napp-it

napp-it cs
Client Server Edition

ZFS Storageserver GUI
for (m)any ZFS server

published: 2025-Aug-04 (c) napp-it.org

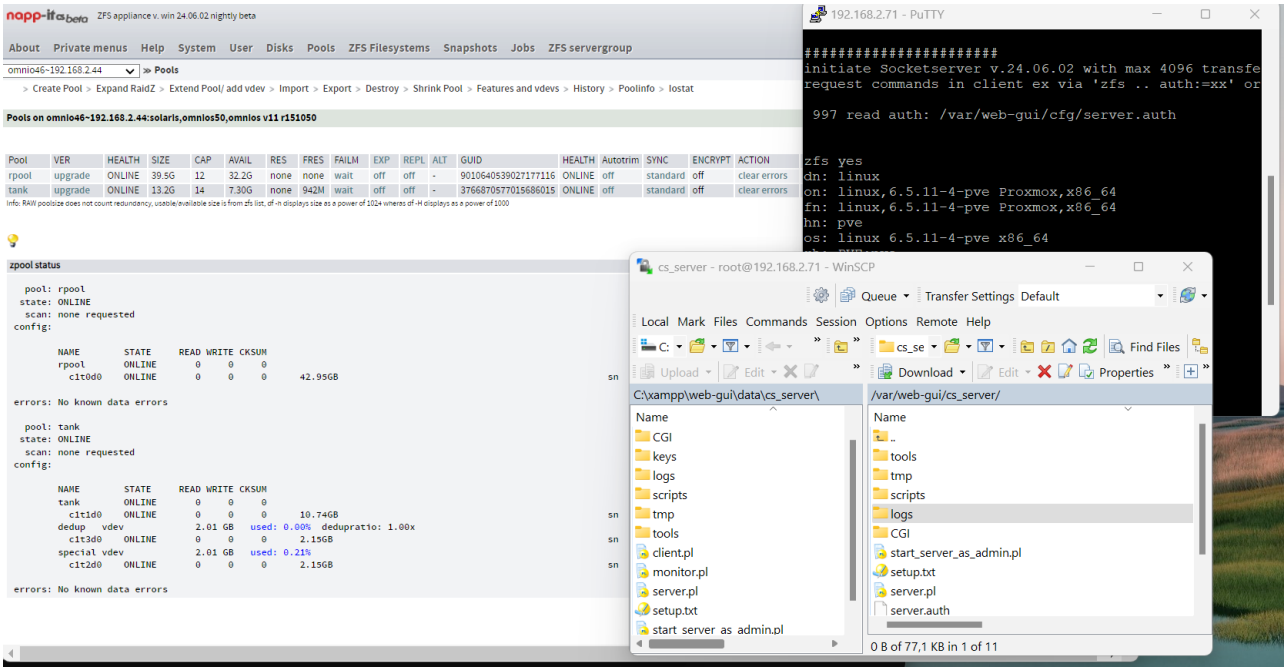
Howto

1. Concept
2. Setup and Management
   web-gui frontend and backend service
3. Features
4. FAQ
5. Technical reference

other manuals
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/illumos-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

1. napp-it cs concept



napp-it cs with Putty and Winscp

About

Napp-it SE (Solaris Edition since 2009) is a webbased management tool for a ZFS server since the early days of ZFS on Sun Solaris, OpenSolaris, Illumos and NexentaCore. Now Open-ZFS is available on BSD, Illumos (Solaris fork with OpenZFS), Linux, OSX and Windows (release candidates). Solaris 11.4 with native ZFS is still an option if you want superiour stability and commercial support for next 10 years. Napp-it gives you ZFS with its unique features like superiour raid concepts with Copy on Write, checksums, unlimited readonly snap views to your datastate with clone and data rollback, safe sync write despite fast rambased writecaches and encryption, compress and realtime dedup per filesystem on any OS. On Windows you can combine the original Windows SMB server with SMB direct that is not only quite the fastest one, but also the most „Windows compatible" one especially regarding ACL permissions where only the Solaris kernelbased SMB server with NFSv4 ACL and direct suppport of Windows sid comes close. Combined with ZFS any OS can be now a dream team. This is why I ported the Solaris based napp-it web-gui for any OS.

Napp-it cs is the key for management of any mixed ZFS environment that allows central management of all members. You can manage any server via web-browser (focus on ZFS, native Solaris ZFS or Open-ZFS and Storage Spaces on Windows) or replicate ZFS filesystems between them (native Solaris ZFS cannot replicate to/from Open-ZFS). This includes low performance devices down to a low RAM ARM Raspberry.

2. Setup and Management of a server or servergroup

Setup

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

Management:

-Open a Browser with adress http(s)://ip of your web-gui server
You may need to open the firewall on the web-gui server ex Windows to allow remote web access.

You can add other servers with running backend services in menu ZFS servergroup

You must sync the ./csweb-gui/cfg/server.auth (with key for remote access on a backend server)
with the memberfile ./csweb-gui/_log/group/member~ip.txt otherwise you get an auth error.

If the server runs the web-gui frontend and the server backend you can manage this server when you select „localhost" in the server selector. If the server is managed remotely as part of a servergroup select the server by its name and ip in the selector. All menus are then related to the selected server, does not matter what OS this server is running on.

3. napp-it cs features:

Napp-it cs is a portable copy and run client server web-gui for a Free-BSD/ Linux/ OSX/ OmniOS/ Windows NAS or multi OS storage servergroup to manage ZFS (Windows: Storage Spaces and OpenZFS).

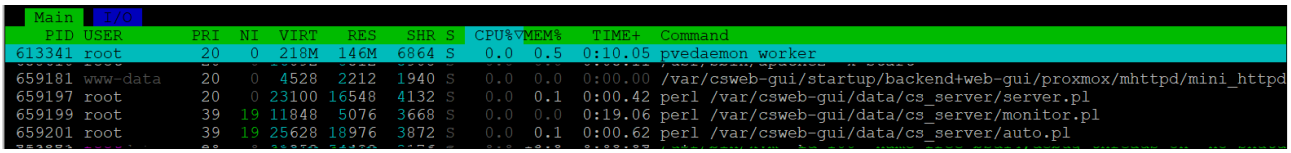
Client server vs cgi

Napp-it se (Solaris edition since 2009) is a cgi based web-gui. This means that the web-gui requests ZFS informations locally ex via a „sudo zfs status" call. While this is quite easy, you need a full web-gui setup on any server in a group, it is Illumos/Solaris only, you must maintain any setup and you have the load of a running webserver on every server.

Napp-it cs (new client/server edition since 2023) works different.
If the web-gui wants a „zpool status" it does not call this command locally but initiates a request for „zpool status" on a remote machine where a server background service is listening to calls, executes the command and returns the result. On Illumos napp-it cs extends the functionality of napp-it se.

Resource needs

Due the client server concept you can remotely manage any server with lowest ram and cpu load for the backend services up from a tiny Raspberry. The more resource hungry webserver frontend is only needed once.



CPU load and RAM need on Proxmox, only a few KB

mini_httpd:	basic http webserver (frontend web-gui)
server.pl:	backend server service
monitor.pl	backend monitor service for command caching (optional)
auto.pl	jobmanagement (optional)

Napp-it se and napp-it cs are optimized for desktop use with a full hd monitor or monitor.
The list items like ZFS filesystems with most important properties and the option to modify them.
It is not an app for mobile phones.

4. FAQ

Is napp-it cs free
Yes, for noncommercial homeuse. If you use it outside private homeuse you need a commercial Pro licence that includes updates and support. You can order single server or multiserver Pro subscriptions on an annual base or with a rebate for several years or perpetual.

Commercial users can request a quotation for a napp-it Pro subscription (this includes napp-it and napp-it cs)
(valid per server, per servernumber or per location): https://www.napp-it.org/extensions/quotation_en.html

Can I modify or redistribute napp-it
Yes for inhouse use. If you want to redistribute or sell you need a licence agreement.

Can I redistribute my own menues and functions that run under napp-it
Yes without any restrictions. If you want to include napp-it, ask for a redistribution licence.

Can I sell or configure servers with preconfigured napp-it
Yes, unless your customer has a Pro license for every server.
Ask for a reseller discount (depend on numbers).

Can I sublicence napp-it to distribute it under my own brand or exclusively for a region?
Yes, ask fo a redistribution license. Costs depend on estimated numbers.

If you already own napp-it se complete (cs is included in newest Solaris Edition):
You can use napp-it cs on same number of servers as you have napp-it SE keys.
With a single napp-it se complete key, you are allowed to manage a servergroup with three members.

Software up/downgrade:
stop services
Up/downgrade copy over newer/older ./csweb-gui/data.new and ./csweb-gui/startup with settings preserved
restart services

Discuss:
https://forums.servethehome.com/index.php?forums/solaris-nexenta-openindiana-and-napp-it.26/

6. Reference

Requirements:

Napp-it cs web-gui is based on an http(s) webserver with Perl and cgi for browser based ZFS management.
We decided to use Xampp portable on Windows for the web-gui as it offers a whole webserver suite with Apache, FileZilla ftp, Mail, mySQL, Perl, PHP and phpMyAdmin and it enables Windows as a ZFS server out of the box.. A physical or virtialized Windows machine should be available everywhere. Size can be minimized by deleting Xampp functions beside Apache and Perl. Napp-it cs is just the /xampp/web-gui folder below a default Xampp. On Proxmox/Illumos you can use the included mini_httpd (http only on port 8080) but prefer Apache with https (**required for servergroup management**).

Napp-it cs server backend (the remote control software that runs on a ZFS server) needs Perl and curl. This is mostly part of a default OS installation on BSD, Illumos, Linux, OSX and Solaris. On Windows it is part of Xampp. As the napp-it cs backend part are basically two small Perl scripts running in the background, RAM and cpu requirement are extremely low. A 64bit CPU is recommended but this is due ZFS.

Napp-it cs_connect (the software that you can use on a public https webserver for encrypted transfers and key downloads) runs on a cgi-capable https webserver. This is a small Perl cgi script without any dependencies. Just upload, set to rwx and edit settings like Perl path, allowed ip and auth value for access.

Platform for the web-gui frontend:
Linux/Proxmox, Windows 10/11/Server, OSX or any other OS with a cgi capable webserver

Platforms for server.pl (remote management)
BSD (Free-BSD 14),Illumos (OmniOS/OpenIndiana/SmartOS) and Solaris, Linux (Debian/Proxmox/Ubuntu), Intel and ARM
OSX (Intel and ARM) and Windows

Platform for cs_connect (key download, encrypted data transfers via a public https server)

Any webserver that allows cgi (/cgi-bin/cs/cs_connect.pl), can be a local or a public https server.

Napp-it cs has three parts

1.) The **„portable Copy and Run"** web-gui frontend that runs on Free-BSD, OSX, Linux, Proxmox, Windows or other systems.

Everything is in c:\xampp\csweb-gui (Windows) or /var/csweb-gui (non Windows).
To „uninstall" just delete the folder.

2.) „Copy and Run" cs_server backend for remote management of ZFS on BSD, Linux (Proxmox), Unix or Windows.

The cs_server app allows management of a remote server exact like the local one. The server app runs on nearly any server as you only need Perl and curl installed., Update or downgrade is a simple stop services copy replace ./data folder with newer/ older one with settings preserved. Use the portable tools Putty for a remote console and WinSCP for upload/download or remote editing of files

3.) „Copy and Run" cs_connect app for https servers W1/W2.
needed for 3rd party https servers

CS connect is a cgi application that allows encrypted https transfers of commands or results from and to your remote ZFS servers and is a possible location for keysplit for encrypted ZFS filesystems. Such a keysplit allows three keyparts locally and on up to two https servers. Not even a server admin has then access to the full key. This additionally allows HA access to encrypted filesystems as you just can unlock a filesystem even when one of the https servers is down. You find the cs_connect folder on Windows in the /xampp/web-gui/data/wwwroot/ cgi-bin/cs folder. The cs_connect app runs locally on Windows and Apache or on your company or university webserver.

No installation required to run cs_connect on your https server, just upload and start. (optional). For encrypted transfers of commands and results or key downloads, you can use the the web-gui Xampp https server with a self signed certificate. You can also use your public https webserver with a valid certificate instead. You can also use cs_connect for a keysplit for encrypted filesystems (3way keysplit, local/W1/W2) or high key availability 2way split (local/W* or W*/W*) where a filesystem can be unlocked when either W1 or W2 is up. Local keyfiles on a ZFS server are neither wanted nor needed.

Data structure

The whole Napp-it cs appliance software is in c:\xampp\csweb-gui (Windows) or /var/csweb-gui (all others)
Within the csweb-gui folder, you find the following subfolders:

1. napp-it cs menues and scripts in (xampp on Windows = var on non/Windows)

c:\xampp\csweb-gui\data	napp-it menues and scripts of a napp-it release
c:\xampp\csweb-gui\data.new	napp-it menues and scripts of a napp-it release update (switched on restart)

To update/downgrade, just stop the web-gui, replace this folder and restart the web-gui

c:\xampp\csweb-gui\startup	startup scripts for web-gui + backend or backemd services only (remote control)

c:\xampp\csweb-gui\data\menues web-gui menues

2. napp-it global settings, common for any release

c:\xampp\csweb-gui_log	napp-it settings, keys, members, passwords or jobs
c:\xampp\csweb-gui\cfg	napp-it defaults
c:\xampp\csweb-gui\tmp	napp-it tmp folder (cleared on napp-it cs startup)

private files (update/downgrade safe) in

\xampp|var\csweb-gui_my

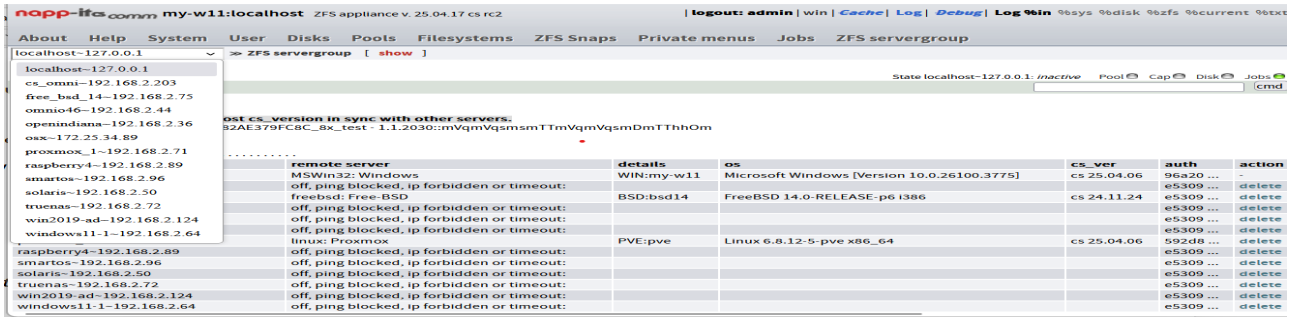
._my\wwwroot	-> data accessible via browser under /my
	-> individual html, logos, js or css files
._my\menues	-> individual web-gui menues
	-> use numbers > 100 with a private menu ex 110_My settings
._my_lib	-> individual language translations, menu settings or scripts

cs_connect

webfolder/cgi-bin/cs	**all files (script, tmp and keyfolder) below**

Uninstall
Uninstall Open-ZFS and delete c:\xampp (on Linux, just delete /var/csweb-gui

ZFS Server groups



After an initial setup, you can manage only the local machine ex with web-gui on Proxmox or Open-ZFS on Windows.
To manage another ZFS server you must

Backend (remotely managed server)
- upload csweb-gui to /var
- start backend service server.pl
  either locally or via Putty: perl /var/web-gui/cs_server/start_server_as_admin.pl

For Windows as a member
start „C:\xampp_start_server_only_as_admin.bat“ as admin

Frontend (server with web-gui):
- add server as a member in menu ZFS servergroup: +add
- sync local auth file example „C:\xampp\csweb-gui_log\group\windows11-1~192.168.2.64.txt“
  with backend auth file „C:\xampp\csweb-gui\cfg\server.auth“ (Linux under /var/csweb-gui)

Restrictions:

OSX: user and share management: OSX System Settings
TrueNAS: manage locally as settings outside the TrueNAS gui can give problems.
SmartOS: most OS settings like users and shares are temporary (smb anonymous share is ok)
Proxmox: no retrictions but everything outside ZFS and basic user and SMB management should be in a VM

For a ZFS server as cs_member, prefer a 64bit CPU/OS and at least 4 GB RAM
Without ZFS, even a 512 MB ARM appliance should be manageable via napp-it cs web-gui.

To manage a remote server
- add server to ZFS Servergroup
- select server

all menu options are now processed against the selected server

Main features

Job management

To execute jobs in auto modus, the auto service must run (started together with the web-gui)

Replication (from any to any
create a replication job with a source and destination among members.
You can use a different snap retention policy for replications

Autosnap
create a snap job with a retention policy

Alerts and status reports (currently Windows only)
create a report job (status and/or alerts for any ZFS memberserver)

Other jobs
either as a simple command ex rsync or as a script.

SMB and NFS shares
are managed as properties of a ZFS filesystem. Avoid sharing of regular folders or nested ZFS filesystema

Encrypted filesystems

Create encrypted filesystems based on a key or password hash where the key is splitted in three parts.

You can enter a key or a short easy to remember pw from which a sha pw hash is created as key.
To unlock you can enter the pw or the key

To lock a filesystem, click on „avail" in „ZFS Filesystems"
To unlock a filesystem, click on „unnavail" in „ZFS Filesystems"
After creation of an encrypted filesystem, the local key on a ZFS server in the ./cs_server/tmp folder is deleted (Do not store whole keys on a server itself). You find the keys on your Windows web-gui in \xampp\web-gui_log\keys as a completekey and the three keyparts. Name of key is member~pool~filesystem.keypart 1-3 or .completekey. Backup at least the .completekey to a safe location. Try a lock/unlock with the keyfile and directly with the key.

Keysplit
If you simply press enter in the unlock menu, cs_server tries to find the completekey or the three keyparts either in the local keyfolder ./cs_server/keys or remotely from a https webserver W1 (web-gui or a public https server) or optionally W2 when configured in About > Settings or enforced in server.pl. On your web-gui, keys are in /xampp/web-gui/_log/keys. On an external keyserver with cs_connect, keys (complete or parts) are in /cgi-bin/cs/keys. Access to a keyserver requires an auth key and should be ip restricted.

3 way keysplit
In such a configuration, place the first keypart in the local keyfolder, and the second and third part in the keyfolder on W1 and W2. Only If cs_server can load the missing keyparts from W1 and W2, it can unlock a filesystem.

2 way HA keysplit
In this configuration keypart 1 is in the local keyfolder and keypart 2 and 3 on W1 and W2. If W1 or W2 is up, cs_server can unlock a filesystem.

The idea behind:
Even if an attacker gets access to a ZFS server or a keyserver, he cannot unlock a filesystem as he needs access to both, in case of 3way split to all three machines as each server has only a part of the key. This even locks out a server admin. The 2way HA method is perfect if you want to ensure access even during webserver maintenance as you do not need to reconfigure anything. As the keyserver is only a small cgi script with keys as files on the keyserver there are no dependencies there.

As a fallback option, you can always unlock a filesystem with the password that you entered during creation (an option) or with the full key. Distribution of keys between local and W1/W2 keyfolders is a manual process under your control. Use WinSCP to upload, check or delete remote keys. As passphrase keys are printable characters you can send them via email or print them out to backup.

Security

Security is always a compromise between performance, easyness of handling and needed software components. This is an important apspect in a client server architecture where critical data like commands (zfs destroy ..), results (get *.config) or keys for encrypted filesystems are transferred over your network.

Napp-it cs adresses this on three levels.
1. all client server connections must be authenticated with an auth string (default a SHA256 hash)
2. all connections can be (and should be) ip restricted to allow only the web-gui (s) to access a server

Only very basic commands like on (are you up and what is your OS familiy and OS release) are remotely done under this restriction alone. All other commands, results and key transfers are additionally encrypted. To encyrypt data transfers there are the two common options ssh server and https server. The complicated and critical part of encryption is done on server side with quite simple clients like your webbrowser or curl. As napp-it cs server should run after a simple copy action with an ultra low resource consumption on ZFS server side. To encrypt all transfers over https as this is what is already there in the web-gui.

Call me back
To avoid installation of a full featured https (or ssh) server with the needed regular key updates on any ZFS memberserver napp-it cs is using a „callback" method where the web-gui tells the ZFS server „there is work for you under this file reference". The ZFS server then loads the file reference via curl and https either from the local Xampp Apache webserver of the web-gui with usually a self signed certificate or a separate webserver ex a public webserver with a valid certificate. (Using an external https with a valid certificate increases security but lowers gui performance)

Keys for encrypted filesystems
Keys are splitted into three parts. A unlock command from the web-gui either sends a pw from which a key is generated, the „real" key or a simple unlock request without key transfers via the callback https method. On a simple unlock request, a ZFS server tries to find all keyparts itself either locally or remote from one or two https servers (W1 and W2). This encrypts all key transfers and avoids local keys. Keysplit also avoids that an admin ex of a remote https server has access to keys.

Cache
Napp-it cs can cache results or recache commands via monitor.pl to increase performance if the same command is repeated or to deliver common results „read ahead" up from the moment the web-gui contacts a ZFS server with the help of the monitor service on a server.

You can disable caching in the napp-it cs top menu (cache enable/disable)

Debug
If you are interested what napp-it is doing, you can enable debug mode in the top menu.
Napp-it cs then dispays comminication details. and you can list the content of internal hashed lik %zfs, %disks and %current where napp-it has collected all needed information prior processing an action.

If you are interested what a server is doing
Open a Putty console and (re)start cs_server

For the local Windows Web-GUI in debug mode
In the taskbar you find an icon with 4 red points. If you click on it it opens thre cmd Windows with the background services server (local cs_server), monitor (local monitor service) and auto (task services)

Monitor jobs
Open a powershell admin terminal in Windows and start a job via (powershell allowes /) ex
\xampp\perl\bin\perl C:\xampp\web-gui\data\menues_lib\windows\scripts\job-report.pl run_123456
(replace 123456 with real jobid)

About Windows Storage Spaces

Windows Storage Spaces can be very confusing, mostly because of the different Windows tools where each only offers a limited set of options, the need to learn Powershell for proper settings and a very inconsistent way of naming between Docs, tools and Powershell commands.

I try to use the following:

1. Physical Disks are HD, SSD, NVMe.
Powershell also lists virtual disks with the get-physicaldisk command

2. Virtual Disks are those based on a file (.vhdx)
In the docs, Storage Spaces are also often named virtual disks (very confusing)

3. Volumes and Partitions
This is what you see in Explorer, ex a NTFS, ReFS or ZFS disk, usually with a driveletter

4. Storage Pool
This is a blackbox where you throw in your physical disks.
In the docs use of this term is often mixed with Storage Space

5. Storage Space
This is a Virtual Device that is treated like a disk as you can place partitions on them.
Redundancy is defined here (not on Pool or Disk level)
In the docs, Storage Spaces are also often named virtual disks (very confusing)

6. SMB Storage Cluster
This is a setup with a whole node treated like a physical disk. Connectivity is over SMB via vhdx files
This setup works with Windows 10/11 and Server. Windows Server adds SMB Direct/RDMA
This is different to Microsoft S2D Cluster which is Windows Server only.

Compatibility
of napp-it cs with napp-it SE (Solaris edition)

You can run napp-it cs (Illumos, Linux, Solaris) on port 8080 as a memberserver in napp-it cs together napp-it se.
If you want to continue napp-it replications in napp-it cs, create the jobs there newly with same source, destination and jobid.

Settings

Basic settings in menu About > Settings

Admin and Operator password	
Language	Use win
Management allowed from	Restrict clients from web-gui access
Grouping allowed from	currently not used
Swap remote adress	If you use a nat vpn, napp-it detects the vpn ip instead the real remote client. Swap can override this

Webserver and Encryption settings

Upserver	no value: callback disabled for requests < 4k „ip“: use calling ip (web-gui) for callback „www.server.com“: use this external server for callback
keylocation:	always /xampp/web-gui/_log/keys
W1 (first webserver for keypart download)	always localhost Apache
W2 (second webserver)	ex: www.server.com
access_id for upsrv/ W2 access	always /xampp/web-gui/_log/members/localhost~127.0.0.1.txt
restrict keyserver access	enter ip or range for zfs server that are allowed to download keys ex 192.168.2.
enable local keyserver	enable/disable keyserver

Upser, W1,W2, allowed ip can enforced differently in server.pl or in cs_connect.pl

Notification settings
email or push data

Hide disks	these disks cannot be selected ex in Pool > Create/Add

For a production server with Open-ZFS, currently prefer ZFS on BSD or Linux or the Solaris fork Ilumos especially OmniOS that use Open-ZFS as upstream but integrates newer features only after additional tests has a proven stability with a ultra low issue rate. Best ZFS stability still has native ZFS from Oracle but this is not free nor compatible with Open-ZFS.