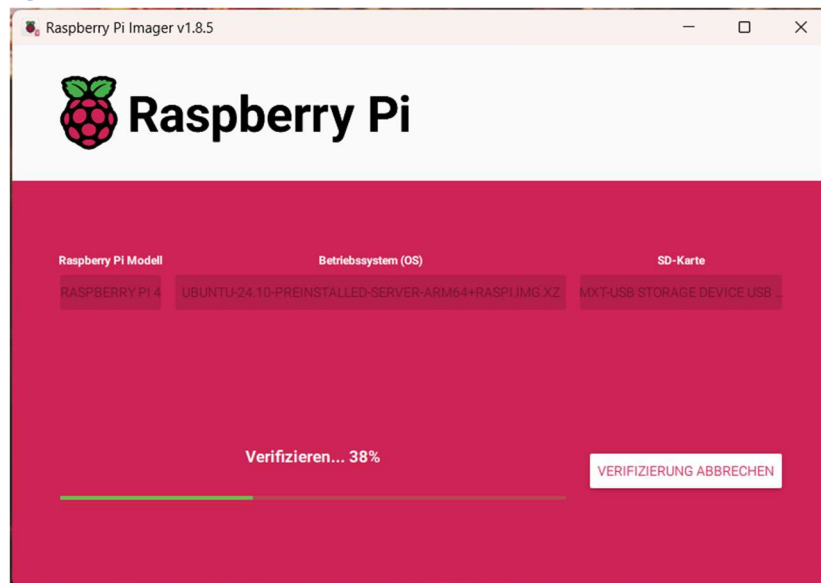


Linux – Windows

Vorbereitung Linux:



Zuerst das Ubuntu-Server Image mit Pi-Imager auf eine SD-Karte schreiben und im Imager die SSH-Verbindung aktivieren. Danach die SD-Karte in den Raspi einlegen und starten. Die IP-Adresse mit einem IP-Scanner ermitteln und über Putty eine SSH-Verbindung aufbauen.

```
adminosaurus@exmachina: ~  
login as: adminosaurus  
adminosaurus@192.168.1.64's password:  
Welcome to Ubuntu 24.10 (GNU/Linux 6.11.0-1004-raspi aarch64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/pro  
  
System information as of Tue Mar 18 08:13:17 CET 2025  
  
System load:  0.87           Temperature:           32.6 C  
Usage of /:   8.5% of 28.69GB Processes:              159  
Memory usage: 7%           Users logged in:       0  
Swap usage:   0%           IPv4 address for wlan0: 192.168.1.64  
  
126 updates can be applied immediately.  
61 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
adminosaurus@exmachina:~$ sudo apt update && sudo apt upgrade -y
```

Nach erfolgter Verbindung und Anmeldung am System wird dieses zuerst aktualisiert.

Samba Installation und Einrichtung:

```

adminosaurus@exmachina:~$ sudo adduser anton
[sudo] password for adminosaurus:
info: Adding user `anton' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `anton' (1004) ...
info: Adding new user `anton' (1004) with group `anton (1004)' ...
info: Creating home directory `/home/anton' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for anton
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
info: Adding new user `anton' to supplemental / extra groups `users' ...
info: Adding user `anton' to group `users' ...
adminosaurus@exmachina:~$

```

Danach werden die User Anton, Anna, Berta und Bert nach diesem Schema erstellt.

```

adminosaurus@exmachina:~$ sudo addgroup agruppe
info: Selecting GID from range 1000 to 59999 ...
info: Adding group `agruppe' (GID 1008) ...
adminosaurus@exmachina:~$ sudo addgroup bgruppe
info: Selecting GID from range 1000 to 59999 ...
info: Adding group `bgruppe' (GID 1009) ...
adminosaurus@exmachina:~$ sudo usermod -aG agruppe anton
adminosaurus@exmachina:~$ sudo usermod -aG agruppe anna
adminosaurus@exmachina:~$ sudo usermod -aG bgruppe bert
adminosaurus@exmachina:~$ sudo usermod -aG bgruppe berta
adminosaurus@exmachina:~$

```

Dann werden die Gruppen agruppe und bgruppe erstellt und die Benutzer entsprechend hinzugefügt.

```

adminosaurus@exmachina:~$ sudo apt install samba samba-common smbclient
Installing:
  samba samba-common smbclient

Installing dependencies:
  attr lib cups2t64 libsmclient0 libtdb1 liburing2 python3-gpg python3-samba python3-tdb samba-libs
  libavahi-client3 libldb2 libtalloc2 libtevent0t64 libwbclient0 python3-ldb python3-talloc samba-common-bin tdb-tools

Suggested packages:
  cups-common ctdb winbind samba-vfs-ceph samba-vfs-glusterfs heimdal-clients cifs-utils

Summary:
  Upgrading: 0, Installing: 21, Removing: 0, Not Upgrading: 3
  Download size: 13.7 MB
  Space needed: 77.1 MB / 26.3 GB available

Continue? [Y/n] y
Get:1 http://ports.ubuntu.com/ubuntu-ports oracular/main arm64 samba-common all 2:4.20.4+dfsg-lubuntul [68.0 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports oracular/main arm64 libavahi-client3 arm64 0.8-13ubuntu6 [27.2 kB]

```

Anschließend wird mit diesem Befehl Samba installiert. Und mit den nachfolgenden Befehlen wird kontrolliert ob die Dienste ordnungsgemäß laufen.

```

adminosaurus@exmachina:~$ sudo service nmbd status
● nmbd.service - Samba NMB Daemon
   Loaded: loaded (/usr/lib/systemd/system/nmbd.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-03-18 08:58:21 CET; 1min 22s ago
   Invocation: 4c8c5c8c140e47c68521c24d83b30fd0
     Docs: man:nmbd(8)
           man:samba(7)
          man:smbr.conf(5)
  Process: 20607 ExecCondition=/usr/share/samba/is-configured nmb (code=exited, status=0/SUCCESS)
 Main PID: 20611 (nmbd)
   Status: "nmbd: ready to serve connections..."
    Tasks: 1 (limit: 3851)
  Memory: 2.9M (peak: 3.2M)
     CPU: 346ms
    CGroup: /system.slice/nmbd.service
            └─20611 /usr/sbin/nmbd --foreground --no-process-group

Mar 18 08:58:21 exmachina systemd[1]: Starting nmbd.service - Samba NMB Daemon...
Mar 18 08:58:21 exmachina systemd[1]: Started nmbd.service - Samba NMB Daemon.
adminosaurus@exmachina:~$ sudo service smb status
● smb.service - Samba SMB Daemon
   Loaded: loaded (/usr/lib/systemd/system/smbd.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-03-18 08:58:18 CET; 1min 38s ago
   Invocation: 27b43b44e1784003954f757ca0289dc6
     Docs: man:smbd(8)
           man:samba(7)
          man:smbr.conf(5)
  Main PID: 20543 (smbd)
   Status: "smbd: ready to serve connections..."
    Tasks: 3 (limit: 3851)
  Memory: 10.8M (peak: 11.2M)
     CPU: 54ms
    CGroup: /system.slice/smbd.service
            └─20543 /usr/sbin/smbd --foreground --no-process-group
                └─20546 "smbd: notifyd"
                  └─20547 "smbd: cleanupd"

```

```
adminosaurus@exmachina:~$ sudo smbpasswd -a anton
New SMB password:
Retype new SMB password:
Added user anton.
```

Mit diesem Befehl werden die User zu Samba-Usern. (Ist bei allen Samba-Usern zu wiederholen).

```
adminosaurus@exmachina:/home$ mkdir shareall
mkdir: cannot create directory 'shareall': Permission denied
adminosaurus@exmachina:/home$ sudo mkdir shareall
adminosaurus@exmachina:/home$ ls
adminosaurus  anna  anton  bert  berta  shareall
adminosaurus@exmachina:/home$ sudo mkdir sharea
adminosaurus@exmachina:/home$ sudo mkdir shareb
adminosaurus@exmachina:/home$ ls
adminosaurus  anna  anton  bert  berta  sharea  shareall  shareb
adminosaurus@exmachina:/home$ █
```

Dann werden die Ordner shareall, sharea und shareb erstellt.

```
[shareall]
comment = Samba Freigabe Alle
path = home/shareall
readonly = no
browseable = yes

[sharea]
comment = Samba Freigabe GruppeA
path = home/sharea
readonly = no
browseable = yes
create mask = 0660
directory mask = 2770
valid users = @agruppe
force group = agruppe

[shareb]
comment = Samba Freigabe GruppeB
path = home/shareb
readonly = no
browseable = yes
create mask = 0660
directory mask = 2770
valid users = @bgruppe
force group = bgruppe
█
```

Ins Root-Verzeichnis wechseln und mit „sudo nano etc/samba/smb.conf“ in den Nano-Editor wechseln und die Freigaben einrichten

```
adminosaurus@exmachina:/home$ sudo chown -R berta:bgruppe shareb/
[sudo] password for adminosaurus:
adminosaurus@exmachina:/home$ sudo chown -R bert:bgruppe shareb/
adminosaurus@exmachina:/home$ sudo chown -R anton:agruppe sharea/
adminosaurus@exmachina:/home$ sudo chown -R anna:agruppe sharea/
adminosaurus@exmachina:/home$
```

Hier werden die Gruppenbesitzer festgelegt.

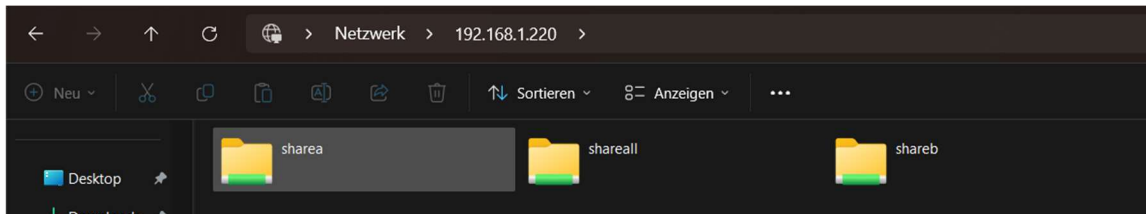
```
adminosaurus@exmachina:/home$ sudo chmod -R 2770 sharea/
adminosaurus@exmachina:/home$ sudo chmod -R 2770 shareb/
adminosaurus@exmachina:/home$ ls
```

Anpassung der Rechte und Sicherheit für die Gruppen sharea und shareb. Die Gruppe sharea kann mit „sudo chmod 777 sharea/“ freigegeben werden.

Mit all diesen Anpassungen wird sichergestellt, dass User aus der agruppe nur auf sharea und sharea zugreifen können und User aus der bgruppe nur auf shareb und sharea zugreifen können.



Unter Windows mit der Adresse des Servers verbinden und danach User und Passwort eingeben.



Installation/Einrichtung von Apache:

```
adminosaurus@exmachina:/home$ sudo apt install apache2
Installing:
  apache2

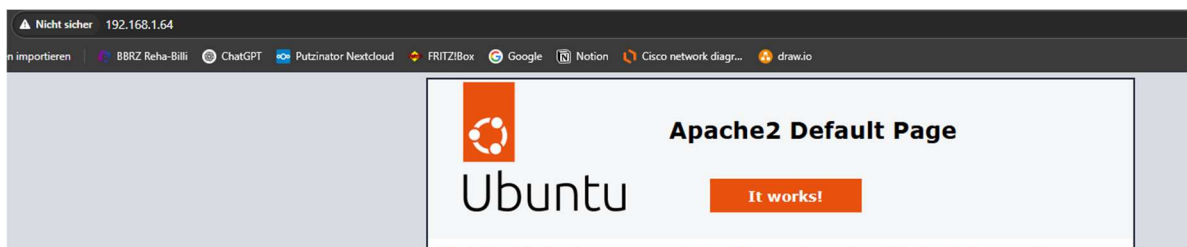
Installing dependencies:
  apache2-bin  apache2-data  apache2-utils  libapr1t64  libaprutil1-dbd-sqlite3  libaprutil1-ldap  libaprutil1t64  liblua5.4-0  ssl-cert

Suggested packages:
  apache2-doc  apache2-suexec-pristine  | apache2-suexec-custom  www-browser

Summary:
  Upgrading: 0, Installing: 10, Removing: 0, Not Upgrading: 3
  Download size: 2076 kB
  Space needed: 14.0 MB / 26.2 GB available

Continue? [Y/n] y
Get:1 http://ports.ubuntu.com/ubuntu-ports oracular/main arm64 libapr1t64 arm64 1.7.2-3ubuntu1 [106 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports oracular/main arm64 libaprutil1t64 arm64 1.6.3-3ubuntu1 [94.9 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports oracular/main arm64 libaprutil1-dbd-sqlite3 arm64 1.6.3-3ubuntu1 [11.4 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports oracular/main arm64 libaprutil1-ldap arm64 1.6.3-3ubuntu1 [9114 B]
Get:5 http://ports.ubuntu.com/ubuntu-ports oracular/main arm64 liblua5.4-0 arm64 5.4.6-3build2 [158 kB]
```

Mit diesem Befehl wird Apache installiert.



Wird jetzt die IP-Adresse des Raspi auf einem Gerät im selben Netzwerk eingegeben erscheint diese Website, welche bestätigt, dass der Apache Web-Server funktioniert.

```
adminosaurus@exmachina:/var/www/html$ sudo mv index.html index.old
adminosaurus@exmachina:/var/www/html$ ls
index.old
```

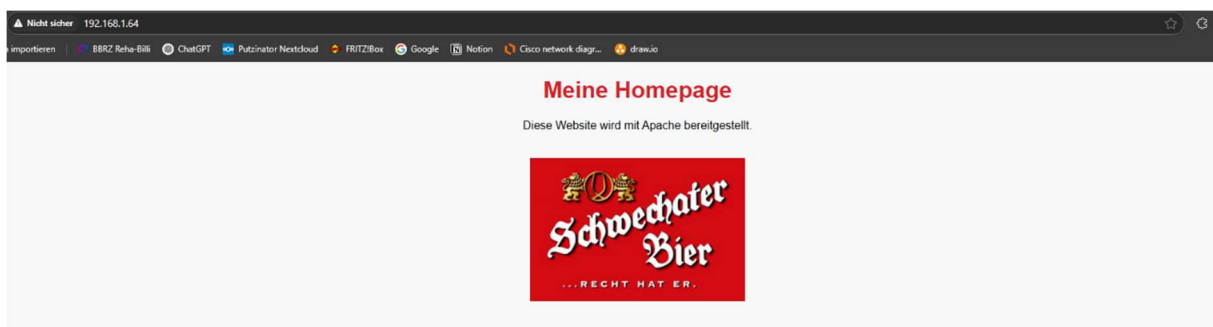
Will man die Originale Startseite behalten kann man diese einfach umbenennen.

```

GNU nano 8.1 index.html
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Meine Homepage</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      background-color: #f8f8f8;
    }
    h1 {
      color: #d2232a;
    }
    img {
      width: 300px;
      margin-top: 20px;
    }
  </style>
</head>
<body>
  <h1>Meine Homepage</h1>
  <p>Diese Website wird mit Apache bereitgestellt.</p>
  
</body>
</html>

```

Danach mit „sudo nano index.html“ wird eine neue Datei erstellt, wo dann der HTML-Text eingegeben wird. Will man ein Bild hinzufügen, muss der Dateiname angegeben werden und das Foto unter /var/www/html gespeichert werden. Danach den Editor verlassen und speichern.



Geht man dann wieder auf die IP-Adresse des Raspi, kommt die Seite, welche man erstellt hat.

Installation von Webmin:

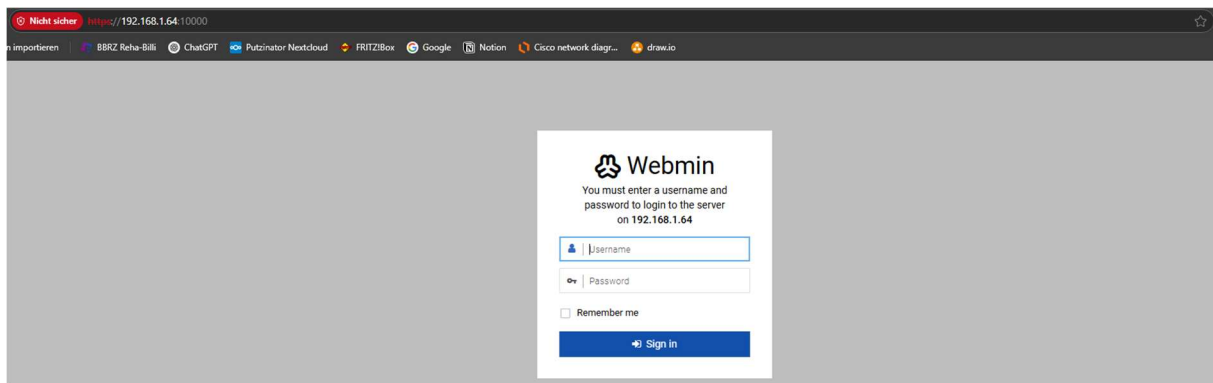
Um Webmin zu installieren, müssen folgende Befehle eingegeben werden:

```

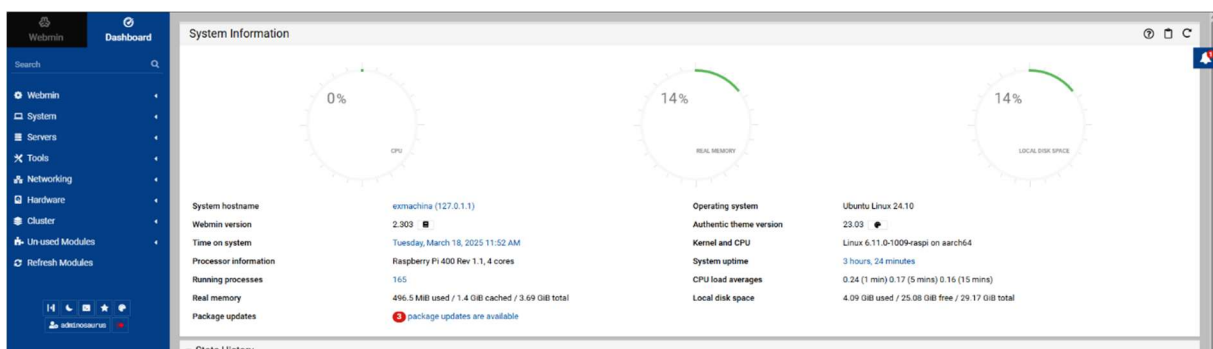
adminosaurus@exmachina: /var/www/html$ curl -o webmin-setup-repo.sh https://raw.githubusercontent.com/webmin/webmin/master/webmin-setup-repo.sh sh webmin-setup-repo.sh
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0
Warning: Failed to open the file webmin-setup-repo.sh: Permission denied
curl: (23) client returned ERROR on write of 810 bytes
0 16246 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0
curl: (6) Could not resolve host: sh
curl: (6) Could not resolve host: webmin-setup-repo.sh
adminosaurus@exmachina: /var/www/html$ sudo curl -o webmin-setup-repo.sh https://raw.githubusercontent.com/webmin/webmin/master/webmin-setup-repo.sh sh webmin-setup-repo.sh
[sudo] password for adminosaurus:
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 16246 100 16246 0 0 106k 0 --:--:-- --:--:-- --:--:-- 106k
curl: (6) Could not resolve host: sh
curl: (6) Could not resolve host: webmin-setup-repo.sh
adminosaurus@exmachina: /var/www/html$ sudo curl -o webmin-setup-repo.sh https://raw.githubusercontent.com/webmin/webmin/master/webmin-setup-repo.sh
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 16246 100 16246 0 0 102k 0 --:--:-- --:--:-- --:--:-- 103k

```

- **sudo curl -o webmin-setup-repo.sh https://raw.githubusercontent.com/webmin/webmin/master/webmin-setup-repo.sh**
- **sudo sh webmin-setup-repo.sh**
- **sudo apt-get install webmin --install-recommends**



Nach der Installation gelangt man mit einem Gerät im selben Netzwerk im Browser mit der Adresse <https://Server-IP:10000> zum Anmeldefenster von Webmin. Die Anmeldedaten sind die vom Administrator des Servers.



Hier ist das Dashboard von Webmin ersichtlich.

Installation von Cockpit:

```
adminosaurus@exmachina:/var/www/html$ sudo apt install cockpit
[sudo] password for adminosaurus:
Installing:
  cockpit

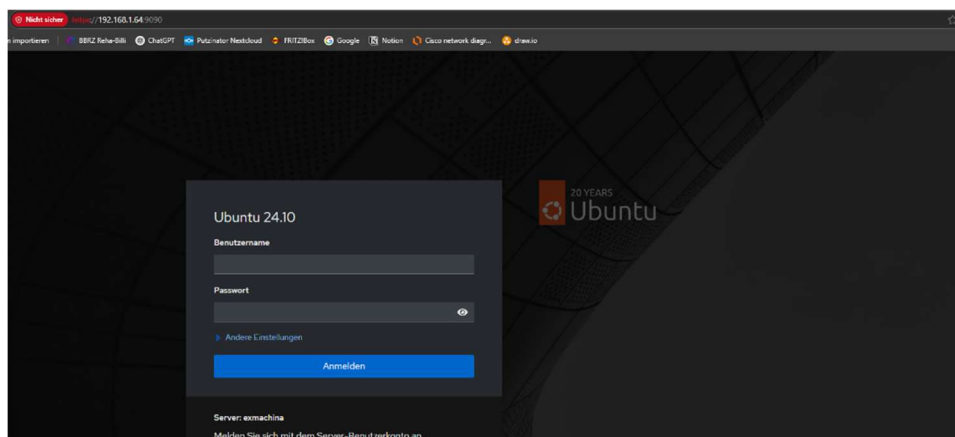
Installing dependencies:
  cockpit-bridge  cockpit-system  dnconf-service  glib-networking-common  libcrack2  libproxyv5  libssh-4  ppp
  cockpit-networkmanager  cockpit-ws  dns-root-data  glib-networking-services  libedconf1  libpwquality-common  libteamdt10  pptp-linux
  cockpit-packagekit  cracklib-runtime  dnsmasq-base  gsettings-desktop-schemas  libndp0  libpwquality-tools  network-manager  wamerican
  cockpit-storaged  dconf-gsettings-backend  glib-networking  libbluetooth3  libnm0  libpwquality1  network-manager-pptp

Suggested packages:
  cockpit-doc  cockpit-sosreport  xdg-utils  udisks2-btrfs  udisks2-lvm2  sssd-dbus  avahi-autoipd  libteam-utils

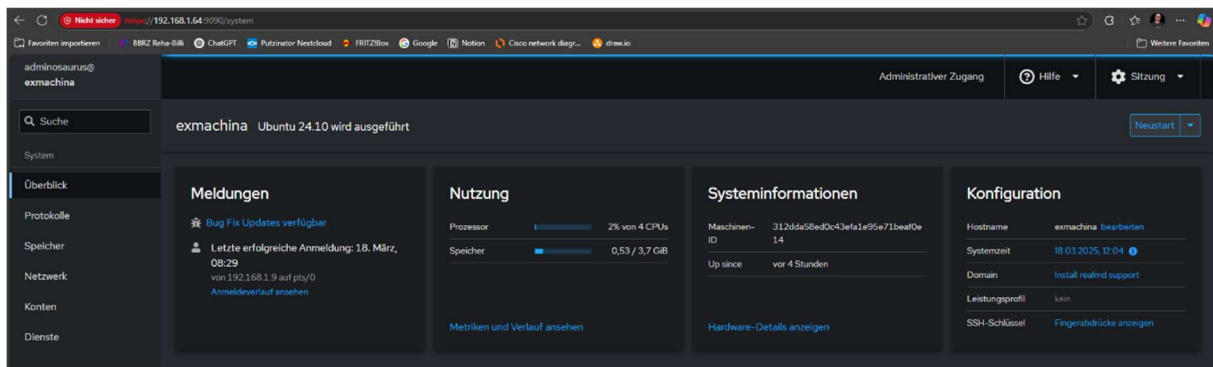
Summary:
  Upgrading: 0, Installing: 32, Removing: 0, Not Upgrading: 3
  Download size: 11.6 MB
  Space needed: 27.1 MB / 26.6 GB available

Continue? [Y/n] y
Get:1 http://ports.ubuntu.com/ubuntu-ports oracular/main arm64 libssh-4 arm64 0.10.6-3ubuntu1 [189 kB]
Get:2 http://ports.ubuntu.com/ubuntu-ports oracular/main arm64 libproxyv5 arm64 0.5.8-1 [27.2 kB]
```

Cockpit wird mit dem Befehl „sudo apt install cockpit“ installiert.

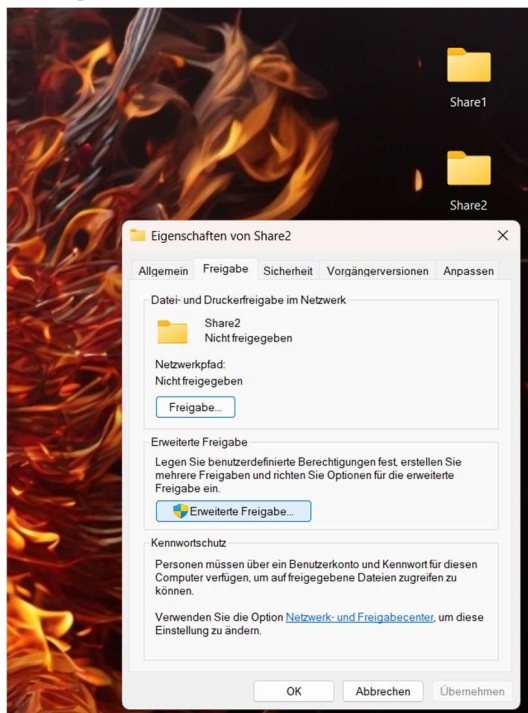


Die Anmeldung erfolgt gleich wie bei Webmin nur wird hier der Port 9090 verwendet.

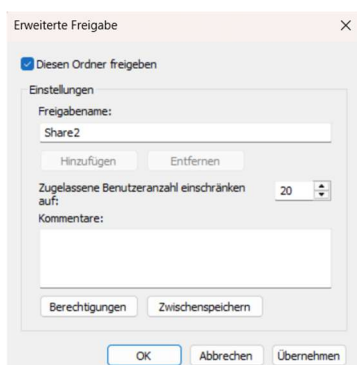


Nach der Anmeldung ist dieser Überblick ersichtlich.

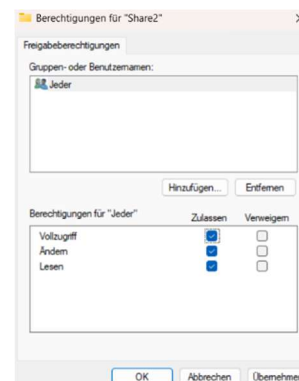
Freigabeorder in Windows erstellen:



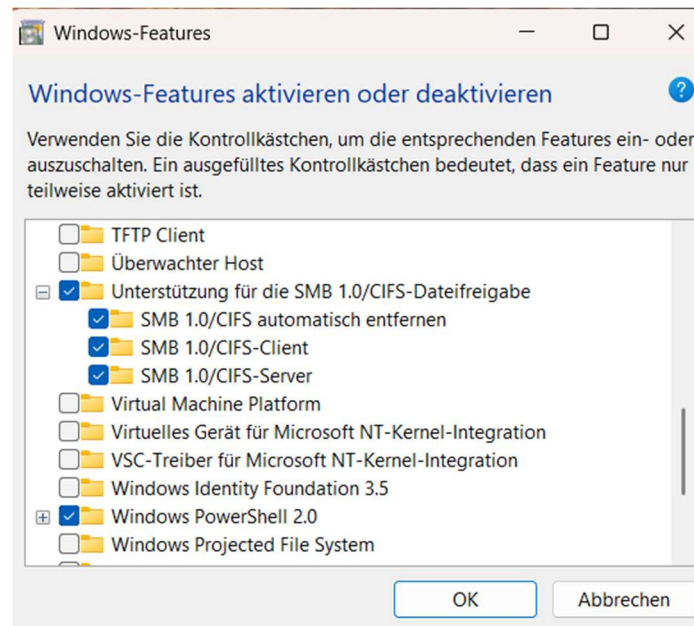
Zuerst werden die zwei Ordner „Share1“ und „Share2“ erstellt. Danach mit rechtsklick in die Ordneigenschaften. Im Reiter „Freigabe“ die „Erweiterte Freigabe“ auswählen.



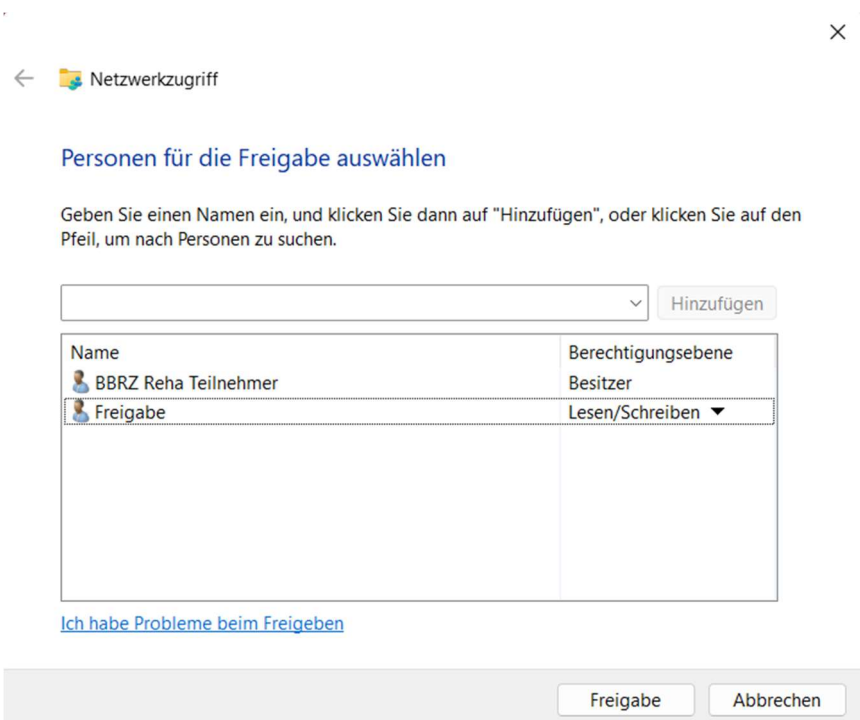
Hier auf „Diesen Ordner freigeben“



Hier für „Jeder“ Vollzugriff zulassen.



Hier muss die Unterstützung für die SMB 1.0/CIFS-Dateifreigabe aktiviert sein. Um in die Windows-Features zu kommen kann man in der Windows-Suchleiste „Features“ eingeben, es wird dann ganz oben vorgeschlagen. Um von einem anderen System auf die Ordner zugreifen zu können muss ein lokaler User erstellt werden, falls keiner existiert.



Sollten beim Zugriff Probleme auftauchen muss in den Ordneigenschaften im Reiter „Freigabe“ unter „Freigabe“ auch der User, welcher verwendet wird, eingetragen werden und die Rechte „Lesen/Schreiben“ vergeben.

CIFS-Mount:

```
Web console: https://exmachina:9090/ or https://192.168.1.64:9090/

Last login: Tue Mar 25 09:09:25 2025 from 192.168.1.9
adminosaurus@exmachina:~$ sudo mount -t cifs -o user=Freigabe,password=Aa_123456 //192.168.1.9/Share1 /mnt/Share1
[sudo] password for adminosaurus:
adminosaurus@exmachina:~$
```

Mit dem Befehl „sudo apt-get install cifs-utils“ werden die CIFS-Tools installiert und mit dem Befehl „sudo mount -t cifs -o user=Windowsuser, password=Passwort //IP-Adresse/Freigabeordner /mnt/Ordner“ wird der Freigabeordner mit dem ausgewählten Ordner im Linux-System gemountet.

Um einen Freigegebenen Netzwerkordner automatisch einzubinden, wird zuerst der Ordner „Share2“ im Ordner /mnt/ erstellen. Danach mit dem Befehl „sudo nano /etc/fstab“ den Editor öffnen und folgenden Eintrag hinzufügen:

```
adminosaurus@exmachina: /mnt/Share2
GNU nano 8.1 /etc/fstab
#LABEL=writeable / ext4 defaults 0 1
#LABEL=system-boot /boot/firmware vfat defaults 0 1

# SMB-Freigabe Windows:
//192.168.1.9/Share2 /mnt/Share2 cifs _netdev,user=Freigabe,password=Aa_123456
```

Mit diesem Eintrag wird die Freigabe automatisch gemountet.

```
adminosaurus@exmachina:/mnt$ ls
Share1 Share2
adminosaurus@exmachina:/mnt$ cd Share1
adminosaurus@exmachina:/mnt/Share1$ ls
adminosaurus@exmachina:/mnt/Share1$ cd ..
adminosaurus@exmachina:/mnt$ cd Share2
adminosaurus@exmachina:/mnt/Share2$ ls
Test
adminosaurus@exmachina:/mnt/Share2$
```

In beiden Share-Ordnern existiert ein Test-Ordner. Wie hier zu sehen ist wird nach einem Neustart nur Share2 gemountet da dieser in der „fstab“ hinterlegt ist.

Docker Installation:

Zuerst mit den Befehlen „sudo apt-get update && sudo apt-get upgrade -y“ das System auf den neuesten Stand bringen. Danach mit dem Befehl „sudo apt-get install ca-certificates curl“

```
adminosaurus@exmachina:/$ sudo install -m 0755 -d /etc/apt/keyrings
adminosaurus@exmachina:/$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
adminosaurus@exmachina:/$
adminosaurus@exmachina:/$ sudo chmod a+r /etc/apt/keyrings/docker.asc
adminosaurus@exmachina:/$ echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
adminosaurus@exmachina:/$
adminosaurus@exmachina:/$ sudo apt-get update
Hit:1 http://ports.ubuntu.com/ubuntu-ports oracular InRelease
Hit:2 http://ports.ubuntu.com/ubuntu-ports oracular-updates InRelease
Hit:3 http://ports.ubuntu.com/ubuntu-ports oracular-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu oracular InRelease
Hit:5 http://ports.ubuntu.com/ubuntu-ports oracular-security InRelease
Ign:6 https://download.webmin.com/download/newkey/repository stable InRelease
Hit:7 https://download.webmin.com/download/newkey/repository stable Release
Reading package lists... Done
```

Danach müssen die oben gezeigten Befehle eingegeben werden, um die Docker-Installation vorzubereiten.

```
adminosaurus@exmachina:/$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras libltdl7 libsllp0 pigz sllp4netns
Suggested packages:
  cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libsllp0 pigz sllp4netns
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 104 MB of archives.
After this operation, 408 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ports.ubuntu.com/ubuntu-ports oracular/universe arm64 pigz arm64 2.8-1 [60.7 kB]
Get:2 https://download.docker.com/linux/ubuntu oracular/stable arm64 containerd.io arm64 1.7.26-1 [22.8 MB]
```

Mit diesem Befehl wird dann Docker heruntergeladen und installiert.

```
adminosaurus@exmachina:/$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c9c5fd25a1bd: Pull complete
Digest: sha256:7e1a4e2d11e2ac7a8c3f768d4166c2defeb09d2a750b010412b6ea13de1efb19
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (arm64v8)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Mit dem Befehl „sudo docker run hello-world“ wird getestet, ob die Docker-Installation erfolgreich war.

Installation Portainer über Docker:

```
adminosaurus@exmachina:/$ sudo docker volume create portainer_data
portainer_data
```

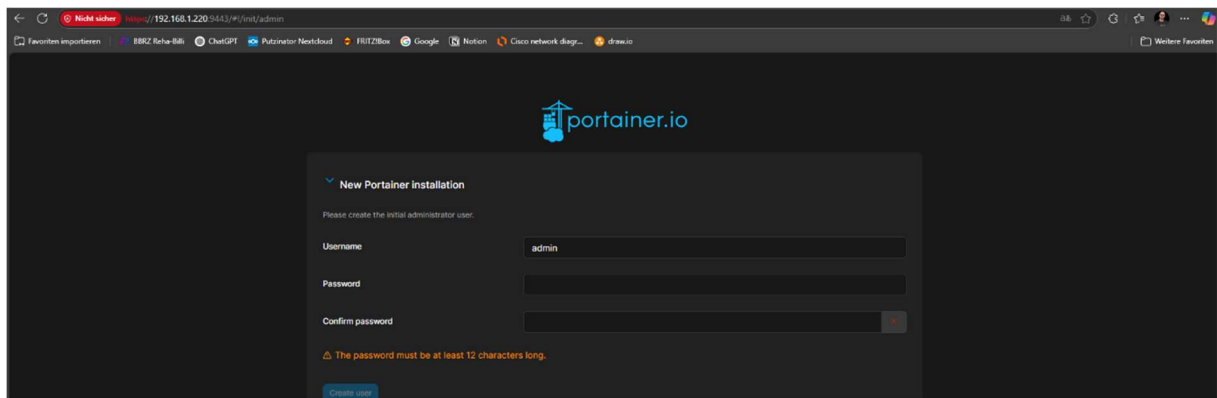
Mit diesem Befehl wird ein Volume erstellt auf dem Portainer seine Datenbank speichert.

```
adminosaurus@exmachina:/$ sudo docker run -d -p 8000:8000 -p 9443:9443 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest
Unable to find image 'portainer/portainer-ce:latest' locally
latest: Pulling from portainer/portainer-ce
e2e06b27b87e: Pull complete
1fed1531b45b: Pull complete
354e5ce52589: Pull complete
b2203424d48c: Pull complete
e811f1a0940a: Pull complete
dc04090222c2: Pull complete
d0ea124b8141: Pull complete
f9da666c8124: Pull complete
29adf12eac77: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:7f10a26bfdda3fc58295ea09b860117ecd86a642d66fb94celf27a4c221d4649
Status: Downloaded newer image for portainer/portainer-ce:latest
d7166a9a9a7938775e79f8e77cbe70c114e7abb53115291cd3f35f2cf4c17925
```

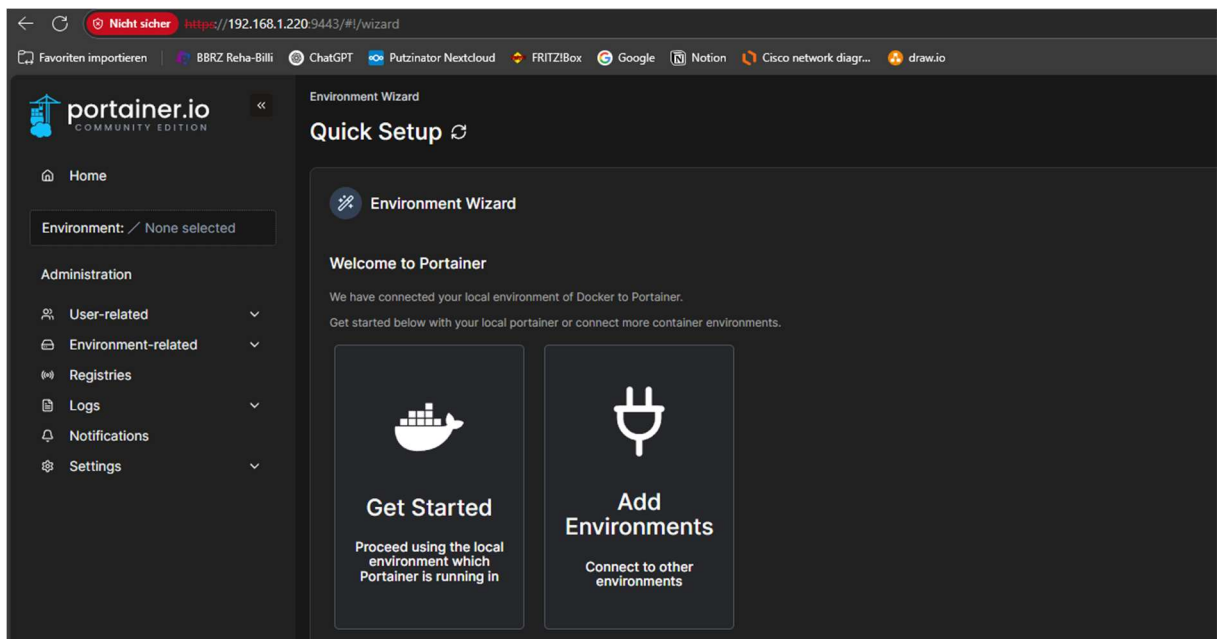
Mit diesem Befehl wird Portainer heruntergeladen und installiert.

```
adminosaurus@exmachina:/$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
d7166a9a9a79   portainer/portainer-ce:latest       "/portainer"            3 minutes ago   Up 2 minutes   0.0.0.0:8000->8000/tcp, [::]:8000->8000/tcp, 0.0.0.0:9443->9443/tcp, [::]:9443->9443/tcp, 9000/tcp
adminosaurus@exmachina:/$
```

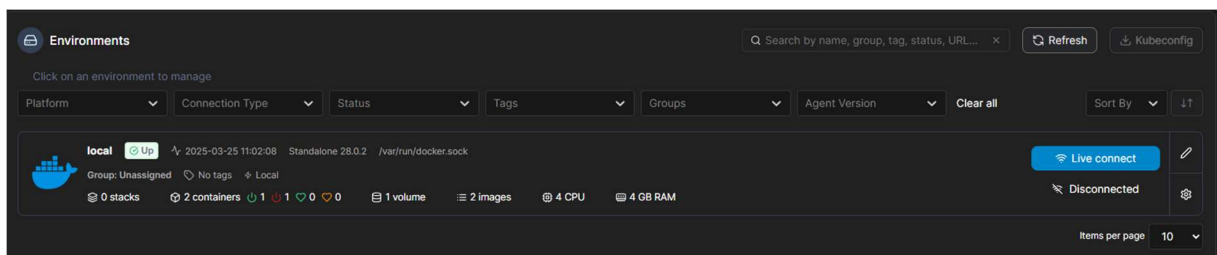
Mit diesem Befehl wird überprüft, ob Portainer läuft.



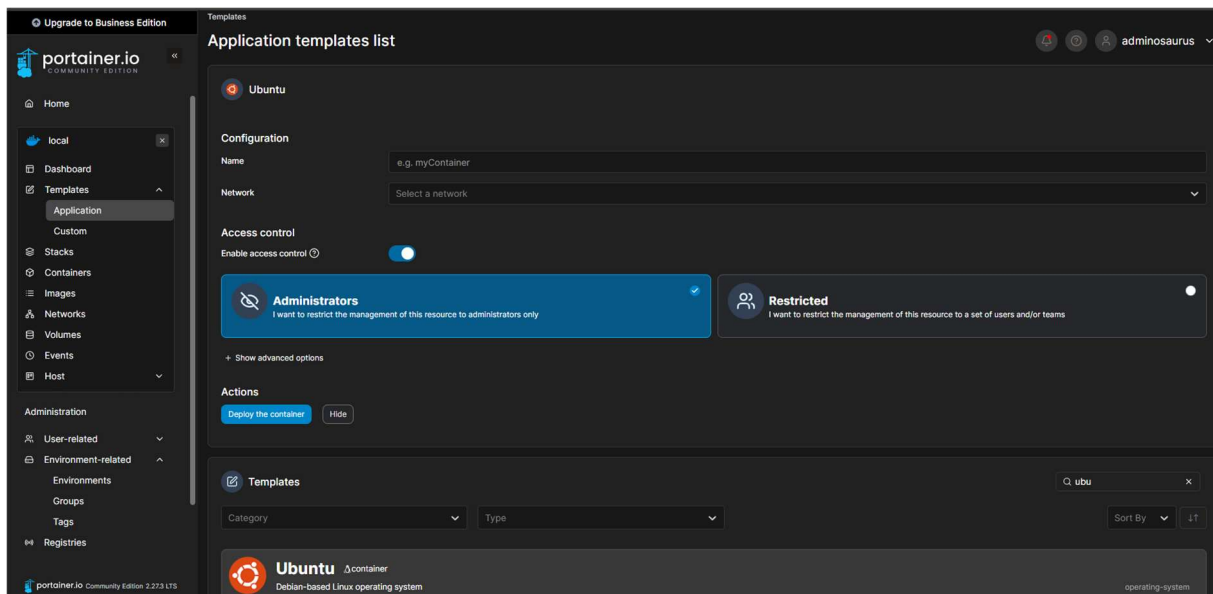
Mit der Adresse: <https://ServerIP:9443> kommt man zum Login Screen von Portainer. Hier muss erst ein User angelegt werden. Nach



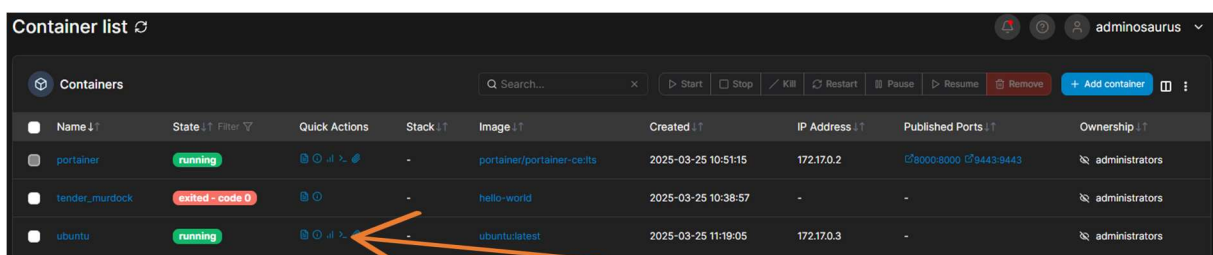
Nach erfolgreicher Anmeldung kommt man zu dieser Startseite. Auf Get Started klicken, um ein Environment auszuwählen bzw. um sich damit zu verbinden.



Hier ist auf „Live connect“ zu klicken, um sich zu verbinden.



Unter Templates finden sich verschiedenste Anwendungen, welche installiert werden können, in diesem Fall wähle ich Ubuntu. Als Name vergebe ich „ubuntu“ und als Network „bridge“.



Unter Containers findet man nun die Ubuntu Maschine. Klickt man auf dieses Symbol (>_) wird eine Terminal-Sitzung aufgebaut.

Ubuntu Historie:

Die Historie wird mit dem Befehl „history > history.txt“ als Textdatei mit dem Namen „history.txt“ gespeichert. Hier ist der komplette Verlauf der Ubuntu Server Maschine:

- 1 ip a
- 2 sudo apt update && sudo apt upgrade -y
- 3 ping 8.8.8.8
- 4 sudo apt update && sudo apt upgrade -y
- 5 sudo reboot now
- 6 sudo apt update && sudo apt upgrade -y
- 7 sudo adduser anton
- 8 sudo adduser anna
- 9 sudo adduser berta
- 10 sudo adduser bert
- 11 sudo addgroup agruppe

```
12 sudo addgroup bgruppe
13 sudo usermod -aG agruppe anton
14 sudo usermod -aG agruppe anna
15 sudo usermod -aG bgruppe bert
16 sudo usermod -aG bgruppe berta
17 sudo apt install samba samba-common smbclient
18 sudo service nmbd status
19 sudo service smbd status
20 sudo smbpasswd -a anton
21 sudo smbpasswd -a anna
22 sudo smbpasswd -a bert
23 sudo smbpasswd -a berta
24 cd /
25 sudo nano etc/samba/smb.conf
26 ls
27 cd home/
28 ls
29 mkdir shareall
30 sudo mkdir shareall
31 ls
32 sudo mkdir sharea
33 sudo mkdir shareb
34 ls
35 sudo chmod 777 shareall
36 sudo chmod 777 sharea
37 sudo chmod 777 shareb
38 ls
39 cd /
40 sudo nano etc/samba/smb.conf
41 ls
42 cd home/
43 ls
```



```
44 sudo chown anton:agruppe sharea/
45 sudo chown anna:agruppe sharea/
46 sudo chown bert:bgruppe shareb/
47 sudo chown berta:bgruppe shareb/
48 sudo -R chown berta:bgruppe shareb/
49 ls
50 sudo chown -R berta:bgruppe shareb/
51 sudo chown -R bert:bgruppe shareb/
52 sudo chown -R anton:agruppe sharea/
53 sudo chown -R anna:agruppe sharea/
54 cd /
55 sudo nano etc/samba/smb.conf
56 cd home/
57 ls
58 sudo chmod -R 2770 sharea/
59 sudo chmod -R 2770 shareb/
60 ls
61 sudo apt install apache2
62 cd /var/www/html/
63 ls
64 cat index.html
65 ls
66 mv index.html index.old
67 sudo mv index.html index.old
68 ls
69 sudo nano index.html
70 ls
71 cd /home/
72 ls
73 cd shareall/
74 ls
75 mv schwechater-bier.jpg /var/www/html/
```

```
76 sudo mv schwechater-bier.jpg /var/www/html/
77 ls
78 cd /var/www/html/
79 ls
80 sudo nano index.html
81 curl -o webmin-setup-repo.sh
https://raw.githubusercontent.com/webmin/webmin/master/webmin-setup-repo.sh sh webmin-
setup-repo.sh
82 sudo curl -o webmin-setup-repo.sh
https://raw.githubusercontent.com/webmin/webmin/master/webmin-setup-repo.sh sh webmin-
setup-repo.sh
83 sudo curl -o webmin-setup-repo.sh
https://raw.githubusercontent.com/webmin/webmin/master/webmin-setup-repo.sh
84 sh webmin-setup-repo.sh
85 sudo sh webmin-setup-repo.sh
86 sudo apt-get install webmin --install-recommends
87 sudo apt install cockpit
88 sudo shutdown now
89 sudo apt-get install cifs-utils
90 sudo mount -t cifs -o user=Freigabe //192.168.1.64/Share1 /mnt
91 ls /
92 cd mnt
93 ls
94 cd /
95 ls
96 cd mnt/
97 ls
98 cd /
99 sudo mount -t cifs -o user=Freigabe //PC86B2F5913145/Share1 /mnt
100 Aa_123456
101 ls
102 cd mnt/
103 ls
```

```
104 sudo mkdir Share1
105 ls
106 cd /
107 sudo mount -t cifs -o user=Freigabe //192.168.1.64/Share1 /mnt
108 sudo mount -t cifs -o user=Freigabe //192.168.1.9/Share1 /mnt
109 sudo mount -t cifs -o user=Freigabe //192.168.1.9/Share1 /mnt/Share1
110 sudo mount -t cifs -o user=Freigabe //PC86B2F5913145/Share1 /mnt/Share1
111 sudo mount -t cifs -o user=Freigabe,domain=WORKGROUP //PC86B2F5913145/Share1
/mnt/Share1
112 sudo mount -t cifs -o user=Freigabe,domain=WORKGROUP //192.168.1.9/Share1 /mnt/Share1
113 sudo mount -t cifs -o //192.168.1.9/Share1 /mnt/Share1
114 sudo mount -t cifs -o //192.168.1.9/Share1 /mnt
115 sudo mount -t cifs -o user=Freigabe,password=Aa_123456 //192.168.1.9/Share1 /mnt/Share1
116 sudo reboot now
117 sudo mount -t cifs -o user=Freigabe,password=Aa_123456 //192.168.1.9/Share1 /mnt/Share1
118 ls /mnt/
119 cd /mnt/
120 cd Share1/
121 ls
122 mkdir Test
123 sudo mkdir Test
124 ls
125 cd /
126 cd etc/
127 ls
128 sudo nano fstab
129 cd /
130 cd mnt/
131 ls
132 sudo mkdir Share2
133 ls
134 sudo mount -a
```

```
135 sudo systemctl daemon-reload
136 sudo mount -a
137 ls
138 cd Share2/
139 ls
140 sudo nano /etc/fstab
141 sudo reboot now
142 cd /mnt/
143 ls
144 cd Share1
145 ls
146 cd ..
147 cd Share2
148 ls
149 cd /usr/local/bin/
150 ls
151 cd /
152 sudo apt-get install ca-certificates curl gnupg lsb-release
153 sudo mkdir -p /etc/apt/keyrings
154 curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
155 cd /etc/apt/
156 ls
157 cd /
158 sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
159 y
160 cd /etc/apt/
161 cd keyrings/
162 ls
163 echo
164 "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
```

```

$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

165 cd /

166 sudo apt-get install docker-ce dock ber-ce-cli containerd.io docker-compose-plugin

167 sudo apt install -y docker-ce docker-ce-cli containerd.io

168 sudo apt remove docker-desktop

169 rm -r $HOME/.docker/desktop

170 sudo rm /usr/local/bin/com.docker.cli

171 sudo apt purge docker-desktop

172 sudo apt-get install ca-certificates curl gnupg lsb-release

173 sudo mkdir -p /etc/apt/keyrings

174 curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg

175 echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \

$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

176 apt-cache madison docker-ce | awk '{ print $3 }'

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

178 for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker containerd
runc; do sudo apt-get remove $pkg; done

179 sudo apt-get update

180 sudo apt-get install ca-certificates curl

181 sudo apt-get upgrade -y

182 sudo apt-get install ca-certificates curl

183 sudo install -m 0755 -d /etc/apt/keyrings

184 sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

185 sudo chmod a+r /etc/apt/keyrings/docker.asc

186 echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \

$(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null

187 sudo apt-get update

188 docker volume create portainer_data

189 sudo docker volume create portainer_data

```


190 ls

191 sudo docker run -d -p 8000:8000 -p 9443:9443 --name portainer --restart=always -v
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest

192 docker ps

193 sudo docker ps

194 [200~docker stop portainer

195 sudo docker stop portainer

196 sudo docker start portainer

197 cd /mnt/Share2

198 history > history.txt

199 sudo history > history.txt

200 ls

201 cd Test

202 ls

203 sudo history > history.txt

204 cd ~

205 ls

206 history > history.txt