Engineer

nginx

веб-сервер і зворотний проксі

Що це таке

nginx — це програма з відкритим кодом, яка приймає HTTP-з'єднання і на кожен запит або відповідає файлами з диска, або передає його іншому серверу й повертає клієнтові отриману відповідь. Назву вимовляють як «engine x», а власні сторінки проєкту пишуть її також NGINX. Автор програми — Ігор Сисоєв, і перша публічна версія, 0.1.0, вийшла 4 жовтня 2004 року. Користуватися нею можна безкоштовно за двопунктовою ліцензією BSD, а F5, Inc. продає корпоративні дистрибутиви, підтримку й навчання.

Проєкт перелічує шість ролей: HTTP-сервер, зворотний проксі, кеш вмісту, балансувальник навантаження, проксі для TCP і UDP та поштовий проксі. На вебхостингу nginx зазвичай виконує одну або обидві з двох робіт: віддає статичні файли з диска і стоїть перед сервером застосунку, де завершує TLS і передає кожен запит усередину.

Під час запуску nginx створює один головний процес (master) і кілька робочих (worker). Головний читає й перевіряє конфігурацію та наглядає за робочими. Робочі процеси працюють від імені непривілейованого користувача й виконують усю обробку запитів. Кожен із них використовує цикл подій, побудований на epoll у Linux і kqueue у FreeBSD, тож кілька робочих процесів можуть тримати відкритими тисячі з'єднань одночасно. Директива worker_processes задає їхню кількість — або фіксованим числом, або значенням auto, тобто по одному на ядро процесора, — а перелік можливостей проєкту оцінює 10 000 неактивних keep-alive-з'єднань приблизно у 2,5 МБ пам'яті.

Конфігурація — це текстовий файл із директив. Проста директива складається з назви й параметрів і закінчується крапкою з комою. Блокова директива обгортає інші директиви у фігурні дужки, а блок, що може містити

директиви, називають контекстом. Контексти events і http розташовані на верхньому рівні. Усередині http кожен блок server описує один віртуальний сервер, який обирають за адресою й портом, де він слухає, а тоді за іменем хоста, яке запитав клієнт. Усередині блоку server кожен блок location визначає, що робити з певною родиною шляхів запиту.

Застосуйте поняття до рішення

Застосуйте «Що таке nginx?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Щоденний догляд за серверами, поштою, обліковими записами й оновленнями, щоб те, на що всі покладаються, працювало тихо й без сюрпризів.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Підтримувала 10 серверів PostgreSQL і MS SQL — 10 серверів баз даних розгорнуто й підтримувано в справному стані

Підтримувала 10 серверів PostgreSQL і MS SQL Зв'язатися з нами →

Чому це важливо

Вебсервер — єдина програма, з якою стикається кожен відвідувач, тож його поведінка під навантаженням і його типові налаштування стосуються всіх. На користь nginx говорять дві властивості.

Перша — те, як дешево він тримає повільні з’єднання. Буферизацію відповідей від сервера за проксі ввімкнено за замовчуванням: nginx забирає відповідь бекенда так швидко, як бекенд її видає, записує її в тимчасовий файл, коли вона не вміщається в пам’ять, а тоді віддає клієнтові в тому темпі, який клієнт витримує. Сервер застосунку вже вільний для наступного запиту, поки телефон зі слабким сигналом ще завантажує попередню відповідь.

Друга — зміни конфігурації набирають чинності без утрати запитів. Відредагована конфігурація діє лише після перезавантаження (reload) або перезапуску. Під час reload головний процес перевіряє нову конфігурацію й пробує її застосувати. Якщо це вдається, стартують нові робочі процеси з нею, а старі перестають приймати з’єднання, завершують запити, які вже обробляють, і виходять. Якщо ні, головний процес відкочує зміни й далі обслуговує зі старою конфігурацією. Проєкт також згадує оновлення самого виконуваного файлу без переривання клієнтів.

Платити за це доводиться мовою конфігурації: саме її правила пріоритету й успадкування ховають більшість несподіванок. Для шляху запиту nginx спершу знаходить location із найдовшим збіжним префіксом і запам’ятовує

його, потім перебирає location із регулярними виразами в тому порядку, в якому вони стоять у файлі, і бере перший, що збігся. Запам'ятований префікс лишається запасним варіантом на випадок, коли регулярні вирази не дали результату. Location, позначений `=`, вимагає точного збігу шляху й одразу завершує пошук, а `^~` на найдовшому префіксі велить nginx пропустити регулярні вирази. Деякі директиви, серед них add_header і proxy_set_header, успадковуються з охопного рівня лише тоді, коли поточний рівень не задає власних.

Сертифікати довго надходили від ACME-клієнта, що працював поруч із nginx. Тепер проєкт пропонує також модуль ngx_http_acme_module, який постачають окремо від основної збірки.

Як це виглядає на практиці

Дві роботи з хостингу, виконані засновницею на попередніх посадах, стосуються nginx; в обох він ділив сервери з Apache, а не замінював його.

Перша — сорок клієнтських вебсайтів, які мали бути швидкими й завжди доступними. Вони стояли на серверах з Ubuntu Linux, де використовували і Apache, і nginx. Робота полягала в тому, щоб писати їхню конфігурацію, налаштовувати швидкодію й доглядати за ними, поки ними користувалися справжні відвідувачі, і всі сорок лишалися високодоступними. Два вебсервери в одному парку означають кожне правило двома діалектами: віртуальний хост Apache — це блок server у nginx, і виправлення, знайдене для одного, треба перекласти для іншого.

Друга тривала з 2015 до 2022 року. Автоматизація на ACME взяла на себе сертифікати ста застосунків Docker: вона видавала кожен сертифікат, поновлювала його ще до кінця строку дії й установлювала на хости з Ubuntu без участі людини, і на тих хостах так само працювали і Apache, і nginx. Для nginx оновлений файл на диску — лише половина такої зміни. Сертифікат, указаний звичайним шляхом, nginx читає, коли завантажує конфігурацію, а документація зазначає, що лише ім'я файлу, складене зі змінних, змушує його завантажувати сертифікат заново під час кожного TLS-рукостискання. Тож поновлення, яке записує нові файли, доходить до відвідувачів лише після reload, і будь-яке автоматичне поновлення для nginx мусить ним завершуватися.

Де виникають проблеми

Заголовок, заданий для всього сайту, зникає на одному шляху. Location, що має хоч один власний add_header, не успадковує заголовків із рівнів server чи http, тож один заголовок кешування, доданий до окремого location, непомітно скасовує всі заголовки безпеки над ним. proxy_set_header підкоряється тому самому правилу. До того ж add_header діє лише для кодів успіху й перенаправлення, як-от 200, 301 і 304, якщо не додати `always`, тож сторінки помилок ідуть без цих додаткових заголовків. Тримайте спільні заголовки в одному файлі, який підключає include на кожному рівні, що задає власні, або з версії 1.29.3 задайте `add_header_inherit merge;`, додайте `always` там, де вони потрібні відповідям із помилками, і перевірте кожен шлях через `curl -sI`.

На запити до одного сайту відповідає інший. nginx спершу зіставляє адресу й порт, а тоді заголовок Host зі server_name. Коли ім'я невідоме всім блокам server на цьому порту, запит отримує сервер за замовчуванням — перший у списку, якщо інший не позначено default_server. Друкарська помилка в server_name нового сайту тихо віддає сторінки старого сусіда. Оголосіть явний сервер за замовчуванням, який на невідомі імена відповідає `return 444`, тобто закриває з'єднання, щоб помилка виявлялася як помилка, а не як чужий сайт.

Завантаження файлів падають з 413 Request Entity Too Large. client_max_body_size за замовчуванням дорівнює 1 МБ, і nginx відхиляє більше тіло запиту ще до того, як його побачить застосунок. Підніміть ліміт у тому location, що приймає завантаження, а не для всього сервера, і тримайте його узгодженим із власним лімітом застосунку.

Бекенд будує посилання на хибний хост або отримує хибний шлях. Якщо не вказати інакше, nginx надсилає серверу за проксі заголовок Host, складений з адреси в proxy_pass, наприклад `127.0.0.1:8080`, тож застосунок генерує перенаправлення й абсолютні посилання на цю адресу. Задайте `proxy_set_header Host $host;` у location, що проксіює запити. Шляхи залежать від форми proxy_pass: записаний з URI, навіть з одним `/`, він замінює ту частину шляху, що збіглася з location, а без URI передає шлях без змін. Обирайте форму свідомо й перевірте вкладений шлях.

Клієнти й далі отримують стару конфігурацію або старий сертифікат. Відредагований файл чи сертифікат, поновлений уночі, не впливає на запущений nginx до reload, а reload зламаної конфігурації лишає на місці стару. Перевіряйте конфігурацію перед кожним reload, тоді читайте журнал помилок і дивіться, що клієнти отримують насправді:

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

Що почитати далі

- nginx documentation — покажчик проєкту з посібниками й довідниками директив для кожного модуля.
- Beginner's Guide — головний і робочі процеси, перезавантаження та будова конфігурації.
- How nginx processes a request — як блок server обирають за адресою, портом і заголовком Host.
- Module ngx_http_core_module — порядок зіставлення location, server_name і client_max_body_size.
- Module ngx_http_proxy_module — proxy_pass, типові заголовки, тайм-аути й буферизація відповідей.
- Module ngx_http_headers_module — add_header, коди відповідей, для яких він діє, і його правило успадкування.

Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке nginx?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

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

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/nginx/