

napp-it cs  
Client Server Edition

Storageserver workgroup  
First Steps

published: 2026-Apr-17 (c) napp-it.org

Licence:  
CC-BY-SA see <http://creativecommons.org/licenses/by-sa/2.0/>

## First Steps

1. What is napp-it cs ?
  - 1.1 Concept
2. Hardware and Software options
  - 2.1 Single disk minimal system
  - 2.2 Bootdisk + dedicated data disk(s)
  - 2.3 Bootdisk + dedicated data disks (hybrid pool)
  - 2.4 RAM
  - 2.5 OS options for fileserver
  - 2.6 Options for AiO (fileserver + hypervisor)
3. Setup and first steps
  - 3.1 Download napp-it cs application folder
  - 3.2 napp-it cs web-gui (any OS)
  - 3.3 napp-it cs servergroup
  - 3.4 Create Storage Pools (Storage Spaces and ZFS)
  - 3.5 Extend a Storage Pool
  - 3.6 Repair a Storage Pool
  - 3.7 Share data over SMB
  - 3.8 Protect data with ACL
  - 3.9 Posix ACL settings (Linux)
4. Special functions
  - 4.1 napp-it cs Autoscrub
  - 4.2 napp-it cs Autosnap
  - 4.3 napp-it cs Replication
  - 4.4 napp-it cs Encryption
  - 4.5 Data versioning and disaster backup

other manuals

[https://www.napp-it.org/doc/downloads/napp-it\\_cs.pdf](https://www.napp-it.org/doc/downloads/napp-it_cs.pdf)

<https://www.napp-it.org/doc/downloads/freebsd-aio.pdf>

<https://www.napp-it.org/doc/downloads/osx-aio.pdf>

<https://www.napp-it.org/doc/downloads/proxmox-aio.pdf>

<https://www.napp-it.org/doc/downloads/windows-aio.pdf>

<https://www.napp-it.org/doc/downloads/windows-rdma.pdf>

# 1. What is „napp-it cs“ ?

Napp-it cs (client server) is a webbased (storage) server management tool. It is based on napp-it se (Solaris edition), available since 2009. Napp-it cs has three parts

- **Web-gui frontend** (Apache webserver + Perl web-gui scripts + user editable menus)
- **Server backend services** (low ram services, ca 60KB RAM for remote control, monitoring and job management)
- **Keyserver backend** for public https server (optional cgi script)

NAME	STATE	READ	WRITE	CKSUM	TRAN	MODEL	SIZE	SN	HEALTH	DEV/ID
winpool	ONLINE	0	0	0						
physicaldrive6	ONLINE	0	0	0	iSCSI	COMSTAR	10.7 GB		Healthy	
mirror-1	ONLINE	0	0	0						
physicaldrive1	ONLINE	0	0	0	iSCSI	COMSTAR	2.1 GB		Healthy	
physicaldrive2	ONLINE	0	0	0	iSCSI	COMSTAR	2.1 GB		Healthy	
physicaldrive7	ONLINE	0	0	0	iSCSI	COMSTAR	2.1 GB		Healthy	

napp-it cs frontend on Windows with a Storage Spaces and ZFS pool and a servergroup of differen hosts,

## 1.1 Concept

Unlike many other web-gui tools, the napp-it cs frontend and backend can run on any OS (Free-BSD, Illumos, Linux, OSX and Windows) where you only need a Apache webserver and Perl. Main platform is Illumos (OmniOS), Linux (Proxmox) and Windows (Windows 11 or Windows Server). One frontend setup can manage multiple hosts (locally and remote) with a select option for localhost and remote hosts. Any menu option affects then the selected host.

Napp-it cs is Copy and Run. This means that installation = download csweb-gui app folder and deinstallation=delete csweb-gui app folder, Update and downgrade (last 5 versions) can be done online (menu About > Update).

Napp-it cs can manage any service and server. You can extend napp-it cs by private menus (update save). Default menus are intended to manage storage, either Storage Spaces on Windows or ZFS (any OS including Windows).

The frontend server with an Apache webserver can be any average desktop or server with 4GB Ram or more. Use whatever is ok for your OS. For any GUI based OS (OSX, Windows etc) 8-16GB is suggested.

The background services server (remote control), monitor (caching) and auto.pl (jobs) are ultra resource efficient. They need less than 80KB on a frontend server plus the RAM you need for GUI and to process storage tools like zfs or zpool plus the RAM you want for storage performance (read/write caching)

Backend services, RAM need

```
PS C:\Users\me> tasklist | findstr _cs
csperl_server.exe          26176 Console          1      31,048 K
csperl_monitor.exe        25524 Console          1      12,252 K
csperl_auto.exe           2744 Console           1      29,448 K

root@omnios:~# ps -a -o pid,vsz,args,rss | grep cs_server | grep -v grep
15890 22924 perl /var/csweb-gui/data/cs_server/server.pl 18732
15892 9468 perl /var/csweb-gui/data/cs_server/monitor.pl 5352
15894 23440 perl /var/csweb-gui/data/cs_server/auto.pl 19392
root@omnios:~#
```

Backend services server, monitor and auto, example on Windows and Unix (frontend)

Backend services run well on ARM or X86 ultra tiny devices like a low RAM Raspberry (backend only server around 30KB RAM).

## 2. Hardware and Software options

### 2.1 Single disk minimal system

with OS and data on a single bootdisk system  
example Intel Nuc or old laptop

#### pro

- very cost and energy efficient

#### contra

- during a possible OS reinstall, you may delete all data unintentionally
- no redundancy, backup is mandatory

This is a possible but not recommended setup  
For low capacity need, use 1x NVMe otherwise 1x hd.

### 2.2 Bootdisk + dedicated data disk(s)

with OS on first disk (32GB min) and data on other disk( s)with or without redundancy  
example 2-4 bay NAS or mini server

#### pro

- easy to setup and maintain
- cost and energy efficient
- easy to move data pools

#### contra

- limited expansion options

### 2.3 Bootdisk + dedicated data disks in a Hybrid setup (hd + NVMe)

with OS on first disk and data on other disk( s), best with redundancy. Data can be on hd or flash with a tiering option to move  
example iTX, uATX or ATX server, M.2 bootdisk + hd + NVMe (M.2/U.2/U.3 or PCIe adapter) or tiny system

#### pro

- easy to setup and maintain
- with recsize vs small blocksize you can define per dataset where data (metadata, small files or all files) is stored
- data move hd vs flash with zfe rewrite (OpenZFS)-  
for high capacity + high performance and medium costs

### 2.4 RAM

Ultra compact systems like a 2GB Raspberry are possible when performance is not so important.

In general, you need a 64bit OS that wants 1-2 GB RAM. With modern Copy on Write filesystems on hd (btrfs, ReFS, ZFS) you need RAM for caching or performance sucks. For low RAM systems prefer SSD/NVMe disks.

If you want a ZFS storage system with good performance

- consider 4-8GB or more for an Illumos/OmniOS NAS and ZFS in its native Unix environment  
This is the most resource efficient and stable ZFS option.
- consider 8-12 GB or more for OpenZFS or Windows Storage Spaces
- consider 8-12GB or more If you want a local desktop or web-gui.
- If you want to local add services or container virtualisation, add 2-4GB RAM
- If you want full OS virtualisation, add 4-16GB RAM per VM
- Add RAM for performance (rambased read/write caching)

## 2.5 OS options for fileserver

### Free-BSD + OpenZFS + SAMBA

- very stable storage systems with ZFS
- advanced NFS4 ACL for permissions (Windows ntfs alike)

### Illumos + Illumos ZFS + ZFS based /inkernel SMB (OpenIndiana, OmniOS)

- the most resource efficient and stable ZFS
- compatible to OpenZFS
- NFS4 ACL with Windows SID as ZFS attribute, SMB groups in groups
- remote permission management via Windows desktop (property > security) based on local users
- ZFS snapshots = Windows previous versions with zero setup

### OSX + APFS/OpenZFS + SAMBA

- OpenZFS released
- if you know or have it, best for desktop use, not so perfect for a server

### Linux + OpenZFS + SAMBA

- best mainstream option but lacks modern NFS4 ACL (only simple Posix ACL)
- best All in One Linux (barebone storage + hypervisor) is Proxmox that comes with ZFS

### Solaris + native Solaris ZFS + ZFS based /inkernel SMB

- only for large enterprises if you need best of all stability and commercial support
- incompatible zu OpenZFS

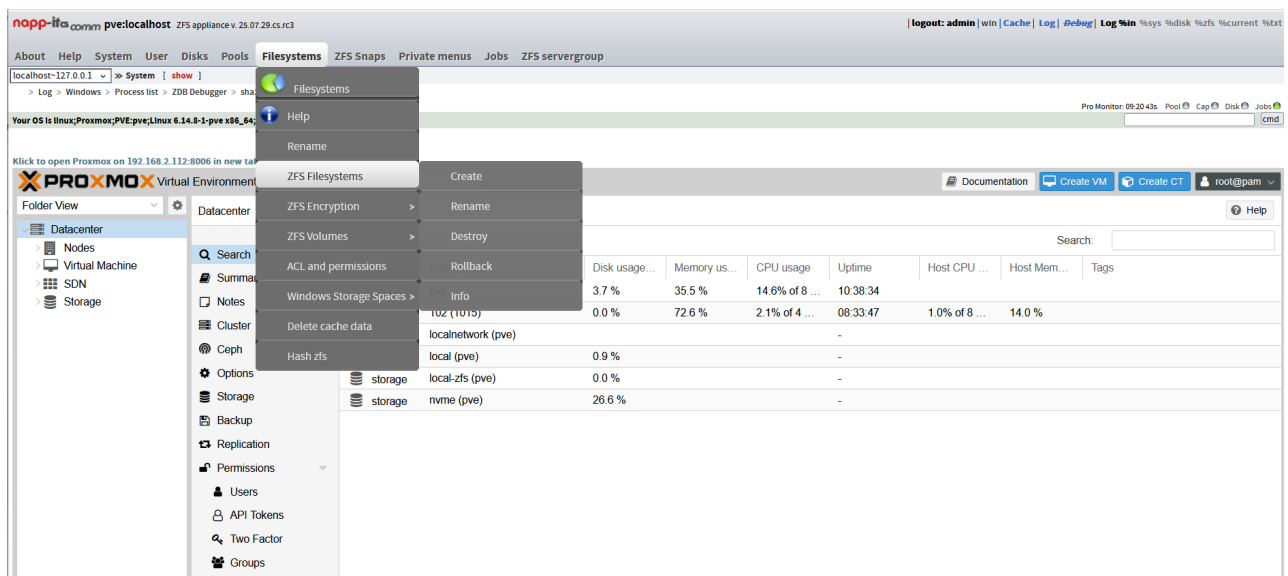
### Windows + ntfs/ReFS/OpenZFS + native kernel Windows SMB

- origin of NFS4 ACL with worldwide unique Windows SID as ACL attribute instead simple uid numbers, SMB groups in groups
- remote permission management via Windows desktop (property > security) based on local users
- best of all SMB performance with SMB Direct (Windows Server or cheap Server Essentials requires for SMB Direct)
- auditing and network Raid over Lan with fast virtual disks .vhdx over SMB
- OpenZFS is release candidate (beta)

## 2.6 OS options for All in One (fileserver + hypervisor)

while any OS can virtualize, there are some with premium features

- ProxMox, this is Debian optimized for hypervisor use and web-gui for VM management
- ZFS and add ons like Proxmox backup server included, for advanced storage management combine with napp-it cs



Proxmox and napp-it cs web-gui combined

- Windows + HyperV

if you know it or need Windows or want Raid over Lan with fast virtual disks .vhdx over SMB

### 3. Setup and first steps

after installing a base OS like Free-BSD, Linux, OmniOS, OSX or Windows

#### 3.1 Download napp-it cs application folder csweb-gui (all OS beside Windows)

[https://www.napp-it.org/doc/downloads/napp-it\\_cs.zip](https://www.napp-it.org/doc/downloads/napp-it_cs.zip)

- upload csweb-gui to /var (WinSCP)

- start napp-it via Putty ex Proxmox: `sh /var/csweb-gui/startup/backend+web-gui/proxmox/apache/apache_proxmox.sh`

on Windows you need an additiona Apache/Perl stack like Xampp, download (and unzip to c:\xampp)

<https://www.napp-it.org/doc/downloads/xampp.zip> (Xampp with napp-it cs)

- start napp-it cs (C:\xampp\\_\_start\_zfs-gui\_as\_admin.bat)

If you want OpenZFS on Windows,

- install newest OpenZFS filesystem driver: <https://github.com/openszsonwindows/openszfs/releases>

- read issue tracker: <https://github.com/openszsonwindows/openszfs/issues>

Manuals per OS:

[https://www.napp-it.org/doc/downloads/napp-it\\_cs.pdf](https://www.napp-it.org/doc/downloads/napp-it_cs.pdf)

<https://www.napp-it.org/doc/downloads/freebsd-aio.pdf>

<https://www.napp-it.org/doc/downloads/osx-aio.pdf>

<https://www.napp-it.org/doc/downloads/proxmox-aio.pdf>

<https://www.napp-it.org/doc/downloads/windows-aio.pdf>

#### 3.2 napp-it cs web-gui (any OS)

Open browser at adress <https://localhost> (local call) or <https://ip> (remote call)

The screenshot shows the napp-it cs web-gui interface. At the top, it displays the user 'admin' and the system version '26.04.16'. The main content area is titled 'Pools on localhost-127.0.0.1:MSWin32;Windows;WIN:my-w11;Microsoft Windows [Version 10.0.26200.8246];zfs2.41552921;cs; 26.04.16'. Below this, there is a table for 'ZFS Pools' and a section for 'Windows Storage Pools'.

Pool (properties)	VER	HEALTH	SIZE	CAP	AVAIL	RES	FRES	FAILM	EXP	REPL	ALT	GUID	HEALTH	Autotrim	SYNC	ENCRYPT	ACTION	ATIME	PriCache	SecCache	ACLinherit	ACLmode	CP	Rdonly
winpool	upgrade	ONLINE	13.2G	0	9.19G	none	942M	wait	off	off	-	6259204261886831355	ONLINE	off	standard	off	clear errors	on	all	all	passthrough	passthrough	-	-

Info: RAW poolsize does not count redundancy, usable/available size is from zfs list, df -h displays size as a power of 1024 whereas df -H displays as a power of 1000

Pool	Operational	Health	Primordial	ReadOnly	Size	Alloc	Disks
Storage pool	OK	Healthy	False	False	30.3 GB	3.2 GiB	5,3,4
Primordial	OK	Healthy	True	False	561.5 GB	29.7 GiB	1,5,0,3,7,6,4,2

pool: winpool  
state: ONLINE  
scan: resilvered 360K in 00:00:08 with 0 errors on Tue Dec 2 11:31:35 2025  
config:

NAME	STATE	READ	WRITE	CKSUM	TRAN	MODEL	SIZE	SN	HEALTH	DEV/ID
winpool	ONLINE	0	0	0						
physicaldrive6	ONLINE	0	0	0	1SCS	CONSTAR	10.7 GB		Healthy	
special	AVAIL [4.2 GB	used:3.19%	small block: 128K]							
mirror-1	ONLINE	0	0	0						
physicaldrive1	ONLINE	0	0	0	1SCS	CONSTAR	2.1 GB		Healthy	
physicaldrive2	ONLINE	0	0	0	1SCS	CONSTAR	2.1 GB		Healthy	
physicaldrive7	ONLINE	0	0	0	1SCS	CONSTAR	2.1 GB		Healthy	

errors: No known data errors  
single disk vdev: diskname

*napp-it cs web-gui (example on Windows with Storage Spaces and ZFS Pool)*

Prior further settings, check for bug and security updates

- is OS up to date, optionally update OS

- is napp-it cs up to date, optionally update with menu About > Update

- is ZFS up to date, optionally update (mainly Windows and OSX otherwise distribution maintains ZFS)

### 3.3 napp-it cs servergroup

If you want to manage not only a local installation but other servers remotely:  
Open menu „ZFS Servergroup“ and click on „++ add ZFS server“ below serverlist  
Enter ip, name and authvalue (for added server from /var/csweb-gui/cfg/server.auth)

You can then select a server from the group and all menu actions affect the selected server

The screenshot shows the napp-it web interface for managing a ZFS servergroup. The top navigation bar includes 'About', 'Help', 'System', 'User', 'Disks', 'Pools', 'Filesystems', 'ZFS Snaps', 'Private menus', 'Jobs', and 'ZFS servergroup'. The main content area shows the 'ZFS server group' configuration page. A table lists the members of the servergroup, including local and remote servers. The table has the following columns: group-members, remote server, details, os, backend\_ver, zfs\_ver, auth, and action. Below the table, there is a button labeled '++ add ZFS server'.

group-members	remote server	details	os	backend_ver	zfs_ver	auth	action
localhost-127.0.0.1	MSWin32: Windows	WIN:my-w11	Microsoft Windows [Version 10.0.26200.8246]	cs 26.04.16	2.41552921	96a20 ...	-
cs_omni-192.168.2.203	solaris: illumos	OMN:omnios	OmniOS omnios-r151056-1acba4f5bd	cs 26.04.17b	-	e5309 ...	delete
free_bsd_142-192.168.2.75	off, ping blocked, ip forbidden or timeout:					e5309 ...	delete
free_bsd_14-192.168.2.140	off, ping blocked, ip forbidden or timeout:					96a20 ...	delete
gw10-192.168.2.46	off, ping blocked, ip forbidden or timeout:					492e0 ...	delete
omnio46-192.168.2.44	off, ping blocked, ip forbidden or timeout:					e5309 ...	delete
osx-192.168.2.139	off, ping blocked, ip forbidden or timeout:					96a20 ...	delete
proxmox_1-192.168.2.112	off, ping blocked, ip forbidden or timeout:					79f58 ...	delete
raspberrry4-192.168.2.89	off, ping blocked, ip forbidden or timeout:					e5309 ...	delete
smartos-192.168.2.96	off, ping blocked, ip forbidden or timeout:					e5309 ...	delete
solaris-192.168.2.50	off, ping blocked, ip forbidden or timeout:					e5309 ...	delete
truenas-192.168.2.72	off, ping blocked, ip forbidden or timeout:					e5309 ...	delete
win2019-ad-192.168.2.124	off, ping blocked, ip forbidden or timeout:					e5309 ...	delete

menu ZFS Servergroup with option to add ZFS server

Local management of a single server with napp-it cs is free even commercially.

After setup, a 30 day eval of grouping features is activated

Management of a servergroup of up to three servers is free for noncommercial home use  
Get a free 3x home key or 2day evalkey for tests: [https://www.napp-it.org/extensions/evaluate\\_en.html](https://www.napp-it.org/extensions/evaluate_en.html)

Remote management of a servergroup outside evaluation or 3x homeuse requires a license  
[https://www.napp-it.org/extensions/quotation\\_en.html](https://www.napp-it.org/extensions/quotation_en.html)

### 3.4 Create a Storage Pool

#### 1st step: you need to create a Storage Pool

**ZFS:** use menu „Pools > ZFS Pools > create new Pool“ ex tank

This creates a new pool with a first vdev

- select one disk for a basic pool (no redundancy) or multiple disks for a raid array (mirror, raidZ, Draid, AnyRaid)

**Windows:** use menu „Pools > Windows Pools > create new Pool“

This creates a new Storage Spaces Pool. A Storage Spaces Pool is a basket for disks of any type or size without diskbased redundancy. Redundancy on a Storage Spaces Pool is done per Space and data copies.

#### 2nd step: you need to create a Storage Space (Windows) or ZFS filesystem

**ZFS:** use menu „Filesystems > ZFS Filesystem > create new Filesystem“ ex tank/data

This creates a new ZFS filesystem where size can grow up to poolsize without settings

- manage usage with quotas (max size) and reservations (guaranteed size)

**Windows:** use menu „Filesystems > Windows Spaces > create Storage Space“

This creates a new Space (virtual disk) with a given size and ntfs/ReFS. Redundancy on a Storage Spaces Pool is done per Space and data copies. You can also define if a Space is created on hd or flash or automatic data tiering between hd and flash.

### 3.5 Extend a Storage Pool

**ZFS: use menu „Pools > ZFS Pools > add vdev to extend pool capacity**

- select a single disk for a basic vdev (Raid-0) or multiple disks for a Raid vdev

!! Data is striped over all vdevs in a ZFS Pool. A vdev lost means a Pool lost, so care about a similar redundancy on any vdev.

If you add a Special Vdev, you need redundancy too as a Special Vdev is the final destination of data, this is not a cache drive that can fail. With a Special Vdev, recsize and small block size defines data storage. A zfs rewrite can move data between hd and flash.

**Windows: use menu „Pools > Windows Pools > expand Pool“ to extend pool capacity**

- this adds the disk(s) of any type or size to a Storage Spaces Pool (no redundancy on Pool level).

You can set type of disk (hd or NVMEe for data location or auto tiering)

### 3.6 Repair a Storage Pool

**ZFS: use menu „Disks > ZFS Management > Replace a disk“**

- this replaces a bad disk with a new or cold spare disk. A hot spare replaces automatically

**Windows: use menu „Pools > Windows Pools > expand Pool“ to add new disks and menu**

„Pools > Windows Pools > Shrink“ to remove bad disks. Data that were on a bad disk is lost without a redundancy on Spaces level that makes data copies on multiple disks

### 3.7 Share data over SMB

The best option to share data over lan is SMB as a client for this protocol is available everywhere and it is best at all for multiuser access with ACL restrictions. With SMB Direct and Windows Server this is also the fastest.

<https://www.napp-it.org/doc/downloads/windows-rdma.pdf>

To enable a NFS/SMB share, click „off“ under NFS or SMB (Windows SMB only).

To disable, click on on or the sharename

NAME (properties)	ORIGIN	MOUNTPOINT	SHARENFS	SHARESMB	CANMOUNT	MOUNTED	NBMAND	REC	AVAILABLE	USED	RES	RFRES	QUO	RFQU	SBS	SYNC	COMPR	DEDUP	CRYPT	ATIME	RDONLY
daten1 (pool)	-	/daten1	off	off	on	yes	off	128K	1.34T	5.78T	none	100G	none	none	0	standard	lz4	off	none	off	off
daten1/_Pool_Benchmark	-	/daten1/_Pool_Benchmark	on	_Pool_Benchmark	on	yes	off	128K	1.24T	176K	none	none	none	none	0	always	off	off	none	off	off
daten1/app	-	/daten1/app	off	app	on	yes	on	128K	1.24T	3.29G	none	none	none	none	0	standard	lz4	off	none	off	off
daten1/backup_appliance	-	/daten1/backup_appliance	off	off	on	yes	off	128K	1.24T	1.25M	none	none	none	none	0	standard	lz4	off	none	off	off
daten1/backup_iso	-	/daten1/backup_iso	off	backup_iso	on	yes	on	128K	1.24T	1.17T	none	none	none	none	0	standard	lz4	off	none	off	off
daten1/backup_iso/nvme_nfs	-	/daten1/backup_iso/nvme_nfs	off	backup_iso	on	yes	on	128K	1.24T	172G	none	none	none	none	0	standard	lz4	off	none	off	on
daten1/conny	-	/daten1/conny	off	conny	on	yes	on	128K	1.24T	1.19T	none	none	none	none	0	standard	lz4	off	none	off	off
daten1/conny.bak	-	/daten1/conny.bak	off	conny.bak.g	on	yes	on	128K	1.24T	201M	none	none	none	none	0	standard	lz4	off	none	off	off
daten1/denise	-	/daten1/denise	off	denise	on	yes	off	128K	1.24T	1.30G	none	none	none	none	0	standard	lz4	off	none	off	off
daten1/enc	-	/daten1/enc	off	off	on	yes	on	128K	1.24T	290K	none	none	none	none	0	standard	lz4	off	avail	off	off
daten1/guenther	-	/daten1/guenther	off	guenther	on	yes	on	128K	1.24T	2.25T	none	none	none	none	0	standard	lz4	off	none	off	off
daten1/key	-	/daten1/key	off	key	on	yes	on	128K	1.24T	384K	none	none	none	none	0	standard	lz4	off	none	off	off
daten1/keyserver	-	/daten1/keyserver	off	keyserver	on	yes	on	128K	1.24T	96K	none	none	none	none	0	standard	lz4	off	none	off	off
daten1/media	-	/daten1/media	on	media	on	yes	on	128K	1.24T	610G	none	none	none	none	0	standard	lz4	off	none	off	off
daten1/napp-it	-	/daten1/napp-it	off	napp-it	on	yes	on	64K	1.24T	216G	none	none	none	none	0	standard	lz4	off	none	off	off
daten1/nfs	-	/daten1/nfs	on	nfs	on	yes	on	128K	1.24T	259G	none	none	none	none	0	standard	lz4	off	none	off	off
daten1/targets_w11	-	/daten1/targets_w11	off	targets_w11	on	yes	on	128K	1.24T	133M	none	none	none	none	0	standard	lz4	off	none	off	off
daten1/targets_w11/iscsi_1764668406	-	[zvol 2G]	-	-	-	-	-	-	1.24T	32.8M	none	none	-	-	-	standard	lz4	off	none	-	off

Menu „ZFS Filesystem“ with most important ZFS settings incl sharing

**Warning: Share ZFS filesystems only**, not regular folders as snaps and critical settings are ZFS properties

**Do not use nested ZFS filesystems below a share** as such a nested filesystem can have different settings like casesensitivity, charset or locking behaviour

Hint: **Set zfs acltype** to nfs4 (Linux: posix) and aclinherit + aclmode to passthrough for Windows alike behaviour

### 3.8 Protect data with ACL (access control list)

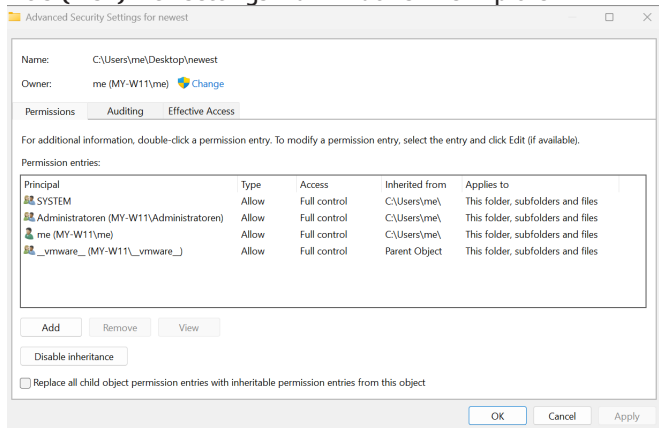
When you enable SMB shares with guest access disabled, you must enter a login name and password to access a share and you can restrict access based on login either for files and folders (file attribute) or shares (share setting). You should set ZFS acltype to nfs4 (if supported) or posix (Linux) and aclinherit + aclmode to passthrough for a Windows alike SMB behaviour (SMB is basically a Microsoft Windows development)

**Set ACL on Windows or Solaris and Solaris forks (native/ kernelbased SMB):**

Use Windows File Explorer and „Property > Security“ for files or folders (locally or remote over lan)

Set ACL for local or AD user and groups (On Windows or Solaris SMB groups can contain groups with fine granular ntfs/nfs4 permissions and ACL inheritance. This is not an option on other Linux or Unix) .

#### Ntfs (nfs4) ACL settings via Windows File Explorer



Advanced ACL settings with Windows File Explorer

### Problems !!

If you move a pool or replicate a ZFS filesystem on Windows or Solaris/ Illumos, ACL remain intact as both use real Windows SID as ACL file reference. On Free-BSD, OSX, other Unix or Linux the file ACL references are simple uid/gid numbers like 101 what can mean a different user on any other server. You need complicated mappings uid-> gid to go around or you must re-assign all ACL after a move.

Another problem is SMB groups in SMB groups. Only Windows or Solaris/ Illumos supports this. All other Linux/Unix do not (Linux/Unid groups cannot contain groups)

Another problem are simple Posix ACL (Linux) vs ntfs/nfs4 ACL. with inheritance. On Linux you use Posix ACL where you can allow read/write/execute for a list of users. Ntfs or nfs4 ACL have more options like create file or folder in this folder only or also in subfolders (inheritance). Posix do not offer ACL inheritance from parent folder(s) that have an immediate effect after modification on parents). With Posix you can only define default behaviours. If something changes, you must redefine all ACL for all files and folders

**Set ACL on Linux (SAMBA or the faster kernelbased ksmbd):**

To use ACL on Linux, situation is quite complicated and a pain compared to Illumos, Solaris and Windows as you

- must create a local user and assign additional SMB passwords with special commands
- must edit smb.conf for general ACL settings
- must change file and folder ACL with setfacl (locally or remote via Putty)

optionally set share ACL

### 3.9 Posix ACL settings on Linux via napp-it cs

#### menu „Filesystems > ACL and Permissions“ (Linux only)

**Posix ACL**

session id 1758788775 invalid Pool Cap Disk Jobs

ACL and permission info

Settings for ZFS filesystem: tank Folder: /

Object	Value	Comment	Option
folder	/tank	folder or filesystem	ZFS filesystem: tank (use pool itself only to set defaults, share non nested sub-fileystems only)
acltype	posix	zfs property	must be posix for ACL on Linux on tank, set to nfsv4 on non-Linux systems
aclmode	passthrough	zfs property	should be passthrough for SMB on tank
aclinherit	passthrough	zfs property	should be passthrough for SMB on tank
xattr	on	zfs property for extended attribute handling	should be sa on tank

User or group	Value	Comment	Option
owner	root	file or folder owner	root or SMB user
ownergroup	root	group of owner	Linux/Unix group
owner permissions	rwX	owner permissions	Linux/Unix user
group permissions	rwX	group permissions	Linux/Unix group
other permissions	rwX	other permissions	other users
modify permission	777 chmod	basic Linux folder permissions	
modify owner	root chown	folder Owner	
modify group	chgrp	folder Group	

**Posix user/group folder ACL (getfacl/setfacl)**

Named user or group	Value	Comment	Option
mask	delete set mask	upper limit for named user/group permissions	d=default, R=recursive
add ACL for user	root rwX add	d=default ACL for new files and folders	R=set recursive for files and folders
add ACL for group	root rwX add	d=default ACL for new files and folders	R=set recursive for files and folders

**Named user/group folder ACL overview**

Named user or group	Value	Comment	Option
---------------------	-------	---------	--------

**Default ACL overview**

Default	Permission	Comment	Options
default:mask	default unset	default max permission for acl	

**User, Groups and members (all SMB users must have set a Linux and SMB password)**

Name	Option
all SMB users	gea, paul, paula, peter, user1, x, y, z
add group	name: [input] add group
del group	group: [input] group must be empty! del group
add groupmember	user: root -> group: root add member
del groupmember	group: root - user: peter del member
add user (Linux+ksmbd)	name: [input] pass: [input] group: (username) add user
del user	user: gea del user
set userpw	user: gea pass: [input] set userpw

**Groups overview**

Group	Members
root	
sambashare	peter, x
staff	
user1	

menu „Filesystems > ACL and Permissions“ (Linux only)

In this menu you can create users, assign to Linux groups. You can display, modify and create Posix ACL and Posix defaults.

#### Warning

If you move or replicate ZFS pools or filesystems between Linux and Windows or Illumos you must re-assign ACL

If you move or replicate ZFS pools or filesystems between Linux hosts you must re-assign ACL or care about mappings or you must create identical users with same uid/gid on any host.

## 4. Special Functions

### 4.1 napp-it ZFS Autoscrub

Napp-it cs autoscrub jobs can verify all data and autorepair corrupt datablocks or bitrot from redundancy, Autoscrub should be run with desktop disks once per month and with enterprise disks twice a year.

### 4.2 napp-it ZFS Autosnap

Napp-it cs autosnap jobs can create snaps per filesystem or recursive with complex keep and hold options based on numbers or age to preserve. You can auto delete zero size snaps or include regular folder in a snap. This is important for Proxmox to include VM settings from /etc/pve in ZFS snaps. Optionally you can call scripts prior or after a job is run.

### 4.3 napp-it ZFS Replication

Napp-it cs replication jobs can replicate ZFS filesystems and zvols from any host with any os to any host (beside OpenZFS to/from Solaris with native ZFS) based on netcat. Netcat is quite the fastest option and is available on any Linux or Unix. For Windows netcat is included in napp-it cs for a copy and run experience. Netcat is the best tool in secure lan environments. Do not use netcat over Internet or insecure lans without VPN. To debug replication jobs, enter rl in napp-it cmd field (both hosts)

Napp-it replication offers complex retention policies like run every 15min and preserve x snaps in last hour, run every hour and preserve y snaps for current day, run every day and preserve z snaps for current month (or per current year).

### 4.4 napp-it ZFS Encryption

You can encrypt whole ZFS pools but you should avoid this. It is better to encrypt filesystems below an unencrypted pool. This allows different keys for different filesystems, you can unencrypt filesystems via zfs send filesystem -> pool and you can use unencrypted filesystems in special cases like VM data with sync enabled. This is quite slow with encrypted filesystems.

When you create an encrypted filesystem in napp-it cs, you can use a short simple and easy to remember password (or any other key). Nappit cs creates a 256hex password hash from this password and the path (pool+filesystem+pw). Such a password has 64 Bytes and avoids characters like `ilIO0` that are hard to read correctly. You can also send them via email or print out for backups. A 64 Byte passphrase is long enough to divide it in three parts where each part has at least 20 Bytes. Each keypart (or the whole key) can be stored on the web-gui host, locally on a server backend host or on one or two (public or private) https servers. For a public https keyserver ex a company or university webserver a `cs_connect` script is included.

To unlock such a filesystem, you can simply click on unlock. Napp-it cs tries then to find all three keyparts (or a complete key) from the options. You can also enter the whole key or the short and easy to remember password to unlock.

Keypart overview

Overview of all keys available in the cs keyfolder /xamp/csweb-gui/\_log/keys  
**Backup them in a save place or print them out and distribute keys locally on member and W1/W2**

ZFS Keypart overview in menu „Filesystems > ZFS Encryption“

## 4.5 Data versioning and disaster backup

Even with the best filesystem and Raid, you can loose data due theft, fire, flash, amok hardware, human errors or bugs

A strategy to avoid dataloss should cover different problems

- Hardware/disk failure

you need redundancy for data (Storage Spaces) or data disks (ZFS Raid) to preseve most current data (backup is good but like bread, always from yesterday)

- Unintentionally deleted files or Ransomware

you need snaps ex 4x for current hour, 24x for current day, 30x for current month, 12x for current year and enough versions

- corrupt OS, example after a bad security or bugfix update

you want ZFS bootenvironments (bootable snaps) and a revovery image to restore OS with an USB bootstick from USB or SMB. This can be Aomei backupper on Windows or Clonezille for any bootdisk, create an image from time to time

- a real disaster like theft, fire, flash, amok hardware, human errors or bugs

For this you need a classic 321 backup, 3x data copies, 2x media, 1x external

External can be cloud, a remote storage server (not same building) or several offline USB disks (beside sync run)

With USB disks and ZFS

- create a sync job (ZFS replication or rsync/robocopy)

- attach disk, with ZFS import pool

- sync datapool -> USB

- export ZFS datapool (or eject with ntfs/ReFS volumes)

With ZFS Export prior USB disconnect is critical. If you disconnect without export, you have waiting io, blocking the USB pool.

You need a reboot to regain access to the ZFS USB pool

Copy on Write filesystems like btrfs, ReFS and ZFS are uncritical regarding a disconnect during write,

If your USB filesystem is Fat\* or ntfs, there is a chance for a corrupted filesystem with dataloss, eject prior disconnect