

Intelligent Customer Help Desk With Smart Document Understanding

Category: Artificial Intelligence

Skills Required:

python, IBM Cloud, IBM Watson

Project Description:

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

In this project, there will be another option. If the customer question is about the operation of a device, the application shall pass the question onto Watson Discovery Service, which has been pre-loaded with the device's owners manual. So now, instead of "Would you like to speak to a customer representative?" we can return relevant sections of the owners 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 owners manual is important and what is not. This will improve the answers returned from the queries.

Scope of Work

- Create a customer care dialog skill in Watson Assistant
- Use Smart Document Understanding to build an enhanced Watson Discovery collection
- Create an IBM Cloud Functions web action that allows Watson Assistant to

post queries to Watson Discovery

- Build a web application with integration to all these services & deploy the same on IBM Cloud Platform

Project Process:

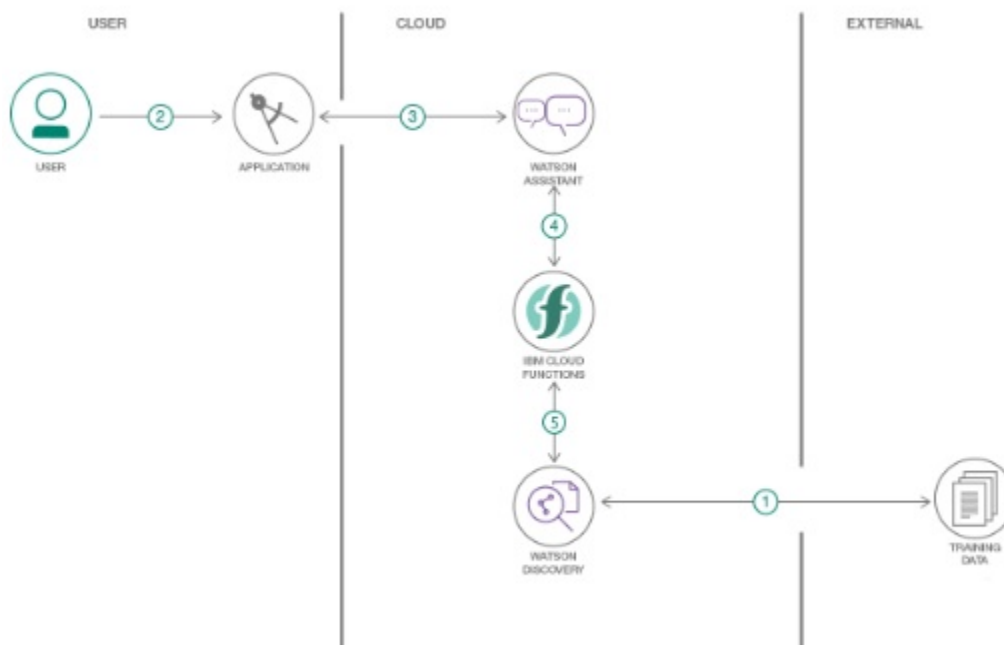
In this code pattern, we walk you through a working example of a web app that utilizes multiple Watson services to create a better customer care experience.

Using the Watson Discovery Smart Document Understanding (SDU) feature, we will enhance the Discovery model so that queries will be better focused to only search the most relevant information found in a typical owner's manual.

Using Watson Assistant, we will use a standard customer care dialog to handle a typical conversation between a customer and a company representative. When a customer question involves operation of a product, the Assistant dialog will communicate with the Discovery service using a webhook.

The webhook will be created by defining a `web action` using IBM Cloud Functions.

Flow:



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

Steps:

1. Create IBM Cloud services
2. Configure Watson Discovery
3. Create IBM Cloud Functions action
4. Configure Watson Assistant
5. Integrate the Services to NODE RED
6. Get IBM Cloud services credentials and add to .env file
7. Run the application

1. Create IBM Cloud services

Create the following services:

- **Watson Discovery**
- **Watson Assistant**

2. Configure Watson Discovery

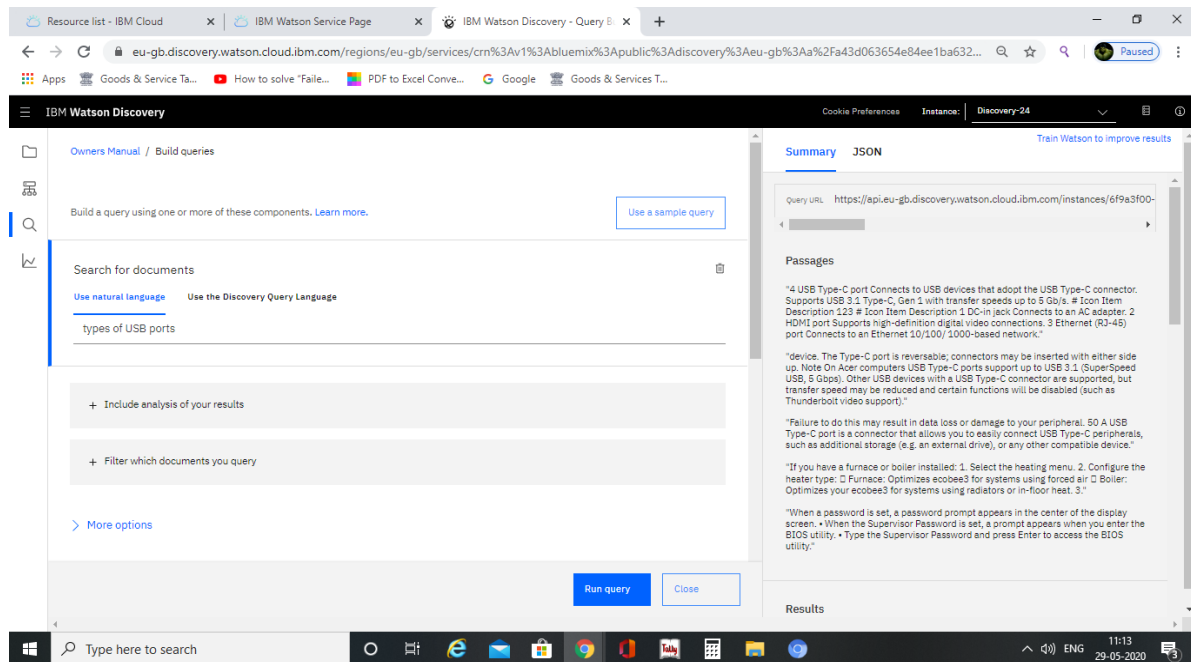
Import the document

As shown below, launch Watson Discovery tool and create a new data collection by selecting the `Upload your Own Data` option. Give the data collection a unique name. When prompted, select and upload the `User Manual_Acer_1.0_A_A` file located in the

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

Click the **Build your own query**

Click the **Build your own query**



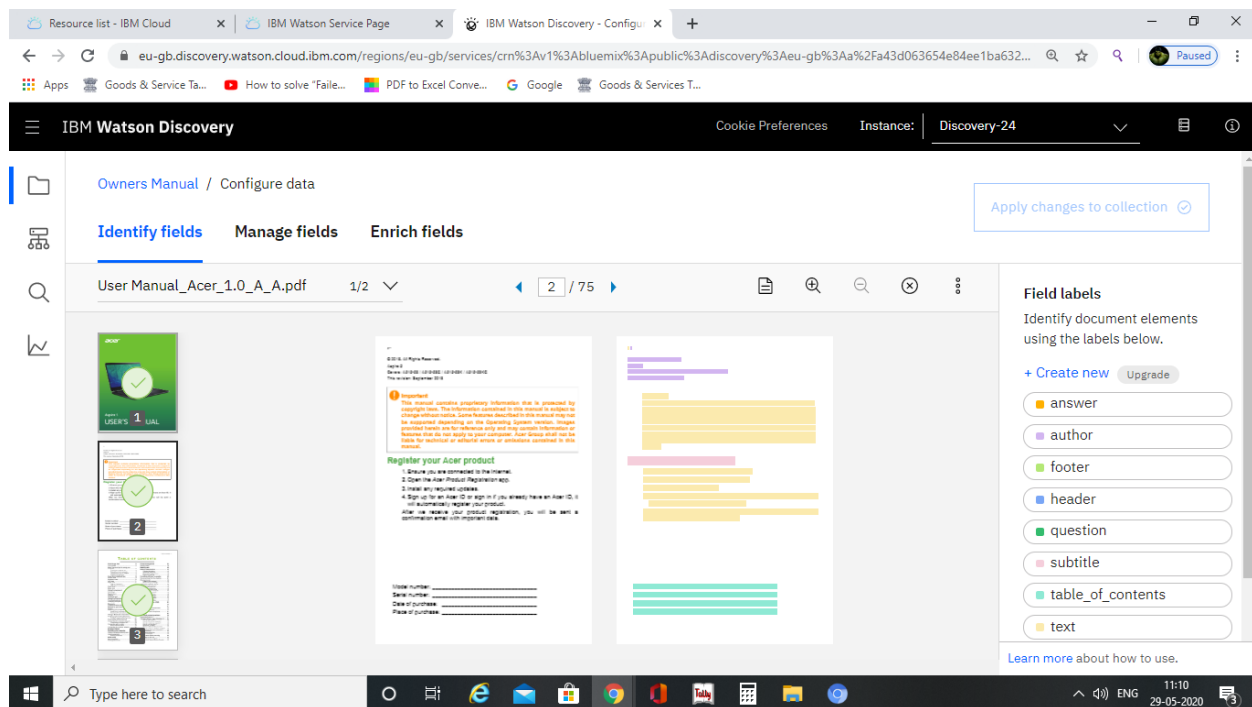
Enter queries related to the operation of the thermostat and view the results. As you will see, the results are not very useful, and in some cases, not even related to the question.

Annotate with SDU

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

From the Discovery collection panel, click the **Configure data** button (located in the top right corner) to start the SDU process.

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



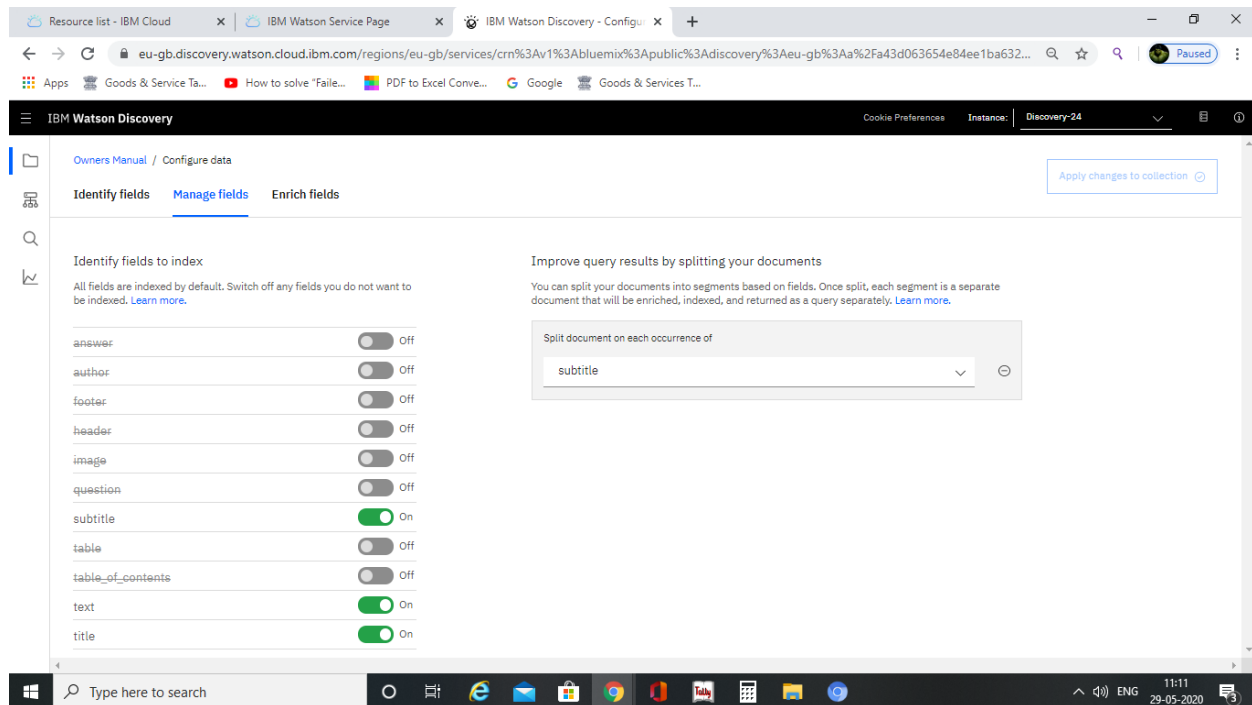
The goal is to annotate all of the pages in the document so Discovery can learn what text is important and what text can be ignored.

Here we need to annotate the page with suitable labels for the text.

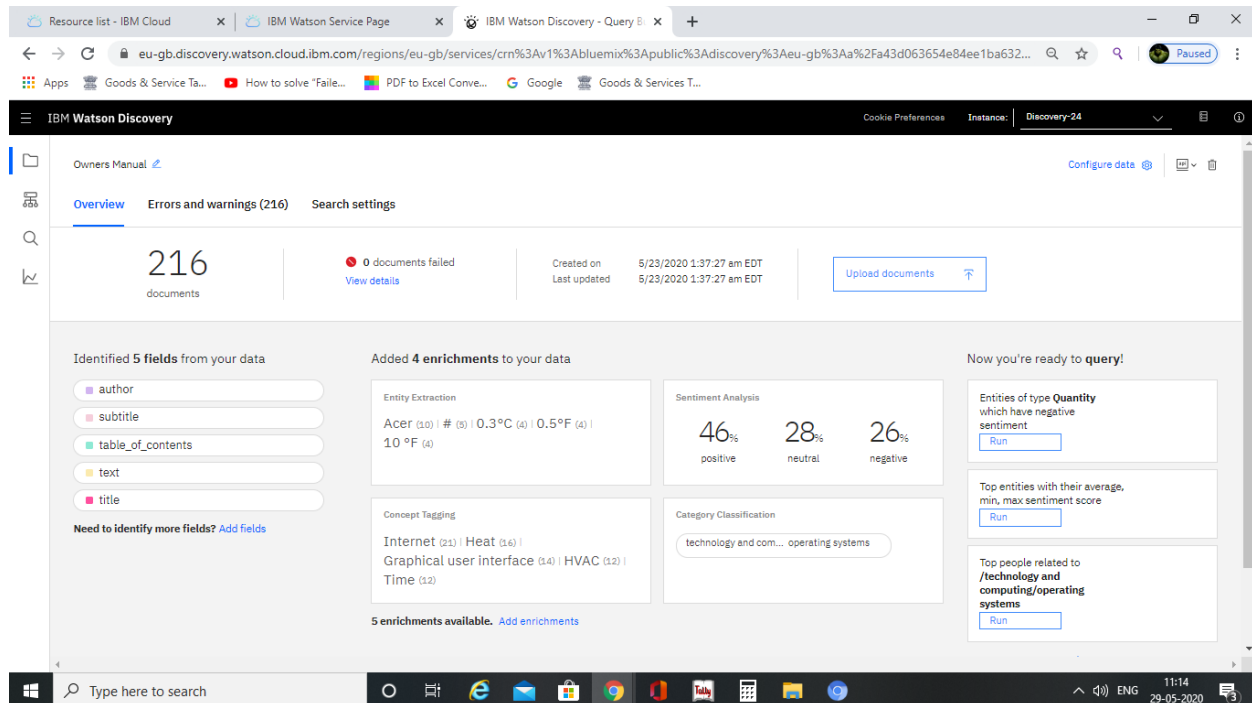
we need to select the text and assign the label and submit the page to Discovery

After Completion of this process click on Apply changes to collection button and reload the Document .Choose the same document used before.

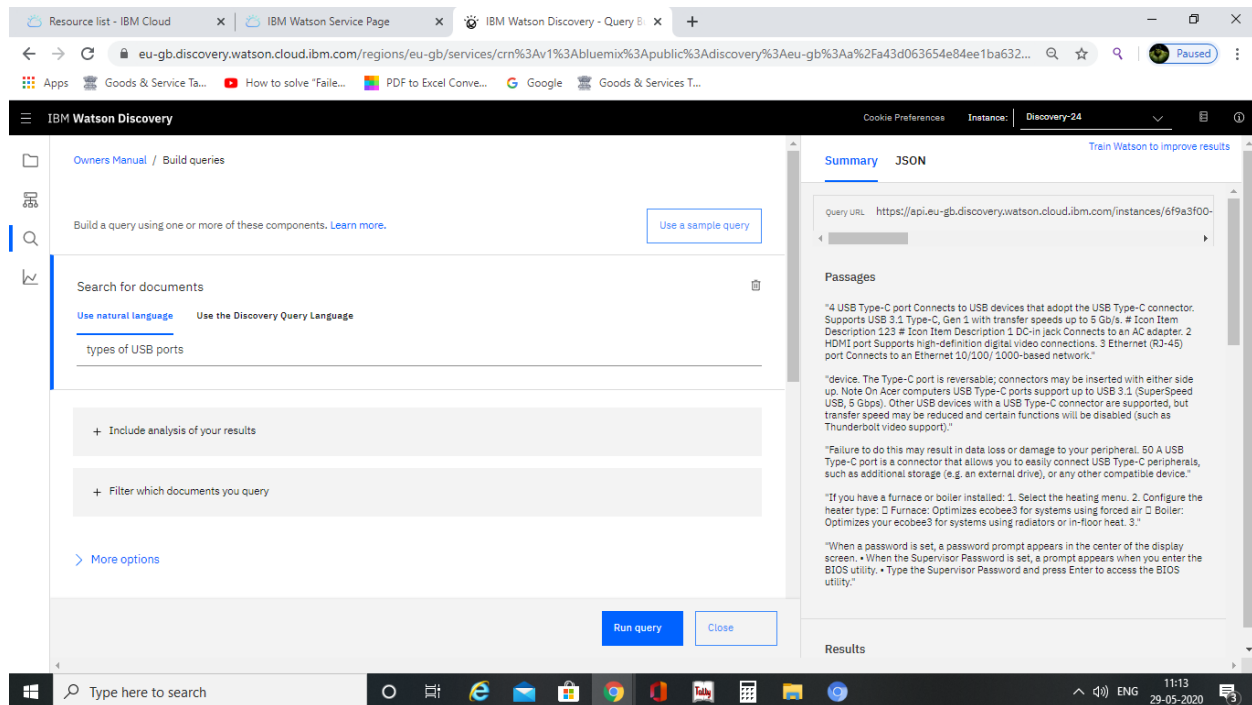
Now,Click on Manage Fields



Select the Required Fields and Select the Field through which we are Splitting the Document.



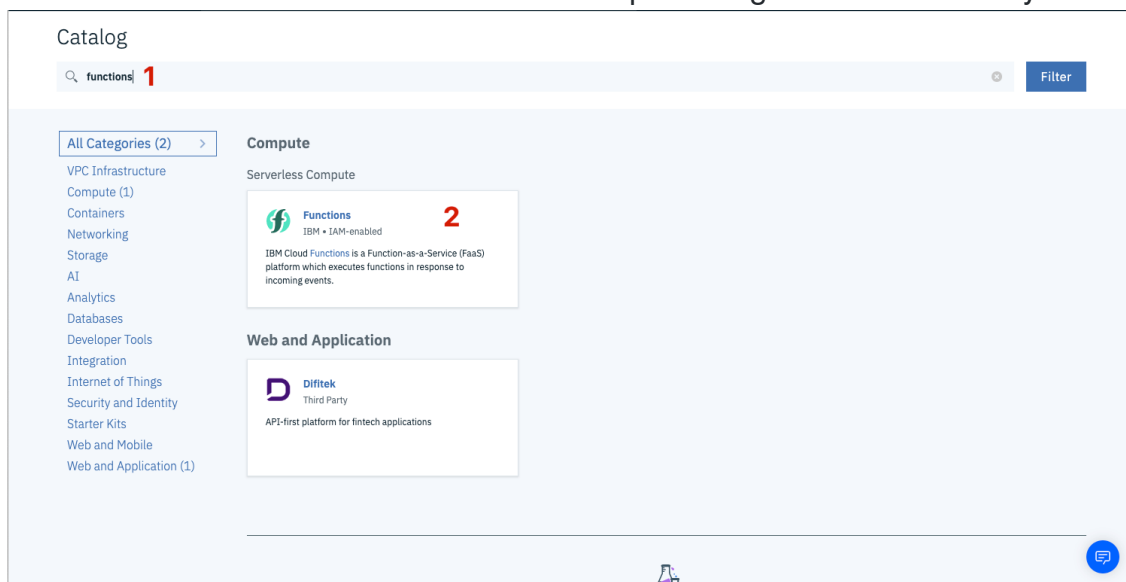
The Document is Split into Multiple Documents Based on the Field we have selected. The Return to the panel and see the Results Again.



Here we get Better Response than before.

3.Create IBM Cloud Functions action

Now lets create the Web action that will make queries against our Discovery collection.



From the create panel ,select the Create Action option

Now provide Action Name ,Default Package,slect Node.js 10 and click the Create

Button.

Functions

Getting Started

Actions

Triggers

APIs

Monitor

Logs

Namespace Settings

Create Action

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

[Learn more about Actions](#)

[Learn more about Packages](#)

Action Name

disco-action-2 1

Enclosing Package ①

(Default Package) 2

Create Package

Runtime ①

Node.js 10 3

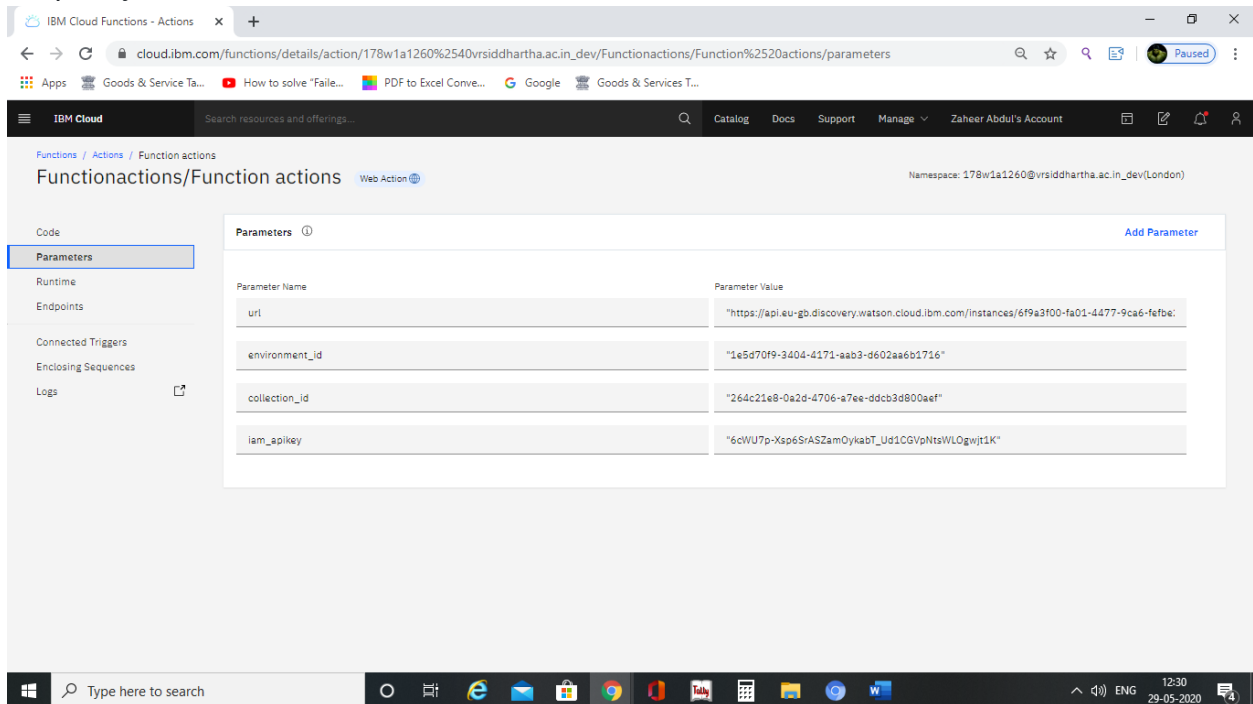
Looking for Java, .NET or Docker? [Docker](#) Actions can be created with the [CLI](#)

Cancel Previous Create 4

Here in Actions we have Code[1],Parameters[2],Runtime[3],Endpoints[4].

In Code [1] we need to import the code from disco-action.js

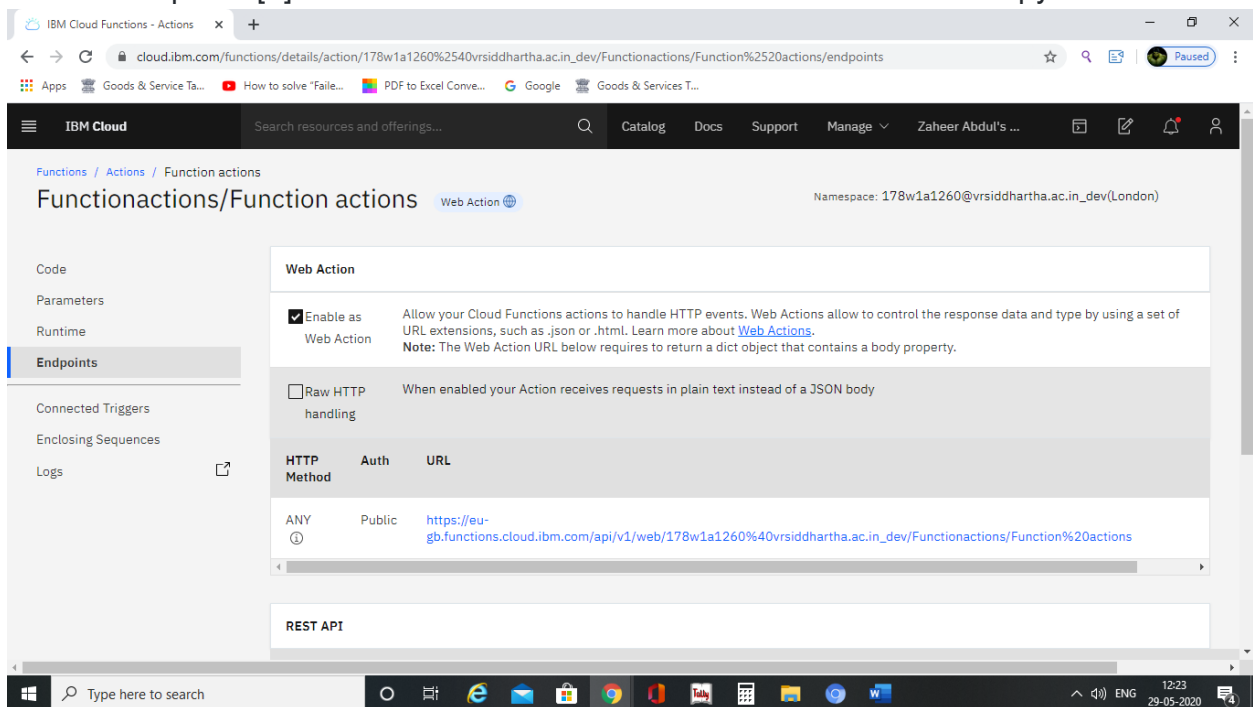
4.Api key



The screenshot shows the IBM Cloud Functions console. The left sidebar has a menu with 'Parameters' selected. The main area is titled 'Functionactions/Function actions' and shows a list of parameters. The parameters are:

Parameter Name	Parameter Value
url	"https://api.eu-gb.discovery.watson.cloud.ibm.com/instances/6f9a3100-fa01-4477-9ca6-fefbe"
environment_id	"1e5d70f9-3404-4171-aab3-d602aa6b1716"
collection_id	"264c21e8-0a2d-4706-a7ee-ddcb3d800aef"
iam_apikey	"6cWU7p-Xsp6SrASZamOykabT_Ud1CGVpNtsWLOgwt1K"

Next in Endpoints[4] we have to select the Enable as WebAction and copy the Url



The screenshot shows the IBM Cloud Functions console. The left sidebar has a menu with 'Endpoints' selected. The main area is titled 'Functionactions/Function actions' and shows the 'Web Action' configuration. The 'Enable as Web Action' checkbox is checked. The 'Raw HTTP handling' checkbox is unchecked. The 'HTTP Method' is set to 'ANY', 'Auth' is 'Public', and the 'URL' is 'https://eu-gb.functions.cloud.ibm.com/api/v1/web/178w1a1260%40vrsiddhartha.ac.in_dev/Functionactions/Function%20actions'.

4.Configure watson Assistant

from the Dashboard select the Watson Assistant and Launch the Watson Assistant and

then Create an Assistant and a Skill in that Assistant.

After creating the skill we need to add Intents ,Entities and Dialog for that Skill

The screenshot shows the IBM Watson Assistant interface. The left sidebar contains a menu with options: Intents, Entities, Dialog, Options, Analytics, Versions, and Content Catalog. The 'Intents' option is selected. The main area displays a table of 10 intents. The table has columns for a checkbox, the intent name, a description, the modification time, and the number of examples. A 'Create intent' button is in the top right corner. The bottom of the screen shows a Windows taskbar with various application icons and a search bar.

<input type="checkbox"/>	Intents (10) ↑	Description	Modified ↑↓	Examples ↑↓
<input type="checkbox"/>	#cancel	cancel the current request	6 days ago	2
<input type="checkbox"/>	#Customer_care_appointments	Schedule or manage an in- store appointment	2 days ago	4
<input type="checkbox"/>	#Customer_care_store_hours	find business hours	2 days ago	4
<input type="checkbox"/>	#Customer_care_store_Location	locate a physical store location or an address	2 days ago	7
<input type="checkbox"/>	#General_contact_to_Agent	Request a human agent	2 days ago	4
<input type="checkbox"/>	#Greeting_sir	greetings	6 days ago	6

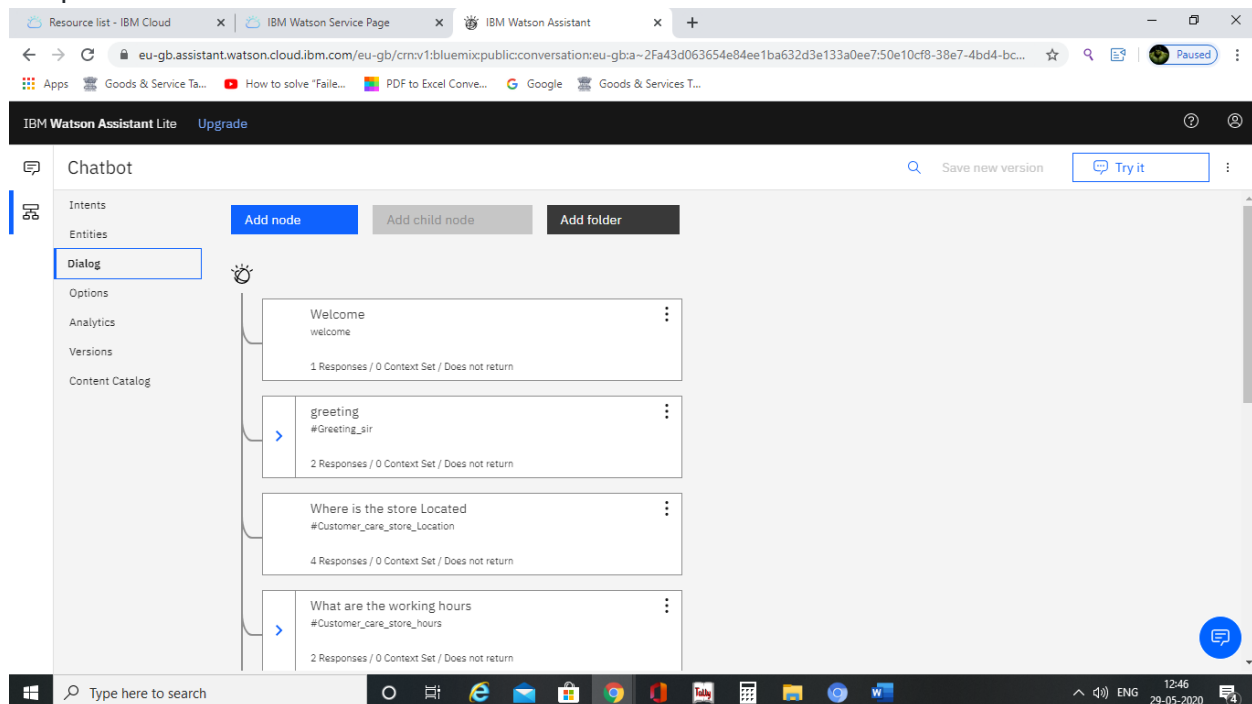
These are the Intents for the skill We can add any number of Intents based on our Requirement

After this we need to add the Entities

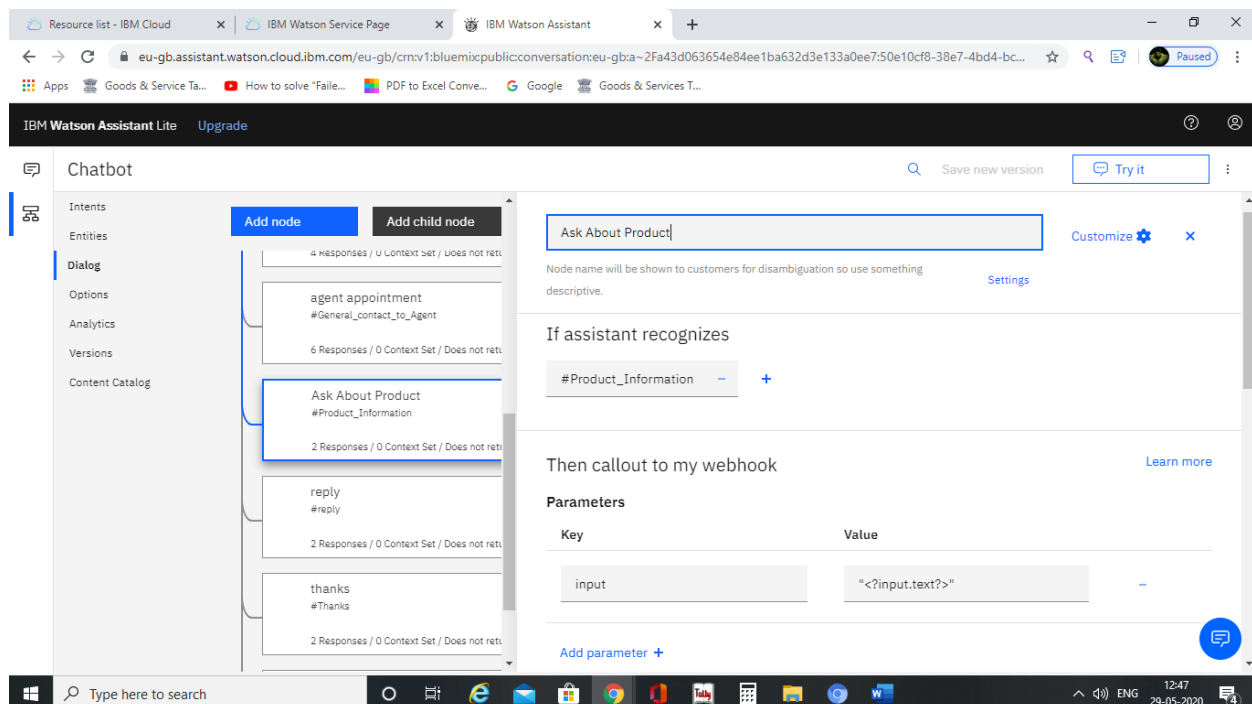
The screenshot shows the IBM Watson Assistant interface with the 'Entities' option selected in the left sidebar. The main area displays a table of 13 entities. The table has columns for a checkbox, the entity name, the values, and the modification time. A 'Create entity' button is in the top right corner. The bottom of the screen shows a Windows taskbar with various application icons and a search bar.

<input type="checkbox"/>	Entity (13) ↑	Values	Modified ↑↓
<input type="checkbox"/>	@agent_appointment	which agent to be contacted, how many agents are available, can u book an appoi...	2 days ago
<input type="checkbox"/>	@Greetings	Good afternoon, Good evening, Good morning	6 days ago
<input type="checkbox"/>	@help	can u help me, i need some help	3 days ago
<input type="checkbox"/>	@holiday	christmas eve, independence day, ramzan, diwali, holiday	6 days ago
<input type="checkbox"/>	@landmark	pvp square, Grand central, landmark, D Address Mall	6 days ago
<input type="checkbox"/>	@reply	Yes, No	2 days ago

These are some of the Entities used .We can use any no of Entities based on our Requirement.



This is the Dialog Section where we give Response for the Relevant Queries asked by the user.



This the Dialogue node where we use the Discovery queries from the Document we

used for Smart Document Understanding SDU.

The screenshot shows the IBM Watson Assistant interface. On the left, a sidebar lists various components: Intents, Entities, Dialog, Options, Analytics, Versions, and Content Catalog. The 'Dialog' section is selected, showing a list of nodes. The 'Ask About Product' node is highlighted. The main area displays the configuration for this node. It includes a 'Node name' field with the value 'Ask About Product' and a 'Node description' field. Below this, the 'Assistant responses' section is visible, showing two response rules. The first rule is triggered by '\$webhook_result_1' and responds with '<?\$webhook_result_1.passag'. The second rule is triggered by 'anything_else' and responds with 'Try Again Later'. A 'Customize' button is located in the top right corner of the configuration area.

We give Direct Response from the Document using this webhook which can be Enabled in the Customize Option.

The screenshot shows the 'Customize' dialog for the 'Ask About Product' node. The dialog has a title bar that says 'Customize "Ask About Product"'. Inside, there is a section for 'Webhooks' with a toggle switch set to 'On'. Below this, there is a text area for 'Webhook URL' and a 'Learn more' link. At the bottom of the dialog, there are 'Cancel' and 'Apply' buttons. A green notification banner at the bottom of the dialog states 'Webhook URL Your webhook URL is configured.' with an 'Options' button and a close button.

After Enabling the Webhooks for the node click on Apply and then go to Options and add the link from the Discovery with .json Extension in the Webhook.

Resource list - IBM Cloud x IBM Watson Service Page x IBM Watson Assistant x +

eu-gb.assistant.watson.cloud.ibm.com/eu-gb/cm/v1:bluemixpublicconversation:eu-gb-a-2Fa43d063654e84ee1ba632d3e133a0ee7:50e10cf8-38e7-4bd4-bc...

Apps Goods & Service Ta... How to solve 'Faile... PDF to Excel Conve... Google Goods & Services T...

IBM Watson Assistant Lite Upgrade

Chatbot Save new version Try it

Intents Entities Dialog Options Webhooks Disambiguation Autocorrection Irrelevance Detection System Entities Analytics Versions Content Catalog

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

1260%40vrsiddhartha.ac.in_dev/Functionactions/Function%20actions.json

Headers

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

Header name	Header value
Add header	Add authorization

Type here to search

After Adding the Webhook Link we Can directly Access the Discovery queries in the Watson Assistant.

5.Integrate the Services to NODE RED

Go to the IBM Dashboard and Select the Node Red

As i have already created the flow click on the link if not created create the node red by clicking in Create.

Resource list - IBM Cloud x Service Details - IBM Cloud x +

cloud.ibm.com/services/cloudantnosql/db/cm%3Av1%3Abluemix%3Apublic%3Acloudantnosql/db%3Ache01%3Aa%2Fa43d063654e84ee1ba632d3e133a...

Apps Goods & Service Ta... How to solve 'Faile... PDF to Excel Conve... Google Goods & Services T...

IBM Cloud Search resources and offerings...

node-red-sample-cloudant-1589816059050 Active Add tags

Details Actions...

Manage Service credentials Plan Connections

Search connections...

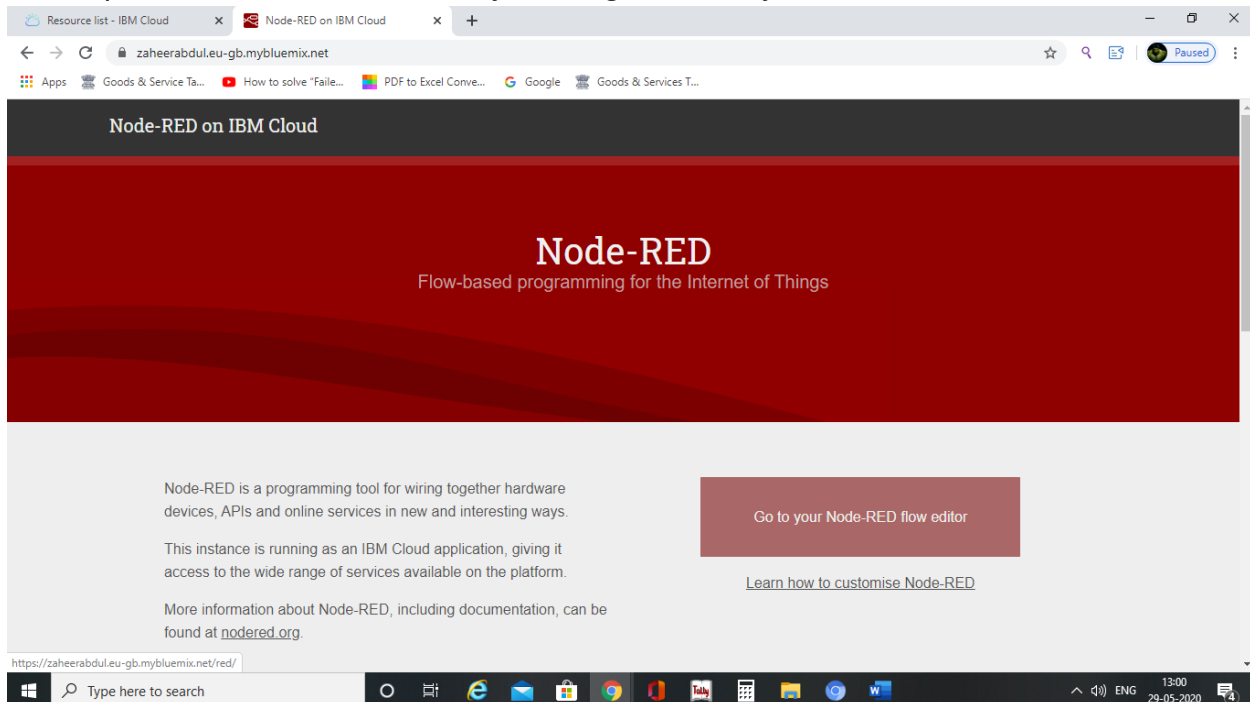
Create connection

Name	Group	Cloud Foundry alias	Route	Status
Node RED Sample	178w1a1260@vrsiddhartha.ac.in / dev	node-red-sample-cloudant-1589816059050-27	Zaheerabdul.eu-gb.mybluemix.net	Started

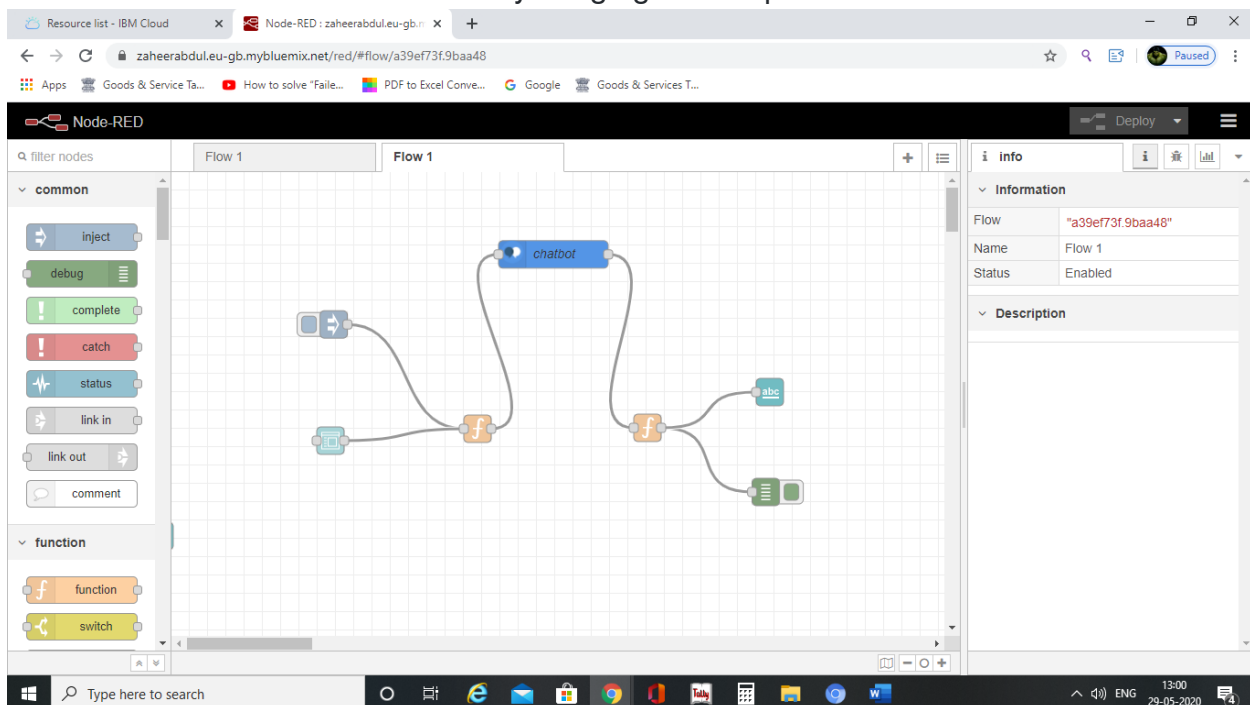
Waiting for dashboard.cloudantnosql.db.cloud.ibm.com...

Type here to search

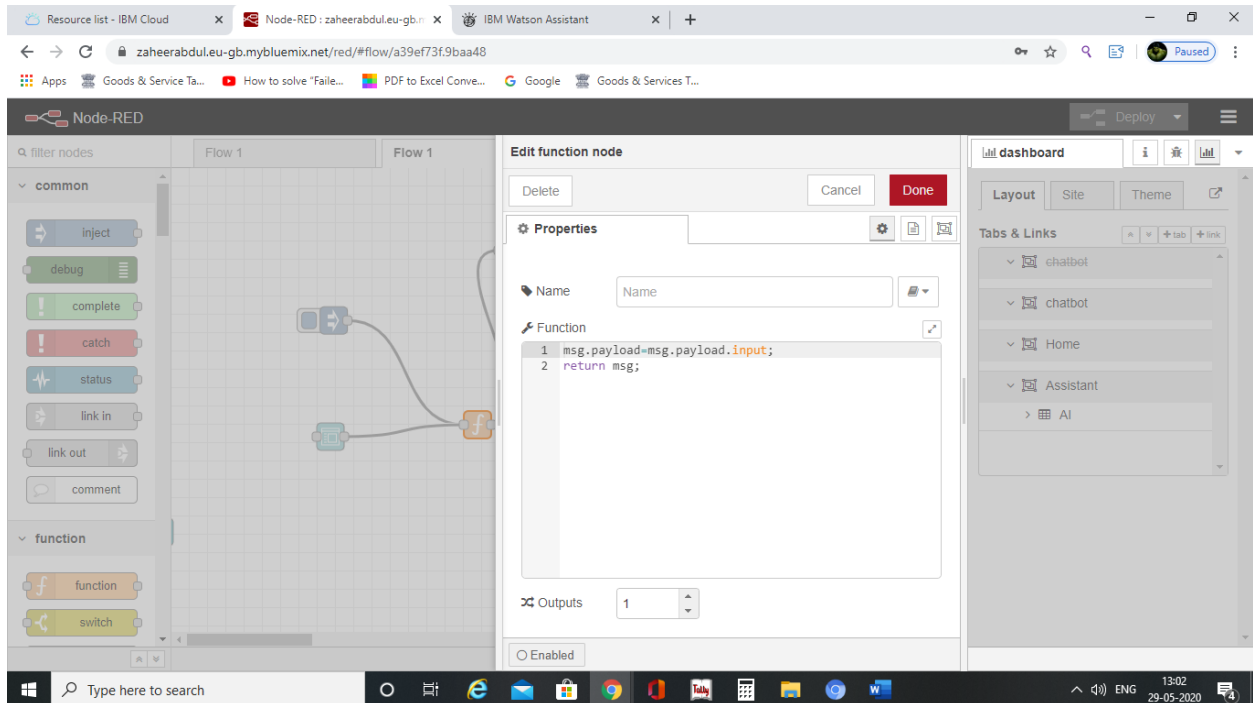
Now Open the NODE RED Flow by clicking on Go to your Node-Red Flow editor



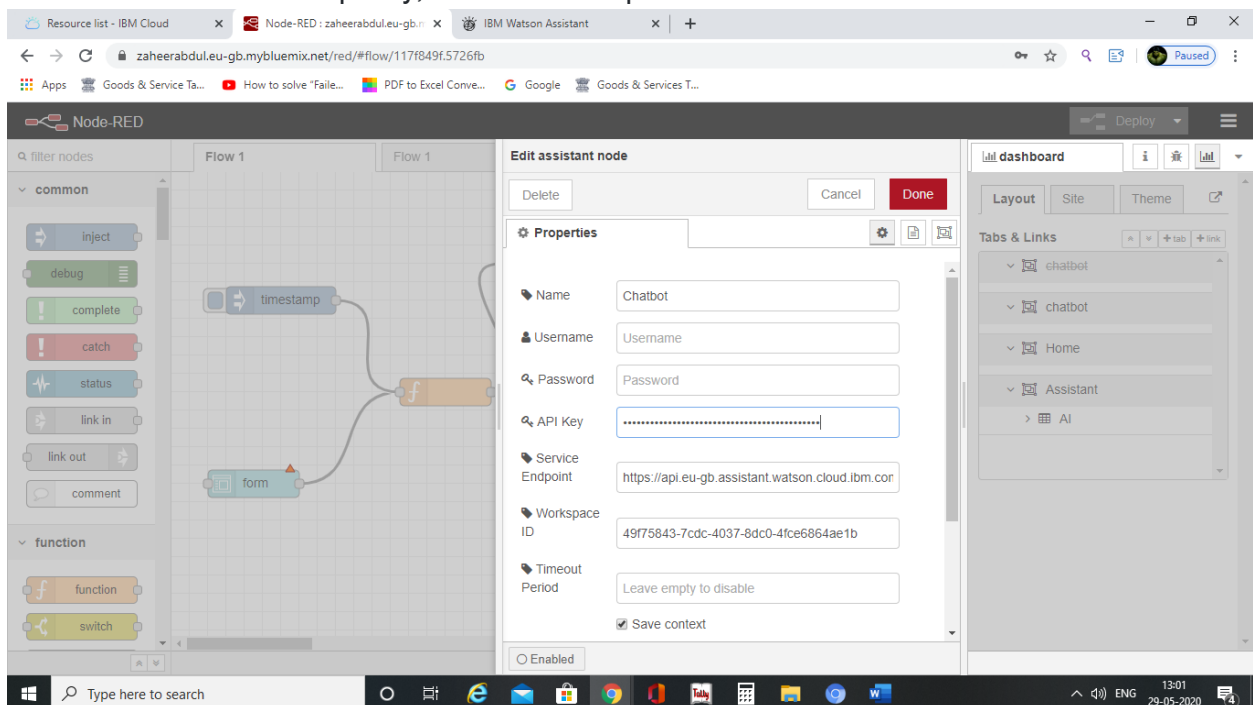
After this create the flow as Below by Dragging the Required Nodes



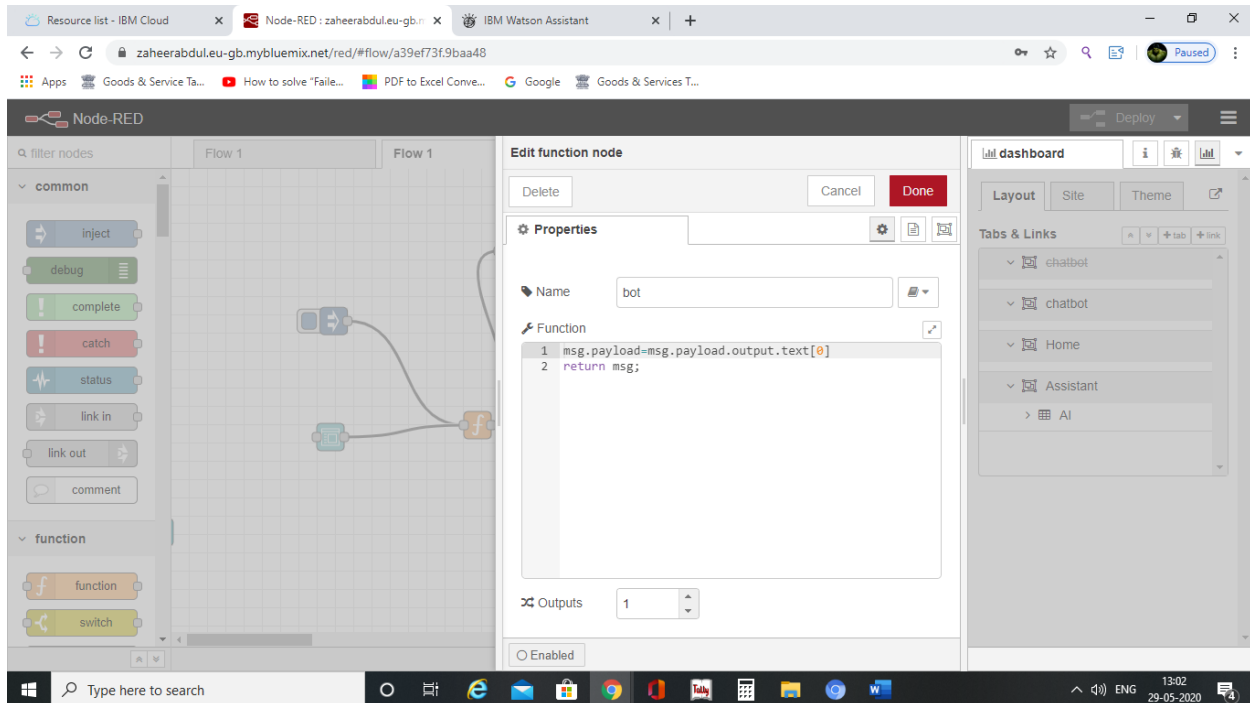
Add Inject node and Form Node Connected to an Input Function Node Containing the input function.



After this Input Function Node Add an Assistant node and give the Credentials of the Watson Assistant like ApiKey,Url and Workspace ID from the Watson Assistant Skill.

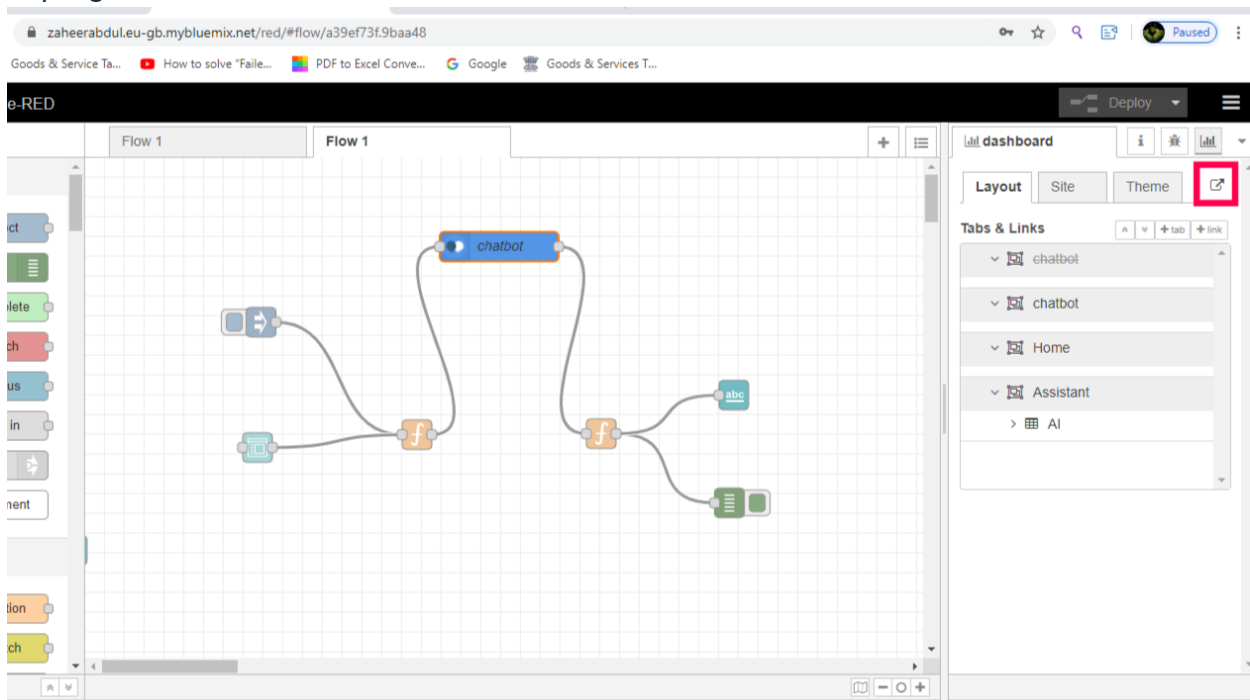


After this Add another function node for Output with the following code



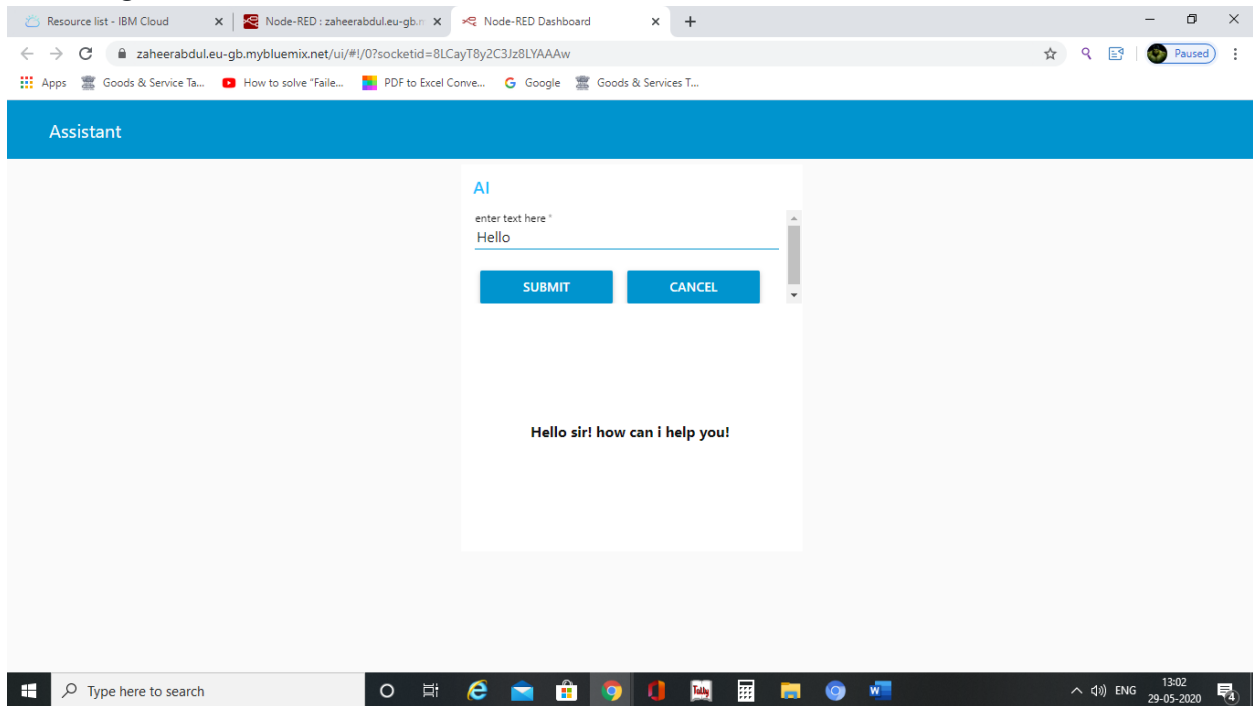
After Adding the Output Function Node add the Text input for displaying output and the Debug node for Debugging as Shown in the Node Red flow Earlier.

Now Deploy the Node Red Flow and Run the Flow by Clicking on the button on the Top-right Corner in Dashboard.



After Clicking on the Sown Red Square box the Node Red flow gets Executed in the

Web Page



This is the Output of the Node Red Flow for the Queries of Watson Assistant as well as Watson Discovery .