# **Matomo**

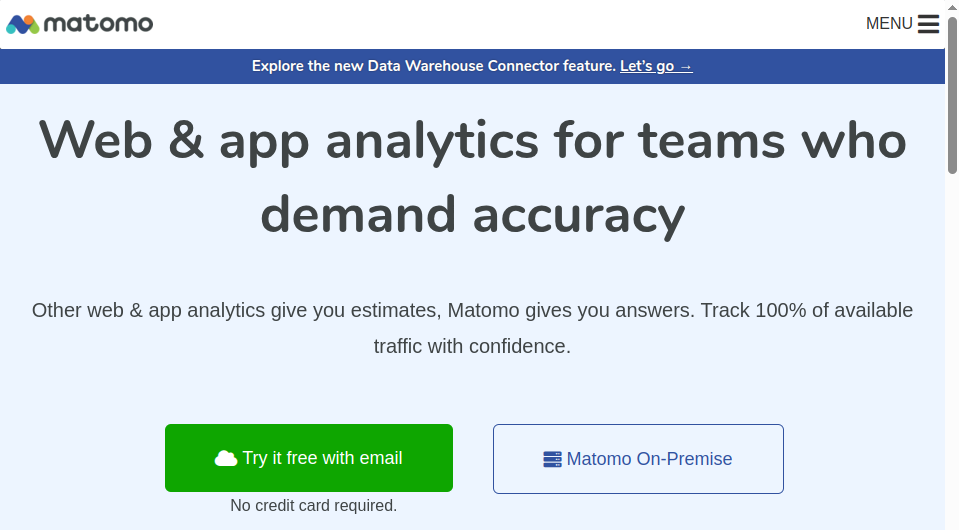
**Matomo** es una plataforma de análisis web de código abierto que permite rastrear y analizar el tráfico de un sitio web sin depender de servicios externos como Google Analytics. Su principal ventaja es que brinda **control total sobre los datos**, ya que puede alojarse en servidores propios, garantizando privacidad y cumplimiento con regulaciones como el **GDPR** (Reglamento General de Protección de Datos).

### **Características principales de Matomo:**

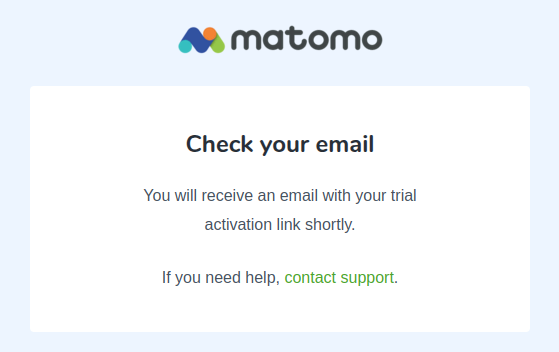
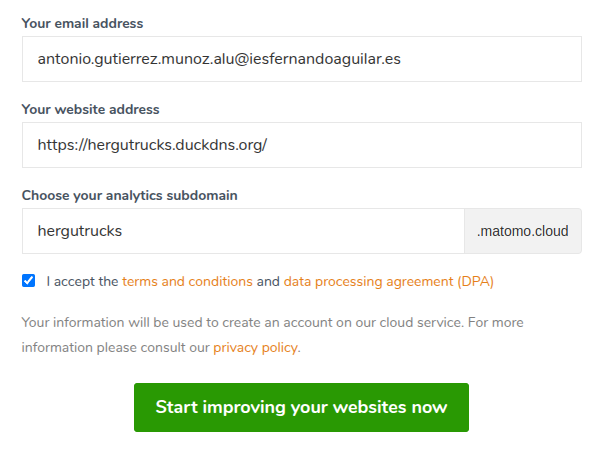
* **Autohospedado o en la nube**: Se puede instalar en un servidor propio o usar la versión en la nube de Matomo.
* **Privacidad y cumplimiento**: No comparte datos con terceros y es compatible con regulaciones de protección de datos.
* **Análisis detallado**: Proporciona métricas sobre visitantes, conversiones, fuentes de tráfico, rendimiento del sitio, entre otros.
* **Personalización**: Admite plugins y módulos adicionales para extender su funcionalidad.
* **Código abierto**: Transparente y flexible, con una comunidad activa de desarrollo.

# **Tutorial: Configuración de Matomo para Analíticas Web**

Accedemos a su pagina web<https://matomo.org/>



Seleccionamos ' Try it free with email' y rellenamos el siguiente formulario:



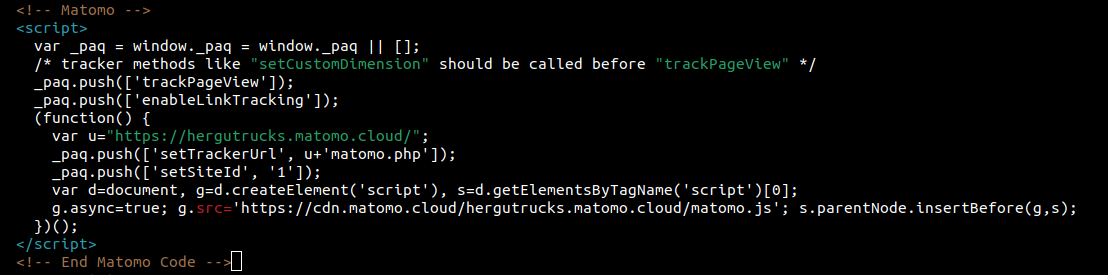
Nos autenticamos y se nos mostrará lo siguiente:



Le daremos a instalar con código JavaScript y seguiremos la guía paso a paso



Primero añadiré el siguiente fragmento de código a mi base.html



Si le damos a probar instalación se verificará la instalación y seremos redireccionados a la siguiente página:

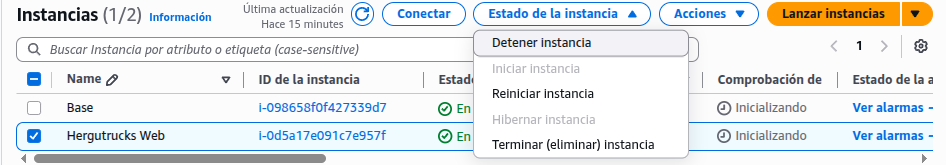


# **MATOMO EN DOCKER**

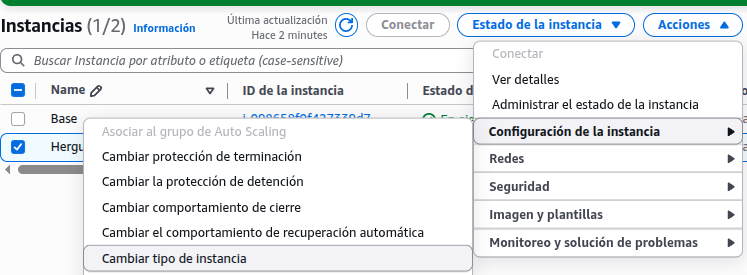
Para aplicar matomo a nuestra aplicación sin depender de sus servicios, lo instalaré tras el proxy inverso donde tenemos la aplicación.

Como Matomo estará corriendo dentro de un contenedor Docker, necesitaremos aumentar los recursos de la instancia para asegurar que haya suficiente espacio, ya que actualmente solo tiene 1 GB. Para lograrlo debemos cambiar el tipo de instancia de t2.micro (actual) a t2.small que tiene 1 GB adicional de memoria RAM, lo que será suficiente para que Matomo funcione sin que la instancia se bloquee.

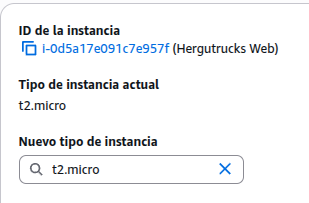
* Detenemos la instancia



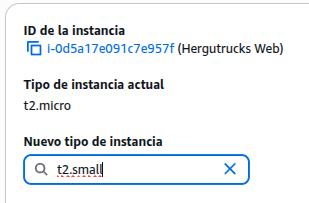
* Una vez detenida clicamos Acciones > Configuración de la instancia > Cambiar tipo de instancia



* Nos aparecerá lo siguiente, debemos clicar sobre el tipo de instancia que tenemos actualmente



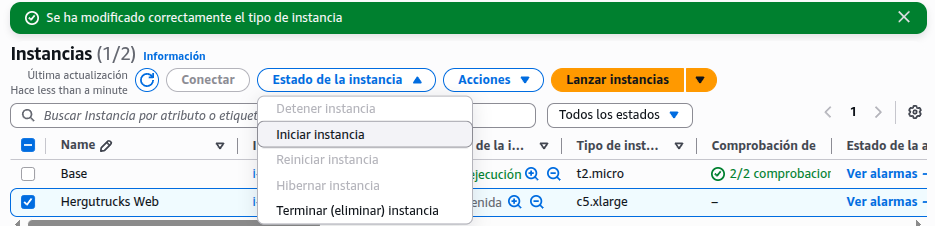
* Escribimos el nombre del nuevo tipo de instancia “t2.small”



* Al seleccionarla, podremos observar como abajo se genera una tabla con la comparación entre el anterior y el nuevo tipo de instancia
* Nos movemos abajo y clicamos en Cambiar



Una vez modificada la instancia, accede a la instancia que aloja mi web, la lanzo de nuevo y una vez sea accesible realizo los siguientes pasos:



Me conecto a la instancia y hago lo siguiente:

sudo apt update

sudo apt install docker.io -y

sudo systemctl enable --now docker

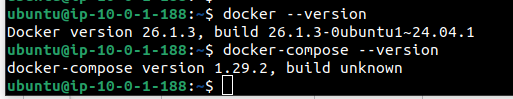
sudo usermod -aG docker $USER

sudo apt install docker-compose -y

#Me aseguro de la instalación

docker --version

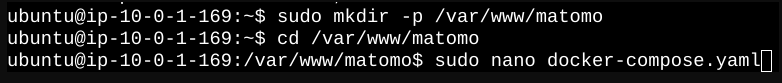
docker-compose --version



Ahora me muevo a var/www/matomo y he puesto un docker-compose.yaml que servirá para ejecutar docker .

sudo mkdir -p /var/www/matomo

cd /var/www/matomo



version: '3'

services:

matomo:

image: matomo

container\_name: matomo

ports:

- "8080:80"

environment:

- MATOMO\_DATABASE\_HOST=db

- MATOMO\_DATABASE\_USERNAME=matomo

- MATOMO\_DATABASE\_PASSWORD=gutimatomo9990

- MATOMO\_DATABASE\_DBNAME=matomo

volumes:

- ./matomo:/var/www/html

depends\_on:

- db

restart: always

db:

image: mariadb

container\_name: matomo\_db

environment:

- MYSQL\_ROOT\_PASSWORD=root

- MYSQL\_DATABASE=matomo

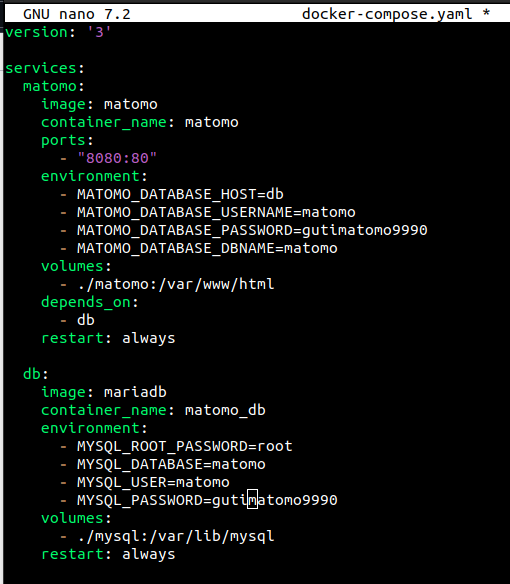
- MYSQL\_USER=matomo

- MYSQL\_PASSWORD=gutimatomo9990

volumes:

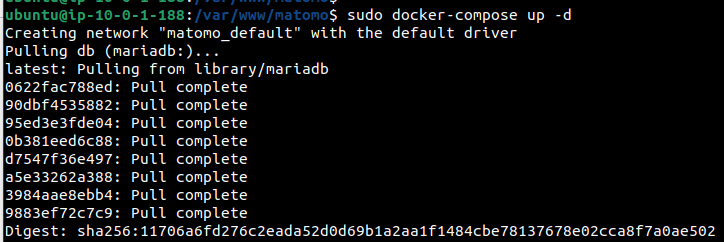
- ./mysql:/var/lib/mysql

restart: always

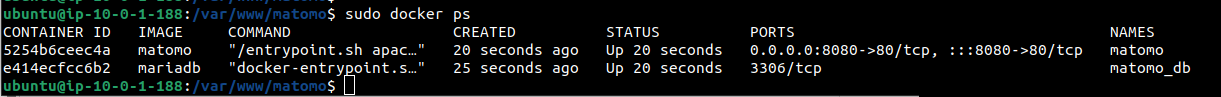


Una vez tengamos el fichero iniciamos los contenedores:

docker-compose up -d

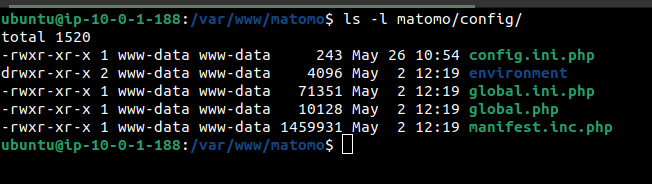


docker ps



Matomo creará automáticamente el archivo config.ini.php después del primer arranque.

ls -l /var/www/matomo/matomo/config/

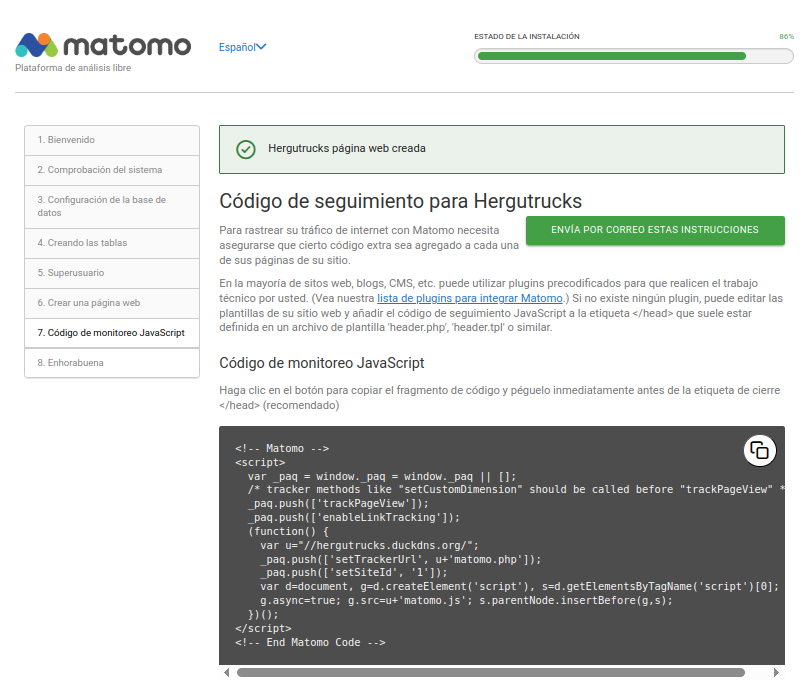
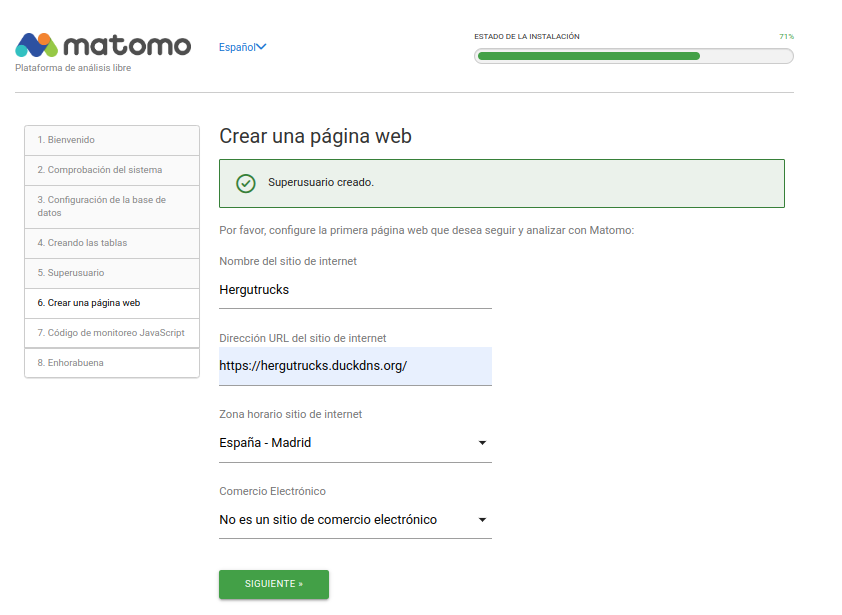
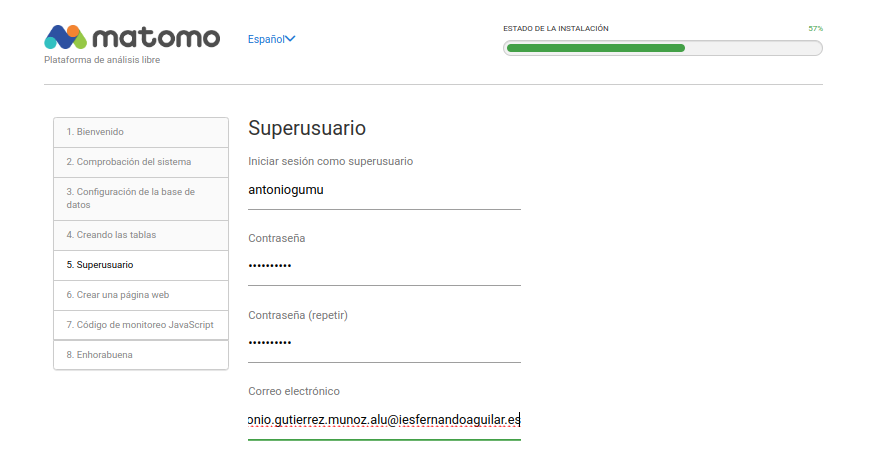
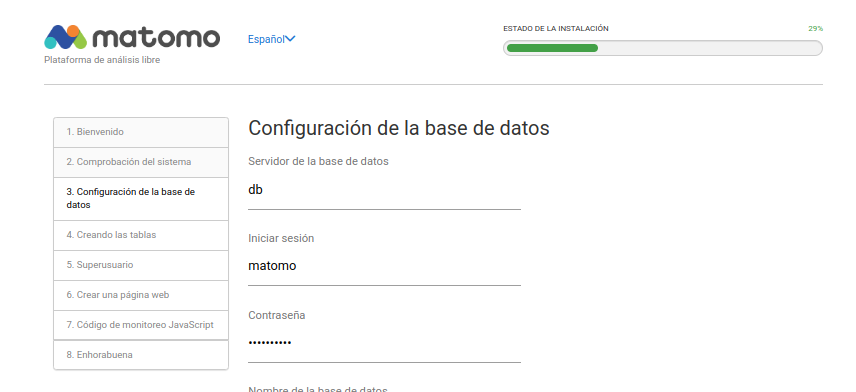


* **Si config.ini.php no existe, ingresa a Matomo en el navegador y sigue el asistente:**
  + Abre tu navegador y ve a http://TU\_IP:8080 o http://TU\_DOMINIO:8080.
  + Completa la configuración inicial de Matomo (te pedirá la base de datos, usa los mismos datos del docker-compose.yaml).
  + Una vez completado, Matomo generará config.ini.php.

En mi caso, no se generó así que accedí de la siguiente forma http://TU\_DOMINIO/matomo/



Simplemente sigo paso a paso



En este paso, se nos ofrece un código que debemos añadir a nuestro base.html como en el ejemplo del principio

<!-- Matomo -->

<script>

var \_paq = window.\_paq = window.\_paq || [];

/\* tracker methods like "setCustomDimension" should be called before "trackPageView" \*/

\_paq.push(['trackPageView']);

\_paq.push(['enableLinkTracking']);

(function() {

var u="//simplexrentalis.duckdns.org:8080/";

\_paq.push(['setTrackerUrl', u+'matomo.php']);

\_paq.push(['setSiteId', '1']);

var d=document, g=d.createElement('script'), s=d.getElementsByTagName('script')[0];

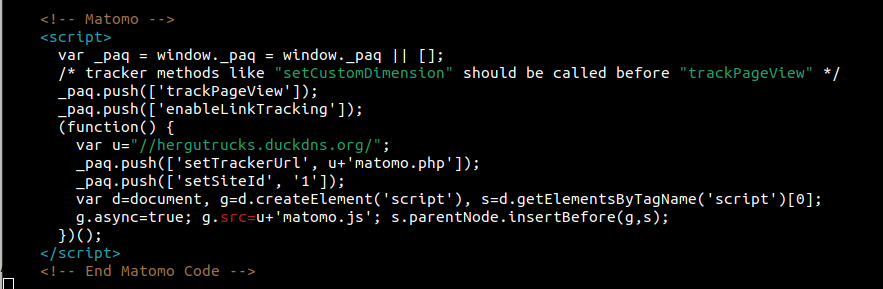
g.async=true; g.src=u+'matomo.js'; s.parentNode.insertBefore(g,s);

})();

</script>

<!-- End Matomo Code -->

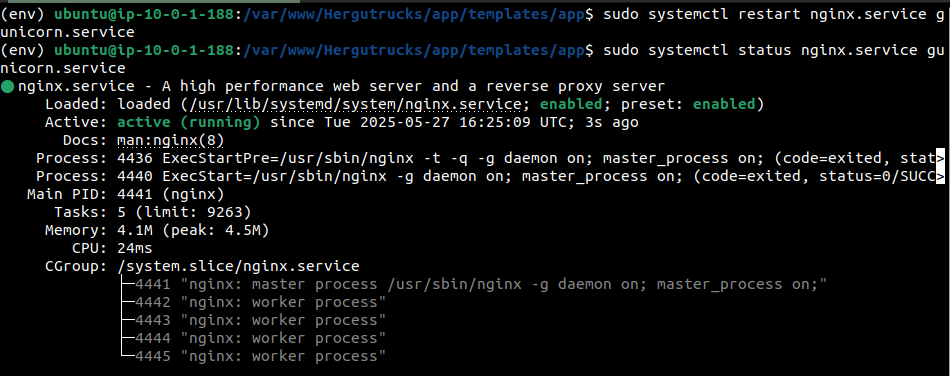
Desde nuestra máquina, editamos el fichero base.html y añadimos este código en la cabecera <head>



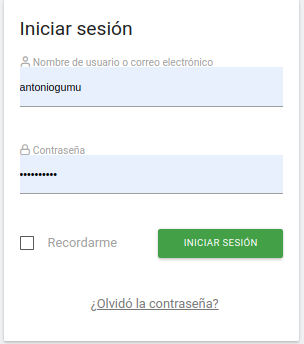
Es importante realizar la siguiente modificación y añadir http: al código que hemos copiado



Una vez terminado, guardamos el archivo y recargamos los servicios de nuestra aplicación para que se detecten los cambios

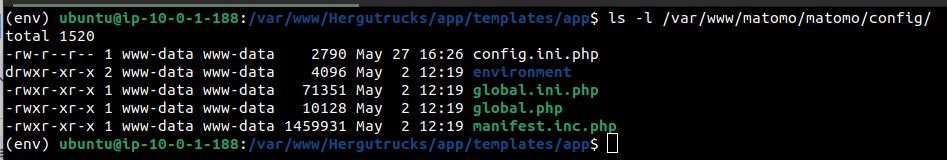


Ahora podemos terminar con la configuración desde la web



Una vez hecho todo esto, se debería de haber generado ya con éxito todos los ficheros de configuración de matomo:

ls -l /var/www/matomo/matomo/config/



Dentro de /var/www/matomo/matomo/config/config.ini.php agregamos las siguientes lineas:

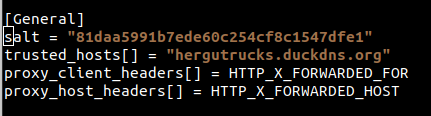
[General]

salt = "xxxxxxxxxxx"

trusted\_hosts[] = "xxxxxxxxxxx"

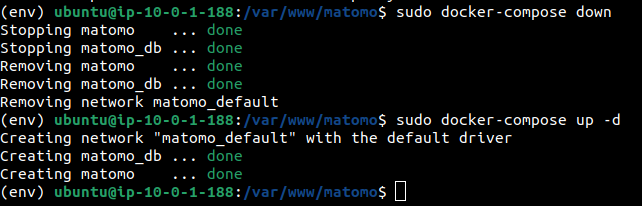
proxy\_client\_headers[] = HTTP\_X\_FORWARDED\_FOR

proxy\_host\_headers[] = HTTP\_X\_FORWARDED\_HOST



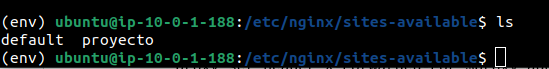
Los parámetros proxy\_client\_headers[] = HTTP\_X\_FORWARDED\_FOR y proxy\_host\_headers[] = HTTP\_X\_FORWARDED\_HOST son necesarios para poder registrar correctamente las direcciones IP reales de los usuarios cuando la aplicación está detrás de un proxy inverso.

Relanzo el docker-compose para reiniciar los contenedores y asi aplicar los cambios realizados en el anterior fichero



Una vez guardado el fichero anterior nos movemos a /etc/nginx/sites-available/default aquí agregamos lo siguiente al fichero de configuración de nuestra aplicación

cd /etc/nginx/sites-available/



Agregamos lo siguiente:

location /matomo/ {

proxy\_pass http://127.0.0.1:8080/;

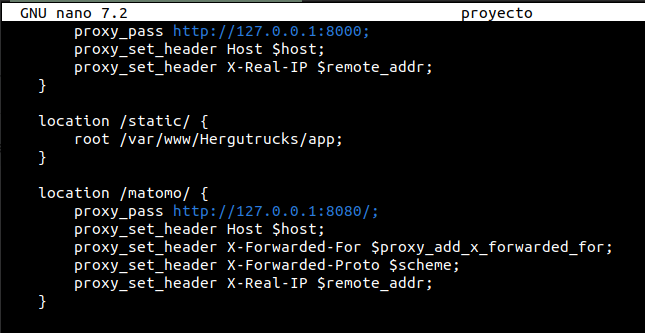
proxy\_set\_header Host $host;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

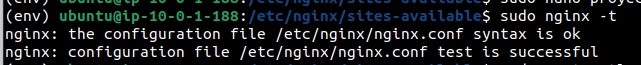
proxy\_set\_header X-Real-IP $remote\_addr;

}

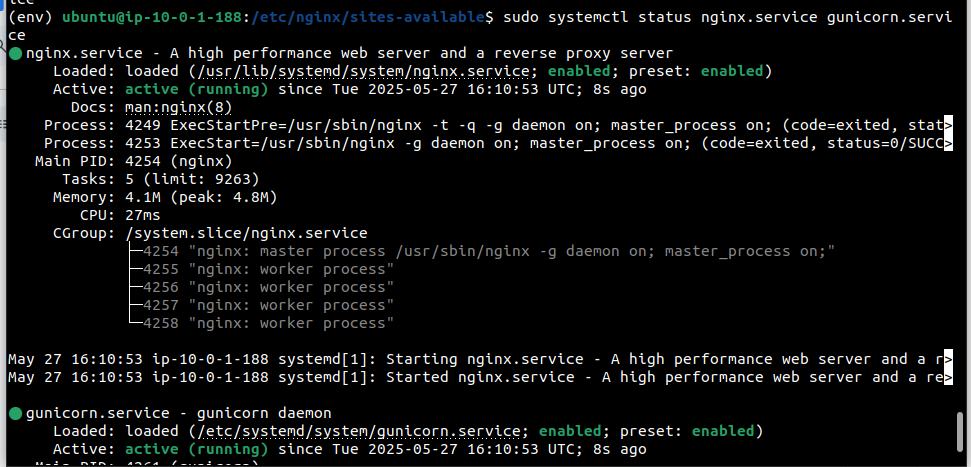


* **proxy\_pass**: Redirige las solicitudes al backend que está escuchando en http://127.0.0.1:8080.
* **proxy\_set\_header Host $host**: Se asegura de que el encabezado Host sea pasado correctamente al servidor backend.
* **proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for**: Añade la dirección IP del cliente original a la cabecera X-Forwarded-For para que el servidor backend pueda saber la dirección IP real del cliente.
* **proxy\_set\_header X-Forwarded-Proto $scheme**: Pasa el esquema (http o https) de la solicitud original al servidor backend.
* **proxy\_set\_header X-Real-IP $remote\_addr**: Pasa la dirección IP del cliente que realizó la solicitud.

Ahora hacemos un sudo nginx -t para ver que todo se encuentre correcto

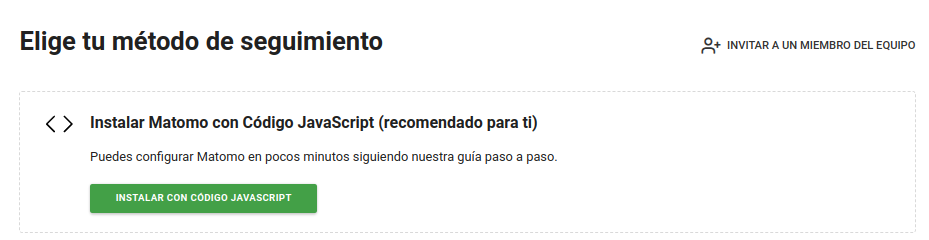


Vamos de nuevo a reiniciar los servicios de nuestra aplicación para aplicar los últimos cambios



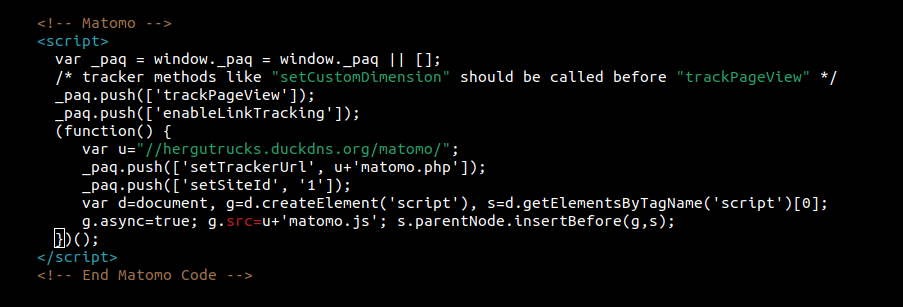
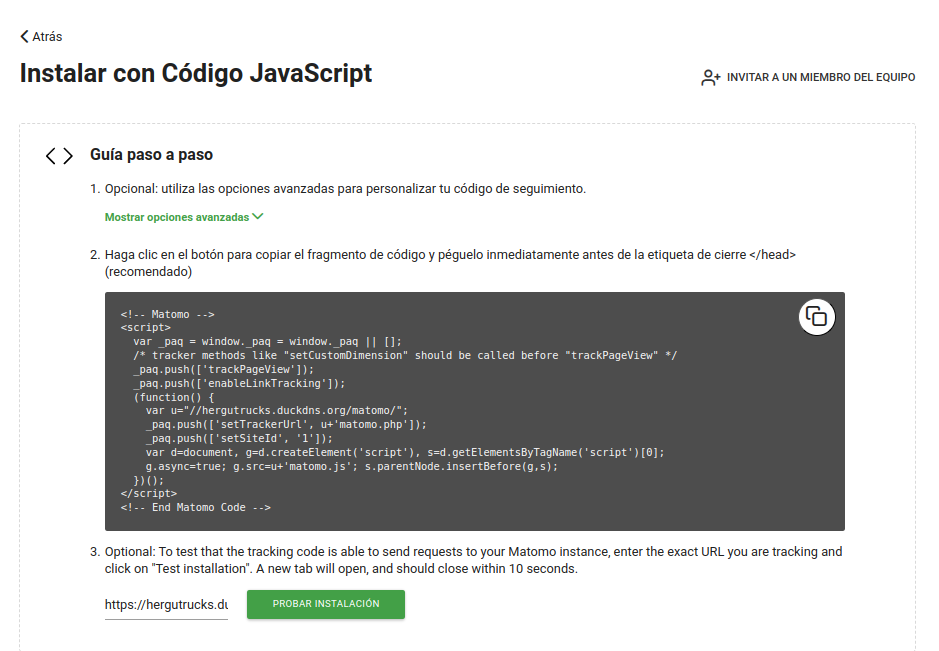
Al confirmar esto, nos moveremos a nuestro navegador y entramos en <nuestro dominio>/matomo

Una vez aquí, probamos la configuración dándole a instalar con código de javascript

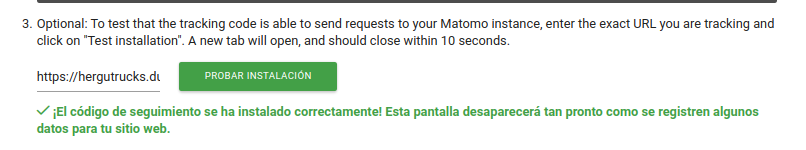


Aunque pensemos que tenemos el mismo código, hay que tener en cuenta que usamos http para poder acceder a la configuración inicial, así que debemos editar el código existente en el base html y eliminar http: en la siguiente línea:

(Recomiendo eliminar el segmento de código y reemplazarlo por el que nos ofrece matomo)



Una vez reemplazado el código, comprobamos la instalación y listo



Si volvemos a nuestro dominio y volvemos a escribir <nuestro dominio>/matomo , esta vez se nos redireccionará a la siguiente página:

