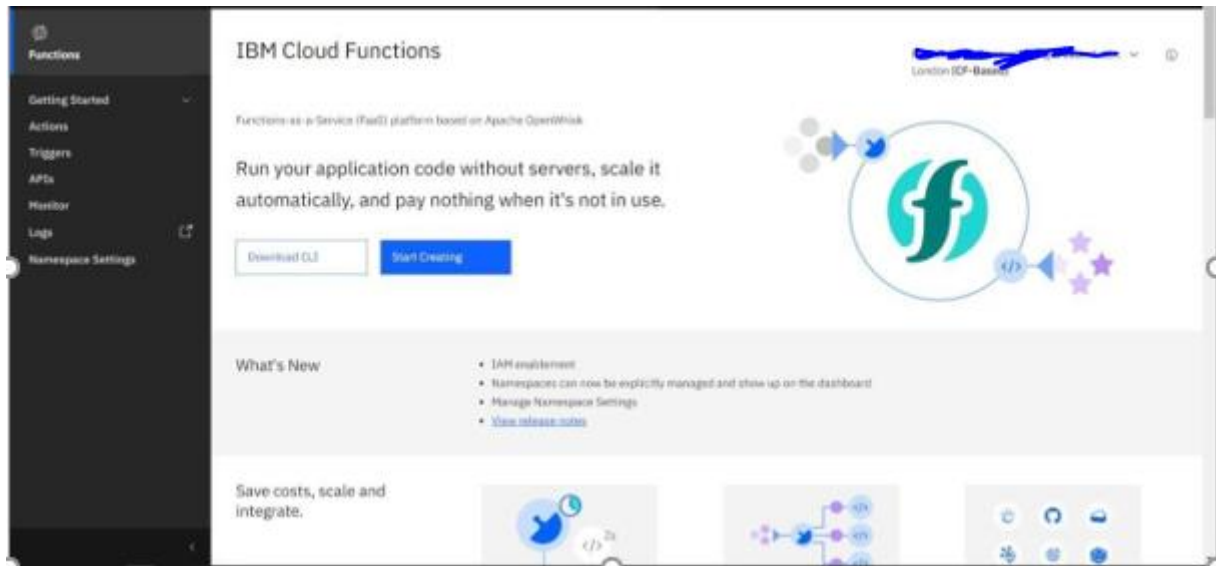
3. Create IBM Cloud Functions action

Now let's create the web action that will make queries against our Discovery collection.

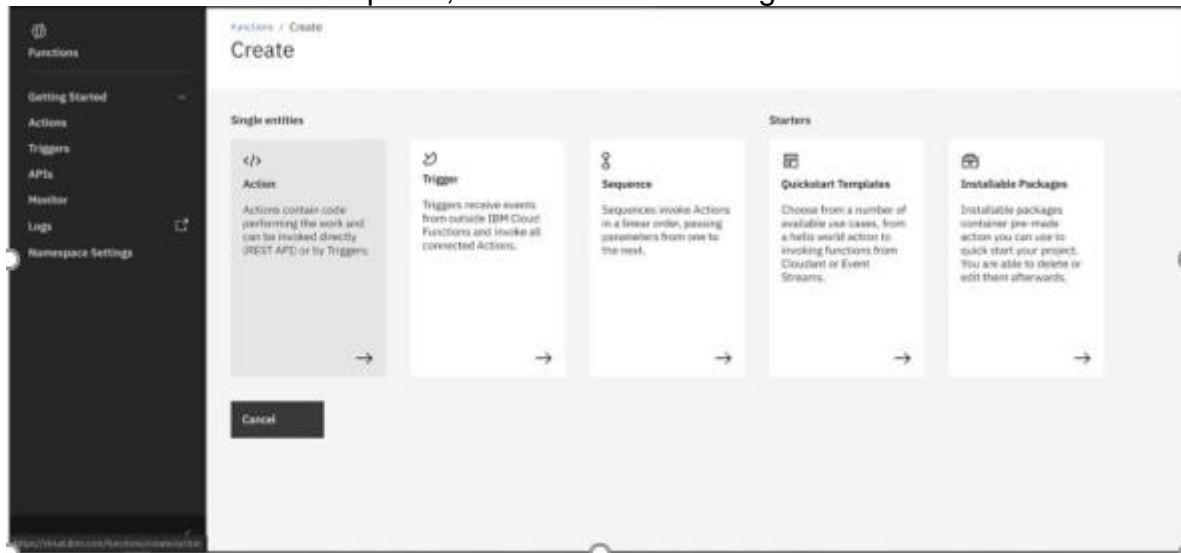
Go to IBM Cloud Dashboard, click on Create Resource and search for Functions



Select the Functions Card.



From the Functions main panel, click on Start Creating.



Here, select Actions.

Functions

Getting Started

Actions

Triggers

APIs

Monitor

Logs

Namespace Settings

Functions / Create / Action

Create Action

Actions contain your function code and are invoked by events or REST API calls.

[Learn more about Actions](#)

[Learn more about Packages](#)

Action Name

Enclosing Package

(Default Package)

Create Package

Runtime

Node.js 10

Looking for Java, .NET or Docker? [Docker Actions](#) can be created with the [CLI](#)

Previous

Cancel

Create

On the Create Action panel, provide a unique Action Name, keep the default package, and select the Node.js 10 runtime. Click the Create button to create the action.

Once function is created, click on code tab.



The screenshot shows the Google Cloud Functions console. On the left, there's a sidebar with navigation links: Code, Parameters, Runtime, Endpoints, Connected Triggers, Enclosing Sequences, and Logs. The main area is titled 'disco_action' and 'Web Action'. The code editor shows the following code:

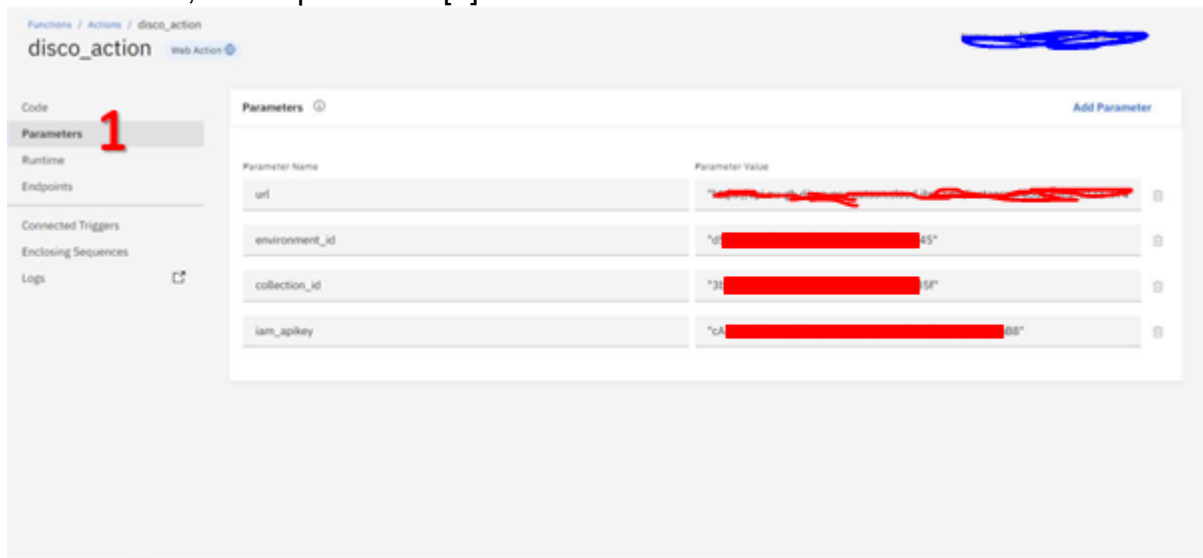
```
1 // **
2 *
3 * @param {object} params
4 * @param {string} params.api_key
5 * @param {string} params.url
6 * @param {string} params.username
7 * @param {string} params.password
8 * @param {string} params.environment_id
9 * @param {string} params.collection_id
10 * @param {string} params.configuration_id
11 * @param {string} params.input
12 *
13 * @returns {object}
14 *
15 */
16
17 const assert = require('assert');
18 const DiscoveryV3 = require('googleapis/discovery/v3');
19
20 /**
21 *
22 * main() will be run when you invoke this action
23 *
24 * @param Cloud Functions actions accept a single parameter, which must be a JSON object.
25 *
26 * @return the output of this action, which must be a JSON object.
27 *
28 */
29 function main(params) {
30   return new Promise(function (resolve, reject) {
31     // let discovery
32
33     // if (params.api_key){
```

On the right side of the code editor, there are two buttons: 'Invoke with parameters' and 'Invoke'. A red '1' is placed over the 'Invoke with parameters' button, and a red '2' is placed over the 'Invoke' button.

In the code editor window [1], cut and paste in the code from the disco-action.js file found in the actions directory of your local repo. The code is pretty straight-forward - it simply connects to the Discovery service, makes a query against the collection, then returns the response.

Now if click the invoke [2] button, it will fail due to credentials not being defined yet.

To solve this, select parameter[1] tab

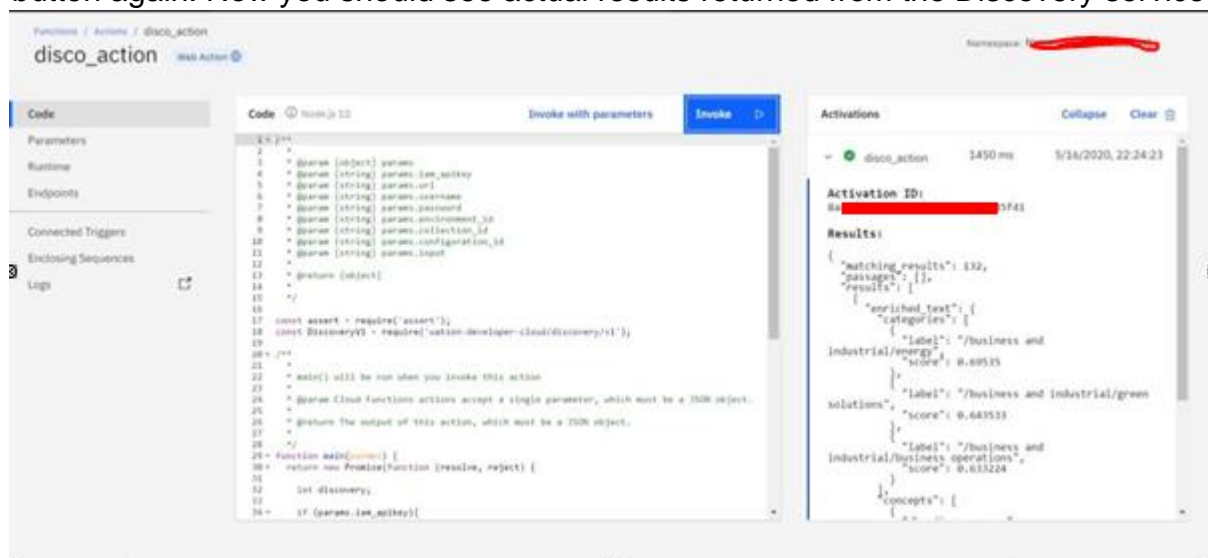


Add the following keys:

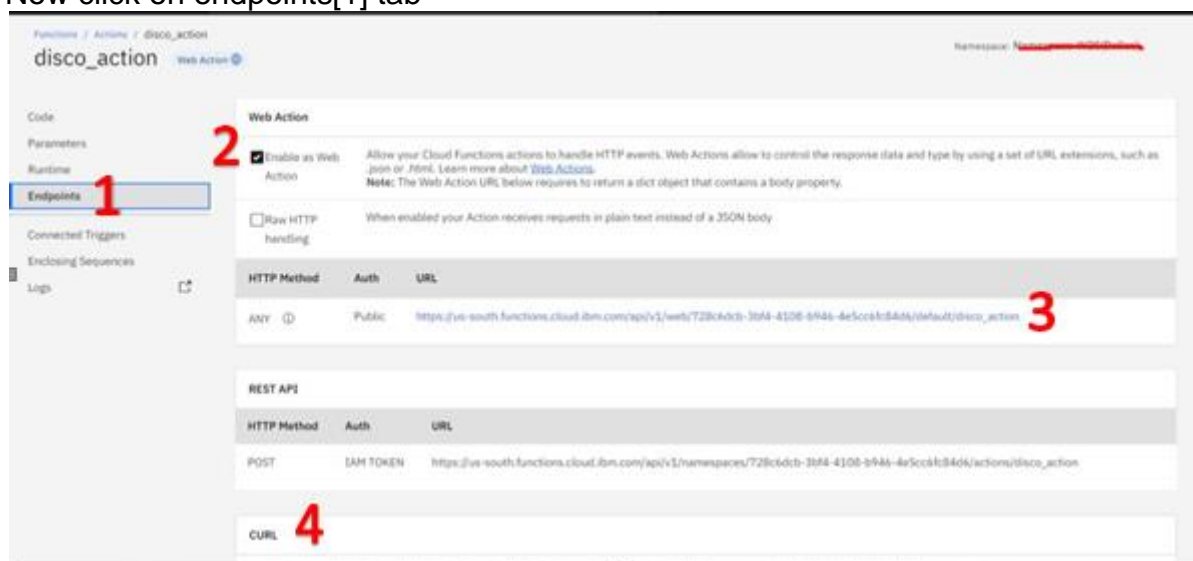
- url
- environment_id
- collection_id
- iam_apikey

For the above values, please use the values associated with the Discovery service you created in the previous step. Enclose your values in double quotes.

Now that the credentials are set, return to the Code panel tab and press the Invoke button again. Now you should see actual results returned from the Discovery service



Now click on endpoints[1] tab




Click the checkbox for Enable as Web Action [2]. This will generate a public endpoint URL [3].

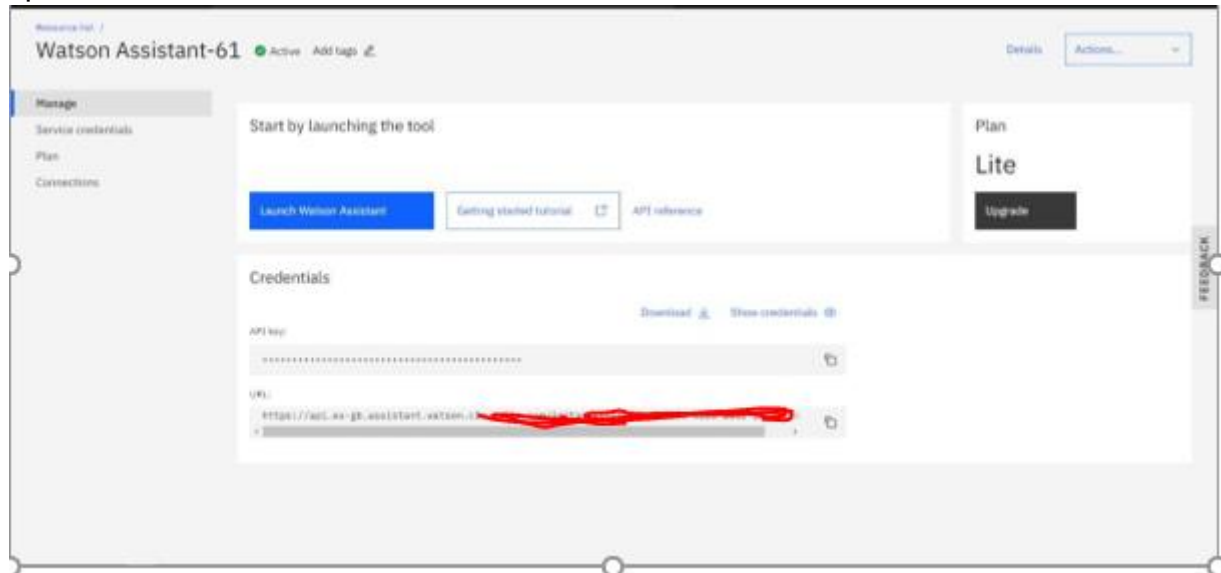
Take note of the URL value [3], as this will be needed by Watson Assistant in a future step.


To verify you have entered the correct Discovery parameters, execute the provided curl command [4]. If it fails, re-check your parameter values.

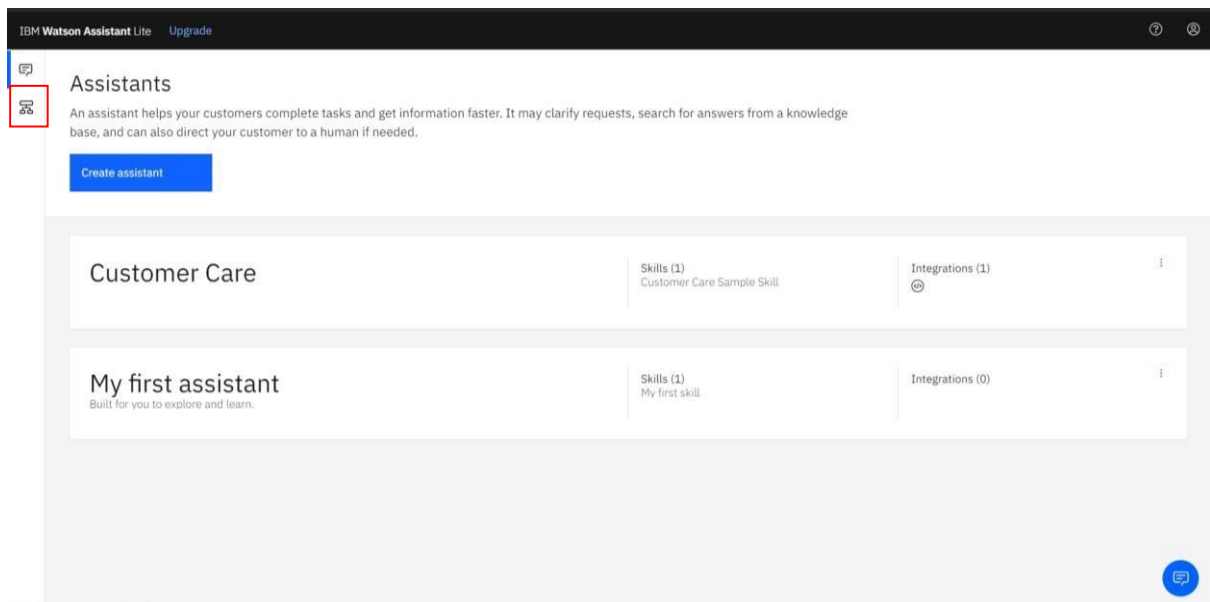
[An IBM Cloud Functions service will not show up in your dashboard resource list. To return to your defined Action, you will need to access Cloud Functions by selecting Create Resource from the main dashboard panel (as shown at the beginning of this step).]

4. Configure Watson Assistant

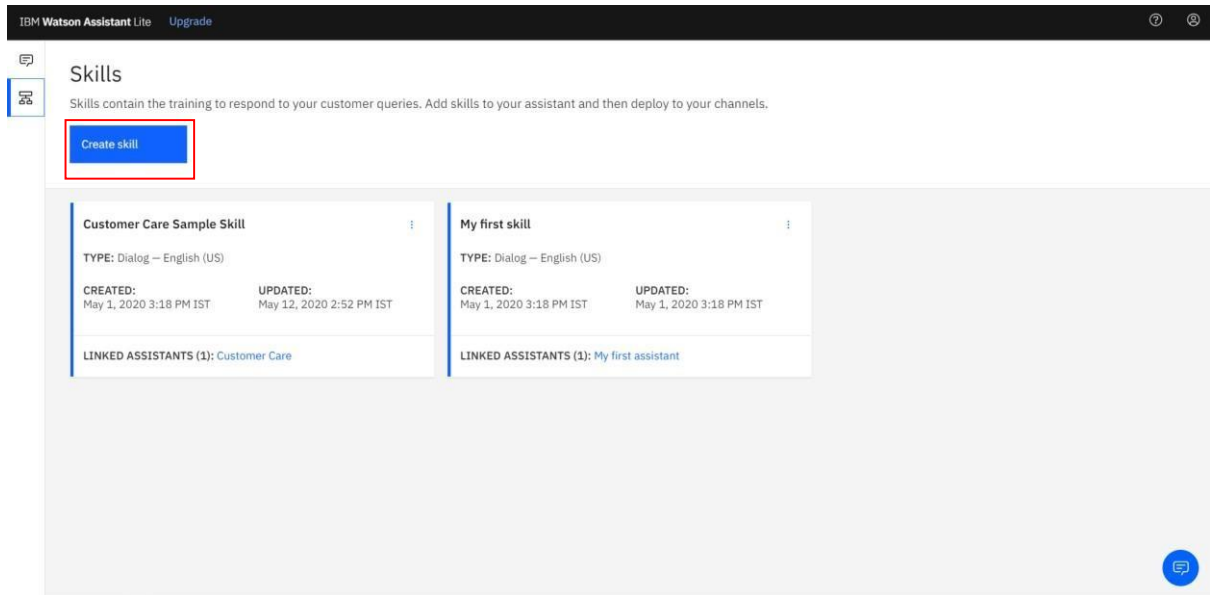
Go back to the IBM Dashboard From the resource list screen,  Watson Assistant-61 click to open Watson Assistant service.



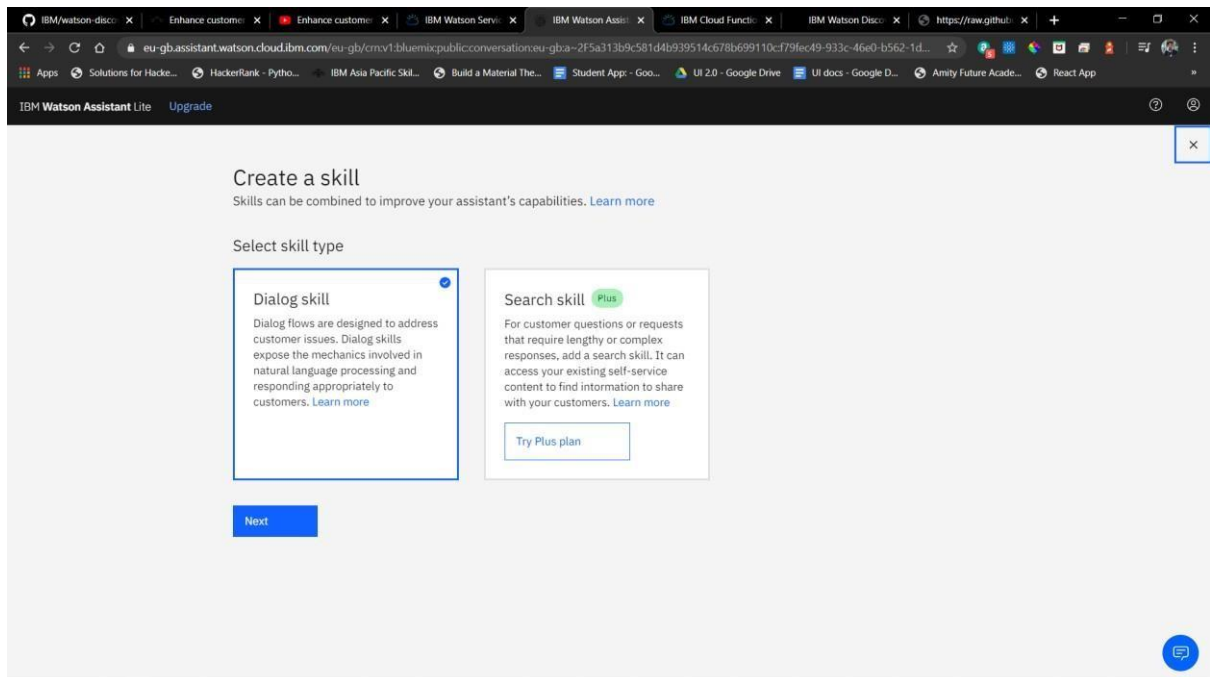
Click on  to launch Watson Assistant.



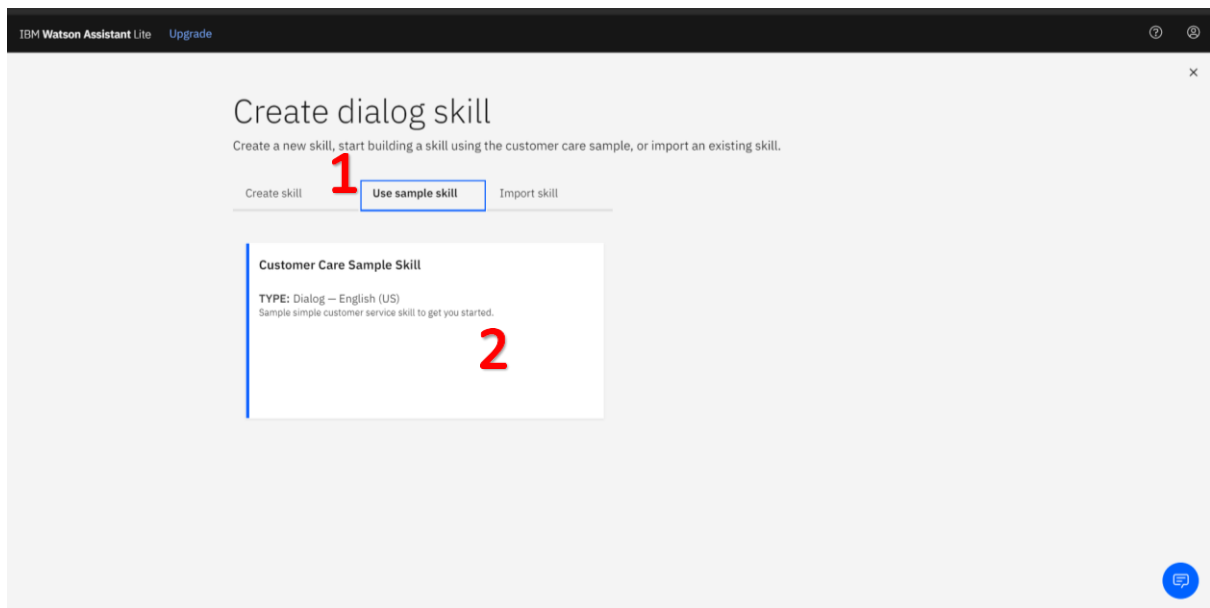
Click on the skills tab



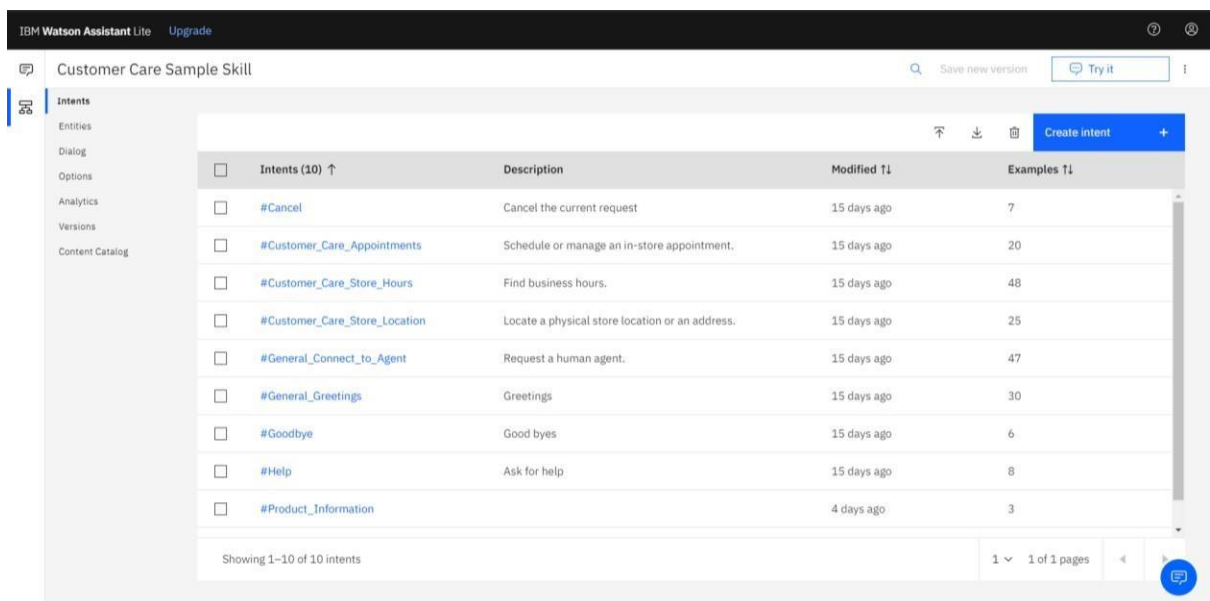
Click Create Skill



Select Dialog Skill Card and Click next



Select Use Sample Skill [1] and Select Customer Care Sample Skill [2]. This dialog skill contains all of the nodes needed to have a typical call centre conversation with a user.



As the default customer care dialog does not have a way to deal with any questions involving outside resources, so we will need to add new intent.

Create a new intent that can detect when the user is asking about operating the Ecobee thermostat.

From the Customer Care Sample Skill panel, select the Intents tab.

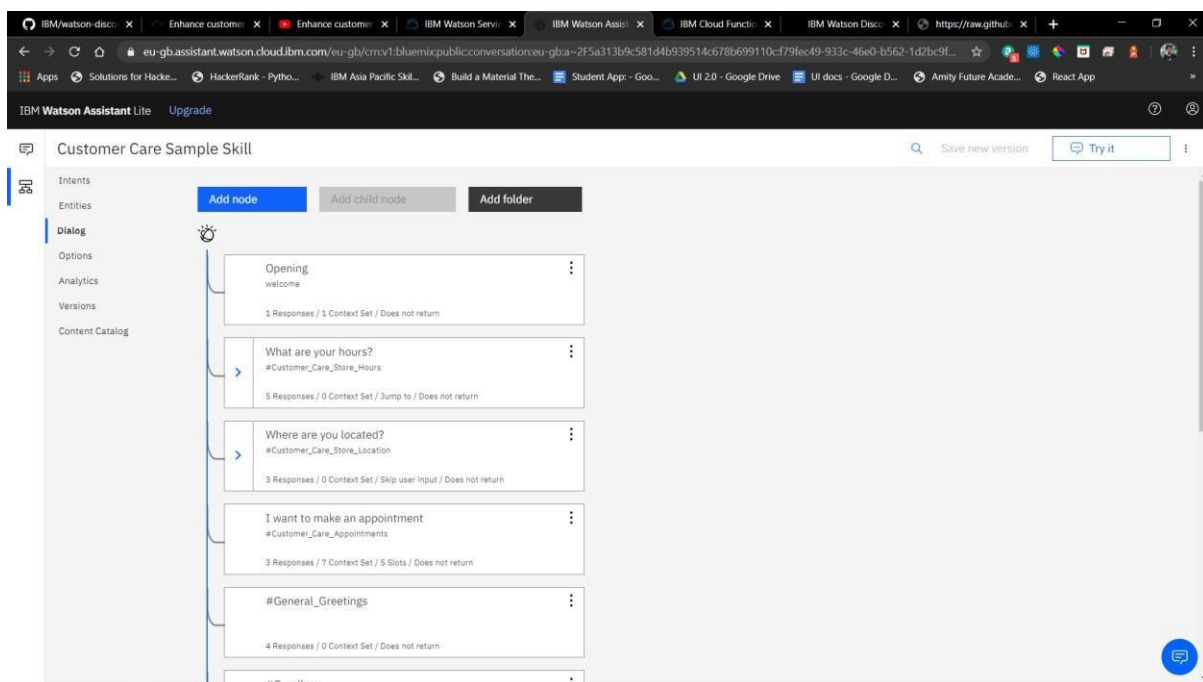
Click the Create intent button.

Name the intent `#Product_Information`, and at a minimum, enter the following example questions to be associated with it.

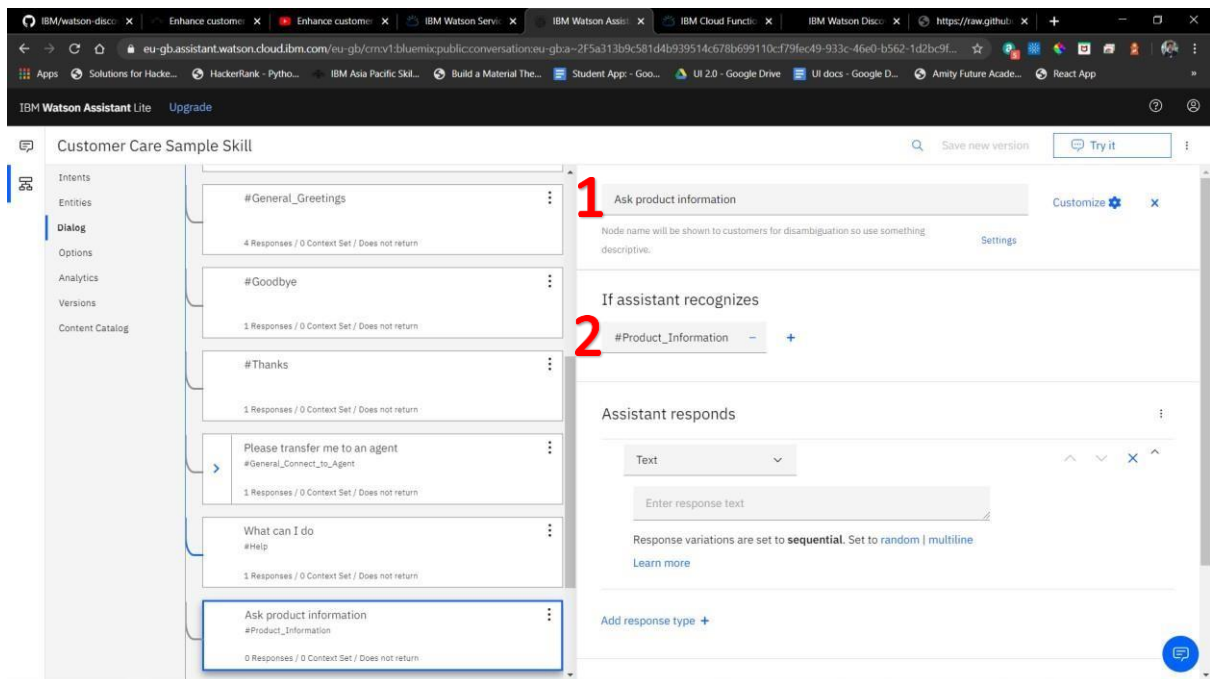
The screenshot shows the 'Intents' configuration page for the intent `#Product_Information`. The page has a dark header with 'IBM Watson Assistant Lite' and an 'Upgrade' button. Below the header, the breadcrumb is '#Product_Information' and the status is 'Last updated: 4 days ago'. The main form area includes:

- Intent name:** A text box containing `#Product_Information`.
- Description (optional):** A text box with the placeholder 'Add a description to this intent'.
- User example:** A text box with the placeholder 'Type a user example here, e.g. I want to pay my credit card bill'.
- Buttons:** 'Add example' and 'Show recommendations'.
- Entity Annotation:** A toggle switch for 'Annotate entities' with a link 'What's this?'.
- User examples table:** A table with 3 examples. The first example is 'User examples (3) ↑' with 'Added 11' days ago. The other two examples are 'How do i access the settings' and 'How do i set the time', both added '4 days ago'.
- Footer:** 'Showing 1-3 of 3 examples' and pagination controls '1 1 of 1 pages'.

Go back to the previous page after doing this, then click on Dialog Tab and add a node below What can I do node.



Name the node "Ask product information" [1] and assign it our new intent #Product_Information [2].

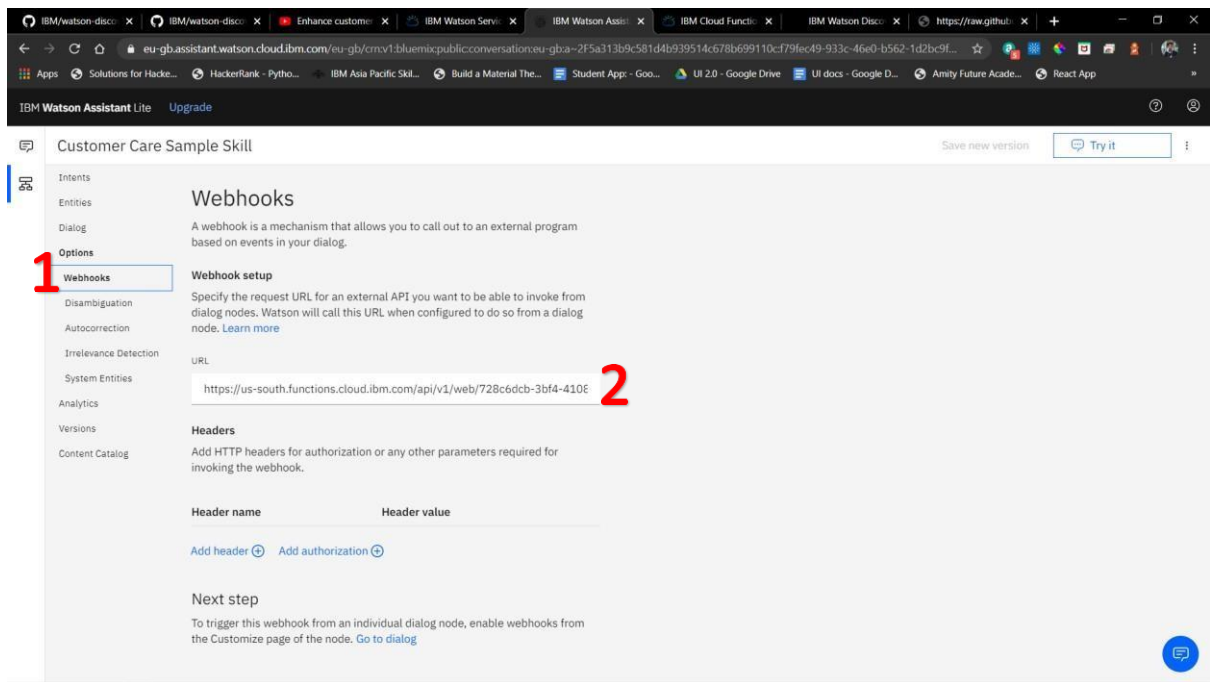


This means that if Watson Assistant recognizes a user input such as "how do I set the time?", it will direct the conversation to this node.

Before proceeding further, let's learn about webhook.

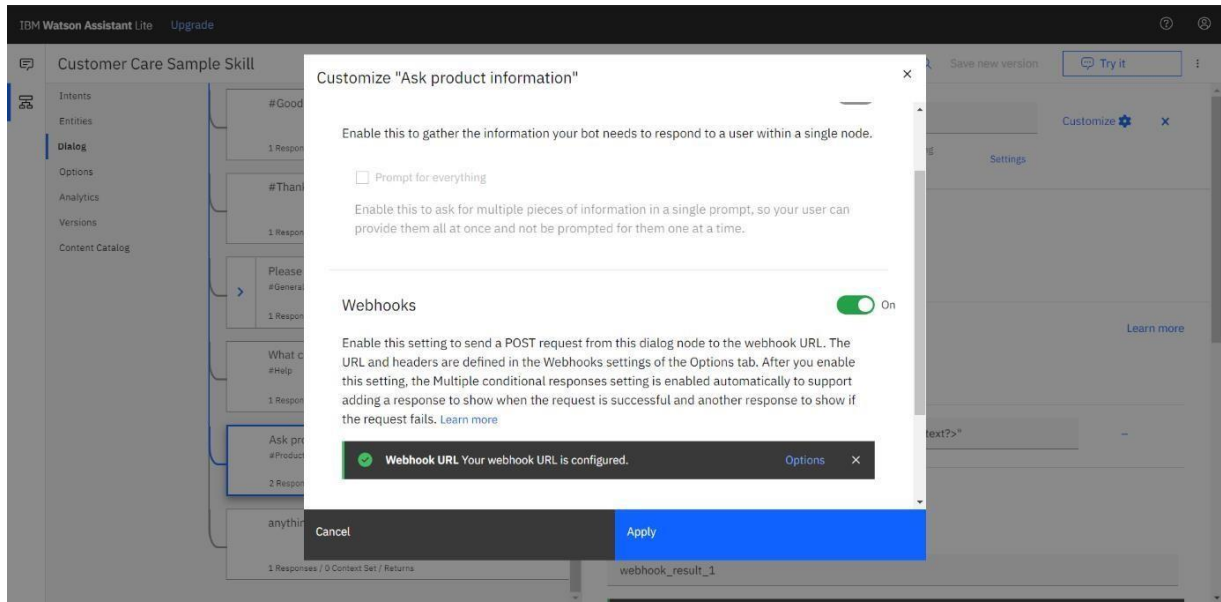
A webhook is a mechanism that allows you to call out to an external program based on something happening in your program. When used in a Watson Assistant dialog skill, a webhook is triggered when the Assistant processes a node that has a webhook enabled. The webhook collects data that you specify or that you collect from the user during the conversation and save in context variables, and sends the data to the Webhook request URL as an HTTP POST request. The URL that receives the webhook is the listener. It performs a predefined action using the information that is provided by the webhook as specified in the webhook definition, and can optionally return a response.

In our example, the webhook will communicate with an IBM Cloud Functions web action, which is connected to the Watson Discovery service.



Click Webhooks[1] tab and enter the URL[2] to enable web hook for the IBM Cloud Functions action you created in Step 3. Add .json to the end of the URL to specify the result should be in JSON format.

Return to the Dialog tab, and click on the Ask product information node. From the details panel for the node, click on Customize, and enable Webhooks for this node



Click

Apply.

The dialog node should have a Return variable [1] set automatically to \$webhook_result_1. This is the variable name you can use to access the result from the Discovery service query.

The screenshot shows the IBM Watson Assistant interface for a 'Customer Care Sample Skill'. On the left, a 'Dialog' tree lists several nodes, with 'Ask product information' (#Product_Information) selected and highlighted with a blue border. A red number '1' points to the 'Return variable' field in the configuration panel, which is set to 'webhook_result_1'. A red number '2' points to the 'input' parameter in the 'Parameters' section, where the 'Value' is set to '<?input.text?>'. A green notification banner at the bottom states 'Webhook URL Your webhook URL is configured. Options'.

You will also need to pass in the users question via the parameter input [2]. The key needs to be set to the value:

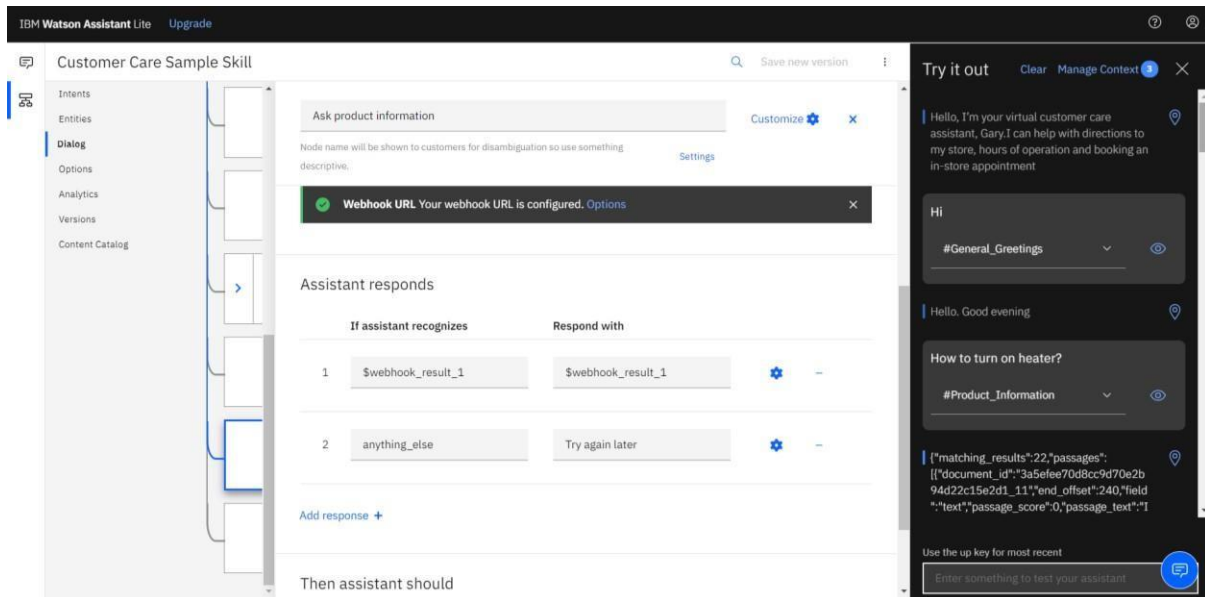
"<?input.text?>"

If you fail to do this, Discovery will return results based on a blank query.

Optionally, you can add these responses to aid in debugging:

This screenshot shows the 'Assistant responses' configuration for the 'Ask product information' node. A red box highlights the 'Respond with' column. For the first response (when the assistant recognizes '\$webhook_result_1'), the response is '\$webhook_result_1'. For the second response (for 'anything_else'), the response is 'Try again later'. The 'If assistant recognizes' column lists the conditions for each response.

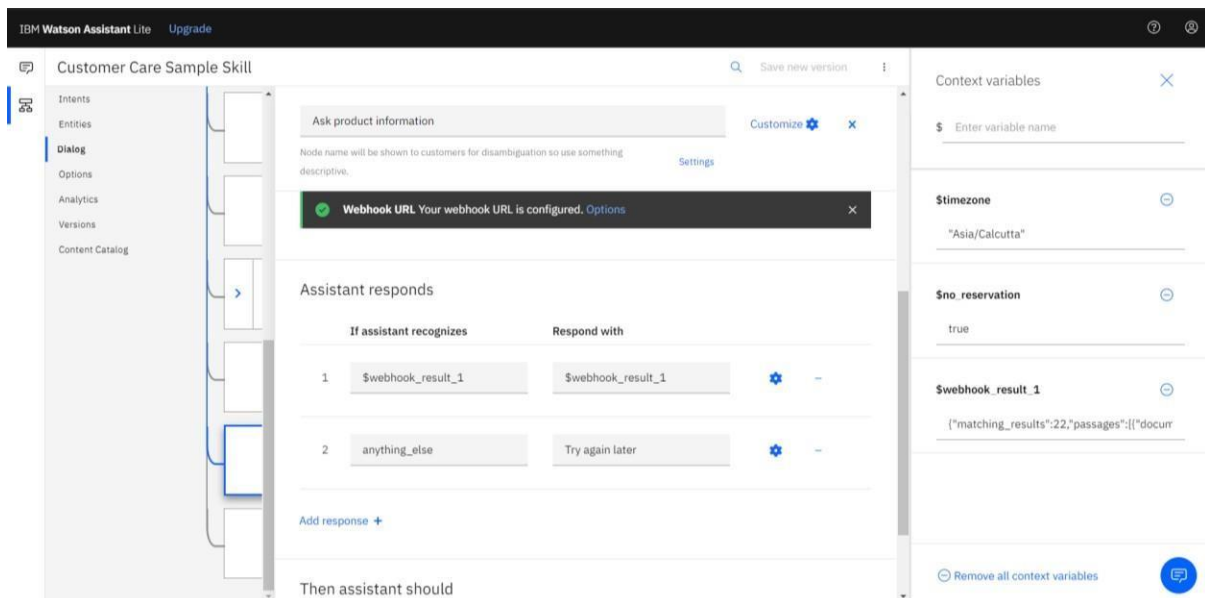
Now we should test the assistant, from click the Try it button located at the top right side of the panel.



Note that the input "how do I turn on the heater?" has triggered our Ask about product dialog node, which is indicated by the #Product_Information response.

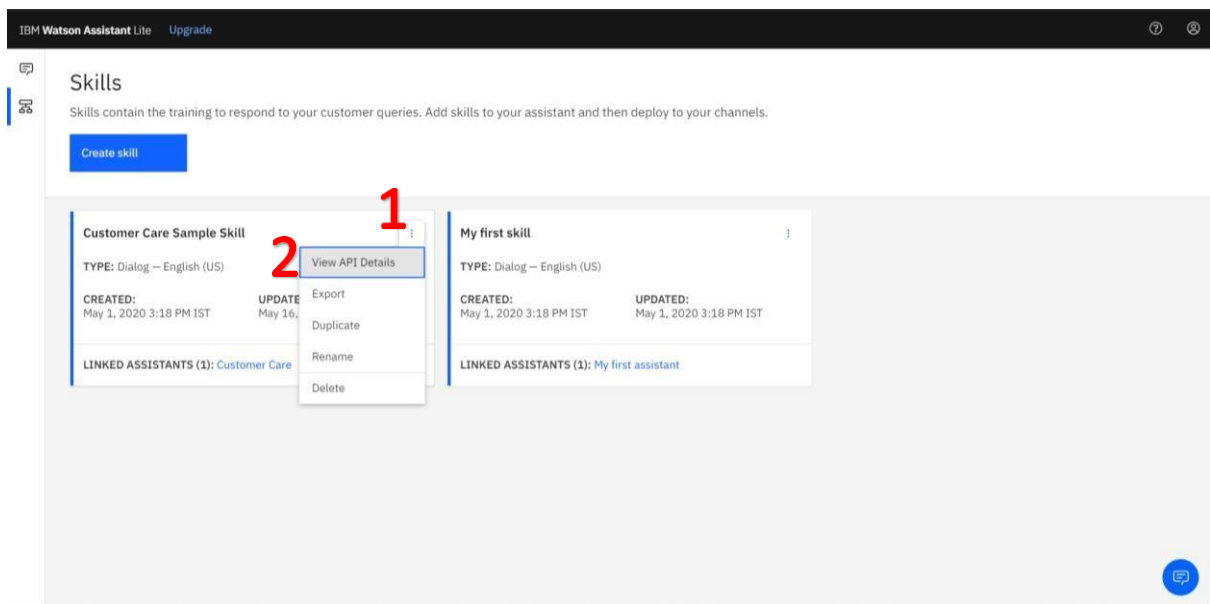
And because we specified that \$webhook_result_1.passages be the response, that value is displayed also.

You can also verify that the call was successfully completed by clicking on the Manage Context button at the top right. The response from the Discovery query will be stored in the \$webhook_result_1 variable



For upcoming steps, you will need to provide some credentials to access your assistant so to store credentials for future use follow these steps below.

Go back to the skills tab, click [1] and then [2]



IBM Watson Assistant Lite Upgrade

Skills

Skills contain the training to respond to your customer queries. Add skills to your assistant and then deploy to your channels.

Create skill

Customer Care Sample Skill
TYPE: Dialog — English (US)
CREATED: May 1, 2020 3:18 PM IST
UPDATED: May 16, 2020 3:18 PM IST
LINKED ASSISTANTS (1): Customer Care

1

2

View API Details

Export

Duplicate

Rename

Delete

My first skill
TYPE: Dialog — English (US)
CREATED: May 1, 2020 3:18 PM IST
UPDATED: May 1, 2020 3:18 PM IST
LINKED ASSISTANTS (1): My first assistant

The Skill ID and API Key is to be noted.



Skill details

Skill name: Customer Care Sample Skill

Skill ID: **[REDACTED]**

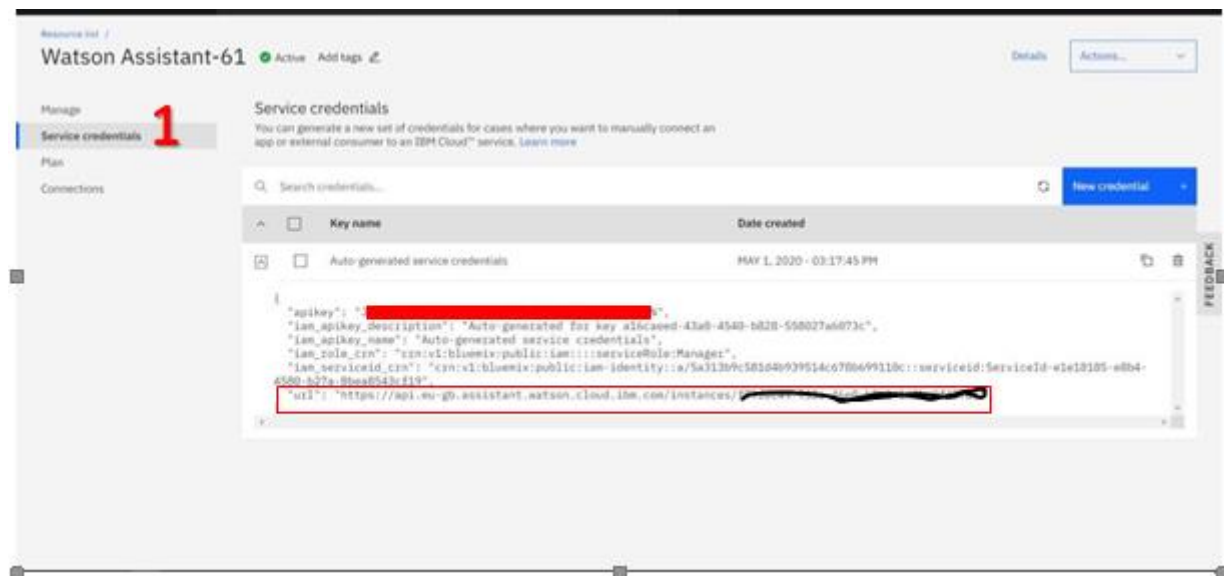
Legacy v1 workspace URL: <https://api.eu-gb.assistant.watson.cloud.ibm.com/instances/.../workspaces/...>

Service credentials

Service credentials name: Auto-generated service credentials

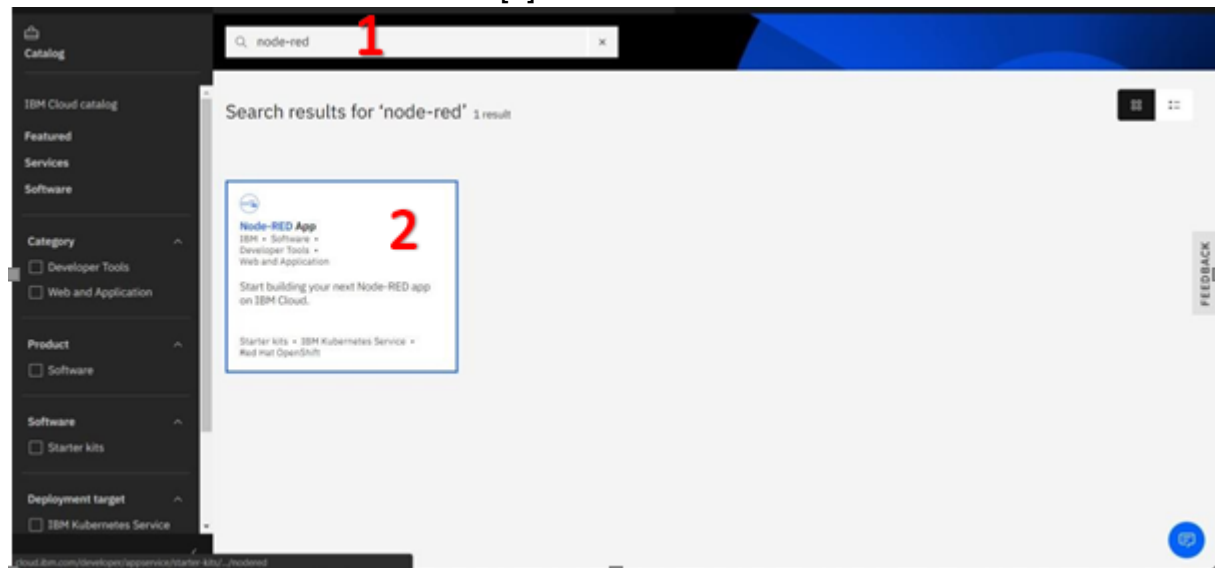
API key: **[REDACTED]**

Go Back to the Watson Assistant Resource List, Select Service Credentials [1] and make note of the URL.APIKEY can be found here too.



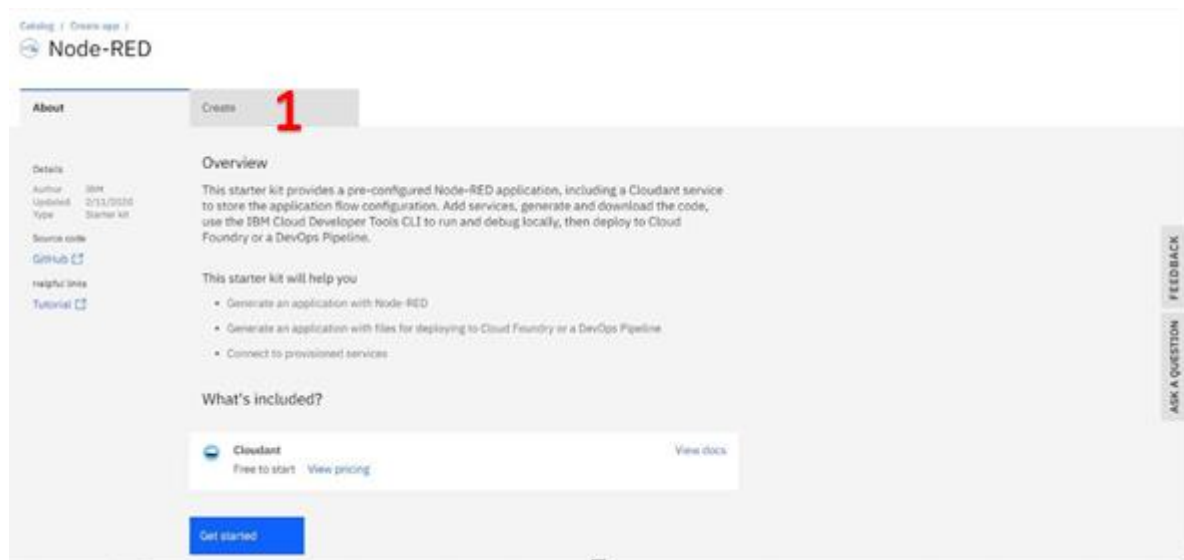
5. Build Node-RED Flow to Integrate All Services

Now it's time to create Node-Red, go to IBM Cloud Dashboard, click on Create Resource and search for node-red[1].

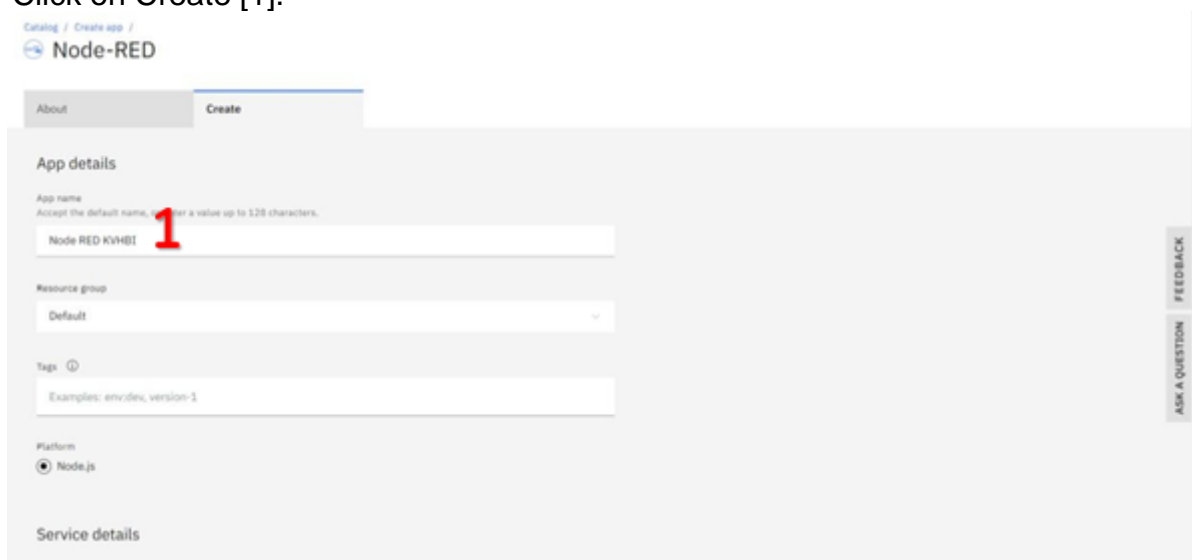


Click on the Node-RED App tile [2].

This will show you an overview of the Starter Kit and what it provides.



Click on Create [1].



Now you need to configure the Node-RED Starter application.

On the App details page, a randomly generated name will be suggested – Node RED KVHBI in the screenshot above. Either accept that default name or provide a unique name for your application [1]. This will become part of the application URL. Note: If the name is not unique, you will see an error message and you must enter a different name before you can continue.

Select the region [1] the service should be created in and what pricing plan it should use. You can only have one Cloudant instance using the Lite plan and You can have more than one Node-RED Starter application using the same Cloudant service instance.

Tags ⓘ

Examples: env:dev, version:1

Platform

☒ Node.js

Service details

Cloudant ⓘ

• You have existing instances of this service available to use in this kit. If you wish to use the existing service, select it from the pricing plan menu.

Region

Chennai 1

Resource group

Default

Pricing plan

Lite 2

[Pricing details](#)

[Terms](#)

ⓘ

Cancel

Create 3

Resource list / App details /

Node RED KVHBI Add tags

Actions...

Details

App URL

You must deploy your app first

Source

Download code

Resource group

Default

Deployment target

You must deploy your app first

Created

5/16/2020

Services (1)

Connect existing services + Create service +

Name	Resources	Actions
Cloudant	Documentation	

Deployment Automation

Configure Continuous Delivery

Continuous Delivery is not enabled for this app. Enable Continuous Delivery to automate builds, tests, and deployments through Delivery Pipeline, GetLab, and more.

Deploy your app

Credentials

Show

```

{
  "cloudant": [
    {
      "name": "node-red-kvmbi-cloudant-159811941473",
      "credentials": {
        "apikey": ".....",
        "host": "00136238-812a-4122-80aa-f7e79c4da796@us-east-1.cio",
        "password": "....."
      }
    }
  ]
}
                
```

FEEDBACK

ASK A QUESTION

You will need to create an IBM Cloud API key to allow the deployment process to access your resources. Click the New button (1) to create the key. A message dialog will appear. Read what it says and then confirm and close the dialog.

Resource list / App details

Node RED KVHBI

Select the deployment target ☒ Configure the DevOps toolchain ☐

Deployment Automation

Select your deployment target and configure your DevOps toolchain. After you click **Create**, the toolchain is created, and the deployment process is started automatically.

Deployment target

Cloud Foundry
IBM
Deploy and run your applications without managing servers or clusters. A Lite plan is available for quick and easy deployment.

IBM Cloud API key

IBM Cloud API key **New** 1

The value is required.

Getting started with apps

Step 1. Select the deployment target
Select your deployment target, and then provide the configuration information.

IBM Cloud Foundry
Cloud Foundry is the premier industry standard Platform-as-a-Service (PaaS) that ensures fast, easy, and reliable deployment of cloud-native apps. Cloud Foundry ensures that the build and deploy aspects of coding remain carefully coordinated with any attached services — resulting in quick, consistent and reliable iterating of applications. Cloud Foundry has a Lite plan that allows quick deployments for testing purposes.

Before you begin

- If your account doesn't have a Cloud Foundry org, you must create one. [Create org](#)

Steps

- Select the number of instances, memory allocation, region, org, and space.

After creating the API Key, Increase the Memory allocation per instance slider [1] to 256MB. If you do not increase the memory allocation, your Node-RED application might not have sufficient memory to run successfully. The Node-RED Starter kit only supports deployment to the Cloud Foundry space of IBM Cloud. Select the region [2] to deploy your application to. This should match the region you created your Cloudant instance in. Click Next [3].

Resource list / App details

Node RED KVHBI

IBM Cloud API key

Number of instances: 1

Memory allocation per instance

64 MB 2000 MB 256 1

Region: London 2 Organization: hashimtan99@gmail.com Space: dev

Host: node-red-kvhbi Domain: eu-gb.mybluemix.net

Cancel **Next** 3

If your account doesn't have a Cloud Foundry org, you must create one. [Create org](#)

Steps

- Select the number of instances, memory allocation, region, org, and space.
- Select the domain and provide a host name.

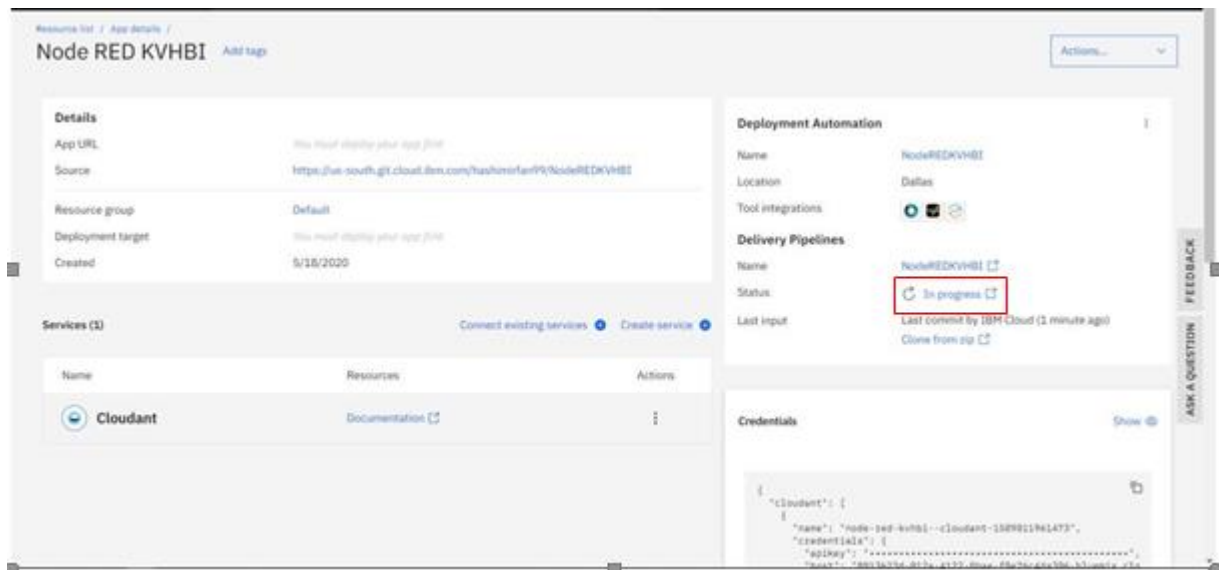
Now, Select the region [1] to create the DevOps toolchain and then Click Create [2].

The screenshot shows the 'Configure the DevOps toolchain' page for an application named 'Node RED KVHBI'. The page has two tabs: 'Select the deployment target' and 'Configure the DevOps toolchain', with the second tab being active. Below the tabs, there's a section titled 'Configure the DevOps toolchain' with the instruction: 'Give your toolchain a name and select the region to create your toolchain in.' A text input field for the toolchain name contains 'NodeREDKVHBI'. Below it, a 'Region' dropdown menu is set to 'Dallas', with a red '1' next to it. At the bottom left, there are 'Back' and 'Create' buttons, with a red '2' next to the 'Create' button. On the right side, there's a 'Getting started with apps' section with a list of steps: 1. Provide a name for your toolchain. 2. Select the region where your toolchain is created. 3. Select the resource group that has access to your new toolchain. 4. After you're finished with your selections, click Create.

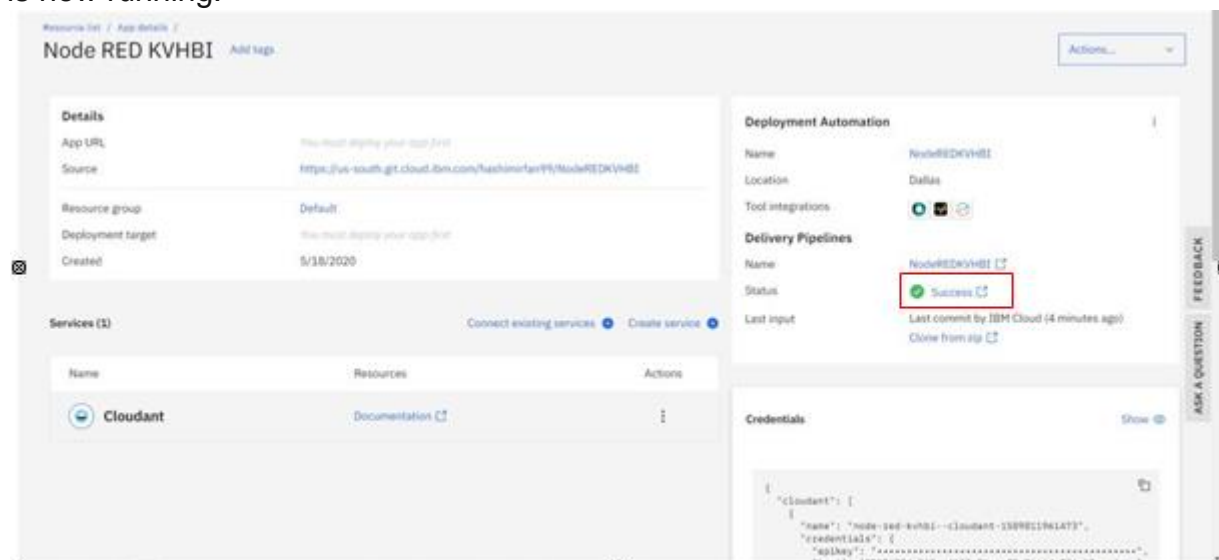
This will take you back to the application details page.

The screenshot shows the 'Node RED KVHBI' application details page. The page is divided into several sections. On the left, there's a 'Details' section with fields for 'App URL', 'Source', 'Resource group' (set to 'Default'), 'Deployment target', and 'Created' (1/18/2020). Below this is a 'Services (1)' section with a table showing one service named 'Cloudant'. On the right, there's a 'Deployment Automation' section with a 'Configure Continuous Delivery' button. Below that, there's a 'Credentials' section showing a list of credentials, including one named 'node-red-kvhbi-cloudant-13881394372'.

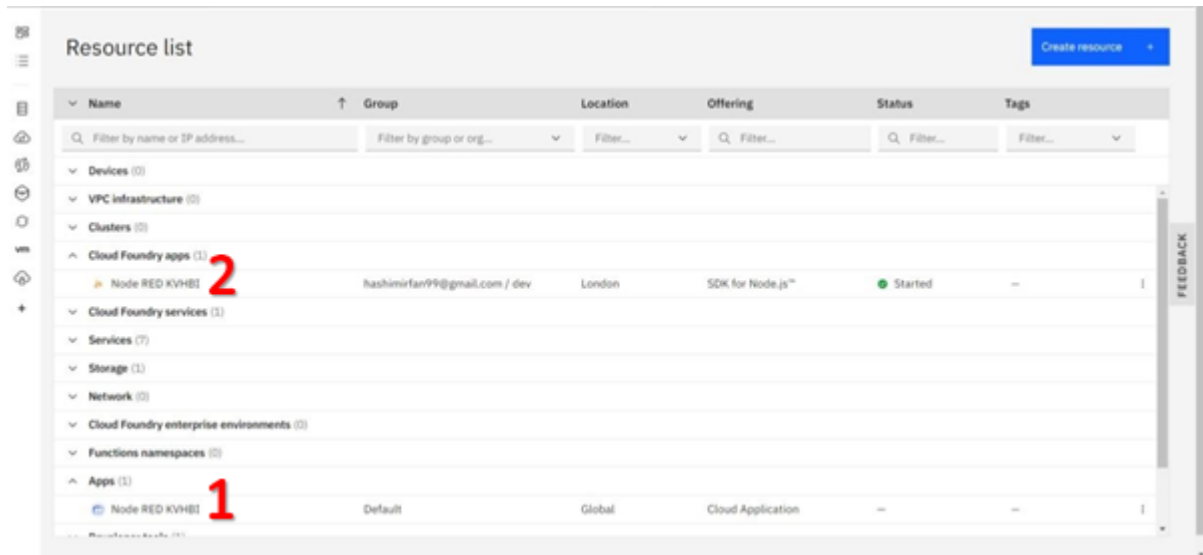
The Continuous Delivery section will refresh with the details of your newly created Toolchain. The Status field of the Delivery Pipeline will show In progress. That means your application is still being built and deployed.



The Deploy stage will take a few minutes to complete. Eventually the Deploy stage will go green to show it has passed. This means your Node-RED Starter application is now running.



Now that you've deployed your Node-RED application, let's open it up! Open your IBM Cloud Resource list. You will see your newly created Node-RED Application listed under the Apps section [1]. You will also see a corresponding entry under the Cloud Foundry apps section [2].

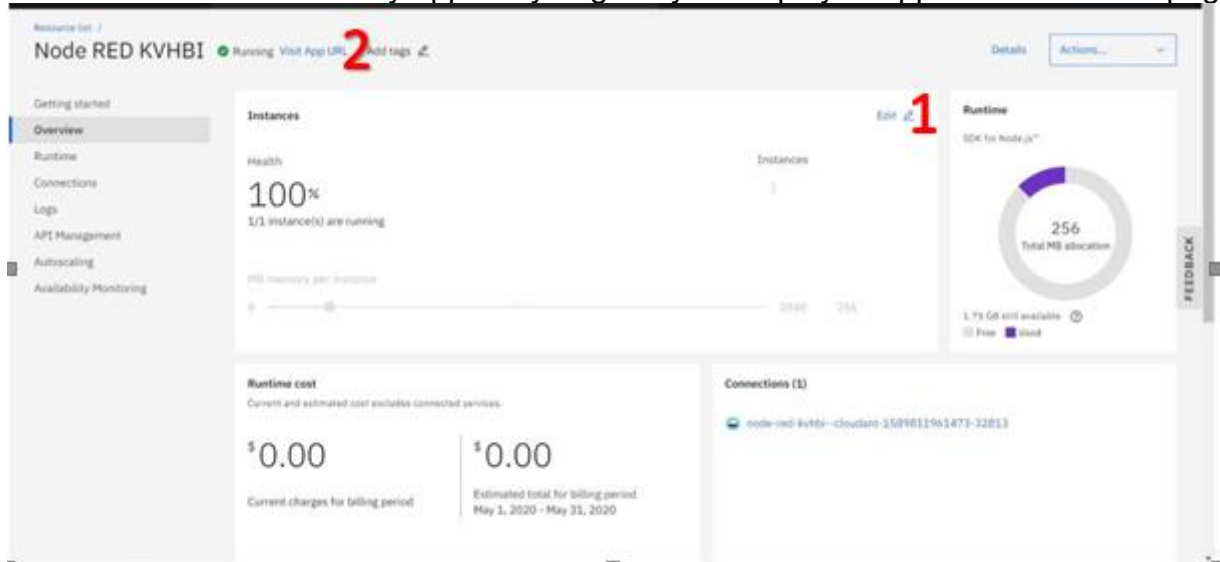


Resource list

Create resource

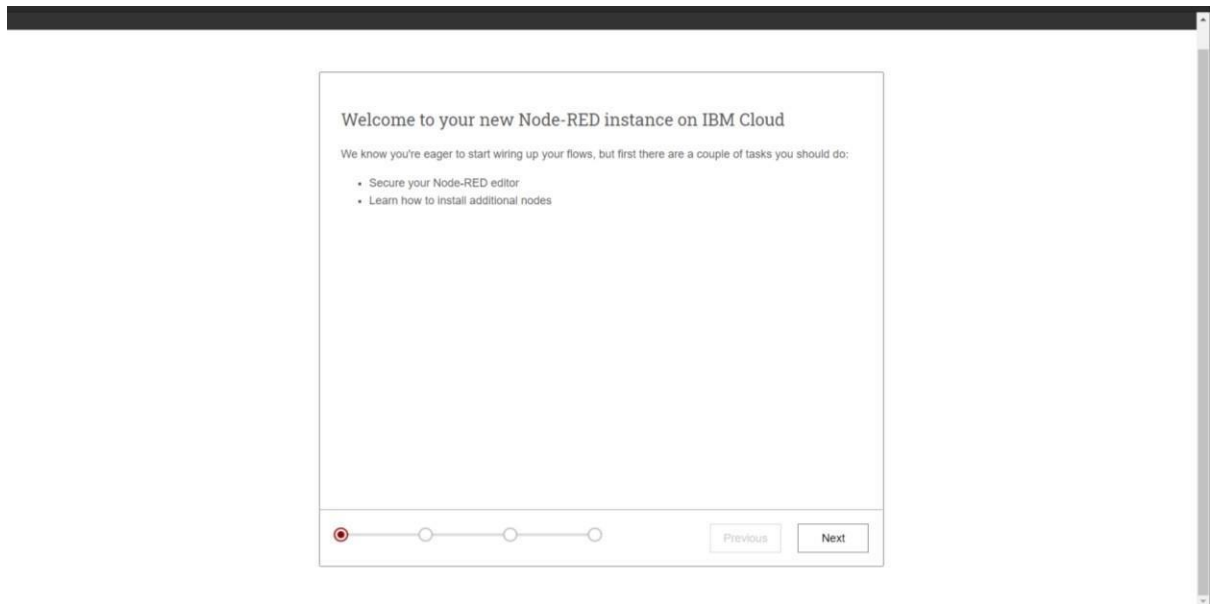
Name	Group	Location	Offering	Status	Tags
Filter by name or IP address... Filter by group or org... Filter... Filter... Filter...					
▼ Devices (0)					
▼ VPC Infrastructure (0)					
▼ Clusters (0)					
▲ Cloud Foundry apps (1)					
Node RED KVHBI	hashimifan99@gmail.com / dev	London	SDK for Node.js™	Started	—
▼ Cloud Foundry services (1)					
▼ Services (7)					
▼ Storage (1)					
▼ Network (0)					
▼ Cloud Foundry enterprise environments (0)					
▼ Functions namespaces (0)					
▲ Apps (1)					
Node RED KVHBI	Default	Global	Cloud Application	—	—

Click on this Cloud Foundry app entry to go to your deployed application's details page.

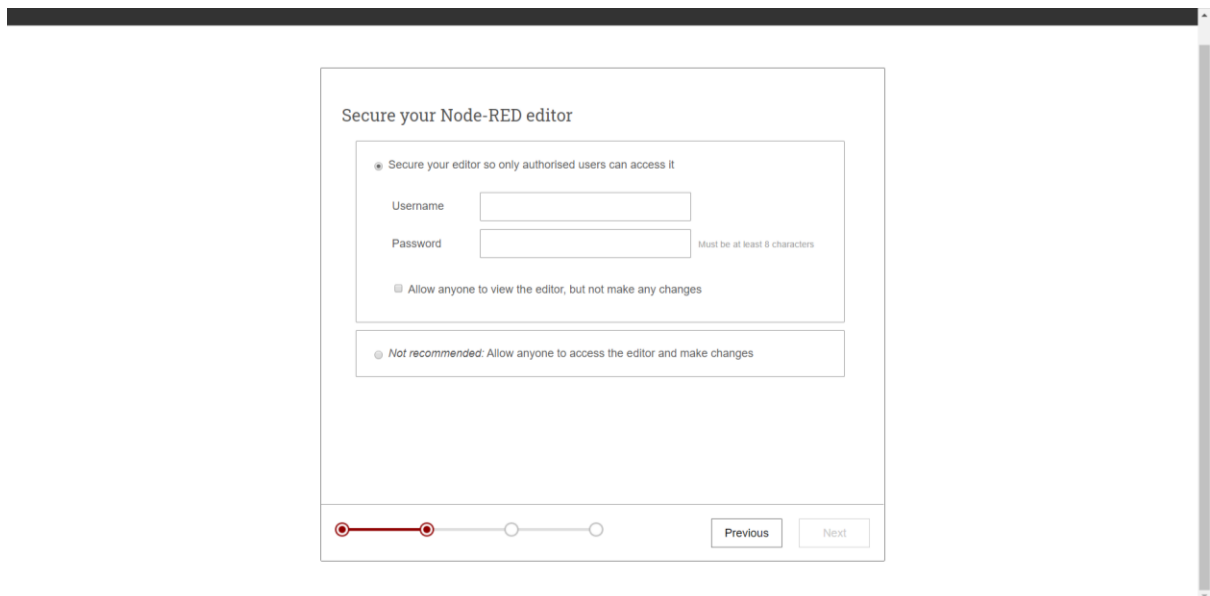


Special Cases: If your Runtime Instance is running full (0MB Free), Click on Edit[1] and reduce memory per instance to 128mb.

If you have Free space on your runtime skip the previous step[1] and Click on Visit App URL[2]



Click Next.



You can choose to Secure your Node-RED editor by providing a username and password. I am selecting the other option which is Allow anyone to access the editor and make changes.

Secure your Node-RED editor

☐ Secure your editor so only authorised users can access it

☒ *Not recommended:* Allow anyone to access the editor and make changes

Your editor will not be secured. Anyone with the URL will be able to access your flows, data and bound services.

☒ Tick this box to confirm you want your editor to be insecure



Previous

Next

Tick the box, and click Next.

Learn how to install additional nodes

Node-RED provides a **huge catalog of extra nodes** you can install into the editor.

Many of these nodes can be installed directly from the editor's palette manager feature. However that can cause issues due to the limited memory of the default Node-RED starter application.

The *recommended approach* is to edit your application's package.json file to include the additional node modules and then redeploy the application. This can be done using the Continuous Delivery feature on the application's IBM Cloud dashboard.

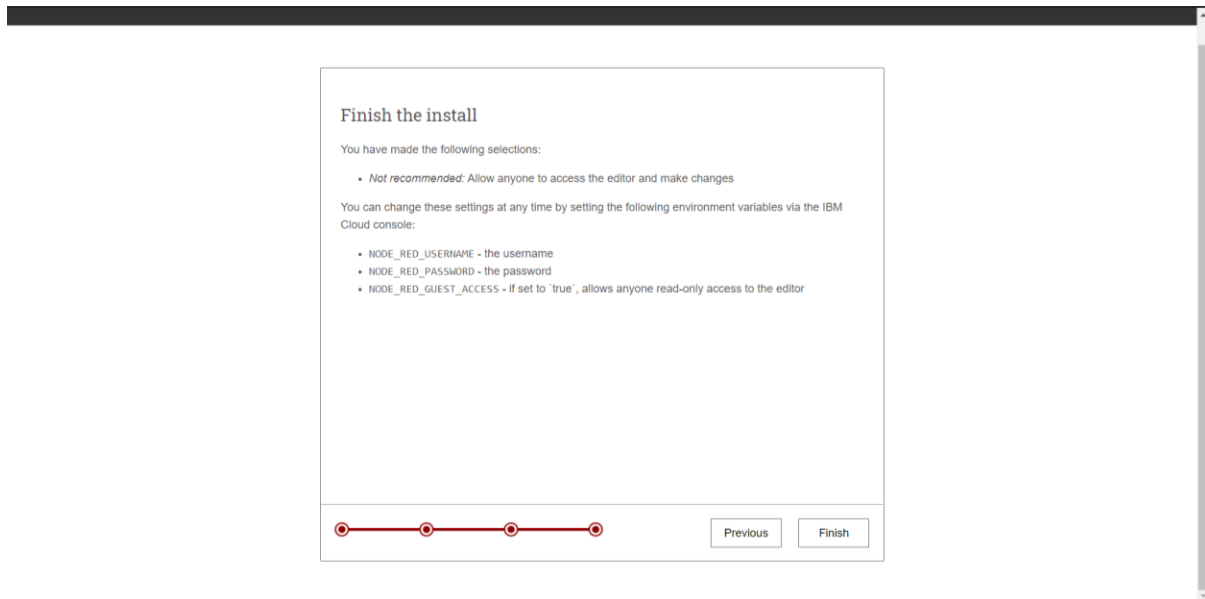
For more information, follow [this tutorial on IBM Developer](#).



Previous

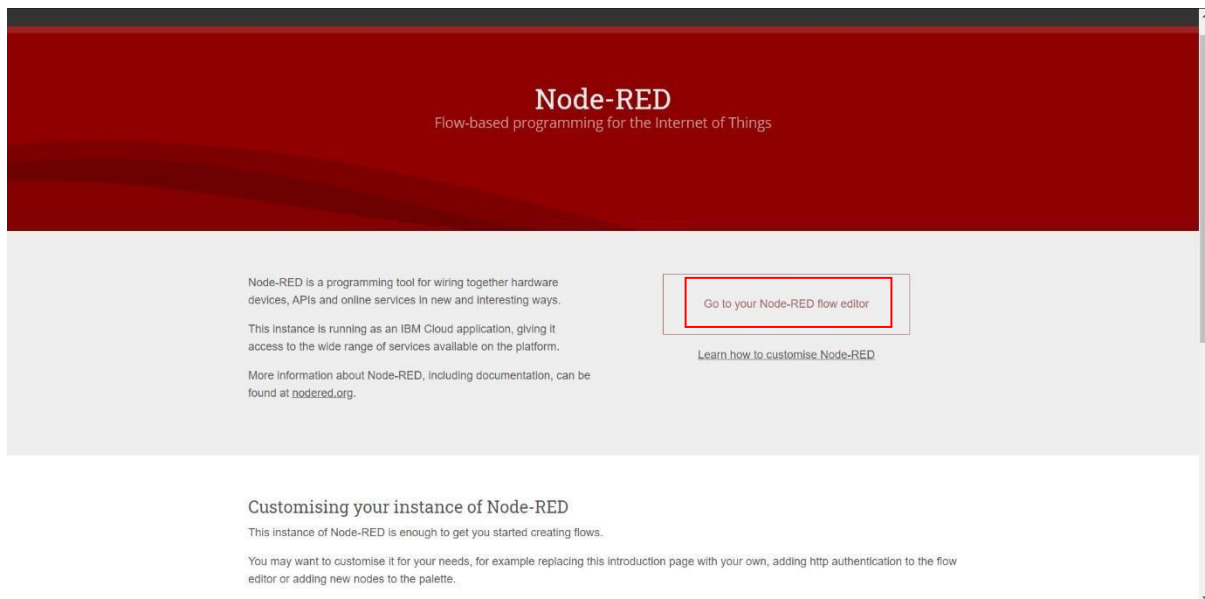
Next

Click Next.



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. 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.

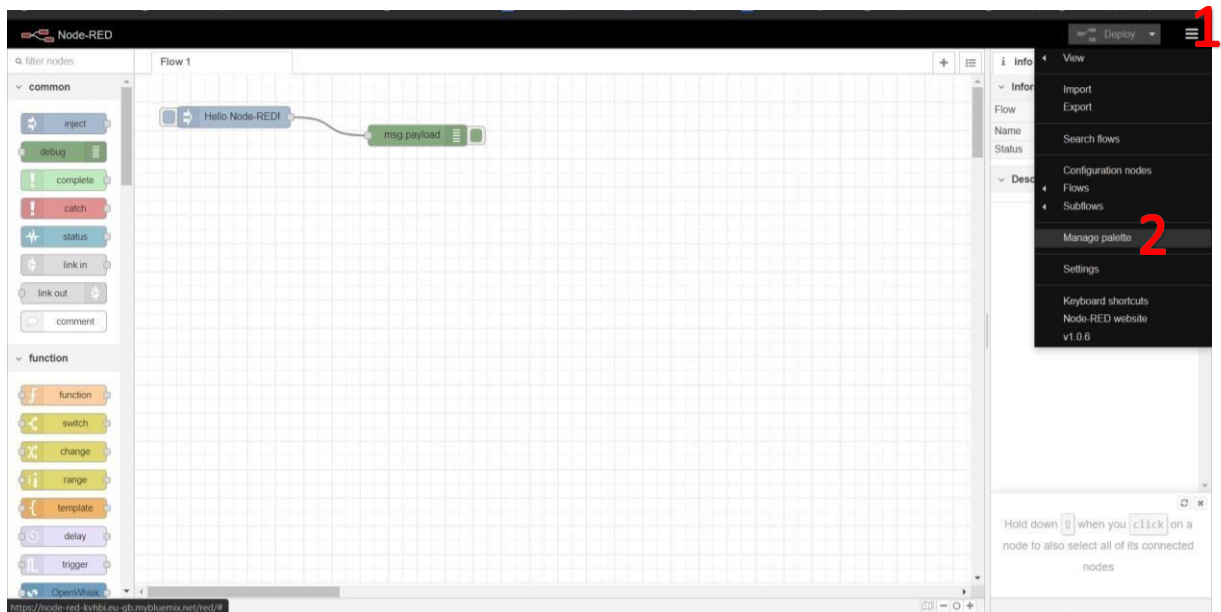




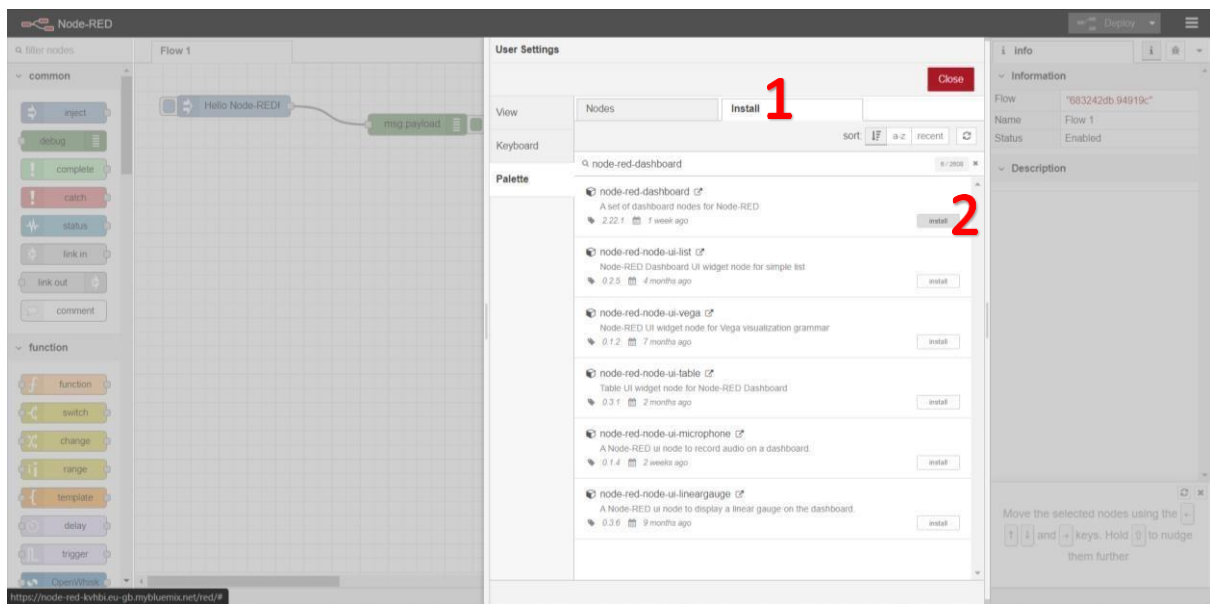
The Node-RED editor opens showing the default flow.

6. Configure the nodes and Build A Web Dashboard in Node-RED

To add Nodes to integrate Assistant, click [1] and then select Manage Palette [2]

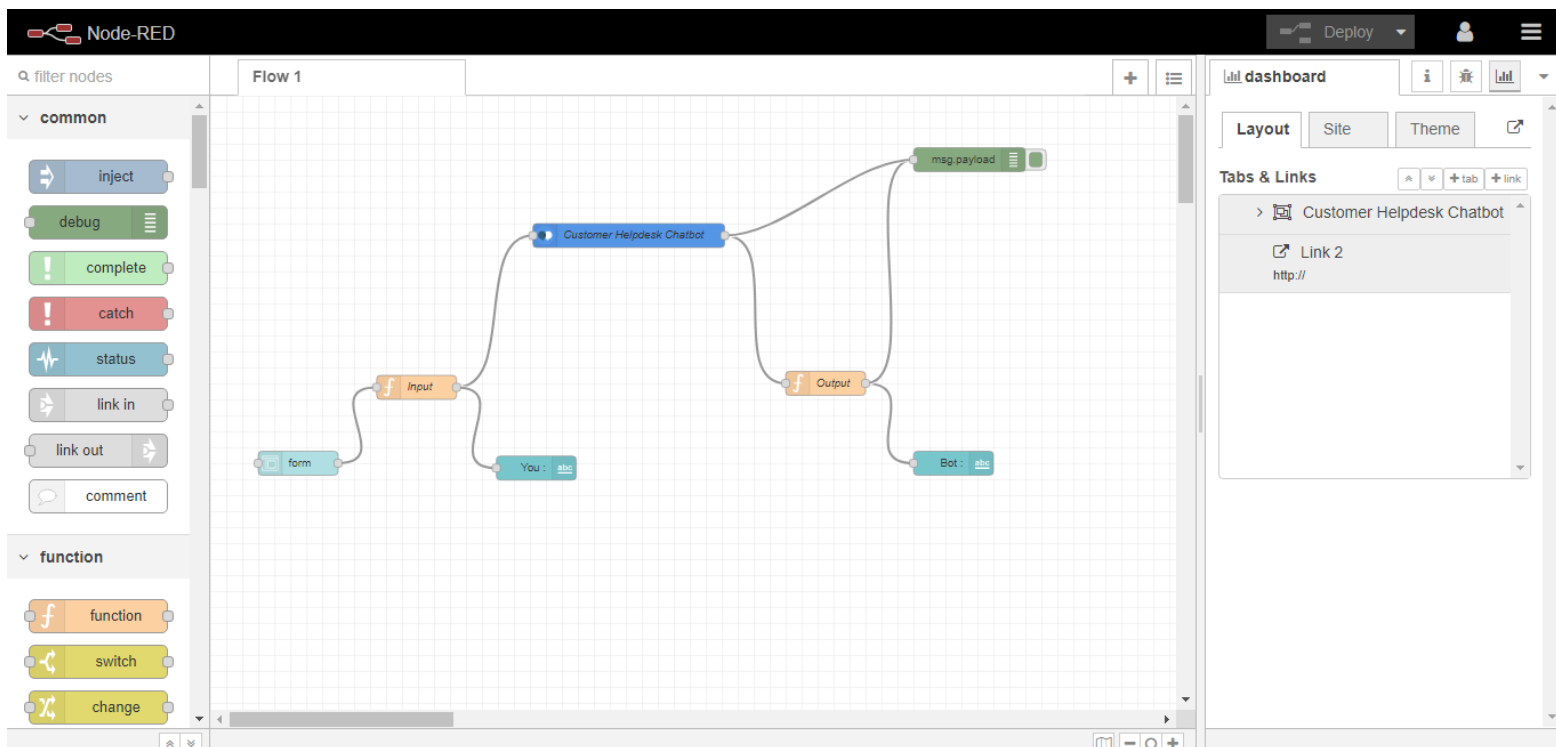


Go to Install Tab [1] and search for node-red-dashboard and Install [2] it.

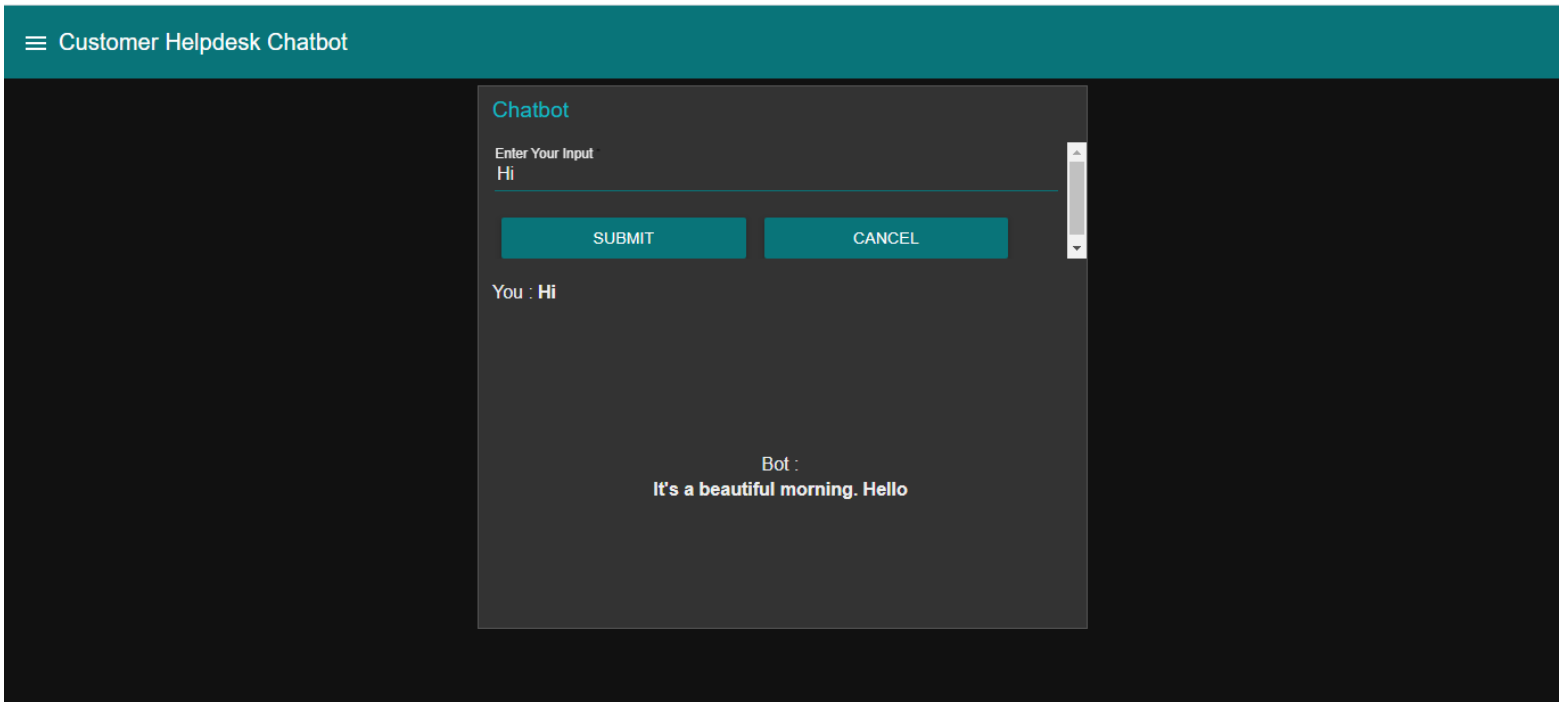


Using the nodes in the palette, Configure the required nodes and build web dashboard in Node-RED.

7. Deploy and Run the application

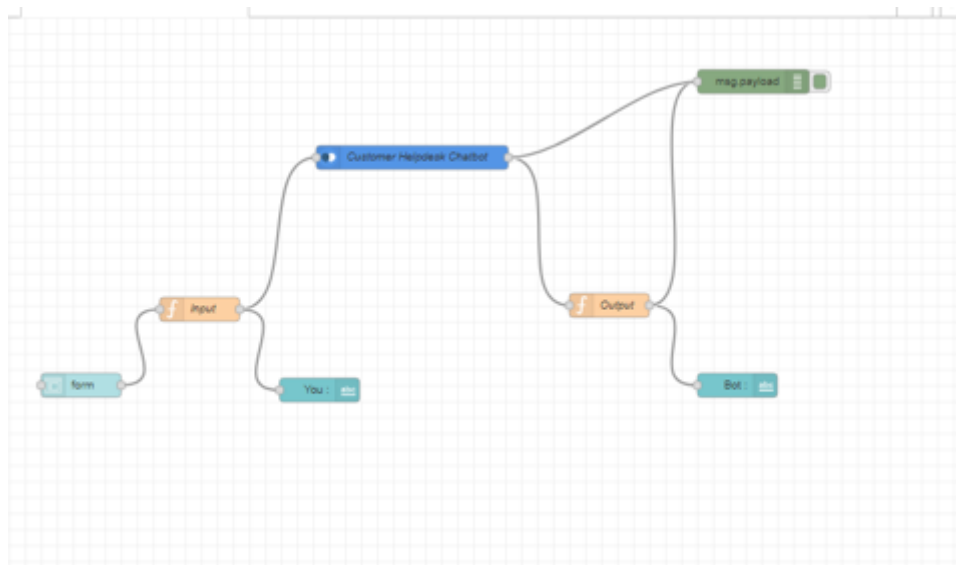


After Deploying the App, Run it.

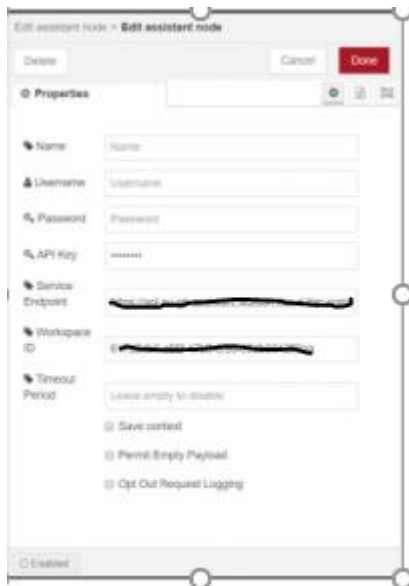


5.

FLOWCHART



First, Add a Form Node. Connect it with a function node and name it input parsing. To that add a Text Node and name it You. To the input parsing node, add assistant node.



Enter the API Key, Service Endpoint (URL) and Workspace ID (Skill ID) from Step 4 and click Done. To the assistant node add debug node. Add another function node to assistant node and name it parsing. Add text node to parsing node and name it Bot. Add another debug node to parsing node.

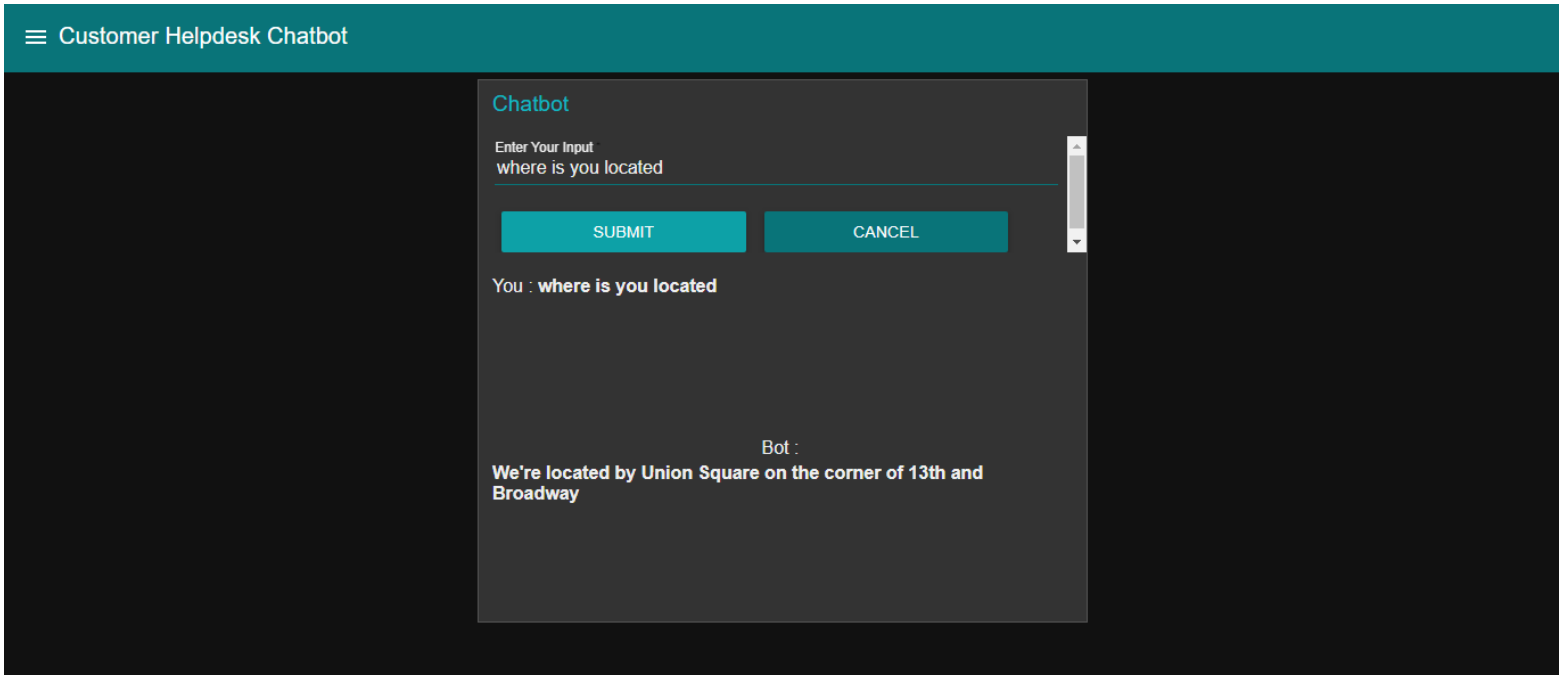
For Function Node named input, use the code below:

```
msg.payload=msg.payload.inpu  
t;  
return msg;
```

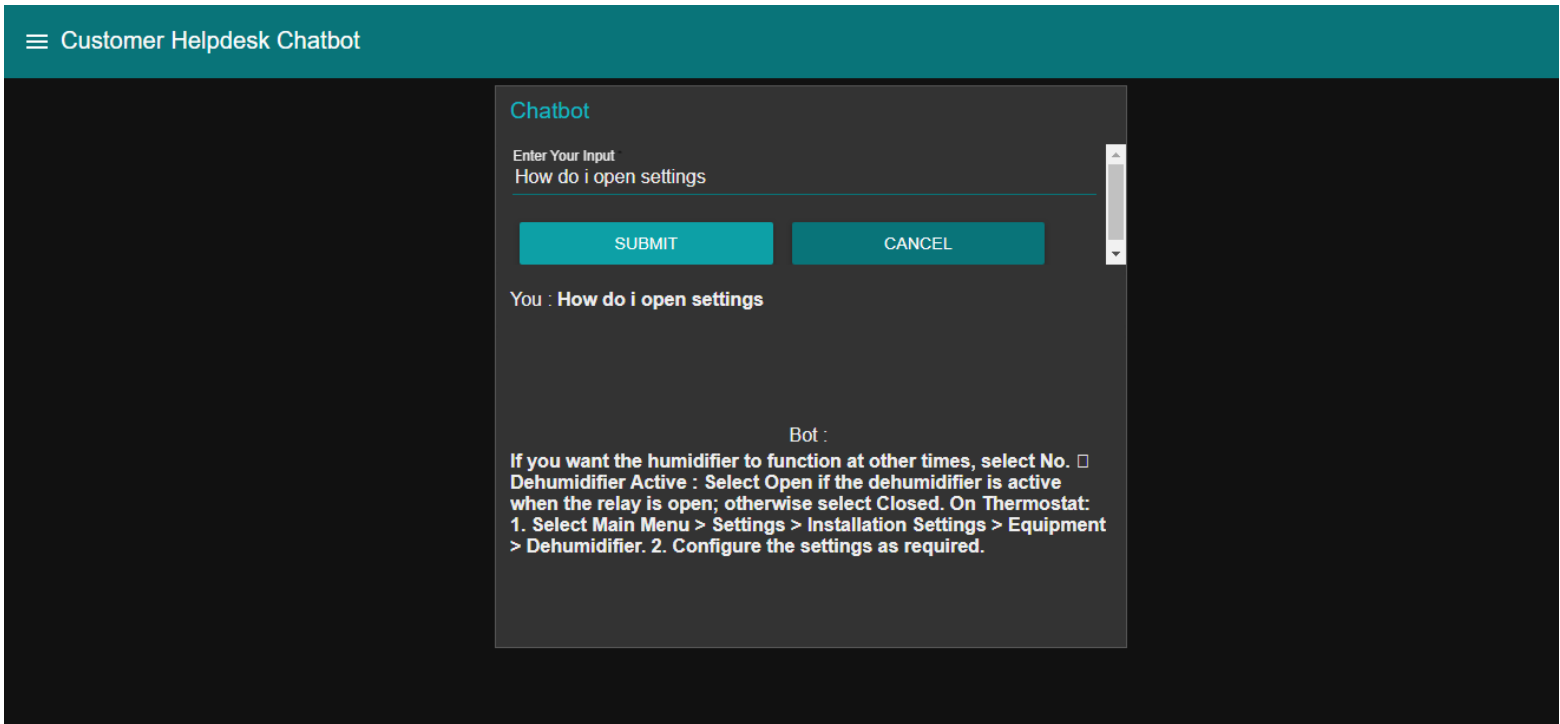
For Function Node named output, use the code below:

```
msg.payload = msg.payload.output.text[0];  
return msg;
```

(a)



(b)



Chatbot Link: <https://node-red-mychatbot.mybluemix.net/ui/#!/1?socketid=9GDTR-33kcXOot7GAAAT>

YouTube link(Project Explanation) - <https://youtu.be/0W7brW6QVJs>

YouTube link(Testimonial) - <https://youtu.be/z7TDI8AKoeQ>

7.

ADVANTAGES & DISADVANTAGES

Advantages:

- Faster Customer Service
- Increased Customer Satisfaction
- Lower Labor Costs
- Variety of Uses
- Data collection
- 24-7 availability
- Multiple Customer Handling

Disadvantages:

- Limited Responses for Customers
- Customers Could Become Frustrated
- Maintenance
- They aren't human
- Time-Consuming

8.

APPLICATIONS

A Product or Software Company Customer Help Desk

9.

CONCLUSION

An Intelligent Customer Helpdesk with Smart Document Understanding is made using various IBM Services like IBM Watson Discovery, IBM Watson and IBM Cloud Function.

10.

FUTURE SCOPE

A More Human Friendly Chatbot, or a personalized Chatbot is to be expected.

11.BIBILOGRAPHY

APPENDIX

A. Source Code

(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);
  });
});
}

```

(Watson Assistant Skill Code)

```

    {
      "intents": [
        {
          "intent": "Help",
          "examples": [
            {
              "text": "can you help"
            },
            {
              "text": "what can i say"
            },
            {
              "text": "what can i do"
            },
            {
              "text": "i need assistance"
            },
            {
              "text": "help me"
            },
            {
              "text": "help me decide"
            },
            {
              "text": "help"
            },
            {
              "text": "can you assist me"
            }
          ],
          "description": "Ask for help"
        },
        {
          "intent": "Customer_Care_Store_Hours",
          "examples": [
            {
              "text": "What time do you close"
            },
            {
              "text": "What time do you open on Saturdays"
            },
            {
              "text": "what time do you close on Sunday"
            }
          ],

```

```

{
  "text": "What time do you close today"
},
{
  "text": "Hours of operation"
},
{
  "text": "Are you open on Sunday"
},
{
  "text": "how late are you open"
},
{
  "text": "how late are you open tonight"
},
{
  "text": "are stores open on sunday"
},
{
  "text": "store open"
},
{
  "text": "store open hours?"
},
{
  "text": "What is the opening time for the washington store?"
},
{
  "text": "Are you closing early today"
},
{
  "text": "Are you closed new Year's eve"
},
{
  "text": "how early do you open on Saturdays"
},
{
  "text": "how late are you there"
},
{
  "text": "how late y'all stay up till"
},
{
  "text": "will you open on christmas"
},
{
  "text": "Are the stores open early?"
},
{
  "text": "Are you open during thanksgiving?"
},
{
  "text": "Are you open on bank holidays?"
},
{
  "text": "Are you open on sundays, and if so what are the hours?"
},
{
  "text": "At what hour can I swing by?"
},
{
  "text": "Can you tell me how late the stores are open till?"
},
{
  "text": "Does the store in the city center opens till 8pm on
weekends?"
},

```

```

{
  "text": "How early do you open?"
},
{
  "text": "How long are you open?"
},
{
  "text": "Is the branch open now?"
},
{
  "text": "What are the business hours of the store nearest to me?"
},
{
  "text": "What are the hours of operation?"
},
{
  "text": "What are the saturday opening times for the local store?"
},
{
  "text": "What are ur opening hours?"
},
{
  "text": "store open now"
},
{
  "text": "open hours store"
},
{
  "text": "when do your stores open"
},
{
  "text": "when does the store close"
},
{
  "text": "when can i visit your store"
},
{
  "text": "store hrs"
},
{
  "text": "store hours"
},
{
  "text": "What are your hous?"
},
{
  "text": "What time does the central manchester store shut on a
saturday?"
},
{
  "text": "What time do stores close?"
},
{
  "text": "What time is your store open on saturday?"
},
{
  "text": "what are your hours"
},
{
  "text": "will you open for christmas"
},
{
  "text": "will you be open Memorial day"
},
{
  "text": "when do you close"
},

```

```

        {
          "text": "What time are you closing today?"
        }
      ],
      "description": "Find business hours."
    },
    {
      "intent": "Thanks",
      "examples": [
        {
          "text": "thank you very much"
        },
        {
          "text": "thx"
        },
        {
          "text": "i appreciate it"
        },
        {
          "text": "thank you"
        },
        {
          "text": "thanks"
        },
        {
          "text": "many thanks"
        },
        {
          "text": "much appreciated"
        },
        {
          "text": "that's nice of you"
        }
      ],
      "description": "Thanks"
    },
    {
      "intent": "Customer_Care_Store_Location",
      "examples": [
        {
          "text": "Where is your office?"
        },
        {
          "text": "please suggest route from times square"
        },
        {
          "text": "how can i get to you from grand central"
        },
        {
          "text": "which cross streets are you on"
        },
        {
          "text": "give me directions"
        },
        {
          "text": "what's your location"
        },
        {
          "text": "where are you"
        },
        {
          "text": "what is the address"
        },
        {
          "text": "how do i find you"
        }
      ],
    }
  ]
}

```

```

        "text": "location please"
    },
    {
        "text": "can you give me directions"
    },
    {
        "text": "where are you located"
    },
    {
        "text": "how do i get to your place"
    },
    {
        "text": "Find store"
    },
    {
        "text": "Where is?"
    },
    {
        "text": "Where are you located?"
    },
    {
        "text": "how do i get to your business"
    },
    {
        "text": "Go to your company"
    },
    {
        "text": "I'd like to go to a store"
    },
    {
        "text": "I need help with find a store"
    },
    {
        "text": "I want to know about a store"
    },
    {
        "text": "Looking for a location"
    },
    {
        "text": "What is the closest store to my address?"
    },
    {
        "text": "What is the nearest branch?"
    },
    {
        "text": "What is the store near my zip code?"
    }
],
"description": "Locate a physical store location or an address."
},
{
    "intent": "Goodbye",
    "examples": [
        {
            "text": "so long"
        },
        {
            "text": "good bye"
        },
        {
            "text": "see ya"
        },
        {
            "text": "arrivederci"
        },
        {
            "text": "ciao"
        }
    ]
}

```

```

    },
    {
      "text": "bye"
    }
  ],
  "description": "Good byes"
},
{
  "intent": "Customer_Care_Appointments",
  "examples": [
    {
      "text": "are you available on tuesday"
    },
    {
      "text": "can i book for tonight"
    },
    {
      "text": "do you have availability next week"
    },
    {
      "text": "can i make an appointment"
    },
    {
      "text": "can you make an appointment for me"
    },
    {
      "text": "i'd like to make an appointment"
    },
    {
      "text": "What time can I meet the staff?"
    },
    {
      "text": "When can I meet with one of your employees at your store?"
    },
    {
      "text": "Want to change my visit"
    },
    {
      "text": "Store appointment"
    },
    {
      "text": "Set up an appt"
    },
    {
      "text": "Make an appointment"
    },
    {
      "text": "i'd like to come in for an appointment"
    },
    {
      "text": "I prefer a face to face visit"
    },
    {
      "text": "meet in store"
    },
    {
      "text": "I want to talk in person with someone about my case"
    },
    {
      "text": "I would like to discuss my situation face to face"
    },
    {
      "text": "I would like to make an appointment to visit the nearest
store to my location."
    },
    {
      "text": "Could I speak to someone in the store next tuesday?"
    }
  ]
}

```

```

    },
    {
      "text": "Can I book an in person session"
    }
  ],
  "description": "Schedule or manage an in-store appointment."
},
{
  "intent": "Product_Information",
  "examples": [
    {
      "text": "How do i access the settings"
    },
    {
      "text": "How do i set the time"
    },
    {
      "text": "How do i turn on the heater"
    },
    {
      "text": "How to install battery"
    }
  ],
  "description": ""
},
{
  "intent": "Cancel",
  "examples": [
    {
      "text": "i don't want a table anymore anymore"
    },
    {
      "text": "never mind"
    },
    {
      "text": "forget it"
    },
    {
      "text": "nevermind"
    },
    {
      "text": "cancel that"
    },
    {
      "text": "i changed my mind"
    },
    {
      "text": "cancel the request"
    }
  ],
  "description": "Cancel the current request"
},
{
  "intent": "General_Connect_to_Agent",
  "examples": [
    {
      "text": "I need to speak to a representative. How would I go about
doing so?"
    },
    {
      "text": "Can I speak to an advisor?"
    },
    {
      "text": "Can I speak with somebody?"
    },
    {
      "text": "Can I talk to someone?"
    }
  ]
}

```



```

    },
    {
      "text": "Can you assist me to connect to an agent?"
    },
    {
      "text": "Can you connect me with a real person?"
    },
    {
      "text": "Connect me to a live operator please."
    },
    {
      "text": "Contact person"
    },
    {
      "text": "Could you please transfer me to your master?"
    },
    {
      "text": "Customer service representative please."
    },
    {
      "text": "Do not want a robot?"
    },
    {
      "text": "Hi can you transfer me"
    },
    {
      "text": "How can I skip the recorded menu and go straight to a live
person?"
    },
    {
      "text": "I don't want to talk to a bot."
    },
    {
      "text": "I don't want to talk to you"
    },
    {
      "text": "Is there anyone there I can actually talk to for real?"
    },
    {
      "text": "I want agent"
    },
    {
      "text": "I want a manager"
    },
    {
      "text": "I want an agent to help me"
    },
    {
      "text": "I want to talk to a person"
    },
    {
      "text": "I would like to speak to a human"
    },
    {
      "text": "I would like to speak to someone"
    },
    {
      "text": "Need help from human"
    },
    {
      "text": "Please assist me to get to an agent"
    },
    {
      "text": "Operator please"
    },
    {
      "text": "Please connect me to a live agent"
    }

```

```

    },
    {
      "text": "Please let me talk to a human being."
    },
    {
      "text": "Pls connect"
    },
    {
      "text": "Put me through to someone"
    },
    {
      "text": "representative"
    },
    {
      "text": "I want to speak to a person"
    },
    {
      "text": "I dont want to talk to a computer"
    },
    {
      "text": "call the manager"
    },
    {
      "text": "I want to speak to a human"
    },
    {
      "text": "I want to talk to the manager"
    },
    {
      "text": "A real agent, please."
    },
    {
      "text": "Call agent"
    },
    {
      "text": "talk to a human"
    },
    {
      "text": "Agent help"
    },
    {
      "text": "Yes, take me to a real person"
    },
    {
      "text": "Where is the closest agent?"
    },
    {
      "text": "Send me to an agent"
    },
    {
      "text": "I don't want to speak with a robot"
    },
    {
      "text": "get me a person"
    },
    {
      "text": "Can I connect to an agent?"
    },
    {
      "text": "Can I speak to a human please?"
    },
    {
      "text": "Can I speak to a live person?"
    }
  ],
  "description": "Request a human agent."
},

```

```
{
  "intent": "General_Greetings",
  "examples": [
    {
      "text": "What's up?"
    },
    {
      "text": "Who is this?"
    },
    {
      "text": "You there"
    },
    {
      "text": "How is it going?"
    },
    {
      "text": "Hey you"
    },
    {
      "text": "Looking good eve"
    },
    {
      "text": "How r u?"
    },
    {
      "text": "hi"
    },
    {
      "text": "How have you been?"
    },
    {
      "text": "How are you today?"
    },
    {
      "text": "How are things going?"
    },
    {
      "text": "yo"
    },
    {
      "text": "hiya"
    },
    {
      "text": "Good day"
    },
    {
      "text": "Good evening"
    },
    {
      "text": "Good morning"
    },
    {
      "text": "Good to see you"
    },
    {
      "text": "Greetings"
    },
    {
      "text": "Have you been well?"
    },
    {
      "text": "Ok take me back"
    },
    {
      "text": "Hi advisor"
    },
    {

```

```

        "text": "Hey twin"
    },
    {
        "text": "Hi there"
    },
    {
        "text": "Hello Agent"
    },
    {
        "text": "Hello"
    },
    {
        "text": "Hello I am looking for some help here"
    },
    {
        "text": "Hey how are you doing"
    },
    {
        "text": "Hey there all"
    },
    {
        "text": "What's new?"
    },
    {
        "text": "Hey there"
    }
],
"description": "Greetings"
}
],
"entities": [
    {
        "entity": "sys-date",
        "values": []
    },
    {
        "entity": "specialist",
        "values": [
            {
                "type": "synonyms",
                "value": "Maria",
                "synonyms": []
            },
            {
                "type": "synonyms",
                "value": "Brenda",
                "synonyms": []
            },
            {
                "type": "synonyms",
                "value": "Barbara",
                "synonyms": [
                    "barbra"
                ]
            },
            {
                "type": "synonyms",
                "value": "Robert",
                "synonyms": [
                    "bob"
                ]
            },
            {
                "type": "synonyms",
                "value": "Nicholas",
                "synonyms": [
                    "nick"
                ]
            }
        ]
    }
]

```

```

    ]
  },
  {
    "type": "synonyms",
    "value": "Derrik",
    "synonyms": [
      "derek",
      "derik",
      "derrik",
      "derrick"
    ]
  }
]
},
{
  "entity": "holiday",
  "values": [
    {
      "type": "synonyms",
      "value": "new years eve",
      "synonyms": [
        "12-31",
        "12/31",
        "dec 31",
        "dec 31st",
        "new year's eve"
      ]
    },
    {
      "type": "synonyms",
      "value": "labor day",
      "synonyms": []
    },
    {
      "type": "synonyms",
      "value": "independence day",
      "synonyms": [
        "7/4",
        "fourth of july",
        "july 4",
        "july 4th",
        "july fourth"
      ]
    },
    {
      "type": "synonyms",
      "value": "valentine's day",
      "synonyms": [
        "valentine day",
        "valentines day"
      ]
    },
    {
      "type": "synonyms",
      "value": "christmas eve",
      "synonyms": [
        "x mas eve",
        "x-mas eve",
        "xmas eve"
      ]
    },
    {
      "type": "synonyms",
      "value": "halloween",
      "synonyms": []
    }
  ],
  {

```

```

        "type": "synonyms",
        "value": "christmas",
        "synonyms": [
            "christmas day",
            "x man day",
            "xmas",
            "x mas",
            "x-mas",
            "x-mas day",
            "xmas day"
        ]
    },
    {
        "type": "synonyms",
        "value": "thanksgiving",
        "synonyms": [
            "turkey day"
        ]
    },
    {
        "type": "synonyms",
        "value": "memorial day",
        "synonyms": []
    },
    {
        "type": "synonyms",
        "value": "new years",
        "synonyms": [
            "1/1",
            "jan 1",
            "jan 1st",
            "jan first",
            "january 1",
            "january 1st",
            "january first",
            "new year",
            "new year day",
            "new years day"
        ]
    }
]
},
{
    "entity": "zip_code",
    "values": [
        {
            "type": "patterns",
            "value": "US Zip",
            "patterns": [
                "(\\b|\\s)\\d{5}(\\b|\\s)"
            ]
        }
    ]
},
{
    "entity": "phone",
    "values": [
        {
            "type": "patterns",
            "value": "US Phone pattern",
            "patterns": [
                "(\\d{3})-(\\d{3})-(\\d{4})"
            ]
        }
    ]
},
{

```

```

    "entity": "sys-number",
    "values": []
  },
  {
    "entity": "reply",
    "values": [
      {
        "type": "synonyms",
        "value": "no",
        "synonyms": [
          "definitely not",
          "don't think so",
          "dont think so",
          "i think not",
          "nope",
          "not at this time",
          "not now"
        ]
      },
      {
        "type": "synonyms",
        "value": "yes",
        "synonyms": [
          "definitely",
          "go for it",
          "let's do it",
          "ok",
          "please",
          "sure",
          "why not",
          "yeah",
          "yes",
          "you bet",
          "you betcha",
          "yep"
        ]
      }
    ]
  },
  {
    "entity": "landmark",
    "values": [
      {
        "type": "synonyms",
        "value": "times square",
        "synonyms": [
          "time sqaure",
          "time square",
          "times sqaure"
        ]
      },
      {
        "type": "synonyms",
        "value": "empire state building",
        "synonyms": [
          "empire state",
          "emprire state"
        ]
      },
      {
        "type": "synonyms",
        "value": "grand central",
        "synonyms": []
      }
    ]
  },
  "fuzzy_match": true
},

```

```

    {
      "entity": "sys-time",
      "values": []
    }
  ],
  "metadata": {
    "api_version": {
      "major_version": "v2",
      "minor_version": "2018-11-08"
    }
  },
  "webhooks": [
    {
      "url": "https://us-
south.functions.cloud.ibm.com/api/v1/web/SI05202000978%40smartinternz.com_dev/
default/Discovery-Chatbot-Function.json",
      "name": "main_webhook",
      "headers": []
    }
  ],
  "dialog_nodes": [
    {
      "type": "event_handler",
      "output": {},
      "parent": "slot_102_1498132501942",
      "context": {
        "date": "@sys-date"
      },
      "metadata": {},
      "conditions": "@sys-date",
      "event_name": "input",
      "dialog_node": "handler_103_1498132501942",
      "previous_sibling": "handler_104_1498132501942"
    },
    {
      "type": "event_handler",
      "output": {
        "text": {
          "values": [
            "Looks like you're trying to make a reservation in the past. Try
again."
          ]
        }
      },
      "parent": "slot_102_1498132501942",
      "metadata": {},
      "next_step": {
        "behavior": "reprompt"
      },
      "conditions": "$date.before(now())",
      "event_name": "filled",
      "dialog_node": "handler_6_1509695999145",
      "previous_sibling": "handler_103_1498132501942"
    },
    {
      "type": "event_handler",
      "output": {
        "text": "What day would you like to come in?"
      },
      "parent": "slot_102_1498132501942",
      "metadata": {},
      "event_name": "focus",
      "dialog_node": "handler_104_1498132501942"
    },
    {
      "type": "response_condition",
      "output": {

```



```

    "generic": [
      {
        "values": [
          {
            "text": "Please try again later..."
          }
        ],
        "response_type": "text",
        "selection_policy": "sequential"
      }
    ],
    "parent": "node_6_1591626152244",
    "conditions": "anything_else",
    "dialog_node": "response_7_1591626363500",
    "previous_sibling": "response_5_1591626359813"
  },
  {
    "type": "response_condition",
    "output": {
      "generic": [
        {
          "values": [
            {
              "text": "<?$webhook_result_1.passages[0].passage_text?"
            }
          ],
          "response_type": "text",
          "selection_policy": "sequential"
        }
      ]
    },
    "parent": "node_6_1591626152244",
    "conditions": "$webhook_result_1",
    "dialog_node": "response_5_1591626359813"
  },
  {
    "type": "slot",
    "output": {},
    "parent": "Reservation using slots",
    "metadata": {
      "_customization": {
        "mcr": true
      }
    },
    "variable": "$time",
    "dialog_node": "slot_105_1498132552870",
    "previous_sibling": "slot_102_1498132501942"
  },
  {
    "type": "event_handler",
    "output": {
      "text": {
        "values": []
      }
    },
    "parent": "Reservation using slots",
    "disabled": true,
    "metadata": {},
    "event_name": "focus",
    "dialog_node": "handler_7_1509696539866",
    "previous_sibling": "handler_16_1509133697261"
  },
  {
    "type": "event_handler",
    "output": {
      "text": {

```

```

        "values": [
            "OK. Canceling your request..."
        ]
    },
    "parent": "Reservation using slots",
    "context": {
        "date": null,
        "time": null,
        "phone": null,
        "confirm": null,
        "specialist": null,
        "user_cancelled": true
    },
    "metadata": {},
    "next_step": {
        "behavior": "skip_all_slots"
    },
    "conditions": "#Cancel",
    "event_name": "generic",
    "dialog_node": "handler_16_1509133697261",
    "previous_sibling": "handler_3_1501275087289"
},
{
    "type": "response_condition",
    "output": {
        "text": {
            "values": [
                "Let me know how else I can help"
            ],
            "selection_policy": "sequential"
        }
    },
    "parent": "Reservation using slots",
    "context": {},
    "metadata": {},
    "conditions": "$user_cancelled",
    "dialog_node": "node_10_1509697567474",
    "previous_sibling": "node_25_1522598839584"
},
{
    "type": "slot",
    "output": {},
    "parent": "Reservation using slots",
    "metadata": {
        "_customization": {}
    },
    "variable": "$phone",
    "dialog_node": "slot_22_1522444583114",
    "previous_sibling": "slot_8_1509132875735"
},
{
    "type": "response_condition",
    "output": {
        "text": {
            "values": [
                "[Use IBM Cloud Functions to connect to to backend systems]"
            ]
        }
    },
    "parent": "Reservation using slots",
    "metadata": {},
    "conditions": "$user_needs_help",
    "dialog_node": "node_25_1522598839584",
    "previous_sibling": "handler_7_1509696539866"
},
{

```

```

    "type": "slot",
    "output": {},
    "parent": "Reservation using slots",
    "metadata": {},
    "variable": "$specialist",
    "dialog_node": "slot_12_1522596437268",
    "previous_sibling": "slot_105_1498132552870"
  },
  {
    "type": "slot",
    "output": {},
    "parent": "Reservation using slots",
    "metadata": {
      "_customization": {
        "mcr": true
      }
    },
    "variable": "$date",
    "dialog_node": "slot_102_1498132501942",
    "previous_sibling": "node_3_1519173961259"
  },
  {
    "type": "response_condition",
    "output": {
      "text": {
        "values": [
          "Let me check availability... [Use IBM Cloud Functions to connect
to backend systems]"
        ]
      }
    },
    "parent": "Reservation using slots",
    "context": {},
    "metadata": {},
    "conditions": "true",
    "dialog_node": "node_3_1519173961259",
    "previous_sibling": "node_10_1509697567474"
  },
  {
    "type": "slot",
    "output": {},
    "parent": "Reservation using slots",
    "metadata": {},
    "variable": "$confirm",
    "dialog_node": "slot_8_1509132875735",
    "previous_sibling": "slot_12_1522596437268"
  },
  {
    "type": "event_handler",
    "output": {
      "text": {
        "values": [
          "I see you need help making an appointment. Let me transfer you to
an agent..."
        ]
      },
      "selection_policy": "sequential"
    },
    "parent": "Reservation using slots",
    "context": {
      "date": null,
      "time": null,
      "phone": null,
      "confirm": null,
      "specialist": null,
      "user_needs_help": true
    }
  },

```

```

    "metadata": {},
    "next_step": {
      "behavior": "skip_all_slots"
    },
    "conditions": "#Help",
    "event_name": "generic",
    "dialog_node": "handler_3_1501275087289"
  },
  {
    "type": "standard",
    "output": {
      "text": "OK. Let me know how I can help"
    },
    "parent": "node_22_1467833484410",
    "metadata": {},
    "conditions": "@reply:no",
    "dialog_node": "node_21_1468350173406",
    "previous_sibling": "node_19_1468350024009"
  },
  {
    "type": "standard",
    "output": {
      "text": {
        "values": [
          "OK. Transferring... [Use IBM Cloud Functions to connect to backend systems]"
        ]
      }
    },
    "parent": "node_22_1467833484410",
    "metadata": {},
    "conditions": "@reply:yes",
    "dialog_node": "node_19_1468350024009"
  },
  {
    "type": "event_handler",
    "output": {},
    "parent": "slot_8_1509132875735",
    "context": {
      "confirm": "@reply && slot_in_focus"
    },
    "metadata": {},
    "conditions": "@reply && slot_in_focus",
    "event_name": "input",
    "dialog_node": "handler_9_1509132875735",
    "previous_sibling": "handler_10_1509132875735"
  },
  {
    "type": "event_handler",
    "output": {
      "text": {
        "values": [
          "Perfect!"
        ]
      }
    },
    "parent": "slot_8_1509132875735",
    "metadata": {},
    "conditions": "@reply:yes",
    "event_name": "filled",
    "dialog_node": "handler_14_1509133469904",
    "previous_sibling": "handler_9_1509132875735"
  },
  {
    "type": "event_handler",
    "output": {
      "text": {

```

```

        "values": [
            "Sorry... let's try again"
        ]
    },
    "parent": "slot_8_1509132875735",
    "context": {
        "date": null,
        "time": null,
        "confirm": null
    },
    "metadata": {},
    "conditions": "@reply:no",
    "event_name": "filled",
    "dialog_node": "handler_17_1509135162089",
    "previous_sibling": "handler_14_1509133469904"
},
{
    "type": "event_handler",
    "output": {
        "text": "Let me confirm: You want an appointment for <?
$date.reformatDateTime(\"EEEEEE\") ?> at <? $time.reformatDateTime(\"h a\") ?>.
Is this correct?"
    },
    "parent": "slot_8_1509132875735",
    "metadata": {},
    "event_name": "focus",
    "dialog_node": "handler_10_1509132875735"
},
{
    "type": "response_condition",
    "output": {
        "text": {
            "values": [
                "Our hours on <? @sys-date.reformatDateTime(\"EEEEEE\") ?> are 11am
to 6pm."
            ],
            "selection_policy": "sequential"
        }
    },
    "parent": "Hours of Operation",
    "context": {},
    "metadata": {},
    "conditions": "@sys-date.reformatDateTime(\"EEEEEE\") == \"Saturday\" ||
@sys-date.reformatDateTime(\"EEEEEE\") == \"Sunday\"",
    "dialog_node": "node_2_1482424204936",
    "previous_sibling": "node_5_1482426503106"
},
{
    "type": "response_condition",
    "output": {
        "text": {
            "values": [
                "Our hours are Monday to Friday 10am to 8pm and Friday and
Saturday 11am to 6pm."
            ],
            "selection_policy": "sequential"
        }
    },
    "parent": "Hours of Operation",
    "context": {},
    "metadata": {},
    "conditions": " true",
    "dialog_node": "node_6_1482426521282",
    "previous_sibling": "node_2_1482424204936"
},
{

```

```

    "type": "response_condition",
    "output": {
      "text": {
        "values": [
          "We are open on @holiday regular hours"
        ],
        "selection_policy": "sequential"
      }
    },
    "parent": "Hours of Operation",
    "context": {},
    "metadata": {},
    "conditions": "@holiday",
    "dialog_node": "node_5_1482426503106",
    "previous_sibling": "node_1_1522387330204"
  },
  {
    "type": "response_condition",
    "output": {
      "text": {
        "values": [
          "We are open on <? @sys-date.reformatDateTime(\"EEEEEE\") ?> from
10am until 8pm"
        ],
        "selection_policy": "sequential"
      }
    },
    "parent": "Hours of Operation",
    "context": {},
    "metadata": {},
    "conditions": "@sys-date.reformatDateTime(\"EEEEEE\") == \"Monday\" ||
@sys-date.reformatDateTime(\"EEEEEE\") == \"Tuesday\" || @sys-
date.reformatDateTime(\"EEEEEE\") == \"Wednesday\" || @sys-
date.reformatDateTime(\"EEEEEE\") == \"Thursday\" || @sys-
date.reformatDateTime(\"EEEEEE\") == \"Friday\"",
    "dialog_node": "node_1_1522387330204",
    "previous_sibling": "node_4_1482425833988"
  },
  {
    "type": "response_condition",
    "output": {
      "text": {
        "values": [
          "We are closed on @holiday"
        ],
        "selection_policy": "sequential"
      }
    },
    "parent": "Hours of Operation",
    "context": {},
    "metadata": {},
    "conditions": "@holiday:christmas || @holiday:thanksgiving ||
@holiday:(new years)",
    "dialog_node": "node_4_1482425833988"
  },
  {
    "type": "event_handler",
    "output": {
      "text": {
        "values": [
          "We only accept appointments between 11am and 5pm"
        ]
      }
    },
    "parent": "slot_105_1498132552870",
    "metadata": {},
    "next_step": {

```

```

        "behavior": "reprompt"
    },
    "conditions": "$time.after('17:30:30') || $time.before('10:59:59')",
    "event_name": "filled",
    "dialog_node": "handler_1_1509694458589",
    "previous_sibling": "handler_106_1498132552870"
},
{
    "type": "event_handler",
    "output": {},
    "parent": "slot_105_1498132552870",
    "context": {
        "time": "@sys-time"
    },
    "metadata": {},
    "conditions": "@sys-time",
    "event_name": "input",
    "dialog_node": "handler_106_1498132552870",
    "previous_sibling": "handler_107_1498132552870"
},
{
    "type": "event_handler",
    "output": {
        "text": "What time on <? $date.reformatDateTime(\"EEEEEE\") ?> do you
want to come in?"
    },
    "parent": "slot_105_1498132552870",
    "metadata": {},
    "event_name": "focus",
    "dialog_node": "handler_107_1498132552870"
},
{
    "type": "response_condition",
    "output": {
        "text": {
            "values": [
                "Hello. Good afternoon",
                "Hi there. It's a beautiful afternoon",
                "Good afternoon. How can I help?"
            ],
            "selection_policy": "sequential"
        }
    },
    "parent": "node_13_1502484041694",
    "metadata": {},
    "conditions": "now().after('12:00:00') && now().before('16:59:59')",
    "dialog_node": "node_1_1495022305143",
    "previous_sibling": "node_16_1488295517679"
},
{
    "type": "response_condition",
    "output": {
        "text": {
            "values": [
                "Hello. Good evening",
                "Hi. Good evening",
                "Hello. How can I help this evening?"
            ],
            "selection_policy": "sequential"
        }
    },
    "parent": "node_13_1502484041694",
    "metadata": {},
    "conditions": "now().after('17:00:00')",
    "dialog_node": "node_15_1488295465298",
    "previous_sibling": "node_1_1495022305143"
},

```

```

{
  "type": "response_condition",
  "output": {
    "text": {
      "values": [
        "Hello",
        "Hi there",
        "Hi. How can I help"
      ],
      "selection_policy": "sequential"
    }
  },
  "parent": "node_13_1502484041694",
  "metadata": {},
  "dialog_node": "node_28_1522448362216",
  "previous_sibling": "node_15_1488295465298"
},
{
  "type": "response_condition",
  "output": {
    "text": {
      "values": [
        "Hello. Good morning",
        "It's a beautiful morning. Hello",
        "Hi there. How can I help you this morning?"
      ],
      "selection_policy": "sequential"
    }
  },
  "parent": "node_13_1502484041694",
  "metadata": {},
  "conditions": "now().after('04:00:00') && now().before('11:59:59')",
  "dialog_node": "node_16_1488295517679"
},
{
  "type": "response_condition",
  "output": {
    "text": {
      "values": [
        "To get to our business from Grand Central, take the 4,5 or 6
train downtown to Union Square."
      ],
      "selection_policy": "sequential"
    }
  },
  "parent": "Directions",
  "metadata": {},
  "conditions": "@landmark:(grand central)",
  "dialog_node": "node_4_1522439442155",
  "previous_sibling": "node_8_1482459217052"
},
{
  "type": "response_condition",
  "output": {
    "text": {
      "values": [
        "To get to our business from the Empire State Building, walk to
Herald Square and take the N train to Union Square"
      ],
      "selection_policy": "sequential"
    }
  },
  "parent": "Directions",
  "metadata": {},
  "conditions": "@landmark:(empire state building)",
  "dialog_node": "node_7_1482459200886",
  "previous_sibling": "node_3_1522439390442"
}

```



```

    },
    {
      "type": "response_condition",
      "output": {
        "text": {
          "values": [
            "To get to our business from Times Square, take the N train
downtown to Union Square"
          ],
          "selection_policy": "sequential"
        }
      },
      "parent": "Directions",
      "metadata": {},
      "conditions": "@landmark:(times square)",
      "dialog_node": "node_8_1482459217052",
      "previous_sibling": "node_7_1482459200886"
    },
    {
      "type": "standard",
      "title": "Provide location",
      "output": {
        "text": {
          "values": [
            "We're located by Union Square on the corner of 13th and Broadway"
          ],
          "selection_policy": "sequential"
        }
      },
      "parent": "Directions",
      "metadata": {},
      "conditions": "true",
      "dialog_node": "node_3_1522439390442"
    },
    {
      "type": "event_handler",
      "output": {},
      "parent": "slot_12_1522596437268",
      "context": {
        "specialist": "@specialist"
      },
      "metadata": {},
      "conditions": "@specialist",
      "event_name": "input",
      "dialog_node": "handler_13_1522596437268",
      "previous_sibling": "handler_14_1522596437268"
    },
    {
      "type": "event_handler",
      "output": {
        "text": {
          "values": [
            "We'll do our best to book you with @specialist"
          ],
          "selection_policy": "sequential"
        }
      },
      "parent": "slot_12_1522596437268",
      "event_name": "filled",
      "dialog_node": "handler_15_1522596463593",
      "previous_sibling": "handler_13_1522596437268"
    },
    {
      "type": "event_handler",
      "output": {},
      "parent": "slot_12_1522596437268",
      "event_name": "focus",

```

```

    "dialog_node": "handler_14_1522596437268"
  },
  {
    "type": "event_handler",
    "output": {
      "text": {
        "values": [
          "Thanks"
        ],
        "selection_policy": "sequential"
      }
    },
    "parent": "slot_22_1522444583114",
    "context": {},
    "metadata": {},
    "conditions": "true",
    "event_name": "filled",
    "dialog_node": "handler_22_1522598191131",
    "previous_sibling": "handler_23_1522444583114"
  },
  {
    "type": "event_handler",
    "output": {},
    "parent": "slot_22_1522444583114",
    "context": {
      "phone": "@phone"
    },
    "metadata": {},
    "conditions": "@phone",
    "event_name": "input",
    "dialog_node": "handler_23_1522444583114",
    "previous_sibling": "handler_24_1522444583114"
  },
  {
    "type": "event_handler",
    "output": {
      "text": "I'll just need a phone to hold your reservation"
    },
    "parent": "slot_22_1522444583114",
    "metadata": {},
    "event_name": "focus",
    "dialog_node": "handler_24_1522444583114"
  },
  {
    "type": "standard",
    "title": "Ask about product",
    "actions": [
      {
        "name": "main_webhook",
        "type": "webhook",
        "parameters": {
          "input": "<?input.text?>"
        },
        "result_variable": "webhook_result_1"
      }
    ],
    "metadata": {
      "_customization": {
        "mcr": true
      }
    },
    "conditions": "#Product_Information",
    "dialog_node": "node_6_1591626152244",
    "previous_sibling": "node_4_1570050459690"
  },
  {
    "type": "standard",

```

```

        "title": "Please transfer me to an agent",
        "output": {
            "text": {
                "values": [
                    "Would you like me to transfer you to a representative?"
                ],
                "selection_policy": "sequential"
            }
        },
        "metadata": {},
        "conditions": "#General_Connect_to_Agent",
        "digress_in": "does_not_return",
        "dialog_node": "node_22_1467833484410",
        "digress_out": "allow_all_never_return",
        "previous_sibling": "node_2_1468243505617"
    },
    {
        "type": "standard",
        "output": {
            "text": {
                "values": [
                    "I didn't understand can you try again"
                ],
                "selection_policy": "sequential"
            }
        },
        "metadata": {},
        "conditions": "anything_else",
        "digress_in": "returns",
        "dialog_node": "node_2_1467831978407",
        "digress_out": "allow_all",
        "previous_sibling": "node_6_1591626152244",
        "disambiguation_opt_out": true
    },
    {
        "type": "standard",
        "output": {},
        "metadata": {},
        "conditions": "#General_Greetings",
        "digress_in": "does_not_return",
        "dialog_node": "node_13_1502484041694",
        "previous_sibling": "Reservation using slots"
    },
    {
        "type": "standard",
        "title": "What can I do",
        "output": {
            "generic": [
                {
                    "values": [
                        {
                            "text": "I can tell you about our store locations and opening hours, or help you set up an appointment."
                        },
                        {
                            "text": "You could also ask me to connect you to an agent."
                        }
                    ],
                    "response_type": "text",
                    "selection_policy": "sequential"
                }
            ]
        },
        "conditions": "#Help",
        "dialog_node": "node_4_1570050459690",
        "previous_sibling": "node_22_1467833484410"
    },

```

```

{
  "type": "standard",
  "title": "Where are you located?",
  "output": {},
  "metadata": {},
  "next_step": {
    "behavior": "skip_user_input"
  },
  "conditions": "#Customer_Care_Store_Location",
  "digress_in": "does_not_return",
  "dialog_node": "Directions",
  "digress_out": "allow_all",
  "previous_sibling": "Hours of Operation"
},
{
  "type": "standard",
  "title": "What are your hours?",
  "output": {},
  "metadata": {},
  "next_step": {
    "behavior": "jump_to",
    "selector": "body",
    "dialog_node": "node_3_1522439390442"
  },
  "conditions": "#Customer_Care_Store_Hours",
  "digress_in": "does_not_return",
  "dialog_node": "Hours of Operation",
  "digress_out": "allow_all",
  "previous_sibling": "Opening"
},
{
  "type": "standard",
  "output": {
    "text": {
      "values": [
        "You're welcome. Just let me know if you need anything else",
        "No problem. Just let me know if you need anything else",
        "My pleasure. Just let me know if you need anything else"
      ],
      "selection_policy": "sequential"
    }
  },
  "metadata": {},
  "conditions": "#Thanks",
  "digress_in": "does_not_return",
  "dialog_node": "node_2_1468243505617",
  "previous_sibling": "node_12_1468329566917"
},
{
  "type": "standard",
  "output": {
    "text": {
      "values": [
        "So long",
        "See ya",
        "Good bye"
      ],
      "selection_policy": "sequential"
    }
  },
  "metadata": {},
  "conditions": "#Goodbye",
  "digress_in": "does_not_return",
  "dialog_node": "node_12_1468329566917",
  "previous_sibling": "node_13_1502484041694"
},
{

```

```

        "type": "frame",
        "title": "I want to make an appointment",
        "output": {},
        "metadata": {
            "fallback": "leave",
            "_customization": {
                "mcr": true
            }
        },
        "conditions": "#Customer_Care_Appointments",
        "digress_in": "does_not_return",
        "dialog_node": "Reservation using slots",
        "digress_out": "allow_all",
        "previous_sibling": "Directions",
        "digress_out_slots": "allow_all"
    },
    {
        "type": "standard",
        "title": "Opening",
        "output": {
            "text": {
                "values": [
                    "Hello, I'm a demo customer care virtual assistant to show you the basics. I can help with directions to my store, hours of operation and booking an in-store appointment"
                ],
                "selection_policy": "sequential"
            }
        },
        "context": {
            "no_reservation": true
        },
        "metadata": {},
        "conditions": "welcome",
        "dialog_node": "Opening"
    }
],
"counterexamples": [],
"system_settings": {
    "tooling": {
        "store_generic_responses": true
    },
    "off_topic": {
        "enabled": true
    },
    "disambiguation": {
        "prompt": "Did you mean:",
        "enabled": true,
        "randomize": true,
        "max_suggestions": 5,
        "suggestion_text_policy": "title",
        "none_of_the_above_prompt": "None of the above."
    },
    "system_entities": {
        "enabled": true
    },
    "spelling_auto_correct": true
},
"learning_opt_out": false,
"name": "Customer Care Sample Skill",
"language": "en",
"description": "Sample simple customer service skill to get you started."
}

```

input (Node Red Function Code)

```
msg.payload=msg.payload.input;  
return msg;
```

output (Node Red Function Code)

```
msg.payload = msg.payload.output.text[0];  
return msg;
```

B. Reference

1. <https://developer.ibm.com/patterns/enhance-customer-help-desk-with-smart-document-understanding/>
2. <https://github.com/IBM/watson-discovery-sdu-with-assistant>
3. <https://www.youtube.com/watch?v=-yniuX-Poyw&feature=youtu.be>
4. <https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/>

THE END