Engineer

objetivo de nivel de servicio

Objetivo medible de fiabilidad que debe cumplir un servicio

Qué es

Un objetivo de nivel de servicio es una meta sobre el comportamiento de un servicio, redactada con tanta precisión que cualquiera pueda comprobar si se cumplió: una proporción de solicitudes que deben tener éxito o recibir respuesta con suficiente rapidez durante un periodo determinado. Casi todo el mundo lo abrevia como SLO. El libro Site Reliability Engineering de Google lo define como un valor objetivo, o un intervalo de valores, para un nivel de servicio medido mediante un indicador.

El indicador de nivel de servicio, o SLI, es una magnitud cuidadosamente definida para describir un aspecto del servicio que realmente reciben los usuarios. La latencia de las solicitudes suele ser la primera opción, seguida de cerca por la tasa de errores y la tasa de procesamiento. El SRE Workbook recomienda expresar cada SLI como una razón, eventos correctos divididos entre todos los eventos, para que vaya del 0 %, donde nada funciona, al 100 %, donde nada falla. El objetivo es entonces una línea en esa escala: el indicador debe mantenerse del lado correcto de la meta o entre un límite inferior y otro superior.

Un objetivo completo especifica todo lo necesario para comprobarlo. El ejemplo del libro dice: el 99 % de las llamadas RPC Get termina en menos de 100 ms, con el porcentaje promediado durante un minuto y la medición en todos los servidores backend. Un indicador puede tener varios objetivos a la vez, como el 90 % de las llamadas en menos de 1 ms, el 99 % en menos de 10 ms y el 99,9 % en menos de 100 ms, porque un solo percentil rara vez describe una distribución completa. El Workbook también separa la especificación, el resultado que importa al usuario, de la implementación, el lugar donde se mide: un registro del servidor, una sonda que ejecuta un navegador o código en la propia página. Cada opción detecta fallos distintos, y la elección debe constar en la definición.

Un SLO no es un SLA. Un acuerdo de nivel de servicio es un contrato con los usuarios, explícito o implícito, que asigna consecuencias al cumplimiento o incumplimiento de sus objetivos. Las más visibles son un reembolso o una penalización. La prueba que propone el libro consiste en preguntar qué ocurre si no se cumple el objetivo: si no hay una consecuencia explícita, es un SLO. Una nota al pie añade que quienes hablan de incumplir un SLA casi siempre se refieren a un SLO no alcanzado: un incumplimiento real de un SLA puede acabar en los tribunales por incumplimiento contractual. El libro señala que la búsqueda de Google no tiene un SLA para el público, pero una interrupción sigue costando reputación e ingresos publicitarios.

Utilice el concepto en una decisión

Aplique «¿Qué es un objetivo de nivel de servicio?» 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

Sin un objetivo declarado, los usuarios inventan el suyo. El libro SRE describe las dos formas de equivocarse: personas que confían en un servicio más de lo debido y posibles usuarios que creen que es menos fiable de lo que es y lo evitan. Su ejemplo es Chubby, el servicio de bloqueos de Google para sistemas distribuidos. Sus interrupciones globales eran tan raras que otros servicios acabaron dependiendo de que nunca fallara, sin que esa dependencia llamara la atención; esos servicios fallaban las pocas veces que ocurría. La respuesta fue hacer que la instancia global cumpliera su SLO sin superarlo demasiado. En cualquier trimestre en que los fallos reales no hubieran reducido la disponibilidad por debajo del objetivo, SRE detenía el sistema a propósito.

El objetivo también resuelve la discusión sobre cuánta fiabilidad es suficiente. «Una fiabilidad del 100 % es el objetivo equivocado», escribe el SRE Workbook, y sus razones son prácticas. Los componentes pueden fallar a la vez, sea cual sea su diseño. La cadena de redes y dispositivos entre un servicio y sus clientes también falla, así que los clientes nunca verían ese 100 %. Cada nueve adicional cuesta más que el anterior mientras su valor para los usuarios se acerca a cero. Y, como los cambios son la principal causa de interrupciones, un servicio que nunca puede fallar tampoco puede actualizarse. Una vez que el objetivo está por debajo del

100 %, la distancia entre él y la perfección deja margen para lanzamientos, experimentos y mantenimiento planificado: el presupuesto de errores, que es sencillamente el 100 % menos el SLO. Alguien debe responsabilizarse de ese equilibrio entre entregar funciones y mantener el servicio disponible. El Workbook señala al CTO en una organización pequeña y al responsable o gestor de producto en una mayor.

Por último, el objetivo convierte la monitorización en decisiones. El libro describe un ciclo: medir los indicadores, compararlos con los objetivos, decidir si hay que hacer algo y hacerlo. Si la latencia sube y cruzará el límite en unas horas, el equipo comprueba si la CPU es el cuello de botella en los servidores y añade más. Sin el objetivo, señala el libro, nadie puede decir si debe actuar ni cuándo.

Cómo se aplica en la práctica

Considere una API de finalización de compra con un objetivo redactado como propone el Workbook: con el indicador separado del lugar donde se mide, una ventana declarada y el razonamiento documentado a su lado:

```
Servicio:    API de finalización de compra
SLI:         fracción de solicitudes POST /checkout respondidas sin estado 5xx,
             contadas en el registro de solicitudes del servidor
Objetivo:    99.9% en una ventana móvil de 28 días
Motivo:      observacional, redondeado a la baja con datos de cuatro semanas
Responsable: gestor de producto, con los ingenieros de guardia
Revisión:    mensual hasta estabilizarse; después, trimestral
```

La ventana importa tanto como el porcentaje. El Workbook prefiere una ventana móvil a un mes natural, porque los usuarios no olvidan una interrupción del último día del mes cuando empieza el siguiente. También prefiere un número entero de semanas, para que todas las ventanas tengan el mismo número de fines de semana. Cuatro semanas es su opción de uso general. Esa ventana equivale a 28 × 24 × 60 = 40.320 minutos, y el 0,1 % equivale a algo más de 40 minutos de interrupción completa. En un mes de 30 días, el mismo 99,9 % permite 43,2 minutos, la cifra de la tabla de disponibilidad del libro.

Contar el tiempo es, sin embargo, la forma menos precisa. El libro SRE explica que Google mide la disponibilidad como tasa de éxito de las solicitudes, en lugar de tiempo en funcionamiento, porque un servicio distribuido por el mundo rara vez está completamente disponible o completamente caído. Expresado en

solicitudes, el ejemplo del Workbook queda así: con un 99,9 %, 3 millones de solicitudes en cuatro semanas dejan margen para 3.000 fallos, y una interrupción que provoque 1.500 consume la mitad del presupuesto. Contar solicitudes también carga al presupuesto los fallos parciales que no refleja el tiempo en funcionamiento. Con tráfico estable, 20 minutos durante los que falla la mitad de las solicitudes cuestan lo mismo que 10 minutos de interrupción total: aproximadamente una cuarta parte del presupuesto de 28 días.

En Prometheus, con un contador de solicitudes etiquetado por estado, el indicador de la ventana es una sola razón, basada en el propio ejemplo del Workbook:

```
sum(rate(http_requests_total{job="checkout", status!~"5.."}[28d]))
  /
sum(rate(http_requests_total{job="checkout"}[28d]))
```

En este sitio, la página de niveles de servicio está del lado de los acuerdos. Para un sistema que la empresa opera por cuenta de un cliente, el tiempo de respuesta a un incidente, las horas cubiertas, la disponibilidad exigida y las ventanas de mantenimiento se escriben en el acuerdo de ese sistema. Se establecen por sistema porque un informe nocturno y un servicio de pagos necesitan respuestas distintas. La monitorización propia de errores y el trazado construidos por la fundadora como parte de su trabajo actual en NextMariner se muestran en una serie de vistas para operadores, una de ellas dedicada a los objetivos de nivel de servicio. Y la nota sobre vender la última entrada exactamente una vez convierte una invariante de existencias en uno: compruébela según un calendario, represéntela en un gráfico y avise al personal de guardia ante cualquier desviación. Así, cero sobreventas se convierte en un objetivo respaldado por una alerta, no en una esperanza.

Dónde surgen los problemas

El objetivo se copia del panel del mes pasado. Adoptar lo que logra hoy el sistema parece seguro, pero el libro SRE advierte de que fijar un objetivo sin reflexión puede obligar al equipo a esfuerzos heroicos para seguir cumpliéndolo en un sistema que no puede mejorar sin rediseñarse. Parta de las necesidades del usuario y defina el objetivo a partir de ellas. Escriba el razonamiento junto a la cifra: el Workbook pide tanto la justificación como si los valores provienen de

experimentos o de observaciones, incluso cuando se reconozca que son valores ad hoc. Revise el documento cada mes al principio y cada trimestre cuando se estabilice.

El objetivo es el 100 % o una cadena de nueves cuyo coste nadie ha calculado. Construir y operar un sistema siempre disponible o que escale sin límite cuesta mucho, y ofrece más de lo que los usuarios notarían. Empiece con un objetivo poco exigente y hágalo más estricto conforme conozca el sistema. El libro lo prefiere a un objetivo estricto que después haya que relajar por resultar inalcanzable.

Una media sustituye a lo que perciben los usuarios. En el ejemplo del libro, una solicitud típica tarda unos 50 ms, mientras que el 5 % tarda veinte veces ese tiempo, y un gráfico de la media no muestra que pase nada durante todo el día. Redacte los objetivos de latencia como percentiles o como proporción de solicitudes por debajo de un umbral, por ejemplo, 300 ms. Nunca como una media.

Hay más objetivos de los que se pueden defender. Una docena de SLO por servicio diluye la importancia de todos. La prueba del libro es clara: si citar un objetivo nunca permite ganar una discusión sobre prioridades, probablemente no merezca la pena tenerlo. Conserve los pocos que cubran lo que los usuarios hacen realmente.

Incumplir el objetivo no cambia nada. El Workbook lo llama un SLO sin efecto práctico: el cumplimiento se convierte en otro KPI de un informe, en lugar de una razón para decidir algo. Acuerde de antemano qué ocurre cuando se agota el presupuesto y haga que lo firmen el gestor de producto, los desarrolladores y los SRE. La política de presupuesto de errores del ejemplo del Workbook suspende los cambios habituales mientras el servicio esté fuera de su objetivo. Solo permite corregir los errores de mayor prioridad y realizar trabajo de seguridad, y dice expresamente que la política no es un castigo.

El SLO tiene la misma cifra que el SLA. Cuando el objetivo interno coincide con el contrato, la primera noticia del problema es el incumplimiento. El libro recomienda un objetivo interno más estricto que el anunciado, como margen ante problemas persistentes. El error opuesto también importa: un servicio que funciona mucho mejor que su objetivo enseña a los usuarios a depender de ese

comportamiento. Por eso el libro sugiere interrupciones planificadas ocasionales, limitación de solicitudes o un sistema que no sea más rápido cuando tenga poca carga.

La ventana es más larga que los datos. Un objetivo de 28 días calculado en un servidor Prometheus que solo conserve 15 días de muestras, el valor predeterminado si no se establece la retención, mide sin avisar 15 días y los presenta como cuatro semanas. Configure una retención superior a la ventana antes de confiar en la cifra:

```
prometheus --storage.tsdb.retention.time=35d
```

Lecturas adicionales

- Objetivos de nivel de servicio — el capítulo 4 del libro Site Reliability Engineering de Google: indicadores, objetivos y acuerdos, y cómo elegir una meta.
- Implementar SLO — el capítulo del SRE Workbook sobre los SLI como razones, los presupuestos de errores, las ventanas y la documentación del objetivo.
- Ejemplo de política de presupuesto de errores — una política completa para un servicio ficticio de juegos: cuándo se detienen los lanzamientos y quién resuelve un desacuerdo.
- Aceptar el riesgo — por qué Google mide la disponibilidad según el éxito de las solicitudes y no por el tiempo en funcionamiento, y cuánto cuesta cada aumento de fiabilidad.
- Tabla de disponibilidad — cuánto tiempo de inactividad permite cada nivel de disponibilidad por año, trimestre, mes, semana, día y hora.
- Almacenamiento — cómo Prometheus conserva las muestras en disco y qué opciones de retención determinan hasta dónde puede retroceder una consulta de SLO.

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 un objetivo de nivel de servicio?» 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/service-level-objective/