Engineer

Virtuelle Maschine

ein ganzer Rechner als Software auf einem physischen Host

Was es ist

Eine virtuelle Maschine bildet einen vollständigen Computer mit Software ab: Prozessoren, Arbeitsspeicher, Laufwerke und Netzwerkkarten stellt eine Virtualisierungsschicht auf einem physischen Host bereit. Darauf startet ein Betriebssystem, als stünde ihm eigene Hardware zur Verfügung. Die übliche Abkürzung ist VM. Manche Geräte können vom Host durchgereicht statt emuliert werden. Der NIST-Leitfaden zur vollständigen Virtualisierung, SP 800-125, definiert sie in einem Satz als „eine durch Virtualisierung geschaffene simulierte Umgebung“. Bei der dort behandelten Virtualisierung laufen ein oder mehrere Betriebssysteme samt ihren Programmen auf virtueller Hardware, jedes in einer eigenen VM. Das jeweilige Betriebssystem heißt Gastbetriebssystem.

Zwei Wörter tragen einen Großteil der Erklärung. Der reale Rechner ist der Host, jeder virtuelle Rechner ein Gast. Das Programm, das den Host unter seinen Gästen aufteilt, heißt Hypervisor, auch Virtual Machine Monitor. SP 800-125 beschreibt ihn als die Komponente, die die Gastbetriebssysteme auf einem Host verwaltet und den Befehlsfluss zwischen ihnen und der physischen Hardware steuert: Prozessor, Laufwerke, Arbeitsspeicher und Netzwerkkarten. Derselbe Leitfaden nennt die beiden Eigenschaften, die diese Anordnung nützlich machen. Der Hypervisor kann die Ressourcen des Hosts so aufteilen, dass jeder Gast auf seine zugewiesenen und auf ausdrücklich gemeinsam genutzte Ressourcen zugreift. Außerdem lässt sich ein Gast vollständig kapseln und dadurch von einem Host auf einen anderen übertragen.

Eine virtuelle Maschine unterscheidet sich von einem Container darin, wo die Grenze verläuft. Ein Container ist ein Prozess auf dem Kernel seines Hosts mit einer eingeschränkten Sicht auf das System. Eine virtuelle Maschine startet einen eigenen Kernel auf der Hardware, die der Hypervisor ihr bereitstellt. Deshalb

kann ein Linux-Gast auf einem Windows- oder macOS-Host laufen. Deshalb braucht aber auch jeder Gast ein eigenes Betriebssystem, das installiert, gepflegt und mit genügend Arbeitsspeicher versorgt werden muss.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist eine virtuelle Maschine?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Server werden nach dem tatsächlichen Bedarf dimensioniert und in die Cloud oder aus ihr heraus verlagert — die monatlichen Kosten stehen vorher fest.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. 20 Docker-Apps auf Google Cloud und AWS betrieben — 20 bereitgestellte und überwachte Containeranwendungen

20 Docker-Apps auf Google Cloud und AWS betrieben Kontakt aufnehmen →

Warum es wichtig ist

Ein bei einem Cloudanbieter gemieteter Server ist gewöhnlich eine virtuelle Maschine; Infrastructure as a Service baut darauf auf. Der Anbieter betreibt die Hypervisoren und den physischen Rechnerbestand. Kunden erhalten Maschinen, die sich über eine API anlegen, vergrößern oder verkleinern und entfernen lassen, statt bestellt, geliefert und in ein Rack eingebaut zu werden. Die Hardware bleibt die Aufgabe des Anbieters, das Betriebssystem wird die des Kunden.

Isolation ist der zweite Grund. Laut KVM-Projektseite hat jede virtuelle Maschine eigene virtualisierte Hardware, darunter Netzwerkkarte, Laufwerk und Grafikadapter. Gäste teilen keinen Kernel. Ein Kernelabsturz oder ein fehlerhaftes Update in einem Gast bleibt deshalb auf ihn beschränkt. Was die physische Hardware erreicht, entscheidet der Hypervisor, nicht der Gast. Zwei Kunden oder zwei Umgebungen eines Kunden können sich einen physischen Rechner teilen, ohne ein Betriebssystem gemeinsam zu nutzen.

Drittens kann eine Maschine bei imagebasiertem Speicher zu einem Satz Dateien werden. Das Laufwerk eines Gasts kann als Imagedatei auf dem Host liegen. Sie lässt sich kopieren, als Vorlage aufbewahren oder zum Start mehrerer gleicher Maschinen verwenden. QEMUs Format qcow2 kann nur die Abweichungen von einem Basisimage aufzeichnen und Snapshots des Laufwerks in derselben Datei speichern. Eine Vorlage mit einer dünnen zusätzlichen Schicht je Gast ist daher eine naheliegende Anordnung.

Viertens lässt Emulation Software für eine Prozessorfamilie auf einer anderen laufen. Bei QEMUs Systememulation arbeitet das Gastbetriebssystem auf einem in Software nachgebildeten Rechner mit eigenem Prozessor, Speicher und Geräten. Mit einem Beschleuniger wie KVM unter Linux, Hypervisor Framework unter macOS oder Windows Hypervisor Platform unter Windows übernimmt die Virtualisierungsunterstützung des Hostprozessors die Hauptarbeit. Ohne einen solchen Beschleuniger übersetzt QEMUs Tiny Code Generator die Gastbefehle in Software. So startet auch das Beispiel der QEMU-Einführung eine Arm64-Maschine mit `-accel tcg` .

Der Preis ist, dass jeder Gast ein vollständiger Computer bleibt, der betreut werden muss. Jeder bringt ein zu pflegendes Betriebssystem, reservierten Arbeitsspeicher und einen eigenen Startvorgang mit. Container vermeiden das, indem sie den Kernel des Hosts teilen. Dadurch sind sie leichter, verzichten aber auf einen eigenen Kernel als Grenze zwischen den Arbeitslasten. Die Wahl zwischen beiden legt fest, wo diese Grenze verlaufen soll.

Wie es in der Praxis aussieht

Unter x86-Linux bilden KVM und QEMU das übliche Paar. KVM ist die Hälfte im Kernel: ein ladbares Modul, `kvm.ko` , mit dem prozessorspezifischen Gegenstück `kvm-intel.ko` oder `kvm-amd.ko` , seit Version 2.6.20 im Linux-Hauptzweig enthalten. Es benötigt die Virtualisierungserweiterungen des Prozessors, Intel VT oder AMD-V. Die KVM-FAQ prüft deren Flags `vmx` und `svm` in `/proc/cpuinfo` . Zu ihren Schritten bei Problemen gehört außerdem, festzustellen, ob `/dev/kvm` existiert und der Benutzer es öffnen darf:

```
grep -E -c '^flags.*(vmx|svm)' /proc/cpuinfo
ls -l /dev/kvm
```

Ein Wert größer als null bedeutet, dass der Prozessor die Erweiterungen meldet. QEMU ist die Hälfte im Userspace, die den Rechner um KVM herum aufbaut. Das Laufwerk eines neuen Gasts kann eine dünne Schicht auf einem gemeinsam verwendeten Basisimage sein:

```
qemu-img create -f qcow2 -b debian-base.qcow2 -F qcow2 web1.qcow2
qemu-img info --backing-chain web1.qcow2
```

Laut qemu-img-Handbuch zeichnet ein mit einer Basisdatei angelegtes Image nur die Abweichungen davon auf. Die Basisdatei bleibt unverändert, solange diese Abweichungen nicht ausdrücklich in sie zurückgeschrieben werden. Der zweite Befehl zeigt jede Datei der Kette. Diese Liste sollte gelesen werden, bevor etwas verschoben oder gelöscht wird. Der Start des Gasts sieht dann so aus:

```
qemu-system-x86_64 -accel kvm -m 2G -smp 2 \
    -drive file=web1.qcow2,if=virtio,format=qcow2 \
    -nographic
```

`-accel kvm` übergibt die Prozessorarbeit an KVM, standardmäßig über `/dev/kvm`. `-m` legt den Arbeitsspeicher des Gasts fest, `-smp` seine Prozessoranzahl. Bei `-drive` bindet `if=virtio` das Laufwerk über die virtio-Schnittstelle an, eine der im Handbuch aufgeführten Schnittstellen. `format=qcow2` gibt das Format ausdrücklich an, statt es von QEMU erkennen zu lassen. `-nographic` schaltet das Grafikfenster aus. QEMU läuft dann als reines Kommandozeilenprogramm; die serielle Konsole des Gasts und der QEMU-Monitor teilen sich das Terminal. Ctrl-a h zeigt, wie zwischen beiden gewechselt wird.

Bei heruntergefahrenem Gast erstellt und lädt dasselbe Werkzeug Snapshots des Laufwerks. `-c` erstellt einen, `-l` listet sie auf und `-a` setzt das Laufwerk auf einen gespeicherten Stand zurück:

```
qemu-img snapshot -c before-upgrade web1.qcow2
qemu-img snapshot -l web1.qcow2
qemu-img snapshot -a before-upgrade web1.qcow2
```

Die Dimensionierung ist eine Entwurfsentscheidung, kein auszufüllendes Formularfeld. Der Portfolioeintrag über den Betrieb der öffentlichen Unternehmensdienste auf einem kleinen Server dokumentiert eine Konfiguration vom September 2026 mit einem Prozessorkern, 10 GB Plattenplatz und 512 MB Arbeitsspeicher, von denen Programme ungefähr 464 MB verwenden können. Dieser Host bedient sieben Domains über TLS und betreibt außerdem die Git-Forge; die Backupautomatisierung wurde gesondert implementiert. Der Arbeitsspeicher ist die Grenze, an der sich dort jede Entscheidung ausrichtet. Ob zusätzlich ein Onion-Service über Tor Platz hätte, entschied ein dokumentierter Tag mit Messwerten. Der niedrigste Wert des

verfügbaren Speichers lag dabei bei etwa 310 MB. Die Einstellung `-m` jedes Gasts stellt dieselbe Frage. Eine Messung beantwortet sie besser als ein größer dimensionierter Tarif.

Der Eintrag über die Verwaltung von dreißig Ubuntu-Linux-VPS-Instanzen zeigt das andere Ende der Skala. Die Einrichtung dieser Instanzen war über Jahre gewachsen, statt geplant zu sein; ihre Konfigurationen hatten sich auseinanderentwickelt. Danach wurden sie als eine gemeinsame Flotte betrieben: mit derselben gehärteten Netzwerkkonfiguration auf jeder Instanz und mit Wiederherstellungen, die tatsächlich erprobt wurden.

Wo es schiefgeht

Die meisten Probleme mit virtuellen Maschinen entstehen, wenn vergessen wird, dass jeder Gast ein echter Computer ist, der betreut werden muss, und ein imagebasiertes Laufwerk zugleich eine gewöhnliche Datei auf einem anderen Rechner.

Beim Start des Gasts fehlt KVM. In der obigen x86-Konfiguration verhindern fehlende oder in der Firmware abgeschaltete Prozessorerweiterungen die KVM-Beschleunigung. Auch `/dev/kvm` kann fehlen. Ein mit `-accel kvm` gestarteter Gast bricht dann sofort ab. Die KVM-FAQ warnt, dass manche Hersteller VT im BIOS so abschalten, dass es sich nicht wieder einschalten lässt. Abhilfe schafft die obige Flag-Prüfung, die Suche nach einer Firmwareoption für Virtualisierung und die Kontrolle, ob `/dev/kvm` existiert und einer Gruppe gehört, der der Benutzer angehört. Die FAQ schlägt dafür eine Gruppe namens `kvm` vor.

Ein Gast ohne festgelegte Speichergröße erhält 128 MiB. Das ist QEMUs Standardwert für `-m`, unabhängig davon, wie viel der Host frei hat. Ein Gast kann deshalb starten und später auf eine Weise scheitern, die nach einem Fehler seiner Software aussieht. `-m` und `-smp` sollten bei jedem Start angegeben und anhand von Messungen gewählt werden, so wie beim kleinen Server ein Tag mit Messwerten die Frage nach dem Onion-Service beantwortete.

Ein während des Gastbetriebs bearbeitetes Image kann zerstört werden. Das qemu-img-Handbuch ist eindeutig: Wird ein Image bearbeitet, während ein Gast oder ein anderer Prozess es verwendet, kann das Image zerstört werden. Ein Snapshot des laufenden Gastlaufwerks mit `qemu-img snapshot -c` ist

genau ein solcher Eingriff. Bevor qemu-img auf das Laufwerk zugreift, muss der Gast heruntergefahren werden. Dieser Schritt gehört in jede betreffende Betriebsanleitung.

Snapshots ersetzen Backups. Mit qemu-img erzeugte Snapshots liegen in derselben Imagedatei; `qemu-img info` führt sie zusammen mit den Laufwerksinformationen auf. Geht die Datei verloren oder wird sie beschädigt, sind auch alle Snapshots betroffen. Backups müssen auf Speicher eines anderen Rechners kopiert und ihre Wiederherstellung muss getestet werden. Eine nie durchgeführte Wiederherstellung beweist nichts.

Ein Basisimage ändert seinen Inhalt oder seinen Ort unter den Overlays. Ein dünnes Image enthält nur Abweichungen. Ohne genau die Basis, auf die es sich bezieht, ist es unbrauchbar. Das Handbuch warnt, dass ein Wechsel der Basisdatei im unsicheren Modus den für den Gast sichtbaren Inhalt beschädigt, wenn die neue Datei die falsche ist. Ein relativer Basispfad wird zudem vom Verzeichnis des Overlays aus aufgelöst. Wird nur eine der Dateien verschoben, zerbricht die Kette. Basisimages sollten deshalb schreibgeschützt behandelt werden; jede neue Version erhält einen neuen Namen. Vor einem Verschieben gehört `qemu-img info --backing-chain` zur Prüfung. `qemu-img commit` ist nur bewusst auszuführen, denn es schreibt in die Basis und leert das Overlay.

Das Laufwerksformat wird aus der Imagedatei erraten. Ohne `format=` erkennt QEMU das Format am Dateiinhalt. Das Handbuch empfiehlt `format=raw`, um einen nicht vertrauenswürdigen Formatheader nicht zu interpretieren. Ein von anderswo heruntergeladenes oder von einem Gast beschriebenes Image ist genau in diesem Sinn nicht vertrauenswürdig. Auf jeder `-drive` -Zeile sollte das Format stehen.

Jeder Gast wird ein individuell gepflegtes Einzelstück. Von Hand konfigurierte Maschinen entwickeln sich Änderung für Änderung auseinander, bis keine zwei mehr gleich sind. An diesem Punkt stand die genannte VPS-Flotte vor ihrer zentralen Verwaltung. Jeder Gast sollte aus demselben Image und derselben Konfiguration entstehen. Die Konfiguration gehört in die Versionsverwaltung, die Maschinen gehören als eine Flotte mit einer gemeinsamen Basis verwaltet.

Weiterführende Lektüre

- Guide to Security for Full Virtualization Technologies (NIST SP 800-125) — die VM-Definition, die Aufgabe des Hypervisors und die Trennung der Gäste.

- KVM — die Projektseite zu Modulen, Prozessorerweiterungen und Kernelversion.
- KVM FAQ — die Prüfung auf vmx und svm, `/dev/kvm` -Berechtigungen und die BIOS-Einstellung.
- QEMU System Emulation introduction — Beschleuniger, Tiny Code Generator und ein vollständiger Beispielbefehl.
- QEMU invocation — alle oben verwendeten Optionen: `-accel` , `-m` , `-smp` , `-drive` , `-nographic` .
- QEMU disk image utility — qcow2, Basisdateien, Snapshots und die Warnung vor Images in Benutzung.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Bereitstellungsweg, die Betriebsrisiken und die Bereitschaftsverantwortlichen. Wir können eine Änderung mit Prüfungen, einem Wiederherstellungsweg und klaren Zuständigkeiten abgrenzen.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist eine virtuelle Maschine?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/virtual-machine/