Engineer

alerta

un mensaje del sistema cuando algo necesita atención humana

Qué es

Una alerta es una señal automática de que una condición vigilada se ha mantenido el tiempo suficiente para justificar interrumpir a alguien. Algunos equipos la llaman alarma, y una alerta lo bastante sonora como para despertar a la persona de guardia suele denominarse aviso de guardia, o page.

La palabra abarca dos objetos que Prometheus separa de forma clara. Una regla de alertas de Prometheus produce alertas: una por cada conjunto distinto de etiquetas para el que la expresión devuelve un resultado. Estas llegan a otro programa, Alertmanager, que las convierte en notificaciones: un correo, un mensaje de chat o una llamada a un servicio de guardias. El proyecto advierte que las reglas por sí solas no constituyen un sistema completo de notificación. La separación importa porque las cifras divergen. Cincuenta alertas pueden llegar en un mensaje, una puede generar recordatorios cada pocas horas y otra silenciada no llega a nadie aunque siga activa.

La alerta tiene un ciclo breve. Mientras la expresión no devuelve nada, está inactiva. Al devolver el primer resultado queda pendiente y permanece así durante lo exigido por `for`. Si la condición sigue cumpliéndose al terminar la espera, se activa y se entrega para su envío. Cuando deja de cumplirse, se resuelve. Una cláusula más reciente, `keep_firing_for`, mantiene abierta una alerta activa un tiempo después de desaparecer la condición; la documentación la propone para evitar oscilaciones y falsas resoluciones. Prometheus registra también los estados pendiente y activo en una serie temporal `ALERTS`, para consultar y representar las propias alertas.

Una regla que espera veinte minutos para activarse y diez más para cerrarse tiene este aspecto:

```
groups:
    - name: exports
      rules:
          - alert: ExportQueueStalled
            expr: exports_waiting > 0 and rate(exports_done_total[15m]) == 0
            for: 20m
            keep_firing_for: 10m
            labels:
                severity: page
                team: billing
            annotations:
                summary: "{{ $value }} exports queued and none finishing"
```

Las etiquetas y las anotaciones se parecen, pero hacen trabajos distintos. Las etiquetas forman parte de lo que identifica esta alerta y la distingue de otra, y Alertmanager las lee para decidir su destino: `severity: page` puede llegar a un teléfono, mientras que `severity: ticket` termina en una cola. Las anotaciones contienen las palabras que necesita una persona una vez recibido el mensaje: un resumen de una línea, una descripción más extensa y un enlace al procedimiento operativo que indica qué hacer primero. Las anotaciones pueden usar plantillas con las etiquetas de la alerta y el valor que la activó; así puede el resumen anterior incluir una cifra.

Alerta e incidente son palabras distintas por un motivo. La alerta informa de una condición cumplida; decidir si constituye un incidente corresponde a una persona.

Utilice el concepto en una decisión

Aplique «¿Qué es una alerta?» 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 alerta es la parte de la monitorización que no espera a que la miren. Una métrica queda almacenada, un panel espera en una pantalla y un registro espera una búsqueda; una alerta irrumpe en la tarde de alguien. Cada regla exige así de

forma permanente una pequeña parte de su atención, y solo es justo si puede hacer algo con ella. La guía de Prometheus recomienda evitar avisos de guardia sin acción posible y mantener sencillo el sistema de alertas.

La segunda razón es aritmética. Una sola causa suele activar muchas reglas a la vez. Cuando una partición de red separa de su base de datos a la mitad de las instancias de un servicio, cada instancia puede disparar su propia alerta, y la documentación de Alertmanager usa exactamente ese caso para mostrar cómo cientos de alertas se convierten en una única notificación. Alertmanager reúne las alertas que comparten las etiquetas nombradas en `group_by`, espera `group_wait` (30 segundos de forma predeterminada) para que el primer mensaje contenga todo el lote, envía actualizaciones del grupo con una frecuencia no mayor que `group_interval` (5 minutos) y repite un mensaje cuya alerta siga activa después de `repeat_interval` (4 horas). La inhibición va un paso más allá y suprime algunas alertas mientras otra está activa, de modo que una alerta que indica que todo un clúster es inaccesible puede silenciar las de los servicios que contiene. Los silencios desactivan las notificaciones de las alertas que coinciden con un conjunto de etiquetas durante un plazo fijo, normalmente durante un mantenimiento planificado.

La tercera razón es el final. La mayoría de receptores envía un mensaje de resolución, pero `send_resolved` del receptor de correo vale false de forma predeterminada. Un equipo que avisa por correo puede conocer el problema y nunca su fin.

Cómo se aplica en la práctica

El host de la empresa utiliza una alarma por ausencia de señal. Sus 464 MB de memoria sostienen un servidor web y una forja, y el agotamiento de la memoria y un disco lleno son dos riesgos importantes de fallo. Un temporizador en el host envía una señal a un servicio externo solo si se superan las comprobaciones de los umbrales de memoria disponible y espacio libre en disco, cada quince minutos con cierta variación; el servicio externo activa la alarma cuando dejan de llegar las señales esperadas. Los fallos que impiden al script ejecutarse o contactar con ese servicio, como la pérdida de red o un arranque fallido, producen una señal ausente. Una señal recibida confirma que el script se ejecutó y se superaron sus comprobaciones de los umbrales de memoria disponible y espacio libre en disco; no certifica el buen estado general del host ni que el sitio esté sirviendo páginas. La monitorización puede revelar señales de advertencia, aunque el momento en que aparecen depende de la carga de trabajo y de las condiciones observadas.

La misma inversión puede proteger el trabajo programado. La implementación de copias de seguridad separa las señales de éxito de copia y restauración. Sus scripts y unidades systemd permiten instantáneas cifradas fuera del host, comprobaciones de integridad y una prueba mensual de restauración de un archivo conocido; solo se envía una señal tras completarse correctamente su flujo. Ese es el diseño de la ruta de alertas, no una prueba de que estuviera activa. El registro de implementación del 2026-10-27 indica que los temporizadores estaban desactivados al no haber un repositorio configurado y no documenta ninguna restauración real completada. El silencio puede producir una alerta externa una vez activados el flujo y su monitor; la configuración por sí sola no demuestra que estén funcionando.

Las alertas invertidas tienen un fallo propio, descubierto al preparar el crecimiento de uno a dos hosts. Compartir un mecanismo de vigilancia entre ambos habría debilitado las alertas que se pretendía ampliar: mientras cualquiera siga enviando señales, la comprobación aparece correcta aunque el otro esté muerto. Cada mecanismo pasó a ser un valor por host y las entradas del host existente se dejaron vacías deliberadamente: la configuración indica que el cableado existe y aún falta proporcionar el valor.

Dónde surgen los problemas

Un valor medido termina en una etiqueta. Como las etiquetas identifican la alerta, poner la cifra actual convierte cada cambio en una alerta nueva y resuelve la anterior. El receptor ve pares de activación y resolución para un solo problema. Reserve etiquetas para identidad y enrutamiento; ponga cifras en anotaciones.

La alerta oscila. Una condición cerca del umbral se activa, resuelve y reactiva cada pocos minutos; cada resolución dice algo falso. Alargue `for` para que un pico desaparezca antes de avisar y use `keep_firing_for` para que una caída breve bajo el umbral no cierre la alerta.

Un silencio sobrevive a su motivo. Alguien silencia una alerta por mantenimiento con una condición de coincidencia amplia y una hora de fin generosa, y el fallo real de la próxima semana llega silenciado. Limite las coincidencias tanto como permita el trabajo, establezca una hora de fin cercana al término previsto y escriba quién lo estableció y por qué en el campo de comentario que ofrece Alertmanager.

Una ruta amplia se traga las específicas. Alertmanager recorre el árbol de enrutamiento en orden y, salvo que una ruta establezca `continue: true`, se detiene en el primer hijo coincidente. Una ruta que lo recoge todo situada antes de la propia de un equipo se lleva las alertas de ese equipo. Sitúe primero las rutas más concretas y pruebe el árbol con etiquetas de muestra antes de confiar en él.

Todo es urgente. Un disco al 70 %, un certificado que caduca el mes siguiente y un servicio de pagos que devuelve errores generan avisos al mismo teléfono, que pronto acaba ignorándose. Reserve los avisos de guardia para condiciones ante las que alguien deba actuar ahora, envíe lo demás a una cola de tickets o a un resumen diario y elimine una regla que lleve meses activándose sin que nadie actúe.

Lecturas adicionales

- Panorama de las alertas — cómo se reparten el trabajo las reglas de alerta de Prometheus y Alertmanager, y los pasos para configurar ambos.
- Reglas de alertas — la sintaxis de las reglas, incluidos `for`, `keep_firing_for`, etiquetas, anotaciones y plantillas.
- Alertmanager — agrupación, inhibición, silencios y ejecución de varias instancias de Alertmanager para alta disponibilidad.
- Configuración de Alertmanager — el árbol de enrutamiento, los temporizadores de agrupación, las reglas de inhibición y los ajustes de cada receptor, entre ellos `send_resolved`.
- Prácticas de alertas — los consejos del propio proyecto sobre qué merece un aviso de guardia y qué corresponde a otro canal.

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 alerta?» 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/alert/