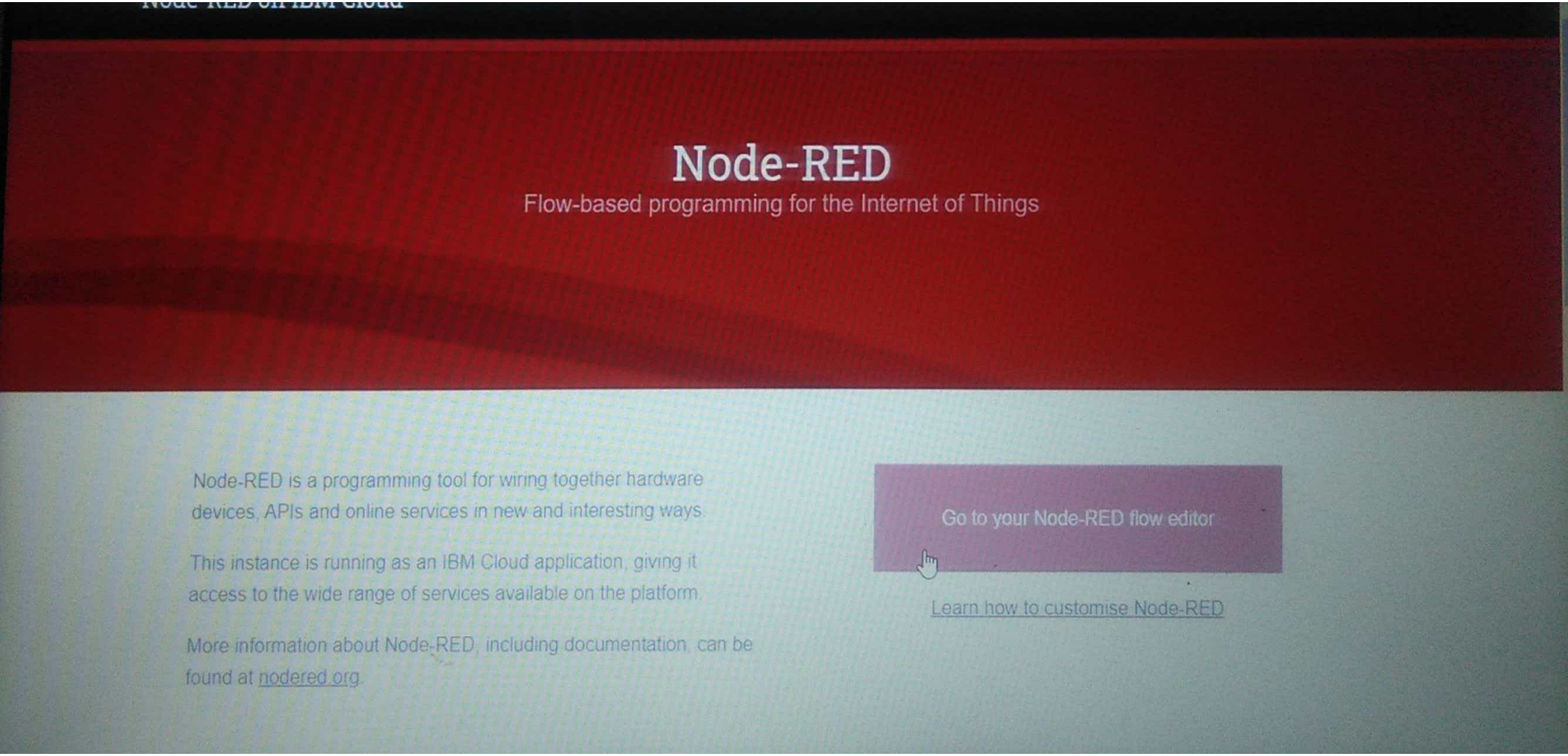

Node-RED on IBM Cloud
Node-RED
Flow-based programming for the Internet of Things
Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.
This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.
More information about Node-RED, including documentation, can be found at nodered.org.
Go to your Node-RED flow editor
Learn how to customise Node-RED

Inbox (5) - suganya.subramania | IBM Watson IoT Platform | Node-RED Dashboard | IBM App Development | Node-RED : node-red-qecix-20

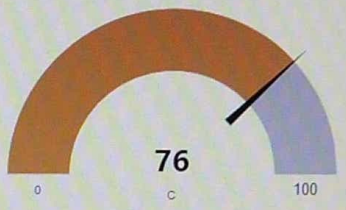
node-red-qecix-2022-11-04.eu-gb.mybluemix.net/ui/#!/0?socketid=gDtDKdk96iTWGuxXAAAH

Reload this page

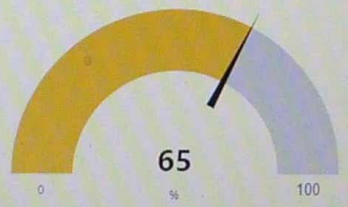
Home

Default

Temperature



Humidity



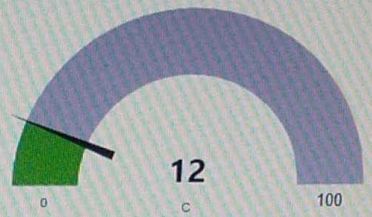
Inbox (5) - suganya.subramani | IBM Watson IoT Platform | Node-RED Dashboard | IBM App Development | Node-RED : node-red-qecix-2(

node-red-qecix-2022-11-04.eu-gb.mybluemix.net

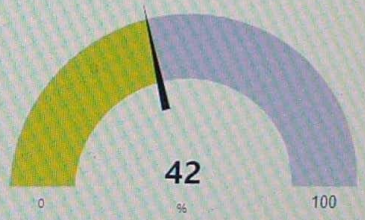
Home

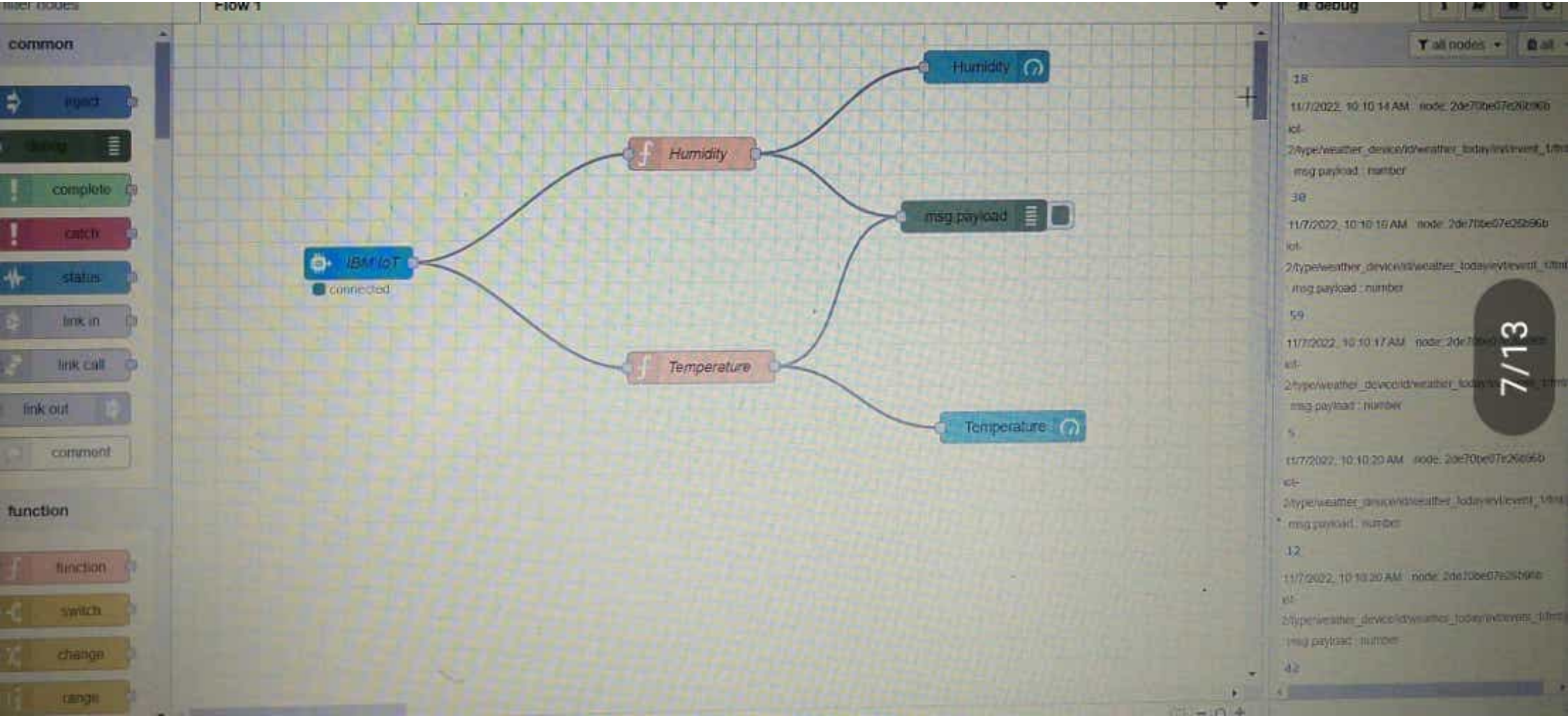
Default

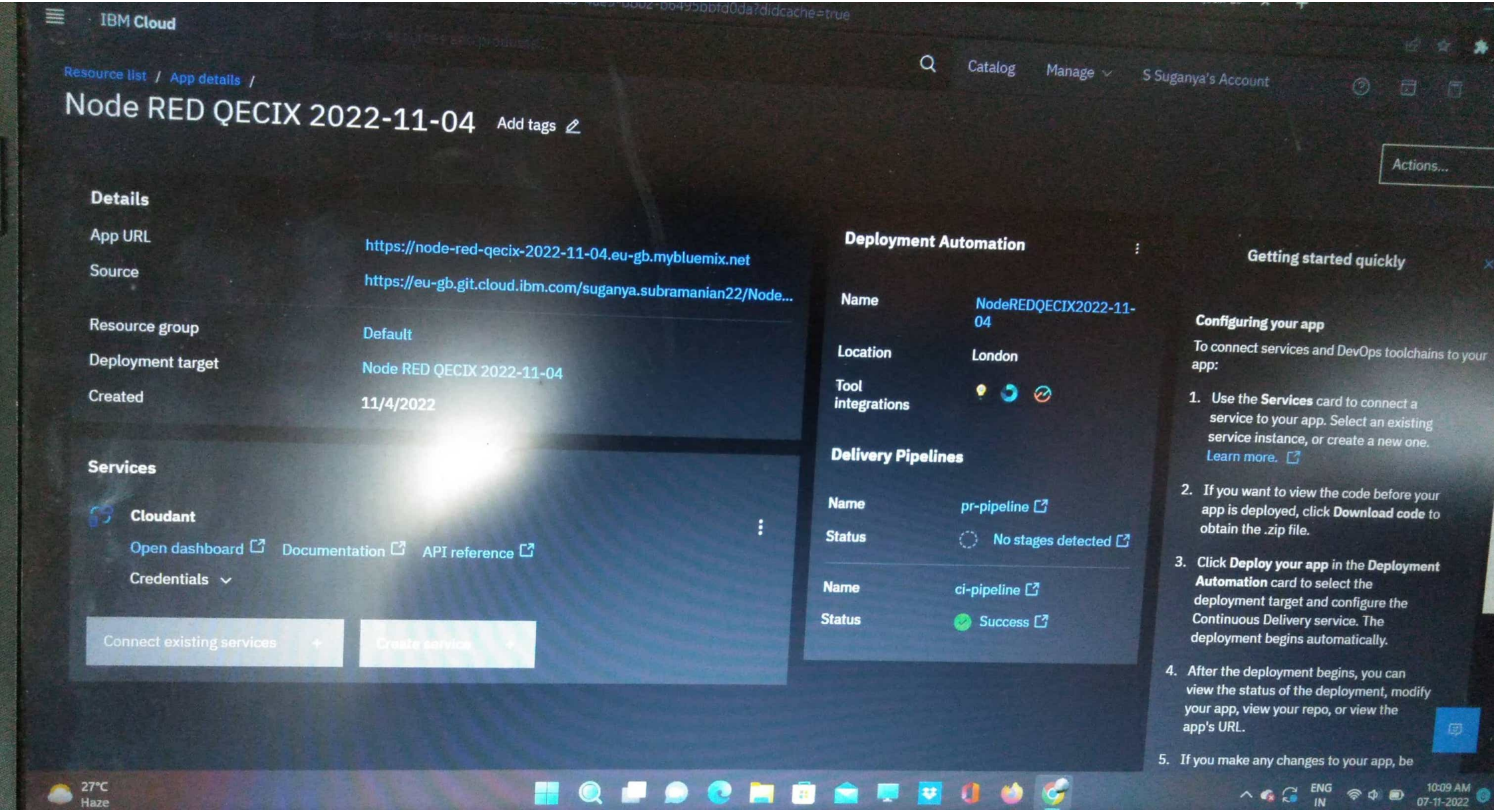
Temperature



Humidity




common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
change
range
Flow 1
Humidity
Humidity
msg.payload
IBM IoT
connected
Temperature
Temperature
debug
all nodes
18
11/7/2022, 10:10:14 AM node: 2de70be07e26b96b
iot-2/type/weather_device/id/weather_today/evt/event_t/fmt/
msg.payload : number
38
11/7/2022, 10:10:16 AM node: 2de70be07e26b96b
iot-2/type/weather_device/id/weather_today/evt/event_t/fmt/
msg.payload : number
59
11/7/2022, 10:10:17 AM node: 2de70be07e26b96b
iot-2/type/weather_device/id/weather_today/evt/event_t/fmt/
msg.payload : number
5
11/7/2022, 10:10:20 AM node: 2de70be07e26b96b
iot-2/type/weather_device/id/weather_today/evt/event_t/fmt/
msg.payload : number
12
11/7/2022, 10:10:20 AM node: 2de70be07e26b96b
iot-2/type/weather_device/id/weather_today/evt/event_t/fmt/
msg.payload : number
42
7/13


IBM Cloud
Catalog
Manage
S Suganya's Account
Resource list / App details /
Node RED QECIX 2022-11-04
Add tags
Actions...
Details
App URL
https://node-red-qecix-2022-11-04.eu-gb.mybluemix.net
Source
https://eu-gb.git.cloud.ibm.com/suganya.subramanian22/Node...
Resource group
Default
Deployment target
Node RED QECIX 2022-11-04
Created
11/4/2022
Services
Cloudant
Open dashboard
Documentation
API reference
Credentials
Connect existing services
Create service
Deployment Automation
Name
NodeREDQECIX2022-11-04
Location
London
Tool integrations
Delivery Pipelines
Name
pr-pipeline
Status
No stages detected
Name
ci-pipeline
Status
Success
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
27°C
Haze
ENG
IN
10:09 AM
07-11-2022

```
File  Edit  Format  Run  Options  Window  Help
rganization = "nym2po"
eviceType = "weather_device"
eviceId = "weather_today"
uthMethod = "token"
uthToken = "4c9LEolzH!97!juTxI"
Initialize GPIO
emp=random.randint(0,100)
ulse=random.randint(0,100)
xygen=random.randint(0,100)
at= 17
on=18


ef myCommandCallback(cmd):
    print("Command received: %s" % cmd.data[ 'command'])
    print(cmd)



ry:
      deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,"ayth-method": authMethod,"auth-token":authtoken}
      deviceCli  = ibmiotf.device.Client(deviceOptions)
      #.....................................................

xcept Exception as e:
       print("Caught exception connecting device: %s" & str(e))
       sys.exit()
Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
eviceCli.connect()

hile True:
      #Get Sensor Data from DHT11
      data = {"d":{ 'temp' : temp, 'pulse' : pulse , 'oxygen' : oxygen,"lat":lat,"lon":lon}}
      #print data
      def myonPublishCallback():
          print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % pulse, "to IBM Watson")
      success = deviceCli.publishEvent("ICTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
      if not success:
          print("Not connected to IoTz")
      time.sleep(1)

      deviceCli.commandCallback = myCommandCallback

 Disconnect the device and application from the cloud
eviceCli.disconnect()
```

```
e  Edit  Format  Run  Options  Window  Help
port time
port sys
port ibmiotf.application
port ibmiotf.device
port random


rovide your IBM Watson Device Credentials
ganization = "nym2po"
viceType = "weather_device"
viceId = "weather_today"
thMethod = "token"
thToken = "4c9LEolzH!97!juTxI"
nitialize GPIO
mp=random.randint(0,100)
lse=random.randint(0,100)
ygen=random.randint(0,100)
t= 17
n=18


f myCommandCallback(cmd):
   print("Command received: %s" % cmd.data[ 'command'])
   print(cmd)



y:
      deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,"auth-method": authMethod,"auth-token":authtoken}
      deviceCli  = ibmiotf.device.Client(deviceOptions)
      #.....................................................

cept Exception as e:
      print("Caught exception connecting device: %s" & str(e))
      sys.exit()
onnect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
viceCli.connect()

le= True:
      #Get Sensor Data from DHT11
      data = {"d":{ 'temp' : temp, 'pulse' : pulse , 'oxygen' : oxygen,"lat":lat,"lon":lon}}
      #print data
      def myOnPublishCallback():
          print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % pulse, "to IBM Watson")
      success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
      if not success:
          print("Not connected to IoTF")
```

Browse Action Device Types Interfaces

Browse Devices

All Devices | Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added
weather_device_1	Connected	weather_device	Device	7 Nov 2022 09:49
weather_today	Disconnected	weather_device	Device	14 Oct 2022 10:10

Items per page 50 | 1–2 of 2 items

1 of 1 page 1

2 Simulations running

Browse Action Device Types Interfaces

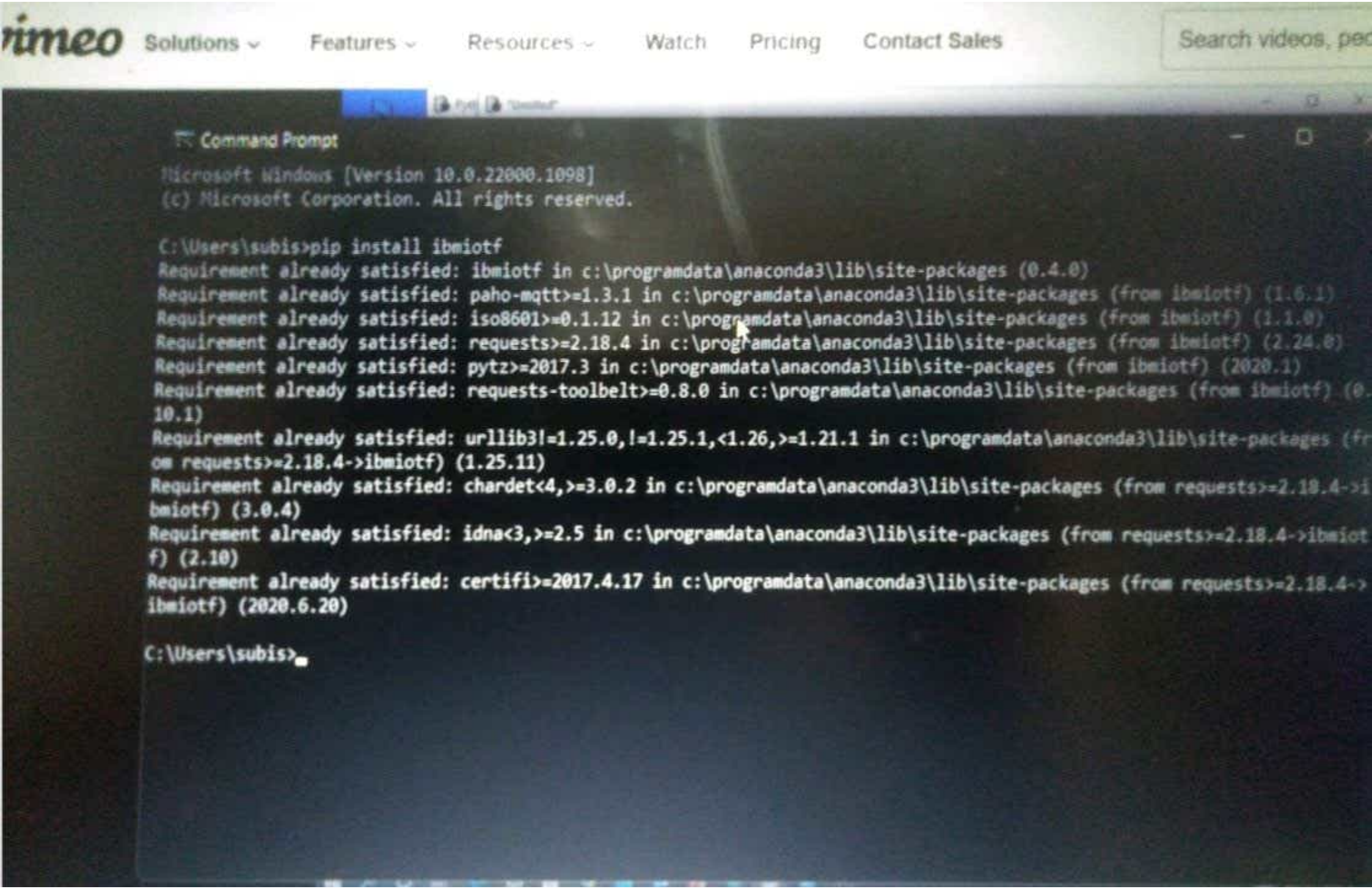
Device ID	Status	Device Type	Class ID	Date Added
weather_device_1	Connected	weather_device	Device	7 Nov 2022 09:49

Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_1	{"temperture":40,"humidity":37}	json	a few seconds ago
event_1	{"temperture":74,"humidity":77}	json	a few seconds ago
event_1	{"temperture":12,"humidity":3}	json	a few seconds ago
event_1	{"temperture":19,"humidity":30}	json	a few seconds ago
event_1	{"temperture":3,"humidity":56}	json	a few seconds ago

2 Simulations running


rimeo
Solutions
Features
Resources
Watch
Pricing
Contact Sales
Search videos, pe
Command Prompt
Microsoft Windows [Version 10.0.22000.1098]
(c) Microsoft Corporation. All rights reserved.
C:\Users\subis>pip install ibmiotf
Requirement already satisfied: ibmiotf in c:\programdata\anaconda3\lib\site-packages (0.4.0)
Requirement already satisfied: paho-mqtt>=1.3.1 in c:\programdata\anaconda3\lib\site-packages (from ibmiotf) (1.6.1)
Requirement already satisfied: iso8601>=0.1.12 in c:\programdata\anaconda3\lib\site-packages (from ibmiotf) (1.1.0)
Requirement already satisfied: requests>=2.18.4 in c:\programdata\anaconda3\lib\site-packages (from ibmiotf) (2.24.0)
Requirement already satisfied: pytz>=2017.3 in c:\programdata\anaconda3\lib\site-packages (from ibmiotf) (2020.1)
Requirement already satisfied: requests-toolbelt>=0.8.0 in c:\programdata\anaconda3\lib\site-packages (from ibmiotf) (0.
10.1)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (fr
om requests>=2.18.4->ibmiotf) (1.25.11)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\programdata\anaconda3\lib\site-packages (from requests>=2.18.4->i
bmiotf) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests>=2.18.4->ibmiot
f) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests>=2.18.4->
ibmiotf) (2020.6.20)
C:\Users\subis>