

**Create Node Red service**

Team ID	PNT2022TMID30432
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 Developer console for an application named "Node RED DXQJC 2022-11-02". The interface is divided into several sections:

- Details:**
  - App URL: You must deploy your app first
  - Source: Download code (with a download icon)
  - Resource group: Default
  - Deployment target: You must deploy your app first
  - Created: 11/2/2022
- Services:**
  - Cloudant service is listed with links for "Open dashboard", "Documentation", and "API reference".
  - Buttons for "Connect existing services" and "Create service" are at the bottom.
- Deployment Automation:**
  - Section titled "Configure Continuous Delivery" with a message: "Continuous Delivery is not enabled for this app. Enable Continuous Delivery to automate builds, tests, and deployments through Delivery Pipeline, GitLab, and more."
  - A prominent blue button labeled "Deploy your app" with a cloud icon.
- Getting started quickly (Sidebar):**
  - Section titled "Configuring your app" with instructions: "To connect services and DevOps toolchains to your app:"
  - Five numbered steps:
    - 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 image shows a Windows taskbar with various application icons (including Edge, Word, and File Explorer) and system status information (26°C, Rain off and on, 20:46, 02-11-2022).

Step 3: Enter the project details and click on create  
 Step 4: click on deploy option and deploy





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


IBM Cloud


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

jbmfyhfuv...nfcumphsw

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

IBM App Development | Meet - aes-ejhp-kjb | (1) WhatsApp | Node-RED

127.0.0.1:1880/#flow/57418c723fe9a62f

Node-RED

Flow 1

debug

all nodes | all

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function
- switch
- change
- range

IBM IoT

connected

debug 1

meet.google.com is sharing your screen. Stop sharing Hide

26°C Rain to stop

ENG IN 18:22 02-11-2022

The screenshot shows the Node-RED web interface in a browser. The top bar includes tabs for IBM, IBM App Development, a Google Meet link, WhatsApp, and Node-RED. The address bar shows the local URL 127.0.0.1:1880/#flow/57418c723fe9a62f. The Node-RED interface has a left sidebar with a 'filter nodes' search bar and two categories: 'common' and 'function'. The 'common' category lists nodes like inject, debug, complete, catch, status, link in, link call, link out, and comment. The 'function' category lists function, switch, change, and 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) and a 'debug 1' node (green). They are connected by a curved line. A large blue diamond shape is also present in the workspace. On the right, a 'debug' sidebar is open, showing 'all nodes' and 'all' filters. At the bottom of the browser window, a notification from meet.google.com is visible, and the Windows taskbar shows the date and time as 18:22 on 02-11-2022.

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



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

The main workspace displays a flow with an **IBM IoT** node connected to a **debug 1** node. The **IBM IoT** node is labeled "connected".

The **Edit ibmiot in node** panel on the right shows the following configuration:

- 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

The bottom status bar shows the system clock: 20:57, 02-11-2022.

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

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

Time	Node	Message
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 }

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



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

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

C:\Users\ARORA_EDIT\OneDrive\Documents> node-red

C:\Users\ARORA_EDIT\OneDrive\Documents> node-red
Nov 12:48:03 - [info]

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

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

