

Create Node Red service

Team ID	PNT2022TMID06078
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 links to Catalog, Manage, and Akshaya M's Account. The main content area displays a grid of application cards. The 'Node-RED App' card is highlighted with a blue box. The card for 'Node-RED App' by IBM is titled 'Node-RED App' and describes it as a tool for building and deploying applications. It lists starter kits for IBM Cloud Kubernetes Service and Red Hat OpenShift. Other visible cards include 'Delphix DevOps Data Platform for IBM Cloud', 'Go Gin App', 'Java Liberty App', 'Java Spring App', 'Node.js Express App', 'PAYTESTER', and 'Plesk'.

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 interface is divided into several sections:

- Details:** This section contains fields for "App URL", "Source" (with a "Download code" button), "Resource group" (set to "Default"), "Deployment target", and "Created" (11/2/2022).
- Services:** This section shows the "Cloudant" service with links to "Open dashboard", "Documentation", and "API reference". It also includes a "Credentials" dropdown and buttons for "Connect existing services" and "Create service".
- Deployment Automation:** This section explains how to "Configure Continuous Delivery" and includes a "Deploy your app" button.
- Getting started quickly:** This sidebar provides a 5-step guide for configuring the app and deploying it.

The bottom of the screen shows a Windows taskbar with various application icons and system status information, including the date and time (02-11-2022, 20:46).

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 URL is cloud.ibm.com/developer/appservice/apps/4d9d88fa-dba5-4b56-986c-76b3909fe692. The page title is "IBM Cloud". The search bar contains "Search resources and products...". The navigation bar includes "Catalog", "Manage", and "Akshaya M's Account".

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

.....

New +

Container registry region

Dallas

Container namespace

jbmfyhfuv...nncumphsw

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. The browser's address bar shows the URL `127.0.0.1:1880/#flow/57418c723fe9a62f`. The 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. A large, semi-transparent 'Wondershare PDFelement' watermark is visible across the center of the workspace. 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).

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

The screenshot displays the Node-RED web interface in a browser. The top bar shows several open tabs: WhatsApp, IBM-Project-48101-1660804426/, IBM App Development, and Node-RED. The address bar indicates the URL 127.0.0.1:1880/#flow/57418c723fe9a62f. The main workspace shows a flow named 'Flow 1' with a single node labeled 'IBM IoT' (with a green 'connected' status) connected to a 'debug 1' node. The left sidebar contains 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 right sidebar is open to the 'Edit ibmiot in node' configuration panel. This panel includes a 'Delete' button, 'Cancel', and 'Done' buttons. The 'Properties' section contains 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), and Service (registered). A yellow tooltip at the bottom of the properties section 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'. At the bottom of the configuration panel is an 'Enabled' checkbox. The bottom status bar shows the system clock at 20:57 on 02-11-2022, the language set to ENG IN, and the weather as 26°C with rain off and on.

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

Node-RED interface showing a flow named "Flow 1" with an "IBM IoT" node connected to a "debug 1" node. The "debug 1" node is highlighted with an orange dashed border. The "IBM IoT" node is labeled "connected".

The left sidebar shows the "common" and "function" node categories. The "common" category includes nodes like inject, debug, complete, catch, status, link in, link call, link out, and comment. The "function" category includes nodes like function, switch, change, and range.

The right sidebar shows the "debug" console with a list of messages. The messages are filtered by "all nodes" and show the following data:

- 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 status bar shows the system clock at 20:57 on 02-11-2022, along with weather information (26°C, Rain off and on) and various system icons.

```
11 Nov 12:34:32 - [info] Dashboard version 3.2.0 started at /ui
11 Nov 12:34:32 - [info] Settings file : C:\Users\AMORA_EDITH\code-red\settings.js
11 Nov 12:34:32 - [info] Context store : default [module=memory]
11 Nov 12:34:32 - [info] User directory : Users\AMORA_EDITH\code-red
11 Nov 12:34:32 - [warn] Projects disabled : editorTheme.projects.enabled=false
11 Nov 12:34:32 - [info] Flows file : Users\AMORA_EDITH\code-red\flows.json
11 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.
-----

11 Nov 12:34:32 - [info] Server now running at http://127.0.0.1:1880/
11 Nov 12:34:32 - [info] Starting flows
11 Nov 12:34:32 - [info] Started flows
11 Nov 12:43:31 - [info] Stopping flows
11 Nov 12:47:51 - [info] Stopped flows
Terminate batch job (Y/N)? y

C:\Users\AMORA_EDITH>code-red
C:\Users\AMORA_EDITH>code-red
11 Nov 12:48:03 - [info]

Welcome to Node-RED

-----
11 Nov 12:48:03 - [info] Node-RED version: v3.0.2
11 Nov 12:48:03 - [info] Node.js version: v14.17.1
11 Nov 12:48:03 - [info] Windows_NT 10.0.19045 x64 IE
11 Nov 12:48:04 - [info] Loading palette nodes
11 Nov 12:48:05 - [info] Dashboard version 3.2.0 started at /ui
11 Nov 12:48:05 - [info] Settings file : C:\Users\AMORA_EDITH\code-red\settings.js
11 Nov 12:48:05 - [info] Context store : default [module=memory]
11 Nov 12:48:05 - [info] User directory : Users\AMORA_EDITH\code-red
11 Nov 12:48:05 - [warn] Projects disabled : editorTheme.projects.enabled=false
11 Nov 12:48:05 - [info] Flows file : Users\AMORA_EDITH\code-red\flows.json
11 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.
-----

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

The screenshot displays the Node-RED web interface. On the left, the 'common' palette contains nodes like inject, debug, complete, catch, status, link in, link call, link out, and comment. The 'function' palette includes function, switch, change, range, template, and delay. The central workspace shows a flow named 'Flow 1' with an 'IBM IoT' node (labeled 'connected') connected to a 'level' sensor node and a 'show notification' node. The right sidebar shows the 'config' tab with settings for 'On all flows', including 'ibmiot' (SWM: 1, TestGogul: 0), 'ui_base' (Node-RED Dashboard), 'ui_group' ([admin] Default: 1, [unassigned] Default: 0), and 'ui_tab' (admin: 1). Below these are sections for 'Flow 1', 'Flow 2', and 'Flow 3'.