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


#Provide your IBM Watson Device Credentials
organization = "9anun7"
deviceType = "1911104"
deviceId = "1911104-iot"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == "alarmon":
        print ("Alarm is on please all Evacuate Fans On")
    elif status == "alarmoff":
        print ("Alarm is off and Fans Off")
    elif status == "sprinkleron":
```

```
    print ("Sprinkler is On Evacuate Faster")
  elif status == "sprinkleroff":
    print("Sprinkler is Off")
  else:
    print("Please send proper 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()

# 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 random function
```

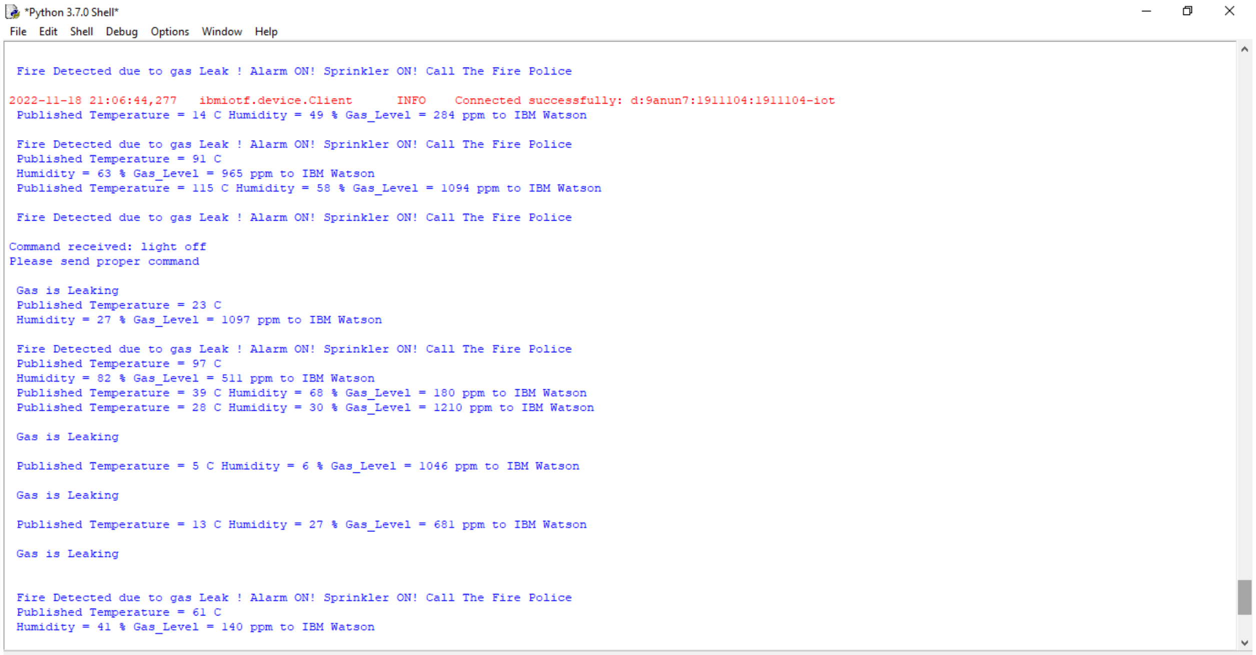
```
        temp=random.randint(0,120)
        Humid=random.randint(0,100)
        gas=random.randint(0,1500)
        data={'temp':temp,'Humid':Humid,'gas':gas}
        #print data
        def myOnPublishCallback():
            print (" Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "Gas_Level = %s ppm" %gas, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("\n Not connected to IoTF")
        if temp>60 :
            print("\n Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police \n")
        elif gas>350:
            print("\n Gas is Leaking \n")

        time.sleep(10)

        deviceCli.commandCallback = myCommandCallback

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



```
 Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police

2022-11-18 21:06:44,277   ibmiotf.device.Client      INFO    Connected successfully: d:9anun7:1911104:1911104-iot
 Published Temperature = 14 C Humidity = 49 % Gas_Level = 284 ppm to IBM Watson

 Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police
 Published Temperature = 91 C
 Humidity = 63 % Gas_Level = 965 ppm to IBM Watson
 Published Temperature = 115 C Humidity = 58 % Gas_Level = 1094 ppm to IBM Watson

 Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police

Command received: light off
Please send proper command

 Gas is Leaking
 Published Temperature = 23 C
 Humidity = 27 % Gas_Level = 1097 ppm to IBM Watson

 Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police
 Published Temperature = 97 C
 Humidity = 82 % Gas_Level = 511 ppm to IBM Watson
 Published Temperature = 39 C Humidity = 68 % Gas_Level = 180 ppm to IBM Watson
 Published Temperature = 28 C Humidity = 30 % Gas_Level = 1210 ppm to IBM Watson

 Gas is Leaking

 Published Temperature = 5 C Humidity = 6 % Gas_Level = 1046 ppm to IBM Watson

 Gas is Leaking

 Published Temperature = 13 C Humidity = 27 % Gas_Level = 681 ppm to IBM Watson

 Gas is Leaking


 Fire Detected due to gas Leak ! Alarm ON! Sprinkler ON! Call The Fire Police
 Published Temperature = 61 C
 Humidity = 41 % Gas_Level = 140 ppm to IBM Watson
```