Engineer

Git

versjonskontrollen dette nettstedets historikk ligger i

Hva det er

Git er et distribuert versjonskontrollsystem: et program som registrerer tilstanden til et sett filer som en kjede av øyeblikksbilder, slik at enhver tidligere tilstand kan ses på, sammenlignes med nåtiden eller hentes tilbake. Prosjektet skriver navnet som Git og kommandoen som `git`, og begge betegner det samme verktøyet.

Hvert øyeblikksbilde er en commit. En commit lagrer hvordan hver sporet fil så ut i det øyeblikket, hvem som gjorde endringen og når, en melding som forklarer hvorfor, og en peker til den eller de commitene som kom før. En fil git allerede har sett, lagres bare én gang: er filen uendret mellom to commits, peker det nye øyeblikksbildet på kopien som allerede ligger på disken, i stedet for å skrive den på nytt. Derfor tar en lang historikk langt mindre plass enn en full kopi per commit ville gjort.

Alt inni er navngitt etter innholdet. Før git lagrer en fil, en katalogoversikt eller en commit, regner den ut en sjekksum over bytene, i det klassiske formatet en SHA-1-hash skrevet som 40 heksadesimale tegn, og den sjekksummen blir objektets navn. Endrer du én enkelt byte, endres navnet med den, og det samme gjør navnet på hver commit som er bygd oppå. Slik oppdager git skadde data, og derfor peker en commit-id fortsatt på nøyaktig den samme koden mange år senere.

Distribuert betyr at hver kopi er komplett. Når du kloner et repo, får du hele historikken og ikke bare de nyeste filene, så å lese loggen, sammenligne to versjoner eller lage en commit skjer på den lokale disken uten at noen server er involvert. Å dele arbeid er et eget, bevisst steg: du pusher commitene dine til en

annen kopi, eller du henter commits fra en og fletter dem inn. Hvilken som helst kopi kan fungere som server. I de fleste team spiller én avtalt kopi, ofte hos en vertstjeneste, den rollen etter konvensjon og ikke fordi verktøyet krever det.

En arbeidsdag beveger seg gjennom tre steder. Arbeidstreet holder filene du redigerer. Oppsamlingsområdet, også kalt indeksen, holder det den neste commiten skal inneholde, slik at du kan committe en del av endringene og ta vare på resten til senere. Selve git-katalogen, `.git` øverst i prosjektet, holder objektdatabasen og alle referansene. En gren er bare et flyttbart navn som peker på én commit, og derfor går det på et øyeblikk å opprette en.

Git ble skrevet i 2005 av Linus Torvalds og miljøet rundt Linux-kjernen, etter at gratisstatusen til BitKeeper, det proprietære verktøyet de hadde brukt, ble trukket tilbake. Målene de satte seg, var fart, en enkel utforming, god støtte for tusenvis av parallelle grener, full distribusjon og evne til å håndtere et prosjekt på størrelse med kjernen.

Bruk begrepet i en beslutning

Bruk «Hva er git?» 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

Versjonskontroll svarer på spørsmålene som dukker opp så snart mer enn én person, eller én person over mer enn en uke, rører de samme filene: hva som ble endret, når, av hvem og hvorfor. Holdes svarene for hånd, lever de i hukommelsen, i filnavn som `final-v3` og i e-posttråder. Med git ligger de ved siden av koden, festet til hver endring, der hvem som helst kan søke i dem.

Den distribuerte utformingen endrer hvordan et team jobber fra dag til dag. Hver klone holder hele historikken, så ingen venter på en sentral server for å lese loggen eller committe, og når serveren er nede, kan alle jobbe videre. Grener er billige, så hver funksjon, retting eller eksperiment kan få sin egen, bli gjennomgått for seg og flettes inn når den er klar, mens hovedlinjen hele tiden holder mål for levering.

Den samme utformingen gjør git forsiktig med data. Nesten hver operasjon legger bare til i objektdatabasen, så arbeid som er committet, er vanskelig å miste: en gren som ble slettet ved en feil, en rebase som gikk skjevt, eller en reset til feil commit kan som regel gjøres om. Sikkerhetsnettet er refloggen, en lokal oversikt over hvor hver gren pekte og når. Som standard tas den vare på i 90 dager for commits som fortsatt kan nås, og i 30 dager for dem som ikke kan det.

Git er også felles grunn under de fleste andre verktøyene i en leveransekjede. Forger, verktøy for kodegjennomgang, byggepipelines, skript for produksjonssetting og statiske nettstedsgeneratorer leser alle fra den, og det gjør historikken til en kilde til fakta langt utover koden: dagen en side sist ble endret, commiten en utgivelse ble bygd fra, endringen en feil kom inn med.

Slik ser det ut i praksis

Selskapets egen kode viser tre sider av dette.

Kildekoden lå tidligere hos en tredjeparts vertstjeneste. Nå ligger den på en git-server selskapet drifter på sin egen maskinvare: én binærfil som styres av operativsystemet, valgt fordi den er SSH-først uten noe administrativt webgrensesnitt. Hvert deklarerte repo er merket som privat, og forgen avviser både anonyme klienter og klienter uten nøkkel. Git-kommandoene snakker med en SSH-lytter som bare er bundet til port 23231 på loopback-grensesnittet, og utenfra går veien via en hoppvert. Siden legger ikke skjul på prisen: å drifte sin egen forge betyr å eie autentisering, tilgangskontroll, lagring og sikkerhetskopier i tillegg.

Fullstendige kopier av historikken på flere maskiner er lette å forveksle med en sikkerhetskopi. Selskapets implementering av sikkerhetskopiering og gjenopprettingstester ble utformet for å utvide gjenopprettingsberedskapen utover leverandørens øyeblikksbilde. restic-skriptene og systemd-enhetene har støtte for en databasedump der dette er konfigurert, krypterte, dedupliserte øyeblikksbilder over SFTP til lagring hos en annen leverandør, opprydding etter oppbevaringsregelen og integritetssjekker. En separat månedlig arbeidsflyt gjenoppretter en kjent fil til et midlertidig arbeidsområde og sammenligner bytene. Denne testen må faktisk kjøres før den dokumenterer en gjenopprettingsberedskap: implementeringsdokumentasjonen datert 2026-10-27 sier at timerne var deaktivert uten et konfigurert sikkerhetskopirepo, og dokumenterer ingen fullført gjenoppretting i drift. Uansett hvor mange kloner som ligger på bærbare maskiner, gir de ikke den manglende dokumentasjonen.

Historikken mater også selve nettstedet. Nettstedets kolofon skriver at W3C Feed Validator advarer om at de fleste oppføringene i feedene har samme endringsdato, og at advarselen er riktig: datoene kommer fra git-historikken, og de fleste av disse sidene ble faktisk skrevet i en håndfull commits. Nettstedet viser heller advarselen enn å finne på en dato. Historikken forteller hva som skjedde, også at mye av det skjedde på én gang.

Én kommando leser et slikt faktum tilbake. Denne skriver ut den korte identifikatoren, commitdatoen i ISO 8601-form og emnelinjen til den siste commiten som rørte én bestemt fil:

```
git log -1 --format='%h %cI %s' -- README.md
```

Hvor det går galt

- **En hemmelighet som er committet én gang, blir liggende i historikken.** Sletter du filen i neste commit, forsvinner den fra det siste øyeblikksbildet, men den blir liggende i det forrige, som hver klone allerede har. Behandle legitimasjonen som lekket og bytt den ut. Å skrive om historikken etterpå begrenser spredningen, men kopier som ble laget i mellomtiden, beholder de gamle commitene.
- **En tvungen push overskriver en kollegas arbeid.** `git push --force` slår av kontrollen av at grenen i fjernrepoet er en forfar til det du pusher, så commits en kollega har pushet i mellomtiden, faller ut av grenen. `--force-with-lease` oppdaterer fjernrepoet bare så lenge det fortsatt peker dit du sist så det.
- **Klonene blir behandlet som sikkerhetskopien.** En kopi på en bærbar maskin henger etter, holder bare de grenene eieren tilfeldigvis har hentet, og ingen har noen gang gjenopprettet fra den med vilje. Saker, tilgangsregler og resten av forgens egne data ligger helt utenfor repoet.
- **En stor binærfil gjør hver klone tyngre for godt.** Siden git tar vare på hver versjon av hver fil, og hver klone laster ned alle sammen, følger en video som ble committet ved et uhell og slettet dagen etter, likevel med til hver maskin som kloner prosjektet senere.
- **Arbeid som bare ligger ett sted, forsvinner med det stedet.** Refloggen som redder en mislykket reset, ligger på én maskin og går ut på dato. Commits som aldri ble pushet, finnes ingen andre steder, og en gren som får ligge i fred i månedsvis, driver så langt unna at det blir et eget prosjekt å flette den inn igjen.

Les mer

- Git-dokumentasjonen — referansemanualen, boken og videoene, fra prosjektet selv.
- Pro Git — hele boken på nett, fra den første commiten til objektdatabasen under.
- What is Git? — øyeblikksbilder, lokale operasjoner, sjekksummer og de tre tilstandene i ett kapittel.
- A Short History of Git — hvorfor kjernemiljøet skrev sitt eget verktøy i 2005, og hva det skulle oppnå.
- git-reflog — hvordan du finner ut hvor en gren pekte før feilen, og hvor lenge den oversikten varer.
- git-push — valgene for tvungen push, og hva hvert av dem beskytter.

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 git?» 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/git/