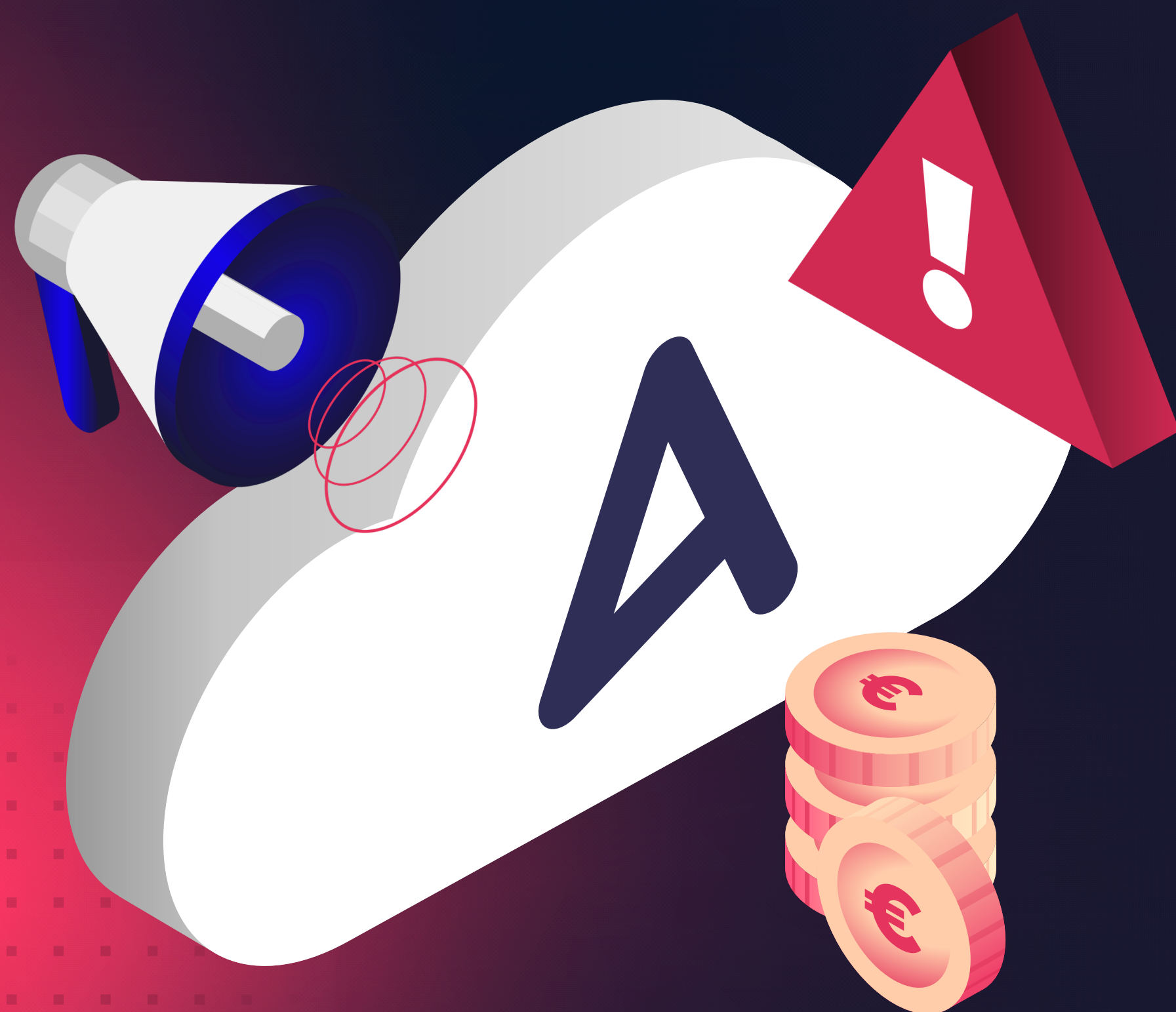




15 Grootste Fouten in Azure

Voorkom dure fouten & leer wat je wél moet doen



Introductie

Welkom!

Microsoft Azure is een uitstekend cloud platform, maar zelfs kleine misstappen kunnen snel uitgroeien tot grote problemen.

Bij Intercept zijn we al jaren actief op het gebied van Azure. In die tijd hebben we veel misgaan zien gaan en herhaalde slechte beslissingen genomen die klanten tijd en geld hebben gekost.

Daarom hebben we een lijst samengesteld met de meest voorkomende Azure mistakes, zodat jij ze niet (nog eens) maakt.

Laten we beginnen!





Fout 1: Subscriptie

Eén subscriptie voor alles

Het gebruiken van één subscriptie voor al je omgevingen (productie, staging, development, etc.) en applicaties doet uiteindelijk meer kwaad dan goed.

De oplossing

Met maar één subscriptie wordt access beheren lastig, en het is makkelijk om developers te veel controle te geven, wat kan leiden tot per ongeluk wijzigingen of deletes in productie. Azure biedt enorm veel RBAC opties, al kunnen die je niet volledig beschermen als alles in die ene box leeft: **de subscriptie**.

De oplossing? Je raadt het al; **aparte subscripties per omgeving**.



Resources organiseren

Hoe je je cloud resources organiseert hangt af van je use case, maar sommige stappen gelden in de meeste scenario's. Als je het goed opzet, blijft je omgeving netjes, kostenefficiënt en makkelijker te managen:

1. Aparte subscripties per omgeving

Zoals Microsoft aanbeveelt: gebruik aparte subscripties voor development, testen en productie. Dit houdt omgevingen geïsoleerd, verlaagt risico, en helpt kosten per omgeving te tracken.

2. Resource Groups per app per omgeving

Maak een Resource Group voor elke app in elke omgeving. Dit houdt resources georganiseerd, maakt access management eenvoudiger, en maakt monitoring en cost tracking makkelijker.

3. *Service Groups om resources over subscripties heen te organiseren

Zoals fancy tags: hiermee kun je resources groeperen over subscripties en resource groups (bijv. "Production apps", "PCI Scope", "MyApp").

**Service Groups is een nieuwe feature sinds mei 2025. Leer meer [hier](#).*

Volg deze tips bij je volgende app, en migreer bestaande resources gaandeweg.





Fout 2: ClickOps Nachtmerrries

Alles bij elkaar klikken

Als je ooit resources in Azure hebt gedeployed, weet je dat de Azure Portal heel intuïtief is en makkelijk om mee te starten. Maar ClickOps schaal niet en zorgt op de lange termijn voor meer problemen. Weet je nog (of documenteer je) alle dingen die je aanklikt?

Naarmate je systemen groeien, wordt het al snel een “mission impossible” om elke klik te onthouden of te documenteren. Het resultaat? Mistakes en inconsistente omgevingen.



De oplossing

Infrastructure as Code (IaC)

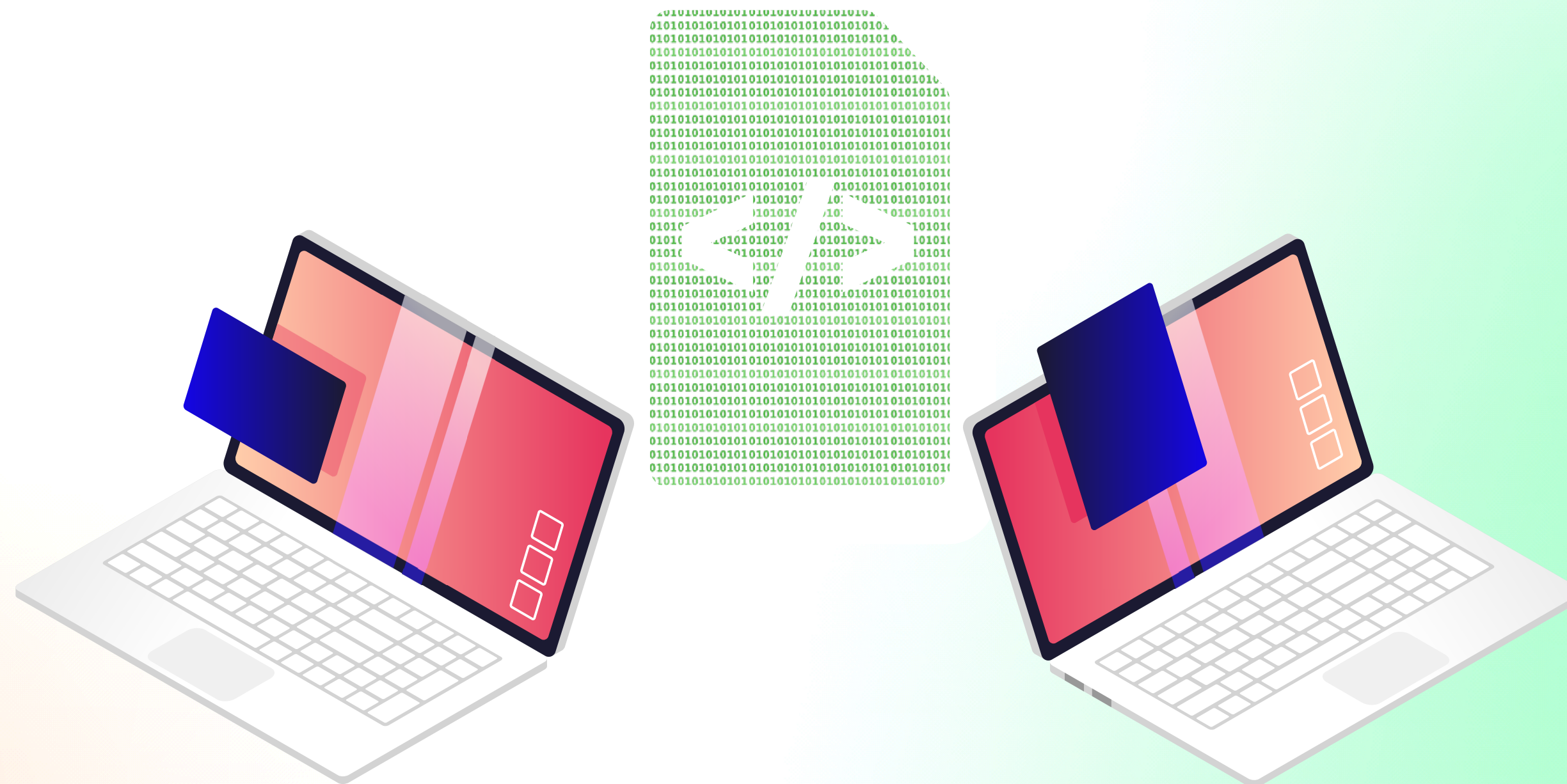
Vertrouwen op de Azure Portal om infrastructure handmatig te deployen en te managen brengt risico's met zich mee en schaal niet. Ga in plaats daarvan weg van ClickOps naar Infrastructure as Code; een betrouwbaardere en efficiëntere manier om je infrastructure te beheren.

Er zijn veel IaC tools breed beschikbaar om consistentie, één single source of truth, en vereenvoudigde documentatie te bereiken. Alles bij elkaar maakt IaC het managen en schalen van je infrastructure efficiënter.



Wil je weten welke IaC tool je voor je workloads moet gebruiken? Kijk dan naar de verschillen tussen de beschikbare tools zoals Azure Bicep, ARM, Terraform, en meer [hier](#).

Het handmatig implementeren en beheren van infrastructuur via de Azure Portal brengt risico's met zich mee.





Fout 3: Cost Management

Voorkom dure rekeningen

Plotselinge, onverwachte kosten. Je opent je Azure-factuur en knippert.

Hebben we echt zóveel uitgegeven?

Laten we eens kijken waarom en dat oplossen!

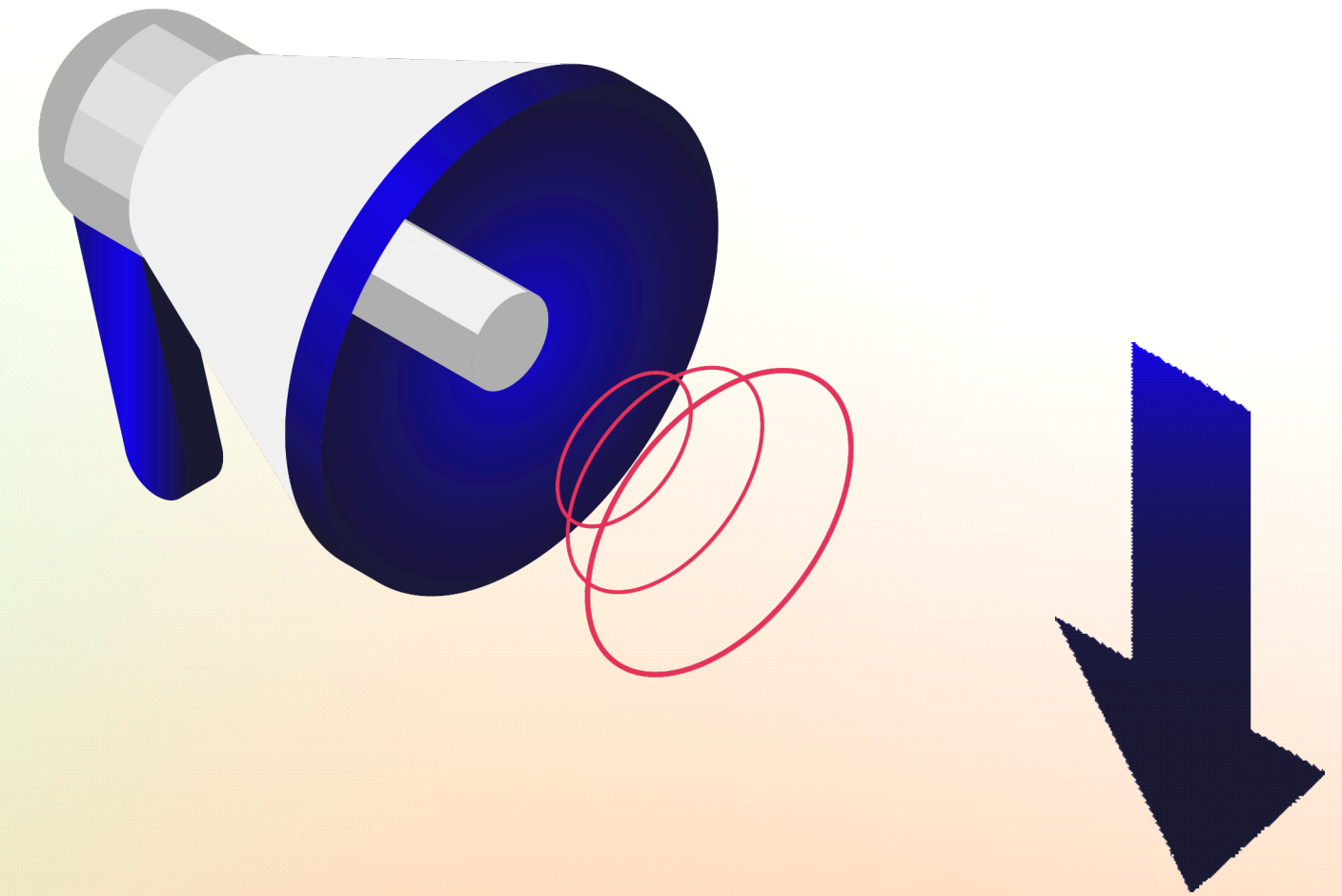


De oplossing

Je teams rollen nieuwe resources uit en zetten services aan; en als je dat niet strak beheert, dan krijg je:

- Geen vroege waarschuwing als de voorspelling wordt overschreden.
- Te weinig inzicht tussen teams.
- Afwijkingen en fraude die je mist.

Het resultaat? Je Azure-omgeving loopt snel uit de hand en de kosten schieten omhoog.



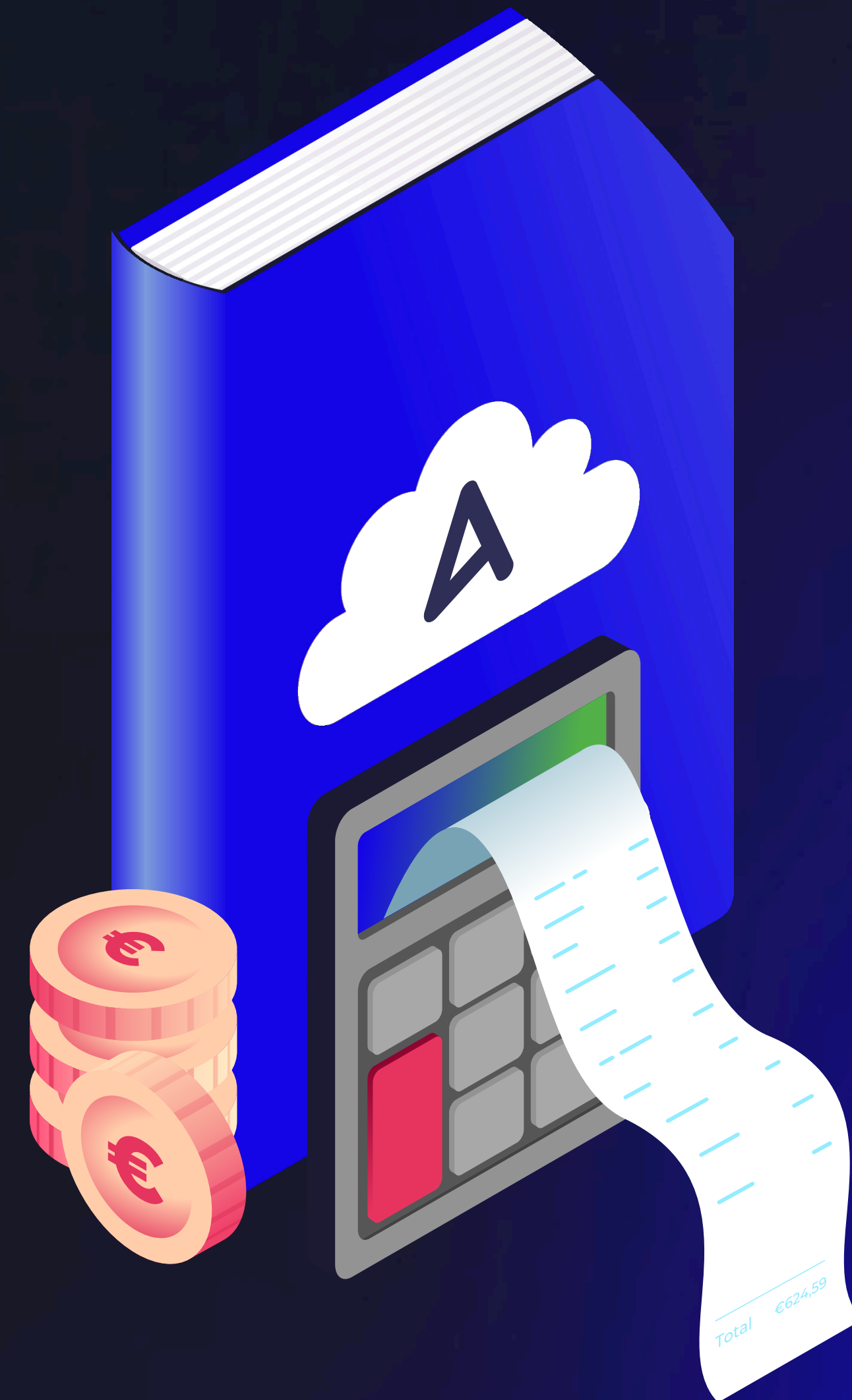
- Definieer een limiet voor uitgaven voor een subscriptie of resourcegroepen.
- Stel meerdere limieten in (50%, 80%, 100%) om vroegtijdige waarschuwingen te krijgen.
- Houd de werkelijke of verwachte kosten bij in de loop van de tijd; maandelijks, per kwartaal of per jaar.
- Trigger Action Groups (zoals e-mail, Functions of Webhooks) wanneer thresholds worden bereikt

Van alle tips is deze wellicht de belangrijkste: configureer niets voordat je Azure budgetten en -alerts hebt ingesteld.

Leer hoe je kosten kunt beheren en optimaliseren in het **Azure Cost Management Whitepaper**, met informatie over Azure-tools, kortingsmethoden en best practices uit de praktijk.

Wil je een lagere rekening voor Azure? Ontdek hoe.

[Download nu!](#)





Fout 5: Tags

Vergeten, of één grote chaos

Naarmate er meer resources worden toegevoegd en door meer mensen worden beheerd, wordt het verleidelijk om te vergeten 'wie wat heeft gemaakt'. Daarom heb je tags nodig in Azure, dit zijn metagegevens die je aan je resources toevoegt. Met deze key-value-paren kun je resources labelen en identificeren op basis van kenmerken die relevant zijn voor je organisatie.

Inconsistente tags maken het moeilijk om vragen te beantwoorden,

zoals:

→ Hoeveel kost deze app ons per maand?

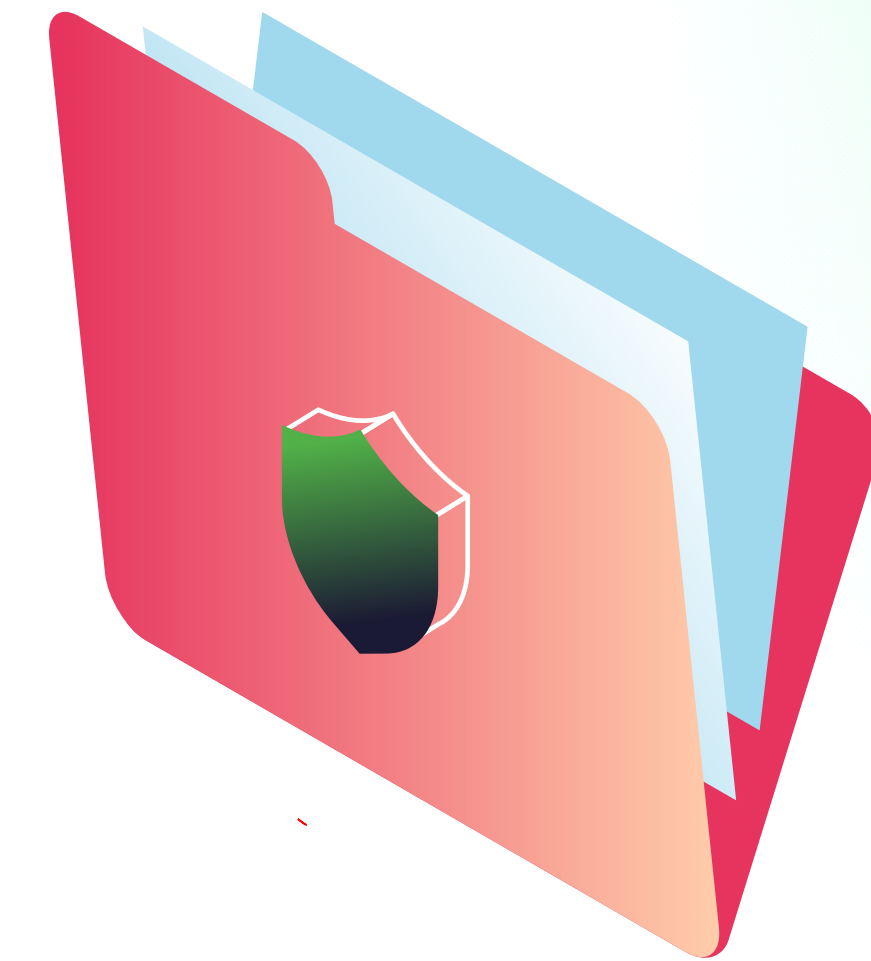
→ Wie is de eigenaar van deze bron?

→ Hoeveel kost de stagingomgeving ons?

De oplossing

Pas tags meteen vanaf het begin toe en implementeer ze op de juiste manier. Stel een taggingstandaard op waarin je een algemene set tags definieert voor elke bron, zoals in deze voorbeelden:

- **De omgeving (dev, staging, prod, enz.)**
- **De eigenaar (contact-e-mailadres)**
- **De bedrijfsunit (financiën, verkoop, enz.)**
- **Applicatie (CRM)**
- **Kostenplaats (globaal)**
- **Serviceklasse (Tier 1, Tier 2, enz.)**





Fout 5: Naamgeving

Noem het maar wat... en krijg
spijt

When you first start your journey in Microsoft Azure, it's tempting to name resources however you like. But the problem is that soon our naming has no consistency across teams, apps and divisions. Consequently, it becomes hard to find stuff, slowing down troubleshooting. Not to forget about the added confusion, and the painful and challenging management of resources.

De oplossing

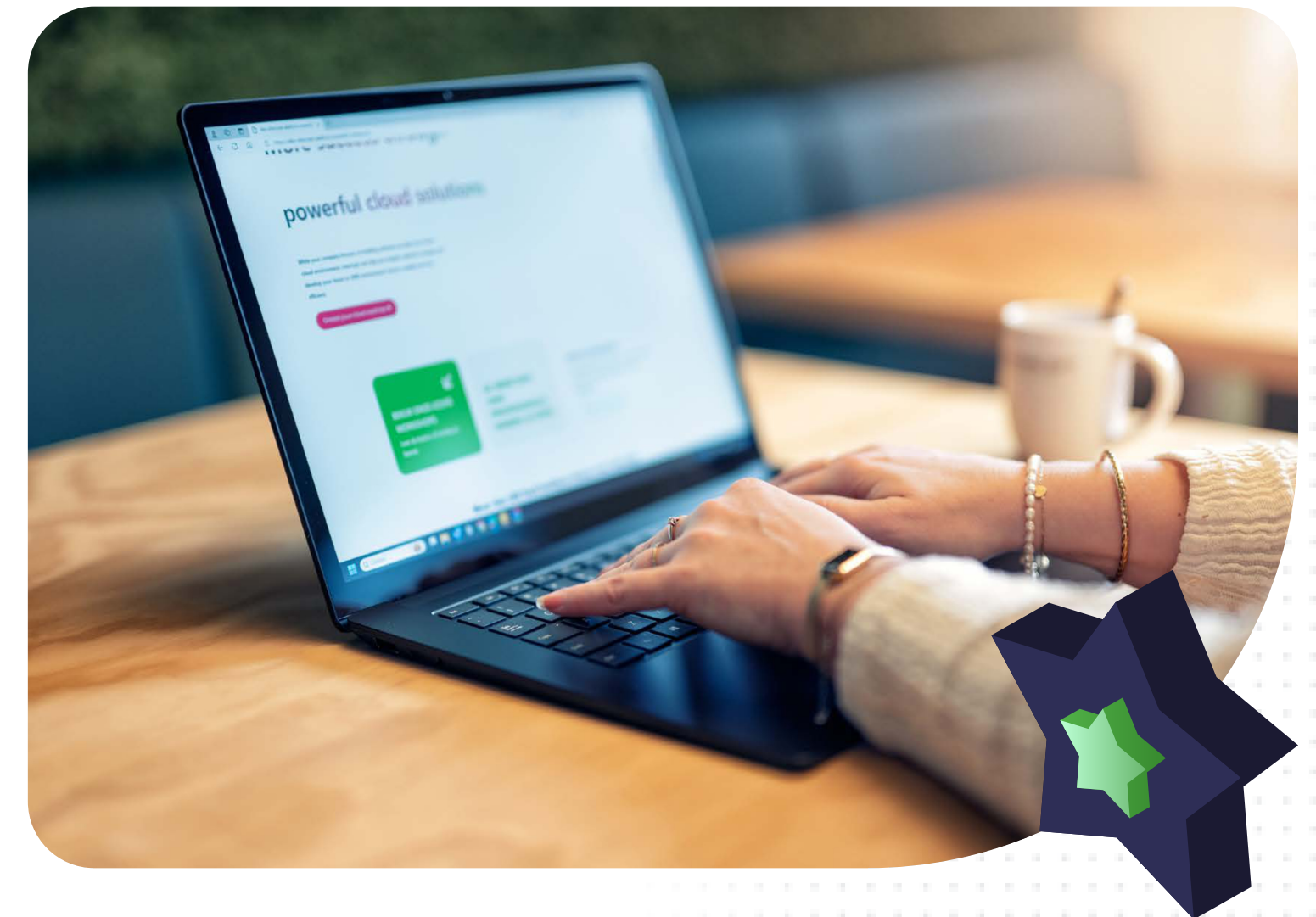
Implementeer een naming standaard:

- Volg de door Microsoft aanbevolen naamgevingsconventies en afkortingen. Een voorbeeld is rg-webapp-database-dev.
- Gebruik de Azure Naming Tool om consistente namen te genereren.

Het belangrijkste hierbij is om een naamgevingsconventie te kiezen en je daar vervolgens aan te houden. Je kunt dit zelfs afdwingen met behulp van Azure Policy (dat we hierna uitleggen) voor consistentie tussen teams.

Let op:

Niet alle resource-namen zijn toegestaan;
bijvoorbeeld namen met koppeltekens in store-
accountnamen.





Fout 6: Azure Policy

Afspraken niet handhaven

Een veelvoorkomende fout: organisaties hebben documenten met standaarden, maar niemand volgt ze echt. Teams kiezen zelf wat ze toepassen en voor je het weet is je omgeving de “Wild West”. Zonder handhaving worden naamgeving, tags, enz. genegeerd → en krijg je “cloud chaos”.



De oplossing

Voorbeeld

Als je deployments niet beperkt op regio of workload type, dan kan iemand een resource in een dure regio uitrollen of een veel te grote VM kiezen. Voor je het weet stijgen kosten, komt compliance onder druk en wordt beheer een nachtmerrie.

Gebruik Azure Policy om naamgeving en tags af te dwingen. Policies kunnen verschillende effecten hebben:

→ **Meerdere versies tegelijkertijd ondersteund**

Voorkom dat non-compliant resources worden aangemaakt (aanbevolen).

→ **Audit**

Markeer resources als non-compliant.

→ **Wijzig**

Wijzig resources automatisch (niet aanbevolen).

Het handhaven van beleid is echter niet alleen bedoeld om fouten te voorkomen, maar ook om resources in te richten zoals het hoort:

→ HTTPS afdwingen in plaats van HTTP

→ SQL Data Encryption at rest aanzetten



Fout 7: Geen monitoring en alerting

Als alles er “prima” uitziet

Als je resources uitrolt, moet je monitoring en alerts niet vergeten. Als er iets misgaat wil je het weten vóórdát je gebruikers het merken.

De oplossing

→ **Monitor**

Volg proactief de gezondheid en prestaties van je applicaties en resources met Azure Monitor.

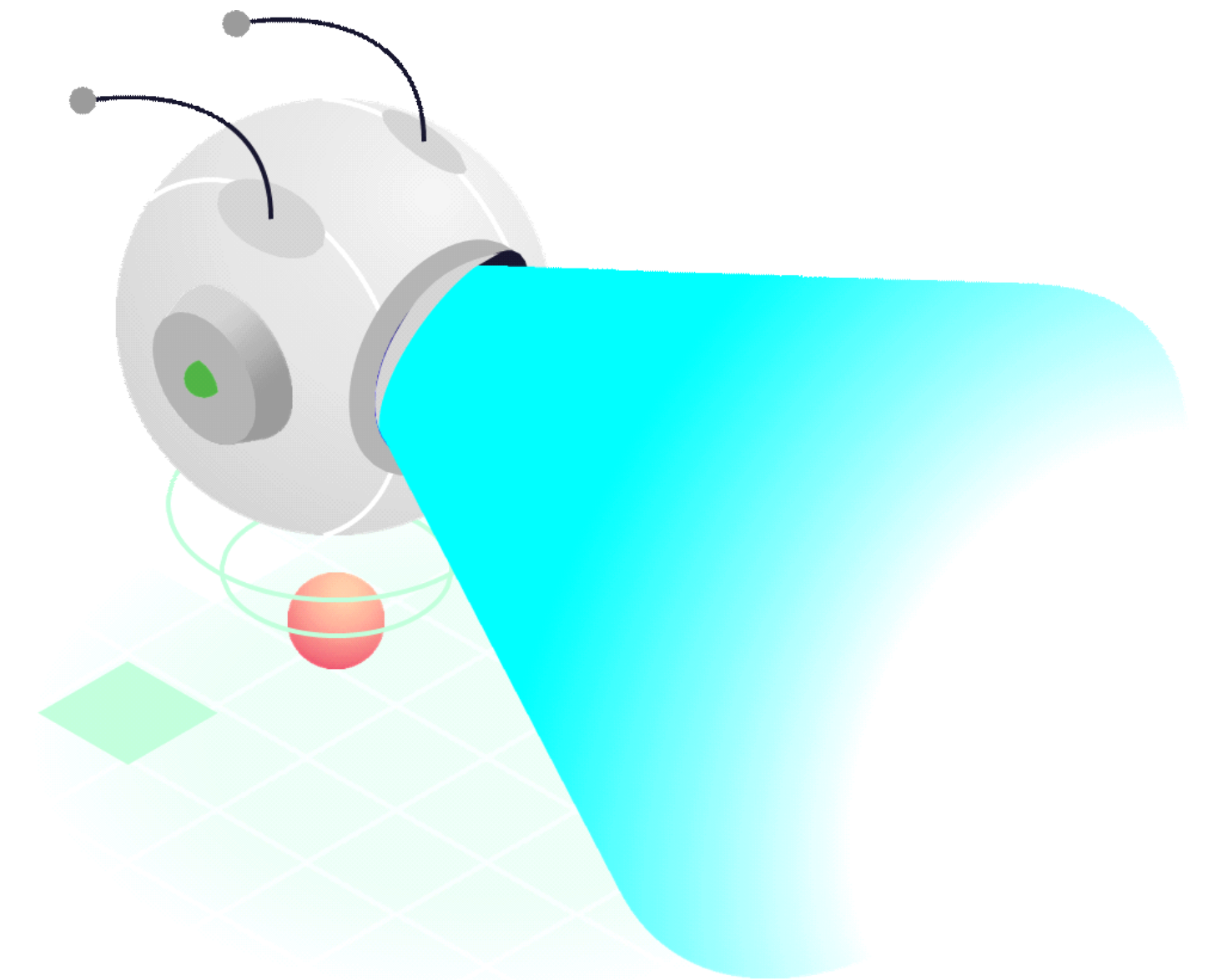
→ **Alerts**

Stel alerts in op metrics en logs die ertoe doen.

→ **Notificaties**

Laat meldingen binnenkomen via e-mail, SMS of andere kanalen.

Zo kun je snel reageren, blijven je applicaties gezond en blijven gebruikers tevreden.





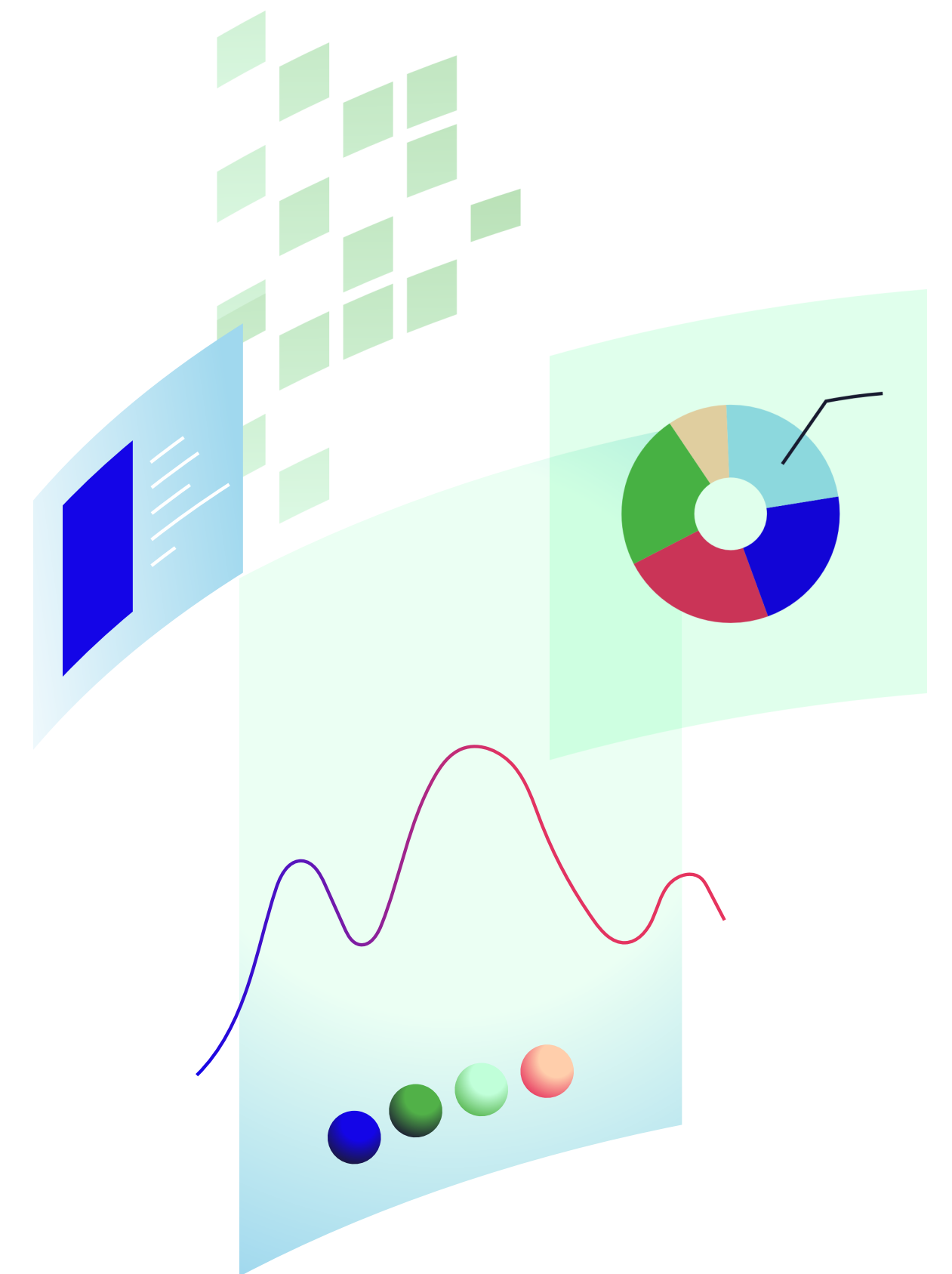
Fout 8: Ongefilterde ingestie van logbestanden

Eerst alles opslaan, later spijt

We zien dit vaak bij developers: te veel logs naar Log Analytics sturen. Als alles rechtstreeks naar Log Analytics gaat, kan gevoelige informatie meekomen en kan het datavolume (en dus de kosten) snel ontploffen.

De oplossing

- Neem een sample van log data om volume te beperken; [bij voorkeur vóór het naar Application Insights](#) gaat.
- [Stel een dagelijkse workspace cap in in Log Analytics](#) om onverwachte kostensprongen te voorkomen (handig bij pieken of veel errors). Let op: zodra de cap is bereikt, worden extra logs niet vastgelegd → dataverlies.
- Volg ook [Microsoft's aanbevelingen voor logging](#).



Fout 9: Idle resources

Zombie resources

In Azure is het heel makkelijk om resources aan te zetten, te groot te kiezen of zelfs te vergeten dat ze bestaan... Orphan resources zijn resources die ooit door iemand zijn gemaakt, maar niemand weet nog door wie.



Het probleem

Vaak blijven ze achter als een deel van een systeem is verwijderd. Typische voorbeelden:

- IP-adressen die niet meer aan machines hangen
- Virtuele machines (VM's) die niet zijn gedeallegeerd
- Losse schijven die gewoon door blijven tellen
- Verwijderde VM's met back-ups die nog niet zijn verlopen; waarvoor je blijft betalen
- Lege abonnementen en resourcegroepen.

Orphaned resources kosten niet altijd geld, maar ze kunnen dat wel. En ze zorgen sowieso voor rommel, soms zelfs een veiligheidsrisico.



De oplossing

- ✓ Gebruik Azure Cost Advisor om orphaned/underutilised/idle resources te vinden.
- ✓ Gebruik Azure Resource Graph om te zoeken naar resources die niet attached zijn of een vreemde status hebben.
- ✓ Ruim dit regelmatig op om kosten te verlagen, beveiliging te verbeteren en je omgeving schoon te houden.





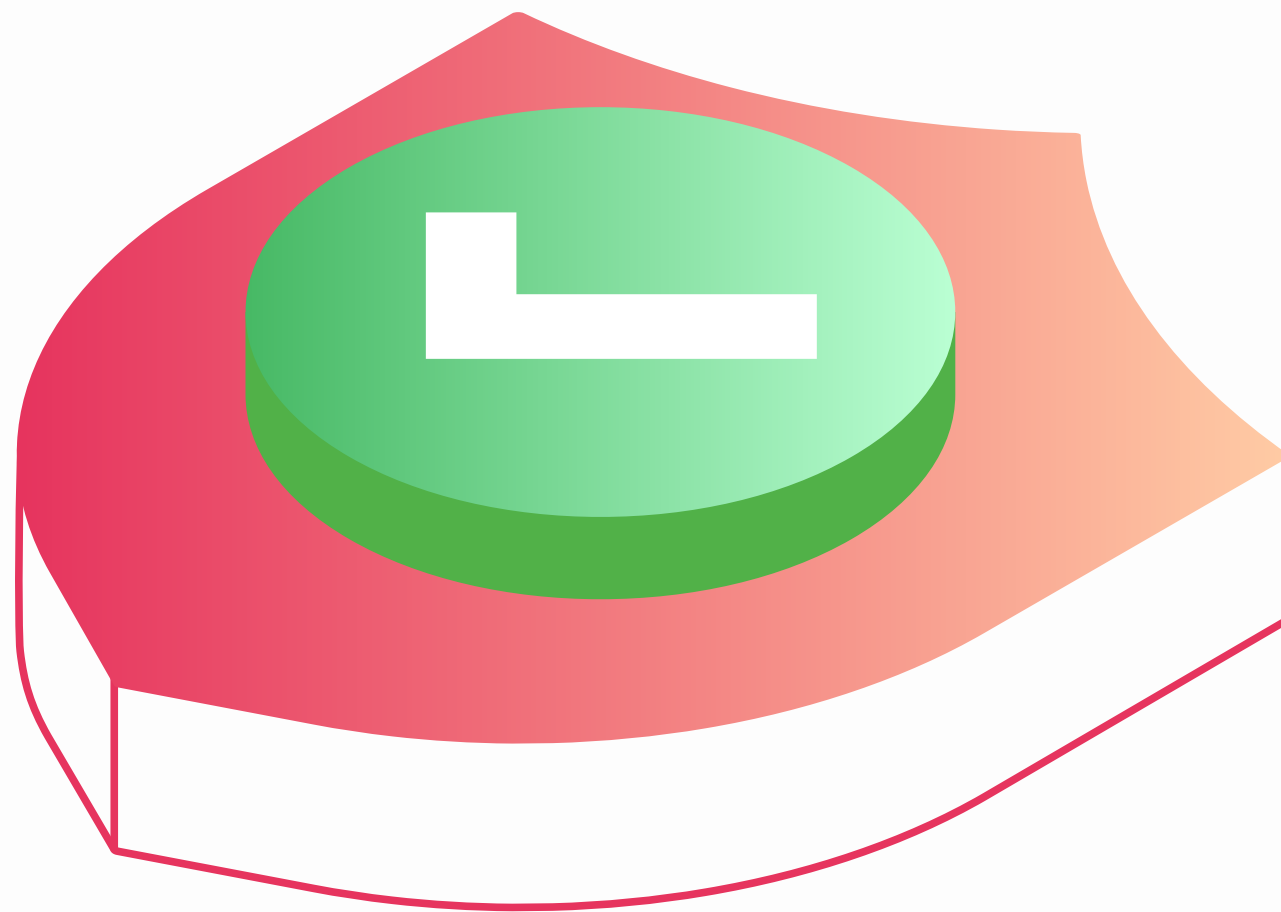
Fout 10: Geen RBAC-strategie

Te veel mensen met te veel rechten

We zien vaak dat “iedereen” hoge rechten krijgt in Azure, met Owners overal. Wat begon als gemak wordt risico. In plaats van least privilege krijg je een omgeving waar één verkeerde klik productie kan slopen.

De oplossing

Werk volgens least privilege en een Zero Trust aanpak: ga uit van “niemand standaard vertrouwd” en geef minimaal toegang, niet meer dan nodig.



- Gebruik role assignments op de juiste scope (management groep, subscriptie, resource groep, etc).
- Beperk Owner en Contributor tot een paar vertrouwde beheerders.
- Maak custom roles als de standaardrollen te breed zijn.
- Controleer role assignments regelmatig en haal weg wat niet meer nodig is.
- Gebruik Privileged Identity Management (PIM) voor tijdelijke rechten in plaats van permanente toegang.



Fout 11: Managed Identity

Alles behalve veilig

We zien nog steeds te vaak dat teams met inloggegevens rommelen. Je maakt ze aan, slaat ze op, wisselt ze af en trekt ze in, totdat iemand ze vergeet. Azure SQL is nog steeds toegankelijk met gewone SQL-authenticatie (gebruikersnaam + wachtwoord) en Key Vault met een client-ID en geheim. Vroeg of laat geeft iemand per ongeluk een geheim prijs.



De oplossing

Gebruik managed identities in Azure. Hiermee kun je een resource laten draaien met zijn 'eigen' identiteit, die volledig door Azure wordt beheerd. Je hoeft je geen zorgen te maken over gebruikersnamen, wachtwoorden, client-ID's of geheimen.

- Wijs een managed identity toe aan een App Service, functie of andere resource.
- Geef die identiteit de juiste toegang om geheimen te lezen, verbinding te maken met databases en meer.
- Helemaal zonder wachtwoorden.
- Gebruik **DefaultAzureCredential** in je code om authenticatie automatisch af te handelen.





Fout 12: Geen Key Vault references gebruiken

Kopiëren en plakken

Te vaak zien we nog steeds teams die secrets behandelen als configuratie. Key Vault bestaat niet voor niets, maar toch belanden inloggegevens in omgevingsvariabelen en configuratiebestanden, totdat iemand ze naar source control pusht of deelt met iedereen die de app-instellingen kan lezen. Na verloop van tijd wordt die ‘tijdelijke’ snelkoppeling een ernstig veiligheidsrisico..

Wat is Azure Key Vault?

Azure Key Vault is een SaaS service voor het veilig opslaan en benaderen van certificates, keys en secrets.



We kunnen vertrouwelijke gegevens zoals wachtwoorden (in Azure secrets genoemd), cryptografische sleutels, certificaten en encryptiesleutels veilig opslaan op een centrale locatie.

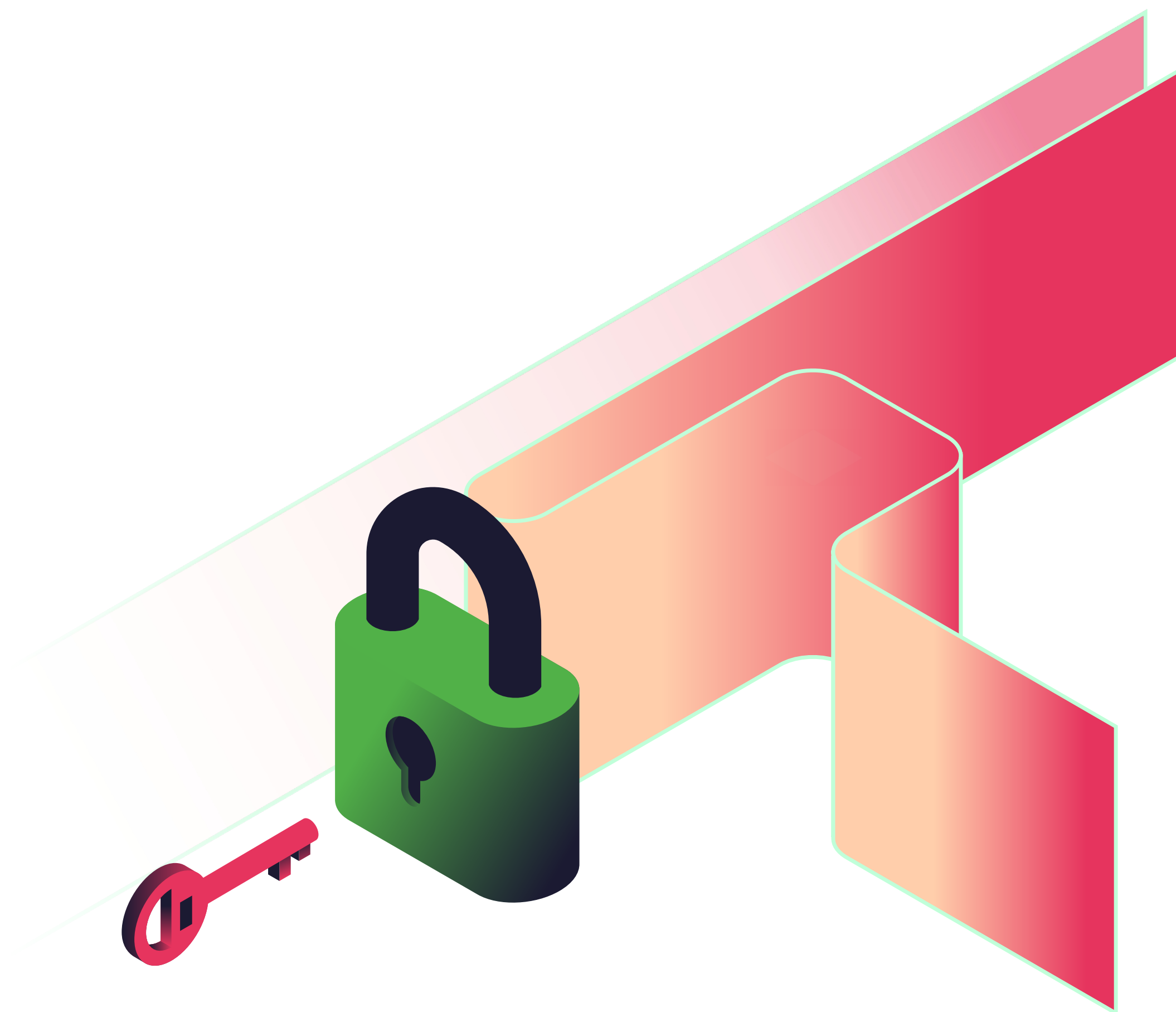
Toch zien we nog steeds dat teams geheimen achterlaten in omgevingsvariabelen of configuratiebestanden. Dit kan leiden tot onbedoelde blootstelling, geheimen die worden vastgelegd in source control, of inloggegevens die toegankelijk zijn voor iedereen met toegang tot de app-configuratie. Na verloop van tijd wordt dit een ernstig veiligheidsrisico.



De oplossing

Gebruik Key Vault References om veilig naar secrets te verwijzen vanuit plekken zonder ze direct op te slaan. Dit houdt betrouwbare en gevoelige data uit code en app settings, terwijl applicaties secrets tijdens runtime kunnen ophalen.

- Kan gebruikt worden met App Services, Functions en andere Azure Services.
- Maakt secret rotatie eenvoudiger; update de secret in Key Vault en apps krijgen automatisch de nieuwe waarde.
- Vermindert het risico op onbedoelde exposure of compromittering aanzienlijk.





Fout 13: Slecht ontworpen netwerken

De deuren wagenwijd open

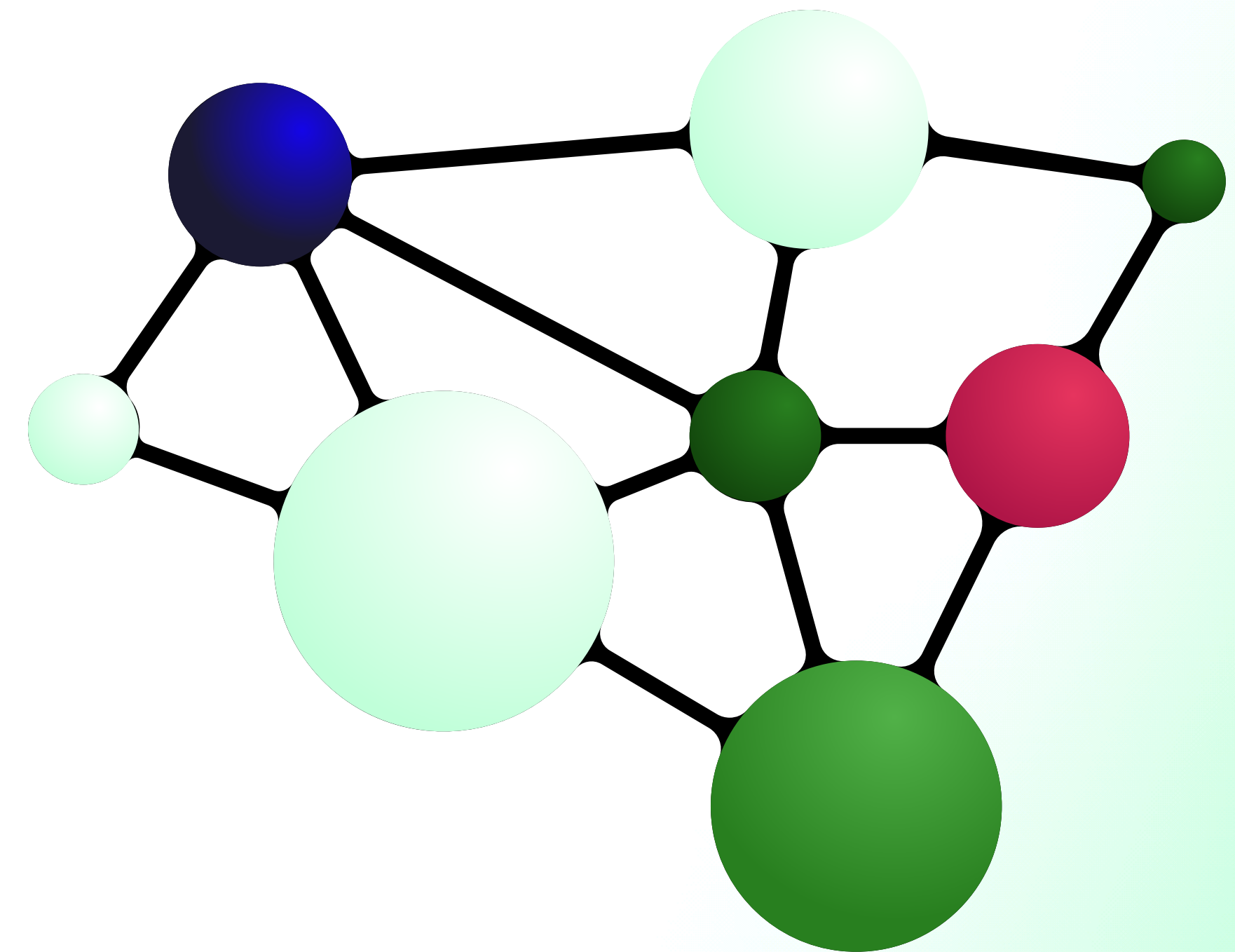
Dit is meer dan alleen een architectuur fout, maar we hebben echt te vaak vlakke network topologieën gezien in Azure. Proberen dat te managen voelt bijna alsof je in het donker door een doolhof loopt.



De oplossing

A better way is to design a hub-spoke model with a firewall: a central point of control and the ability to filter or block unwanted traffic.

- ✓ **Central control:** One place for ingress/egress and shared services.
- ✓ **Consistent filtering:** Inspect, allow/deny, and log traffic centrally.
- ✓ **Strong segmentation:** Isolate apps/environments and reduce blast radius.
- ✓ **Predictable routing:** UDRs force traffic through the firewall/NVA.
- ✓ **Standard patterns:** Repeatable spokes per workload/subscription.
- ✓ **Better visibility:** Clearer flow logs and faster troubleshooting.



Net result: your network becomes easier to manage, more secure, and far less chaotic, because it's designed like a system, not grown like a patchwork.



Fout 14: Overprovisioning van resources

Schalen door te raden

Meer resources alloceren dan je echt nodig hebt is een makkelijke manier om geld te verspillen. We hebben genoeg klanten gezien die niet zeker weten welke SKU ze nodig hebben, dus geven ze VMs extra CPUs en memory “voor de zekerheid” en betalen uiteindelijk voor ongebruikte capacity. Tel daar handmatig schalen bij op en de rekening loopt snel op.

De oplossing

Volg deze tips:

Tip 1: Authenticatie

- **Right-size eerst:** Begin met de SKU die voldoet aan je minimale baseline behoeftes.
- **Gebruik autoscaling:** Laat aanbod je capaciteit bepalen, niet aannames.

Tip 2: De juiste VM kiezen

- **Know your workloads:** Analyse what each application needs in terms of CPU, memory, throughput, and network.
- **Use heatmaps:** Visualise usage over time to spot spikes, trends, and persistent waste.
- **Test different configurations:** Compare smaller and larger sizes to find a cost-effective sweet spot.

Let op:

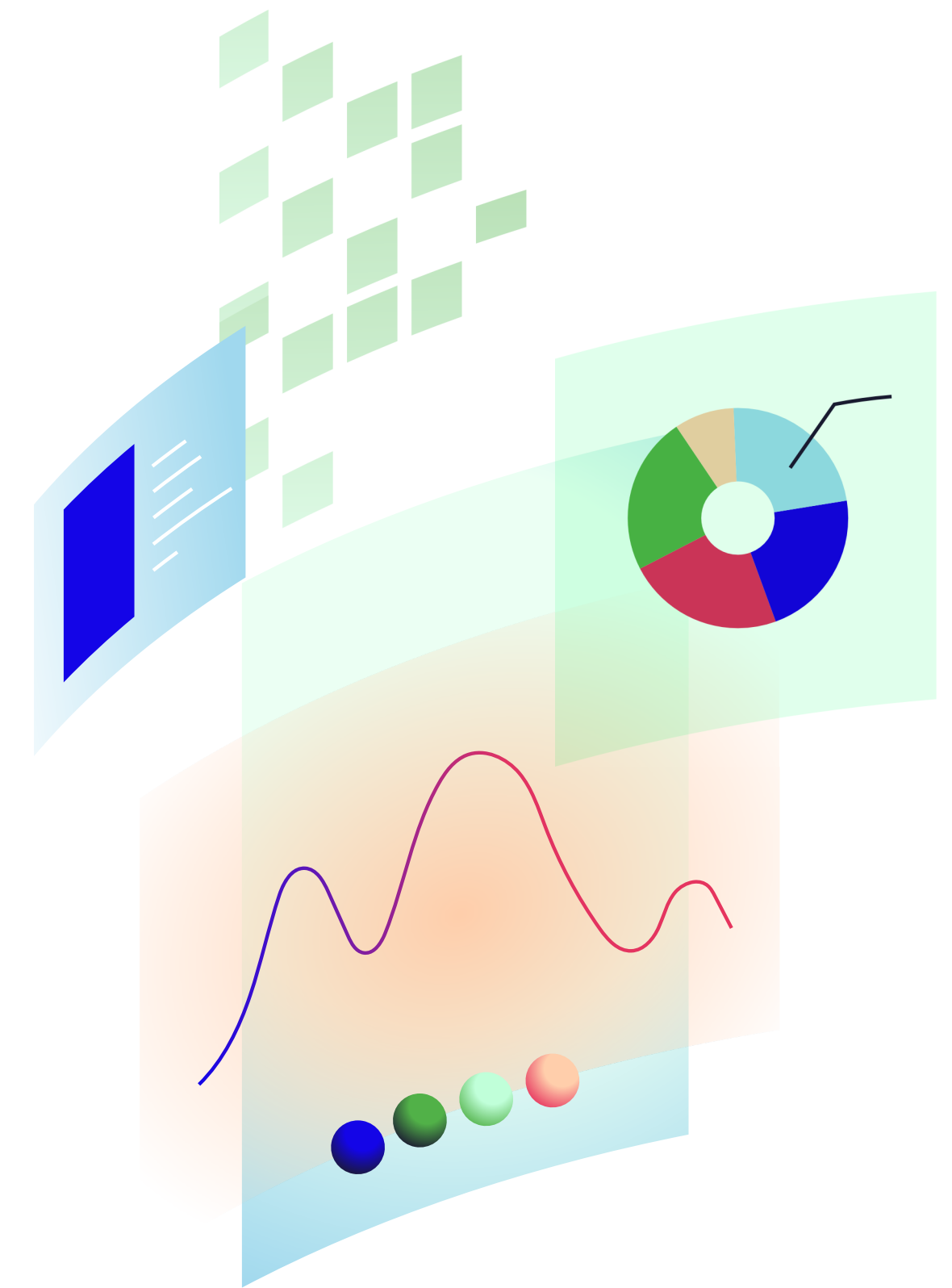
Right-sizing betekent resources matchen op je behoeften, zodat je alleen betaalt voor wat je gebruikt. Een praktische start is ondergebruikte instances identificeren (bijvoorbeeld ~20–30% CPU gebruik) en ze downsize. Dit kan vaak 10–15% op de maandelijkse factuur besparen.

Tip 3: Gebruik VM Scale Sets

- Definieer duidelijke regels voor schalen (metrics + thresholds).
- Combineer met load balancing zodat verkeer wordt verspreid.

Tip 4: Gebruik Azure Advisor

- **Bekijk regelmatig aanbevelingen:** suggesties voor de juiste omvang en kosten veranderen naarmate de workload verandert.
- **Houd besparingen bij:** maak van snelle winsten een blijvende gewoonte.



Fout 15: Backup en Disaster Recovery

Hoop is geen plan

Vraag jezelf af:

- Wat gebeurt er als een natuurramp je Azure region raakt?
- Wat zijn de gevolgen als je omgeving down gaat, of customer data verloren gaat?
- Hoe lang kan je business draaien zonder toegang tot kritieke systemen of data?

Als je bedrijf niet voorbereid is, kan elke ramp onmiddellijke gevolgen hebben voor de bedrijfsinkomsten of zelfs leiden tot faillissement.



De oplossing

- Gebruik Azure Backup om je kritieke data te beschermen. Daarmee kun je backups maken en restoren van je data en resources, om je omgeving te beschermen tegen accidentele verwijdering, corruptie of kwaadwillige aanvallen.
- Gebruik Azure Site Recovery om workloads te repliceren en de bedrijfscontinuïteit te waarborgen in geval van een storing.
- Overweeg geo- en es-zoneredundantie: replicateer gegevens over regio's (GRS, GTM, ASR) en verdeel workloads over beschikbaarheidszones (ZRS, zoneredundante services) om bescherming te bieden tegen storingen op regionaal of zonelevel.
- Behandel back-up en disaster recovery (DR) in Azure als een routineonderdeel van je omgeving door regelmatig te testen.





Tot slot

Azure biedt eindeloze en geweldige mogelijkheden, maar Azure managen brengt ook flink wat valkuilen met zich mee. Sommige van de genoemde common mistakes zijn risky en kostbaar, maar nu weet je hoe je ze voorkomt en wat je in plaats daarvan doet met best practices, wat op zijn beurt kan leiden tot:

- ✓ **Lagere totale Azure kosten**
- ✓ **Minder handmatig werk voor security en account management**
- ✓ **Een betere, meer beveiligde cloud**
- ✓ **Betere app performance**

En de lijst gaat door...





Welke fouten bestaan er in jouw omgeving?

Vraag een Well-Architected Framework Scan aan en ontvang duidelijke, geprioriteerde aanbevelingen over de 5 pijlers, afgestemd op de best practices van Microsoft.

→ Vraag nu een gratis scan aan!

Azure Workshops

Wil je meer leren over Azure? We hebben meerdere gratis online Azure workshops om je te helpen de ins en outs van Azure te begrijpen.

→ Bekijk alle workshops