Vaultwarden / Bitwarden

- Install Vaultwarden, and Host Your Own Password Manager

Install Vaultwarden, and Host Your Own Password Manager

https://www.youtube.com/embed/mq7n_0Xs1Kg

Vaultwarden is a compatible, open source, back end for Bitwarden written in Rust. It is extremely light on resource requirements, and will work with any of the Bitwarden clients in your browsers and mobile devices.

Why Run My Own Password Manager

There are several reasons why you might want to run your own password manager:

- Data privacy
- Monetary savings
- More control over *your* data.

What Are the Risks?

There are definitely risks, as with any self hosted systems, the ownership falls on you, your infrastructure, and your habits of redundancy in case of catastrophe.

Essentially, you take on the responsibility for keeping regular backups, and ensuring those backups are stored in safe, accessible locations in case they are ever needed.

What Are the Benefits?

You get to own your data. You own it, you maintain it, you control it. You can also take advantage of some of the pay features of the hosted Bitwarden service when you host your own server.

- Organizations where you can share your passwordw with team members (family).
- Multiple users
- Allow or Disallow registrations
- Restrict registrations to emails from specific domains

- more...

How Do I Install It?

What You'll Need

- Docker
- Docker-Compose
- A Proxy Server (I use NGinX Proxy Manager)
- Your own Domain Name (can be a Dynamic Domain from DuckDNS if you prefer)
- (optional) Portainer-CE
- about 20 minutes of time.

Install Docker, Docker-CE, NGinX Proxy Manager, and Portainer-CE (the easy way).

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 Gitlab. Just use the command:

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

To download the script to your desired host machine.

Change the permissions to make the script executable:

`chmod +x ./install_docker_nproxyman.sh`

and then run the script with the command:

`./install_docker_nproxyman.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.

At some point, you'll 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.

Why Do I Need a Domain Name?

Usually you have the option to use a service on your private network without a domain, and you can absolutely do this with Vaultwarden as well, but you'll need to create a self signed certificate so you can access the Web UI and connect up clients on your local network using **https**.

Valutwarden, being a system for securely storing encrypted passwords and private information, requires that you use an SSL encrypted connection when accessing the server from anywhere but localhost (i.e. if you access the web ui from http://localhost:<port> you can use just http, otherwise your connection will have to be https, or certain actions will result in errors).

We will create a subdomain four our Vaultwarden install, and then use NGinX Proxy Manager to route any requests to our Valtwarden subcomain, to our Vaultwarden server.

Setup our Router / Firewall

Think about what you want to call your Vaultwarden subdomain. I'm going to call mine vault.exemplified.org for our tutorial. You can call yours anything you want, just remember you have to be able to point the domain to your home / host **public** ip address.

So, I create the subdomain vault.exemplified.org in my domain registrars (hover, godaddy, ghandi.net, etc) DNS settings area. I create an A record, that points my subdomain to my home / host public IP. I find my public IP using a simple site like https://ipchicken.com. I simply copy the public IP and paste it into my subdomain A record entry.

> Note: Some subdomains may take up to 48 hours to update in DNS, but usually it will update in about 30 minutes to an hour.

Next, I need to forward ports 80 and 443 in my Router / Firewall to my host private IP address. (If you are using a VPS this may be unnecessary).

Port forwarding means when my Router / Firewall receive a request on port 80 or 443 from the outside internet, I teel the router / firewall to pass that request to my host machine inside my network. So make sure to enter the correct private IP for your host machine.

Once the port forwarding is set, we now need to setup NGinX Proxy Manager to handle the request, but first we need to actually get Vaultwarden installed.

Install Vaultwarden

To install Vaultwarden, we'll be using Docker and Docker-Compose. I like to organize my docker applications inside of a single folder called "docker".

On your host machine, if you used my script to install docker, docker-compose, and NGinX Proxy Maanger, you should find a "docker" folder already in place. If you don't have one yet, just use the command:

`mkdir docker`

Now, move into that "docker" folder with the command:

`cd docker`

Next, we'll make a new folder called "vaultwarden":

`mkdir vaultwarden`

and move into that folder with:

`cd vaultwarden`

Finally, let's make a new file called "docker-compose.yml":

`nano docker-compose.yml`

And we'll copy the text block below, and paste it into the file we just created.

```
version: '3'

services:
  vaultwarden:
    restart: always
    container_name: vaultwarden
    image: vaultwarden/server:latest
    volumes:
      - ./vw-data/:/data/
    ports:
      - 8062:80
    environment:
      - SMTP_HOST=mail.example.com
      - SMTP_FROM=iamawesome@example.com
      - SMTP_FROM_NAME=VaultWarden
      - SMTP_SECURITY=starttls
      - SMTP_PORT=587
      - SMTP_USERNAME=iamawesome@example.com
      - SMTP_PASSWORD=some-long-strong-password-for-your-email-user-i-hope
      - SMTP_TIMEOUT=30
```

```
      - SMTP_AUTH_MECHANISM="Plain"
      - LOGIN_RATELIMIT_MAX_BURST=10
      - LOGIN_RATELIMIT_SECONDS=60
      - DOMAIN=https://homevault.example.org
      - INVITATION_ORG_NAME=HomeVault
      - INVITATIONS_ALLOWED=true
      - ADMIN_TOKEN=another-really-long-Str0n6-passw0rol-you-will-need
      - SIGNUPS_ALLOWED=false
      - SIGNUPS_DOMAINS_WHITELIST=example.com,mydomain.net,myotherdomain.org
      - SIGNUPS_VERIFY=true
      - SIGNUPS_VERIFY_RESEND_TIME=3600
      - SIGNUPS_VERIFY_RESEND_LIMIT=6
      - EMERGENCY_ACCESS_ALLOWED=true
      - SENDS_ALLOWED=true
      - WEB_VAULT_ENABLED=true
```

You'll want to look at the right side of the "=" (equal sign) under the "environment" section in the text you just pasted.

Change the values to the correct and real values for your system.

In particular the SMTP_ values are extremely important to get correct, as this is how the system will send you verification emails when registering. You won't be able to login until you verity your email. Along with the email verification, the DOMAIN value is important, as this is used in the email to redirect your verification token to the correct server. Make sure you have the proper subdomain / domain in this value. Finally, also make sure your values for the SIGNUPS_DOMAINS_WHITELIST are set to correct domains you want to allow sign-ups from, or you can disable signups completely after creating your initial user, and remove the WHITELIST values.

Once you've updated the environment values, make note of the port mapping under the "ports" section, and adjust the **left** side of the : if you want to change the port you access the Vaultwarden system on.

Now save this file with CTRL + O, then press Enter to confirm, and use CTRL + X to exit the nano editor.

Finally, enter

```
docker-compose up -d
```

to download and run Vaultwarden.

You can confirm it's running when you see the "done" message in the terminal, but opening your browser, nad navigating to your host machine's private IP address and the port you used.

I went to `http://192.168.10.25:8062` to verify the main UI loaded and prompted for a login.

Note, you will not be able to create a user until we complete our Proxy setup and get an SSL certificate.

Setup NGinX Proxy Manager

In NPM we want to create a new Proxy Host. In the URL field, enter the domain / subdomain you setup with your domain registar. For our example we'll enter "vault.exemplified.org", then press Enter or Tab in order to get the entry to turn into a chip. If you don't do this, the entry is removed.

Next, move to the IP field, and enter the private IP of your host machine. If you are running NPM and Vaultwarden on the same host, you have 2 other options.

1. You can use the docker gateway IP for your Vaultwarden instance, and the port you set, or
2. If NPM and Vaultwarden are on the same docker network (as long as it's not the default docker network), you can simply enter the container name for Vaultwarden (in our case "vaultwarden") in the IP field.

Next, enter the Port you entered (if you are using the container name, use enter "80"), in our case we used 8062.

Tick the box to Block Common Exploits, and to Cache Assets, and allow Websockets support.

Now move to the SSL tab. Select to "Request a New Certificate" from the drop down. Tick the option for "Force SSL", then accept the terms of service and enter your email in the field shown.

Click 'Save'.

If the pop-up window closes with no errors, you chould now be able to open the new domain / subdomain in your browser, and you should be taken to your Valutwarden server login page.

Now, you'll be on the page with SSL encryption, and you can choose to create an account, and start setting up your new Vaultwarden server.

Check out the Video

Check out the video at the top of this page for instructions, and tips, as well as an overview of the Vaultwarden UI and Admin interface.

Support My Channel

Support my Channel and ongoing efforts through Patreon:
https://www.patreon.com/bePatron?u=234177