SPRINT-2

TEAM ID: PNT2022TMID54441

PROJECT NAME : IoT based smart crop protection system for Agriculture

STEPS:

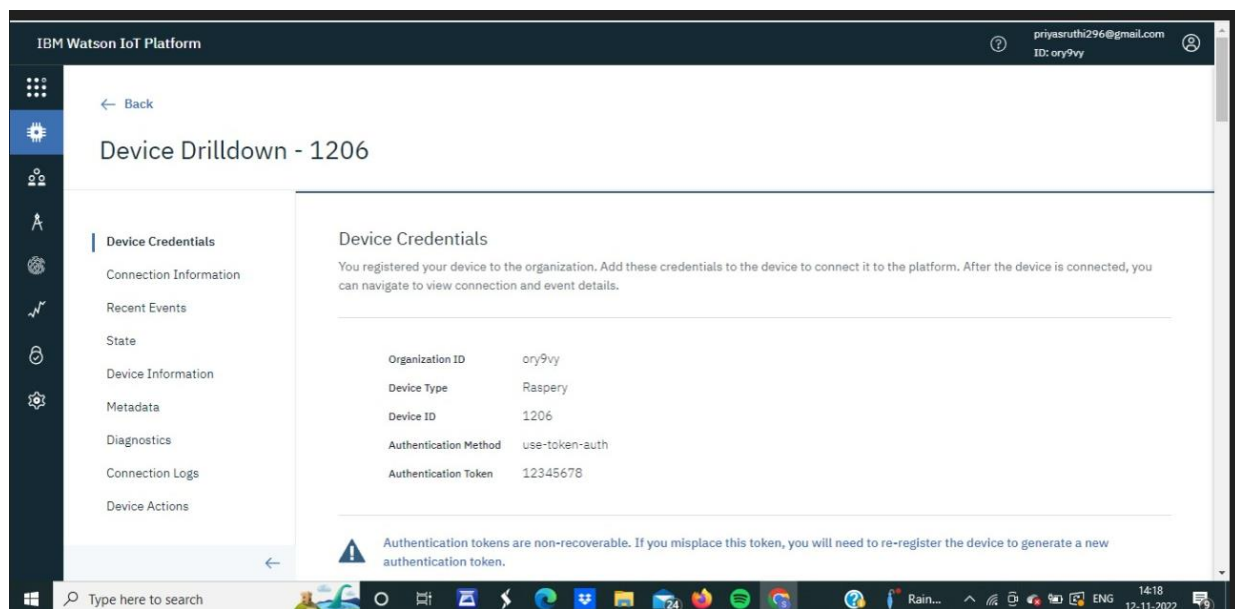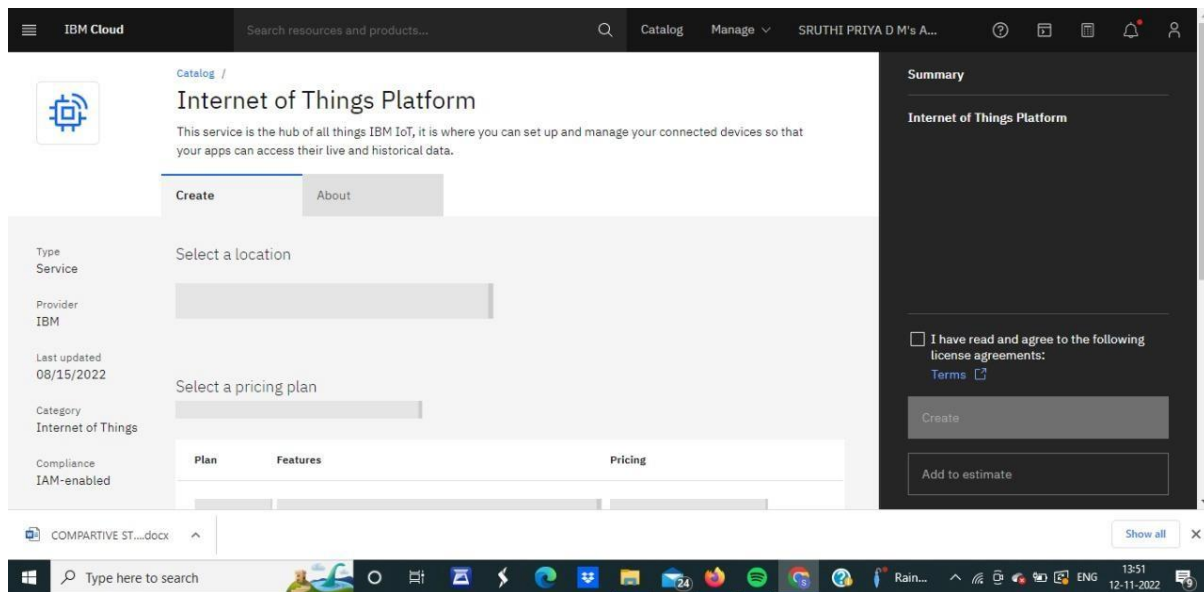The random sensor data's are generated and automation has been implemented through the python code instead of using hardware to implement IOT based crop protection system. And the python code need to upload the data's in IBM cloud are written in this python script.

CREATION OF IBM WATSON CLOUD PLATFROM:
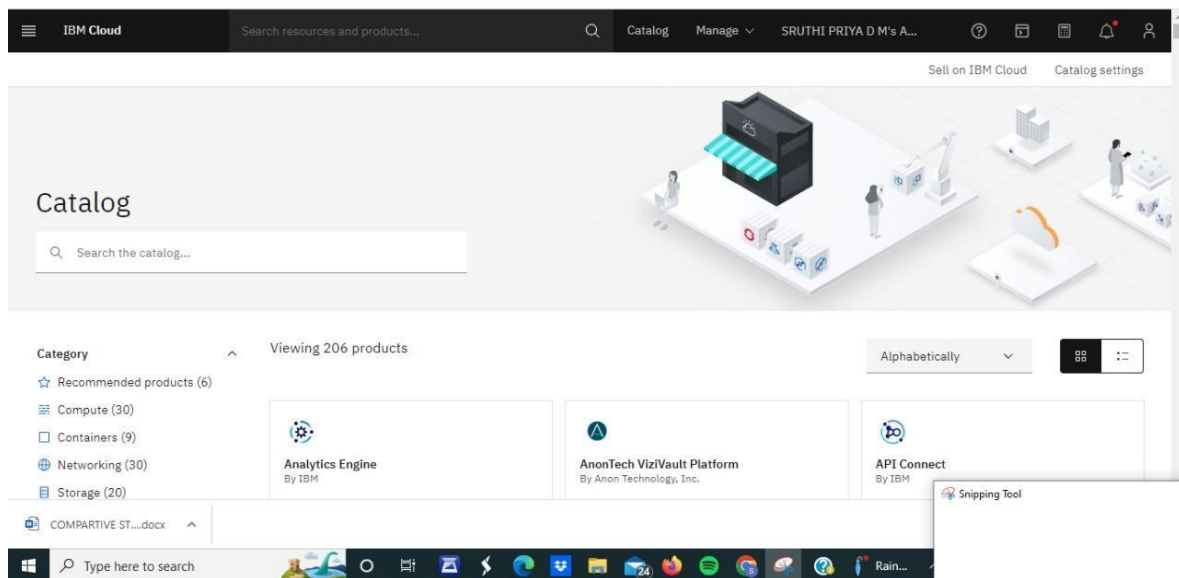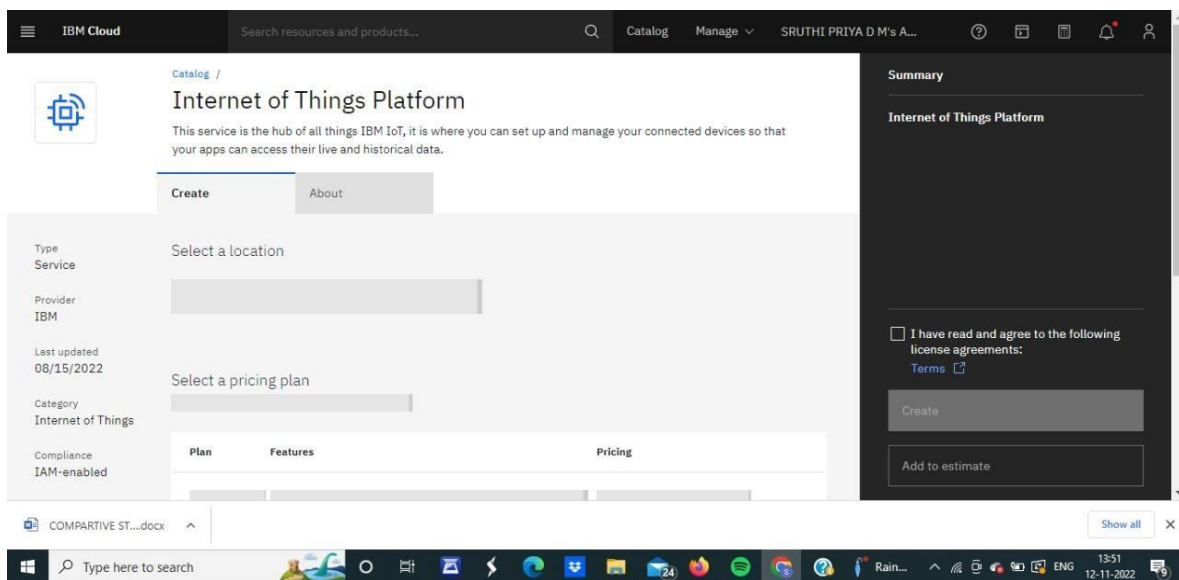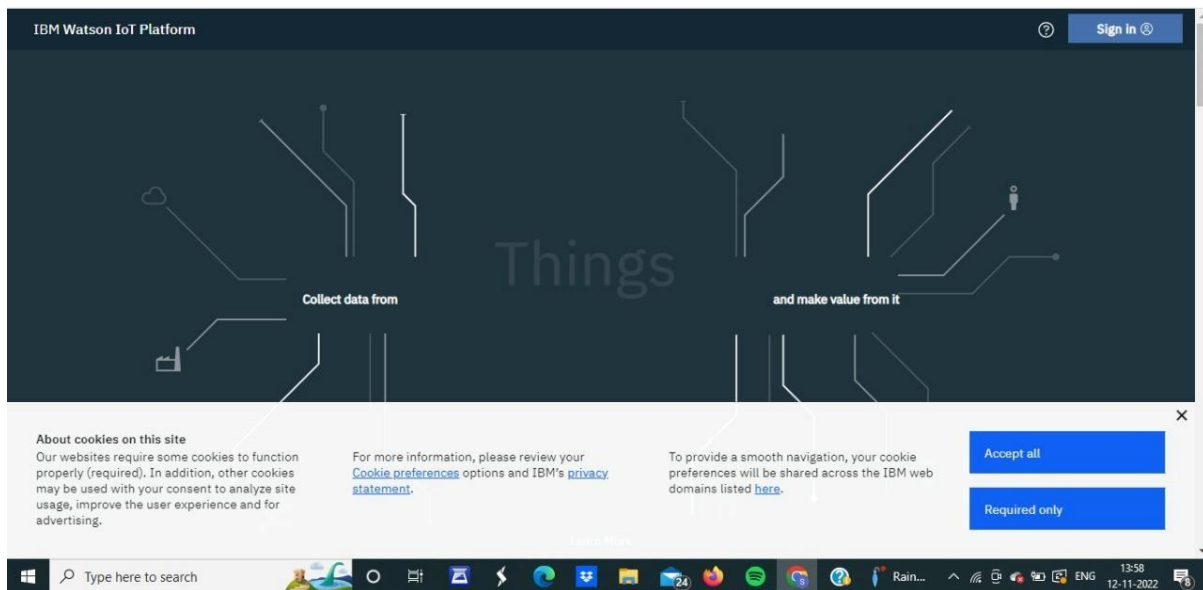
CODE:

```
import random
import ibmiotf.application
import ibmiotf.device
from time import sleep
```

```python
import sys
#IBM Watson Device Credentials.

organization = "ory9vy"

deviceType = "Raspery"

deviceId = "1206"

authMethod = "token"

authToken = "12345678"
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="sprinkler_on":
        print ("sprinkler is ON")
    else :
        print ("sprinkler is OFF")
#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:
```

```python
    print("Caught exception connecting device: %s" % str(e))
sys.exit()
#Connecting to IBM watson.
deviceCli.connect()
while True:
#Getting values from sensors.
 temp_sensor = round( random.uniform(0,80),2)
 PH_sensor = round(random.uniform(1,14),3)
 camera = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
 camera_reading = random.choice(camera)
 flame = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
 flame_reading = random.choice(flame)
 moist_level = round(random.uniform(0,100),2)
 water_level = round(random.uniform(0,30),2)

#storing the sensor data to send in json format to cloud.

 temp_data = { 'Temperature' : temp_sensor }
 PH_data = { 'PH Level' : PH_sensor }
 camera_data = { 'Animal attack' : camera_reading}
 flame_data = { 'Flame' : flame_reading }
```

```python
    moist_data = { 'Moisture Level' : moist_level}
    water_data = { 'Water Level' : water_level}


# publishing Sensor data to IBM Watson for every 5-10 seconds.
    success = deviceCli.publishEvent("Temperature sensor", "json", temp_data, qos=0)
    sleep(1)
    if success:
        print (" ..........................publish ok. .......................... ")
        print ("Published Temperature = %s C" % temp_sensor, "to IBM Watson")


    success = deviceCli.publishEvent("PH sensor", "json", PH_data, qos=0)
    sleep(1)
    if success:
        print ("Published PH Level = %s" % PH_sensor, "to IBM Watson")


    success = deviceCli.publishEvent("camera", "json", camera_data, qos=0)
    sleep(1)
    if success:
```

```python
    print ("Published Animal attack %s " % camera_reading, "to
IBM Watson")

 success = deviceCli.publishEvent("Flame sensor", "json",
flame_data, qos=0)

 sleep(1)

 if success:

   print ("Published Flame %s " % flame_reading, "to IBM
Watson")


 success = deviceCli.publishEvent("Moisture sensor", "json",
moist_data, qos=0)

 sleep(1)

 if success:

   print ("Published Moisture Level = %s " % moist_level, "to
IBM Watson")


 success = deviceCli.publishEvent("Water sensor", "json",
water_data, qos=0)

 sleep(1)

 if success:

   print ("Published Water Level = %s cm" % water_level, "to
IBM Watson")

   print ("")

 #Automation to control sprinklers by present temperature an
to send alert message to IBM Watson.
```

```python
    if (temp_sensor > 35):
        print("sprinkler-1 is ON")
    success = deviceCli.publishEvent("Alert1", "json",{ 'alert1' :
"Temperature(%s) is high, sprinkerlers are turned ON"
%temp_sensor }
, qos=0)
    sleep(1)
    if success:
        print( 'Published alert1 : ', "Temperature(%s) is high,
sprinkerlers are turned ON" %temp_sensor,"to IBM Watson")
        print("")
    else:
        print("sprinkler-1 is OFF")
        print("")


    #To send alert message if farmer uses the unsafe fertilizer to
crops.


    if (PH_sensor > 7.5 or PH_sensor < 5.5):
        success = deviceCli.publishEvent("Alert2", "json",{ 'alert2'
: "Fertilizer PH level(%s) is not safe,use other fertilizer"
%PH_sensor } ,
qos=0)
```

```python
    sleep(1)
  if success:
    print('Published alert2 : ' , "Fertilizer PH level(%s) is not
safe,use other fertilizer" %PH_sensor,"to IBM Watson")
    print ("")


# To send alert message to farmer that animal attack on crops.


  if (camera_reading == "Detected"):
    success = deviceCli.publishEvent("Alert3", "json", {
'alert3' : "Animal attack on crops detected" }, qos=0)
    sleep(1)
  if success:
     print('Published alert3 : ' , "Animal attack on crops
detected","to IBM Watson","to IBM Watson")
    print("")
 #To send alert message if flame detected on crop land and
turn ON the splinkers to take immediate action.


  if (flame_reading == "Detected"):
   print("sprinkler-2 is ON")
    success = deviceCli.publishEvent("Alert4", "json", { 'alert4'
: "Flame is detected crops are in danger,sprinklers turned ON"
}, qos=0)
```

```python
    sleep(1)
if success:
    print( 'Published alert4 : ' , "Flame is detected crops are in
danger,sprinklers turned ON","to IBM Watson")
    print("")
else:
    print("sprinkler-2 is OFF")
    print("")
#To send alert message if Moisture level is LOW and to Turn
ON Motor-1 for irrigation.
if (moist_level < 20):
    print("Motor-1 is ON")
    success = deviceCli.publishEvent("Alert5", "json", { 'alert5'
: "Moisture level(%s) is low, Irrigation started" %moist_level
}, qos=0)
    sleep(1)
if success:
    print('Published alert5 : ' , "Moisture level(%s) is low,
Irrigation started" %moist_level,"to IBM Watson" )
    print("")
else:
    print("Motor-1 is OFF")
    print("")
```

```python
 #To send alert message if Water level is HIGH and to Turn ON Motor-2 to take water out.
if (water_level > 20):
    print("Motor-2 is ON")
    success = deviceCli.publishEvent("Alert6", "json", { 'alert6' : "Water level(%s) is high, so motor is ON to take water out " %water_level }, qos=0)
    sleep(1)
    if success:
        print('Published alert6 : ' , "water level(%s) is high, so motor is ON to take water out " %water_level,"to IBM Watson" )
        print("")
    else:
        print("Motor-2 of OFF")
        print("")
 #command recived by farmer
deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

File Edit Format Run Options Window Help

```python
import random
import ibmiotf.device
from time import sleep
import sys
#IBM Watson Device Credentials.
organization = "ory9vy"
deviceType = "Raspery"
deviceId = "1206"
authMethod = "token"
authToken = "12345678"
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="Motoron":
        print ("Motor is ON")
    else :
        print ("Motor is OFF")
#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()
#Connecting to IBM watson.
deviceCli.connect()
while True:
    #Getting values from sensors
    temp_sensor = round( random.uniform(0,80),2)
    PH_sensor = round(random.uniform(1,14),3)
    camera = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
    camera_reading = random.choice(camera)
    flame = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
    flame_reading = random.choice(flame)
    moist_level = round(random.uniform(0,100),2)
    water_level = round(random.uniform(0,30),2)
#storing the sensor data to send in json format to cloud.
    temp_data = { 'Temperature' : temp_sensor }
    PH_data = { 'PH Level' : PH_sensor }
    camera_data = { 'Animal attack' : camera_reading}
    flame_data = { 'Flame' : flame_reading }
    moist_data = { 'Moisture Level' : moist_level}
    water_data = { 'Water Level' : water_level}
# publishing Sensor data to IBM Watson for every 5-10 seconds.
    success = deviceCli.publishEvent("Temperature sensor", "json", temp_data, qos=0)
    if success:
        success = deviceCli.publishEvent("Temperature sensor", "json", temp_data, qos=0)
```

Ln: 14 Col: 23

30°C Haze   ENG IN   12:51 19-11-2022

```python
#Connecting to IBM watson.
deviceCli.connect()
while True:
    #Getting values from sensors
    temp_sensor = round( random.uniform(0,80),2)
    PH_sensor = round(random.uniform(1,14),3)
    camera = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
    camera_reading = random.choice(camera)
    flame = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
    flame_reading = random.choice(flame)
    moist_level = round(random.uniform(0,100),2)
    water_level = round(random.uniform(0,30),2)
#storing the sensor data to send in json format to cloud.
    temp_data = { 'Temperature' : temp_sensor }
    PH_data = { 'PH Level' : PH_sensor }
    camera_data = { 'Animal attack' : camera_reading}
    flame_data = { 'Flame' : flame_reading }
    moist_data = { 'Moisture Level' : moist_level}
    water_data = { 'Water Level' : water_level}
# publishing Sensor data to IBM Watson for every 5-10 seconds.
    success = deviceCli.publishEvent("Temperature sensor", "json", temp_data, qos=0)
    if success:
        success = deviceCli.publishEvent("Temperature sensor", "json", temp_data, qos=0)
        print (" ..........................publish ok.......................... ")
        print ("Published Temperature = %s C" % temp_sensor, "to IBM Watson")
    success = deviceCli.publishEvent("PH sensor", "json", PH_data, qos=0)
    sleep(1)
    if success:
        print ("Published PH Level = %s" % PH_sensor, "to IBM Watson")
    success = deviceCli.publishEvent("camera", "json", camera_data, qos=0)
    sleep(1)
    if success:
        print ("Published Animal attack %s " % camera_reading, "to IBM Watson")
    success = deviceCli.publishEvent("Flame sensor", "json", flame_data, qos=0)
    sleep(1)
    if success:
        print ("Published Flame %s " % flame_reading, "to IBM Watson")
    success = deviceCli.publishEvent("Moisture sensor", "json", moist_data, qos=0)
    sleep(1)
    if success:
        print ("Published Moisture Level = %s " % moist_level, "to IBM Watson")
    success = deviceCli.publishEvent("Water sensor", "json", water_data, qos=0)
    sleep(1)
    if success:
        print ("Published Water Level = %s cm" % water_level, "to IBM Watson")
    print ("")
#Automation to control sprinklers by present temperature an to send alert message to IBM Watson.
```

Ln: 14 Col: 23

30°C Haze   ENG IN   12:52 19-11-2022

```python
        success = deviceCli.publishEvent("Alert1", "json",{ 'alert1' : "Temperature(%s) is high, sprinkerlers are turned ON" %temp_sensor }, qos=0)
        sleep(1)
        if success:
            print( 'Published alert1 : ', "Temperature(%s) is high, sprinkerlers are turned ON" %temp_sensor,"to IBM Watson")
    print("")
#To send alert message if farmer uses the unsafe fertilizer to crops.
    if (PH_sensor > 7.5 or PH_sensor < 5.5):
        success = deviceCli.publishEvent("Alert2", "json",{ 'alert2' : "Fertilizer PH level(%s) is not safe,use other fertilizer" %PH_sensor } ,qos=0)
        sleep(1)
        if success:
            print('Published alert2 : ' , "Fertilizer PH level(%s) is not safe,use other fertilizer" %PH_sensor,"to IBM Watson")
    print("")
#To send alert message to farmer that animal attack on crops.
    if (camera_reading == "Detected"):
        success = deviceCli.publishEvent("Alert3", "json", { 'alert3' : "Animal attack on crops detected" }, qos=0)
        sleep(1)
        if success:
            print('Published alert3 : ' , "Animal attack on crops detected","to IBM Watson","to IBM Watson")
    print("")
#To send alert message if flame detected on crop land and turn ON the splinkers to take immediate action.
    if (flame_reading == "Detected"):
        print("sprinkler-2 is ON")
        success = deviceCli.publishEvent("Alert4", "json", { 'alert4' : "Flame is detected crops are in danger,sprinklers turned ON" }, qos=0)
        sleep(1)
        if success:
            print( 'Published alert4 : ' , "Flame is detected crops are in danger,sprinklers turned ON","to IBM Watson")
#To send alert message if Moisture level is LOW and to Turn ON Motor-1 for irrigation.
    if (moist_level < 20):
        print("Motor-1 is ON")
        success = deviceCli.publishEvent("Alert5", "json", { 'alert5' : "Moisture level(%s) is low, Irrigation started" %moist_level }, qos=0)
        sleep(1)
        if success:
            print('Published alert5 : ' , "Moisture level(%s) is low, Irrigation started" %moist_level,"to IBM Watson" )
    print("")
#To send alert message if Water level is HIGH and to Turn ON Motor-2 to take water out.
    if (water_level > 20):
        print("Motor-2 is ON")
        success = deviceCli.publishEvent("Alert6", "json", { 'alert6' : "Water level(%s) is high, so motor is ON to take water out " %water_level }, qos=0)
        sleep(1)
        if success:
            print('Published alert6 : ' , "water level(%s) is high, so motor is ON to take water out " %water_level,"to IBM Watson" )
    print("")
#command recived by farmer
deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

OUTPUT

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:\Users\SRUTHI PRIYA D.M\AppData\Local\Programs\Python\Python37\god.py
...........................publish ok............................ 2022-11-19 12:54:55,830   ibmiotf.device.Client    INFO    Connected successfully: d:ory9vy:Raspery:1206

Published Temperature = 35.0 C to IBM Watson
Published PH Level = 11.062 to IBM Watson
Published Animal attack Not Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
Published Moisture Level = 88.66  to IBM Watson
Published Water Level = 14.05 cm to IBM Watson

Published alert1 :  Temperature(35.0) is high, sprinkerlers are turned ON to IBM Watson

Published alert2 :  Fertilizer PH level(11.062) is not safe,use other fertilizer to IBM Watson

Published alert3 :  Animal attack on crops detected to IBM Watson to IBM Watson

Published alert4 :  Flame is detected crops are in danger,sprinklers turned ON to IBM Watson
Published alert5 :  Moisture level(88.66) is low, Irrigation started to IBM Watson

Published alert6 :  water level(14.05) is high, so motor is ON to take water out  to IBM Watson

 ...........................publish ok............................
Published Temperature = 47.13 C to IBM Watson
Published PH Level = 9.839 to IBM Watson
```

```
sprinkler-1 is ON
Published alert1 :  Temperature(63.42) is high, sprinkerlers are turned ON to IBM Watson

Published alert2 :  Fertilizer PH level(13.488) is not safe,use other fertilizer to IBM Watson

Published alert3 :  Animal attack on crops detected to IBM Watson to IBM Watson

Published alert4 :  Flame is detected crops are in danger,sprinklers turned ON to IBM Watson
Published alert5 :  Moisture level(34.18) is low, Irrigation started to IBM Watson
Motor-2 is ON
Published alert6 :  water level(28.57) is high, so motor is ON to take water out  to IBM Watson

   ...........................publish ok...........................
Published Temperature = 23.72 C to IBM Watson
Published PH Level = 6.268 to IBM Watson
Published Animal attack Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
Published Moisture Level = 23.07  to IBM Watson
Published Water Level = 2.08 cm to IBM Watson

Published alert1 :  Temperature(23.72) is high, sprinkerlers are turned ON to IBM Watson

Published alert2 :  Fertilizer PH level(6.268) is not safe,use other fertilizer to IBM Watson

Published alert3 :  Animal attack on crops detected to IBM Watson to IBM Watson

Published alert4 :  Flame is detected crops are in danger,sprinklers turned ON to IBM Watson
Published alert5 :  Moisture level(23.07) is low, Irrigation started to IBM Watson

Published alert6 :  water level(2.08) is high, so motor is ON to take water out  to IBM Watson

   ...........................publish ok...........................
Published Temperature = 56.24 C to IBM Watson
Published PH Level = 10.569 to IBM Watson
Published Animal attack Not Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
Published Moisture Level = 13.24  to IBM Watson
Published Water Level = 3.55 cm to IBM Watson

sprinkler-1 is ON
Published alert1 :  Temperature(56.24) is high, sprinkerlers are turned ON to IBM Watson

Published alert2 :  Fertilizer PH level(10.569) is not safe,use other fertilizer to IBM Watson
```

```
Published alert6 :  water level(8.0) is high, so motor is ON to take water out  to IBM Watson

   ...........................publish ok...........................
Published Temperature = 14.93 C to IBM Watson
Published PH Level = 5.894 to IBM Watson
Published Animal attack Not Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
Published Moisture Level = 74.14  to IBM Watson
Published Water Level = 16.49 cm to IBM Watson

Published alert1 :  Temperature(14.93) is high, sprinkerlers are turned ON to IBM Watson

Published alert2 :  Fertilizer PH level(5.894) is not safe,use other fertilizer to IBM Watson

Published alert3 :  Animal attack on crops detected to IBM Watson to IBM Watson

Published alert4 :  Flame is detected crops are in danger,sprinklers turned ON to IBM Watson
Published alert5 :  Moisture level(74.14) is low, Irrigation started to IBM Watson

Published alert6 :  water level(16.49) is high, so motor is ON to take water out  to IBM Watson

   ...........................publish ok...........................
Published Temperature = 34.66 C to IBM Watson
Published PH Level = 9.944 to IBM Watson
Published Animal attack Not Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
Published Moisture Level = 63.26  to IBM Watson
Published Water Level = 6.15 cm to IBM Watson

Published alert1 :  Temperature(34.66) is high, sprinkerlers are turned ON to IBM Watson

Published alert2 :  Fertilizer PH level(9.944) is not safe,use other fertilizer to IBM Watson

Published alert3 :  Animal attack on crops detected to IBM Watson to IBM Watson

Published alert4 :  Flame is detected crops are in danger,sprinklers turned ON to IBM Watson
Published alert5 :  Moisture level(63.26) is low, Irrigation started to IBM Watson

Published alert6 :  water level(6.15) is high, so motor is ON to take water out  to IBM Watson

   ...........................publish ok...........................
Published Temperature = 70.94 C to IBM Watson
Published PH Level = 1.112 to IBM Watson
Published Animal attack Not Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
```

```
 ...........................publish ok...........................
Published Temperature = 51.5 C to IBM Watson
Published PH Level = 11.531 to IBM Watson
Published Animal attack Not Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
Published Moisture Level = 92.18  to IBM Watson
Published Water Level = 2.38 cm to IBM Watson

sprinkler-1 is ON
Published alert1 :  Temperature(51.5) is high, sprinkerlers are turned ON to IBM Watson

Published alert2 :  Fertilizer PH level(11.531) is not safe,use other fertilizer to IBM Watson

Published alert3 :  Animal attack on crops detected to IBM Watson to IBM Watson

Published alert4 :  Flame is detected crops are in danger,sprinklers turned ON to IBM Watson
Published alert5 :  Moisture level(92.18) is low, Irrigation started to IBM Watson

Published alert6 :  water level(2.38) is high, so motor is ON to take water out  to IBM Watson

 ...........................publish ok...........................
Published Temperature = 13.1 C to IBM Watson
Published PH Level = 13.165 to IBM Watson
Published Animal attack Not Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
Published Moisture Level = 98.04  to IBM Watson
Published Water Level = 20.72 cm to IBM Watson

Published alert1 :  Temperature(13.1) is high, sprinkerlers are turned ON to IBM Watson
```