Node-RED and Watson Visual Recognition

Hands-on Workshop

Urs Witzig
Yamini Rao

IBM.

Overview

The IBM Watson Developer Cloud (WDC) offers a variety of services for developing cognitive applications. Each Watson service provides a Representational State Transfer (REST) Application Programming Interface (API) for interacting with the service. Some services, such as the Speech to Text service, provide additional interfaces.

This workshop will use the Visual Recognition service to show the integration into Watson Cloud services.

Node-RED is a visual tool for wiring the Internet of Things. It is easy to connect devices, data and API's (services). It can also be used for other types of applications to quickly assemble flows of services. Node-RED is available as open source and has been implemented by the IBM Emerging Technology organization. Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range of nodes in the palette. Flows can be then deployed to the runtime in a single-click. While Node-Red is based on Node.js, JavaScript functions can be created within the editor using a rich text editor. A built-in library allows you to save useful functions, templates or flows for re-use.

Node-RED is included in the Node-RED starter application in IBM Cloud but you can also deploy it as a stand alone Node.js application. Node-RED can not only be used for IoT applications, but it is a generic event-processing engine. For example, you can use it to listen to events from http, websockets, tcp, Twitter and more and store this data in databases without having to program much if at all. You can also use it for example to implement simple REST APIs. You can find many other sample flows on the Node-RED website.

This app in this lab will be created and run on your IBM Cloud Account.

- In a first step, a IBM Cloud Node-RED environment will be created and setup. This will then host your Node-RED flows.

- Next you will create a simple Node-RED flow that allows you to past an Image URL from the Web and pass it to the IBM Watson Visual Recognition service you have created in an earlier lab.

Objectives

- Learn how to use the Node-Red environment on IBM Cloud

- Learn how to implement the Image-Analysis application using Node-Red

Prerequisites

- The prerequisite for this lab is an IBM Cloud account and a Watson Visual Recognition service

- Steps to create a Visual Recognition Service in the IBM Cloud console and retrieve the key from there (Service Credentials) is included below.

Section 1: *Creating services in IBM Cloud*

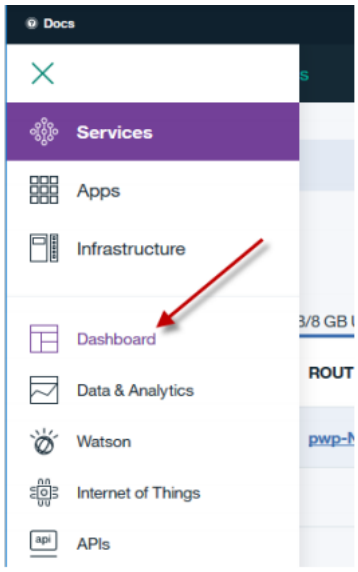
IBM Cloud offers services, or cloud extensions, that provide additional functionality that is ready to use by your application's running code.

You have two options for working with applications and services in IBM Cloud. You can use the IBM Cloud web user interface or the Cloud Foundry command-line interface (cf cli).

Step 1: Create a Visual Recognition service (On the IBM Cloud Console)

a) Open a Browser and navigate to your IBM Cloud Interface

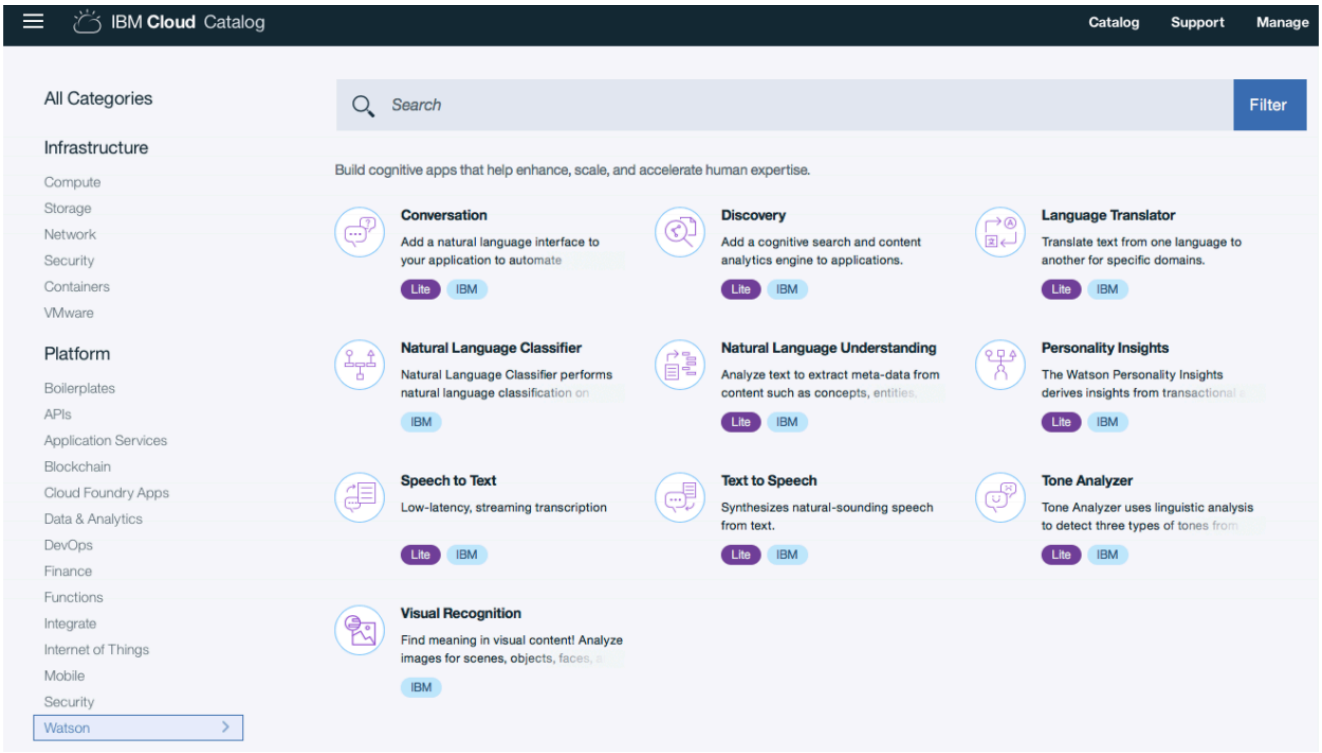
b) Click the menu icon and select Services and then Dashboard.



IBM

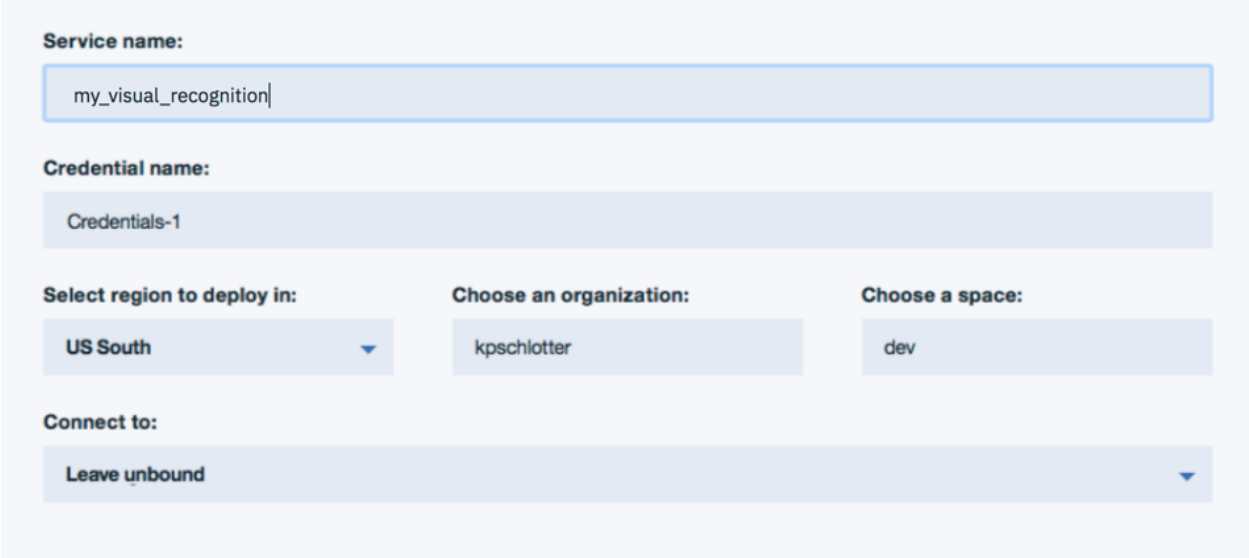
c) Click the Create Resource button

d) Under Services select the category Watson

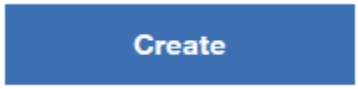


e) Click the **Visual Recognition** service and as *Service name* specify **my_visual_recognition**

You can leave the *Credentials name* as the default.

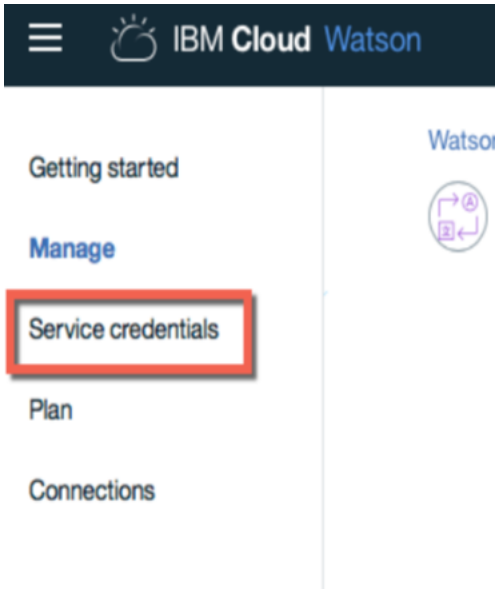


f) Click the **Create** button



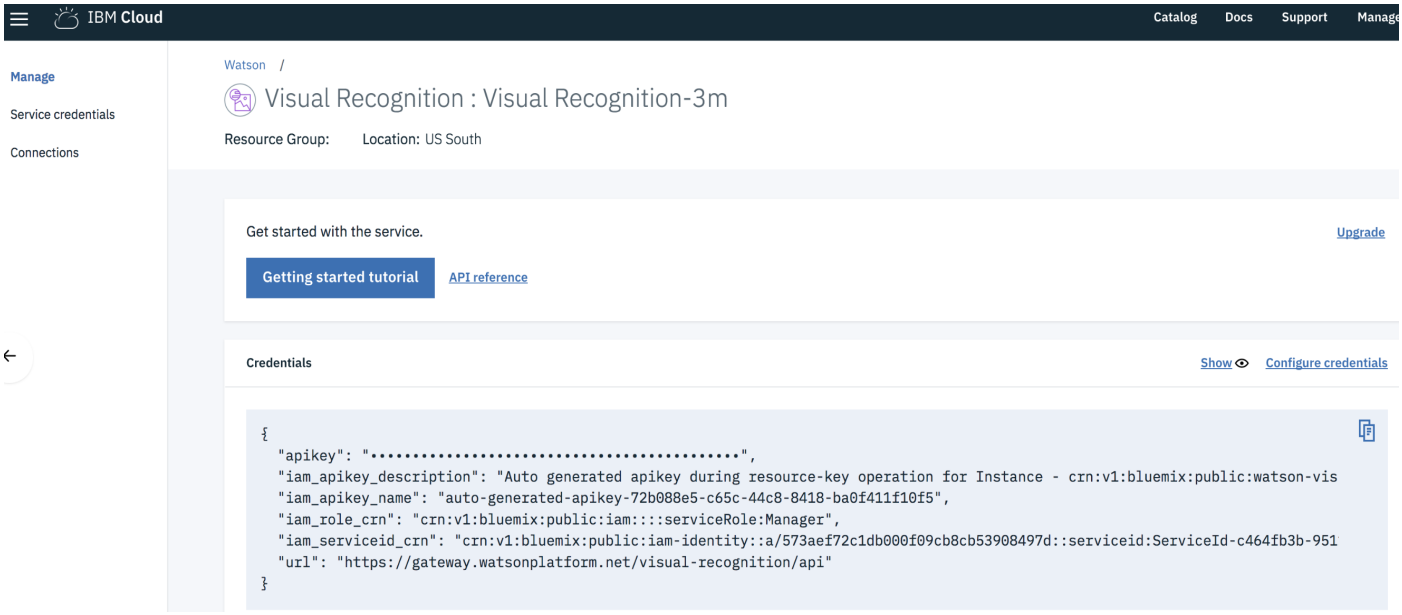
g) After the service is created you will see the **Manage** page of the service.

Click the **Service Credentials** to display them



IBM

h) In the list of credentials select the Action **View credentials** to display the “Credentials-1”.

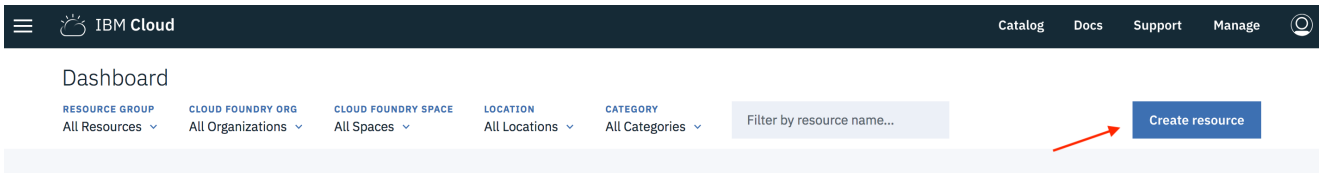


i) Copy the API Key of the service for later use into a file on your workstation or you can always go to the IBM Cloud console to copy the credentials from your list of services when you need them.

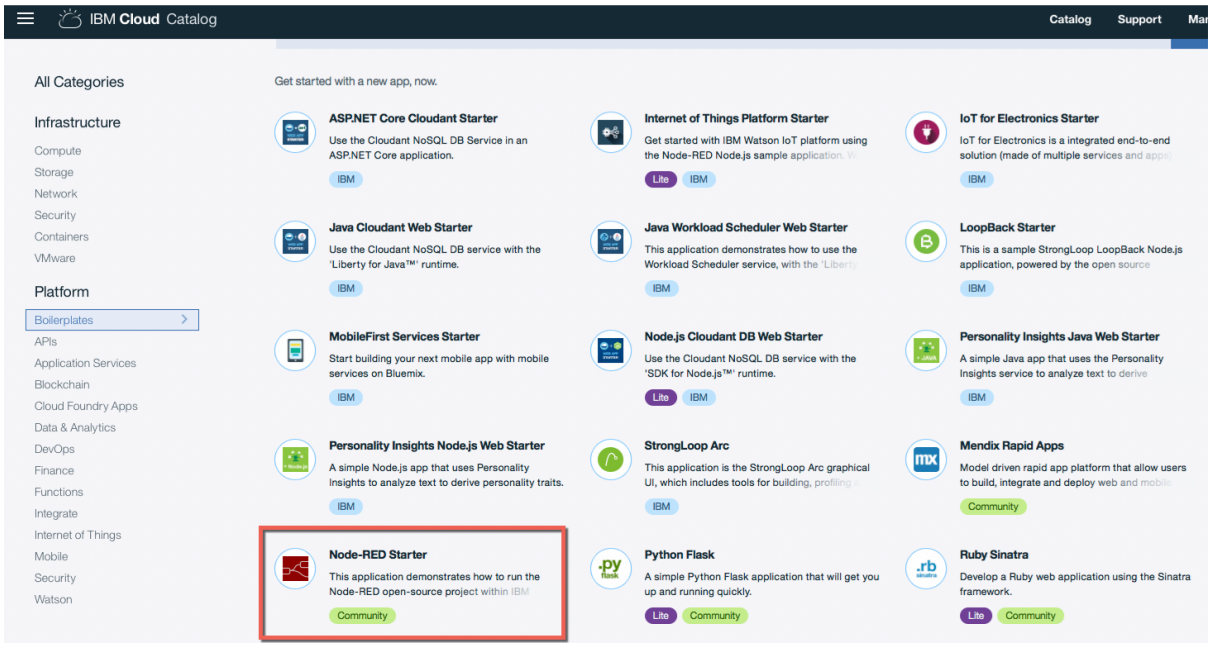
IBM

Section 2: Creating a Node-RED Visual Recognition Flow on IBM Cloud

Step 1 Log in to your IBM Cloud console and create a new application.



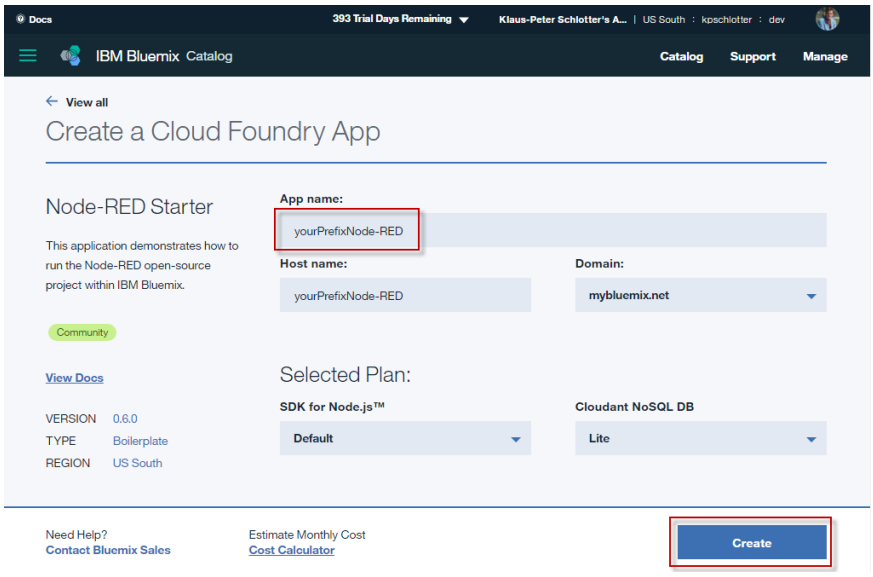
Step 2 Under the Apps → Boilerplates category **click** the **Node-RED Starter**



IBM

Step 3

Give your App a unique name and click the **Create** button.



It may take a while until a Node-RED environment is instantiated and started in your IBM Cloud environment

Step 4 In your App dashboard click the link for the running Node-RED application. You will first see the Welcome screen that lists some configuration steps. Click **Next**.

Step 5 You can secure your Node-RED editor by specifying a user id and password. Select the options that best suit your needs and click **Next**.

Step 6 On the *Browse Available IBM Cloud nodes* just click **Next**.

IBM

Step 7 On the *Finish the Install* screen click **Finish**.

Your Node-RED environment will be started.

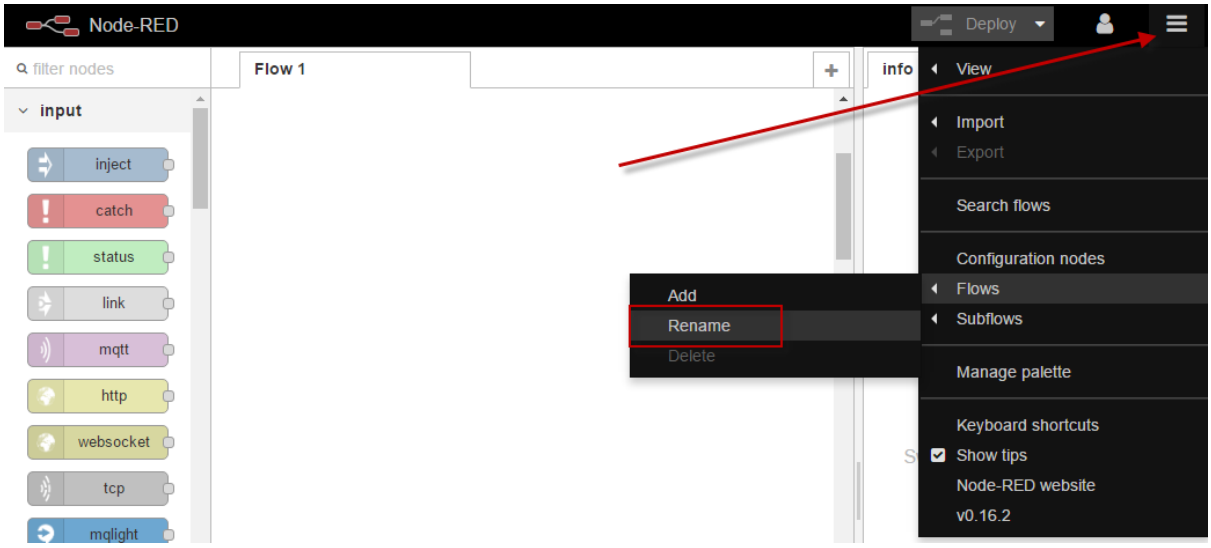
Step 8 On the Node-RED on IBM Cloud screen just click

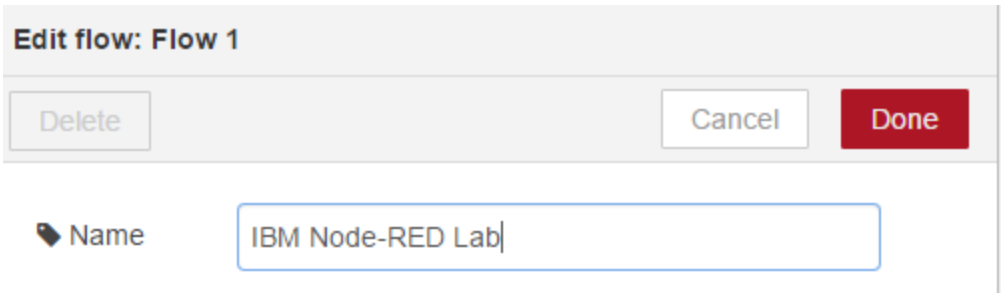
Go to your Node-RED flow editor.

Step 9 Eventually enter the username and password created in Step 5 and click **Login**.

Step 10 A Node-RED Flow editor opens with an empty panel named *Flow 1*.

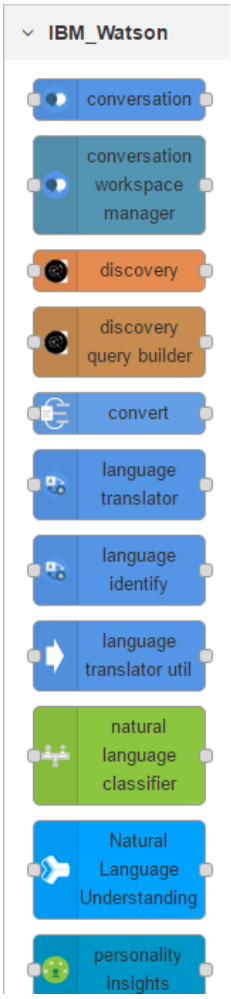
Step 11 Rename your Flow via the menu to IBM Node-RED Lab.



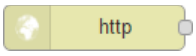


Step 12 Click **Done**.

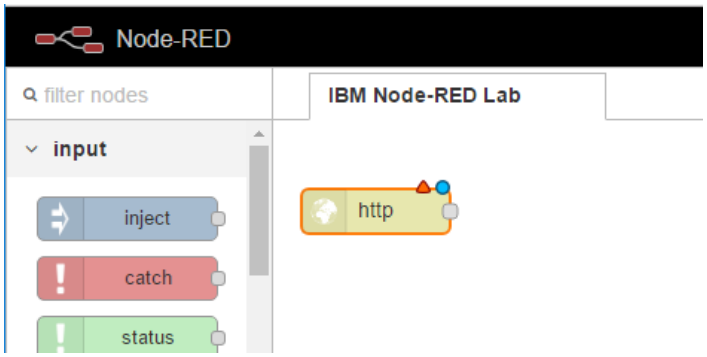
Step 13 In the left side bar of the Editor you see a lot of standard Node-RED nodes. At the end of the list there are certain categories already customized by IBM, such as **IBM_Watson**.



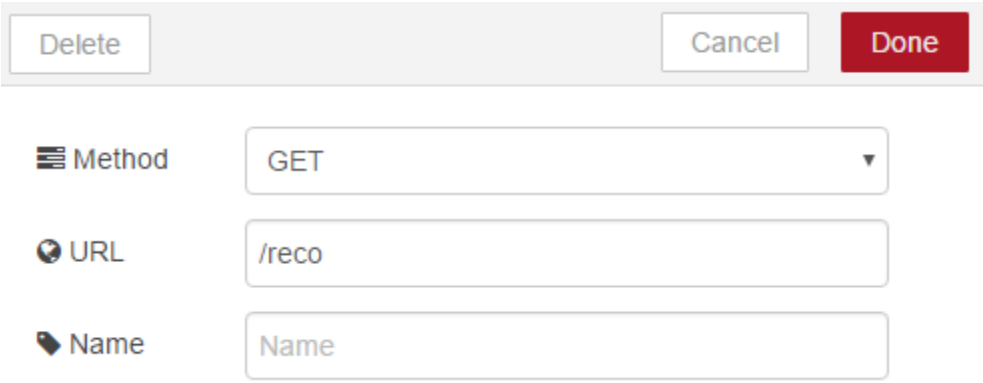
IBM

Step 14 From the input category of nodes drag the 

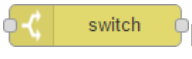
node and drop it on the editor pane. The Info pane on the right side bar give you some information about the node.



Step 15 Double click the **http** input node and specify the http **GET** request and an url of **/reco** and click **Done**.

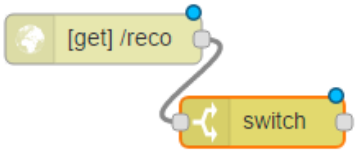


This is the URL appended to your Cloud Node-RED application to initialize the new flow **IBM Node-RED lab**.

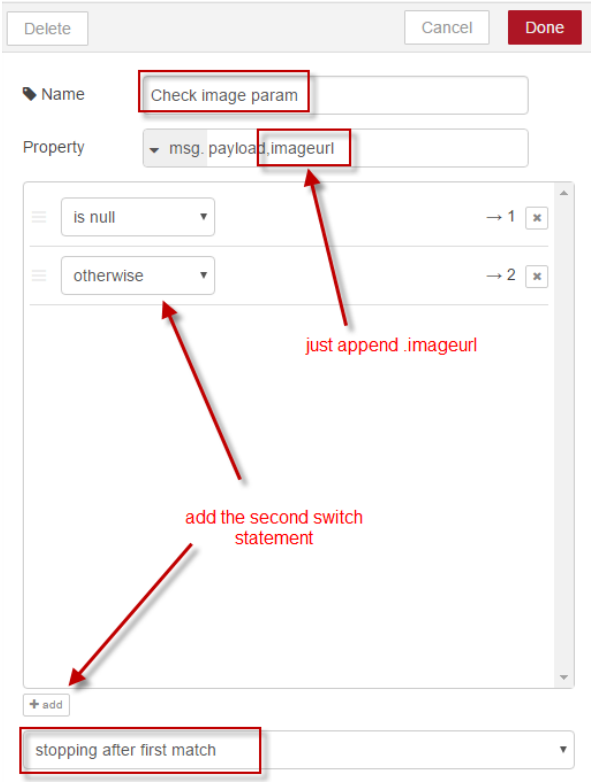
Step 16 In the nodes function section drag a 

node and drop it on the editor pane.

Step 17 Select the *output* of the /reco node and drop it on the *input* of the **switch** node.



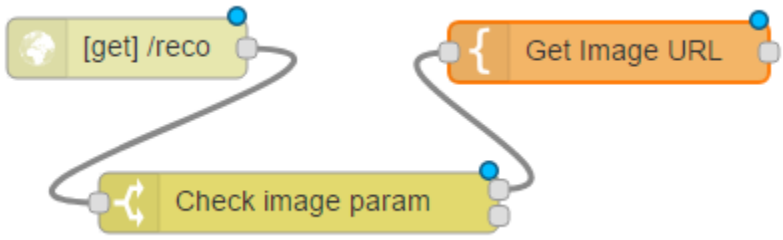
Step 18 Double click the **switch** node and enter the info shown on the following image and click **Done**.



Step 19 From the *function* section of nodes drag a 

node and drop it on the editor pane and connect the “is null” **output** of the switch to the **input** of the template.

IBM



Step 20 Double click the **template** and specify the following information:

a) Name: Get Image URL

b) Leave property as msg.payload

c) Syntax Highlight: HTML

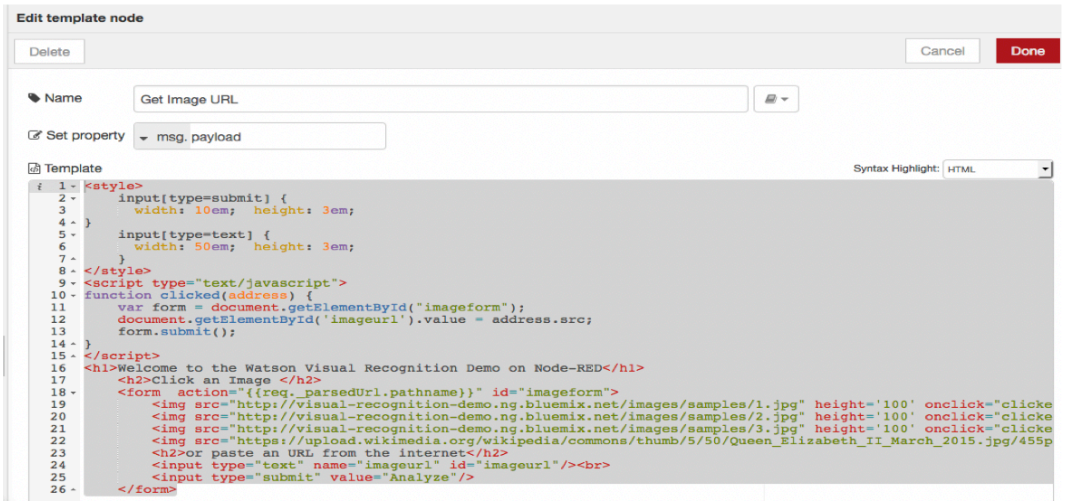
d) Format: Moustache template Template:

e) Template:

```
<style>
input[type=submit]{
    background-color: rgb(85,150,230);
    border: none;
    color: white;
    padding: 16px 32px;
    text-decoration: none;
    margin: 4px 2px;
    cursor: pointer;
    font-size: 15px;
    border-radius: 10px;
)
input[type=text] {
    width: 50em;  height: 3em;
}
</style>
<script type="text/javascript">
function clicked(address) {
```

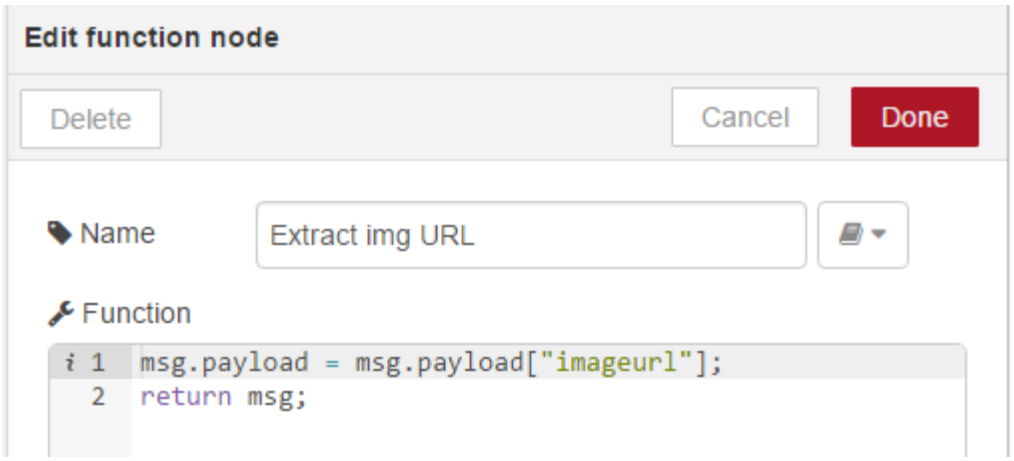
```
 var form = document.getElementById("imageform");
 document.getElementById('imageurl').value = address.src;
 form.submit();
}
</script>
<h1>Welcome to the Watson Visual Recognition Demo on Node-RED</h1>
 <h2>Click an Image </h2>
 <form  action="{{req._parsedUrl.pathname}}" id="imageform">
 <img src="http://visual-recognition-
demo.ng.bluemix.net/images/samples/1.jpg" height='100'
onclick="clicked(this)"/>
 <img src="http://visual-recognition-
demo.ng.bluemix.net/images/samples/2.jpg" height='100'
onclick="clicked(this)"/>
 <img src="http://visual-recognition-
demo.ng.bluemix.net/images/samples/3.jpg" height='100'
onclick="clicked(this)"/>
 <img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/5/50/Queen_
Elizabeth_II_March_2015.jpg/4
55px-Queen_Elizabeth_II_March_2015.jpg" height='100'
onclick="clicked(this)"/>
 <h2>or paste an URL from the internet</h2>
        <input type="text" name="imageurl" id="imageurl"/><br>
        <input type="submit" value="Analyze"/>
</form>
```

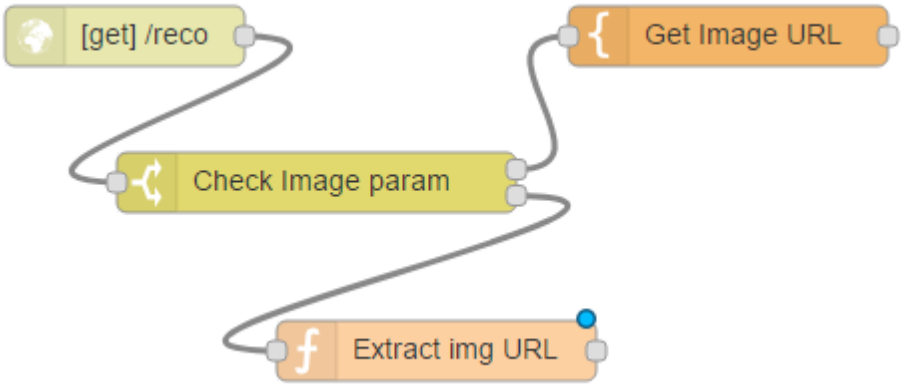
f) Click **Done**.

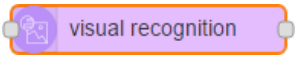


Step 21 Add a 

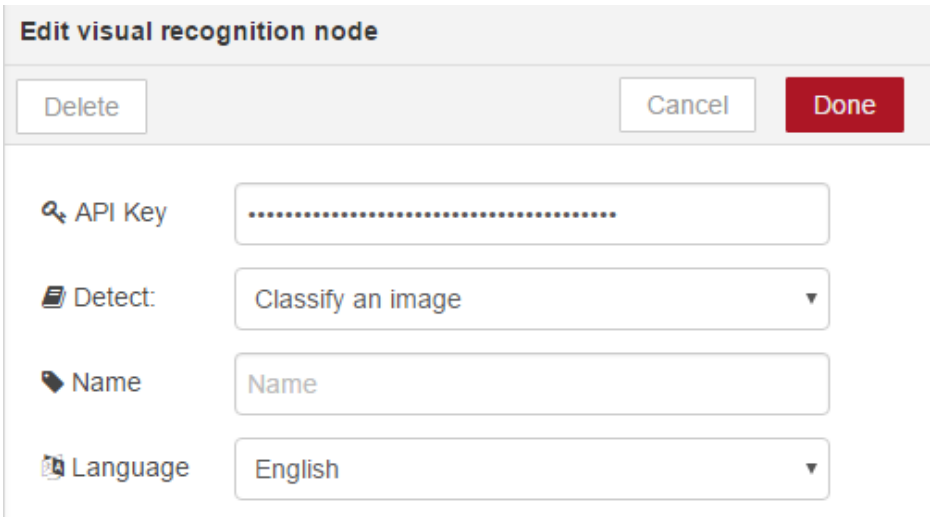
node (named *Extract img URL* here) to convert the *imageurl* JSON object to a string and assign it to the payload to be provided as input to the Visual Recognition node:

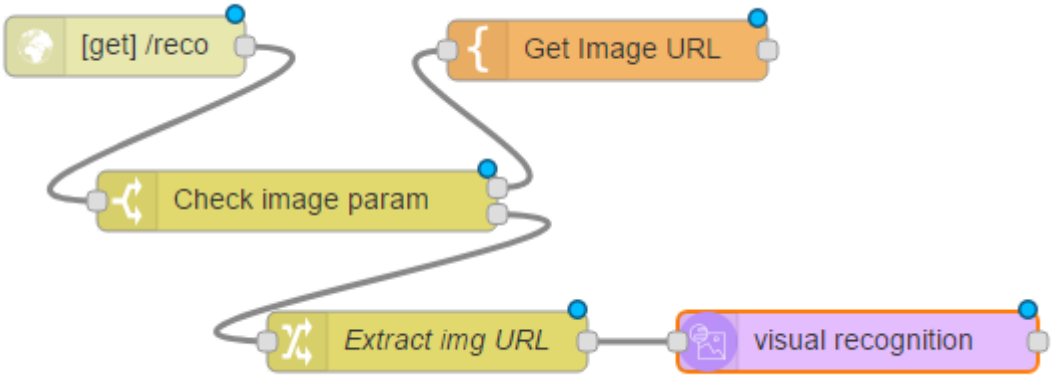




Step 22 From the **IBM_Watson** category select the 

node and drop it behind the change node from previous step. Double click the **Visual Recognition** node and enter your **API key** you have created for the Visual Recognition service in Lab 01. Click **Done**.





Step 23 Add another 

behind the Visual Recognition node. Double Click the Template Node to Edit with fields mentioned below

a) Name: Report

b) property: message.payload

c) Syntax Highlight: mustache

d) Format: Mustache template

e) Template

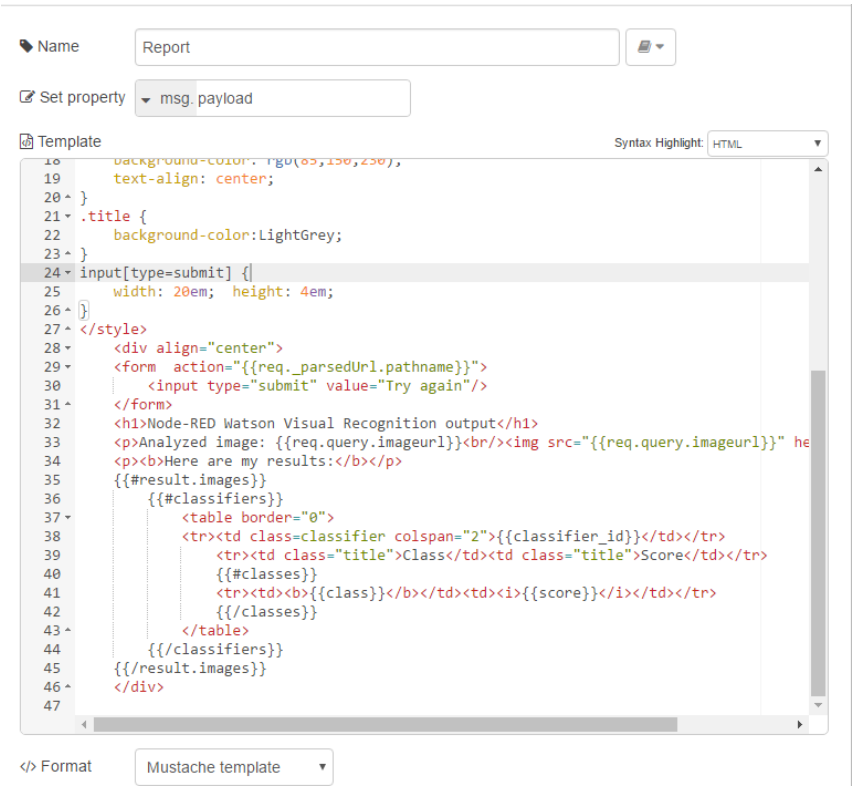
```
<div align="center">
    <form  action="{{req._parsedUrl.pathname}}">
        <input type="submit" value="Try again"/>
    </form>
<h1>Node-RED Watson Visual Recognition output</h1> <p>Analyzed image:
{{req.query.imageurl}}<br/><img src="{{req.query.imageurl}}" height='200'/></p>
<p><b>Here are my results:</b></p> {{#result.images}}

        {{#classifiers}}
            <table border="0">
            <tr><td class=classifier
colspan="2">{{classifier_id}}</td></tr>
                <tr><td class="title">Class</td><td
class="title">Score</td></tr>
                {{#classes}}

<tr><td><b>{{class}}</b></td><td><i>{{score}}</i></td></tr>
                {{/classes}}
            </table>
        {{/classifiers}}
    {{/result.images}}
</div>
```

Click **Done**.

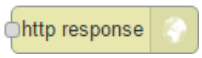
f) Optional Styling at the top of the template

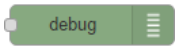
```
<style> h4 {

    text-align: center;
    margin: 10px;
}
table {
    width: 480px;
    margin-top: 10px;
}
th, td {
    padding: 8px;
```

```
    text-align: left;
    border-bottom: 1px solid #ddd;
    background-color: #FFFFFF;
    width: 50%;
}
.classifier {
    background-color: rgb(85,150,230);
    text-align: center;
}
.title {
    background-color:LightGrey;
}
input[type=submit]{
    background-color: rgb(85,150,230);
    border: none;
    color: white;
    padding: 16px 32px;
    text-decoration: none;
    margin: 4px 2px;
    cursor: pointer;
    font-size: 15px;
    border-radius: 10px;
) </style>
```

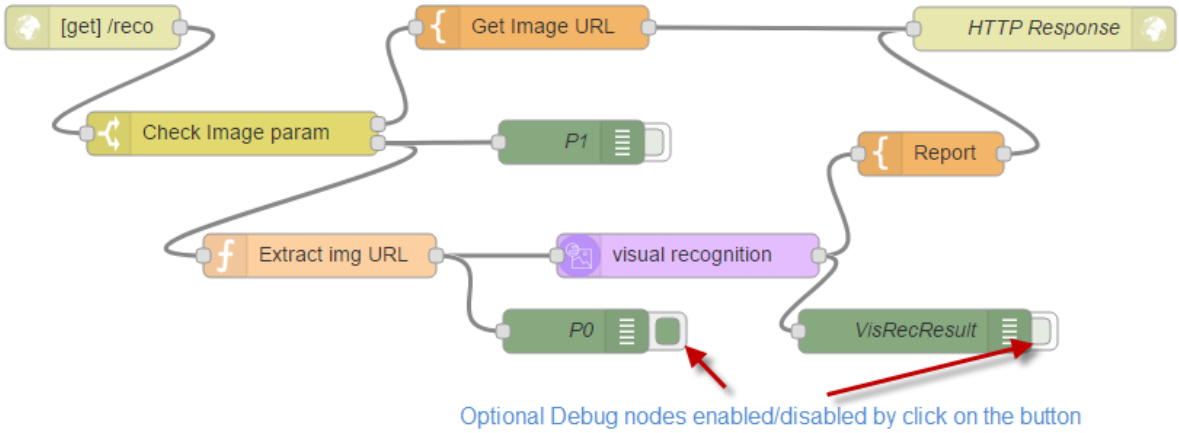


IBM

Step 24 As the final node add a 

, name it *HTTP Response,* and connect it as shown in the following picture. Optional 

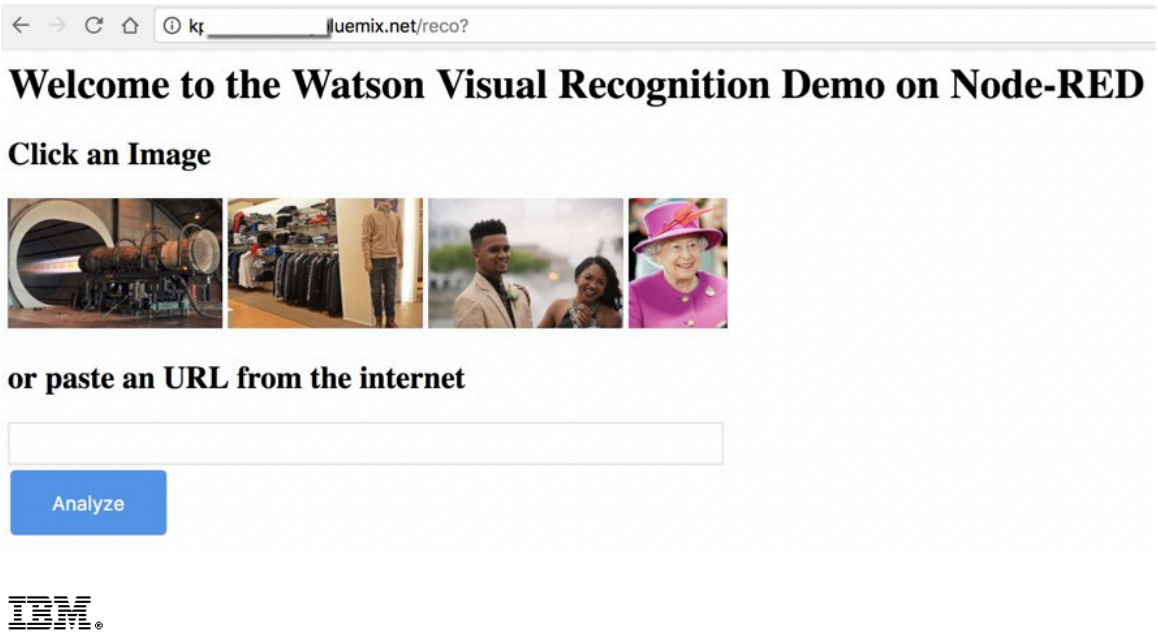
nodes display the output of the node in the right side bar on the debug tab.



Step 25 Click the 

button at the top of the Node-RED application to de- ploy the flow in your environment.

Step 26 Test the flow by calling the url http://<yourNode-REDHost>.mybluemix.net/reco



Step 27 **Click** on of the images or **Copy and Paste** any image URL from the internet and click **Analyse**

Try again

Node-RED Watson Visual Recognition output

Analyzed image: https://upload.wikimedia.org/wikipedia/commons/thumb/5/50/Queen_Elizabeth_II_March_2015.jpg/455px-Queen_Elizabeth_II_March_2015.jpg

Here are my results:

default	
Class	Score
queen	*0.908*
person	*0.976*
Queen of England	*0.706*
queen mother	*0.5*
Tyrian purple color	*0.81*
fuschia color	*0.788*

This completes Your Lab!

IBM