

Sicherung von KVM-virtualisierten Maschinen

Christopher Beppler

25. September 2012



Gliederung

- 1 **Einleitung**
 - Sicherungsmethoden
 - Umgebung

Gliederung

1 Einleitung

- Sicherungsmethoden
- Umgebung

2 Snapshot-Sicherung

- Überlegungen
- Realisierung
- Fazit

Gliederung

1 Einleitung

- Sicherungsmethoden
- Umgebung

2 Snapshot-Sicherung

- Überlegungen
- Realisierung
- Fazit

3 Base-Sicherung

- Überlegungen
- Realisierung
- Fazit

Gliederung

1 Einleitung

- Sicherungsmethoden
- Umgebung

2 Snapshot-Sicherung

- Überlegungen
- Realisierung
- Fazit

3 Base-Sicherung

- Überlegungen
- Realisierung
- Fazit

4 Fazit

Einleitung

Zitat

Amy, technology isn't intrinsically good or evil. It's how it's used. Like the Death Ray. — Hubert J. Farnsworth

Sicherungsmethoden

Sicherungsmethoden

■ Snapshot-Sicherung

Sicherungsmethoden

- Snapshot-Sicherung
 - KVM-Funktionalität *suspend*

Sicherungsmethoden

- Snapshot-Sicherung
 - KVM-Funktionalität *suspend*
 - LVM-Snapshot-Funktionalität

Sicherungsmethoden

- Snapshot-Sicherung
 - KVM-Funktionalität *suspend*
 - LVM-Snapshot-Funktionalität
- wie physikalische Maschine

Sicherungsmethoden

- Snapshot-Sicherung
 - KVM-Funktionalität *suspend*
 - LVM-Snapshot-Funktionalität
- wie physikalische Maschine
 - Klassisch

Sicherungsmethoden

- Snapshot-Sicherung
 - KVM-Funktionalität *suspend*
 - LVM-Snapshot-Funktionalität
- wie physikalische Maschine
 - Klassisch
 - Deduplizierung durch Baculas *Base-Backups*

Umgebung

Umgebung

- Test- und Entwicklungsumgebungen für aktuelle Projekte

Umgebung

- Test- und Entwicklungsumgebungen für aktuelle Projekte
 - mindestens 3 Systeme pro Umgebung

Umgebung

- Test- und Entwicklungsumgebungen für aktuelle Projekte
 - mindestens 3 Systeme pro Umgebung
 - gleiches Betriebssystem

Umgebung

- Test- und Entwicklungsumgebungen für aktuelle Projekte
 - mindestens 3 Systeme pro Umgebung
 - gleiches Betriebssystem
 - sehr ähnliche Installationsbasis

Umgebung

- Test- und Entwicklungsumgebungen für aktuelle Projekte
 - mindestens 3 Systeme pro Umgebung
 - gleiches Betriebssystem
 - sehr ähnliche Installationsbasis
- Einsatz von CentOS 5.x und 6.x

Umgebung

- Test- und Entwicklungsumgebungen für aktuelle Projekte
 - mindestens 3 Systeme pro Umgebung
 - gleiches Betriebssystem
 - sehr ähnliche Installationsbasis
- Einsatz von CentOS 5.x und 6.x
- Virtualisierung durch KVM



Umgebung

Umgebung

- Logical Volumes zur Festplattenemulation

Umgebung

- Logical Volumes zur Festplattenemulation
- Eigenes Logical Volume für Swap-Partition

Umgebung

- Logical Volumes zur Festplattenemulation
- Eigenes Logical Volume für Swap-Partition
- Feste Namenskonvention für verwendete Logical Volumes

Umgebung

- Logical Volumes zur Festplattenemulation
- Eigenes Logical Volume für Swap-Partition
- Feste Namenskonvention für verwendete Logical Volumes
 - Präfix `vm_`

Umgebung

- Logical Volumes zur Festplattenemulation
- Eigenes Logical Volume für Swap-Partition
- Feste Namenskonvention für verwendete Logical Volumes
 - Präfix `vm_`
 - `Hostname`

Umgebung

- Logical Volumes zur Festplattenemulation
- Eigenes Logical Volume für Swap-Partition
- Feste Namenskonvention für verwendete Logical Volumes
 - Präfix `vm_`
 - `Hostname`
 - Suffix `-diskn` für Festplatten-Volumes und `-swapn` für Swap-Volumes

Umgebung

- Logical Volumes zur Festplattenemulation
- Eigenes Logical Volume für Swap-Partition
- Feste Namenskonvention für verwendete Logical Volumes
 - Präfix `vm_`
 - `Hostname`
 - Suffix `-diskn` für Festplatten-Volumes und `-swapn` für Swap-Volumes
 - Beispiel: `vm_raun-backup-1-disk0`

Umgebung

- Logical Volumes zur Festplattenemulation
- Eigenes Logical Volume für Swap-Partition
- Feste Namenskonvention für verwendete Logical Volumes
 - Präfix `vm_`
 - `Hostname`
 - Suffix `-diskn` für Festplatten-Volumes und `-swapn` für Swap-Volumes
 - Beispiel: `vm_raun-backup-1-disk0`
- Alle zu sichernden Volumes passen auf Pattern
`/dev/$(hostname)/vm_*-disk*`

Snapshot-Sicherung

Zitat

Simple things should be simple; complex things should be possible.

— Alan Kay



Überlegungen

Überlegungen

- Suspend würde Maschine anhalten und die Erreichbarkeit beeinflussen.

Überlegungen

- Suspend würde Maschine anhalten und die Erreichbarkeit beeinflussen.
- LVM Snapshots haben keinen Einfluss auf Erreichbarkeit.

Überlegungen

- Suspend würde Maschine anhalten und die Erreichbarkeit beeinflussen.
- LVM Snapshots haben keinen Einfluss auf Erreichbarkeit.
- Keine Notwendigkeit für `bacula-fd` auf VM.

Überlegungen

- Suspend würde Maschine anhalten und die Erreichbarkeit beeinflussen.
- LVM Snapshots haben keinen Einfluss auf Erreichbarkeit.
- Keine Notwendigkeit für `bacula-fd` auf VM.
- Leichte Automatisierung durch Namenskonvention.

Realisierung

Realisierung

- Backup-Jobs auf Hypervisor, nicht auf VM selbst.

Realisierung

- Backup-Jobs auf Hypervisor, nicht auf VM selbst.
- Einzelner Bacula-Job pro VM.

Realisierung

- Backup-Jobs auf Hypervisor, nicht auf VM selbst.
- Einzelner Bacula-Job pro VM.
 - 1 ClientRunBeforeJob zur Erzeugung des Snapshots-Volumes.

Realisierung

- Backup-Jobs auf Hypervisor, nicht auf VM selbst.
- Einzelner Bacula-Job pro VM.
 - 1 ClientRunBeforeJob zur Erzeugung des Snapshots-Volumes.
 - 2 Sicherung der Snapshot-Volumes.

Realisierung

- Backup-Jobs auf Hypervisor, nicht auf VM selbst.
- Einzelner Bacula-Job pro VM.
 - 1 ClientRunBeforeJob zur Erzeugung des Snapshots-Volumes.
 - 2 Sicherung der Snapshot-Volumes.
 - 3 ClientRunAfterJob zur Entfernung des Snapshot-Volumes.



Realisierung

Realisierung

```
VG= ' '$(hostname) ' '
```

```
VM= ' '$1 ' '
```

Realisierung

```
VG=' '$(hostname) '
```

```
VM=' '$1' ' — Umsetzung von %n, Job Name
```

Realisierung

```
VG=' '$(hostname)'
```

```
VM=' '$1' — Umsetzung von %n, Job Name
```

```
for i in `ls /dev/$VG/vm_$VM-disk*` ; do
```

Realisierung

```
VG='$(hostname)'
```

```
VM='$(1)' — Umsetzung von %n, Job Name
```

```
for i in `ls /dev/$VG/vm_$VM-disk*` ; do  
    lvcreate -s "$i" -n "snap_$i" -L 5G
```

Realisierung

```
VG='$(hostname)'
```

```
VM='$(1)' — Umsetzung von %n, Job Name
```

```
for i in `ls /dev/$VG/vm_$VM-disk*` ; do  
    lvcreate -s "$i" -n "snap_$i" -L 5G  
done
```

Realisierung

```
VG='$(hostname)'
```

```
VM='$(1)' — Umsetzung von %n, Job Name
```

```
for i in `ls /dev/$VG/vm_$VM-disk*` ; do  
    lvcreate -s `ls $i` -n `snap_$i` -L 5G  
done
```

Full-Backup aller Volumes, die auf Pattern
`/dev/*/snap_vm-*-disk*` passen.



Fazit – Vorteile

Fazit – Vorteile

- Funktioniert mit jedem virtualisierten Betriebssystem.

Fazit – Vorteile

- Funktioniert mit jedem virtualisierten Betriebssystem.
- Keine besondere Einrichtung auf VM notwendig.

Fazit – Vorteile

- Funktioniert mit jedem virtualisierten Betriebssystem.
- Keine besondere Einrichtung auf VM notwendig.
- Keine Einschränkung der Erreichbarkeit der VM.

Fazit – Nachteile

Fazit – Nachteile

- Backupdauer sehr lang.

Fazit – Nachteile

- Backupdauer sehr lang.
- Backups sehr groß.

Fazit – Nachteile

- Backupdauer sehr lang.
- Backups sehr groß.
- Jedes Backup ist ein Full-Backup.

Fazit – Nachteile

- Backupdauer sehr lang.
- Backups sehr groß.
- Jedes Backup ist ein Full-Backup.
- Restore einzelner Dateien nur umständlich und von Hand möglich.

Base-Sicherung

Zitat

Elegance is not a dispensable luxury but a quality that decides between success and failure. — Edsger W. Dijkstra



Überlegungen

Überlegungen

- Eingesetzte Systeme sind sich im Grunde sehr ähnlich.

Überlegungen

- Eingesetzte Systeme sind sich im Grunde sehr ähnlich.
- Betriebssysteme werden (automatisch) aktualisiert.

Überlegungen

- Eingesetzte Systeme sind sich im Grunde sehr ähnlich.
- Betriebssysteme werden (automatisch) aktualisiert.
- Klassische Base-Deduplizierung verliert mit der Zeit an Effizienz. Basis weicht immer weiter von aktuellem Stand ab.

Überlegungen

- Eingesetzte Systeme sind sich im Grunde sehr ähnlich.
- Betriebssysteme werden (automatisch) aktualisiert.
- Klassische Base-Deduplizierung verliert mit der Zeit an Effizienz. Basis weicht immer weiter von aktuellem Stand ab.
- KVM ermöglicht direktes Booten eines Linux-Kernels

Überlegungen

- Eingesetzte Systeme sind sich im Grunde sehr ähnlich.
- Betriebssysteme werden (automatisch) aktualisiert.
- Klassische Base-Deduplizierung verliert mit der Zeit an Effizienz. Basis weicht immer weiter von aktuellem Stand ab.
- KVM ermöglicht direktes Booten eines Linux-Kernels
⇒ direkter Aufruf der Installation mit Kickstart-Datei.

Überlegungen

Überlegungen

- RPM-Inhalte haben auf jedem System gleiche ctime, mtime, atime und Dateiberechtigungen.

Überlegungen

- RPM-Inhalte haben auf jedem System gleiche ctime, mtime, atime und Dateiberechtigungen.
⇒ Funktioniert mit Accurate-Funktionalität ohne aufwendige Prüfsummenberechnung.

Überlegungen

- RPM-Inhalte haben auf jedem System gleiche ctime, mtime, atime und Dateiberechtigungen.
⇒ Funktioniert mit Accurate-Funktionalität ohne aufwendige Prüfsummenberechnung.
- Eventuelle Symlink-Inhalte haben den jeweiligen Zeitstempel der Installation.

Überlegungen

- RPM-Inhalte haben auf jedem System gleiche ctime, mtime, atime und Dateiberechtigungen.
⇒ Funktioniert mit Accurate-Funktionalität ohne aufwendige Prüfsummenberechnung.
- Eventuelle Symlink-Inhalte haben den jeweiligen Zeitstempel der Installation.
Dies kann allerdings vernachlässigt werden, da die Dateigröße sehr gering ist.

Realisierung

Realisierung

- Pro Betriebssystemversion eigene VM.

Realisierung

- Pro Betriebssystemversion eigene VM.
- Pro Betriebssystemversion eigene Kickstartdatei mit Minimalinstallation.

Realisierung

- Pro Betriebssystemversion eigene VM.
- Pro Betriebssystemversion eigene Kickstartdatei mit Minimalinstallation.
- Einmal monatlich wird eine Referenz-VM des jeweiligen Betriebssystems neu installiert und auf den aktuellen Stand gepatcht.

Realisierung

- Pro Betriebssystemversion eigene VM.
- Pro Betriebssystemversion eigene Kickstartdatei mit Minimalinstallation.
- Einmal monatlich wird eine Referenz-VM des jeweiligen Betriebssystems neu installiert und auf den aktuellen Stand gepatcht.
- Anschließend läuft ein Base-Job für diese Referenz-VM.

Realisierung

- Pro Betriebssystemversion eigene VM.
- Pro Betriebssystemversion eigene Kickstartdatei mit Minimalinstallation.
- Einmal monatlich wird eine Referenz-VM des jeweiligen Betriebssystems neu installiert und auf den aktuellen Stand gepatcht.
- Anschließend läuft ein Base-Job für diese Referenz-VM.
- Backup-Jobs enthalten die jeweils passende Referenz-VM.

Realisierung

Realisierung

Job-Definition

```
Job {  
    Name = van-aps-1
```

Realisierung

Job-Definition

```
Job {  
    Name = van-aps-1  
    Base =
```

Realisierung

Job-Definition

```
Job {  
    Name = van-aps-1  
    Base = van-aps-1
```

Realisierung

Job-Definition

```
Job {  
    Name = van-aps-1  
    Base = van-aps-1, CentOS6
```


Realisierung

Job-Definition

```
Job {  
    Name = van-aps-1  
    Base = van-aps-1, CentOS6  
    Accurate = yes
```

Realisierung

Job-Definition

```
Job {  
    Name = van-aps-1  
    Base = van-aps-1, CentOS6  
    Accurate = yes  
    ...  
}
```

Realisierung

Job-Definition

```
Job {  
    Name = van-aps-1  
    Base = van-aps-1, CentOS6  
    Accurate = yes  
    ...  
}
```

CentOS6 ist der Job, der die Referenz-VM sichert.

Realisierung

Automatische Installation der Referenz-VM

Realisierung

Automatische Installation der Referenz-VM

```
/usr/libexec/qemu-kvm \
```

Realisierung

Automatische Installation der Referenz-VM

```
/usr/libexec/qemu-kvm \  
-kernel /media/centos6/isolinux/vmlinuz \  

```

Realisierung

Automatische Installation der Referenz-VM

```
/usr/libexec/qemu-kvm \  
-kernel /media/centos6/isolinux/vmlinuz \  
-initrd /media/centos6/isolinux/initrd.img \  

```

Realisierung

Automatische Installation der Referenz-VM

```
/usr/libexec/qemu-kvm \  
-kernel /media/centos6/isolinux/vmlinuz \  
-initrd /media/centos6/isolinux/initrd.img \  
-cmdline ks=http://10.0.0.180/ks/centos6-min.cfg \  

```


Realisierung

Automatische Installation der Referenz-VM

```
/usr/libexec/qemu-kvm \  
-kernel /media/centos6/isolinux/vmlinuz \  
-initrd /media/centos6/isolinux/initrd.img \  
-cmdline ks=http://10.0.0.180/ks/centos6-min.cfg \  
...
```

Realisierung

Automatische Installation der Referenz-VM

```
/usr/libexec/qemu-kvm \  
-kernel /media/centos6/isolinux/vmlinuz \  
-initrd /media/centos6/isolinux/initrd.img \  
-cmdline ks=http://10.0.0.180/ks/centos6-min.cfg \  
...
```

Wird als Bestandteil der Sicherung der Referenz-VM gestartet.

Fazit – Vorteile

Fazit – Vorteile

- Reduzierte Backup-Größe.

Fazit – Vorteile

- Reduzierte Backup-Größe.
- Deduplizierungs Nachteile werden durch monatliche Aktualisierung der Referenz-VM reduziert.

Fazit – Nachteile

Fazit – Nachteile

- Benötigt Ressourcen für Referenz-VMs.

Fazit – Nachteile

- Benötigt Ressourcen für Referenz-VMs.
- Bei zu unterschiedlichen Systemen kein Vorteil gegenüber *klassischer* Deduplizierung.

Fazit

Zitat

I refuse to answer that question on the grounds that I don't know the answer. — Zaphod Beeblebrox

Fazit

Fazit

- Snapshot-Sicherung lohnt sich nur in Ausnahmefällen.

Fazit

- Snapshot-Sicherung lohnt sich nur in Ausnahmefällen.
- Für viele gleichartige Systeme lohnt sich das Pflegen einer Referenz-VM.

Fazit

- Snapshot-Sicherung lohnt sich nur in Ausnahmefällen.
- Für viele gleichartige Systeme lohnt sich das Pflegen einer Referenz-VM.
- Weichen die Systeme zu weit von einander ab, lohnt sich die feinere Unterteilung der Referenz-VMs (z.B. *centos6-min*, *centos6-ws*, *centos6-db*, ...).

Fazit

- Snapshot-Sicherung lohnt sich nur in Ausnahmefällen.
- Für viele gleichartige Systeme lohnt sich das Pflegen einer Referenz-VM.
- Weichen die Systeme zu weit von einander ab, lohnt sich die feinere Unterteilung der Referenz-VMs (z.B. *centos6-min*, *centos6-ws*, *centos6-db*, ...).
- Lokale Spiegelung des CentOS-Repositories verringert die Installationsdauer und spart Bandbreite.

Fazit

- Snapshot-Sicherung lohnt sich nur in Ausnahmefällen.
- Für viele gleichartige Systeme lohnt sich das Pflegen einer Referenz-VM.
- Weichen die Systeme zu weit von einander ab, lohnt sich die feinere Unterteilung der Referenz-VMs (z.B. *centos6-min*, *centos6-ws*, *centos6-db*, ...).
- Lokale Spiegelung des CentOS-Repositories verringert die Installationsdauer und spart Bandbreite.
- Ähnliche Resultate ließen sich auch mit Debian-basierten Distributionen erreichen (debootstrap).

Vielen Dank für Ihre Aufmerksamkeit!