Engineer

Monitoring

ein laufendes System beobachten, damit es zuerst warnt

Was es ist

Monitoring ist das fortlaufende Lesen dessen, was ein laufendes System über sich selbst meldet: Ein Mechanismus vergleicht diese Meldungen immer wieder mit festgelegten Grenzen und entscheidet, wann ein Mensch davon erfahren muss. Googles Buch Site Reliability Engineering definiert Monitoring als das Erfassen, Verarbeiten, Zusammenfassen und Anzeigen quantitativer Systemdaten in Echtzeit: wie viele Anfragen welcher Art eingehen, wie viele auf welche Weise scheitern, wie lange sie dauern und wie lange jeder Server bereits läuft. Das Ziel ist eine Reihenfolge: Das System meldet seine Schwierigkeiten, bevor ein Nutzer es tut.

Die Daten kommen aus zwei Richtungen. White-Box-Monitoring liest, was ein System über sein Inneres offenlegt, etwa einen Metrik-Endpunkt, eine Warteschlangenlänge oder einen Wiederholungszähler. Es kann damit Probleme entstehen sehen, bevor jemand sie spürt. Black-Box-Monitoring steht außerhalb und verhält sich wie ein Nutzer: Es ruft eine Seite ab, löst einen Namen auf oder schließt eine Anmeldung ab und meldet, ob das gelang und wie lange es dauerte. Es sieht nur, was schon kaputt ist. Gerade das macht es zu einem verlässlichen Grund, jemanden zu wecken.

Ein Monitoring-System speichert überwiegend Zahlen, die über die Zeit stichprobenartig erfasst werden. Zahlen lassen sich günstig speichern und leicht gegen einen Schwellenwert prüfen. Dasselbe SRE-Kapitel nennt vier als Ausgangspunkt für alles, was Nutzern dient: Latenz, also die Dauer einer Anfrage; Datenverkehr, also die eingehende Last; Fehler, also den Anteil gescheiterter Anfragen; und Sättigung, also die Nähe der am stärksten beanspruchten Ressource zur Vollauslastung.

Um die Zahlen herum gibt es zwei Ausgaben. Ein Dashboard dient zum Nachsehen, wenn jemand bereits einen Verdacht hat. Ein Alarm dient zum Unterbrechen, wenn alle mit etwas anderem beschäftigt sind. Im Prometheus-Ökosystem wertet der Server die Regeln aus, und Alertmanager gruppiert, dedupliziert und verteilt die ausgelösten Alarme.

Zwei verwandte Begriffe werden leicht damit verwechselt. Logging hält einzelne Ereignisse fest, eine Zeile pro Geschehen. Nach einem Alarm liest jemand darin nach, um den Grund zu finden. Observability ist nach der Einordnung des OpenTelemetry-Projekts die weiter reichende Fähigkeit, einem System von außen neue Fragen zu stellen. Monitoring ist die engere und ältere Praxis: vorher festgelegte Fragen, fortlaufend gestellt, mit einer Antwort, die einen Menschen alarmieren kann.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist Monitoring?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Monitoring und Alarme melden sich, solange ein Fehler noch klein ist; Kennzahlen zeigen, wie oft das gelingt.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. Das erste Observability-Dashboard aufgebaut — der erste Einblick des Unternehmens in die eigenen Systeme

Das erste Observability-Dashboard aufgebaut Kontakt aufnehmen →

Warum es wichtig ist

Die Alternative ist, erst durch die Betroffenen von einem Problem zu erfahren, zu welcher Uhrzeit sie es auch bemerken. Monitoring verlegt diesen Moment nach vorn und leitet die Meldung an jemanden weiter, der handeln kann.

Die knappe Ressource ist eher Aufmerksamkeit als Software. Ein Alarm unterbricht tagsüber die Arbeit und nachts den Schlaf. Das SRE-Buch sagt deutlich, was bei zu häufigen Alarmen passiert: Menschen überfliegen sie, zweifeln sie an und ignorieren sie schließlich, mitunter samt dem echten Alarm im Rauschen. Die Empfehlung des Prometheus-Projekts passt in einen Satz: „Die Alarmierung einfach halten, auf Symptome alarmieren, gute Konsolen zum Eingrenzen der Ursachen bieten und Alarme ohne Handlungsmöglichkeit vermeiden.“ Jede Regel, die jemanden wecken kann, ist das Versprechen, dass beim Auslösen etwas Dringendes zu tun ist.

Damit dreht sich die zentrale Entscheidung darum, was Nutzer spüren. Eine zu 85 % volle Platte ist eine Ursache; ein scheiternder Bezahlvorgang ein Symptom. Auf Symptome zu alarmieren hält die Zahl der Regeln klein: Es gibt weit weniger Arten, wie Nutzer beeinträchtigt werden, als Arten, wie Maschinen sich falsch verhalten. Ursachen gehören weiterhin auf Dashboards und in Tickets. Dort helfen sie bei der Diagnose, ohne jemanden zu wecken. Kapazität ist die ehrliche Ausnahme: Eine fast volle Ressource schadet heute noch niemandem, wird aber bald einen Menschen brauchen.

Dann stellt sich die Frage nach Eigenbetrieb oder gemietetem Dienst. Eigenbetrieb bedeutet einen Zeitreihenspeicher, einen Alarmverteiler und jemanden, der beide gesund hält. Ein gehosteter Dienst nimmt diese Arbeit ab und rechnet nach gesendetem Volumen ab. Jeder neue Host erscheint damit auf einer Rechnung. Laut SRE-Buch sind selbst bei Google in einem Team von 10–12 Ingenieurinnen und Ingenieuren typischerweise ein oder zwei Mitglieder hauptsächlich für Monitoring zuständig. In beiden Fällen ist es ein System mit eigenem Wartungsbedarf, dessen Größe zum überwachten System passen muss.

Wie es in der Praxis aussieht

Zwei frühere Tätigkeiten der Gründerin und der unternehmenseigene Host zeigen dieselbe Idee in sehr unterschiedlichen Größenordnungen.

Von 2015 bis 2022 umfasste eine Tätigkeit als Infrastrukturingenieurin den 24/7-Support für etwa 1.000 IPTV-/OTT-Streamingserver sowie zahlreiche kundeneigene Systeme für Kunden in China, den USA und Deutschland. Bei Kunden in drei Zeitzonen gab es keine ruhige Stunde. Ein Problem um 3 Uhr morgens Ortszeit fiel anderswo in die Hauptsendezeit. Ein großer Teil der Arbeit bestand darin, schleichende Veränderungen vor einem Ausfall zu bemerken. Probleme wurden behoben, bevor sie den Bildschirm eines Zuschauers erreichten. Bei einem solchen Dienst sieht Erfolg so aus, als passiere nichts. Monitoring erlaubt dem Team, das zu wissen, statt es zu hoffen.

In einer Tätigkeit als Datenbankingenieurin von 2022 bis 2024 reduzierte automatisiertes Monitoring über 100 PostgreSQL-, MS-SQL- und BigQuery-Datenbanken hinweg die Ausfallzeit um 50 %.

Das Monitoring des unternehmenseigenen Hosts für öffentliche Dienste liegt am anderen Ende der Größenordnung. Eine einzelne Maschine mit 512 MB betreibt diese Dienste. Davon sind etwa 464 MB nutzbar, und der Speicher entscheidet

dort jede Entwurfsfrage. Die Berichte werden auf Anfrage abgerufen, statt über Agent, Pipeline und Dashboard zu laufen: sieben Befehle, die den Host lesen und Markdown ausgeben, nichts ändern und nur auf Anfrage laufen. Die Entscheidung über den Onion-Dienst zeigt die Methode: Als die Frage aufkam, wurde der Host einen Tag lang stichprobenartig gemessen, und die Messwerte entschieden. In diesen aufgezeichneten Stichproben lag der Tiefststand des verfügbaren Speichers bei etwa 310 MB von 464, die Swap-Nutzung bei 2,6 % und der Prozessor war 99,7 % der Zeit untätig. In dieser Größenordnung ist Monitoring eine Messung genau dann, wenn eine Entscheidung sie braucht.

Wo es schiefgeht

Der Pager meldet sich so oft, dass niemand mehr liest. Jeder Vorfall hinterlässt eine Regel. Viele schlagen bei Ursachen an, die von selbst verschwinden. Nach wenigen Monaten wird der echte Alarm zusammen mit den übrigen überflogen. Alarmieren Sie nur bei dem, was Nutzer bemerken würden, und nur an einer Stelle im Stack. Alles andere geht in eine Ticket-Warteschlange oder auf ein Dashboard. Lassen Sie die Bedingung dann eine Weile bestehen, bevor sie auslöst. So weckt eine nach einer Minute verschwundene Störung niemanden:

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

Das Monitoring fällt aus, und seine Stille wirkt wie eine gute Nachricht. Eine Regel, die nie auslöst, sieht genauso aus wie ein System, das nie ausfällt. Fallen der Prometheus-Server, Alertmanager oder das Mail-Relay aus, enden alle Alarme zugleich. Prometheus selbst rät, das Monitoring zu überwachen. Noch besser ist ein Test des gesamten Wegs von Anfang bis Ende: Eine Prüfung belegt dann, dass ein Alarm von der Regel bis ins Postfach gelangt. Zeitgesteuerte Aufgaben geraten in dieselbe Falle. Eine Sicherung, die nicht mehr läuft, wirft keinen Fehler; sie hinterlässt einfach keine Spuren mehr. Alarmieren Sie, wenn der letzte Erfolg zu lange zurückliegt, und lassen Sie mindestens zwei vollständige Durchläufe zu. Für eine einstündige Aufgabe, die alle 4 Stunden läuft, empfiehlt das Projekt 10 Stunden:

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

Der Durchschnitt meldet Entwarnung, während manche Nutzer warten. Durchschnitte lassen sich leicht darstellen und verbergen den langsamen Teil der Verteilung. Das SRE-Buch nennt als Beispiel einen Dienst mit durchschnittlich 100 ms bei 1.000 Anfragen pro Sekunde, bei dem 1 % der Anfragen leicht 5 Sekunden dauern kann. Hängt eine Seite von mehreren solchen Diensten ab, kann das 99. Perzentil eines Backends zum Median des Frontends werden. Schnelle Fehler verschärfen das, weil ein rascher Fehler den Durchschnitt nach unten zieht. Erfassen Sie Latenz in Klassen, alarmieren Sie auf ein hohes Perzentil und verfolgen Sie fehlgeschlagene Anfragen getrennt von erfolgreichen.

Jede interne Zahl ist grün, während Nutzer den Dienst nicht erreichen. White-Box-Metriken kommen aus dem Inneren des Systems und teilen deshalb seine blinden Flecken: Ein abgelaufenes Zertifikat, ein defekter DNS-Eintrag oder eine Firewall-Änderung kann jede dieser Zahlen gesund aussehen lassen. Eine Black-Box-Prüfung von einer Maschine außerhalb des Systemnetzes erkennt, was sonst unsichtbar bleibt. Sie arbeitet auch dann weiter, wenn das interne Monitoring zusammen mit allem anderen ausfällt.

Weiterführende Lektüre

- Prometheus: Überblick — was ein metrikbasiertes Monitoring-System erfasst, aus welchen Teilen es besteht und wann Genauigkeit für Einzelereignisse etwas anderes verlangt.
- Alarmierung — die Empfehlungen des Prometheus-Projekts zu alarmwürdigen Zuständen: von Online-Diensten und Stapelaufgaben bis zum Monitoring des Monitorings.
- Alertmanager — wie Alarme dedupliziert, gruppiert, stummgeschaltet, durch andere Alarme unterdrückt und an einen Empfänger verteilt werden.
- Verteilte Systeme überwachen — Kapitel 6 von Googles Buch Site Reliability Engineering: Symptome und Ursachen, die vier goldenen Signale und den langsamen Teil der Verteilung.
- Einführung in Observability — OpenTelemetrys Darstellung von Signalen, SLIs und SLOs und davon, wo Observability über Monitoring hinausgeht.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Bereitstellungsweg, die Betriebsrisiken und die Bereitschaftsverantwortlichen. Wir können eine Änderung mit Prüfungen, einem Wiederherstellungsweg und klaren Zuständigkeiten abgrenzen.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist Monitoring?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/monitoring/