Engineer

nginx

ウェブサーバー兼リバースプロキシ

どのようなものか

nginxは、HTTP接続を受け付け、リクエストごとに、ディスク上のファイルで応答するか、別のサーバーへ渡してその応答を中継する、オープンソースのプログラムです。名前は「engine x」と発音し、プロジェクト自身のページでもNGINXと表記します。Igor Sysoevが開発し、最初の公開版0.1.0は2004年10月4日に登場しました。2条項BSDライセンスで無料で利用でき、F5, Inc.は企業向けのディストリビューション、サポート、研修を販売しています。

プロジェクトは、HTTPサーバー、リバースプロキシ、コンテンツキャッシュ、ロードバランサー、TCPおよびUDPプロキシ、メールプロキシという六つの役割を挙げます。ウェブホストでは通常、二つの仕事の一方、または両方を担います。ディスクから静的ファイルを配信することと、アプリケーションサーバーの前に立ってTLSを終端し、各リクエストを内部へ転送することです。

起動時には、単一のマスタープロセスと、複数のワーカープロセスが動きます。マスターは設定を読み取って検査し、ワーカーを管理します。ワーカーは特権のないユーザーで動き、リクエスト処理をすべて担います。各ワーカーは、Linuxではepoll、FreeBSDではkqueueに基づくイベントループを使うため、少数でも数千の接続を同時に開いておけます。worker_processesディレクティブで数を指定し、固定数か、CPUコアごとに一つを割り当てるautoを選びます。プロジェクトの機能一覧では、アイドル状態のキープアライブ接続10,000本に必要なメモリーは約2.5 MBとしています。

設定は、ディレクティブを記したテキストファイルです。単純なディレクティブは名前と引数からなり、セミコロンで終わります。ブロックディレクティブは、ほかのディレクティブを波括弧で囲み、ディレクティブを含められるブロックをコンテキストと呼びます。eventsとhttpのコンテキストは最上位にあります。httpの中では、各serverブロックが一つの仮想サーバーを記述します。まず待ち

受けるアドレスとポート、次にクライアントが要求したホスト名で選ばれます。serverブロック内の各locationブロックは、ある一群のリクエストパスをどう処理するかを定めます。

概念を判断に生かす

「nginx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

ウェブサーバーは、すべての訪問者が接するプログラムなので、負荷がかかったときの動作と既定値が全員に影響します。nginxを選ぶ理由となる性質は二つあります。

一つ目は、遅い接続を低いコストで維持できることです。プロキシ先サーバーの応答のバッファリングは、既定で有効です。nginxは、バックエンドが応答を生成する速さでそれを受け取り、メモリーに収まらなくなれば一時ファイルへ書き出し、クライアントが受け取れる速さで送ります。電波の弱い携帯電話が前の応答をまだダウンロードしている間にも、アプリケーションサーバーは次のリクエストを処理できます。

二つ目は、リクエストを切らずに設定変更を反映できることです。設定の編集が有効になるのは、再読み込みか再起動のときだけです。再読み込みでは、マスターが新しい設定を検査し、適用を試みます。成功すると、新しいワーカーがその設定で起動し、古いワーカーは接続の受け付けを止め、抱えているリクエストを完了して終了します。失敗した場合は、マスターが変更を戻し、それまでの設定で配信を続けます。プロジェクトは、クライアントの通信を中断せず、バイナリー自体を更新する機能も挙げています。

その代償は設定言語にあり、優先順位と継承の規則が、意外な動作の大半を生みます。リクエストパスに対して、nginxはまず、一致するプレフィックスが最も長いlocationを見つけて覚えます。次に、正規表現のlocationをファイルに書かれた順に調べ、最初の一致を採用します。正規表現がどれも一致しなければ、覚えたプレフィックスを使います。 `=` 付きのlocationは、パスが完全一致すると検索を直ちに終え、最長のプレフィックスに `^~` が付いていれば、正規表現を調

べません。add_headerやproxy_set_headerなどのディレクティブは、現在の階層に、そのディレクティブの独自定義が一つもないときだけ、外側の階層から継承されます。

証明書は長い間、nginxと並行して動くACMEクライアントから取得していました。現在は、コアのビルドとは別に配布するngx_http_acme_moduleも提供しています。

実際の活用例

創業者の以前の職務で行った二つのホスティング業務に、nginxが使われています。どちらもApacheを置き換えず、同じサーバーで併用していました。

一つ目は、高速で、常にアクセスできる必要があった四十の顧客サイトです。ApacheとnginxのUbuntu Linuxサーバーで動かしていました。設定を書き、速度を調整し、実際の訪問者が利用している間も保守する仕事で、四十サイトすべてが高い可用性を維持しました。同じサーバー群で二つのウェブサーバーを使うと、すべての規則を二つの方言で表す必要があります。Apacheの仮想ホストはnginxのserverブロックに当たり、一方で見つけた修正は、もう一方に合わせて訳す必要があります。

二つ目は2015年から2022年にかけての業務です。ACMEの処理フローで、百のDockerアプリケーションの証明書管理を自動化しました。各証明書を発行し、期限前に更新し、人手を介さずUbuntuホストへ配置しており、そのホストでもApacheとnginxを併用していました。nginx側では、ディスク上のファイルを更新しても、変更は半分しか終わっていません。単純なパスで指定した証明書は、設定の読み込み時に読み取ります。ドキュメントによると、TLSハンドシェイクのたびに証明書を読み直すのは、変数から組み立てたファイル名の場合だけです。新しいファイルを書き込む更新が訪問者に届くのは、再読み込みの後です。そのため、nginx向けの自動更新は、最後に再読み込みを行う必要があります。

問題が起きるところ

サイト全体に設定したヘッダーが、一つのパスで消えます。 locationに独自のadd_headerが一つでもあれば、serverやhttp階層のadd_headerを一つも継承しません。一つのlocationにキャッシュヘッダーを足すだけで、上位にあるセキュリティーヘッダーがすべて、気付かないうちに消えます。proxy_set_headerも同じ規則です。add_headerは、`always`を付けなければ、200、301、304などの成功とリダイレクトのコードにしか適用しないため、エラーページには、その追加ヘッダーが付きません。共有するヘッダーを一つのファイルにまとめ、独自の設

定を持つ各階層でincludeするか、1.29.3以降なら `add_header_inherit merge;` を設定します。エラー応答でも必要な場所には `always` を付け、各パスを `curl -sI` で確認します。

あるサイトへのリクエストに、別のサイトが応答します。 nginxは、先にアドレスとポートを照合し、次にHostヘッダーをserver_nameと照合します。そのポートのどのserverブロックも名前を認識しなければ、既定のサーバーへ送ります。別のサーバーにdefault_serverを指定しない限り、最初に列挙したものが既定です。新しいサイトのserver_nameに入力ミスがあると、以前からある隣のサイトのページを、気付かないうちに配信します。未知の名前には、接続を閉じる `return 444` で応答する既定のサーバーを明示します。間違ったサイトを表示するのでなく、エラーとして分かるようにするためです。

アップロードが413 Request Entity Too Largeで失敗します。
client_max_body_sizeの既定値は1 MBで、それより大きいリクエストボディーは、アプリケーションに届く前にnginxが拒否します。サーバー全体でなく、アップロードを受け付けるlocationで上限を上げ、アプリケーション側の上限とも揃えます。

バックエンドが、違うホストへのリンクを作るか、違うパスを受け取ります。
別の指定がなければ、nginxは、 `127.0.0.1:8080` などのproxy_pass内のアドレスを基にしたHostヘッダーを、プロキシ先へ送ります。そのため、アプリケーションは、そのアドレスへのリダイレクトや絶対リンクを生成します。プロキシするlocationには、 `proxy_set_header Host $host;` を設定します。パスの扱いはproxy_passの書き方で変わります。URI付きなら、単に `/` だけでも、locationに一致したパスの部分を置き換え、URIなしなら、パスを変えずに渡します。どの変更の後も、入れ子になったパスを試します。

クライアントが、古い設定や証明書を受け取り続けます。 ファイルを編集したり、夜間に証明書が更新されたりしても、再読み込みまでは、稼働中のnginxに影響しません。壊れた設定の再読み込みでは、古い設定が残ります。毎回、再読み込み前にテストし、その後、エラーログを読み、クライアントが実際に受け取るものを確認します。

```
nginx -t && nginx -s reload
openssl s_client -connect example.com:443 \
  -servername example.com </dev/null |
  openssl x509 -noout -enddate
```

参考資料

- nginxドキュメント — ガイドと、モジュールごとのディレクティブ参照資料をまとめた、プロジェクトの索引です。
- 入門ガイド — マスターとワーカーのプロセス、再読み込み、設定の構造です。
- nginxによるリクエスト処理 — アドレス、ポート、Hostヘッダーからserverブロックを選ぶ方法です。
- ngx_http_core_moduleモジュール — locationの照合順序、server_name、client_max_body_sizeです。
- ngx_http_proxy_moduleモジュール — proxy_pass、ヘッダーの既定値、タイムアウト、応答のバッファリングです。
- ngx_http_headers_moduleモジュール — add_header、対象の応答コード、継承規則です。

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

「nginx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/nginx/