Engineer

nginx

en webbserver och omvänd proxy

Vad det är

nginx är ett program med öppen källkod som tar emot HTTP-anslutningar och för varje förfrågan antingen svarar med filer från disken eller lämnar den vidare till en annan server och skickar tillbaka svaret. Namnet uttalas ”engine x”, och projektets egna sidor skriver det också NGINX. Igor Sysoev skrev det, och den första publika versionen, 0.1.0, kom den 4 oktober 2004. Det är fritt att använda under en BSD-licens med två klausuler, och F5, Inc. säljer företagsdistributioner, support och utbildning.

Projektet räknar upp sex roller: HTTP-server, omvänd proxy, innehållscache, lastbalanserare, proxy för TCP och UDP, och e-postproxy. På en webbserver har det oftast ett av två jobb, eller båda: att leverera statiska filer från disken och att stå framför en applikationsserver, där det terminerar TLS och skickar varje förfrågan vidare inåt.

När nginx startar kör det en masterprocess och flera arbetsprocesser (”worker processes”). Masterprocessen läser och kontrollerar konfigurationen och tar hand om arbetsprocesserna. Arbetsprocesserna körs som en användare utan privilegier och sköter all hantering av förfrågningar. Var och en använder en händelsebaserad loop, byggd på epoll i Linux och kqueue i FreeBSD, så att en handfull arbetsprocesser kan hålla tusentals anslutningar öppna samtidigt. Direktivet worker_processes anger hur många de är, antingen som ett fast antal eller som auto för en per processorkärna, och projektets lista över funktioner anger att 10 000 inaktiva keep-alive-anslutningar tar ungefär 2,5 MB minne.

Konfigurationen är en textfil med direktiv. Ett enkelt direktiv är ett namn och dess parametrar och slutar med semikolon. Ett blockdirektiv omsluter andra direktiv med klammerparenteser, och ett block som kan innehålla direktiv kallas en kontext (”context”). Kontexterna events och http ligger på översta nivån. Inne

i http beskriver varje server-block en virtuell server, som väljs efter den adress och port den lyssnar på och sedan efter det värdnamn klienten frågade efter. Inne i ett server-block säger varje location-block vad som ska göras med en familj av sökvägar i förfrågningar.

Använd begreppet i ett beslut

Använd ”Vad är nginx?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Daglig skötsel av servrar, e-post, konton och uppdateringar, så att det som alla förlitar sig på förblir odramatiskt.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Drev 10 PostgreSQL- och MS SQL-servrar på Ubuntu — 10 databasservrar som driftsattes och hölls i gott skick

Drev 10 PostgreSQL- och MS SQL-servrar på Ubuntu Hör av er →

Varför det spelar roll

Webbservern är det enda program som varje besökare rör, så dess beteende under belastning och dess standardvärden når alla. Två egenskaper talar för nginx.

Den första är hur billigt det bär långsamma anslutningar. Buffring av svar från en server bakom proxyn är påslagen som standard: nginx tar emot svaret från backend så fort backend kan ta fram det, skriver över till en temporär fil när det växer ur minnet och matar sedan ut det till klienten i den takt klienten klarar. Applikationsservern är fri för nästa förfrågan medan en telefon med svag signal fortfarande laddar ned den förra.

Den andra är att ändringar i konfigurationen går ut utan att förfrågningar tappas. Redigeringar i konfigurationen får verkan först vid en omladdning eller en omstart. Vid en omladdning kontrollerar masterprocessen den nya konfigurationen och försöker tillämpa den. Om det går startar nya arbetsprocesser med den, medan de gamla slutar ta emot anslutningar, gör klart de förfrågningar de har och avslutas. Om det misslyckas rullar masterprocessen tillbaka och fortsätter med den konfiguration den redan hade. Projektet räknar också upp att själva binären kan uppgraderas utan att klienterna avbryts.

Priset betalas i konfigurationsspråket, vars regler för företräde och arv är där de flesta överraskningarna bor. För en sökväg letar nginx först upp den location som har längst matchande prefix och kommer ihåg den, och provar sedan de location som bygger på reguljära uttryck i den ordning de står i filen och tar den första

som matchar. Det ihågkomna prefixet används när alla reguljära uttryck misslyckas. En location märkt `=` matchar sökvägen exakt och avslutar sökningen direkt, och `^~` på det längsta prefixet säger åt nginx att hoppa över de reguljära uttrycken. Vissa direktiv, bl.a. add_header och proxy_set_header, ärvs från nivån utanför bara om den aktuella nivån saknar egna.

Certifikat kom länge från en ACME-klient som kördes bredvid nginx. Projektet erbjuder nu också ngx_http_acme_module, som levereras separat från kärnbygget.

Så ser det ut i praktiken

Två delar av hostingarbetet i grundarens tidigare roller rör nginx, och i båda delade det servrarna med Apache i stället för att ersätta det.

Den första var fyrtio kundwebbplatser som måste vara snabba och alltid nåbara, som kördes på servrar med Ubuntu Linux där både Apache och nginx användes. Jobbet var att skriva deras konfiguration, trimma dem för fart och sköta dem medan verkliga besökare använde dem, och alla fyrtio förblev högt tillgängliga. Två webbservrar i samma park betyder varje regel på två dialekter: en virtuell värd i Apache är ett server-block i nginx, och en lösning som hittats för den ena måste översättas för den andra.

Den andra pågick från 2015 till 2022. Ett ACME-arbetsflöde tog över certifikaten för hundra Docker-applikationer: det utfärdade varje certifikat, förnyade det innan det gick ut och installerade det på värdarna med Ubuntu utan någon människa i kedjan, och även på de värdarna användes både Apache och nginx. På nginx-sidan är en förnyad fil på disken bara halva en sådan ändring. Ett certifikat som anges med en vanlig sökväg läses när nginx läser in sin konfiguration, och dokumentationen noterar att bara ett filnamn byggt av variabler får det att läsa in certifikatet igen vid varje TLS-handskakning. En förnyelse som skriver nya filer når därför besökarna först efter en omladdning, så varje automatiserad förnyelse för nginx måste sluta med en.

Var det går fel

En header som satts för hela webbplatsen försvinner på en sökväg. En location som har någon egen add_header ärver inga från server- eller http-nivån, så en enda cacheheader som läggs till i en location tappar i tysthet varje säkerhetsheader ovanför. proxy_set_header följer samma regel. add_header

gäller dessutom bara för svarskoder för lyckade anrop och omdirigeringar, som 200, 301 och 304, om den inte får `always`, så felsidor går ut utan dem. Håll gemensamma headers i en fil som tas in med include på varje nivå som sätter egna, eller sätt `add_header_inherit merge;` från och med 1.29.3, lägg till `always` där felsvar behöver dem och kontrollera varje sökväg med `curl -sI`.

Förfrågningar till en webbplats besvaras av en annan. nginx matchar först adress och port och sedan Host-headern mot server_name. När namnet är okänt för varje server-block på den porten går förfrågan till standardservern, som är den första i listan om inte en annan är märkt default_server. Ett stavfel i server_name för en ny webbplats levererar i tysthet en gammal grannes sidor. Deklarera en uttrycklig standardserver som svarar på okända namn med `return 444`, som stänger anslutningen, så att ett misstag syns som ett fel i stället för som fel webbplats.

Uppladdningar misslyckas med 413 Request Entity Too Large. client_max_body_size har standardvärdet 1 MB, och nginx avvisar en större förfrågningskropp innan applikationen ser den. Höj gränsen i den location som tar emot uppladdningar i stället för för hela servern, och håll den i takt med applikationens egen gräns.

Backend bygger länkar till fel värd eller får fel sökväg. Om inget annat anges skickar nginx en Host-header till servern bakom proxyn som bildas av adressen i proxy_pass, t.ex. `127.0.0.1:8080`, så applikationen genererar omdirigeringar och absoluta länkar till den adressen. Sätt `proxy_set_header Host $host;` i den location som proxar. Sökvägarna ändras med formen på proxy_pass: skriven med en URI, även ett enda `/`, ersätter den den del av sökvägen som matchade location, medan den utan en URI skickar sökvägen vidare oförändrad, så testa en nästlad sökväg efter varje ändring.

Klienterna får fortfarande den gamla konfigurationen eller det gamla certifikatet. En redigerad fil, eller ett certifikat som förnyats under natten, har ingen verkan på en nginx som kör förrän vid en omladdning, och en omladdning av en trasig konfiguration lämnar den gamla på plats. Testa före varje omladdning, och läs sedan felloggen och kontrollera vad klienterna faktiskt får:

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

```
openssl x509 -noout -enddate
```

Läs mer

- nginx documentation — projektets index över guider och referenser för direktiven i varje modul.
- Beginner’s Guide — master- och arbetsprocesser, omladdning och hur konfigurationen är uppbyggd.
- How nginx processes a request — hur ett server-block väljs efter adress, port och Host-header.
- Module ngx_http_core_module — ordningen för att matcha location, server_name och client_max_body_size.
- Module ngx_http_proxy_module — proxy_pass, standardvärden för headers, tidsgränser och buffring av svar.
- Module ngx_http_headers_module — add_header, dess svarskoder och dess regel för arv.

Ordlista

Välj utifrån era faktiska förutsättningar

Ta med driftsättningsvägen, driftsriskerna och dem som har jour. Vi kan avgränsa en förändring med kontroller, en väg tillbaka och tydligt ansvar.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är nginx?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

welcome@engineer.company +45 50 10 61 84 Köpenhamn, Danmark

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/nginx/