Engineer

Migration

🚚 geplanter Umzug auf eine neue Version oder Plattform

Was es ist

Eine Migration ist der geplante, schrittweise Umzug von etwas bereits Genutztem auf eine neue Version oder Plattform. Am Ende wird die alte stillgelegt. Umziehen können eine ganze Arbeitslast, eine Anwendung, ein Datenbestand oder die Struktur, in der die Daten liegen. Von einem Neuaufbau unterscheidet sich eine Migration dadurch, dass das System während des Umzugs genutzt wird: Nutzer, Datensätze und Integrationen ziehen mit, und die alte Version arbeitet weiter, bis die neue übernommen hat.

Fachleute verwenden den Begriff auf zwei Ebenen, oft ohne zu sagen, welche gemeint ist.

Im Großen ist es der Wechsel auf eine neue Plattform oder Hauptversion. Microsofts Cloud Adoption Framework behandelt den Umzug von Arbeitslasten nach Azure, von lokalen Servern oder aus einer anderen Cloud, in fünf Schritten: planen, die Arbeitslasten vorbereiten, ausführen, optimieren und das Ausgangssystem stilllegen. Esris Kunden ziehen ihre Karten und Daten von ArcMap nach ArcGIS Pro um. Eine Next.js-Anwendung wechselt vom Pages Router zum App Router, und `getServerSideProps` weicht dem Datenabruf in Server Components. Ein PostgreSQL-Cluster wechselt auf eine neue Hauptversion, die häufig den Aufbau der Systemtabellen ändert. Die Daten werden deshalb mit `pg_upgrade` oder durch einen Dump mit anschließender Wiederherstellung übertragen.

Im Kleinen ist eine Migration eine einzelne versionierte Änderung an der Datenbankstruktur. Sie liegt als Datei im Repository und wird von einem Werkzeug in einer festen Reihenfolge angewandt. Django nennt seine Migrationen ein Versionskontrollsystem für das Datenbankschema: Jede Datei hält eine Änderung an den Modellen fest, wird mit dem Code eingecheckt und

läuft auf dem Entwicklungslaptop, in der Staging-Umgebung und in Produktion in derselben Reihenfolge. GitLab liefert in Batches aufgeteilte Hintergrundmigrationen, die große Tabellen nach einem Upgrade Batch für Batch umschreiben, etwa indem sie Ganzzahlspalten vor einem Überlauf auf `bigint` erweitern. Auch eine Datenmigration gehört hierher: Sie ändert die Zeilen statt der Spalten, etwa indem sie ein neues Feld aus zwei alten füllt.

Es ist dieselbe Idee in zwei Größenordnungen. Beide haben einen bekannten Ausgangszustand, eine geordnete Folge von Schritten, ein Ziel und einen Weg zurück, wenn ein Schritt schiefgeht. Ein Plattformwechsel enthält gewöhnlich Schemamigrationen. Eine lange Folge von Schemamigrationen wiederum ist über die Jahre eine Plattform, die sich unter ihren Nutzern Stück für Stück verändert.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist eine Migration?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Daten aus einer veralteten Lösung in eine unterstützte Umgebung überführen, während der Geschäftsbetrieb weiterläuft.

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. GIS-Karte mit PostgreSQL 10-mal schneller — 10x schnellere Karten nach dem Wechsel von MSSQL

GIS-Karte mit PostgreSQL 10-mal schneller Kontakt aufnehmen →

Warum es wichtig ist

Software überlebt die Grundlage, auf der sie gebaut wurde. Cloud-Verträge werden neu verhandelt, Frameworks ändern ihr Modell und Anbieter stellen Produkte ein. Ein System, das ein Jahrzehnt genutzt wird, wird in dieser Zeit mehrmals migriert. Wie gut ein Team migriert, wird deshalb am Ende wichtiger als die Wahl der ersten Plattform.

- **Der geteilte Zustand verursacht die Kosten.** Solange ein Teil des Systems auf der alten und ein Teil auf der neuen Plattform läuft, müssen beide betrieben, überwacht und konsistent gehalten werden. Das Cloud Adoption Framework rät, diese Zeit kurz zu halten und unmittelbar voneinander abhängige Komponenten in derselben Welle umzuziehen.

- **Eine Datenbank lässt sich nicht neu bereitstellen.** Code kann neu gebaut und ersetzt werden, die Produktionsdaten nicht. Schemamigrationen machen aus einer Änderung der laufenden Struktur eine Datei, die sich prüfen, an einer Kopie testen, geordnet anwenden und bei den meisten Änderungen rückgängig machen lässt. Die Alternative ist ein Produktionsschema, dessen Zustand davon abhängt, was die letzte Person mit Konsolenzugang hinterlassen hat.
- **Ein geplanter Umzug lässt sich rückgängig machen.** Das Cloud Adoption Framework fordert dazu auf, Rollback-Kriterien und -Verfahren vor Beginn einer Migration festzulegen, den Rollback zu testen und unterwegs Prüfpunkte für Fortfahren oder Abbrechen vorzusehen.

  Die Abwägungen liegen in der Umschaltung. Das Cloud Adoption Framework unterscheidet Migrationen mit Ausfallzeit von solchen mit nahezu keiner Ausfallzeit. Bei Letzteren werden die Daten laufend repliziert und erst am Ende wird umgeschaltet. Beim Wechsel auf eine neue PostgreSQL-Hauptversion verkürzt `pg_upgrade` die Umschaltung: Es verwendet die vorhandenen Datendateien weiter und überspringt damit den sonst typischen Dump samt Wiederherstellung. Im Modus `--link`, der keine Dateien kopiert, ist es nochmals deutlich schneller.

Wie es in der Praxis aussieht

Drei Projekte auf dieser Website zeigen den Begriff in drei Größenordnungen, von einer ganzen Plattform bis zur Reihenfolge der Dateien in einem Repository.

Eine Migration führte weg von einem Cloud-Anbieter. Die AWS-Umgebung eines Kunden war überdimensioniert. Zunächst wurde geprüft, was tatsächlich genutzt wurde: Unnötige Dienste entfielen, die verbleibende Kapazität wurde angepasst. Danach verließen die Arbeitslasten AWS. Der Auftragsnachweis belegt diese Infrastrukturänderungen. Ungenutzte Dienste vor dem Umzug zu entfernen bedeutete, dass weniger zu übertragen war. Daraus lassen sich weder ein Kostenverhältnis bei gleichem Leistungsumfang noch ein Produktivitätsergebnis oder ein störungsfreier Umzug ableiten.

Der zweite Fall war ein Wechsel des Datenbanksystems mit einem neuen Entwurf am Ziel. Die Datenbank einer geografischen Kartenanwendung wechselte von MSSQL zu PostgreSQL mit der Erweiterung PostGIS. Das Ziel war keine Kopie der alten Struktur. Das neue Schema wurde für Geodaten entworfen, mit GiST- und SP-GiST-Indizes sowie strengen Fremdschlüssel- und Prüfbedingungen, die Validierungsregeln für räumliche Koordinaten durchsetzen. Über 50 Millionen Geodatensätze gelangten über ETL-Pipelines hinein, die sie nach EPSG:4326 transformierten. Der Umzug auf einen neuen Entwurf bietet die Gelegenheit, die

Regeln der Daten im Schema selbst festzuhalten. Die Seite berichtet von einer 10-fachen Verbesserung der Leistung räumlicher Abfragen und einer um 65 % gesunkenen Backend-CPU-Last.

Der dritte Fall ist die kleine Größenordnung, über Jahre beibehalten. Das Schema einer Plattform ist die Summe aus 1.022 Migrationen, nummeriert von 036 bis 1102. Die Seite nennt die Disziplin ihrer Reihenfolge unspektakulär und nicht verhandelbar. Die Historie im Repository beschreibt die Produktion. Wahr bleibt sie nur, solange sich jede Datenbank, von der frischen Entwicklungskopie bis zur Wiederherstellung, durch erneutes Anwenden von der ersten bis zur letzten Datei aufbauen lässt.

Wo es schiefgeht

Die Planung konzentriert sich auf die Umschaltung. Die Schwierigkeiten sammeln sich meist in den Wochen darum herum, in vier erkennbaren Formen.

Zwei Systeme laufen länger als geplant nebeneinander. Der Umzug bleibt auf halbem Weg stecken, die schwierigen Teile bleiben übrig und das Team betreibt beide Plattformen monatelang. Nutzer spüren die Naht: Laut Next.js-Leitfaden ist der Wechsel zwischen einer Pages-Router-Seite und einer App-Router-Seite eine vollständige Seitennavigation; `next/link` lädt über diese Grenze hinweg nicht vor. Halten Sie die Wellen klein und den jeweils nächsten Schritt günstig. Ein Frontend hier zieht seine Seiten auf eine gemeinsame Factory um, die jede Seite in einem einzigen Anfrage-Antwort-Zyklus lädt. Jede umgestellte Seite übernimmt damit dieses Verhalten.

Eine Umschaltung ohne Weg zurück. Manche Schritte ändern neben dem Ziel auch die Quelle. Bei `pg_upgrade --link` schreibt der neue Cluster nach seinem ersten Start in Dateien, die er mit dem alten teilt. Der alte Cluster muss dann aus einer Sicherung wiederhergestellt werden. Esri warnt, dass ein für ArcGIS Pro aktualisierter Datenbestand in ArcMap unbrauchbar werden kann, und rät, ihn vorher zu kopieren. In Django löst eine `RunSQL` -Operation ohne umgekehrtes SQL einen `IrreversibleError` aus, sobald jemand rückwärts migriert. Legen Sie die Rollback-Kriterien vor dem ersten Schritt fest, üben Sie den Rückweg und machen Sie daraus eine Prüfung: Ein Repository hier führt eine Prüf-Engine aus, zu deren Validatoren eine Prüfung, ob eine Migration rückgängig gemacht werden kann gehört.

Eine bereits ausgeführte Migration wird geändert. Migrationswerkzeuge halten fest, welche Dateien angewandt wurden, und überspringen sie danach. Ändern Sie eine Datei nach ihrer Anwendung in Produktion, behält die Produktion die alte Fassung, während jede frische Datenbank die neue erhält: zwei Schemata unter einer Historie. Lassen Sie angewandte Migrationen unverändert und korrigieren Sie sie mit einer neuen. Ein Build, der die Historie von einer leeren Datenbank aus erneut anwendet und das Ergebnis mit einer migrierten Kopie vergleicht, erkennt diese Abweichung.

Die aktualisierte Datenbank ist am ersten Tag langsam. `pg_upgrade` übernimmt die meisten Planerstatistiken, aber nicht alle. Beim Rest muss der Planer raten, bis sie neu aufgebaut sind. Der Modus `--check` ändert nichts und kann laufen, während der alte Server weiter Anfragen bedient. Führen Sie ihn deshalb zuerst aus und erstellen Sie direkt nach dem Upgrade zunächst die fehlenden Statistiken neu, bevor Sie die übrigen erfassen:

```
pg_upgrade --check -b "$OLD/bin" -B "$NEW/bin" -d "$OLD/data" -D "$NEW/data"
pg_upgrade -b "$OLD/bin" -B "$NEW/bin" -d "$OLD/data" -D "$NEW/data"
vacuumdb --all --analyze-in-stages --missing-stats-only
vacuumdb --all --analyze-only
```

Weiterführende Lektüre

- Die Migration planen — Microsofts Methode für den Umzug von Arbeitslasten nach Azure: Wellen, Umschaltvarianten und Rollback-Planung.
- Migrationen — Djangos Anleitung zu versionierten Schemaänderungen, ihrer Umkehrung und Datenmigrationen.
- pg_upgrade — das PostgreSQL-Upgrade-Werkzeug mit Prüfmodus, Link-Modus, Rückkehr und den anschließenden Statistikschritten.
- Vom Pages Router zum App Router migrieren — eine Next.js-Anwendung Seite für Seite auf den neuen Router umstellen.
- Vorhandene Arbeit migrieren — was von ArcMap nach ArcGIS Pro mitkommt, was konvertiert werden muss und was zuerst zu sichern ist.
- Migrationen vor einem Upgrade prüfen — wie GitLabs Hintergrundmigrationen große Tabellen nach einem Upgrade stapelweise umschreiben und wie sich ihr Abschluss feststellen lässt.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Bringen Sie einen repräsentativen Datensatz, die Fragen, die er beantworten muss, und den aktuellen Engpass mit. Wir können vereinbaren, wie Qualität, Durchsatz und Wiederherstellung gemessen werden.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist eine Migration?“ 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/migration/