

# VPS - IONOS

- [vps](#)
- [docker](#)
- [Basic config & Tools](#)
- [Docker Compose](#)

# vps

## ionos - vps - CloudPanel

Hostname	vserver.amrhein.info		
IP	82.165.119.50		
Betriebssystem	Debian 12		
Typ	Virtuelle Maschine		
Größe	VPS 4-8-160		
CPU	4 vCore		
RAM	8 GB		
Datenträger	160 GB NVMe SSD		
Firewall			
TCP	2022	SSH geänderter Port	
TCP	80, 443	NGINX Proxy Manager	
TCP	21115, 21117, 2118,2119	Rustdesk	
TCP, UDP	21116	Rustdesk	
NPM Proxy Manager			
Proxy Hosts			
<u>npm.amrhein.info</u>	http://localhost:81		

<u>book.amrhein.info</u>	http://vserver.amrhein.info:6875		
<u>rustdesk.amrhein.info</u>	http://vserver.amrhein.info:21114	Aktuell nicht verfügbar	Rustdesk OSS
<u>rustadmin.amrhein.info</u>	http://vserver.amrhein.info:21114	Aktuell nicht verfügbar	Rustdesk OSS
<b>Streams</b>			
TCP	21115	vserver.amrhein.info:21115	
TCP, UDP	21116	vserver.amrhein.info:21116	
TCP	21117	vserver.amrhein.info:21117	
TCP	21118	vserver.amrhein.info:21118	
TCP	21119	vserver.amrhein.info:21119	
<b>SSL Certificats</b>			
npm, book, rustdesk, rust admin, vserver, www	Letsencrypt	DNS Challenge STRATO	

# docker

## Installation von Docker

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

# Basic config & Tools

root

```
# ~/.bashrc: executed by bash(1) for non-login shells.

# Note: PS1 and umask are already set in /etc/profile. You should not
# need this unless you want different defaults for root.
PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ '
umask 002

# You may uncomment the following lines if you want `ls' to be colorized:
export LS_OPTIONS='--color=auto'
eval "$(dircolors)"
alias ls='ls $LS_OPTIONS'
alias ll='ls $LS_OPTIONS -l'
alias la='ls $LS_OPTIONS -lA'
#
# Some more alias to avoid making mistakes:
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

. ~/.bash_aliases
```

.bash\_aliases

```
alias ag='alias | grep -i $1'
alias hg='history | grep -i $1'
alias va='vi ~/.bash_aliases && . ~/.bashrc'
alias ..='cd ..'
alias ...='cd ../..'

alias l='ls -ltra'
alias ll='ls -l'
alias la='ls -la'
alias ltra='ls -ltra'
```

```
alias lr='ls -lr'
```

```
alias rebuild='cd ~ && docker-compose down && docker-compose up --no-deps --build -d'
```

```
alias cddocker='cd /opt/docker'
```

```
alias cdbook='cddocker && cd bookstack'
```

```
alias cdrust='cddocker && cd rustdesk'
```

```
alias cdnpm='cddocker && cd proxy-manager'
```

## eigener Benutzer

```
sudo useradd -m -s /bin/bash gerald
```

```
sudo usermod -aG sudo
```

## /etc/sudoers.d/91-gerald

```
gerald ALL=(ALL) NOPASSWD:ALL
```

## Packages

```
sudo apt-get install vim git
```

## VIM Ultimate

```
git clone --depth=1 https://github.com/amix/vimrc.git /opt/vim_runtime
```

```
sh /opt/vim_runtime/install_awesome_parameterized.sh /opt/vim_runtime root gerald
```

```
# to install for all users with home directories, note that root will not be included
```

```
sh /opt/vim_runtime/install_basic_vimrc.sh /opt/vim_runtime --all
```

## Verzeichnisse für Docker

```
sudo -i
```

```
mkdir /opt/docker
```

```
cd /opt
```

```
chown -R root:docker docker
```

```
chmod 775 -R docker
```

```
chmod -R g+s docker
```



# Docker Compose

IDs für bookstack ermitteln

```
vserveramrheininfo:~# id gerald
uid=1000(gerald) gid=1000(gerald) Gruppen=1000(gerald),27(sudo)

vserveramrheininfo:~# grep docker /etc/group
docker:x:994:
```

docker-compose.yml

- NGINX Proxy Manager
- Bookstack
- Rustdesk

```
---
version: "3.8"

services:

  npm:
    image: 'jc21/nginx-proxy-manager:latest'
    restart: unless-stopped
    ports:
      # These ports are in format <host-port>:<container-port>
      - '80:80' # Public HTTP Port
      - '443:443' # Public HTTPS Port
      - '81:81' # Admin Web Port
      # Add any other Stream port you want to expose
      # - '21:21' # FTP

      # Uncomment the next line if you uncomment anything in the section
    environment:
      # Uncomment this if you want to change the location of
      # the SQLite DB file within the container
      # DB_SQLITE_FILE: "/data/database.sqlite"
```



# Uncomment this if IPv6 is not enabled on your host

DISABLE\_IPV6: 'true'

volumes:

- /opt/docker/proxy-manager/data:/data
- /opt/docker/proxy-manager/letsencrypt:/etc/letsencrypt

bookstack:

image: lscr.io/linuxserver/bookstack

container\_name: bookstack

environment:

- PUID=1000
- PGID=994
- TZ=Europe/Berlin
- APP\_URL=https://book.amrhein.info
- DB\_HOST=bookstack\_db
- DB\_PORT=3306
- DB\_USER=bookstack
- DB\_PASS=We\$d45\$\$%
- DB\_DATABASE=bookstackapp

volumes:

- /opt/docker/bookstack:/config

ports:

- 6875:80

restart: unless-stopped

depends\_on:

- bookstack\_db

bookstack\_db:

image: lscr.io/linuxserver/mariadb

container\_name: bookstack\_db

environment:

- PUID=1000
- PGID=994
- TZ=Europe/Berlin
- MYSQL\_ROOT\_PASSWORD=P34we\$\$%
- MYSQL\_DATABASE=bookstackapp
- MYSQL\_USER=bookstack
- MYSQL\_PASSWORD=We\$d45\$\$%

volumes:

- /opt/docker/bookstack:/config

restart: unless-stopped

hbbs:

container\_name: hbbs

image: rustdesk/rustdesk-server:latest

command: hbbs

volumes:

- /opt/docker/rustdesk/data:/root

network\_mode: "host"

depends\_on:

- hbbr

restart: unless-stopped

hbbr:

container\_name: hbbr

image: rustdesk/rustdesk-server:latest

command: hbbr

volumes:

- /opt/docker/rustdesk/data:/root

network\_mode: "host"

restart: unless-stopped