Engineer

ネットワークサーバー

ネットワーク経由で他の相手にサービスを提供するコンピューターやプログラム

どのようなものか

ネットワークサーバーは、他のプログラムがネットワーク越しに接続してくるのを待ち、届いた要求に応答するプログラムです。日常的には単に_サーバー_と呼び、そのプログラムと、それを動かすコンピューターの両方を指します。それぞれのやり取りの相手側にいるプログラムがクライアントです。ブラウザー、スマートフォンのアプリ、スクリプト、メールソフト、あるいは自分の問い合わせを送る別のサーバーなどが該当します。

ウェブを支える標準は、まず役割を定義します。HTTPを定義するRFC 9110は、HTTPサーバーを「HTTP応答を送ってHTTPリクエストを処理するために、接続を受け付けるプログラム」と説明し、クライアントとサーバーという言葉は、特定の接続でプログラムが果たす役割だけを表すと補足しています。同じプログラムが、ある接続では要求に応じ、次の接続では問い合わせることもあります。背後のアプリケーションへリクエストを転送するウェブサーバーは、その2つ目の接続ではアプリケーションのクライアントです。

ほとんどのサーバーの土台にあるのは、TCPでパッシブオープンと呼ぶ操作です。プログラムは、特定のローカルアドレス、またはすべてのローカルアドレスで着信接続を受け付けたいとOSに伝え、そのまま待ちます。クライアントは、そのアドレスとポートへアクティブオープンを行います。誰も接続していない間、サーバーの目に見える動きはほとんどありません。

待ち方はサーバーによって異なります。多くは何か月も連続して動き、訪問者の有無にかかわらずメモリーを確保し続けます。一方、接続のたびに新しく起動するものもあります。監督するプログラムが待機し、訪問者ごとにサーバーのコピーを起動し、訪問者が去るとそのコピーが終了します。

次に、機械としての意味があります。NISTの用語集は、サーバーを「ネットワークリソースを管理する、ネットワーク上のコンピューターまたは機器」と説明し、ファイルサーバー、プリントサーバー、ネットワークサーバー、データベースサーバーを例に挙げています。企業がサイトを小さなサーバー1台で動かしていると言う場合、サーバープログラムを動かすために用意したコンピューターを指します。現在では、通常は事業者から借りる仮想マシンです。この用語集では、2つの言葉を明確に使い分けています。応答するものがサーバーです。名前とアドレスを持ち、構築し、安全を確保し、台数を数える対象としての機械がホストです。1台のホストで多数のサーバーを動かせます。また、名前も一緒に移せば、サーバーは別のホストへ移動できます。

よく使う種類は、何を提供するかで名付けられています。HTTPのウェブサーバー、SMTPとIMAPのメールサーバー、名前を扱うDNSサーバー、クエリに答えるデータベースサーバー、ファイルサーバー、gitサーバー、ゲームサーバーなどです。アプリケーションサーバーは、通常はウェブサーバーの背後に置かれ、製品そのもののコードを実行します。

概念を判断に生かす

「ネットワークサーバーとは？」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。 サーバー、メール、アカウント、更新を日々管理し、誰もが頼る仕組みを平穏に動かし続けます。

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。 UbuntuでPostgreSQL・MS SQLサーバー10台を運用 — 10 導入し正常に保ったDBサーバー

UbuntuでPostgreSQL・MS SQLサーバー10台を運用 お問い合わせ →

なぜ重要なのか

インターネットで訪問者がアクセスするものには、どれも何らかのサーバーが応答します。そのため、サーバーを選ぶことは、何を外部に公開するか、いくらかかるか、何が一緒に止まるかを選ぶことです。

予算の話でこの言葉を使う際、最初に役立つのは、どの意味で話しているのかを確認することです。「サーバーが必要です」は、インストールするプログラム、借りる機械、他社が運用するマネージドサービス、あるいは支払い済みの機械に追加する処理能力を意味し得ます。それぞれ、価格、セキュリティ更新を適用する担当者、停止したときに夜中に起こされる人が異なります。

2つ目は費用です。サーバーの費用の大半は、待っている時間にかかります。何もしていなくてもメモリーを保持するプログラムは、24時間そのメモリーを占有します。小さな機械では、常駐サーバーの1つ1つが予算の項目になります。接続ごとに起動するサーバーは、訪問のたびに少し時間がかかる代わりに、訪問と訪問の間は何も保持しません。

3つ目は外部への公開です。接続を待ち受けるサーバーは、どれも見知らぬ相手から入力を受け付けます。そのため、パッチを適用し、意図に沿って設定し、監視する必要があります。目的を果たせる最小限のサーバーだけを動かすことは、節約と同じくらいセキュリティ上の判断でもあります。

4つ目は、自前で動かさずにその役割を購入できることです。マネージドデータベース、ホスティング型のメールボックス、コンテンツ配信ネットワークは、他社が運用するサーバーです。自分でパッチを当てる対象を減らす代わりに、制御できる範囲と内部を把握できる範囲を手放します。小さな企業がウェブサーバーは自前で動かし、メールは借りるのも合理的な選択です。

実際の活用例

当サイトのサーバーはすべて1台の機械に収まり、クレジットページにその上で動くすべてのサーバープログラムを記載しています。Caddyはすべてのウェブリクエストに応じ、HTTPS証明書も自分で更新します。AgateとGophernicusはGeminiとGopherに、Torはonionアドレスに応じ、Soft Serveはgitサーバーを担います。サイトの4つの公開版は1つの共有フォルダーから配信し、1回のデプロイですべてを同時に更新します。onionサービスは、配信することとポートで待ち受けることが別である例です。torがイントロダクションポイントへ外向きに接続するため、ポートを開かずに追加できた唯一の入口であり、機械で開いているポートはまったく変わりませんでした。メールはホスティングサービスへ送り、自前で動かす代わりに役割を購入しています。

プライバシーページは、接続ごとに起動する方式を、あまり注目されない角度から示しています。Gopher版は、訪問者ごとにOSが新しく起動します。OSは、その起動に訪問元のアドレスに基づく名前を付け、サーバーソフトウェアからは見えない独自のログへ記録します。当サイトがそのログを発見したのは、初めてリクエスト数を数えた日でした。現在、そのログの保存期間は30日です。

また、機械が1台であることは、載せられるサーバーを制限します。会社の公開サービスを512 MBのホスト1台で動かす記事では、意図的にその機械へメールを載せていません。メールサーバーは、専用のホストを待つインストール用のひな

型として存在します。メールサーバーが必要とする余力は、その機械ではすでに使い切っているためです。後から判明するのではなく、あらかじめ記録してある制約です。

問題が起きるところ

機械は応答しても、サーバーは応答しません。 ping、開いたSSHセッション、仮想マシンの状態を示す緑の表示は、どれもホストが動いていることを示します。しかし、ウェブサーバーが応答すること、正しいページを返すこと、証明書がまだ有効であることは示しません。クライアントが使うのと同じ方法で、実際に問い合わせ、答えを読んで確認します。

```
curl -sS -o /dev/null -w '%{http_code} %{time_total}s\n' https://example.com/
```

訪問者が必要とするのは、妥当な時間内に返る200です。それ以外なら、機械のグラフが何を示していても障害です。

ディスク上のファイルと、動作中のサーバーの状態が違います。 多くのサーバーは、起動時や再読み込み時に設定を読んでメモリーに保持します。そのため、ファイルを編集しても、次の再読み込みまでは何も変わりません。壊れたファイルで再起動すれば、動くサーバーがなくなります。ファイルを検証し、動作中のサーバーに渡し、その後サーバーへ問い合わせます。Caddyのコマンドラインは最初の2段階に対応しています。`validate` は、設定を起動するのと同じようにすべてのモジュールを読み込んで準備しますが、実際には起動しません。`reload` は、新しい設定を動作中のインスタンスへ渡します。

```
caddy validate --config /etc/caddy/Caddyfile
caddy reload --config /etc/caddy/Caddyfile
```

サーバーを2つにしても、1つ分の仕事が半分になるとは限りません。 ロードバランサーの背後にアプリケーションサーバーをもう1つ追加すると、最初のサーバーが持つ状態のすべてを見直す必要があります。メモリー内のセッションは片方だけのものなので、もう片方へ送られた訪問者はログアウトします。一方のディスクへアップロードしたファイルは、他方にはありません。両方に入れた定期処理は2回実行されます。2つ目を追加する前に、セッション、アップロード、実行予定を両方で共有する場所へ移すか、1つのサーバーのまま高速化します。

1台の機械上のサーバーは、すべて運命を共にします。 1台のホストで多数のサーバーを動かすのは安く単純ですが、再起動1回、ディスク容量の枯渇1回、メモリー不足1回で、すべてのサーバーが同時に止まります。小さなサイトなら、意識して選んだ場合は妥当な選択です。再構築できるよう機械をコードで記述し、データは機械の外へバックアップし、専用の余力が必要なサーバーには専用の機械を用意します。

参考資料

- server — NIST CSRC用語集 — NISTの各出版物が共有する定義と、そこで挙げられるサーバーの種類です。
- RFC 9110、HTTP Semantics：接続、クライアント、サーバー — 1つの接続でプログラムが果たす役割として、クライアントとサーバーを定義しています。
- RFC 9293、Transmission Control Protocol — パッシブオープンとアクティブオープン、待ち受けるプログラムが1つのアドレスまたはすべてのアドレスで待つ仕組みを説明しています。
- Caddy：コマンドライン — 当サイトが使うウェブサーバーの `validate` 、 `reload` などのコマンドです。

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

「ネットワークサーバーとは？」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/server/