napp-it

napp-it cs
Client Server Edition

ZFS Encryption

published: 2024-Sep-03 (c) napp-it.org

Howto

1. ZFS Encryption

   Advantages
   Handling
   Napp-it cs improvements

2. Create encrypted filesystems

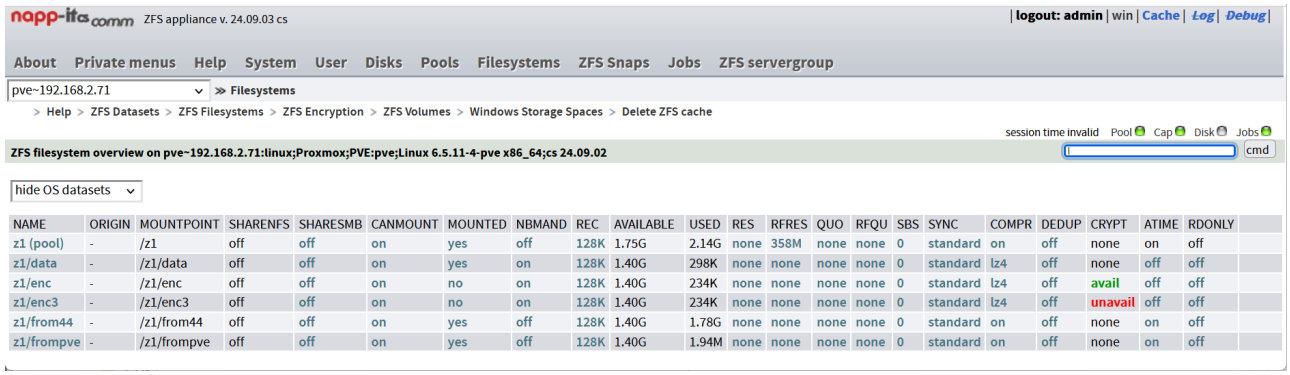
   Step by Step

3. Keysplit and Keyserver

   Howto
   Keyserver (your company webserver)
   HA setup

4. Info and FAQ

1. ZFS Encryption



NAME	ORIGIN	MOUNTPOINT	SHARENFS	SHARESMB	CANMOUNT	MOUNTED	NBMAND	REC	AVAILABLE	USED	RES	RFRES	QUO	RFQU	SBS	SYNC	COMPR	DEDUP	CRYPT	ATIME	RDONLY
z1 (pool)	-	/z1	off	off	on	yes	off	128K	1.75G	2.14G	none	358M	none	none	0	standard	on	off	none	on	off
z1/data	-	/z1/data	off	off	on	yes	on	128K	1.40G	298K	none	none	none	none	0	standard	lz4	off	none	off	off
z1/enc	-	/z1/enc	off	off	on	no	on	128K	1.40G	234K	none	none	none	none	0	standard	lz4	off	avail	off	off
z1/enc3	-	/z1/enc3	off	off	on	no	on	128K	1.40G	234K	none	none	none	none	0	standard	lz4	off	unavail	off	off
z1/from44	-	/z1/from44	off	off	on	yes	off	128K	1.40G	1.78G	none	none	none	none	0	standard	on	off	none	on	off
z1/frompve	-	/z1/frompve	off	off	on	yes	off	128K	1.40G	1.94M	none	none	none	none	0	standard	on	off	none	on	off

ZFS offers encryption per filesystem with the following advantages:

- You can use different keys for different datasets (filesystems or volumes) depending on use cases or users
- You only need to unlock the filesystem you are working on, not the whole pool
- You can unlock via prompt, files or on some ZFS versions via an https keyserver
- You can always unlock via prompt, even when normal keysource is file
- You can backup/replicate encrypted filesystems either unencrypted or in encrypted raw mode (preserves key)

ZFS encrypts data per ZFS datablock not the datastream from disk

This is why encryption is transparent to compress or dedup but this reduces performance with small datablocks especially sync write loggings where you must write very small datablocks that cannot be encrypted efficiently. On a very fast ZFS server and an Intel Optane pool you can get a sequential nonsync write performance > 2000 MB/s. With encryption enabled you can get 1000 MB/s that goes down to maybe 100 MB/s when you enable sync. .

If you need sync and performance, avoid encryption!!
This is important for VMs on ZFS ex with ESXi, Proxmox or SmartOS where you need sync write to protect VM filesystems

Handling of encrypted filesystems
Some thumb rules

Encryption protects your data from anyone who has physical access to your offline server or disks, it cannot protect online data when a dataset is unlocked ex against Ransomware attacks. For this you need backups and snaps for a rollback.

Never store a filebased key on the same server to unlock as any thieve will be able to unlock your data then.
Prefer keys on a remote https server with encrypted transfers

Use long keys. Key quality depends on key length, not key complexity.
A good key has 20 random characters or more

Use keys with printable characters and avoid characters like „o0O" or „iIlL".
Prefer sha256hex keys to avoid problems like short keys or misreading.

Backup/Replicate data in raw mode.
In this case the zfs send datastream is encrypted with the key of the source filesystem.

Do not encrypt on pool level
to be able to create unencrypted filesystems or unencrypt already encrypted filesystems

Encrypt datasets (filesystems or volumes) depending on use case

Avoid nested encrypted datasets or when using SMB of filesystems.

Most important !!
Try a lock/ unlock after dataset creation prior further use.
Backup your keys on multiple locations or print them out
Without the key your data is lost.

How can napp-it cs improve handling of encrypted ZFS datasets

- Keys or keyparts can be local or remote on an https server (encrypted transfers)
  - You can use the included web-gui https server as keyserver with self signed certificates
  - You can use your company/university webserver(s) with public certificates
  - You can use two webservers w1 and w2
  - Keyserver access is passphrase protected and can be ip restricted

- Keys are sha256hex format
  - Keys are readable/ printable without confusion about o00 or liLl characters
  - You can generate a passphrase or create it as hash from a short and easy to remember password

- Keys are splitted in three parts
  - You can distribute the keyparts locally and on w1 or w2 (each keypart with sha256hex > 20 char)
  - During unlock, napp-it cs actively searches missing keyparts, just distribute them
  - Fast overview of keys available on different locations

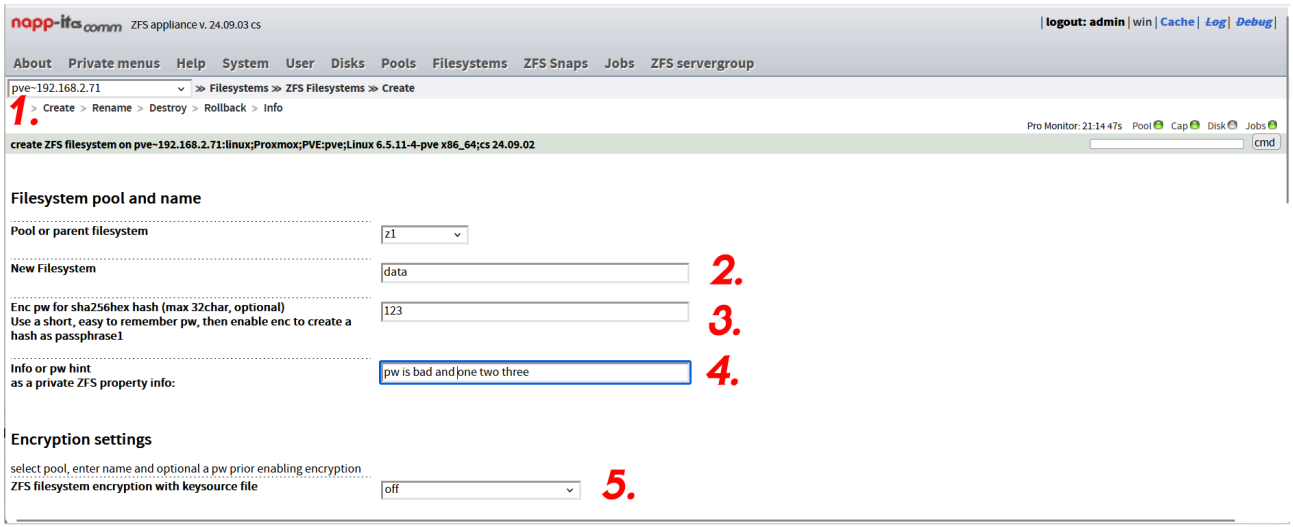
- Keys are „admin" protected
  - Not even local root or an admin of an https server has access to the full key

- Keys can be entered directly during unlock
  - Unlock possible even when Internet is not available
  - Either as whole key or a short pw when the key was hash generated from short/ easy to remember pw

- HA setup
  - Use keypart 1 of a key locally and keypart 2/3 on w1 and w1 to allow an unlock if either w1 or w2 are offline

2. Create ZFS encrypted filesystems

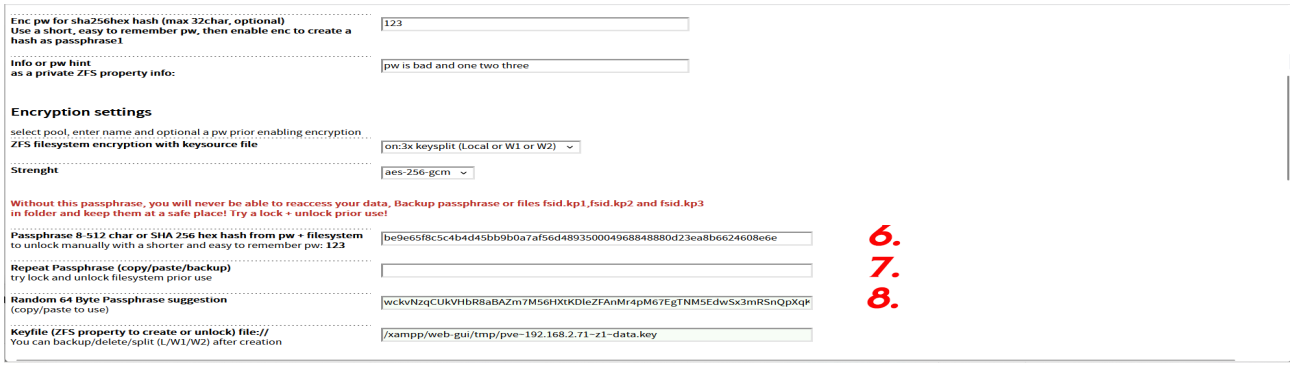
Use menue Filesystems > ZFS Filesystems > Create



1. Select memberserver from memberlist ex Proxmox pve
2. Select pool and enter name of the filesystem you want to create
3. Enter a short and easy to remember pw to create a sha256hex from (option)
4. Enter a password hint or any other description about the filesystem (option)
5. Swich Encryption to on: 3x keysplit

Menu will change and show additional encryption settings

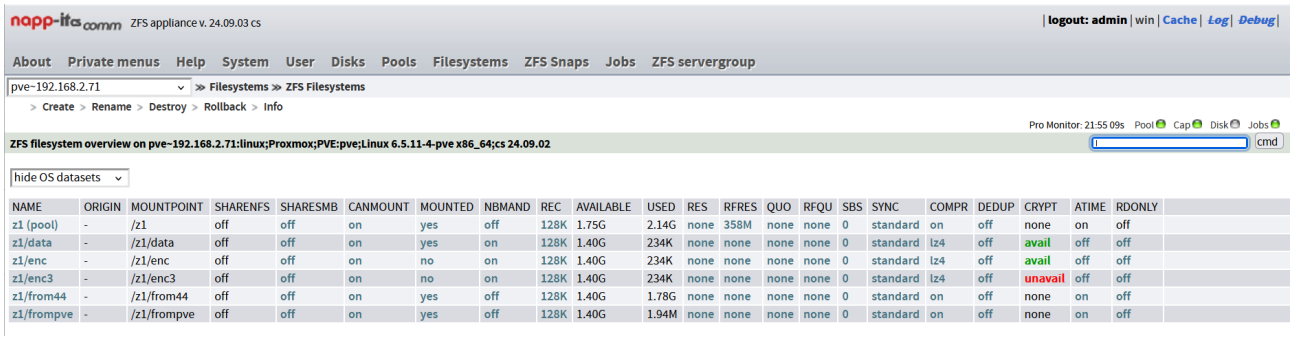
Additional Settings with encryption = on



6. This is the key, sha256hex generated from the pw „123“
If you want to use this key, copy/ paste it to 7.

You can optionally copy the passphrase from 8. into field 6. and 7.
or use any other passphrase.

Submit the form and you have created the filesystem ex z1/data

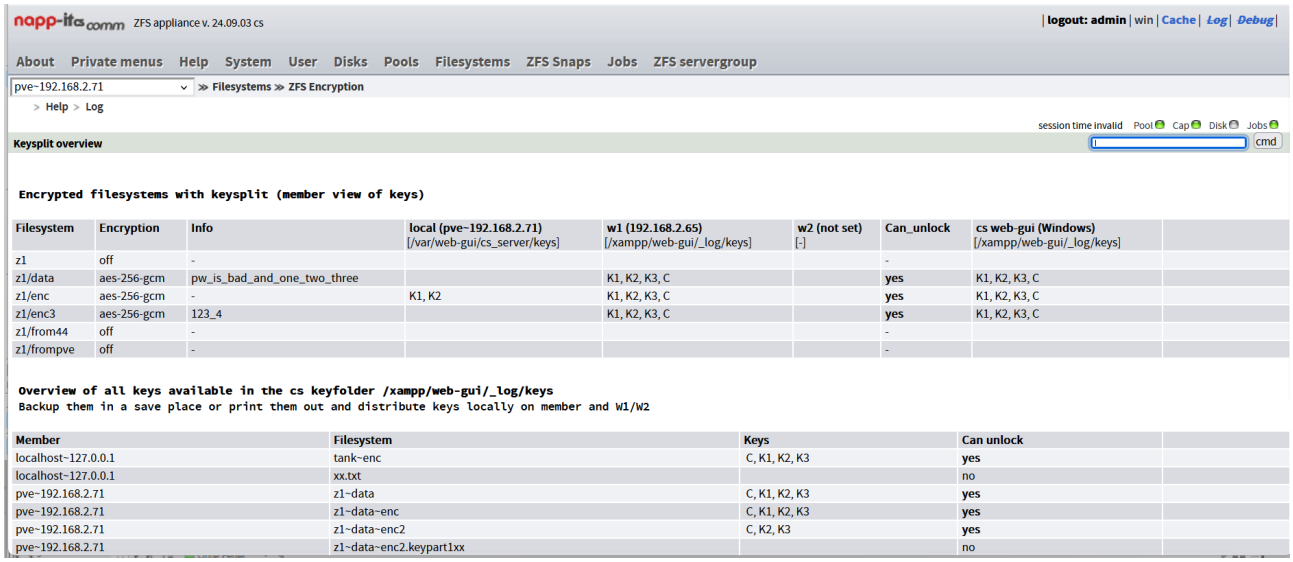


NAME	ORIGIN	MOUNTPOINT	SHARENFS	SHARESMB	CANMOUNT	MOUNTED	NBMAND	REC	AVAILABLE	USED	RES	RFRES	QUO	RFQU	SBS	SYNC	COMPR	DEDUP	CRYPT	ATIME	RDONLY
z1 (pool)	-	/z1	off	off	on	yes	off	128K	1.75G	2.14G	none	358M	none	none	0	standard	on	off	none	on	off
z1/data	-	/z1/data	off	off	on	yes	on	128K	1.40G	234K	none	none	none	none	0	standard	lz4	off	avail	off	off
z1/enc	-	/z1/enc	off	off	on	no	on	128K	1.40G	234K	none	none	none	none	0	standard	lz4	off	avail	off	off
z1/enc3	-	/z1/enc3	off	off	on	no	on	128K	1.40G	234K	none	none	none	none	0	standard	lz4	off	unavail	off	off
z1/from44	-	/z1/from44	off	off	on	yes	off	128K	1.40G	1.78G	none	none	none	none	0	standard	on	off	none	on	off
z1/frompve	-	/z1/frompve	off	off	on	yes	off	128K	1.40G	1.94M	none	none	none	none	0	standard	on	off	none	on	off

In this example we have three encrypted filesystems, two are available and one is unavailable/locked.
To test a lock/unlock sequence of the just created z1/data filesystem, click on the green avail in the line of z1/data and confirm „lock“. The crypt state is changing to a red unavail. Now click on the red unavail and confirm the shown temp filepath to get status avail again. This filepath is not used and was needed only as a temp value to create an encrypted filesystem without an interactive console prompt. You can overwrite the filepath optionally with the full key or the short easy to remember password („123“) to unlock the filesystem for example when the key is remote on a webserver and Internet is offline.

Key overview

see menu Filesystems > ZFS encryption with a memberserver view of keys
The keys of the just created filesystem are under /xampp/web-gui/_log/keys



Encrypted filesystems with keysplit (member view of keys)

Filesystem	Encryption	Info	local (pve~192.168.2.71) [/var/web-gui/cs_server/keys]	w1 (192.168.2.65) [/xampp/web-gui/_log/keys]	w2 (not set) [-]	Can_unlock	cs web-gui (Windows) [/xampp/web-gui/_log/keys]
z1	off	-				-	
z1/data	aes-256-gcm	pw_is_bad_and_one_two_three		K1, K2, K3, C		yes	K1, K2, K3, C
z1/enc	aes-256-gcm	-	K1, K2	K1, K2, K3, C		yes	K1, K2, K3, C
z1/enc3	aes-256-gcm	123_4		K1, K2, K3, C		yes	K1, K2, K3, C
z1/from44	off	-				-	
z1/frompve	off	-				-	

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

Member	Filesystem	Keys	Can unlock
localhost~127.0.0.1	tank~enc	C, K1, K2, K3	yes
localhost~127.0.0.1	xx.txt		no
pve~192.168.2.71	z1~data	C, K1, K2, K3	yes
pve~192.168.2.71	z1~data~enc	C, K1, K2, K3	yes
pve~192.168.2.71	z1~data~enc2	C, K2, K3	yes
pve~192.168.2.71	z1~data~enc2.keypart1xx		no

3. Keysplit

After you have tested a lock/unlock sequence, you should backup the key on two different locations as a key lost is a data lost.
The keys of created encrypted filesystems are under /xampp/web-gui/_log/keys ex (key of z1/data is saved as file z1~data).
Then you can delete/distribute the keyparts to the local keyfolder of a ZFS server or a webserver

Keyfiles:
z1~data.keypart1
z1~data.keypart2
z1~data.keypart3
z1~data.completekey (part1+part2+part3)

To unlock a filesystem, you need the completekey but the unlock command in napp-it cs can unlock a filesystem if it finds the keyparts 1-3 either in the local keyfolder /var/web-gui/cs_server/keys or on a remote https webserver in ./cgi-bin/cs/keys.

Keys for the local Windows ZFS server
In this case, the keyfolder is also the keyfolder of the local web-gui webserver (w1). As a memberserver is using this server as default w1 webserver, you can lock/unlock filesystems ex on a Proxmox memberserver without additional settings. No key is needed on Proxmox.

2x location keysplit
Distribute keypart1 on Proxmox and keypart 2+3 on webserver w1.

3x location keysplit
Distribute keypart1 on Proxmox and keypart 2 on webserver w1 and keypart 3 on webserver w2

HA keysplit (redundant webserver)
Distribute keypart1 on Proxmox and keypart 2+3 on webserver w1 and keypart 2+3 on webserver w2

Info
To unlock a filesystem, you need either keypart 1-3 or the completekey on any location
To set w1 not to the web-gui but to another webserver (your company/university webserver):
- set w1 or w2 in cs_server.pl
- upload cs_connect (folder „C:\xampp\web-gui\data\wwwroot\cgi-bin\cs“) to the cgi-bin folder of your webserver
- set „C:\xampp\web-gui\data\wwwroot\cgi-bin\cs\cs_connect.pl“ to executable
- edit „C:\xampp\web-gui\data\wwwroot\cgi-bin\cs\cs_connect.pl: first line must be #!/usr/bin/perl

To distribute keys between ZFS server and w1/w2, use WinSCP
local keyfolder: /var/web-gui/cs_server/keys
local keyfolder (Windows localhost): /xampp/web-gui/_log/keys
w1 keyfolder (Windows localhost): /xampp/web-gui/_log/keys
w1/w2 keyfolder (company/university webserver): ./cgi-bin/cs/keys

3. Setup and FAQ

Setup napp-it cs: https://www.napp-it.de/downloads/napp-it-cs_en.html.
Manuals: https://www.napp-it.de/manuals/index_en.html
Is napp-it cs free: yes for noncommercial homeuse for up to three serves