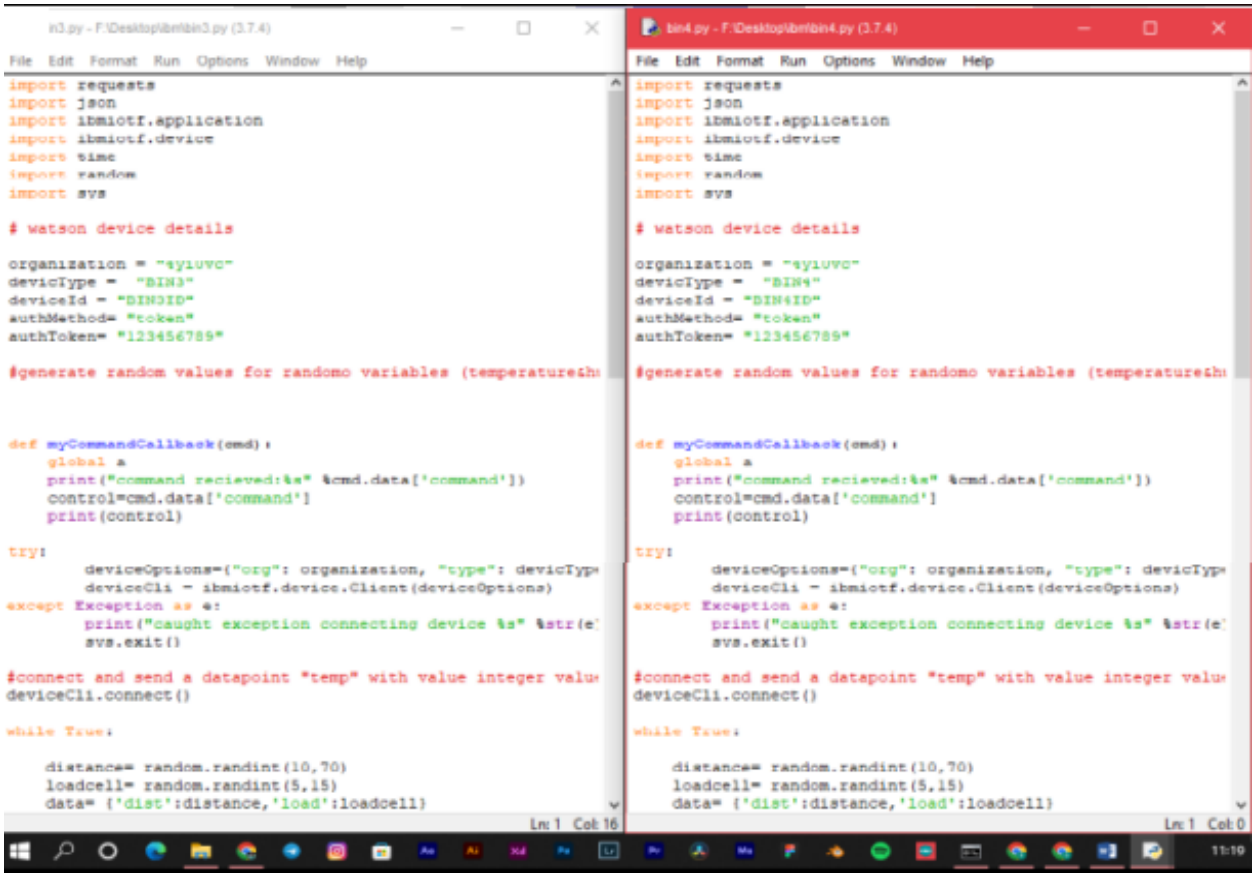
Team ID	PNT2022TMID29903
Project Name	SmartWaste Management for Metropolitan cities

Python script:



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

# watson device details

organization = "sy1uvc"
devicType =  "BIN3"
deviceId = "BIN3ID"
authMethod= "token"
authToken= "123456789"

#generate random values for randomo variables (temperature&h

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

try:
        deviceOptions={"org": organization, "type": devicTyp
        deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
        print("caught exception connecting device %s" %str(e
        sys.exit()

#connect and send a datapoint "temp" with value integer valu
deviceCli.connect()

while True:

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

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

# watson device details

organization = "sy1uvc"
devicType =  "BIN4"
deviceId = "BIN4ID"
authMethod= "token"
authToken= "123456789"

#generate random values for randomo variables (temperature&h

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

try:
        deviceOptions={"org": organization, "type": devicTyp
        deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
        print("caught exception connecting device %s" %str(e
        sys.exit()

#connect and send a datapoint "temp" with value integer valu
deviceCli.connect()

while True:

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

bin2.py - F:\Desktop\ibm\bin2.py (3.7.4)

File Edit Format Run Options Window Help

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

# watson device details

organization = "4yi0vc"
devicType =  "BIN2"
deviceId = "BIN2ID"
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)
```

```
#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":a
        deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
        print("caught exception connecting device %s" %str(e))
        sys.exit()

#connect and send a datapoint "temp" with value integer value into the cloud as a type of event for every 10 seconds
deviceCli.connect()

while True:

    distance= random.randint(10,70)
    loadcell= random.randint(5,15)
    data= {'dist':distance,'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 distance < 15:
          dist = 'Risk warning:' 'Dumpster poundage getting high, Time to collect :) 90 %'


    elif distance < 40 and distance >16:
          dist = 'Risk warning:' 'dumpster is above 60%'
```