Bluemix IoT LAB

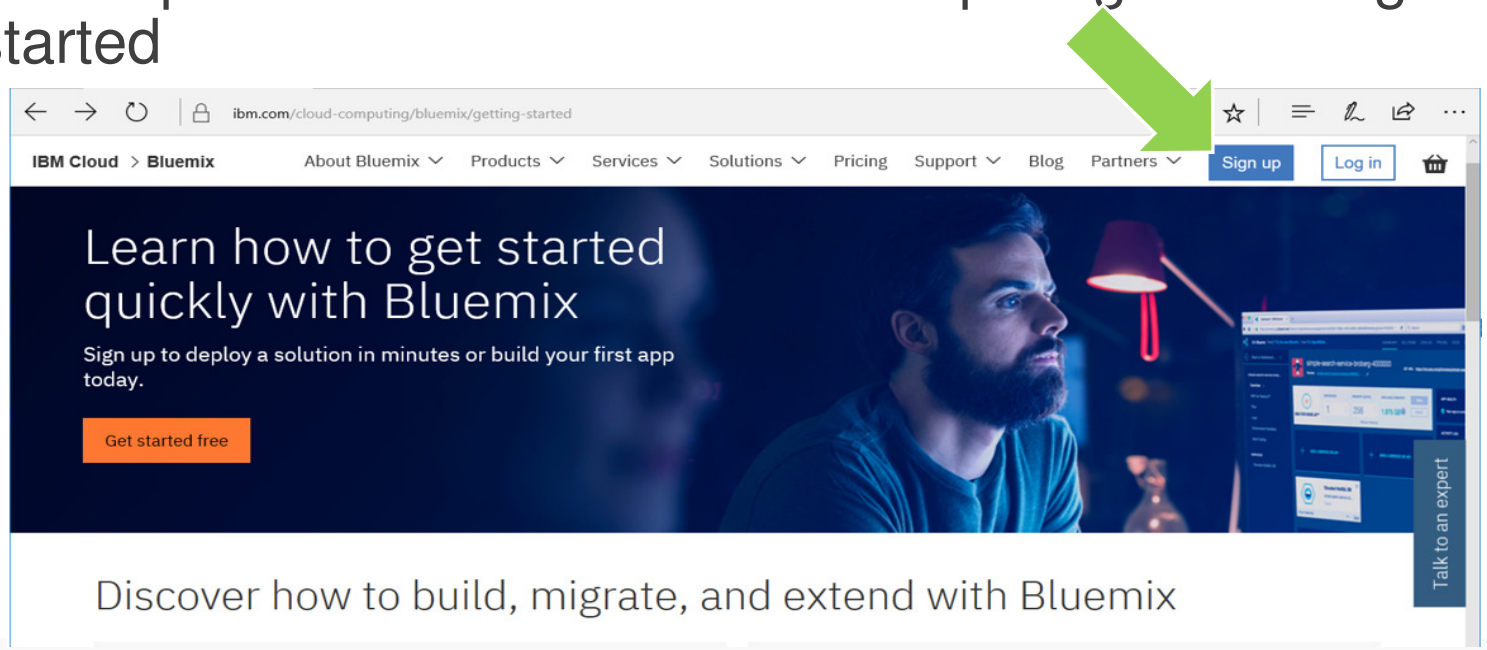
Attila Mák
06.12.2017

ARROW

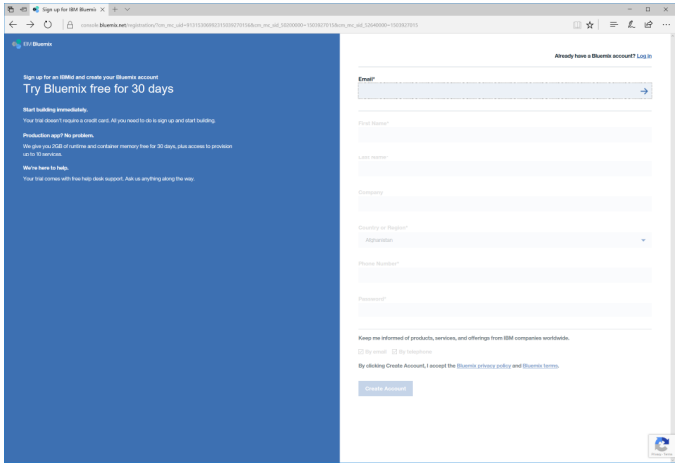
Five Years Out

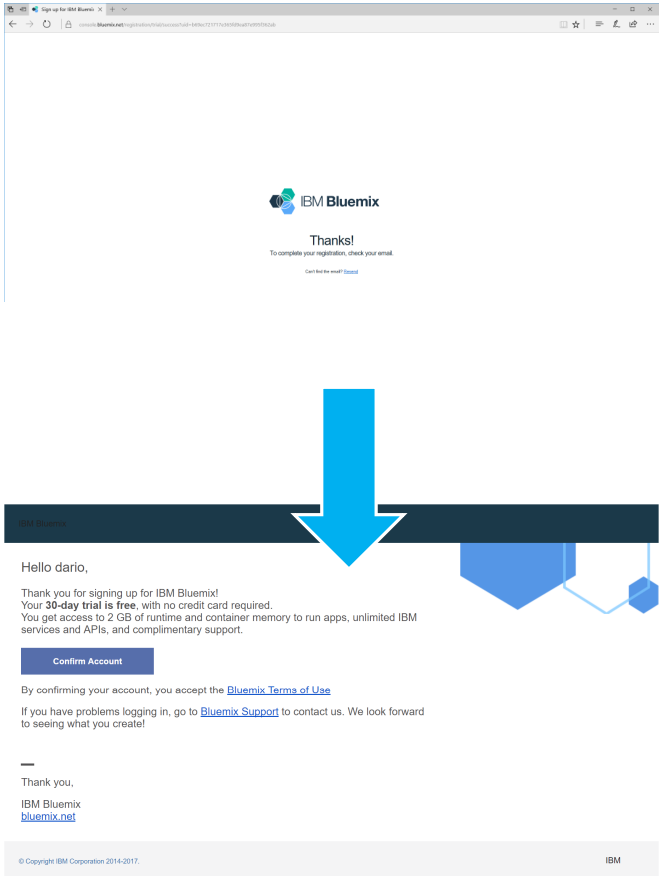
Getting started…

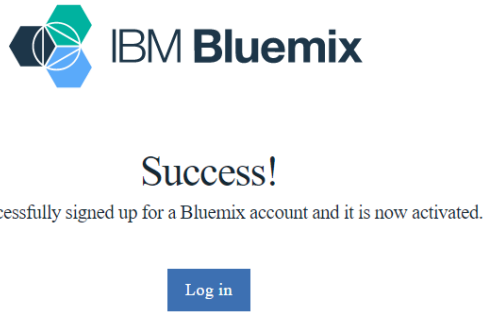
- Sign up for 30 day trial in bluemix at https://www.ibm.com/cloud-computing/bluemix/getting-started



Create account



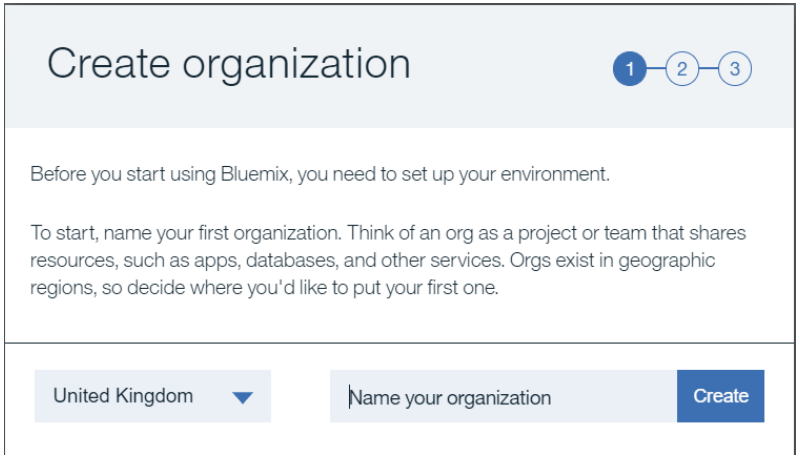




Login and create organization

- Get back to bluemix (https://www.ibm.com/cloud-computing/bluemix/getting-started) and sign in...

- You will be asked to create an organization this will be ne name of your test domain chose closest region to your site.

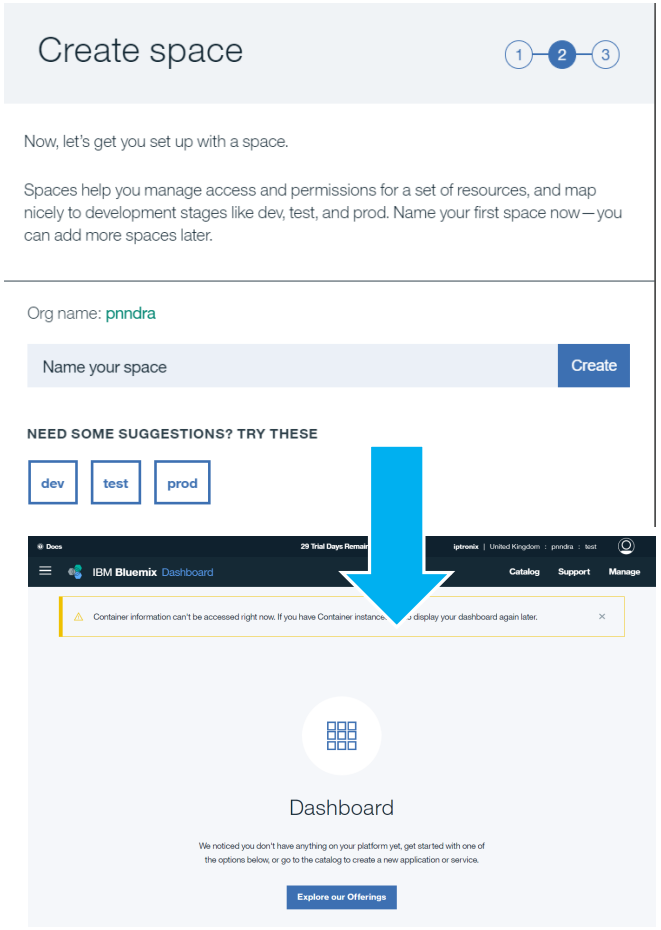


Create your space

- Next you will be asked to create a space

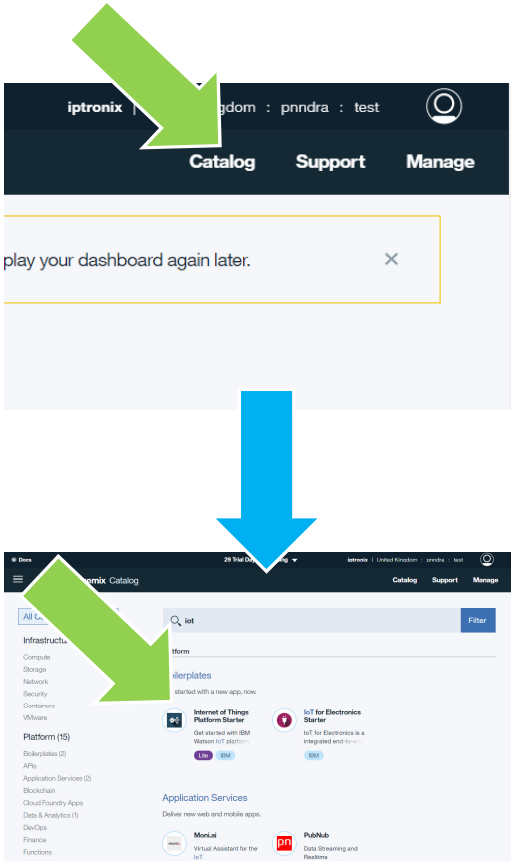
Spaces are containers for projects, you can create as many as you like.

Once finished you will get to the dashboard



Create a Watson IoT application

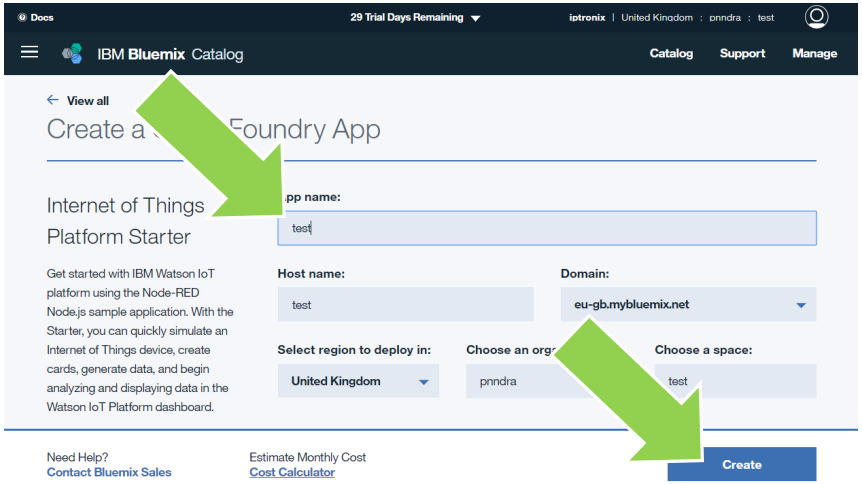
- From Dashboard click on Catalog



- Search for «iot» and select Internet of Things Platform Starter

Configure application

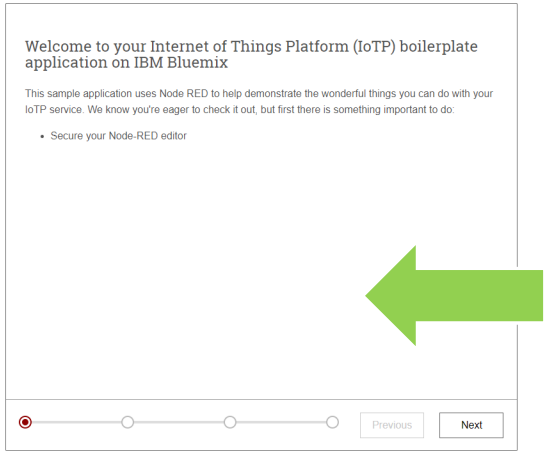
- Just enter some name in “app name” and leave everything as default.
  Note that name must be unique and that it must not contain symbols such as underscore

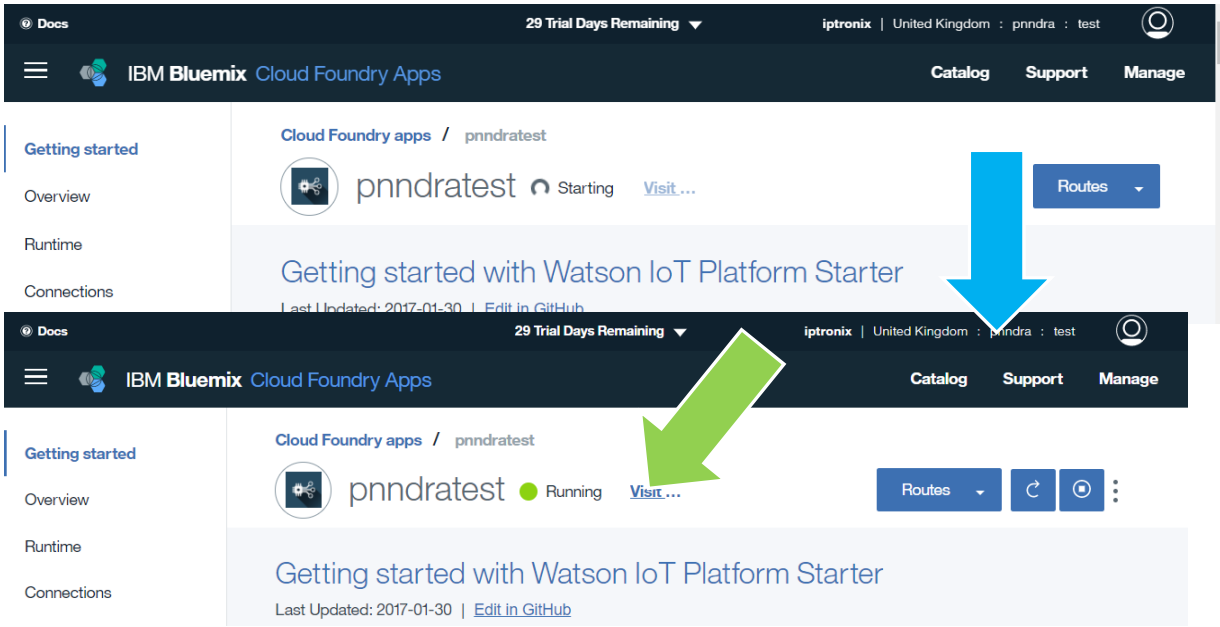


- Then click on Create

Getting around

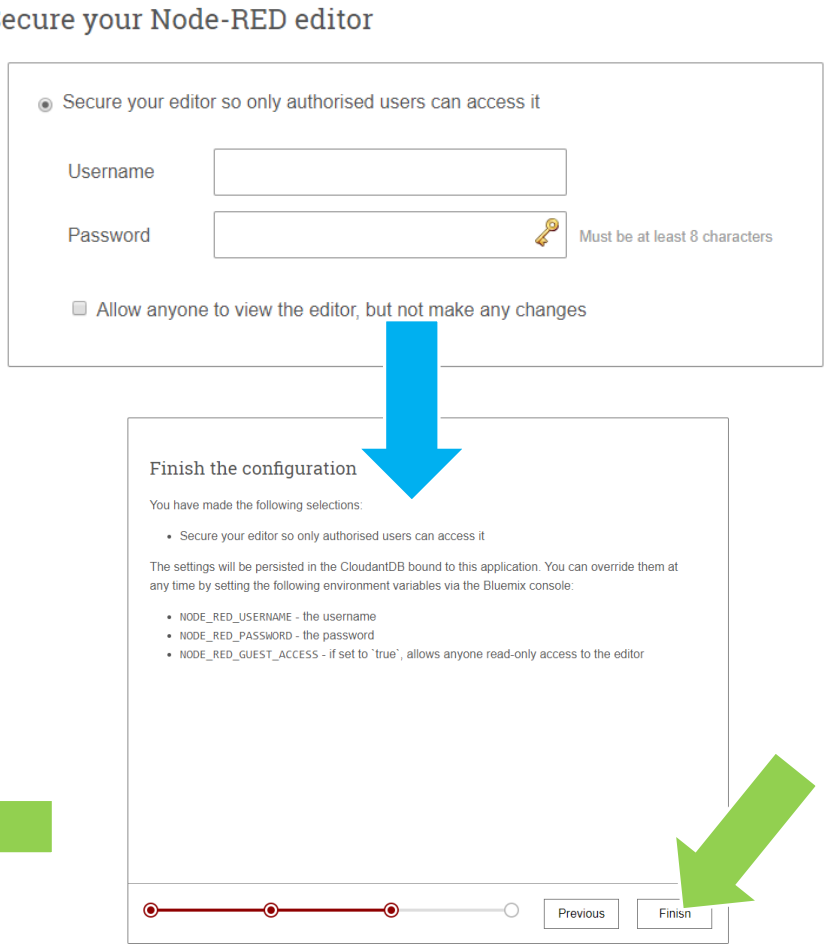
- After creating application it will take some time to start up. Please wait until status is running, then click on visit…
- You will get to node red setup screen…

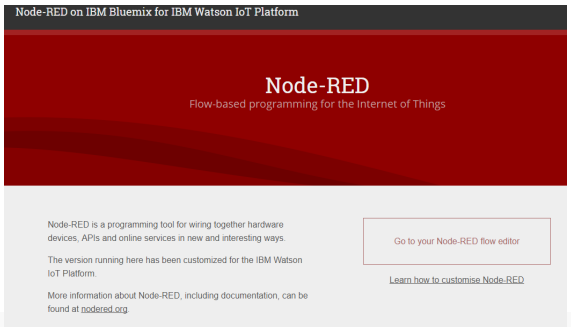




Node Red setup

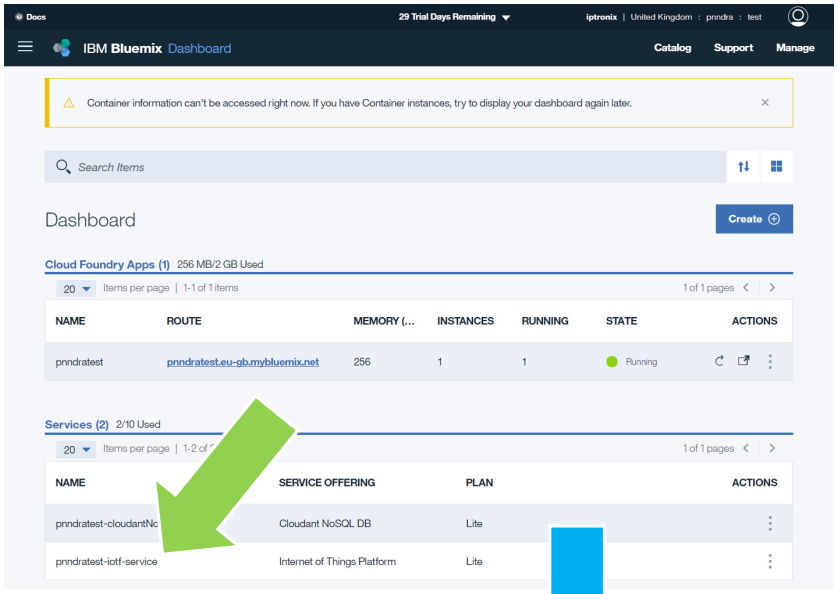
- Enter a username and password. These are not linked to your IBM credentials
- The next screen is just informative. Just click Finish. You will get to node red startup screen. Save this URL for later…

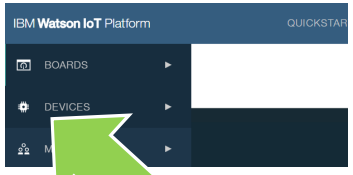


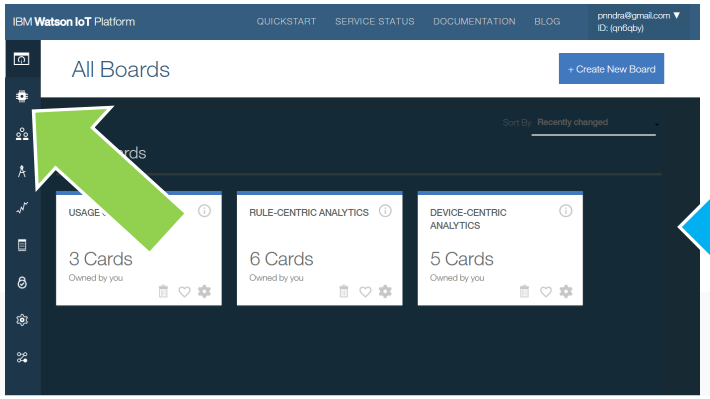


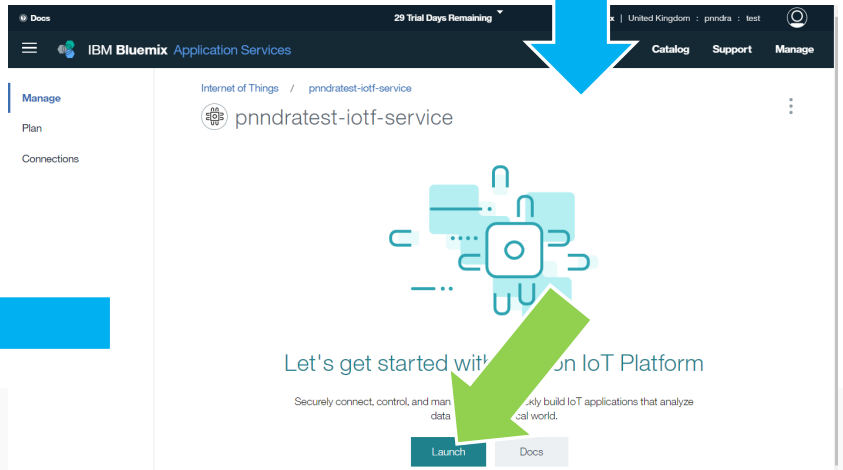
Set up Watson IoT

- Get Back to the dashboard (https://console.bluemix.net/dashboard) and click on <yourappname>-iotf-service
- You will get to the Watson IoT Platform home page. Click on Launch to get to the application. Click on the chip icon on the left bar…







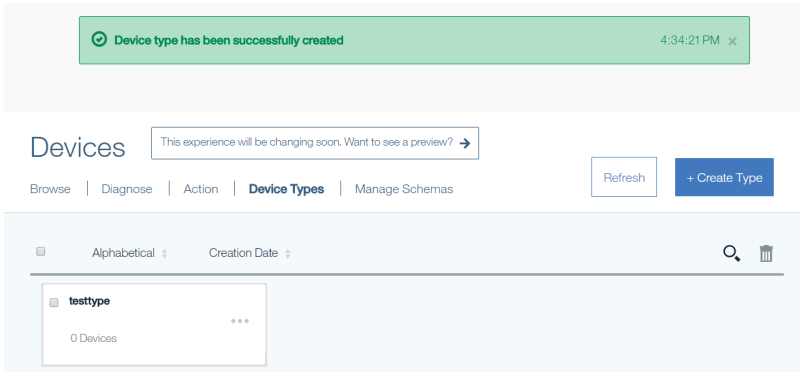


Watson IoT Concepts

- Watson IoT acts as a secure gateway between devices and the cloud
- Devices are categorized in device types which contain common information about devices
- Data is usually transferred through MQTT which allows sending and receiving messages.
- Messages over MQTT are usually transferred using JSON formatting

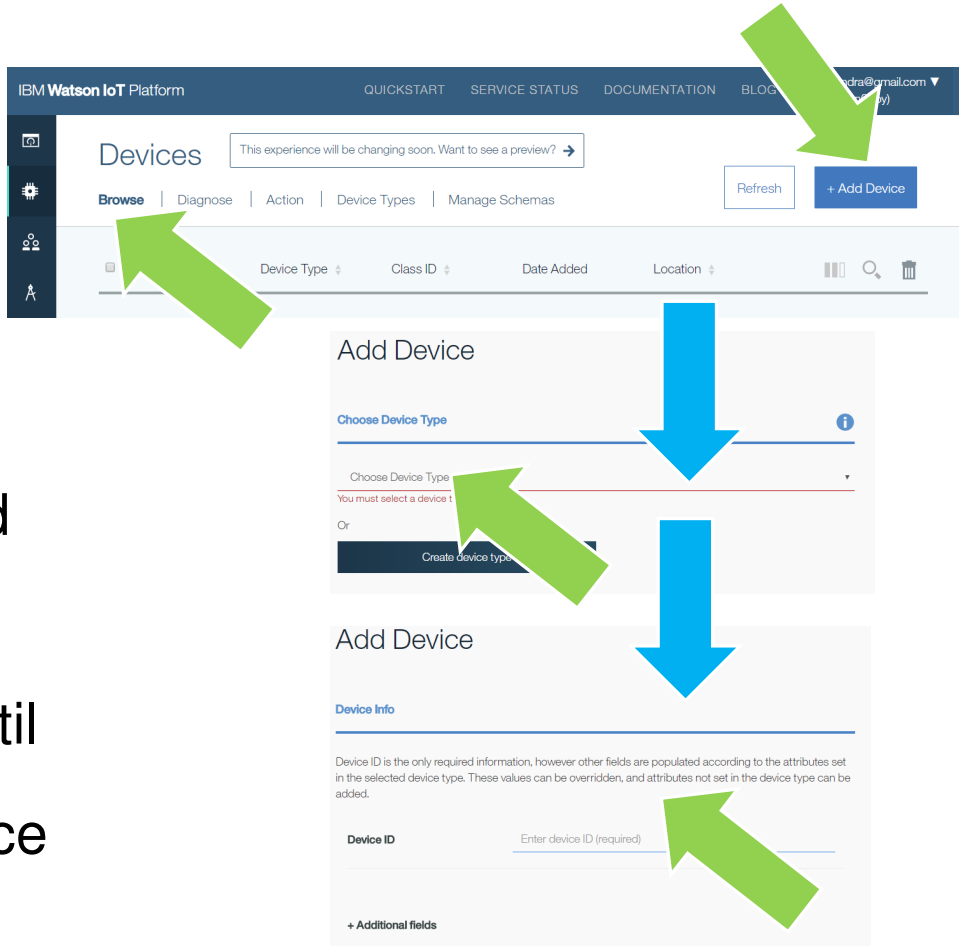
Set up device types

- Select Device Types and click on Create Type
- Chose Create Device Type and enter device type name and description. Enter any name here (no spaces or symbols) and click next
- In the subsequent pages just click next until you get back to console



Set Up Devices

- From Watson IoT Dashboard click on Browse and then on Add Device
- Select the device type we just created and click next
- Give device a name and click next until you get to the summary page, then click on add… you will get to the device credentials page



Device Credentials

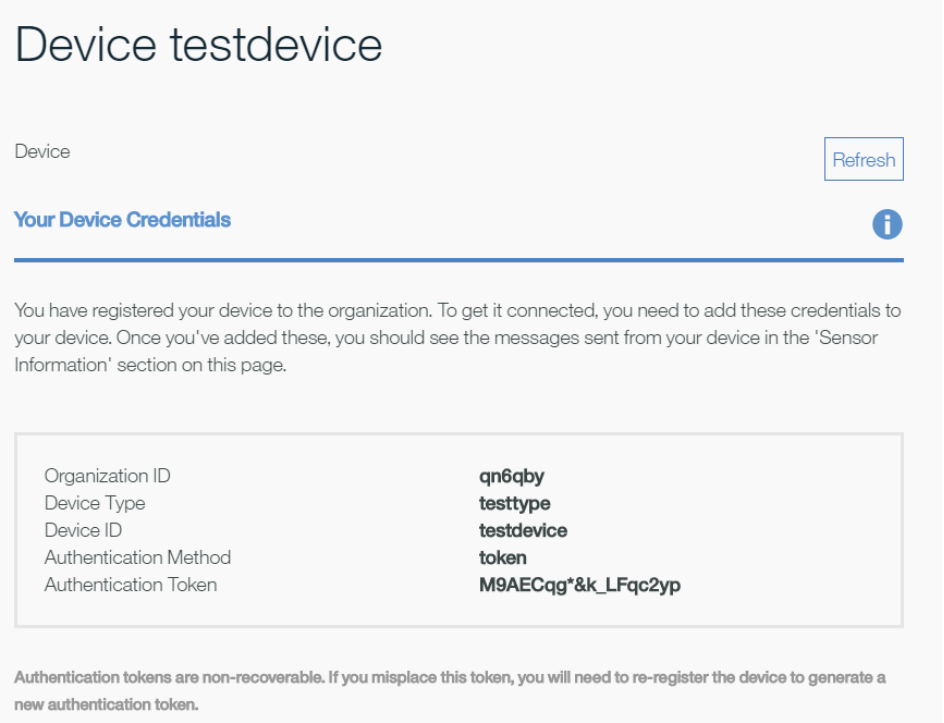
- In the device credentials page you will get the ONLY chance to write down Device's Authentication token. Write it down for later use…
- Write down also Organization ID as it will be needed to access MQTT broker.
- In the sample screenshot:

Device Yype: testtype

Device ID: testdevice

Organization ID: qn6qby

Authentication Token: M9AECqg*&k_LFqc2typ



Set up your hardware

- Provided you have a MQTT enabled board, your device should be set up as follows:
- MQTT Host:

  tcp://<org-id>.messaging.internetofthings.ibmcloud.com:1883

  ssl://<org-id>.messaging.internetofthings.ibmcloud.com:8883
- Username: use-token-auth
- Password: <Authentication token from device creation> (eg. M9AECqg*&k_LFqc2typ in previous screenshot)
- Device id: d:<Organization id>:<device type>:<device name> (eg. d: qn6qby:testtype:testdevice)
- Topic: iot-2/evt/message1/fmt/json

Setting up your hardware

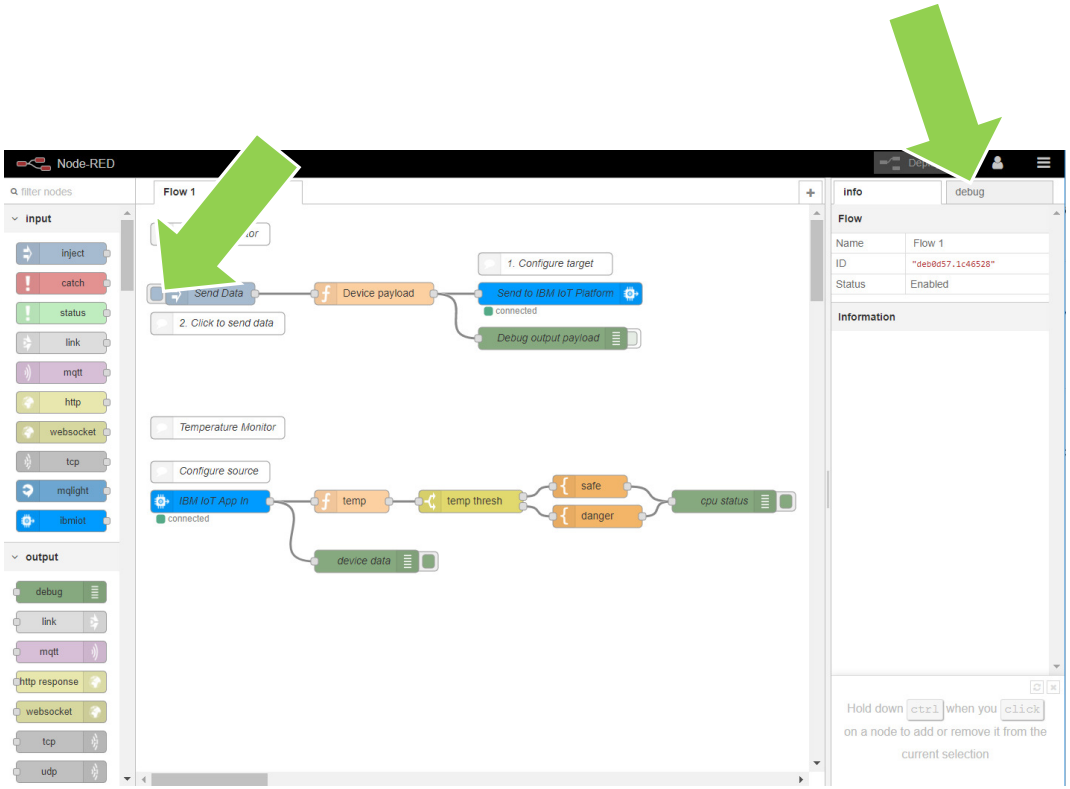
- Make sure your MQTT payload is in the following JSON format:

```
{
 “d”: {
          “temperature”: 27.0
          “status” : “ok”
        }
}
```

Getting started…

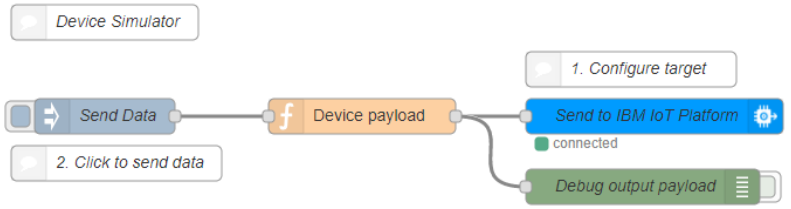
- Get back to node red login and enter your credentials
- You will be presented with a readymade graph that can be useful to get started.

- This simple graph allows injecting packets by clicking on send data and displaying them in the debug pane.



How it works… device simulator

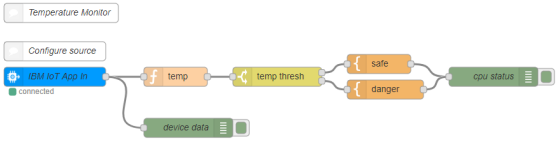
- Top part is a device simulator
- Send data block generates an event on click of the right square
- Device payload creates JSON packet through Js code
- Send to IBM IoT injects data in IBM MQTT broker
- Debug output payload shows sent data on debug pane (if enabled by clocking on the right square)

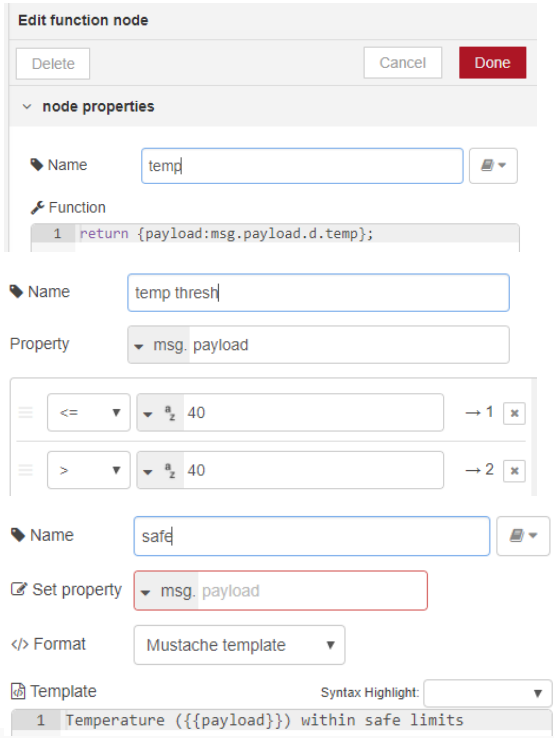



```
// Thermostat's location:
var longitude1 = -98.49;
var latitude1 = 29.42;

// Array of pseudo random temperatures
var temp1 = [15,17,18.5,20,21.5,23,24,22.2,19,18];

// Array of pseudo random relative humidities
var humidity1 = [50,55,61,68,65,60,53,49,45,47];

// Counter to select from array.
var counter1 = context.get('counter1')||0;
counter1 = counter1+1;
if(counter1 > 9) counter1 = 0;
context.set('counter1',counter1);

// Create MQTT message in JSON
msg = {
  payload: JSON.stringify(
    {
      d:{
        "temp" : temp1[counter1],
        "humidity" : humidity1[counter1],
        "location" :
        {
          "longitude" : longitude1,
          "latitude" : latitude1
        },
      }
    }
  )
};
return msg;
```

How it works… Temperature Monitor

- IBM IoT App In fetches data from MQTT broker (demo gets any device)
- Device data sends received data to debug pane
- Temp function extracts temperature from JSON payload
- Temp threshold sends event to either safe or danger, based on data value
- Safe and danger blocks modify event payload to a string with result message
- Cpu status sends final message to debug pane





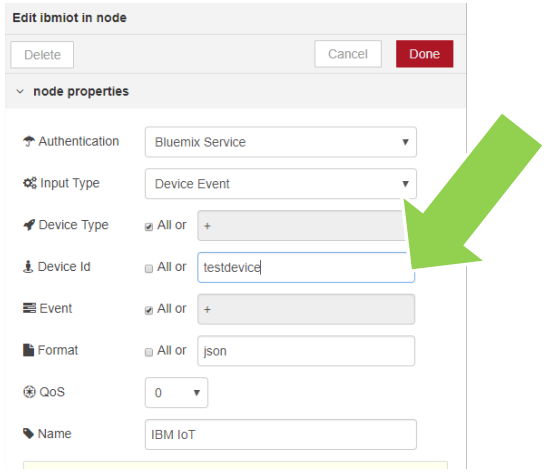
Running demo…

- Clicking on send data will create a message and send it to MQTT broker.
- Data will be received and dumped by the two debug blocks
- The device data debug block dumps all the incoming data (you can drill down on structures)

Receiving data from your device

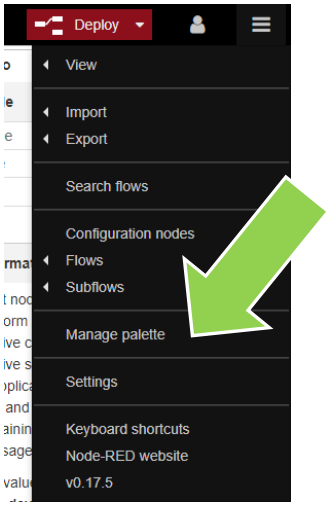
- Setup properties of IBM IoT App in
  - Enter your device id in Device Id
- Click on deploy 

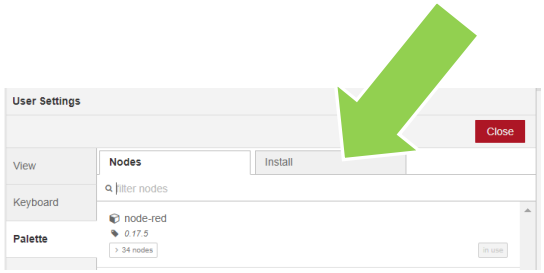
- Now you should be receiving data from your physical device and display it on debug pane

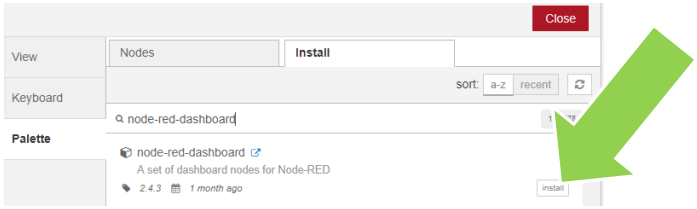


Setting up data visualization

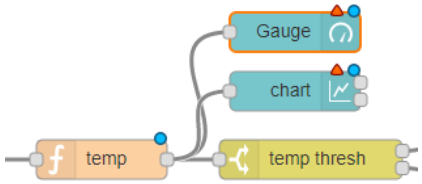
- Add ui blocks to palette:
  - Select manage palette from menu
  - Select install tab
  - Enter node-red-dashboard in the search box
  - Click install
  - Confirm install dialog
- This will install the dashboard palette at the end of your node list (left pane)



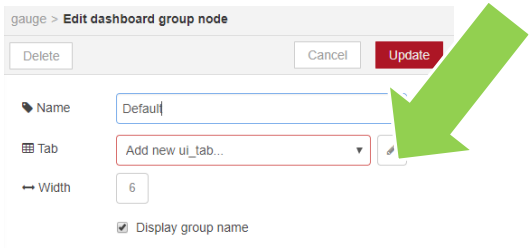


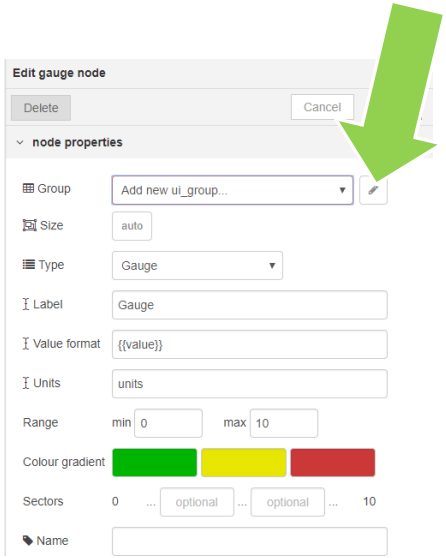


Add some visualization



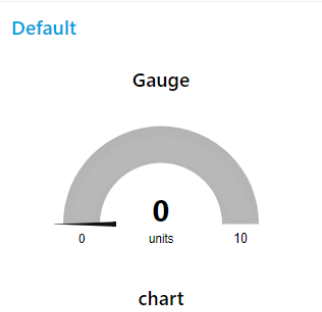
- Add and connect a gauge and a chart to the temp function
- Edit gauge:
  - Add a new UI group
  - Add a group to a new tab
  - Edit range
- Edit chart:
  - Just open and confirm… this will assign chart to the same group
- Deploy…
- Open your dashboard at the same url of nodered replacing red with ui, eg: https://pnndratest.eu-gb.mybluemix.net/ui

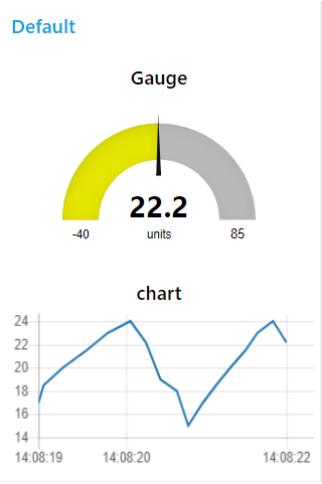




Testing the visualization

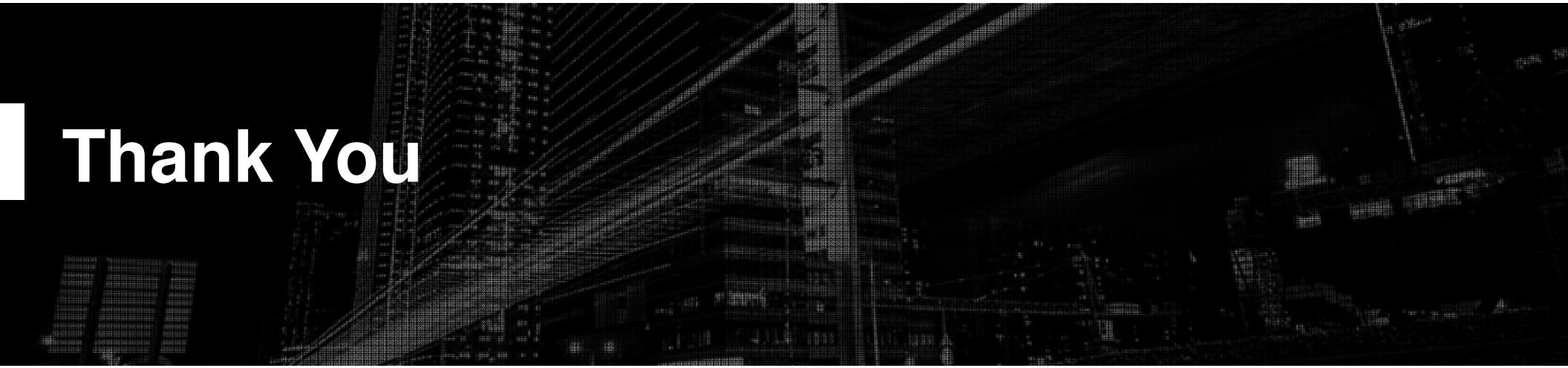
- Initially you will get an almost empty page
  - Blank chart and gauge set to 0
- Start injecting data from the device simulator or from you physical device…
- You will see chart and gauge updating in real time





Putting it all together

- Creating flow graphs with node red can be very easy but
  - Each dashboard is globally identical for each viewer (you have to have a flow for each device)
  - Node red graphs can grow very complex and can have a lot of overhead
- Node red is great for quick demos
- Once the Proof of concept is ok, a custom web application needs to be developed. Using NodeJS work done on Node red can be reused and extended
- Various sample graphs can be obtained from Arrow to demonstrate interface with various reference designs



Thank You