

# Das Client-Management-System OPSI

Bei diesem Server handelt es sich um eine Art mächtigen-WDS, der mehr kann als nur Windows Betriebssysteme ausrollen, sondern zusätzlich auch Linux Clients. Dazu kommt noch eine automatische Softwareverteilung

## Installation

Um opsi zu installieren habe ich mir folgenden Container erstellt. Ganz wichtig für Domänenmitgliedschaft und Docker Es muss ein privilegierter Container sein

Create: LXC Container

General

Template

Disks

CPU

Memory

Network

DNS

Confirm

Key ↑	Value
cores	2
hostname	Yukikaze
memory	2048
nameserver	192.168.2.7
net0	name=eth0,bridge=vbr0,firewall=1,ip6=dhcp,ip=dhcp
nodename	soryu
ostemplate	local:vztmpl/ubuntu-22.04-standard_22.04-1_amd64.tar.zst
pool	
rootfs	local:30
searchdomain	sotoba.de
ssh-public-keys	
swap	2048
vmid	101

☐ Start after created

Advanced ☒

Back

Finish

```
apt-get update && apt-get upgrade && reboot
```

## Nach dem Neustart installiere ich das Programm curl

```
root@Yukikaze:~# apt install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libbrotli1 libcurl4 libldap-2.5-0 libldap-common librtmp1 libssh-4
The following NEW packages will be installed:
  curl libbrotli1 libcurl4 libldap-2.5-0 libldap-common librtmp1 libssh-4
0 upgraded, 7 newly installed, 0 to remove and 5 not upgraded.
Need to get 1243 kB of archives.
After this operation, 3403 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

## Active Directory

Und danach lade ich den Container in die Domäne ein. Dazu verwende ich folgende Befehle.

Samba installieren

```
apt install sssd-ad sssd-tools adcli realmd krb5-user libnss-sss libpam-sss  
samba-common-bin oddjob oddjob-mkhomedir
```

Die Erreichbarkeit der Domäne testen

```
realm discover deinedomaene.local
```

```
root@Yukikaze:~# realm discover sotoba.de  
Sotoba.de  
type: kerberos  
realm-name: SOTOBA.DE  
domain-name: Sotoba.de  
configured: no  
server-software: active-directory  
client-software: sssd  
required-package: sssd-tools  
required-package: sssd  
required-package: libnss-sss  
required-package: libpam-sss  
required-package: adcli  
required-package: samba-common-bin  
sotoba.de  
type: kerberos  
realm-name: SOTOBA.DE  
domain-name: sotoba.de  
configured: no  
root@Yukikaze:~#
```

Mit dieser Ausgabe sehe ich das Samba installiert und die Domäne erkannt wurde. Jetzt werde ich den Container der Domäne hinzufügen

```
realm join deinedomaene.local
```

```
root@Yukikaze:~# realm join sotoba.de  
Password for Administrator:  
See: journalctl REALMD_OPERATION=r49063.2260  
realm: Couldn't join realm: Necessary packages are not installed: sssd-tools sssd libnss-sss libpam-sss adcli  
root@Yukikaze:~# ^C  
root@Yukikaze:~#
```

Leider gibt er da eine Fehlermeldung also schaue ich ob der Samba-Dienst läuft

systemctl status sssd

```
root@Yukikaze:~# systemctl status sssd
* sssd.service - System Security Services Daemon
   Loaded: loaded (/lib/systemd/system/sss.service; enable>
   Active: inactive (dead)
   Condition: start condition failed at Mon 2025-05-26 18:53:5>
               |- ConditionPathExists=/etc/sss/sss.conf was >
               `-- ConditionDirectoryNotEmpty=/etc/sss/conf.d >

May 26 18:53:57 Yukikaze systemd[1]: Condition check resulted>
May 26 18:53:57 Yukikaze systemd[1]: Condition check resulted>
May 26 18:53:57 Yukikaze systemd[1]: Condition check resulted>
May 26 18:53:57 Yukikaze systemd[1]: Condition check resulted>
May 26 18:53:57 Yukikaze systemd[1]: Condition check resulted>
May 26 18:53:59 Yukikaze systemd[1]: Condition check resulted>
lines 1-13/13 (END)
```

Da sagt er mir das er Probleme hat die sssd.conf zu finden. Also schaue ich nach, ob sie existiert,

ls -l /etc/sss/sss.conf

```
root@Yukikaze:~# ls -l /etc/sss/sss.conf
ls: cannot access '/etc/sss/sss.conf': No such file or direc
tory
root@Yukikaze:~#
```

Da sie nicht existiert erstelle ich mir eine eigene sssd.conf mit folgendem Inhalt

nano /etc/sss/sss.conf

```
[sss]
```

```
services = nss, pam, ssh
```

```
config_file_version = 2
```

```
domains = sotoba.de
```

```
[domain/sotoba.de]
```

```
id_provider = ad
```

```
access_provider = ad
```

```
ad_domain = sotoba.de
```

```
krb5_realm = SOTObA.DE
```

```
realmd_tags = manages-system joined-with-adcli
```

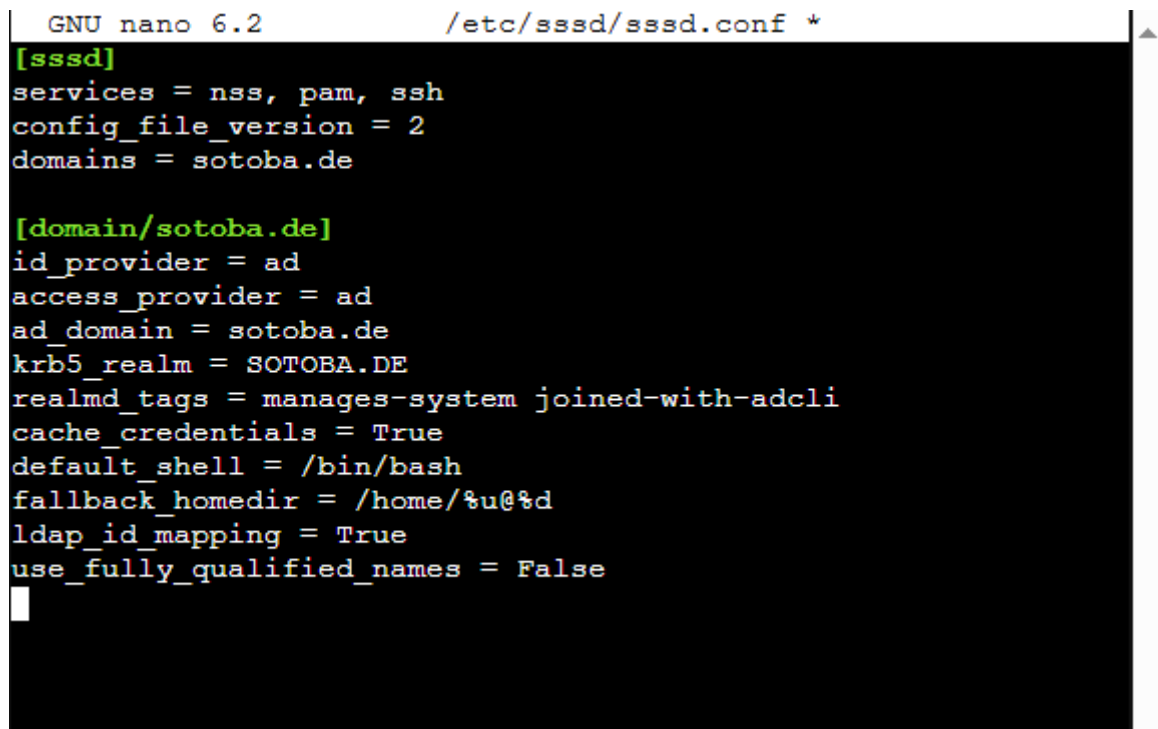
```
cache_credentials = True
```

```
default_shell = /bin/bash
```

```
fallback_homedir = /home/%u@%d
```

```
ldap_id_mapping = True
```

```
use_fully_qualified_names = False
```



```
GNU nano 6.2 /etc/sss/sss.conf *
[sss]
services = nss, pam, ssh
config_file_version = 2
domains = sotoba.de

[domain/sotoba.de]
id_provider = ad
access_provider = ad
ad_domain = sotoba.de
krb5_realm = SOTObA.DE
realmd_tags = manages-system joined-with-adcli
cache_credentials = True
default_shell = /bin/bash
fallback_homedir = /home/%u@%d
ldap_id_mapping = True
use_fully_qualified_names = False
```

Nach dem ich diese Datei gespeichert habe, ändere ich noch die Berechtigungen und schaue ob die Berechtigungen übernommen hat

```
chmod 600 /etc/sss/sss.conf
chown root:root /etc/sss/sss.conf
ls -l /etc/sss/sss.conf
```

```
root@Yukikaze:~# chmod 600 /etc/sss/sss.conf
chown root:root /etc/sss/sss.conf
root@Yukikaze:~# ls -l /etc/sss/sss.conf
-rw----- 1 root root 365 May 26 19:06 /etc/sss/sss.conf
root@Yukikaze:~#
```

Die Datei hat also erweiterte Berechtigungen

Besitzer: root

Gruppe: root

Modus: 600 (also nur lesen und schreiben für root)

Nun starte ich den Dienst neu und lasse mir seinen Status ausgeben

```
systemctl start sssd
```

```
systemctl status sssd
```

```
root@Yukikaze:~# systemctl start sssd
systemctl status sssd
Job for sssd.service failed because the control process exited
with error code.
See "systemctl status sssd.service" and "journalctl -xeu sssd.
service" for details.
* sssd.service - System Security Services Daemon
   Loaded: loaded (/lib/systemd/system/sss.service; enable>
   Active: activating (auto-restart) (Result: exit-code) si>
   Process: 2277 ExecStart=/usr/sbin/sss -i ${DEBUG_LOGGER}>
   Main PID: 2277 (code=exited, status=1/FAILURE)
   CPU: 907ms
lines 1-6/6 (END)
^C
```

```
root@Yukikaze:~# systemctl status sssd
* sssd.service - System Security Services Daemon
   Loaded: loaded (/lib/systemd/system/sss.service; enable>
   Active: activating (start) since Mon 2025-05-26 19:12:40>
   Main PID: 2326 (sss)
   Tasks: 1 (limit: 9329)
   Memory: 4.6M
   CPU: 345ms
   CGroup: /system.slice/sss.service
           └─2326 /usr/sbin/sss -i --logger=files

May 26 19:12:40 Yukikaze sssd_be[2328]: Failed to read keytab>
May 26 19:12:42 Yukikaze sssd_be[2329]: Starting up
May 26 19:12:42 Yukikaze sssd_be[2329]: krb5_kt_start_seq_get>
May 26 19:12:42 Yukikaze sssd_be[2329]: krb5_kt_start_seq_get>
May 26 19:12:42 Yukikaze sssd_be[2329]: krb5_kt_start_seq_get>
May 26 19:12:42 Yukikaze sssd_be[2329]: krb5_kt_start_seq_get>
May 26 19:12:42 Yukikaze sssd_be[2329]: krb5_kt_start_seq_get>
```

Der Fehler „Failed to read keytab“ das sssd Probleme mit Kerberos hat. Also schaue ich nach, ob die keytab-Datei existiert

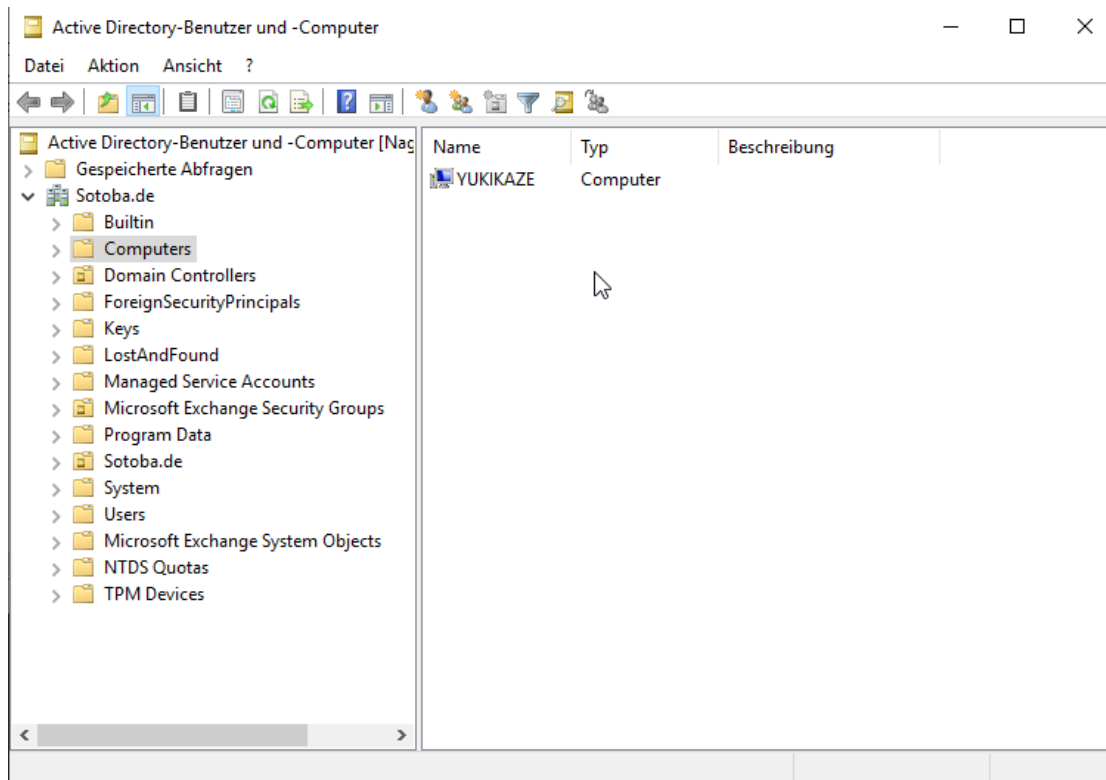
ls -l /etc/krb5.keytab

```
root@Yukikaze:~# ls -l /etc/krb5.keytab
ls: cannot access '/etc/krb5.keytab': No such file or director
y
root@Yukikaze:~#
```

Die fehlt also. Realm join geht also nicht. Da mache ich halt nen  
adcli join sotoba.de -U Administrator

```
root@Yukikaze:~# realm join sotoba.de
realm: Already joined to this domain
root@Yukikaze:~# adcli join sotoba.de -U Administrator
Password for Administrator@SOToba.DE:
root@Yukikaze:~#
```

Und schaue, ob der Container im AD angekommen ist



## Docker Installieren

Da ich für opsi einen Docker Container nutzen möchte, installiere ich zuerst Docker. Dazu muss ich zuerst Docker dem apt hinzufügen.

```
apt-get update
apt-get install ca-certificates curl
install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

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" | \
tee /etc/apt/sources.list.d/docker.list > /dev/null
apt-get update
```

```
root@Yukikaze:~# apt-get update
apt-get install ca-certificates curl
install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
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" | \
tee /etc/apt/sources.list.d/docker.list > /dev/null
apt-get update
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203-22.04.1).
curl is already the newest version (7.81.0-1ubuntu1.20).
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://archive.ubuntu.com/ubuntu jammy-security InRelease
Get:4 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Get:5 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [48.8 kB]
Fetched 97.6 kB in 1s (191 kB/s)
Reading package lists... Done
root@Yukikaze:~#
```

## Docker selbst installieren

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

```
root@Yukikaze:~# 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:
  dbus-user-session docker-ce-rootless-extras git git-man libcurl3-gnutls liberror-perl libgdbm-compat4 libltdl7 libperl5.34 libslirp0 patch perl
  perl-modules-5.34 pigz slirp4netns
Suggested packages:
  cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn diffutils-doc
  perl-doc libterm-readline-gnu-perl | libterm-readline-perl-perl make libtap-harness-archive-perl
The following NEW packages will be installed:
  containerd.io dbus-user-session docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin git git-man libcurl3-gnutls
  liberror-perl libgdbm-compat4 libltdl7 libperl5.34 libslirp0 patch perl perl-modules-5.34 pigz slirp4netns
0 upgraded, 20 newly installed, 0 to remove and 5 not upgraded.
Need to get 133 MB of archives.
After this operation, 510 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```



Nun teste ich, ob Docker einwandfrei läuft, dazu versuche ich Alpine zu starten

`docker run --rm -it alpine`

```
root@Yukikaze:~# docker run --rm -it alpine
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
f18232174bc9: Pull complete
Digest: sha256:a8560b36e8b8210634f77d9f7f9efd7ffa463e380b75e2e74aff4511df3ef88c
Status: Downloaded newer image for alpine:latest
docker: Error response from daemon: AppArmor enabled on system but the docker-default profile could not be loaded: running '/usr/sbin/apparmor_parser -Xr /var/lib/docker/tmp/docker-default269373107' failed with output: apparmor_parser: Unable to replace "docker-default". Permission denied; attempted to load a profile while confined?
error: exit status 243

Run 'docker run --help' for more information
root@Yukikaze:~#
```

Damit ich Alpine erfolgreich installieren kann, muss ich noch die Konfigurationsdatei des Containers ändern. Also füge ich der folgenden Datei auf dem Host folgendes hinzu und starte den Container neu

`/etc/pve/lxc/<ID>.conf`

`lxc.cgroup.devices.allow: a`

`lxc.cap.drop:`

`lxc.apparmor.profile: unconfined`

`features: nesting=1,keyctl=1`

```
GNU nano 7.2 /etc/pve/lxc/101.conf
arch: amd64
cores: 2
features: nesting=1,keyctl=1
hostname: Yukikaze
memory: 2048
nameserver: 192.168.2.7
net0: name=eth0,bridge=vbr0,firewall=1,hwaddr=BC:24:11:F2:A4:7D,ip=dhcp,ip6=dhcp,type=veth
ostype: ubuntu
rootfs: local:101/vm-101-disk-1.raw,size=30G
searchdomain: sotoba.de
swap: 2048
lxc.cgroup.devices.allow: a
lxc.cap.drop:
lxc.apparmor.profile: unconfined
```

Nach dem Neustart des Containers kann ich nun Alpine ausführen

```
root@Yukikaze:~# docker run --rm -it alpine
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
f18232174bc9: Pull complete
Digest: sha256:a8560b36e8b8210634f77d9f7f9efd7ffa463e380b75e2e74aff4511df3ef88c
Status: Downloaded newer image for alpine:latest
/ #
```

## Opsi Installieren

Danach beginne ich OPSI zu installieren, zu lege ich zuerst einen Ordner an, den ich Opsi\_server

mkdir opsi-server

```
root@Yukikaze:~# mkdir opsi-server
root@Yukikaze:~# ls
opsi-server
root@Yukikaze:~# ^C
root@Yukikaze:~#
```

Danach lade ich mit dem Befehl wget folgendes Skript runter

wget -O script.sh <https://raw.githubusercontent.com/opsi-org/opsi-docker/main/opsi-server/opsi-server.sh>

```
root@Yukikaze:~# mkdir opsi-server
root@Yukikaze:~# ls
opsi-server
root@Yukikaze:~# ^C
root@Yukikaze:~# wget -O script.sh https://raw.githubusercontent.com/opsi-org/opsi-docker/main/opsi-server/opsi-server.sh
--2025-05-26 15:22:55-- https://raw.githubusercontent.com/opsi-org/opsi-docker/main/opsi-server/opsi-server.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.110.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8805 (8.6K) [text/plain]
Saving to: 'script.sh'

script.sh                               100%[=====>]  8.60K  --.-KB/s  in 0.002s

2025-05-26 15:22:55 (4.68 MB/s) - 'script.sh' saved [8805/8805]

root@Yukikaze:~# ls
opsi-server  script.sh
root@Yukikaze:~#
```

Danach mache ich das Skript, das ich gerade runtergeladen habe, ausführbar mache.

chmod +x script.sh

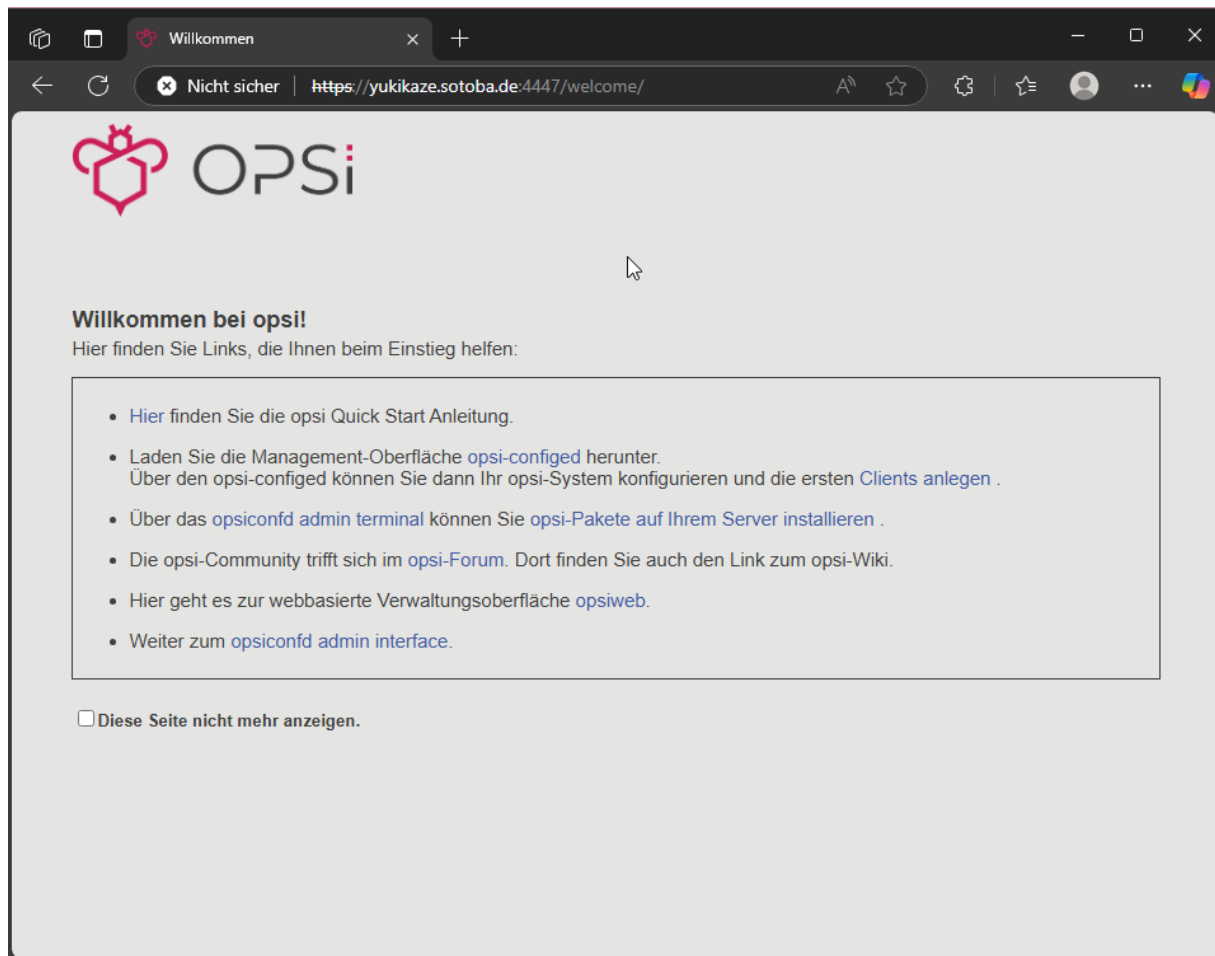
```
root@Yukikaze:~/opsi-server# chmod +x script.sh
root@Yukikaze:~/opsi-server#
```

Danach führe ich mit ./script.sh start das Skript aus.

```
root@Yukikaze:~/opsi-server# ./script.sh start
Download docker-compose.yml
2025-05-27 07:26:21 URL:https://raw.githubusercontent.com/opsi-org/opsi-docker/main/opsi-server/docker-compose.yml [2816/2816] -> "docker-compose.yml" [1]
Start containers
docker compose up -d
WARN[0000] /root/opsi-server/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 3/38
[+] Running 38/38      Pulling                               38.3s
✓ grafana Pulled      430.3s
✓ redis Pulled        374.0s
✓ opsi-server Pulled  570.1s
✓ mysql Pulled        197.1s

[+] Running 9/9
✓ Network opsi-server_default      Created      0.1s
✓ Volume "opsi-server_mysql_data"  Created      0.0s
✓ Volume "opsi-server_redis_data"  Created      0.0s
✓ Volume "opsi-server_grafana_data" Created      0.0s
✓ Volume "opsi-server_opai_data"   Created      0.0s
✓ Container opsi-server-grafana-1  Started      1.1s
✓ Container opsi-server-mysql-1    Started      1.2s
✓ Container opsi-server-redis-1    Started      1.1s
✓ Container opsi-server-opai-server-1 Started      1.7s
root@Yukikaze:~/opsi-server# cd opsi-server
-bash: cd: opsi-server: No such file or directory
root@Yukikaze:~/opsi-server# cd opsi-server#
```

Nach der erfolgreichen Installation kann ich über die URL <https://yukikaze.sotoba.de:4447> auf die Verwaltungsoberfläche zugreifen



Und in den Administrationsbereich

