

Create Node Red service

Team ID	PNT2022TMID31725
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. The top navigation bar includes the IBM Cloud logo, a search bar, and user account information (Akshaya M's Account). The main content area displays a grid of application cards. The 'Node-RED App' card is highlighted with a blue underline. The left sidebar shows filters for Location (Dallas, Frankfurt, London, Montreal, Osaka, Sao Paulo) and Support (IBM supported, Third party supported). The bottom status bar shows the weather (26°C, Rain off and on) and system icons.

Application Name	By	Description	Starter Kits
Delphix DevOps Data Platform for IBM Cloud	By catalog:filter.ibm_third_party	Deliver terabytes of data in minutes to accelerate application development in IBM Cloud.	Terraform • IBM Cloud Schematics • Third party supported
GeneXus	By GeneXus	Create and evolve apps in the most efficient way: automatically. Agile development tool that generates and maintain everything from databases to code,...	Server Images • IBM Cloud Schematics • Free • Third party supported
Go Gin App	By IBM	Start building your next Go Gin app on IBM Cloud.	Starter kits • IBM Cloud Kubernetes Service • Red Hat OpenShift
Java Liberty App	By IBM	Start building your next Java Liberty app on IBM Cloud.	Starter kits • IBM Cloud Kubernetes Service • Red Hat OpenShift
Java Spring App	By IBM	Start building your next Java Spring app on IBM Cloud.	Starter kits • IBM Cloud Kubernetes Service • Red Hat OpenShift
Node-RED App	By IBM	Start building your next Node-RED app on IBM Cloud.	Starter kits • IBM Cloud Kubernetes Service • Red Hat OpenShift
Node.js Express App	By IBM	Start building your next Node.js Express app on IBM Cloud.	
PAYTESTER	By CLAI PAYMENTS USA LLC	Test any payment system from any channel, and get rid of the complexity of testing multiple channels and transactions.	
Plesk	By Plesk International GmbH	Plesk is the leading WebOps platform to build, secure and run websites, applications and hosting businesses.	

Step 3: Enter the project details and click on create

Step 4: click on deploy option and deploy

The screenshot shows the IBM Cloud Developer console interface. The browser address bar displays the URL: `cloud.ibm.com/developer/appservice/apps/4d9d88fa-dba5-4b56-986c-76b3909fe692`. The page title is "Node RED DXQJC 2022-11-02" with an "Add tags" link. The "Details" section shows the App URL, Source (with a "Download code" button), Resource group (Default), Deployment target, and Created date (11/2/2022). The "Services" section shows the Cloudant service with links to Open dashboard, Documentation, and API reference, and buttons for Connect existing services and Create service. The "Deployment Automation" section shows a "Configure Continuous Delivery" card with a "Deploy your app" button. The "Getting started quickly" sidebar on the right provides a list of steps for configuring the app and deploying it. The bottom of the screen shows the Windows taskbar with various application icons and the system tray displaying the date and time.

Resource list / App details /

Node RED DXQJC 2022-11-02 [Add tags](#)

Actions...

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

ASK A QUESTION

26°C Rain off and on

20:46 02-11-2022

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

The screenshot shows the IBM Cloud Developer console interface. The browser tabs include WhatsApp, IBM-Project-48101-1660804426/, and IBM App Development. The address bar shows the URL: cloud.ibm.com/developer/appservice/apps/4d9d88fa-dba5-4b56-986c-76b3909fe692. The IBM Cloud header is visible with a search bar and navigation links.

Deployment target

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

..... [Key icon] [New +]

Container registry region: Dallas

Container registry namespace: jbmfyhfuvvmqrrymgrbnnfcumhsw

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

20:47 02-11-2022

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

The screenshot displays the Node-RED web interface in a browser window. The browser's address bar shows the URL `127.0.0.1:1880/#flow/57418c723fe9a62f`. The Node-RED interface includes a left sidebar with a 'filter nodes' search bar and two categories of nodes: 'common' and 'function'. The 'common' category contains nodes like 'inject', 'debug', 'complete', 'catch', 'status', 'link in', 'link call', 'link out', and 'comment'. The 'function' category contains 'function', 'switch', 'change', and 'range'. The main workspace, titled 'Flow 1', shows a flow with two nodes: an 'IBM IoT' node (blue with a gear icon and a green 'connected' status indicator) and a 'debug 1' node (green). A curved line connects the output of the 'IBM IoT' node to the input of the 'debug 1' node. On the right side, there is a 'debug' console with a 'Deploy' button and a 'debug' tab. The bottom of the screen shows a Windows taskbar with various application icons, a system tray with weather information (26°C, Rain to stop), and the date/time (18:22, 02-11-2022). A semi-transparent 'Wondershare PDFelement' watermark is visible across the center of the Node-RED workspace.

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

Node-RED interface showing the configuration of the IBM IoT node.

The interface includes a sidebar with node categories (common, function) and a central workspace with a flow diagram. The flow diagram shows an IBM IoT node connected to a debug node.

The right panel displays the configuration for the IBM IoT node, titled "Edit ibmiot in node".

Properties:

- 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

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

127.0.0.1:1880/#editor-tab-properties

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

The screenshot shows the Node-RED web interface in a browser. The top bar indicates the current flow is 'Flow 1'. The left sidebar shows the 'common' and 'function' node categories. The main workspace contains a flow with two nodes: an 'IBM IoT' node (blue) and a 'debug 1' node (green). The 'IBM IoT' node is connected to the 'debug 1' node. The 'debug 1' node is highlighted with an orange border. The right sidebar shows the 'debug' console, which displays a series of messages received from the 'debug 1' node. The messages are JSON objects containing temperature and humidity data.

Flow Diagram:

```

graph LR
    IoT[IBM IoT] --> Debug[debug 1]
  
```

Debug Console Output:

```

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 }
  
```

```
12: Nov 12:34:12 - [info] Dashboard version 3.1.0 started at /ui
12: Nov 12:34:12 - [info] Settings file : C:\Users\VARORA_EDITH\.node-red\settings.js
12: Nov 12:34:12 - [info] Content store : 'default' [module=memory]
12: Nov 12:34:12 - [info] User directory : Users\VARORA_EDITH\.node-red
12: Nov 12:34:12 - [warn] Projects disabled : editorTheme.projects.enabled=false
12: Nov 12:34:12 - [info] Flows file : Users\VARORA_EDITH\.node-red\flows.json
12: Nov 12:34:12 - [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.

12: Nov 12:34:12 - [info] Server now running at http://127.0.0.1:1880/
12: Nov 12:34:12 - [info] Starting flows
12: Nov 12:34:12 - [info] Started flows
12: Nov 12:47:51 - [info] Stopping flows
12: Nov 12:47:51 - [info] Stopped flows
Terminate batch job (Y/N)? y

C:\Users\VARORA_EDITH>cd /a
C:\Users\VARORA_EDITH>node-red
12: Nov 12:48:03 - [info]
Welcome to Node-RED
-----
12: Nov 12:48:03 - [info] Node-RED version: v3.0.2
12: Nov 12:48:03 - [info] Node.js version: v14.17.1
12: Nov 12:48:03 - [info] Windows_NT 10.0.19045 x64 LE
12: Nov 12:48:04 - [info] Loading palette nodes
12: Nov 12:48:05 - [info] Dashboard version 3.1.0 started at /ui
12: Nov 12:48:05 - [info] Settings file : C:\Users\VARORA_EDITH\.node-red\settings.js
12: Nov 12:48:05 - [info] Content store : 'default' [module=memory]
12: Nov 12:48:05 - [info] User directory : Users\VARORA_EDITH\.node-red
12: Nov 12:48:05 - [warn] Projects disabled : editorTheme.projects.enabled=false
12: Nov 12:48:05 - [info] Flows file : Users\VARORA_EDITH\.node-red\flows.json
12: Nov 12:48:05 - [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.

12: Nov 12:48:05 - [info] Server now running at http://127.0.0.1:1880/
12: Nov 12:48:05 - [info] Starting flows
12: Nov 12:48:05 - [info] Started flows
```

Node-RED

filter nodes

common

inject

debug

complete

catch

status

link in

link call

link out

comment

function

switch

change

range

template

delay

Flow 1

IBM IoT

level

show notification

config

all unused

On all flows

ibmiot

SWM1

TestGogul0

ui_base

Node-RED Dashboard

ui_group

[admin] Default1

[unassigned] Default0

ui_tab

admin1

Flow 1

Flow 2

Flow 3