Engineer

nginx

servidor web y proxy inverso

Qué es

nginx es un programa de código abierto que acepta conexiones HTTP y, para cada petición, responde con archivos del disco o la entrega a otro servidor y transmite su respuesta. El nombre se pronuncia «engine x», y las propias páginas del proyecto también lo escriben NGINX. Lo creó Igor Sysoev, y la primera versión pública, la 0.1.0, apareció el 4 de octubre de 2004. Se puede usar gratuitamente bajo una licencia BSD de 2 cláusulas, y F5, Inc. vende distribuciones empresariales, soporte y formación.

El proyecto enumera seis funciones: servidor HTTP, proxy inverso, caché de contenido, balanceador de carga, proxy TCP y UDP y proxy de correo. En un servidor web suele realizar una o ambas de estas tareas: servir archivos estáticos del disco y situarse delante de un servidor de aplicaciones, donde termina TLS y reenvía cada petición hacia el interior.

Al iniciarse, nginx ejecuta un único proceso maestro y varios procesos de trabajo. El maestro lee y comprueba la configuración y supervisa los procesos de trabajo. Estos se ejecutan con un usuario sin privilegios y se encargan de todas las peticiones. Cada uno utiliza un bucle basado en eventos, apoyado en epoll en Linux y kqueue en FreeBSD, de modo que unos pocos procesos pueden mantener miles de conexiones abiertas a la vez. La directiva worker_processes establece cuántos hay, mediante un número fijo o con auto para tener uno por núcleo de CPU, y la lista de características del proyecto cifra el consumo de 10.000 conexiones keep-alive inactivas en unos 2,5 MB de memoria.

La configuración es un archivo de texto con directivas. Una directiva simple consta de un nombre y sus parámetros y termina en punto y coma. Una directiva de bloque encierra otras directivas entre llaves, y un bloque que puede contener directivas se denomina contexto. Los contextos events y http están en el nivel

superior. Dentro de http, cada bloque server describe un servidor virtual, que se elige por la dirección y el puerto en los que escucha y después por el nombre de host que pidió el cliente. Dentro de un bloque server, cada bloque location indica qué hacer con un conjunto de rutas de petición.

Utilice el concepto en una decisión

Aplique «¿Qué es nginx?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. El cuidado diario de servidores, correo, cuentas y actualizaciones para que los servicios de los que depende todo el mundo sigan funcionando sin sorpresas.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. 10 servidores PostgreSQL y MS SQL en Ubuntu — 10 servidores de bases de datos desplegados y mantenidos

10 servidores PostgreSQL y MS SQL en Ubuntu Póngase en contacto →

Por qué importa

El servidor web es el programa por el que pasa cada visitante, así que su comportamiento bajo carga y sus valores predeterminados afectan a todos. Dos propiedades explican el interés de nginx.

La primera es el bajo coste con el que mantiene conexiones lentas. El almacenamiento en búfer de las respuestas de un servidor situado detrás del proxy está activado de forma predeterminada: nginx recibe la respuesta del backend tan rápido como este puede producirla, la vuelca a un archivo temporal cuando supera la capacidad de la memoria y después la entrega al cliente al ritmo que este pueda recibirla. El servidor de aplicaciones queda libre para la siguiente petición mientras un teléfono con poca cobertura aún descarga la respuesta anterior.

La segunda es que los cambios de configuración se aplican sin cortar peticiones. Las modificaciones solo entran en vigor al recargar o reiniciar. Al recargar, el maestro comprueba la nueva configuración e intenta aplicarla. Si lo consigue, arranca nuevos procesos de trabajo con ella, mientras los antiguos dejan de aceptar conexiones, completan las peticiones en curso y terminan su ejecución. Si falla, el maestro revierte el cambio y sigue sirviendo con la configuración que ya tenía. El proyecto también contempla actualizar el propio binario sin interrumpir a los clientes.

El coste se paga en el lenguaje de configuración, cuyas reglas de precedencia y herencia concentran la mayoría de las sorpresas. Para una ruta de petición, nginx busca primero el bloque location cuyo prefijo coincidente sea más largo y lo recuerda; después prueba los bloques location de expresiones regulares en el orden en que aparecen en el archivo y elige el primero que coincide. El prefijo recordado se usa cuando ninguna expresión regular coincide. Un bloque location marcado con `=` coincide exactamente con la ruta y termina la búsqueda de inmediato, y `^~` en el prefijo más largo indica a nginx que omita las expresiones regulares. Algunas directivas, entre ellas add_header y proxy_set_header, solo se heredan del nivel contenedor si el nivel actual no define ninguna propia.

Durante mucho tiempo, los certificados procedían de un cliente ACME que se ejecutaba junto a nginx. Ahora el proyecto también ofrece ngx_http_acme_module, distribuido por separado de la compilación del núcleo.

Cómo se aplica en la práctica

Dos trabajos de alojamiento realizados en puestos anteriores de la fundadora incluyeron nginx, y en ambos compartía servidores con Apache en lugar de sustituirlo.

El primero abarcaba cuarenta sitios web de clientes que debían ser rápidos y estar siempre accesibles, que funcionaban en servidores Ubuntu Linux con Apache y nginx. El trabajo consistía en escribir su configuración, ajustarlos para mejorar la velocidad y mantenerlos mientras recibían visitantes reales, y los cuarenta mantuvieron una alta disponibilidad. Dos servidores web en una misma flota implican expresar cada regla en dos dialectos: un servidor virtual de Apache equivale a un bloque server de nginx, y una corrección encontrada para uno debe traducirse al otro.

El segundo se desarrolló de 2015 a 2022. Un flujo ACME se encargó de los certificados de cien aplicaciones Docker: emitía cada uno, lo renovaba antes de su caducidad y lo instalaba en los servidores Ubuntu sin intervención humana; Apache y nginx volvían a coexistir en esos servidores. En nginx, renovar un archivo del disco es solo la mitad del cambio. Un certificado indicado mediante una ruta simple se lee cuando nginx carga su configuración, y la documentación señala que solo un nombre de archivo construido con variables hace que el certificado se vuelva a cargar en cada negociación TLS. Por tanto, una renovación que escribe archivos nuevos solo llega a los visitantes tras una recarga, así que cualquier renovación automatizada para nginx debe terminar con una.

Dónde surgen los problemas

Una cabecera configurada para todo el sitio desaparece en una ruta. Un bloque location que tenga alguna directiva add_header propia no hereda ninguna del nivel server o http, de modo que añadir una sola cabecera de caché a un bloque location elimina silenciosamente todas las cabeceras de seguridad del nivel superior. proxy_set_header sigue la misma regla. Además, add_header solo se aplica a códigos de éxito y redirección como 200, 301 y 304, salvo que se indique `always`, por lo que las páginas de error salen sin esas cabeceras. Mantenga las cabeceras compartidas en un archivo incorporado con include en cada nivel que defina las suyas, o, desde la versión 1.29.3, establezca `add_header_inherit merge;`; añada `always` donde las respuestas de error las necesiten y compruebe cada ruta con `curl -sI`.

Otro sitio responde a las peticiones del sitio esperado. nginx compara primero la dirección y el puerto y después la cabecera Host con server_name. Cuando ningún bloque server de ese puerto reconoce el nombre, la petición llega al servidor predeterminado, que es el primero de la lista salvo que otro esté marcado como default_server. Una errata en server_name de un sitio nuevo sirve silenciosamente las páginas de un vecino antiguo. Declare un servidor predeterminado explícito que responda a nombres desconocidos con `return 444`, que cierra la conexión, para que el error se manifieste como tal en vez de mostrar el sitio equivocado.

Las subidas fallan con 413 Request Entity Too Large. client_max_body_size tiene un valor predeterminado de 1 MB, y nginx rechaza un cuerpo de petición mayor antes de que la aplicación lo vea. Aumente el límite en el bloque location que acepte subidas, en lugar de hacerlo para todo el servidor, y manténgalo alineado con el límite de la propia aplicación.

El backend construye enlaces al host equivocado o recibe una ruta incorrecta. Salvo que se indique lo contrario, nginx envía al servidor situado detrás del proxy una cabecera Host basada en la dirección de proxy_pass, como `127.0.0.1:8080`, por lo que la aplicación genera redirecciones y enlaces absolutos hacia esa dirección. Establezca `proxy_set_header Host $host;` en el bloque location que actúa como proxy. Las rutas cambian según la forma de proxy_pass: si se escribe con una URI, aunque solo sea `/`, sustituye la parte de la ruta que coincidió con el bloque location; sin URI, transmite la ruta sin cambios. Por eso, pruebe una ruta anidada después de cualquier cambio.

Los clientes siguen recibiendo la configuración o el certificado antiguos. Editar un archivo o renovar un certificado durante la noche no afecta a un nginx en ejecución hasta que se recarga, y recargar una configuración incorrecta deja la anterior en uso. Pruebe antes de cada recarga; después lea el registro de errores y compruebe lo que reciben realmente los clientes:

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

Lecturas adicionales

- Documentación de nginx — índice del proyecto con guías y referencias de directivas por módulo.
- Guía para principiantes — procesos maestro y de trabajo, recarga y estructura de la configuración.
- Cómo procesa nginx una petición — elección de un bloque server por dirección, puerto y cabecera Host.
- Módulo ngx_http_core_module — orden de coincidencia de location, server_name y client_max_body_size.
- Módulo ngx_http_proxy_module — proxy_pass, valores predeterminados de cabeceras, tiempos de espera y almacenamiento de respuestas en búfer.
- Módulo ngx_http_headers_module — add_header, sus códigos de respuesta y su regla de herencia.

Glosario

Elija según sus restricciones reales

Describa la ruta de despliegue, los riesgos operativos y quiénes se encargan de las guardias. Podemos definir el alcance de un cambio con comprobaciones, una vía de recuperación y responsabilidades claras.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es nginx?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

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

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

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

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/nginx/