Configuración aplicación FiL de procesamiento de datos

Contenido

[Introducción 1](#_Toc69123448)

[InfluxDB 1](#_Toc69123449)

[Grafana 4](#_Toc69123450)

# Introducción

Esta aplicación FiL recoge datos de la BBDD del sistema para conseguir información con ellos y poder corregir u optimizar la producción. En este caso, analiza el rendimiento de las maquinas calculando su OEE (Overall Equipment Effectiveness) y lanzando un evento si es necesario.

En esta aplicación participan varias tecnologías, por lo tanto, durante este tutorial veremos cómo configurarlas para poder desplegar esta aplicación FiL.

# InfluxDB

Utilizaremos esta herramienta para guardar los cálculos obtenidos de la aplicación FiL. Estos cálculos serán **disponibilidad**, **rendimiento** y el **OEE**.

Primero de todo utilizaremos la imagen base oficial de InfluxDB para desplegarla en nuestro cluster.

*docker-compose.yml:*

influx-db:

image: influxdb

container\_name: influx-db

environment:

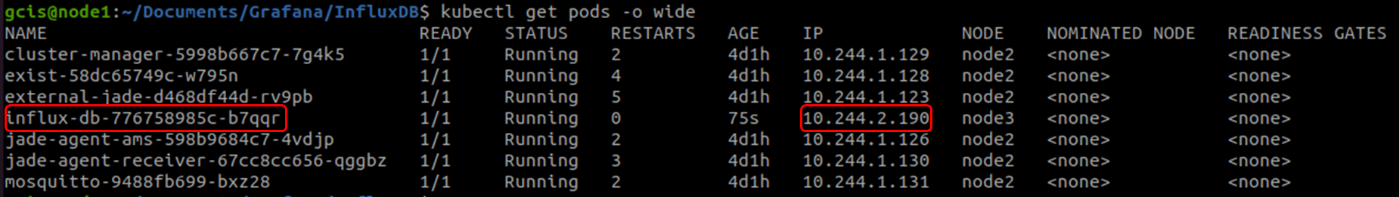
- PRE\_CREATE\_DB=gcis

ports:

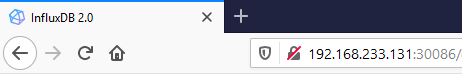
- 8083:8083

- 8086:8086

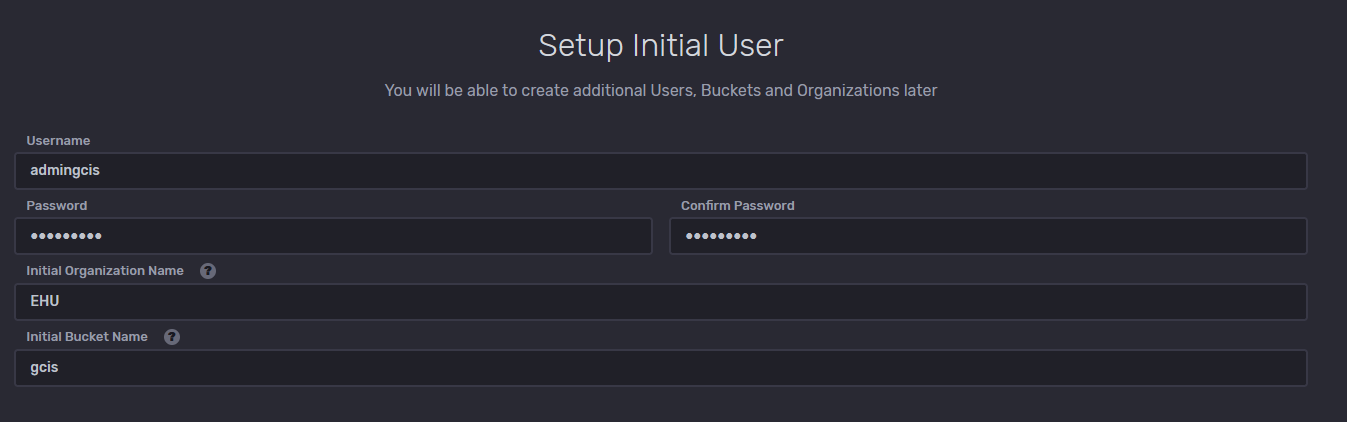
Cuando despleguemos InfluxDB comprobaremos que dirección IP se le ha asignado para poder acceder mediante el navegador.



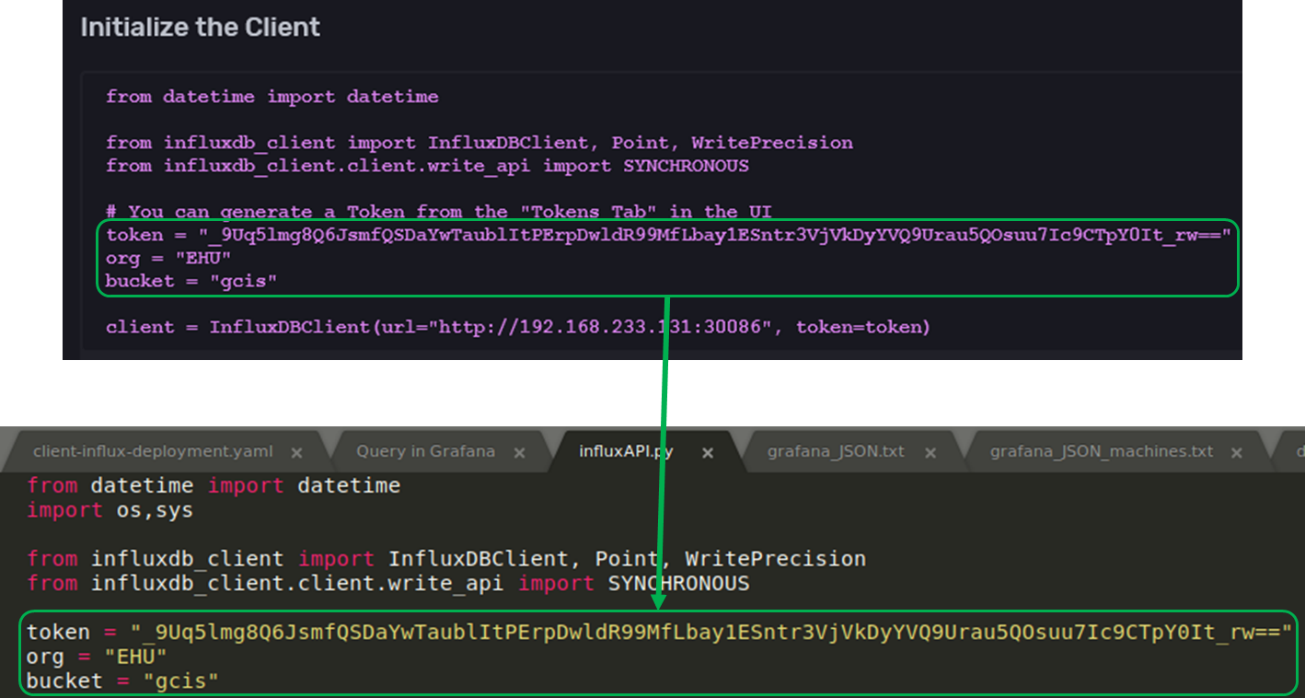
Ahora, accederemos al servicio mediante el navegador. Añadiremos el puerto que hayamos seleccionado en archivo *influx-db-service.yaml*.



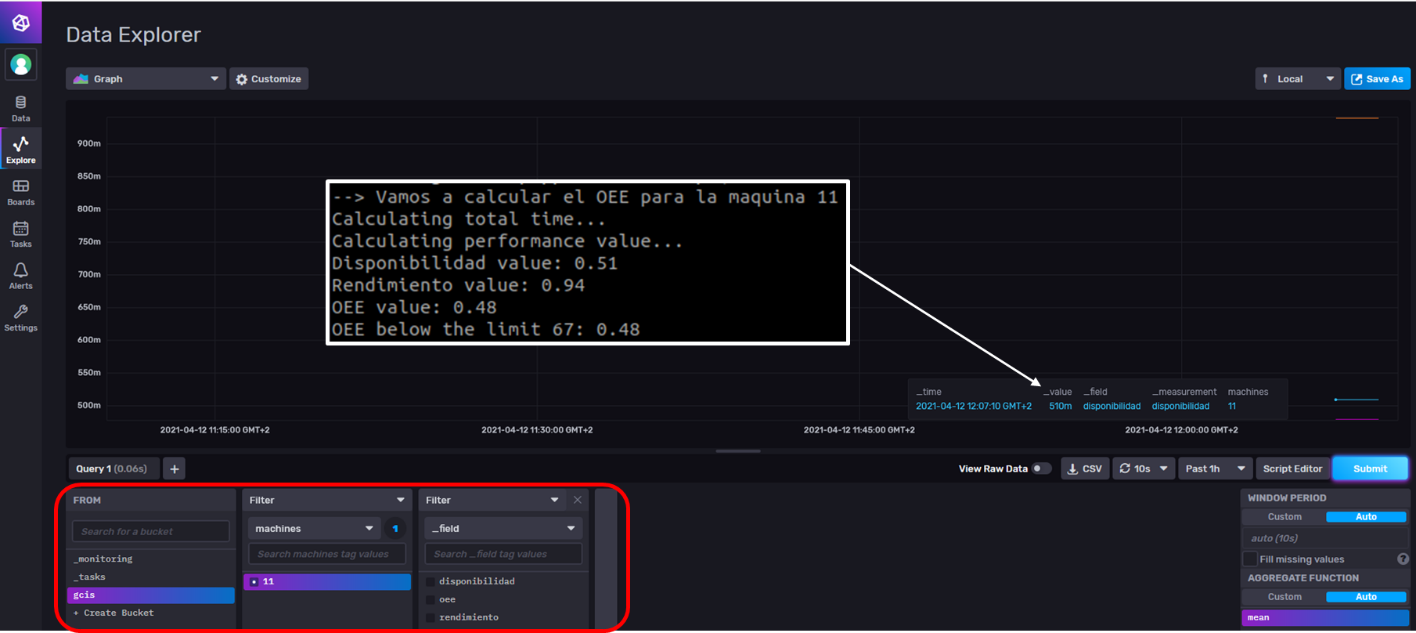
Nos pedirá crear un nuevo usuario. Podremos poner lo que queramos, pero hay que tener en cuenta que estos datos será los que usen los componentes. En este caso, hemos creado el usuario *admingcis*, con la contraseña igual al nombre, y la BBDD con el nombre *gcis*.



La API que guardará los cálculos en Influx necesitará un token para la autentificación. Este token lo conseguiremos en Influx entrando en Data > Python (el lenguaje en el que hemos desarrollado la API). Aquí nos mostrará apartados de código para la inicialización del cliente, escribir datos…



En este punto, ya tendremos la herramienta InfluxDB configurada. Si se almacenan datos y los queremos ver, Influx nos ofrece una interfaz muy sencilla para poder visualizarlos, en el apartado *Explore*.



# Grafana

Grafana es la herramienta para la visualización grafica de datos. Necesitaremos configurar Grafana para que consiga los datos almacenados en Influx y los muestre gráficamente. Para ello, empezaremos como en el apartado anterior, desplegando Grafana en nuestro sistema utilizando su imagen base oficial.

*docker-compose.yml:*

grafana:

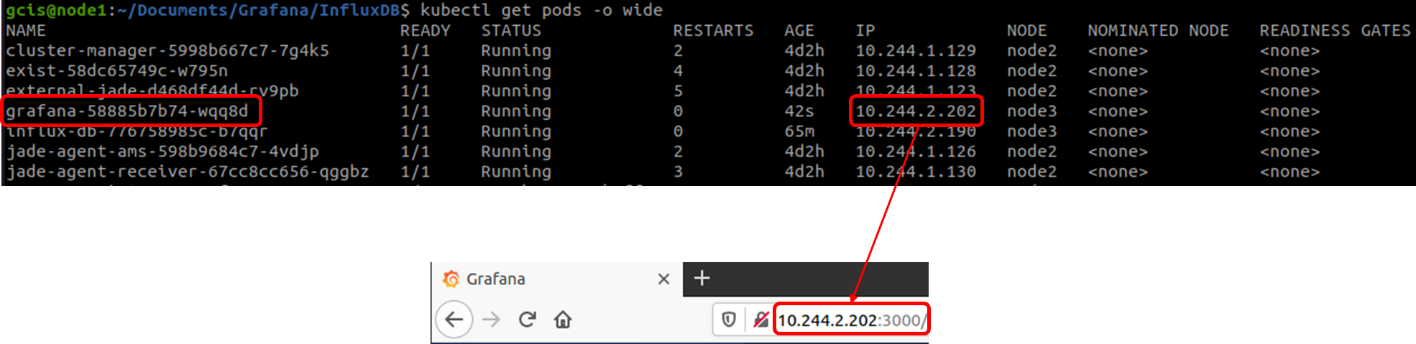
image: grafana/grafana:latest

container\_name: grafana

ports:

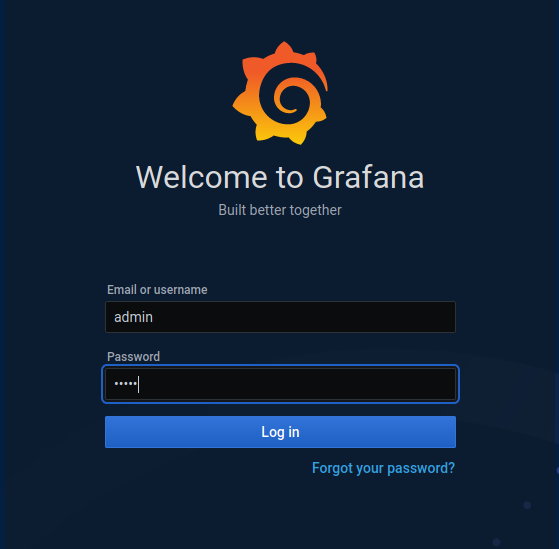
- 3000:3000

Al igual que con Influx, nos fijaremos en su dirección IP para acceder mediante el navegador. Utilizaremos el puerto en el que escucha Grafana: 3000.

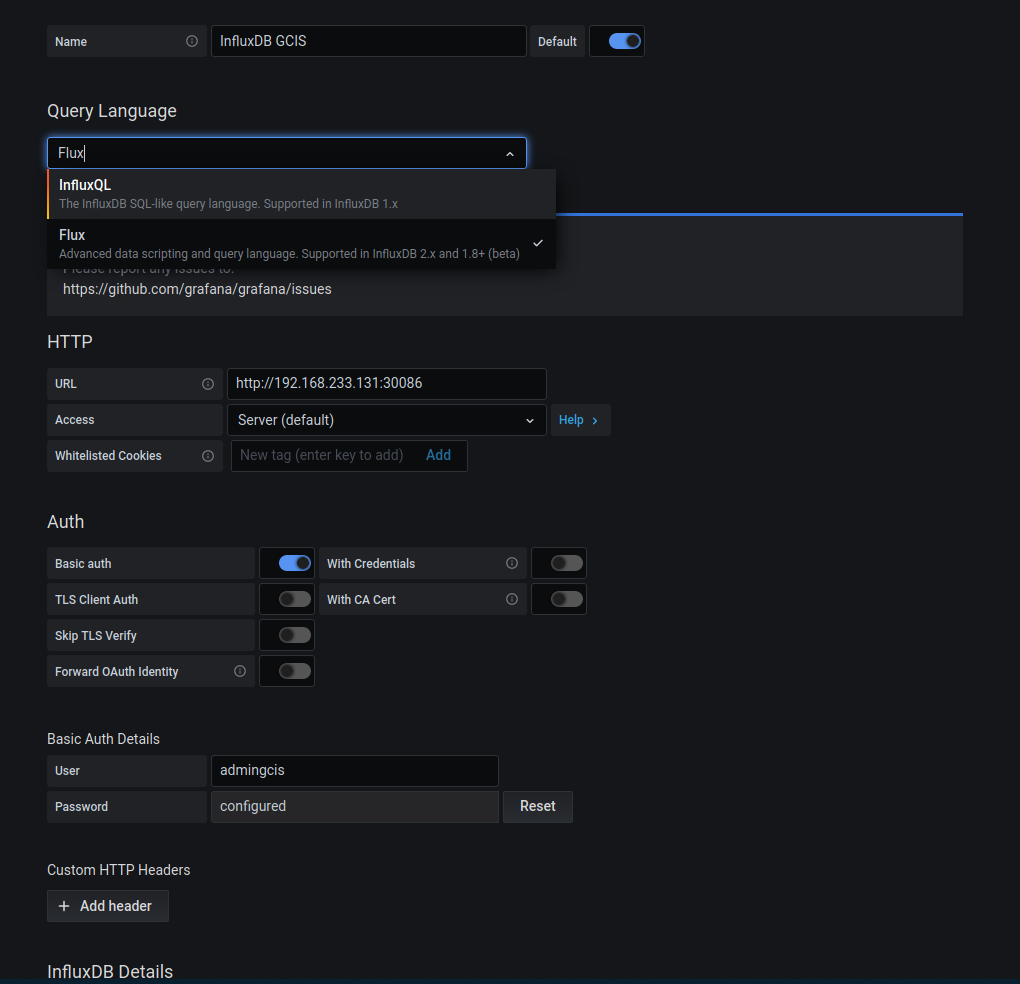


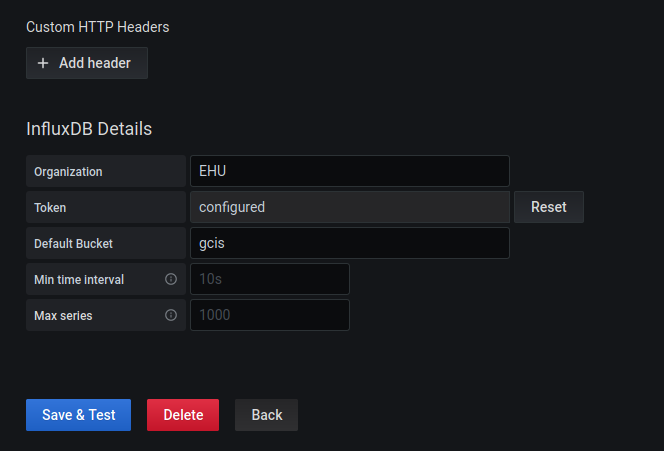
Lo primero nos pedirá autentificarnos. Utilizaremos el usuario de adminisitrador

* user: admin
* password: admin

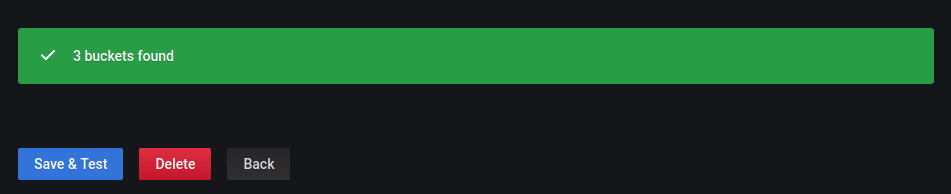


Una vez dentro, configuraremos la conexión con Influx. Para ello, nos iremos al apartado de *Configuration* > *Data Source > Add data source > InfluxDB*. Aquí tendremos que añadir la información de nuestro elemento Influx (dirección IP, usuario, token…). Tendremos que seleccionar la opción de *Flux* en el apartado *Query Language*.





Si hemos metido los datos correctos, nos aparecerá esto:



Ahora, ya podremos leer cualquier dato almacenado en Influx. Para crear las gráficas, utilizaremos los archivos *grafana\_JSON.txt* y *grafana\_JSON\_machines.txt*. Estos son definiciones en JSON de Dashboards de Grafana. Dentro de ellos se ejecutan querys de este tipo:

from(bucket: "gcis")

|> range(start: v.timeRangeStart, stop: v.timeRangeStop)

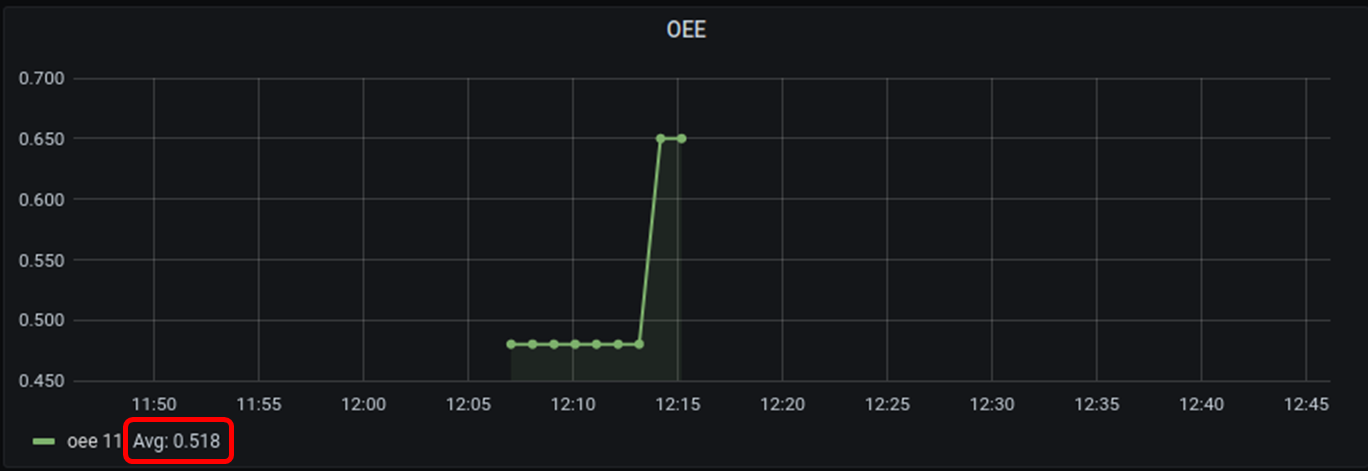
|> filter(fn: (r) => r.machines == "11" and r.\_field == "oee")

Se pueden hacer el tipo de peticiones que se desee, en este caso, se mostraría los cálculos OEE de la maquina 11.

Para añadir estas definiciones en JSON iremos al apartado *Create > Import > Import via panel json*. Aquí, añadiremos el JSON de una maquina en concreto (*grafana\_JSON.txt*) y cambiaremos los que sea necesario: **título** e **id de la máquina**. Así, ya se podrán visualizar los cálculos referidos a la maquina 11. Se crearán 4 graficas: una para reunir los tres valores y una por cada valor.



Como se puede apreciar en las leyendas, en las gráficas individuales también añade el promedio de valor global (average).



También, podremos crear un dashboard para visualizar los cambios de estos cálculos en diferentes maquinas. Esto lo haremos gracias a la definición del archivo *grafana\_JSON\_machines.txt* y repetiremos el mismo proceso que la anterior. Se crearán 3 graficas, cada una con el cálculo pertinente, y los datos de todas las maquinas:

