Engineer

віртуальна машина

📺 цілий комп'ютер, що працює програмно на фізичному хості

Що це таке

Віртуальна машина — це програмне подання цілого комп’ютера: процесори, пам’ять, диски й мережеві карти надає шар віртуалізації на фізичному хості, а операційна система завантажується на них так, ніби це виділене обладнання. Її часто скорочують до VM, за англійською назвою virtual machine. Деякі пристрої можна передати безпосередньо з хоста, а не емулювати. Посібник NIST з повної віртуалізації, SP 800-125, визначає її одним рядком: симульоване середовище, яке створила віртуалізація. У тій формі віртуалізації, якій присвячено посібник, одна чи кілька операційних систем разом зі своїми програмами працюють на віртуальному обладнанні, кожна в окремій VM, і таку систему називають гостьовою.

Більшу частину роботи тут виконують два слова. Справжня машина — це хост, а кожна віртуальна машина на ньому — гість. Програма, яка ділить хост між гостями, — гіпервізор, його ще називають монітором віртуальних машин (VMM); SP 800-125 описує його як компонент, що керує гостьовими операційними системами на хості й контролює потік інструкцій між ними та фізичним обладнанням: процесором, дисками, пам’яттю й мережевими картами. Той самий посібник називає дві речі, завдяки яким такий устрій вартий зусиль. Гіпервізор може розділити ресурси хоста так, щоб кожен гість мав доступ до виділених йому ресурсів і будь-яких ресурсів, які явно надано у спільне користування, а гостя можна повністю запакувати, і тоді його легко перенести з одного хоста на інший.

Віртуальна машина відрізняється від контейнера тим, де проведено межу. Контейнер — це процес на ядрі хоста з обмеженим поглядом на систему. Віртуальна машина завантажує власне ядро на обладнанні, яке їй надає гіпервізор. Саме тому гість із Linux може працювати на хості з Windows чи macOS, і саме тому кожному гостю потрібна власна операційна система, яку треба встановити, оновлювати й забезпечити достатнім обсягом пам’яті.

Застосуйте поняття до рішення

Застосуйте «Що таке віртуальна машина?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Сервери під те навантаження, яке у вас справді є, перенесені в хмару чи назад, із порахованим наперед місячним рахунком.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Вела 20 застосунків Docker у Google Cloud і AWS — 20 контейнеризованих застосунків розгорнуто й поставлено під моніторинг

Вела 20 застосунків Docker у Google Cloud і AWS Зв'язатися з нами →

Чому це важливо

Сервер, орендований у хмарного провайдера, зазвичай є віртуальною машиною, і саме на них побудовано інфраструктуру як послугу. Провайдер керує гіпервізорами та фізичним парком; клієнт отримує машини, які можна створювати, збільшувати чи зменшувати та знищувати через API, а не замовляти, чекати на доставку й монтувати в стійку. Обладнання лишається чужою турботою, а операційна система стає турботою клієнта.

Друга причина — ізоляція. Сторінка проєкту KVM каже, що кожна віртуальна машина має власне приватне віртуалізоване обладнання: мережеву карту, диск, графічний адаптер тощо. Гості не мають спільного ядра, тож збій ядра чи невдале оновлення в одному гостеві в ньому й лишається, а вирішує, що дійде до фізичного обладнання, гіпервізор, а не гості. Два клієнти або два середовища одного клієнта можуть ділити фізичну машину, не ділячи операційної системи.

По-третє, зі сховищем на основі образів машина може перетворитися на набір файлів. Диск гостя може бути файлом образу на хості, який можна скопіювати, зберегти як шаблон або використати для запуску кількох однакових машин. Формат qcow2 у QEMU вміє записувати лише відмінності від базового образу й зберігати знімки диска в тому самому файлі, тож один шаблон плюс тонкий шар на кожного гостя — природне компонування.

По-четверте, емуляція дає змогу запускати програми, зібрані для процесорів однієї родини, на процесорах іншої. Системна емуляція QEMU надає віртуальну модель машини — процесор, пам’ять і пристрої, — на якій працює гостьова операційна система. З прискорювачем, як-от KVM у Linux, Hypervisor Framework у macOS чи Windows Hypervisor Platform у Windows, важку роботу бере на себе апаратна підтримка віртуалізації в процесорі хоста. Без

прискорювача Tiny Code Generator у QEMU перекладає інструкції гостя програмно; саме так приклад зі вступу до документації QEMU запускає машину Arm64 з `-accel tcg` .

Ціна в тому, що кожен гість — повноцінний комп'ютер, який треба доглядати. Кожен має власну операційну систему, яку треба оновлювати, власну зарезервовану пам'ять і власне завантаження. Контейнери уникають цього, бо ділять ядро хоста: кожен стає легшим, але окреме ядро як стіна між робочими навантаженнями зникає. Вибір між ними — це вибір того, де має стояти ця стіна.

Як це виглядає на практиці

У Linux на x86 звична пара — KVM і QEMU. KVM — це частина в ядрі: завантажуваний модуль `kvm.ko` з процесорно-специфічним напарником `kvm-intel.ko` або `kvm-amd.ko` ; в офіційному ядрі Linux він є з версії 2.6.20. Йому потрібні розширення віртуалізації в процесорі, Intel VT або AMD-V. FAQ KVM шукає їхні прапорці, `vmx` і `svm` , у `/proc/cpuinfo` , а серед кроків діагностики там є перевірка, що `/dev/kvm` існує і що користувач має права його відкрити:

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

Число, більше за нуль, означає, що процесор повідомляє про розширення. QEMU — частина в просторі користувача, яка будує машину навколо KVM. Диск нового гостя може бути тонким шаром поверх спільного базового образу:

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

Посібник qemu-img пояснює, що образ, створений поверх базового, записує лише відмінності від нього, а базовий образ ніколи не змінюється, якщо ці відмінності не перенести в нього командою `qemu-img commit` . Друга команда показує кожен файл у ланцюжку, і її вивід варто прочитати, перш ніж щось переносити чи видаляти. Сам запуск гостя виглядає так:

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

`-accel kvm` передає роботу процесора KVM, за замовчуванням через `/dev/kvm`. `-m` задає пам’ять гостя, а `-smp` — кількість процесорів. У `-drive` параметр `if=virtio` під’єднує диск через інтерфейс virtio, один із типів інтерфейсу, які перелічує посібник, а `format=qcow2` явно називає формат, щоб QEMU не визначав його сам. `-nographic` вимикає графічне вікно, і QEMU стає звичайною програмою командного рядка: послідовна консоль гостя з’являється в терміналі разом із монітором QEMU, а Ctrl-a h показує, як перемикатися між ними.

Коли гість вимкнений, той самий інструмент створює і відновлює знімки диска. `-c` створює знімок, `-l` показує список, а `-a` повертає диск до збереженого стану:

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

Вибір розміру — це проєктне рішення, а не поле у формі. У записі в портфоліо про роботу публічних сервісів компанії на одному невеликому сервері задокументовано конфігурацію вересня 2026 року з одним процесорним ядром, диском на 10 ГБ і 512 МБ пам’яті, з яких програмам доступно приблизно 464 МБ. Цей хост обслуговує сім доменів через TLS і також тримає git-фордж, а автоматизацію резервного копіювання реалізовано окремо; пам’ять є межею, якій там підпорядковано кожне проєктне рішення. Чи витримає він ще й onion-сервіс у Tor, вирішили на основі вибірок, зафіксованих протягом доби, у яких найнижче значення доступної пам’яті становило близько 310 МБ. Параметр `-m` кожної гостьової машини ставить те саме питання, і вимірювання відповідає на нього краще, ніж тарифний план із більшими ресурсами.

Запис про керування тридцятьма інстансами Ubuntu Linux VPS показує інший кінець шкали. Налаштування цих інстансів роками накопичувалися без жодного плану, і конфігурації поступово розійшлися. Після цього ними почали керувати як єдиним парком: одна захищена конфігурація мережі на кожному інстансі та відновлення, які перевіряли на ділі, а не брали на віру.

Де виникають проблеми

Більшість клопотів із віртуальними машинами починається тоді, коли забувають, що кожна з них — справжній комп'ютер, який треба доглядати, а диск на основі образу — це ще й звичайний файл на іншій машині.

KVM недоступний, коли гість стартує. У наведеній вище конфігурації x86 відсутні або вимкнені в прошивці розширення процесора унеможливлюють прискорення KVM, а `/dev/kvm` може бути відсутнім, і гість, запущений з `-accel kvm`, одразу зупиняється. FAQ KVM попереджає, що деякі виробники вимикають VT у BIOS так, що знову ввімкнути її вже неможливо. Вихід — виконати перевірку прапорців, наведену вище, пошукати в прошивці параметр, що вмикає віртуалізацію, і переконатися, що `/dev/kvm` існує й належить групі, до якої входить користувач; FAQ радить групу з назвою `kvm`.

Гість без заданого обсягу пам'яті отримує 128 МіБ. Це значення `-m` у QEMU за замовчуванням, і воно діє незалежно від того, скільки пам'яті вільно на хості, тож гість може запуститися, а потім збоїти так, що це схоже на вади програм усередині нього. Вихід — щоразу задавати `-m` і `-smp` і визначати їх за вимірюваннями, так само як запис про невеликий сервер вирішив питання onion-сервісу добою вимірювань.

Образ, який змінюють під час роботи гостя, може зруйнуватися. Посібник qemu-img висловлюється прямо: ніколи не змінювати ним образ, яким користується запущена віртуальна машина чи інший процес, бо це може знищити образ. Створення знімка диска запущеного гостя командою `qemu-img snapshot -c` — саме такий випадок. Вихід — вимикати гостя, перш ніж qemu-img торкнеться його диска, і внести це окремим кроком у кожну інструкцію, де таке трапляється.

Знімки підміняють резервні копії. Знімки, зроблені qemu-img, зберігаються в тому самому файлі образу, і `qemu-img info` показує їх поряд з даними про диск. Якщо файл втрачено чи пошкоджено, разом із ним зникають усі знімки. Вихід — копіювати резервні копії в сховище на іншій машині й перевіряти відновлення, бо відновлення, яке ніколи не запускали, нічого не доводить.

Базовий образ змінюють або переносять з-під похідних. Тонкий образ містить лише відмінності, тож без точного базового образу, на якому його створено, він нічого не означає. Посібник попереджає, що коли в небезпечному режимі (unsafe) підставити як базовий не той файл, вміст,

який бачить гість, буде пошкоджено, а відносний шлях до базового образу визначається від каталогу самого похідного образу, тож перенесення одного файлу без іншого розриває ланцюжок. Вихід — вважати базові образи доступними лише для читання й давати кожній новій версії нове ім'я, читати `qemu-img info --backing-chain` перед будь-яким перенесенням і запускати `qemu-img commit` лише свідомо, бо ця команда записує зміни в базовий образ і спорожнює похідний.

Формат диска вгадують за вмістом самого образу. Без `format=` QEMU визначає формат за вмістом файлу, і посібник радить `format=raw`, щоб не інтерпретувати заголовок формату, якому не можна довіряти. Образ, завантажений деінде, або той, у який писав гість, якраз такий. Вихід — називати формат у кожному рядку `-drive`.

Кожен гість поступово стає домашнім улюбленцем. Машини, налаштовані вручну, розходяться по одній зміні за раз, доки серед них не лишиться двох однакових; саме в такому стані був парк VPS, описаний вище, до того як його взяли під керування. Вихід — збирати кожного гостя з того самого образу й тієї самої конфігурації, тримати конфігурацію в системі контролю версій і керувати машинами як одним парком з одним базовим рівнем.

Що почитати далі

- Guide to Security for Full Virtualization Technologies (NIST SP 800-125) — визначення віртуальної машини, завдання гіпервізора й ізоляція гостей.
- KVM — сторінка проєкту: модулі, розширення процесора й перша версія ядра.
- KVM FAQ — перевірка vmx і svm, права на `/dev/kvm` і параметр BIOS.
- QEMU System Emulation introduction — прискорювачі, Tiny Code Generator і повний приклад команди.
- QEMU invocation — усі параметри, ужиті вище: `-accel`, `-m`, `-smp`, `-drive`, `-nographic`.
- QEMU disk image utility — qcow2, базові образи, знімки й застереження щодо образів у роботі.

Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке віртуальна машина?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

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

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/virtual-machine/