

PROJECT REPORT

Intelligent Customer Help Desk with Smart Document Understanding

Category: Artificial Intelligence Developer

Application ID: SPS_APL_20200000638

Project ID:

SPS_PRO_99

Internship at SmartInternz

Rahil Desai

rahildesai.rd99@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

1.

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.

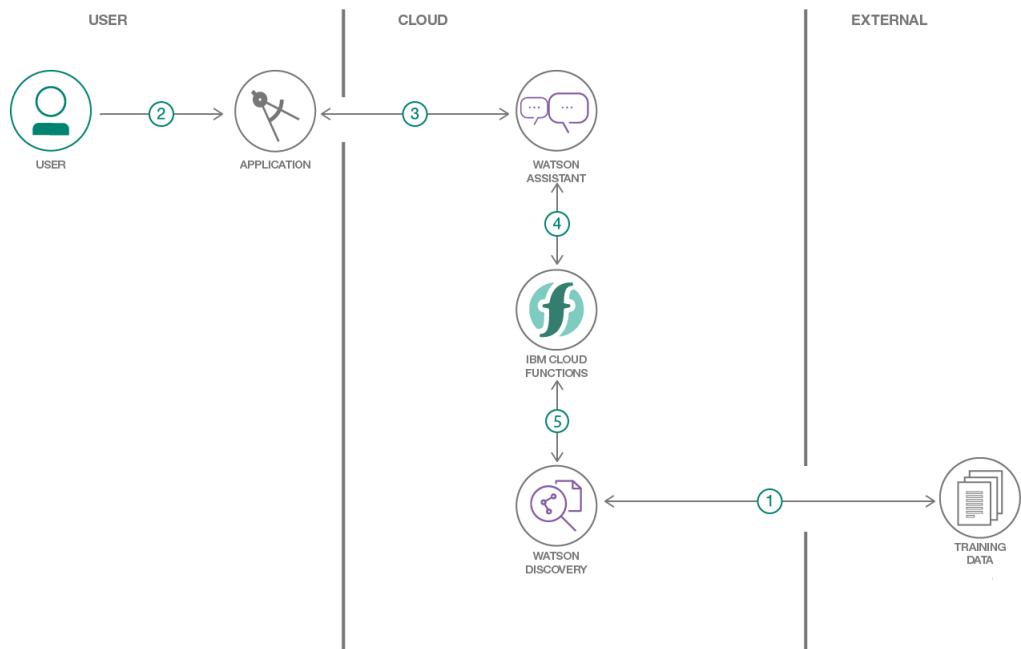
2.2 Proposed solution

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



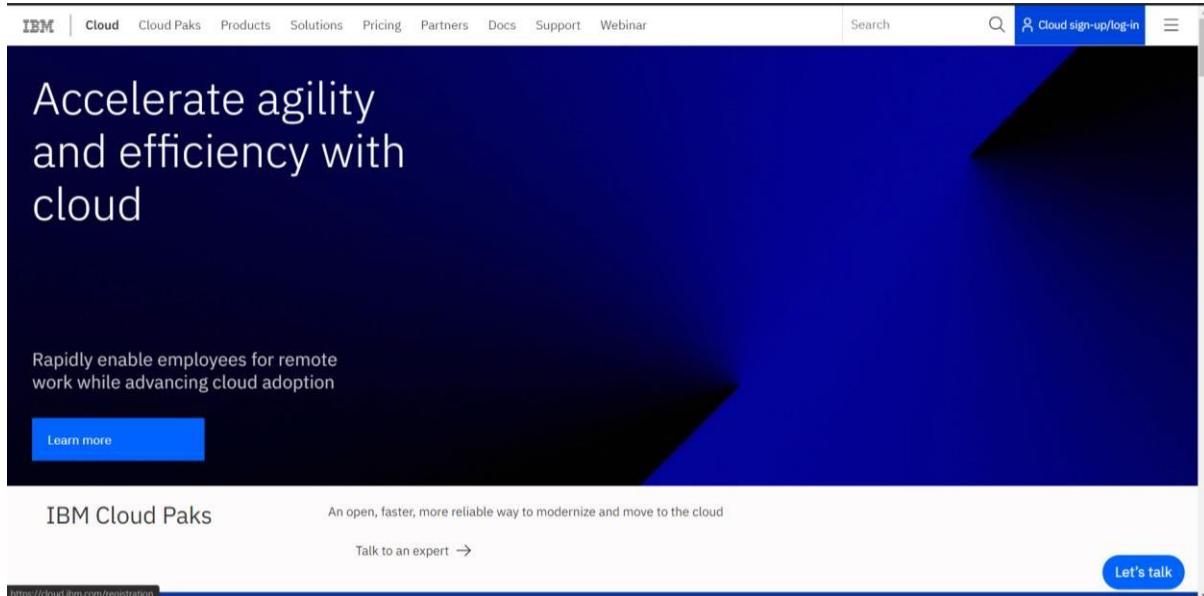
3.2 Hardware / Software designing

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

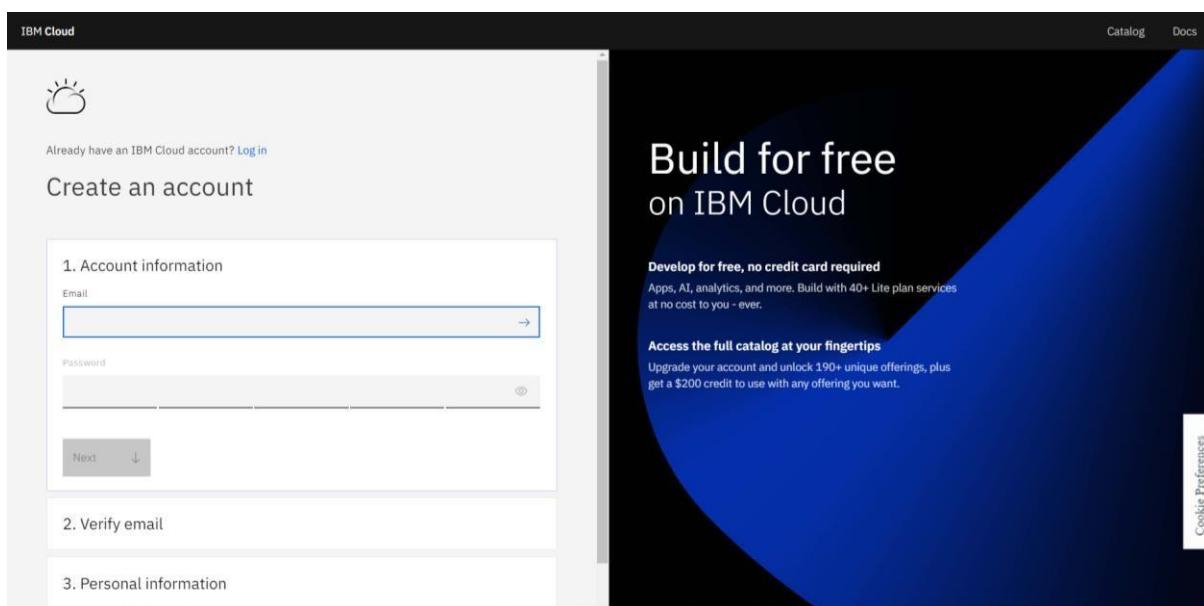
1. Create IBM Cloud Services

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



The screenshot shows the top navigation bar of the IBM Cloud website. It includes links for IBM, Cloud, Cloud Paks, Products, Solutions, Pricing, Partners, Docs, Support, and Webinar. A search bar and a "Cloud sign-up/log-in" button are also present. The main banner features the text "Accelerate agility and efficiency with cloud" and a subtext about enabling remote work. A blue "Learn more" button is visible. Below the banner, there's a section for "IBM Cloud Paks" with a subtext about modernizing and moving to the cloud, a "Talk to an expert" link, and a "Let's talk" button.

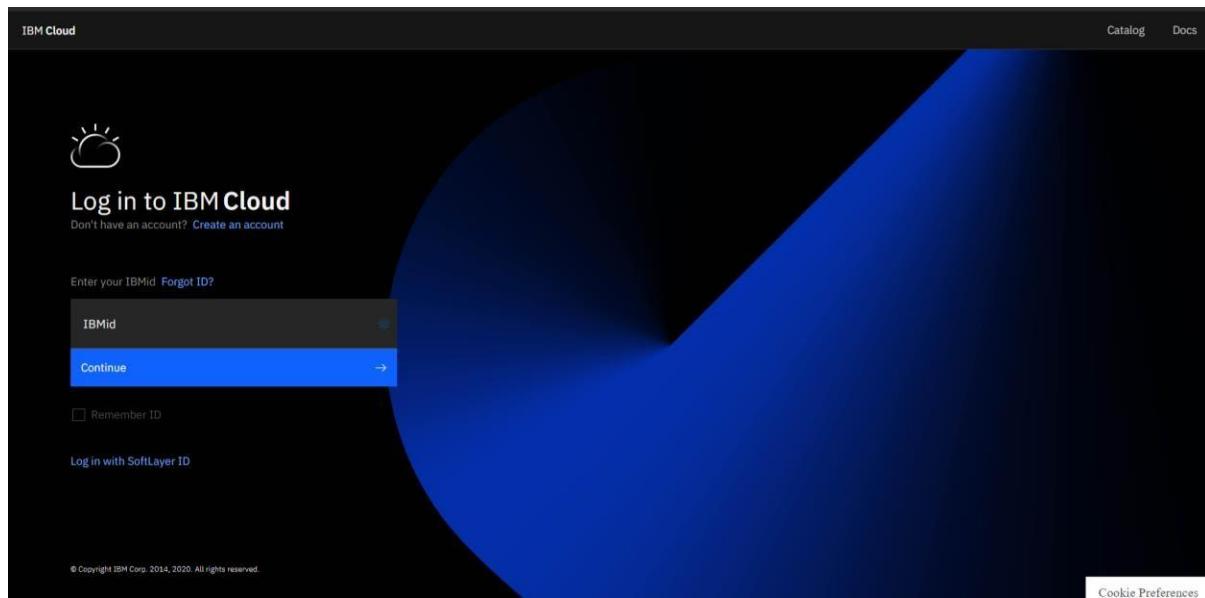
Click on [Cloud sign-up/log-in](#) to sign up or login to IBM Cloud.



The screenshot shows the "Create an account" page. It has three steps: 1. Account information (Email and Password fields), 2. Verify email (Email field), and 3. Personal information (Name and Phone number fields). To the right, there's a promotional section for "Build for free on IBM Cloud" with text about no credit card required and a \$200 credit. There are also links for "Catalog" and "Docs". A "Cookie Preferences" link is at the bottom 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 [Log in](#) and Log in to your cloud account.



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

A screenshot of the IBM Cloud Dashboard. The top navigation bar includes 'IBM Cloud', a search bar, and links for 'Catalog', 'Docs', 'Support', 'Manage', and a user profile. The dashboard features several cards: 'Resource summary' (13 resources, including Cloud Foundry apps, services, storage, functions namespaces, and apps), 'Planned maintenance' (clear skies), 'For you' (lightboard video on Infrastructure as Code), 'Leverage code patterns, practises, and architectures to modernize workloads in stages or through a complete...', 'User access' (enter email addresses), 'IBM Cloud status' (world map showing global reach), and news sections for Airtel, Vodafone, and the Annual IBM List of Global Women Leaders.

To Create any Resource (Services/Apps/etc), click on [Create resource](#) +

The screenshot shows the IBM Cloud Catalog homepage. On the left, there's a sidebar with navigation links like Catalog, IBM Cloud catalog, Featured, Services, and Software. The main area features a large banner with the text "IBM Cloud products" and "Over 190+ products available for you to customize and build the solutions that you need for your business". Below the banner is a search bar with the placeholder "Search the catalog...". A section titled "Recommended for you" lists several services: App ID, Object Storage, Visual Recognition, Streaming Analytics, Continuous Delivery, and Speech to Text. Each service card includes a brief description and status (Lite • Free • IAM-enabled).

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

The screenshot shows the IBM Cloud Catalog search results for the term "discovery". The search bar at the top contains "discovery". Below it, the results are displayed under the heading "Search results for 'discovery' 2 results". Two services are listed: "Discovery" and "HashiCorp Consul". The "Discovery" service card is highlighted with a blue border. It includes a brief description: "Add a cognitive search and content analytics engine to applications." and its status: "Lite • Free • IAM-enabled". The "HashiCorp Consul" service card includes a brief description: "Highly available and distributed service discovery and key-value store designed with support for the modern data cente..." and its status: "Helm charts • IBM Kubernetes Service • Free".

Click on



A screenshot of the IBM Cloud service creation interface for 'Discovery'. The top navigation bar shows 'IBM Cloud', a search bar, and various menu items like Catalog, Docs, Support, Manage, and a user profile. The main area shows the 'Discovery' service card with a 'Summary' section and a 'Create' button. The 'Create' button is highlighted with a blue border. The summary section displays details such as Region: London, Plan: Lite, Service name: Discovery-qk, and Resource group: Default. On the right side, there's a 'Feedback' button and a sidebar with 'Create', 'Add to estimate', and 'View terms' buttons.

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

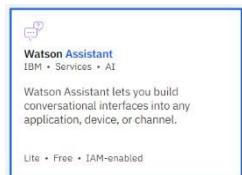
Your Watson Discovery service is created successfully.

(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

The screenshot shows the IBM Cloud Catalog interface. In the top navigation bar, there is a search bar with the text "assistant". Below the search bar, the results are displayed under the heading "Search results for 'assistant' 1 result". The single result is a card for "Watson Assistant" from IBM, categorized under Services > AI. The card includes a brief description: "Watson Assistant lets you build conversational interfaces into any application, device, or channel." It also indicates the plan: "Lite • Free • IAM-enabled". The entire service card is highlighted with a blue border.

Click on



The screenshot shows the "Watson Assistant" creation page. At the top, there is a breadcrumb navigation: Catalog / Services / Watson Assistant. Below the title, it says "Author: IBM • Date of last update: 05/14/2020 • Docs • API docs". There are two tabs: "Create" (which is selected) and "About". The main area has two sections: "Select a region" and "Select a pricing plan". Under "Select a region", a dropdown menu is open, showing "London" as the selected option. Under "Select a pricing plan", it says "Displayed prices do not include tax. Monthly prices shown are for country or region: United States". A table compares the "Lite" plan against the "Features" and "Pricing". The "Lite" plan includes 10,000 Messages/Month, AI-Based Intent and Entity Recognition, Entity Synonym Recommendations, Visual Dialog Edit with Simple Response Types (Text, Options, Images, etc...), Prebuilt Content Available, Analytics Dashboard with 7 Days of Storage, 5 Dialog Skills, Each with 100 Dialog Nodes, and Shared Public Cloud. The "Pricing" column shows "Free". On the right side, there is a "Summary" panel with the following details: Watson Assistant (Service name), Region: London, Plan: Lite, Service name: Watson Assistant-ro, Resource group: Default. Below the summary, there are buttons for "Create", "Add to estimate", and "View terms". A feedback button is located at the bottom right of the summary panel.

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

13 Resources

| Category | Count |
|------------------------|-------|
| Cloud Foundry apps | 1 |
| Cloud Foundry services | 2 |
| Services | 6 |
| Storage | 1 |
| Functions namespaces | 1 |
| Apps | 1 |

Add resources +

View all

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

IBM Cloud

Search resources and offerings...

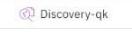
Catalog Docs Support Manage Hashim Irfan Al... Create resource

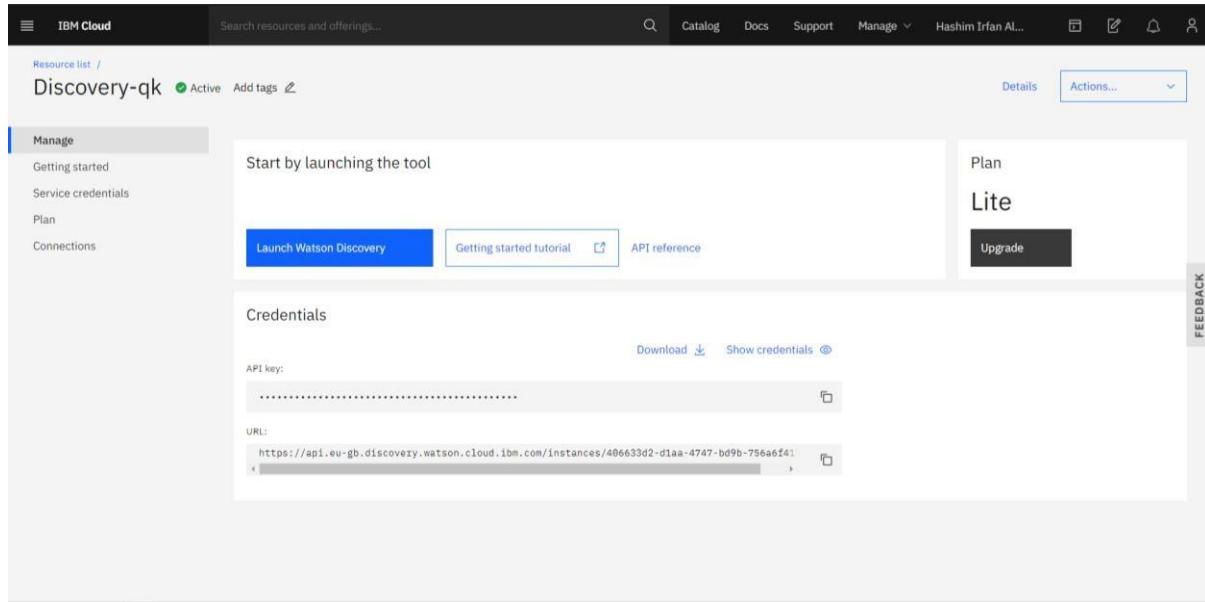
Resource list

| Name | Group | Location | Offering | Status | Tags |
|---------------------------------------|---------|------------|---------------------|--------|---------|
| Continuous Delivery | Default | London | Continuous Delivery | Active | |
| Discovery-qk | 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-eq | Default | Dallas | Visual Recognition | Active | cpda... |

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

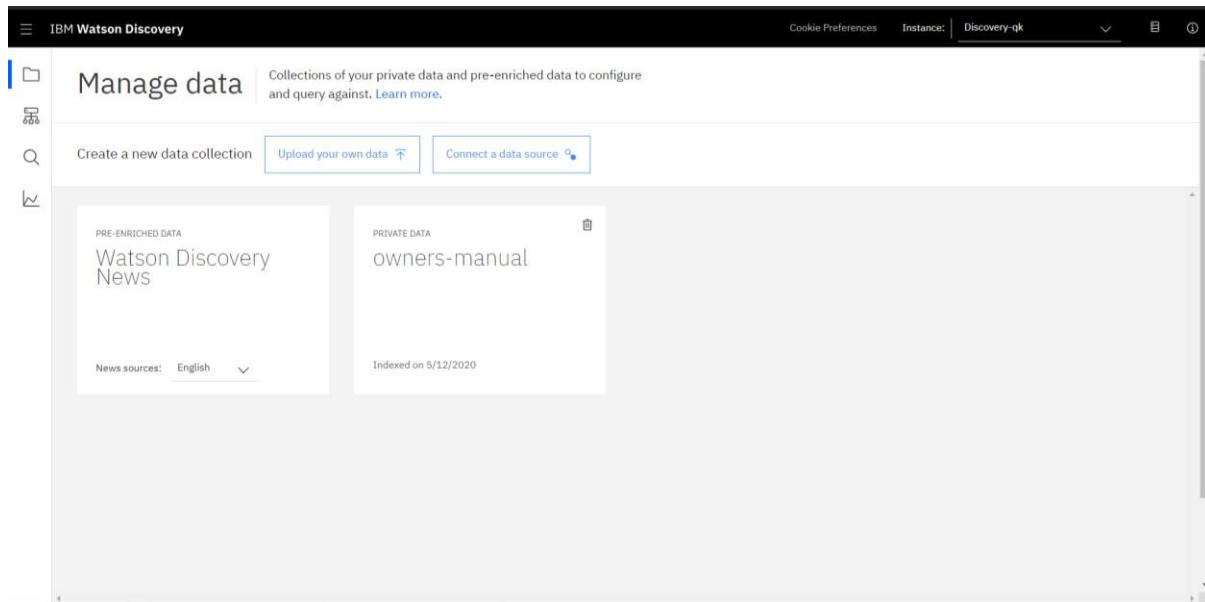
2. Configure Watson Discovery

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



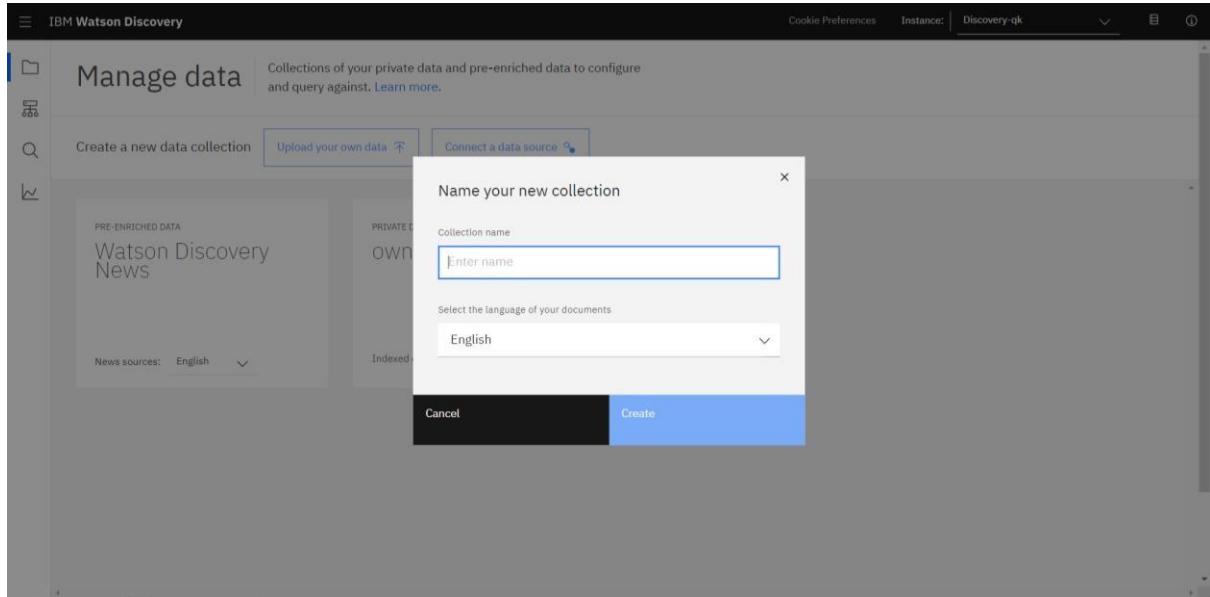
The screenshot shows the IBM Cloud Resource List interface. In the center, there's a card for the "Discovery-qk" service. On the left side of the card, under the "Manage" section, there's a blue button labeled "Launch Watson Discovery". To the right of this button are links for "Getting started tutorial" and "API reference". On the far right of the card, there's a "Plan" section showing "Lite" and a "Upgrade" button. At the top of the page, there's a navigation bar with links for Catalog, Docs, Support, Manage, and a user profile.

Click on  to launch Watson Discovery Service.

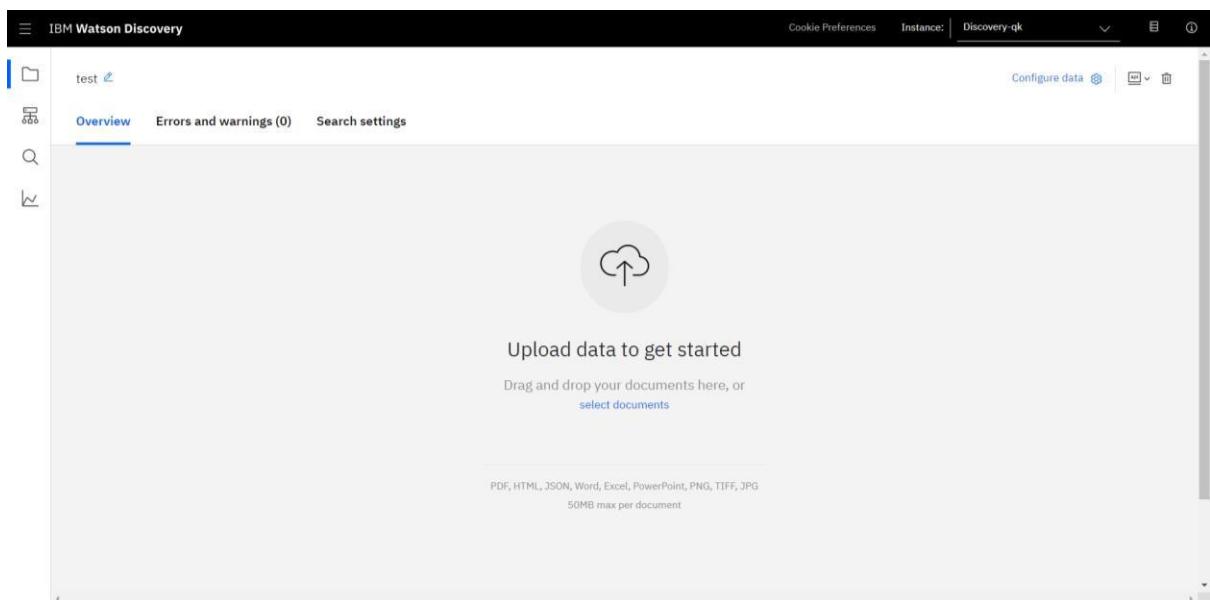


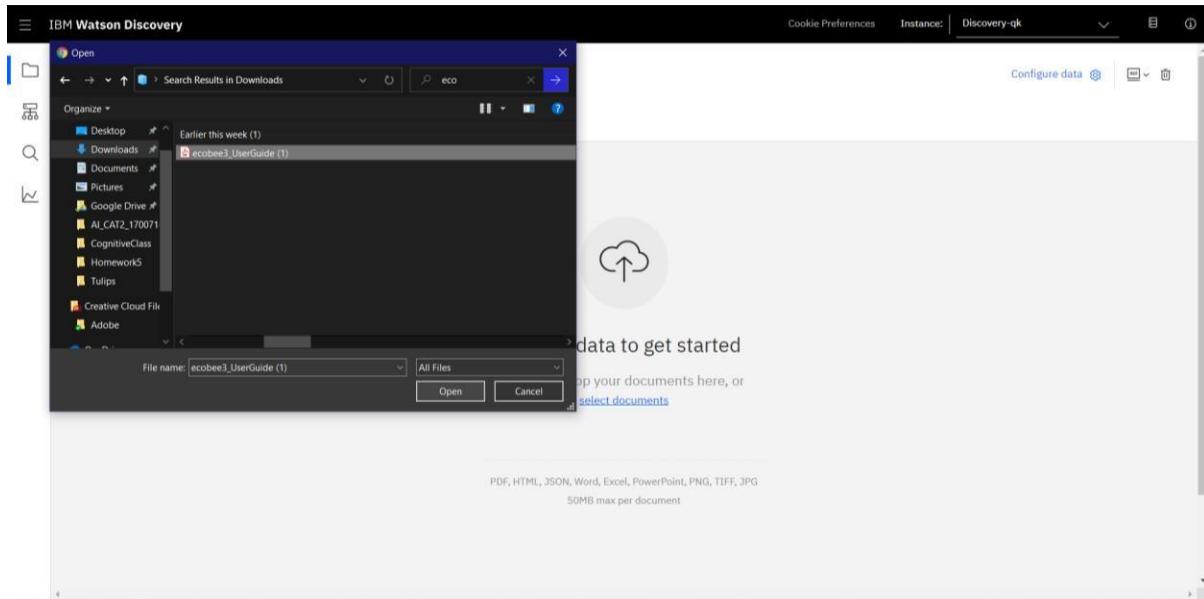
The screenshot shows the IBM Watson Discovery service interface. At the top, there's a header with the service name and navigation links for Cookie Preferences, Instance, and a dropdown menu. Below the header, there's a sidebar with icons for Manage data, Create a new data collection, Upload your own data, and Connect a data source. The main area is titled "Manage data" and contains two data collections: "PRE-ENRICHED DATA" (Watson Discovery News) and "PRIVATE DATA" (owners-manual). Each collection has a preview, a status indicator, and a delete icon. At the bottom of the main area, there's a note about news sources and indexing date.

Click on  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.

The screenshot shows the Watson Discovery interface after processing the uploaded document. Key details include:

- Overview:** Shows 1 document.
- Identified fields:** 1 field identified, labeled "text".
- Added enrichments:** 4 enrichments added, including Entity Extraction (104 °F, 20 min, etc.), Sentiment Analysis (100%, 0%, 0%), Concept Tagging (Air conditioner, Energy recovery, Geothermal heat pump), and Category Classification.
- Analysis metrics:** Entity Extraction (104 °F, 20 min, etc.), Sentiment Analysis (100%, 0%, 0%), Concept Tagging (Air conditioner, Energy recovery, Geothermal heat pump), and Category Classification.
- Ready to query:** A sidebar suggests queries like "Most common entity types and their top entities" and "Documents that contain Air conditioner, but not Energy recovery".

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.

The screenshot shows the IBM Watson Discovery interface. At the top, there are tabs for 'Overview', 'Errors and warnings (1)', and 'Search settings'. Below the tabs, it displays '1 document' with '0 documents failed'. It shows the creation date as 5/16/2020 11:29:11 am EDT and the last update as 5/16/2020 11:29:11 am EDT. There is a button to 'Upload documents'. On the left, there are icons for file, search, and refresh. In the center, it says 'Identified 1 field from your data' (text) and 'Added 4 enrichments to your data' (Entity Extraction, Sentiment Analysis, Concept Tagging, Category Classification). On the right, it says 'Now you're ready to query!' with three cards: 'Most common entity types and their top entities', 'Documents that contain Air conditioner, but not Energy recovery', and 'Top people related to /business and industrial/energy'. A red box highlights the 'Build your own query →' button at the bottom right.

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.

The screenshot shows the 'Build queries' page. It has a sidebar with 'test / Build queries' and a search bar. The main area has a heading 'Build a query using one or more of these components. Learn more.' with a 'Use a sample query' button. Below this, there is a search bar with 'Search for documents' and options 'Use natural language' and 'Use the Discovery Query Language'. The query 'how do i turn on the heater' is entered. There are buttons for '+ Include analysis of your results' and '+ Filter which documents you query'. At the bottom, there are 'More options', 'Run query', and 'Close' buttons. To the right, under 'Summary' and 'JSON', it shows a query URL: 'Query URL https://api.eu-gb.discovery.watson.cloud.ibm.com/instances/4'. Under 'Passages', it shows a snippet of text about the ecobee3 smart Wi-Fi thermostat. Under 'Results', it shows 'Showing 1 of 1 matching documents' with a result for 'ecobee3_UserGuide (1).pdf' with sentiment 'positive'.

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)

IBM Watson Discovery

Overview Errors and warnings (1) Search settings

1 document

0 documents failed View details

Created on Last updated 5/16/2020 11:29:11 am EDT 5/16/2020 11:29:11 am EDT

Upload documents

Identified 1 field from your data
text

Need to identify more fields? Add fields

Added 4 enrichments to your data

- Entity Extraction: 104 °F (1), 20 min (1), 20 minutes (1), 20 °C (1), 24-hour (1)
- Sentiment Analysis: 100% positive, 0% neutral, 0% negative
- Concept Tagging: Air conditioner (1), Energy recovery (1), Geothermal heat pump (1)
- Category Classification: business and industr... energy

Now you're ready to query!

- Documents about 104 °F as a Quantity with a very negative sentiment
- Top people related to /business and industrial/energy
- Most common entity types and their top entities

5 enrichments available. Add enrichments

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

IBM Watson Discovery

Identify fields Manage fields Enrich fields

ecobee3_UserGuide (1).pdf 1/1 41

1

2 User Guide ecobee3

3

4

5 Submit page

Field labels

Identify document elements using the labels below.

+ Create new Upgrade

- answer
- author
- footer
- header
- question
- subtitle
- table_of_contents
- text
- title
- image
- table

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,

The screenshot shows the IBM Watson Discovery interface. On the left, there is a sidebar with icons for folder, file, search, and refresh. The main area displays a PDF document titled "ecobee3_UserGuide.pdf" with page numbers 37, 38, 39, 40, and 41. The first few pages contain a table of contents and some text. A large portion of the page content is highlighted in green, indicating annotated text. To the right of the document, there is a "Field labels" sidebar. It includes a heading "Field labels" and a sub-heading "Identify document elements using the labels below." Below this are several color-coded labels with corresponding icons: answer (yellow), author (blue), footer (green), header (blue), question (green), subtitle (pink), table_of_contents (light blue), text (pink), title (red), image (light blue), and table (orange). There is also a link "+ Create new" and an "Upgrade" button. At the bottom of the sidebar, there is a link "Learn more about how to use." At the top right of the interface, there are buttons for "Cookie Preferences", "Instance: Discovery-qk", and "Apply changes to collection".

Click the **Apply changes to collection** [5]

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

IBM Watson Discovery

owners-manual / Configure data

Identify fields Manage fields Enrich fields

Identify fields to index

All fields are indexed by default. Switch off any fields you do not want to be indexed. [Learn more.](#)

answer Off
author Off
footer Off
header Off
image Off
question Off
subtitle On
table Off
table_of_contents Off
text On
title Off

Improve query results by splitting your documents

You can split your documents into segments based on fields. Once split, each segment is a separate document that will be enriched, indexed, and returned as a query separately. [Learn more.](#)

Split document on each occurrence of

subtitle

Apply changes to collection

[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

IBM Watson Discovery

owners-manual

Configure data

Overview Errors and warnings (133) Search settings

133 documents

0 documents failed View details

Created on Last updated

5/12/2020 4:09:01 am EDT 5/12/2020 4:09:01 am EDT

Upload documents

Identified **6 fields** from your data

author
footer
subtitle
table_of_contents
text
title

Need to identify more fields? [Add fields](#)

Added **4 enrichments** to your data

Entity Extraction
0.3°C (4) | 0.5°F (4) | 10°F (4) |
20 min (3) | 4-digit (3)

Sentiment Analysis
54% positive
35% neutral
11% negative

Concept Tagging
HVAC (13) | Heat (13) | Internet (12) |
Temperature (11) | Netscape (10)

Category Classification
technology and com... operating systems

Now you're ready to **query**!

Top people related to /technology and computing/operating systems
[Run](#)

Entities of type **Quantity** which have negative sentiment
[Run](#)

Entities of type **Quantity** which have positive sentiment
[Run](#)

5 enrichments available. [Add enrichments](#)

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

The screenshot shows the IBM Watson Discovery interface. On the left, there's a sidebar with a folder icon labeled "owners-manual / Build queries". Below it is a search bar with placeholder text "Build a query using one or more of these components. Learn more." and a "Use a sample query" button. The main area has a search bar with the query "how do i turn on the heater?". Below the search bar are three buttons: "+ Include analysis of your results", "+ Filter which documents you query", and "More options". At the bottom are "Run query" and "Close" buttons. To the right, there's a "Summary" tab selected, showing a "Query URL" field with the value "https://api.eu-gb.discovery.watson.cloud.ibm.com/instances/4". Below it is a "Passages" section with a detailed description of HVAC system settings and a note about screen sleep. There's also a note about filter replacement and a general note about system dependencies.

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

The screenshot shows the IBM Watson Discovery interface on the "Overview" page. It displays "132 documents". Key statistics include "0 documents failed" and "Created on 5/12/2020 4:09:01 am EDT". On the right, there's a panel titled "Configure data" with fields for "Collection ID" (redacted), "Configuration ID" (aa5 [redacted] b20), and "Environment ID" (d5f [redacted] 345). Below this are sections for "Identified 6 fields from your data" (author, footer, subtitle, table_of_contents, text, title) and "Added 4 enrichments to your data" (Entity Extraction, Sentiment Analysis, Concept Tagging, Category Classification). Each enrichment has a "Run" button. A note at the bottom says "5 enrichments available. Add enrichments".

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.

The screenshot shows the 'Service credentials' section for a service named 'Discovery-qk'. On the left, there's a sidebar with 'Manage', 'Getting started', 'Service credentials' (which is selected and highlighted in blue), 'Plan', and 'Connections'. The main area has a search bar 'Search credentials...' and a table header with 'Key name' and 'Date created'. A single row is listed under 'Auto-generated service credentials':

| Key name | Date created |
|----------|----------------------------|
| apikey | MAY 12, 2020 - 01:06:25 PM |

The 'apikey' row contains the following JSON data:

```
{ "apikey": "c[REDACTED]88", "iam_apikey_description": "Auto-generated for key 94fa8a96-9c49-404d-8072-5dee56e0d754", "iam_apikey_name": "Auto-generated service credentials", "iam_crn": "cnr:v1:bluemix:public:iam::::serviceRole:Manager", "iam_serviceid_crn": "cnr:v1:bluemix:public:iam:identity:a/5a313b9c581d4b939514c678b699110c::serviceid:ServiceId-d18ad745-ed3f-44a6-a175-38db2b17307f", "url": "https://api.eu-gb.discovery.watson.cloud.ibm.com/instances/406633d2-d1aa-4747-bd9b-756a6f417348" }
```

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

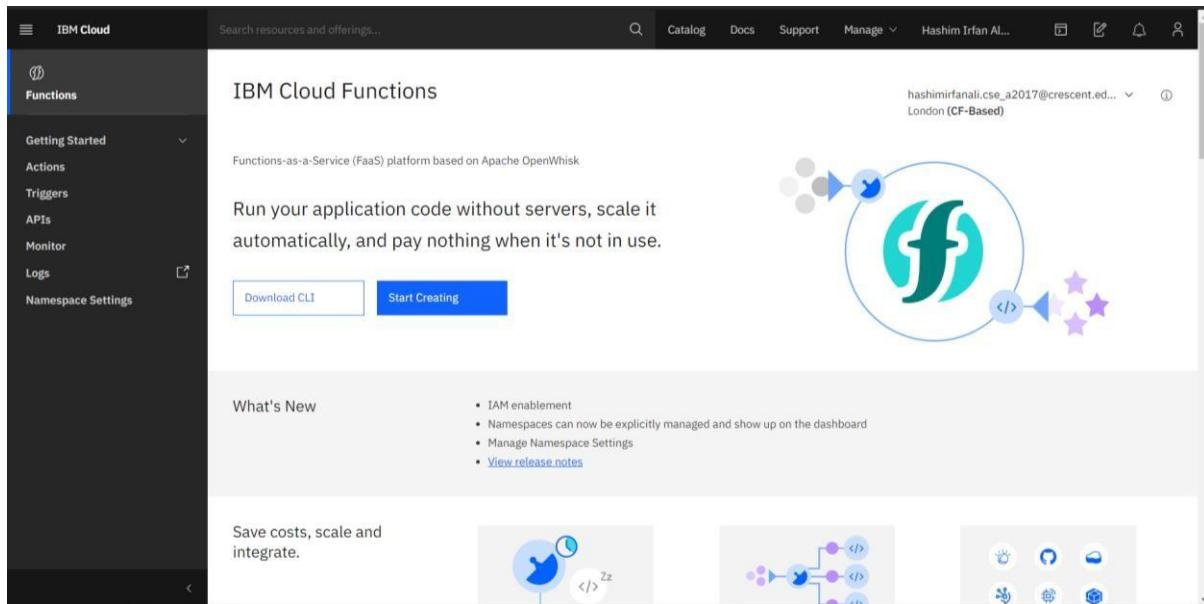
The screenshot shows the IBM Cloud Catalog search results for 'functions'. The search bar at the top has 'functions' typed into it. Below the search bar, the results are displayed with a title 'Search results for \'functions\' 1 result'. One result is shown in a box:

 **Functions**
IBM + Services • Compute

IBM Cloud Functions is a Function-as-a-Service (FaaS) platform which executes functions in response to incoming events.

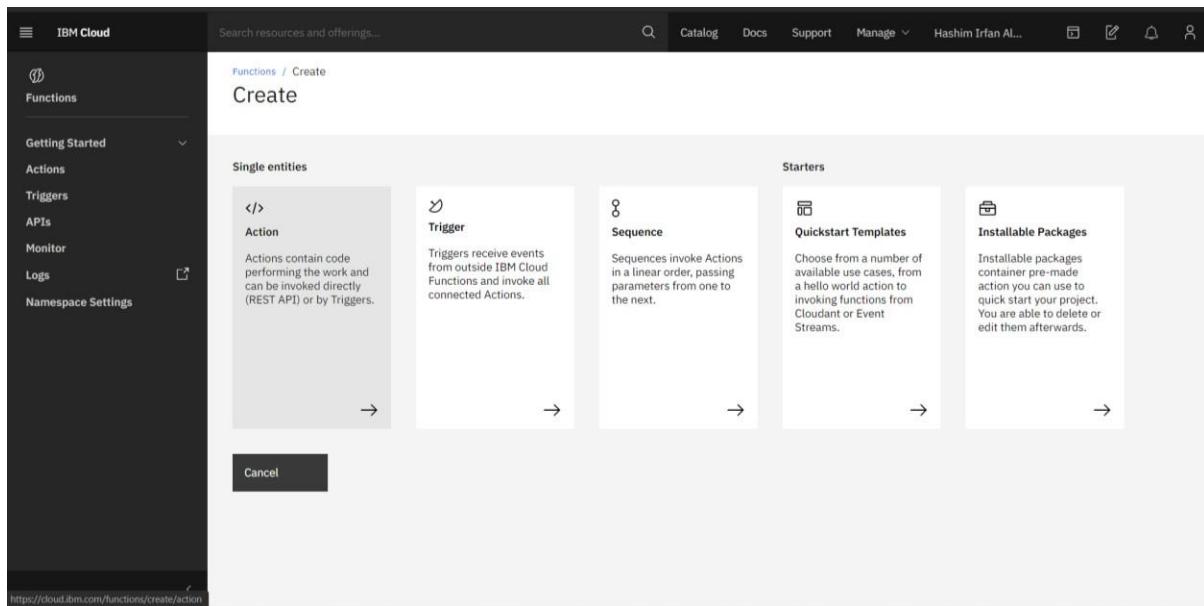
IAM-enabled

Select the Functions Card.



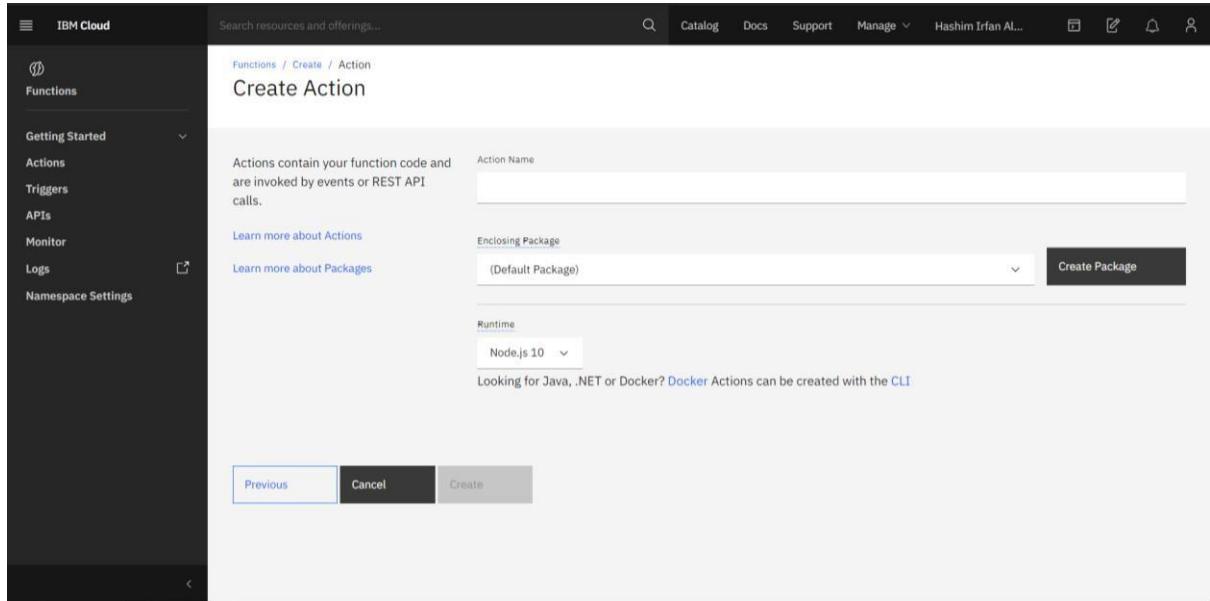
The screenshot shows the IBM Cloud Functions main panel. On the left, there's a sidebar with a 'Functions' card selected. The main content area is titled 'IBM Cloud Functions' and describes it as a 'Functions-as-a-Service (FaaS) platform based on Apache OpenWhisk'. It encourages users to 'Run your application code without servers, scale it automatically, and pay nothing when it's not in use.' Below this, there are two buttons: 'Download CLI' and 'Start Creating'. To the right, there's a large circular icon featuring a stylized 'f' and arrows, with the text 'hashimirfanali.cse_a2017@crescent.edu... London (CF-Based)' above it. The bottom section is titled 'What's New' and lists recent changes like IAM enablement and namespace management. There are also three small icons illustrating cost savings, scaling, and integration.

From the Functions main panel, click on Start Creating.



The screenshot shows the 'Create' panel within the IBM Cloud Functions interface. The left sidebar remains the same. The main area is titled 'Create' and contains several cards under the heading 'Single entities'. These include 'Action' (described as containing code for work), 'Trigger' (described as receiving events from outside), 'Sequence' (described as invoking Actions in a linear order), 'Quickstart Templates' (described as choosing from available use cases), and 'Installable Packages' (described as containing pre-made actions). Each card has a right-pointing arrow indicating a next step. At the bottom left is a 'Cancel' button.

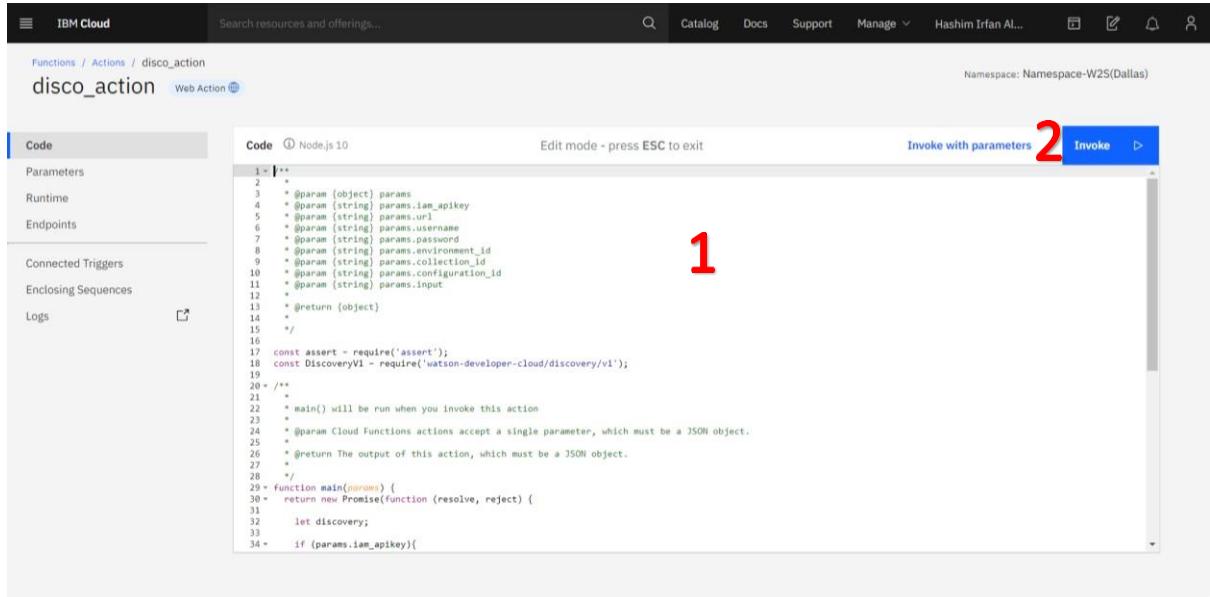
Here, select Actions.



The screenshot shows the 'Create Action' page in the IBM Cloud Functions interface. On the left, there's a sidebar with 'Functions' selected. The main area has a title 'Create Action'. It includes fields for 'Action Name' (empty), 'Enclosing Package' (set to '(Default Package)'), and 'Runtime' (set to 'Node.js 10'). Below these are links for 'Actions', 'Triggers', 'APIs', 'Monitor', 'Logs', and 'Namespace Settings'. At the bottom are buttons for 'Previous', 'Cancel', and '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 'disco_action' function details page. The 'Code' tab is selected, showing the Node.js 10 code. The code is a simple function that uses the Watson Developer Cloud Discovery service to query a collection. A red number '1' is placed over the code editor window. A red number '2' is placed over the 'Invoke' button in the top right corner of the editor.

```
1 /**
2  * @param {object} params
3  * @param {string} params.iam_apikey
4  * @param {string} params.url
5  * @param {string} params.username
6  * @param {string} params.password
7  * @param {string} params.collection_id
8  * @param {string} params.environment_id
9  * @param {string} params.configuration_id
10 * @param {string} params.input
11 */
12 *
13 * @return {object}
14 *
15 */
16
17 const assert = require('assert');
18 const DiscoveryV1 = require('watson-developer-cloud/discovery/v1');
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 * @param {Object} params
26 * @return {Object} 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     if (params.iam_apikey){
```

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

The screenshot shows the IBM Cloud Functions Actions interface. The top navigation bar includes 'IBM Cloud', a search bar, and links for Catalog, Docs, Support, Manage, and Hashim Irfan Al... A red number '1' is overlaid on the 'Parameters' tab in the left sidebar. The main content area is titled 'Parameters' and lists four parameters: 'url' (value: "https://api.eu-gb.discovery.watson.cloud.ibm.com/instances/406633d2-d1aa-474"), 'environment_id' (value: "disco_8a...5f41"), 'collection_id' (value: "31...5f41"), and 'iam_apikey' (value: "cA...B8"). An 'Add Parameter' button is located at the top right of the parameters table.

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

The screenshot shows the IBM Cloud Functions Actions interface. The top navigation bar includes 'IBM Cloud', a search bar, and links for Catalog, Docs, Support, Manage, and Hashim Irfan Al... A red number '1' is overlaid on the 'Code' tab in the left sidebar. The main content area is titled 'Code' and shows a Node.js file named 'Node.js 10'. The code defines a Cloud Function named 'disco_action' that performs a discovery search. The 'Invoke with parameters' button is highlighted in blue. To the right, the 'Activations' panel shows a single activation entry for 'disco_action' with an ID of '8a...5f41' and a duration of '1450 ms' on '5/16/2020, 22:24:23'. The 'Results' section displays the search results, which include matching results, categories, and concepts. The results show business and industrial energy solutions, business and industrial/green solutions, and business and industrial/business operations.

Now click on endpoints[1] tab

The screenshot shows the IBM Cloud Functions Actions interface. The top navigation bar includes 'IBM Cloud', a search bar, and links for 'Catalog', 'Docs', 'Support', 'Manage', and user profile 'Hashim Irfan Al...'. The main area shows 'Functions / Actions / disco_action' with a 'Web Action' sub-section. On the left, a sidebar lists 'Code', 'Parameters', 'Runtime', and the 'Endpoints' tab, which is highlighted with a blue box and a red number '1'. The 'Web Action' section contains two checkboxes: 'Enable as Web Action' (checked) and 'Raw HTTP handling'. Below these are sections for 'HTTP Method' (ANY), 'Auth' (Public), and 'URL' (https://us-south.functions.cloud.ibm.com/api/v1/web/728c6dcb-3bf4-4108-b946-4e5cc6fc84d6/default/disco_action). Another section for 'REST API' shows 'HTTP Method' (POST), 'Auth' (IAM TOKEN), and 'URL' (https://us-south.functions.cloud.ibm.com/api/v1/namespaces/728c6dcb-3bf4-4108-b946-4e5cc6fc84d6/actions/disco_action). At the bottom, a 'CURL' section provides a command to copy: curl -X POST "https://us-south.functions.cloud.ibm.com/api/v1/namespaces/728c6dcb-3bf4-4108-b946-4e5cc6fc84d6/actions/disco_action". Red numbers '2', '3', and '4' are overlaid on the screenshot to point to specific elements: '2' points to the checked checkbox, '3' points to the URL, and '4' points to the curl command.

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

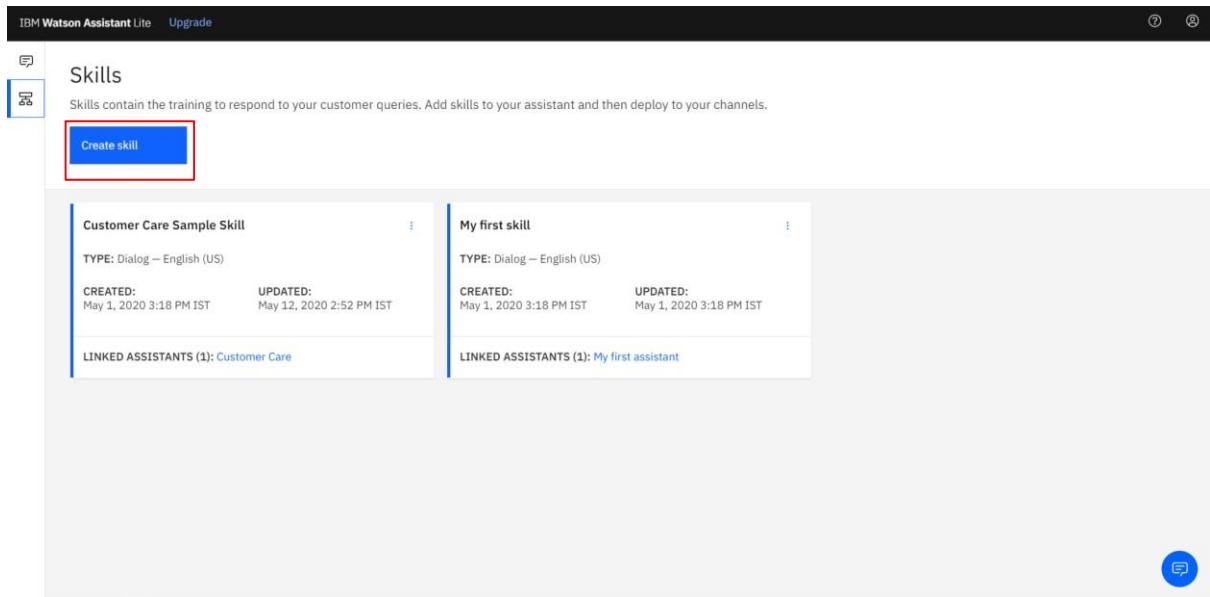
Go back to the IBM Dashboard From the resource list screen, click to open Watson Assistant service.

The screenshot shows the IBM Cloud Resource list interface. A service named "Watson Assistant-61" is listed as active. The service card includes sections for "Manage" (Service credentials, Plan, Connections), "Start by launching the tool" (Launch Watson Assistant button, Getting started tutorial, API reference), and "Plan" (Lite, Upgrade). The "Credentials" section displays an API key and a URL. A "Feedback" button is visible on the right side of the card.

Click on **Launch Watson Assistant** to launch Watson Assistant.

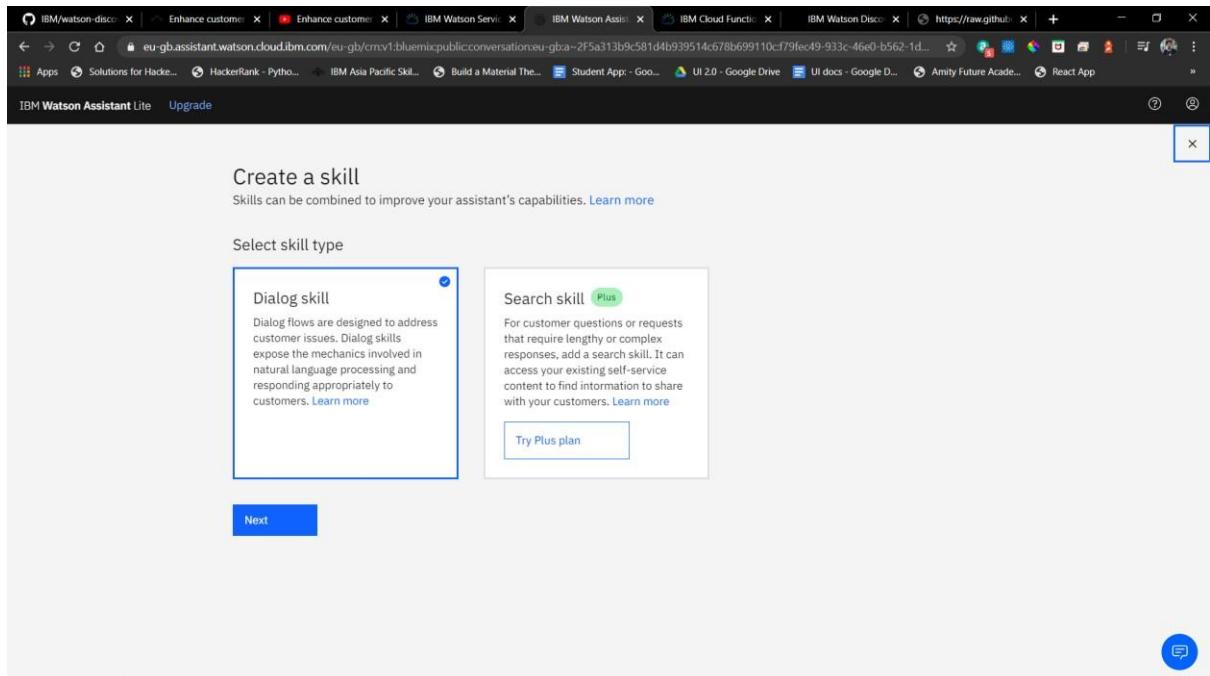
The screenshot shows the "IBM Watson Assistant Lite" service details page. It features a sidebar with "Assistants" and a "Create assistant" button, which is highlighted with a red box. Below the sidebar, there are two assistant cards: "Customer Care" and "My first assistant". Each card displays its name, a brief description, and metrics for Skills and Integrations. A blue speech bubble icon is located at the bottom right of the page.

Click on the skills tab



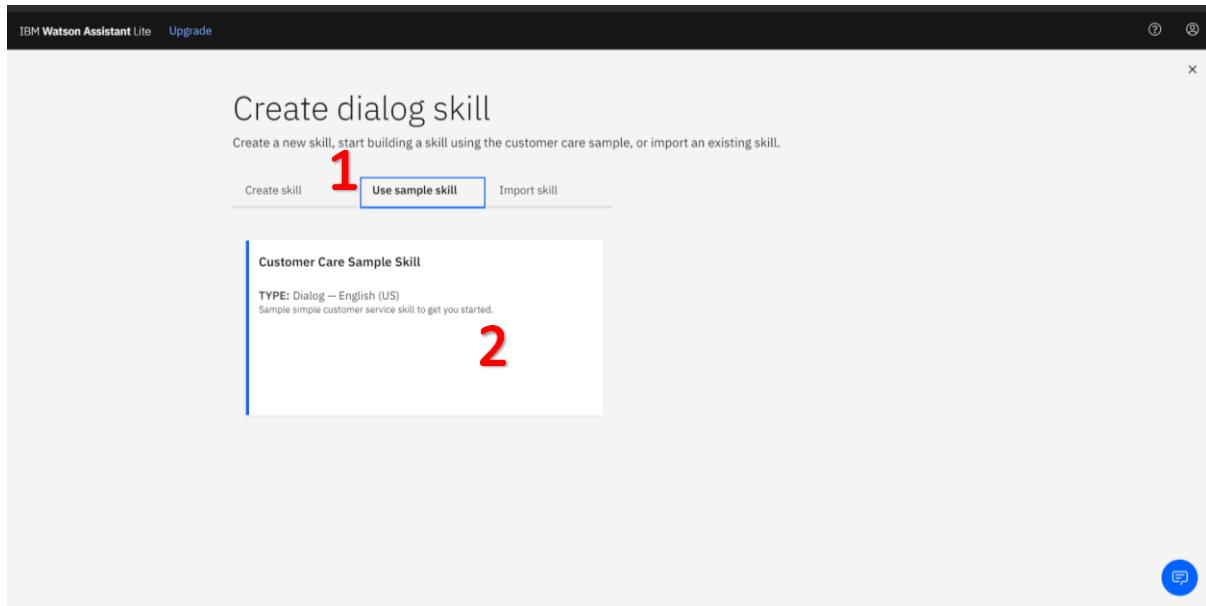
The screenshot shows the 'Skills' section of the IBM Watson Assistant Lite interface. At the top, there's a header bar with the IBM Watson Assistant Lite logo and an 'Upgrade' link. Below the header, the word 'Skills' is displayed next to a small icon. A red box highlights the 'Create skill' button, which is located in a blue-bordered box. Below the button, there are two skill cards: 'Customer Care Sample Skill' and 'My first skill'. Each card provides details like type (Dialog – English (US)), creation and update dates, and linked assistants.

Click Create Skill



The screenshot shows the 'Create a skill' wizard. The title 'Create a skill' is at the top, followed by a sub-instruction: 'Skills can be combined to improve your assistant's capabilities. [Learn more](#)'. Below this, there's a heading 'Select skill type'. Two cards are shown: 'Dialog skill' (selected) and 'Search skill'. The 'Dialog skill' card has a blue border and contains a brief description: 'Dialog flows are designed to address customer issues. Dialog skills expose the mechanics involved in natural language processing and responding appropriately to customers.' It also includes a 'Learn more' link. The 'Search skill' card has a green 'Plus' icon and a brief description: 'For customer questions or requests that require lengthy or complex responses, add a search skill. It can access your existing self-service content to find information to share with your customers.' It includes a 'Try Plus plan' button. A 'Next' button is at the bottom of the screen.

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.

The screenshot shows the 'Customer Care Sample Skill' panel with the 'Intents' tab selected. On the left, a sidebar lists 'Entities', 'Dialog', 'Options', 'Analytics', 'Versions', and 'Content Catalog'. The main area displays a table of intents:

| | Intents (10) ↑ | Description | Modified ↑↑ | Examples ↑↑ |
|--------------------------|-------------------------------|---|-------------|-------------|
| <input type="checkbox"/> | #Cancel | Cancel the current request | 15 days ago | 7 |
| <input type="checkbox"/> | #Customer_Care_Appointments | Schedule or manage an in-store appointment. | 15 days ago | 20 |
| <input type="checkbox"/> | #Customer_Care_Store_Hours | Find business hours. | 15 days ago | 48 |
| <input type="checkbox"/> | #Customer_Care_Store_Location | Locate a physical store location or an address. | 15 days ago | 25 |
| <input type="checkbox"/> | #General_Connect_to_Agent | Request a human agent. | 15 days ago | 47 |
| <input type="checkbox"/> | #General_Greetings | Greetings | 15 days ago | 30 |
| <input type="checkbox"/> | #Goodbye | Good byes | 15 days ago | 6 |
| <input type="checkbox"/> | #Help | Ask for help | 15 days ago | 8 |
| <input type="checkbox"/> | #Product_Information | | 4 days ago | 3 |

At the bottom, it says 'Showing 1–10 of 10 intents'. A red box highlights the 'Create intent' button at the top right of the table.

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.

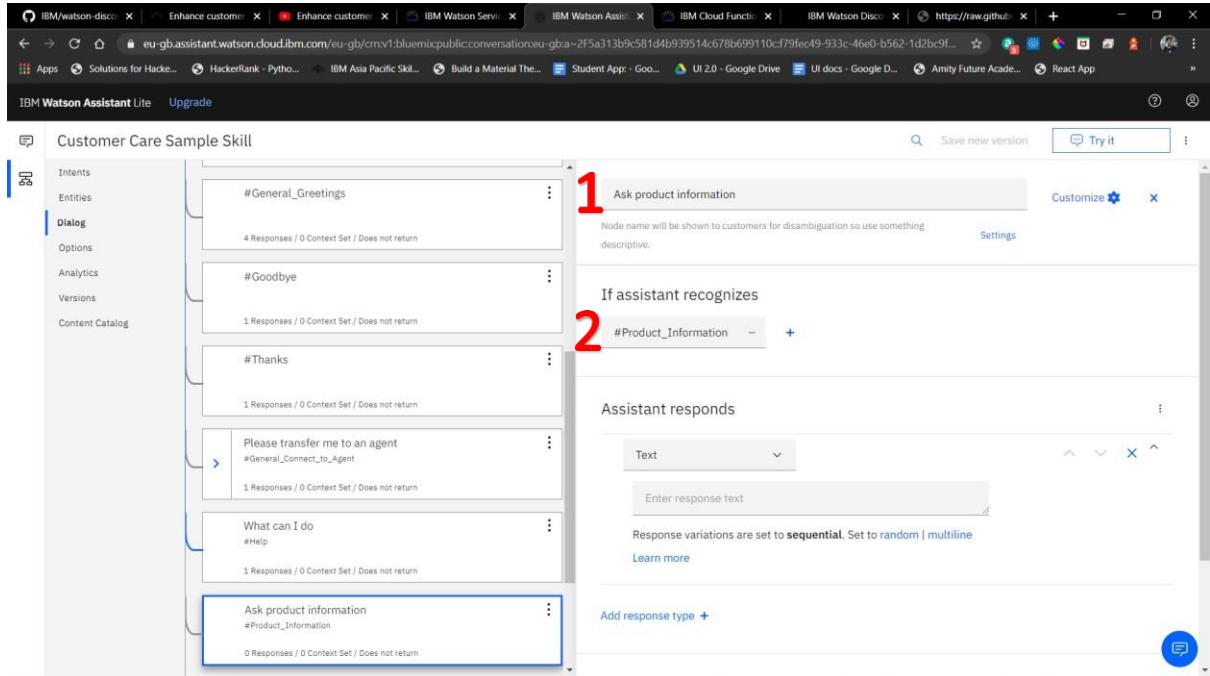
Name the intent #Product_Information, and at a minimum, enter the following example questions to be associated with it.

The screenshot shows the IBM Watson Assistant Lite interface. At the top, there's a header with 'IBM Watson Assistant Lite' and 'Upgrade' buttons. Below the header, the title bar says '#Product_Information'. On the right side of the title bar, there are buttons for 'Last updated: 4 days ago', a download icon, a search icon, and a 'Try it' button. The main area has sections for 'Intent name' (containing '#Product_Information'), 'Description (optional)' (with placeholder 'Add a description to this intent'), and 'User example' (with placeholder 'Type a user example here, e.g. I want to pay my credit card bill'). Below these sections is a button bar with 'Add example' and 'Show recommendations'. A list of user examples is shown, each with a checkbox, a title, and a timestamp ('4 days ago'). The list starts with 'How do i access the settings' and 'How do i set the time'. At the bottom of the list, it says 'Showing 1-3 of 3 examples'.

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

The screenshot shows the 'Customer Care Sample Skill' dialog in the IBM Watson Assistant Lite interface. The left sidebar has tabs for 'Intents', 'Entities', 'Dialog' (which is selected and highlighted in blue), 'Options', 'Analytics', 'Versions', and 'Content Catalog'. In the main area, there's a 'Dialog' section with a tree view. The root node is 'What can I do?'. Below it are several child nodes: 'Opening welcome', 'What are your hours? #Customer_Care_Store_Hours', 'Where are you located? #Customer_Care_Store_Location', 'I want to make an appointment #Customer_Care_Appointments', '#General_Greetings', and '#Goodbyes'. Each node has a small icon and three dots to its right. Above the tree, there are buttons for 'Add node' (highlighted in blue), 'Add child node', and 'Add folder'. On the right side, there are buttons for 'Save new version' and 'Try it'.

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.

Customer Care Sample Skill

Webhooks

A webhook is a mechanism that allows you to call out to an external program based on events in your dialog.

Webhook setup

Specify the request URL for an external API you want to be able to invoke from dialog nodes. Watson will call this URL when configured to do so from a dialog node. [Learn more](#)

URL

`https://us-south.functions.cloud.ibm.com/api/v1/web/728c6db-3bf4-410e...`

Headers

Add HTTP headers for authorization or any other parameters required for invoking the webhook.

| Header name | Header value |
|-------------|-------------------|
| Add header | Add authorization |

Next step

To trigger this webhook from an individual dialog node, enable webhooks from the Customize page of the node. [Go to dialog](#)

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

Customer Care Sample Skill

Dialog

Customize "Ask product information"

Enable this to gather the information your bot needs to respond to a user within a single node.

Prompt for everything

Enable this to ask for multiple pieces of information in a single prompt, so your user can provide them all at once and not be prompted for them one at a time.

Webhooks

Enable this setting to send a POST request from this dialog node to the webhook URL. The URL and headers are defined in the Webhooks settings of the Options tab. After you enable this setting, the Multiple conditional responses setting is enabled automatically to support adding a response to show when the request is successful and another response to show if the request fails. [Learn more](#)

Webhook URL Your webhook URL is configured.

Apply

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 Lite interface. On the left, the 'Dialog' tab is selected, displaying a list of nodes: '#Goodbye', '#Thanks', 'Please transfer me to an agent', 'What can I do', and 'Ask product information'. The 'Ask product information' node is highlighted with a blue border. A red number '1' is placed next to the 'Return variable' field, which contains '\$webhook_result_1'. A red number '2' is placed next to the 'Value' field, which contains '<?input.text?>'. On the right, the 'Then callout to my webhook' section is shown, with a 'Webhook URL' status message indicating it is configured. A 'Try it' button is located at the top right of the panel.

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 same interface as above, but with additional configuration in the 'Assistant responds' section. Under 'If assistant recognizes', there are two entries: '1 \$webhook_result_1' and '2 anything_else'. Under 'Respond with', there are two corresponding entries: '\$webhook_result_1' and 'Try again later'. The second row ('anything_else') and its response ('Try again later') are highlighted with a red box. The 'Try it' button is also visible at the top right.

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

The screenshot shows the IBM Watson Assistant Lite interface. On the left, the sidebar includes options like Intents, Entities, Dialog (which is selected), Options, Analytics, Versions, and Content Catalog. The main workspace displays a dialog node named "Ask product information". The "If assistant recognizes" section contains two entries: "1 \$webhook_result_1" and "2 anything_else". The "Respond with" section shows responses: "\$webhook_result_1" for the first entry and "Try again later" for the second. A "Customize" button is available for the node. A "Webhook URL" configuration message is displayed: "Webhook URL Your webhook URL is configured. Options". To the right, a "Try it out" window shows a conversation history: "Hello, I'm your virtual customer care assistant, Gary. I can help with directions to my store, hours of operation and booking an in-store appointment", "Hi", "#General_Greetings", "Hello. Good evening", "How to turn on heater?", "#Product_Information", and a JSON response: {"matching_results":22,"passages":[{"document_id":"35a5eef70d8cc9d70e2b94d2c15e2d1_11","end_offset":240,"field_type","text","passage_score":0,"passage_text":"I"}]. Below the conversation is a note: "Use the up key for most recent". At the bottom, there's a text input field: "Enter something to test your assistant" and a blue "Send" button.

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

This screenshot shows the same workspace as the previous one, but with a visible "Context variables" panel on the right. The panel lists four variables: \$timezone, \$no_reservation, \$webhook_result_1, and \$Enter_variable_name. \$timezone is set to "Asia/Calcutta", \$no_reservation is set to "true", and \$webhook_result_1 contains the JSON response from the Discovery query. There is also a "Remove all context variables" button at the bottom of the panel.

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]

The screenshot shows the 'Skills' section of the IBM Watson Assistant Lite interface. On the left, there's a card for 'Customer Care Sample Skill' with details like TYPE: Dialog – English (US), CREATED: May 1, 2020 3:18 PM IST, and UPDATED: May 1, 2020 3:18 PM IST. It also lists LINKED ASSISTANTS (1): Customer Care. A context menu is open over this card, with the 'View API Details' option highlighted. A red '1' is placed above the menu, and a red '2' is placed on the 'View API Details' button. To the right, another skill card is visible: 'My first skill' with similar details.

The Skill ID and API Key is to be noted.

The screenshot shows the 'Skill details' page for the 'Customer Care Sample Skill'. Under the 'Skill name' section, it shows 'Skill name: Customer Care Sample Skill' and 'Skill ID: 617d5db3-c85f-47b5-8f90-35db631250aa' (which is highlighted with a red box). Below that, it shows 'Legacy v1 workspace URL: https://api.eu-gb.assistant.watson.cloud.ibm.com/instances/f79fec49-933c-46e0-b562-1d2bc9f497b5/v1/workspaces/617d5db3-c85f-47b5-8f90-35db631250aa/message'. Under the 'Service credentials' section, it shows 'Service credentials name: Auto-generated service credentials' and 'API key: JNnaA2SthfUKDFm50ct6g6D0_cWAmw69V6yrbdk_UIpN' (which is highlighted with a red box).

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

The screenshot shows the IBM Cloud Watson Assistant resource details page. The 'Service credentials' tab is selected, indicated by a red number 1. Below it, a table lists an auto-generated service credential. The 'url' field in the JSON data is highlighted with a red box, showing the full API endpoint: `"url": "https://api.eu-gb.assistant.watson.cloud.ibm.com/instances/f77fec49-933c-46e0-b562-1d2bc9f497b5"`.

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].

The screenshot shows the IBM Cloud Catalog search results for 'node-red'. A red number 1 is placed over the search bar. The search results show one result for the 'Node-RED App', which is highlighted with a red box and labeled with a red number 2. The tile provides information about the app being an IBM Software Developer Tools Web and Application category, and links to Starter kits and IBM Kubernetes Service.

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

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

The screenshot shows the IBM Cloud interface for the Node-RED Starter Kit. At the top, there's a navigation bar with 'IBM Cloud' and a search bar. Below the navigation is a breadcrumb trail: Catalog / Create app / Node-RED. The main content area has tabs for 'About' and 'Create'. A large red number '1' is overlaid on the 'Create' tab. The 'About' tab section includes 'Details' (Author: IBM, Updated: 2/11/2020, Type: Starter kit), 'Source code' (GitHub link), and 'Helpful links' (Tutorial link). The 'Create' tab section is titled 'Overview' and contains text about the pre-configured Node-RED application, including a Cloudant service. It also lists 'This starter kit will help you' with three bullet points: 'Generate an application with Node-RED', 'Generate an application with files for deploying to Cloud Foundry or a DevOps Pipeline', and 'Connect to provisioned services'. Below this is a 'What's included?' section showing a Cloudant icon with the text 'Cloudant Free to start View pricing'. At the bottom is a blue 'Get started' button. On the right side of the page, there are 'ASK A QUESTION' and 'FEEDBACK' buttons.

Click on Create [1].

The screenshot shows the 'App details' page for the Node-RED application. The 'Create' tab is selected, indicated by a red number '1'. The 'App details' section includes fields for 'App name' (Node RED KVHBI), 'Resource group' (Default), 'Tags' (Examples: env:dev, version-1), and 'Platform' (Node.js). Below these is a 'Service details' section. On the right side of the page, there are 'ASK A QUESTION' and 'FEEDBACK' buttons.

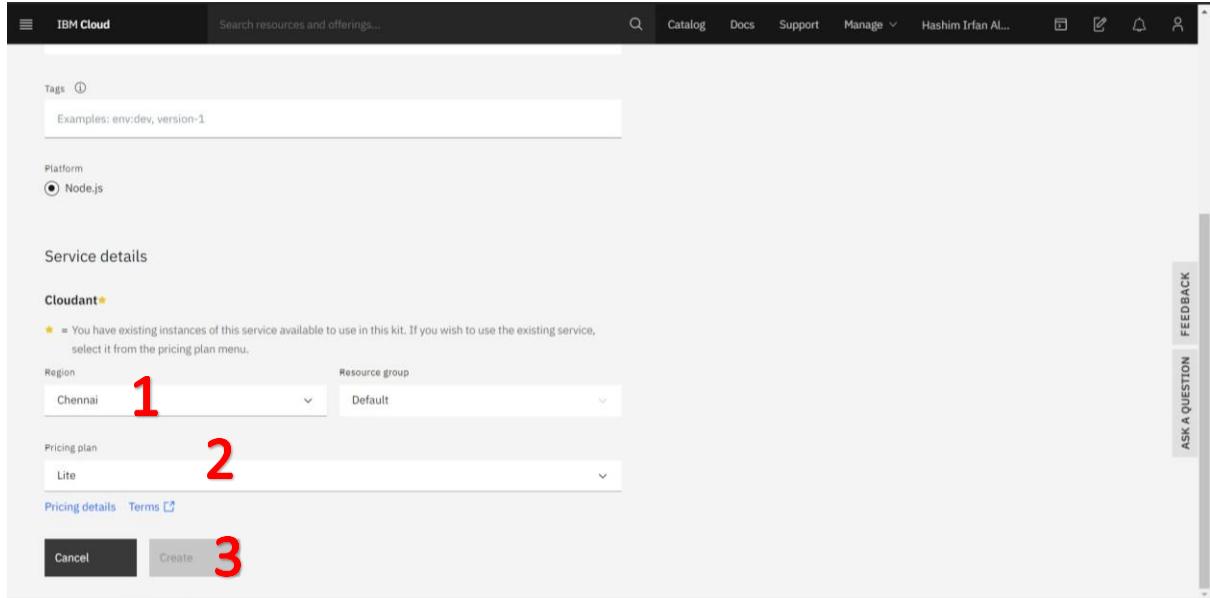
Now you need to configure the Node-RED Starter application.

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.

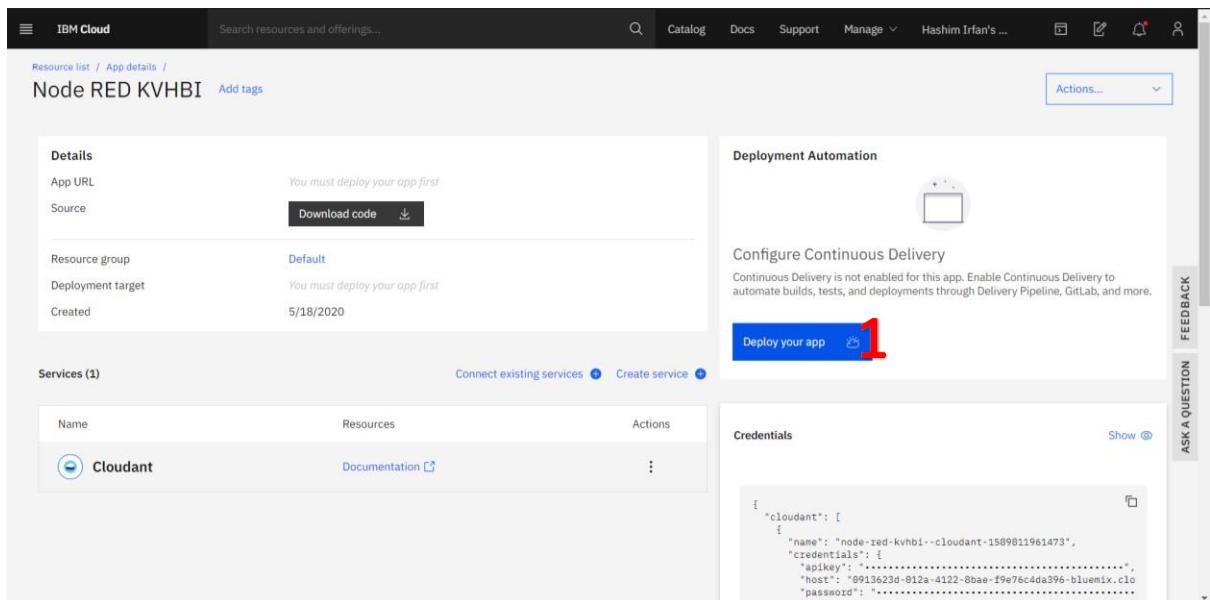
The Node-RED Starter application requires an instance of the Cloudant database service to store your application flow configuration. To do this,

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.

If you have already got an instance, you will be able to select it from the Pricing plan select box [2]. Click the Create button [3] to continue. This will create your application, but it is not yet deployed to IBM Cloud.



At this point, you have created the application and the resources it requires, but you have not deployed it anywhere to run, so this step shows how to setup the Continuous Delivery feature that will deploy your application into the Cloud Foundry space of IBM Cloud. Click on Deploy your App[1]



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.

The screenshot shows the IBM Cloud interface for creating an app. The top navigation bar includes 'IBM Cloud', 'Search resources and offerings...', 'Catalog', 'Docs', 'Support', 'Manage', and 'Hashim Irfan's ...'. Below the search bar, it says 'Resource list / App details' and shows 'Node RED KVHBI'. Under 'Deployment Automation', there are two tabs: 'Select the deployment target' (selected) and 'Configure the DevOps toolchain'. The 'Deployment target' section shows 'Cloud Foundry IBM' as the option, with a note: 'Deploy and run your applications without managing servers or clusters. A Lite plan is available for quick and easy deployment.' The 'IBM Cloud API key' field is empty and highlighted with a red box, with a note: 'The value is required.' To its right is a 'New' button with a red number '1' over it. The right sidebar contains sections like 'Getting started with apps', 'Step 1. Select the deployment target', and 'Before you begin' with instructions about creating a Cloud Foundry org.

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].

The screenshot shows the continuation of the app creation process. The top navigation bar is identical. The main area shows the deployment configuration. The 'Number of instances' is set to 1. The 'Memory allocation per instance' slider is currently at 64 MB, with a red number '1' placed over it. The 'Region' dropdown is set to 'London', with a red number '2' placed over it. The 'Organization' dropdown is set to 'hashimirfan99@gmail.com' and the 'Space' dropdown is set to 'dev'. At the bottom, the 'Host' field is 'node-red-kvhbi' and the 'Domain' field is 'eu-gb.mybluemix.net'. The 'Cancel' and 'Next' buttons are at the bottom, with a red number '3' placed over the 'Next' button.

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

The screenshot shows the 'Resource list / App details' page for 'Node RED KVHBI'. At the top, there are two radio buttons: 'Select the deployment target' (unchecked) and 'Configure the DevOps toolchain' (checked). Below this, a section titled 'Configure the DevOps toolchain' asks for a name and region. The 'DevOps toolchain name' field contains 'NodeREDKVHBI'. The 'Region' dropdown is set to 'Dallas' (Step 1). At the bottom are 'Back' and 'Create' buttons, with 'Create' highlighted in red (Step 2). To the right, a sidebar titled 'Getting started with apps' provides instructions for Step 2.

This will take you back to the application details page.

The screenshot shows the 'Resource list / App details / Node RED KVHBI' page. It includes sections for 'Details' (App URL, Source, Resource group, Deployment target, Created), 'Services (1)' (Cloudant), and 'Deployment Automation' (Continuous Delivery configuration). A 'Credentials' section displays JSON code for a Cloudant service. A sidebar on the right offers 'ASK A QUESTION' and 'FEEDBACK' options.

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.

Details

App URL: You must deploy your app first
Source: <https://us-south.git.cloud.ibm.com/hashimirfan99/NodeREDKVHBI>

Resource group: Default
Deployment target: You must deploy your app first
Created: 5/18/2020

Services (1)

| Name | Resources | Actions |
|----------|---------------|---------|
| Cloudant | Documentation | ⋮ |

Deployment Automation

Name: NodeREDKVHBI
Location: Dallas
Tool integrations:

Delivery Pipelines

Name: NodeREDKVHBI
Status: In progress
Last input: Last commit by IBM Cloud (1 minute ago)
Clone from zip

Credentials

```
{
  "cloudant": [
    {
      "name": "node-zed-kvhbi--cloudant-1589811961473",
      "credentials": {
        "apikey": "...",
        "host": "0013423d-012a-4122-9bae-f9e74c4de204 bluemix.cln"
      }
    }
  ]
}
```

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.

Details

App URL: You must deploy your app first
Source: <https://us-south.git.cloud.ibm.com/hashimirfan99/NodeREDKVHBI>

Resource group: Default
Deployment target: You must deploy your app first
Created: 5/18/2020

Services (1)

| Name | Resources | Actions |
|----------|---------------|---------|
| Cloudant | Documentation | ⋮ |

Deployment Automation

Name: NodeREDKVHBI
Location: Dallas
Tool integrations:

Delivery Pipelines

Name: NodeREDKVHBI
Status: Success
Last input: Last commit by IBM Cloud (4 minutes ago)
Clone from zip

Credentials

```
{
  "cloudant": [
    {
      "name": "node-zed-kvhbi--cloudant-1589811961473",
      "credentials": {
        "apikey": "...",
        "host": "0013423d-012a-4122-9bae-f9e74c4de204 bluemix.cln"
      }
    }
  ]
}
```

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].

The screenshot shows the IBM Cloud Resource list interface. On the left, there's a sidebar with navigation icons and a tree view of resources. The main area displays a table with columns: Name, Group, Location, Offering, Status, and Tags. A single row is selected, highlighted with a red box labeled '2'. The row details are: Name - Node RED KVHBI, Group - hashimirfan99@gmail.com / dev, Location - London, Offering - SDK for Node.js™, Status - Started, and Tags - none. A red box labeled '1' highlights the 'Cloud Foundry apps' section in the sidebar.

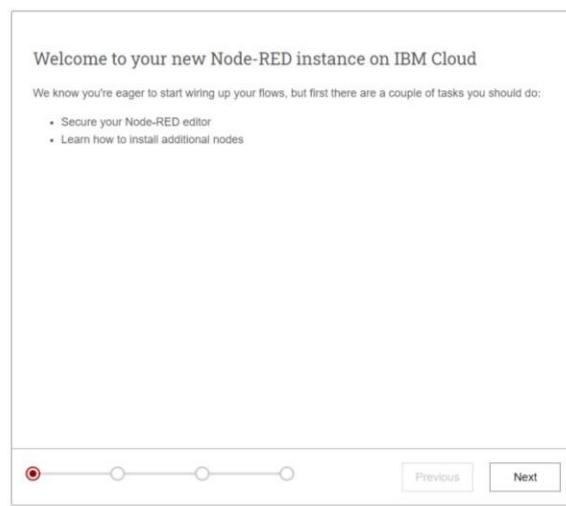
Click on this Cloud Foundry app entry to go to your deployed application's details page.

The screenshot shows the IBM Cloud App Details page for the 'Node RED KVHBI' application. The top bar includes the application name, status (Running), and links to Visit App URL and Add tags. The main content area has tabs for Overview, Runtime, Connections, and Logs. The Overview tab is selected, showing the following details:

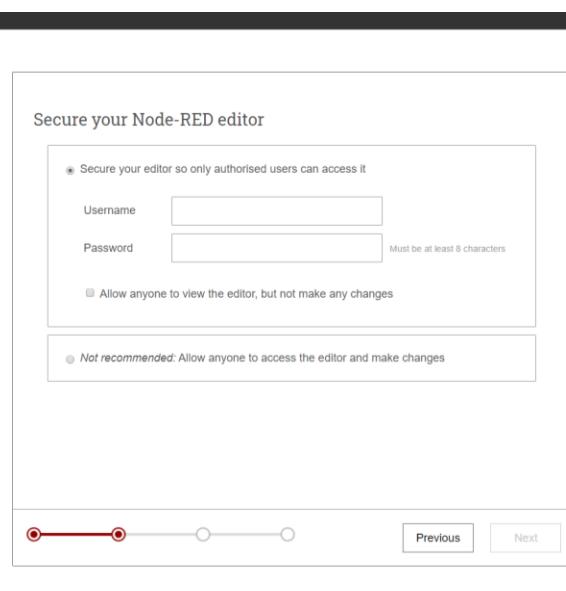
- Instances**: Health is 100%, with 1/1 instance(s) running. Memory allocation is set at 256 MB.
- Runtime**: Shows a donut chart with 256 Total MB allocation, where 1.75 GB is still available. A red box labeled '1' points to the 'Edit' link next to the runtime settings.
- Runtime cost**: Current charges for billing period May 1, 2020 - May 31, 2020 are \$0.00.
- Connections**: There is 1 connection listed: node-red-kvhbi--cloudant-1589811961473-32813.

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.

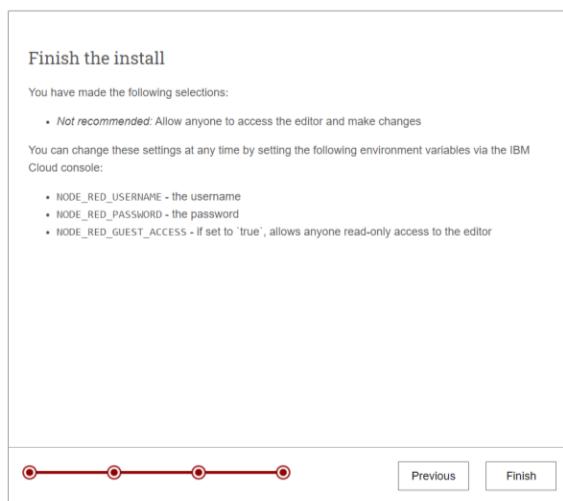
For more information, follow this tutorial on [IBM Developer](#).



Previous

Next

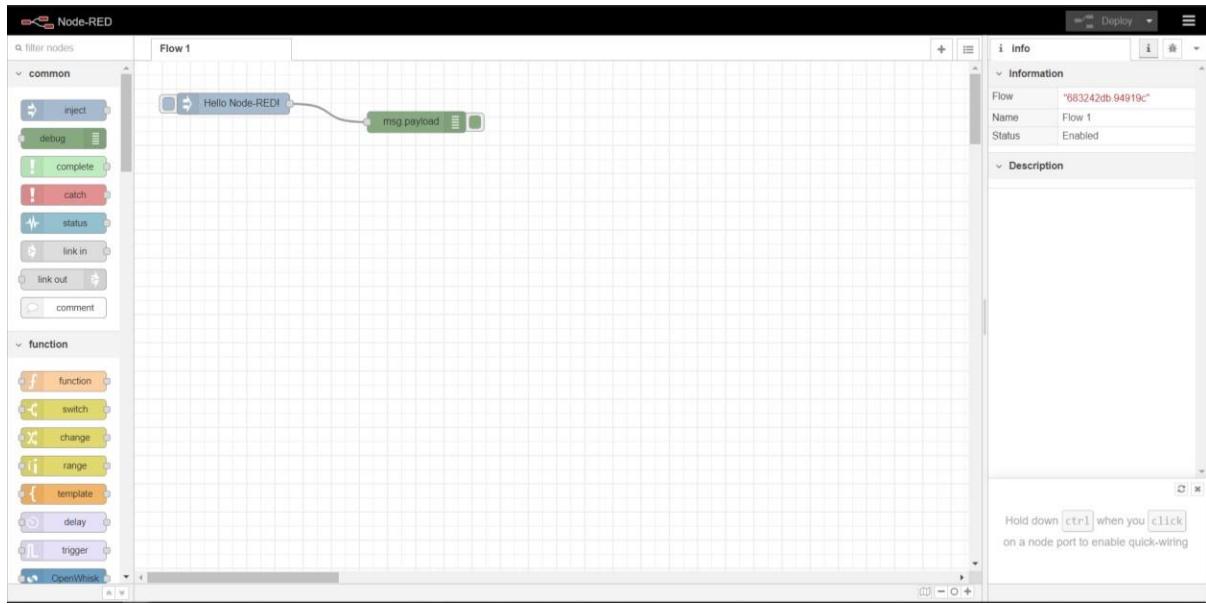
Click Next.



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.

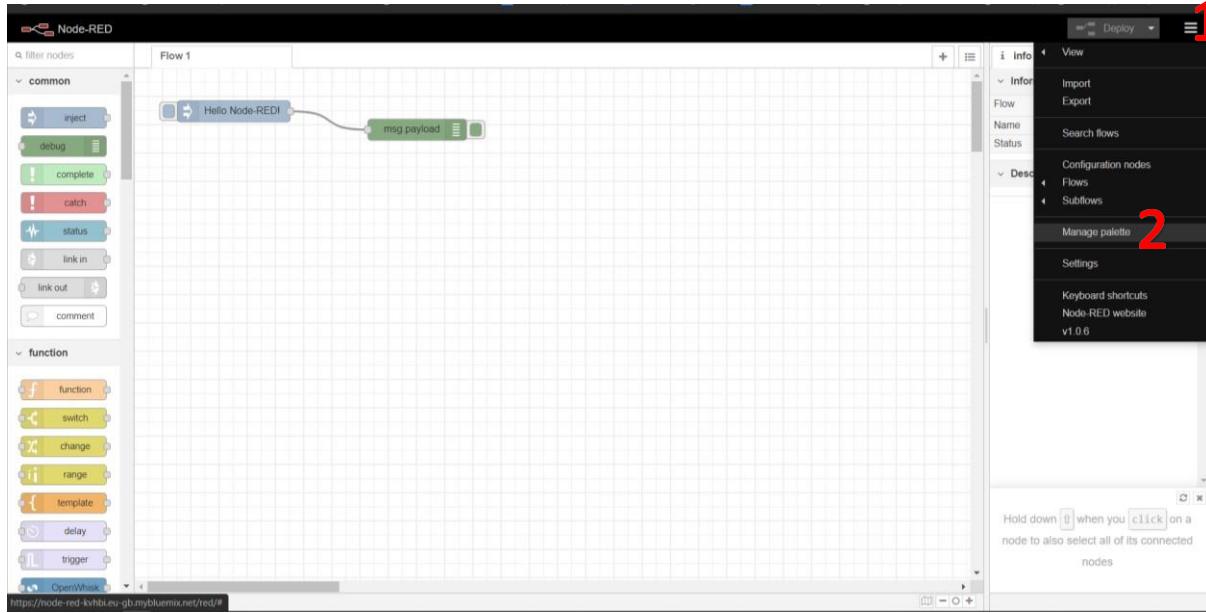
A screenshot of the Node-RED application interface. The header is red with the text 'Node-RED' and 'Flow-based programming for the Internet of Things'. The main content area has a light gray background. It contains text about Node-RED being a tool for wiring together hardware devices, APIs, and online services. It mentions that this instance is running on IBM Cloud. There is a red-bordered button labeled 'Go to your Node-RED flow editor'. Below it is a link 'Learn how to customise Node-RED'. At the bottom, there is a section titled 'Customising your instance of Node-RED' with a note about starting to create flows and links to documentation and the Node-RED website.



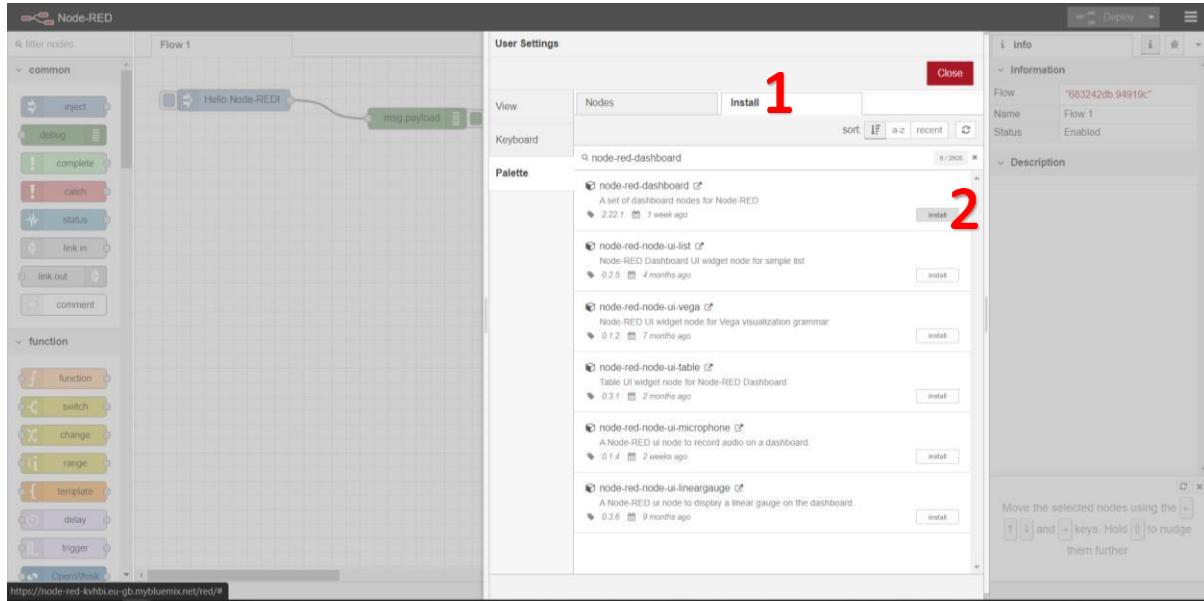
The Node-RED editor opens showing the default flow.

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]

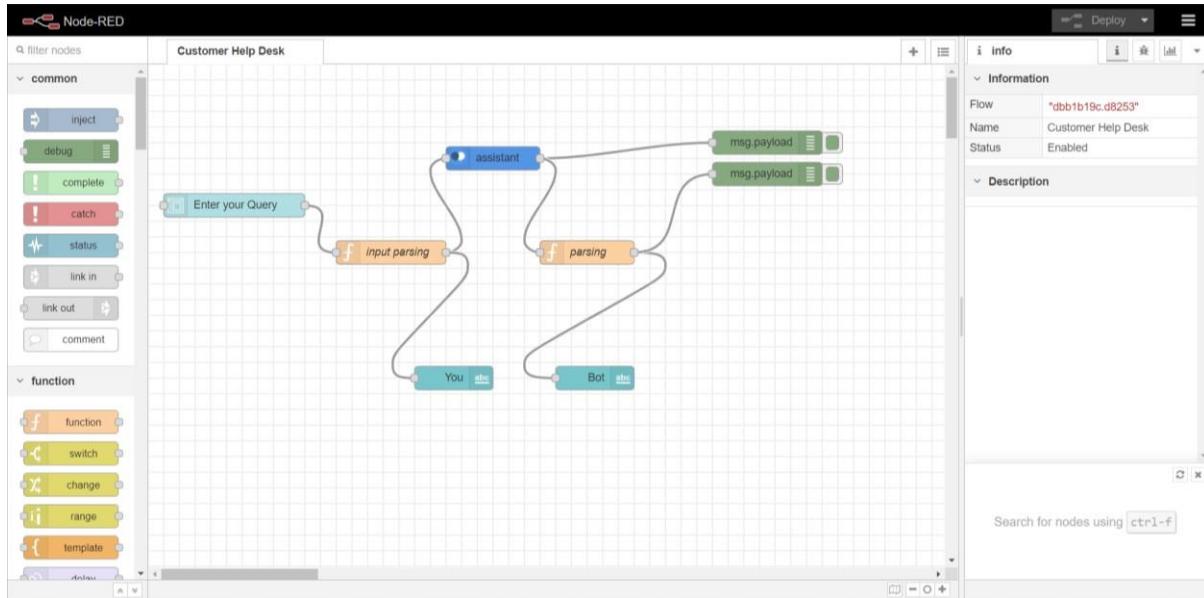


Go to Install Tab [1] and search for node-red-dashboard and Install [2] it.

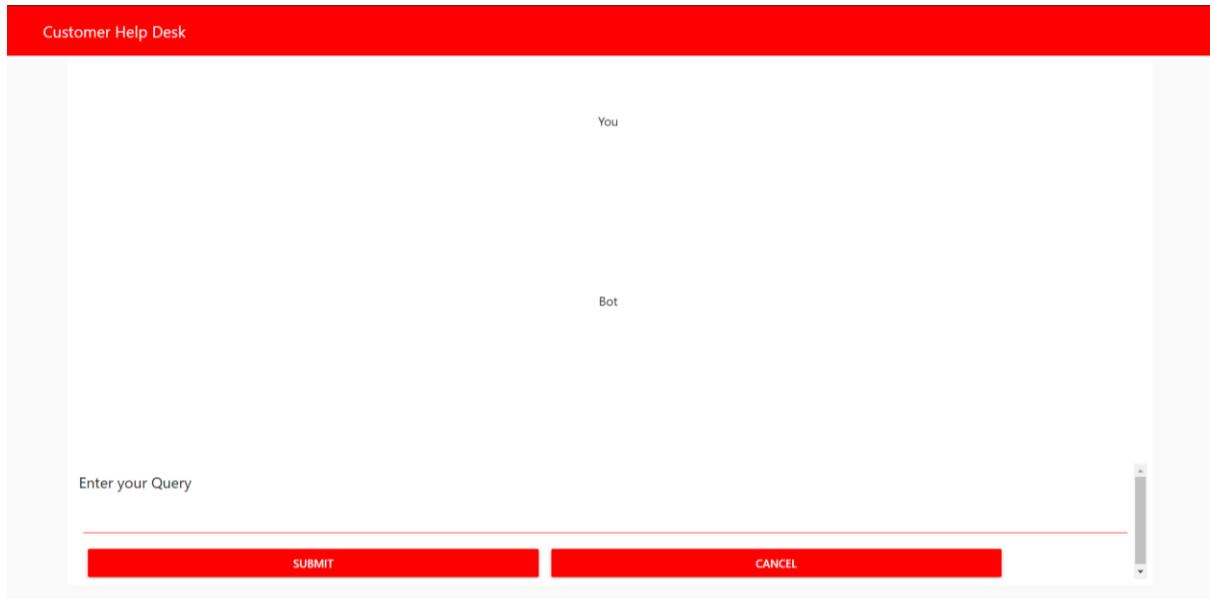


Using the nodes in the palette, Configure the required nodes and build web dashboard in Node-RED.

7. Deploy and Run the application

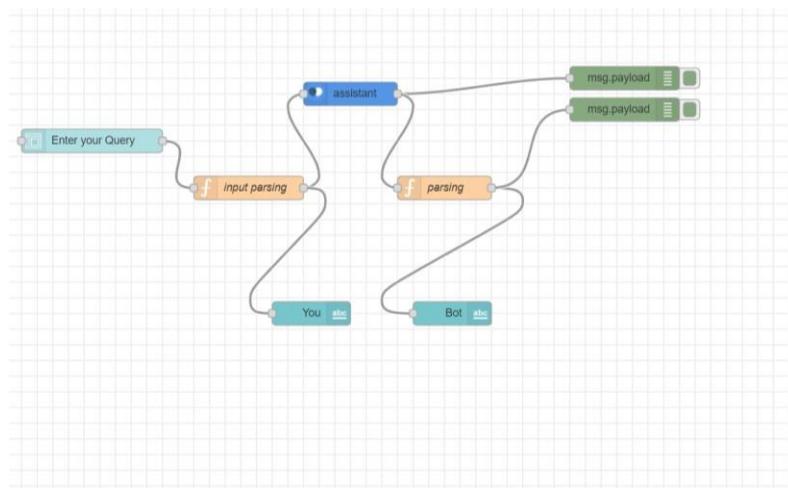


After Deploying the App, Run it.

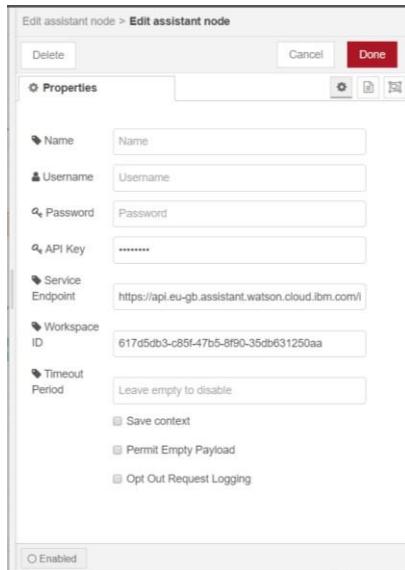


5.

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 parsing, use the code below:

```
msg.payload=msg.payload.text;  
return msg;
```

For Function Node named parsing, use the code below:

```
msg.payload.text="";  
if(msg.payload.context.webhook_result_1){  
    for(var i in msg.payload.context.webhook_result_1.results){  
  
        msg.payload.text=msg.payload.text+"\n"+msg.payload.context.webhook_result_1.re  
sults[i].text;  
    }  
    msg.payload=msg.payload.text;  
}
```

```

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

```

6.

RESULT

The screenshot shows a chatbot interface titled "Customer Help Desk". The conversation log is as follows:

- You: Hi
- Bot: Hello. Good afternoon
- User input field: Enter your Query
- User input: Hi
- Buttons: SUBMIT (red), CANCEL (white)

The second part of the screenshot shows a similar interface for a query about "How to set time?". The conversation log is as follows:

- You: How to set time?
- Bot: Smart recovery lets your ecobee3 learn how your heating and cooling system works, taking into account weather and historical operating performance so that your home reaches the scheduled set point at the time in which the change occurs (i.e. not afterwards). For example, if you wake up at 6:00 AM, you do not need to schedule your Home period to start at 5:30 AM. Smart Recovery will start the HVAC equipment to ensure that at 6:00 AM, the house is at your desired temperature. On Thermostat: 1. Select Main Menu > Settings > Preferences 2. Select Heating Smart Recovery or Cooling Smart Recovery. 3. Touch Enable or Disable. On Web: 1. Select Settings tile. 2. Select Preferences. 3. Select Smart Recover Heat Mode or Smart Recovery Cool Mode. 4. Select Enable or Disable. You can configure how long it takes for the standby screen to activate after you have finished using the thermostat. On Thermostat: 1. Select Main Menu > Settings > Preferences 2. Select Active to standby screen timer. 3. Adjust the activation time as required. On Web: 1. Select Settings tile. 2. Select Preferences. 3. Select Active to Standby Sleep Timer. 4. Select the length of time before the screen sleeps by sliding the values left or right. You can configure how long a manual change to the set point will remain in effect. On Thermostat: 1. Select Main Menu > Settings > Preferences 2. Select Hold action. 3. Select the hold action from the list: □ 2 hours 4 hours □ Until the next scheduled activity □ Until you change it (default value) Decide at time of change On Web: 1. Select Settings tile. 2. Select Preferences. 3. Select Hold Action. 4. Select the hold action from the list: □ 2 hours 4 hours □ Until the next scheduled activity Until you change it (default value) □ Decide at time of change
- User input field: Enter your Query
- User input: How to set time?
- Buttons: SUBMIT (red), CANCEL (white)

Chatbot Link:

https://node-red-qeyad.eu-gb.mybluemix.net/ui/#!/0?socketid=GDfQDCBt_qiix49XAAAE

Explanation Video Link:

<https://youtu.be/r4FZyhfafoU>

7.

ADVANTAGES & DISADVANTAGES

Advantages:

- Faster Customer Service
- Increased Customer Satisfaction
- Lower Labour 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

disco_action.js(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  
 *  
 * 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);
});
}
);
}
}

```

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

```
{
  "intents": [
    {
      "intent": "General_Greetings",
      "examples": [
        {
          "text": "Good morning"
        },
        {
          "text": "Good day"
        },
        {
          "text": "hiya"
        },
        {
          "text": "yo"
        },
        {
          "text": "How are things going?"
        },
        {
          "text": "How are you today?"
        },
        {
          "text": "How have you been?"
        },
        {
          "text": "hi"
        },
        {
          "text": "How r u?"
        },
        {
          "text": "Looking good eve"
        },
        {
          "text": "Hey how are you doing"
        }
      ]
    }
  ]
}
```

```
},
{
  "text": "Hello I am looking for some help here"
},
{
  "text": "Hello"
},
{
  "text": "Hello Agent"
},
{
  "text": "Have you been well?"
},
{
  "text": "Greetings"
},
{
  "text": "Good to see you"
},
{
  "text": "Good evening"
},
{
  "text": "Hey you"
},
{
  "text": "How is it going?"
},
{
  "text": "You there"
},
{
  "text": "Who is this?"
},
{
  "text": "What's up?"
},
{
  "text": "What's new?"
},
{
  "text": "Hi there"
},
{
  "text": "Hey twin"
},
{
  "text": "Hi advisor"
},
{
  "text": "Ok take me back"
},
{
  "text": "Hey there"
},
{
  "text": "Hey there all"
}
],
"description": "Greetings"
},
```

```
{
  "intent": "Help",
  "examples": [
    {
      "text": "what can i say"
    },
    {
      "text": "can you help"
    },
    {
      "text": "can you assist me"
    },
    {
      "text": "help"
    },
    {
      "text": "help me decide"
    },
    {
      "text": "help me"
    },
    {
      "text": "i need assistance"
    },
    {
      "text": "what can i do"
    }
  ],
  "description": "Ask for help"
},
{
  "intent": "Customer_Care_Store_Hours",
  "examples": [
    {
      "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 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": "how late are you open"
    },
    {
      "text": "Are you open on Sunday"
    }
  ],
  "description": "Get store hours information"
}
}
```

```
{  
    "text": "how late are you open tonight"  
},  
{  
    "text": "are stores open on sunday"  
},  
{  
    "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": "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": "What are your hous?"
},
{
    "text": "What time are you closing today?"
},
{
    "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?"
}
],
"description": "Find business hours."
},
```

```
{
  "intent": "Product_Information",
  "examples": [
    {
      "text": "How do i turn on the heater"
    },
    {
      "text": "How do i set the time"
    },
    {
      "text": "How do i access the settings"
    }
  ],
  "description": ""
},
{
  "intent": "Customer_Care_Appointments",
  "examples": [
    {
      "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"
    },
    {
      "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": "What is the best time to meet with you today?"
    },
    {
      "text": "Is there a specific time you are available to meet with me?"
    },
    {
      "text": "When is the earliest time you can meet with me?"
    },
    {
      "text": "When is the latest time you can meet with me?"
    },
    {
      "text": "Do you have any availability this week?"
    },
    {
      "text": "When is the next available time for an appointment?"
    },
    {
      "text": "Is there any time this week that works for you?"
    },
    {
      "text": "When is the best time for you to meet with me?"
    },
    {
      "text": "What days this week are you available for an appointment?"
    },
    {
      "text": "When is the earliest time you can meet with me this week?"
    },
    {
      "text": "When is the latest time you can meet with me this week?"
    },
    {
      "text": "Do you have any availability this month?"
    },
    {
      "text": "When is the next available time for an appointment this month?"
    },
    {
      "text": "Is there any time this month that works for you?"
    },
    {
      "text": "When is the best time for you to meet with me this month?"
    },
    {
      "text": "What days this month are you available for an appointment?"
    },
    {
      "text": "When is the earliest time you can meet with me this month?"
    },
    {
      "text": "When is the latest time you can meet with me this month?"
    },
    {
      "text": "Do you have any availability this year?"
    },
    {
      "text": "When is the next available time for an appointment this year?"
    },
    {
      "text": "Is there any time this year that works for you?"
    },
    {
      "text": "When is the best time for you to meet with me this year?"
    },
    {
      "text": "What days this year are you available for an appointment?"
    },
    {
      "text": "When is the earliest time you can meet with me this year?"
    },
    {
      "text": "When is the latest time you can meet with me this year?"
    }
  ]
}
```

```
        "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"
}
],
"description": "Schedule or manage an in-store appointment."
},
{
    "intent": "General_Connect_to_Agent",
    "examples": [
        {
            "text": "I want an agent to help me"
        },
        {
            "text": "I want a manager"
        },
        {
            "text": "I want agent"
        },
        {
            "text": "I need to speak to a representative. How would I go about doing so?"
        },
        {
            "text": "Is there anyone there I can actually talk to for real?"
        },
        {
            "text": "I don't want to talk to you"
        },
        {
            "text": "I don't want to talk to a bot."
        },
        {
            "text": "How can I skip the recorded menu and go straight to a live person?"
        },
        {
            "text": "Hi can you transfer me"
        },
        {
            "text": "Do not want a robot?"
        },
        {
            "text": "Customer service representative please."
        },
    ]
}
```

```
        "text": "Could you please transfer me to your master?"  
    },  
    {  
        "text": "Contact person"  
    },  
    {  
        "text": "Connect me to a live operator please."  
    },  
    {  
        "text": "Can you connect me with a real person?"  
    },  
    {  
        "text": "Can you assist me to connect to an agent?"  
    },  
    {  
        "text": "Can I talk to someone?"  
    },  
    {  
        "text": "Can I speak with somebody?"  
    },  
    {  
        "text": "Can I speak to an advisor?"  
    },  
    {  
        "text": "Can I speak to a live 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": "Agent help"  
    },  
    {  
        "text": "talk to a human"  
    },  
    {  
        "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": "I want to speak to a person"
},
{
  "text": "representative"
},
{
  "text": "Put me through to someone"
},
{
  "text": "Pls connect"
},
{
  "text": "Please let me talk to a human being."
},
{
  "text": "Please connect me to a live agent"
},
{
  "text": "Operator please"
},
{
  "text": "Please assist me to get to an agent"
},
{
  "text": "Need help from human"
},
{
  "text": "I would like to speak to someone"
},
{
  "text": "I would like to speak to a human"
},
{
  "text": "I want to talk to a person"
}
],
"description": "Request a human agent."
},
{
  "intent": "Thanks",
  "examples": [
    {
      "text": "thx"
    },
    {
      "text": "thank you very much"
    },
    {
      "text": "that's nice of you"
    },
  ]
}
```

```
{
    "text": "many thanks"
},
{
    "text": "much appreciated"
},
{
    "text": "i appreciate it"
},
{
    "text": "thank you"
},
{
    "text": "thanks"
}
],
"description": "Thanks"
},
{
    "intent": "Goodbye",
    "examples": [
        {
            "text": "so long"
        },
        {
            "text": "good bye"
        },
        {
            "text": "see ya"
        },
        {
            "text": "arrivederci"
        },
        {
            "text": "ciao"
        },
        {
            "text": "bye"
        }
    ],
"description": "Good byes"
},
{
    "intent": "Cancel",
    "examples": [
        {
            "text": "forget it"
        },
        {
            "text": "never mind"
        },
        {
            "text": "cancel that"
        },
        {
            "text": "i changed my mind"
        },
        {
            "text": "cancel the request"
        }
    ]
}
```

```
        "text": "i don't want a table anymore anymore"
    },
    {
        "text": "nevermind"
    }
],
"description": "Cancel the current request"
},
{
"intent": "Customer_Care_Store_Location",
"examples": [
    {
        "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?"
    }
]
```



```

        "value": "no",
        "synonyms": [
            "definitely not",
            "don't think so",
            "dont think so",
            "i think not",
            "nope",
            "not at this time",
            "not now"
        ]
    }
]
},
{
    "entity": "landmark",
    "values": [
        {
            "type": "synonyms",
            "value": "grand central",
            "synonyms": []
        },
        {
            "type": "synonyms",
            "value": "empire state building",
            "synonyms": [
                "empire state",
                "emprise state"
            ],
            {
                "type": "synonyms",
                "value": "times square",
                "synonyms": [
                    "time sqaure",
                    "time square",
                    "times sqaure"
                ]
            }
        ],
        "fuzzy_match": true
    },
    {
        "entity": "holiday",
        "values": [
            {
                "type": "synonyms",
                "value": "christmas eve",
                "synonyms": [
                    "x mas eve",
                    "x-mas eve",
                    "xmas eve"
                ]
            },
            {
                "type": "synonyms",
                "value": "labor day",
                "synonyms": []
            },
            {
                "type": "synonyms",
                "value": "independence day",
                "synonyms": []
            }
        ]
    }
]
```

```
        "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": "memorial day",
    "synonyms": []
},
{
    "type": "synonyms",
    "value": "thanksgiving",
    "synonyms": [
        "turkey day"
    ]
},
{
    "type": "synonyms",
    "value": "christmas",
    "synonyms": [
        "christmas day",
        "x man day",
        "xmas",
        "x mas",
        "x-mas",
        "x-mas day",
        "xmas day"
    ]
},
{
    "type": "synonyms",
    "value": "halloween",
    "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"
    ]
}
```

```
},
{
  "type": "synonyms",
  "value": "new years eve",
  "synonyms": [
    "12-31",
    "12/31",
    "dec 31",
    "dec 31st",
    "new year's eve"
  ]
}
],
{
  "entity": "specialist",
  "values": [
    {
      "type": "synonyms",
      "value": "Robert",
      "synonyms": [
        "bob"
      ]
    },
    {
      "type": "synonyms",
      "value": "Brenda",
      "synonyms": []
    },
    {
      "type": "synonyms",
      "value": "Maria",
      "synonyms": []
    },
    {
      "type": "synonyms",
      "value": "Derrik",
      "synonyms": [
        "derrick",
        "derek",
        "derik",
        "derrik"
      ]
    },
    {
      "type": "synonyms",
      "value": "Nicholas",
      "synonyms": [
        "nick"
      ]
    },
    {
      "type": "synonyms",
      "value": "Barbara",
      "synonyms": [
        "barbra"
      ]
    }
  ]
}
{
```

```
"entity": "phone",
"values": [
  {
    "type": "patterns",
    "value": "US Phone pattern",
    "patterns": [
      "(\\d{3})-(\\d{3})-(\\d{4})"
    ]
  }
],
{
  "entity": "zip_code",
  "values": [
    {
      "type": "patterns",
      "value": "US Zip",
      "patterns": [
        "(\\b|\\s)\\d{5}(\\b|\\s)"
      ]
    }
  ]
},
{
  "entity": "sys-date",
  "values": []
},
],
"metadata": {
  "api_version": {
    "major_version": "v2",
    "minor_version": "2018-11-08"
  }
},
"webhooks": [
  {
    "url": "https://us-south.functions.cloud.ibm.com/api/v1/web/728c6dcbe3bf4-4108-b946-4e5cc6fc84d6/default/disco_action.json",
    "name": "main_webhook",
    "headers": []
  }
],
"dialog_nodes": [
  {
    "type": "response_condition",
    "output": {
      "generic": [
        {
          "values": [
            {
              "text": "Try again later"
            }
          ],
          "response_type": "text",
          "selection_policy": "sequential"
        }
      ]
    },
    "parent": "node_8_1589275295349",
    "conditions": "anything_else",
    "dialog_node": "response_2_1589275317844",
```

```
        "previous_sibling": "response_4_1589275315523"
    },
    {
        "type": "response_condition",
        "output": {
            "generic": [
                {
                    "values": [
                        {
                            "text": "$webhook_result_1"
                        }
                    ],
                    "response_type": "text",
                    "selection_policy": "sequential"
                }
            ]
        },
        "parent": "node_8_1589275295349",
        "conditions": "$webhook_result_1",
        "dialog_node": "response_4_1589275315523"
    },
    {
        "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": {
                "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": {
            "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": "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 <?"
  },
  "parent": "slot_8_1509132875735",
  "metadata": {},
  "event_name": "focus",
  "date.reformatDateTime(\"EEEE\") ?> at <? $time.reformatDateTime(\"h a\") ?>. Is this correct?"
},

```

```

        "dialog_node": "handler_10_1509132875735"
    },
    {
        "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": [
                    "Our hours on <? @sys-date.reformatDateTime(\"EEEE\") ?> are 11am to 6pm."
                ],
                "selection_policy": "sequential"
            }
        },
        "parent": "Hours of Operation",
        "context": {},
        "metadata": {},
        "conditions": "@sys-date.reformatDateTime(\"EEEE\") == \"Saturday\" || @sys-date.reformatDateTime(\"EEEE\") == \"Sunday\"",
        "dialog_node": "node_2_1482424204936",
        "previous_sibling": "node_5_1482426503106"
    },
    {
        "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(\"EEEE\") ?>
from 10am until 8pm"
    ],
    "selection_policy": "sequential"
}
},
"parent": "Hours of Operation",
"context": {},
"metadata": {},
"conditions": "@sys-date.reformatDateTime(\"EEEE\") == \"Monday\" || @sys-
date.reformatDateTime(\"EEEE\") == \"Tuesday\" || @sys-
date.reformatDateTime(\"EEEE\") == \"Wednesday\" || @sys-
date.reformatDateTime(\"EEEE\") == \"Thursday\" || @sys-
date.reformatDateTime(\"EEEE\") == \"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": "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": "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": {
    "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 <? $date.reformatDateTime(\"EEEE\") ?> do you want to come in?"
  },
  "parent": "slot_105_1498132552870",
  "metadata": {},
  "event_name": "focus",
  "dialog_node": "handler_107_1498132552870"
},
{
  "type": "slot",
  "output": {},
  "parent": "Reservation using slots",
  "metadata": {
    "_customization": {}
  },
  "variable": "$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": "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": "response_condition",
    "output": {
        "text": {
            "values": [
                "Let me know how else I can help"
            ],
            "selection_policy": "sequential"
        }
    },
    "parent": "Reservation using slots",

```

```

"context": {},
"metadata": {},
"conditions": "$user_cancelled",
"dialog_node": "node_10_1509697567474",
"previous_sibling": "node_25_1522598839584"
},
{
  "type": "event_handler",
  "output": {
    "text": {
      "values": [
        "OK. Canceling your request..."
      ]
    }
  },
  "parent": "Reservation using slots",
  "context": {
    "date": null,
    "time": null,
    "phone": null,
    "confirm": null,
    "specialist": null,
    "user_cancelled": true
  },
  "metadata": {},
  "next_step": {
    "behavior": "skip_all_slots"
  },
  "conditions": "#Cancel",
  "event_name": "generic",
  "dialog_node": "handler_16_1509133697261",
  "previous_sibling": "handler_3_1501275087289"
},
{
  "type": "slot",
  "output": {},
  "parent": "Reservation using slots",
  "metadata": {
    "_customization": {
      "mcr": true
    }
  },
  "variable": "$time",
  "dialog_node": "slot_105_1498132552870",
  "previous_sibling": "slot_102_1498132501942"
},
{
  "type": "slot",
  "output": {},
  "parent": "Reservation using slots",
  "metadata": {},
  "variable": "$confirm",
  "dialog_node": "slot_8_1509132875735",
  "previous_sibling": "slot_12_1522596437268"
},
{
  "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": "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": "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 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. 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": "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": "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",
  "context": {
    "specialist": "@specialist"
  },
  "metadata": {},
  "conditions": "@specialist",
  "event_name": "input",
  "dialog_node": "handler_13_1522596437268",
  "previous_sibling": "handler_14_1522596437268"
},
{
  "type": "event_handler",
```

```

    "output": {},
    "parent": "slot_12_1522596437268",
    "event_name": "focus",
    "dialog_node": "handler_14_1522596437268"
},
{
    "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_8_1589275295349",
    "disambiguation_opt_out": true
},
{
    "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",
    "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": "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": "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",
    "output": {},
    "metadata": {},
    "conditions": "#General_Greetings",
    "digress_in": "does_not_return",
    "dialog_node": "node_13_1502484041694",
    "previous_sibling": "Reservation using slots"
},
{
    "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": "standard",
  "title": "Ask product information",
  "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_8_1589275295349",
  "previous_sibling": "node_4_1570050459690"
},
{
  "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",
  "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",
  "title": "Opening",
  "output": {

```

```

    "text": {
        "values": [
            "Hello, I'm your virtual customer care assistant, Gary. 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 parsing(Node Red Function Code)

```

msg.payload=msg.payload.text;
return msg;

```

parsing(Node Red Function Code)

```

msg.payload.text="";
if(msg.payload.context.webhook_result_1) {

```

```
for(var i in msg.payload.context.webhook_result_1.results){  
  
msg.payload.text=msg.payload.text+"\n"+msg.payload.context.webhook_result_1  
.results[i].text;  
}  
  
msg.payload=msg.payload.text;  
}  
  
else  
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