IBM Bluemix™

Hands-on Exercise

View Yr weather meteogram and information on the map using Bluemix and Node-RED.

IBM Bluemix
Wednesday, October 21, 2015

Contents

Introduction

This exercise will show you how to deploy an application from the IBM Bluemix Web User Interface. We will create a Node-RED application running on SDK for Node.js runtime. The application is a simple Google Maps application, where local weather information from the Norwegian weather provider Yr.no is included. The user can drag a current geoposition marker around the map to get Yr weather information for particular places (if available). The client will send requests to Node-RED endpoint running in the Bluemix application to lookup the Yr weather information. Custom Node-RED nodes will be used for this, and we will use the Cloud Foundry cf CLI tool to push updates made locally to the application to Bluemix.

About IBM Bluemix

Bluemix is an open-standard, cloud-based platform for building, managing, and running applications of all types (web, mobile, big data, new smart devices, and so on).

- The developer can choose any language runtime or bring their own. Zero to production in one command.
- A catalog of IBM, third party, and open source API services allow the developer to stitch an application together in minutes.
- Development, monitoring, deployment, and logging tools allow the developer to run the entire application.
- Sign up in minutes. Pay as you go and subscription models offer choice and flexibility.

About Node-RED

Node-RED is a visual tool for wiring the internet of things – connecting hardware devices, APIs and online services in a new and interesting way. Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range nodes in the palette. Flows can be then deployed to the runtime in a single-click.

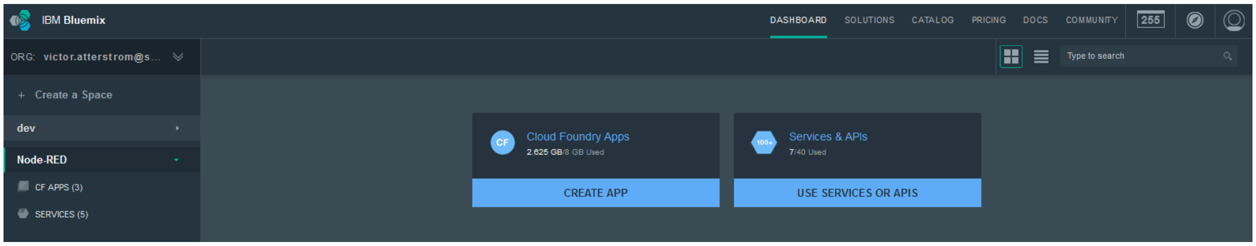
- JavaScript functions can be created within the editor using a rich text editor.
- A built-in library allows you to save useful functions, templates or flows for re-use.
- See https://nodered.org for more information.

Prerequisites

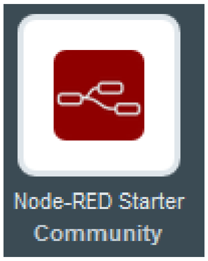
- Register on IBM Bluemix at https://bluemix.net

Step 1. Create Your Node-RED Application

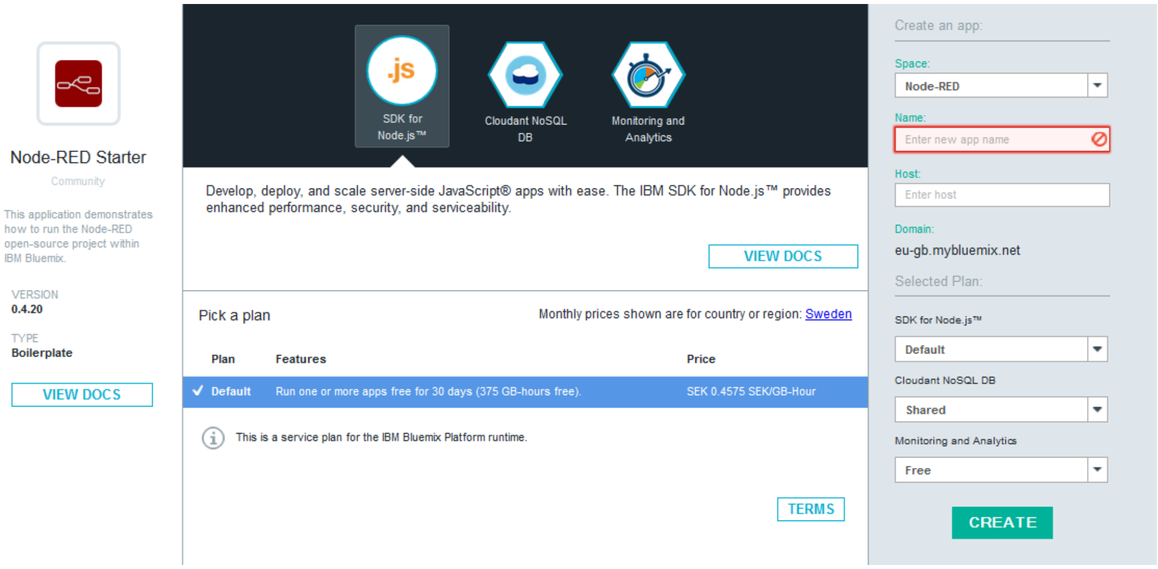
1. In a browser navigate to https://console.eu-gb.bluemix.net (or https://console.ng.bluemix.net if stated by the instructor).
2. Select ‘LOG IN’ then enter your log in information and press ‘SIGN IN’. You should be seeing your dashboard view:



3. Select the ‘CATALOG’ view.
4. Locate the **NODE-RED Starter** in the boilerplate section of the catalog and click on it. Make sure your ‘REGION’ is set to ‘United Kingdom’ (or ‘US South’ if stated by the instructor) in the upper right corner (click on icon in the upper right corner).



5. Enter a name for your application, as shown below (host will automatically be completed). The host name must be unique on Bluemix, so please choose a name with your company name or initials to try to make a unique name. This will be referred to as **<your application directory>** later. Press **‘CREATE’.**



Node-RED Starter

Community

This application demonstrates how to run the Node-RED open-source project within IBM Bluemix.

VERSION
0.4.20

TYPE
Boilerplate

Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides enhanced performance, security, and serviceability.

Pick a plan

Monthly prices shown are for country or region: Sweden

	Plan	Features	Price
✓	Default	Run one or more apps free for 30 days (375 GB-hours free).	SEK 0.4575 SEK/GB-Hour

This is a service plan for the IBM Bluemix Platform runtime.

Create an app:

Space: Node-RED

Name: Enter new app name

Host: Enter host

Domain: eu-gb.mybluemix.net

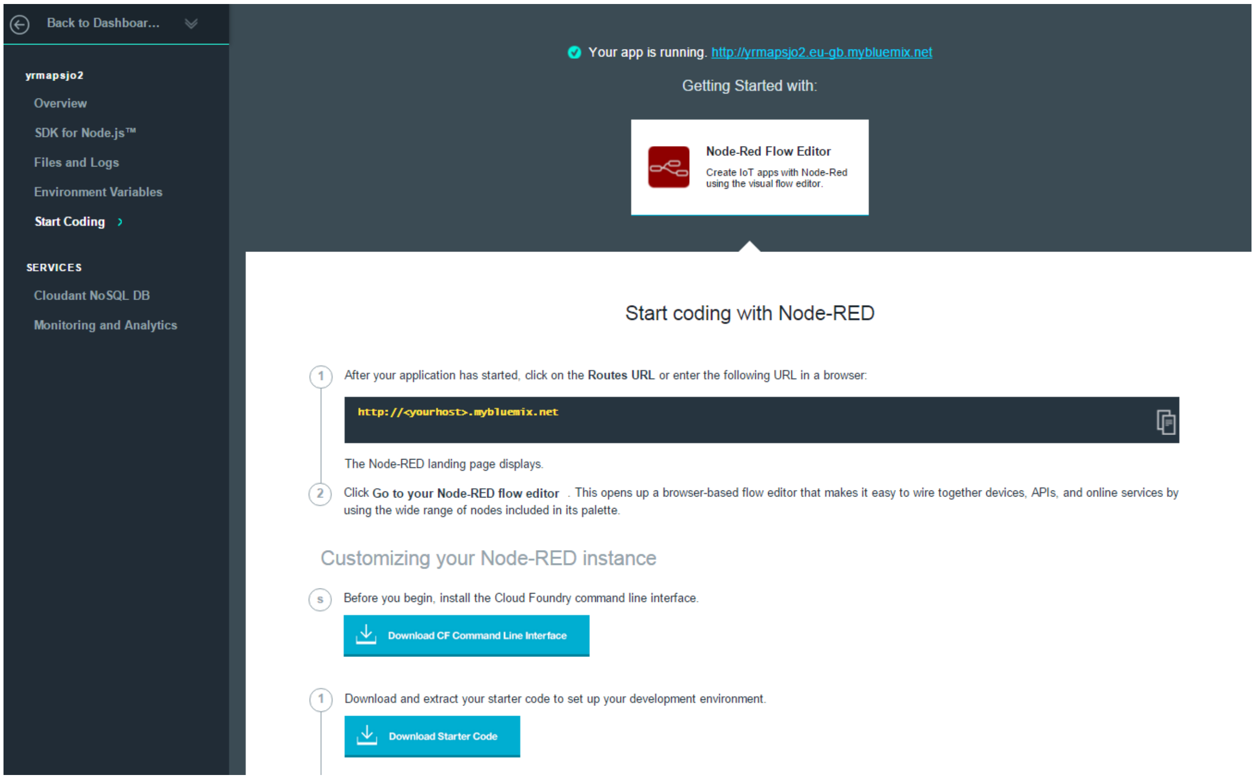
Selected Plan:

SDK for Node.js™: Default

Cloudant NoSQL DB: Shared

Monitoring and Analytics: Free

6. Your application is now staging and will be up and running in a short while. Press **'Start Coding'** to see information about getting started. This page contains information on using the Cloud Foundry cf CLI (Command Line Interface) to connect to your Bluemix environment and push updates.



Your app is running. http://yrmapsjo2.eu-gb.mybluemix.net

Getting Started with:

Node-Red Flow Editor

Create IoT apps with Node-Red using the visual flow editor.

Start coding with Node-RED

1 After your application has started, click on the **Routes URL** or enter the following URL in a browser:

```
http://<yourhost>.mybluemix.net
```

The Node-RED landing page displays.

2 Click **Go to your Node-RED flow editor** . This opens up a browser-based flow editor that makes it easy to wire together devices, APIs, and online services by using the wide range of nodes included in its palette.

Customizing your Node-RED instance

s Before you begin, install the Cloud Foundry command line interface.

1 Download and extract your starter code to set up your development environment.

7. If you have not already installed the cf CLI, press the **'Download CF Command Line Interface'**. Select the installer for your local environment (e.g. Windows 64-bit) and follow the instructions to install cf.

8. Press **'Download Starter Code'** to download the starter application to a new directory of your choice on your local machine (referred to later as **<your application directory>**). The download file will have the name **<your application name>**.zip. Extract the file in **<your application directory>.**

9. Follow the instructions on the 'Start Coding' page to connect with cf to your Bluemix environment. E.g. like this:

   a. Open a CLI interface and enter **cd <your application directory>**

   b. cf api api.eu-gb.bluemix.net

   c. cf login

   i. Email: your IBM ID email address

   ii. Password: your IBM ID password

   iii. Org: your IBM ID email address

10. We are going to use two Node-RED nodes to get the Yr weather information to display. The first node, lookup-place is in the node-red-contrib-geonames npm package (https://www.npmjs.com/package/node-red-contrib-geonames). The lookup-place node uses Geonames.org API to lookup a place name from geolocation (latitude and longitude). The second node, weather-info, is in the node-red-contrib-yr npm package (https://www.npmjs.com/package/node-red-contrib-yr). The weather-info node builds links to Yr weather information for the place returned by lookup-place.

    To get the npm packages installed (we do a cf push in step 11 to do this), edit the package.json file in **<your application directory>** to add these 2 lines at the end of the 'dependencies' property:

    a. "node-red-contrib-geonames":"0.x",

    b. "node-red-contrib-yr":"0.x"

    The file should then look similar to this. Please make sure you remember to add a comma at the end of the line above where you inserted the two new lines ☺

```
{
  "name"        : "node-red-bluemix",
  "version"     : "0.4.20",
  "dependencies": {
    "when": "~3.x",
    "mongodb": "~1.4.x",
    "nano": "~5.11.0",
    "cfenv":"~1.0.0",
    "feedparser":"~0.19.2",
    "redis":"~0.10.1",
    "node-red": "0.x",
    "node-red-bluemix-nodes":"0.x",
    "node-red-node-cf-cloudant":"0.x",
    "node-red-contrib-scx-ibmiotapp":"0.x",
    "node-red-contrib-ibmpush":"0.x",
    "node-red-contrib-bluemix-hdfs":"0.x",
    "node-red-nodes-cf-sqldb-dashdb":"0.x",
    "node-red-contrib-geonames":"0.x",
    "node-red-contrib-yr":"0.x"
  },
  "engines": {
    "node": "0.10.x"
  }
}
```

11. Push the update back to the application in Bluemix by entering this cf command:

    a. **cf push <your application name>**

    This might take a few minutes, the application will be stopped, rebuilt, restaged and started again. You can try to look for something like this in the cf output log:

    ```
    -----> Build succeeded!
           +-- node-red-contrib-geonames@0.1.10
           +-- node-red-contrib-yr@0.1.10
    ```

    The cf output log should end with something like this, to show that your application is running:

    ```
          state     since                    cpu    memory         disk          details
    #0    running   2015-10-19 02:32:37 PM   0.0%   219M of 512M   283.7M of 1G
    ```

12. **Optional:** Yo can always view the log of your application, you can open another command prompt window and use this cf command to tail the application log:

    a. **cf logs <your application name>**

13. **Optional:** If you would like to download your application, you can do this by installing a cf plugin. You find the cf download plugin on GitHub: https://github.com/ibmjstart/cf-download

    To install the plugin, issue these commands from the command prompt:

    a. **cf add-plugin-repo CF-Community http://plugins.cloudfoundry.org/**

    b. **cf install-plugin cf-download -r CF-Community**

    You now can download your application this way (it might be a good idea to do this in an empty directory first time to avoid interfering with existing files):

    a. **cf download <your application name>**

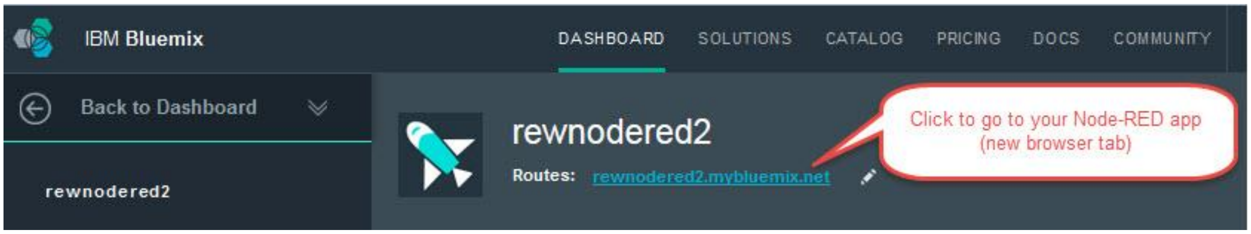
    This will create a new directory named **<your application name>**-download with the applications files.

    This blog post is a good read on cf download, and also mentions the '--omit' option to omit desired files or directories from the download: http://blog.ibmjstart.net/2015/05/22/cf-download/

    As discussed in the blog post, the 'cf download' will download everything, including e.g. dependent node.js modules. Specific directories can be omitted by using the 'omit' option. E.g.

    cf download <your application name> --verbose --omit "app/bin; app/node_modules; app/vendor; app/.app-management; logs; tmp"

14. When fully staged, press the **route** for your application, this launches the Node-RED main page.

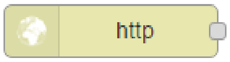


15. Now click **Go to your Node-RED flow editor** to open the flow editor.

16. When using Node-RED we build our apps using this graphical editor interface to wire together the blocks we need. We can simply drag and drop the blocks from the left menu into the workspace in the center of the screen and connect them to create a new flow.

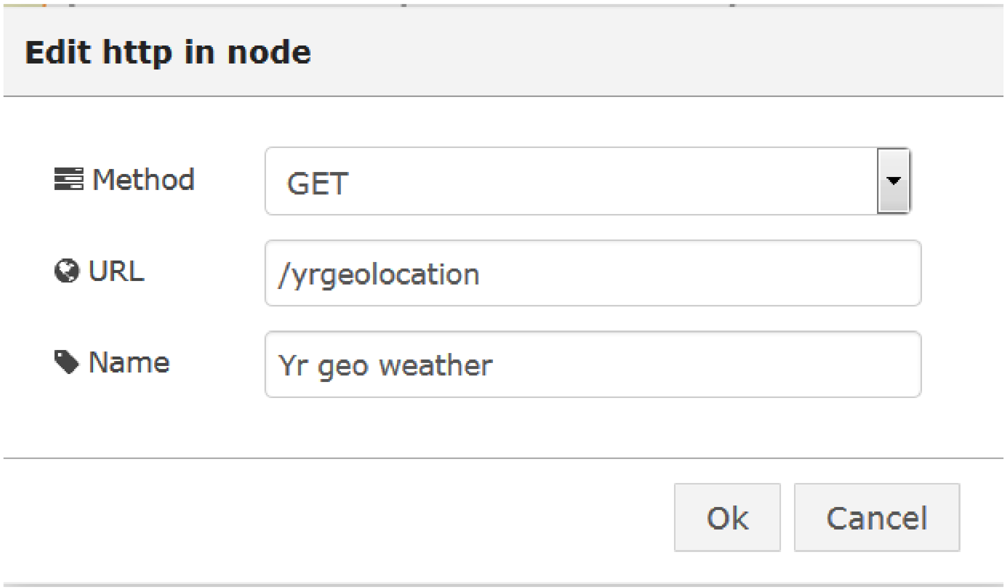
Step 2. Create your Node-RED flow.

1. Locate the http node under the input section in the palette window.



2. Drag the node onto the workspace in the middle of the screen.

3. Double click on your new http node and enter a value for the URL field, e.g. '/yrgeolocation', and optionally a name.



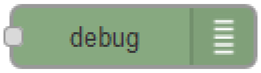
Edit http in node

Method: GET

URL: /yrgeolocation

Name: Yr geo weather

4. Locate the debug node under the output section in the palette window.

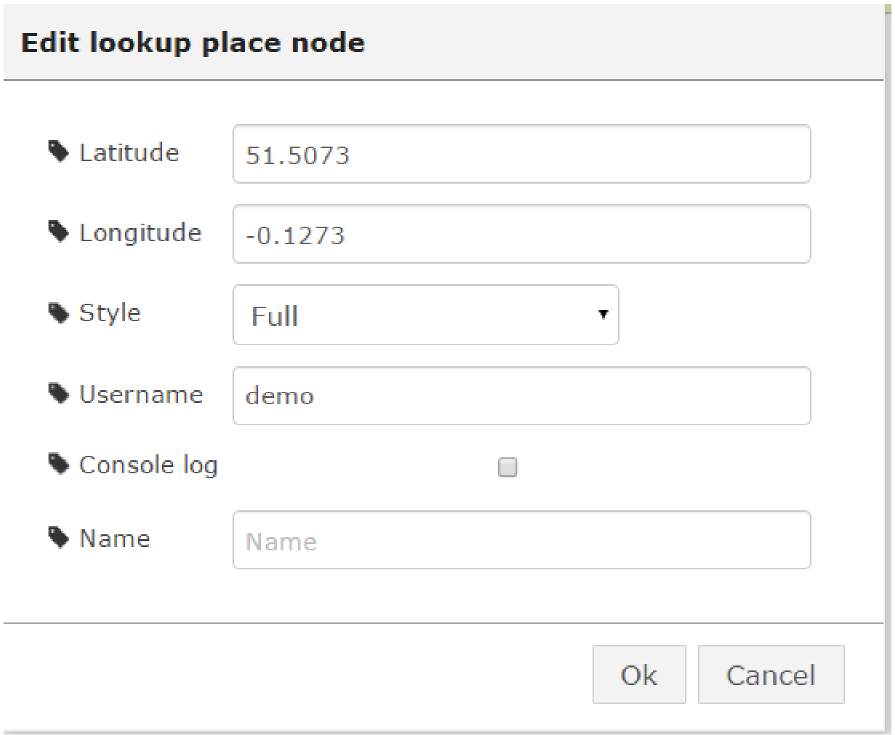


5. Drag the node onto the workspace, to the right of the http node.

6. Connect the http node with the debug node by dragging the output and input dots together.

7. Locate the lookup-place node in the geonames section (you might have to scroll down to the bottom of the palette) in the palette window and drag the node onto the workspace, to the right of the http node.

8. Double-click the lookup-place node and modify the ‘Style’ property to ‘Full’:



Edit lookup place node

Latitude: 51.5073

Longitude: -0.1273

Style: Full

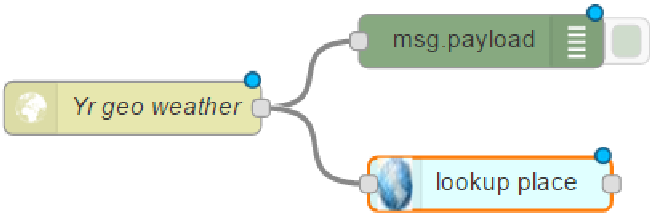
Username: demo

Console log: □

Name: Name

Please note that the default ‘Username’ is ‘**demo**’. This is Geonames.org demo user with limited number of requests. You can register your own username for free here: http://www.geonames.org/login

9. Connect the http node with the lookup-place node by dragging the http node output to the lookup-place node input. Your flow should now look something like this:



10. Locate the weather-info node in the yr section in the palette window and drag the node onto the workspace, to the right of the lookup-place node.

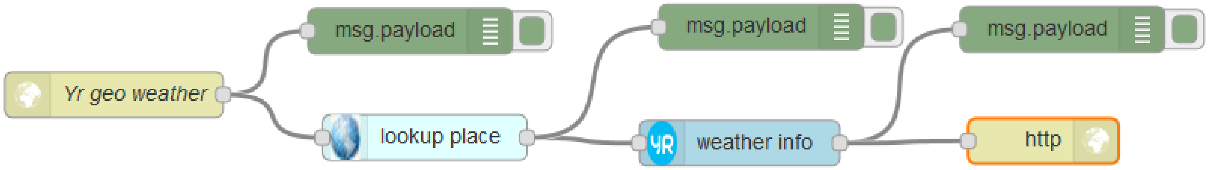
11. Connect the lookup-place node with the weather-info node by dragging the lookup-place node output to the weather-info node input.

12. Locate the http-response node in the output section in the palette window and drag the node onto the workspace, nearby the function node.

13. Connect the weather-info node with the http-response node by dragging the function node output to the http-response node input

14. **Optional**: you can also add debug nodes to the output of the lookup-place and weather-info nodes to log the payload in the debug window to the right.

15. The flow should now look like below. Press the ‘Deploy’ button to save and deploy the flow.



16. You can now test the flow by entering this URL in a browser (substitute ‘jotest’ with **<your application name>**)

    http://jotest.eu-gb.mybluemix.net/yrgeolocation?latitude=60.4&longitude=5.33

    This should return this JSON payload:

    {

    - **yr**: "http://www.yr.no/place/Norway/Hordaland/Bergen/Bergenhus/",
    - **meteogram**: "http://www.yr.no/place/Norway/Hordaland/Bergen/Bergenhus/meteogram.png",
    - **weather-data**: "http://www.yr.no/place/Norway/Hordaland/Bergen/Bergenhus/forecast.xml",
    - **place**: "Bergenhus,Hordaland,Norway",
    - **latitude**: "60.40123",
    - **longitude**: "5.32561"

    }

17. To use the flow in Google Maps, we create a new flow, locate the http node in the input section in the palette window and drag the node onto the workspace, below the existing flow.

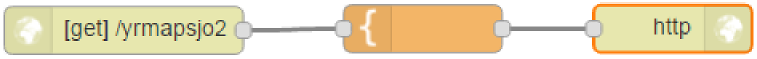
18. Double click on your new http node and enter a value for the URL field, e.g. ‘/yrmaps’, and optionally a name.

19. Locate the template node in the function section in the palette window and drag the node onto the workspace.

20. A sample Google Maps client is provided here: https://raw.githubusercontent.com/tverilytt/NODE-Red/master/samples/yrmaps.html

    The file is in this GitHub project: https://github.com/tverilytt/NODE-Red

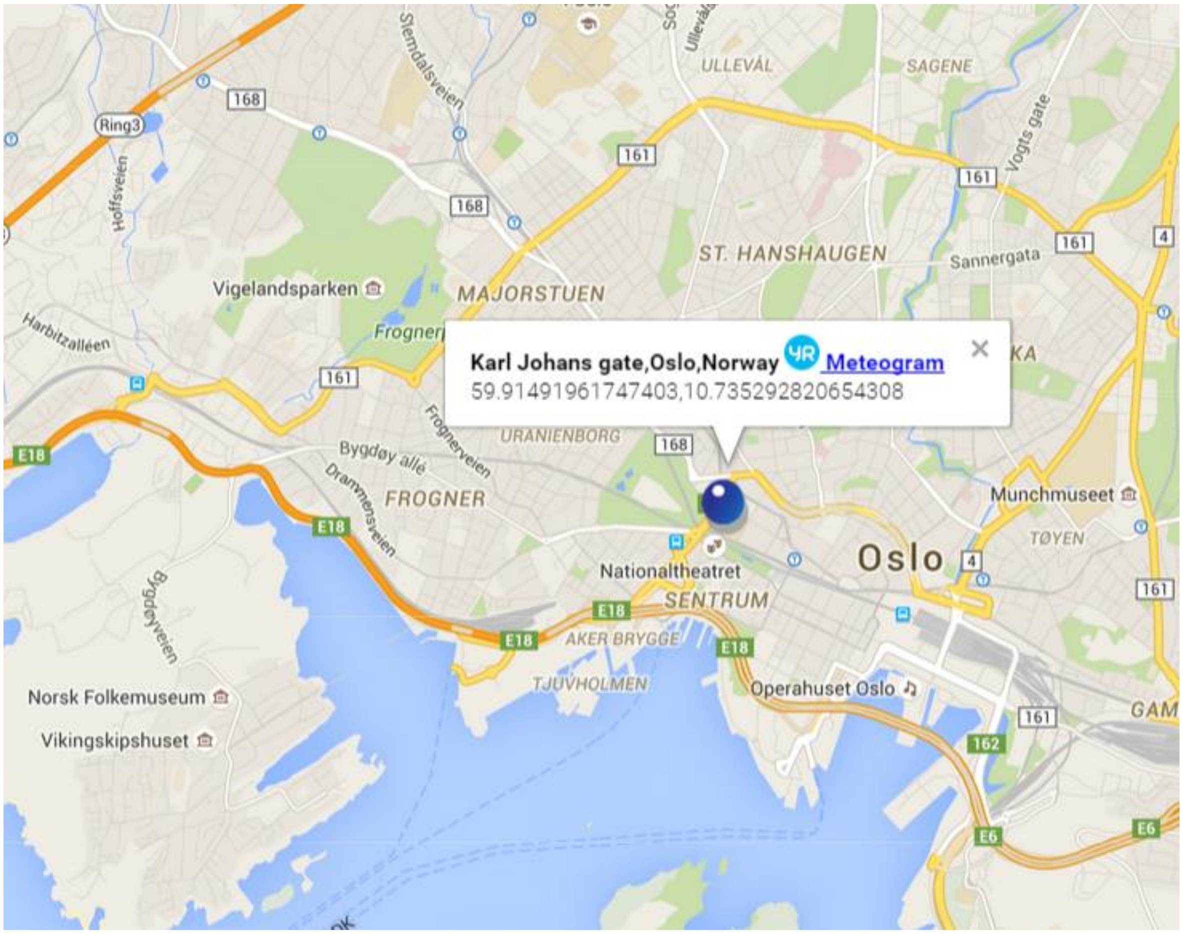
21. Copy the content of the sample and double-click on the template node and paste into the Template text area section, **replacing the existing content**, and press ‘Ok’ to save.
22. Connect the function node with the http-response node by dragging the function node output to the http-response node input
23. Locate the http-response node in the output section in the palette window and drag the node onto the workspace.
24. Connect the template node with the http-response node by dragging the template node output to the http-response node input.
25. The flow should now look like this. Press the ‘Deploy’ button to save and deploy the flow.



26. Test the sample client with this URL in a browser (substitute ‘jotest’ with **<your application name>**): http://jotest.eu-gb.mybluemix.net/yrmaps

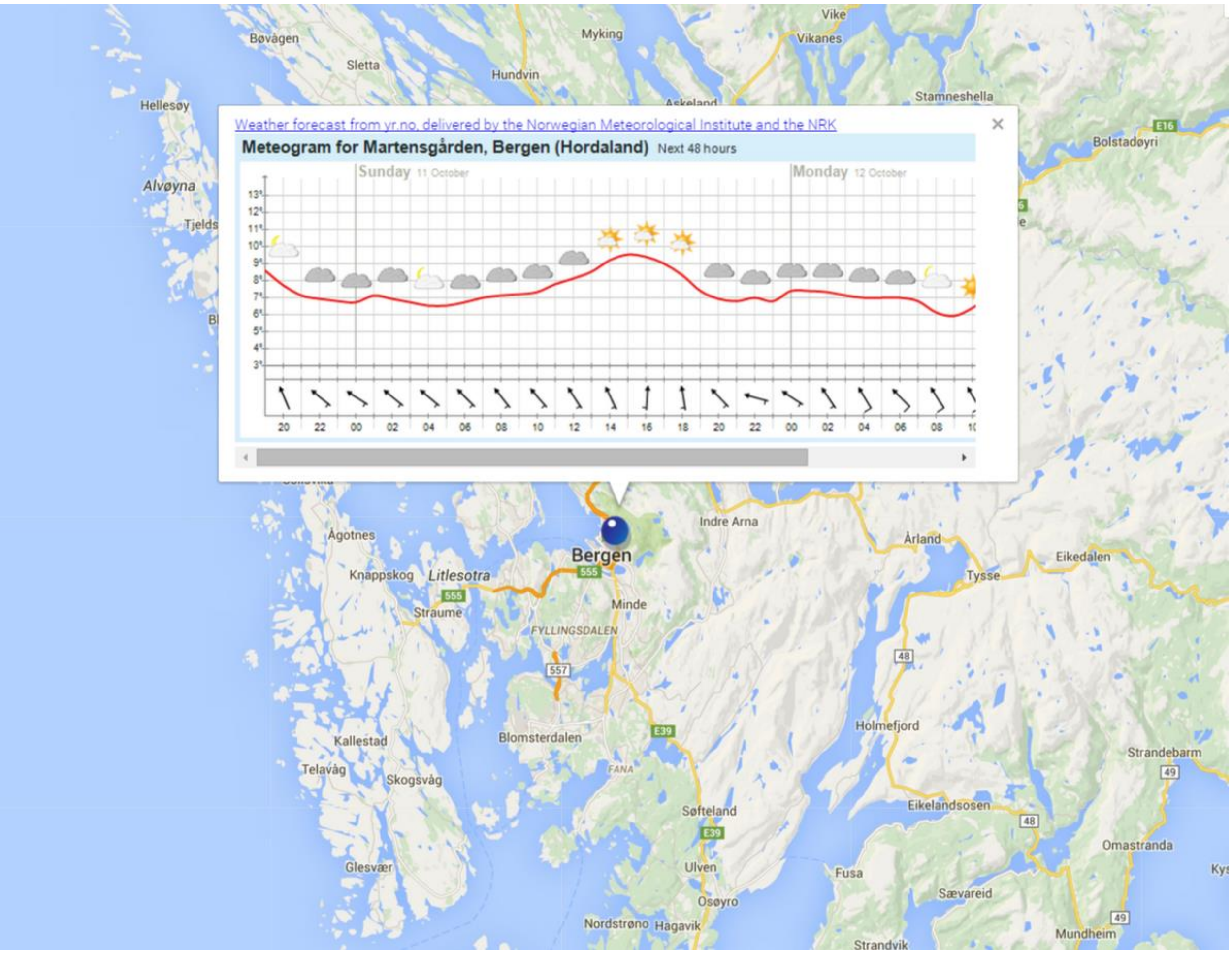
    The browser will ask if you want to share your location. If you choose to not share location, the start position will be in Oslo.

The sample client should look something like below. You can drag the blue ball marker around and release it to get Yr weather information for a desired geolocation. Click on the blue ball to get information about the name of the current location (if it exists), and link to the Yr meteogram. Click on the Yr meteogram link to open the meteogram in a larger Google Maps InfoWindow. Click on the Yr logo icon to open in a new browser pane, Yr.no weather information for the current location.



Karl Johans gate,Oslo,Norway Meteogram
59.91491961747403,10.735292820654308

If you click in the meteogram area, a new browser pane will open with Yr.no hour by hour weather information for this location.



Weather forecast from yr.no, delivered by the Norwegian Meteorological Institute and the NRK

Meteogram for Martensgården, Bergen (Hordaland) Next 48 hours

Please feel free to experiment with your own client!

Congratulations! You’ve successfully completed this lab where you've deployed a Bluemix application and used Node-RED to provide Yr.no weather information in Google Maps context.