

Web Application Development With Node-RED Service

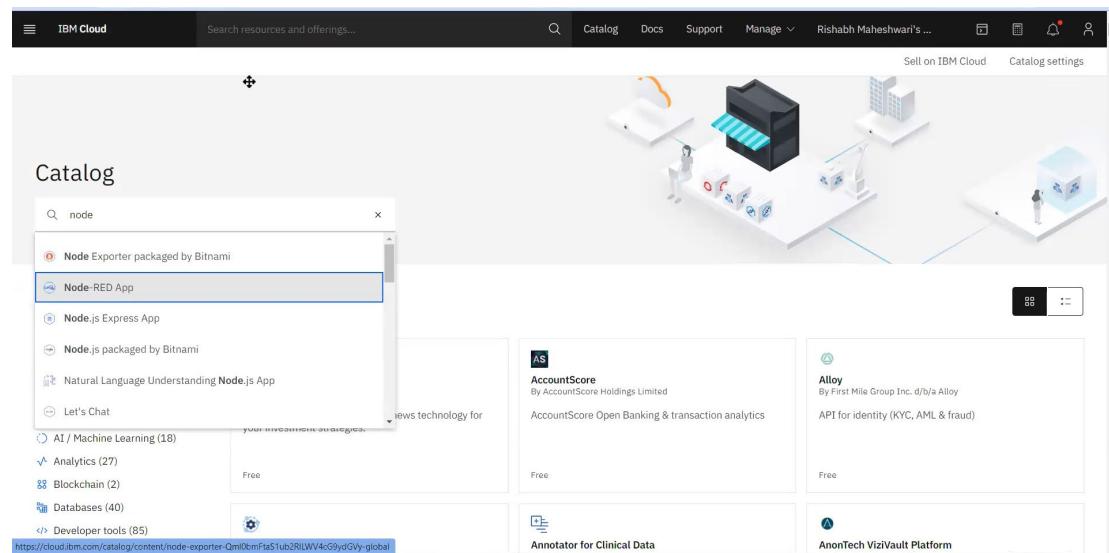
Rishabh Maheshwari

19BCY10145

Introduction to Node-RED and its features. Install the required nodes and configure the nodes to get the sensor data from the IBM IoT platform. Develop a Web UI to display the sensor parameters and configure the buttons for sending commands to the IBM IoT platform. configure the Node-RED to send message notifications.

Procedure to Create Node-RED service on IBM Cloud:

- Open IBM Cloud Service and Search Node-RED in Catalog.



- Create a Node-RED service by Cloudant. Click on **Get Started**.

The screenshot shows the IBM Cloud interface for the Node-RED service. At the top, there's a navigation bar with 'IBM Cloud', a search bar, and various links like Catalog, Docs, Support, Manage, and a user profile. Below the header, the service name 'Node-RED' is displayed with a gear icon. The main content area has two tabs: 'About' (selected) and 'Create'. Under 'About', there's a 'Details' section with information about the author (IBM), updated date (2/11/2020), and type (Starter kit). It also links to 'Source code' on GitHub, 'Helpful links', 'Terms', and 'Tutorial'. The 'Overview' section describes the starter kit as providing a pre-configured Node-RED application with a Cloudant service. It includes a bulleted list: 'Generate an application with Node-RED', 'Generate an application with files for deploying to Cloud Foundry or a DevOps Pipeline', and 'Connect to provisioned services'. A 'What's included?' section lists 'Cloudant' with a 'Get started' button. On the right side, there's a vertical sidebar with an 'ASK A QUESTION' button.

- Change Location to London region. Then click on **Create**.

The screenshot shows the 'Create' form for the Node-RED service. The top part is identical to the previous screenshot, showing the IBM Cloud header and the 'Node-RED' service page. The main form has sections for 'Resource group' (set to 'Default'), 'Tags' (empty), and 'Platform' (set to 'Node.js'). Below this is a 'Service details' section for 'Cloudant'. It includes fields for 'Region' (set to 'London') and 'Resource group' (set to 'Default'). There's also a 'Pricing plan' dropdown set to 'Lite'. At the bottom of the form are 'Pricing details' and 'Terms' links, followed by 'Cancel' and 'Create' buttons. The right side of the screen features a vertical sidebar with an 'ASK A QUESTION' button.

- After that click on Deploy your App to create pipelines and networks.

The screenshot shows the IBM Cloud interface for an application named "Node RED IKZIW 2021-12-02". The "Details" section displays the following information:

- App URL:** You must deploy your app first
- Source:** Download code
- Resource group:** Default
- Deployment target:** You must deploy your app first
- Created:** 12/2/2021

The "Services" section lists "Cloudant" and provides links to "Open dashboard", "Documentation", and "API reference". Below this are buttons for "Connect existing services" and "Create service".

The "Deployment Automation" section contains a "Getting started quickly" guide titled "Configuring your app". It includes steps for connecting services, viewing code, deploying the app, and managing the deployment pipeline. Step 1: Use the Services card to connect a service to your app. Step 2: If you want to view the code before your app is deployed, click Download code to obtain the .zip file. Step 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. Step 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. Step 5: If you make any changes to your app, be sure to deploy it again.

- Generate a new API key available in the next window. Change the region to London. Click on NEXT.

The screenshot shows the "Deploy your app" configuration window. The "IBM Cloud API key" field contains a masked API key. The "Number of instances" is set to 1. Under "Memory allocation per instance", the slider is at 64 MB, with 2000 MB and 256 MB as options. The "Region" is set to London. The "Organization" is rishabh.maheshwari2019@vitbhc, and the "Space" is dev. The "Host" is node-red-ikziw-2021-12-02 and the "Domain" is eu-gb.mybluemix.net. At the bottom are "Cancel" and "Next" buttons. The status bar at the bottom right shows system information: 25°C Rain showers, ENG IN, 6:23 PM, 12/2/2021.

- Now in the IBM Cloud Dashboard new resources should now be available.

The screenshot shows the IBM Cloud Dashboard with several service cards displayed:

- Explore IBM Cloud with this selection of easy starter tutorials and services.**
- Smart Document Understanding Demo**: Check out this Smart Document Understanding demo to learn how this powerful feature allows you to visually apply structure to your documents.
- IBM Watson Machine Learning**: Deploy, monitor and optimize machine learning models quickly and easily. Leverage auto-generated APIs to infuse AI into your applications.
- Create a custom dashboard**: Create a shareable dashboard that you can customize with widgets, scope, and your own layout.
- Use Watson Studio**: Watson Studio provides a suite of tools and a collaborative environment for data scientists, developers and domain experts.
- Learn about API Keys**: Create API Keys to identify the application or user calling an API. Most IBM Cloud services require API Keys when calling their APIs.

Below the cards, there's a **Resource summary** section showing 6 resources across categories: Cloud Foundry services, Services and software, Apps, and Developer tools. There's also a **Planned maintenance** section indicating "Clear skies! You can view your scheduled maintenance events here." and an **IBM Cloud status** section showing a world map with "No issues".

- For opening Node-RED the app URL will be available once we open the app from IBM Clouds' resource list.

The screenshot shows the IBM Cloud App details page for the app "Node RED IKZIW 2021-12-02".

Details section:

- App URL: <https://node-red-ikziw-2021-12-02.eu-gb.mybluemix.net>
- Source: <https://eu-gb.git.cloud.ibm.com/rishabh.maheshwari2019/Node-RED>
- Resource group: Default
- Deployment target: Node RED IKZIW 2021-12-02
- Created: 12/2/2021

Services section:

- Cloudant
 - [Open dashboard](#)
 - [Documentation](#)
 - [API reference](#)
- Credentials

Deployment Automation section:

- Name: NodeREDIKZIW2021-12-02
- Location: London
- Tool integrations

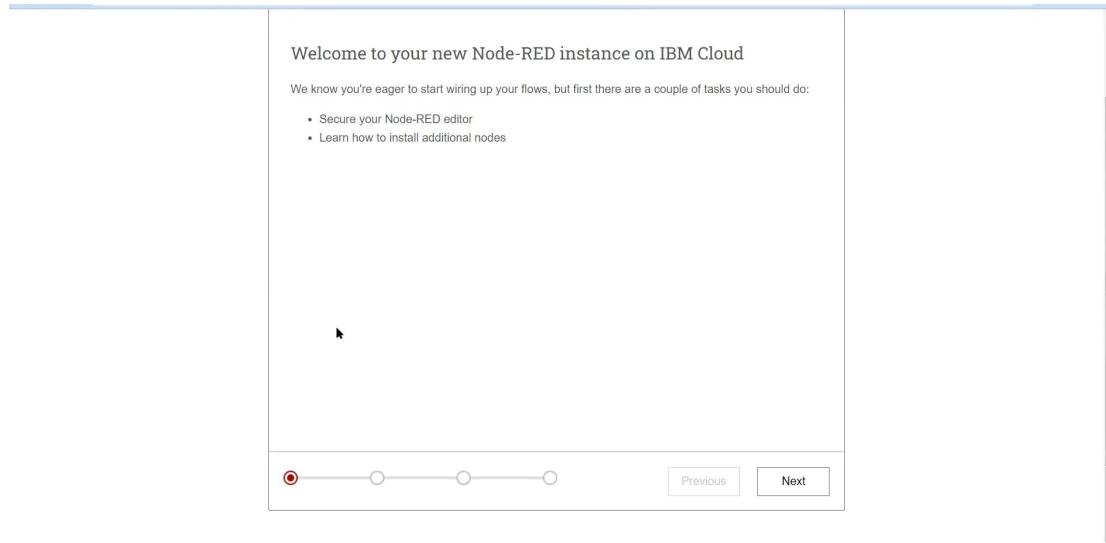
Delivery Pipelines section:

- Name: pr-pipeline
- Status: No stages detected
- Name: ci-pipeline
- Status: Success

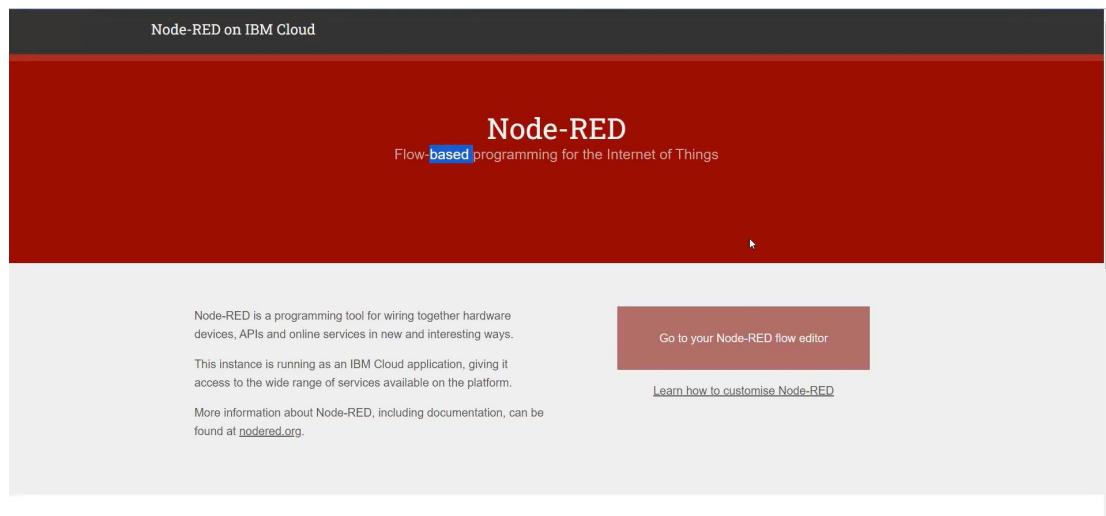
Getting started quickly section:

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

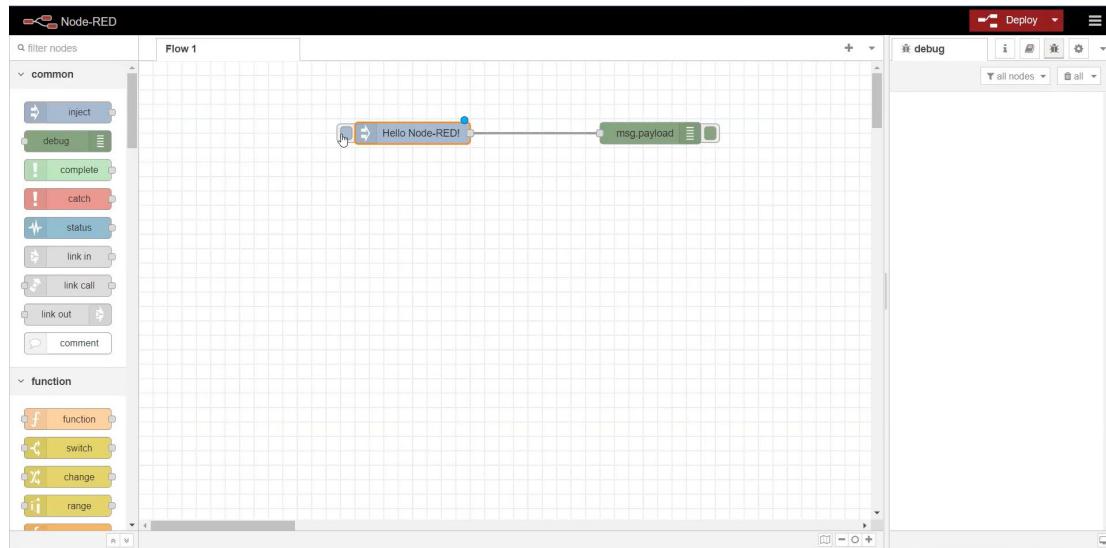
- This window shown below will be visible the very first time you open the Node-RED app. Click **Next** three times and then click on Finish.



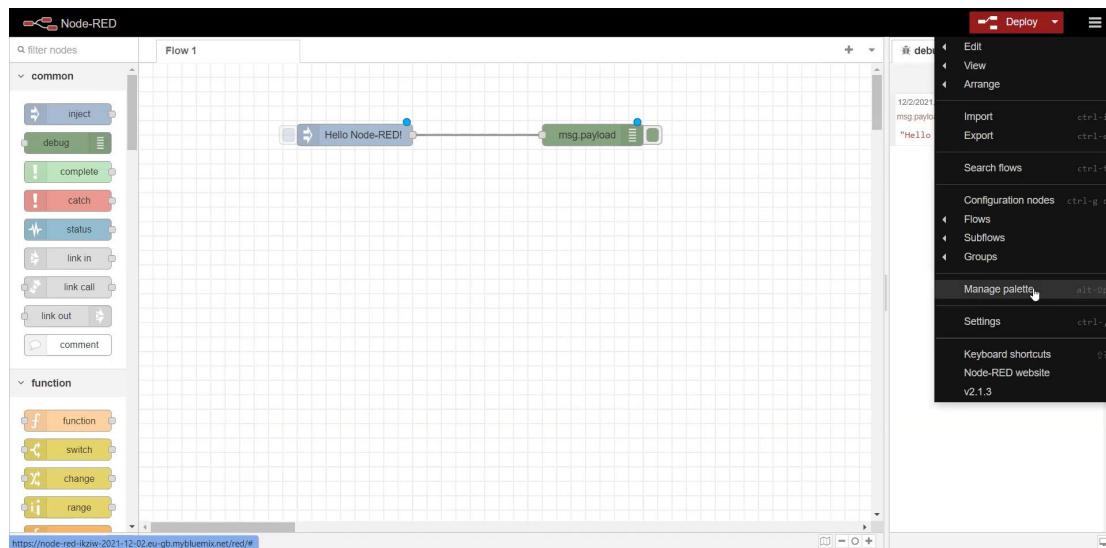
- Click on the Go to your Node-RED flow editor icon to open the flow window.



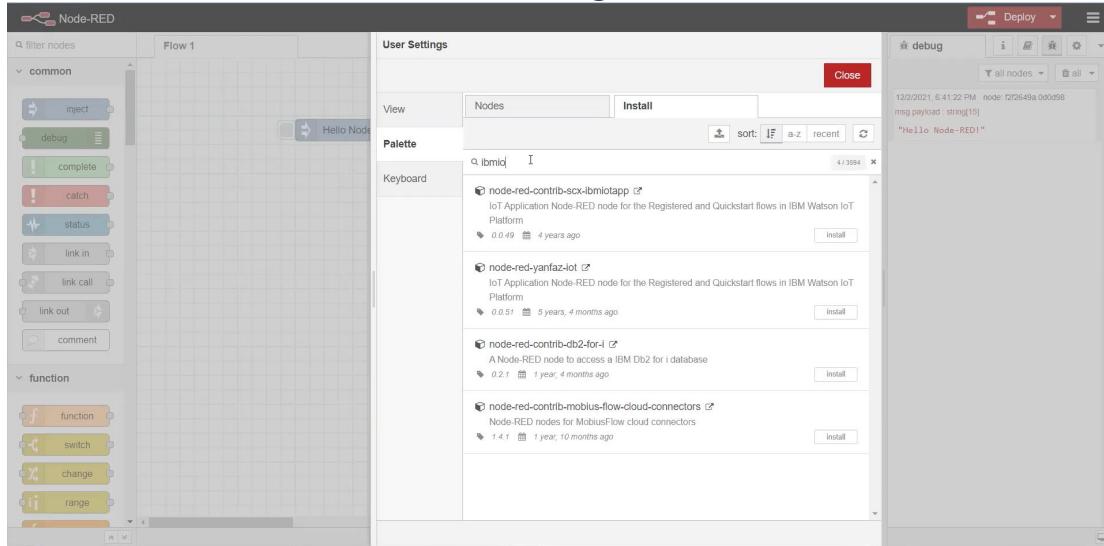
- For Developing the Web UI and displaying the sensor parameters and configure the buttons for sending commands to the IBM IoT platform follow the steps shown below-



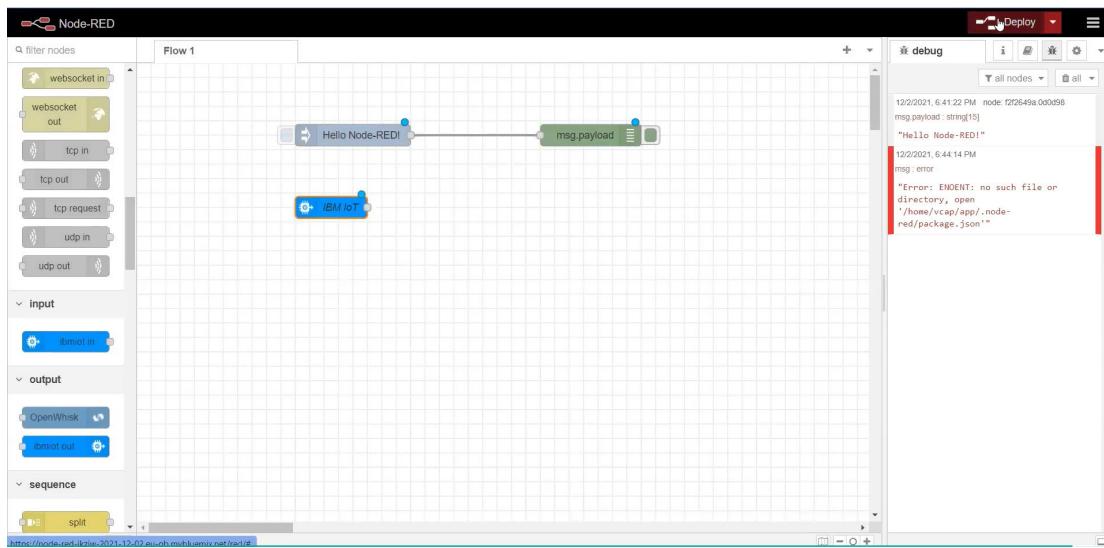
- For installing the repositories and get Pallets open Manage Pallets.



- Search for **ibmiot** in the Install section and install the first option visible in the image.



- Now follow the steps shown in sequence to proceed.



Node-RED Flow 1 (Top):

```

graph LR
    In1(( )) --> WebS1[websocket in]
    WebS1 --> Hello1[Hello Node-RED!]
    Hello1 --> IoT1[IBM IoT]
    IoT1 --> Out1[OpenWhisk]
    IoT1 --> Out2[directed out]

```

Edit ibmiot in node (Top):

Properties:

- Authentication: Quickstart
- Input Type: API Key (selected)
- Device Id: device id e.g. ab12cd231a21
- Name: IBM IoT
- Service: quickstart

Quickstart: Use the Input Type property to configure this node to receive Events sent by 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.

Debug (Top):

```

12/2/2021, 6:41:22 PM node: f22649a.000d98
msg.payload: string[15]
"Hello Node-RED!"

12/2/2021, 6:44:14 PM
msg.error
"Error: ENOENT: no such file or directory, open '/home/vcap/app/.node-red/package.json'"

```

Node-RED Flow 2 (Middle):

```

graph LR
    In2(( )) --> WebS2[websocket in]
    WebS2 --> Hello2[Hello Node-RED!]
    Hello2 --> IoT2[IBM IoT]
    IoT2 --> Out3[OpenWhisk]
    IoT2 --> Out4[directed out]

```

Edit ibmiot in node (Middle):

Properties:

- Authentication: API Key
- API Key: Add new ibmiot... (dropdown)
- Input Type: Device Event
- Device Type: All or +
- Device Id: device id e.g. ab12cd231a21
- Event: All or +
- Format: json
- QoS: 0
- Name: IBM IoT
- Service: registered

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.

Debug (Middle):

```

12/2/2021, 6:41:22 PM node: f22649a.000d98
msg.payload: string[15]
"Hello Node-RED!"

12/2/2021, 6:44:14 PM
msg.error
"Error: ENOENT: no such file or directory, open '/home/vcap/app/.node-red/package.json'"

```

Node-RED Flow 3 (Bottom):

```

graph LR
    In3(( )) --> WebS3[websocket in]
    WebS3 --> Hello3[Hello Node-RED!]
    Hello3 --> IoT3[IBM IoT]
    IoT3 --> Out5[OpenWhisk]
    IoT3 --> Out6[directed out]

```

Edit ibmiot in node (Bottom):

Properties:

- Authentication: API Key
- API Key: IBM IOT API (dropdown)
- Input Type: Device Event
- Device Type: All or device
- Device Id: 01 (dropdown)
- Event: All or +
- Format: json
- QoS: 0
- Name: IBM IoT
- Service: registered

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.

Debug (Bottom):

```

12/2/2021, 6:41:22 PM node: f22649a.000d98
msg.payload: string[15]
"Hello Node-RED!"

12/2/2021, 6:44:14 PM
msg.error
"Error: ENOENT: no such file or directory, open '/home/vcap/app/.node-red/package.json'"

```

IBM Watson IoT Platform

Browse Action Device Types Interfaces Add Device +

Criteria: To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator

Device ID Status Device Type Class ID Date Added Descriptive Location

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
01	Disconnected	device	Device	Dec 1, 2021 7:12 PM	

Identity Device Information Recent Events State Logs

Device ID: 01
Device Type: device
Date Added: Dec 1, 2021 7:12 PM
Added By: rishabh.maheshwari2019@vitbhopal.ac.in
Connection Status: Disconnected

Items per page: 50 | 1–1 of 1 item

1 of 1 page < 1 >

IBM Watson IoT Platform

Boards Devices Members Apps Diagnose Add Device +

Manage Devices

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

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
01	Disconnected	device	Device	Dec 1, 2021 7:12 PM	

50 | 1–1 of 1 item

1 of 1 page < 1 >

IBM Watson IoT Platform

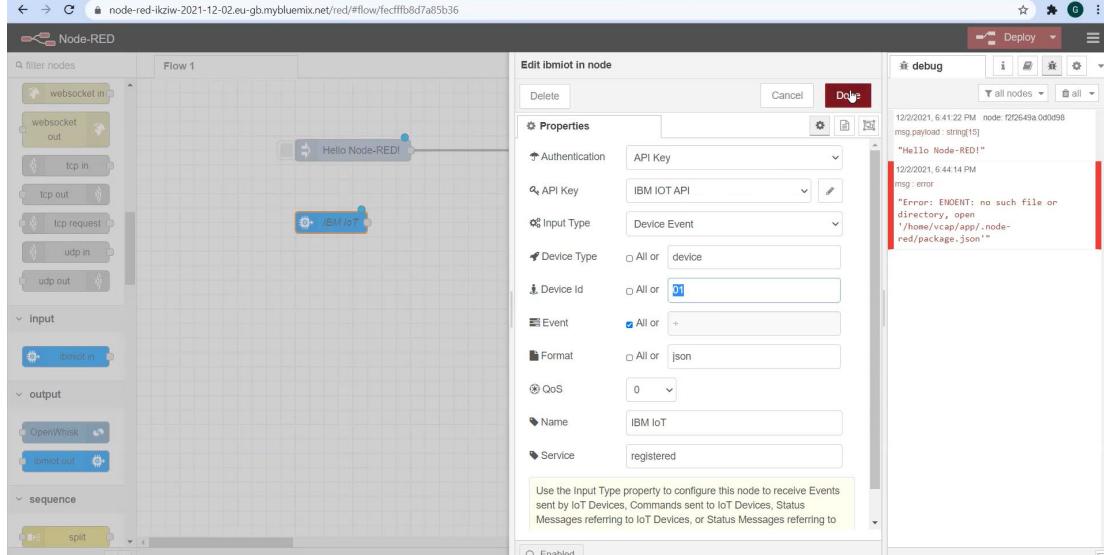
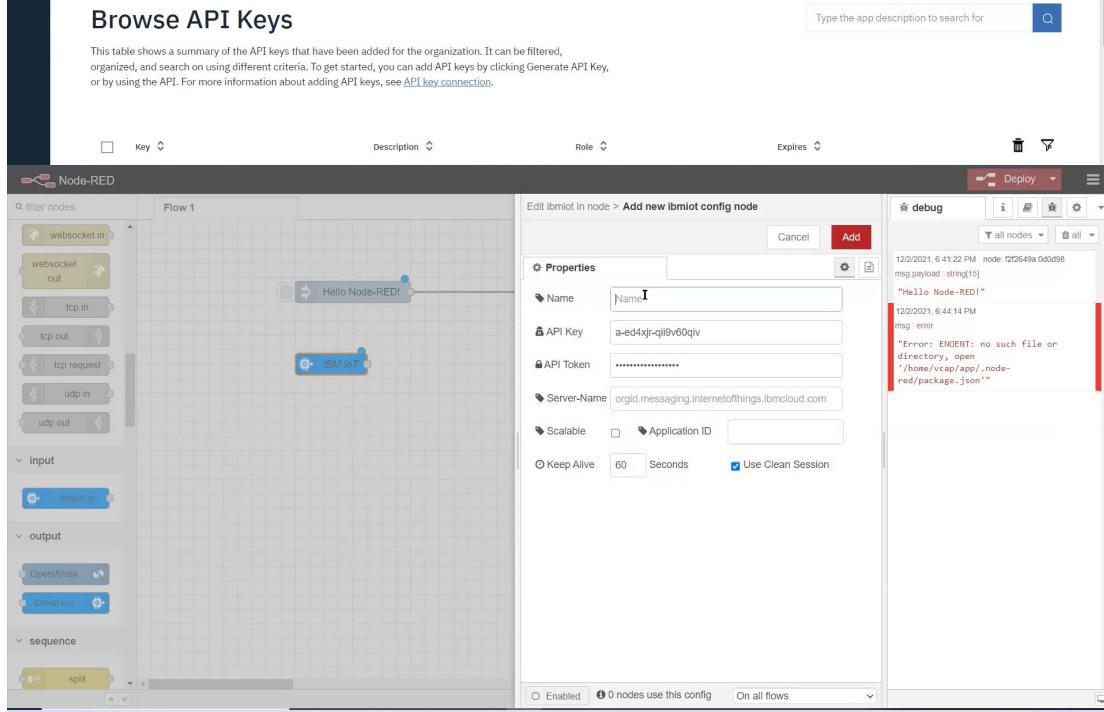
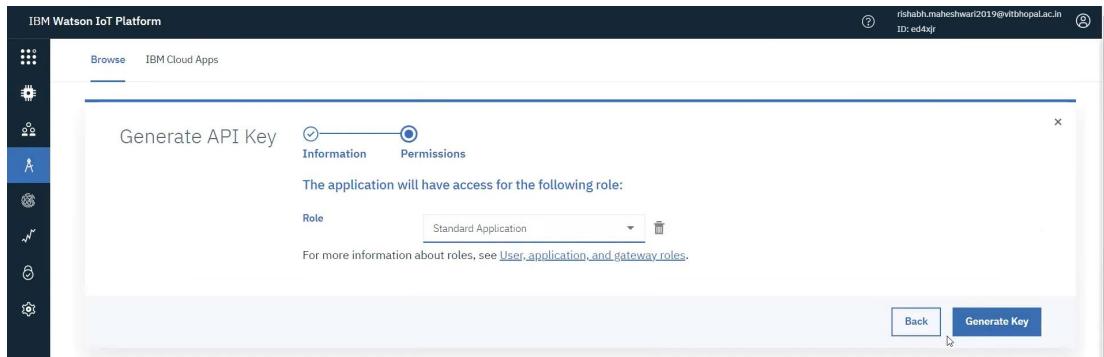
Browse IBM Cloud Apps Generate API Key

Type the app description to search for

Browse API Keys

This table shows a summary of the API keys that have been added for the organization. It can be filtered, organized, and search on using different criteria. To get started, you can add API keys by clicking Generate API Key, or by using the API. For more information about adding API keys, see [API key connection](#).

Key	Description	Role	Expires	
a-ed4xjr-wx0pdrgmgn	API Key for the device simulator	Standard Application	-	



IBM Watson IoT Platform

Device Type: device

Events 1

Event type name: event_1

Schedule: Every Minute

Payload:

```

0 {
1   "temperature": random(0, 100),
2   "humidity": random(0, 100),
3 }
4

```

Upload a CSV file

Cancel Save

Type here to search

Node-RED

Flow 1

common nodes:

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function nodes:

- function
- switch
- change
- range

Nodes in Flow 1:

```

graph LR
    inject[inject] --> hello[Hello Node-RED!]
    hello --> payload1[msg.payload]
    hello --> iot[IBM IoT]
    iot --> payload2[msg.payload]

```

Debug Log:

- 12/2/2021, 6:41:22 PM node: 022649a.000d98 msg payload: string(1) "Hello Node-RED!"
- 12/2/2021, 6:44:14 PM msg: error "Error: ENOENT: no such file or directory, open '/home/vcap/app/.node-red/package.json'"
- 12/2/2021, 6:58:17 PM node: 1072331d531af792 id: 2/type/deviceId/01/lev/lev/event_1/fm/json msg payload: Object
- temperature: 65, humidity: 27

IBM Watson IoT Platform

Browse Action Device Types Interfaces

Device ID: 01 Status: Disconnected Device Type: device Class ID: Device Date: 2021-12-02 11:41:22

Search by Device ID: 01

Device Type: device

Event type name: event_1

Schedule: Every Minute

Payload:

```

0 {
1   "temperature": random(0, 100),
2   "humidity" : random(0,100)
3 }
4

```

Upload a CSV file

Cancel Save

Items per page: 50 | 1-1 of 1 item

Node-RED

Flow 1

common nodes:

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function nodes:

- function
- switch
- change
- range

Flow 1:

```

graph LR
    inject[inject] --> Hello[Hello Node-RED!]
    Hello --> msg_payload1[msg.payload]
    msg_payload1 --> IBM[IBM IoT]
    IBM --> msg_payload2[msg.payload]

```

msg.payload:

```

12/2/2021, 6:41:22 PM node: [22649a:000098]
msg.payload: string[1]
"Hello Node-RED!"
```

msg.error:

```

12/2/2021, 6:44:14 PM
msg.error: Error: ENOENT: no such file or
directory, open
'/home/vcap/app/.node-
red/package.json'
```

msg.payload:

```

12/2/2021, 6:58:17 PM node: f072331d531af792
iot-2/type/device/0/0/1/eve/event_1/m/json
msg.payload: Object
```

Object:

```

temperature: 65
humidity: 27
```

msg.payload:

```

12/2/2021, 6:58:24 PM node: f072331d531af792
iot-2/type/device/0/0/1/eve/event_1/m/json
msg.payload: Object
```

Object:

```

{ temperature: 17, humidity: 3 }
```

msg.payload:

```

12/2/2021, 6:58:28 PM node: f072331d531af792
iot-2/type/device/0/0/1/eve/event_1/m/json
msg.payload: Object
```

Object:

```

{ temperature: 13, humidity: 69 }
```