**INTELLIGENT CUSTOMER HELP DESK USING SMART DOCUMENT UNDERSTANDING.**

**ALMASFIZA SHAIKH**

**INDEX**

**1.INTRODUCTION**

**1.1 OVERVIEW**

The title of the project is “Intelligent Customer Help Desk with Smart Document Understanding.

Category: Artificial Intelligence

Skills Required: Python, IBM Cloud and IBM Watson

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.

**1.2 PURPOSE**

The purpose of this project from the developer’s point of view is to learn about the IBM Cloud functions and develop the required skills to create the project. Furthermore, creating a help desk customer chat bot to address the issues of the people is the purpose with the project point of view.

The topic selected in this project is MacBook troubleshooting. If any MacBook user needs to troubleshoot issues they are having with their device, they can use the chatbot created as the end product of this project.

**2.LITERATURE SURVEY**

**2.1 EXISTING PROBLEM**

Chatbots are integrated in multiple sites these days. It not only helps reduce the manual labour required from the company but also provides quick access to solutions to the customer. The customer need not wait to get in touch with a human technician in case of an emergency. Most of the frequently asked issues are updated into the chatbot data and it saves a lot of time for both the customer and the human tech help.  
  
However, a chatbot may be able to answer only some specific questions which exist in its data dialog. If a developer has feeded three questions into the dialog of the chatbot-   
1.How to switch on the MacBook?  
2.How to use multiple usb ports?  
3.How to optimise the battery?  
  
The chatbot can only provide the answers for these questions, because it has been trained on them. If another question is asked like - How do I clean my MacBook storage, the chatbot may fail to answer the query and redirects the customer to human tech support.

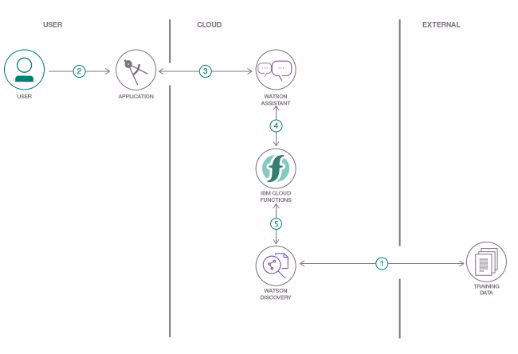
**2.2 PROPOSED SOLUTION**

In the proposed solution, instead of being redirected to human tech support, the chatbot should use Artificial Intelligence to find the relevant answer to an unknown query from the User Manual or any data set that has been provided to it for reference.

This can be accomplished by using IBM Watson Discovery. The document to refer to is uploaded to the system, the text, titles, subtitles etc are marked and trained, the relevancy of the answers to some queries are rated and a webhook is used to connect the Discovery model to the Watson AI.

**3.THEORETICAL ANALYSIS**

**3.1 BLOCK DIAGRAM**



* 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

**3.2 HARDWARE AND SOFTWARE DESIGNING.**

The IBM Cloud services that we used in the project are- IBM Watson, IBM DIscovery, Node RED Starter Kit Application and Cloud Function.

With IBM Cloud™ Continuous Delivery, we can use open toolchains to build, deploy, and manage our applications. Toolchains are integrated sets of tools that make development and operations tasks repeatable and manageable. Toolchains can include IBM Cloud services, open source tools, and third-party tools.

The Continuous Delivery service includes these features:

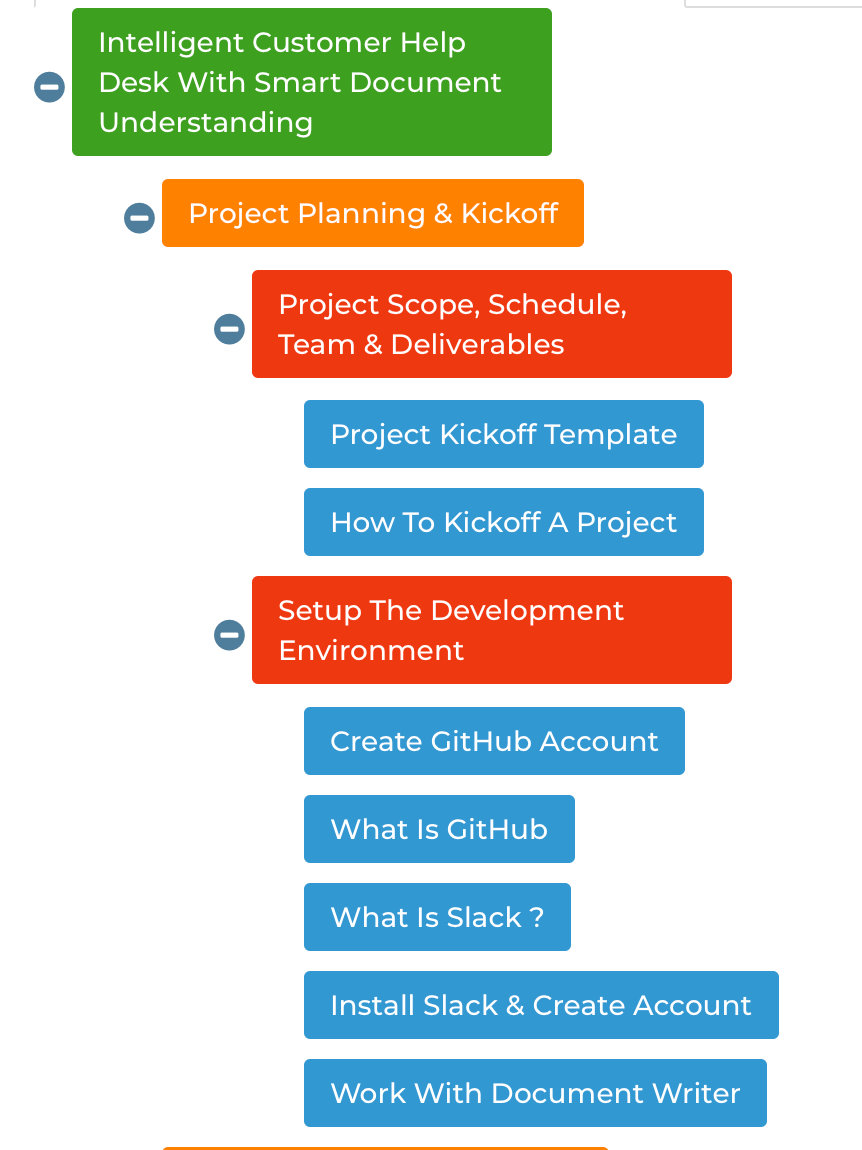
* Toolchain templates: To get started quickly, you can create a toolchain from a template that includes a specific configuration of tools to develop, build, deploy, and manage an application. For example, you might create a toolchain from the Microservices toolchain template or the Simple container toolchain template.
* Pipeline: Automate builds, unit tests, deployments, and more. The stages in the pipeline can automatically build when you push changes to a linked repository and then deploy to one or more environments on IBM Cloud. You can also incorporate builds and deployments into your toolchains to connect builds to other tools. You can create a composite pipeline for a toolchain and use it to view and manage deployments across the toolchain.
* Web IDE: From the web-based integrated development environment (IDE), you can seamlessly move from editing your code to deploying it into production.

We can also add tools for team collaboration, planning, source-code management, and testing from companies such as GitHub, PagerDuty, Slack, and Sauce Labs.

**4. EXPERIMENTAL INVESTIGATIONS**

**Intelligent Customer Help Desk with Smart Document Understanding**

Milestone 1: Project Planning and Kick Off



Category: Artificial Intelligence

Skills Required: Python, IBM Cloud, IBM Watson

Project Description:A typical chat bot can also answer the questions whose answers have been feeded into into its intents and stories by the developer. Questions like - "How do I bake a cake in the oven..." provides with the oven specifications to be set before baking the cake.

However, a question out of the scope of the chatbot's pre-defined questions like "Can I bake a cake without using xyz specification mode?" leads to a prompt to speak to a real IT solution person behind the scenes like "Would you like to speak to the Customer Support?"

*In this project, in cases of out of scope questions the chatbot should pass the questions to the IBM Watson Discovery Service, which has been preloaded with the device's manual for the owner*.

To make the project efficient a Smart Document understanding feature of the IBM Watson is used to mark the text in the document which is more useful than the rest of the text manual. This will improve the query processing of the application.

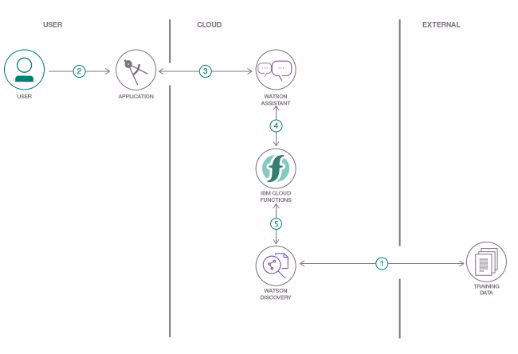
Following is the Scope of Work as described in the project.

1.Create a customer care dialog skill in Watson Assistant

2.Use Smart Document Understanding to build an enhanced Watson Discovery collection

3.Create an IBM Cloud Functions web action that allows Watson Assistant to post queries to Watson Discovery

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

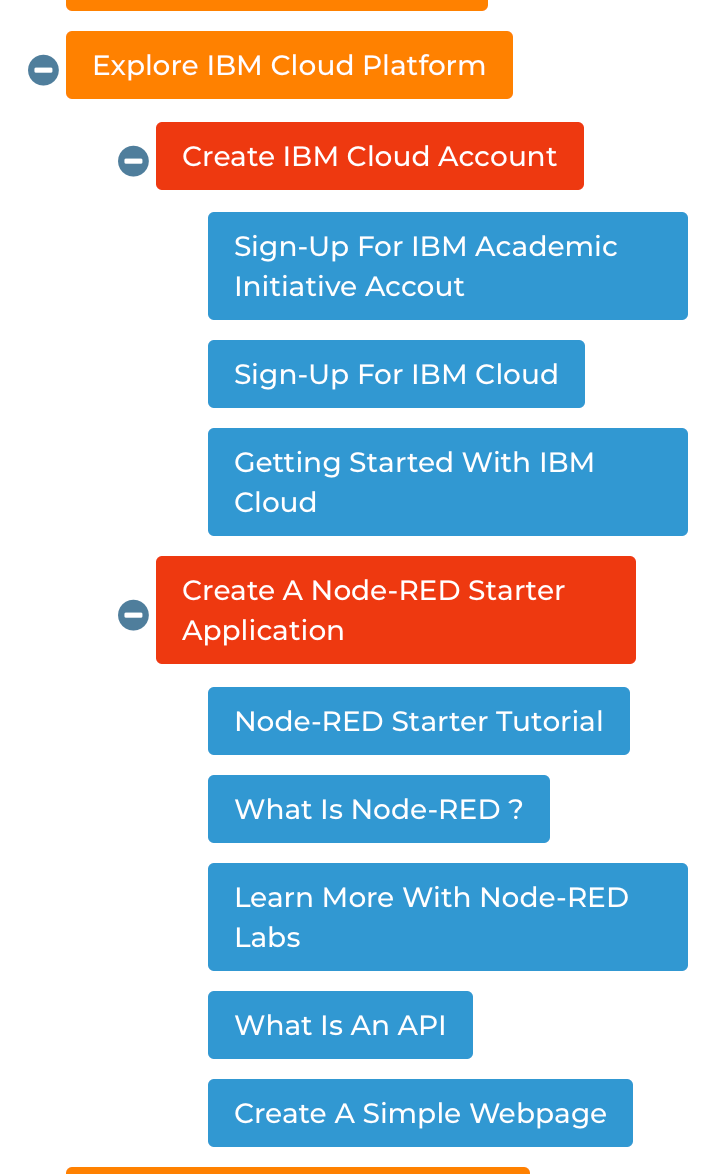


Note: Github Account already existed and the slack channel invitation was accepted

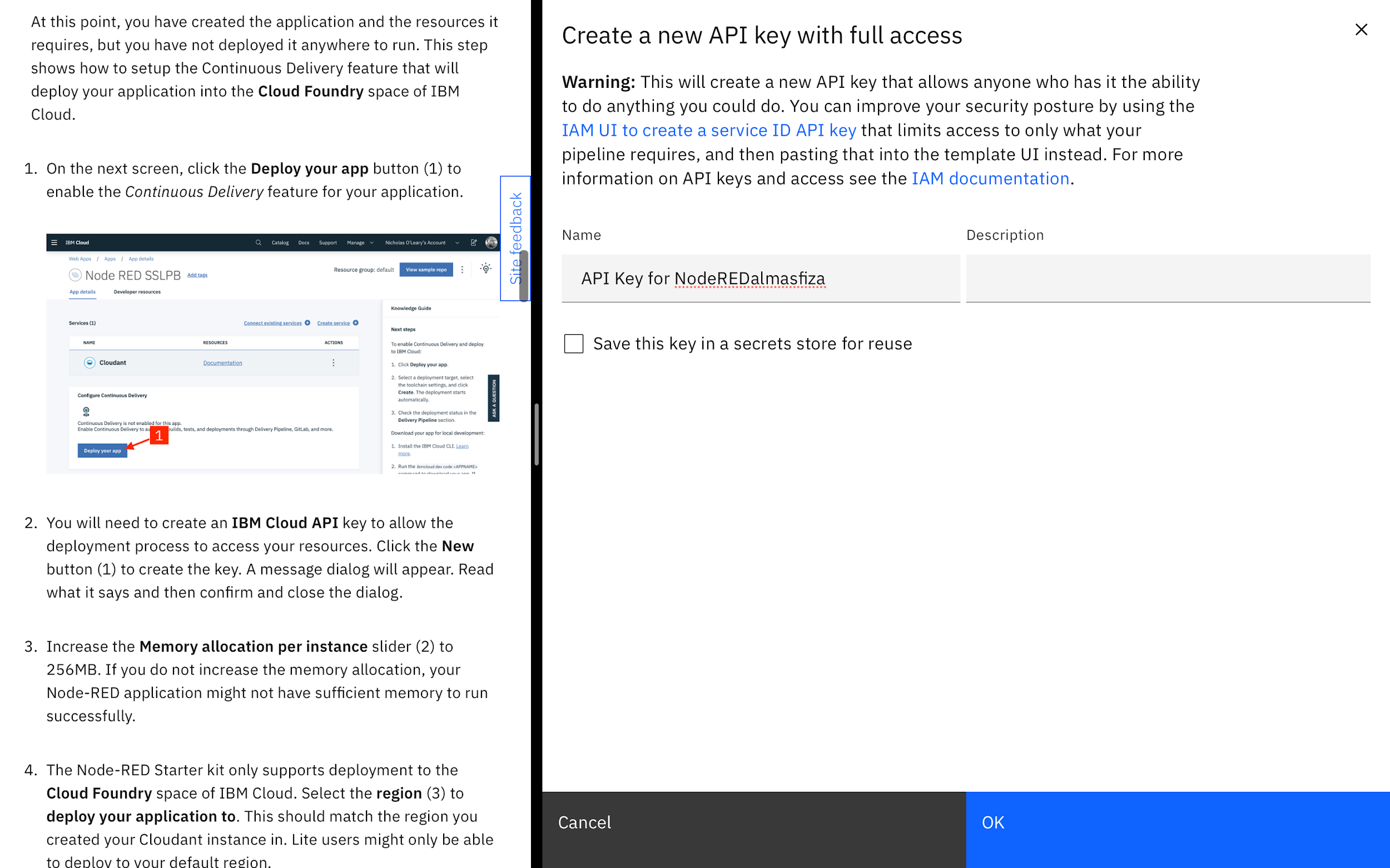
<https://github.com/almasfiza>

<https://app.slack.com/client/T013TF6BFU4/C013ZU8772M>

Milestone 2:Explore IBM Cloud Platform

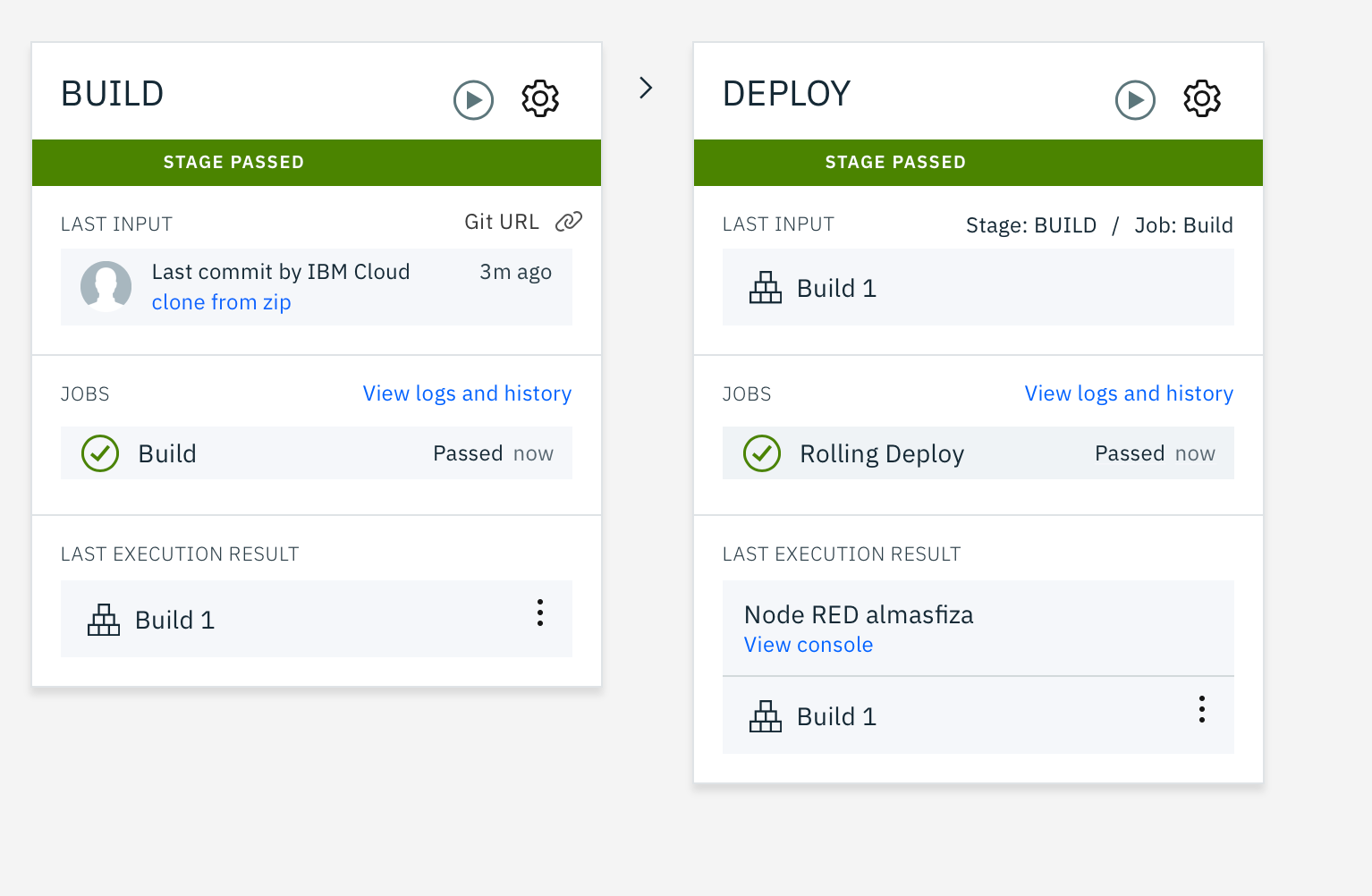


* Created an IBM Account: I used my personal email id instead of an institution email ID. I explored articles on what IBM Cloud was.
* Created a Node-RED Starter Application: Followed the resources under Node-RED Starter Tutorial to build my first Node-RED App. While following the tutorial, adding a red-node-dashboard using package.json, my deploy failed. I saw similar queries on the slack channel. One of the mentors had replied saying its okay to proceed further as Node RED is requered at the final stage and this was just a demo.



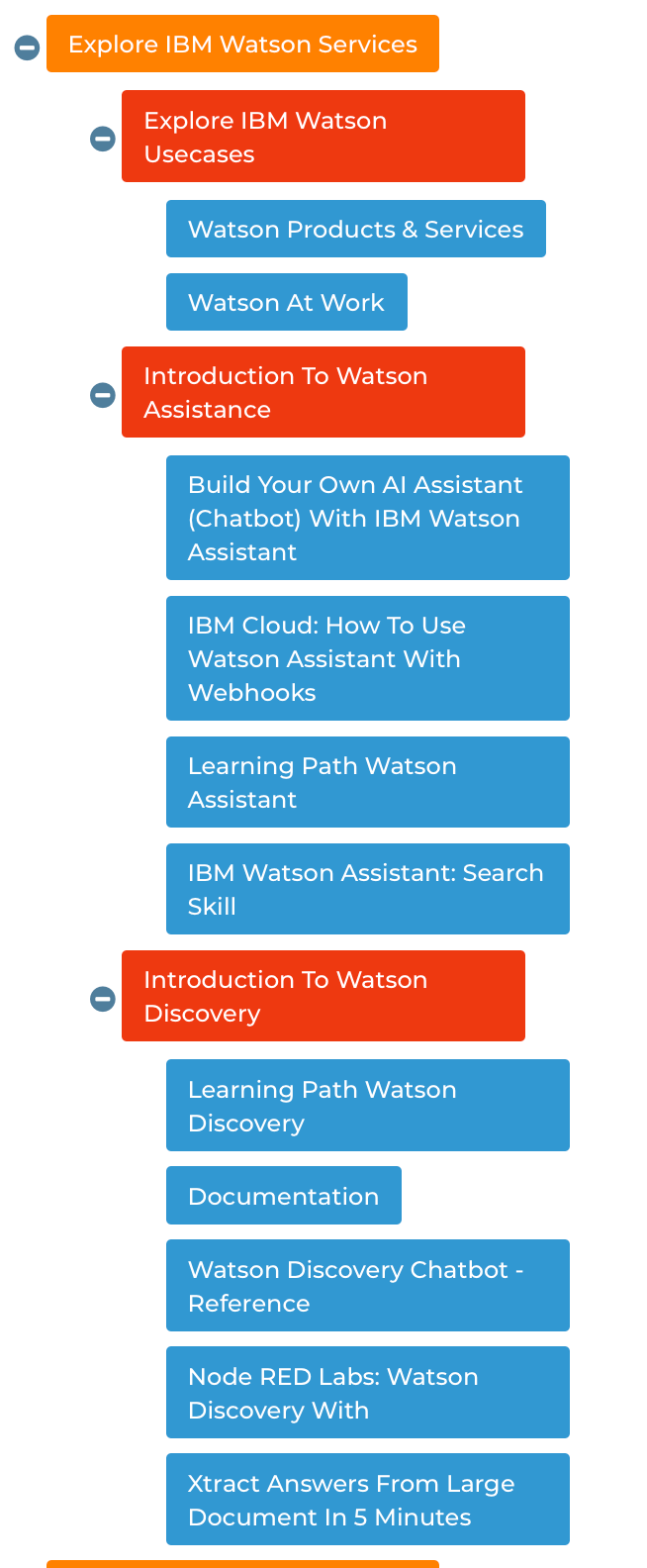
Getting an API Key for Node RED almasfiza (my application).

Edit: I re-created the Node RED application and this time named it Almasfiza



First deploy successful.

Milestone 3:Explore IBM Watson Sevices.



1.Watched through the links provided in the resource material.

2.Explored about the Watson Assisstant. Familiarised myself with the intents, entities and dialogs of the chat bot model.

3. Watched demonstration on Watson Discovery.

This milestone was all about discovering all the helpful youtube resource videos and buikding upon skills**.**

**Milestone 4:Explore IBM Cloud Functions**

The videos given in the references were watched. Videos on restful APIs and web functions etcetera.

Milestone 5:Customer Help Desk with Smart Documentation Understanding.

The real work begins.

----> Watson AI was trained with intents, responses and dialogs.

----> Watson Discovery was provided with the document of MacBook manual in pdf form. The one document was divided into 92 documents

based on the titles and sub headings. The Watson was trained by marking relevant and irrelevant answers to some

questions which were predefined in the intents.

----> A webhook from Watson Discovery was used to connect to the Watson AI.

----> A cloud function was used to create a .json file to connect the Watson AI and Discovery.

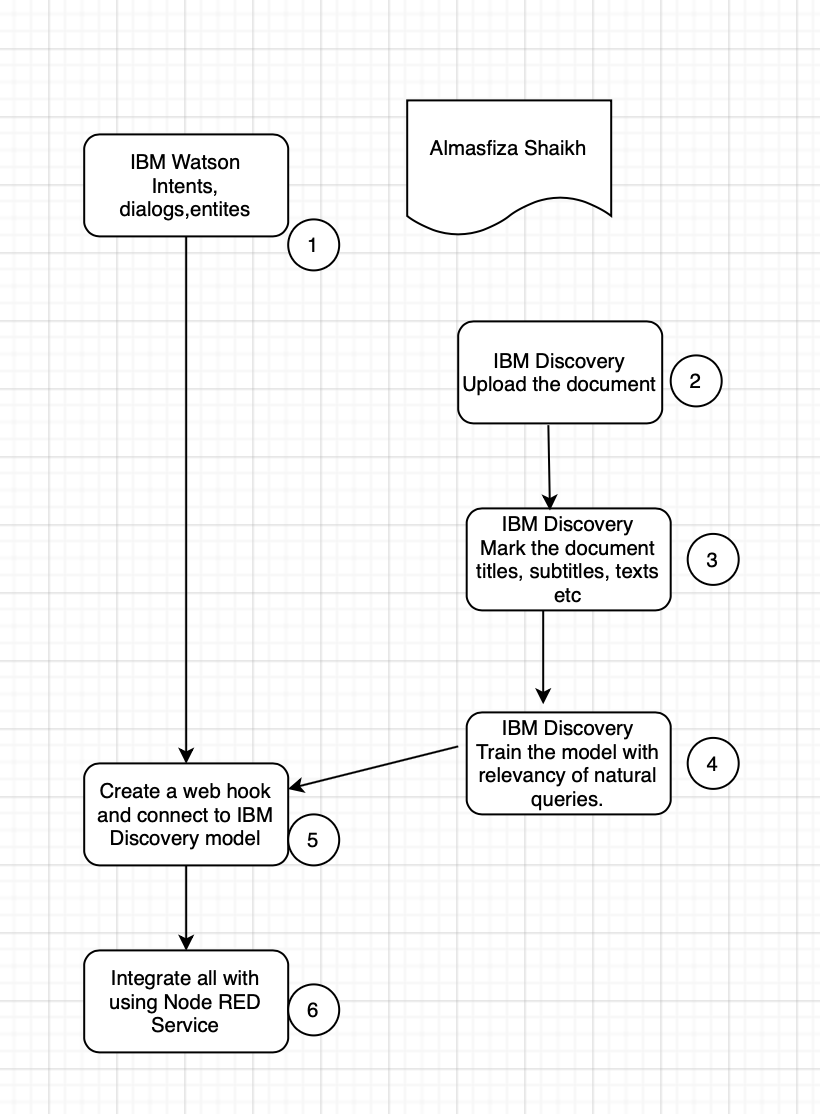
----> Node RED flow was imported. The Watson chatbot was used with the API credentials and a basic home page was made to test

the chatbot.

----> The report was completed on the internship and a video was uploaded demonstrating the working of the model.

----> All the code was uploaded on github.

**5. FLOWCHART**



6.RESULT

I was able to create a Node Red product in the end.

Find the link to the project below, as well as the git repository.

YOUTUBE VID:<https://www.youtube.com/embed/v8eJOrD01Z8>

GIT REPO: <https://github.com/almasfiza/llSPS-INT-1238-Intelligent-Customer-Help-Desk-with-Smart-Document-Understanding>

Some common questions that my bot can answer with high accuracy are:

* How to connect multiple USB ports?
* What is firewire
* Battery Optimisation
* Low brightness on screen
* Transfer files from one computer to another.

7. ADVANTAGES AND DISADVANTAGES

The main advantage of this project is that instead of the chatbot redirecting to a human tech support, it brings up the relevant passages from the MacBook Manual that can be helpful for the customer in need.

However, in cases of irrelevant information being suggested by the Watson model, the customer can feel helpless and in need of human tech help.

Watson AI is afterall a machine. Sometimes it might not work as expected, and therefore it's necessary to train a model to near perfection, if not perfection.

8. APPLICATIONS

Our bot can be used to address to the queries of MacBook users. It reads up the data from the MacBook Manual/user’s guide.

Other applications of Watson IBM have been incorporated in many chatbots. Recently the IBM channel on YouTube released a COVID19 Customer Care chatbot to aid the queries of the people.

9. CONCLUSION

Working on this project was my first step in the AI field. I have never been so fascinated by the power of what an AI model can do. I worked with Watson AI which is similar to other cloud services for chatbots like dialoglow which I have a gist about.

However, Watson Discovery changes the entire game and takes the chatbot to a whole different level.

I acknowledge that my Watson model might not be that well trained as a professional developer’s but I know what potential it contains.

10. FUTURE SCOPE

The future scope of the watson model that i created is probably training Watson discovery better to produce relevant search results.

Maybe my model can do better with more intents and training.

11. BIBLIOGRAPHY

<https://www.ibm.com/in-en/cloud/continuous-delivery>

[Watson Discovery - Overview | IBMwww.ibm.com › in-en › cloud › watson-discovery](https://www.ibm.com/in-en/cloud/watson-discovery)

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

<https://www.ibm.com/watson/products-services>

APPENDIX

A.SOURCE CODE

The source code can be found on the git repo:

<https://github.com/almasfiza/llSPS-INT-1238-Intelligent-Customer-Help-Desk-with-Smart-Document-Understanding>