**Engineer**

# Tilbakerulling av en utrulling

å gjenopprette en tidligere utrulling etter en feil

## Hva det er

Tilbakerulling av en utrulling fører tjenesten tilbake til en tidligere installert versjon eller konfigurasjon når en endring skaper problemer. Den konkrete fremgangsmåten avhenger av hva utrullingssystemet styrer. Det kan sende trafikken tilbake til en versjon som fortsatt kjører. Det kan også starte nye prosesser med en tidligere revisjon. Når Kubernetes ruller tilbake en Deployment, gjenoppretter den podmalen fra en lagret revisjon. Andre deler, som databasen, går videre på sin egen tidslinje.

En reversering i kodelageret lager en ny revisjon som opphever virkningen av en tidligere endring. Den nye koden må fortsatt bygges og installeres i en kjørende tjeneste. En tilbakerulling kan aktivere en pakke man allerede har bygget og beholdt. Etter et nødgrep må teamet samordne det som faktisk kjører, med versjonen den ordinære leveranseprosessen forventer.

En tidligere programversjon kan møte rader, hendelser og filer som ble skrevet av den nyere versjonen. Hvis den fortsatt kan lese dem, kan teamet beholde brukernes siste arbeid. En gjenoppretting av databasen til et bestemt tidspunkt er en egen operasjon, med konsekvenser for innholdet og egne kontrollbehov. Avklar hva som skal skiftes, og hva som skal bevares.

### Bruk begrepet i en beslutning

Bruk «Hva er tilbakerulling av en utrulling?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Testet, utgitt og mulig å reversere, slik at endringer når kundene en tirsdag ettermiddag i stedet for sent en fredagskveld.

**Se tjenestene våre** Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Automatiserte GIS-SaaS-utrulling i GitHub Actions — produksjonssetting, data og rapportering, alt uten manuelle grep

**Automatiserte GIS-SaaS-utrulling i GitHub Actions** Ta kontakt →

## Hvorfor det er viktig

Ved en feil etter en utrulling må teamet vurdere hva som vil få tjenesten til å fungere igjen. En kjent tidligere versjon og en prøvd måte å flytte trafikken på gjør tilbakerulling til ett mulig grep. Det kan gå fortere enn å utvikle en ny rettelse mens feilene pågår. Å bytte program kan stoppe nye feil, mens data og handlinger som allerede er berørt, fortsatt kan trenge egen oppfølging. Før en beslutning er det nyttig å beskrive akkurat hvilken del som skal gjenopprettes.

En startet prosess er bare ett tegn på gjenoppretting. Sjekk hva brukerne faktisk kan gjøre. Ved en betalingsfeil kan en kontroll med en egen testkonto gi konkrete opplysninger om hvorvidt det riktige svaret kommer fra riktig versjon. Mål også enkelte bakgrunnsprosesser og køer. Hvis den opprinnelige feilen bare oppstod for en viss trafikk, må observasjonen omfatte slik trafikk eller et egnet kontrollanrop.

Behold nødvendig revisjonshistorikk, pakker og tilgang til konfigurasjon. En revisjon kan være synlig selv når bildet den viser til, ikke lenger finnes. En hemmelighet kan også ha utløpt. Test en gjenoppretting med de samme rettighetene den som har vakt vil få. Øvelsen bør gi et svar på hvorvidt en versjon fortsatt er mulig å bruke. Noter hvem som har tilgang til reserveløsningen og hvor de finner den hvis hovedverktøyet er nede.

## Slik ser det ut i praksis

Porteføljens sak om repeterbare utrullinger til Azure beskriver en fast vei via utvikling, test og produksjon. En slik rute gir et utgangspunkt for å planlegge gjenoppretting. Teamet vet hvilke miljøer en versjon skal prøves i, og hva som er forventet før produksjonstrinnet. Saken gjelder utrullingen. Eksempelet nedenfor illustrerer tilbakerulling under et annet, tenkt forløp.

Anta at versjon B av en bestillingstjeneste feiler for noen kunder med lagret adresse. Versjon A virket kort tid før, og pakke og konfigurasjon er tilgjengelige. Vurder først om B allerede har skrevet data som A ikke forstår. Støtter begge

versjoner dagens skjema og innhold, kan teamet sende trafikken tilbake til A. Gjennomfør deretter en kontrollbestilling og bekreft i loggen hvilken versjon som svarte. Følg også med på hendelser og feil i bakgrunnskøen.

Hvis B har fjernet en kolonne A må bruke, kan en umiddelbar tilbakerulling gi flere feil. Det kan være tryggere å stanse de berørte skrivingene og installere en kompatibilitetsrettelse fremover. Ved en planlagt endring kan teamet først legge til ny struktur, flytte lesing og skriving trinnvis og vente med å slette den gamle kolonnen. Test at begge versjoner virker på mellomstadiet mens den gamle fortsatt kan trenges.

## Hvor det går galt

En kommando kan gi en vellykket status lenge før kundene merker en forbedring. En styring kan ha godtatt malen, mens prosessene fortsatt starter opp. Sjekk hvilken applikasjon som faktisk tar imot trafikken og om den avtalte funksjonstesten passerer. Deretter må teamet se etter den opprinnelige feilen lenge nok til å kunne konkludere. Dukker feilen opp igjen, må årsaken vurderes på nytt.

Hvilke brukere som får en ny funksjon, kan styres separat fra programversjonen. En bryter, en trafikkregel eller en planlagt jobb kan holde en ny oppførsel aktiv selv etter at bildet er byttet tilbake. Kartlegg disse kontrollene. Lag en rekkefølge for hva som bør byttes, hvem som kan gjøre det, og hva som skjer med klienter som fortsatt sender forespørsler i det nye formatet.

En styring som kontinuerlig følger den deklarerte versjonen, kan også rulle frem den samme feilen igjen. Endre ønsket tilstand på en kontrollert måte samtidig som tjenesten gjenopprettes. Noter de berørte revisjonene og tidspunktene, og hvilke følger brukerne opplevde. Bruk den vanlige leveranseveien med tilhørende tester når en ny versjon er klar.

## Les mer

- Kubernetes om Deployments og tilbakerulling forklarer revisjoner og hva podmalen styrer.
- DORAs leveransemålinger skiller mellom flyt, feil og gjenoppretting.
- PostgreSQL om endring av tabeller viser hvordan tabellstrukturen kan endres.
- Helm rollback beskriver valg av tidligere revisjon.

Ordliste

Velg ut fra deres faktiske rammer

Ta med utrullingsløpet, driftsrisikoene og dem som har vakt. Vi kan avgrense en endring med kontroller, en vei tilbake og tydelig ansvar.

**Se hva det koster** Ta kontakt →

# Gjør begrepene praktiske

Spør hvordan «Hva er tilbakerulling av en utrulling?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

**Se tjenestene våre**

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/deployment-rollback/