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

# watson device details

organization = "7tbblb"
devicType =  "Trashbin2"
deviceId = "Trashbinid2"
authMethod= "token"
authToken= "123456789"

#generate random values for randomo variables (temperature&humidity)



def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)

try:
        deviceOptions={"org": organization, "type": devicType,"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:

    ultrasonic= random.randint(10,70)
    loadcell= random.randint(5,15)
    data= {'dist':ultrasonic,'load':loadcell}


    if loadcell < 13 and loadcell > 15:
          load = "90 %"

    elif loadcell < 8 and loadcell > 12:
          load = "60 %"

    elif loadcell < 4 and loadcell > 7:
          load = "40 %"
    else:
          load = "0 %"

    if ultrasonic < 10:
          dist = ' 90 %'


    elif ultrasonic < 20 and ultrasonic >11:
          dist = '60%'

    elif ultrasonic < 60 and ultrasonic > 41:
          dist =  '40 %'
    elif ultrasonic < 80 and ultrasonic > 61:
          dist =  '20 %'




    if load == "90 %" or ultrasonic == "90 %":
          warn  = 'alert :' ' Dumpster poundage getting high, Time to collect :)'

    elif load == "60 %" or ultrasonic == "60 %":

          warn = 'alert :' 'dumpster is above 60%'
    else :
          warn = 'alert :' 'No need to collect right now '
    def myOnPublishCallback(lat=10.678991,long=78.177731):
        print("")
        print("published distance = %s " %ultrasonic,"loadcell:%s " %loadcell,"lon = %s " %long,"lat
= %s" %lat)
```

```
        print(load)
        print(dist)
        print(warn)

    time.sleep(5)


    success=deviceCli.publishEvent ("IoTSensor","json",warn,qos=0,on_publish= myOnPublishCallback)

    success=deviceCli.publishEvent ("IoTSensor","json",data,qos=0,on_publish= myOnPublishCallback)



    if not success:
        print("not connected to ibmiot")
    time.sleep(5)




    deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect()
```