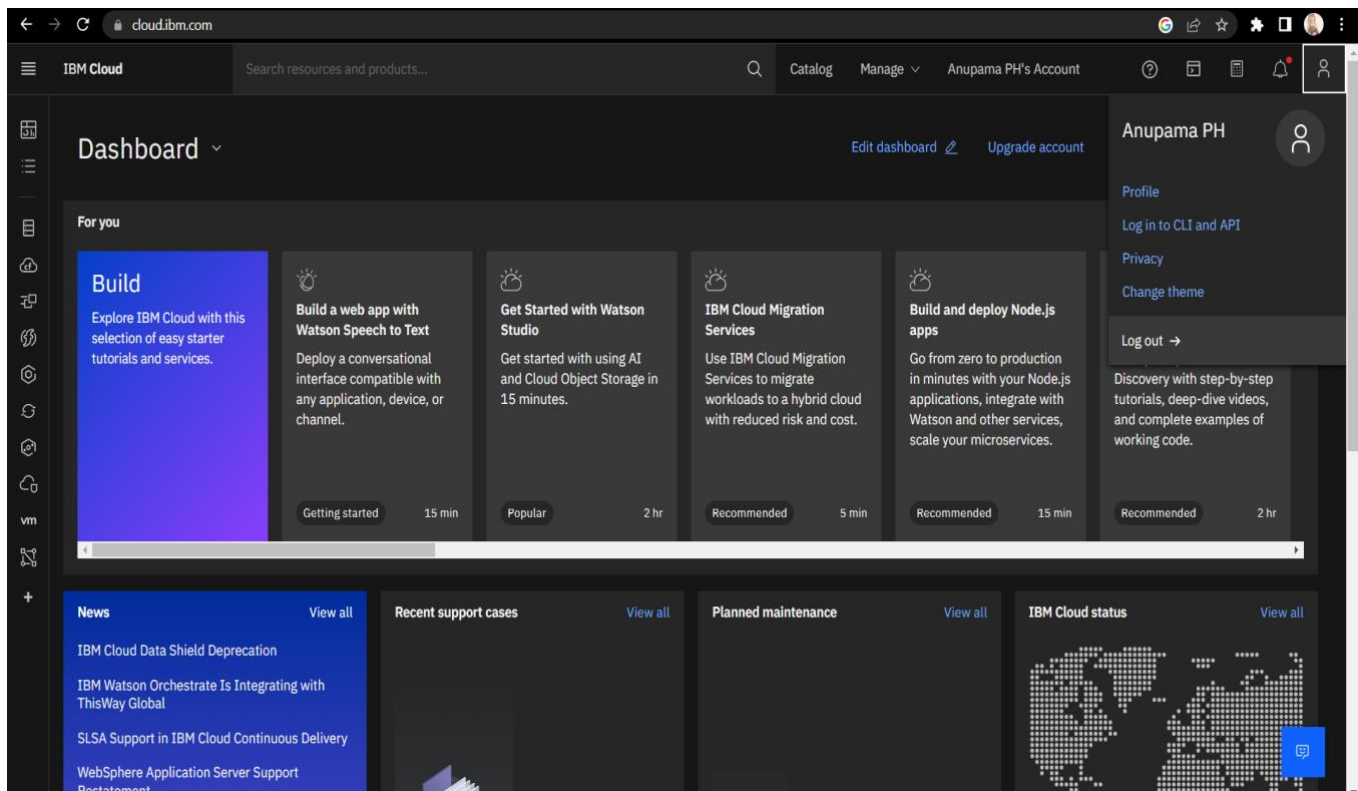


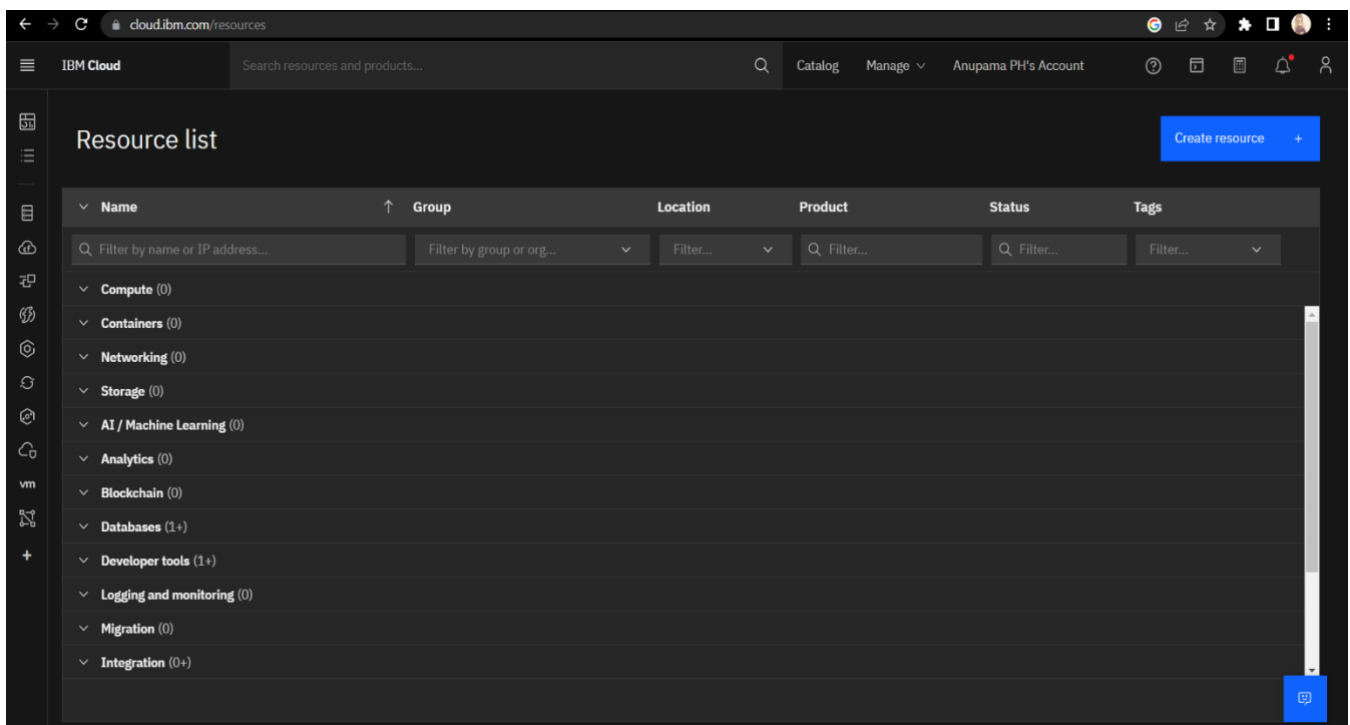
Sprint-1

Date	17 November 2022
Team ID	PNT2022TMID27330
Project Name	Project: Signs with Smart Connectivity for Better Road Safety
Marks	20 Marks

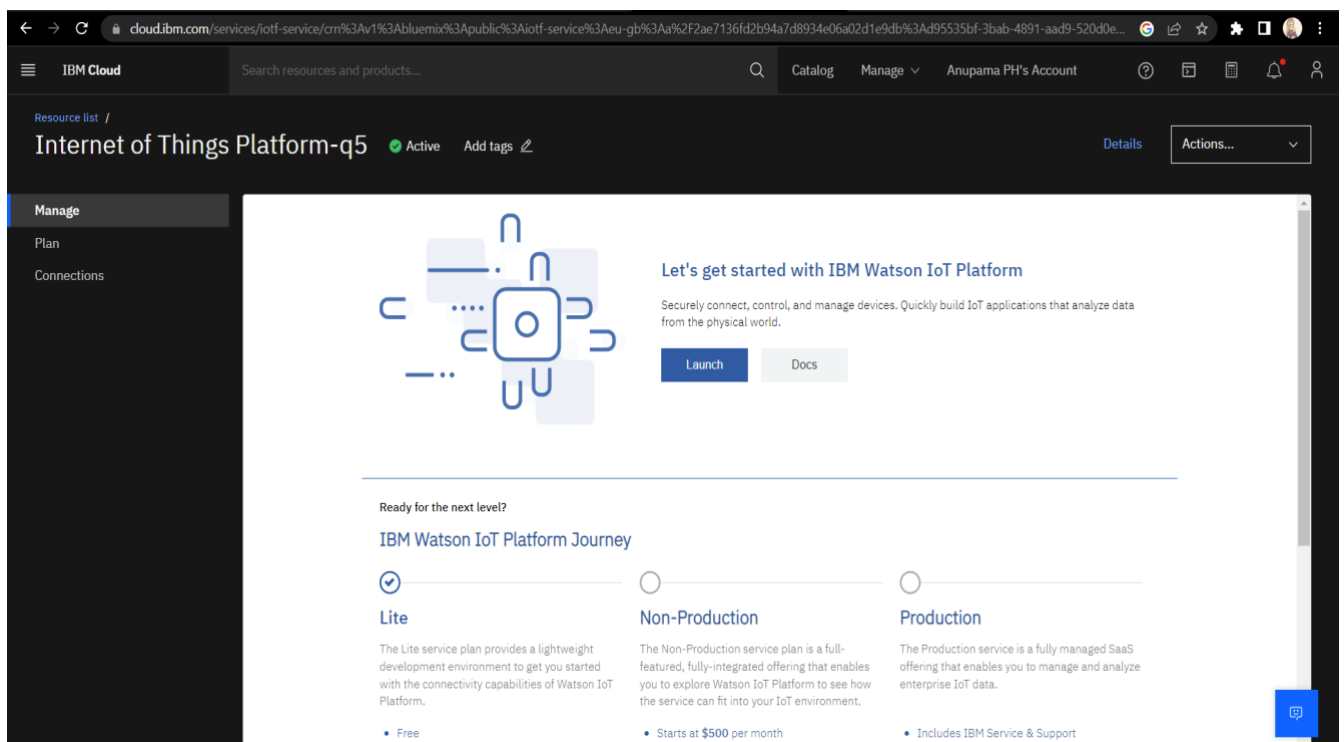
US-1: Create the IBM Cloud services which are being used in this project.



US-2: Configure the IBM Cloud service which are being used in completing this project.



US-3: IBM Watson IoT Platform acts as a mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform.



US-4: In order to connect the IoT device to the IBM Cloud, create a device in IBM Watson IoT platform and get device credentials.

gkgt1cinternetofthings.ibmcloud.com/dashboard/devices/browse

IBM Watson IoT Platform

anupama.hemu@gmail.com
ID: gkgt1c

Browse

Action

Device Types

Interfaces

Add Device

Search by Device ID

Device Simulator

	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
>	12345	Disconnected	NodeMCU	Device	Oct 20, 2022 2:25 PM	
▼	PNT2022TMID27330	Disconnected	PNT2022TMID27330	Device	Nov 16, 2022 9:44 PM	→ ...

Identity

Device Information

Recent Events

State

Logs

Device ID

Device Type

Date Added

Added By

Connection Status

PNT2022TMID27330

PNT2022TMID27330

Nov 16, 2022 9:44 PM

anupama.hemu@gmail.com

Disconnected

Items per page 50 | 1-2 of 2 items

1 of 1 page

1

Identity

Device Information

Recent Events

State

Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_1	{"randomNumber":99}	json	a few seconds ago
event_1	{"randomNumber":98}	json	a few seconds ago
event_1	{"randomNumber":9}	json	a few seconds ago
event_1	{"randomNumber":81}		
event_1	{"randomNumber":75}		

1 Simulation running

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format
event_1	{"Temperature":11,"Humidity":65,"Rain":25}	json
event_1	{"Temperature":9,"Humidity":26,"Rain":42}	json
event_1	{"Temperature":29,"Humidity":27,"Rain":34}	json
event_1	{"Temperature":69,"Humidity":87,"Rain":57}	json
event_1	{"Temperature":43,"Humidity":6,"Rain":99}	json

Items per page 50 | 1-1 of 1 item

Events 1 New event type +

Event type name: event_1 Send 🗑️

Schedule
20 Every Minute

Payload
Specify the event payload in the editor window or by uploading a CSV file.

```

0 {
1   "Temperature": random(0, 100),
2   "Humidity": random(0, 100),
3   "Rain": random(0, 100)
4 }
5

```

Upload a CSV file Cancel Save

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_1	{"Temperature":27,"Humidity":29,"Rain":22}	json	a few seconds ago
event_1	{"Temperature":11,"Humidity":65,"Rain":25}	json	a few seconds ago
event_1	{"Temperature":9,"Humidity":26,"Rain":42}	json	a few seconds ago
event_1	{"Temperature":29,"Humidity":27,"Rain":34}	json	a few seconds ago
event_1	{"Temperature":69,"Humidity":87,"Rain":57}	json	a few seconds ago

Items per page 50 | 1-1 of 1 item

1 of 1 page < 1 >

1 Simulation running

Python code

```
import requests as reqs
```

```
def get(myLocation,APIKEY):
    apiURL = f"https://api.openweathermap.org/data/2.5/weather?q={myLocation}&appid={APIKEY}"
    responseJSON = (reqs.get(apiURL)).json()
    responseObject = {
        "temperature" : responseJSON['main']['temp'] - 273.15,
        "weather" : [responseJSON['weather'][_]['main'].lower() for _ in range(len(responseJSON['weather']))],
        "visibility" : responseJSON['visibility']/100, # visibility in percentage where 10km is 100% and 0km is 0%
    }
    if("rain" in responseJSON):
        responseObject["rain"] = [responseJSON["rain"][key] for key in responseJSON["rain"]]
    return(responseObject)
```

```
# Python code
```

```
# IMPORT SECTION STARTS
```

```
import brain
```

```
# IMPORT SECTION ENDS
```

```
# -----
```

```
# USER INPUT SECTION STARTS
```

```
myLocation = "Chennai,IN"
```

```
APIKEY = "a-gkgt1c-t8k5ryqa64"
```

```
localityInfo = {
```

```
    "schools" : {
```

```
        "schoolZone" : True,
```

```
        "activeTime" : ["7:00","17:30"] # schools active from 7 AM till 5:30 PM
```

```
    },
```

```
    "hospitalsNearby" : False,
```

```
    "usualSpeedLimit" : 40 # in km/hr
```

```
}
```

```
# USER INPUT SECTION ENDS
```

```
# -----
```

```
# MICRO-CONTROLLER CODE STARTS
```

```
print(brain.processConditions(myLocation,APIKEY,localityInfo))
```