

Install

Everything important for setup of the server in the environment of COe.

Table of Contents:

- Setup VPN
 - Install Proxmox
 - Format Disks
-
- [VPN zur FRITZ!Box unter Linux einrichten](#)
 - [Lxc Container](#)
 - [Website with nginx](#)
 - [Firefly-III](#)
 - [HA OS in VM](#)
 - [OctoPi](#)
 - [Grafana](#)

VPN zur FRITZ!Box unter Linux einrichten

Vorraussetzungen

- Pakete: vpnc, network-manager-vpnc und network-manager-vpnc-gnome
- MyFRITZ!-Adresse der FRITZ!Box:
Beispiel: pi80ewgfi72d2os42.myfritz.net
- Benutzername des FRITZ!Box-Benutzers:
Beispiel: Max Mustermann
- Kennwort des FRITZ!Box-Benutzers:
Beispiel: geheim1234
- Shared Secret des FRITZ!Box-Benutzers:
Beispiel: Zj7hPCouK65lrPU4

Vorbereitungen

MyFRITZ! einrichten

Registriere FRITZ!Box bei MyFRITZ! für feste MyFRITZ!-Adresse. [Hier wäre eine Anleitung.](#)

IP-Netzwerk der FRITZ!Box anpassen

Richte eine IP-Adresse ein, über die Sie sich mit der FRITZ!Box verbinden:

1. Klicken Sie in der Benutzeroberfläche der FRITZ!Box auf "Heimnetz".
2. Klicken Sie im Menü "Heimnetz" auf "Heimnetzübersicht".

3. Klicken Sie auf die Registerkarte "Netzwerkeinstellungen".
4. Klicken Sie auf die Schaltfläche "IPv4-Adressen". Falls die Schaltfläche nicht angezeigt wird, aktivieren Sie zunächst die Erweiterte Ansicht.
5. Tragen Sie die gewünschte IP-Adresse und Subnetzmaske ein.
6. Klicken Sie zum Speichern der Einstellungen auf "OK".

VPN-Verbindung in FRITZ!Box einrichten

Richte in der FRITZ!Box für jede VPN-Verbindung einen eigenen Benutzer ein:

1. Klicken Sie in der Benutzeroberfläche der FRITZ!Box auf "System".
2. Klicken Sie im Menü "System" auf "FRITZ!Box-Benutzer".
3. Klicken Sie neben dem Benutzer, der VPN-Verbindungen zur FRITZ!Box herstellen soll, auf die Schaltfläche (Bearbeiten) oder richten Sie einen neuen Benutzer für die VPN-Verbindung ein:
 1. Klicken Sie auf die Schaltfläche "Benutzer hinzufügen".
 2. Tragen Sie in den Eingabefeldern einen Namen und ein Kennwort für den Benutzer ein.
4. Aktivieren Sie die Option "VPN".
5. Klicken Sie zum Speichern der Einstellungen auf "OK".
6. Jetzt richtet die FRITZ!Box die VPN-Verbindung für den Benutzer ein und öffnet ein Fenster mit den VPN-Einstellungen des Benutzers.

VPN-Verbindung am Computer einrichten und herstellen

Richte die VPN-Verbindung am Computer ein:

1. Starten Sie die Advanced Network Configuration, z.B. im Terminal über "nm-connection-editor".

2. Klicken Sie im Abschnitt "VPN" auf das Pluszeichen. Das Fenster "VPN hinzufügen" wird geöffnet.
3. Wählen Sie im Fenster "VPN hinzufügen" den Eintrag "Cisco-kompatibler VPN-Client (vpnc)" aus.
4. Tragen Sie im Eingabefeld "Name" einen beliebigen Namen (FRITZ!Box-VPN) ein.
5. Tragen Sie im Eingabefeld "Gateway" die MyFRITZ!-Adresse der FRITZ!Box (pi80ewgfi72d2os42.myfritz.net) ein.
6. Tragen Sie im Eingabefeld "Benutzername" den Benutzernamen des FRITZ!Box-Benutzers (Max Mustermann) ein.
7. Klicken Sie auf das Symbol im Eingabefeld "Benutzerpasswort", wählen Sie den Option "Passwort nur für diesen Benutzer speichern" aus und tragen Sie dann das Kennwort des FRITZ!Box-Benutzers (geheim1234) ein.
8. Tragen Sie im Eingabefeld "Gruppenname" den Benutzernamen des FRITZ!Box-Benutzers (Max Mustermann) ein.
9. Klicken Sie auf das Symbol im Eingabefeld "Gruppenpasswort", wählen Sie den Eintrag "Passwort nur für diesen Benutzer speichern" aus und tragen Sie dann das "Shared Secret" des FRITZ!Box-Benutzers (Zj7hPCouK65lrPU4) ein. Das "Shared Secret" wird in den VPN-Einstellungen des Benutzers im Abschnitt "iPhone, iPad oder iPod Touch" angezeigt.
10. Klicken Sie auf "Erweitert".
11. Tragen Sie im Eingabefeld "Tunnel-Schnittstellenname" tun0 ein und klicken Sie auf auf "Apply".
12. Klicken Sie auf die Schaltfläche "Speichern", um die Einrichtung abzuschliessen. Falls die Schaltfläche "Speichern" ausgegraut ist, klicken Sie noch einmal auf "Erweitert" und dann auf "Apply" und wiederholen Sie das Speichern.
13. Klicken Sie bei der VPN-Verbindung auf den An/Aus-Schalter, um die Verbindung herzustellen.

Quelle

[Tutorial VPN zur FritzBox einrichten](#)

Lxc Container

Use Ubuntu 24 because otherwise the Neovim Version is too old for Lazy-Vim.

Create Container

You can use the helperscript with this command:

```
bash -c "$(wget -qLO - https://github.com/tteck/Proxmox/raw/main/ct/ubuntu.sh)"
```

For the Container ID use a number in the range 200-299. It is useful to give the Container this ID as the ending of the IP-Address later.

Nutzer erstellen

Create a user:

```
adduser <username>
```

Add user to sudo group:

```
usermod -aG sudo <username>
```

Switch to user.

System vorbereiten

Update the system and install Lazy-Vim using:

```
sudo apt update
sudo apt upgrade -y
sudo apt install -y git build-essential neovim tree curl
git clone https://github.com/LazyVim/starter ~/.config/nvim
```

```
rm -rf ~/.config/nvim/.git
```

Run those Lines also as `<username>`.

```
git clone https://github.com/LazyVim/starter ~/.config/nvim
rm -rf ~/.config/nvim/.git
```

Install Zsh with autocompletions:

```
sudo apt install -y zsh zsh-syntax-highlighting
sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
git clone https://github.com/zsh-users/zsh-history-substring-search ${ZSH_CUSTOM:~/.oh-my-zsh/custom}/plugins/zsh-history-substring-search
git clone --depth 1 https://github.com/junegunn/fzf.git ~/.config/fzf
~/.config/fzf/install
```

Replace "`~/.zshrc`" with:

```
ZSH_AUTOSUGGEST_STRATEGY=(history)

plugins=(cabal colorize colored-man-pages cp copyfile copypath fzf git gitignore last-working-dir sudo vi-mode
web-search zsh-interactive-cd)

ZSH_THEME="jonathan"
source ~/.oh-my-zsh/oh-my-zsh.sh

HISTSIZE="1000000000000"
SAVEHIST="1000000000000"

HISTFILE="/home/elias/.local/share/zsh/zsh_history"
mkdir -p "$(dirname "$HISTFILE")"

setopt HIST_FCNTL_LOCK
unsetopt APPEND_HISTORY
setopt HIST_IGNORE_DUPS
unsetopt HIST_IGNORE_ALL_DUPS
setopt HIST_IGNORE_SPACE
setopt HIST_EXPIRE_DUPS_FIRST
setopt SHARE_HISTORY
unsetopt EXTENDED_HISTORY
```

```
PROMPT='%F{green}%n%f@%F{magenta}%m%f %F{blue}%B%~%b%f %# '
RPROMPT='%F{yellow}%?%f'

bindkey "^[[A" history-beginning-search-backward
bindkey "^[[B" history-beginning-search-forward

source /usr/share/zsh-syntax-highlighting/zsh-syntax-highlighting.zsh
ZSH_HIGHLIGHT_HIGHLIGHTERS+=()

export FZF_BASE=~/.config/fzf/bin/fzf

[ -f ~/.fzf.zsh ] && source ~/.fzf.zsh
```

Please give the Server a fixed local IP-Adress.

You can add the ssh login, if you run the following command from the computer you want to access with:

```
ssh-copy-id <usr>@<ip>
```

Website with nginx

Nginx

Install Nginx with

```
sudo apt install nginx
sudo service nginx start
sudo systemctl enable nginx
```

Create a new Configuration File with:

```
sudo vim /etc/nginx/sites-available/mywebsite
```

and test it with `sudo nginx -t` and then restart with `sudo service nginx restart`

PHP

install php 8.3 with

```
apt-get install ca-certificates apt-transport-https software-properties-common
# Add Ondrej's PPA
sudo add-apt-repository ppa:ondrej/php
sudo apt update

# Install new PHP 8.3 packages
sudo apt install php8.3 php8.3-cli php8.3-{bz2,curl,mbstring,intl}

# Install FPM OR Apache module
sudo apt install php8.3-fpm
# OR
# sudo apt install libapache2-mod-php8.2

# On Apache: Enable PHP 8.3 FPM
sudo a2enconf php8.3-fpm
```


Firefly-III

Preparation

```
sudo apt update
sudo apt upgrade -y
sudo apt install -y nginx curl software-properties-common php8.3 php8.3-
{cli,zip,gd,fpm,common,mysql,zip,mbstring,curl,xml,bcmath,imap,ldap,intl} php-json
```

Check to see if php is running

`php -v` and `systemctl status php8.3-fpm`

Adjust some php settings

```
sudo nvim /etc/php/8.3/fpm/php.ini
```

search for and change or enable the following lines of code

```
memory_limit = 512M

[Date]
date.timezone = Europe/Berlin
```

stop apache

```
sudo systemctl stop apache2
sudo systemctl disable apache2
```

remove nginx file

```
sudo rm /etc/nginx/sites-enabled/default
```

create "firefly.dodekaeder.name" in sites-enabled folder and then paste in the config below

```
sudo nvim /etc/nginx/sites-enabled/firefly.dodekaeder.name
```

```

server {
    listen      80 default_server;
    listen      [::]:80 default_server;
    #server_name subdomain.domain.com;
    root        /var/www/firefly-iii/public;
    index index.html index.htm index.php;

    location / {
        try_files $uri /index.php$is_args$args;
        autoindex on;
        sendfile off;
    }

    location ~ \.php$ {
        fastcgi_pass unix:/run/php/php8.3-fpm.sock;
        fastcgi_index index.php;
        fastcgi_read_timeout 240;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
    }

}

```

restart

```
sudo systemctl restart nginx php8.3-fpm
```

If you get an error due to duplicate web servers, you need to remove the symbolic link in sites-available (ls -l) and then rm the link

Install mariaDB

```

curl -Ls https://r.mariadb.com/downloads/mariadb_repo_setup | sudo bash -s -- \
    --mariadb-server-version="mariadb-10.11" --os-type="ubuntu" --os-version="noble"
sudo apt install -y mariadb-server mariadb-client
sudo mysql_secure_installation

```

test installation

```
mysql -u root -p
```

in the mysql shell, check version with the command in bold

```
SELECT VERSION();
```

while still inside the mariaDB shell:

```
CREATE DATABASE firefly_db;  
CREATE USER 'fireflyuser'@'localhost' IDENTIFIED BY 'yourpasswordhere';  
GRANT ALL PRIVILEGES ON firefly_db.* TO 'fireflyuser'@'localhost';  
FLUSH PRIVILEGES;  
exit;  
  
cd ~  
curl -sS https://getcomposer.org/installer -o composer-setup.php  
sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer
```

test composer

```
composer -V
```

Download the latest Version from [here](#).

Unpack it with

```
unzip -o FireflyIII-v6.1.24.zip -d /var/www/firefly-iii  
sudo chown -R elias /var/www/firefly-iiicd /var/www/firefly-iii
```

Change in the ".env" file those lines:

```
DB_HOST=127.0.0.1  
DB_DATABASE=firefly_db  
DB_USERNAME=<usr>  
DB_PASSWORD=<pwd>  
  
TZ=Europe/Berlin
```

Set up Composer and install

```
composer install --no-dev --no-scripts  
php artisan key:generate  
php artisan migrate --seed  
php artisan firefly-iii:decrypt-all
```

```
php artisan cache:clear  
php artisan view:clear  
php artisan firefly-iii:upgrade-database  
php artisan firefly-iii:laravel-passport-keys
```

change the owner of the folder with

```
sudo chown -R www-data:www-data /var/www/firefly-iii  
sudo chmod -R 775 /var/www/firefly-iii/storage
```

uncomment those lines in the file "/etc/locale.gen"

```
de_DE ISO-8859-1  
de_DE.UTF-8 UTF-8  
de_DE@euro ISO-8859-15
```

and generate the locale using

```
sudo locale-gen
```

HA OS in VM

<https://tteck.github.io/Proxmox/#home-assistant-os-vm>

IP 192.168.100

OctoPi

We will use the [Raspberry Pi Imager](#) for installation.

Open the Imager and select the Pi Version and OctoPi image under “Choose OS”, by selecting “Other Specific Purpose OS” > “3D printing” > “OctoPi” and then the “stable” version.

Open advanced options by using the keyboard shortcut ctrl+shift+x and then:

- Configure your wifi options: Set your SSID, password and WiFi country.
- Change the system password in “Set username and password” by entering a new password to use for the system user “pi”. This is not the password you’ll use for logging into OctoPrint but one that you’ll have to use to log into your Pi via SSH should you ever need to.
- Optionally: Change the configured timezone in “Set locale settings”
- Optionally: Change the hostname in “Set hostname”

Install the image to your SD card, then plug everything in to your Raspberry Pi and boot it up. Do not format the SD card after installing, even if prompted to do so. This will break the installation and you will have to start over!

Access OctoPrint from your browser via <http://octopi.local> or the hostname you chose (if your computer supports bonjour) or <http://<your pi's ip address>>. https is available too, with a self-signed certificate (which means your browser will warn you about it being invalid - it isn't, it's just not recognized by your browser).

Grafana

<https://logz.io/blog/grafana-tutorial/>