Install WG-Easy

https://www.youtube.com/embed/BRLB4wRL4cM

WG-Easy is essentially exactly what it sounds like. It's a containerized setup that includes a web user interface intended to make setting up a Wireguard network as easy as possible. Well, it pretty much lives up to its name. The setup of the server is very straight-forward, and the web user interface is very easy to use once everything is up and running. Adding devices, using the native Wireguard clients on each OS is also a breeze.

What You'll Need

- A machine to act as a server with Docker-CE and Docker Compose installed
- Wireguard tools installed and a Linux Kernel with Wireguard installed
- A Public IP address
- (Optional) A domain or subdomain for your server
- (Optional) a Reverse Proxy
- Access to open ports 51820 and 51821 on your firewall for the server
- About 10 minutes of your time.

Install

I used a Digital Ocean droplet to setup my server. The benefit here is that I get a public IP address and a virtual appliance type firewall in front of that server. Full discolsure, I have an affiliate link for Digital Ocean. If you use that link, you'll get a $ 50.00 credit for 60 days to test out Digital Ocean. The VPS I setup cost only $6.00 / month. So you could setup a lot of servers in 60 days to try out. If you stay with them and become a paying customer, I'll get a credit as well. If you don't, I don't.

Installation of Docker-CE and Docker Compose via a Simple Script

You can easily install Docker-CE, Docker-Compose, Portainer-CE, and NGinX Proxy manager by using this quick install script I created and maintain on Github. Just use the command:

```
wget -O install-docker.sh https://gitlab.com/bmcgonag/docker_installs/-/raw/main/install_docker_nproxyman.sh
```

To download the script to your desired host.

Change the permissions to make the script executable:

`chmod +x ./install-docker.sh`

and then run the script with the command:

`./install-docker.sh`

When run, the script will prompt you to select your host operating system, then will ask you which bits of software you want to install.

Simply enter 'y' for each thing you want to install. In this case we definitely want Docker-CE and Docker Compose. You can optionally elect to install and setup NGinX Proxy Manager if desired.

At some point, you may be asked for your super user (sudo) password as well.

Allow the script to complete installation.

At this point, you might want to log out and back in, as this will allow you to use the `docker` and `docker-compose` commands without the need of sudo in front of them.

Installing WG-Easy

WG-Easy is really a straight forward setup. First, let's create the folder structure we want on our server. I always like to create a parent docker folder, and inside that parent folder, create a folder for each application I'll be running. This makes it easy to update applications as needed, as well as backing them up by zipping up the parent 'docker' folder, and copying that zipped version off to my backup servers. Let's create our folders now, with one simple command:

`mkdir -p docker/wg-easy`

Now, we'll move into our foldeer and create a new file called "docker-compose.yml":

`cd docker/wg-easy`

`nano docker-compose.yml`

Now that we are inside our "docker-compose.yml" file, we'll add the code block below to it. Then, you'll want or need to change a couple of the variables quickly to make sure that you have everything ready to go.

```
version: '3'
services:
  wg-easy:
```

```
    container_name: wg-easy
    environment:
      - LANG=en
      - WG_HOST=<your ip or domain name here>
      - PASSWORD=<a-long-str06-pa5sw0rD-Her3>
    volumes:
      - ./wg-easy:/etc/wireguard
    ports:
      - 51820:51820/udp
      - 51821:51821/tcp
    cap_add:
      - NET_ADMIN
      - SYS_MODULE
    sysctls:
      - net.ipv4.conf.all.src_valid_mark=1
      - net.ipv4.ip_forward=1
    restart: unless-stopped
    image: ghcr.io/wg-easy/wg-easy
```

In the file you just copied and pasted, change any of the values inside the less than "<" and greater than ">" symbols. For WG_HOST you can use either the IP address of the host server, or the domain / subdomain for the server. To use the domain / sub-domain you need to have an A-record pointing the domain / sub-domain pointing to your server's public IPv4 address.

Enter a really good, long strong password for the PASSWORD entry, and optionally you can change the left side of the port mappings if needed, but usually 51820 and 51821 should be free if you haven't already setup a wireguard server on this same machine.

Save the file with CTRL + O, then press Enter to confirm. Exit the nano editor with CTRL + X.

> Don't forget to setup your A-record if using a domain / sub-domain, and additionally, setup your reverse proxy if using a reverse proxy for the web UI. The port for accessing the Web UI is 51821.

Now, let's startup our server. We'll use two commands on one line:

```
docker compose up -d && docker compose logs -f
```

The first part of the command tells docker compose to start the service running. The second part tells docker compose to show us the running log out put after the service is started. Here we can look for any errors or other issues we may need to address. If you don't see any errors, just use CTRL + C to exit the logging.

In your favorite modern browser go to either the IPv4 address and port 51821, or your domain and port 51821, or if you've setup a reverse proxy like I did in the video, the domain.

You should be greeted with a Login prompt asking for your password. This is the PASSWORD value you put in the docker-compose.yml file.

Once logged in, you can create a new configuration for your first device. ONce named, and created, you can use the QR code to scan with the Wireguard app on your mobile device, or download the config using the download icon for your desktops and servers.

On Linux, you want to make sure you have Wireguard and Wireguard tools installed an ready. Use wg-quick to bring your devices up and down, or find a suitable desktop tool with a UI to do this with.

You'll see when a device is connected in the server, and see when it has traffic flowing as well.

congratulations, you've setup WG-Easy and are ready to start using it for private, secure tunneled traffic all over the internet.

Support My Channel and Content

Support my Channel and ongoing efforts through Patreon:

https://www.patreon.com/awesomeopensource

Revision #2
Created 2024-02-20 12:35:33 UTC by Brian McGonagill
Updated 2024-02-20 16:11:32 UTC by Brian McGonagill