



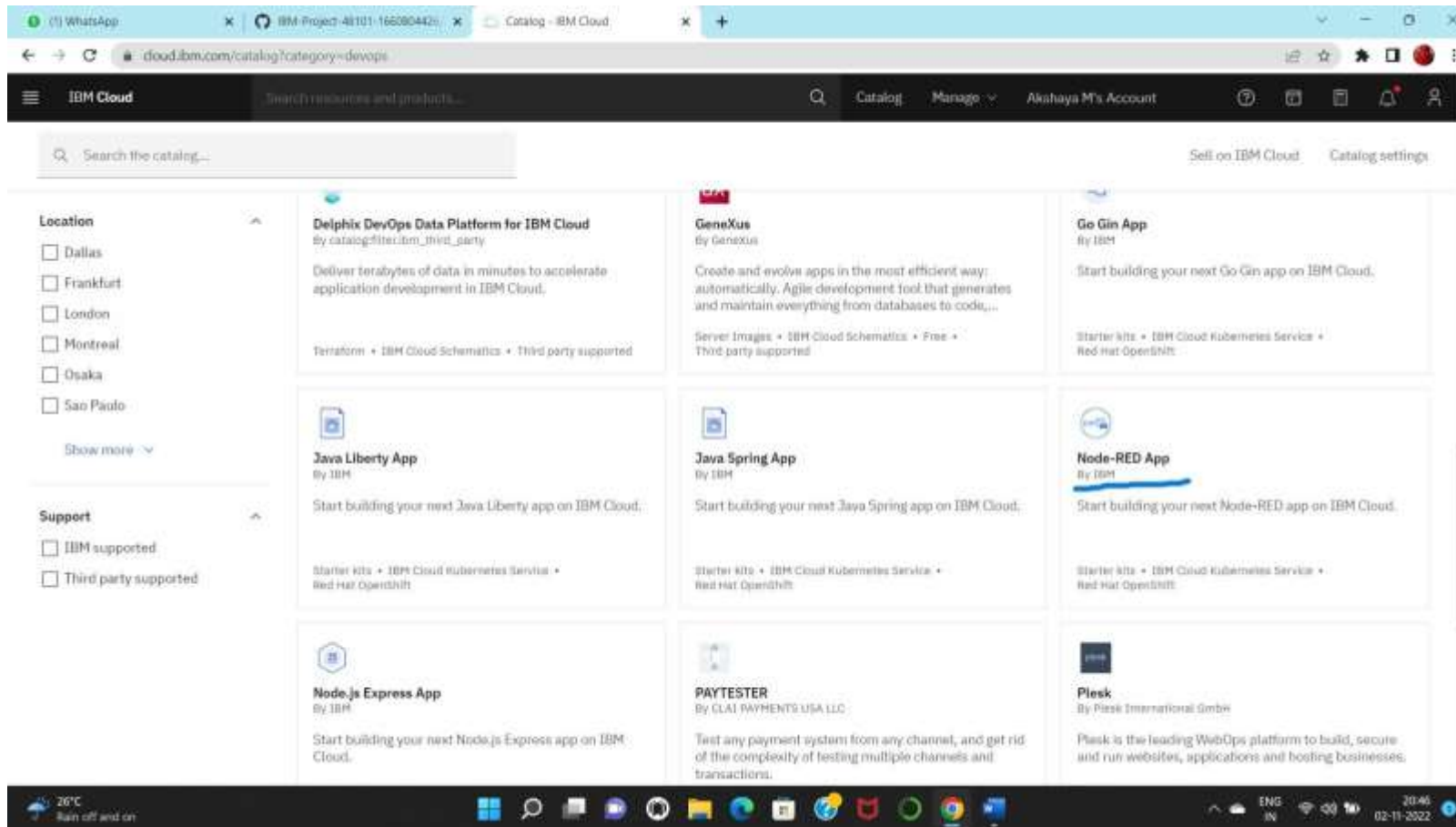
### Create Node Red service

Team ID	PNT2022TMID16061
Project Name	Smart waste management system for metropolitan cities

Step 1: Login into IBM CLOUD account

Step2: In catalog, search for node red application





Step 3: Enter the project details and click on create

Step 4: click on deploy option and deploy

Resource list / App details /

## Node RED DXQJC 2022-11-02 [Add tags](#)

Actions...

### Details

App URL You must deploy your app first

Source [Download code](#)

Resource group [Default](#)

Deployment target You must deploy your app first

Created 11/2/2022

### Services

**Cloudant**

[Open dashboard](#) [Documentation](#) [API reference](#)

Credentials

[Connect existing services](#) [Create service](#)

### Deployment Automation

**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](#)

### Getting started quickly

#### 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

ASK A QUESTION

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

The screenshot shows the IBM Cloud Developer console interface. The top navigation bar includes the IBM Cloud logo, a search bar, and links for Catalog, Manage, and the user's account (Akshaya M's Account). The main content area is titled "Select your deployment target and configure your DevOps toolchain. After you click **Create**, the toolchain is created, and the deployment process is started automatically."

Under the "Deployment target" section, four options are displayed:

- Kubernetes Service**: Deploy, scale, and manage your containerized application workloads to highly available clusters.
- Red Hat OpenShift**: Deploy your apps on highly available clusters that come installed with Red Hat OpenShift on IBM Cloud.
- Cloud Foundry**: Deploy and run your applications without managing servers or clusters. A Lite plan is available for quick and easy deployment.
- Code Engine**: Run your app, job, or container on a managed serverless platform. Auto-scale workloads, and pay only for the resources that you consume.

Below the deployment targets, the "IBM Cloud API key" field is shown with a masked key and a "New" button. The "Container registry region" is set to "Dallas" and the "Container registry namespace" is "jbmfyhfuvmyqrrymgrbnncumhsw". The "Cluster region" is "Frankfurt", the "Cluster resource group" is "Default", the "Cluster namespace" is "default", and the "Cluster name" is "mycluster-free".

On the right side, a sidebar titled "Step 1. Select the deployment target" provides additional information about the IBM Cloud Kubernetes Service, including a description of Kubernetes and a list of steps to follow:

**Before you begin**

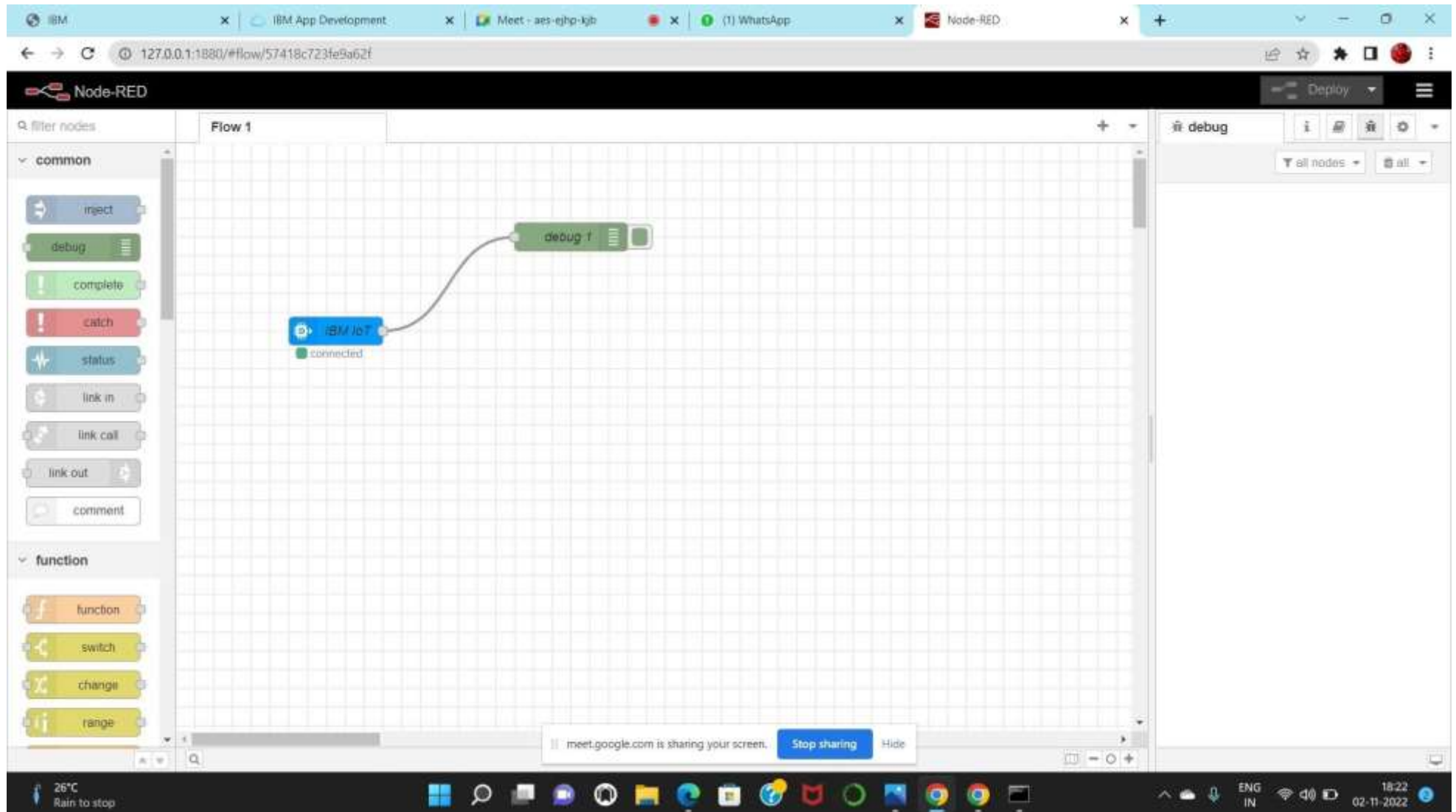
- One free Kubernetes cluster is available per account.
- If you don't have an available cluster, you must create one before continuing. Allow 10-20 minutes for the cluster to be provisioned. [Create cluster.](#)

**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 various application icons and a system tray displaying the temperature (26°C), weather (Rain off and on), and the date/time (02-11-2022, 20:47).

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



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 top bar includes tabs for WhatsApp, IBM-Project-48101-1660004426, IBM App Development, and Node-RED. The address bar shows the URL 127.0.0.1:1880/#flow/57418c723fe9a62f. The main workspace displays a flow with a blue 'IBM IoT' node connected to a green 'debug' node. The left sidebar shows the 'common' and 'function' node categories. The right sidebar is open to the 'Edit IBM IoT in node' configuration panel. This panel includes a 'Properties' section with 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), and 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 bottom status bar shows the system clock as 20:57 on 02-11-2022, along with weather and network icons.

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

Node-RED

Deploy

filter nodes

Flow 1

common

function

IBM IoT

debug 1

debug

all nodes

all

11/2/2022, 8:57:33 PM node: debug 1

iot-2/type/TestId/Test123/evt/status/fmt/json : msg.payload

Object

{ temperature: 103, humidity: 31 }

11/2/2022, 8:57:35 PM node: debug 1

as-2/type/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/type/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/type/TestId/Test123/evt/status/fmt/json : msg.payload

Object

{ temperature: -4, humidity: 13 }

11/2/2022, 8:57:41 PM node: debug 1

as-2/type/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/type/TestId/Test123/evt/status/fmt/json : msg.payload

Object

{ temperature: 50, humidity: 37 }

26°C

Rain off and on

20:57

02-11-2022

```

11 Nov 12:00:02 [info] Backend server 0.0.0 started at 100
11 Nov 12:00:02 [info] Settings file : /usr/local/etc/HTTPD.conf-modhttpd.conf
11 Nov 12:00:02 [info] Current user : httpd [daemon-process]
11 Nov 12:00:02 [info] User directory : /usr/local/etc/httpd
11 Nov 12:00:02 [info] ProxyName disabled : address=www.projects.mobilweb.fr
11 Nov 12:00:02 [info] User file : /usr/local/etc/httpd-modhttpd.conf
11 Nov 12:00:02 [info]
11 Nov 12:00:02 [info]

```

Now the `modhttpd.conf` file is overwritten using a system-generated file.

If the system-generated file is lost for any reason, your `modhttpd.conf` file will not be recoverable, you will then be forced to get a new one.

You should use your own file using the `modhttpd.conf` option in the settings file. Now `HTTPD` will then re-overwrite your `modhttpd.conf` file using your chosen file the next time you finish a change.

---

```

11 Nov 12:00:02 [info] Server now running at http://10.0.0.1:8080
11 Nov 12:00:02 [info] Starting Proxy
11 Nov 12:00:02 [info] Starting Proxy
11 Nov 12:00:02 [info] Starting Proxy
11 Nov 12:00:02 [info] Starting Proxy
11 Nov 12:00:02 [info] Starting Proxy
11 Nov 12:00:02 [info]
11 Nov 12:00:02 [info]

```

11 Nov 12:00:02 [info] Backend server 0.0.0 started at 100

11 Nov 12:00:02 [info] Settings file : /usr/local/etc/HTTPD.conf-modhttpd.conf

11 Nov 12:00:02 [info] Current user : httpd [daemon-process]

11 Nov 12:00:02 [info] User directory : /usr/local/etc/httpd

11 Nov 12:00:02 [info] ProxyName disabled : address=www.projects.mobilweb.fr

11 Nov 12:00:02 [info] User file : /usr/local/etc/httpd-modhttpd.conf

11 Nov 12:00:02 [info]

11 Nov 12:00:02 [info]

