Engineer

DNS

katalogen som gjør et navn om til en adresse

Hva det er

Nesten hver tilkobling på internett begynner med et spørsmål til DNS, domenenavnsystemet (Domain Name System), det distribuerte oppslagsverket som maskiner slår opp i før noe annet kan skje. Et program som har et navn som `example.com`, spør etter adressen det skal koble seg til, og det samme systemet svarer på en håndfull andre spørsmål om navnet: hvilke servere som tar imot e-post for det, hvilken tekst eieren har publisert, og hvilke servere som har ansvaret for det.

Oppslagsverket er et tre som leses fra høyre mot venstre. Øverst står roten, under den toppdomener som `.com`, `.dk` eller `.no`, under dem domenene folk registrerer, og under dem igjen de underdomenene eierne lager. Hele treet ligger ikke på én server. Hvert nivå overlater ansvaret for nivået under til navngitte servere, og den delen av treet som ett sett med servere svarer for, kalles en sone. Den som styrer en sone, bestemmer hva navnene i den peker på.

To slags servere gjør jobben. Autoritative navnetjenere holder en sone og svarer for den med siste ord. Rekursive resolvere, drevet av internettleverandører, bedrifter eller offentlige tjenester, går veien på vegne av en klient: de spør en rotserver hvor `.com` holder til, spør `.com` hvor `example.com` holder til, spør domenets egne servere etter posten og gir svaret tilbake. Operativsystemet ditt kjører bare en liten stub-resolver som sender spørsmålet videre til en rekursiv resolver og venter.

Svarene er typede poster, og en sonefil er den klassiske måten å skrive dem ned på:

```
$ORIGIN example.com.
$TTL 3600
@     IN  SOA    ns1.example.net. hostmaster.example.com. (
                 2026101701 7200 900 1209600 300 )
@     IN  NS     ns1.example.net.
@     IN  A      192.0.2.10
@     IN  AAAA   2001:db8::10
www   IN  CNAME  example.com.
@     IN  MX     10 mail.example.net.
@     IN  TXT    "v=spf1 include:_spf.example.net -all"
```

`A` og `AAAA` gir IPv4- og IPv6-adresser. `CNAME` gjør ett navn til et alias for et annet. `MX` navngir serveren som tar imot e-post, og `TXT` bærer fri tekst, som i praksis betyr policyer for autentisering av e-post og bevis på at noen styrer domenet. `NS` og `SOA` beskriver selve sonen.

Hver post har en levetid, time to live eller TTL, her 3 600 sekunder, og det tallet er nøkkelen til hvordan DNS oppfører seg. En resolver som har hentet et svar, kan bruke det om igjen til TTL-en har løpt ut, uten å spørre på nytt. Med slik caching blir de fleste oppslag besvart av en kopi nær klienten, og systemet fortsetter å svare når en autoritativ server en kort stund er utilgjengelig. Prisen er at ingen kan tvinge verden til å glemme et gammelt svar før tiden.

Bruk begrepet i en beslutning

Bruk «Hva er DNS?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Wi-Fi, brannmurer og fjerntilgang satt opp slik at de ansatte kan jobbe hvor som helst, og bare de rette personene slipper inn.

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. Administrerte 30 Ubuntu Linux-VPS-instanser — 30 Linux-servere administrert, med gjenoppretting planlagt på forhånd

Administrerte 30 Ubuntu Linux-VPS-instanser Ta kontakt →

Hvorfor det er viktig

Hver tilkobling til en tjeneste med navn begynner med et oppslag. Når DNS er feil, kan serverne være i fin form og nettstedet, API-et og e-posten likevel se nede ut, fordi ingenting når fram til dem. Når DNS er riktig, legger ingen merke til det, og derfor er det gjerne den minst dokumenterte delen av et system.

DNS er også stedet der kontrollen over et domene utøves og bevises. E-postleverandører leser policyene for SPF, DKIM og DMARC fra `TXT` -poster for å avgjøre om en melding som sier den kommer fra domenet ditt, er ekte, så en gal eller manglende post sender ekte e-post rett i søppelposten. Sertifikatutstedere og hostede tjenester ber om en post med en bestemt verdi som bevis på at du eier navnet. Den som kan redigere sonen, kan omdirigere nettstedet, fange opp e-posten og skaffe seg et sertifikat for domenet. Det gjør tilgang til DNS-kontoen like følsom som tilgang til serverne.

Caching former hver endring. Det folk kaller propagering, er egentlig tusenvis av resolvere som hver venter på at sin egen kopi skal gå ut på tid. Vil du flytte et nettsted med lite forstyrrelse, senker du TTL-en et døgn eller så i forveien, venter til den gamle TTL-en har gått, gjør endringen og hever TTL-en igjen når alt har satt seg.

Klassisk DNS går i klartekst, og verken svarene eller spørsmålene er private. DNSSEC lar en resolver sjekke at et svar virkelig kom fra eieren av sonen og ikke er endret underveis, og etter sin egen spesifikasjon gir det ingen konfidensialitet. Krypterte transporter som DNS over HTTPS skjuler spørsmålene for nettverket på veien, mens resolveren i den andre enden fortsatt ser hvert navn som blir slått opp. Å velge resolver er å velge hvem som får den listen.

Slik ser det ut i praksis

Selskapets egen sone viser det vanligste utgangspunktet og en vei ut av det. Den hadde omtrent tjue poster, holdt i en leverandørs nettkonsoll, fordelt på en nettilstedeværelse, et git-underdomene, to personlige omdirigeringer, e-postruting gjennom en leverandør av hostede postbokser, et domene for transaksjonell utsending og autentiseringspostene for begge. I en konsoll er den gjeldende tilstanden det den siste som klikket, har etterlatt seg. Nå er sonen data i et repo: aktive poster, applikasjonsposter, poster for transportpolicy og en egen liste over poster som venter på å bli fjernet. Ett play avstemmer den deklarasjonen mot leverandørens API. To inngangspunkter til ligger bak tagger som må slås på eksplisitt: en revisjon som rapporterer forskjellen uten å endre noe, og en eksport som skriver ut sonen i standard sonefilformat, slik at noe annet enn repoet kan lese den. Revisjonen er den delen som brukes mest, fordi det oftere er forskjellen man vil vite, enn at den skal lukkes.

Den samme leverandøren har en bevisst smal jobb. Kolofonen forklarer at Cloudflare svarer på DNS for nettstedet, og bare DNS, uten noen proxy foran, og den skriver ut serverens adresse rett ut, fordi DNS allerede gir den adressen til alle som spør.

Å lese DNS hører også med til den faste beskrivelsen av driftsmiljøet. Blant sju rapporteringsroller som bare leser og gjengir verten i drift som Markdown, er en leverandøroversikt på tvers av fire leverandørflater: DNS, to sky-API-er og et API for dedikerte servere. Regelen for alle sju er at det ikke skrives ut noe tall som kjøringen ikke har målt.

Hvor det går galt

Feil i DNS er ofte forsinket, mellomlagret eller delt, så det samme navnet virker for én person og feiler for en annen. Her er de som går igjen.

En endring vil ikke slå gjennom. Noen endrer en post med en TTL på et døgn og venter at den nye adressen skal gjelde med en gang, men resolvere som har lagret det gamle svaret, fortsetter å dele det ut til kopien deres går ut på tid. En stillere variant rammer nye navn: ble et navn slått opp før posten fantes, lagres også det negative svaret, så lenge sonens `SOA` tillater (RFC 2308). Senk TTL-en før planlagte endringer, og opprett poster før noe spør etter dem.

Et alias står der det ikke kan stå. RFC 1034 sier at et navn som har en `CNAME`, ikke kan ha andre data. Selve domenet, `example.com`, har alltid `SOA`- og `NS`-poster, så en sone som følger standarden, kan ikke gjøre toppen av sonen til et alias, og en `CNAME` på et underdomene som også trenger `MX`- eller `TXT`-poster, bryter den samme regelen. Leverandører tilbyr utflatede poster eller `ALIAS`-poster som sin egen omvei. Hver av dem oppfører seg litt forskjellig, så les hva din leverandør faktisk serverer.

Autentiseringen av e-post velter. Hver `include` i en SPF-post fører til DNS-oppslag, og RFC 7208 setter et tak på ti til sammen. Én tjeneste til for nyhetsbrev eller saker kan presse posten over grensen, og mottakerne behandler da policyen som en permanent feil, så e-posten den skulle autentisere, blir vurdert uten den. Tell oppslagene hver gang en ny avsender legges til.

En post lever lenger enn målet sitt. Et underdomene peker med `CNAME` på en bucket i skylagring eller en hostet app som siden er slettet. Lar leverandøren hvem som helst ta det ressursnavnet, kan en fremmed gjøre krav på det og servere eget innhold under underdomenet ditt, med domenets omdømme på kjøpet. Fjern posten i den samme endringen som fjerner det den peker på.

Sonen finnes bare i en konsoll. Der er det verken historikk, gjennomgang eller diff, så en `MX` -post som slettes ved et uhell, blir oppdaget når e-posten slutter å komme, og den forrige verdien må gjenskapes etter hukommelsen. Å eksportere sonen jevnlig, eller enda bedre å deklarere den som kode, gir hver endring et spor og en vei tilbake.

Les mer

- RFC 1035: Domain Names, Implementation and Specification — postformatene, syntaksen for sonefiler og protokollen på linjen.
- RFC 1034: Domain Names, Concepts and Facilities — treet, sonene, delegeringen og hvordan resolvere går gjennom dem.
- RFC 9499: DNS Terminology — nøyaktige definisjoner av ordene denne siden bruker.
- RFC 2308: Negative Caching of DNS Queries — hvor lenge en resolver kan huske at et navn ikke finnes.
- RFC 4033: DNS Security Introduction and Requirements — hva DNSSEC beskytter, og hva det lar være.
- RFC 8484: DNS Queries over HTTPS — oppslag fraktet inne i vanlig kryptert webtrafikk.

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 DNS?» 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/dns/