

Software Deployment Server

(c) Günther Alka, 25.6.2013, CC-BY-SA

We want to:

- create backups and multiple master images from Macs and PCs (physical or virtual via ESXi) on our SAN
- deploy master images from our SAN to Macs and PCs (pool and personal usage)

- create updated or different master images and deploy them to our pools on demand
- keep it as simple as possible (even if you need to do something manually)

Our hardware:

- ZFS storage server appliance (AFP/NFS or SMB shares)
with all the ZFS goodies like snapshots

- Macs (boot via netboot or via usb stick), OSX 10.8
- PCs with PXE boot, Windows 8 pro-64, net-clone via clonezilla

Our software:

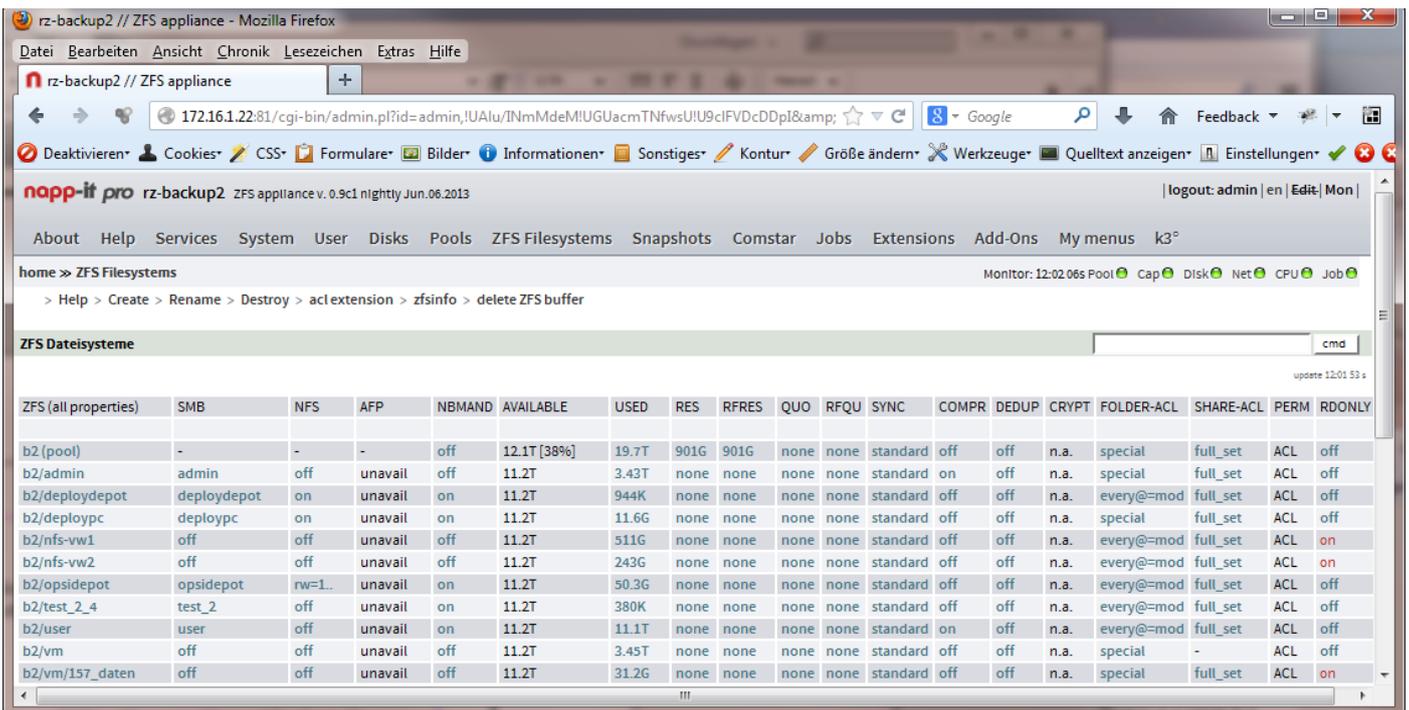
- NAS/SAN: OmniOS + napp-it webbased storage appliance (storage, free)
- OSX 10.8 + server-addon + deploy studio (free)

Our users/environment:

- university use/ pools and staff

Setup: NFS NAS/SAN

You need standard server hardware with some disks, download OmniOS, napp-it and and setup a NFS share <http://www.napp-it.org/doc/downloads/napp-it.pdf>



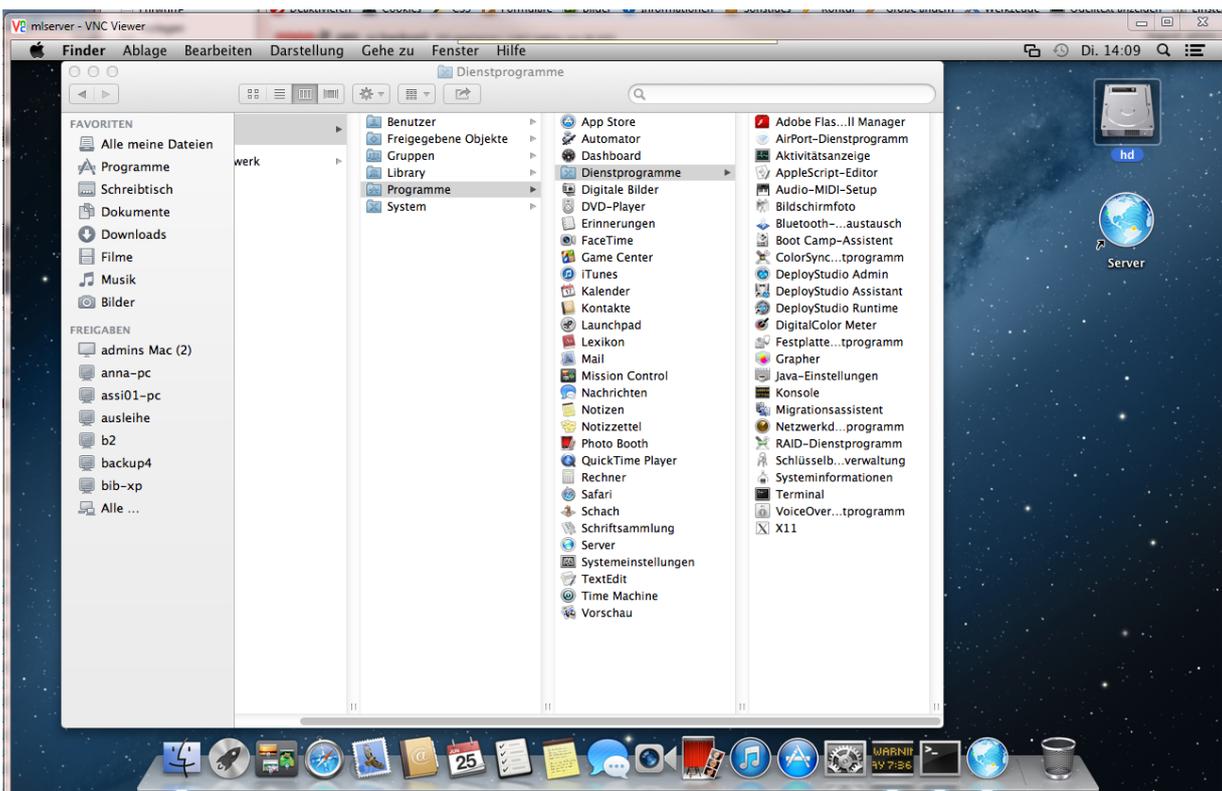
The screenshot shows the napp-it web interface for a ZFS appliance. The main content area displays a table of ZFS filesystems with various properties and status indicators.

ZFS (all properties)	SMB	NFS	AFP	NBMAND	AVAILABLE	USED	RES	RFRES	QUO	RFQU	SYNC	COMPR	DEDUP	CRYPT	FOLDER-ACL	SHARE-ACL	PERM	RDONLY
b2 (pool)	-	-	-	off	12.1T [38%]	19.7T	901G	901G	none	none	standard	off	off	n.a.	special	full_set	ACL	off
b2/admin	admin	off	unavail	off	11.2T	3.43T	none	none	none	none	standard	on	off	n.a.	special	full_set	ACL	off
b2/deploydepot	deploydepot	on	unavail	on	11.2T	944K	none	none	none	none	standard	off	off	n.a.	every@=mod	full_set	ACL	off
b2/deploypc	deploypc	on	unavail	on	11.2T	11.6G	none	none	none	none	standard	off	off	n.a.	special	full_set	ACL	off
b2/nfs-vw1	off	off	unavail	off	11.2T	511G	none	none	none	none	standard	off	off	n.a.	every@=mod	full_set	ACL	on
b2/nfs-vw2	off	off	unavail	off	11.2T	243G	none	none	none	none	standard	off	off	n.a.	every@=mod	full_set	ACL	on
b2/opsidepot	opsidepot	rw=1..	unavail	on	11.2T	50.3G	none	none	none	none	standard	off	off	n.a.	every@=mod	full_set	ACL	off
b2/test_2_4	test_2	off	unavail	on	11.2T	380K	none	none	none	none	standard	off	off	n.a.	every@=mod	full_set	ACL	off
b2/user	user	off	unavail	on	11.2T	11.1T	none	none	none	none	standard	on	off	n.a.	every@=mod	full_set	ACL	off
b2/vm	off	off	unavail	off	11.2T	3.45T	none	none	none	none	standard	off	off	n.a.	special	-	ACL	off
b2/vm/157_daten	off	off	unavail	off	11.2T	31.2G	none	none	none	none	standard	off	off	n.a.	special	full_set	ACL	on

Install Deployment server

Install OSX 10.8 on a Mac (prefer virtual installations on ESXi), add server addon and Deploystudio <http://www.deploystudio.com/Home.html>

- Install deploy studio
- Goto folder programmes/utilities and start Deploystudio Assistant and run setup Deploystudio, Deploystudio PC, create netboot and create bootable external drive (USB stick for Macs)



Configure Deploystudio for Macs

- see docs from Deploystudio and Internet, (we are struggling with that and OSX 10.8 server)
We are looking for a good HowTo as well

Configure Deploystudio PC

there are nearly no documents around, so i will show it in detail:

- start Deploystudio PC assistant on Mac OSX server ex with ip 172.16.1.80
- adress `http://172.16.1.80:60080`
with user admin and pw

Add a repository (we use a NFS share on our SAN): ex
`nfs://172.16.1.22/pool/filesystem`

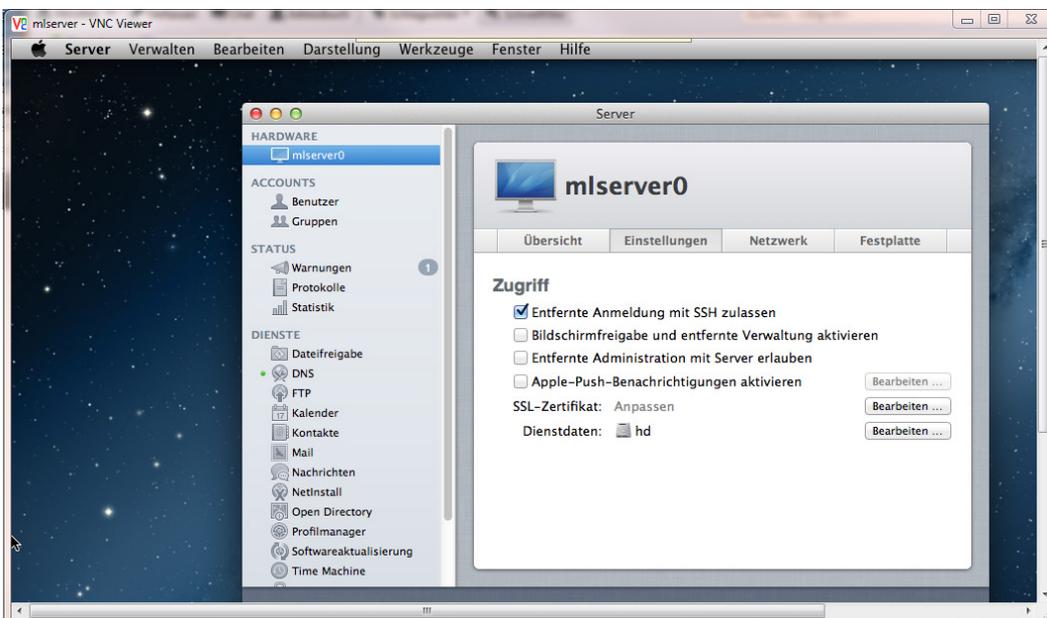
Activate root user on OSX

- goto system-preferences: user and groups, click on „connect“ network account server
on the new window click on „open directory services“

- Within the application „directory services,, click on the padlock to enable editing
- In menu edit, click on enable root and set a root pw

Activate SSH access

- Start OSX server manage and select/connect local server
- select hardware >> local servername and click on tab settings, allow remote ssh



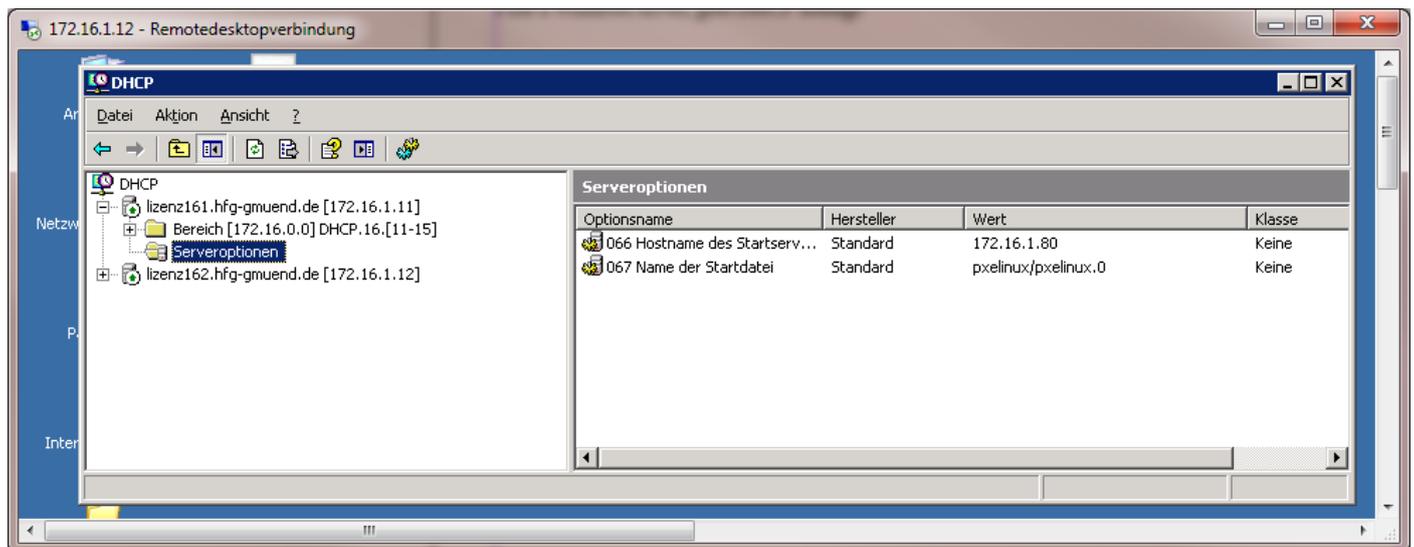
You are now able to connect your OSX server from a PC via Putty (remote console)
and WinSCP (remote file access as root)

Get clonezilla

download Clonezilla as ZIP file from <http://clonezilla.org/downloads.php>
(I needed newest testing release for my 1150 mainboards)

Setup your DHCP server for PXE booting (set bootserver and bootfile)

I use a Windows server, see DHCP settings

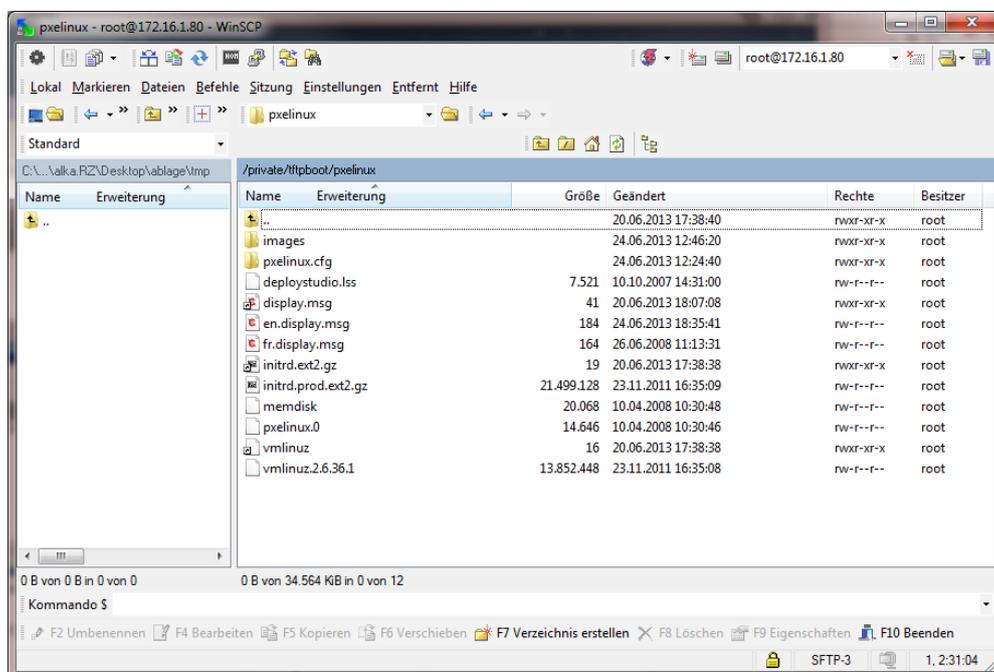


During PXE bootup your DHCP must deliver a bootserver (TFTP server) and a bootfilename

- 066 Hostname of startserver: 172.16.1.80 (=OSX deploy server)
- 067 Name of startfile: pxelinux/pxelinux.0

(Relative path below TFTP server root = /private/tftpboot on your Mac)

Start WinSCP and connect to OSX server as root



Important files/ folders

TFTP server root: /private/tftpboot/ with PXE bootfiles in /private/tftpboot/pxelinux/

Boot-Menu file: /private/tftpboot/pxelinux/display.msg (textfile)

Bootoptions: /private/tftpboot/pxelinux/pxelinux.cfg/default

Clonezillafiles (from zip-download) in (create folder images/clonezilla manually)

/private/tftpboot/pxelinux/images/clonezilla/

example of Menu file /private/tftpboot/pxelinux/display.msg (Startup message)

```
5deploystudio.lss
07Please press:
07- ,0' to boot on the local disk (default)
07- ,1' to boot memtest
07- ,2' to boot clonezilla
```

example of /private/tftpboot/pxelinux/pxelinux.cfg/default (PXE boot configuration)

```
default 0
timeout 100
prompt 1
display display.msg

label 0
    LOCALBOOT 0

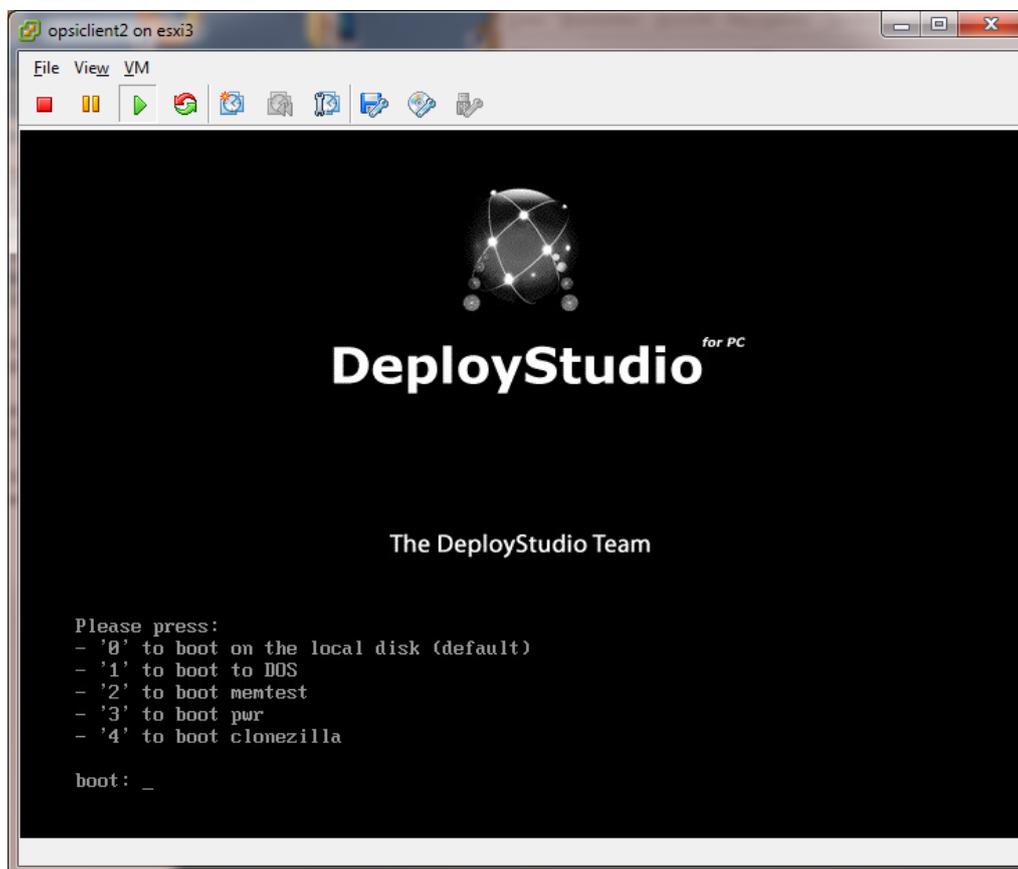
label 1
    KERNEL images/clonezilla/live/memtest

label 2
    KERNEL images/clonezilla/live/vmlinuz
    APPEND initrd=images/clonezilla/live/initrd.img boot=live config noswap fetch=tftp://172.16.1.80/private/tftpboot/pxelinux/images/clonezilla/live/filesystem.squashfs
```

Now you can enable PXE boot on your PC

- Netboot requests server and startupfile from DHCP and loads pxelinux.0 with a menu
- default (0 is boot from HD), other options are a memtest and start clonezilla

example (with some extra options): PXE menu after reboot



We use ESXi (with netboot activated in VM-Bios settings) for tests and to create master images

Create a basic master-image (Pool machines)

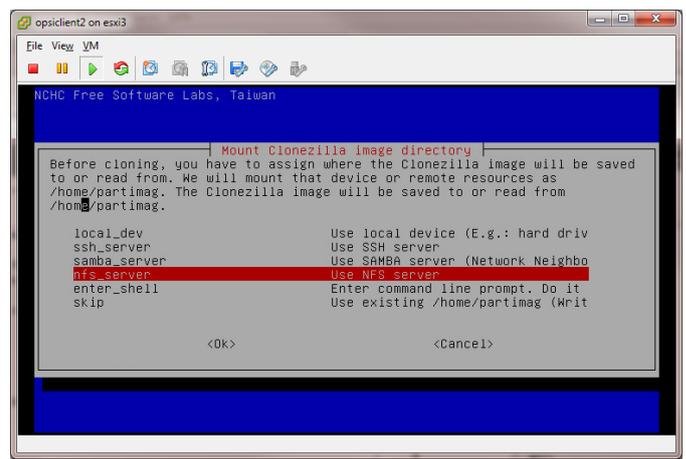
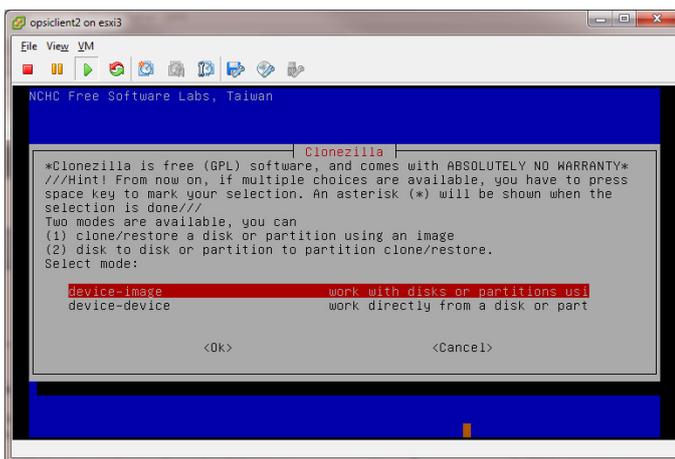
- Create a new VM in ESXi ex with 30 GB virtual disk, connect the CD drive with a Windows 8 ISO. Setup Windows 8 in workgroup mode, register Windows, update to newest patchlevel
- Create a snap (we use ZFS snaps on a NFS datastore, much more efficient than ESXi snaps)
- Clone the snap to a new VM2
- Add applications for pool 1
- Create a snap (can go back)
- Use the cloned VM2 of the base install, add applications for pool 2
- create a snap (can go back)

Save a master image on your NFS storage

Now you have two VMs with a different setup

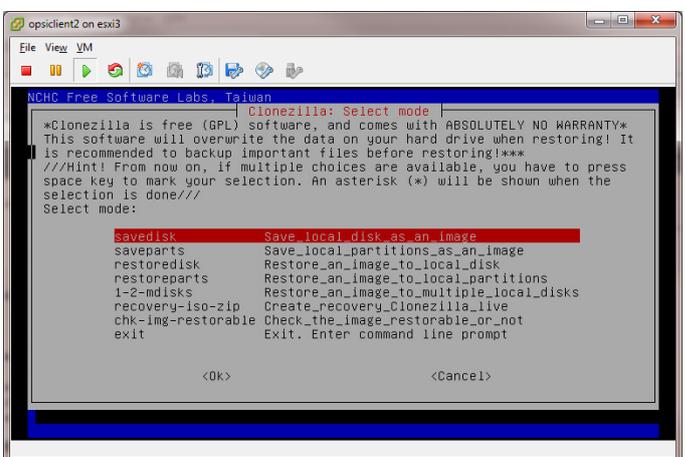
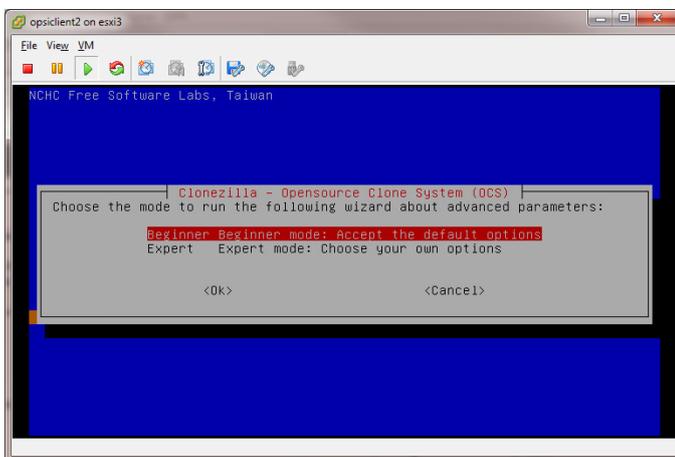
Edit Bios settings of a VM to boot via PXE, startup

On PXE boot, select keyboard and language and „start clonezilla,, with „device - image“ and NFS



Select IP of your NFS storage and your share with an absolute path (on ZFS /pool/filesystem)

We start with „Beginner,, and „Save disk to image,,



Attention: Windows 8 creates a small additional system partition.

With Windows 8, you must use „save disk,,

Next step is to confirm a filename and start imaging disk -> NFS server

Restore an image

Startup your PC with PXE boot enabled, select „start Clonezilla„ on bootmenu,

- select device-image,
- select NFS server
- select Beginner
- select restoredisks
- select a image from list

and press ok to restore the image (you need to confirm this twice with „y„)

Bootup Windows 8

On first startup, Windows 8 loads the correct driver, does a hd check and bootup.

After this you need to:

- size partition to use the whole disk (ex master image was 30 GB, current disk is 240 GB),
use computer - management >> disk - management: extend partition size

Last steps

Optionally you need to modify hostname, security ID and add the computer to a domain.