

Lab Center – Hands-On Lab – Part ONE

Session 1239 - IBM Think2020 IoT Lab

Hyper-Local Weather and Crop prediction using Watson: Setting up your personal weather station

Markus van Kempen – mvk@ca.ibm.com

20200504



DISCLAIMER

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results like those stated here.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies.



IV

A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at:
www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

OpenShift is a trademark of Red Hat, Inc.

UNIX is a registered trademark of The Open Group in the United States and other countries.

© 2020 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

We Value Your Feedback

Don't forget to submit your Think 2020 session and speaker feedback! Your feedback is very important to us – we use it to continually improve the conference.

Access the Think 2020 agenda tool to quickly submit surveys from your smartphone or laptop.

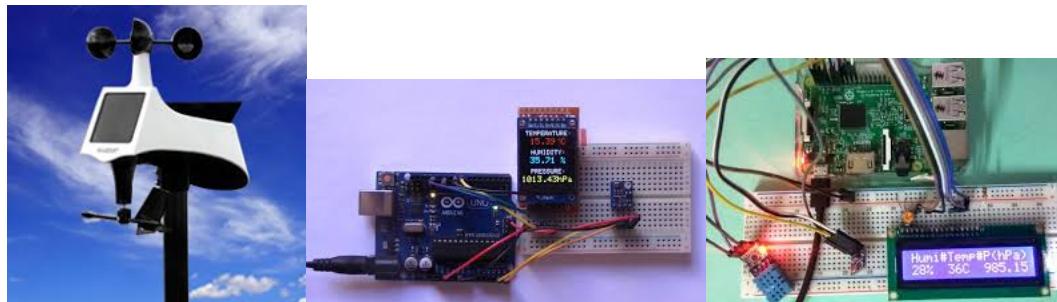
Table of Contents

1. Lab Setup Instructions	5
Prerequisite (details follow).....	5
Note:.....	5
1.1 LAB Part 1	6
1.2 Node-RED Basics	7
1.3 Add your LABID, API key and LAT/LON	8
1.4 Set Up Your Sensor	9
1.5 Setup Your PWS Dashboard	11
1.5.1 Overview.....	11
1.5.2 Accessing the Dashboard	12
1.6 Displaying All PWS on a Map	21

Objective

In this Lab will use local weather information from your location as your Personal Weather Station. Since we do have an in-person lab on this occasion it requires some prior setup.

1. Lab Setup Instructions



Prerequisite (details follow)

1. Sign up for WatsonStudio account [Register for WatsonStudio](#) You will need this in the second part of the Lab
2. Sign up for Weather Company API Key [TWC APIKey](#) We have API keys already set up in the lab but it would be ideal if you could get use your own before the Lab.
3. Determine the Geolocation for your city using this link: [Get your Location](#) Note the Latitude and Longitude. We will use this to personalize your Weather Station Location.

Note:

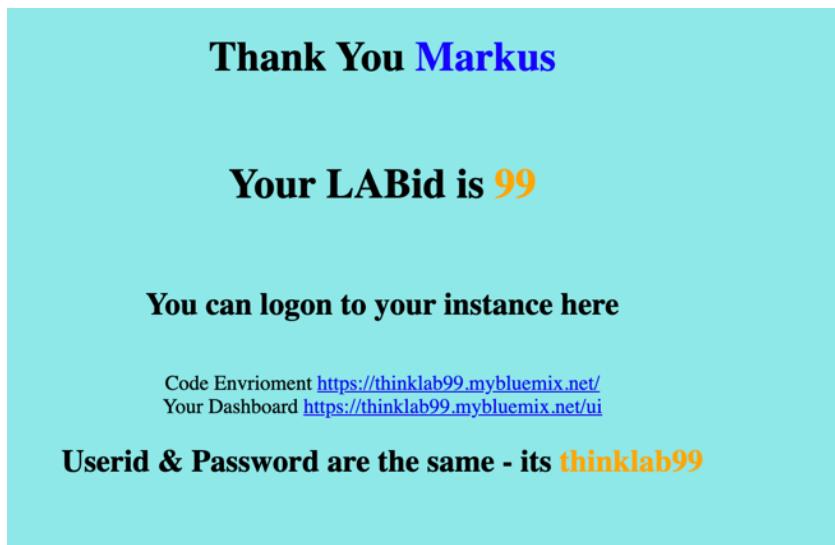
Since we are using VM box all our locations would be the same based on the DataCenter location. So please get your location before the lab using the link above in order to personalize your experience and results.

1.1 LAB Part 1

Before the LAB start we will go through some introductions and a walkthrough of the LAB and the exercises. You will setup your Node-RED instance with your API key and the Lat/Lon as well as your LAB ID.

Request your Node-RED environment with assign Username/Password and LABID – note information down

<https://thinklab1239.mybluemix.net/claimid>



Once you have your LAB instance login to access the coding environment.

Get Your Latitude and Longitude here <https://www.latlong.net/>

Latitude and Longitude Finder

Latitude and Longitude are the units that represent the *coordinates* use the name of a place, city, state, or address, or click the location c

Place Name

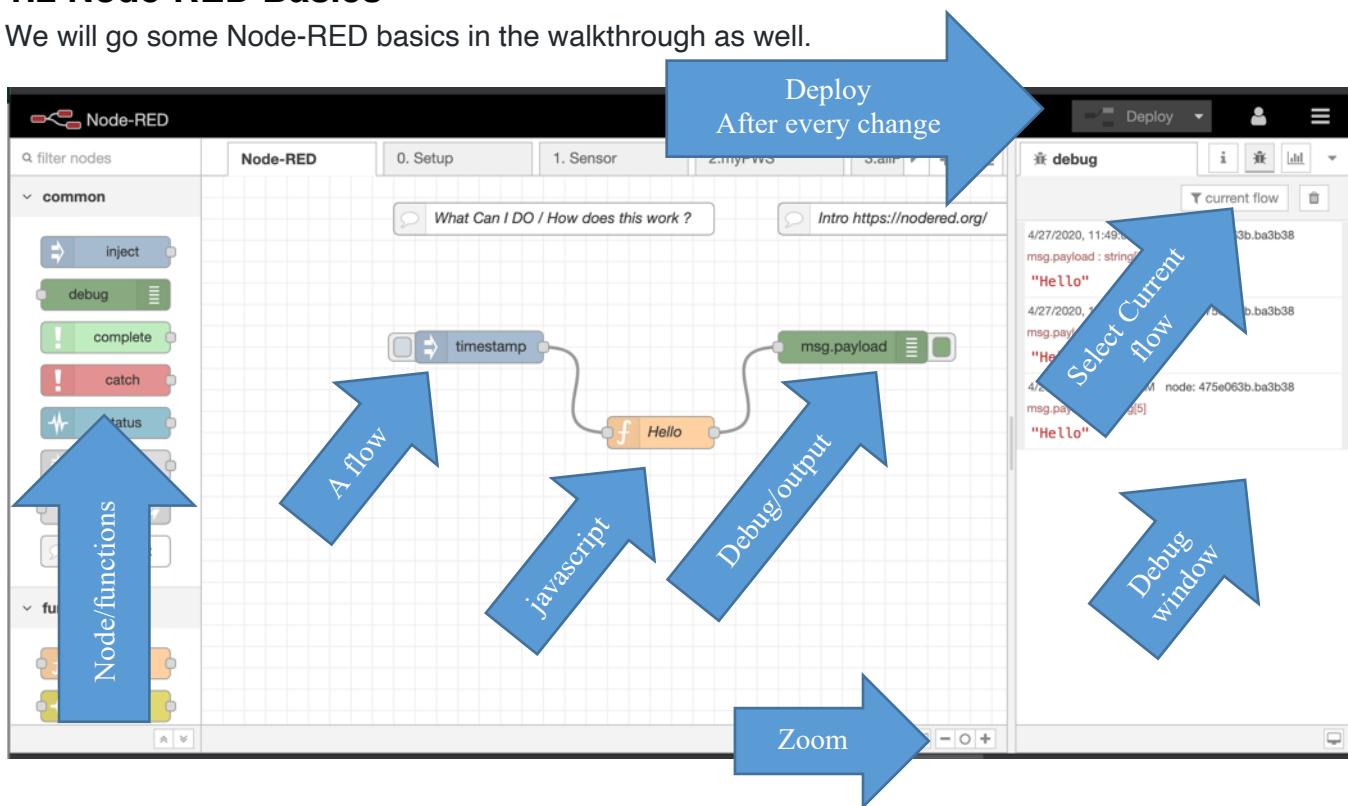
Find

Add the country code for better results. Ex: London, UK

Latitude	Longitude
<input type="text" value="43.653225"/>	<input type="text" value="-79.383186"/>

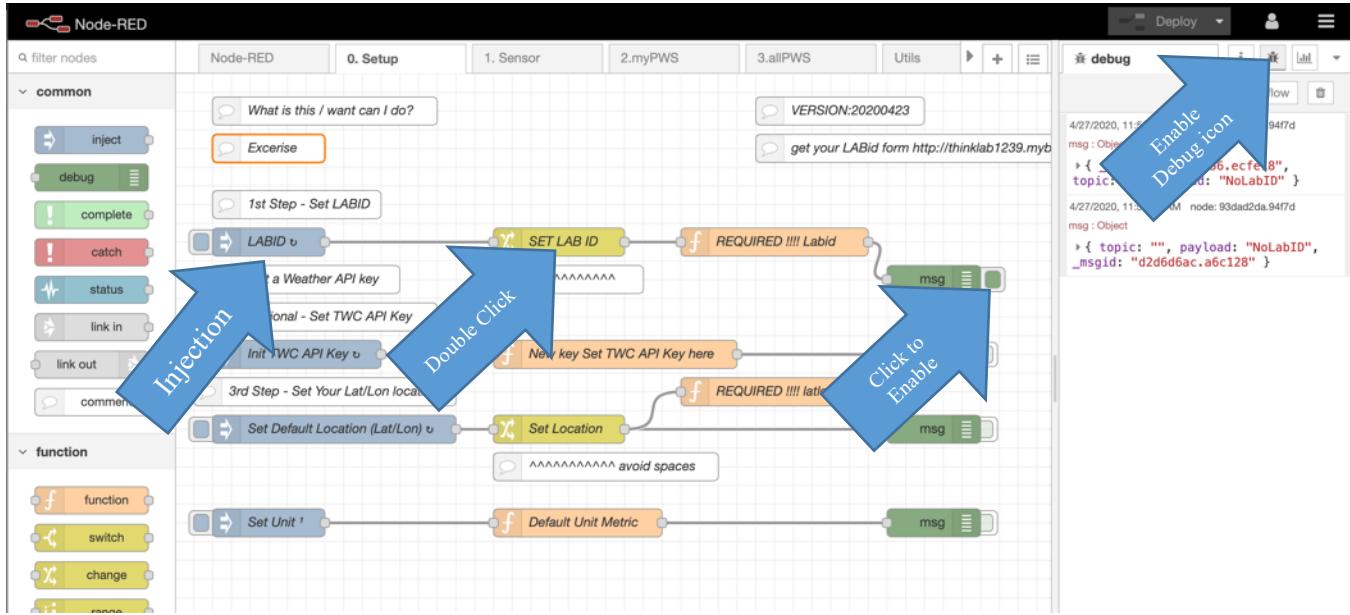
1.2 Node-RED Basics

We will go some Node-RED basics in the walkthrough as well.



1.3 Add your LABID, API key and LAT/LON

Click on the Setup Tab. There you will see some instruction and Node/Icons and flows.



Important: Make sure to deploy your changes and execute the nodes to set your information.

Exercise:

Double click the yellow icons and overwrite the values

The screenshot shows the Node-RED interface with the 'Setup' tab selected. A 'change' node is open in the edit dialog:

Edit change node

Properties

- Name: SET LAB ID

Rules

- Set global.labid to "NoLabID"

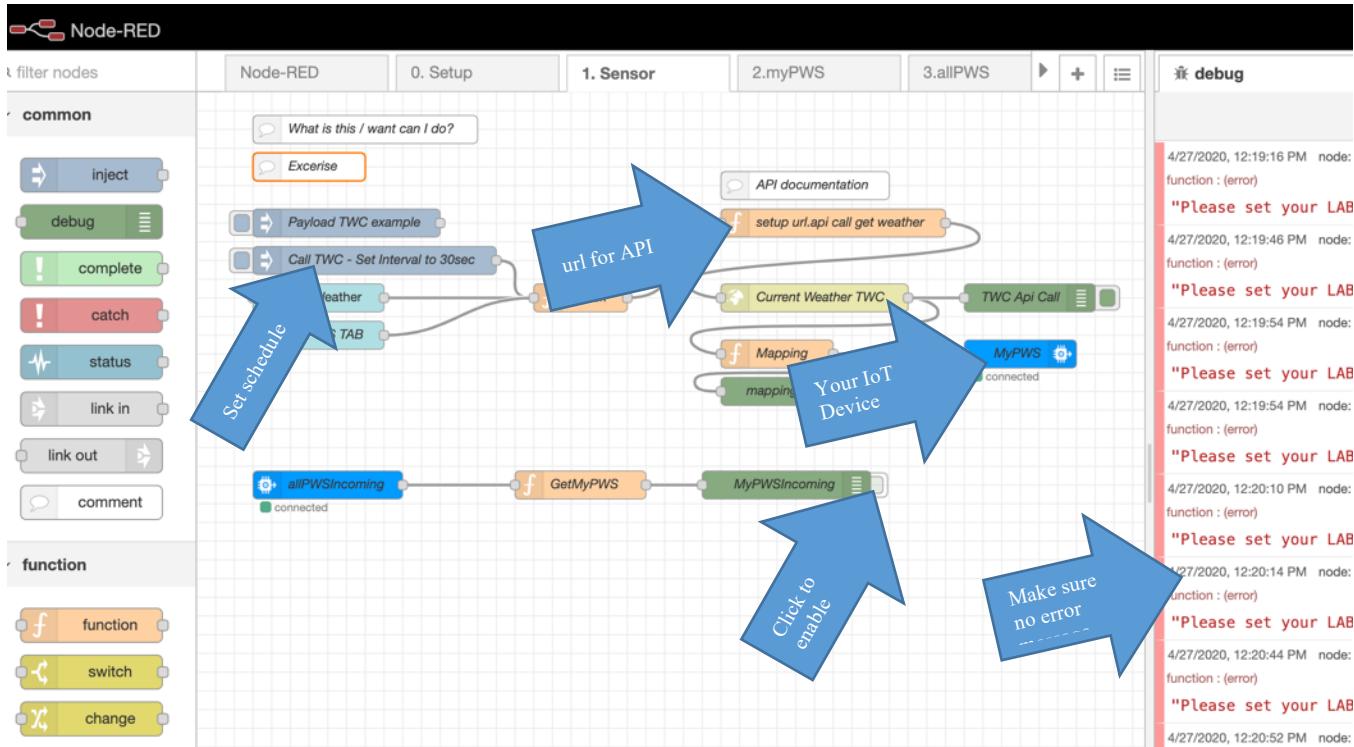
Annotations with blue arrows:

- "Double Click" points to the yellow 'SET LAB ID' node in the main canvas.
- "Overwrite With your ID" points to the 'to' field in the 'Rules' section of the dialog, which currently contains "global.labid" and "NoLabID".

Repeat the same steps for the Latitude and Longitude

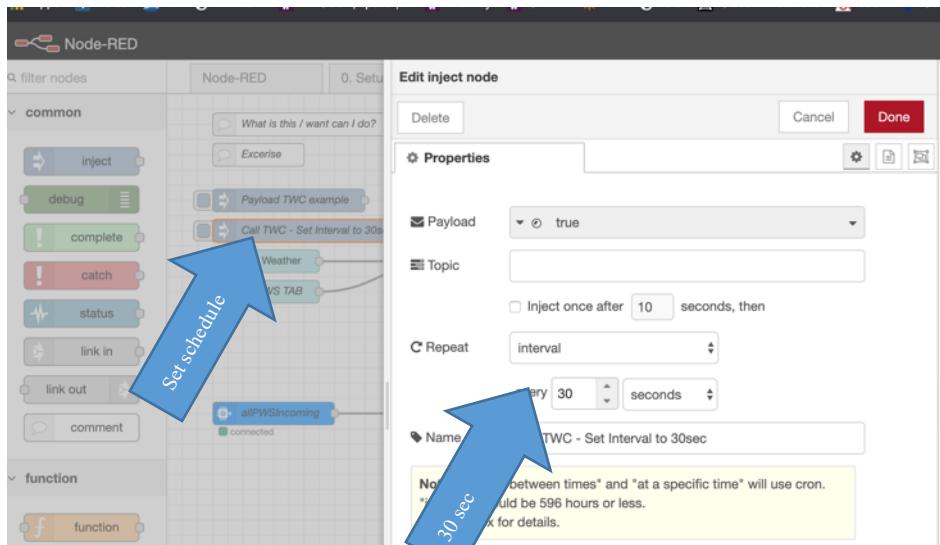
1.4 Set Up Your Sensor

Click On the “1. Sensor” Tab. In this version of the lab will simulate a Raspberry PI that has sensors by using the weather company API to get the information for our location. Once we have the information, we will fish some data out and send it to our IoT device. The IoT device are already set up. The device ID of your IoT Device is pws+LABID ...like pws44.



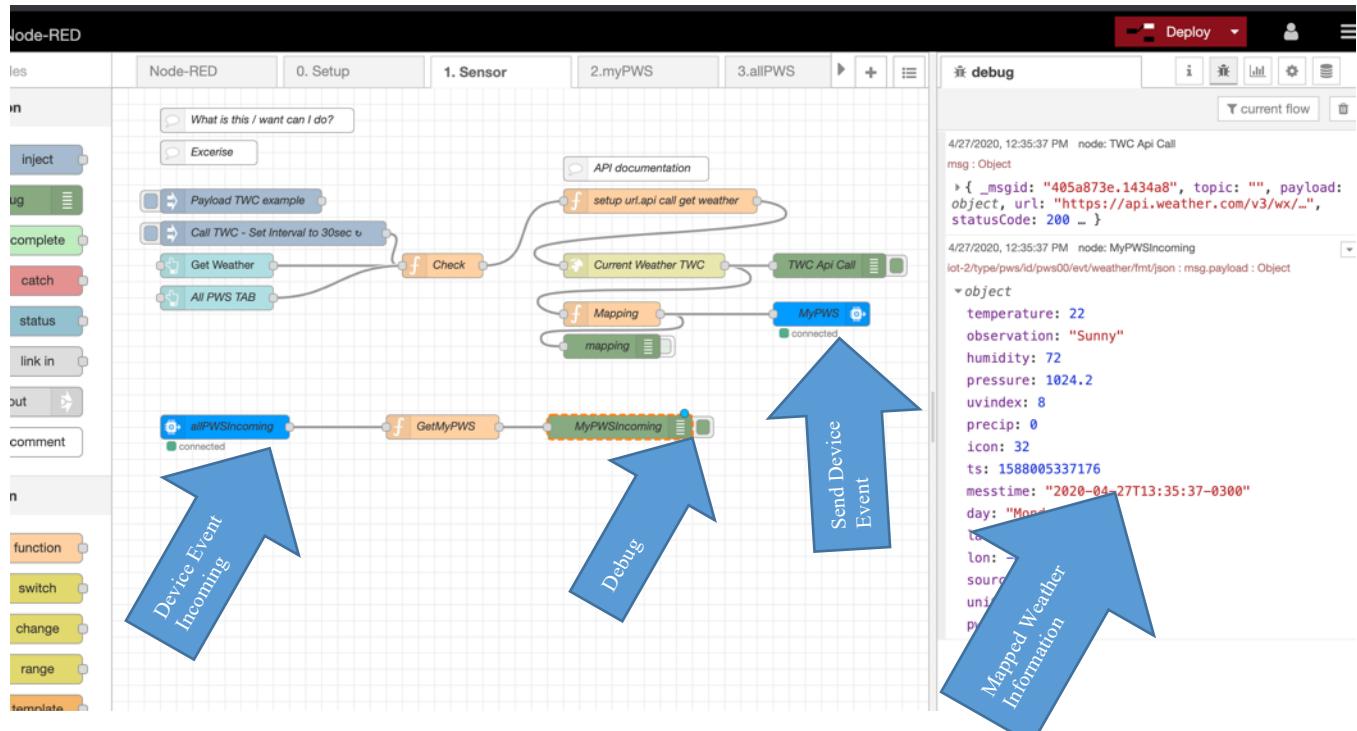
Exercise:

Set 30sec schedule for the API call to the weather company. Double click the Call TWC icon.



Note: Do not forget to deploy your changes using the red deploy button.

Verify that your PWS device is sending and receiving information by enabling the debug node at the bottom.



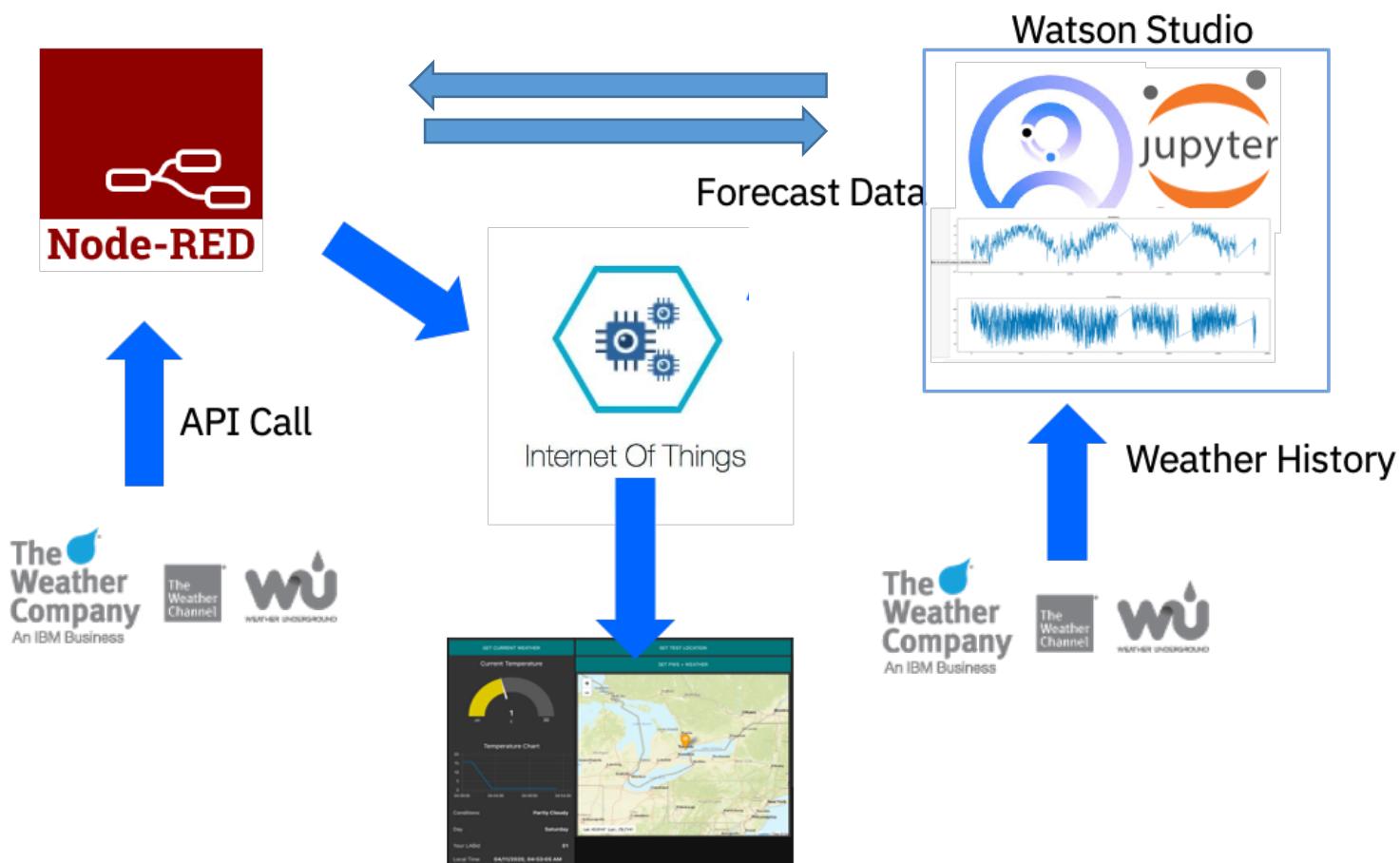
Above the example of the successful result.

1.5 Setup Your PWS Dashboard

Since we have data now weather information on our PWS IoT device, we can visualize the data using a dashboard within Node-RED.

1.5.1 Overview

Here is the High-level Architecture of our current setup.

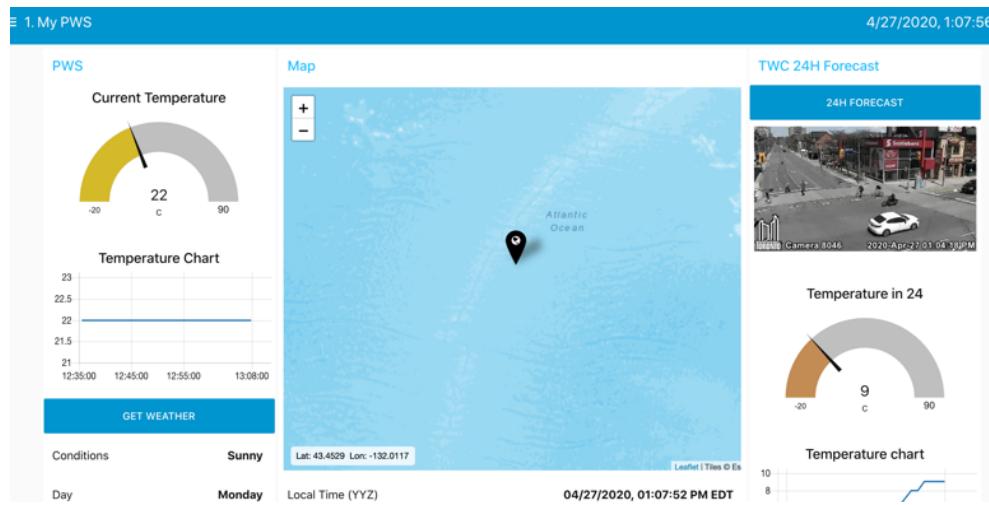


1.5.2 Accessing the Dashboard

To access the Dashboard, open a new browser tab or window and use the dashboard node-red URL, which is something like <https://thinklab<LABID>.mybluemix.net/ui>

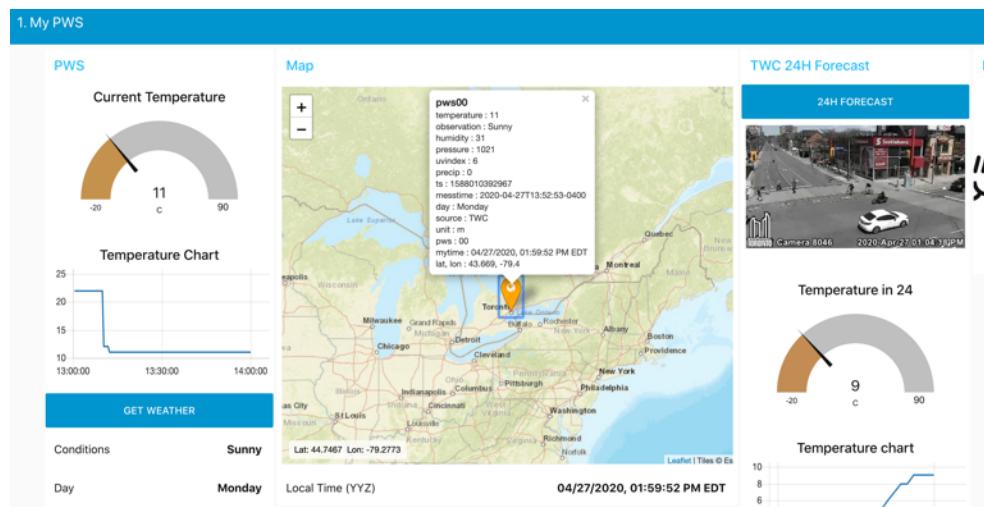
Note: it's the same as your coding environment URL but with the addition of “/ui”

You should see a screen like the one below. Click the “Get Weather” button to trigger the weather API call manually. You should get the temperature and a marker.



Important: Make sure the Marker Icon on the map points to your location and is NOT black.

If the icon is black, this means your LABID is not setup in Node-RED and if the marker does not point to your location and you need to adjust the Latitude and Longitude in Node-RED -- see steps above. It should look like something like the picture below if everything is set up correctly:



Clicking on the marker in the map should show your weather information.

Note: For the exercises it is optimal if you have two browser windows/tabs:
One with a node-red coding UI and one with the dashboard, so you can flip back and forth.

EXERCISES

1. Exercise: Add some data to our dashboard – Air Pressure.

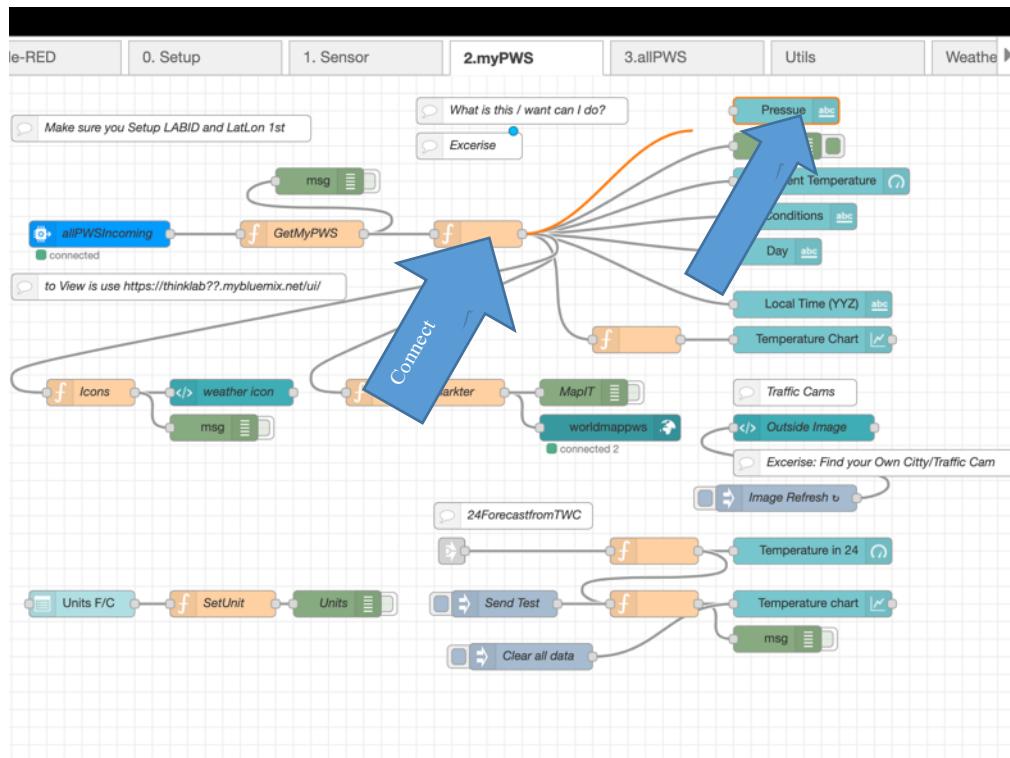
Since pressure is already part of your PWS payload

Example Payload

```
{"temperature":12,"observation":"Fair","humidity":30,"pressure":1020.2,"uvindex":6,"precip":0,"icon":34,"ts":1588011323116,"messtime":"2020-04-27T14:12:59-0400","day":"Monday","lat":43.669,"lon":-79.4,"source":"TWC","unit":"m","pws":"00","mytime":"04/27/2020, 02:15 :23 PM EDT"}
```

We just need to add a UI Text element and wire it to the data source.

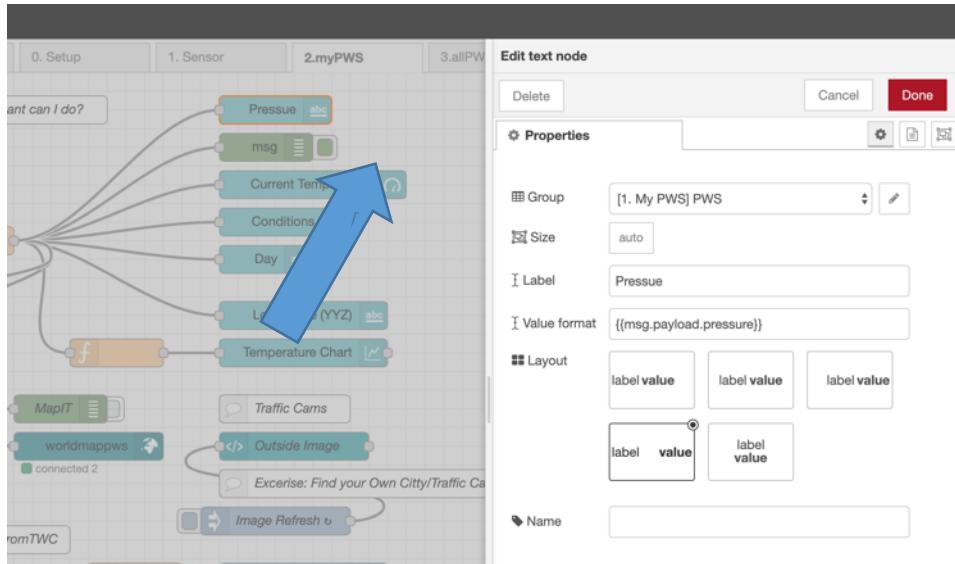
Go back to the Node-RED coding UI and select the myPWS Tab



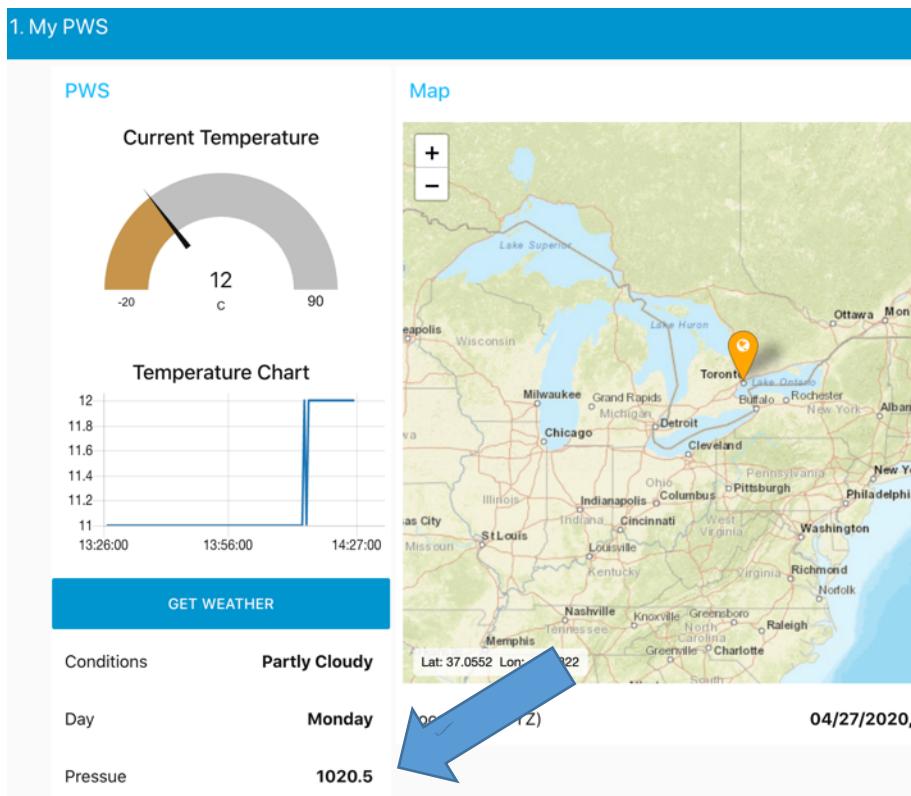
Connect the data and the UI elements.

Info: The Blue Node is the IoT Platform connection. It will get all events from the device types PWS. The GetMyPWS will filter out just your PWS based on the LABID.

Double click the pressure UI text node and adjust the values with msg.payload.pressure



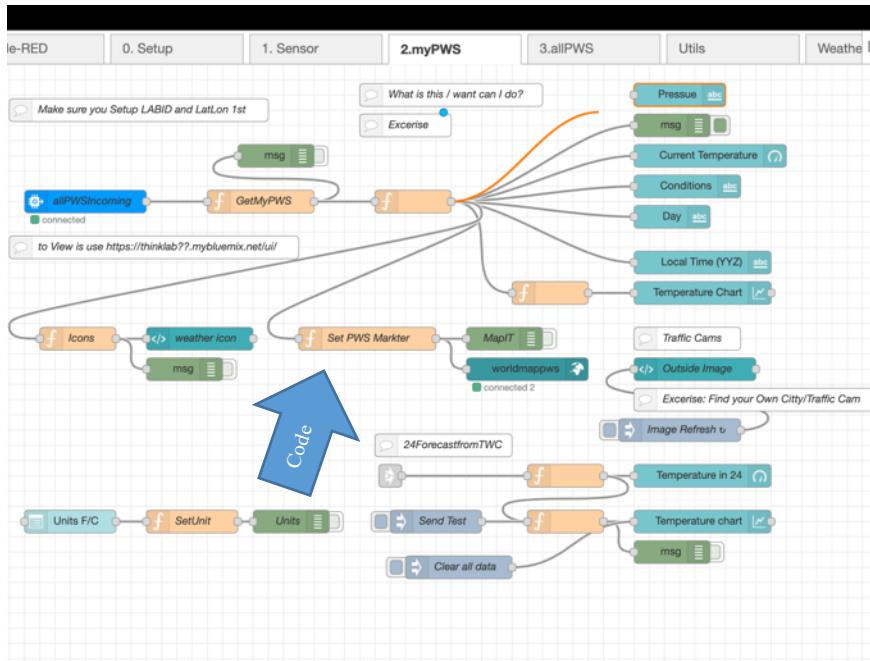
Click “Done” and Deploy your changes. You should see that the pressure value. On the dashboard.



Feel free to play with the dashboard and UI Elements. For example, use a Gauge instead of text.

2. Exercise – Change your marker Icon to an emoji

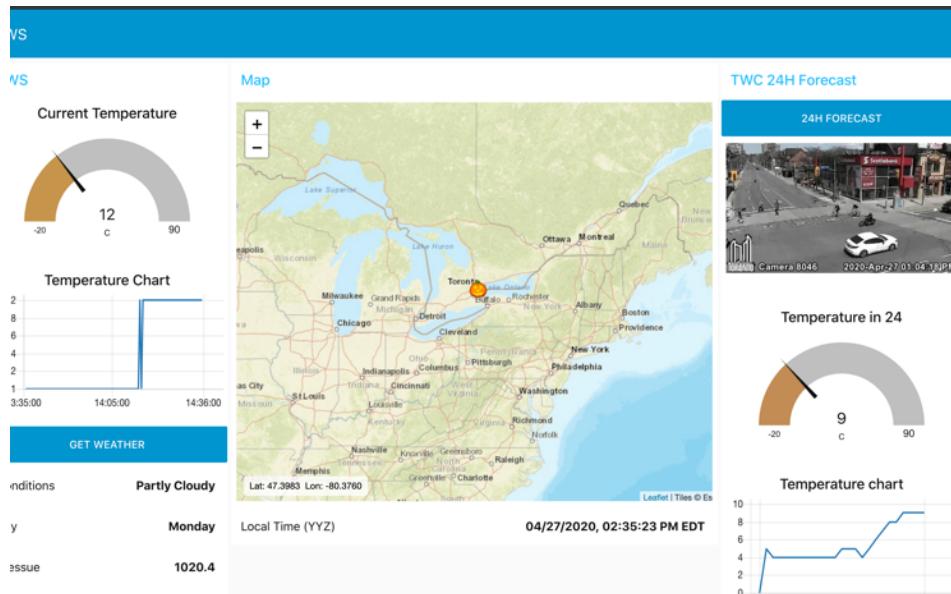
<https://github.com/dceejay/RedMap/blob/master/emojilist.md>



Check out the code on the JS node “Set PWS Marker” you can find the list of icons here

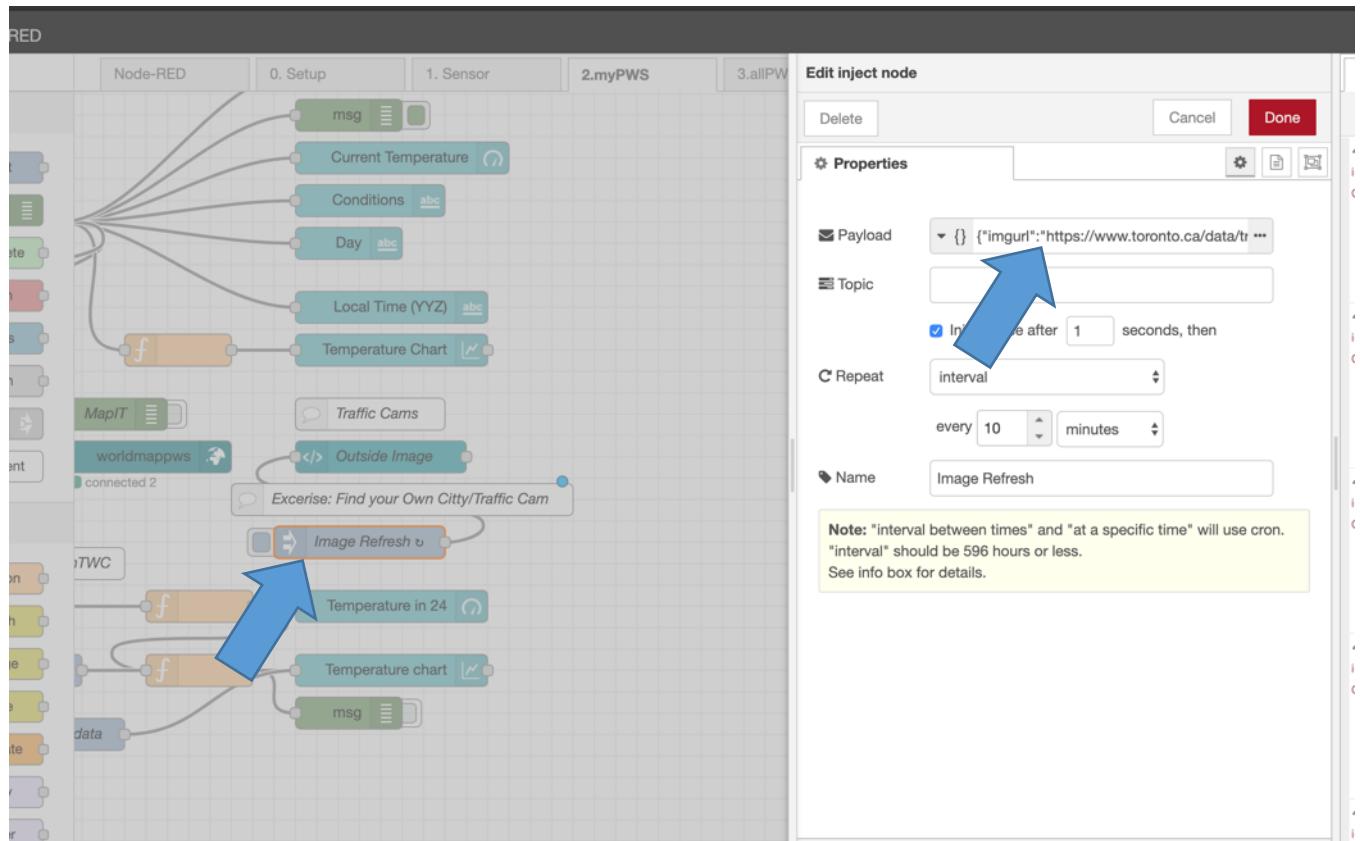
<https://github.com/dceejay/RedMap/blob/master/emojilist.md>

Change the line `msg.payload.icon = ":jack_o_lantern:";` with the emoji of your choice like `:jack_o_lantern:";`



3. Exercise – Change the Traffic cam Image

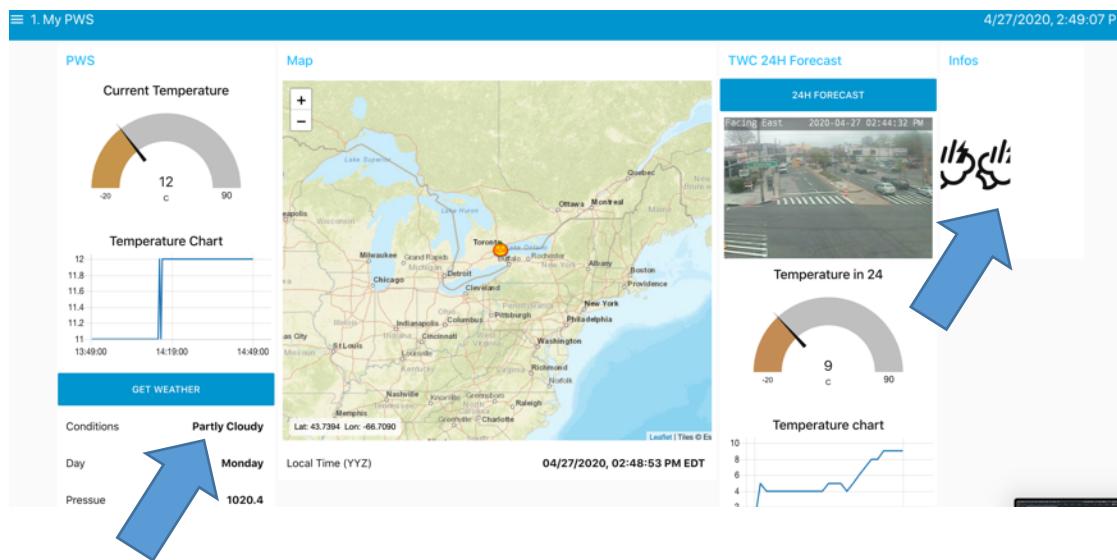
Change the feed from the traffic cam. If you do not have a local one, NYC traffic cams can be found here. <https://511ny.org/map#Camera>



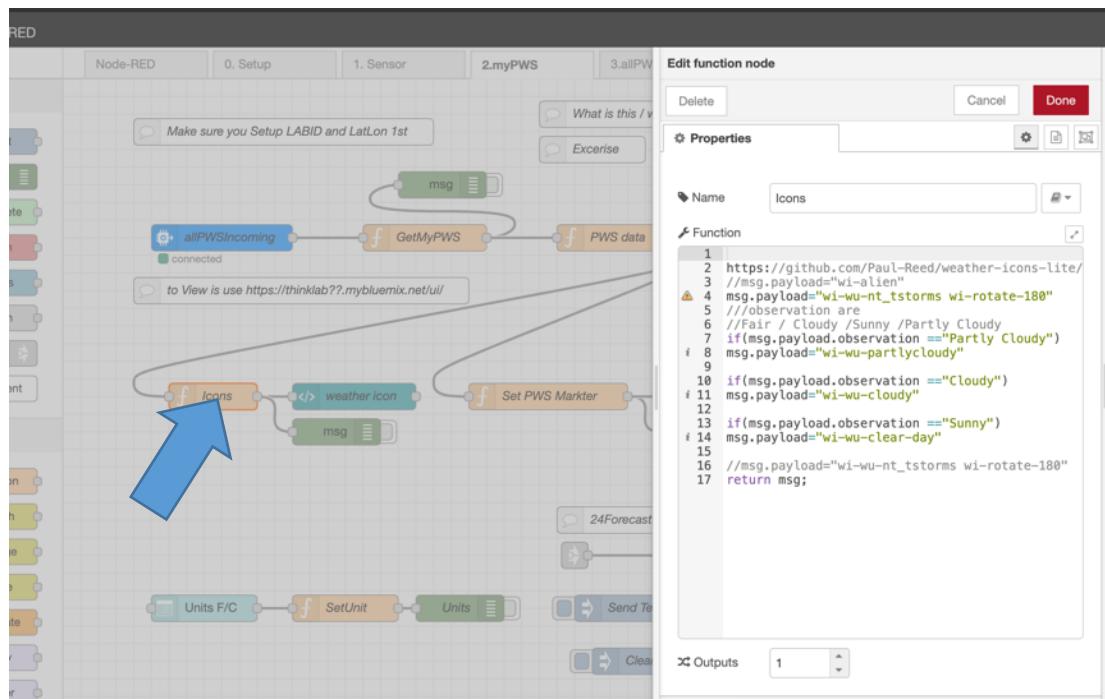
Change it with a different URL, like <https://511ny.org/map/Cctv/4616648--17>

```
{"imgurl1": "https://www.toronto.ca/data/transportation/roadrestrictions/CameraImages/loc8046.jpg", "imgurl": "https://511ny.org/map/Cctv/4616652--17"}
```

4. Exercise: Adjust the weather icons with the matching conditions



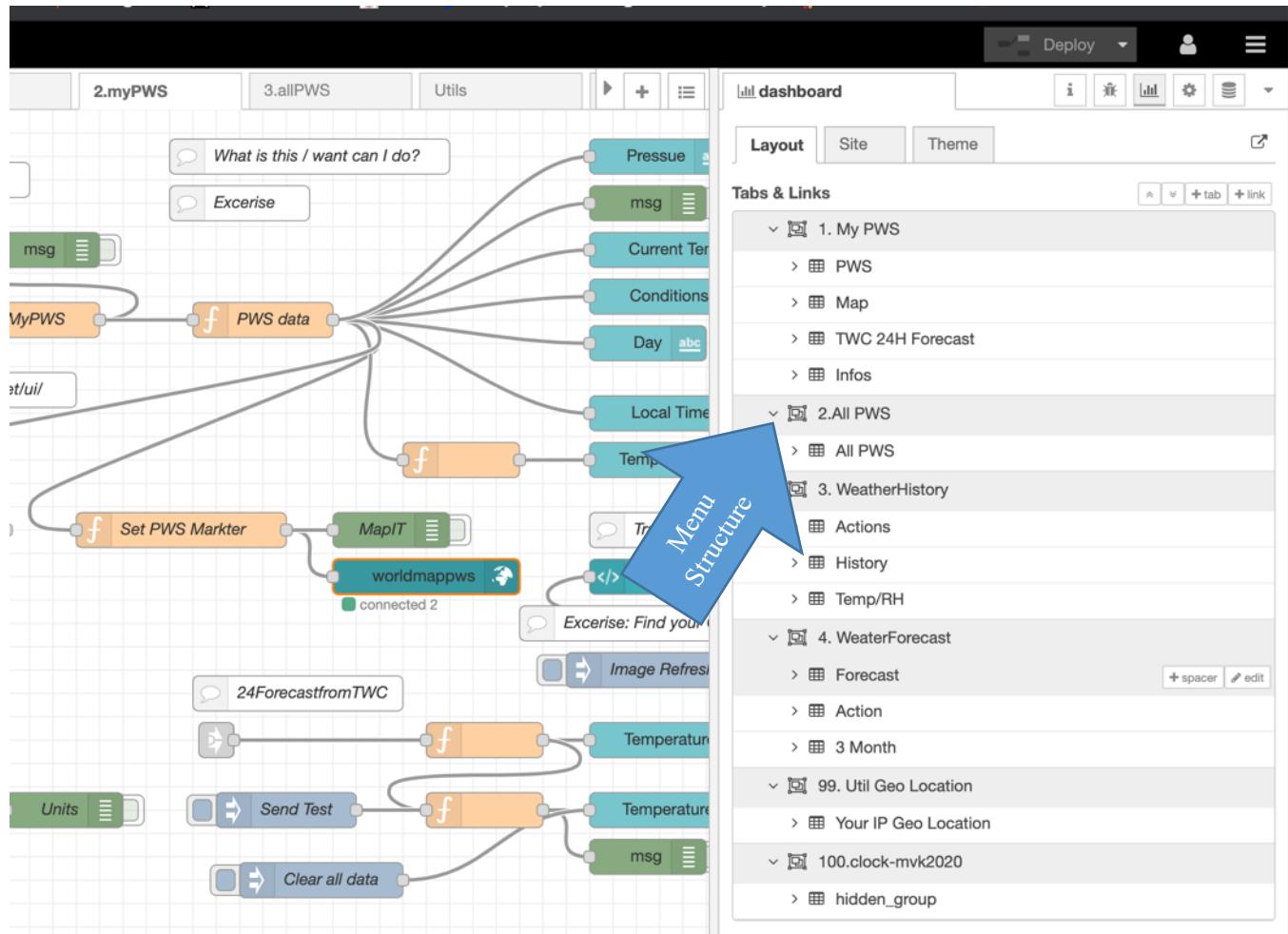
Code is in the node “Icons”:



The Icon mapping can be found here:

https://github.com/Paul-Reed/weather-icons-lite/blob/master/css_mappings.md

Feel free to adjust the layout via the Dashboard Menu on the right via drop and drag



5. Add more information like the SunRise and Wind direction to your mapping on the “Sensor” tab and add the information to your myPWS dashboard

The screenshot shows the Node-RED interface. On the left, a flow starts with a 'Sensor' node, followed by a 'Check' function node, and ends with a 'GetMyPWS' function node. A blue arrow labeled 'Mapping' points to the function node. The 'Edit function node' dialog for the 'Mapping' node is open, showing the code:

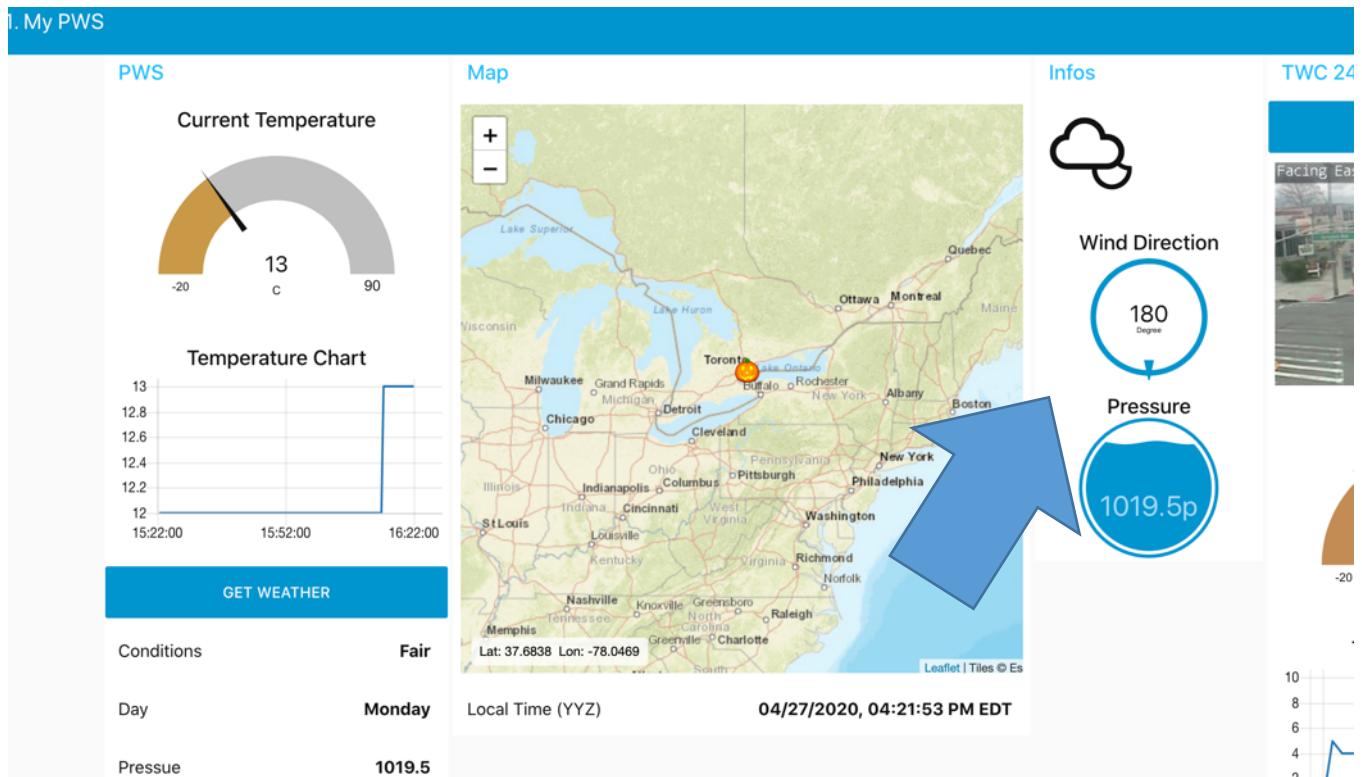
```

1 msg.payload.old=msg.payload
2 msg.payload={}
3 msg.payload.temperature=msg.payload.old.temperature
4 msg.payload.observation=msg.payload.old.wxPhraseL
5 msg.payload.humidity=msg.payload.old.relativeH
6 msg.payload.pressure=msg.payload.old.pressureM
7 msg.payload.uvindex=msg.payload.old.uvIndex
8 msg.payload.precip=msg.payload.old.precip1H
9 msg.payload.icon=msg.payload.old.iconCode
10 msg.payload.date=Date.now()
11 msg.payload.time=msg.payload.old.validTime
12 msg.payload.dayOfWeek=msg.payload.old.dayOfWeek
13 msg.latitude=pparseFloat(global.get("lat"))
14 msg.longitude=pparseFloat(global.get("lon"))
15 msg.unit=global.get("unit")
16 msg.pws=msg.payload.pws
17 msg.iconCode=global.get("labin")
18 msg.icon=global.get("icon")
19 msg.validTime=msg.payload.validTime
20 msg.validTimeLocal=msg.payload.validTimeLocal
21 msg.cloudCeiling=5000,
22 msg.cloudCoverPhrase="Mostly Cloudy",
23 msg.dayOfWeek="Friday",
24 msg.dayOrNight="D",
25 msg.expirationTimeUtc=1586545455,
26 msg.validTimeUtc=msg.payload.validTimeUtc
    
```

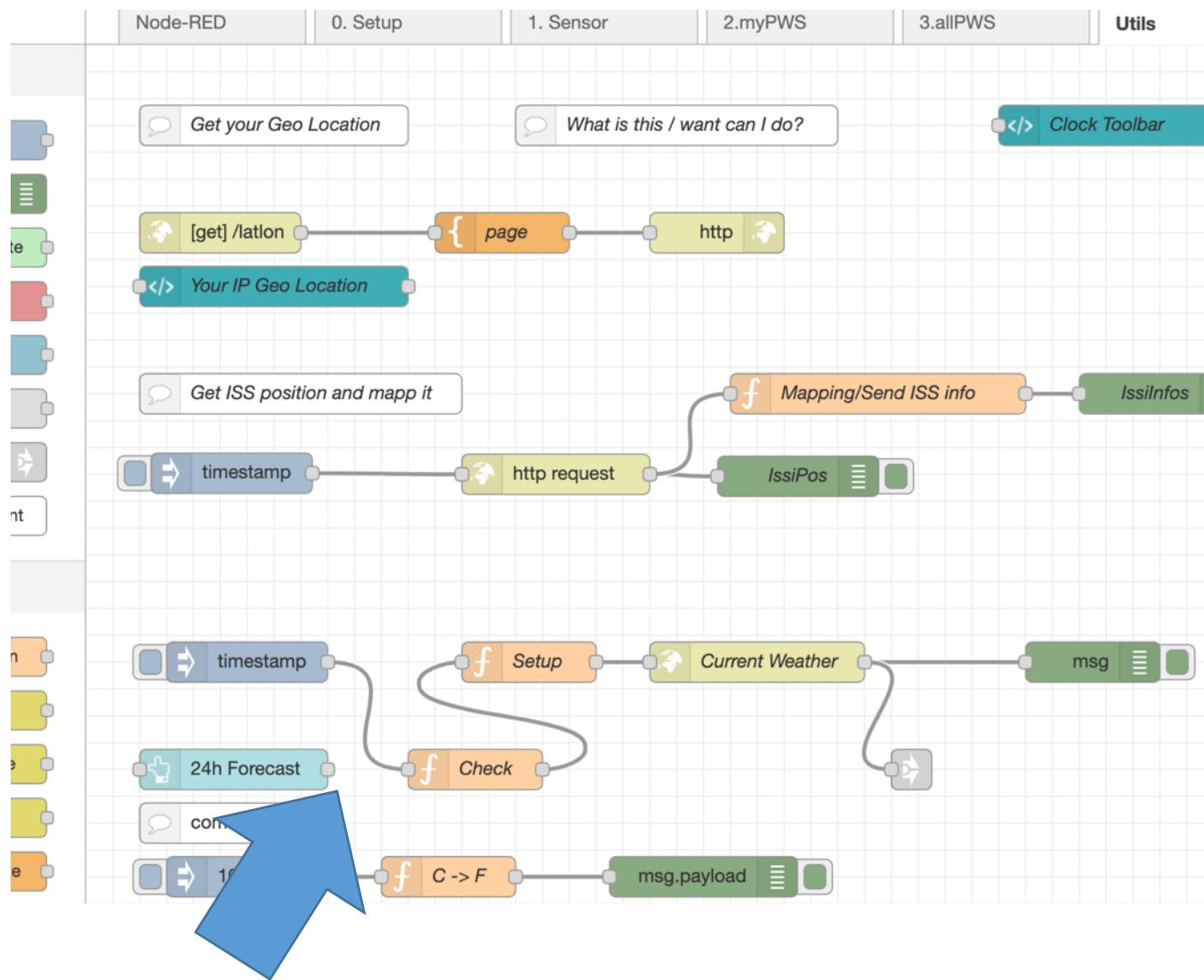
The 'debug' tab on the right shows the API payload data:

```

pressureMeanSeaLevel: 1019.9
pressureTendencyCode: 2
pressureTendencyTrend: "Falling"
relativeHumidity: 30
snow1Hour: 0
snow6Hour: 0
snow24Hour: 0
sunriseTimeLocal: "2020-04-27T06:14"
sunriseTimeUtc: 1587982486
sunsetTimeLocal: "2020-04-27T20:15:5"
sunsetTimeUtc: 1588032953
...
    
```



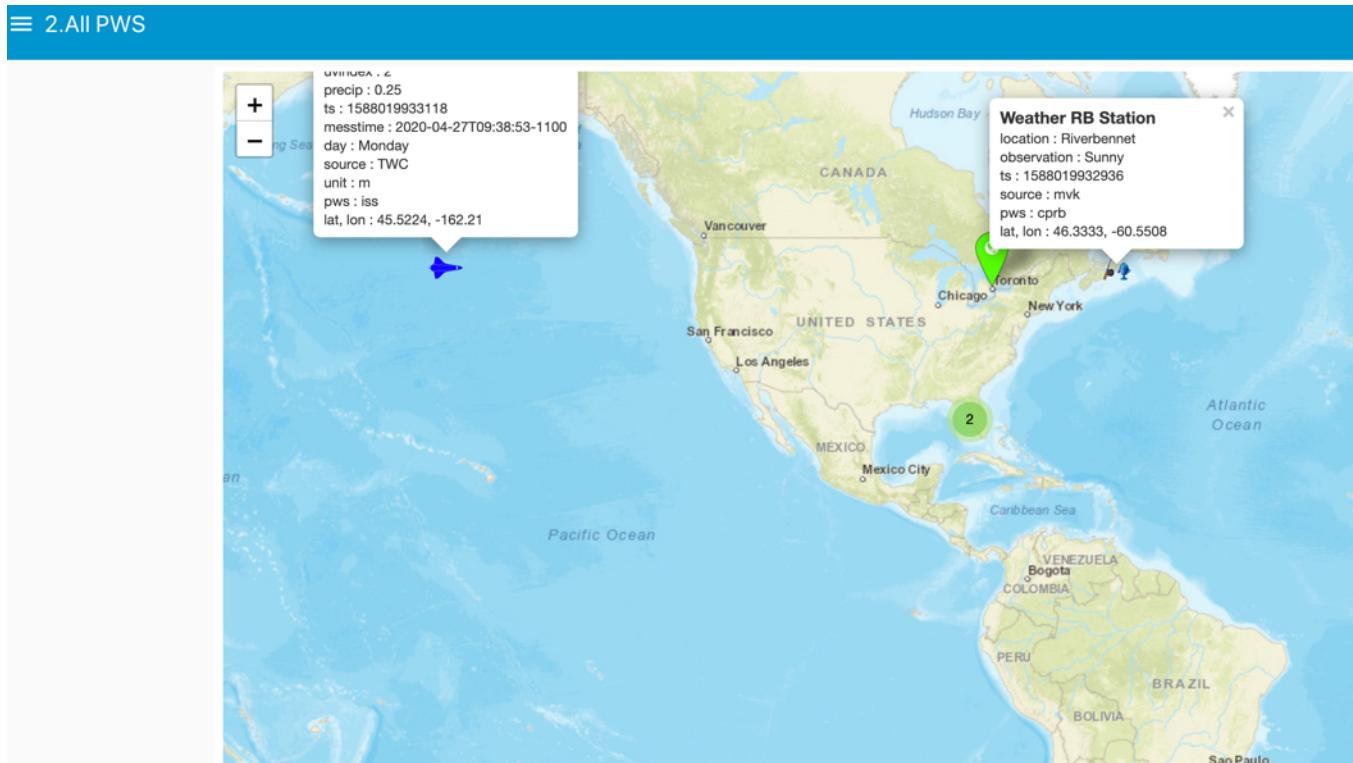
6. Connect the 24 Forecast button on the Util pages



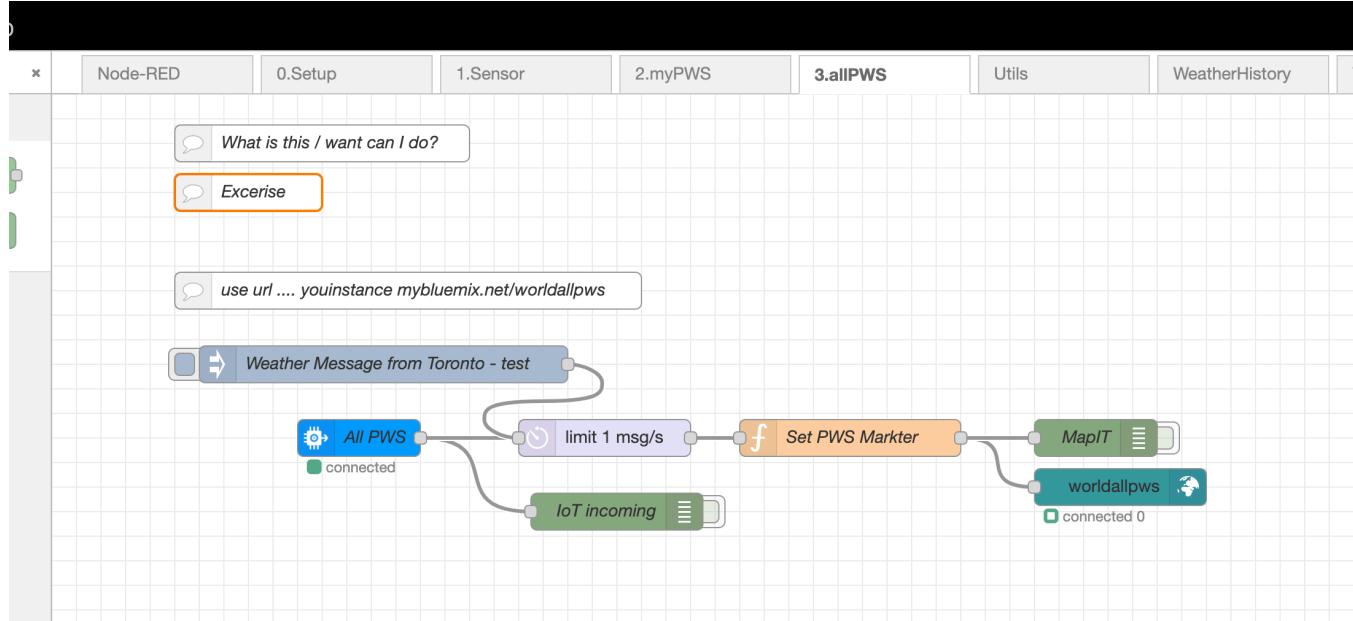
1.6 Displaying All PWS on a Map.

Go to the All PWS dashboard. Click “Display my PWS.” Your PWS with all the information should be displayed as well as everybody else’s. (Please note that this map just uses the markers, not your emoji from the previous change on MyPWS Code Tab. See the code on the allPWS Tab for details.)

You should also see the ISS/Shuttle and a couple other weather stations. Click on the icons to get more information. Feel free to explore and change the PWS dashboard.



Explore the Code on Tab “all PWS”.



Try to change the Icon of your Weather Station / Device or add more / less information.

Troubleshooting

If you run into issues download the flow for github and import it into Node-RED/
<https://github.com/markusvankempen/ThinkLab1239/tree/master/node-red>

SDK info

You can use the python or node.js sdk code to connect to your PWS from your desktop if node.js or python is installed (Note: On the VM we only have the browser)

Examples:

<https://github.com/markusvankempen/ThinkLab1239/tree/master/extras>

More infos

<https://github.com/ibm-watson-iot/iot-python>

In Part Two of the LAB we will work with Watson Studio so please make sure to sign up for a free account. [Register for WatsonStudio](#)

<https://dataplatform.cloud.ibm.com/registration/stepone>

Get the instructions for LAB Part two here

<https://github.com/markusvankempen/ThinkLab1239/tree/master/instructions>

For more details got to the github

<https://github.com/markusvankempen/ThinkLab1239>

Cheers

Markus van Kempen

mvk@ca.ibm.com

Version:20200501