Engineer

copia de seguridad

una copia de datos frente a pérdidas y su creación

Qué es

Una copia de seguridad es aquello desde lo que se restauran los datos cuando el original se borra, se corrompe o se pierde junto con la máquina que lo contenía, por lo que debe residir donde esos sucesos no puedan alcanzarla. El glosario de NIST la define como una copia de archivos y programas cuya finalidad es facilitar la recuperación. El sustantivo inglés backup, plural backups, nombra las copias, y el verbo back up, escrito en dos palabras, nombra el acto: una tarea nocturna respalda la base de datos y deja una copia de seguridad.

Se confunden con ella otras cosas cercanas. Una réplica recibe cada cambio en cuestión de segundos, lo que protege frente a un disco averiado y copia un `DELETE` erróneo con la misma fidelidad. Una instantánea congela un volumen en un instante y solo se convierte en una copia de seguridad cuando se guarda donde no puedan seguirla los fallos del original. Una copia de seguridad se define por su historia y su separación: devuelve los datos tal como estaban en un momento anterior, desde un lugar que sobrevivió a lo que destruyó el original.

Las bases de datos lo hacen concreto. El manual de PostgreSQL describe tres enfoques fundamentalmente distintos. Un volcado SQL, realizado con `pg_dump`, escribe las sentencias que reconstruyen la base de datos como una instantánea coherente del momento en que comenzó el volcado. Una copia de seguridad a nivel del sistema de archivos copia el propio directorio de datos. El archivado continuo combina una copia base de ese tipo con el registro de escritura anticipada, WAL, que produce el servidor después, y reproducir ese registro permite restaurar la base de datos a cualquier instante desde la copia base. Esa capacidad se llama recuperación a un instante determinado.

Utilice el concepto en una decisión

Aplique «¿Qué es una copia de seguridad?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Copias de seguridad que se prueban en vez de darse por buenas y un plan escrito para la mañana en que un servidor no vuelva a encenderse.

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. 100 copias críticas automatizadas con Barman — 100 copias críticas automatizadas en las bases de datos

100 copias críticas automatizadas con Barman Póngase en contacto →

Por qué importa

Dos cifras convierten un deseo general de seguridad en un diseño. NIST define el objetivo de punto de recuperación, RPO, como el punto en el tiempo al que deben recuperarse los datos después de una interrupción: en la práctica, cuánto trabajo reciente puede permitirse perder el negocio. El objetivo de tiempo de recuperación, RTO, es cuánto tiempo puede permanecer un sistema en recuperación antes de que la interrupción perjudique la finalidad de la organización.

Las dos cifras afectan a partes distintas del diseño. El RPO depende de la frecuencia con la que se toma una copia: un volcado nocturno puede perder un día de escrituras, mientras que archivar WAL reduce la pérdida a los últimos momentos de registro que aún están en camino al archivo. El RTO depende de cuánto tarda una restauración. El manual de PostgreSQL señala que el servidor reproduce todos los segmentos WAL desde la última copia base, algo que puede llevar tiempo si esa copia es antigua, por lo que la frecuencia de las copias base es tanto una decisión de RTO como de almacenamiento.

Ambas cifras son afirmaciones sobre una restauración, y por eso una copia de seguridad se juzga restaurándola. Una tarea que termina con estado cero demuestra que se escribieron bytes en algún lugar. Para saber si reconstruyen un sistema funcional dentro del RTO, hay que restaurar de forma programada en un lugar distinto de producción y comparar lo recuperado.

La separación es la otra mitad. Una copia en la misma cuenta y bajo las mismas credenciales comparte los fallos a los que debería sobrevivir: una cuenta cerrada, una clave robada o un script ejecutado contra el destino equivocado. La pregunta útil para cualquier copia es qué suceso, por sí solo, la eliminaría junto con el original.

Cómo se aplica en la práctica

A la escala más pequeña, la implementación de copias de seguridad y pruebas de restauración de la empresa se diseñó para ampliar la recuperación más allá de una instantánea del proveedor. Un rol Ansible instala scripts restic y unidades systemd para realizar un volcado de la base de datos cuando esté configurado, crear instantáneas cifradas y deduplicadas, depurar según la retención y comprobar la integridad. Un flujo mensual separado restaura un archivo conocido a una ubicación temporal y compara sus bytes. La copia de seguridad y la restauración tienen señales de éxito separadas, y los temporizadores requieren un repositorio configurado. La contraseña procede de un archivo protegido, en lugar de un valor de entorno interpretado por systemd. Son controles implementados y requisitos previos a la activación: el registro de implementación del 2026-10-27 indica que los temporizadores estaban desactivados al no haber repositorio y no documenta ninguna restauración real completada. Respalda el trabajo de automatización, no una afirmación de que la recuperación haya tenido éxito o de que ya se pueda perder el host sin riesgo.

A mayor escala, se automatizaron cien copias de datos críticos con Barman y Google Cloud para un producto que depende de sus datos SIG. Barman se ocupaba de PostgreSQL, Google Cloud almacenaba las copias fuera de las instalaciones y Bash y Python orquestaban y validaban las ejecuciones. Las comprobaciones de integridad confirmaban que cada copia era válida y no simplemente que existía.

En un tercer producto, el trabajo consistió en definir qué se podía recuperar en vez de darlo por supuesto. La retención de copias de seguridad reside en la plantilla Bicep junto a la base de datos, de modo que restaurar a un instante determinado es una propiedad de la plantilla y no un ajuste que alguien seleccionó una vez en un portal; el entorno que la rodea también es código. La auditoría dejó las lagunas por escrito: la georredundancia está desactivada y el RTO es una propuesta, no una medición, con el simulacro que lo mediría descrito y pendiente de ejecutar.

Dónde surgen los problemas

Los problemas suelen ocultarse hasta restaurar; conviene buscarlos en un día tranquilo.

Una copia de archivos de una base de datos activa se restaura como un clúster averiado. Copiar con `tar` el directorio de datos de PostgreSQL mientras funciona el servidor reúne archivos de momentos distintos, y el manual establece que el servidor debe detenerse para obtener una copia utilizable de esa manera. Una instantánea del sistema de archivos tomada en un único instante atómico sí funciona si incluye el WAL, que el servidor reproduce al arrancar como haría después de una caída; con los datos y el WAL en sistemas de archivos distintos, una instantánea simultánea puede resultar imposible. `pg_dump` o una copia base para archivado continuo ofrecen una copia coherente mientras el servidor sigue funcionando.

Una laguna en el archivo WAL corta en ese punto la recuperación a un instante determinado. La recuperación necesita una secuencia ininterrumpida de archivos WAL archivados que se remonte al comienzo de la copia base. Cuando el comando de archivado sigue fallando, PostgreSQL continúa reintentándolo y el WAL se acumula en `pg_wal/`, por lo que el primer síntoma suele ser un disco que se va llenando. El manual aconseja configurar y probar el archivado antes de la primera copia base, y vigilar que no se retrase.

La tarea se detiene y el silencio parece éxito. Una tarea de temporizador que falla escribe su error en un registro que nadie lee, y cada día silencioso pasa por uno bueno. La monitorización debe tratar el silencio como alarma: una señal enviada solo después de una ejecución correcta a un servicio que avisa cuando se retrasa detecta tanto una tarea detenida como una que falla de forma evidente.

La clave se pierde o llega modificada. El cifrado hace que el acceso dependa de las claves. restic admite varias contraseñas para un repositorio; su advertencia de pérdida irrecuperable de datos se aplica cuando no queda ninguna clave utilizable del repositorio. La clave debe residir en un lugar que sobreviva a la máquina. En la configuración restic anterior, la modificación podría haberla provocado el propio supervisor: interpreta secuencias de escape en los valores de entorno, por lo que una contraseña con una barra inversa, entregada así, habría llegado a restic como bytes distintos de aquellos con los que se creó el repositorio y sin ninguna advertencia. Leerla desde un archivo cierra esa vía.

La retención elimina la copia que hacía falta. La corrupción suele descubrirse semanas después de producirse, y una política que conserve siete copias diarias habrá sustituido para entonces todas las copias sanas por copias dañadas. Una política escalonada conserva una historia larga a bajo coste, y `--dry-run` de restic muestra qué eliminaría una política antes de borrar nada:

```
restic forget --keep-daily 7 --keep-weekly 5 --keep-monthly 12 \
  --prune --dry-run
```

`forget` elimina instantáneas y `prune` elimina los datos que solo ellas referenciaban, por lo que, después de una depuración, las copias antiguas han desaparecido definitivamente.

La comprobación de integridad pasa y la restauración sigue fallando. De forma predeterminada, `restic check` verifica la estructura del repositorio sin leer todos los archivos pack, porque eso supondría descargarlos todos; `--read-data` lo lee todo y `--read-data-subset=1/5` lee una quinta parte, de modo que cinco ejecuciones, de `1/5` a `5/5`, cubren el conjunto. Una lectura completa solo demuestra que los bytes almacenados están intactos. Si reconstruyen una base de datos funcional, con los roles y ajustes correctos, es algo que demuestra una restauración.

Lecturas adicionales

- Glosario NIST: backup — la definición utilizada aquí y las publicaciones que la respaldan.
- NIST SP 800-34 Rev. 1 — la guía de planificación de contingencias que define RPO y RTO.
- PostgreSQL: copias de seguridad y restauración — los tres enfoques y el coste de cada uno.
- PostgreSQL: archivado continuo y recuperación a un instante determinado — copias base, archivado WAL y restauración a un momento elegido.
- restic: preparar un repositorio nuevo — contraseñas, archivos de contraseñas y la advertencia sobre su pérdida.
- restic: eliminar instantáneas de copias de seguridad — políticas de retención, `forget`, `prune` y `--dry-run`.

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 copia de seguridad?» 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/backup/