XEN

- virtuelle Maschine per dd über das Netzwerk clonen
- Debian XEN-Server via debootstrap auf Server4You Root-Server installieren
- Frag is bigger than frame. fatal error disabling device
- kleine XEN-Begriffserklärung
- LVM verwaltete Disk vergrößern
- mehrere Netzwerkinterfaces in den DomU
- Uhrzeit auf den domU geht falsch
- Zufällige MAC-Adressen für XEN VMs erzeugen

virtuelle Maschine per dd über das Netzwerk clonen

Diese Methode funktioniert wunderbar bei XEN-Servern, deren virtuelle Maschinen auf einem logischen Volume (z.B. LVM) liegen.

Auf dem Zielsystem wird nc gestartet und die Standardausgabe auf das Zielvolume gelegt:

```
nc –l –p 10000 | dd of=/dev/lvm-raid/server2-disk
```

Auf dem Quellsystem (die virtuelle Maschine muss vorher gestoppt werden!) wird die Standardausgabe von dd an nc gepiped:

```
dd if=/dev/lvm-raid/server1-disk | nc ziel.meinserver.de 10000 –q 10
```

Debian XEN-Server via debootstrap auf Server4You Root-Server installieren

Dies ist eine Anleitung um Debian Squeeze und XEN-Server auf einem dedizierten Host der Server4You zu installieren.

Der Server, für den diese Anleitung erstellt wurde, hat folgende Stats:

- Fujitsu PRIMERGY MX130 S2
- AMD Opteron 3280 Octacore
- 2x 2000 GB SATA 3,5„ 7.200 rpm (für Softraid)
- 16x Gigabyte RAM
- +3 zusätzliche IP Adressen

Der Server sollte Virtualisierung unterstützen. Ob das Virtualisierungs-Bit aktiviert ist, läßt sich vor der Installation über dieses Kommando herausfinden: „egrep '(vmx|svm)' /proc/cpuinfo“. Sobald der Server mit einem XEN-Kernel gebootet ist, verschwindet die Anzeige aus /proc/cpuinfo. Die Info taucht dafür aber im dmesg von XEN auf: „xm dmesg | egrep -i '(vmx|svm)'“. Bitte beachten: bei AMD-Prozessoren heißt das Flag SVM, bei Intel hingegen VMX.

Statt vif-bridge verwende ich für meine Server vif-nat. vif-bridge hat den Nachteil, dass man von den 4 IP-Adressen letztlich nur 2 für die VMs verwenden kann.

Ich habe alle öffentlichen IP-Adressen direkt auf dem Xen-Host konfiguriert und natte diese auf die privaten IPs der virtuellen Server. Nach außen wird jede VM auf die zugehörige externe IP genattet. Ich habe das ganze mit Shorewall umgesetzt. Wer sich für meine Regeln interessiert möge mich anschreiben.

1. Sofern noch nicht geschehen unbedingt Daten sichern. Es handelt sich hier um eine Neuinstallation, alle vorhandenen Daten gehen verloren.

2. Recovery System über https://my.server4you.de aktivieren und per SSH verbinden

3. Im Recovery System den alten Server löschen, d.h:

- mdadm raids auflösen und löschen
- per fdisk alle Partitionen löschen

4. neue Partitionen erstellen:

neue Partitionen auf beiden Disks exakt gleich konfigurieren (z.B.: 500MB /boot, 8GB SWAP, 100GB System, Rest für LVM + XEN)

5. md-raids konfigurieren:

- /dev/sda1 + /dev/sdb1 ? md0 (boot)
- /dev/sda3 + /dev/sdb3 ? md1 (XEN dom0)
- /dev/sda4 + /dev/sdb4 ? md2 (Daten)

```
# in neueren mdadm Versionen:
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda1 /dev/sdb1
```

```
# alt:
mdadm --create /dev/md0 --raid-level=1 /dev/sda1 /dev/sdb1
mdadm --create /dev/md1 --raid-level=1 /dev/sda3 /dev/sdb3
mdadm --create /dev/md2 --raid-level=1 /dev/sda4 /dev/sdb4
```

6. Partitionen formatieren

```
mkfs.ext2 -L boot /dev/md0
mkfs.ext4 -L system /dev/md1

mkswap -L swap-sda /dev/sda2
mkswap -L swap-sdb /dev/sdb2
```

7. Partitionen für debootstrap mounten

```
mkdir /mnt/debinst
mount /dev/md1 /mnt/debinst
mkdir /mnt/debinst/boot /mnt/debinst/work
mount /dev/md0 /mnt/debinst/boot
cd /mnt/debinst/work
```

8. debootstrap besorgen:

```
wget http://ftp.us.debian.org/debian/pool/main/d/debootstrap/debootstrap_1.0.26+squeeze1_all.deb
ar -xf debootstrap_1.0.26+squeeze1_all.deb
```

9. Installation starten

```
cd /
/usr/sbin/debootstrap --arch amd64 squeeze /mnt/debinst http://ftp.de.debian.org/debian
```

10. nötiges Zeugs mounten

```
mount sysfs /mnt/debinst/sys -t sysfs
mount proc /mnt/debinst/proc -t proc
mount -o bind /dev /mnt/debinst/dev
```

11. ins neue System chrooten

```
cd /
LANG= chroot /mnt/debinst /bin/bash
```

12. ggf. /etc/apt/sources.list anpassen:

```
deb http://ftp.de.debian.org/debian/ stable main contrib non-free
deb-src http://ftp.de.debian.org/debian/ stable main contrib non-free
deb http://security.debian.org/ stable/updates main contrib non-free
deb-src http://security.debian.org/ stable/updates main contrib non-free
```

13. eventuell vorhandene Updates installieren

```
aptitude update
aptitude full-upgrade
```

14. vim installieren

```
aptitude install vim
```

15. /etc/fstab anpassen

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0

/dev/md0 /boot  ext2 defaults,noatime,ro,nosuid,nodev 0 2
/dev/md1 /  ext4 defaults,noatime   0 1

LABEL=swap-sda none  swap sw,pri=1 0 0
LABEL=swap-sdb none  swap sw,pri=1 0 0
```

16. proc mounten

```
mount -t proc proc /proc
```

17. Konsolenoptionen

```
aptitude install console-setup
```

folgendes auswählen: Germany - Eliminate dead keys UTF-8 . Combined - Latin; Slavic Cyrillic; Greek

18. Netzwerkkonfiguration

in /etc/network/interfaces folgendes eintragen (entsprechend für deinen Server anpassen!):

```
# /etc/network/interfaces - network interfaces configuration

# loopback interface
auto lo
iface lo inet loopback

# ethernet interface
auto eth0
iface eth0 inet static
  address XXX.XXX.XXX.XXX
  network XXX.XXX.XXX.XXX
  netmask 255.255.255.XXX
  broadcast XXX.XXX.XXX.XXX
  gateway XXX.XXX.XXX.XXX
```

in /etc/resolv.conf folgendes eintragen (eigene Domains einsetzen):

```
nameserver 8.8.8.8
nameserver 8.8.4.4
search meinserver.de
domain meinserver.de
options rotate
options timeout:3
```

den Hostnamen in /etc/hostname eintragen

```
xen1.meinserver.de
```

die Hosts-Datei anpassen (/etc/hosts)

```
XXX.XXX.XXX.XXX    xen1.meinserver.de xen1

127.0.0.1 localhost
::1   localhost ip6-localhost ip6-loopback
fe00::0  ip6-localnet
ff00::0  ip6-mcastprefix
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
```

19. Sprachpaket installieren

```
aptitude install locales
dpkg-reconfigure locales
```

Auswählen: de_DE.UTF-8

```
          en_US.UTF-8
```

Danach je nach gewünschter Systemsprache englisch oder deutsch auswählen

20. Systemzeit konfigurieren

```
   dpkg-reconfigure tzdata
```

1. > Europe / Berlin auswählen

21. SSH für den späteren Remote-Zugriff installieren

```
aptitude -y install ssh
```

22. Kennwörter und SSH-Keys verteilen

Root-Kennwort setzen:

```
passwd root
```

und SSH-Key verteilen

```
   mkdir /root/.ssh
   vi /root/.ssh/authorized_keys
```

deinen SSH-Pubkey reinkopieren

23. mdadm + mta installieren

```
   aptitude install mdadm postfix
```

Postfix Einstellungen: Internet site, System mail name: <FQDN>

24. mdadm-Settings anpassen und überprüfen, ob das Raid korrekt erkannt wurde

```
   mdadm --detail --scan # und mit /etc/mdadm/mdadm.conf vergleichen
```

sollte etwa so aussehen:

```
# mdadm.conf
#
# Please refer to mdadm.conf(5) for information about this file.
#

# by default, scan all partitions (/proc/partitions) for MD superblocks.
# alternatively, specify devices to scan, using wildcards if desired.
DEVICE partitions

# auto-create devices with Debian standard permissions
CREATE owner=root group=disk mode=0660 auto=yes

# automatically tag new arrays as belonging to the local system
HOMEHOST <system>

# instruct the monitoring daemon where to send mail alerts
MAILADDR info@meinserver.de

# definitions of existing MD arrays
ARRAY /dev/md/0 metadata=1.2 UUID=0ae98489:3d0011dd:422ef339:7c089e1f name=disk1176:0
ARRAY /dev/md/1 metadata=1.2 UUID=d89a438e:7f0f2925:dfc1dd45:59788ac8 name=disk1176:1
ARRAY /dev/md/2 metadata=1.2 UUID=bbab94a1:003343e6:6b2629ab:2f8193f7 name=disk1176:2

# This file was auto-generated on Sun, 23 Sep 2012 14:39:40 +0200
# by mkconf 3.1.4-1+8efb9d1+squeeze1
```

25. Kernel installieren

```
aptitude install linux-image-2.6-amd64 firmware-linux-nonfree
```

26. Boot-Loader installieren

```
aptitude install grub2
```

1. > Command line: „quiet“

```
grub-install /dev/md0
update-grub
```

27. alles sauber unmounten und Recovery-Modus beenden:

```
/etc/init.d/postfix stop
/etc/init.d/mdadm stop
exit # (jetzt wieder in der Recovery shell
umount /mnt/debinst/dev /mnt/debinst/sys /mnt/debinst/proc /mnt/debinst/boot
umount /mnt/debinst
<code>

Recovery-Modus über https://my.server4you.de beenden und das System rebooten.

28. Aufräumen

jetzt sollte der Server hoffentlich ordentlich gebootet wieder hochkommen

einloggen und das /work-Verzeichnis löschen
<code>
ssh xen.meinserver.de
rm -rf /work
```

29. mdadm Resync beenden lassen

```
nach dem Reboot werden unmounted md-Raids üblicherweise als "active (auto-read-only)" angezeigt
<code>
```

```
mdadm --readwrite /dev/md0
mdadm --readwrite /dev/md2
```

md1 ist bereits rw, da das unser root-Filesystem ist. md0 ist /boot (per fstab read-only gemounted) und auf md2 kommt gleich unser LVM

30. nützliche Tools installieren

</code> aptitude install htop tcpdump mtr-tiny strace sysstat lshw </code>

31. LVM konfigurieren (kommt auf das freie md-Raid md2)

```
aptitude install lvm2
pvcreate /dev/md2
vgcreate lvm-raid /dev/md2
```

32. XEN installieren

```
mount -o remount,rw /boot
aptitude install xen-hypervisor-amd64 xen-linux-system-2.6-xen-amd64 xen-tools
```

33. XEN-Kernel höher priorisieren und ein paar Boot-Optionen setzen

```
dpkg-divert --divert /etc/grub.d/08_linux_xen --rename /etc/grub.d/20_linux_xen
update-grub
```

folgendes ab Zeile 10 in /etc/default/grub einfügen (nicht mehr als 1GB RAM für dom0):

```
# Xen boot parameters for all Xen boots
GRUB_CMDLINE_XEN="dom0_mem=1024M"
```

und in /etc/xen/xend-config.sxp die Einträge wie folgt aktivieren/ändern:

```
(network-script network-nat)
(vif-script vif-nat)
(dom0-min-mem 512)
(enable-dom0-ballooning no)
(total_available_memory 0)
(dom0-cpus 0)
(vncpasswd '')
```

34. in den XEN-Kernel rebooten, wieder einloggen und XEN prüfen

```
uname -a
```

sollte so aussehen:

```
   Linux xen1.meinserver.de 2.6.32-5-xen-amd64 #1 SMP Sun May 6 08:57:29 UTC 2012 x86_64 GNU/Linu
```

„xm info“ sollte auch jede Menge Infos liefern

35. /etc/xen-tools/xen-tools.conf anpassen bzw. nach eigenen Vorlieben konfigurieren:

```
lvm = lvm-raid
arch = amd64
```

```
install-method = debootstrap
vcpus  = 4        # number of cores
size   = 50Gb     # Disk image size.
memory = 1024Mb   # Memory size
swap   = 128Mb    # Swap size
noswap = 1        # Don't use swap at all for the new system.
fs     = ext4     # use the EXT3 filesystem for the disk image.
dist   = squeeze  # Default distribution to install.
image  = sparse   # Specify sparse vs. full disk images.
gateway    = XXX.XXX.XXX.1
netmask    = 255.255.255.0
broadcast  = XXX.XXX.XXX.255
genpass = 0
hash_method = sha512
passwd = 1
kernel = /boot/vmlinuz-`uname -r`
initrd = /boot/initrd.img-`uname -r`
arch = amd64
mirror = http://ftp2.de.debian.org/debian/
```

33. die erste virtuelle Maschine erzeugen:

```
xen-create-image --hostname=vm1.myserver.de --ip=XXX.XXX.XXX.123
```

sollte dann so aussehen:

```
  General Information
--------------------
Hostname       :  vm1.myserver.de
Distribution   :  squeeze
Mirror         :  http://ftp2.de.debian.org/debian/
Partitions     :  /               50Gb  (ext4)
Image type     :  full
Memory size    :  1024Mb
Kernel path    :  /boot/vmlinuz-2.6.32-5-xen-amd64
Initrd path    :  /boot/initrd.img-2.6.32-5-xen-amd64

Networking Information
----------------------
IP Address 1   : XXX.XXX.XXX.123 [MAC: 00:16:3E:B4:09:6A]
Netmask        : 255.255.255.0
Broadcast      : XXX.XXX.XXX.255
Gateway        : XXX.XXX.XXX.1


Creating ext4 filesystem on /dev/lvm-raid/vm1.myserver.de-disk
Done
Installation method: debootstrap
Done

Running hooks
Done

No role scripts were specified.  Skipping

Creating Xen configuration file
Done
Setting up root password
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
All done


Logfile produced at:
  /var/log/xen-tools/vm1.myserver.de.log

Installation Summary
---------------------
Hostname        :  vm1.myserver.de
Distribution    :  squeeze
IP-Address(es)  :  XXX.XXX.XXX.123
RSA Fingerprint :  24:29:f0:67:60:f1:4e:f6:41:7b:ad:0a:8e:88:87:8f
Root Password   :  N/A
```

36. die VM starten und beim Booten mit Hochfahren

```
mkdir /etc/xen/auto
mv /etc/xen/vm1.myserver.de.cfg /etc/xen/auto/
xm create /etc/xen/auto/vm1.myserver.de.cfg
```

Frag is bigger than frame. fatal error disabling device

Es gibt aktuell einen Bug, vermutlich im XEN-Kernel, der DomU sporadisch mal das Netzdevice herunterfährt (http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=701744)

Im Syslog der Dom0 steht dann folgendes:

```
Jul 24 11:29:00 dom0 kernel: [4358248.003726] vif vif-8-0: vif8.0: Frag is bigger than frame.
Jul 24 11:29:00 dom0 kernel: [4358248.003747] vif vif-8-0: vif8.0: fatal error; disabling device
Jul 24 11:29:00 dom0 kernel: [4358248.005158] public: port 9(vif8.0) entering forwarding state
```

Mit folgendem Workaround läßt sich das Problem abstellen (für jedes Interface ausführen):

```
ethtool -K eth0 gso off
```

Hier ein Script um alle Ethernet-NICs entsprechend zu konfigurieren:

```
#!/bin/bash
if_modes="gso"
for iface in $(ifconfig | awk '$0 ~ /Ethernet/ { print $1 }'); do
 for if_mode in ${if_modes}; do
   ethtool -K $iface $if_mode off 2>/dev/null
 done
done
```

kleine XEN-Begriffserklärung

dom0

Unter XEN wird die physikalische Maschine als dom0 bezeichnet.

domU

Die Gastmaschinen unter XEN werden als domU bezeichnet, wobei das U wird für einen konkreten Gast durch einen Index ersetzt wird.

LVM verwaltete Disk vergrößern

LVM basierte Disks für die VMs sind der übliche Standard, da sie deutlich performanter als Disk-Images sind. Sollte der Platz in einer DomU mal knapp werden, läßt sich das LogicalVolume problemlos vergrößern.

Die VM muss vorher runtergefahren werden. Auf dem Xen-Host läßt sich das Volume (in diesem Fall wurde es mit ext3 formatiert) dann folgendermaßen vergrößern:

```
lvresize -L +20G /dev/lvname/server1-disk
e2fsck -f /dev/lvname/server1-disk
resize2fs /dev/lvname/server1-disk
e2fsck -f /dev/lvname/server1-disk
xm create /etc/xen/auto/server1.cfg
```

bei XFS schaut es so aus:

```
lvresize -L +20G /dev/lvname/server1-disk
xfs_growfs -d /dev/lvname/server1-disk
xm create /etc/xen/auto/server1.cfg
```

mehrere Netzwerkinterfaces in den DomU

Diese Anleitung zeigt, wie auf einem Xen-Server zwei Bridges eingerichtet werden, um den Servern z.B. ein internes und ein externes Netz zur Verfügung zu stellen.

Es wird ein neues Script zur Verwaltung der Bridges angelegt (und mit „chmod a+x ausführbar gemacht):

```
#!/bin/sh
dir=$(dirname "$0")

"$dir/network-bridge.orig" "$@" "netdev=eth0"
"$dir/network-bridge.orig" "$@" "netdev=eth1"
```

In /etc/xen/xend-config.sxp wird eine Option wie folgt geändert:

```
- (network-script network-bridge)
+ (network-script my-network-bridge)
```

Danach den xend neu starten (service xend restart).

In den Configs der VM die Zeile für das vif wie folgt ändern:

```
- vif         = [ 'ip=192.168.125.163,mac=00:16:3E:D4:4D:FB' ]
+ vif         = [ 'ip=192.168.125.163,mac=00:16:3E:D4:4D:FB,bridge=eth0','ip=192.168.249.179,mac
```

Es sind 2 Blöcke durch das ' zusammengefasst.

Jetzt die betroffene VM neu starten (besser ganz ausschalten und neu anfahren (xm shutdown und xm create)). Nun sollten 2 Interfaces sichtbar sein, die sich aus der VM heraus ganz normal konfigurieren lassen.

Uhrzeit auf den domU geht falsch

Diese Anleitung wurde mit einer CentOS 5.3 XEN-Installation und CentOS 5.3 Gästen getestet.

Dies ist ein bekannter Bug, der wohl noch nicht gefixt werden konnte. Im Logfile tauchen immer wieder folgende Meldungen auf:

```
May 26 07:59:11 files ntpd[1378]: time reset -97.058129 s
May 26 08:04:01 files ntpd[1378]: synchronized to 212.34.164.250, stratum 2
May 26 08:20:12 files ntpd[1378]: time reset -97.290284 s
May 26 08:25:22 files ntpd[1378]: synchronized to 212.34.164.250, stratum 2
May 26 08:42:34 files ntpd[1378]: synchronized to 212.34.160.80, stratum 2
May 26 08:42:34 files ntpd[1378]: time reset -97.499207 s
May 26 08:47:04 files ntpd[1378]: synchronized to 212.34.164.250, stratum 2
May 26 08:48:45 files ntpd[1378]: synchronized to 212.34.174.99, stratum 2
```

Umgehen kann man dies, indem man den Gästen erlaubt, ihre Uhrzeit selbst zu setzen und einen NTP-Client installiert.

Dazu muss folgendes in /etc/sysctl.conf eingetragen werden:

```
xen.independent_wallclock = 1
```

danach den NTP-Dämon installieren:

```
yum -y install ntp
```

Konfiguriert wird dieser in /etc/ntp.conf. Dort könnte man z.B. eigene lokale Zeitserver eintragen. Bevor man nun den NTP-Dämon startet, sollte man vorher nochmal die Zeit angleichen:

```
ntpdate 0.ubuntu.pool.ntp.org
service ntpd start
```

Zufällige MAC-Adressen für XEN VMs erzeugen

Beim Klonen von vorhandenen oder virtualisieren von physischen Servern ist dieser Zufallsgenerator für XEN Mac-Adressen immer wieder nützlich.