

# How to Install Wg-Easy - An Opensource Web UI for WireGuard VPN

Learn how to install Wg-Easy, a user-friendly open-source web interface for WireGuard VPN that simplifies configuration and management for your network.

---

# Contents

01	Introduction	3
02	Prerequisites	3
03	Install Wg-Easy	3
04	Install Nginx Proxy Manager to Secure the Wg-Easy Web Management Interface	9
05	Access Wg-Easy	16
06	Conclusion	22

# Introduction

Wg-Easy also known as WireGuard-easy is an open-source web-based graphical interface for WireGuard VPN to manage configurations, clients, and VPN connections. Wg-Easy includes all required WireGuard tools used to create multiple VPN interfaces and client configurations on a server.

This article explains how to install Wg-Easy on Ubuntu 24.04 to create and manage WireGuard VPN connections.

## Prerequisites

Before you begin:

- Deploy an [Ubuntu 24.04 instance](#) on Vultr and enable the [Limited User Login](#) feature
- [Set up a new domain A record pointing to the instance's IP address](#). For example, `wg-easy.example.com`.
- Access the instance using [SSH](#).
- Install [Docker and Docker Compose](#).

## Install Wg-Easy

Wg-Easy runs as a Docker container to manage WireGuard VPN interfaces, users, and connections on a server. Follow the steps below to verify that Docker is installed, add your current user to the Docker group, and install Wg-Easy.

1. View the Docker service status and verify that it's running.

CONSOLE

```
$ sudo service docker status
```

Output:

```
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   enabled)
   Active: active (running) since Wed 2024-12-04 13:09:32 UTC; 15min ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
   Main PID: 2074 (dockerd)
     Tasks: 7
    Memory: 19.4M
       CPU: 191ms
    CGroup: /system.slice/docker.service
           └─2074 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/
           containerd.sock
```

If you receive a `docker service not found` error, run the following command to install Docker and Docker compose.

CONSOLE

```
$ sudo apt install docker.io docker-compose
```

If the Docker service is inactive, run the following command to start Docker.

CONSOLE

```
$ sudo service docker start
```

2. Print your current user.

CONSOLE

```
$ whoami
```

Your output should be similar to the one below:

```
linuxuser
```

3. Add your active user to the Docker group. Replace `linuxuser` with your actual user.

CONSOLE

```
$ sudo usermod -aG docker linuxuser
```

4. Enter a new shell to apply the user group changes.

CONSOLE

```
$ exec su - $USER
```

5. Create a new administrator password using the `htpasswd` utility with a bcrypt hash value to use with Wg-Easy.

CONSOLE

```
$ htpasswd -nbB admin strongpassword
```

Copy the generated hashed password in your output similar to the one below.

```
admin:$2y$05$R0ESp.kfeIwWHyRkV0XXEu/xKCXq03hTRxr4Y8ppxj6jtaiEuJ7Su
```

## Option 1: Install Wg-Easy using Docker CLI

Docker CLI installs Wg-Easy in a single command with minimal environment configuration options. Follow the steps below to install Wg-Easy using Docker CLI on your server.

1. Pull the Wg-Easy image.

CONSOLE

```
$ docker pull ghcr.io/wg-easy/wg-easy
```

2. Install Wg-Easy using Docker CLI. Replace `wg-easy.example.com` with your actual domain and add your hashed password as the `PASSWORD_HASH` value.

## CONSOLE

```
$ docker run --detach \  
  --name wg-easy \  
  --env LANG=en \  
  --env WG_HOST=wg-easy.example.com \  
  --env PASSWORD_HASH='$2y$05$R0ESp.kfeIwWHyRkV0XXEu/  
xKCXq03hTRxr4Y8ppxj6jttaiEuJ7Su' \  
  --env PORT=51821 \  
  --env WG_PORT=51820 \  
  --volume ~/.wg-easy:/etc/wireguard \  
  --publish 51820:51820/udp \  
  --publish 51821:51821/tcp \  
  --cap-add NET_ADMIN \  
  --cap-add SYS_MODULE \  
  --sysctl 'net.ipv4.conf.all.src_valid_mark=1' \  
  --sysctl 'net.ipv4.ip_forward=1' \  
  --restart unless-stopped \  
  ghcr.io/wg-easy/wg-easy
```

Save and close the file.

The above Docker CLI command creates a new `wg-easy` container to manage WireGuard connections on the server. Within the Docker CLI command:

- `--name wg-easy`: Sets the Docker container name.
- `--env LANG=en`: Enables English as the default language in the Wg-Easy web management interface.
- `--env WG_HOST=wg-easy.example.com`: Enables Wg-Easy to listen for network connections using the specified domain or server IP address.
- `--env PASSWORD_HASH`: Sets the administrator password used to access the Wg-Easy interface.
- `--env PORT=51821`: Sets the Wg-Easy port used to access the web management interface.
- `--env WG_PORT=51820`: Sets the default WireGuard connections port.

- `--volume ~/.wg-easy:/etc/wireguard` : Forwards WireGuard configurations in the `wg-easy` directory in your user home directory to the container's `/etc/wireguard` directory.
- `--sysctl 'net.ipv4.ip_forward=1'` : Enables network forwarding to allow WireGuard clients to access the Internet and other external networks through the server.

3. View all active Docker containers and verify that the `wg-easy` container is running.

#### CONSOLE

```
$ docker ps
```

Output:

CONTAINER ID	IMAGE	COMMAND	CREATED
6224cb5009fe	ghcr.io/wg-easy/wg-easy	"docker-entrypoint.s..."	7 seconds ago
		Up 6 seconds (health: starting)	0.0.0.0:51820->51820/udp, :::51820->51820/udp, 0.0.0.0:51821->51821/tcp, :::51821->51821/tcp
		wg-easy	

## Option 2: Install Wg-Easy using Docker Compose

Docker Compose allows you to customize and install Wg-Easy with advanced configurations including the forwarding of specific files such as WireGuard configurations from the server to the `wg-easy` container. Follow the steps below to install Wg-Easy using Docker Compose.

1. Switch to your user's home directory.

#### CONSOLE

```
$ cd
```

2. Create a new `wg-easy.yml` configuration file using a text editor such as `nano`.

## CONSOLE

```
$ nano wg-easy.yml
```

3. Add the following configurations to the file. Replace the hash password with your actual password.

## YAML

```
volumes:
  etc_wireguard:

services:
  wg-easy:
    environment:
      - LANG=en
      - WG_HOST=wg-easy.example.com
      - PASSWORD_HASH=$2y$05$ROESp.kfeIwWHyRkVOXXEu/
xKCXq03hTRxr4Y8ppxj6jtaiEuJ7Su
      - PORT=51821
      - WG_PORT=51820
      - WG_CONFIG_PORT=92820

    image: ghcr.io/wg-easy/wg-easy
    container_name: wg-easy
    volumes:
      - etc_wireguard:/etc/wireguard
    ports:
      - "51820:51820/udp"
      - "51821:51821/tcp"
    restart: unless-stopped
    cap_add:
      - NET_ADMIN
      - SYS_MODULE
    sysctls:
      - net.ipv4.ip_forward=1
      - net.ipv4.conf.all.src_valid_mark=1
```

Save the file.

The above Docker Compose configuration installs Wg-Easy with specific environment options and forwards configuration files from the `/etc/`

wireguard directory on the server to the `/etc/wireguard` in the Wg-Easy container.

4. Apply the Docker Compose configuration to install and run Wg-Easy in detached mode.

#### CONSOLE

```
$ docker-compose -f wg-easy.yml up -d
```

Output:

```
Creating wg-easy ... done
```

5. View all active Docker containers and verify that the Wg-Easy container is running.

#### CONSOLE

```
$ docker ps
```

Output:

CONTAINER ID	IMAGE	COMMAND
1b9da56f5109	ghcr.io/wg-easy/wg-easy	"docker-entrypoint.s..."
CREATED	STATUS	
21 seconds ago	Up 20 seconds (health: starting)	0.0.0.0:51820->51820/udp, :::51820->51820/udp, 0.0.0.0:51821->51821/tcp, :::51821->51821/tcp
PORTS	NAMES	wg-easy

## Install Nginx Proxy Manager to Secure the Wg-Easy Web Management Interface

Nginx Proxy Manager is a reverse proxy application that forwards external connections to internal applications or services in a Docker environment. Follow the steps below to install Nginx Proxy Manager and secure the Wg-Easy web

management interface using your `wg-easy.example.com` domain and generate trusted SSL certificates on the server.

1. Create a new `nginx-proxy.yml` configuration file.

```
CONSOLE
$ nano nginx-proxy.yml
```

2. Add the following configurations to the `nginx-proxy.yml` file.

```
YAML
version: '3.8'
services:
  app:
    image: 'jc21/nginx-proxy-manager:latest'
    container_name: nginx-proxy-man
    restart: unless-stopped
    ports:
      - '80:80'
      - '443:443'
      - '81:81'
    volumes:
      - ./data:/data
      - ./letsencrypt:/etc/letsencrypt
```

Save and close the file.

The above Docker Compose configuration installs Nginx Proxy Manager to manage network connections using HTTP and HTTPS ports. Within the configuration:

- `image: 'jc21/nginx-proxy-manager:latest'`: Sets the Nginx Proxy Manager Docker image version to install.
- `container_name: nginx-proxy-man`: Sets the Nginx Proxy Manager container name for identification and management purposes.
- `- '80:80'`: Forwards the container port `80` to the server port `80` to enable HTTP network connections.
- `- '443:443'`: Forwards the container port `443` to the server port `443` to enable HTTPS network connections.

- - '81:81': Forwards the container port 81 to the server port 81 to enable access to the Nginx Proxy Manager web management interface.
- ./data:/data: Mounts the data directory from the server to the container.
- - ./letsencrypt:/etc/letsencrypt: Mounts Let's Encrypt SSL certificates from the active project directory to the container.

### 3. Apply the Docker Compose configuration to install Nginx Proxy Manager.

#### CONSOLE

```
$ docker-compose -f nginx-proxy.yml up -d
```

#### Output:

```
Creating nginx-proxy-man ... done
```

### 4. View all active Docker containers and verify that Nginx Proxy Manager is running.

#### CONSOLE

```
$ docker ps
```

#### Output:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
45b5f402e197	jc21/nginx-proxy-manager:latest	"/init"	50 seconds ago	Up 48 seconds	0.0.0.0:80-81->80-81/tcp, :::80-81->80-81/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp	nginx-proxy-man
1b9da56f5109	ghcr.io/wg-easy/wg-easy	"docker-entrypoint.s..."	3 minutes ago	Up 3 minutes (healthy)	0.0.0.0:51820->51820/udp, :::51820->51820/udp, 0.0.0.0:51821->51821/tcp, :::51821->51821/tcp	wg-easy

5. Create a new Docker network to connect the Nginx Proxy Manager container to Wg-Easy.

CONSOLE

```
$ docker network create wg-easy
```

6. List all Docker networks and verify that the `wg-easy` network is available.

CONSOLE

```
$ docker network ls
```

Output:

NETWORK ID	NAME	DRIVER	SCOPE
f0f272b127cc	bridge	bridge	local
e5fc3caac88f	host	host	local
be3fd932daf7	none	null	local
d531ed105cde	root_default	bridge	local
1782969ddb95	wg-easy	bridge	local

7. Attach the Nginx Proxy Manager container to the `wg-easy` network.

CONSOLE

```
$ docker network connect wg-easy nginx-proxy-man
```

8. Attach the Wg-Easy container to the `wg-easy` network.

CONSOLE

```
$ docker network connect wg-easy wg-easy
```

## Configure Nginx Proxy Manager as a Reverse Proxy Manage Connections to Wg-Easy

Nginx Proxy Manager generates SSL certificates using Let's Encrypt and forwards network connections to internal ports in a Docker network. Follow the steps below to access the Nginx Proxy Manager interface, forward connections to the Wg-Easy container, and generate trusted SSL certificates using your `wg-easy.example.com` domain.

1. Allow network connections to the HTTP port `80`, HTTPS port `443`, and Nginx Proxy Manager port `81` through the default firewall.

CONSOLE

```
$ sudo ufw allow 80,443,81/tcp
```

2. Reload UFW to apply the firewall configuration changes.

CONSOLE

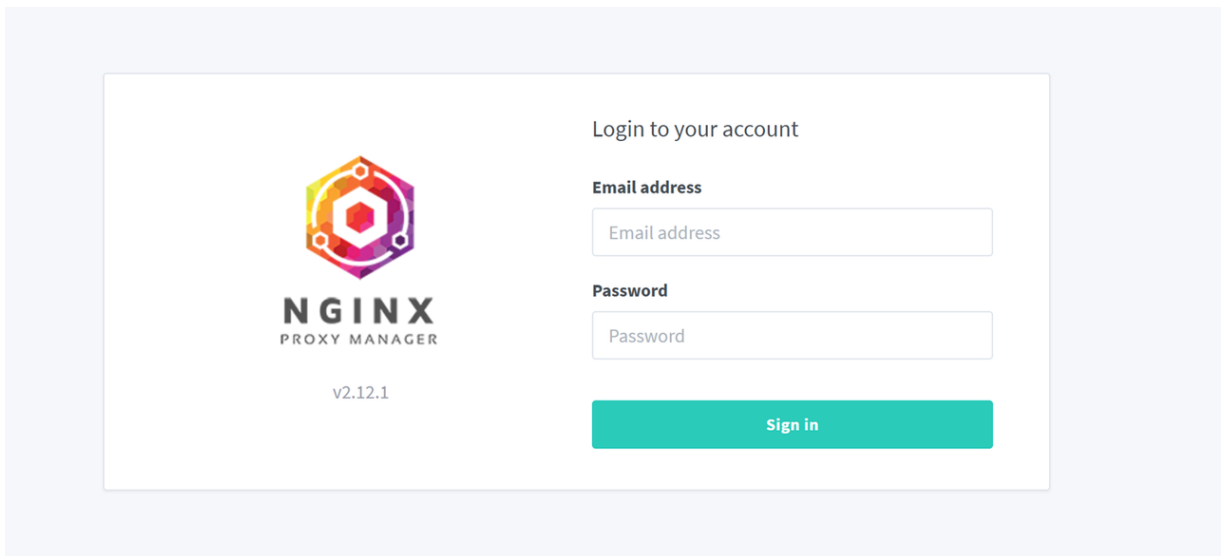
```
$ sudo ufw reload
```

3. Access the Nginx Proxy Manager port `81` using your server's IP address in a web browser such as Chrome.

```
http://SERVER-IP:81
```

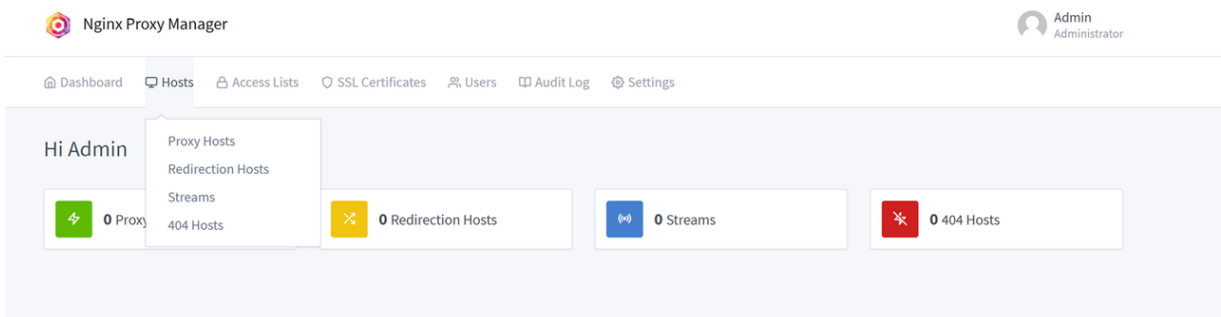
4. Enter the following administrator credentials to log in to Nginx Proxy Manager.

- Username: `admin@example.com`
- Password: `changeme`

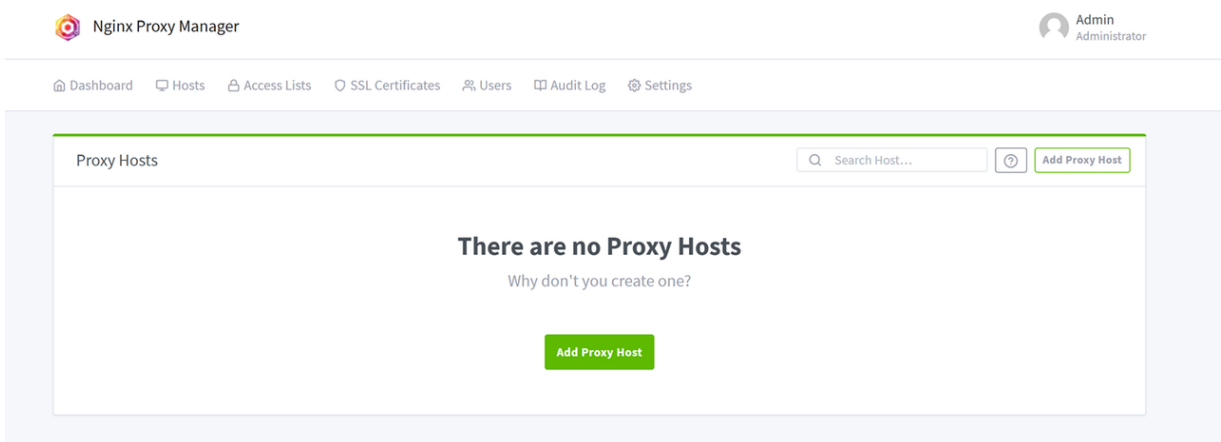


Replace the default administrator credentials when prompted to secure Nginx Proxy Manager.

5. Click **Hosts** on the main navigation menu and select **Proxy Hosts** from the list of options.

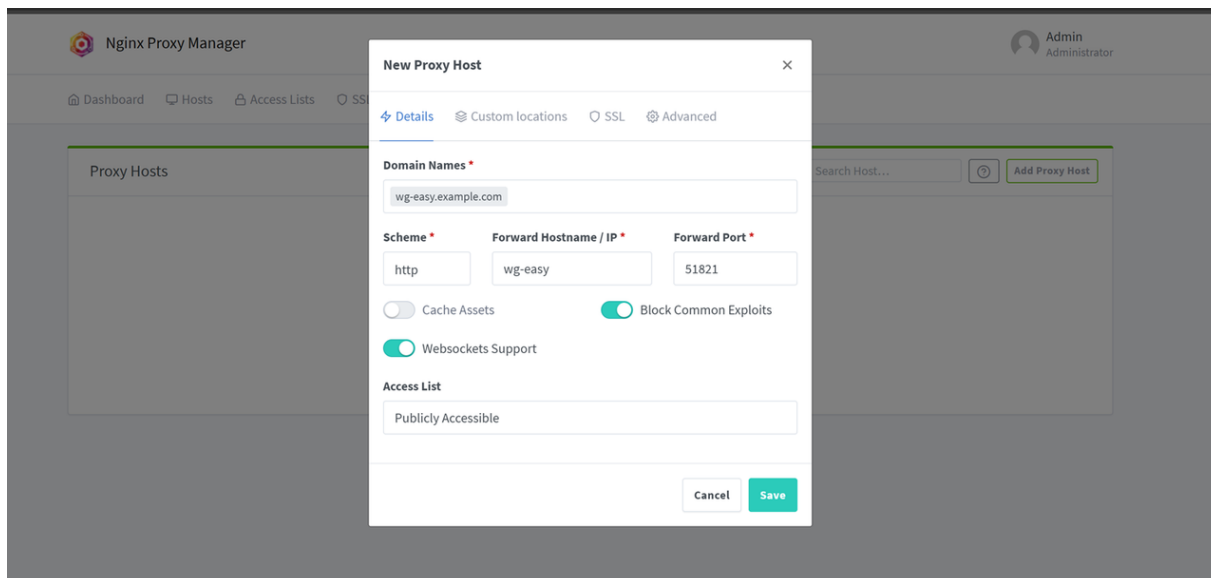


6. Click **Add Proxy Host** to create a new reverse proxy connection.

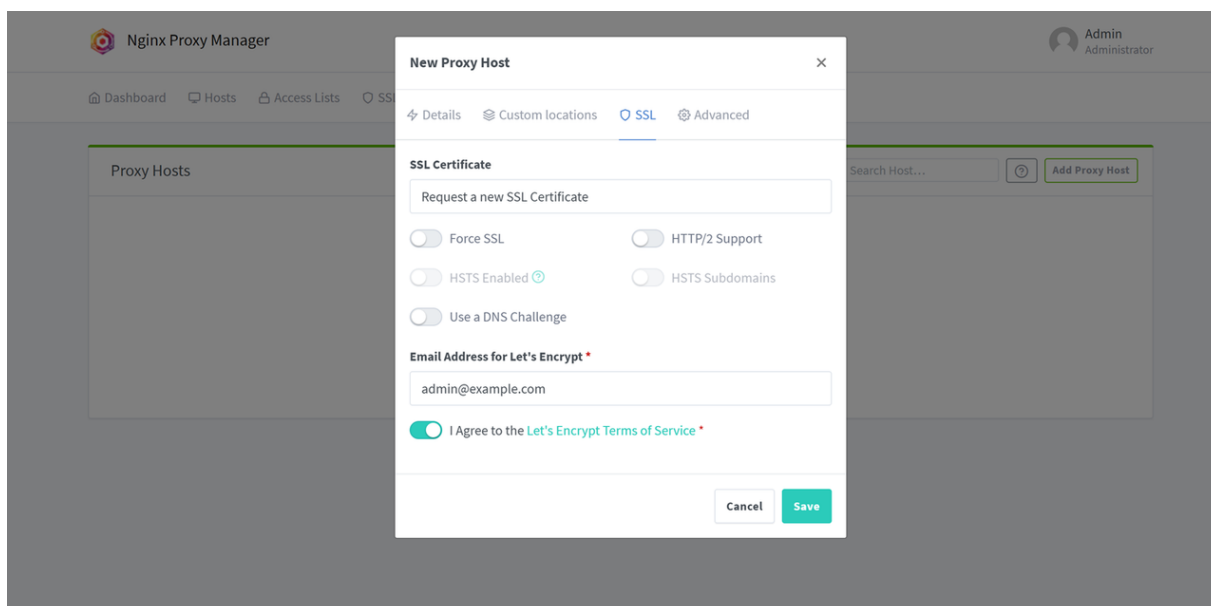


7. Enter your domain name and keep `http` as the connection scheme.

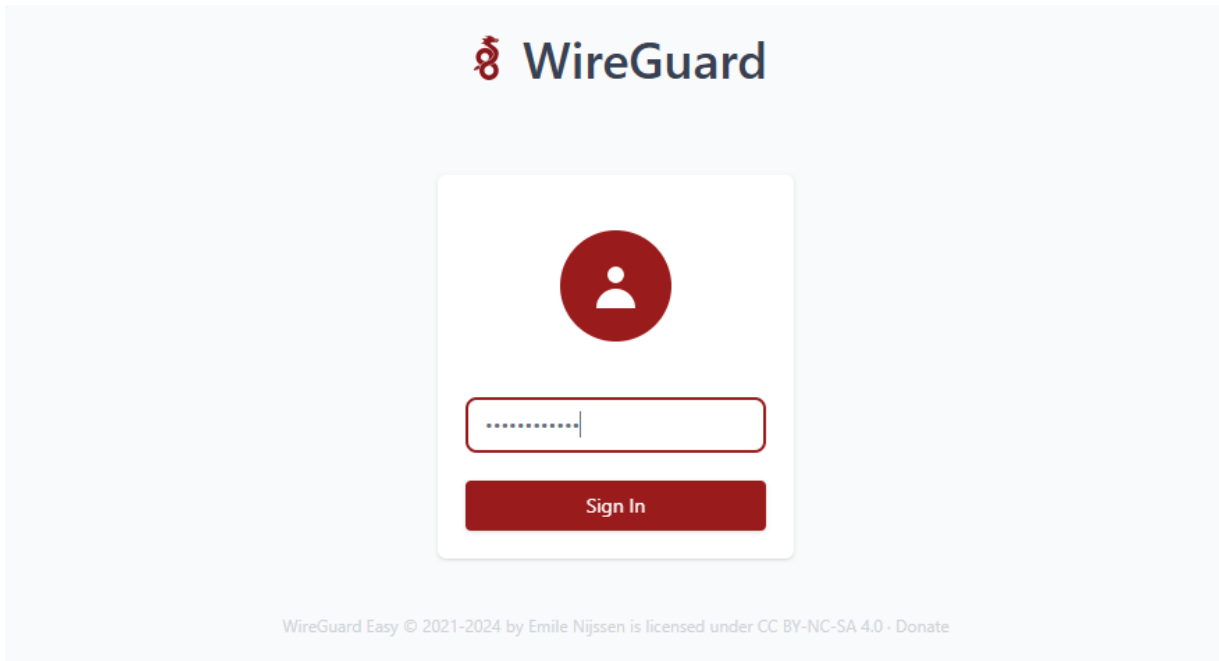
8. Enter the Wg-Easy container name in the **Forward Hostname/IP** field and its web management port `51821` in the **Forward Port** field.



9. Turn on the **Block Common Exploits** and **Websockets Support** options to secure the reverse proxy connection.
10. Keep `Publicly Accessible` as the **Access List** value and navigate to the **SSL** tab to manage the domain's SSL certificate.
11. Click the **SSL Certificate** drop-down and select **Request for a new SSL Certificate with Let's Encrypt** from the list of options.



12. Enter your active email in the **Email Address** field and click to agree to the Let's Encrypt terms of use.
13. Click **Save** to apply the reverse proxy configuration and generate a new SSL certificate.
14. Click your domain in the **Proxy Hosts** list and verify that you can access the Wg-Easy login page.



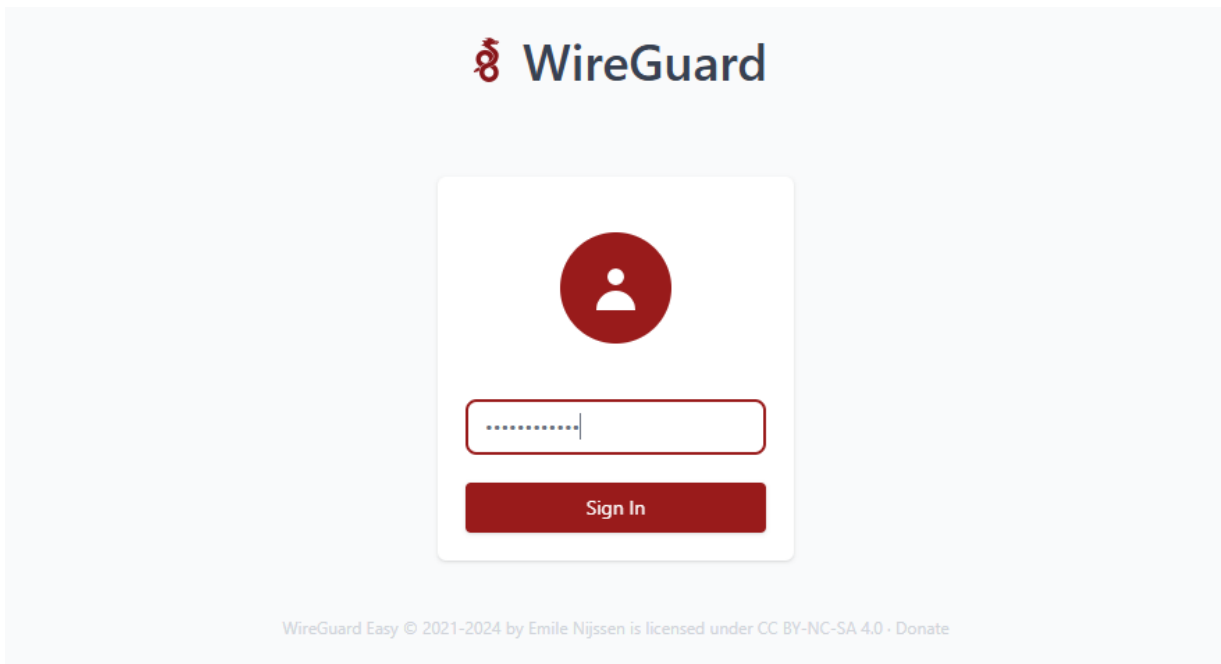
## Access Wg-Easy

Follow the steps below to access Wg-Easy and create a new WireGuard client to connect to the server.

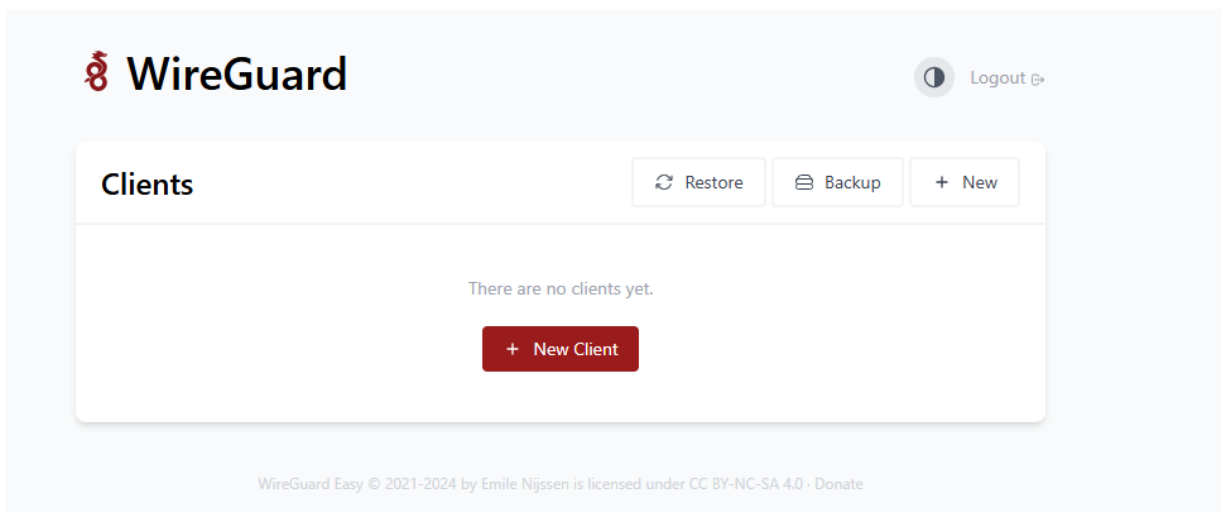
1. Access Wg-Easy using your `wg-easy.example.com` domain.

```
https://server-ip-address
```

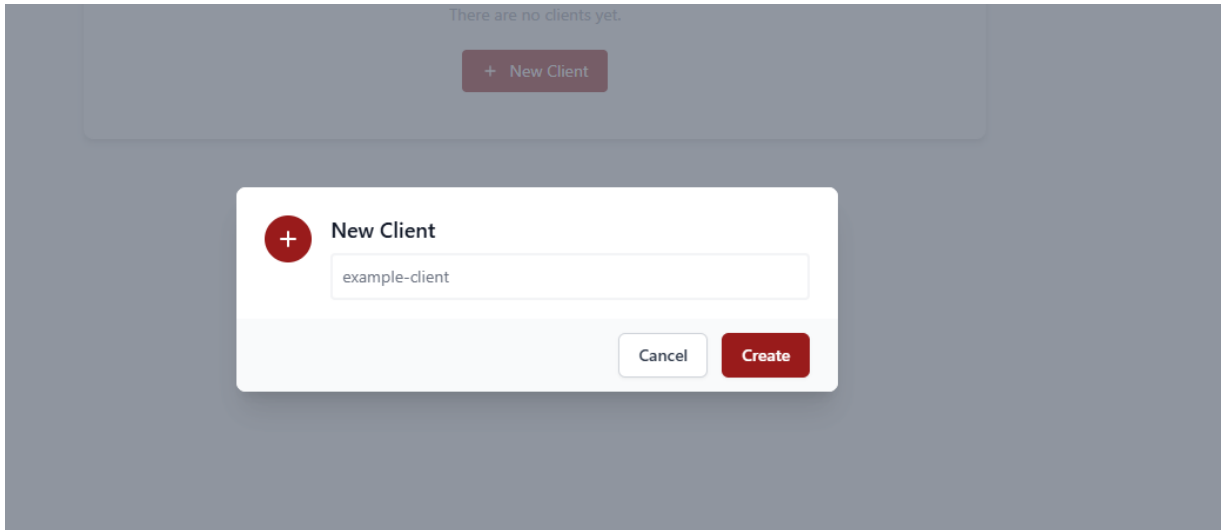
2. Enter the administrator password you hashed during installation to log in to the Wg-Easy interface.



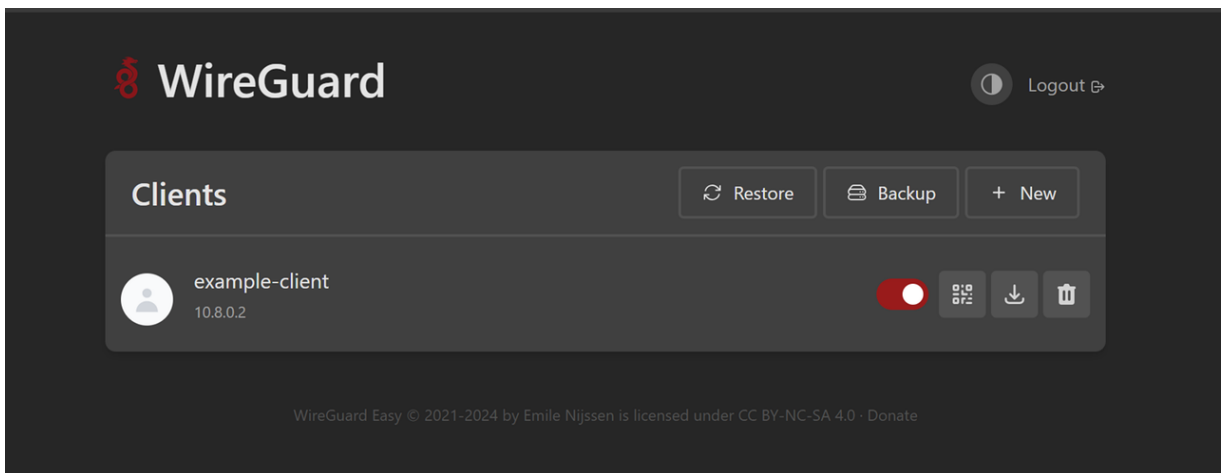
3. Click **New Client** to create a new WireGuard client.



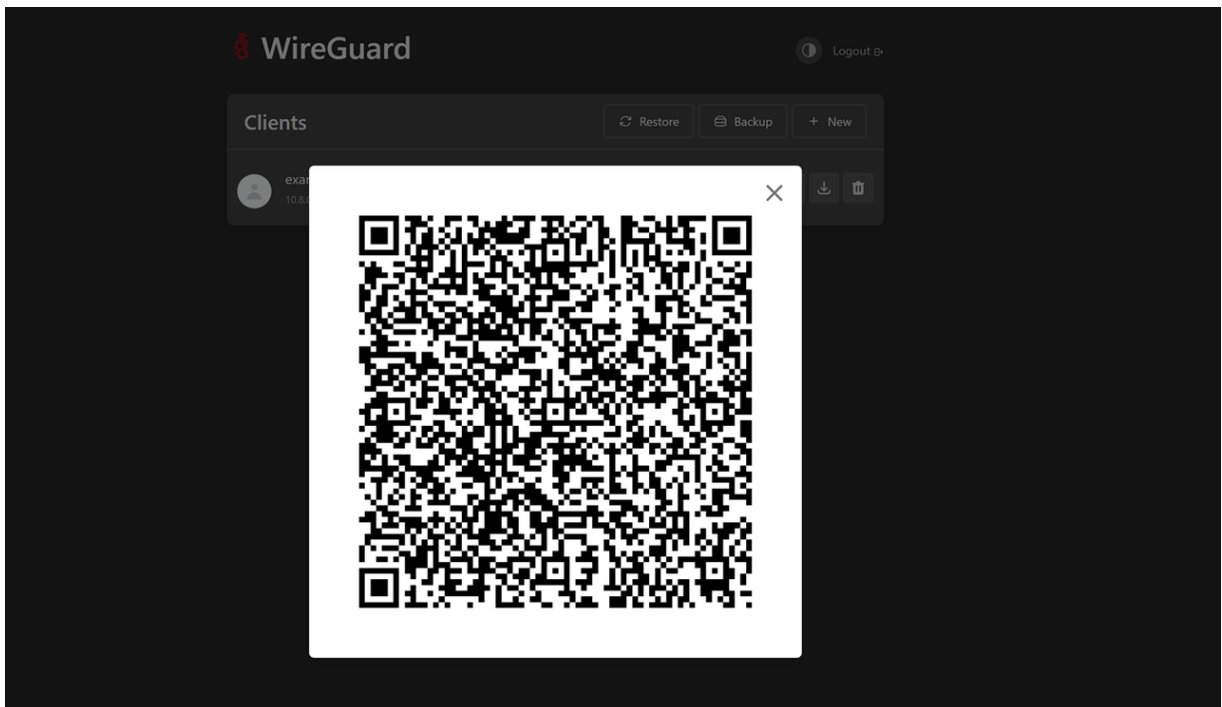
4. Enter a new client name and click **Create**.



5. Verify that the new client is created and click **Download Configuration** to download the client's WireGuard configuration file.



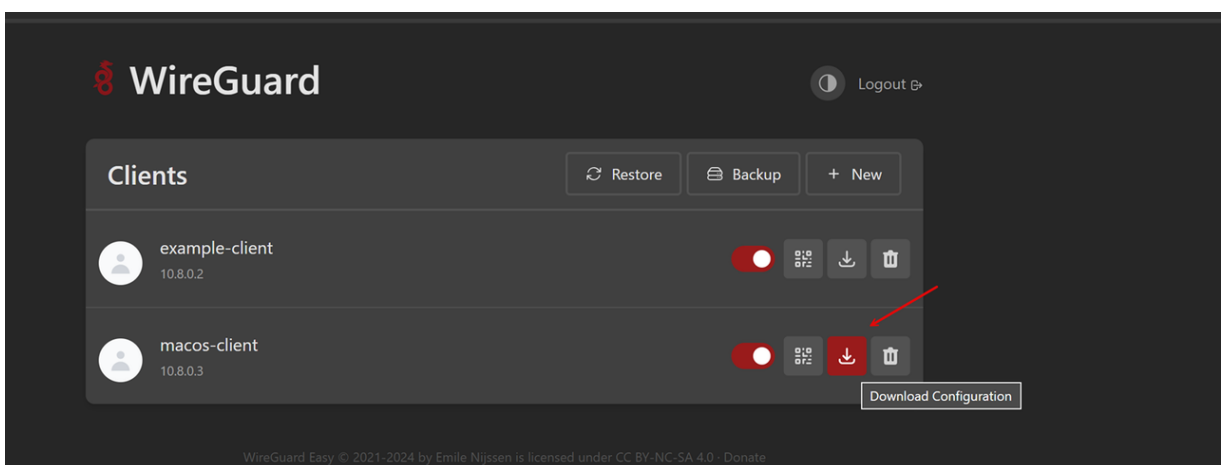
6. Click the **QR Code** option to reveal a QR Code to scan and connect mobile device clients to the WireGuard server.



## Connect a WireGuard Client to Wg-Easy

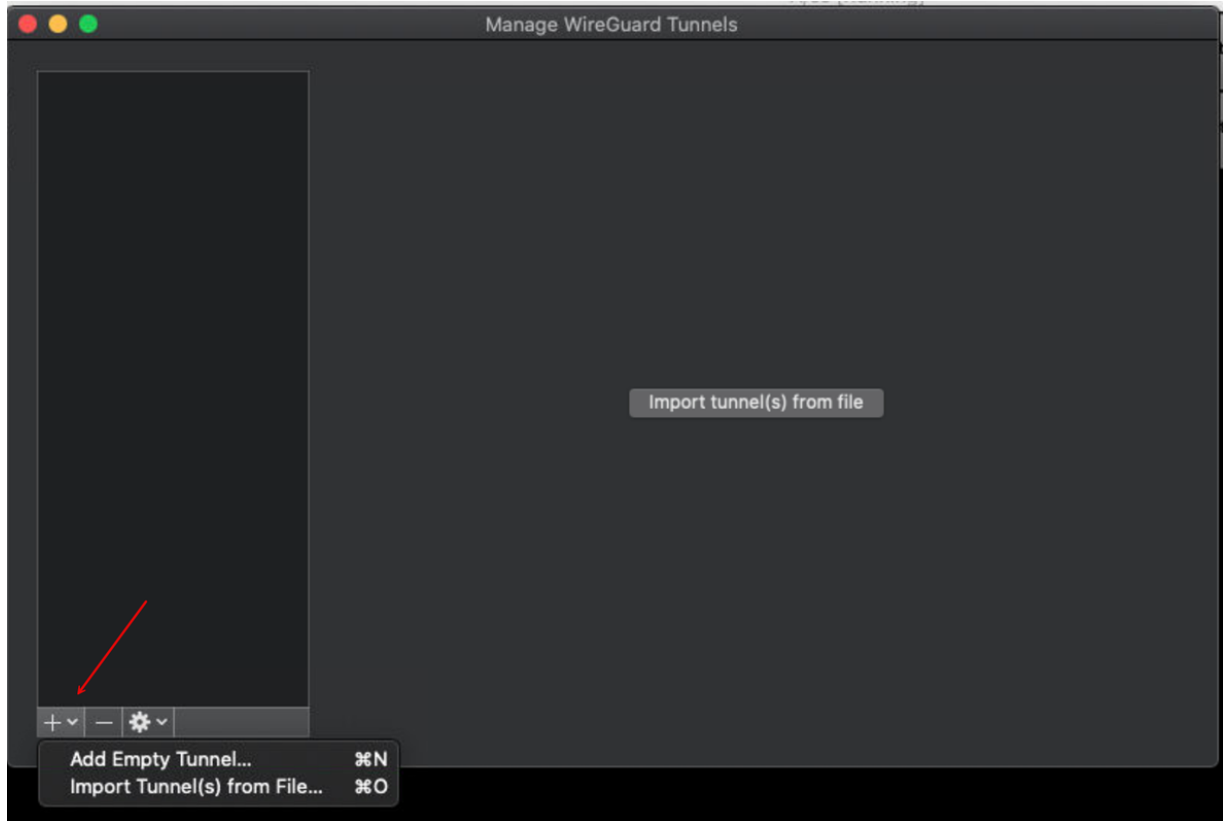
Follow the steps below to connect a WireGuard desktop client to the Wg-Easy server and verify that the user can access the Internet through the server.

1. Create a new WireGuard client and download the client configuration from the Wg-Easy interface to connect the client.

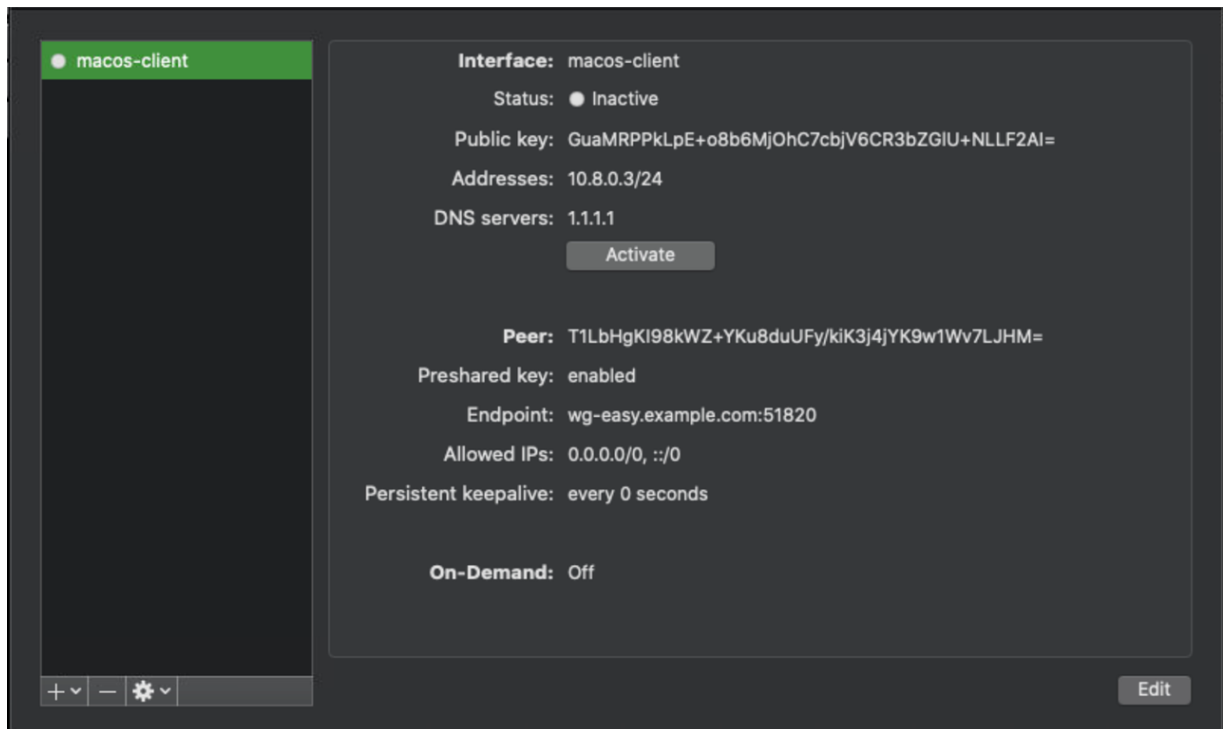


2. Download the [WireGuard client package for your device](#). For example, WireGuard for macOS.
3. Open WireGuard from your applications menu.

4. Click **Manage Tunnels** to open the WireGuard configuration interface.
5. Click the **Add Tunnel** drop-down, and select **Import Tunnel(s) from File** to browse and open your WireGuard client configuration.



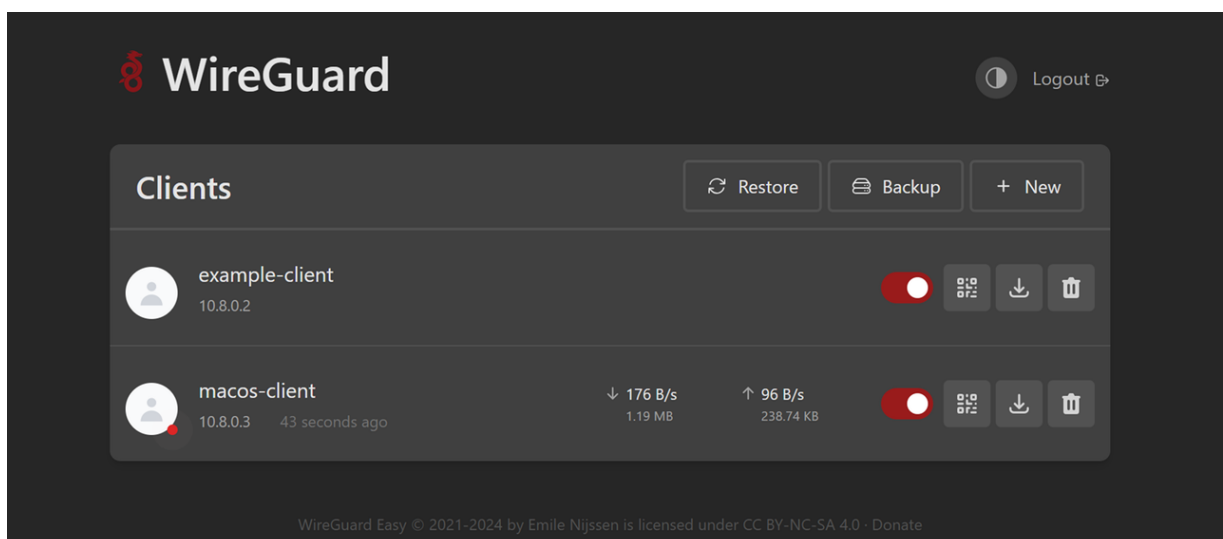
6. Verify that a new WireGuard tunnel is available and click **Activate** to connect to the Wg-Easy server.



7. Verify that the connection is active and open the Wg-Easy management interface to view the network statistics.

<https://wg-easy.example.com>

8. Monitor and manage the WireGuard client's network usage information.



## Conclusion

---

You have installed Wg-Easy and managed WireGuard clients on Ubuntu 24.04 using Docker. You can create multiple WireGuard clients, assign multiple network addresses, enable and disable connections using the Wg-Easy management interface. In addition, you can integrate Wg-Easy with existing WireGuard configurations to manage multiple connections and hosts on your server. For more information and configuration options, please visit the [Wg-Easy project repository](#).



VULTR

