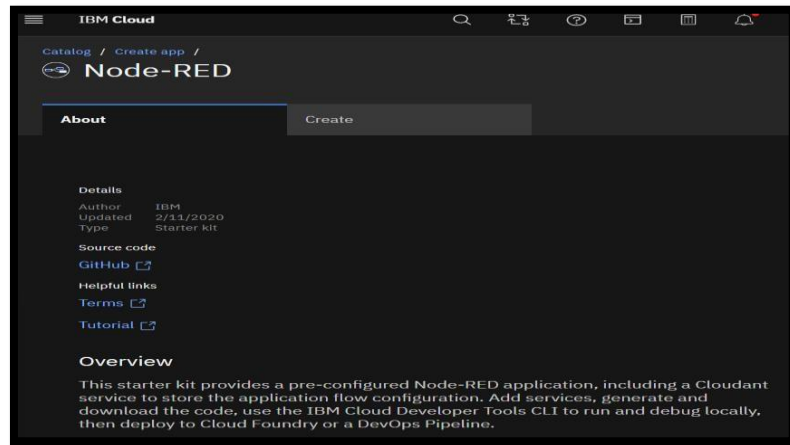


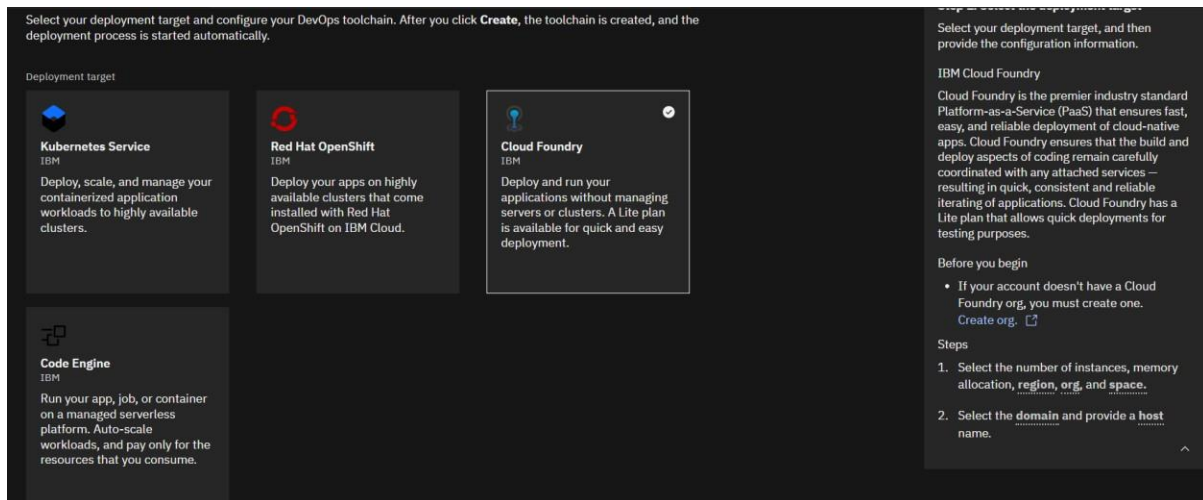
TO CREATE AND CONFIGURE NODE-RED

Team ID	PNT2022TMID12811
Project Name	Project – Industry Specific Intelligent Fire Management system

1) Access the Node-Red using IBM Watson Cloud Platform under Catalog.






2) Choose Cloud Foundry Services.



TO CREATE AND CONFIGURE NODE-RED

3) Generate a IBM Cloud API Key


IBM Cloud API key

.....    [New](#) +

Number of instances

1

Memory allocation per instance

64 MB  2000 MB 256

Region Organization Space

London PNT2022TMID53651 53651

Host Domain

node-red-werht-2022-11-11 eu-gb.mybluemix.net

[Cancel](#) [Next](#)

4) A Node-Red link gets generated.

Resource list / App details /

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

Details

App URL <https://node-red-werht-2022-11-11.eu-gb.mybluemix.net>


Source <https://eu-gb.git.cloud.ibm.com/2019ec0331/NodeREDWERH...>

Resource group [Default](#)

Deployment target [Node RED WERHT 2022-11-11](#)

Created 11/11/2022

Services

 **Cloudant**

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




Credentials ▾

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

Deployment Automation


Name [NodeREDWERHT2022-11-11](#)

Location London


Tool integrations   

Delivery Pipelines

Name [pr-pipeline](#)

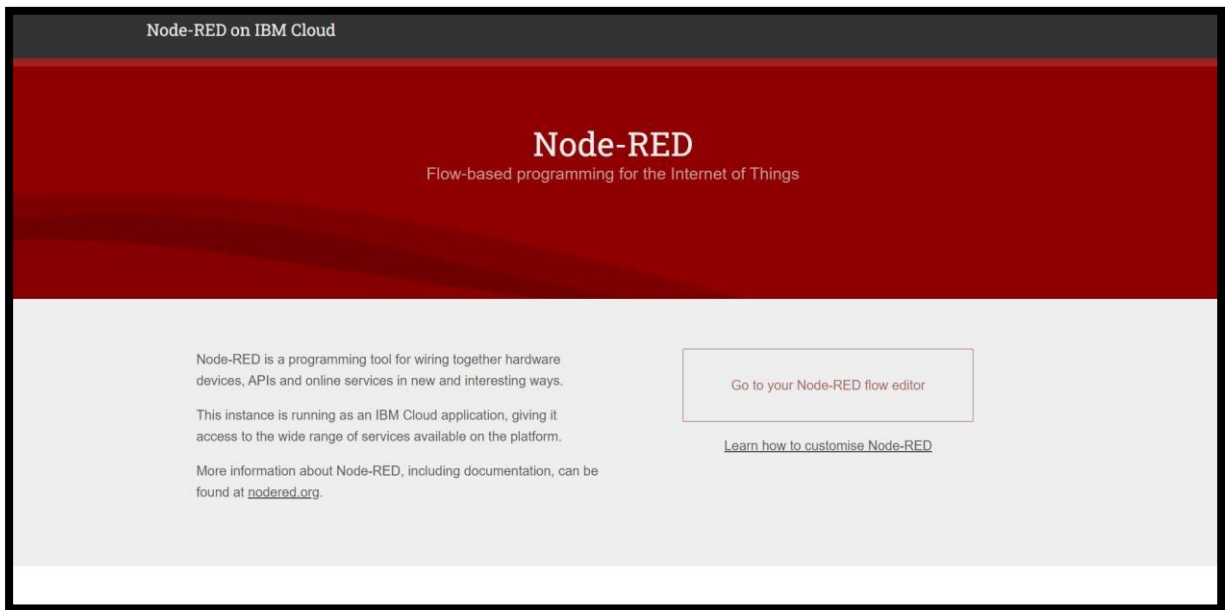
Status  No stages detected

Name [ci-pipeline](#)

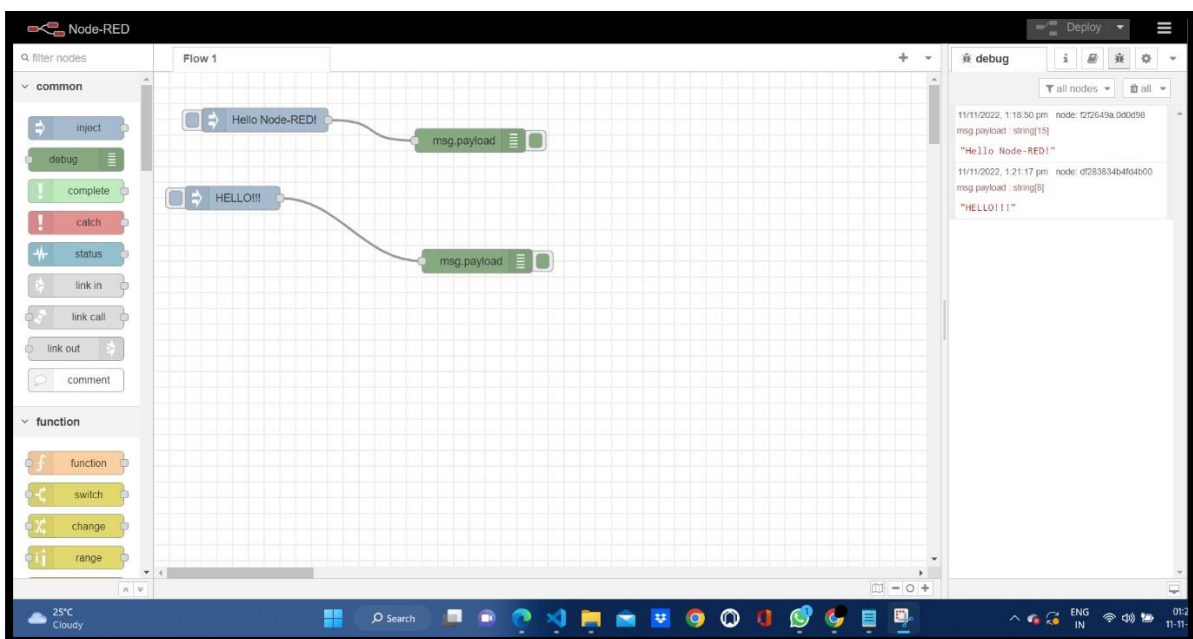
Status  Success

TO CREATE AND CONFIGURE NODE-RED

5) Using that link, access the Node-RED flow Editor

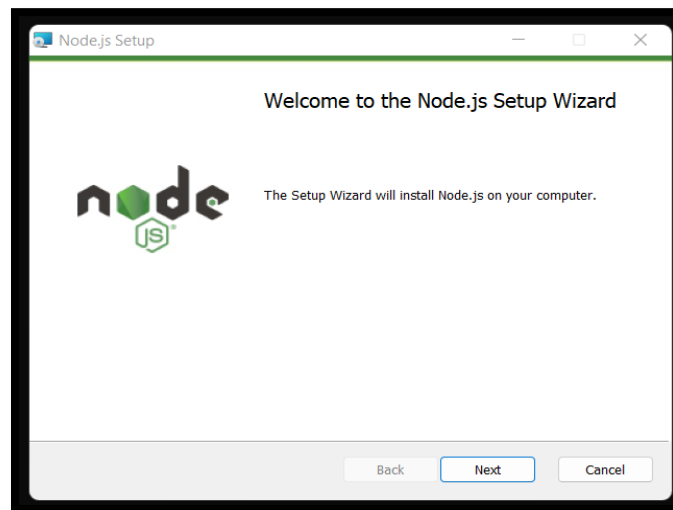


6) Using that editor deploy Nodes accordingly.



TO CREATE AND CONFIGURE NODE-RED

NODE-RED can also be accessed by installing the application.



7) Install node-red by using **node-red** command in command prompt.

```
added 54 packages, and audited 55 packages in 5s

found 0 vulnerabilities

C:\Users\sujitha>node-red
[11 Nov 14:10:01] - [info]
welcome to Node-RED
=====
[11 Nov 14:10:01] - [info] Node-RED version: v2.0.2
[11 Nov 14:10:01] - [info] Node.js version: v18.12.1
[11 Nov 14:10:01] - [info] Windows_NT 10.0.22000 x64 LE
[11 Nov 14:10:02] - [info] Loading palette nodes
[11 Nov 14:10:03] - [info] Dashboard version 3.2.0 started at /ui
[11 Nov 14:10:03] - [info] Settings file : C:\Users\sujitha\.node-red\settings.js
[11 Nov 14:10:03] - [info] Context store : 'default' (module=memory)
[11 Nov 14:10:03] - [info] User directory : C:\Users\sujitha\.node-red
[11 Nov 14:10:03] - [warn] Projects disabled : editorTheme.projects.enabled=false
[11 Nov 14:10:03] - [info] Flows file : C:\Users\sujitha\.node-red\flows.json
[11 Nov 14:10:03] - [info] Creating new flow file
[11 Nov 14:10:03] - [warn]

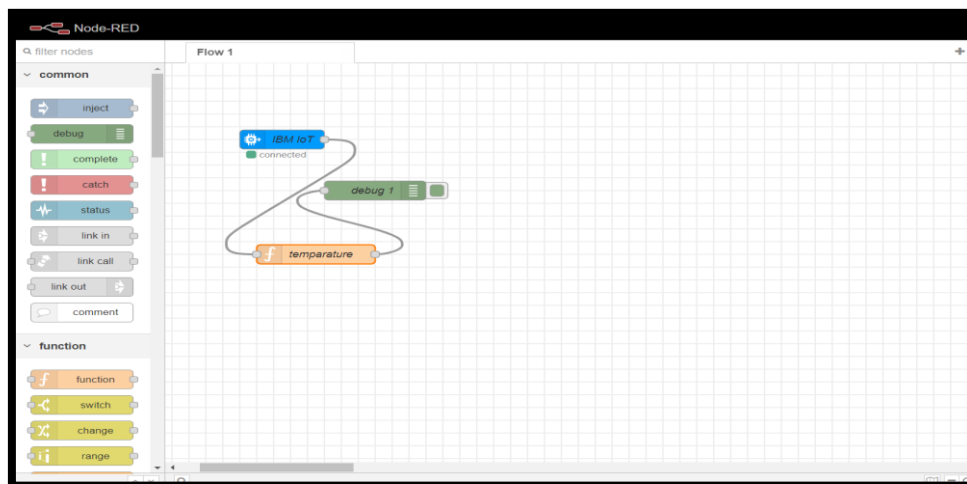
-----
Your flow credentials file is encrypted using a system-generated key.

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 14:10:03] - [info] Server now running at http://127.0.0.1:1880/
[11 Nov 14:10:03] - [warn] Encrypted credentials not found
[11 Nov 14:10:03] - [info] Starting flows
[11 Nov 14:10:03] - [info] Started flows
```

8) Using localhost, access Node-RED



TO CREATE AND CONFIGURE NODE-RED

9) Use Python coding to connect to IBM Cloud and Node-Red

```
ibm_weather_monitoring.py
34
35 except Exception as e:
36     print("Caught exception connecting device: %s" % str(e))
37     sys.exit()
38
39 # Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
40 deviceCli.connect()
41
42 while True:
43     #Get Sensor Data from DHT11
44
45     data={"d":{"temperature":temp,'pulse':pulse,'oxygen':oxygen,'latitude':latitude,'longitude':longitude}}
46     #print data
47
48     def myOnPublishCallBack():
49         print("Published Temperature = %s C" %temp , "Humidity = %s %" %pulse)
50
51     success=deviceCli.publishEvent("IotSensor","json",data,qos=0,on_publish=myOnPublishCallBack)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Published Temperature = 28 C Humidity = 3 %
Published Temperature = 28 C Humidity = 3 %
Published Temperature = 28 C Humidity = 3 %
Published Temperature = 28 C Humidity = 3 %
Published Temperature = 28 C Humidity = 3 %
2022-11-11 15:10:11,987 ibmiotf.device.Client INFO Connected successfully: d:j0mda0:Weather_monitoring_Device:weather_today

10) Deploy the Nodes to get output in the Node-Red platform

