Engineer

Tiempo de inactividad

período en que los usuarios no pueden usar un servicio

Qué es

El tiempo de inactividad es el período durante el que un servicio no funciona para las personas que dependen de él. También se denomina interrupción o indisponibilidad; su opuesto es el tiempo de actividad, y la proporción de tiempo durante la que el servicio funciona es su disponibilidad, normalmente expresada como un porcentaje mensual o anual.

El caso más claro es una caída total: el sitio no carga, la API rechaza conexiones o la base de datos está caída. El caso habitual es más complicado. Un proceso de pago que agota el tiempo de espera mientras el catálogo funciona, una página de acceso que carga pero rechaza todas las contraseñas, una API que responde a una de cada tres solicitudes: todos son tiempo de inactividad para quien necesita esa ruta, aunque una comprobación de estado dirigida a la página de inicio indique que todo está bien. Una definición útil cuenta lo que pueden hacer los usuarios, no si hay un proceso en ejecución.

Hay dos tipos de inactividad. La no planificada es la interrupción que nadie eligió: un fallo de disco, un despliegue defectuoso, un certificado caducado o una dependencia que cayó primero. La planificada se elige de antemano, normalmente dentro de una ventana de mantenimiento, para actualizar, migrar o trasladar algo que no puede cambiar mientras atiende tráfico. Es más fácil convivir con ella porque se puede avisar a los usuarios, pero sigue descontándose de la disponibilidad a menos que el acuerdo la excluya expresamente.

HTTP tiene un vocabulario para ese momento. RFC 9110 define la clase 5xx de códigos de estado para un servidor que sabe que ha cometido un error o no puede hacer lo solicitado. Tres de ellos significan cosas distintas durante una interrupción. 502 Bad Gateway indica que un proxy recibió una respuesta no válida del servidor situado detrás. 504 Gateway Timeout indica que el proxy

esperó demasiado a que llegara una respuesta. Y 503 Service Unavailable es el que se debe enviar deliberadamente: la especificación lo vincula a una sobrecarga temporal o un mantenimiento programado y permite al servidor indicar cuándo volver:

```
HTTP/1.1 503 Service Unavailable
Retry-After: 120
```

Eso comunica a navegadores, clientes de API y rastreadores de buscadores que la situación es temporal y sugiere esperar dos minutos. El mensaje es muy distinto al de una página rota servida con un código de éxito.

Utilice el concepto en una decisión

Aplique «¿Qué es el tiempo de inactividad?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Monitorización y alertas que avisan cuando un fallo aún es pequeño, y cifras que muestran con qué frecuencia se mantiene así.

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. Primer panel de observabilidad de la empresa — la primera visión de la empresa sobre sus propios sistemas

Primer panel de observabilidad de la empresa Póngase en contacto →

Por qué importa

Los objetivos de disponibilidad se expresan en nueves, y esos nueves se traducen en tiempo. En un año de 365 días, equivalente a 8.760 horas, cada objetivo permite este tiempo de inactividad:

- 99 %: 87,6 horas al año, unos tres días y medio;
- 99,9 %: 8,76 horas al año, o 43,2 minutos en un mes de 30 días;
- 99,95 %: 4,38 horas al año, o 21,6 minutos en un mes de 30 días;
- 99,99 %: 52,56 minutos al año;
- 99,999 %: unos 5,26 minutos al año.

Cada nueve adicional divide por diez el margen permitido, y el coste de conseguirlo aumenta mucho. Con un 99,9 %, hay tiempo para despertar a alguien y para que esa persona inicie sesión, encuentre el problema y lo corrija dentro del

presupuesto mensual. Con un 99,99 %, todo el año permite menos de una hora, quizá un solo incidente, por lo que la recuperación debe ser automática. Con un 99,999 %, incluso un reinicio rutinario debe ocurrir sin que lo noten los usuarios.

Las dependencias se multiplican. Una solicitud que necesita un balanceador de carga, una aplicación y una base de datos, cada uno disponible el 99,9 % del tiempo y con fallos independientes, solo se atiende cuando funcionan los tres: $0{,}999^3$, aproximadamente el 99,7 %, unas 26 horas de inactividad al año en lugar de 8,76. La disponibilidad de un servicio no puede superar la de la cadena de elementos que necesita, salvo que se eliminen eslabones o se dupliquen.

La planificación empieza por el negocio. La guía de planificación de contingencias para sistemas federales del NIST, SP 800-34, identifica tres cifras que conviene acordar antes de que algo falle. El tiempo máximo de inactividad tolerable (MTD) es la interrupción total que aceptará el responsable del sistema. El objetivo de tiempo de recuperación (RTO) es cuánto tiempo puede permanecer caído un sistema antes de que el impacto sea inaceptable, y debe ser menor que el MTD. El objetivo de punto de recuperación (RPO) indica hasta qué momento anterior pueden restaurarse los datos desde la última copia; mide, por tanto, la pérdida de datos tolerable, no el tiempo fuera de servicio. Con esos tres valores escritos, la programación de las copias y el diseño de la conmutación por error responden a una pregunta en lugar de ser conjeturas.

Cómo se aplica en la práctica

En una plataforma de streaming IPTV y OTT con clientes en China, Estados Unidos y Alemania, cada hora era el horario de máxima audiencia de alguien, así que no existía una hora tranquila para el mantenimiento. El soporte funcionaba las veinticuatro horas para unos mil servidores y un buen número de sistemas propiedad de los clientes. Gran parte del trabajo consistía en detectar desviaciones antes de que se convirtieran en interrupciones: en una emisión en directo, una caída corta lo que alguien está viendo en la televisión. La plataforma mantuvo su disponibilidad continua para su audiencia mundial.

En una empresa mediana, diez servidores de bases de datos PostgreSQL y SQL Server funcionaban con un 99,9 % de tiempo de actividad. Según las cuentas anteriores, eso permite 8,76 horas al año por servidor. Detrás de esa cifra había copias automatizadas con `pg_dump` y trabajos de SQL Server Agent, con políticas

de retención y almacenamiento fuera de la sede, monitorización en Prometheus y Grafana, y procedimientos de recuperación ante desastres documentados para restauraciones a un momento dado y conmutación por error.

Y en un conjunto de treinta servidores Ubuntu cuya configuración había crecido de forma accidental, se reconstruyó la parte de recuperación ante desastres con copias y procedimientos de restauración que sí se probaron. La red pasó a compartir una configuración de referencia endurecida en todas las instancias. El riesgo de inactividad disminuyó, con vías de recuperación probadas antes de que alguien las necesitara.

Dónde surgen los problemas

El monitor está dentro de la interrupción. Una comprobación de estado ejecutada en el mismo servidor o centro de datos sigue indicando que todo está bien mientras los usuarios encuentran un certificado TLS caducado, un registro DNS roto o un balanceador de carga que no apunta a nada. Compruebe desde fuera, desde más de un lugar y por el camino que sigue un usuario real, y genere alertas a partir de lo que vea esa comprobación.

La página de error dice 200. Un framework que representa una página de error amable con un código de éxito oculta la interrupción a la monitorización, a los clientes que reintentarían y a los rastreadores, que pueden indexarla en lugar de la página real. Los errores deberían devolver un código de estado de error, y el mantenimiento planificado debería responder con 503 y una cabecera Retry-After.

La copia nunca se restauró. Una copia solo demuestra que se copiaron bytes. Hasta que se haya ejecutado una restauración, medido su duración y comprobado el resultado con los datos, el tiempo de recuperación es una conjetura, a menudo muy superior al RTO escrito. Restaure periódicamente en un entorno separado y anote cuánto tardó. Lo mismo vale para la conmutación por error: un servidor en espera que nunca ha asumido el servicio es una esperanza; ensayar la conmutación en condiciones tranquilas ayuda a detectar problemas antes de un incidente.

Los lanzamientos provocan las interrupciones que debían evitar. Un despliegue que detiene la versión anterior antes de que esté preparada la nueva deja un hueco en cada lanzamiento. Un cambio de esquema que reescribe una tabla grande bajo un bloqueo exclusivo bloquea todas sus consultas hasta

terminar. Despliegue las instancias nuevas condicionando su entrada en servicio a una comprobación de preparación, antes de detener las antiguas; divida los cambios de esquema en una fase de ampliación y otra posterior de retirada, de modo que el código antiguo y el nuevo funcionen entre ambas, y, en Kubernetes, configure un PodDisruptionBudget para que el vaciado de un nodo no pueda detener todas las réplicas a la vez.

Lecturas adicionales

- RFC 9110: semántica de HTTP, sección 15.6 — Los códigos de estado 5xx, incluidos 503 y su indicación Retry-After.
- NIST SP 800-34, revisión 1: guía de planificación de contingencias para sistemas de información federales — Donde se definen el tiempo máximo de inactividad tolerable, RTO y RPO.
- PostgreSQL: alta disponibilidad, balanceo de carga y replicación — Los servidores en espera y la conmutación por error, tal como los describe el manual de la base de datos.
- PostgreSQL: archivado continuo y recuperación a un momento dado — La copia base y el archivo WAL de los que depende una restauración a un momento dado.
- Kubernetes: interrupciones — Las interrupciones voluntarias y los PodDisruptionBudgets que las limitan.

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 el tiempo de inactividad?» 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/downtime/