Engineer

Ausfallzeit

⊗ Zeit, in der Nutzer einen Dienst nicht verwenden können

Was es ist

Ausfallzeit ist die Zeit, in der ein Dienst für die Menschen, die ihn brauchen, nicht nutzbar ist. Andere Bezeichnungen sind Ausfall und Nichtverfügbarkeit; das Gegenteil ist Betriebszeit. Der Anteil der Zeit, in dem ein Dienst funktioniert, ist seine Verfügbarkeit, meist als Prozentsatz über einen Monat oder ein Jahr angegeben.

Der eindeutige Fall ist ein Totalausfall: Die Website lädt nicht, die API verweigert Verbindungen, die Datenbank ist ausgefallen. Der häufigere Fall ist unübersichtlicher. Ein Checkout mit Zeitüberschreitung bei funktionierendem Katalog, eine ladende Anmeldeseite, die jedes Passwort ablehnt, eine API, die nur eine von drei Anfragen beantwortet: Jeder Fall ist Ausfallzeit für die Menschen, die diesen Weg benötigen, obwohl ein auf die Startseite gerichteter Healthcheck grün melden würde. Eine brauchbare Definition zählt, was Nutzer tun können, nicht ob ein Prozess läuft.

Es gibt zwei Arten von Ausfallzeit. Ein ungeplanter Ausfall wurde von niemandem gewählt: ein Festplattenausfall, eine fehlerhafte Bereitstellung, ein abgelaufenes Zertifikat, eine zuvor ausgefallene Abhängigkeit. Geplante Ausfallzeit wird vorher festgelegt, meist in einem Wartungsfenster, um etwas zu aktualisieren, zu migrieren oder zu verlagern, das sich bei laufendem Verkehr nicht ändern lässt. Mit geplanten Ausfällen kann man leichter umgehen, weil Nutzer vorgewarnt werden können. Sie zählen aber weiterhin gegen die Verfügbarkeit, sofern die Vereinbarung sie nicht ausdrücklich ausschließt.

HTTP hat Begriffe für den Ausfall selbst. RFC 9110 definiert die Statuscodeklasse 5xx für einen Server, der einen eigenen Fehler erkennt oder das Verlangte nicht ausführen kann. Drei Codes bedeuten bei einem Ausfall Verschiedenes. 502 Bad Gateway sagt, dass ein Proxy vom dahinterliegenden Server eine ungültige

Antwort erhielt. 504 Gateway Timeout sagt, dass der Proxy zu lange darauf wartete. 503 Service Unavailable ist der bewusst zu sendende Code: Die Spezifikation verknüpft ihn mit vorübergehender Überlastung oder geplanter Wartung und lässt den Server angeben, wann ein neuer Versuch sinnvoll ist:

```
HTTP/1.1 503 Service Unavailable
Retry-After: 120
```

Das sagt Browsern, API-Clients und Suchmaschinencrawlern, dass der Zustand vorübergehend ist, und empfiehlt zwei Minuten Wartezeit. Das ist eine ganz andere Botschaft als eine kaputte Seite mit Erfolgsstatus.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist Ausfallzeit?“ 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

Verfügbarkeitsziele werden anhand ihrer Neunen angegeben. Daraus lässt sich die zulässige Ausfallzeit berechnen. In einem Jahr mit 365 Tagen, also 8.760 Stunden, erlaubt jedes Ziel folgende Ausfallzeit:

- 99 %: 87,6 Stunden pro Jahr, etwa dreieinhalb Tage;
- 99,9 %: 8,76 Stunden pro Jahr oder 43,2 Minuten in einem Monat mit 30 Tagen;
- 99,95 %: 4,38 Stunden pro Jahr oder 21,6 Minuten in einem Monat mit 30 Tagen;
- 99,99 %: 52,56 Minuten pro Jahr;
- 99,999 %: etwa 5,26 Minuten pro Jahr.

Jede zusätzliche Neun verkleinert den Spielraum um den Faktor zehn, und die Kosten dafür steigen steil. Bei 99,9 % kann jemand geweckt werden, sich anmelden, das Problem finden und innerhalb des Monatsbudgets beheben. Bei

99,99 % erlaubt das ganze Jahr weniger als eine Stunde, vielleicht genug für einen Vorfall. Die Wiederherstellung muss deshalb automatisch erfolgen. Bei 99,999 % muss selbst ein gewöhnlicher Neustart für Nutzer unbemerkt bleiben.

Abhängigkeiten multiplizieren sich. Eine Anfrage, die einen Loadbalancer, eine Anwendung und eine Datenbank benötigt, die jeweils zu 99,9 % verfügbar sind und unabhängig voneinander ausfallen, wird nur bedient, wenn alle drei laufen: $0{,}999^3$ oder etwa 99,7 %. Das sind ungefähr 26 Stunden Ausfallzeit pro Jahr statt 8,76. Die Verfügbarkeit eines Dienstes kann die seiner Abhängigkeitskette nicht übersteigen, es sei denn, er entfernt Glieder daraus oder dupliziert sie.

Die Planung beginnt beim Geschäftsbetrieb. NISTs Notfallplanungsleitfaden für Bundessysteme, SP 800-34, nennt drei Zahlen, die vor einem Ausfall geklärt sein sollten. Die maximal tolerierbare Ausfallzeit (MTD) ist die gesamte Ausfalldauer, die der Systemverantwortliche akzeptiert. Das Wiederanlaufzeitziel (RTO) ist die Zeit, die ein System ausgefallen bleiben darf, bevor die Auswirkungen untragbar werden. Es muss kürzer als die MTD sein. Das Wiederherstellungspunktziel (RPO) sagt, wie weit zurück Daten aus der letzten Sicherung wiederhergestellt werden dürfen. Es misst also tolerierbaren Datenverlust statt Offlinezeit. Sind die drei festgehalten, beantworten Sicherungsplan und Failoverentwurf eine Frage, statt auf Vermutungen zu beruhen.

Wie es in der Praxis aussieht

Auf einer IPTV- und OTT-Streamingplattform mit Kunden in China, den USA und Deutschland war jede Stunde irgendwo Hauptsendezeit. Eine ruhige Wartungsstunde gab es deshalb nicht. Der Support lief rund um die Uhr für ungefähr tausend Server und eine beträchtliche Zahl kundeneigener Systeme. Ein Großteil der Arbeit bestand darin, Abweichungen vor einem Ausfall zu bemerken. Denn bei einem Livestream bedeutet ein Ausfall, dass bei jemandem das Fernsehbild stehen bleibt. Die Plattform blieb für ihr weltweites Publikum durchgehend verfügbar.

In einem mittelgroßen Unternehmen liefen zehn Datenbankserver mit PostgreSQL und SQL Server bei 99,9 % Verfügbarkeit. Nach obiger Rechnung entspricht das einem Spielraum von jeweils 8,76 Stunden pro Jahr. Hinter der Zahl standen automatisierte Sicherungen mit `pg_dump` und Jobs des SQL Server Agent samt Aufbewahrungsregeln und externer Lagerung, Monitoring in Prometheus und Grafana sowie dokumentierte Disaster-Recovery-Verfahren für zeitpunktbezogene Wiederherstellungen und Failover.

Auf dreißig Ubuntu-Servern mit ungeplant gewachsener Konfiguration wurde die Disaster Recovery mit tatsächlich getesteten Sicherungen und Wiederherstellungsverfahren neu aufgebaut. Die Netzwerkkonfiguration wurde zu einer gehärteten gemeinsamen Grundlage aller Instanzen. Das Ausfallrisiko sank, mit Wiederherstellungswegen, die schon getestet waren, bevor jemand sie brauchte.

Wo es schiefgeht

Das Monitoring liegt innerhalb des Ausfalls. Ein Healthcheck auf demselben Server oder im selben Rechenzentrum meldet weiter grün, während Nutzer auf ein abgelaufenes TLS-Zertifikat, einen kaputten DNS-Eintrag oder einen ins Leere verweisenden Loadbalancer treffen. Prüfen Sie von außen, von mehr als einem Ort und auf dem Weg eines echten Nutzers. Alarmieren Sie anhand dieser Prüfergebnisse.

Die Fehlerseite sagt 200. Ein Framework, das eine freundliche Fehlerseite mit Erfolgsstatus rendert, verbirgt den Ausfall vor Monitoring, Clients, die sonst erneut anfragen würden, und Crawlern, die die Fehlerseite statt der eigentlichen indexieren könnten. Fehler sollten einen Fehlerstatus tragen. Geplante Wartung sollte mit 503 und einem Retry-After-Header antworten.

Die Sicherung wurde nie wiederhergestellt. Eine Sicherung belegt nur, dass Bytes kopiert wurden. Solange keine Wiederherstellung ausgeführt, gemessen und mit den Daten abgeglichen wurde, bleibt die Wiederanlaufzeit geschätzt und ist oft viel länger als das RTO auf dem Papier. Stellen Sie Sicherungen nach Zeitplan in einer getrennten Umgebung wieder her und halten Sie die Dauer fest. Dasselbe gilt für Failover: Ein Standby, der nie übernommen hat, ist eine Hoffnung. Ein unter ruhigen Bedingungen geprobtes Failover hilft, Probleme vor einem Vorfall aufzudecken.

Releases verursachen die Ausfälle, die sie verhindern sollten. Eine Bereitstellung, die die alte Version stoppt, bevor die neue bereit ist, hinterlässt bei jedem Release eine Lücke. Eine Schemaänderung, die eine große Tabelle unter exklusiver Sperre umschreibt, blockiert jede Abfrage darauf bis zum Abschluss. Rollen Sie neue Instanzen aus und lassen Sie sie eine Bereitschaftsprüfung bestehen, bevor Sie die alten stoppen. Teilen Sie Schemaänderungen in einen Erweiterungsschritt und einen späteren

Rückbauschritt, sodass alter und neuer Code dazwischen funktionieren. Setzen Sie auf Kubernetes ein PodDisruptionBudget, damit das Leeren eines Knotens nicht alle Replikate zugleich herunterfahren kann.

Weiterführende Lektüre

- RFC 9110: HTTP-Semantik, Abschnitt 15.6 — die 5xx-Statuscodes einschließlich 503 und seines Retry-After-Hinweises.
- NIST SP 800-34 Rev. 1: Contingency Planning Guide for Federal Information Systems — die Definitionen von maximal tolerierbarer Ausfallzeit, RTO und RPO.
- PostgreSQL: Hochverfügbarkeit, Lastverteilung und Replikation — Standby-Server und Failover nach dem eigenen Datenbankhandbuch.
- PostgreSQL: Kontinuierliche Archivierung und zeitpunktbezogene Wiederherstellung — die Basissicherung und das WAL-Archiv für eine zeitpunktbezogene Wiederherstellung.
- Kubernetes: Unterbrechungen — freiwillige Unterbrechungen und die sie begrenzenden PodDisruptionBudgets.

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 Ausfallzeit?“ 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/downtime/