**Engineer**

# Metriken

Zahlen, die ein System über sich selbst meldet

## Was es ist

Metriken sind Messungen eines laufenden Diensts. Jede wird zu einem Zeitpunkt erhoben und als Wert mit Zeitstempel und einigen Labels aufbewahrt. Dieselbe Größe lässt sich dadurch eine Minute, einen Tag oder einen Monat später erneut ablesen. Eine benannte Größe ist eine Metrik. Als Folge von Werten über die Zeit gespeichert wird sie zur Zeitreihe. OpenTelemetry zählt Metriken neben Traces und Logs zu seinen Signalen. Beantwortete Anfragen, belegter Speicher und wartende Aufträge in einer Warteschlange sind typische Metriken.

Was Metriken von den anderen Signalen unterscheidet, ist die Aggregation. Ein Log hält eine Zeile pro Ereignis fest, ein Trace verfolgt eine Anfrage durch das System. Eine Metrik fasst viele Messungen über ein Zeitfenster in wenigen statistischen Angaben zusammen: einer Anzahl, einer Summe, einem aktuellen Wert oder einer Verteilung auf Klassen. Die OpenTelemetry-Dokumentation nennt das Ergebnis aggregierte statistische Information. Darin liegen Stärke und Grenze zugleich: Eine Metrik sagt, wie viele Anfragen in der letzten Minute fehlschlugen, ein Log sagt, welche.

OpenTelemetry und Prometheus stimmen bei einer kleinen Auswahl von Messwerttypen überein. Den richtigen zu wählen macht den größten Teil des Entwurfs aus:

- Ein **Zähler**, englisch Counter, summiert auf und steigt nur, wie der Kilometerzähler eines Autos: beantwortete Anfragen, gesendete Bytes, erkannte Fehler. Prometheus erlaubt eine Ausnahme, das Zurücksetzen auf null beim Neustart des Prozesses.
- Ein **Auf-und-ab-Zähler**, bei OpenTelemetry Up-Down Counter genannt, summiert auf, kann aber auch sinken, etwa die Länge einer Warteschlange, wenn Arbeit hinzukommt und abgearbeitet wird.

- Eine **Gauge** ist ein zum Ablesezeitpunkt erfasster aktueller Wert, wie die Tankanzeige desselben Autos: belegter Speicher, Temperatur oder gerade laufende Anfragen.
- Ein **Histogramm** ordnet Messungen Klassen zu und hält ihre Anzahl und Summe fest. Es kann dadurch beantworten, wie viele Anfragen weniger als eine Sekunde dauerten.

Prometheus hat außerdem eine Summary, die Quantile im Programm berechnet. Das OpenTelemetry-Datenmodell führt sie nur als historischen Typ, weil sich Summaries nicht immer sinnvoll zusammenführen lassen.

Das von Prometheus abgefragte Textformat zeigt die Teile zusammen: eine Hilfezeile, eine Typzeile und danach eine Zeile je Labelkombination mit Namen, Labels, Wert und optionalem Zeitstempel in Millisekunden seit der Epoche:

```
# HELP http_requests_total The total number of HTTP requests.
# TYPE http_requests_total counter
http_requests_total{method="post",code="200"} 1027 1395066363000
http_requests_total{method="post",code="400"} 3 1395066363000
```

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was sind Metriken?“ 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

Metriken sind günstig genug, um alles zu erfassen. Ihr Speicherbedarf wächst mit der Zahl unterschiedlicher Labelkombinationen, nicht mit dem Datenverkehr. Das sagt die OpenTelemetry-Dokumentation im Gegensatz zu Logs ausdrücklich. Ein Zähler, der eine Milliarde Mal erhöht wurde, ist weiterhin eine einzige Zahl. Deshalb kann ein Team jede Anfrage statt einer Stichprobe zählen und die Zahlen lange genug aufbewahren, um diesen Monat mit dem letzten zu vergleichen.

Sie machen aus Diskussionen Messungen. Ob ein Host groß genug ist, ob eine Veröffentlichung Antworten langsamer gemacht hat, wie bald eine Platte voll wird: Jede Frage hat eine Antwort in einer schon vorher aufgezeichneten Zeitreihe. Alarme beruhen auf denselben Zahlen. Eine Alarmregel ist eine Bedingung über einer Metrik, für die jemand entschieden hat, dass sie das Aufwecken wert ist.

Metriken überleben auch den Code, der sie ausgibt. Ihre Form sollte deshalb stimmen. Der Prometheus-Namensleitfaden verlangt eine Einheit und eine Größe pro Metrik, in Basiseinheiten wie Sekunden und Bytes statt Millisekunden oder Megabytes. Die Einheit steht als Suffix im Plural, eine auflaufende Anzahl erhält `_total`. Namen wie `http_request_duration_seconds` und `process_cpu_seconds_total` sagen dann schon selbst, was sie enthalten.

Auch die Übertragung von Summen ist wichtig. Prometheus meldet kumulative Gesamtwerte, sodass sich Lücken durch gelegentliche Erfassungsfehler ausgleichen. Der Preis ist Speicher im Sender proportional zu dessen Labelkombinationen. StatsD meldet Deltas, also die Änderung seit der letzten Meldung, und verlagert diese Kosten aus dem Prozess.

## Wie es in der Praxis aussieht

Der eigene Host für öffentliche Dienste des Unternehmens ist ein Rechner mit einem Kern und 512 MB Speicher, wovon ungefähr 464 MB nutzbar sind. Auf ihm wurde Speicher zum Argument, das Entscheidungen klärte. Die Berichte werden aktiv abgerufen: sieben Befehle, die den Host auslesen und Markdown erzeugen, nichts verändern und nur auf Anforderung laufen. Als die Frage aufkam, ob dieser eine Host mit 512 MB auch einen Onion-Dienst tragen könnte, kam die Antwort aus einem aufgezeichneten Tag mit Messstichproben statt aus einer Meinung: Der verfügbare Speicher sank nie unter ungefähr 310 MB von 464, Swap lag bei 2,6 Prozent und der Prozessor war zu 99,7 Prozent untätig.

Später mussten diese Berichte angeben, was ihre Zahlen wert waren. Sechs rein lesende Kollektoren mussten erklären, was sie nie lasen, nachdem klar geworden war, dass eine blinde Erfassung und ein gesunder Host dieselbe Seite erzeugten: Ein Mirror, dessen Daemon das Logformat geändert hatte, lieferte dieselbe Null wie einer, den niemand besuchte. Grenzwerte wurden an die 116 bereits vom Metrikbericht erhobenen Stichproben angepasst; die beobachteten Bereiche stehen neben den Grenzen. Der Bericht unterscheidet drei Ergebnisse: Grenzverletzung, keine Grenzverletzung und nicht bewertet, weil nichts

gemessen wurde. Jeder Kollektor meldet seine Laufzeit und seine Blindstellen, etwa gelesene gegenüber unlesbaren Dateien, selbst wenn die Antwort null lautet. Die Grenzwerte gelten für Tagesmittelwerte, die eine einminütige Spitze verbergen können. Der Schalter mit seinem Fünfzehn-Minuten-Takt prüft bei jedem Lauf den verfügbaren Speicher und den freien Plattenplatz. Eine kurze Grenzverletzung zwischen seinen Prüfungen kann trotzdem unbemerkt bleiben.

Die nützlichsten Metriken stammen oft aus dem Problem statt vom Host. Die Notiz zum Verkauf begrenzter Bestände ohne Überverkauf zeigt Wartezeiten auf Sperren, abgelaufene Reservierungen und den Abstand zwischen der Zählung des Zugangstors und der Datenbank auf einem Dashboard. Sie prüft die Invariante `available + held + sold = total` nach Zeitplan, stellt sie grafisch dar und benachrichtigt bei jeder Abweichung den Bereitschaftsdienst. Jede dieser Zahlen entsteht aus dem Wissen, wie dieser Verkauf schiefgeht.

## Wo es schiefgeht

Metriken scheitern selten laut. Die Kurve wird weiter gezeichnet, und die angezeigte Zahl bedeutet unbemerkt etwas anderes, als ihr Name sagt.

**Ein sinkender Wert wird als Zähler veröffentlicht.** Die Prometheus-Seite zu Metriktypen sagt, keinen Zähler für etwas zu verwenden, das sinken kann, etwa die Zahl laufender Prozesse. Das ist eine Gauge. Die Abfragefunktionen behandeln jedes Absinken eines Zählers als Neustart und korrigieren es. Ein absichtlich sinkender Wert erscheint dadurch als Anstieg, der nie stattgefunden hat.

**Einheiten driften zwischen Diensten auseinander.** Ein Dienst meldet eine Dauer in Millisekunden, ein anderer in Sekunden. Ein vergleichendes Panel liegt um den Faktor tausend falsch. Der Namensleitfaden sagt, dass sich eine Metrik auf eine einzige Einheit beziehen muss und Basiseinheiten verwenden sollte. Einmal an der Quelle umzurechnen ist günstiger, als den Unterschied in jeder Abfrage mitzudenken.

**Eine Summe über Labels sagt nichts aus.** Die Faustregel des Namensleitfadens lautet, dass die Summe oder der Durchschnitt über alle Labels einer Metrik sinnvoll sein sollte. Sein eigenes Beispiel: Die Kapazitäten mehrerer Warteschlangen gehören in einer Metrik zusammen, die Kapazität einer Warteschlange und ihre aktuelle Länge dagegen nicht.

**Nicht gemessen wird als null gelesen.** Ein Kollektor, der seine Eingabe nicht mehr liest, hinterlässt eine Null oder eine fehlende Zeile, die genau wie ein ruhiges System aussieht. Das OpenTelemetry-Datenmodell hat ein Flag für einen Punkt ohne aufgezeichneten Wert, das Gegenstück zum Prometheus-Marker für nicht mehr aktuelle Werte. Fehlende Daten und der Wert null sind verschiedene Tatsachen. Melden Sie sie verschieden und geben Sie „nicht bewertet“ ein eigenes Ergebnis, wie die Kollektoren oben.

**Das Zeitfenster versteckt die Spitze.** Eine Gauge liefert eine Momentaufnahme. Das OpenTelemetry-Datenmodell meldet nur den letzten Wert jedes Intervalls, sodass ein Ausschlag zwischen zwei Messungen keine Spur hinterlässt. Ein Tagesmittel glättet genauso eine Minute an der Obergrenze weg. Wählen Sie das Fenster nach dem zu erkennenden Fehler und lassen Sie eine schnelle Prüfung neben einer langsamen laufen, wenn beides zählt.

**Ein Überlauf lässt eine gefilterte Abfrage zu wenig zählen.** Das OpenTelemetry-SDK verfolgt standardmäßig höchstens 2.000 unterschiedliche Attributkombinationen je Metrikdatenstrom und fasst alle weiteren in einem einzigen Datenpunkt mit `otel.metric.overflow=true` zusammen. Die Gesamtsumme bleibt richtig. Der Überlauf ersetzt aber die gesamte Kombination, nicht nur ihren störenden Teil. Im eigenen Beispiel der Dokumentation trägt ein Anfragezähler `url.path` und `success`. Nach dem Überlauf zählt eine Abfrage nach `success=false` zu wenig, und ein darauf beruhender Fehlerratenalarm kann aufhören auszulösen. Beschränken Sie Attribute auf kleine, bekannte Mengen und achten Sie auf den Überlaufmarker.

## Weiterführende Lektüre

- OpenTelemetry: Metriken — Instrumentarten, Aggregation und die Kardinalitätsgrenze.
- OpenTelemetry: Metrikdatenmodell — Gauges, Summen, Histogramme, Zeitbezug und das Flag für nicht aufgezeichnete Werte.
- Prometheus: Metriktypen — Counter, Gauge, Histogramm und Summary.
- Prometheus: Metrik- und Labelnamen — Basiseinheiten, Einheitensuffixe und `_total`.
- Prometheus: Histogramme und Summaries — wo Quantile berechnet werden und welche sich zusammenführen lassen.
- Prometheus: Ausgabeformate — das Textformat Zeile für Zeile.

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 sind Metriken?“ 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/metrics/