Engineer

configuración

ajustes externos al código que determinan cómo se ejecuta

Qué es

La configuración es todo lo que un programa lee fuera de su propio código fuente para decidir cómo ejecutarse en un lugar concreto: el puerto en el que escucha, la base de datos a la que se conecta, las cuentas a las que deja entrar y las funciones que activa. En conversación casi siempre se abrevia como config, y los archivos que la contienen se llaman archivos de configuración. NIST, en SP 800-128, su guía de gestión de configuración centrada en la seguridad, la define como «las posibles condiciones, parámetros y especificaciones con los que puede describirse u organizarse un sistema de información o uno de sus componentes». La palabra clave es describirse: una configuración es una descripción, y el sistema puede coincidir con ella o no.

Un programa rara vez obtiene su configuración de un solo lugar. Hay valores predeterminados compilados en el binario, un archivo principal en disco con un directorio de fragmentos adicionales a su lado, y variables de entorno y opciones de línea de comandos que llegan al arrancar. El programa los combina en un conjunto de valores, con una regla para resolver qué ocurre cuando dos fuentes definen la misma clave, y cada programa tiene su propia regla. El servidor OpenSSH conserva el primer valor que lee para la mayoría de las palabras clave, mientras que otras herramientas permiten que los valores de un archivo posterior prevalezcan sobre los de otro anterior. Los mismos archivos leídos en otro orden producen otro sistema.

Los formatos varían: líneas de estilo INI, TOML, YAML, JSON y formatos propios de un programa. TOML 1.0.0 aspira a ser «un formato mínimo de archivo de configuración», mientras que YAML 1.2.2 es un lenguaje de datos de propósito general que utilizan los archivos de configuración, y cada uno tiene sus propias reglas para convertir texto en valores.

Utilice el concepto en una decisión

Aplique «¿Qué es la configuración?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Toda la configuración escrita en archivos para poder recrearla con exactitud y evitar que exista solo en la memoria de una persona.

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. 1.000 reglas RBAC para servidores Linux — 1,000 reglas de acceso escritas para cerrar una brecha de seguridad

1.000 reglas RBAC para servidores Linux Póngase en contacto →

Por qué importa

La configuración permite que una misma compilación sirva para muchos entornos. El mismo programa se ejecuta en un portátil, un servidor de preproducción y producción, y lo que cambia es su configuración. The Twelve-Factor App, una metodología para servicios desplegables, traza ahí la frontera: la configuración es «todo lo que probablemente varíe entre despliegues», como las referencias a servicios de apoyo, las credenciales de servicios externos y el nombre de host de cada despliegue. Exige separar estrictamente la configuración del código y propone una prueba aplicable en cualquier lugar: si «la base de código podría publicarse como código abierto en cualquier momento sin comprometer ninguna credencial».

El mismo texto delimita el otro extremo. La estructura interna que permanece igual en todos los despliegues, como la tabla de rutas de un framework web, queda fuera de su definición y «es mejor expresarla en el código». Un valor idéntico en todos los despliegues es código disfrazado de ajuste, y llevarlo a un archivo solo añade otro lugar donde puede haber un error.

La configuración es también donde un sistema declara su seguridad: quién puede iniciar sesión, qué puertos responden y qué deja pasar un cortafuegos. El endurecimiento de un servidor depende exactamente de los valores que realmente está utilizando, y entre un archivo y esos valores están el analizador, las reglas de precedencia, los valores predeterminados compilados, la versión del programa y el entorno. Cualquiera de esos elementos puede cambiar el resultado sin que cambie el archivo, así que el hábito fiable es preguntar al programa qué está utilizando y considerar esa respuesta como la configuración.

Cómo se aplica en la práctica

Este sitio se configura en un archivo TOML, hugo.toml, que establece, entre muchas otras cosas, sus idiomas, taxonomías y montajes. Una tabla de hugo.toml muestra cuánto pueden decidir unas pocas líneas:

```
[frontmatter]
  date = ["date", ":git", ":fileModTime"]
  lastmod = ["lastmod", ":git", ":fileModTime"]
```

Cada lista es un orden de búsqueda: primero una fecha escrita en la propia página, después la fecha del último commit que modificó el archivo y, por último, la fecha y hora de modificación del archivo. Una opción cerca del principio, `enableGitInfo = true`, permite a Hugo leer el historial de Git. El comentario situado sobre la tabla explica la primera entrada: hay que nombrar el campo literal de los metadatos iniciales; de lo contrario, un `lastmod` escrito en una página se ignora sin aviso mientras el sitio se construye con normalidad.

Esa tabla es la corrección descrita en la reparación de un sitemap en el que la mayoría de las páginas compartía una marca de tiempo. El sitemap había indicado a los buscadores que 172 de las 176 páginas del sitio habían cambiado por última vez en el mismo instante: al usar como fecha la marca de tiempo de modificación del archivo, cada ejecución de la exportación de contenido, que vuelve a escribir todos los archivos, renovaba de golpe sus marcas de tiempo. Con ese orden de búsqueda, el historial de versiones pasó a ser el origen de la fecha. En aquella fase documentada de implementación, una nueva sincronización tras un mes de cambios de contenido modificó ocho archivos de 186.

En un servidor, la misma lección aparece en el endurecimiento de SSH con tres etapas de validación. Un rol de automatización llevaba meses escribiendo la configuración del demonio SSH cuando, en septiembre de 2026, un informe de acceso leyó los valores efectivos de configuración del demonio y encontró dos que discrepaban del archivo del rol. Los valores predeterminados del sistema operativo se leían antes que el fragmento adicional y, para esos dos ajustes, prevalece la primera aparición, así que la solución fue un prefijo de nombre de archivo que se ordenara antes que el del proveedor. Junto con esa corrección, el rol valida ahora en tres etapas: se comprueba la sintaxis del archivo candidato antes de instalarlo, se comprueba la configuración combinada después y se lee la salida con los valores efectivos para verificar 24 directivas. Dos se excluyeron deliberadamente porque el demonio ya no las implementa.

Tres protecciones contra ataques de fuerza bruta a SSH que nunca funcionaron muestra el fallo desde el otro lado. En agosto de 2026, una revisión comparó la configuración de cada protección con el servidor en ejecución. Las tres estaban escritas, las tres aparecían en todos los informes y ninguna había cumplido su función desde que se instaló el servidor. La configuración de la jail todavía indicaba el puerto 22, un valor que quedó atrás cuando el endurecimiento inicial trasladó el demonio al puerto 1986. La línea del límite de frecuencia era correcta, pero el orden de las reglas del cortafuegos la anulaba: una regla más amplia situada encima capturaba primero el mismo tráfico. La configuración de la acción de bloqueo llamaba a un binario de filtrado de paquetes que el sistema de paquetes recomienda sin exigirlo, y en este servidor nunca se encontraba ese programa. De la revisión salieron dos reglas: cada control de seguridad se respalda con una aserción que lo prueba, nunca con un comentario que lo describe; y un cortafuegos se comprueba por la posición de cada regla en el orden, no por el mero hecho de que figure en la lista.

Dónde surgen los problemas

En todos los fallos siguientes, el texto del archivo es correcto, pero el comportamiento del sistema no; cada caso cuenta con una comprobación que observa ese comportamiento.

- Prevalece la fuente equivocada. El manual de configuración del servidor OpenSSH dice: «Salvo que se indique lo contrario, se utilizará el primer valor obtenido para cada palabra clave». Por tanto, un valor del proveedor leído antes que el fragmento adicional prevalece sobre este, y una herramienta en la que gana el último valor invierte el efecto de la misma disposición. Lea la regla de precedencia en el manual del propio programa y después pregúntele: `sshd -t` comprueba si la configuración es válida y `sshd -T` imprime la efectiva.
- Una clave TOML acaba en la tabla equivocada. En TOML 1.0.0, los pares situados bajo un encabezado `[table]` pertenecen a esa tabla «hasta el siguiente encabezado o el final del archivo», y la sangría se ignora. Una clave añadida al final del archivo pertenece a la última tabla aunque se escriba pegada al margen izquierdo. Mantenga las tablas al final y las claves del nivel superior encima. Definir una clave o tabla dos veces es inválido, así que un duplicado es un error que debe corregirse, nunca una forma de sobrescribir.

- Un valor YAML sin comillas cambia de tipo. Con el esquema básico, que YAML 1.2.2 recomienda como predeterminado, un valor sin comillas que coincide con uno de sus patrones se convierte en número, booleano o null: `1.10` es el número de coma flotante 1,1, `0o17` es el entero 15, `0123` se devuelve como 123, `FALSE` es un booleano y `~` es null. Entrecomille cada valor que deba seguir siendo texto, empezando por las cadenas de versión y los identificadores con ceros iniciales. La especificación también incluye las claves repetidas de un mapeo entre las causas de una carga incompleta, por lo que escribir una clave dos veces es un defecto.
- Un ajuste parece activo y no hace nada. Una regla sin efecto por la precedencia de otra anterior, una directiva eliminada en una versión nueva o un puerto que el servicio ya no utiliza permanecen en el archivo y parecen deliberados. Tanto la jail del puerto 22 como el límite bajo una regla más amplia aparecían como correctos. Una comprobación que lee cada archivo por separado los aprueba, así que verifique el orden y la presencia frente al sistema en ejecución.
- Un secreto viaja con el archivo. Una contraseña en un archivo de configuración va donde vaya el archivo: al repositorio, a las copias de seguridad y al informe de error. Twelve-Factor mantiene la configuración en variables de entorno en parte porque, a diferencia de los archivos de configuración, «hay pocas posibilidades de que se incorporen accidentalmente al repositorio de código».

Lecturas adicionales

- Glosario de NIST CSRC: configuración — las definiciones del término en las publicaciones de NIST, con el documento de origen de cada una.
- TOML v1.0.0 — la especificación completa, lo bastante breve para leerla de una sentada.
- YAML Ain't Markup Language, versión 1.2.2 — la especificación; su capítulo 10 contiene el esquema básico que decide en qué se convierte un valor sin comillas.
- The Twelve-Factor App: configuración — los motivos para mantener fuera del código los valores propios de cada despliegue, con la prueba de publicación como código abierto.
- sshd_config(5) — el manual de configuración del servidor OpenSSH, con su regla del primer valor, Include y Match.
- sshd(8) — el manual del demonio, donde `-t` comprueba la configuración y `-T` imprime la efectiva.



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 la configuración?» 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/configuration/