Engineer

GitLab

plataforma Git que también se puede operar por cuenta propia

Qué es

GitLab es una aplicación única construida alrededor de repositorios Git: aloja el código y añade merge requests para revisarlo, incidencias para planificar y un sistema CI/CD integrado que compila, prueba y despliega desde el mismo lugar. Lo que lo distingue es la elección de dónde se ejecuta. El mismo producto se ofrece como GitLab.com, el servicio que opera GitLab; como GitLab Self-Managed, que se instala en servidores propios; y como GitLab Dedicated, una instancia de un solo cliente que GitLab aloja y mantiene en la región de AWS que ese cliente prefiera.

Por debajo hay Git normal. Cada proyecto es un repositorio Git que se puede clonar y al que se pueden enviar o del que se pueden recuperar cambios con las órdenes habituales; el historial de un clon es completo. Todo lo que GitLab añade, desde hilos de revisión e incidencias hasta resultados de las cadenas de entrega y permisos, lo guarda el propio GitLab fuera del repositorio y nada de ello viaja con un clon.

Merge request es el nombre que GitLab da a un cambio propuesto pendiente de fusionarse: la misma idea que otras plataformas llaman pull request. La documentación lo describe como un lugar central para revisar código, mantener la discusión y seguir el cambio. Una página muestra la descripción, los cambios de código con comentarios de revisión en las propias líneas, el estado de las cadenas CI/CD, los informes sobre si se puede fusionar y la lista de commits. Un merge request se puede vincular a una incidencia y cerrarla automáticamente al fusionarse. Con la configuración inicial, la rama predeterminada del proyecto, normalmente main, está protegida y solo quienes tienen el rol Maintainer o uno superior pueden fusionar en ella.

El sistema CI/CD se configura en un archivo llamado `.gitlab-ci.yml` en la raíz del proyecto. Declara las etapas, trabajos y scripts que forman una cadena de entrega: las etapas marcan el orden, como compilar, probar y después desplegar, y los trabajos son las tareas que se realizan en cada etapa. Los ejecuta GitLab Runner, un programa separado escrito en Go y distribuido como un único binario. Un runner se conecta a una instancia de GitLab, espera trabajos, ejecuta cada uno en la propia máquina, dentro de un contenedor Docker o en un servidor remoto por SSH y devuelve el resultado.

Utilice el concepto en una decisión

Aplique «¿Qué es GitLab?» 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 mayor parte de lo que hace un equipo entre escribir código y entregarlo puede ocurrir dentro de una instancia de GitLab: el código, la revisión, la cadena que lo prueba, la incidencia que lo pidió y el registro de quién aprobó qué. Una aplicación significa un conjunto de cuentas y permisos, un lugar donde buscar y enlaces que se mantienen entre una incidencia, el merge request que la atendió y la cadena que lo comprobó.

La opción de gestión propia es la razón por la que muchas organizaciones eligen GitLab. Si el código no puede salir de la infraestructura de la empresa por un contrato, un regulador o una simple política, un servicio alojado queda descartado y GitLab se puede instalar dentro de la red que debe retenerlo. La documentación lo trata como un caso de uso plenamente contemplado: publica arquitecturas de referencia para instalaciones de entre 1.000 y 50.000 usuarios, junto al paquete de Linux y un chart de Helm para Kubernetes.

Ese control tiene un precio, que se paga en trabajo de operación y no solo en licencias. Operar GitLab exige dimensionar un servidor, mantener en buen estado su base de datos PostgreSQL y su almacén Redis o Valkey, proporcionar

máquinas para los runners, hacer copias de seguridad que realmente se puedan restaurar y actualizar en el orden que exige el proyecto. Las opciones alojadas delegan ese trabajo en GitLab; la de gestión propia lo deja en sus manos.

Cómo se aplica en la práctica

El trabajo de la empresa ofrece ejemplos de ambas perspectivas sobre GitLab.

Durante varios años de ingeniería de datos, las prácticas de CI/CD basadas en GitHub y GitLab, con Bash y Python como base de la automatización, sustituyeron pasos manuales tanto del análisis de datos como del desarrollo de software. Se estandarizó cómo pasaba el trabajo del desarrollo a la entrega para que fuese igual cada vez, en lugar de reinventarlo por proyecto, y el análisis, tratado durante mucho tiempo como un mundo aparte y más manual, entró en la misma cadena que el desarrollo. La página registra unas 4.000 horas ahorradas y entregas más consistentes porque ya no se hacían de una manera ligeramente distinta cada vez.

El código propio de la empresa muestra el otro extremo del autoalojamiento. Al dejar un servicio de alojamiento ajeno, pasó a una forja propia de un solo binario, centrada en SSH y sin interfaz web administrativa, elegida bajo el principio de que cuanto menos interfaz haya, menos habrá que defender. Esa forja comparte un host de servicios de acceso público con 512 MB de memoria, unos 464 MB utilizables, con un servidor web para siete dominios, un servicio onion de Tor, un servidor Gemini, otro Gopher y bloqueo de intrusos. La automatización de copias de seguridad está implementada por separado; su implementación es distinta de tener copias activas.

Ninguna de las dos páginas menciona GitLab; la comparación que sigue es un cálculo propio de este artículo. GitLab documenta 8 GB de memoria como el mínimo con el que puede funcionar una instalación de un solo nodo: todo aquel host tiene una decimosexta parte de esa cantidad. Poder operar GitLab implica tener un servidor dimensionado para GitLab; una forja pequeña y dedicada a un solo fin es la otra manera de mantener código en máquinas propias.

Un archivo mínimo muestra la forma de la configuración CI/CD, con dos etapas que se ejecutan en orden y un trabajo en cada una:

```
stages:
    - build
    - test

build-site:
    stage: build
    script:
        - make build

run-tests:
    stage: test
    script:
        - make test
```

Dónde surgen los problemas

- **Un servidor pequeño ralentiza toda la instancia.** La documentación de GitLab da 16 GB de memoria como base para una instalación de un nodo y 8 GB como mínimo en un entorno con memoria limitada. También advierte de que, bajo carga, la memoria de intercambio puede degradar considerablemente el rendimiento y pide desactivarla cuando sea posible o disponer de memoria suficiente para que GitLab nunca la use.
- **Una copia se restaura en una instancia que no puede leer sus propios datos.** La orden de copia de seguridad excluye el directorio de configuración `/etc/gitlab` y, con él, `gitlab-secrets.json`, las claves que descifran los valores cifrados de la base de datos. Sin ese archivo, esos valores siguen bloqueados tras restaurar la base. La documentación pide copiar la configuración por separado, como mínimo `gitlab-secrets.json` y `gitlab.rb`.
- **Una actualización se salta una parada obligatoria.** Las rutas de actualización incluyen versiones que deben instalarse antes de cualquier otra posterior; desde GitLab 17.5, esas paradas están en x.2, x.5, x.8 y x.11. Las migraciones en segundo plano de cada parada deben terminar antes de iniciar la siguiente actualización, de modo que una instancia atrasada varias versiones se pone al día parada por parada.
- **Un almacén gestionado resulta no estar soportado.** PostgreSQL es la única base de datos admitida. Redis o Valkey guarda sesiones y tareas en segundo plano y debe ser una instancia independiente: no se admiten Redis Cluster ni variantes sin servidor de Redis y Valkey. Se permite un servicio de nube gestionado siempre que sea de los tipos independientes.
- **Las cadenas esperan máquinas que nadie planificó.** GitLab.com ofrece runners preparados, pero en una instancia de gestión propia el administrador proporciona y gestiona la infraestructura donde se ejecutan los trabajos CI/CD, incluida la capacidad suficiente de los runners para la carga de la organización. Un equipo ocupado con un solo runner pequeño hace cola detrás de él.

Lecturas adicionales

- Documentación de GitLab — La documentación completa de uso, instalación y administración de GitLab.
- Requisitos de instalación — Memoria, CPU, almacenamiento y versiones admitidas de PostgreSQL y Redis.
- Copias de seguridad de GitLab — Qué contiene una copia, qué deja fuera y cómo guardar los secretos junto a ella.
- Rutas de actualización — Las paradas obligatorias y por qué deben terminar las migraciones en segundo plano entre una y otra.
- Primeros pasos con GitLab CI/CD — El archivo `.gitlab-ci.yml` , las etapas, los trabajos y las cadenas de entrega.
- GitLab Runner — El programa que ejecuta los trabajos y sus formas de hacerlo.

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 GitLab?» 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/gitlab/