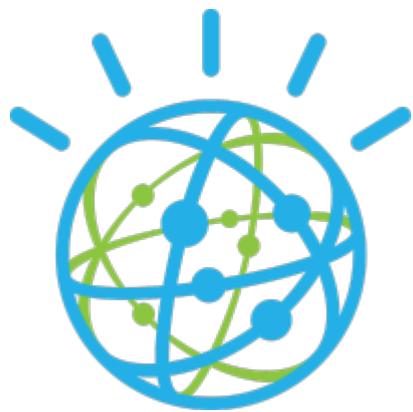
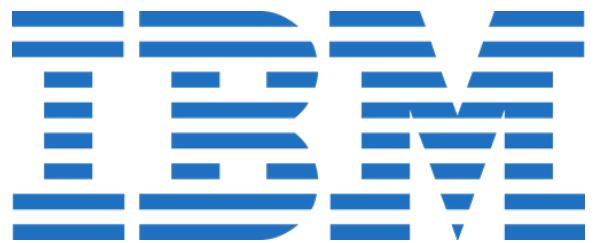


Hands-on Lab



Bluemix

Watson Visual Recognition with Node-RED

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at 'Copyright and trademark information' at www.ibm.com/legal/copytrade.shtml.

This document is current as of the initial date of publication and may be changed by IBM at any time.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

Table of Contents

Introduction	4
Objective	4
Pre-Requisites	4
Steps	4
Step 1: Sign up / Login to Bluemix	5
Step 2 : Create a Node-RED service	5
Step 3: Create the Visual Recognition service	7
Step 4: Create the application flow in the Node-RED editor	8
Step 5: Run the Application	16
Appendix: Application flow file	16
Resources	17

Introduction

In this lab we will build a travel advisor application that recognizes the content of the image which can be used to provide more information about the destination.

InterConnect 2017 Watson Visual Recognition Demo

The screenshot shows the Watson Visual Recognition demo interface. At the top left is a circular icon containing a camera and a bar chart. To its right is the title "Visual Recognition". Below the title is a brief description: "Visual Recognition allows you to derive insights from an image based on its visual content. You can organize image libraries, understand an individual image, and create custom classifiers for specific results that are tailored to your needs." Underneath the description is a section titled "Select an image URL" with a grid of seven thumbnail images representing different landmarks: Taj Mahal, Eiffel Tower, Golden Gate Bridge, Statue of Liberty, White House, Pyramids, and a traditional Chinese building. Below the thumbnails is a text input field with placeholder text "Copy above image location URL or enter any image URL:" followed by a teal "Analyze" button.

Objective

In the following lab, you will learn:

- How to deploy a new Node-RED boilerplate app
- How to create a new Watson Visual Recognition service to analyze images
- How to assemble a Node-RED flow to quickly expose an API and create a process flow

Pre-Requisites

- Get a [Bluemix IBM id](https://bluemix.net) (<https://bluemix.net>) or use an existing account

Steps

1. Sign up / Login to Bluemix
2. Create a Node-RED service
3. Create the Visual Recognition service
4. Create the application flow in the Node-RED editor
5. Run the application

Step 1: Sign up / Login to Bluemix

<https://console.ng.bluemix.net/>

Step 2 : Create a Node-RED service

Click on Node-RED to create the application:

The screenshot shows the IBM Bluemix Catalog interface. A search bar at the top contains the text "Node-RED". On the left, there's a sidebar with categories like Infrastructure, Compute, Storage, Network, Security, Apps (2), Services (1), and a "View all" link. The main area displays search results under the "Apps" section. One result is "Boilerplates", which includes the "Internet of Things Platform Starter" (marked with an IBM icon) and the "Node-RED Starter" (marked with a red robot icon). Both have brief descriptions and "Community" buttons. Below this is another section for "Services", listing "Internet of Things" (with an X-R icon) and "XpertRule Decision Automation for node-RED" (with a third-party icon).

Specify a unique App name:

This screenshot shows the "Create a Cloud Foundry App" page for the "Node-RED Starter" application. At the top, there's a "View all" link. The main form has fields for "App name" (set to "watson-app-lab"), "Host name" (set to "watson-app-lab"), and "Domain" (set to "mybluemix.net"). Below these are sections for "Selected Plan" (choosing "SDK for Node.js™" and "Default") and "Cloudant NoSQL DB" (choosing "Lite"). On the left, there's a sidebar with "Node-RED Starter" details, a "Community" button, and a "View Docs" link. It also lists the app's version (0.5.1), type (Boilerplate), and region (US South).

Staging will take a few minutes

Hands-on lab | Using Watson Visual Recognition w/Node-RED

Check the Overview section to see when it's up and running:

You can verify now that a Node-RED application is created with a Cloudant NoSQL database:

Step 3: Create the Visual Recognition service

Go to the catalog and search for "Visual Recognition":

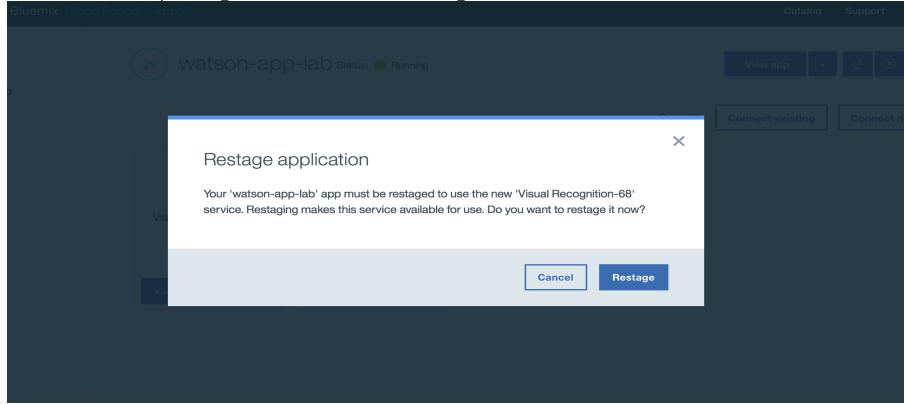
The screenshot shows the IBM Bluemix Catalog interface. In the top navigation bar, there are links for Catalog, Support, and Account. On the left sidebar, there are categories like Infrastructure, Apps, and Services. The main search bar has the word 'Visual' typed into it. Below the search bar, there's a section titled 'Services' under 'Watson'. It includes three service cards: 'AlchemyAPI' (deprecated), 'Tradeoff Analytics', and 'Visual Recognition'. The 'Visual Recognition' card is highlighted with a blue border. At the bottom of the page, there's a section for 'Internet of Things' with a service card for 'AT&T M2X'.

Select the Visual Recognition service and connect it to the Node-RED application from the dropdown:

This screenshot shows the detailed product page for the Visual Recognition service. At the top, there's a brief description: "Find meaning in visual content! Analyze images for scenes, objects, faces, and other content. Choose a default model off the shelf, or create your own custom classifier. Find similar images within a collection. Develop smart applications that analyze the visual content of images or video frames to understand what is happening in a scene." Below this, there's a "Service name:" field containing "Visual Recognition-68". The "Connect to:" dropdown is set to "watson-app-lab". Under the "Features" section, there are four bullet points: "General Classification", "Visual Training", "Face Detection", and "Similar Image Search (BETA)". At the bottom, there's a "Pricing Plans" section and a "Create" button.

To bind the Visual Recognition service to the Node-RED application, click 'Restage' to restart the node-red application.

Hands-on lab | Using Watson Visual Recognition w/Node-RED



Your Visual recognition service is now bound to the Node-RED application

A screenshot of the IBM Bluemix dashboard. On the left, a sidebar menu is open with the "Connections" option selected. The main area displays the "watson-app-lab" application details. Under the "Connections" section, two services are listed: "Visual Recognition-68" (Visual Recognition free) and "watson-app-lab-clouda..." (Cloudant NoSQL DB Lite). Each service card has "View credentials" and "Docs" buttons at the bottom. The "Visual Recognition-68" card also has a "..." button.

Click on the app URL and navigate to the Node-RED flow editor

Go to your Node-RED flow editor

Step 4: Create the application flow in the Node-RED editor

4.1) Drag and drop the http node from the input section on the left and also get the http response node from the output section.



4.2) Double-click on the http input node to configure the GET URL as shown below and click 'Ok'

Edit http in node

Method	GET
URL	/imagerecognition
Name	Name

Ok **Cancel**

4.3) Double-click on the http response node and provide a name for the node

Edit http response node

Name	HTTP Response
------	---------------

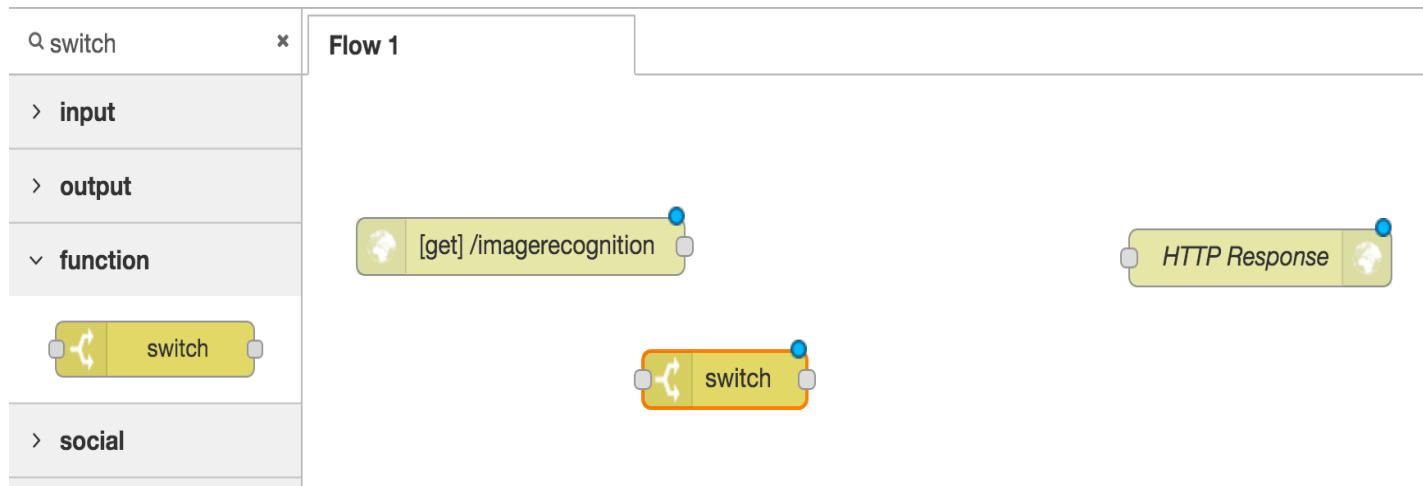
The messages sent to this node **must** originate from an *http input* node

Ok **Cancel**

Click the 'Deploy' button on the top-right section of the editor to deploy the application



4.4) Search for the 'switch' node and add it to the flow editor



Configure the switch node that checks if no URL has been provided then reload the same page (switch nodes are mostly used for if-else cases)

Edit switch node

Name: Check Image Paramter

Property: msg.payload.imageurl

```

graph TD
    A[is null] --> 1
    B[otherwise] --> 2

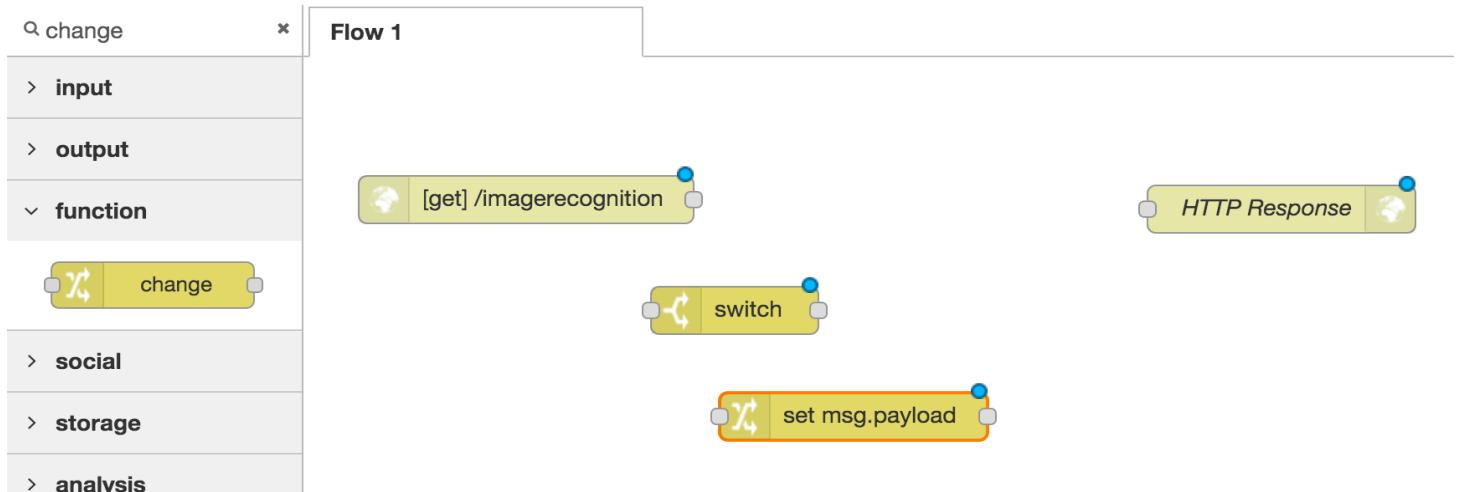
```

+ rule

stopping after first match

Ok Cancel

4.5) Search for the 'change' node and add it to the flow editor



Configure it to parse and pass imageurl from the payload object

Edit change node

Name Extract Image URL

Rules

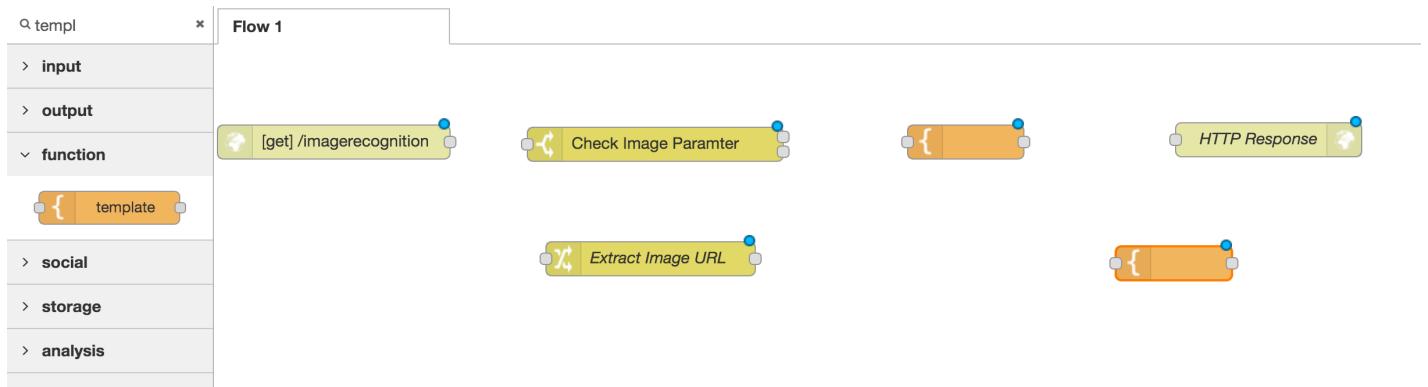
```

Set      ▾ msg.payload
to      ▾ msg.payload.imageurl
  
```

+ rule

Ok Cancel

4.6) Search for the 'template' node and add 2 template nodes to the editor - 1- Building homepage and 2- Generate report



Configure one of the template nodes as Creates Application Homepage

Change the Syntax Highlight to HTML from the dropdown on the left

Copy the raw html content from this same data to the template section [Sample HTML Form Content](#) (https://github.com/varshanagraj/interconnect17/blob/master/watsonvisualrecognition/application_forms/application_homepage.html)

Edit template node

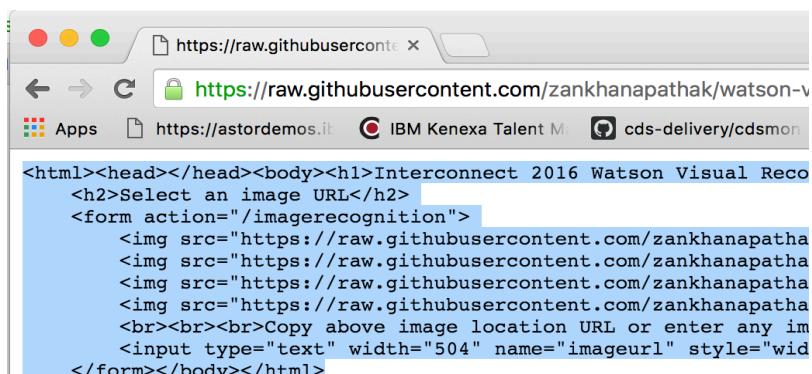
Name Create Application Homepage

Set property msg.payload

Template Syntax Highlight:

```
i 1 <html><head></head><body><h1>Interconnect 2016 Watson Visual Recognition De
2   <h2>Select an image URL</h2>
3   <form action="/imagerecognition">
4     <br><br>Copy above image location URL or enter any image URL:
9     <input type="text" width="504" name="imageurl" style="width: 400;">
10    </form></body></html>
```

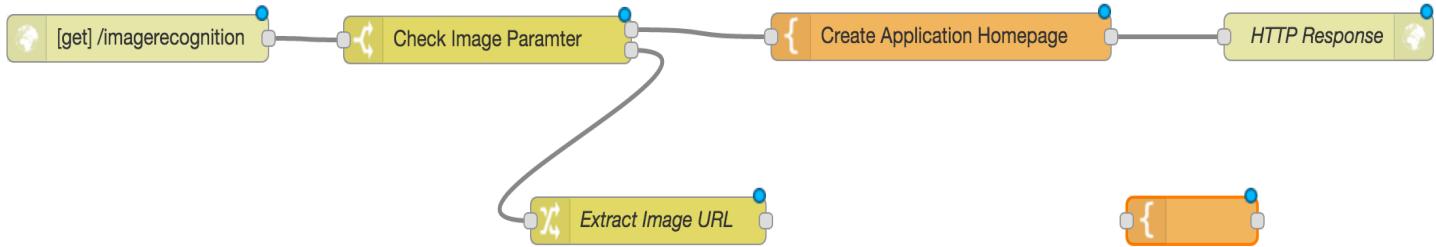
Format Mustache template



Connect the Nodes as shown below:

Hands-on lab | Using Watson Visual Recognition w/Node-RED

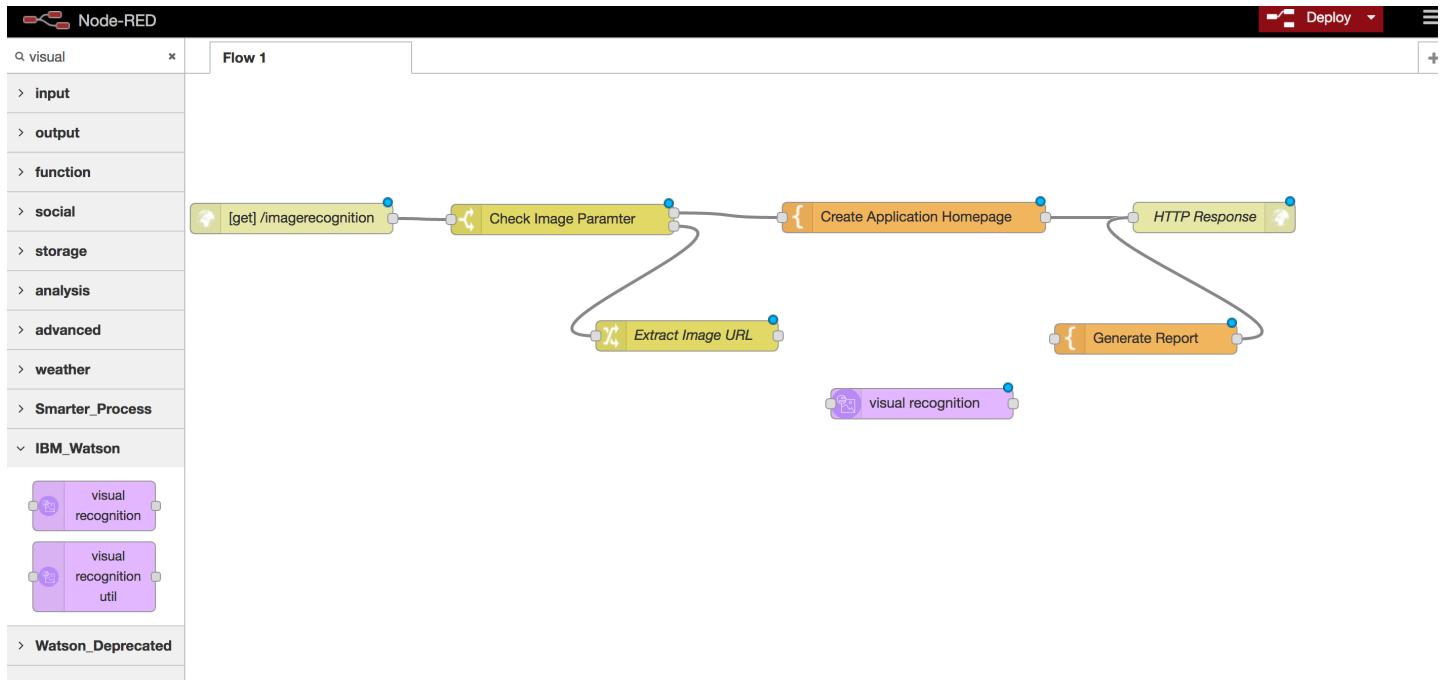
To explain the flow- the Get call will pass the imageurl info as payload to the Image parameter node, which then will parse the input and check if it's null (will reload the same home page and send as response), and if any input is provided it will send the payload to extract the image URL node to process it further.



Click the 'Deploy' button on the top right section of the editor to deploy the application



4.7) Search for the 'Visual Recognition' node and add it to the flow editor and connect it to the Extract Image URL node as shown below:



4.8) Configure the remaining 'template' node to generate reports - this template node gets the result from the 'visual recognition' node and sends the output to the 'HTTP Response' node

Copy the raw html content from this same data to the template section [Sample HTML Form Content](#) (https://github.com/varshanagraj/interconnect17/blob/master/watsonvisualrecognition/application_forms/report_form.html)

Edit template node

Name: Generate Report

Set property: msg. payload

Template: Syntax Highlight: HTML

```

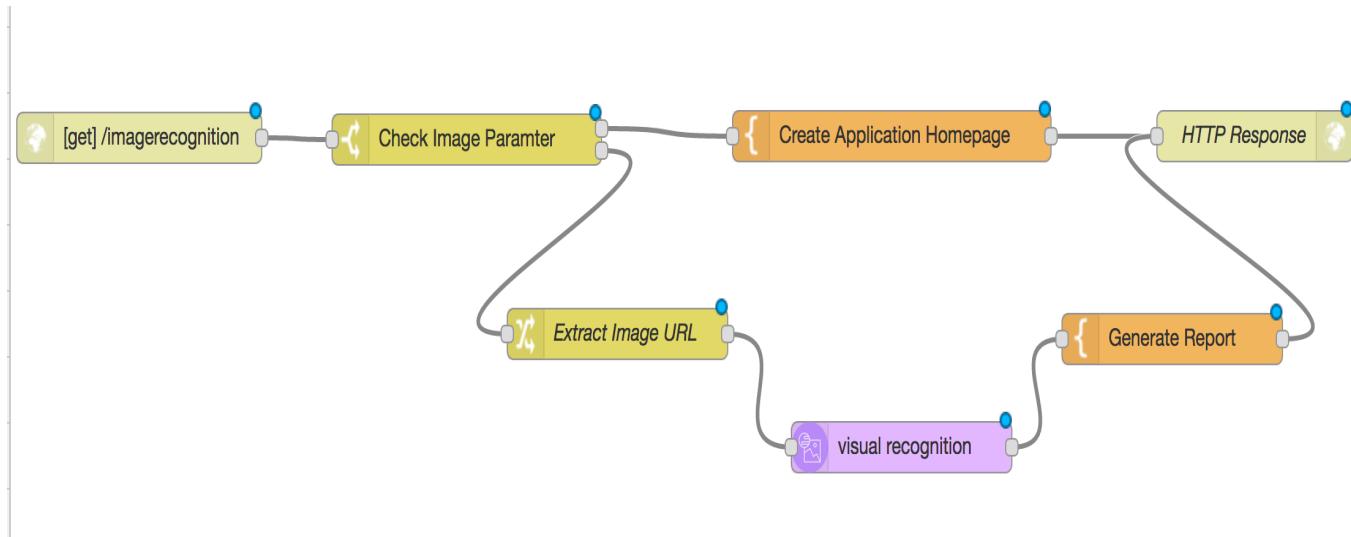
1  <h1>Node-RED Watson Visual Recognition output</h1>
2  <p>Analyzed image: {{payload}}<br/><img src='{{image}}' border='1'>
3  <table border='1'>
4      <thead><tr><th>Name</th><th>Score</th></tr>
5      <tbody>{{#labels}}
6          <tr><td><b>{{label_name}}</b></td><td>{{score}}</td></tr>
7      {{/labels}}
8  </tbody>
9  </table>
10 <p>{{msg}}</p>
11 <form action="{{req._parsedUrl.pathname}}" method="post">
12     <input type="submit" value="Try again"/>
13 </form>

```

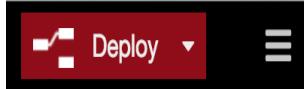
Format: Mustache template

Buttons: Ok, Cancel

Connect the Visual Recognition, Generate Report and HTTP Response nodes as shown below:



Click the 'Deploy' button on the top right section of the editor to deploy the application



The application is now ready for use

Step 5: Run the Application

Go to your browser and change the URL from the Node-RED application '/red' to '/imagerecognition':

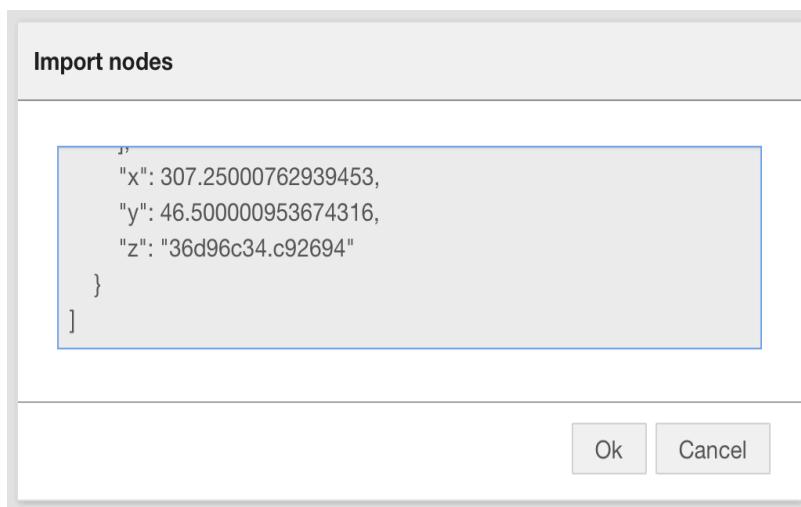
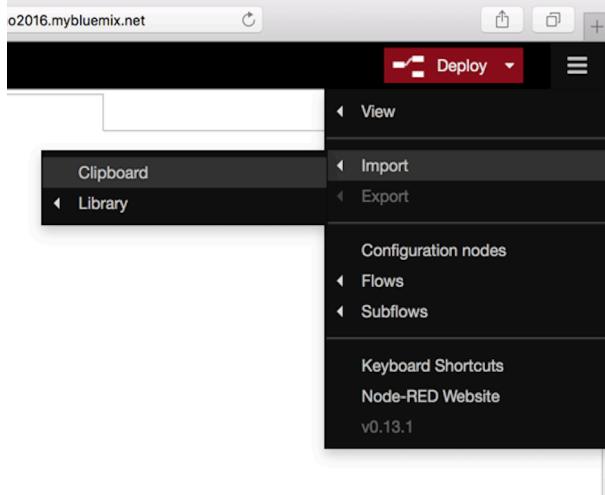
The screenshot shows a web browser window with the following details:

- Address Bar:** watson-lab.mybluemix.net/imagerecognition
- Title Bar:** InterConnect 2017 Watson Visual Recognition Demo
- Header:** Visual Recognition (with a camera icon)
- Text:** Visual Recognition allows you to derive insights from an image based on its visual content. You can organize image libraries, understand an individual image, and create custom classifiers for specific results that are tailored to your needs.
- Section:** Select an image URL
- Image Preview:** A row of seven thumbnail images representing various landmarks: Taj Mahal, Eiffel Tower, Golden Gate Bridge, Statue of Liberty, White House, Pyramids, and a traditional Chinese building.
- Input Field:** Copy above image location URL or enter any image URL:
- Button:** Analyze

Appendix: Application flow file

You can download the complete flow file [here](#)

(https://github.com/varshanagraj/interconnect17/blob/master/watsonvisualrecognition/application_forms/visualapp_node_red_export.json) and import it into your Node-RED instance.



Resources

For additional information regarding development and Bluemix:

- [GitHub Guides](https://guides.github.com/) (<https://guides.github.com/>)
- [Get started guides for your favorite runtimes](https://www.ibm.com/blogs/bluemix/2017/03/runtimes-get-started-guides/?social_post=829410659&fst=Learn&linkId=35308736) (https://www.ibm.com/blogs/bluemix/2017/03/runtimes-get-started-guides/?social_post=829410659&fst=Learn&linkId=35308736)