

Create Node Red service

Team ID	PNT2022TMID01046
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 browser tabs include 'WhatsApp', 'IBM-Project-48101-1660804426', and 'Catalog - IBM Cloud'. The address bar shows 'cloud.ibm.com/catalog?category=devops'. The page header includes 'IBM Cloud', a search bar, and navigation links like 'Catalog', 'Manage', and 'Akshaya M's Account'. The main content area displays a grid of application cards. On the left, there are filters for 'Location' (Dallas, Frankfurt, London, Montreal, Osaka, Sao Paulo) and 'Support' (IBM supported, Third party supported). The application cards include:

- 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.

The 'Node-RED App' card is highlighted with a blue underline. The bottom of the screen shows a Windows taskbar with various icons and a system tray displaying '26°C Rain off and on', 'ENG IN', and the date '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. At the top, there's a navigation bar with the IBM Cloud logo, a search bar, and user account information. The main content area is divided into several sections:

- Details:** A table showing app information. The 'App URL' and 'Deployment target' fields both display the message 'You must deploy your app first'. The 'Source' field has a 'Download code' button. The 'Resource group' is set to 'Default', and the 'Created' date is '11/2/2022'.
- Services:** A section for managing services, currently showing 'Cloudant'. It includes links for 'Open dashboard', 'Documentation', and 'API reference', along with a 'Credentials' dropdown. At the bottom are buttons for 'Connect existing services' and 'Create service'.
- Deployment Automation:** A section titled 'Configure Continuous Delivery'. It states that Continuous Delivery is not enabled for this app and provides instructions on how to enable it. A prominent blue button labeled 'Deploy your app' is visible.
- Getting started quickly:** A sidebar on the right with a list of steps for configuring the app, connecting services, and deploying.

The bottom of the screen shows a Windows taskbar with various application icons, the system clock indicating 20:46 on 02-11-2022, and weather information for 26°C with rain off and on.

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


(1) WhatsApp x IBM-Project-48101-1660804426/ x IBM App Development x +


cloud.ibm.com/developer/appservice/apps/4d9d88fa-dba5-4b56-986c-76b3909fe692


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

.....

Key New +

Container registry region

Dallas

Container registry namespace

jbmfyhfuvvmqrrymgrbnnfcumphsw

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

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' 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 notification from 'meet.google.com' is visible at the bottom center of the screen.

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 navigation bar shows several 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, titled 'Flow 1', contains a flow with an 'IBM IoT' node (blue icon with a gear) connected to a 'debug 1' node (green icon with a list). The 'IBM IoT' node is marked as 'connected'. On the left, the 'common' node palette includes inject, debug, complete, catch, status, link in, link call, link out, and comment. The 'function' node palette includes function, switch, change, and range. The right sidebar is open to the 'Edit ibmiot in node' configuration panel. It features a 'Delete' button, 'Cancel', and 'Done' buttons. The 'Properties' section includes: 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, there is an 'Enabled' checkbox. 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.

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 contains a palette of nodes, including 'common' (inject, debug, complete, catch, status, link in, link call, link out, comment) and 'function' (function, switch, change, range). The main workspace shows a flow with an 'IBM IoT' node (blue icon, labeled 'connected') connected to a 'debug 1' node (green icon). The right sidebar displays the 'debug' console, showing a list of messages received from the 'debug 1' node. The messages are JSON objects containing 'temperature' and 'humidity' values.

Node-RED interface showing a flow with an IBM IoT node connected to a debug node. The debug console on the right displays the following messages:

- 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 }

