

history

jose@avtp239:~\$ ssh root@192.168.1.76

root@192.168.1.76's password:

Linux nas 5.4.78-2-pve #1 SMP PVE 5.4.78-2 (Thu, 03 Dec 2020 14:26:17 +0100)
x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Last login: Wed Dec 9 20:20:33 2020 from 192.168.1.86

root@nas:~# apt-get install sdparm

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following NEW packages will be installed:

sdparm

0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.

Need to get 131 kB of archives.

After this operation, 427 kB of additional disk space will be used.

Get:1 http://ftp.caliu.cat/debian buster/main amd64 sdparm amd64 1.10-1 [131
kB]

Fetched 131 kB in 0s (385 kB/s)

Selecting previously unselected package sdparm.

(Reading database ... 74895 files and directories currently installed.)

Preparing to unpack .../sdparm_1.10-1_amd64.deb ...

Unpacking sdparm (1.10-1) ...

Setting up sdparm (1.10-1) ...

Processing triggers for man-db (2.8.5-2) ...

root@nas:~# sdparm /dev/sda

/dev/sda: ATA ST8000DM004-2CX1 0001

Read write error recovery mode page:

AWRE 1 [cha: n, def: 1]

ARRE 0 [cha: n, def: 0]

PER 0 [cha: n, def: 0]

Caching (SBC) mode page:

IC 0 [cha: n, def: 0]

WCE 0 [cha: y, def: 0]

RCD 0 [cha: n, def: 0]

Control mode page:

TST 0 [cha: n, def: 0]

SWP 0 [cha: n, def: 0]

root@nas:~# apt-get install smartctl

Reading package lists... Done

Building dependency tree

Reading state information... Done

Package smartctl is not available, but is referred to by another package.

This may mean that the package is missing, has been obsoleted, or
is only available from another source

```

E: Package 'smartctl' has no installation candidate
root@nas:~# apt-get install smartmontools
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  gsmartcontrol smart-notifier
The following NEW packages will be installed:
  smartmontools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 539 kB of archives.
After this operation, 2,077 kB of additional disk space will be used.
Get:1 http://download.proxmox.com/debian/pve buster/pve-no-subscription
amd64 smartmontools amd64 7.1-pve2 [539 kB]
Fetched 539 kB in 1s (611 kB/s)
Selecting previously unselected package smartmontools.
(Reading database ... 74908 files and directories currently installed.)
Preparing to unpack .../smartmontools_7.1-pve2_amd64.deb ...
Unpacking smartmontools (7.1-pve2) ...
Setting up smartmontools (7.1-pve2) ...
Created symlink /etc/systemd/system/smartd.service →
/lib/systemd/system/smartmontools.service.
Created symlink /etc/systemd/system/multi-
user.target.wants/smartmontools.service →
/lib/systemd/system/smartmontools.service.
Processing triggers for man-db (2.8.5-2) ...
Processing triggers for systemd (241-7~deb10u5) ...
root@nas:~# smartctltop
-bash: smartctltop: command not found
root@nas:~# top

```

```

top - 20:27:47 up 27 min,  1 user,  load average: 5.50, 5.40, 3.03
Tasks: 230 total,  1 running, 229 sleeping,  0 stopped,  0 zombie
%Cpu(s):  0.3 us,  1.2 sy,  0.0 ni, 21.8 id, 76.5 wa,  0.0 hi,  0.2 si,  0.0
st

```

```

MiB Mem : 15964.2 total,  6414.7 free,  8866.3 used,  683.1 buff/cache
MiB Swap:  2048.0 total,  2048.0 free,    0.0 used.  6422.2 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
783	root	1	-19	0	0	0	D	1.0	0.0	0:11.80	
z_wr_iss											
138	root	0	-20	0	0	0	I	0.7	0.0	0:05.61	
kworker/0:1H-kblockd											
33540	root	20	0	975916	209440	740	D	0.7	1.3	0:03.20	fio
791	root	0	-20	0	0	0	S	0.3	0.0	0:00.51	
z_wr_int											
792	root	0	-20	0	0	0	S	0.3	0.0	0:00.50	
z_wr_int											
35672	root	20	0	13832	2868	2136	R	0.3	0.0	0:00.03	top
1	root	20	0	23672	8184	5152	S	0.0	0.1	0:04.70	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.25	

kthreadd											
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	
rcu_par_gp											
5	root	20	0	0	0	0	I	0.0	0.0	0:00.15	
kworker/0:0-events											
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	
kworker/0:0H-kblockd											
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	
mm_percpu_wq											
9	root	20	0	0	0	0	S	0.0	0.0	0:00.10	
ksoftirqd/0											
10	root	20	0	0	0	0	I	0.0	0.0	0:00.33	
rcu_sched											
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.02	
migration/0											
12	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	
idle_inject/0											
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	
16	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
idle_inject/1											
17	root	rt	0	0	0	0	S	0.0	0.0	0:00.19	
migration/1											
18	root	20	0	0	0	0	S	0.0	0.0	0:00.08	
ksoftirqd/1											
20	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	
kworker/1:0H-kblockd											
21	root	20	0	0	0	0	S	0.0	0.0	0:00.00	
kdevtmpfs											
22	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
23	root	20	0	0	0	0	S	0.0	0.0	0:00.00	
rcu_tasks_kthre											
24	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kauditd
28	root	20	0	0	0	0	S	0.0	0.0	0:00.06	
khungtaskd											
29	root	20	0	0	0	0	S	0.0	0.0	0:00.00	
oom_reaper											
30	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	
writeback											
31	root	20	0	0	0	0	S	0.0	0.0	0:00.00	
kcompactd0											
32	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
33	root	39	19	0	0	0	S	0.0	0.0	0:00.00	
khugepaged											
79	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	
kintegrityd											
80	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kblockd
81	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	
blkcg_punt_bio											
82	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	

```

tpm_dev_wq
 83 root      0 -20      0      0      0 I    0.0    0.0    0:00.00 ata_sff
 84 root      0 -20      0      0      0 I    0.0    0.0    0:00.00 md
 85 root      0 -20      0      0      0 I    0.0    0.0    0:00.00 edac-
poller
root@nas:~# ps -ef | grep fio
root      33540      1  0 20:19 ?          00:00:03 fio --size=16G --name=create
--filename=fio_file --bs=1M --nrfiles=1 --direct=0 --sync=0 --randrepeat=0 -
-rw=write --refill_buffers --end_fsync=1 --iodepth=200 --ioengine=libaio --
fallocate=none
root      35674 34377  0 20:27 pts/3      00:00:00 grep fio
root@nas:~# ps -ef | grep fio
root      33540      1  0 20:19 ?          00:00:03 fio --size=16G --name=create
--filename=fio_file --bs=1M --nrfiles=1 --direct=0 --sync=0 --randrepeat=0 -
-rw=write --refill_buffers --end_fsync=1 --iodepth=200 --ioengine=libaio --
fallocate=none
root      35676 34377  0 20:28 pts/3      00:00:00 grep fio
root@nas:~# kill -9 33540
root@nas:~# ps -ef | grep fio
root      35703 34377  0 20:28 pts/3      00:00:00 grep fio
root@nas:~# smartctl ^C
root@nas:~# sdparm --set=WCE --save /dev/sda
/dev/sda: ATA          ST8000DM004-2CX1  0001
change_mode_page: mode page indicates it is not savable but
'--save' option given (try without it)
root@nas:~# sdparm /dev/sda
/dev/sda: ATA          ST8000DM004-2CX1  0001
Read write error recovery mode page:
  AWRE          1 [cha: n, def:  1]
  ARRE          0 [cha: n, def:  0]
  PER           0 [cha: n, def:  0]
Caching (SBC) mode page:
  IC            0 [cha: n, def:  0]
  WCE           0 [cha: y, def:  0]
  RCD           0 [cha: n, def:  0]
Control mode page:
  TST           0 [cha: n, def:  0]
  SWP           0 [cha: n, def:  0]
root@nas:~# sdparm --set=WCE /dev/sda
/dev/sda: ATA          ST8000DM004-2CX1  0001
root@nas:~# sdparm /dev/sda
/dev/sda: ATA          ST8000DM004-2CX1  0001
Read write error recovery mode page:
  AWRE          1 [cha: n, def:  1]
  ARRE          0 [cha: n, def:  0]
  PER           0 [cha: n, def:  0]
Caching (SBC) mode page:
  IC            0 [cha: n, def:  0]
  WCE           1 [cha: y, def:  1]
  RCD           0 [cha: n, def:  0]
Control mode page:

```

```

TST          0 [cha: n, def: 0]
SWP          0 [cha: n, def: 0]
root@nas:~# sdparm --set=WCE /dev/sdb
/dev/sdb: ATA      ST8000DM004-2CX1  0001
root@nas:~# sdparm --set=WCE /dev/sdc
/dev/sdc: ATA      ST8000DM004-2CX1  0001
^[[Aroot@nas:~# sdparm --set=WCE /dev/sdd
/dev/sdd: ATA      ST8000DM004-2CX1  0001
root@nas:~# sdparm /dev/sde
/dev/sde: ATA      KINGSTON SA400S3  0004
Read write error recovery mode page:
AWRE         1 [cha: n, def: 1]
ARRE         0 [cha: n, def: 0]
PER          0 [cha: n, def: 0]
Caching (SBC) mode page:
IC           0 [cha: n, def: 0]
WCE          0 [cha: y, def: 0]
RCD          0 [cha: n, def: 0]
Control mode page:
TST          0 [cha: n, def: 0]
SWP          0 [cha: n, def: 0]
root@nas:~# sdparm --set=WCE /dev/sde
/dev/sde: ATA      KINGSTON SA400S3  0004
root@nas:~# reboot
Connection to 192.168.1.76 closed by remote host.
Connection to 192.168.1.76 closed.
jose@avtp239:~$ ssh root@192.168.1.76
root@192.168.1.76's password:
Linux nas 5.4.78-2-pve #1 SMP PVE 5.4.78-2 (Thu, 03 Dec 2020 14:26:17 +0100)
x86_64

```

The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

Last login: Wed Dec 9 20:21:55 2020 from 192.168.1.86

root@nas:~#

root@nas:~#

root@nas:~#

root@nas:~# sdparm

one or more device names required

```

Usage: sdparm [--all] [--dbd] [--flexible] [--get=STR] [--hex] [--long]
           [--num-desc] [--page=PG[,SPG]] [--quiet] [--readonly]
           [--six] [--transport=TN] [--vendor=VN] [--verbose]
           DEVICE [DEVICE...]

```

```

sdparm [--clear=STR] [--defaults] [--dummy] [--flexible]
       [--page=PG[,SPG]] [--quiet] [--readonly] [--save] [--set=STR]
       [--six] [--transport=TN] [--vendor=VN] [--verbose]
       DEVICE [DEVICE...]

```

```

where mode page read (1st usage) and change (2nd usage) options are:
--all | -a          list all known fields for given DEVICE
--clear=STR | -c STR  clear (zero) field value(s)
--dbd | -B          set DBD bit in mode sense cdb
--defaults | -D      set a mode page to its default values
--dummy | -d         don't write back modified mode page
--flexible | -f       compensate for common errors, relax some checks
--get=STR | -g STR   get (fetch) field value(s)
--hex | -H           output in hex rather than name/value pairs
--long | -l          add description to field output
--num-desc | -n       report number of mode page descriptors
--page=PG[,SPG] | -p PG[,SPG]  page (and optionally subpage) number
                        [or abbrev] to output, change or enumerate
--quiet | -q         suppress DEVICE vendor/product/revision string
line
--readonly | -r      force read-only open of DEVICE (def: depends
                        on operation). Mainly for ATA disks
--save | -S          place mode changes in saved page as well
--set=STR | -s STR   set field value(s)
--six | -6           use 6 byte SCSI mode cdb (def: 10 byte)
--transport=TN | -t TN  transport protocol number [or abbrev]
--vendor=VN | -M VN    vendor (manufacturer) number [or abbrev]
--verbose | -v         increase verbosity

```

View or change SCSI mode page fields (e.g. of a disk or CD/DVD drive). STR can be <acronym>[=val] or <start_byte>:<start_bit>:<num_bits>[=val]. Use '-h' or '--help' twice for help on other usages including executing some simple commands, reading and decoding VPD pages, enumerating internal tables of mode and VPD pages, and decoding response data supplied in a file or stdin (rather than from a DEVICE).

```

root@nas:~# sdparm /dev/sda
/dev/sda: ATA          ST8000DM004-2CX1  0001

```

Read write error recovery mode page:

```

AWRE      1 [cha: n, def: 1]
ARRE      0 [cha: n, def: 0]
PER       0 [cha: n, def: 0]

```

Caching (SBC) mode page:

```

IC        0 [cha: n, def: 0]
WCE       0 [cha: y, def: 0]
RCD       0 [cha: n, def: 0]

```

Control mode page:

```

TST       0 [cha: n, def: 0]
SWP       0 [cha: n, def: 0]

```

```

root@nas:~# ls

```

```

chroot.sh  deploy.sh  deploy.sh.old  prova.sh  test.sh

```

```

root@nas:~# cd

```

```

(failed reverse-i-search)`fip': ps -ef | grep ^Co

```

```

(reverse-i-search)`fio': ps -ef | grep ^Co

```

```

root@nas:~# history

```

```

6  ls -al /target

```

```

7  mkdir /target/mnt /target/dev /target/sys /target/proc

```

```
8  ls -al /target
9  rm -rf /target/*
10 ls -al /target
11 history | less
12 nano deploy.sh
13 history | less
14 cat /etc/hosts
15 cat /etc/fstab
16 history | less
17 nano deploy.sh
18 nano /etc/systemd/system/zfs-import-bpool.service
19 ls -al /etc/network/interfaces
20 cat /etc/network/interfaces
21 ip link
22 ip link | grep "state UP"
23 ip link | grep "state UP" | cut -d ':' -f 2
24 ip link | grep "state UP" | cut -d ':' -f 2 | tr -d ' '
25 ip a
26 ip a | grep "scope global" | cut -d ' ' -f 6
27 ip a | grep "scope global" | cut -d ' ' -f 1
28 ip a | grep "scope global" | cut -d ' ' -f 2
29 ip a | grep "scope global" | cut -d ' ' -f 11
30 zpool status
31 zpool destroy rpool bpool
32 zpool destroy rpool
33 zpool destroy bpool
34 nano deploy.sh
35 ./deploy.sh
36 ls -al /target
37 ls -al /target/boot
38 ls /sys/class/net
39 ls -l /sys/class/net
40 history
41 ip a | grep "scope global" | cut -d ' ' -f 11
42 ip a eno1
43 ip a dev eno1
44 ip a help
45 ip a show dev eno1
46 ip a show dev eno1 | grep "inet "
47 for iface in $(ls -l /sys/class/net); do ip a show dev iface | grep
"inet ";done
48 for iface in $(ls -l /sys/class/net); do ip a show dev $iface | grep
"inet ";done
49 for iface in $(ls -l /sys/class/net); do ip a show dev $iface | grep
"scope global";done
50 nano deploy.sh
51 history
52 nano deploy.sh
53 for iface in $(ls -l /sys/class/net); do ip a show dev $iface | grep
"scope global";done|head -n 1
54 for iface in $(ls -l /sys/class/net); do if [ -n $(ip a show dev
```

```
$iface | grep "scope global" ] then IFACE=$iface; exit;fi ;done; echo $IFACE
55 for iface in $(ls -l /sys/class/net); do if [ -n $(ip a show dev
$iface | grep "scope global") ] then IFACE=$iface; exit;fi ;done; echo
$IFACE
56 for iface in $(ls -l /sys/class/net); do if [ -n $(ip a show dev
$iface | grep "scope global") ]; then IFACE=$iface; exit;fi ;done; echo
$IFACE
57 for iface in $(ls -l /sys/class/net); do if [ -n $(ip a show dev
$iface | grep "scope global") ]; then IFACE=$iface;fi ;done; echo $IFACE
58 for iface in $(ls -l /sys/class/net); do if [ -n $(ip a show dev
$iface | grep "scope global") ]; then IFACE=$iface;echo $IFACE ;fi ;done
59 for iface in $(ls -l /sys/class/net); do ip a show dev $iface | grep
"scope global" IFACE=$iface;echo $IFACE ;fi ;done
60 for iface in $(ls -l /sys/class/net); do ip a show dev $iface | grep
"scope global"; IFACE=$iface;echo $IFACE ;fi ;done
61 for iface in $(ls -l /sys/class/net); do ip a show dev $iface | grep
"scope global"; IFACE=$iface;echo $IFACE ;done
62 for iface in $(ls -l /sys/class/net); do ip a show dev $iface | grep
' scope global '; IFACE=$iface;echo $IFACE ;done
63 for iface in $(ls -l /sys/class/net); do if [ -n "$(ip a show dev
$iface | grep ' scope global ')" ];then IFACE=$iface;echo $IFACE ;fi ;done
64 for iface in $(ls -l /sys/class/net); do if [ -n "$(ip a show dev
$iface | grep ' scope global ')" ];then IFACE=$iface; exit; fi ;done
65 nano prova.sh
66 ./prova.sh
67 nano prova.sh
68 ./prova.sh
69 cat prova.sh
70 nano deploy.sh
71 nano prova.sh
72 ./prova.sh
73 ls -al /target/etc/network/interfaces.d/enol
74 cat /target/etc/network/interfaces.d/enol
75 nano deploy.sh
76 nano prova.sh
77 ./prova.sh
78 cat /target/etc/network/interfaces.d/enol
79 ./prova.sh
80 nano prova.sh
81 nano /target/root/chroot.sh
82 nano deploy.sh
83 nano /target/root/chroot.sh
84 ./prova.sh
85 whereis bash
86 nano prova.sh
87 umount -l /target/dev
88 umount -l /target/sys
89 umount -l /target/proc
90 mount
91 ./prova.sh
92 ls
```



```
93 exit
94 ls
95 zpool status
96 zpool import
97 nano deploys.sh
98 nano deploy.sh
99 nano prova.sh
100 nano prova.sh
101 nano deploy.sh
102 nano prova.sh
103 ls
104 ./deploy.sh
105 zpool destroy bpool
106 zpool destroy rpool
107 zpool status
108 nano deploy.sh
109 ./deploy.sh
110 nano deploy.sh
111 cat .prova.sh
112 cat prova.sh
113 nano prova.sh
114 nano deploy.sh
115 nano prova.sh
116 lsblk -o UUID
117 lsblk -o WWN
118 declare -a BOOTSSD;BOOTSSD+="0x50026b7682efdcf4";echo ${BOOTSSD[@]}
119 declare -a
BOOTSSD;BOOTSSD+="0x50026b7682efdcf4";BOOTSSD+="0x50026b7682efdcf4";echo
${BOOTSSD[@]}
120 unset BOOTSSD
121 declare -a
BOOTSSD;BOOTSSD+="0x50026b7682efdcf4";BOOTSSD+="0x50026b7682efdcf4";echo
${BOOTSSD[@]}
122 nano prova.sh
123 tftp
124 ping 1.1.1.1
125 ls -al
126 cat prova.sh
127 nano deploy.sh
128 cat /target/etc/default/grub
129 sed -i
's/GRUB_CMDLINE_LINUX=""/GRUB_CMDLINE_LINUX="root=ZFS=rpool/R00T/pve"/g'
/target/default/grub
130 sed -i
's/GRUB_CMDLINE_LINUX=""/GRUB_CMDLINE_LINUX="root=ZFS=rpool/R00T/pve"/g'
/target/etc/default/grub
131 sed
's#GRUB_CMDLINE_LINUX=""#GRUB_CMDLINE_LINUX="root=ZFS=rpool/R00T/pve"#'
/target/etc/default/grub
132 sed
's#GRUB_CMDLINE_LINUX=""#GRUB_CMDLINE_LINUX="root=ZFS=rpool/R00T/pve"#'
```

```
/target/etc/default/grub
133 sed 's#_LINUX_DEFAULT="quiet"#_LINUX_DEFAULT=""#'
/target/etc/default/grub
134 sed 's#_LINUX_DEFAULT="quiet"#_LINUX_DEFAULT=""#'
/target/etc/default/grub
135 sed -i
's#GRUB_CMDLINE_LINUX=""#GRUB_CMDLINE_LINUX="root=ZFS=rpool/ROOT/pve"#'
/target/etc/default/grub
136 sed -i 's#_LINUX_DEFAULT="quiet"#_LINUX_DEFAULT=""#'
/target/etc/default/grub
137 cat /target/etc/default/grub
138 history
139 nano deploy.sh
140 mount
141 zpool destror rpool
142 zpool destroy bpool
143 zpool destroy rpool
144 ./deploy.sh
145 ls -al
146 molunt
147 mount
148 umount -l /target/dev
149 umount -l /target/proc
150 umount -l /target/sys
151 ls -al /target/root
152 cat /target/root/chroot.sh
153 mount --rbind /dev /target/dev
154 mount --rbind /sys /target/sys
155 mount --rbind /proc /target/proc
156 chroot /bin/bash /root/chroot.sh --login
157 chroot /bin/bash /root/chroot.sh
158 cat deploy.sh
159 chroot /target /bin/bash /root/chroot.sh --login
160 zfs list -t snapshots
161 zfs list -t snapshot
162 exit
163 zpool status
164 reboot
165 nano deploy.sh
166 cat /etc/default/grub
167 ./deploy.sh
168 mount
169 reboot
170 nano deploy.sh
171 zpool status
172 ./deploy.sh
173 zpool status
174 mount -o bind /dev /target/dev
175 mount -o bind /sys /target/sys
176 mount -o bind /proc /target/proc
177 chroot /target /bin/bash --login
```

```
178 umount -l /target/dev
179 umount -l /target/sys
180 umount -l /target/proc
181 zpool status
182 zfs list -t snapshot
183 zfs list
184 ls -al /target/boot
185 zpool export -a
186 df -h
187 mount
188 history
189 reboot
190 nano deploy.sh
191 zpool status
192 ./deploy.sh
193 zpool status
194 echo $HOST
195 echo $HOSTNAME
196 nano deploy.sh
197 ./deploy.sh
198 zpool status
199 nano deploy.sh
200 ./deploy.sh
201 reboot
202 history |less
203 nano deploy.sh
204 nano prova.sh
205 ./prova.sh
206 ./prova.sh pve
207 cat prova.sh
208 nano prova.sh
209 ./prova.sh
210 echo "$NAME"
211 echo "$NAME"; NAME="{ $NAME:-pve}"; echo "$NAME"
212 echo "$NAME"; NAME={ $NAME:-pve}; echo "$NAME"
213 unset NAME
214 echo "$NAME"; NAME={ $NAME:-pve}; echo "$NAME"
215 echo "$NAME"; NAME="{NAME:-pve}"; echo "$NAME"
216 read -p "Enter your gender (just press ENTER to not tell us): "
GENDER;echo "Your gender is ${GENDER:-a secret}."
217 nano prova.sh
218 ./prova.sh
219 ./prova.sh pepep
220 ./prova.sh pep selga
221 cat prova.sh
222 nano deploy.sh
223 ./deploy.sh
224 zpool status
225 zpool export -a
226 zpool status
227 reboot
```

```
228  reboot
229  zpool status
230  cp deploy.sh deploy.sh.old
231  nano deploy.sh
232  sgdisk --help
233  sgdisk -L
234  fdisk /dev/sda
235  sgdisk -L | grep -i RAID
236  sgdisk -L | grep -i Linux
237  nano deploy.sh
238  blkid | grep "{B00TSSD[0]}-part3
239  blkid | grep "{B00TSSD[0]}-part3"
240  blkid | grep "-part3"
241  blkid
242  blkid /dev/sda3
243  nano deploy.sh
244  zpool status
245  ./deploy.sh
246  nano deploy.sh
247  ./deploy.sh
248  nano deploy.sh
249  ./deploy.sh
250  blkid
251  lsblk -o NAME,FSTYPE,UUID
252  nano deploy.sh
253  ./deploy.sh
254  zpool status
255  export -a
256  zpool status
257  umount /target/boot
258  zpool status
259  export -a
260  zpool status
261  mount
262  zpool destroy rpool
263  mount
264  zpool status
265  ./deploy.sh
266  blkid
267  blkid /dev/sda3 | cut -d '"' -f 2
268  cat /target/fstab
269  cat /target/etc/fstab
270  umount /target/boot
271  zpool destroy rpool
272  nano deploy.sh
273  ./deploy.sh
274  reboot
275  nano deploy.sh
276  zpool status
277  ./deploy.sh
278  zpool status
```

```
279 zpool status
280 mount
281 zpool destroy rpool
282 df -h
283 nano deploy.sh
284 ./deploy.sh
285 zpool status
286 lsblk
287 sgdisk -p /dev/sda
288 reboot
289 zpool status
290 df -h
291 ls -al /tmp
292 nano deploy.sh
293 ls /tmp
294 nano deploy.sh
295 ./deploy.sh
296 zpoll status
297 zpool status
298 nano deploy.sh
299 ./deploy.sh
300 nano deploy.sh
301 nano deploy.sh
302 zpool status
303 nano deploy.sh
304 ./deploy.sh
305 zpool status
306 reboot
307 ip a
308 ip a
309 uname -a
310 ls -al /etc/apt/sources.list.d/
311 sed -i 's/stretch/buster/g' /etc/apt/sources.list
312 sed -i 's/stretch/buster/g' /etc/apt/sources.list.d/*
313 apt updatew
314 apt update
315 nano /etc/apt/sources.list.d/pve-install-repo.list
316 wget http://download.proxmox.com/debian/proxmox-ve-release-6.x.gpg -O
/etc/apt/trusted.gpg.d/proxmox-ve-release-6.x.gpg
317 apt update
318 apt full-upgrade
319 apt upgrade
320 cat /etc/os-release
321 ls -al /boot
322 ls -al /boot/grub
323 dpkg -l | grep kernel
324 apt dist-upgrade
325 apt install pve-kernel
326 apt search pve-kernel
327 apt install pve-kernel-5.4
328 dpkg -l | grep zfs
```

```
329 apt install pve-headers-5.4
330 dpkg -l | grep firmware
331 apt search pve-firmware
332 systemctl reboot
333 uname -a
334 ls -al /boot
335 ls -al /
336 rm /vmlinuz.old
337 rm /initrd.img.old
338 mv /vmlinuz /vmlinuz.o
339 mv /initrd.img /initrd.img.old
340 ls -al /
341 cd ..
342 ln -s boot/vmlinuz-5.4.65-1-pve vmlinuz
343 ln -s boot/initrd.img-5.4.65-1-pve initrd.img
344 ls -al
345 systemctl reboot
346 uname -a
347 apt install dosfstools
348 apt install screen
349 apt autoremove
350 apt autoclean
351 ls -al /
352 ls /boot
353 ls /boot/
354 ls -al /boot/
355 ls -al /boot/pve
356 systemctl reboot
357 uname -a
358 history | grep "tar "
359 history
360 systemctl poweroff
361 cd
362 ls
363 cat chroot.sh
364 ./chroot.sh
365 zpool status
366 zfs list
367 zfs list -t snapshot
368 cd /tmp/
369 ls
370 cd /root/
371 l
372 ld
373 ls
374 ls -l
375 cat deploy.sh
376 vi deploy.sh
377 ls -al /dev/disk/by-id/
378 wipefs -a -f /dev/sda
379 wipefs -a -f /dev/sdb
```

```
380 wipefs -a -f /dev/sdc
381 wipefs -a -f /dev/sdd
382 ls -al /dev/disk/by-id/
383 partprobe
384 mdadm --detail
385 cat /proc/mdstat
386 history
387 mdadm --help
388 mdadm --stop /dev/md0
389 cat /proc/mdstat
390 mdadm --zero-superblock /dev/sda
391 cat /proc/mdstat
392 partprobe
393 mdadm --zero-superblock /dev/sdb
394 vim deploy.sh
395 vi deploy.sh
396 sgdisk --zap-all /dev/sdb1
397 sgdisk --zap-all /dev/sdb
398 partprobe
399 mount| grep sdb
400 mdadm --zero-superblock /dev/sdb
401 mdadm --zero-superblock /dev/sdb1
402 wipefs -a -f /dev/sdb
403 wipefs -a -f /dev/sdb1
404 partprobe
405 systemctl reboot
406 ip a s
407 lsblk
408 lsblk -d -o name,wwn
409 ls -la /dev/disk/by-id/
410 partprobe
411 cat /proc/mdstat
412 swapon -s
413 lsblk
414 lsblk -d
415 lsblk -d -o name,rota
416 vi deploy.sh
417 #zpool create -o ashift=12 \
418     -0 acltype=posixacl -0 canmount=off -0 compression=lz4
419 #zpool create -o ashift=12 -0 acltype=posixacl -0 canmount=off -0
compression=lz4 -0 dnodesize=auto -0 normalization=formD -0 reltime=on -0 xattr=sa
420 zpool create -o ashift=12 -0 acltype=posixacl -0 compression=lz4 -0
reltime=on -0 xattr=sa dades raidz1 wwn-0x5000c500c520c2f7
wwn-0x5000c500cf855885 wwn-0x5000c500cf875876 wwn-0x5000c500cf8fda13
421 zpool status
422 df -h
423 cd /dades/
424 ls -la
```

```
425 df -h .
426 cd
427 zfs create dades/prova
428 zfs list
429 apt-get update
430 apt dist-upgrade
431 apt-get update
432 apt-get autoremove
433 apt-get install nfs-kernel-server
434 vim /etc/exports
435 apt-get install vim
436 vim /etc/exports
437 zfs get all
438 zfs get all | grep compress
439 zfs set
sharenfs=no_subtree_check,no_root_squash,async,rw=@192.168.1.0/24 dades
440 vim /etc/modprobe.d/zfs.conf
441 sysctl vm.swappiness
442 vim /etc/sysctl.conf
443 arc_summary | more
444 reboot
445 arc_summary | more
446 history
447 vim /etc/modprobe.d/zfs.conf
448 sysctl vm.swappiness
449 ((12*1024*1024*1024))
450 echo ((12*1024*1024*1024))
451 vim /etc/modprobe.d/zfs.conf
452 lsmod
453 lsmod | grep nfs
454 lsmod | grep zfs
455 lsmod | grep arc
456 arc_summary
457 arc_summary | more
458 free -m
459 vim /etc/modprobe.d/mdadm.conf
460 rm /etc/modprobe.d/mdadm.conf
461 apt-get remove --purge mdadm
462 touch /dades/prova/jur
463 vim /etc/initramfs-tools/conf.d/resume
464 update-initramfs -k all -u
465 ls -l /etc/network/interfaces
466 cat /etc/network/interfaces
467 apt-get install fio
468 cd /dades/prova/
469 fio --size=16G --name=create --filename=fio_file --bs=1M --nrfiles=1
--direct=1 --sync=0 --randrepeat=0 --rw=write --refill_buffers --end_fsync=1
--iodepth=200 --ioengine=libaio --fallocate=none
470 df -h
471 ls -l /dev/disk/by-id/
472 blkid /dev/disk/by-id/wwn-0x502b2a201d1c1b1a-part3
```



```
473 vi test.sh
474 cat test.sh
475 cd /dades/
476 fio --size=16G --name=create --filename=fio_file --bs=1M --nrfiles=1
--direct=0 --sync=0 --randrepeat=0 --rw=write --refill_buffers --end_fsync=1
--iodepth=200 --ioengine=libaio --fallocate=none
477 cd /
478 ls
479 df -h .
480 fio --size=16G --name=create --filename=fio_file --bs=1M --nrfiles=1
--direct=0 --sync=0 --randrepeat=0 --rw=write --refill_buffers --end_fsync=1
--iodepth=200 --ioengine=libaio --fallocate=none
481 apt-get install sdparm
482 sdparm /dev/sda
483 apt-get install smartctl
484 apt-get install smartmontools
485 smartctltop
486 top
487 ps -ef | grep fio
488 ps -ef | grep fio
489 kill -9 33540
490 ps -ef | grep fio
491 sdparm --set=WCE --save /dev/sda
492 sdparm /dev/sda
493 sdparm --set=WCE /dev/sda
494 sdparm /dev/sda
495 sdparm --set=WCE /dev/sdb
496 sdparm --set=WCE /dev/sdc
497 sdparm --set=WCE /dev/sdd
498 sdparm /dev/sde
499 sdparm --set=WCE /dev/sde
500 reboot
501 sdparm
502 sdparm /dev/sda
503 ls
504 cd
505 history
root@nas:~# fio --size=16G --name=create --filename=fio_file --bs=1M --
nrfiles=1 --direct=0 --sync=0 --randrepeat=0 --rw=write --refill_buffers --
end_fsync=1 --iodepth=200 --ioengine=libaio --fallocate=none
create: (g=0): rw=write, bs=(R) 1024KiB-1024KiB, (W) 1024KiB-1024KiB, (T)
1024KiB-1024KiB, ioengine=libaio, iodepth=200
fio-3.12
Starting 1 process
create: Laying out IO file (1 file / 16384MiB)
Connection to 192.168.1.76 closed by remote host.PS][eta 00m:47s]
Connection to 192.168.1.76 closed.
```

```
root@nas:~# sdparm /dev/sda
/dev/sda: ATA ST8000DM004-2CX1 0001
```

Read write error recovery mode page:

```
AWRE          1 [cha: n, def: 1]
ARRE          0 [cha: n, def: 0]
PER           0 [cha: n, def: 0]
```

Caching (SBC) mode page:

```
IC            0 [cha: n, def: 0]
WCE           0 [cha: y, def: 0]
RCD           0 [cha: n, def: 0]
```

Control mode page:

```
TST           0 [cha: n, def: 0]
SWP           0 [cha: n, def: 0]
```

En el conf para poner la memoria a 12gbs sense cache12*1024*1024*1024:
options zfs zfs_arc_max=12884901888 zfs_prefetch_disable=1

Lo cambia al reiniciar

```
root@nas:~# arc_summary | grep "Target size "
```

Target size (adaptive):	100.0 %	12.0 GiB
-------------------------	---------	----------

No faci swap si està apurat:

```
sysctl vm.swappiness
vm.swappiness = 80
```

Añadir al final /etc/sysctl.conf:

```
vm.swappiness=10
```

Lo cambia al reiniciar

```
root@nas:~# sysctl vm.swappiness
vm.swappiness = 10
```

No se que es

```
# cat /etc/initramfs-tools/conf.d/resume
RESUME=UUID=d896c0fc-3a4f-48c6-8a52-0b93acdb4677
```

```
root@nas:~# cat /etc/modprobe.d/zfs.conf
options zfs zfs_arc_max=12884901888 zfs_prefetch_disable=1
```

fio haria d'anar a 200MiB/s IOPS. Habilitar HCI

```
fio --size=16G --name=create --filename=fio_file --bs=1M --nrfiles=1 --
direct=1 --sync=0 --randrepeat=0 --rw=write --refill_buffers --end_fsync=1 -
-iodepth=200 --ioengine=libaio --fallocate=none
```

```
#!/bin/bash
```

```

NAME="$1"; shift
IFACE="$1"; shift
NAME="${NAME:-pve}"
IFACE="${IFACE:-enol}"
declare -a BOOTSSD
declare -a FSTAB
echo "Setting ZFS root on $NAME"
#get WWN for a /dev/sdX disk device
get_wnn () {
    ls -l /dev/disk/by-id/wnn-* | grep -v "part" | grep $1 | cut -d "-"
    -f 3|tr -d ' '
}
#find up to 2 smallest non-rotational disks of same size
find_boot_drives () {
    local BOOTSSD1=''
    local BOOTSSD2=''
    eval "$(lsblk -dn -o NAME,SIZE,TYPE,ROTA -x SIZE | sort| awk 'BEGIN
{ i = 0 } /disk/ { if ($4 < 1) {i=i+1; if (i == 1) {size=$2;print
"BOOTSSD1="$1} ; if (i == 2) {if (size == $2) print "BOOTSSD2="$1}}}')")
    if [ -n "$BOOTSSD1" ]; then
        BOOTSSD[0]="wnn-$(get_wnn $BOOTSSD1)"
    fi
    if [ -n "$BOOTSSD2" ]; then
        BOOTSSD[1]="wnn-$(get_wnn $BOOTSSD2)"
    fi
}
#wipe existing filesystems in each one partition
wipe_disk () {
    for p in $(ls -l /dev/disk/by-id/$1* | grep "part" |sort -r); do
        echo "...wiping partition $p"
        wipefs -a -f $p
    done
    echo "...wiping whole disk $1"
    wipefs -a -f /dev/disk/by-id/$1
}
#create partitions
create_partitions () {
    echo
    echo "Creating GPT partition table..."
    sgdisk --zap-all /dev/disk/by-id/$1
    echo "Creating BIOS boot partition..."
    sgdisk -al -n1:24K:+1000K -t1:EF02 -c 1:"BIOS boot" /dev/disk/by-id/$1
    #echo "Creating UEFI partition (unused, just in case).."
    #sgdisk -n2:1M:+512M -t2:EF00 -c 2:"EFI" /dev/disk/by-id/$1
    if [ ${#BOOTSSD[@]} -gt 1 ]; then
        echo "Creating Linux RAID boot partition..."
        sgdisk -n2:0:+2048M -t2:fd00 -c 2:"boot" /dev/disk/by-id/$1
    else
        echo "Creating Linux boot partition..."
        sgdisk -n2:0:+2048M -t2:8300 -c 2:"boot" /dev/disk/by-id/$1
    fi
}

```

<http://wiki.legido.com/>

```
wipe_disk $disk
create_partitions $disk
format_swap $disk
done
partprobe
sleep 2
sync

#Create boot filesystem

if [ ${#BOOTSSD[@]} -gt 1 ]; then
    echo "Creating mirror for /boot filesystem"
    BOOT_PART="/dev/md0"
    mdadm --create $BOOT_PART --level=1 --raid-devices=2 /dev/disk/by-
id/${BOOTSSD[0]}-part2 /dev/disk/by-id/${BOOTSSD[1]}-part2
else
    BOOT_PART="/dev/disk/by-id/${BOOTSSD[0]}-part2"
fi

echo "Formatting boot partition with ext4 filesystem ..."
mkfs.ext4 -F $BOOT_PART

# Create zfs root pool rpool

zpool create -o ashift=12 \
    -o acltype=posixacl -o canmount=off -o compression=lz4 \
    -o dnodesize=auto -o normalization=formD -o relatime=on -o xattr=sa
\
    -o mountpoint=/ -R /target \
    rpool ${BOOTSSD[0]}-part4

# Add mirror partitions if there are two boot disks

if [ ${#BOOTSSD[@]} -gt 1 ]; then
    zpool attach rpool ${BOOTSSD[0]}-part4 ${BOOTSSD[1]}-part4
fi

zpool status

#Create filesystem datasets for root
zfs create -o canmount=off -o mountpoint=none rpool/ROOT
zfs create -o canmount=noauto -o mountpoint=/ rpool/ROOT/pve
zfs mount rpool/ROOT/pve

#Create datasets
zfs create                                rpool/home
zfs create -o mountpoint=/root            rpool/home/root

#Mount /boot on /target/boot
mkdir /target/boot
```



```
#Prepare script for chroot
echo "Preparing chroot"
echo '#!/bin/bash' > /target/root/chroot.sh
chmod +x /target/root/chroot.sh
echo "declare -a BOOTSSD" >>/target/root/chroot.sh

for disk in ${BOOTSSD[@]}; do
    echo "BOOTSSD+=${disk}" >>/target/root/chroot.sh
done

if [ -e "/dev/md0" ]; then
    echo "mdadm --detail --scan >> /etc/mdadm/mdadm.conf"
>>/target/root/chroot.sh
fi

#echo "apt install --yes grub-pc" >>/target/root/chroot.sh

#echo "systemctl enable zfs-import-bpool.service" >>/target/root/chroot.sh
#echo "sleep 2" >>/target/root/chroot.sh
#echo "zfs set mountpoint=legacy bpool/B00T/pve" >>/target/root/chroot.sh
#echo "zpool set bootfs=bpool/B00T/pve bpool" >>/target/root/chroot.sh
#echo "sleep 2" >>/target/root/chroot.sh
#echo "mount -a" >>/target/root/chroot.sh

echo "grub-probe /boot" >>/target/root/chroot.sh
echo "update-initramfs -u -k all" >>/target/root/chroot.sh
echo "update-grub" >>/target/root/chroot.sh
for disk in ${BOOTSSD[@]}; do
    echo "grub-install /dev/disk/by-id/${disk}" >>/target/root/chroot.sh
done
echo "ls /boot/grub/*/zfs.mod" >>/target/root/chroot.sh

#echo "zfs snapshot bpool/B00T/pve@install" >>/target/root/chroot.sh
echo "zfs snapshot rpool/R00T/pve@install" >>/target/root/chroot.sh
echo 'echo "Exiting from chroot"' >>/target/root/chroot.sh
#exit
#Prepare mounts and jump to chroot
echo "Jump to chroot"
mount -o bind /dev /target/dev
mount -o bind /sys /target/sys
mount -o bind /proc /target/proc
#chroot /target /bin/bash /root/chroot.sh --login
chroot /target /bin/bash --login
exit
#Unmount all filesystems and export pools
umount -l /target/dev
umount -l /target/sys
umount -l /target/proc
umount -l /target/boot
sleep 2
```

```
zpool export -a  
sleep 2  
echo "Setup done!"
```

Create RAID:

```
zpool create -o ashift=12 -O acltype=posixacl -O compression=lz4 -O  
relatime=on -O xattr=sa dades raidz1 wwn-0x5000c500c520c2f7  
wwn-0x5000c500cf855885 wwn-0x5000c500cf875876 wwn-0x5000c500cf8fda13
```

Create shared:

```
zfs create dades/prova
```

Canviar disk

```
zfs replace  
zpool status
```

Crear nfs:

```
zfs set sharenfs=no_subtree_check,no_root_squash,async,rw=@192.168.1.0/24  
dades/prova
```

backup de zfs

send receive nfs

From:

<http://wiki.legido.com/> - **Legido Wiki**

Permanent link:

<http://wiki.legido.com/doku.php?id=informatica:microservers:history>



Last update: **2020/12/10 10:00**