VISUAL RECOGNITION HELPER FOR VISUALLY CHALLENGED

Mini Project CMPE272

Venkatesh Ramanujam Rangarajan (010737831)

Balamurugan Avudaiyappan (010818717)

Shreyams Jain (010736427)

Instructor: Prof. Rakesh Ranjan

APRIL 10, 2016

SJSU

San Jose

Contents

Project Details

Project Name: Visual recognition helper for visually challenged

Git Hub URL: https://github.com/bavudaia/Visual-Recognition

Website URL: http://visualtoaudio.mybluemix.net/reco

Boiler Plate Used: Node Red Starter Community

Service Used: Text to Speech, Visual Recognition

Introduction

This document briefs about the image to audio description web app developed by team #17 as part of mini project assignment.

Objective

This app aims to help the visually challenged person in identifying the object that is present in the image.

Below fig. shows the difference on how a normal person sees an image and how a visually challenged person sees the image.

Fig 1. How normal person sees an image

Fig 2. How color blinded person sees the image

As illustrated in Fig 1 and Fig 2, visually challenged person misses red outline and sees the entire image as black.

Development

Boiler plate

We have used Node Red Started Community BETA in our project to get the things started.

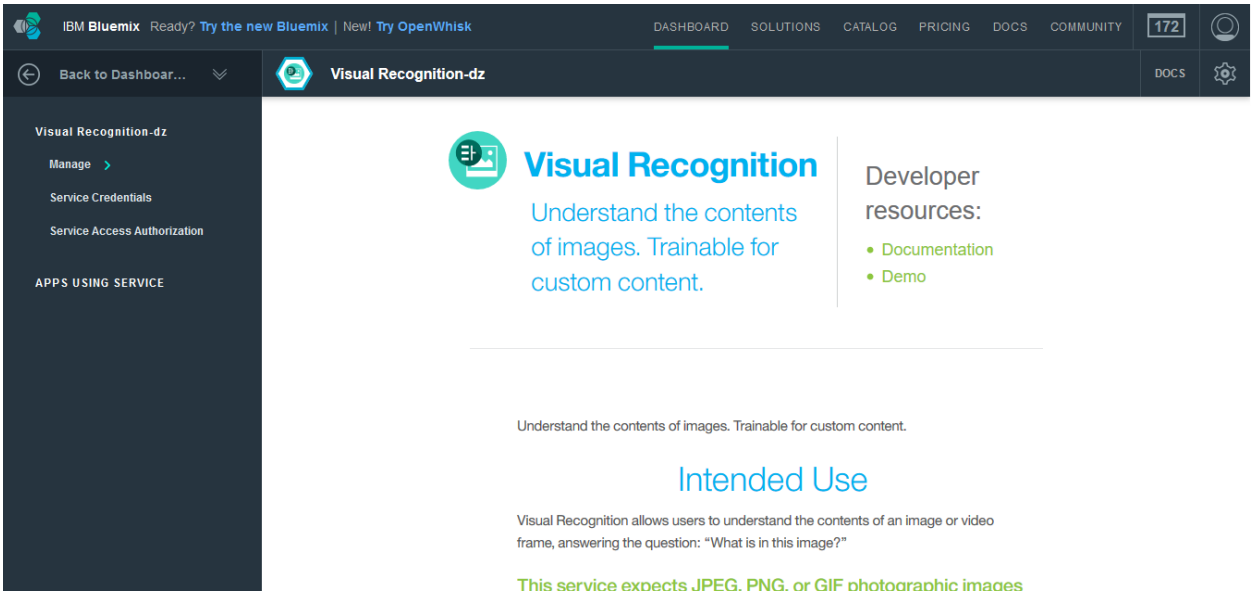
Services

We have used two services namely,

1. Text to speech
2. Visual Recognition

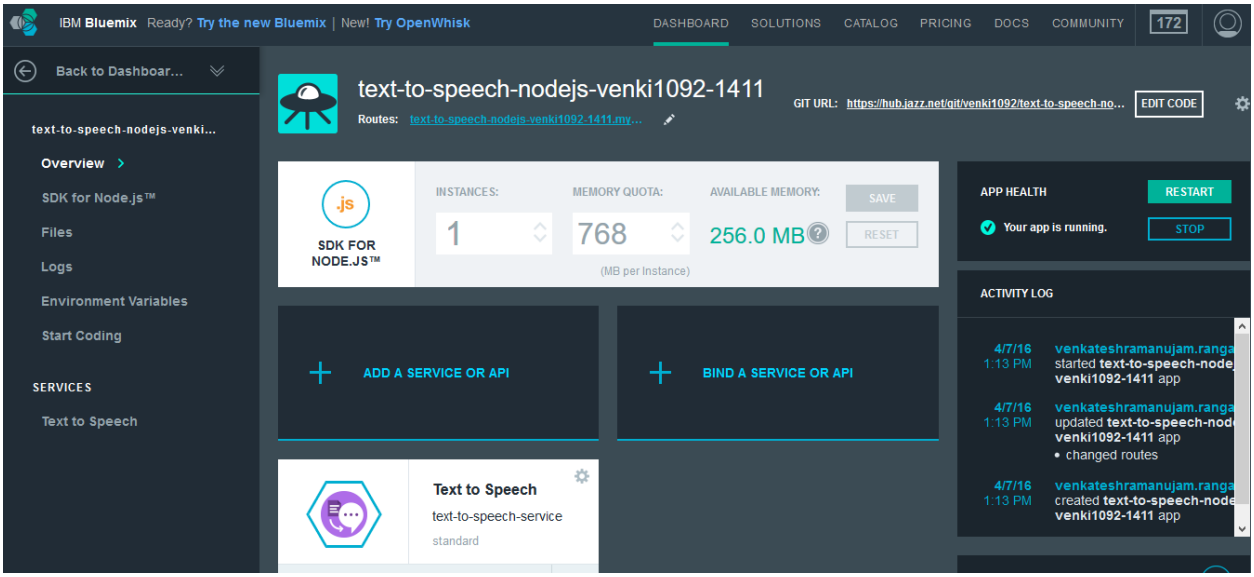
Visual recognition

This service was used to identify the content of the image. The output will be the possible contents of the image in son format.



Text to speech

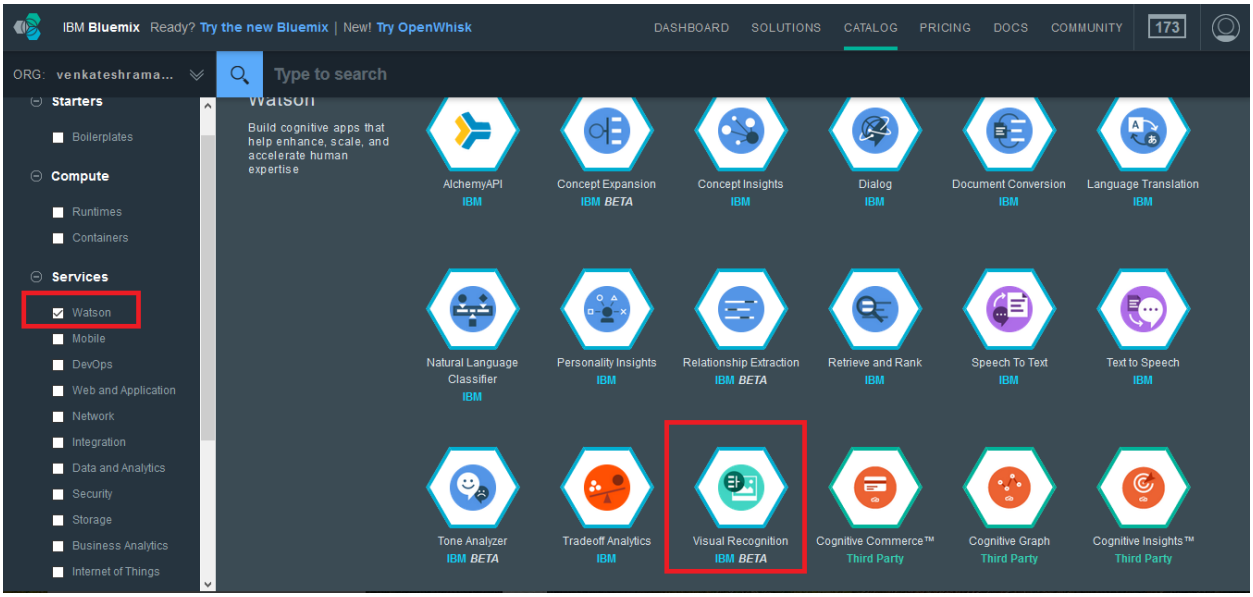
This service was used to give audio output to the visually color blinded user about the content of the image which we will get from the visual recognition engine.



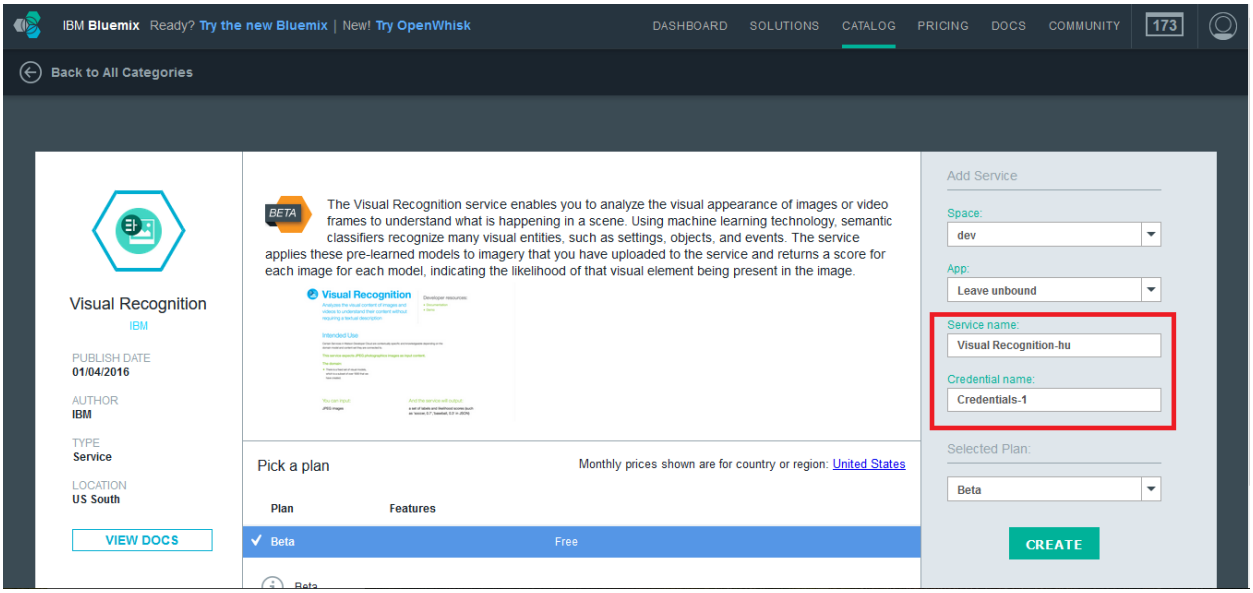
Implementation

Creating Visual Recognition Service

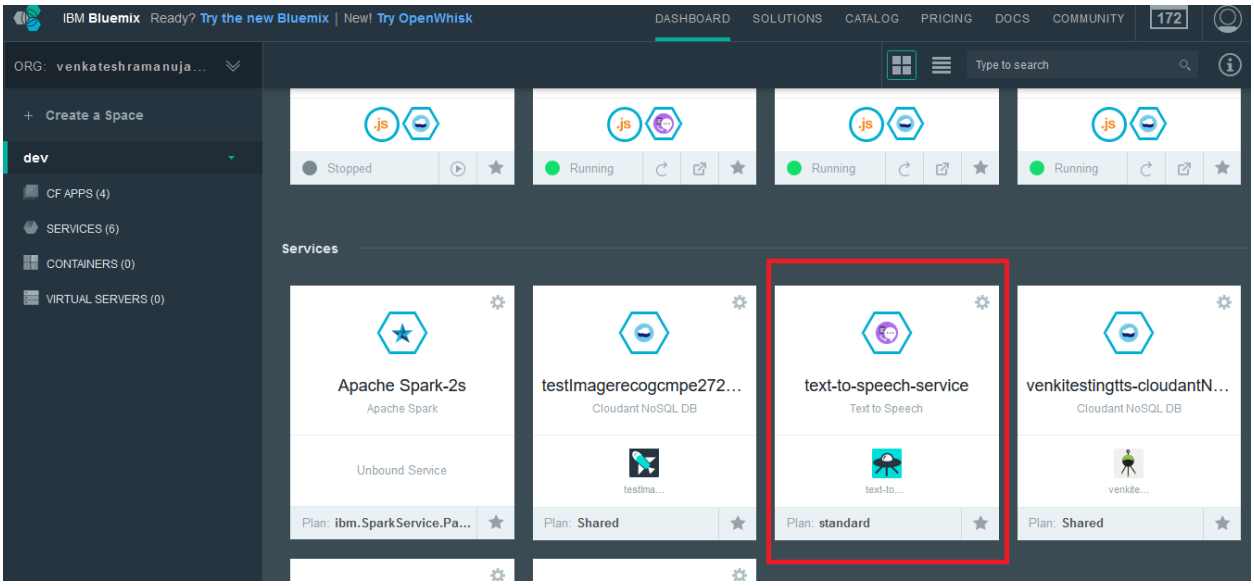
1. Goto IBM bluemix Catalog. Click on Watson services -> Visual Recognition



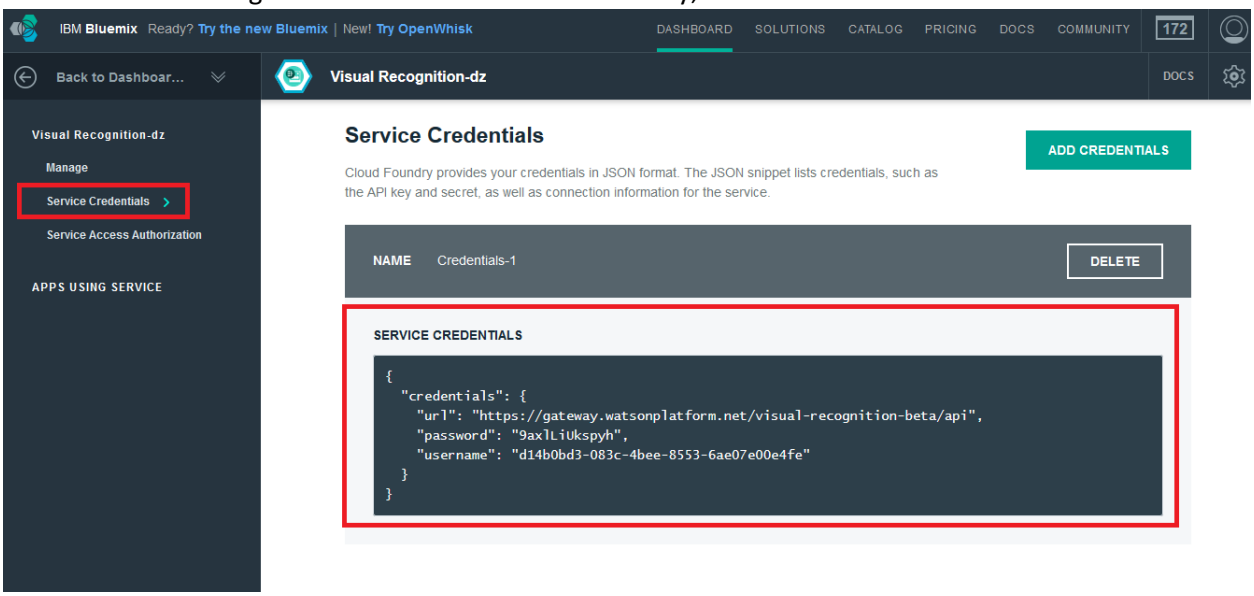
2. In the next screen, enter the service name and credentials as shown in the figure below and press on the create button.



3. After the successful creation of the service, you can view it in your IBM bluemix dashboard.

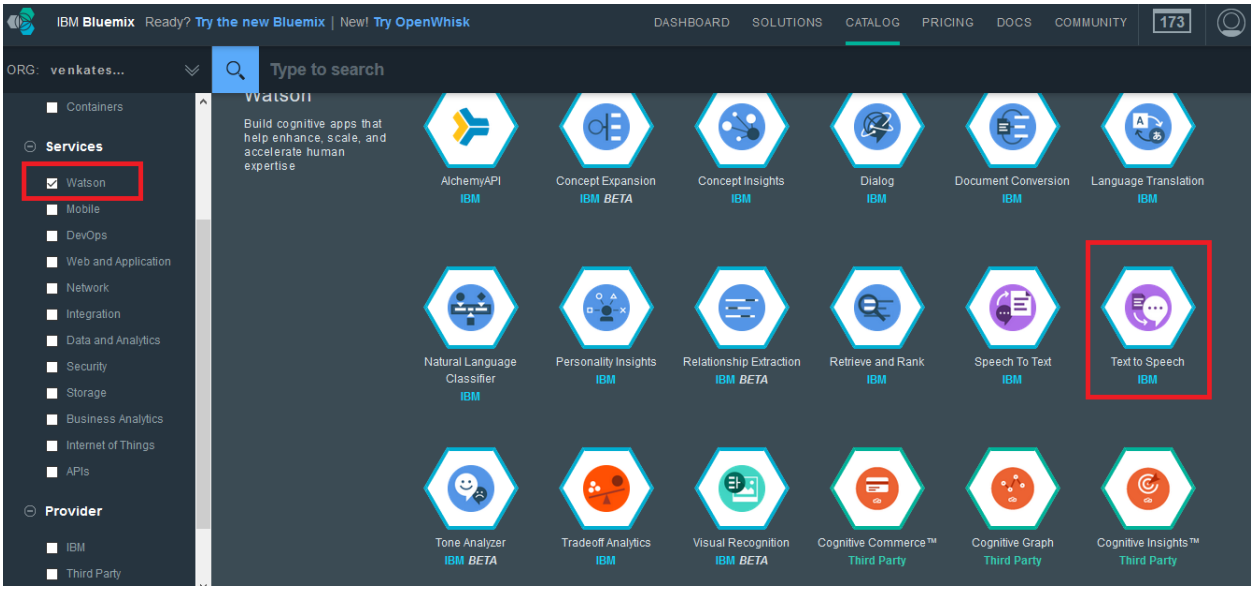


4. Once the Visual recognition service is created successfully, make note of the service credentials.

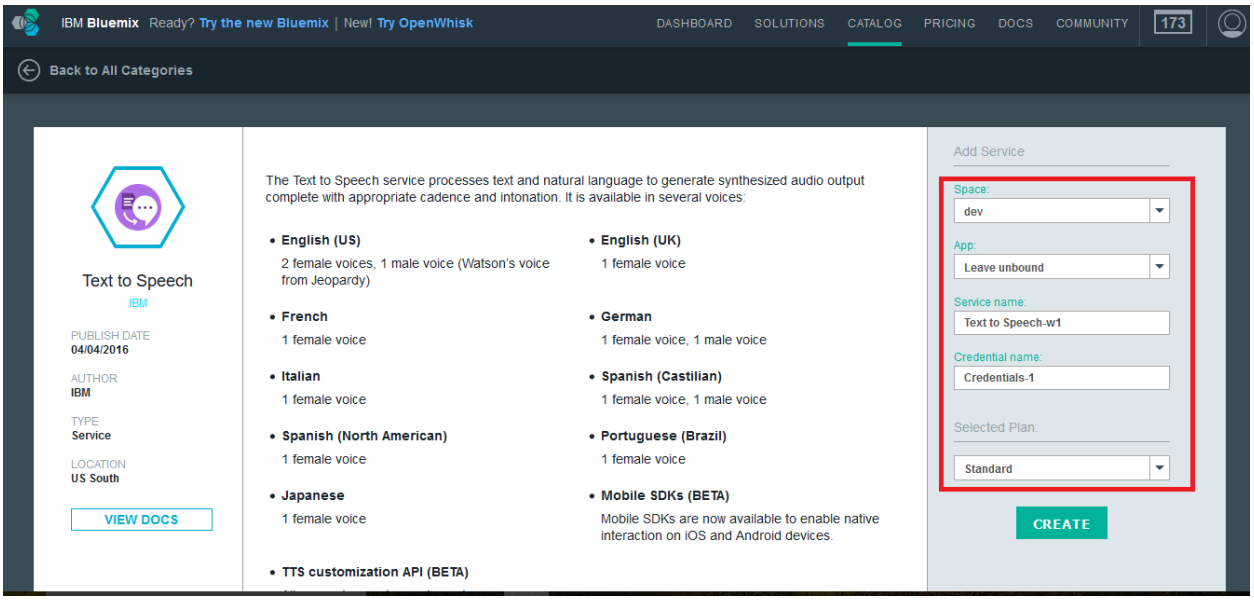


Creating Text to speech service

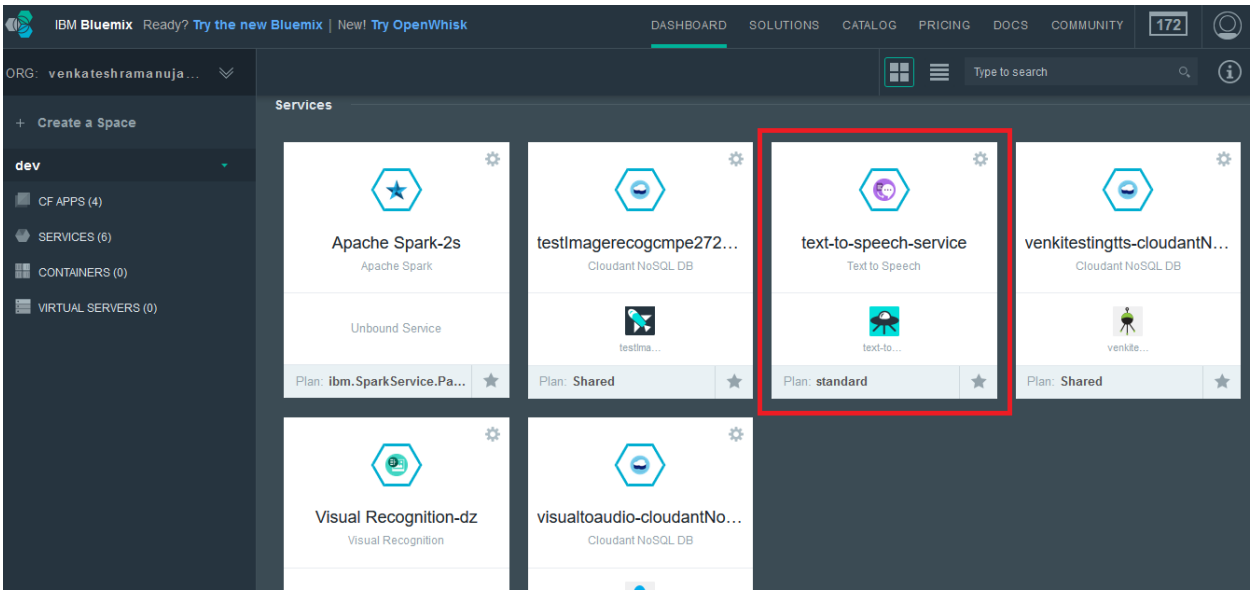
1. Goto IBM Bluemix Catalog. Click on Watson services -> Speech to Text.



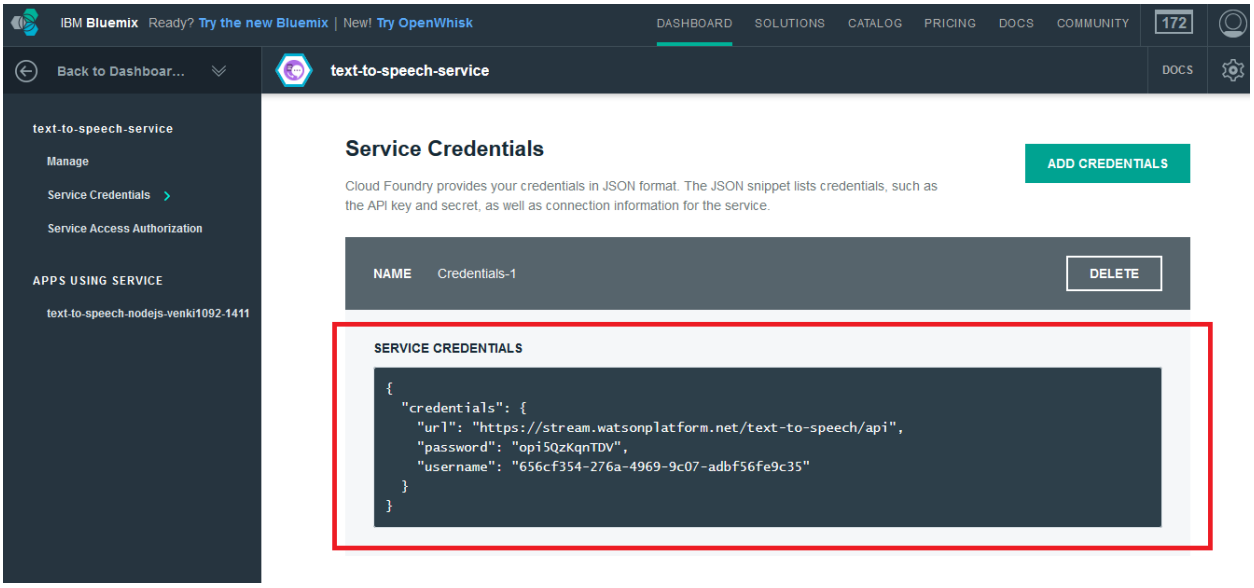
2. In the next screen, enter the service name, credential and select plan for the text to speech service.



3. After the successful creation of the service, you will be able to view it in your dashboard. Click on it in the dashboard.

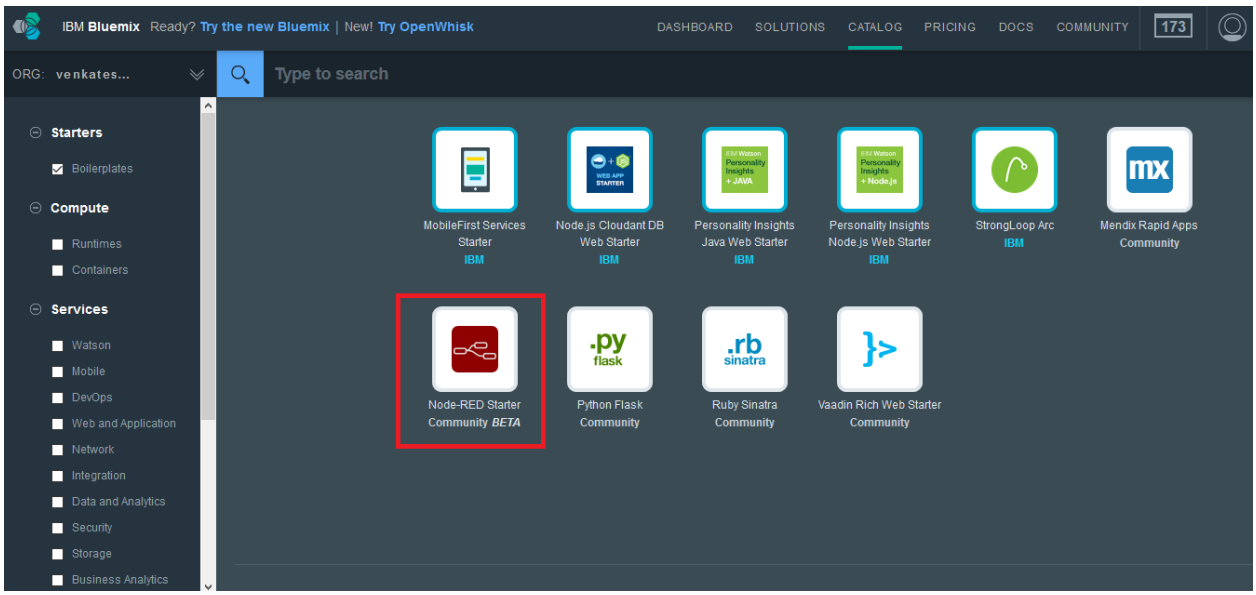


4. Make note of the credentials for the Text to speech service.

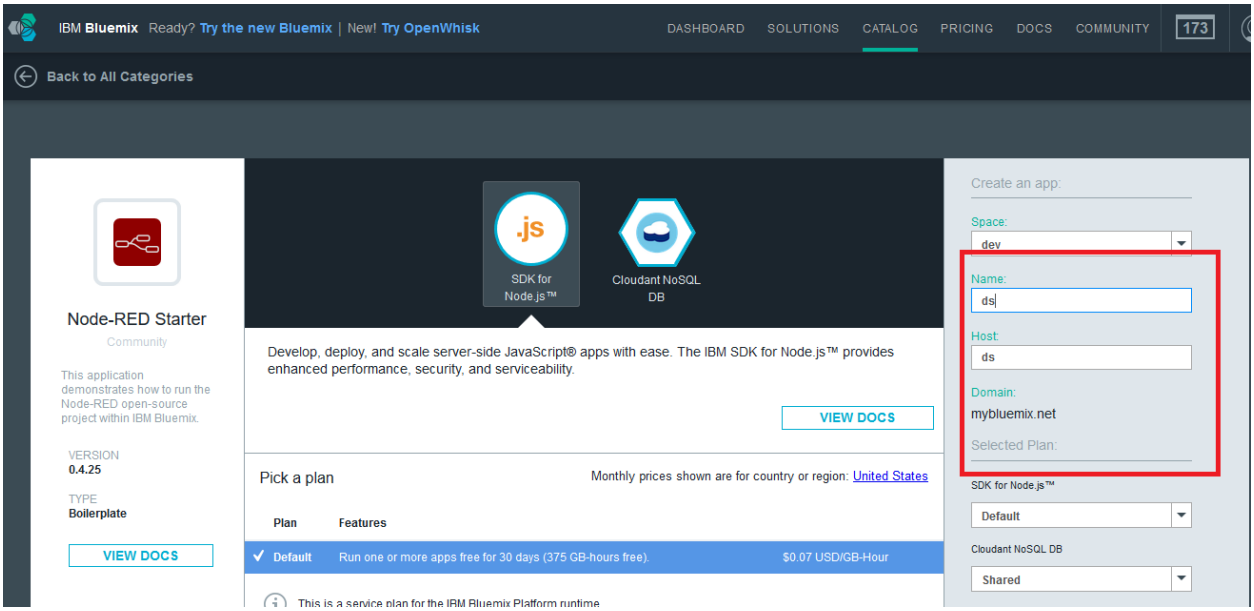


Create Node Red Boilerplate

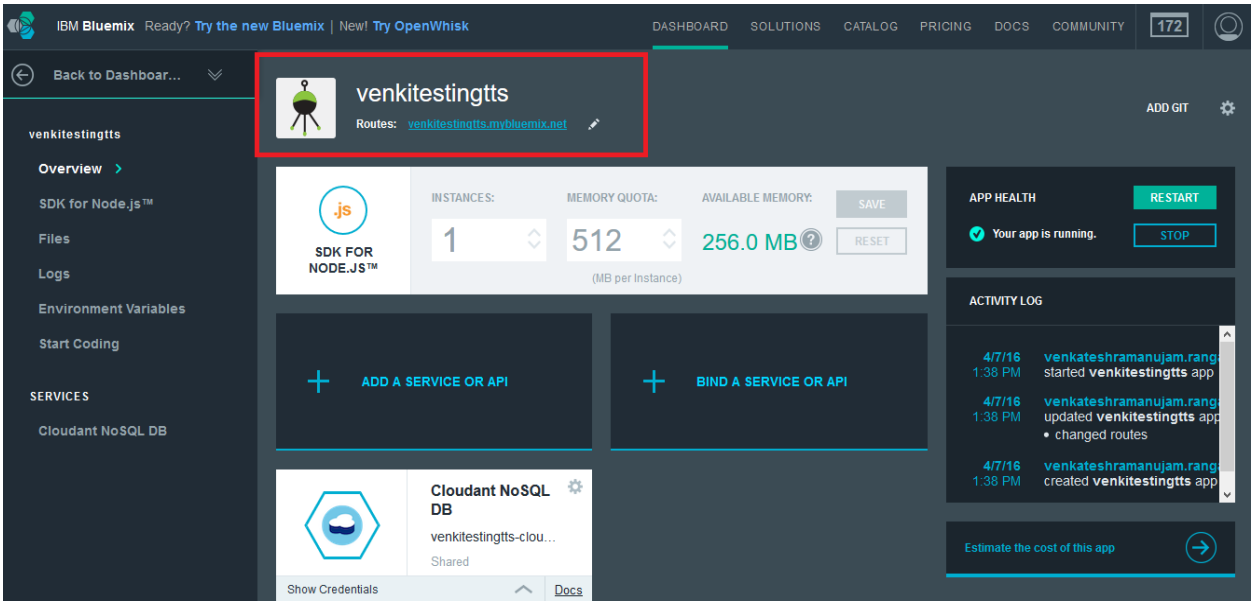
1. Under catalog, click on Boilerplaters -> Node Red Starter Community BETA



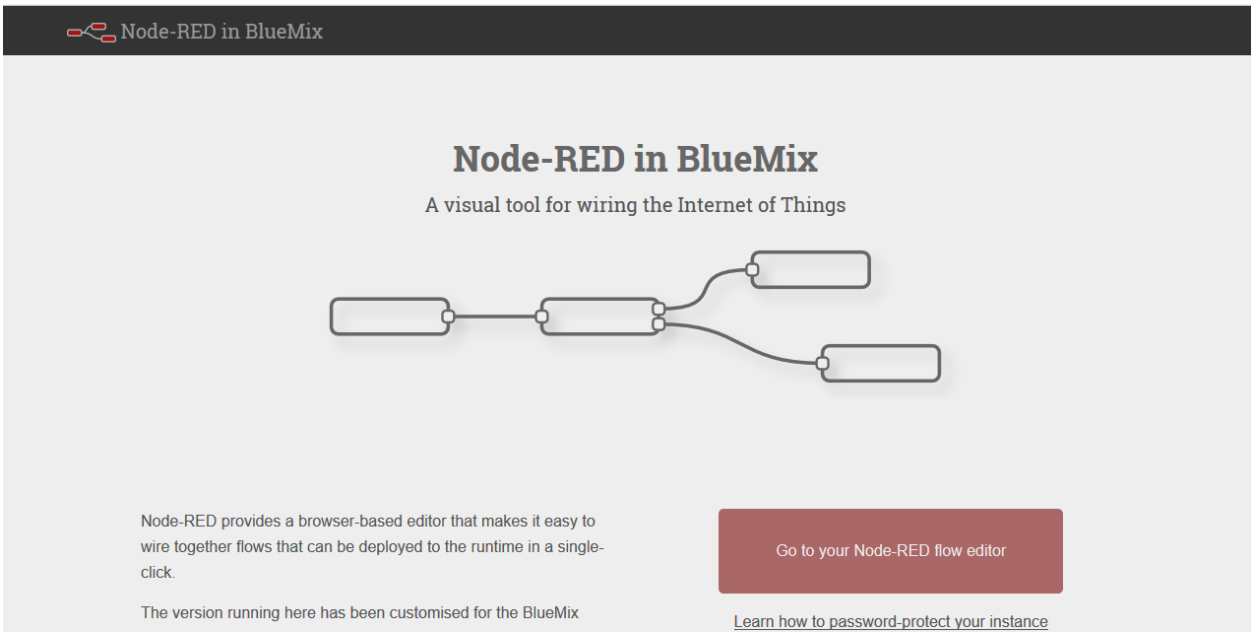
2. Enter the Name and Host fields and click on create button.



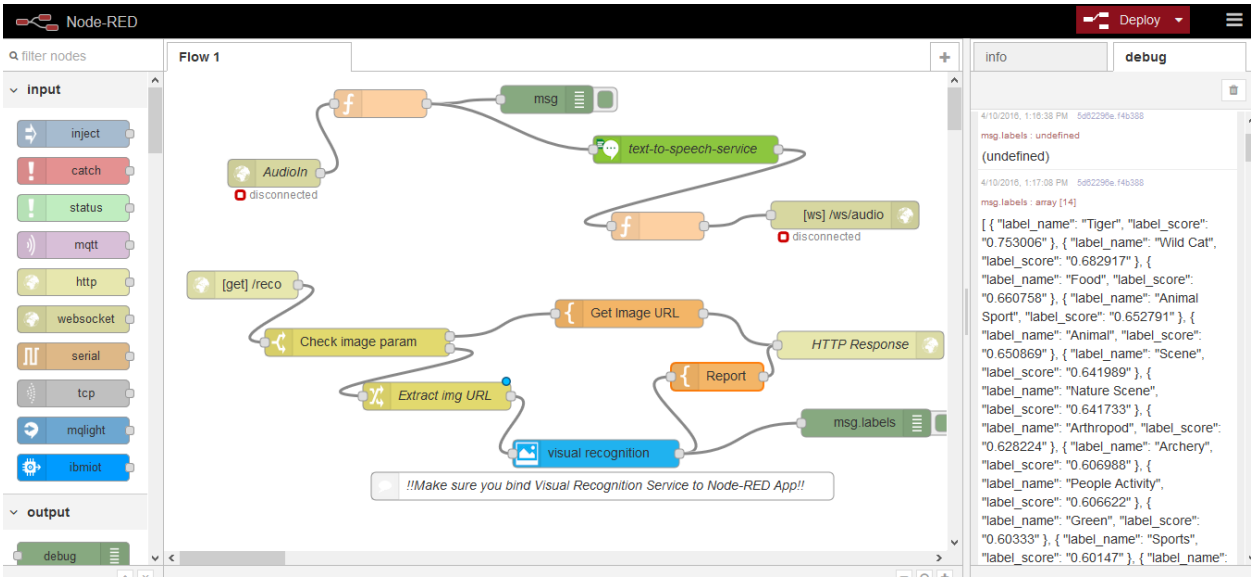
3. Once the boiler plate is created you will be able to view it in your dash board click on it. You will be shown the screen below



4. Click on the routes URL present to go to your node red service page.

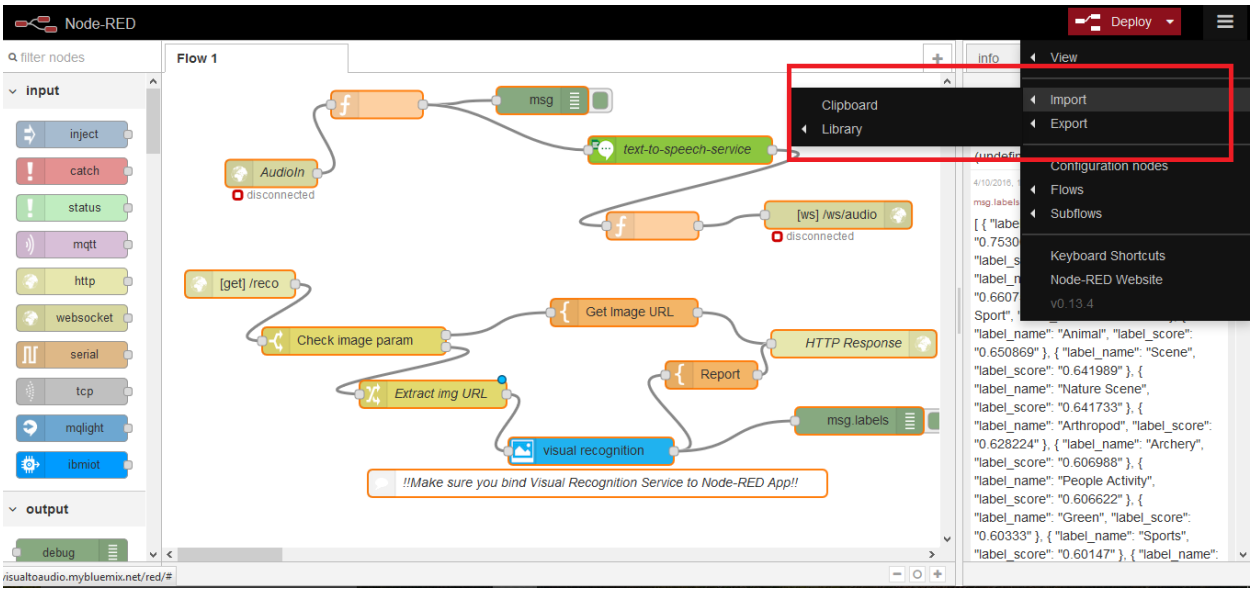


5. Click on the “Go to your node red flow editor” to go to the node red editor



Source code and Deployment

1. Once in the node red editor click in the node red editor, click on the menu -> import -> clipboard



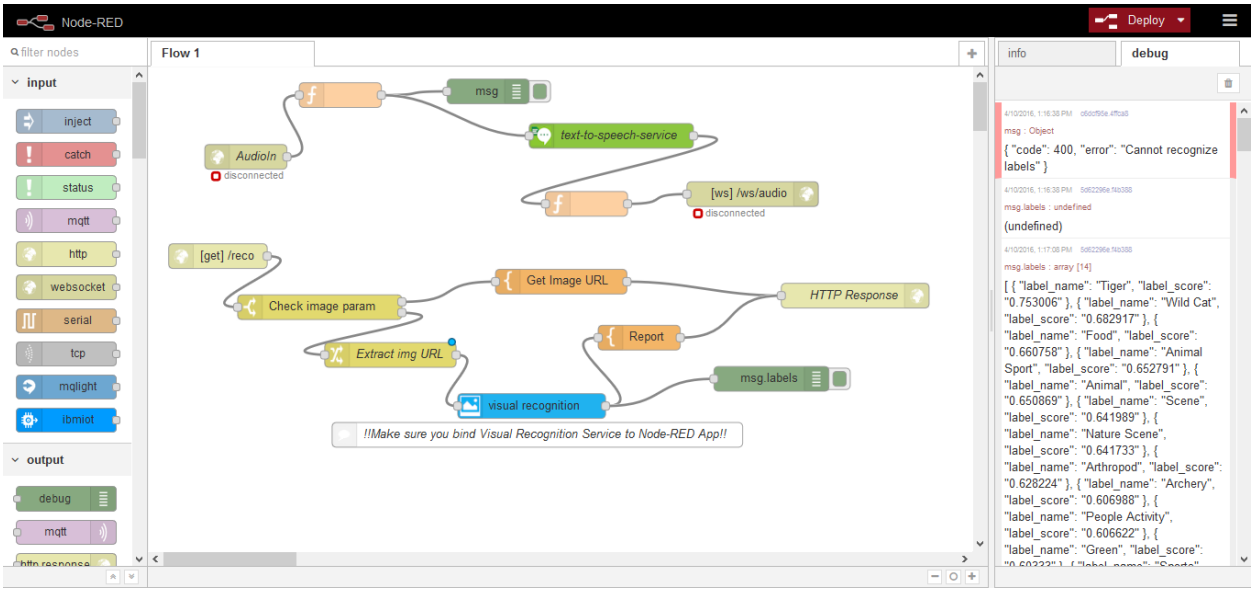
2. Paste the following source which was exported from our implementation. This will include all the nodes along with the codes and settings that each node should possess.

[{"id":"3856d0ac.a1c778","type":"websocket-listener","path":"/ws/audio","wholemsg":"false"},{"id":"90f6f525.9e94e8","type":"websocket-listener","z":"a0671454.b2d3a8","path":"/ws/AudioIn","wholemsg":"false"},{"id":"c6dcf95e.4ffca8","type":"watson-visual-recognition","z":"a0671454.b2d3a8","name":"","x":285.0000305175781,"y":446,"wires":[["9d376a6e.1780a"]]},{"id":"30ba3778.d61a58","type":"http in","z":"a0671454.b2d3a8","name":"","url":"/reco","method":"get","swaggerDoc":"","x":105,"y":230,"wires":[["ed74f92d.255528"]]},{"id":"5a84309d.fe5b4","type":"http response","z":"a0671454.b2d3a8","name":"HTTP Response","x":880.9999694824219,"y":278,"wires":[]},{"id":"ed74f92d.255528","type":"switch","z":"a0671454.b2d3a8","name":"Check image

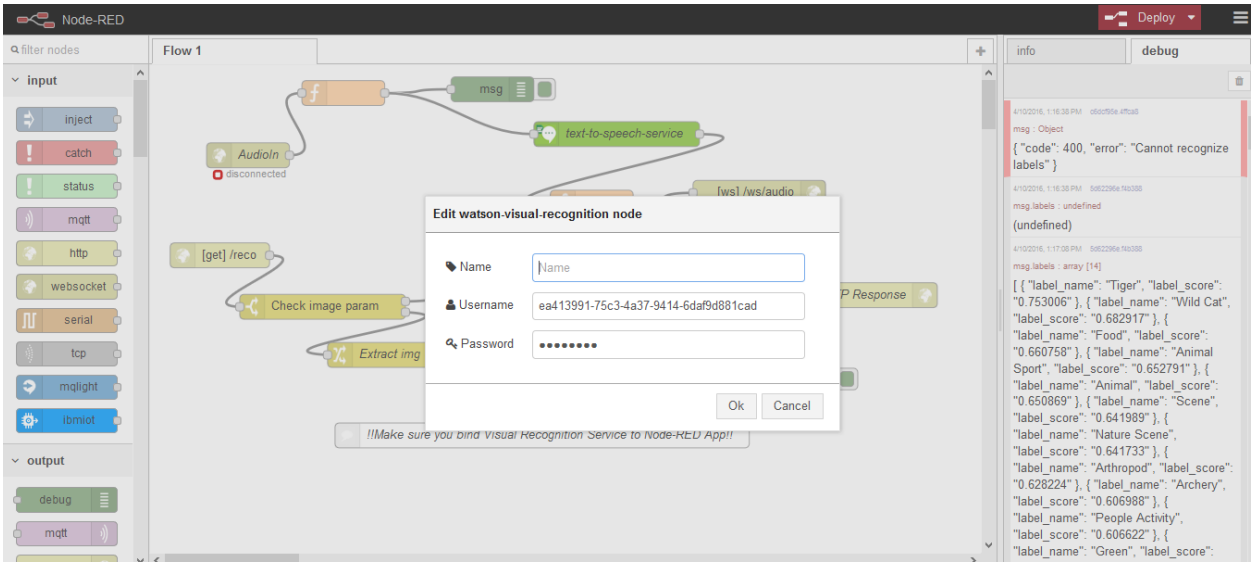
*param","property":"payload.imageurl","rules":[{"t":"null"},{"t":"else"}],"checkall":"false","outputs":2,"x":229,"y":292,"wires":[["dedb6c35.ddbe08"],["e4bdc86b.71b638"]]},{"id":"dedb6c35.ddbe08","type":"template","z":"a0671454.b2d3a8","name":"Get Image URL","field":"payload","fieldType":"msg","format":"html","syntax":"mustache","template":"<html>\n<head>\n<title>CMPE 272 Mini Project Team #17</title>\n<script type=\"text/javascript\">\nfunction checkvalue() { \n var mystring = document.getElementById('imageurl').value; \n if(!mystring.match(/\\S/)) {\n alert (\"The Text input is empty\");\n return false;\n } else {\n return true;\n }\n}\n</script>\n</head>\n<body background = \"#303030\">\n\n<div id = \"header\">\n<h1>Image recognition with Audio feedback</h1>\n \n<div id = \"q\">\n<h2>Sample Images</h2>\n \n<form align = \"center\" onsubmit = \t\"return checkvalue(this)\" action=\"{{req._parsedUrl.pathname}}\">\n<div align = \"center\" >\n\t\n \n \n \n\t\n\t\n\n \n<div align = \"center\">\n\t<h3>Copy above image location URL or enter any image URL<h3>
\n <input type=\"text\" class=\"textbox\" id = \"imageurl\" name=\"imageurl\"/>\n <input type=\"submit\" class=\"button button1\" value=\"Analyze\" id = \"inputurl\" />\n \n</form>\n</body>\n</html>\n<style>\n\n#header {\n background-color: #264a60;\n color:white;\n text-align:center;\n\tposition: relative;\n padding:5px;\n\tfont-family: Gill Sans, Verdana;\n\tfont-size: 11px;\n\tline-height: 14px;\n\tletter-spacing: 2px;\n\tfont-weight: bold;\n}\n#top {\n background: #264a60;\n height: 72px;\n position: relative;\n z-index: 100;\n}\n#q {\n background-color:#848484;\n color:white;\n text-align:center;\n padding:5px;\n\tfont-family: Gill Sans, Verdana;\n\tfont-size: 11px;\n\tline-height: 14px;\n\tletter-spacing: 2px;\n\tfont-weight: bold;\n}\n \n.textbox { \n border: 1px solid #848484; \n -webkit-border-radius: 30px; \n -moz-border-radius: 30px; \n border-radius: 30px; \n outline:0; \n height:25px; \n width: 275px; \n padding-left:10px; \n padding-right:10px; \n } \n\n.button {\n background-color: #4CAF50; \n border: none;\n color: white;\n padding: 16px 32px;\n text-align: center;\n text-decoration: none;\n display: inline-block;\n font-size: 16px;\n margin: 4px 2px;\n -webkit-transition-duration: 0.4s; \n transition-duration: 0.4s;\n cursor: pointer;\n}\n.button1 {\n background-color: white; \n color: black; \n border: 2px solid #4CAF50;\n}\n\n.button1:hover {\n background-color: #4CAF50;\n color: white;\n}\n\n#section {\n\tbackground-color: #808080;\n\tmargin: auto;\n padding: 10px;\n}\n#i1 {\n padding:10px;\n}\n#i2{","x":523,"y":261,"wires":[["5a84309d.fe5b4"]]},{"id":"9d376a6e.1780a","type":"template","z":"a0671454.b2d3a8","name":"Report","field":"payload","fieldType":"msg","format":"html","syntax":"mustache","template":"<html>\n<head>\n<title>CMPE 272 Mini Project Team #17</title>\n<script src=\"https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js\"></script>\n <script type=\"text/javascript\">\n var audioOuputSocket = \"ws://\" + window.location.host + \"/ws/audio\";\n\tvar wsUri = \"ws://{{req.headers.host}}/ws/AudioIn\";\n\tvar ws = new WebSocket(wsUri);\n $(document).ready(function(){\n var audioPlayer = document.getElementById('audioPlayer')\n wsAudioSock = new WebSocket(audioOuputSocket);\n wsAudioSock.onopen = function(){\n var table=document.getElementById(\"lab\");\n\t\tif(table === null || table === undefined || table === \"\"){\n\t\t\tmessageToPlay = \"I am not sure about the image can you retry it once again\";\n\t\t}\n\t\telse{\n\t\t\tmessageToPlay = \"The image strongly resembles \" + table.value;}\n ws.send(JSON.stringify(messageToPlay));\n\n\t\t console.log(\"Connected websocket\");\n };\n wsAudioSock.onerror = function(){ \n console.log(\"Websocket error\"); \n };\n wsAudioSock.onclose = function () {\n \n }\n wsAudioSock.onmessage = function(evt){\n console.log(\"Websocket message\", evt); \n audioPlayer.src = window.URL.createObjectURL(evt.data);\n audioPlayer.play();\n };\n });\n\tfunction sendMessage() {\n var table=document.getElementById(\"lab\");\n if(table === null || table === undefined || table === \"\"){\n messageToPlay = \"Sorry Boss ! I cannot find out who you are\";\n }\n else{\n\t\t messageToPlay = \"The image strongly resembles \" + table.value;}\n ws.send(JSON.stringify(messageToPlay));\n message.value = \"\";\n };\n</script>\n</head>\n<body>\n<div id = \"header\">\n<h1>Image Analysis Result</h1>\n \n<div id = \"q\">\n<h3>Analyzed image:*

{{messageToPlay}}<h3>\n \n<div id = \"image\" align = \"center\" >\n</p>\n \n<div align = \"center\">\n<table border='1'>\n <thead><tr><th>Name</th><th>Score</th></tr></thead>\n{{#labels}}\n <tr><td>{{label_name}}</td><td>{{label_score}}</td></tr>\n{{/labels}}\n</table>\n \n{{#labels}}\n<input type = \"label\" hidden value = \"{{label_name}}\" id = \"lab\"/>\n{{/labels}}\n<div align = \"center\">\n<form action=\"{{req._parsedUrl.pathname}}\">\n <input type=\"submit\" class=\"button button1\" value=\"Try again\"/>\n\t <input type=\"button\" class=\"button button1\" id=\"send_btn\" value=\"Audio\" onclick=sendMessage()>\n</form>\n \n <audio id=\"audioPlayer\"></audio>\n</body>\n</html>\n<style>\n.button {\n background-color: #4CAF50; \n border: none;\n color: white;\n padding: 16px 32px;\n text-align: center;\n text-decoration: none;\n display: inline-block;\n font-size: 16px;\n margin: 4px 2px;\n -webkit-transition-duration: 0.4s; \n transition-duration: 0.4s;\n cursor: pointer;\n}\n.button1 {\n background-color: white; \n color: black; \n border: 2px solid #4CAF50;\n}\n\n.button1:hover {\n background-color: #4CAF50;\n color: white;\n}\n\n#image{\nalign = \"center\";\npadding: 8px;\n}\n\n#header {\n background-color: #264a60;\n color:white;\n text-align:center;\n\tposition: relative;\n padding:5px;\n\tfont-family: Gill Sans, Verdana;\n\tfont-size: 11px;\n\tline-height: 14px;\n\tletter-spacing: 2px;\n\tfont-weight: bold;\n}\n\n#q {\n background-color:#848484;\n color:white;\n text-align:center;\n padding:5px;\n\tfont-family: Gill Sans, Verdana;\n\tfont-size: 11px;\n\tline-height: 14px;\n\tletter-spacing: 2px;\n\tfont-weight: bold;\n}\n\ntable {\n border-collapse: collapse;\n}\n\nth, td {\n text-align: left;\n padding: 8px;\n}\n\ntr:nth-child(even){background-color: #f2f2f2}\n\nth {\n background-color: #4CAF50;\n color: white;\n}\nbody {\n\tbackground-color: #f0f0f0;\n}\n</style>","x":621,"y":329,"wires":[["5a84309d.fe5b4","5d62296e.f4b388"]]},{"id":"e4bdc86b.71b638","type":"change","z":"a0671454.b2d3a8","name":"Extract img URL","rules":[{"t":"set","p":"payload","pt":"msg","to":"payload.imageurl","tot":"msg"}],"action":"","property":"","from":"","to":"","reg":false,"x":315,"y":350,"wires":[["c6dcf95e.4ffca8"]]},{"id":"5d62296e.f4b388","type":"debug","z":"a0671454.b2d3a8","name":"d2","active":false,"console":"false","complete":"labels","x":759,"y":379,"wires":[]},{"id":"1aba718.b5ceb8f","type":"websocket in","z":"a0671454.b2d3a8","name":"AudioIn","server":"90f6f525.9e94e8","client":"","x":137,"y":111.5,"wires":[["e17151bd.5baec8"]]},{"id":"e17151bd.5baec8","type":"function","z":"a0671454.b2d3a8","name":"","func":"var newMsg = { topic:\"\",payload:msg.payload};\nreturn newMsg;","outputs":1,"noerr":0,"x":254,"y":34.5,"wires":[["5b8bc9db.da11f8","57f28a34.26a564"]]},{"id":"5b8bc9db.da11f8","type":"debug","z":"a0671454.b2d3a8","name":"d1","active":false,"console":"false","complete":"true","x":434,"y":30,"wires":[]},{"id":"57f28a34.26a564","type":"watson-text-to-speech","z":"a0671454.b2d3a8","name":"text-to-speech-service","lang":"english","voice":"en-US_MichaelVoice","x":537.0000305175781,"y":91,"wires":[["9dbfe9a.918ee98"]]},{"id":"9dbfe9a.918ee98","type":"function","z":"a0671454.b2d3a8","name":"","func":"msg.payload = msg.speech;\nreturn msg;","outputs":1,"noerr":0,"x":597.818115234375,"y":158.54547119140625,"wires":[["7ca83433.b44664"]]},{"id":"7ca83433.b44664","type":"websocket out","z":"a0671454.b2d3a8","name":"","server":"3856d0ac.a1c778","client":"","x":798.0909423828125,"y":139.72726440429688,"wires":[]},{"id":"d150d77c.ccabe","type":"change","z":"a0671454.b2d3a8","name":"Extract text","rules":[{"t":"set","p":"payload","pt":"msg","to":"payload.message","tot":"msg"}],"action":"","property":"","from":"","to":"","reg":false,"x":843.88330078125,"y":435.8833312988281,"wires":[[]]}]

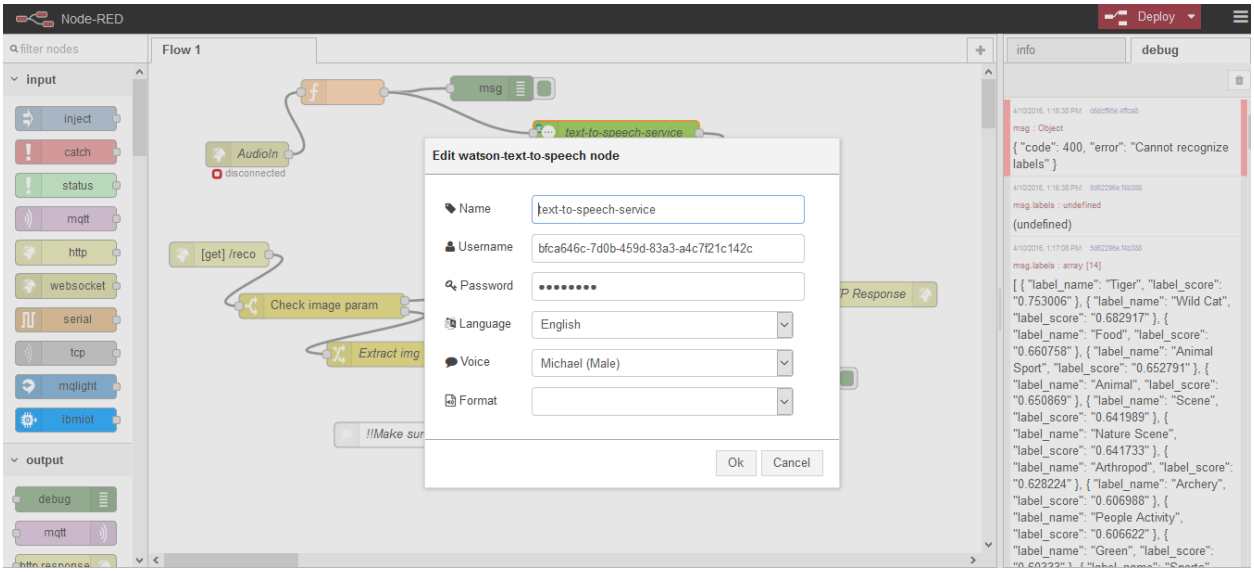
3. You will have the following screen if the nodes are imported properly



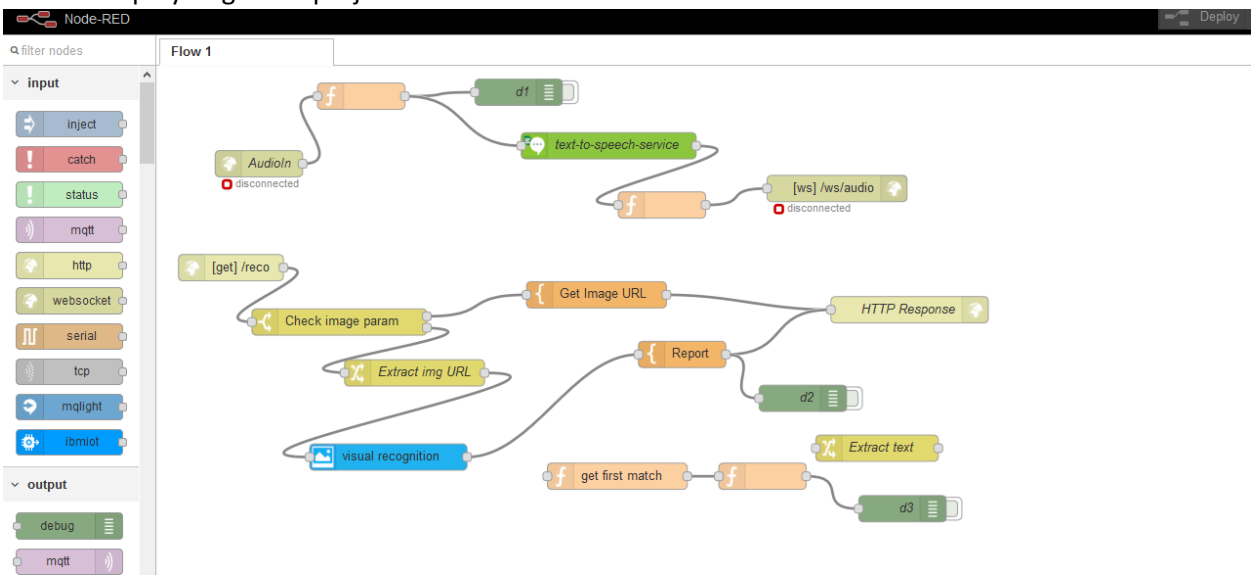
4. Click on the Visual recognition node and enter the credentials that you have noted earlier.



5. Click on the Text to Speech node and enter the credentials that you have saved earlier.



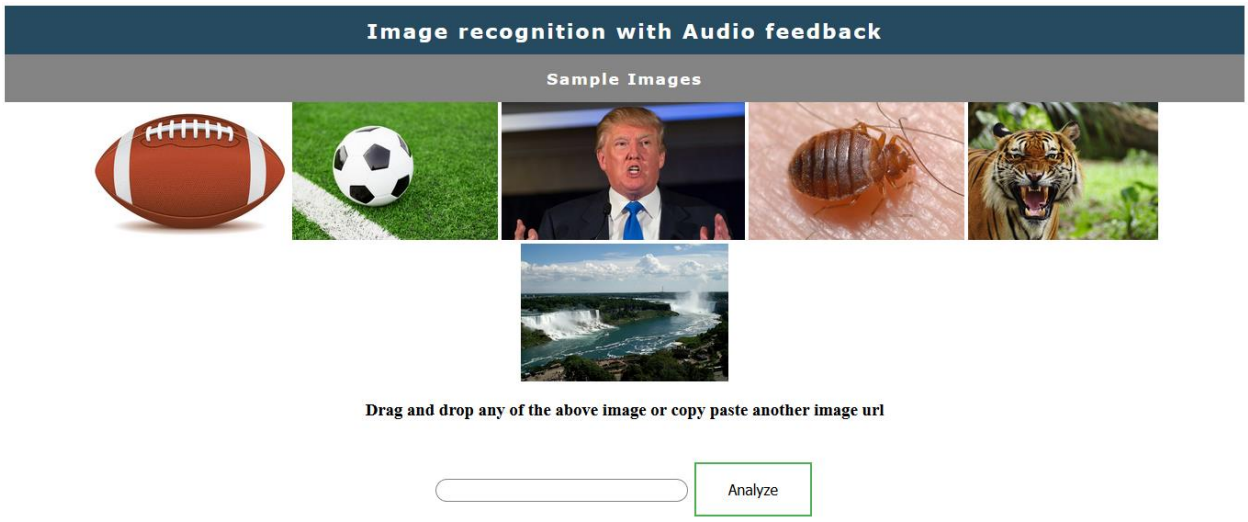
6. Click on deploy to get the project live.



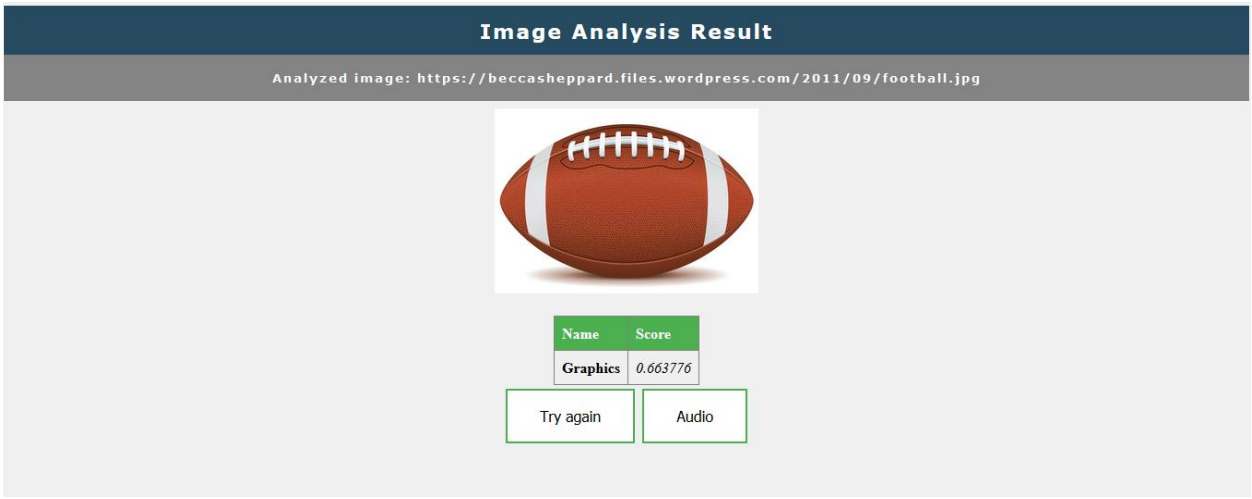
7. Goto http://visualtoaudio.mybluemix.net/reco to see the project that we have implemented.

Screen shots of App

Screen Shot 1



Screen Shot 2



Future work that can be done for improvement

IBM blue visual recognition has means to upload image data for training.

The visual recognition engine can be trained using this feature so that the accuracy can be improved.