Engineer

仮想マシン

物理ホスト上のソフトウェアで動くコンピューター一式

どのようなものか

仮想マシンは、ソフトウェアを通じてコンピューター一式を提供します。物理ホスト上の仮想化層がプロセッサー、メモリー、ディスク、ネットワークカードを用意し、専用ハードウェアであるかのように、その上でOSが起動します。VMと略されることもあります。一部の装置はエミュレートせず、ホストから直接渡すこともできます。NISTの完全仮想化の指針SP 800-125は、仮想マシンを「仮想化によって作られる模擬環境」と1行で定義しています。この指針が扱う仮想化では、1つ以上のOSがプログラムと共に、それぞれ専用のVM内の仮想ハードウェア上で動きます。そこで動くOSをゲストOSと呼びます。

周辺の説明では、主に2つの呼び名を使います。実機がホストで、それぞれの仮想マシンがゲストです。ホストをゲスト間で分けるプログラムがハイパーバイザーで、仮想マシンモニターとも呼びます。SP 800-125はこれを、ホスト上のゲストOSを管理し、ゲストと物理的なプロセッサー、ディスク、メモリー、ネットワークカードとの間で命令の流れを制御する構成要素と説明しています。同じ指針は、この仕組みの有用性を2点挙げています。ホストの資源を分割し、各ゲストが自分に割り当てられた資源と、明示的に共有された資源にアクセスするようにできます。また、ゲスト全体をひとまとめにできるため、別のホストへ移せます。

仮想マシンとコンテナーの違いは、どこに境界を引くかです。コンテナーはホストのカーネル上で動くプロセスで、見えるシステムの範囲を制限します。仮想マシンはハイパーバイザーが見せるハードウェア上で、自分自身のカーネルを起動します。だからこそWindowsやmacOSのホスト上でもLinuxゲストを動かせます。同時に各ゲストには、自分用のOSをインストールし、パッチを当て、動作に十分なメモリーを与える必要があります。

概念を判断に生かす

「仮想マシンとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 実際の使用量に合わせてサーバーを構成し、月額費用を事前に把握したうえで、クラウドへの移行もクラウドからの移転も進めます。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 Google Cloud・AWSでDockerアプリ20個を運用 — 20 デプロイし監視したコンテナアプリケーション

Google Cloud・AWSでDockerアプリ20個を運用 お問い合わせ →

なぜ重要なのか

クラウド事業者から借りるサーバーは通常、仮想マシンです。IaaSもこれを基盤にしています。事業者がハイパーバイザーと物理サーバー群を運用し、顧客はAPI経由で作成、サイズ変更、破棄できるマシンを受け取ります。注文して配送を待ち、ラックに取り付ける必要はありません。ハードウェアの責任は他者に残り、OSの責任は顧客に移ります。

第2の理由は分離です。KVMのプロジェクトページは、各仮想マシンが専用の仮想化ハードウェアを持ち、それぞれにネットワークカード、ディスク、グラフィックスアダプターがあると説明しています。ゲストはカーネルを共有しないため、1つのゲスト内のカーネル障害や不具合のある更新はその中にとどまります。物理ハードウェアに何を届けるかを決めるのはゲストではなくハイパーバイザーです。2者の顧客、あるいは同じ顧客の2つの環境が、OSを共有せずに物理マシンを共有できます。

第3に、イメージファイルを使うストレージなら、マシンを一組のファイルとして扱えます。ゲストのディスクをホスト上のイメージファイルにすれば、コピーしてテンプレートとして保存したり、同一のマシンを複数起動したりできます。QEMUのqcow2形式はベースイメージとの差分だけを記録し、同じファイル内にディスクのスナップショットを保存できます。そのため、1つのテンプレートと各ゲスト用の薄い差分層という構成が自然に成り立ちます。

第4に、エミュレーションを使うと、ある系統のプロセッサー向けに作られたソフトウェアを別の系統で動かせます。QEMUのシステムエミュレーションでは、ゲストOSは専用のプロセッサー、メモリー、装置を持つ、ソフトウェアでモデル化されたマシン上で動きます。LinuxのKVM、macOSのHypervisor Framework、WindowsのWindows Hypervisor Platformなどのアクセラレーターを使うと、重い処理をホストプロセッサーの仮想化支援機能が担います。アクセ

ラレーターがなければQEMUのTiny Code Generatorがゲストの命令をソフトウェアで変換します。QEMUの導入文書の例が `-accel tcg` でArm64マシンを起動するのは、この仕組みです。

代償は、すべてのゲストが管理対象のコンピューター一式になることです。各ゲストにはパッチを当てるOS、予約するメモリー、独立した起動があります。コンテナーはホストのカーネルを共有してこれを避けるため、1つずつは軽くなりますが、ワークロード間を隔てる独立したカーネルは手放します。両者の選択は、その壁をどこに置くかの選択です。

実際の活用例

x86 LinuxではKVMとQEMUの組み合わせが一般的です。KVMはカーネル側を担当します。ロード可能なモジュール `kvm.ko` と、プロセッサー別の `kvm-intel.ko` または `kvm-amd.ko` から成り、Linux 2.6.20以降のメインラインに含まれます。Intel VTやAMD-Vというプロセッサーの仮想化拡張が必要です。KVM FAQは `vmx` と `svm` フラグを `/proc/cpuinfo` で確認し、トラブル対処の項目には `/dev/kvm` が存在することと、利用者にそれを開く権限があることの確認も挙げています。

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

件数がゼロより大きければ、プロセッサーがその拡張を報告しています。QEMUはユーザー空間側を担当し、KVMの周りにマシンを組み立てます。新しいゲストのディスクは、共有ベースイメージに重ねる薄い差分層にできます。

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

qemu-imgのマニュアルによると、バッキングファイルを指定して作ったイメージが記録するのはそのファイルとの差分だけです。差分を元へコミットしない限り、バッキングファイル自体は変更しません。2番目のコマンドはチェーン内のすべてのファイルを列挙するため、何かを移動または削除する前に確認する価値があります。続いてゲストを起動します。

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

`-accel kvm` はプロセッサーの処理をKVMに渡します。既定では `/dev/kvm` を通ります。 `-m` はゲストのメモリー、 `-smp` はプロセッサー数を指定します。 `-drive` の `if=virtio` は、マニュアルに列挙されたインターフェースの1つであるvirtioにディスクを接続します。 `format=qcow2` はQEMUに自動検出させず形式を明示します。 `-nographic` はグラフィカルなウィンドウを無効にするため、QEMUは通常のコマンドラインプログラムとなり、端末をゲストのシリアルコンソールとQEMUモニターで共用します。Ctrl-a hを押すと両者の切り替え方法が表示されます。

ゲストを停止した状態なら、同じツールでディスクのスナップショットを取得し、復元できます。 `-c` で作成し、 `-l` で一覧を表示し、 `-a` で保存済みの状態へ戻します。

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

サイズ決定は、入力欄を埋めるだけの作業ではなく設計上の判断です。会社の公開サービスを小さなサーバー1台で動かす記録は、2026年9月時点の構成を記しています。プロセッサーは1コア、ディスクは10 GB、メモリーは512 MBで、プログラムが使えるのは約464 MBです。このホストはTLSで7ドメインを提供し、Gitのフォージも動かしています。バックアップ自動化は別途実装されており、そこではあらゆる設計判断がメモリーの制約に従います。Torのonionサービスも追加できるかは、1日分の記録されたサンプルで判断しました。サンプルで読み取った利用可能メモリーの最小値は約310 MBでした。どのゲストの `-m` 設定も同じ問いを投げかけます。大きなプランを選ぶより、測定の方がよい答えを出します。

Ubuntu LinuxのVPS 30台を管理した記録は、もう一方の規模を示します。それらの構成は設計されたのではなく何年もかけて積み重なり、互いにずれていました。それ以降は1つのサーバー群として運用しました。すべてのインスタンスに同じ堅牢化済みのネットワーク構成を適用し、復元できると信じるだけでなく、実際に復元を試しました。

問題が起きるところ

仮想マシンの問題の多くは、どの1台も管理が必要な実際のコンピューターであり、イメージファイルを使うディスクは別のマシン上の通常のファイルでもあることを忘れるために起きます。

ゲストの起動時にKVMが使えない。 上記のx86構成では、プロセッサーの拡張がない場合やファームウェアで無効にされている場合、KVMアクセラレーションを使えず、`/dev/kvm` が存在しないこともあります。そのため `-accel kvm` で起動したゲストは直ちに停止します。KVM FAQは、メーカーによってはBIOSでVTを再有効化できない形で無効にすると警告しています。対処は、上記のフラグ確認を行い、仮想化を有効にするファームウェア設定を探し、`/dev/kvm` が存在し、その所有グループに利用者が所属していることを確かめることです。FAQは `kvm` というグループを提案しています。

メモリー容量を指定せずに起動したゲストには128 MiBが割り当てられる。 これはQEMUの `-m` の既定値で、ホストの空きメモリー量には関係ありません。ゲストはいったん起動してから、内部のソフトウェアのバグに見える形で失敗することがあります。毎回 `-m` と `-smp` を明示し、測定に基づいてサイズを決めます。小さなサーバーの記録で、1日の測定によってonionサービスの追加可否を判断したのと同じです。

ゲストの稼働中にイメージを変更すると破損することがある。 qemu-imgのマニュアルは明確です。ゲストや別のプロセスが使用中のイメージを編集すると、そのイメージを破壊することがあります。`qemu-img snapshot -c` で稼働中のゲストのディスクをスナップショット化するのも、まさにその操作です。qemu-imgでディスクに触る前にゲストを停止し、それを使うすべての手順書に停止の工程を入れます。

スナップショットをバックアップの代わりにする。 qemu-imgで作るスナップショットは同じイメージファイルの中にあります。`qemu-img info` はディスクと共にそれらを一覧に表示します。ファイルが失われたり壊れたりすれば、すべてのスナップショットも失われます。別のマシンのストレージへバックアップをコピーし、復元を試します。一度も実行していない復元は、何も証明しません。

差分イメージの土台となるベースが変更または移動される。 差分イメージが保持するのは変更点だけなので、作成時のベースと完全に同じものがなければ意味を持ちません。マニュアルは、unsafeモードでバッキングファイルを変更して不適切なファイルを指定すると、ゲストから見える内容が壊れると警告しています。また、バッキングファイルの相対パスは差分イメージ自身のディレクトリー

を基準に解決されるため、片方だけ移動するとチェーンが切れます。ベースイメージは読み取り専用として扱い、新しい版には新しい名前を付け、移動前に `qemu-img info --backing-chain` を読みます。 `qemu-img commit` は意図して行う場合だけにします。ベースへ書き込み、差分イメージを空にするためです。

ディスク形式をイメージの内容から推測する。 `format=` がなければ、QEMUはファイルの内容から形式を検出します。マニュアルは、信頼できない形式ヘッダーを解釈しないために `format=raw` を推奨しています。他所からダウンロードしたイメージや、ゲストが書き込んだイメージは、まさにこの意味で信頼できません。すべての `-drive` 行で形式を明示します。

各ゲストが個別に世話をする存在へ変わっていく。 手作業で構成したマシンは、1回ずつの変更で互いにずれ、同じものが2台とない状態になります。上記のVPS群も、管理を統一する前はその状態でした。すべてのゲストを同じイメージと構成から作り、その構成をバージョン管理し、同じ基準を持つ1つのサーバー群として管理します。

参考資料

- 完全仮想化技術のセキュリティ指針（NIST SP 800-125） — VMの定義、ハイパーバイザーの役割、ゲストを分離する方法。
- KVM — モジュール、プロセッサー拡張、メインラインへの導入バージョンを説明するプロジェクトページ。
- KVM FAQ — vmxとsvmの確認、 `/dev/kvm` の権限、BIOS設定。
- QEMUシステムエミュレーション入門 — アクセラレーター、Tiny Code Generator、完全なコマンド例。
- QEMUの起動 — 上記で使うすべてのオプション： `-accel` 、 `-m` 、 `-smp` 、 `-drive` 、 `-nographic` 。
- QEMUディスクイメージユーティリティー — qcow2、バッキングファイル、スナップショット、使用中のイメージについての警告。

用語集

実際の制約に合わせて選ぶ

デプロイの流れ、運用上のリスク、緊急対応の担当者をお知らせください。検査、復旧手段、明確な責任の所在を伴う変更の範囲を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「仮想マシンとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

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

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/virtual-machine/