Engineer

リバースプロキシ

← ウェブサービスに代わってリクエストを受けるサーバー

どのようなものか

リバースプロキシは、ウェブサービスに代わってリクエストを受けるサーバーです。ブラウザーが公開アドレスに接続すると、まずリバースプロキシがリクエストを受け取ります。処理を担当するサーバーへ転送し、そこから戻ったレスポンスをブラウザーへ渡します。背後で応答するサーバーをオリジンサーバーと呼びます。オリジンを別のマシンやポートに移しても、利用者は以前と同じ公開URLを使えます。HTTPの仕様では、この役割をゲートウェイとも呼びます。フォワードプロキシはクライアント側に置かれ、クライアントに代わって外部の宛先に接続します。

たとえば、トップページはあるサービスが提供し、`/api/reports/42` にあるレポートは別のサービスが提供する構成を考えます。ブラウザーはどちらも同じドメインにリクエストします。プロキシはHTTPメッセージからパスを読み、転送先を選びます。同じレポートを処理できるサーバーが複数あれば、作業を分散することもできます。これは負荷分散です。背後にサーバーが一台の場合も、内部アドレスと公開URLを切り離すためにリバースプロキシを使えます。パスに応じて別のサービスを選ぶことと、同じ処理を複数のサーバーに割り振ることを、構成図では別々に示すと判断しやすくなります。

製品によっては、暗号化されたTCP接続をそのまま転送先へ中継できます。この方式でも、接続に関する情報や、利用できる場合はTLSの接続開始時に見える情報を用いて転送先を選べます。暗号化されたHTTPのパスを選択条件にするには、HTTPメッセージを読める場所が必要です。構成図に、ブラウザーからのTLS接続がどこで終わり、オリジンまで新しいTLS接続が張られるかを書き込みます。どこでどの情報が見えるかを確認できます。

概念を判断に生かす

「リバースプロキシとは？」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。 Wi-Fi、ファイアウォール、リモートアクセスを整え、どこからでも働け、権限のある人だけが接続できるようにします。

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 Linux VPSインスタンス30台を管理 — 30 復旧計画とともに管理したLinuxサーバー

Ubuntu Linux VPSインスタンス30台を管理 お問い合わせ →

なぜ重要なのか

入口を一つ設けると、外部から使えるサービスを定義しやすくなります。公開ドメインの証明書をプロキシで管理し、アプリケーションは内部ネットワークで待ち受ける構成を取れます。入口でアクセスログを記録したり、リクエストの大きさを制限したり、古いURLから転送したりすることもできます。設定はパスの使い方に合わせます。ファイルを受け付けるパスでは十分な上限が必要です。データを逐次送るパスでは、応答をまとめてから送る設定によって表示が遅れることがあります。

プロキシがTLSを終端すると、リクエストを復号します。処理する情報には、Cookie、検索の条件、閲覧者が送信したデータが含まれることがあります。オリジンへの接続を改めて暗号化すれば、次の区間を流れる通信を保護できます。復号するプロキシの運用者はその内容を扱えるため、信頼する範囲にはこの運用者も含まれます。どの項目を記録し、誰がログを読めて、いつ消去するかは、サイトがデータの扱いを説明する前に決める事項です。

通信の経路が一つ増えるので、障害時の調べ方も変わります。ゲートウェイが上流から無効なレスポンスを受け取ると `502` を返すことがあります。必要な応答が期限内に返らなければ `504` です。アプリケーションが独自にエラーを返す場合もあります。時刻を記録したテスト用のリクエストを一件送り、プロキシとオリジンの両方で同じ処理を探します。オリジンに記録がなければ、アプリケーションの処理に加えて、そこまでの接続も確認できます。

実際の活用例

Engineer ApSのDNSゾーンをコードで管理し、実際の設定と照合した事例では、正常に機能したCDNプロキシを取り外した経緯が説明されています。CDNを経由すると、訪問者のアドレスと要求したURLを、まず事業者のネットワークが扱います。この事例では、同社は追加の攻撃対策よりも訪問者のプライバシーを優先

しました。DNSゾーンは引き続き変更履歴を管理し、監査できます。名前を解決するサービスをどこに任せるかと、ウェブのリクエストそのものを誰が受けるかを、別の判断として考えられる例です。

動作を確認するなら、検証環境で `/health` と `/api/reports/42?view=summary` にアクセスします。アプリケーションに届いたメソッドとパス、ブラウザーで受け取ったステータス、リダイレクト先のスキームとホスト名を記録します。レポートを返すサーバーの内部アドレスを変え、同じ公開URLでもう一度試します。移行後も、利用者が保存しているURLの意味が変わらないことを確認できます。

共有キャッシュを使う前に、二つのテスト用アカウントで別々のセッションを開きます。それぞれのレスポンスに無害で異なる目印を出し、毎回そのセッションに対応する内容だけが届くか確認します。最後に検証環境のオリジンを意図的に停止し、待ち時間、エラーの内容、両側のログを観察します。アプリケーションから返す別のエラーも一度試しておけば、本番で似た症状が出たときに異なる原因を見分ける材料になります。

問題が起きるところ

外部から届くリクエストには、利用者が自分で追加した `Forwarded` や `X-Forwarded-For` が含まれることがあります。アプリケーションがこれらの値を使うなら、どの中継点が送る値を信頼するか決めます。外部からの値を入口で取り除くか置き換え、オリジンに直接接続できる相手も制限します。検証時には意図的に誤ったアドレスを含めて送り、ログに残る値と、アクセス許可の判断に実際に使われた値を確認します。

リダイレクトが繰り返されたり、画面のリンクに内部ポートが現れたりするのもよくある問題です。ブラウザーからはHTTPSで接続したのに、アプリケーションが手前の接続を見てHTTPのURLを作った場合に起こります。公開されるスキームとホスト名を検証し、その情報をアプリケーションへ渡します。ログインの最初から戻り先に着くまでを通して、リダイレクトとCookieを確認します。パスの先頭を書き換える場合は、複数の区切りとクエリーを含む入力で、どの部分が変わったか確かめます。

タイムアウト後の再送には、処理の重複という問題があります。プロキシが待つのをやめた時点で、アプリケーションはすでに支払いを記録しているかもしれません。データを変更するリクエストの自動再試行を認める前に、アプリケーションが同じ操作の繰り返しをどう見分けるか決めておきます。検証環境で処理を記録してから応答だけを遅らせます。どの段階で接続が打ち切られ、後から最初の処理と再試行を区別できるかを調べます。

参考資料

- RFC 9110のHTTP中継に関する節は、プロキシ、ゲートウェイ、トンネルの役割を定義しています。
- RFC 7239のForwarded仕様は、ヘッダーの形式とセキュリティーおよびプライバシー上の注意点を説明しています。
- NGINXのHTTPプロキシモジュールは、転送先、ヘッダー、バッファー、タイムアウトの設定をまとめています。
- Caddyのreverse_proxyディレクティブは、上流サーバーの選び方と転送ヘッダーの扱いを説明しています。
- Apache HTTP Serverのリバースプロキシガイドは、公開サーバーから内部のサービスへリクエストを送る構成を例示しています。

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

「リバースプロキシとは？」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/reverse-proxy/