# HAZARDOUS AREA MONITORING FOR INDUSTRIAL PLANT POWERED BY IOT

Team id: PNT2022TMID36143

CODE:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "t59c74"
deviceType = "team"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"
def myCommandCallback(cmd):
    print ("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status== "sensoron":
        print ("sensor is on")
    elif status == "sensoroff":
        print ("sensor is off")
    else:
        print ("please send proper command")

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 evention connecting device: %s" % str(e))
      sys.exit()


deviceCli.connect()
while True:
      temp=random.randint (90,110)
      Humid=random.randint (60,100)
      Air_quality=random.randint (0,1000)
      data = {'temp': temp,'Humid': Humid,'Air_quality': Air_quality}
      def myonPublishCallback():
          print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid,"Air_quality = %s
  %%" % Air_quality, "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
  deviceCli.disconnect()
```