Following are the tasks which we perform in our lab

Task 1: Create Node-Red Application

Task 2: Create Internet of Things Platform service and create a Device.

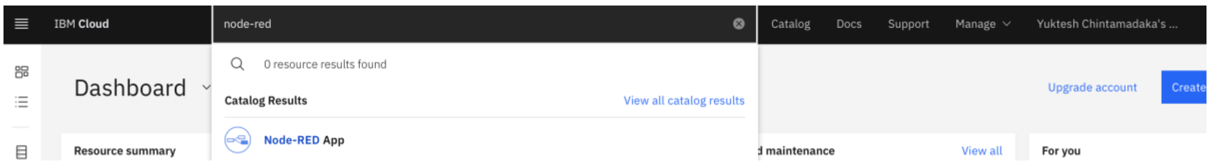
Task 3: Connect device to Watson-IoT sensor simulator

Task 4: Build Node-Red Application

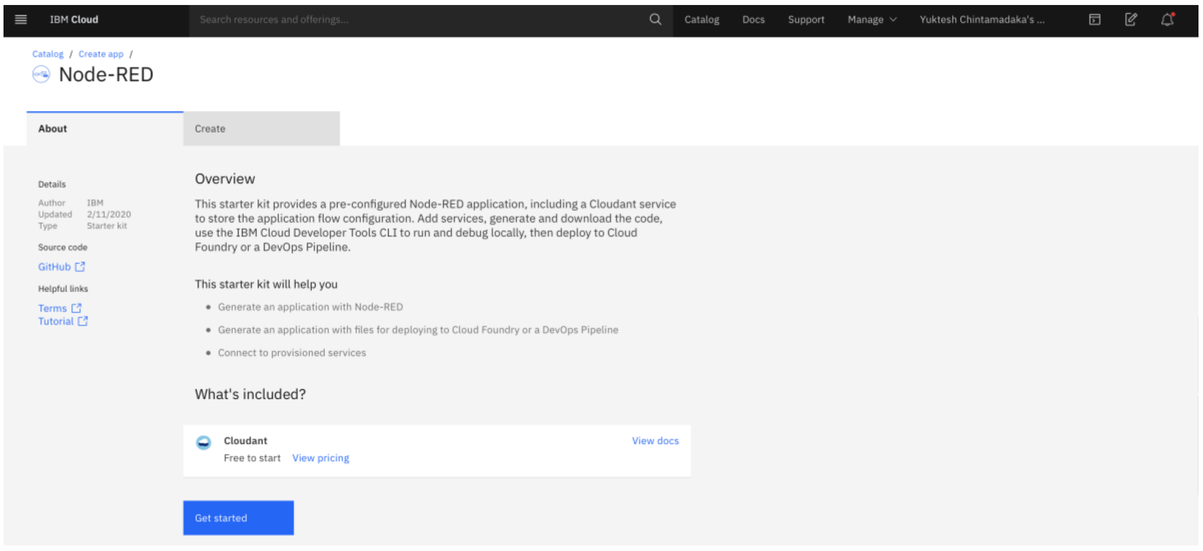
(i) Connect to device and fetch device data into Node-Red flow editor
(ii) Connect to database from Node-Red flow editor
(iii) Connect to a mobile from Node-Red flow editor

Task 1: Create Node-Red application:

Step 1: Type ***node-red*** in search you will see Catalog Results ***Node-RED App*** click on it



Step 2: Click on ***Get started***



Overview

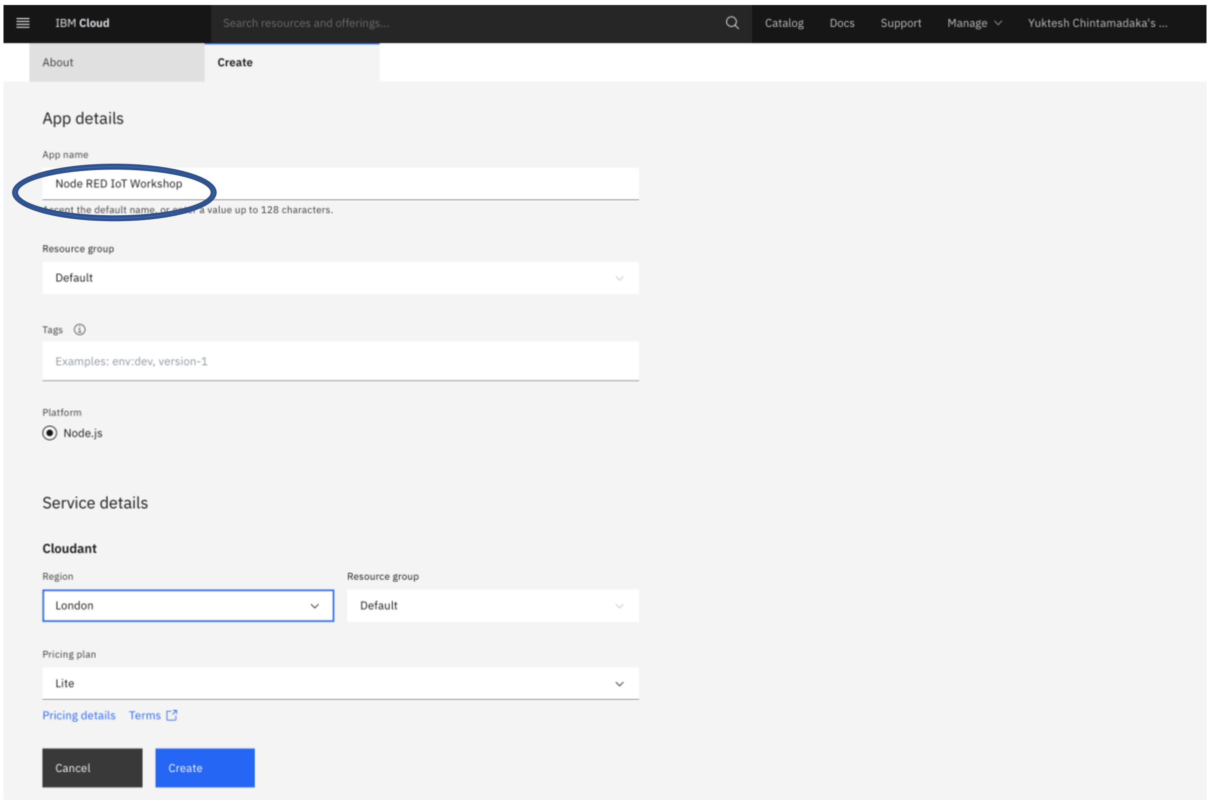
This starter kit provides a pre-configured Node-RED application, including a Cloudant service to store the application flow configuration. Add services, generate and download the code, use the IBM Cloud Developer Tools CLI to run and debug locally, then deploy to Cloud Foundry or a DevOps Pipeline.

This starter kit will help you

- Generate an application with Node-RED
- Generate an application with files for deploying to Cloud Foundry or a DevOps Pipeline
- Connect to provisioned services

What's included?

Step 3: If you want you can change app name or leave it default and click on ***Create***



App details

App name: Node RED IoT Workshop

Accept the default name, or enter a value up to 128 characters.

Resource group: Default

Tags: Examples: env:dev, version-1

Platform: Node.js

Service details

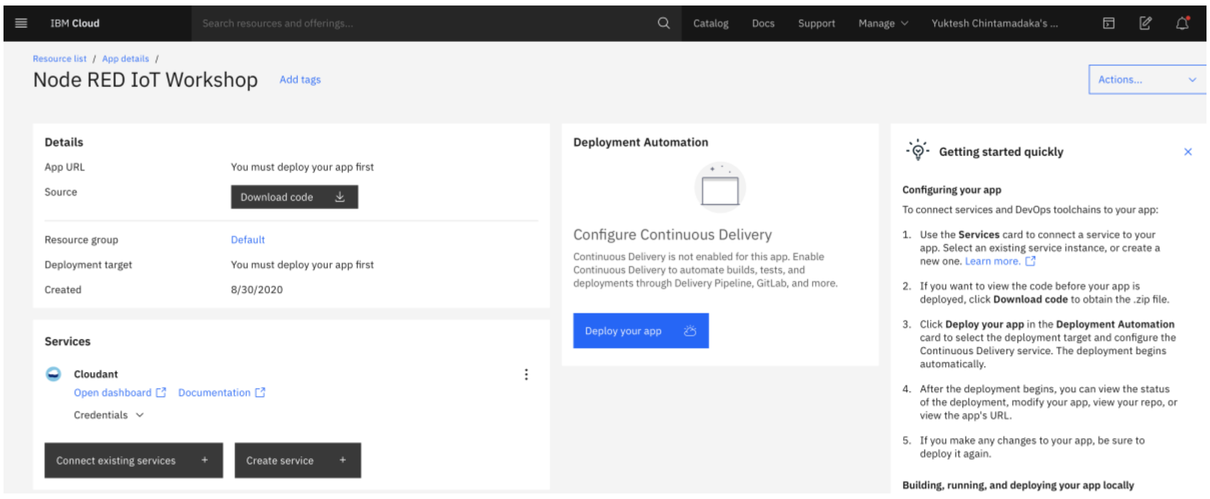
Cloudant

Region: London

Resource group: Default

Pricing plan: Lite

Step 4: click on ***Deploy your*** app to Configure Continuous Delivery


IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
Yuktesh Chintamadaka's ...
Resource list / App details /
Add tags
Actions...

Node RED IoT Workshop

Details

App URL	You must deploy your app first
Source	Download code
Resource group	Default
Deployment target	You must deploy your app first
Created	8/30/2020

Services

Cloudant

Open dashboard
Documentation
Credentials
Connect existing services +
Create service +

Deployment Automation

Configure Continuous Delivery

Continuous Delivery is not enabled for this app. Enable Continuous Delivery to automate builds, tests, and deployments through Delivery Pipeline, GitLab, and more.

Getting started quickly

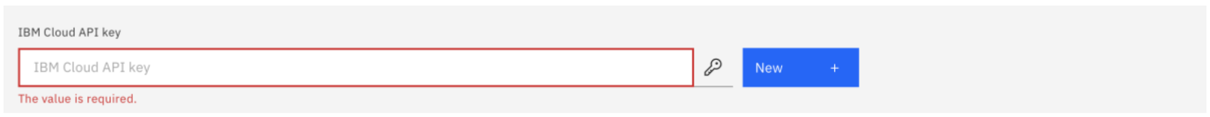
Configuring your app

To connect services and DevOps toolchains to your app:

1. Use the **Services** card to connect a service to your app. Select an existing service instance, or create a new one. Learn more.
2. If you want to view the code before your app is deployed, click **Download code** to obtain the .zip file.
3. Click **Deploy your app** in the **Deployment Automation** card to select the deployment target and configure the Continuous Delivery service. The deployment begins automatically.
4. After the deployment begins, you can view the status of the deployment, modify your app, view your repo, or view the app's URL.
5. If you make any changes to your app, be sure to deploy it again.

Building, running, and deploying your app locally

Step 5: click on ***New***

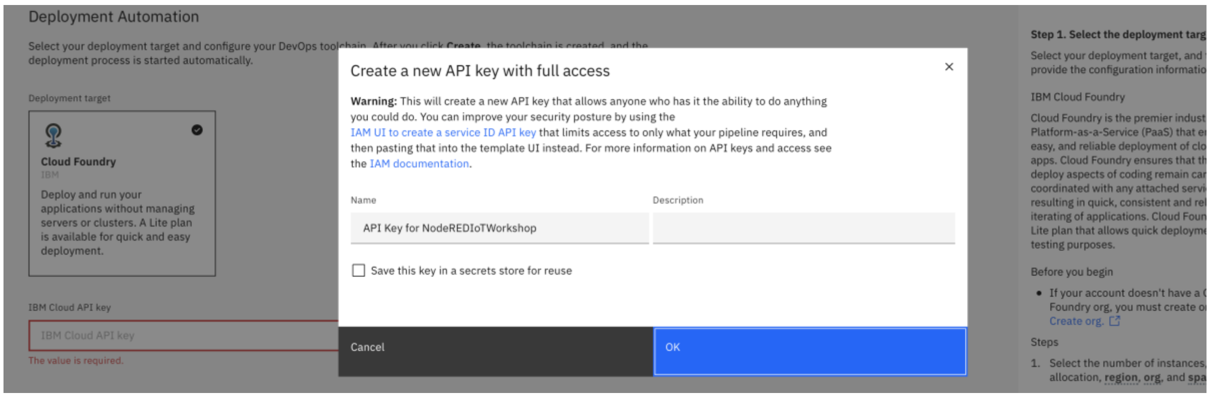


IBM Cloud API key

IBM Cloud API key

The value is required.

Step 6: click ***Ok***


Deployment Automation

Select your deployment target and configure your DevOps toolchain. After you click **Create**, the toolchain is created, and the deployment process is started automatically.

Deployment target

Cloud Foundry
IBM
Deploy and run your applications without managing servers or clusters. A Lite plan is available for quick and easy deployment.

IBM Cloud API key
IBM Cloud API key
The value is required.

Create a new API key with full access

Warning: This will create a new API key that allows anyone who has it the ability to do anything you could do. You can improve your security posture by using the IAM UI to create a service ID API key that limits access to only what your pipeline requires, and then pasting that into the template UI instead. For more information on API keys and access see the IAM documentation.

Name	Description
API Key for NodeREDIoTWorkshop	

☐ Save this key in a secrets store for reuse

Step 1. Select the deployment targ
Select your deployment target, and
provide the configuration informatio

IBM Cloud Foundry

Cloud Foundry is the premier indust
Platform-as-a-Service (PaaS) that e
easy, and reliable deployment of clo
apps. Cloud Foundry ensures that th
deploy aspects of coding remain car
coordinated with any attached servi
resulting in quick, consistent and rel
iterating of applications. Cloud Foun
Lite plan that allows quick deployme
testing purposes.

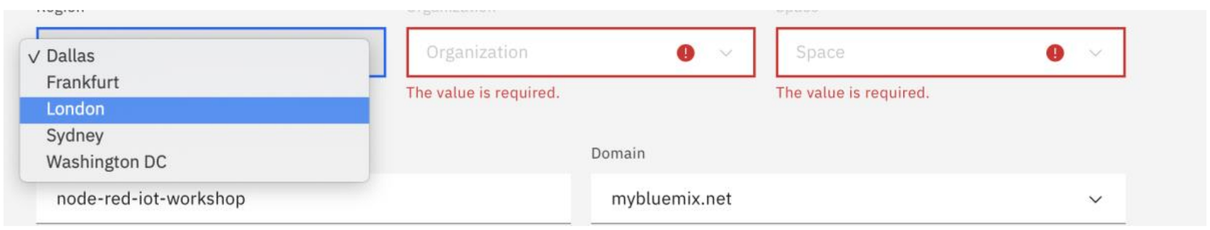
Before you begin

- If your account doesn't have a C
Foundry org, you must create o
Create org.

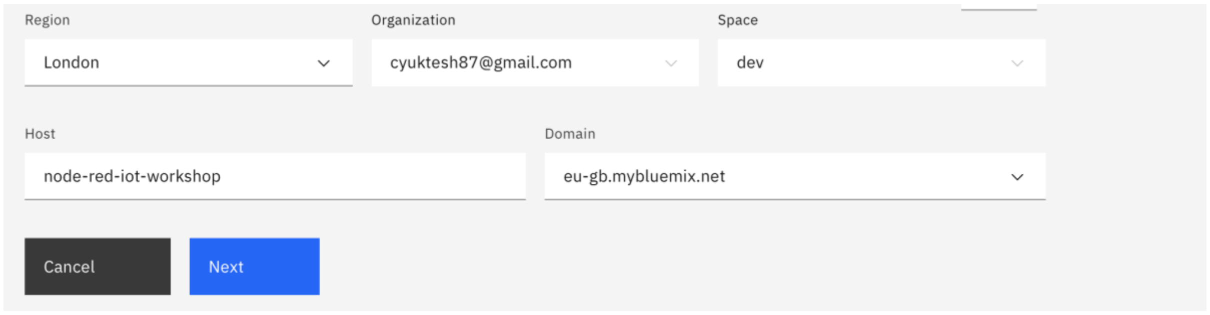
Steps

1. Select the number of instances,
allocation, region, org, and spa

Step 7: select ***London***



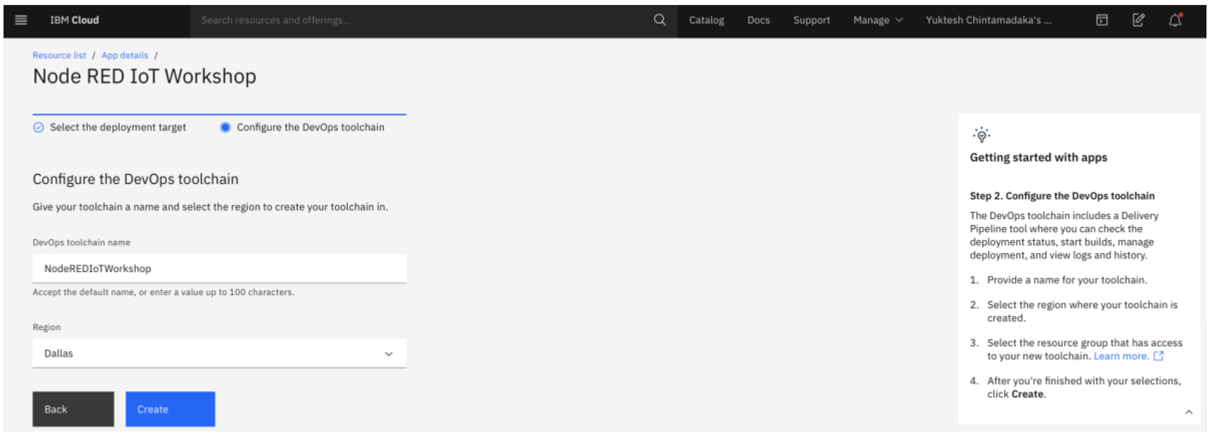
Step 8: click on ***Next***



Region	Organization	Space
London	cyuktesh87@gmail.com	dev

Host	Domain
node-red-iot-workshop	eu-gb.mybluemix.net

Step 9: click on ***Create***



IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
Yuktesh Chintamadaka's ...
Resource list / App details /

Node RED IoT Workshop

Select the deployment target · Configure the DevOps toolchain

Configure the DevOps toolchain

Give your toolchain a name and select the region to create your toolchain in.

DevOps toolchain name
NodeREDIoTWorkshop
Accept the default name, or enter a value up to 100 characters.

Region
Dallas

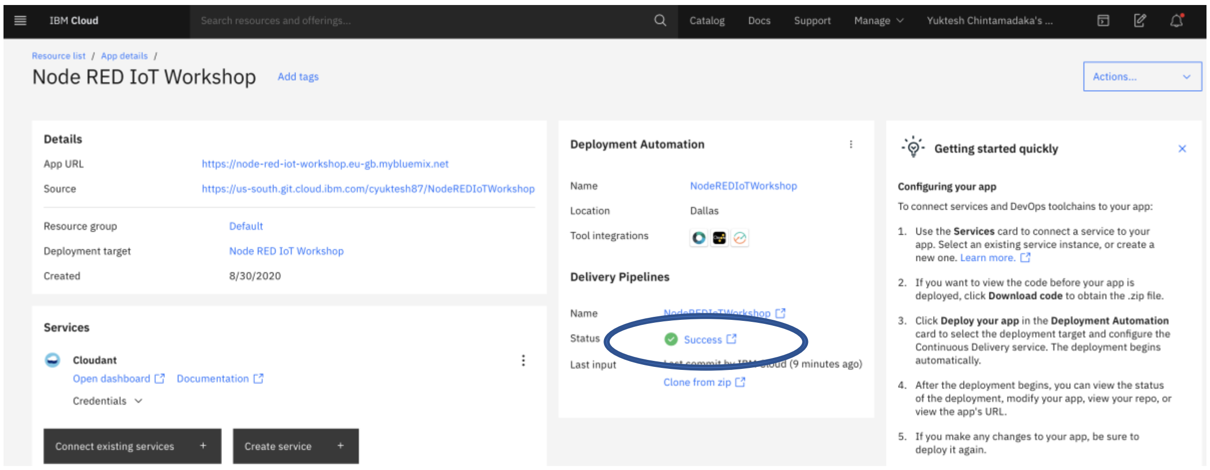
Getting started with apps

Step 2. Configure the DevOps toolchain

The DevOps toolchain includes a Delivery Pipeline tool where you can check the deployment status, start builds, manage deployment, and view logs and history.

1. Provide a name for your toolchain.
2. Select the region where your toolchain is created.
3. Select the resource group that has access to your new toolchain. Learn more.
4. After you're finished with your selections, click **Create**.

Step 10: Refresh the page so that and watch status ***Progress*** to status ***Success***


IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
Yuktesh Chintamadaka's ...
Resource list / App details /
Add tags
Actions...
Connect existing services +
Create service +

Node RED IoT Workshop

Details

App URL	https://node-red-iot-workshop.eu-gb.mybluemix.net
Source	https://us-south.git.cloud.ibm.com/cyuktesh87/NodeREDIoTWorkshop
Resource group	Default
Deployment target	Node RED IoT Workshop
Created	8/30/2020

Services

Cloudant
Open dashboard   Documentation
Credentials

Deployment Automation

Name	NodeREDIoTWorkshop
Location	Dallas
Tool integrations	

Delivery Pipelines

Name	NodeREDIoTWorkshop
Status	Success
Last input	Last commit by IBM Cloud (9 minutes ago)
	Clone from zip

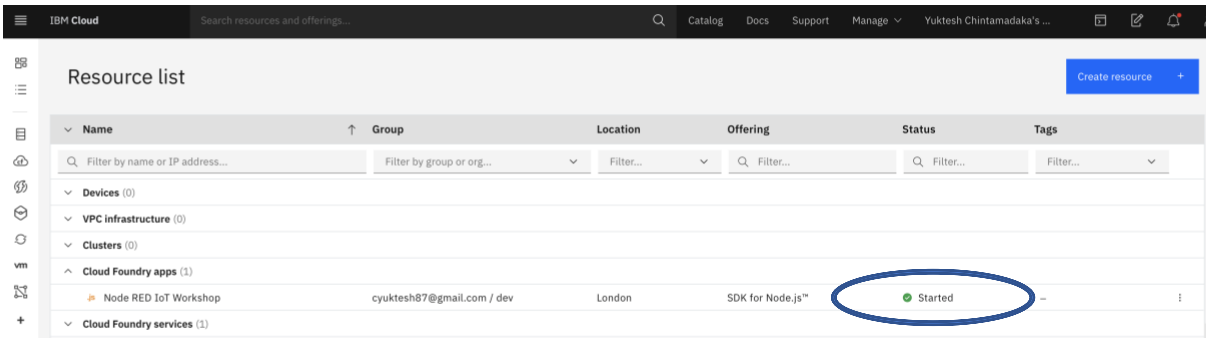
Getting started quickly

Configuring your app

To connect services and DevOps toolchains to your app:

1. Use the **Services** card to connect a service to your app. Select an existing service instance, or create a new one. Learn more.
2. If you want to view the code before your app is deployed, click **Download code** to obtain the .zip file.
3. Click **Deploy your app** in the **Deployment Automation** card to select the deployment target and configure the Continuous Delivery service. The deployment begins automatically.
4. After the deployment begins, you can view the status of the deployment, modify your app, view your repo, or view the app's URL.
5. If you make any changes to your app, be sure to deploy it again.

Step 11: Click on ***IBM Cloud*** → ***Cloud Foundry Apps*** → Ensure of ***Node RED IoT Workshop*** is started like below

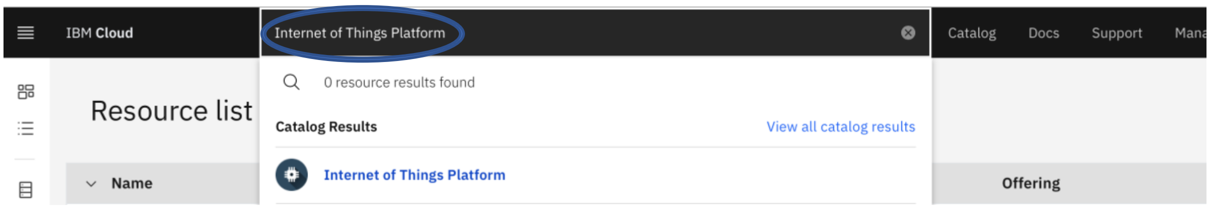

IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
Yuktesh Chintamadaka's ...
Create resource +
Filter by name or IP address...
Filter by group or org...
Filter...
Filter...
Filter...
Filter...

Resource list

Name	Group	Location	Offering	Status	Tags
Devices (0)					
VPC infrastructure (0)					
Clusters (0)					
Cloud Foundry apps (1)					
Node RED IoT Workshop	cyuktesh87@gmail.com / dev	London	SDK for Node.js™	Started	–
Cloud Foundry services (1)					

Task 2: Create Internet of Things Platform service

Step 1: search for ***Internet of Things Platform*** and click that service from Catalog Results

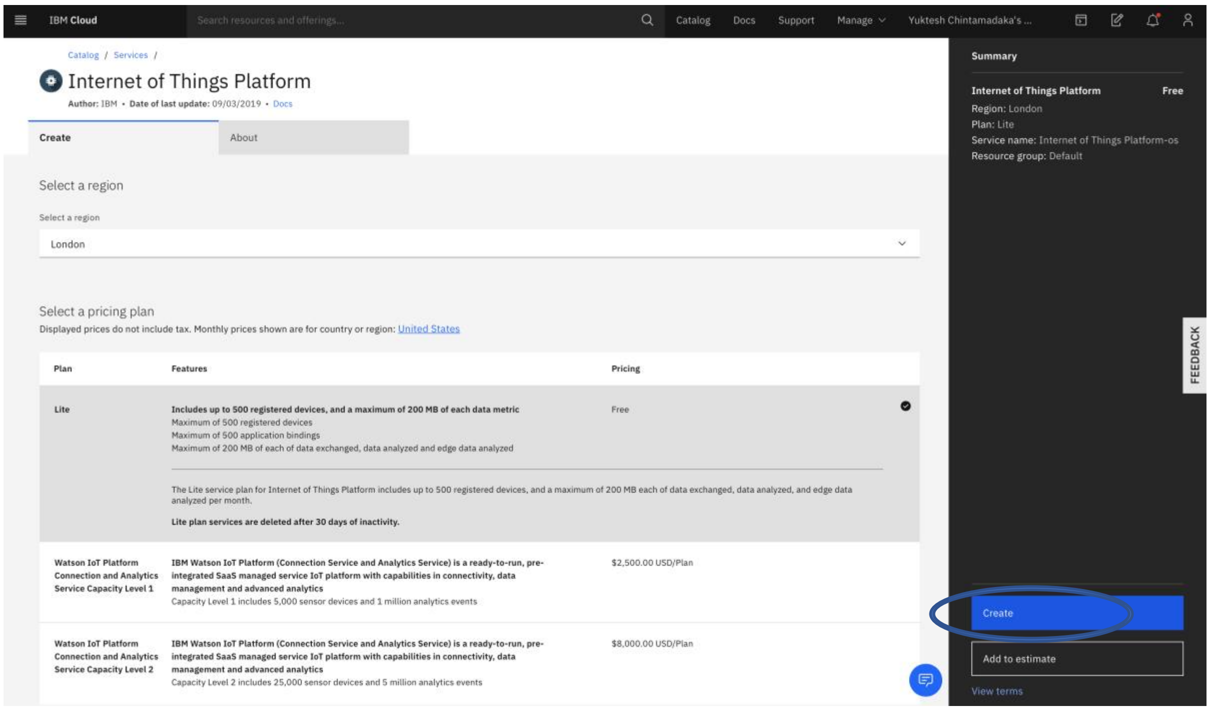


0 resource results found

Catalog Results

Internet of Things Platform

Step 2: Click on ***Create***



Internet of Things Platform

Author: IBM • Date of last update: 09/03/2019 • Docs

Select a region

Select a region: London

Select a pricing plan

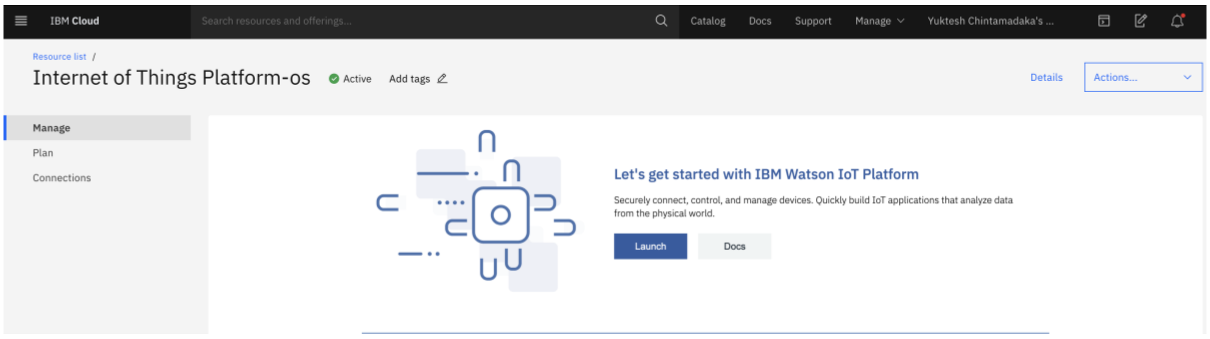
Displayed prices do not include tax. Monthly prices shown are for country or region: United States

Plan	Features	Pricing
Lite	**Includes up to 500 registered devices, and a maximum of 200 MB of each data metric** Maximum of 500 registered devices Maximum of 500 application bindings Maximum of 200 MB of each of data exchanged, data analyzed and edge data analyzed The Lite service plan for Internet of Things Platform includes up to 500 registered devices, and a maximum of 200 MB each of data exchanged, data analyzed, and edge data analyzed per month. **Lite plan services are deleted after 30 days of inactivity.**	Free
Watson IoT Platform Connection and Analytics Service Capacity Level 1	**IBM Watson IoT Platform (Connection Service and Analytics Service) is a ready-to-run, pre-integrated SaaS managed service IoT platform with capabilities in connectivity, data management and advanced analytics** Capacity Level 1 includes 5,000 sensor devices and 1 million analytics events	$2,500.00 USD/Plan
Watson IoT Platform Connection and Analytics Service Capacity Level 2	**IBM Watson IoT Platform (Connection Service and Analytics Service) is a ready-to-run, pre-integrated SaaS managed service IoT platform with capabilities in connectivity, data management and advanced analytics** Capacity Level 2 includes 25,000 sensor devices and 5 million analytics events	$8,000.00 USD/Plan

Summary

Internet of Things Platform — Free
Region: London
Plan: Lite
Service name: Internet of Things Platform-os
Resource group: Default

Step 3: click on ***Launch*** to get started with IBM Watson IoT Platform

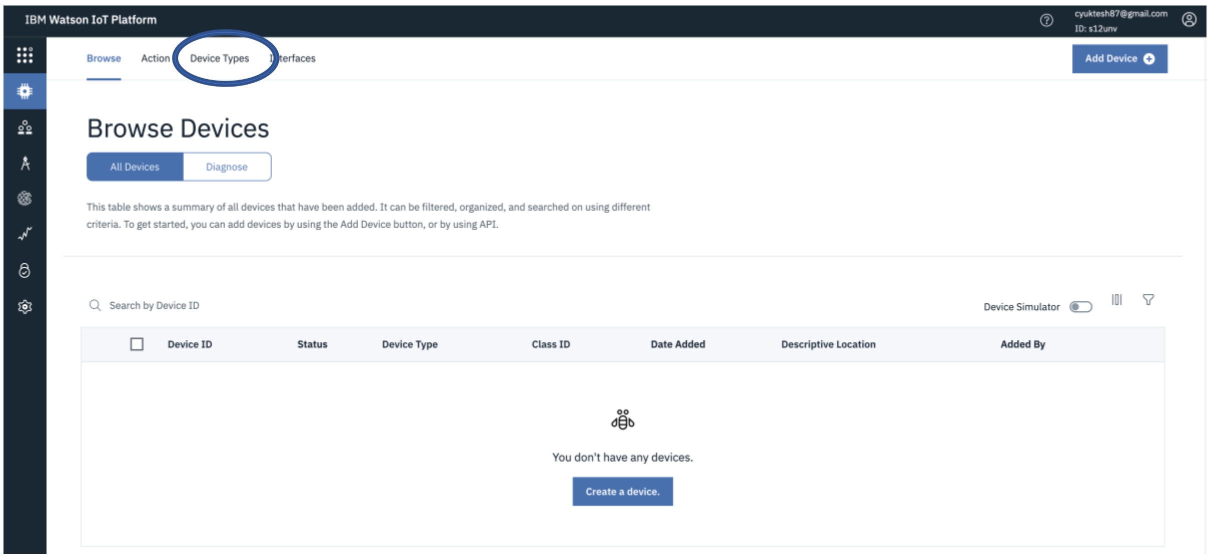


Internet of Things Platform-os ● Active Add tags

Let's get started with IBM Watson IoT Platform

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

Step 4: select ***Device Types*** in menu.



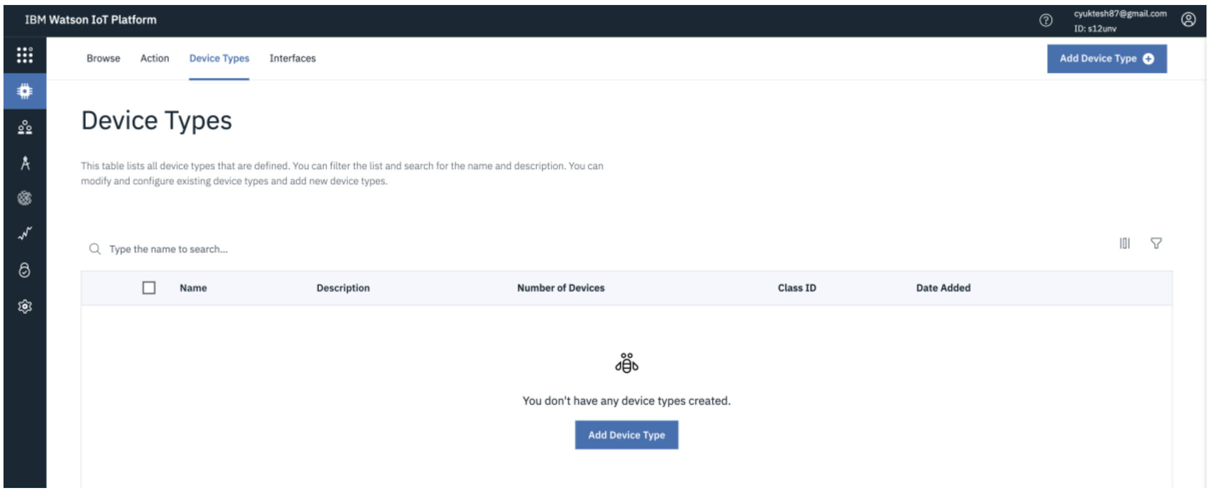
Browse Devices

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

☐	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	Added By

You don't have any devices.

Step 5: Click on ***Add Device Type***


IBM Watson IoT Platform

cyuktesh87@gmail.com
ID: s12unv

Browse Action Device Types Interfaces

Add Device Type

Device Types

This table lists all device types that are defined. You can filter the list and search for the name and description. You can modify and configure existing device types and add new device types.

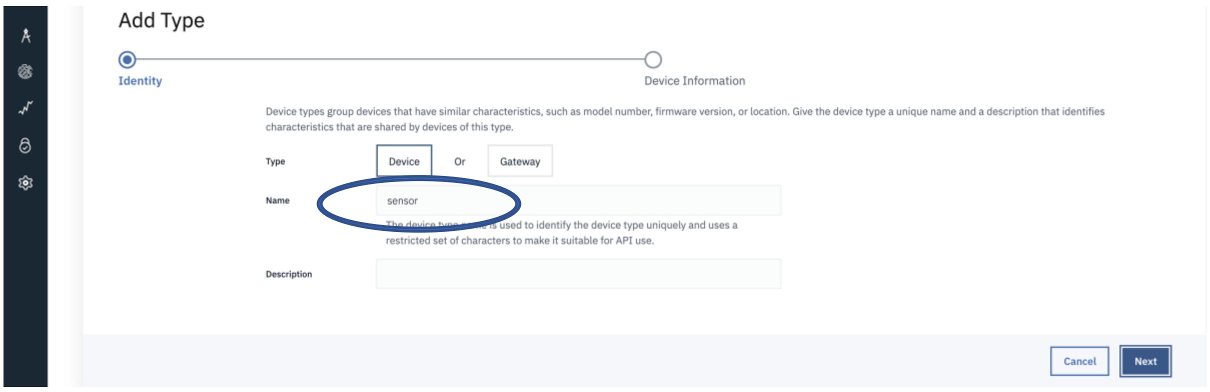
Type the name to search...

	Name	Description	Number of Devices	Class ID	Date Added

You don't have any device types created.

Add Device Type

Step 6: Give Name as ***sensor*** and click on ***Next***


Add Type

Identity

Device Information

Device types group devices that have similar characteristics, such as model number, firmware version, or location. Give the device type a unique name and a description that identifies characteristics that are shared by devices of this type.

Type: Device Or Gateway

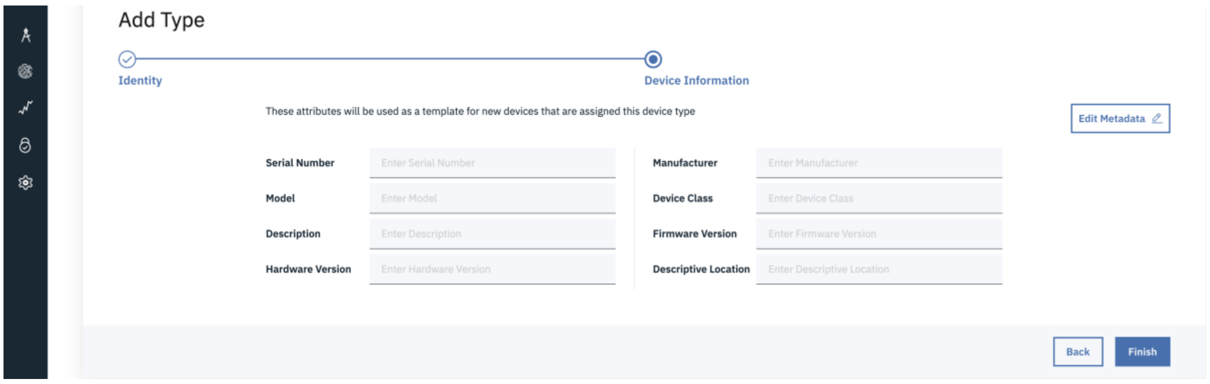
Name: sensor

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

Description:

Cancel Next

Step 7: click on ***Finish***


Add Type

Identity

Device Information

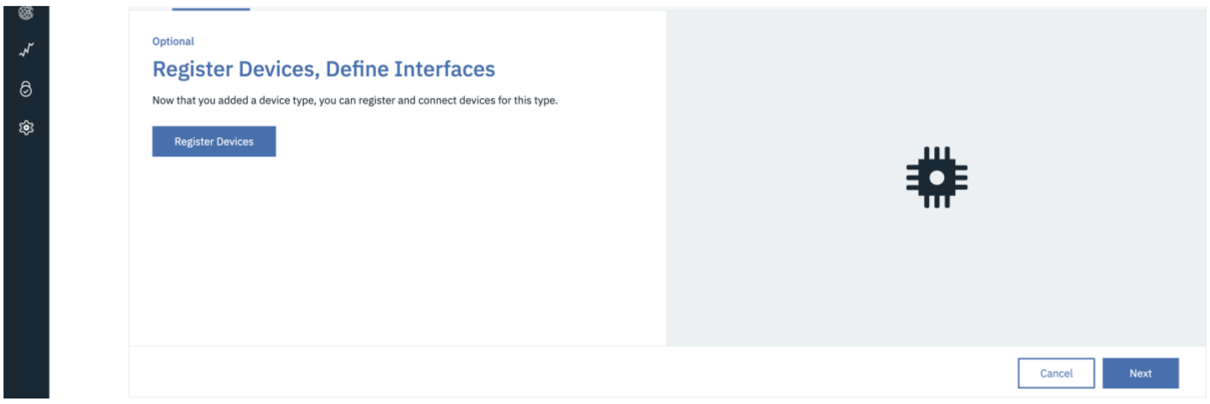
These attributes will be used as a template for new devices that are assigned this device type

Edit Metadata

Serial Number	Enter Serial Number	Manufacturer	Enter Manufacturer
Model	Enter Model	Device Class	Enter Device Class
Description	Enter Description	Firmware Version	Enter Firmware Version
Hardware Version	Enter Hardware Version	Descriptive Location	Enter Descriptive Location

Back Finish

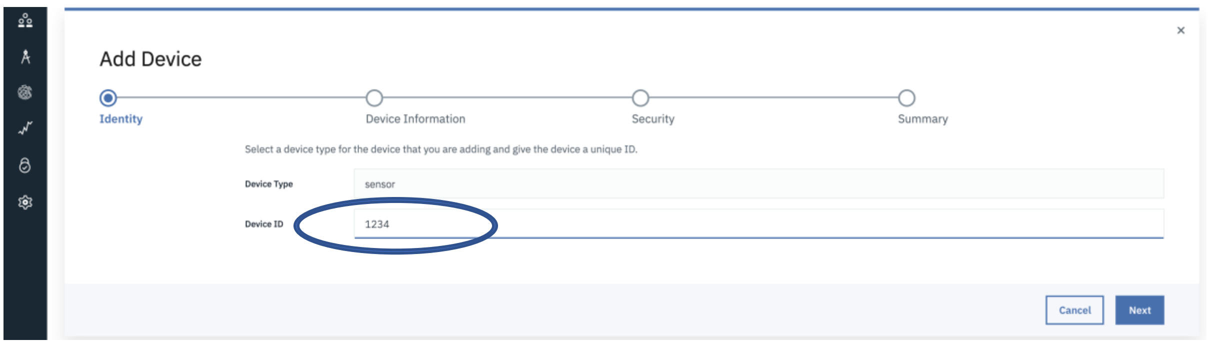
Step 8: click on ***Register Devices***


Optional

Register Devices, Define Interfaces

Now that you added a device type, you can register and connect devices for this type.

Step 9: Enter **Device ID** as ***1234*** and click on ***Next***

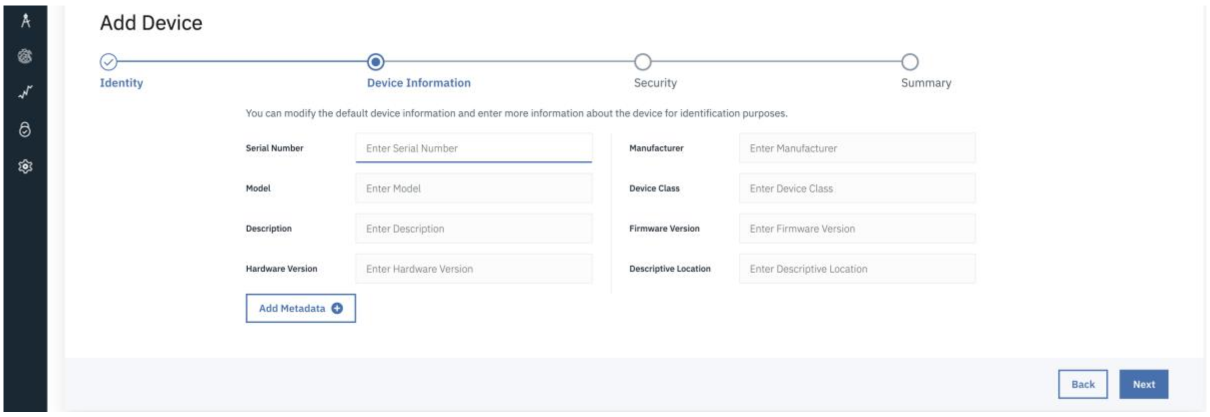

Add Device

Identity — Device Information — Security — Summary

Select a device type for the device that you are adding and give the device a unique ID.

Device Type	sensor
Device ID	1234

Step 10: click on ***Next***

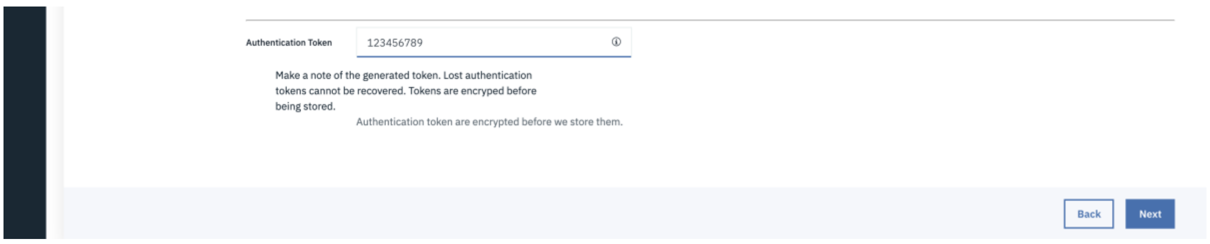

Add Device

Identity — Device Information — Security — Summary

You can modify the default device information and enter more information about the device for identification purposes.

Serial Number	Enter Serial Number	Manufacturer	Enter Manufacturer
Model	Enter Model	Device Class	Enter Device Class
Description	Enter Description	Firmware Version	Enter Firmware Version
Hardware Version	Enter Hardware Version	Descriptive Location	Enter Descriptive Location

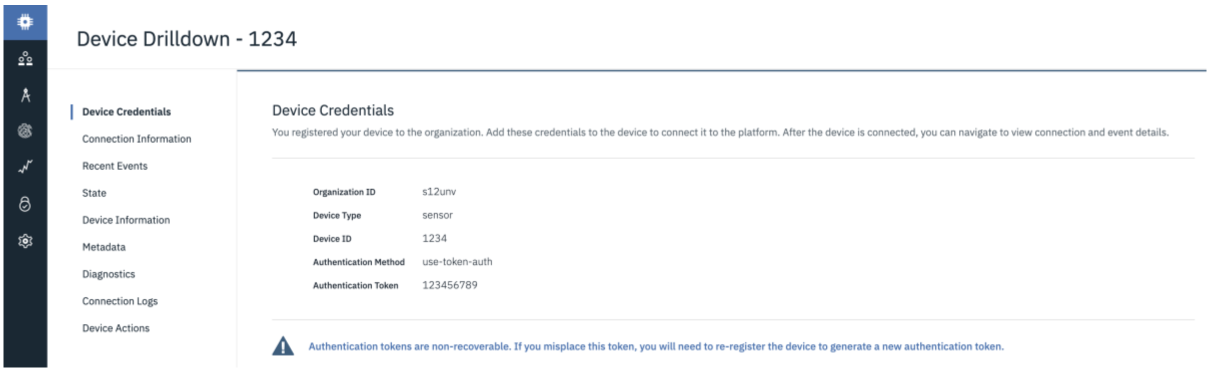
Step 11: Enter ***Authentication Token*** give as ***123456789*** and click ***Next*** and ***Finish*** in next page.


Authentication Token	123456789

Make a note of the generated token. Lost authentication tokens cannot be recovered. Tokens are encryped before being stored.

Authentication token are encrypted before we store them.

Step 12: Save ***Device Credentials*** for future use in a notepad or word doc



Device Drilldown - 1234

Device Credentials

You registered your device to the organization. Add these credentials to the device to connect it to the platform. After the device is connected, you can navigate to view connection and event details.

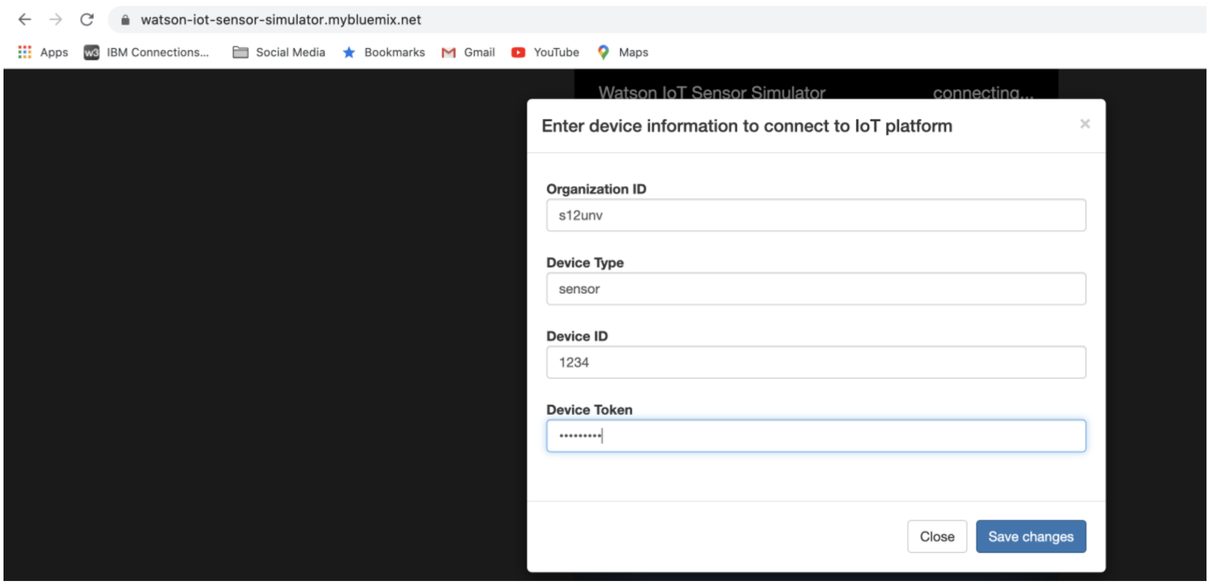
Organization ID	s12unv
Device Type	sensor
Device ID	1234
Authentication Method	use-token-auth
Authentication Token	123456789

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

Organization ID:	s12unv
Device Type:	sensor
Device ID:	1234
Authentication Method:	use-token-auth
Authentication Token:	123456789

Task 3: Connect device to Watson-IoT sensor simulator

Step 1: Open URL https://watson-iot-sensor-simulator.mybluemix.net/ give below details and click on ***Save changes***


watson-iot-sensor-simulator.mybluemix.net

Watson IoT Sensor Simulator connecting...

Enter device information to connect to IoT platform

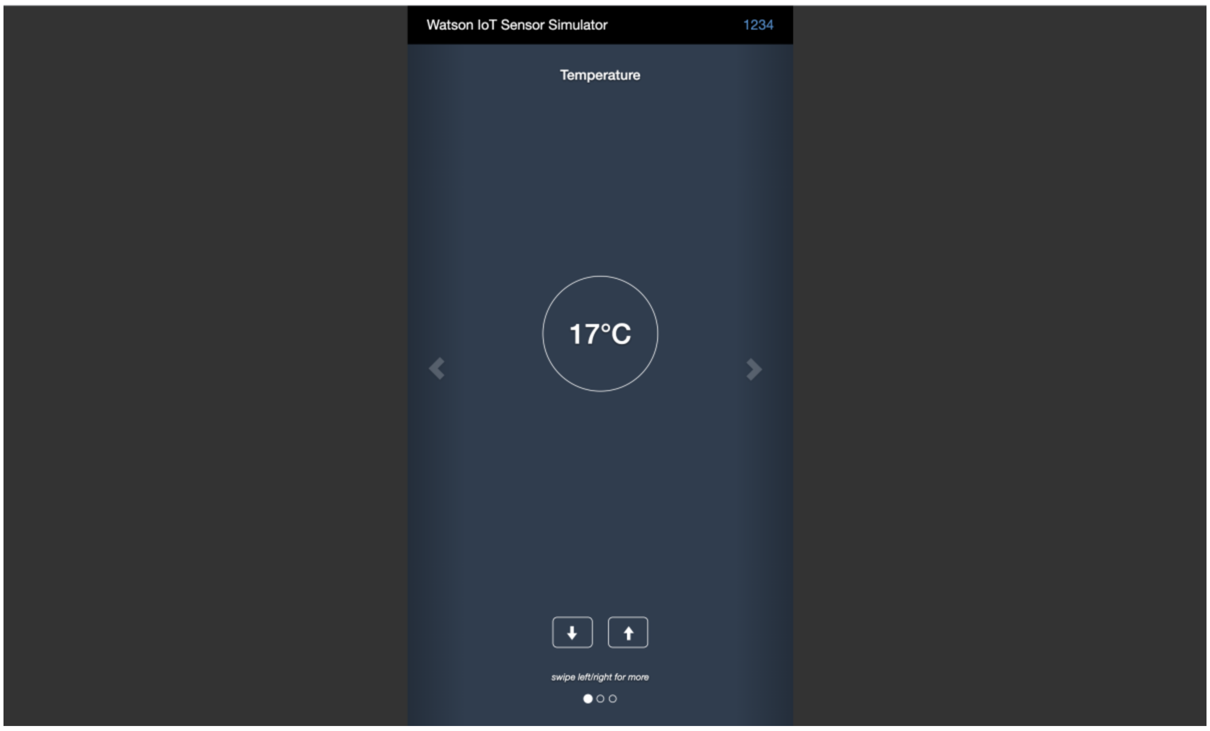
Organization ID: s12unv

Device Type: sensor

Device ID: 1234

Device Token: •••••••••

Step 2: You can change ***Temperature, Humidity, Object Temperature*** values by clicking ***arrows***

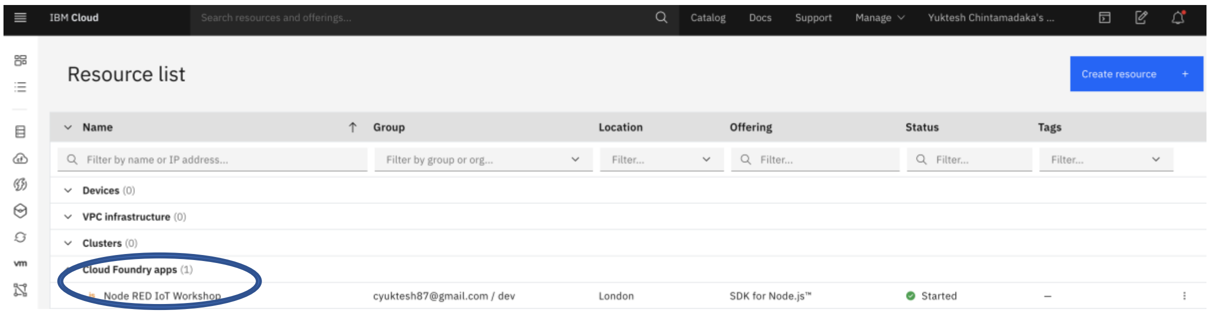


Task 4: Build Node-Red Application

Sub Task 1: Connect to device and fetch device data into Node-Red app
Sub Task 2: Connect Node-Red app to Database
Sub Task 3: Connect Node-Red app to Mobile

Sub Task 1: Connect to device and fetch device data into Node-Red app

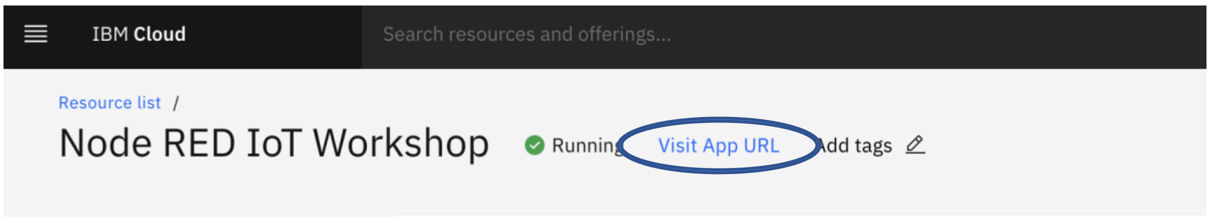
Step 1: Go to ***IBM Cloud Dashboard*** click on ***Cloud Foundry Apps*** and click on ***Node RED IoT Workshop***


IBM Cloud | Search resources and offerings... | Catalog | Docs | Support | Manage | Yuktesh Chintamadaka's ...

Resource list | Create resource +

Name	Group	Location	Offering	Status	Tags
Filter by name or IP address...	Filter by group or org...	Filter...	Filter...	Filter...	Filter...
Devices (0)					
VPC infrastructure (0)					
Clusters (0)					
Cloud Foundry apps (1)					
Node RED IoT Workshop	cyuktesh87@gmail.com / dev	London	SDK for Node.js™	Started	–

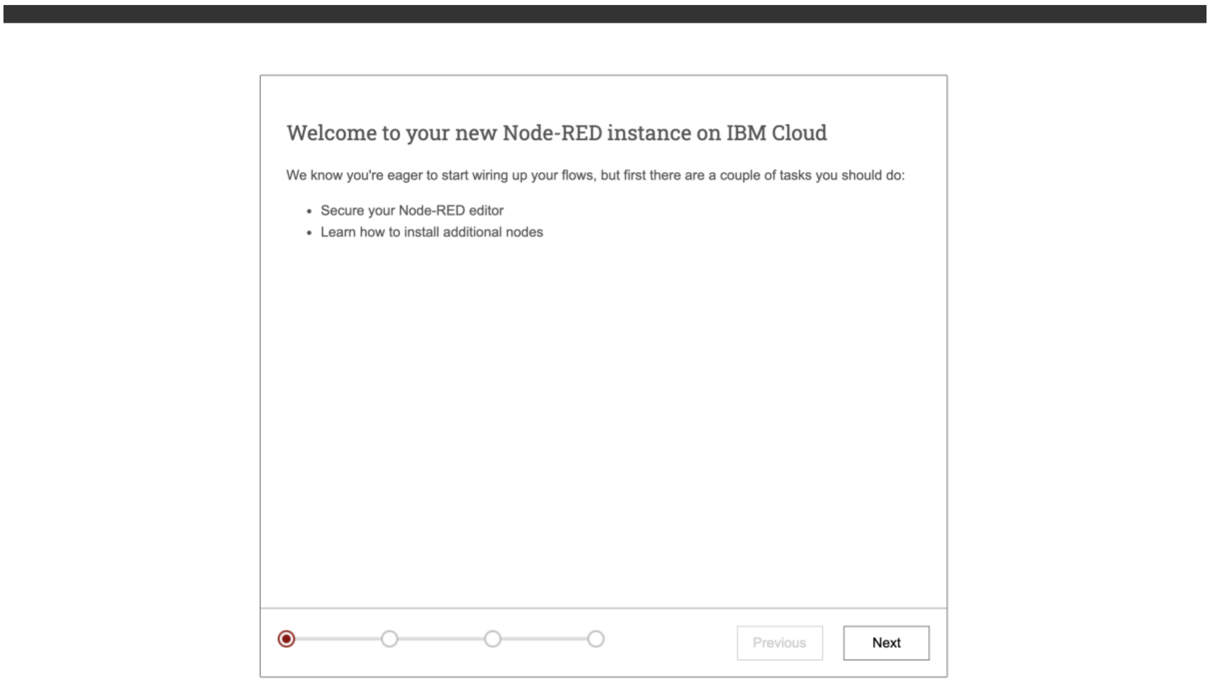
Step 2: click on ***Visit App URL***


IBM Cloud | Search resources and offerings...

Resource list /

Node RED IoT Workshop | Running | Visit App URL | Add tags

Step 3: click on ***Next***

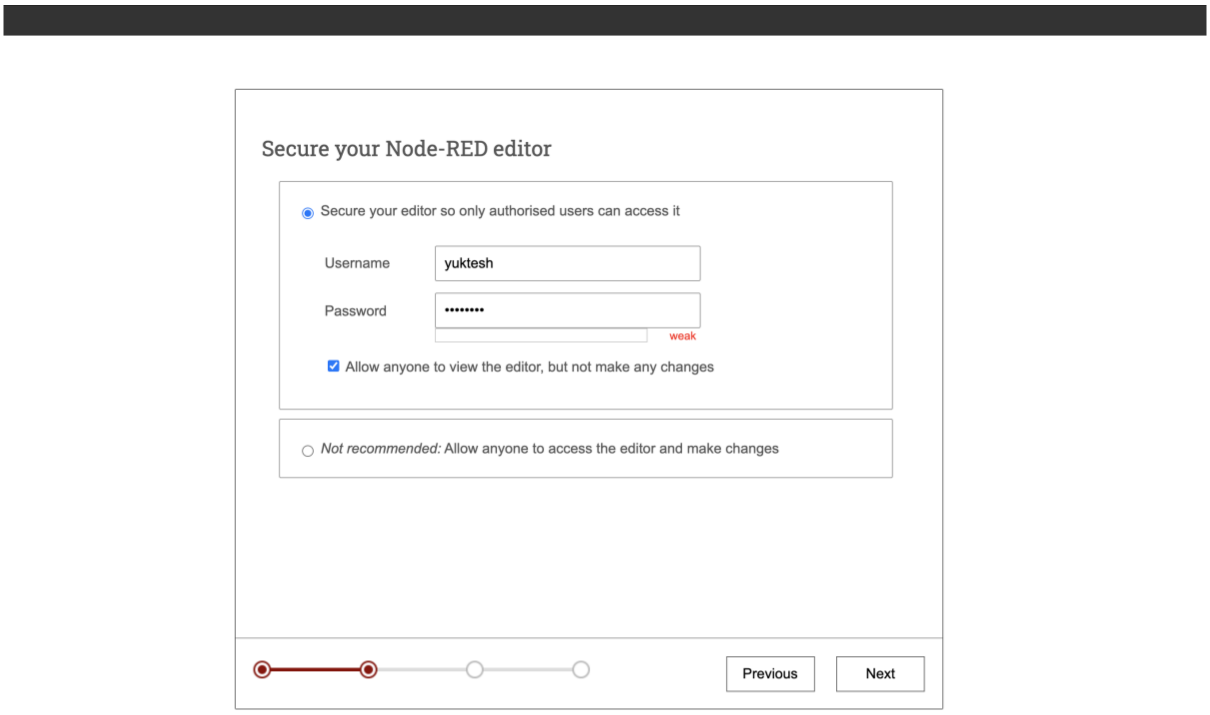

Welcome to your new Node-RED instance on IBM Cloud

We know you're eager to start wiring up your flows, but first there are a couple of tasks you should do:

- Secure your Node-RED editor
- Learn how to install additional nodes

Previous | Next

Step 4: Enter ***Username*** and ***Password*** to secure your Node-RED editor and click on ***Next*** → ***Next*** → ***Finish***



Secure your Node-RED editor

● Secure your editor so only authorised users can access it

Username: yuktesh

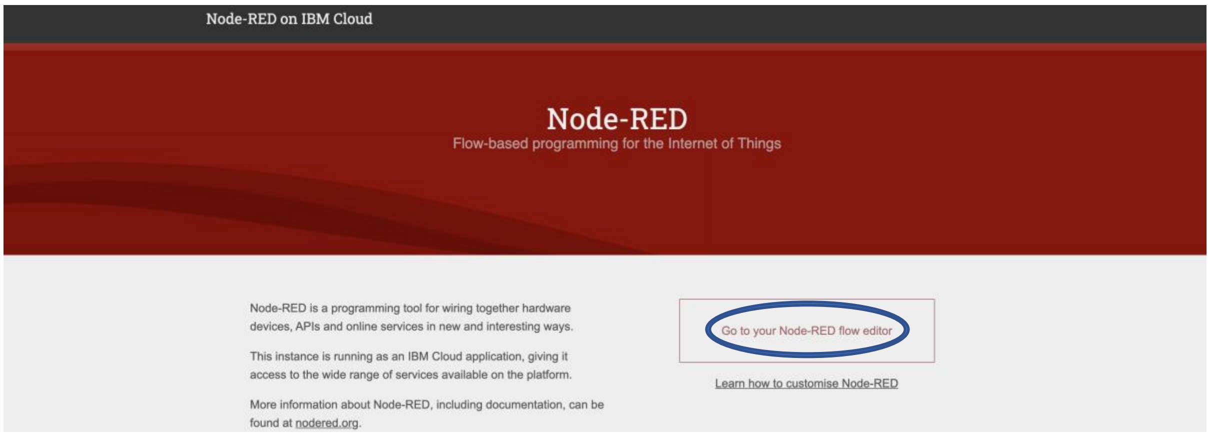
Password: ••••••••

weak

☑ Allow anyone to view the editor, but not make any changes

○ *Not recommended:* Allow anyone to access the editor and make changes

Step 5: Click on ***Go to your Node-RED flow editor***



Node-RED on IBM Cloud

Node-RED

Flow-based programming for the Internet of Things

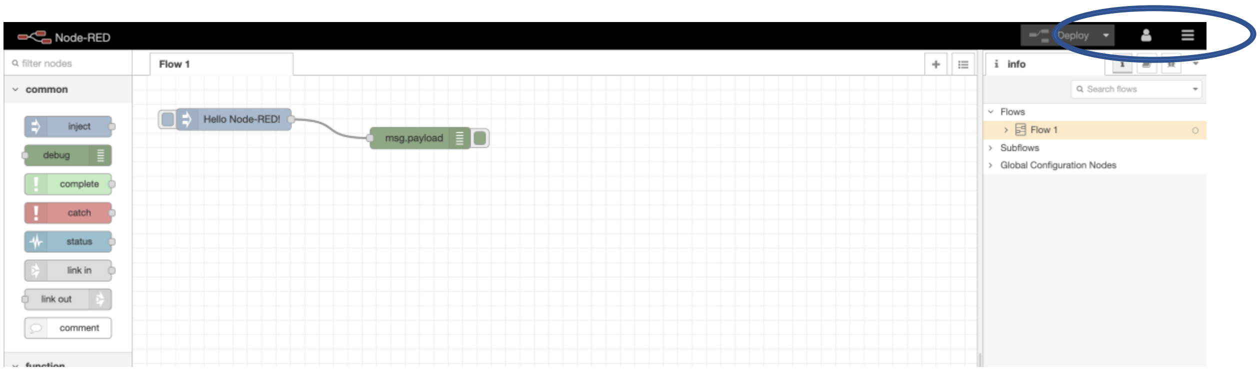
Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.

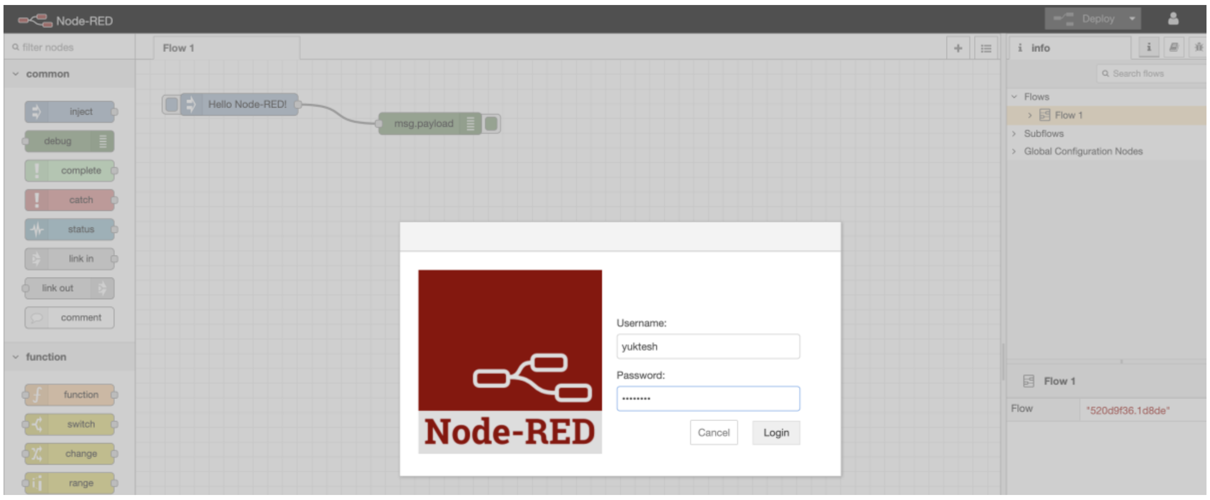
More information about Node-RED, including documentation, can be found at nodered.org.

Go to your Node-RED flow editor
Learn how to customise Node-RED

Step 6: Click on icon and click on Login

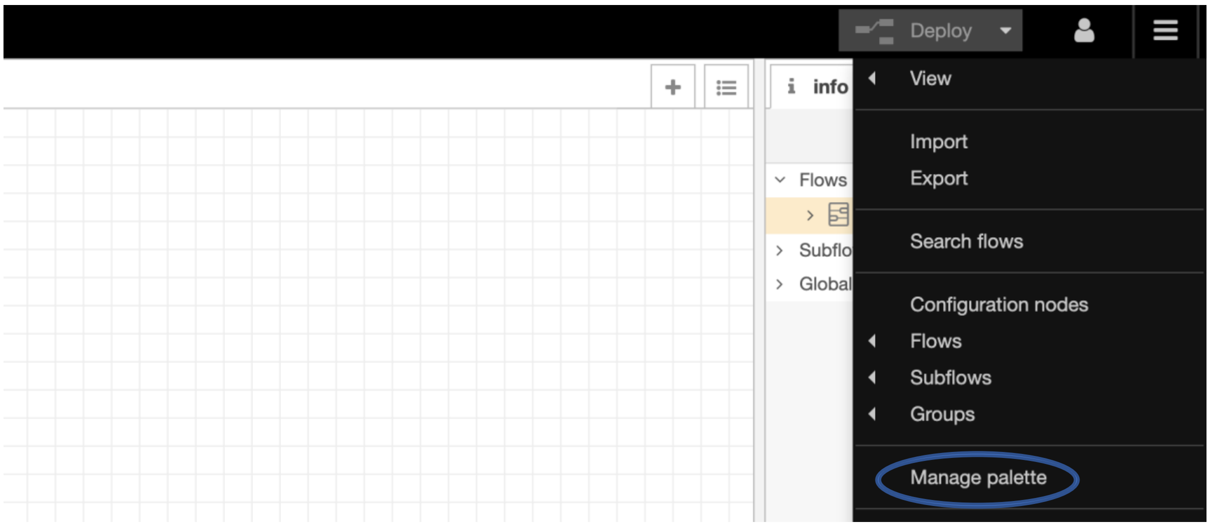


Step 7: Enter Username and **Password** which you created and in **step 4** and click on ***Login***

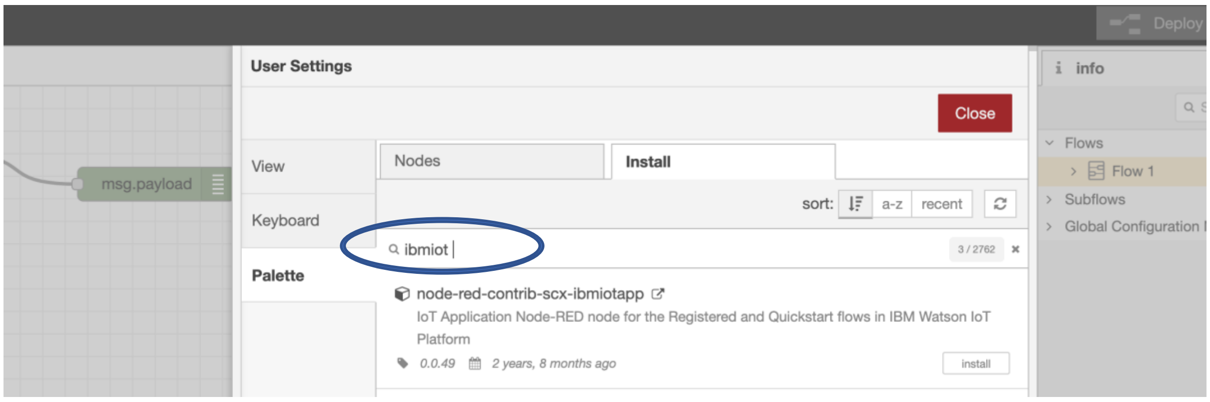


Username: yuktesh
Password: ········

Step 8: Install ***ibmiotapp*** node by click ☰ icon and click on ***Manage palette***



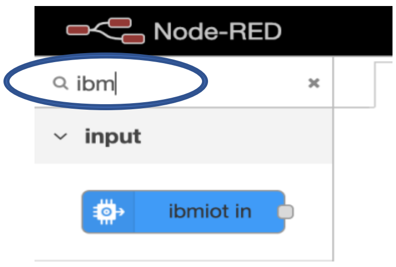
Step 9: Under **Install tab** search for ***ibmiot*** and click on ***install*** and again click on ***install*** in next window**.** Therefore ***node-red-contrib-scx-ibmiotapp*** node gets installed. Click on ***close.***



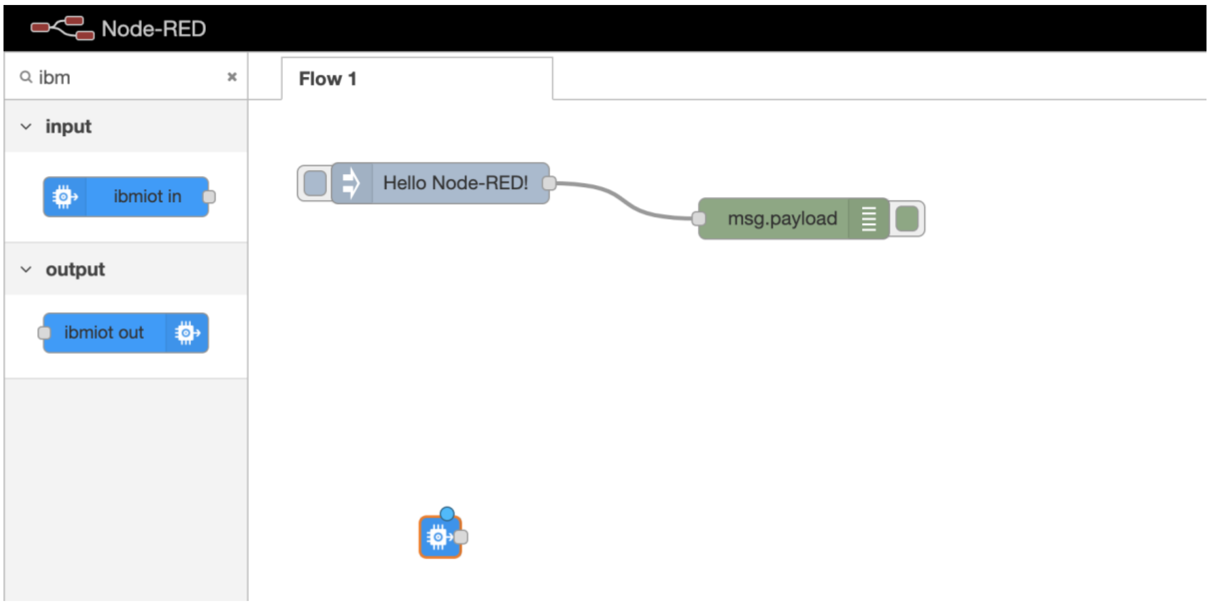
ibmiot
3 / 2762

node-red-contrib-scx-ibmiotapp
IoT Application Node-RED node for the Registered and Quickstart flows in IBM Watson IoT Platform
0.0.49 2 years, 8 months ago install

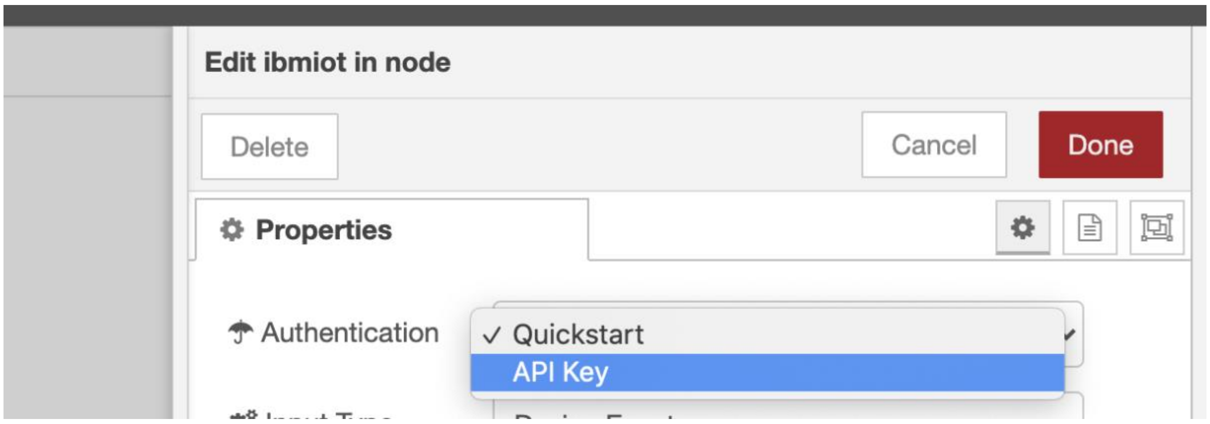
Step 10: Search for ***IBM***



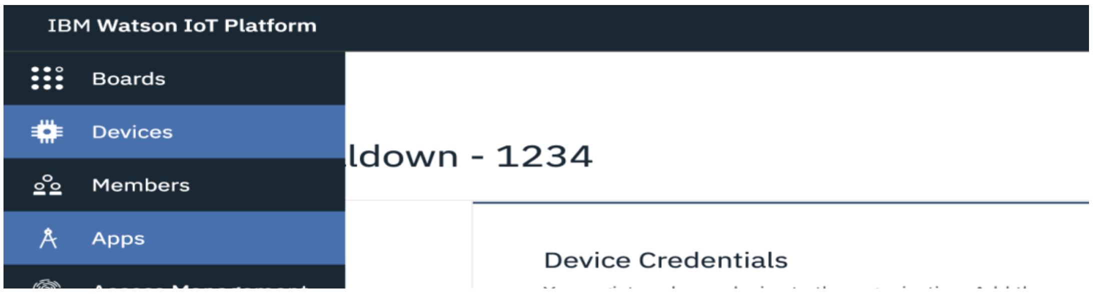
Step 11: Drag and drop the **ibmiot** in node into **flow**



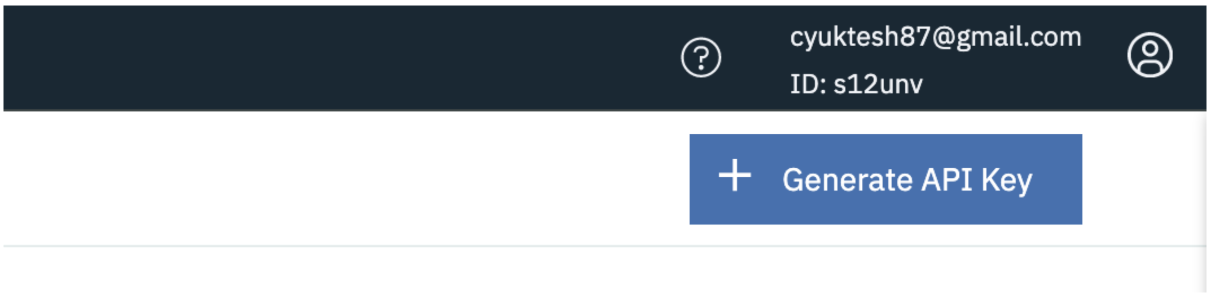
Step 12: Double click on ***IBM IoT* in** node and change Authentication ***API Key***



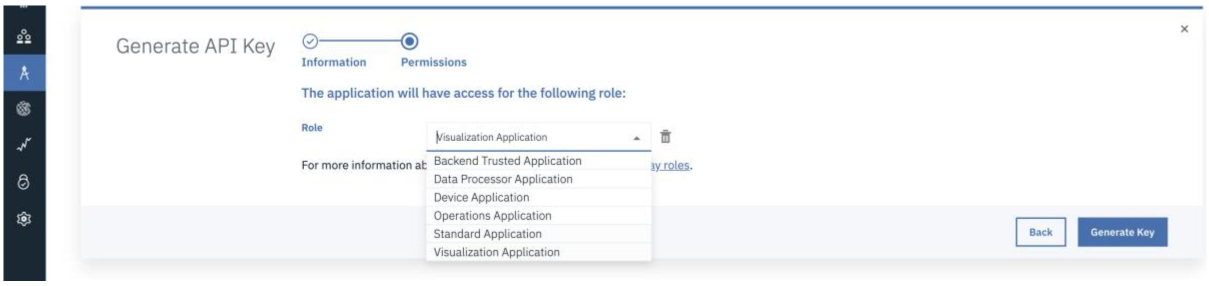
Step 12: Go to *IBM Watson IoT Platform* and click on ***Apps***



Step 13: Click on ***Generate API Key*** and click on ***Next*** in next page



Step 14: Select any role and click on **Generate Key**



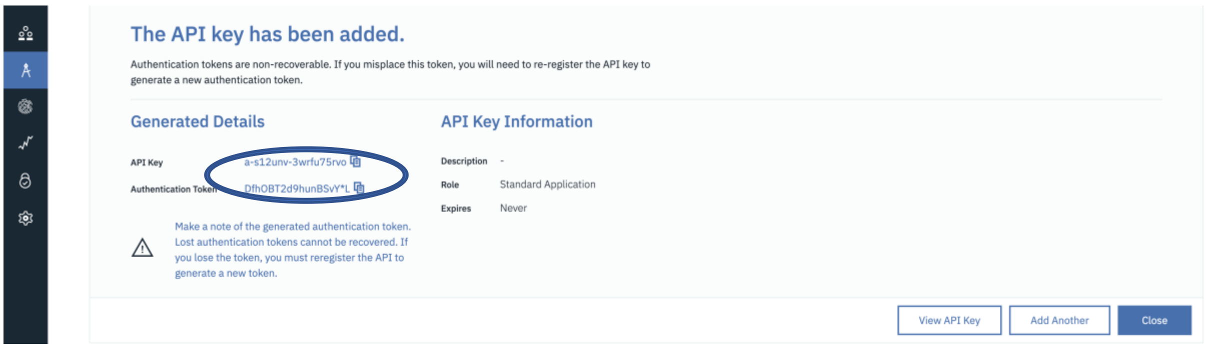
The application will have access for the following role:

Role: Visualization Application

For more information ab … ay roles.

- Backend Trusted Application
- Data Processor Application
- Device Application
- Operations Application
- Standard Application
- Visualization Application

Step 15: Copy ***API Key*** and ***Authentication*** in a notepad or word doc for future use.



The API key has been added.

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

Generated Details

API Key	a-s12unv-3wrfu75rvo
Authentication Token	DfhOBT2d9hunBSvY*L

Make a note of the generated authentication token. Lost authentication tokens cannot be recovered. If you lose the token, you must reregister the API to generate a new token.

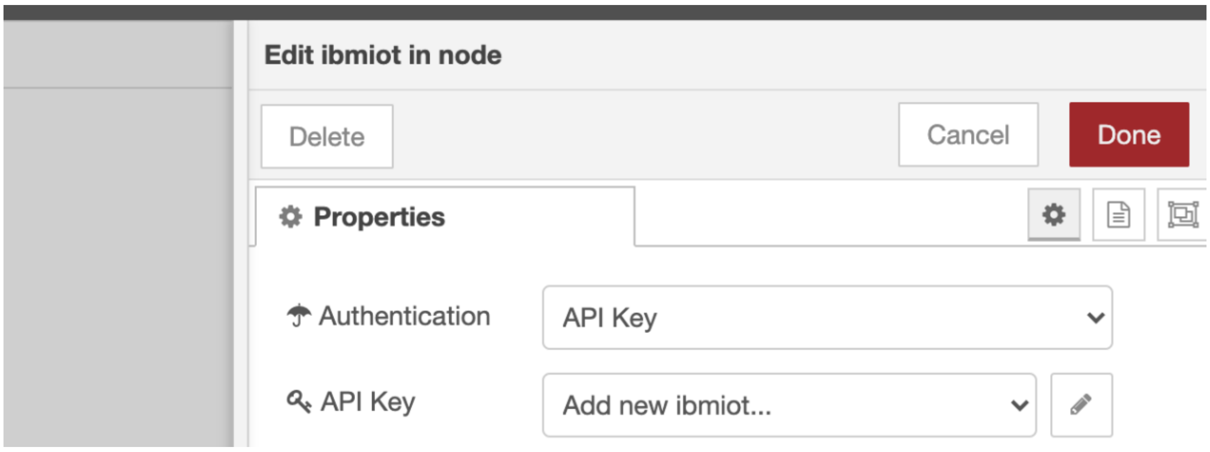
API Key Information

Description	-
Role	Standard Application
Expires	Never

My Device API Key: a-s12unv-3wrfu75rvo

Authentication Token: DfhOBT2d9hunBSvY*L

Step 16: Click on ✎ in below page

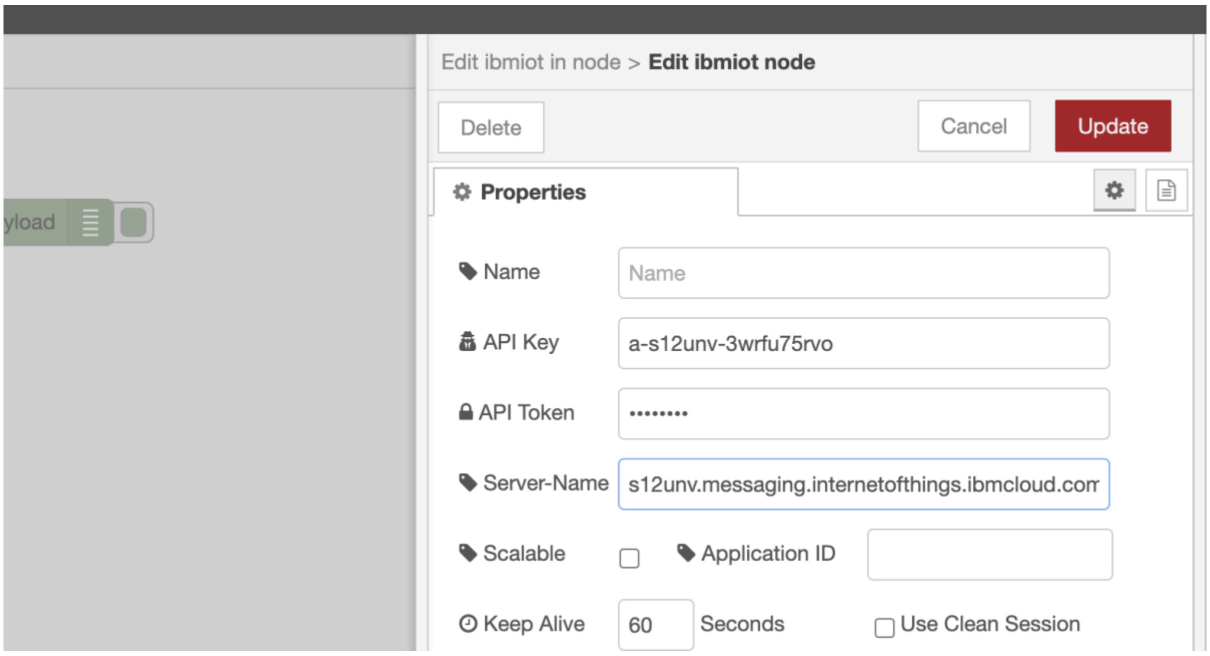


Authentication: API Key

API Key: Add new ibmiot...

Step 17: Enter ***API Key***, ***Authentication Token*** taken in **step 15** and ***Server-name*** as ***s12unv.messaging.internetofthings.ibmcloud.com*** and uncheck ***Use Clean Session*** and click on ***Update***

Note: Orgid take from Task 2, step 12



Name: Name

API Key: a-s12unv-3wrfu75rvo

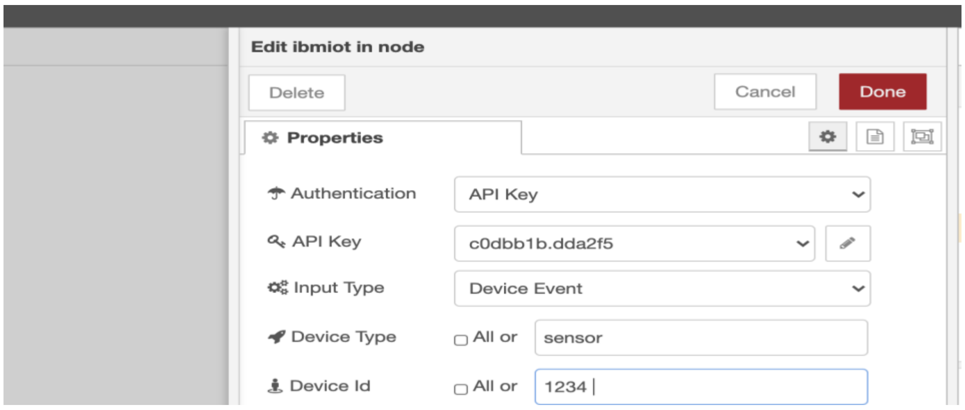
API Token: ••••••••

Server-Name: s12unv.messaging.internetofthings.ibmcloud.com

Scalable ☐ Application ID

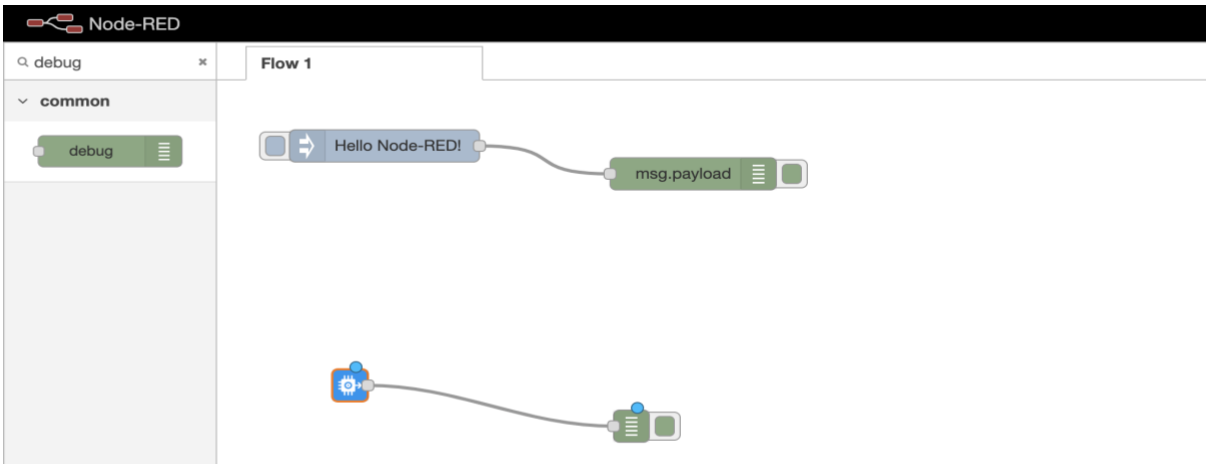
Keep Alive 60 Seconds ☐ Use Clean Session

Step 18: enter ***Device Type*** and ***Device Id*** as ***sensor*** and ***1234*** click on ***Done*** these are details taken from ***Task 1 step 12.***

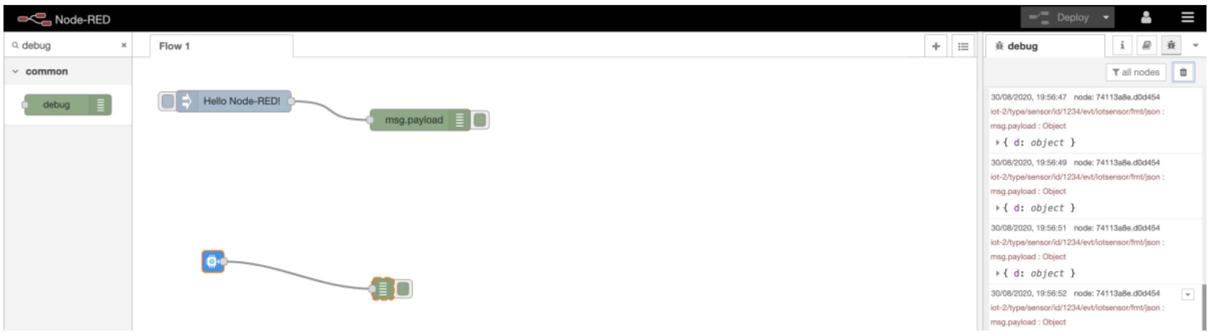

Edit ibmiot in node

Authentication	API Key
API Key	c0dbb1b.dda2f5
Input Type	Device Event
Device Type	☐ All or sensor
Device Id	☐ All or 1234

Step 19: Search for ***debug*** in filter nodes, Now drag and drop ***Debug*** node into flow and connect both the nodes like below



Step 20: Click on ***Deploy*** on right corner and click on ***Debug messages*** icon.

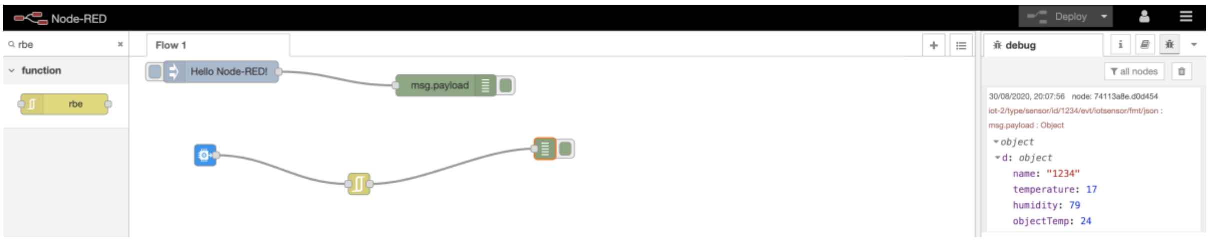


```
30/08/2020, 19:56:47   node: 74113a8e.d0d454
iot-2/type/sensor/id/1234/evt/iotsensor/fmt/json :
msg.payload : Object
 ▸{ d: object }
30/08/2020, 19:56:49   node: 74113a8e.d0d454
iot-2/type/sensor/id/1234/evt/iotsensor/fmt/json :
msg.payload : Object
 ▸{ d: object }
30/08/2020, 19:56:51   node: 74113a8e.d0d454
iot-2/type/sensor/id/1234/evt/iotsensor/fmt/json :
msg.payload : Object
 ▸{ d: object }
30/08/2020, 19:56:52   node: 74113a8e.d0d454
iot-2/type/sensor/id/1234/evt/iotsensor/fmt/json :
msg.payload : Object
```

Now data from created device is reading by our node-red app.

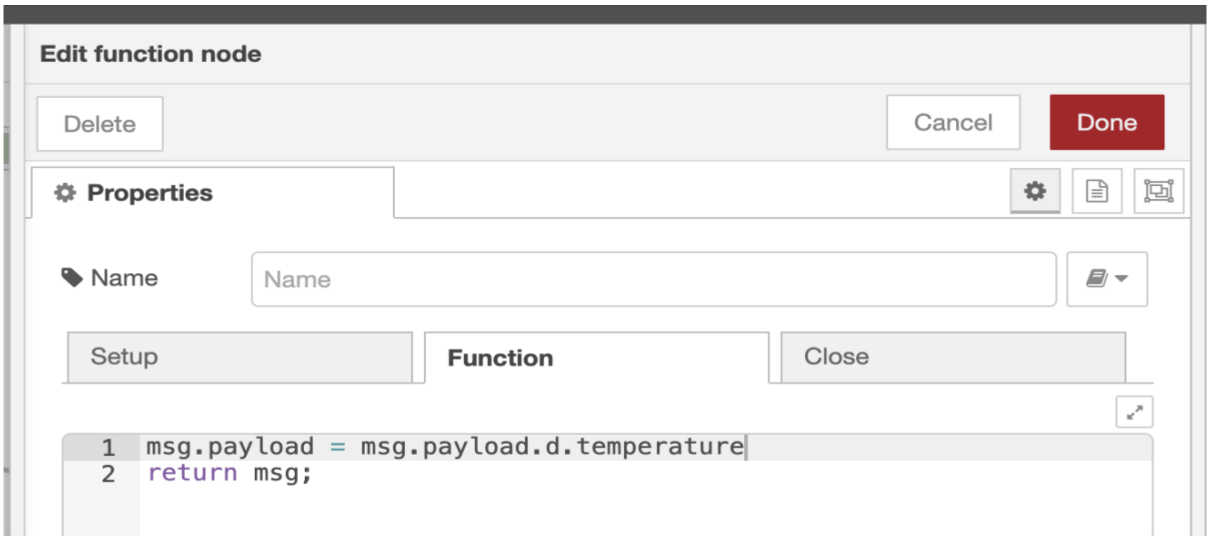
Simulator – Watson IoT Platform Device – IBM Cloud Node-RED application

Step 21: To stop continuous incoming flow of messages keep ***rbe*** node in between both ***IBM IoT in*** node and ***Debug*** node



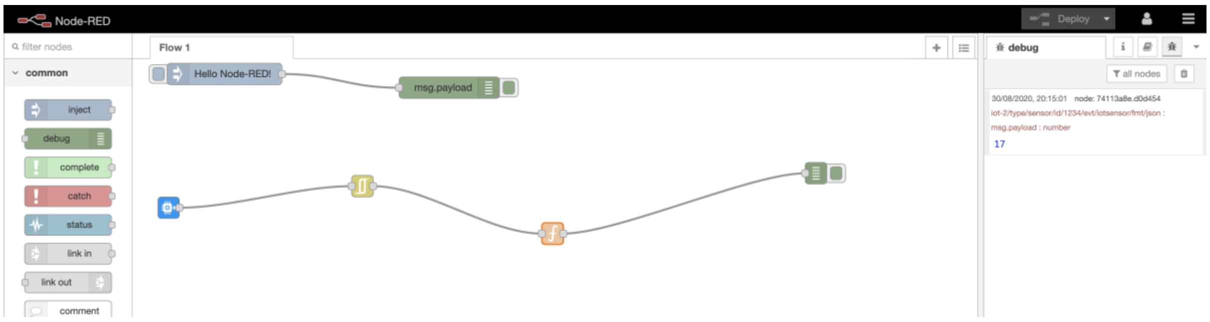
```
30/08/2020, 20:07:56   node: 74113a8e.d0d454
iot-2/type/sensor/id/1234/evt/iotsensor/fmt/json :
msg.payload : Object
▾object
 ▾d: object
    name: "1234"
    temperature: 17
    humidity: 79
    objectTemp: 24
```

Step 22: Drag and drop ***function*** node and edit like below



```
msg.payload = msg.payload.d.temperature
return msg;
```

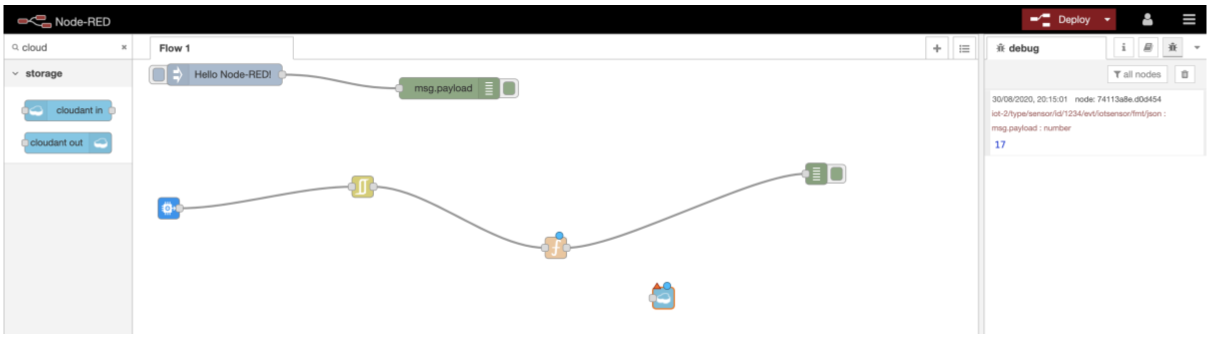
Step 23: click on 🗑 clear and click on Deploy and your flow looks like below



```
30/08/2020, 20:15:01   node: 74113a8e.d0d454
iot-2/type/sensor/id/1234/evt/iotsensor/fmt/json :
msg.payload : number
17
```

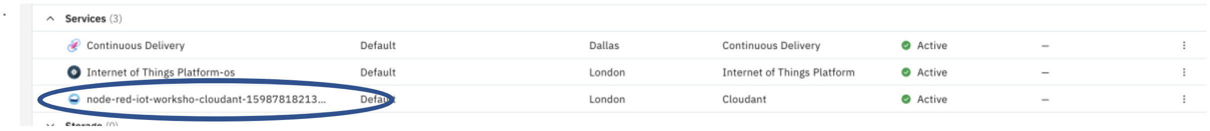
Sub Task 2: Connect Node-Red app to Database

Step 1: Drag and drop ***cloudant in*** node into flow and connect like below



30/08/2020, 20:15:01 node: 74113a8e.d0d454
iot-2/type/sensor/id/1234/evt/iotsensor/fmt/json :
msg.payload : number
17

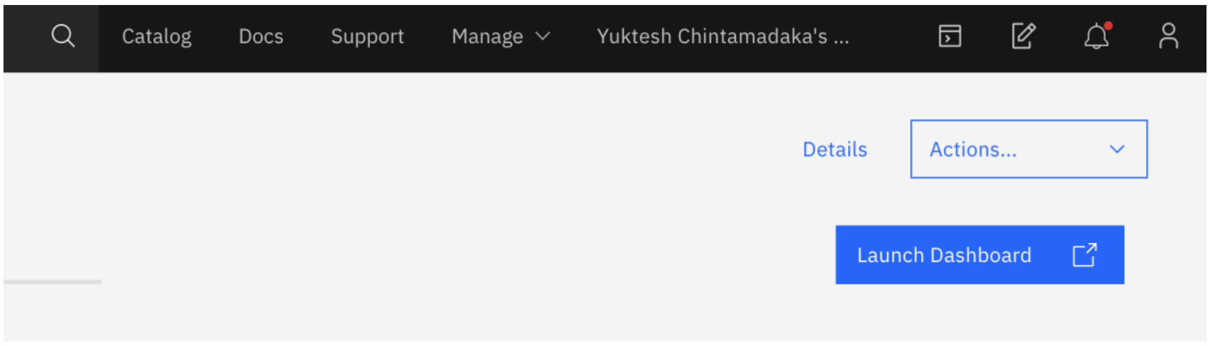
Step 2: Go to ***IBM Cloud Dashboard*** → ***Services*** → click on ***node-red-iot-workshop-cloudant-15987818213***



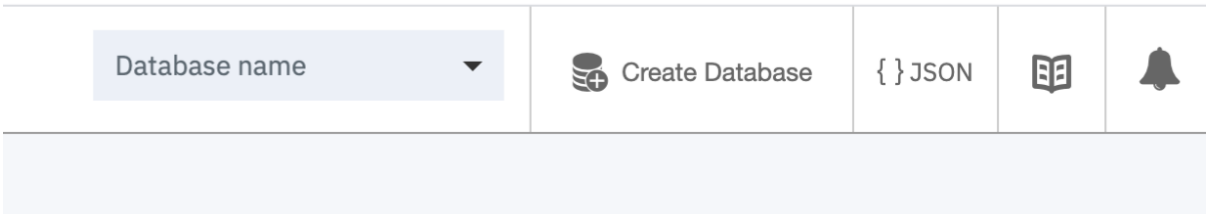
Services (3)

Continuous Delivery	Default	Dallas	Continuous Delivery	Active	–	⋮
Internet of Things Platform-os	Default	London	Internet of Things Platform	Active	–	⋮
node-red-iot-worksho-cloudant-15987818213...	Default	London	Cloudant	Active	–	⋮

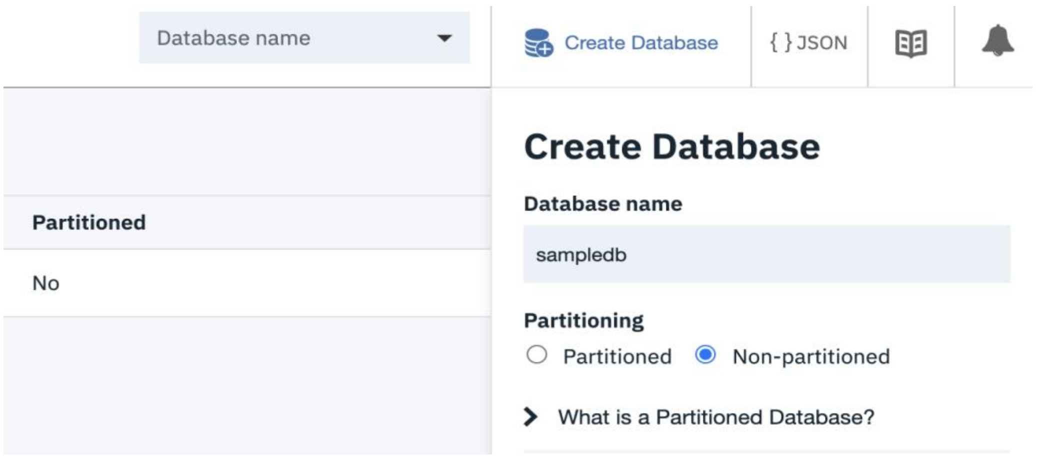
Step 3: click on **Launch**



Step 4: click on ***Create Database***



Step 5: Give Database name as ***sampledb*** and click on ***create***



Create Database

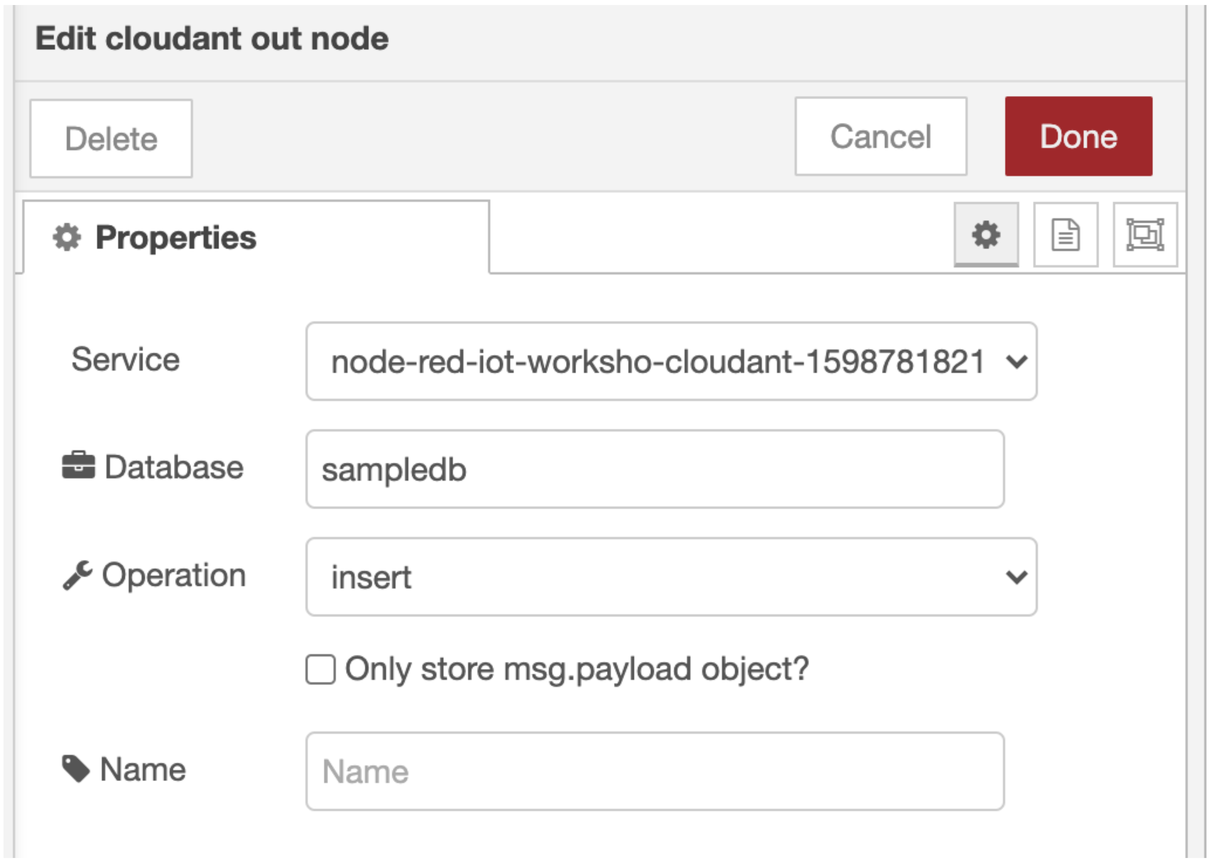
Database name: sampledb

Partitioning: ○ Partitioned [✓ Non-partitioned]

What is a Partitioned Database?

Partitioned: No

Step 6: Double click on *cloudant in* node, give Database name as ***sampledb*** and click on ***Done***



Edit cloudant out node

Service: node-red-iot-worksho-cloudant-1598781821

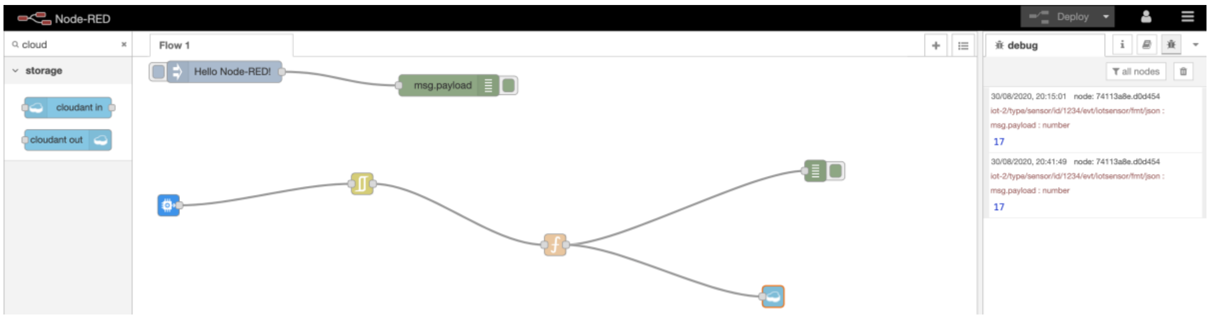
Database: sampledb

Operation: insert

☐ Only store msg.payload object?

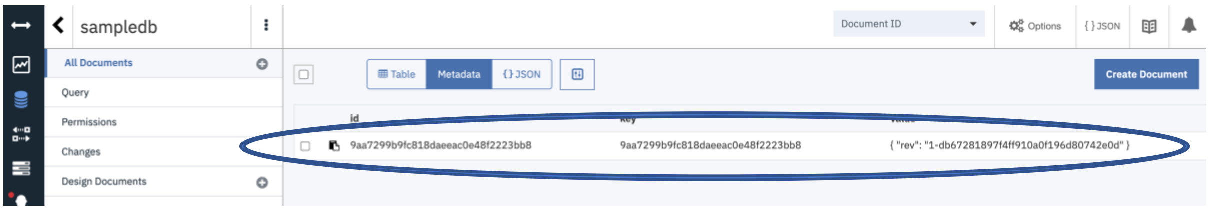
Name: Name

Step 7: Click on ***Deploy*** so that you can see ***Debug window*** 1 more message with Temperature as ***17***



```
30/08/2020, 20:15:01   node: 74113a8e.d0d454
iot-2/type/sensor/id/1234/evt/iotsensor/fmt/json :
msg.payload : number
17
30/08/2020, 20:41:49   node: 74113a8e.d0d454
iot-2/type/sensor/id/1234/evt/iotsensor/fmt/json :
msg.payload : number
17
```

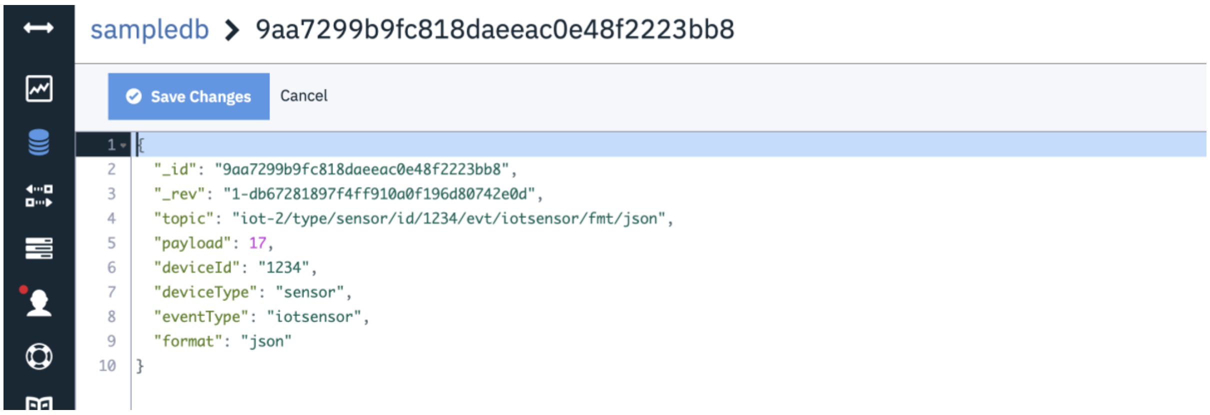
Step 8: Go to **Cloudant DB** and refresh the ***sample db*** you can see one document.



id	key	value
9aa7299b9fc818daeeac0e48f2223bb8	9aa7299b9fc818daeeac0e48f2223bb8	{ "rev": "1-db67281897f4ff910a0f196d80742e0d" }

This is the Document came from Node-RED app

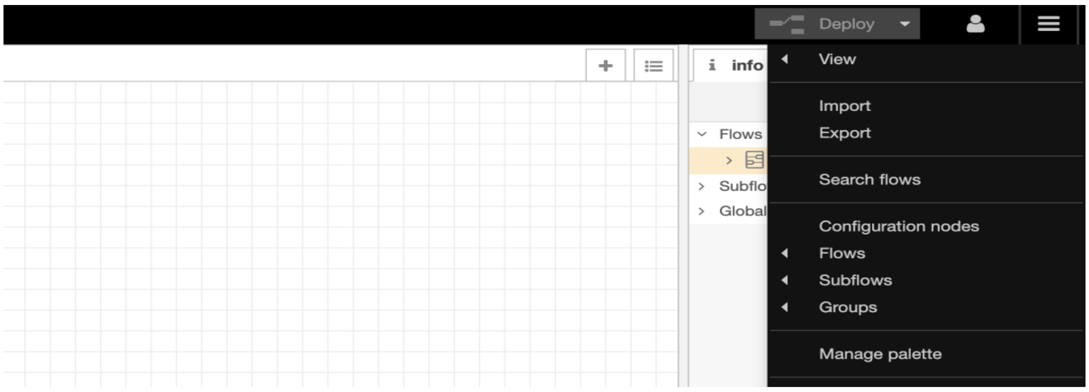
Step 9: Click on that one Document you can see the full details of the Document in json format



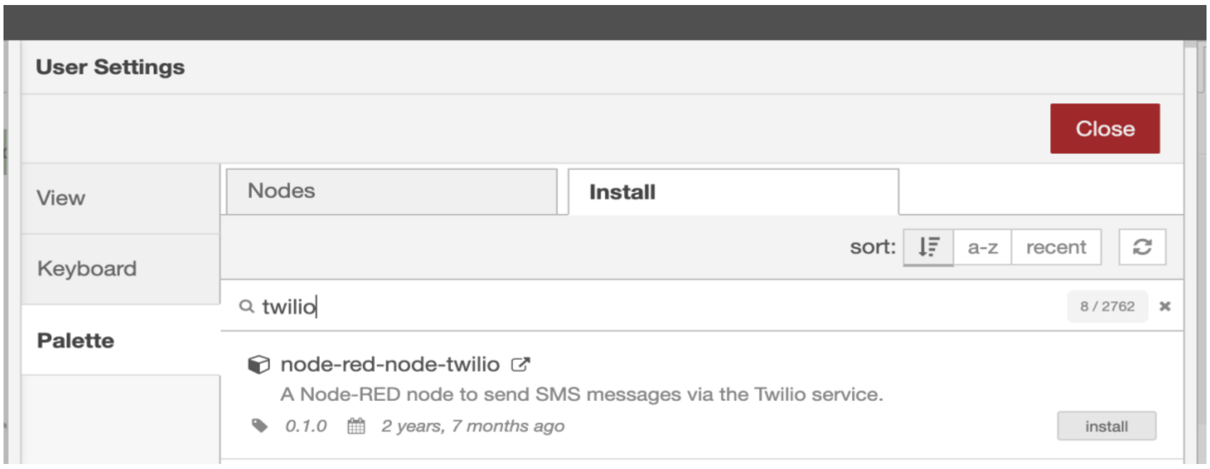
```json
{
  "_id": "9aa7299b9fc818daeeac0e48f2223bb8",
  "_rev": "1-db67281897f4ff910a0f196d80742e0d",
  "topic": "iot-2/type/sensor/id/1234/evt/iotsensor/fmt/json",
  "payload": 17,
  "deviceId": "1234",
  "deviceType": "sensor",
  "eventType": "iotsensor",
  "format": "json"
}
```

Sub Task 3: Connect Node-Red app to Mobile

Step 1: Install ***twilio*** node by click ☰ icon and click on ***Manage palette***



Step 2: Under **Install tab** search for ***twilio*** and click on ***install*** and again click on ***install*** in next window**.** Therefore ***node-red-node-twilio*** node gets installed. Click on ***close.***



User Settings

Close

View | Nodes | Install

Keyboard

Palette

sort: a-z recent

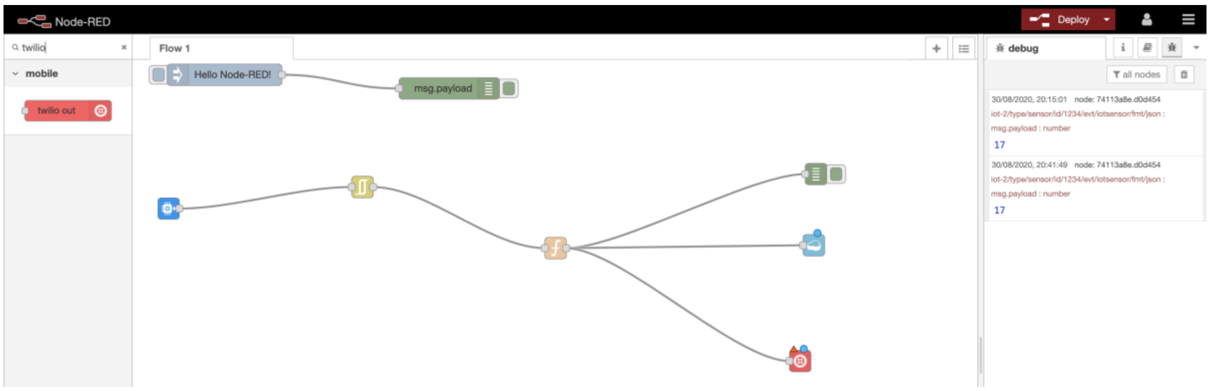
twilio 8 / 2762

node-red-node-twilio

A Node-RED node to send SMS messages via the Twilio service.

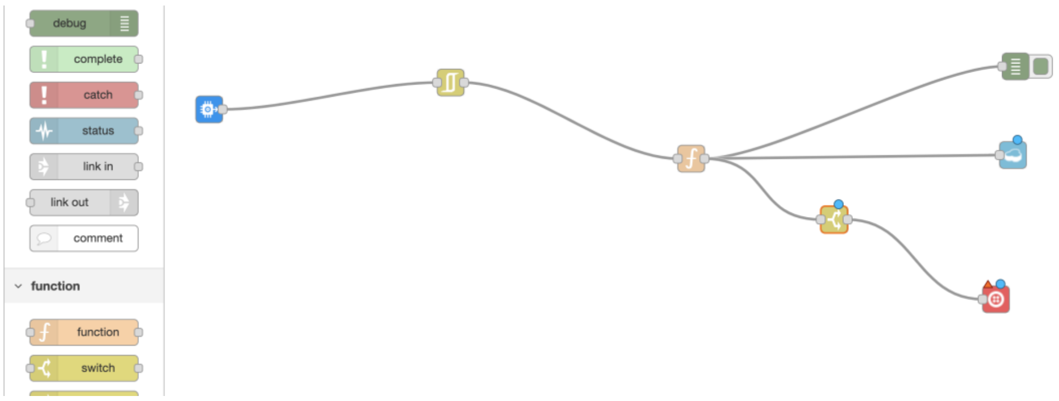
0.1.0 2 years, 7 months ago install

Step 3: Search for ***twilio*** in input nodes, drag and drop the ***twilio** out* node and connect like below.

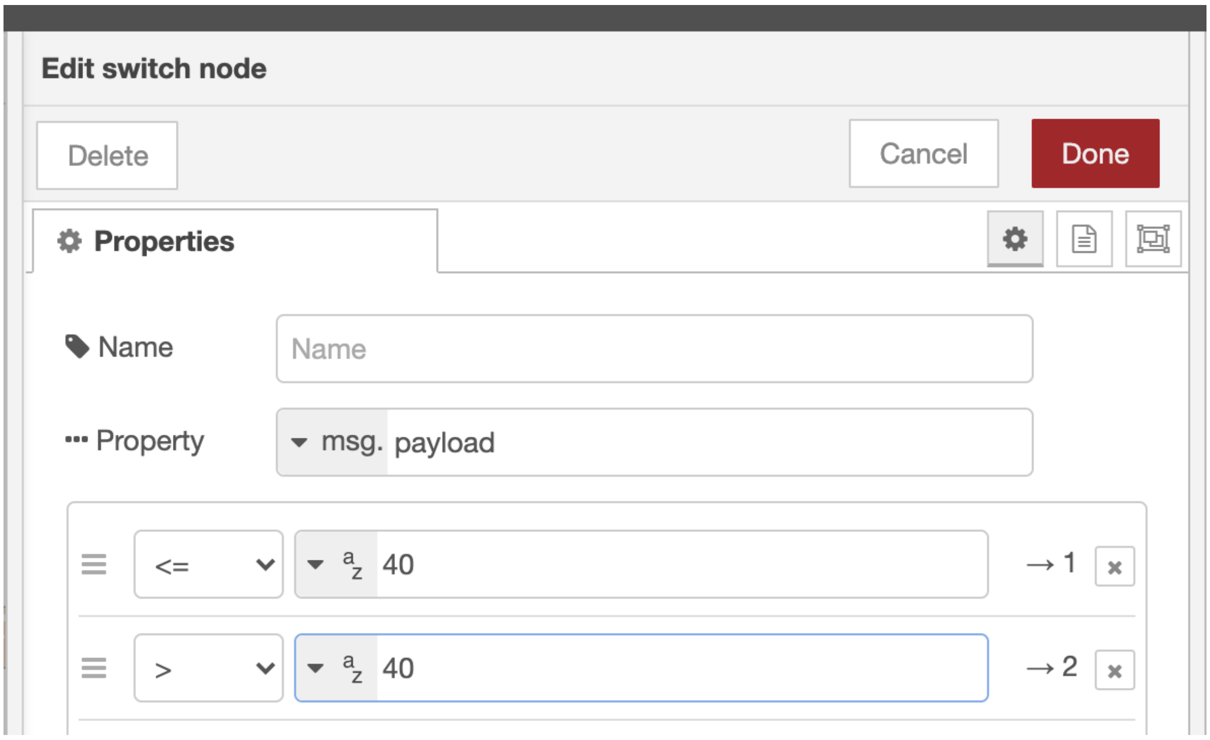


```
30/08/2020, 20:15:01   node: 74113a8e.d0d454
iot-2/type/sensor/id/1234/evt/iotsensor/fmt/json :
msg.payload : number
17
30/08/2020, 20:41:49   node: 74113a8e.d0d454
iot-2/type/sensor/id/1234/evt/iotsensor/fmt/json :
msg.payload : number
17
```

Step 4: Drag and drop ***switch*** node like below



Step 5: click on ***switch*** node and build 2 rules like below. To create second rule, click on + add icon


Edit switch node

Name: Name

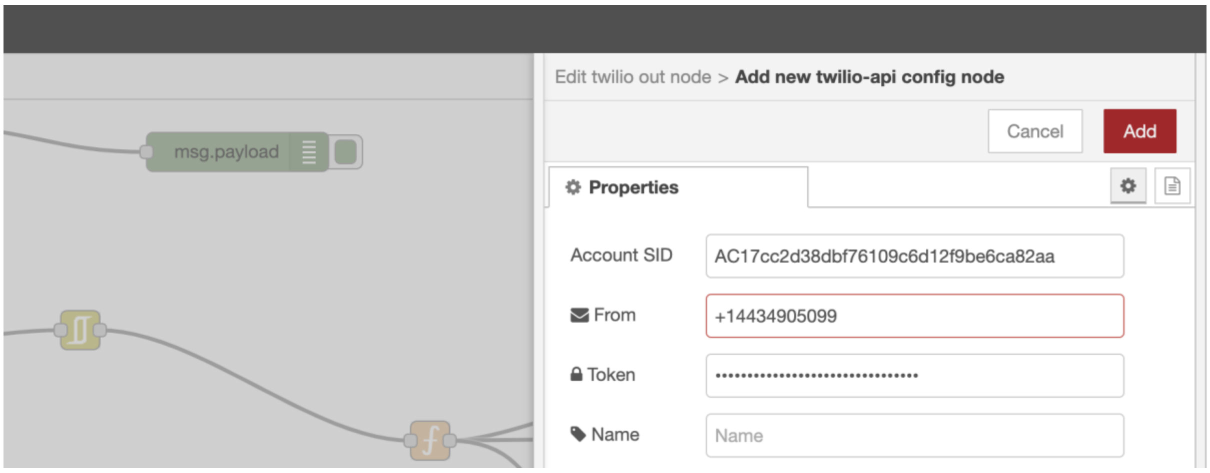
Property: msg. payload

<= az 40 → 1

> az 40 → 2

Step 6: Double click on **twilio out** node and click on ✎ and add ***Account SID, From*** and ***Token*** and click on ***Add***

Note: All these 3 values have taken from twilio credentials which we got after creating twilio account


Edit twilio out node > Add new twilio-api config node

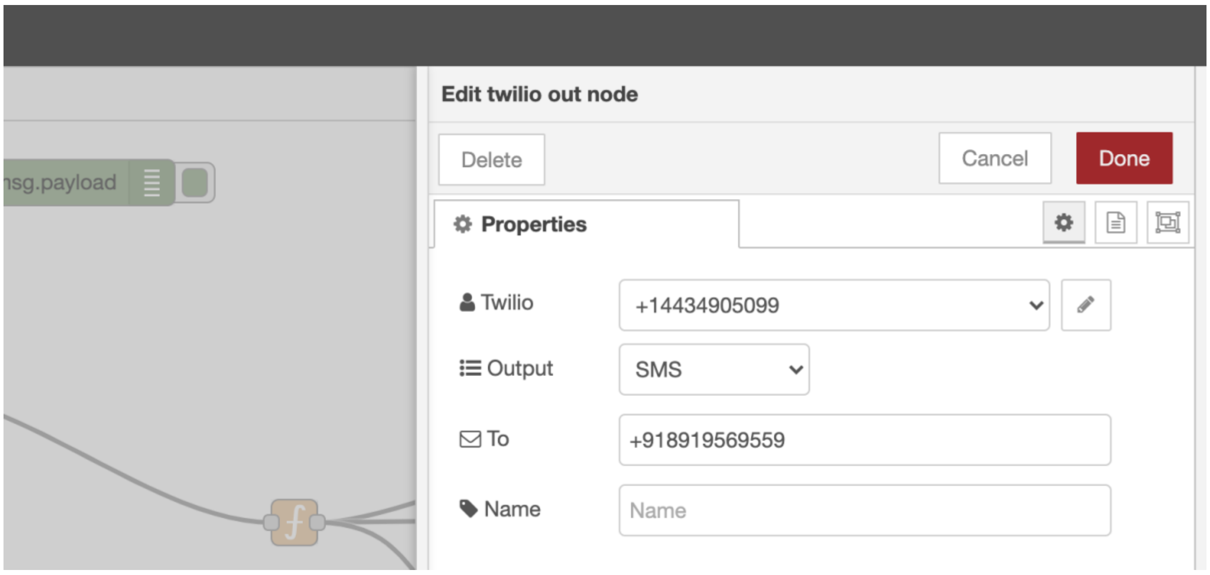
Account SID: AC17cc2d38dbf76109c6d12f9be6ca82aa

From: +14434905099

Token: ••••••••••••••••••••••••••••••••

Name: Name

Step 7: Enter your mobile number in **To** field and click on ***Done***


Edit twilio out node

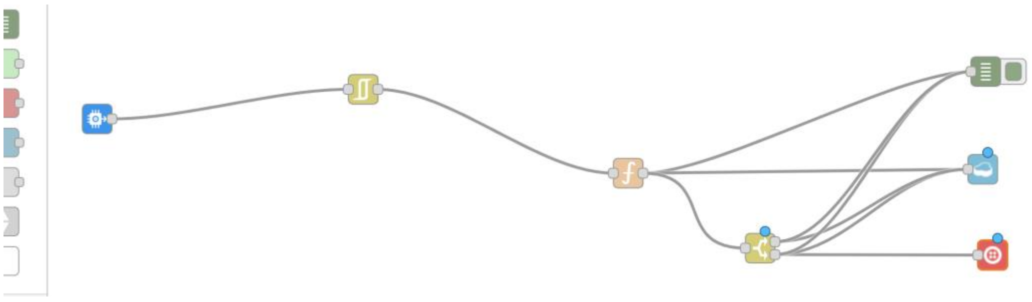
Twilio: +14434905099

Output: SMS

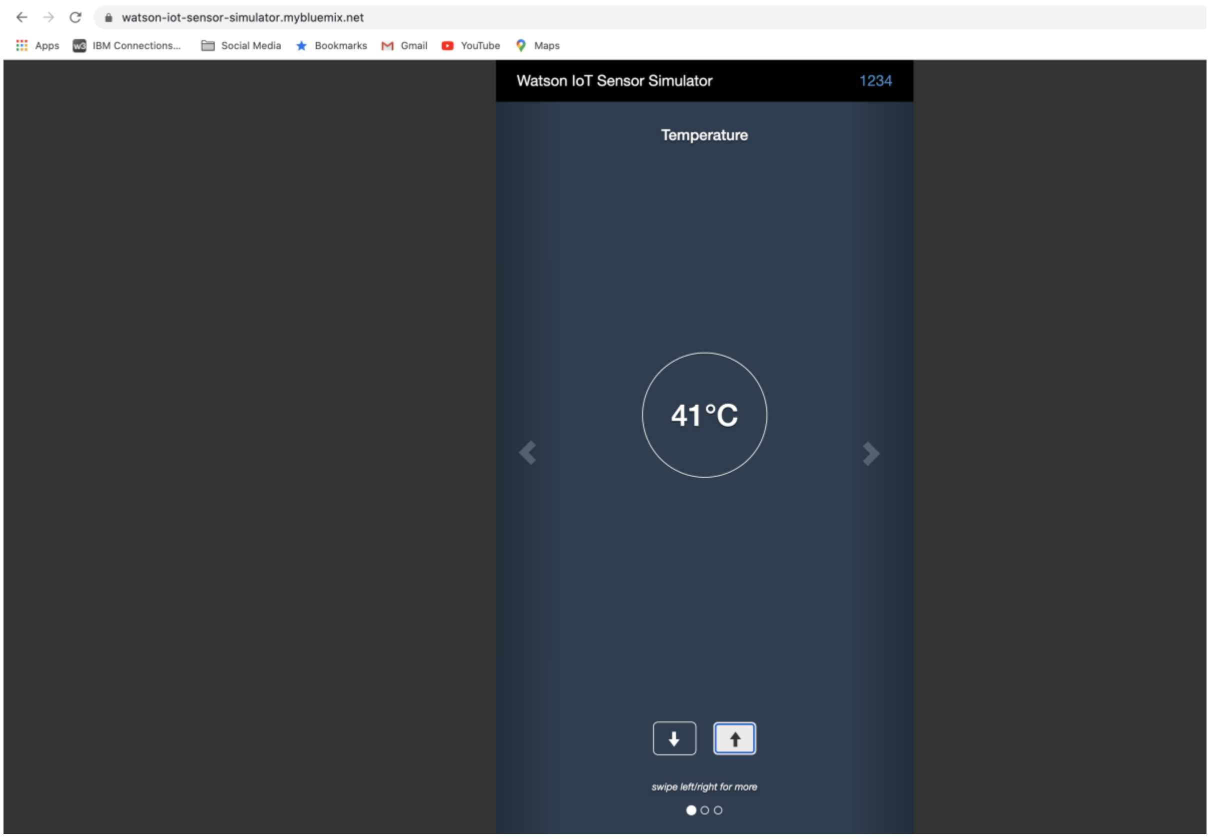
To: +918919569559

Name: Name

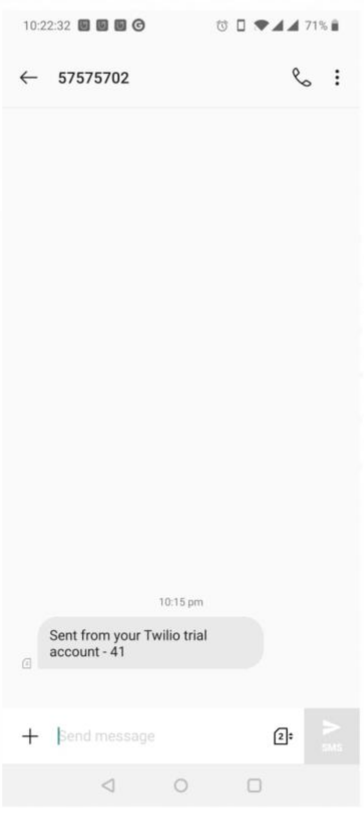
Step 8: Connect ***switch*** node both outputs to ***Debug*** node and ***Cloudant*** node and connect ***Twilio*** node to only Rule 2 like below



Step 9: Change ***Watson IoT Sensor simulator*** value to **41**


watson-iot-sensor-simulator.mybluemix.net
Apps
IBM Connections...
Social Media
Bookmarks
Gmail
YouTube
Maps
Watson IoT Sensor Simulator
1234
Temperature
41°C
swipe left/right for more

Step 10: Now, you should be able to see a push notification (message) sent to your registered mobile



Sent from your Twilio trial account - 41