napp-it

napp-*it*

ZFS Storage Server
User's Guide

Setup
First steps

published: 2018-Dec-20 (c) napp-it.org

Content:

1. The Solaris Family - OPENSOLARIS based operating systems

Developed by Sun with its initial release in 2008, based on Unix System 5, Release 4 with revolutionary features like ZFS (http://en.wikipedia.org/wiki/ZFS), Comstar (enterprise ready iSCSI and FC technology), Dtrace, Crossbow virtual networking, virtualization with Zones (http://en.wikipedia.org/wiki/Solaris_Zones), ZFS integrated kernel based CIFS/SMB and NFS server among other features like ZFS boot mirrors, disk unique WWN enumeration or fault management service with auto hotspare support.

If you want to follow the way from Sun OpenSolaris to the free forks like the current Illumos based distributions, you should look at Bryan Cantrills (Joyent) slideshow at
http://www.slideshare.net/bcantrill/fork-yeah-the-rise-and-development-of-illumos or
http://www.youtube.com/watch?v=-zRN7XLCRhc

Some essentials from the slideshow:

„In the mid-1990 it becomes a foregone conclusion that UNIX would die at the hands of Windows NT Hard to believe, but SUN seems the only computer vendor whose Windows NT strategy was to beat Windows NT

Sun's dedication to this vision - the operating system as a nexus of innovation - attracted an entire new generation of engineers to the company Development started on more radical ideas, each of it would become revolutionary in its own right (ZFS, DTrace, Zones, Crossbow, Comstar, Fault Management, Service Management, Least Privilege) These were technologies invented and initiated by engineers, not managers or marketers These projects reflected the people behind them..Organizations don't innovate - people do

As the rise of Linux forced the market price of OS aquisition to zero, that open sourcing the (Solaris) operating system was the right business decision (in 2005. Unfortunately, not all elements of the OS could be open-sourced, some contracts prevented some small but important bits from beeing opensourced. To allow such proprietary drivers, Sun developed a file based copy left licence (CDDL) ... this was not done to deliberatly GPL incompatible!

Ailing Sun was bought by Oracle in 2009. Over 2010, it becomes clear that Oracle had obsolutely no interest in OpenSolaris... There was.. a move to close the system (OpenSolaris)

Starting in the summer of 2010, Garett D'Amore at Nexenta - with the help of Rich Lowe, Jason King and others - began the process of ether writing the closed bits from scratch or porting from BSD. Dubbed „Illumos" (from illuminare, Latin for illuminate) and made available on August 3, 2010

Illumos was not designed to be a fork, but rather an entirely open downstream repository of OpenSolaris

Solaris 11 was released on November 9, 2011 - and there was no source release
The entire DTrace team, all primary ZFS inventors and primary engineers for zones and networking had left Oracle.. nearly all of these engineers went to companies betting on Illumos.

In Illumos, we have seen critical innovations and bug fixes in ZFS, DTrace, Zones and other core technologies. These innovations ill never be in Oracle Solaris. Joyent team ported KVM from Linux. Illumos distributions SmartOS and OpenIndiana have KVM support by default."

Solarish

If we talk about common features of Oracle Solaris and the free Solaris fork Illumos (ex OmniOS, OpenIndiana, SmartOS) the term Solarish is common.

2. ZFS Distributions

Unlike storage appliances that are based on their own distribution of BSD, Illumos or Linux,
napp-it is a „Nasifier" for some general-use enterprise operating systems that you can keep up to date like

Based on Solarish

- Oracle Solaris 11.4 (commercial OS) www.oracle.com/technetwork/server-storage/solaris11/downloads
- OpenIndiana Hipster (with a desktop option), community project based on Illumos www.openindiana.org
- OmniOS (free and stable Solaris fork), community project based on Illumos www.omniosce.org

Wiki:	https://github.com/jfqd/OmniOSce-wiki
Downloads:	https://downloads.omniosce.org/media/ or a mirror like http://openzfs.hfg-gmuend.de
Community repo is	https://pkg.omniosce.org/r151022/core/en/index.shtml and
Changelog	https://github.com/omniosorg/omnios-build/blob/r151022/doc/ReleaseNotes.md

Based on Linux (beta, no support)

- Ubuntu
- Debian

Linux - support is limited to ZFS management, Autosnap, Autoscrub and AutoJob features
and does not include the advanced features of the Solaris release.

Between ZFS distributions you can move ZFS pools with the follwing restrictions
- From/to Oracle Solaris: Pools must be V28, ZFSv5
- From BSD based systems: Possible with GEOM or with GPT partitions spanning the whole disk

- Beside that: OpenZFS distributions must support same features. No problem with current releases.

The reasons for a Solarish based ZFS System

- 100% focus on ZFS that was developed for Solaris 10 years ago and is most stable and complete there
- a complete storage OS from one hand, not the toolbox with a core OS and many options and vendors
- stable support for mirrorred ZFS boot systems with boot environments (restore a former bootable OS state)
- WWN enumeration of disks (disk unique identification) to keeps disk id identical over controller, server or OS
- fault management with active hot-spares that replaces faulted disks automatically
- SMB, NFS and iSCSI integrated in the core OS and maintained by Oracle or Illumos (OmniOS)
- virtual networking with virtual switches and virtual nics with vlan support
- service management SMF with service auto restart

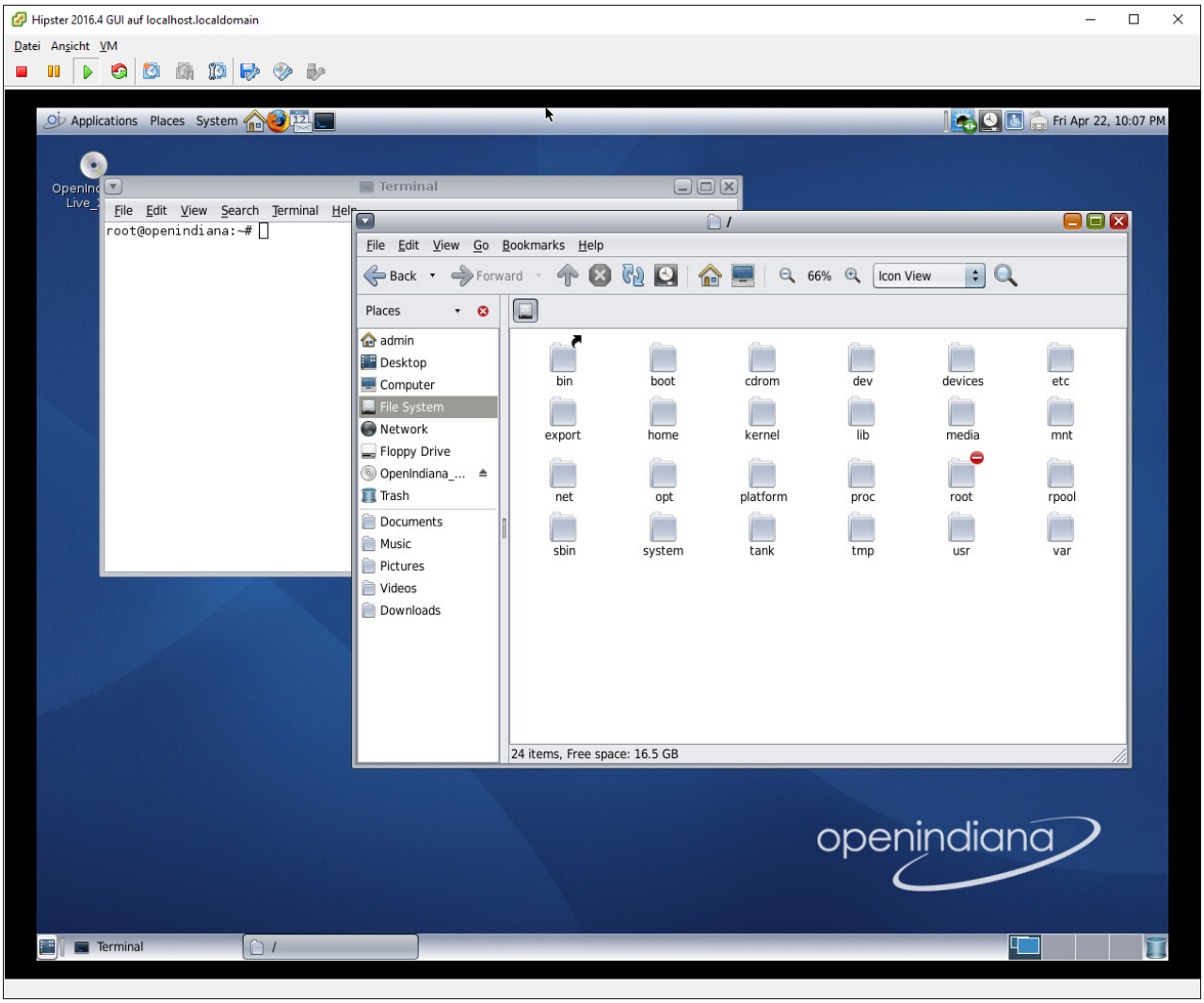
- Solarish CIFS server with NFS4 ACL (more Windows NTFS alike than Posix ACL) and AD support,
  Windows SID as extended ZFS attribute (Permissions stay intact after a restore to another AD server),
- ZFS snaps as Windows „previous version", stable and just working out of the box since years

OS setup manual for all OS options

see http://www.napp-it.org/doc/downloads/setup_napp-it_os.pdf

2.1. OpenIndiana Hipster

as minimal server text edition, regular text edition or with a Mate GUI



You can run napp-it on OpenIndiana Hipster 2016.10 (OpenSource) or Oracle Solaris 11.3. Both support SMB 2.1 and come with a GUI. You manage storage via the napp-it Web-UI but the local GUI helps to transfer and organize data locally or to setup things like ip v6. If you want the GUI in production systems, prefer Solaris.

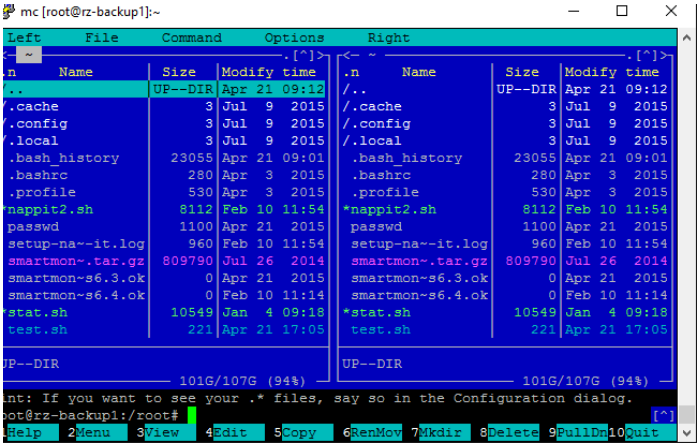
Alternatively you can use the „text-edition" of OpenIndiana or Solaris or OmniOS, a minimalistic and very stable OpenSource distribution for storage and other production use cases (This is the preferred platform for napp-it) Details see **http://www.napp-it.org/doc/downloads/setup_napp-it_os.pdf**

2.2. OmniOS

or OpenIndiana or Oracle Solaris server edition

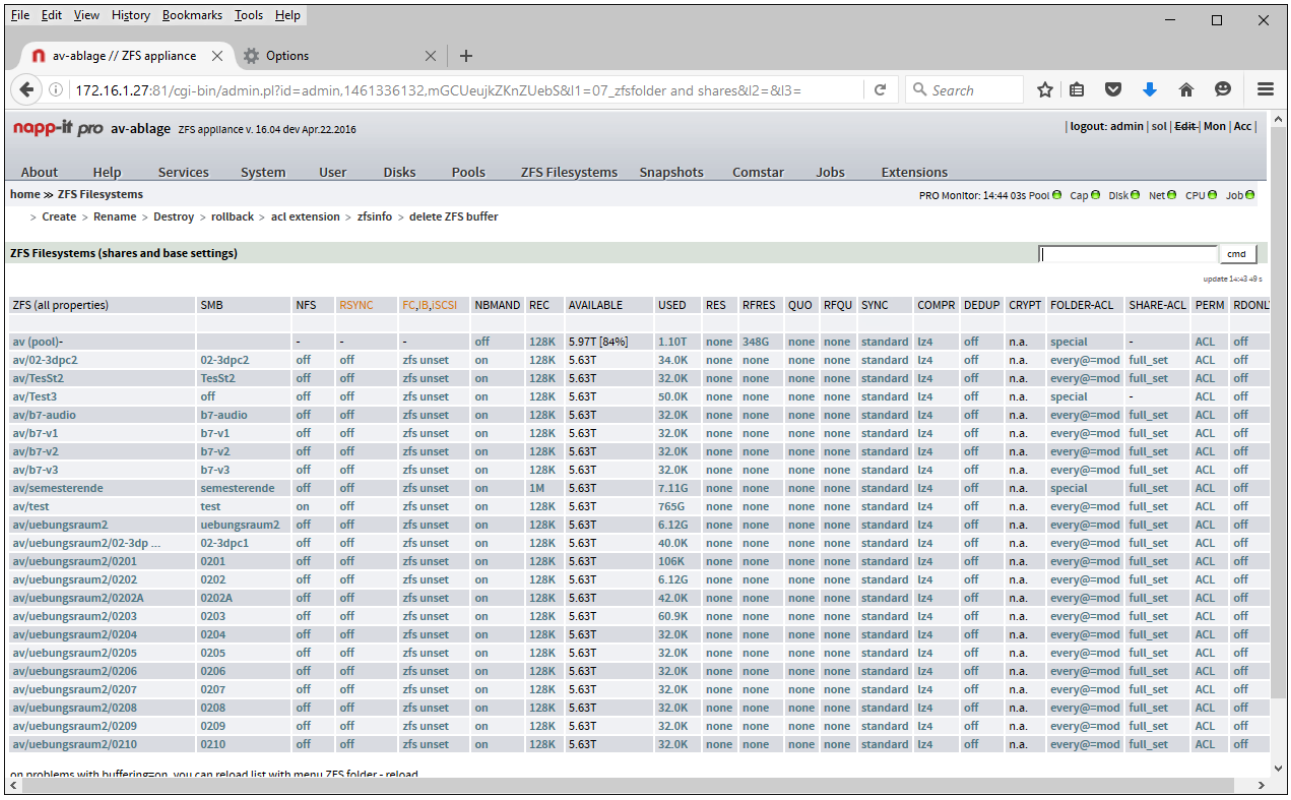
They come with a console only interface. To manage local data, you can use Midnight Commander via Putty.

Storage management is done via the napp-it Webinterface. As they include only the absolute needed software, they are the most stable option.



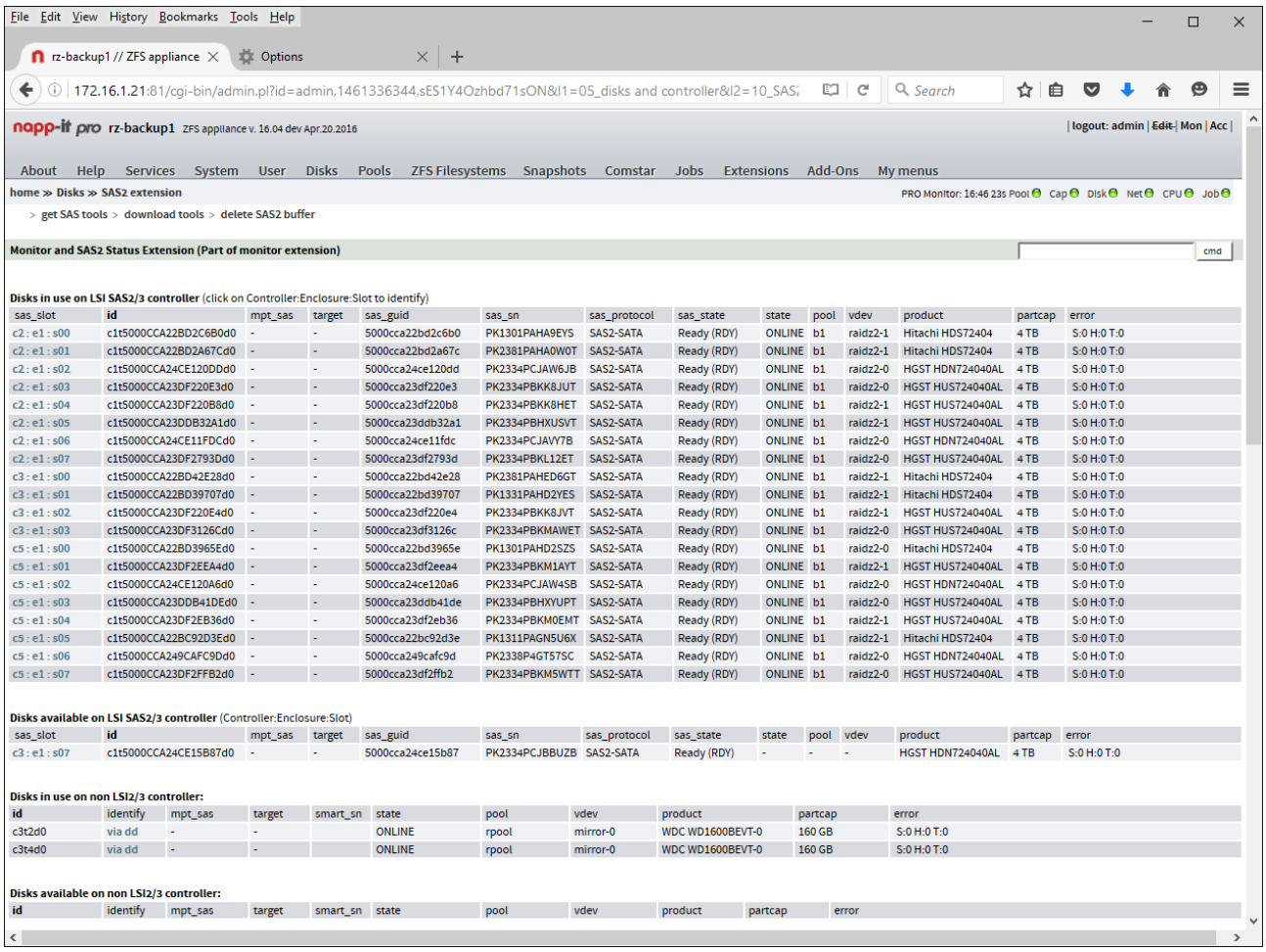
2.3. Storage-Management via napp-it

on OmniOS, OpenIndiana or Oracle Solaris



ZFS (all properties)	SMB	NFS	RSYNC	FC,IB,iSCSI	NBMAND	REC	AVAILABLE	USED	RES	RFRES	QUO	RFQU	SYNC	COMPR	DEDUP	CRYPT	FOLDER-ACL	SHARE-ACL	PERM	RDONL
av (pool)-		-	-	-	off	128K	5.97T [84%]	1.10T	none	348G	none	none	standard	lz4	off	n.a.	special	-	ACL	off
av/02-3dpc2	02-3dpc2	off	off	zfs unset	on	128K	5.63T	34.0K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/TesSt2	TesSt2	off	off	zfs unset	on	128K	5.63T	32.0K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/Test3	off	off	off	zfs unset	on	128K	5.63T	50.0K	none	none	none	none	standard	lz4	off	n.a.	special	-	ACL	off
av/b7-audio	b7-audio	off	off	zfs unset	on	128K	5.63T	32.0K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/b7-v1	b7-v1	off	off	zfs unset	on	128K	5.63T	32.0K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/b7-v2	b7-v2	off	off	zfs unset	on	128K	5.63T	32.0K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/b7-v3	b7-v3	off	off	zfs unset	on	128K	5.63T	32.0K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/semesterende	semesterende	off	off	zfs unset	on	1M	5.63T	7.11G	none	none	none	none	standard	lz4	off	n.a.	special	full_set	ACL	off
av/test	test	on	off	zfs unset	on	128K	5.63T	765G	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/uebungsraum2	uebungsraum2	off	off	zfs unset	on	128K	5.63T	6.12G	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/uebungsraum2/02-3dp ...	02-3dpc1	off	off	zfs unset	on	128K	5.63T	40.0K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/uebungsraum2/0201	0201	off	off	zfs unset	on	128K	5.63T	106K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/uebungsraum2/0202	0202	off	off	zfs unset	on	128K	5.63T	6.12G	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/uebungsraum2/0202A	0202A	off	off	zfs unset	on	128K	5.63T	42.0K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/uebungsraum2/0203	0203	off	off	zfs unset	on	128K	5.63T	60.9K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/uebungsraum2/0204	0204	off	off	zfs unset	on	128K	5.63T	32.0K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/uebungsraum2/0205	0205	off	off	zfs unset	on	128K	5.63T	32.0K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/uebungsraum2/0206	0206	off	off	zfs unset	on	128K	5.63T	32.0K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/uebungsraum2/0207	0207	off	off	zfs unset	on	128K	5.63T	32.0K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/uebungsraum2/0208	0208	off	off	zfs unset	on	128K	5.63T	32.0K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/uebungsraum2/0209	0209	off	off	zfs unset	on	128K	5.63T	32.0K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off
av/uebungsraum2/0210	0210	off	off	zfs unset	on	128K	5.63T	32.0K	none	none	none	none	standard	lz4	off	n.a.	every@=mod	full_set	ACL	off

ZFS Filesystems and disk details



Disks in use on LSI SAS2/3 controller (click on Controller:Enclosure:Slot to identify)

sas_slot	id	mpt_sas	target	sas_guid	sas_sn	sas_protocol	sas_state	state	pool	vdev	product	partcap	error
c2 : e1 : s00	c1t5000CCA22BD2C6B0d0	-	-	5000cca22bd2c6b0	PK1301PAHA9EYS	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-1	Hitachi HDS72404	4 TB	S:0 H:0 T:0
c2 : e1 : s01	c1t5000CCA22BD2A67Cd0	-	-	5000cca22bd2a67c	PK2381PAHA0W0T	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-1	Hitachi HDS72404	4 TB	S:0 H:0 T:0
c2 : e1 : s02	c1t5000CCA24CE120DDd0	-	-	5000cca24ce120dd	PK2334PCJAW6JB	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-0	HGST HDN724040AL	4 TB	S:0 H:0 T:0
c2 : e1 : s03	c1t5000CCA23DF220E3d0	-	-	5000cca23df220e3	PK2334PBKK8JUT	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-0	HGST HUS724040AL	4 TB	S:0 H:0 T:0
c2 : e1 : s04	c1t5000CCA23DF220B8d0	-	-	5000cca23df220b8	PK2334PBKK8HET	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-1	HGST HUS724040AL	4 TB	S:0 H:0 T:0
c2 : e1 : s05	c1t5000CCA23DDB32A1d0	-	-	5000cca23ddb32a1	PK2334PBHXUSVT	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-1	HGST HUS724040AL	4 TB	S:0 H:0 T:0
c2 : e1 : s06	c1t5000CCA24CE11FDCd0	-	-	5000cca24ce11fdc	PK2334PCJAVY7B	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-0	HGST HDN724040AL	4 TB	S:0 H:0 T:0
c2 : e1 : s07	c1t5000CCA23DF2793Dd0	-	-	5000cca23df2793d	PK2334PBKL12ET	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-0	HGST HUS724040AL	4 TB	S:0 H:0 T:0
c3 : e1 : s00	c1t5000CCA22BD42E28d0	-	-	5000cca22bd42e28	PK2381PAHED6GT	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-1	Hitachi HDS72404	4 TB	S:0 H:0 T:0
c3 : e1 : s01	c1t5000CCA22BD39707d0	-	-	5000cca22bd39707	PK1331PAHD2YES	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-1	Hitachi HDS72404	4 TB	S:0 H:0 T:0
c3 : e1 : s02	c1t5000CCA23DF220E4d0	-	-	5000cca23df220e4	PK2334PBKK8JVT	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-1	HGST HUS724040AL	4 TB	S:0 H:0 T:0
c3 : e1 : s03	c1t5000CCA23DF3126Cd0	-	-	5000cca23df3126c	PK2334PBKMAWET	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-0	HGST HUS724040AL	4 TB	S:0 H:0 T:0
c5 : e1 : s00	c1t5000CCA22BD3965Ed0	-	-	5000cca22bd3965e	PK1301PAHD2SZS	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-0	Hitachi HDS72404	4 TB	S:0 H:0 T:0
c5 : e1 : s01	c1t5000CCA23DF2EEA4d0	-	-	5000cca23df2eea4	PK2334PBKM1AYT	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-1	HGST HUS724040AL	4 TB	S:0 H:0 T:0
c5 : e1 : s02	c1t5000CCA24CE120A6d0	-	-	5000cca24ce120a6	PK2334PCJAW4SB	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-0	HGST HDN724040AL	4 TB	S:0 H:0 T:0
c5 : e1 : s03	c1t5000CCA23DDB41DEd0	-	-	5000cca23ddb41de	PK2334PBHXYUPT	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-0	HGST HUS724040AL	4 TB	S:0 H:0 T:0
c5 : e1 : s04	c1t5000CCA23DF2EB36d0	-	-	5000cca23df2eb36	PK2334PBKM0EMT	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-1	HGST HUS724040AL	4 TB	S:0 H:0 T:0
c5 : e1 : s05	c1t5000CCA22BC92D3Ed0	-	-	5000cca22bc92d3e	PK1311PAGN5U6X	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-1	Hitachi HDS72404	4 TB	S:0 H:0 T:0
c5 : e1 : s06	c1t5000CCA249CAFC9Dd0	-	-	5000cca249cafc9d	PK2338P4GT57SC	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-0	HGST HDN724040AL	4 TB	S:0 H:0 T:0
c5 : e1 : s07	c1t5000CCA23DF2FFB2d0	-	-	5000cca23df2ffb2	PK2334PBKM5WTT	SAS2-SATA	Ready (RDY)	ONLINE	b1	raidz2-0	HGST HUS724040AL	4 TB	S:0 H:0 T:0

Disks available on LSI SAS2/3 controller (Controller:Enclosure:Slot)

sas_slot	id	mpt_sas	target	sas_guid	sas_sn	sas_protocol	sas_state	state	pool	vdev	product	partcap	error
c3 : e1 : s07	c1t5000CCA24CE15B87d0	-	-	5000cca24ce15b87	PK2334PCJBBUZB	SAS2-SATA	Ready (RDY)	-	-	-	HGST HDN724040AL	4 TB	S:0 H:0 T:0

Disks in use on non LSI2/3 controller:

id	identify	mpt_sas	target	smart_sn	state	pool	vdev	product	partcap	error
c3t2d0	via dd	-	-		ONLINE	rpool	mirror-0	WDC WD1600BEVT-0	160 GB	S:0 H:0 T:0
c3t4d0	via dd	-	-		ONLINE	rpool	mirror-0	WDC WD1600BEVT-0	160 GB	S:0 H:0 T:0

Disks available on non LSI2/3 controller:

id	identify	mpt_sas	target	smart_sn	state	pool	vdev	product	partcap	error

3. Hardware

OmniOS runs on common desktop and serverhardware, see http://illumos.org/hcl/
You should mainly care about the network and the disk controller.

Rules for a trouble free minimal system

- use Sata/ AHCI with a 30GB Bootsdisk (60GB when using the napp-it ToGo image)
- prefer Intel nics
- use at least 4 GB ECC RAM (2 GB is the absolute minimum, more gives performance as readcache)
- use at least 8GB if you enable na--it realtime monitoring or acceleration
- enabling dedup may increase these minimums

Rules for barebone „quality storage" or „Napp-in-One"

- use server class hardware/ server chipsets and IPMI remote management
- use ECC RAM and an Sata bootdidk/ DOM with 60GB or more (SuperMicro DOM or prefer an Intel S3510)
- use a Celeron G4400 or a Xeon as they offers ECC support and vt-d (virtualized SAN)

- use Intel Nics (10 GbE preferred, ex Intel X540)
- use LSI HBA with raidless IT firmware, ex LSI 9207 that comes with IT firmware per default

- prefer SAS disks with an expander
- use SAS disks for HA solutions based on Multipath IO
- prefer Sata disks with multiple HBAs (without expander)
- prefer 24/7 enterprise disks

Rules for high capacity storage

- prefer raid Z2 vdevs with 6 or 10 disks or Z3 vdevs with 11 disks
- prefer enterprise SAS disks or 24/7 high quality Sata/ NAS drives
- with desktop disks, check reliability, ex with backplaze annual failure rates
  https://www.backblaze.com/blog/best-hard-drive-q4-2014/

Rules for High-Performance storage

- prefer Enterprise SSD with powerloss protection and build in overprovisioning
  like Intel S3500-S3710 or Samsung PM/SM 863 series - optionally use manual overprovisioning
  example with a host protected area (HPA) on new SSDs
  You can create a HPA with hdat2, http://www.hdat2.com/

Rules for fast but write secure storage like ESXi datastores

- use sync write with a dedicated high perfomance Slog device

Rules for a dedicated Slog device

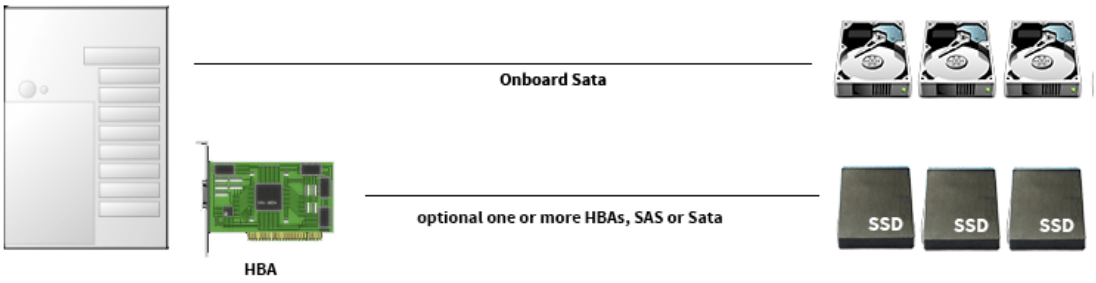
- ultra low latency
- continous high write iops, even under load
- powerloss protection
- should be noticable faster than your pool like Intel S3700 with a disk pool or a ZeusRAM with an SSD pool

- examples: ZeusRAM, a Dram based device (best of all) or Intel S3700/ 3710

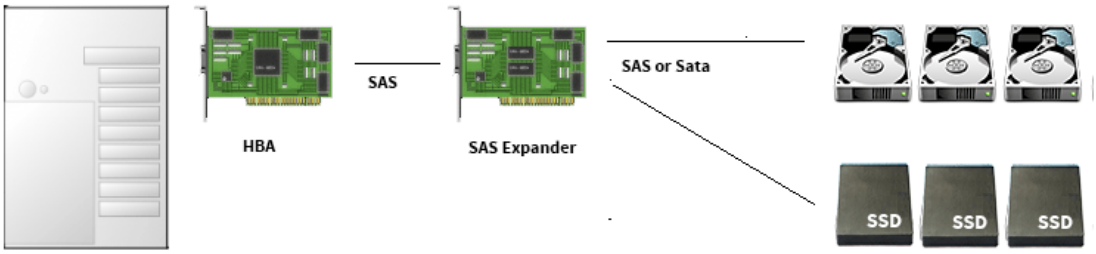
more: http://napp-it.org >> Sample Configs

3.1 ZFS Configurations

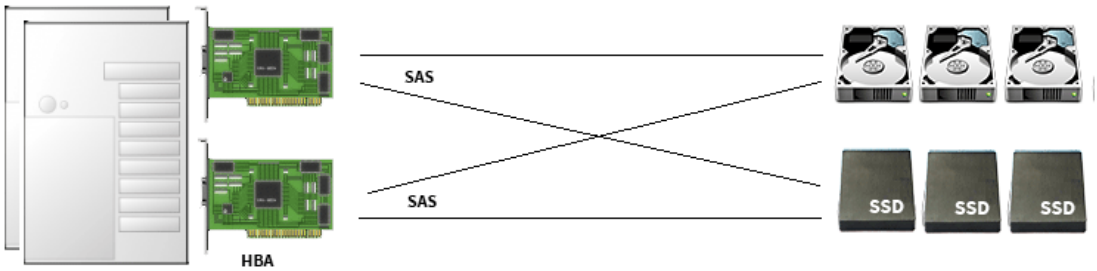
Basic napp-it ZFS server with onboard Sata (AHCI) or one or more SAS HBAs



Basic napp-it ZFS server with SAS HBA + Expander

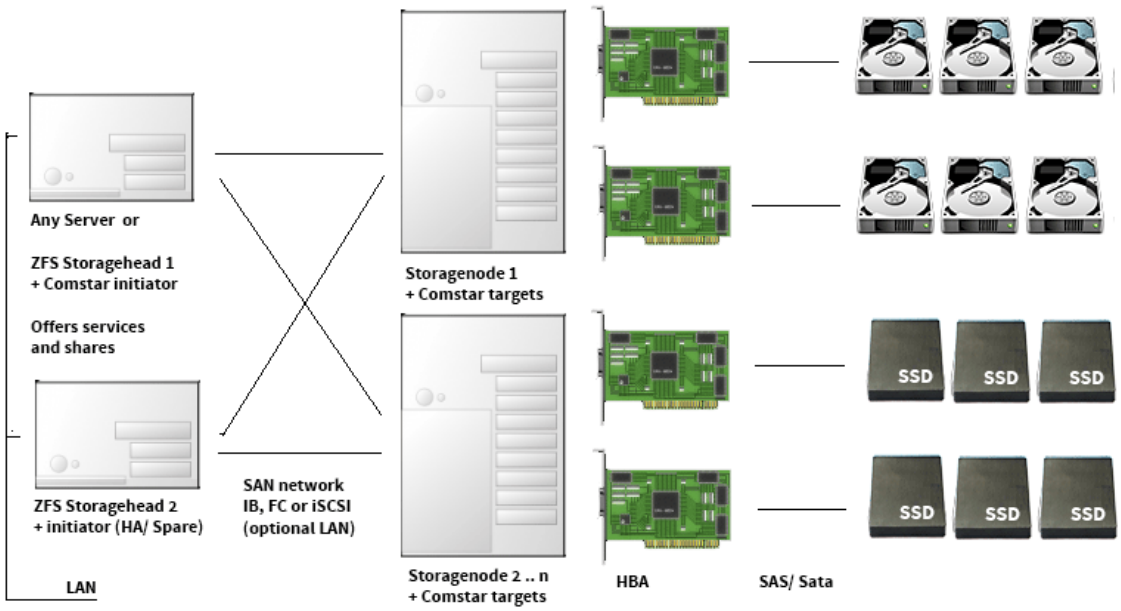


One or two napp-it ZFS server with SAS Multipath (HA over SAS disks)



use:
disks for high capacity and
SSDs for high performance

Napp-it storagehead with napp-it storagenodes



Storage availability options with Comstar targets on storageheads and/or storagenodes:
- Mirror over Comstar targets on storagenodes at OS level (any Server/OS ex ESXi, Linux, OSX, Windows or Solaris/ OmniOS) or reshared node targets (a complete ZFS node server treated as a disk) via Comstar initiators on a storagehead

Service availability options (File-, Mail-, Webservices etc)
- Manual bootup a preconfigured spare storagehead (KISS, simple)
- HA on ZFS/storagehead level (http://www.high-availability.com/zfs-ha-plugin/)
- HA based on ESXi HA functions for redundant storage or VM failover

4. napp-it ToGo deployment and disaster recovery)

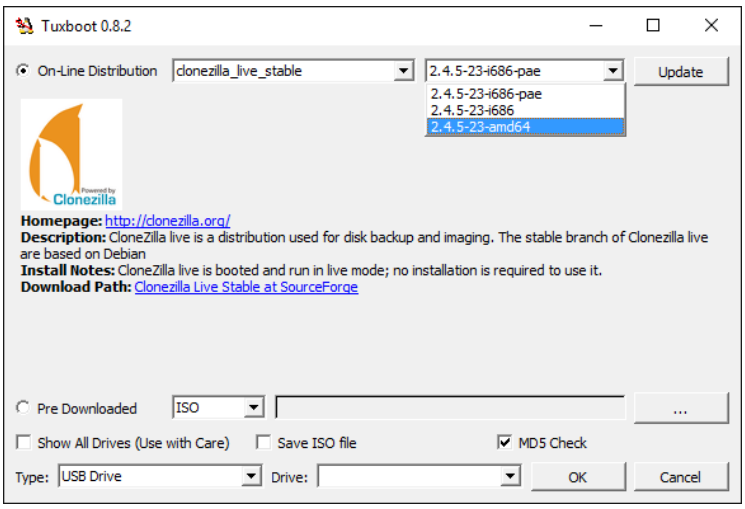
There are two options, one for a barebone setup and one for a virtualized ESXi setup

4.1 ready to use barebone napp-it ToGo distribution for an Sata bootdisk

This is the easiest and suggested setup method of installing, backup and recover a napp-it barebone appliance based on a preconfigured image for a ready to use ZFS storage server. It works best for OEM use with known hardware. It may not work with any hardware or disk. On success, please send a mail to community@napp-it.org with your mainboard, interface, disk type and remark. I will create then a list of known to work setups.

Download and clone the 512n or 512e image. For best reliability use an Intel S3510 - 80 GB with powerloss protection (512e). You may use a 64 GB SuperMicro Sata DOM or a 60GB+ Sata disk or SSD (mostly 512n). Use an Sata port for your bootdisk. As Clonezilla currently is not ZFS aware it clones the whole disk via dd.

Create a bootable USB stick (256MB is ok) with Clonezilla, a free cloning tool.
You can use a 16/32GB USB FAT-32 stick (depend on size), USB/Sata disk or share for the systemimage.



Download tuxboot.exe from
https://sourceforge.net/projects/tuxboot/files/

This is a tool to create a bootable USB stick with Clonezilla live on it. Start tuxboot on a Windows machine, select Update and an amd64 clonezilla stable or testing edition and clone it to the USB stick

For deployment or disaster recovery, you can create an image form your bootdisk with Clonezilla and restore it to an identical disk.

You need:
- a bootable USB stick (256MB+) with CloneZilla
- a suitable napp-it image on an 16/32 GB USB stick, disk or network share.
  If you use an USB stick, size depends on the size of the systemimage
  The image will work with many mainboards/systems but you must use the same bootdisk

A restore of the image is quite simple:
Boot with the clonezilla USB stick, select restore disk, select the image from a second stick/disk or a share and restore the systemdisk
The default image configures all nics automatically for DHCP.. Root pw is unset.

Another method for disaster recovery and system deployment is (this works online, does not require a shutdown)
- replicate the bootenvironment to a removable disk with a zpool on it (or a file as target)
- install a minimal default OmniOS form an USB stick
- restore this bootenvironment
- boot it

napp-it supports bootenvironments as source and target for replications

Create a master image for CloneZilla or BE deployment

1. Install OmniOS, OpenIndiana or Oracle Solaris to rpool (do not use another name)
   Download the ISO or USB image and burn a bootable CD/DVD or clone the USB stick
   Boot the installer and run a default setup. You can set network and base settings during setup or:

2. enable compress
```
set compress=lz4 rpool
set atime=off rpool
```

3. set network manually
```
dladm show-link
ipadm create-if e1000g0
```

3.1 use DHCP (preferred method on initial setup)
```
ipadm create-addr -T dhcp e1000g0/dhcp
```

3.2 add nameserver
```
echo ‚nameserver 8.8.8.8‘ >> /etc/resolv.conf
```

3.3 use dns (copy over DNS template)
```
cp /etc/nsswitch.dns /etc/nsswitch.conf
```

If something happens (typo error), retry, opt. delete interface ex ipadm delete-if e1000g0

4 Setup the napp-it appliance
```
wget -O - www.napp-it.org/nappit | perl
```

5. reset root root (Unix and SMB), no password (press enter twice)
```
passwd root
```

6. enable SSH root access via napp-it menu Services > SSH

7. Login via Putty (to copy/paste commands with a mouse right-click)

8. set TLS for encrypted mai ex Gmail (run the from Putty/ Console and confirn any questions with enter)
```
perl -MCPAN -e shell
notest install Net::SSLeay
notest install IO::Socket::SSL
notest install Net::SMTP::TLS
exit;
```

9. disable services SSH root access and sendmail service in napp-it (menu Service)
   set netbios_enable=true (Windows network browsing in OmniOS 151016 and up, menu Service>SMB>prop.)

10. set basic tunings in napp-it System > Tuning,

11. delete all BE

12. if you want to use this setup as template, run
```
perl /var/web-gui/data/tools/other/prepare_image.pl
```

13 beadm create backup.setup (console, creates a backup bootenvinment with nics not configured)
14. Enter halt and power off, start cloneZilla
15. clone the systemdisk with defaults (quite fast, medium compress). name like date-img18-board-disk
or select advanced settings with z2p compress , compromise between compression time and size

5.1 Manual setup of a napp-it storage appliance with DHCP

Download the OmniOS or Solaris installer (ISO dvd or USB installer), boot the installer and install the OS.
Select UTC timezone, with bloody versions keep all defaults incl keyboard
After OS setup, your need to configure your network either with DHCP (5.1) or manually (5.2).

5.1.1 Initial setup of the napp-it appliance with DHCP

- boot OmniOS and login as root (no password)

- list available network adapters with their linkname (ex e1000g0):

```
dladm show-link
```

- create an ip interface based on the linkname

```
ipadm create-if e1000g0
```

- enable DHCP (requires a DHCP server)

```
ipadm create-addr -T dhcp e1000g0/dhcp
```

- add nameserver

```
echo ‘nameserver 8.8.8.8‘ >> /etc/resolv.conf
```

- use DNS name resolution (copy over DNS template)

```
cp /etc/nsswitch.dns /etc/nsswitch.conf
```

- install napp-it online (default/ free version)

```
wget -O - www.napp-it.org/nappit | perl
```

- reboot (or set current bootenvironment as default)

```
reboot
```

- set a root password (this password is valid for Unix shell logins and SMB connects)

```
passwd root
passwd napp-it  (napp-it user, account is disabled but sudo requires a pw)
```

optional: check current network settings

```
ifconfig -a
```

optional: If something happens (typo error), delete interface and retry

```
ipadm delete-if e1000g0

optional for ESXi: pkg install open-vm-tools, pkg install vmxnet3s
export template with dhcp, without pass-through devices or cd isos

optional with newest OmniOS: svcadm disable sendmail and sendmail-client
(https://blogs.oracle.com/souvik/entry/my_unqualified_host_name_sleeping)
```

5.2 Setup napp-it storage appliance with manual ip settings

After OS setup, your need to configure your network either with DHCP (5.1) or manually (5.2).

5.2.1 Initial setup of the napp-it appliance with manual ip settings

If you have a working DHCP server: use DHCP and set network manually later in napp-it

- boot OmniOS and login as root (no password)

- list available network adapters with their linkname (ex e1000g0):

```
dladm show-link
```

- create an ip interface based on the linkname

```
ipadm create-if e1000g0
```

- set a manual ip address

```
ipadm create-addr -T static -a 192.168.0.1/24 e1000g0/v4
```

- add default route (enter your router ip)

```
route -p add default 192.168.0.254
```

- add nameserver

```
echo ‘nameserver 8.8.8.8‘ >> /etc/resolv.conf
```

- use DNS name resolution (copy over DNS template)

```
cp /etc/nsswitch.dns /etc/nsswitch.conf
```

- install napp-it 0.9 online

```
wget -O - www.napp-it.org/nappit | perl
```

- reboot (or set current bootenvironment as default)

```
reboot
```

- set a root password (this password is valid for Unix shell logins and SMB connects)

```
passwd root
```

```
passwd napp-it  (napp-it user, account is disabled but sudo requires a pw)
```

optional: check current network settings

```
ifconfig -a
```

optional: If something happens (typo error), delete interface and retry

```
ipadm delete-if e1000g0
```

5.3 stop/ remove/ manual deinstall of napp-it

Napp-it installer creates a bootenvironment with the state prior the installation so you can always go back.

As napp-it is a pure copy and run installation, it copies everything to /var/web-gui If you delete this folder, the init file /etc/init.d/napp-it, the user nappit and an entry in /etc/sudoers and user_attr you have wiped it beside tools that are installed during setup like smartmontools, midnight commander, iperf or netcat. You do not need napp-it for regular storage operations. If you want to stop napp-it, use /etc/init.d/napp-it stop (start | restart)

If you install add-ons like the AMP stack, they are using pkgsrc from Joyent with files in /opt

6. ZFS manuals and infos

You should now download and optionally printout some basic manuals.

6.1 napp-it manuals

http://www.napp-it.org/doc/downloads/napp-it.pdf
http://www.napp-it.org/doc/downloads/napp-in-one.pdf
http://www.napp-it.org/doc/downloads/advanced_user.pdf

http://www.napp-it.org/doc/manuals/flash_x9srh-7tf_it.pdf

6.2 manuals for Oracle Solaris 11

Download and print out needed manuals from

https://docs.oracle.com/en/operating-systems/

6.3 manuals for Oracle Solaris 11 Express and OmniOS

Download and print out needed manuals. As Oracle offers only manuals for their current release, you must use an archive search for Solaris 11 Express manuals (OmniOS is more or less a fork of Solaris 11 Express)

http://archive.is/snZaS

The archive.is page refers to the old Solaris Express 11 page. If you click on a description, you are forwarded to the current Solaris 11 page. If you click on „Download" you get the correct manual.

Download manuals and print out at least the ZFS administration guides

6.4 other books and manuals, Less known Solaris features

http://www.c0t0d0s0.org/pages/lksfbook.html
http://www.c0t0d0s0.org/archives/6639-Recommended-SolarisSun-Books.html

6.5 Maillists, forums and IRC

Join the following maillists, threads and IRC discussions to keep you informed

http://lists.omniti.com/mailman/listinfo/omnios-discuss
http://echelog.com/logs/browse/illumos/
http://echelog.com/logs/browse/omnios

http://hardforum.com//showthread.php?t=1573272 (Hardforum)
https://forums.servethehome.com (Solaris/napp-it subforum)
http://www.hardwareluxx.de/community/f101/zfs-stammtisch-570052.html (DE)

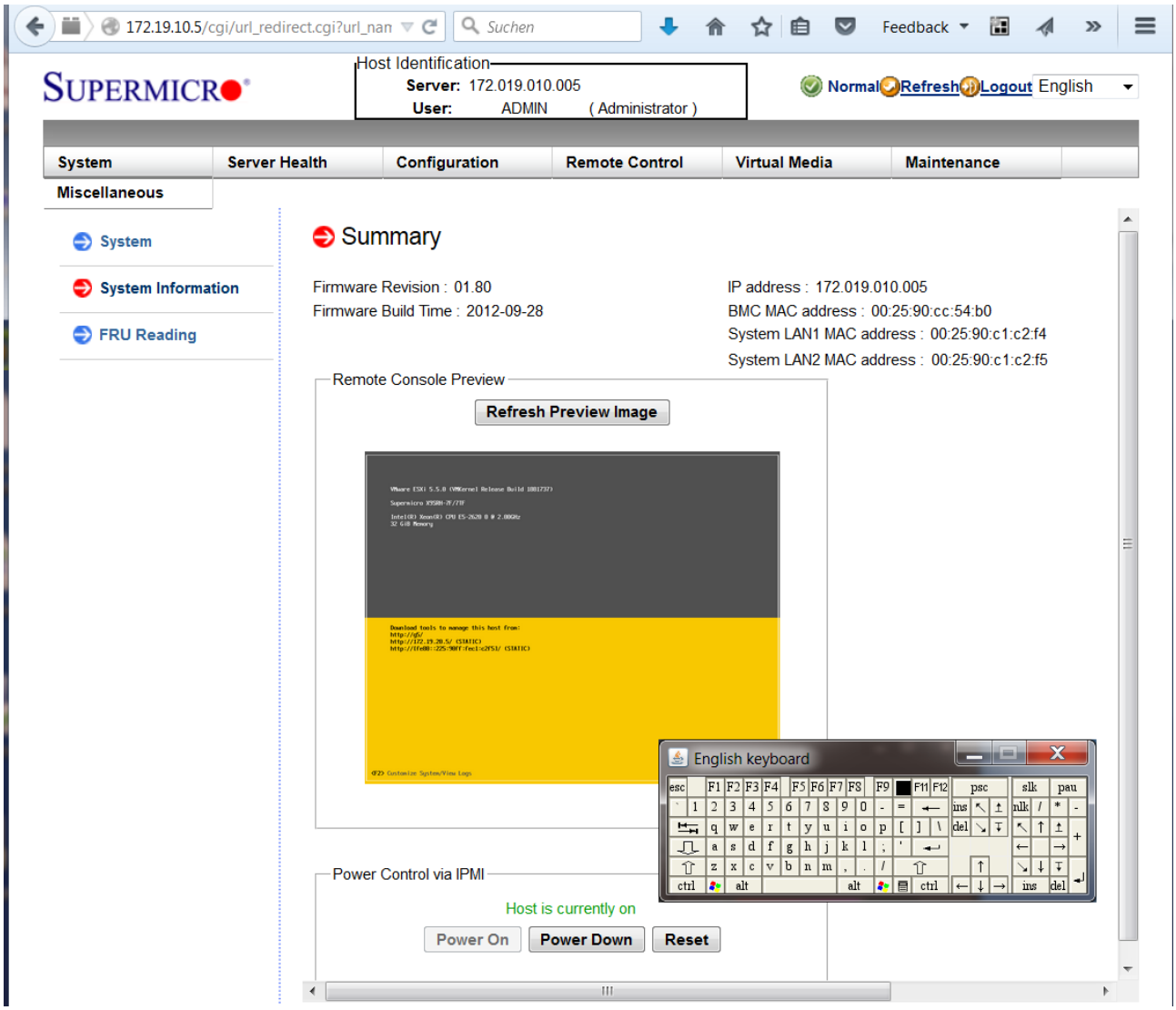
7. Remote management

A Server can be managed remotely, use these tools

7.1 IPMI

IPMI ist a must have for a server: https://en.wikipedia.org/wiki/Intelligent_Platform_Management_Interface

IPMI is a remote management microcontroller on serverclass hardware like Supermicro mainboards that ends with a „-F„ . You can connect the microcontroller remotely with a webbrowser even when the server is in a power-off state. Functions are mainly power on/off/reset, a remote console/keyboard and the ability to mount ISOs like a lokal CD/DVD drive.



IPMI window (Java applet) with a virtual keyboard and a console preview that can be displayed full size.

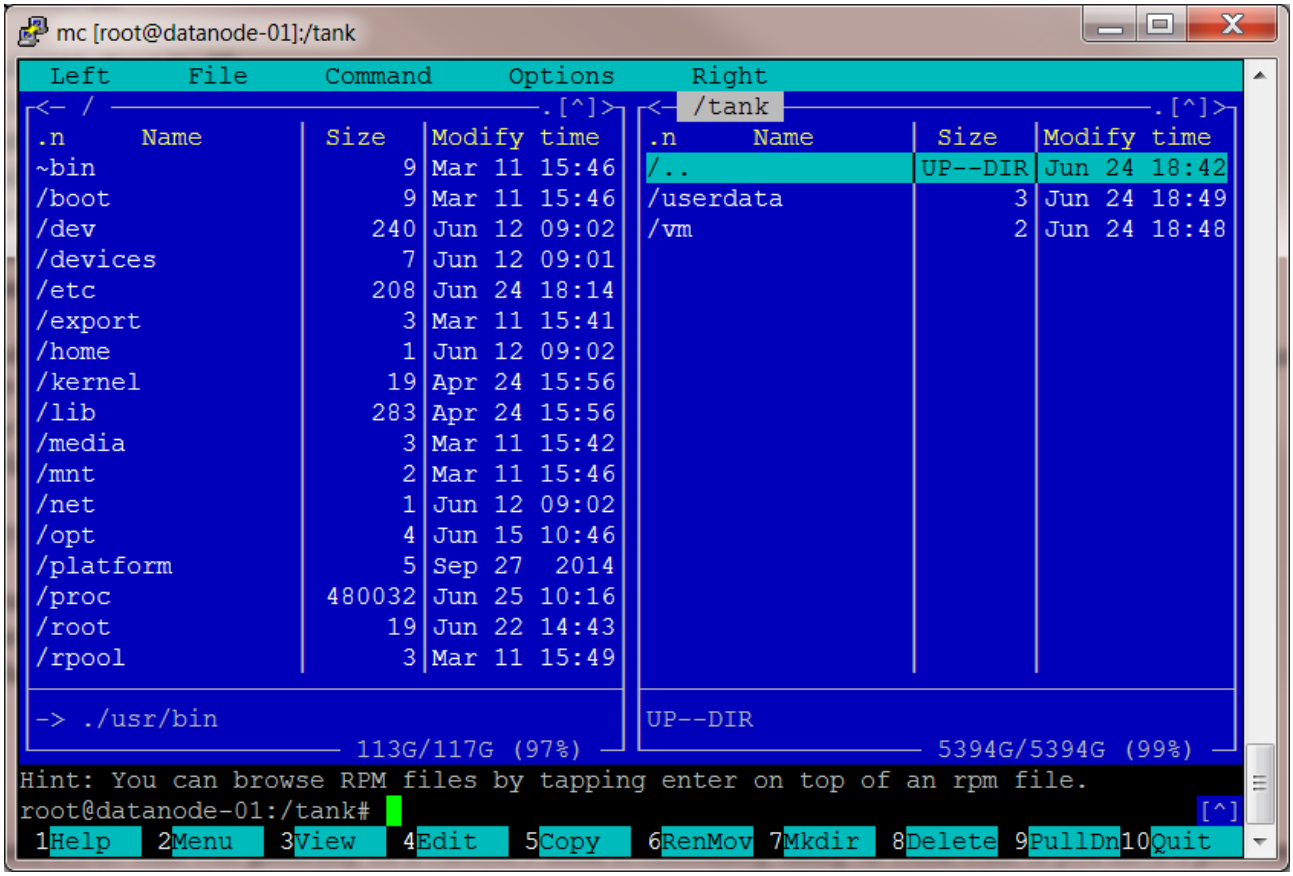
You can enable IPMI and its ip adress in your mainboard bios. It comes with a dedicated network port so you can connect with a dedicated and isolated management network. As an option, you can use your regular Lan port (insecure). IPMI requires a current Java (free download from www.java.com). For security reasons, you must allow the ip of your server (ex https://172.19.10.5) for java applets.

SuperMicro default IPMI user/pw (you should change that)

user: ADMIN
pw: ADMIN

7.2 Remote Console via Putty

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html



Putty is a „must have" tool.

Daily storage management is done via the napp-it Web-Interface. Some tasks require console access. This can be done locally or remotely via Putty, a free Windows application. Download and run - no installation required.

To use Putty, you must enable SSH on OmniOS. This is the case per default but per default only regular users can login, not root. So you must either create a regular user than can login. After this you can gain admin permissions wit a su command. Other option is to enable remote root access in the napp-it Web-GUI in menu „Services >> SSH >> allow root". As this can be a security problem, you should disable remote root afterwards with menu „Services >> SSH >> deny root"

Tips:

You can copy/ paste CLI commands with a „right mouse click" into the Putty Window.
The same is the case when you mark text within the Putty console.

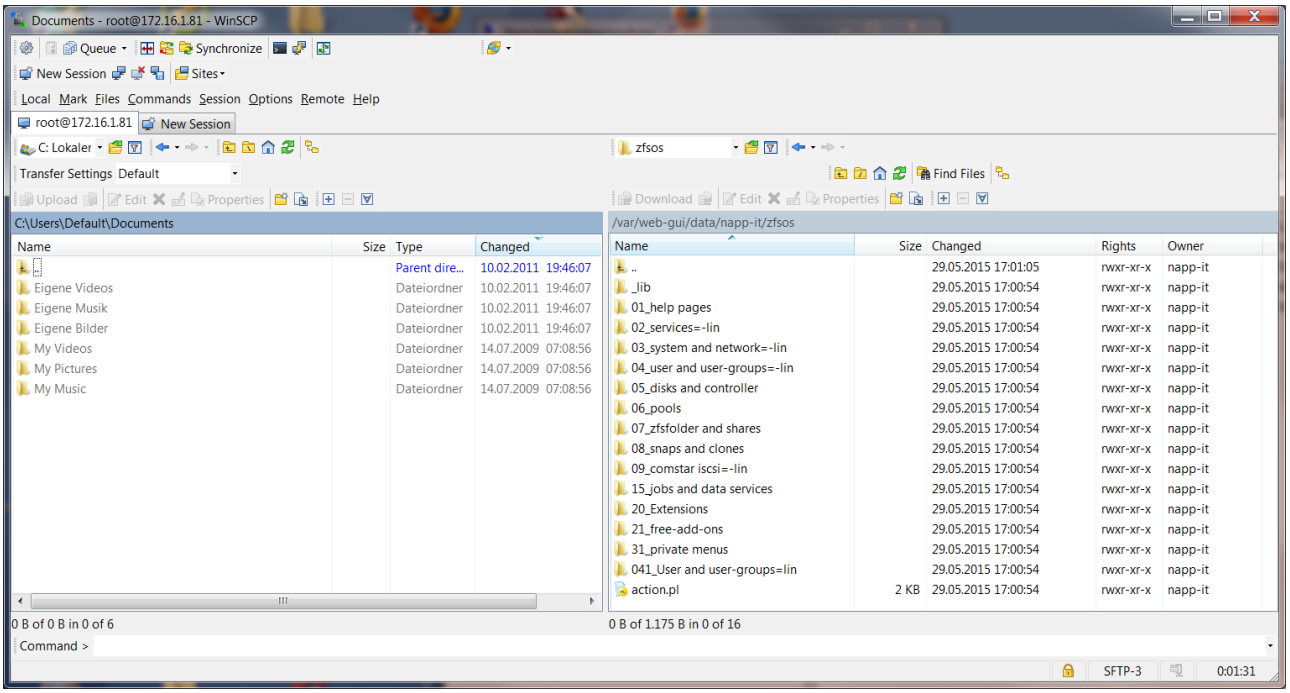
Midnight commander, a console filebrowser that runs on OmniOS with an optional usermenu is installed automatically by napp-it to do local file management (copy/move/delete/edit).
This is the fastest way to copy/move files as it is done locally and not over your network.

You can start midnight commander when you enter „mc" at console and quit with F10.

If Midnight commander is showing wrong characters, you can either set a proper environment variable or call Midnight commander directly with a LANG envronment ex (German), start mc like:
LANG=de_DE mc or change Putty language settings (Try UTF-8 or ISO 8859 Latin-1)
http://www.andremiller.net/content/getting-midnight-commander-line-drawing-work-putty)

7.3 Remote Filemanagement/ Fileediting on Windows with WinSCP

http://winscp.net/eng/download.php



WinSCP is a „must have“ tool.

WinSCP is a free Windows application that allows

- upload/download files (binary/text/auto) like a ftp client but encrypted
- edit /find files on your server (you can use different editors for differen filetypes)
- delete/copy/move files (not as fast as Midnight Commander as files must be transferred encrypted over LAN)
- check/modify Unix permissions and ownership

To use WinSCP you must enable SSH on OmniOS. This is the case per default but per default only allows that regular users can login, not root. An option is to enable remote root access in the napp-it Web-GUI in menu „Services >> SSH >> allow root“. As this can be a security problem, you should disable remote root afterwards with menu „Services >> SSH >> deny root“ .

Tips:

When you connect as root, you have full permissions to edit all files on OmniOS including systemfiles. This makes Unix magagement a lot easier as you can manage remotely from Windows and do not need to use ancient editors like vi.

8. First steps with the napp-it Web-Gui

Use your browser to manage napp-it: http://serverip:81 example

http://192.168.1.1:81

If you are unsure about your ip, enter the following console command

ifconfig -a

If you start napp-it the first time, you are asked to setup napp-it passwords and email.

Setup options (Menu About >> Settings):

User accounts: Admin and oprtator

These are napp-it only accounts and are used only for web-management and appliance grouping

default-user

User operator has a reduced set of administration options

Menu and Language

You can select a menu-set that allows different menu descriptions and translatiions
You can also restrict menu items:

Default set is Solaris (sol) that offers only Storage options supported by Solaris or OmniOS
You should use this menu set in production environments.

As Solaris is an enterprise OS, you can use it for other services or applications as well example a Webserver or Databese Server. You can use these services without support

Napp-it offers some menus that may help to manage them like the Apache webserver
You will find according menus for this example under Services > Apache with the options to edit the config files, includes, modules or related like php condig files.
To enable these unsupported menus, you must select another menuset, example en or de.

You can force a special menu set if you rename /_my/zfsos/_lib/lang/MY to /_my/zfsos/_lib/lang/MY!
You must place a menu description file about_menus.txt in this folder.

email

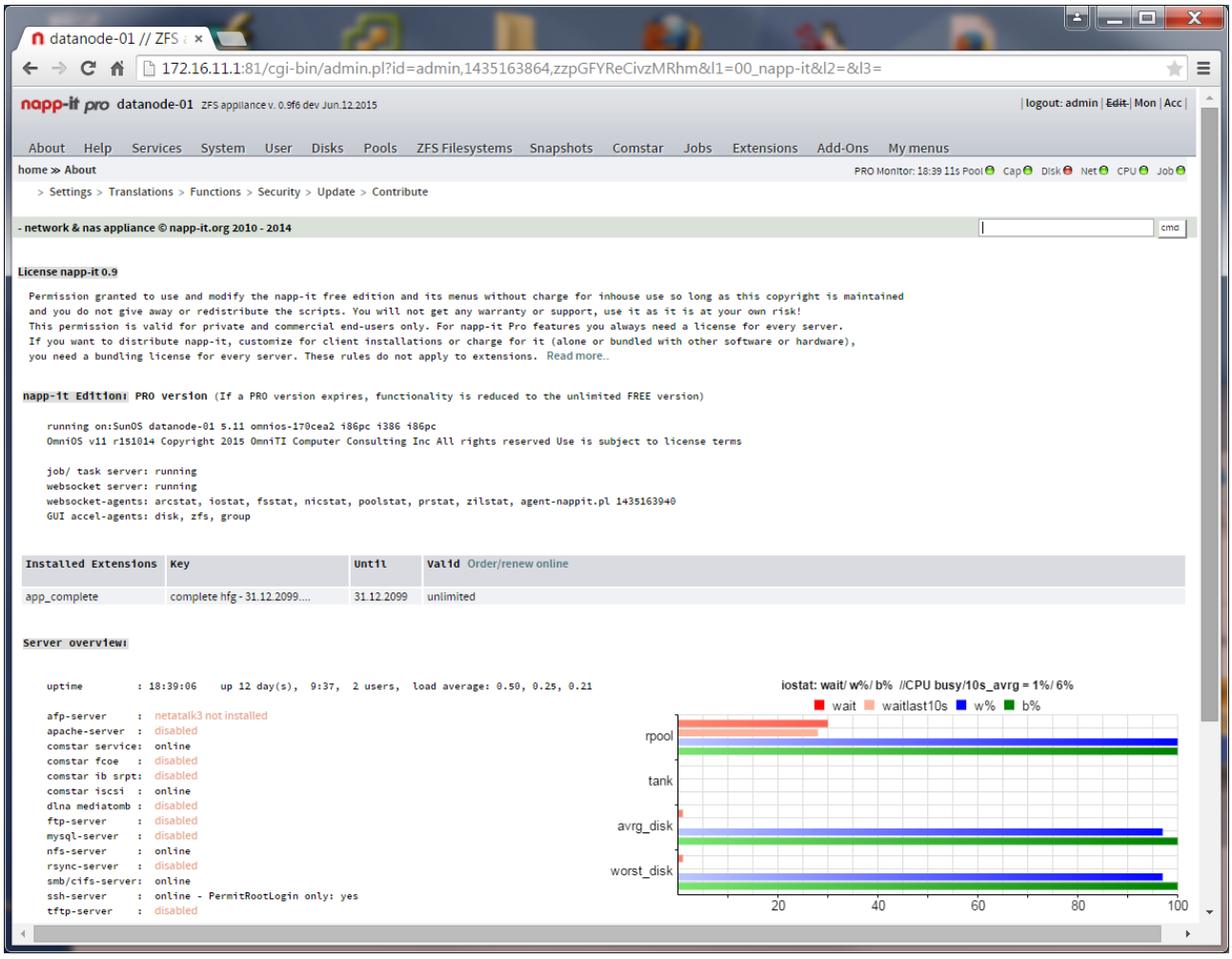
Napp-it can send alert and status emaile, either unencrypted or TLS encrypted
Enter the account details about your mailsetup.

Be aware: napp-it must store your cleartext password

push

Push alerts is an option to send alerts to your desktop or smartphone.

Welcome to the napp-it Web-Gui



Napp-it welcomes you with the above startscreen that shows basic infos about your OS/ napp-it release, the state of services and your current iostat. After an initial setup, napp-it comes with a 30 day evaluation of Pro features. This includes realtime monitoring of single appliances and appliance groups, WWN enclosure management, remote replication, advanced ACL management and an improved GUI performance due background acceleration agents.

Napp-it Pro includes support options like access to developer or bugfix releases or email-support for the complete edition. After the 30 day evaluation, you can continue to use napp-it free without a time or capacity restriction -even commercially- with all features that are needed to manage a ZFS storage appliance.

Napp-it free is not crippleware or a product that is limited in essential features. It is sufficient for many cases. It is a stable state of napp-it that is updated from the dev release from time to time. If you want to support napp-it or use Pro features or require immediate access to bugfix releases commercially or as a homeoffer, check http://napp-it.org/extensions/quotation_en.html

There are two menus

1. the regular menus like About, Services, System, Disk, ..
You use them for Storage management.

2. The top level menu with Logout, Edit, Menu-set, Mon and Acc
The menu-set allows to switch between menu sets (you can select in About> settings) of force a MY! menu. Edit allows to control execution of menus, view menu sources or display internal hash-data that are processed by a menu. Mon enables/ disables realtime graphs and Acc enables/ disables background agents in napp-it Pro.

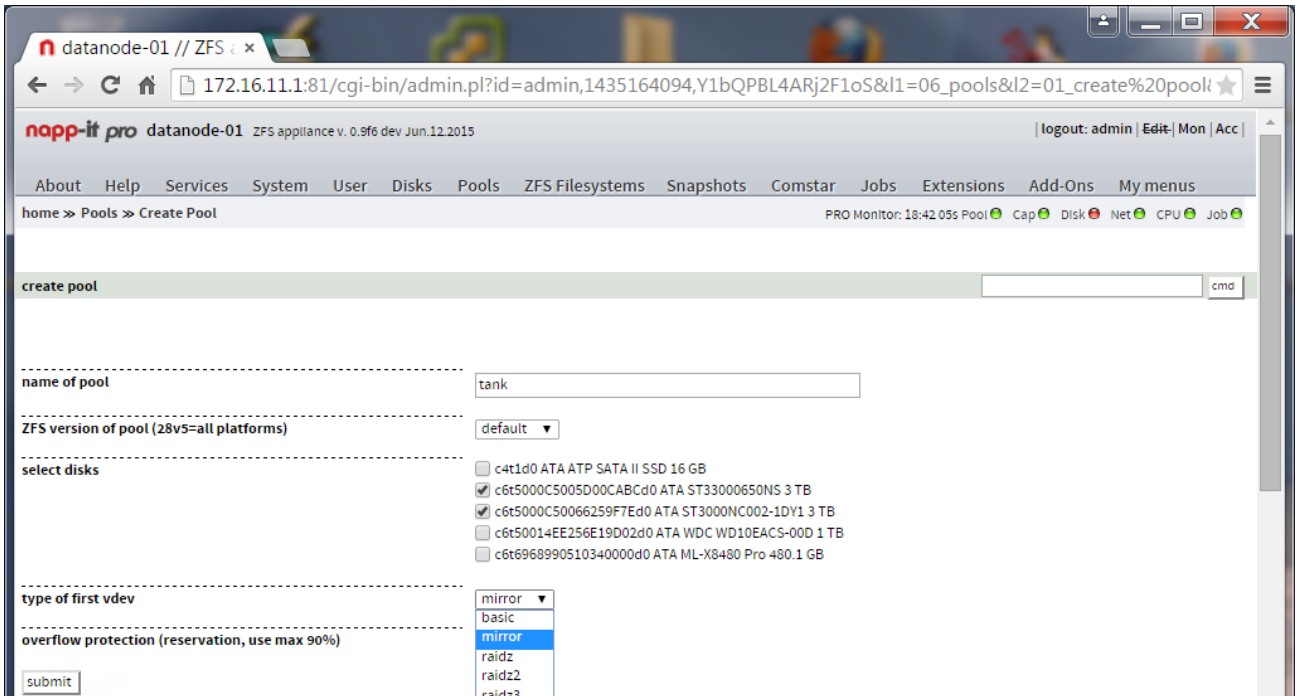
9. Create a ZFS datapool

From other systems like Windows, you know disks where you can create one or more partitions with a fixed size. You can combine single disks or partitions to a Raid that is treated like a single disk. It is possible to increase a partition up to disk or Raid-array size. But you cannot span a partition afterwards over multiple disks or raid-arrays without destroying the old partition. If a disk is full, you must create a new one and copy over data.

ZFS allows a more flexible handling of disc capacity with a concept that is called storage virtualization. Base of this is a storage pool. Unlike a disk or conventional raid array, the size of a ZFS pool can grow dynamically. You start with a new pool that is build from a Raid-Array example a ZFS Raid-1 or Raid-Z without Raid problems like the write hole problem of conventional Raid. If you need more capacity, you add more Raid-Arrays to build storage up to the Petabyte range as this is the real design goal of ZFS.

Similar to oldstyle partitions, you create ZFS filesystems on your pool but unlike old partitions, you do not set a size of a filesystem as it can grow dynamically up to the poolsize. If you increase the pool, the additional capacity is immediately available to all filesystems. You can limit capacity with quotas and ensure with reservations.

create a ZFS pool with menu Pools >> create Pool



- name your pool (ex tank)
- select version (only needed for compatibility mainly with Oracle Solaris and pool v28/5)
- select disks that you want to use for your first raid-array/ vdev (ex two 3 TB disks in a mirror)
- enable overflow protection (a 10% pool reservation that limits usable capacity to prevent a full/slow pool)
  You can reduce/ delete the reservation in menu ZFS Filesystems at any time

click submit and your pool „tank„ is created. Details about the pool: see menu Pools

Pool	VER	RAW SIZE/ USABLE	ALLOC	RES	FRES	AVAIL zfs [df -h/df -H]	DEDUP	FAILM	EXP	REPL	ALT	GUID	HEALTH	SYNC	ENCRYPT	ACTION	ATIME
rpool	-	149G/ **143.9GB**	26.8G	-	-	**113G** [114G /122G]	1.00x	wait	off	off	-	162800115664189026	ONLINE	standard	n.a.	clear errors	off
tank	-	5.44T/ **5.8TB**	427K	-	543G	**5.27T** [5.3T /5.8T]	1.00x	wait	off	off	-	16273592907840458286	ONLINE	standard	n.a.	clear errors	-

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

click on poolname to list all pool-properties or on a property to modify.

Extend a ZFS datapool

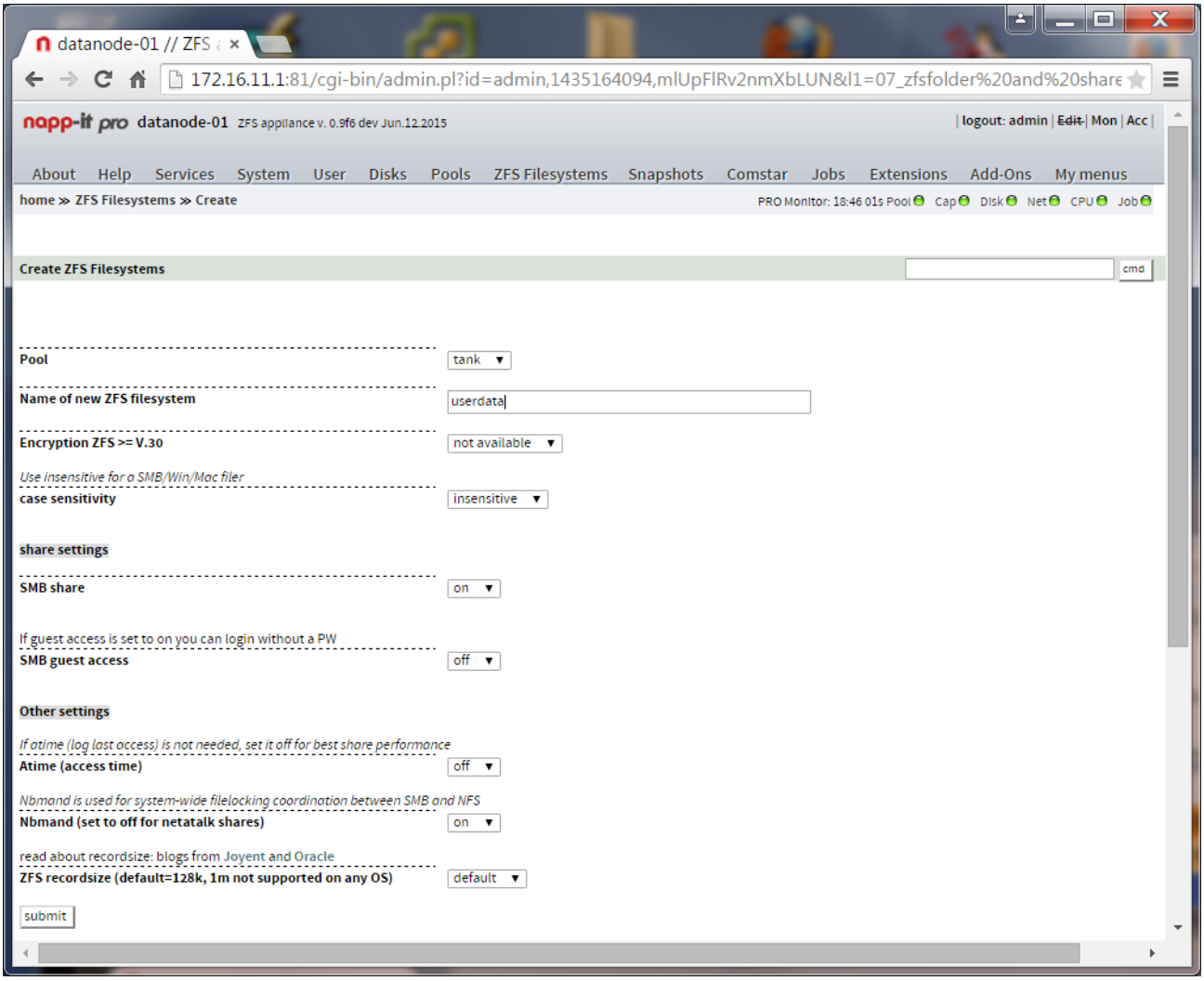
If you want to increase capacity, use menu Pools >> extend pool and add another Raid-array (ZFS call it vdev) ex a mirror or raid-Z. ZFS will stripe data over all vdevs to increase not only capacity but performance.

10. Create a ZFS filesystem

From other systems like Windows, you know partitions that you can format in FAT32 or NTFS. This is similar to OmniOS and ZFS with the difference, that you always format to ZFS and that the size of a filesystem can grow dynamically up to poolsize. You can limit the available capacity of a filesystem with quotas and ensure with reservations. This is called storage virtualization.

Basically it is enough to create a single filesystem and use traditional folders below to organize your data. But as every filesystem can have different ZFS properties, can be replicated and has its own snapshots, it is common to use as many filesystems as you like, up to thousands (example one filesystem per user).

create a ZFS pool with menu ZFS Filesystems >> create

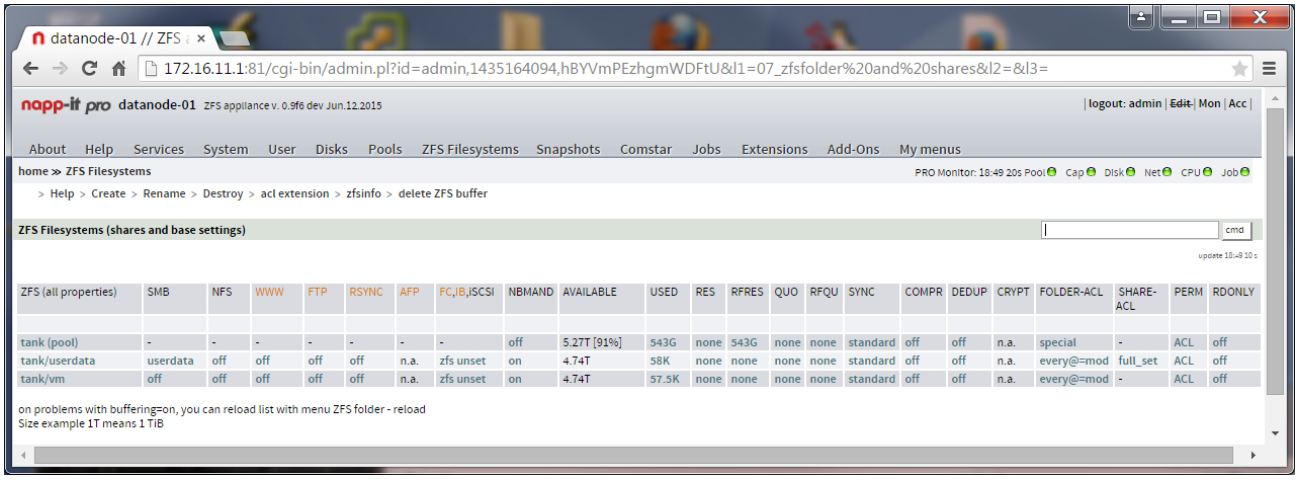


- select your pool (ex tank)
- name your new filesystem (ex userdata)
- select case sensitivity (Unix is case sensitive, Windows not - for a SMB server use the „Windows-behaviour")
- set immediate SMB sharing on or off
- other settings like atime, nbmand and recordsize

click submit and you have created a filesystem, optionally with SMB sharing enabled. You can now connect from Windows as user root as you do not have created other users yet. Default permission is everyone=modify.

Create more filesystems ex vm when needed (ESXi datastore)

Menu ZFS filesystems



Most settings about share and filesystem properties are ZFS filesystem properties that can can be set/controlled in this menu. You can click on an editable setting (they are blue coloured) to modify. Examples:

enable/ disable a SMB share:
click in the row of a filesystem example tank/userdata to the entry under the column SMB

enable/ disable a NFS share
click in the row of a filesystem example tank/vm to the entry under the column NFS

enable/ disable a iSCSI share (a ZFS volume as a blockdevice)
click in the row of a filesystem example tank/vm to the entry under the column iSCSI

set a quota for a filesystem
click in the row of a filesystem example tank/userdata to the entry under the column QUO or RFQU

set a reservation for a filesystem
click in the row of a filesystem example tank/userdata to the entry under the column RES or RFRES

enable sync write for a filesystem
click in the row of a filesystem example tank/userdata to the entry under the column SYNC
Sync write setting affects data security. Off means fast cached writes but last 5s are lost on a powerloss.

enable LZ4 compress for a filesystem
click in the row of a filesystem example tank/userdata to the entry under the column COMPR

enable dedup for a filesystem
click in the row of a filesystem example tank/userdata to the entry under the column DEDUP
Warning: dedup works poolwide. With low RAM this can dramatically reduce performance.

set/reset ACL for files and folders
click in the row of a filesystem example tank/userdata to the entry under the column Folder-ACL
Reset ACL is free. Other features are part of an extension. You can set ACL via Windows in napp-it free.

list all ZFS filesystem properties
click on the filesystem name ex tank/userdata

11. Solarish SMB/ CIFS Server

SMB/CIFS is a filesharing protocol from the Windows world. It is widely used on any platform.
Even Apple switched to SMB in their newer OSX releases as the default sharing protocol.

On OmniOS/ OpenIndiana/ Solaris (Solarish) you have two options for an SMB server. One is SAMBA that is available on any Linux/ Unix system. The other is the Solaris hCIFS server that is available on Solaris based systems only and is the de facto standard SMB server there.

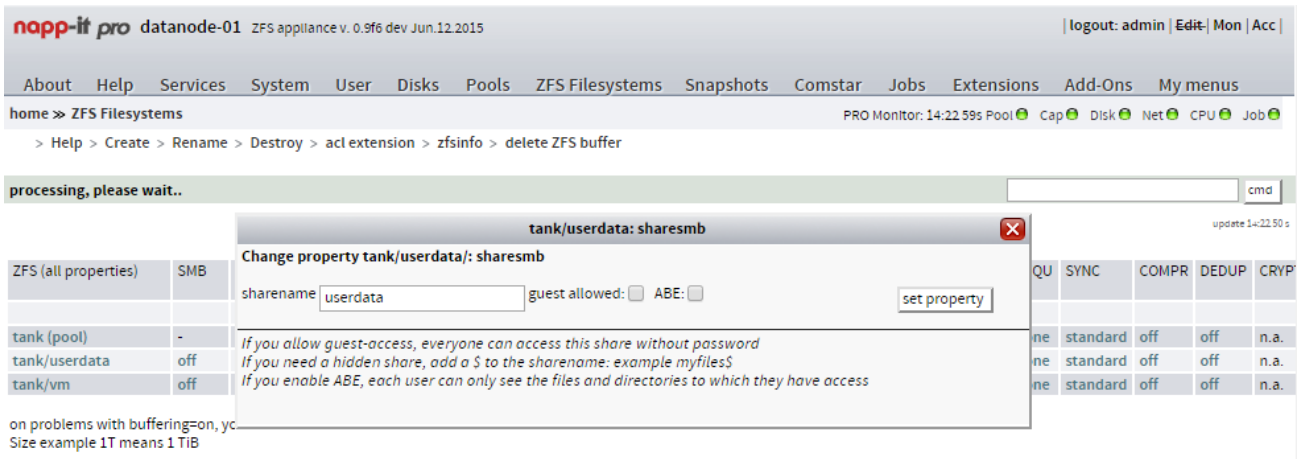
If one compare SAMBA with Solarish CIFS you will find many features in SAMBA that are not available in Solarish CIFS. But Solarish CIFS has some advantages that are not in SAMBA, mainly because SAMBA must run on any X-System with any filesystem. Some of these features are killer features as they affect easyness, performance or Windows compatibility like:

some Advantages of SAMBA over Solarish CIFS

- same server on any X-system
- can act as AD server
- a lot of sharing options
- nested shares/ shares independent from ZFS filesystems
- permissions are based on Unix UID/GID/ Posix ACL, this is a plus if you work mainly in a Unix world
https://en.wikipedia.org/wiki/Access_control_list

some Advantages of Solarish CIFS over SAMBA (used by napp-it)

- fully integrated in ZFS as a filesystem property, easy handling via zfs set command
  there is no configuration file, enable it and set permissions as file/share attribut.
- multithreaded and fast
- integration of ZFS snaps as „Windows previous version"
- manageable via Windows management console (connected users, open files, share level permissions)
- share and file/folder level permissions (Windows server alike)
- permissions are based on NFS4 ACL. allow/ deny with inheritance settings.
  They work very similar to Windows NTFS
- Windows SID as extended ZFS attribute. This allow a move/backup of data in a Windows AD
  environment between servers where permissions are preserved.



enable SMB sharing in menu ZFS Filesystems, click on off in the row of a filesystem under SMB

11.1 SMB related settings (Solarish CIFS)

SMB Service

The SMB service is started automatically when you enable a share.
Some modifications (like share level ACL) require a service restart. This is done automatically by napp-it.

On problems with the SMB server or if you are in a AD Domain that was temporarely unavailable,
it may be needed to restart the service manually in menu „Services >> SMB"
If you import a pool with shares enabled and SMB service disabled, you may get a warning that the SMB service is not enabled. You can ignore as the service is started automatically or after a share off/on.
Set netbios_enable=true (allow Windows network browsing) in menu Services > SMB > properties

SMB Share On

As SMB Sharing is a in Solarish and ZFS integrated property of filesystem, you can enable a share in menu ZFS Filesystems when you click on off in the row of a filesystem under SMB with the following options:

- sharename: The share is visible to a client like Windows under this name
  if you add a „$" to the name, the share is hidden, example userdata$
  To connect such a hidden share, you must connect from Windows like \\datanode-01\userdata$
- guest allowed: You do not need to login with a name and password to access the share (ex from Windows)
- ABE (access based enumeration): Only files and folders are visible where you have permissions

SMB Share off

To disable a share, click on the sharename in the row of the filesystem and set sharesmb = off

SMB permissions

In contrast to other Unix services, Solarish CIFS uses Windows alike NFS4 ACL with permission inheritance, not traditional Unix permissions like 755 or Posix ACLs (https://en.wikipedia.org/wiki/Access_control_list).
This is the reason why you should not set Unix permissions like 755 on files/folders that are shared over SMB as this would delete ACL inheritance settings that are not know in traditional Unix.

Always use ACL to set permissions on Solarish. As tradititional Unix permissions are a subset of the ACL possibilities, they are reduced automatically to fit the ACL permissions.
As ZFS is a Unix filesystem, it must use Unix UID and GID as file security attributes. Solarish CIFS additionally store Windows Security ID's (SID) as extended ZFS attributs. They are used by the CIFS server only and allows file movements/ backups where Windows NTFS alike permissions were preserved - does not matter what UID a user has. This is an advantage especially in an AD environment.

When you create a new ZFS filesystem with napp-it, the default permission is set to
root = full access
everyone@ = modify

This allows that any user can connect a SMB share with read/write permissions as default.
If you do not create new users, only root has (full) access to regular SMB shares at the moment unless you do not had enabled the guest option that allows a connect without login.

If you replicate or move a pool to another Solarish server that is also a domain-member,
all permissions stay intact as the Windows SID/ security ID is stored as an extended ZFS attribute.
This is unique for a Unix filesystem.

SMB guest access

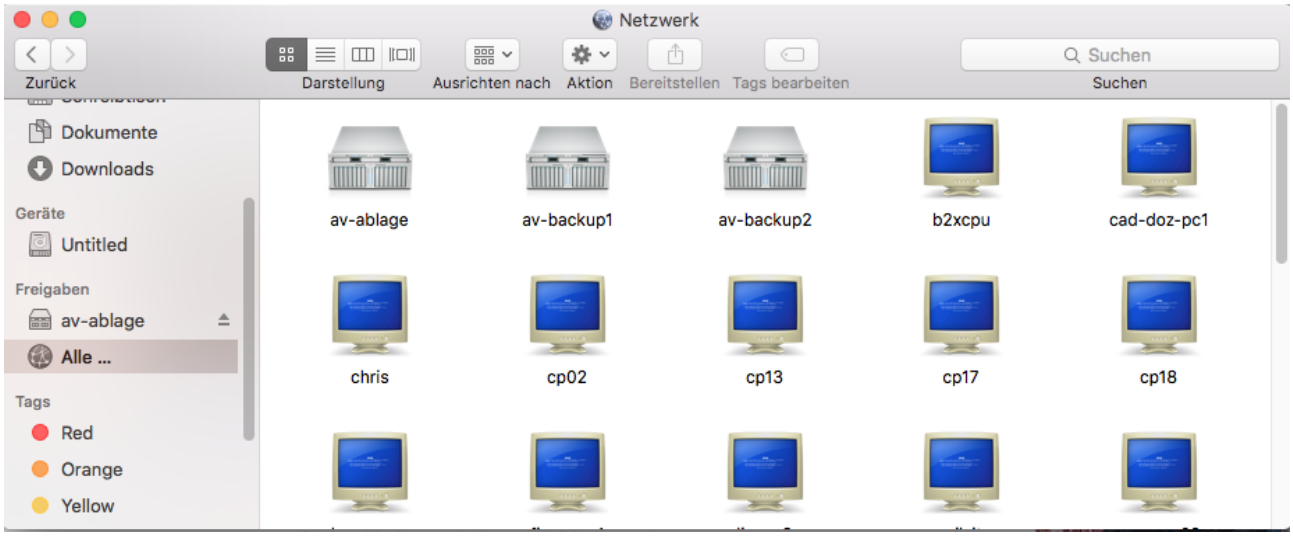
Create a user with name guest (pw does not matter), and enter „smbadm enable-user guest“ at console or the napp-it cmd field and use guest to connect. (OmniOS 151018 and up)

SMB server listing in Windows under „network environment“

Set netbios_enable=true in menu Services > SMB > Properties (OmniOS 151018+).
This function requires an additional Windows Master Browser ex a Windows Server

SMB server browsing in OSX via Bonjour (ZFS server displayed as Apple Xserve)

Enable Bonjour (multicast/dns) in menu Services > Bonjour. This function requires a valid Pro/Dev release



OSX finder view of SMB servers (the Xserve icons are OmniOS ZFS SMB storage servers with Bonjour enabled)

SMB identity mapping

In an Active Directory environment you can map AD-users ex the Domainadmin to root to allow full access for these users to all files - even when there was no explicit permission set for admins as root has always full access. ID mapping is available in menu Users (napp-it ACL extension)

Attention: Do not map local Unix users to local Unix users. Map only AD users to Unix users. You can map local SMB groups to local Unix groups. But for a SMB only usage, you do not need any mappings. They are only needed to use the same users on SMB and other Unix applications.

User backup and restore

If you need to restore an OmniOS/ Solaris ZFS server and want to keep local permissions intact, you must recreate local users with the same user-id.

If you have enabled the napp-it backup function in menu Jobs, you can restore napp-it and all user settings in menu Users > Restore settings (the ACL extension on a napp-it Pro or Dev edition is required)

Howto: Reinstall OS, setup napp-it, insert a license key, update napp-it to a Pro edition and import datapool. Now you can use the menu User > restore to restore settings from your datapool

ACLs on files and folders

Each file and folder on OmniOS has an owner (root or the creator), Unix permissions (traditional Unix permissions like 755) and NFS4 ACL permissions.

If you enter for example at console

```
/usr/bin/ls -V /var/web-gui/napp-it
```

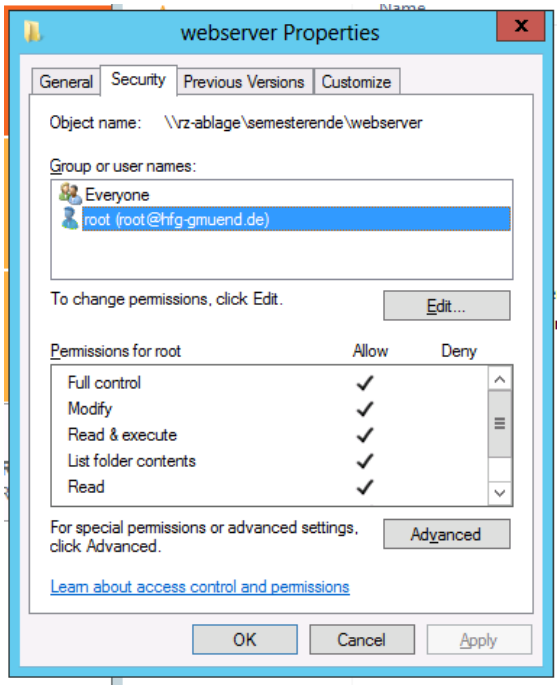
you may get as a result

```
root@datanode-01:/root# /usr/bin/ls -V /var/web-gui/_my
total 2
drwxr-xr-x   2 napp-it  root           2 Mar 19 15:00 wwwroot
                owner@:rwxp--aARWcCos:-------:allow
                group@:r-x---a-R-c--s:-------:allow
             everyone@:r-x---a-R-c--s:-------:allow
drwxr-xr-x   3 napp-it  root           3 Mar 19 15:00 zfsos
        user:2147484183:rwxp-DaARWc--s:fd-----:allow
                owner@:rwxp--aARWcCos:-------:allow
                group@:r-x---a-R-c--s:-------:allow
             everyone@:r-x---a-R-c--s:-------:allow
```

Owner (napp-it), group (root), Unix permissions like drwxr-xr-x and ACLs like owner@:rwxp--aARWcCos:-------:allow or auser ACL are shown.
Windows SID informations are not shown here as they are used in the CIFS server only.

ACLs can be assigned to OmniOS/AD users, OmniOS/SMB or AD groups or as a trivial ACL to owner@, group@ or everyone@ to be compatible to traditional Unix permissions.

You can modify ACL permissions with the /usr/bin/chmod command, per Windows or per napp-it ACL extension. Modifying ACL via CLI command is really stupid. Especially with napp-it free, you can use Windows (beside Home editions) to modify permissions. To do this, you can login to the SMB share from Windows as user root. A right-mouse click >> Property on a file or folders opens the Windows property dialog where you can select Security. Set Permissions just like you would do on a real Windows server.



Good to know:

User root or the owner/creator have always full access, you cannot lock them out - even if permission is not set explicitely. This is normal on Unix and different to Windows (and a boon for any admin doing backups)

If you set ACL on a folder, they are per default inherited to newly created files and folders unless you set „inherit to this folder only" The user that is logged in is the owner of new files and folders (with full permission).

You can override this behaviour with the ZFS property acl inheritance = discard or restricted (default is pass-through).

Windows processes first deny rules then allow. Solaris processes them in their order where the first matching rule is relevant. To set correct deny rules, use napp-it/ ACL extension

ACLs on shares

ACL on shares was introduced by Windows to restrict access independently and additionally to permissions on files and folders, mainly to restrict access without the need to modify file attributes.

On Solaris, a share control file is used that is created when you enable a share.
ex if you enabler a SMB share share tank/userdata, you find the share control file as

```
/tank/userdata/.zfs/shares/userdata
```

You can set share level ACL via napp-it/ ACL extension, or remotely via Windows server management
(You must connect a share/ server management as a user that is a member of the SMB admin group.)

examples:

File ACL: full access
Share ACL: readonly

real permission: readonly

File ACL: readonly
Share ACL: full

real permission: readonly

File ACL: full
Share ACL: user: paul=full

real permission: paul= full, no other user allowed

The default share-level ACL is: full access
(only file attributes are relevant)

Modifications on share level ACL require that you disable/enable a share
or restart the SMB service to take effect.

ZFS Properties aclinherit and aclmode

From http://docs.oracle.com/cd/E36784_01/html/E36835/gbaaz.html#scrolltoc

„ aclinherit

- Determine the behavior of ACL inheritance. Values include:

discard - For new objects, no ACL entries are inherited when a file or directory is created. The ACL on the file or directory is equal to the permission mode of the file or directory.

noallow - For new objects, only inheritable ACL entries that have an access type of deny are inherited.

restricted - For new objects, the write_owner and write_acl permissions are removed when an ACL entry is inherited.

passthrough - When property value is set to passthrough, files are created with a mode determined by the inheritable ACEs. If no inheritable ACEs exist that affect the mode, then the mode is set in accordance to the requested mode from the application.

passthrough-x - Has the same semantics as passthrough, except that when passthrough-x is enabled, files are created with the execute (x) permission, but only if execute permission is set in the file creation mode and in an inheritable ACE that affects the mode.

The default mode for the aclinherit is passthrough (napp-it only).

aclmode

- Modifies ACL behavior when a file is initially created or controls how an ACL is modified during a chmod operation. Values include the following:

discard - A file system with an aclmode property of discard deletes all ACL entries that do not represent the mode of the file. This is the default value.

mask - A file system with an aclmode property of mask reduces user or group permissions. The permissions are reduced, such that they are no greater than the group permission bits, unless it is a user entry that has the same UID as the owner of the file or directory. In this case, the ACL permissions are reduced so that they are no greater than owner permission bits. The mask value also preserves the ACL across mode changes, provided an explicit ACL set operation has not been performed.

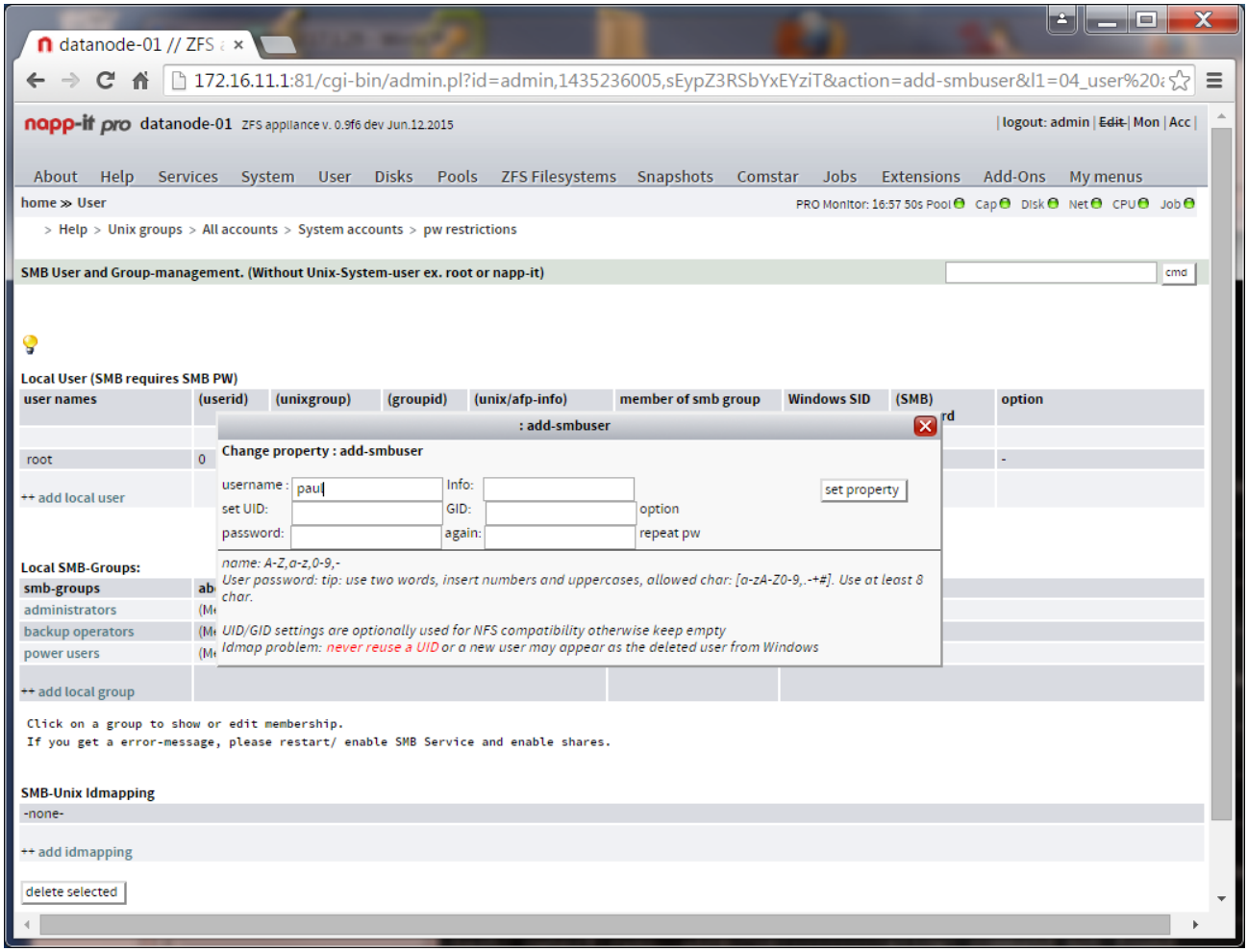
passthrough - A file system with an aclmode property of passthrough indicates that no changes are made to the ACL other than generating the necessary ACL entries to represent the new mode of the file or directory.

The default mode for the aclmode is pass-through (napp-it only). “

aclmode = restricted is added in Illumos/OmniOS to avoid permission modifications with a chmod command (ex via NFS)

12. User and Groups

If you do not need to restrict access to a SMB share, you can enable guestaccess and you are ready.
If you want to restrict access, you can create users with napp-it menu „User“



When you create a user, you only need to enter a username and a password. This user is valid for SMB access and is a valid Unix user. You can assign a UID/GID for a new user when needed ex for NFS (optional).

Attention:

Windows groups and Unix groups behaves different. This is the reason why the Solaris CIFS server come with an own SMB group management that works independently from Unix groups.

If you need groups to restrict SMB access, you must do this with SMB groups.
Menu „User“ allows to create SMB groups add add users to these groups.

There is an idmapping option Winuser -> Unixuser and Wingroup -> Unixgroup in Solaris/ OmniOS.
While a usermapping makes sense only in an AD-environment to map an AD user to a Unix user (never map a local Unixuser to a local Unixuser) you can map local SMB groups to local Unixgroups to achieve a similar permission behaviour within SMB and locally on Unix.

ACL settings for multi-user SMB access

some basic examples for File and Folder ACL settings

Goal:
- everyone can access and read files from a share like data (data is a filesystem below a pool named tank)
- everyone can modify files in data/common and below
- user paul is the only one to access data/paul and below

needed ACL settings:

folder /tank/data:
```
allow everyone@=readx  (read and execute), no-inherit, this folder only
```

folder /tank/data/common:
```
allow everyone@=modify, inherit to folders and subfolders
```

folder /tank/data/paul:
```
allow paul=modify or full, inherit to folders and subfolders
```

Goal:
- everyone can access and read from a share like data (data is a filesystem below a pool named tank)
  read should be allowed only from folder /tank/data, not folders below
- everyone can create new folders but not files on data
- only the creator of a folder (=owner) has access to the new folder and below

needed ACL settings (aclmode must not restrict ownership when creating folders):

folder /tank/data:
```
allow everyone@=readx  (read and execute), no-inherit, this folder only
allow everyone@=create_folder_set, inherit to folders and subfolders
allow owner@=modify or full, inherit to folders and subfolders
```

Goal:
- everyone can access and read from a share like data (data is a filesystem below a pool named tank)
- everyone can read files files on /tank/data/common and below
- members of SMB group „professors" are allow to modify /tank/data/common and below
- only members of SMB group „professors" are allowed to modify /tank/data/professors and below

needed ACL settings (aclmode must not restrict ownership when creating folders):

folder /tank/data:
```
allow everyone@=readx  (read and execute), no-inherit, this folder only
```

folder /tank/data/common:
```
allow everyone@=readx, inherit to folders and subfolders
allow group:professors=modify, inherit to folders and subfolders
```

folder /tank/data/professors:
```
allow group:professors=modify, inherit to folders and subfolders
```

Active Directory: If you want to assign ACL to AD users, this may require that the AD user was logged in once to a SMB share.

12.1 Active Directory

With a few users especially if you use only one or two fileservers, a simple local user management is sufficient, simple and idiot proof. With more users, many servers you need to have the same credidentials on any machine. To achieve this you need a centralized user database like Ldap or Active Directory.

The Solaris CIFS server is prepared to be a member computer in an AD environment.
You can join a Domain with menu „Services >> SMB >> Active Directory

Enter:

Domainname ex	ex myuniversity.edu
IP-Adress of your AD-Master server	ex: 172.16.1.11
Domainadmin username:	admin
Domaiadmin password:	*****

When you click submit, OmniOS syncronizes time and sets DNS to your AD server and joins the domain.
If OmniOS lost connectivity to your AD server, you can disable/enable the SMB service or rejoin the domain.

Care about

Your napp-it Server can be either a member of a workgroup (use local user)
or a domain (can use either local or domain user).

If you switch from domain to workgroup-mode, remove all mappings with idmap remove -a
If you join a domain, you should create at least one idmapping (give domainadmin root permission):
Other mappings are not needed, replace domainadmin with your admin username.

```
idmap add winuser:domainadmin unixuser:root
```

Possible Problems:

If you get a „UNSUCCESSFULL" error

If you want to join a domain newly, please verify that your domain does not already have a computer member with the name of your OmniOS server. In this case you get UNSUCCESSFULL

other reasons for UNSUCCESSFULL:
- wrong username/password

Sometimes UNSUCCESSFULL happens for unknown or timeout reasons.
- If you try again a second or third time it may work.

If you get a „INTERNAL ERROR"
- Check if SMB service is online (you must have enabled a SMB share)

If you get a „Failed to find any domain controller"
- check network, ip and DNS settings
- try another lmauth level (4 is ok for Windows 2012)

- check if there is already a server account in the domain, if so delete prior joining

If you get a „Failed to find any domain controller" on Oracle Solaris 11.4

This is due a new management of DNS service in Solaris where you must set parameters via svcs
compare http://www.fjwilhelm.de/solaris-11-dns-konfiguration/

Step 1, configure client

svccfg -s network/dns/client
svc:/network/dns/client> setprop config/search = astring: („domain.lan")
svc:/network/dns/client> setprop config/domain = astring: („domain.lan")
setprop config/nameserver = net_address: (1.2.3.4 1.2.3.5 1.2.3.6)
svc:/network/dns/client> apply
Usage: apply file

Apply a profile.
svc:/network/dns/client> quit

refresh and restart
svcadm refresh dns/client
svcadm restart dns/client

Step 2: configure DNS

svccfg -s name-service/switch
svc:/system/name-service/switch> setprop config/host = astring: „files dns"
svc:/system/name-service/switch> setprop config/ipnodes = astring: „files dns"
svc:/system/name-service/switch> apply
Usage: apply file

Apply a profile.
svc:/system/name-service/switch> exit

Name Service Dienst refresh and restart

svcadm refresh name-service/switch
svcadm restart name-service/switch

root@server01:~# svcadm refresh name-service/switch
root@server01:~# svcadm restart name-service/switch

btw
complicated was yesterday, this is overcomplicated!!!

13. NFS shares

NFS is a filesharing protocol from the Unix world that is supported in NFS v3 and NFS v4.
Mostly NFS3 is used in secure environments where you mainly need performance as NFS3 lacks Authenthification (to login with name/pw) and Authorisation (no restrictions based on file permissions) beside some „good will" settings based on client ip and client UID.

You can enable/disable NFS in menu „ZFS filesystems" in the row of a filesystem under NFS as a filesystem property, similar to SMB sharing. Mostly you set on or off. Other option is to restrict access based on client ip or allow full acces independently from the client UID.

Enable NFS

```
set NFS = on

or instead on on something like
rw=@192.168.1.0/24,root=@192.168.1.0/24 tank/vm
```

Disable NFS

```
set NFS = off
```

14. iSCSI shares

iSCSI is not a multiuser filesharing protocoll like SMB or NFS. It offers blockstorage to a single client that is treated there like a local disk and formatted from the client with a filesystem. You need a cluster filesystem if you want to allow access from two clients simultaniously.

Originally, Sun implemented iSCSI sharing as a filesystem property like NFS and SMB. As iSCSI is mostly used in large and complex HA environments, this approach was replaced by COMSTAR, a enterprise framework to manage iSCSI and FC environments.

When should you use iSCSI/ FC

- when you need a non-ZFS filesystem like ext4, HFS+, NTFS or VMFS
  ex ESXi environments or a Windows Server based on ZFS blockstorage
- in HA environments with a setup similar to SAS multipath but with iSCSI datanode multipath
  to allow large capacity, high performance or remote installations (not limited by cable length)
- in HA environments with dataheads or clients (ex storageserver with services or a server) and datanodes that provide their storage via iSCSI. This can be a simple datanode mirror or a raid-Z over datanodes.

Enable iSCSI via menu Comstar

1. create a Logical Unit (LU). This can be a ZFS volume, a file or a RAW disk
2. create a target (this is the part that you connect from a client)
3. create a target group with targets as members
4. add a view from your logical units to a target group to make them visible as a LUN.

Enable iSCSI via menu ZFS Filesystems

For smaller installations, Comstar is quite complex. Napp-it offers a way where you can enable iSCSI on a per filesystem way with a on/off switch in menu ZFS filesystems in the row of a filesystem under iSCSI.
If you enable iSCSI here, you create a logical unit, a target, a target group and a view in a 1:1 relation.
If you need more than on/off or basic settings, you can manage the targets with menu Comstar as well.

15. Data Scrubbing

With older filesystems like Ext4 or NTFS you have had the following problem.
You edit a file and you have a crash or powerloss while you write/ update the file. As these filesystems do inline modifications (modifiy parts of the current data on disk) the following can happen:

regarding your data
1. nothing written, old data is valid
2. new valid data, modifications are correct
3. modificated data is parly written, no chance to detect or repair the problem

regarding the metadata
1. metadate correct updated
2. metadata corrupt=corrupt filesystem, an offline fschk is needed to repair the filesystem structure

No chance to detect metadata problems beside a offline fschk that can last days and this does not even help to detect or repair data corruptions. The result is only valid metadata structures.

ZFS, a new generation filesystem

https://en.wikipedia.org/wiki/ZFS

ZFS stands for a new generation of filesystem that do not try to reduce these problems but to avoid them completely with two basic principles: CopyOnWrite and End to End Checksums on data/OS level.
CopyOnWrite means, that you do not do inline updates of old data but write datablocks always new.
In the above powerloss szenario, ZFS behaves like:

regarding your data and metadata
1. modified data is written completely new, data pointers are updated, new data is valid and verified
2. modified data is not written completely, data pointers are not updated, old data keeps valid and verified
3. If anything goes wrong, this will be detected by checksums on next read and auto repaired (self healing)

That does not mean, that you cannot have a dataloss. If you write/update a large file with a powerloss, this file is corrupt and damaged (no miracle with ZFS) but your filesystem is not damaged ans always valid.

reasons for corrupted data on disk

- Powerloss or a system crash
- I/O errors due to a bad disk, controller, cabling, backplane or PSU
- On-disk data corruption due to cosmic rays, magnetic or electromagnetic fields
- Driver bugs resulting in data being transferred to or from the wrong location
- A user or Virus modifying data by accident or intention

All beside the last problem can be at least detected and mostly autorepaired by checksums and redundancy.
For the last point ZFS offers snapshots. On a multiTerabyte Array you will always find corrupted data over time.

Scrub on every read, on demand or as a planned job

On every read, data is checked and repaired from Raid redundancy when a checksum error is detected. (auto self healing filesystem). If you want to check the whole pool you can start a scrub manually or as a planned napp-it Job. With desktop disks I would do this once a month ex on a low io day like saturday. Unlike a traditional fschk that require an offline filesystem for quite a long time without a real repair option, a scrub is a online process that runs in the background with low priority and verifies/repair all data.

I own many server and systems and see checksum repairs quite often: This feature is mandatory!

16. Data Snapshots, Versioning and Backup

If you care about your data, you do backups. If you really care about your data, you do multiple backups like tapes that you rotate on a daily or weekly base for a data history. If your data was deleted or modified by accident or intention (virus, staff) you have a chance to regain original data from a former backup.

While you always need a backup for a real disaster like fire or a thief, this concept has three weak points.
The first is that the number of backups is mostly limited due to limited resources.
The second is, that access to a backup means mostly a restore that is at least inconvenient.
The third is, that you often cannot trust the backup, because it usually has no checksum verification and no repair option on problems. Mostly you or the admin discover this when you need the backup - too late..

Data versioning/ secure backup

For the data versioning problem with regular user access to former states, you can save several versions of a file like report-2015.doc, report-2015v2.doc or report-2015-this is the latest.doc.

Another option is a mechanism like Apple Timemachine, where you copy/sync dataversions to a another disk on a regular base like once a day. While this work it is annoying because of the delay when you must copy or restore huge data.

Another option is Windows shadow copies on a Windows server (https://en.wikipedia.org/wiki/Shadow_Copy). This is a block level snapshot mechanism of the whole volume. The result is a versioning filesystem. If you do daily snaps, you can browse/ restore the data with Windows „Previous Versions".

The problem remains, that you should not do too many snaps. I also had a problem with Windows VSS snaps in the past that they were lost after a system crash with a fress OS install. Main problem: you cannot really trust NTFS filesysystems (not always consistent like a CopyOnWrite filesystem) and no checksums (no verified data, no autorepair). This may be different in future with ReFS but currently this is not a comparable option to ZFS.

ZFS snapshots

ZFS snapshots are far better than the former solutions. ZFS is a CopyOnWrite filesystem where all modified datablocks are written newly while the former datablocks can be overwritten after a succesfull write.
A ZFS snapshot means that the former datablocks are blocked and cannot be re-used unless the snap is deleted. This requires only to keep some datapointers and can be done without delay and no initial space consumption. Even ten-thousands of snaps can be hold without any problem (Okay as the former state blocks capacity, sometimes the pool is full). As this is done on ZFS storage, checksum verification, scrubbing and autorepair is working - does not matter how old a snap is - ideal for long term storage and archives with regular scrubs.
As this is managed by the ZFS pool itself, you are not in danger too loose them when you move a pool. You can also trust these snaps absolutely. An admin can destoy a snap but cannot modify data as a snap is readonly.
The best is, this is transparent to a user. You can access ZFS snaps on Solaris via „Windows Previous Versions" with all ACL or AD permissions intact even from a Backup. With a snapshot rule like take a snap every 15 Min-keep 4, every hour-keep 24, every week-keep 4, every months- keep 24 you can go back two years on a filer.

Snapshots > Mass delete allows to destroy snaps based on name or age. On a valid Pro or Dev edition, you can keep on snap per filesysten per day/week/month or mass destroy in the background (required for many snaps)
.

ZFS backup

While you can backup data from a ZFS storage to any system, ZFS offers remote and ultrafast incremential replication based on snaps where only modified datablocks are transferred with ZFS security and their own snapshot history. In my own setups, I use two main backup systems in my serverroom where I replicate data based on even or uneven days and a third backup system in another building for important data and a snapshot history that covers at least 90 days (daily snaps)

17. Async Data Replikation/ Avalability

Unlike other backup or sync methods, ZFS replication can keep an active ZFS filesystem with open files in sync with another filesystem on the same or a remote ZFS server. You can run replications down to every minute to sync filesystems near realtime.

A replication initially transfers the whole filesystem with ZFS properies based on a snapshot. After the first run, replication works as an incremental transfer based on snaps on source and target that must be in sync and represents the same data state. On an incremental run, the target filesystem is resetted to the same state as the last source filesystem snap. A new source snap represents the datablock differences between this snap and the current state. Only this snap is transferred as a datastream. This makes replication ultra fast. Napp-it replicates over a buffered netcat connection up to wireframe performance. As replication is based on snaps , each data state is like a hard shutdown, As a replication run resets a target filesystem to a former state to be in sync with the source filesystem, you should should not use it beside reads between replications. If you need to switch to a replicated filesystem a main filesystem. you must stop replication and set the filesystem to read/write. To switch back, you can replicate the filesystem back.

Snapshots on replicated filesystems.
As a replication target filesystem is resetted to a base snap on every replication run, you can keep older replication snaps based on date or name. On a source filesystem you can use regular snaps for versioning.

17.1 Sync Data Replikation

If you need realtim sync with the exact same datastate at any time you can use a mirror between appliances. You need two or more storage nodes (independent ZFS storage servers) that offer a ZFS file-system over a fast network connection as a FC or iSCSI target. A storagehead can the built a ZFS pool over these iSCSI targets as a mirror or raid-Z over nodes.

17.2 Data Availability from Backup to active/active HA

There are several steps to increase data availability.

1. Availability due Backup
First step is backup from time to time to a physically separate place.
Especially on a disaster you can restore a former datastate. Restore can last a very long time.

2. Availability due Replication to a backup/spare system
This is near realtime local copy to a spare/backup system. On problems you can switch services to this datastate in a very short time. If you have enough free bays on your backup/spare system and your main pool is working, you can also move and import the pool to regain service availability.
From a crash, time to regain services is between 15 minutes and an hour

3. High avaiability with a multipath SAS storage and two nodes.
Mostly problems are not related to disks but HBAs, Mainboards or a system configuration.
A typical HA config can share dualpath SAS disks to two identical appliances. A HA Software like RSF-1 can manage the two appliances and switch services between them in a few seconds.

As an addition you can use two appliances and two storage nodes and connect them via iSCSI. On a storage node or a storage head problem, you can automatically switch between under control of RSF-1.

HA is usually quite complicated and should be done only under support. Napp-it does not support HA in the GUI, but you can use RSF-1 from http://www.high-availability.com/zfs-ha-plugin/

18. Data Rollback and Clones (on valid Pro/ Dev editions)

If you need any sort of protection against unintenionally or intentionally deleted or modified data like a file delete, file overwrite or file ecncypt ex by a malware, you need versioning with snaps.

Once you have snaps, you have several options to access data from snaps.

18. 1. Access snaps on a per file base

Snaps are readonly files in folder /filesystem/.zfs/snapshot. You can use a filebrowser
or you can Windows and „previous versions" to access previous versions of files.

18.2 Rollback a whole filesystem to a snap.

You can use the zfs rollback to discard all changes made to a file system since a specific snapshot was created. The file system reverts to its state at the time the snapshot was taken.

Warning! If you roll back, all intermediate and all clones based on them are destroyed without an undo option. For most rollback cases, using Windows and Previous version is the suggested method for a roll back as you can restrict the rollback to files and folders.

You can use napp-it menu „ZFS-Filesystem >> Rollback to initiate a rollback

18.3 Create a Clone (writeable filesystem from a snap).

This is called a clone that you can use like a regular filesystem. You can create clones from console
or Napp-it menu Snapshots >> Clones (napp-it Pro only)

A clone is a writable volume or file system whose initial contents are the same as the snapshot from which it was created. Creating a clone is nearly instantaneous and initially consumes no additional disk space. In addition, you can snapshot a clone.

Clones can only be created from a snapshot. When a snapshot is cloned, an implicit dependency is created between the clone and snapshot. Even though the clone is created somewhere else in the dataset hierarchy, the original snapshot cannot be destroyed as long as the clone exists.

The origin property exposes this dependency, and the zfs destroy command lists any such dependencies, if they exist. Clones do not inherit the properties of the dataset from which it was created.
Because a clone initially shares all its disk space with the original snapshot, its used property value is initially zero. As changes are made to the clone, it uses more disk space. The used property of the original snapshot does not include the disk space consumed by the clone.

The parent filesystem of the parent snap must be writable or you cannot mount the clone.
Clones must be destroyed before the parent snapshot can be destroyed.

18.4 Promote a Clone (Replacing a ZFS File System With a ZFS Clone)

You can use the zfs promote command to replace an active ZFS file system with a clone of that file system. This feature enables you to clone and replace file systems so that the original file system becomes the clone of the file system. In addition, this feature makes it possible to destroy the file system from which the clone was originally created. Without clone promotion, you cannot destroy an original file system of active clones.

19. Basic operational settings

Your storage appliance is now up and running. Care about the following settings

Napp-it settings (menu About >> Settings)

all settings are stored in /var/web-gui/_log/napp-it.cfg

- set passwords for admin and operator (encrypted one way hashvalues)
- set email (mailserver, mailuser, mailpw, store unencrypted)
- set push data (alerts to your desktop or smartphone)

System-Settings

- Menu Sytem >> HW and Localization >> Localization
  ex America > New_York, set Language en_US.UTF-8 and your keyboard, you need a reboot

- create bootable snapshots (=BE, bootenvoronments) manually prior serious system modifications
  This is done automatically on OS or napp-it updates and allows a bootup on a former OS state.

Auto-Job Settings

- Enable napp-it auto-job to 5min (Jobs >> autos ervice)
- set other job to sync time via ntpdate > AD server or any other ntpserver

- Set email-alert and status jobs in menu Jobs >> Email >> alert or status
  Per default napp-it sends email unencrypted over port 25
  If your smtp server requires TLS encrypted mail example Googlemail over port 587, you must
    - install TLS modules, see http://napp-it.org/downloads/tls.html
    - switch napp-it to use TLS in menu Jobs >> TLS Email >> enable TLS

- Set push-alert (Pushalot or Pushover) for your desktop or smartphone
    - more see www.pushalot.com (Windows8 and -Phone, free) and www.pushover.net (ios, Android)

- Set a backup job (Jobs >backup >> create backup job) tp backup basic OS and napp-it settings to a pool
  Restore all user, SMB groups, idmappings and other napp-it settings then via User > Restore (ACL extension)

- Set autoscrub jobs (see 15.)
  for your pools in menu Jobs >> scrub >> create autoscrub job
    - ex set autoscrub of your system pool (rpool) to every first sat (of a month)
    - set autoscrub of your datapools (with desktop disks I would use once a monty as well)

- Set autosnap jobs (see 16.)
  for your pools in menu Jobs >> snap >> create autosnap job
  As ZFS snaps are readonly and cannot be modified/destroyed from a share, they are virus/user save
  This is your first and most important method to avoid dataloss and undo unwanted modifications ex:
    - set autosnap: snap every hour, keep 24
    - set autosnap: snap every day, keep 31
    - set autosnap: snap every 1st of a month, keep 12

Your primary storage ist where you should care about a highest possible level of raid and data security. Data restoring can be done mostly from your primary storage as ZFS is a versioning filesystem with snaps.

To be prepared for a real disaster (sabotage, fire, overvoltage or a thief), you need a disaster backup at least with some snapshots. If data is important, this should be done to two different systems where one must be on a different physical location like another building or offline within a save. You can do this via sync (rsync or robocopy) or via the faster ZFS incremental replication that can be done every few minutes.
- set a replication job to another napp-it appliance (require replication extension and grouping)

20. Security

Restrict access to management functions

- Web management is done via port 81 for http or port 82 for https
  Realtime graphic/ websocket is displayed over port 3000 (https/wss port 36000)
  If you enable wss in About > Settings, you must install the SSL/TLS modules, see Jobs > TLS email
  If you want to use your own certificate, place it at /var/web-gui/_log/mini_httpd.pem, otherwise
  the sample certificate at /var/web-gui/data/tools/httpd/mini_httpd.pem is used

- Remote console via Putty and remote fileaccess via WinSCP is done on SSH port 22
- Replications are done over a random port > 49000

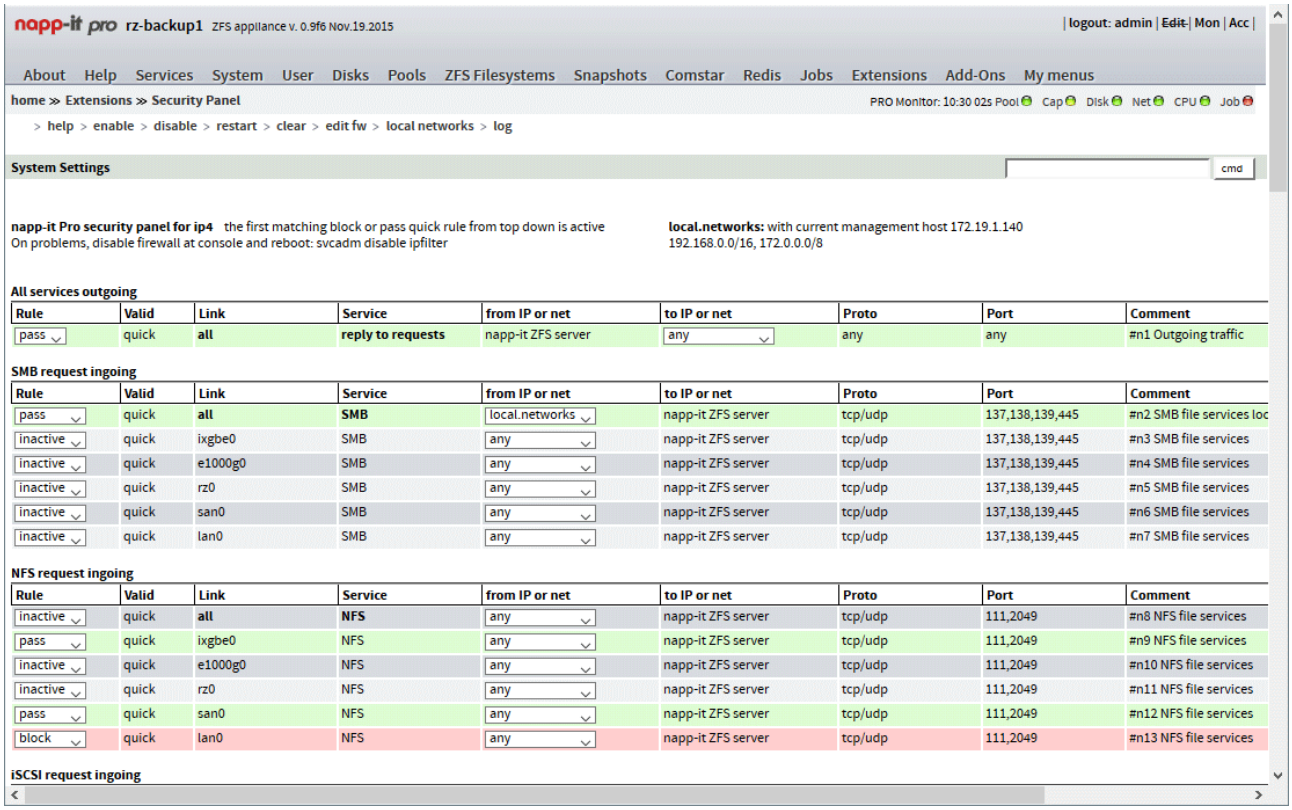
In an unsecure environment, you should restrict the above ports to a secure environment,
either based on a network adapter (link) or based on your networks

Restrict access to file services

- Fileserveices like NFS3 do not offer authentication. Access can be only limited to a fakeable source ip. This can be a security problem example when you offer NFS for ESXi where your storage server is accessable over untrusted networks for management or other services.

In an unsecure environment, you should restrict access to services like iSCSI, NFS, SMB or WWW
either based on a network adapter (link) or based on your local networks or single ip adresses.

Firewall settings/ Security panel (available on a valid Pro or Dev edition)



You can use the napp-it Pro security panel to restrict access based on a set of ip adresses or local networks or based on a network adapter. With napp-it free, set the according rules manually.

21. Appliance Tuning

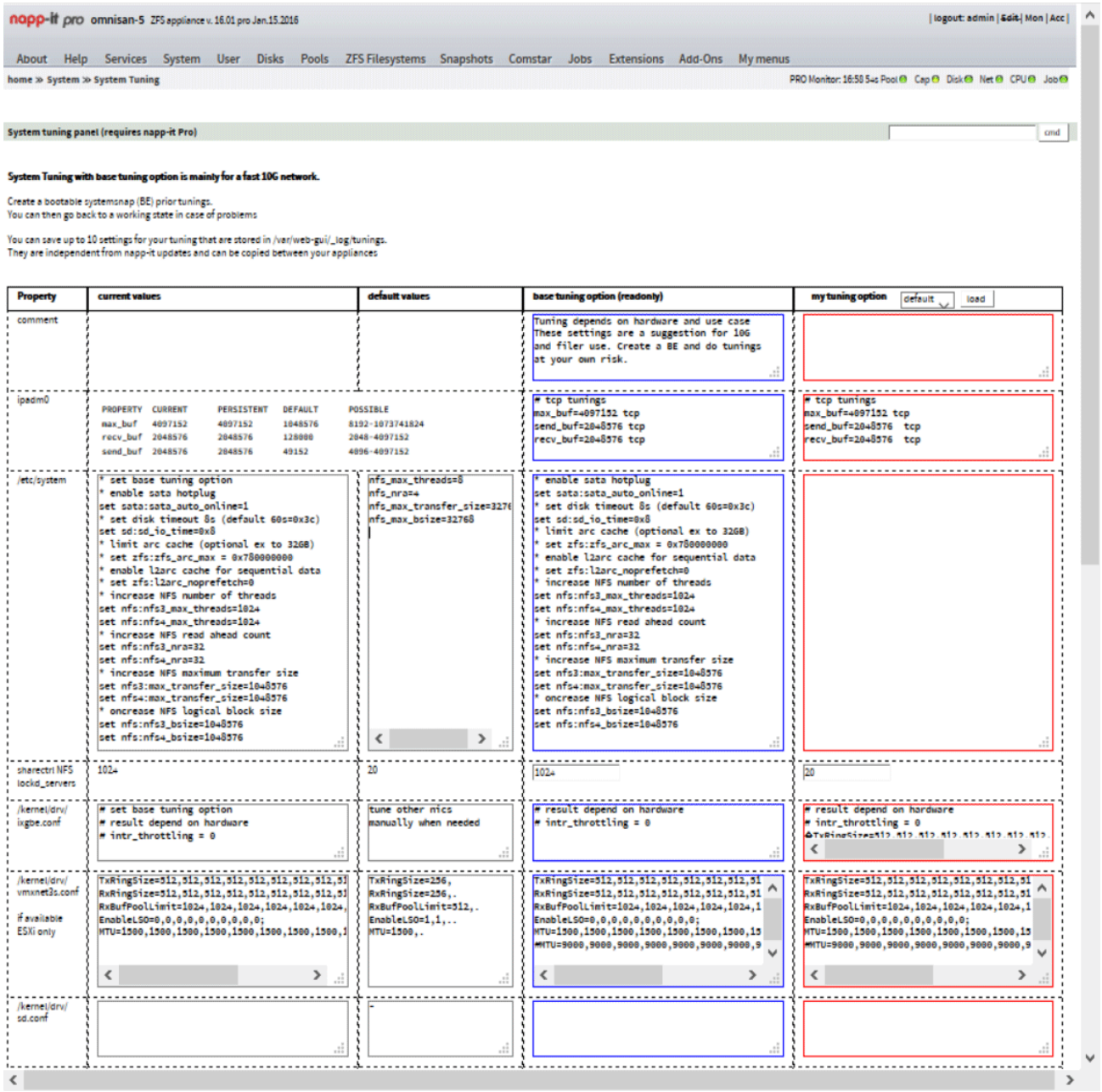
(available on a valid Pro or Dev complete edition on Nic, System/IP or service level (iSCSI, NFS, SMB)

Storage performance depend on your physical network, your server and storage hardware and your settings that can be optimized to a special hardware environment or a special use case. The defaults are intended for 1Gb/s Ethernet networks and a typical filer use. Above that a Solaris based system offer a lot of tuning options.

When 1Gb/s networks (about 100MB/s) came up many years ago, this was sufficient as storage capacity was quite low and performance of an average internal disk and network performance was quite similar.

Now storage is in the multiple Terabyte area with a view to Petabyte. A simple backup or move of data from one pool to another over network can last days or weeks. Local storage with the new NVMe SSDs can give a performance of 1000MB/s or more. This is why we need faster networks like 10Gb/s Ethernet (about 1000 MB/s) and for this we need some tuning settings to achieve full performance on your hardware or for a workload..

Napp-it Appliance Tuning Panel (napp-it Pro complete only)



21.1 Tuning aspects

- TCP/IP tuning of Server side

This includes send and receive buffers that can be modified on a running system and mtu settings (Jumboframes, mtu=9000) that require a reboot.

- TCP/IP tuning on switches
you must enable mto=9000 or Jumboframes or the switch blocks Jumboframes

- TCP/IP on client side
This depend on your hardware. For examle with Windows and the mainstream 10G adapter Intel X540, you should disable interrupt throtteling.

- System and service tuning in /etc/system that require a reboot
like NFS buffers, hotplug behaviour of AHCI Sata, timeout of disks on problems and many other aspects.

- Nic (ex in ixgbe.conf) and vnic (vmxnet3s.conf) settings
These are mainly mtu and buffer settings that should be increased for 10G

- Disk settings in sd.conf
Mainly settings for advanced sector disks, power management and other aspects of disks

- SAS controller settings ex in mpt_sas.conf
Settings like mpio and other controller dependent SAS aspects

- napp-in-one tuning
If you use napp-in one to connect ESXi over the internal vswitch with the ZFS storage VM, all transfers are in software so most netwok and Ethernet centric tunings aspects are not needed, You should use vmxnet3 as vnic as it is much faster than the e1000 vnic with the base vmxnet3 tuning. The other tuning aspects are mainly relevant for external access or if you use a 10G nic in pass-through mode for fastest external access.

Remarks about napp-in-one tuning (ESXi related aspects)
You mainly use NFS to offer ZFS storage for ESXi. If you modify OmniOS settings like the vmxnet3 settings, it can happen that you need to reboot OmniOS twice (check console on boot as vmxnet3 settings are displayed on bootup). Some settings result in an „All Path Down=NFS not availabe„ error in ESXi. You can fix this with an ESXi reboot. Some NFS service modifications hinder ESXi to reconnect at all. You can fix this when you delete are VMs from inventory (NOT from disk), delete the NFS mount, readd the NFS mount and reimport the VMs to inventory (use ESXi filebrowser and a right mouse click on the .vmx file in the VM folder)

Tuning for napp-it Free
You can manually modify system settings via console, Putty or remotly edit files via WinSCP
You can try the settings from the napp-it Pro tuning console

Tuning for napp-it Pro
You can use the tuning console that allows
- enable a basic tuning with the options for an immediate reboot and a BE creation
- use up to ten private editable tuning configurations that can be easily moved to other appliances

Hints about other aspects like SSD, iSCSI settings or Pool considerations
see napp-it menu System > System tuning

21.2 Network aspects

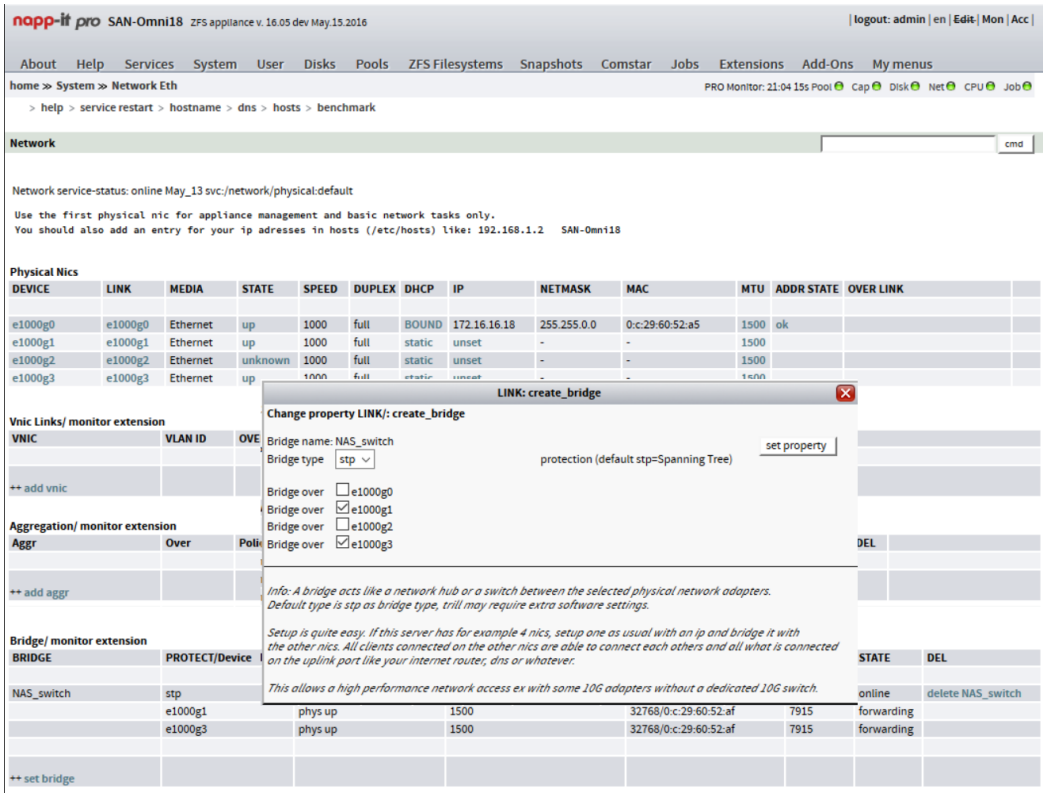
A modern NVMe PCI-e SSD can offer 1000MB/s sequentially and more. A regular disk offers sequentially up to 200 MB/s. If you build a Raid-Z over disks, your pool is capable of a similar performance than an NVMe or better. This and a typical capacity of a NAS in the Terabyte to Petabyte range where you need days or weeks to copy this over to another NAS or backup system requires a faster network than the mostly available 1G/s network with a transfer rate of up to about 100MB/s.

So an adequate network performance that has a relation to your pool performance and that allows to transfer data in the Terabyte range is simply not possible if you only use 1G/s networks.
10G networks with a transfer capacity up to 1000MB/s is the solution. If you buy a new serverclass mainboard you can find boards with a minimal premium with 10G onboard, You should use these offers. There are also quite cheap 10G nics around. Outside the Windows and Linux world, you should stay with Intel cards and even with Windows they are usually the fastest and most stable option.
Then connect your NAS with a 10G capable switch, either via SFP+ (optical up to 70km or copper up to 5m) or 1000G Base T, the traditional ethernet cable up to 100m. If there are 1G clients attached to that switch, this will give up to 10 clients full 1G each. If you have more 10G clients, you need a switch with enough 10G ports. This introduces two problems. One is that 10G switches with more than two ports are quite expensive. Then they are quite loud and only an option in a serverroom. In a Lab or Desktop environment this is not an option,

Napp-it = 1/10G Switch = Cheap and silent high performance storage/networks for labs/ office/ desktops

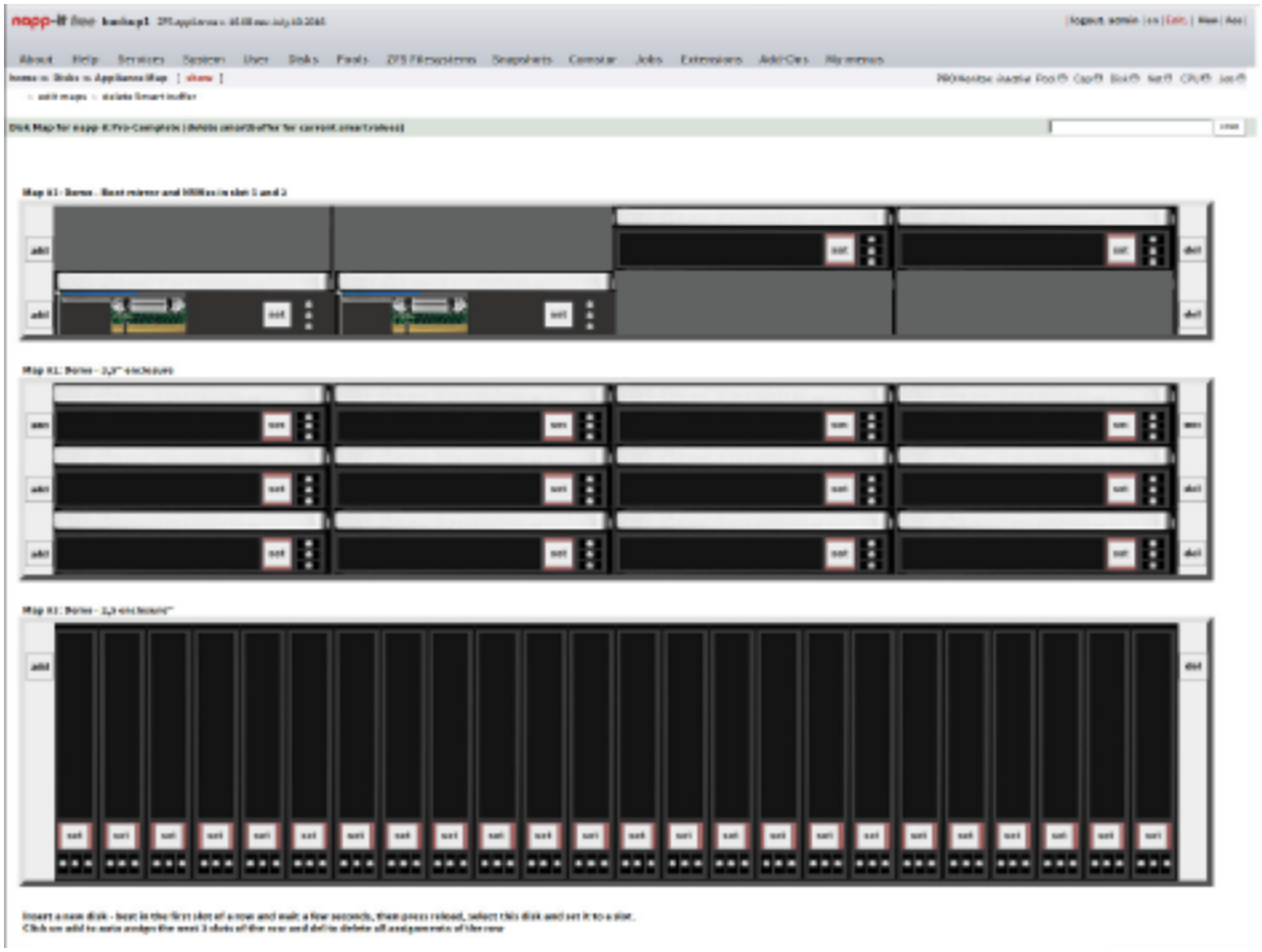
If you mainly need 10G access from your NAS to your backupsystem, some ESXi hosts or some videoediting clients, you can simply add some 10/40G nics to your NAS. This can range from Intel X520-D2 to X540-T2 over the Intel X710 where driver support for OmniOS is on the way. The X710 is a single or dual QSFP+ card that offers up to 2 x 40G/s or alternatively up to 8 SFP+. If you use cheap chopper QSFP+ to 4 x SFP+ cables (up to 5m), you can provide fast 10G access to a workgroup of 8 clients per nic. For longer distances you must use optical cables or an additional X520 or 640 adapter.

If you enable the bridging functionality, napp-it acts like a regular 1/10/40G switch. Every computer can connect the NAS, each other and the internet that is connected for example on the 1G onboard nic of the NAS. You can enable the bridging/ switching functionality per CLI commands or with napp-it Pro monitor up from 2016.05 dev where you enable switching in menu System > Network Eth. If you use for example the Super-Micro X9SRL-F you can add up to 5 x NVMe PCI-e adapters, up to 10 Sata SSDs with a SM-M28SAB-OEM and two additional nics (10/40G). This storage is a ultrafast and ultrasilent Office NAS without an extra 10G switch.



22. Appliance Maps (16.4f and 16.08 dev or newer with napp-it Pro complete)

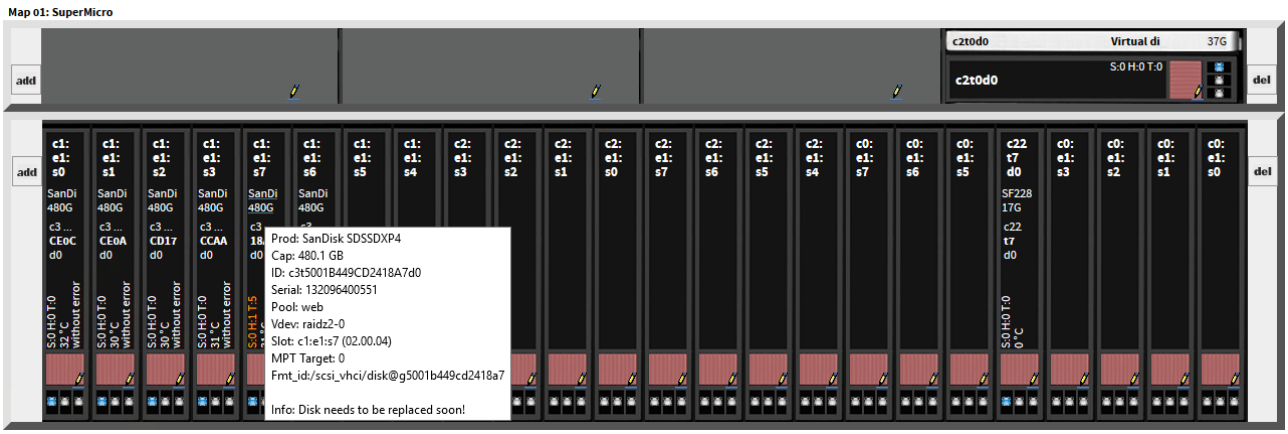
Appliance Maps works with Sata or LSI HBAs where it uses the sasircu tool (like Disks > DISK Location). It allows to display maps of your enclosures (up from napp-it 16.8). You can create up to 9 maps and assign disks to the map. You can then printout a screenshot of a map an place it on the server as a reference. If a disk fails, the map allows to identify the slot of the failed disk.



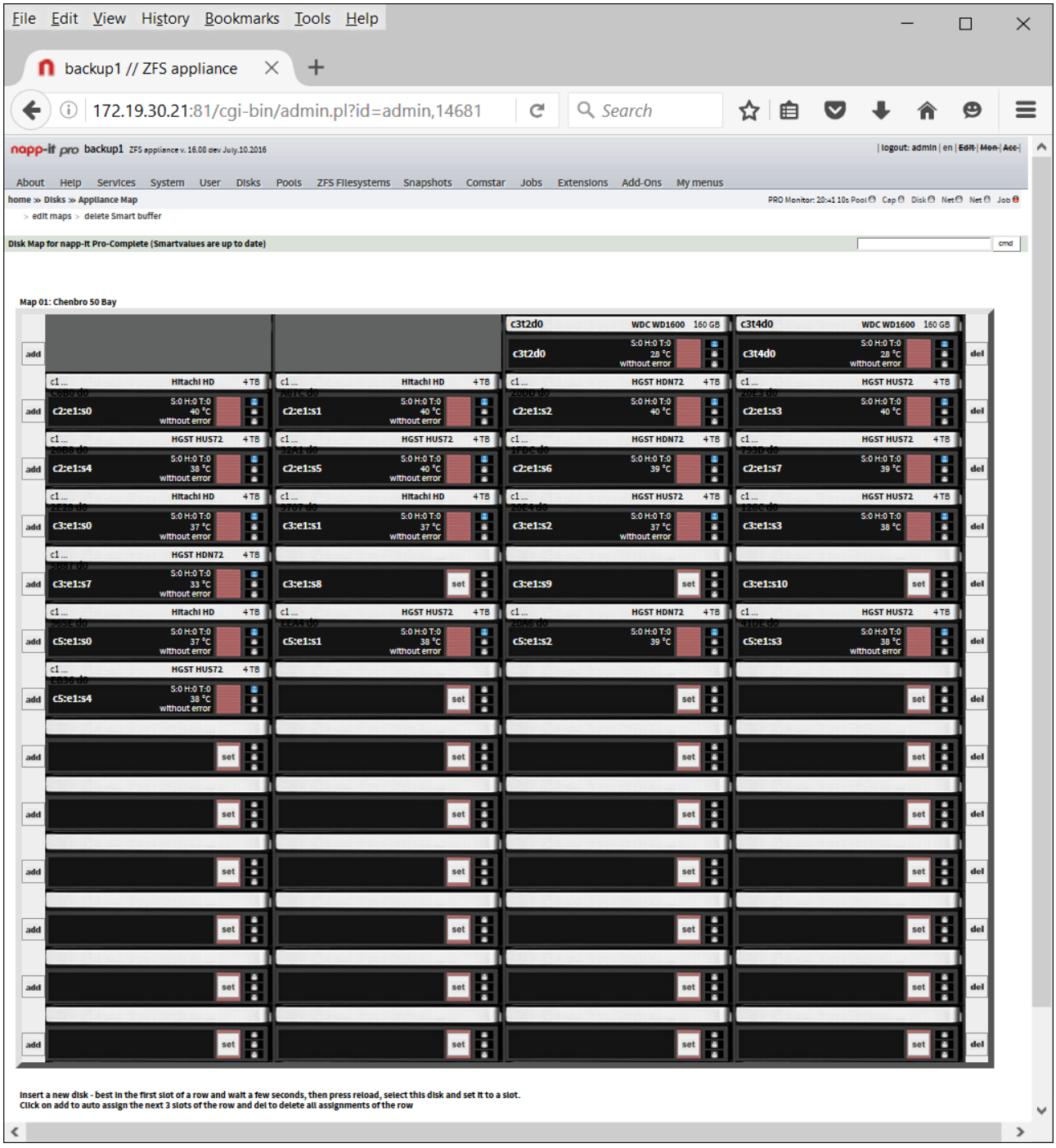
Example: demo maps (without disk assignment)

Map 1: your boot enclosure with 2 x 3,5" disks and 2 x NVMe in PCI-e Slot 1
Map 2: a 12 x 3,5" Bay enclosure (Max 90 Bay SuperMicro)
Map 3: a 24 x 3,5" Enclosure.

Example: bootdisk + 24 x 2,5" SSD enclosure with new editing option to add remarks to a slot

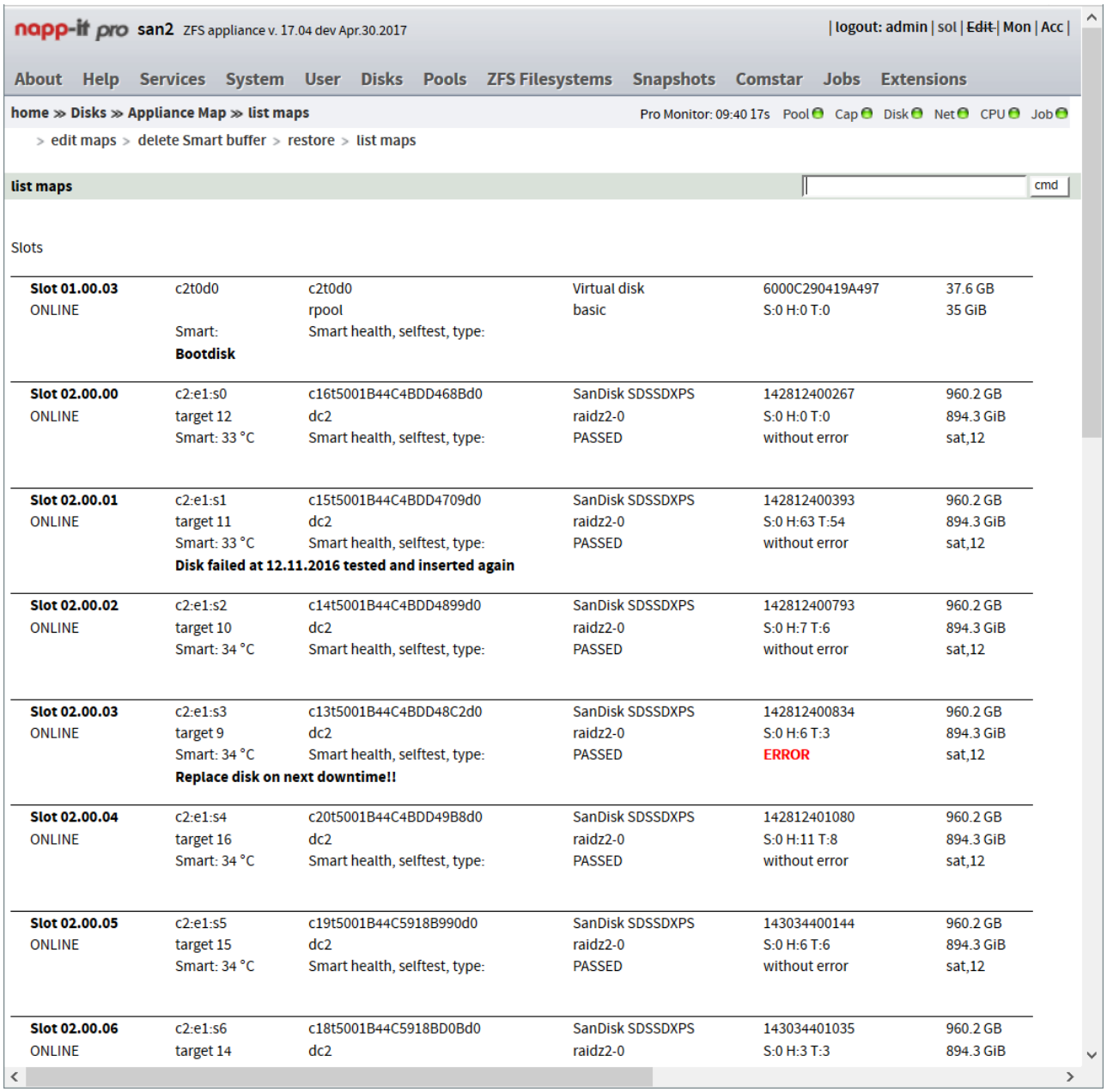


Example:
Map Chenbro 50 x 3,5“ Bay



Map list:

this list is generated from map data, disk location and smart values



napp-it pro san2 ZFS appliance v. 17.04 dev Apr.30.2017 | logout: admin | sol | Edit | Mon | Acc |

About Help Services System User Disks Pools ZFS Filesystems Snapshots Comstar Jobs Extensions

home » Disks » Appliance Map » list maps — Pro Monitor: 09:40 17s Pool Cap Disk Net CPU Job

> edit maps > delete Smart buffer > restore > list maps

list maps — cmd

Slots

Slot 01.00.03	c2t0d0	c2t0d0	Virtual disk	6000C290419A497	37.6 GB
ONLINE		rpool	basic	S:0 H:0 T:0	35 GiB
	Smart:	Smart health, selftest, type:			
	Bootdisk				
Slot 02.00.00	c2:e1:s0	c16t5001B44C4BDD468Bd0	SanDisk SDSSDXPS	142812400267	960.2 GB
ONLINE	target 12	dc2	raidz2-0	S:0 H:0 T:0	894.3 GiB
	Smart: 33 °C	Smart health, selftest, type:	PASSED	without error	sat,12
Slot 02.00.01	c2:e1:s1	c15t5001B44C4BDD4709d0	SanDisk SDSSDXPS	142812400393	960.2 GB
ONLINE	target 11	dc2	raidz2-0	S:0 H:63 T:54	894.3 GiB
	Smart: 33 °C	Smart health, selftest, type:	PASSED	without error	sat,12
	Disk failed at 12.11.2016 tested and inserted again				
Slot 02.00.02	c2:e1:s2	c14t5001B44C4BDD4899d0	SanDisk SDSSDXPS	142812400793	960.2 GB
ONLINE	target 10	dc2	raidz2-0	S:0 H:7 T:6	894.3 GiB
	Smart: 34 °C	Smart health, selftest, type:	PASSED	without error	sat,12
Slot 02.00.03	c2:e1:s3	c13t5001B44C4BDD48C2d0	SanDisk SDSSDXPS	142812400834	960.2 GB
ONLINE	target 9	dc2	raidz2-0	S:0 H:6 T:3	894.3 GiB
	Smart: 34 °C	Smart health, selftest, type:	PASSED	ERROR	sat,12
	Replace disk on next downtime!!				
Slot 02.00.04	c2:e1:s4	c20t5001B44C4BDD49B8d0	SanDisk SDSSDXPS	142812401080	960.2 GB
ONLINE	target 16	dc2	raidz2-0	S:0 H:11 T:8	894.3 GiB
	Smart: 34 °C	Smart health, selftest, type:	PASSED	without error	sat,12
Slot 02.00.05	c2:e1:s5	c19t5001B44C5918B990d0	SanDisk SDSSDXPS	143034400144	960.2 GB
ONLINE	target 15	dc2	raidz2-0	S:0 H:6 T:6	894.3 GiB
	Smart: 34 °C	Smart health, selftest, type:	PASSED	without error	sat,12
Slot 02.00.06	c2:e1:s6	c18t5001B44C5918BD0Bd0	SanDisk SDSSDXPS	143034401035	960.2 GB
ONLINE	target 14	dc2	raidz2-0	S:0 H:3 T:3	894.3 GiB

The values of the list:

Slotnumber	Enclosure	WWN or slot id	Product	Serialnr	capacity1
Disk state	Target nr	Pool	Vdev	iostat messages	capacity2
	Smart temp	Smart info	Smart health	Smart selftest	Smart type
	Slotinfo				

Print out the disk map and the map list and place them on top of your storageserver as a disc and location documentation. It will we very valuable on any kind of problems.

23. Disaster and General Data Security

ZFS is the best available free technology to protect your data. It adresses nearly all problems of conventional filesystems and raid technologies with many disks or large capacity. Unlike checksummed backup technologies like Snapraid it works in realtime and protects your data from the moment when you click save. It can detect every problem due data and metadata checksums in the chain disk-controller-driver > controller > cabling > disk. On access or with a online scrubbing that you should run on a regular base it can repair all bitrot/silent data errors or write/read errors on the fly. ZFS software raid protects against raid problems like bitrot on a Raid-1. A conventional system reads from one or the other part of a mirror but cannot detect if a datablock contains garbage and if it detects that both parts of a mirror are different, it cannot decide, with one is good and which on is bad due the missing data checksums. ZFS detects the faulted part and repairs on the fly.

With Snapshots that are readonly (nobody can modify/ delete them on a share and not even root can modify them locally) you are protected against unwanted file modification or delete. Even malware like those who encrypt all your data continously in the background asking for money to decrypt cannot encrypt snapshot data, not even when running as admin. This is important as you quite often detect the problem too late when even data on backup is encrypted or modified. You need readonly longterm versioning with ZFS snaps to be safe.

With ZFS you can use raid-levels with a unique protection agains disk failures. With Raid-Z2/3 you can use vdevs where 2 or 3 disks can fail per vdev without a dataloss. The statistical chance to loose data due disk failures is then near zero. ZFS Raid is software raid with CopyOnWrite that protects against raid write hole problems. On a powerloss during a write on a conventional Raid 1/5/6 it can happen that datamodification is done on one half of a mirror but not on the other or that data is modified but metadata not updated or in case of a raid 5/6 that a stripeset is written on some disks but not all. ZFS use Copy on Write what means that an atomic write (ex data modification + metadate update) is done completely or discared ex due a crash. So different states of mirrored disks or partly written raid stripes are not possible on ZFS by design.

So for daily use, you are protected agains all sorts of problem?
Yes - nearly. Three problems must be adressed outside ZFS, this is data corruption in RAM, a crash or power outage and a disaster like a sabotage by an employee, human errors, fire, theft or overvoltage due a flash.

23.1. Data Corruption due RAM problems

This is not a ZFS problem but a problem of any sort of data processing. A bitflip in RAM modifies data. This can result in a crash, in a wrong calculation or in modified data during a write or with a checksummed filesystem during a read when processing the checksum. While these problems affects all computer systems, ZFS promises 100% data protection against any sort of problems. Undetected RAM problems/bitflips is quite the only problem that ZFS cannot detect nor repair. The chance of such problems may be acceptable with less RAM like a few hundred Megabyte. A modern workstation or storage server use Gigabytes of RAM. In such a cese a bitflip is not a theoretical chance but a real problem that happens too often. To adress this problem you must use RAM with checksums = ECC Ram, more https://en.wikipedia.org/wiki/ECC_memory

So it is unsafe to use ZFS without ECC?
In the end yes, its unsafe as ZFS cannot protect against all typical computer problems that can happen on daily use without RAM protection. You should simply not buy new systems without ECC, does not matter if its a storage server or a workstation when you process sensitive data. The premium is small and data is too valid.

Another question is: Can I use ZFS on my old systems without ECC?
The answer is more yes than no, especially with RAM in the lower Gigabyte area. If you use any other filesystem, you are affected by the same bitflip problem but there are many other problems where ZFS can help. Write errors due RAM problems are the same with every filesystem. ZFS increases the chance of bitflip problems a little due processing checksums on read to repair bitrot or silent errors what means a higher RAM usage. You can disable checksumming but bitrot on a Multi-TB disk array happens more often than a bitflip on 2-4 GB RAM. So again yes, more advantages than disadvantages without ECC when using ZFS instead a checksum less filesystem without CopyOnWrite but for sure a NoGo for valuable data.

23.2 Powerloss or Crash Problems

Your OS, raid subsystem and your disks offer advanced features to ensure data security during regular operation but what happens on a sudden powerloss or system crash - Are you protected against such problem?

The answer is, it depends. So what happens on a sudden powerloss?
Problems can occur for the file that is currently written, for the raid or filesystem structure or the data validity on a single disk or SSD.

File consistency

The first inspection is, what happens if you save a Word document and power fails:
This will always mean, that this document or the newest state is corrupt as it is only partly on disk.
A solutution are local tmp files that allows Word to revert to a last state. Another protection is an UPS for your desktop, switch and storage server. Mainly you are only protected on the application level like Word. ZFS cannot offer help.

Raid and filesystem consistency

But what happens to the Raid or the filesystem on the storage system during this crash?
The basic problem is that you cannot update data on all disks at the same time. If you write data ex to a Raid-1 the first disk is updated then the next. A power outage can result in a situation where one disk is updated but not the other or that data is updated but not metadata. The same happens with a Raid 5/6 where a raid-stripe is only partly written. Result can be a damaged raid and/or a damaged filesystem structure. This problem is called Write-Whole-Problem (http://www.raid-recovery-guide.com/raid5-write-hole.aspx). A conventional Raid or filesystem cannot protect against such problems as the raid has no control or knwoledge about data content or transaction groups or write atomicity to ensure that a data update MUST include a metadata update or must be done on all disks of a Raid. The solution against this problem is CopyOnWrite what means that you do not inline update data structures but write evetry datablock new. The old state is then available for overwriting unless you do not block with a snap. A write action on ZFS that affects filestructure or Raid consistency is done ompletely or discared completely on a crash.

Result: The CopyOnWrite mechanism protects ZFS Software Raid and the ZFS filesystem against powerloss problems. This is crash insensitivity by design. Traditional filesystem problems require a offline chkdsk run (that can last days on large arrays) after a crash to hopely regain a consistent metadata structure. This is not needed with ZFS as these problems cannot occur. There is no chkdsk on ZFS. All what you need is online scrubbing to repair bitrot problems.

Transaction Consistency

ZFS metadata structures are always consistent. But what happens if your application writes transactions that depends on each other like a financial transactions. A move money means, remove it from one account and THEN add it to another. Or if you use ZFS as a storage for a virtualisation environment. Data is then a filesystem like ext4 or ntfs that is not CopyOnWrite. A valid write must consist of a data update followed by a metadata update. In both cases you need transaction safety under control of the database application or the virtualized OS that writes the data. The key is that you must ensure that after a commit from the storage unit, the data must to be on stable storage and not lost in the disk writeback or ZFS storage cache on a crash. One option would be, disable all write caching but this is a bad idea. Caching is where performance comes from when you connect a very fast unit (CPU/RAM) with a very slow unit (storage). ZFS offers unique read and write caching options that you do not want to disable as you will then fall back to pure disk performance what is a fraction of the overall storage performance with caching. You are in a dilemma now as you need caches for performance and you need a behaviour that every committed write is on stable disk what means uncached sync write.

ZFS can do both: process a fast combined write where you cache all slow and small random writes for a few seconds and write them then as a single fast sequential write and ensure that every single action is on disk.

Sync Write (Log every commited write to a ZIL device)

ZFS always collects small and slow randow writes in a rambased write cache for a few seconds and write them together as a single large sequential write. A commit to a writing application means, yes data is in cache. It does not mean data is on disk. This behaviour is essential to achieve performance. A powerloss can result in a few seconds of lost data. As the writing application has no control this can affect transactions where for exampe the first transaction is on disk after a write and the next dependent transaction is lost in cache on a powerloss.

If you need a transaction safe behaviour where a commit must means, yes data is on disk, you can enforce sync for a filesystem or allow on request of a writing application (sync=always or default) . In such a case you enable a sync write logging for every commited small random write directly to stable storage to a special device called ZIL followed by a regular fast sequential cached write of multiple transactions to the pool .

Basically you have not achieved anything on first view. In contrast, you have fast combined cached writes and additionally uncached commited random writes of every single datablock as both write actions go to the same pool, This is why you sometimes discover that on a slow pool enabling sync reduces your pool performance to 10% of the value that you can achieve with sync=disabled.

Dedicated Slog (ZIL on a separate device)

This is why you can add an Slog to the pool. This is an additional DRAM or Flash based disk or NVMe. If you use an Slog device with powerlosss protection and a much lower latency, much higher iops values than your pool, you can ensure a safe uncached sync write behaviour without such a dramatically reduced performance.

Remember: The ZIL is not a write cache device. This is already offered by ZFS in RAM. Its an additional logdevice that contains all committed data what means that it only need to be able to store about 10s of writes. Even with a single 10G connection, about 8GB is enough. This is why one of the fastest Slog devices, a ZeusRAM has only 8GB of battery buffered DRAM. The Slog device content is only read on a reboot after a crash to redo commited writes. A good newer Slog device is an Intel S/P DC 3700. You do not need to mirror the Slog unless you do not need to ensure performance on a Slog failure as ZFS will otherwise revert performance to the slow onpool ZIL. With current ZFS even a pool import with a missing Slog is possible.

SSD Powerloss problems

In the past, SSDs were mainly used as an L2Arc cache device or an Slog. As a single spindle disk offers only about 100 iops while enterprise SSDs can offer 40000 - 80000 iops under constant steady load. Do not believe the 100000 iops of cheap desktop SSDs as they offer this only for a very short time and performancy degrades then to a few thousand iops while enterprise SSD can hold their performance on a steady constant write load.

While SSDs improves IO performance dramatically over spindels, they come with a serious security problem. As writes can be only done in pages and you need an erase cycle prior rewriting data, the SSD firmware is constanly reorganising data in the background (garbage collection). A power outage can mean a data loss at any time with SSDs, does not matter if there is storage activity or not and not even CopyOnWrite can protect you against. Raid and checksums can help to repair problems on access.

For a professional setup, you should avoid this problem by using SSDs with powerloss protection. All enterprise SSDs offer this as a feature. Some cheaper desktop SSDs offers also powerloss protection but it is sometimes not clear if this affects data written from the OS and the background data from garbage collection

I prefer HGST, Intel or Samsungs in the datacenter editions ex a small Intel S3510 as bootdisk and Intel S3510 for readoptimized pools and the 3610 or 3700/3710 models with better write iops values.

23.3 Disaster Protection

As long as your storage server is working or you can repair the system, you will need nothing but snaps to regain any state of data, current or a previous version based on readonly snaps that you can create nearly without limits, without a delay or a initial space consumption and that you can use for long term storage and accessability. Monthly scrubs will check and repair any sort of bitrot to ensure long term data security.
What you must care about is a real disaster like fire, theft, sabotage, human errors, overvoltage or a massive computer problem that results in a pool loss. To address such problems, you need backups that are not affectes by any of the disaster reasons. Your backup system can also adress availability needs below realtime switchover of storage or services that requires a hot failover system and the ZFS-1 plugin from high-availability.com as you can access your data from a backupsystem at any time. With ZFS replication to can sync data between your active storage and a backup system down to a few minutes even when the system is under heavy load.

Several Levels of disaster protection

Level 1: home use
Buy one or more disks depending to your security concerns like external USB disks or disks that you can insert into your storage server with free disk bays. You can also connect USB disks to your laptop or desktop. You can then access the backup there but without the snapshot security. Then use a sync tools, best is zfs send that allows readonly snapshots on the backup disk. Other options are rsync or robocopy to sync folders from your storage server to such a disk. Remove the disk (can be a mirror in case of ZFS) on a regular base and place it on a safe place and insert the next disk to continue backups. Change this disk or disks with the former or next and re-sync data on a daily, weekly or monthly base.

Level 2: SoHo use
Add another small ZFS storage server and place it on another physical location/ room. This can be a cheap HP Microserver that gives you 4 disk bays where you can insert disks for a backup pool. Replicate data from your primary storage on a hourly or daily base. Keep replication snaps there for previous versions. On problems you can access or continue working on the backup system. From time to time, replace the backup pool with a second one and keep the disks on a safe location outside.

Level 3: Professional use
Care additionally about two backups on different systems and different locations. As availability may be a concern, use a second similar/ same system in a HA config or with async replication that you can run down to some minutes delay. It would be perfect if it offers enough free disk bays to just move the disk from your primary storage on hardware problems to continue work after a pool import with the original data state. Use another backupsystem that allows to continue backup even when your first backup system is out of order or on service. Care about a snapbased longterm backup history with replication snaps on your backup system. This can be different from the snap retention polica on your primary system that you set with autosnap.
As an additional suggestion, use different disks on your backups systems to avoid problems like I had with Seagate Constellations 3 TB. They work for more than a year without problems and nearly simultaniously they fail one by another. Another suggestion is stay on a former OS release on your backup system. Bugs on a newer OS will then not affect backups.

Level 4. Critical data

Remains the basic problem of maintenance and availability
Any server requires maintenance from time to time like security fixes and updates. These updates can affect stability or some functions can give problems. If you want to test these fixes or updates on the same system prior using them on the production machine or if you cannot allow the downtime for update and tests you can consider a Cluster solution. A basic Cluster consists of two identical head nodes with access to common storage (preferable dualpath SAS). One head has access to storage and offer services, the other is a standby failover system where you can do tests and maintenance. A failover can be done on demand or automatically if a head fails with jobs, shares, permissions in sync, see http://www.napp-it.org/doc/downloads/z-raid.pdf

As an extension you can use a twin cluster with two Jbods in an dual expander setup. Thgis would allow a failure of any head or storage box. Use additionally backups. A vCluster in a Box on ESXi is an option.

24. Napp-it Pro

24.1 Differences napp-it Free vs napp-it Pro

Napp-it is a web-based management interface to build a NAS/SAN on top of an enterprise class default Operating System installation like Oracle Solaris or the free Solaris forks like OmniOS or OpenIndiana.
You can install napp-it also on Linux but the Linux release is restricted to ZFS, Snap- and Jobmanagement while the Solarish (Oracle Solaris and the free forks) edition offers enterprise grade features.

napp-it Free v18

Is a ZFS storage server appliance with the most needed functions. As an enduser you can download and install napp-it Free and use it at home or within your company or organisation without a capacity or time limit even commercially. The Free edition includes all functions to run a NAS/SAN with many enterprise features. There are no restrictions regarding OS features and it comes with all basic features to run and manage a ZFS Storage System in a SoHo, lab, office or school environment. Napp-it is not an OpenSource application but all sources are open (Perl) and you are allowed to edit or extend the functionality of napp-it. You are not allowed to distribute napp-it outside your home, office or organisation and you are not allowed to remove license restrictions. If you create your own add-ons or extensions you can distribute your menus and scripts but without napp-it itself example with an online installer.

If you initially install napp-it v18 Free it comes with a 30 day evalkey to evaluate the Pro features that are available in the v18 free edition. Newer Pro features ex Clustering or support for newest ZFS features require an update to a v19 or v20 napp-it edition. Such updates require a valid Pro key. (beside the v19.10 fro noncommercial homeuse only)

napp-it Pro

extends the functionality of napp-it Free. It is the version for professional users. It offers:

- Support with immediate access to bugfixes, newer Pro and developer editions with online update/downgrade of last 5 releases. With a napp-it complete extension you have email support for setup problems.
- Notable improved GUI performance, background agents for system parameters
  You can enable/ disable background acceleration in toplevel menu „acc“ nearby logout
- Edit menu with the option to display the actions of a menu, internal napp-it values and the menu actions.
- Clone menu (create, promote, destroy, care about clones in menu snaps and filesystems)
- Snap mass delete features: mass destroy as a background task (no timeout with many snaps)
- Snap mass delete features: keep at least one snap per day/week/month

If you want to install napp-it on behalf of someone or if you want to bundle napp-it with hardware either under the napp-it brand or as an OEM backend with your own logos and menus or basic messages, you need always a Pro license with the allowance of bundling. This allowance is included when you order a Pro license with a bundling remark when you request a quotation at http://napp-it.org/extensions/quotation_en.html
For larger quantities or distribution ask for conditions.

Napp-it Pro is the mandatory base of additional extensions like ACL/user management, monitoring or remote replication and requires a nonfree license key, either as a subscription or perpetual.

napp-it Pro Complete

For professional users there is napp-it Pro complete with all functions. For special single functions you can aquire the extensions separately. Appliance functions like Appliance Maps or Clustering require the complete edition.

24.2 Complete extension (not working on Linux)

If you want to unlock all Pro features from all extensions you can use the Napp-it Pro complete edition.
Complete adds Appliance menus like Appliance Maps, Appliance Security and Appliance Tuning.
Complete additionally adds email support for setup problems but only for regular commercial licenses.

24.3 ACL and user management extension (Solarish only) includes

- File and Folder ACL settings via Web-GUI
- Care about order of ACL settings (not possible from Windows)
- Allow deny rules (not possible from Windows)
- Share based ACL settings (Permissions on a share itself, not on files or folders)
- Trivial ACL (similar to unix permissions) like owner@, group@ and everyone@
- Control of ACL order (Solaris ACL are order sensitive; Windows cares first about deny then about allow rules)
- Control of ACL inheritance for newly created files and folders
- Local user and groups
- Active Directory user and groups
- human readable ACL set names like
  full_set, modify_set, read_set, create_fileset, create_folderset, owner_default, etc
- Control of ACL-inherit and ACL-mode property
- reset all ACL (recursively) to defaults like modify, roor only, owner only, etc

You can modify permissions and permission related ZFS properties in menu ZFS filesystems under Folder-ACL. Below current permissions and settings you find the controls to modify settings. A resest ACL to everyone@ recursively is available even in the napp-it Free edition.

- User-Quota, Group-Quota settings via Web-UI (napp-it menu ZFS filesystem -> used)
- IDmappings via Web-UI (Menu user)
- Restore all napp-it, user, smbgroup and idmapping settings from backup job data
  (Menu user - restore settings)

Without the extension, you can control file permissions either from console via /usr/bin/chmod or via Windows when you connect a share as root. The main restriction from Windows is that Windows processes all deny rules prior allow rules while Solarish respects the order of rules. If you need deny rules, you should set them on Solarish. The other restriction is that you can remove admin permissions on Windows with the result that an admin has no access unless he or she modifies permissions. On Solarish, root has always access even without a dedicated ACL rule. This makes recursive modifications like reset ACL easier.

24.4 Monitor extension (Solarish only) includes

Realtime monitoring can be enabled/disabled with toplevel menu „mon" nearby logout

- Realtime monitoring of important system parameters via websocket (status lights)
- Shortterm monitoring (last 60s) of cpu%, w%, b% and iops/s of reads, writes and waits
- Longterm monitoring (next release)
- Realtime Update of pages (2 Min after last access)
- Monitoring of LSI SAS2 slots (ZFS WWN disk-id to physical slot mapping)
- Editing features (menus, actions and translations)
- Logs and hashed with internal infos and return values of napp-it commands on a page load
- Health-Status for multiple napp-it appliances (with status lights) via netscan or on appliance groups
- Vnic, Vlan and Link-Aggregation settings
- Disk, SAS2 Bays and Smartvalues are discovered in the background and buffered > 1h to increase GUI performance
- Bridge Management in Menu System > Network Eth (2016.05dev up): Use OmniOS like a 1/10 G switch

24.5 Async highspeed/ network replication (Solarish and Linux)

- Async Replication between appliances (near realtime) with remote appliance management and monitoring
- Based on ZFS send/ receive and snapshots with snap retention policy (hours, days, months, years, numbers)
- After an initial full transfer, only modified datablocks are transferred
- High speed transport via (buffered on Solaris) netcat
- (unencrypted transfer, intended for secure LANs)
- Replication is always pull data. You only need a key on the target server, not on sources

How to setup

- You need a licence key on a target server only (you can request evaluation keys)
- Register the key: copy/paste the whole key-line into menu extension-register, example:
  replicate h:server2 - 20.06.2012::VcqmhqsmVsdcnetqsmVsTTDVsK
- Group your appliances with menu extension -appliance group.
  Klick on ++ add to add members to the group
- Create a replication job with menu Jobs - replicate - create replication job
- Start the job manually or timer based
- After the initial transfer (can last some time), all following transfers are copying only modifies blocks
- You can setup transfers down to every minute (near realtime)

- If one of your server is on an unsecure network like Internet: buld a secure VPN tunnel between appliances
- If you use a firewall with deep inspection: This may block netcat, set a firewall rule to allow port 81 and replication ports

Use it for

Highspeed inhouse replication for backup or near realtime failover/redundancy on secure networks
External replication over VPN links with fixed ip's and a common DNS server (or manual host entries)

How replication works

- On initial run, it creates a source snap jobid...nr_1 and transfers the complete ZFS dataset over a netcat highspeed connection. When the transfer is completed successfully, a target snap jobid.._nr_1 is created. A replication can be recursive (ex whole pool with all filesystems). Use this for a pool transfer only.

- The next replication run is incremental and based on this snap-pair.
  A new source snap jobid.._nr_2 with modified datablocks is created and transfered.
  When the transfer is completed successfully, a new target snap jobid.._nr_2 is created.

  And so on. Only modified datablocks are transfered to provide near realtime syncs when run every few minutes. If a replication fails for whatever reason, you have a higher source than target snapnumber. This does not matter. The source snap is recreated on next incremental run.

  Incremental replications can be recursive. But you should avoid that as zfs send does not care of newly created or modified filesystems. In such a case, an incremental recursive replication run fails with an error.

Difference to other sync methods like rsync

Both rsync and ZFS replication are methods to syncronise data on two servers or filesystems/folders.
While rsync can sync any folders, ZFS replication can sync ZFS filesystems. The main difference is that rsync scans all source and target folders on a run and transfers modified files based on a comparison. Rsync must therefor travers the whole data structures what makes it really slow especially with many small files and rsync cannot sync open files. The pro is that you can run it any time without any special restrictions.

ZFS replication works completely different to rsync. First it is based on ZFS snapshots what allows to sync open files in their last state on disk. Next it creates a pure datastream from the content of a snap that is send to the target filesystem. Therefor performance is not reduced if you transfer many small files and only limited by network and the disk subsystem. You can sync two filesystems with replication even when they are in the Petabyte range down to some minutes delay (near realtime) what would not be possible with rsync.

This is a huge improvement over sync methods like rsync but you must accept the serious restriction that ZFS replication does not compare source and target data. This is not a problem on an initial replication as the whole filesystem is transferred. On the following incremental replications, you must rollback the target filesystem to the exact state of the according source snap that represents the state at last replication run. A new source snap that contains modified datablocks related to the last common snap can then be transferred to update the target filesystem. The only control that replication has is that it can detect if the target base snap for a rollback is not exact identical to the according base snap on the source machine. In such a case replication is cancelled with an snap mismatch error. If something happens during the transfer like zfs send not started correctly, ZFS problems on source or target side, network problems or pool busy, you will get only a „receive: failed to read from stream" error without more details.

In case of problems, check

- Replication requires that all source machines and target machine are grouped

Open menu Extensions > Appliance group and klick on ZFS on an appliance under group members.
If you do not see a listing of the remote filesystems, try to re-add the appliance to the group, optionally delete the group member in /var/web-gui/_log/group on source and target (problem may result from a host renaming) or delete the whole group on all members with Extensions > Appliance group > delete members and rebuild the group.

- Replication requires that the napp-it webserver is running on port 81 on source and target machine. The replication itself is done via a netcat datastream on a port > 50000. These ports must be open.

Check if napp-it is working on both machines. If you have a firewall, check that port 81 and the replication port (see menu Jobs) are open.

- Replication requires an identical ZFS snap-state on source and target

Check if you have a snap-pair with same max jobid_nr_n numbers on source or target. If you do not have identical corresponding snaps, you must restart with an initial sync after a rename of the target file system as a backup. If the following initial replication is successfull you can delete the renamed filesystem.
This means: be careful when you delete replication snaps. Especially the last two snap numbers on target and the same snapnumbers on source should not be destroyed without need. If for whatever reason a snap-pair with same number is not exact identical (can happen ex due follwing snaps with size=0 and results in a snap mismatch or stream error) and you have a former snap pair_nr you can destroy the last target snap. If you start a replication run then, the last snap-pair is used. If you delete a replication job, destroy its repli snaps manually.

- Other problems
Check if you have enough space on source and target (check also reservations and quota settings)
If the receiver and sender is starting but no data is transferred, check for network or routing problems problems.

Check if source or target filesystem is blocking, example with a list snapshots or list filesystem command. A re-boot may help then. If a filesysestem or snap listing lasts longer than 45s this may resuld in a timeout problem.

optionally: Set several replication to a different time to reduce load, optionally reduce max_arc cache to increase free RAM. With MTU 9000 and transfer problems/ stream error, try MTU 1500

Check free memory in menu System > Basic Statistic > Memory (free/freelist should be > 2GB) with many replications or reduce ARC usage example to 80% or 60% in menu System > Appliance tuning (Pro complete)
Free meomry is shown on a source system in menu Jobs > Remote log (napp-it 16.11+)

Replication settings

- First you must create an appliance group, then you can create a replication job in menu Jobs > Replicate > create replication job with following options

Source host:	Can be localhost or any server of your appliance group
Source ZFS	This is the source filesystem that you want to replicate
Enable recursiv	When enabled, other ZFS filesystems below source will be replicated as well You should use this only to replicate a pool once. Recursiv fails if you add or delete a filesystem
Incremental option	The „i" option is the default, the „I" option includes intermediate snaps
Replicate to ZFS	The replicated filesystem will be located below If you want to replicate a filesystem below the root pool folder, select ex video/data as source and backup as target to create backup/video
Replication Type	netcat is the only and the fastest option
SMB/ABE share	On a replicated filesystem with shares, all shares are disabled by default. You must activate this if you want the replicated filesystem shared
Force Job id	If you reinstall your backupsystem without old jobdata available, you can create a job with sam source and target and the old job-id (from snaplist) This allows to continue the replication without a new initial run
Keep	This allows to set a retention policy for your backupsystem, example hours:24,days:32,months:12,years:2 This will keep one snap per hour for today, one snep per day for last 32 days, one snap per month for last 12 months, one snap per year for last two years This setting is needed as an incremential replication does a filesystem rollback to the last common base snap. If you create snaps manually or via the autosnap function, they are lost on next replication run An additional keep parameter is char ex day=12,1000 This will keep any snap with a number ending to 1000
Hold	With hold you can hold snaps based on numbers or days (n or ns) If you set for example 10, this means hold all replication snaps for ten days If you set for example 10s, this means hold always last 10 replication snaps You can combine keep and hold. Both settings are respected. Read it like a „AND" relation or destroy a snap only if the keep setting and the hold setting will allow.
Timetable	Set month, day, hour or minute settings as a trigger Set a status like manual or active Active requires, that Jobs > Autoservice is enabled ex to a trigger every 15 min

Protect snaps
If you want to protect snaps, you can set them manually to „hold"
see http://docs.oracle.com/cd/E19253-01/819-5461/gjdfk/index.html

Debugging

Check loggings on target side in menu Jobs. Either click on „replicate" under Joblog for an overview of the last replication runs. A more detailled view is available when you click under „Last" to the date entry that shows loggings of the very last run.

You can also check loggings on source side in menu „Jobs > Remote"

The communication details are logged in menu „Jobs > Monitor".
Clear the monitor prior a replication run to display only new events.

Sometimes it is helpful to start a replication as root from a console. You can start a job with the command:
perl /var/web-gui/data/napp-it/zfsos/_lib/scripts/job-replicate.pl run_12345 123

Replace 12345 with the job-id, 123 are debug levels when called via console, 1=repli, 2=remote, 3=keep

Sometimes a pool is blocking/ remains busy for whatever reason
Try to export/ import the pool (zpool export -f poolname) . If this fails, reboot

Continue a replication if you reinstall an OS

If you reinstall the OS you must either restore group and job settings. The easiest way is when you have used a napp-it backup job (Menu Jobs) that creates a backup of group and job settings on your first datapool under backup_napp-it. In such a case you can restore the files under /var/web-gui/_logs/jobs and /groups manually.

With napp-it Pro and the ACL extension, you can easily restore all user, group and napp-it settings in menu User > Restore Settings

If you do not have a backup of the job and group settings, first rebuild the group with menu
Extension > Appliance Group > ++ add. Then list the snaps and check for the jobid of the former replication job.

Then create a new replication job with the same source and target filesystem.
Enter the former job-id in the field „Force jobid". You can then continue the replication job without the need of a new initial transfer.

If you must do a new initial transfer, rename the old target filesystem for backup reasons. The start an initial transfer. On success you can delete the old renamed filesystem.

Naming of napp-it replication snaps:

On the source system ex:
mail/data@1404222498_repli_zfs_backup1_nr_528 where

mail/data	is the source filesystem that is replicated
1404222498	is the jobid
backup1	is the hostname where it replicates to
528	is the ongoing number of the snap (to identify identical snaps on source and target)

On the target system ex:
backup/repli_mail/data@1404222498_repli_zfs_backup1_nr_528

backup/repli_mail/data	is the target filesystem of the replication
1404222498	is the jobid
backup1	is the hostname where it replicates to (this host)
528	is the ongoing number of the snap (to identify identical snaps on source and target)

25. ZFS (v)Cluster and Appliance Z-RAID with SSF

Single server vs Cluster

A ZFS Cluster is a configuration with two identical server nodes with access to the same storage, either via the shared disk options of ESXi (Sata vCluster), dualpath SAS or iSCSI with shared access.

A failover between the two server heads can be dona manually in napp-it free. If you add a (Pro feature) cluster-control head, a very fast failover can be done manually or in auto mode via napp-it.

single storage vs dual storage (Z-Raid)

ZFS and Raid-Z is a perfect solution when you want data protection against bitrot with end to end data checksums and a crash resistent CopyOnWrite filesystem with snaps and versioning. ZFS Raid-Z protects against disk failures. ZFS replication adds a ultrafast method for async backups even when files are open.

A basic Cluster allows a failure of a head. If you want to allow a full storage failure you need two independent storage systems, either two SAS Jbod in a dual expander setup or two iSCSI storage server in a network raid-1.

RAID-Z

Traditionally you build a ZFS pool from RAID-Z or Raid-1 vdevs from disks. To be protected against a disaster like a fire or flash, you do backups and snaps for daily access of previous versions to access deleted or modified files. In case of a disaster, you can restore data and re-establish services based the last backup state.

Main Problem: there is a delay between your last data state and the backup state. You can reduce the gap with ZFS Async Replication but the problem remains that backup is never up to date. An additional critical point are open files. As ZFS Replication is based on snaps, the last state of a replication is like a sudden poweroff what means that files (or VMs in a virtualisation environment) may be in a corrupted state on the backup.

Another problem is time to re-establish services like NFS or SMB on a server crash. If you need to be back online in a short time, you use a second backup system that is able and prepared to takeover services based on the last backup state. As on Solarish systems, NFS and SMB are integrated in the OS/Kernel/ZFS with the Windows security identifier SID as an extended ZFS attribute (Solarish SMB), this is really troublefree. Even in a Windows AD environment, you only need to import a pool, takeover the ip of the former server, and your clients can access their data with all AD permission settings intact without any additional settings to care about.

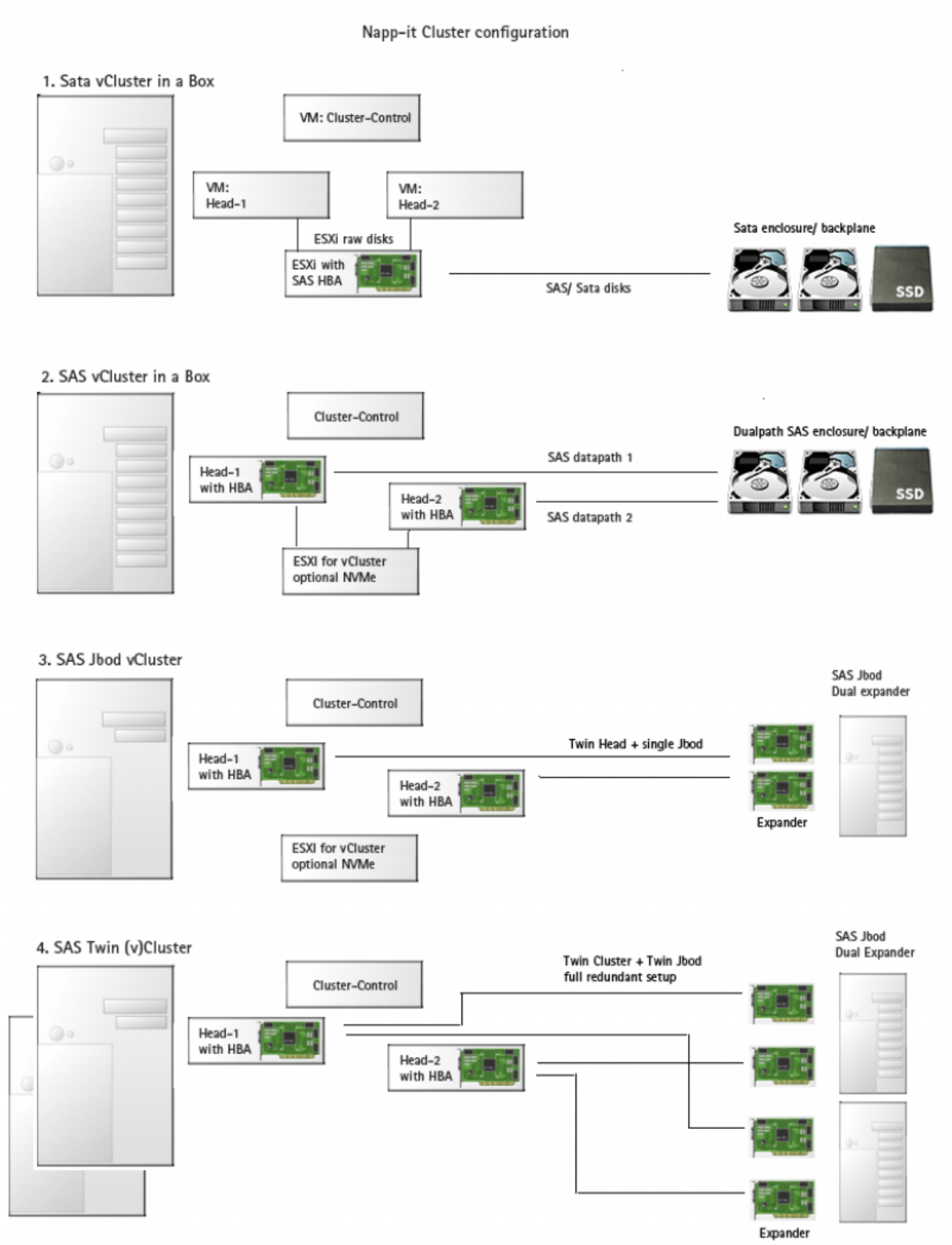
A napp-it ZFS Cluster or Twin Cluster
read: https://www.napp-it.org/doc/downloads/z-raid.pdf

- All-in-One setup of a vCluster or Twin vCluster under ESXi (vCluster in a Box)
- Barebone setup on one or all nodes possible
- Active active or active/passive mode with manual or auto failover
- Stonith support to remove a hanging system from a cluster

- Supports shared ESXi disks (SAS, Sata, NVMe), multipath SAS or iSCSI Luns
- Supports single and dual storage via multipath SAS
- Supports network pool mirroring via iSCSI Luns
- Supports a ZFS pool, user with permissions and job failover
- Failover time on a manual failover around 20s

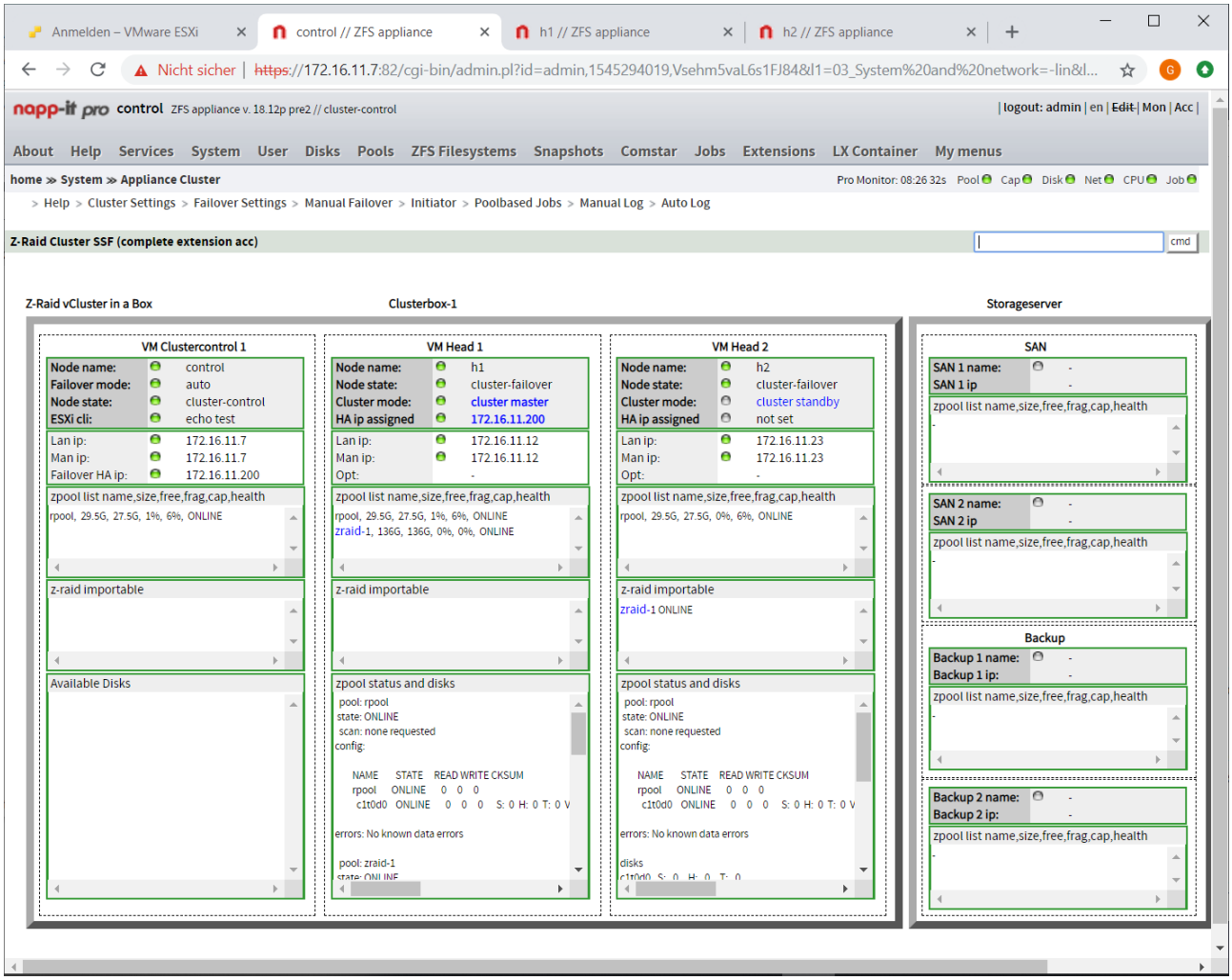
- Fully free setup with manual failover on napp-it free
- Managed manual or auto failover with user and jobs and Ciuster-Control
- requires single napp-it Pro complete on Cluster-Control or a quad cluster or location licence

25.2 Setup options of a ZFS Cluster (barebone) or vCluster (AiO on ESXi)



Cluster Management:

Menu System > Appliance Cluster (final up from napp-it 18.12)



more, see
http://www.napp-it.org/doc/downloads/z-raid.pdf

26. Addendum: About Storage Problems and Solutions

RAID, Backup and ECC

exist in order to mitigate the probability of data loss. While the technical basics are clear, one now has to estimate the data loss endangerment and relevance of all components involved. Included in this is an assessment of which problems can occur and how often they occur in addition to an evaluation of how one can minimize each risk on its own as well as relative to the file system.

What are the problems one has to take care of?

Broken Hardware

Broken hard drives, sectors/flash cells, RAM, controllers, cables, PSU and so on are manageable risks by using real-time RAID, checksums and backups and redundand hardware for example a second PSU.

Statistic Problems and Silent Errors

When prior to data reading one fills a multi-terabyte array only with binary „zeros", one may find binary „ones" when reading the array. If one then leaves the array laying around for some years and reads it again, the number of „ones" will have increased. This is a massive problem of long-term storage and is called „bit rot". The same problem of flipping bits exists in RAM and can lead to program or system crashes and data corruption. In addition, a bad power supply - even bad cables or plugs in the backplane can also cause data corruption. Those problems can be taken care of with real-time RAID, end to end data checksums, scrubs and ECC RAM. If using single-drive backups, with ZFS the parameter „Copies" can be set to „2" resulting in every block of data being written a second time to a different location on the drive.

Transaction Problems

Disks are preferring RAM-cached writes as this offers a much better write performance. Database applications cannot tolerate this. File or dataset locking example for a data warehouse application is only possible with a secure uncached write behaviour. When an operating system is signaled that „a" was written, „a" really has to be written to the stable storage. When „a" and „b" should be written, they either have to be written together at the same time or not at all. A solution for this problems on ZFS are secure SyncWrites with a ZIL device as this allows a fast write behaviour over a RAM cache paired with the safety that a commited write is really on disk.

Time Problems

Accidental deletion, wanting to read older versions of a document, sabotage, Trojans which start to encrypt data secretively in the background and the like are often only identified when it is too late and even backups are already affected.

Those are problems which only can be solved to a certain degree with simple backups. Ideally one could use a read-only versioning with many snapshots that hold a previous data state on the main file system or a second backup system. ZFS Replication would be a real time solution which can keep a backup system up to date down to minutes.

Disasters

Like fire, overvoltage, theft or a defective disk array (meaning more disks fail than the RAID level can handle). Only an external backup can ease those cases – where the external backup should be located at least in a different fire section. In case of very important data it is recommended to have at least two backup systems so that in case of a failure of the first backup system the working second backup system can take over.

Involved System Components

RAID

RAID mainly exists in order to make failure of one or several disks possible without data loss. It may include Self-Healing Features, meaning that corrupted or changed files can be repaired or restored on access. Contrary to RAID-similar backup solutions (like Snapraid and Unraid), RAID protects the data in real time because it distributes, stripes or mirrors data blocks to different drives on every write..
In addition, the sequential performance of a real time RAID scales with the number of drives and the IOPS scale with the number of RAID sub systems (for example doubled performance on a RAID50 or RAID60). This means that RAID boosts not only availability but also performance.

But still there is one major problem with RAID: If more disks fail than allowed by the selected RAID level, the whole array is lost. With ZFS RAID-Z3, up to three disks per vdev can fail without data loss.

Addtionally, on a powerloss or crash a RAID1/5/6 might have corrupted data because the data is written to the different disks of the array sequentially, which leads to different data on each disk for RAID1 or only partly written data stripes on a RAID5 or RAID6. That problem is called the „Write hole" phenomenon in RAID5, RAID6, RAID1, and other arrays. A newer CopyOnWrite (COW) file system with software RAID like ZFS can fix the write hole problem as a data modification is then done on all disks or completely cancelled. Still, an SSD without powerloss protection might introduce a problem nevertheless when the firmware of the controller does garbage collection in the background. When using a RAID1, a system would need to have additional checksums in order to detect bad data and restore the good version of a datablock from the other disk which may be not corrupted.

Another problem is posed by storage systems with a write-cache. On a crash, all data in the cache is lost. The logical conclusion would be to disable the write-cache, but that would result in unbearably slow operations as safe sync write without using a RAM write cache can slow down write performance up to 10%. If you use a write cache some seconds of last writes may get lost on a crash. Older file systems may have corruption problems even without a cache because data may be written to disk but meta data might not have been updated.

More recent CopyOnWrite file systems like ZFS use atomic operations to either update data and meta data as one operation or not at all. Such file systems will never become corrupt after a crash.
One problem remains. If you need filelocking or transactions for a database or store older filesystems for example on a VM datastore for ESXi these databases may become inconsistent or the older filesystems become corrupted after a crash - even when ZFS remains intact as on application or guest OS level you do not have any way to control what goes into the cache and what is written to disk directly.

There is a solution, though. One can use a controller with cache and BBU for Hardwareaid on older file systems. When an operating system receives a confirmation for a write-action, it will really be on stable disk – or in case of a crash after the next system boot. But all in all a RAID controller cache is quite small and slow compared to system RAM and the Write Hole Problem remains so this is not perfect.

In the case of ZFS, the problem was solved with a logging function of the cached data which was committed as "written to disk" to the OS. ZFS always uses a RAM-based write-cache which can buffer several seconds of data to write multiple small random write operations as a single large and fast sequential one. The logging of cached data can be done additionally on a device called ZIL to guarantee a secure write behaviour.

For performance reasons, one may not use the pool as ZIL, but rather an extra logging device called Slog, which is optimized for this kind of operation. The following hardware might be used for this kind of device: 8GB ZeusRAM, NVMe like Intel P-Series (750, 3600, 3700) or an Intel S 3700/3710 SATA SSD that offer powerloss protection, ultra low latency and high and constant write-iops for QD/QueueDepth=1.

Of course all those techniques do not help when a user program, like Word, writes files to the disk. In case of a crash, they will be lost. Only the user program can prevent that kind of failure by creating temp-files.
Versioning and snapshots can help to recover older versions of a file.

ECC

ECC is something like RAID for RAM. It can detect bit errors and repair them from redundancy. Those data errors must not be due damaged RAM in all cases, but might just be a statistical problem which occurs more often if more RAM is in use. Data errors might lead to crashes or data modifications, either during processing or on read/write. Only ECC RAM prevents from those kind of errors. Even newer file systems can not protect against RAM errors. Hence it is recommended to use ECC RAM on all servers and recommended on critical workstations.

Some people claim that ECC RAM is mandatory for newer file systems or else there would be an extreme danger of data loss. Those people base their theory on the idea that hard drives would always deliver correct data and data corruption could only be caused by RAM errors.

While data corruption due RAM problems can occur on ZFS as on every other filesystem, it can repair data corruptions on the chain controller to disk. This and the crash resistent CopyOnWrite behaviour will give you over all a better security level when using ZFS without ECC compared to older filesyytems without ECC.

Personally I strongly recommend using ECC. The possibility of RAM errors increase with RAM size.
With several Gigabyte of RAM ECC is mandatory for a storage server.

Backup

is used to restore a certain data version after a disk or array crash. There are different types of backup, example a rotating multi-level backup which might be rotated each week and data keept up to three weeks.

Since backups are never current and only keep a limited number of versions, backups are only useful as a kind of “disaster backup”, not for quick data recovery. Also, Trojans which encrypt data (for example ransom ware) might stay undiscovered for a long time, leading to backups being encrypted as well.

For every day operations it is preferable to use a file system with versioning. Windows, for example, uses Shadow Copy. But even better than that is a versioning which is based on a CopyOnWrite file system with read-only snapshots (see btrfs, Netapp, WAFL, ReFS or ZFS) on a NAS. Then, even an admin user cannot change the snapshots any more. Ideally, such snapshots should be created on a weekly, daily or even hourly basis - even thousands of them. In order to access snapshots, ZFS offers Clones (which is a writeable file system made from a snapshot). On Windows you can use “previous versions” to access ZFS snaps.

ZFS additionally offers “Rollback” which resets the whole file system to an earlier version from a snapshot. Contrary to a restore from a backup, a rollback is instant.

tl;dr: Solution in order to minimize the probability of a data loss

Next to good hardware and backups, use advanced file systems, like ZFS. They were developed in order to work with large arrays (multiple terabyte up to petabytes) safely. Unlike arrays with ext4 or NTFS which do not implement CopyOnWrite and checksums, they are crash resistent, do not require a long running offline fschk and offer checksums to repair bitrot on access or scrubs without the write hole problem of hardware raid.

As for hardware

- Stay safe, use server class hardware and ECC RAM

As for file systems

- CopyOnWrite: Data changes are written as complete new data to the disk together with their meta data in one (atomic) operation. This leads to a file system which can never be corrupt

- Online Scrubbings in order to repair silent errors and bit rust with the help of checksums. The old data block will be freed in order to be overwritten (unless it is protected by a snapshot)

- Checksums: They are needed on order to detect and repair errors. When using redundant data (like in a RAID), all data will be repaired during file access and data scrubs.

As for the RAID

- Use ZFS software RAID. ZFS is not only a file system, but also offers softwareraid and raid, volume and share management. Especially in combination with CopyOnWrite it offers superior reliability and performance compared to a hardware RAID controller with BBU.

- When using SSDs, prefer those with Powerloss Protection.
  Prefer enterprise class SSDs (Intel S3x or Samsung PM/SM) where you can trust the powerloss feature

As for files

- Use versioning and snapshots. They will be generated in any number without time consumption.
  The space requirement is the number of changed data blocks compared to the previous version.

- Backup through replication, based on snapshots. This is very fast as it only transfers changed data blocks after the initial backup. Open files will also be synced together with the version on the disk.

27. Other manuals and more tuning infos ..

All-In-One (virtualised SAN)_	http://www.napp-it.org/doc/downloads/napp-in-one.pdf
Build Examples	http://www.napp-it.org/doc/downloads/napp-it_build_examples.pdf
SMB 10G Tunings on OSX/ Windows:	http://napp-it.org/doc/downloads/performance_smb2.pdf
Advanced user:	http://www.napp-it.org/doc/downloads/advanced_user.pdf
Tuning:	http://napp-it.org/manuals/tuning_en.html
more:	http://napp-it.org/manuals/index_en.html

- Oracle Solaris 11.4 (commercial OS) www.oracle.com/technetwork/server-storage/solaris11/downloads
  manuals http://www.oracle.com/technetwork/documentation/solaris-11-192991.html

- OpenIndiana Hipster (with a desktop option), community project based on Illumos www.openindiana.org
  Wiki, manuals https://www.openindiana.org/

- OmniOS (free and stable Solaris fork), community project based on Illumos www.omniosce.org

Wiki:	https://github.com/jfqd/OmniOSce-wiki
Downloads:	https://downloads.omniosce.org/media/ or a mirror like http://openzfs.hfg-gmuend.de
Community repo is	https://pkg.omniosce.org/r151022/core/en/index.shtml and
Changelog	https://github.com/omniosorg/omnios-build/blob/r151022/doc/ReleaseNotes.md

OmniOS and OpenIndiana are distributions of the free Solaris fork Illumos.
This fork is based on the last release of OpenSolaris and beside encryption quite identical to Solaris 11 Express

The manuals for Solaris 11 Express can be found under (use download pdf links only)
http://archive.is/snZaS

Update napp-it
You can update napp-it in menu About > Update. This initiates:

- dowload new version ex napp-it_18.12 as .zip and extract to /var/web-gui/data_18.12
- rename current active version /var/web-gui/data to /var/web-gui/data_last
- copy /var/web-gui/data_18.12 to /var/web-gui/data
- restart all napp-it services via /etc/init.d/napp-it restart

If something during an update (web-ui does not come up again):
- at console: use midnight commender to copy /var/web-gui/data_18.12 to /var/web-gui/data
(or use another version as source) and restart all napp-it services via /etc/init.d/napp-it restart

or
Reboot and select the backup bootenvironment that napp-it creates prior an update
ex Bootenvironment pre_18.12. You are now in the state prior the update. Set wanted BE to default.

or
Rerun the online wget installer.
All napp-it settings are kept

If you reinstall the OS completely, restore all napp-it settings in /var/wen-gui/_log/*
either manually or via a backup job and User > Restore (Pro fetaure).

Oprionally re-create all users with same uid/gid