

Setup

- Setup Environment like server-side and Prepare ZFS Pool for Prefill
 - Setup Proxmox
 - Setup LXC
- Alternative Setups
 - Alternative: Setup Pool and Service on plain Debian or Ubuntu
 - Alternative: Send Empty HDD, We Create The ZFS Pool For You
- Create ZFS Pool
- Install Minio S3 Storage Server on Debian/Ubuntu
- Install Syncthing
- Finishing Steps in LXC
- Troubleshoot

Setup Environment like
server-side and Prepare ZFS
Pool for Prefill

Setup Environment like server-side and Prepare ZFS Pool for Prefill

Setup Proxmox

Proxmox Documentation

<https://www.proxmox.com/en/proxmox-virtual-environment/get-started>

<https://pve.proxmox.com/pve-docs/chapter-pve-installation.html>

Enable No-Subscription Repository in web-GUI

node on the left -> Updates -> Repositories.

Disable the enterprise repository and add the No-Subscription repository

Go to the upper menu Update, press Refresh, then Upgrade

Untested Community Script

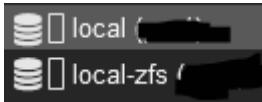
This script is quite popular in the homelab-community, but I never tried it. Look at the source code, if in doubt. No warranties at all.

<https://community-scripts.github.io/ProxmoxVE/scripts?id=post-pve-install>

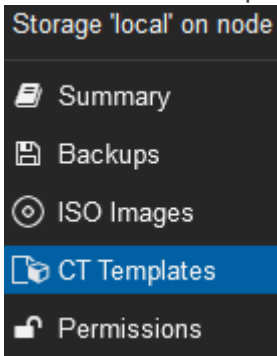
Setup Environment like server-side and Prepare ZFS Pool for Prefill


Setup LXC

1. Go to "local" in webgui



2. Click on "CT Templates"



3. Then click button "Templates"
 4. Download Debian 12
 5. Click "Create CT" on top right
- 
- A screenshot of a blue button with a white cube icon and the text 'Create CT'.
6. Give it Hostname and Password, Make sure Unprivileged and Nesting is checked
 7. Click next, choose the previous downloaded Template
 8. Click next, default size 8GB is sufficient
 9. Click next, assign 2 Cores
 10. Click next, type 2048MiB Memory
 11. Click next, choose IP according your network (or DHCP)
 12. Click through finish

Alternative Setups

Alternative: Setup Pool and Service on plain Debian or Ubuntu

First install Debian or Ubuntu

<https://www.debian.org/CD/netinst/index.en.html>

<https://ubuntu.com/tutorials/install-ubuntu-desktop#1-overview>

Install ZFS

<https://wiki.debian.org/ZFS#Installation>

Only user and group ids are different when not using LXC. Simply change them after prefill. Otherwise it will be changed by us and delay deployment (no extra fee).

Proceed with [Create ZFS Pool](#) and following chapters.

Alternative: Send Empty HDD, We Create The ZFS Pool For You

If you don't want to prefill, just send the empty HDD. We will create the ZFS Pool and services according your needs. No extra handling fee. But be aware, traffic shaping can occure after a certain amount of traffic, plase check [GTC](#)

Create ZFS Pool

Links

https://pve.proxmox.com/wiki/ZFS_on_Linux

Introduction

ZFS pool can be built with `WWN` or `partuuid`, to be able to run the pool even from an USB case if needed. For normal service, `disk ID` is sufficient.

Poolname should be customerid, to make deployment easier. Poolname can be changed

Create a ZFS pool with 2 HDD as mirror

Get WWN from HDD

```
ls -l /dev/disk/by-id/  
#richtige HDD suchen, zB ata-WDC_WD40EFRX-68WT0N0_WD-WCC4E4DF9NJD -> ../../sdj  
#daher ist unsere gesuchte HDD sdj  
#WWN suchen, welche auf sdj verweist, hier:  
#wwn-0x50014ee20c6324e6 -> ../../sdj  
#zweite HDD suchen, zB ata-WDC_WD40EFRX-68WT0N0_WD-WCC4E6FN513C -> ../../sdk  
#daher ist unsere gesuchte HDD sdk  
#WWN suchen, welche auf sdk verweist, hier:  
#wwn-0x50014ee20c629629 -> ../../sdk
```

Create ZFS Pool with WWN from above


```
zpool create -o ashift=12 -O compression=zstd poolname mirror /dev/disk/by-id/wwn-0x50014ee20c6324e6  
/dev/disk/by-id/wwn-0x50014ee20c629629
```

Create ZFS Pool in Raidz1

Get WWN from HDD same as above for each HDD

Create ZFS Pool with WWN from above

```
zpool create -o ashift=12 -O compression=zstd poolname raidz1 /dev/disk/by-id/wwn-<1> /dev/disk/by-id/wwn-  
<2> /dev/disk/by-id/wwn-<3>
```

Create a ZFS pool on a single disk with no data redundancy

Find WWN of the HDD

```
ls -l /dev/disk/by-id/  
#find the right HDD, for this example: ata-WDC_WD40EFRX-68WT0N0_WD-WCC4E4DF9NJD -> ../../sdj  
#the letter of our example HDD is currently sdj  
#find WWN of sdj  
#wwn-0x50014ee20c6324e6 -> ../../sdj
```

Create ZFS Pool with WWN from above

```
zpool create -o ashift=12 -O compression=zstd poolname /dev/disk/by-id/wwn-0x50014ee20c6324e6
```

Create a ZFS pool on a single disk with 5 partitions for data redundancy

Attention: Very slow performance

```
apt install parted
parted /dev/disk/by-id/ata-WDC_WD40EFRX-68WT0N0_WD-WCC4E4DF9NJD mklabel gpt
parted /dev/disk/by-id/ata-WDC_WD40EFRX-68WT0N0_WD-WCC4E4DF9NJD mkpart zfs 0% 20%
parted /dev/disk/by-id/ata-WDC_WD40EFRX-68WT0N0_WD-WCC4E4DF9NJD mkpart zfs 20% 40%
parted /dev/disk/by-id/ata-WDC_WD40EFRX-68WT0N0_WD-WCC4E4DF9NJD mkpart zfs 40% 60%
parted /dev/disk/by-id/ata-WDC_WD40EFRX-68WT0N0_WD-WCC4E4DF9NJD mkpart zfs 60% 80%
parted /dev/disk/by-id/ata-WDC_WD40EFRX-68WT0N0_WD-WCC4E4DF9NJD mkpart zfs 80% 100%
```

Find the letter of the disk

```
ls -l /dev/disk/by-id/ata-WDC_WD40EFRX-68WT0N0_WD-WCC4E4DF9NJD
```

Result

```
/dev/disk/by-id/ata-WDC_WD40EFRX-68WT0N0_WD-WCC4E4DF9NJD -> ../../sdh
```

Find all Partuuid of sdh

```
ls -l /dev/disk/by-partuuid/ | grep sdh
```

Result (just one displayed as example)

```
2c49c49f-4221-324e-afca-23bedbb06677 -> ../../sdh1 #2c49c49f-4221-324e-afca-23bedbb06677 is the
partuuid1
```

Create ZFS pool (adjust ashift if needed)

```
zpool create -o ashift=12 -O compression=zstd poolname raidz1 /dev/disk/by-partuuid/<partuuid1>
/dev/disk/by-partuuid/<par
```

Install Minio S3 Storage Server on Debian/Ubuntu

Introduction

Install

Download DEB (Debian/Ubuntu) setup file and install as shown in minio documentation:

<https://min.io/docs/minio/linux/operations/install-deploy-manage/deploy-minio-single-node-single-drive.html#download-the-minio-server>

Install Syncthing

Multiple instances are possible, but for user- and hostnames only “syncthing”, “syncthing2” to “syncthing9” are allowed. Further adoptions needed

- Username
- Datasetname and ownage (chown)
- .service file
 - User=
 - -home
 - -gui-address increase port by one

Install syncthing

```
apt install curl apt-transport-https ca-certificates
curl -o /usr/share/keyrings/syncthing-archive-keyring.gpg https://syncthing.net/release-key.gpg
echo "deb [signed-by=/usr/share/keyrings/syncthing-archive-keyring.gpg] https://apt.syncthing.net/ syncthing
stable" | tee /etc/apt/sources.list.d/syncthing.list
apt update
apt install syncthing
```

Create and adopt this file for every instance you want `/etc/systemd/system/syncthing.service`

```
[Unit]
Description=Syncthing - Open Source Continuous File Synchronization for syncthing
Documentation=man:syncthing(1)
After=network.target
StartLimitIntervalSec=60
StartLimitBurst=4

[Service]
User=syncthing
ExecStart=/usr/bin/syncthing serve --no-browser --no-restart --logflags=0 --home=/poolname/syncthing --gui-
address=0.0.0.0:8384
Restart=on-failure
RestartSec=1
SuccessExitStatus=3 4
RestartForceExitStatus=3 4
```

```
# Hardening
ProtectSystem=full
PrivateTmp=true
SystemCallArchitectures=native
MemoryDenyWriteExecute=true
NoNewPrivileges=true

# Elevated permissions to sync ownership (disabled by default),
# see https://docs.syncthing.net/advanced/folder-sync-ownership
#AmbientCapabilities=CAP_CHOWN CAP_FOWNER

[Install]
WantedBy=multi-user.target
```

Copy `/etc/systemd/system/syncthing` to the ZFS pool:

```
mkdir /customerid/lxcbackups/systemd
cp -a /etc/systemd/system/syncthing /customerid/lxcbackups/systemd/
```

```
systemctl daemon-reload
systemctl start syncthing.service
systemctl status syncthing.service
```

Finishing Steps in LXC

Only applies when using LXC on Proxmox.

Make ZFS Pool available in LXC

On Host

```
nano /etc/pve/lxc/100.conf  
# add this line, edit poolname if necessary:  
mp0: mp=/poolname,/poolname  
# mp=/poolname is from host  
# ./poolname is the mountpoint in LXC
```

Troubleshoot

Change Pool to use WWN

```
zpool export poolname ; sleep 5 ; zpool import -d /dev/disk/by-id poolname ; sleep 5 ; zpool list -v poolname
```

Change Poolname

```
zpool export poolname  
zpool import poolname newpoolname
```