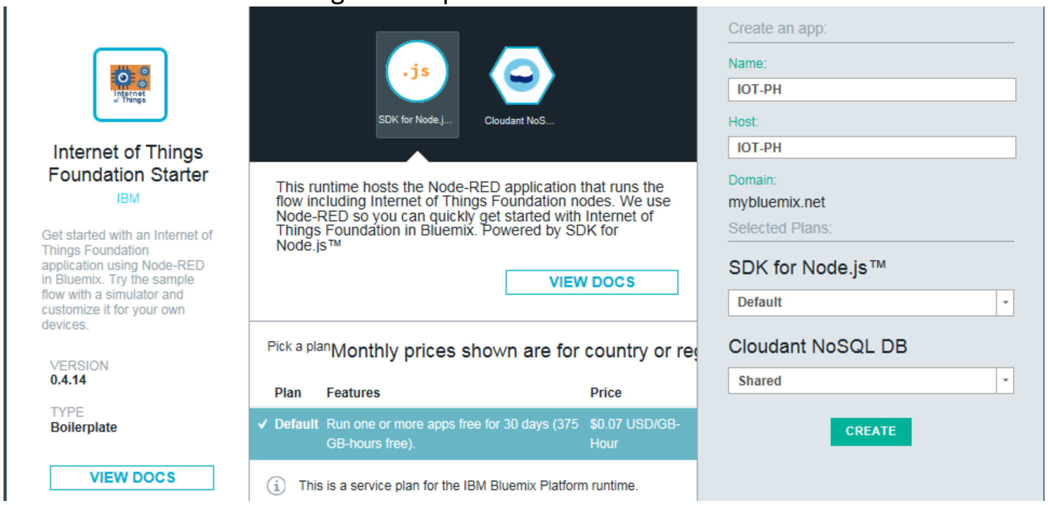
Node-RED Internet of Things

It's easy to build an application with Node-RED and use the data that you've published to the IBM Internet of Things cloud. This hands-on ab has a ready-made flow that processes temperature readings from a simulated sensor device. The flow checks these readings against a threshold, determines whether the temperature is safe. Actions can then be taken to alert operations staff with e-mail or SMS messages, record event data in databases or publish the data to external sources thru standard web interfaces such as sockets, ports or datafeeds. This use case would fit a scenario for monitoring CPU temperature or refrigerated transport containers from a real device.

If you want to know more about Node-RED, see http://nodered.org/docs/.

Step 1 – Create your Internet of Things application

- Login to Bluemix (https://ace.ng.bluemix.net)
- Select the 'CATALOG' tab
- Select the 'Internet of Things' boilerplate



Internet of Things Foundation Starter
IBM

Get started with an Internet of Things Foundation application using Node-RED in Bluemix. Try the sample flow with a simulator and customize it for your own devices.

VERSION
0.4.14

TYPE
Boilerplate

This runtime hosts the Node-RED application that runs the flow including Internet of Things Foundation nodes. We use Node-RED so you can quickly get started with Internet of Things Foundation in Bluemix. Powered by SDK for Node.js™

Pick a plan Monthly prices shown are for country or reg

Plan	Features	Price
✓ Default	Run one or more apps free for 30 days (375 GB-hours free).	$0.07 USD/GB-Hour

This is a service plan for the IBM Bluemix Platform runtime.

Create an app:

Name: IOT-PH

Host: IOT-PH

Domain: mybluemix.net

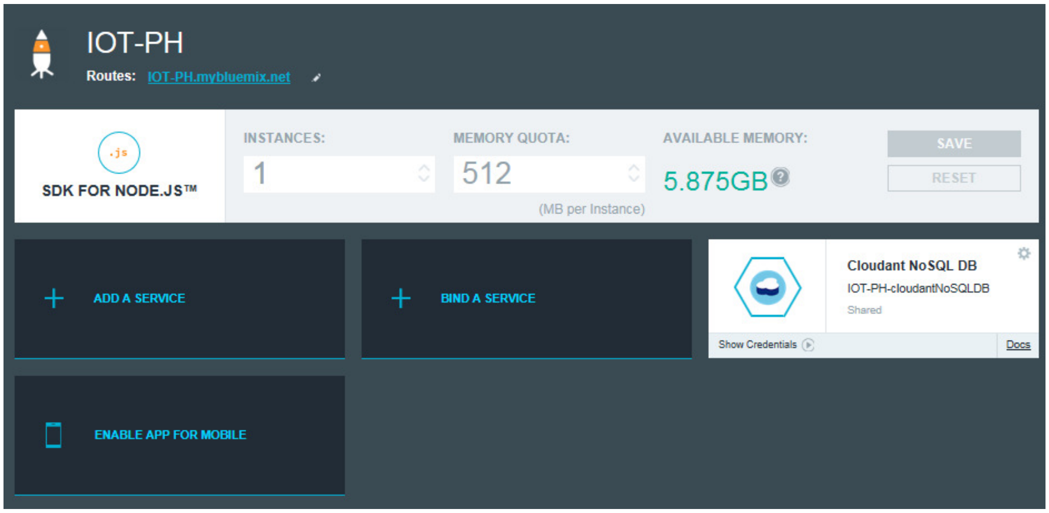
Selected Plans:

SDK for Node.js™: Default

Cloudant NoSQL DB: Shared

- Under "Create an App" on the right, provide a name for the application – the same name will be used for the hostname for the application URL, so make sure to choose a name that is unique in the "mybluemix.net" domain namespace.
- Select the 'CREATE' button to deploy the boilerplate.

- When the application is created it will appear in your Bluemix space. After a short wait, your application starts



IOT-PH

Routes: IOT-PH.mybluemix.net

SDK FOR NODE.JS™

INSTANCES: 1

MEMORY QUOTA: 512 (MB per Instance)

AVAILABLE MEMORY: 5.875GB

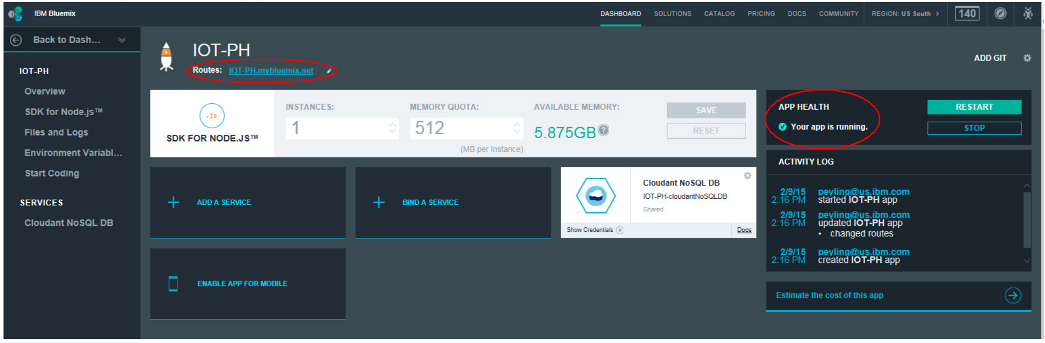
Cloudant NoSQL DB

IOT-PH-cloudantNoSQLDB

Shared

Step 2 – Launch the Node-RED flow editor

- Check if the application has started and is running, on the right side of the application overview, in “APP HEALTH”.
- If your application is not running, click the START (or RESTART) button and wait for the application to start
- Once the application is running, launch your app by clicking on the URL indicated by the “ROUTES” setting under the application name and icon at the top of the application overview:



IOT-PH

Routes: IOT-PH.mybluemix.net

INSTANCES: 1

MEMORY QUOTA: 512 (MB per Instance)

AVAILABLE MEMORY: 5.875GB

APP HEALTH

Your app is running.

Cloudant NoSQL DB
IOT-PH-cloudantNoSQLDB
Shared

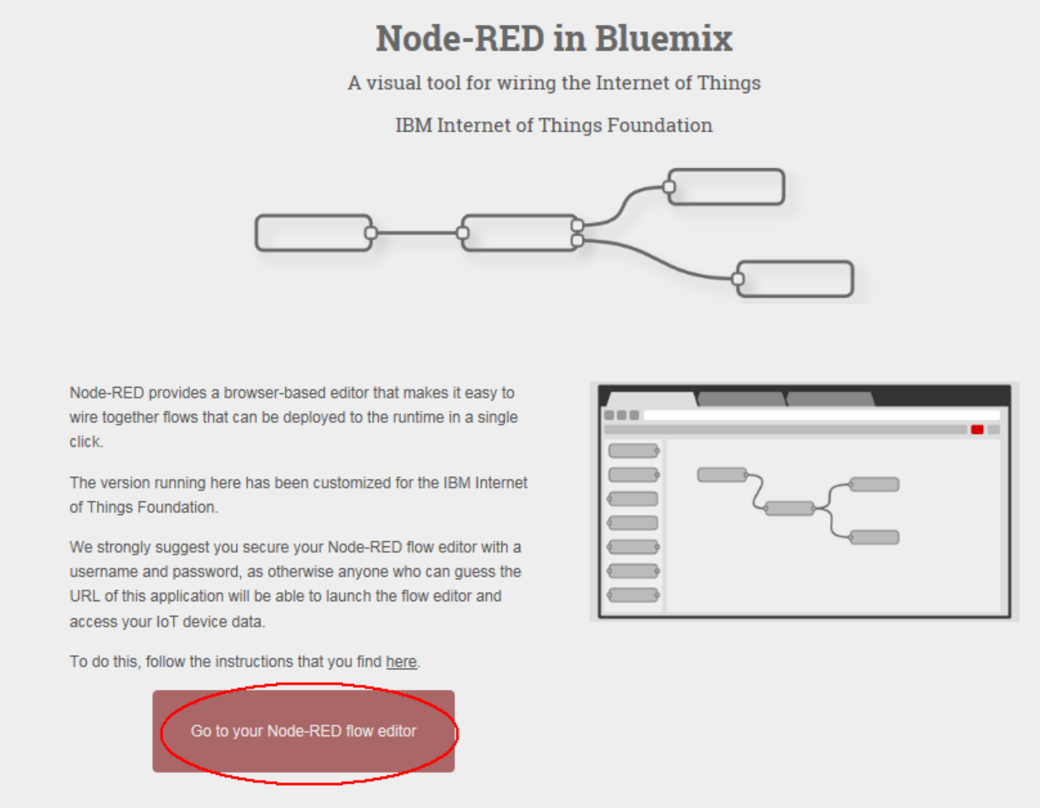
ACTIVITY LOG

2/9/15 2:16 PM peyling@us.ibm.com started **IOT-PH** app

2/9/15 2:16 PM peyling@us.ibm.com updated **IOT-PH** app
- changed routes

2/9/15 2:16 PM peyling@us.ibm.com created **IOT-PH** app

- The Node-RED application will open in your browser. Scroll down and select the ‘Go to your Node-RED flow editor’ button



Node-RED in Bluemix

A visual tool for wiring the Internet of Things

IBM Internet of Things Foundation

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

The version running here has been customized for the IBM Internet of Things Foundation.

We strongly suggest you secure your Node-RED flow editor with a username and password, as otherwise anyone who can guess the URL of this application will be able to launch the flow editor and access your IoT device data.

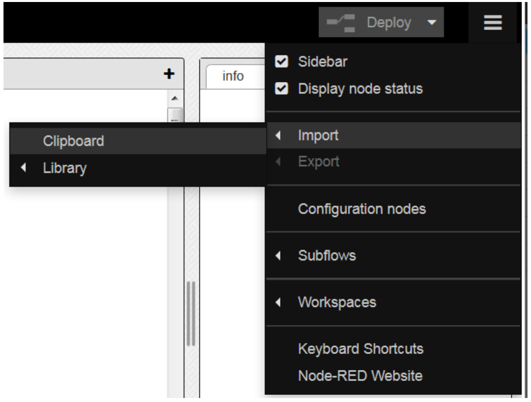
To do this, follow the instructions that you find here.

Copyright IBM Corporation 2014-2015. All rights reserved

Step 3 – Import the sample flow

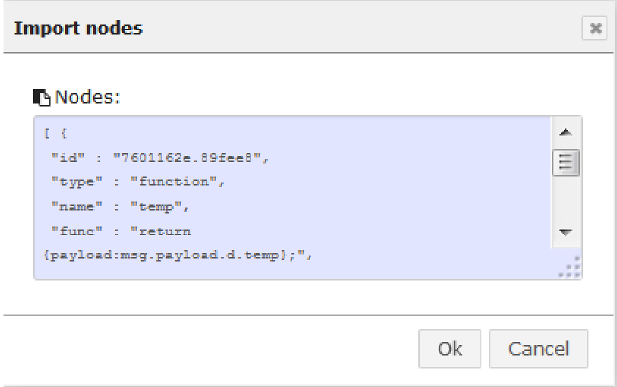
This step imports the sample flow to the Node-RED editor. Node-RED flows can be imported and exported as JSON text, so to import the flow you are going to copy the flow below and import it to Node-RED

- In the Node-RED flow editor select the menu ☰ (upper right of screen). Select **Import** then **Clipboard**



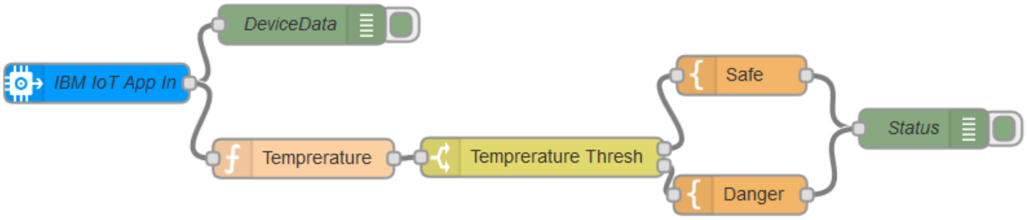
- This will open an Import nodes window. Copy and paste following text into the content of your system clipboard window, then press **OK**

```
 [{
        "id": "26a2dad.fd95d26",
        "type": "ibmiot in",
        "inputType": "evt",
        "mac": "",
        "deviceType": "+",
        "eventCommandType": "+",
        "format": "json",
        "name": "IBM IoT App In",
        "service": "quickstart",
        "macChecked": "",
        "deviceTypeChecked": true,
        "eventChecked": true,
        "formatChecked": "",
        "x": 217,
        "y": 225,
        "z": "67a2f363.985d0c",
        "wires": [["7601162e.89fee8",
        "bd385f48.42c7a"]]
},{
        "id": "7601162e.89fee8",
        "type": "function",
        "name": "temp",
        "func": "return {payload:msg.payload.d.temp};",
        "outputs": 1,
        "x": 390.5,
        "y": 271,
        "z": "67a2f363.985d0c",
        "wires": [["3e1f743c.c1e08c"]]
},{
        "id": "3e1f743c.c1e08c",
        "type": "switch",
        "name": "temp thresh",
        "property": "payload",
        "rules": [{
                "t": "lte",
                "v": 40,
                "v2": null
        },
        {
                "t": "gt",
                "v": 40,
                "v2": null
        }],
        "checkall": "true",
        "outputs": 2,
        "x": 556.5,
        "y": 271,
        "z": "67a2f363.985d0c",
        "wires": [["4d14d8db.b2eb28"],
        ["f5565792.0aa9a8"]]
},{
        "id": "43a6791f.bc5988",
        "type": "debug",
        "name": "Status",
        "active": true,
        "complete": "false",
        "x": 867.5,
        "y": 268,
        "z": "67a2f363.985d0c",
        "wires": []
},{
        "id": "bd385f48.42c7a",
        "type": "debug",
        "name": "device data",
        "active": false,
        "complete": "false",
        "x": 407.5,
        "y": 181,
        "z": "67a2f363.985d0c",
        "wires": []
},{
        "id": "4d14d8db.b2eb28",
        "type": "template",
        "name": "safe",
        "template": "Temperature ({{payload}}) within safe
limits",
        "x": 720.5,
        "y": 222,
        "z": "67a2f363.985d0c",
        "wires": [["43a6791f.bc5988"]]
},{
        "id": "f5565792.0aa9a8",
        "type": "template",
        "name": "danger",
        "template": "Temperature ({{payload}}) critical",
        "x": 719.5,
        "y": 316,
        "z": "67a2f363.985d0c",
        "wires": [["43a6791f.bc5988"]]
}]
```



```
[ {
 "id" : "7601162e.89fee8",
 "type" : "function",
 "name" : "temp",
 "func" : "return
{payload:msg.payload.d.temp};",
```

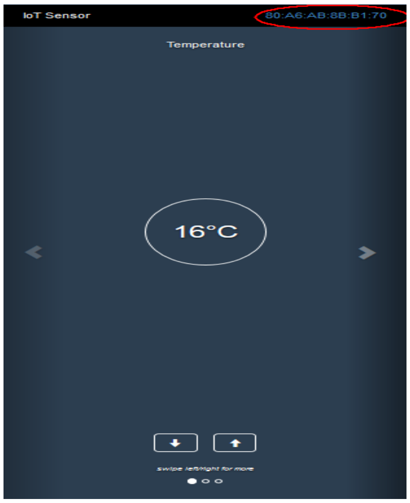
- You should see the flow in the flow editor connected to your cursor. Move the cursor to the left of the window to place the flow and left-click
- We'll be connecting the flow for input in the "IoT APP Input" node in the flow:



Step 4 – Configure the flow for a simulated device and deploy the flow

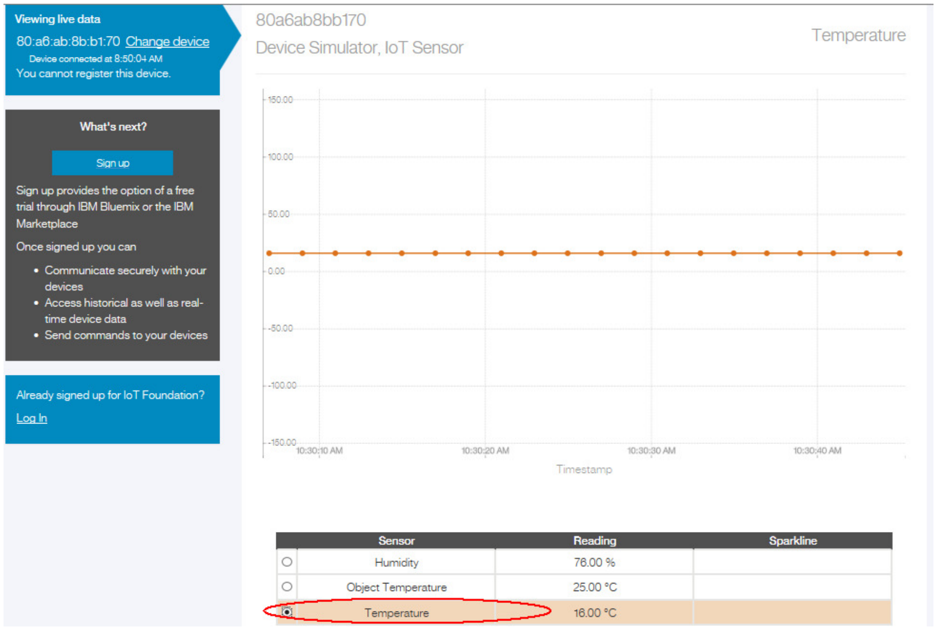
There is a simulated device running on BlueMix ready for you to test.

- In a new browser window open http://quickstart.internetofthings.ibmcloud.com/iotsensor



- The hardware MAC address of the simulated device is in the top right corner.

- You can click the MAC address to open up the Data Dashboard web console which shows what the device is publishing to the IBM Internet of Things cloud. Make sure to select “Temperature” for this lab exercise.



Viewing live data

80:a6:ab:8b:b1:70 Change device

Device connected at 8:50:04 AM

You cannot register this device.

What's next?

Sign up

Sign up provides the option of a free trial through IBM Bluemix or the IBM Marketplace

Once signed up you can

- Communicate securely with your devices
- Access historical as well as real-time device data
- Send commands to your devices

Already signed up for IoT Foundation?

Log In

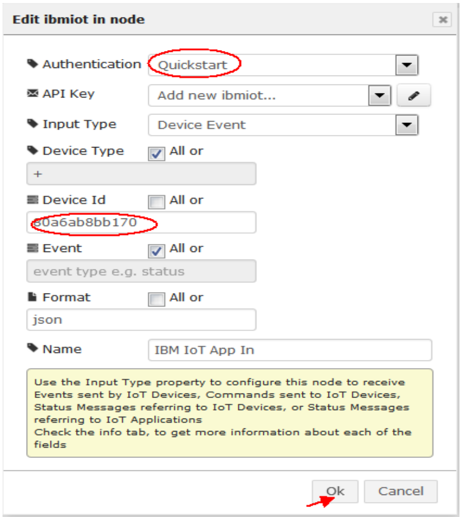
80a6ab8bb170

Device Simulator, IoT Sensor

Temperature

	Sensor	Reading	Sparkline
○	Humidity	76.00 %	
○	Object Temperature	25.00 °C	
●	Temperature	16.00 °C	

- In the Node-RED flow editor double-click the “IoT App Input” node (first node in the flow). This will bring up the node editor window. Select Quickstart in drop off list of the **Authentication** filed, check all for **Device Type** and **Event Type** and enter your own MAC address in lower case and without colon in between into the **Device ID** field. Then click **OK**.



Edit ibmiot in node

Authentication: Quickstart

API Key: Add new ibmiot...

Input Type: Device Event

Device Type ☑ All or

+

Device Id □ All or

80a6ab8bb170

Event ☑ All or

event type e.g. status

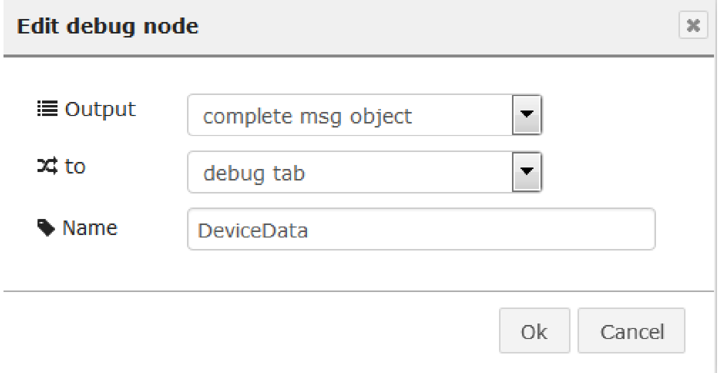
Format □ All or

json

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

- Check and update Device Data debug node by double-click on the node. Enter “Device Data” in **Name** field, select “complete msg object” from the drop down list of **Output** field and “debug tab” for **to** field then click **OK**.



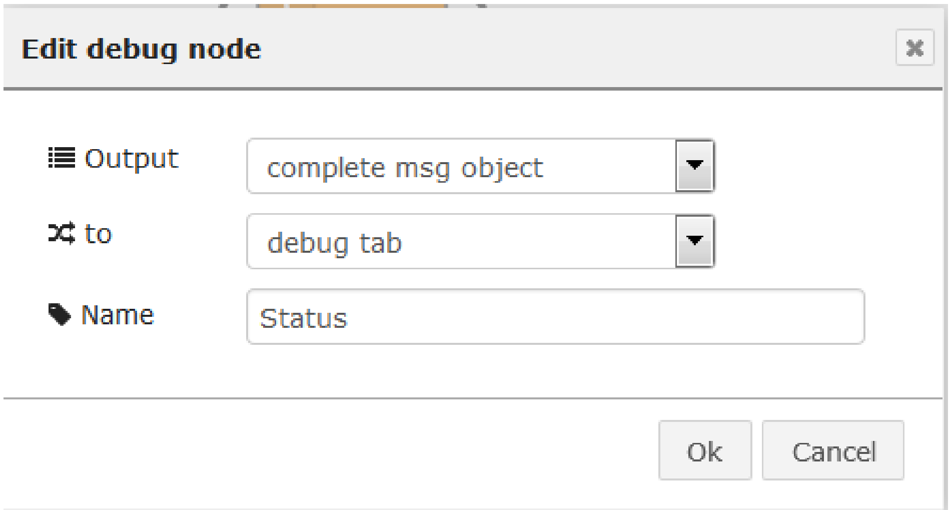
Edit debug node

Output: complete msg object

to: debug tab

Name: DeviceData

- Check and update Status debug node by double-click on the node. Enter “Status” in **Name** field, select “complete msg object” from the drop down list of **Output** field and “debug tab” for **to** field then click **OK**.



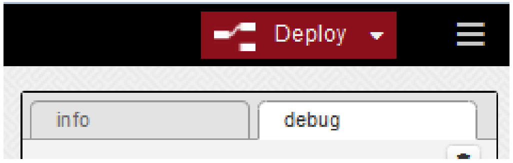
Edit debug node

Output: complete msg object

to: debug tab

Name: Status

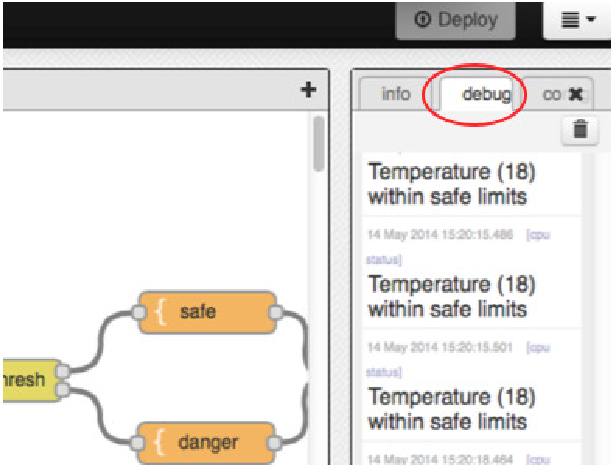
- The **Deploy** button now becomes active, so click **Deploy** to deploy the flow



- You will get a confirmation message that the flow is deployed.

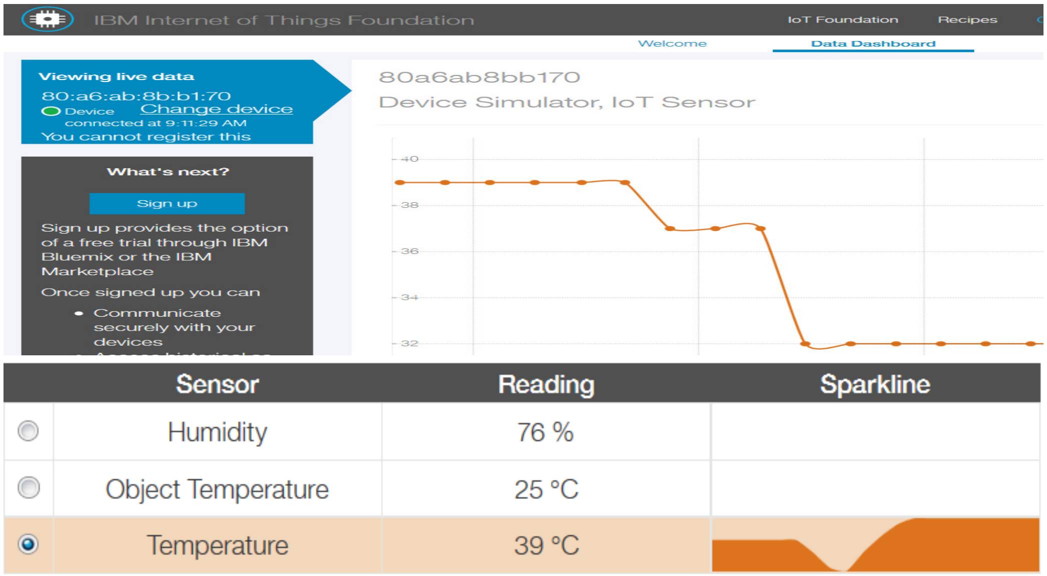
Step 5 – Test the flow

- In the side bar switch to the debug tab. You should now see the debug messages from the flow, showing the temperature published by the sensor.



```
Temperature (18)
within safe limits
14 May 2014 15:20:15.486 [cpu
status]
Temperature (18)
within safe limits
14 May 2014 15:20:15.501 [cpu
status]
Temperature (18)
within safe limits
14 May 2014 15:20:18.464 [cpu
```

- Try enabling the device data debug node to see the data being received by the iot input node.
- In the sensor simulator window try increasing the temperature being reported – you can view the realtime sensor data on the sensor device Data Dashboard app you opened from its MAC address previously:



Viewing live data
80:a6:ab:8b:b1:70
Device Change device
connected at 9:11:29 AM
You cannot register this

What's next?

Sign up provides the option of a free trial through IBM Bluemix or the IBM Marketplace

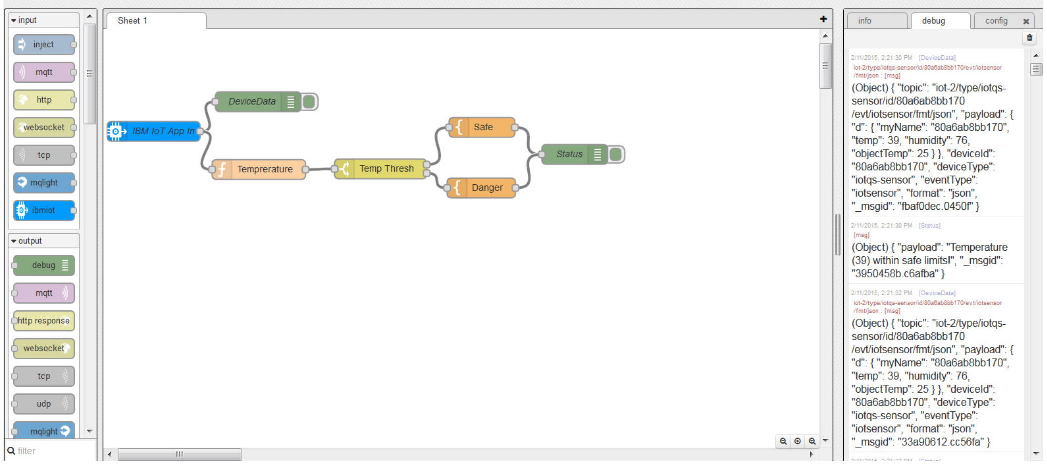
Once signed up you can

- Communicate securely with your devices

80a6ab8bb170
Device Simulator, IoT Sensor

	Sensor	Reading	Sparkline
○	Humidity	76 %	
○	Object Temperature	25 °C	
●	Temperature	39 °C	

- Back in the Node-RED flow editor, watch the debug messages report the new temperature.

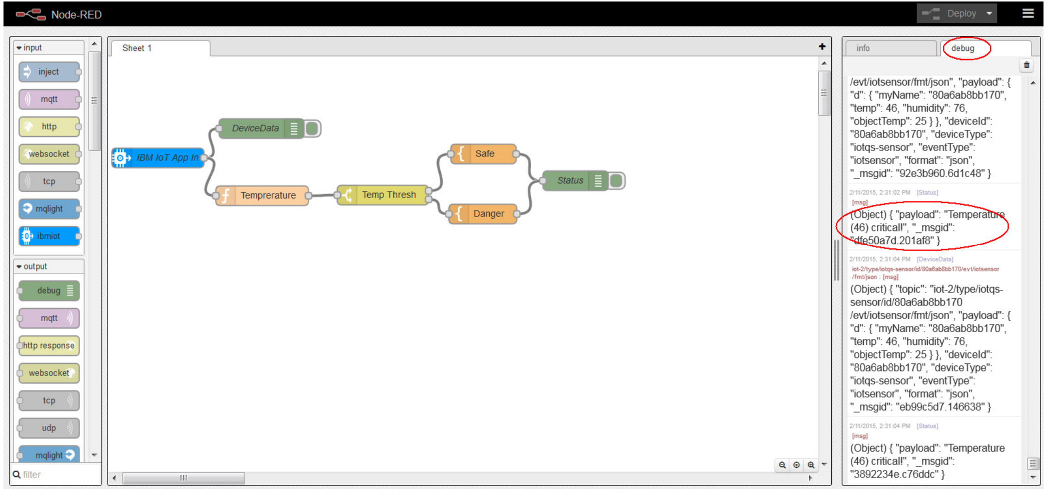


```
2/11/2015, 2:21:30 PM  [DeviceData]
iot-2/type/iotqs-sensor/id/80a6ab8bb170/evt/iotsensor
/fmt/json : [msg]
(Object) { "topic": "iot-2/type/iotqs-sensor/id/80a6ab8bb170
/evt/iotsensor/fmt/json", "payload": {
"d": { "myName": "80a6ab8bb170",
"temp": 39, "humidity": 76,
"objectTemp": 25 } }, "deviceId":
"80a6ab8bb170", "deviceType":
"iotqs-sensor", "eventType":
"iotsensor", "format": "json",
"_msgid": "fbaf0dec.0450f" }

2/11/2015, 2:21:30 PM  [Status]
[msg]
(Object) { "payload": "Temperature
(39) within safe limits!", "_msgid":
"3950458b.c6afba" }

2/11/2015, 2:21:32 PM  [DeviceData]
iot-2/type/iotqs-sensor/id/80a6ab8bb170/evt/iotsensor
/fmt/json : [msg]
(Object) { "topic": "iot-2/type/iotqs-sensor/id/80a6ab8bb170
/evt/iotsensor/fmt/json", "payload": {
"d": { "myName": "80a6ab8bb170",
"temp": 39, "humidity": 76,
"objectTemp": 25 } }, "deviceId":
"80a6ab8bb170", "deviceType":
"iotqs-sensor", "eventType":
"iotsensor", "format": "json",
"_msgid": "33a90612.cc56fa" }
```

- Keep increasing the temperature until you get to 41ºC. The debug message should now be advising that the temperature is critical



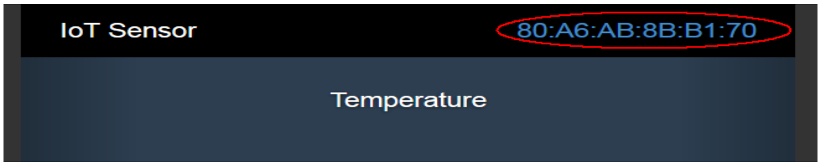
```
/evt/iotsensor/fmt/json", "payload": {
"d": { "myName": "80a6ab8bb170",
"temp": 46, "humidity": 76,
"objectTemp": 25 } }, "deviceId":
"80a6ab8bb170", "deviceType":
"iotqs-sensor", "eventType":
"iotsensor", "format": "json",
"_msgid": "92e3b960.6d1c48" }

2/11/2015, 2:31:02 PM  [Status]
[msg]
(Object) { "payload": "Temperature
(46) critical!", "_msgid":
"dfe50a7d.201af8" }

2/11/2015, 2:31:04 PM  [DeviceData]
iot-2/type/iotqs-sensor/id/80a6ab8bb170/evt/iotsensor
/fmt/json : [msg]
(Object) { "topic": "iot-2/type/iotqs-sensor/id/80a6ab8bb170
/evt/iotsensor/fmt/json", "payload": {
"d": { "myName": "80a6ab8bb170",
"temp": 46, "humidity": 76,
"objectTemp": 25 } }, "deviceId":
"80a6ab8bb170", "deviceType":
"iotqs-sensor", "eventType":
"iotsensor", "format": "json",
"_msgid": "eb99c5d7.146638" }

2/11/2015, 2:31:04 PM  [Status]
[msg]
(Object) { "payload": "Temperature
(46) critical!", "_msgid":
"3892234e.c76ddc" }
```

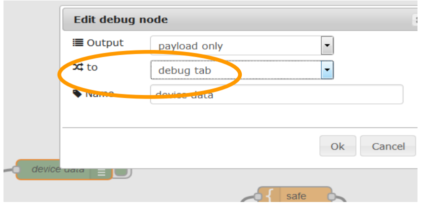
Visualize
Real-time visualization of the data
Note the MAC address of the simulated device (displayed in the top right corner, it's something like 80:A6:AB:8B:B1:70. Write down your own MAC address for later use.



- Let's view more of the device data direct from the sensor event's data object using the debug node – in the Node-RED flow editor, **double click** on the debug node for **device data**

Copyright IBM Corporation 2014-2015. All rights reserved

- In the **Edit debug node** dialog, in the **to** setting, select **debug tab** as in the image above – this will send the object payload data to the debug console tab for the sensor event publishing to Internet of Things. Click **Ok** and **Deploy** to redeploy the flow



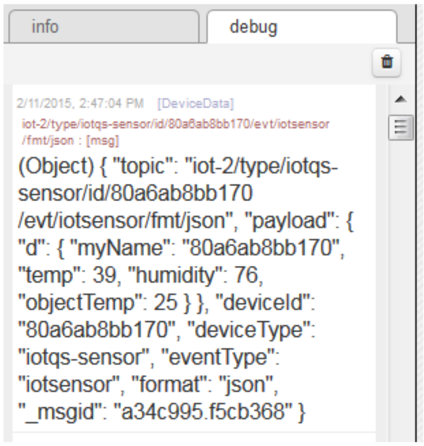
Edit debug node

Output: payload only

to: debug tab

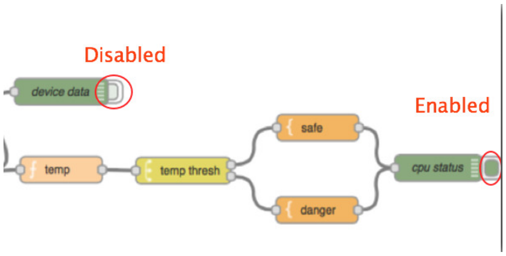
Name: device data

- In the **Debug** console tab you should now see not only the messages displayed but also the sensor event data object for temperature, humidity and device temperature



```
2/11/2015, 2:47:04 PM   [DeviceData]
iot-2/type/iotqs-sensor/id/80a6ab8bb170/evt/iotsensor
/fmt/json : [msg]
(Object) { "topic": "iot-2/type/iotqs-
sensor/id/80a6ab8bb170
/evt/iotsensor/fmt/json", "payload": {
"d": { "myName": "80a6ab8bb170",
"temp": 39, "humidity": 76,
"objectTemp": 25 } }, "deviceId":
"80a6ab8bb170", "deviceType":
"iotqs-sensor", "eventType":
"iotsensor", "format": "json",
"_msgid": "a34c995.f5cb368" }
```

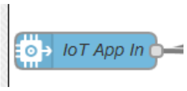
- You can enable or disable either of the 2 debug nodes in the flow by clicking in the area to the right of the debug node, without needing to re-deploy the flow



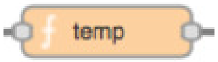
Step 6 - Understanding the flow

You can examine the content of any node by **double clicking** it. If you modify a node or add additional nodes the deploy button will become active allowing you to deploy the modified flow.

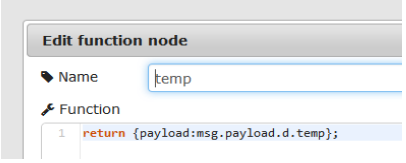
The nodes in our flow each implement the steps as follows:



1. The iot input node (above) subscribes to receive messages from the Internet of Things cloud for a specific device



→



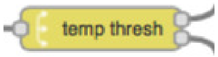
Edit function node

Name: temp

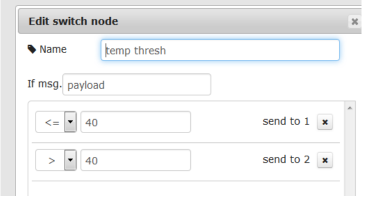
Function

```
return {payload:msg.payload.d.temp};
```

2. A function node (above) takes the data from the iot node and extracts the temperature from the message



→



Edit switch node

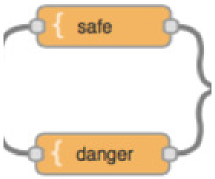
Name: temp thresh

If msg. payload

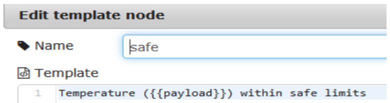
<= 40 send to 1

> 40 send to 2

3. A decision switch node (above) compares the temperature value and sends the flow one of 2 ways depending on the temperature value



→



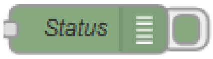
Edit template node

Name: safe

Template

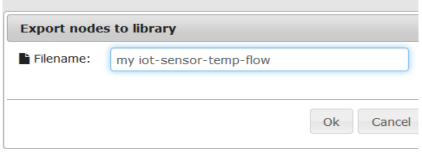
```
Temperature ({{payload}}) within safe limits
```

4. Function nodes (above) create the message to be displayed in the debug node



5. The debug node (above) displays the message to the debug panel

Once you have created a flow it can be exported to the clipboard window or your storage via the Export options in the flow editor menu. This allows you to share a flow or put it into source control.

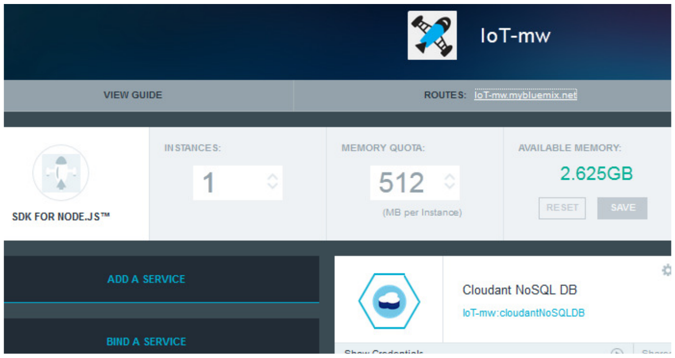


Export nodes to library

Filename: my iot-sensor-temp-flow

Step 7 – Use a Cloudant database to store the IoT sensor temperature data

The IoT starter application template has an instance of the Cloudant database created and bound to the application – you may have noticed the Cloudant service instance in the application definition in the Bluemix dashboard



ROUTES: IoT-mw.mybluemix.net

INSTANCES: 1

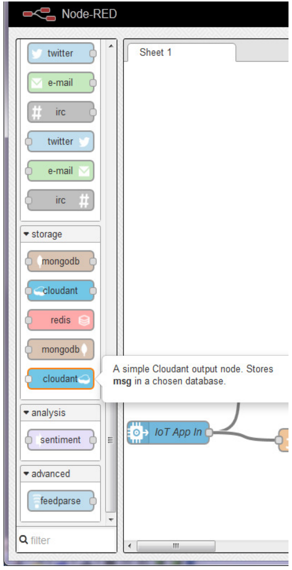
MEMORY QUOTA: 512 (MB per Instance)

AVAILABLE MEMORY: 2.625GB

Cloudant NoSQL DB
IoT-mw:cloudantNoSQLDB

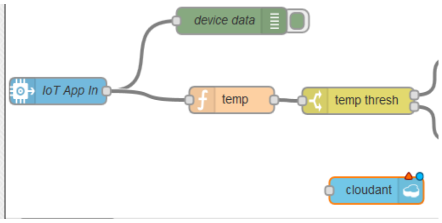
We could also create specific instances of any services in the Bluemix catalog for your applications' use, including Cloudant. Let's use the Cloudant NoSQL Database service with our application to store the IoT sensor device temperature data:

1. In the Node-RED flow editor, scroll down thru the node selections on the left panel to the category for Data storage, select the Cloudant output node

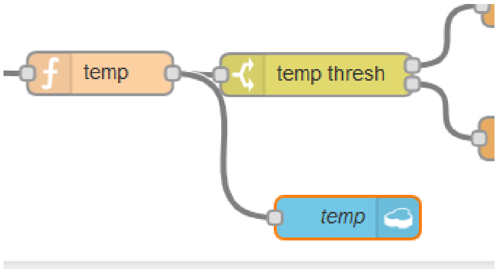


A simple Cloudant output node. Stores **msg** in a chosen database.

2. Left-click the Cloudant output node and drag it onto the flow editor to the below the "temp threshold" decision node



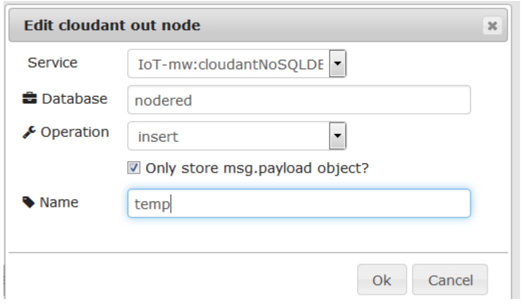
3. Select the output of the **temp** node in the flow and connect it to the input of the **cloudant** node.



A connector between the nodes is established, now let's specify the data storage service attributes and actions

4. Double-click the **cloudant** node , in the **Edit cloudant out node** dialog, select or enter the following in each field:

    - Service – select the service instance name for the **cloudantNoSQLDB** from the list (the instance name should be <appname>:cloudantNoSQLDB)
    - Database – specify **nodered**
    - Operation – select **insert**
    - Name – specify **temp**



Edit cloudant out node

Service: IoT-mw:cloudantNoSQLDE

Database: nodered

Operation: insert

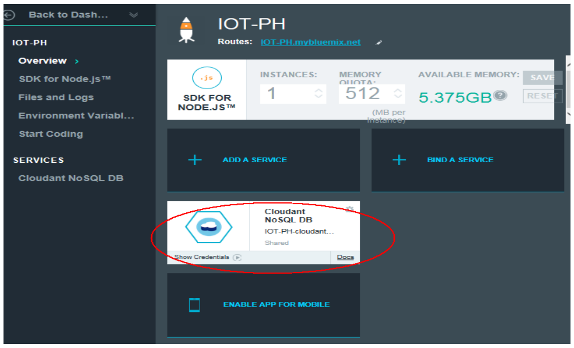
☑ Only store msg.payload object?

Name: temp

Click **OK,** then click **Deploy** to redeploy the flow

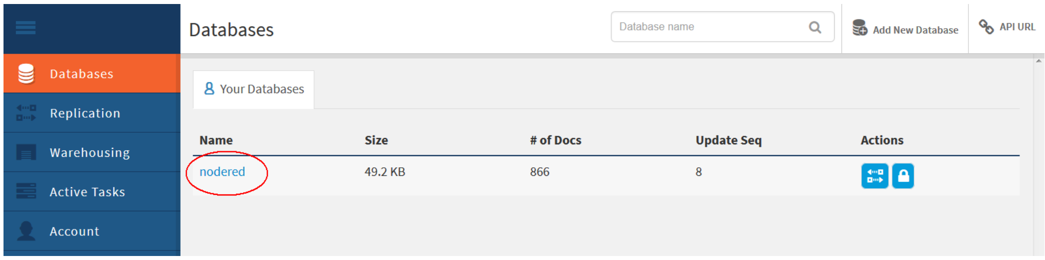
Entries should now be inserting to the Cloudant "nodered" database in our Cloudant data storage instance for the application, we will use the Cloudant console to examine them:

5. In your Bluemix Dashboard, select your IoT application and locate the **Cloudant NoSQLDB** service instance, click on the service instance:



6. The Cloudant service dialog presents options to work with Cloudant, click on **Launch** to launch the Cloudant web console for your instance.

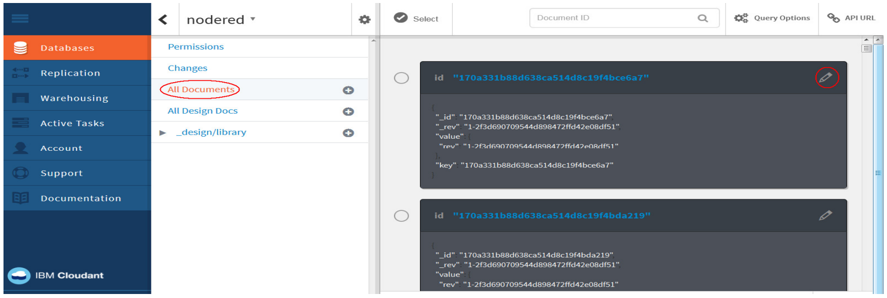
7. Click on **Databases –** the list of Cloudant databases in your instance are displayed, there should be only one, **nodered**



Name	Size	# of Docs	Update Seq	Actions
nodered	49.2 KB	866	8	

8. Click on the **nodered** database and select **All documents.**Database entries have stored the sensor events' data temperature object from our IoT applications's Node-RED flow

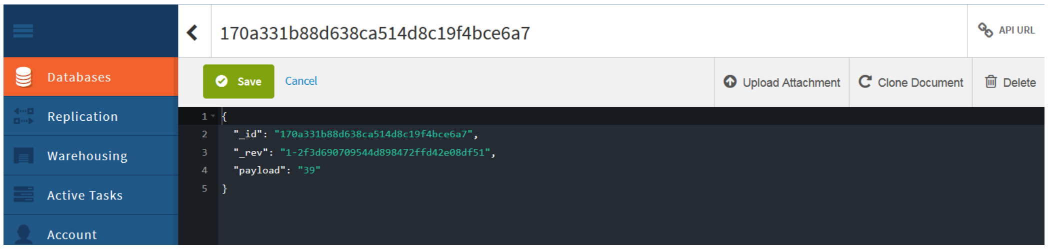
9. Select one of the entries by hovering over the entry – the **Edit doc** button should become active, select the **Edit doc** button



```
id "170a331b88d638ca514d8c19f4bce6a7"
{
 "_id": "170a331b88d638ca514d8c19f4bce6a7",
 "_rev": "1-2f3d690709544d898472ffd42e08df51",
 "value": {
  "rev": "1-2f3d690709544d898472ffd42e08df51"
 },
 "key": "170a331b88d638ca514d8c19f4bce6a7"
}

id "170a331b88d638ca514d8c19f4bda219"
{
 "_id": "170a331b88d638ca514d8c19f4bda219",
 "_rev": "1-2f3d690709544d898472ffd42e08df51",
 "value": {
  "rev": "1-2f3d690709544d898472ffd42e08df51"
```

10. The data object from the Node-RED flow for our IoT application is shown, we selected to store the "temp" payload attribute showing the sensor reading event's temperature setting - the other attributes for "id" and "rev" are generated by the Cloudant NoSQL DB storage service:

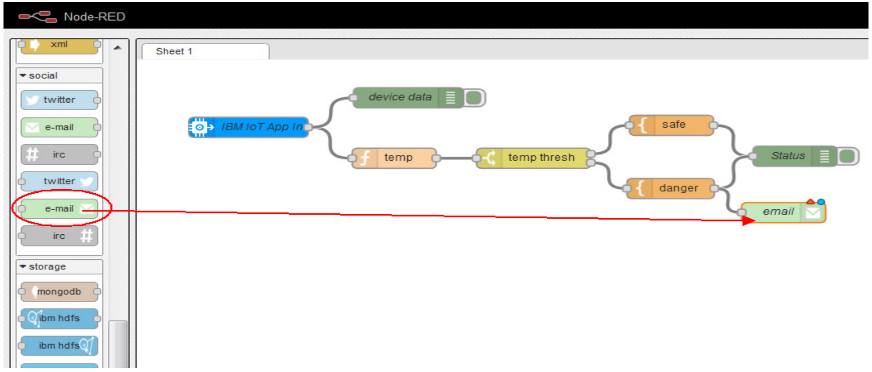


```
{
  "_id": "170a331b88d638ca514d8c19f4bce6a7",
  "_rev": "1-2f3d690709544d898472ffd42e08df51",
  "payload": "39"
}
```

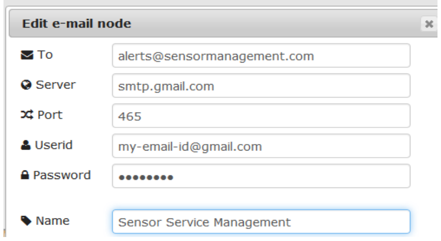
Step 8 – Integrate email into the flow to send "danger" level IoT sensor temperature data events for remediation or action

We'll now add email to the flow in order to send events for critical "danger" level temperature alerts to business entities for attention and any remediation actions required.

1. In the Node-RED flow editor, scroll down thru the node selections on the left panel to the category for Social, select the e-mail output node and drop it onto the flow to the right of the "danger" temperature decision node. Also create a connector from the output of the "danger" temperature decision node to the input of the email node.



2. Double-click the new output node and enter configuration settings for your e-mail account you'd like to use to send the email from the flow – any email service provider can be used, so just enter the destination recipient email address, email server, (optional) port, userid, and password for the email account to be used to send the notification:



Edit e-mail node

To: alerts@sensormanagement.com
Server: smtp.gmail.com
Port: 465
Userid: my-email-id@gmail.com
Password: ••••••••
Name: Sensor Service Management

Content of the e-mail will be set from the msg.payload property, with a subject line of msg.topic.

Click **OK,** then click **Deploy** to redeploy the flow

Use the sensor device simulator to generate sensor events for temperature reading above the danger level of 40 degress – alerts should now be sent to the configured e-mail address.
Congratulations – you successfully completed the IoT hands-on exercise using a Node-RED flow to work with events from from a simulated sensor device, see how to construct and debug flows in Node-RED and use Cloudant to store data from the Node-RED IoT flow.

More information on Node-RED and the IBM Internet of Things can be found:

http://nodered.org
https://www.ibmdw.net/iot/
Internet of Things Homepage
Connected Car – Traffic Simulator
Connected Car – Traffic Simulator – how to implement the IoT App
Recommended technical articles and examples:

- Bluemix and the Internet of Things
- Build a cloud-ready temperature sensor with the Arduino Uno and the IBM IoT Foundation, Part 1: Build the circuit and set up the environment
- Build a cloud-ready temperature sensor with the Arduino Uno and the IBM IoT Foundation, Part 2: Write the sketch and connect to the IBM IoT Foundation Quickstart
- Build a cloud-ready temperature sensor with the Arduino Uno and the IBM IoT Foundation, Part 3: Build a custom application with Node-RED
- Build a cloud-ready temperature sensor with the Arduino Uno and the IBM IoT Foundation, Part 4: Deploy a query GUI