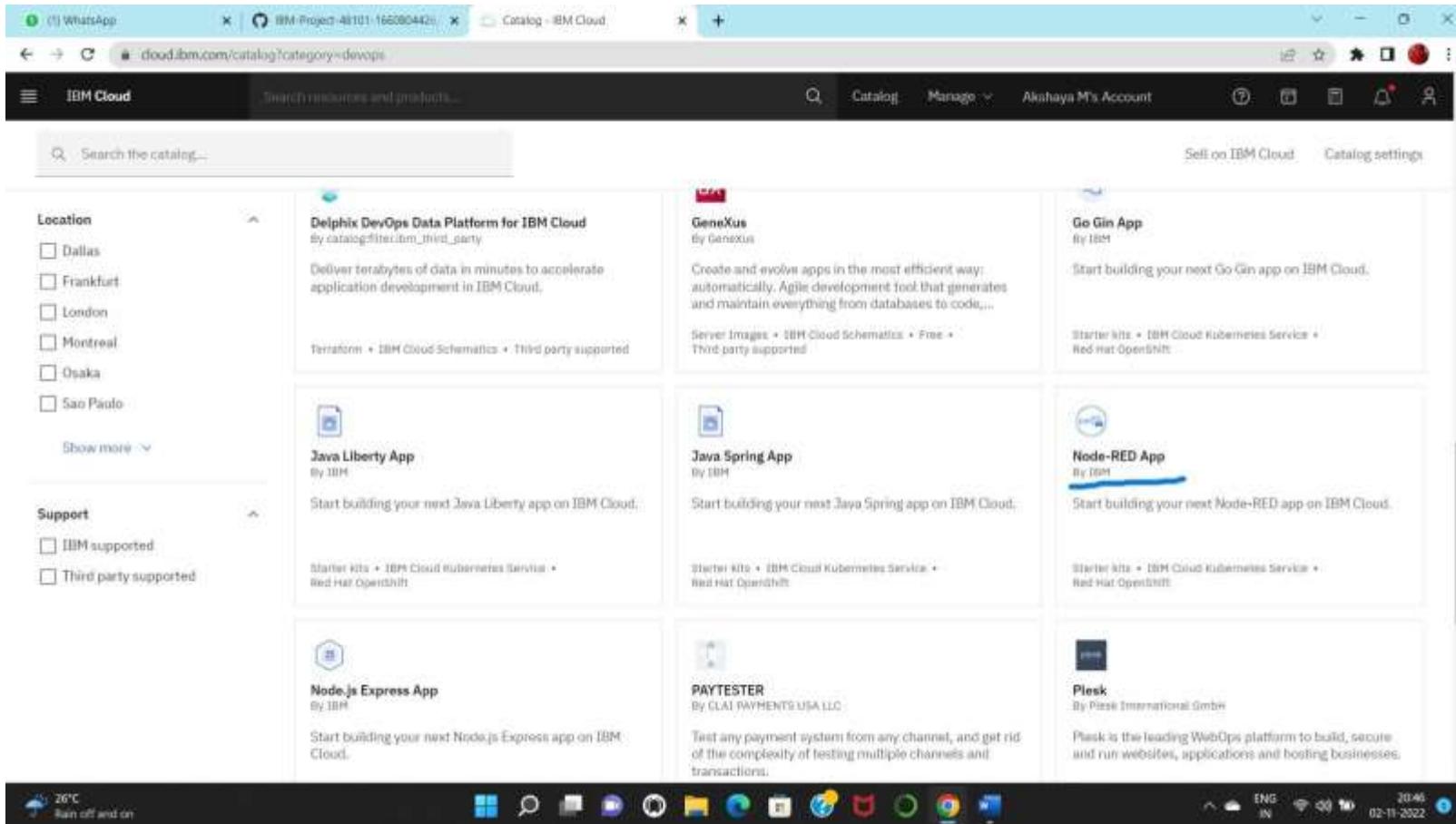


Create Node Red service

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



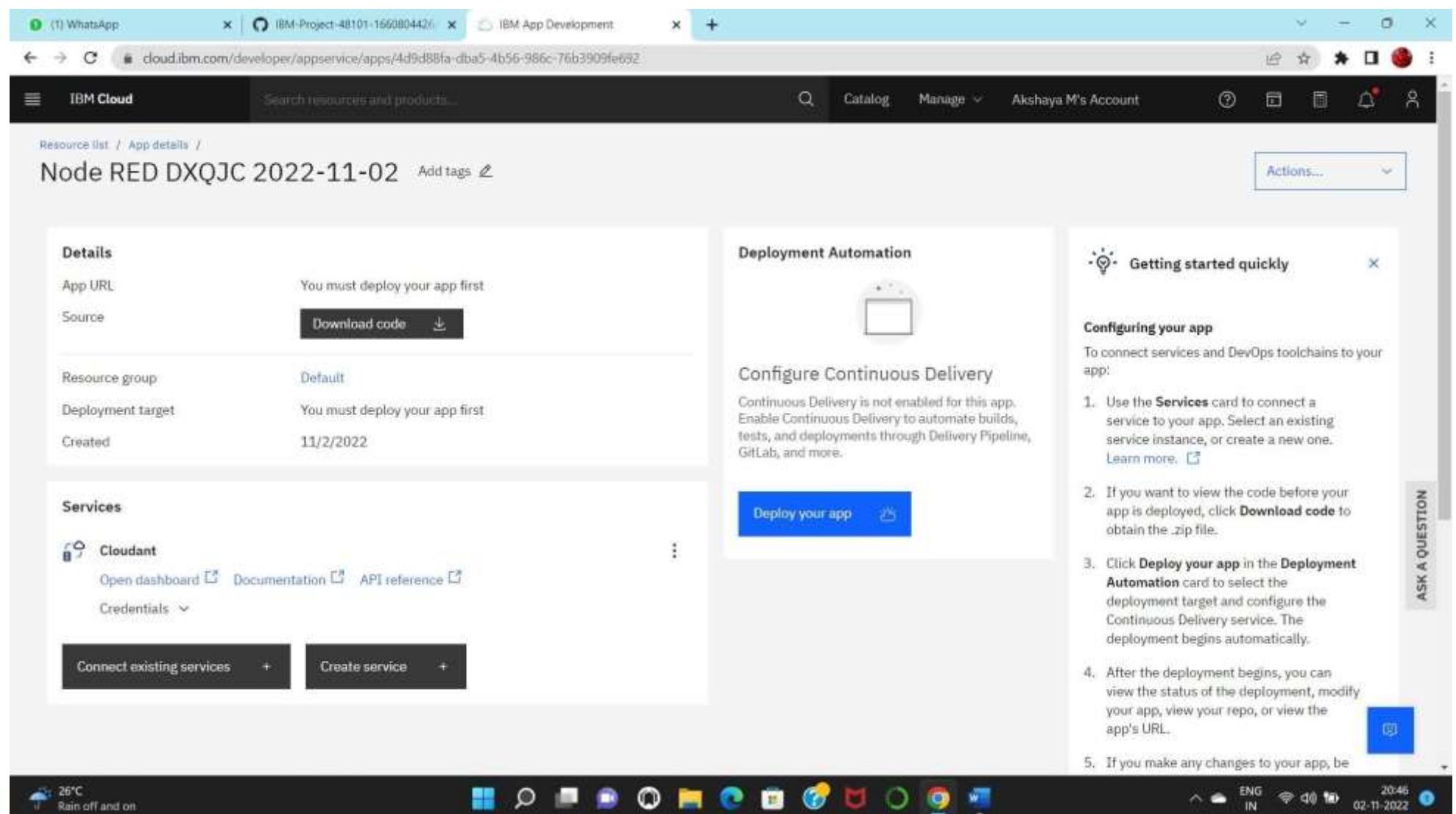
The screenshot shows the IBM Cloud Catalog interface. On the left, there are filters for Location (Dallas, Frankfurt, London, Montreal, Osaka, Sao Paulo) and Support (IBM supported, Third party supported). The main area displays several application cards:

- Delphix DevOps Data Platform for IBM Cloud** by [catalyst@ibm.com_third_party](#): Deliver terabytes of data in minutes to accelerate application development in IBM Cloud.
- GeneXus** by [GeneXus](#): Create and evolve apps in the most efficient way: automatically. Agile development tool that generates and maintains everything from databases to code,...
- Go Gin App** by [IBM](#): Start building your next Go Gin app on IBM Cloud.
- Java Liberty App** by [IBM](#): Start building your next Java Liberty app on IBM Cloud.
- Java Spring App** by [IBM](#): Start building your next Java Spring app on IBM Cloud.
- Node-RED App** by [IBM](#): Start building your next Node-RED app on IBM Cloud.
- 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.

At the bottom, the taskbar shows the system status: 26°C, Rain off and on, ENG, IN, 20:46, 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 App Development interface for a Node RED application named "Node RED DXQJC 2022-11-02". The page is divided into several sections:

- Details:** Shows the App URL (You must deploy your app first), Source (Download code), Resource group (Default), Deployment target (You must deploy your app first), and Created (11/2/2022).
- Deployment Automation:** A card with a "Deploy your app" button.
- Configuring Continuous Delivery:** A section explaining how to enable Continuous Delivery for the app.
- Getting started quickly:** A sidebar with steps to connect services and view deployment status.
- Services:** A section for managing services, currently showing "Cloudant" with options to open dashboard, documentation, API reference, and credentials.

The bottom of the screen shows a Windows taskbar with various icons and system status information.

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

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** jbmfyhfuvvmyqrrymgrbnfcumphsw

Cluster region Frankfurt **Cluster resource group** Default **Cluster namespace** default **Cluster name** mycluster-free

Deployment type  26°C Rain off and on

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

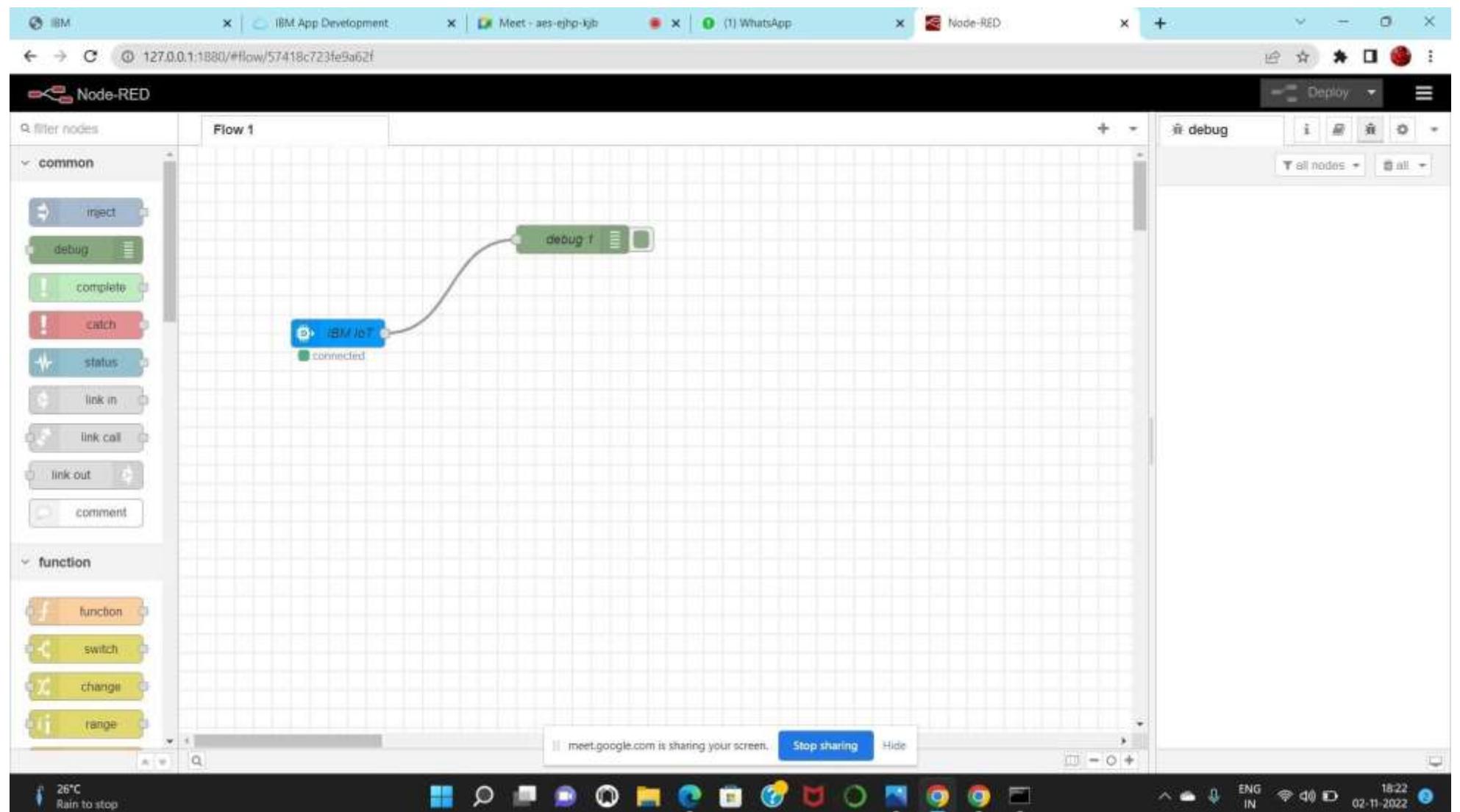
- One free Kubernetes cluster is available per account.
- If you don't have an available cluster, you must create one before continuing. Allow 10-20 minutes for the cluster to be provisioned. [Create cluster](#).

Steps

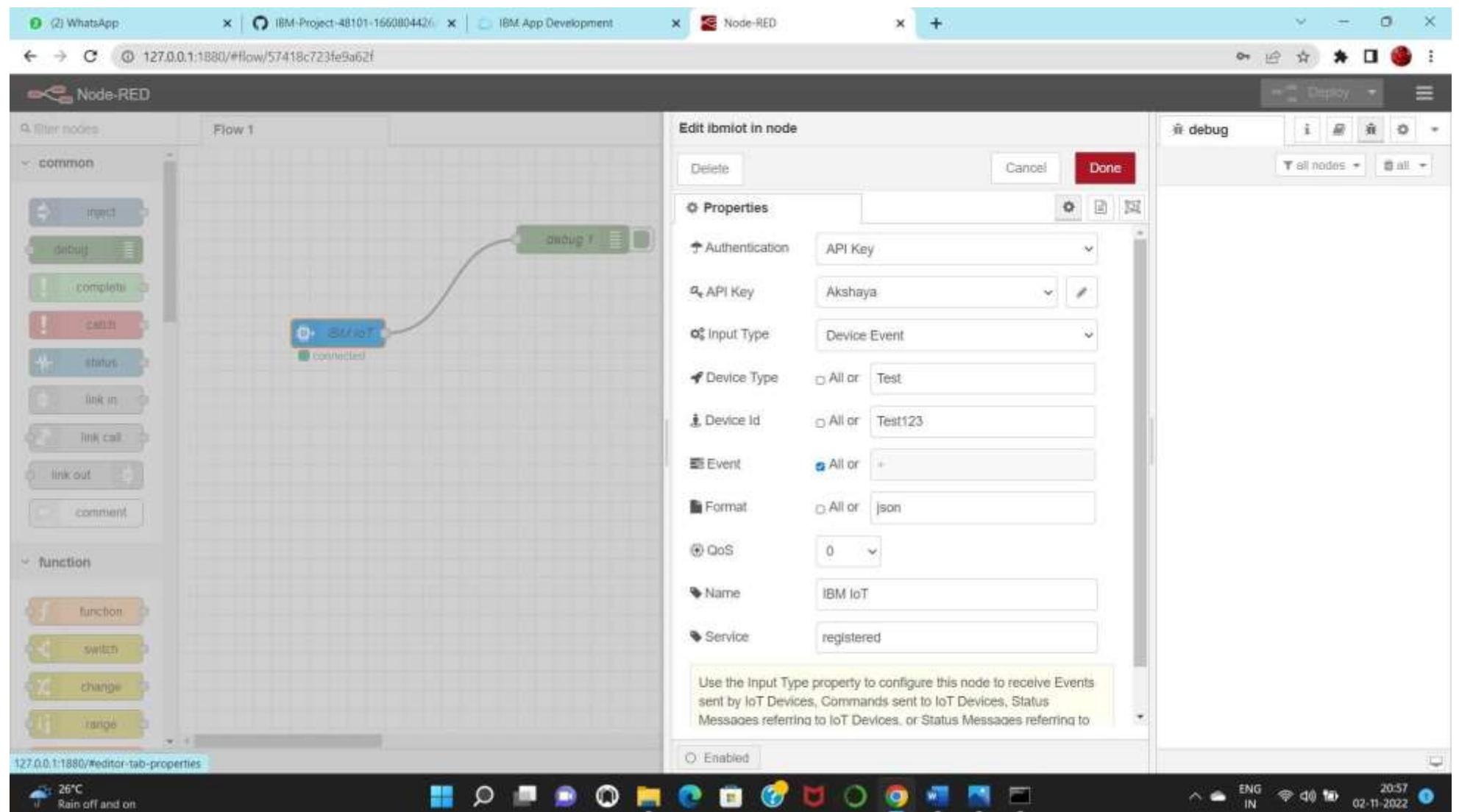
- Create an IBM Cloud API key, or select an existing one from a secrets store.
- Select the container registry region.
- Enter the container registry namespace if it is not already completed.
- Select the region where your Kubernetes cluster is located.
- Select the resource group, cluster namespace, and the cluster name.

ENG IN 20:47 02-11-2022

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



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

Node-RED

Flow 1

The screenshot shows the Node-RED interface with a single flow named "Flow 1". The flow starts with an "IBM IoT" node, which has a green "connected" status indicator. A single output wire from this node connects to a "debug" node. The "debug" node has the identifier "debug 1" and contains a single line of JSON payload: "temperature: 103, humidity: 31". To the right of the canvas is a "debug" sidebar window displaying a log of messages. The log shows several entries, all of which are identical to the payload sent by the "debug" node: "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 }". The sidebar also includes a dropdown menu for "all nodes" and a "clear" button.

Flow 1

IBM IoT

connected

debug 1

debug

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
iot-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
iot-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 }

