Engineer

nginx

ein Webserver und Reverse Proxy

Was es ist

nginx ist ein quelloffenes Programm, das HTTP-Verbindungen annimmt und jede Anfrage entweder mit Dateien von der Platte beantwortet oder an einen anderen Server weitergibt und dessen Antwort zurückreicht. Der Name wird „engine x“ ausgesprochen; die Projektseiten schreiben ihn auch NGINX. Igor Sysoev entwickelte das Programm. Die erste öffentliche Version, 0.1.0, erschien am 4. Oktober 2004. Es lässt sich unter einer BSD-Lizenz mit 2 Klauseln kostenlos nutzen. F5, Inc. verkauft Unternehmensdistributionen, Support und Schulungen.

Das Projekt nennt sechs Rollen: HTTP-Server, Reverse Proxy, Inhaltscache, Lastverteiler, TCP- und UDP-Proxy sowie Mail-Proxy. Auf einem Webhost übernimmt es gewöhnlich eine oder beide von zwei Aufgaben: statische Dateien von der Platte ausliefern und einem Anwendungsserver vorgeschaltet sein, dabei TLS terminieren und jede Anfrage nach innen weiterleiten.

Beim Start erzeugt nginx einen einzelnen Masterprozess und mehrere Workerprozesse. Der Master liest und prüft die Konfiguration und betreut die Worker. Die Worker laufen unter einem unprivilegierten Benutzer und übernehmen die gesamte Anfragebearbeitung. Jeder verwendet eine ereignisbasierte Schleife, unter Linux auf epoll und unter FreeBSD auf kqueue aufgebaut. Wenige Worker können damit Tausende Verbindungen zugleich offen halten. Die Direktive worker_processes legt ihre Zahl fest: als feste Zahl oder mit auto für einen pro CPU-Kern. Laut Funktionsliste des Projekts benötigen 10.000 inaktive Keep-Alive-Verbindungen etwa 2,5 MB Speicher.

Die Konfiguration ist eine Textdatei aus Direktiven. Eine einfache Direktive besteht aus Name und Parametern und endet mit einem Semikolon. Eine Blockdirektive umschließt weitere Direktiven mit geschweiften Klammern. Ein Block, der Direktiven enthalten kann, heißt Kontext. Die Kontexte events und

http stehen auf oberster Ebene. In http beschreibt jeder server-Block einen virtuellen Server. Gewählt wird er zunächst nach der Adresse und dem Port, an denen er lauscht, und dann nach dem vom Client angefragten Hostnamen. In einem server-Block sagt jeder location-Block, was mit einer Gruppe von Anfragepfaden geschehen soll.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist nginx?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Server, E-Mail, Konten und Updates werden im Alltag betreut, damit die Dinge, auf die alle angewiesen sind, unauffällig funktionieren.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. 10 PostgreSQL-/MS-SQL-Server auf Ubuntu — 10 bereitgestellte und funktionsfähig gehaltene Datenbankserver

10 PostgreSQL-/MS-SQL-Server auf Ubuntu Kontakt aufnehmen →

Warum es wichtig ist

Der Webserver ist das eine Programm, mit dem jeder Besucher in Kontakt kommt. Sein Verhalten unter Last und seine Voreinstellungen betreffen deshalb alle. Zwei Eigenschaften sprechen für nginx.

Die erste ist, wie günstig es langsame Verbindungen bedient. Die Pufferung von Antworten eines nachgelagerten Servers ist standardmäßig aktiviert: nginx übernimmt die Antwort des Backends so schnell, wie dieses sie erzeugen kann, lagert bei zu wenig Speicher in eine temporäre Datei aus und liefert sie dann im Tempo des Clients aus. Der Anwendungsserver ist für die nächste Anfrage frei, während ein Telefon mit schwachem Empfang die vorige Antwort noch herunterlädt.

Die zweite ist, dass Konfigurationsänderungen ohne Abbruch von Anfragen wirksam werden. Änderungen greifen erst beim Neuladen oder Neustart. Beim Neuladen prüft der Master die neue Konfiguration und versucht, sie anzuwenden. Gelingt das, starten neue Worker damit. Die alten nehmen keine neuen Verbindungen mehr an, schließen die laufenden Anfragen ab und beenden sich anschließend. Scheitert der Versuch, macht der Master die Änderung rückgängig und bedient Anfragen weiter mit der vorhandenen Konfiguration. Das Projekt nennt auch den Austausch der Programmdatei selbst ohne Unterbrechung für Clients.

Der Preis steckt in der Konfigurationssprache. Ihre Regeln zu Vorrang und Vererbung verursachen die meisten Überraschungen. Für einen Anfragepfad sucht nginx zunächst den location-Block mit dem längsten passenden Präfix und merkt ihn sich. Dann prüft es die location-Blöcke mit regulären Ausdrücken in der Reihenfolge der Datei und nimmt den ersten Treffer. Der gemerkte Block greift, wenn kein regulärer Ausdruck passt. Eine mit `=` markierte location trifft den Pfad exakt und beendet die Suche sofort. `^~` am längsten Präfix weist nginx an, die regulären Ausdrücke zu überspringen. Manche Direktiven, darunter add_header und proxy_set_header, werden nur von der umgebenden Ebene geerbt, wenn die aktuelle Ebene keine eigenen definiert.

Zertifikate kamen lange von einem ACME-Client, der neben nginx lief. Das Projekt bietet inzwischen auch ngx_http_acme_module an, das getrennt vom Kern-Build ausgeliefert wird.

Wie es in der Praxis aussieht

Zwei Hosting-Arbeiten aus früheren Tätigkeiten der Gründerin betrafen nginx. In beiden teilte es die Server mit Apache, statt Apache zu ersetzen.

Die erste umfasste vierzig Kundenwebsites, die schnell und jederzeit erreichbar sein mussten. Sie liefen auf Ubuntu-Linux-Servern, auf denen Apache und nginx eingesetzt wurden. Die Aufgabe bestand darin, ihre Konfiguration zu schreiben, sie auf Geschwindigkeit abzustimmen und sie bei laufendem Besucherverkehr zu betreuen. Alle vierzig blieben hochverfügbar. Bei zwei Webservern in einer Serverflotte muss jede Regel in zwei Dialekten vorliegen: Ein virtueller Host in Apache ist ein server-Block in nginx. Eine Korrektur für den einen Webserver muss für den anderen übersetzt werden.

Die zweite lief von 2015 bis 2022. Ein ACME-Ablauf übernahm die Zertifikate für hundert Docker-Anwendungen: ausstellen, vor Ablauf erneuern und auf den Ubuntu-Hosts installieren, ohne menschlichen Eingriff. Auch dort liefen Apache und nginx nebeneinander. Auf der nginx-Seite ist eine erneuerte Datei auf der Platte nur die halbe Änderung. Ein Zertifikat mit einem einfachen Dateipfad wird beim Laden der Konfiguration gelesen. Laut Dokumentation bewirkt nur ein aus Variablen zusammengesetzter Dateiname ein erneutes Laden bei jedem TLS-Handshake. Neue Zertifikatsdateien erreichen Besucher daher erst nach einem Neuladen. Jede automatisierte Erneuerung für nginx muss damit enden.

Wo es schiefgeht

Ein für die ganze Website gesetzter Header verschwindet auf einem Pfad. Eine location mit eigenem add_header erbt keinen aus der server- oder http-Ebene. Ein einzelner Cache-Header in einer location entfernt damit unbemerkt alle darüber gesetzten Sicherheitsheader. proxy_set_header folgt derselben Regel. add_header gilt außerdem nur für Erfolgs- und Weiterleitungscodes wie 200, 301 und 304, sofern nicht `always` angegeben wird. Fehlerseiten werden sonst ohne diese Header ausgeliefert. Halten Sie gemeinsame Header in einer Datei, die auf jeder Ebene mit eigenen Headern per include eingebunden wird. Oder setzen Sie ab 1.29.3 `add_header_inherit merge;`. Ergänzen Sie `always`, wo Fehlerantworten die Header benötigen, und prüfen Sie jeden Pfad mit `curl -sI`.

Anfragen für eine Website beantwortet eine andere. nginx gleicht zuerst Adresse und Port ab, dann den Host-Header mit server_name. Ist der Name jedem server-Block an diesem Port unbekannt, geht die Anfrage an den Standardserver. Das ist der zuerst aufgeführte, sofern kein anderer mit default_server markiert ist. Ein Tippfehler im server_name einer neuen Website liefert unbemerkt die Seiten eines alten Nachbarn. Definieren Sie einen ausdrücklichen Standardserver, der unbekannte Namen mit `return 444` beantwortet und damit die Verbindung schließt. Ein Fehler zeigt sich dann als Fehler statt als falsche Website.

Uploads scheitern mit 413 Request Entity Too Large. client_max_body_size ist auf 1 MB voreingestellt. nginx weist einen größeren Anfragekörper zurück, bevor die Anwendung ihn sieht. Erhöhen Sie die Grenze nur in der location, die Uploads annimmt, statt für den ganzen Server, und stimmen Sie sie mit der eigenen Grenze der Anwendung ab.

Das Backend erzeugt Links zum falschen Host oder erhält den falschen Pfad. Ohne andere Vorgabe sendet nginx dem nachgelagerten Server einen Host-Header aus der Adresse in proxy_pass, etwa `127.0.0.1:8080`. Die Anwendung erzeugt dann Weiterleitungen und absolute Links zu dieser Adresse. Setzen Sie `proxy_set_header Host $host;` in der weiterleitenden location. Pfade ändern sich mit der Form von proxy_pass: Mit einer URI, selbst einem einzelnen `/`, ersetzt es den zur location passenden Teil des Pfads. Ohne URI reicht es den Pfad unverändert durch. Testen Sie deshalb nach jeder Änderung einen verschachtelten Pfad.

Clients erhalten weiter die alte Konfiguration oder das alte Zertifikat. Eine bearbeitete Datei oder ein über Nacht erneuertes Zertifikat ändert am laufenden nginx bis zum Neuladen nichts. Beim Neuladen einer defekten Konfiguration bleibt die alte bestehen. Testen Sie vor jedem Neuladen, lesen Sie danach das Fehlerlog und prüfen Sie, was Clients tatsächlich erhalten:

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

Weiterführende Lektüre

- nginx-Dokumentation — das Projektverzeichnis mit Anleitungen und Direktivenreferenzen zu den einzelnen Modulen.
- Einstieg — Master- und Workerprozesse, Neuladen und Aufbau der Konfiguration.
- Wie nginx eine Anfrage verarbeitet — die Wahl eines server-Blocks anhand von Adresse, Port und Host-Header.
- Modul ngx_http_core_module — Reihenfolge der location-Treffer, server_name und client_max_body_size.
- Modul ngx_http_proxy_module — proxy_pass, Header-Voreinstellungen, Zeitlimits und Antwortpufferung.
- Modul ngx_http_headers_module — add_header, seine Antwortcodes und seine Vererbungsregel.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Bereitstellungsweg, die Betriebsrisiken und die Bereitschaftsverantwortlichen. Wir können eine Änderung mit Prüfungen, einem Wiederherstellungsweg und klaren Zuständigkeiten abgrenzen.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist nginx?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

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

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/nginx/