

# **IOT ENABLED SMART FARMING APPLICATION.**

## **Sprint Delivery – 1**

TEAM ID: PNT2022TMID47731

TEAM MEMBER:

Yuvasri

Moofika Begam

Siva Bala

Divya Bharathi

## 1. Introduction

The main aim of this project is to help farmers automate their farms by providing them with a Web App through which they can monitor the parameters of the field like Temperature, soil moisture, humidity and etc and control the equipment like water motor and other devices remotely via internet without their actual presence in the field.

## 2. Problem Statement

Farmers are to be present at farm for its maintenance irrespective of the weather conditions. They have to ensure that the crops are well watered and the farm status is monitored by them physically. Farmer have to stay most of the time in field in order to get a good yield. In difficult times like in the presence of pandemic also they have to work hard in their fields risking their lives to provide food for the country.

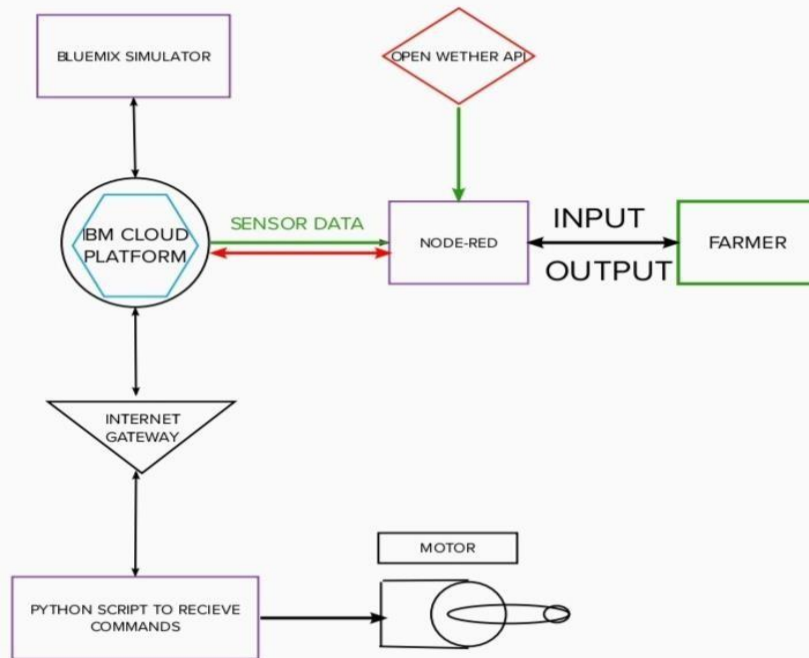
## 3. Proposed Solution

In order to improve the farmer's working conditions and make them easier, we introduce IoT services to him in which we use cloud services and internet to enable farmer to continue his work remotely via internet. He can monitor the field parameters and control the devices in farm.

## 4. Theoretical Analysis

### 4.1 Block Diagram

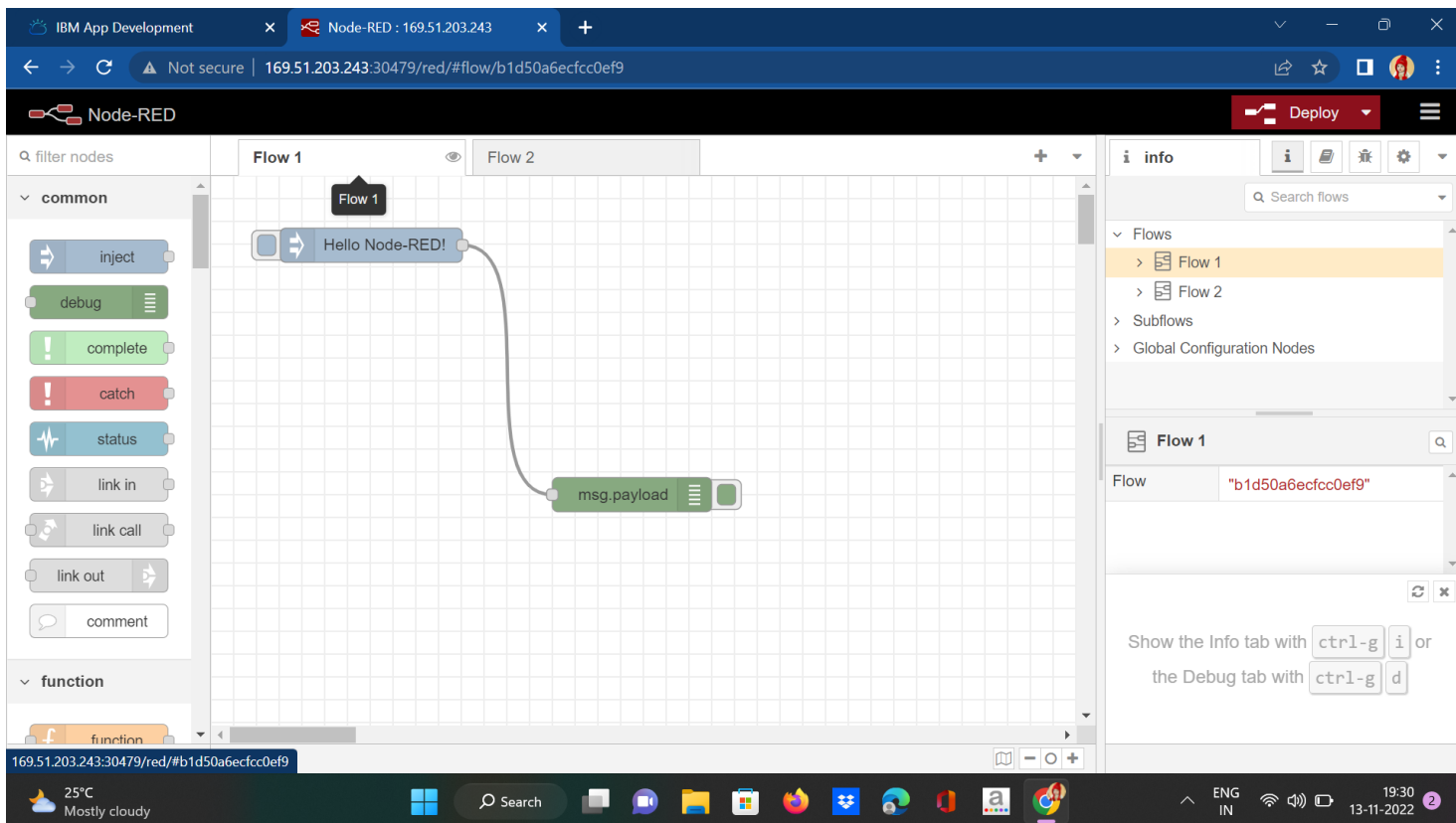
In order to implement the solution , the following approach as shown in the block diagram is used



## 4.2 Required Software Installation

### 4.2.A Node-Red

Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things. Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions.



## Installation :

- First install npm/node.js
- Open cmd prompt
- Type => npm install node-red

## To run the application :

- Open cmd prompt
- Type=>node-red
- Then open <http://localhost:1880/> in browser

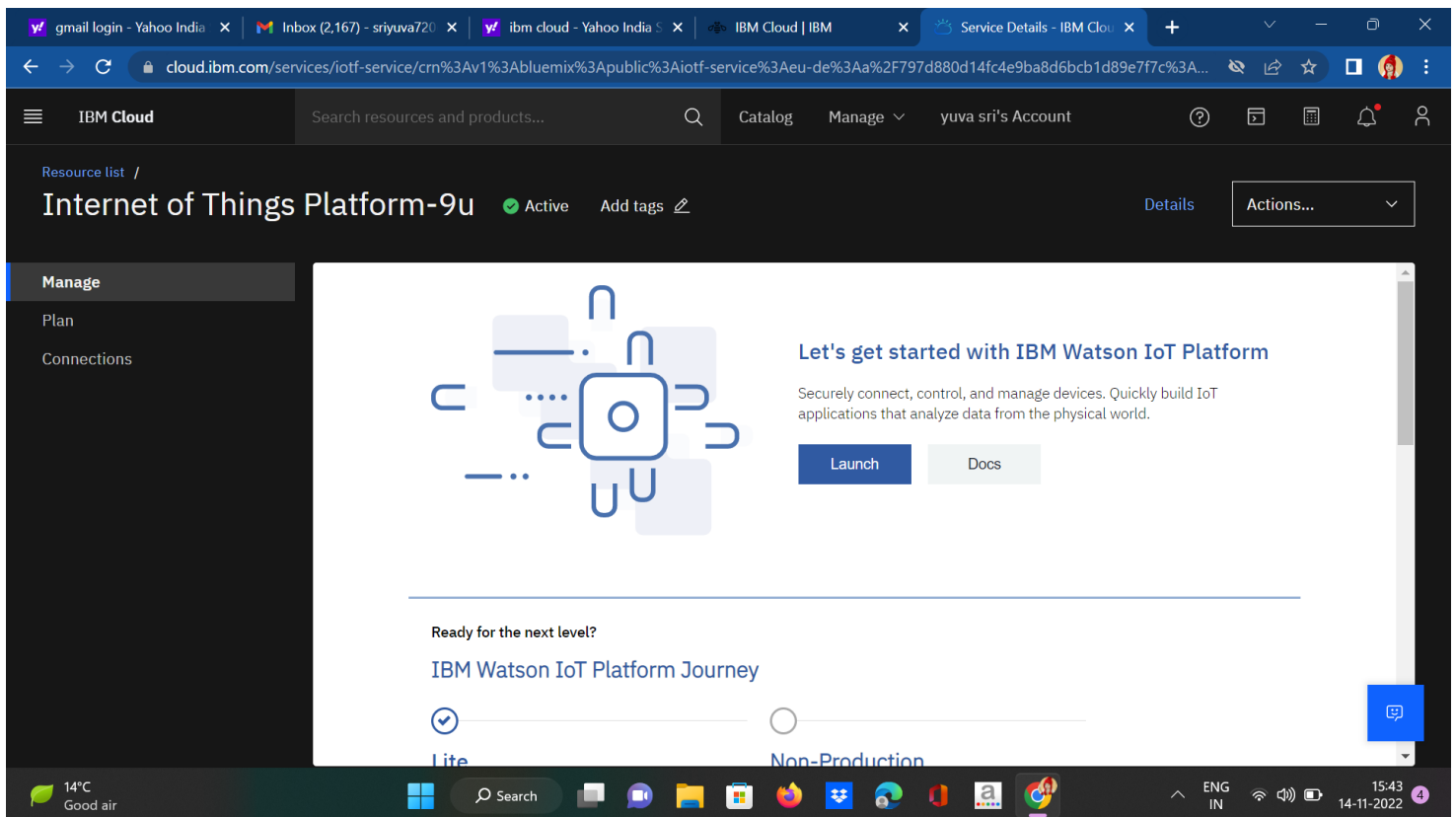
## Installation of IBM IoT and Dashboard nodes for Node-Red

In order to connect to IBM Watson IoT platform and create the Web App UI these nodes are required

1. IBM IoT node
2. Dashboard node

## 4.2.B IBM Watson IoT Platform

A fully managed, cloud-hosted service with capabilities for device registration, connectivity, control, rapid visualization and data storage. IBM Watson IoT Platform is a managed, cloud-hosted service designed to make it simple to derive value from your IoT devices.



### Steps to configure:

- Create an account in IBM cloud using your email ID
- Create IBM Watson Platform in services in your IBM cloud account
- Launch the IBM Watson IoT Platform
- Create a new device
- Give credentials like device type, device ID, Auth. Token
- Create API key and store API key and token elsewhere.

The screenshot displays the IBM Watson IoT Platform interface. The browser address bar shows the URL: `yt599.internetofthings.ibmcloud.com/dashboard/devices/drilldown/abcd:13?returnTo=/devices/browse`. The page title is "Device Drilldown - 13". A left sidebar contains a menu with options: "Device Credentials", "Connection Information", "Recent Events", "State", "Device Information", "Metadata", "Diagnostics", "Connection Logs", and "Device Actions". The "Device Credentials" section is active, showing a table of credentials for device "abcd".

Property	Value
Organization ID	yt599
Device Type	abcd
Device ID	13
Authentication Method	use-token-auth
Authentication Token	!yOeFn(cL-rsKE-!i+

Below the table, a status bar indicates "0 Simulations running". The bottom of the screen shows a Windows taskbar with the date and time as 15:44 on 14-11-2022.

## 4.2.C Python IDE

Install Python3 compiler

Install any python IDE to execute python scripts, in my case I used Spyder to execute the code.

IDLE Shell 3.10.4

File Edit Shell Debug Options Window Help

```
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("smart farmer-IOT enabled smartfarming application")
smart farmer-IOT enabled smartfarming application
>>> |
```

Ln: 5 Col: 0



### Code:

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

#Provide your IBM Watson Device Credentials
organization ID= "ytk599"
deviceType = "abcd"
deviceId = "13"
authMethod = "use-
token-auth"
authToken = "12345678"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("motor is on")
    elif status == "motoroff":
        print ("motor 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 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
```

```
    temp=random.randint(90,110)
```

```
    Humid=random.randint(60,100)
```

```
    Mois=random.randint(20,120)
```

```
    data = { 'temp' : temp, 'Humid': Humid, 'Mois' :Mois}
```

```
    #print data
```

```
    def myOnPublishCallback():
```

```
print ("Published Temperature = %s C" % temp, "Humidity = %s %" % Humid, "Moisture =%s deg c" %Mois, "to IBM Watson")
```

```
success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
```

```
if not success:
```

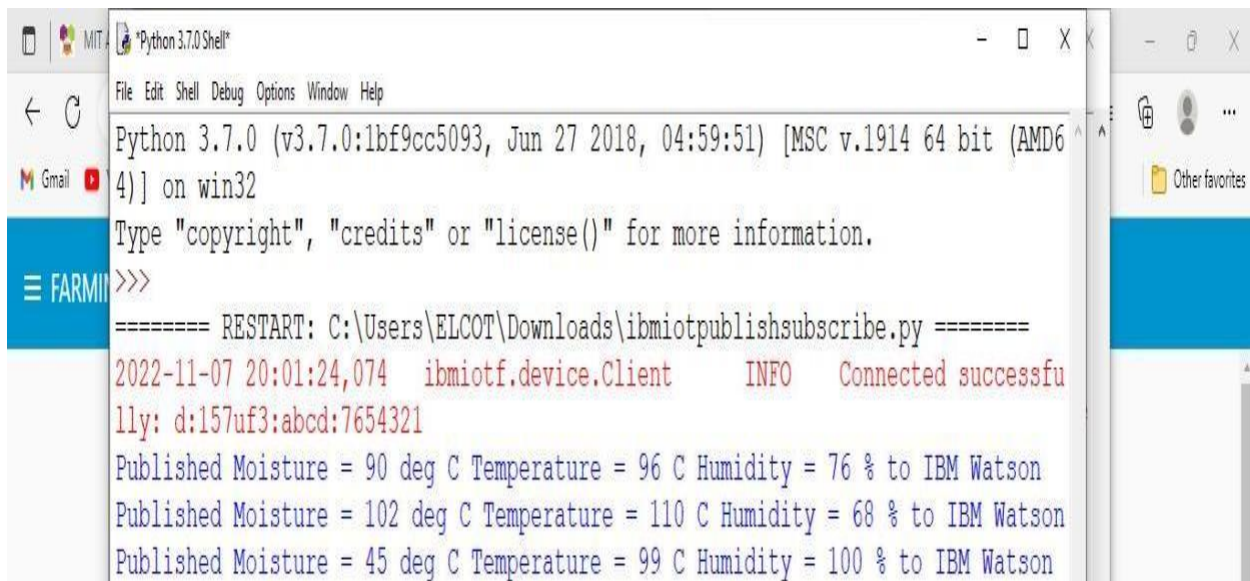
```
    print("Not connected to IoT")
```

```
time.sleep(10)
```

```
deviceCli.commandCallback = myCommandCallback
```

```
# Disconnect the device and application from the cloud
```

```
deviceCli.disconnect()
```



```
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\ELCOT\Downloads\ibmiotpublishsubscribe.py =====
2022-11-07 20:01:24,074 ibmiotf.device.Client INFO Connected successfully: d:157uf3:abcd:7654321
Published Moisture = 90 deg C Temperature = 96 C Humidity = 76 % to IBM Watson
Published Moisture = 102 deg C Temperature = 110 C Humidity = 68 % to IBM Watson
Published Moisture = 45 deg C Temperature = 99 C Humidity = 100 % to IBM Watson
```

### 4.3 IoT Simulator

In our project in the place of sensors we are going to use IoT sensor simulator which give random readings to the connected cloud.

The link to simulator:

<https://watson-iot-sensor-simulator.mybluemix.net/>

We need to give the credentials of the created device in IBM Watson IoT Platform to connect cloud to simulator.

### 4.4 OpenWeather API

OpenWeatherMap is an online service that provides weather data. It provides current weather data, forecasts and historical data to more than 2 million customer.

Website link: <https://openweathermap.org/guide>

#### **Steps to configure:**

- o Create account in OpenWeather
- o Find the name of your city by searching
- o Create API key to your account
- o Replace “city name” and “your api key” with your city and API key in below red text

[api.openweathermap.org/data/2.5/weather?q={city name}&appid={your api key}](https://api.openweathermap.org/data/2.5/weather?q={city name}&appid={your api key})