# PYTHON SCRIPT

| | |
|---|---|
| TEAM ID | PNT2022TMID32983 |
| PROJECT NAME | SMART SOLUTIONS FOR RAILWAYS |

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



#Provide your IBM Watson Device Credentials
organization = "lfkvn6"
deviceType = "SOLUTION"
deviceId = "SOLUTION_1"
authMethod = "token"
authToken = "12345678"


# Initialize GPIO



def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="Emergency Alert":
```

```
        print ("Emergency Alert")


    #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


        trainnumber=random.randint(10000,20000)
        lat=random.randfloat(10,11)
        lon=random.randfloat(77,78)
```

```
        data = { 'trainnumber' : trainnumber, 'lat': lat,'lon': lon}
        #print data
        def myOnPublishCallback():
            print ("Published trainnumber = %s 'C" % trainnumber, "lat = %s %%" %
lat,"lon = %s %%" % lon, "to IBM Watson")


        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(10)


        deviceCli.commandCallback = myCommandCallback


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