

***Cognitive Home using Bluemix Watson
IoT and Intel Edison kit***

-Rajesh K Jeyapaul, IBM

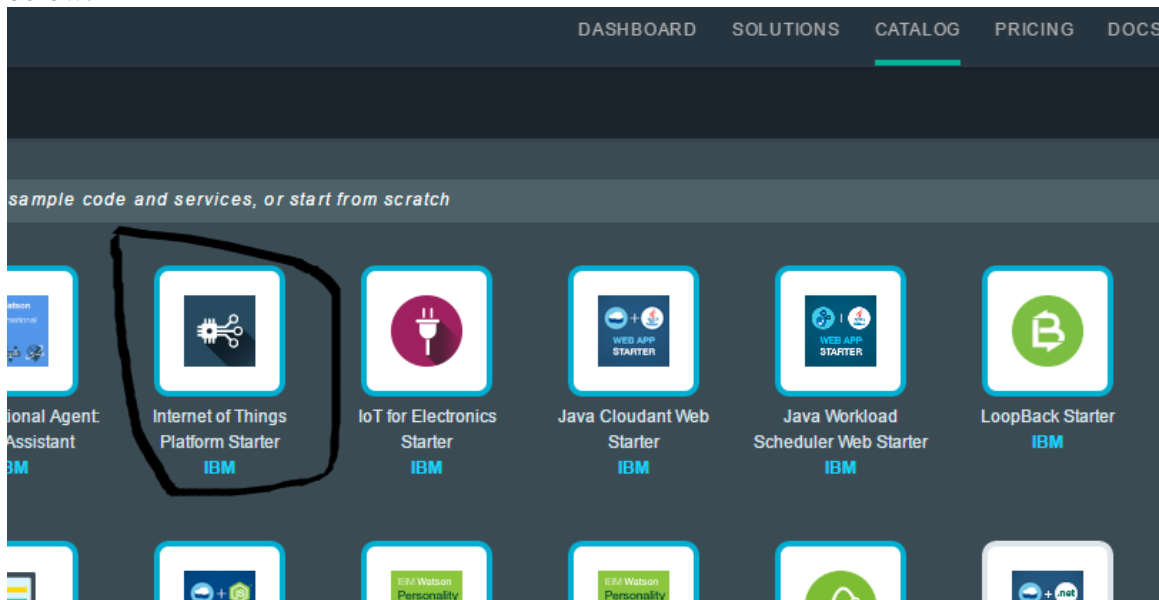
-Shubhadradeep IBM



Step 1: Intel Edison kit registration with IBM Bluemix


Deploy IoT Boilerplate

Go to Bluemix catalog page and select “Internet of Things Platform Starter” as shown below:




Provide an unique name for your application as shown:


DASHBOARD SOLUTIONS CATALOG PRICING DOCS COMMUNITY



SDK for Node.js™



Cloudant NoSQL DB



Internet of Things Platform

1 scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides nce, security, and serviceability.

[VIEW DOCS](#)

Monthly prices shown are for country or region: [India](#)

or more apps free for 30 days (375 GB-hours free). ₹4,2263 INR/GB-Hour

Create an app:

Space:

Name:

Host:

Domain:

Selected Plan:

SDK for Node.js™

Cloudant NoSQL DB

Procced to create

Internet of Things Platform

[CREATE](#)

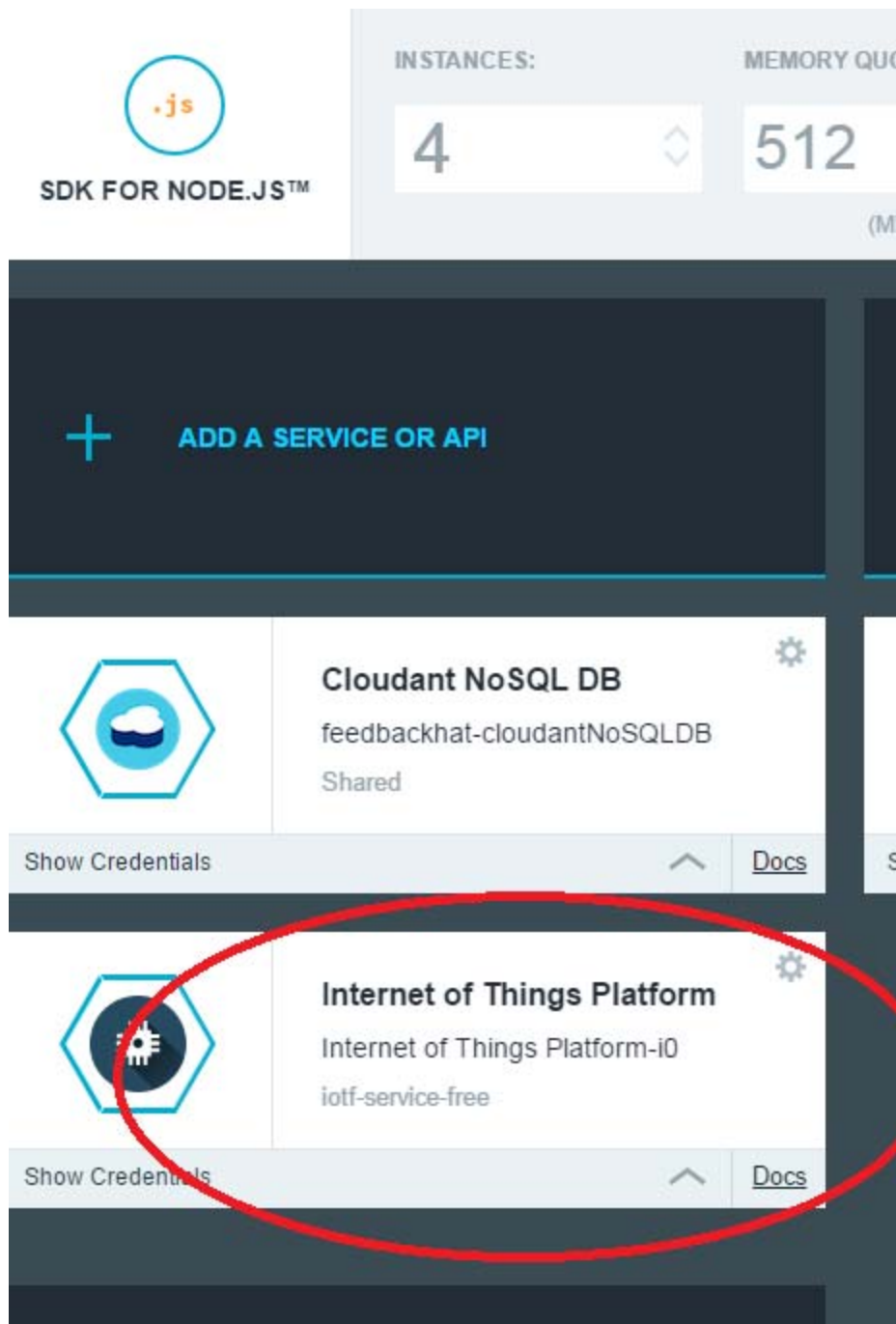
Register the device with the Bluemix IoT platform.

You need to provide

- (i) *Device name and*
- (ii) *DeviceID*

to identify the device. In return Bluemix IoT platform will return the auth credentials which can be used for data communication from the device to the Bluemix IoT Platform

Go back to Dashboard and select the application that got deployed just now as shown



Click on “Internet of Things Platform” as shown above

IoT Platform console page is reached. Click on Launch Dashboard as shown below:

feedbackhat

- Overview
- SDK for Node.js™
- Files
- Logs
- Environment Variables
- Start Coding

SERVICES

- Cloudant NoSQL DB
- Internet of Things Platform
- Internet of Things Platform >**

Hi! Welcome to the Watson IoT Platform

Take a look at the steps below to get you going with your Internet of Things app

Connect your devices

Use our [recipes](#) to find out how to add your devices. We work with partners and have sample connection recipes for many devices.

Launch the Watson IoT Platform dashboard and add your devices by clicking the 'Add Device' button under the 'Devices' tab.

Launch dashboard

Learn how to build your app

When you have added your devices, you can come back to Bluemix to start building your app using your real-time and historical device data.

Read the docs to find out how to make the most out of your app.

Go to docs

Learn

Use other i app to st

Here are s

Twilio
Third Party

Now its time to Add Device. Follow the screen shot below:

IBM Watson IoT Platform

QUICKSTART SERVICE STATUS DOCUMENTATION BLOG jkumar@in.ibm.com ID: (dufwpq)

Devices

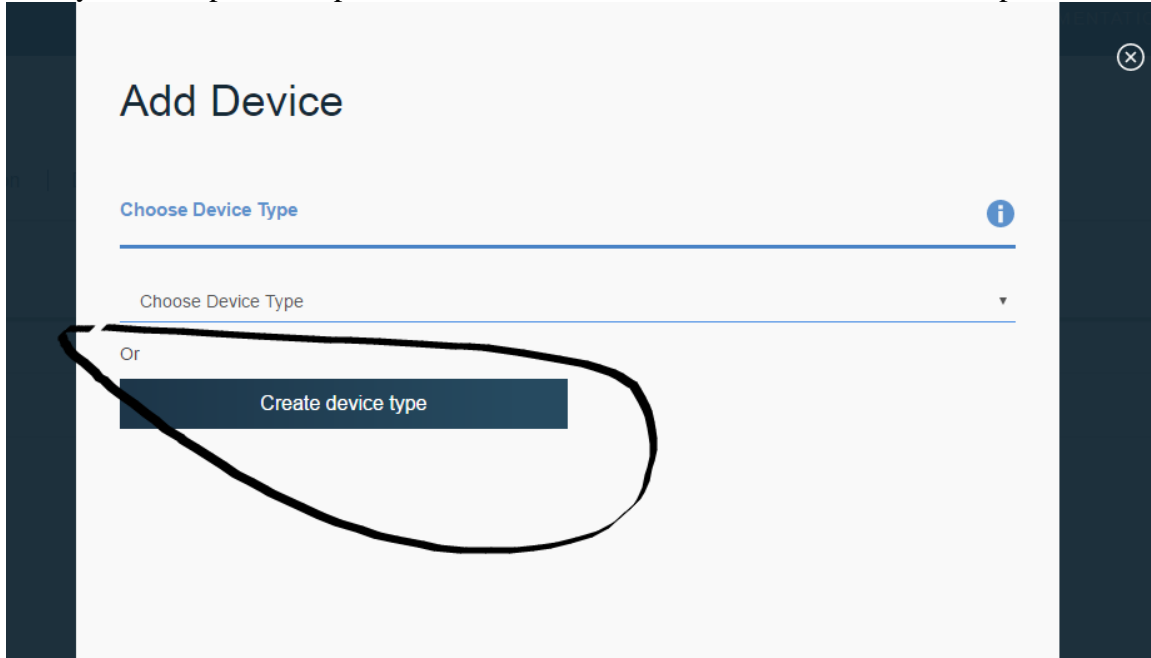
Browse | Diagnose | Action | Device Types

Refresh + Add Device

Device ID	Device Type	Class ID	Date Added	Location
Results 1-1 of 1				
cognitivehomeID	cognitivehome	Device	Jul 5, 2016 2:52:29 PM	

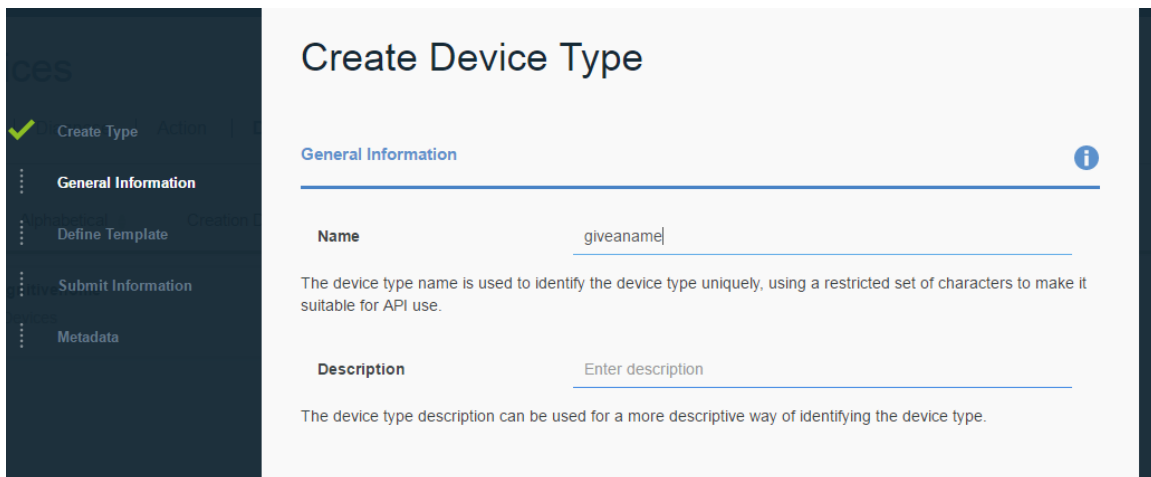
Click on Add Device

Here you are expected to provide Device name and Device ID. Follow the steps below.

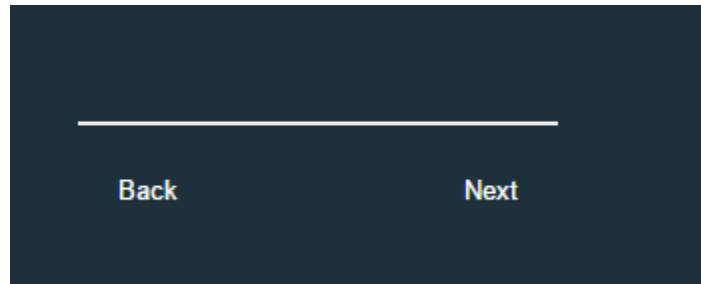


Click on “Create Device Type” and give a name to the device type. Pls .note that this will be used while connecting from Bluemix node. Hence provide any unique name which is easy to remember.

Note: A device type is intended to represent a group of devices that are identical.



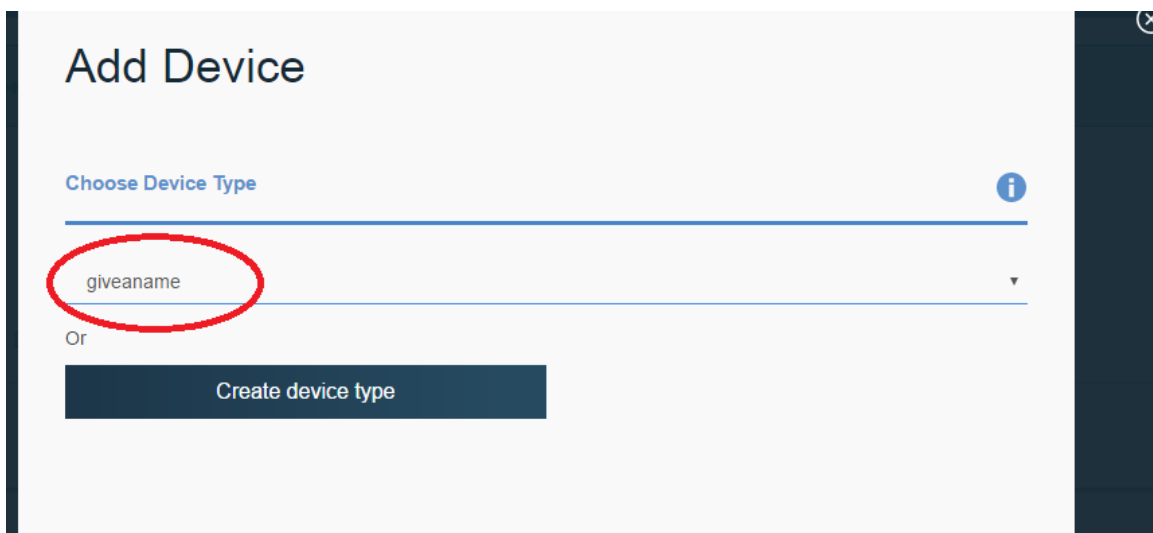
Procced with Next...



Take the default values for rest of the screens and complete it with create..

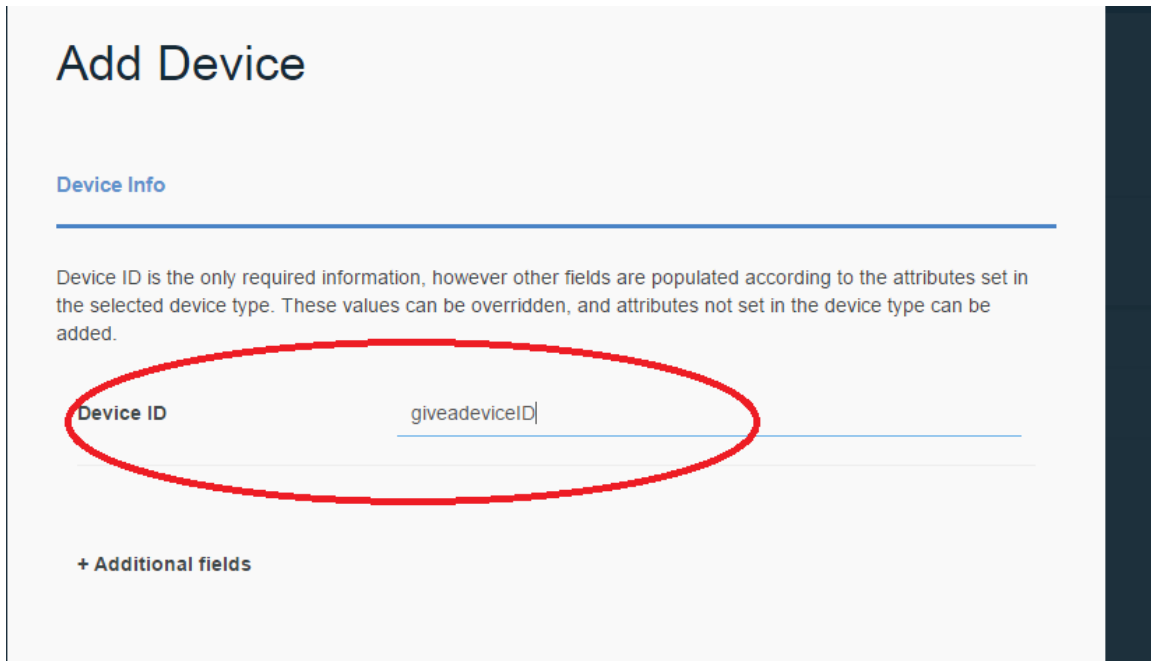


You will be back to create Device with the device type just created been listed as shown below..



Proceed with

Give a deviceID for the devicetype just created. Let it be an unique easy to remember ID. This ID will be used at the Bluemix side while sending commands back to the device from Bluemix.



The screenshot shows a web form titled "Add Device". Under the "Device Info" section, there is a text input field labeled "Device ID" containing the text "giveadeviceID". This field is circled in red. Below the field is a link that says "+ Additional fields".

Procced with



A dark blue rectangular button bar containing two white text buttons: "Back" on the left and "Next" on the right, separated by a horizontal white line.

Take the default for the subsequent screens till you reach the following page:

Add Device

Summary

Please check that all submitted information for this device is correct before adding this device.

Device Type	giveaname
Device ID	giveadeviceID
Serial Number	-
Manufacturer	-
Model	-
Class	-
Description	-
Firmware Version	-
Hardware Version	-

Back **Add**

Procced with Add...

Tthe device is been created. Bluemix returns back with auth credentials, as shown below:

Pls. save this as **device.json**. This is required to connect with the Bluemix Watson IoT platform. Once lost, to retrieve, need to re-register the device again.

Note: In our Lab, we refer it in the client program which runs on the Intel Edison board

```
{
  "org" : "d: g",
  "id" : "cognitivehomeID",
  "type" : "cognitivehome",
  "auth-method" : "token",
  "auth-token" : ")g pZVRAL"
```

Device giveadeviceID

Device

Refresh

Your Device Credentials



You have registered your device to the organization. To get it connected, you need to add these credentials to your device. Once you've added these, you should see the messages sent from your device in the 'Sensor Information' section on this page.

Organization ID	dufupq
Device Type	giveaname
Device ID	giveadeviceID
Authentication Method	token
Authentication Token	baDe7wBZD2)u3g@k

Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the device to generate a new authentication token

Congratulations !! your step 1 is over

Proceed to **step 2** with Intel configuration

- ➔ Install the Intel drivers
- ➔ <https://software.intel.com/en-us/iot/hardware/edison/downloads>

Step 2: Get Started with Intel Edison Board

1) Configure Intel Edison kit - <https://software.intel.com/en-us/get-started-edison-windows-32>

- (i) Install the USB and Serial drivers
 - a. <https://software.intel.com/en-us/get-started-edison-windows-32-step2>
- (ii) Install XDK tool kit
 - a. <https://software.intel.com/en-us/iot/software/ide>
- (iii) Flash the Intel chip firmware
 - a. <https://software.intel.com/en-us/get-started-edison-windows-32-step3>

2) connect the Intel kit using the USB and chk that the kit is been recognized as a filesystem at windows (say D:/Edison)

3)Use putty – connect to the device using the static IP : 192.168.2.15 with “root” as user

Congratulations !! now you are connected with the Intel device...proceed to next step in executing the code in making mqtt connection with Bluemix Watson IoT platform

Step 3: Download the code and run it

- Code available at github.com/EcoDIndia/cognitivehome
- Download the files:
 - o Cognitivehome.js
 - o Device.json
- Modify the device.json file to replace it with your organization and auth token details as created in step1
- Run it as **node cognitivehome.js**

Congratulations !! Now your client program is ready to receive commands from Bluemix Watson

Step 4: Train Watson with your selfie

-Follow the lab instruction

`https:// ECoDIndia/cognitivehome/Cognitive Home
- Watson Visual Recognition Lab.pdf`

Step 5: Sending commands from Bluemix:

So far:

- ➔ Device is registered with Bluemix Watson IoT
- ➔ Device is configured and powered on
- ➔ Device client is ready to receive the command
- ➔ Watson is trained to recognize you

In this step, we give your image to Watson and based on the response from Watson, the commands will be sent back to device for further action .

Execute the application created as in step1:

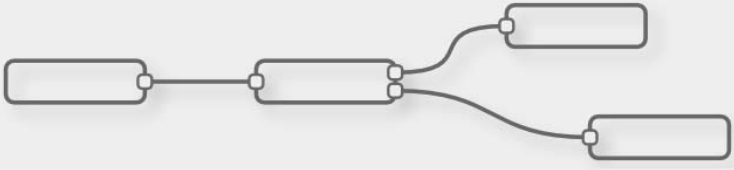
For example: myiotexample.mybluemix.net

Proceed with “Go to your Node-RED flow editor”

Node-RED in Bluemix

A visual tool for wiring the Internet of Things

IBM Watson IoT Platform



Node-RED provides a browser-based editor that makes it easy to get flows that can be deployed to the runtime in a single click.

Version running here has been customized for the IBM Watson IoT Platform.

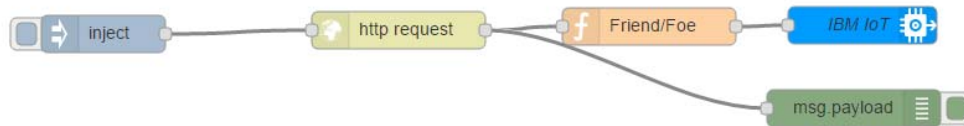
[Go to your Node-RED flow editor](#)

[Learn how to password-protect your instance](#)

Create a new flow as shown below:

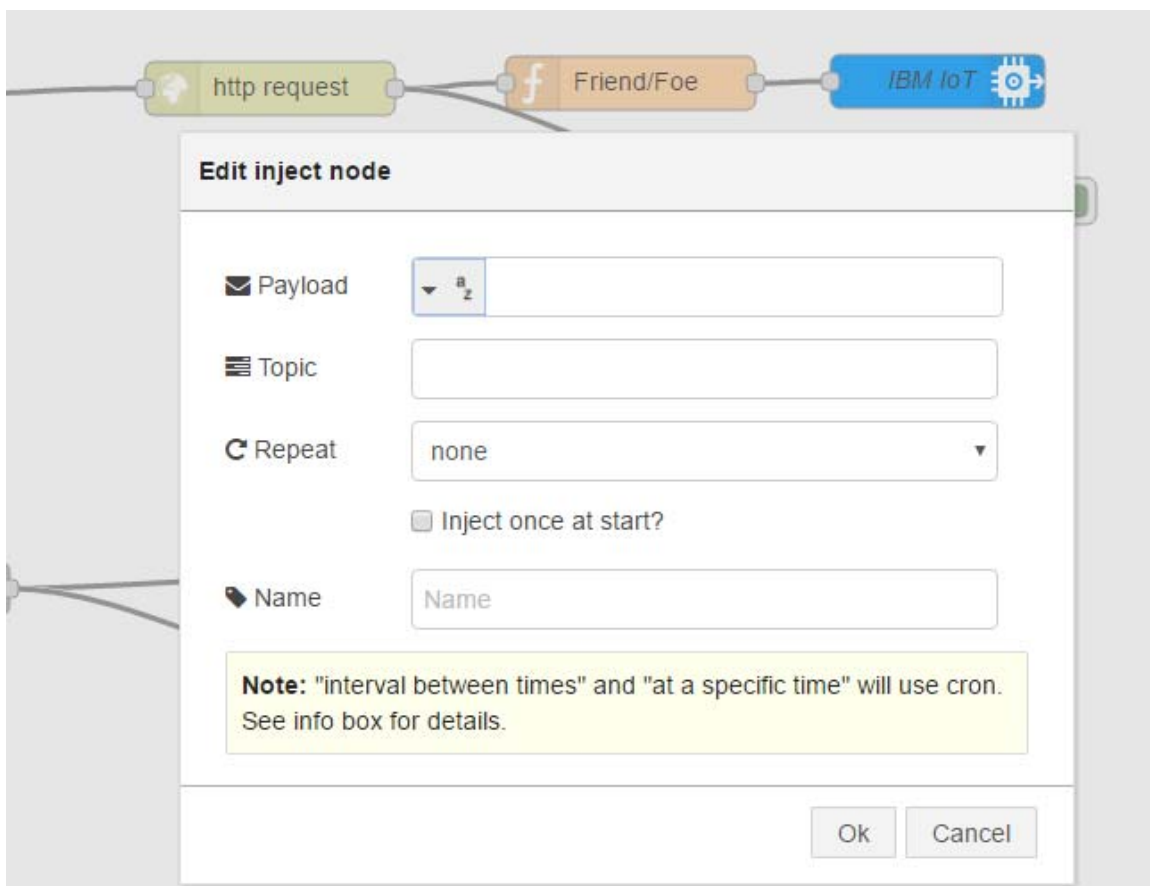
- (i) select “Inject “ node from Input

- (ii) select “http request” node from function
- (iii) select “function” node from function (in this case Friend/Foe)
- (iv) select “IBM IoT” node from output
- (v) select “debug node” from output



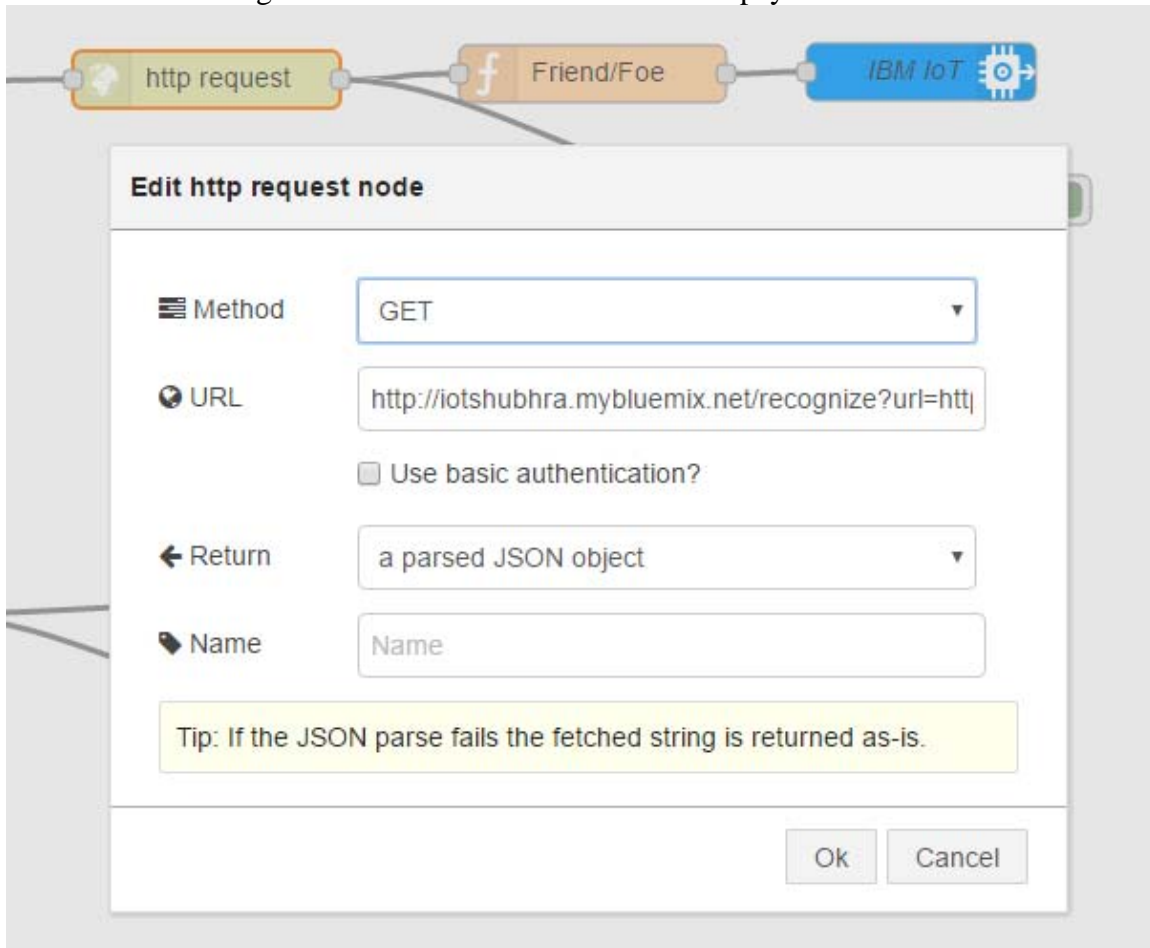
Inject node”

No additional input required. Leave it as default. We use this to inject a request to http request node



http request node:

In this node, provide the URL where the image of selfie is uploaded. This could be a Watson trained image or a normal one. Based on this the payload will be created.



Sample URL:

http://iotshubhra.mybluemix.net/recognize?url=https://s31.postimg.org/q68nw62p7/rajes_h.jpg

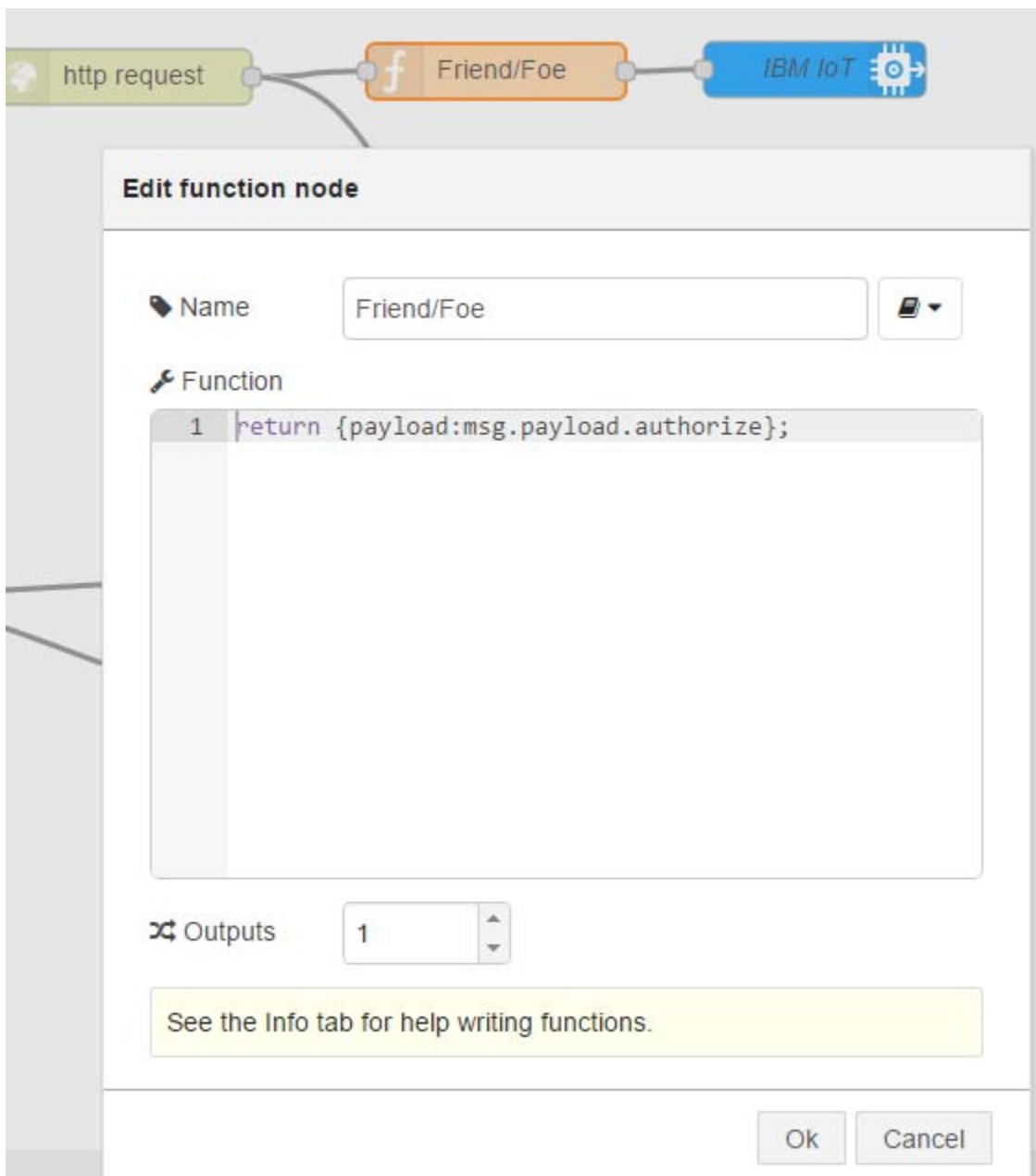
Note:

- ➔ iotshubhra.mybluemix.net in the above link is the URL of the Watson application which is trained to recognize the image. If your case use the application you had deployed or reuse the above link. Go with Instructors advice.
- ➔ https://s31.postimg.org/q68nw62p7/rajes_h.jpg is the link where the image is been uploaded

Function node:

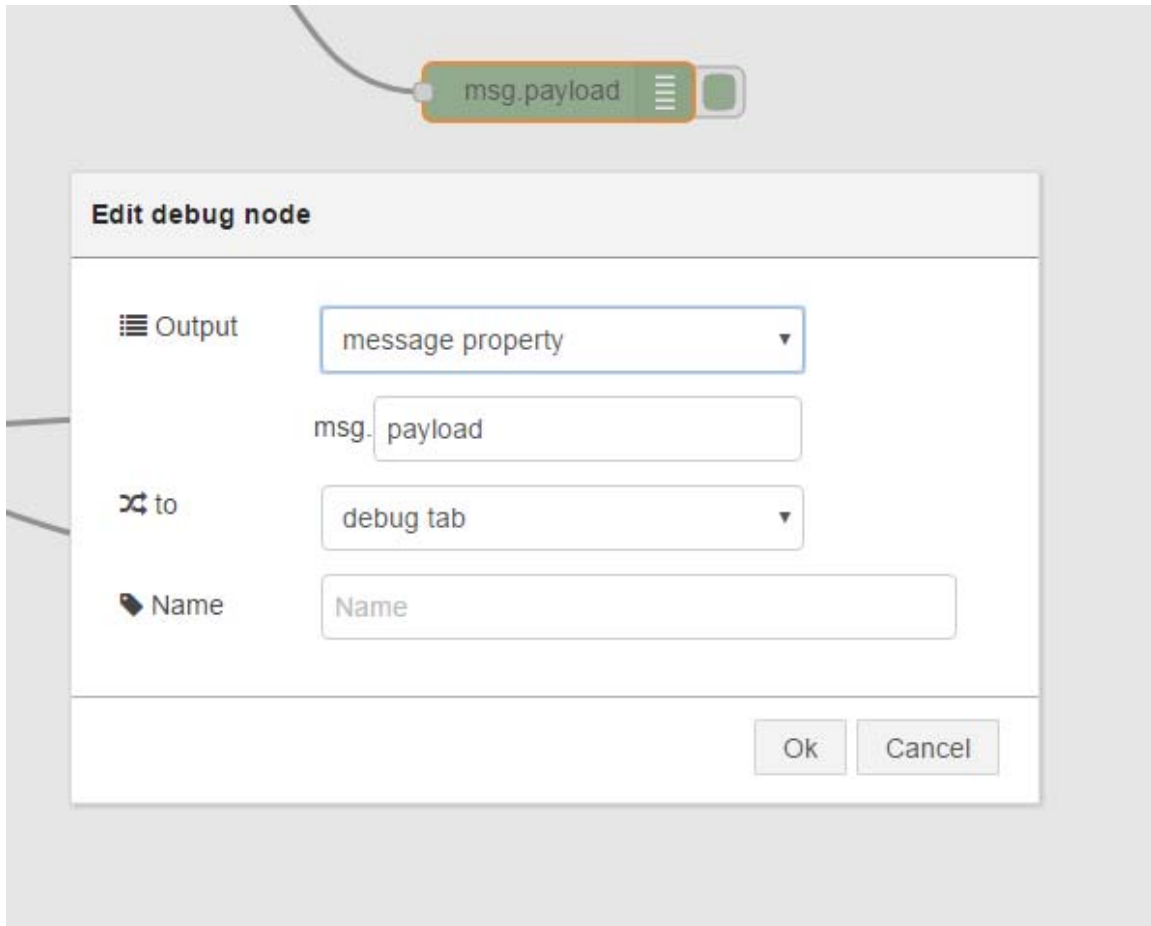
Add the following function “return {payload:msg.payload.authorize;}” as shown below.

This returns the payload “authorize” from the http URL from the previous node



Debug node:










This is a debug node to understand the output from payload



IBM IoT Node:

->select “API Key” as Authentication. Thus we use the key provided by Bluemix (as in step1) for device authentication. The following screen shot demonstrates from where the API key should be taken

Edit ibmiot out node

 Authentication	API Key
 API Key	cognitivehome
 Output Type	Device Command
 Device Type	cognitivehome
 Device Id	cognitivehomeID
 Command Type	identifyme
 Format	text
 Data	
 Name	IBM IoT

Note: If there is a property in the message that corresponds to any of the values entered above, then the property in the message takes precedence. See the Info tab for more details.

Example JSON device event: {"d":{"myName":"Arduino Uno", "temperature":989}}

OkCancel

Edit ibmiot out node

Authentication

API Key

API Key

cognitivehome

Output Type

Device Command

Device Type

cognitivehome

Device Id

cognitivehomeID

Command Type

command type e.g. blink

Edit ibmiot config node

Global

Name

cognitivehome

API Key

a-...0af9

API Token

.....

4 nodes use this config

Delete

Update

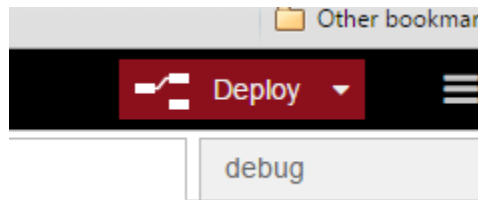
Cancel

The screenshot shows the 'Internet of Things Platform' console. At the top, there's a header with the IoT Platform logo and the text 'Internet of Things Platform' and 'Internet of Things Platform-i0'. Below this, there's a 'Show Credentials' button and a 'Docs' link. The main content area is titled 'Instantiating Credentials' and displays a JSON configuration for the 'iotf-service'. The JSON includes fields for name, label, plan, and credentials. The credentials section contains fields for mqtt_host, mqtt_u_port, mqtt_s_port, base_uri, http_host, org, apiKey, and apiToken, many of which are redacted with yellow boxes.

```
{
  "iotf-service": [
    {
      "name": "Internet of Things Platform-i0",
      "label": "iotf-service",
      "plan": "iotf-service-free",
      "credentials": {
        "iotCredentialsIdentifier": "a2gok55516r5",
        "mqtt_host": "dufwpq.messaging.internetofthings.ibmcloud.com",
        "mqtt_u_port": 1883,
        "mqtt_s_port": 8883,
        "base_uri": "http://dufwpq.messaging.internetofthings.ibmcloud.com",
        "http_host": "dufwpq.internetofthings.ibmcloud.com",
        "org": "dufwpq",
        "apiKey": "a2gok55516r5",
        "apiToken": "a2gok55516r5+3qf+@s"
      }
    }
  ]
}
```

Then proceed to provide the Device name and Device ID as created in step 1:

Good to send the command through node..



→ Watch out the debug monitor for the payload... Watch out the response from the Intel device ..

If the payload is “1” -> Watson identifies the known person as friend LED glows)

If the payload is “0” -> Watson identifies as an unknown person (Buzzer sound)

Congrats !! you have completed the **Cognitive home** Lab !!
