Engineer

AWS

molnplattformen från Amazon

Vad det är

AWS, förkortning för Amazon Web Services, är Amazons verksamhet med att hyra ut datorkraft per timme och per gigabyte: virtuella servrar, lagring av filer, hanterade databaser, privata nätverk, köer och en lång katalog med ytterligare tjänster, som alla slås på via ett och samma API och faktureras till ett och samma konto. Amazon har sålt det sedan 2006. Två av de mest kända tjänsterna förklarar mycket av resten. Amazon EC2 hyr ut virtuella maskiner, som AWS kallar instanser, och Amazon S3 lagrar filer som objekt i behållare som kallas buckets.

Med ordförrådet i NIST:s definition är AWS ett publikt moln: vem som helst med ett betalkort kan öppna ett konto, resurser dyker upp inom några minuter, och varje sekund eller gigabyte som används mäts. Det vägs oftast mot Microsofts Azure och Google Cloud, som säljer ungefär samma byggstenar under andra namn. Det som AWS kallar en EC2-instans är en virtuell maskin i Azure eller en Compute Engine-instans i Google Cloud.

Varje resurs i AWS finns på en bestämd plats. Tjänsten är indelad i regioner (Regions), var och en utformad för att vara isolerad från de andra, och varje region rymmer flera tillgänglighetszoner (Availability Zones). En zon är ett eller flera separata datacenter med redundant strömförsörjning, nätverk och anslutningar, fysiskt åtskilda från de andra zonerna i samma region. En resurs skapas i en region, ofta i en enda zon, och ett team väljer plats med latens, lagstiftning och pris i åtanke.

Åtkomsten till allt detta går genom AWS Identity and Access Management, eller IAM. Ett konto börjar med en rotanvändare som kan göra vad som helst, och IAM lägger till användare, grupper, roller och policyer som anger vilket API-anrop som är tillåtet på vilken resurs och under vilka villkor. De flesta frågor om säkerheten i AWS visar sig vara frågor om de policyerna.

Använd begreppet i ett beslut

Använd ”Vad är AWS?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Servrar dimensionerade efter vad du faktiskt använder, flyttade till molnet eller tillbaka ut ur det, med månadskostnaden uträknad först.

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. Drev 20 Docker-appar på Google Cloud och AWS — 20 containerbaserade applikationer, driftsatta och övervakade

Drev 20 Docker-appar på Google Cloud och AWS Hör av er →

Varför det spelar roll

Team väljer AWS för bredden. En produkt som behöver en hanterad PostgreSQL, en kö, ett CDN, objektlagring och någonstans att köra containrar kan få allt detta på en faktura och i ett identitetssystem, dokumenterat efter ett gemensamt mönster. Bredden är också priset. Varje tjänst har sina egna prisdimensioner, sina egna gränser och sitt eget hörn av IAM:s policyspråk, och ett team som tar tio tjänster i bruk underhåller därmed tio små integrationer med Amazons gränssnitt. Att lämna senare innebär att ersätta var och en av dem.

Det andra beslutet gäller vad som stannar hos kunden. AWS publicerar en modell för delat ansvar och drar gränsen med två uttryck: AWS ansvarar för säkerheten ”för” molnet, alltså den hårdvara, mjukvara, de nätverk och de anläggningar som kör tjänsterna, och kunden för säkerheten ”i” molnet. Var gränsen hamnar beror på tjänsten. På en EC2-instans patchar kunden gästens operativsystem, sköter varje program som är installerat på den och konfigurerar säkerhetsgruppen, den brandvägg som står framför instansen. På S3 eller DynamoDB kör AWS operativsystemet och plattformen, och kunden blir kvar med datan, dess alternativ för kryptering och de behörigheter som avgör vem som får läsa den.

Det tredje är kostnaden, som i AWS är en egenskap hos designen mer än en faktura som kommer senare. Amazons Well-Architected Framework namnger sex pelare som en arbetslast granskas mot: operativ förträfflighet, säkerhet, tillförlitlighet, prestandaeffektivitet, kostnadsoptimering och hållbarhet. Kostnaden förtjänar sin plats på listan eftersom samma system kan byggas billigt eller dyrt genom val som görs den första veckan: instansstorlekar, hur många zoner, hur mycket data som lämnar regionen och vilka hanterade tjänster som ersätter en process på en vanlig server.

Så ser det ut i praktiken

På sidan om teknikstacken står AWS bredvid Google Cloud som ett av de två moln som företaget arbetar i som standard, med Azure där kunden redan finns.

Ett uppdrag använde båda de molnen samtidigt: tjugo Docker-applikationer kördes i Google Cloud och AWS, några av dem R-baserade analysappar. I rollen som DevOps-ingenjör mellan 2022 och 2024 driftsattes och underhölls de med samma konfiguration och övervakning i båda molnen, så att ingen app blev ett specialfall för sig. När något gick fel skedde felsökningen med Podman och felsökningen av orkestreringen med Kubernetes, och analysapparna hölls stabila och reproducerbara i produktion på båda sidor.

Ett annat uppdrag gick åt motsatt håll. För en internationell kund hade en AWS-miljö blivit överdimensionerad och samlat på sig oanvända tjänster, och kostnaden hade vuxit ur proportion till verksamhetens behov. Arbetet gick igenom vilka tjänster som användes, tog bort onödiga tjänster, anpassade den återstående kapaciteten och flyttade arbetslaster från AWS. Det är dessa levererade infrastrukturändringar som underlaget beskriver. Det fastställer inget kostnadsförhållande på jämförbar grund, ingen produktivitetsmätning och ingen flytt utan driftstörningar. Exemplet är användbart för sin metod att undersöka den befintliga miljön och ändra det verksamheten behöver driva, inte som ett löfte om en viss besparing.

Båda fallen kokar ned till samma vana. En tjänst som tar betalt per användning fakturerar allt som har slagits på, så en AWS-miljö förblir frisk när någon fortsätter att jämföra det som körs med det som verksamheten faktiskt använder.

Var det går fel

AWS förlorar sällan data eller ligger nere länge. De dyra incidenterna brukar börja inne i kundens eget konto.

Rotanvändaren gör det dagliga arbetet. Varje konto börjar med en rotanvändare, den e-postadress och det lösenord som öppnade kontot, med obegränsad åtkomst till varje resurs och inställning, inklusive möjligheten att stänga kontot. Att logga in som root för rutinuppgifter innebär att ett enda läckt

lösenord når allt. AWS:s råd är att skydda rotanvändarens inloggningsuppgifter, sätta multifaktorautentisering på dem och spara dem till de få uppgifter som kräver dem, medan det dagliga arbetet sker via federerad inloggning och roller.

Långlivade åtkomstnycklar läcker. Ett nyckelpar för åtkomst fungerar tills någon raderar det, och det färdas lätt: in i dotfilerna på en bärbar dator, en CI-variabel, en containeravbildning, en commit. IAM:s lista över bästa praxis börjar från andra hållet. Människor loggar in via en identitetsleverantör och får tillfälliga inloggningsuppgifter, och arbetslaster tar tillfälliga inloggningsuppgifter från en IAM-roll som är kopplad till den beräkningsresurs de körs på. Där en långlivad nyckel verkligen behövs, rotera den, och ta bort nycklar, användare och roller som inte längre används.

Ett budgetlarm tas för en broms. AWS Budgets skickar en avisering när faktiska eller prognostiserade kostnader passerar ett tröskelvärde, och dess siffror uppdateras upp till tre gånger per dag, så ett larm rapporterar pengar som redan är spenderade. Resurserna fortsätter att köras och att faktureras. Budgets kan också utföra en åtgärd, t.ex. tillämpa en IAM-policy eller en service control policy som hindrar att nya resurser skapas, eller agera på specifika EC2- eller RDS-instanser. Det sker bara när åtgärden har konfigurerats i förväg, med en roll att agera genom.

Allt ligger i en enda tillgänglighetszon. En enda instans med en enda disk i en zon är den billigaste uppställningen, och den delar öde med just den platsen. Zoner finns för att ett fel i ström eller nätverk i en zon ska lämna de andra igång, och AWS:s dokumentation rekommenderar att en applikation sprids över flera. En databas som spelar roll får en standby i en andra zon, och ett webblager får instanser i minst två, bakom en lastbalanserare som spänner över båda.

En bucket som öppnades för en demo förblir öppen. Nya S3-buckets blockerar publik åtkomst som standard, men en bucket-policy eller en behörighet på ett objekt kan ändra det på en minut, och en länk som delades med en kund för en eftermiddag är fortfarande publik ett år senare. S3 Block Public Access, inställt för hela kontot eller organisationen, går före de policyerna. Slå på det överallt där ett konto inte har något skäl att publicera filer, och servera publika filer från en bucket som är byggd för just det.

Läs mer

- AWS Documentation — registret över användarguider, API-referenser och handledningar för varje tjänst i AWS.
- Shared responsibility — säkerhetspelaren i Well-Architected och dess redogörelse för vilka lager AWS säkrar och vilka kunden säkrar.
- The pillars of the framework — de sex områden som AWS vill att en arbetslast granskas mot, bl.a. kostnad och tillförlitlighet.
- AWS Regions and Availability Zones — hur regioner och zoner hålls isolerade, och varför en applikation bör spänna över flera zoner.
- Security best practices in IAM — federering, roller, multifaktorautentisering, rotanvändaren och minsta privilegium, i en och samma lista.
- Managing your costs with AWS Budgets — budgetar för faktiska och prognostiserade kostnader, deras aviseringar och de åtgärder de kan köra.

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 AWS?” 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/aws/