Microsoft

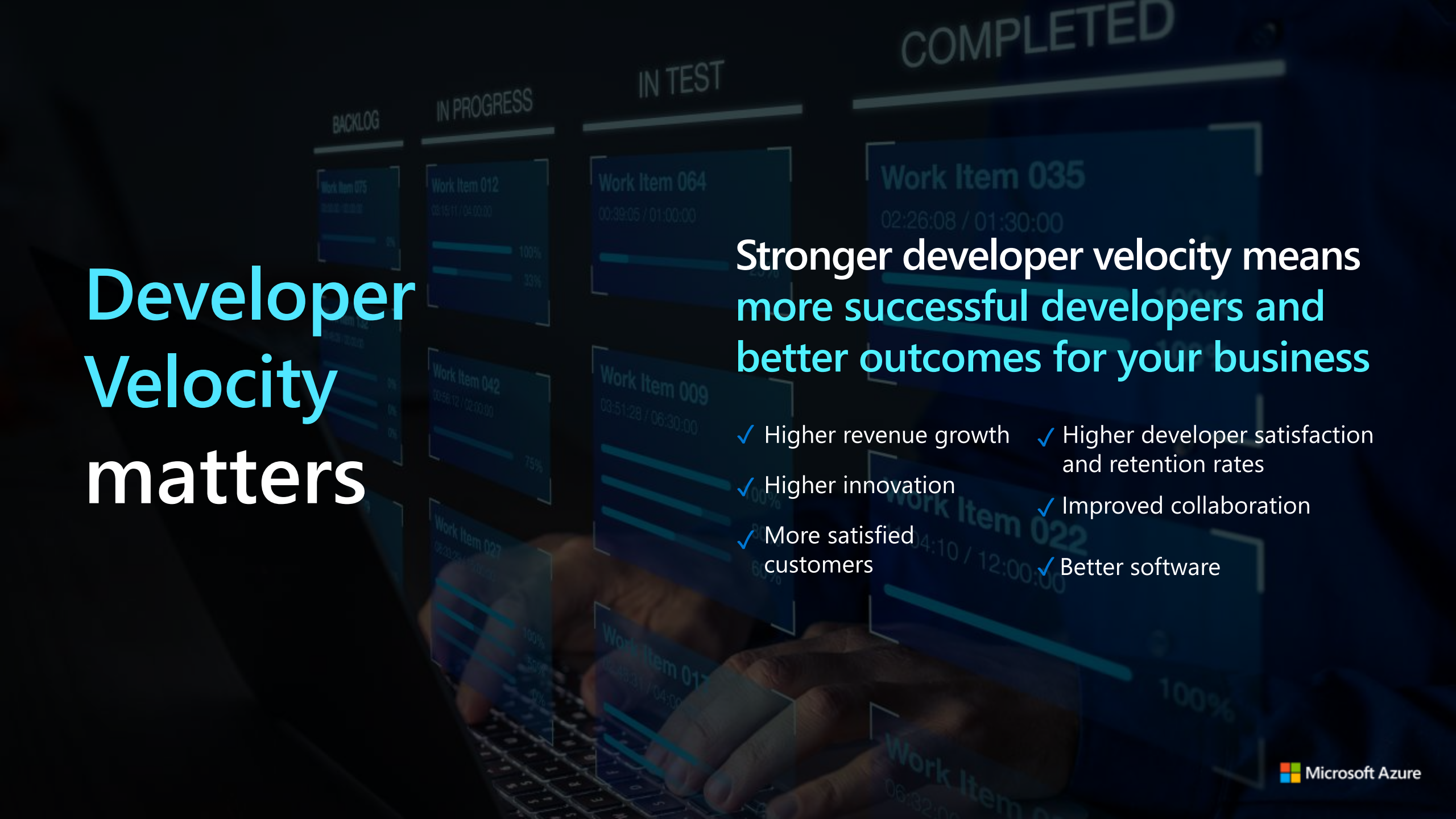
Azure Container Apps

The app of the future is cloud native

50%

of organizations will use applications built on managed services including cloud-native technologies by 2024

IDC FutureScape



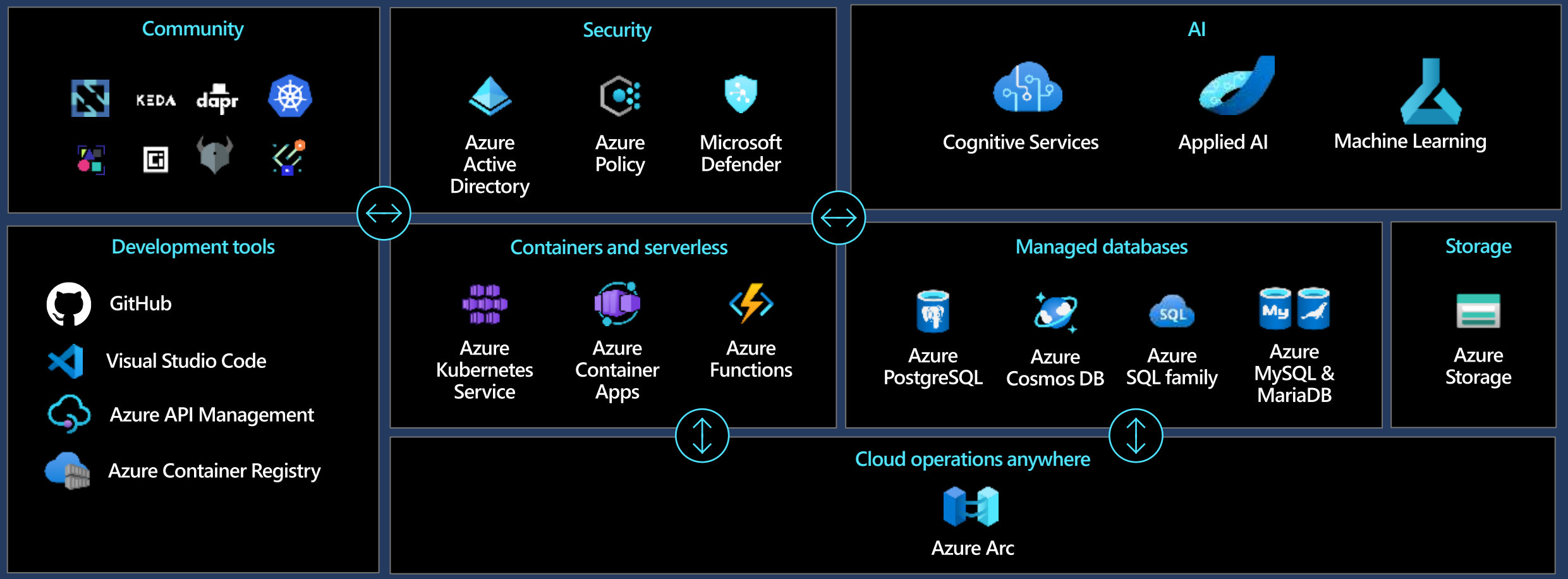
Developer Velocity matters

Stronger developer velocity means more successful developers and better outcomes for your business

- ✓ Higher revenue growth
- ✓ Higher innovation
- ✓ More satisfied customers
- ✓ Higher developer satisfaction and retention rates
- ✓ Improved collaboration
- ✓ Better software

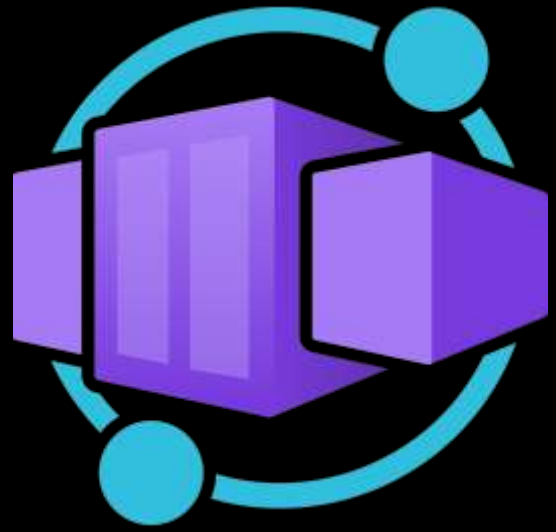
Microsoft Azure

Building cloud-native on Azure



Azure Container Apps

Generally Available



Serverless containers for microservices

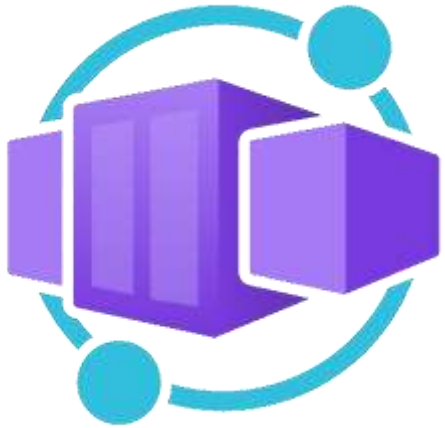
Build modern apps on open source

Focus on apps, not infrastructure

Scale dynamically based on events

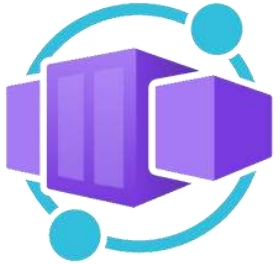
Azure Container Apps

A new serverless container platform for building modern apps and microservices



Built on a foundation of AKS, KEDA, Dapr, and Envoy

Azure Container Apps

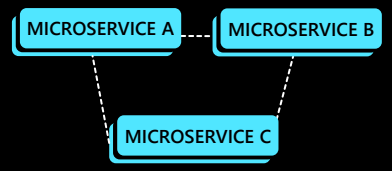


"Azure Container Apps enables executing application code packaged in any container and is unopinionated about runtime or programming model."

- Enjoy the **benefits of running containers** while leaving behind the concerns of **managing cloud infrastructure** and **complex container orchestrators**.
- **Serverless** (scale to zero support)
- **Scale** on HTTP requests, events, or run always-on background jobs
- **Automatic encryption** for ingress and service-to-service communications
- Built on a foundation of AKS, KEDA, Dapr, and Envoy

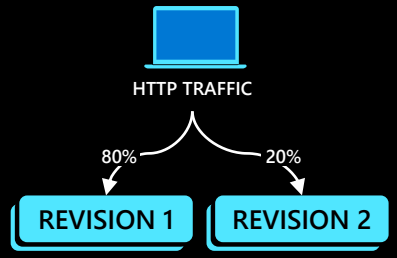
What can you build with Azure Container Apps?

Microservices



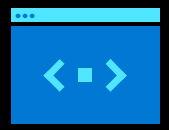
Microservices architecture with the option to integrate with Dapr

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

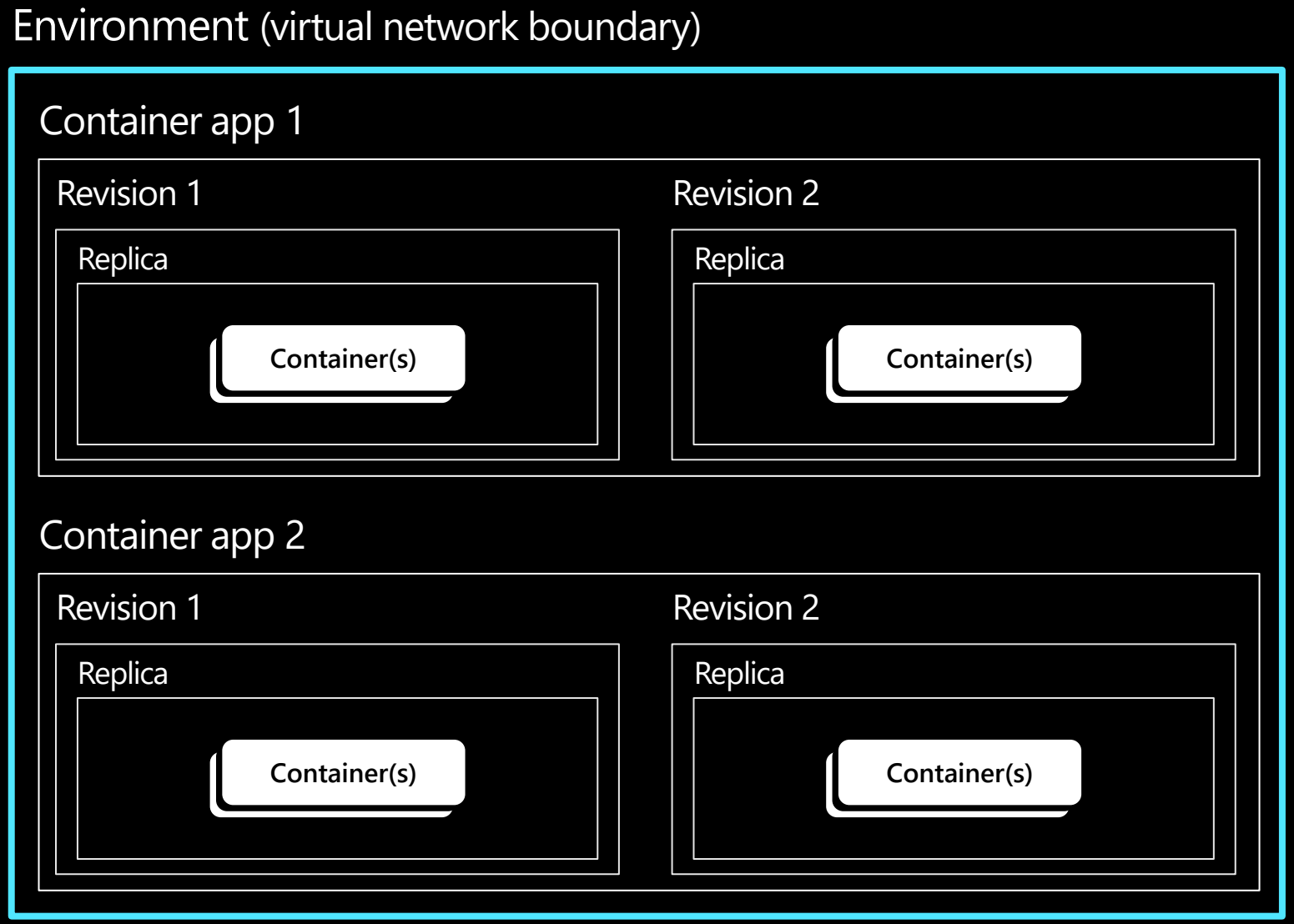
Scaling is determined by the number of concurrent HTTP requests

Scaling is determined by the number of messages in the queue

Scaling is determined by the level of CPU or memory load

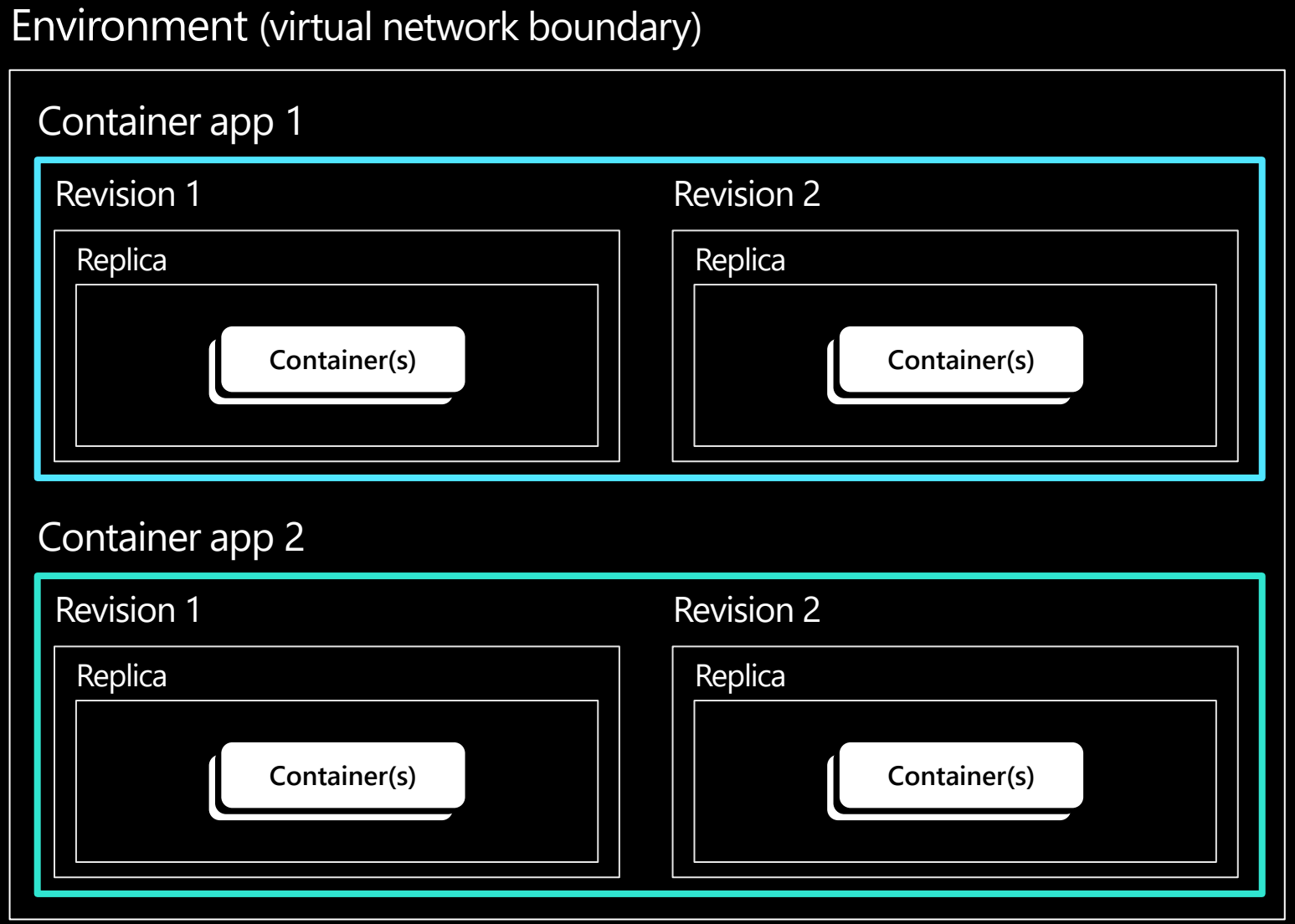
Environments

Environments define an isolation and observability boundary around a collection of container apps deployed in the same virtual network



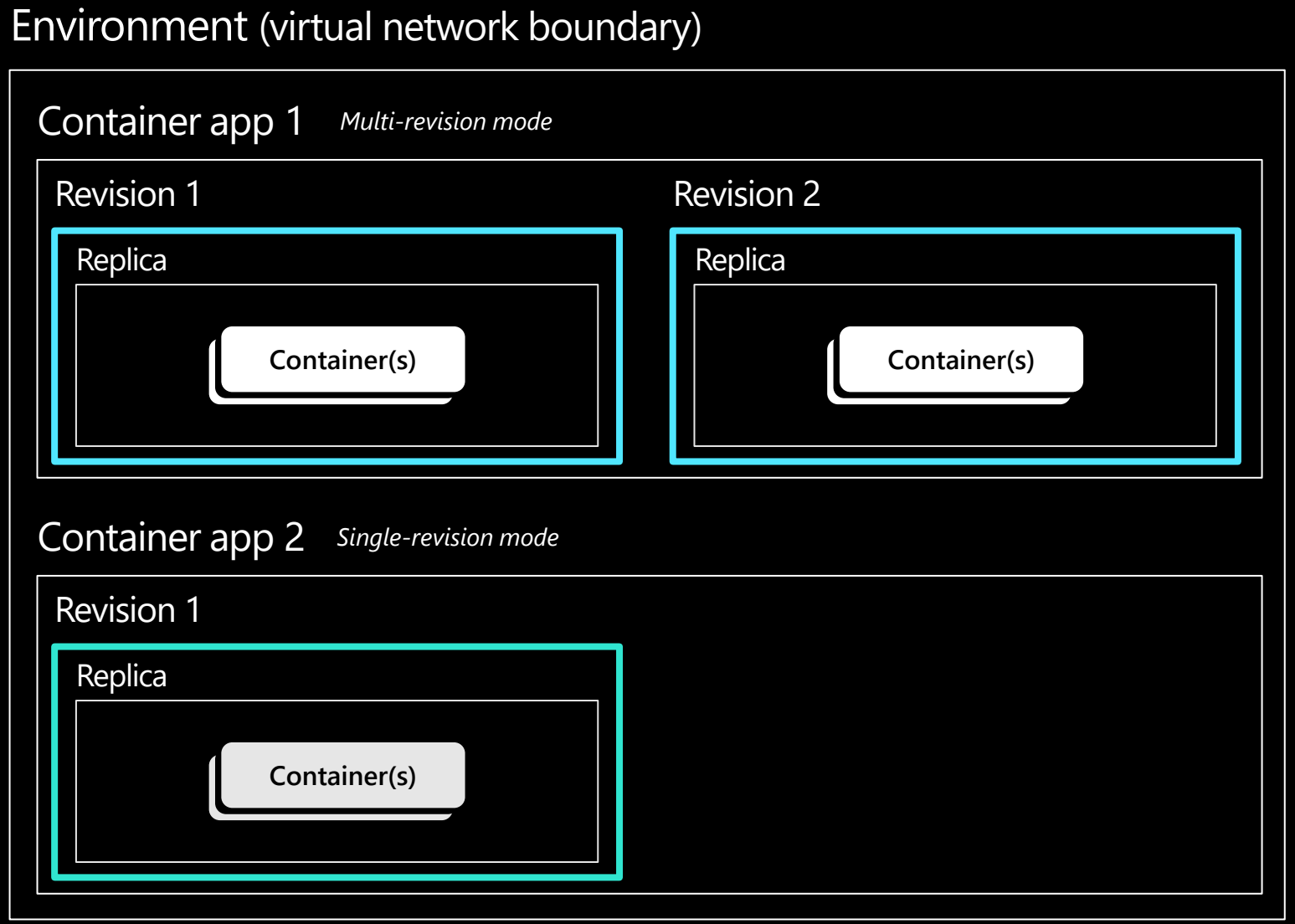
Container Apps

A Container App hosts a single, independent microservice and includes its desired state configuration



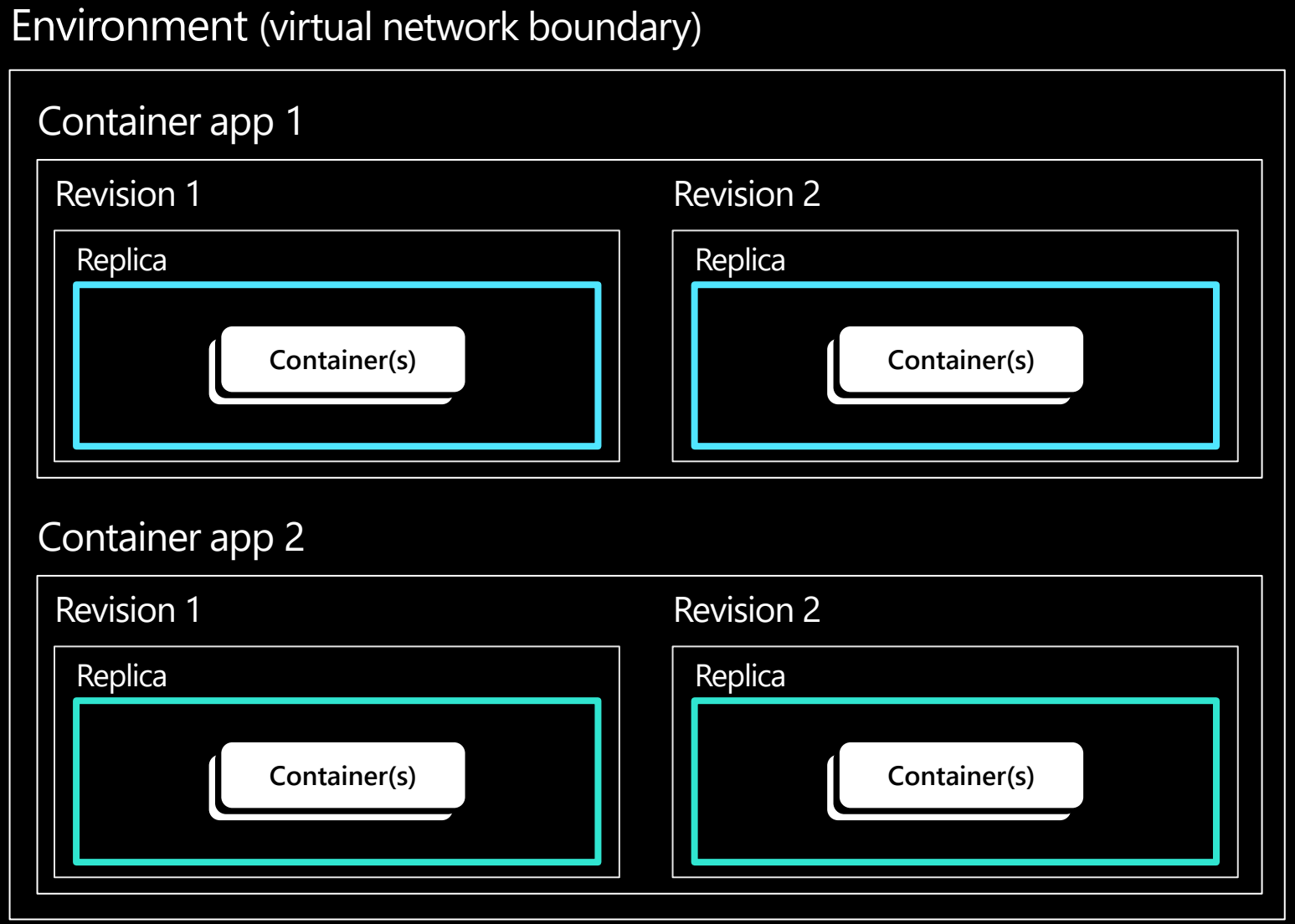
Revisions

Revisions are immutable version snapshots of a container app



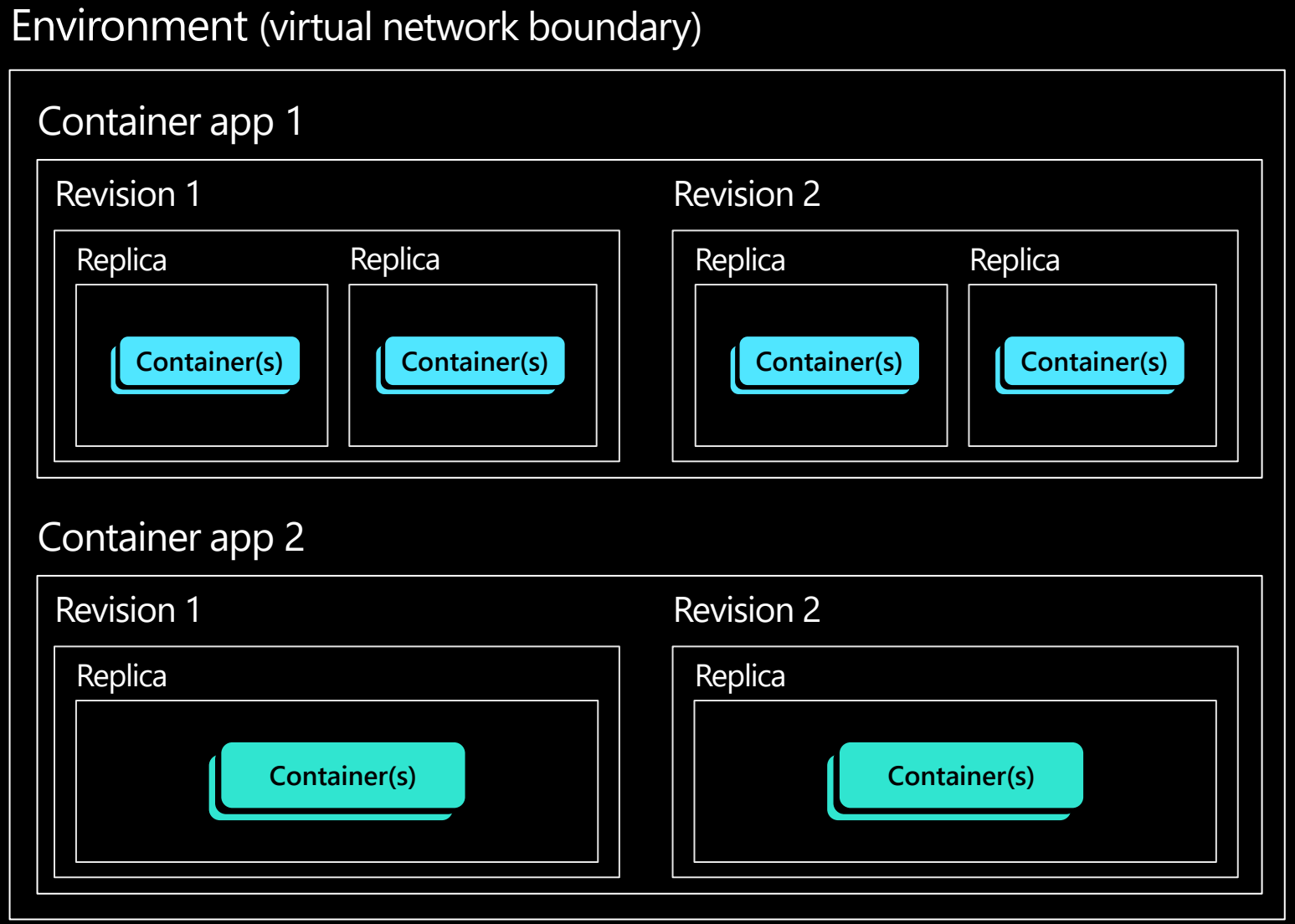
Replicas

Replicas are the unit of scale in container apps, with the default replica count being 0



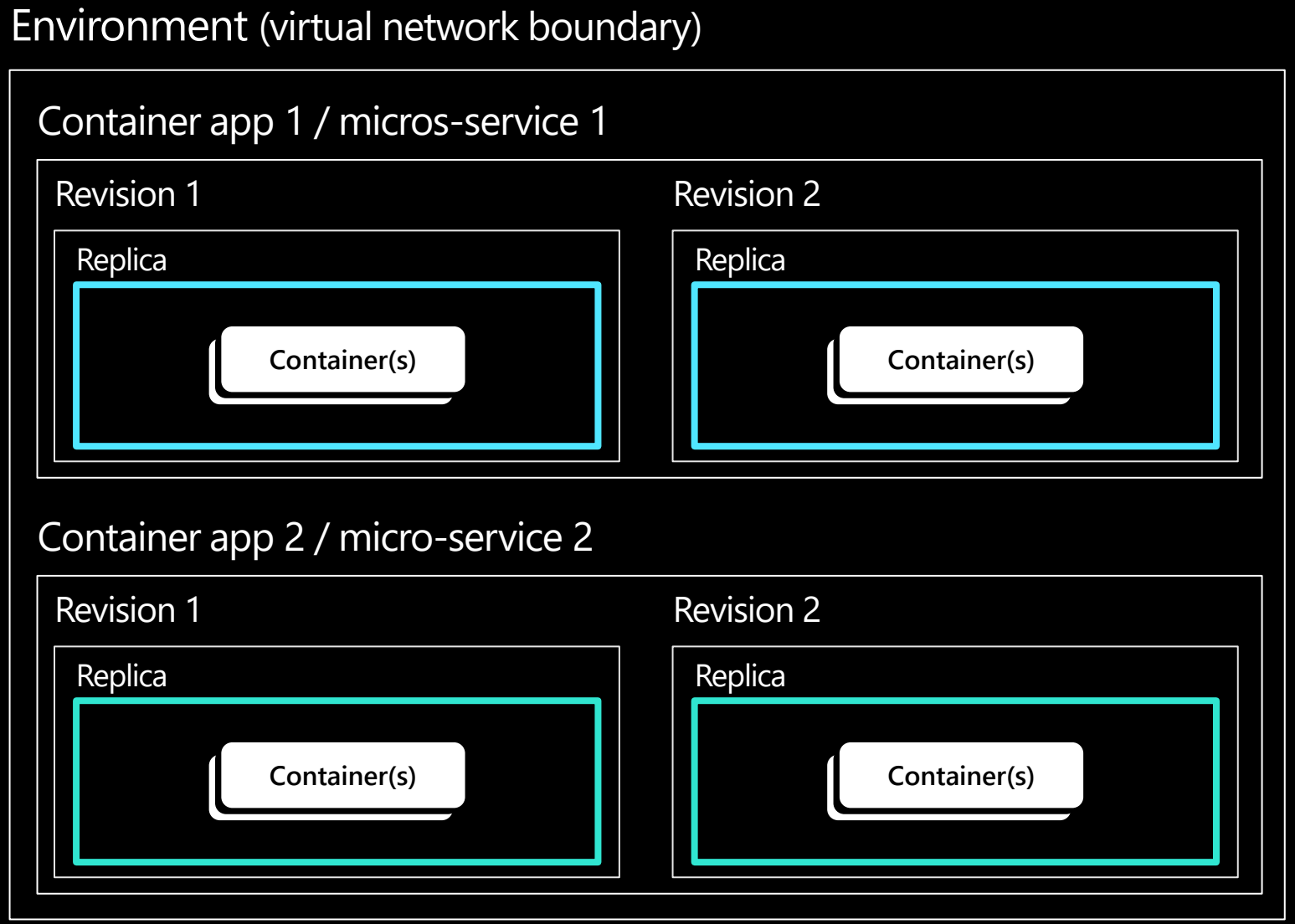
Containers

Containers in Azure Container Apps can use any development stack of your choice



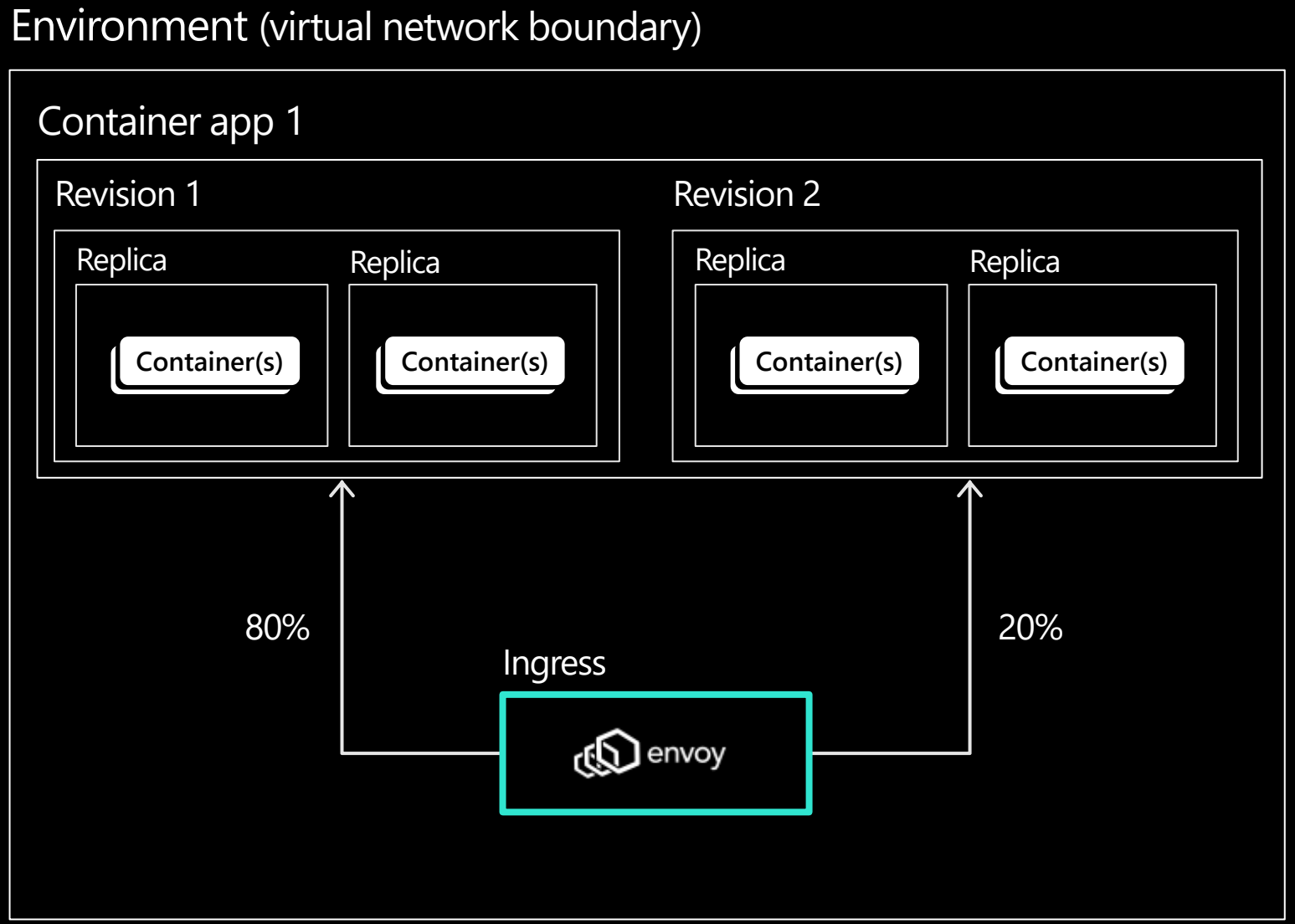
Microservices

Microservices are deployed as container apps



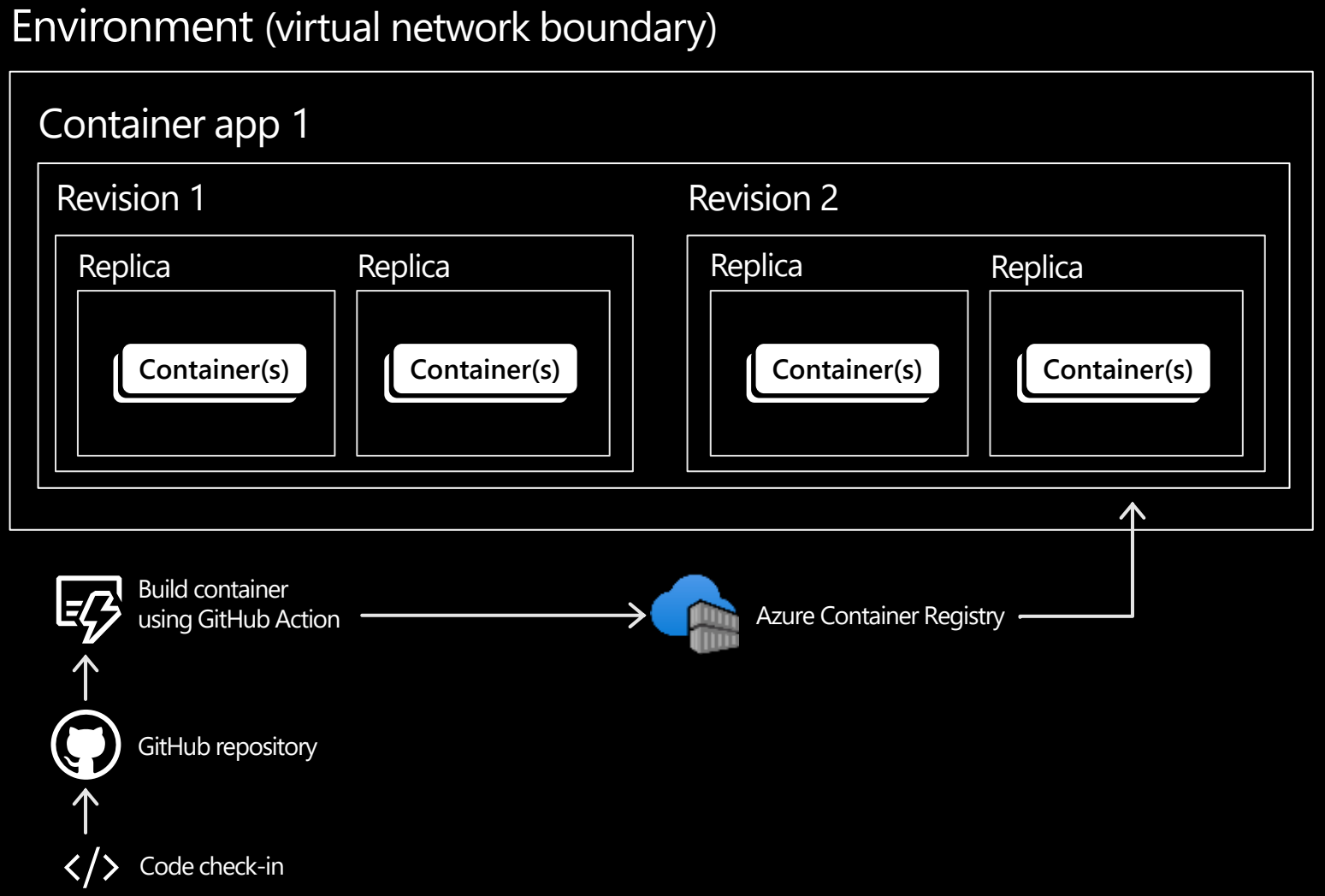
Ingress

Internal or external visibility with TLS termination and support for HTTP/1.1 and HTTP/2



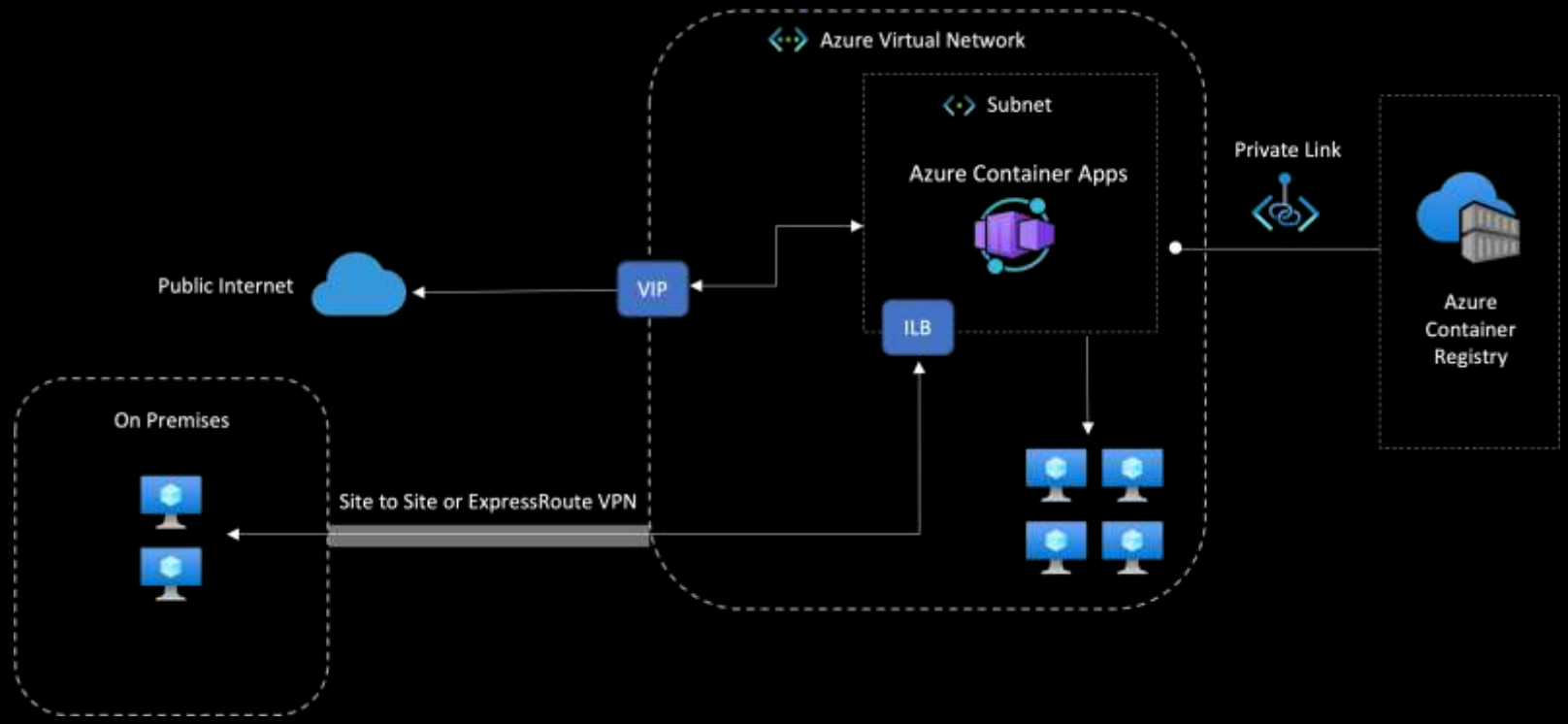
GitHub Actions integration

Publish revisions as commits are pushed to your GitHub repository by triggering a GitHub Action to build a new container image



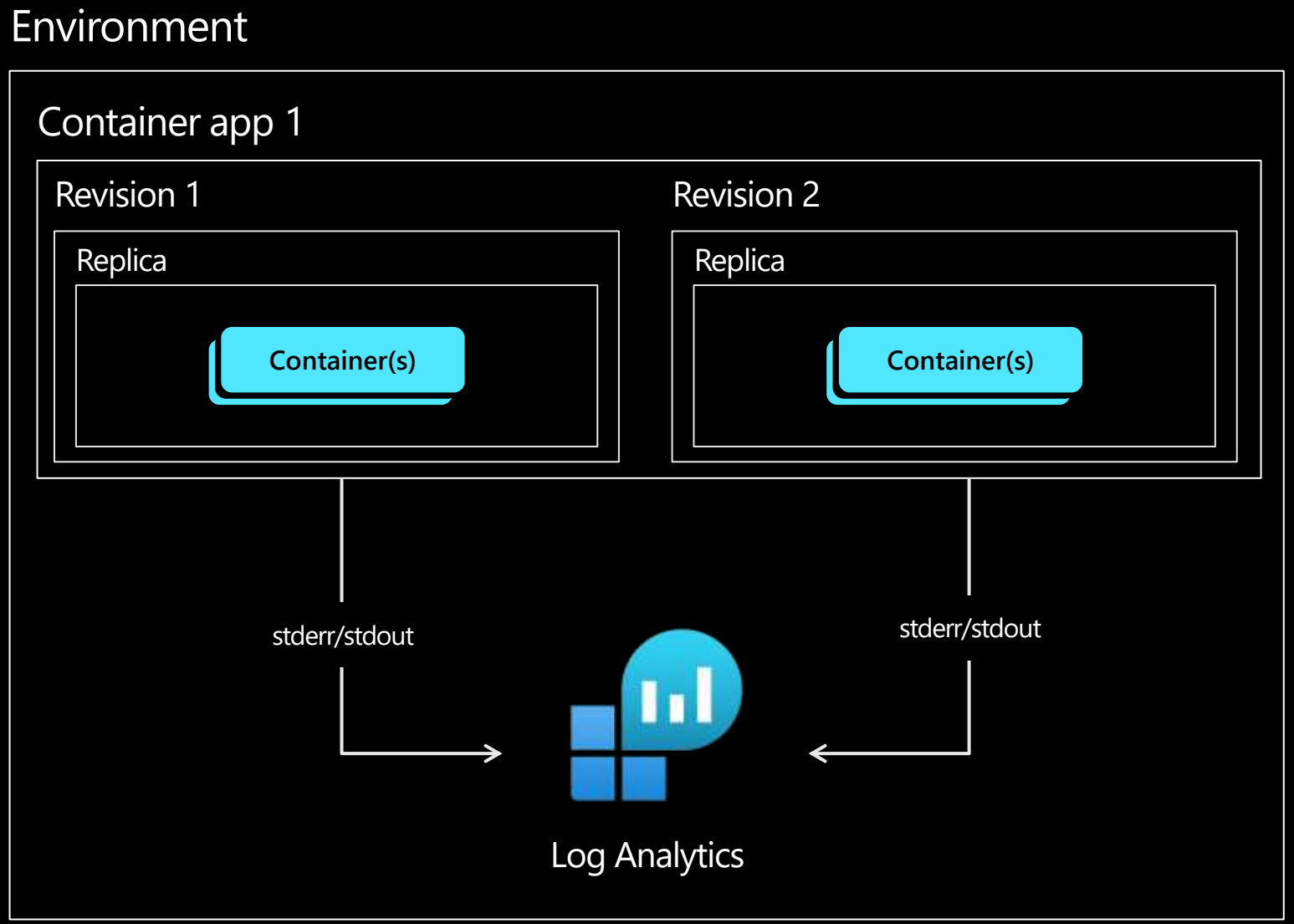
Bring your own Virtual Network

Deploy an environment using a custom virtual network and optionally without a public IP address



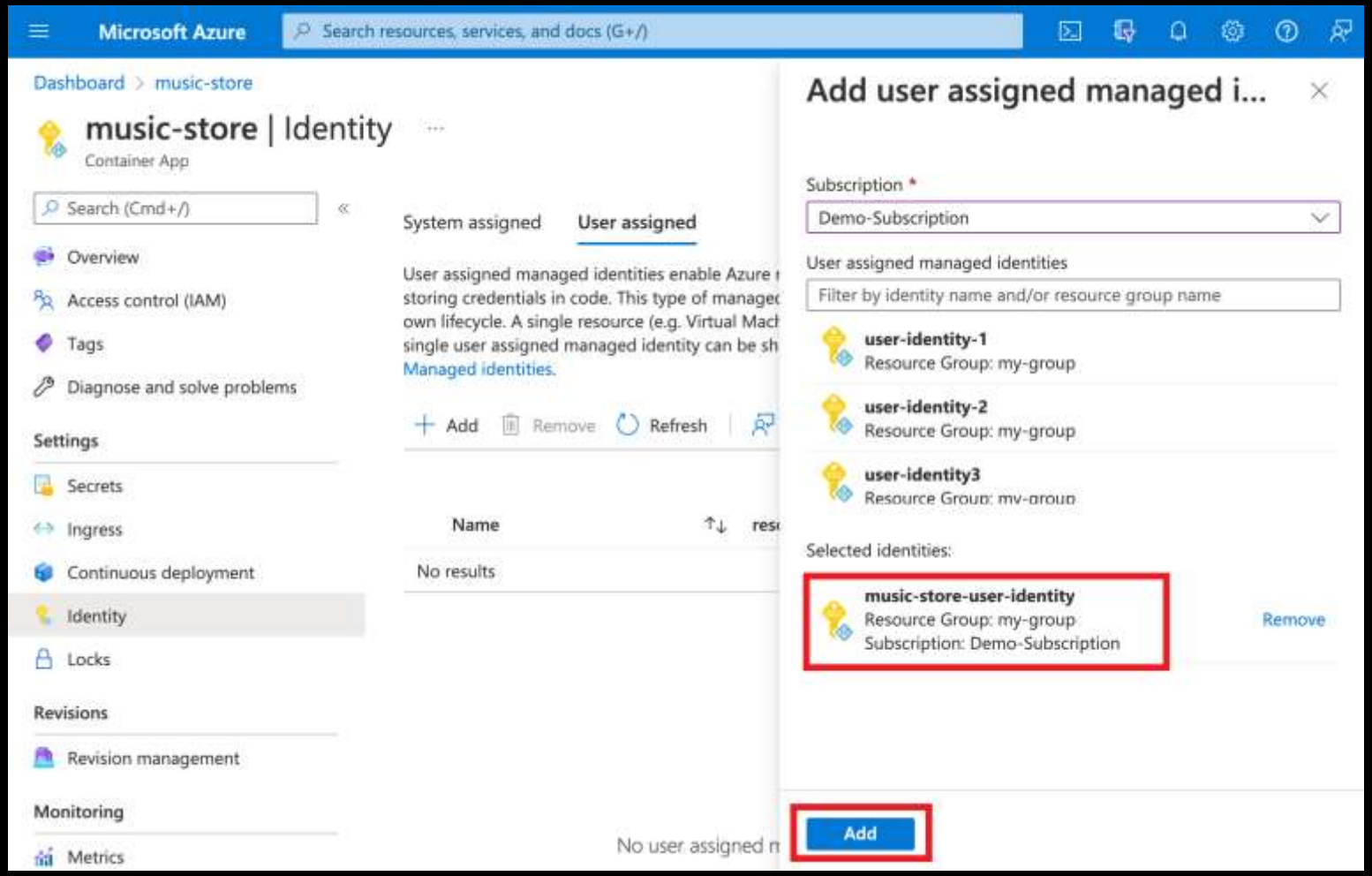
Logging

Containers write logs to standard output or standard error streams surfaced via Log Analytics



Managed identity

Access Azure resources without secrets



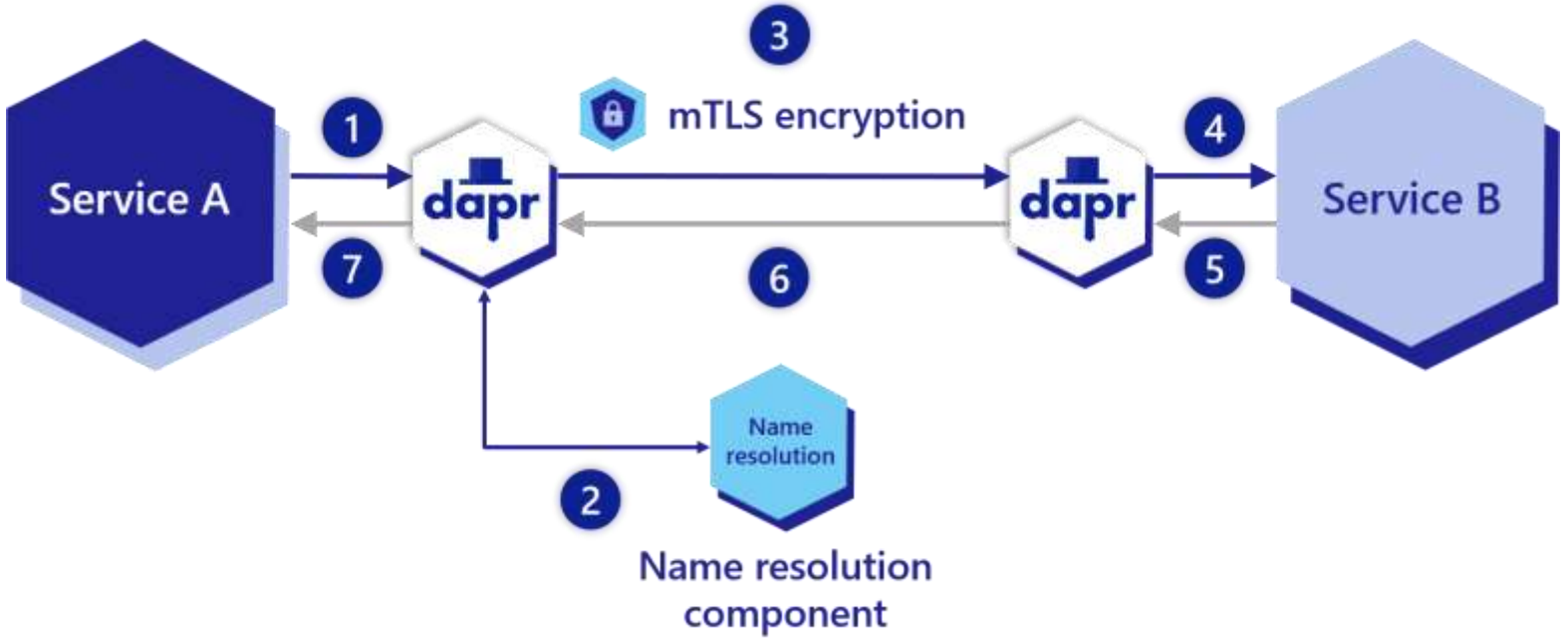
Secrets management

Securely store sensitive configuration elements that are then available to containers through environment variables, scale rules, and Dapr

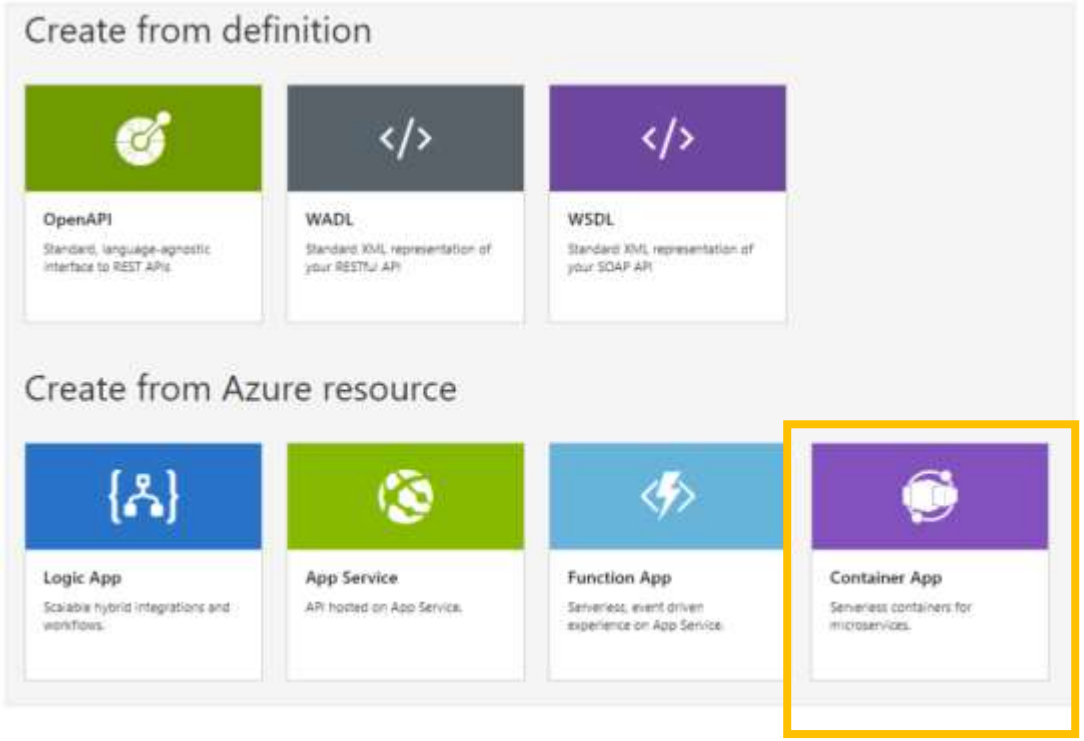
```
"template": {
    "containers": [
        {
            "image": "myregistry/myQueueApp:v1",
            "name": "myQueueApp",
            "env": [
                {
                    "name": "QueueName",
                    "value": "myqueue"
                },
                {
                    "name": "ConnectionString",
                    "secretref": "queue-connection-string"
                }
            ]
        }
    ],
}
```

demo

Dapr integration (mTLS, service discovery, tracing, etc.)



API Management Import

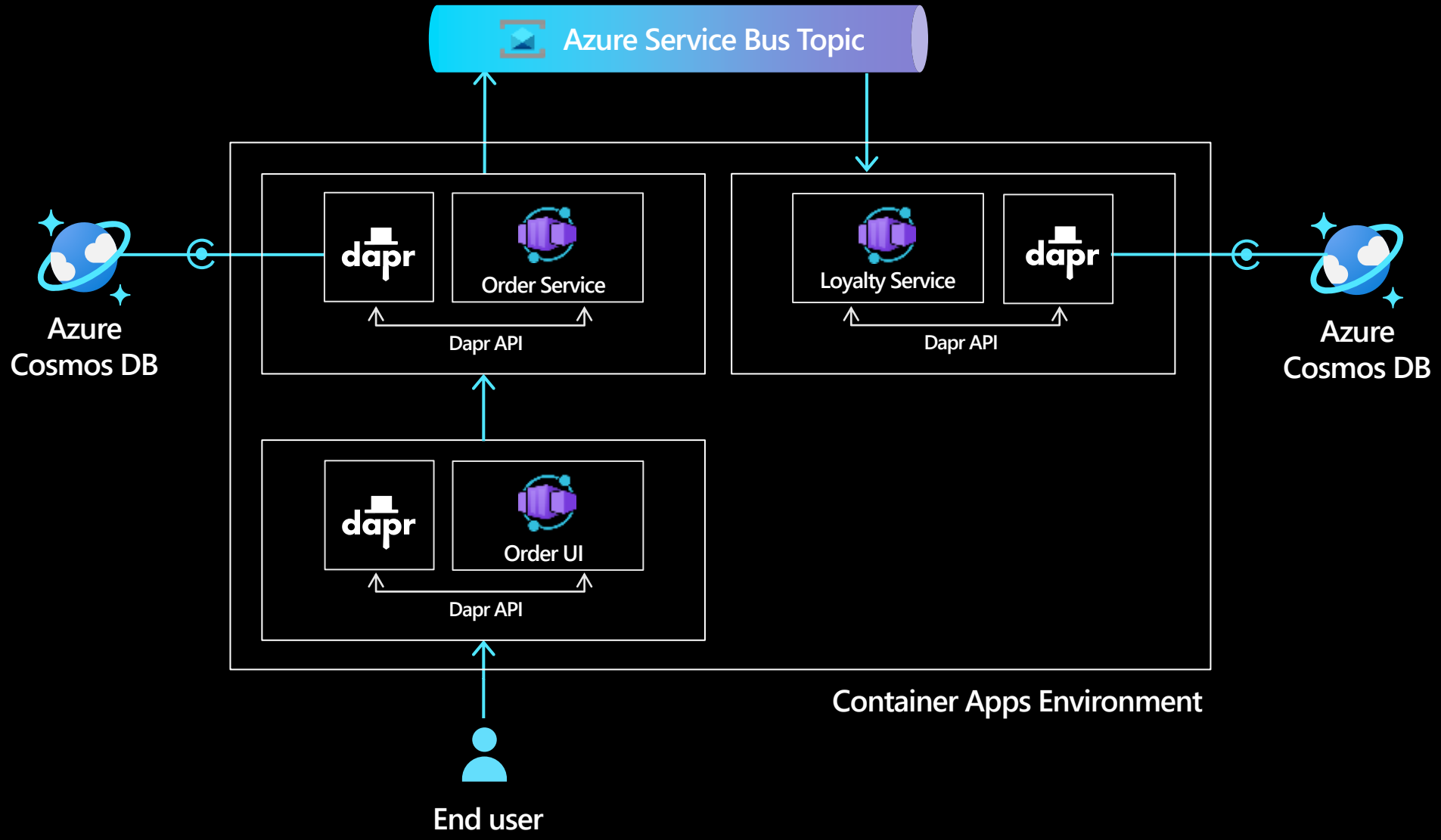


API Management will look in several locations for an OpenAPI Specification:

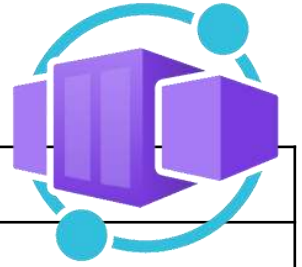
- The Container App configuration
- /openapi.json
- /openapi.yml
- /swagger/v1/swagger.json

https://docs.microsoft.com/en-us/azure/api-management/import-container-app-with-oas

Serverless microservices: Contoso Retail

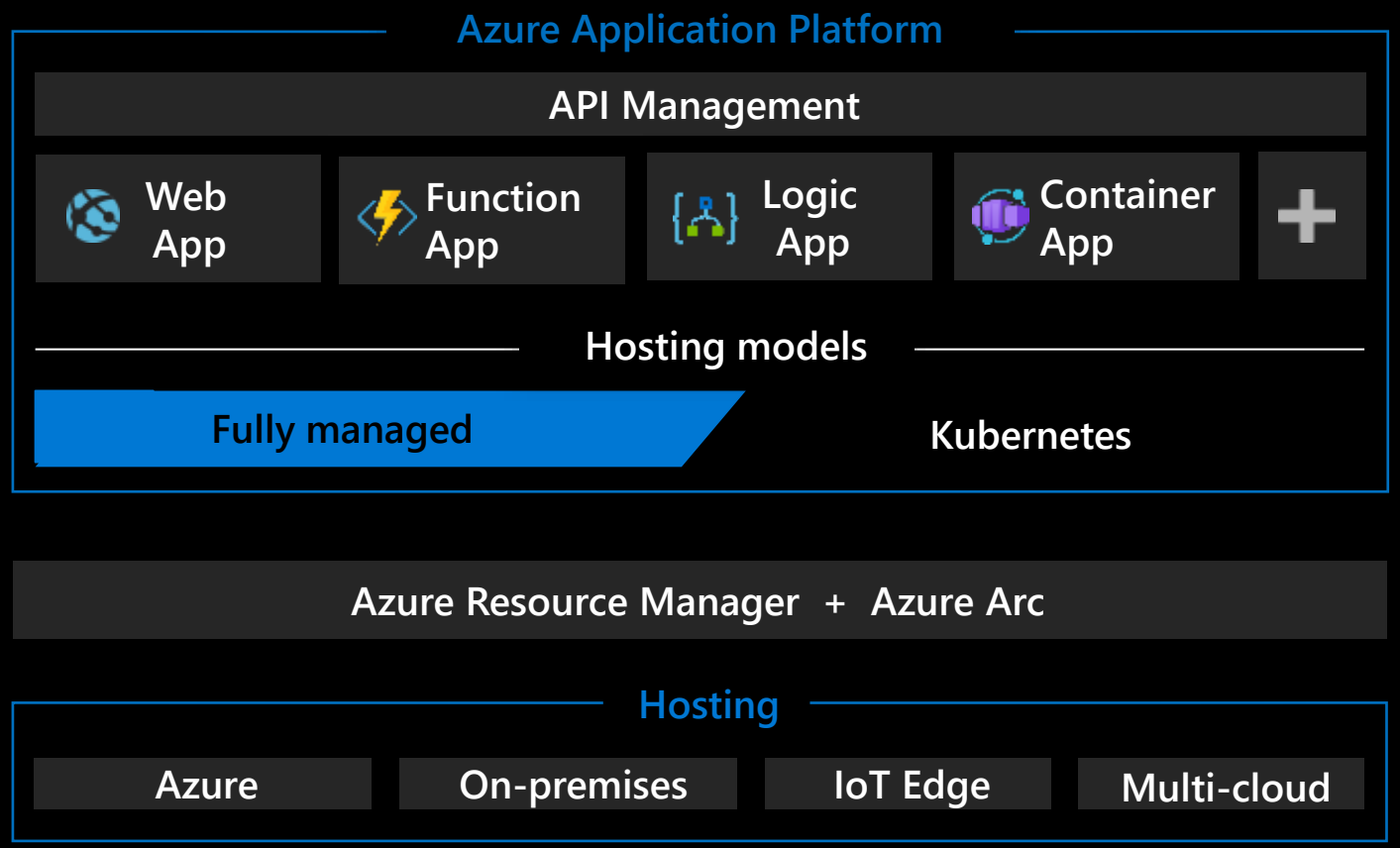


Apps and Jobs



Container	Compute	Notes
An HTTP server that serves web content and API requests	App	Configure an HTTP scale rule.
A process that generates financial reports nightly	Job	Use the *Schedule* job type and configure a cron expression.
A continuously running service that processes messages from an Azure Service Bus queue	App	Configure a custom scale rule.
A job that processes a single message or a small batch of messages from an Azure queue and exits	Job	Use the *Event* job type and configure a custom scale rule to trigger job executions.
A background task that's triggered on-demand and exits when finished	Job	Use the *Manual* job type and start executions manually or programmatically using an API.
A self-hosted GitHub Actions runner or Azure Pipelines agent	Job	Use the *Event* job type and configure a GitHub Actions or Azure Pipelines scale rule.
An Azure Functions app	App	Deploy Azure Functions to Container Apps.
An event-driven app using the Azure WebJobs SDK	App	Configure a scale rule for each event source.

Azure Dev Compute



How does ACA compare to AKS?

	 **Azure Kubernetes Service (AKS)** Infrastructure focus, higher flexibility	 **Azure Container Apps (ACA)** 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 • Operations flexibility with advanced customization • Experienced Kubernetes operators	• Platform-as-a-Service experience with serverless scale • Developer productivity with low operations overhead • Linux-based, general-purpose stateless containers
Interaction model	• Operators deploy node-based AKS clusters using Azure Portal, CLI or Infrastructure-as-Code templates (IaC) • 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 • 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. • 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 • Envoy: managed ingress and traffic splitting • KEDA: managed, event-driven autoscale • Dapr: codified best practices for microservices

App Service

- Azure App Service provides fully managed hosting for web applications including websites and web APIs.
- These web applications may be deployed using code or containers. Azure App Service is optimized for web applications.
- Azure App Service is integrated with other Azure services including Azure Container Apps or Azure Functions.
- When building web apps, Azure App Service is an ideal option.

Scaling and using the Kubernetes Event Driven Autoscaling (KEDA)

Application autoscaling made simple

Open-source, extensible, and vendor agnostic

Kubernetes-based Event Driven Autoscaler

Drive the scaling of any container based on a growing list of 35+ event sources, known as: scalers

Metrics Adapter | Controller | Scaler

Event-driven

Intelligently scale your event-driven applications

Built-in scalers

Out-of-the-box scalers for various vendors, databases, messaging systems, telemetry systems, CI/CD, and more

Vendor-agnostic

Support for triggers across variety of cloud providers & products

Rich capabilities

Bring rich scaling to every workload

keda.sh

CLOUD NATIVE COMPUTING FOUNDATION

Scaling

KEDA

HTTP

```
{
 "name": "http-rule",
  "http": {
   "metadata": {
    "concurrentRequests": 50
   }
  }
}
```

Event-driven

artemis-queue, kafka, aws-cloudwatch, aws-kinesis-stream, aws-sqs-queue, azure-blob, azure-eventhub, azure-servicebus, azure-queue, cron, external, gcp-pubsub, huawei-cloudeye, ibmmq, influxdb, mongodb, mssql, mysql, postgresql, rabbitmq, redis, redis-streams, selenium-grid, solace-event-queue, ..

CPU

```
{
 "name": "cpu-rule",
  "custom": {
   "type": "cpu",
    "metadata": {
     "type": "Utilization",
      "value": "50"
   }
  }
}
```

Memory

```
{
 "name": "mem-rule",
  "custom": {
   "type": "memory",
    "metadata": {
     "type": "AverageValue",
      "value": "512"
   }
  }
}
```

Support for scale to zero and specifying minimum/maximum replicas

Support for specifying minimum/maximum replicas