Tutorial Virtualisierung mit ArchivistaMini

Egg, 3. Oktober 2016: *Der nachfolgende Blog gibt einen allgemein verständlichen Einstieg in die Welt der Virtualisierung mit ArchivistaMini 2016/X. Dabei wird aufgezeigt, was Virtualisierung ist, welche Arten der Virtualisierung es gibt, was für eine Rolle ArchivistaMini dabei spielt und wie ArchivistaMini auf- und eingesetzt wird. Und wer gleich zum Einstieg* ***ArchivistaMini beziehen möchte, bitte hier klicken.***

Virtualisierung

Mit **Virtualisierung können dank Software mehrere Rechner (Gäste) auf einem Rechner (Host bzw. Hauptrechner) verwaltet und betrieben werden (mehrere Boxen in einer Box).** Die Gäste verfügen dabei nicht über je eine eigene physikalische Hardware, vielmehr wird den Gästen eine solche Hardware nur 'vorgegaukelt'. Dies immerhin derart gut, dass die Gäste-Rechner bzw. die darauf laufenden Betriebssysteme und Applikationen den 'Zaubertrick' nicht derart durchschauen, dass sie die Arbeit verweigern.

Virtualisierung kann praktisch unter jedem Betriebssystem zur Anwendung gelangen. Dabei gibt es **mehrere Konzepte.** Einmal kann ein **gesamter Rechner (Gast) mit Unterstützung der Hardware (CPU) virtualisiert werden** (Beispiele: KVM, VMware ESXi, Microsoft Hypervisor). Zwar können damit nur Gäste virtualisiert werden, die kompatibel zur gleichen CPU sind (ein Handy-OS mit ARM geht damit z.B. mit einer Intel/AMD-CPU nicht), dafür ist die Ausführungsgeschwindigkeit sehr gut.

Bei der **zweiten Art** der Virtualisierung wird der **Gast ohne Unterstützung der CPU virtualisiert** (Beispiel: QEMU). Dies hat den Vorteil, dass beliebige Rechner auf einer beliebigen Hardware laufen können, als Nachteil resultiert allerdings eine geringere Ausführungsgeschwindigkeit.

Bei der **dritten Variante** werden die **Gäste nur softwaretechnisch aufgebaut,** hier ist oft von Containern (z.B. LXC) die Rede. Vorteil dieses Art von Virtualisierung ist, dass sehr geringe Ressourcen benötigt werden, doch

wird dies damit erkauft, dass die Container-Gäste faktsich nicht vom Haupt-Rechner isoliert sind und dass bei späteren Updates zwischen dem Rechner für die Virtualisierung und den Gästen diese immer synchron geführt werden müssen, d.h die Gäste können gerade nicht mit je unterschiedlichen Betriebssystemen betrieben werden.
Im Prinzp kann Virtualsierung auf jedem Betriebssystem ausgeführt werden. Dennoch bietet Open Source bzw. Linux einen entscheidenden Vorteil, weil bereits das Betriebssystem genau so konfiguriert werden kann, wie es für die Virtualisierung passt. Dies ist beim 'Platzhirsch' VMware und in der Windows-Welt bei Hypervisor nicht bzw. nur sehr eingeschränkt möglich.

KVM und QEMU mit ArchivistaMini

Wie eingangs erwähnt, realisiert ArchivistaMini Virtualisierung mit KVM, wobei es dem anzufügen gilt, das KVM und QEMU seit einigen Jahren in einem Paket ausgeliefert werden. Da **KVM deutlich schneller als QEMU arbeitet, werden Gäste normalerweise mit KVM betrieben,** QEMU hilft dann weiter, wenn Gäste zum Einsatz kommen sollen, die auf der vorhandenen Hardware gar nicht erst laufen würden. Dies ist insbesondere dann von Vorteil, wenn Betriebssysteme aus fernen Zeiten (z.B. Windows95 oder ein Linux mit Kernel 2.4.x) gestartet werden sollen.
Nun kann QEMU und KVM praktisch auf jedem Rechner (selbst unter Windows) installiert werden. Auf den ersten Blick erscheint eine eigene Distribution wie ArchivistaMini alleine für die Virtualisierung daher nicht zwingend. Es genügt(e) an sich, zu einem beliebigen Betriebssystem (z.B. Debian) zu greifen, das System zu installieren und danach die Software zur Virtualisierung aufzuspielen. Faktisch ist genau dies aber nicht ganz trivial. Bis z.B. Debian aufgespielt ist, vergehen gut und gerne 20 bis 30 Minuten, ehe das Grundsystem installiert ist. Auch wenn Linux nach wie vor ohne grafische Oberfläche installiert werden kann, so sind installierte Instanzen mittlerweile alles andere als klein. Natürlich sind ein oder zwei GByte mit den heutigen Bordmitteln schon verkraftbar, doch elegant ist anders. Doch auch bei der anschliessendes Installation von qemu fragt es sich, warum mit 'apt-get install qemu' nochmalig 200 MByte an Software notwendig sind:

```
archivista@avbox179: ~
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  ipxe-qemu libbluetooth3 libboost-thread1.55.0 libbrlapi0.6 libfdt1 libiscsi2
  librados2 librbd1 libseccomp2 libspice-server1 libusbredirparser1
  libvdeplug2 libxen-4.4 libxenstore3.0 openbios-ppc openbios-sparc
  openhackware qemu-slof qemu-system qemu-system-arm qemu-system-common
  qemu-system-mips qemu-system-misc qemu-system-ppc qemu-system-sparc
  qemu-system-x86 qemu-user qemu-utils seabios
Suggested packages:
  qemu-user-static vde2 sgabios ovmf debootstrap
Recommended packages:
  qemu-user-binfmt sharutils
The following NEW packages will be installed:
  ipxe-qemu libbluetooth3 libboost-thread1.55.0 libbrlapi0.6 libfdt1 libiscsi2
  librados2 librbd1 libseccomp2 libspice-server1 libusbredirparser1
  libvdeplug2 libxen-4.4 libxenstore3.0 openbios-ppc openbios-sparc
  openhackware qemu qemu-slof qemu-system qemu-system-arm qemu-system-common
  qemu-system-mips qemu-system-misc qemu-system-ppc qemu-system-sparc
  qemu-system-x86 qemu-user qemu-utils seabios
0 upgraded, 30 newly installed, 0 to remove and 9 not upgraded.
Need to get 26.5 MB of archives.
After this operation, 200 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Warum z.B. libxenstore3.0 und andere Kandidaten installiert werden sollen, bleibt unverständlich. Es soll ja nicht XEN zum Einsatz kommen, sondern KVM.

Doch selbst wenn qemu installiert ist, und Gäste eingerichtet werden können, ist der Aufruf von KVM mit vielen Parametern auf der Konsole zu starten, dazu ein Beispiel:

```
/usr/bin/kvm -monitor unix:/var/run/qemu-server/101.mon,server,nowait -monitor telnet:127.0.0.1:1,server,nowait -display vnc=127.0.0.1:-5900,password -pidfile /var/run/qemu-server/101.pid -daemonize -usbdevice tablet -name test -smp sockets=1,cores=1 -boot c -vga cirrus -tdf -enable-kvm -k de-ch -drive file=/var/lib/vz/images/101/vm-101-disk-1.raw,if=ide,index=0,cache=writeback -drive file=/var/lib/vz/template/iso/avtest1.iso,if=ide,index=2,media=cdrom,cache=writeback -m 2048 -net tap,vlan=0,ifname=vmtab101i0,script=/var/lib/qemu-server/bridge-vlan0 -net nic,vlan=0,model=rtl8139,macaddr=EA:BF:94:E3:52:79
```

Damit ist die Instanz zwar gestartet, aber um auf die Gäste zuzugreifen, sind weitere Pakete notwendig. Benötigt wird nun ein kompletter X-Server sowie VNC, um den Bildschirm eines Gastes darstellen zu können. Sowohl die Installation des X-Servers als auch von VNC, dies alles ist nicht ganz einfach, vorallem aber kostet es viel Zeit. **Selbst Profis sind mit einer manuellen Installation gut und gerne einen halben Tag oder mehr beschäftigt.**

ArchivistaMini = 90 MByte und einige Minuten

Nun existieren zu KVM und QEMU einige grafische (webbasierte) Konsolen (GUIs), welche das Zusammenstellen einer Instanz vereinfachen, allerdings sind die meisten davon nicht für den Allgemeingebrauch verfügbar, für fast alle GUIs sind Lizenzkosten fällig. Eine löbliche Ausnahme stellt hier an sich Proxmox dar. Allerdings liegt der **Fokus von Proxmox heute deutlich stärker bei DataCenter-Lösungen denn bei einer einfachen Lösung für die Virtualisierung.** Dazu stellvertretend die Kommentare bei der Ankündigung von Proxmox 4.2 bei **prolinux.de.** Originalzitat zu einer Frage, warum es nicht möglich sei, Proxmox mit dem grafischen Installer auf eine bestehende Festplatte zu installieren, ohne diese zu zerstören: **"Ich mache immer eine Debian Installation und packe Proxmox erst danach drauf."** Danach erfolgt eine Diskussion über ca. 20 Posts, in denen es z.B. um Fragen um Raid-Levels und Partionierung des Basis-Systems geht. Einfach ist irgendwie anders.

Nun wurde ArchivistaMini im Jahre 2009 eben gerade aus diesen Gründen von Proxmox 'geforkt', ein Update mit 'apt-get update', 'apt-get upgrade' oder 'apt-get dist-upgrade' auf der Konsole ist und bleibt für Laien zu komplex.

Entstanden ist über die Jahre ArchivistaMini, ein Mini-Server für die Server-Virtualisierung. **Mit gegenwärtig 90 MByte für das gesamte Betriebssystem (Debian), KVM/QEMU, den benötigem Web-Server Apache für die Verwaltung der Konsolen und vielen Tools ist ArchivistaMini um Faktoren kleiner** als es andere Lösungen für die Virtualisierung sind.

Im Unterschied zu anderen Lösungen wird **ArchivistaMini auch nicht installiert, sondern einzig und alleine in den Hauptseicher hochgefahren (RAM-Modus).** Findet ArchivistaMini beim Hochfahren eine leere Festplatte vor, wird diese einegerichtet, ist sie schon eingerichtet, wird sie selbstverständlich nicht 'angetastet'. Der ganze Vorgang dauert keine Minute, der RAM-Bedarf der Distribution liegt installiert bei unter 300 MByte.

Neu in ArchivistaMini 2016/X

Bisher wurde eine jede Installation von ArchivistaMini webbasiert über **shop.archivista.ch** erstellt. Dabei konnten die gewünschten Mac-Adressen für die Netzwerkkarte und die IP-Kenndaten für ein oder zwei Rechner in einem Web-Formular erfasst werden, sodass fixfertig konfigurierte ISO-Dateien erstellt

und bezogen werden konnten.

1 x ArchivistaBox Mini

Please fill in IP specifications (all fields are mandatory):

Mac address:	0a:0b:0c:0d:0e:0f
IP address:	192.168.0.250
Submask:	255.255.255.0
Gateway:	192.168.0.2
DNS server:	192.168.0.2

Auch wenn der Download komplett kostenfrei erfolgen konnte, so schreckte die Registrierung über den Shop dann und wann vom Download ab. Neu kann die Installation auch ohne den (Um-)Weg über den Webshop erfolgen. Die Variante über den Webshops steht weiter zur Verfügung, die Installation eines Clusters z.B. kann webbasiert einfacher nicht realisiert werden. Zurück zur 'manuellen Installation'.

Hochfahren von ArchivistaMini (Installation)

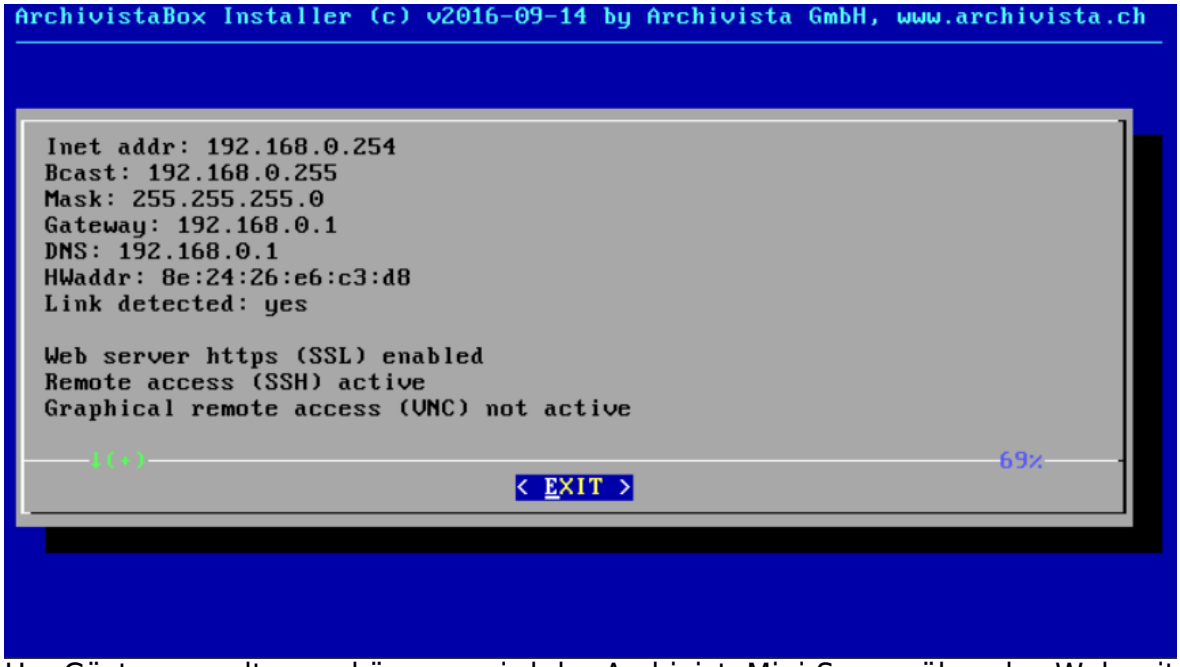
Die ISO-Datei kann unter **https://archivista.ch/avmini.iso** bezogen werden. Für einen ersten Test dürfte ArchivistaMini wohl meist selber virtualisiert aufgesetzt werden. Folglich kann einfach die ISO-Datei hochgefahren werden, ansonten müsste die ISO-Datei auf eine Scheibe 'gebrannt' werden oder ein **bootbarer USB-Stick (mehr dazu hier)** erstellt werden. Auch wenn diese Anleitung Windows-Screens enthält, UNetbootin steht sowohl unter Linux, Windows und Mac zur Verfügung. Alternativ kann die ISO-Datei auch über das Netzwerk (PXE) gestartet werden. Beim **Starten der ISO sind die IP-Kenndaten beim ersten Bildschirm festzulegen.** Dazu ein Beispiel in seiner minimalsten Variante:

ram ip.192.168.0.254

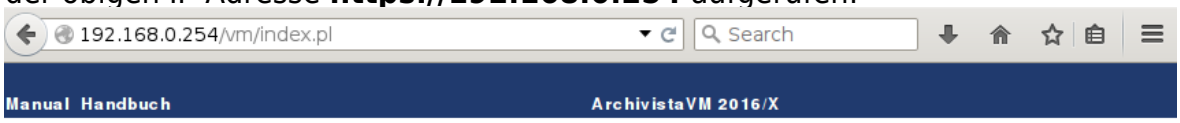
ArchivistaBox 64Bit — in 100 seconds
to a ready to use server infrastructure

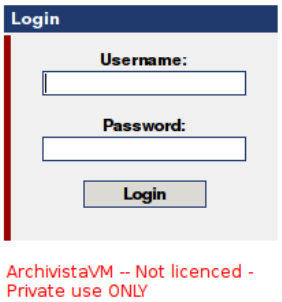
```
Press Return to install ArchivistaBox Environment.
ArchivistaMini (build 2016-10-02) - http://www.archivista.ch

boot: ram ip.192.168.0.254_
```

Diese Konfiguration installiert ArchivistaMini mit einer Netmaske 255.255.255.0, dem Gateway 192.168.0.1 und DNS 192.168.0.1. Wer eine andere Konfiguration wünscht, kann dies selbstverständlich mit angeben:
ram ip.192.168.1.251 submask.255.255.255.0 gw.192.168.1.1 dns.192.168.1.1
Mit der Enter-Taste wird die Installation gestartet. Nach etwa 40 bis 50 Sekunden steht der Server zum Arbeiten bereit. Sofern ArchivistaMini eine leere Festplatte vorfindet, wird diese formatiert. Wer den Zugriff auf Festplatten unterdrücken möchte, kann die Option **ramonly** mit eingeben, ArchivistaMini läuft dann komplett im Hauptspeicher. Um die **deutsche Sprache** in ArchivistaMini zu aktivieren, ist die Option **de** anzugeben.
Wichtig: *Ohne Eingabe der obigen Optionen, startet ArchivistaMini nach 10 Sekunden automatisch, ohne IP-Kenndaten muss die Konfiguration des Netzwerkes dann manuell erledigt werden, dies ist nicht zu empfehlen. Sind die 10 Sekunden beim ersten Anlauf verpasst, so kann mit Ctrl+Alt+Del ein Neustart erzwungen werden, um die IP-Kenndaten wie obenstehend einzugeben.*
Am Ende des Vorganges erscheint der untenstehende Bildschirm:

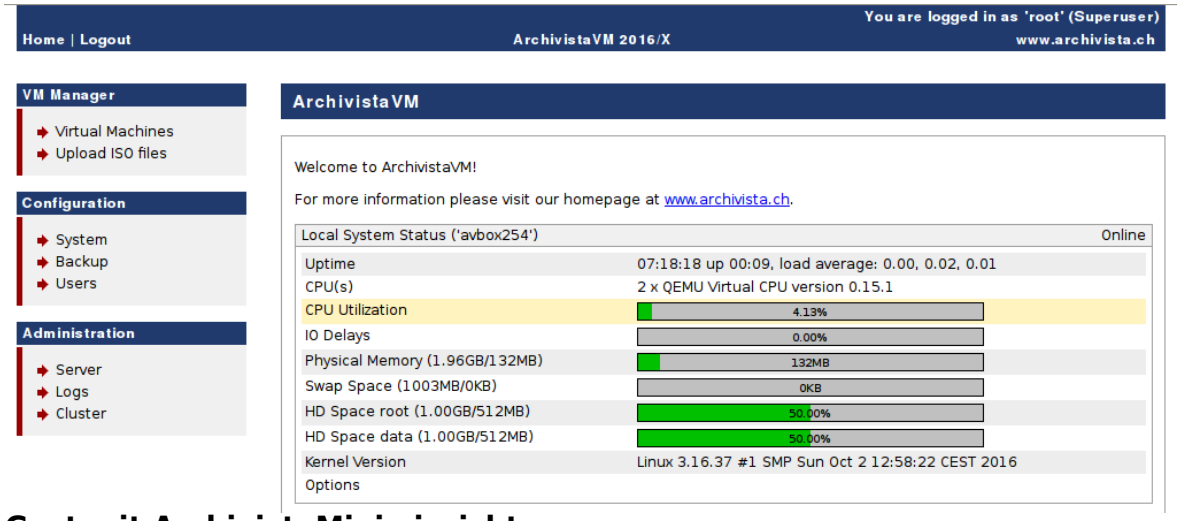


Um Gäste verwalten zu können, wird der ArchivistaMini-Server über das Web mit der obigen IP-Adresse **https://192.168.0.254** aufgerufen:



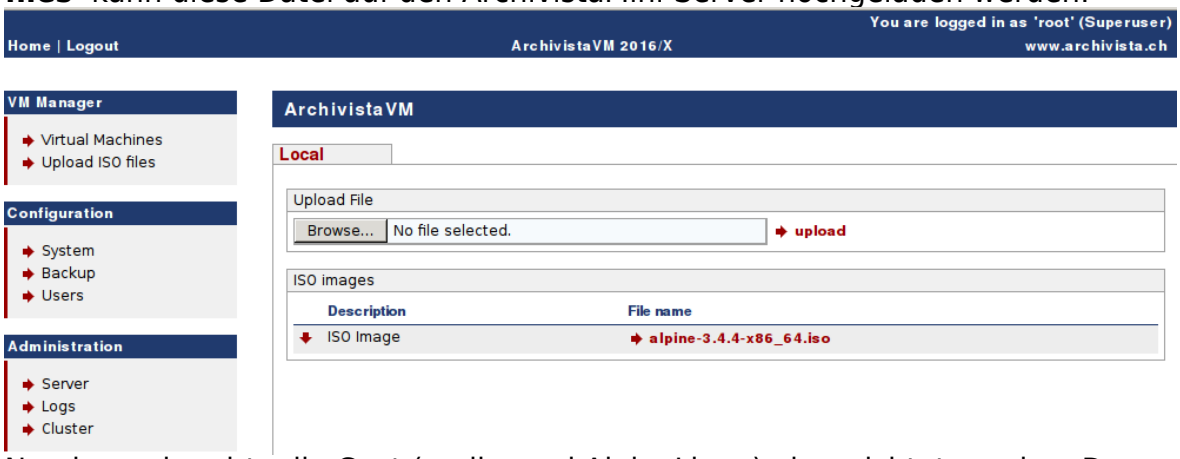


Das Anmelden erfolgt mit dem **Benutzer 'root' und dem Passwort 'archivista',** ehe die folgende Oberfläche erscheint:

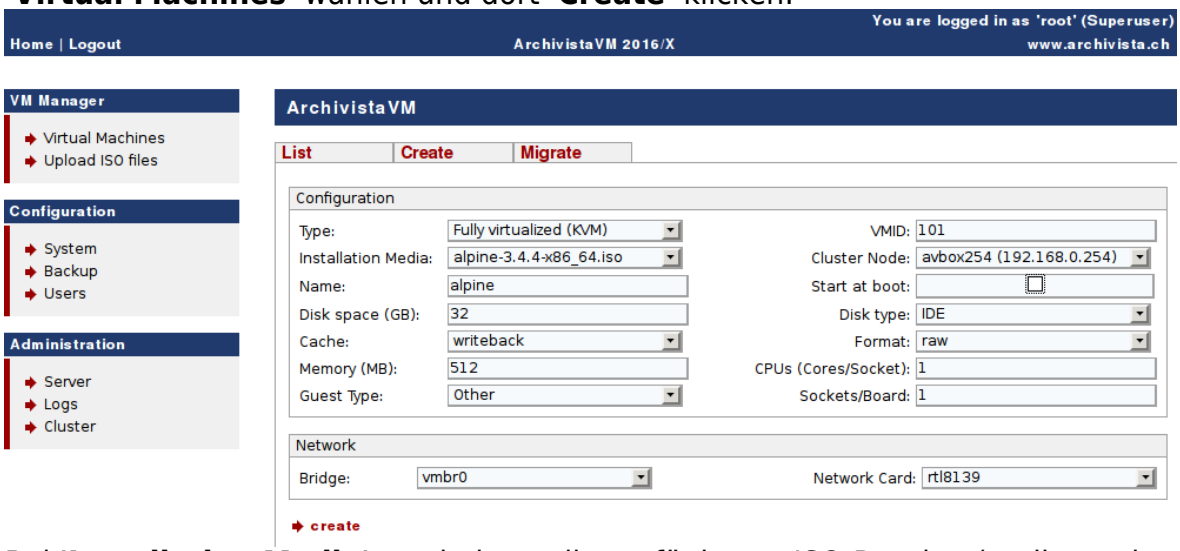


Gast mit ArchivistaMini einrichten

Um einen ersten virtuellen Gast einrichten zu können, wird ein Installationsmedium (ISO-Datei) benötigt. Vorliegend wird dies am Beispiel von Alpine-Linux aufgezeigt. Dazu bitte die ISO-Datei von **alpinelinux.org** beziehen. Aktuell trägt diese den Namen **alpine-3.4.4-x86_64.iso.** Mit **'Upload ISO files'** kann diese Datei auf den ArchivistaMini-Server hochgeladen werden:

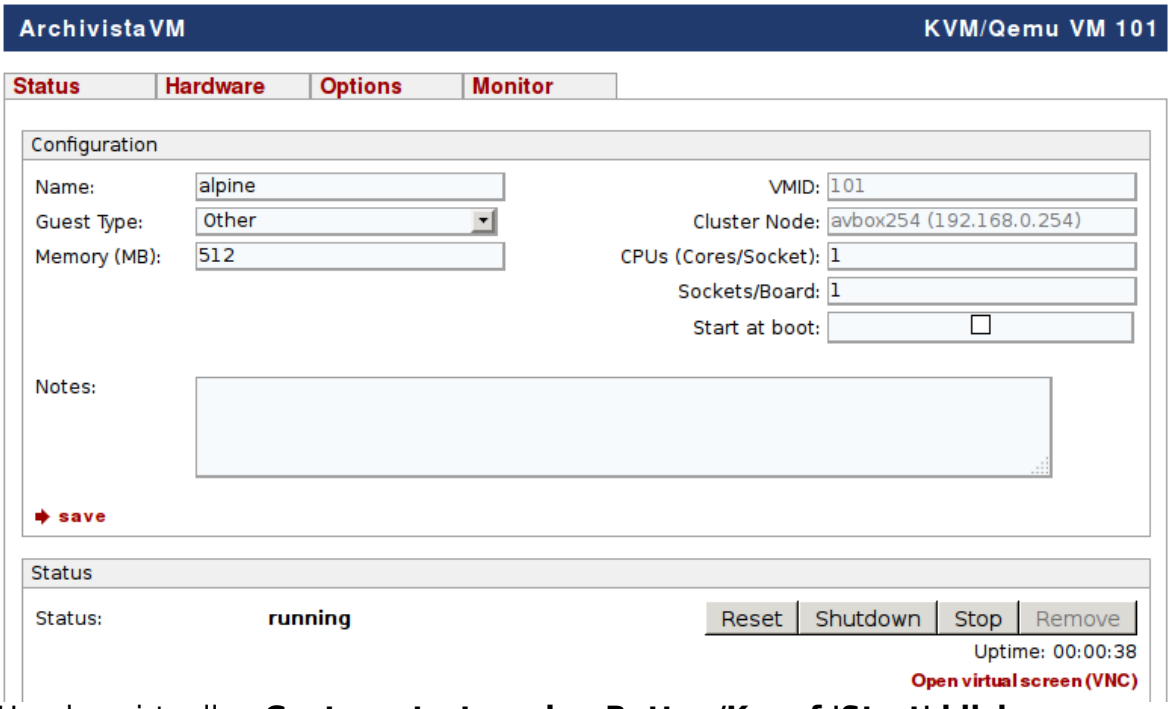


Nun kann der virtuelle Gast (vorliegend AlpineLinux) eingerichtet werden. Dazu **'Virtual Machines'** wählen und dort **'Create'** klicken:

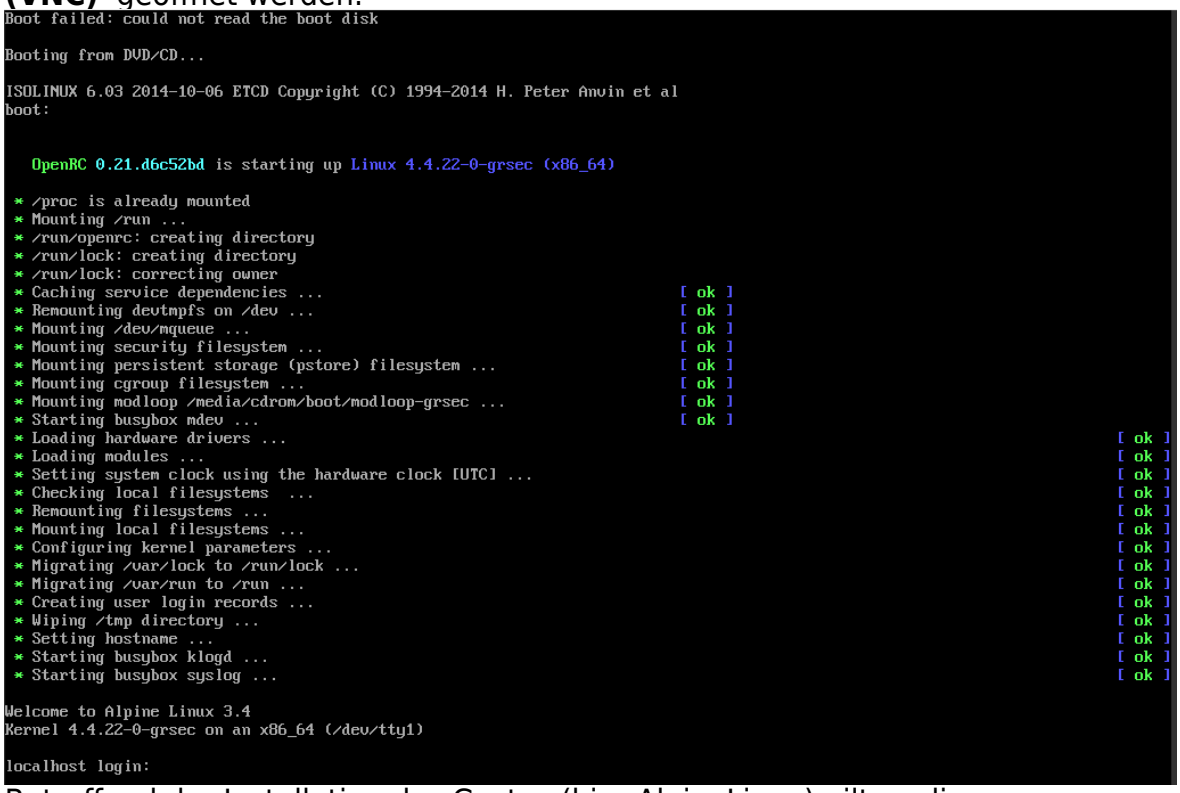


Bei **'Installation Media'** erscheinen alle verfügbaren ISO-Dateien (vorliegend **alpine-3.4.4-x86_64.iso).** Weiter ist zwingend ein Name (hier 'alpinelinux'

einzugeben. Selbstverständlich können die übrigen Parameter wie Hauptspeicher, Anzahl CPU-Kerne sowie die Grösse der Festplatten ebenfalls angepasst werden. Für das vorliegende Beispiel sind die 512 MByte jedoch ausreichend. MIt **'create' wird der virtuelle Rechner eröffnet:**



Um den virtuellen **Gast zu starten, den Button/Knopf 'Start' klicken.** Danach kann der Bildschirm des virtuellen Rechners mit **'Open virtual screen (VNC)'** geöffnet werden:



Betreffend der Installation des Gastes (hier AlpineLinux) gilt es die Dokumentation des Gastes zu konsultieren.

Gäste mit ArchivistaMini auf der Konsole verwalten
Gäste können mit ArchivistaMini nicht nur mit der grafischen Web-Oberfläche, sondern auch auf der Konsole verwaltet werden. Dabei gibt es zwei Möglichkeiten. Entweder wird auf dem Startbildschirm von ArchivistaMini der Knopf 'EXIT' gedrückt, um lokal mit der Konsole zu arbeiten, oder es erfolgt ein **Zugriff mit ssh (vorligend 'ssh 192.168.0.254').** Nachfolgend dazu die wichtigsten Befehle, die wir alle mit **'qm info'** erfragen können:

```
root@avbox254:~# qm info
qm <command> <vmid> [OPTIONS]
qm [create|set] <vmid>
        --memory  <MBYTES>    memory in MB (64 - 8192)
        --smp  <N>            set number of CPUs (Cores) to
<N>
        --sockets  <N>        set number of Sockets/Board to
<N>
        --ostype NAME         specify OS type
        --onboot [yes|no]     start at boot
        --keyboard XX         set vnc keyboard layout
        --cpuunits <num>      CPU weight for a VM
        --name <text>         set a name for the VM
        --description <text>  set VM description
        --boot [a|c|d|n]      specify boot order
        --bootdisk <disk>     enable booting from <disk>
        --acpi (yes|no)       enable/disable ACPI
        --kvm (yes|no)        enable/disable KVM
        --tdf (yes|no)        enable/disable time drift fix
        --localtime (yes|no)  set the RTC to local time
        --vga (gd5446|vesa)   specify VGA type

        --vlan[0-9u]
MODEL=XX:XX:XX:XX:XX:XX[,MODEL=YY:YY:YY:YY:YY:YY]

        --ide<N>    [file=]file,][,media=d]
                    [,cyls=c,heads=h,secs=s[,trans=t]]
[,snapshot=on|off][,cache=on|off][,format=f]
        --ide<N> <GBYTES>     create new disk
        --format <format>     qcow|raw => type of disk
format
        --cache <cache>       writeback|writethrough|none
        --ide<N> delete       delete disk
        --cdrom <file>        is an alias for --ide2
<file>,media=cdrom

        --scsi<N>   [file=]file,][,media=d]
                    [,cyls=c,heads=h,secs=s[,trans=t]]
[,snapshot=on|off][,cache=on|off][,format=f]
        --scsi<N> <GBYTES>    create new disk
        --scsi<N> delete      delete disk

        --virtio<N> [file=]file,][,media=d]
                    [,cyls=c,heads=h,secs=s[,trans=t]]
[,snapshot=on|off][,cache=on|off][,format=f]
        --virtio<N> <GBYTES>  create new disk
        --virtio<N> delete    delete disk

qm monitor <vmid>       connect to vm control monitor
```

```
qm start <vmid>          start vm
qm shutdown <vmid>       gracefully stop vm (send poweroff)
qm wait <vmid> [time]    wait until vm is stopped
qm stop <vmid>           kill vm (immediate stop)
qm reset <vmid>          reset vm (stop, start)
qm suspend <vmid>        suspend vm
qm resume <vmid>         resume vm
qm cad <vmid>            sendkey ctrl-alt-delete
qm destroy <vmid>        destroy vm (delete all files)
qm status <vmid>         shows the container status

qm cdrom <vmid> [] <path>  set cdrom path. <device is ide2
by default>
qm cdrom <vmid> [] eject   eject cdrom

qm unlink <vmid> <file>  delete unused disk images
qm vncproxy <vmid> <ticket>  open vnc proxy
qm vnc <vmid>            start (X11) vncviewer (experimental)
qm showcmd <vmid>        show command line (debug info)
qm list                  list all virtual machines

qm startall              start all virtual machines (when
onboot=1)
qm stopall [timeout]     stop all virtual machines (default
timeout is 3 minutes)
```

Das Arbeiten mit der Konsole ist selbstverständlich freiwillig, alle Befehle lassen sich auch über das Web-Interface realisieren. Die wichtigsten Konsole-Befehle seien hier anhand der erstellten Instanz 101 von AlpineLinux kurz aufgezeigt. Mit **qm start 101** wird ein Gast gestartet, mit **qm stop 101** unsanft gestoppt und mit **qm list** kann der Zustand aller Gäste erfragt werden:

```
      VMID NAME                 STATUS     MEM(MB)
BOOTDISK(GB) PID
       101 alpine               running    512
32.00 2486
```

Vorliegend gibt es nur den Gast 'alpine', bei weiteren virtuellen Gästen würden diese tabellarisch gelistet. Zum Abschluss sei hier angefügt, dass sich die virtuellen Gäste unter **/var/lib/vz/images** befinden und die Eckdaten der Gäste unter **/etc/qemu-server** in einer Konfigurationsdatei gespeichert sind. Mit **cat /etc/qemu-server/101.conf** kann (analog zu unserem Beispiel) der AlpineLinux-Gast betrachtet werden:

ide0: vm-101-disk.raw,cache=writeback
smp: 1
memory: 512
format: raw
ostype: other
sockets: 1
ide2: alpine-3.4.4-x86_64.iso,media=cdrom,cache=writeback
vlan0: rtl8139=4A:FF:30:BE:33:5D
bootdisk: ide0
name: alpine
cache: writeback

Im Vergleich zum nativen Format von QEMU/KVM ist das Format von ArchivistaMini sehr einfach, die Eckdaten des Gastes dürften plus/minus selbstverklärend sein.

Kenndaten permanent festlegen

ArchivistaMini läuft zu 100 Prozent im Hauptspeicher (RAM). Wird der Rechner

ausgeschaltet, sind die IP-Kenndaten 'verloren', d.h. sie sind beim erneuten Starten von ArchivistaMini erneut einzugeben. Um diese permanent zu hinterlegen, kann die Datei 'isolinux.cfg' (beim Start ab ISO/CD) und/oder 'syslinux.cfg' (USB-Stick/Festplatte) innerhalb der ISO-Datei mit den zuvor eingegebnen Parametern ergänzt werden, um den Vorgang dauerhaft verfügbar zu halten. Ein Eintrag analog zum ersten Hochfahren sähe z.B. so aus (Auszug aus isolinux.cfg):

```
label ram
kernel vmlinuz
APPEND initrd=initrd.img insmod progress loglevel=0 edd=off
pci=nocrs auto ip.192.168.0.254 ramdisk_size=81920
```

Die Informationen in Rot sind in 'isolinux.cfg' bzw. 'syslinux.cfg' hinzuzufügen. Es darf eingewendet werden, dass das Anpassen der ISO-Datei für Laien nicht einfach ist, doch steht für diesen Fall ja ausdrücklich die oben beschreibene kostenfrei Variante über **shop.archivista.ch/oscommunity/catalog/advanced_search_result.php?keywords=mini** zur Verfügung.

Hinweis: *Beim Hochfahren kann ArchivistaMini fast beliebig konfiguriert werden. Mehr Informationen finden sich unter* ***/de/media/archivistavm-linuxday2012.pdf.*** *Dort wird ArchivistaMini (inkl. der Aufbau eines DRBD-Clusters) ausführlich beschrieben. Ferner emphohlen werden kann* ***https://archivista.ch/cms/de/aktuell-blog/blogs-2011/archivistaram-amp-archivistausb.*** *Auch wenn der Blog aus dem Jahre 2011 stammt, so beschreibt er das Konzept von ArchivistaMini (wenn auch unter dem Namen ArchivistaRAM bzw. ArchivistaUSB) sehr gut.*

Fazit und Ausblick zu ArchivistaMini

Zum Abschluss sei erwähnt, dass es zu ArchivistaMini bzw. dem Verwalten der Gäste, der Datensicherung und vielem mehr eine ausführliche Dokumentation unter **https://archivista.ch/cgi-bin/ridhb/search.pl?query=archivistavm** gibt. Mit diesen Bordmitteln dürfte dem Einsatz von ArchivsitaMini nichts mehr im Wege stehen.

ArchivistaMini existiert in dieser Form seit mehreren Jahren. Im Unterschied zu anderen Lösungen, hat ArchivistaMini nicht den Anspruch, eine DataCenter-Lösung zu sein, dafür ist ArchivistaMini deutlich einfach(er) und schlank(er). Dafür sind die von ArchivistaMini verwendeten Komponenten wie Debian Jessie, Apache und QEMU/KVM in einem Guss auf kleinste Grösse zusammengepackt, der **RAM-Modus von ArchivistaMini bietet sowohl bei der Installation als auch beim täglichen Arbeiten viel Komfort. So kann z.B. eine neue Version jederzeit durch den Austausch der ISO-Datei realisiert werden.**

Bei der Weiterentwicklung von ArchivistaMini wird es nicht darum gehen, möglichst viele neue Features in möglichst kurzer Zeit zu realisieren, vielmehr geht es darum, die bestehende Lösung noch schlanker und einfacher zu machen, ohne die Kompatiblität zu bisherigen Versionen zu brechen. Diesen Job erledigt ArchivistaMini seit dem Jahre 2009. In dieser Zeit 'schrumpfte' die Lösung von anfänglich fast 700 MByte auf unter 90 MByte. Das mag in einer Zeit, wo alle mit GByte und TByte hantieren, fast schon kleinlich wirken, ist es aber nicht.

Wo andere Lösungen die ersten Gigabytes für die Systempartition benötigen, begüngt sich ArchivistaMini mit 300 MByte des Hauptspeichers, um das gesamte System in einer RAM-Disk vorzuhalten. Diskussionen um den besten Raid-Level für die Installation der Lösung wie bei **prolinux.de** gehören damit ein für alle Male der Vergangenheit an. Einfach USB-Stick oder CD einschieben und los geht es.