Engineer

Migrering

en planlagt overgang til en ny version eller platform

Hvad det er

En migrering er en planlagt flytning af noget, der allerede er i brug, til en ny version eller en ny platform. Den sker i trin, og til sidst bliver det gamle lukket ned. Det, der flyttes, kan være en hel workload, en applikation, en mængde data eller den struktur, dataene ligger i. Forskellen fra en genopbygning er, at systemet bliver ved med at være i brug undervejs: brugere, data og integrationer følger med, og den gamle version arbejder videre, indtil den nye har taget over.

Ordet bruges om arbejde i to størrelser, ofte uden at nogen siger, hvilken af dem der er tale om.

I den store skala er det en flytning til en ny platform eller en ny hovedversion. Microsofts Cloud Adoption Framework beskriver det at flytte workloads til Azure, fra egne servere eller fra en anden cloud, som fem trin: planlæg, forbered workloads, gennemfør, optimer og afvikl kilden. Esris kunder flytter deres kort og data fra ArcMap til ArcGIS Pro. En Next.js-applikation flytter fra Pages Router til App Router, hvor `getServerSideProps` afløses af datahentning inde i Server Components. En PostgreSQL-klynge flytter til en ny hovedversion, som ofte ændrer opbygningen af systemtabellerne, så dataene skal føres over med `pg_upgrade` eller med et dump og en gendannelse.

I den lille skala er en migrering én versioneret ændring af en databases struktur, gemt som en fil i repositoryet og udført af et værktøj i en fast rækkefølge. Django kalder sine migreringer et versionsstyringssystem for databaseskemaet: hver fil beskriver en ændring af modellerne, bliver committet sammen med koden og køres i samme rækkefølge på udviklerens maskine, på staging og i produktion. GitLab leverer batchede baggrundsmigreringer, der omskriver store tabeller en portion ad gangen efter en opgradering, for eksempel når

heltalskolonner gøres til `bigint`, før de løber over. En datamigrering hører også hjemme her: den ændrer rækkerne frem for kolonnerne, for eksempel ved at udfylde et nyt felt ud fra to gamle.

Det er én idé i to størrelser. Begge har en kendt starttilstand, en ordnet række trin, et mål og en vej tilbage, hvis et trin går galt. En platformsflytning rummer som regel skemamigreringer, og en lang række skemamigreringer er med årene en platform, der skifter stykke for stykke under dem, der bruger den.

Brug begrebet i en beslutning

Brug »Hvad er en migrering?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Data taget ud af noget aldrende og ind i noget, der er understøttet, mens forretningen kører videre.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har forbedret GIS-kortydeevne 10x med PostgreSQL — 10x hurtigere kort, efter skiftet fra MSSQL

Har forbedret GIS-kortydeevne 10x med PostgreSQL Kontakt os →

Hvorfor det betyder noget

Software lever længere end det, den er bygget på. Cloudkontrakter bliver genforhandlet, frameworks skifter model, og leverandører udfaser produkter. Et system, der skal leve i ti år, bliver migreret flere gange i løbet af de ti år, så hvor godt et team migrerer, ender med at betyde mere end hvor godt det valgte den første platform.

- **Mellemstadiet er der, regningen lander.** Mens en del af systemet kører på den gamle platform og en del på den nye, skal begge drives, overvåges og holdes i takt. Cloud Adoption Framework råder til at holde den periode kort og til at flytte komponenter, der afhænger direkte af hinanden, i samme bølge.
- **En database kan ikke bare udrulles igen.** Kode kan bygges igen og skiftes ud; det kan rækkerne i produktion ikke. Skemamigreringer gør en ændring af strukturen i en kørende database til en fil, der kan reviewes, testes mod en kopi, køres i rækkefølge og for de fleste ændringers vedkommende rulles tilbage. Alternativet er et produktionsskema, der ser ud, som den sidste person med en konsol efterlod det.
- **En planlagt flytning kan fortrydes.** Cloud Adoption Framework beder om, at kriterier og procedurer for tilbagerulning ligger fast, før en migrering begynder, at tilbagerulningen er testet, og at der er go/no-go-kontrolpunkter undervejs.

Afvejningerne ligger i selve omstillingen. Cloud Adoption Framework skelner mellem migrering med nedetid og migrering med næsten nul nedetid, hvor data replikeres løbende, og skiftet sker til sidst. Ved en ny hovedversion af PostgreSQL forkorter `pg_upgrade` omstillingen: værktøjet springer det dump og den gendannelse over, som en sådan opgradering typisk kræver, ved at genbruge de eksisterende datafiler, og i `--link` -tilstand, hvor intet kopieres, går det langt hurtigere.

Sådan ser det ud i praksis

Tre projekter på dette site viser begrebet i tre størrelser, fra en hel platform ned til rækkefølgen af filerne i ét repository.

En migrering var en flytning væk fra en cloududbyder. En kundes AWS-opsætning var overdimensioneret, og arbejdet begyndte med en gennemgang af, hvad der faktisk blev brugt: unødvendige tjenester forsvandt, og den resterende kapacitet blev tilpasset. Derefter flyttede kundens workloads væk fra AWS. Beskrivelsen af engagementet underbygger de infrastrukturændringer. Fordi ubrugte tjenester blev fjernet før flytningen, var der mindre at bære; beskrivelsen dokumenterer ikke et omkostningsforhold opgjort på sammenligneligt grundlag, et produktivitetsresultat eller en migrering uden driftsforstyrrelser.

Det andet var et skift af databasemotor med et nyt design i den anden ende. En geografisk kortapplikations database flyttede fra MSSQL til PostgreSQL med udvidelsen PostGIS, og målet var ikke en kopi af den gamle struktur. Det nye skema blev designet til spatiale data, med GiST- og SP-GiST-indeks og strenge foreign key- og check-constraints, der håndhæver valideringsregler for de spatiale koordinater. Over 50 millioner spatiale poster flyttede ind gennem ETL-pipelines, der transformerede dem til EPSG:4326. En flytning til et nyt design er chancen for at skrive reglerne for dataene ind i selve skemaet. Siden melder om 10 gange bedre ydeevne for spatiale forespørgsler og en CPU-belastning i backend, der faldt med 65 %.

Det tredje er den lille skala, holdt ved lige i årevis. En platforms skema er summen af 1.022 migreringer, nummereret fra 036 til 1102, og siden kalder disciplinen om deres rækkefølge kedelig og ikke til forhandling. Historikken i repositoryet er en beskrivelse af produktion, og den holder kun stik, så længe enhver database, fra en udviklers friske kopi til en gendannelse, kan bygges op igen ved at afspille den fra første til sidste fil.

Hvor det går galt

Omstillingen får planlægningen, og problemerne samler sig gerne i ugerne omkring den, i fire genkendelige former.

To systemer kører side om side længere end planlagt. Flytningen går i stå halvvejs, de stykker, der er tilbage, er de svære, og teamet driver begge platforme i månedsvis. Brugerne mærker sømmen: Next.js-guiden gør opmærksom på, at et skift mellem en side under Pages Router og en side under App Router er en hård navigation, og at `next/link` ikke prefetcher på tværs af de to. Hold bølgerne små, og gør hvert næste trin billigt. Én frontend her flytter sine sider over på en fælles fabrik, der henter hver side på én rundtur, så hver side, der flyttes, arver opførslen.

En omstilling uden vej tilbage. Nogle trin ændrer kilden såvel som målet. Med `pg_upgrade --link` har den nye klynge, så snart den er startet, skrevet i filer, som den gamle deler, og den gamle klynge må derefter gendannes fra backup. Esri advarer om, at et datasæt, der er opgraderet til ArcGIS Pro, måske ikke længere kan bruges i ArcMap, og beder brugerne tage en kopi først. I Django udløser en `RunSQL` -operation uden omvendt SQL en `IrreversibleError` , i det øjeblik nogen migrerer baglæns. Fastlæg kriterierne for tilbagerulning før første trin, øv vejen tilbage, og gør den til et tjek: ét repository her har en kontrolmotor, hvis validatorer blandt andet efterprøver, at en migrering kan rulles tilbage.

En migrering, der allerede er kørt, bliver redigeret. Migreringsværktøjer registrerer, hvilke filer der er udført, og springer dem over fra da af. Ændrer man en fil, efter at den har nået produktion, beholder produktion den gamle udgave, mens hver ny database får den nye: to skemaer under én historik. Lad udførte migreringer være, og ret dem med en ny; et build, der afspiller hele historikken fra en tom database og sammenligner resultatet med en migreret kopi, fanger afvigelsen.

Den opgraderede database er langsom første dag. `pg_upgrade` fører de fleste af planlæggerens statistikker over, men ikke alle, og planlæggeren gætter på resten, indtil de er bygget op igen. Tilstanden `--check` ændrer intet og kan køre, mens den gamle server stadig betjener brugere, så kør den først, og genopbyg de manglende statistikker lige efter opgraderingen, før resten indsamles:

```
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
```

Læs mere

- Plan your migration — Microsofts metode til at flytte workloads til Azure: bølger, valg af omstilling og planlægning af tilbagerulning.
- Migrations — Djangos vejledning i versionerede skemaændringer, i at rulle dem tilbage og i datamigreringer.
- pg_upgrade — PostgreSQL's opgraderingsværktøj med check-tilstand, link-tilstand, tilbagerulning og statistiktrinnene bagefter.
- How to migrate from Pages to the App Router — sådan flyttes en Next.js-applikation til den nye router én side ad gangen.
- Migrate existing work — hvad der følger med fra ArcMap til ArcGIS Pro, hvad der skal konverteres, og hvad der bør kopieres først.
- Check migrations before upgrade — hvordan GitLabs batchede baggrundsmigreringer omskriver store tabeller efter en opgradering, og hvordan man ser, at de er færdige.

Ordliste

Vælg ud fra jeres faktiske vilkår

Medbring et repræsentativt datasæt, de spørgsmål det skal besvare, og den aktuelle flaskehals. Vi kan aftale målinger af kvalitet, gennemløb og gendannelse.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er en migrering?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/migration/