Engineer

nginx

Serveur web et proxy inverse

Ce que c’est

nginx est un programme open source qui accepte les connexions HTTP et, pour chaque requête, répond depuis les fichiers sur disque ou la transmet à un autre serveur avant d’en relayer la réponse. Le nom se prononce « engine x » et les pages du projet l’écrivent aussi NGINX. Igor Sysoev l’a écrit ; la première version publique, 0.1.0, est sortie le 4 octobre 2004. Son utilisation est libre sous licence BSD à 2 clauses ; F5, Inc. vend des distributions pour entreprises, de l’assistance et des formations.

Le projet énumère six rôles : serveur HTTP, proxy inverse, cache de contenu, équilibreur de charge, proxy TCP et UDP et proxy de messagerie. Sur un hôte web, il assure généralement l’une ou les deux tâches suivantes : servir des fichiers statiques sur disque et se placer devant un serveur applicatif pour assurer la terminaison TLS et transmettre chaque requête vers l’intérieur.

Au démarrage, nginx lance un processus maître unique et un certain nombre de processus de travail. Le maître lit et vérifie la configuration et s’occupe des processus de travail. Ceux-ci tournent sous un utilisateur non privilégié et traitent toutes les requêtes. Chacun utilise une boucle événementielle, fondée sur epoll sous Linux et kqueue sous FreeBSD : quelques processus peuvent ainsi garder des milliers de connexions ouvertes simultanément. La directive worker_processes fixe leur nombre, par valeur fixe ou avec auto pour un processus par cœur CPU ; la liste des fonctionnalités annonce environ 2,5 Mo de mémoire pour 10 000 connexions keep-alive inactives.

La configuration est un fichier texte de directives. Une directive simple est un nom et ses paramètres, terminés par un point-virgule. Une directive de bloc entoure d’autres directives d’accolades ; un bloc pouvant en contenir s’appelle un contexte. Les contextes events et http sont au premier niveau. Dans http, chaque

bloc server décrit un serveur virtuel, choisi par l’adresse et le port d’écoute, puis par le nom d’hôte demandé par le client. Dans un bloc server, chaque bloc location indique le traitement d’une famille de chemins de requête.

Utilisez le concept pour prendre une décision

Utilisez « Qu’est-ce que nginx ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. L'entretien quotidien des serveurs, de la messagerie, des comptes et des mises à jour, pour que les outils indispensables restent sans histoire.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. 10 serveurs PostgreSQL et MS SQL sur Ubuntu — 10 serveurs de bases de données déployés et maintenus

10 serveurs PostgreSQL et MS SQL sur Ubuntu Contactez-nous →

Pourquoi c’est important

Le serveur web est le programme avec lequel tous les visiteurs interagissent : son comportement sous charge et ses valeurs par défaut les concernent donc tous. Deux propriétés plaident pour nginx.

La première est le faible coût des connexions lentes. La mise en tampon des réponses d’un serveur en amont est activée par défaut : nginx récupère la réponse aussi vite que le backend la produit, utilise un fichier temporaire si elle dépasse la mémoire disponible, puis la transmet au rythme du client. Le serveur applicatif peut traiter la requête suivante pendant qu’un téléphone au signal faible télécharge encore la réponse précédente.

La seconde est l’application des changements de configuration sans abandon de requêtes. Les modifications ne prennent effet qu’au rechargement ou au redémarrage. Au rechargement, le maître vérifie la nouvelle configuration et tente de l’appliquer. En cas de succès, de nouveaux processus de travail démarrent avec elle, tandis que les anciens cessent d’accepter des connexions, terminent leurs requêtes et s’arrêtent. En cas d’échec, le maître revient en arrière et continue avec l’ancienne configuration. Le projet cite aussi la mise à niveau du binaire lui-même sans interrompre les clients.

Le prix se paie dans le langage de configuration, dont les règles de priorité et d’héritage réservent la plupart des surprises. Pour un chemin de requête, nginx trouve d’abord le bloc location au plus long préfixe correspondant et le mémorise, puis essaie les blocs location à expressions régulières dans l’ordre où ils apparaissent dans le fichier et retient le premier qui correspond. Le préfixe

mémorisé est utilisé si toutes les expressions échouent. Un bloc location marqué `=` correspond exactement au chemin et arrête aussitôt la recherche ; `^~` sur le plus long préfixe demande d'ignorer les expressions régulières. Certaines directives, dont add_header et proxy_set_header, ne sont héritées du niveau englobant que si le niveau actuel n'en définit aucune.

Pendant longtemps, les certificats provenaient d'un client ACME fonctionnant à côté de nginx. Le projet propose maintenant aussi ngx_http_acme_module, fourni séparément de la version compilée du cœur.

À quoi cela ressemble en pratique

Deux travaux d'hébergement issus des rôles antérieurs de la fondatrice font intervenir nginx ; dans les deux cas, il partageait les serveurs avec Apache au lieu de le remplacer.

Le premier concernait quarante sites clients devant rester rapides et toujours joignables, sur des serveurs Ubuntu Linux où Apache et nginx étaient utilisés. Le travail consistait à écrire leurs configurations, les régler pour gagner en rapidité et les entretenir pendant que de vrais visiteurs les utilisaient ; les quarante sont restés hautement disponibles. Deux serveurs web sur une flotte demandent chaque règle dans deux dialectes : un hôte virtuel Apache est un bloc server nginx et une correction trouvée pour l'un doit être traduite pour l'autre.

Le second s'est déroulé de 2015 à 2022. Un mécanisme ACME a pris en charge les certificats de cent applications Docker, en émettant chaque certificat, en le renouvelant avant expiration et en l'installant sur les hôtes Ubuntu sans intervention humaine ; Apache et nginx y étaient à nouveau tous deux utilisés. Côté nginx, renouveler un fichier sur disque n'est que la moitié du changement. Un certificat nommé par un chemin simple est lu au chargement de la configuration ; la documentation précise que seul un nom de fichier construit à partir de variables provoque sa relecture à chaque négociation TLS. Un renouvellement écrivant de nouveaux fichiers n'atteint donc les visiteurs qu'après rechargement : toute automatisation du renouvellement pour nginx doit finir par celui-ci.

Où les problèmes surviennent

Un en-tête défini pour tout le site disparaît sur un chemin. Un bloc location possédant son propre add_header n'en hérite aucun du niveau server ou http : ajouter un en-tête de cache à un seul bloc location supprime silencieusement tous les en-têtes de sécurité supérieurs. proxy_set_header suit la même règle. Par ailleurs, add_header ne s'applique qu'aux codes de succès ou de redirection comme 200, 301 et 304, sauf avec `always` : les pages d'erreur partent sans ces en-têtes. Gardez les en-têtes communs dans un fichier chargé avec include à chaque niveau qui définit les siens, ou, depuis la version 1.29.3, utilisez `add_header_inherit merge;` ; ajoutez `always` si les réponses d'erreur en ont besoin et vérifiez chaque chemin par `curl -sI` .

Un site répond aux requêtes destinées à un autre. nginx compare d'abord adresse et port, puis l'en-tête Host à server_name. Si aucun bloc server de ce port ne connaît le nom, la requête va au serveur par défaut, premier de la liste sauf si un autre porte default_server. Une faute dans le server_name d'un nouveau site sert discrètement les pages d'un ancien voisin. Déclarez un serveur par défaut explicite répondant aux noms inconnus par `return 444` , qui ferme la connexion : l'erreur devient visible au lieu de montrer le mauvais site.

Les téléversements échouent avec 413 Request Entity Too Large. client_max_body_size vaut 1 Mo par défaut et nginx refuse un corps de requête plus grand avant que l'application ne le voie. Relevez la limite dans le bloc location acceptant les téléversements plutôt que sur tout le serveur, et alignez-la sur celle de l'application.

Le backend crée des liens vers le mauvais hôte ou reçoit le mauvais chemin. Sans instruction contraire, nginx transmet au serveur en amont un en-tête Host construit à partir de l'adresse de proxy_pass, comme `127.0.0.1:8080` : l'application génère alors redirections et liens absolus vers celle-ci. Définissez `proxy_set_header Host $host;` dans le bloc location du proxy. La forme de proxy_pass change les chemins : avec une URI, même un simple `/` , il remplace la partie du chemin correspondant au bloc location ; sans URI, il transmet le chemin inchangé. Testez donc un chemin imbriqué après toute modification.

Les clients reçoivent encore l'ancienne configuration ou l'ancien certificat. Un fichier modifié ou un certificat renouvelé la nuit n'a aucun effet sur nginx en fonctionnement avant rechargement ; recharger une configuration cassée laisse

l'ancienne en place. Testez avant chaque rechargement, puis lisez le journal d'erreurs et vérifiez ce que les clients reçoivent réellement :

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

Pour aller plus loin

- Documentation nginx — index des guides et références des directives par module.
- Guide pour débuter — processus maître et de travail, rechargement et structure de la configuration.
- Traitement d'une requête par nginx — choix du bloc server par adresse, port et en-tête Host.
- Module ngx_http_core_module — ordre de correspondance des blocs location, server_name et client_max_body_size.
- Module ngx_http_proxy_module — proxy_pass, en-têtes par défaut, délais et mise en tampon des réponses.
- Module ngx_http_headers_module — add_header, codes de réponse et règle d'héritage.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le parcours de déploiement, les risques opérationnels et les personnes d'astreinte. Nous pouvons définir un changement avec des vérifications, une procédure de reprise et une responsabilité claire.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce que nginx ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

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

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/nginx/