



Connecting Devices to the IBM Watson IoT Foundation with Node Red

IOT Developer | Miracle Innovation Labs

Aditya Chinni

Lead – Miracle Innovation Labs
Miracle Software Systems, Inc.

February 18, 2016

Connecting Devices to the IBM Bluemix IoT Foundation with Node Red

Goal

In this lab the users will try and connect a Raspberry Pi to the IBM Internet of Things Foundation and get the sensor data to the IoT F Service Dashboard.

Pre-Requisites

The following installations will need to be completed for this lab to be run successfully,

- Browser for accessing IBM Bluemix
- Active Email ID for registering to IBM Bluemix

Technology Involved

- Raspberry Pi and Linux Commands
- IBM Bluemix
- IBM IoT Foundation

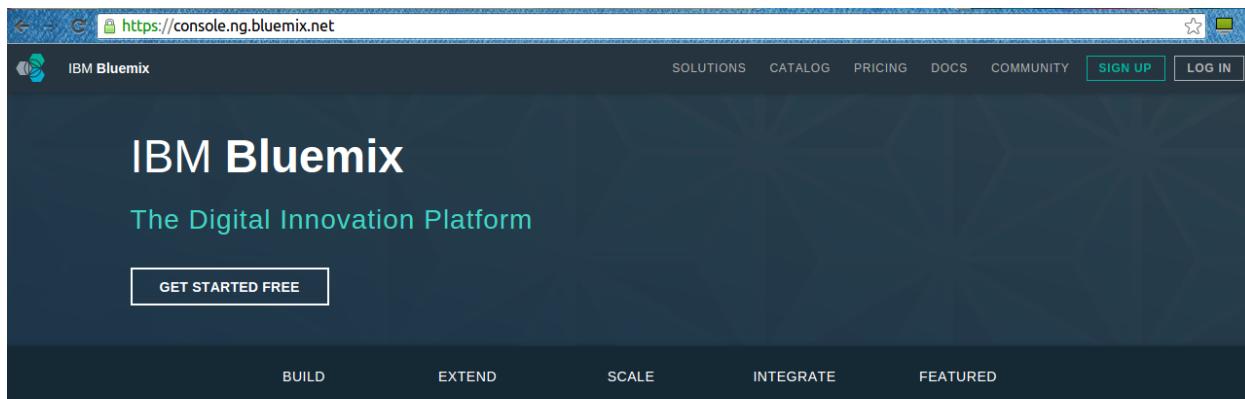
Lab Steps

So, let us get started with the lab!

#1 | Access Bluemix

The first step will be to make sure that we have access to the IBM Bluemix Console with either the free trial option (or) the paid subscription option.

Login to Bluemix at <http://bluemix.net> (or) Register today at <https://console.ng.bluemix.net/registration/>



The screenshot shows the IBM Bluemix homepage. At the top, there's a navigation bar with links for SOLUTIONS, CATALOG, PRICING, DOCS, and COMMUNITY, along with buttons for SIGN UP and LOG IN. The main title "IBM Bluemix" is prominently displayed, followed by the subtitle "The Digital Innovation Platform". Below this is a large "GET STARTED FREE" button. At the bottom of the main section, there are five categories: BUILD, EXTEND, SCALE, INTEGRATE, and FEATURED.

Build your apps, your way.

Use a combination of the most prominent open-source compute technologies to power your apps. Then, let Bluemix handle the rest.

Instant Runtimes
App-centric runtime

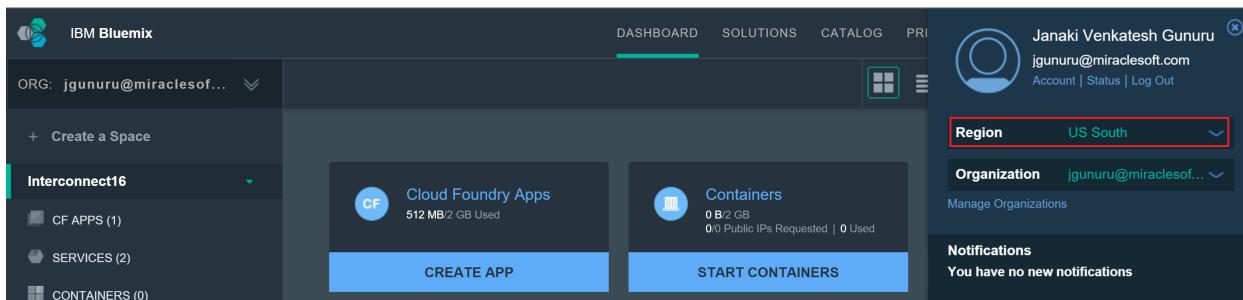
IBM Containers

Portable and consistent delivery

Virtual Machines

Get the most flexibility and

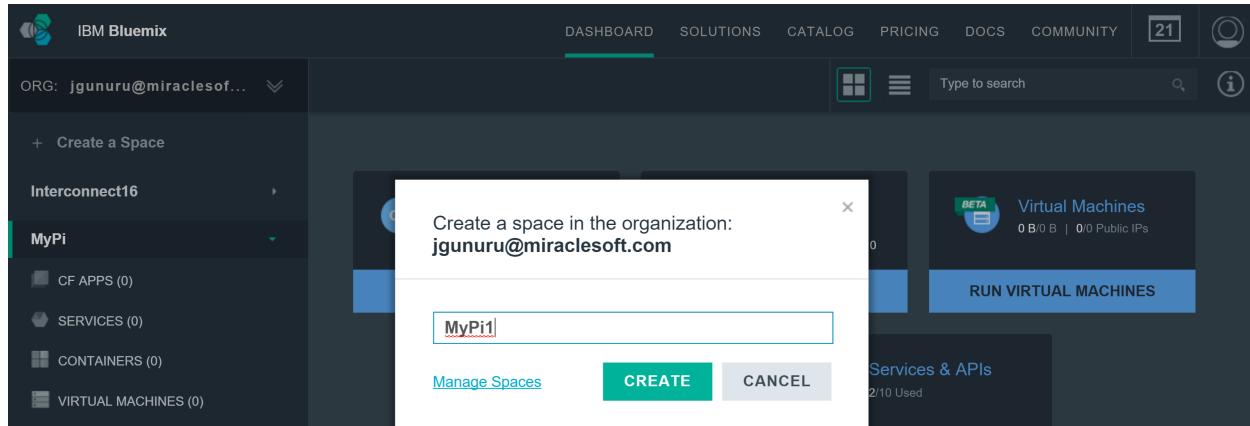
After you login, you can see the dashboard where you can take a look at your applications and services. You can also go to the profile icon at the top and change which Cloud Availability Region you are working in. Make sure you select **US South** region, as IoT Foundation Service is only available in that region.



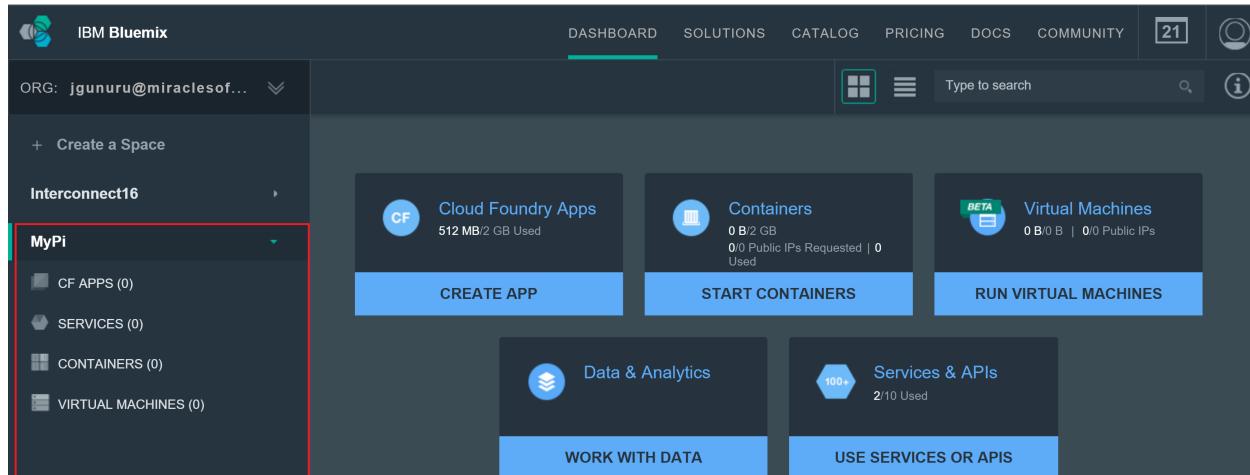
The screenshot shows the IBM Bluemix dashboard. On the left, there's a sidebar with options like "Create a Space", "Interconnect16", "CF APPS (1)", "SERVICES (2)", and "CONTAINERS (0)". The main dashboard area has two cards: "Cloud Foundry Apps" (512 MB / 2 GB Used) with a "CREATE APP" button, and "Containers" (0 B / 2 GB, 0.0 Public IPs Requested | 0 Used) with a "START CONTAINERS" button. On the right, there's a user profile for "Janaki Venkatesh Gunuru" with an email "jgunuru@miraclesoft.com" and links for "Account", "Status", and "Log Out". A dropdown menu shows the "Region" set to "US South" and the "Organization" set to "jgunuru@miraclesof...". A "Manage Organizations" link is also present. At the bottom, there's a "Notifications" section stating "You have no new notifications".

#2 | Create Space

Click on the “Create a Space” on left side of the Dashboard to create a space in Bluemix Cloud.

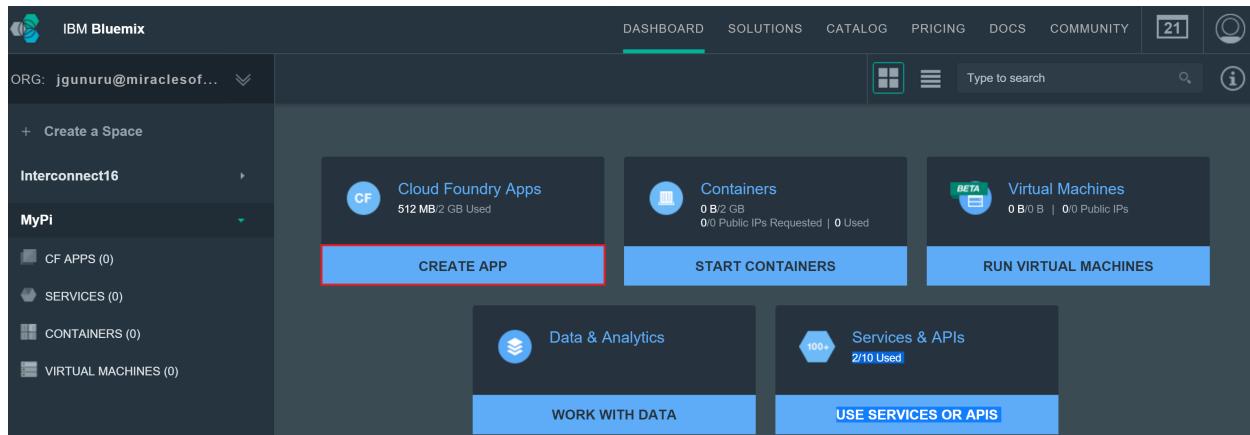


Once the space is created, you can find it at the left side menu of the Dashboard.

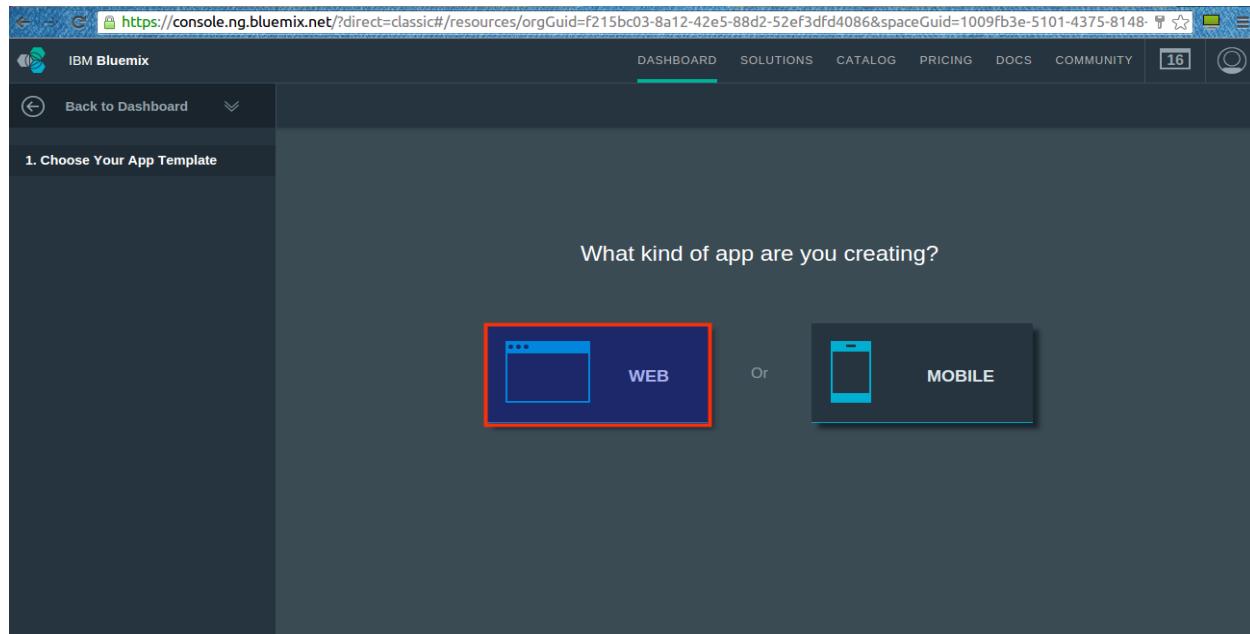


#3 | Create Application

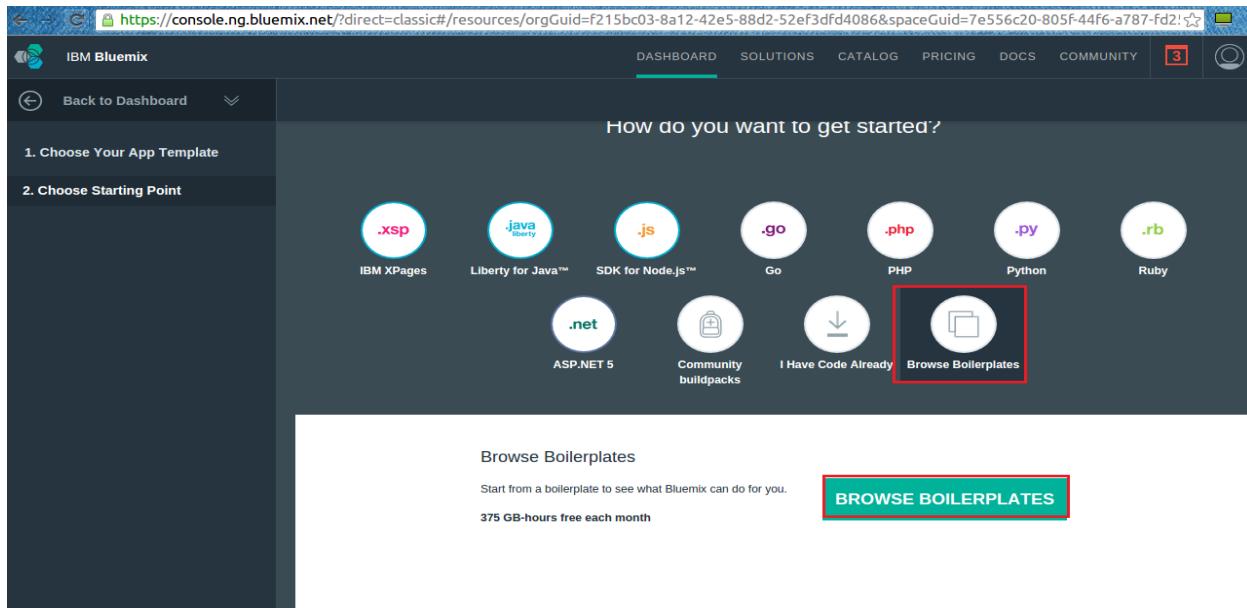
Click on the CloudFoundry Apps Module on the Dashboard to start creating the application.



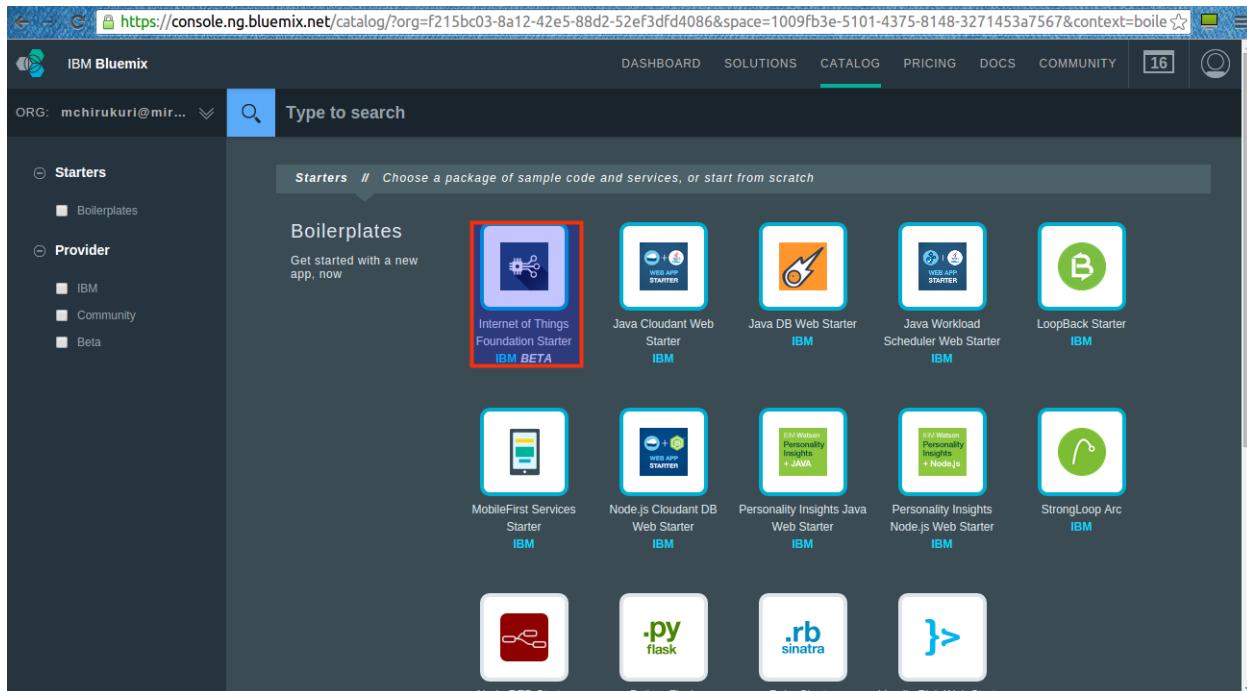
When prompted for the kind of application you want to create please select Web Application.



Select “Browse Boilerplates” in the page named “How do you want to get started?”. Click on “Browse Boilerplates” button.



You will be provided with available Boilerplates, Select “IoT Foundation Starter”.



This Boilerplate will have “SDK for Node.js” and “Cloudant NoSQL DB” services by default.

Give a unique name to your application here and click on “Create”. Application Names must be unique as they will be on a public domain.

IBM Bluemix

DASHBOARD SOLUTIONS CATALOG PRICING DOCS COMMUNITY 21

Back to Boilerplates

Internet of Things Foundation Starter

SDK for Node.js™

Cloudant NoSQL DB

Create an app:

Space: MyPi

Name: Enter new app name

Host: **MyPi1006**

Domain: mybluemix.net

Selected Plan:

SDK for Node.js™

Default

Cloudant NoSQL DB

Shared

Note : Once created, the application will take about 2 minutes for staging and start running.

IBM Bluemix

DASHBOARD SOLUTIONS CATALOG PRICING DOCS COMMUNITY 21

Back to Dashboard... ▾

MyPi1006

Overview

SDK for Node.js™

Files

Logs

Environment Variables

Start Coding >

SERVICES

cloudantNoSQLDB

Your application is staging. <http://MyPi1006.mybluemix.net>

Getting Started with:

Internet of Things

Create Internet of Things apps with Node-RED visual editor and the Internet of Things service.

Start coding with Internet of Things

1 After your application has started, click on the **Routes URL** or enter the following URL in a browser:

<http://<yourhost>.mybluemix.net>

The Node-RED for Internet of Things landing page displays.

#4 | Adding IoT F service to our application

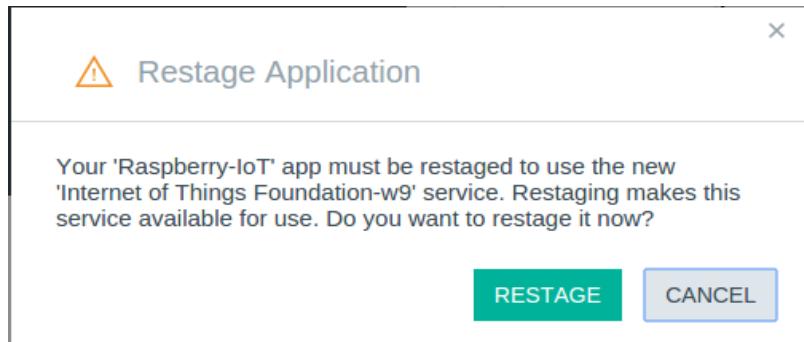
Go to “Catalog” and select “Internet of things” in the left side menu. Click on “Internet of Things Foundation”.

The screenshot shows the IBM Bluemix Catalog interface. On the left, there is a sidebar with categories like Compute, Services, and Internet of Things (which is highlighted with a red box). The main area displays a list of services under 'Services // The building blocks of any great app'. The 'Internet of Things Foundation' service by IBM is highlighted with a red box. Other services listed include 'IoT Real-Time Insights' and 'flowthings.io'. Below the services, there is a section titled 'Looking for more?' with a link to 'Bluemix Labs Catalog'.

Choose your created application in the “App” field and click on “Create”.

The screenshot shows the IBM Bluemix Catalog interface with the 'Internet of Things Foundation' service selected. On the right, there is a form for adding a new service. The 'Space' dropdown is set to 'MyPi', and the 'App' dropdown is set to 'Leave unbound'. The 'Service name:' field contains 'Internet of Things Foundation-wf'. Under 'Selected Plan:', the 'Free' plan is selected. A large green 'CREATE' button is at the bottom right of the form.

The application should be restaged to use the new service. When prompted, click on “Restage”.



Note : The application will take about 2 minutes for staging and start running again.

#5 | Accessing Raspberry Pi through Putty

To install putty on Windows, go to the following URL,

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Click on “**putty-0.66-installer.exe**”.

This will generally be a version we think is reasonably likely to work well. If you have a problem with the release version, it might be worth trying out the development snapshot (below) to see if we've already fixed the bug, before reporting it.

For Windows on Intel x86

PuTTY:	putty.exe	(or by FTP)	(signature)
PuTTYtel:	puttytel.exe	(or by FTP)	(signature)
PSCP:	pscp.exe	(or by FTP)	(signature)
PSFTP:	psftp.exe	(or by FTP)	(signature)
Plink:	plink.exe	(or by FTP)	(signature)
Pageant:	pageant.exe	(or by FTP)	(signature)
PUTTYgen:	puttygen.exe	(or by FTP)	(signature)

A .ZIP file containing all the binaries (except PuTTYtel), and also the help files

Zip file:	putty.zip	(or by FTP)	(signature)
-----------	---------------------------	-------------	-----------------------------

A Windows installer for everything except PuTTYtel

Installer:	putty-0.66-installer.exe	(or by FTP)	(signature)
------------	------------------------------------------	-------------	-----------------------------

Checksums for all the above files

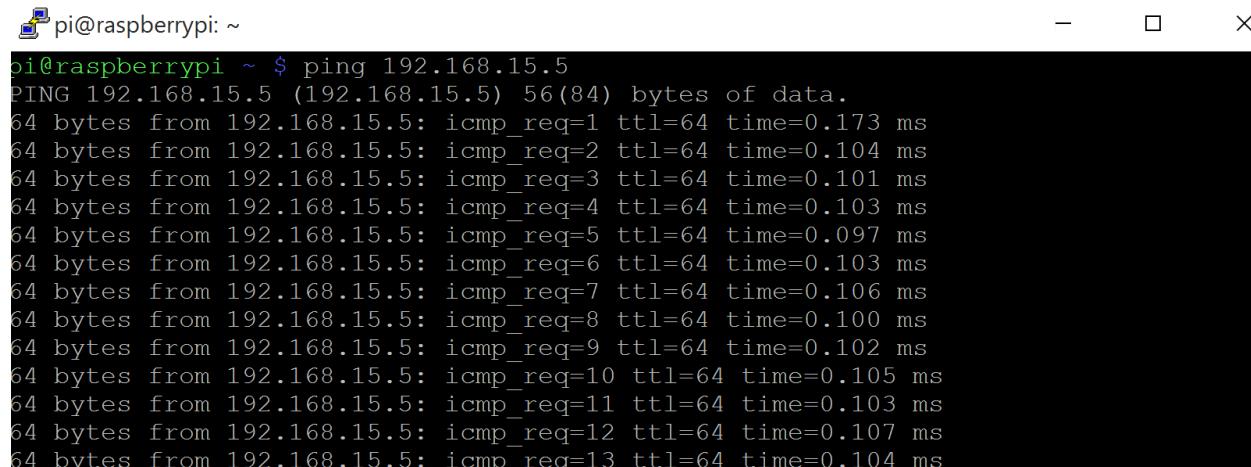
MD5	SHA1	GPG
---------------------	----------------------	---------------------

Click on the installer and go ahead and install Putty. Once installed open up the application.

#8 | Connecting Raspberry Pi to IoT Foundation

Power up the Raspberry Pi using a USB, which is provided along with the IoT Kit. Plug in a WiFi module for your Raspberry Pi to get connected to Internet. Ping the IP Address of your Raspberry Pi.

ping 192.168.15.5

A terminal window titled "pi@raspberrypi: ~" showing the output of a ping command. The command "ping 192.168.15.5" was run, and the terminal shows 16 ICMP echo requests being sent to the specified IP address. Each request is labeled with its sequence number (icmp_req=1 to icmp_req=16), TTL (64), time taken (e.g., 0.173 ms, 0.104 ms, etc.), and the source IP (192.168.15.5).

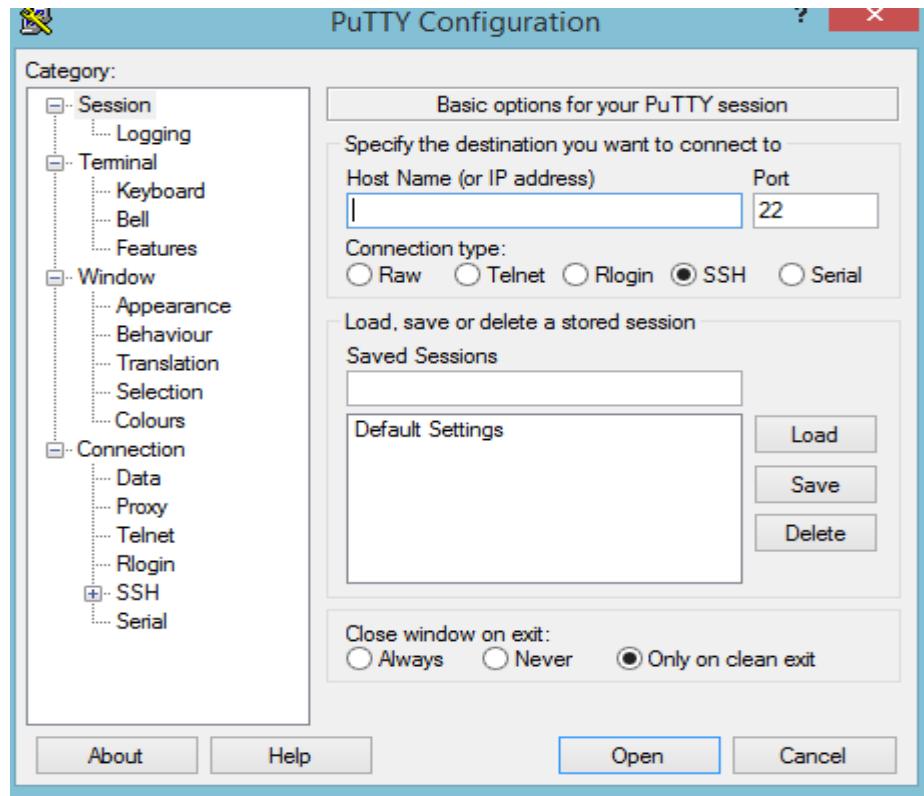
```
pi@raspberrypi ~ $ ping 192.168.15.5
PING 192.168.15.5 (192.168.15.5) 56(84) bytes of data.
64 bytes from 192.168.15.5: icmp_req=1 ttl=64 time=0.173 ms
64 bytes from 192.168.15.5: icmp_req=2 ttl=64 time=0.104 ms
64 bytes from 192.168.15.5: icmp_req=3 ttl=64 time=0.101 ms
64 bytes from 192.168.15.5: icmp_req=4 ttl=64 time=0.103 ms
64 bytes from 192.168.15.5: icmp_req=5 ttl=64 time=0.097 ms
64 bytes from 192.168.15.5: icmp_req=6 ttl=64 time=0.103 ms
64 bytes from 192.168.15.5: icmp_req=7 ttl=64 time=0.106 ms
64 bytes from 192.168.15.5: icmp_req=8 ttl=64 time=0.100 ms
64 bytes from 192.168.15.5: icmp_req=9 ttl=64 time=0.102 ms
64 bytes from 192.168.15.5: icmp_req=10 ttl=64 time=0.105 ms
64 bytes from 192.168.15.5: icmp_req=11 ttl=64 time=0.103 ms
64 bytes from 192.168.15.5: icmp_req=12 ttl=64 time=0.107 ms
64 bytes from 192.168.15.5: icmp_req=13 ttl=64 time=0.104 ms
```

Note : You should have been provided the IP Address before the lab, else you can connect a screen and get the IP Address of the device.

Using that IP Address go ahead and connect to the RPi terminal with Putty. The default username and password for the RPis are “pi” and “raspberry” respectively.

Note : If you are a Mac (or) Linux user you can directly connect to the RPis with the SSH command in your terminal.

ssh pi@<IP Address>



Give the IP address of your Raspberry Pi device in the “Host Name” field and click on “Open” and when prompted give login name and password.

A screenshot of a terminal window titled 'pi@milpi: ~'. The window displays a standard Debian Linux login screen. It shows the user 'pi' has logged in from IP '192.168.137.4'. The terminal then displays the standard Debian copyright notice and ends with a prompt 'pi@milpi ~ \$'. The background of the terminal window is black with white text.

Download the `iot-raspberrypi` installer from GitHub using the command,

```
curl -LO https://github.com/ibm-messaging/iot-raspberrypi/releases/download/1.0.2.1/iot_1.0-2_armhf.deb
```

```
pi@milpi ~ $ curl -LO https://github.com/ibm-messaging/iot-raspberrypi/releases/download/1.0.2.1/iot_1.0-2_armhf.deb
% Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
                                         Dload  Upload   Total   Spent   Left  Speed
100  596     0  596      0      0   367       0  --:--:--  0:00:01  --:--:--  525
100 110k  100 110k      0      0 17354       0  0:00:06  0:00:06  --:--:-- 30576
pi@milpi ~ $
```

Install the downloaded debian package with the command,

```
sudo dpkg -i iot_1.0-2_armhf.deb
```

```
pi@milpi ~ $ sudo dpkg -i iot_1.0-2_armhf.deb
Selecting previously unselected package iot.
(Reading database ... 78904 files and directories currently installed.)
Unpacking iot (from iot_1.0-2_armhf.deb) ...
Setting up iot (1.0-1) ...
Starting the iot program
pi@milpi ~ $
```

The iot process starts and publishes events to the IBM Internet of Things Foundation Quickstart. The process runs as a system service, and starts whenever Raspberry Pi starts. To verify the process is running, use the command,

```
service iot status
```

```
pi@milpi ~ $ service iot status
[ ok ] iot is running.
pi@milpi ~ $
```

To know the MAC ID (device ID) of your device, use the following command,

```
service iot getdeviceid
```

Note down the device ID for later purpose.

```
pi@milpi ~ $ service iot getdeviceid
The device ID is b827eba64d35
For Real-time visualization of the data, visit http://quickstart.internetofthings.ibmcloud.com/?deviceId=b827eba64d35
pi@milpi ~ $
```

#8 | Register your device with the IoT Foundation

Go back to your Bluemix account. Click on your application.

The screenshot shows the IBM Bluemix dashboard. On the left, there's a sidebar with options like 'Create a Space', 'Interconnect16', and 'MyPi'. Under 'MyPi', it lists 'CF APPS (1)', 'SERVICES (2)', 'CONTAINERS (0)', and 'VIRTUAL MACHINES (0)'. The main area displays the 'Data & Analytics' and 'Services & APIs' sections. Below that, under 'Applications', the 'MyPi1006' app is shown as 'Running'. The app icon features a blue and white design with a gear and a lightning bolt.

Click on “Internet of Things Foundation” service.

This screenshot shows the 'MyPi1006' application details page. The left sidebar includes links for 'Overview', 'SDK for Node.js™', 'Files', 'Logs', 'Environment Variables', 'Start Coding', and 'SERVICES'. Under 'SERVICES', the 'Cloudant NoSQL DB' and 'Internet of Things Foundation' services are listed. The 'Internet of Things Foundation' service is highlighted with a red border around its icon and 'Show Credentials' button. The main panel shows the app's configuration with 1 instance, 512 MB memory quota, and 1.000 GB available memory. It also features sections for 'ADD A SERVICE OR API' and 'BIND A SERVICE OR API'. On the right, there's an 'APP HEALTH' section indicating the app is running, and an 'ACTIVITY LOG' showing recent events like app creation and route updates. At the bottom, there's a link to 'Estimate the cost of this app'.

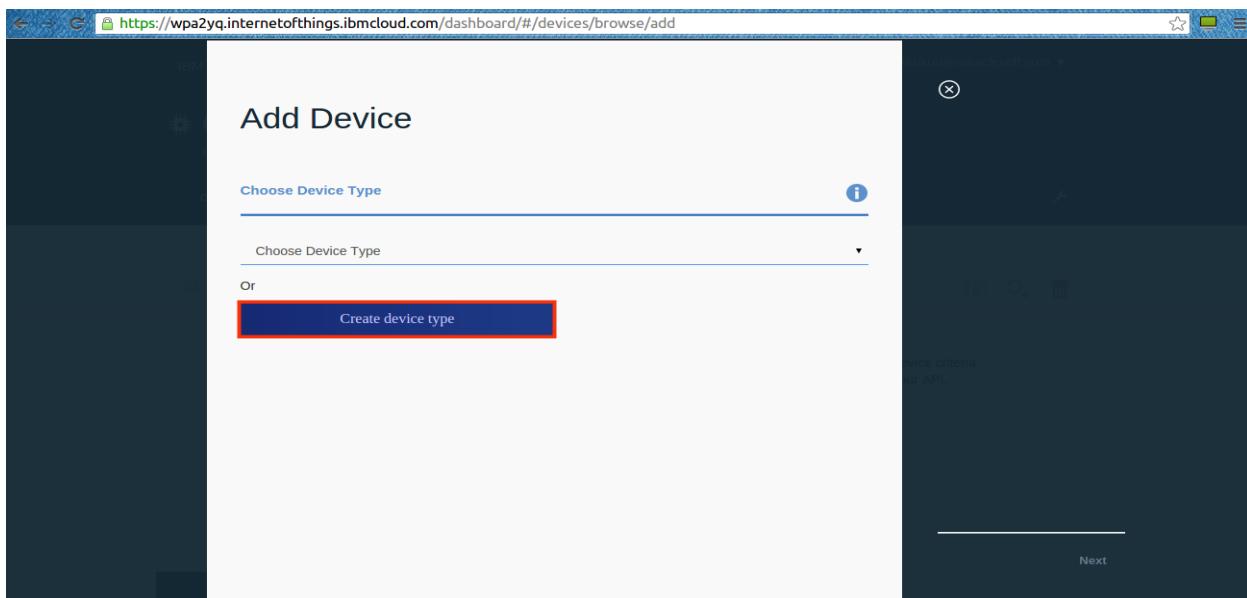
Click “Launch Dashboard” button. This will take you to your IoT Foundation Organization space.

The screenshot shows the IBM Bluemix interface for the Internet of Things Foundation. On the left, there's a sidebar with options like Overview, SDK for Node.js™, Files, Logs, Environment Variables, Start Coding, and SERVICES (Cloudant NoSQL DB, Internet of Things Foundation). The main content area has three columns. The first column contains a "Connect your devices" section with a "Launch dashboard" button highlighted with a red box. The second column contains a "Learn how to build your app" section with a "Go to docs" button. The third column contains a "Learn how to extend your app" section with a list of services: Twilio (Third Party), Cloudant NoSQL DB (IBM), Dash DB (IBM), Geospatial Analytics (IBM), Time Series Database (IBM), and IBM Analytics for Hadoop (IBM).

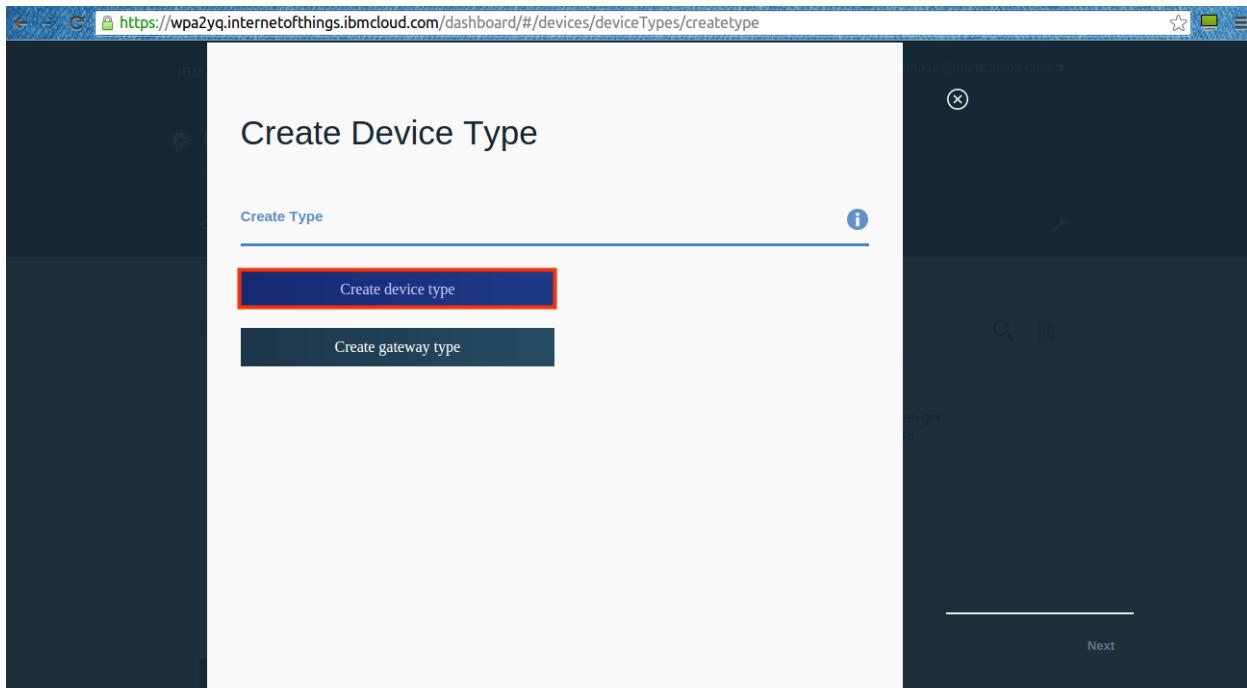
Observe that a new Organization is created with a unique ID. As the Organization is new, there will be no registered devices. Click on “Add a device”.

The screenshot shows the IBM Watson IoT Platform Overview dashboard. It features four main cards: DEVICE TYPES (No devices have been added, with an "Add Device" button highlighted with a red box), STORAGE (0.0 MB used today, 0.0 MB this month, 0.0 MB previous month), DATA CONSUMED (0.0 MB consumed today), and DEVICE (0 currently registered).

Click on “Create Device Type”.



Observe that we have 2 options. As we want to create a Device Type and not a Gateway, click “Create device type”.



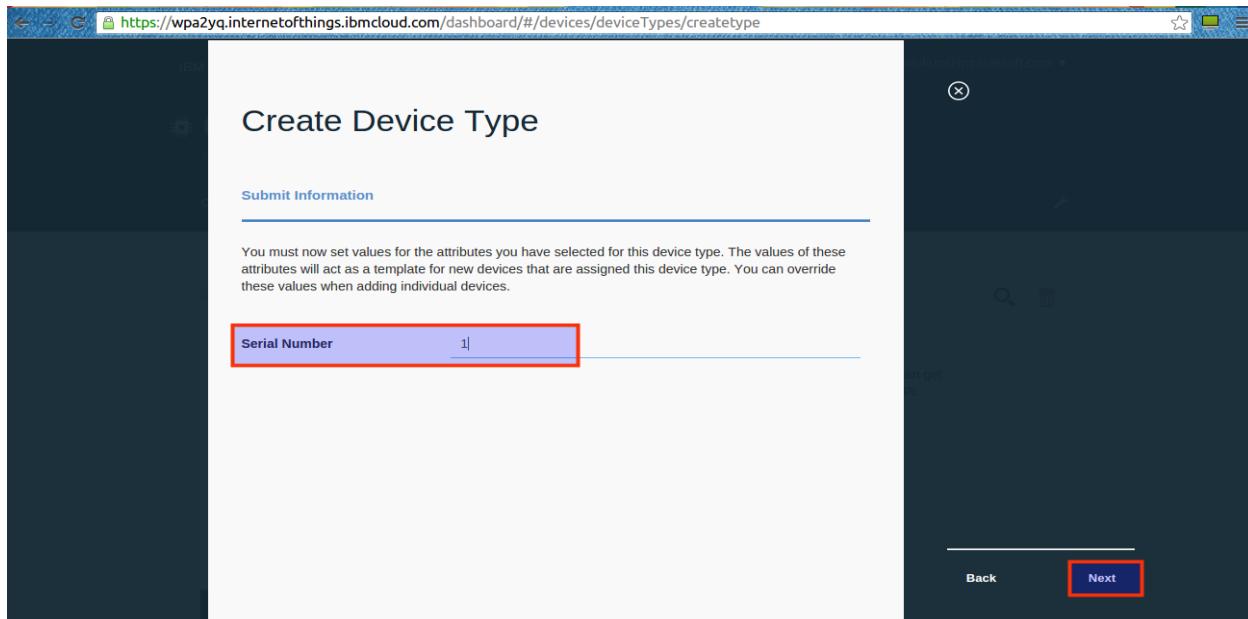
Specify a name for the device type to be added and give a description. For example you can give “**mypi**” for “Name” field. Click on “Next”.

The screenshot shows a web browser window with the URL <https://wpa2yq.internetofthings.ibmcloud.com/dashboard/#/devices/deviceTypes/createtype>. The page title is "Create Device Type". Under the "General Information" section, the "Name" field contains "mypi" and is highlighted with a red border. Below the field, a note states: "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." The "Description" section has a placeholder "Enter description" and a note: "The device type description can be used for a more descriptive way of identifying the device type." At the bottom right are "Back" and "Next" buttons, with "Next" being highlighted with a red border.

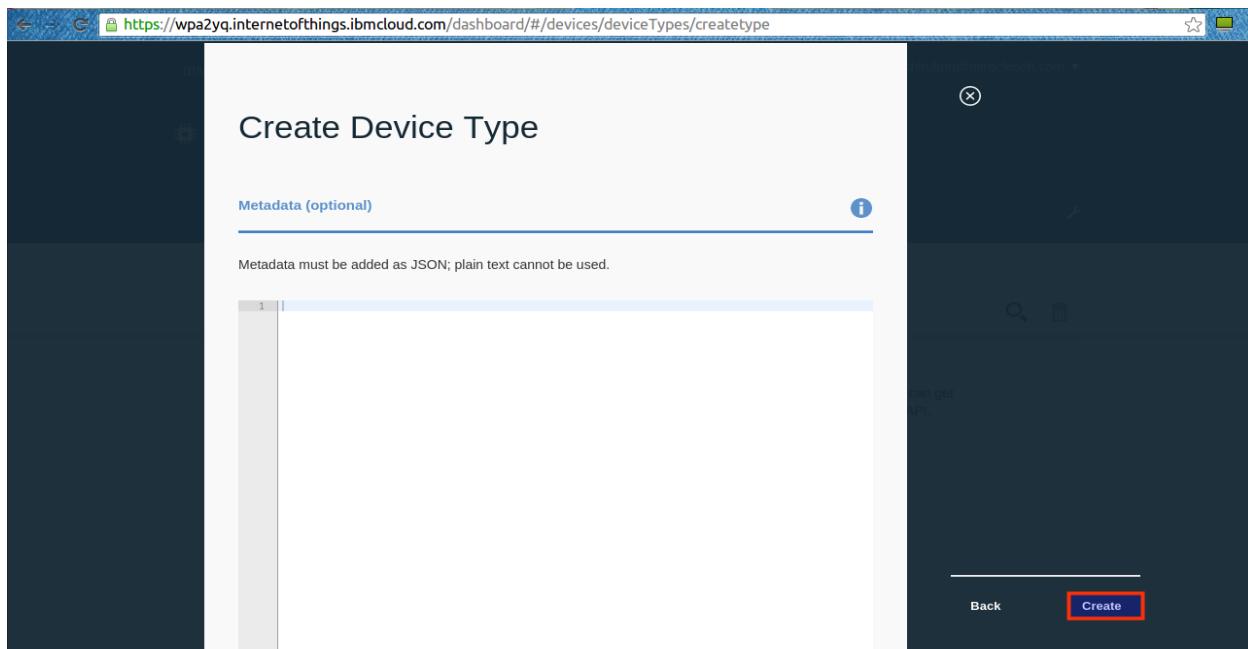
Select one or more attributes and click on “Next”.

The screenshot shows the same web browser window at the "Define Template" step. The page title is "Create Device Type". Under the "Define Template" section, a note says: "Use the options below to select attributes for the device type. All of these attributes are optional. They will be used as a template for new devices that are assigned this device type. Attributes you do not define may still be edited individually on devices that are assigned this device type." A grid of checkboxes allows selecting attributes: "Serial Number" (checked), "Description", "Manufacturer", "Firmware Version", "Model", "Hardware Version", "Class", and "Descriptive Location". The "Serial Number" checkbox is highlighted with a blue border. At the bottom right are "Back" and "Next" buttons, with "Next" being highlighted with a red border.

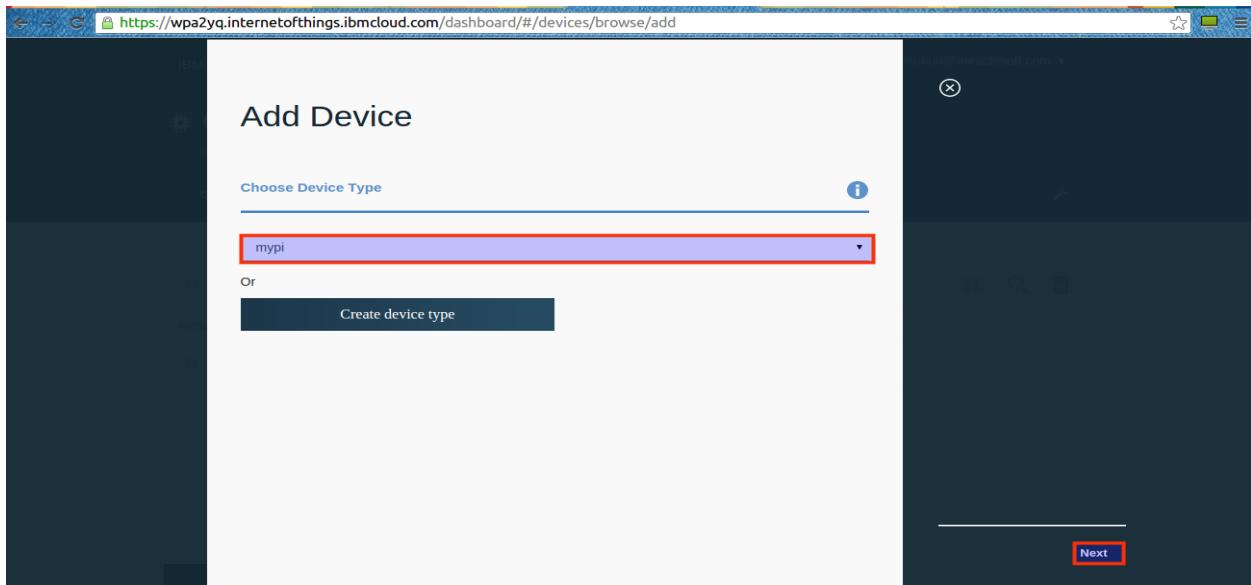
Define the attributes and click on “Next”. For example, you can give “1” for “Serial Number” field.



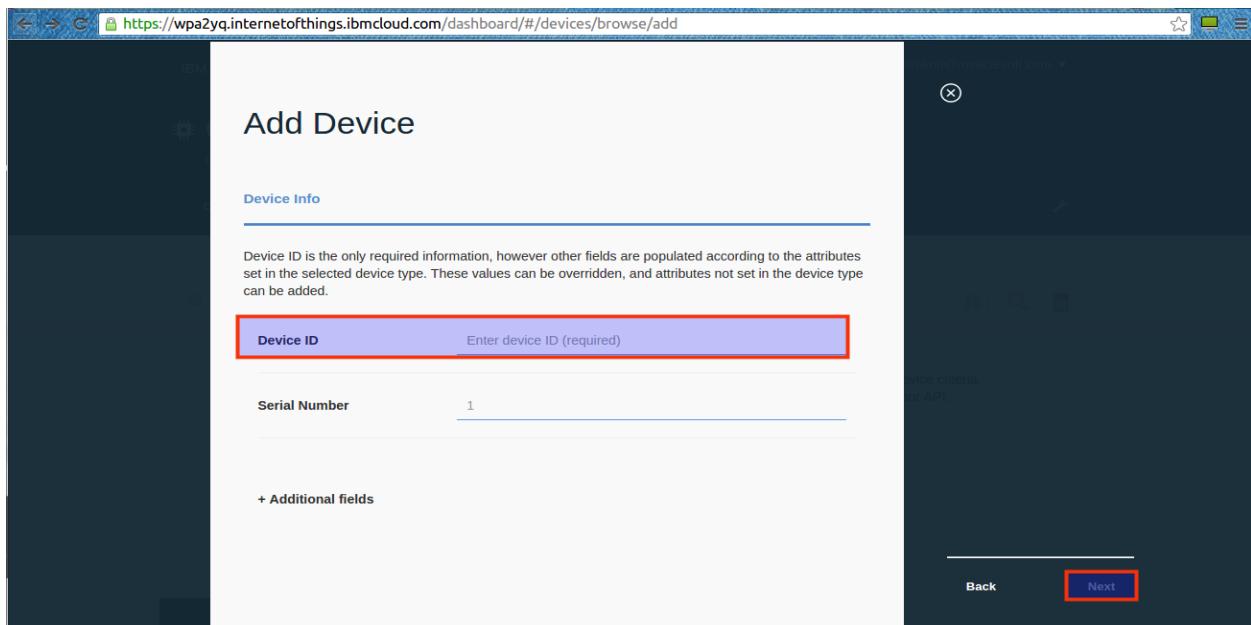
If you want to insert any Metadata, you can insert it here. Otherwise just click on “Create”.



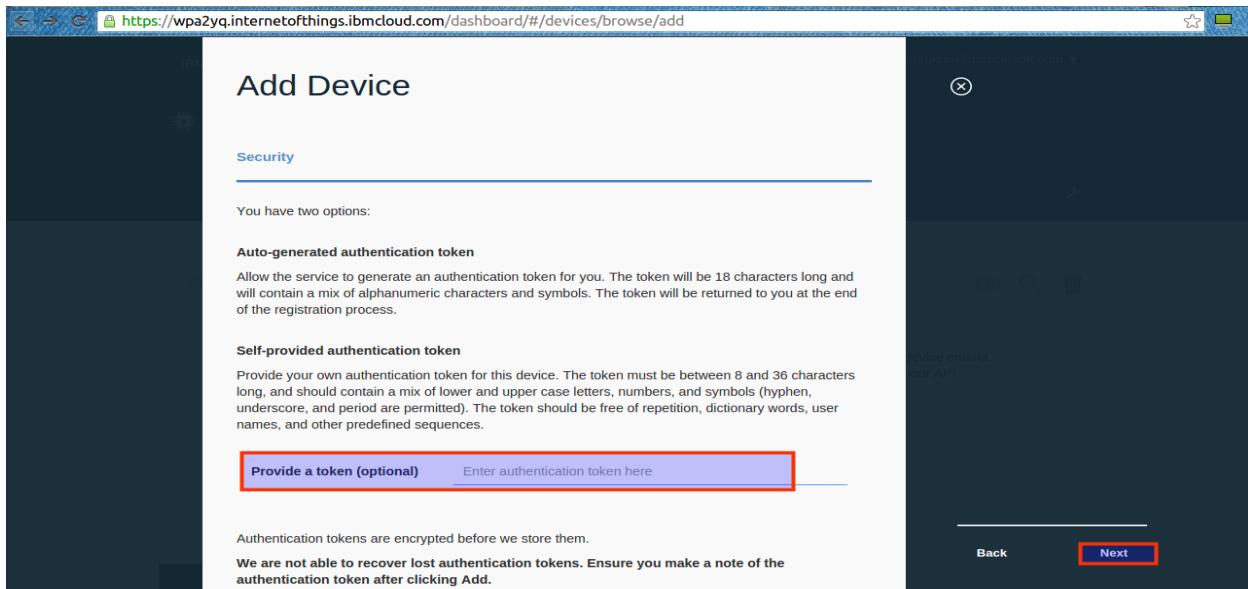
Once the Device Type is successfully created, we can add a device in that. Choose the device type that was created in the last step and click on “Next”.



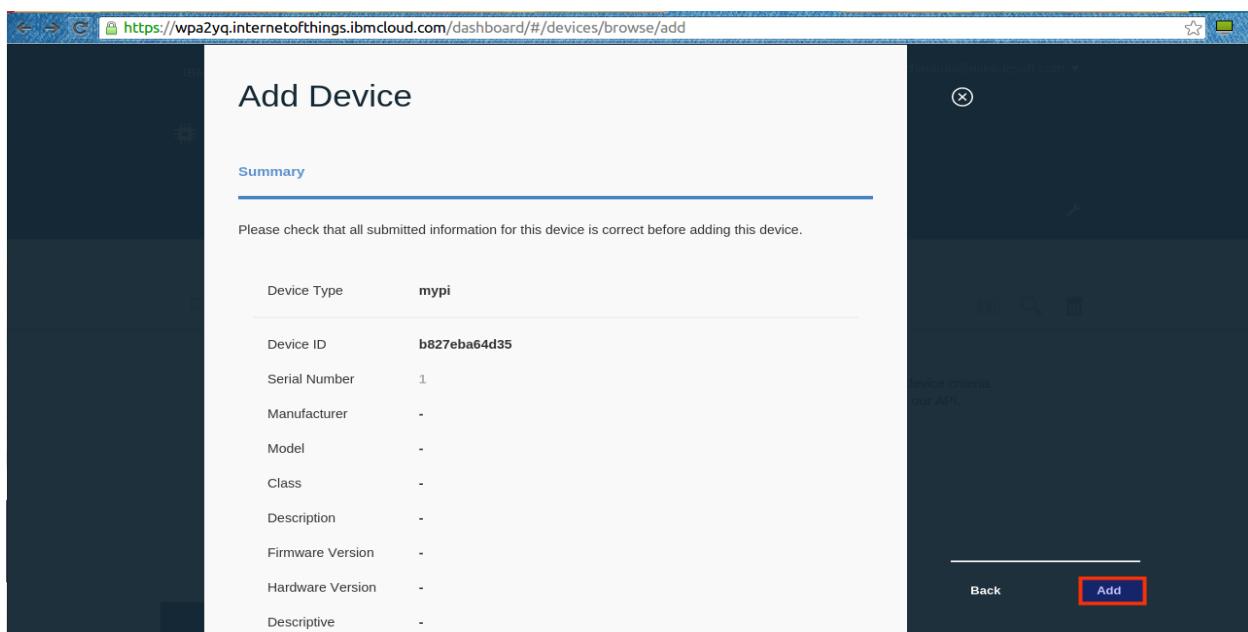
Give a unique Device ID (You have to give your Raspberry Pi Mac address of your Raspberry Pi device which you got in previous step) in the “Device ID” field and click on “Next”.



If you want to insert any Metadata, you can insert it here. Click on “Next”. In the next page, you can either add your own authentication token, or allow the IoT Foundation to generate a token for you. If you want to add your own token, enter the token. Otherwise leave the field empty. Click on “Next”.



In the next page, summary of the device will be given. Verify and click on “Add”.



The device is successfully added to the Organization. The registration details will be provided. To get your device connected, you need to add these credentials to your device.

The screenshot shows a web browser window with the URL <https://wpa2yq.internetofthings.ibmcloud.com/dashboard/#/devices/browse/drilldown/mypi/b827eba64d35>. The page title is "Device b827eba64d35". Under the heading "Your Device Credentials", there is a table with the following data:

Organization ID	wpa2yq
Device Type	mypi
Device ID	b827eba64d35
Authentication Method	token
Authentication Token	-IBDUK4R1ODFY0TVX*

A note below the table states: "Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the device to generate a new authentication token." A link "Find out how to add these credentials to your device" is also present.

Click the cross button to go back to the main dashboard. Observe that the device is added in your Organization but still not yet connected.

The screenshot shows the "DEVICES" tab selected on the IBM Internet of Things Foundation dashboard. The page title is "Organization ID: wpa2yq". The search bar contains "Device ID" and the results table shows one entry:

	b827eba64d35	mypi	Device	Feb 1, 2016 8:21:00 AM
--	--------------	------	--------	------------------------

A red box highlights the first row of the table.

#9 | Add the device.cfg file to your Device

Stop the IoT process using the command,

```
sudo service iot stop
```

```
pi@milpi ~ $ sudo service iot stop
sudo: unable to resolve host milpi
Stopping the program
Stopping the IoT-Raspberry Pi program with pid : 2326
pi@milpi ~ $
```

Create a new **device.cfg** file under '/etc/iotsample-raspberrypi' using the command,

```
sudo nano /etc/iotsample-raspberrypi/device.cfg
```

```
pi@milpi ~ $ sudo nano /etc/iotsample-raspberrypi/device.cfg
```

Fill in the configuration file with the details that you copied when registering the device. The format should be as follows,

```
org = <your-Organization-name>
type = <your-Device-Type>
id = <your-Device-ID>
auth-method = "token"
auth-token = <Your-Authentication-Token>
```

Save the file by pressing CTRL-X.

```
GNU nano 2.2.6      File: /etc/iotsample-raspberrypi/device.cfg      Modified

#Device Configuration
org=wpa2yq
type=mypi
id=b827eba64d35
auth-method=token
auth-token=-!BDUk4R1ODfYOTVX*^

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text^T To Spell
```

Start the IoT process for the device to start in registered mode using the command,

sudo service iot start

```
pi@milpi ~ $ sudo service iot start
sudo: unable to resolve host milpi
Starting the iot program
pi@milpi ~ $
```

#10 | Check Device Data in IoT Dashboard

Go to your IoT Foundation Dashboard and click on the device created in the previous steps.

The screenshot shows the IBM Internet of Things Foundation dashboard at <https://wpa2yq.internetofthings.ibmcloud.com/dashboard/#/devices/browse>. The top navigation bar includes links for IBM Internet of Things Foundation, Quickstart, Service Status, Documentation, Blog, and a user account (mchirukuri@miraclesoft.com). The main title is "Organization ID: wpa2yq" with a subtitle "Bluemix Free (go to Bluemix service)". Below this, there are tabs for OVERVIEW, DEVICES (selected), ACCESS, and USAGE. Under the DEVICES tab, sub-tabs include Browse (selected), Diagnose, Action, and Device Types. A search bar and filter options (Device ID, Device Type, Class ID, Date Added, Location) are present. The results section shows "Results 1-0 of 0" and a single entry for a device with ID b827eba64d35, type mypi, added on Feb 3, 2016 at 10:26:34 AM. A "Add Device" button is at the bottom.

You can find the Device information like - Recent Events, Sensor Information, Logs, etc.

The screenshot shows the device details page for "Device b827eba64d35" at <https://wpa2yq.internetofthings.ibmcloud.com/dashboard/#/devices/browse/drilldown/mypi/b827eba64d35>. The page has a header "Device b827eba64d35". It contains sections for "Connection Information", "Recent Events", and "Sensor Information". The Connection Information section lists: Device ID (b827eba64d35), Device Type (mypi), Date Added (Wednesday, February 3, 2016), Added By (mchirukuri@miraclesoft.com), and Connection State (Connected on Wednesday, February 3, 2016 at 10:28:37 AM from 182.72.208.42 with a secure connection [Refresh](#)). The Recent Events section states "There are currently no events". The Sensor Information section has an "i" icon. The right side of the screen shows a dark sidebar with a user profile picture and the text "mchirukuri@miraclesoft.com" and "Logout".

Recent Events data will look something like below.

The screenshot shows a web browser window with the URL <https://wpa2yq.internetofthings.ibmcloud.com/dashboard/#/devices/browse/drilldown/mypi/b827eba64d35>. The page displays a table of recent events and sensor information. The event table has columns for Event, Format, and Time Received. The sensor information table has columns for Event, Datapoint, Value, and Time Received.

Event	Format	Time Received
status	json	Feb 3, 2016 10:29:11 AM
status	json	Feb 3, 2016 10:29:12 AM
status	json	Feb 3, 2016 10:29:13 AM
status	json	Feb 3, 2016 10:29:14 AM
status	json	Feb 3, 2016 10:29:15 AM
status	json	Feb 3, 2016 10:29:16 AM
status	json	Feb 3, 2016 10:29:17 AM
status	json	Feb 3, 2016 10:29:18 AM
status	json	Feb 3, 2016 10:29:19 AM
status	json	Feb 3, 2016 10:29:20 AM

Event	Datapoint	Value	Time Received
-------	-----------	-------	---------------

We can also view the JSON data present in each event by clicking on it.

The screenshot shows a detailed view of a selected event. The event is labeled "status (json)" and was received at 10:29:39 AM. The JSON data is displayed in a modal window:

```
{"d": {"myName": "myPi", "cputemp": 47.08, "cpuload": 0.01, "memoryusage": 4224.0, "sine": "-0.92"}}
```

Sensor Information will look like this,

Event	Datapoint	Value	Time Received
status	d.myName	myPi	Feb 3, 2016 10:29:31 AM
status	d.cputemp	48.15	Feb 3, 2016 10:29:31 AM
status	d.cpuupload	0.01	Feb 3, 2016 10:29:31 AM
status	d.memoryusage	4224	Feb 3, 2016 10:29:31 AM
status	d.sine	0.92	Feb 3, 2016 10:29:31 AM

#11 | Creating Node Red Flow

Let's create a flow, which tells you whether the CPU temperature of your device is in Safe State or Critical State. Go to your Bluemix Application Dashboard and click on **Overview**.

Click on your application route.

The screenshot shows the IBM Bluemix dashboard for the application "MyPi1006". The left sidebar contains links for Overview, SDK for Node.js™, Files, Logs, Environment Variables, Start Coding, and SERVICES (Cloudant NoSQL DB, Internet of Things Foundation). The main area displays the application's configuration with 1 instance, 512 MB memory quota, and 1.000 GB available memory. It includes sections for "ADD A SERVICE OR API" and "BIND A SERVICE OR API". The "ACTIVITY LOG" section shows the following entries:

- 2/18/16 12:17 AM jgunuru@miraclesoft.com started MyPi1006 app
- 2/18/16 12:17 AM jgunuru@miraclesoft.com updated MyPi1006 app
 - changed routes
- 2/18/16 12:17 AM jgunuru@miraclesoft.com created MyPi1006 app

[Estimate the cost of this app](#)

This will redirect you to Node Red tool in Bluemix. Click on “Go to your Node Red flow editor”.

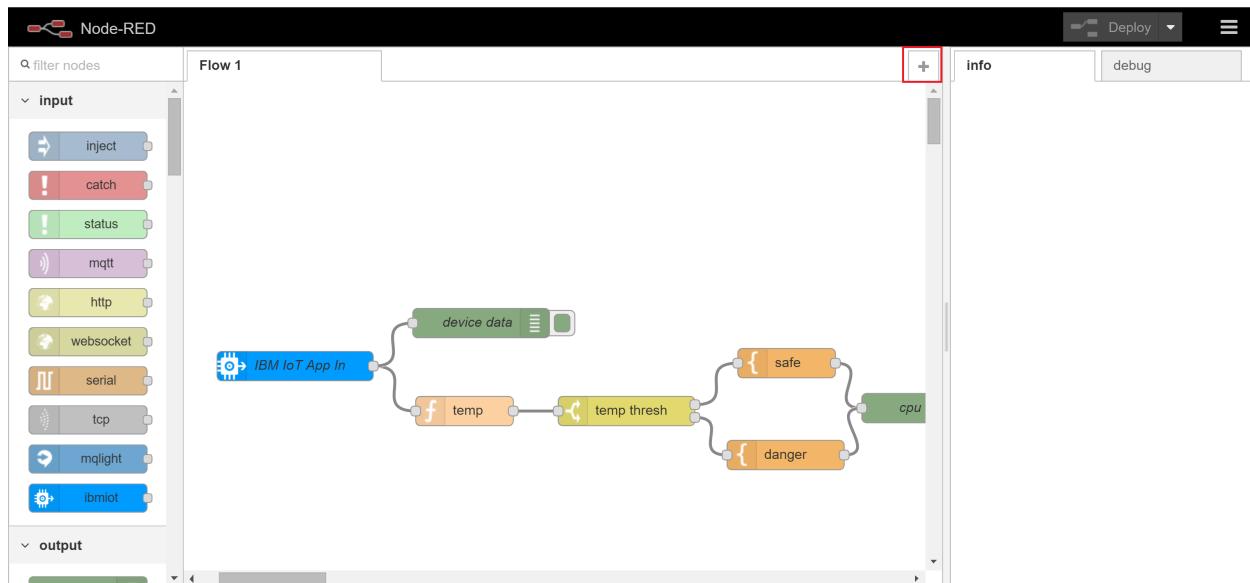
The screenshot shows the Node-RED interface in Bluemix. At the top, the URL is "raspberry-iot.mybluemix.net" and the title is "Node-RED in Bluemix". Below the title, it says "A visual tool for wiring the Internet of Things" and "IBM Internet of Things Foundation". The main area displays a visual flow editor with several nodes connected by wires. A purple button at the bottom right is highlighted with a red border, containing the text "Go to your Node-RED flow editor". Below this button are two links: "Learn how to password-protect your instance" and "Learn how to customise Node-RED".

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 password.

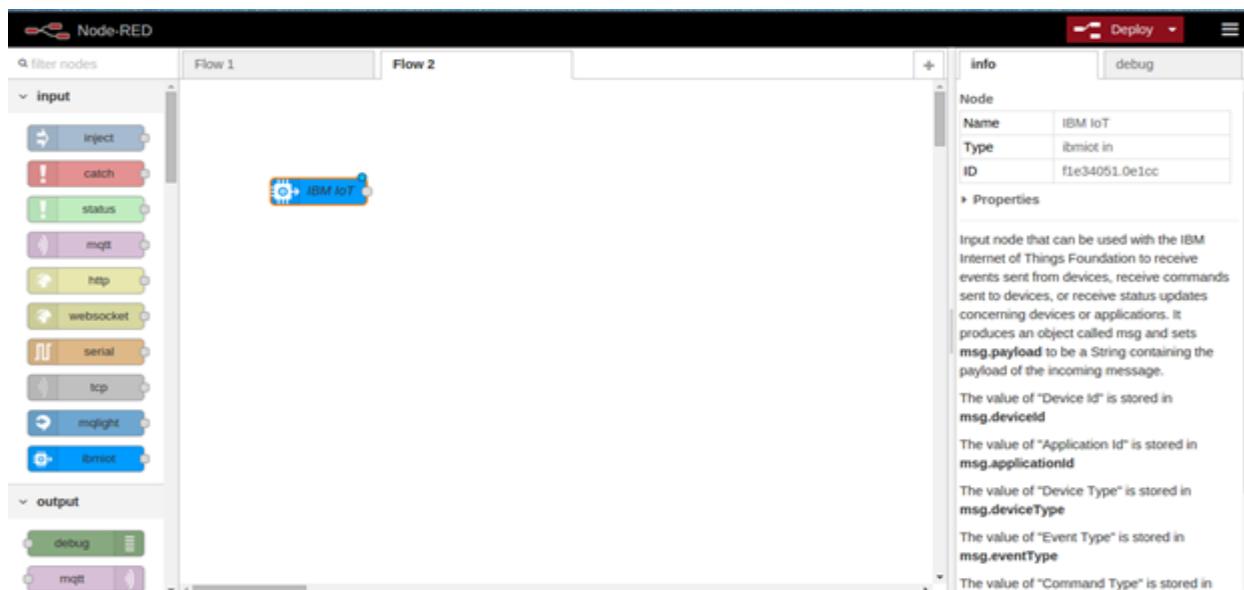
Node Red Flow sheet will be opened with a set of input and output nodes. The first sheet will be filled with a default flow given by Node Red. To get a new flow sheet, **Click on “+” icon** and rename it as you wish.



You will get a new flow sheet like this,



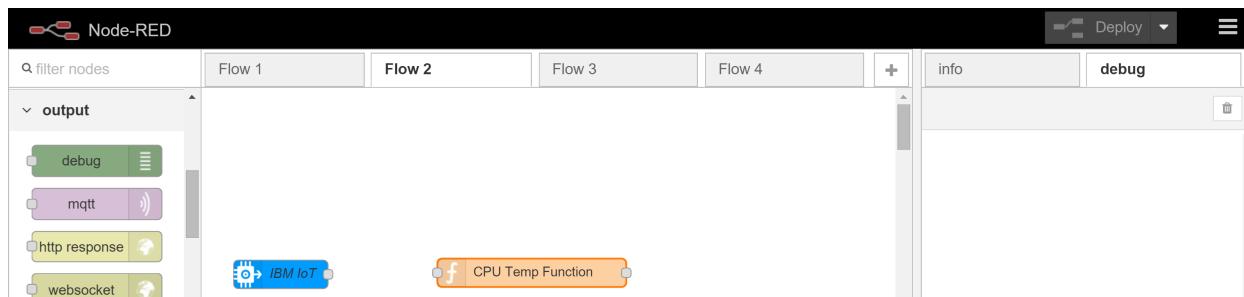
Drag and drop “ibmiot” node onto the flow sheet.



Double click on the “**ibmiot**” node. Give the following for prompted fields,

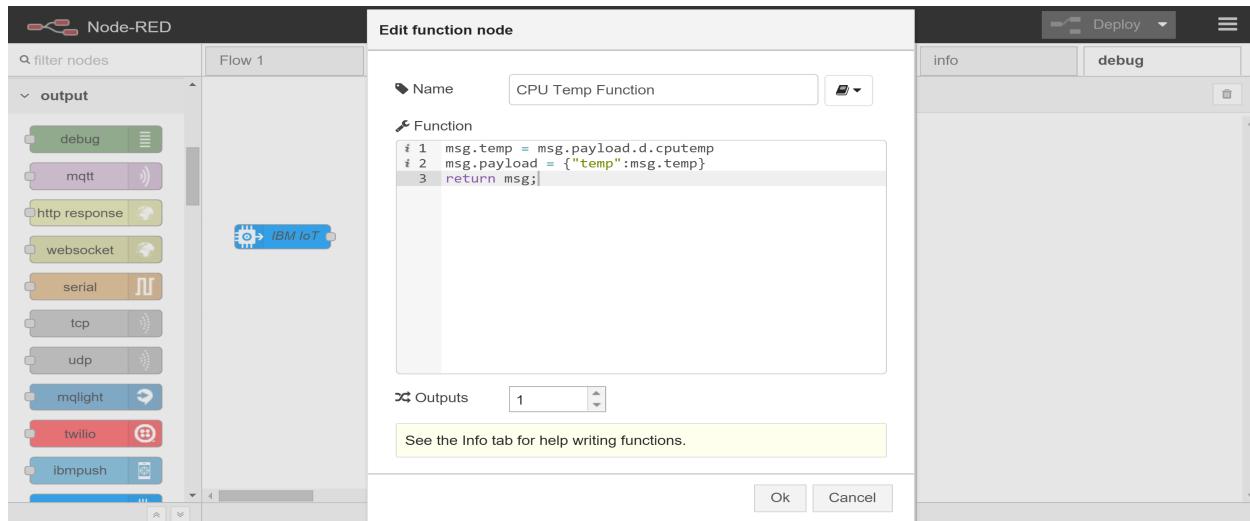
Field	Value
Authentication	Bluemix Service
Input Type	Device Event
Device Id	<Your-Device-ID(MAC ID)>
Format	Json

Leave the rest of the fields with their default values and click on enter to close the dialog. Drag and drop a function node from the list of function nodes in the left side menu.



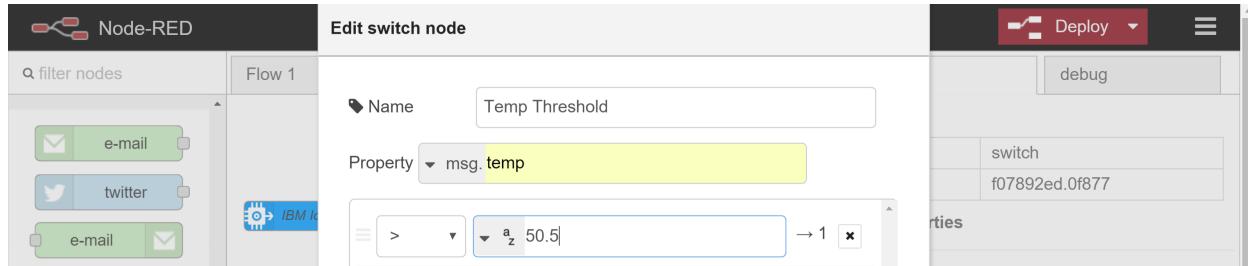
Give connection between IBM IoT node and function node. Double click on the function node. Give any name in the “name” field and Place the following in “Function” field,

```
msg.temp = msg.payload.d.cputemp
msg.name = msg.payload.d.myname
msg.payload = {"temp":msg.temp,"name":msg.name}
return msg;
```



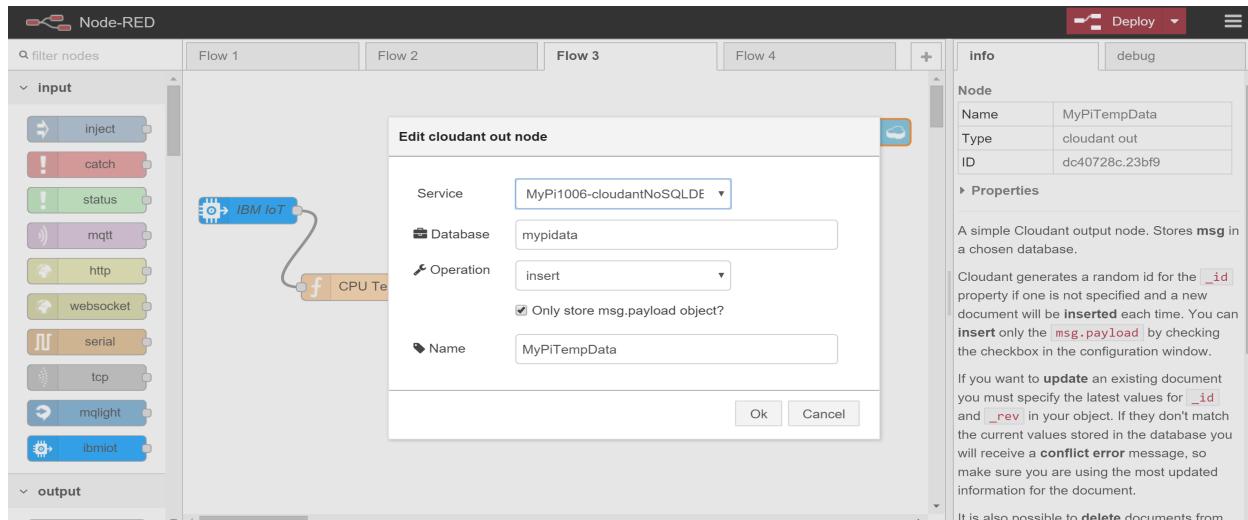
Drag and drop a switch node from the list of function nodes in the left side menu. Give connection between switch node and Function node.

To specify the conditions in the switch node, double click on it. You can give any number of rules for this node. Select left-side drop down box to select an operator and give temperature condition in the box next to it. You can add another rule by clicking on “+rule”. When you are done, click on “Ok”.

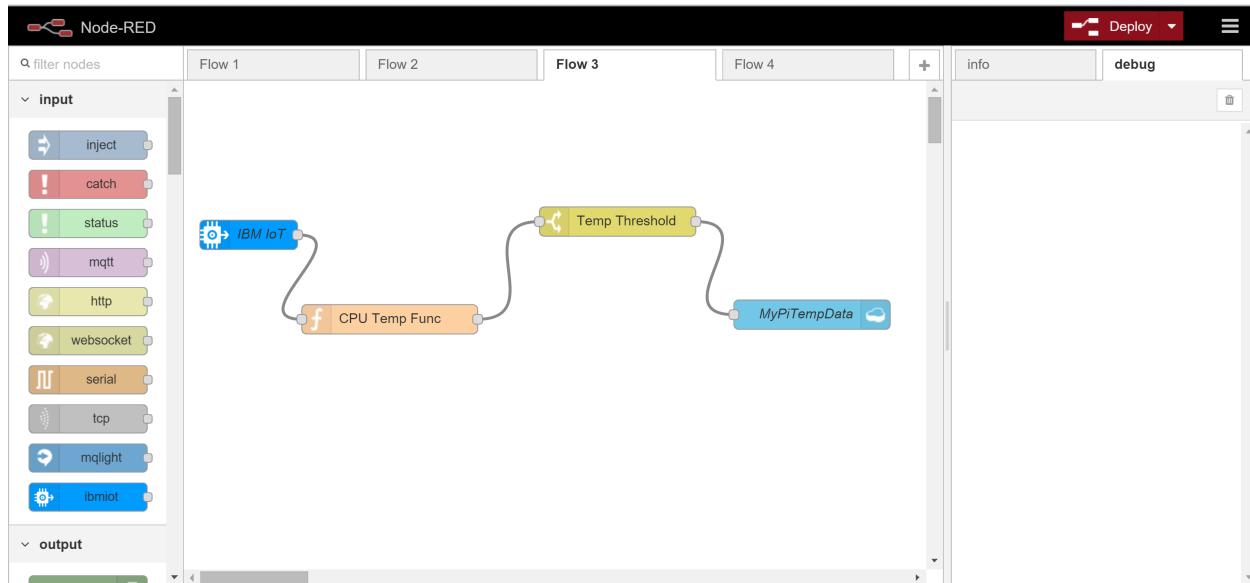


Drag the **Cloudant** node on to the Dashboard. And double click on the node to configure it. Add the Bluemix service in the Drop down as shown in below image.

Note : Here you can give any database name, if it is there or not it will create it for you.



Once the flow is successfully deployed, Go to your Twitter account to find the tweets, which is nothing but the CPU temperature coming from your IoT device (Raspberry Pi).



Click on the Cloudant Service from your Bluemix Application.

The screenshot shows the IBM Bluemix application dashboard for 'MyPi1006'. The left sidebar shows the app overview and services bound: 'Cloudant NoSQL DB' and 'Internet of Things Foundation'. The main panel displays the app configuration, showing 1 instance, 512 MB memory quota, and 1.000 GB available memory. The activity log on the right shows recent events:

- 2/18/16 12:17 AM jgunuru@miraclesoft.com started MyPi1006 app
- 2/18/16 12:17 AM jgunuru@miraclesoft.com updated MyPi1006 app
 - changed routes
- 2/18/16 12:17 AM jgunuru@miraclesoft.com created MyPi1006 app

Now click on the “Launch” button which is present at top right corner as shown in below image.

The screenshot shows the IBM Bluemix dashboard for the application 'MyPi1006'. The main title is 'MyPi1006 - CloudantNoSQLDB'. On the left sidebar, under 'SERVICES', 'Cloudant NoSQL DB' is selected. A red box highlights the 'LAUNCH' button in the top right corner of the main content area.

It will redirect you to the Cloudant DB Dashboard where we can see all the Databases and Data, which is persisted.

The screenshot shows the Cloudant DB Dashboard. The left sidebar has options like Databases, Replication, Warehousing, Active Tasks, Account, and Support. The main area shows a table of databases:

Name	Size	# of Docs	Update Seq	Actions
mypidata	5.2 KB	156	1	
nodered	26.3 KB	4	5	

Click on the mypi database to see the Temperature records, which are persisted in the form of JSON.

The screenshot shows the Cloudant DB Details view for the 'mypidata' database. The left sidebar has options like Databases, Replication, Warehousing, Active Tasks, Account, and Support. The main area shows a table of documents:

Permissions	Changes	All Documents	Query	All Design Docs

The 'All Documents' section shows two JSON documents:

```

id "0738d83d6092b6128e04463fc1816641"
{
  "_id": "0738d83d6092b6128e04463fc1816641",
  "_rev": "1-252101543a7d43077fcf970b1044f5ab",
  "value": {
    "rev": "1-252101543a7d43077fcf970b1044f5ab"
  },
  "key": "0738d83d6092b6128e04463fc1816641",
  "doc": {
    "id": "0738d83d6092b6128e04463fc1816641",
    "rev": "1-252101543a7d43077fcf970b1044f5ab",
    "temp": 51.92
  }
}

id "0738d83d6092b6128e04463fc18adbd9"
{
}

```