

Classify an Image using Watson VR through Node-Red Editor – 20mins Lab

Rajesh K Jeyapaul , jrkmur@in.ibm.com

Pre-req: IBM cloud access (<https://bluemix.net>)

Flow of the Lab:

Step 1: Deploy a Boilerplate which has node-red editor (templated nodejs) as service

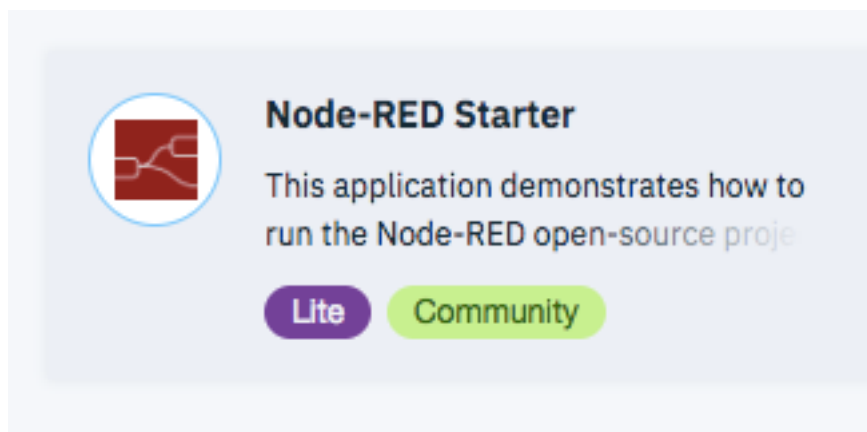
Step 2: Deploy Watson Visual Recognition Service

Step 3: Connect the Watson VR with the deployed Boilerplate

Step 4: Use NodeRed Editor to invoke Watson VR service

Step 1: Deploy a Boilerplate which has node-red editor as service

Go to catalog and deploy a nodered Boilerplate:



Provide the Appname, an unique name should be provided. In this case “BengaluruCFC”. Pls. do not use this App name but provide an unique one.

App name:		
<input type="text" value="BengaluruCFC"/>		
Host name:	Domain:	
<input type="text" value="BengaluruCFC"/>	<input type="text" value="mybluemix.net"/>	
Choose a region/location to deploy in:	Choose an organization:	Choose a space:
<input type="text" value="US South"/>	<input type="text" value="jrkmur@in.ibm.com"/>	<input type="text" value="dev"/>
Selected Plan:		
SDK for Node.js™	Cloudant	
<input type="text" value="Default"/>	<input type="text" value="Lite"/>	

Wait for couple of minutes for the app to get deployed

Cloud Foundry apps /



BengaluruCFC



Starting

[Visit App URL](#)

Cloud Foundry apps /



BengaluruCFC




Running

[Visit App URL](#)


Step 2: Deploy Watson Visual Recognition Service

Go back to Catalog and select the visual recognition service to be deployed and linked (connected) back to the just created application

 **visual re**

Platform

Watson
Build cognitive apps that help enhance, scale, and accelerate human expertise.

**Visual Recognition**
Find meaning in visual content! Analyze images for scenes, objects, faces, and other content

Lite **IBM**

Create

Step 3: Connect the Watson VR with the deployed Boilerplate

Connect it with the just created application

Plan

Resource Group: default Location: US South

Connections

Create connection


Select your app name, in this case "BengaluruCFC"

Connections

US South jrkumar@in.ibm.com dev Filter by resource name...

CONNECTION LOCATION

dev jrkumar@in.ibm.com // us-south

CLOUD FOUNDRY APPS	STATUS	
 BengaluruCFC	0/1 Not running	CONNECT

Restage app

Your 'BengaluruCFC' app needs to be restaged. Do you want to restage it now?

Cancel

Restage

Step 4: Use NodeRed Editor to invoke Watson VR service

Go back to dashboard and get prepared to launch the application



Cloud Foundry



Containers



Infrastructure



VMware



Dashboard



APIs



Apple Development New



Blockchain

Ensure that the application deployed is in running state.

BengaluruCFC	US South	jrkumar@in.ibm.cor	dev	256	Running	
--------------	----------	--------------------	-----	-----	----------------------	--

Run the application by selecting the app URL

[Visit App URL](#)

At this stage, the node red framework is available to further use the Watson VR service.

Welcome to your new Node-RED instance on IBM Cloud

We know you're eager to start wiring up your flows, but first there are a couple of tasks you should do:

- Secure your Node-RED editor
- Browse available IBM Cloud nodes



Previous

Next

For the current lab requirement , you can skip the security aspect. Continue as shown below:

- *Not recommended:* Allow anyone to access the editor and make changes

Your editor will not be secured. Anyone with the URL will be able to access your flows, data and bound services.

☒ Tick this box to confirm you want your editor to be insecure



Take the default value for the rest of the fields and Finish

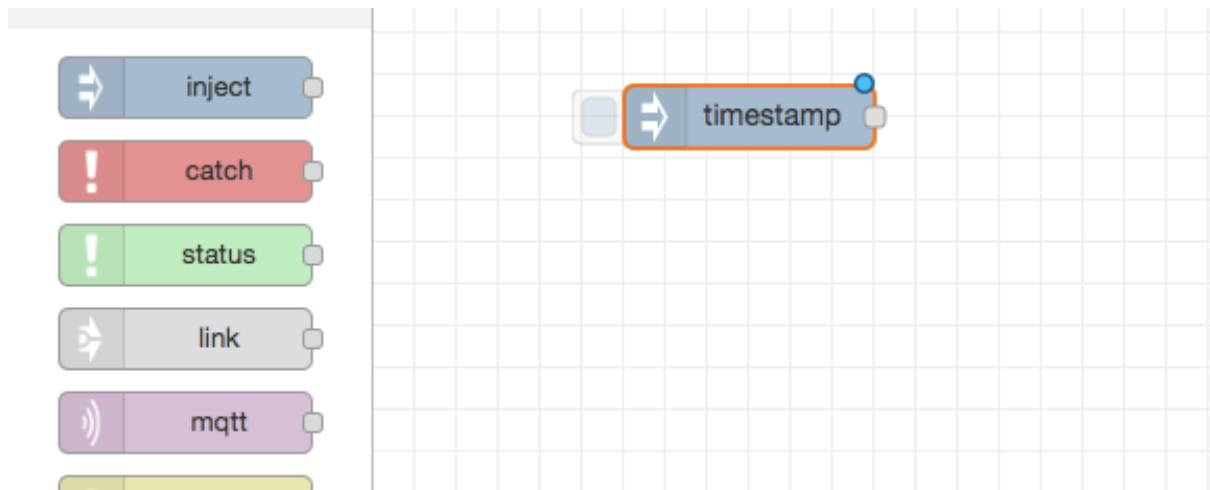


Proceed to Launch the NodeRed Editor:

Go to your Node-RED flow editor

Add three nodes in the Editor as shown:

- 1) Inject
- 2) Watson Visual Recognition
- 3) Debug node



Q visual

x

> input

> output

> function

> social

> storage

> analysis

> advanced

> weather

> Smarter Process

▼ IBM Watson

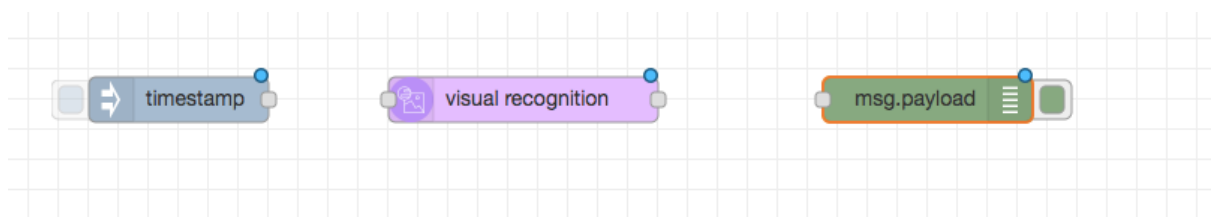
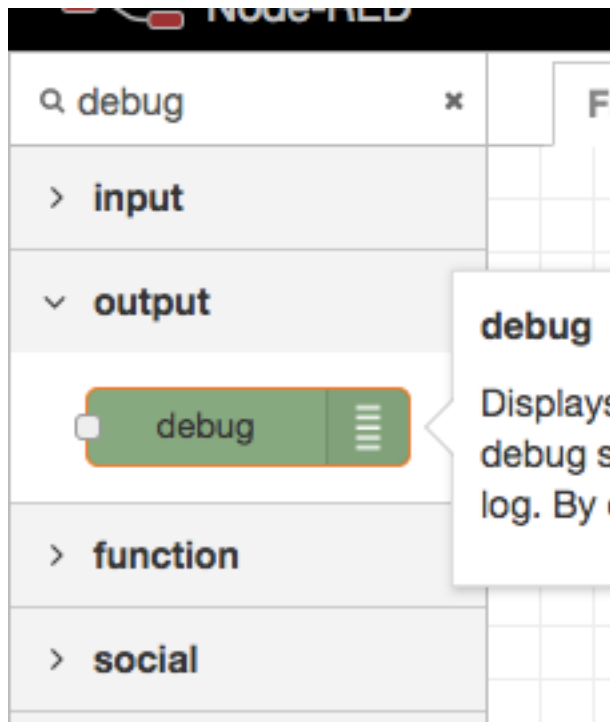
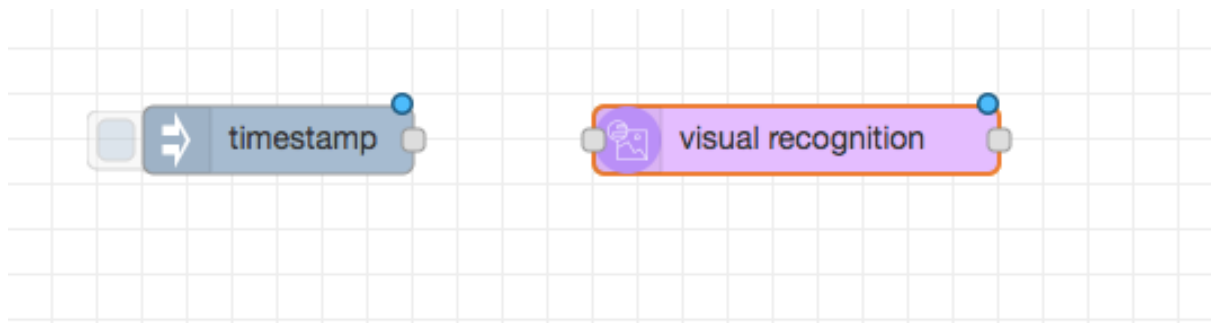
visual
recognition

visual
recognition
util

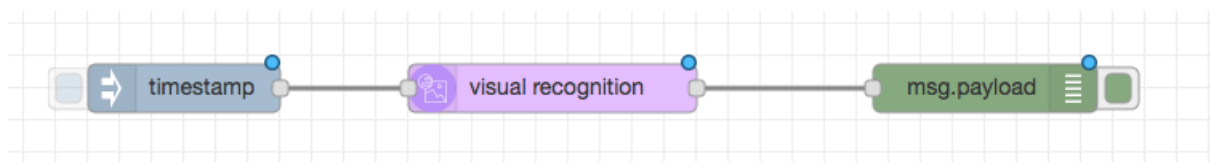
visua

visua

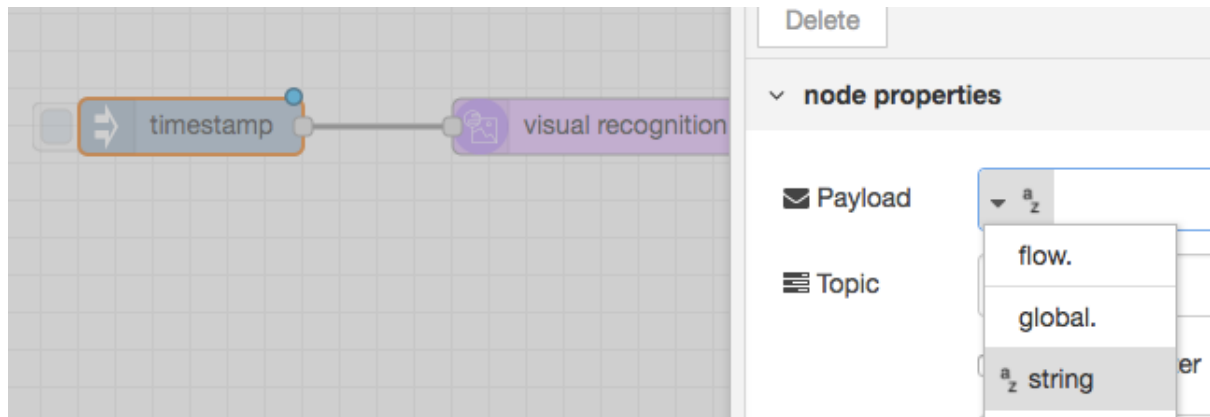
Using
use t
to ide
imag
creat
subje



Now, connect all the nodes and configure the nodes as shown:



Change from timestamp to Strings to take URL as input



Watson VR provides 3 types of image Detection: Text Identification, face recognition and classification. Change to Classification as shown:

Detect: Classify an image

Name Name

Language English

Change the Debug node to “Complete msg object”

node properties

Output complete msg object

To ☒ debug window

☐ system console

Name Name

Provide the input image URL in the Inject node as shown. With this you are ready to deploy the Watson VR and ready to classify the image:

Edit inject node

Delete

Cancel

Done

▼ node properties

✉ Payload

▼ a_z yes/68251000/jpg/_68251914_68251777.jpg

☰ Topic

☐ Inject once after

0.1

seconds, then

Deploy and see the result in the console:

Deploy ▼

☰

debug

ation

"2b9fce96.7ec782"

inject

show more ▼

elp

essage into a flow either manually or
tervals. The message payload can

all nodes

05/07/2018, 11:04:19 node: c513b47a.938808

msg : Object

```
▶ { _msgid: "12799915.5b5dc7",  
  topic: "", payload:  
    "https://ichef.bbci.co.uk/news/...",  
  result: object }
```