Engineer

Datensicherung

eine Datenkopie zum Schutz vor Verlust und ihr Anlegen

Was es ist

Eine Datensicherung ist das, woraus Daten wiederhergestellt werden, wenn das Original gelöscht, beschädigt oder mit seinem Rechner verloren ist. Sie muss deshalb an einem Ort liegen, den diese Ereignisse nicht erreichen. Das NIST-Glossar definiert sie als Kopie von Dateien und Programmen zur Erleichterung der Wiederherstellung. Das englische Substantiv backup, im Plural backups, bezeichnet die Kopien. Das zweiteilige Verb back up bezeichnet das Anlegen: Ein nächtlicher Auftrag sichert die Datenbank und hinterlässt eine Sicherung.

Benachbarte Dinge werden damit verwechselt. Eine Replik empfängt jede Änderung binnen Sekunden. Sie schützt vor einer defekten Platte und kopiert ein falsches `DELETE` ebenso treu. Ein Snapshot friert ein Volume zu einem Zeitpunkt ein. Zur Sicherung wird er erst dort, wo Fehler des Originals ihn nicht ebenfalls treffen. Geschichte und Trennung definieren die Sicherung: Sie bringt den früheren Datenstand von einem Ort zurück, der den Verlust des Originals überstanden hat.

Datenbanken machen das konkret. Das PostgreSQL-Handbuch beschreibt drei grundlegend verschiedene Wege. Ein SQL-Dump mit `pg_dump` schreibt die zum Wiederaufbau nötigen Anweisungen als konsistenten Stand vom Dumpbeginn. Eine Dateisystemsicherung kopiert das Datenverzeichnis. Kontinuierliche Archivierung verbindet eine solche Basissicherung mit dem später erzeugten Write-Ahead Log (WAL). Durch Wiedergabe dieses Logs lässt sich die Datenbank auf jeden Zeitpunkt seit der Basissicherung wiederherstellen. Das heißt Point-in-Time Recovery.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist eine Datensicherung?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Sicherungen, deren Funktion geprüft und nicht vorausgesetzt wird, und ein schriftlicher Plan für den Morgen, an dem ein Server nicht wieder startet.

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. 100 wichtige Sicherungen mit Barman automatisiert — 100 automatisierte geschäftskritische Datenbanksicherungen

100 wichtige Sicherungen mit Barman automatisiert Kontakt aufnehmen →

Warum es wichtig ist

Zwei Zahlen machen aus allgemeinem Sicherheitswunsch einen Entwurf. NIST definiert das Recovery Point Objective, RPO, als den Zeitpunkt, auf dessen Stand nach einem Ausfall zurückgekehrt werden muss. Praktisch ist es die Menge jüngster Arbeit, die das Unternehmen verlieren darf. Das Recovery Time Objective, RTO, beschreibt, wie lange ein System in Wiederherstellung bleiben kann, bevor der Ausfall den Organisationszweck schädigt.

Beide beeinflussen unterschiedliche Teile des Entwurfs. Das RPO hängt von der Kopierhäufigkeit ab. Ein nächtlicher Dump kann einen Tag Schreibarbeit verlieren. WAL-Archivierung begrenzt den Verlust auf die Logdaten der letzten Augenblicke, die noch auf dem Weg ins Archiv sind. Das RTO hängt von der Wiederherstellungsdauer ab. PostgreSQL spielt laut Handbuch alle WAL-Segmente seit der letzten Basissicherung nach. Bei einer alten Basis kann das dauern. Die Häufigkeit von Basissicherungen ist deshalb ebenso eine RTO- wie eine Speicherentscheidung.

Beide Zahlen behaupten etwas über Wiederherstellung. Deshalb wird eine Sicherung durch Wiederherstellen beurteilt. Exitstatus null zeigt lediglich geschriebene Bytes an irgendeinem Ort. Ob sie innerhalb des RTO wieder ein laufendes System ergeben, beantwortet regelmäßiges Wiederherstellen außerhalb der Produktion mit anschließendem Vergleich.

Trennung ist die andere Hälfte. Eine Kopie im selben Konto unter denselben Zugangsdaten teilt die Fehler, die sie überleben soll: Kontoschließung, gestohlener Schlüssel oder Skript gegen das falsche Ziel. Fragen Sie bei jeder Kopie, welches einzelne Ereignis sie zusammen mit dem Original beseitigen würde.

Wie es in der Praxis aussieht

Im kleinsten Maßstab wurde die eigene Sicherungs- und Wiederherstellungstest-Implementierung entworfen, um Wiederherstellung über einen Anbietersnapshot hinaus zu ermöglichen. Eine Ansible-Rolle installiert restic-Skripte und systemd-Units für Datenbank-Dumps bei entsprechender Konfiguration, verschlüsselte deduplizierte Snapshots, Bereinigung nach Aufbewahrungsregeln und Integritätsprüfungen. Ein eigener monatlicher Ablauf stellt eine bekannte Datei an einem temporären Ort wieder her und vergleicht ihre Bytes. Sicherung und Wiederherstellung haben getrennte Erfolgssignale. Die Timer brauchen ein konfiguriertes Repository. Die Passphrase kommt aus einer geschützten Datei statt aus einem von systemd interpretierten Umgebungswert. Das sind implementierte Kontrollen und Aktivierungsvoraussetzungen. Der Stand vom 2026-10-27 hält fest, dass die Timer ohne Repository deaktiviert waren, und dokumentiert keinen abgeschlossenen Wiederherstellungslauf in einer realen Umgebung. Er belegt die Automatisierungsarbeit, aber weder den Erfolg einer Wiederherstellung noch, dass der Host bereits gefahrlos verloren gehen kann.

In größerem Maßstab wurden hundert kritische Datensicherungen mit Barman und Google Cloud automatisiert für ein auf seine GIS-Daten angewiesenes Produkt. Barman übernahm PostgreSQL, Google Cloud die externen Kopien. Bash und Python steuerten und validierten die Läufe. Integritätsprüfungen bestätigten nicht bloß Vorhandensein, sondern brauchbare Sicherungen.

Bei einem dritten Produkt wurde die wiederherstellbare Ausgangslage definiert statt angenommen. Die Aufbewahrung steht im Bicep-Code neben der Datenbank. Point-in-Time Restore ist damit Vorlageneigenschaft statt früherer Portalklick. Auch die umgebende Infrastruktur ist Code. Das Audit dokumentierte die Lücken: Georedundanz ist ausgeschaltet, das RTO vorgeschlagen statt gemessen. Die Messübung ist beschrieben und wartet auf ihre Ausführung.

Wo es schiefgeht

Sicherungsprobleme bleiben häufig bis zum nötigen Restore verborgen. Deshalb lohnt es sich, an einem ruhigen Tag nach jedem zu suchen.

Eine Dateikopie der laufenden Datenbank ergibt einen defekten Cluster. `tar` über das aktive PostgreSQL-Datenverzeichnis sammelt unterschiedliche Zeitpunkte. Laut Handbuch muss der Server für eine brauchbare Sicherung auf diesem Weg abgeschaltet sein. Ein atomarer Dateisystem-Snapshot funktioniert,

wenn er das WAL einschließt. Beim Start spielt der Server es wie nach einem Absturz nach. Liegen Daten und WAL auf verschiedenen Dateisystemen, kann ein gleichzeitiger Snapshot unmöglich sein. `pg_dump` oder eine Basissicherung für kontinuierliche Archivierung erzeugt dagegen bei laufendem Server eine konsistente Kopie.

Eine Lücke im WAL-Archiv beendet die zeitpunktgenaue Wiederherstellung an dieser Lücke. Benötigt wird eine ununterbrochene Folge archivierter WAL-Dateien bis zurück zum Beginn der Basissicherung. Scheitert der Archivbefehl fortlaufend, versucht PostgreSQL es weiter. WAL häuft sich in `pg_wal/`, sodass oft eine sich füllende Platte das erste Symptom ist. Das Handbuch empfiehlt, Archivierung vor der ersten Basissicherung einzurichten und zu testen sowie Verzögerungen zu überwachen.

Der Auftrag stoppt, Schweigen gilt als Erfolg. Ein defekter Timerauftrag schreibt seine Fehlermeldung in ein Protokoll, das niemand liest. Jeder stille Tag wirkt gut. Überwachung muss Schweigen als Alarm behandeln: ein nur nach Erfolg gesendetes Lebenszeichen an einen bei Verspätung alarmierenden Dienst. Das entdeckt einen ganz gestoppten Auftrag ebenso wie einen laut scheiternden.

Der Schlüssel fehlt oder kommt verändert an. Bei der Verschlüsselung hängt der Zugriff von den Schlüsseln ab. restic unterstützt mehrere Passwörter für ein Repository; die Warnung vor unwiederbringlichem Datenverlust gilt, wenn kein nutzbarer Repository-Schlüssel mehr vorhanden ist. Der Schlüssel muss den Rechner überleben. Im genannten Aufbau hätte die Veränderung vom Dienstmanager selbst kommen können. Er interpretiert Escape-Sequenzen in Umgebungswerten. Eine so übergebene Passphrase mit Backslash wäre deshalb mit anderen Bytes bei restic angekommen als beim Anlegen des Repositorys, ohne Warnung. Das Lesen aus einer Datei schließt diesen Weg.

Die Aufbewahrung entfernt die nötige Kopie. Beschädigung fällt oft erst Wochen später auf. Eine Regel, die sieben tägliche Kopien aufbewahrt, hat dann alle sauberen Stände durch beschädigte ersetzt. Eine gestaffelte Regel erhält günstig einen langen Verlauf. restics `--dry-run` zeigt geplante Löschungen vorab:

```
restic forget --keep-daily 7 --keep-weekly 5 --keep-monthly 12 \
  --prune --dry-run
```

`forget` entfernt Snapshots, `prune` die ausschließlich von ihnen referenzierten Daten. Nach dem Prune sind die alten Kopien endgültig weg.

Die Integritätsprüfung besteht, der Restore scheitert. Standardmäßig prüft `restic check` die Struktur, ohne alle Pack-Dateien zu lesen, weil dafür alle heruntergeladen werden müssten. `--read-data` liest alles, `--read-data-subset=1/5` ein Fünftel. Fünf Läufe von `1/5` bis `5/5` decken den Bestand ab. Vollständiges Lesen beweist nur unbeschädigte gespeicherte Bytes. Ob sie eine funktionierende Datenbank mit richtigen Rollen und Einstellungen ergeben, zeigt die Wiederherstellung.

Weiterführende Lektüre

- NIST-Glossar: Backup — die hier verwendete Definition und ihre Veröffentlichungsquellen.
- NIST SP 800-34 Rev. 1 — der Notfallplanungsleitfaden mit RPO- und RTO-Definitionen.
- PostgreSQL: Sicherung und Wiederherstellung — die drei Verfahren und ihr jeweiliger Aufwand.
- PostgreSQL: Kontinuierliche Archivierung und Point-in-Time Recovery — Basissicherungen, WAL-Archivierung und Rückkehr zu einem gewählten Zeitpunkt.
- restic: Ein neues Repository vorbereiten — Passwörter, Passwortdateien und die Warnung vor Verlust.
- restic: Sicherungssnapshots entfernen — Aufbewahrungsregeln, `forget`, `prune` und `--dry-run`.

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 eine Datensicherung?“ 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/backup/