Engineer

nginx

en webserver og omvendt proxy

Hva det er

nginx er et program med åpen kildekode som tar imot HTTP-forbindelser og, for hver forespørsel, enten svarer med filer fra disken eller sender forespørselen videre til en annen server og leverer svaret tilbake. Navnet uttales «engine x», og prosjektets egne sider skriver det også NGINX. Igor Sysoev skrev programmet, og den første offentlige versjonen, 0.1.0, kom 4. oktober 2004. Det er gratis å bruke under en BSD-lisens med to klausuler, og F5, Inc. selger bedriftsdistribusjoner, support og opplæring.

Prosjektet lister opp seks roller: HTTP-server, omvendt proxy, innholdscache, lastbalanserer, proxy for TCP og UDP, og e-postproxy. På en webvert har nginx som regel én av to jobber, eller begge: å levere statiske filer fra disken, og å stå foran en applikasjonsserver, der det terminerer TLS og sender hver forespørsel videre innover.

Når nginx starter, kjører det én hovedprosess (master) og flere arbeidsprosesser (workers). Hovedprosessen leser og kontrollerer konfigurasjonen og passer på arbeidsprosessene. Arbeidsprosessene kjører som en bruker uten privilegier og gjør alt arbeidet med forespørslene. Hver av dem bruker en hendelsesbasert løkke, bygd på epoll i Linux og kqueue i FreeBSD, og derfor kan en håndfull arbeidsprosesser holde tusenvis av forbindelser åpne samtidig. Direktivet worker_processes bestemmer hvor mange det er, enten som et fast tall eller som auto for én per prosessorkjerne, og ifølge prosjektets funksjonsliste bruker 10 000 inaktive keep-alive-forbindelser ca. 2,5 MB minne.

Konfigurasjonen er en tekstfil med direktiver. Et enkelt direktiv er et navn med parametere og slutter med semikolon. Et blokkdirektiv rammer inn andre direktiver i krøllparenteser, og en blokk som kan inneholde direktiver, kalles en kontekst. Kontekstene events og http ligger på øverste nivå. Inne i http beskriver

hver server-blokk én virtuell server, som velges først ut fra adressen og nettverksporten den lytter på, og deretter ut fra vertsnavnet klienten spurte etter. Inne i en server-blokk sier hver location-blokk hva som skal skje med en familie av forespørselsstier.

Bruk begrepet i en beslutning

Bruk «Hva er nginx?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Daglig ettersyn av servere, e-post, kontoer og oppdateringer, slik at det alle er avhengige av, forblir kjedelig.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Drev 10 PostgreSQL- og MS SQL-servere på Ubuntu — 10 databaseservere produksjonssatt og holdt i god stand

Drev 10 PostgreSQL- og MS SQL-servere på Ubuntu Ta kontakt →

Hvorfor det er viktig

Webserveren er det ene programmet alle besøkende er innom, så oppførselen under belastning og standardverdiene rammer alle. To egenskaper taler for nginx.

Den første er hvor billig nginx bærer trege forbindelser. Bufring av svar fra serveren bak proxyen er slått på som standard: nginx tar imot svaret fra backenden så fort backenden klarer å lage det, skriver det over i en midlertidig fil når det blir for stort for minnet, og gir det deretter videre til klienten i det tempoet klienten klarer. Applikasjonsserveren er ledig for neste forespørsel mens en mobil med svakt signal fremdeles laster ned den forrige.

Den andre er at endringer i konfigurasjonen rulles ut uten at forespørsler faller bort. En endret konfigurasjon virker først etter en omlasting (reload) eller en omstart. Ved en omlasting kontrollerer hovedprosessen den nye konfigurasjonen og prøver å ta den i bruk. Går det, starter nye arbeidsprosesser med den, mens de gamle slutter å ta imot forbindelser, gjør ferdig forespørslene de har, og avslutter. Går det ikke, ruller hovedprosessen tilbake og fortsetter med konfigurasjonen den allerede hadde. Prosjektet oppgir også at selve binærfilen kan oppgraderes uten at klientene blir avbrutt.

Prisen betales i konfigurasjonsspråket, der reglene for prioritet og arv er kilden til de fleste overraskelsene. For en sti finner nginx først den lengste prefiks-location som passer, og husker den. Så prøver det location-blokkene med regulære uttrykk i den rekkefølgen de står i filen, og tar den første som passer.

Det huskede prefikset brukes bare når alle de regulære uttrykkene bommer. En location merket `=` matcher stien eksakt og avslutter søket med en gang, og `^~` på det lengste prefikset ber nginx hoppe over de regulære uttrykkene. Noen direktiver, bl.a. add_header og proxy_set_header, arves fra nivået utenfor bare hvis det gjeldende nivået ikke definerer noen egne.

Sertifikater kom lenge fra en ACME-klient som kjørte ved siden av nginx. Prosjektet tilbyr nå også ngx_http_acme_module, som leveres separat fra kjernebygget.

Slik ser det ut i praksis

To hostingoppgaver fra grunnleggerens tidligere roller involverer nginx, og i begge delte det serverne med Apache i stedet for å erstatte det.

Den første var førti kundenettsteder som måtte være raske og alltid tilgjengelige, som kjørte på servere med Ubuntu Linux der både Apache og nginx var i bruk. Jobben var å skrive konfigurasjonen, justere dem for fart og holde dem ved like mens ekte besøkende brukte dem, og alle førti holdt høy tilgjengelighet. To webservere i én flåte betyr at hver regel finnes på to dialekter: en virtuell vert i Apache er en server-blokk i nginx, og en løsning dere finner for den ene, må oversettes til den andre.

Den andre gikk fra 2015 til 2022. En ACME-arbeidsflyt overtok sertifikatene for hundre Docker-applikasjoner, utstedte hvert av dem, fornyet det før det gikk ut og installerte det på vertene med Ubuntu uten en person i løkken, og også der var både Apache og nginx i bruk. På nginx-siden er en fornyet fil på disken bare halve endringen. Et sertifikat oppgitt med en vanlig sti leses når nginx laster konfigurasjonen, og dokumentasjonen påpeker at bare et filnavn bygd av variabler får nginx til å laste sertifikatet på nytt ved hvert TLS-håndtrykk. En fornyelse som skriver nye filer, når derfor de besøkende først etter en omlasting, og enhver automatisk fornyelse for nginx må avslutte med en.

Hvor det går galt

En header satt for hele nettstedet forsvinner på én sti. En location som har så mye som én add_header selv, arver ingen fra server- eller http-nivået, så én cache-header lagt til i en enkelt location fjerner i det stille alle sikkerhetsheaderne over den. proxy_set_header følger samme regel. add_header gjelder dessuten bare for vellykkede svar og omdirigeringer, som 200, 301 og

304, med mindre den får `always`, så feilsider går ut uten. Hold felles headere i én fil som hentes inn med include på hvert nivå som setter egne, eller sett `add_header_inherit merge;` fra 1.29.3, legg til `always` der feilsvar trenger headerne, og sjekk hver sti med `curl -sI`.

Forespørsler til ett nettsted besvares av et annet. nginx matcher adressen og nettverksporten først, og deretter Host-headeren mot server_name. Når navnet er ukjent for alle server-blokkene på den nettverksporten, går forespørselen til standardserveren, som er den første i listen med mindre en annen er merket default_server. En skrivefeil i server_name for et nytt nettsted serverer dermed i det stille sidene til en gammel nabo. Definer en eksplisitt standardserver som svarer ukjente navn med `return 444`, som lukker forbindelsen, så en tabbe viser seg som en feil og ikke som feil nettsted.

Opplastinger feiler med 413 Request Entity Too Large. client_max_body_size er 1 MB som standard, og nginx avviser en større forespørselskropp før applikasjonen får se den. Hev grensen i den location-blokken som tar imot opplastinger, ikke for hele serveren, og hold den i takt med applikasjonens egen grense.

Backenden lager lenker til feil vert eller får feil sti. Med mindre noe annet er satt, sender nginx serveren bak proxyen en Host-header laget av adressen i proxy_pass, f.eks. `127.0.0.1:8080`, og applikasjonen lager da omdirigeringer og absolutte lenker til den adressen. Sett `proxy_set_header Host $host;` i location-blokken som videresender. Stiene endrer seg med formen på proxy_pass: skrevet med en URI, selv bare `/`, erstatter den delen av stien som matchet location, mens skrevet uten sender den stien videre uendret. Test derfor en nøstet sti etter hver endring.

Klientene får fortsatt den gamle konfigurasjonen eller det gamle sertifikatet. En redigert fil, eller et sertifikat som ble fornyet i løpet av natten, har ingen virkning på en kjørende nginx før en omlasting, og en omlasting av en ødelagt konfigurasjon lar den gamle bli stående. Test før hver omlasting, les så feilloggen og sjekk hva klientene faktisk får:

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

Les mer

- nginx documentation — prosjektets oversikt over veiledninger og direktivreferanser for hver modul.
- Beginner's Guide — hovedprosess og arbeidsprosesser, omlasting og hvordan konfigurasjonen er bygd opp.
- How nginx processes a request — hvordan en server-blokk velges ut fra adresse, nettverksport og Host-header.
- Module ngx_http_core_module — rekkefølgen location-blokker matches i, server_name og client_max_body_size.
- Module ngx_http_proxy_module — proxy_pass, standardheadere, tidsavbrudd og bufring av svar.
- Module ngx_http_headers_module — add_header, hvilke svarkoder den gjelder for, og arveregelen.

Ordliste

Velg ut fra deres faktiske rammer

Ta med utrullingsløpet, driftsrisikoene og dem som har vakt. Vi kan avgrense en endring med kontroller, en vei tilbake og tydelig ansvar.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er nginx?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/nginx/