

FINAL CODE

Program:

```
import random
import time
import sys
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "xfptfb"
deviceType = "NodeMCU"
deviceId = "19141"
authMethod = "use-token-auth"
authToken = "1914137383010209"

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("Light on")
    else :
        print ("Light off")

    #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()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11
    pH=random.randint(0,100)
    conductivity=random.randint(0,100)
    T=random.randint(0,100)
    oxygen=random.randint(0,100)
    turbidity=random.randint(0,100)

    data = { 'temperature' : T, 'pH': pH, 'conductivity':
conductivity, 'oxygen':oxygen,'turbidity' :turbidity }
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "ph = %s %" % ph,"turbidity = %s NTU " % turb ,"to IBM
Watson")

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoTf")
        time.sleep(1)
        deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()

```