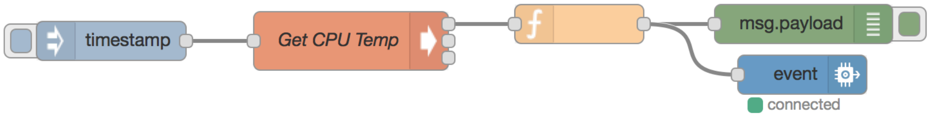
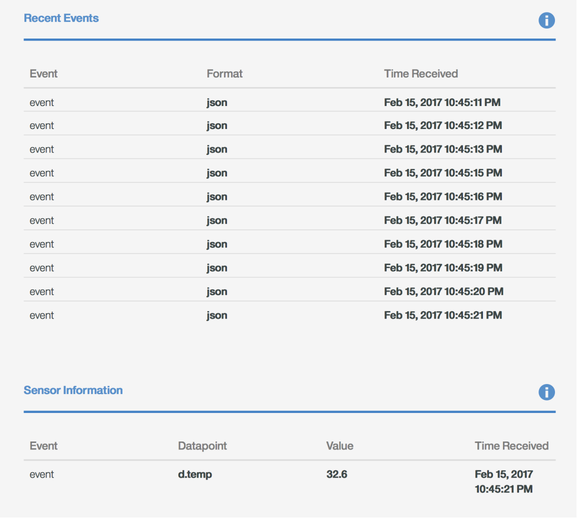
# Getting Started with Watson IoT and Raspberry Pi Temperature Sensor

Author: JeanCarl Bisson | jbisson@us.ibm.com | @dothewww



*Use Node-RED to read a temperature sensor on the Raspberry Pi (see pg. 2)*



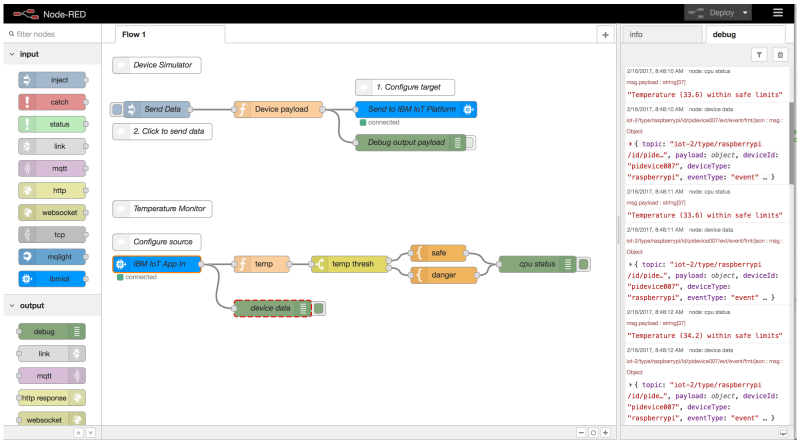
Recent Events

| Event | Format | Time Received |
|---|---|---|
| event | json | Feb 15, 2017 10:45:11 PM |
| event | json | Feb 15, 2017 10:45:12 PM |
| event | json | Feb 15, 2017 10:45:13 PM |
| event | json | Feb 15, 2017 10:45:15 PM |
| event | json | Feb 15, 2017 10:45:16 PM |
| event | json | Feb 15, 2017 10:45:17 PM |
| event | json | Feb 15, 2017 10:45:18 PM |
| event | json | Feb 15, 2017 10:45:19 PM |
| event | json | Feb 15, 2017 10:45:20 PM |
| event | json | Feb 15, 2017 10:45:21 PM |

Sensor Information

| Event | Datapoint | Value | Time Received |
|---|---|---|---|
| event | d.temp | 32.6 | Feb 15, 2017 10:45:21 PM |

*Use Watson IoT Platform to emit sensor data to the Cloud (see pg. 5)*



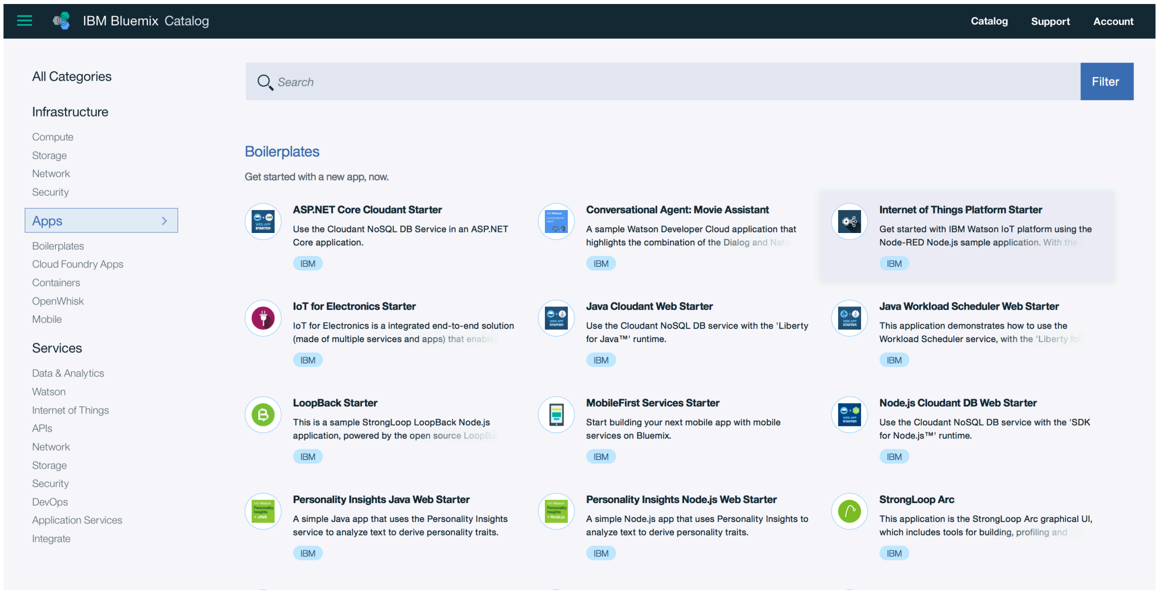
```
2/16/2017, 8:48:10 AM   node: cpu status
msg.payload : string[37]
"Temperature (33.6) within safe limits"
2/16/2017, 8:48:10 AM   node: device data
iot-2/type/raspberrypi/id/pidevice007/evt/event/fmt/json : msg :
Object
▸{ topic: "iot-2/type/raspberrypi
/id/pide…", payload: object, deviceId:
"pidevice007", deviceType:
"raspberrypi", eventType: "event" … }
2/16/2017, 8:48:11 AM   node: cpu status
msg.payload : string[37]
"Temperature (33.6) within safe limits"
2/16/2017, 8:48:11 AM   node: device data
iot-2/type/raspberrypi/id/pidevice007/evt/event/fmt/json : msg :
Object
▸{ topic: "iot-2/type/raspberrypi
/id/pide…", payload: object, deviceId:
"pidevice007", deviceType:
"raspberrypi", eventType: "event" … }
2/16/2017, 8:48:12 AM   node: cpu status
msg.payload : string[37]
"Temperature (34.2) within safe limits"
2/16/2017, 8:48:12 AM   node: device data
iot-2/type/raspberrypi/id/pidevice007/evt/event/fmt/json : msg :
Object
▸{ topic: "iot-2/type/raspberrypi
/id/pide…", payload: object, deviceId:
"pidevice007", deviceType:
"raspberrypi", eventType: "event" … }
```

*Subscribe to IoT events in the Cloud (see pg. 9)*

A digital copy of this lab and completed flow can be found at:
`http://ibm.biz/node-red-raspberrypi-temperature`

# Connect to Watson IoT Platform

1. Sign up for a IBM Bluemix account at `bluemix.net` and/or sign into your IBM Bluemix console.
2. Click on the **Catalog** link in the top-right corner.
3. Under the **Boilerplates** section, select the **Internet of Things Platform Starter** service tile.



Boilerplates

Get started with a new app, now.

**ASP.NET Core Cloudant Starter**
Use the Cloudant NoSQL DB Service in an ASP.NET Core application.
IBM

**Conversational Agent: Movie Assistant**
A sample Watson Developer Cloud application that highlights the combination of the Dialog and Natu
IBM

**Internet of Things Platform Starter**
Get started with IBM Watson IoT platform using the Node-RED Node.js sample application. With the
IBM

**IoT for Electronics Starter**
IoT for Electronics is a integrated end-to-end solution (made of multiple services and apps) that enable
IBM

**Java Cloudant Web Starter**
Use the Cloudant NoSQL DB service with the 'Liberty for Java™' runtime.
IBM

**Java Workload Scheduler Web Starter**
This application demonstrates how to use the Workload Scheduler service, with the 'Liberty for
IBM

**LoopBack Starter**
This is a sample StrongLoop LoopBack Node.js application, powered by the open source LoopB
IBM

**MobileFirst Services Starter**
Start building your next mobile app with mobile services on Bluemix.
IBM

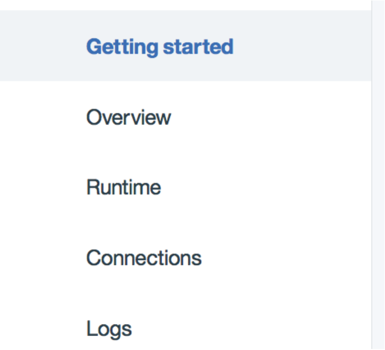
**Node.js Cloudant DB Web Starter**
Use the Cloudant NoSQL DB service with the 'SDK for Node.js™' runtime.
IBM

**Personality Insights Java Web Starter**
A simple Java app that uses the Personality Insights service to analyze text to derive personality traits.
IBM

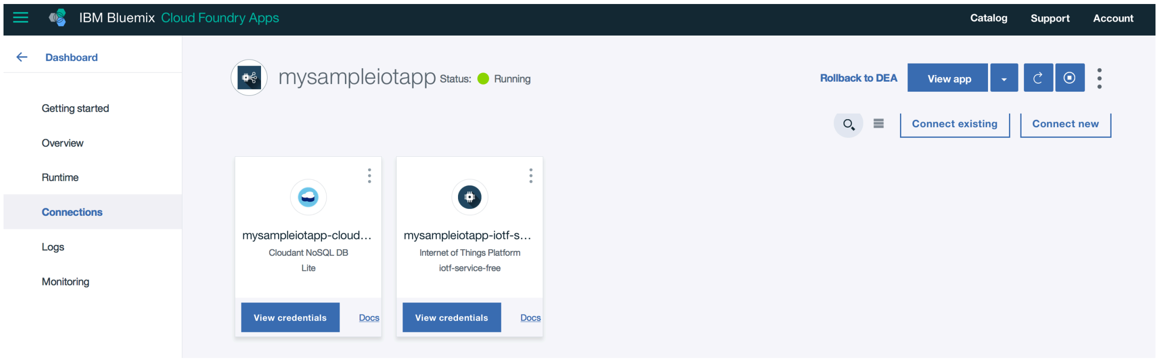
**Personality Insights Node.js Web Starter**
A simple Node.js app that uses Personality Insights to analyze text to derive personality traits.
IBM

**StrongLoop Arc**
This application is the StrongLoop Arc graphical UI, which includes tools for building, profiling and
IBM

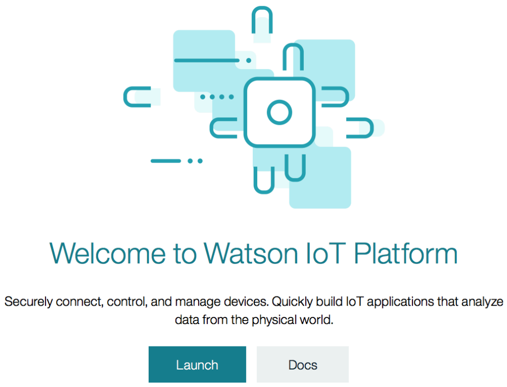
4. Enter an application name and host name which will be used for your application's URL. Click **Create** to create application.
5. In the left sidebar menu, click on **Connections**.



6. Click on the **Internet of Things** service title.



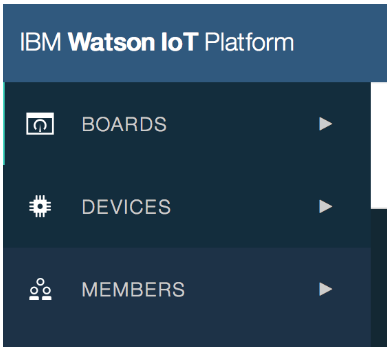
7. Click on the green **Launch** button to launch the Watson IoT Platform dashboard.



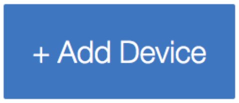
Welcome to Watson IoT Platform

Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.

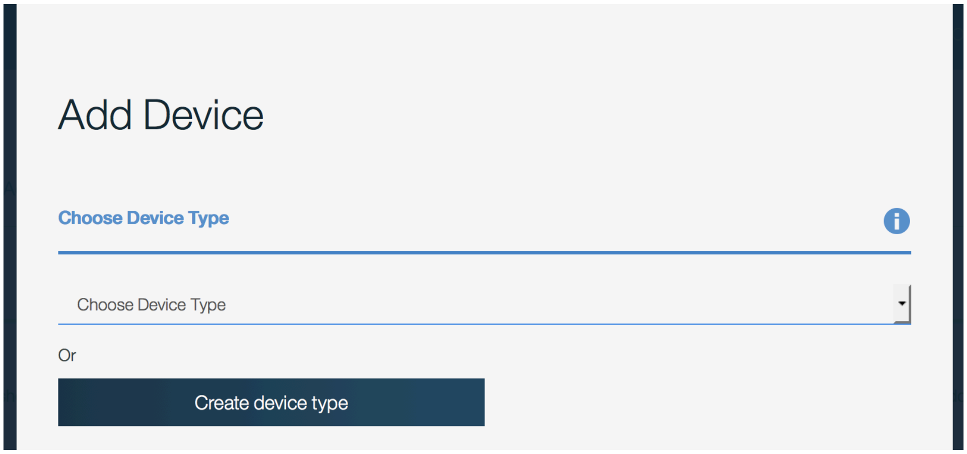
8. In the left sidebar menu, click on **Devices**.



9. Click **Add Device** in the top-right corner to create a new category of devices.



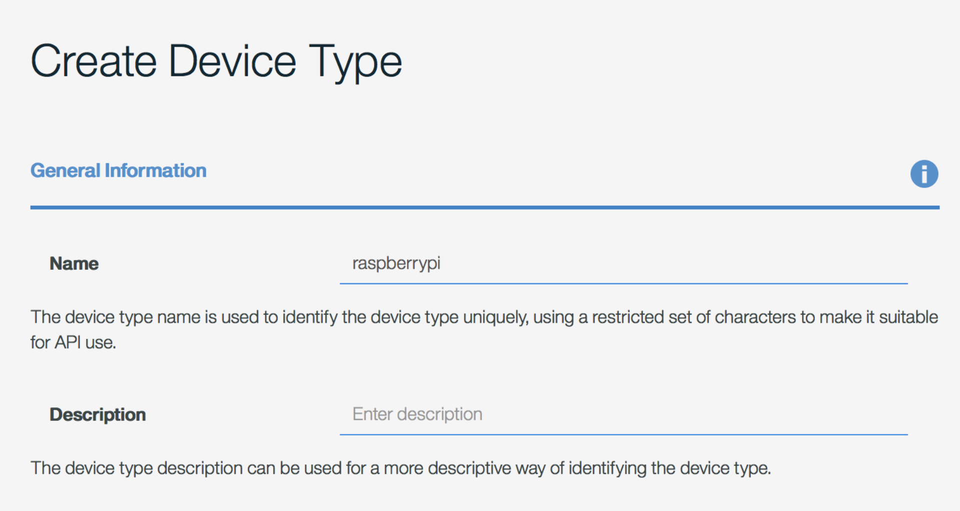
10. Click on **Create device type**. Click on **Create device type** on the next screen.



Add Device

Choose Device Type

11. We'll create a device type labeled `raspberrypi`. In the field labeled **Name**, enter `raspberrypi`. Click **Next** several times to skip through the following screens, and finally **Create**.



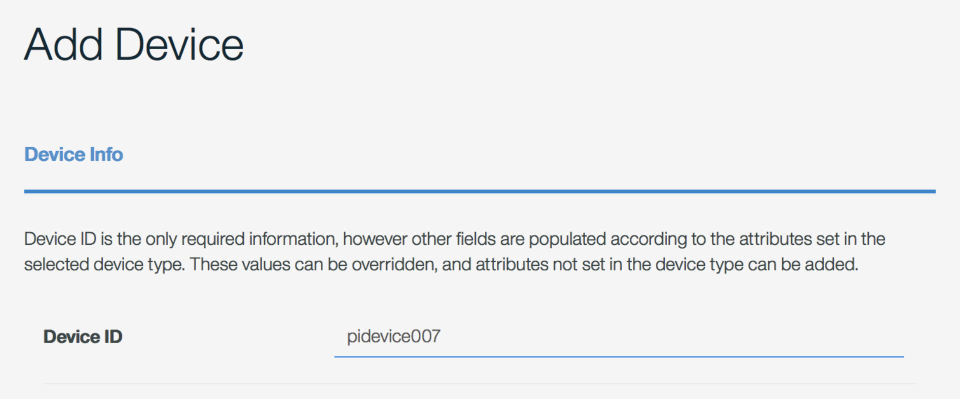
Name: raspberrypi

The device type name is used to identify the device type uniquely, using a restricted set of characters to make it suitable for API use.

Description: Enter description

The device type description can be used for a more descriptive way of identifying the device type.

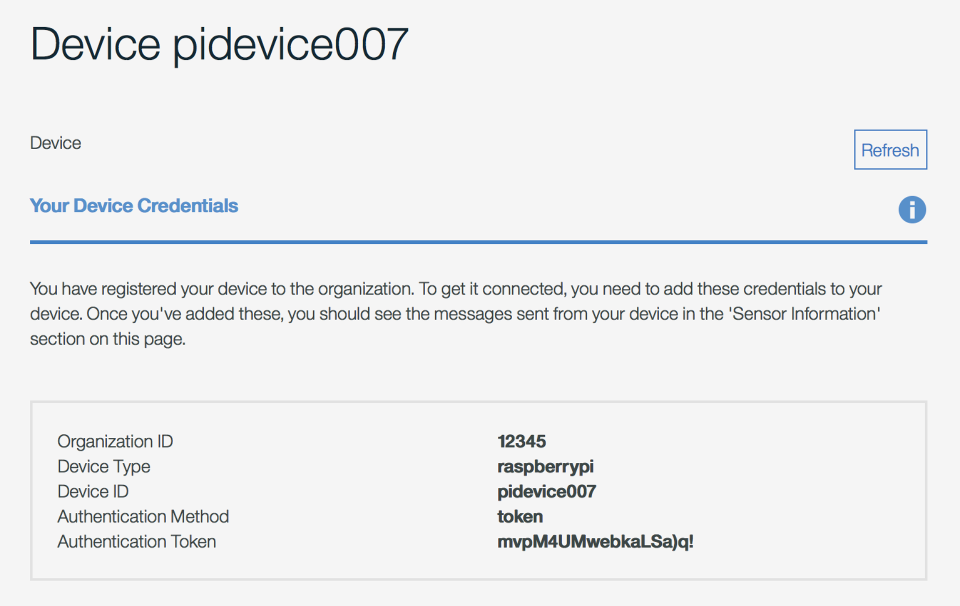
12. The new device type should be selected in the drop-down menu. Click **Next** to begin registering an individual device of type `raspberrypi`. Enter a device ID, `pidevice007`, which uniquely identifies the device we're registering.



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: pidevice007

13. Click on **Next** to skip through the following screens, and finally **Add**. On the next screen, you'll see device credentials. Make sure to copy the **Authentication Token** and other values in this box. The Authentication Token cannot be retrieved.



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 | **12345** |
| Device Type | **raspberrypi** |
| Device ID | **pidevice007** |
| Authentication Method | **token** |
| Authentication Token | **mvpM4UMwebkaLSa)q!** |

# Setup Raspberry Pi and Connect to Watson IoT

In this section, we will use Node-RED to read the temperature value of the CPU on a Raspberry Pi and emit the value to the IBM Watson IoT Platform where you can use the data in your Cloud application. Before we use Node-RED, we need to install the Watson IoT nodes.

1. SSH into your Raspberry Pi and confirm it is connected to the Internet. You need to be on the same network as the Raspberry Pi.

```
    $ ssh raspberry@<<IP ADDRESS>>
```

2. Create a `node_modules` in the directory `~/.node-red`

```
    $ mkdir -p ~/.node-red/node_modules
```

3. Change into the `node_modules` directory

```
    $ cd ~/.node-red/node_modules
```

   Install the Watson IoT Node-RED package (npm not found? Use the commands `sudo apt-get update` and `sudo apt-get install npm` to install)

```
    $ npm install node-red-contrib-ibm-watson-iot
```

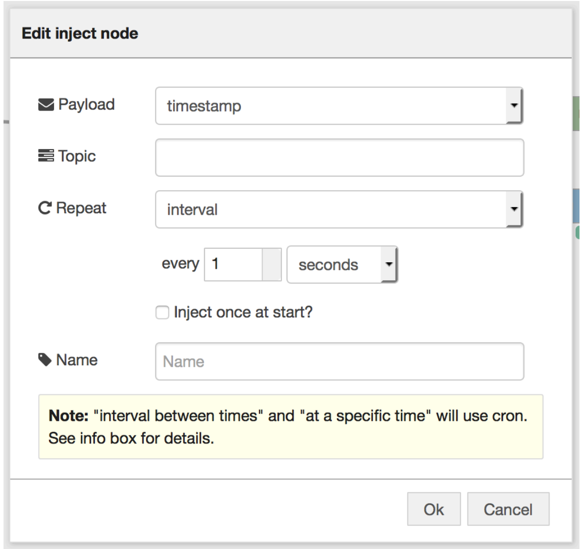
4. Start Node-RED

```
    $ node-red
```

5. Node-RED is a visual editor that makes connecting Internet of Things devices, such as the Raspberry Pi, to the Internet and Cloud platforms such as IBM Bluemix. To access the web-based editor, open a web browser and enter the IP address of the Raspberry Pi, followed by :1880.

```
http://<<IP ADDRESS>>:1880
```

6. On the left sidebar are nodes you can drag into the middle pane to create flows. Drag a inject node into the middle pane. Double click on the node, and change the values as shown below. Click **Ok** when finished.



Edit inject node

Payload: timestamp

Topic:

Repeat: interval

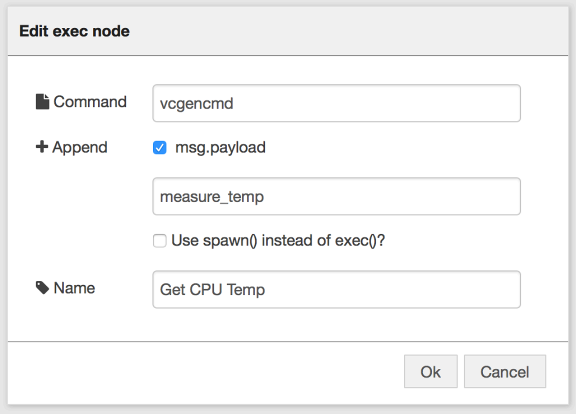
every 1 seconds

☐ Inject once at start?

Name: Name

**Note:** "interval between times" and "at a specific time" will use cron. See info box for details.

7. Add a exec node as shown below. Click **Ok** when finished.



**Edit exec node**

Command: vcgencmd

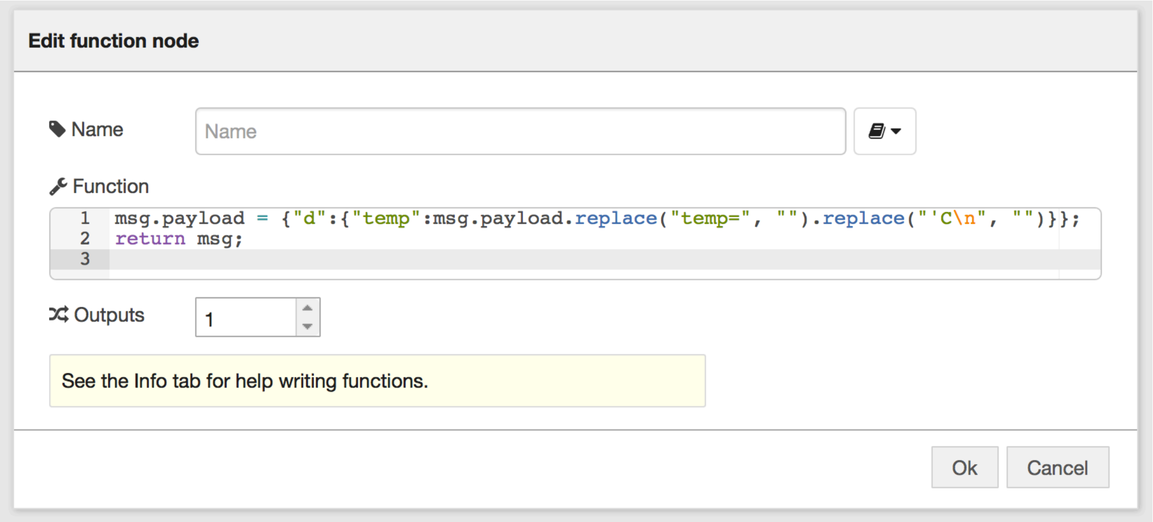
Append: ☑ msg.payload

measure_temp

□ Use spawn() instead of exec()?

Name: Get CPU Temp

8. Add a function node as shown below. Click **Ok** when finished.



**Edit function node**

Name: Name

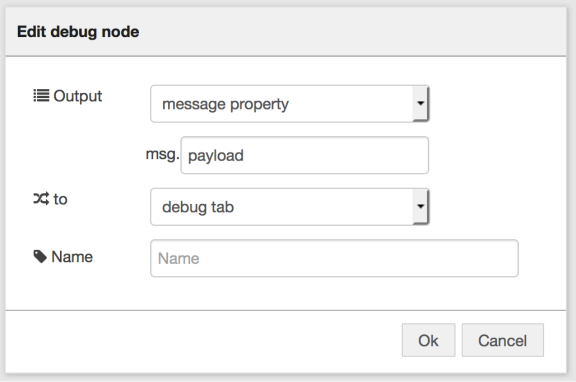
Function

```
msg.payload = {"d":{"temp":msg.payload.replace("temp=", "").replace("'C\n", "")}};
return msg;
```

Outputs: 1

See the Info tab for help writing functions.

9. Add a debug node as shown below. Click **Ok** when finished.



**Edit debug node**

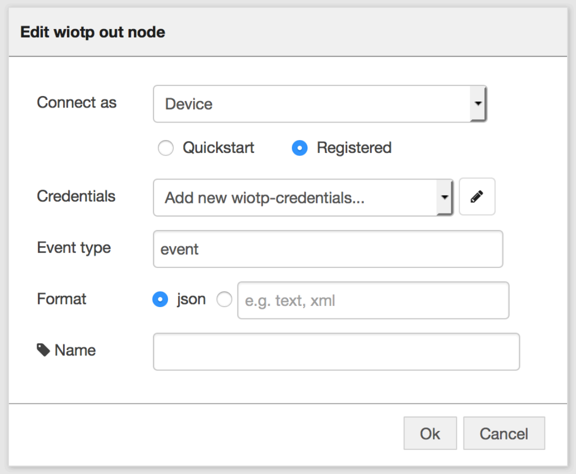
Output: message property

msg. payload

to: debug tab

Name: Name

10. Add a Watson IoT node. Check **Registered** and click on the pencil icon.



Edit wiotp out node

Connect as: Device

○ Quickstart ● Registered

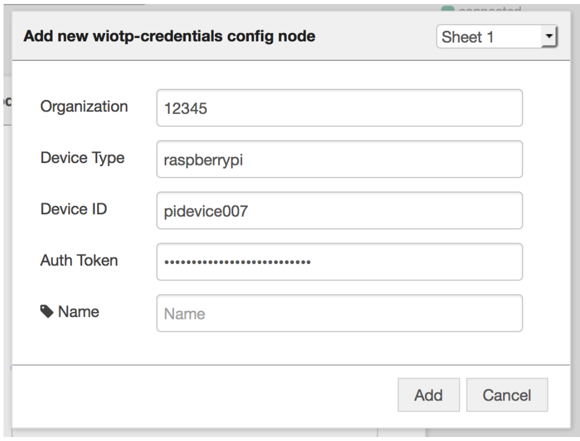
Credentials: Add new wiotp-credentials...

Event type: event

Format: ● json ○ e.g. text, xml

Name:

11. Enter the device credentials you got when you registered a new device from Step #11 on page 4. Click **Add** and **OK**.



Add new wiotp-credentials config node

Sheet 1

Organization: 12345

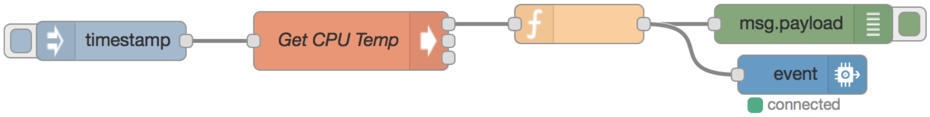
Device Type: raspberrypi

Device ID: pidevice007

Auth Token: ••••••••••••••••••••••••••

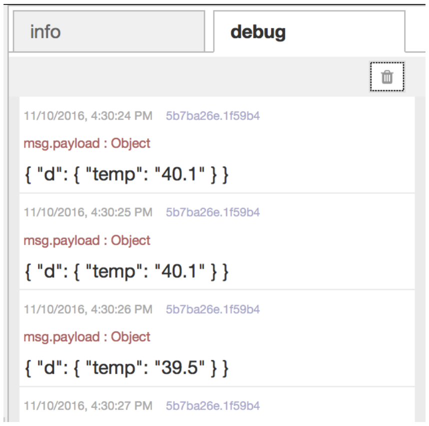
Name: Name

12. Connect the nodes together as shown below.



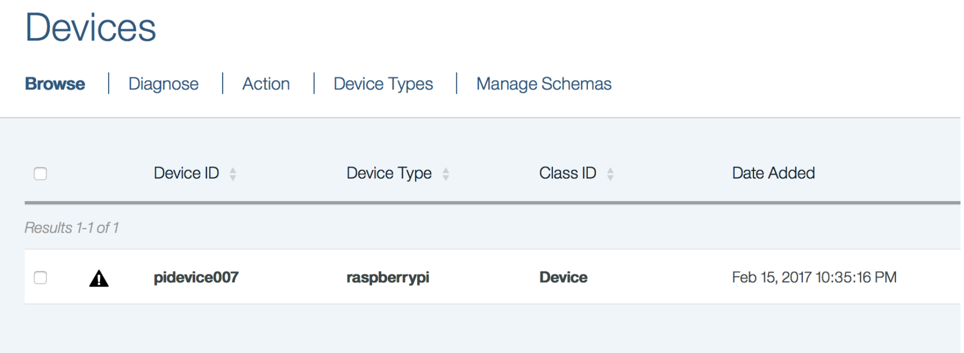
13. Click on the Deploy button in the top-right corner to save and deploy your changes.

14. In the **Debug** tab in the right sidebar, you should see the temperature data outputted every second.



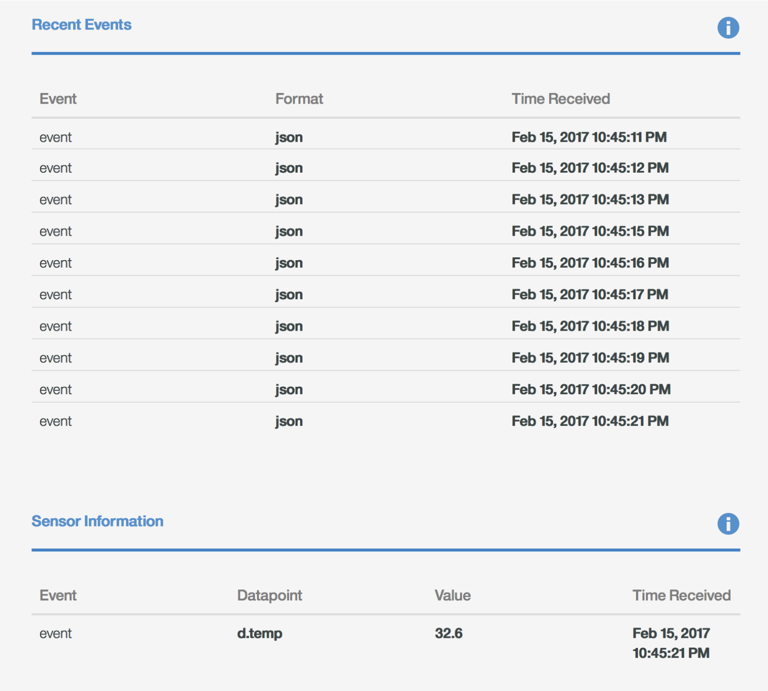
```
11/10/2016, 4:30:24 PM   5b7ba26e.1f59b4
msg.payload : Object
{ "d": { "temp": "40.1" } }
11/10/2016, 4:30:25 PM   5b7ba26e.1f59b4
msg.payload : Object
{ "d": { "temp": "40.1" } }
11/10/2016, 4:30:26 PM   5b7ba26e.1f59b4
msg.payload : Object
{ "d": { "temp": "39.5" } }
11/10/2016, 4:30:27 PM   5b7ba26e.1f59b4
```

15. Return to the Watson IoT Platform dashboard. Click on the device from the **Devices** list.



| □ |  | Device ID | Device Type | Class ID | Date Added |
|---|---|---|---|---|---|
| Results 1-1 of 1 |  |  |  |  |  |
| □ | ⚠ | **pidevice007** | **raspberrypi** | **Device** | Feb 15, 2017 10:35:16 PM |

16. Scroll down to the sections labeled **Recent Events** and **Sensor Information**. You can view the sensor data coming into the dashboard in real-time.


Recent Events

| Event | Format | Time Received |
|---|---|---|
| event | **json** | **Feb 15, 2017 10:45:11 PM** |
| event | **json** | **Feb 15, 2017 10:45:12 PM** |
| event | **json** | **Feb 15, 2017 10:45:13 PM** |
| event | **json** | **Feb 15, 2017 10:45:15 PM** |
| event | **json** | **Feb 15, 2017 10:45:16 PM** |
| event | **json** | **Feb 15, 2017 10:45:17 PM** |
| event | **json** | **Feb 15, 2017 10:45:18 PM** |
| event | **json** | **Feb 15, 2017 10:45:19 PM** |
| event | **json** | **Feb 15, 2017 10:45:20 PM** |
| event | **json** | **Feb 15, 2017 10:45:21 PM** |

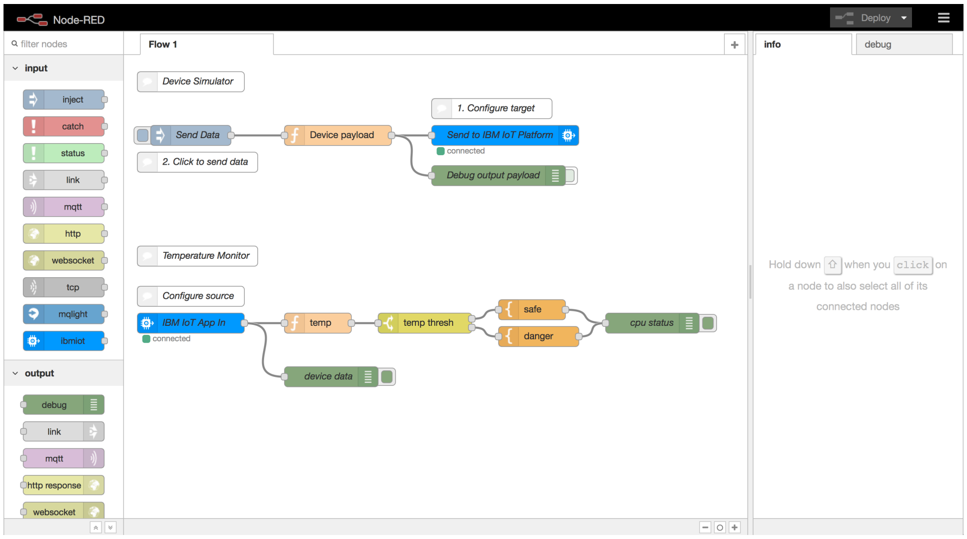
Sensor Information

| Event | Datapoint | Value | Time Received |
|---|---|---|---|
| event | **d.temp** | **32.6** | **Feb 15, 2017 10:45:21 PM** |

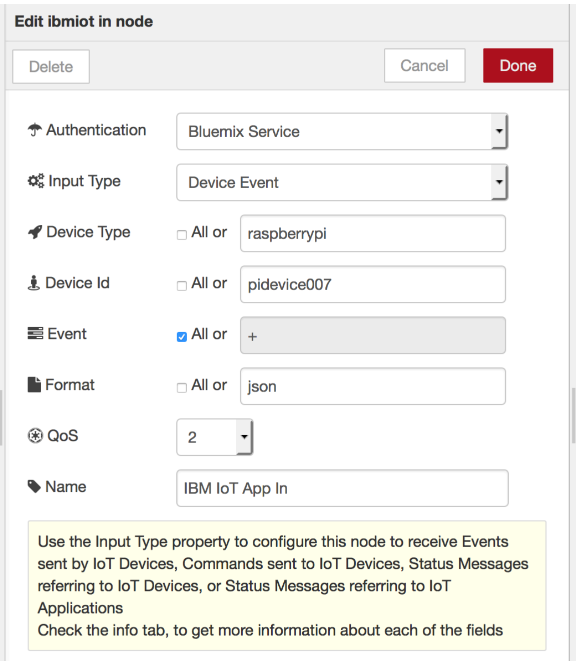
# Responding To Events In The Cloud

The Internet of Things Starter boilerplate also includes an instance of Node-RED hosted in the Cloud. In this section, we'll modify the Node-RED application to subscribe to the events emitted from the Raspberry PI we connected earlier.

1. Visit your Node-RED application in the Cloud by using the host name you chose in Step #4 on page 2, appended with the domain `.mybluemix.net`. If I chose `myapp` as the host name, the URL would be `http://myapp.mybluemix.net`.
2. Click on the button labeled **Go to your Node-RED flow editor**.
3. Two flows have been added for you. The first flow allows you to trigger events for a simulated device. The second flow subscribes to IoT events. Double click on the node labeled **IBM IoT App In.**



4. Configure the node as shown below using the **Device Type** and **Device ID** you used in Step #11 and #12 on page 4.



**Edit ibmiot in node**

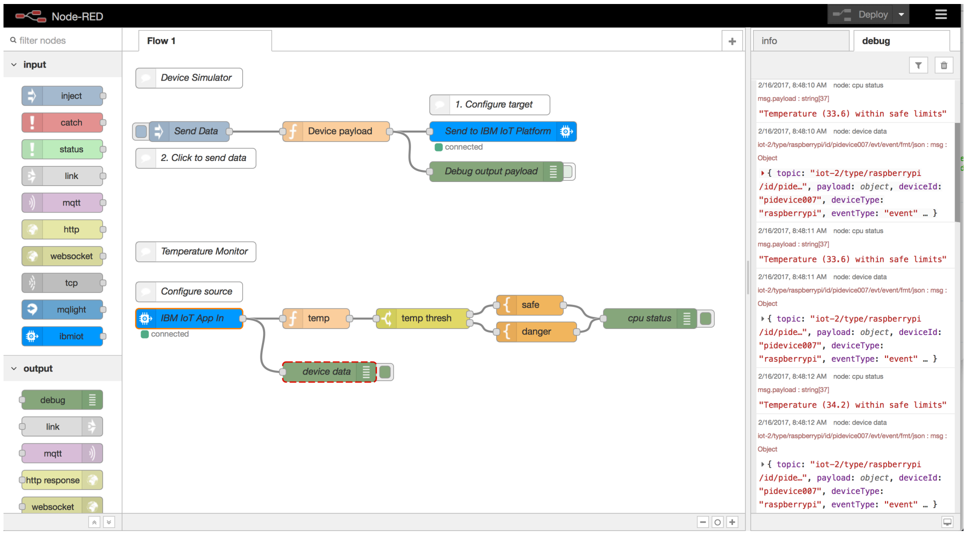
Delete | Cancel | Done

| | | |
|---|---|---|
| Authentication | | Bluemix Service |
| Input Type | | Device Event |
| Device Type | ☐ All or | raspberrypi |
| Device Id | ☐ All or | pidevice007 |
| Event | ☑ All or | + |
| Format | ☐ All or | json |
| QoS | | 2 |
| Name | | IBM IoT App In |

Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to IoT Applications
Check the info tab, to get more information about each of the fields

5. Click on the Deploy button in the top-right corner to save and deploy your changes.

6. In the **Debug** tab in the right sidebar, you should see the temperature data outputted every second.



```
2/16/2017, 8:48:10 AM   node: cpu status
msg.payload : string[37]
"Temperature (33.6) within safe limits"
2/16/2017, 8:48:10 AM   node: device data
iot-2/type/raspberrypi/id/pidevice007/evt/event/fmt/json : msg :
Object
▸{ topic: "iot-2/type/raspberrypi
/id/pide…", payload: object, deviceId:
"pidevice007", deviceType:
"raspberrypi", eventType: "event" … }
2/16/2017, 8:48:11 AM   node: cpu status
msg.payload : string[37]
"Temperature (33.6) within safe limits"
2/16/2017, 8:48:11 AM   node: device data
iot-2/type/raspberrypi/id/pidevice007/evt/event/fmt/json : msg :
Object
▸{ topic: "iot-2/type/raspberrypi
/id/pide…", payload: object, deviceId:
"pidevice007", deviceType:
"raspberrypi", eventType: "event" … }
2/16/2017, 8:48:12 AM   node: cpu status
msg.payload : string[37]
"Temperature (34.2) within safe limits"
2/16/2017, 8:48:12 AM   node: device data
iot-2/type/raspberrypi/id/pidevice007/evt/event/fmt/json : msg :
Object
▸{ topic: "iot-2/type/raspberrypi
/id/pide…", payload: object, deviceId:
"pidevice007", deviceType:
"raspberrypi", eventType: "event" … }
```

Notice the second flow has a switch node that tests the temperature value and splits the program flow. If the temperature is above 40°C, a “danger” message is outputted. Otherwise, the temperature is considered “safe.”

Now that you have the temperature data in the Cloud, you may choose to respond to device events using the Bluemix application and, for example, send a text message using Twilio. For other Node-RED labs showing how to use IBM Watson, please visit `ibm.biz/node-red-labs`.