

# 3.16 Adguard. Navega por Internet sin anuncios y con seguridad



*Imagen obtenida de <https://adguard.com/es/welcome.html>*

## Esta herramienta sirve para...

como dicen en su web:

“

- Bloquea todo tipo de anuncios
- Elimina elementos web molestos
- Ahorra tráfico y acelera carga de páginas
- Funciona para navegadores y aplicaciones
- Mantiene la funcionalidad y la apariencia de sitios

# Web de proyecto y otros enlaces de interés

Web del proyecto <https://adguard.com/es/welcome.html>

Repositorio de código: <https://github.com/AdguardTeam/AdGuardHome>

Imagen docker oficial: <https://hub.docker.com/r/adguard/adguardhome>

## Despliegue

En esta ocasión he tenido muchas dudas sobre como recomendaros el despliegue del servicio pues es posible la instalación de un modo muy sencillo a través de cualquiera de estas 3 opciones

- vía **script**: simplemente ejecutando: `curl -s -S -L https://raw.githubusercontent.com/AdguardTeam/AdGuardHome/master/scripts/install.sh | sh -s -- -v`
- vía **snap** (sistemas linux): en la snap store desde este enlace <https://snapcraft.io/adguard-home>
- vía **docker**

Por mantener cierta coherencia con lo trabajado en el curso voy a optar por seguir trabajando con docker y docker-compose pero lo cierto es que fuera del curso había sido mi 3ª elección debido a la simplicidad de los otros 2 sistemas. Vamos allá:

Accedemos a la terminal y escribimos

```
cd $HOME
mkdir adguard
cd adguard
nano docker-compose.yml
```

Dentro del fichero escribimos el siguiente contenido

```
version: '3.3'
services:
  adguardhome:
    image: adguard/adguardhome
```

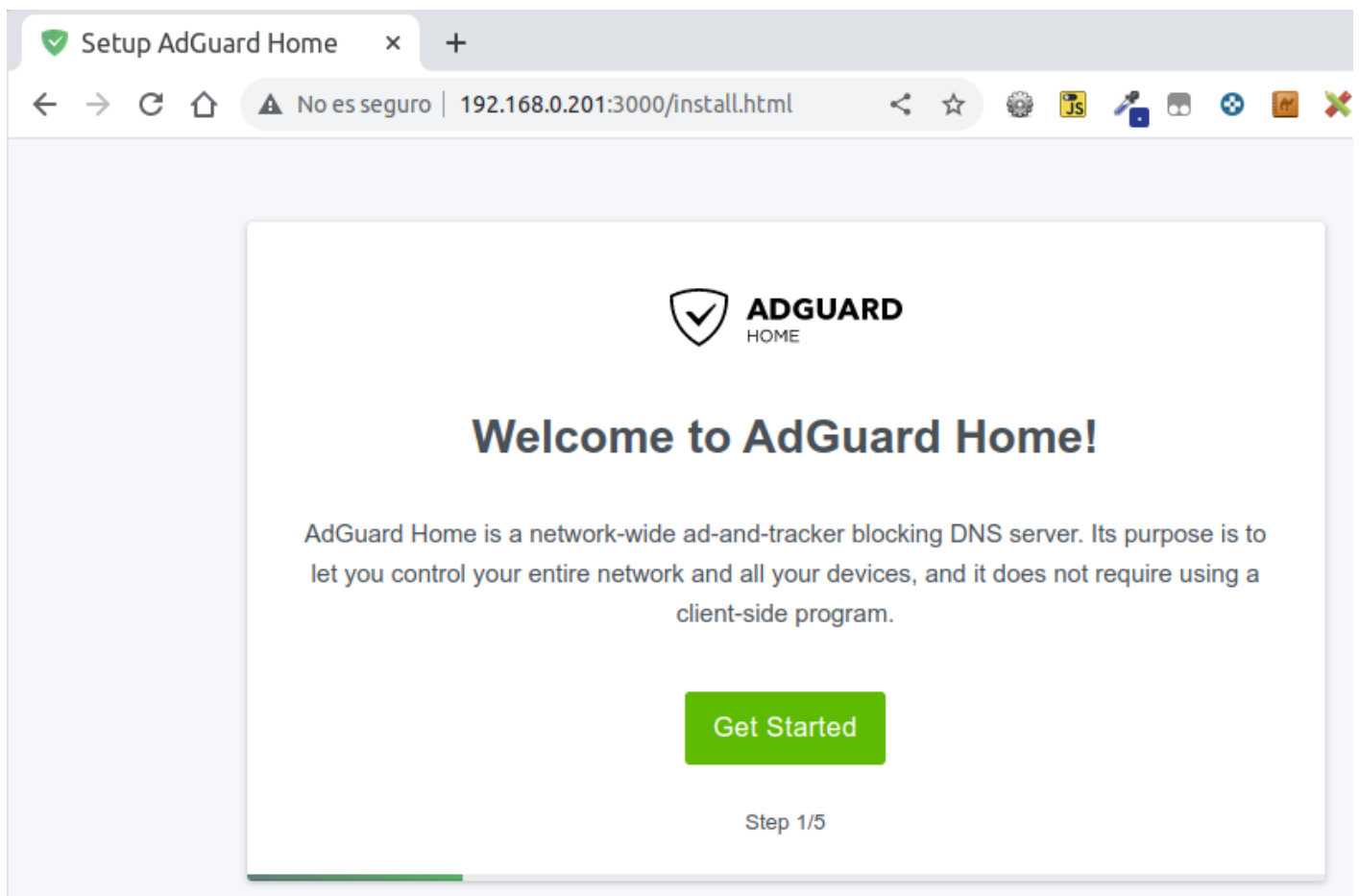
```
container_name: adguardhome
restart: unless-stopped
volumes:
  - './workdir:/opt/adguardhome/work'
  - './confdir:/opt/adguardhome/conf'
ports:
  - '53:53/tcp'
  - '53:53/udp'
  - '67:67/udp'
  - '80:80/tcp'
  - '443:443/tcp'
  - '443:443/udp'
  - '3000:3000/tcp'
  - '853:853/tcp'
  - '784:784/udp'
  - '853:853/udp'
  - '8853:8853/udp'
  - '5443:5443/tcp'
  - '5443:5443/udp'
```

Para salir del fichero pulsaremos `control + x` y guardaremos los cambios.

Si algún otro servicio está utilizando los puertos que en este servicio vamos a utilizar se generará un conflicto y puede que ninguno de los servicios funcione o, mas probable, el último que pongamos en marcha.

Para usar un puerto diferente puedes cambiar el valor que aparece ANTES de los : por un valor que no esté en uso. También puedes acceder al directorio dónde se encuentra el otro servicio y ejecutar `docker-compose down`.

Posteriormente ponemos en marcha los contenedores con `docker compose up -d`. Si accedemos, como en ocasiones anteriores, a `http://IP:PUERTO` siendo en mi caso <http://192.168.0.201:3000> veremos algo como:



*Elaboración propia*

## Funcionamiento

Una vez hemos accedido a la pantalla anterior es momento de configurar el servicio a través de la interface gráfica. A través de 5 pasos estableceremos los valores de configuración. No es necesario cambiar nada de la configuración. Si que deberás establecer un usuario y contraseña de acceso. En el paso 4 te dirá como debes configurar tus dispositivos para que adguard pueda hacer su tarea (si ya los tenías configurados con pi-hole no deberás hacer nada adicional). Una vez configurado la interface gráfica te pedirá el usuario y contraseña que has establecido y accederás a un panel de control como el siguiente:



Username

admin

Password

.....

Sign In

[Forgot password?](#)

Please follow [these steps](#) to create a new password  
for your user account.

*Elaboración propia*

Revision #6

Created 14 February 2023 21:30:16 by Pablo Ruiz

Updated 20 July 2023 18:10:48 by Pablo Ruiz