Engineer

Säkerhetskopia

en kopia av data som skydd mot förlust, och att göra den

Vad det är

En säkerhetskopia är det som data återställs från när originalet raderas, skadas eller går förlorat tillsammans med maskinen där det lagrades, så den måste ligga någonstans dit de händelserna inte når. NIST:s ordlista definierar den som en kopia av filer och program som görs för att underlätta återställning. I vardagligt tal heter den ofta backup, och verbet är säkerhetskopiera: ett nattligt jobb säkerhetskopierar databasen och lämnar en säkerhetskopia efter sig.

Närliggande saker tas lätt för en säkerhetskopia. En replik tar emot varje ändring inom några sekunder, vilket skyddar mot en trasig disk och kopierar en felaktig `DELETE` lika troget. En ögonblicksbild (snapshot) fryser en volym i ett enda ögonblick och blir en säkerhetskopia först när den lagras där originalets fel inte kan följa efter. Det som definierar en säkerhetskopia är historik och åtskillnad: den ger tillbaka data så som de såg ut vid en tidigare tidpunkt, från en plats som överlevde det som tog originalet.

Databaser gör det här konkret. PostgreSQL:s manual beskriver tre i grunden olika angreppssätt. En SQL-dump, gjord med `pg_dump`, skriver ut de satser som bygger upp databasen igen, som en konsistent ögonblicksbild av läget när dumpen började. En säkerhetskopia på filsystemsnivå kopierar själva datakatalogen. Kontinuerlig arkivering kombinerar en sådan bassäkerhetskopia (base backup) med den write-ahead log (WAL) som servern skriver efteråt, och genom att spela upp loggen igen kan databasen återställas till valfritt ögonblick sedan bassäkerhetskopian togs. Den förmågan kallas point-in-time recovery, återställning till valfri tidpunkt.

Använd begreppet i ett beslut

Använd ”Vad är en säkerhetskopia?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Säkerhetskopior som testas i stället för att tas för givna, och en skriftlig plan för den morgon då en server inte startar igen.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Automatiserade 100 kritiska säkerhetskopior — 100 automatiserade kritiska säkerhetskopior av databaserna

Automatiserade 100 kritiska säkerhetskopior Hör av er →

Varför det spelar roll

Två siffror förvandlar en allmän önskan om att vara på den säkra sidan till en design. NIST definierar målet för återställningspunkt, RPO (recovery point objective), som den tidpunkt till vilken data måste återställas efter ett avbrott: i praktiken hur mycket av det senaste arbetet som verksamheten har råd att förlora. Målet för återställningstid, RTO (recovery time objective), är hur länge ett system kan vara under återställning innan avbrottet skadar det som organisationen finns till för.

De två drar i olika delar av designen. RPO bestäms av hur ofta en kopia tas: en nattlig dump kan förlora en dags skrivningar, medan WAL-arkivering krymper förlusten till de sista ögonblicken av logg som fortfarande är på väg till arkivet. RTO bestäms av hur lång tid en återställning tar. PostgreSQL:s manual påpekar att servern spelar upp varje WAL-segment sedan den senaste bassäkerhetskopian, vilket kan ta en stund när den kopian är gammal, så hur ofta bassäkerhetskopior tas är lika mycket ett beslut om RTO som om lagring.

Båda siffrorna är påståenden om en återställning, och därför bedöms en säkerhetskopia genom att den återställs. Ett jobb som avslutas med status noll visar att byte skrevs någonstans. Om de bygger upp ett fungerande system inom RTO besvaras genom att återställa enligt ett schema, någon annanstans än i produktion, och jämföra det som kommer tillbaka.

Åtskillnaden är den andra halvan. En kopia i samma konto, under samma inloggningsuppgifter, delar de fel som den ska överleva: ett stängt konto, en stulen nyckel, ett skript som körs mot fel mål. Den användbara frågan om varje kopia är vilken enskild händelse som skulle ta den med sig tillsammans med originalet.

Så ser det ut i praktiken

I den lilla änden utformades företagets implementering av säkerhetskopiering och återställningstest för att ge återställningen ett underlag utöver leverantörens ögonblicksbild. En Ansible-roll installerar restic-skript och systemd-enheter för databasdump där det är konfigurerat, krypterade och deduplicerade ögonblicksbilder, gallring enligt lagringsregler och integritetskontroller. Ett separat månatligt arbetsflöde återställer en känd fil till en tillfällig plats och jämför dess byte. Säkerhetskopiering och återställning har var sin framgångssignal, och tidsstyrningarna kräver ett konfigurerat lager. Lösenfrasen läses från en skyddad fil i stället för ett miljövärde som systemd tolkar. Detta är implementerade kontroller och förutsättningar för aktivering: implementationsunderlaget daterat 2026-10-27 anger att tidsstyrningarna var avaktiverade utan ett lager och redovisar ingen genomförd återställning under verkliga driftförhållanden. Det styrker automatiseringsarbetet, inte ett påstående om att återställning har lyckats eller att värden redan kan gå förlorad utan att äventyra data.

I större skala, i rollen som DevOps-ingenjör mellan 2022 och 2024, automatiserades hundra kritiska säkerhetskopior av data med Barman och Google Cloud för en produkt som lever på sina GIS-data. Barman skötte PostgreSQL-delen, Google Cloud höll kopiorna på annan plats, och Bash och Python orkestrerade och validerade körningarna. Integritetskontroller bekräftade att varje säkerhetskopia faktiskt var bra och inte bara fanns där.

I en tredje produkt bestod arbetet i att definiera återställningsförmågan i stället för att ta den för given. Lagringstiden för säkerhetskopiorna ligger i Bicep bredvid databasen, så återställning till valfri tidpunkt är en egenskap hos mallen snarare än en inställning som någon en gång klickade fram i en portal, och miljön runt den är också kod. Granskningen skrev ned luckorna: georedundans är avslagen, och målet för återställningstid är föreslaget snarare än uppmätt, med den övning som skulle mäta det beskriven och i väntan på att köras.

Var det går fel

Problem med säkerhetskopior brukar förbli dolda tills en återställning behövs, så vart och ett är värt att leta efter en lugn dag.

En filkopia av en databas som körs återställs som ett trasigt kluster. Att kopiera PostgreSQL:s datakatalog med `tar` medan servern körs samlar in filer från olika ögonblick, och manualen anger att servern måste stängas av för att en säkerhetskopia av det slaget ska gå att använda. En ögonblicksbild av filsystemet som tas i ett enda atomärt ögonblick fungerar om den innehåller WAL, som servern spelar upp vid start precis som efter en krasch; med data och WAL på olika filsystem kan en samtidig ögonblicksbild vara omöjlig. `pg_dump`, eller en bassäkerhetskopia för kontinuerlig arkivering, ger en konsistent kopia medan servern fortsätter att köras.

En lucka i WAL-arkivet sätter stopp för återställningen vid luckan. Återställning till en given tidpunkt kräver en obruten följd av arkiverade WAL-filer som når tillbaka till starten av bassäkerhetskopian. När arkiveringskommandot fortsätter att misslyckas fortsätter PostgreSQL att försöka igen, och WAL hopar sig i `pg_wal/`, så det första symptomet är ofta en disk som fylls. Manualens råd är att sätta upp och testa arkiveringen före den första bassäkerhetskopian, och att övervaka den för fördröjningar.

Jobbet stannar, och tystnaden ser ut som framgång. Ett timerjobb som går sönder skriver sitt fel till en logg som ingen läser, och varje tyst dag passerar som en bra dag. Övervakningen måste behandla tystnad som larmet: en incheckning som skickas bara efter en lyckad körning, till en tjänst som larmar när den uteblir, fångar ett jobb som har stannat lika väl som ett som misslyckas högljutt.

Nyckeln försvinner, eller kommer fram förändrad. Kryptering gör åtkomsten beroende av nycklarna. restic stöder flera lösenord för samma repo; varningen om oåterkallelig dataförlust gäller när ingen användbar nyckel till repot finns kvar. Nyckeln måste finnas någonstans som överlever maskinen. I uppsättningen med restic ovan kunde förändringen ha kommit från tjänstehanteraren själv: den tolkar escape-tecken i miljövärden, så en lösenfras med ett omvänt snedstreck, överlämnad på det sättet, hade nått restic som andra byte än dem som repot skapades med, och utan ett ord till varning. Att läsa den från en fil stänger den vägen.

Gallringen tar bort den kopia som behövdes. Korruption märks ofta veckor efter att den uppstod, och en policy som behåller sju dagliga kopior har då ersatt varje ren kopia med en skadad. En policy i flera nivåer behåller en lång svans till låg kostnad, och restics `--dry-run` visar vad en policy skulle ta bort innan något försvinner:

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

`forget` tar bort ögonblicksbilder och `prune` tar bort de data som bara de refererade till, så efter en gallring med `prune` är de gamla kopiorna borta för gott.

Integritetskontrollen går igenom och återställningen misslyckas ändå. Som standard verifierar `restic check` repots struktur utan att läsa varje packfil, eftersom det skulle innebära att alla laddas ned; `--read-data` läser allt, och `--read-data-subset=1/5` läser en femtedel, så fem körningar från `1/5` till `5/5` täcker alltihop. En fullständig läsning bevisar bara att de lagrade byten är intakta. Om de bygger upp en fungerande databas, med rätt roller och inställningar, är något som bara en återställning visar.

Läs mer

- NIST glossary: backup — definitionen som används här och publikationerna bakom den.
- NIST SP 800-34 Rev. 1 — vägledningen för kontinuitetsplanering som definierar RPO och RTO.
- PostgreSQL: Backup and Restore — de tre angreppssätten och vad vart och ett kostar.
- PostgreSQL: Continuous Archiving and Point-in-Time Recovery — bassäkerhetskopior, WAL-arkivering och återställning till ett valt ögonblick.
- restic: Preparing a new repository — lösenord, lösenordsfiler och varningen om vad som händer när de går förlorade.
- restic: Removing backup snapshots — bevarandepolicyer, `forget`, `prune` och `--dry-run`.

Ordlista

Välj utifrån era faktiska förutsättningar

Ta med driftsättningsvägen, driftsriskerna och dem som har jour. Vi kan avgränsa en förändring med kontroller, en väg tillbaka och tydligt ansvar.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är en säkerhetskopia?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

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

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/backup/