Engineer

Forja

servicio para alojar código y colaborar en su desarrollo

Qué es

Una forja, también llamada forja de software, de código o Git, es el servidor que un equipo acuerda tratar como hogar de sus repositorios, junto con los servicios que ejecuta a su alrededor. Cada clon completo de un repositorio Git lleva todo el historial, por lo que, en principio, ninguna copia prevalece sobre otra. En la práctica, un equipo necesita un lugar al que enviar cambios, donde se decida quién puede leer y escribir y donde se converse sobre el código. Ese lugar es la forja.

Por debajo, una forja guarda repositorios bare, la forma que solo tiene historial y ningún archivo extraído al directorio de trabajo, y habla uno o varios protocolos de transferencia de Git. El libro de Git enumera cuatro: el local, para un directorio de la misma máquina; HTTP; SSH; y el propio de Git, un demonio en el puerto 9418 sin autenticación alguna. Todo lo demás se construye encima y su alcance varía mucho.

La documentación de Forgejo describe el extremo más completo: seguimiento de incidencias, pull requests, wiki integrada, registro de paquetes para muchos ecosistemas de lenguajes, Forgejo Actions para ejecutar trabajos, réplicas de repositorios, migración desde otros servicios, protección de ramas y etiquetas y webhooks. Soft Serve está en el otro extremo. Se describe como un «servidor Git autoalojable para la línea de comandos»: un único binario llamado `soft` que sirve repositorios por SSH, HTTP o el protocolo Git, controla el acceso con claves públicas SSH y admite colaboradores, repositorios privados y webhooks. Ambos son forjas; difieren en cuánto trabajo del equipo intentan albergar.

Las forjas también han empezado a hablar entre sí. ForgeFed, un protocolo que federa plataformas de alojamiento y colaboración en proyectos de software, construido como extensión de ActivityPub, modela repositorios, gestores de

incidencias, gestores de parches, proyectos y equipos como actores que se envían actividades. Aspira a un lenguaje común entre redes de forjas y herramientas de gestión de proyectos, para que alguien de un servidor pueda abrir una incidencia en un repositorio de otro y un repositorio pueda cambiar de servidor dejando la copia antigua archivada en solo lectura y apuntando al nuevo hogar.

Utilice el concepto en una decisión

Aplique «¿Qué es una forja?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Cambios probados, publicados y reversibles, para que lleguen a sus clientes un martes por la tarde en vez de un viernes por la noche.

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. Despliegue GIS SaaS con GitHub Actions — despliegue, datos e informes automatizados

Despliegue GIS SaaS con GitHub Actions Póngase en contacto →

Por qué importa

La forja permite hacer cumplir las políticas. Una regla en la máquina de un desarrollador solo se aplica mientras esté bien configurada; un hook del servidor se ejecuta en cada push. El hook `pre-receive` lee cada referencia enviada como identificador antiguo, identificador nuevo y nombre por la entrada estándar. Si termina con estado distinto de cero, no se actualiza ninguna referencia. La protección de ramas, las revisiones obligatorias y las comprobaciones de commits firmados en las forjas mayores refinan la misma idea: la copia compartida rechaza lo que el equipo acordó no aceptar.

Elegir forja también es elegir quién guarda el código. Un servicio alojado asume costes de funcionamiento y disponibilidad; a cambio, código, lista de acceso y todas las conversaciones quedan en hardware ajeno y bajo condiciones ajenas. El autoalojamiento invierte el intercambio. Ninguna opción es gratuita y la diferencia importa sobre todo a una empresa que explica a sus clientes dónde están sus datos.

Una forja también guarda datos que Git no tiene. Commits y etiquetas viajan con cada clon, mientras que incidencias, hilos de revisión, listas de acceso y hooks del servidor pertenecen solo a la forja. Esa separación decide qué debe cubrir una copia de seguridad y qué debe llevarse un traslado entre forjas.

Cómo se aplica en la práctica

Esta empresa opera su propia forja. Tres proyectos de este sitio muestran lo que hizo falta.

El código fuente de la empresa estaba antes en un servicio de alojamiento ajeno. El proyecto para desplegar su propia forja Git parte de que operar una forja implica operar una forja: autenticación, control de acceso, almacenamiento, copias de seguridad y una cara pública. Es Soft Serve, un solo binario supervisado por el sistema operativo, elegido por priorizar SSH y no tener interfaz web administrativa. Es privado de forma predeterminada: rechaza acceso anónimo y sin clave y cada repositorio declarado está marcado como privado. Su servicio SSH solo escucha en la interfaz de bucle local, en el puerto 23231, accesible desde fuera mediante un host de salto. Un multiplexor para juntar HTTPS y Git sobre SSH en un puerto público está escrito y apagado, porque un servicio que escucha sin usarse es superficie de ataque. El resultado se dice sin rodeos: la empresa aloja su código en su propio hardware.

El repositorio de este sitio también está allí. Sus instrucciones nombran la forja autoalojada, Soft Serve por SSH, como remoto canónico, llaman réplica heredada a la copia alojada anterior y dicen que no hay CI alojada: el control de calidad se ejecuta localmente y en hooks de Git.

Un hook del servidor debe probarse en el entorno que recibe un push real de Git, así que se ensayó con un repositorio bare y un push reales antes de confiar en él. El ensayo crea el repositorio, instala el hook, clona, hace commits y pushes y comprueba las rutas de aceptación y rechazo. Encontró dos defectos, ninguno visible en el archivo. El segundo es ilustrativo: Git exporta una variable con el directorio del repositorio al ejecutar un hook; la batería de pruebas que este inició la heredó y diecinueve pruebas fallaron con código correcto hasta limpiar el entorno en esa frontera. La página también señala el límite: el ensayo no genera actualizaciones forzadas ni eliminaciones de ramas, que también son pushes.

Todo comparte un host de un solo núcleo y 512 MB de memoria con los demás servicios públicos de la empresa: siete dominios TLS, el servicio onion, servidores Gemini y Gopher y bloqueo de intrusos. La automatización de las copias de seguridad está implementada por separado; eso es distinto de tener copias activas. Una forja con panel web habría sido otra aplicación residente dentro de

ese presupuesto, por lo que se descartaron esas plataformas al diseñarlo. Durante un día de muestreo registrado, la memoria disponible en ese host llegó a un mínimo de unos 310 MB de 464.

Dónde surgen los problemas

Algunas forjas conceden de forma predeterminada acceso de lectura a los visitantes anónimos. La documentación de Soft Serve dice que ofrece de entrada acceso de solo lectura a conexiones anónimas, gobernado por `anon-access` y `allow-keyless`. Una forja privada pone el primero en `no-access` y el segundo en false; merece la pena probarlo desde una máquina sin clave cargada. El demonio de Git es aún más tajante: según su libro, un repositorio servido está disponible para que cualquiera lo clone o no se sirve en absoluto.

Un hook que llama a Git en otro directorio actúa sobre el repositorio equivocado. Durante un push, Git ejecuta los hooks receptores dentro del repositorio bare y exporta `GIT_DIR` y variables relacionadas para que las órdenes del hook lo encuentren. Todo lo que inicia las hereda, incluida una ejecución de pruebas en otro checkout. El manual githooks indica que se limpien esas variables antes de trabajar en un repositorio ajeno. Un hook `pre-receive` mínimo que prueba cada commit enviado en un directorio desechable lo hace en un subshell:

```
while read -r _ newrev _; do
  git cat-file -e "$newrev^{commit}" 2>/dev/null || continue
  tree=$(mktemp -d)
  git archive "$newrev" | tar -x -C "$tree"
  (
    unset $(git rev-parse --local-env-vars)
    cd "$tree" && ./run-tests
  ) || exit 1
done
```

Una forja restaurada recupera su código, pero no sus reglas. Cualquier clon puede repoblar los repositorios de un servidor perdido; el libro de Git advierte de que los hooks del servidor y ajustes similares se quedan atrás. Incidencias, revisiones, claves y listas de acceso nunca estuvieron en los clones. Hacer una copia de seguridad de una forja exige copiar su directorio de datos y configuración, restaurar ambos en algún lugar y comprobar que se rechaza un push que el hook debería rechazar.

Cada servicio a la escucha que abre una forja es algo que defender. La configuración de ejemplo de Soft Serve tiene SSH en 23231, HTTP en 23232, el demonio Git en 9418 y un servidor de estadísticas en 23233. Un equipo que solo envía por SSH puede apagar los demás servicios o configurarlos para que solo escuchen en la interfaz de bucle local, de modo que los puertos abiertos se ajusten a lo que se utiliza.

Trasladarse a otra forja lleva el código y deja atrás las conversaciones. Un push de réplica copia todas las referencias bajo `refs/`, incluidas ramas y etiquetas, y nada más. Incidencias, hilos de revisión, etiquetas y listas de acceso requieren la herramienta de migración de la nueva forja o un protocolo como ForgeFed, que copia commits, etiquetas, incidencias, solicitudes de integración y comentarios, y deja el repositorio antiguo archivado y apuntando al nuevo.

Lecturas adicionales

- Especificación ForgeFed — El protocolo de federación para forjas: actores, actividades y el flujo de traslado que archiva el repositorio antiguo.
- Soft Serve — La documentación del proyecto para un servidor Git de un solo binario y centrado en SSH, incluidos `anon-access` y `allow-keyless`.
- Documentación de Forgejo — Guías de administración y uso de una forja autoalojada completa, desde instalación y copias hasta Actions y réplicas.
- githooks — Todos los hooks de Git, incluido `pre-receive` y el entorno que hereda un hook de push.
- Los protocolos — Las cuatro formas de transferir datos entre máquinas con Git y el coste de configurar y proteger cada una.
- ActivityPub — La recomendación del W3C que amplía ForgeFed.

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 una forja?» 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/forge/