Engineer

Monitorización

vigilar un sistema en marcha para que avise primero

Qué es

La monitorización consiste en leer continuamente lo que un sistema en funcionamiento informa sobre sí mismo: algún mecanismo compara esos informes con límites definidos y decide cuándo debe avisar a una persona. El libro Site Reliability Engineering de Google la define como la recogida, el procesamiento, la agregación y la presentación de datos cuantitativos sobre un sistema en tiempo real: cuántas solicitudes llegan y de qué tipo, cuántas fallan y cómo, cuánto tardan y cuánto lleva funcionando cada servidor. El objetivo es un orden de acontecimientos: que el sistema avise de sus problemas antes que un usuario.

Los datos llegan por dos vías. La monitorización de caja blanca lee lo que el sistema expone sobre su interior, como un punto de acceso a métricas, la longitud de una cola o un recuento de reintentos; así puede detectar problemas en formación antes de que alguien los note. La monitorización de caja negra se sitúa fuera y se comporta como un usuario: solicita una página, resuelve un nombre o completa un inicio de sesión e informa de si funcionó y cuánto tardó. Solo ve lo que ya está fallando; por eso ofrece una razón fiable para despertar a alguien.

La mayor parte de lo que conserva un sistema de monitorización son cifras muestreadas a lo largo del tiempo: son baratas de almacenar y fáciles de comparar con un umbral. El mismo capítulo de SRE identifica cuatro como punto de partida para cualquier sistema que atienda a usuarios: latencia, el tiempo que tarda una solicitud; tráfico, la demanda que llega; errores, la proporción de solicitudes que fallan; y saturación, lo cerca que está de llenarse el recurso más ocupado.

Alrededor de las cifras hay dos salidas. Un panel sirve para consultar cuando alguien ya sospecha que ocurre algo. Una alerta sirve para interrumpir cuando todo el mundo está ocupado con otra cosa. En el ecosistema Prometheus, el servidor evalúa las reglas y Alertmanager agrupa, elimina duplicados y encamina las alertas que se activan.

Es fácil confundirla con dos prácticas cercanas. El registro en logs anota eventos individuales, una línea por suceso, y se consulta tras una alerta para averiguar la causa. La observabilidad, según la plantea el proyecto OpenTelemetry, es la capacidad más amplia de hacer preguntas nuevas a un sistema desde fuera. La monitorización es una práctica más acotada y antigua: preguntas decididas de antemano, formuladas continuamente, con una respuesta capaz de avisar al personal de guardia.

Utilice el concepto en una decisión

Aplique «¿Qué es la monitorización?» 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

La alternativa es enterarse de los problemas por quienes los sufren, a la hora que los detecten. La monitorización adelanta ese momento y pone el aviso en manos de alguien que pueda actuar.

El recurso escaso es la atención, más que el software. Un aviso de guardia interrumpe el trabajo de día y el sueño de noche; el libro de SRE es tajante sobre las consecuencias de recibir demasiados: la gente los lee por encima, desconfía de ellos y acaba ignorándolos, a veces también el verdadero que queda oculto entre el ruido. La orientación del proyecto Prometheus cabe en una frase: «mantenga las alertas sencillas, alerte sobre síntomas, disponga de buenas consolas para localizar las causas y evite los avisos de guardia ante los que no haya nada que hacer». Cada regla que puede despertar a alguien promete que, al activarse, habrá algo urgente que esa persona deba hacer.

Así, la decisión central se refiere a lo que sienten los usuarios. Un disco al 85 % es una causa; un proceso de pago que falla es un síntoma. Alertar sobre síntomas reduce el número de reglas: hay muchas menos maneras de perjudicar a los usuarios que de que las máquinas se comporten mal. Las causas siguen teniendo sitio en paneles y tickets, donde ayudan a diagnosticar sin despertar a nadie. La capacidad es la excepción justificada: un recurso casi lleno no perjudica a nadie hoy, pero pronto necesitará una intervención.

Después hay que decidir si operarlo o contratarlo. Operarlo exige un almacén de series temporales, un enrutador de alertas y alguien que mantenga ambos en buen estado. Un servicio alojado elimina ese trabajo y factura por el volumen enviado: cada servidor nuevo se refleja en una factura. El libro de SRE señala que, incluso en Google, un equipo de 10–12 ingenieros suele tener una o dos personas dedicadas principalmente a la monitorización. En cualquier caso, es un sistema con mantenimiento propio, que debe dimensionarse en función del sistema vigilado.

Cómo se aplica en la práctica

Dos puestos anteriores de la fundadora y el servidor propio de la empresa muestran la misma idea a escalas muy distintas.

Entre 2015 y 2022, en un puesto de ingeniería de infraestructura, el trabajo consistió en dar soporte 24/7 a unos 1.000 servidores de streaming IPTV/OTT, además de bastantes sistemas propiedad de los clientes, para clientes de China, Estados Unidos y Alemania. Con clientes en tres husos horarios no había una hora tranquila: un problema a las 3 de la madrugada local coincidía con el horario de máxima audiencia en otro lugar. Buena parte del trabajo era detectar desviaciones antes de que se convirtieran en interrupciones, y los problemas se resolvían antes de llegar a la pantalla del espectador. En un servicio así, el éxito se manifiesta en que no ocurre nada; la monitorización permite saberlo, en lugar de confiar en que sea así.

En un puesto de ingeniería de bases de datos entre 2022 y 2024, con 100 bases de datos PostgreSQL, MS SQL y BigQuery, la monitorización automatizada redujo el tiempo de indisponibilidad en un 50 %.

Monitorizar el servidor propio de la empresa para servicios de acceso público supone trabajar en el otro extremo de la escala. Una sola máquina de 512 MB ejecuta esos servicios, con unos 464 MB utilizables; la memoria disponible

condiciona cada decisión de diseño. Los informes se obtienen a petición, en lugar de emplear un agente, un flujo de procesamiento y un panel: siete órdenes leen el servidor y generan Markdown, sin cambiar nada y solo cuando alguien las solicita. La decisión sobre el servicio onion muestra el método: una vez planteada la pregunta, se tomaron muestras del servidor durante un día y las lecturas la resolvieron. En esas muestras registradas, el mínimo de memoria disponible fue de unos 310 MB de 464; el uso de swap se mantuvo en el 2,6 % y el procesador estuvo inactivo el 99,7 % del tiempo. A esa escala, monitorizar es tomar una medición en el momento en que una decisión la necesita.

Dónde surgen los problemas

El avisador suena tanto que la gente deja de leerlo. Cada incidente deja una regla; muchas se activan por causas que desaparecen solas y, en pocos meses, el aviso verdadero se pasa por alto junto con los demás. Avise al personal de guardia solo de lo que notarían los usuarios, en un único punto de la arquitectura, y envíe lo demás a una cola de tickets o a un panel. Exija después que la condición se mantenga un tiempo antes de activar la alerta, para que una anomalía que se resuelve en un minuto no despierte a nadie:

```
groups:
    - name: checkout
      rules:
          - alert: CheckoutErrorsHigh
            expr: job:checkout_errors:ratio5m > 0.02
            for: 10m
            labels:
                severity: page
```

La monitorización se detiene y su silencio se interpreta como una buena noticia. Una regla que nunca se activa parece exactamente igual que un sistema que nunca falla. Si cae el servidor Prometheus, Alertmanager o el retransmisor de correo, todas las alertas se detienen a la vez. Prometheus recomienda alertar sobre la propia monitorización y, mejor aún, probar todo el recorrido de extremo a extremo: una comprobación demuestra así que una alerta puede viajar desde la regla hasta la bandeja de entrada. Los trabajos programados caen en la misma trampa: una copia de seguridad que deja de ejecutarse no genera errores; simplemente deja de producir rastros. Avise cuando el último éxito sea demasiado antiguo, dejando margen para al menos dos ejecuciones completas; para un trabajo de una hora que se ejecuta cada 4 horas, el proyecto propone 10 horas:

```
time() - backup_last_success_timestamp_seconds > 10 * 3600
```

El promedio dice que todo va bien mientras algunos usuarios esperan. Los promedios son fáciles de representar y ocultan la cola de la distribución. El ejemplo del libro de SRE es un servicio con un promedio de 100 ms a 1.000 solicitudes por segundo, donde un 1 % de las solicitudes puede tardar fácilmente 5 segundos. Si una página depende de varios servicios así, el percentil 99 de un backend puede convertirse en la mediana del frontend. Los fallos rápidos lo empeoran: un error inmediato reduce el promedio. Registre la latencia por intervalos, alerte sobre un percentil alto y mida las solicitudes fallidas por separado de las correctas.

Todas las cifras internas están en verde, pero los usuarios no pueden acceder al servicio. Las métricas de caja blanca proceden del interior del sistema y comparten sus puntos ciegos: un certificado caducado, un registro DNS incorrecto o un cambio en el cortafuegos pueden dejarlas todas en buen estado. Una comprobación de caja negra desde una máquina ajena a la red del sistema detecta lo que de otro modo sería invisible y sigue funcionando cuando la monitorización interna cae con todo lo demás.

Lecturas adicionales

- Prometheus: visión general — qué recoge un sistema de monitorización basado en métricas, qué partes lo componen y cuándo la precisión por evento exige otra herramienta.
- Alertas — la orientación de Prometheus sobre cuándo avisar al personal de guardia, desde servicios en línea y trabajos por lotes hasta la vigilancia de la propia monitorización.
- Alertmanager — cómo se eliminan duplicados, se agrupan, silencian, inhiben y encaminan las alertas hacia un receptor.
- Monitorización de sistemas distribuidos — el capítulo 6 del libro Site Reliability Engineering de Google: síntomas y causas, las cuatro señales de oro y la cola de la distribución.
- Introducción a la observabilidad — la explicación de OpenTelemetry sobre señales, SLI y SLO y sobre cómo la observabilidad va más allá de la monitorización.

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 monitorización?» 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/monitoring/