

Create Node Red service

Team ID	NM2023TMID04301
Project Name	Smart waste management system

Step 1: Login into IBM CLOUD account

Step2: In catalog, search for node red application

The screenshot shows the IBM Cloud Catalog interface. The browser address bar displays 'cloud.ibm.com/catalog?category=devops'. The page header includes 'IBM Cloud', a search bar, and navigation links for '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. Other visible cards include 'Delphix DevOps Data Platform for IBM Cloud', 'GeneXus', 'Go Gin App', 'Java Liberty App', 'Java Spring App', 'Node.js Express App', 'PAYTESTER', and 'Plesk'. 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 temperature as 26°C and the date as 02-11-2022.

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 tabs include '(1) 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 of the IBM Cloud console includes the 'IBM Cloud' logo, a search bar, and navigation links for 'Catalog', 'Manage', and 'Akshaya M's Account'.

The page title is 'Node RED DXQJC 2022-11-02' with an 'Add tags' link and an 'Actions...' dropdown menu.

The 'Details' section shows the following information:

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

The 'Services' section shows the 'Cloudant' service with links to 'Open dashboard', 'Documentation', and 'API reference'. There are buttons for 'Connect existing services' and 'Create service'.

The 'Deployment Automation' section has a 'Deploy your app' button. Below it, a message states: 'Continuous Delivery is not enabled for this app. Enable Continuous Delivery to automate builds, tests, and deployments through Delivery Pipeline, GitLab, and more.'

A 'Getting started quickly' sidebar on the right provides instructions for configuring the app:

- 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

The bottom of the screen shows a Windows taskbar with various application icons, a system tray with weather information (26°C, Rain off and on), and the date/time (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 main content area displays four deployment target options: Kubernetes Service (selected), Red Hat OpenShift, Cloud Foundry, and Code Engine. Below these, there are fields for the IBM Cloud API key, Container registry region (Dallas), and Cluster configuration (Frankfurt, Default resource group, default namespace, mycluster-free cluster name). The right-hand sidebar provides instructions for Step 1: Select the deployment target, including a description of Kubernetes Service and a list of steps to follow.

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 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' 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 overlaid diagonally across the workspace. On the right side, there is a 'debug' console with a 'Deploy' button and a 'debug 1' tab. The bottom of the screen shows a Windows taskbar with various application icons and a system tray displaying the temperature (26°C), weather (Rain to stop), and time (18:22, 02-11-2022).

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 indicates the URL is 127.0.0.1:1880/#flow/57418c723fe9a62f. The interface includes a left sidebar with node categories (common, function), a central workspace with a flow diagram, and a right sidebar with a properties editor and a debug console.

In the central workspace, a flow named "Flow 1" is visible. It contains an "IBM IoT" node (blue icon with a gear) connected to a "debug 1" node (green icon with a list). The "IBM IoT" node has a "connected" status indicator.

The right sidebar shows the "Edit ibmiot in node" configuration panel. The "Properties" section includes 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

Below the properties, there is a note: "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 panel, there is an "Enabled" checkbox.

The bottom of the screen shows a Windows taskbar with various application icons and a system tray displaying the date and time (20:57 02-11-2022).

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

The screenshot shows the Node-RED web interface in a browser. The address bar indicates the URL is `127.0.0.1:1880/#flow/57418c723fe9a62f`. The interface includes a left sidebar with node categories (common and function), a central workspace with a flow diagram, and a right sidebar with a debug console.

Flow Diagram: A flow named "Flow 1" contains two nodes: an "IBM IoT" node (blue) and a "debug 1" node (green). The "IBM IoT" node is labeled "connected" and is connected to the "debug 1" node via a curved line.

Debug Console: The debug console on the right shows a list of messages. Each message is a JSON object with the following structure:

```

{
  "type": "Test",
  "id": "Test123",
  "evt/status/fmt/json": "msg.payload",
  "msg.payload": {
    "temperature": ,
    "humidity": 
  }
}

```

The messages show the following temperature and humidity values:

- 11/2/2022, 8:57:33 PM: temperature: 103, humidity: 31
- 11/2/2022, 8:57:35 PM: temperature: 96, humidity: 76
- 11/2/2022, 8:57:37 PM: temperature: 56, humidity: 90
- 11/2/2022, 8:57:39 PM: temperature: -4, humidity: 13
- 11/2/2022, 8:57:41 PM: temperature: 3, humidity: 19
- 11/2/2022, 8:57:43 PM: temperature: 50, humidity: 37

The bottom of the screen shows a Windows taskbar with the date and time as 20:57 on 02-11-2022, and the weather as 26°C with rain off and on.

```
13 Nov 12:34:32 - [info] Dashboard version 3.2.0 started at /ui
13 Nov 12:34:32 - [info] Settings file : C:\Users\AMORA_EDITH\code-red\settings.js
13 Nov 12:34:32 - [info] Context store : default [module-memory]
13 Nov 12:34:32 - [info] User directory : Users\AMORA_EDITH\code-red
13 Nov 12:34:32 - [warn] Projects disabled : editorTheme.projects.enabled=false
13 Nov 12:34:32 - [info] Flows file : Users\AMORA_EDITH\code-red\flows.json
13 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.

13 Nov 12:34:32 - [info] Server now running at http://127.0.0.1:1880/
13 Nov 12:34:32 - [info] Starting flows
13 Nov 12:34:32 - [info] Started flows
13 Nov 12:47:51 - [info] Stopping flows
13 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
13 Nov 12:48:03 - [info]

Welcome to Node-RED

13 Nov 12:48:03 - [info] Node-RED version: v3.0.2
13 Nov 12:48:03 - [info] Node.js version: v14.17.1
13 Nov 12:48:03 - [info] Windows_NT 10.0.19045 x64 IE
13 Nov 12:48:04 - [info] Loading palette nodes
13 Nov 12:48:05 - [info] Dashboard version 3.2.0 started at /ui
13 Nov 12:48:05 - [info] Settings file : C:\Users\AMORA_EDITH\code-red\settings.js
13 Nov 12:48:05 - [info] Context store : default [module-memory]
13 Nov 12:48:05 - [info] User directory : Users\AMORA_EDITH\code-red
13 Nov 12:48:05 - [warn] Projects disabled : editorTheme.projects.enabled=false
13 Nov 12:48:05 - [info] Flows file : Users\AMORA_EDITH\code-red\flows.json
13 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.

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

