

Soldier Health Monitoring System Using IBM Cloud

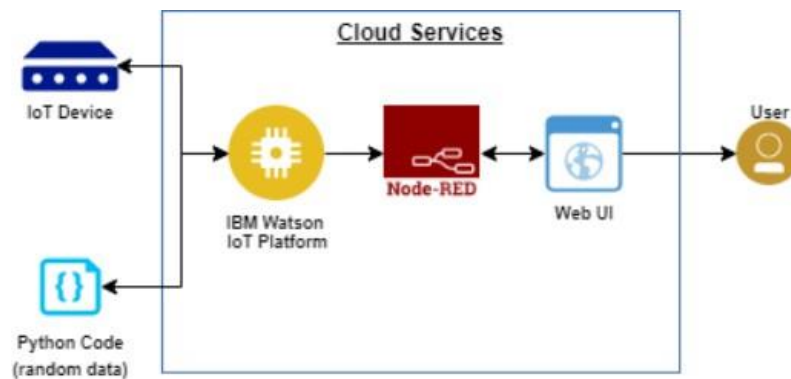
Skills Required: -

Python, IOT Application Development, IBM Cloud, IBM Watson

Project Description: -

Features:

- Continuous health monitoring of soldiers.
- Tracking the position of the soldiers.
- pulse rate and temperature are monitored by sending the sensor data to the cloud.
- Health conditions can be monitored through a web application.
- Admin can view the location of every soldier in the web application using maps.



Setup Environment

Install Python IDE

Step 1) To download and install Python, visit the official website of Python <https://www.python.org/downloads/> and choose your version.

Step 2) Once the download is completed, run the .exe file to install Python. Now click on Install Now.

Step 3) You can see Python installing at this point.

Step 4) When it finishes, you can see a screen that says the Setup was successful. Now click on "Close".

Create IBM Account

Step 1) Create an IBM Cloud (Trial) Account and login using the credentials.

Step 2) Open Dashboard and Explore

Step 3) For the first time the Resources Card would be empty. This is where all the Services and Tools are Displayed.

Step 4) To get or create Services click Catalog option from the top Menu. You can search any resources which are provided by the Cloud Service.

Step 5) To manage Account Settings go to Accounts option in the Manage drop down menu

Step 6) Open Account Settings and generate a key using API and Paste it in the Subscription and feature codes Section and select Trial Account.

Step 7) Rest all services would be visible on the Dashboard just as same

Create Node-Red Application

Step1) Find the Node-RED Starter in the IBM Cloud catalogue

Step2) Create your application

Step3) Enable the Continuous Delivery feature

Step4) Open the Node-RED application

Step5) Configure your Node-RED application

Step6) Add extra nodes to your Node-RED palette

Create IBM Watson IOT Platform

Step1) Log in to IBM Cloud.

Step2) Click Create resource + in the top right corner.

Step3) In the Services menu on the left, select the Internet of Things category.

Step4) Click the Internet of Things Platform card.

Step5) Select the following options and click Create.

- Region: Choose the one closest to your location.
- Plan: Lite.

Step6) In the next screen, click Launch to open the IBM Watson IoT Platform.

Step7) In the Browse Devices screen, click on Create a device.

Step8) In the Add Device screen, fill in the form with the Identity information and then click Next.

- Device Type: android.
- Device ID: phone.

Step9) In the Add Device screen, the Device Information form contains optional fields that are not used for authentication. Fill in whatever you see fit, just for the purpose of differentiating between devices, then click Next.

Step10) In the Add Device screen, you can enter a custom Authentication Token or leave it blank for automatically generating one, then click Next.

Step11) Confirm the information displayed before clicking Finish.

Step12) Write down the Device Credentials information. You cannot retrieve the token after you leave this page.

- Organization ID: this value will be referred to as ORG_ID.
- Device Type: this value will be referred to as DEV_TYPE.
- Device ID: this value will be referred to as DEV_ID.
- Authentication Method: this value will be referred to as AUTH_METHOD.
- Authentication Token: this value will be referred to as AUTH_TOKEN.

Create Cloudbant DB (Database Service) In IBM Cloud

Step 1: Creating a service instance

- Log in to your IBM Cloud account, and click Create resource
- Type Cloudbant in the Search bar and click to open it:
- Select an offering and an environment.
- Type an instance name
- Select your plan
- To create the service click create

Step 2: Creating service credentials

- To create the connection information that your application needs to connect to the instance, click New credential.
- Enter a name for the new credential in the Add new credential window.
- Accept the Manager role.
- (Optional) Create a service ID or have one automatically generated for you.

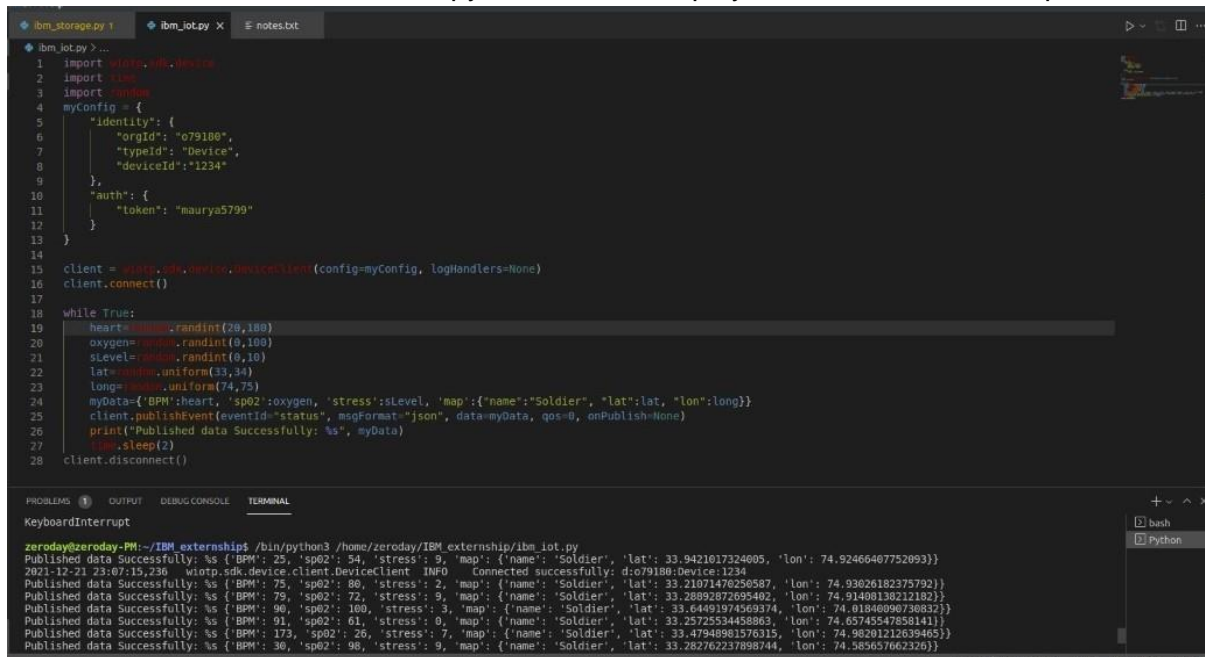
- (Optional) Add inline configuration parameters. This parameter isn't used by IBM Cloudant service credentials, so ignore it.
- Click Add.
- To see the credentials that are required to access the service, click the chevron.

Setup Hardware and Develop the Code

Creating A Code Snippet

Create a code snippet using python to:

- Send random location of soldiers to be tracked using multiple python programs.
- Sending soldier health data like temperature, pulse and blood pressure to the cloud.
- The sensor data sent from python shall be displayed in the web UI on Map.



```

1 import sys, os, random
2 import time
3 import random
4 myConfig = {
5     "identity": {
6         "orgId": "o79180",
7         "typeId": "Device",
8         "deviceId": "1234"
9     },
10    "auth": {
11        "token": "maurya5799"
12    }
13 }
14
15 client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
16 client.connect()
17
18 while True:
19     heart=random.randint(70,180)
20     oxygen=random.randint(0,100)
21     sLevel=random.randint(0,10)
22     lat=random.uniform(33,34)
23     long=random.uniform(74,75)
24     myData={"BPM":heart, 'sp02':oxygen, 'stress':sLevel, 'map':{'name':'Soldier', 'lat':lat, 'lon':long}}
25     client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
26     print("Published data Successfully: %s", myData)
27     time.sleep(2)
28 client.disconnect()
  
```

```

zeroday@zeroday-PM:~/IBM externship/bin/python3 /home/zeroday/IBM externship/ibm_iot.py
Published data Successfully: %s {'BPM': 25, 'sp02': 54, 'stress': 9, 'map': {'name': 'Soldier', 'lat': 33.9421017324005, 'lon': 74.92466407752093}}
2021-12-21 23:07:15.236 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:o79180:Device:1234
Published data Successfully: %s {'BPM': 75, 'sp02': 80, 'stress': 2, 'map': {'name': 'Soldier', 'lat': 33.21071470250587, 'lon': 74.93026182375792}}
Published data Successfully: %s {'BPM': 79, 'sp02': 72, 'stress': 9, 'map': {'name': 'Soldier', 'lat': 33.28892872695402, 'lon': 74.91400138212182}}
Published data Successfully: %s {'BPM': 90, 'sp02': 100, 'stress': 3, 'map': {'name': 'Soldier', 'lat': 33.64491974569374, 'lon': 74.01840090736832}}
Published data Successfully: %s {'BPM': 91, 'sp02': 61, 'stress': 0, 'map': {'name': 'Soldier', 'lat': 33.25725534458863, 'lon': 74.65745547858141}}
Published data Successfully: %s {'BPM': 173, 'sp02': 20, 'stress': 7, 'map': {'name': 'Soldier', 'lat': 33.47948981376315, 'lon': 74.90201212639463}}
Published data Successfully: %s {'BPM': 30, 'sp02': 90, 'stress': 9, 'map': {'name': 'Soldier', 'lat': 33.282762237898744, 'lon': 74.385657662326}}
  
```

Publish Data to The IBM Cloud

To publish one or more of any type of asset from a project to a catalogue:

- From the project, select the assets that you want to publish to the catalogue, and click Publish.
- Choose the catalogue and fill in the asset properties. If any of the assets require a connection, those connections are listed and will be added to the catalogue as connection assets, including connections to IBM Cloud Object Storage for uploaded files. If you don't want to publish a connection, you can remove it. All assets that require the removed connection are removed from the list of assets to publish.

- If you are publishing a dashboard asset, you can provide a preview by taking and attaching screen captures.
- Click Publish. The assets are added to the catalogue and you are the owner of them.

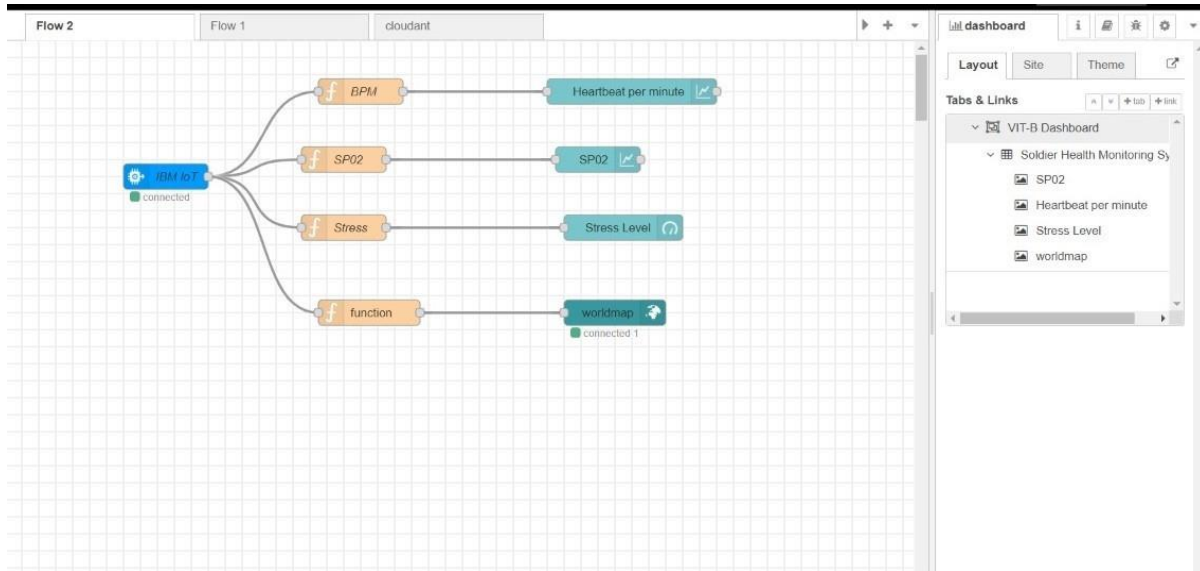
Python code

```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "o79180",
        "typeId": "Device",
        "deviceId": "1234"
    },
    "auth": {
        "token": "maurya5799"
    }
}
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    heart=random.randint(20,180)
    oxygen=random.randint(0,100)
    sLevel=random.randint(0,10)
    lat=random.uniform(33,34)
    long=random.uniform(74,75)
    myData={'BPM':heart, 'spO2':oxygen, 'stress':sLevel,
'map':{'name':"Soldier", "lat":lat, "lon":long}}
    client.publishEvent(eventId="status", msgFormat="json", data=myData,
qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    time.sleep(2)
client.disconnect()
```

Building Web App

Create Node Red Flow to Get Data from Device

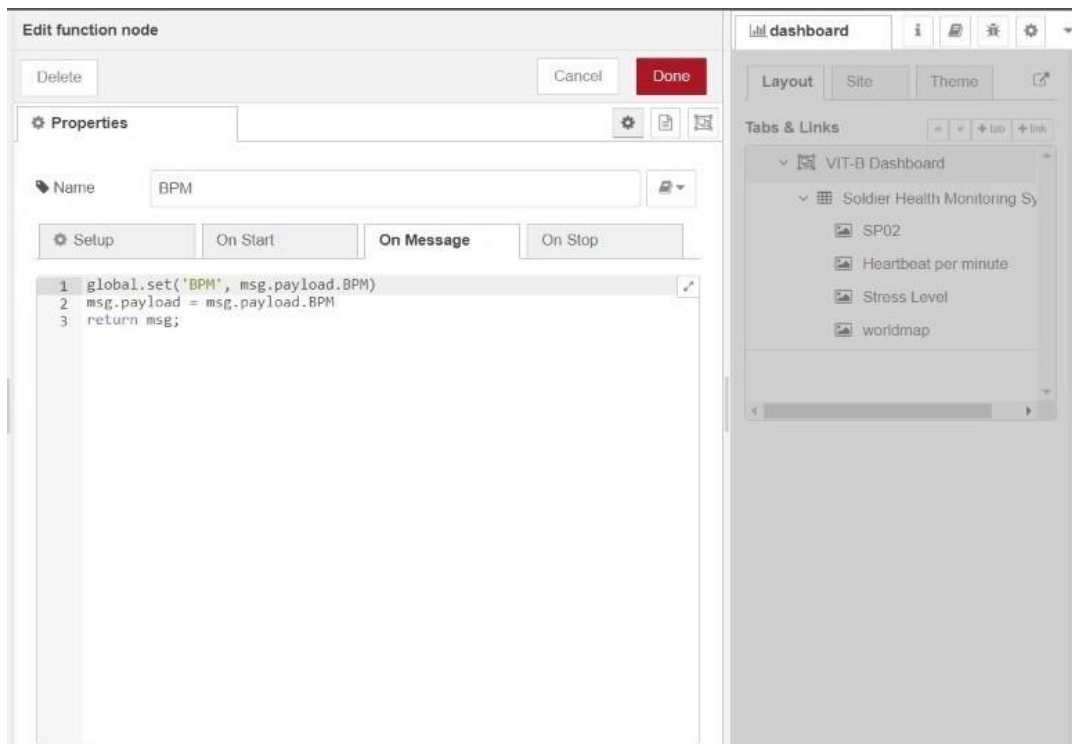
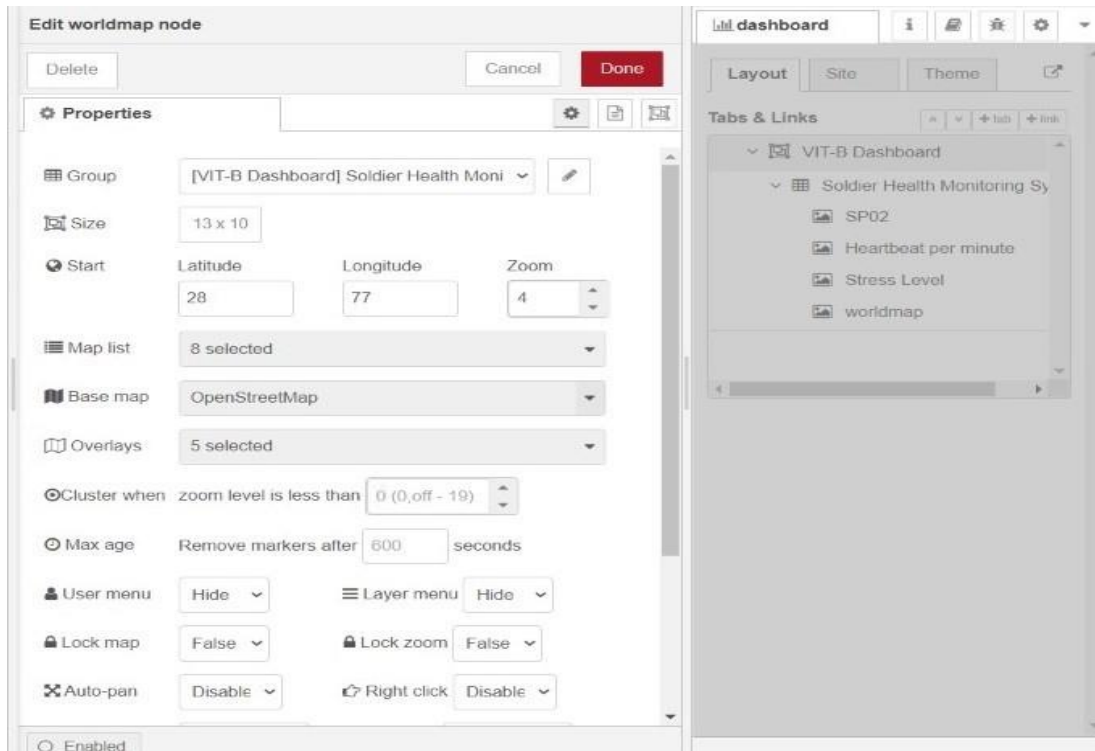


Use Dashboard Nodes for Creating UI (Web App)

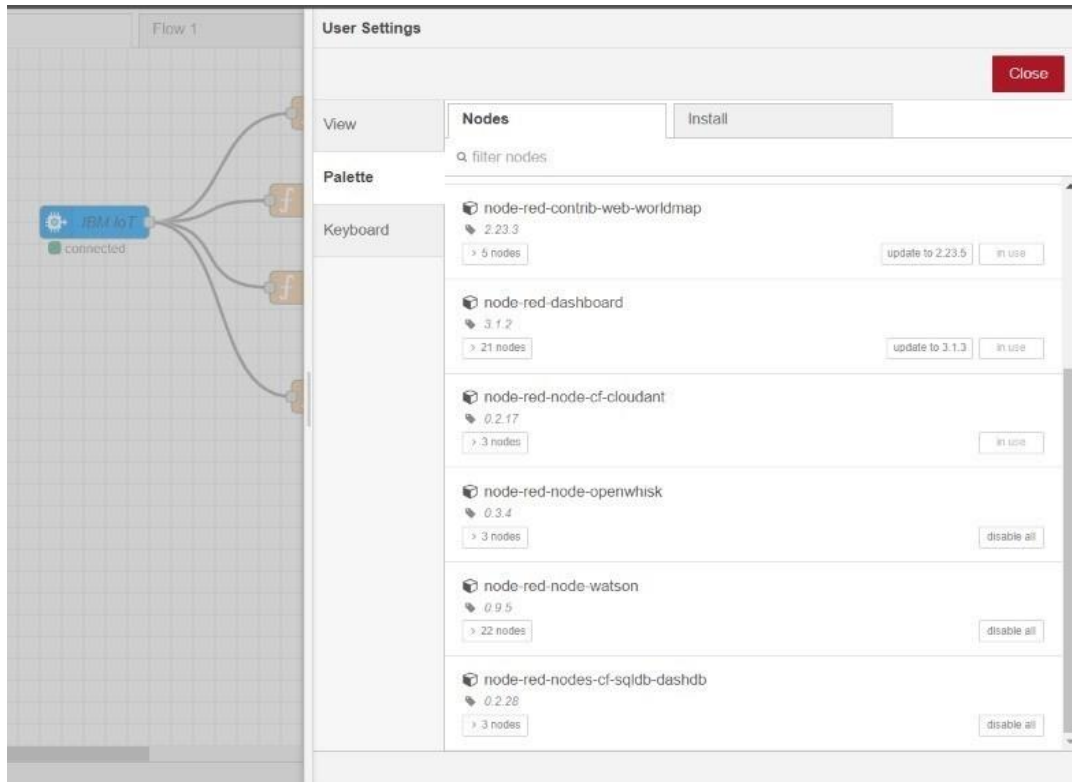
Dashboard interface showing device information and recent events.

Device ID	Status	Device Type	Class ID	Date Added	Description Location	Added by
1234	Connected	Device	Device	10 Dec 2021 10:37		pratul.maurya2019@vitbhopal.ac.in

Event	Value	Format	Last Received
status	{"BPM":132,"sp02":20,"stress":2,"map":{"name":...}}	json	a few seconds ago
status	{"BPM":132,"sp02":95,"stress":6,"map":{"name":...}}	json	a few seconds ago
status	{"BPM":56,"sp02":94,"stress":3,"map":{"name":...}}	json	a few seconds ago
status	{"BPM":78,"sp02":91,"stress":8,"map":{"name":...}}	json	a few seconds ago
status	{"BPM":80,"sp02":98,"stress":0,"map":{"name":...}}	json	a few seconds ago



UI Nodes Installation



WEB APP Screenshots

