

# Project Development Phase

## Sprint – 2

Date	06 Nov 2022
Team ID	PNT2022TMID06691
Project Name	Real-Time River Water Quality Monitoring and Control System
Maximum Marks	8 Marks

### USN – 4: Create the IBM Watson IoT and device Settings

As a user, I can create the IBM Watson IoT Platform and integrate the microcontroller with it, to send the sensed data on Cloud.

### IBM Watson IoT Platform:

The screenshot displays the IBM Cloud console interface for the IBM Watson IoT Platform. The top navigation bar includes the IBM Cloud logo, a search bar, and links to Catalog, Manage, and the user's account (Palani M's Account). The main content area shows the 'Internet of Things Platform-fv' resource, which is active. A sidebar on the left lists 'Manage', 'Plan', and 'Connections'. The main content area features a large graphic of a microcontroller connected to various devices, followed by the heading 'Let's get started with IBM Watson IoT Platform' and a description: 'Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.' Below this, there are 'Launch' and 'Docs' buttons. Further down, a section titled 'Ready for the next level?' introduces the 'IBM Watson IoT Platform Journey' with three service plans: Lite, Non-Production, and Production. Each plan includes a brief description and a list of features.

**Internet of Things Platform-fv** Active Add tags Details Actions...

**Manage**

- Plan
- Connections

**Let's get started with IBM Watson IoT Platform**

Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.

[Launch](#) [Docs](#)

**Ready for the next level?**

**IBM Watson IoT Platform Journey**

- Lite**  
The Lite service plan provides a lightweight development environment to get you started with the connectivity capabilities of Watson IoT Platform.
  - Free
  - 200 MB data-transfer limit
- Non-Production**  
The Non-Production service plan is a full-featured, fully-integrated offering that enables you to explore Watson IoT Platform to see how the service can fit into your IoT environment.
  - Starts at \$500 per month
  - Capacity limit based on device type
- Production**  
The Production service is a fully managed SaaS offering that enables you to manage and analyze enterprise IoT data.
  - Includes IBM Service & Support
  - Pricing based on number of devices per device type

## IBM Watson IoT Device:

IBM Watson IoT Platform

61071912130@smartinternz.com  
ID: um5y3e

Browse

Action

Device Types

Interfaces

Add Device +

Browse Devices

All Devices

Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Q Search by Device ID

Device Simulator ☐

|||

Y

	Device ID	Status	Device Type	Class ID	Date Added
>	<input type="checkbox"/> 13448	Disconnected	ESP32	Device	16 Nov 2022 15:42

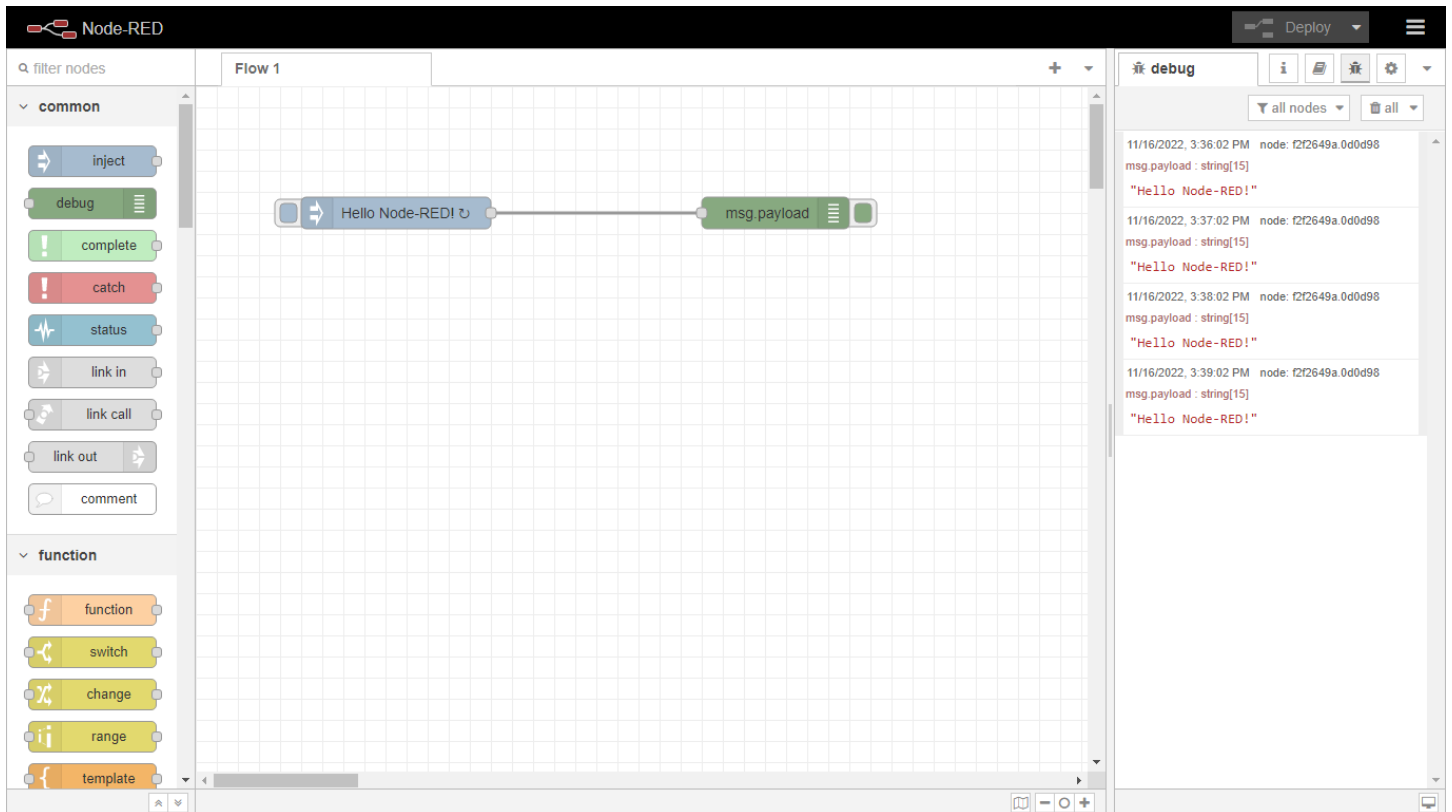
Items per page 50 ▾ | 1–1 of 1 item

1 of 1 page

< 1 ▾ >

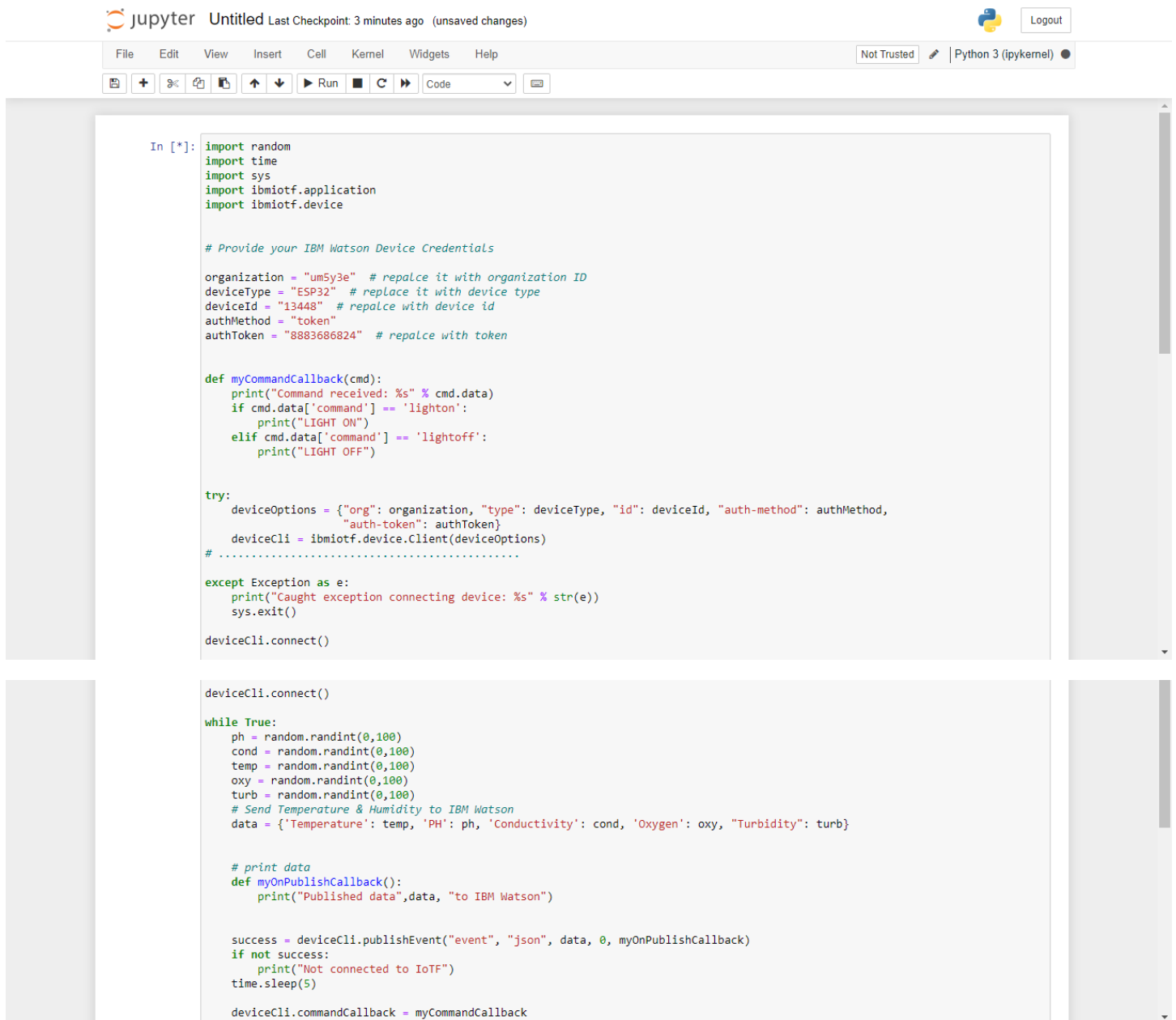
## USN – 5: Create a node red service

As a user, I can create a node red service to integrate the IBM Watson along with the Web UI



## USN – 6: To develop a Python code

As a user, I can create a python code to sense the physical quantity and store data.



The image shows a Jupyter Notebook interface with a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar. The notebook is titled "Untitled" and shows "Last Checkpoint: 3 minutes ago (unsaved changes)". The code is written in Python 3 (ipykernel) and is not trusted. The code defines a command callback function, sets up device options, connects to the IBM Watson IoT device, and publishes random data to the cloud.

```
In [*]: import random
import time
import sys
import ibmiotf.application
import ibmiotf.device

# Provide your IBM Watson Device Credentials

organization = "um5y3e" # replace it with organization ID
deviceType = "ESP32" # replace it with device type
deviceId = "13448" # replace with device id
authMethod = "token"
authToken = "8883686824" # replace with token

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    if cmd.data['command'] == 'lighton':
        print("LIGHT ON")
    elif cmd.data['command'] == 'lightoff':
        print("LIGHT OFF")

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()

deviceCli.connect()

while True:
    ph = random.randint(0,100)
    cond = random.randint(0,100)
    temp = random.randint(0,100)
    oxy = random.randint(0,100)
    turb = random.randint(0,100)
    # Send Temperature & Humidity to IBM Watson
    data = {'Temperature': temp, 'PH': ph, 'Conductivity': cond, 'Oxygen': oxy, 'Turbidity': turb}

    # print data
    def myOnPublishCallback():
        print("Published data",data, "to IBM Watson")

    success = deviceCli.publishEvent("event", "json", data, 0, myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
    time.sleep(5)

    deviceCli.commandCallback = myCommandCallback
```