Engineer

virtuell maskin

en hel dator som körs i mjukvara på en fysisk värd

Vad det är

En virtuell maskin presenterar en hel dator genom programvara: processorer, minne, diskar och nätverkskort som tillhandahålls av ett virtualiseringslager på en fysisk värd, medan ett operativsystem startar på dem som om de vore dedikerad maskinvara. Den förkortas ofta VM, efter engelskans virtual machine. Vissa enheter kan göras direkt tillgängliga från värden i stället för att emuleras. NIST:s vägledning om full virtualisering, SP 800-125, definierar den på en rad som en simulerad miljö skapad genom virtualisering. I den form av virtualisering som vägledningen handlar om körs ett eller flera operativsystem med sina program på virtuell maskinvara, vart och ett i en egen VM, där det kallas ett gästoperativsystem.

Två ord gör det mesta av arbetet runt den. Den verkliga maskinen är värden, och varje virtuell maskin på den är en gäst. Programmet som delar upp värden mellan gästerna är hypervisorn, även kallad virtual machine monitor, som SP 800-125 beskriver som den komponent som hanterar gästoperativsystemen på en värd och styr flödet av instruktioner mellan dem och den fysiska maskinvaran: processor, diskar, minne och nätverkskort. Samma vägledning pekar ut två saker som gör upplägget användbart. Hypervisorn kan dela upp värdens resurser så att varje gäst får tillgång till sina tilldelade resurser och de resurser som uttryckligen delas, och en gäst kan kapslas in helt, så att den kan flyttas från en värd till en annan.

En virtuell maskin skiljer sig från en container i var gränsen dras. En container är en process på värdens kärna med en begränsad bild av systemet. En virtuell maskin startar en egen kärna på maskinvara som hypervisorn presenterar för den. Därför kan en Linux-gäst köras på en Windows- eller macOS-värd, och därför behöver varje gäst ett eget operativsystem som ska installeras, uppdateras och få tillräckligt med minne för att köras.

Använd begreppet i ett beslut

Använd ”Vad är en virtuell maskin?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Servrar dimensionerade efter vad du faktiskt använder, flyttade till molnet eller tillbaka ut ur det, med månadskostnaden uträknad först.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Drev 20 Docker-appar på Google Cloud och AWS — 20 containerbaserade applikationer, driftsatta och övervakade

Drev 20 Docker-appar på Google Cloud och AWS Hör av er →

Varför det spelar roll

En server som hyrs av en molnleverantör är oftast en virtuell maskin, och infrastruktur som tjänst bygger på dem. Leverantören driver hypervisorerna och den fysiska maskinparken; kunden får maskiner som kan skapas, ändras i storlek och tas bort via ett API i stället för att beställas, levereras och monteras i ett rack. Maskinvaran förblir någon annans bekymmer, medan operativsystemet blir kundens.

Isolering är det andra skälet. KVM-projektets sida säger att varje virtuell maskin har privat virtualiserad maskinvara, med eget nätverkskort, egen disk och eget grafikkort. Gästerna delar ingen kärna, så en kärnkrasch eller en misslyckad uppdatering i en gäst stannar där, och det är hypervisorn, inte gästerna, som avgör vad som når den fysiska maskinvaran. Två kunder, eller två miljöer hos samma kund, kan dela en fysisk maskin utan att dela operativsystem.

För det tredje kan en maskin med lagring i avbildningar bli en uppsättning filer. En gästs disk kan vara en avbildningsfil på värden, som kan kopieras, sparas som mall eller användas för att starta flera identiska maskiner. QEMU:s format qcow2 kan registrera enbart skillnaderna mot en basavbildning och kan spara ögonblicksbilder av disken i samma fil, så en mall plus ett tunt skikt per gäst är en naturlig uppläggning.

För det fjärde låter emulering program som är byggda för en processorfamilj köras på en annan. QEMU:s systememulering ger en virtuell modell av en maskin, med processor, minne och enheter, som ett gästoperativsystem kan köras på. Med en accelerator, som KVM på Linux, Hypervisor Framework på macOS eller Windows Hypervisor Platform på Windows, gör virtualiseringsstödet i värdens processor det tunga arbetet. Utan accelerator översätter QEMU:s Tiny Code

Generator gästens instruktioner i programvara, och det är så introduktionen till QEMU:s dokumentation startar en Arm64-maskin i sitt eget exempel, med `-accel tcg` .

Priset är att varje gäst är en fullständig dator som ska skötas. Var och en har sitt eget operativsystem att uppdatera, sitt eget reserverade minne och sin egen uppstart. Containrar undviker det genom att dela värdens kärna, vilket gör var och en lättare och ger upp en egen kärna som väggen mellan arbetslasterna. Att välja mellan dem är att välja var den väggen ska stå.

Så ser det ut i praktiken

På x86-Linux är det vanliga paret KVM och QEMU. KVM är kärndelen: en laddbar modul, `kvm.ko` , med en processorspecifik följeslagare, `kvm-intel.ko` eller `kvm-amd.ko` , och den har ingått i huvudlinjen av Linux sedan 2.6.20. Den behöver processorns virtualiseringstillägg, Intel VT eller AMD-V. KVM:s FAQ letar efter deras flaggor, `vmx` och `svm` , i `/proc/cpuinfo` , och felsökningslistan där omfattar att bekräfta att `/dev/kvm` finns och att användaren har rättigheter att öppna den:

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

Ett tal över noll betyder att processorn rapporterar tilläggen. QEMU är delen i användarutrymmet, som bygger maskinen runt KVM. Disken för en ny gäst kan vara ett tunt skikt ovanpå en delad basavbildning:

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

Manualen för qemu-img säger att en avbildning som skapats ovanpå en basavbildning bara registrerar skillnaderna mot den, och att basavbildningen aldrig ändras om inte skillnaderna förs tillbaka in i den med `qemu-img commit` . Det andra kommandot listar varje fil i kedjan, och utdatan bör läsas innan något flyttas eller tas bort. Själva starten av gästen ser ut så här:

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

`-accel kvm` lämnar processorarbetet till KVM, som standard via `/dev/kvm`. `-m` anger gästens minne och `-smp` antalet processorer. På `-drive` ansluter `if=virtio` disken via virtio-gränssnittet, ett av de gränssnitt som manualen listar, och `format=qcow2` anger formatet i stället för att låta QEMU avgöra det själv. `-nographic` stänger av det grafiska fönstret, så att QEMU blir ett vanligt kommandoradsprogram med gästens seriella konsol i terminalen, delad med QEMU-monitorn; Ctrl-a h visar hur man växlar mellan de två.

Med gästen avstängd tar samma verktyg ögonblicksbilder av disken och återställer dem. `-c` skapar en, `-l` listar dem och `-a` återför disken till ett sparat tillstånd:

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

Dimensionering är ett designbeslut, inte ett fält i ett formulär. Portföljposten om att köra företagets publika tjänster på en liten server dokumenterar en uppsättning från september 2026 med en processorkärna, en disk på 10 GB och 512 MB minne, varav programmen kan använda omkring 464 MB. Värddatorn levererar sju domäner över TLS och kör dessutom git-forgen, medan automatisering av säkerhetskopiering har implementerats separat; minnet är gränsen som varje designval där rättar sig efter. Om den också klarade en onion-tjänst för Tor avgjordes utifrån ett dokumenterat dygns stickprov, där den lägsta avläsningen av tillgängligt minne låg på omkring 310 MB. Inställningen `-m` för varje gäst ställer samma fråga, och en mätning besvarar den bättre än ett större abonnemang.

Posten om förvaltning av trettio VPS-instanser med Ubuntu Linux visar andra änden av skalan. Uppsättningen av de instanserna hade byggts på bit för bit under åren utan någon plan, och konfigurationerna hade glidit isär. Därefter förvaltades de som en samlad maskinpark: en härdad nätverkskonfiguration på varje instans och återställningar som provades i praktiken i stället för att tas för givna.

Var det går fel

De flesta problem med virtuella maskiner kommer av att man glömmer att var och en är en verklig dator som ska skötas, och att en disk som bygger på en avbildning också är en vanlig fil på en annan maskin.

KVM saknas när gästen startar. I x86-uppsättningen ovan hindrar saknade eller firmwareavstängda processortillägg KVM-acceleration, och `/dev/kvm` kan saknas, så en gäst som startas med `-accel kvm` stannar direkt. KVM:s FAQ varnar för att vissa tillverkare stänger av VT i BIOS på ett sätt som inte går att slå på igen. Lösningen är att köra flaggkontrollen ovan, leta efter ett alternativ i firmware som slår på virtualisering, och bekräfta att `/dev/kvm` finns och tillhör en grupp som användaren ingår i; FAQ:n föreslår en grupp som heter `kvm`.

En gäst som startas utan minnesstorlek får 128 MiB. Det är QEMU:s standardvärde för `-m`, och det gäller oavsett hur mycket värden har ledigt, så en gäst kan starta och sedan fallera på sätt som liknar buggar i programvaran i den. Lösningen är att ange `-m` och `-smp` varje gång och dimensionera dem utifrån mätningar, så som posten om den lilla servern avgjorde frågan om onion-tjänsten med ett dygns mätningar.

En avbildning som ändras medan gästen körs kan förstöras. Manualen för qemu-img är rak: använd aldrig verktyget för att ändra en avbildning som används av en körande virtuell maskin eller någon annan process, eftersom det kan förstöra avbildningen. Att ta en ögonblicksbild av disken för en körande gäst med `qemu-img snapshot -c` är precis det. Lösningen är att stänga av gästen innan qemu-img rör dess disk, och att göra det till ett eget steg i varje driftrutin där det förekommer.

Ögonblicksbilder får ersätta säkerhetskopior. Ögonblicksbilder tagna med qemu-img ligger i samma avbildningsfil, vilket `qemu-img info` visar genom att lista dem tillsammans med disken. När filen går förlorad eller skadas försvinner alla ögonblicksbilder med den. Lösningen är att kopiera säkerhetskopior till lagring på en annan maskin och att prova återställningen, eftersom en återställning som aldrig har körts inte bevisar någonting.

En basavbildning ändras eller flyttas under de avbildningar som bygger på den. En tunn avbildning innehåller bara skillnader, så den betyder ingenting utan exakt den basavbildning den skapades mot. Manualen varnar för att om basavbildningen byts i osäkert läge (unsafe) och den nya filen är fel fil, blir

innehållet som gästen ser korrupt, och en relativ sökväg till basavbildningen tolkas från katalogen där den härledda avbildningen ligger, så att flytta den ena filen utan den andra bryter kedjan. Lösningen är att behandla basavbildningar som skrivskyddade och ge varje ny version ett nytt namn, att läsa `qemu-img info --backing-chain` innan något flyttas, och att köra `qemu-img commit` bara medvetet, eftersom kommandot skriver in i basavbildningen och tömmer den härledda.

Diskformatet gissas utifrån själva avbildningen. Utan `format=` avgör QEMU formatet utifrån filens innehåll, och manualen rekommenderar `format=raw` för att undvika att tolka ett formathuvud som inte går att lita på. En avbildning som hämtats från annat håll, eller som en gäst har skrivit till, är just en sådan. Lösningen är att ange formatet på varje `-drive` -rad.

Varje gäst glider över till att bli ett husdjur. Maskiner som konfigureras för hand glider isär en ändring i taget tills inga två är lika, och där befann sig VPS-parken ovan innan den kom under förvaltning. Lösningen är att bygga varje gäst från samma avbildning och samma konfiguration, hålla konfigurationen i versionshantering och förvalta maskinerna som en park med en baskonfiguration.

Läs mer

- Guide to Security for Full Virtualization Technologies (NIST SP 800-125) — definitionen av en virtuell maskin, hypervisorns uppgift och hur gäster hålls isär.
- KVM — projektsidan: moduler, processortillägg och första kärnversion.
- KVM FAQ — kontrollen av vmx och svm, rättigheter till `/dev/kvm` och BIOS-inställningen.
- QEMU System Emulation introduction — acceleratorerna, Tiny Code Generator och ett komplett exempelkommando.
- QEMU invocation — alla flaggor som används ovan: `-accel` , `-m` , `-smp` , `-drive` , `-nographic` .
- QEMU disk image utility — qcow2, basavbildningar, ögonblicksbilder och varningen om avbildningar i bruk.

Ordlista

Välj utifrån era faktiska förutsättningar

Ta med driftsättningsvägen, driftsriskerna och dem som har jour. Vi kan avgränsa en förändring med kontroller, en väg tillbaka och tydligt ansvar.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är en virtuell maskin?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

welcome@engineer.company +45 50 10 61 84 Köpenhamn, Danmark

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/virtual-machine/