Wireguard-Easy VPN Setup Guide

Setup your own VPN!

Overview

What is WG-Easy?

Introduction to WG-Easy

WG-Easy is an all-in-one solution designed to simplify the deployment, management, and operation of a **WireGuard VPN** server. It combines a powerful VPN protocol (WireGuard) with a user-friendly web-based interface, making it easy for users to control their VPN server, manage clients, and view traffic statistics. WG-Easy leverages Docker, making it straightforward to deploy and maintain across various environments.

WireGuard is known for being a fast, modern VPN protocol, but setting it up can be a bit daunting for beginners. **WG-Easy** removes that complexity by providing an intuitive web UI and pre-configured environment for quick deployment and management of WireGuard.

Key Features of WG-Easy

WG-Easy offers a rich set of features aimed at making WireGuard VPN management as effortless as possible:

1. **All-in-One WireGuard Solution**: Combines WireGuard with a simple web UI for managing VPN clients and settings.
2. **Web UI for Management**: Allows you to create, modify, and delete clients, as well as view their activity in real-time.
3. **Client Management**:
   - List and manage VPN clients.
   - Generate QR codes for easy mobile client configuration.

    - Download configuration files for clients.
4. **Traffic Statistics**: Provides detailed TX/RX charts for each connected client, allowing you to monitor bandwidth usage.
5. **Multi-language Support**: Supports multiple languages, including English, German, French, and more.
6. **Automatic Light/Dark Mode**: Web UI adjusts to your system theme preferences.
7. **Client Expiry and One-Time Links**: Set client expiration times and generate temporary download links for configurations.
8. **Prometheus Metrics Support**: Integration with Prometheus for detailed traffic metrics, perfect for monitoring via Grafana or other tools.
9. **Gravatar Support**: Displays user avatars based on email addresses, offering a personalized touch.
10. **Security Features**:
    - Admin login secured with a bcrypt password hash.
    - TLS support for secure API communication.

Getting Started

Getting Started

WG-Easy Step-by-Step Guide

Setting up WG-Easy to manage your WireGuard VPN server is a simple process that requires a few steps using Docker. In this guide, we'll walk through the steps to get WG-Easy up and running, including detailed configurations, environment variables, and port forwarding requirements.

Step 1: Install Docker

Step 2: Enable IP Forwarding

What is Portforwarding?

What is Port Forwarding?

Port forwarding is a networking technique used to redirect traffic from an external IP address and port number to an internal IP address and port. It's commonly used to make services running on a private network (such as a home or corporate network) accessible to devices outside that network, such as the internet.

When a request is made to a specific port on a router's external IP address (such as your home or business's public IP address), the router forwards that request to the appropriate port on an internal server or device running the service. This is crucial for applications like web servers, game servers, or VPN services like WireGuard.

For instance, if you are running a **WireGuard VPN** server on your home network, you need to configure port forwarding so that VPN clients on the internet can connect to your server through the public IP address.

Example of Port Forwarding:

- External IP: `203.0.113.5`
- External Port: `51820` (WireGuard VPN)
- Internal IP: `192.168.1.10`
- Internal Port: `51820`

With port forwarding enabled, any traffic coming to `203.0.113.5:51820` will be forwarded to `192.168.1.10:51820` on your local network, allowing the VPN clients to connect to your WireGuard server.

To allow VPN traffic to pass through your server, you need to enable IP forwarding on your host machine. Follow these steps:

1. Open the system configuration file:

```
sudo nano /etc/sysctl.conf
```

2. Uncomment the line:

```
net.ipv4.ip_forward=1
```

3. Apply the changes:

```
sudo sysctl -w net.ipv4.ip_forward=1
```

4. Set up IP masquerading and forwarding rules with `iptables`:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i wg0 -o eth0 -j ACCEPT
sudo iptables -A FORWARD -i eth0 -o wg0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Step 3: Configure WG-Easy with Docker Compose for Dockerswarm

To easily manage and maintain WG-Easy, we will use **Docker Compose**. Here's a sample `docker-compose.yml` file that configures WG-Easy.
If you want to start the Container for Docker Standalone Engine then remove the deploy: block.

Create a `docker-compose.yml` file:

```
version: "3.8"
services:
  wg-easy:
    image: ghcr.io/wg-easy/wg-easy
    networks:
      - management_net  # Required for Traefik or Portainer access (Change this to your Network)
    deploy:
      mode: replicated
      replicas: 1
      labels:   # If you dont use Traefik, remove this whole labels block
        - 'traefik.enable=true'
        - 'traefik.http.services.wg-easy.loadbalancer.server.port=51821'
        - 'traefik.http.services.wg-easy.loadbalancer.server.port=51820'
    container_name: wg-easy
    volumes:
      - /path/to/wg-easy-data:/etc/wireguard  # Changed volume mount path
    ports:
      - "51820:51820/udp"
      - "51821:51821/tcp"
    restart: unless-stopped
    environment:
      - WG_HOST=your-server-ip-or-domain
      - PASSWORD_HASH=your-web-ui-password-hash
      - PORT=51821
      - WG_PORT=51820
      - UI_TRAFFIC_STATS=true
      - UI_CHART_TYPE=1
      - WG_ALLOWED_IPS=0.0.0.0/0
      - WG_DEFAULT_ADDRESS=10.8.0.x  # Client IP address range
```

```
      - WG_DEFAULT_DNS=1.1.1.1       # DNS server for VPN clients
      - WG_POST_UP=iptables -A FORWARD -i %i -j ACCEPT; iptables -t nat -A POSTROUTING -j MASQUERADE;
iptables -A FORWARD -o %i -j ACCEPT
      - WG_POST_DOWN=iptables -D FORWARD -i %i -j ACCEPT; iptables -t nat -D POSTROUTING -j MASQUERADE;
iptables -D FORWARD -o %i -j ACCEPT
    cap_add:
      - NET_ADMIN
      - SYS_MODULE
    sysctls:
      - net.ipv4.ip_forward=1
      - net.ipv4.conf.all.src_valid_mark=1
networks:
  management_net:
    external: true
```

Key Parameters

- **`WG_HOST`**: The public IP address or domain name of your VPN server.
- **`PASSWORD_HASH`**: The password for logging into the web interface.
- **`WG_PORT`**: The WireGuard port (default 51820).
- **`WG_DEFAULT_ADDRESS`**: The IP range that the VPN clients will use (default is 10.8.0.x).
- **`WG_DEFAULT_DNS`**: The DNS server used by VPN clients (e.g., 1.1.1.1 or 8.8.8.8).

Step 4: Create `PASSWORD_HASH`

```
docker run -it ghcr.io/wg-easy/wg-easy wgpw write-your-password-here
```

Check this Link

Step 5: Start WG-Easy

Now that you've created the Docker Compose file, start WG-Easy by the following command to deploy a stack:

```
docker stack deploy -c docker-compose.yml wg-easy
```

> This command will deploy WG-Easy as a containerized service. The web UI will be availableat `http://your-server-ip:51821`, and WireGuard will listen on port `51820` for incoming VPN connections.

Log in using the password you specified in the Docker Compose file. From the web UI, you can add new clients, view traffic statistics, and download client configurations.

Environment Variables and Configuration Options

Here’s a list of common **environment variables** that you can set in the Docker Compose file:

Env Variable	Default	Description
`WG_HOST`	-	Public IP or domain of the VPN server
`WG_PORT`	`51820`	WireGuard's listening port
`PASSWORD_HASH`	-	Password Hash for accessing the web UI
`WG_DEFAULT_ADDRESS`	`10.8.0.x`	IP range for VPN clients
`WG_DEFAULT_DNS`	`1.1.1.1`	DNS server for VPN clients
`UI_TRAFFIC_STATS`	`false`	Enable detailed RX/TX traffic stats
`UI_CHART_TYPE`	`1`	Type of chart to use in web UI (1 = line chart)
`WG_ALLOWED_IPS`	`0.0.0.0/0, ::/0`	IPs allowed for routing through the VPN
`LANG`	`en`	Language for the Web UI (supports multiple languages)
`ENABLE_PROMETHEUS_METRICS`	`false`	Enable Prometheus metrics for monitoring

Step 6: Port Forwarding

For external access, you need to set up **port forwarding** on your router or firewall. Forward the following ports:

- **51820/udp**: WireGuard VPN port.
- **51821/tcp**: WG-Easy Web UI port.

These ports allow external clients to connect to your VPN server and administrators to access the web interface.

> If you dont know how to do that go visit this Website from NordVPN.

Conclusion

WG-Easy simplifies the process of managing a WireGuard VPN server. With features like an intuitive web UI, automatic client management, detailed traffic stats, and easy deployment using Docker, it’s an excellent solution for both beginners and advanced users alike. Whether you're setting up a personal VPN or managing one for your organization, WG-Easy provides all the tools needed for secure, efficient operation.