Engineer

AWS

Amazons skyplattform

Hva det er

AWS, kort for Amazon Web Services, er Amazons virksomhet med å leie ut datakraft per time og per gigabyte: virtuelle servere, fillagring, administrerte databaser, private nettverk, køer og en lang katalog med andre tjenester, alt slått på gjennom ett API og fakturert til én konto. Amazon har solgt det siden 2006. To av de best kjente tjenestene forklarer mye av resten. Amazon EC2 leier ut virtuelle maskiner, som AWS kaller instanser, og Amazon S3 lagrer filer som objekter i beholdere som kalles bøtter (buckets).

Med ordene i NISTs definisjon er AWS en offentlig sky: alle med et betalingskort kan åpne en konto, ressursene er på plass i løpet av minutter, og hvert sekund eller hver gigabyte som brukes, blir målt. Den veies som regel opp mot Microsofts Azure og Google Cloud, som selger omtrent de samme byggeklossene under andre navn. Det AWS kaller en EC2-instans, er en virtuell maskin i Azure eller en Compute Engine-instans i Google Cloud.

Hver ressurs i AWS befinner seg et bestemt sted. Tjenesten er delt inn i regioner, som hver er laget for å være isolert fra de andre, og hver region har flere tilgjengelighetssoner (Availability Zones). En sone er ett eller flere separate datasentre med redundant strøm, nettverk og tilkobling, fysisk atskilt fra de andre sonene i regionen. En ressurs opprettes i én region, ofte i én enkelt sone, og et team velger stedet ut fra latens, lovverk og pris.

All tilgang går gjennom AWS Identity and Access Management, eller IAM. En konto starter med en rotbruker som kan gjøre alt, og IAM legger til brukere, grupper, roller og policyer som sier hvilket API-kall som er tillatt på hvilken ressurs og under hvilke betingelser. De fleste spørsmål om sikkerhet i AWS viser seg å være spørsmål om de policyene.

Bruk begrepet i en beslutning

Bruk «Hva er AWS?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Servere dimensjonert etter det du faktisk bruker, flyttet inn i skyen eller ut av den igjen, med månedsregningen regnet ut på forhånd.

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. Driftet 20 Docker-apper på Google Cloud og AWS — 20 containeriserte applikasjoner produksjonssatt og overvåket

Driftet 20 Docker-apper på Google Cloud og AWS Ta kontakt →

Hvorfor det er viktig

Team velger AWS for bredden. Et produkt som trenger en administrert PostgreSQL, en kø, et CDN, objektlagring og et sted å kjøre containere, kan få alt dette på én regning og i ett identitetssystem, dokumentert etter et felles mønster. Bredden er også prisen. Hver tjeneste har sine egne prisdimensjoner, sine egne grenser og sitt eget hjørne av policyspråket i IAM, og et team som tar i bruk ti tjenester, vedlikeholder nå ti små integrasjoner mot Amazons grensesnitt. Å gå ut senere betyr å erstatte hver av dem.

Den andre beslutningen er hva som blir liggende hos kunden. AWS publiserer en modell for delt ansvar og trekker grensen med to uttrykk: AWS har ansvaret for sikkerheten i selve skyplattformen («of the cloud»), altså maskinvaren, programvaren, nettverket og anleggene som kjører tjenestene, og kunden for sikkerheten «i» skyen. Hvor grensen går, avhenger av tjenesten. På en EC2-instans oppdaterer kunden gjesteoperativsystemet, tar seg av hvert program som er installert på det, og konfigurerer sikkerhetsgruppen, brannmuren foran instansen. På S3 eller DynamoDB driver AWS operativsystemet og plattformen, og kunden sitter igjen med dataene, valgene for kryptering og tillatelsene som avgjør hvem som kan lese dem.

Den tredje er kostnad, som i AWS er en egenskap ved designet mer enn en faktura som kommer senere. Amazons Well-Architected Framework nevner seks søyler en arbeidslast skal vurderes etter: operasjonell fortreffelighet, sikkerhet, pålitelighet, ytelseseffektivitet, kostnadsoptimalisering og bærekraft. Kostnad fortjener plassen på den listen fordi det samme systemet kan bygges billig eller dyrt gjennom valg som tas den første uka: størrelsen på instansene, hvor mange soner, hvor mye data som forlater regionen, og hvilke administrerte tjenester som tar over for en prosess på en vanlig server.

Slik ser det ut i praksis

På siden om teknologiene står AWS ved siden av Google Cloud som én av de to skyene selskapet jobber i som standard, med Azure lagt til der en kunde allerede holder til.

Ett oppdrag brukte begge de skyene samtidig: tjue Docker-applikasjoner kjørte på tvers av Google Cloud og AWS, noen av dem R-baserte analyseapper. De ble satt i produksjon og vedlikeholdt med den samme konfigurasjonen og overvåkingen i hver sky, slik at ingen app ble et eget spesialtilfelle. Når noe gikk galt, gikk feilsøkingen gjennom Podman og feilsøkingen av orkestreringen gjennom Kubernetes, og analyseappene ble holdt stabile og reproduserbare i produksjon på begge sider.

Et annet oppdrag gikk den andre veien. For en internasjonal kunde var et AWS-oppsett blitt overdimensjonert og hadde samlet opp ubrukte tjenester, og kostnaden hadde vokst ut av proporsjon med virksomhetens behov. Arbeidet kartla hvilke tjenester som ble brukt, fjernet unødvendige tjenester, justerte den gjenværende kapasiteten og flyttet arbeidslaster bort fra AWS. Dette er de leverte infrastrukturendringene som registreringen beskriver. Den dokumenterer ikke et kostnadsforhold på sammenlignbart grunnlag, en produktivitetsmåling eller en migrering uten driftsforstyrrelser. Eksemplet er nyttig for metoden for å undersøke det eksisterende oppsettet og endre det virksomheten trenger å drifte, ikke som et løfte om en bestemt besparelse.

Begge tilfellene koker ned til den samme vanen. En tjeneste med betaling etter bruk fakturerer for alt som er slått på, så et AWS-miljø holder seg sunt når noen fortsetter å sammenligne det som kjører, med det virksomheten faktisk bruker.

Hvor det går galt

AWS mister sjelden data eller blir liggende nede lenge. De dyre hendelsene starter som regel inne i kundens egen konto.

Rotbrukeren gjør dagens arbeid. Hver konto begynner med en rotbruker, e-postadressen og passordet som åpnet den, med ubegrenset tilgang til hver ressurs og innstilling, også til å stenge kontoen. Å logge inn som rot for rutineoppgaver betyr at ett lekket passord når alt. AWS anbefaler å sikre

rotbrukerens påloggingsinformasjon, legge flerfaktorautentisering på den og spare den til den lille håndfullen oppgaver som krever den, mens det daglige arbeidet skjer gjennom føderert pålogging og roller.

Langvarige tilgangsnøkler lekker. Et nøkkelpar for tilgang virker til noen sletter det, og det reiser godt: inn i dotfilene på en bærbar maskin, en CI-variabel, et containerbilde, en commit. Listen over beste praksis for IAM starter fra den andre enden. Folk logger inn gjennom en identitetsleverandør og får midlertidig påloggingsinformasjon, og arbeidslaster henter midlertidig påloggingsinformasjon fra en IAM-rolle knyttet til maskinene de kjører på. Der en langvarig nøkkel virkelig trengs, bør den roteres, og nøkler, brukere og roller som ikke har vært i bruk, bør fjernes.

Et budsjettvarsel blir tatt for en brems. AWS Budgets sender et varsel når faktisk eller anslått forbruk krysser en terskel, og tallene oppdateres opptil tre ganger i døgnet, så et varsel melder om penger som allerede er brukt. Ressursene fortsetter å kjøre og fortsetter å koste. Budgets kan også utføre en handling, f.eks. å legge på en IAM-policy eller en service control policy som hindrer at nye ressurser opprettes, eller å gjøre noe med bestemte EC2- eller RDS-instanser. Det skjer bare når handlingen er satt opp på forhånd, med en rolle å handle gjennom.

Alt ligger i én tilgjengelighetssone. Én instans med én disk i én sone er det billigste oppsettet, og det deler skjebne med det ene stedet. Soner finnes for at en feil i strøm eller nettverk i én skal la de andre kjøre videre, og dokumentasjonen fra AWS anbefaler å spre en applikasjon over flere. En database som betyr noe, får en reserve i en annen sone, og et webnivå får instanser i minst to, bak en lastbalanserer som spenner over begge.

En bøtte som ble åpnet for en demo, blir stående åpen. Nye S3-bøtter blokkerer offentlig tilgang som standard, men en bøttepolicy eller en tillatelse på et objekt kan endre det på et minutt, og en lenke som ble delt med en kunde for én ettermiddag, er fortsatt offentlig et år senere. S3 Block Public Access, satt for hele kontoen eller organisasjonen, overstyrer de policyene. Slå det på overalt der en konto ikke har noen grunn til å publisere filer, og server offentlige filer fra en bøtte som er laget for akkurat det.

Les mer

- AWS Documentation — oversikten over brukerveiledninger, API-referanser og opplæring for hver tjeneste i AWS.
- Shared responsibility — sikkerhetssøylen i Well-Architected om hvilke lag AWS sikrer, og hvilke kunden sikrer.
- The pillars of the framework — de seks områdene AWS vil at en arbeidslast skal vurderes etter, med kostnad og pålitelighet blant dem.
- AWS Regions and Availability Zones — hvordan regioner og soner er isolert fra hverandre, og hvorfor en applikasjon bør spenne over flere soner.
- Security best practices in IAM — føderering, roller, flerfaktorautentisering, rotbrukeren og minste privilegium, i én liste.
- Managing your costs with AWS Budgets — budsjetter for faktisk og anslått forbruk, varslene de sender, og handlingene de kan utføre.

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 AWS?» 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/aws/