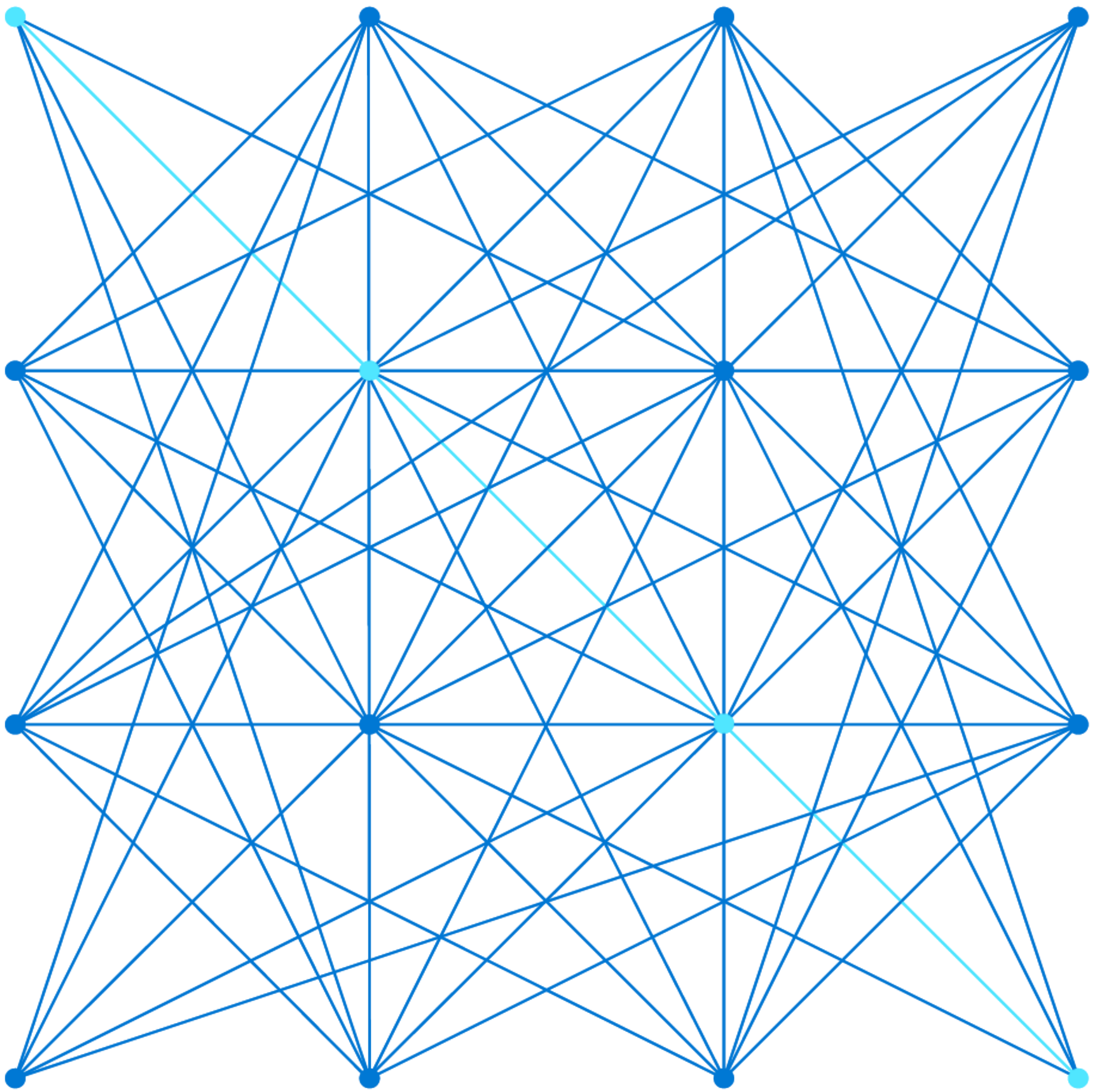
Microsoft Azure

# Azure Architects Connect:

# Sneak Peak – Azure Updates der letzten Monate

Judith Freiberger - Cloud Solution Architect

Timo Knapp – Cloud Solution Architect



# Agenda

- Introduction
- Azure Virtual Network Manager
- Azure cross-region Load Balancer
- Secrets in Azure Container Apps
- Microsoft Dev Box

Microsoft Azure

# Azure Virtual Network Manager

# Customer challenges with network management

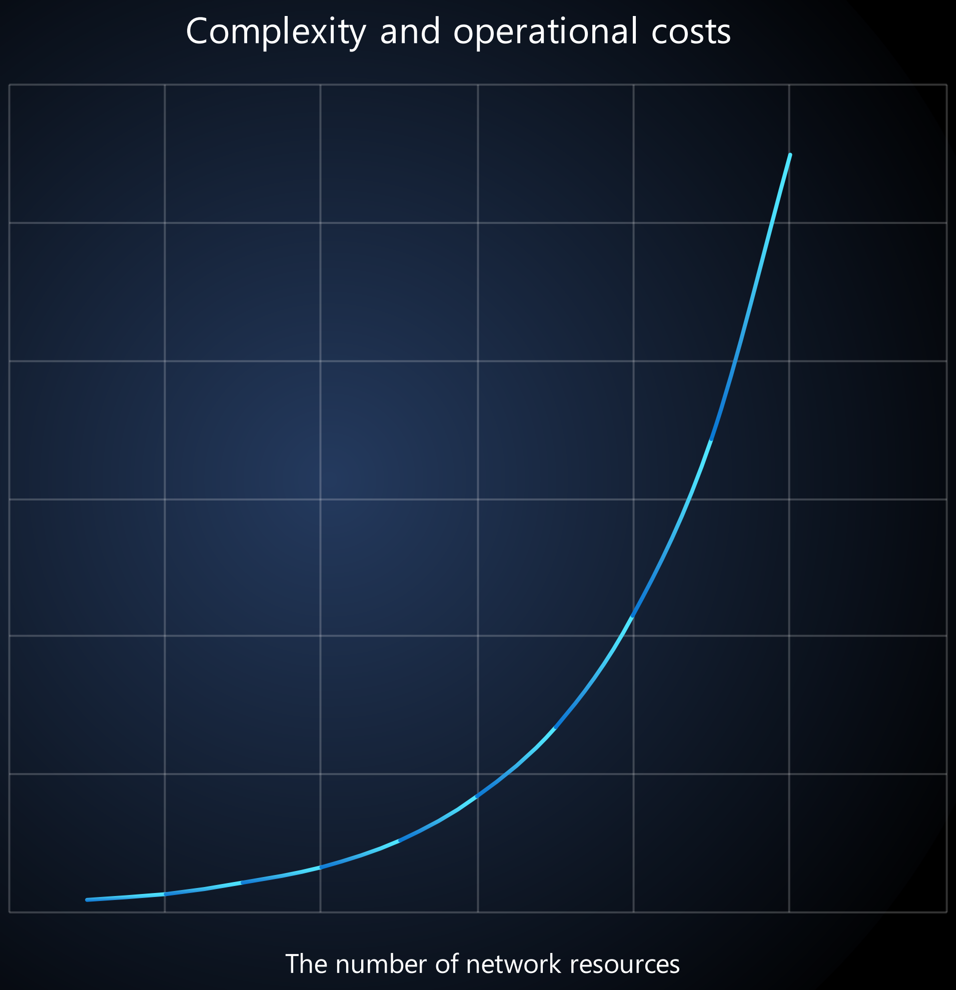
Building networks at scale

Operational overhead and cost

Using multiple solutions

Errors are costly

Re-architecting to adapt to changes



# Azure Virtual Network Manager

## Simplify and centrally manage Azure Networks at scale

### Features

Network segmentation features:

Create network groups to segment network resources by org/function

Define network group across regions and subscriptions

Automatically apply network configurations for changes in network groups

Connectivity configuration features:

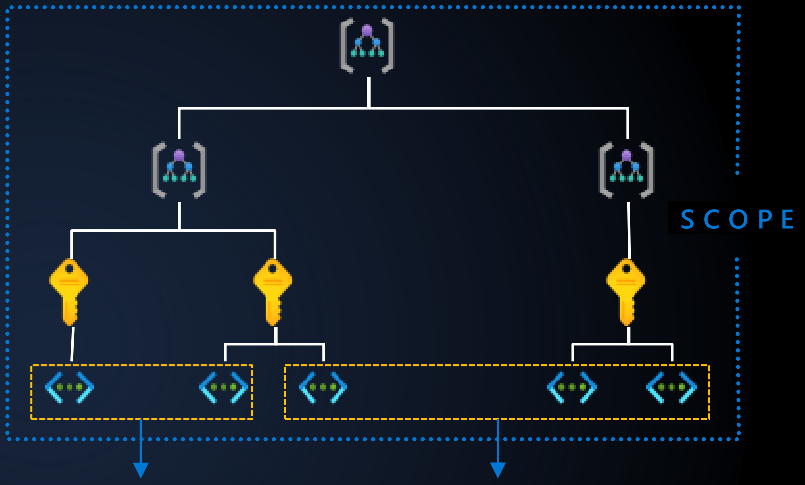
Build and manage complex network topologies

- Mesh
- Hub-and-Spoke/direct connectivity

Security configuration features:

Admin rules

- Enforce organizational level rules without being overwritten
- Apply automatically to old/new resources



Define network group by selecting vNets

Define network group by define criteria (e.g., vNets w/ tag: Prod)

Apply connectivity and security config and more in the future

Deploy the desired config to the regions and roll forward

# Network segmentation features

## Network Group

### Simplified management

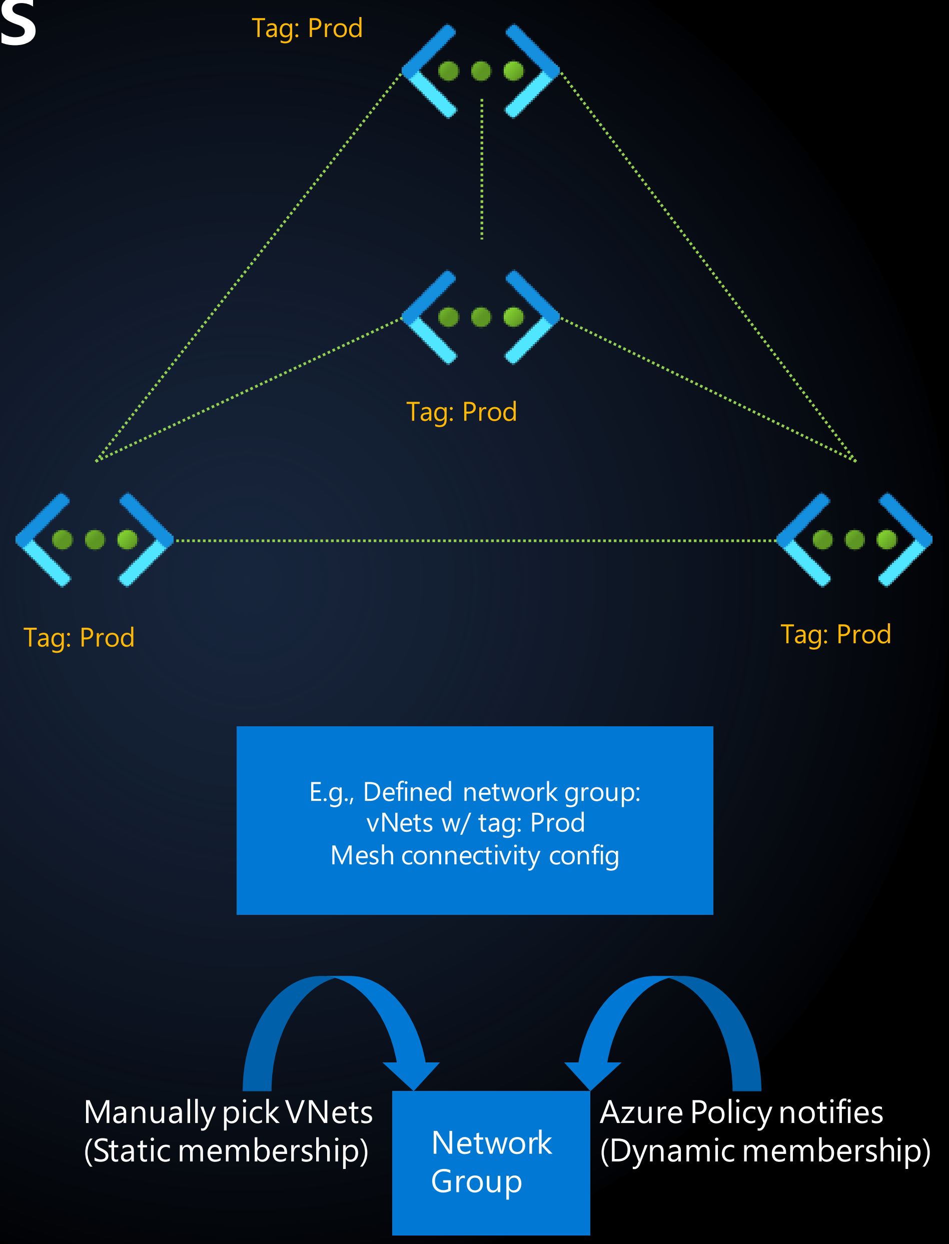
Segment your network into Dev, Prod, Test or by team

Group VNets at subscription, management group or tenant level

Static grouping

Dynamic grouping using name or tags

Apply configurations to your network groups



Tag: Prod

Tag: Prod

Tag: Prod

Tag: Prod

E.g., Defined network group:
vNets w/ tag: Prod
Mesh connectivity config

Manually pick VNets
(Static membership)

Network
Group

Azure Policy notifies
(Dynamic membership)

# Connectivity configuration features

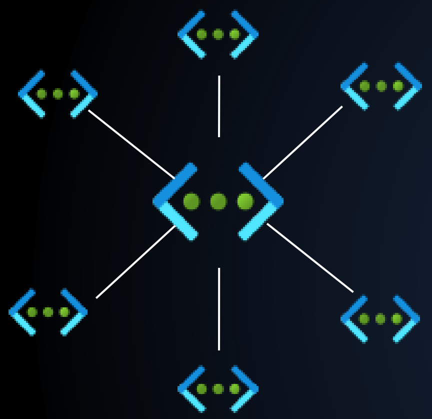
## Create different topologies with a few clicks

- Hub-and-Spoke
- Mesh
- Hub-and-Spoke with direct connectivity

Higher scale topology with 1000+ VNets

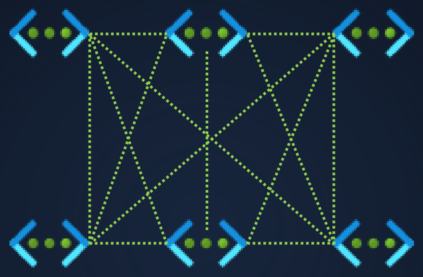
Connectivity across regions, subscriptions, and tenants

Hub and Spoke



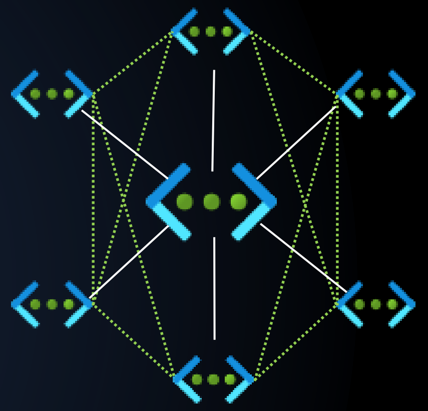
Use case: Gateways (ExpressRoute, VPN Gateways), Firewall, **common infrastructure** shared by spoke virtual networks in the hub

Mesh



Use case: All workloads in the virtual networks can **communicate to each other**

Hub-and-Spoke with direct connectivity between spokes



Use case: Spokes can utilize the common infrastructure in the hub, at the same time, and **talk to each other directly without a hop in the hub**.

# Security configuration features

## Secure at scale with admin rules and NSG management

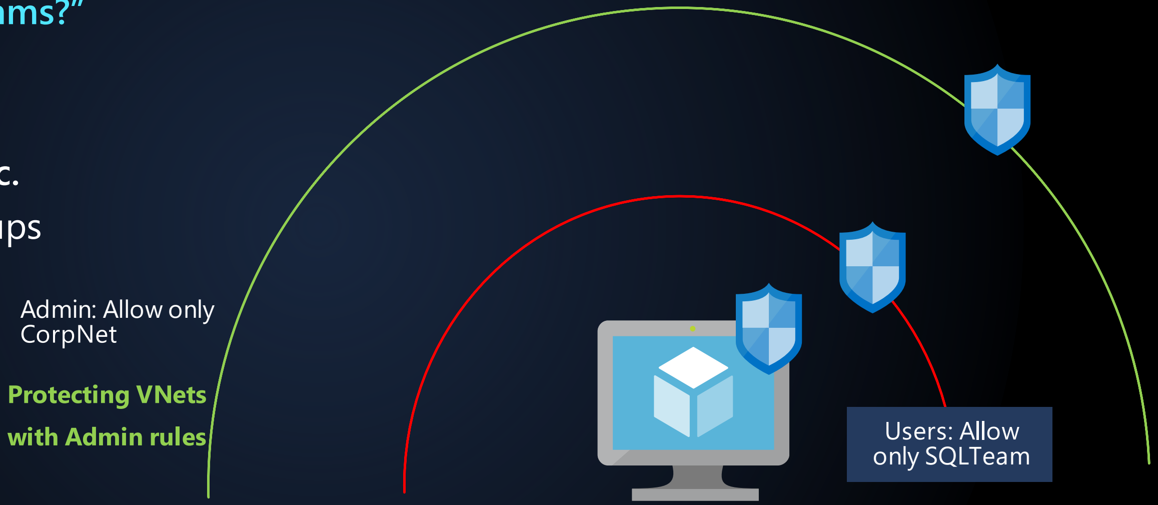
Problem statement: "As an admin, how can I enforce some security rules while the rest of the application specific rules are maintained by app teams?"

### Admin rule (this is not NSG)

- **Target audience: network admins, central governance teams, etc.**
- Admin level rules applied to all resources in desired network groups
  - Overwrite all conflicting rules
- Input: security policy -> output: admin rule
- New VMs will get these rules after they are created
- Enforced rules

### User rules created and managed by ANM:

- **NSG management capability**
- **Target audience: product/service teams**
- Input: security policy -> output: NSGs, ASGs
- Micro segmentation (Mail, DNS, ...)
- Conflict-free rules with modularity
  - Teams can edit and work together

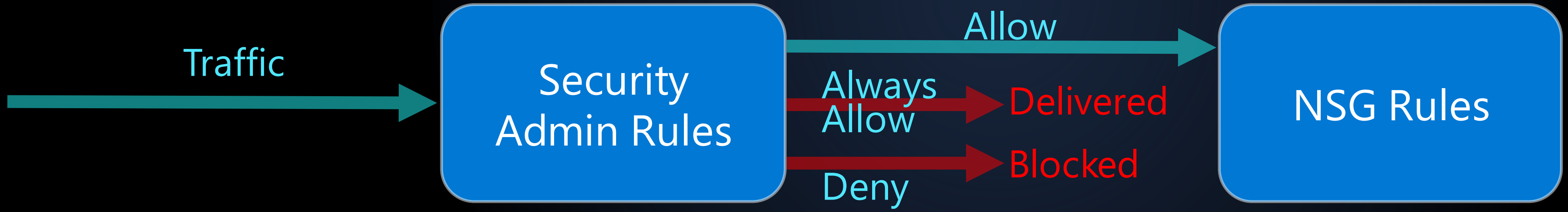

Admin: Allow only CorpNet
Protecting VNets with Admin rules
Users: Allow only SQLTeam

# Security admin rules vs NSGs

## How security admin rules work with NSGs

## The order of network traffic evaluation:

Security admin rules are evaluated **prior** to NSG rules

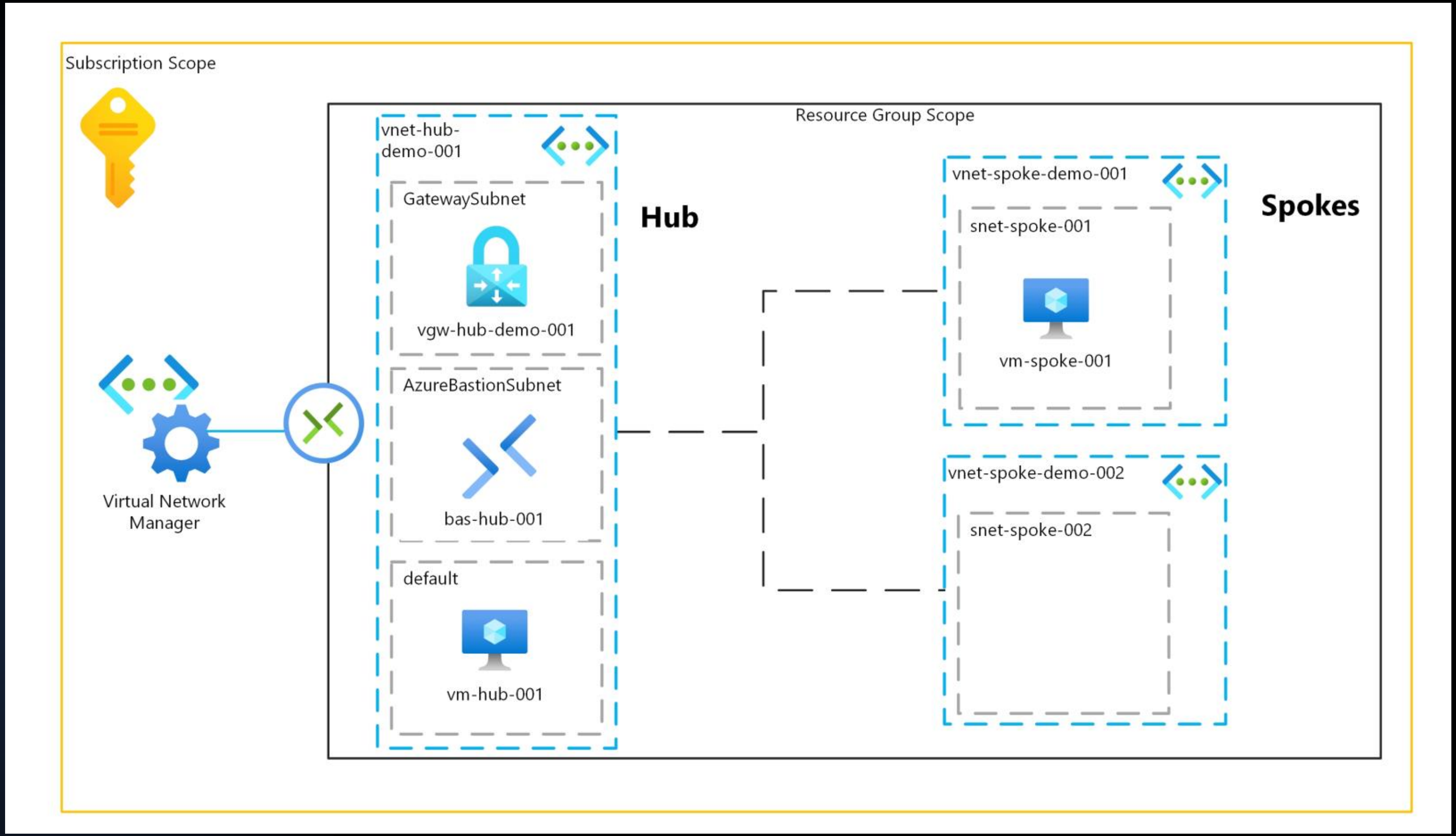


Three types of rules:

- Allow: Non-terminating
- Always Allow: Terminating
- Deny: Terminating

Microsoft Azure

# Demo



https://github.com/timoknapp/az-afm-avnm-demo

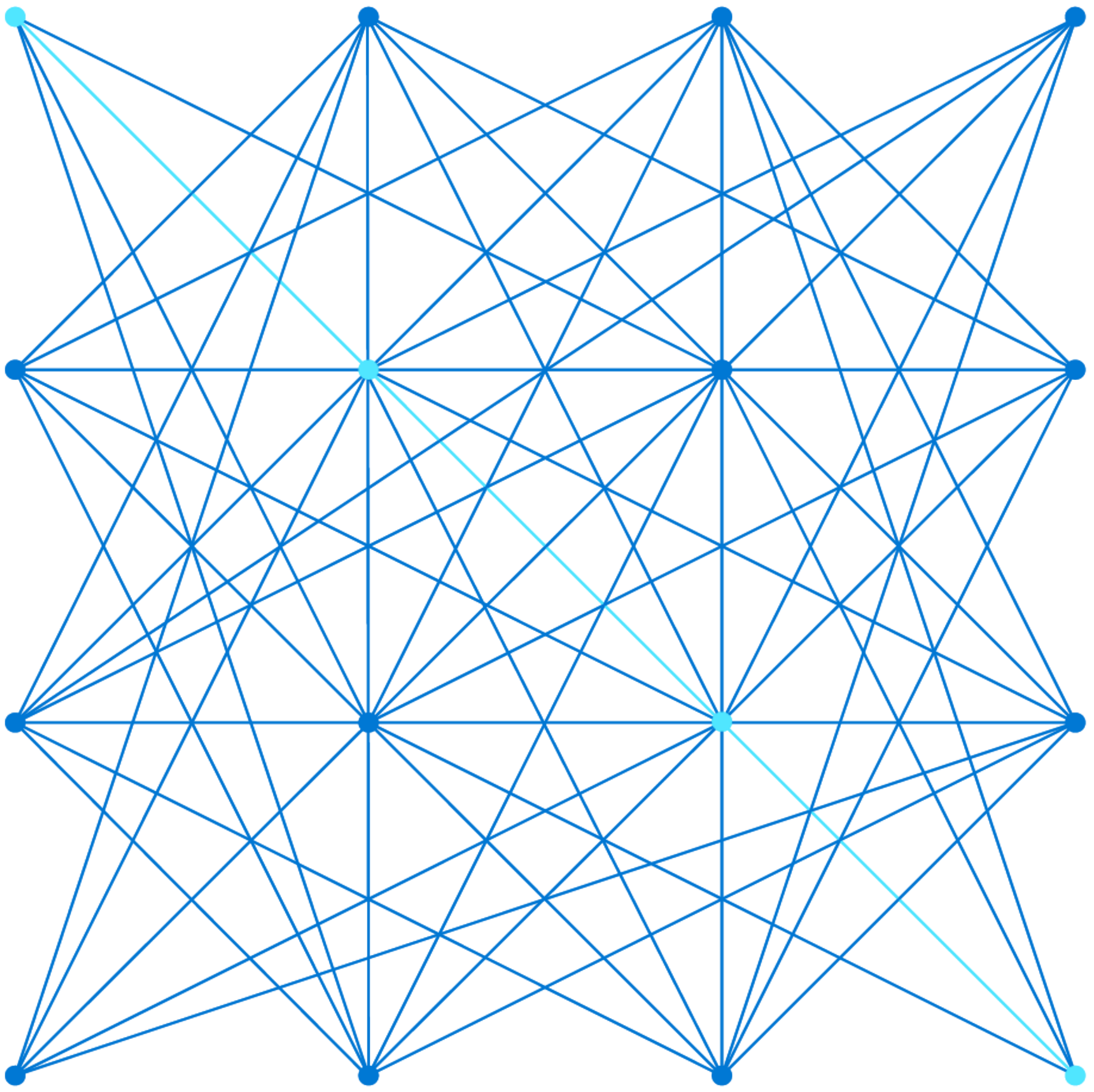
Microsoft Azure

# Feature Summary AVNM

- Network segmentation
- Connectivity configuration
- Security configuration
- More to come

Microsoft Azure

# Azure cross-region Load Balancer



# Agenda

- Introduction to Global Load balancing
- Azure cross-region Load Balancer Overview
- Azure cross-region Load Balancer Scenarios
- Demo
- Summary

# Customer global load balancing needs

## Reliability

Ensure high availability

Resilient to regional data center failures

## Scalability

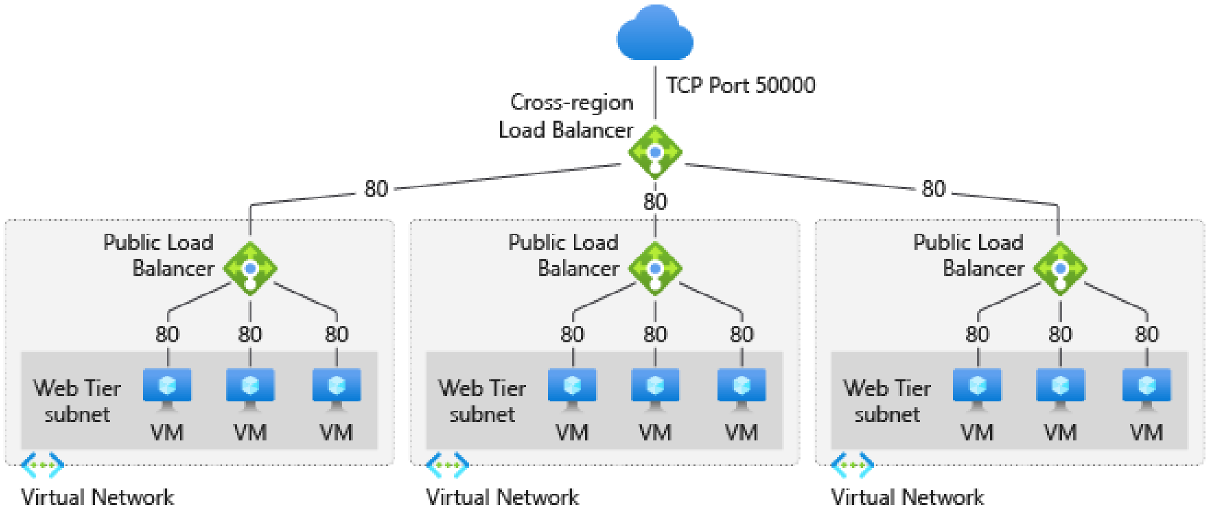
Ability to scale backend resources without interruption to customers

## Performance

Global traffic is served with low latency and traffic is sent to resources closest to users.

# Why use Azure cross-region Load Balancer?

- Global layer 4 (TCP/UDP) traffic load balancing
- Pass-through/transparent load balancer
- Static global anycast IP address
- Ultra-low latency with geo-proximity routing
- Seamlessly scale backend load balancers
- Automatic health probes
- Seamless failover for a global customer base behind a single endpoint



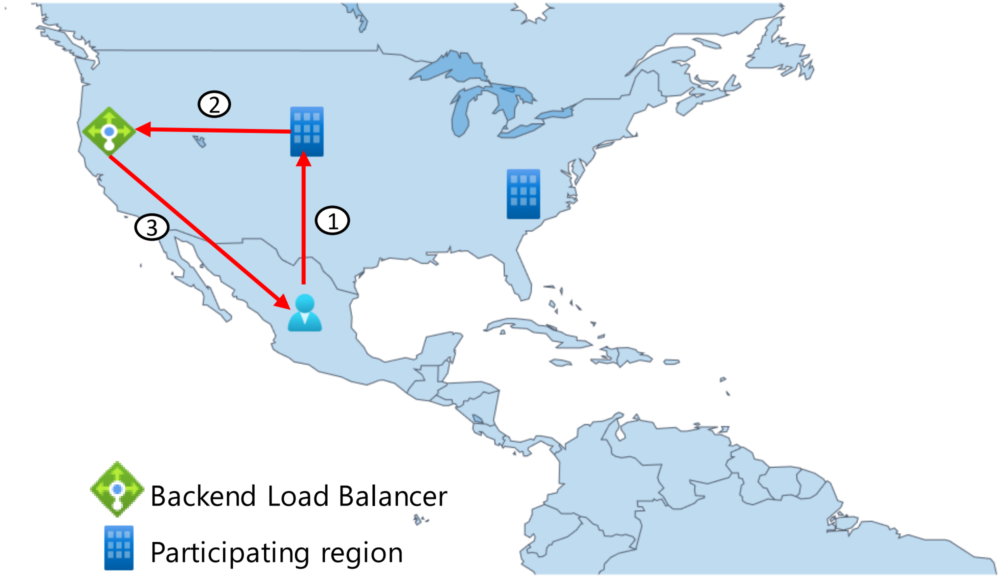
# Azure cross-region Load Balancer Components

## Home Regions

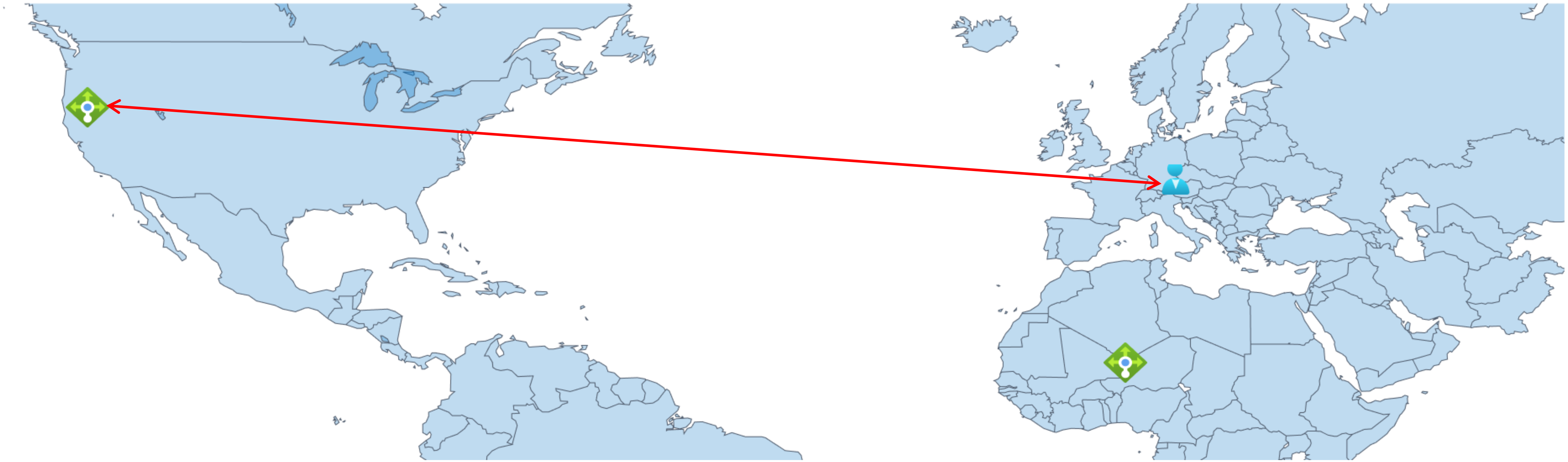
- The Azure region, where your CRLB will be deployed
- Traffic will **not** always pass through your home region
- Control plane operations reside in these regions

## Participating regions

- 10+ Azure regions that advertise your global IP address
- Traffic will be routed to closest participating to a user before being forwarded to the backend regional LB
- Backend regional load balancers aren't limited to the participating regions



# Scenario 1: Low-latency Load Balancing



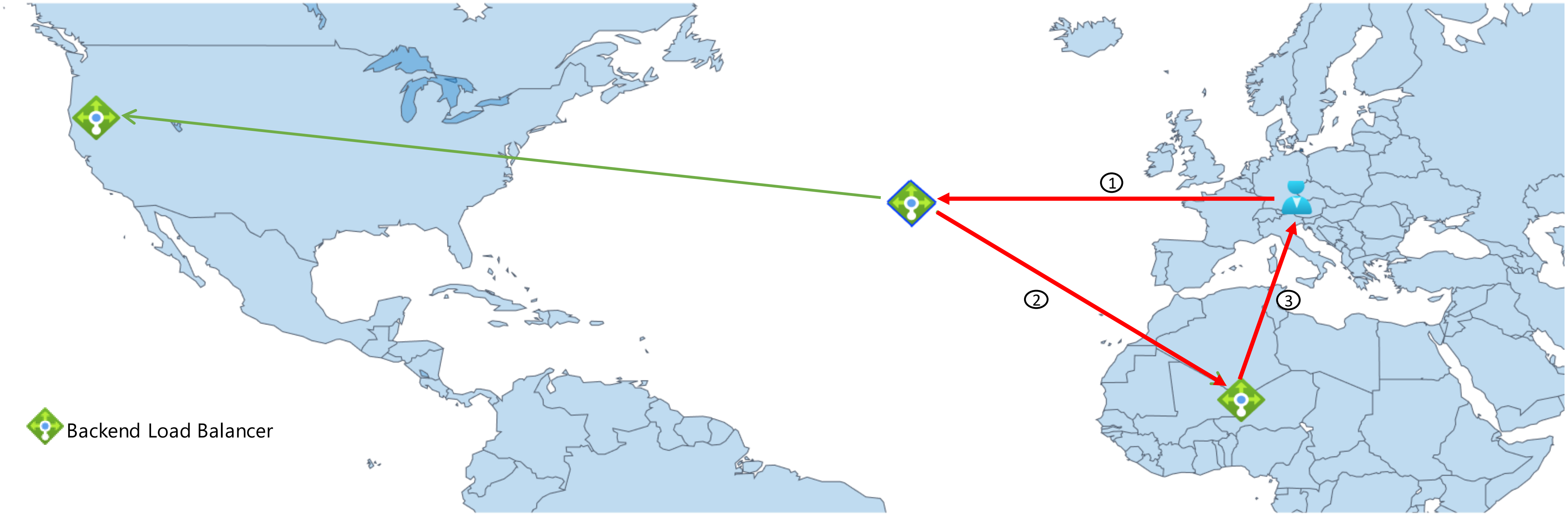
## Who is the customer

- Small IoT customer with a limited number of Azure deployments
- All applications require low latency

## Challenges

- Making sure global traffic is distributed with ultra low-latency
- Avoiding long traffic routes that will cause high latency for end users
- Ensuring users are routed to the Azure deployment closest to them

# Scenario 1: Low-latency Load Balancing



## Benefits of Azure cross-region Load Balancer

- Geo-proximity routing will ensure traffic is being sent to the Azure deployment closest to the user
  - Drastically **improves** the latency for users and applications

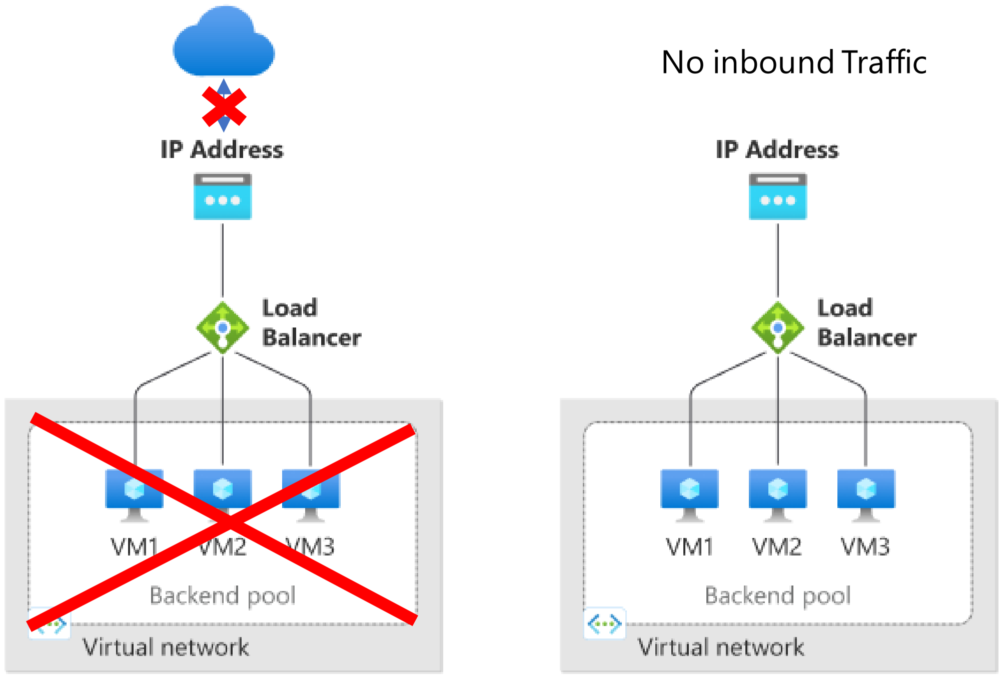
# Scenario 2: High Availability/Disaster Recovery

## Who is the customer

- Wholesale distributor with a global presence
- Backend application is replicated in multiple regions for DR purposes

## Challenges

- All incoming traffic needs to be routed to the next available region in case primary region is unhealthy
- Reducing downtime to users is critical during an outage/issue
  - Solutions like DNS-based may store the impacted IP address in its cache, which will cause traffic to still hit the impacted region.



Active

Passive

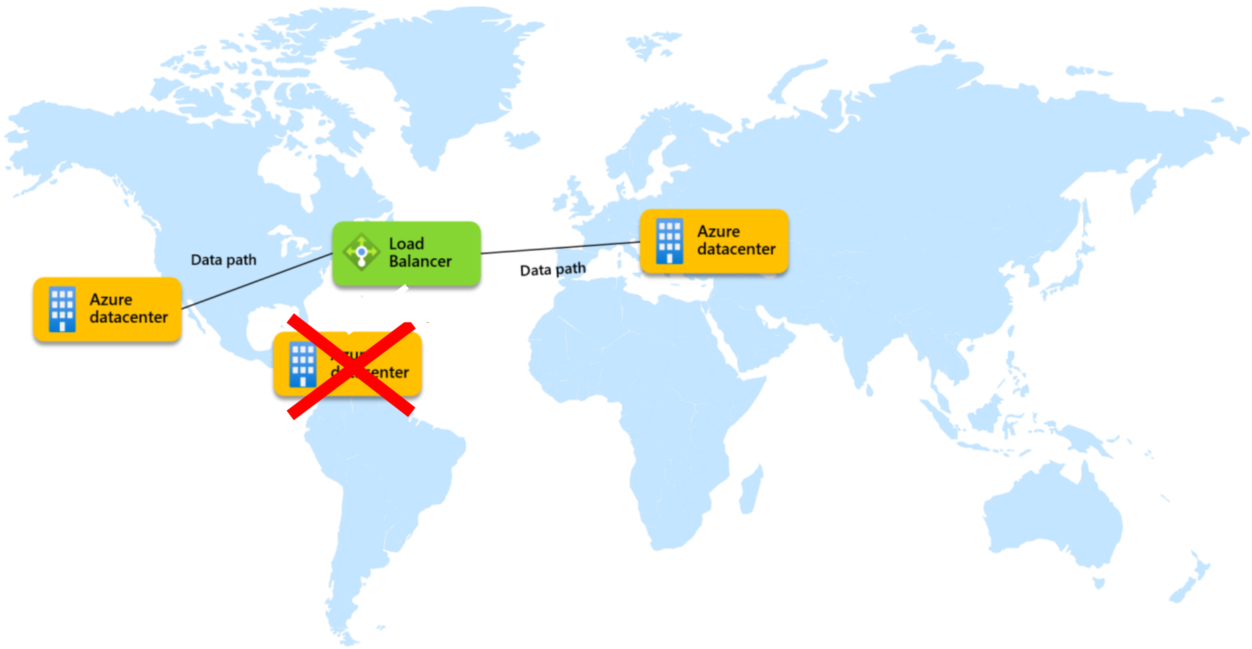
# Scenario 2: High Availability/Disaster Recovery

## Challenges

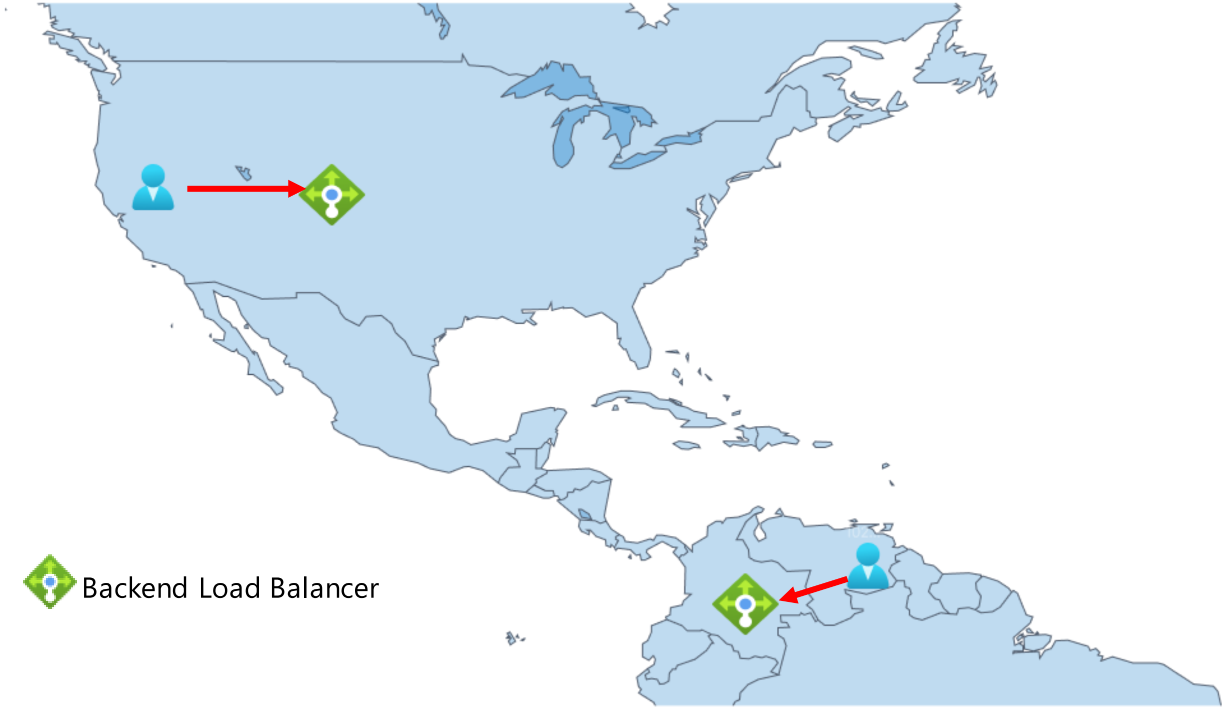
- All incoming traffic needs to be routed to the next available region in case primary region is unhealthy
- Reducing downtime to users is critical during an outage/issue

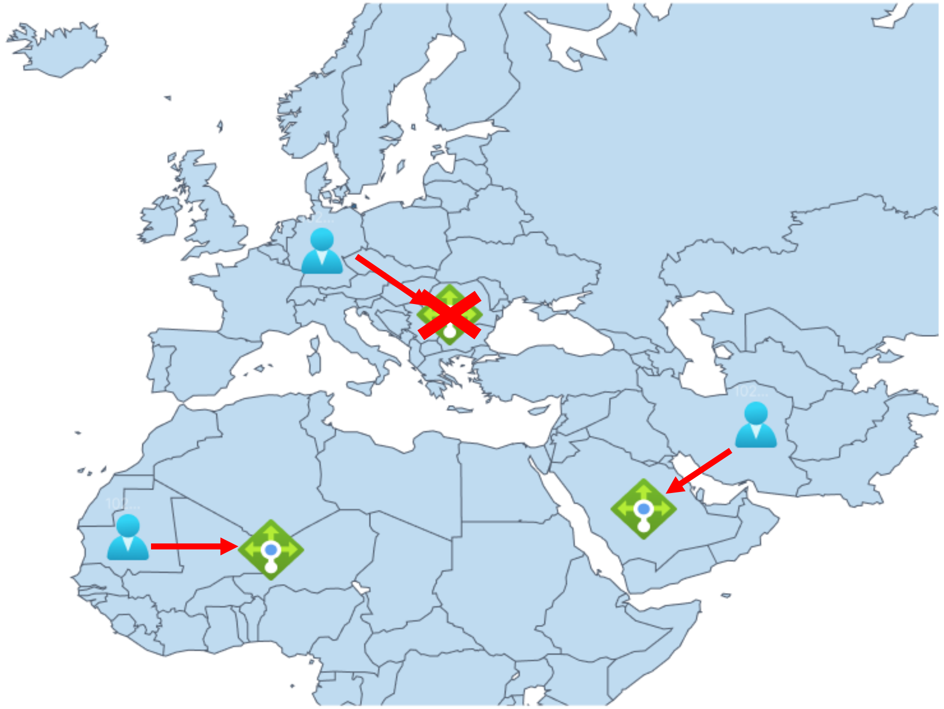
## Benefits of Azure cross-region Load Balancer

- Seamless failover behind a single endpoint
  - Health probes automatically detect an impacted region
  - New connections are sent to the next healthy deployment
- Impacted resources automatically are added back into the pool once they are healthy



# Scenario 3: Static IP Address





## Who is the customer

- Automotive company with a large global customer base
- Azure deployments around the globe to ensure low latency for their customer
- Has plans to add additional deployments as their business scales

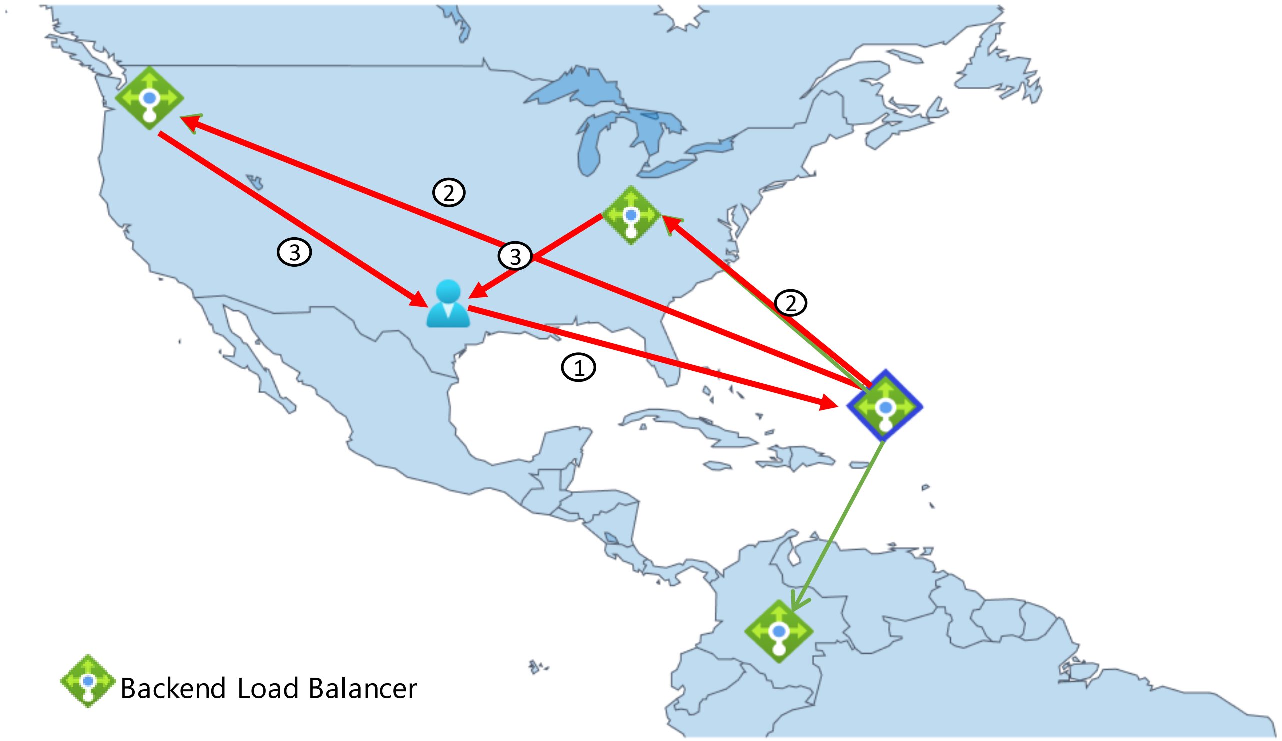
## Challenges

- As user demand grows, customers need to scale up their applications to meet demand
- Additional deployments create additional IP management/overhead for the customer
  - Making sure users are given the correct IP address.
- Ensuring no impact to users as deployments scale up or down

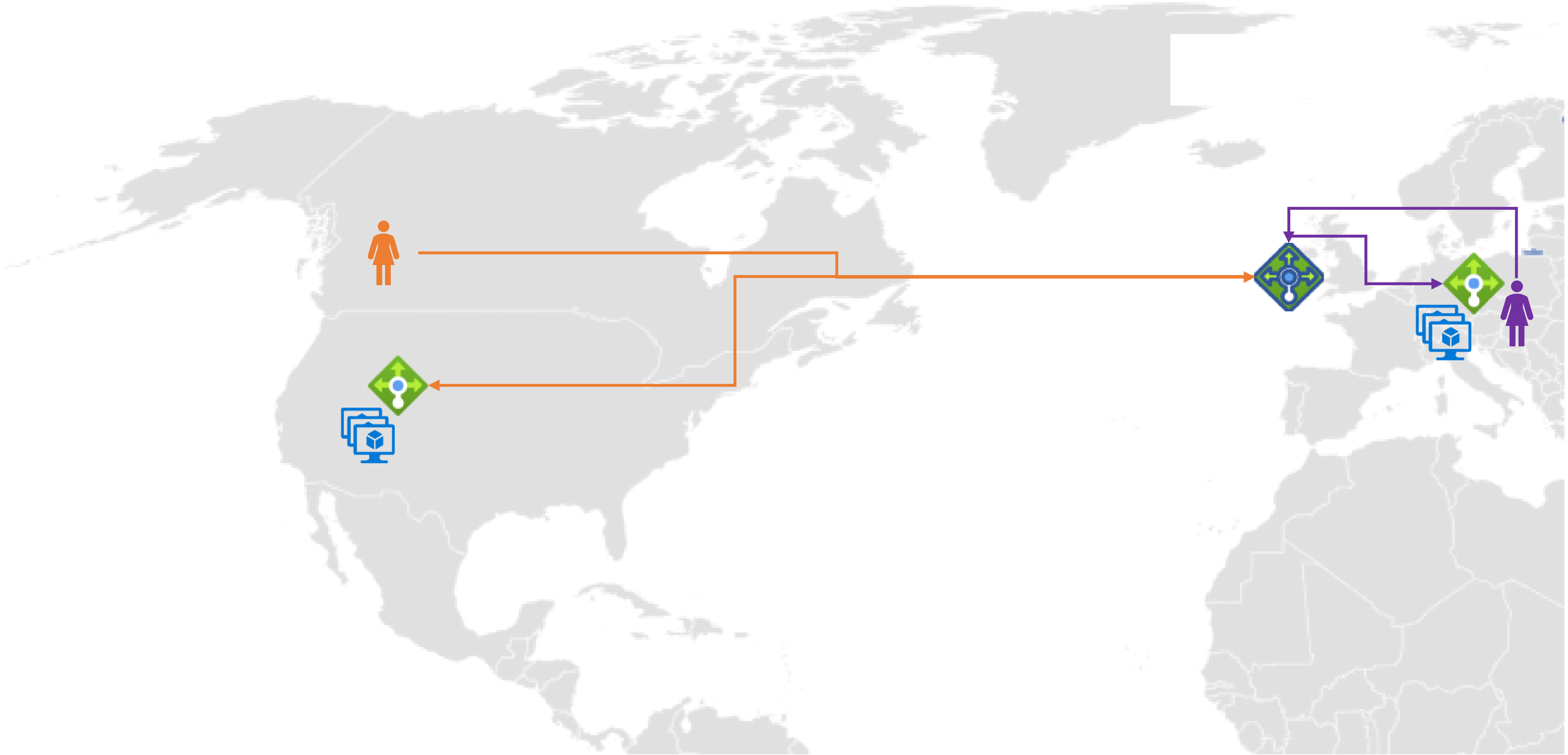
# Scenario 3: Static IP Address

## How Azure cross-region Load Balancer helps

- Scale up/down backend load balancers, all behind a single global IP address
- Add/remove backend regional load balancers without any interruption



# Demo



# Want to learn more about Azure cross-region LB?

**Public Docs**

- Azure cross-region Load Balancer overview
- Tutorial: Build a globally resilient architecture with Azure cross-region Load Balancer

**GA Announcement**

- GA Blog
- Azure Update

**Blogs**

- Choose the best global distribution solution for your applications with Azure
- Build a globally resilient architecture with Azure Load Balancer

Microsoft Azure

# Secrets in Azure Container Apps

# Azure Container Apps

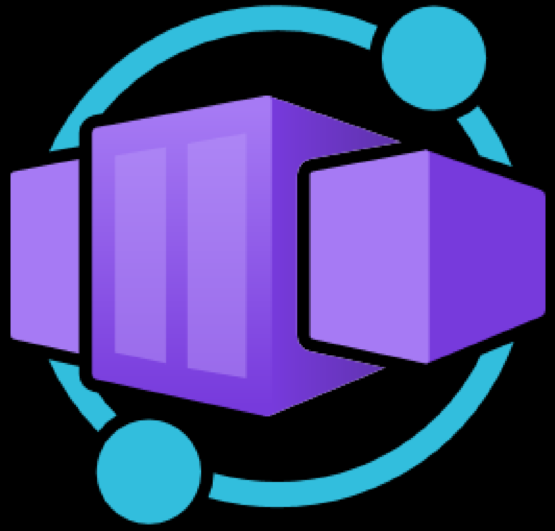
## Serverless containers for microservices

Build modern apps on open source

Focus on apps, not infrastructure

Scale dynamically based on events

Kubernetes KEDA dapr DAPR envoy Envoy



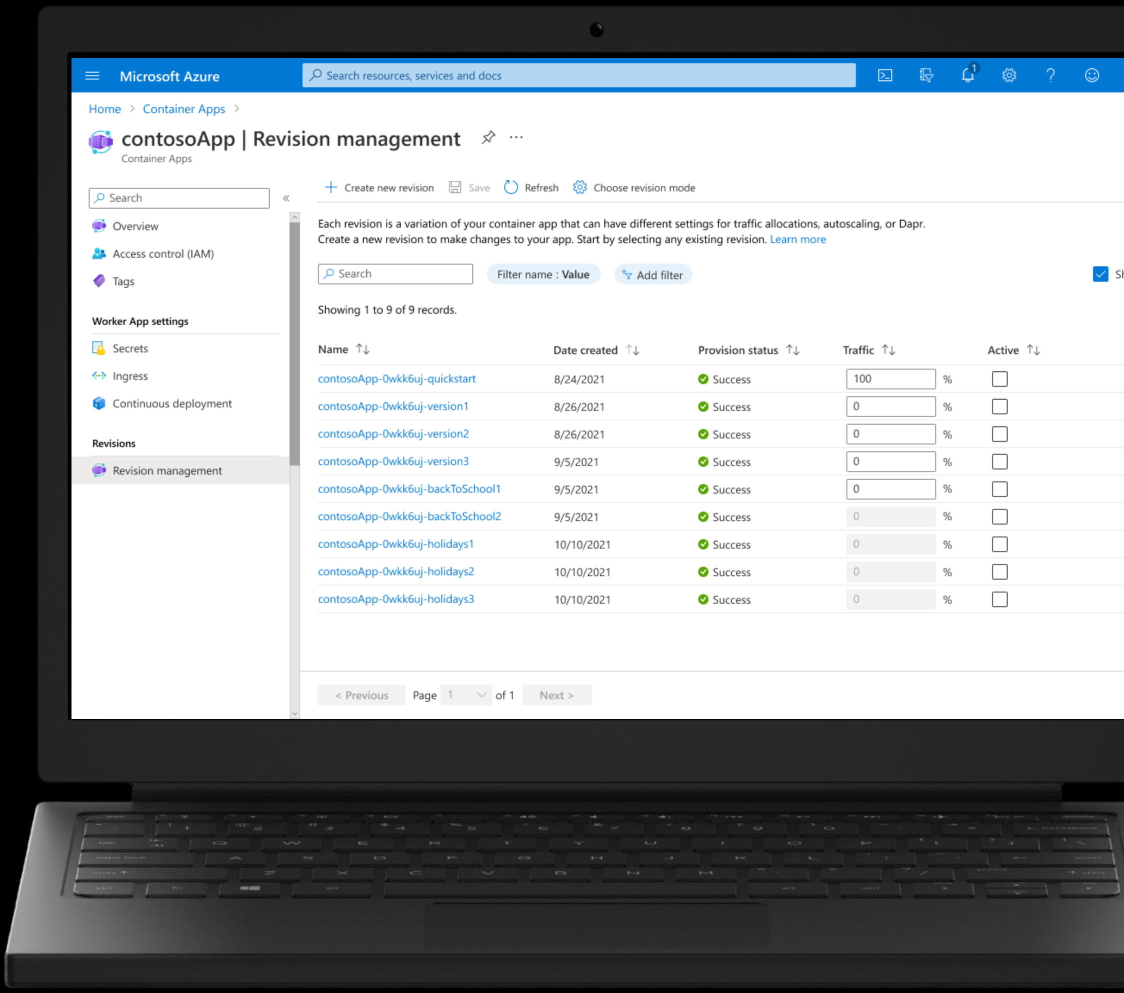
Build modern apps on open-source

Focus on apps, not infrastructure

Scale dynamically based on events

# Build modern apps on open-source

- → App portability powered by open standards and APIs
- → App patterns and best practices encapsulated by products like Dapr
- → Service capabilities influenced by OSS contributions
- → Benefit from streamlined application lifecycle for upgrades and versioning, traffic shifting, service discovery, and monitoring.



Each revision is a variation of your container app that can have different settings for traffic allocations, autoscaling, or Dapr. Create a new revision to make changes to your app. Start by selecting any existing revision. Learn more

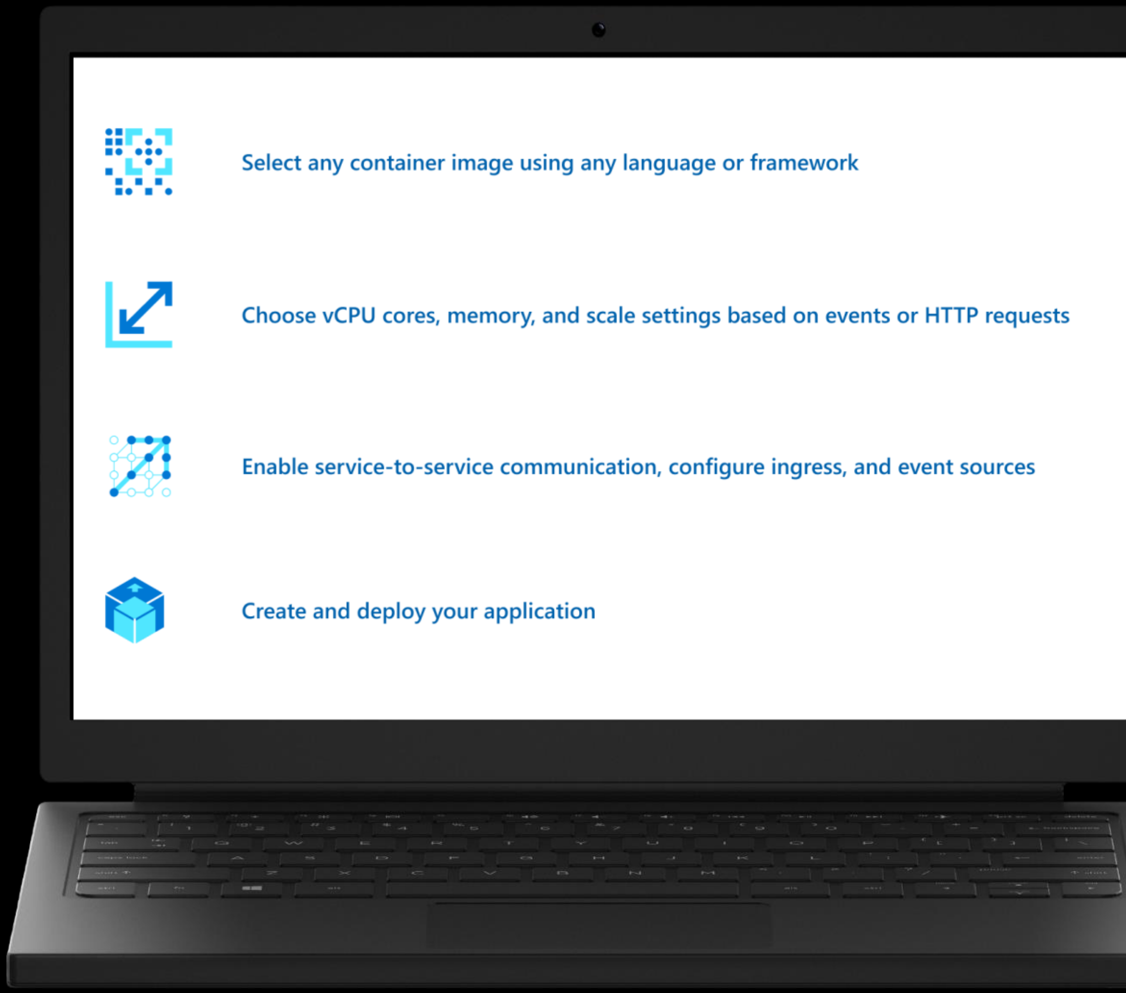
Showing 1 to 9 of 9 records.

| Name ↑↓ | Date created ↑↓ | Provision status ↑↓ | Traffic ↑↓ | | Active ↑↓ |
|---|---|---|---|---|---|
| contosoApp-0wkk6uj-quickstart | 8/24/2021 | ✓ Success | 100 | % | □ |
| contosoApp-0wkk6uj-version1 | 8/26/2021 | ✓ Success | 0 | % | □ |
| contosoApp-0wkk6uj-version2 | 8/26/2021 | ✓ Success | 0 | % | □ |
| contosoApp-0wkk6uj-version3 | 9/5/2021 | ✓ Success | 0 | % | □ |
| contosoApp-0wkk6uj-backToSchool1 | 9/5/2021 | ✓ Success | 0 | % | □ |
| contosoApp-0wkk6uj-backToSchool2 | 9/5/2021 | ✓ Success | 0 | % | □ |
| contosoApp-0wkk6uj-holidays1 | 10/10/2021 | ✓ Success | 0 | % | □ |
| contosoApp-0wkk6uj-holidays2 | 10/10/2021 | ✓ Success | 0 | % | □ |
| contosoApp-0wkk6uj-holidays3 | 10/10/2021 | ✓ Success | 0 | % | □ |

< Previous Page 1 of 1 Next >

# Focus on apps, not infrastructure

- Apps with any development stack, any Linux container image
- No opinionated programming model
- High productivity development experience
- Set up a code-to-cloud pipeline using GitHub Actions.



Select any container image using any language or framework

Choose vCPU cores, memory, and scale settings based on events or HTTP requests

Enable service-to-service communication, configure ingress, and event sources

Create and deploy your application

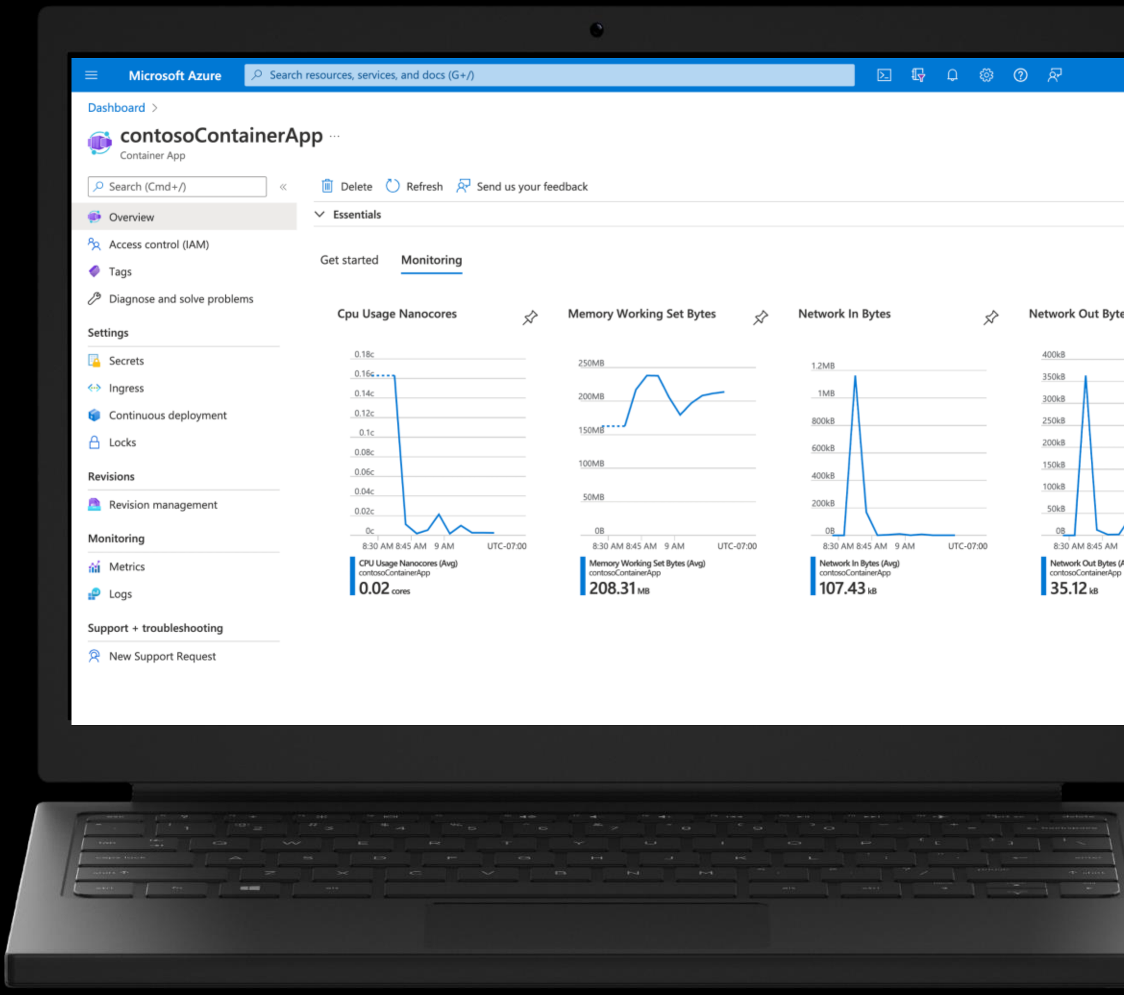
Build modern apps on open source

Focus on apps, not infrastructure

Scale dynamically based on events

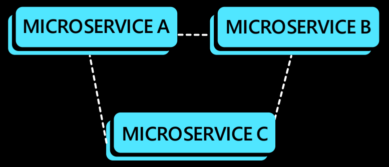
# Scale dynamically based on events

- → Serverless autoscale based on HTTP requests, KEDA event scale triggers, or CPU and Memory
- → Declarative scaling rules eliminate the need to manage complex infrastructure
- → Scale to 0 and pay per use by second



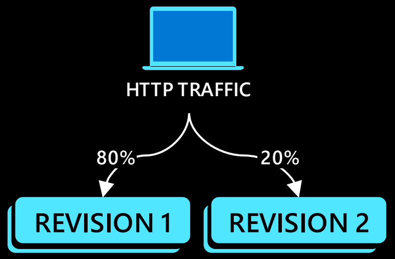
# What can you build with Azure Container Apps?

## Microservices



Microservices architecture with the option to integrate with Dapr

## Public API endpoints



E.g., API app with HTTP requests split between two revisions of the app

## Web Apps

E.g., Web app with custom domain, TLS certificates, and integrated authentication

## Event-driven processing

E.g., Queue reader app that processes messages as they arrive in a queue

## Background processing

E.g., Continuously running background process transforms data in a database

## AUTO-SCALE CRITERIA

Individual microservices can scale independently using any KEDA scale triggers

Scaling is determined by the number of concurrent HTTP requests

Scaling is determined by the number of concurrent HTTP requests

Scaling is determined by the number of messages in the queue

Scaling is determined by the level of CPU or memory load

# How does ACA compare to AKS?

| | **Azure Kubernetes Service (AKS)**<br>Infrastructure focus, higher flexibility | **Azure Container Apps (ACA)**<br>Application focus, infrastructure abstraction |
|---|---|---|
| Core value proposition | Managed Kubernetes cluster in Azure with full access to the Kubernetes API server and high level of control over cluster configuration with a node-based pricing model | Fully-managed serverless abstraction on top of Kubernetes infrastructure, purpose built for managing and scaling event-driven microservices with a consumption-based pricing model |
| Optimized for | • Upstream feature parity with a managed control plane<br>• Operations flexibility with advanced customization<br>• Experienced Kubernetes operators | • Platform-as-a-Service experience with serverless scale<br>• Developer productivity with low operations overhead<br>• Linux-based, general-purpose stateless containers |
| Interaction model | • Operators deploy node-based AKS clusters using Azure Portal, CLI or Infrastructure-as-Code templates (IaC)<br>• Developers deploy containers via Kubernetes deployment manifests or HELM charts to logically-isolated namespaces within the cluster | • Developers deploy containers as individual Container Apps using Azure Portal, CLI or IaC templates without any Kubernetes manifests required<br>• Related container apps are deployed to a shared Container Apps environment comparable to a Kubernetes namespace |
| OSS Integration | • Provides a set of cluster extensions and add-ons for operators to enable OSS components in-cluster including Dapr, KEDA, Open Service Mesh, GitOps (Flux), Pod Identity, etc.<br>• Supports manual installation via Kubernetes manifests | Includes opinionated platform capabilities powered by CNCF projects including Dapr, KEDA and Envoy which are fully platform-managed and supported<br>• Envoy: managed ingress and traffic splitting<br>• KEDA: managed, event-driven autoscale<br>• Dapr: codified best practices for microservices |

# Secrets in Azure Container Apps

## Environment Variables

- Not built for storing sensitive date
- Lives in the scope of a container
- Can't share between multiple containers

## Secrets

- Built for storing sensitive information
- Lives in the application scope
- Can be shared between multiple containers

## Azure Key Vault

- Specialized service for storing secrets in keys
- Separate service
- Good if you have to share the keys between multiple apps

# Demo

Microsoft Azure

# Microsoft Dev Box -

Secure, cloud workstations built for developer productivity

# Supporting developers is tantamount to business success

70%

of top economic performers are using their software to differentiate themselves,[1] yet **a growing tech talent gap** means organizations must invest to **keep devs happy and productive**

## Empower devs to work where they feel productive

62% of developer prefer to work remotely or in hybrid settings, and over 75% only want to be in office 2-3 days per week[2]

## Maximize productivity with the power of the cloud

Standardizing workloads around cloud-based developer tools and services can help increase developer productivity by as much as 30%[3]

## Keep devs and source code secure in a hybrid world

Organizations that integrate security workflows earlier in development are 1.6x more likely to meet or exceed their goals[4]

1. McKinsey, 2022, 2. Zenhub, 2022, 3. Forrester, 2021, 4. Google, 2021

# Traditional VDI solutions enable more flexible workflows but fail to meet key developer needs

### Virtual desktops

Existing Virtual Desktop Infrastructure (VDI) and Desktop as a Service (DaaS) offerings enable organizations to outfit remote workers with virtual desktops

## Limitations of VDI and DaaS

### Limited productivity gains

Vanilla Virtual Machines (VMs) suffer from many of the same problems devs already face with physical workstations

### Lack of dev integrations

Traditional VDI lacks specialized dev tool and dev services integrations that are crucial for maximizing dev productivity

### Increased security concerns

It's difficult to maximize security by project due to limited, generic security policies enforced for each VM

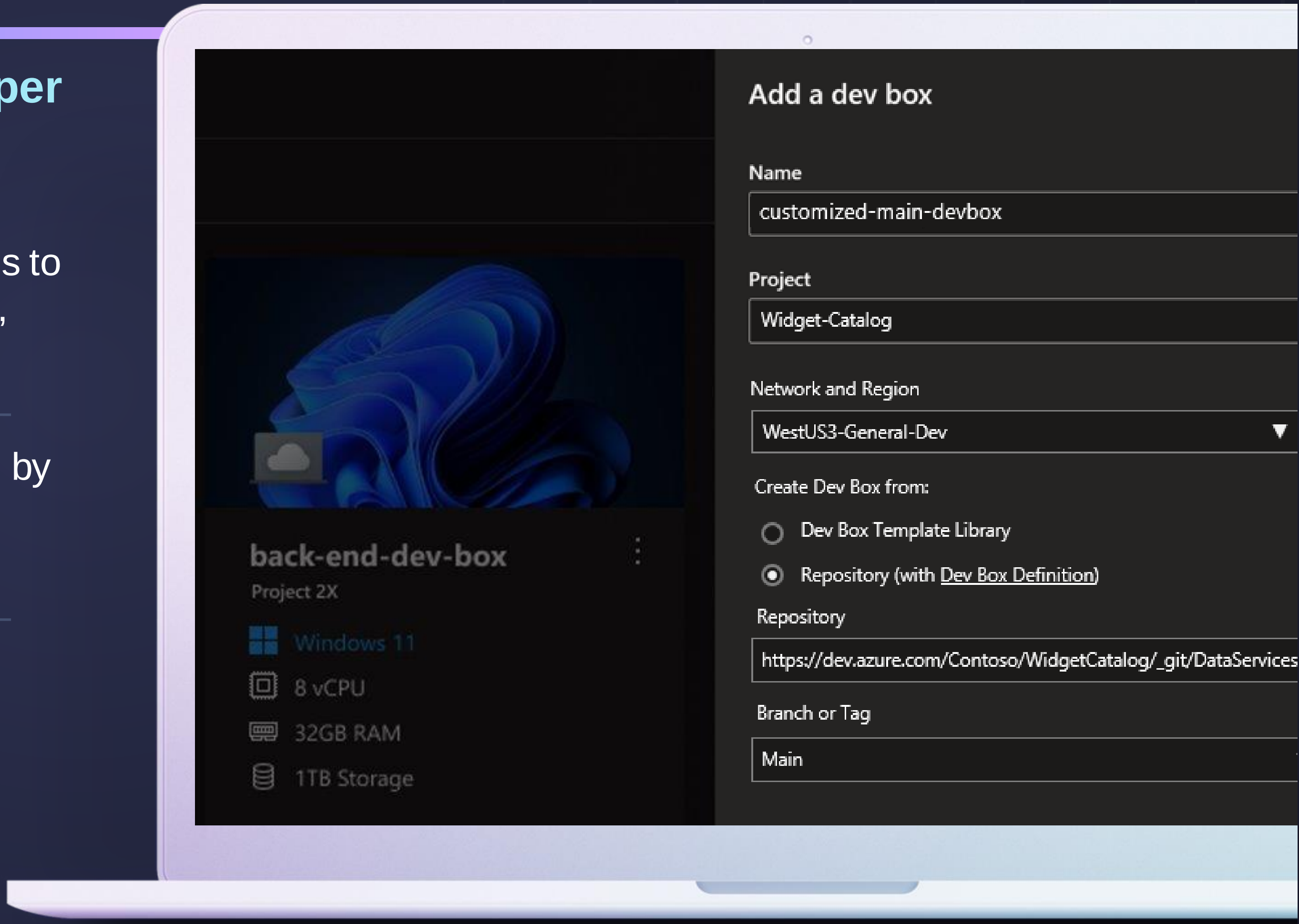
# What is Microsoft Dev Box?

## Secure cloud workstations built for developer productivity

**Ready-to-code:** Self-service, on-demand access to task-specific workstations with scalable compute, available instantly.

**Project-based:** Preconfigured workstations built by dev teams with the right tools and resources for their projects

**Managed and secure:** Centralized governance based on organizational standards for security, compliance, and cost controls.



back-end-dev-box

Project 2X

Windows 11

8 vCPU

32GB RAM

1TB Storage

Add a dev box

Name: customized-main-devbox

Project: Widget-Catalog

Network and Region: WestUS3-General-Dev ▼

Create Dev Box from:

○ Dev Box Template Library

● Repository (with Dev Box Definition)

Repository: https://dev.azure.com/Contoso/WidgetCatalog/_git/DataServices

Branch or Tag: Main

# Transform key dev scenarios with Microsoft Dev Box

### Developer onboarding

Get devs up and running with ready-to-code, preconfigured dev boxes

### Complex configurations

Empower devs to deploy multiple workstations tailored to different tasks

### Legacy applications

Quickly spin up dev boxes built for troubleshooting legacy apps

### Remote dev experience

Ensure a low-latency, high-performance dev experience wherever they are in the world

### Security and compliance

Centrally manage dev devices and keep them secure across locations

### User permissions

Provide different permissions for contingent staff and fulltime devs

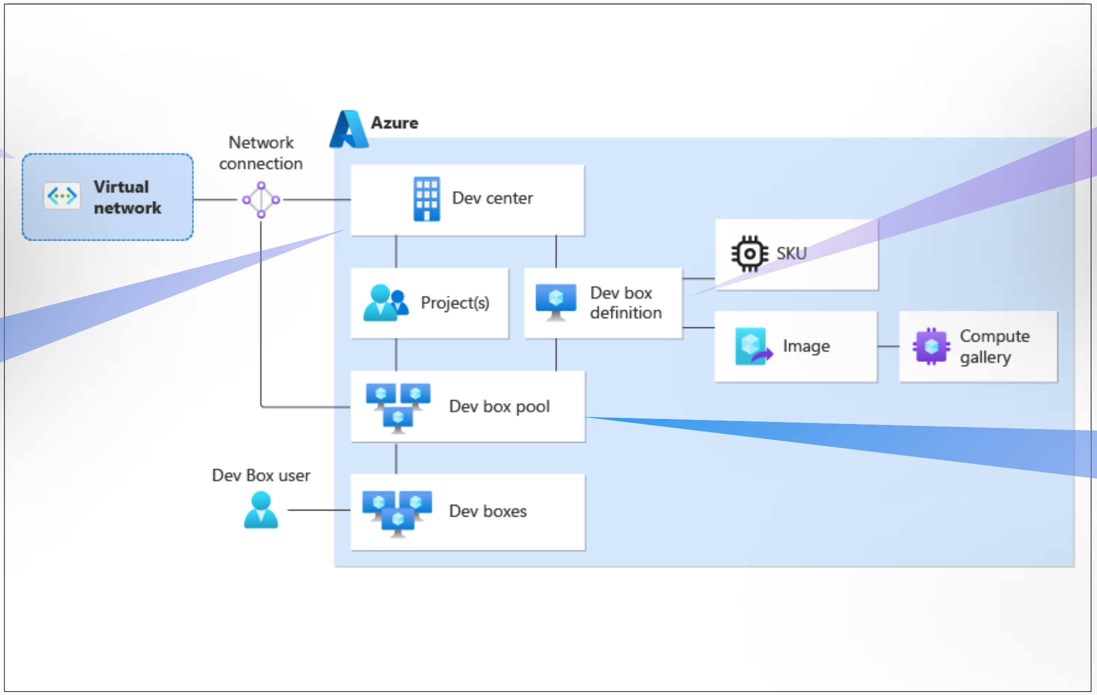
# High-level conceptual architecture

### Network connection

- Azure or Hybrid

### Dev Center

- Logical container to help organize dev box



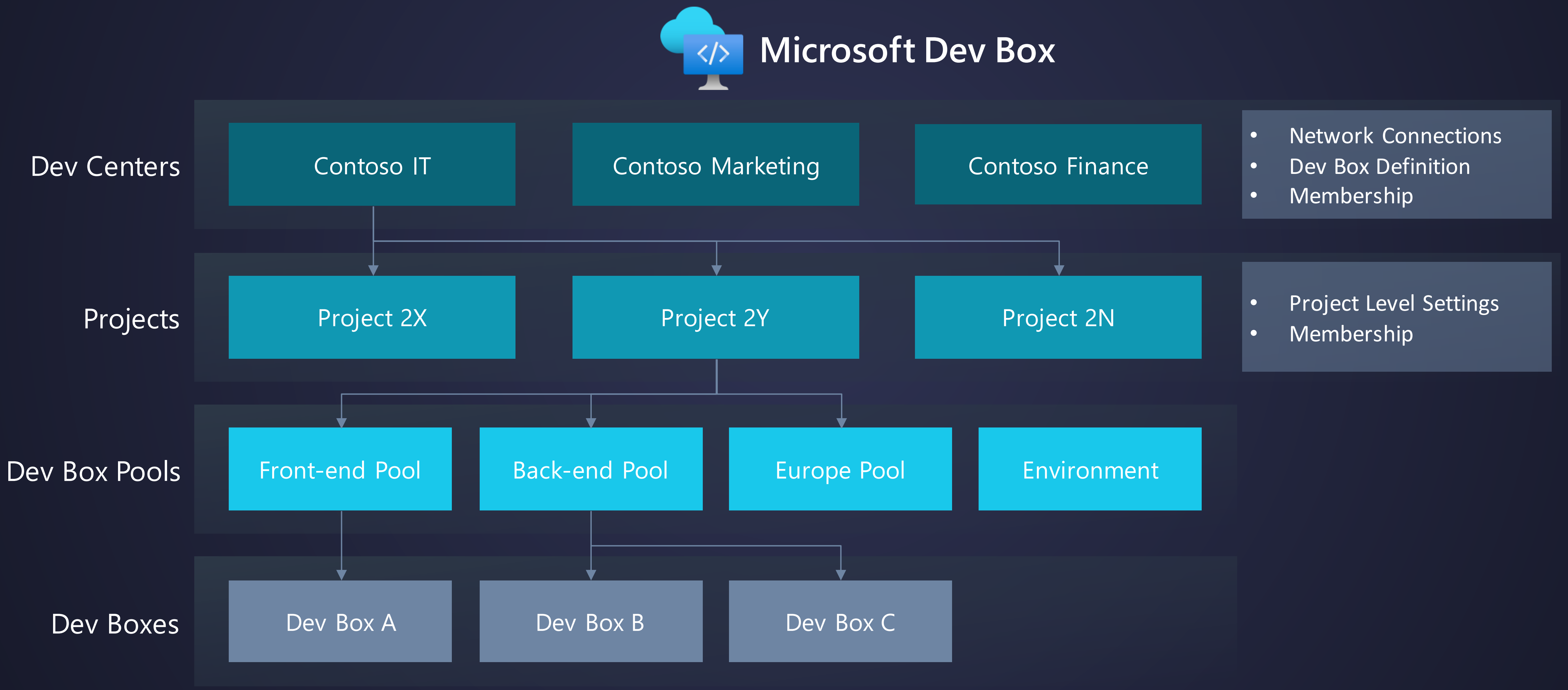
### Dev Box definitions

- Defines configuration of the Dev Boxes (Image)

### Dev Box pools

- Combines definitions and projects (groups)

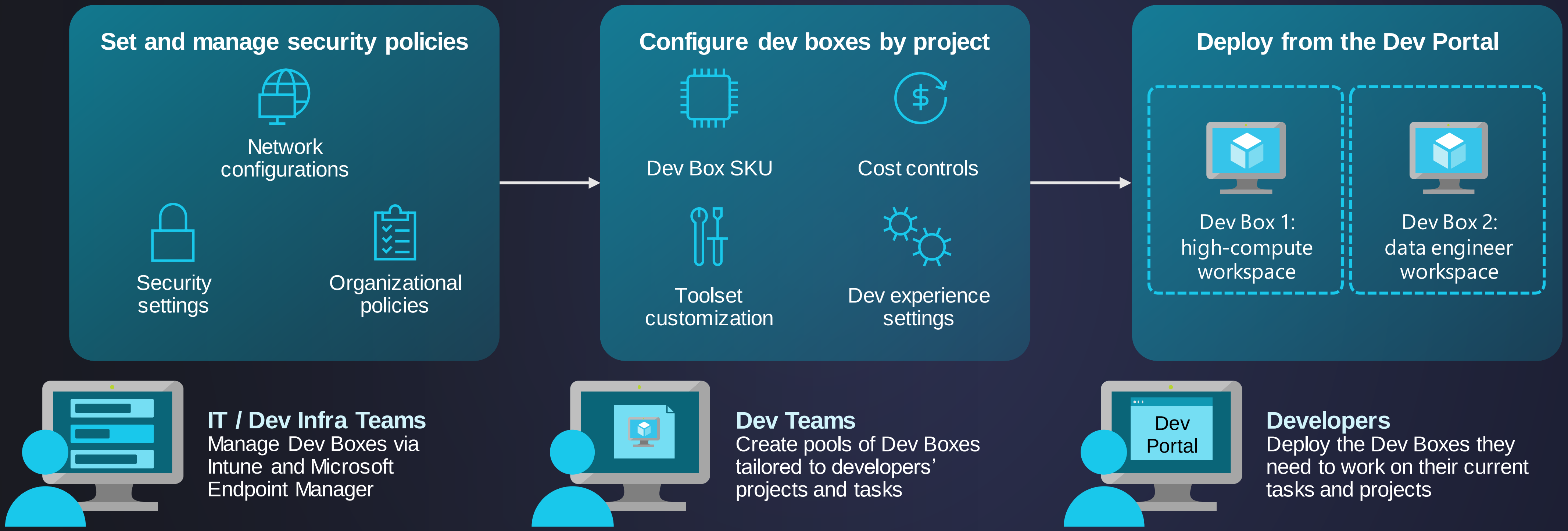
# High-level conceptual architecture



- Network Connections
- Dev Box Definition
- Membership

- Project Level Settings
- Membership

# How different roles use Microsoft Dev Box



**Set and manage security policies**

Network configurations

Security settings

Organizational policies

**Configure dev boxes by project**

Dev Box SKU

Cost controls

Toolset customization

Dev experience settings

**Deploy from the Dev Portal**

Dev Box 1: high-compute workspace

Dev Box 2: data engineer workspace

Dev Portal

**IT / Dev Infra Teams**
Manage Dev Boxes via Intune and Microsoft Endpoint Manager

**Dev Teams**
Create pools of Dev Boxes tailored to developers' projects and tasks

**Developers**
Deploy the Dev Boxes they need to work on their current tasks and projects

# GitHub Codespaces and Microsoft Dev Box

| | **Microsoft Dev Box**<br>Full dev workstations in the cloud optimized for enterprise-grade dev productivity and security | **GitHub Codespaces**<br>Cloud-based dev environments for fast, on-demand coding on any device |
|---|---|---|
| **Operating system** | Windows | Linux |
| **SCM Support** | Any version control system | Repos on GitHub |
| **Tool support** | Any Windows-based tool | Visual Studio Code |
| **Target workloads** | Any workload<br>Including: Desktop, IoT, mobile, games, & more<br>(Windows or cross-plat) | Cloud native apps<br>Including: web apps, APIs, backends |
| **IT management** | Microsoft Intune, Microsoft Azure | GitHub.com |

# Demo

[Quickstart: Configure Microsoft Dev Box - Microsoft Dev Box | Microsoft Learn]

https://github.com/timoknapp/az-dev-box

Microsoft Azure

# Thank you.