

Create NodeRedservice

Team ID	PNT2022TMID12601
ProjectName	Smart waste management system formetropolitan cities

Step1:Logininto IBMCLLOUDaccount

Step2:Incatalog,searchfornodered application

The screenshot displays the IBM Cloud Catalog interface. At the top, there's a navigation bar with the IBM Cloud logo, a search bar, and links for Catalog, Manage, and the user's account (Akshaya M's Account). Below the navigation bar, a search bar prompts the user to "Search the catalog...". On the left side, there are filters for Location (Dallas, Frankfurt, London, Montreal, Osaka, Sao Paulo) and Support (IBM supported, Third party supported). The main area shows a grid of application templates. The "Node-RED App" by IBM is highlighted with a blue underline. Other visible templates 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. Each template card provides a brief description and links to starter kits or related services.

IBM Cloud Search resources and products...

Search the catalog...

Location

- ☐ Dallas
- ☐ Frankfurt
- ☐ London
- ☐ Montreal
- ☐ Osaka
- ☐ Sao Paulo

Show more

Support

- ☐ IBM supported
- ☐ Third party supported

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.

26°C Rain off and on

ENG IN 20:46 02-11-2022

Step3:Enter theprojectdetailsand
Step4:clickondeployoptionand deploy

The screenshot displays the IBM Cloud console interface for a specific application. At the top, the navigation bar includes the IBM Cloud logo, a search bar, and links to Catalog, Manage, and the user's account (Akshaya M's Account). The main header shows the resource list and app details for 'Node RED DXQJC 2022-11-02', with an 'Add tags' link and an 'Actions...' dropdown menu.

The 'Details' section on the left provides information about the app: App URL (placeholder), Source (with a 'Download code' button), Resource group (Default), Deployment target (placeholder), and Created date (11/2/2022).

The 'Services' section below details the 'Cloudant' service, offering links to the Open dashboard, Documentation, and API reference, along with a 'Credentials' dropdown. At the bottom of this section are buttons for 'Connect existing services' and 'Create service'.

The 'Deployment Automation' section on the right explains how to 'Configure Continuous Delivery' and includes a prominent blue 'Deploy your app' button.

A 'Getting started quickly' sidebar on the far right provides a five-step guide for configuring the app, from connecting services to deploying and managing the app. A vertical 'ASK A QUESTION' button is also visible on the right edge.

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: Setup the environment for deploying and click on create

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**
IUM
Deploy, scale, and manage your containerized application workloads to highly available clusters.

**Red Hat OpenShift**
IUM
Deploy your apps on highly available clusters that come installed with Red Hat OpenShift on IBM Cloud.

**Cloud Foundry**
IUM
Deploy and run your applications without managing servers or clusters. A Lite plan is available for quick and easy deployment.

**Code Engine**
IUM
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

Step6:Nowdraganddropthenodesand

The screenshot displays the Node-RED web interface. On the left, the 'common' node palette is visible, containing nodes like inject, debug, complete, catch, status, link in, link call, link out, and comment. Below it, the 'function' palette shows function, switch, change, and range nodes. 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, the 'debug' console is open, showing a list of all nodes and a 'all' button. At the bottom of the interface, a system tray shows the temperature as 26°C with the text 'Rain to stop', a taskbar with various application icons, and a notification from 'meet.google.com' stating 'is sharing your screen.' with 'Stop sharing' and 'Hide' buttons. The system clock in the bottom right corner shows the time as 18:22 on 02-11-2022.

Step7:setup thesettingsthatconnectsnoderedservicewithWatson

The screenshot displays the Node-RED web interface. On the left, the 'common' node palette includes 'inject', 'debug', 'complete', 'catch', 'status', 'link in', 'link call', 'link out', and 'comment'. The 'function' palette includes 'function', 'switch', 'change', and 'range'. The main workspace, titled 'Flow 1', contains an 'IBM IoT' node (labeled 'connected') connected to a 'debug 1' node. On the right, the 'Edit ibmiot in node' panel is open, showing 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

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 'Enabled' checkbox is checked. The top right of the interface shows a 'Deploy' button and a 'debug' console. The bottom status bar indicates the system time as 20:57 on 02-11-2022, with weather information (26°C, Rain off and on) and various system icons.

Step8:Finally,outputcanbeseenin noderedservice

The screenshot shows the Node-RED web interface. On the left, the 'common' nodes palette is visible, including 'inject', 'debug', 'complete', 'catch', 'status', 'link in', 'link call', 'link out', and 'comment'. The 'function' nodes palette includes 'function', 'switch', 'change', and 'range'. In the center workspace, a flow named '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). A wire connects the output of the 'IBM IoT' node to the input of the 'debug 1' node. On the right, the 'debug console' is open, showing a list of messages. Each message is a JSON object with 'temperature' and 'humidity' fields. The messages are timestamped and include the node name 'node: debug 1' and the payload type 'msg.payload: Object'.

Node-RED interface showing a flow named 'Flow 1' with two nodes: 'IBM IoT' (connected) and 'debug 1'. The 'debug 1' node is outputting data to the debug console.

Debug console output (messages):

- 11/2/2022, 8:57:33 PM node: debug 1
iot-2/hyper/TestId/Test123/evt/status/fmt/json : msg.payload : Object
{ temperature: 103, humidity: 31 }
- 11/2/2022, 8:57:35 PM node: debug 1
iot-2/hyper/TestId/Test123/evt/status/fmt/json : msg.payload : Object
{ temperature: 96, humidity: 76 }
- 11/2/2022, 8:57:37 PM node: debug 1
iot-2/hyper/TestId/Test123/evt/status/fmt/json : msg.payload : Object
{ temperature: 56, humidity: 90 }
- 11/2/2022, 8:57:39 PM node: debug 1
iot-2/hyper/TestId/Test123/evt/status/fmt/json : msg.payload : Object
{ temperature: -4, humidity: 13 }
- 11/2/2022, 8:57:41 PM node: debug 1
iot-2/hyper/TestId/Test123/evt/status/fmt/json : msg.payload : Object
{ temperature: 3, humidity: 19 }
- 11/2/2022, 8:57:43 PM node: debug 1
iot-2/hyper/TestId/Test123/evt/status/fmt/json : msg.payload : Object
{ temperature: 50, humidity: 37 }

