SPRINT 1

Team id: PNT2022TMID49101

Project name: smart waste management for metropolitan cities

Create and configure IBM cloud services

Create IBM Watson IoT Platform And Device

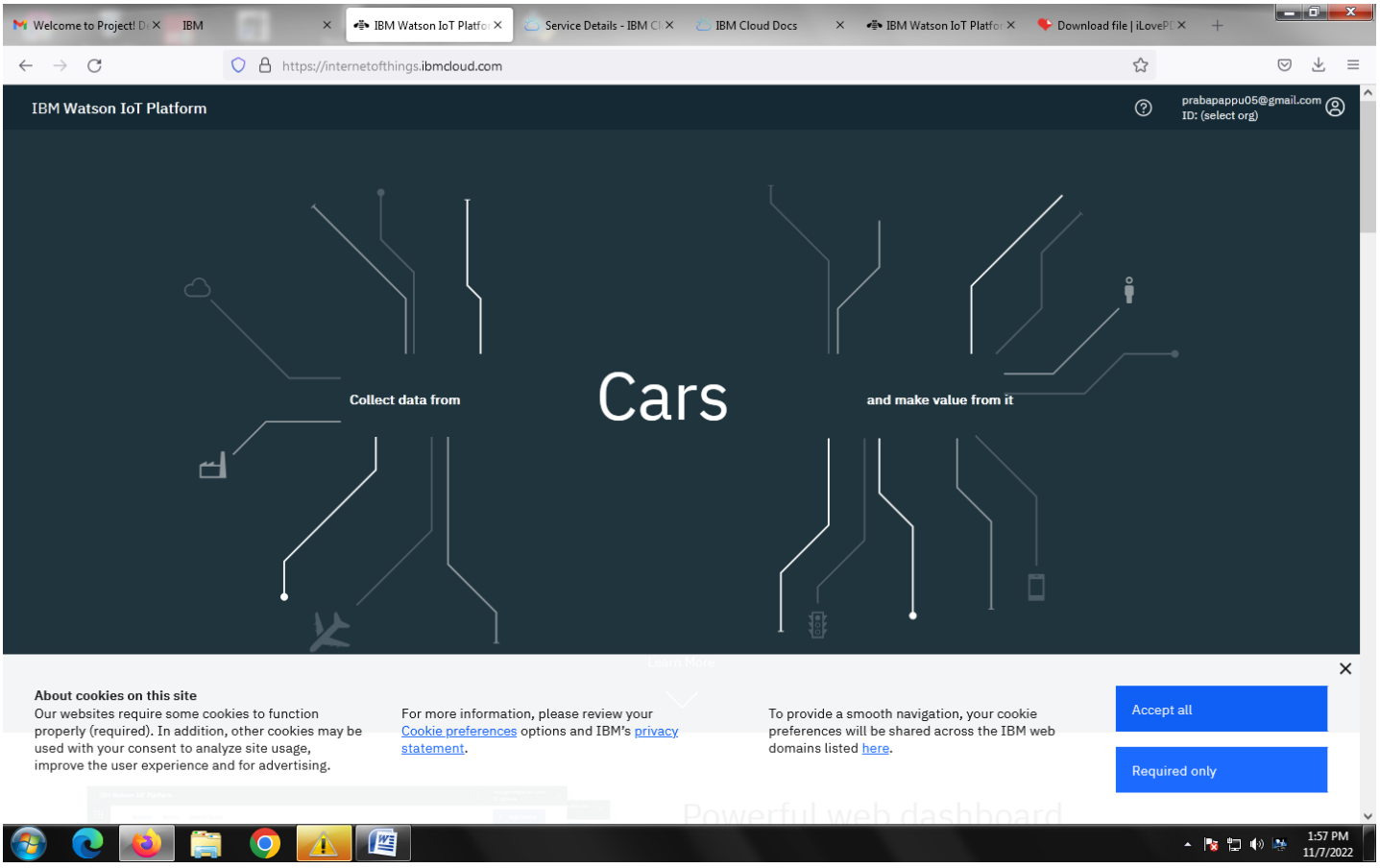
IBM Watson IoT Platform is a complete end-to-end solution for IoT needs. It integrates a bundled set of services to connect, capture, register, analyze, and archive your IoT devices and data.

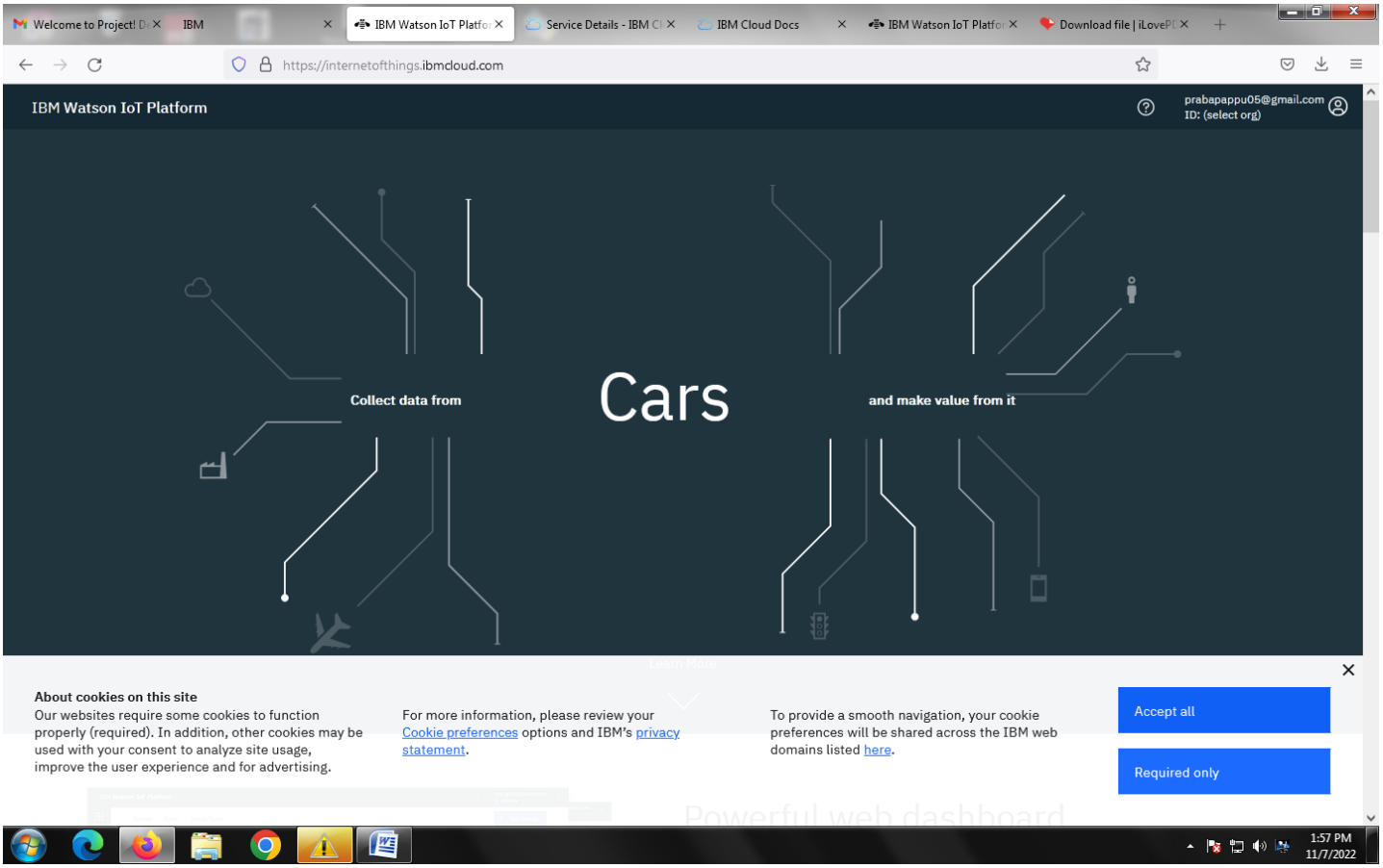
This is part two of a two-part series. This document is a simple, easy to follow process toconnect a device to IBM Watson IoT Platform. It will go through connecting via an MQTT connection.

I. Login to IBM Watson IoT Platform to Verify Connection

STEP1:
Go to URL - https://internetofthings.ibmcloud.com/

Click Sign in

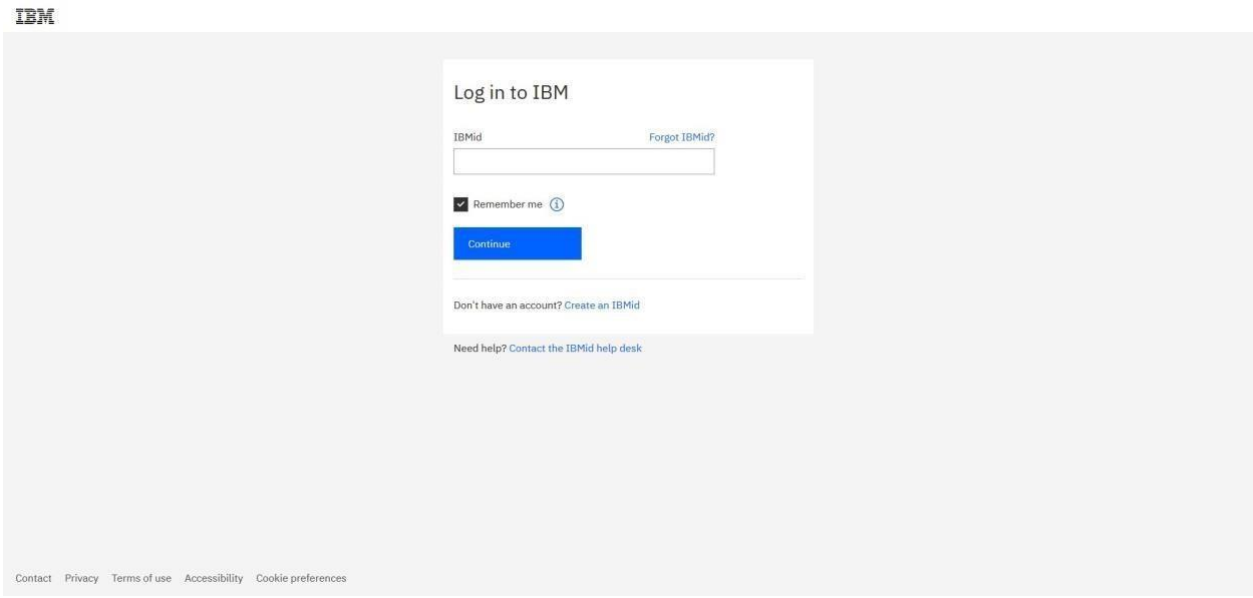




STEP2:

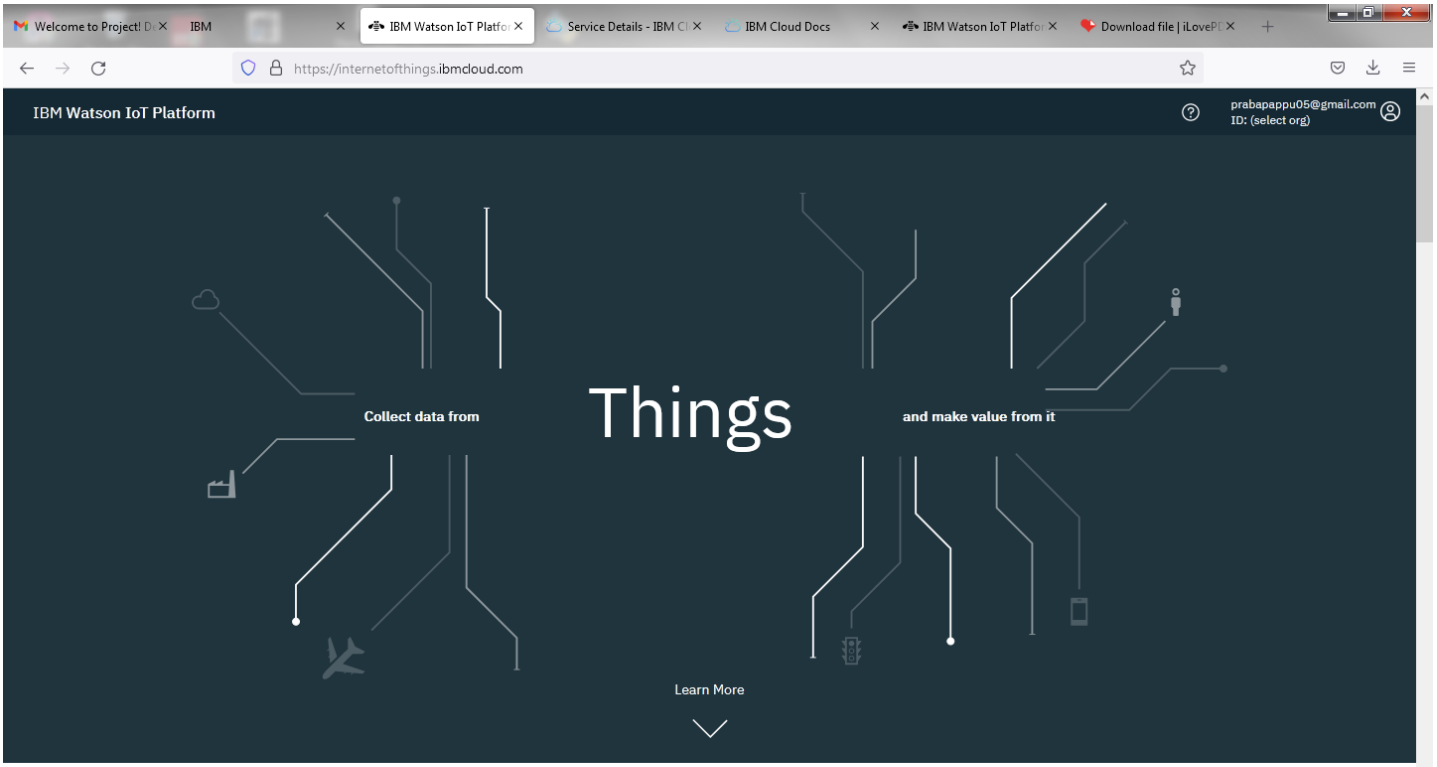
Enter an IBMid and click Continue

(Click Remember Me if you want)

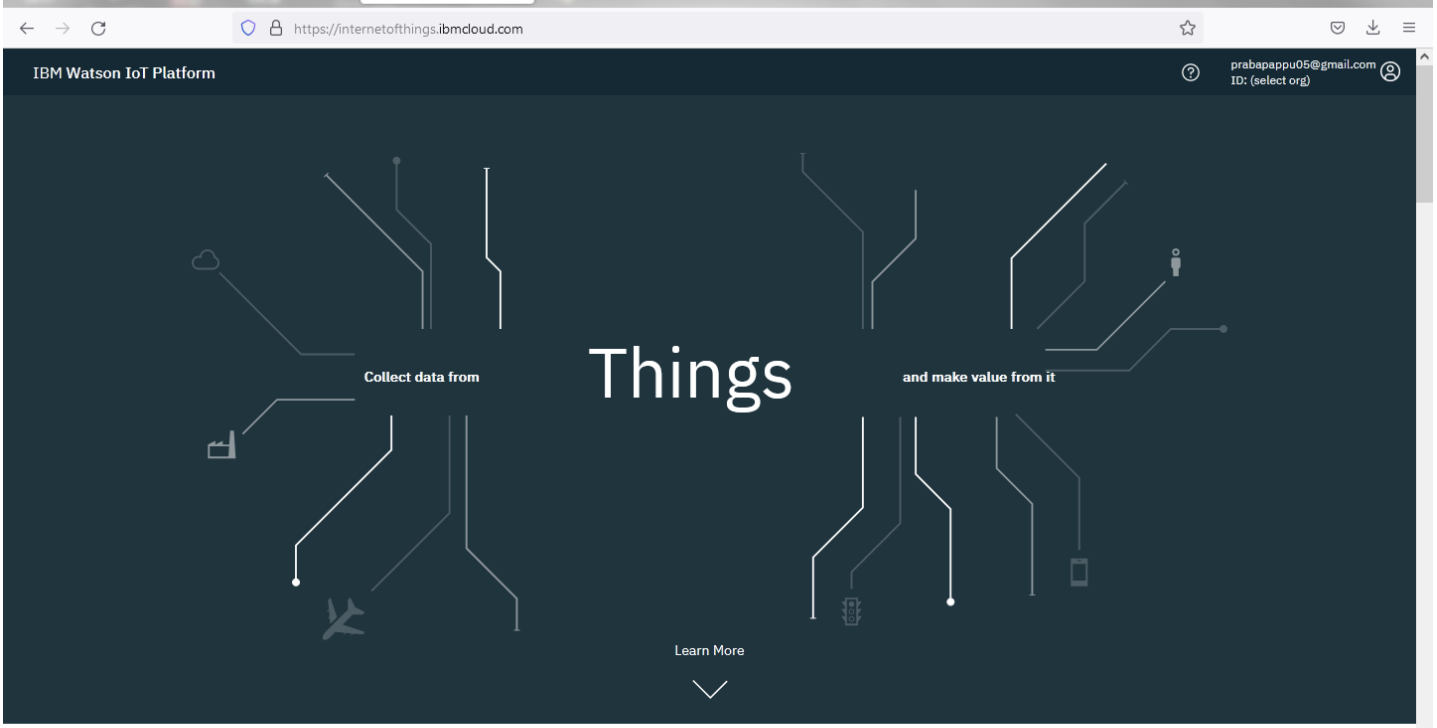


Enter the Password and click Login
(Click Remember Me if you want)

You are now logged into IBM Watson IoT Platform


IBM Watson IoT Platform
prabapappu05@gmail.com
ID: (select org)
Collect data from
Things
and make value from it
Learn More

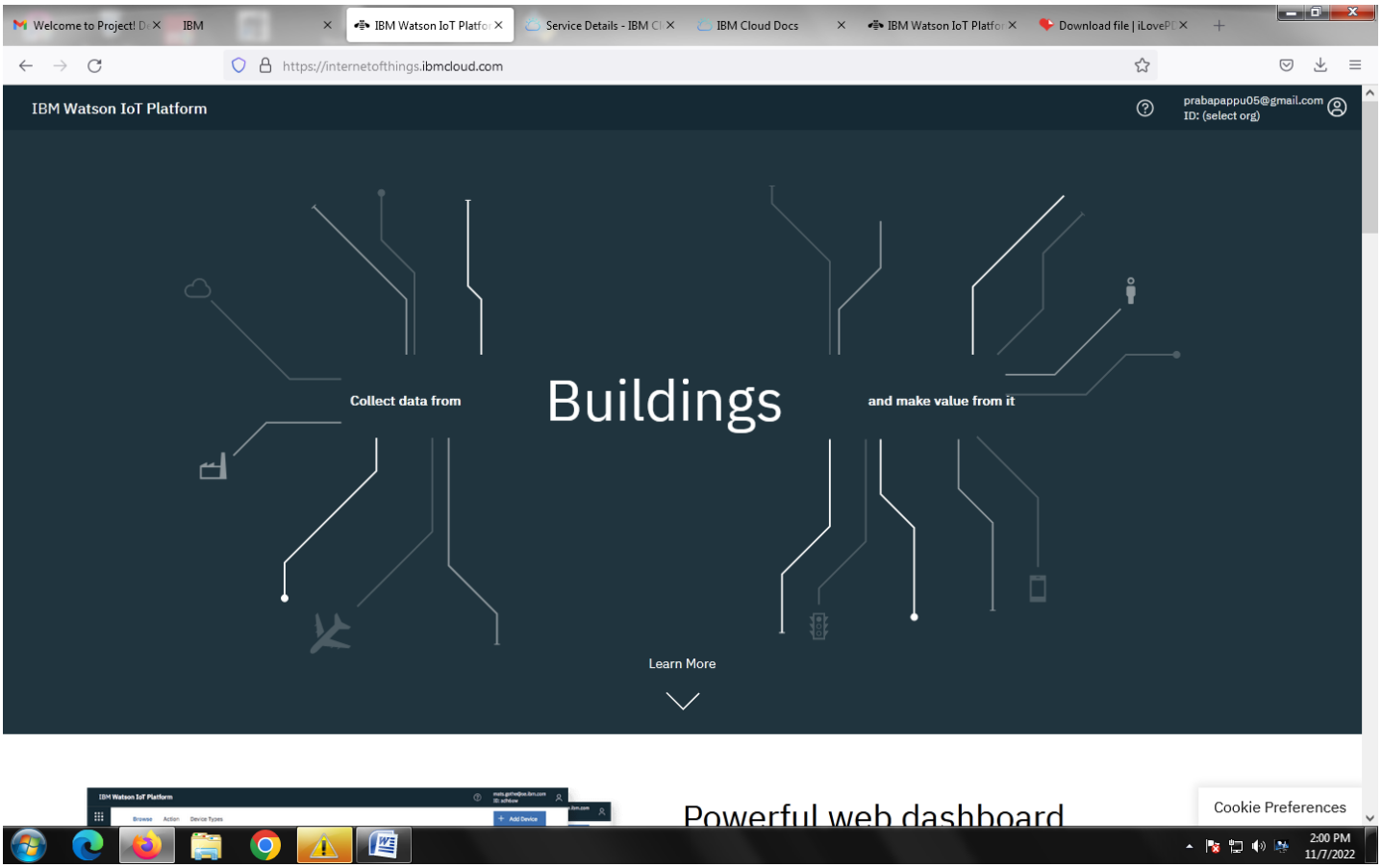

Powerful web dashboard
Cookie Preferences
2:03 PM
11/7/2022


IBM Watson IoT Platform
prabapappu05@gmail.com
ID: (select org)
Collect data from
Things
and make value from it
Learn More


Powerful web dashboard
Cookie Preferences
2:03 PM
11/7/2022

STEP3:

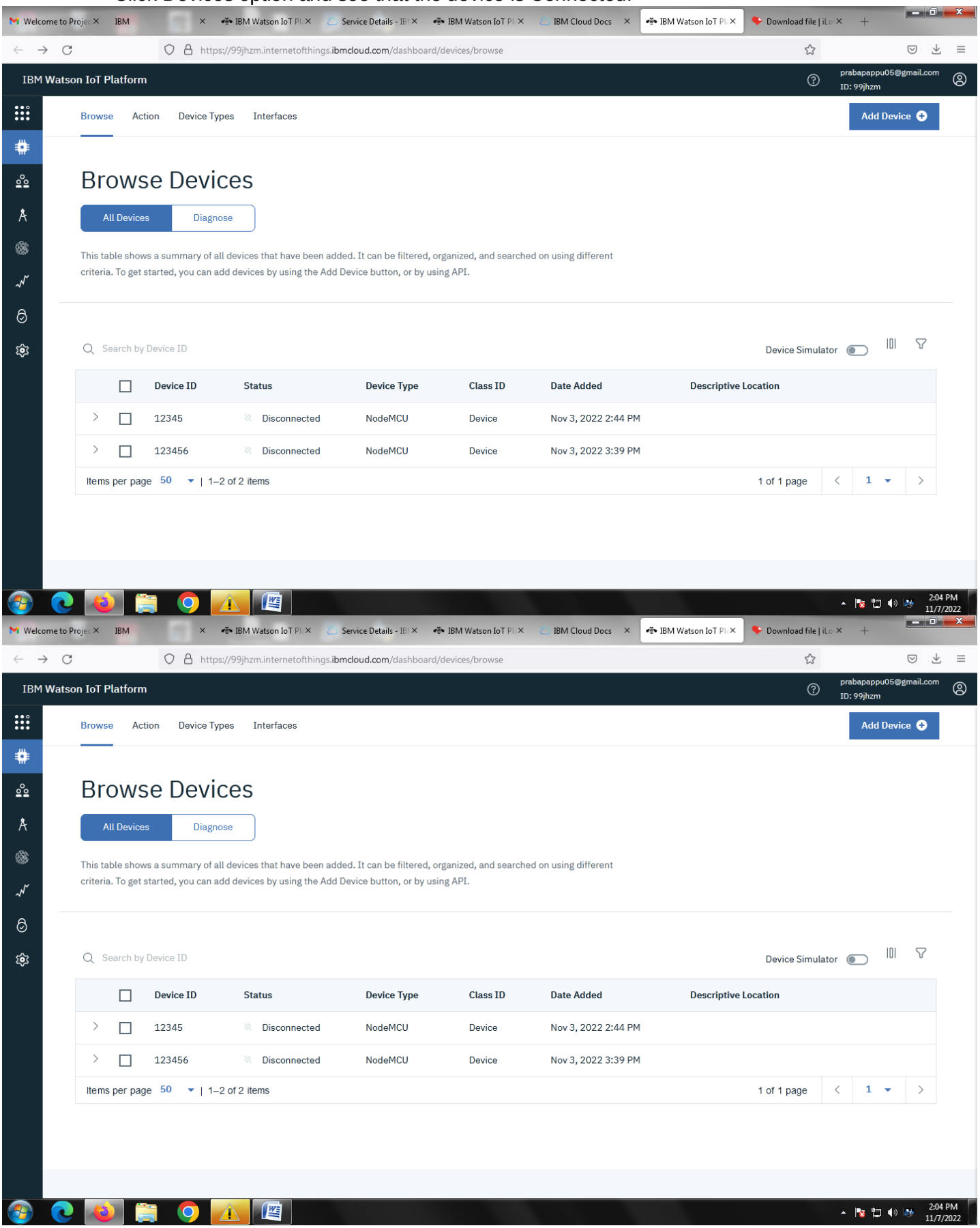
Click select org

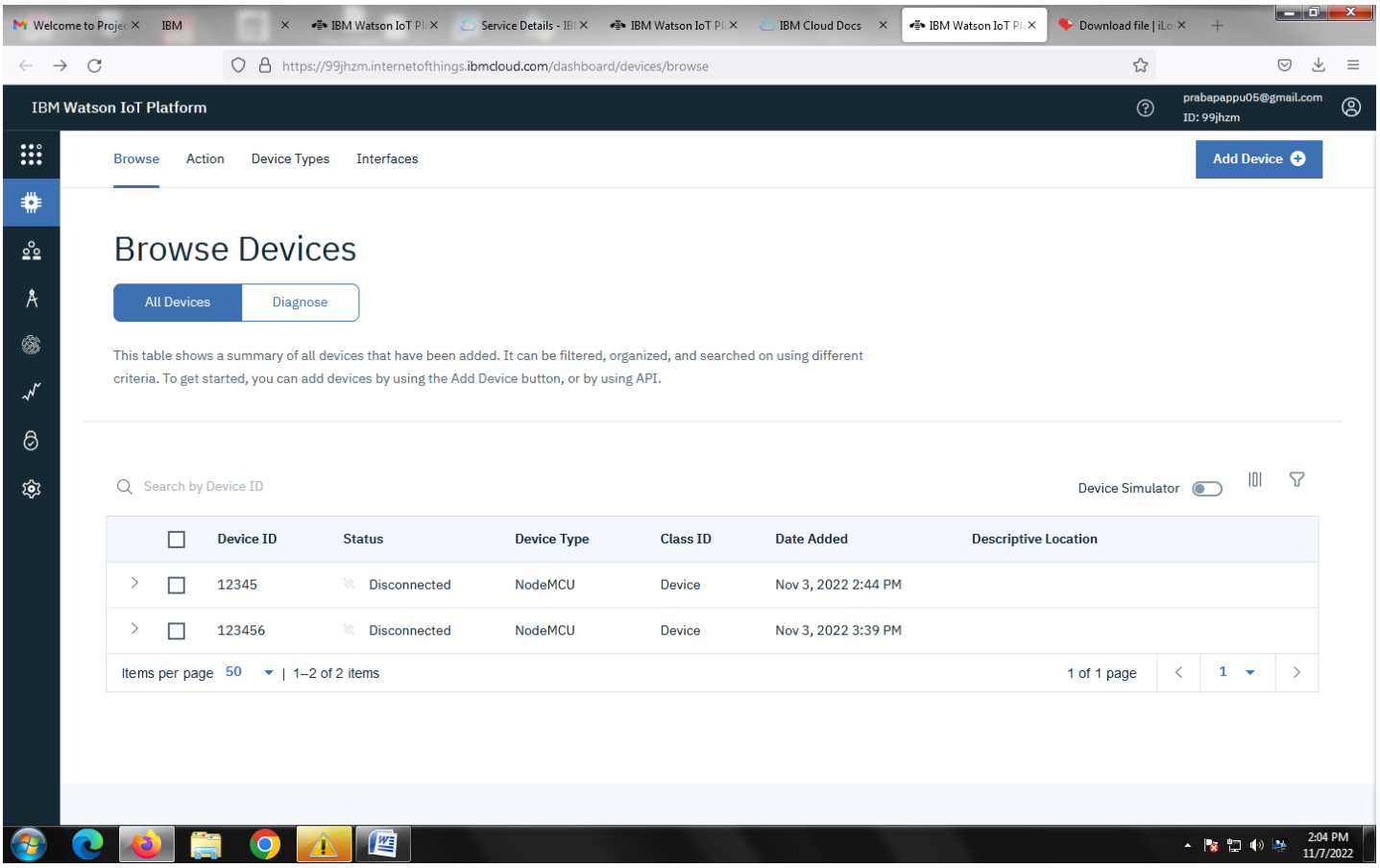


Select the org

STEP4:

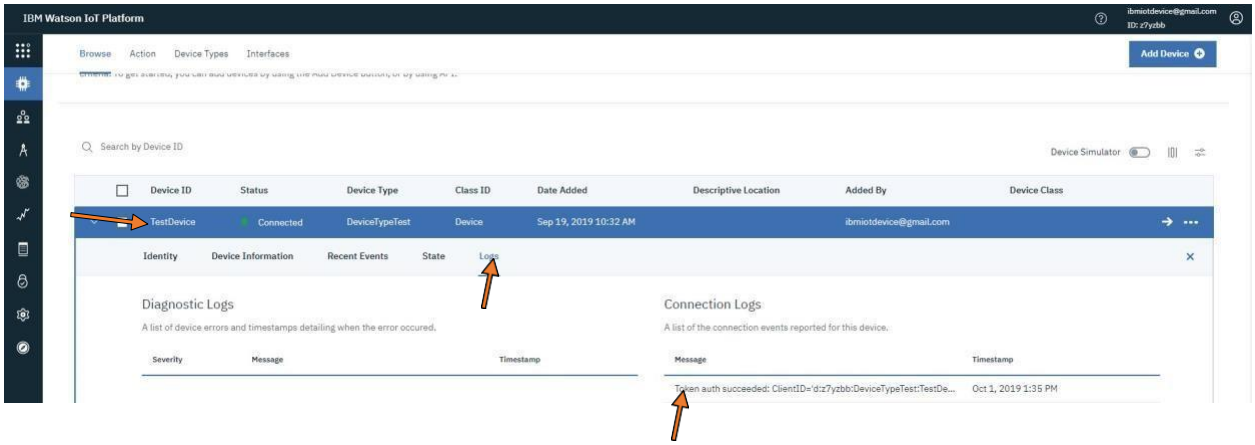
Click Devices option and see that the device is Connected.



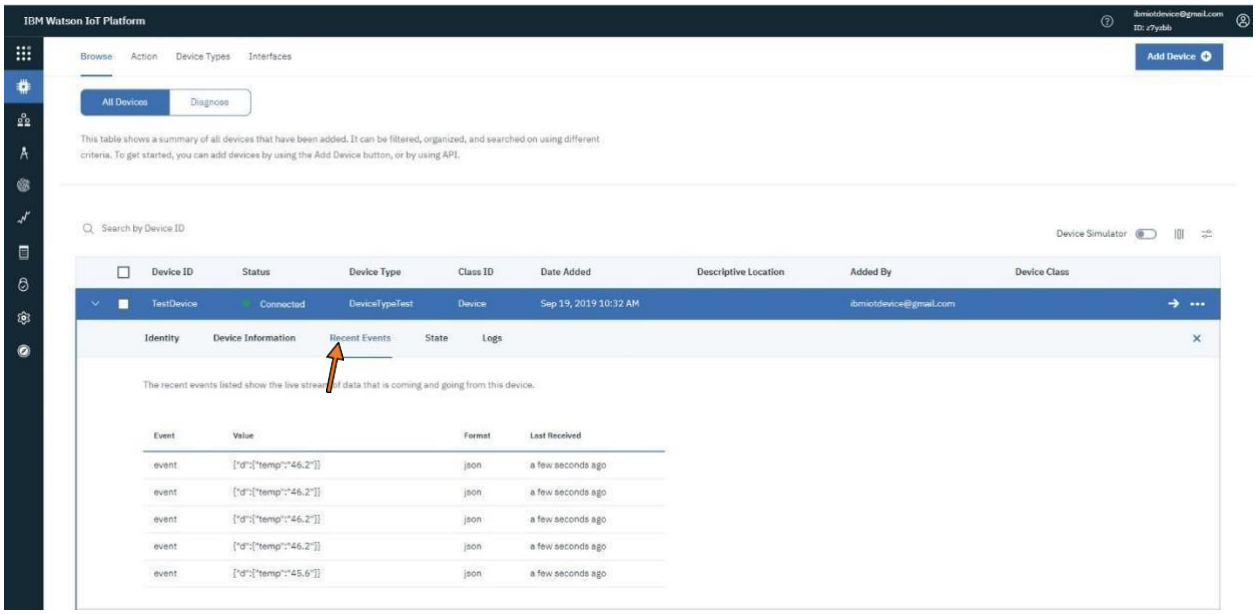


STEP 5:

Click the device and Logs and see that the connection was made

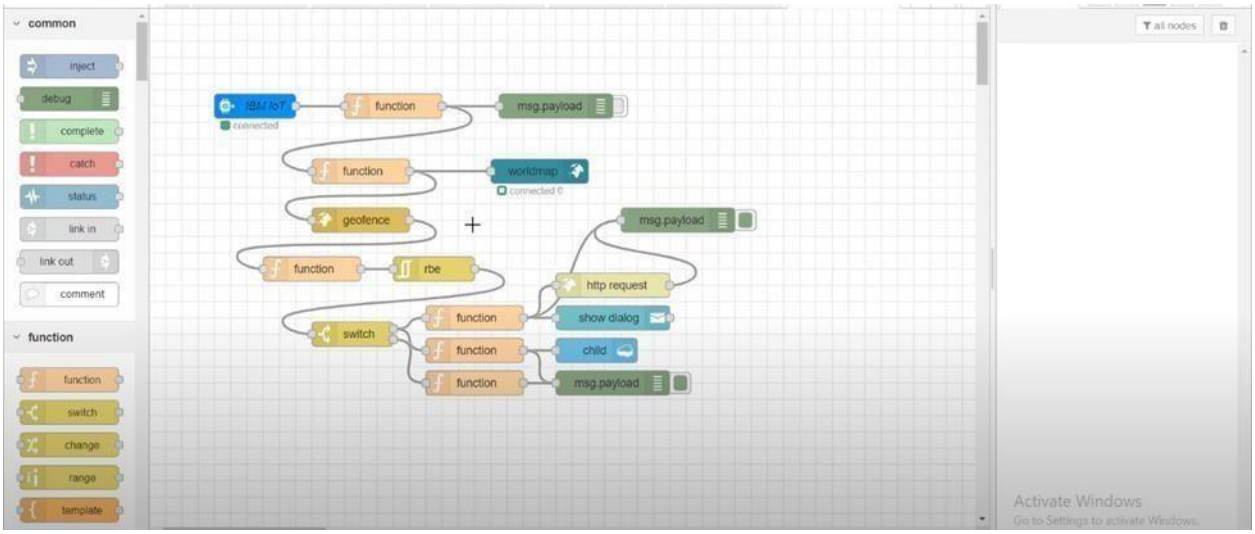


STEP6:
Click Recent Events and see the data that was sent from the device to IBM WatsonIoT Platform .



NODE RED SERVICE

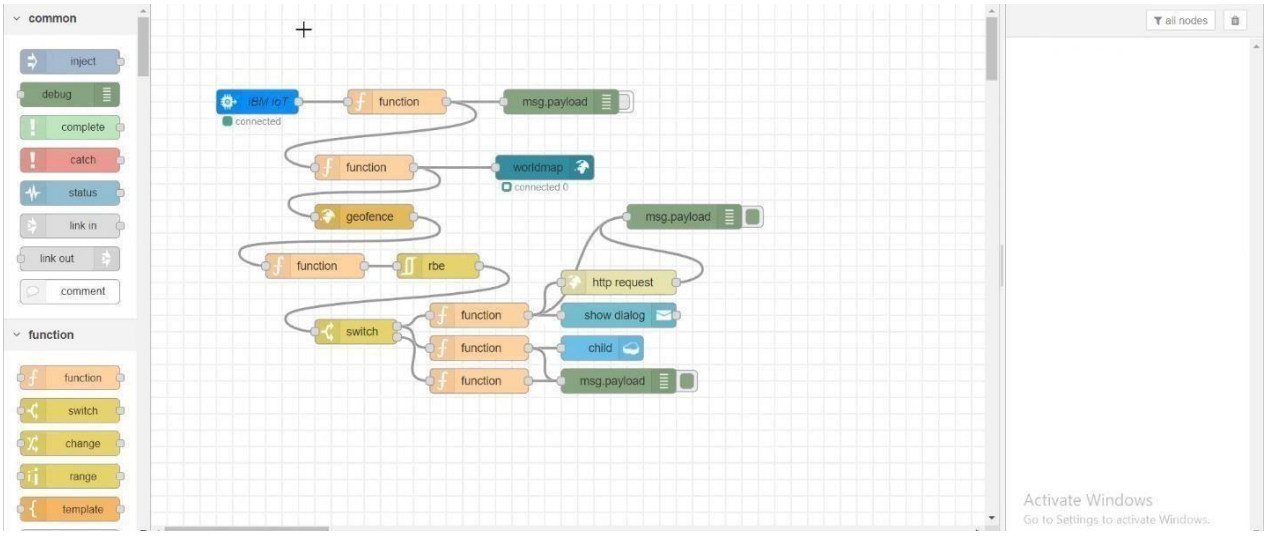
Step 1: Connect the blocks.




```
import json
import wiotp.sdk.device
import time

myConfig = {
    "identity": {
        "orgId": "hj5fmy",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
        name= "Smartbridge"
        #in area location

        latitude= 17.4225176
        longitude= 78.5458842

        #out area location

        #latitude= 17.4219272
        #longitude= 78.5488783
        myData={'name': name, 'lat':latitude,'lon':longitude}
        client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
        print("Data published to IBM IoT platfrom: ",myData)
        time.sleep(5)

client.disconnect()
```

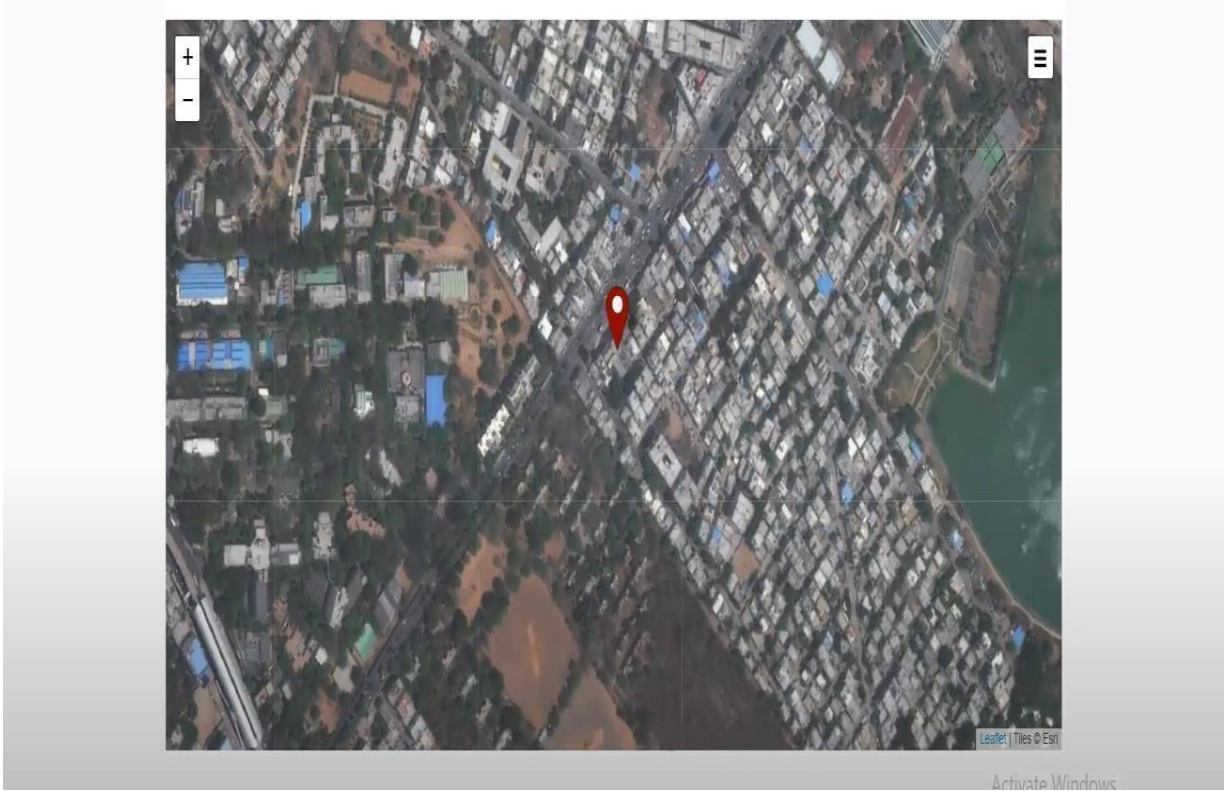
Step

Step 3: Click the geo-fence node.

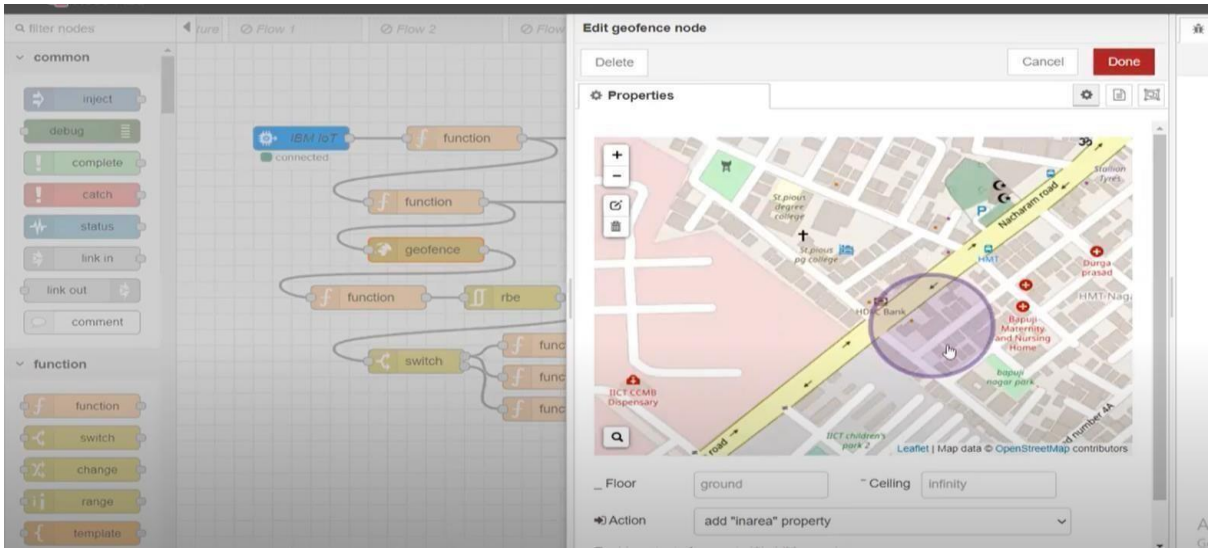


Step 4: Create the geo-fence area in the map.

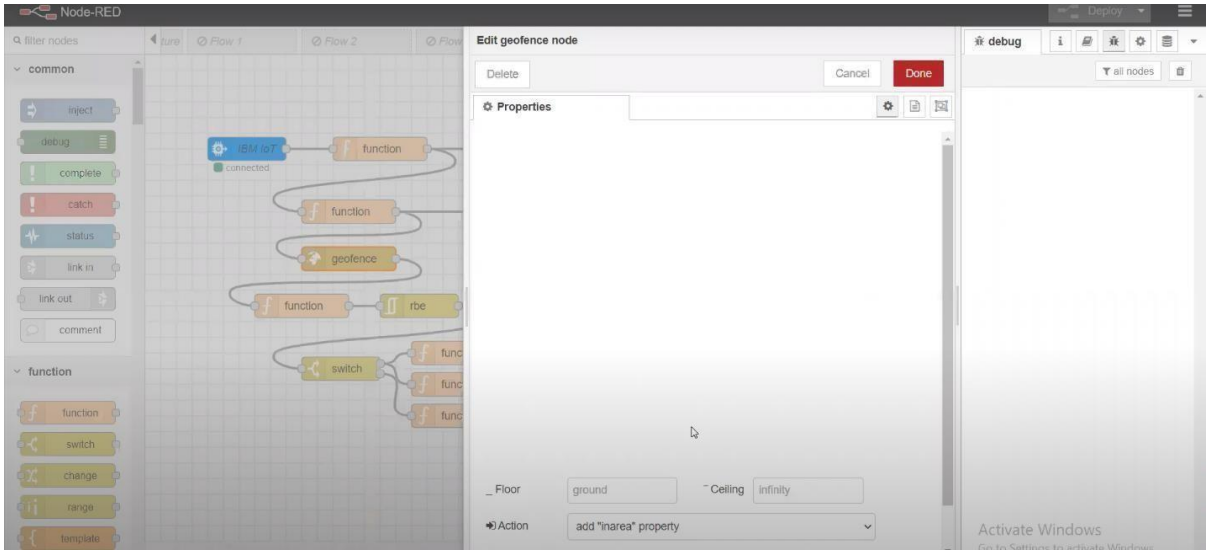
Step



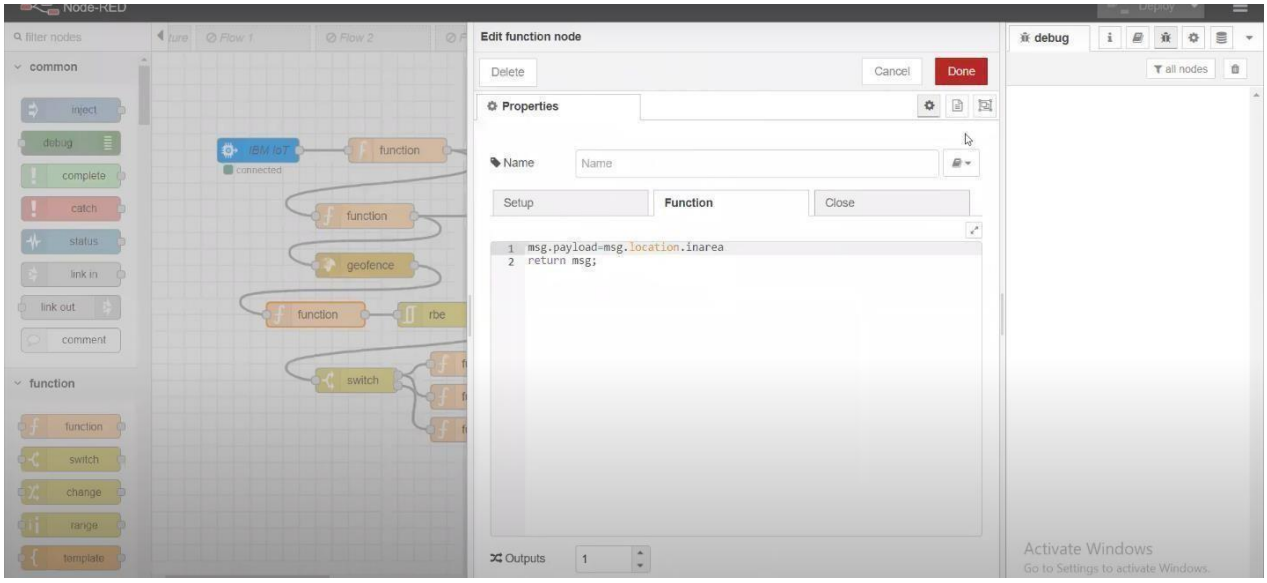
5: Create geo-fence in a particular area.



Step

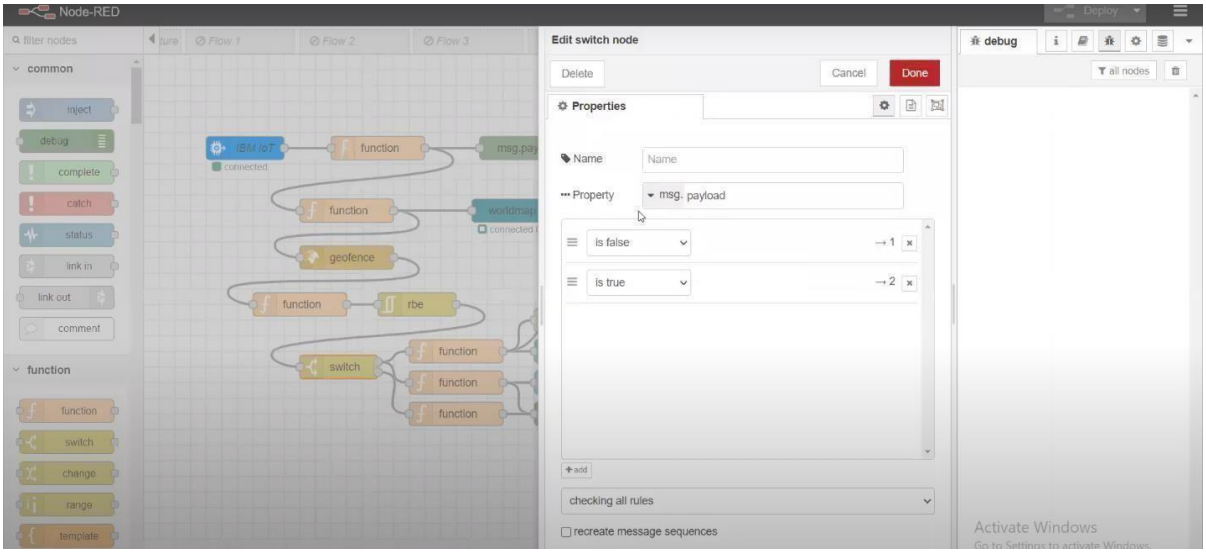


Step 7: Select the message payload.

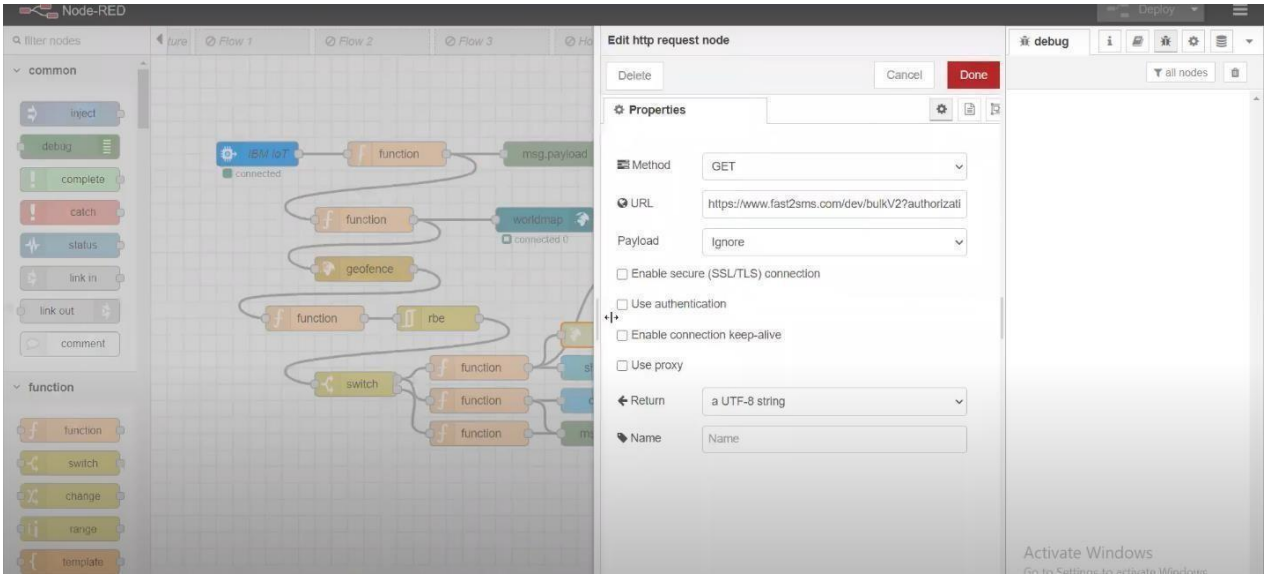


Step 8: To identify the person in area.

Step

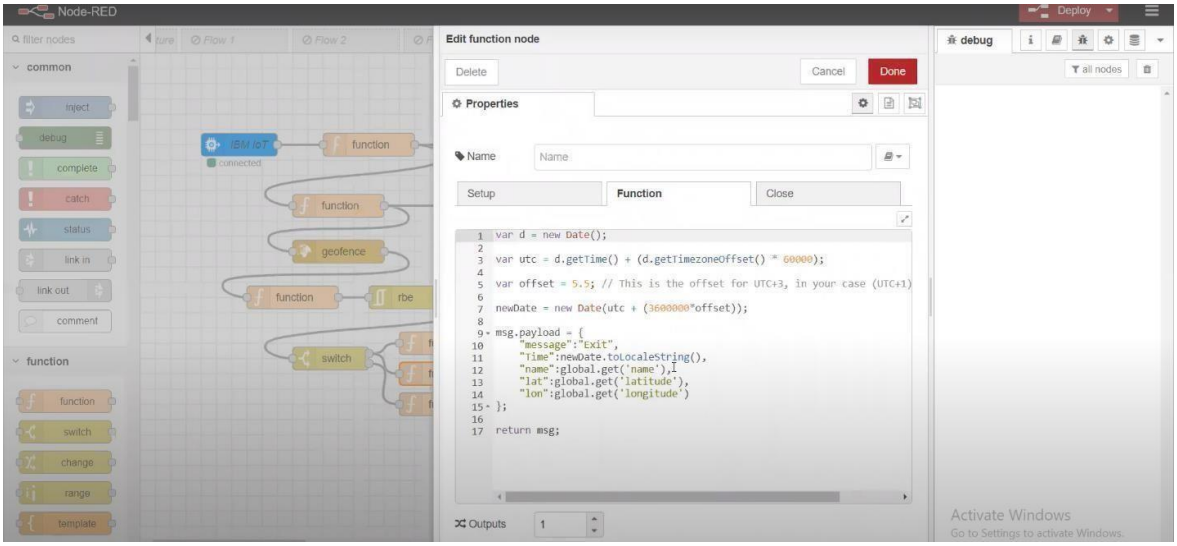


9: Select the http request to send message to parent or guardian.



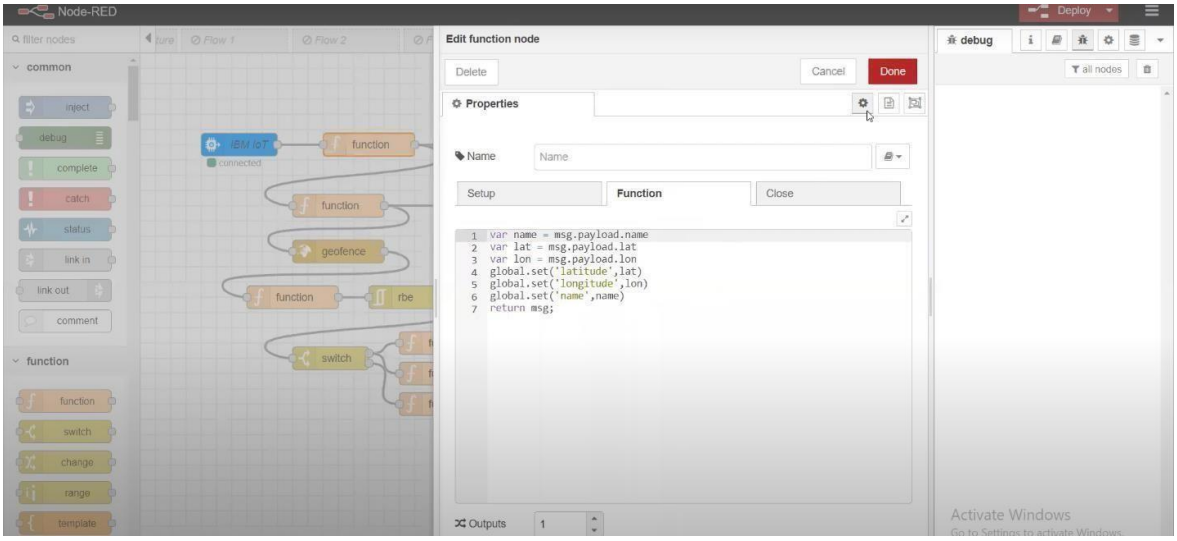
Step 10: For sending the message with time.

Step

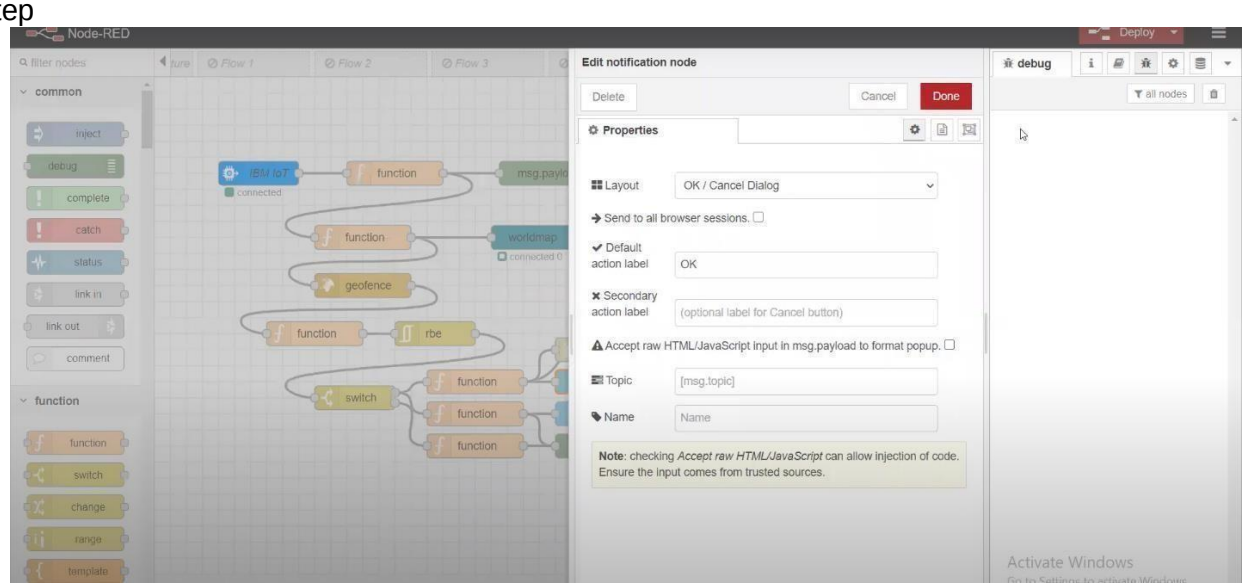

Node-RED
Deploy
filter nodes
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
range
template
IBM IoT
connected
function
function
geofence
function
rbe
switch
Edit function node
Delete
Cancel
Done
Properties
Name
Name
Setup
Function
Close
1 var d = new Date();
2
3 var utc = d.getTime() + (d.getTimezoneOffset() * 60000);
4
5 var offset = 5.5; // This is the offset for UTC+3, in your case (UTC+1)
6
7 newDate = new Date(utc + (3600000*offset));
8
9 msg.payload = {
10 "message":"Exit",
11 "Time":newDate.toLocaleString(),
12 "name":global.get('name'),
13 "lat":global.get('latitude'),
14 "lon":global.get('longitude')
15 };
16
17 return msg;
Outputs
1
debug
all nodes
Activate Windows
Go to Settings to activate Windows.

Step

10: Click show dialog for notifying the popup alert.



Step

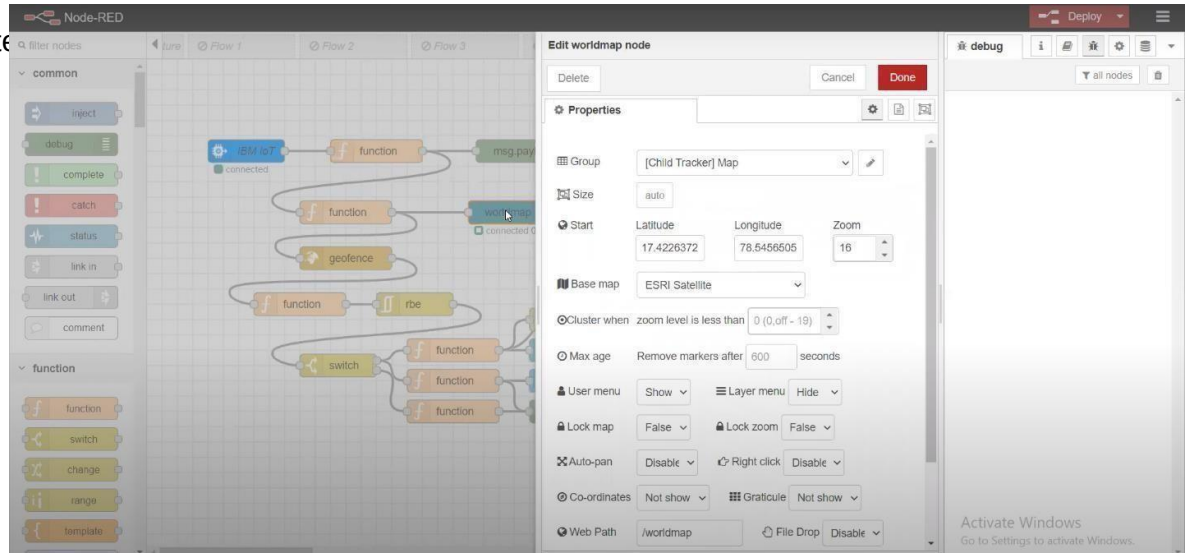

Node-RED
Deploy
filter nodes
Flow 1
Flow 2
Flow 3
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
range
template
IBM IoT
connected
function
msg.payl
function
worldmap
connected 0
geofence
function
rbe
function
switch
function
function
Edit notification node
Delete
Cancel
Done
Properties
Layout
OK / Cancel Dialog
Send to all browser sessions.
Default action label
OK
Secondary action label
(optional label for Cancel button)
Accept raw HTML/JavaScript input in msg.payload to format popup.
Topic
[msg.topic]
Name
Name
Note: checking Accept raw HTML/JavaScript can allow injection of code. Ensure the input comes from trusted sources.
debug
all nodes
Activate Windows

Step

Step 11: Create another payload and to pass the data to geo-fence and world map.

12: Click the world map to see the location.

Ste



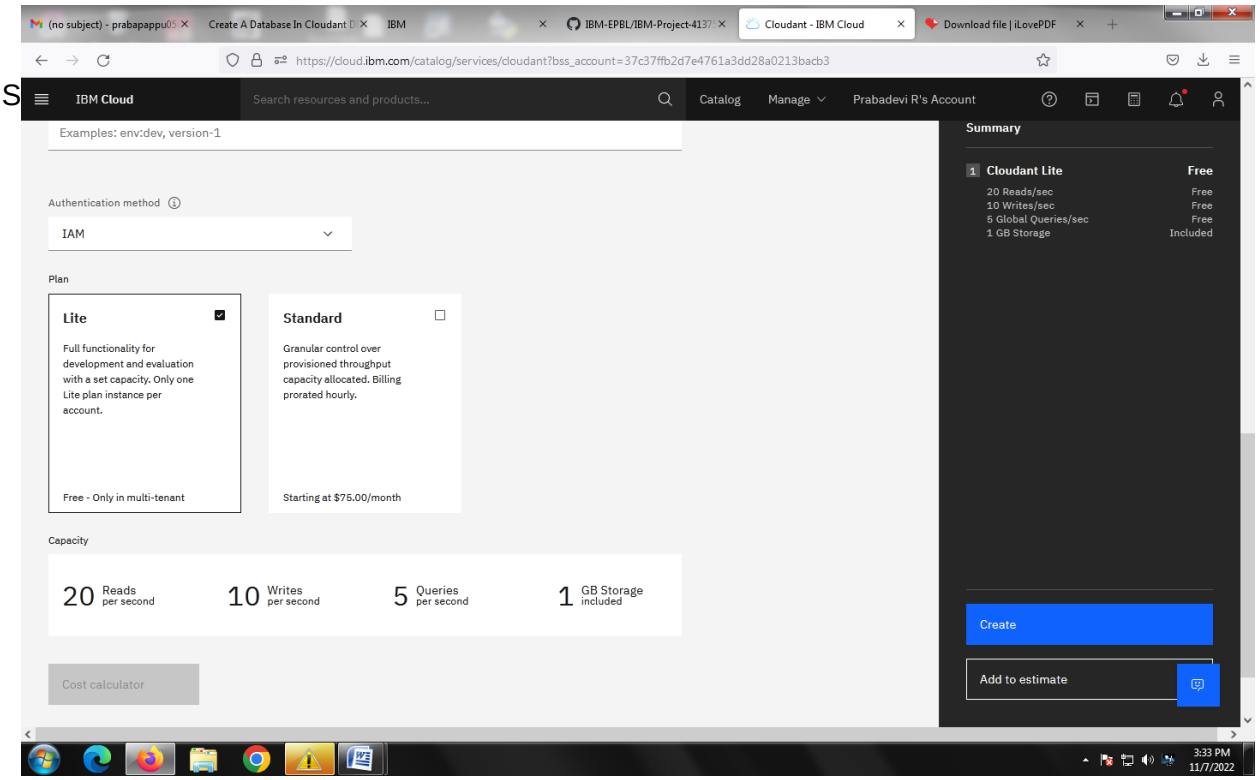
Create A Database In Cloudant DB

Aim: To create a database in Cloudant DB to store location data.

Steps followed:

- Logged in to IBM Cloud account
- Navigated to `./resources`
- Clicked on the "Create Resource +" button
- Searched for "Cloudant"
- Chose the "Lite Version" and clicked on "Create"

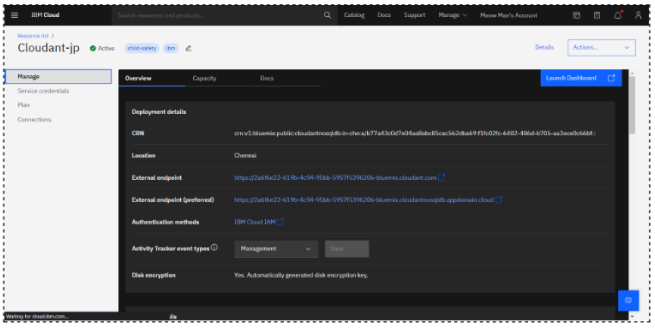
S



- The Cloudant database resource was created successfully

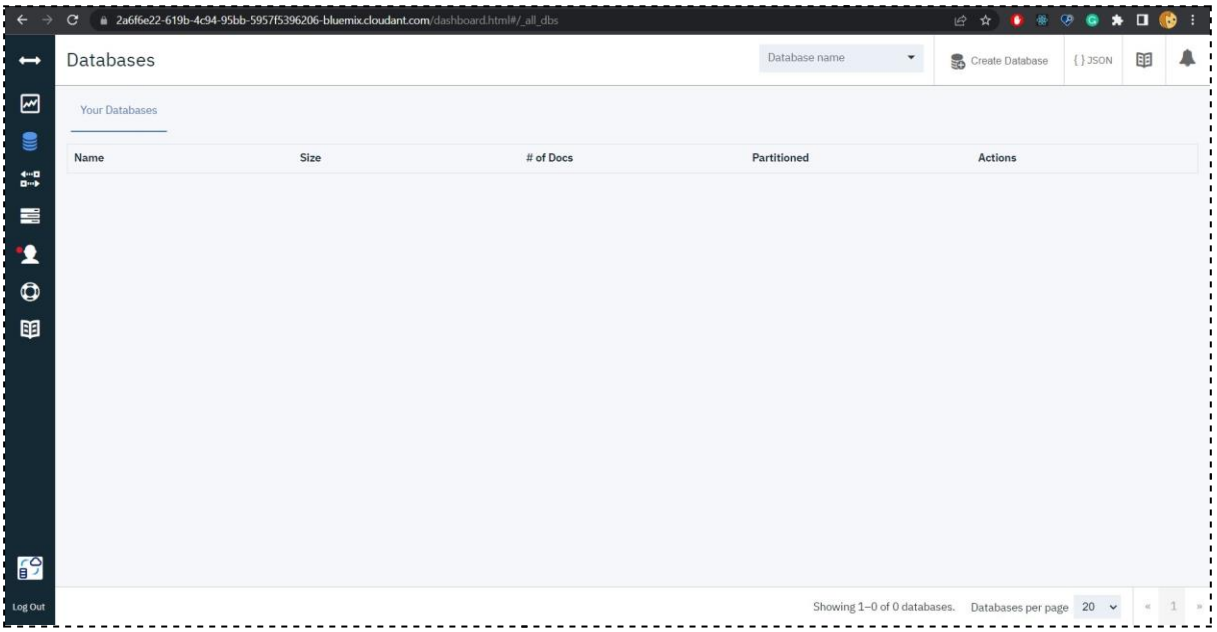
S

- The Cloudant database resource was created successfully

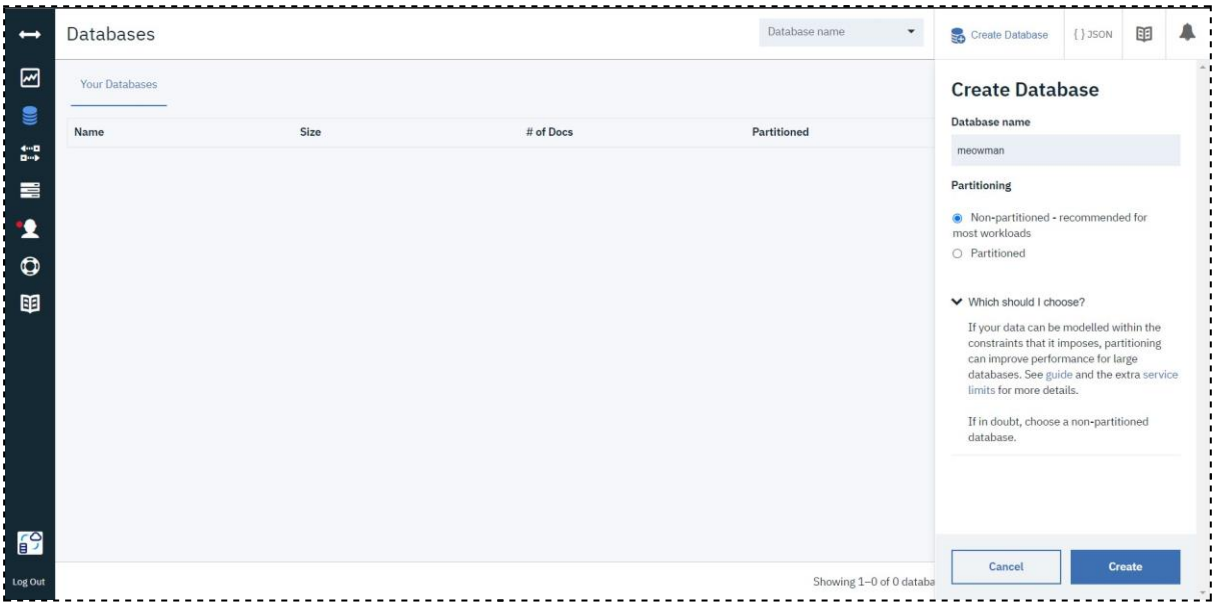


- Clicked on Launch Dashboard

Step



- Clicked on “Create Database”. Entered “PRABADEVI” as the databasename and the “Non-partitioned” option



- The database “PRABADEVI” was created successfully

Step

Result:

A database to store the location data was created successfully onCloudant DB