Engineer

Nedetid

⛒ den periode hvor en tjeneste ikke svarer nogen

Hvad det er

Nedetid er den tid, hvor en tjeneste er ude af drift for dem, der er afhængige af den. Man taler også om udfald eller nedbrud; det modsatte er oppetid, og den andel af tiden, en tjeneste er oppe, er dens tilgængelighed, som regel opgivet i procent over en måned eller et år.

Det rene tilfælde er totalt: siden loader slet ikke, API'et afviser forbindelser, databasen er nede. Det almindelige tilfælde er mere rodet. En betaling, der løber ud i timeout, mens kataloget virker, en login-side, der loader, men afviser hver eneste adgangskode, et API, der svarer på én ud af tre forespørgsler: hver af dem er nedetid for den, der skal bruge netop den vej, selv om et health check rettet mod forsiden melder grønt. En brugbar definition tæller, hvad brugerne kan gøre, og ikke om en proces kører.

Der findes to slags nedetid. Uplanlagt nedetid er det udfald, ingen valgte: en disk, et dårligt deploy, et udløbet certifikat, en afhængighed, der gik ned først. Planlagt nedetid vælges på forhånd, som regel i et servicevindue, for at opgradere, migrere eller flytte noget, der skal stå stille, mens det ændres. Planlagt nedetid er lettere at leve med, fordi brugerne kan varsles, men den tæller stadig imod tilgængeligheden, medmindre aftalen udtrykkeligt undtager den.

HTTP har et ordforråd til selve øjeblikket. RFC 9110 definerer 5xx-klassen af statuskoder til en server, der ved, at den har fejlet, eller som ikke kan udføre det, der blev bedt om. Tre af dem betyder hver sit under et udfald. 502 Bad Gateway siger, at en proxy fik et ugyldigt svar fra serveren bag den. 504 Gateway Timeout siger, at proxyen ventede for længe på et. Og 503 Service Unavailable er den,

man sender med vilje: specifikationen knytter den til midlertidig overbelastning eller planlagt vedligeholdelse og lader serveren sige, hvornår man skal komme igen:

```
HTTP/1.1 503 Service Unavailable
Retry-After: 120
```

Det fortæller browsere, API-klienter og søgemaskinernes crawlere, at tilstanden er midlertidig, og foreslår at vente to minutter. Det er et helt andet budskab end en fejlside serveret med en succeskode.

Brug begrebet i en beslutning

Brug »Hvad er nedetid?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Overvågning og alarmer, der siger til, mens fejlen stadig er lille, og tallene, der viser, hvor tit den bliver ved med at være det.

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 bygget det første observerbarhedsdashboard — virksomhedens første overblik over sine egne systemer

Har bygget det første observerbarhedsdashboard Kontakt os →

Hvorfor det betyder noget

Mål for tilgængelighed skrives i nitaller, og nitallerne kan regnes om til tid. Over et år på 365 dage, altså 8.760 timer, giver hvert mål plads til så meget nedetid:

- 99 %: 87,6 timer om året, cirka tre et halvt døgn;
- 99,9 %: 8,76 timer om året eller 43,2 minutter på en måned med 30 dage;
- 99,95 %: 4,38 timer om året eller 21,6 minutter på en måned med 30 dage;
- 99,99 %: 52,56 minutter om året;
- 99,999 %: omkring 5,26 minutter om året.

Hvert ekstra nital deler rammen med ti, og prisen for at nå det stiger brat. Ved 99,9 % kan en person nå at blive vækket, logge ind, finde fejlen og rette den inden for månedens budget. Ved 99,99 % giver hele året under en time, hvilket måske rækker til én hændelse, så genopretningen må ske automatisk. Ved 99,999 % skal selv en rutinemæssig genstart ske, uden at brugerne mærker det.

Afhængigheder ganges med hinanden. En forespørgsel, der kræver en load balancer, en applikation og en database, der hver er oppe 99,9 % af tiden og fejler uafhængigt af hinanden, bliver kun besvaret, når alle tre er oppe: $0{,}999^3$, eller omkring 99,7 %, cirka 26 timers nedetid om året i stedet for 8,76. En tjeneste kan ikke være mere tilgængelig end kæden af ting, den har brug for, medmindre den fjerner led fra kæden eller dublerer dem.

Planlægningen begynder i forretningen. NIST's vejledning i beredskabsplanlægning for amerikanske føderale systemer, SP 800-34, navngiver tre tal, der bør være på plads, før noget går i stykker. Maximum tolerable downtime (MTD) er det samlede udfald, systemets ejer vil acceptere. Recovery time objective (RTO) er, hvor længe et system kan være nede, før konsekvenserne bliver uacceptable, og den skal være kortere end MTD. Recovery point objective (RPO) er, hvor langt tilbage i tiden data må gendannes fra den seneste backup, så den måler, hvor meget datatab man kan tåle, og ikke hvor længe man er offline. Når de tre tal er skrevet ned, bliver en backupplan og et failoverdesign svar på et spørgsmål i stedet for gæt.

Sådan ser det ud i praksis

På en IPTV- og OTT-streamingplatform med kunder i Kina, USA og Tyskland var hver time nogens bedste sendetid, så en stille time til vedligeholdelse fandtes simpelthen ikke. Supporten kørte døgnet rundt for omkring tusind servere plus en god portion systemer, kunderne selv ejede. En stor del af arbejdet var at opdage noget, der begyndte at skride, før det blev til et udfald, for på en direkte stream er nedetid nogens fjernsyn, der bare stopper. Platformen forblev kontinuerligt tilgængelig for sit globale publikum.

I en mellemstor virksomhed kørte ti PostgreSQL- og SQL Server-databaseservere med 99,9 % oppetid, hvilket efter regnestykket ovenfor er en ramme på 8,76 timer om året for hver. Bag tallet lå automatiserede backups med `pg_dump` og SQL Server Agent-jobs med opbevaringsregler og offsite-lagring, overvågning i Prometheus og Grafana og dokumenterede procedurer for disaster recovery, herunder point-in-time-gendannelse og failover.

Og på en flåde af tredive Ubuntu-servere, hvis opsætning var vokset mere ved tilfældighed end plan, blev disaster recovery-delen bygget om med backups og gendannelsesprocedurer, der faktisk blev testet. Netværkskonfigurationen blev én hærdet baseline på tværs af alle instanser. Nedetidsrisikoen faldt, med genopretningsveje, der var afprøvet, før nogen fik brug for dem.

Hvor det går galt

Overvågningen sidder inde i udfaldet. Et health check, der kører på samme server eller i samme datacenter, bliver ved med at melde grønt, mens brugerne møder et udløbet TLS-certifikat, en forkert DNS-post eller en load balancer, der peger ud i luften. Mål udefra, fra mere end ét sted, ad den vej en rigtig bruger tager, og alarmér på det, den måling ser.

Fejlsiden siger 200. Et framework, der viser en venlig fejlside med en succeskode, skjuler udfaldet for overvågningen, for klienter, der ellers ville prøve igen, og for crawlere, som kan indeksere fejlsiden i stedet for den rigtige. Fejl skal have en fejlstatus, og planlagt vedligeholdelse skal svare 503 med en Retry-After-header.

Backuppen blev aldrig gendannet. En backup beviser kun, at bytes blev kopieret. Indtil en gendannelse er kørt, timet og tjekket mod data, er genopretningstiden et gæt, og den er ofte langt længere end RTO'en på papiret. Gendan efter en fast plan, i et separat miljø, og skriv ned, hvor lang tid det tog. Det samme gælder failover: en standby, der aldrig har overtaget, er et håb, og øvelser i failover under rolige forhold hjælper med at afsløre problemer før en hændelse.

Udrulninger skaber de udfald, de skulle forhindre. Et deploy, der stopper den gamle version, før den nye er klar, efterlader et hul ved hver udgivelse. En skemaændring, der omskriver en stor tabel under en eksklusiv lås, blokerer alle forespørgsler mod den, indtil den er færdig. Rul nye instanser ud bag et readiness-tjek, før de gamle stopper, del skemaændringer op i et expand-trin og et senere contract-trin, så gammel og ny kode begge virker imens, og sæt på Kubernetes et PodDisruptionBudget, så en node-drain ikke kan tage alle replikaer ned på én gang.

Læs mere

- RFC 9110: HTTP Semantics, section 15.6 — 5xx-statuskoderne, herunder 503 og dens Retry-After-hint.
- NIST SP 800-34 Rev. 1: Contingency Planning Guide for Federal Information Systems — hvor maximum tolerable downtime, RTO og RPO er defineret.
- PostgreSQL: High Availability, Load Balancing, and Replication — standbyservere og failover, sådan som databasens egen manual beskriver dem.

- PostgreSQL: Continuous Archiving and Point-in-Time Recovery — den basebackup og det WAL-arkiv, en point-in-time-gendannelse bygger på.
- Kubernetes: Disruptions — frivillige afbrydelser og de PodDisruptionBudgets, der begrænser dem.

Ordliste

Vælg ud fra jeres faktiske vilkår

Medbring udrulningsforløbet, driftsrisiciene og dem, der har vagten. Vi kan afgrænse en ændring med kontroller, en vej tilbage og tydeligt ansvar.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er nedetid?« 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/downtime/