Netbird with Wireguard

Setting Up Netbird

Install and Setup the Netbird Wireguard System

https://www.youtube.com/embed/_-vfSgqmOUI

Netbird is a new offering in the Wireguard "flat" network space. They offer a great set of options and some fairly simple setup. We'll be installing this on a VPS from Digital Ocean toady.

Netbird reached out and asked if I'd be willing to cover their offering. I am an open source advocate, and since they are making this awesome offering open source, it only makes sense that I cover it! The best news is, they did not pay me to cover this. I do it, because this is what i do. They did however, offer up a discount code for all of you to try out their paid services. Their pricing model is already extremely affordable, so getting a 20% discount on top of that is just amazing, and I truly appreciate them offering this out to my viewers!

> You can get a **20% discount** on a Team or Business plan by signing up for the plan, then emailing the **discount code AWESOME_OSS** to hello@**netbird.io** from the email address you use to sign up for the service. It's that simple!

Full transparency: I do have a Digital Ocean affiliate link in the description of this video. If you use it you'll get anywhere from $50 to $100 for 60 days to try out the services and offerings of Digital Ocean. If you stay with the service, I will eventually get a credit on my account as well. If you don't like the service and cancel, then I won't. Simple as that.

What you'll need

- A server (highly recommend a cloud hosted VPS for a VPN server) - Netbird recommends 1 CPU with 2 GB RAM minimum. This is a $12 / month server on Digital Ocean.
- Docker and Docker Compose installed.
- Curl installed
- jq installed
- A domain name with an A record pointing to your public IP address (the server's public IP)
- About 20 minutes of your time.

Quick Step List

1. Update your server / VPS to make sure you have the most recent packages and patches available.
2. Install Docker and Docker-Compose
3. Install jq and curl.
4. Setup your domain name and A-record. NOTE: If you are using Cloudflare, you need to enable gRPC on the domain network.
5. Check your server hosts file (/etc/hosts) and make sure that the FQDN (your domain / sub-domain) is not being pointed to by localhost 127.0.0.1 or the loopback address 127.0.1.1 (this caused me some trouble in getting the client to connect properly).
6. Once all of this is setup, run the quick-start script on the Netbird page.

   https://docs.netbird.io/selfhosted/selfhosted-quickstart
7. When complete, you'll be up and running. You'll have Zitadel setup for authentication, and Netbird setup with a management service that can be accessed via CLI or Web GUI.
8. Install the Netbird client. Find the client for your OS, and install it. If using Linux, as with most tools like this, there is only a CLI option, but hopefully Netbird will get us a nice GUI option in the future.

Detailed Steps

First, let's setup our domain name and A-Record.

You need to own a domain, or go register a new domain. You'll need to have a public IPv4 address to the server where you plan to run Netbird. There are a whole host of ports that need to be available from the server as well, so we really need to make sure that we have all the ports forwarded if you are running behind a firewall, or from your LAN. In my case, I prefer to run a VPN from a VPS as this gives much better up-time, and isn't dependent on my home network being available.

I created a new droplet on Digital Ocean at the $12 / month rate which is 1 CPU and 2 GB or RAM, as detailed on the Netbird quick start guide.

Next, make sure to setup a new domain / subdomain name. In my case I went to Cloudflare to my domain "opensourceisawesome.com". I selected to add a new A-Record, and copied the public IPv4 address from Digital Ocean for my new VPS, and pasted it into the A-Record I created. Now I have the domain netbird.opensourceisawesome.com pointing to my public IP address on Digital Ocean.

> NOTE: In Cloudflare, you need to enable "gRPC" for the domain you are setting this up on. You can do this by clicking on the domain, the selecting 'Network' from the left side menu. In there you'll find an option for "gRPC". Make sure this is enabled.

Now, I'll login to my VPS via SSH to continue my setup.

Fix Localhost / Loopback Issues

Quickly, look at the 'hosts' file, and make sure that you don't have your domain / subdomain name on the same line as either localhost or the loopback addresses (127.0.0.1 or 127.0.1.1).

In my case, I saw this

```
127.0.0.1          localhost
127.0.1.1          netbird.opensourceisawesome.com netbird
```

This caused issues when trying to connect my clients. Make sure to remove the entry if your subdomain or domain are on either of these lines. Afterward you should see something like this only:

```
127.0.0.1          localhost
```

This will allow your domain to be reached properly if your DNS A-Record is setup correctly.

Install Docker-CE and Docker Compose

Next, we'll install Docker and Docker Compose. I have a script that will help you install this on the various Linux distributions out there. You can run the script by doing the following:

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

This will pull down the script, and name it "install-docker.sh".

Next, make the script executable with

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

Then run the script with

```
./install-docker.sh
```

You'll be prompted for your sudo password if you are not running as root. After that answer 'y' to any items you want to install. In this instance we just need the first two options: Docker-CE and Docker Compose. You can answer 'n' to the rest of the options.

The script will install Docker-CE and Docker Compose, as well as setting the currently logged in user in the 'docker' group, and create a new docker network. You won't need the docker network in this case.

To make sure your group privileges are set on your user, you should log out and back in after the script completes.

Next, let's install jq and curl. I'm using Ubuntu, so I'll give those commands here, but if you use a different Linux variant, make sure to use the proper package manager for your distribution.

Install jq and curl

`sudo apt install jq curl -y`

When that completes, we'll be ready to run the one-liner command that Netbird provides to get us up and running quickly and easily.

Run the install script for Netbird:

`export NETBIRD_DOMAIN=netbird.example.com; curl -fsSL https://github.com/netbirdio/netbird/releases/latest/download/getting-started-with-zitadel.sh | bash`

where you change out the NETBIRD_DOMAIN for your actual FQDN.

In my case I ran

`export NETBIRD_DOMAIN=netbird.opensourceisawesome.com curl -fsSL` `https://github.com/netbirdio/netbird/releases/latest/download/getting-started-with-zitadel.sh` `bash`

Allow the script to complete. You'll have several docker containers running when it's done and at the bottom, you should see a message saying that the install is complete, and providing you a link to your new Netbird management site, as well as an initial username and password.

Copy the password, and username, and then open the site. You should be greeted by a Zitadel login page. Enter the username provided, then click 'Next'.

Now enter the password provided, and click 'Next'.

Here you'll be prompted to setup MFA (Multi-factor Authentication). Feel free to use whatever works best for you. I chose to setup a TOTP using Bitwarden.   Verify the code, or method you choose, then continue. You'll be prompted to change the password, and asked for the verification code / method one more time. After that process, you'll be directed to the Netbird management page.

It's time to add a client.

Adding a Client

> NOTE: I did not have to run the Netbird client as sudo, which is a change from how Tailscale works.

Once you are at the Management console for Netbird, you can click the "Add a Peer" button. When you do you'll be presented with a pop-up message showing you a one liner to install the Netbird client on a Linux pc. You'll also be shown a multi-line method if you prefer to do it that way. At the bottom of the pop-up you'll be given a one-line command to connect your client to your Netbird

server after the client application is successfully installed.

If you are on Windows, MacOS, Android, etc, just make sure that the tab on the pop-up is set to the OS you are using and follow the instructions as provided.

For Linux, we can install the client with this command:

```
curl -fsSL https://pkgs.netbird.io/install.sh | sh
```

After the install completes, you'll see a message saying to use `netbird up` to connect If you use this command, it will attempt to connect you to the Netbird.io servers, and not your self-hosted server. To connect to your server, go back to your management page and scroll to the bottom of the 'Add Peer' pop-up window. There you'll find a more full command to bring up your Netbird client on your self hosted server. It should look something like

```
netbird up --management-url https://netbird.youramazingdomain.com
```

Where you'll swithh out the URL https://netbird.yojramazingdomain.com for your actual netbird server url.

When you do this, if you are doing so from a desktop machine, you'll see a browser window open up, and ask you to authenticate with your Netbird credentials. Authenticate, and you'll be told your setup is successful. You can close the browser window, and now do

```
netbird status
```

in your terminal, to see that Netbird is connected.

What about a server with no GUI?

You can also setup your Netbird VPN on a server with no GUI. Using the command line, install the Netbird client in the same way as above. Once installed, you'll need to go to your management ui in a browser, and select the "Setup Keys" tab. Here click the button for "Add Key".

Fill in a name for your key to identify it easily in a list.

Next, choose how many machines you want to setup using this key. Maybe you have 10 servers you need to add. You can do this more rapidly by using the same key for each , instead of separate keys for each.

If you are setting up devices in groups, select the group for the devices that will use this key, and finally set an expiration for using this key.

Click "Create Key".

Copy the Key! The won't show you this key again, so copy it as soon as you see it and store it in a password manager or encrypted file.

Now, on your server use the command line to add your device to the Netbird network with this command line structure:

```
netbird up --management-url https://netbird.youramazingdomain.com --setup-key 9129F217-15CA-4DA0-2107-8ED020109879
```
<--- not a real key

You server is now connected, no separate authentication needed since you used the setup key. You can use all the Netbird CLI commands to bring your client up / down, check status, and so on.

Congratulations! You are now setup with a Wireguard VPN using Netbird. You have a great Web based management portal, and you have Multi-factor authentication using Zitadel for your user management.

A brief word on User Management

While there is a "Users" tab on your Netbird management interface, in order to add users, since it's using Zitadel for Authentication, you need to add the users through Zitadel. The good news is this is a fully functional Zitadel Authentication and IdP server. You can get to the console at

https://netbird.youramazingdomain.com/ui/console

Login, using your admin credentials, and from there you can complete the Zitadel setup with branding, and user grants, as well as use Zitadel for other SSO systems you want to authenticate with.

This isn't a tutorial on Zitadel, but I felt like this was important for you to know. We'll cover Zitadel some other time.

Support My Channel and Content

Support my Channel and ongoing efforts through Patreon:

https://www.patreon.com/awesomeopensource

Netbird - Update to add Relays

https://www.youtube.com/embed/ro7hbqLyv9M?si=AJ1v-54IeKWeAaEn

Netbird has put out an update with version 0.29.0 of ther amazing open source Wireguard server software that adds a new feature called "Relays".

Here's a link to their release notes on version 0.29.0

https://github.com/netbirdio/netbird/releases/tag/v0.29.0

Note, that at this point, they've also released 0.29.1 and 0.29.2 in rapid succession as they bring updates to these new features and functionalities.

This new addition brings the creation of peer-to-peer networks (devices connected directly to each other) out of the Turn server realm, and into the websockets realm. This isn't a terribly difficult update, so if you're running a version older than 0.29.0, let's go through the update process together, and we'll be on our way to having a great updated server.

The new "relay" sections of the docker-compose.yml and the manifest.json will be needing a few variables replaced with your server information.

- Your domain name (the name or ip address you use to reach the Netbird server from your client machines)
- A Port number (specifically port 33080)
- A new secure key that we'll generate down below using our command line.

First, log into your netbird server with the command line utility (terminal). You may do this via SSH if it's a remote server like mine is.

Next, we'll navigate to the correct folder. You setup may differ slightly from their ideal setup, and even my setup, but as long as you can find your base level "netbird" folder, you'll be able to get to these files.

My base level "netbird" directory is inside of a parent level "docker" directory. So I went to docker/netbird/infrastructure_files/artifacts with the command

```
cd   docker/netbird/infrastructure_files/artifacts
```

You would modify this path if needed. Many people may not have the parent level "docker" folder. If you don't, then you can likely just do:

```
cd netbird/infrastructure_files/artifacts
```

Once in this folder, we can view the files inside by doing the command:

`ls`

We need to update two of the files in this location.

1. docker-compose.yml
2. manifest.json

Let's do the docker-compose.yml file first. Using the nano editor, we'll open this file, and then using the arrow keys move almost to the end of the file.

`nano docker-compose.yml`

I stopped and added my change just above the #Coturn section. Place your cursor above the line that says

`# Coturn`

and copy the following code snippet, then paste it into your docker-compose.yml file.

```
 # relay
   relay:
     image: netbirdio/relay:latest
     restart: unless-stopped
     environment:
     - NB_LOG_LEVEL=info
     - NB_LISTEN_ADDRESS=:<new-port> # this port should be 33080 by default
     - NB_EXPOSED_ADDRESS=<your-netbird-domain>:<new-port>  # this port should be 33080 by
 default
     - NB_AUTH_SECRET=<new-auth-key>
     ports:
       - 33080:33080
     logging:
       driver: "json-file"
       options:
         max-size: "500m"
         max-file: "2"
```

Once pasted in, your docker-compose.yml should look something like this:

```
 # relay
   relay:
     image: netbirdio/relay:latest
```

```yaml
    restart: unless-stopped
    environment:
    - NB_LOG_LEVEL=info
    - NB_LISTEN_ADDRESS=:<new-port> # this port should be 33080 by default
    - NB_EXPOSED_ADDRESS=<your-netbird-domain>:<new-port>  # this port should be 33080 by
  default
    - NB_AUTH_SECRET=<new-auth-key>
    ports:
      - 33080:33080
    logging:
      driver: "json-file"
      options:
        max-size: "500m"
        max-file: "2"

  # Coturn
  coturn:
    image: coturn/coturn:latest
    restart: unless-stopped
    domainname: netbird.sysmainit.com
    volumes:
```

Now, replace the placeholders surrounded by less than "<" and greater than ">" signs:

This one needs to be entered in two places, so pay attention.
`<new-port>`

`<your-netbird-domain>`

To generate a new auth key, we'll save our changes with CTRL + O, then press Enter to confirm, and exit the nano editor with CTRL + X.

Now, use the command:
`openssl rand -base64 32 | sed 's/=//g'`

Now copy the newly generated key, and replace `<new-auth-key>` back in the docker-compose.yml file. Store this key somewhere safe, as we'll need it again in our management.json file in just a minute.

Once you've replaced all the values, save the file with CTRL + O, then press Enter to confirm, and exit the editor with CTRL + X.

Next, we need to update our manifest.json file. Again, we'll use the nano editor to do this.

```
nano manifest.json
```

Now we'll add the snipped:

```
    "Relay": {
        "Addresses": ["rel://<DOMAIN>:<PORT>"],
        "CredentialsTTL": "24h",
        "Secret": "<AUTH_SECRET>"
    },
```

Again, arrow down to the section just above the reverse proxy section, almost to the end of the file. Once updated with the new snippet, your manifest should look something like this:

```
    "StoreConfig": {
        "Engine": "sqlite"
    },
    "Relay": {
        "Addresses": ["rel://<DOMAIN>:<PORT>"],
        "CredentialsTTL": "24h",
        "Secret": "<AUTH_KEY>"
    },
    "ReverseProxy": {
        "TrustedHTTPProxies": [],
```

Replace <DOMAIN>, <PORT>, and <AUTH_KEY> with the values you used in the docker-compose.yml file.

Save your changes with CTRL + O, then press Enter to confirm, and exit the nano editor with CTRL + X.

Now do the command:

```
docker compose pull
```

Followed by the command to re-create your containers:

```
docker compose up -d --force-recreate
```

Be patient as each command completes. Once done, wait about 30 seconds to 2 minutes, then try to login to your netbird web management system. If all went well, you're now running the latest version of Netbird with Realy support.

Well done!

Support My Channel and Content

Support my Channel and ongoing efforts through Patreon:

https://www.patreon.com/awesomeopensource

Buy me a Beer / Coffee:

https://paypal.me/BrianMcGonagill