

IBM Bluemix Node-RED Watson Starter

Cognitive Solutions Application Development

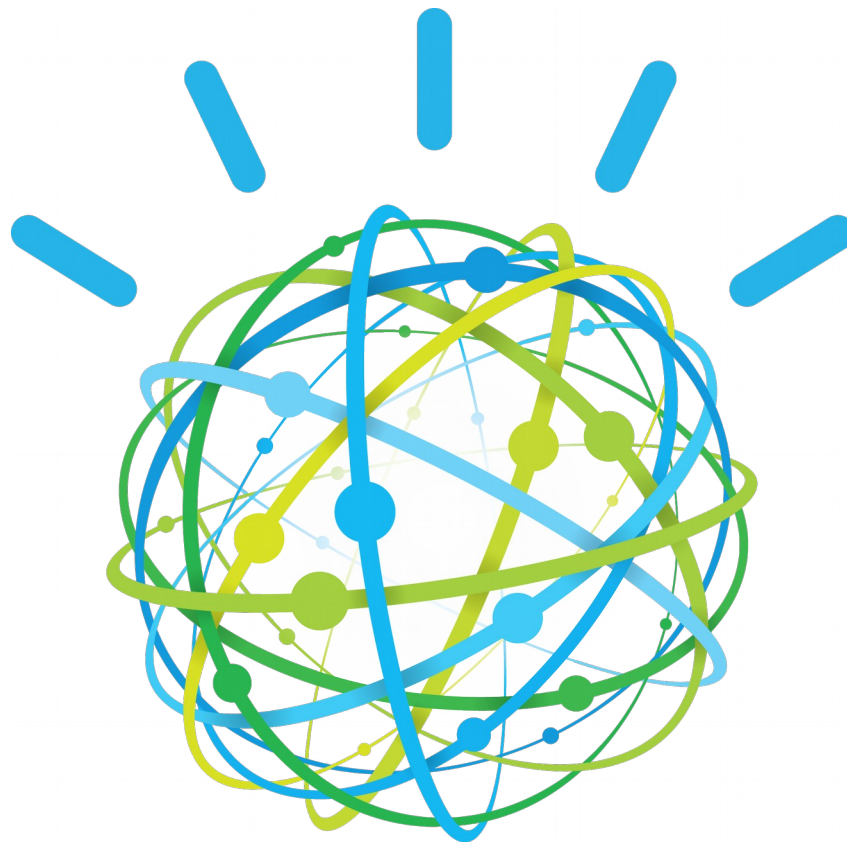
IBM Global Business Partners

Duration: 45 minutes

Updated: Nov 28, 2018

Klaus-Peter Schlotter

kps@de.ibm.com



IBM

Version 1.5

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.

The Node-Red Watson Starter 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

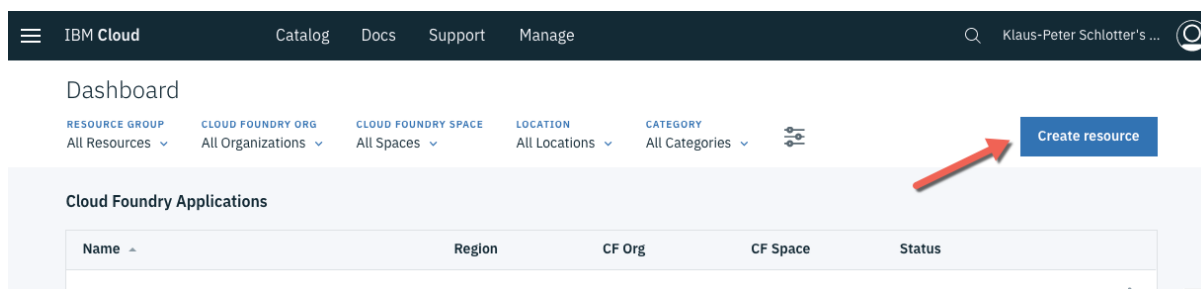
- Learn how to use the Node-Red environment on IBM Cloud
- Learn how to implement the Image-Analysis application using Node-Red

Prerequisites

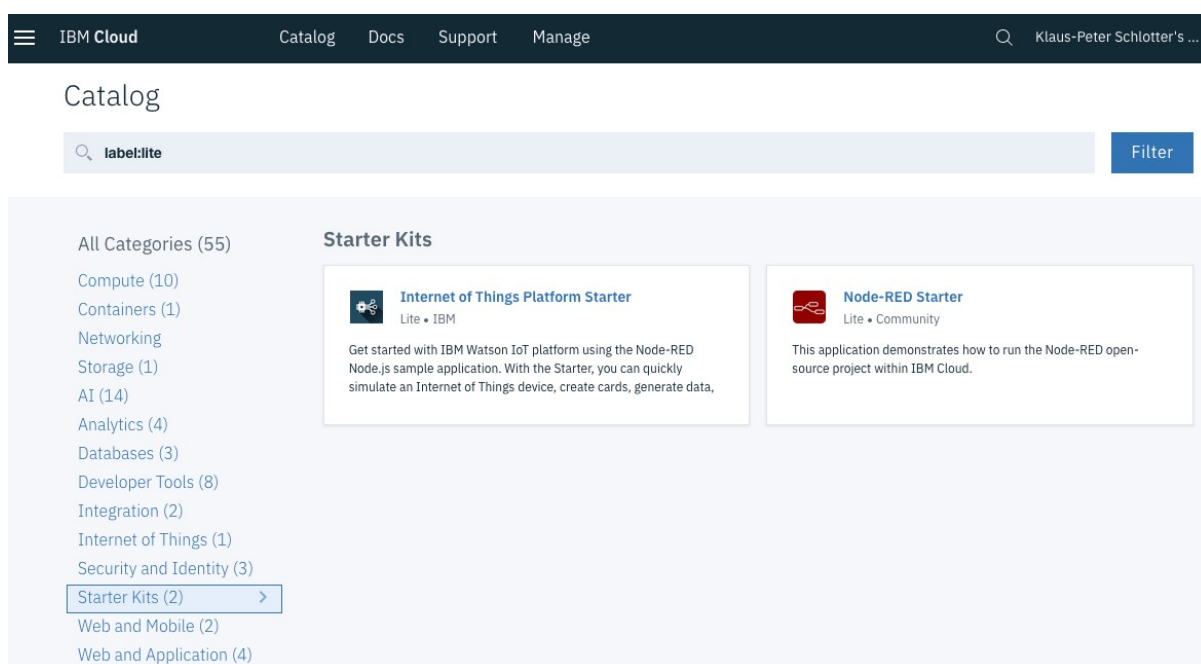
The prerequisite for this lab is an IBM Cloud account and a Watson Visual Recognition service created in an earlier lab (**Lab 1**), or create a Visual Recognition Service in the IBM Cloud console and retrieve the key from there (Service Credentials).

Section 1: Creating a Node-RED Visual Recognition Flow on IBM Cloud

Step 1 Log in to your IBM Cloud console and click [Create resource](#).



Step 2 Under the *Starter Kits* category click the **Node-RED Starter**



Step 3 Give your App a unique name and click the **Create** button.

The screenshot shows the 'Create a Cloud Foundry App' interface for 'Node-RED Starter'. The form includes the following fields and options:

- App name:** A text input field containing 'kp-1234567890'.
- Host name:** A text input field containing 'kp-1234567890'.
- Domain:** A dropdown menu showing 'eu-gb.mybluemix.net'.
- Choose a region/location to deploy in:** A dropdown menu showing 'London'.
- Choose an organization:** A dropdown menu showing 'kp-1234567890'.
- Choose a space:** A dropdown menu showing 'dev_gb'.
- Selected Plan:** A section with two dropdowns: 'SDK for Node.js™' set to 'Lite' and 'Cloudant' set to 'Lite'.
- SDK for Node.js™:** A section with a '.js' icon and the text 'SDK for Node.js™'.
- Cloudant:** A section with a Cloudant icon and the text 'Cloudant'.
- Pricing Plans:** A section with the text 'Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides enhanced performance, security, and serviceability.' and a link to 'Monthly prices shown are for country or region: Germany'.
- Need Help?** A link to 'Contact IBM Cloud Support'.
- Estimate Monthly Cost:** A link to 'Cost Calculator'.
- Create:** A blue button to submit the form.

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**.

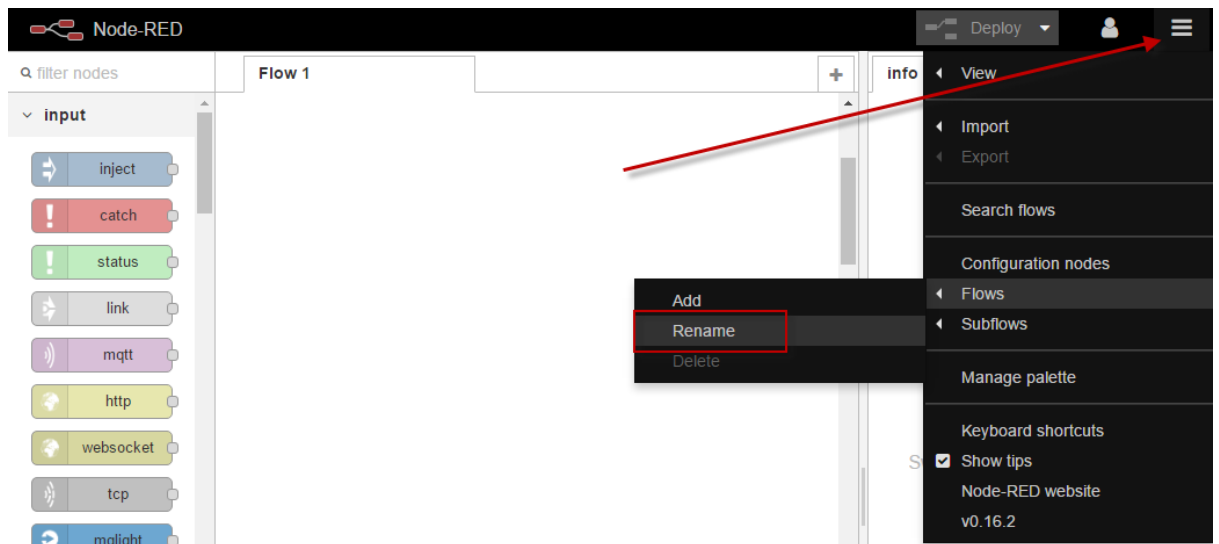
Step 6 On the *Finish the Install* screen click **Finish**.
Your Node-RED environment will be started.

Step 7 On the Node-RED on IBM Bluemix screen just click **Go to your Node-RED flow editor**.

Step 8 Eventually enter the username and password created in Step 5 and click **Login**.

Step 9 A Node-RED Flow editor opens with an empty panel named *Flow 1*.

Step 10 Rename your Flow via the menu f.e. IBM Node-RED Lab.

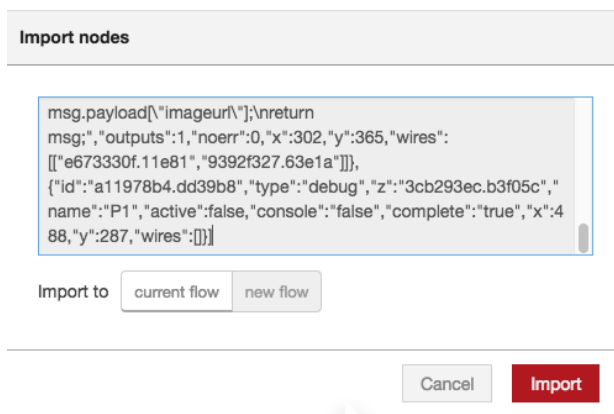


Step 11 Click **Done**.

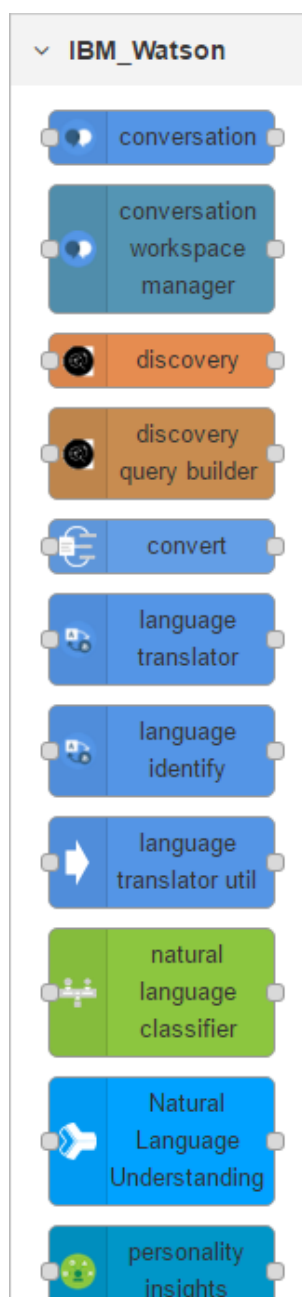



Step 12 (Optional:) If you have problems creating the flow described in the following steps, you can also import the flow from [here](#).

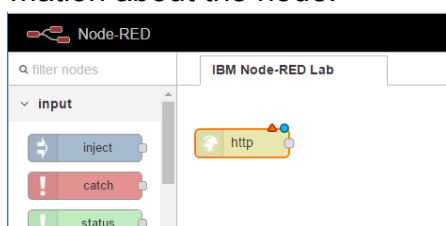
- Open the file in a text editor.
- Copy all content to the clipboard
- From the Node-Red editor menu select **Import** → **Clipboard**, paste the code, select **import to current flow** and click **Import**.



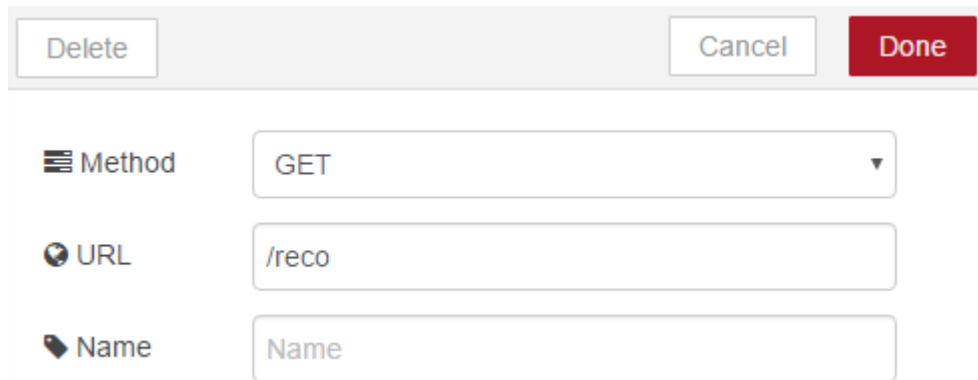
Step 13 In the left side bar of the Editor you see a lot of standard Node-RED nodes. At the end of the list there are certain categories already customized by IBM, such as **IBM_Watson**.



Step 14 From the input category of nodes drag the  **http** node and drop it on the editor pane. The Info pane on the right side bar give you some information about the node.

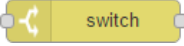


Step 15 Double click the **http** input node and specify the http **GET** request and an url of **/reco** and click **Done**.

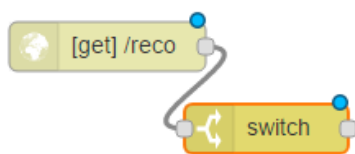


A screenshot of the Node-RED configuration dialog for an HTTP node. At the top are three buttons: 'Delete', 'Cancel', and 'Done'. Below are three input fields: 'Method' with a dropdown menu showing 'GET', 'URL' with the text '/reco', and 'Name' with the text 'Name'.

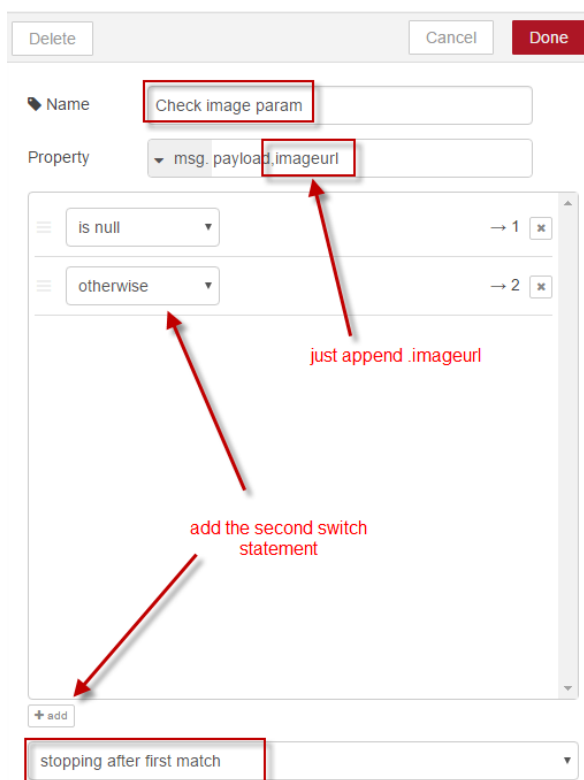
This is the URL appended to your Bluemix Node-RED application to initialize the new flow **IBM Node-RED lab**.

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.

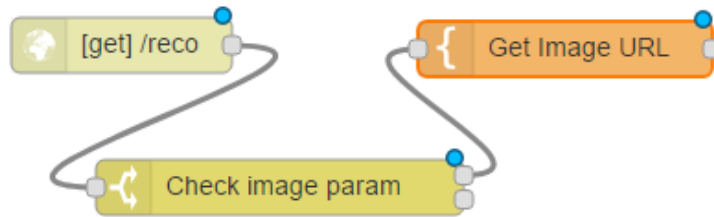


Step 18 Double click the **switch** node and enter the info shown on the following image and click **Done**.



A screenshot of the Node-RED switch node configuration dialog. At the top are three buttons: 'Delete', 'Cancel', and 'Done'. Below are two input fields: 'Name' with the text 'Check image param' and 'Property' with a dropdown menu showing 'msg.payload.imageUrl'. Below these are two rows of configuration: 'is null' with a dropdown menu and 'otherwise' with a dropdown menu. To the right of these rows are two buttons: '→ 1' and '→ 2'. Below these are two red arrows pointing to the 'otherwise' dropdown and the '→ 2' button. The first arrow is labeled 'just append .imageUrl' and the second arrow is labeled 'add the second switch statement'. At the bottom is a button labeled '+ add' and a dropdown menu showing 'stopping after first match'.

Step 19 From the *function* section of nodes drag a **template** node and drop it on the editor pane and connect the “is null” **output** of the switch to the **input** of the template.

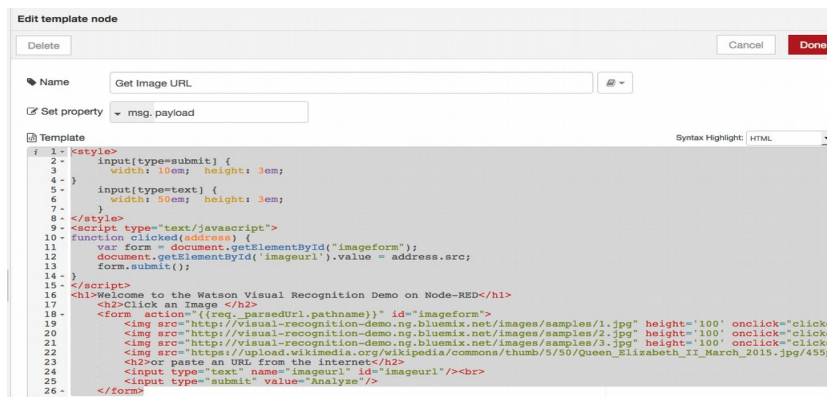



Step 20 Double click the **template** and specify the following information:

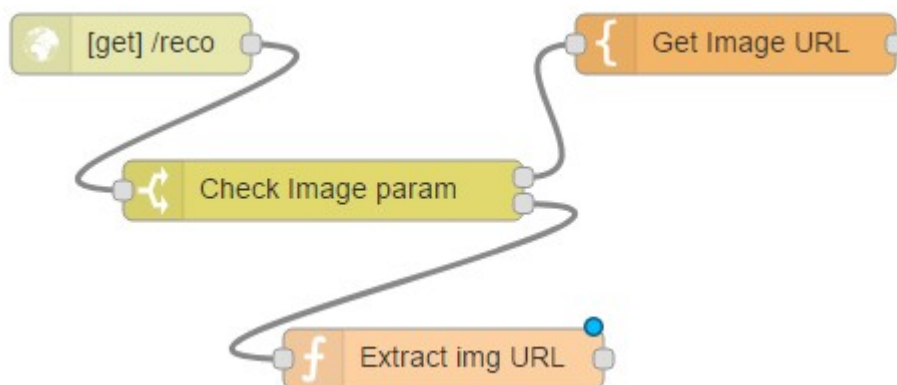
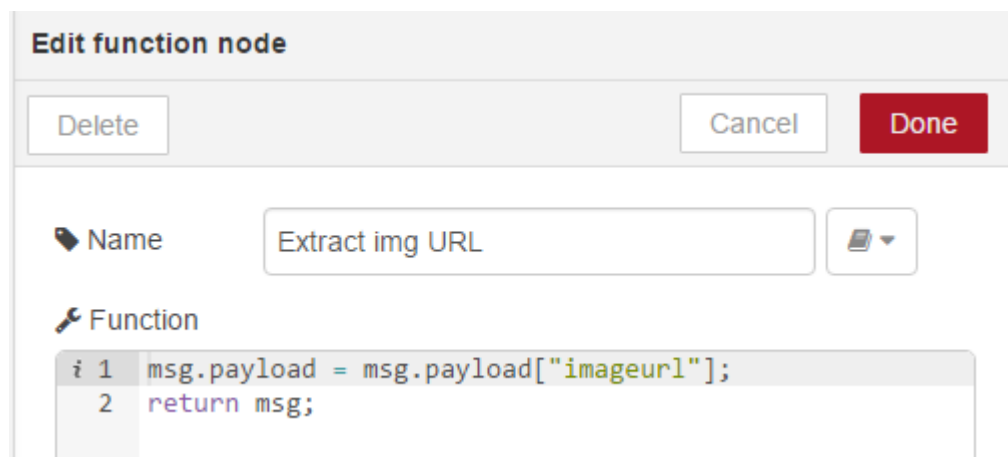
- Name: Get Image URL
- Leave property as msg.payload
- Syntax Highlight: HTML
- Format: Moustache template
- Template:

```
<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>
<div align="center">
  <h1>Welcome to the Watson Visual Recognition Demo on Node-RED</h1>
  <h2>Click an Image </h2>
  <form action="{{req.parsedUrl.pathname}}" id="imageform">
    
    
    
    
    <h2>or paste an URL from the internet</h2>
    <input type="text" name="imageurl" id="imageurl"/><br>
    <input type="submit" value="Analyze"/>
  </form>
</div>
```


f) Click **Done**.



Step 21 Add a  **function** 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 **visual recognition** 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**.

Edit visual recognition node

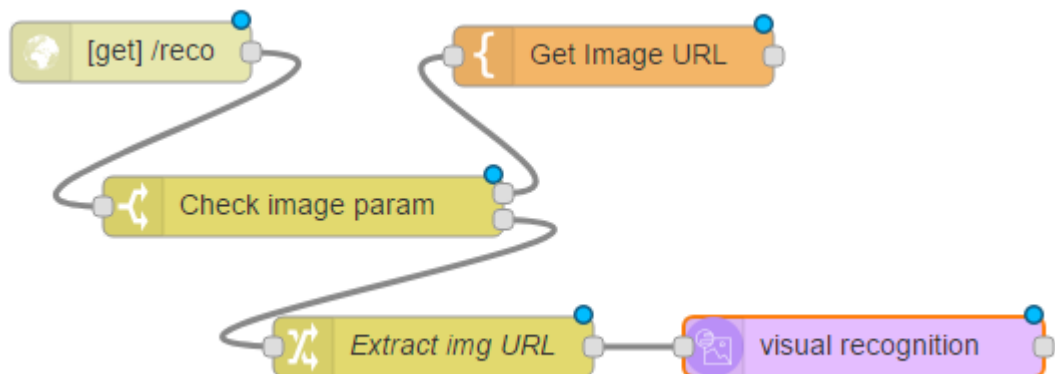
Delete
Cancel
Done


API Key
.....

Detect:
Classify an image

Name
Name

Language
English



Step 23 Add another  behind the Visual Recognition node. Click **Done**.

- 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/></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>
```

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>
```

Name

☒ Set property

Template
Syntax Highlight: HTML

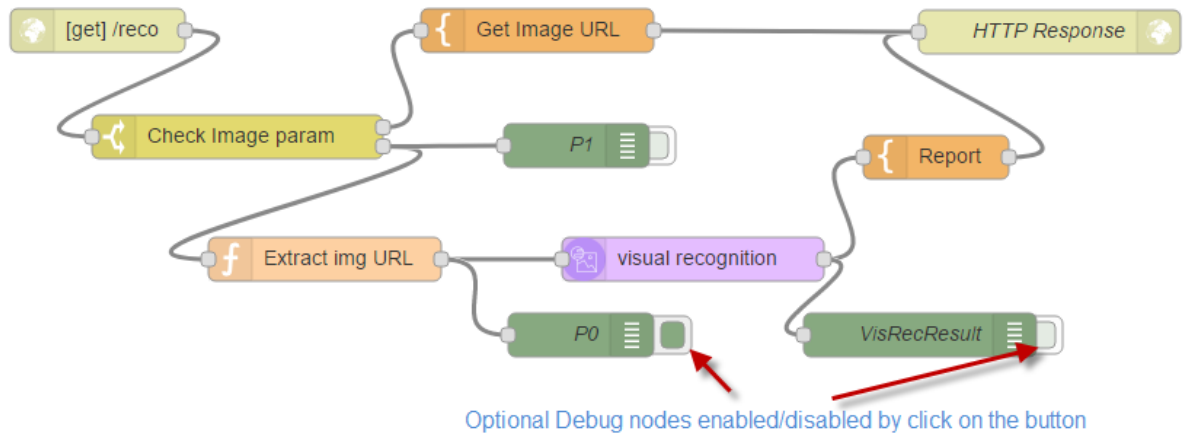
```

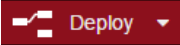
18 background-color: rgb(85,150,230);
19 text-align: center;
20 }
21 .title {
22 background-color: LightGrey;
23 }
24 input[type=submit] {
25 width: 20em; height: 4em;
26 }
27 </style>
28 <div align="center">
29 <form action="{{req._parsedUrl.pathname}}">
30 <input type="submit" value="Try again"/>
31 </form>
32 <h1>Node-RED Watson Visual Recognition output</h1>
33 <p>Analyzed image: {{req.query.imageUrl}}<br/><b>Here are my results:</b></p>
35 {{#result.images}}
36 {{#classifiers}}
37 <table border="0">
38 <tr><td class=classifier colspan="2">{{classifier_id}}</td></tr>
39 <tr><td class="title">Class</td><td class="title">Score</td></tr>
40 {{#classes}}
41 <tr><td><b>{{class}}</b></td><td><i>{{score}}</i></td></tr>
42 {{/classes}}
43 </table>
44 {{/classifiers}}
45 {{/result.images}}
46 </div>
47

```

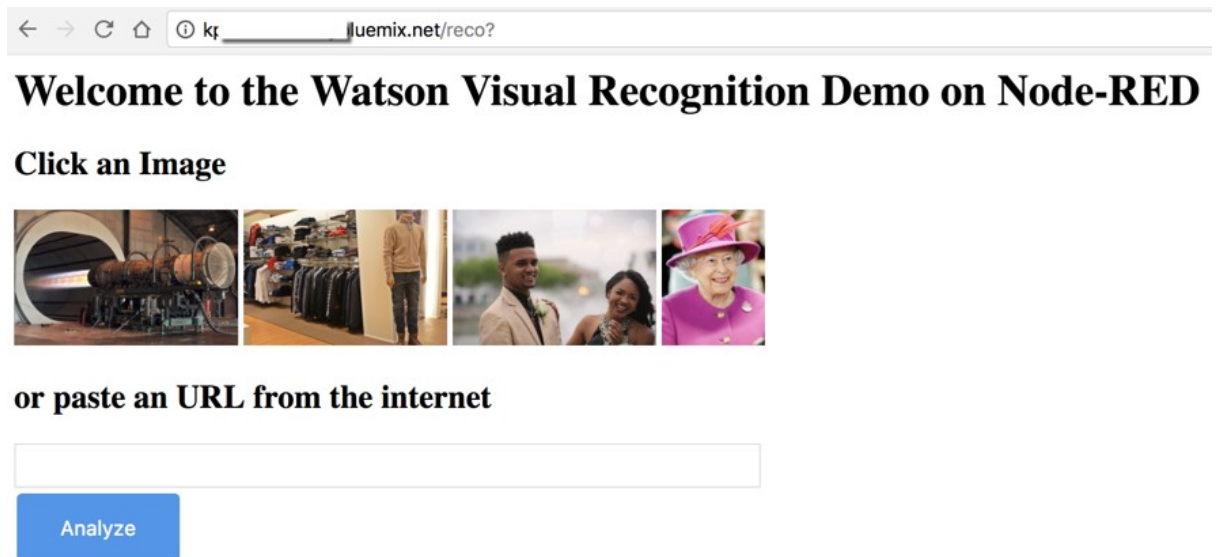
Format
Mustache template

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 deploy 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 **Analyze**

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 Lab 02 of the IBM Watson Services Workshop.