

# DOKUMENTATION

## INSTALLATION OPENSOLARIS ALS BACKUPSERVER ESXI4 HOST

Um die Vorteile zu nutzen, die eine Installation eines ESXi bietet, wird ein NFS Share benötigt auf welchem Backupdaten abgelegt werden können. Falls man auf der Suche nach billigem Festplattenspeicher ist und die Vorteile von ZFS kennt kommt man an OpenSolaris nicht vorbei.

Siehe [http://www.solarisinternals.com/wiki/index.php/ZFS\\_Best\\_Practices\\_Guide#ZFS\\_Administration\\_Considerations](http://www.solarisinternals.com/wiki/index.php/ZFS_Best_Practices_Guide#ZFS_Administration_Considerations) für den Einstieg und <http://www.heise.de/open/OpenSolaris-als-Fileserver--/artikel/119995/0> weiterführend.

ZFS bietet sich an, da ZFS ein sogenanntes scrubbing der Festplatten bietet. Hierbei werden in einem RAID mirror oder raidz die Festplatten auf defekte Sektoren überprüft und gegebenenfalls repariert. Das weitere Handling ist einfach und auch ein Freigeben der Daten und Einbinden in entfernte System klappt problemlos. Für diese Dokumentation wird das Augenmerk auf NFS und SMB gelegt.

Bei der hier vorgestellten Verfahrensweise werden alle Daten auf den verwendeten Festplatten unwiederbringlich GELÖSCHT. Ich übernehme keinerlei Verantwortung für Probleme die hierdurch entstehen. Die Vorgehensweise hat bei mir zum Erfolg geführt.

## INHALT

1	Einführung .....	3
1.1	Auswahl der Hardware .....	3
1.2	Bemerkung .....	3
2	Installation .....	4
2.1	Grafische Installation .....	4
2.2	Format -e .....	4
2.3	Vorgehensweise – Best Practice Mirror rpool .....	5
2.4	Installgrub .....	5
2.5	zpool scrub [Datenpool] .....	5
2.6	Erstellung RAIDz (zpool create) .....	6
2.7	Share NFS .....	6
2.8	Share SMB .....	7
3	Automatisieren .....	8
3.1	Backup Script von Remote .....	8
3.1.1	Plink.exe .....	8
3.1.2	Blat.exe .....	8
3.1.3	ghettoVCB4i.sh .....	8
3.2	Cronjob Opensolaris .....	8
3.3	Fault Report .....	9
4	Anhang .....	11
4.1	zpool get all daten .....	11
4.2	zpool iostat .....	11
4.3	zfs get all daten/nfsshare .....	11
4.4	zfs get all daten/smbshare .....	12
4.5	Enable SSH on ESXi4 .....	13
4.6	Hinweise zum ESXi4 .....	13

# 1 EINFÜHRUNG

## 1.1 Auswahl der Hardware

OpenSolaris ist anspruchsvoll was die Auswahl der Hardware angeht. Der erste Versuch mit einem Supermicro Server scheiterte da OpenSolaris keine Festplatten erkannte. In diesem Fall waren 5 SCSI HDs angebunden. Dies ist dem eingebauten Adaptec 29160 (7902a) zu verdanken. Normalerweise sollte dieser Controller erkannt werden, aber aufgrund von Lizenzstreitigkeiten zwischen den Parteien wurden wohl etliche Adaptec Treiber entfernt.

Als zweiter Server wurde der ältere FSC RX 200 mit 2 GB RAM in Betrieb genommen. Dieser ist mit einem Adaptec HW RAID ausgestattet. Sobald man dieses im BIOS des Controllers abschaltet werden die Platten einzeln erkannt.

Ein Server sollte den Mindestanforderungen genügen, außerdem muss das Motherboard und der HD Controller sorgfältig ausgewählt werden. OpenSolaris braucht mindestens 9 GB für die Systemplatte, da diese ge'mirrored werden soll werden als zwei Platten, möglichst gleicher Bauart (entscheidend ist die Größe, es kann sogar IDE und SATA gemischt werden), benötigt.

Siehe [http://de.wikipedia.org/wiki/ZFS\\_%28Dateisystem%29](http://de.wikipedia.org/wiki/ZFS_%28Dateisystem%29) und <http://dlc.sun.com/osol/docs/content/2009.06/getstart/devdriver.html> bzw. <http://www.sun.com/bigadmin/hcl/data/os/>

Das ist ein wenig tricky, wird aber dann durch ein wundervolles System belohnt ;-)

- Um die optimale Performance rauszuholen sollte ein System ausgewählt werden, welches 64 Bit unterstützt.

### Memory and Swap Space

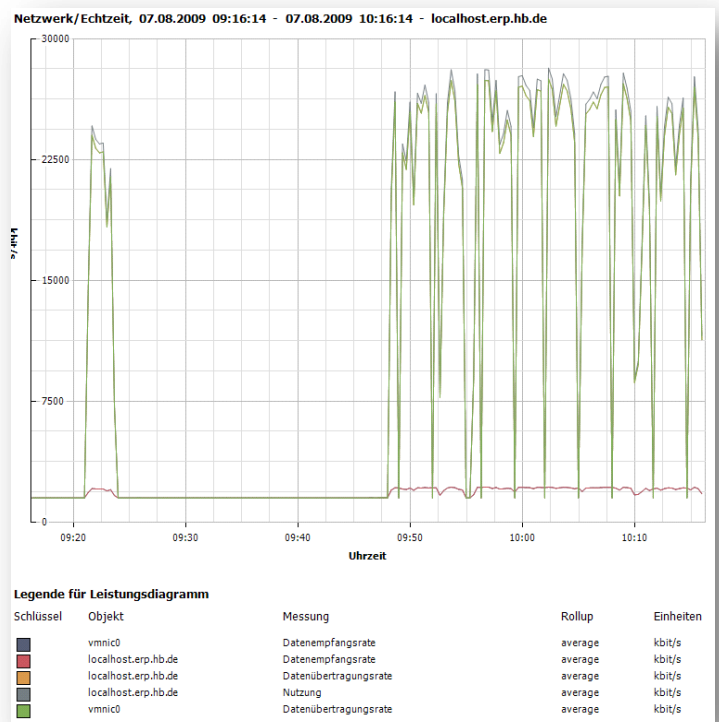
- 1 GB RAM sollte es mindestens sein
- 64 KB werden pro gemountetem ZFS File System verbraucht. Auf Systemen mit tausenden von ZFS File Systemen sollte 1 GB extra pro 10.000 mounted file systems berechnet werden. Desweiteren sollte man sich auf lange Boot Zeiten einstellen.
- Zur weiteren Lektüre empfehle ich [#Memory and Dynamic Reconfiguration Recommendations](#).
- ([http://www.solarisinternals.com/wiki/index.php/ZFS\\_Best\\_Practices\\_Guide#Systems](http://www.solarisinternals.com/wiki/index.php/ZFS_Best_Practices_Guide#Systems))

## 1.2 Bemerkung

So ganz ist das obige Thema noch nicht vertieft, ich werde die Dokumentation bei Bedarf updaten.

Die Testbackups laufen auf einem GBit Netzwerk mit dem Opensolaris NFS Share Datenraten von 26 MB/s. Es lohnt sich also, bedenkt man noch das das unten vorgestellte Script die Festplatten im thin -Modus kopiert verbraucht eine Sicherung von 5 VMs weniger als eine Stunde Zeit.

(Screenshot VMware vSphere-Client / Leistung Netzwerk)



## 2 INSTALLATION

Die jeweils aktuelle Version von OpenSolaris kann unter <http://www.opensolaris.com/get/index.jsp> heruntergeladen werden. Es wurde mit der Version 2009.06 x86 getestet.

### 2.1 Grafische Installation

Das heruntergeladene ISO File wird einfach auf eine CD gebrannt und schon kann man über die Live CD Version OpenSolaris installieren. Hierbei bietet sich an den Gerätemanager kurz zu starten um zu sehen ob alle Controller und Festplatten ordentlich erkannt wurden.

Auf die weitere Installation gehe ich nicht ein, sie ist einfach. Falls die grafische Oberfläche nicht so anspricht steht nach der Installation ein SSH Zugang bereit. Dazu bitte putty benutzen <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Nach der Installation steht der ein Datenpool zur Verfügung welcher ‚rpool‘ heißt.

```
root@opensolaris:~# zpool status -v
pool: rpool
state: ONLINE
scrub: none requested
config:

    NAME                STATE          READ  WRITE CKSUM
    rpool                ONLINE        0     0     0
        c8t0d0s0        ONLINE        0     0     0

errors: No known data errors
```

Um die Systemplatte zu spiegeln muss wie folgt vorgegangen werden:

### 2.2 Format -e

Welche Systemplatten bekannt sind kann mit dem Befehl ‚format -e‘ erkannt werden, nach dieser Ansicht mit CTRL-C abbrechen.

```
root@opensolaris:~# format -e
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. c8t0d0 <DEFAULT cyl 17846 alt 2 hd 255 sec 63>
     /pci@3,0/pci1734,1011@4/sd@0,0
  1. c8t1d0 <DEFAULT cyl 17846 alt 2 hd 255 sec 63>
     /pci@3,0/pci1734,1011@4/sd@1,0
  2. c8t2d0 <SEAGATE-ST3146707LC-7804-136.73GB>
     /pci@3,0/pci1734,1011@4/sd@2,0
  3. c8t3d0 <SEAGATE-ST3146707LC-7804-136.73GB>
     /pci@3,0/pci1734,1011@4/sd@3,0
  4. c8t5d0 <FUJITSU-MAW3147NC-5203-136.73GB>
     /pci@3,0/pci1734,1011@4/sd@5,0
```

Das System in unserem Fall auf der ersten Platte installiert, also c8t0d0. Die zweite Platte (c8t1d0) soll dem rpool als mirror hinzugefügt werden. Hierbei ist zu beachten das OpenSolaris den Platten beim Einbinden in das System ein EFI Label verpasst. Dieses Label allerdings kann man nicht zu einem rpool hinzufügen, es muss durch ein SMI Label ersetzt werden. Hierzu wird ‚format -e‘ und anschließend fdisk benötigt. Noch ein Hinweis: (c8) ist der Controller (t0) ist das Gerät am Controller (die Disknummer), (do) ist die SCSI LUN, bei „normalen“ Festplatten meist 0. Und (s0) ist das Slice, einfacher gesagt die Sub-Partition in der Solaris Partition (<http://docs.sun.com/app/docs/doc/819-3159/fgomr?a=view>) . Diese Infos sind nur für die Installation des rpool nötig, wenn man später Platten zu einem RAIDZ zusammenfasst, dann sollte man ganze Platten verwenden, also z.B. c8t5d0.

## 2.3 Vorgehensweise – Best Practice Mirror rpool

Das Vorgehen in diesem Fall ist nicht immer ganz einfach, die Fehlermeldungen können unterschiedlich sein. Als Best Practice hat sich folgende Vorgehensweise herausgestellt.

- Über `format -e` das EFI Label durch ein SMI Label ersetzen
- Wechseln in `fdisk`
- Löschen der Partition
- Erstellen einer Solaris2 Partition, auf aktiv setzen.
- Nochmals SMI labeln
- Format Befehl verlassen
- Mit dem Befehl `prtvtoc /dev/rdisk/c8t0d0s0|fmthard -s- /dev/rdisk/c8t1d0s0` die Partitionstabelle der ersten HD auf die zweite HD kopieren.
- Dann kann der Mirror mit `zpool attach -f rpool c8t0d0s0 c8t1d0s0` aufgebaut werden.

Das Ergebnis sollte so aussehen (einen Moment braucht das System natürlich um die Spiegelung abzuschliessen).

```
root@opensolaris:~# zpool status

pool: rpool
state: ONLINE
scrub: resilver completed after 0h10m with 0 errors on Tue ...

config:

    NAME                STATE          READ WRITE CKSUM
    rpool                ONLINE         0     0     0
      mirror            ONLINE         0     0     0
        c8t0d0s0        ONLINE         0     0     0
        c8t1d0s0        ONLINE         0     0     0 5.10G resilvered

errors: No known data errors
```

## 2.4 Installgrub

```
installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c8t1d0s0
```

Mit diesem Befehl wird der Masterbootrecord der zweiten Platte geschrieben, damit im Fehlerfalle auch von dieser gebootet werden kann.

## 2.5 zpool scrub [Datenpool]

Mit ‚zpool scrub [Datenpool]‘ können die Festplatten jederzeit auf defekte Sektoren geprüft werden, werden hierbei Fehler festgestellt kopiert Opensolaris die defekten Sektoren in einen „sauberen“ Bereich um.

Beispiel:

```
root@opensolaris:~# zpool scrub daten
root@opensolaris:~# zpool status
pool: daten
state: ONLINE
scrub: scrub in progress for 0h1m, 2,92% done, 0h33m to go
config:

    NAME                STATE          READ WRITE CKSUM
    daten               ONLINE         0     0     0
      raidz1            ONLINE         0     0     0
        c8t2d0          ONLINE         0     0     0
        c8t3d0          ONLINE         0     0     0
        c8t5d0          ONLINE         0     0     0

errors: No known data errors
```

Im obigen Beispiel ist ein scrub rpool daten aktiv braucht noch 33 Minuten zur Vollendung (3 x 147 GB SCSI Platten).

## 2.6 Erstellung RAIDz (zpool create)

Um den Datenpool Daten zu erstellen ist folgender Befehl nötig:

```
zpool create daten raidz c8t2d0 c8t3d0 c8t5d0
```

Wie in Punkt 2.2 zu sehen wurden die restlichen 3 Platten zu einen RAIDz (RAID 5) zusammengeführt, ohne die Angaben von einem Mountpoint wird der Datenpool in das `/` Verzeichnis eingehangen, in diesem Fall also `/daten`. Und schon ist das Datengrab online, klingt unglaublich, ist aber so!

Weiterführende Informationen liefert der Solaris ZFS Administration Guide <http://docs.sun.com/app/docs/doc/819-5461/>

## 2.7 Share NFS

Eine Freigabe bestimmter Daten als NFS funktioniert mit...

```
zfs create daten/nfsshare
```

... wird der Datenpool erstellt, welcher dann mit folgendem Befehl als NFS Share freigegeben wird.

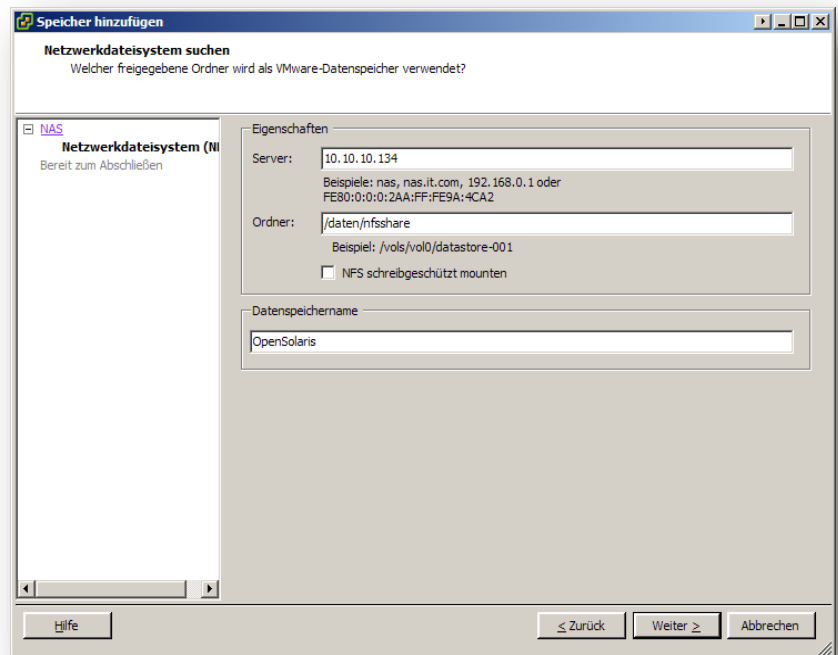
```
zfs set sharenfs=on daten/nfsshare (Datenpool/Verzeichnis)
```

Hierbei wird automatisch das Verzeichnis `/daten/nfsshare` erstellt und per NFS freigegeben, das funktioniert out off the box.

Anschließend sollte man noch Rechte verteilen, da ja alle Dateien root gehören. Ich habe es mir einfach gemacht und einfach einen Vollzugriff auf alle Ordner unterhalb `/daten/nfsshare` erteilt.

```
/user/bin/chmod a+rwX /daten/nfsshare/
```

Anschließend kann das NFS Share in den ESXi Server eingebunden werden.



ZFS bietet eine Fülle an ACLs – weitere Informationen sind hier zu finden - <http://docs.sun.com/app/docs/doc/819-5461/ftyxi?a=view>

Hier noch ein kurzes Beispiel:

```
/usr/bin/ls -dv /... (zeigt die ACLs an)
/usr/bin/chmod A+owner@:rwxpcCosRrWaAdD:fd:allow,group@:rwxp:fd:allow,everyone@::fd:allow /daten (setzt ACLs für /daten)
```

## 2.8 Share SMB

Der Sun-SMB-Server muss in OpenSolaris nachinstalliert werden. Er steckt in den Softwarepaketen SUNWsmbs und SUNWsmbskr, die man über den grafischen Paket-Manager, zu finden im Menü Systemverwaltung, einspielt. Um den SMB-Server im System registrieren, bootet man das System anschließend einfach neu.

Nach dem Neustart muss der Dienst mit `svcadm enable -r smb/server` im System eingetragen und gestartet werden. Mit `svcs smb/server` kann überprüft werden ob der Dienst läuft.

Funktioniert das soweit muss die Datei `/etc/pam.conf` angepasst werden. Einfach die folgende Zeile hinten anhängen.

```
other password required pam_smb_passwd.so.1 nowarn
```

Obige Befehlszeile muss angehängen werden, damit ein Zugriff über SMB möglich ist. Nachdem die `/etc/pam.conf` angepasst wurde, ist es nötig den User, der per SMB zugreifen soll, ein neues Passwort zu erteilen (mit `passwd [username]`, da SMB in die obige Datei schaut, und das Passwort dort abrufen. (<http://docs.sun.com/app/docs/doc/819-2252/pam-smb-passwd-5?a=view>)

Jetzt kann mit ...

```
zfs create -o casesensitivity=mixed daten/smbshare
zfs set sharesmb=on daten/smbshare
```

```
/usr/bin/chmod a+rwx /daten/smbshare/
```

...der Datenpool erstellt und das Verzeichnis per SMB freigegeben werden. Die Option `casesensitivity=mixed` emuliert Windows beim Zugriff auf die Freigabe, d.h. OpenSolaris unterscheidet nicht nach Groß- und Kleinschreibung.

Mit `zfs set sharesmb=name=Alle daten/smbshare`` kann noch der Sharename gesetzt werden, in unserem Fall lautet er nämlich `daten_smbshare`, wie der Befehl

```
root@opensolaris:~# sharemgr show -vp
default nfs=()
zfs
  zfs/daten/nfsshare nfs=()
    /daten/nfsshare
  zfs/daten/smbshare smb=()
    daten_smbshare=/daten/smbshare
```

...zeigt.

Anschliessend ist es möglich z.B. von einem XP-Client per `net use * \\10.10.10.134\daten_smbshare /U:roman` Passwort` auf das SMB Share zuzugreifen.

## 3 AUTOMATISIEREN

### 3.1 Backup Script von Remote

#### 3.1.1 Plink.exe

Es gibt viele Wege das Backup zu automatisieren. Mein Favorit ist die Nutzung von plink.exe (<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>). Plink.exe ist die Kommandozeilenversion von Putty. Einfach eine Batchdatei auf dem Server erstellt, als geplanten Task eingerichtet und schon ist das Backup fertig. Hier ein Beispiel (alles natürlich in einer Zeile):

```
plink -ssh -pw P@ssw0rd root@1.2.3.4 /vmfs/volumes/opensolaris/BackupScript/ghettoVCB4i.sh /vmfs/volumes/opensolaris/BackupScript/vmbackup.one >> ESXi_Backup.log
```

#### 3.1.2 Blat.exe

Um eine eMail Benachrichtigung einzurichten bietet sich Blat (<http://sourceforge.net/projects/blat/files/>) an, folgende Zeile verschickt das oben erstellte ESXi\_Backup.log an die unten angegebenen [korrekte@email.eu](mailto:korrekte@email.eu) über den SMTP Server smtp.server.ip.oder.name.

Es macht weiterhin Sinn das Backuplog bei Ausführung zu überschreiben und anschliessend die Mail zu schicken ;-)

```
blat.exe ESXi_Backup.log -to korrekte@email.de -subject "ESXi BACKUP Log" -server smtp.server.ip.oder.name -f esxi.backup@kbb.eu
```

#### 3.1.3 ghettoVCB4i.sh

In diesem Beispiel wird als root mit dem Passwort P@ssw0rd auf den ESXi Server 10.10.123.123 zugegriffen (Punkt 3.1.1). Es wird das Script ghettoVCB4i.sh ausgeführt, welches auf dem Opensolaris Backup Server im Unterverzeichnis Backupscript liegt. Welche VMs zu sichern sind wird im .txt File vmbackup.one festgelegt. Siehe <http://communities.vmware.com/docs/DOC-8760>. Im Header des Scripts sind einige Angaben zu ändern, diese habe ich **rot** markiert.

```
#!/bin/ash
# Author: william2003[at]gmail[dot]com
# Creation Date: 05/27/2008

# directory that all VM backups should go (e.g. /vmfs/volumes/SAN_LUN1/mybackupdir)
VM_BACKUP_VOLUME=/vmfs/volumes/opensolaris/Backup

# Format output of VMDK backup
# 1 = zeroedthick
# 2 = 2gbsparse
# 3 = thin
# 4 = eagerzeroedthick
VMDK_OUTPUT_FORMAT=3

# Number of backups for a given VM before deleting
VM_BACKUP_ROTATION_COUNT=5
```

### 3.2 Cronjob Opensolaris

Um ein automatisches Scrub'en der verwendeten Platten zu ermöglichen ist es nötig einen cronjob zu erstellen. Als root „Crontab –e“ ausführen, folgende Zeile startet ein scrubing jede Stunde und 40 Minuten.

```
40 * * * * /root/scrub_script
```

Untere Zeile startet das Skript jeden Tag um 03:10 Uhr.

```
10 3 * * * /root/scrub_script
```

Das Script liegt im root Homeverzeichnis und ist für root ausführbar:

```
root@opensolaris:~# ls -l /root/scrub_script
-rwxr--r-- 1 root root 684 2009-08-04 14:18 /root/scrub_script
```

Inhalt des Skriptes, nur MAILRECIPIENT muss angepasst werden. Es verschickt Fehlermeldungen die beim Scrubben auftauchen.

```
#
# this script will go through all pools and scrub them one at a time
#
MAILRECIPIENT=valid@email.eu

ZPOOL=/usr/sbin/zpool
TMPFILE=/tmp/scrub.sh.$$RANDOM

scrub_in_progress() {
    if $ZPOOL status $1 | grep "scrub in progress" >/dev/null; then
        return 0
    else
        return 1
    fi
}

$ZPOOL list | sed 1d | cut -d' ' -f1 | while read pool; do
    $ZPOOL scrub $pool

    while sleep 60; do
        if ! scrub_in_progress $pool; then
            break
        fi
    done

    if ! $ZPOOL status $pool | grep "with 0 errors" >/dev/null; then
        $ZPOOL status $pool >>$TMPFILE
    fi
done

if [ -s $TMPFILE ]; then
    cat $TMPFILE | mailx -s "zpool scrub on `hostname` generated errors" $MAILRECIPIENT
fi

rm -f $TMPFILE
```

Damit Crontab die Änderungen annimmt muss er neu gestartet werden:

```
svcadm restart cron
```

### 3.3 Fault Report

Noch ein Cronjob, der Fehler des Systems an einen eMail Empfänger sendet, Vorgehensweise wie 3.2

```
#!/bin/sh
REPORT=/tmp/faultreport.txt
SYSTEM=$1
rm -f $REPORT
/usr/sbin/fmadm faulty 2>&1 > $REPORT

if test -s $REPORT;

    then

        cat $REPORT | mailx -s "$SYSTEM Fehler Alarm" vaild@email.eu

fi
```

Crontab:

```
0,15,30,45 * * * * /root/fault_management
```

Wird alle 15 Minuten ausgeführt.

## 4 ANHANG

Folgend noch ein paar hilfreiche Hinweise und Befehle um ZFS und den ESXi kennenzulernen.

### 4.1 zpool get all daten

```
root@opensolaris:~# zpool get all daten
NAME    PROPERTY      VALUE      SOURCE
daten   size          408G      -
daten   used          276G      -
daten   available     132G      -
daten   capacity      67%       -
daten   altroot       -          default
daten   health        ONLINE    -
daten   guid          4991702268873899159  default
daten   version       14        default
daten   bootfs        -          default
daten   delegation    on         default
daten   autoreplace   off        default
daten   cachefile     -          default
daten   failmode      wait      default
daten   listsnapshots off        default
```

### 4.2 zpool iostat

```
root@opensolaris:~# zpool iostat
          capacity      operations      bandwidth
pool      used  avail  read  write  read  write
-----  -
daten     279G  129G   99   133  12,1M  13,1M
rpool     4,46G  132G    0    0   11,1K  2,54K
-----  -
```

### 4.3 zfs get all daten/nfsshare

ZFS bietet noch eine Fülle an Optionen die gesetzt werden können, die wichtigsten habe ich **rot** markiert. Mehr unter <http://docs.sun.com/app/docs/doc/817-2271/gazsp?a=view>.

```
root@opensolaris:~# zfs get all daten/nfsshare
NAME                PROPERTY      VALUE      SOURCE
daten/nfsshare     type          filesystem  -
daten/nfsshare     creation      Fr Jul 31 17:03 2009 -
daten/nfsshare     used          187G       -
daten/nfsshare     available     76,3G      -
daten/nfsshare     referenced    187G       -
daten/nfsshare     compressratio 1.00x      -
daten/nfsshare     mounted       yes         -
daten/nfsshare     quota        none        default
daten/nfsshare     reservation none        default
daten/nfsshare     recordsize    128K       default
daten/nfsshare     mountpoint    /daten/nfsshare default
daten/nfsshare     sharenfs      on          local
daten/nfsshare     checksum      on          default
daten/nfsshare     compression off         default
daten/nfsshare     atime         on          default
daten/nfsshare     devices       on          default
daten/nfsshare     exec          on          default
daten/nfsshare     setuid        on          default
daten/nfsshare     readonly      off         default
daten/nfsshare     zoned         off         default
daten/nfsshare     snapdir       hidden      default
daten/nfsshare     aclmode       groupmask   default
daten/nfsshare     aclinherit    passthrough inherited from daten
daten/nfsshare     canmount      on          default
```

```

daten/nfsshare shareiscsi off default
daten/nfsshare xattr on default
daten/nfsshare copies 1 default
daten/nfsshare version 3 -
daten/nfsshare utf8only off -
daten/nfsshare normalization none -
daten/nfsshare casesensitivity sensitive -
daten/nfsshare vscan off default
daten/nfsshare nbmand off default
daten/nfsshare sharesmb off default
daten/nfsshare refquota none default
daten/nfsshare refreservation none default
daten/nfsshare primarycache all default
daten/nfsshare secondarycache all default
daten/nfsshare usedbysnapshots 0 -
daten/nfsshare usedbydataset 187G -
daten/nfsshare usedbychildren 0 -
daten/nfsshare usedbyrefreservation 0 -

```

#### 4.4 zfs get all daten/smbshare

```

root@opensolaris:~# zfs get all daten/smbshare
NAME          PROPERTY      VALUE          SOURCE
daten/smbshare type          filesystem    -
daten/smbshare creation      Fr Jul 31 17:21 2009 -
daten/smbshare used          4,51G        -
daten/smbshare available    75,3G        -
daten/smbshare referenced    4,51G        -
daten/smbshare comprssratio 1.00x        -
daten/smbshare mounted      yes          -
daten/smbshare quota         none         default
daten/smbshare reservation  none         default
daten/smbshare recordsize    128K         default
daten/smbshare mountpoint    /daten/smbshare default
daten/smbshare sharenfs      off          default
daten/smbshare checksum      on           default
daten/smbshare compression  off          default
daten/smbshare atime         on           default
daten/smbshare devices      on           default
daten/smbshare exec         on           default
daten/smbshare setuid       on           default
daten/smbshare readonly     off          default
daten/smbshare zoned         off          default
daten/smbshare snapdir      hidden       default
daten/smbshare aclmode      groupmask   default
daten/smbshare aclinherit    passthrough inherited from daten
daten/smbshare canmount     on           default
daten/smbshare shareiscsi   off          default
daten/smbshare xattr        on           default
daten/smbshare copies      1           default
daten/smbshare version      3           -
daten/smbshare utf8only     off          -
daten/smbshare normalization none         -
daten/smbshare casesensitivity sensitive    -
daten/smbshare vscan        off          default
daten/smbshare nbmand       off          default
daten/smbshare sharesmb     on           local
daten/smbshare refquota     none         default
daten/smbshare refreservation none         default
daten/smbshare primarycache all          default
daten/smbshare secondarycache all          default
daten/smbshare usedbysnapshots 0           -
daten/smbshare usedbydataset 4,51G      -
daten/smbshare usedbychildren 0           -
daten/smbshare usedbyrefreservation 0          -

```

#### 4.5 Enable SSH on ESXi4

Um SSH für den ESXi zu aktivieren muss folgende Prozedur durchgeführt werden.

- Auf der Konsole ESXi ALT-F1 – unsupported eintippen (wie die Vorgehensweise schon vermuten lässt wird das nicht von VMware supported ;-)
- Mit Root Passwort bestätigen
- vi /etc/inetd.conf
- die Zeile suchen in der #ssh zu finden ist und „ent“kommentieren (# entfernen), speichern (ESC :wq)
- ps | grep inetd (die Prozess ID merken)
- kill -HUP [Prozess ID]

Und schon kann per Putty SSH root Login auf den ESXi zugegriffen werden.

#### 4.6 Hinweise zum ESXi4

Nach einem Reboot vergisst der ESXi Server das Root Passwort und die eingebundenen VMs (sind leicht einzubinden / Datastore browsen und rechtecklick auf die entsprechende .vmx Datei und mit „zur Bestandsliste hinzufügen“ wieder aufnehmen).

Habe ich noch was vergessen?