Engineer

Machine virtuelle

un ordinateur complet virtualisé sur un hôte physique

Ce que c’est

Une machine virtuelle présente un ordinateur entier au moyen de logiciels : processeurs, mémoire, disques et cartes réseau sont fournis par une couche de virtualisation sur un hôte physique. Un système d’exploitation y démarre comme s’il s’agissait de matériel dédié. Son nom est souvent abrégé en VM, pour virtual machine. Certains périphériques peuvent être transmis directement depuis l’hôte plutôt qu’émulés. Le guide du NIST sur la virtualisation complète, SP 800-125, la définit en une ligne comme « un environnement simulé créé par la virtualisation ». Dans le type de virtualisation couvert par ce guide, un ou plusieurs systèmes d’exploitation s’exécutent avec leurs programmes sur du matériel virtuel, chacun dans sa propre VM, où il est appelé système d’exploitation invité.

Deux mots portent l’essentiel de l’explication. La machine réelle est l’hôte, et chaque machine virtuelle est un invité. Le programme qui répartit l’hôte entre ses invités est l’hyperviseur, aussi appelé moniteur de machines virtuelles. SP 800-125 le décrit comme le composant qui gère les systèmes d’exploitation invités sur un hôte et contrôle le flux d’instructions entre eux et le matériel physique : processeur, disques, mémoire et cartes réseau. Le même guide cite les deux propriétés qui rendent ce dispositif utile. L’hyperviseur peut répartir les ressources de l’hôte pour que chaque invité accède aux ressources qui lui sont allouées et à celles explicitement partagées. Un invité peut aussi être entièrement encapsulé, ce qui le rend transportable d’un hôte à un autre.

Une machine virtuelle se distingue d’un conteneur par l’endroit où passe la frontière. Un conteneur est un processus sur le noyau de son hôte, avec une vue restreinte du système. Une machine virtuelle démarre son propre noyau sur le matériel que lui présente l’hyperviseur. Un invité Linux peut ainsi fonctionner sur

un hôte Windows ou macOS. Chaque invité a donc aussi besoin de son propre système d'exploitation, installé, corrigé et doté de suffisamment de mémoire pour fonctionner.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce qu'une machine virtuelle ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Des serveurs dimensionnés selon vos usages réels, migrés vers le cloud ou hors du cloud, avec une estimation préalable de la facture mensuelle.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. 20 applications Docker sur Google Cloud et AWS — 20 applications conteneurisées déployées et surveillées

20 applications Docker sur Google Cloud et AWS Contactez-nous →

Pourquoi c'est important

Un serveur loué à un fournisseur cloud est généralement une machine virtuelle, et l'infrastructure en tant que service repose sur ces machines. Le fournisseur exploite les hyperviseurs et le parc physique ; le client reçoit des machines qui peuvent être créées, redimensionnées et supprimées par une API au lieu d'être commandées, livrées et montées en baie. Le matériel reste à la charge du fournisseur, tandis que le client prend en charge le système d'exploitation.

La deuxième raison est l'isolation. Le site du projet KVM indique que chaque machine virtuelle possède du matériel virtualisé privé, avec sa propre carte réseau, son disque et son adaptateur graphique. Les invités ne partagent aucun noyau : un plantage du noyau ou une mauvaise mise à jour dans un invité reste dans cet invité. C'est l'hyperviseur, et non les invités, qui décide de ce qui atteint le matériel physique. Deux clients, ou deux environnements d'un même client, peuvent partager une machine physique sans partager un système d'exploitation.

Troisièmement, avec un stockage fondé sur des images, une machine peut devenir un ensemble de fichiers. Le disque d'un invité peut être un fichier image sur l'hôte, que l'on peut copier, conserver comme modèle ou utiliser pour démarrer plusieurs machines identiques. Le format qcow2 de QEMU peut ne consigner que les différences par rapport à une image de base et stocker des instantanés du disque dans le même fichier. Un modèle accompagné d'une couche mince par invité constitue donc une organisation naturelle.

Quatrièmement, l'émulation permet à un logiciel conçu pour une famille de processeurs de fonctionner sur une autre. Dans l'émulation système de QEMU, le système d'exploitation invité fonctionne sur une machine modélisée par logiciel, avec son propre processeur, sa mémoire et ses périphériques. Avec un accélérateur, tel que KVM sous Linux, Hypervisor Framework sous macOS ou Windows Hypervisor Platform sous Windows, la prise en charge de la virtualisation par le processeur hôte assure l'essentiel du travail. Sans accélérateur, le Tiny Code Generator de QEMU traduit les instructions de l'invité par logiciel. C'est ainsi que l'exemple de l'introduction de QEMU démarre une machine Arm64 avec `-accel tcg` .

La contrepartie est que chaque invité constitue un ordinateur complet à entretenir. Chacun possède son propre système d'exploitation à corriger, sa propre mémoire réservée et son propre démarrage. Les conteneurs évitent cela en partageant le noyau de l'hôte. Chacun est ainsi plus léger, mais aucun noyau distinct ne sépare les charges de travail. Choisir entre les deux revient à décider où cette frontière doit se trouver.

À quoi cela ressemble en pratique

Sous Linux sur x86, le duo habituel est KVM et QEMU. KVM est la partie noyau : un module chargeable, `kvm.ko` , accompagné d'un module propre au processeur, `kvm-intel.ko` ou `kvm-amd.ko` , intégré au noyau Linux principal depuis la version 2.6.20. Il nécessite les extensions de virtualisation du processeur, Intel VT ou AMD-V. La FAQ de KVM recherche leurs indicateurs, `vmx` et `svm` , dans `/proc/cpuinfo` . Sa liste de dépannage comprend aussi la vérification de l'existence de `/dev/kvm` et des droits de l'utilisateur pour l'ouvrir :

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

Un résultat supérieur à zéro signifie que le processeur annonce ces extensions. QEMU est la partie en espace utilisateur, qui construit la machine autour de KVM. Le disque d'un nouvel invité peut être une couche mince sur une image de base partagée :

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

Le manuel de qemu-img indique qu'une image créée avec un fichier de base ne consigne que les différences par rapport à celui-ci et que ce fichier de base n'est jamais modifié, sauf si ces différences y sont réintégrées. La seconde commande énumère tous les fichiers de la chaîne ; mieux vaut examiner cette liste avant de déplacer ou de supprimer quoi que ce soit. Le démarrage de l'invité prend ensuite cette forme :

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

`-accel kvm` confie le travail du processeur à KVM, par `/dev/kvm` dans la configuration initiale. `-m` fixe la mémoire de l'invité et `-smp` son nombre de processeurs. Pour `-drive`, `if=virtio` raccorde le disque à l'interface virtio, l'un des types d'interface énumérés par le manuel, et `format=qcow2` précise le format au lieu de laisser QEMU le détecter. `-nographic` désactive la fenêtre graphique : QEMU devient un simple programme en ligne de commande, et la console série de l'invité partage le terminal avec le moniteur QEMU ; Ctrl-a h indique comment passer de l'un à l'autre.

Une fois l'invité arrêté, le même outil crée et restaure des instantanés du disque. `-c` en crée un, `-l` les énumère et `-a` rétablit le disque dans un état sauvegardé :

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

Le dimensionnement est une décision de conception, pas un champ de formulaire. L'étude de cas sur l'exploitation des services de l'entreprise accessibles au public sur un petit serveur documente une configuration de septembre 2026 avec un cœur de processeur, un disque de 10 Go et 512 Mo de mémoire, dont environ 464 Mo sont utilisables par les programmes. Cet hôte dessert sept domaines en TLS et fait aussi tourner la forge git, avec une automatisation des sauvegardes implémentée séparément. La mémoire est la limite à laquelle répond chaque choix de conception. La possibilité d'y ajouter un service onion sur Tor a été tranchée par des relevés enregistrés pendant une

journée, dont la plus basse valeur de mémoire disponible avoisinait 310 Mo. Le réglage `-m` de chaque invité pose la même question, et une mesure y répond mieux qu'un abonnement offrant plus de ressources.

L'étude de cas sur l'administration de trente instances VPS Ubuntu Linux montre l'autre extrémité de l'échelle. Ces instances avaient été mises en place au fil des années sans conception d'ensemble, et leurs configurations avaient divergé. Elles ont ensuite été exploitées comme un parc unique : une même configuration réseau durcie sur chaque instance et des restaurations effectivement essayées au lieu d'être prises pour acquises.

Où les problèmes surviennent

La plupart des difficultés avec les machines virtuelles viennent de l'oubli que chacune est un véritable ordinateur à entretenir et qu'un disque fondé sur une image est aussi un fichier ordinaire sur une autre machine.

KVM manque au démarrage de l'invité. Dans la configuration x86 présentée plus haut, des extensions de processeur absentes ou désactivées dans le micrologiciel empêchent l'accélération KVM, et `/dev/kvm` peut être absent. Un invité lancé avec `-accel kvm` s'arrête donc immédiatement. La FAQ de KVM avertit que certains fabricants désactivent VT dans le BIOS d'une manière qui empêche toute réactivation. La solution consiste à vérifier les indicateurs comme ci-dessus, à rechercher dans le micrologiciel une option activant la virtualisation et à confirmer que `/dev/kvm` existe et appartient à un groupe dont l'utilisateur est membre ; la FAQ suggère un groupe nommé `kvm` .

Un invité lancé sans taille de mémoire reçoit 128 Mio. C'est la valeur utilisée par QEMU pour `-m` sans réglage explicite, quelle que soit la mémoire libre sur l'hôte. Un invité peut donc démarrer, puis échouer d'une manière qui ressemble à un défaut de son propre logiciel. La solution consiste à toujours définir `-m` et `-smp` et à les dimensionner à partir de mesures, comme l'étude du petit serveur a tranché la question du service onion par une journée de mesures.

Une image modifiée pendant que l'invité fonctionne peut être détruite. Le manuel de qemu-img est explicite : modifier une image utilisée par un invité ou un autre processus peut la détruire. Prendre un instantané du disque d'un invité en fonctionnement avec `qemu-img snapshot -c` , c'est précisément cela. La solution consiste à arrêter l'invité avant que qemu-img touche à son disque et à en faire une étape de toute procédure d'exploitation qui le manipule.

Les instantanés remplacent les sauvegardes. Les instantanés créés avec qemu-img résident dans le même fichier image, ce que `qemu-img info` montre en les énumérant à côté des informations du disque. Si le fichier est perdu ou endommagé, tous ses instantanés le sont aussi. La solution consiste à copier les sauvegardes sur le stockage d'une autre machine et à tester la restauration : une restauration jamais exécutée ne prouve rien.

Une image de base change ou se déplace sous ses surcouches. Une image qui ne conserve que les différences reste inutilisable sans la base exacte à partir de laquelle elle a été créée. Le manuel avertit que changer le fichier de base en mode non sûr corrompt le contenu visible par l'invité si le nouveau fichier est erroné. Un chemin relatif vers la base est résolu à partir du répertoire de la surcouche : déplacer un fichier sans l'autre casse donc la chaîne. La solution consiste à traiter les images de base en lecture seule, à donner un nouveau nom à chaque version, à lire `qemu-img info --backing-chain` avant tout déplacement et à n'exécuter `qemu-img commit` que délibérément, car il écrit dans la base et vide la surcouche.

Le format du disque est deviné d'après l'image elle-même. Sans `format=`, QEMU détecte le format à partir du contenu du fichier. Le manuel recommande `format=raw` pour éviter d'interpréter un en-tête de format non fiable. Une image téléchargée ailleurs ou dans laquelle un invité a écrit est non fiable précisément en ce sens. La solution consiste à préciser le format dans chaque ligne `-drive`.

Chaque invité devient un « animal de compagnie ». Des machines configurées à la main divergent une modification après l'autre jusqu'à ne plus se ressembler. C'était l'état du parc VPS ci-dessus avant sa prise en charge. La solution consiste à construire chaque invité à partir de la même image et de la même configuration, à conserver celle-ci sous contrôle de versions et à gérer les machines comme un parc unique avec une configuration de référence commune.

Pour aller plus loin

- Guide de sécurité des technologies de virtualisation complète (NIST SP 800-125) — la définition d'une VM, le rôle de l'hyperviseur et l'isolation des invités.
- KVM — la page du projet : modules, extensions de processeur et version d'intégration dans le noyau principal.
- FAQ de KVM — la vérification de vmx et svm, les droits sur `/dev/kvm` et le réglage du BIOS.

- Introduction à l'émulation système de QEMU — les accélérateurs, le Tiny Code Generator et une commande d'exemple complète.
- Invocation de QEMU — toutes les options utilisées ci-dessus : `-accel`, `-m`, `-smp`, `-drive`, `-nographic`.
- Utilitaire d'images disque de QEMU — qcow2, fichiers de base, instantanés et avertissement sur les images en cours d'utilisation.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le parcours de déploiement, les risques opérationnels et les personnes d'astreinte. Nous pouvons définir un changement avec des vérifications, une procédure de reprise et une responsabilité claire.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce qu'une machine virtuelle ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

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

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/virtual-machine/