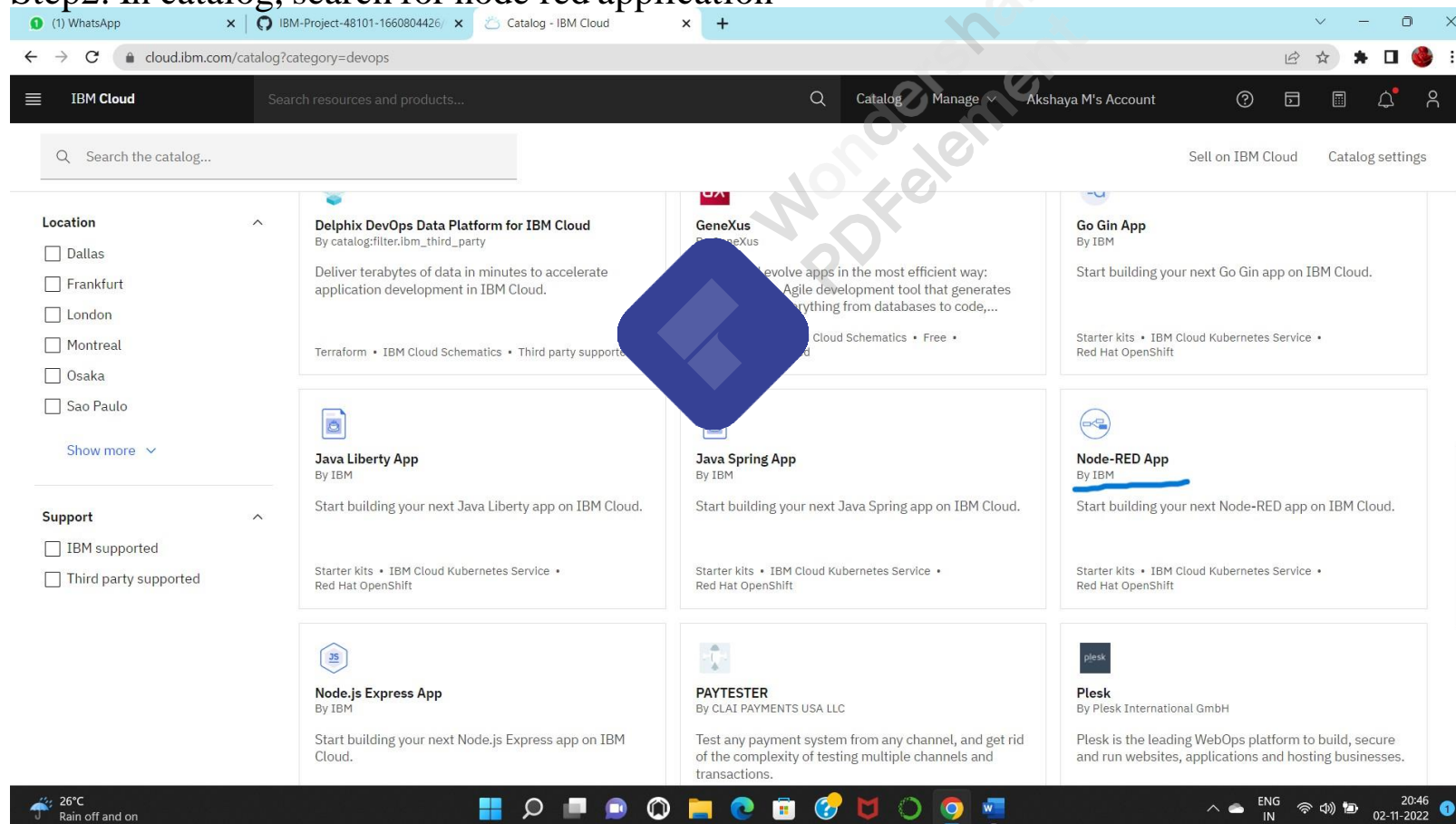


## Create Node Red service

Team ID	PNT2022TMID06075
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 the user's account (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)
- GeneXus** (By GeneXus)
- Go Gin App** (By IBM)
- Java Liberty App** (By IBM)
- Java Spring App** (By IBM)
- Node-RED App** (By IBM) - This card is highlighted with a blue box.
- Node.js Express App** (By IBM)
- PAYTESTER** (By CLAI PAYMENTS USA LLC)
- Plesk** (By Plesk International GmbH)

The bottom of the screen shows a Windows taskbar with the date 02-11-2022 and time 20:46.

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 address bar displays the URL: `cloud.ibm.com/developer/appservice/apps/4d9d88fa-dba5-4b56-986c-76b3909fe692`. The page title is "Node RED DXQJC 2022-11-02".

**Details Section:**

- App URL:** You must deploy your app first
- Source:** Download code (button)
- Resource group:** Default
- Deployment target:** You must deploy your app first
- Created:** 11/2/2022

**Services Section:**

- Cloudant:** Open dashboard, Documentation, API reference, Credentials (dropdown)
- Buttons:** Connect existing services (+), Create service (+)

**Deployment Automation Section:**

- Configure Continuous Delivery:** Continuous Delivery is not enabled for this app. Enable Continuous Delivery to automate builds, tests, and deployments through Delivery Pipeline, GitLab, and more.
- Deploy your app** (button)

**Getting started quickly Section:**

- Configuring your app:** To connect services and DevOps toolchains to your app:

  1. Use the **Services** card to connect a service to your app. Select an existing service instance, or create a new one. [Learn more.](#)
  2. If you want to view the code before your app is deployed, click **Download code** to obtain the .zip file.
  3. Click **Deploy your app** in the **Deployment Automation** card to select the deployment target and configure the Continuous Delivery service. The deployment begins automatically.
  4. After the deployment begins, you can view the status of the deployment, modify your app, view your repo, or view the app's URL.
  5. If you make any changes to your app, be

The bottom of the screen shows a Windows taskbar with various application icons and a system tray displaying the date and 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 IBM Cloud API key, Container registry region (Dallas), Cluster region (Frankfurt), Cluster resource group (Default), Cluster namespace (default), and Cluster name (mycluster-free). A large blue watermark is visible over the API key field.

**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.

The bottom of the screen shows a Windows taskbar with the date 02-11-2022 and time 20:47.

## 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 window. 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, 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 address bar shows the URL `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 for node configuration and debugging.

In the central workspace, a flow named "Flow 1" 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 is labeled "connected".

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

A note at the bottom of the configuration panel states: "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 as 20:57 on 02-11-2022, and the weather as 26°C with rain off and on.



## 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 `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 debug console.

**Flow Diagram:**

- A flow named "Flow 1" is shown.
- An "IBM IoT" node (blue) is connected to a "debug 1" node (green).
- The "IBM IoT" node has a status indicator showing "connected".

**Debug Console Output:**

The debug console displays a series of messages from "node: debug 1". Each message is a JSON object containing temperature and humidity 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 }
```

```
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 12:47:51 - [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
```

Node-RED interface showing a flow diagram and configuration panel.

**Flow 1:**

- Inject node
- Debug node
- Complete node
- Catch node
- Status node
- Link in node
- Link call node
- Link out node
- Comment node

**Function palette:**

- function
- switch
- change
- range
- template
- delay

**Flow diagram:**

```
graph LR; IBM[IBM IoT] --> level[level]; level --> show[show notification];
```

**Configuration panel (On all flows):**

- ibmiot: SWM (1), TestGogul (0)
- ui\_base: Node-RED Dashboard
- ui\_group: [admin] Default (1), [unassigned] Default (0)
- ui\_tab: admin (1)

**Flow list:**

- Flow 1
- Flow 2
- Flow 3