

Introducción a MING: Mosquitto, InfluxDB, Node-RED y Grafana

Juan Félix Mateos Barrado
juanfelixmateos@gmail.com

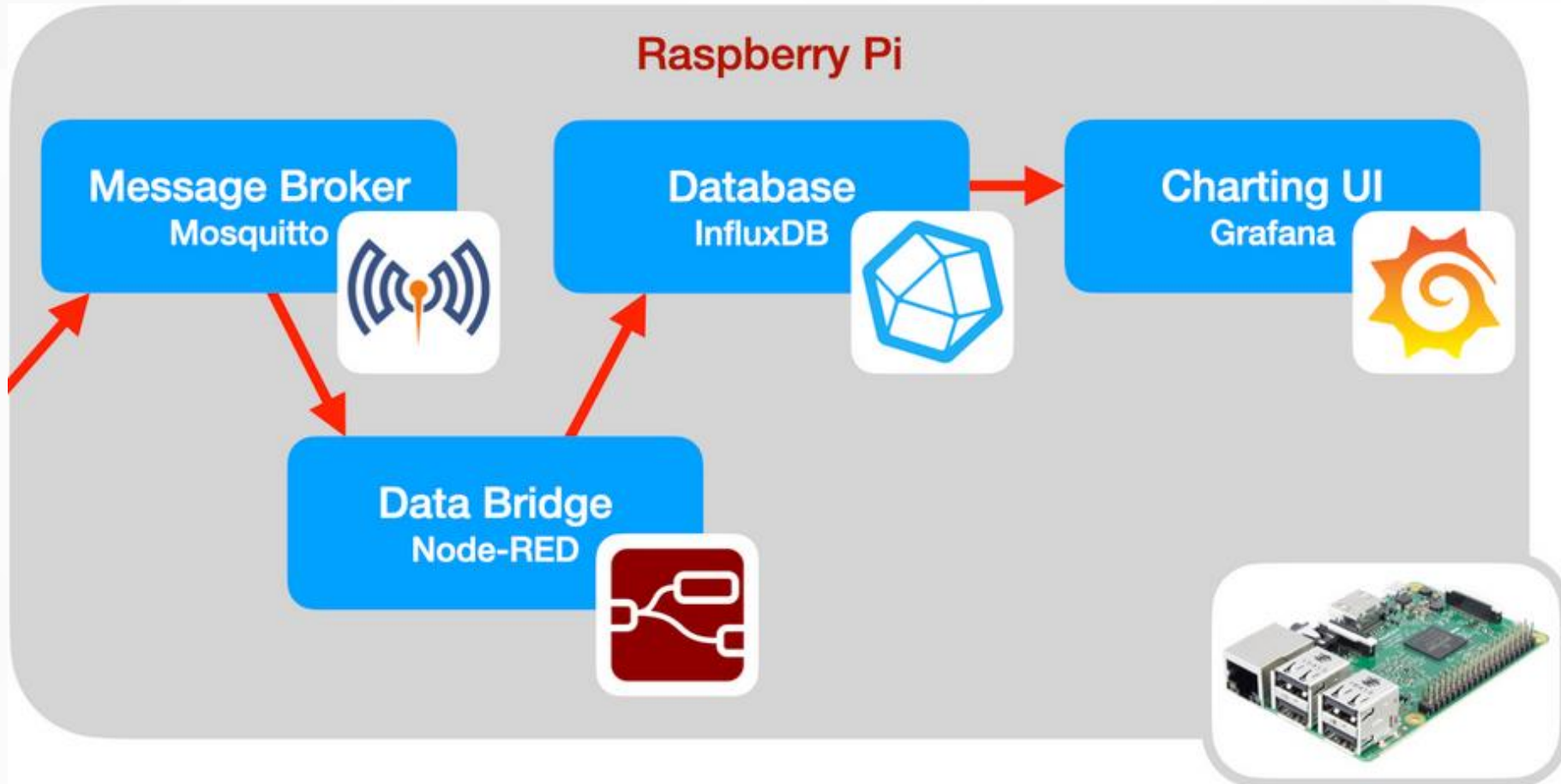
The Things Network Madrid
Junio 2022

Introducción

- Raspberry Pi es un SBC (single board computer)
- Utiliza microprocesadores ARM de la marca Broadcom
- Utiliza como SO una variante de Debian llamada Raspberry Pi OS (anteriormente Raspbian)
- Tiene un header de 40 pines que pueden actuar como entradas y salidas digitales (tolerantes a 3.3v)
- Generalmente se programa en Python
- No tiene disco duro; usa una tarjeta SD.
- Tiene un precio muy contenido (disponibilidad?)
- Puede configurarse como headless (funcionamiento sin monitor ni teclado)



MING

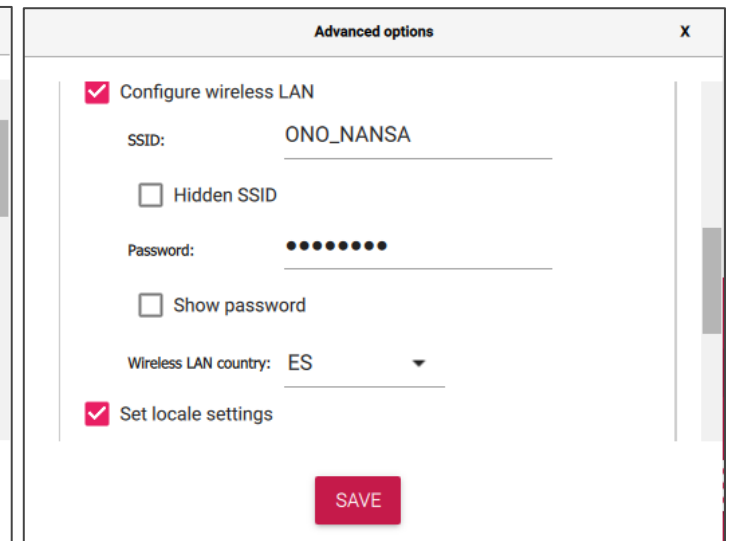
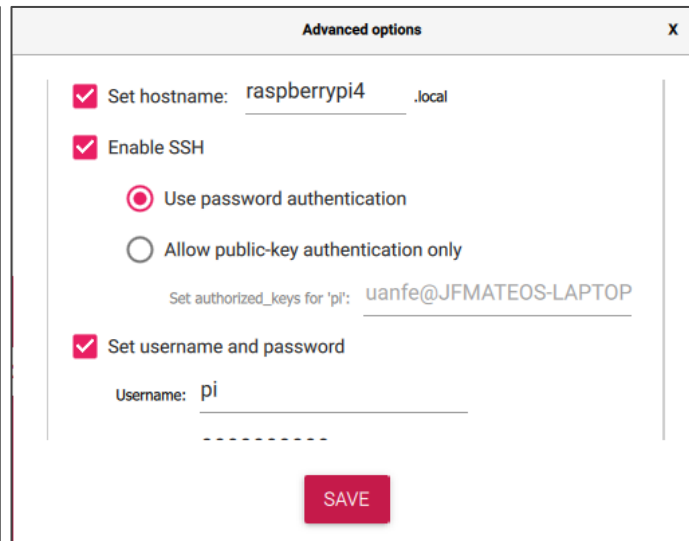
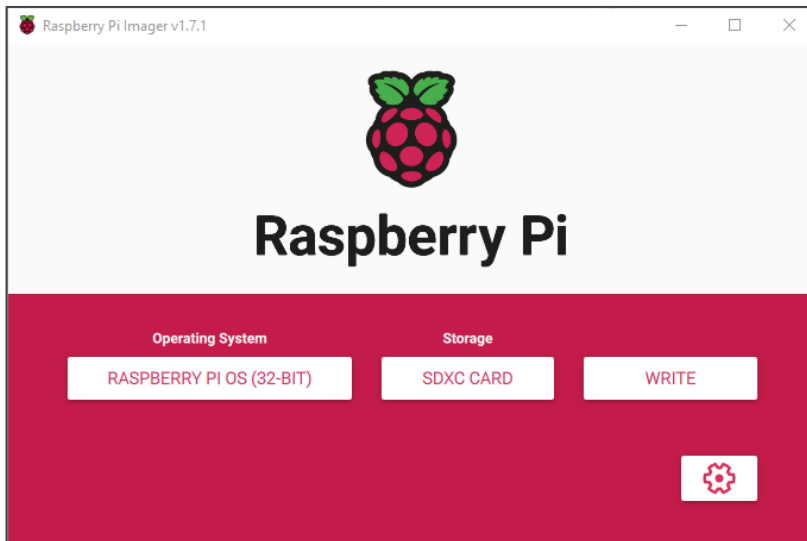


Primeros pasos

1. Crear un imagen para configurar la Raspberry Pi en modo headless
2. Acceder a la Raspberry Pi remotamente mediante ssh
3. Acceder a la Raspberry Pi remotamente mediante VNC
4. Actualización del sistema operativo
5. Introducción a Docker
6. MING con IoTStack

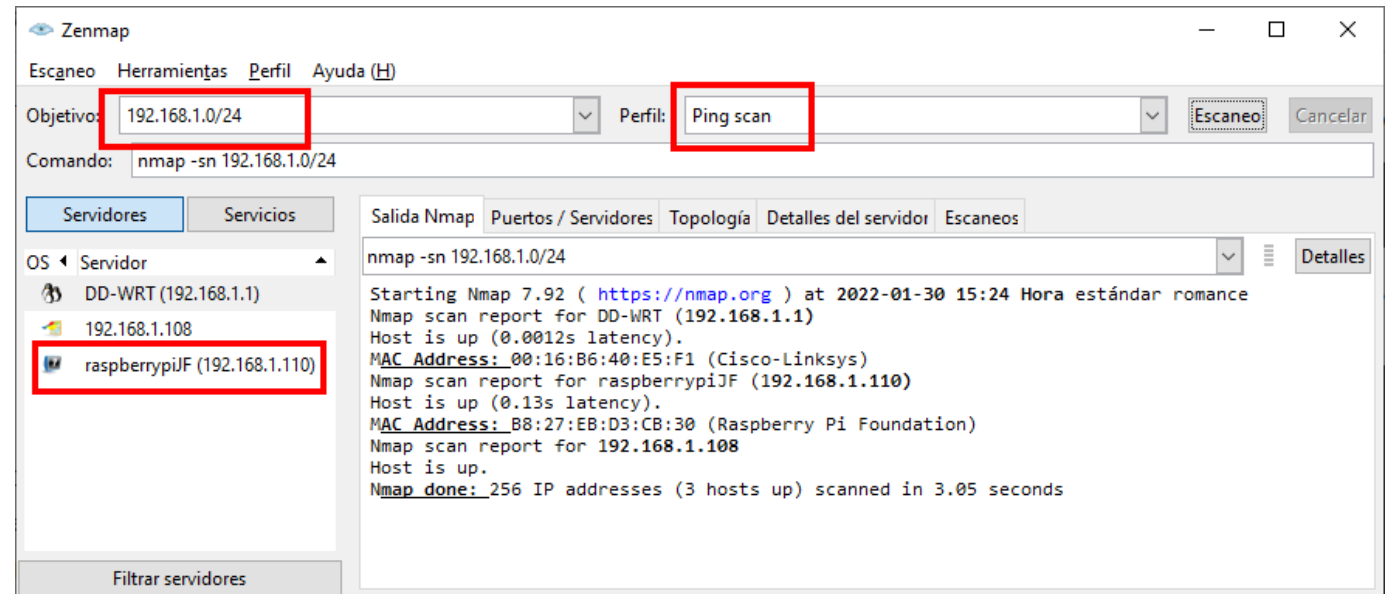
Crear un imagen para configurar la Raspberry Pi en modo headless

- Descargar e instalar Raspberry Pi Imager (el software que nos va a permitir copiar la imagen del sistema operativo en la tarjeta SD)
 - <https://www.raspberrypi.com/software/>
- Elegir el sistema operativo (vamos a elegir una versión de 32 bits, a pesar de que ya está disponible la de 64 bits porque esta última aún es un poco inmadura), y la tarjeta SD de destino.
- Pulsar Control+Mayús+X para acceder a la configuración avanzada (o usar el icono del engranaje).
 - Asignar un nombre local a la Raspberry Pi y habilitar el acceso SSH, configurar la WiFi y los locales



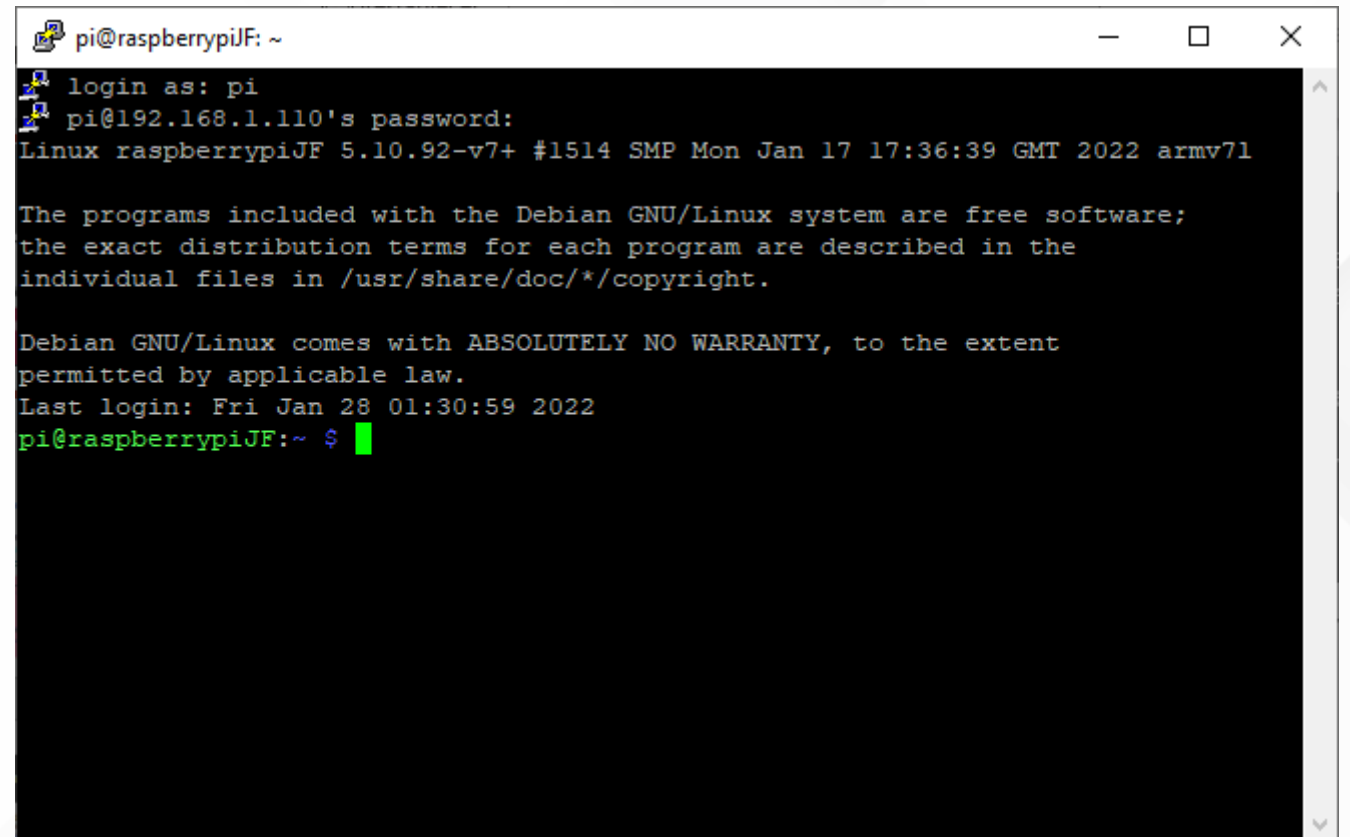
¿Y cómo obtenemos la dirección IP de la Raspberry Pi?

- Si podemos acceder al servidor DHCP, veremos directamente la dirección que ha recibido la Raspberry Pi.
- Si nuestra red tiene multicast DNS (mDNS) podemos intentar hacer un ping directamente a la dirección local
 - ping raspberrypiJF.local
- Si lo anterior no es posible, se recomienda instalar Nmap (<https://nmap.org/download.html>) y usar el comando:
 - nmap -sn 192.168.1.0/24



Acceder a la Raspberry Pi por SSH

- Instalar Putty
 - <https://www.putty.org/>



The screenshot shows a terminal window titled "pi@raspberrypiJF: ~". The text inside the terminal is as follows:

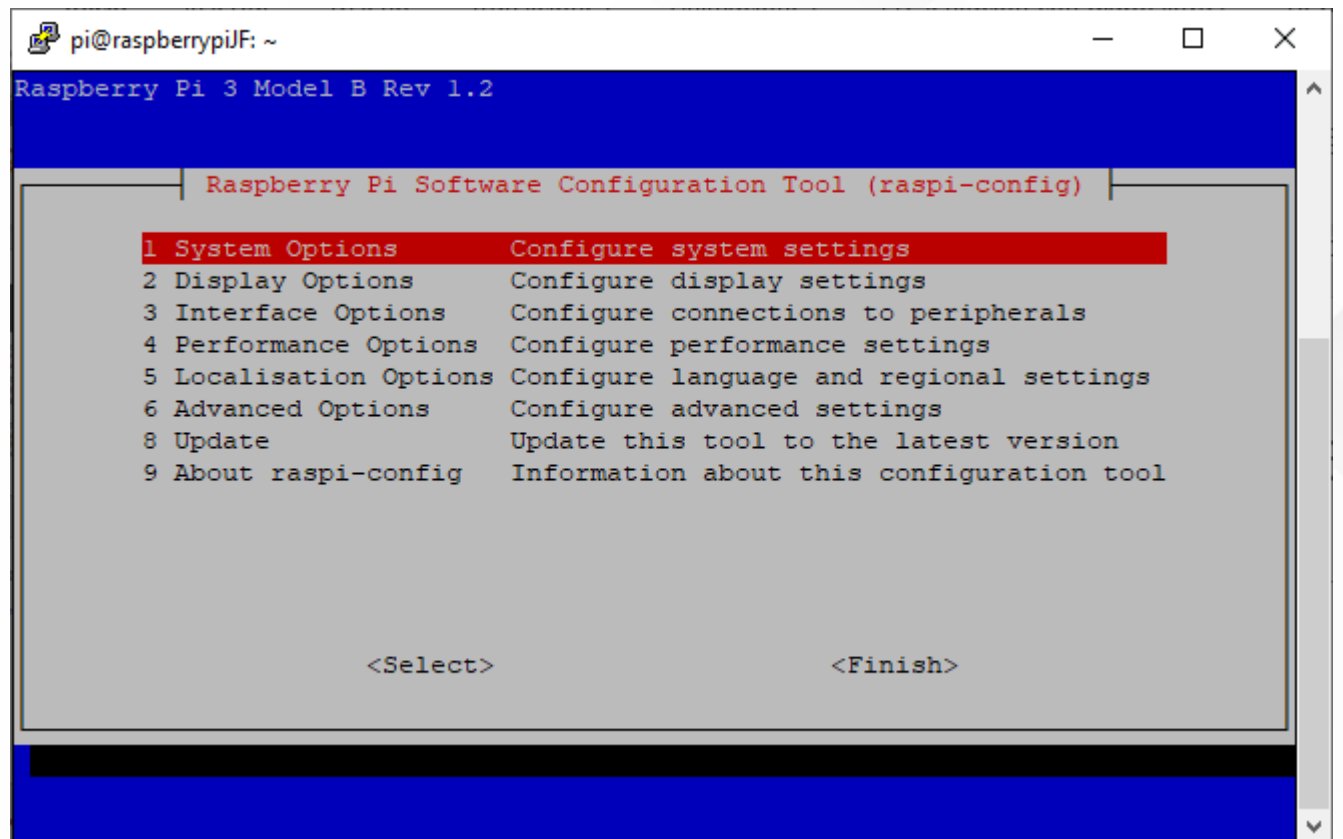
```
login as: pi
pi@192.168.1.110's password:
Linux raspberrypiJF 5.10.92-v7+ #1514 SMP Mon Jan 17 17:36:39 GMT 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jan 28 01:30:59 2022
pi@raspberrypiJF:~ $
```

Configuración de la Raspberry Pi

- `sudo raspi-config`
 - Localisation Options
 - Interface Options
 - Enable VNC



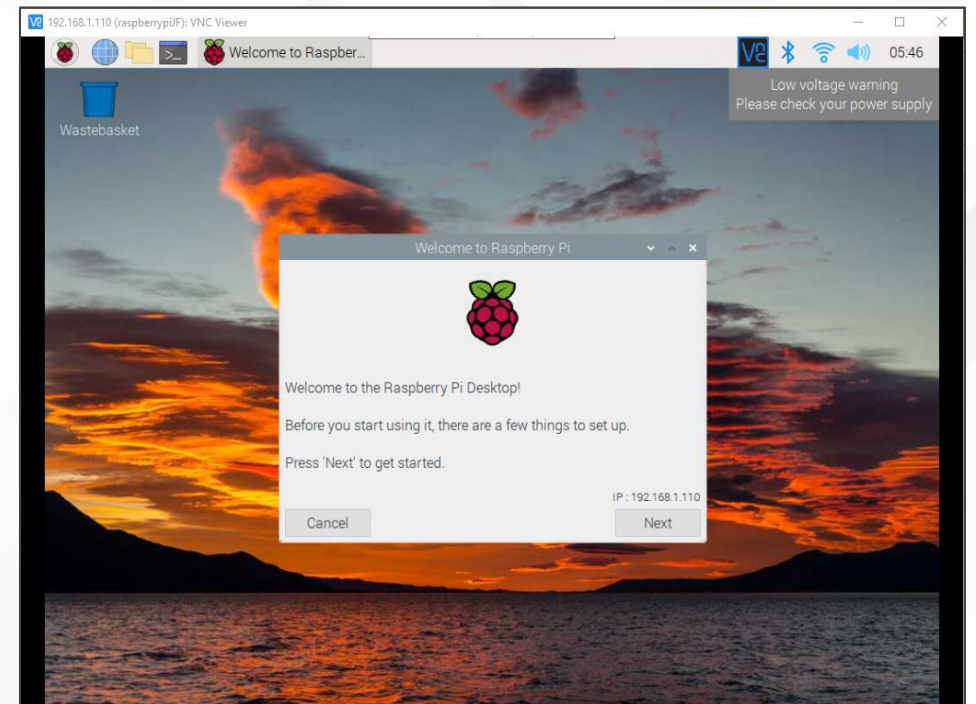
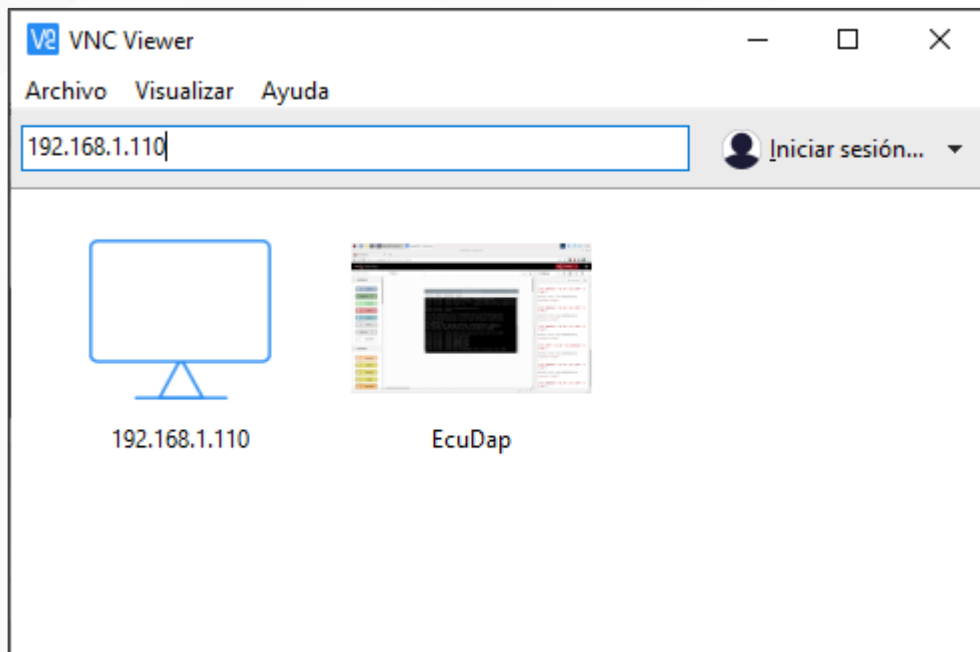
The screenshot shows a terminal window titled "pi@raspberrypiJF: ~" with the command prompt "Raspberry Pi 3 Model B Rev 1.2". The main content is the "Raspberry Pi Software Configuration Tool (raspi-config)" menu. The menu is displayed in a table-like format with a red header row. The options are numbered 1 through 9. At the bottom of the menu, there are two options: "<Select>" and "<Finish>".

1	System Options	Configure system settings
2	Display Options	Configure display settings
3	Interface Options	Configure connections to peripherals
4	Performance Options	Configure performance settings
5	Localisation Options	Configure language and regional settings
6	Advanced Options	Configure advanced settings
8	Update	Update this tool to the latest version
9	About raspi-config	Information about this configuration tool

<Select> <Finish>

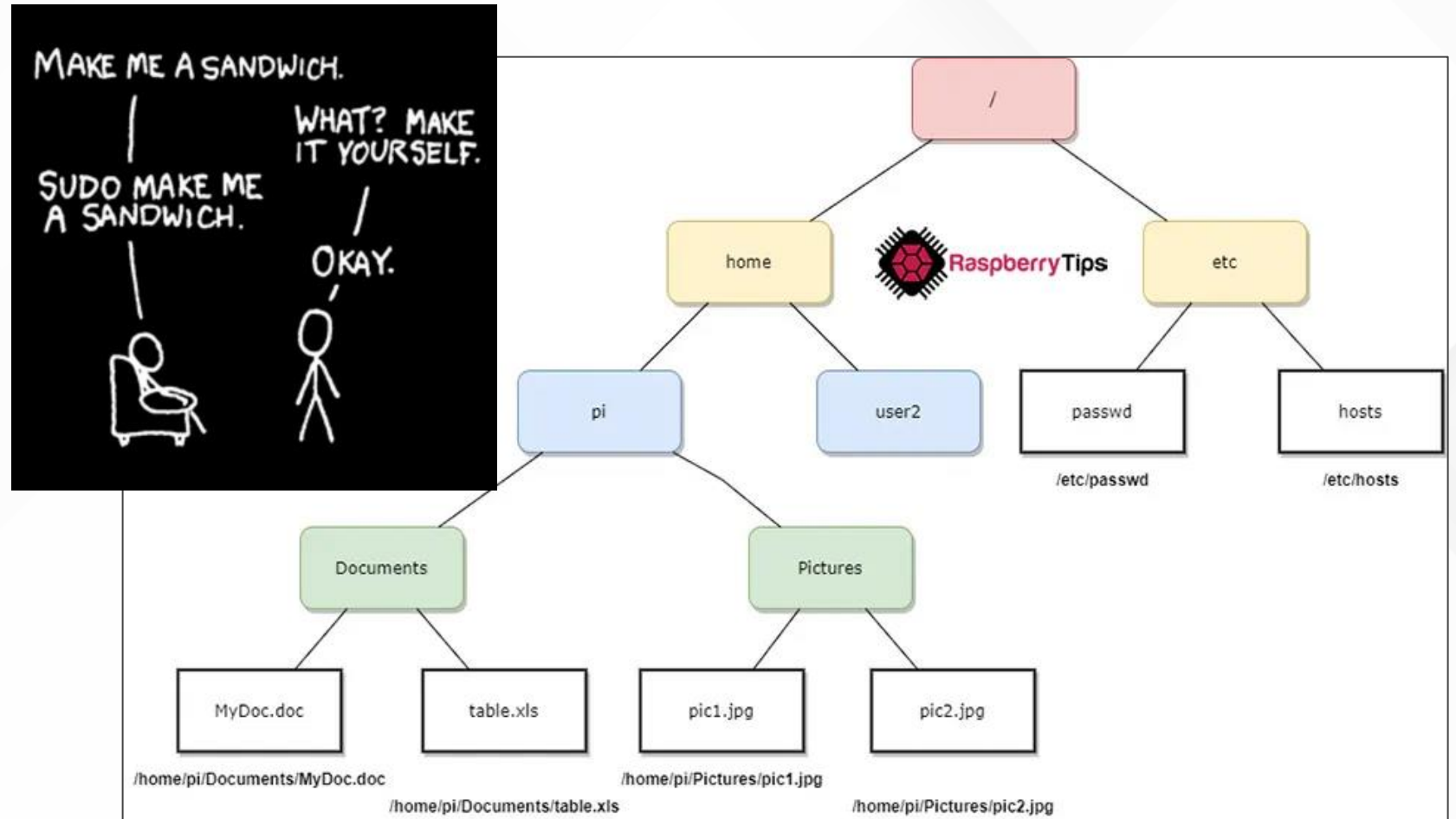
Acceder a la Raspberry Pi por VNC

- Instalar VNC Viewer
 - <https://www.realvnc.com/es/connect/download/viewer/>



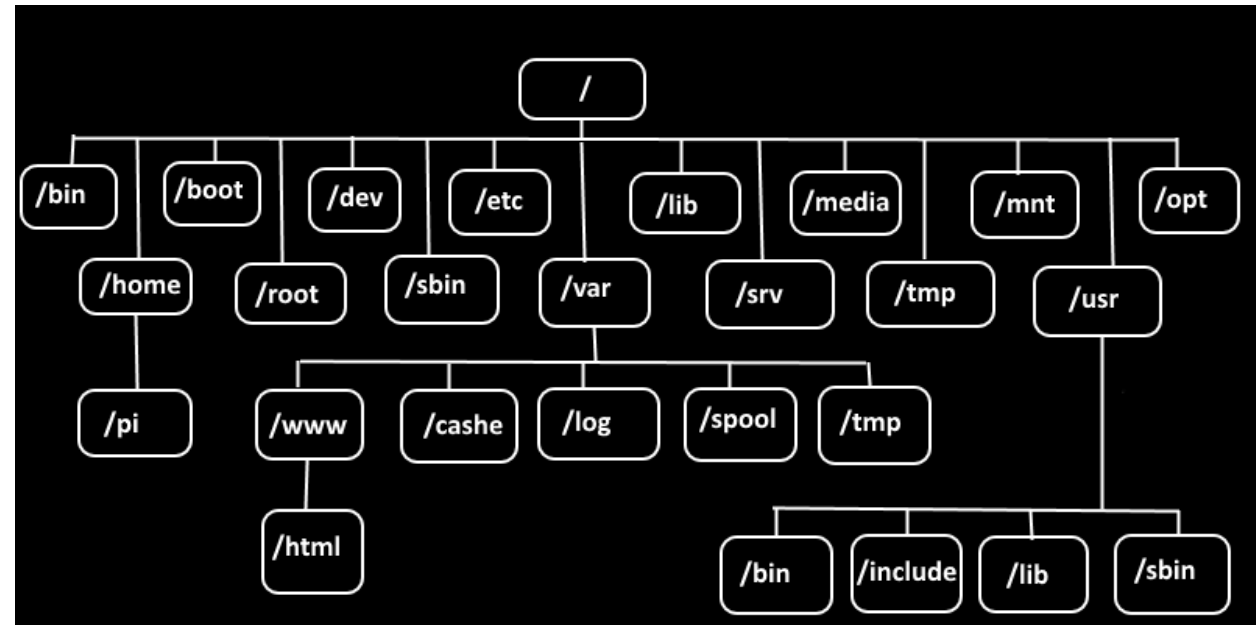
Algunos comandos básicos

- sudo shutdown -h now
- sudo reboot
- sudo poweroff
- sudo su
- ls -la (list **a**ll files)
- cd .. (subir un nivel)
 - cd: ir al directorio home
- cp: copiar un archivo
- mv: mover un archivo
- rm: elimina un archivo
 - rm -r: elimina una carpeta
- mkdir: crea una carpeta
- cat: visualiza un archivo
- nano: editor de texto
- pwd: print working directory



Los directorios principales

- **root (/)**
- **home:** Contiene un directorio para cada usuario, en el que se encuentran los archivos de ese usuario
 - Videos, Pictures, Desktop...
- **etc** (Editable Text Configuration): Contiene los archivos de configuración
 - sudoers, crontab, apache, php...
- **var:** Contiene los archivos que se modifican durante la ejecución del sistema
 - log, www, mail...
- **usr** y **bin:** Contienen archivos de los programas, que no debemos tocar



El sistema de permisos de Linux

```
$ ls -l samplefile
```

1	2	3	4	5	6	7
-rw-r--r--	1	pi	pi	5	Nov 29 06:09	samplefile

Diagram illustrating the components of the `ls -l` output for the file `samplefile`:

- File Type:** Indicated by the first character of the permissions (dash for regular file).
- Owner Permissions:** The next three characters (r, w, r) represent permissions for the file owner.
- Group Permissions:** The next three characters (--, r, --) represent permissions for the group.
- Other Permissions:** The final three characters (r, --, --) represent permissions for others.
- Number of Hard Links:** The number '1' indicates the file has one hard link.
- User Owner:** The string 'pi' indicates the file is owned by the user 'pi'.
- Group Owner:** The string 'pi' indicates the file belongs to the group 'pi'.
- File Size in Bytes:** The number '5' indicates the file size is 5 bytes.
- Modification Date:** The string 'Nov 29 06:09' indicates the last modification date and time.
- File Name:** The string 'samplefile' is the name of the file.

- Cada archivo/directorio tiene un **usuario propietario** y un **grupo propietario**
- Cada usuario puede pertenecer a varios grupos
 - **groups pi:** Muestra los grupos a los que pertenece el usuario pi
 - **usermod** permite administrar los grupos
- Para cada archivo/directorio se definen permisos de lectura, escritura y ejecución a nivel de:
 - usuario propietario
 - grupo propietario
 - otros
- `chmod [modo] archivo:` Sirve para cambiar los permisos de un archivo/directorio

chmod

chmod [modo] archivo

- [modo] son 3 caracteres
 - El primer carácter indica si queremos modificar los permisos para
 - u: El usuario propietario
 - g: El grupo propietario
 - o: Otros usuarios
 - a: Todos
 - El segundo carácter indica si queremos añadir un rescindir un permiso
 - +
 - -
 - El tercer carácter indica el permiso
 - r: Lectura
 - w: Escritura
 - x: Ejecución para archivos
 - X: Ejecución para carpetas

Ejercicio

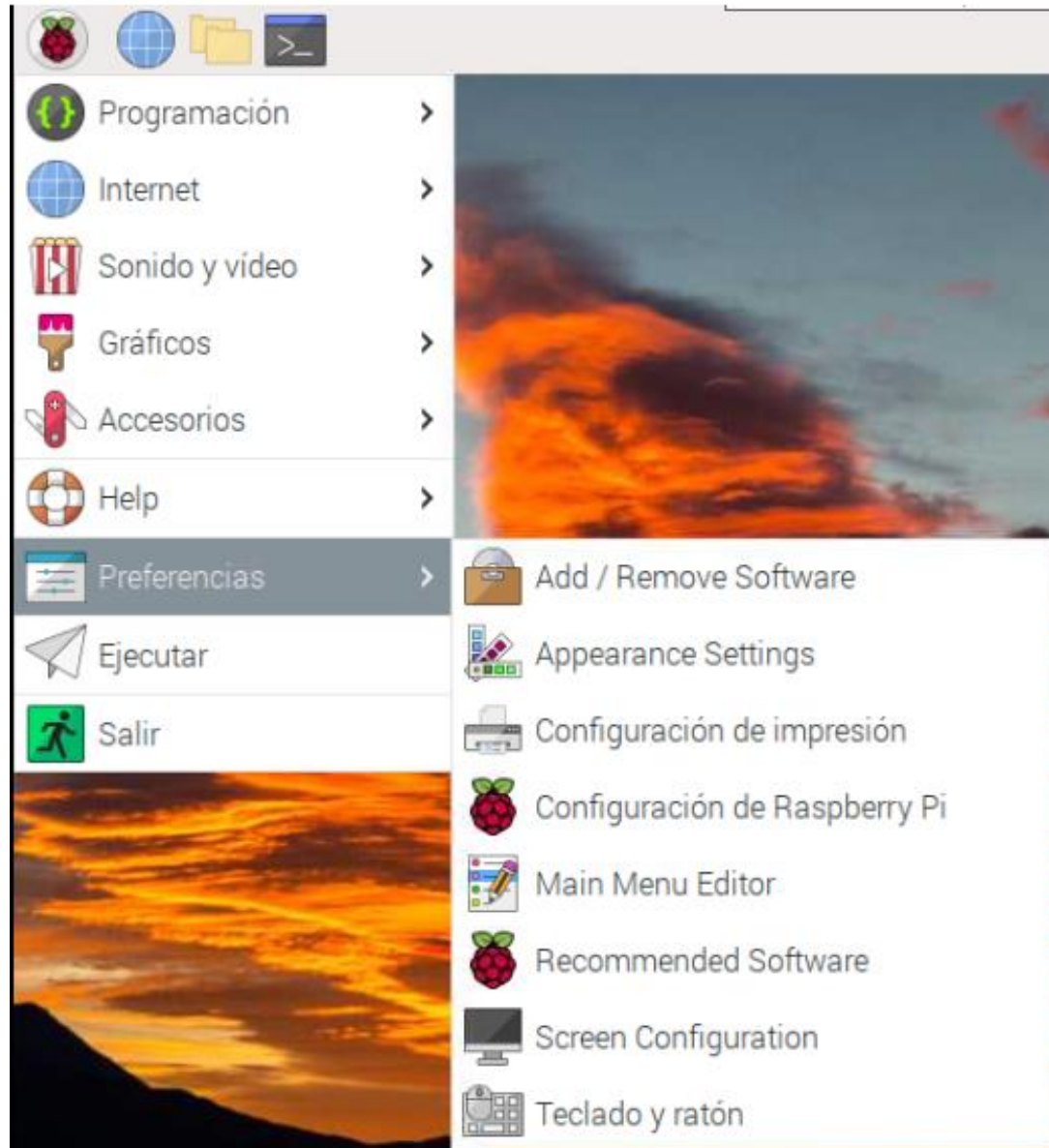
- Crear un directorio llamado **miDirectorio** y un archivo llamado **miArchivo.txt** dentro del directorio `/home/pi/Documents`
 - `mkdir`
 - `nano`
- Comprobar los permisos del archivo **miArchivo**
 - `ls -la`
- Añadir al grupo propietario el permiso de escritura
 - `chmod`
- Mover el archivo **miArchivo** al interior de **miDirectorio**
 - `mv`
- Borrar el archivo y el directorio
 - `rm -r`

Actualización del sistema operativo

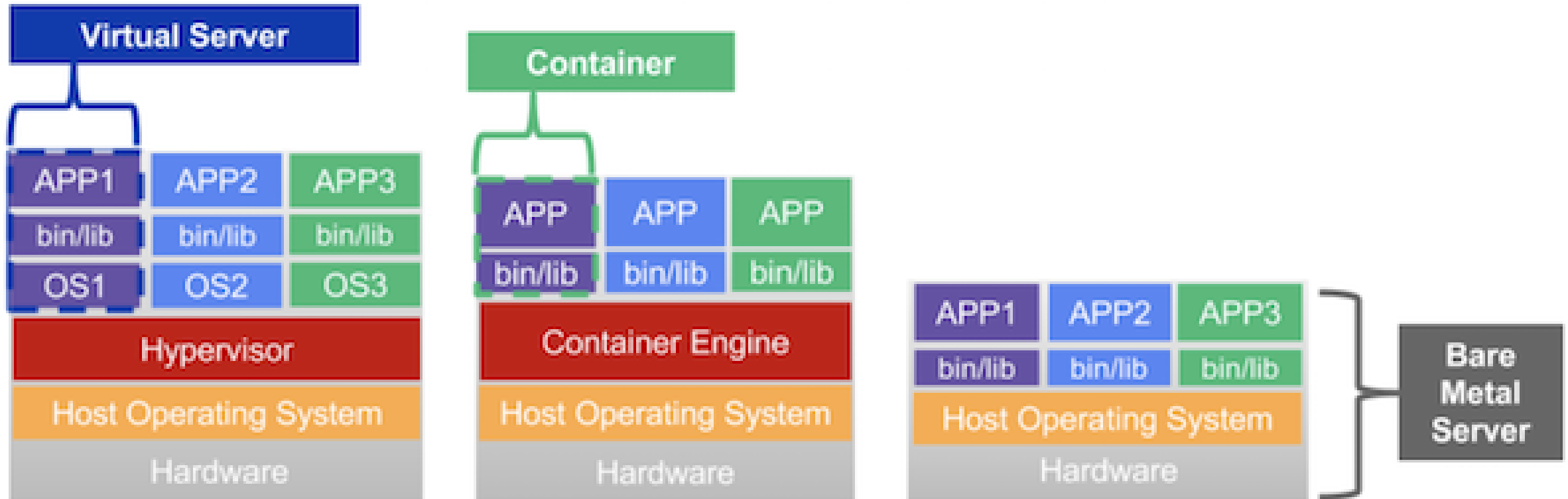
- `sudo apt update`
 - Actualiza los orígenes del software a las últimas versiones en `/etc/apt/sources.list` pero no instala nada
- `sudo apt full-upgrade`
 - `full-upgrade` actualiza también las dependencias

Instalación de software (Bare Metal)

raspberrypi4.local (raspberrypi4): VNC Viewer



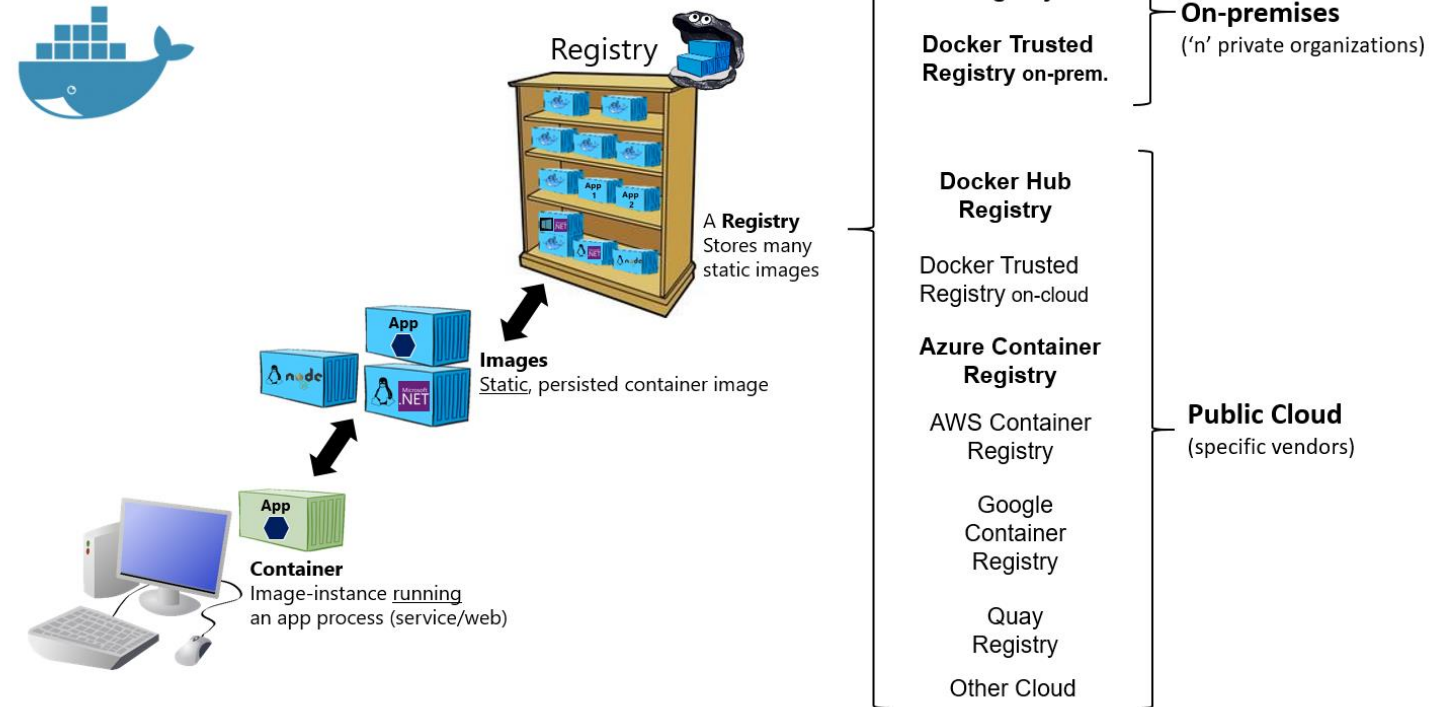
Introducción a Docker



Conceptos básicos de Docker

- **Registro** (hub.docker.com)
- **Imagen:** Es una plantilla con todo lo necesario para ejecutar una aplicación (código y dependencias)
- **Contenedor:** Un contenedor es la ejecución de una imagen.

Basic taxonomy in Docker

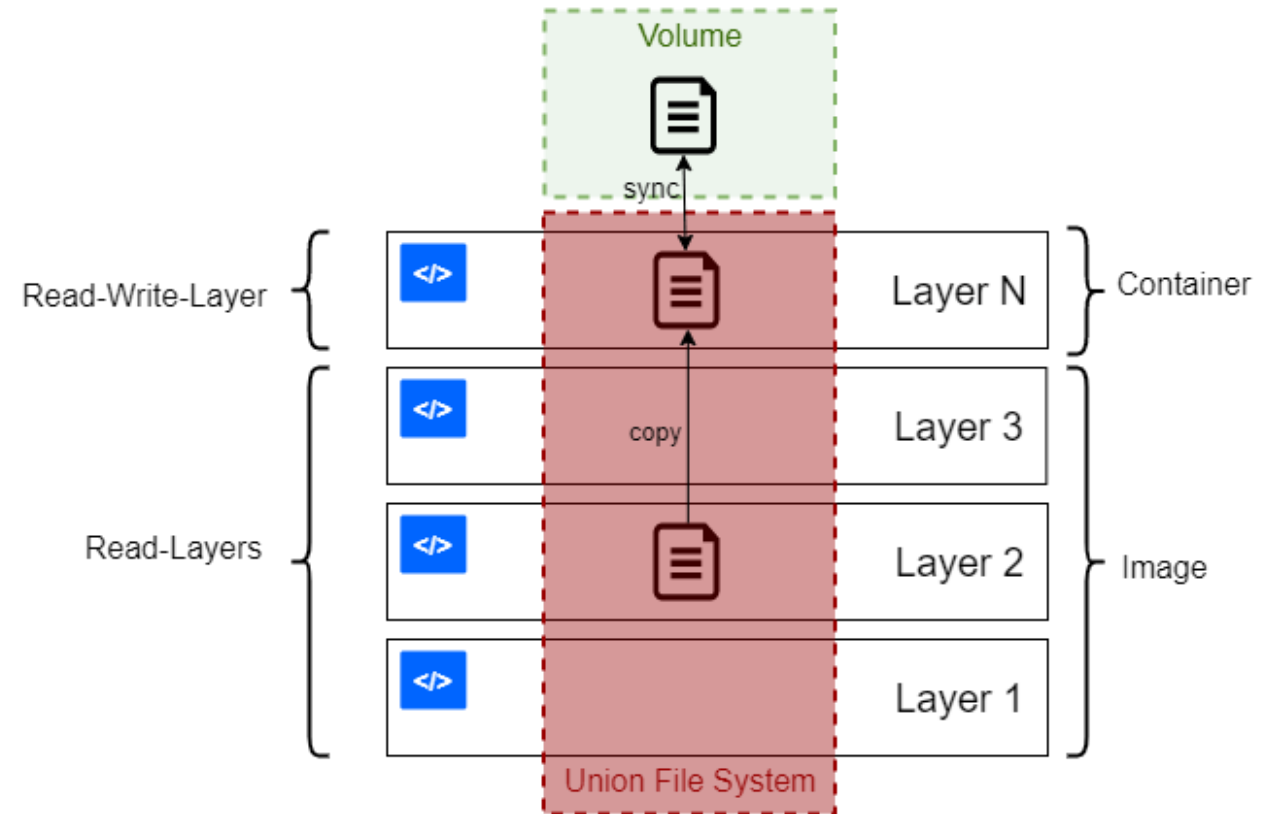


Por qué Docker

- **Footprint:** Menor tamaño que las máquinas virtuales y se inician más rápido, pero es más lento que Bare Metal.
- **Seguridad:** Los contenedores independizan unas aplicaciones de otras, evitando conflictos (por ejemplo, imaginemos que necesitamos una aplicación que depende de Python 2 y otra que depende de Python 3)
- **Flexibilidad:** Permite desplegar nuevas versiones con agilidad, incluso entre sistemas/arquitecturas diferentes
- **Principal inconveniente:** No permite instalar en un mismo servidor aplicaciones que requieren distintos sistemas operativos.

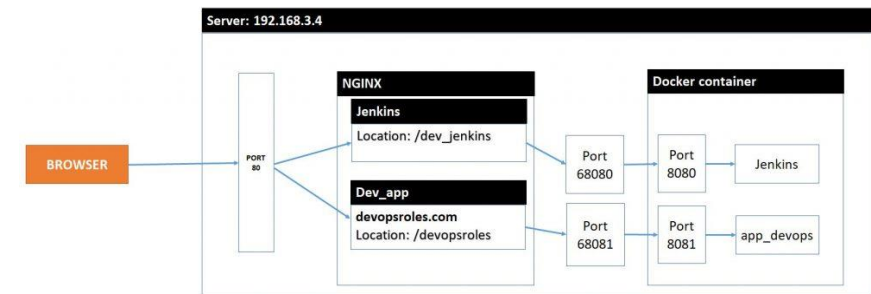
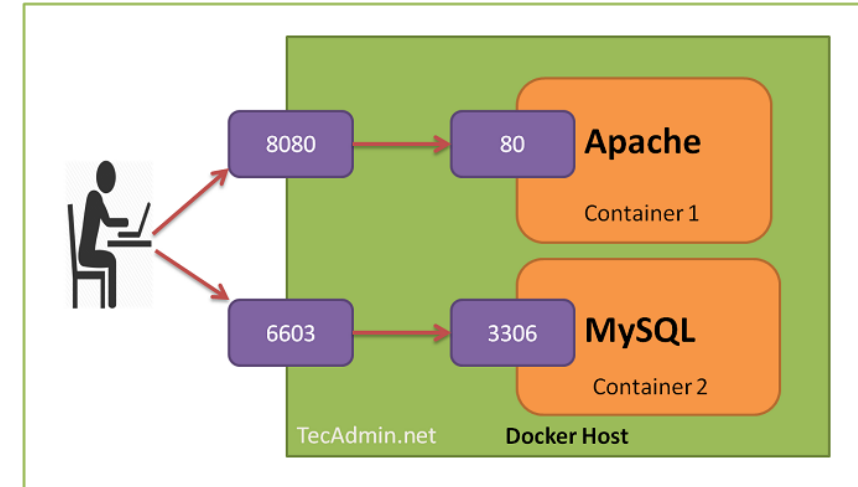
Cómo funciona Docker

- Las imágenes de Docker (los ejecutables) se almacenan como archivos de sólo lectura.
- Cuando ejecutamos un contenedor, Docker crea una capa de archivos de lectura/escritura sobre la imagen.
- Si la ejecución de la imagen requiere modificar un archivo, ese archivo se copia de la capa de sólo lectura a la capa de lectura/escritura, y las modificaciones se realizan en este archivo copiado.
- Pero cuando la imagen se detiene, estas modificaciones se pierden 😞
- Para evitarlo utilizamos **Volúmenes**, que trasladan los archivos modificados del contenedor al sistema de archivos del Host.



Comunicación de los contenedores con el exterior

- La mayoría de las aplicaciones que utilizan contenedores (nodeRED, OpenHAB, mosquitto...) disponen de interfaces web a las que se accede mediante un puerto HTTP.
- Existen 2 formas de que un contenedor sea accesible desde el exterior:
 - **Binding:** Consiste en vincular el puerto de la aplicación al puerto real del sistema host. El principal inconveniente de este sistema es que ese puerto ya sólo podrá ser utilizado por ese contenedor.
 - **Reverse Proxy:** Instalar un proxy inverso, como Nginx, que redirigirá las peticiones del host al puerto correspondiente de los contenedores basándose en el dominio o ruta de la solicitud.



Instalación de Docker

- `curl -sSL https://get.docker.com | sh`
 - Ejecutar con sh (la consola) el contenido de la dirección `https://get.docker.com` transferidos con cifrado de datos
- Comprobar la instalación con el comando **docker**

```
pi@raspberrypiJF:/etc/apt $ docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
      --config string      Location of client config files (default
                           "/home/pi/.docker")
  -C, --context string     Name of the context to use to connect to the
                           daemon (overrides DOCKER_HOST env var and
                           default context set with "docker context use")
  -D, --debug              Enable debug mode
  -H, --host list          Daemon socket(s) to connect to
```

Añadir el usuario pi al grupo docker

Vamos a añadir el usuario **pi** al grupo **docker** para poder ejecutar comandos docker sin problemas de permisos

```
sudo usermod -aG docker pi
```

-aG indica "append group", es decir, añadir el grupo docker al conjunto de grupos a los que pertenece el usuario pi.

newgrp docker → PARA QUE EL CAMBIO TENGA EFECTO

```
pi@raspberrypi:~/Documents $ sudo usermod -aG docker pi
pi@raspberrypi:~/Documents $ groups pi
pi : pi adm dialout cdrom sudo audio video plugdev games users input render netd
ev spi i2c gpio lpadmin docker
pi@raspberrypi:~/Documents $ newgrp docker
```

Instalación de docker-compose

- docker-compose se instala a través de pip3

```
sudo apt install -y python3 python3-dev python3-pip libffi-dev libssl-dev
sudo pip3 install docker-compose
sudo systemctl enable docker
```
- Comprobamos que se ha instalado correctamente con

```
docker-compose --version
```

```
pi@raspberrypi:~/Documents $ docker-compose --version
docker-compose version 1.29.2, build unknown
```


Operaciones básicas con contenedores

- `docker ps -a`: Muestra todos los contenedores y su estado
- `docker stop`: Detiene el contenedor
 - Si no responde, podemos usar `docker kill`
- `docker start`: Inicia un contenedor que se ha detenido
- `docker restart`: Reinicia un contenedor que se está ejecutando
- `docker rm`: Borra un contenedor. Para poder borrar un contenedor debe estar detenido.
- `docker rmi`: Borra una imagen

FILED	BOILERPLATE\BOILERPLATE-C6:J9F6A6	"\BOILERPLATE"	4	WYNNER	8000\LCB'	3443\LCB'	0'0'0'0:2000->2000\LCB'	:::2000->2000\LCB'	BOILERPLATE
VIEW ID	IMAGE	COMMAND	CHECKED	STATUS	BOOKS				NAME
@tsabreilab\jle:\dev\jpr\goccker\olei\j9\3# goccker ba -s									

Docker Compose

- Es un orquestador de contenedores, es decir, una herramienta que nos permite configurar aplicaciones en las que intervienen múltiples contenedores y que se ejecutan en un **entorno independiente**.
 - Por ejemplo: Una aplicación Joomla! puede estar compuesta por 3 contenedores
 - El CMS Joomla
 - La base de datos MySQL
 - PHPmyAdmin para gestionar la base de datos
- La aplicación se configura en un archivo llamado `docker-compose.yml`

Ejemplo de un archivo docker-compose.yml

version: '2'

services:

joomla:

image: joomla

restart: always

ports:

- 8080:80

environment:

JOOMLA_DB_HOST: **joomladb**

JOOMLA_DB_PASSWORD: **example**

joomladb:

image: mysql:5.6

restart: always

environment:

MYSQL_ROOT_PASSWORD: **example**

phpmyadmin:

image: phpmyadmin/phpmyadmin

container_name: phpmyadmin

environment:

- PMA_ARBITRARY=1

restart: always

ports:

- 8081:80

volumes:

- /sessions

IoTstack: Instalación 1/3

- Asegurarse de que está instalado curl
 - `sudo apt install -y curl`
- Descargar el script de instalación de IoTstack
 - `curl -fsSL https://raw.githubusercontent.com/Sensorslot/IOTstack/master/install.sh | bash`
- Cambiar al directorio de IoTstack
 - `cd ~/IoTstack`
- Ejecutar el script de generación del docker-compose (agrandar la ventana de Putty si fuera necesario)
 - `sudo ./menu.sh`
 - sudo es necesario porque actualmente hay un error en el script del menú
 - <https://github.com/Sensorslot/IOTstack/issues/222>

IoTstack: Instalación 2/3

- Instalar las dependencias requeridas
- En Native Install -> Upgrade Docker and Docker-Compose
 - Reiniciar
 - `cd ~/IoTstack`
 - `sudo ./menu.sh`
- Build Stack
 - Grafana
 - InfluxDB
 - Mosquitto
 - NodeRED
 - Añadir los addons predeterminados NodeRED

The image displays three terminal windows illustrating the IoTstack installation process. The top window, titled 'Python 3 and Dependencies', shows a prompt asking to install Python 3.6.9 or later (current is 3.9.2), ruamel.yaml 0.16.12 or later, and pip3. The bottom-left window shows the 'menu.sh' script with options: Build Stack (highlighted), Docker Commands, Miscellaneous Commands, Backup and Restore, Native Installs, and Exit. The bottom-right window shows the 'Select NodeRed Addons' screen with a list of addons, where 'node-red-configurable-ping' is highlighted and selected with an 'X'.

```
Python 3 and Dependencies

Python 3.6.9 or later (Current = 3.9.2), ruamel.yaml 0.16.12 or later
(Current = not.not.not), blessed and pip3 are required for IOTstack to
function correctly. Install these now?

<S> <No>

-> Build Stack <-
  Docker Commands
  Miscellaneous Commands
  Backup and Restore
  Native Installs
  Exit

Select NodeRed Addons (npm) to install on initial run

(X) -> node-red-configurable-ping <-
(X)   node-red-contrib-boolean-logic
(X)   node-red-contrib-influxdb
(X)   node-red-dashboard
(X)   node-red-node-pi-gpiod
(X)   node-red-node-rbe
( )   node-red-contrib-alexa-local
( )   node-red-contrib-bert-tokenizer
( )   node-red-contrib-bigtimer
( )   node-red-contrib-blynk-ws
```

IoTstack: Instalación 3/3

- Docker Commands
→ Start stack

```

Select Docker Command to run

-> Start stack <-
Restart stack
Stop stack
Monitor Logs
Stop ALL running docker containers
Update all containers (may take a long time)
Delete all stopped containers and docker volumes (prune volumes)
Delete all images not associated with container
Back

Controls:
[Up] and [Down] to move selection cursor
[H] Show/hide this text
[Enter] to run command
[Escape] to go back to main menu

Start Stack:
docker-compose up -d --remove-orphans
WARNING: Some networks were defined but are not used by any service: nextcloud
Creating network "iotstack_default" with driver "bridge"
Pulling grafana (grafana/grafana)...
latest: Pulling from grafana/grafana
57fb4b5fla47: Pull complete
eecldeac55e4: Pull complete
9d627d22c017: Pull complete
44402a81b422: Pull complete
17d2ae62c7ac: Pull complete
9ac7996e04d3: Downloading [=====] 28.44MB/75.85MB
8595927c6fb8: Download complete
80551ad5d37f: Download complete
926e1715lee8: Download complete
```

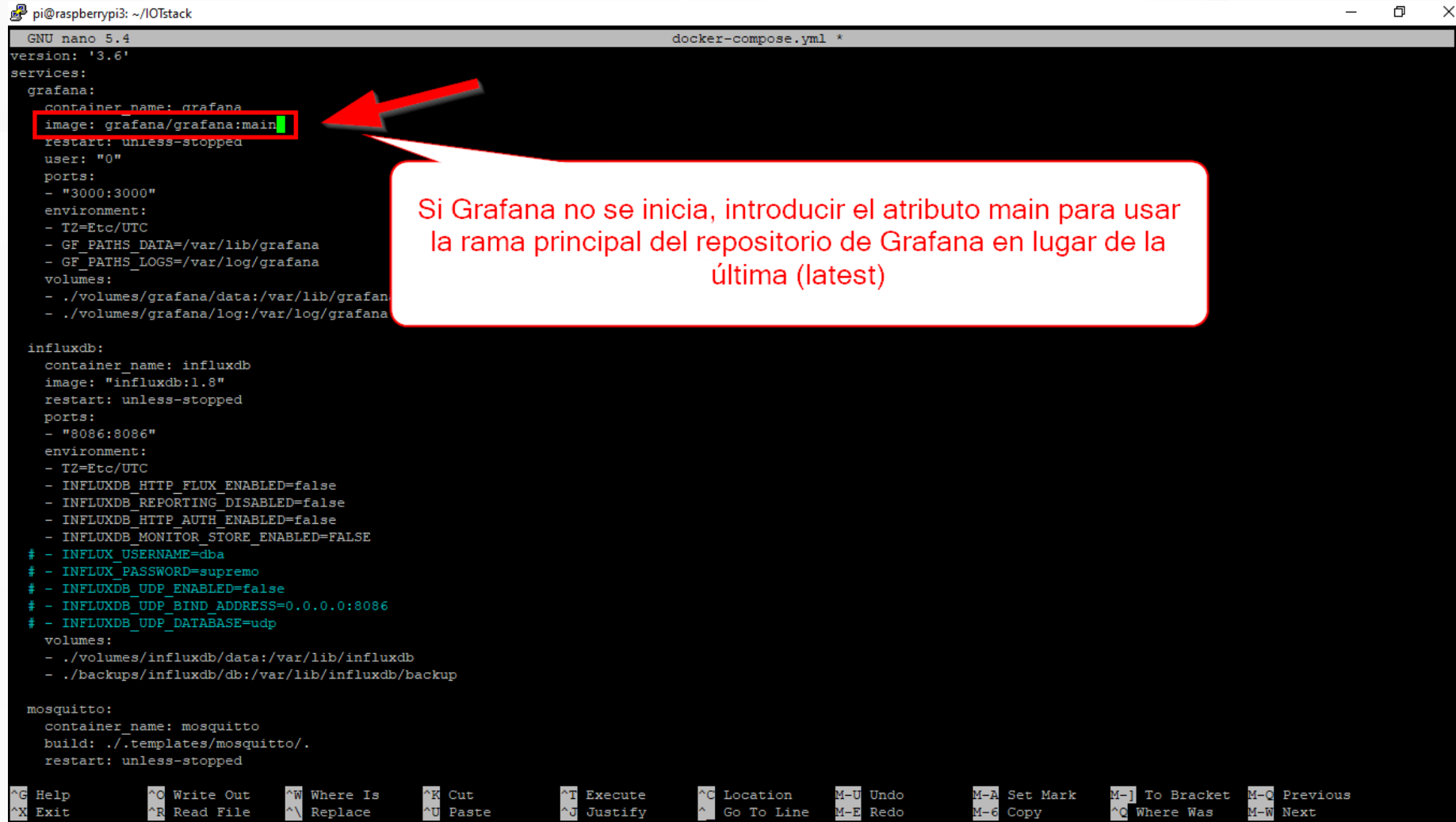
docker ps -a

- Este comando nos mostrará el estado de los contenedores

```
pi@raspberrypi3:~/IOTstack $ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d602a81aec56	grafana/grafana:main	"/run.sh"	2 minutes ago	Up 2 minutes	0.0.0.0:3000->3000/tcp, :::3000->3000/tcp	grafana
030d62e53aa7	iotstack_nodered	"npm --no-update-not..."	6 minutes ago	Up 6 minutes (healthy)	0.0.0.0:1880->1880/tcp, :::1880->1880/tcp	nodered
00fa7f6d2109	iotstack_mosquitto	"/docker-entrypoint...."	6 minutes ago	Up 6 minutes (healthy)	0.0.0.0:1883->1883/tcp, :::1883->1883/tcp	mosquitto
1217a4d344e7	influxdb:1.8	"/entrypoint.sh infl..."	6 minutes ago	Up 6 minutes	0.0.0.0:8086->8086/tcp, :::8086->8086/tcp	influxdb

El archivo docker-compose.yml



```
pi@raspberrypi3: ~/IoTstack
GNU nano 5.4 docker-compose.yml *
version: '3.6'
services:
  grafana:
    container_name: grafana
    image: grafana/grafana:main
    restart: unless-stopped
    user: "0"
    ports:
      - "3000:3000"
    environment:
      - TZ=Etc/UTC
      - GF_PATHS_DATA=/var/lib/grafana
      - GF_PATHS_LOGS=/var/log/grafana
    volumes:
      - ./volumes/grafana/data:/var/lib/grafana
      - ./volumes/grafana/log:/var/log/grafana

  influxdb:
    container_name: influxdb
    image: "influxdb:1.8"
    restart: unless-stopped
    ports:
      - "8086:8086"
    environment:
      - TZ=Etc/UTC
      - INFLUXDB_HTTP_FLUX_ENABLED=false
      - INFLUXDB_REPORTING_DISABLED=false
      - INFLUXDB_HTTP_AUTH_ENABLED=false
      - INFLUXDB_MONITOR_STORE_ENABLED=FALSE
      # - INFLUX_USERNAME=dba
      # - INFLUX_PASSWORD=supremo
      # - INFLUXDB_UDP_ENABLED=false
      # - INFLUXDB_UDP_BIND_ADDRESS=0.0.0.0:8086
      # - INFLUXDB_UDP_DATABASE=udp
    volumes:
      - ./volumes/influxdb/data:/var/lib/influxdb
      - ./backups/influxdb/db:/var/lib/influxdb/backup

  mosquitto:
    container_name: mosquitto
    build: ../templates/mosquitto/
    restart: unless-stopped
```

Si Grafana no se inicia, introducir el atributo main para usar la rama principal del repositorio de Grafana en lugar de la última (latest)

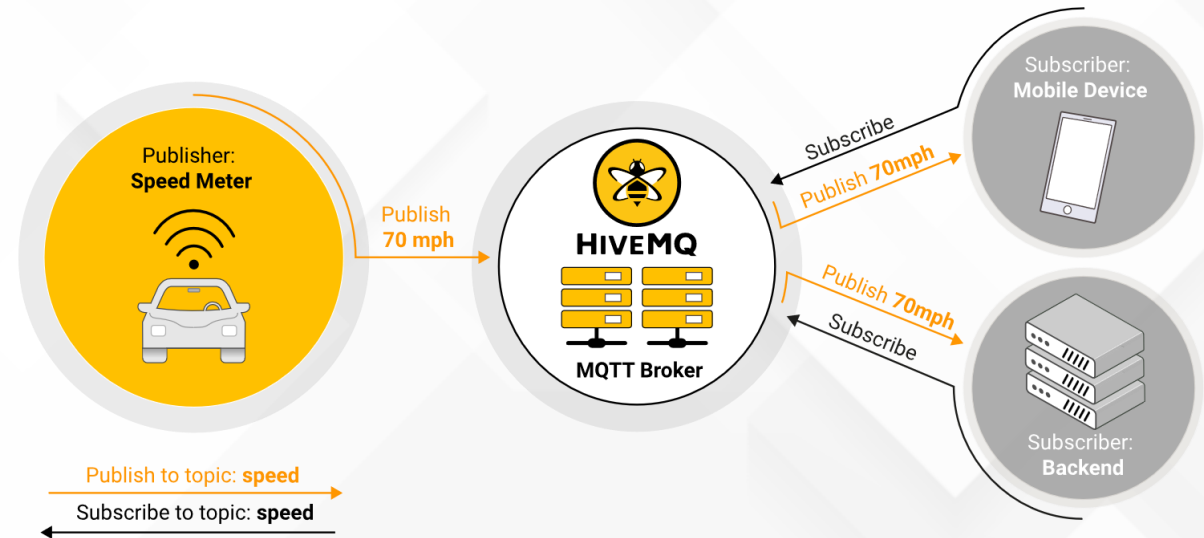
Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark M-J To Bracket M-Q Previous
Exit Read File Replace Paste Justify Go To Line M-E Redo M-C Copy ^C Where Was M-W Next

Configuración predeterminada de IoTstak

- Mosquitto
 - Puerto: 1883
- InfluxDB
 - Puerto: 8086
- Grafana
 - Puerto: 3000
 - Al acceder por primera vez usar admin/admin; nos obligará a cambiar el password
- Node-RED
 - Puerto: 1880

MQTT

- Es un protocolo:
 - abierto
 - de transporte de mensajes
 - entre clientes y servidores
 - mediante la técnica de publicación/suscripción
- Los servidores se denominan Brokers
- Ventajas para IoT/M2M
 - Requiere poca carga de procesamiento (existen clientes para prácticamente cualquier tipo de dispositivo, incluidos microcontroladores).
 - Consume muy poco ancho de banda (es muy compacto)

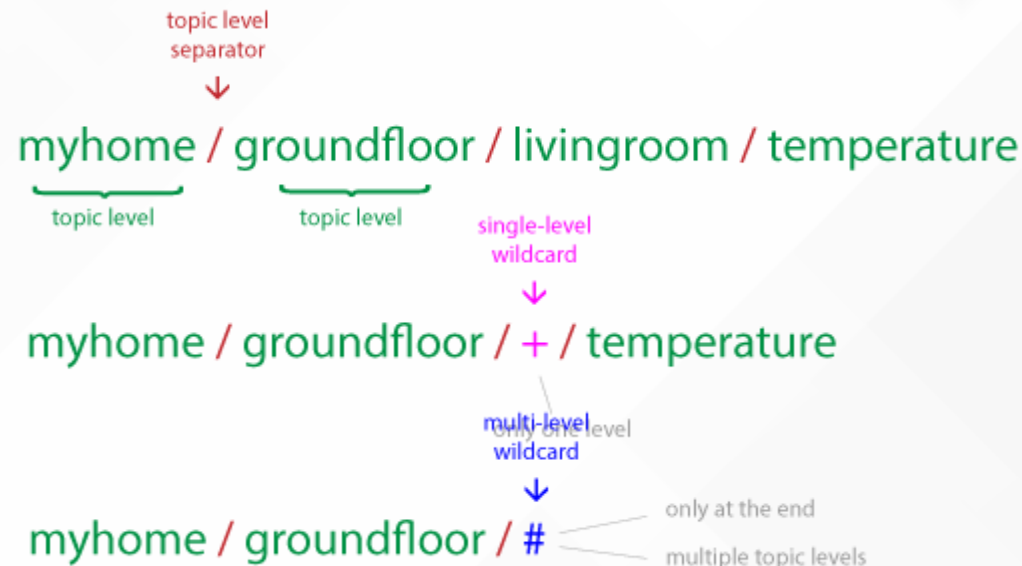


MQTT: Desacoplamiento

- Los dispositivos no se comunican directamente entre sí, sino a través del bróker
 - **Espacio:** Los dispositivos no tienen que intercambiar sus direcciones IP; sólo tienen que conocer la del bróker
 - **Tiempo:** Los dispositivos no tienen que estar activos a la vez
 - **Asíncrono:** Es un protocolo tan ligero que pueden realizarse las operaciones de publicación y suscripción en segundo plano, sin interrumpir las operaciones principales de los dispositivos.

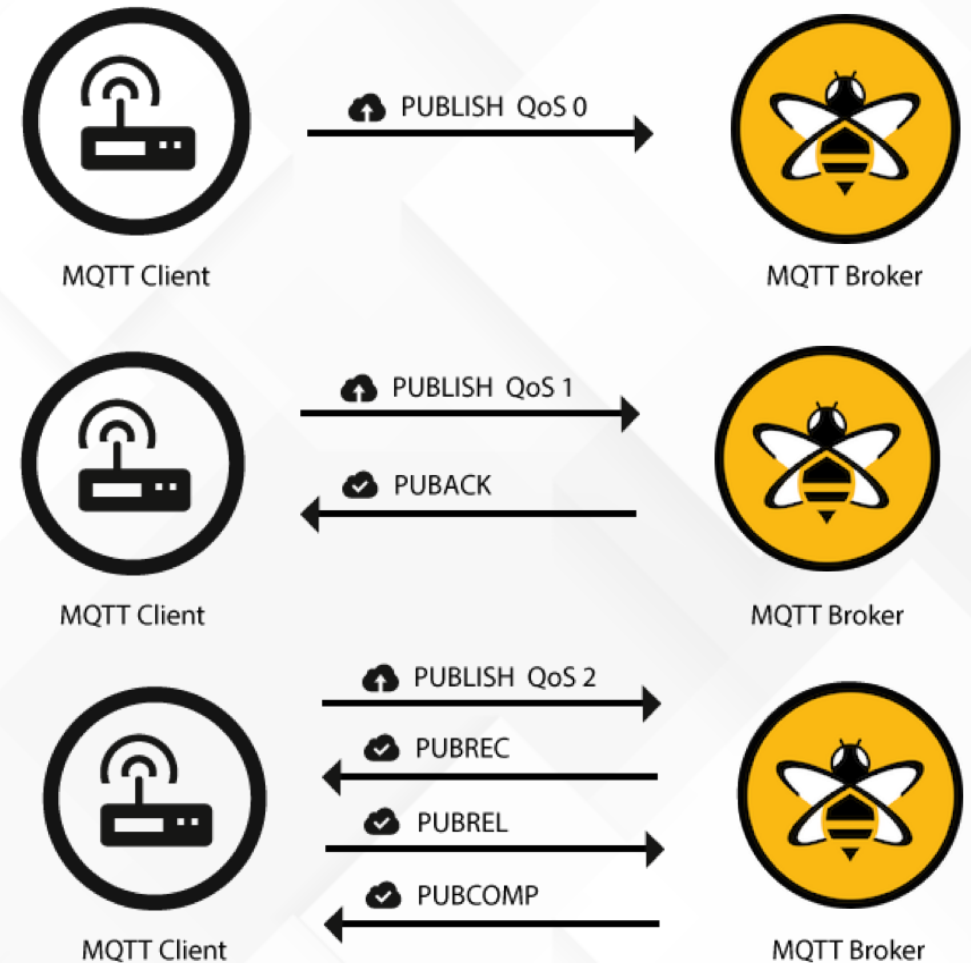
MQTT: Topics

- Los dispositivos publican y se suscriben a topics, que podríamos considerar como buzones.
- Varios dispositivos pueden publicar y suscribirse al mismo topic (no tiene por qué ser una comunicación 1 a 1).
- Los topics pueden crearse dinámicamente (no es necesario crearlos a priori)



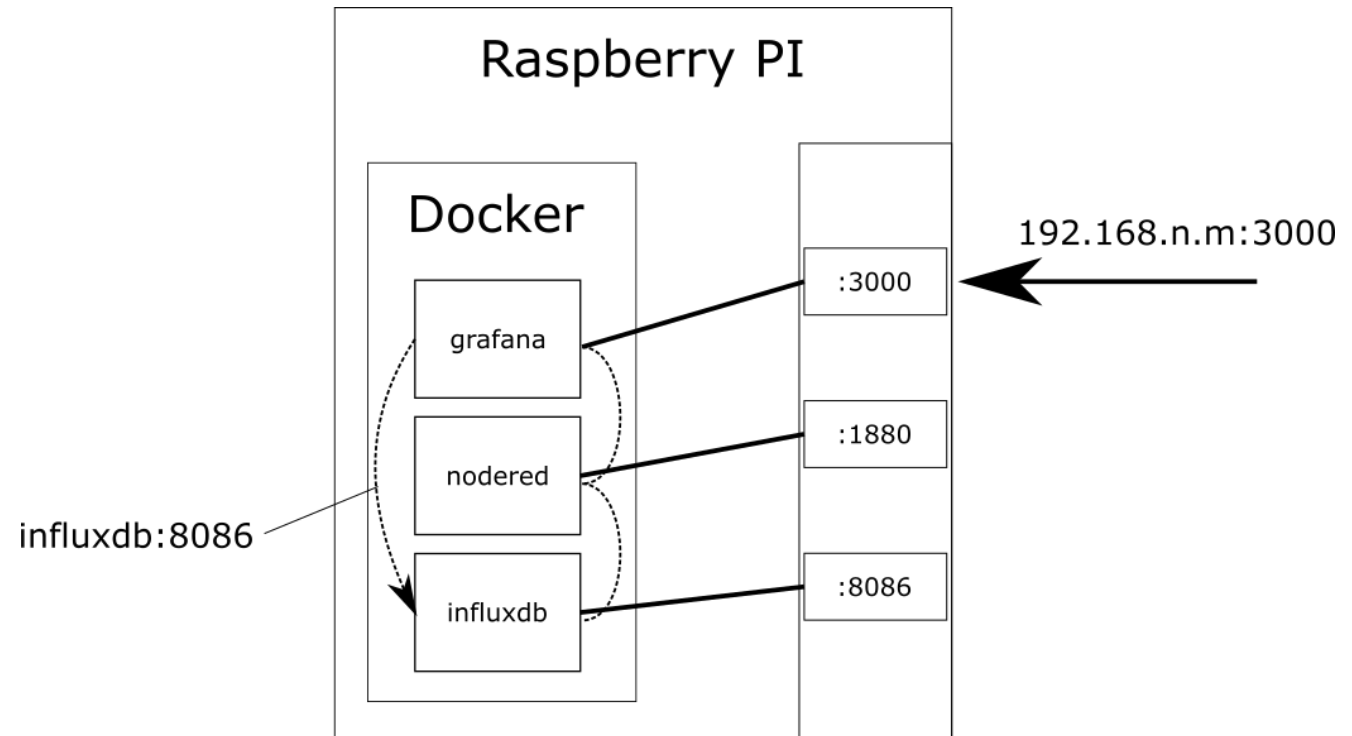
MQTT: Calidad de servicio

- QoS: Garantía de entrega
 - 0: **A lo sumo una vez**
 - Disparar y olvidar
 - El receptor no hace ACK al remitente
 - 1: **Al menos una vez**
 - Es posible que el remitente reciba varias copias del mensaje porque el bróker los procesa en cuanto lo recibe y, si tarda en enviar el PUBACK, el remitente podría reenviarlo.
 - 2: **Exactamente una vez**
 - Es el método más lento
 - **El bróker no despacha el mensaje hasta que no recibe el PUBREL** y así tiene la certeza de que el remitente no lo va a reemitir.

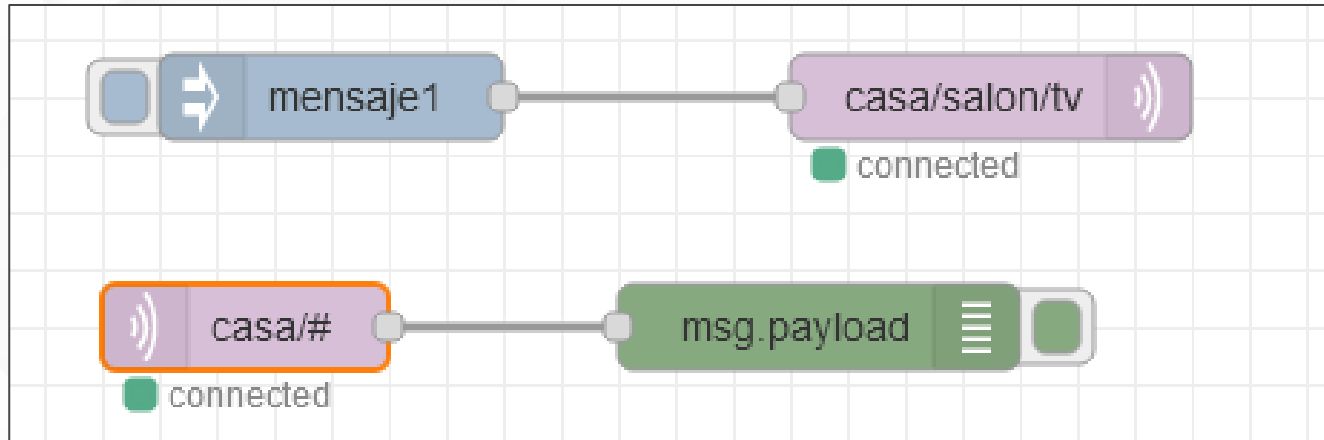


Networking en IoTstack

- Docker-compose crea una red interna para los contenedores con DNS, de modo que **los contenedores se comunican entre sí por su nombre** (en lugar de por la dirección IP del sistema)
 - Por ejemplo, para llamar a Mosquitto desde Node-RED tendríamos que indicar que el Broker de Mosquitto está en la dirección mosquitto:1883
 - Los nombres de los contenedores van siempre en minúsculas



Mosquitto desde Node-RED



Edit mqtt in node

Delete Cancel Done

Properties

Server mosquitto:1883

Action Subscribe to single topic

Topic casa/#

Edit mqtt out node

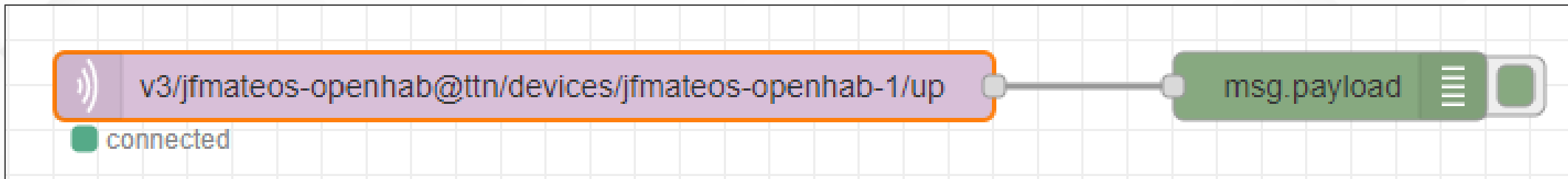
Delete Cancel Done

Properties

Server mosquitto:1883

Topic casa/salon/tv

TTN desde Node-RED



Edit mqtt in node > Edit mqtt-broker node

Delete Cancel Update

Properties

Name TTN

Connection Security Messages

Server eu1.cloud.thethings.network:1883 Port 1883

☒ Connect automatically

☐ Use TLS

Protocol MQTT V3.1.1

Client ID Leave blank for auto generated

Keep Alive 60

Session ☒ Use clean session

Edit mqtt in node > Edit mqtt-broker node

Delete Cancel Update

Properties

Name TTN

Connection application id@ttn ges

Username jfmateos-openhab@ttn

Password MQTT API KEY

InfluxDB 1/3

- InfluxDB es una base de datos de series temporales (TSDB) open source
- Los datos se almacenan en
 - measurements (medidas)
 - series (son points que comparten tags)
 - points = measurement + timestamp + tags + campos
- Sintaxis
 - `<measurement>[,<tag-key>=<tag-value>...] <field-key>=<field-value>[,<field2-key>=<field2-value>...] [unix-nano-timestamp]`
 - `temperatura,nevera=25,unidad=C exterior=32,interior=3.8`
- Se pueden establecer **políticas de retención** sobre los measurements:
 - **Duración**
 - **Número de réplicas**
 - Duración de los **shard groups**: Un shard group es el conjunto de archivos (shards) en el que se almacenan los datos correspondientes a un determinado periodo de tiempo (la duración).

InfluxDB 2/3

- Las consultas utilizan un **lenguaje "parecido" a SQL**, pero no se requiere un esquema fijo para los datos (no hay que definir las columnas; se pueden crear ad-hoc).
- En cierto modo un **measurement** es como una **tabla**, en la que:
 - El timestamp es el **índice primario**
 - Los tags están **indexados**
 - Los campos (fields) **no están indexados**

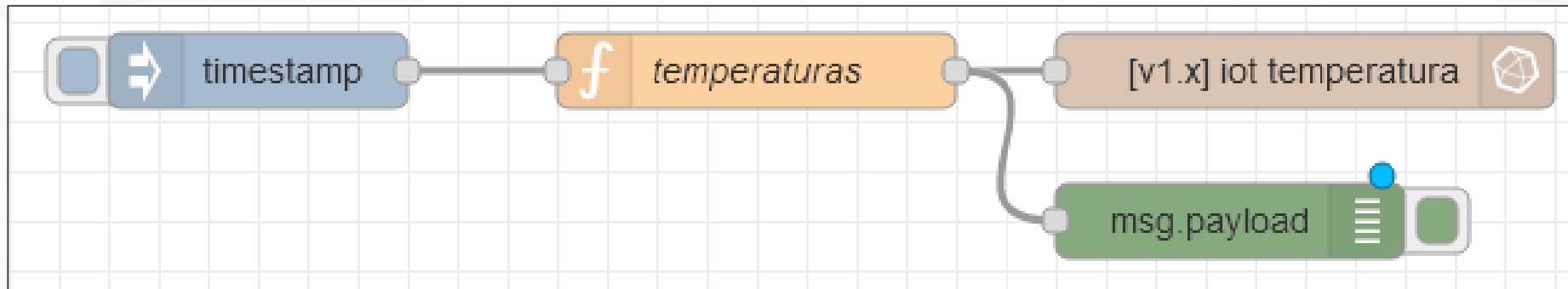
InfluxDB 3/3

La consola de InfluxDB

- Desde la carpeta /IoTstack, abrir un Shell (para salir Control + D)
 - `docker-compose exec influxdb bash`
- Acceder a la consola (para salir Control + D)
 - `influx`
- Mostrar bases de datos: `SHOW databases`
- Crear una base de datos: `CREATE database iot`
- Activar una base de datos: `USE iot`
 - Mostrar las measurements de la base de datos: `show measurements`
 - Insertar un punto: `INSERT temperatura,nevera=25,unidad=C exterior=32,interior=3.8`
 - Consultar: `SELECT nevera,exterior FROM temperatura`

```
pi@raspberrypi3:~/IoTstack $ docker-compose exec influxdb bash
WARNING: Some networks were defined but are not used by any service: nextcloud
root@1217a4d344e7:/# influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> show databases
name: databases
name
----
telegraf
iot
> use iot
Using database iot
> show measurements
> INSERT temperatura,nevera=25,unidad=C exterior=32,interior=3.8
> show measurements
name: measurements
name
----
temperatura
> SELECT nevera,exterior FROM temperatura
name: temperatura
time                nevera exterior
----                -
1650964091065690705 25      32
>
```

Node-RED e InfluxDB



Edit function node

Delete Cancel Done

Properties

Name: temperaturas

Setup On Start On Message On Stop

```
1 msg.payload=[];
2 msg.payload[0]={"exterior":20+Math.random()*5,"interior":Math.random()*5};
3 msg.payload[1]={"nevera":25,"unidad":"C"};
4 return msg;
```

Edit influxdb out node > Add new influxdb config node

Cancel Add

Properties

Name: iot

Versión: 1.x

Host: influxdb Port: 8086

Database: iot

Username:

Password:

☐ Activar conexión (SSL/TLS) segura

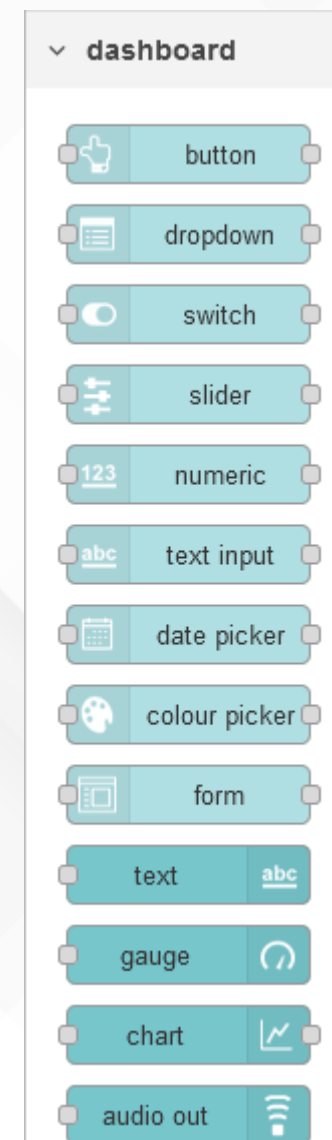
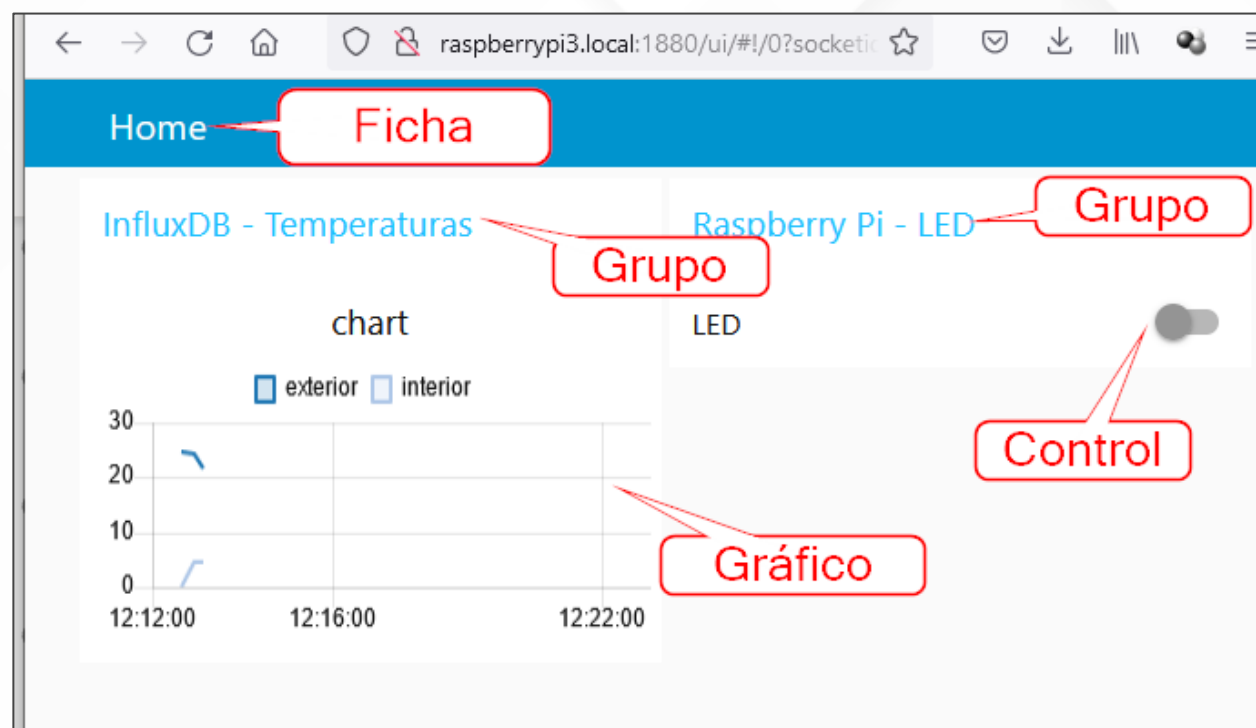
No usar 127.0.0.1

Node-RED: Dashboards 1/3

Node-RED ofrece un sistema de dashboards básico, pero muy conveniente (cómodo).

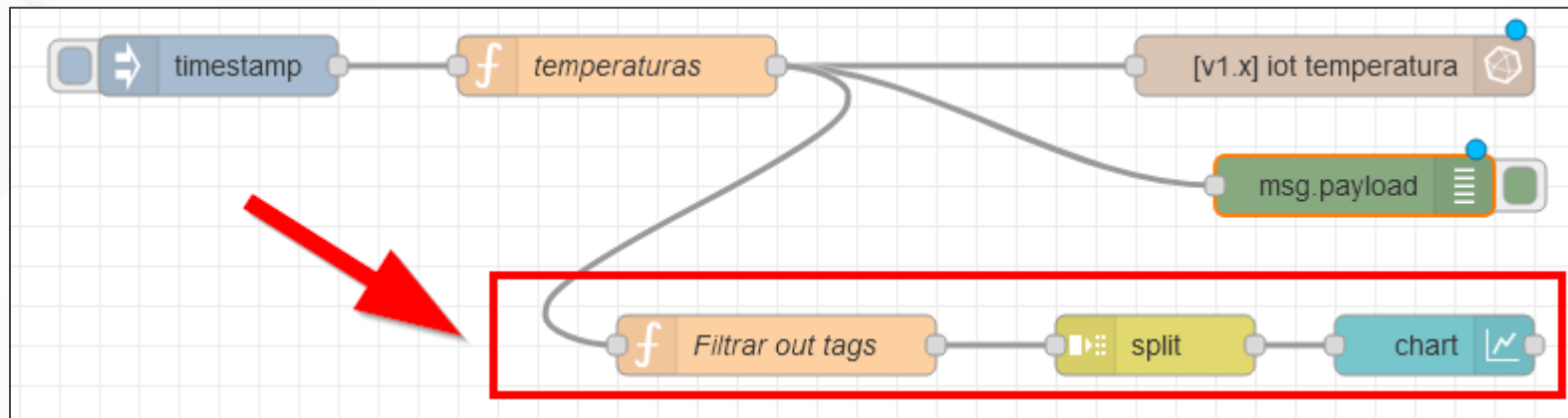
La estructura es:

- Ficha (tab)
 - Grupo
 - Control o gráfico



Node-RED: Dashboards 2/3

Añadir un gráfico que muestre las temperaturas



Edit function node

Delete Cancel Done

Properties

Name: Filtrar out tags

Setup On Start On Message On Stop

```
1 msg.payload=msg.payload[0];
2 return msg;
```

Edit split node

Delete Cancel Done

Properties

Split msg.payload based on type:

String / Buffer

Split using: a-z \n

☐ Handle as a stream of messages

Array

Split using: Fixed length of 1

Object

Send a message for each key/value pair

☒ Copy key to: msg.topic

Edit chart node

Delete Cancel Done

Properties

Group: [Home] InfluxDB - Temperaturas

Size: auto

Label: chart

Type: Line chart ☐ enlarge points

X-axis: last 1 hours OR 1000 points

X-axis Label: HH:mm:ss ☐ as UTC

Y-axis: min max

Legend: Show ☐ interpolate linear

Grafana

- Grafana es un servicio web open source que permite **visualizar** datos y generar **alertas** (no está especializado en controlar los dispositivos, aunque es limitadamente posible mediante plugins como Button Panel).
- Conceptos fundamentales:
 - **Data Source** u Origen de datos (InfluxDB, MongoDB, MySQL, PostgreSQL...)
 - **Carpeta** (conjunto de dashboards)
 - **Dashboard** (conjunto de paneles)
 - Dashboard snapshot
 - Playlist: Secuencia de paneles
 - **Panel**



Añadir InfluxDB como origen de datos para Grafana

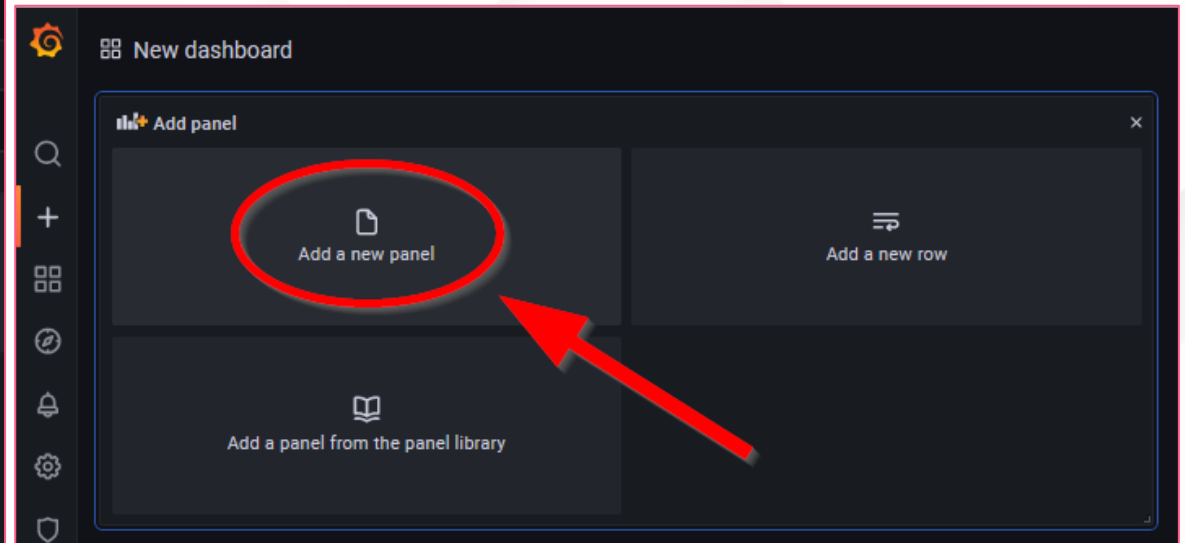
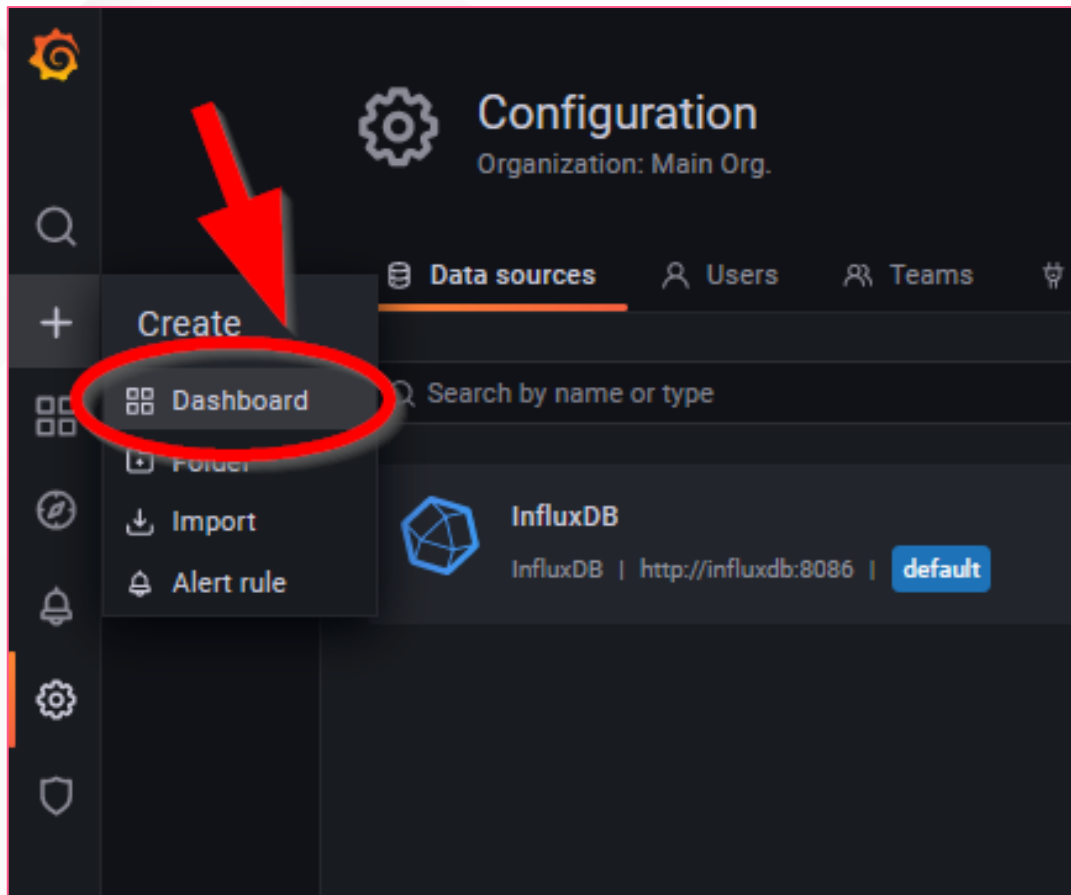
The image shows a sequence of three overlapping screenshots from the Grafana web interface, illustrating the process of adding an InfluxDB data source. The browser address bar at the top indicates the URL is `raspberrypi3.local:3000/datasources`.

Left Screenshot: The 'Configuration' page for 'Main Org.' is shown. The 'Data sources' tab is selected in the top navigation bar. In the left sidebar, the 'Data sources' menu item is circled in red, with a red arrow pointing towards the right screenshot.

Middle Screenshot: The 'Data Sources / InfluxDB' configuration page is displayed. The 'Settings' tab is active. The 'Name' field is set to 'InfluxDB'. Under the 'HTTP' section, the 'URL' field is circled in red and contains the value `http://influxdb:8086`. A red arrow points from this field towards the right screenshot.

Right Screenshot: The 'InfluxDB Details' modal is open. It shows the 'Database Access' section with a warning icon. Below this, the 'Database' field is set to 'iot' and the 'User' field is set to 'nodered'. The 'Password' field is set to 'configured' and has a 'Reset' button next to it. A red box highlights the 'Database' and 'User' fields. A red arrow points from the 'User' field to a red callout box containing the text 'nodered'. At the bottom of the modal, there are buttons for 'Back', 'Explore', 'Delete', and 'Save & test'.

Añadir un dashboard y un panel con las temperaturas 1/2



Añadir un dashboard y un panel con las temperaturas 2/2

New dashboard / Edit Panel

Table view ☐ Fill Actual Last 6 hours

Time series

Panel Title

30
20
10
0

08:00 08:30 09:00 09:30 10:00 10:30 11:00 11:30 12:00 12:30 13:00 13:30

temperatura.exterior temperatura.interior

Query 2 Transform 0 Alert 0

Data source InfluxDB Query options MD = auto = 1187 Interval = 20s Query inspector

> A (InfluxDB) SELECT "exterior" FROM "temperatura" WHERE \$timeFilter

> B (InfluxDB)

FROM default temperatura WHERE

SELECT field(interior) +

GROUP BY +

TIMEZONE (optional) ORDER BY TIME

LIMIT (optional)

FORMAT AS Time series ALIAS naming pattern

+ Query + Expression

Panel options

Title Temperaturas

Description

Transparent background

Panel links

Repeat options

Tooltip

Tooltip mode Single All Hidden

Legend

Legend mode List Table Hidden

Legend placement Bottom Right

Hacer clic sobre un valor muestra sus opciones...

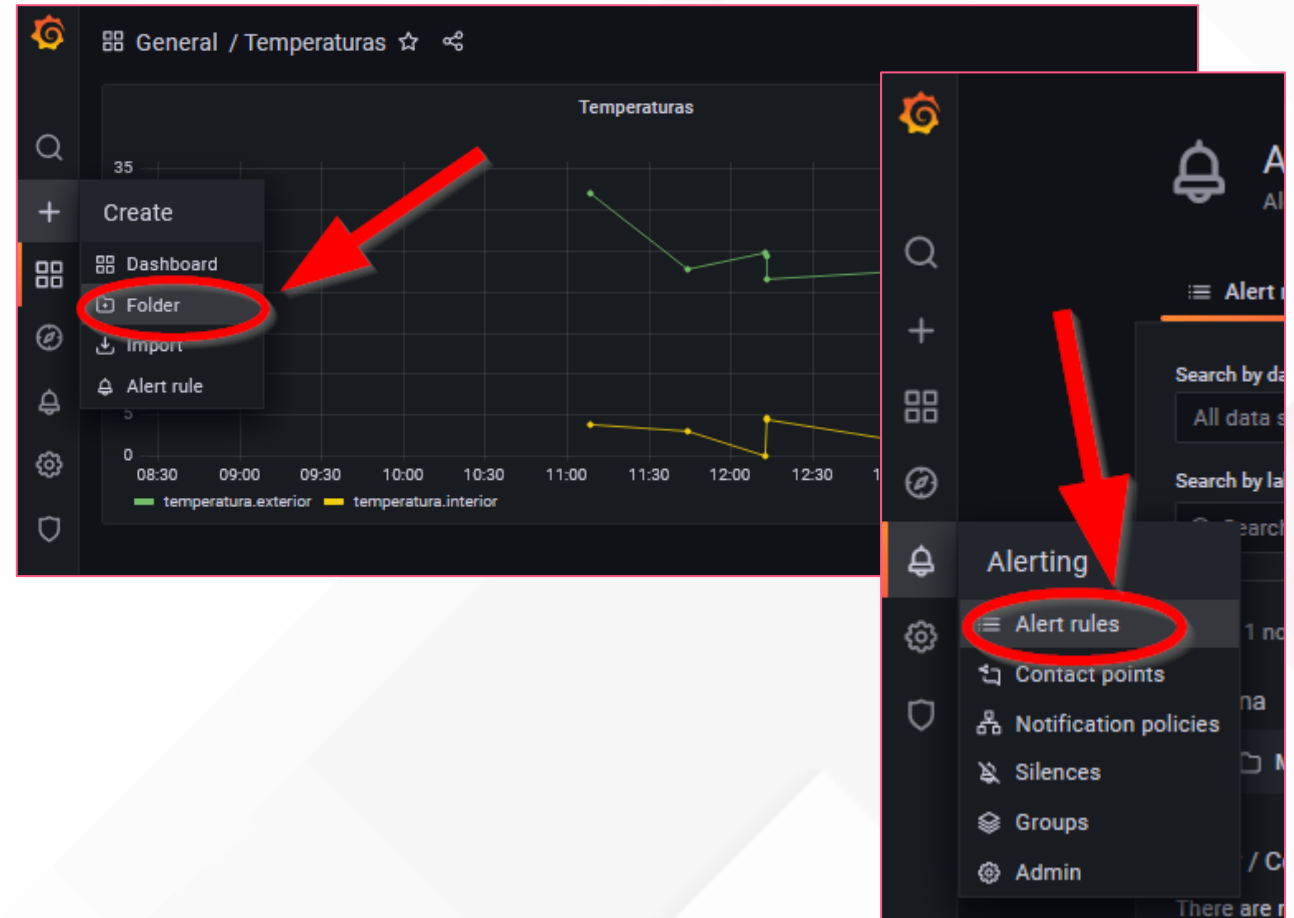
Hacer clic sobre un campo muestra la opción Remove

Grafana: Alertas

- **Alerting rule:** La regla que define el disparador de la alerta. Las reglas se ejecutan a intervalos regulares (por ejemplo, cada 10 minutos) y, si se cumple el criterio, pasan a estado **pending** durante un periodo de cautela/cortesía/histéresis y, luego, a estado **alerting/firing**.
- **Contact point:** Destino de las notificaciones, como email, Telegram, Microsoft Teams...
- **Notification policy:** Las alertas pueden incluir **labels** (etiquetas). Mediante estas etiquetas podemos declarar una política que indique a que contact point hay que enviar cada alerta.
- **Silences:** Permite establecer periodos para no enviar alertas (fines de semana, periodos de descanso...)

Grafana: Crear una alerta para la temperatura interior 1/2

1. Las alertas deben almacenarse en una carpeta; empezaremos creando una carpeta (folder) llamada **miCarpeta**.
2. Iniciar la creación de la Alert rule.



Grafana: Crear una alerta para la temperatura interior 2/2

3. Definir la regla que disparará la alerta (1 y 2).
4. Configurar la pauta de evaluación, el intervalo de pending... (3 a 6).

2 Alert evaluation behavior

Evaluate
Evaluation interval applies to every rule within a group. It can overwrite the interval of an existing alert rule.

Evaluate every ☐ 1m for ☐ 5m **3**

> Configure no data and error handling

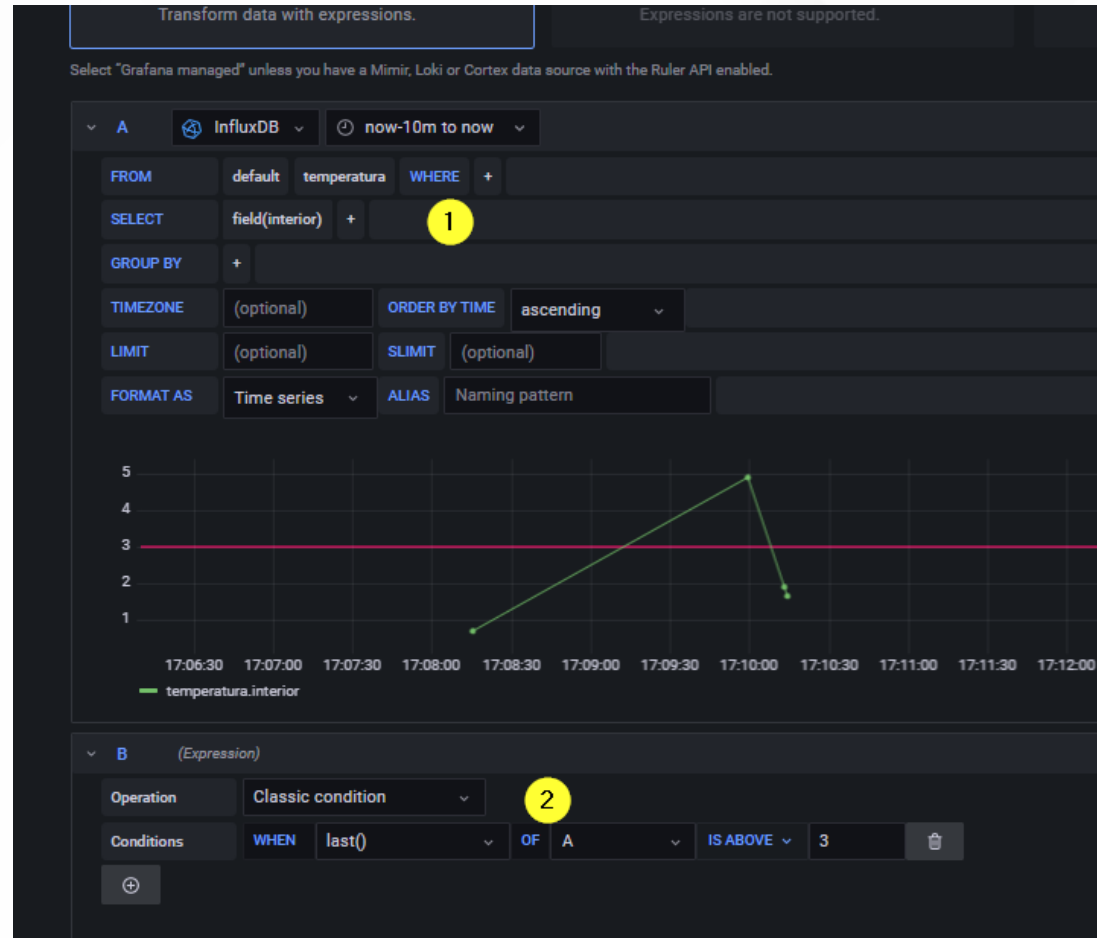
Preview alerts

3 Add details for your alert
Write a summary and add labels to help you better manage your alerts

Rule name **4**
Temperatura interior muy alta

Folder ☐ **5**
Select a folder to store your alert in
miCarpeta

Group **6**
Rules within the same group are evaluated after the same time interval.
Grupo general

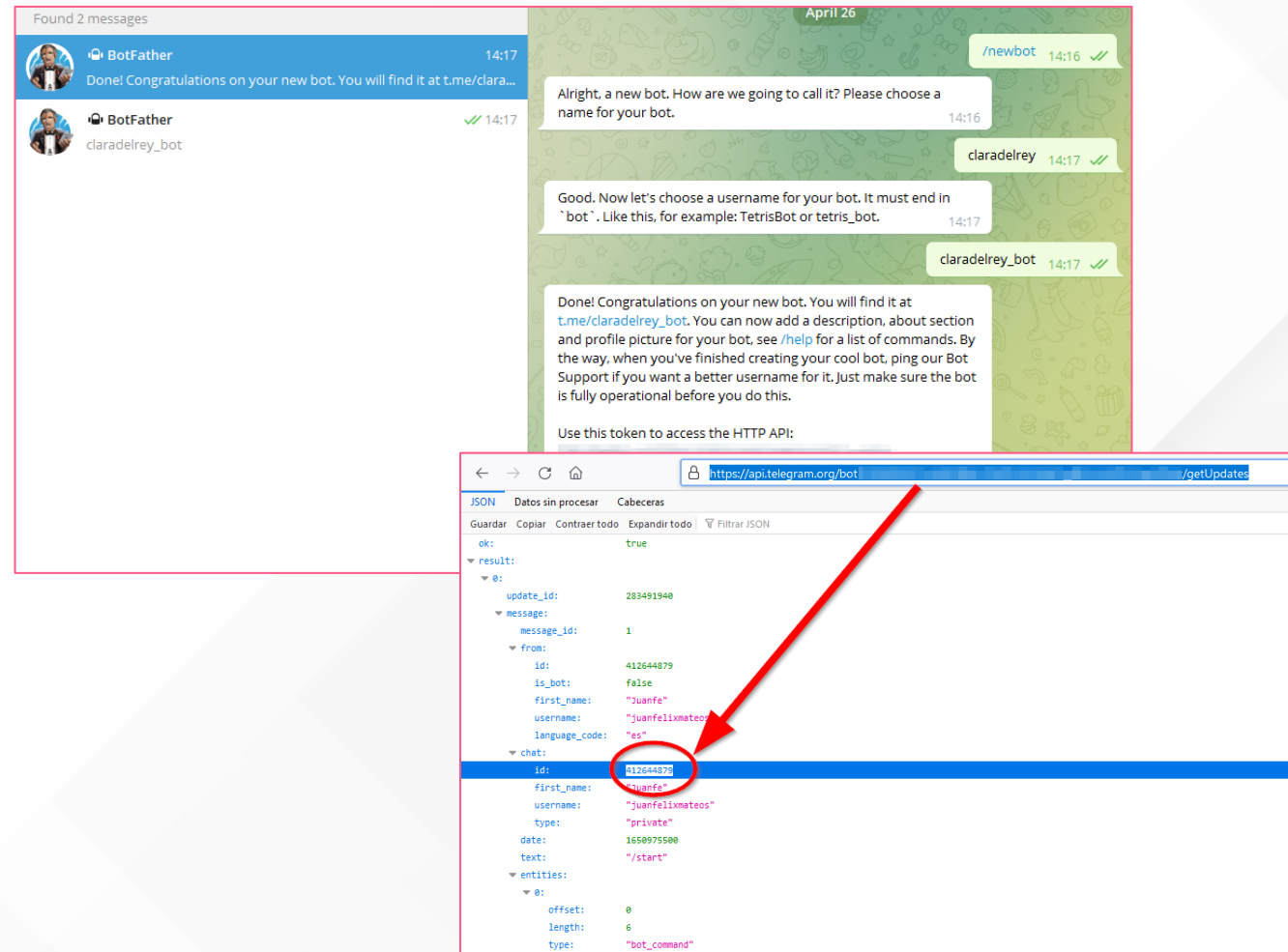


Grafana: Crear un contact point para la alerta

1/3

Vamos a solicitar que la alerta nos llegue a Telegram.

1. Crear un bot Telegram con BotFather y obtener el **token**.
2. Iniciar conversación con el bot creado en el paso anterior.
3. Obtener el **chatid** de la conversación accediendo a la dirección:
 - [https://api.telegram.org/bot\[token\]/getUpdates](https://api.telegram.org/bot[token]/getUpdates)



Grafana: Crear un contact point para la alerta

2/3

3. Definir el contact point.
4. Crear una plantilla (utiliza el lenguaje de plantillas de Go)

```
{{ define "myalert" }}
[[{.Status}]] {{ .Labels.alertname }}

Labels:
{{ range .Labels.SortedPairs }}
  {{ .Name }}: {{ .Value }}
{{ end }}

{{ if gt (len .Annotations) 0 }}
Annotations:
{{ range .Annotations.SortedPairs }}
  {{ .Name }}: {{ .Value }}
{{ end }}
{{ end }}

{{ if gt (len .SilenceURL) 0 }}
Silence alert: {{ .SilenceURL }}
{{ end }}
{{ if gt (len .DashboardURL) 0 }}
Go to dashboard: {{ .DashboardURL }}
{{ end }}
{{ end }}
```

****Firing****

Value: [var='B0' metric='temperatura.interior' labels={}
value=3.405288423980526]
Labels:
- alertname = Temperatura interior muy alta
Annotations:
Source: http://localhost:3000/alerting/grafana/ceHaBYzgk/view
Silence: http://localhost:3000/alerting/silence/new?
alertmanager=grafana&matcher=alertname%3DTemperatura+interior+muy+alta

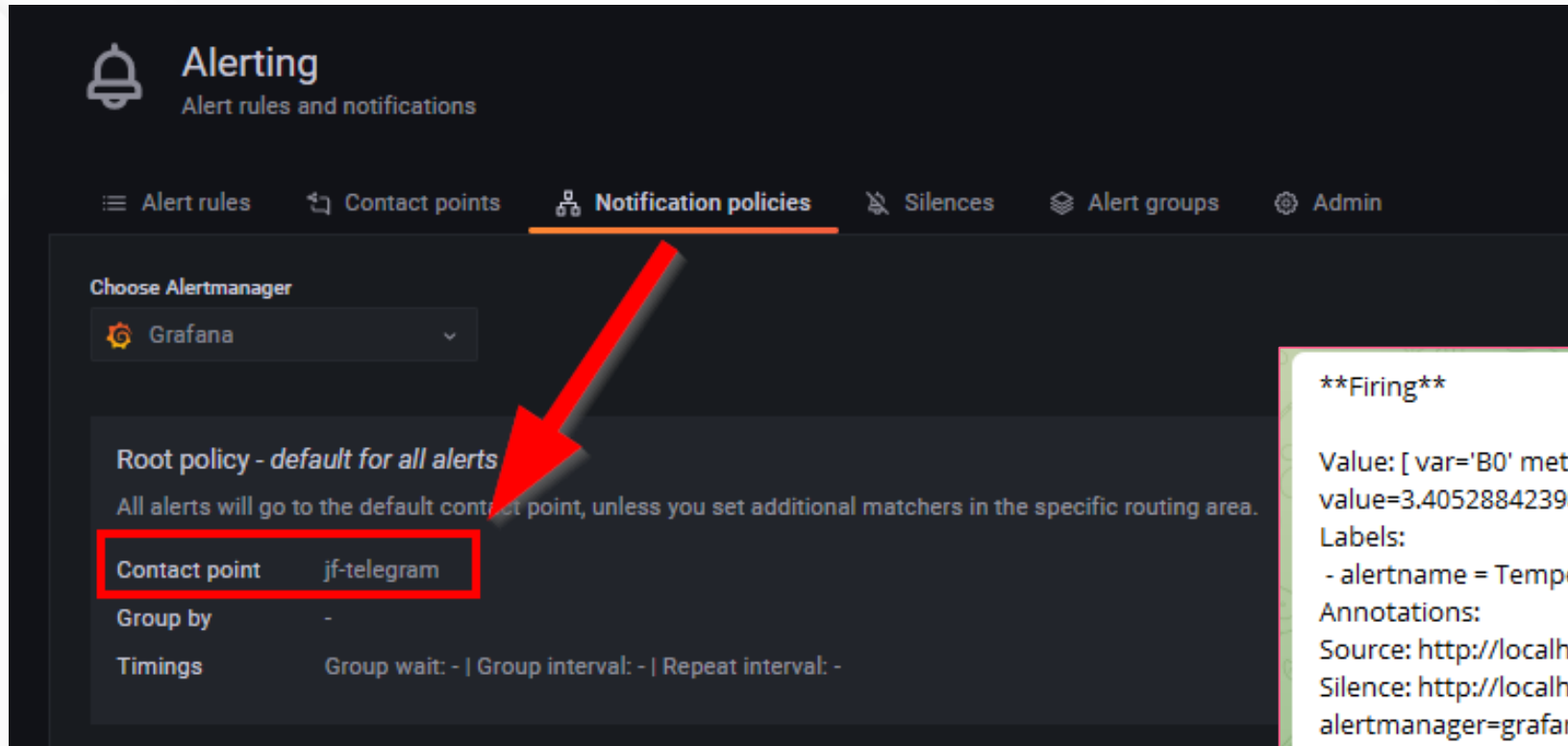
14:50

The screenshot shows the 'Alerting' section of the Grafana UI, specifically the 'Contact points' tab. The 'Alertmanager' is set to 'Grafana'. Under 'Update contact point', the 'Name' is 'Juanfe-telegram'. The 'Contact point type' is 'Telegram', which is circled in red. Below it, the status is 'Configured'. The 'Chat ID' field is labeled 'Integer Telegram Chat Identifier' and contains a redacted value. There is also an 'Optional Telegram settings' section partially visible.

The screenshot shows the 'Edit message template' dialog in the Grafana Alerting interface. It displays the Go templating language code for the alert template, which is the same code shown in the first code block. The 'Template name' is 'Temperatura interior muy alta'. At the bottom, there are 'Save template' and 'Cancel' buttons.

Grafana: Crear un contact point para la alerta 3/3

5. Asignar el contact point a la política



Alerting
Alert rules and notifications

Alert rules | Contact points | **Notification policies** | Silences | Alert groups | Admin

Choose Alertmanager
Grafana

Root policy - default for all alerts
All alerts will go to the default contact point, unless you set additional matchers in the specific routing area.

Contact point	jf-telegram
Group by	-
Timings	Group wait: - Group interval: - Repeat interval: -

****Firing****

Value: [var='B0' metric='temperatura.interior' labels={} value=3.405288423980526]

Labels:

- alertname = Temperatura interior muy alta

Annotations:

Source: <http://localhost:3000/alerting/grafana/ceHaBYzgk/view>

Silence: [http://localhost:3000/alerting/silence/new?](http://localhost:3000/alerting/silence/new?alertmanager=grafana&matcher=alertname%3DTemperatura+interior+muy+alta)

[alertmanager=grafana&matcher=alertname%3DTemperatura+interior+muy+alta](http://localhost:3000/alerting/silence/new?alertmanager=grafana&matcher=alertname%3DTemperatura+interior+muy+alta)

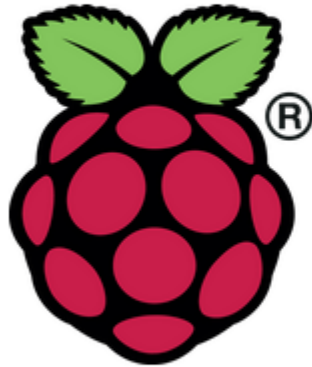
14:50

Una excelente lista de recursos para Raspberry Pi

- <https://github.com/thibmaek/awesome-raspberry-pi>

Awesome Raspberry Pi

 awesome  all contributors  43



The Raspberry Pi is a series of credit card-sized single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and developing countries. Official Link: [Raspberry Pi Foundation Homepage](#), [Raspberry Pi Computer Homepage](#)

This list is a collection of tools, projects, images and resources conforming to the [Awesome Manifesto](#)

Contributions *very welcome* but first see [Contributing](#)

<https://linuxjourney.com/>

Una excelente referencia de Linux



jfmateos@educa.madrid.org

GRACIAS

