Engineer

Azure

Microsofts cloud-platform

Hvad det er

Azure, fuldt ud Microsoft Azure, er et katalog af computerkraft, som Microsoft udlejer fra sine egne datacentre: virtuelle maskiner, lagerkonti, administrerede databaser, hosting af containere, private netværk og værktøjerne til at drive dem. Azures tjenester betales efter forbrug. Microsoft solgte det hele som Windows Azure indtil april 2014, hvor det fik sit nuværende navn.

Hver anmodning om at administrere Azure-ressourcer, hvad enten den begynder som et klik i portalen, en kommando i Azure CLI eller PowerShell, et kald fra et SDK eller en template, ankommer til den samme hoveddør: Azure Resource Manager. Den godkender anmodningen, kontrollerer rettighederne og sender den videre til den tjeneste, der udfører arbejdet. En virtuel maskine, en databaseserver og en DNS-zone er alle ressourcer i dens forstand, hver med et id, en type og et sæt egenskaber.

Ressourcerne er ordnet i et hierarki med fire niveauer. Hver ressource hører til præcis én ressourcegruppe, en mappe til ting, der udrulles, opdateres og slettes sammen; sletter man gruppen, slettes alt i den. Ressourcegrupper ligger i et abonnement (subscription), som på én gang er en faktureringsenhed og en grænse for adgang og kvoter. Abonnementer samles i administrationsgrupper (management groups), der kan indlejres i op til seks niveauer og alle samles under én rodgruppe for kataloget. Rolletildelinger, politikker og budgetter, der sættes på ét niveau, gælder for alt under det.

Over hele hierarkiet ligger identiteten. En organisations brugere, grupper og applikationer bor i en tenant i Microsoft Entra ID, det katalog, Microsoft kaldte Azure Active Directory indtil 2023, og hvert Azure-abonnement har tillid til netop én sådan tenant, der logger folk ind og afgør, hvad de må.

Brug begrebet i en beslutning

Brug »Hvad er Azure?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Servere dimensioneret efter det, I rent faktisk bruger, flyttet op i skyen eller hjem igen, med den månedlige regning regnet ud først.

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 drevet 20 Docker-apps på Google Cloud og AWS — 20 containeriserede applikationer udrullet og overvåget

Har drevet 20 Docker-apps på Google Cloud og AWS Kontakt os →

Hvorfor det betyder noget

De fleste hold, der kører på Azure, er kommet dertil via noget, de allerede havde. En virksomhed, hvis medarbejdere logger ind i Microsoft 365, har allerede sine folk i en Entra-tenant, og et abonnement, der har tillid til den tenant, kan give de samme konti adgang til servere og databaser uden et ekstra brugerkatalog at vedligeholde. For en leverandør afhænger valget af Azure derfor mindre af funktionsoversigter end af, hvor kundens identiteter og data allerede ligger.

Hierarkiet er den anden grund, fordi styringen bor i det. Microsofts vejledning om at organisere ressourcer anbefaler separate abonnementer til udvikling, test og produktion, så miljøerne ikke udvikler afhængigheder af hinanden, og så hvert af dem kan have sine egne politikker og sin egen linje på regningen. Azure Policy håndhæver derefter husets regler på det niveau, politikken er tildelt. Den kan afvise en ændring, for eksempel en ressource oprettet i en region, organisationen ikke bruger, eller logge ændringen, så nogen kan se på den, og Microsoft råder til at starte en ny politik i audit-tilstand, før den får lov at blokere noget.

Den tredje grund er, at Azure kan beskrives frem for at klikkes sammen. Resource Manager tager imod deklarative templates, skrevet i JSON eller Bicep, og regner selv ud, hvad der skal oprettes eller ændres. Kører man den samme template to gange, bliver resultatet det samme, og Azure gemmer selv tilstanden for det udrullede, så der ikke er en separat state-fil at passe på. Et miljø, der er skrevet ned på den måde, kan reviewes som en pull request, genopbygges efter en dårlig dag og kopieres til et nyt miljø ved at ændre parametrene. Det er det vigtigste værn mod en Azure-opsætning, som kun én person forstår.

Sådan ser det ud i praksis

Siden om teknologistakken nævner Google Cloud og AWS som virksomhedens faste clouds, med Azure, hvor kunden allerede er. To forløb i porteføljen viser Azure-arbejdet fra hver sin ende.

På infrastruktursiden blev tre Azure-miljøer skrevet ned som kode, alle fra de samme Bicep-templates: api'et og www kørende som Azure Container Apps på et managed environment, en PostgreSQL Flexible Server, Redis til caching, Front Door med en WAF-politik foran og et VNet, NSG og private DNS nedenunder, med Log Analytics koblet på til diagnostik og images hentet fra projektets Azure Container Registry. Fordi templates er parametriserede, er et nyt miljø eller en ændring af et eksisterende en pull request. Afvigelser fra koden holdt op med at være et mysterium, fordi koden er kilden til sandhed.

På releasesiden blev deployment til Azure ejet fra ende til anden. Releases følger en fast rute, udvikling, så staging, så produktion, frem for at nogen pusher direkte til et live miljø. Udrulningspipelinen bygger og sender images, og Bicep-templates sørger for, at målinfrastrukturen i Azure er den samme i hvert miljø. Konfiguration, der er forskellig pr. miljø, holdes adskilt fra secrets, så det samme byggede artefakt bevæger sig gennem alle miljøer og samler de rette indstillinger op, hvor det lander, uden manuelle ad hoc-deploys undervejs.

Hvor det går galt

De dyre fejl i Azure har ofte samme form: en kløft mellem det, nogen tror, Azure gjorde eller tillod, og det, Resource Manager faktisk gjorde.

En template-udrulning efterlader ting og sætter ting tilbage. Resource Manager udruller som standard i inkrementel tilstand. I den tilstand bliver en ressource, der findes i ressourcegruppen, men mangler i templaten, stående, så når en database fjernes fra template-filen, kører serveren videre og koster penge. Den samme tilstand genanvender hver egenskab for de ressourcer, den udruller igen, og en egenskab, som templaten udelader, nulstilles til standardværdien. En indstilling, nogen har ændret i portalen, bliver derfor overskrevet ved næste rutinemæssige release. What-if-forhåndsvisningen kan advare om begge dele, men Microsoft gør opmærksom på, at dens output kan indeholde støj, falske positiver, som en reel ændring let drukner i. Templaten

bør angive hver værdi, der afviger fra standarden, og en deployment stack, der frakobler eller sletter de ressourcer, som falder ud af en template, tager sig af resterne.

Alle miljøer deler ét abonnement. Det er den hurtigste måde at starte på: ét abonnement, en ressourcegruppe pr. miljø og brede roller tildelt øverst, så ingen bliver blokeret. Produktion og test deler så rolletildelinger, politikker, kvoter og én regning, og et script, der er rettet mod testgruppen, kører med rettigheder, der også når produktion. Separate abonnementer under en administrationsgruppe giver hvert miljø sin egen adgang, sine egne regler og sine egne omkostninger.

En katalogadministrator bliver taget for en ressourceadministrator. Microsoft Entra-roller som Global Administrator styrer kataloget: brugere, grupper, adgangskoder og domæner. Azure-roller som Owner og Contributor styrer ressourcer. Som standard rækker ingen af de to sæt ind i det andet. En Global Administrator kan slå en kontakt til, der giver vedkommende rollen User Access Administrator på rodniveau og dermed lov til at se alle ressourcer og tildele adgang i ethvert abonnement i tenanten. Kontakten findes for at genvinde adgang til et abonnement, og Microsofts vejledning er at fjerne den forhøjede adgang, når arbejdet er gjort. Bliver den stående, er det en katalogkonto, der kan give sig selv adgang til enhver ressource.

En container-app, der aldrig sover. Azure Container Apps kan skalere de fleste applikationer ned til nul replikaer, når ingen kalder dem, og det er en del af tiltrækningen for små tjenester. Applikationer, der skalerer på CPU- eller hukommelsesbelastning, er undtagelsen og kan ikke skalere til nul. En app med en CPU-regel holder mindst én replika kørende hver stille nat, så en tjeneste, der er valgt, fordi den kan stå billigt i tomgang, bør skalere på HTTP-trafik eller en anden hændelseskilde.

Læs mere

- Azure documentation — Microsofts oversigt over vejledninger, quickstarts og referencer til hver Azure-tjeneste.
- What is Azure Resource Manager? — laget, som administrationsanmodninger passerer igennem, og hvordan ressourcegrupper bør afgrænses.
- Organize your Azure resources — Cloud Adoption Framework om administrationsgrupper, abonnementer og ressourcegrupper.

- Deployment modes — hvad inkrementel tilstand lader køre, og hvilke egenskaber en ny udrulning nulstiller.
- Deployment stacks — at styre en templates ressourcer som én enhed, også dem, der er fjernet fra den.
- Azure roles and Microsoft Entra roles — hvor katalogroller slutter, hvor ressourceroller begynder, og hvordan forhøjet adgang forbinder 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 Azure?« 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/azure/