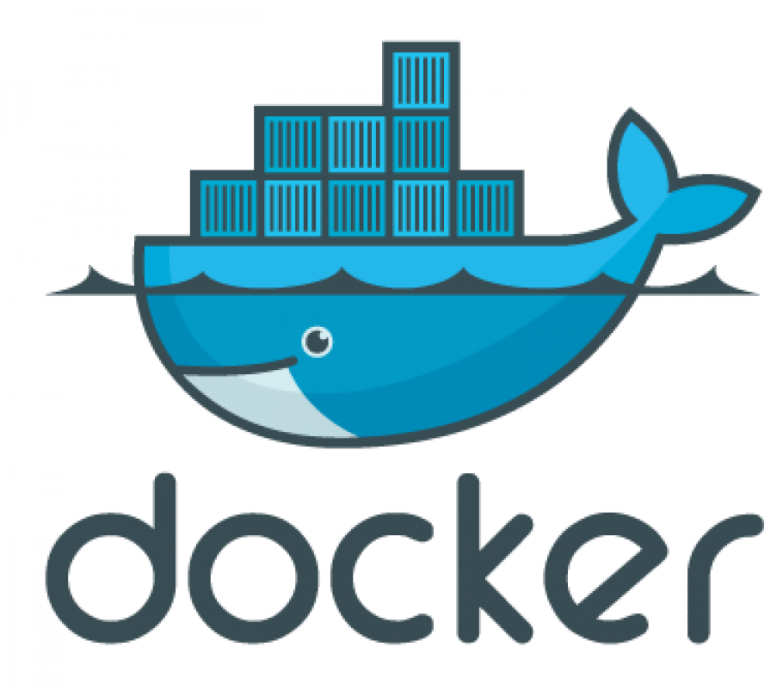


## Sistemas Industriales

### Practica 2 - Ejercicio de Docker

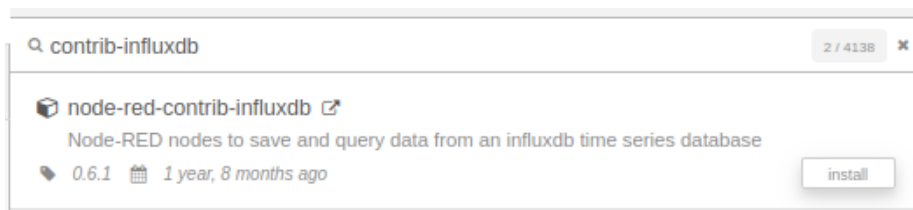


Elvi Mihai Sabau Sabau<sup>[51254875L]</sup>

<sup>1</sup> Universidad de Alicante, Alicante, España.  
[emss5@alu.ua.es](mailto:emss5@alu.ua.es)

## Despliegue y Interacción con una DB usando docker y docker-compose, y registro de un historico de los eventos de nuestra instalación domotica.

Para este objetivo, usaremos una base de datos influxdb, en node usaremos el nodo influxdb para interactuar con esta.

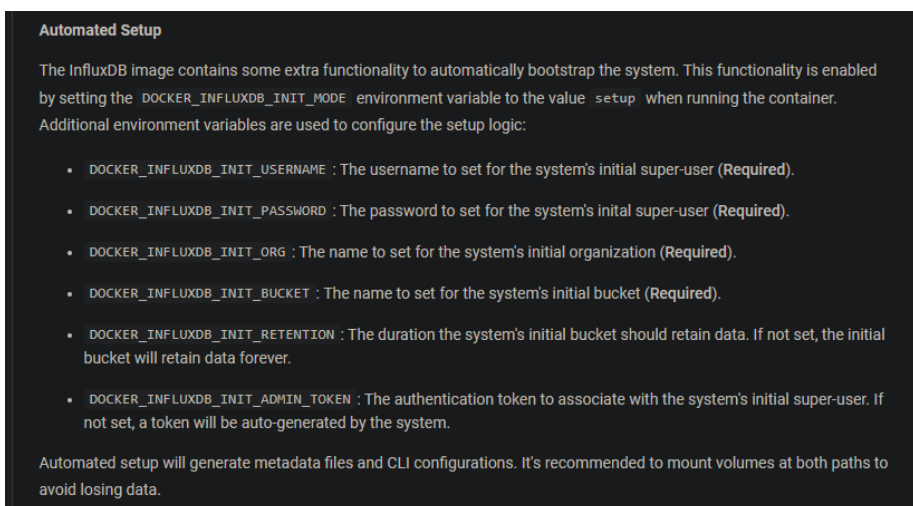


Este nodo nos permite interactuar con bases de datos influxdb. InfluxDB es una base de datos que guarda datos por series de tiempos, por lo tanto difiere al esquema al que estamos acostumbrados de las bases de datos relacionales.

Primero antes de todo, vamos a desplegar la base de datos y node-red via docker, usando docker compose.

Para ello, creamos un archivo llamado “docker-compose.yml”, en este desplegaremos un contenedor docker y influxdb, compartiendo la misma red, y con un volumen propio para cada contenedor para tener persistencia al apagar el contenedor, también vamos a mapear los puertos de cada contenedor.

La documentación de la imagen de docker de influxdb nos pide que especifiquemos mediante variables de entorno las credenciales y otros datos de configuración para el contenedor.

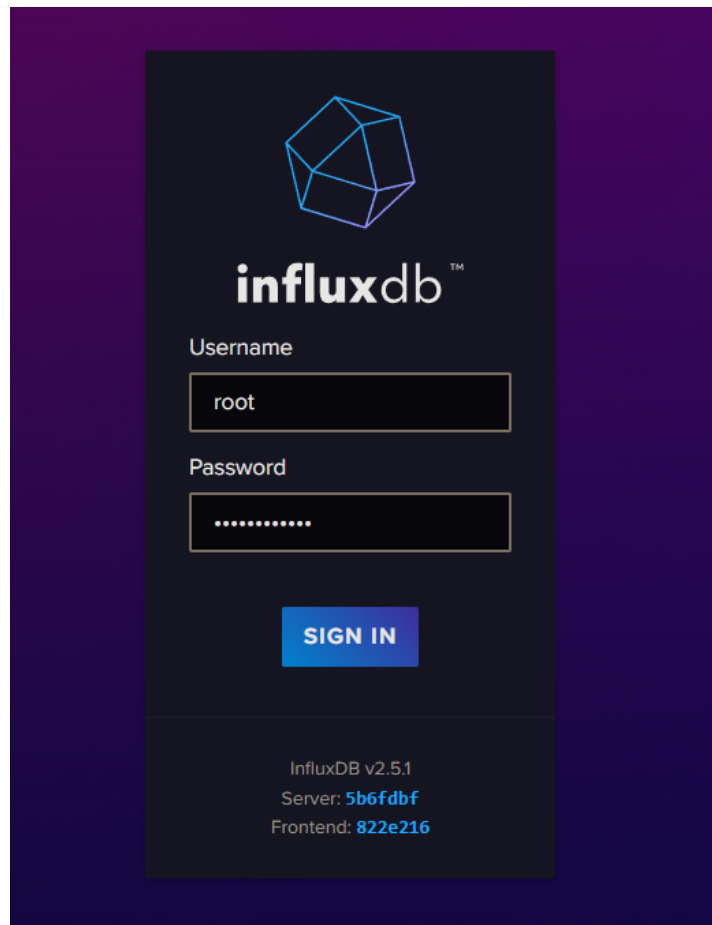


Nuestro “docker-compose.yml” debería quedar así:

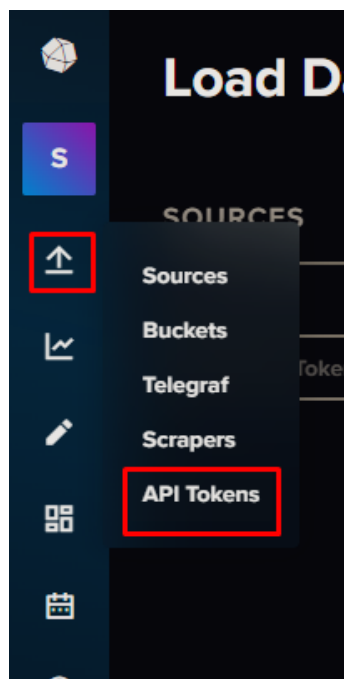
```
docker-compose.yml
1  # @Author: Frenzoid - ELVI MIHAI SABAU SABAU
2  # @Description: Docker Compose para Node-RED y InfluxDB
3
4  version: '3'
5  services:
6    influxdb:
7      image: influxdb:latest
8      container_name: INFLUXDB_SERVER
9      restart: always
10     ports:
11       - 8086:8086
12     volumes:
13       - influxdb_vol:/var/lib/influxdb
14     environment:
15       - DOCKER_INFLUXDB_INIT_MODE: setup
16       - DOCKER_INFLUXDB_INIT_USERNAME: root
17       - DOCKER_INFLUXDB_INIT_PASSWORD: rootrootroot
18       - DOCKER_INFLUXDB_INIT_ORG: SIND
19       - DOCKER_INFLUXDB_INIT_BUCKET: SIND_B
20     networks:
21       - intranet
22
23     nodered:
24       image: nodered/node-red:latest
25       container_name: MODERED_SERVER
26       restart: always
27       ports:
28         - 1880:1880
29       volumes:
30         - nodered_vol:/data
31       networks:
32         - intranet
33
34     networks:
35       default:
36         external: true
37         name: intranet
38
```

Y desplegamos el contenedor con el comando: “docker compose up -d”. También, para apagar el stack de contenedores ejecutaremos el comando “docker compose down”.

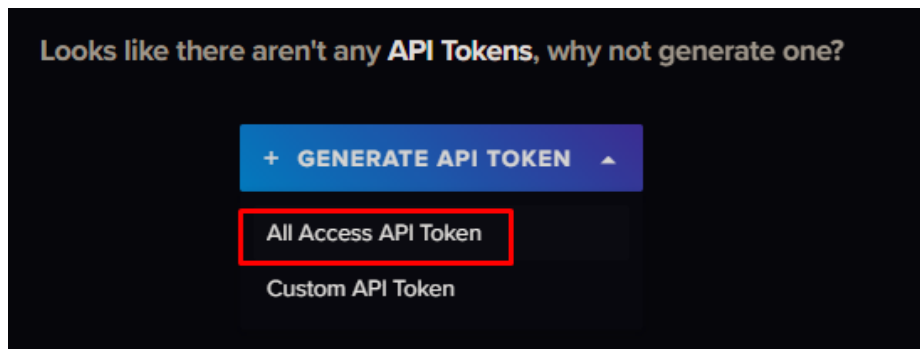
Accedemos a localhost:8086 y iniciamos sesión con las credenciales que hemos definido en las variables de entorno del contenedor.



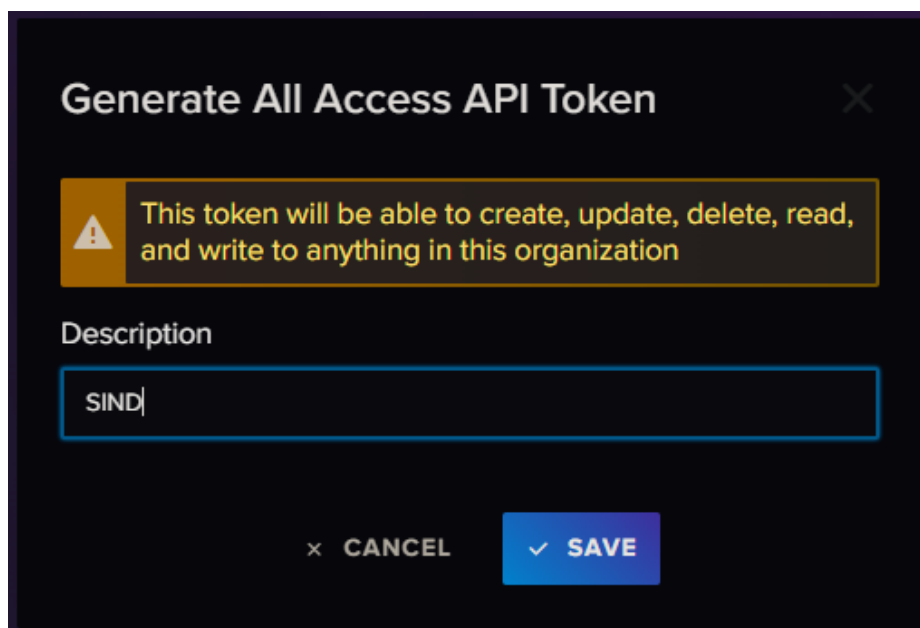
Omitimos el tutorial, y vamos a al apartado de API Tokens:



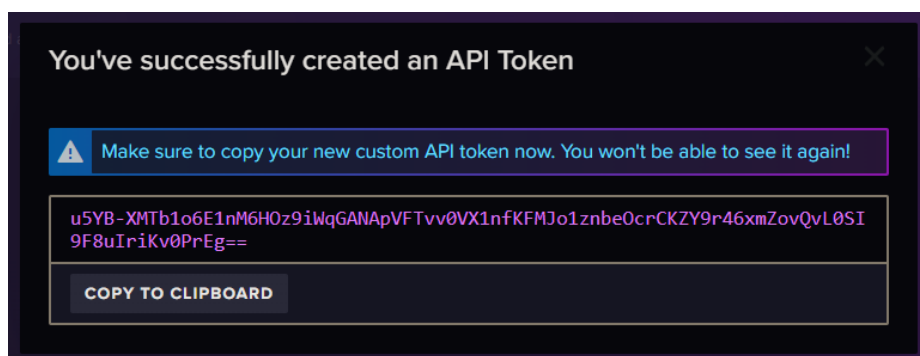
Y creamos un token con todos los permisos.



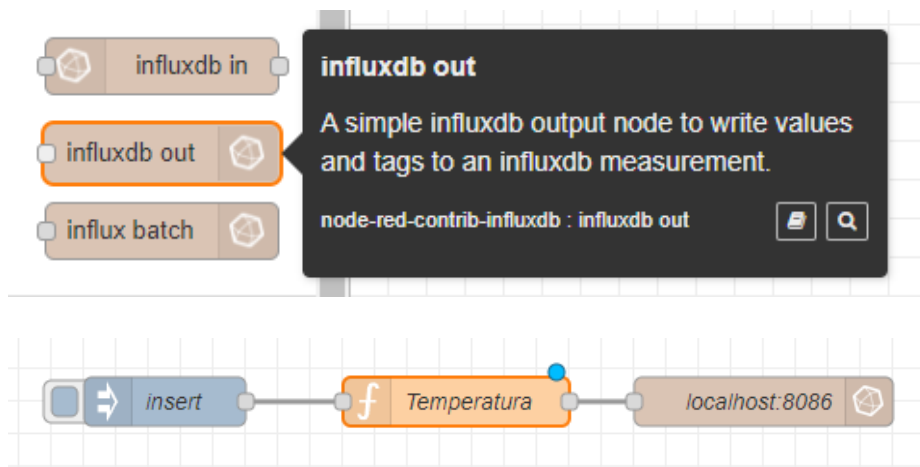
Le damos un nombre.



Y copiamos el token. Este token es lo que usaremos en nodered como credencial para acceder a la base de datos.



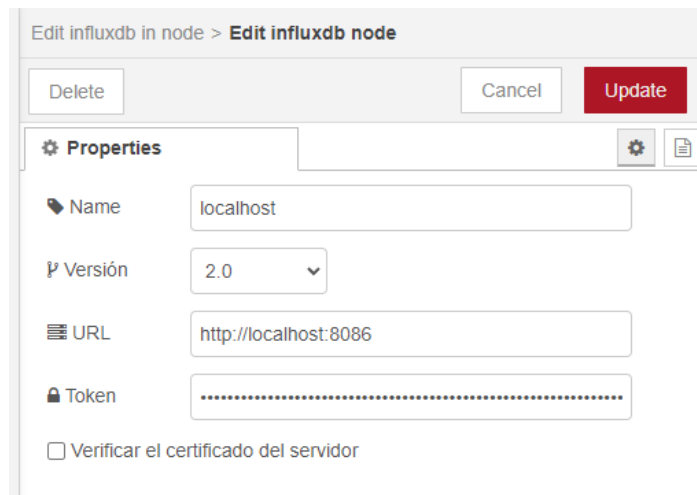
Ahora, en nodered, accediendo a “http://localhost:1880”, vamos a crear un nodo “influxdb out” para insertar datos, haremos una pequeña prueba para comprobar y familiarizarnos con este nodo.



Accedemos al nodo de “influxdb out” > Servidor: y añadimos un nuevo servidor:

The image shows the 'Edit influxdb out node' dialog box. It has a 'Delete' button, a 'Cancel' button, and a 'Done' button. The 'Properties' section contains the following fields: 'Name' (localhost:8086), 'Servidor' (Add new influxdb...), and 'Medición' (datos). There is a red box around the 'Add new influxdb...' dropdown menu. At the bottom, there is a checkbox for 'Opciones avanzadas' and a yellow box with the text: 'Consejo: Si no se especifica una política de retención, se asume autogen.'

Y rellenamos los datos para agregar un nuevo servidor. En el campo de Token, pegamos el token que hemos generado antes.



Edit influxdb in node > Edit influxdb node

Delete Cancel Update

**Properties**

Name localhost

Versión 2.0

URL http://localhost:8086

Token .....

☐ Verificar el certificado del servidor

Una vez añadido el servidor, volvemos, y ahora nos aparecerán campos nuevos, especificamos la organización y la “medida” que en el caso de influxdb es un campo discriminante para asociar datos insertados. ( Bucket corresponde con el nombre dado al bucket que se creará por defecto mediante la variable de entorno especificada anteriormente ).



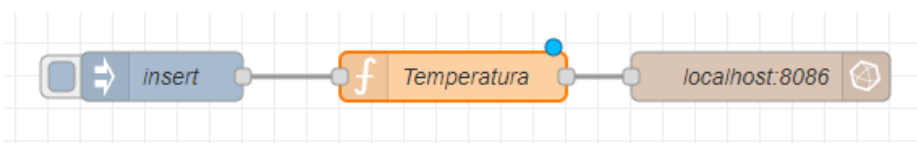
Organización SIND

Bucket SIND\_B

Medición datosejemplo

Precisión Milliseconds (ms)

Ahora en nuestro flujo:



En la función tenemos los siguiente:



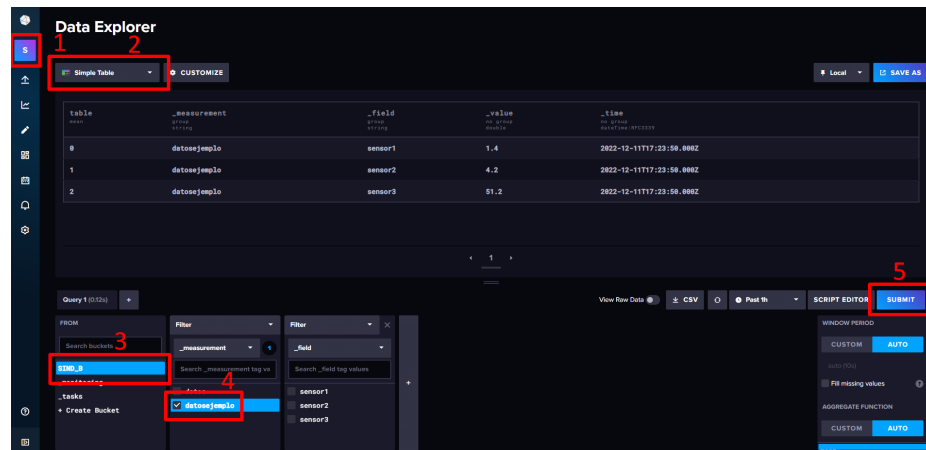
Name Temperatura

Setup On Start On Message On Stop

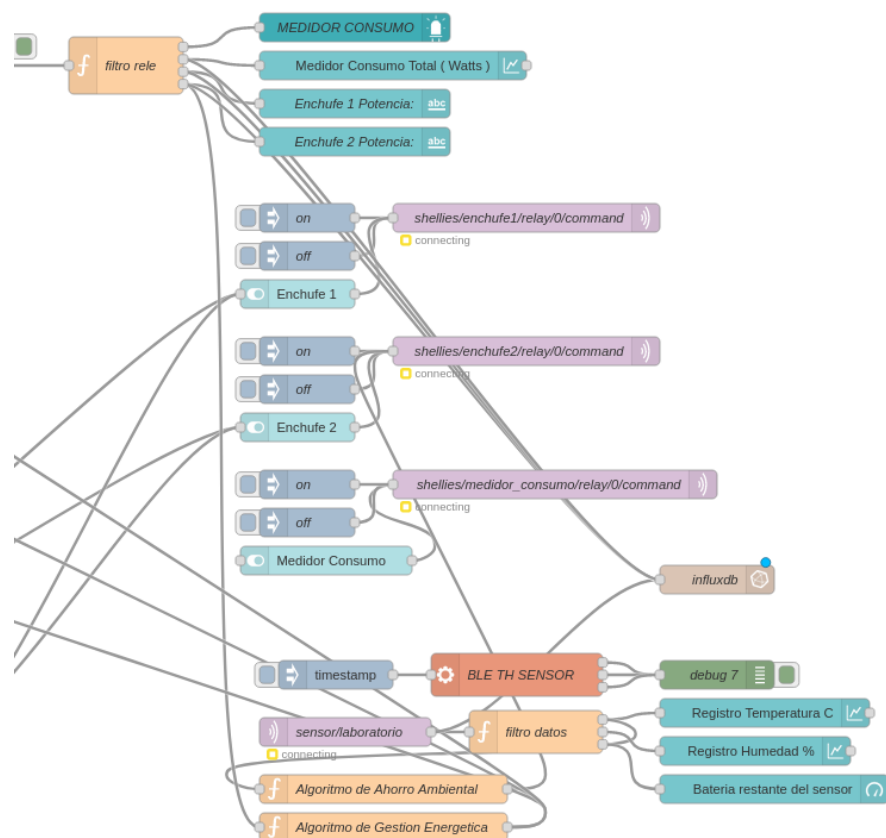
```
1 msg.payload = {sensor1: 1.4, sensor2: 4.2, sensor3: 51.2};
2 return msg;
```

Ahora, ejecutamos el inject “insert”, y vamos a localhost:8086 > Data Explorer, ponemos la vista de “tabla simple”, seleccionamos el bucket que hemos puesto al principio en las variables de entorno de docker que es la misma que hemos puesto en el nodo de nodered, seleccionamos la \_measurement que hemos especificado en el nodo, y a continuación los campos que concuerdan con las claves que hemos definido en la función.

Estos son los parametros de nuestra petición, ahora para solicitar los datos pulsamos en “Submit” y podremos ver en la tabla los datos insertados via nodered.



Ahora, para la práctica, aplicaremos este ejemplo para el dispositivo de medición de consumo, la temperatura y la humedad. En nuestro flujo, hemos añadido el nodo de influxdb, y insertamos 6 campos: temperatura, humedad, %bateria ( sensor de temperatura ), consumo enchufe 1, consumo enchufe 2, consumo total.





Y ahora, si vamos al panel de Data Explorer, podremos ver estos datos.

The screenshot displays the 'Data Explorer' interface. At the top, there's a 'Simple Table' view and a 'CUSTOMIZE' button. Below this is a table with the following data:

table	_measurement	_field	_value	_time
id	group	field	value	timestamp
0	datosensores	average	3	2022-12-12T14:14:56.726Z
0	datosensores	average	3	2022-12-12T14:14:58.726Z
1	datosensores	battery	62	2022-12-12T14:14:56.726Z

Below the table, there are pagination controls showing 111 items. The bottom section contains a 'Query Editor' with a 'FROM' clause set to 'SIMUL\_B'. There are two 'Filter' panels with search fields and a list of available fields: average, battery, enchufed2\_power, humidity, temperature, total, and value. On the right, there are options for 'View Raw Data', 'CSV', 'Post Th', and 'SCRIPT EDITOR'. At the bottom right, there are settings for 'WINDOW PERIOD' (CUSTOM/AUTO), 'Fill missing values', and 'AGGREGATE FUNCTION' (mean, median, max).