



# Azure Container Apps

## Jiří Landsman





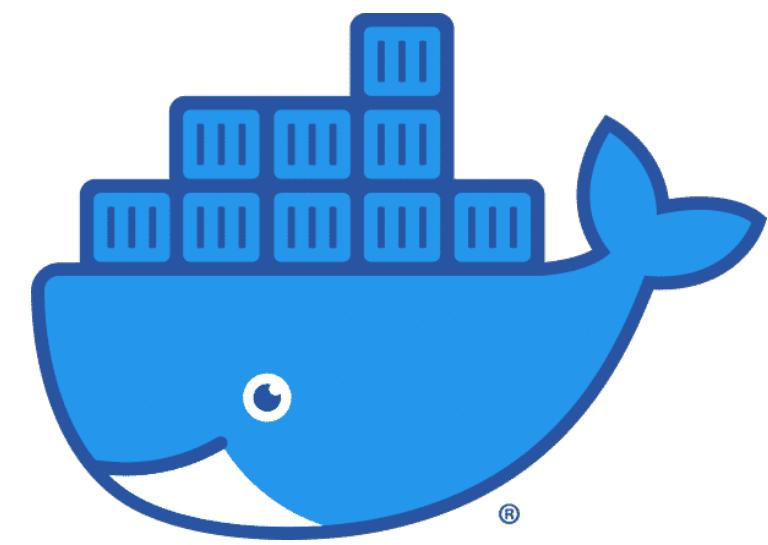




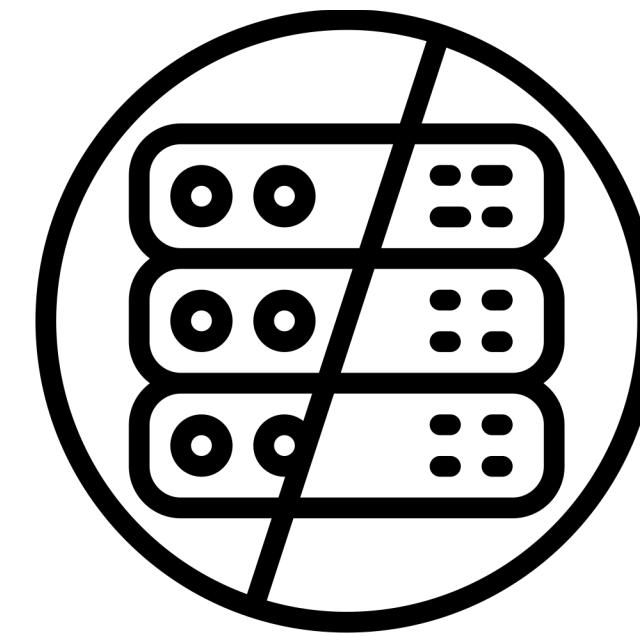


# Azure Container Apps

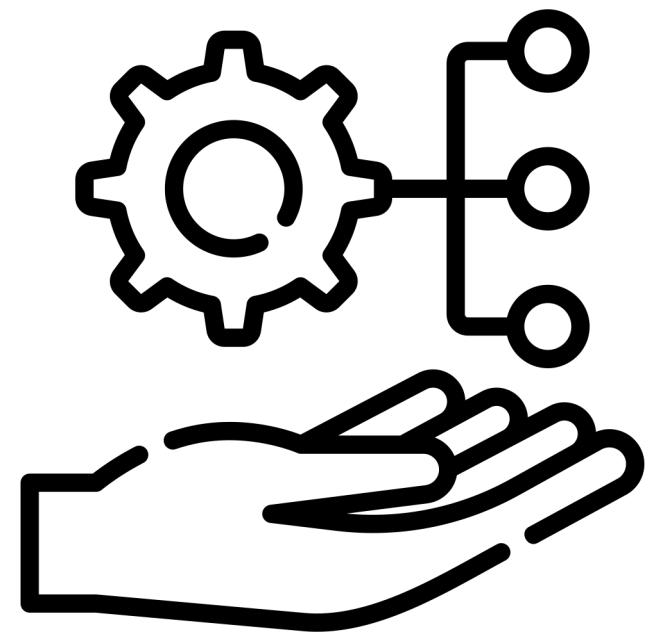
# Azure Container Apps



Docker



Serverless



Fully managed  
service



Build using  
opensource

# Azure Container Apps



Kubernetes

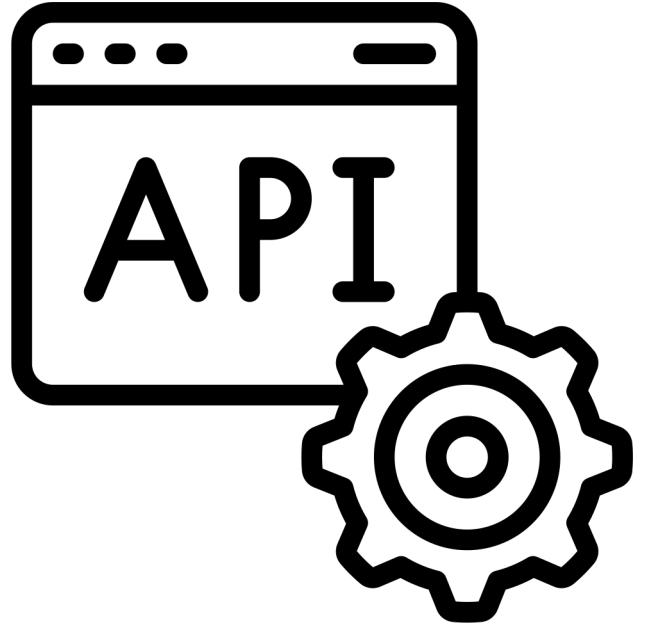


Kubernetes event  
driven autoscaler

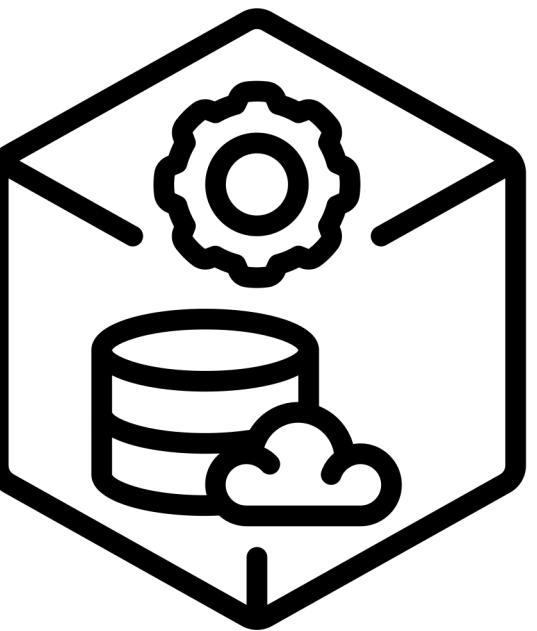


Distributed application  
runtime

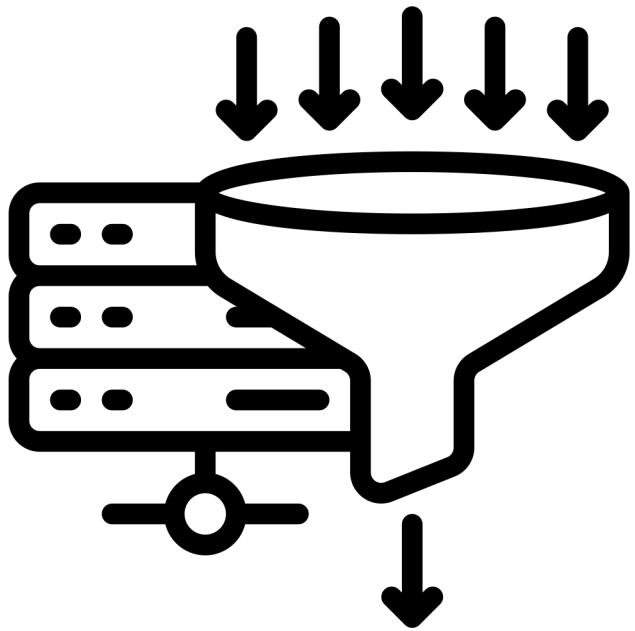
# Azure Container Apps



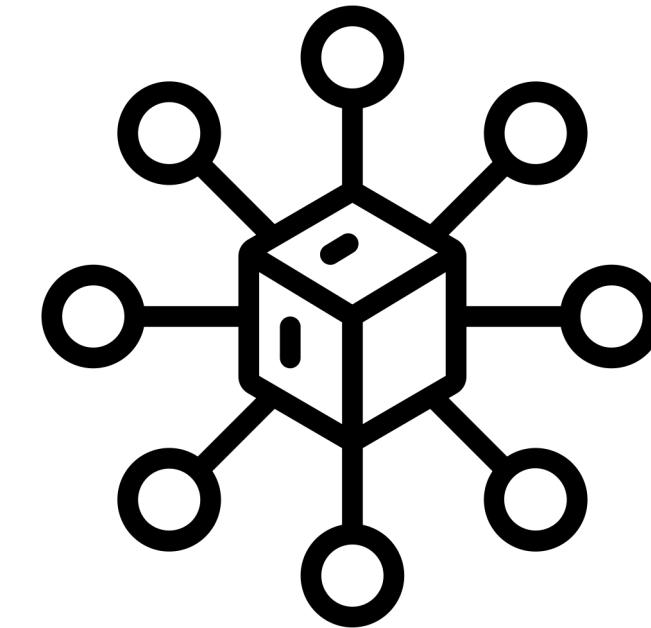
Public API endpoints



Background  
Processing



Event-driven  
processing

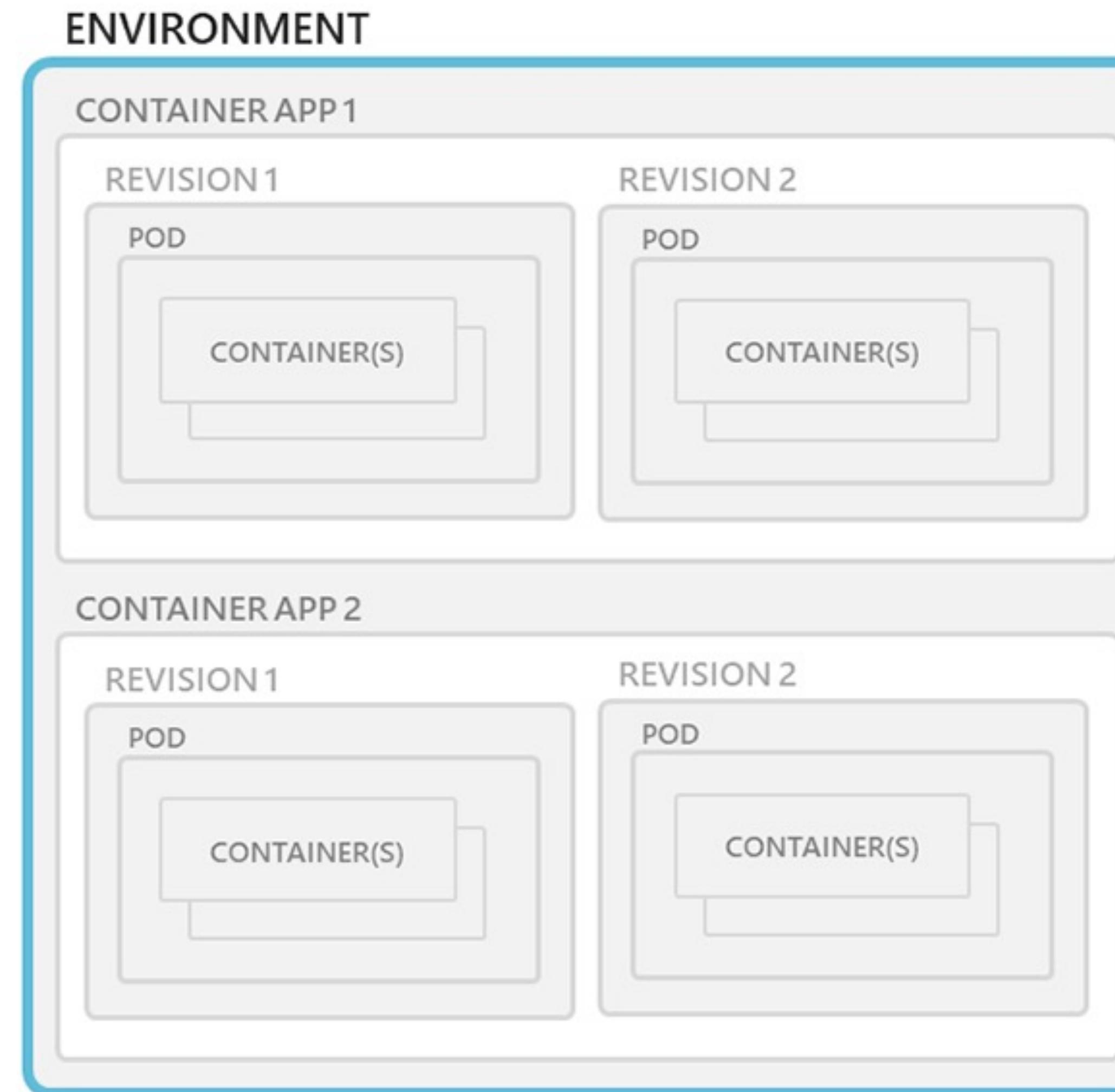


Microservices

# Azure Container Apps - Environments



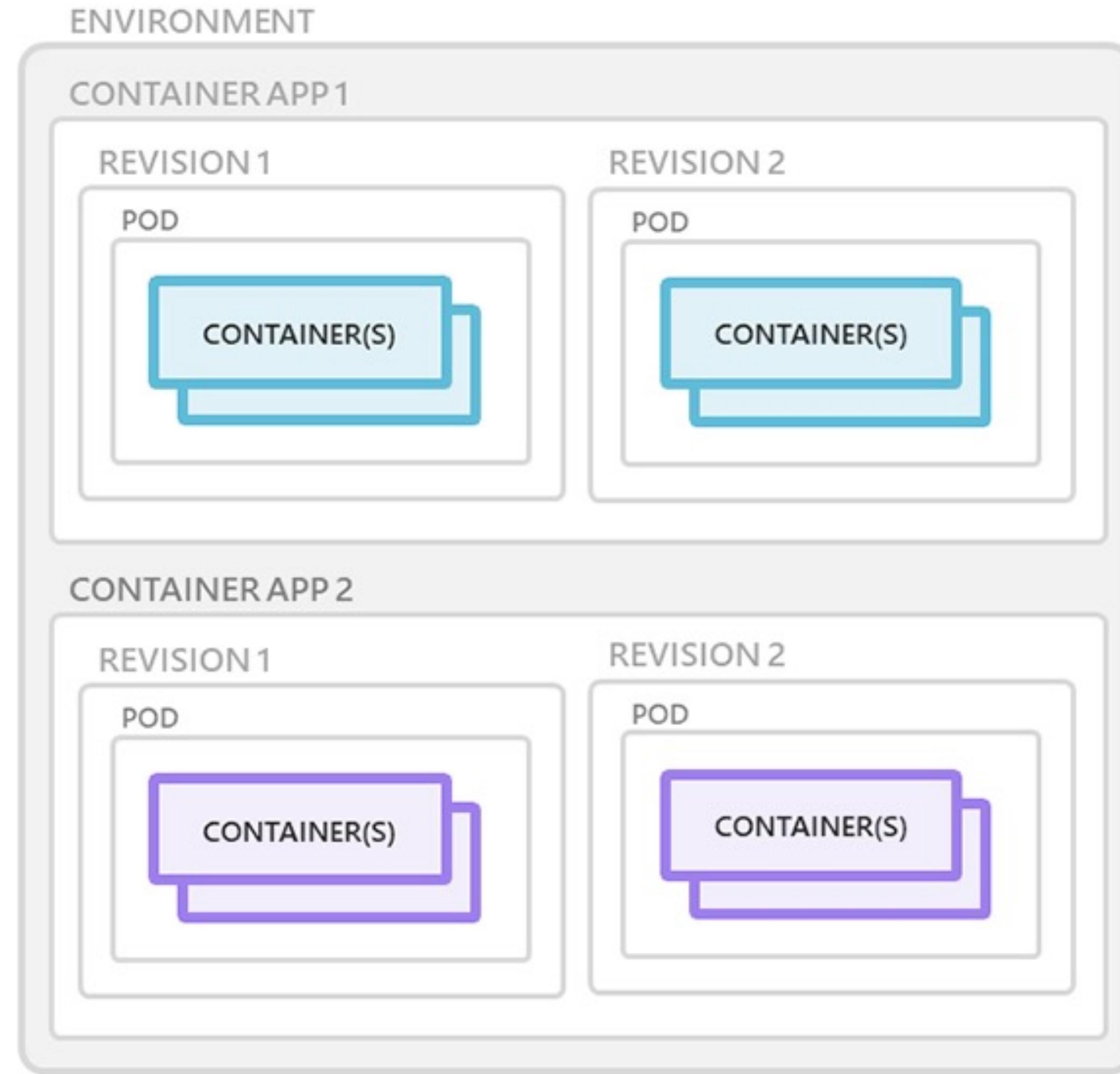
**Environments** are an isolation boundary around a collection of container apps.



# Azure Container Apps - Containers



**Containers** for an Azure Container App are grouped together in pods inside revision snapshots.



# Azure Container Apps - Containers



Any Docker repository.  
Public or private



Linux images

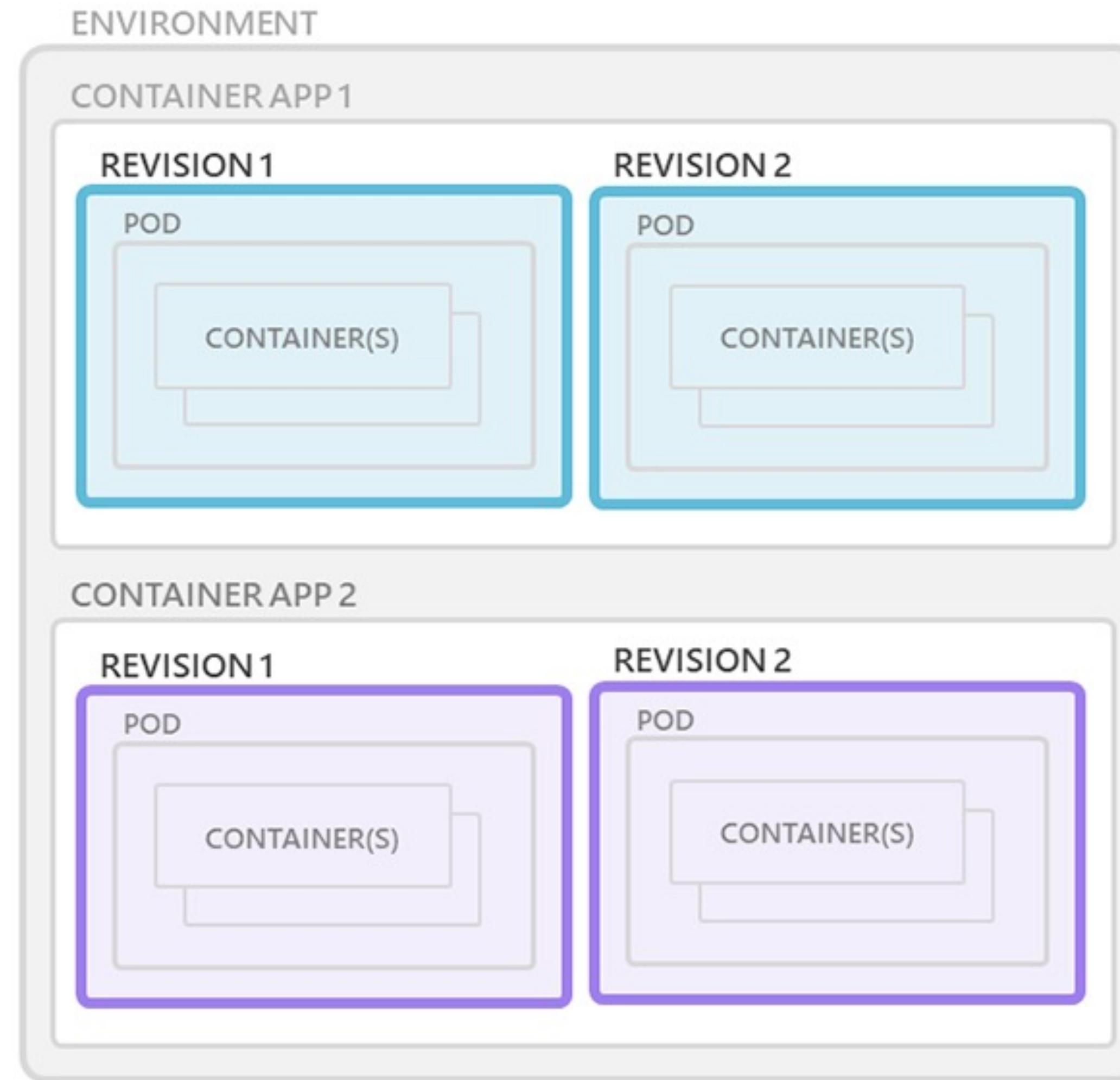


No Privileged  
containers

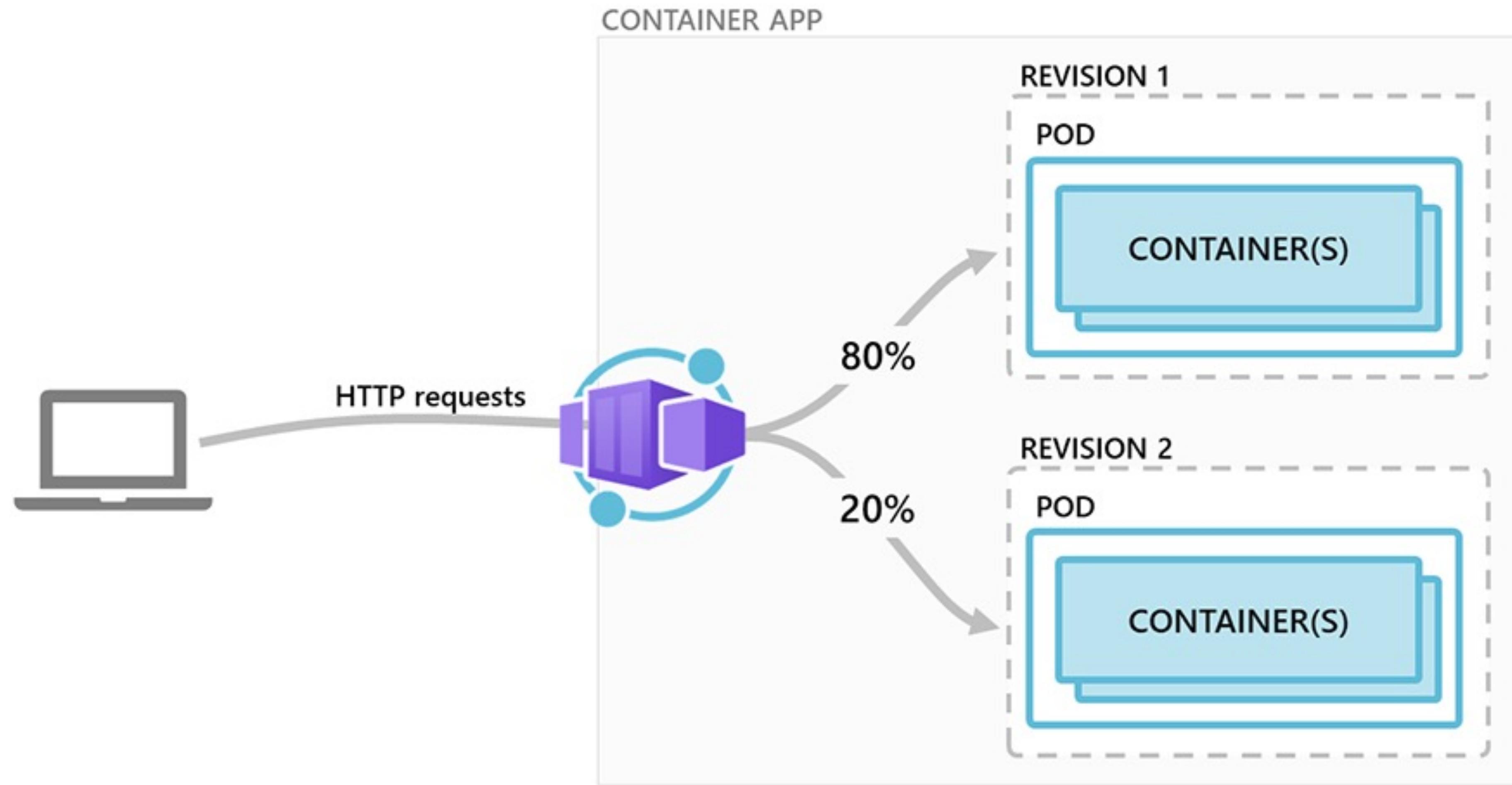
# Azure Container Apps - Revisions



**Revisions** are immutable snapshots of a container app.



# Azure Container Apps - Revisions



# Azure Container Apps - Observability

/liveness

/startup

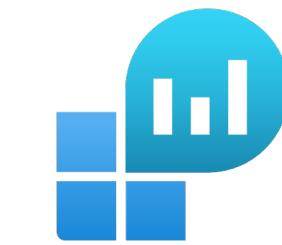
/readiness

Health probes

HTTP / TCP



Azure Monitor



Log Analytics Workspace



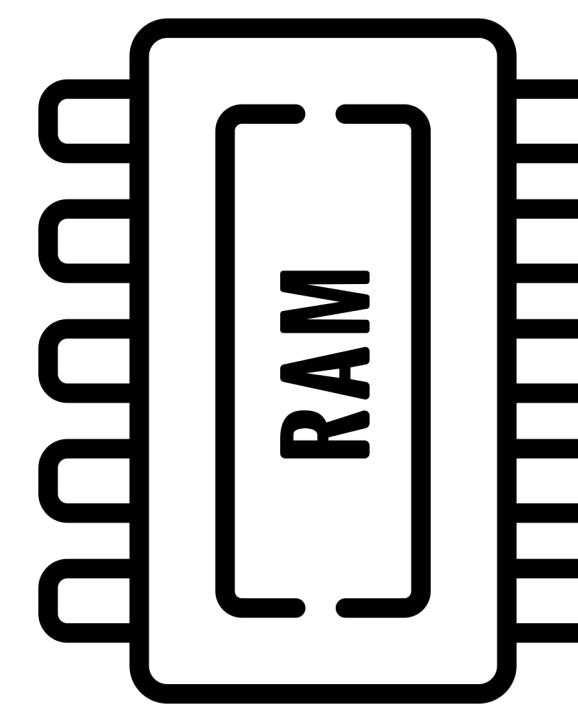
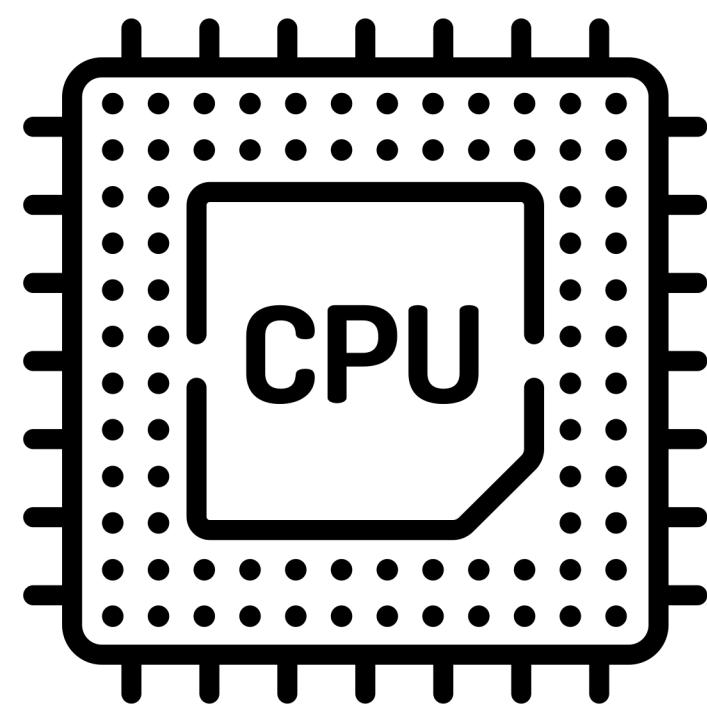
Metrics



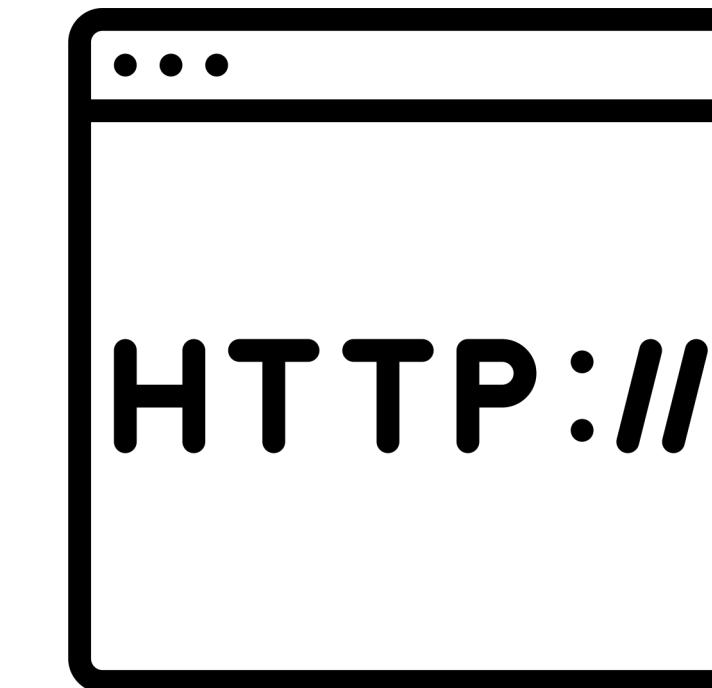
Alerts

# Azure Container Apps - Scaling

Vertical

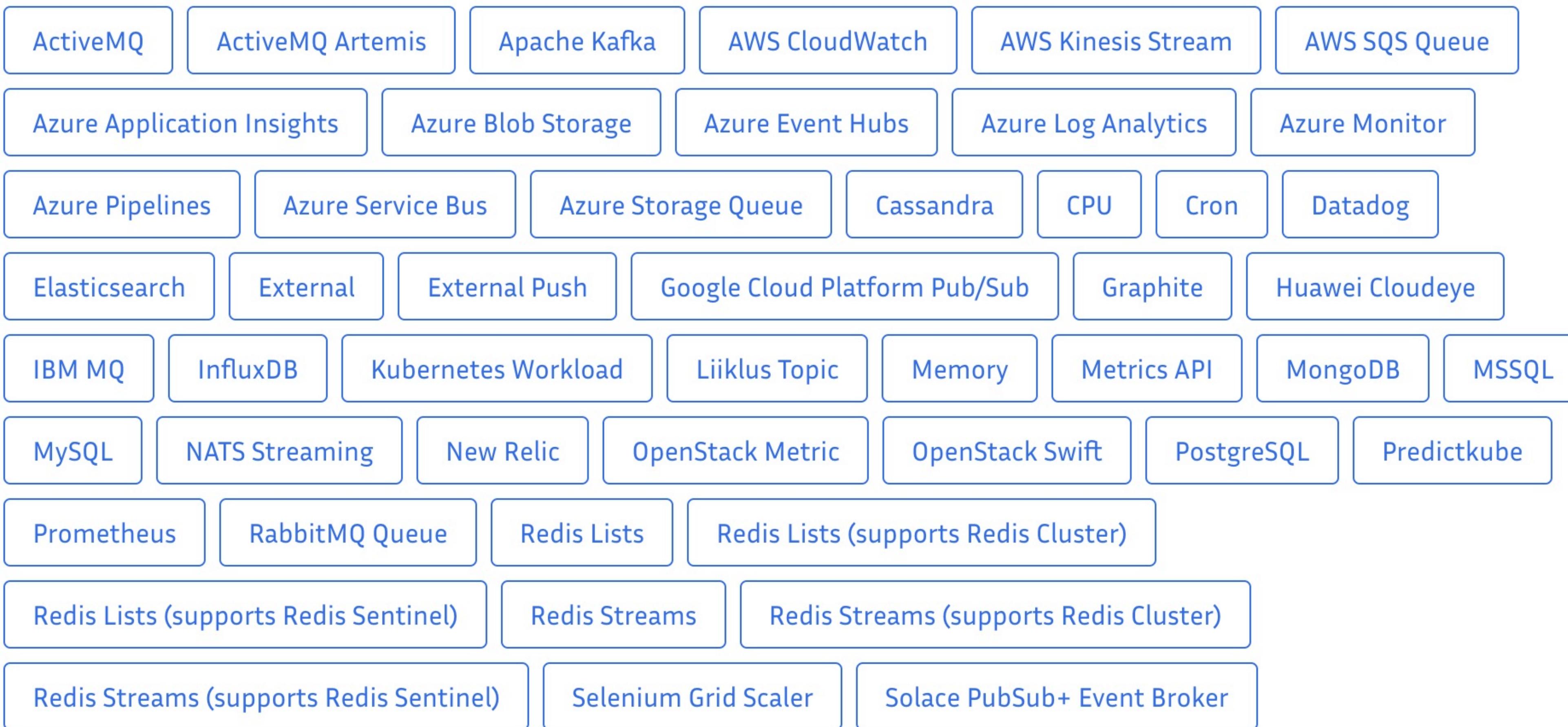


Horizontal

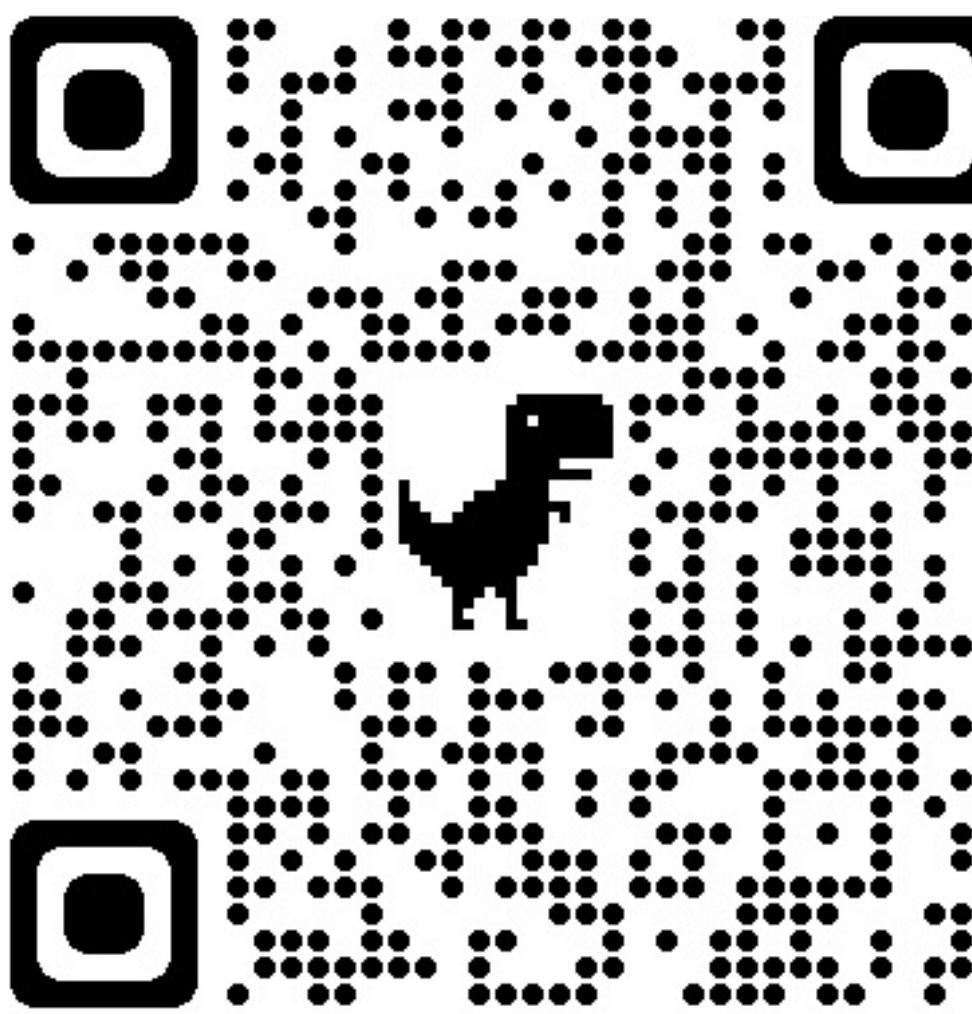


0 – 10 replicas

# Azure Container Apps - Scaling

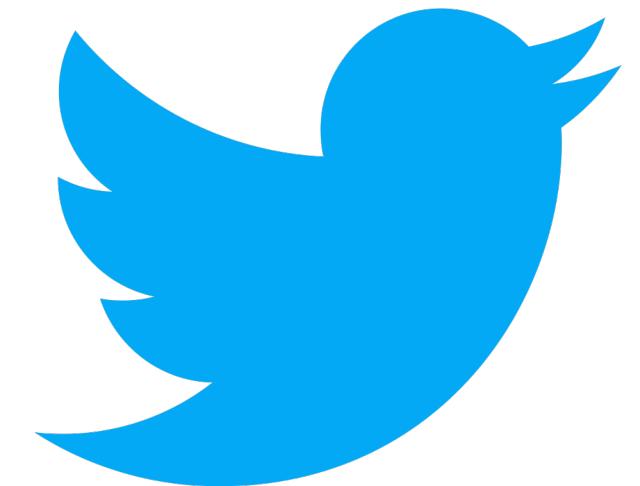
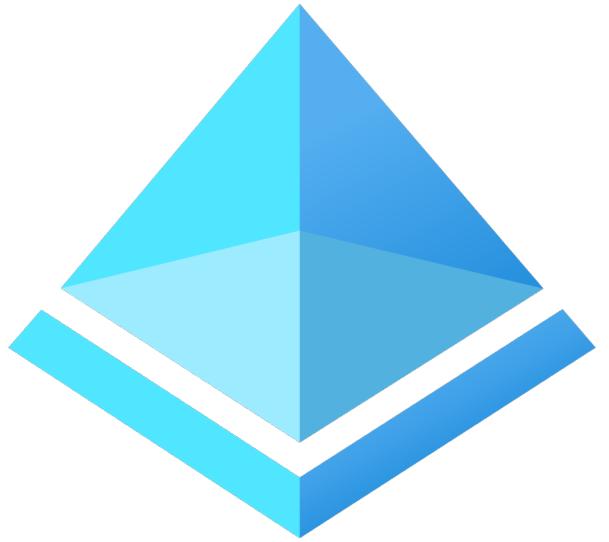


# KEDA - Documentation



<https://keda.sh/>

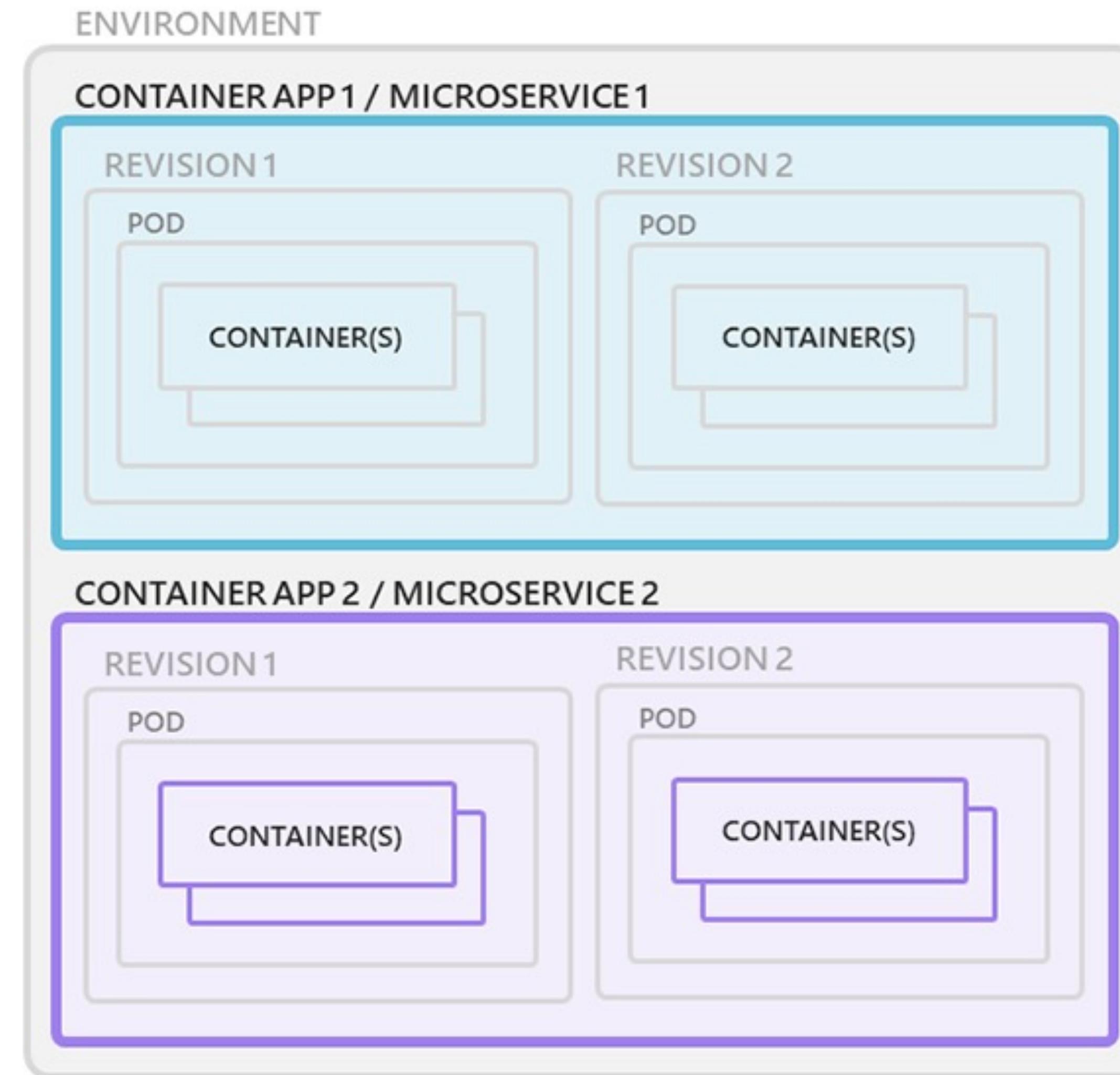
# Azure Container Apps - Authentication



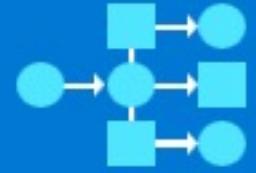
# Azure Container Apps - Microservices



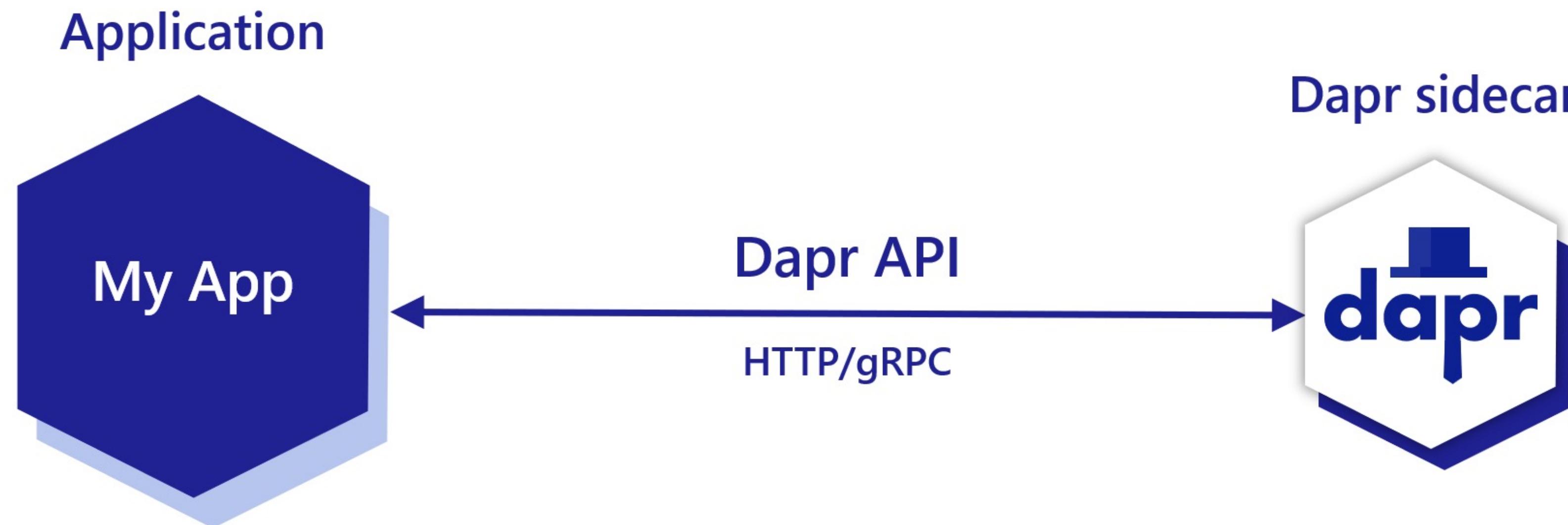
Container apps are deployed as **microservices**.



# Distributed Application Runtime - Dapr

 Service-to-service invocation	 State management	 Publish and subscribe	 Bindings (input/output)	 Actors	 Observability	 Secrets	 Configuration
Perform direct, secure, service-to-service method calls	Create long running, stateless and stateful services	Secure, scalable messaging between services	Trigger code through events from a large array of inputs Input and output bindings to external resources including databases and queues	Encapsulate code and data in reusable actor objects as a common microservices design pattern	See and measure the message calls across components and networked services	Securely access secrets from your application	Access application configuration and be notified of updates

# Dapr - Architecture



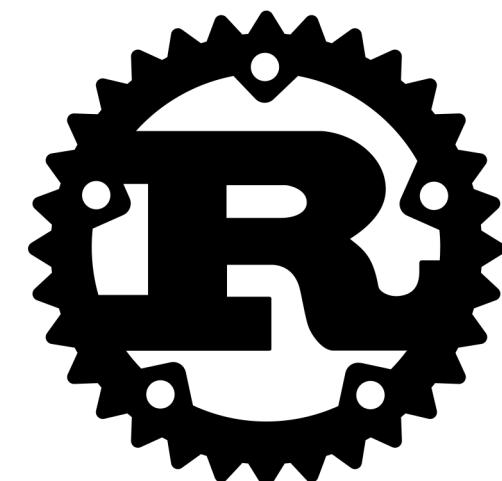
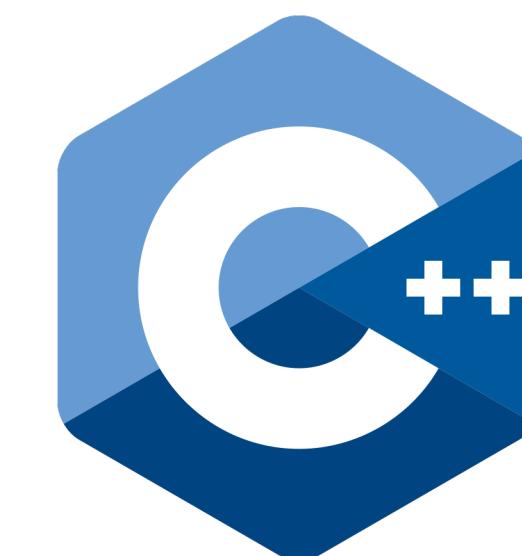
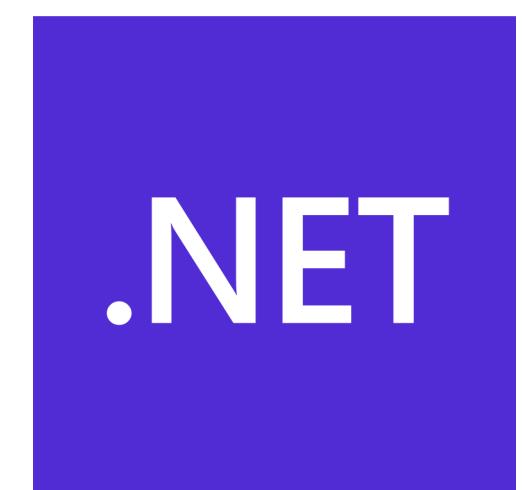
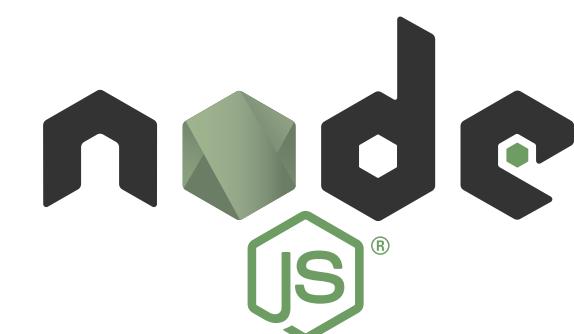
**POST** `http://localhost:3500/v1.0/invoke/cart/method/neworder`

**GET** `http://localhost:3500/v1.0/state/inventory/item67`

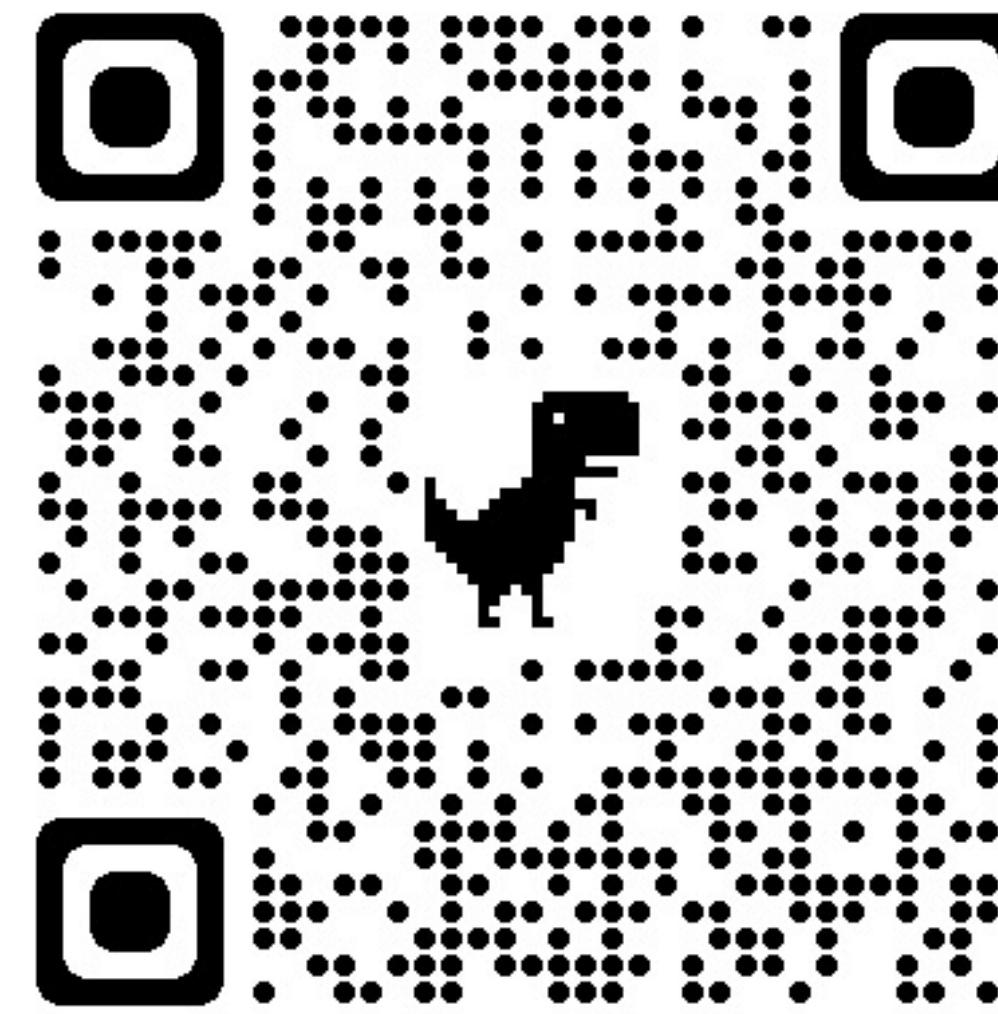
**POST** `http://localhost:3500/v1.0/publish/shipping/orders`

**GET** `http://localhost:3500/v1.0/secrets/keyvault/password`

# Dapr – Platforms/Languages

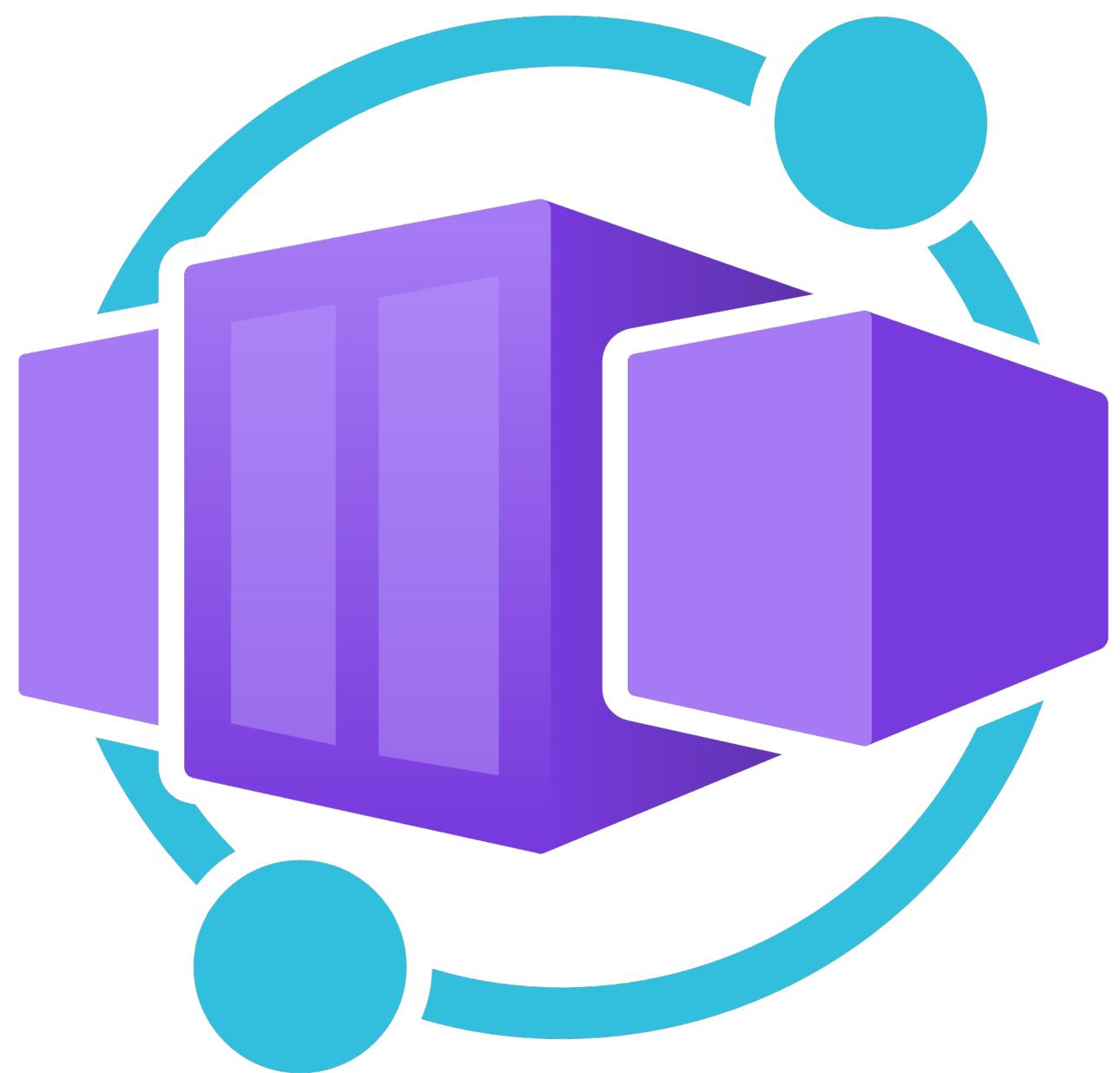


# Dapr - Documentation

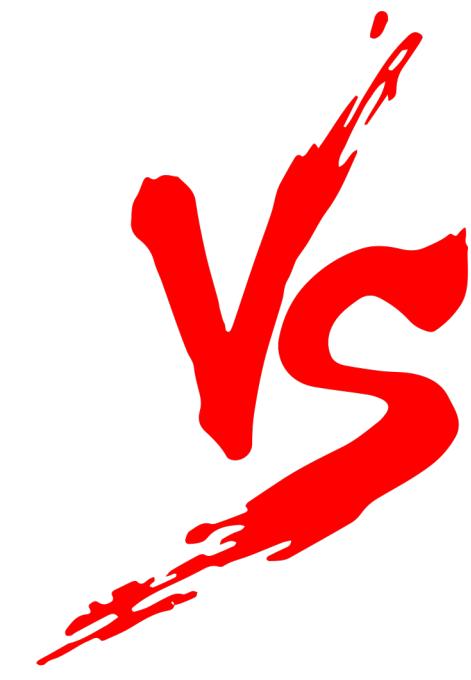


<https://docs.dapr.io/>

# Azure Container Apps - Alternatives

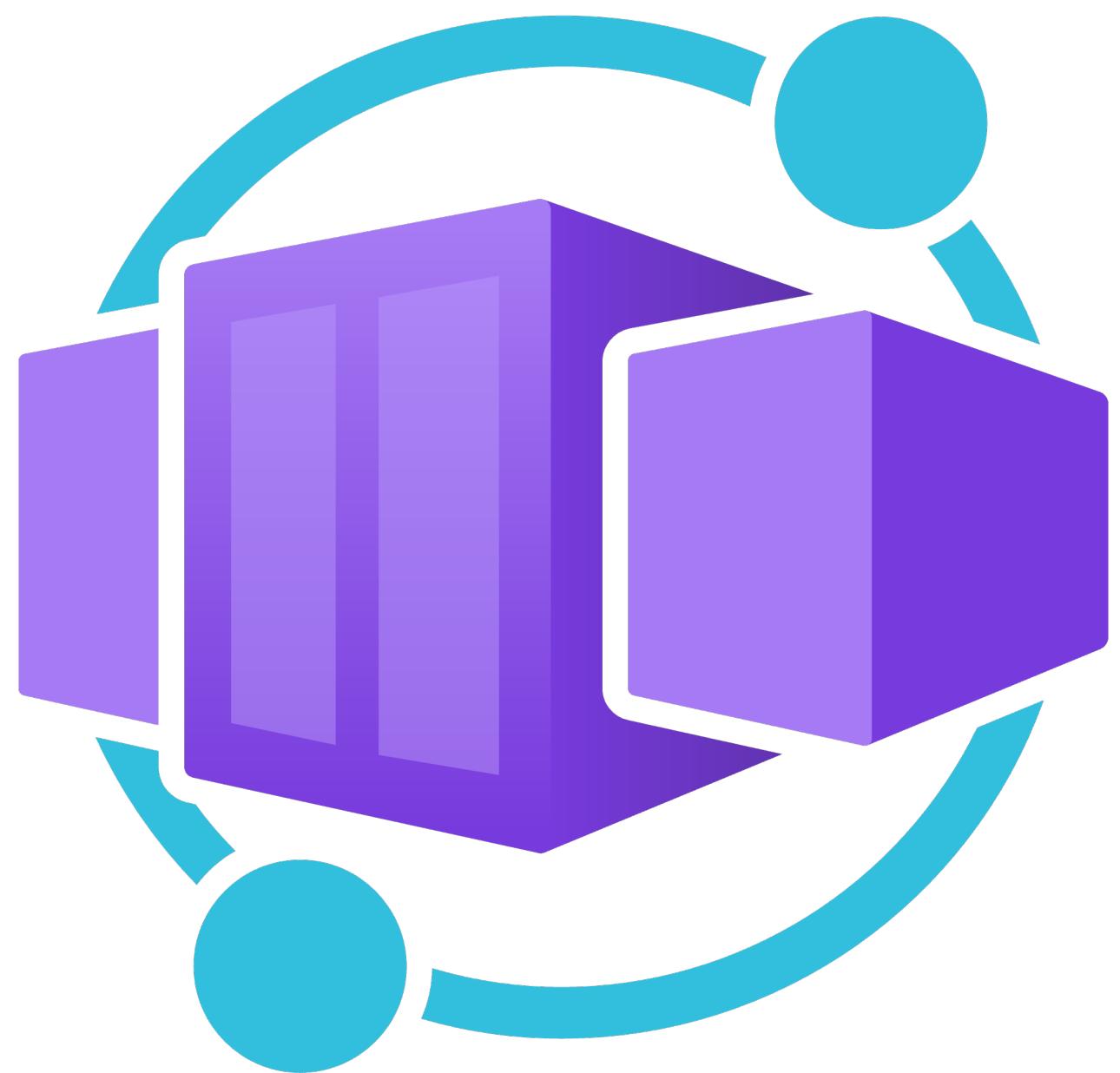


Azure Container Apps

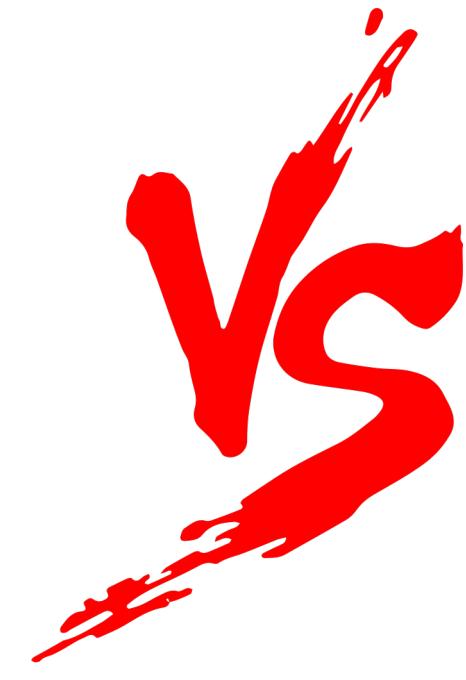


Azure App Service

# Azure Container Apps - Alternatives

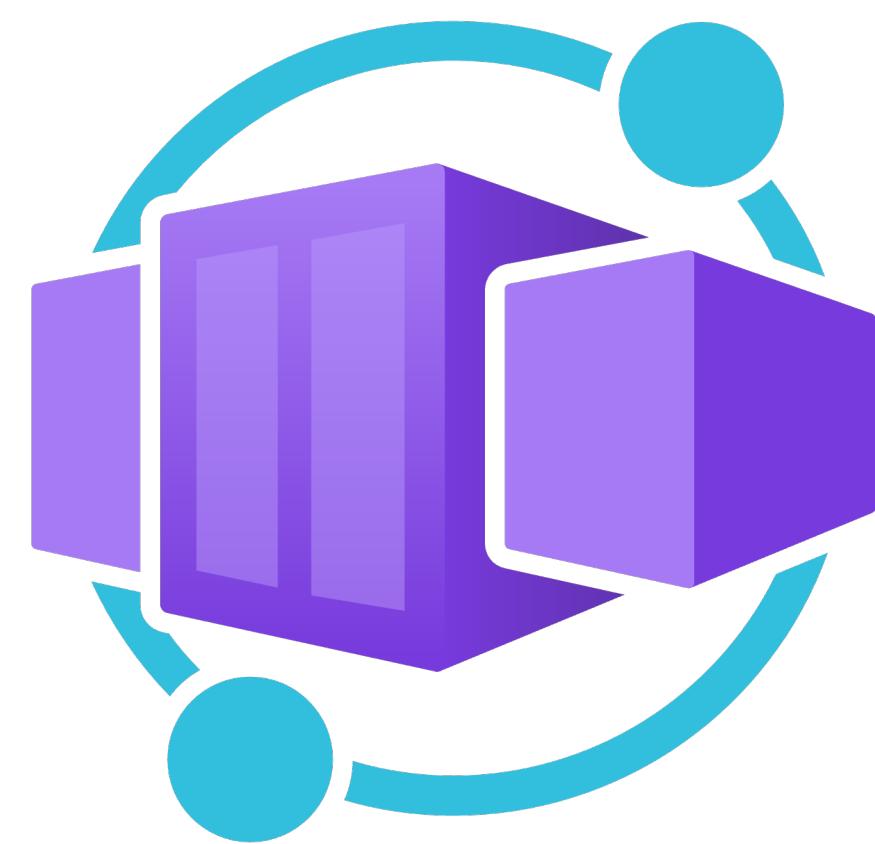


Azure Container Apps

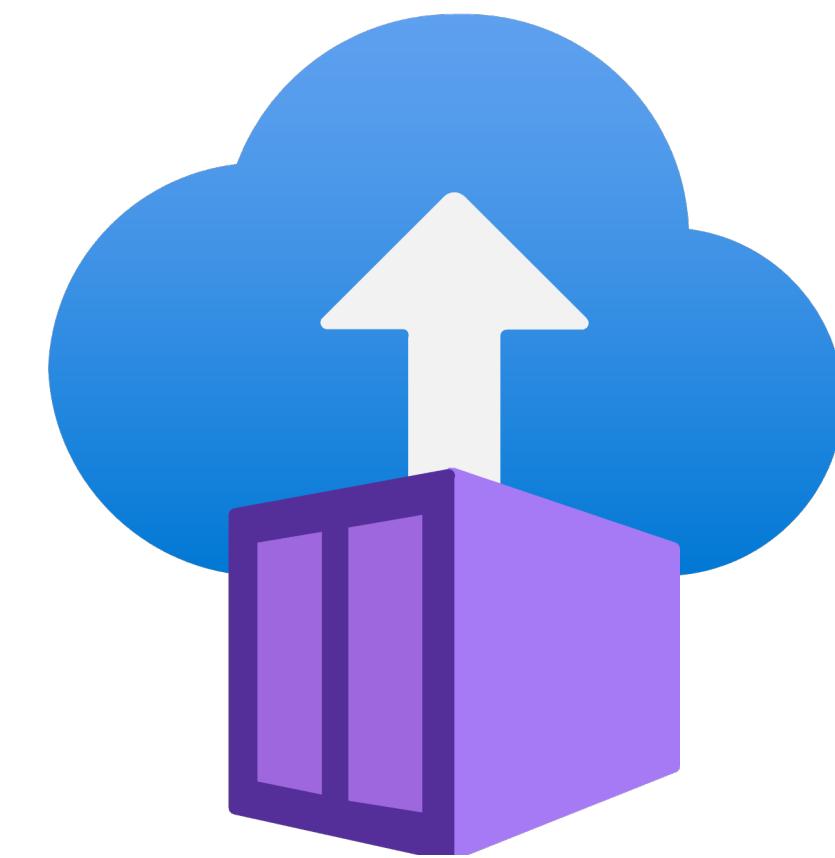


Azure Functions

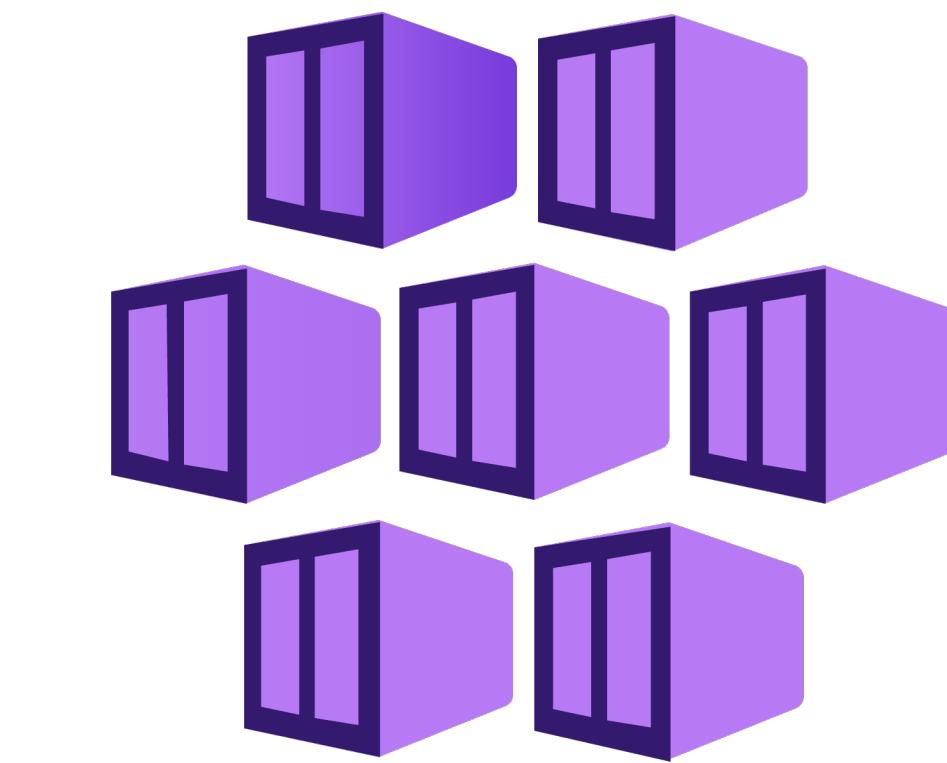
# Azure Container Apps - Alternatives



VS



VS

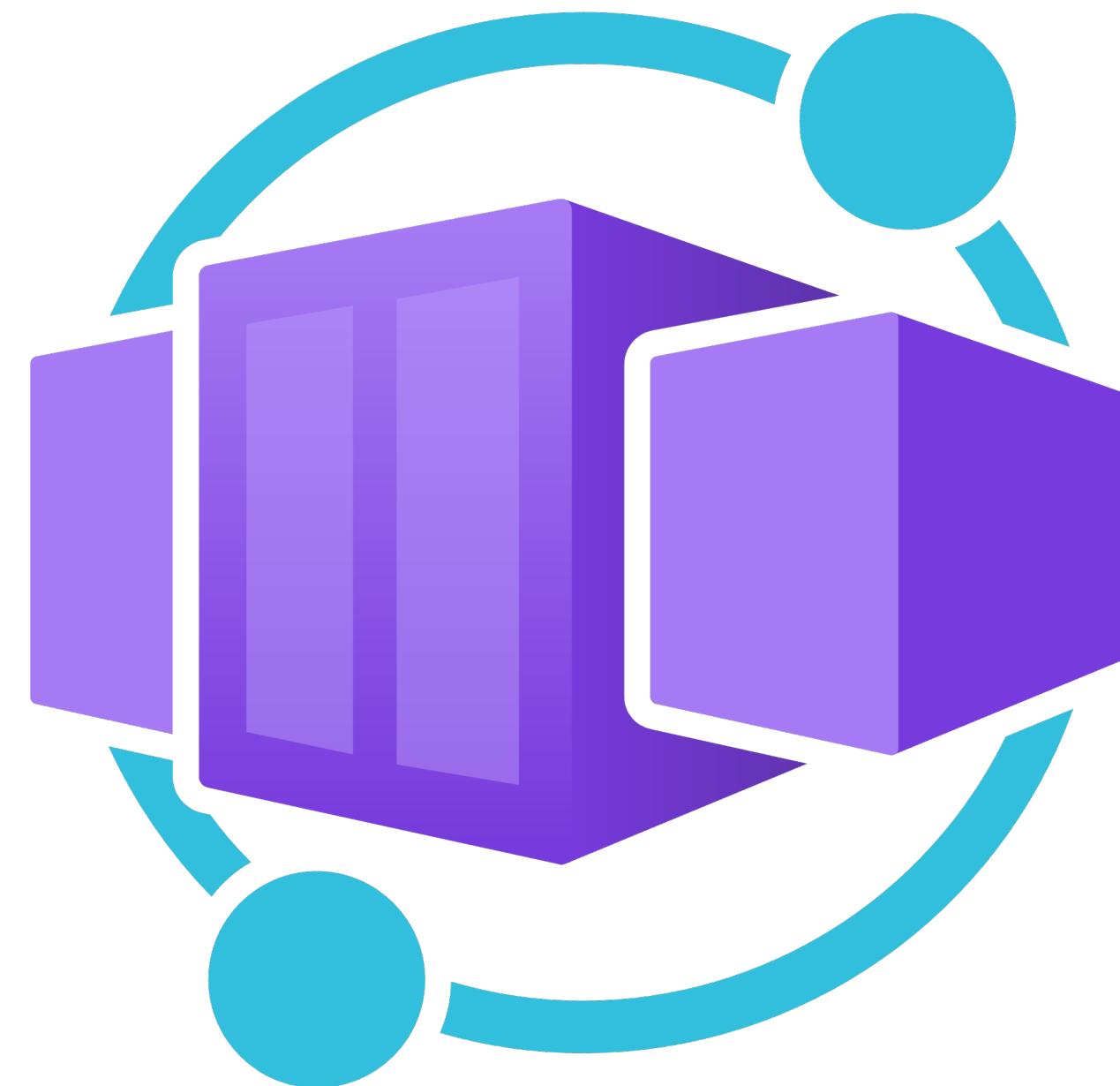


Azure Container Apps

Azure Container  
instances

Azure Kubernetes  
services

# Azure Container Apps - Documentation



<https://docs.microsoft.com/en-us/azure/container-apps/>

# Thank you for your attention.



Jiří Landsman  
Architect & Senior  
Developer