```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import ibmiotf.api
import requests
import json
#Provide your IBM Watson Device Credentials
organization = "lryrya"
deviceType = "Temp"
deviceId = "T1"
authMethod = "token"
authToken = "123456789"
temp=0
alert=0
def Alert():
    url = "https://www.fast2sms.com/dev/bulkV2"
    my_data = {
            'sender_id': 'FSTSMS',
            'message': 'Alert Fire has been detected in XYZ industry! Kindly Evacuvate from the
place',
            'language': 'english',
            'route': 'p',
            'numbers': '9095057479'
     }
    headers = {
           'authorization':
'o0azwVFNHOM5B3hrRxdenyU2cfZujqSpYEX7t8LAgJPb9kliWCugDvo1n0kcY8TGHOt3dIQws
KpLbAJU',
           'Content-Type': "application/x-www-form-urlencoded",
           'Cache-Control': "no-cache"
     }
    response = requests.request("POST",
                                            url,
                                            data = my_data,
                                            headers = headers)
    returned_msg = json.loads(response.text)
    print(returned_msg['message'])

def myCommandCallback(cmd):
  print("Command received: %s" % cmd.data['command'])
  status=cmd.data['command']
  if status=="lighton":
        print ("led is on")
    else :
        print ("led is 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:
    if temp<100:
        #Get Sensor Data from DHT11
        temp=temp+1
        data = { 'Temperature' : temp}
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" %temp, "to IBM Watson")
        if(temp>70) and (alert==0):
            Alert();
            time.sleep(1)
            alert=alert+1
        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
    else:
        temp=0;
    if not success:
        print("Not connected to IoTF")
    time.sleep(1)
deviceCli.disconnect()
```