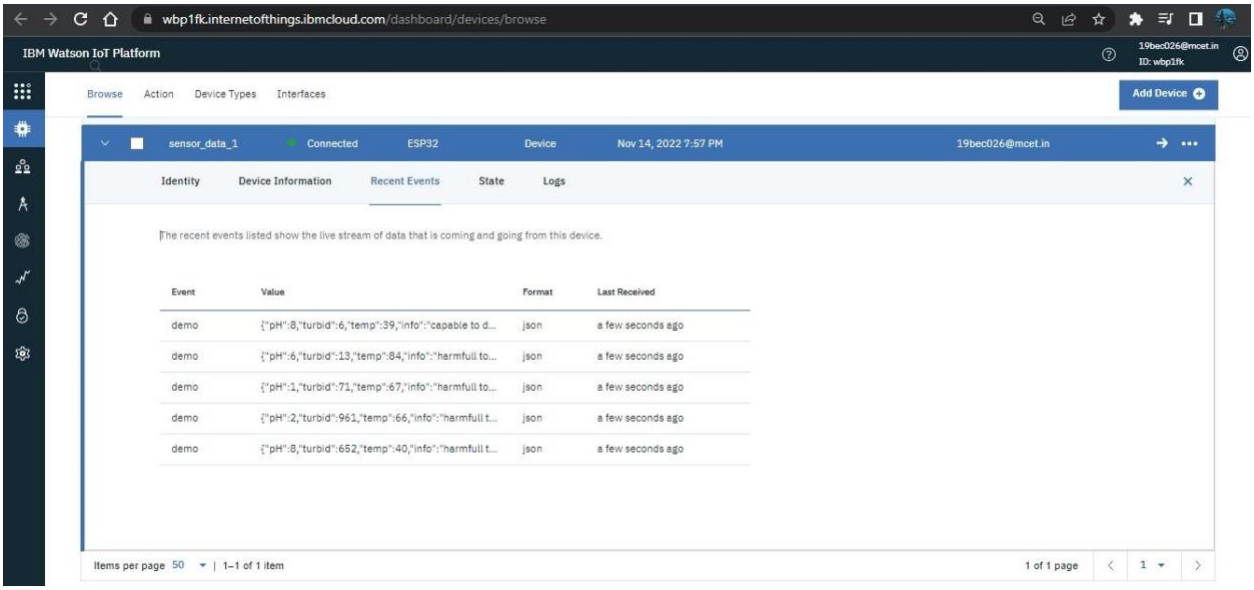
Develop a python script

Team ID	PNT2022TMID08766
Project Name	Project - Real time River water quality monitoring and control system

Published data to IBM cloud:



Develop a python program:

```
Import ibmiotf.application

Import ibmiotf.device

Import time

Import random

Import sys

From twilio.rest import Client

Account_sid = 'AC18b4d7a136b9a07a181a837c23ad1358'

Auth_token ='adc9782f6520041c84ac4930daad0625 '

Client = Client(account_sid, auth_token)


Organization = "wbp1fk"

deviceType = "ESP32"
```

```
deviceId = “sensor_data_1”
authMethod = “token”
authToken = “prototype_1”

pH = random.randint(1, 14)
turbidity = random.randint(1, 1000)
temperature = random.randint(0, 100)
info=””

def myCommandCallback(cmd):
   print(“Command Received: %s” % cmd.data[‘command’])
   print(cmd)



try:
   deviceOptions={“org”:organization,”type”:deviceType,
            “id”:deviceId,”auth-method”:authMethod,”auth-token”:authToken}
   deviceCli = ibmiotf.device.Client(deviceOptions)

except Exception as e:
   print(“caught exception connecting device: %s” % str(e))
   sys.exit()

deviceCli.connect()

while True:

   pH = random.randint(1, 14)
   turbidity = random.randint(1, 1000)
```

```
    temperature = random.randint(0, 100)

    if temperature>70 or pH<6 or pH>8 or turbidity>500:
        print(“high”)
        info=”harmfull to drink”
        message = client.messages.create(from_=’+14632588702’,
                                         body =’This water is harmfull to drink’,
                                         to =’+91 95856 17613’)
    else:
        info=”capable to drinking”
        message = client.messages.create(from_=’+14632588702’,
                                         body =’This water is good to drink’,
                                         to =’+91 95856 17613’)
    data = {‘pH’: pH, ‘turbid’: turbidity,’temp’: temperature,’info’:info}


    def myOnPublishCallback():
        print(“Published pH= %s” % pH, “Turbidity:%s” % turbidity,
              “Temperature:%s” % temperature)

    Success = deviceCli.publishEvent(“demo”, “json”, data, qos=0, on_publish=myOnPublishCallback)
    If not success:
        print("Not Connected to ibmiot")
    time.sleep(5)
    deviceCli.commandCallback = myCommandCallback

deviceCli.disconnect()
```

```
========== RESTART: C:\Users\srihath\Desktop\IBM\final.py ==========
high2022-11-14 20:03:53,055   ibmiotf.device.Client      INFO    Connected succe
ssfully: d:wbp1fk:ESP32:sensor_data_1

Published pH= 9 Turbidity:595 Temperature:24
high
Published pH= 10 Turbidity:259 Temperature:98
Published pH= 14 Turbidity:163 Temperature:59
high
Published pH= 1 Turbidity:109 Temperature:56
high
Published pH= 8 Turbidity:527 Temperature:7
high
Published pH= 11 Turbidity:874 Temperature:62
Published pH= 9 Turbidity:76 Temperature:40
high
Published pH= 12 Turbidity:478 Temperature:91
high
Published pH= 7 Turbidity:887 Temperature:54
Published pH= 13 Turbidity:18 Temperature:64
Published pH= 13 Turbidity:219 Temperature:47
high
Published pH= 10 Turbidity:764 Temperature:36
high
Published pH= 11 Turbidity:545 Temperature:88
```