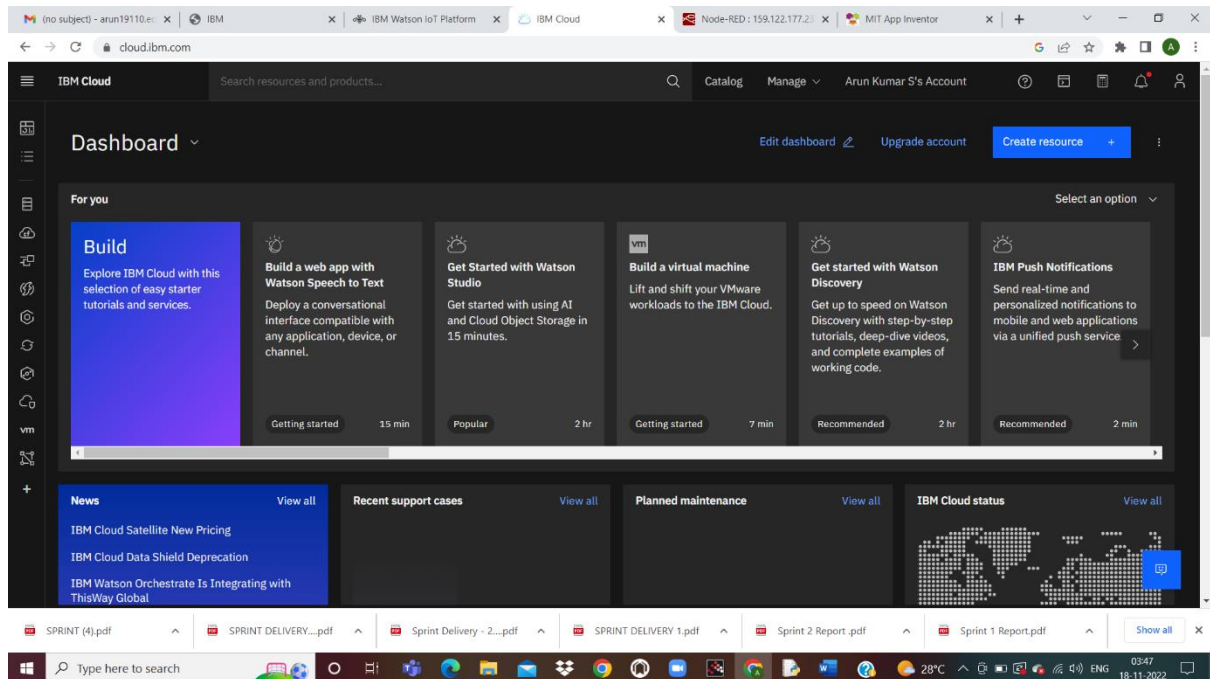


# SmartFarmer - IoT Enabled Smart Farming Application

## Project Development Phase – Sprint 1

### IBM Cloud

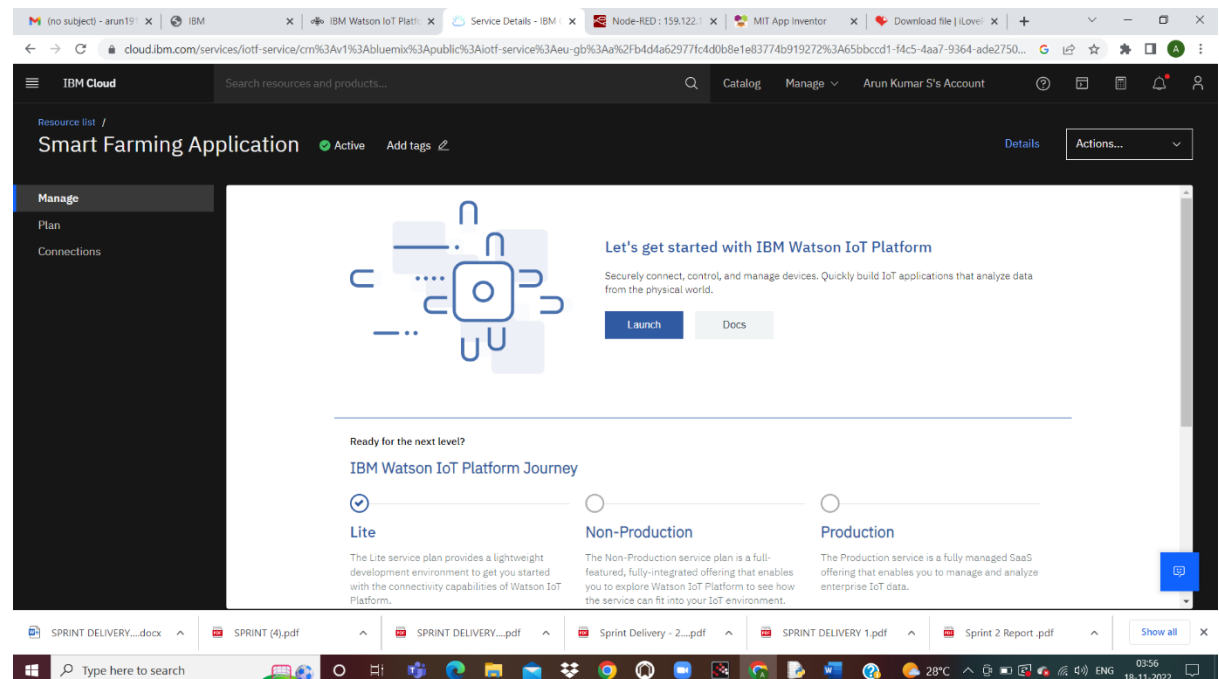


- Search URL [cloud.ibm.com](https://cloud.ibm.com)
- Create an account
- Sign up with login credentials

### Create and Configure IBM Cloud services

- Create the IBM Watson IOT Platform
- Create Node red service

# 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.**

## Procedure

- **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.**

IBM Watson IoT Platform

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.

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Connected	abcd	Device	Nov 16, 2022 10:39 PM	

Items per page 50 | 1-1 of 1 item

1 of 1 page

SPRINT (4).pdf SPRINT DELIVERY...pdf Sprint Delivery - 2...pdf SPRINT DELIVERY 1.pdf Sprint 2 Report .pdf Sprint 1 Report.pdf Show all

Type here to search

28°C

03:52 18-11-2022

## Create Node red Service

IBM Cloud

Node-RED

About Create

Details

Author IBM

Updated 2/11/2020

Type Starter kit

Source code

GitHub

Helpful links

Terms

Tutorial

Overview

This starter kit provides a pre-configured Node-RED application, including a Cloudant service to store the application flow configuration. Add services, generate and download the code, use the IBM Cloud Developer Tools CLI to run and debug locally, then deploy to Cloud Foundry or a DevOps Pipeline.

This starter kit will help you

- Generate an application with Node-RED
- Generate an application with files for deploying to Cloud Foundry or a DevOps Pipeline
- Connect to provisioned services

What's included?

Cloudant

Free to start View pricing

View docs View API reference

Get started

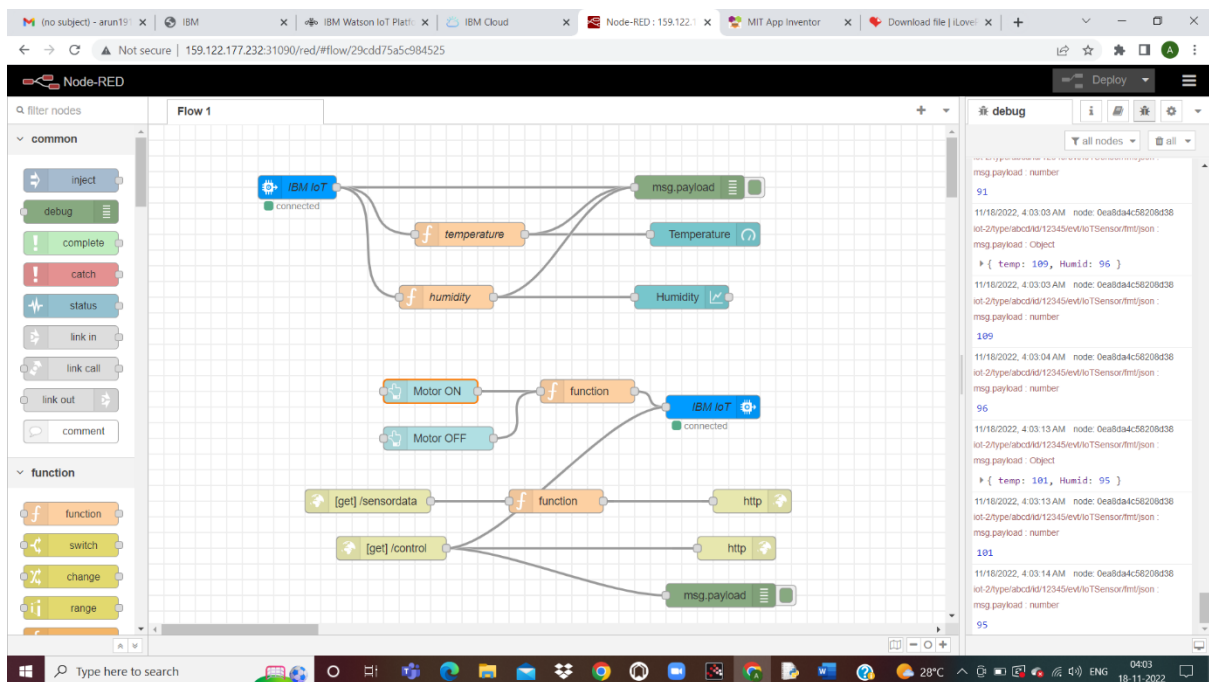
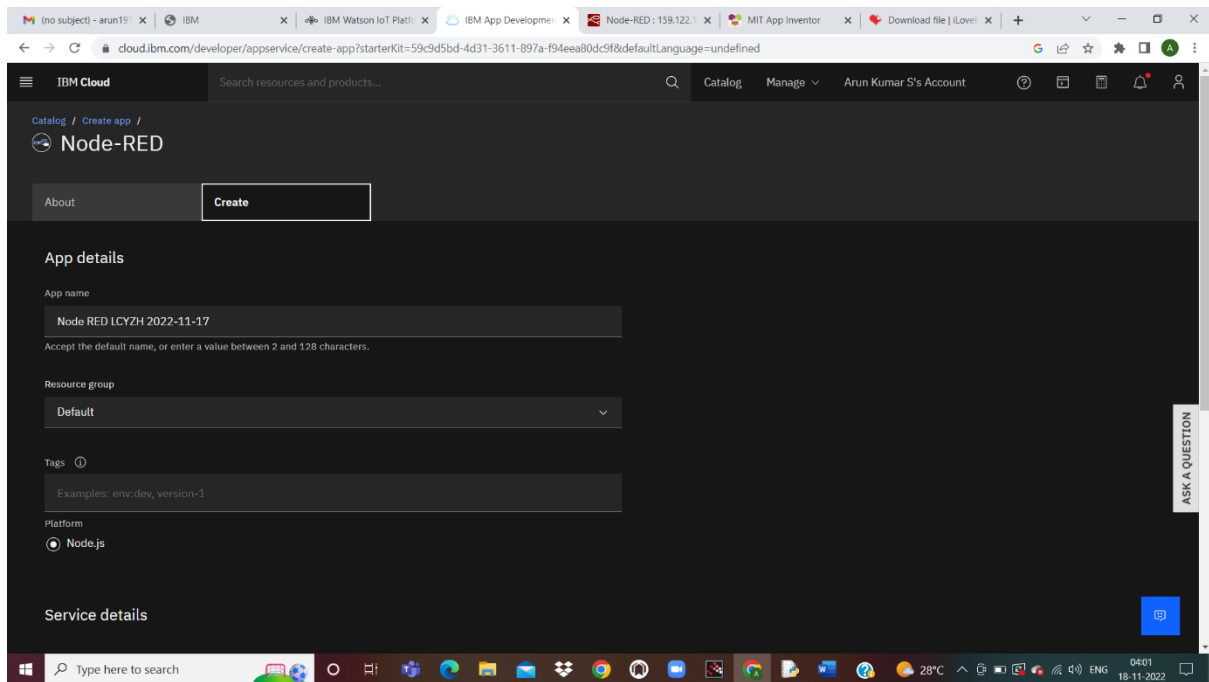
ASK A QUESTION

Type here to search

28°C

04:01 18-11-2022

**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.**



## Python IDLE

- Install python 3.7.0 version
- Import necessary libraries

## **Code**

```
organization = "ojzlch"  
deviceType = "abcd"  
deviceId = "12345"  
authMethod = "token"  
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)

    data = { 'temp' : temp, 'Humid': Humid }
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp,
"Humidity = %s %" % Humid, "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()
```