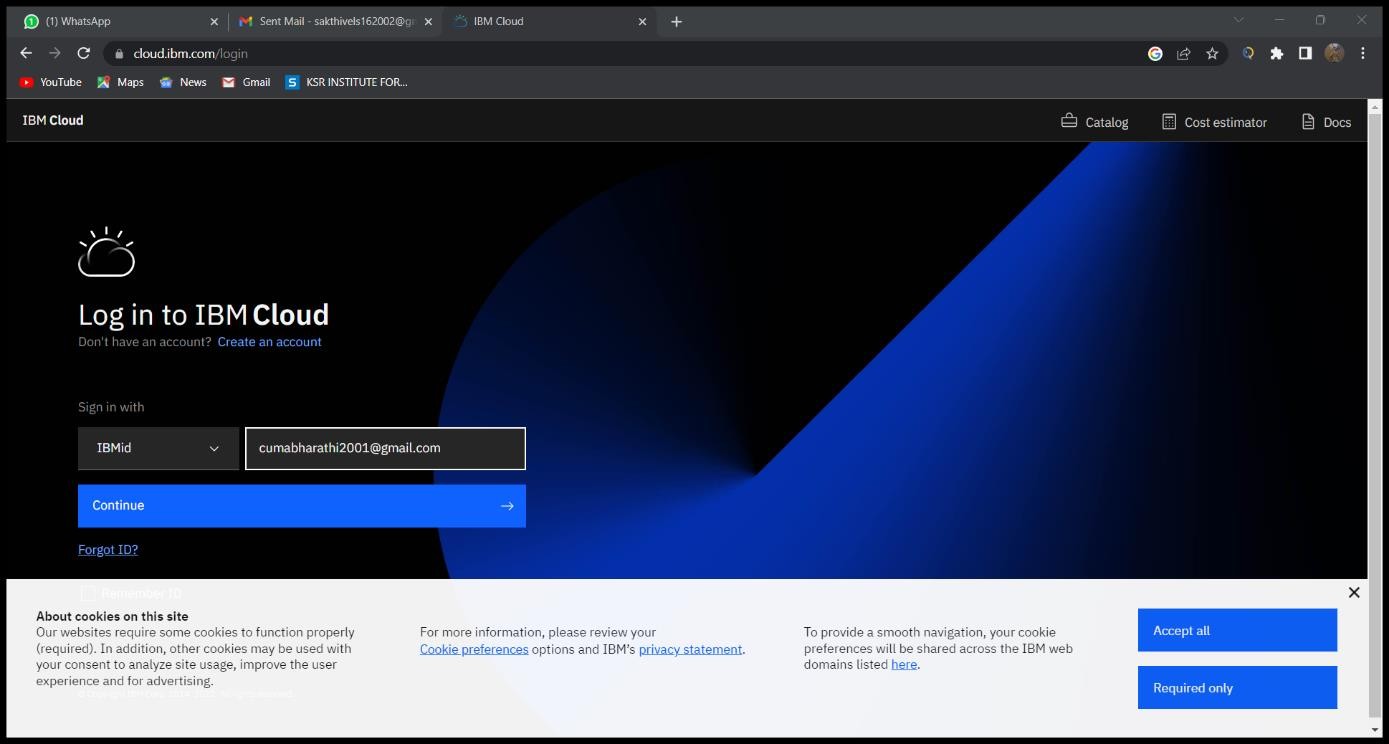
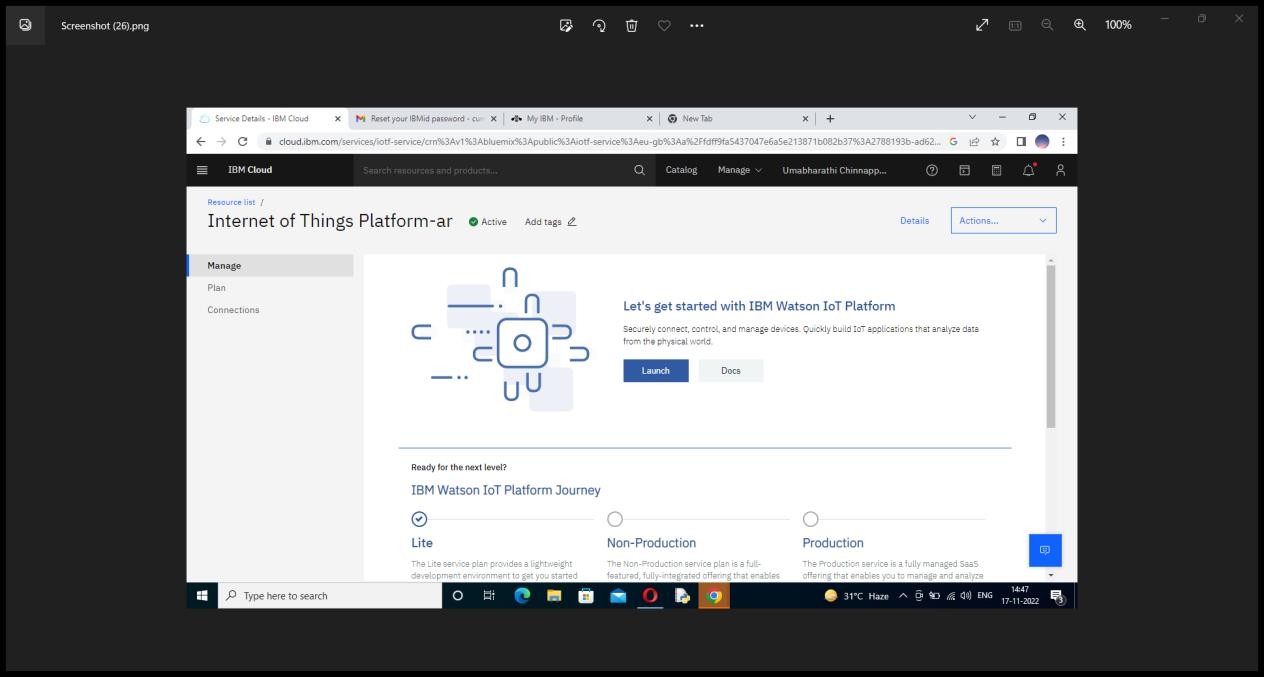
**DEVELOP THE PYTHON SCRIPT**

|  |  |
| --- | --- |
| **TEAM ID** | **PNT2022TMID28861** |
| **PROJECT NAME** | **IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE** |

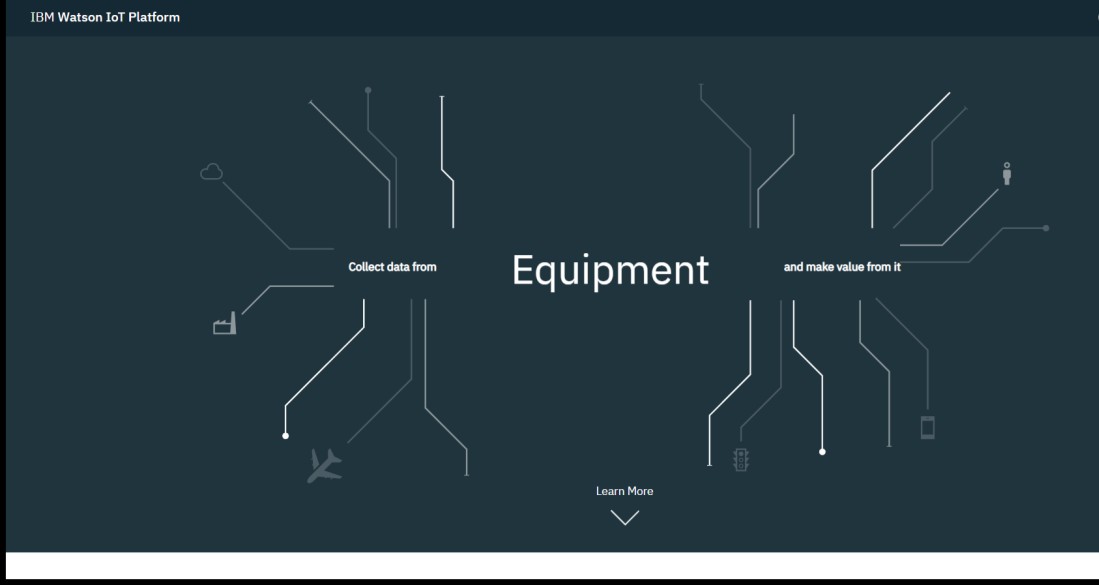
**STEP 1:** First login your IBM cloud account



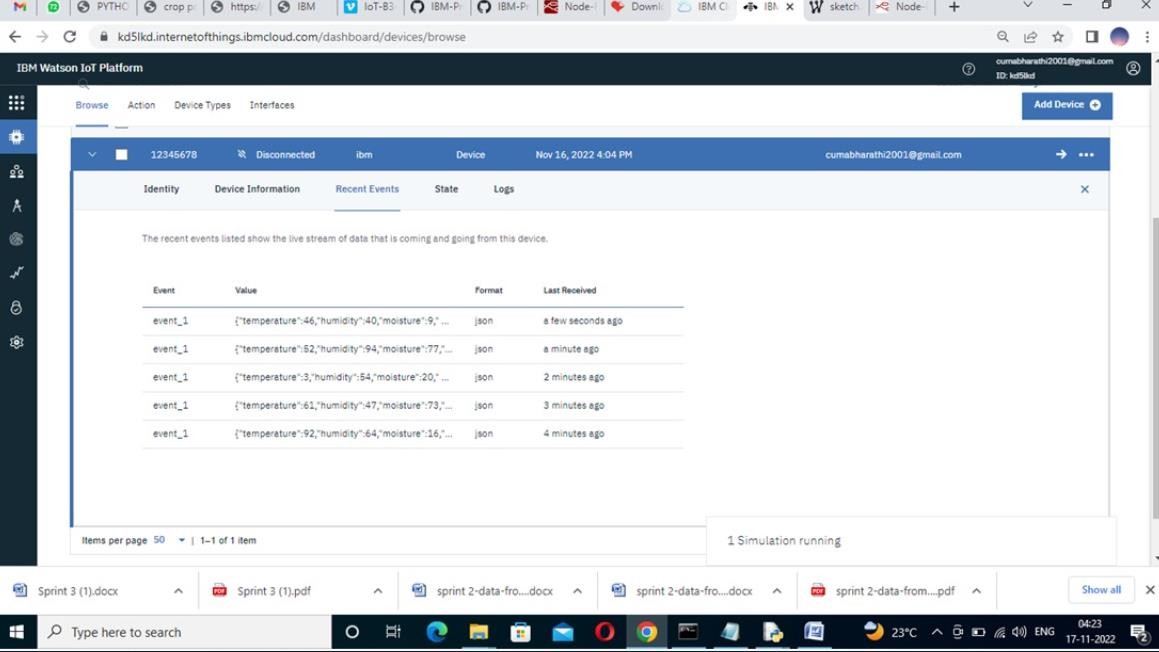
**STEP 2:** Internet of things platform smart crop protection will be created, where there are different options like manage, plan, and connection.



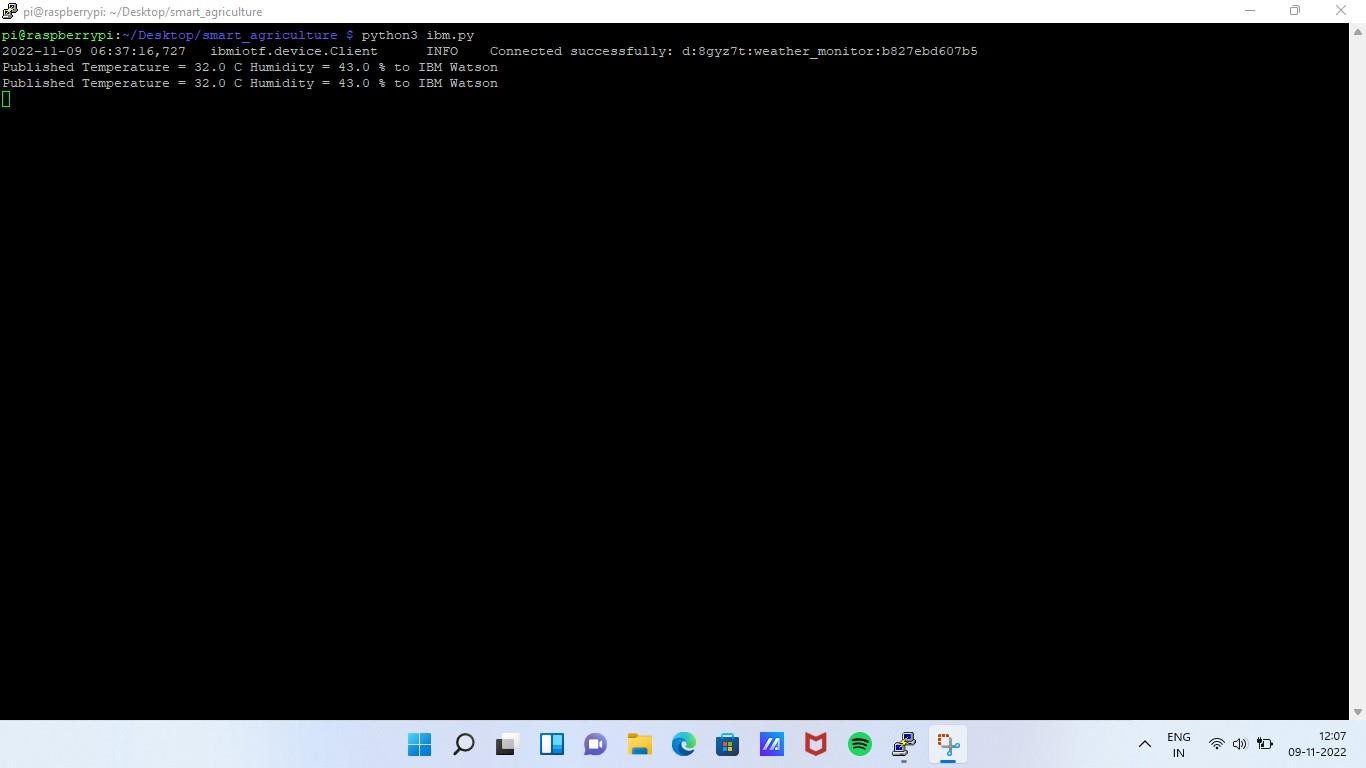
**STEP 3:** Clicking on the launch button in the manage tab, it will open to this.



**STEP 4:** while running python code temperature and humidity value are published in IBM IoT Watson platform.



**STEP 4:** This is the python program output which is published in IBM IoT Watson platform.



**PYTHON CODE:**

import time import sys import ibmiotf.application import ibmiotf.device import Adafruit\_DHT

#Provide your IBM Watson Device Credentials organization = "kd5lkd" deviceType = "smartcropprotection" deviceId = "87654321"

authMethod = "use-token-auth" authToken = "12345678"

# Initialize GPIO and DHT11 sensor

= Adafruit\_DHT.DHT11 pin=4

def myCommandCallback(cmd):

print("Command received: %s" % cmd.data['command']) status = cmd.data['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 exception connecting device: %s" % str(e)) sys.exit()

deviceCli.connect()

while True:

#Get Sensor Data from DHT11

Humidity, Temperature = Adafruit\_DHT.read\_retry(sensor, pin) data = {"d":{'Temperature' :Temperature, 'Humidity': Humidity}}

#print data def myOnPublishCallback():

print ("Published Temperature = %s C" % Temperature, "Humidity = %s

%%" % Humidity, "to IBM Watson")

success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on\_publish=myOnPublishCallback) if not success:

print("Not connected to IoTF") time.sleep(1)

deviceCli.commandCallback = myCommandCallback

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