

libvirt / kvm

Jens Link

jl@jenslink.net

libvirt / kvm

- Freiberuflicher Consultant
- **IPv6**
- komplexe Netzwerke
- Netzwerksecurity
- Netzwerkmonitoring
- Troubleshooting
- Wenn alles klappt in den nächsten Monaten viel KVM / OpenVSwitch / OpenStack / ...

- 1 Virtualisierung
- 2 kvm / libvirt
- 3 Automatisierung
- 4 Infrastruktur
- 5 ToDo
- 6 ToDo

Übersicht

- 1 Virtualisierung
- 2 kvm / libvirt
- 3 Automatisierung
- 4 Infrastruktur
- 5 ToDo
- 6 ToDo

- Virtualisierung gab es schon in den 1960er auf Großrechnern
- Im “PC” Bereich ab ca. 1999

- “Teilen” des OS der darunter liegenden Hardware
- Emulieren von Hardware

Was gibt es heute auf dem Markt?

- VMWARE (ESX, Workstation, ...)
- Microsoft HyperV
- Virtual Box
- XEN
- KVM
- ...

Warum / Wozu nutze ich Virtualisierung / KVM?

- Neuer Server mit viel RAM und CPU
- Provider gibt (auf Anfrage) ein /28 IPv4 bzw /48 IPv6
- Trennung von Diensten und Personen
- “schnell” Testsysteme aufsetzen, neue Sachen für Kunden testen, lernen, . . .
- Demo Umgebung für Schulungen / Vorträge
- Bekannter fluchte ständig über XEN
- Windows (so 2-3x im Monat)
- Wer sich mit Cisco / Juniper beschäftigen will: (Testversionen) von Juniper Firefly und Cisco Virtual Service Router gibt es kostenlos zum Download, auch für KVM

Wo nutze **ich** Virtualisierung?

- Laptop (KVM + VMWARE Workstation)
- Test ESXi Server @home
- Root Server (kvm)

Nachteile Virtualisierung?

- Layer-8 Probleme
- mal “eben schnell” einen neuen Server installieren
- Wenn man es flasch macht: Fehlende Ausfallsicherheit (2 virtuelle DNS Server auf einer Hardware)
- Overhead (technisch und organisatorisch)
- ggf. mehr Komplexität
- Server Admins machen zum Teil auch Netzwerke

- Doku - ja auch für die eigenen Systeme
- Automatisierung - Kostet am Anfang Zeit, spart aber auf längere Sicht
- Eine passend Infrastruktur
- Für einen Server im Internet: Einen passenden Provider. Mit nur 1 o. 2 IP Adressen bringt das nur wenig

Übersicht

- 1 Virtualisierung
- 2 **kvm / libvirt**
- 3 Automatisierung
- 4 Infrastruktur
- 5 ToDo
- 6 ToDo

- **Nicht:** Keyboard - Video - Mouse Switch
- Virtualisierungs Lösung für Linux x86
- Erfordert Hardware Unterstützung (Intel VT / AMD V)
- `egrep -c '(vmx|svm)' /proc/cpuinfo`
- Kernel Modul (seit 2.6.20 im Kernel)
- Es geht wohl auch KVM in KVM

A toolkit to interact with the virtualization capabilities of recent versions of Linux (and other OSes), see our project goals for details.

- The KVM/QEMU Linux hypervisor
- The Xen hypervisor on Linux and Solaris hosts
- The LXC Linux container system
- The OpenVZ Linux container system
- The User Mode Linux paravirtualized kernel
- The VirtualBox hypervisor
- The VMware ESX and GSX hypervisors
- The VMware Workstation and Player hypervisors
- The Microsoft Hyper-V hypervisor
- The IBM PowerVM hypervisor
- The Parallels hypervisor

- NAT
- Bridging
- VEPA (Virtual Ethernet Port Aggregator, 802.1Qbg)
- OpenVSwitch

- Platte (IDE, SCSI, USB)
- FibreChannel
- LVM
- iSCSI
- NFS
- Filesystem

- “Shared Folders”
- USB Passthrough
- OVA

- **virsh** - Management via CLI
- **virt-install** - Installieren von VMs
- **virt-top** - Monitoring
- **virt-viewer** - Virtuelle Konsole (aka vncviewer)
- **virt-manager** - GUI
- **virt-clone** - clonen von VMs

Demo

Als Referenz: `virt-install -name test -ram 128
-file=/opt/vms/test.img,format=qcow2 -file-size 5
-location=http://ftp.de.debian.org/debian/dists/wheezy
-extra-args=netcfg/get_ipaddress=192.168.99.1
netcfg/get_netmask=255.255.255.0
netcfg/get_gateway=192.168.99.1
netcfg/get_nameservers=192.168.99.1
netcfg/disable_dhcp=true auto=true interface=eth0
hostname=test domain=example.com
url=http://192.168.98.100/preseed.cfg-vnc`

Übersicht

- 1 Virtualisierung
- 2 kvm / libvirt
- 3 Automatisierung**
- 4 Infrastruktur
- 5 ToDo
- 6 ToDo

- preseed + virt-install
- puppet
- Alternativen gibt es viele: FAI anstelle von pressed, chef / cfengine / ansible anstelle von puppet

*Preseeding provides a way to set answers to questions asked during the installation process, without having to manually enter the answers while the installation is running. This makes it possible to fully automate most types of installation and even offers some features not available during normal installations.*¹

¹<https://wiki.debian.org/DebianInstaller/Preseed>

Ausflug: preseed (II)

```
d-i debian-installer/locale string en_US

# The values can also be preseeded individually for greater
#d-i debian-installer/language string en
#d-i debian-installer/country string NL
#d-i debian-installer/locale string en_GB.UTF-8
# Optionally specify additional locales to be generated.
#d-i localechooser/supported-locales multiselect en_US.UTF-
```

Welche Variablen kann man überhaupt setzen?

```
$ debconf-get-selections --installer > file  
$ debconf-get-selections >> file
```

- Config Management System
- Grundsatz: “Code your Config”
- Bei mir (noch) nur die Basis Config: ssh, Monitoring, Pakete, . . .
- Kannte ich aus Kundenprojekten

- Bei mir: Master - Slave Config, IPv6 only (geplant)
- Ohne Master geht auch: Client zieht Config aus git, puppet wird lokal aufgerufen

Bei Master - Slave:

- Kommunikation über HTTPS
- Bis ca. 100 Clients reicht der integrierte HTTP Server, danach wird Apache mit `mod_passenger` empfohlen
- Bei der ersten Verbindung erstellt der Client ein Zertifikat welches vom Server signiert werden muss
- Wer lernen will: Es gibt bei puppetlabs eine VM + Tutorial

Ausflug: puppet (IV)

```
/etc/puppet/manifests/site.pp
```

```
node default {
    include apt_conf
    include packages
    include ssh
    include timezone
}

node 'web.quux.de' inherits 'default' {
    firewall { '80 allow http(s)':
        port    => ['80', '443'],
        proto   => tcp,
        action  => accept,
    }
    include ssmtp
}
```

Ausflug: puppet (V)

```
/etc/puppet/modules/apt_conf/manifests/init.pp
```

```
class apt_conf {  
  file {'/etc/apt/sources.list':  
    source => 'puppet:///apt_conf/sources.list',  
  } ->  
  
  file {'/etc/apt/apt.conf.d/80proxy':  
    source => 'puppet:///apt_conf/apt_proxy',  
  } ->  
  
  exec {'APT_Update':  
    command => '/usr/bin/apt-get update',  
    require => File['/etc/apt/sources.list']  
  }  
}
```

/etc/puppet/modules/ssmtp/manifests/config.pp

```
class ssmtp::config {
  File {
    owner => "root",
    group => "root",
    mode  => 0644,
  }

  file {'/etc/ssmtp/ssmpt.conf':
    ensure => present,
    content => template("ssmtp/ssmtp.conf.erb"),
    require => Class["ssmtp::install"],
  }
}
```



```
/etc/puppet/modules/ssmtp/templates/ssmtp.conf.erb
```

```
root=postmaster  
mailhub=mail.quux.de  
rewriteDomain=quux.de  
hostname= <%= hostname %>  
FromLineOverride=yes
```

Übersicht

- 1 Virtualisierung
- 2 kvm / libvirt
- 3 Automatisierung
- 4 Infrastruktur**
- 5 ToDo
- 6 ToDo

Bei mir auf dem VM-Server:

- recursive DNS (unbound)
- Proxy (apt-cacher-ng)
- puppet Sever (VM)
- Monitoring (externer Server + munin)

Übersicht

- 1 Virtualisierung
- 2 kvm / libvirt
- 3 Automatisierung
- 4 Infrastruktur
- 5 ToDo**
- 6 ToDo

Was muss ich noch alles tun:

- Mehr Automatisieren!
- Besseres Monitoring (Anfänge eine Nagios Plugins sind da)
- OpenVSwitch einbinden
- Wrapper für virt-install: Automatischer Eintrag ins DNS
- Snapshots einrichten
- Ziel: Alles über puppet

Übersicht

- 1 Virtualisierung
- 2 kvm / libvirt
- 3 Automatisierung
- 4 Infrastruktur
- 5 ToDo
- 6** ToDo

- KSM - Kernel Samepage Merging
- Datenbackup: gpg Verschlüsselt mit obnam in die Cloud