

# Introduction to IoT: Build Your Own Basic IoT App

## World of Watson Hands-On Workshop

Download this PDF and Node-RED flows at :

<https://github.com/johnwalicki/WorldofWatson-IoT-Lab>

Author:

John Walicki | walicki@us.ibm.com

| @johnwalicki



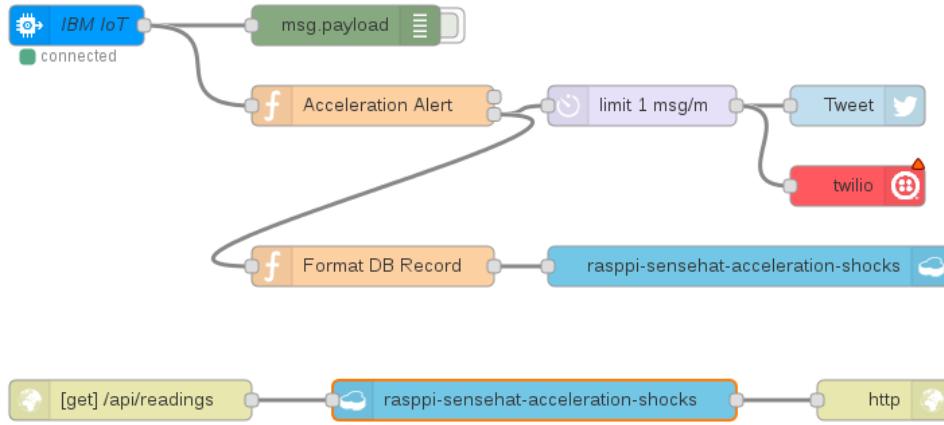
## Table of Contents

---

Topics covered: Raspberry Pi, Node-RED, Watson Internet of Things Platform, Cloudant NoSQL Database, Twitter, Twilio, and Weather Company Data for IBM Bluemix.

In this lab, we will unbox and set up a Raspberry Pi and Raspberry Pi SenseHat and connect to the IBM Bluemix and Watson IoT Platform. We will query the temperature via the SenseHat and send data to the Watson IoT platform. Watson IoT Platform will monitor the temperature and alert maintenance of a high temperature. Using Node-RED, running on the Raspberry Pi and in Bluemix, the application will read the temperature value from a SenseHat. If the temperature rises, a LED light is turned on. The temperature will be sent via Watson's Internet of Things Platform service to a Node-RED application hosted on IBM Bluemix. If the SenseHAT experiences excessive acceleration shocks, we'll send a tweet via Twitter. We will retrieve outside weather data and display it on a LED screen connected to the SenseHat. We'll store acceleration shocks in a Cloudant NoSQL database so we can track patterns. Finally, we'll create an HTTP endpoint that exposes the historical acceleration shocks that third-party applications could consume and perform analysis or other fun stuff.

Raspberry Pi Set Up.....	3
Send data from the Raspberry Pi to Watson IoT.....	6
Local Node-RED Temperature Alert.....	10
Send Sensor data to a Bluemix Application.....	11
Send command data from Bluemix Application.....	16
Registered Watson IoT Platform Devices .....	19
Send SenseHAT Acceleration data to Watson IoT.....	23
Connect to Twitter and Tweet Acceleration Shocks.....	27
Use Twilio to Text Acceleration Shocks.....	29
Store Acceleration Shock Events in a Cloudant NoSQL Database.....	30
Retrieve Acceleration Shocks from a Cloudant NoSQL DB.....	33
Send Weather Alerts to a Raspberry Pi.....	35
Additional Raspberry Pi / SenseHAT Resources.....	39



# Lesson 1 – Raspberry Pi Set Up

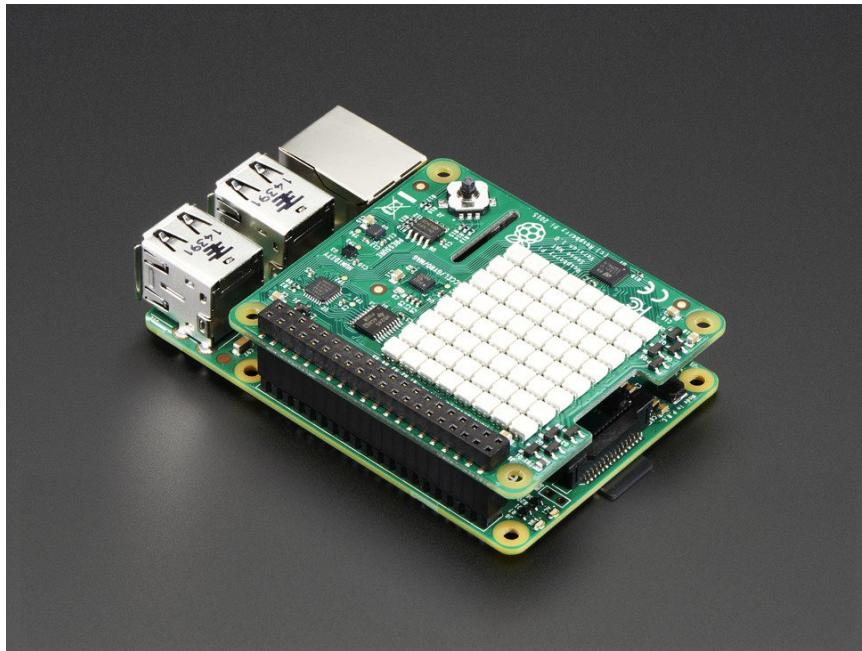
---

## Step 1 - Unbox the Raspberry Pi and Raspberry Pi SenseHAT

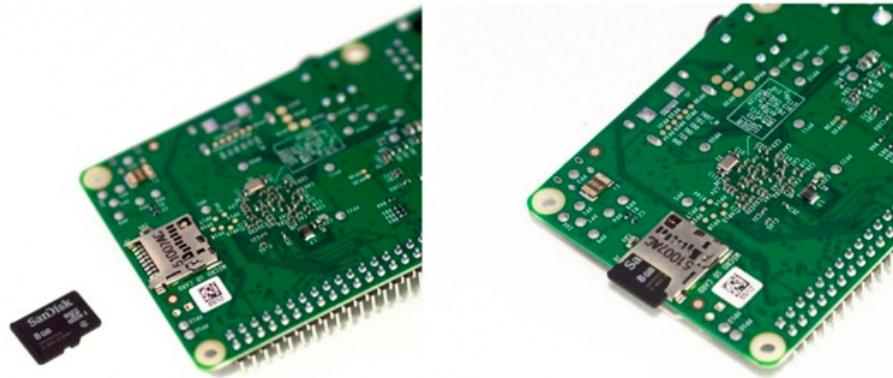


## Step 2 – Connect the Raspberry Pi SenseHAT to the Raspberry Pi

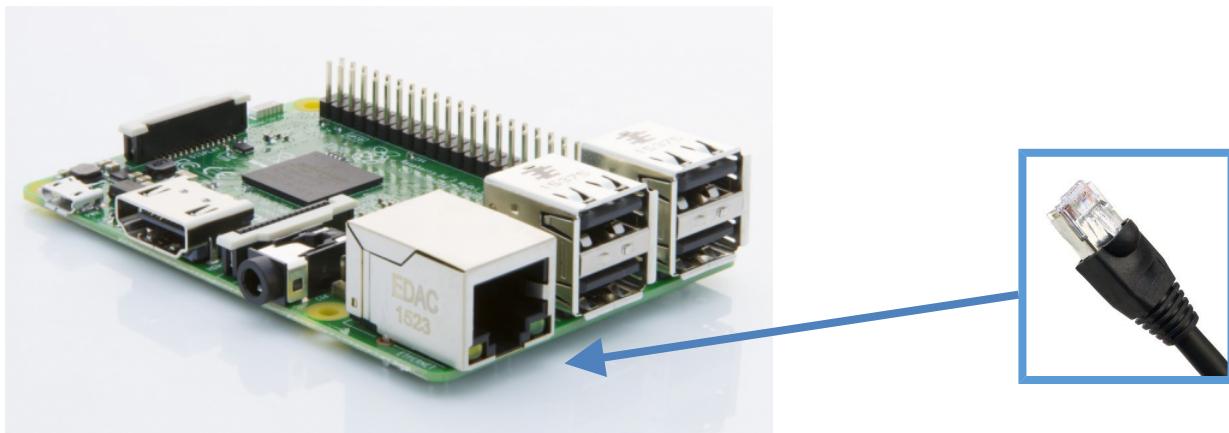
Connect your SenseHAT to the Raspberry Pi via the 40 GPIO Pins.



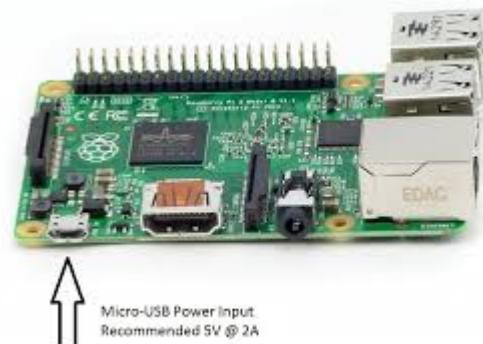
### Step 3 - Insert the Raspbian Linux microSD card into the Raspberry Pi



### Step 4 - Connect the Ethernet cable to the Raspberry



### Step 5 - Power the Raspberry Pi via a microUSB Cable or 5v microUSB Power supply



## Step 6 - Allow the Raspberry Pi to boot

The Raspberry Pi will display a red light if it is powered. After around 2 minutes, the IP addresses (Ethernet wired and WiFi) will scroll across the LED of the SenseHAT. The IP addresses will scroll for a few minutes and then turn off.

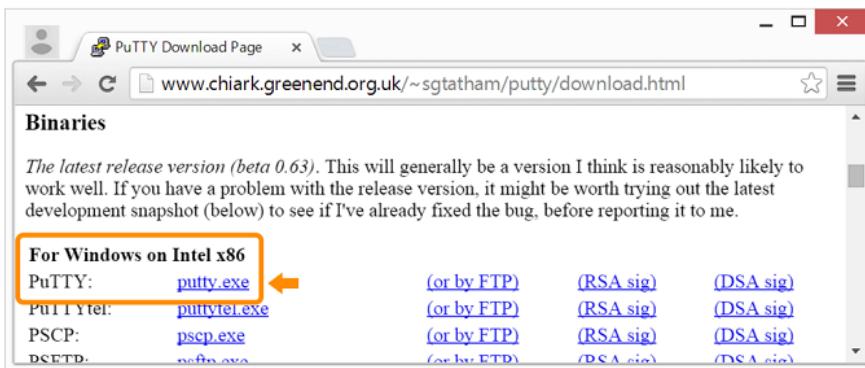
### Write Down the IP address

## Step 7 - Connect to the Raspberry Pi with a Terminal Client

In order to perform advanced configuration of the Raspberry Pi either a monitor and keyboard, or a Secure Shell (SSH) connection is required. On OSX and Linux there are default programs that can do this - Screen and SSH respectively. However on Windows no default exists, however PuTTY is light weight and easy to install and can be used to SSH into the Raspberry Pi.

### For Windows Users:

- o PuTTY is installed on the lab laptops
- o or Visit the [PuTTY download page](#).
- o Under the "For Windows on Intel x86" heading, click on the "putty.exe" link to download the latest release version to your computer.



- o Double-click putty.exe on your computer to launch PuTTY.
- o Enter IP address of the Gateway
- o You can login with the username **pi** and the password **raspberry**

### For Mac and Linux Users:

- o Open a Terminal
- o Type `$ ssh pi@<<IP Address>>`. Replace <<IP Address>> with the IP address of your gateway.
- o Enter **raspberry** as password

## Step 8 - Open a Browser and enter `http://<ip-address>:1880`

The Node-RED graphical programming tool will open in your browser.

# Lesson 2 - Send data from the Raspberry Pi to Watson IoT

The Raspberry Pi Raspbian image provided in the World of Watson lab has a program installed and running that is automatically sending the CPU temperature of your Raspberry Pi to the IBM Watson IoT Platform QuickStart. In the terminal window, enter the **following two commands marked below in bold** and copy the device ID it reports.

```
pi@raspberrypi:~ $ sudo service iot restart
```

```
pi@raspberrypi:~ $ service iot getdeviceid
```

The device ID is b827eb6effa7

For Real-time visualization of the data, visit

<http://quickstart.internetofthings.ibmcloud.com/?deviceId=b827eb6effa7>

Open a browser tab to <http://quickstart.internetofthings.ibmcloud.com/?deviceId=<device id>>

The screenshot shows the IBM Watson IoT Platform Quickstart interface. At the top, there is a navigation bar with links for QUICKSTART, SERVICE STATUS, DOCUMENTATION, and BLOG, along with a SIGN IN button. The main content area features a title "Quickstart" with a subtitle "No sign-up required to see how easy it is to connect your device to Watson IoT Platform and view live sensor data". Below this is a checkbox labeled "I accept IBM's Terms of Use". A text input field contains the device ID "b827eb6effa7" and a "Go" button. To the right, a message says "Last message received at 10:05:52 PM".

The central part of the screen displays a line graph titled "myPi" for the datapoint "status.cpuTemp". The graph shows fluctuating data over time, with a prominent peak around 22:05:33. Below the graph is a table of events:

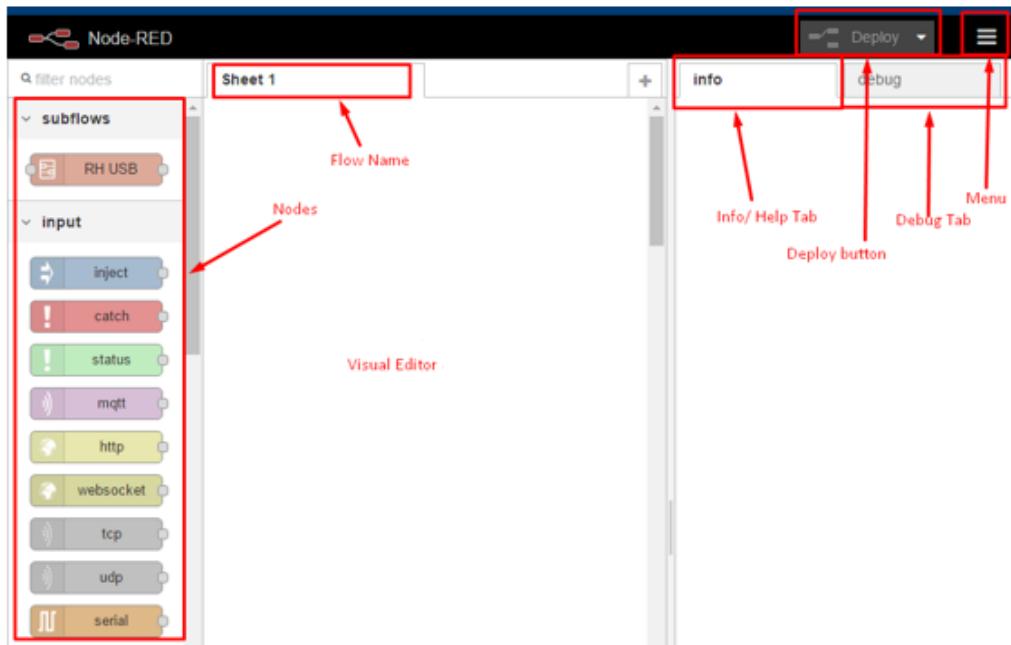
Event	Datapoint	Value	Time Received
status	myName	myPi	Oct 20, 2016 10:05:52 PM
status	cpufreq	48.31	Oct 20, 2016 10:05:52 PM
status	cpuload	0.01	Oct 20, 2016 10:05:52 PM
status	sine	0.38	Oct 20, 2016 10:05:52 PM

On the right side, there is a sidebar with the heading "I've seen my data, what next?". It includes several options with notes:

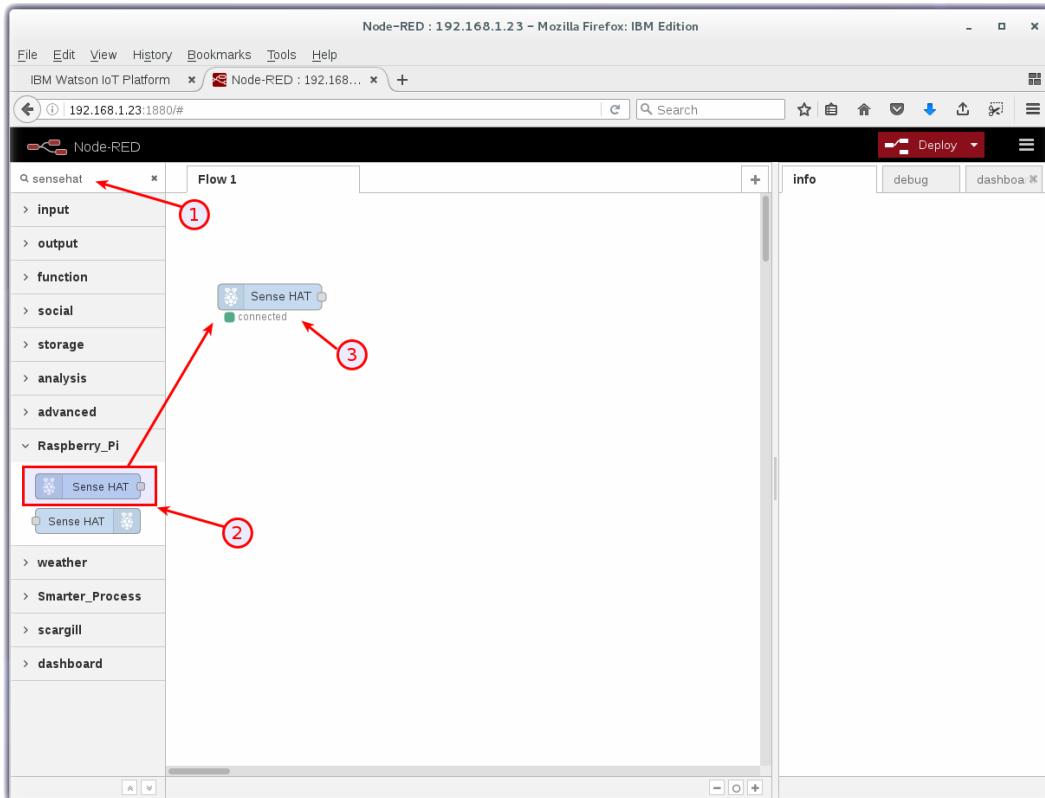
- "Use your device in an application created with IBM Bluemix." with a "Click here for more details." link.
- "Go to your Bluemix account" with "SIGN UP" and "LOG IN" buttons. Note: "When you sign up for a trial you may have to wait up to 24 hours to receive your log-in information".
- "Create an app using the Internet of Things Starter from the Catalog" with a "CREATE APP" button. Note: "You will have to name your app and wait for a few minutes for it to start running".
- "When your app is running, select the app URL or type it into the browser to open the Node-RED flow editor" with the URL "http://<appname>.mybluemix.net".
- "Import the flow for your device into the Node-RED flow editor" with an "IMPORT FLOW" button.

## Sending SenseHAT data to IBM Watson IoT Platform Quickstart

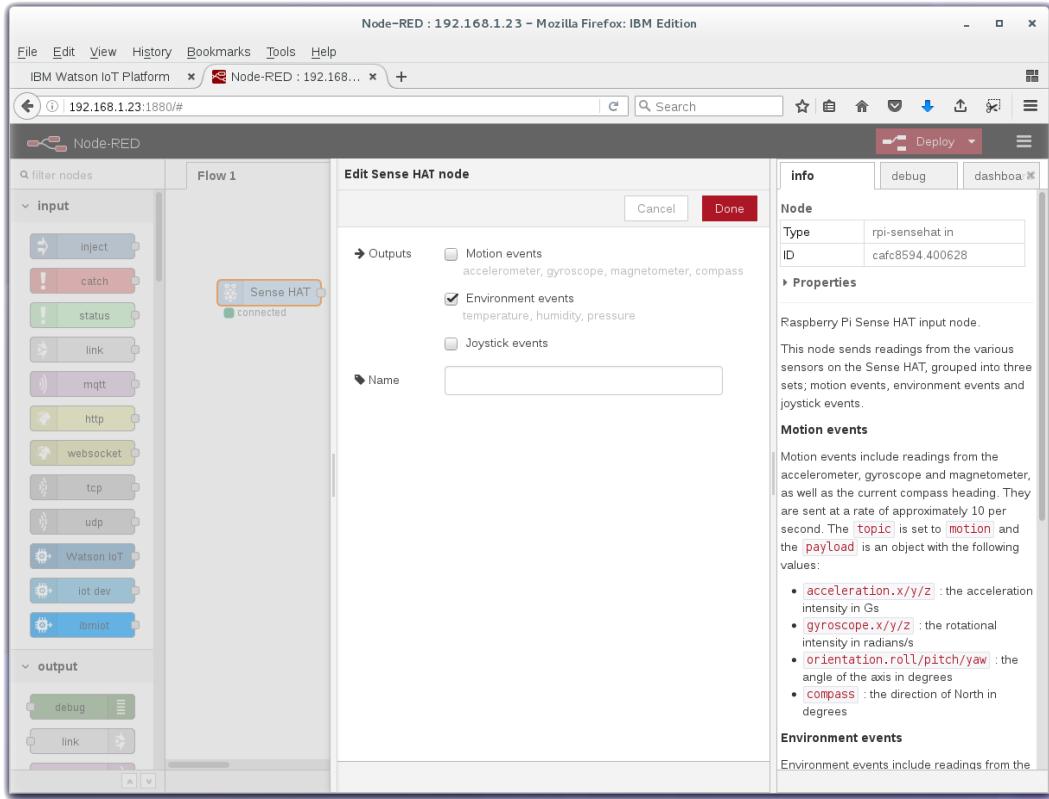
Now that we have sent the Raspberry Pi CPU temperature to Watson IoT, let's also send SenseHAT environmental sensor readings by using Node-RED. Create a new flow to program the rest of the exercises by clicking on the + button in the upper right corner of the Node-RED interface to open a Flow 1.



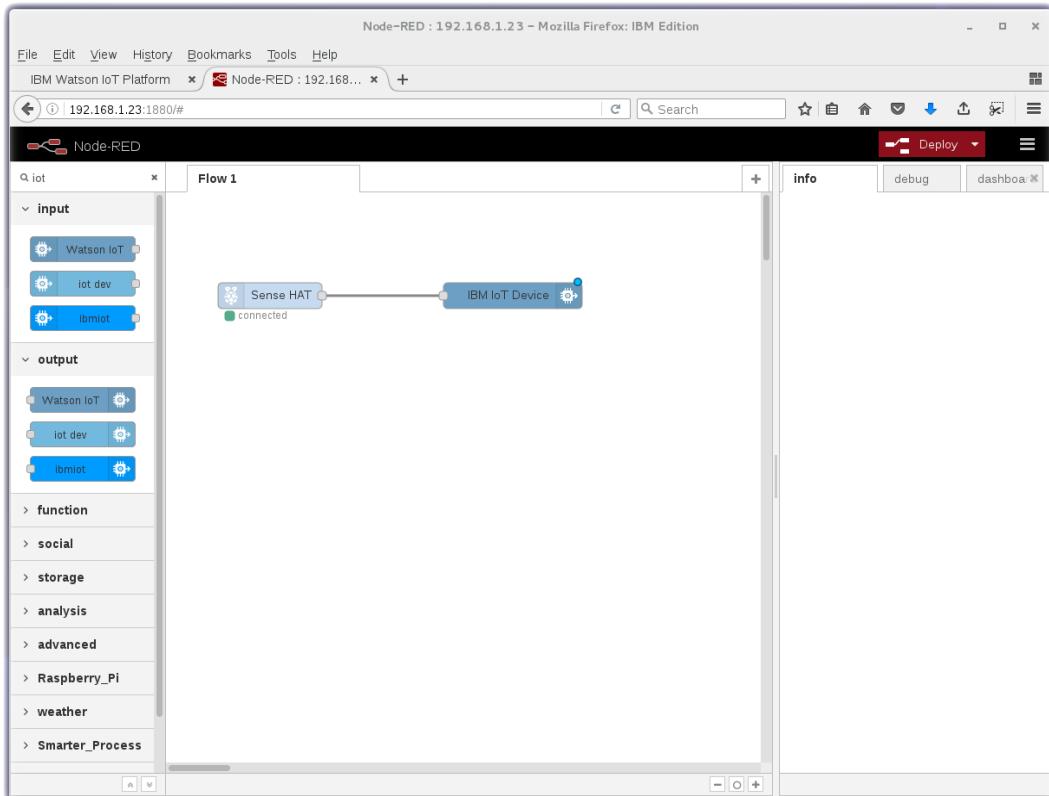
- Find a SenseHAT node by searching in the filter node box (1). Drag the SenseHAT node (2) from the palette onto the flow (3).



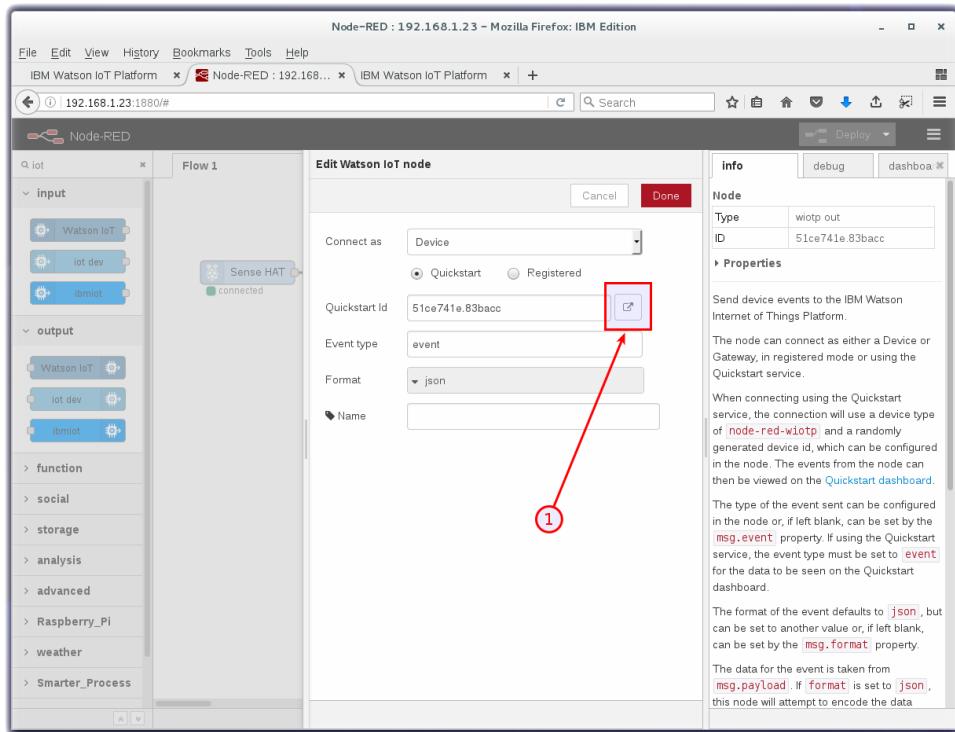
- Double click on the SenseHAT node on the Flow and select the “Environment events” checkbox and press the Done button.



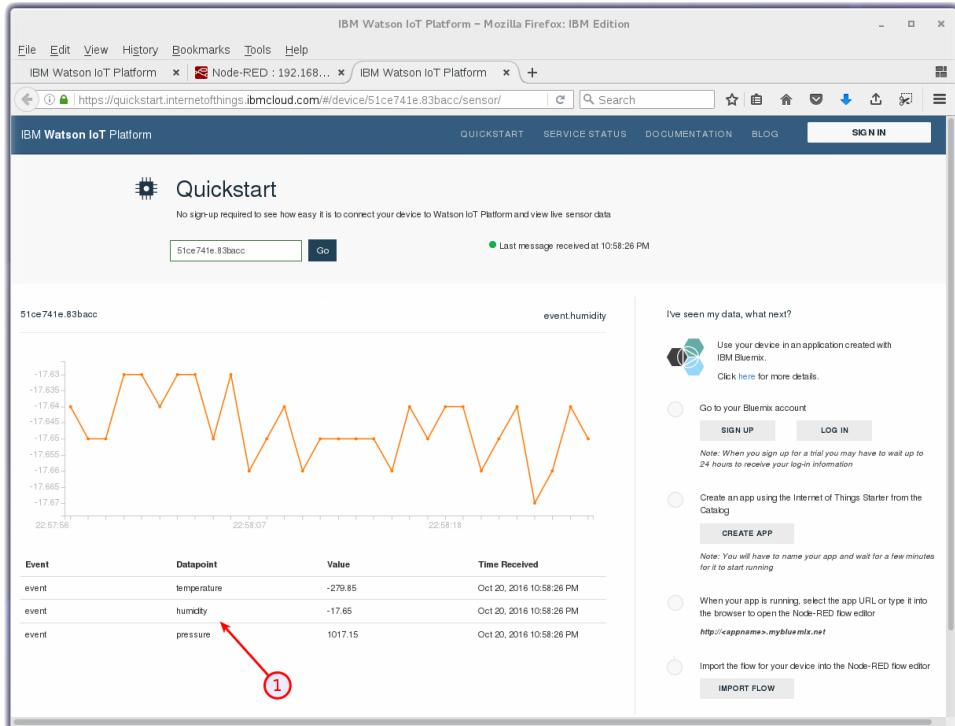
- Search for a Watson IoT output node on the palette and drag it onto the flow. Wire it to the SenseHAT node by dragging your mouse from one node point to the other. A wire will connect the two nodes.



- Click the Deploy  button on the top of menu bar to deploy the Node-RED flow.
- Doubleclick on the IBM IoT Device node and then click on the box (1) to the right of the Quickstart Id. A new browser tab will open. **Note the unique Quickstart Id**. This Id will be used in Lesson 4 and 5.

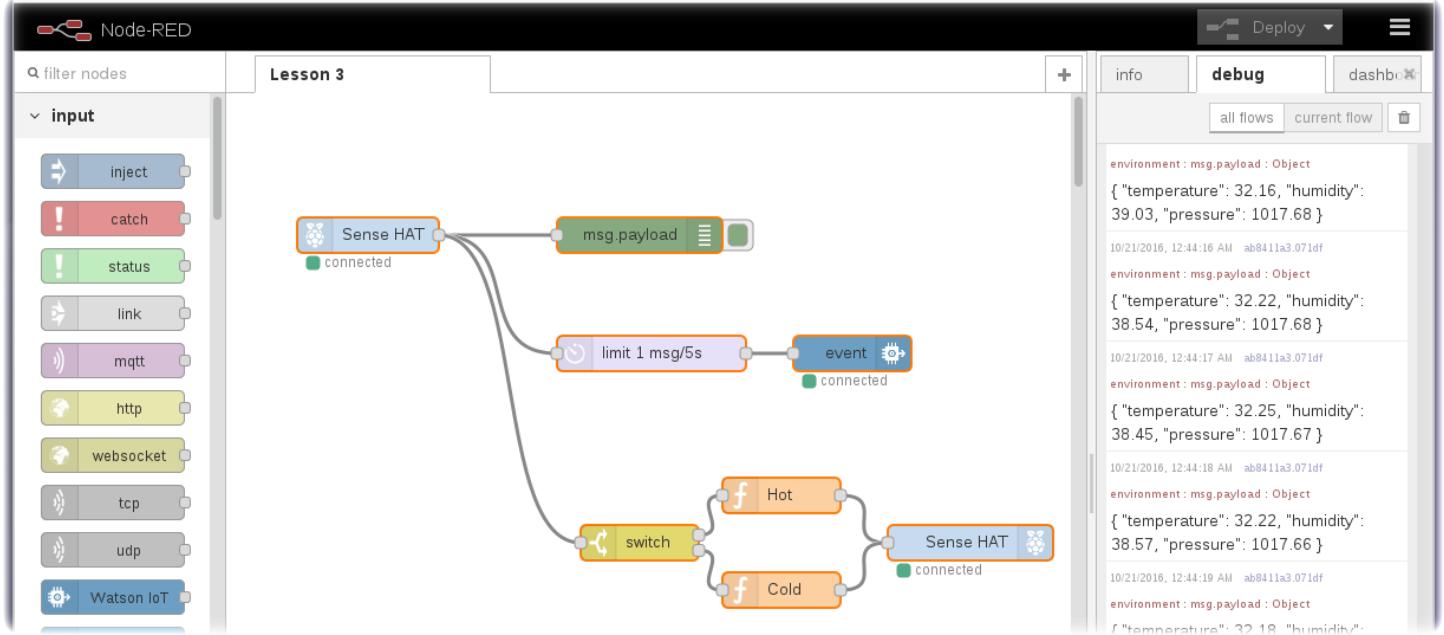


- Room temperature, humidity and air pressure sensor data from the SenseHAT is now being sent to the Watson IoT Platform and is being plotted in the Quickstart landing page. You can click (1) on any of the three datapoints to plot them.



# Lesson 3 – Local Node-RED Temperature Alert

To practice with Node-RED visual programming, we might want to limit the amount of sensor data that is sent to Watson IoT Quickstart to 1 msg / 5 secs. It is also interesting to watch the SenseHAT environmental data by reporting it to a debug node before it is sent to Quickstart. Depending on the temperature, display if the room is hot or cold by changing the SenseHAT LEDs to red / green.



The text below is a Note-RED representation of the above graphical flow. You can copy it to the clipboard and paste it in your Node-RED browser tab. From the **Node-RED menu, select Import → Clipboard, paste the nodes and press the red Import button**. Finally press the Deploy button.

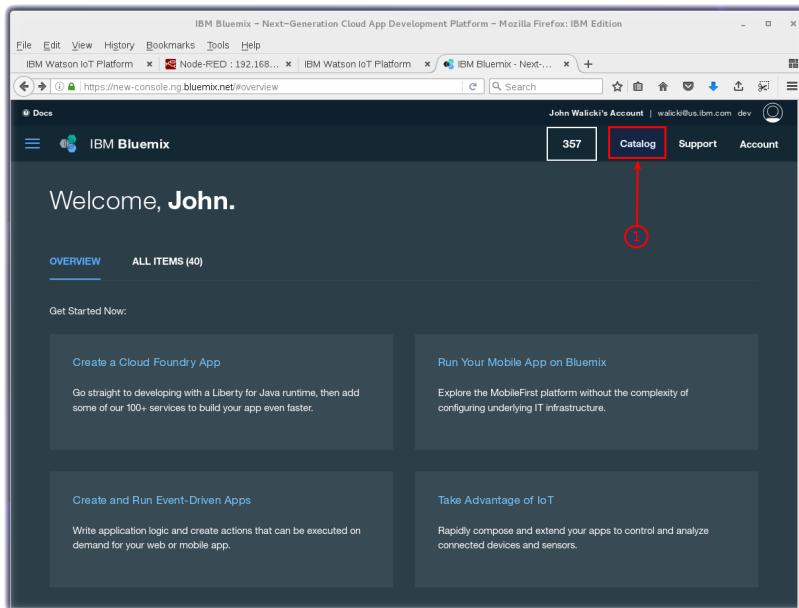
```
[{"id":"cafc8594.400628","type":"rpi-sensehat
in","z":"9e7e27aa.f908a","name":"","motion":false,"env":true,"stick":false,"x":120,"y":120,"wires":[]},
[{"id":"ab8411a3.071df","z":"9eedaa.700b1a8","a8888a05.b142e"}], {"id":"sampleDeviceID","type":"wiotp
out","z":"9e7e27aa.f908a","authType":"d","qs":"true","qsDeviceId":"sampleDeviceID","deviceKey":"","deviceType":"","deviceId":"","event":"event","format":"json","name":"","x":530,"y":220,"wires":[]},
 {"id":"ab8411a3.071df","type":"debug","z":"9e7e27aa.f908a","name":"","active":true,"console":"false","complete":"false","x":350,"y":120,"wires":[]},
 {"id":"9eedaa.700b1a8","type":"delay","z":"9e7e27aa.f908a","name":"","pauseType":"rate","timeout":"5","timeoutUnits":"seconds","rate":"1","nbRateUnits":"5","rateUnits":"second","randomFirst":"1","randomLast":"5","randomUnits":"seconds","drop":true,"x":360,"y":220,"wires":[]},
 [{"id":"a8888a05.b142e","type":"switch","z":"9e7e27aa.f908a","name":"","property":"payload.temperature","propertyType":"msg","rules":[{"t":"gte","v":"32","vt":"str"}, {"t":"lt","v":"32","vt":"str"}],"checkall":true,"outputs":2,"x":350,"y":380,"wires":[]},
 [{"id":"6c9a9bfc.7b5ce4","type":"rpi-sensehat out","z":"9e7e27aa.f908a","name":"","x":630,"y":380,"wires":[]}, {"id":"6860c85c.1c05f","type":"function","z":"9e7e27aa.f908a","name":"Hot","func":"msg.payload = \"*,*,red\";\nreturn msg;","outputs":1,"noerr":0,"x":470,"y":340,"wires":[]},
 {"id":"11a8bcf5.683c4b","type":"function","z":"9e7e27aa.f908a","name":"Cold","func":"msg.payload = \"*,*,green\";\nreturn msg;","outputs":1,"noerr":0,"x":470,"y":420,"wires":[]}]}
```

This flow is available at : <https://github.com/johnwalicki/WorldofWatson-IoT-Lab>

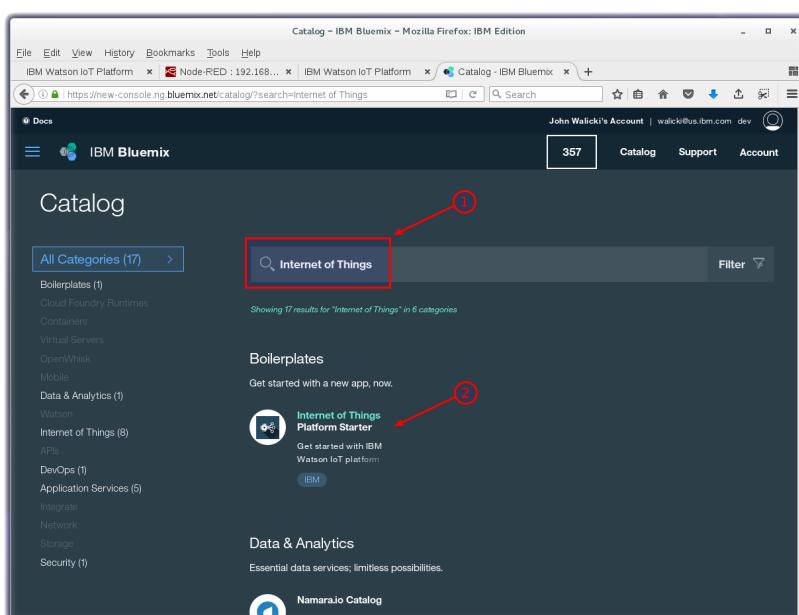
# Lesson 4 - Send sensor data to a Bluemix Application

Quickstart is a fast way to send and see sensor data on the Watson IoT Platform. The next step is to create a Bluemix Watson IoT application that receives the Raspberry Pi sensor data via the Quickstart service and takes actions on it.

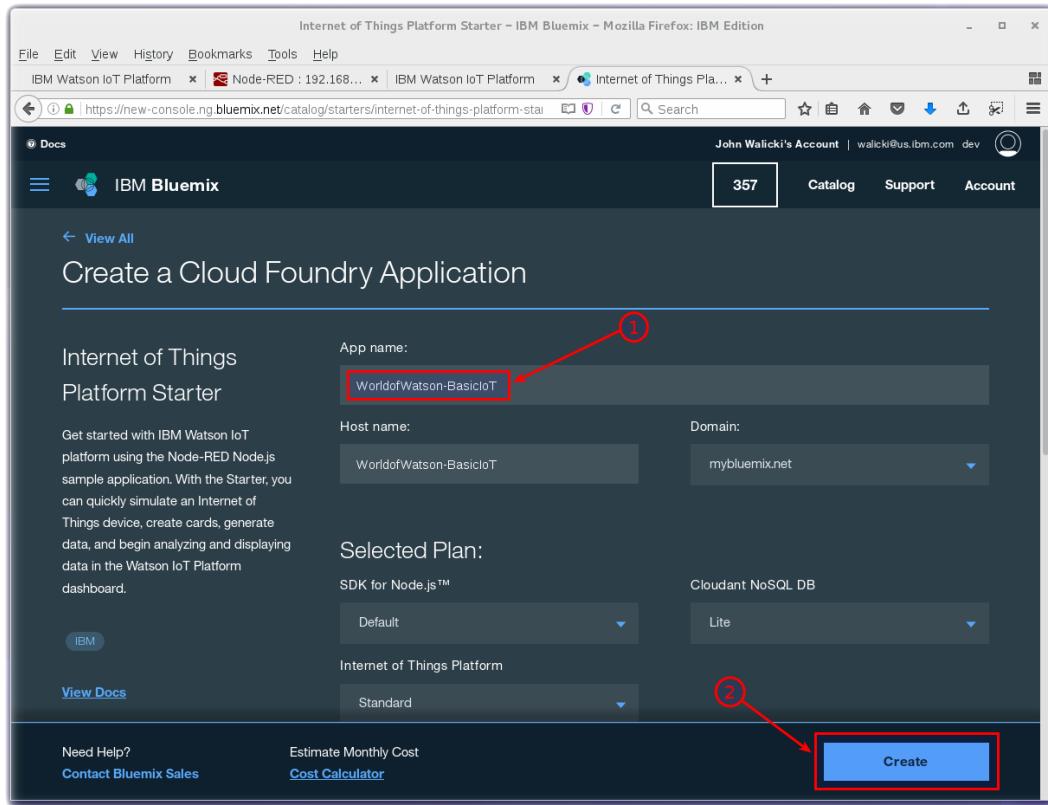
- If you do not yet have a Bluemix account, visit <http://bluemix.net/registration> and enter the requested information. Check your email to confirm the Bluemix account creation.
- See the lab instructor for a promocode to extend your 30 day free trial.
- Login into <http://bluemix.net> with your new account userid and password.
- Click on the word **Catalog** on the upper right side.



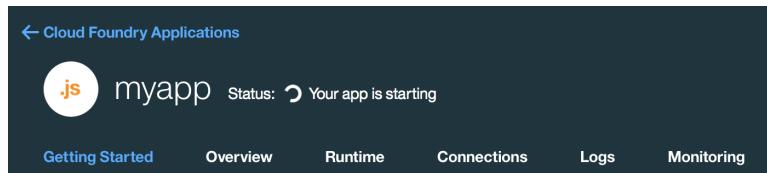
- Search for **Internet of Things Platform Starter** – The boilerplates are designed with pre-assembled services that work together. The Internet of Things Platform Starter includes a Node-RED Node.js web server, Cloudant database to store sensor data, and the IoT platform service so you can connect devices



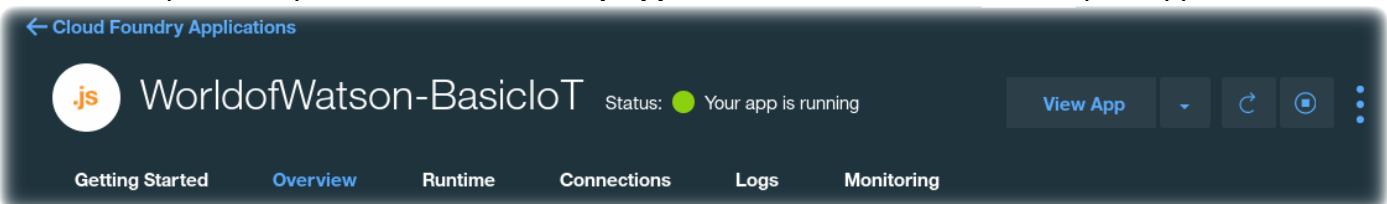
- Name your application something unique (1). If you choose **myapp**, your application will be located at <http://myapp.mybluemix.net>. There can only be one “**myapp**” application registered in IBM Bluemix. You can try adding your initials in front of the host if the host you choose is already taken by someone else. Click on **Create** (2) to create the application instance.



- IBM Bluemix will create an application in your account based on the services in the boilerplate. This is called staging an application. It can take a few minutes for this process to complete. While you wait, you can click on the **Logs** tab and see activity logs from the platform and Node.js runtime.

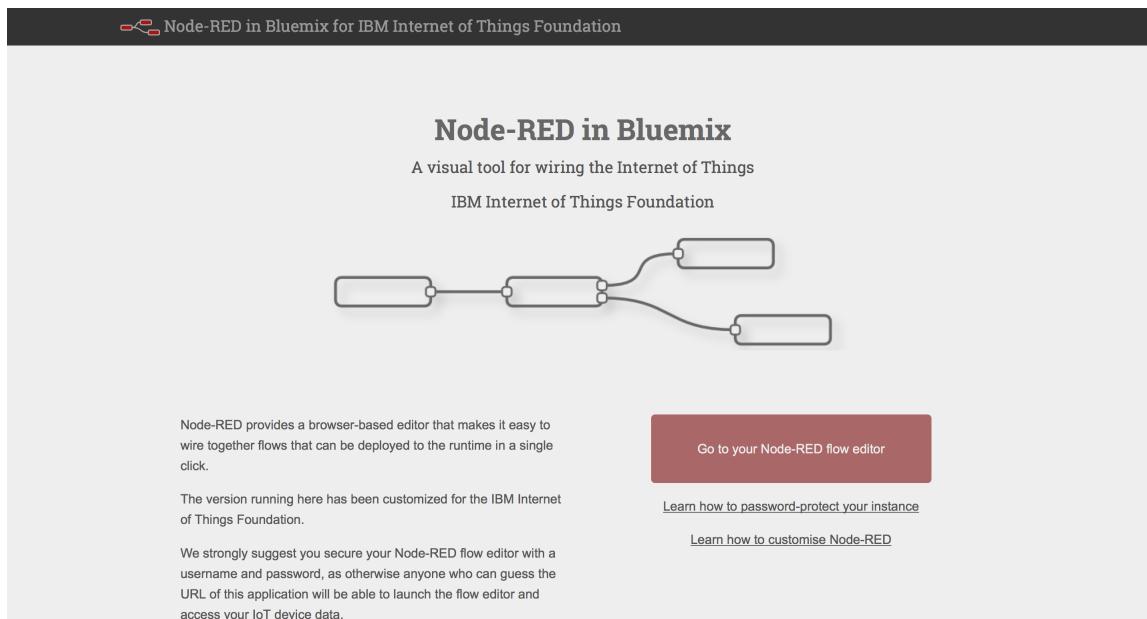


- You may need to press the **Cloud Foundry Applications** link and then re-enter your application.

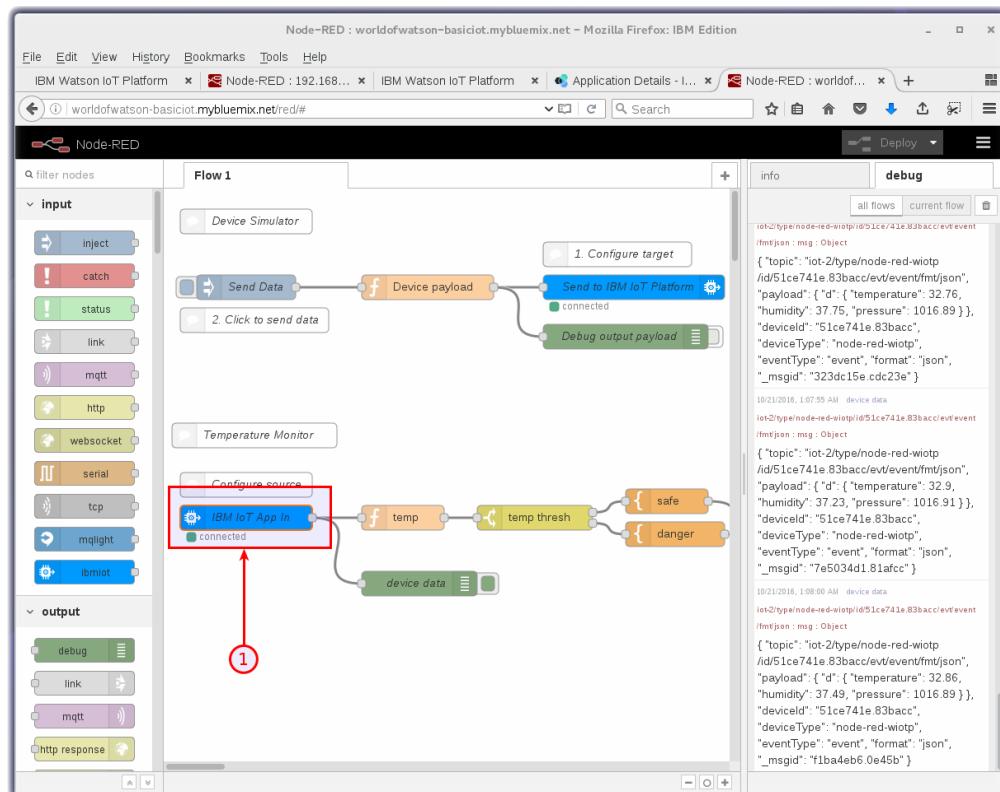


- Open your application by pressing the **View App** button.

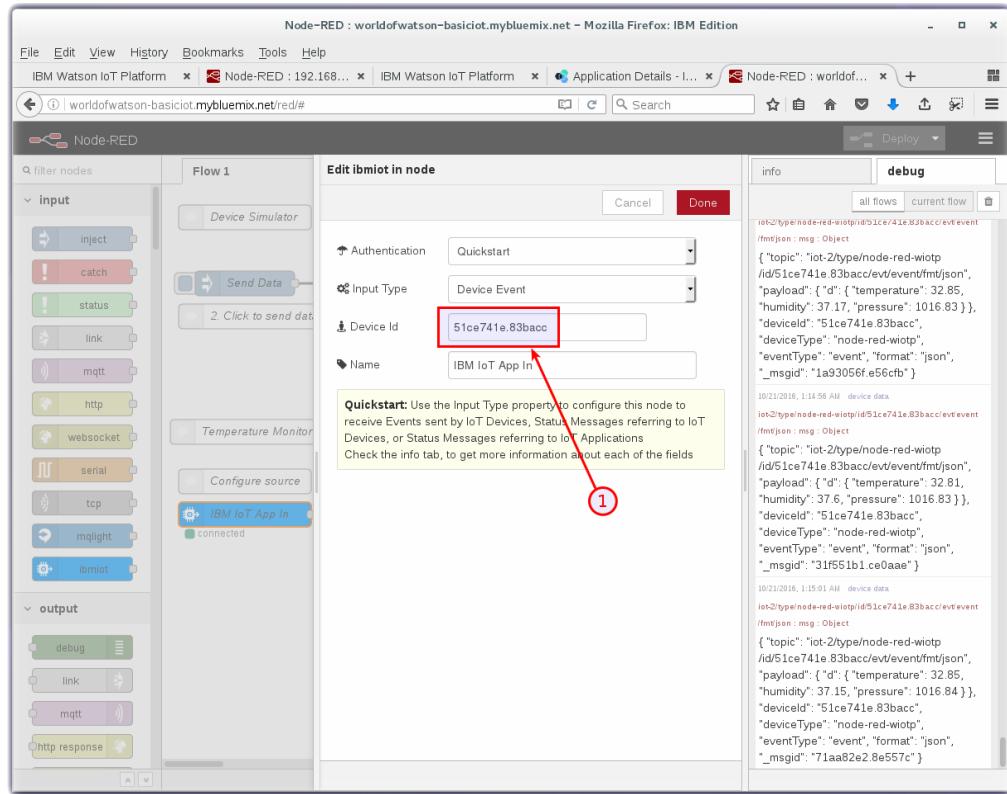
- A new browser tab will open to the Node-RED start page. Node-RED is an open-sourced Node.js application that provides a visual editor that makes it easy to wire together flows. Click on the red button **Go to your Node-RED flow editor** to launch the editor.



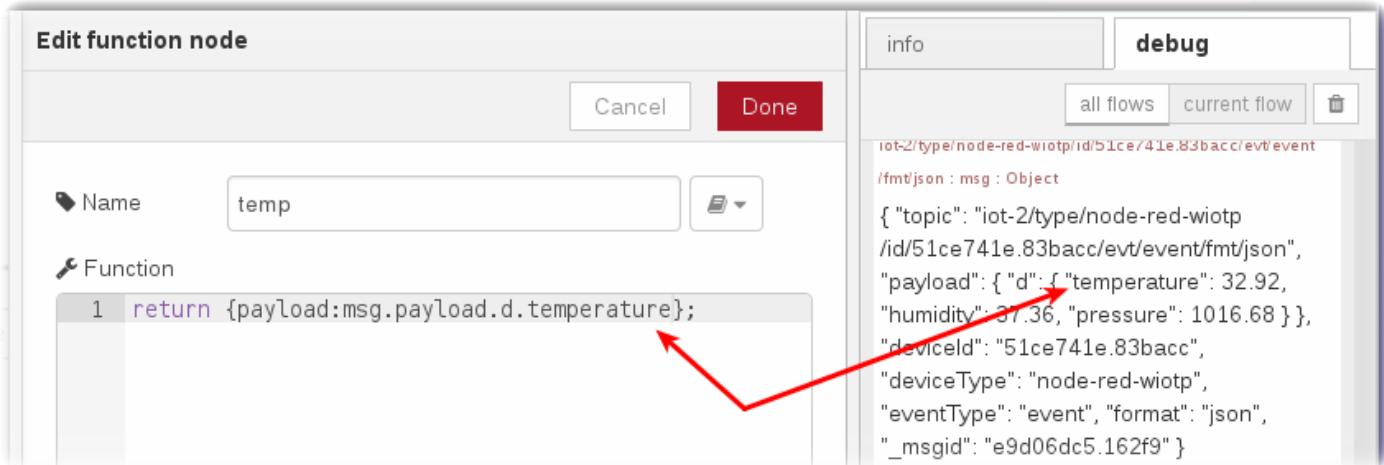
- The IoT Starter app has a default flow that can be modified. To receive the Raspberry Pi SenseHAT data from your **Quickstart ID** within our new Bluemix application, click on the IBM IoT App In node (1).



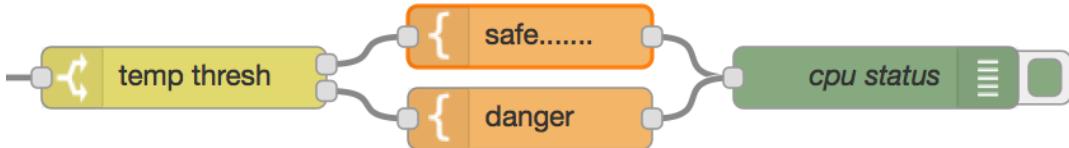
- Authentication by default is set to Quickstart. Provide the Device Id (1) from the local Node-RED Watson IoT node in Lesson 2. You can Copy and Paste the Device Id between browser tabs.



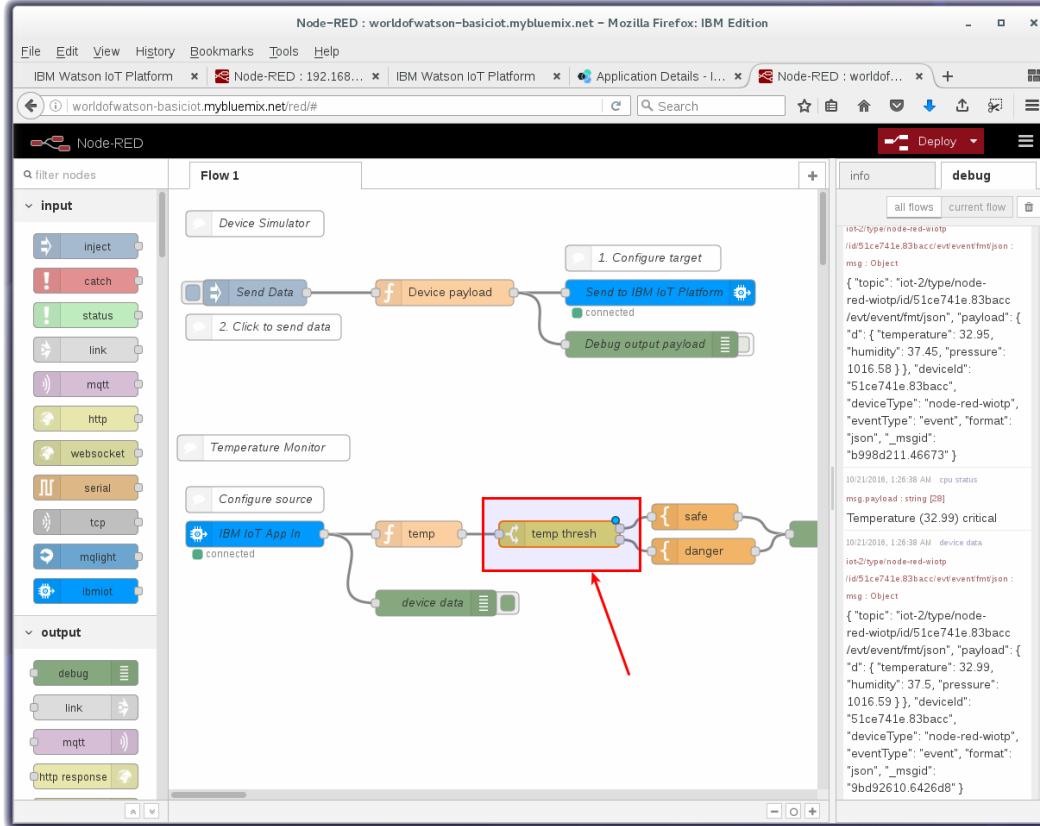
- Click on the “temp” Function node. Fix up the temp function so that it extracts the **msg.payload.d.temperature** value that is being sent from the Raspberry Pi SenseHAT.



- The yellow node is called a  node. You can program logic using a switch node and split a flow into two or more flows based on a property's value. In this example, if the temperature is less than or equal to 30°C, it is considered "safe" and continues with the flow to the template labeled safe. If the temperature is greater than 30°C, it is considered "danger[ous]" and continues with the flow to the template labeled danger.



- Set the temp threshold calibrated to your SenseHAT by clicking on the temp thresh node.



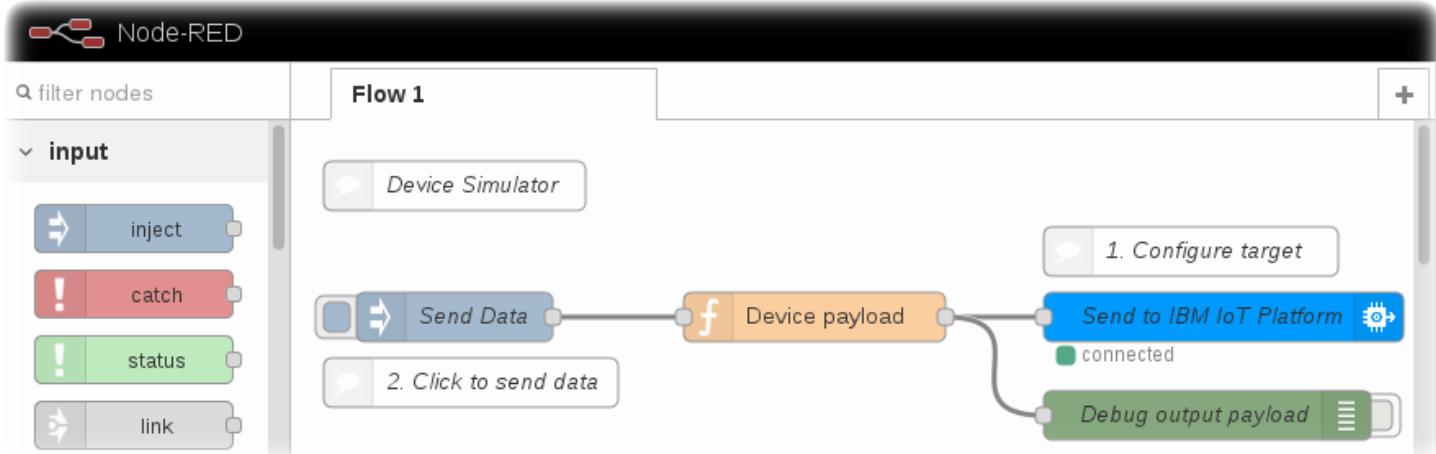
- Click on the  button in the top right of the screen to save and deploy your changes.
- Click on the debug tab in the right-hand pane. Every 5 second, the SenseHAT emits a device event to the Watson IoT platform with temperature and other data. The Node-RED application subscribes to these events and the IBM IoT In node triggers the flow with the temperature data in the message. When the debug nodes are processed, contents of the message object are in the debug tab. Adding debug nodes can be helpful when something doesn't work right and you want to see the values being passed around. The Quickstart sensor data should appear in the debug tab and the Temperature is reported as Critical or Safe within our Bluemix Watson IoT application.

# Lesson 5 – Send command data from Bluemix Application

Until this point, the lessons have sent data from the Raspberry Pi to Watson IoT. Lesson 5 will send command data to the Raspberry Pi. Recall that in Lesson 3, the SenseHat LED red/green alert colors were set if the temperature exceeded a hard coded threshold value inside the switch statement. In this section, the threshold will be sent from the Bluemix application. You will be able to push the threshold temperature down to the Raspberry Pi and dynamically decide at what temperature you want the LED alert colors to change.

**Warning:** These modifications will require changes on both the Bluemix Node-RED and on the local Node-RED flows. Be careful as you make changes in the two browser tabs. It is easy to become disoriented by which flow you are editing.

1. Let's start the modifications on the Bluemix Watson IoT Application side. Recall that the default flow for a Watson IoT Starter Application looks like this:



Two changes are necessary:

A. Instead of sending simulated data that the default “Device payload” function generates, the function must be modified to set the threshold temperature at which you want to control the color of the SenseHat LED display. If you double click on the “Device payload” function, select all of the javascript code and delete it. Replace it with this javascript function. Pick your tempthresh.

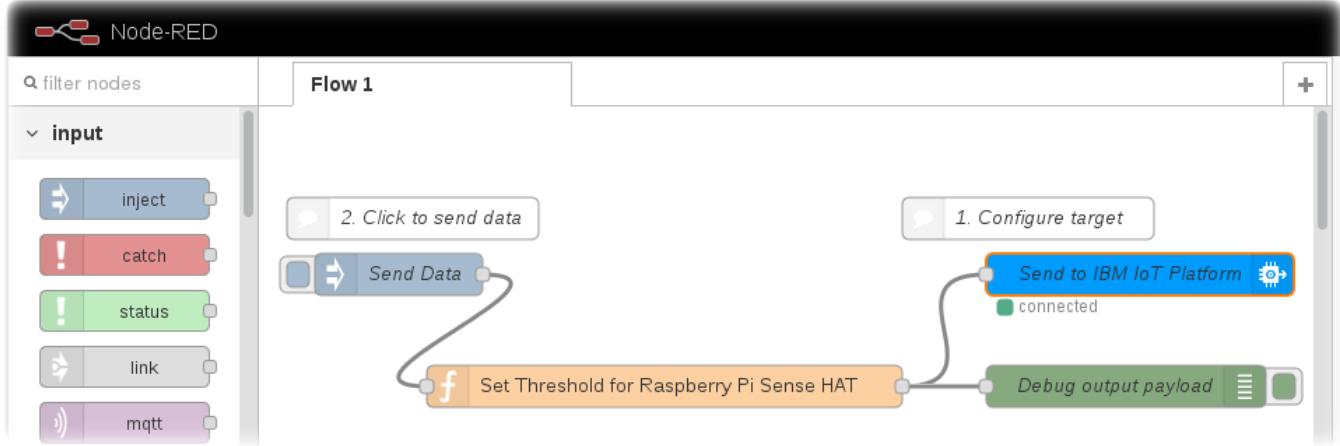
```
// Set the temperature threshold that will trigger
// a colored alert on the SenseHat LED display.
var tempthresh = 35;

// Create MQTT message in JSON
msg = {
  payload: JSON.stringify(
    {
      d:{
        "tempthresh" : tempthresh
      }
    }
  );
return msg;
```

This flow is available at <https://github.com/johnwalicki/WorldofWatson-IoT-Lab>

B. The **Send to IBM IoT Platform** node needs to be configured to send the data to your Quickstart Id. Click on the Send to IBM IoT Platform node and paste in the **Quickstart Device Id** from Lesson 2 and Lesson 4.

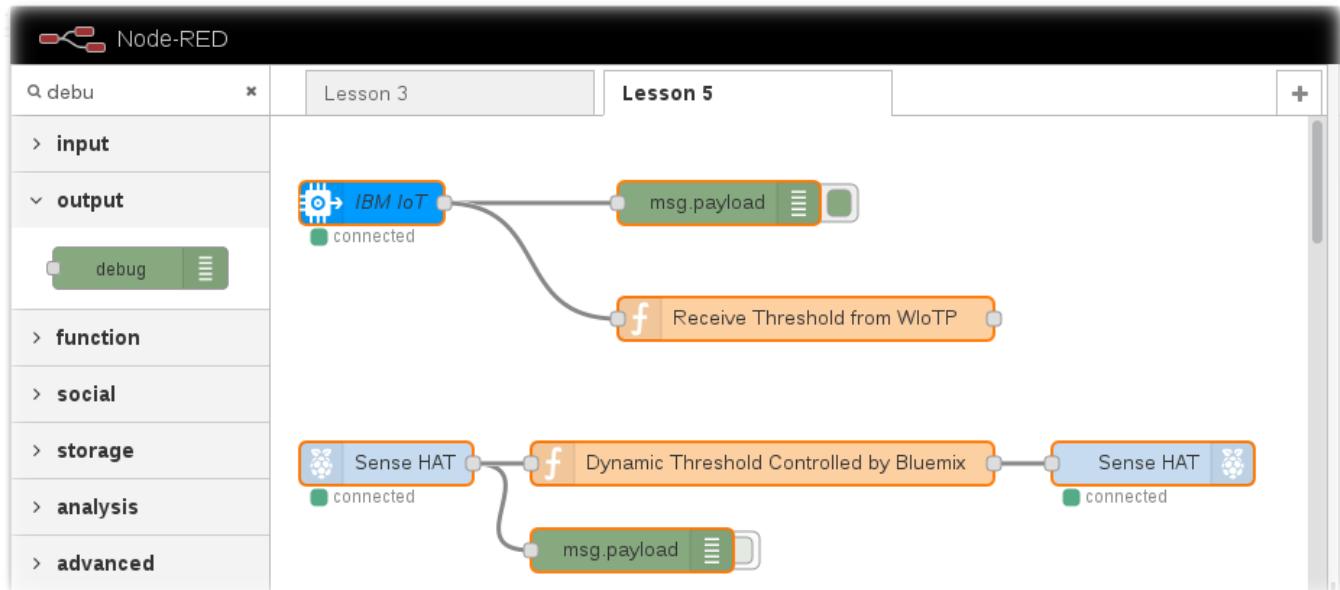
The new flow might look like this:



Remember to Deploy this flow.

2. On the local Node-RED browser tab, a new Lesson 5 flow will be added. It receives MQTT messages from Bluemix and saves the temperature threshold in a Node-RED context object. Any variables inside the node are lost every time the code is run (whenever the node gets a message input). In order to preserve the variables across iterations, the context object is used. There are also contexts for the entire flow and even across flows. Think of them as global variables. For more info see: <http://nodered.org/docs/creating-nodes/context>

The 2<sup>nd</sup> part of the Lesson 5 flow will look similar to Lesson 3 flow. It reads the environmental temperature from the SenseHAT but instead of a switch statement that decides whether to light the LED display in RED / GREEN, there is a function which compares the temperature to the global threshold variable received from Bluemix. Create a new flow (press the + sign) and Menu → Import → Clipboard paste the following page into Node-RED



If your LED display starts blinking RED / GREEN, it is possible you didn't disable part of the Lesson 3 flow. Turn to that flow and delete the output wire to the SenseHat node. Don't delete the entire flow however. You still want to send environmental sensor data to Watson IoT Platform.

## Lesson 5 Local Node-RED flow :

This flow is available at <https://github.com/johnwalicki/WorldofWatson-IoT-Lab>

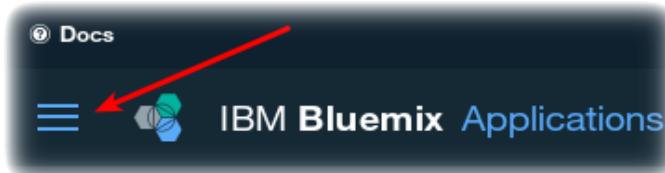
```
[{"id":"77c4e223.22c9cc","type":"ibmiot  
in","z":"56e5b0e4.4f7e3","authentication":"quickstart","apiKey":"","inputType":"evt","deviceID":"sampleDeviceID","applicationId":"","deviceType":"+","eventType":"+","commandType":","", "format":"json","name":"IBM  
IoT","service":"quickstart","allDevices":","", "allApplications":","", "allDeviceTypes":true,"allEvents":true,"allCommands":","", "allFormats":","", "qos":0,"x":70,"y":60,"wires":  
[["84207463.6d2f7","e333bf4e.c5cb28"]]},  
{"id":"84207463.6d2f7","type":"debug","z":"56e5b0e4.4f7e3","name":"","active":true,"console":false,"complete":false,"x":310,"y":60,"wires":[]},  
{"id":"7acef8e.e779a88","type":"rpi-sensehat  
in","z":"56e5b0e4.4f7e3","name":"","motion":false,"env":true,"stick":false,"x":80,"y":240,"wires":[[{"6cacb1f7.6fea78","faba8af6.7b1668"}]}, {"id":"88303e7c.0db3f8","type":"rpi-  
sensehat out","z":"56e5b0e4.4f7e3","name":"","x":610,"y":240,"wires":[]},  
 {"id":"faba8af6.7b1668","type":"function","z":"56e5b0e4.4f7e3","name":"Dynamic  
Threshold Controlled by Bluemix","func":"threshold = 0;\nthreshold =  
global.get('threshold');\n\nif ( msg.payload.temperature > threshold ) {\n  msg.payload  
= '*,*;red';\n} else if (msg.payload.temperature <= threshold){\n  msg.payload  
= '*,*;green';\n}\nreturn msg;","outputs":"1","noerr":0,"x":340,"y":240,"wires":  
[["88303e7c.0db3f8"]]},  
 {"id":"e333bf4e.c5cb28","type":"function","z":"56e5b0e4.4f7e3","name":"Receive  
Threshold from WIoTP","func":"global.set('threshold',msg.payload.d.temphresh);\nreturn  
msg;","outputs":1,"noerr":0,"x":370,"y":140,"wires":[]},  
 {"id":"6cacb1f7.6fea78","type":"debug","z":"56e5b0e4.4f7e3","name":"","active":false,"console":false,"complete":false,"x":250,"y":300,"wires":[]}]
```

# Lesson 7 Registered Watson IoT Platform Devices

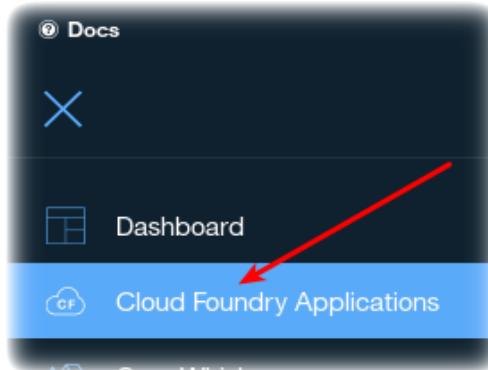
The IBM Watson IoT Platform is a fully managed, cloud-hosted service that makes it simple to derive value from Internet of Things (IoT) devices. When combined with the IBM Bluemix platform, Watson IoT provides simple, but powerful application access to IoT devices and data. You can rapidly compose analytics applications, visualization dashboards and mobile IoT apps. Create IoT applications that feed insights to your backend enterprise applications.

This section shows how one can setup an Watson IoT Organization and register devices in it. When you created the “Internet of Things Platform” Starter application in IBM Bluemix, a new Organization was created for you. This IBM Watson IoT Platform organization is a space used for connecting and managing devices to the IBM Watson IoT Platform, so that your applications can access their live and historical data.

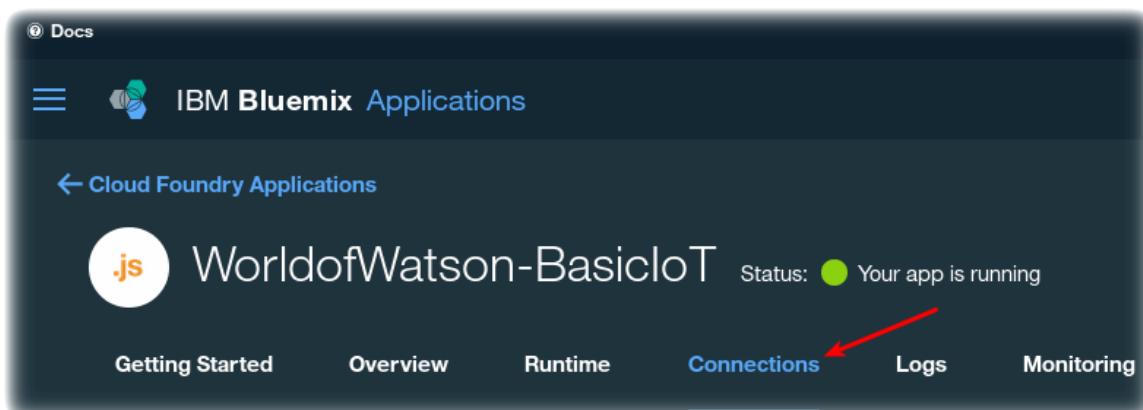
1. Return to your Bluemix browser tab – <http://bluemix.net>
2. Click on the top left menu



3. Click on **Cloud Foundry Applications**



4. From the list of Cloud Foundry Applications, click on the left side of your Bluemix application to enter
5. Turn to the **Connections** tab



6. Find the Internet of Things Platform tile and click on it

The screenshot shows the IBM Bluemix Applications interface. At the top, there's a navigation bar with 'Docs' and the 'IBM Bluemix Applications' logo. Below that is a breadcrumb navigation '← Cloud Foundry Applications'. The main area displays two application tiles. The first tile is for 'WorldofWatson-BasicIoT' (Cloudant NoSQL DB Lite) and the second tile is for 'WorldofWatson-BasicIoT' (Internet of Things Platform iotf-service-standard). The second tile is highlighted with a blue border around its icon and title. Below each tile are 'View Credentials' and 'Docs' buttons. The 'Connections' tab is currently selected, indicated by a blue underline.

7. Scroll down underneath the **Connect your devices** and press the **Launch dashboard** button.

The screenshot shows the 'IBM Bluemix Internet of Things' dashboard. At the top, there's a header with 'Docs' and the 'IBM Bluemix Internet of Things' logo. Below the header is a large circular icon containing a stylized chip or microprocessor symbol. To the right of the icon, the text 'Connect your devices' is displayed in a large, bold font. Underneath this, there's a paragraph of text: 'Use our [recipes](#) to find out how to add your devices. We work with partners and have sample connection recipes for many devices.' Further down, another paragraph of text reads: 'Launch the Watson IoT Platform dashboard and add your devices by clicking the 'Add Device' button under the 'Devices' tab.' At the bottom of the page is a prominent 'Launch dashboard' button.

9. Click on the Devices chip logo.



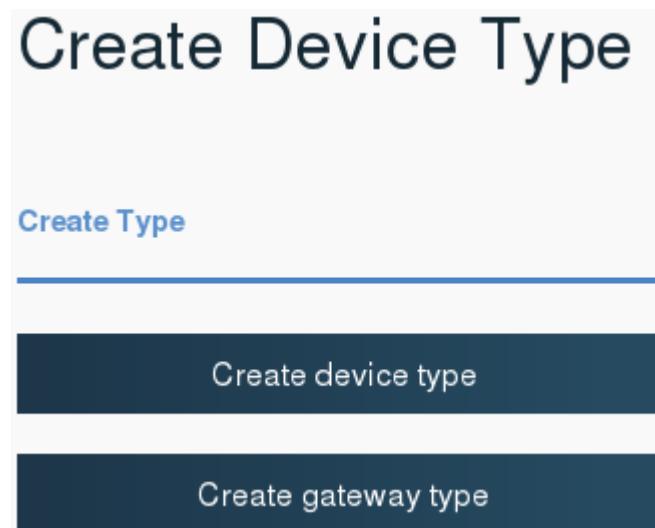
10. Click on the + Add Device



11. Click on the Create device type

A dark blue rectangular button with the white text "Create device type".

12. Create a gateway type



13. Name your gateway “RaspberryPi”

## Create Gateway Type

### General Information

Name

RaspberryPi

14. Press the **Next** button 3 times

15. Press the **Create** button

16. Now finally Add Device by pressing the **Next** button

17. Enter a Device ID name

## Add Device

### Device Info

Device ID is the only required information, however other fields are populated according to the attributes set in the selected device type. These values can be overridden, and attributes not set in the device type can be added.

Device ID

WoWSenseHat

18. Press the **Next** button twice

19. **Provide a token** that you will remember

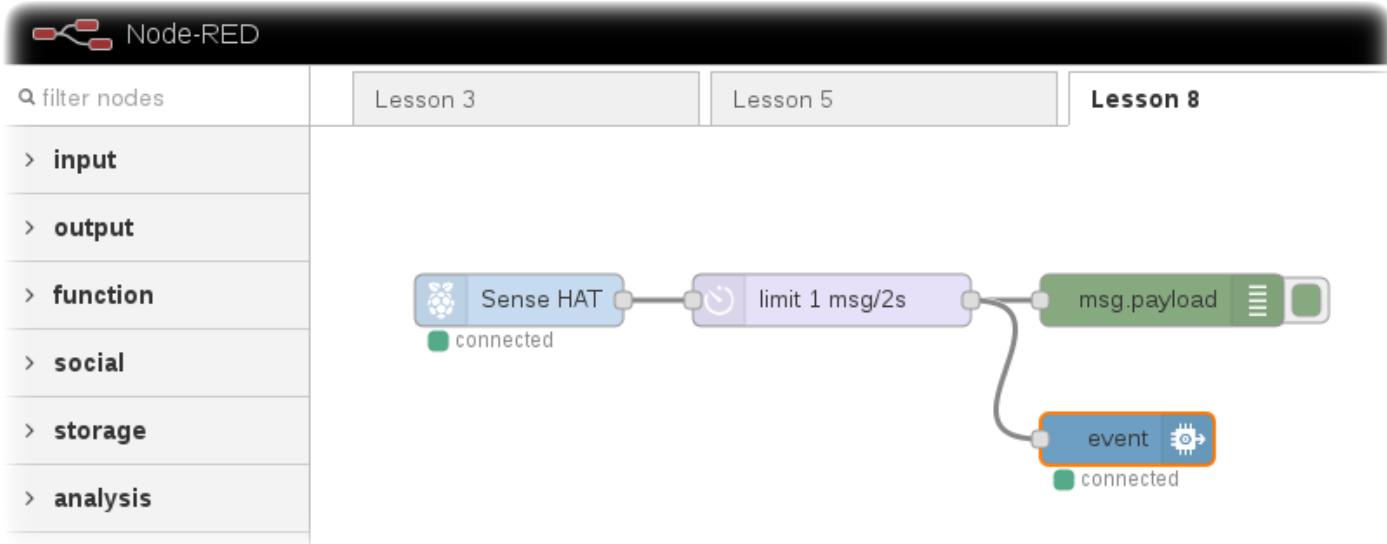
20. Press the **Next** button

21. Press the **Add** button

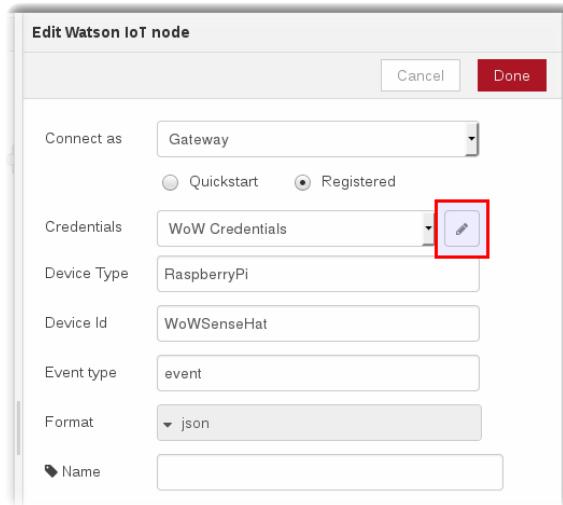
22. Take a screenshot of the **Device Credentials** summary page. **It is the only time that you will see the token.**

# Lesson 8 - Send SenseHAT Acceleration data to Watson IoT

In this section, the Acceleration data from the SenseHAT will be sent to the Watson IoT Platform registered device that was created in Lesson 7. Return to the local Node-RED browser tab and create a new flow by pressing the + sign. Name this flow “Lesson 8”



1. Drag a **SenseHat** node from the palette. Double click on it and only select the **Motion events** data.
2. There will be rapid motion data from the SenseHAT so limit the amount of data that is sent by adding a **delay** node. Set it to Limit to 1 msg per 2 seconds and check the “drop intermediate messages”
3. Drag a **Debug** node to the flow.
4. Press the **Deploy** button to observe the SenseHat data
5. Drag a **Watson IoT** output node from the palette onto your flow
6. Double Click on the Watson IoT Node to configure it with your WIoTP credentials
7. Enter all of the information from Lesson 7



8. Click on the Credentials pencil to enter your authentication token
9. Deploy the Flow

## Visualizing Bluemix Watson IoT Application Sensor data

There are several ways to visualize the acceleration data that is being sent from the Raspberry Pi SenseHAT to Watson IoT Platform. WIoTP has preconfigured dashboards called Boards that display Cards of the data.

1. Return to the Watson IoT Platform dashboard
2. You will find the Boards on the top of the list.
3. Click on the DEVICE-CENTRIC ANALYTICS card

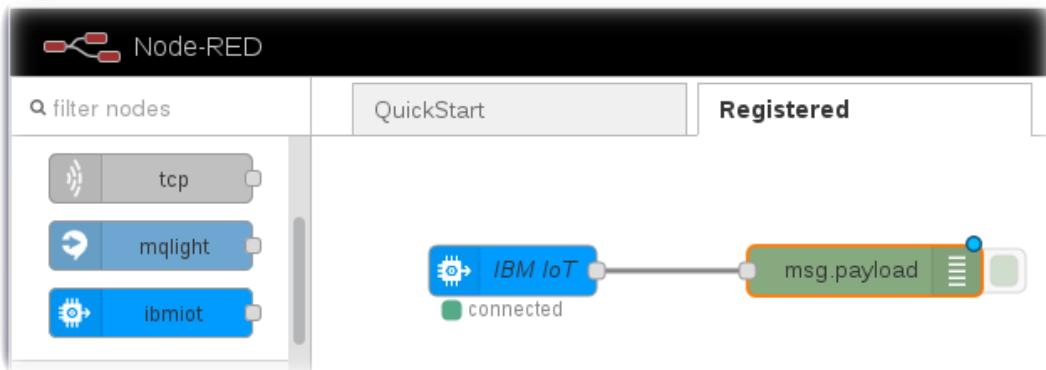
The screenshot shows the Watson IoT Platform interface. On the left is a sidebar with various icons. The main area features a title 'Device-Centric Analytics' with a circled '1'. Below it is a card titled 'Devices I Care About' showing one device: 'WoWSenseHat' (RaspberryPi). To the right are two cards: 'Device Info' (circled '2') and 'Device Properties' (circled '3'). The 'Device Info' card displays device details like name, type, client ID (g:kx7och:RaspberryPi:WoWSenseHat), creator, and creation time. The 'Device Properties' card shows streaming sensor data for 'acceleration.x' at '0.3067'.

4. You will see Device Info and Device Properties with streaming acceleration data
5. Return to your Bluemix Node-RED browser tab

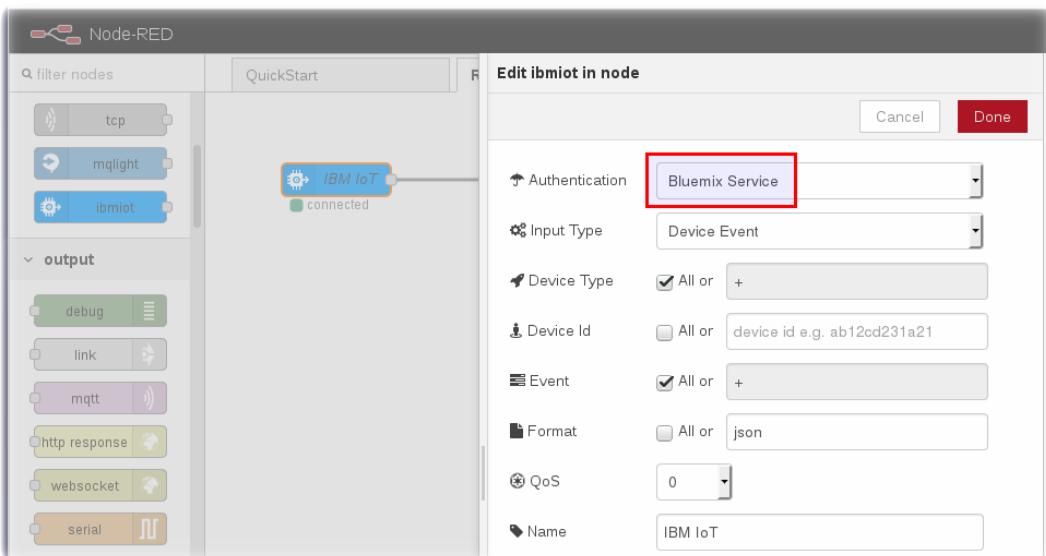
## Node-RED Bluemix Watson IoT App Acceleration Sensor data

In this section of Lesson 8 you will create a Node-RED flow on your Bluemix Watson IoT Application which receives motion sensor data from the Raspberry Pi gateway and SenseHAT sensor device. If the x / y / z acceleration exceeds a warning threshold, tweet and text the alert.

1. Create a new flow by pressing the + in the upper right corner of the Node-RED visual programming editor.
2. Name the flow “Registered” by double clicking on the tab. Name the first flow “Quickstart” by double clicking on that tab.
3. Drag a **IBM IoT input** node onto the canvas
4. Drag a **Debug** output node onto the canvas
5. Wire the two nodes together



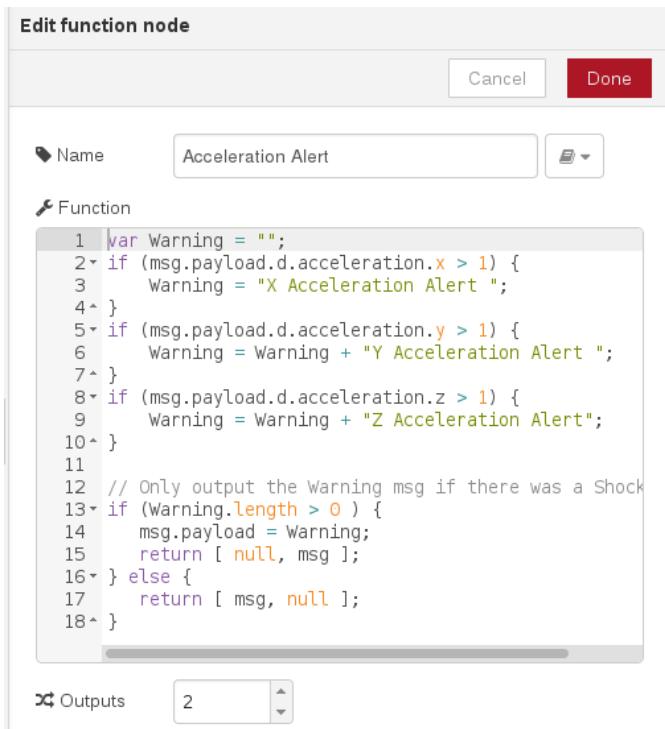
6. Double Click on the IBM IoT node. Once you select the Authentication as **Bluemix Service** it already knows your credentials.



7. Click on to save and deploy your changes.
8. You should see acceleration data from the SenseHAT in the Debug tab.

9. Next we will create a function that tests for excessive acceleration in the X / Y / Z coordinates.

10. Add a  node as shown below.

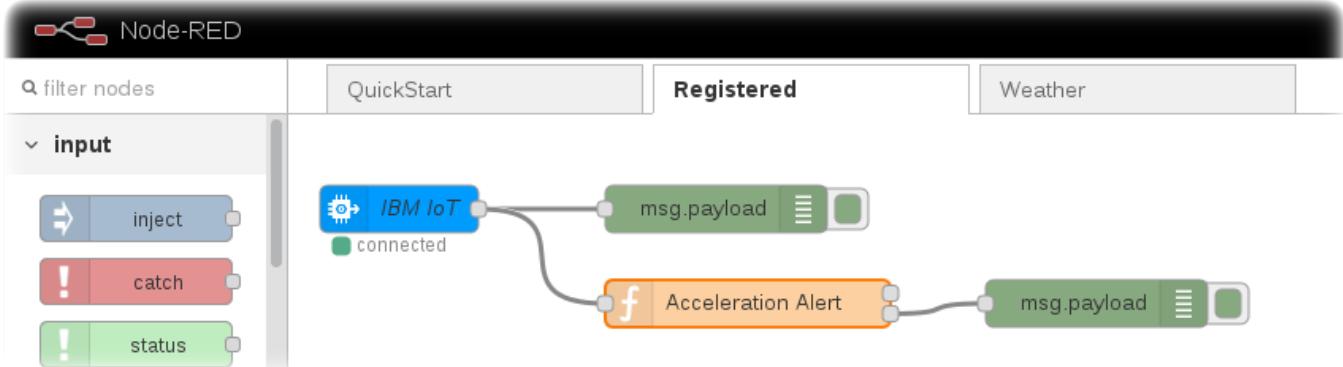


```
var Warning = "";
if (msg.payload.d.acceleration.x > 1) {
    Warning = "X Acceleration Alert ";
}
if (msg.payload.d.acceleration.y > 1) {
    Warning = Warning + "Y Acceleration Alert ";
}
if (msg.payload.d.acceleration.z > 1) {
    Warning = Warning + "Z Acceleration Alert";
}

// Only output the Warning msg if there was a Shock
if (Warning.length > 0 ) {
    msg.payload = Warning;
    return [ null, msg ];
} else {
    return [ msg, null ];
}
```

```
var Warning = "";
if (msg.payload.d.acceleration.x > 1) {
    Warning = "X Acceleration Alert ";
}
if (msg.payload.d.acceleration.y > 1) {
    Warning = Warning + "Y Acceleration Alert ";
}
if (msg.payload.d.acceleration.z > 1) {
    Warning = Warning + "Z Acceleration Alert";
}

// Only output the Warning msg if there was a Shock
if (Warning.length > 0 ) {
    msg.payload = Warning;
    return [ null, msg ];
} else {
    return [ msg, null ];
}
```

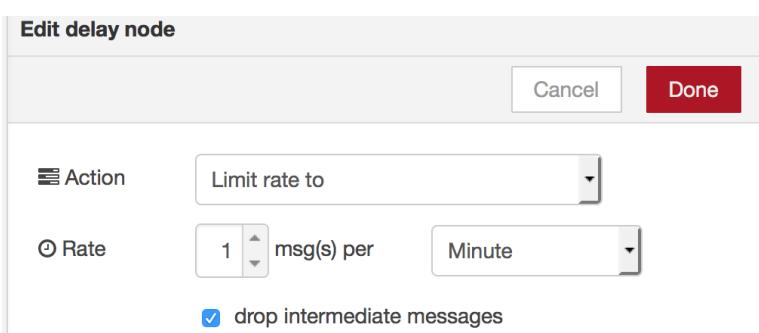


These flows are available at <https://github.com/johnwalicki/WorldofWatson-IoT-Lab>

# Lesson 9 - Connect to Twitter and Tweet Acceleration Shocks

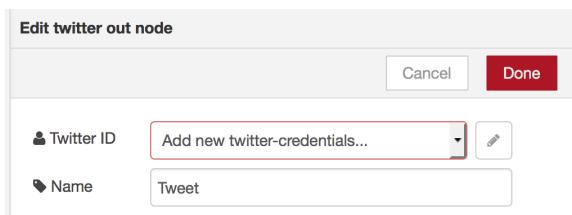
In this section, we will connect a Twitter account and use the Twitter account to tweet when the acceleration data from the Raspberry Pi SenseHAT accelerometer reaches a shock threshold. This section is optional and may be skipped.

1. Sign up for a Twitter account at <http://twitter.com> . If you already have a Twitter account, proceed to step 2.
2. Add a  node as shown below.



The delay node limits how often the flow is run. Since the accelerometer can be set off often, without this node, a tweet would be sent every second. With this node limiting messages to once a minute, the Twitter node will send a tweet once a minute dropping any additional messages during the 60 second timeframe.

3. Add a  node. Click on the pencil button and authenticate with Twitter. The account you sign in with will be used to send tweets.





## Authorize Node RED to use your account?

[Authorize app](#)

[Cancel](#)



Node RED

nodered.org

Node-RED Twitter node

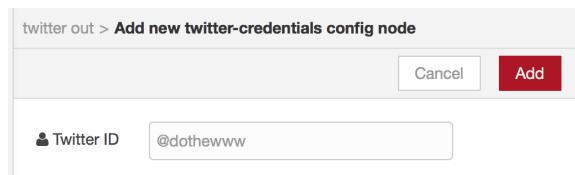
### This application will be able to:

- Read Tweets from your timeline.
- See who you follow, and follow new people.
- Update your profile.
- Post Tweets for you.
- Access your direct messages.

### Will not be able to:

- See your Twitter password.

Return back to the node configuration. The settings for the Twitter node should have your username set. Click **Add**.



4. Connect the nodes as shown below.

- Get the code:  
<https://github.com/johnwalicki/WorldofWatson-IoT-Lab>
5. Click on Deploy to save and deploy the changes.
  6. Shake your Raspberry Pi / SenseHat. Be careful not to unplug it from the microUSB power.
  7. Since the acceleration is a shock, the Alert function will compose a message, and pass it to the Twitter node. The Twitter node uses this message as the content for the tweet.
  8. Visit the Twitter timeline for the user you authenticated with and verify the message has been tweeted.

Tweets	Tweets & replies	Media
	John Walicki @johnwalicki · 1m X Acceleration Alert Z Acceleration Alert	
	John Walicki @johnwalicki · 7m Z Acceleration Alert	

# Lesson 10 - Use Twilio to Text Acceleration Shocks

In this section, we will connect a Twilio phone number to the application and send a text message notification when an acceleration shock occurs. This section is optional and may be skipped.

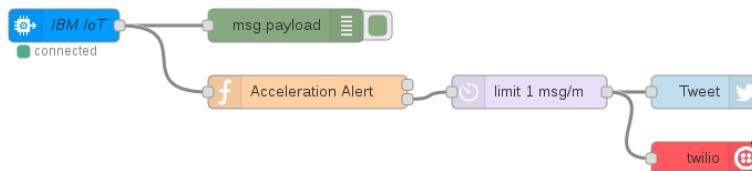
1. Sign up for a Twilio account at <http://twilio.com>. If you already have a Twilio account, sign in. Create a Twilio phone number that will be used in step #3.
2. In the Console Dashboard, click on the lock icon next to Auth Token. Copy the Account SID and Auth Token.

The screenshot shows the Twilio Console Beta dashboard. The top navigation bar includes 'CONSOLE BETA' and 'GIVE FEEDBACK'. The main area has a 'Console Dashboard' header. On the left, there's a sidebar with 'Home', 'Dashboard', 'Account', 'Billing', 'Logs', 'Usage', and 'What's New'. The 'Account Summary' section displays 'ACCOUNT SID' (AC12345ab67890cd12efg34567890hi1234), 'AUTH TOKEN' (a12345c67d8ef980123456789012345g), and a 'BALANCE' of '+ \$6.632'. A note says 'Auto Recharge is OFF'. Below this are 'Account Details' and 'Recently Used Products' sections for Programmable SMS, Phone Numbers, Programmable Voice, and Add-ons.

3. Add a node. Click on the Pencil to provide your **Account SID** and **Auth Token** from step #2, and **From** phone number using the Twilio phone number from step #1. Fill in the **SMS to** textbox with a phone number that will be texted to.

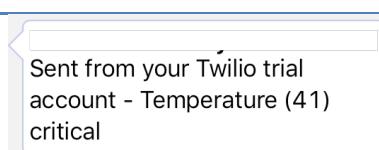
The screenshot shows the Node-RED editor with a node titled 'twilio out > Edit twilio-api node'. It has four input fields: 'Account SID' (AC12345ab67890cd12efg34567890hi1234), 'From' (15555555555), 'Token' (\*\*\*\*\*), and 'Name' (Name). There are 'Delete', 'Cancel', and 'Update' buttons at the top right.

4. Connect the nodes together as shown.



Get the code:  
<https://github.com/johnwalicki/WorldofWatson-IoT-Lab>

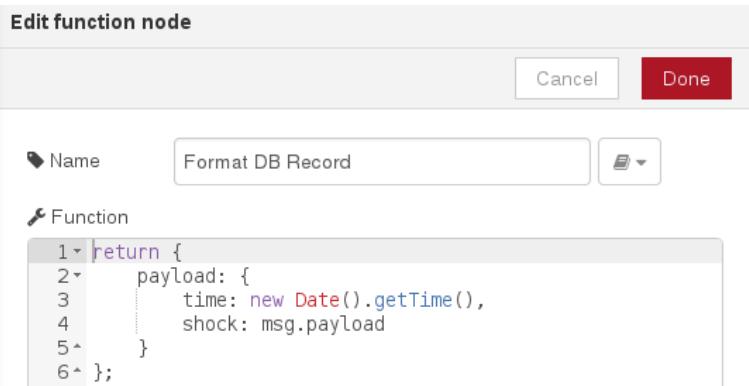
5. Click on to save and deploy the changes.
6. Shake your Raspberry Pi. You should receive a text message shortly.

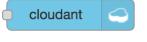


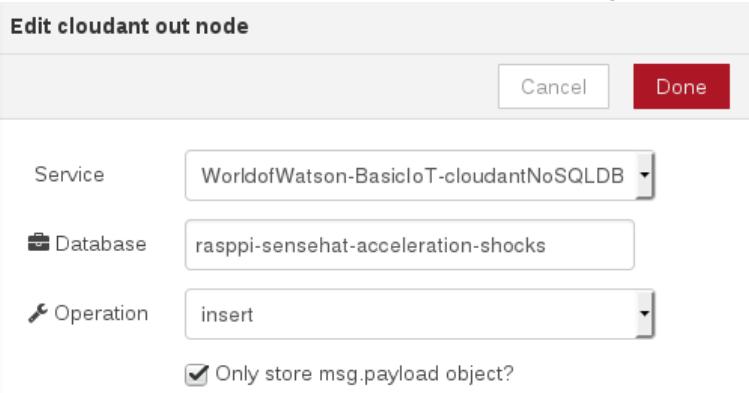
# Lesson 11 - Store Acceleration Shock Events in a Cloudant NoSQL Database

This section will show how to add a Cloudant NoSQL database to the Node-RED application and store acceleration shock reports (one per minute). This functionality can be useful to run historical analysis (outside the scope of this lab) or find patterns over time. This section is optional and can be skipped. However, it is a prerequisite for the **Retrieve Acceleration Shocks From Cloudant NoSQL Database** section (below).

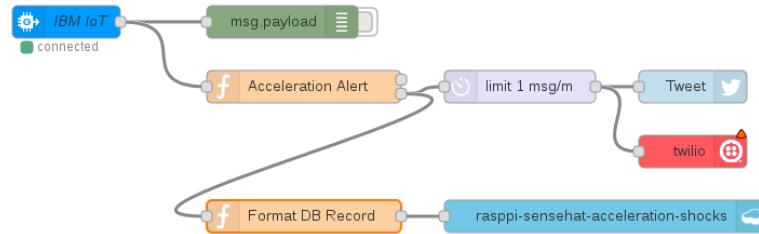
1. Add a  node as shown below.



2. Add a  node as shown below. Name your database using lowercase characters.

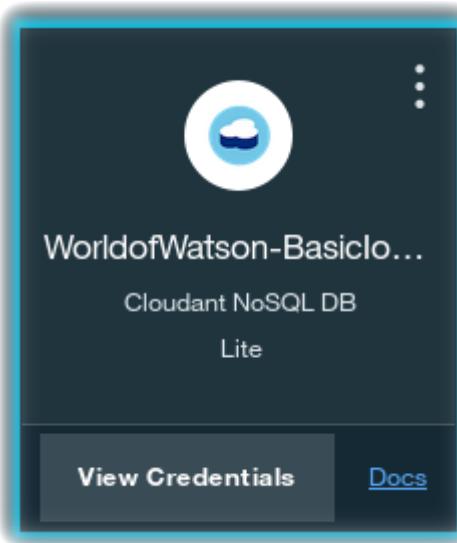


3. Connect the nodes together as shown below.

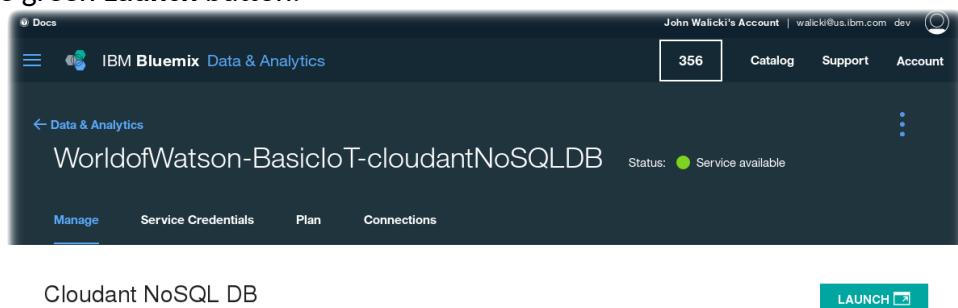


**Get the Code:**  
<https://github.com/johnwalicki/WorldofWatson-IoT-Lab>

4. Click on Deploy to save and deploy your changes.
5. Go back to the IBM Bluemix application dashboard and the **Connections** tab. Click on the **Cloudant NoSQL DB** service tile.



6. Click on the green **Launch** button.



7. This is the Cloudant NoSQL database dashboard. A list of databases is displayed. The database named raspi-sensehat-acceleration-shocks contains documents representing each extensive acceleration shock event that has been stored by the Node-RED application. Click on the database named **raspi-sensehat-acceleration-shocks**

The screenshot shows the Cloudant NoSQL Database dashboard. On the left is a sidebar with links: Usage, Databases (selected), Replication, Analytics, Active Tasks, Account, Support, Documentation, and IBM Cloudant. The main area is titled "Databases" with a search bar for "Database name". It lists "Your Databases" with columns: Name, Size, # of Docs, and Actions. There are two entries: "nodered" (9.4 KB, 4 docs) and "raspi-sensehat-acceleration-shocks" (1.4 KB, 10 docs). The "Actions" column for each entry includes icons for replication, locking, and deleting.

8. To see the expanded view of the documents, click on **Query**, check the box to include **Docs**.

The screenshot shows the "Query" results for the "raspi-sensehat-acceleration-shocks" database. The sidebar on the left is identical to the previous dashboard. The main area shows a table of documents with columns: Metadata, shock, and time. The table lists ten documents, each with a preview icon, a document ID, and its details. The "shock" column shows values like "X Acceleration Alert Z ...", "Y Acceleration Alert", and "Z Acceleration Alert". The "time" column shows timestamps such as "1477106930735" and "1477106925578". The interface includes buttons for JSON, Table, Docs (which is checked), Options, API, and a bell icon.

9. Each box represents one document (in our case one acceleration shock event) that contains the payload (time and coordinate shock) we stored earlier.

The screenshot shows a detailed view of a single document in the "raspi-sensehat-acceleration-shocks" database. The sidebar is the same as before. The main area shows a "Save Changes" button and a "Cancel" button. Below is a code editor displaying the document's payload in JSON format:

```

1: {
2:   "_id": "35efec938a1ae96552d300f6653ecfb0",
3:   "_rev": "1-40814299f43d49d66f2e130f8a831a41",
4:   "time": 1477106930735,
5:   "shock": "X Acceleration Alert Z Acceleration Alert"
6: }

```

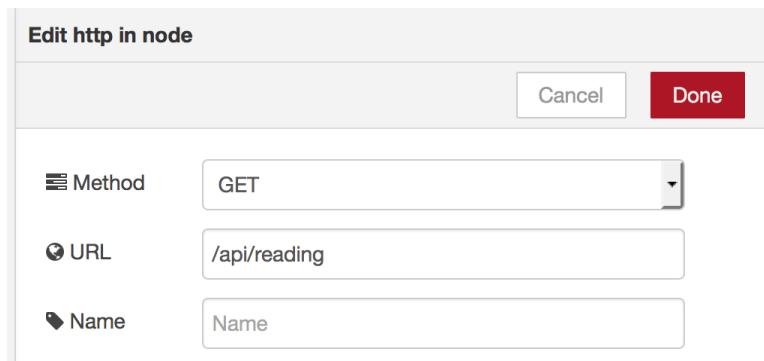
Other buttons in the top right include "Upload Attachment", "Clone Document", and "Delete".

# Lesson 12 - Retrieve Acceleration Shocks from a Cloudant NoSQL DB

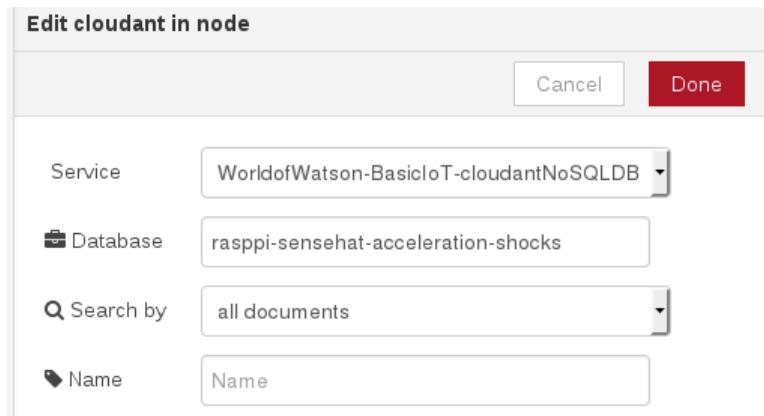
This section shows how to retrieve acceleration shock event data from a Cloudant NoSQL database and exposes it as a HTTP endpoint. This functionality can be useful to run historical analysis (outside of the scope of this lab) or find usage patterns over time. This section is optional and can be skipped. Completion of the section titled **Store Acceleration Shocks in a Cloudant NoSQL Database** is required before beginning this section.

Now that we have a Cloudant NoSQL database containing reported shocks, let's expose the data as an HTTP endpoint.

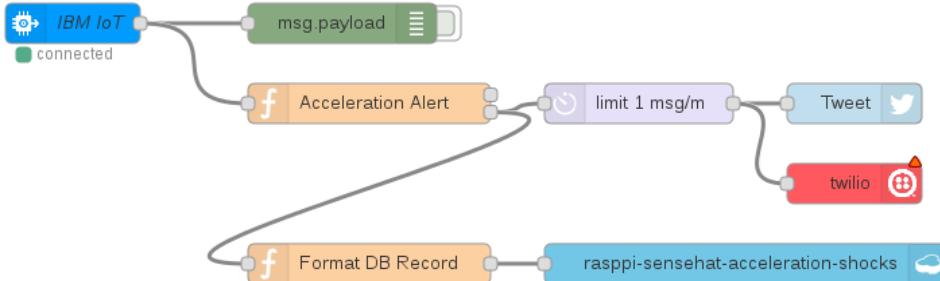
1. Add a  node as shown below.



2. Add a  node as show below.



3. Finally, add a  node.
4. Connect the nodes together as shown below.



Get the Code:

<https://github.com/johnwalicki/WorldofWatson-IoT-Lab>



5. Click on Deploy to save and deploy your changes.
6. Open a browser tab and visit your application's URL, appended by /api/reading. If you chose **myapp** when setting up your application, the URL would be:

<https://myapp.mybluemix.net/api/reading>

You should see data returned similar to the following:

```

[{"_id": "35efec938a1ae96552d300f6653ecfb0", "_rev": "1-40814299f43d49d66f2e130f8a831a41", "time": 1477106930735, "shock": "X Acceleration Alert Z Acceleration Alert"}, {"_id": "3d869832c694f1dfecce9482e4990d15", "_rev": "1-aef9f16bc05acccbd119d30e089290aca", "time": 1477106925578, "shock": "X Acceleration Alert"}, {"_id": "55ef4497cbafaf776ebafad113e6514c", "_rev": "1-9b49eb8dc86b6353e7ea83e3934dca2", "time": 1477106912136, "shock": "X Acceleration Alert Y Acceleration Alert Z Acceleration Alert"}, {"_id": "5edbea59192e37572c467172fdbd62a2", "_rev": "1-40da923607ca9e49045a39fb683d2d62", "time": 1477106927634, "shock": "Z Acceleration Alert"}, {"_id": "8b0aebb3d039d5e0f0e743c33c62813a", "_rev": "1-8f19696d41109f8f8fa28e538e546a07", "time": 1477106910066, "shock": "Y Acceleration Alert Z Acceleration Alert"}, {"_id": "9350d0517e2a229a8df260152d23c3c1", "_rev": "1-8376d127b37278e962641f667157d81", "time": 1477106906967, "shock": "Y Acceleration Alert"}, {"_id": "deb27c3e40b8bbd9777e3742739cef20", "_rev": "1-596a189819fce35bfcdea8b8c5947bd5", "time": 1477106909028, "shock": "Z Acceleration Alert"}, {"_id": "deb27c3e40b8bbd9777e374273a133ea", "_rev": "1-28f83fb0f203e55c62359a7bbe49e327", "time": 1477106904911, "shock": "Z Acceleration Alert"}, {"_id": "deb2/c3e40b8bbd9777e374273a15453", "_rev": "1-615908/15/3aa128fae/609ae1ba454", "time": 1477106931782, "shock": "X Acceleration Alert"}, {"_id": "deb2/c3e40b8bbd9777e374273a15453", "_rev": "1-615908/15/3aa128fae/609ae1ba454", "time": 1477106932809, "shock": "Z Acceleration Alert"}]

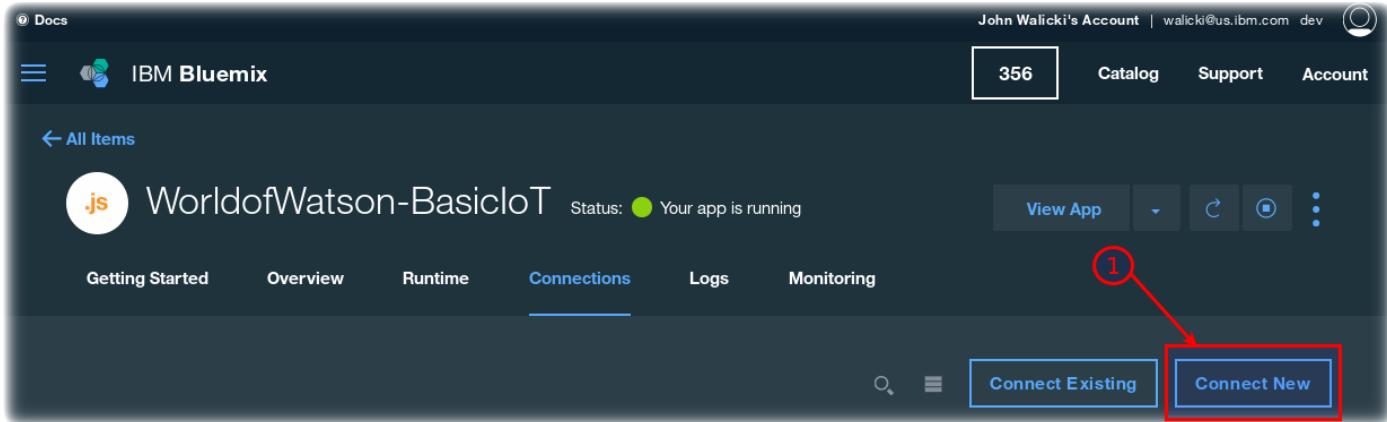
```

The shock events values are returned in a JSON array. You can use this data in web applications or analytical tools. As you inject more data in the IoT example in Node-RED, this dataset will expand to include that data.

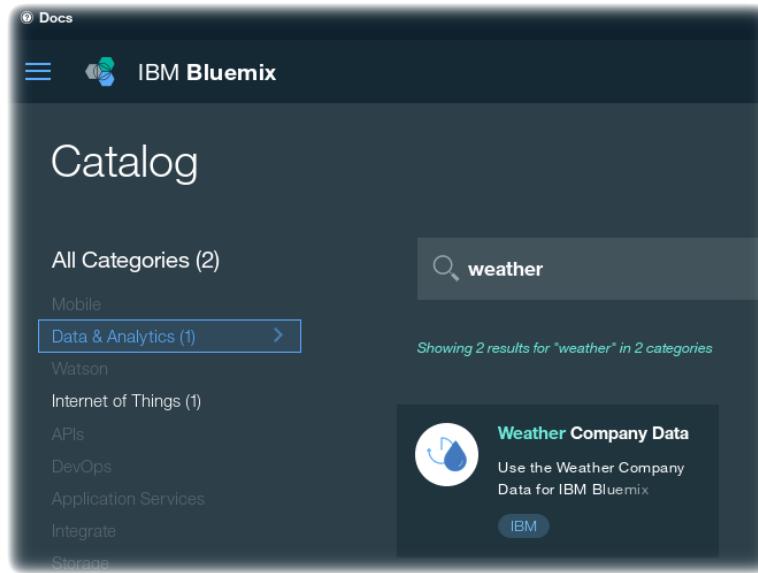
# Lesson 13 - Send Weather Alerts to a Raspberry Pi

In this section, we will display the temperature of your favorite home town. The outside temperature will be retrieved via the **Weather Company Data for IBM Bluemix** service.

1. Return to the **Connections** tab of your Internet of Things Starter Application
2. Click on the **Connect New** button (1) of your Node-RED application in IBM Bluemix.



3. Select the **Weather Company Data for IBM Bluemix** under the Data & Analytics category.



4. Click **Create** to add the service to your application.

- When prompted to restage the application, click **Restage** to restart the application and update the environment with the credentials to the Weather service.

IBM Bluemix Applications

← Cloud Foundry Applications

WorldofWatson-BasicIoT Status: C Your app is restaging

Getting Started Overview Runtime Connections Logs Monitoring

Weather Company Data... Weather Company Data Free-v2

View Credentials Docs

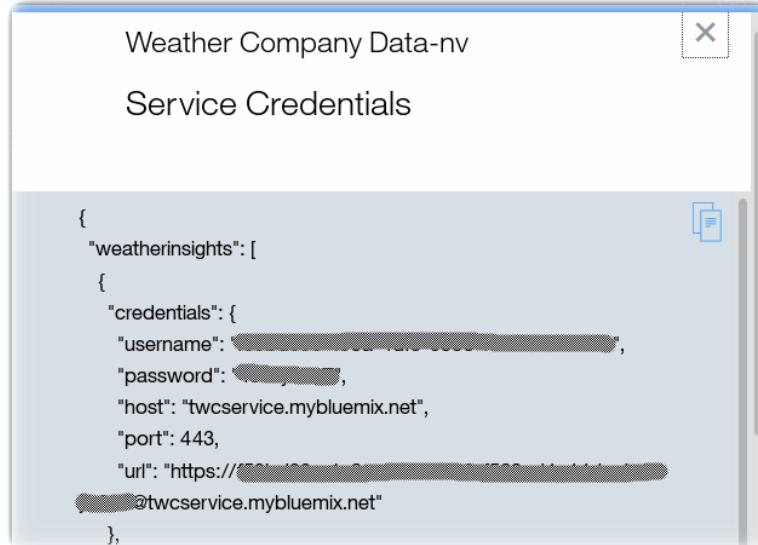
WorldofWatson-BasicIoT Cloudant NoSQL DB Lite

View Credentials Docs

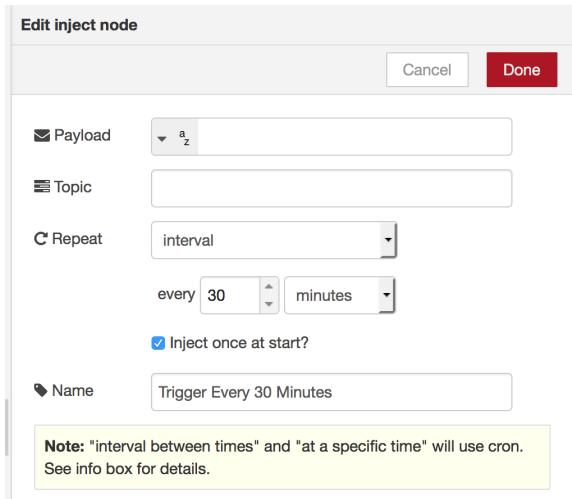
WorldofWatson-BasicIoT Internet of Things Platform iotf-service-standard

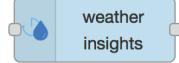
View Credentials Docs

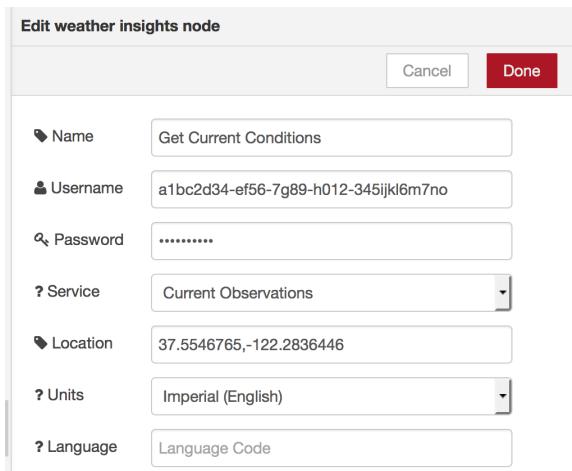
- When the application has restarted, a third service tile will appear in the **Connections** tab. Click on **View Credentials** in the Weather Company Data service tile to show the service credentials.
- Copy the username and password values and save them for use in step #10.



8. Return to your Node-RED application flows on Bluemix and add a new flow tab by clicking the +. Name the flow **Weather**
9. Add a  node as shown below. This will trigger the flow once at startup, and then every 30 minutes.

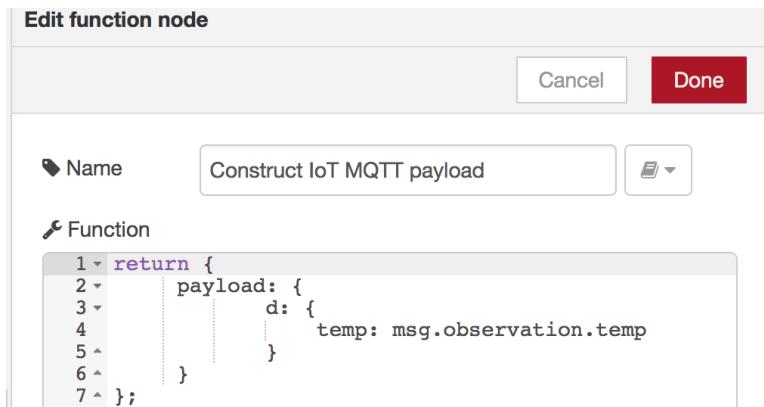


10. Add a  node as shown below. Use the Weather Company Data API credentials from step #7.



Look up the Geocode of your home town and enter the Longitude,Latitude in the Location field. Do not enter any spaces. Note the - minus symbol / negative sign if you live in the western hemisphere.

11. Add a  node as shown below.



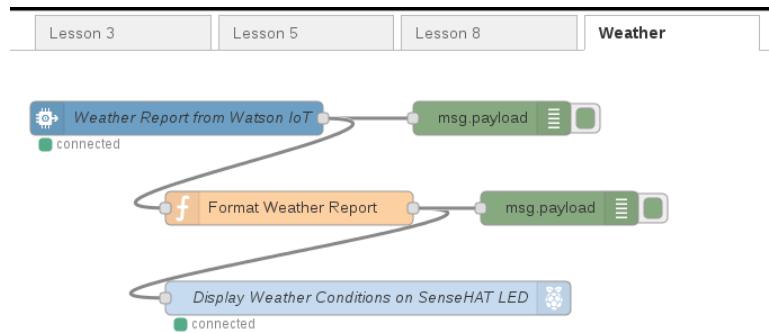
This will format the temperature in a MQTT message payload, for use in step #11. If the weather condition changes every 30 minutes, the value will be sent to the Raspberry Pi for display on the SenseHAT LED

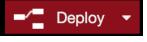
12. On IBM Bluemix, connect the nodes together as shown below.



Get the Code:  
<https://github.com/johnwalicki/WorldofWatson-IoT-Lab>

13. On the Raspberry Pi, connect the nodes together as shown.



14. Click on  to save and deploy your changes.

15. You will need to **disable** the SenseHAT output nodes created in Lesson 3 and Lesson 5 so that those flows do not overwrite the LED. The LED should display the outside temperature and will update the temperature from the Weather service every 5 minutes.

# Additional Raspberry Pi / SenseHAT Resources

---

In this final section, we will challenge the workshop attendees to implement additional SenseHAT projects.

There are many Raspberry Pi Recipes available at:

<https://developer.ibm.com/recipes>

An excellent Marble Maze project is great fun:

## **Connecting a Sense HAT to Watson IoT using Node-RED**

<https://developer.ibm.com/recipes/tutorials/connecting-a-sense-hat-to-watson-iot-using-node-red/>