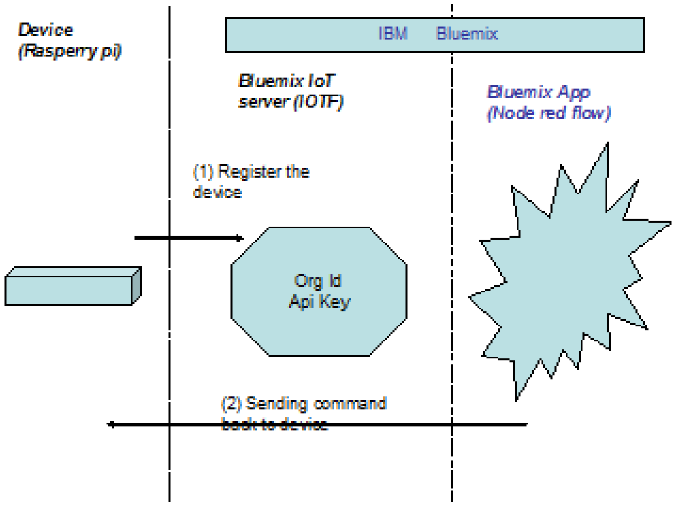
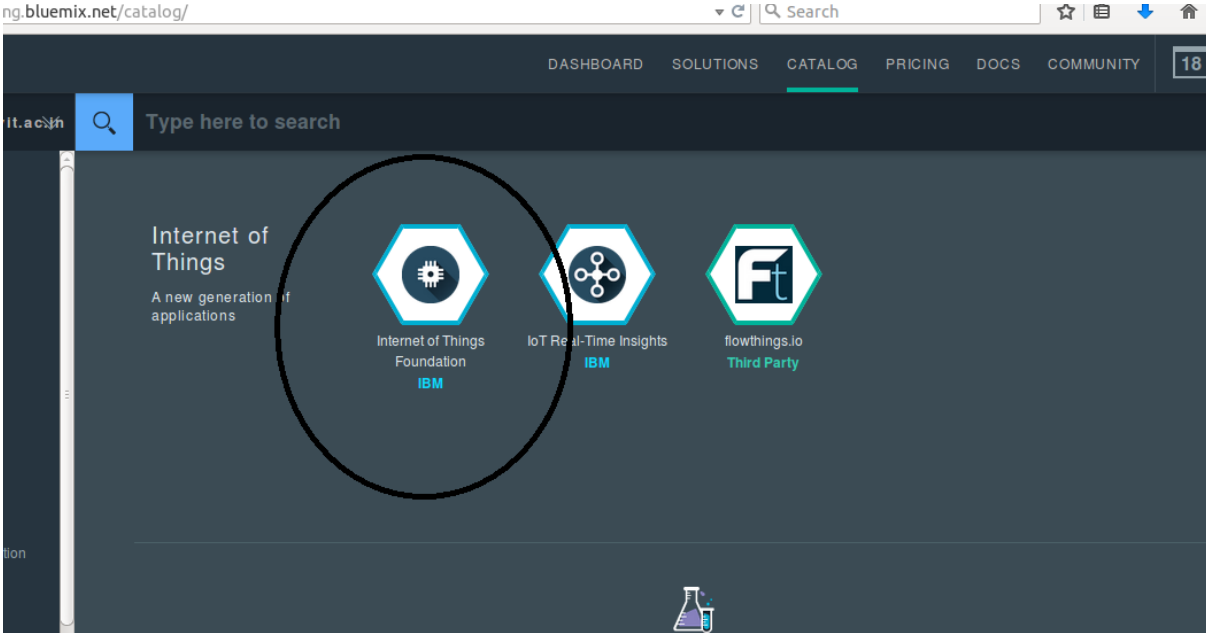
Lab 3; Register the device with Bluemix IoT server (IOTF) and send the command back to the device (Designed for Raspberry pi device/can be used for Intel)

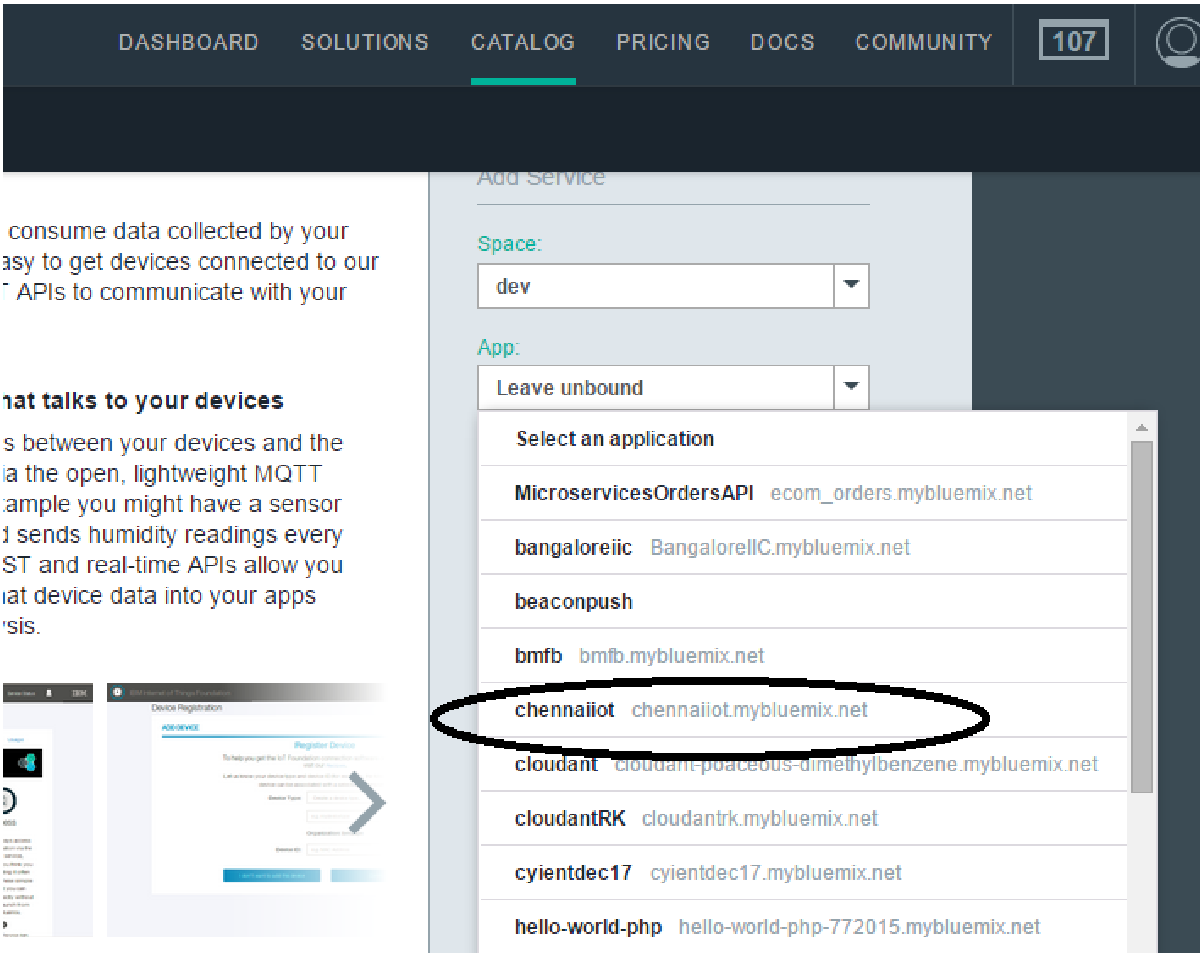


STEPS:

1. In catalog choose the “Internet of Things Foundation” service



2. Bind it to the application created in Lab1, in our case, Chennaiiot



consume data collected by your
asy to get devices connected to our
APIs to communicate with your

at talks to your devices

s between your devices and the
ia the open, lightweight MQTT
ample you might have a sensor
d sends humidity readings every
ST and real-time APIs allow you
at device data into your apps
sis.

Add Service

Space: dev

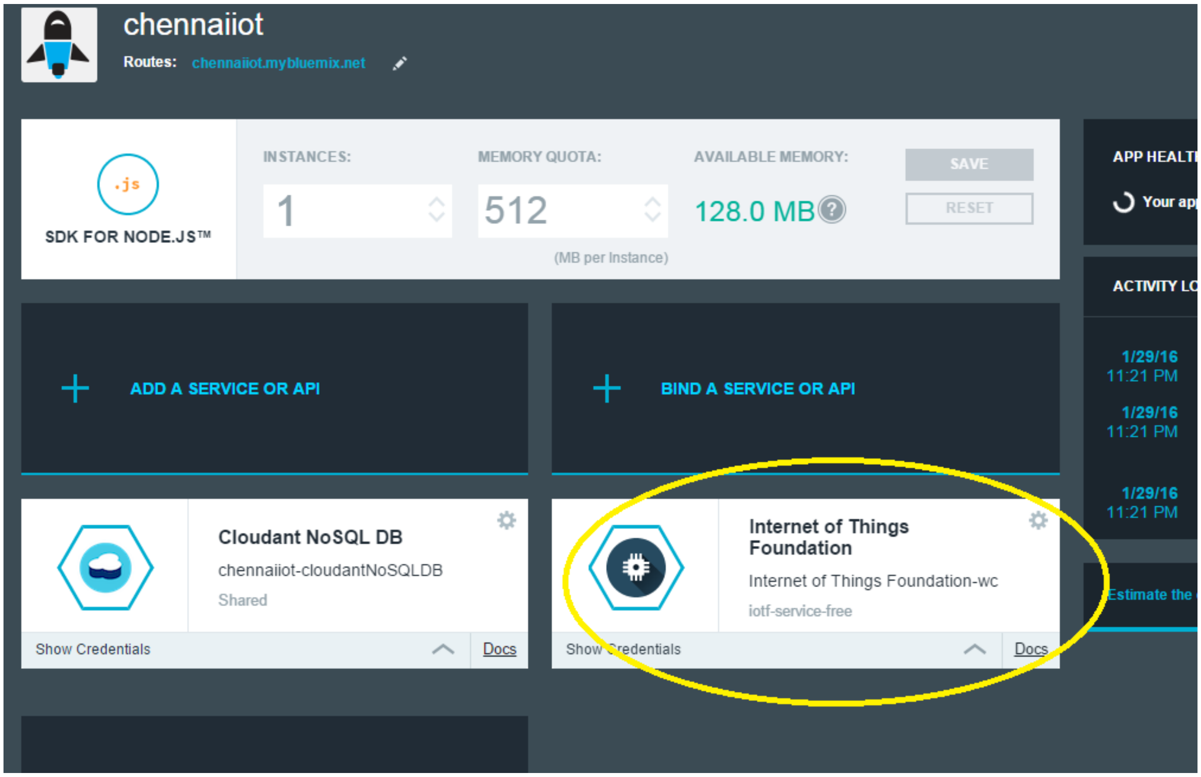
App: Leave unbound

Select an application

MicroservicesOrdersAPI	ecom_orders.mybluemix.net
bangaloreiic	BangaloreIIC.mybluemix.net
beaconpush	
bmfb	bmfb.mybluemix.net
chennaiiot	chennaiiot.mybluemix.net
cloudant	cloudant-poaceous-dimethylbenzene.mybluemix.net
cloudantRK	cloudantrk.mybluemix.net
cyientdec17	cyientdec17.mybluemix.net
hello-world-php	hello-world-php-772015.mybluemix.net

Note: At this stage, the application gets restaged

3. Launch the Internet of things dashboard



INSTANCES: 1

MEMORY QUOTA: 512 (MB per Instance)

AVAILABLE MEMORY: 128.0 MB

1/29/16 11:21 PM

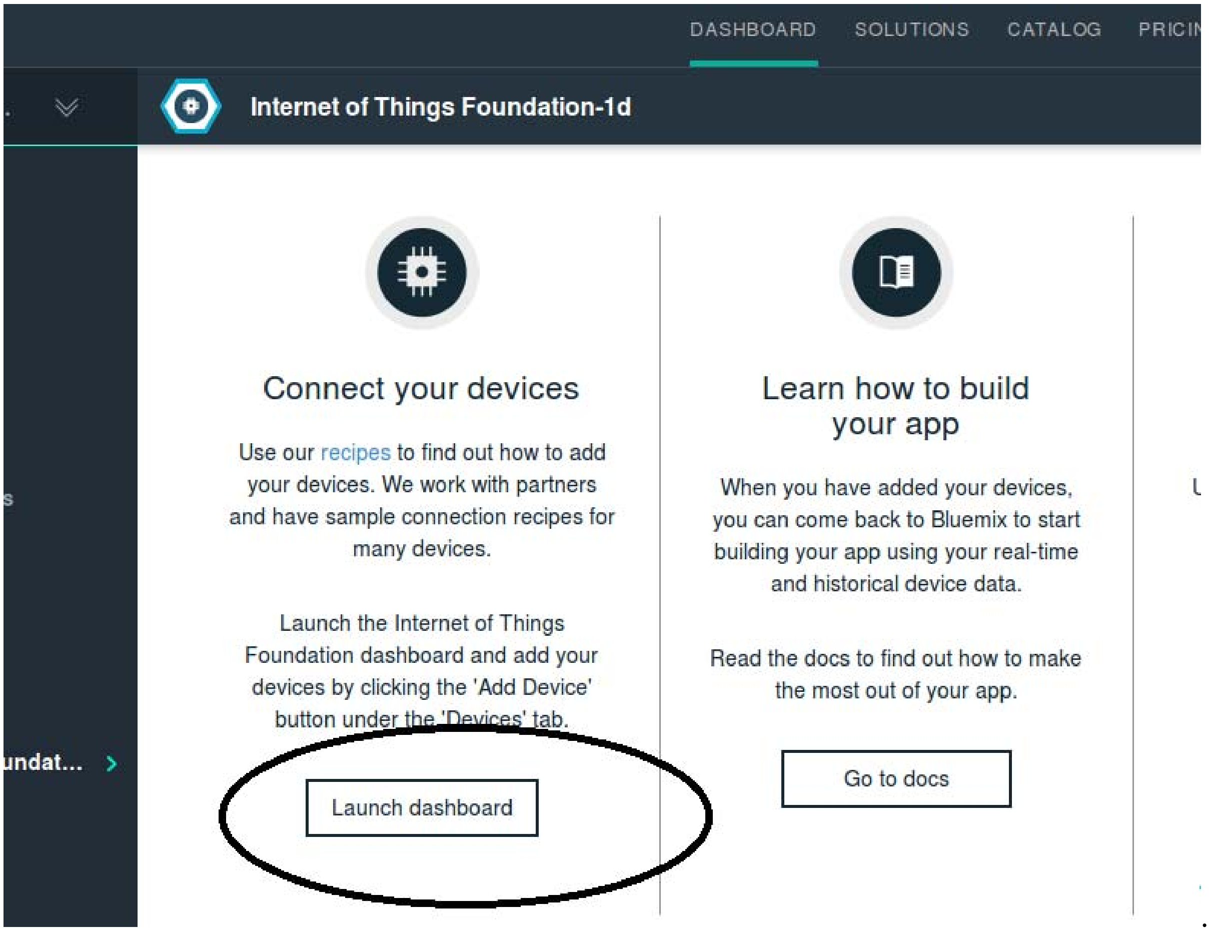
1/29/16 11:21 PM

1/29/16 11:21 PM

Cloudant NoSQL DB
chennaiiot-cloudantNoSQLDB
Shared

Internet of Things Foundation
Internet of Things Foundation-wc
iotf-service-free

4 . Launch dashboard



Internet of Things Foundation-1d

Connect your devices

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

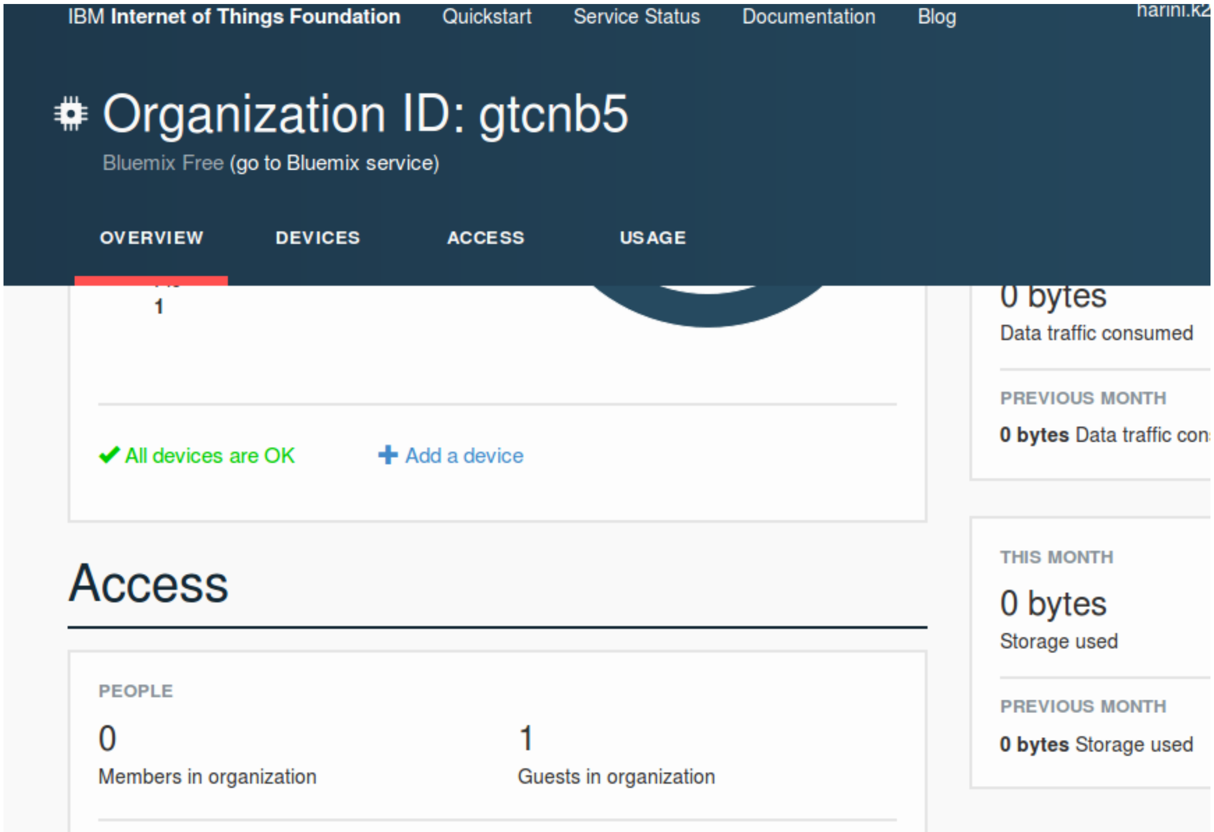
Launch the Internet of Things Foundation dashboard and add your devices by clicking the 'Add Device' button under the 'Devices' tab.

Learn how to build your app

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

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

5. Then Click on Add a device.



IBM **Internet of Things Foundation** Quickstart Service Status Documentation Blog harini.k2

Organization ID: gtcnb5

Bluemix Free (go to Bluemix service)

OVERVIEW DEVICES ACCESS USAGE

1

✓ All devices are OK

+ Add a device

0 bytes
Data traffic consumed

PREVIOUS MONTH
0 bytes Data traffic con

Access

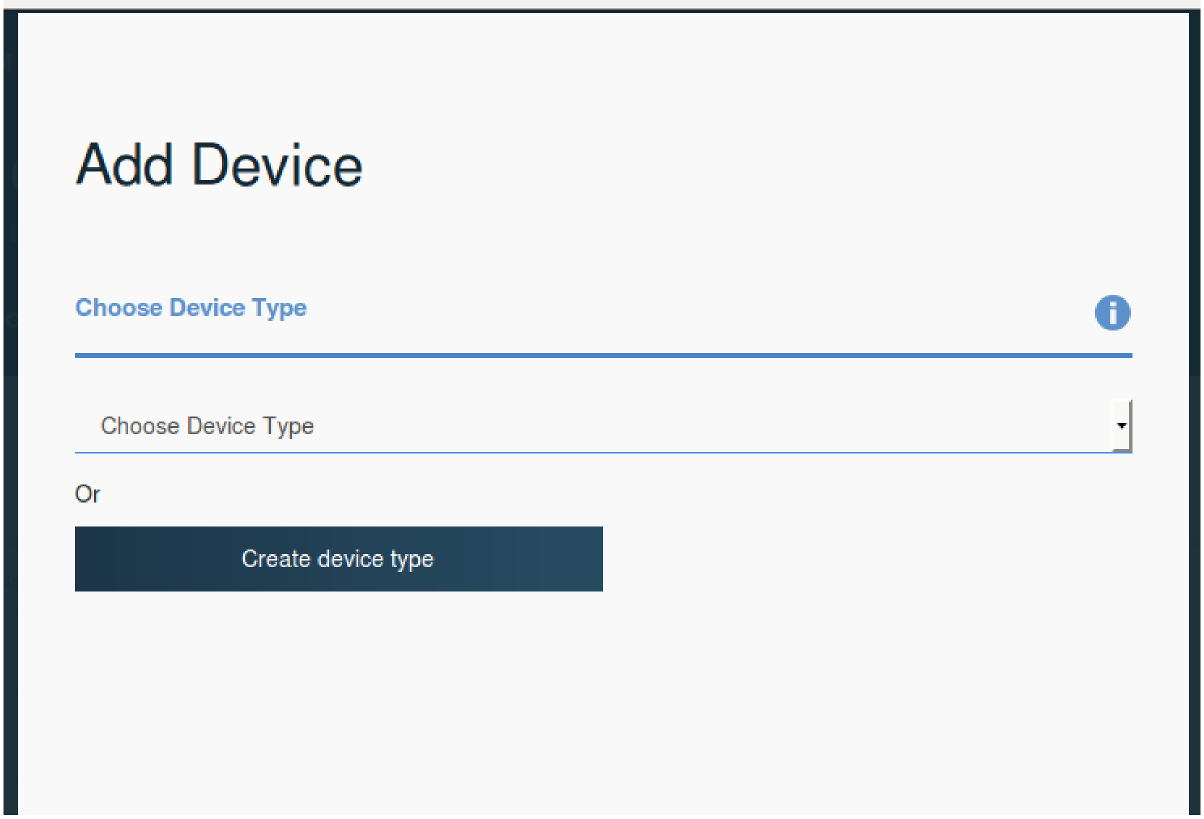
PEOPLE

0
Members in organization

1
Guests in organization

THIS MONTH
0 bytes
Storage used

PREVIOUS MONTH
0 bytes Storage used



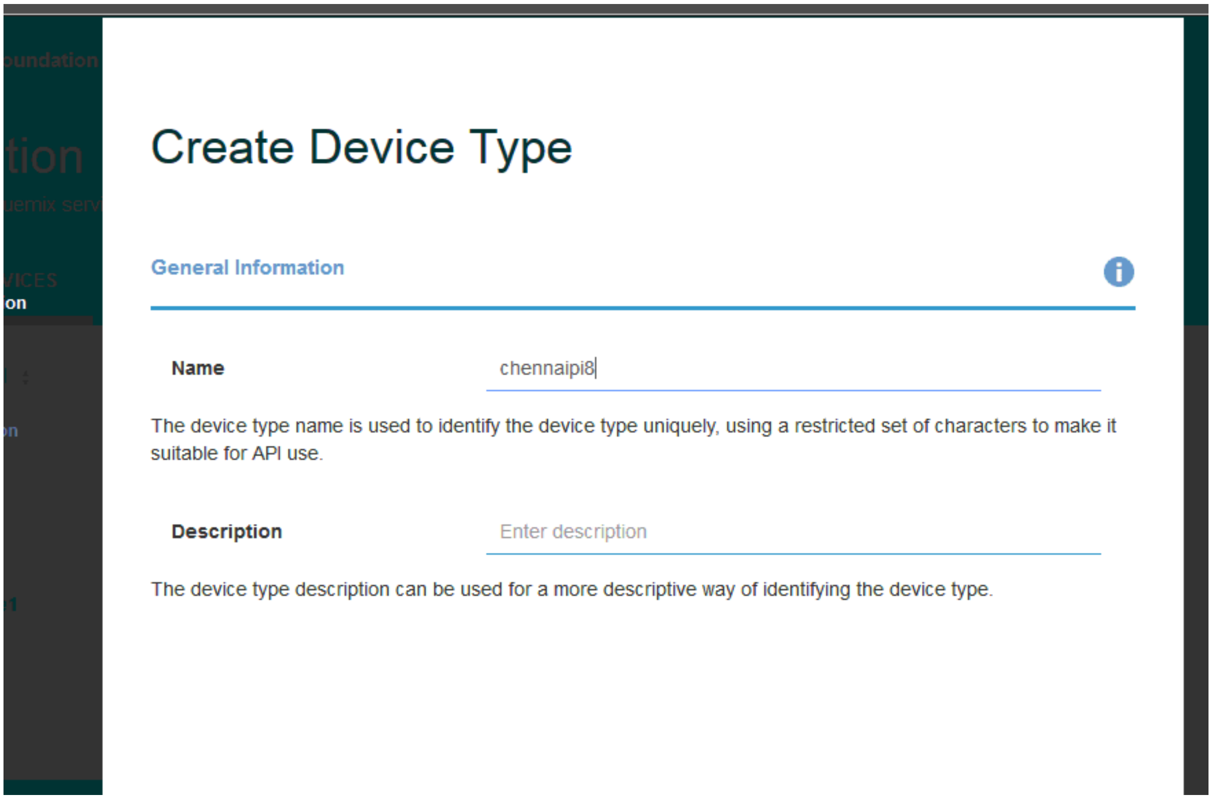
Add Device

Choose Device Type

Choose Device Type

Or

Create device type



Create Device Type

General Information

Name chennaipi8

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.

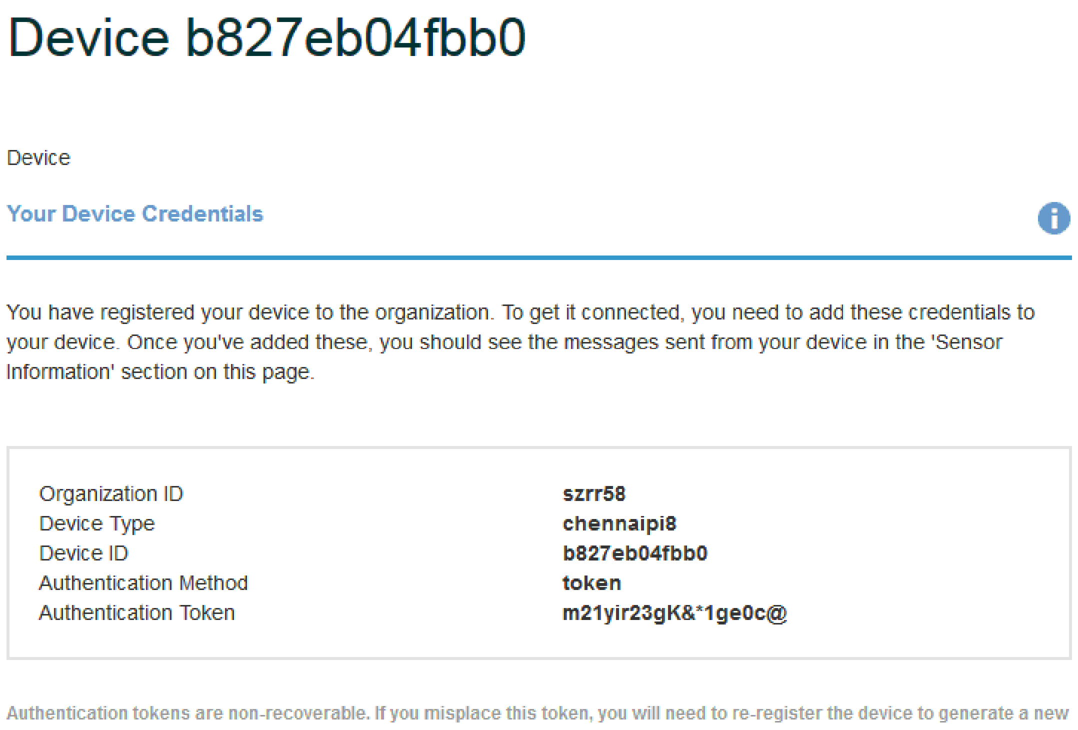
Description Enter description

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

Note: Name the device, in this case , chennaipi8. You can give any other name

6. Provide the Device Id as MAC address of the raspberry pi which you will be connecting (to be provided by the event admin)

7. Bluemix IoT Server , IOTF , returns access credentials as shown



Device b827eb04fbb0

Device

Your Device Credentials

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

Organization ID	**szrr58**
Device Type	**chennaipi8**
Device ID	**b827eb04fbb0**
Authentication Method	**token**
Authentication Token	**m21yir23gK&*1ge0c@**

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

Note: It is important to note down these credetials in a notepad

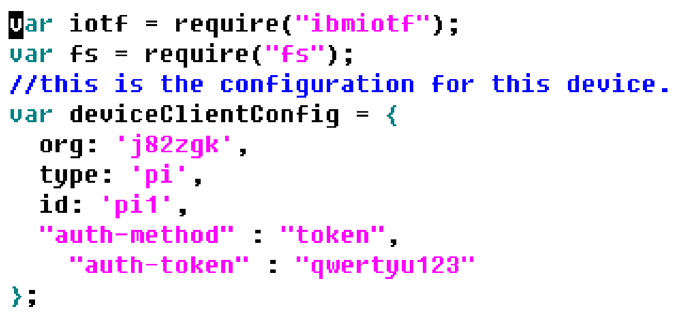
In our case, it is as below:

Organization ID - szrr58
Device Type - chennaipi8
Device ID - b827eb04fbb0
Authentication Method -token
Authentication Token -@(SW?o4+yNG6Y)1zx&

STEP B

Provide these credentials to the client application running in the raspberry pi device

1. Open the file main.js and provide the credentials what you have got

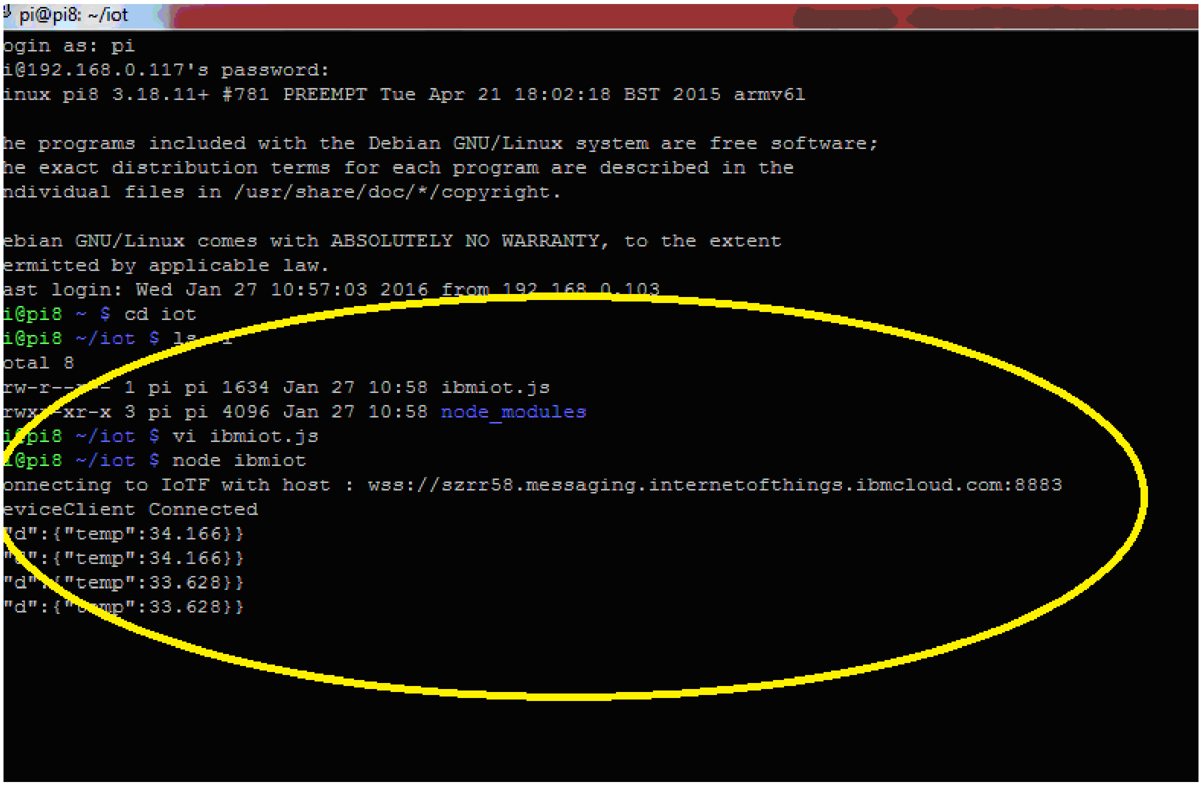


```
var iotf = require("ibmiotf");
var fs = require("fs");
//this is the configuration for this device.
var deviceClientConfig = {
  org: 'j82zgk',
  type: 'pi',
  id: 'pi1',
  "auth-method" : "token",
    "auth-token" : "qwertyu123"
};
```

2. Save this file and run the client program

Run it as “node ibmiot.js”

Wait for a while to see that it connects with the device and start publishing the (built in) sensor data , as shown below



pi@pi8: ~/iot

```
ogin as: pi
i@192.168.0.117's password:
inux pi8 3.18.11+ #781 PREEMPT Tue Apr 21 18:02:18 BST 2015 armv6l

he programs included with the Debian GNU/Linux system are free software;
he exact distribution terms for each program are described in the
ndividual files in /usr/share/doc/*/copyright.

ebian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
ermitted by applicable law.
ast login: Wed Jan 27 10:57:03 2016 from 192.168.0.103
i@pi8 ~ $ cd iot
i@pi8 ~/iot $ ls -l
otal 8
rw-r--r-- 1 pi pi 1634 Jan 27 10:58 ibmiot.js
rwxr-xr-x 3 pi pi 4096 Jan 27 10:58 node_modules
i@pi8 ~/iot $ vi ibmiot.js
i@pi8 ~/iot $ node ibmiot
onnecting to IoTF with host : wss://szrr58.messaging.internetofthings.ibmcloud.com:8883
eviceClient Connected
"d":{"temp":34.166}}
"d":{"temp":34.166}}
"d":{"temp":33.628}}
"d":{"temp":33.628}}
```

STEP C

Go back to the Bluemix node red flow editor and send the command back to the device.

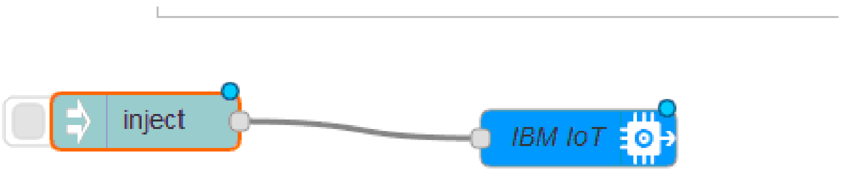
In our case, we send the command to ‘createfile” and update the file with a string content

1. Go back to bluemix console and run the boilerplate app created as in lab 1. In our case http://chennaiiot.mybluemix.net/

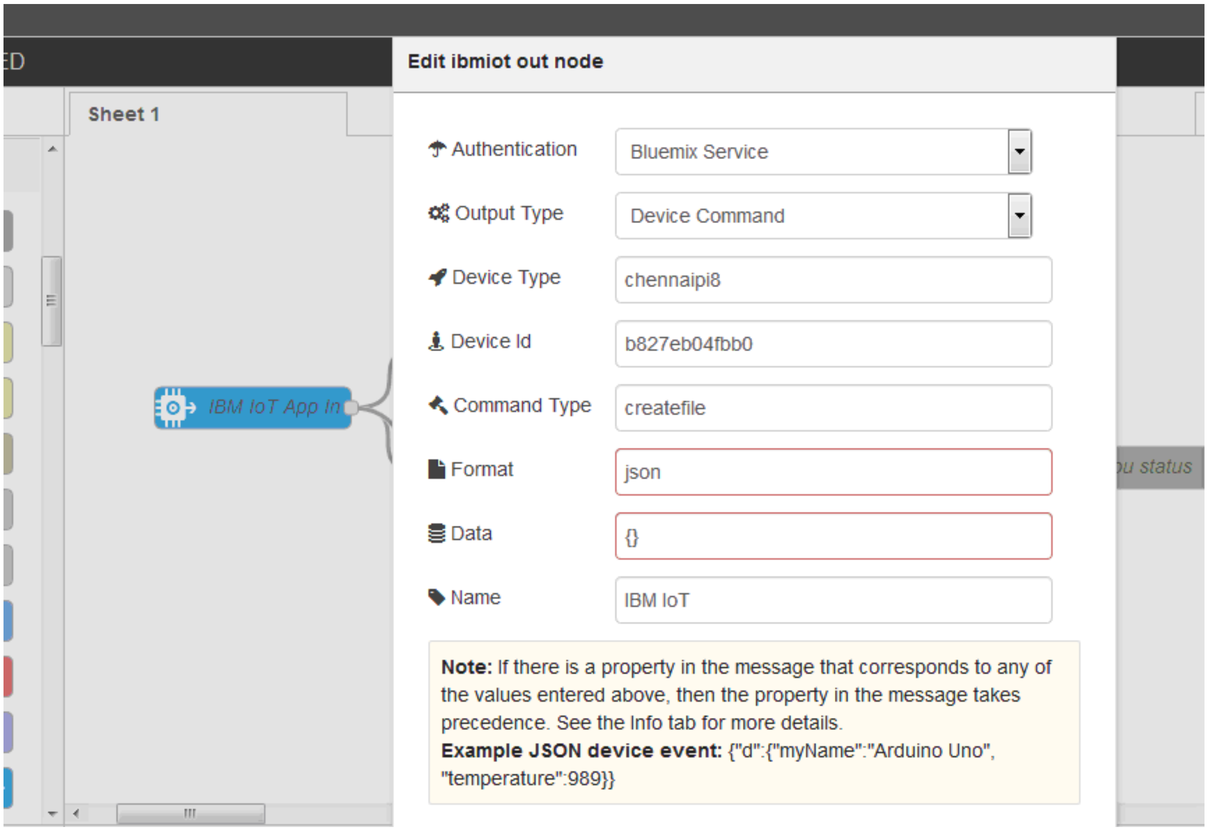
2.open the node red editor

4. Add 2 new nodes to inject commands back to the device

Node 1 : Inject node from **input**
Node 2 : ibm iot out node



5. provide details in the IBM IOT outnode as shown in the image below



Edit ibmiot out node

Authentication: Bluemix Service

Output Type: Device Command

Device Type: chennaipi8

Device Id: b827eb04fbb0

Command Type: createfile

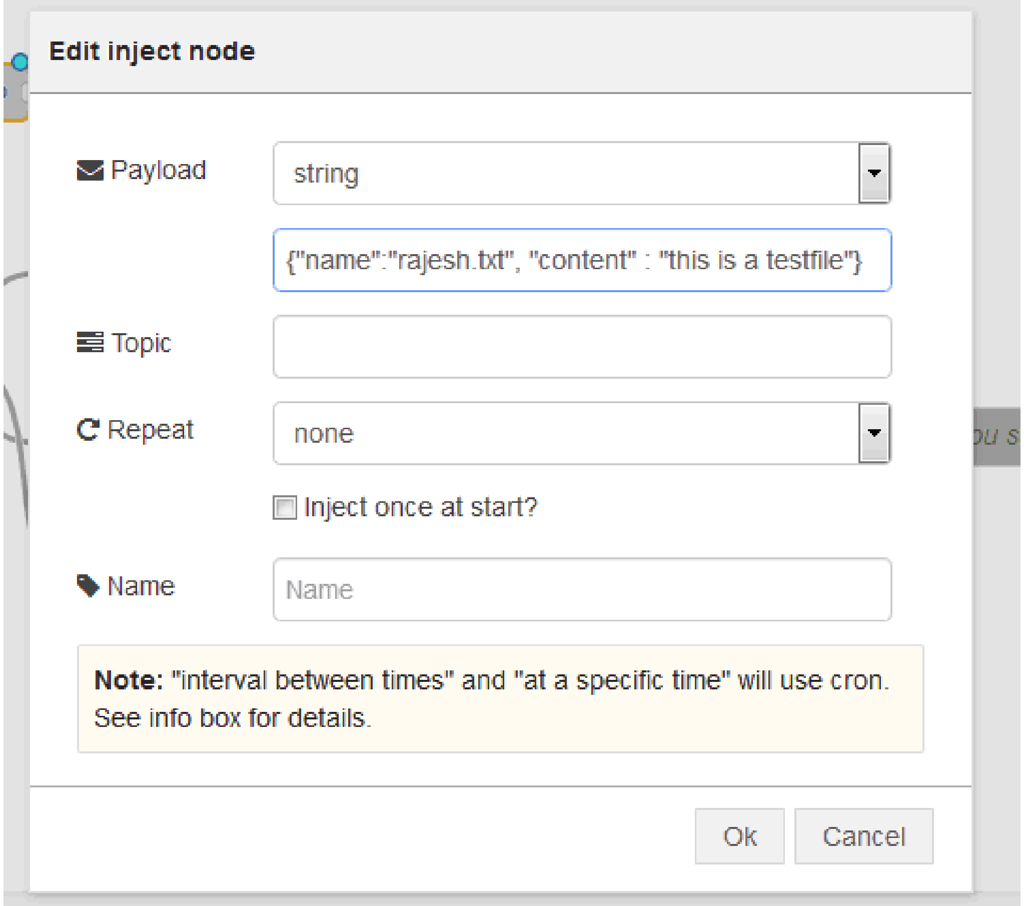
Format: json

Data: {}

Name: IBM IoT

Note: If there is a property in the message that corresponds to any of the values entered above, then the property in the message takes precedence. See the Info tab for more details.
Example JSON device event: {"d":{"myName":"Arduino Uno", "temperature":989}}

6. In the Inject node, send the command to create file



Edit inject node

Payload string

{"name":"rajesh.txt", "content" : "this is a testfile"}

Topic

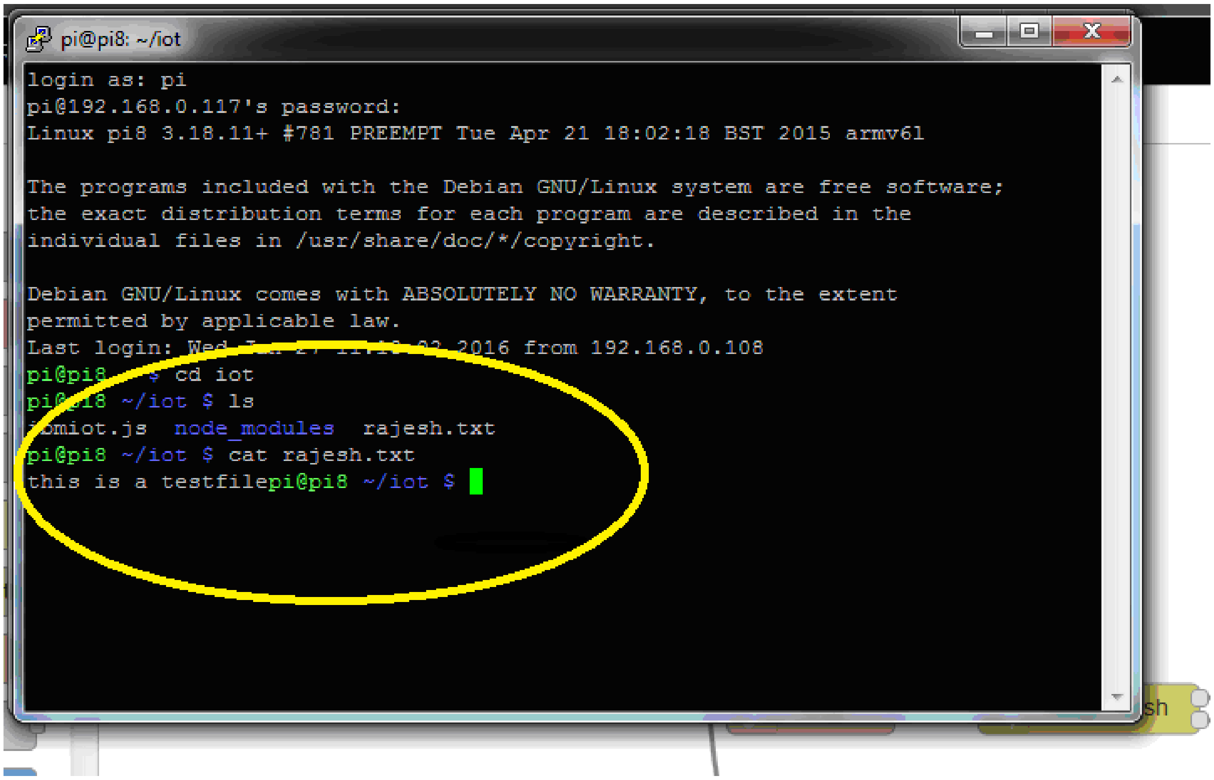
Repeat none

□ Inject once at start?

Name Name

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

That’s it….now deploy the node and see the file is been created under the fole /iot…



```
login as: pi
pi@192.168.0.117's password:
Linux pi8 3.18.11+ #781 PREEMPT Tue Apr 21 18:02:18 BST 2015 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jan 27 11:10:02 2016 from 192.168.0.108
pi@pi8 ~ $ cd iot
pi@pi8 ~/iot $ ls
ibmiot.js  node_modules  rajesh.txt
pi@pi8 ~/iot $ cat rajesh.txt
this is a testfilepi@pi8 ~/iot $
```