

Create Node Red service

Team ID	PNT2022TMID05128
Project Name	Smart waste management system for metropolitan cities

Step 1: Login into IBM CLOUD account

Step2: In catalog, search for node red application

The screenshot shows the IBM Cloud Catalog interface. At the top, there's a navigation bar with 'IBM Cloud', a search bar, and links for 'Catalog', 'Manage', and 'ABARNA S's Account'. Below the navigation bar, the 'Node-RED' application is selected. The left sidebar contains links for 'About', 'Create', 'Details', 'Source code', 'Helpful links', 'Terms', and 'Tutorial'. The main content area displays the 'Overview' section, which includes a description of the starter kit, a list of features, and a 'What's included?' section. A 'Get started' button is visible at the bottom left of the main content area. The Windows taskbar is visible at the bottom of the screen.

IBM Cloud

Search resources and products...

Catalog Manage ABARNA S's Account

Catalog / Create app /

Node-RED

About Create

Details

Author IBM

Updated 11/2/2020

Type Starter kit

Source code

GitHub

Helpful links

Terms

Tutorial

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?

Cloudant

Free to start View pricing

View docs View API reference

Get started

ASK A QUESTION

Step 3: Enter the project details and click on create

Step 4: click on deploy option and deploy

The screenshot displays the IBM Cloud console interface for a specific application. The top navigation bar includes the IBM Cloud logo, a search bar, and user account information. The main content area is titled 'Node RED DXQJC 2022-11-02' and includes an 'Actions...' dropdown menu.

Details

App URL	You must deploy your app first
Source	Download code
Resource group	Default
Deployment target	You must deploy your app first
Created	11/2/2022

Services

Cloudant

[Open dashboard](#) [Documentation](#) [API reference](#)

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.

[Deploy your app](#)

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

Step 5: Set up the environment for deploying and click on create

IBM Cloud

Search resources and products...


Catalog


Manage


Akshaya M's Account


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

**Kubernetes Service**
IBM
Deploy, scale, and manage your containerized application workloads to highly available clusters.

**Red Hat OpenShift**
IBM
Deploy your apps on highly available clusters that come installed with Red Hat OpenShift on IBM Cloud.

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

**Code Engine**
IBM
Run your app, job, or container on a managed serverless platform. Auto-scale workloads, and pay only for the resources that you consume.

IBM Cloud API key

.....

New +

Container registry region

Dallas

Container namespace

jbmfyhfv...nfcumphsw

Cluster region

Frankfurt

Cluster resource group

Default

Cluster namespace

default

Cluster name

mycluster-free

Deployment type

Step 1. Select the deployment target

Select your deployment target, and then provide the configuration information.

IBM Cloud Kubernetes Service

Kubernetes is an open source platform for managing containerized workloads and services across multiple hosts, and offers management tools for deploying, automating, monitoring, and scaling containerized apps with minimal to no manual intervention. [Learn more.](#)

Before you begin


- One free Kubernetes cluster is available per account.
- If you don't have an available cluster, you must create one before continuing. Allow 10-20 minutes for the cluster to be provisioned. [Create cluster.](#)

Steps

- Create an IBM Cloud API key, or select an existing one from a secrets store.
- Select the container registry region.
- Enter the container registry namespace if it is not already completed.
- Select the region where your Kubernetes cluster is located.
- Select the resource group, cluster namespace, and the cluster name.

ASK A QUESTION

26°C Rain off and on

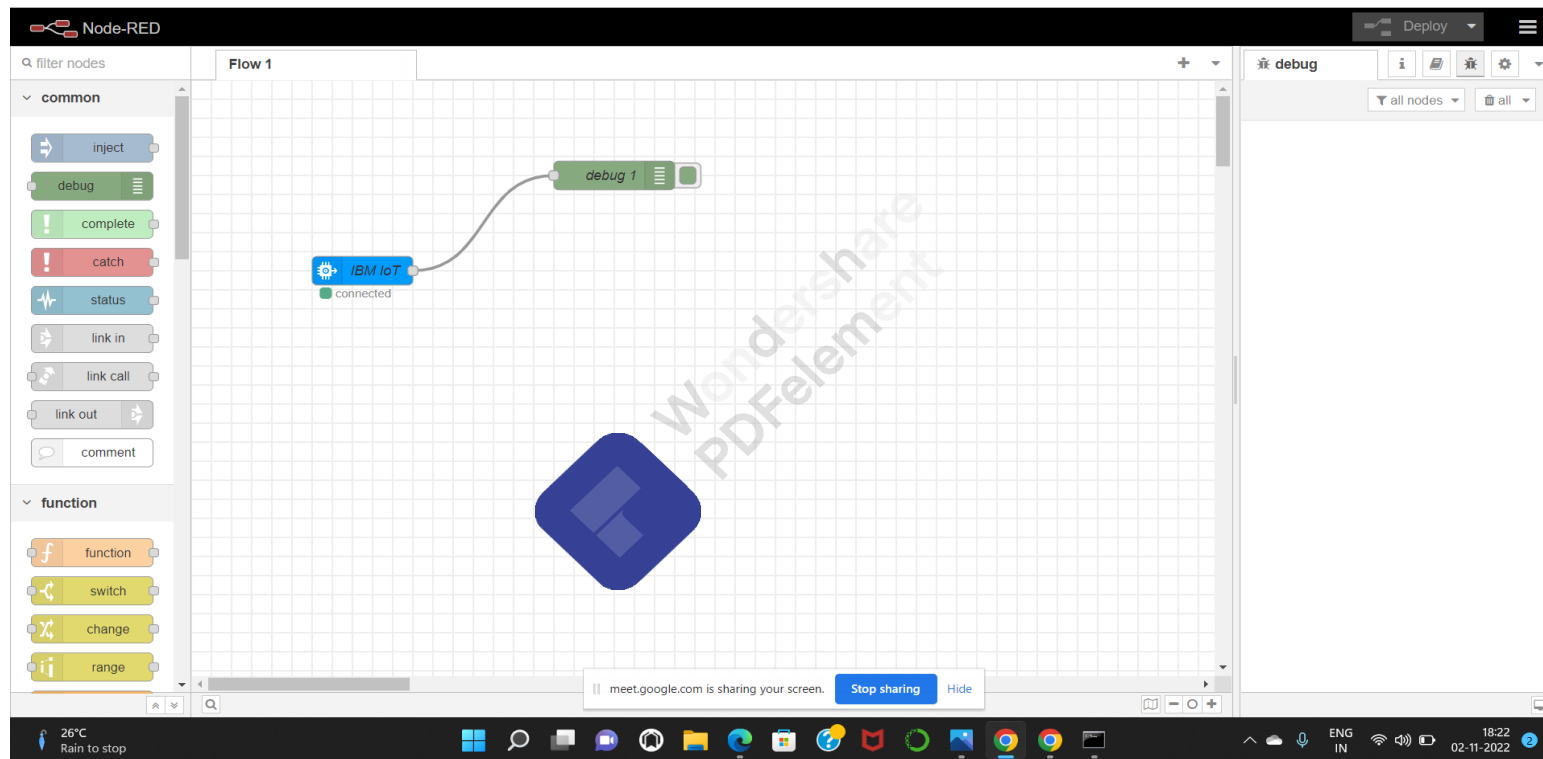


ENG IN

20:47

02-11-2022

Step 6: Now drag and drop the nodes and connect nodes with IOT Watson platform



Step 7: setup the settings that connects node red service with Watson IOT

The screenshot displays the Node-RED web interface. On the left, the 'common' nodes palette is visible, containing nodes like inject, debug, complete, catch, status, link in, link call, link out, and comment. The central workspace shows a flow named 'Flow 1' with an 'IBM IoT' node connected to a 'debug 1' node. The 'IBM IoT' node is marked as 'connected'. On the right, the 'Edit ibmiot in node' configuration panel is open, showing the following settings:

- Authentication:** API Key
- API Key:** Akshaya
- Input Type:** Device Event
- Device Type:** All or Test
- Device Id:** All or Test123
- Event:** All or +
- Format:** All or json
- QoS:** 0
- Name:** IBM IoT
- Service:** registered

A yellow tooltip at the bottom of the configuration panel reads: "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". The bottom status bar shows the system clock as 20:57 on 02-11-2022, along with weather information (26°C, Rain off and on) and various system icons.

Step 8: Finally, output can be seen in node red service

The screenshot shows the Node-RED web interface. On the left, the 'common' nodes palette includes inject, debug, complete, catch, status, link in, link call, link out, and comment. The 'function' nodes palette includes function, switch, change, and range. The main workspace, titled 'Flow 1', contains a flow with two nodes: an 'IBM IoT' node (blue with a gear icon and 'connected' status) and a 'debug 1' node (green with a list icon). A curved line connects the output of the IBM IoT node to the input of the debug 1 node. On the right, the 'debug' console is open, showing a list of messages. Each message is a JSON object with 'temperature' and 'humidity' fields. The messages are as follows:

- 11/2/2022, 8:57:33 PM node: debug 1
iot-2/type/Test/id/Test123/evt/status/fmt/json : msg.payload : Object
↳ { temperature: 103, humidity: 31 }
- 11/2/2022, 8:57:35 PM node: debug 1
iot-2/type/Test/id/Test123/evt/status/fmt/json : msg.payload : Object
↳ { temperature: 96, humidity: 76 }
- 11/2/2022, 8:57:37 PM node: debug 1
iot-2/type/Test/id/Test123/evt/status/fmt/json : msg.payload : Object
↳ { temperature: 56, humidity: 90 }
- 11/2/2022, 8:57:39 PM node: debug 1
iot-2/type/Test/id/Test123/evt/status/fmt/json : msg.payload : Object
↳ { temperature: -4, humidity: 13 }
- 11/2/2022, 8:57:41 PM node: debug 1
iot-2/type/Test/id/Test123/evt/status/fmt/json : msg.payload : Object
↳ { temperature: 3, humidity: 19 }
- 11/2/2022, 8:57:43 PM node: debug 1
iot-2/type/Test/id/Test123/evt/status/fmt/json : msg.payload : Object
↳ { temperature: 50, humidity: 37 }

The bottom of the screen shows a Windows taskbar with various application icons, a system tray with weather (26°C, Rain off and on), language (ENG, IN), and date/time (20:57, 02-11-2022).

```
11 Nov 12:04:32 - [info] Dashboard version 3.1.0 started at /ui
11 Nov 12:04:32 - [info] Settings file : C:\Users\AMORA_IDITH\code-red\settings.js
11 Nov 12:04:32 - [info] Context store : 'default' (local-memory)
11 Nov 12:04:32 - [info] User directory : Users\AMORA_IDITH\code-red
11 Nov 12:04:32 - [warn] Projects disabled - editorFlow.projects.enabled=false
11 Nov 12:04:32 - [info] Flow file : Users\AMORA_IDITH\code-red\flow.json
11 Nov 12:04:32 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
11 Nov 12:04:32 - [info] Server now running at http://127.0.0.1:1880/
11 Nov 12:04:32 - [info] Starting flow
11 Nov 12:04:32 - [info] Started flow
11 Nov 12:07:51 - [info] Stopping flow
11 Nov 12:07:51 - [info] Stopped flow
Terminate batch job (Y/N)? y

C:\Users\AMORA_IDITH>color a
C:\Users\AMORA_IDITH>code-red
11 Nov 12:08:03 - [info]

Welcome to Node-RED

-----
11 Nov 12:08:03 - [info] Node-RED version: v2.0.2
11 Nov 12:08:03 - [info] Node.js version: v14.17.1
11 Nov 12:08:03 - [info] Windows_NT 10.0.19045 x64 IE
11 Nov 12:08:04 - [info] Loading palette nodes
11 Nov 12:08:05 - [info] Dashboard version 3.1.0 started at /ui
11 Nov 12:08:05 - [info] Settings file : C:\Users\AMORA_IDITH\code-red\settings.js
11 Nov 12:08:05 - [info] Context store : 'default' (local-memory)
11 Nov 12:08:05 - [info] User directory : Users\AMORA_IDITH\code-red
11 Nov 12:08:05 - [warn] Projects disabled - editorFlow.projects.enabled=false
11 Nov 12:08:05 - [info] Flow file : Users\AMORA_IDITH\code-red\flow.json
11 Nov 12:08:05 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
11 Nov 12:08:05 - [info] Server now running at http://127.0.0.1:1880/
11 Nov 12:08:05 - [info] Starting flow
11 Nov 12:08:05 - [info] Started flow
```

