PiHeadquarters

or How to Take Control of your House with Raspberry Pi and Docker

João Pedro Dias

Sat. 30 Sept. 2017

PIXELS camp

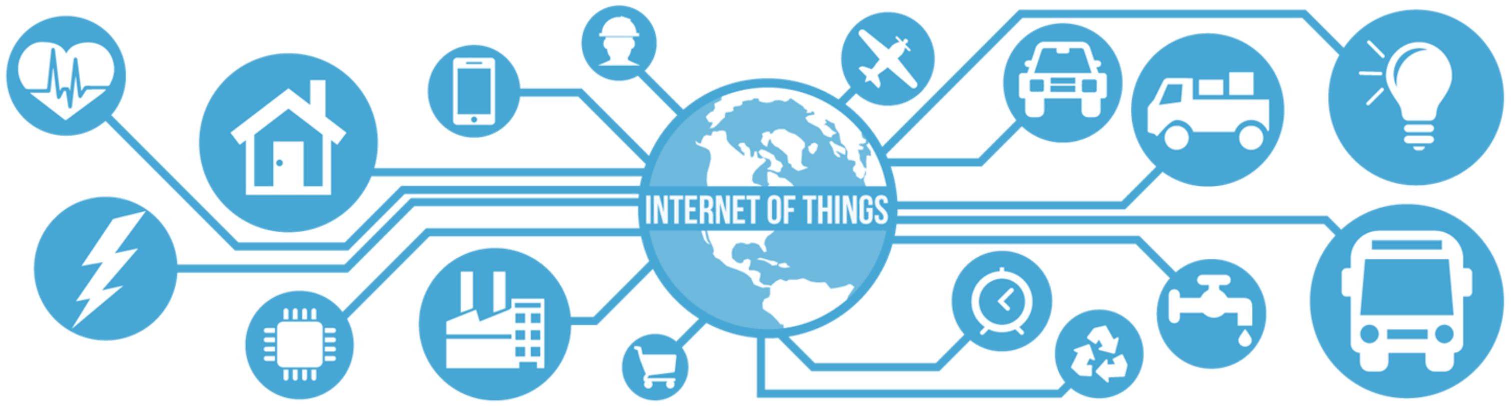
$ whoami

- From Porto, PT
- Researcher @ INESC TEC (since May, 2017)
- ProDEI Student @ FEUP
  - *Doctoral Programme in Informatics Engineering*
- Where to find me:
  - https://jpdias.me/
  - https://keybase.io/jpdias/

$ cat outline.txt

- Smart-*spaces* context
- Hardware overview
- Software overview
- Architecture
- Talk is cheap. *Show me the* ***code****!*
- *it's* DEMO *time!*

$ info

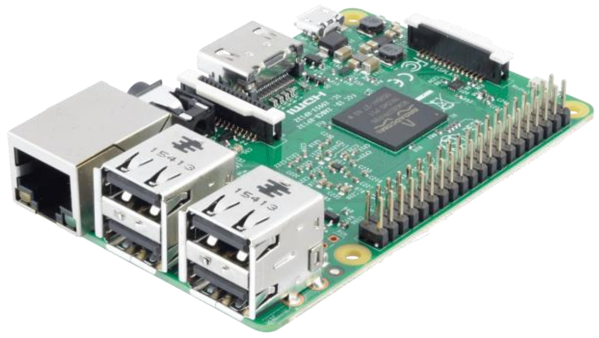

INTERNET OF THINGS

SUPPLY CHAIN & LOGISTICS
HEALTHCARE
METERING
AGRICULTURE
ENVIRONMENT
INDUSTRIAL CONTROL
CITIES
HOME & BUILDINGS

$ hwinfo Raspberry Pi 3

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN
- Bluetooth Low Energy (BLE)
- 40-pin extended GPIO

- Price: ~ 35$ (e.g. amazon)



$ uname -a

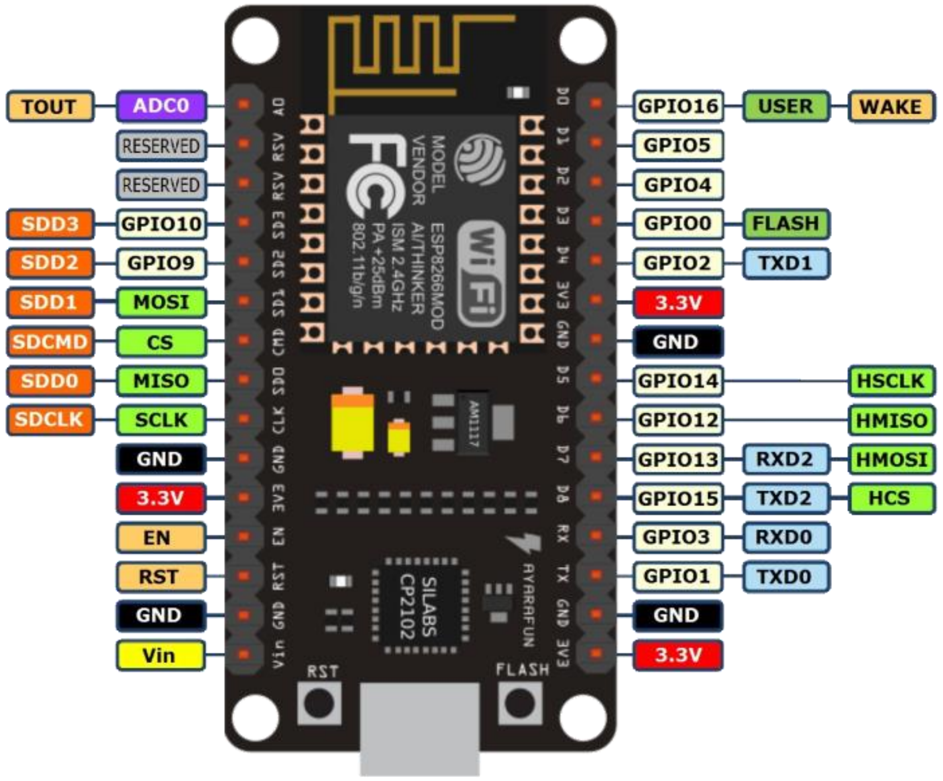
- **MINIBIAN image**
- Raspbian *Jessie* based
- Kernel 4.1.18+ #846
- ~ 15 secs boot
- ~ 31 MB RAM used
- ~ 477 MB disk space used
- DHCP client enabled
- SSHD enabled
- root user enabled
  - *(default password: raspberry – change it ASAP)*

Resources:

https://minibianpi.wordpress.com/

$ hwinfo Node-MCU ESP-12E (ESP8266)

- Wireless 802.11 b / g / n
  - *Support STA / AP / STA + AP*
- OTA
  - *Remote firmware upgrade*
- Programming Languages:
  - *C (Arduino compatible), Lua and MicroPython*

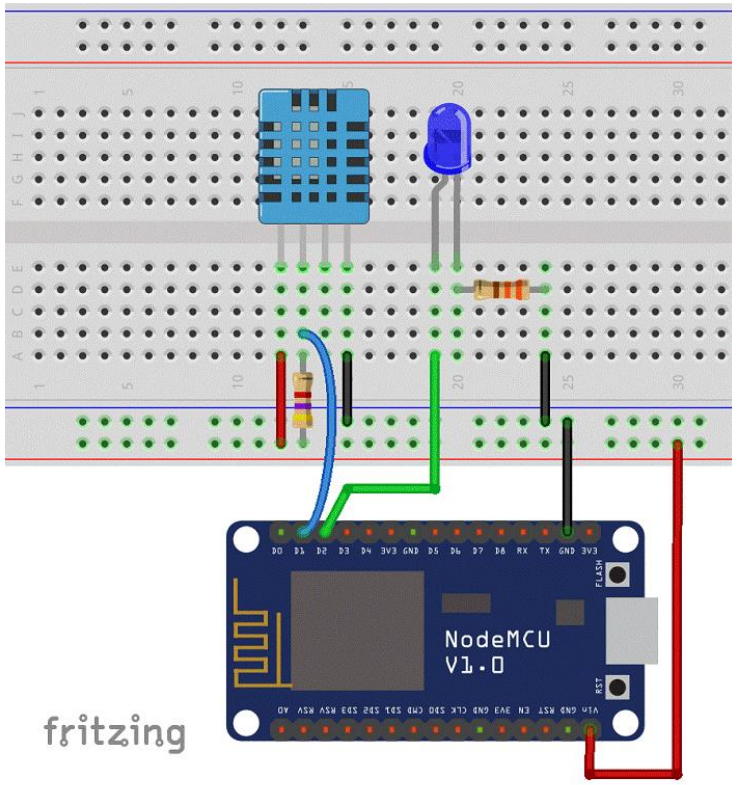


- Price: ~ 4$ (e.g. ebay)

$ hwinfo circuit

- 1x LED
- 1x DHT11 or DHT22 (temp/humidity sensor)
- Jump Cables
- 1x Breadboard
- 330 Ω resistor
- 4700 Ω resistor

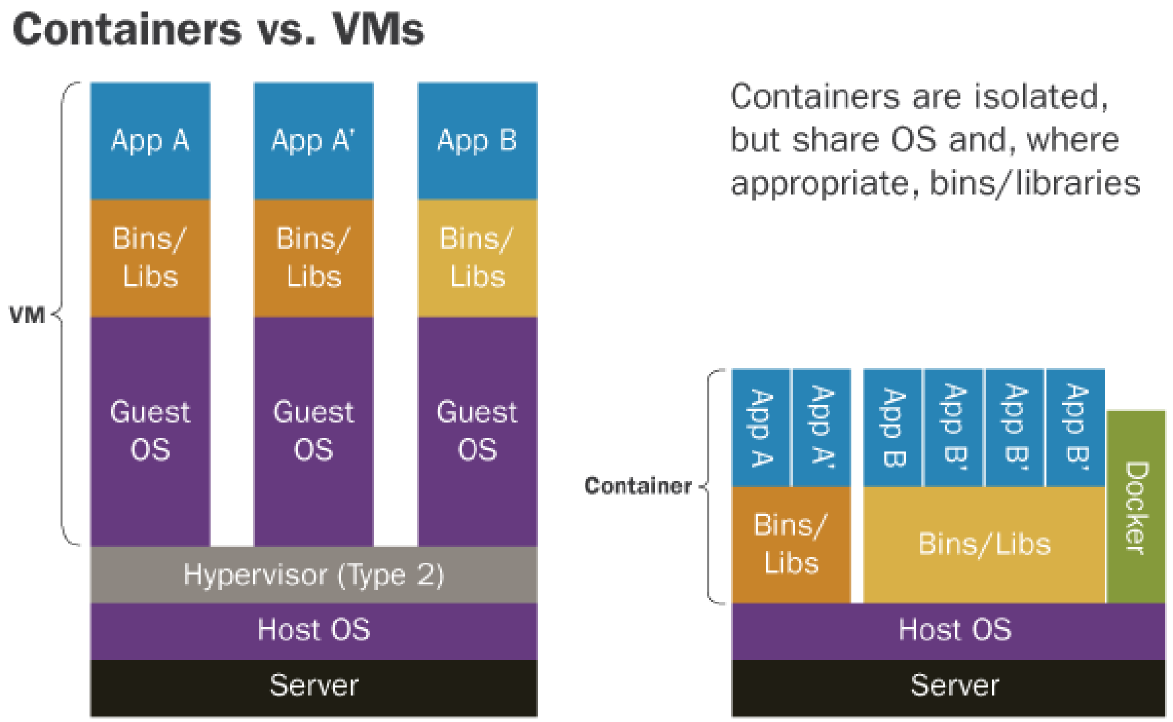
```
WiFi.begin(wifi_ssid, wifi_password);

Serial.println(WiFi.localIP());
```



$ man docker

"Docker is a tool that can package an application and its dependencies in a virtual container that can run on any Linux server. This helps enable flexibility and portability on where the application can run, whether on premise, public cloud, private cloud, bare metal, etc."

Source: https://www.linux.com/news/docker-shipping-container-linux-code

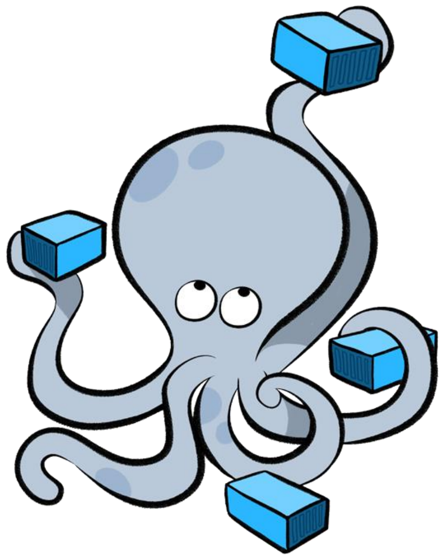


$ man docker-compose

“Compose is a tool for **defining and running multi-container Docker applications**. With Compose, you use a Compose file to configure your application's services. Then, using a single command, you create and start all the services from your configuration.”

Source: https://github.com/docker/compose

For package install help (using apt-get on Raspbian): https://blog.hypriot.com/post/your-number-one-source-for-docker-on-arm/



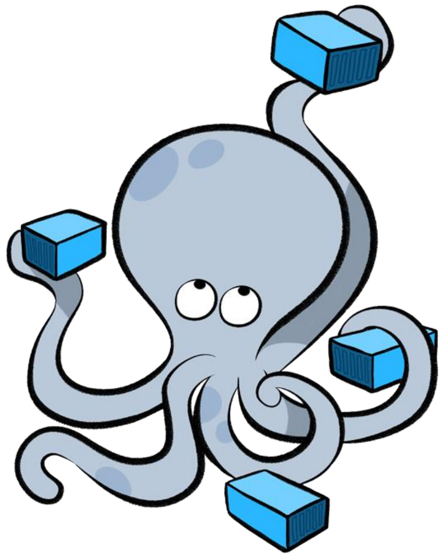
$ man docker-compose

1. Define your app's environment with a `Dockerfile` so it can be reproduced anywhere.
2. Define the services that make up your app in `docker-compose.yml` so they can be run together in an isolated environment.
3. Lastly, run `docker-compose up` and Compose will start and run your entire app.

Source: https://github.com/docker/compose

Sample `docker-compose.yml`

```
version: '2'

services:
  web:
    build: .
    ports:
      - "5000:5000“
    volumes:
      - .:/code
  redis:
    image: redis
```



$ man mqtt

"MQTT is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport."

Source: http://mqtt.org/

```
PubSubClient client(espClient);

void callback(char* topic, byte* payload, unsigned int length) { … }

void setup() { …
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);}

void reconnect(){ …
    client.connect((char *)clientName.c_str())
    client.subscribe(actuator_led_status);}

void loop() { …
    client.publish(temperature_celsius_topic, String(t).c_str(), true);}
```

$ man mosquitto

"Eclipse Mosquitto™ is an open source (EPL/EDL licensed) message broker that implements the MQTT protocol versions 3.1 and 3.1.1."

Source: https://mosquitto.org/

$ man influxdb

"InfluxDB is an open-source time series database. It is written in Go and **optimized for fast, high-availability storage and retrieval of time series data** in fields such as operations monitoring, application metrics, Internet of Things sensor data, and real-time analytics."

Source: https://docs.influxdata.com/influxdb/v1.3/guides/

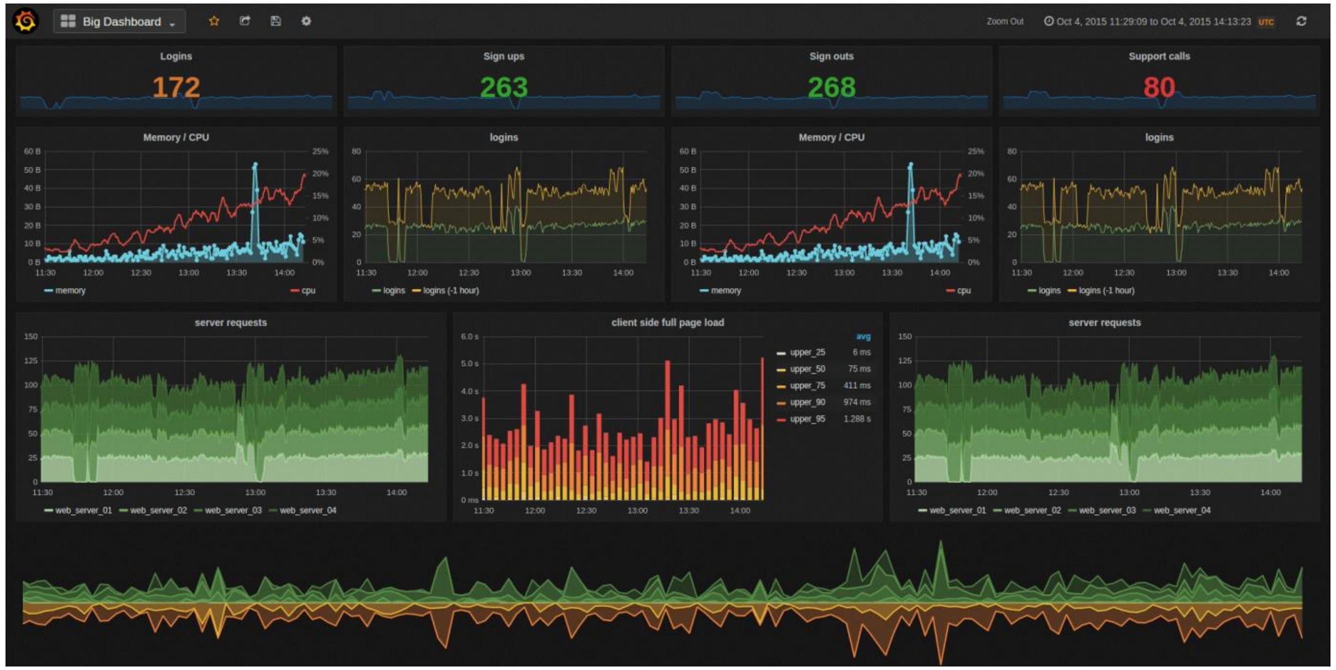
Resources:
https://docs.influxdata.com/influxdb/v1.3/guides/

$ man grafana

"Grafana is an open source, feature rich metrics dashboard and graph editor for Graphite, Elasticsearch, OpenTSDB, Prometheus and *InfluxDB*."

Source: https://github.com/grafana/grafana

Resources:
http://docs.grafana.org/guides/getting_started/



$ man telegraf

"Telegraf is an agent written in Go for **collecting, processing, aggregating, and writing metrics.**

Design goals are to have a minimal memory footprint with a plugin system so that developers in the community can easily add support for collecting metrics from well known services (like Hadoop, Postgres, or Redis) and third party APIs (like Mailchimp, AWS CloudWatch, or Google Analytics)."

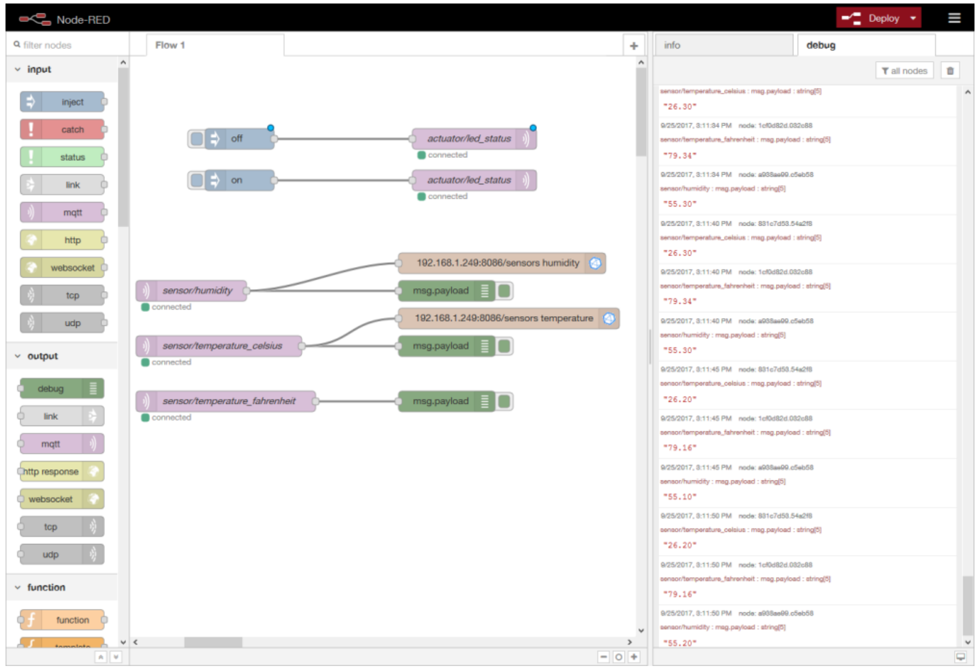
Source: https://github.com/influxdata/telegraf

Resource: https://www.influxdata.com/time-series-platform/telegraf/

$ man node-red

"Node-RED is a programming tool (VPL) for wiring together hardware devices, APIs and online services (…).

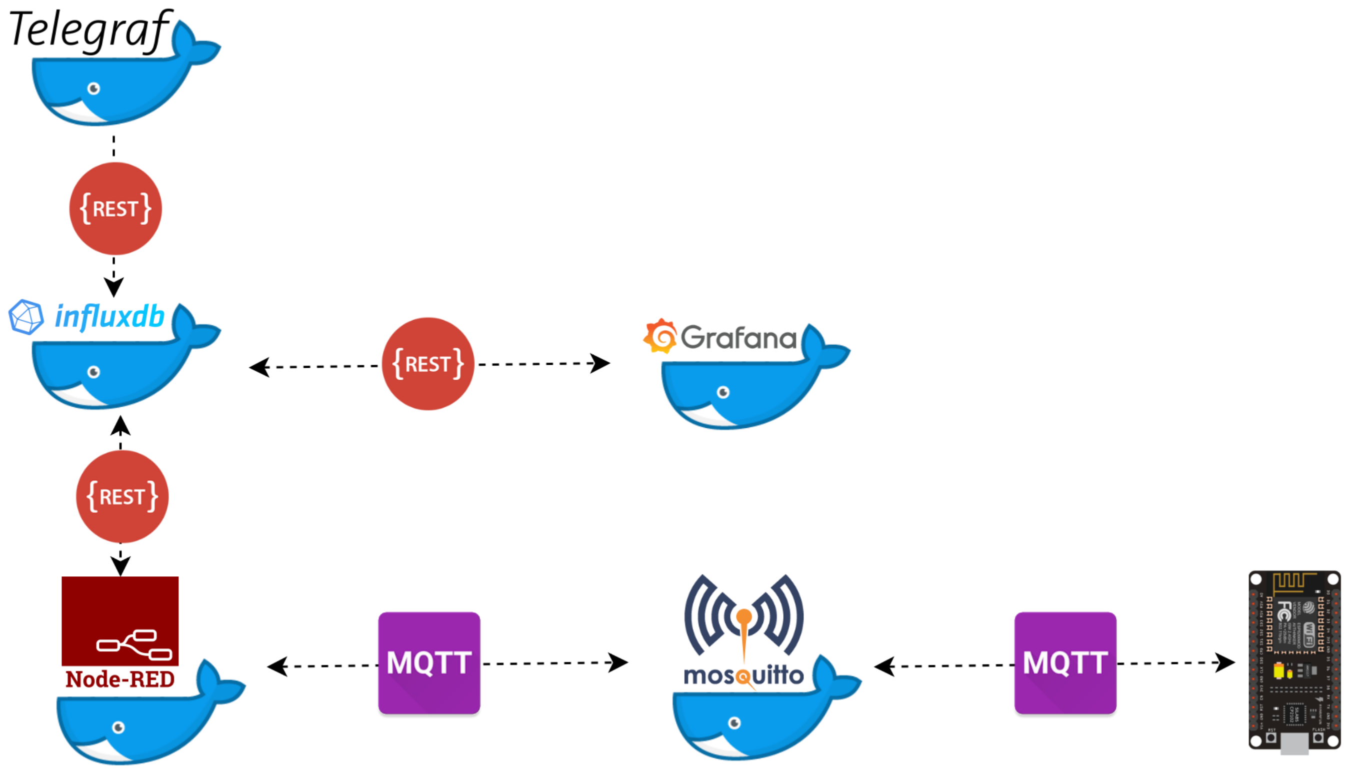
It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click."



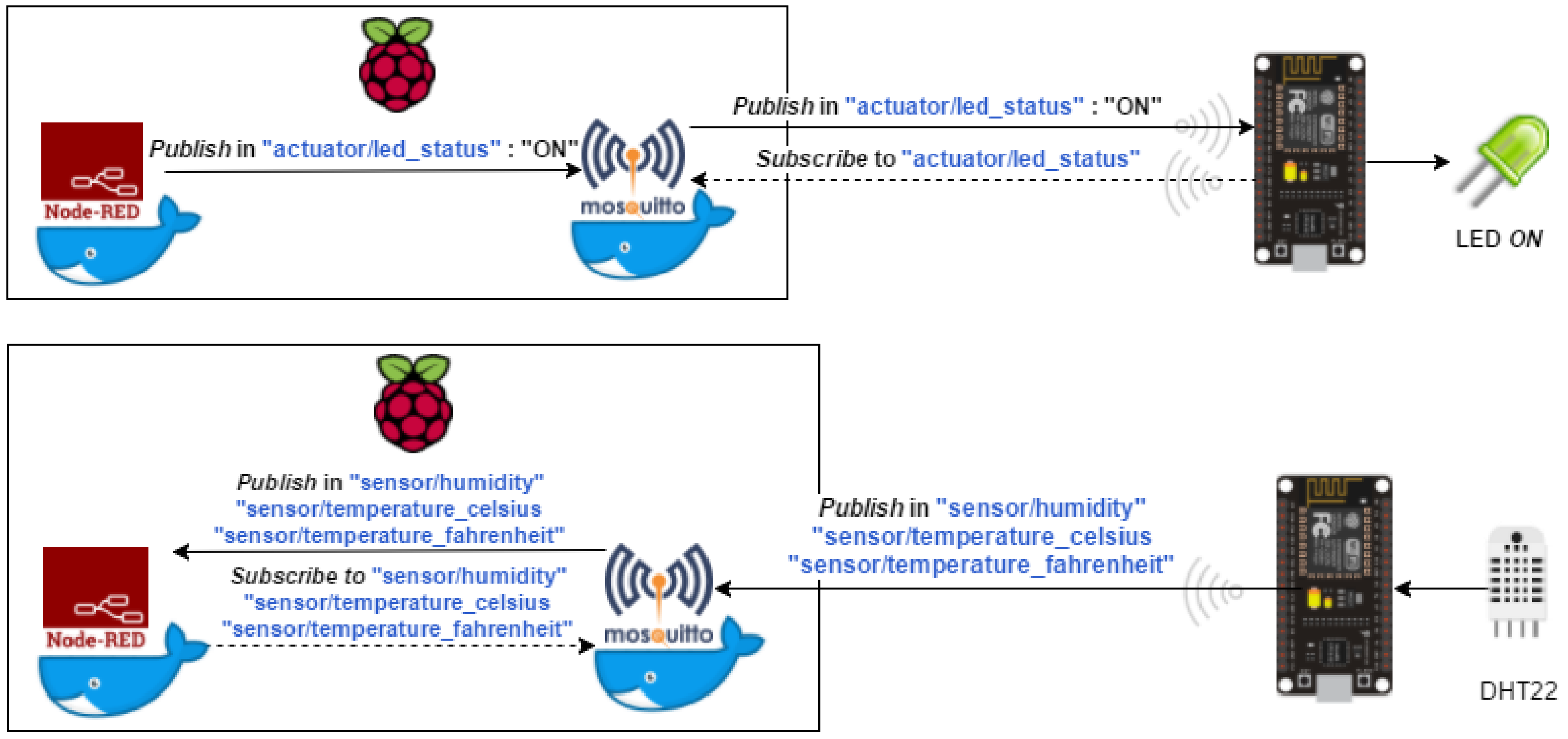
```
sensor/temperature_celsius : msg.payload : string[5]
 "26.30"
9/25/2017, 8:11:34 PM   node: 1cf0d82d.032c88
sensor/temperature_fahrenheit : msg.payload : string[5]
 "79.34"
9/25/2017, 8:11:34 PM   node: a938ae99.c5eb58
sensor/humidity : msg.payload : string[5]
 "55.30"
9/25/2017, 8:11:40 PM   node: 831c7d53.54a2f8
sensor/temperature_celsius : msg.payload : string[5]
 "26.30"
9/25/2017, 8:11:40 PM   node: 1cf0d82d.032c88
sensor/temperature_fahrenheit : msg.payload : string[5]
 "79.34"
9/25/2017, 8:11:40 PM   node: a938ae99.c5eb58
sensor/humidity : msg.payload : string[5]
 "55.30"
9/25/2017, 8:11:45 PM   node: 831c7d53.54a2f8
sensor/temperature_celsius : msg.payload : string[5]
 "26.20"
9/25/2017, 8:11:45 PM   node: 1cf0d82d.032c88
sensor/temperature_fahrenheit : msg.payload : string[5]
 "79.16"
9/25/2017, 8:11:45 PM   node: a938ae99.c5eb58
sensor/humidity : msg.payload : string[5]
 "55.10"
9/25/2017, 8:11:50 PM   node: 831c7d53.54a2f8
sensor/temperature_celsius : msg.payload : string[5]
 "26.20"
9/25/2017, 8:11:50 PM   node: 1cf0d82d.032c88
sensor/temperature_fahrenheit : msg.payload : string[5]
 "79.16"
9/25/2017, 8:11:50 PM   node: a938ae99.c5eb58
sensor/humidity : msg.payload : string[5]
 "55.20"
```

Node-RED

$ tree piheadquarters



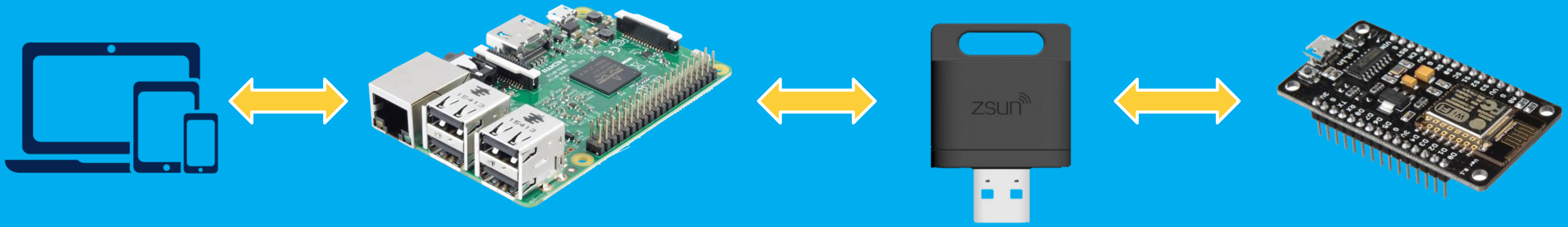
$ traceroute mqtt


Node-RED
Publish in "actuator/led_status" : "ON"
mosquitto
Publish in "actuator/led_status" : "ON"
Subscribe to "actuator/led_status"
LED ON
Publish in "sensor/humidity"
"sensor/temperature_celsius
"sensor/temperature_fahrenheit"
Subscribe to "sensor/humidity"
"sensor/temperature_celsius
"sensor/temperature_fahrenheit"
Node-RED
mosquitto
Publish in "sensor/humidity"
"sensor/temperature_celsius
"sensor/temperature_fahrenheit"
DHT22

Talk is cheap.

*Show me the **code**!*

https://github.com/jpdias/piheadquarters



it's ~~hammer~~ DEMO *time*!

it's over.

Thank you!

Q&A

João Pedro Dias
@jpd1as

Porto, Portugal

http://jpdias.me
https://keybase.io/jpdias

jpmdias@fe.up.pt

Sat. 30 Sept. 2017 @ Pixels.Camp