

Create Node Red service

Team ID	PNT2022TMID49399
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 displays the IBM Cloud Catalog website. The browser's address bar shows the URL `cloud.ibm.com/catalog?category=devops`. The page features a dark navigation bar with the IBM Cloud logo, a search bar, and links for 'Catalog', 'Manage', and 'Akshaya M's Account'. Below the navigation bar, there's a search bar and a 'Sell on IBM Cloud' link. The main content area is a grid of application templates. On the left, there are filters for 'Location' (Dallas, Frankfurt, London, Montreal, Osaka, Sao Paulo) and 'Support' (IBM supported, Third party supported). The grid contains the following templates:

- 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. Includes Terraform, IBM Cloud Schematics, and is 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,...
- Go Gin App** (By IBM): Start building your next Go Gin app on IBM Cloud. Includes Starter kits, IBM Cloud Kubernetes Service, and Red Hat OpenShift.
- Java Liberty App** (By IBM): Start building your next Java Liberty app on IBM Cloud. Includes Starter kits, IBM Cloud Kubernetes Service, and Red Hat OpenShift.
- Java Spring App** (By IBM): Start building your next Java Spring app on IBM Cloud. Includes Starter kits, IBM Cloud Kubernetes Service, and Red Hat OpenShift.
- Node-RED App** (By IBM): Start building your next Node-RED app on IBM Cloud. Includes Starter kits, IBM Cloud Kubernetes Service, and 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.

The Windows taskbar at the bottom shows the date as 02-11-2022 and the time as 20:46.

Step 3: Enter the project details and click on create

Step 4: click on deploy option and deploy

The screenshot displays the IBM Cloud Developer console for an application named "Node RED DXQJC 2022-11-02". The browser address bar shows the URL: `cloud.ibm.com/developer/appservice/apps/4d9d88fa-dba5-4b56-986c-76b3909fe692`. The page header includes the IBM Cloud logo, a search bar, and navigation links for "Catalog", "Manage", and the user account "Akshaya M's Account".

Details Section:

- App URL:** You must deploy your app first
- Source:** Includes a "Download code" button with a download icon.
- Resource group:** Default
- Deployment target:** You must deploy your app first
- Created:** 11/2/2022

Services Section:

- Cloudant:** Includes links for "Open dashboard", "Documentation", and "API reference". A "Credentials" dropdown is also present.
- Buttons: "Connect existing services" and "Create service".

Deployment Automation Section:

- Configure Continuous Delivery:** A card explaining that Continuous Delivery is not enabled for this app and providing instructions on how to enable it.
- A prominent blue button labeled "Deploy your app" with a cloud icon.

Getting started quickly Section:

- Configuring your app:** A section with a numbered list of steps to connect services and DevOps toolchains to the app.

Bottom Status Bar:

- Weather: 26°C, Rain off and on.
- System icons: Windows taskbar icons for various applications.
- Language: ENG IN.
- Time: 20:46.
- Date: 02-11-2022.

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

The screenshot displays the IBM Cloud Developer console interface. At the top, there are browser tabs for 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 main header includes the IBM Cloud logo, a search bar, and navigation links for Catalog, Manage, and Akshaya M's Account. The main content area is titled "Select your deployment target and configure your DevOps toolchain. After you click **Create**, the toolchain is created, and the deployment process is started automatically." Below this, there are four deployment target options: Kubernetes Service (IBM), Red Hat OpenShift (IBM), Cloud Foundry (IBM), and Code Engine (IBM). Each option includes a brief description. To the right, a sidebar titled "Step 1. Select the deployment target" provides additional information about IBM Cloud Kubernetes Service, including a description and a list of steps to follow. The main content area also includes fields for "IBM Cloud API key", "Container registry region" (Dallas), "Container registry namespace" (jbmfyhfuvvmymqrrymgrbnncumphsw), "Cluster region" (Frankfurt), "Cluster resource group" (Default), "Cluster namespace" (default), and "Cluster name" (mycluster-free). At the bottom, there is a "Deployment type" section and a system tray showing weather, taskbar, and system clock.

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

Container registry region: Dallas

Container registry namespace: jbmfyhfuvvmymqrrymgrbnncumphsw

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.

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. 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' (inject, debug, complete, catch, status, link in, link call, link out, comment) and 'function' (function, switch, change, range). The main workspace, titled 'Flow 1', contains a flow with two nodes: an 'IBM IoT' node (blue with a gear icon and a '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, a 'debug' console is open, showing 'all nodes' and a 'debug' button. At the bottom of the screen, a Windows taskbar is visible with various application icons, a system tray showing '26°C Rain to stop', and a clock indicating '18:22 02-11-2022'. A notification from 'meet.google.com' is also present at the bottom center.

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

The screenshot shows the Node-RED web interface in a browser. The address bar displays the URL `127.0.0.1:1880/#flow/57418c723fe9a62f`. The interface includes a left sidebar with node categories (common, function), a central workspace with a flow containing an 'IBM IoT' node and a 'debug 1' node, and a right sidebar with the 'Edit ibmiot in node' configuration panel.

The 'Edit ibmiot in node' panel has 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 in a browser. The top bar includes tabs for '2) WhatsApp', 'IBM-Project-48101-1660804426', 'IBM App Development', and 'Node-RED'. The address bar shows the URL '127.0.0.1:1880/#flow/57418c723fe9a62f'. The main workspace, titled 'Flow 1', contains two nodes: an 'IBM IoT' node (blue with a gear icon and a green 'connected' indicator) and a 'debug 1' node (green with a list icon). They are connected by a curved line. The left sidebar shows a 'filter nodes' search bar and two categories: 'common' (with nodes like inject, debug, complete, catch, status, link in, link call, link out, comment) and 'function' (with nodes like function, switch, change, range). The right sidebar, titled 'debug', shows a log of messages. The messages are as follows:

Timestamp	Node	msg.payload
11/2/2022, 8:57:33 PM	node: debug 1	{ temperature: 103, humidity: 31 }
11/2/2022, 8:57:35 PM	node: debug 1	{ temperature: 96, humidity: 76 }
11/2/2022, 8:57:37 PM	node: debug 1	{ temperature: 56, humidity: 90 }
11/2/2022, 8:57:39 PM	node: debug 1	{ temperature: -4, humidity: 13 }
11/2/2022, 8:57:41 PM	node: debug 1	{ temperature: 3, humidity: 19 }
11/2/2022, 8:57:43 PM	node: debug 1	{ temperature: 50, humidity: 37 }

The bottom of the image shows a Windows taskbar with various application icons and a system tray indicating the temperature is 26°C and it is raining.

```

C:\Users\AMORA> node-red
12 Nov 12:34:32 - [info] Dashboard version 3.2.0 started at /ul
12 Nov 12:34:32 - [info] Settings file : C:\Users\AMORA\Documents\node-red\settings.js
12 Nov 12:34:32 - [info] Context store : 'default' [module=memory]
12 Nov 12:34:32 - [info] User directory : Users\AMORA\Documents\node-red
12 Nov 12:34:32 - [warn] Projects disabled : editorTheme.projects.enabled=false
12 Nov 12:34:32 - [info] Flow file : Users\AMORA\Documents\node-red\flows.json
12 Nov 12:34:32 - [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.
-----

12 Nov 12:34:32 - [info] Server now running at http://127.0.0.1:1880/
12 Nov 12:34:32 - [info] Starting flows
12 Nov 12:34:32 - [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\AMORA\Documents> node-red
12 Nov 12:48:03 - [info]
Welcome to Node-RED
-----

12 Nov 12:48:03 - [info] Node-RED version: v2.0.2
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 12:48:03 - [info] Loading palette nodes
12 Nov 12:48:03 - [info] Dashboard version 3.2.0 started at /ul
12 Nov 12:48:03 - [info] Settings file : C:\Users\AMORA\Documents\node-red\settings.js
12 Nov 12:48:03 - [info] Context store : 'default' [module=memory]
12 Nov 12:48:03 - [info] User directory : Users\AMORA\Documents\node-red
12 Nov 12:48:03 - [warn] Projects disabled : editorTheme.projects.enabled=false
12 Nov 12:48:03 - [info] Flow file : Users\AMORA\Documents\node-red\flows.json
12 Nov 12:48:03 - [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

```


