

**SoftwareOne
Cloud Experience Webinar:
Application
Modernization
tramite Container**

**AKS, ACA e ARO:
Kubernetes e OpenShift in
salsa Azure**



L'altra volta abbiamo parlato di App Services e altri servizi PaaS: come si integrano nel mondo dei container a 360°?



**Molti Clienti
hanno già iniziato
a lavorare su
container on-
premises: come
aggiungere valore
con il Cloud
Azure?**



Per i Clienti che hanno già modernizzato con container su Azure, quali i vantaggi e le differenze tra le varie soluzioni?

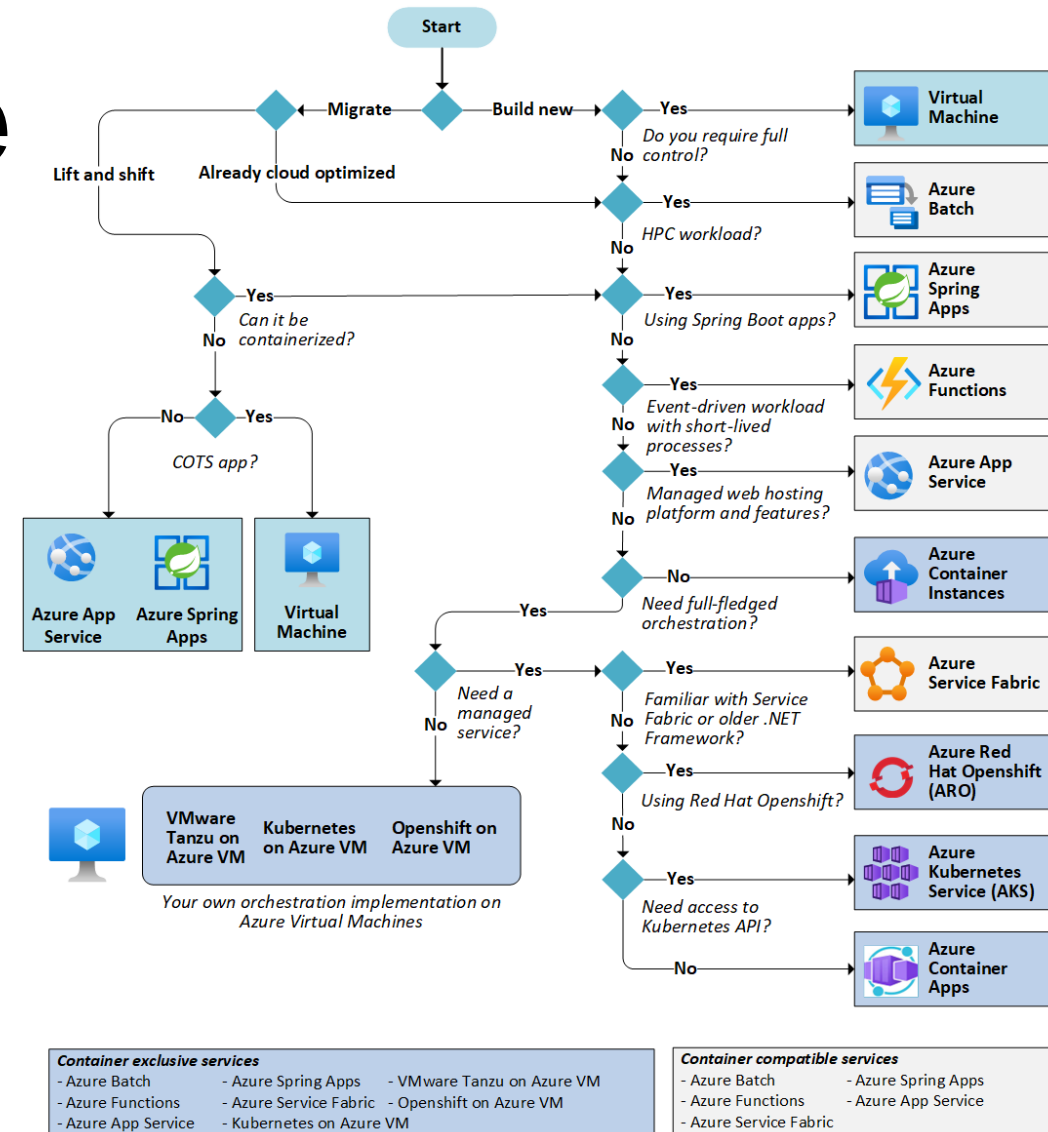


Azure ❤️ Containers

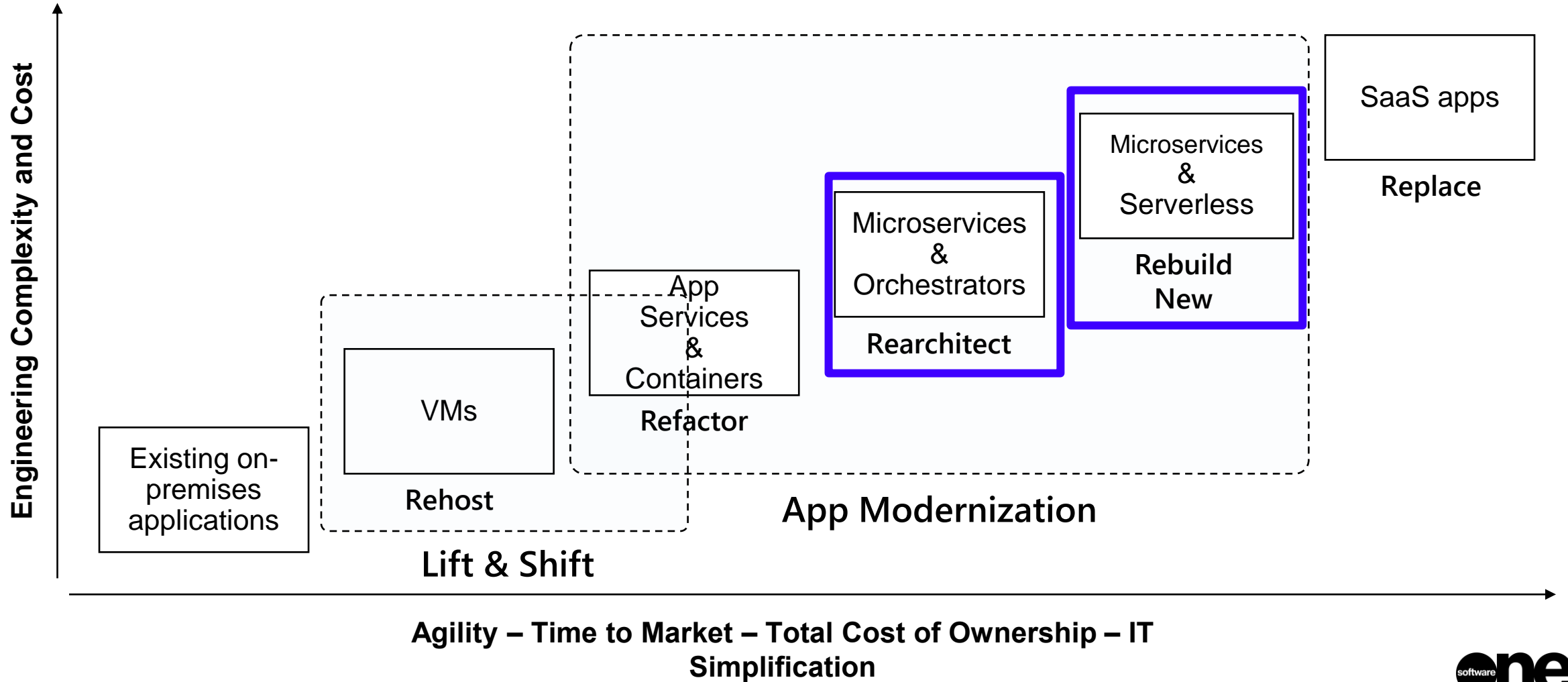


The Paradox of Choice

<https://learn.microsoft.com/en-us/azure/architecture/guide/technology-choices/compute-decision-tree>



(Hybrid) Cloud app continuum

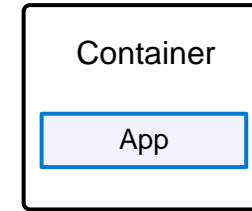


What is a **container**?



Virtual machines

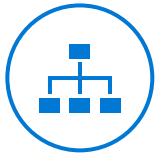
Virtualize the hardware
VMs as units of scaling



Containers

Virtualize the operating system
Applications as units of scaling

What are **microservices**?



A Software Architectural Style

Applications are composed of small, independent modules that communicate with each other using well-defined APIs. Not platform specific.



Decoupled

These service modules are highly decoupled building blocks that are small enough to implement a single functionality but together can form larger systems



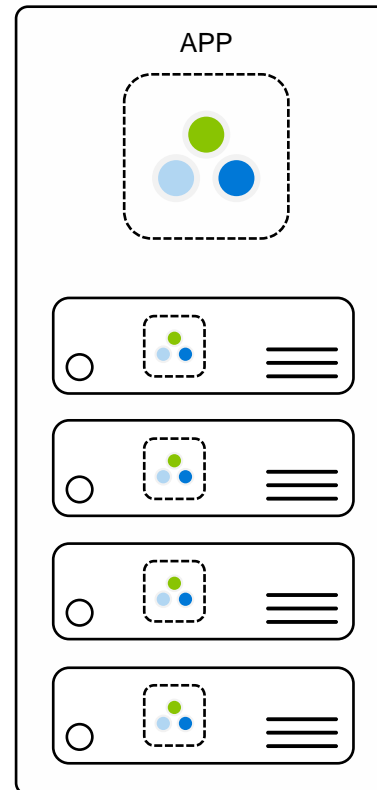
Independently versioned, deployed & scaled

With a microservices architecture, developers can create, manage and improve application services independently, even using different languages

Containers provide the consistent format and isolation desired by microservices.

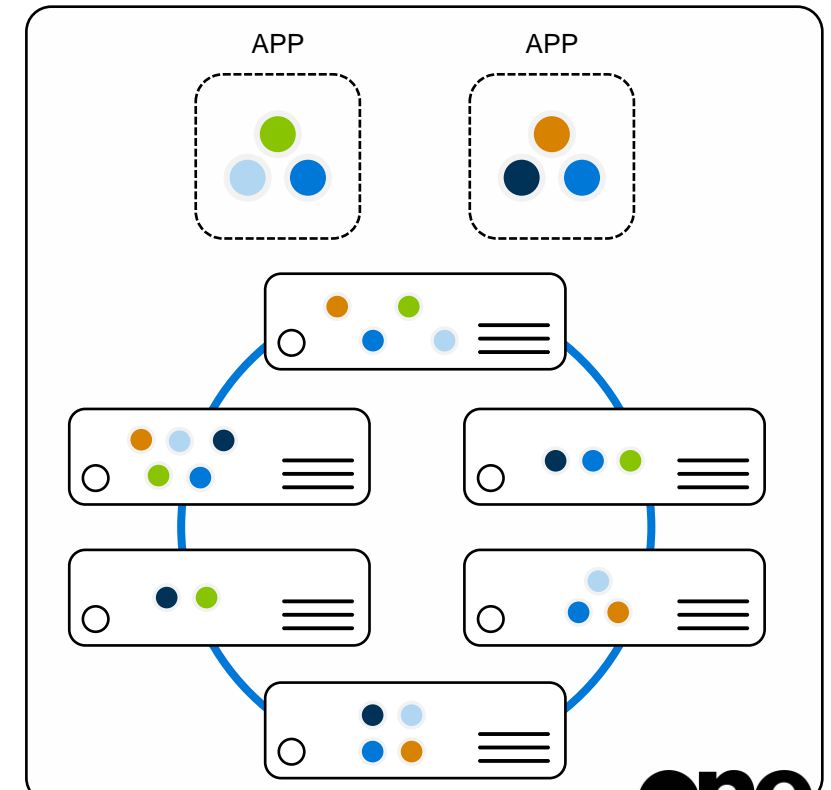
Monolithic

Large, all-inclusive app



Microservices

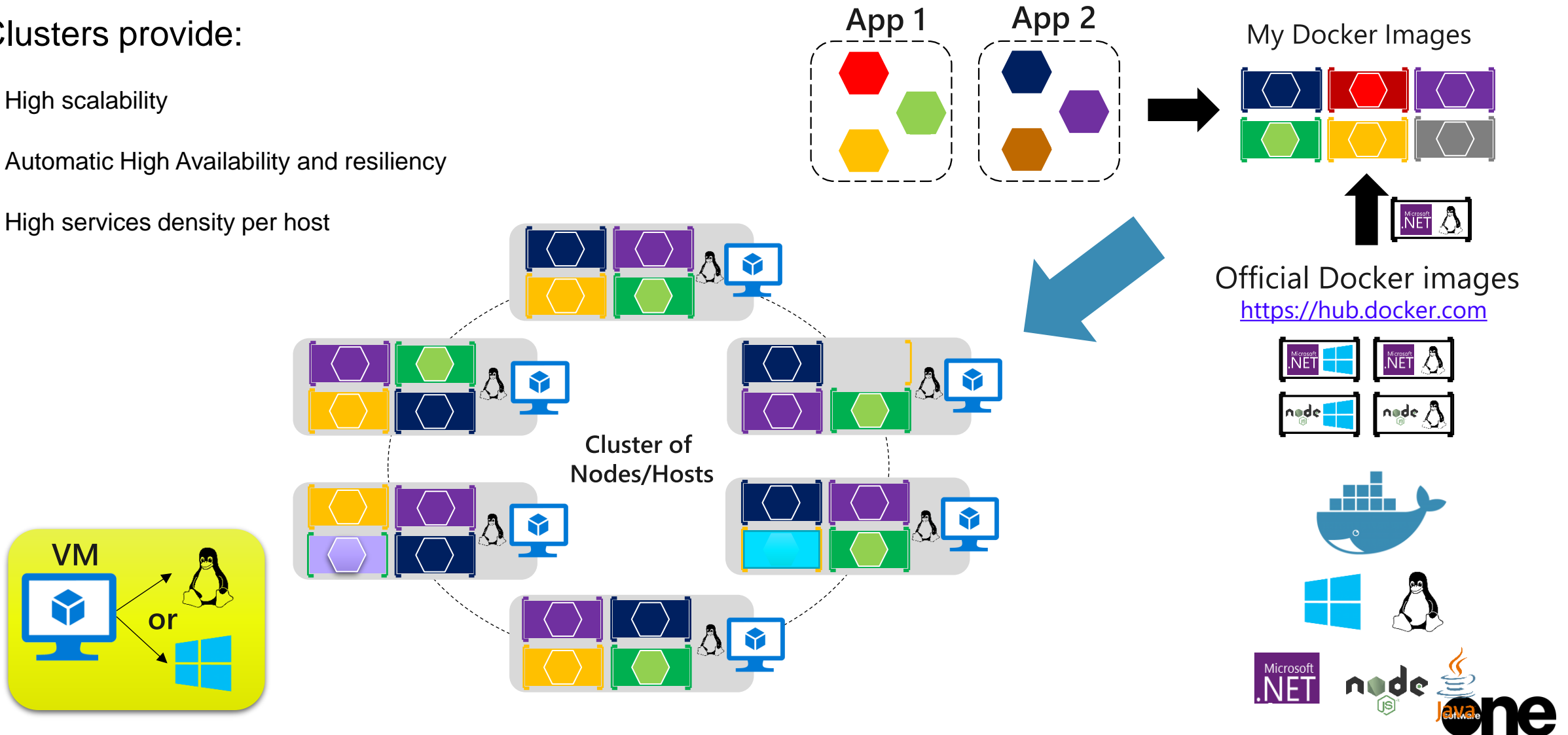
Small, independent services



Orchestrator's cluster managing microservices/containers

Clusters provide:

- High scalability
- Automatic High Availability and resiliency
- High services density per host



Containers in Azure



App Service

Deploy web apps or APIs using containers in a PaaS environment

DEPRECATED



Service Fabric

Modernize **.NET Framework** applications to microservices



Kubernetes Service

Scale and orchestrate containers using Kubernetes



Container Instance

Elastically burst from your Azure Kubernetes Service (AKS) cluster



Container Apps

Host managed microservices and containerized applications on a serverless platform.



Azure Container Registry



Docker Hub



GitHub Packages

Other registries



Ecosystem

Bring your Partner solutions that run great on Azure

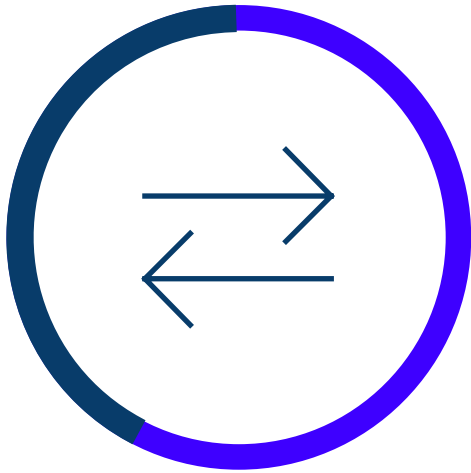


Kubernetes



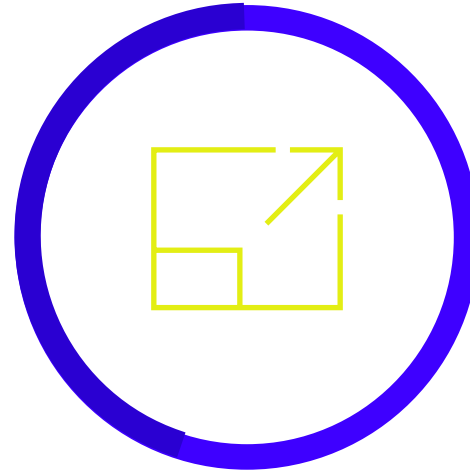
What's behind Kubernetes' growth?

Kubernetes: the leading orchestrator shaping the future app development and management



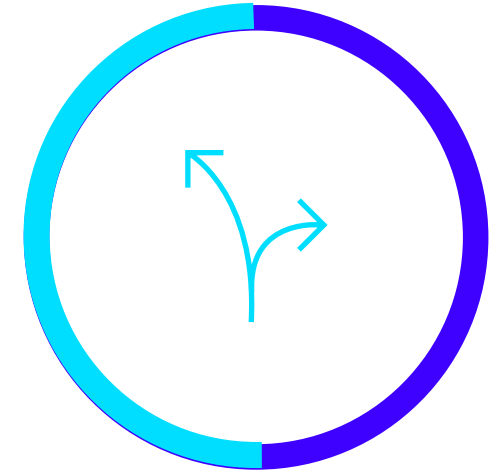
42%

portability



45%

scalability



50%

agility

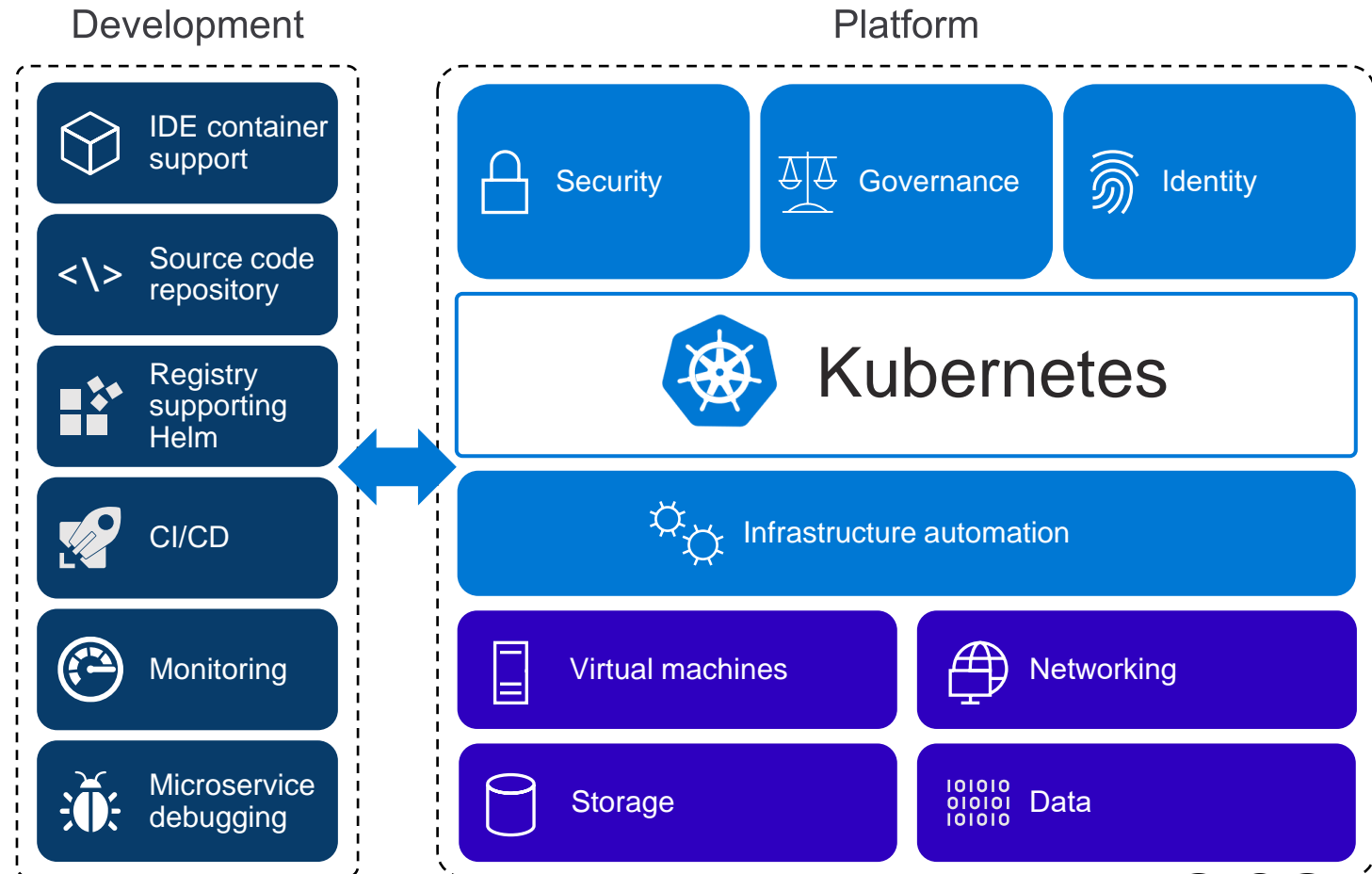
The perceived benefits of Kubernetes

Kubernetes on its own is not enough

Save time from infrastructure management and roll out updates faster without compromising security

Unlock the agility for containerized applications using:

- **Infrastructure automation** that simplifies provisioning, patching, and upgrading
- Tools for **containerized app development and CI/CD workflows**
- Services that support **security, governance, and identity and access management**



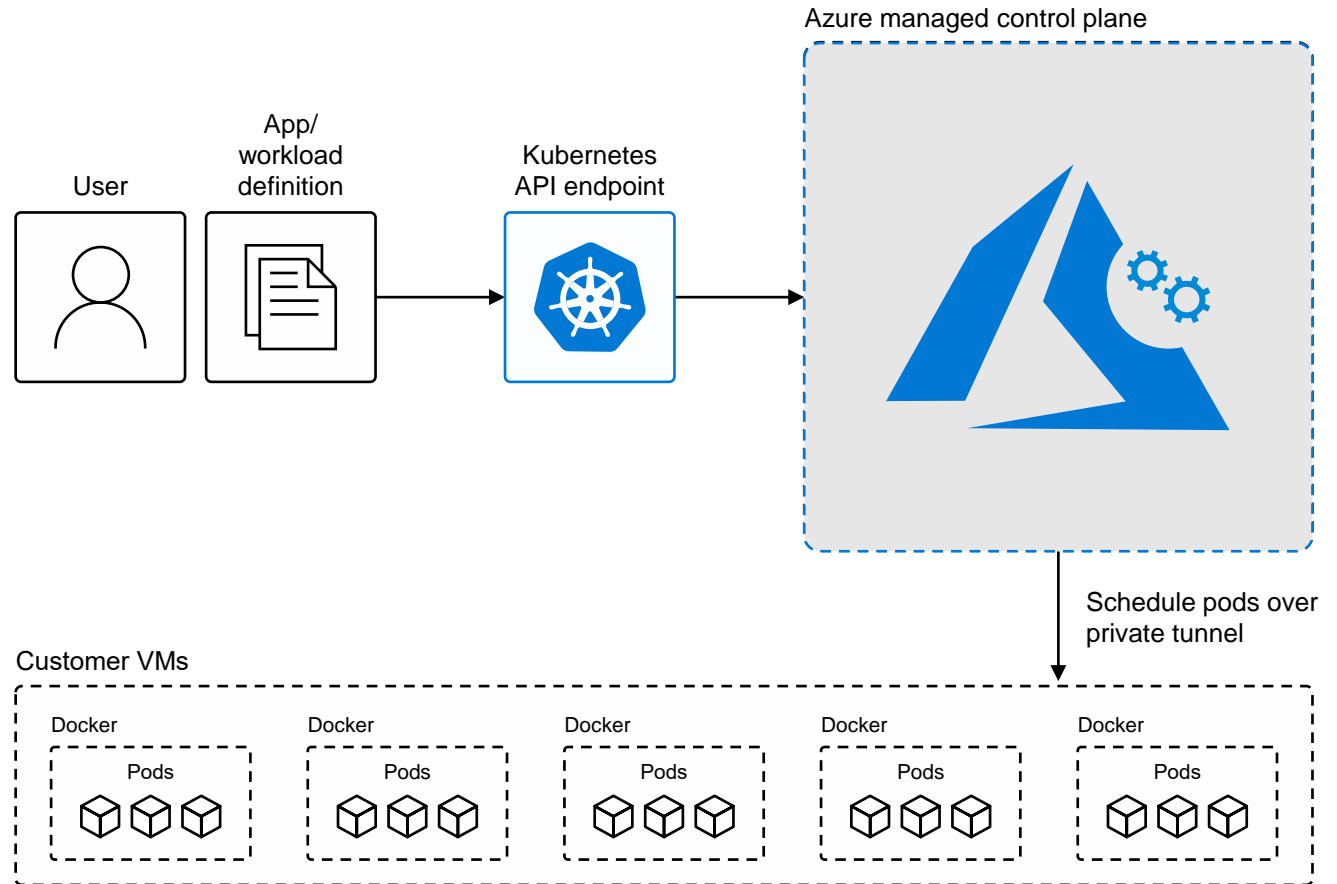
Kubernetes on Azure



Manage Kubernetes with ease

Infrastructure automation

- Automated provisioning, upgrades, patches
- High reliability, availability
- Easy, secure cluster scaling
- Self-healing
- API server monitoring
- At no charge (you don't pay the managed control plane)

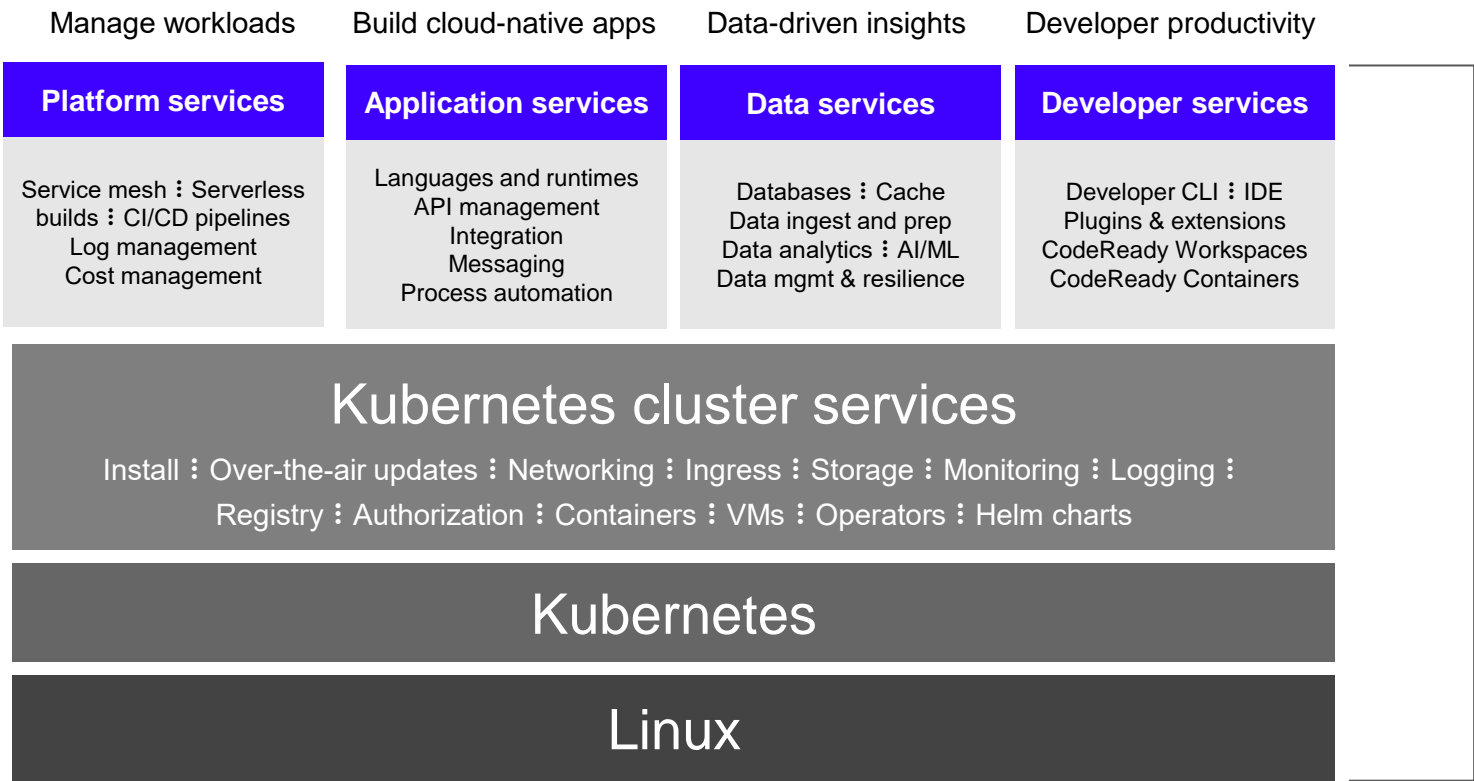


RedHat OpenShift on Azure



The value of Red Hat OpenShift

A complete application platform



Automated, full-stack installation from the container host to application services

Seamless Kubernetes deployment to any cloud or on-premises environment

Configurable autoscaling of cloud resources

Automated updates for cluster software



What is Azure Red Hat OpenShift?

- Focus on building and scaling applications while we manage the rest.



Highly available, fully managed clusters on-demand, built on industry-leading Red Hat OpenShift Container Platform, and managed on a leading public cloud, Microsoft Azure.



Jointly monitored and operated by Microsoft and Red Hat with an integrated support experience.



Turnkey application development platform, with integrations into Azure ecosystem



Enterprise-grade operations, security and compliance



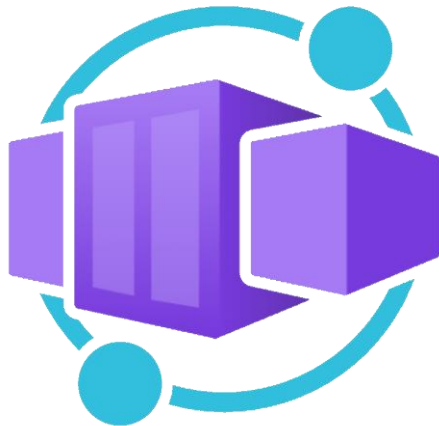
Backed by the experience of global site reliability expert (SRE) teams.

**"No one has
ever been
fired for
choosing AKS
or ARO..."**



Azure Container Apps

A new serverless container platform for building modern apps and microservices

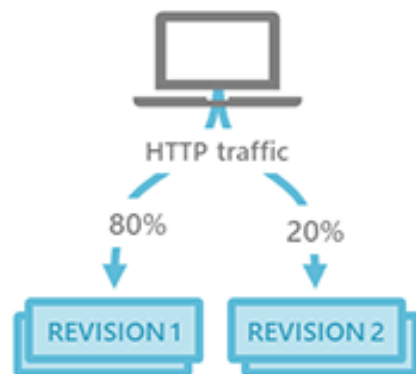


Built on a foundation of **AKS**, **KEDA**, **Dapr**, and **Envoy**



Azure Container Apps: Example scenarios

PUBLIC API ENDPOINTS



HTTP requests are split between two versions of the container app where the first revision gets 80% of the traffic, while a new revision receives the remaining 20%.

BACKGROUND PROCESSING



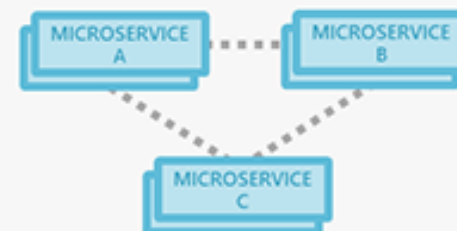
A continuously-running background process that transforms data in a database.

EVENT-DRIVEN PROCESSING



A queue reader application that processes messages as they arrive in a queue.

MICROSERVICES



Deploy and manage a microservices architecture with the option to integrate with Dapr.

AUTO-SCALE CRITERIA

Scaling is determined by the number of concurrent HTTP requests.

AUTO-SCALE CRITERIA

Scaling is determined by the level of CPU or memory load.

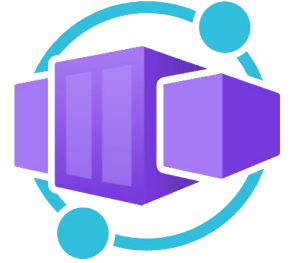
AUTO-SCALE CRITERIA

Scaling is determined by the number of messages in the queue.

AUTO-SCALE CRITERIA

Individual microservices can scale according to any KEDA scale triggers.

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

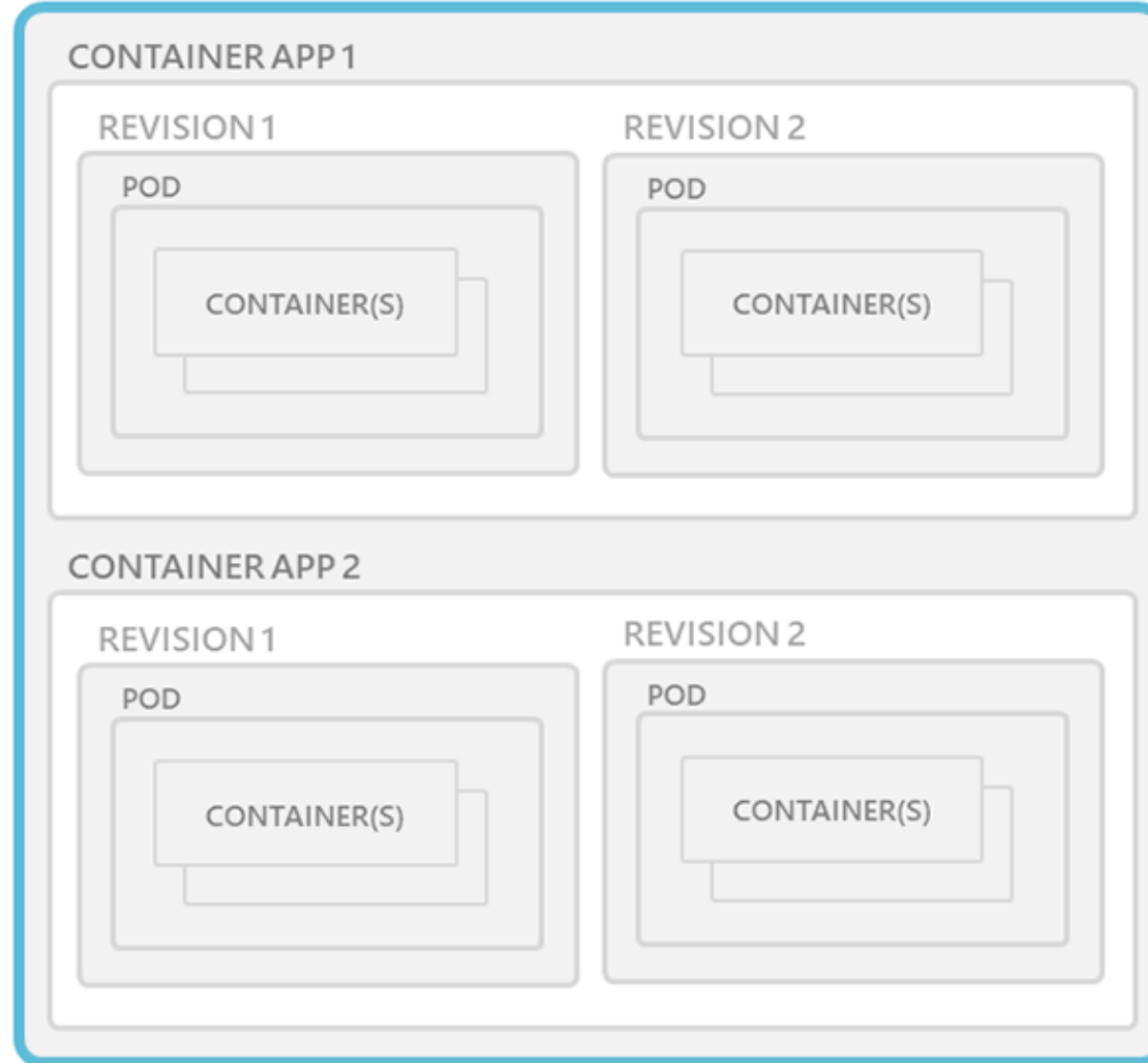
Automatic encryption for ingress and service-to-service communications

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



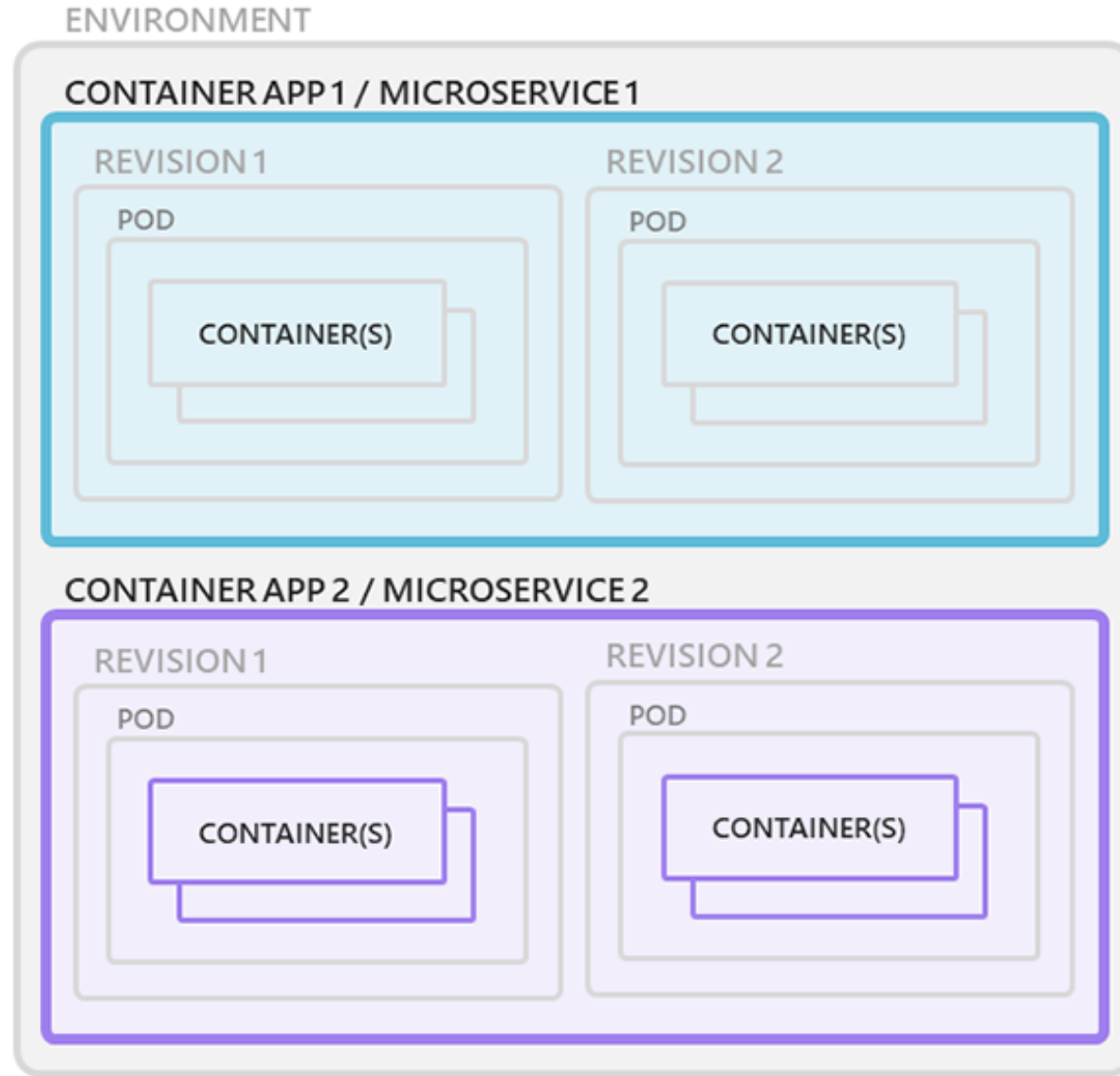
Environments are an isolation boundary around a collection of container apps.

ENVIRONMENT: OPTIONAL CUSTOM VIRTUAL NETWORK



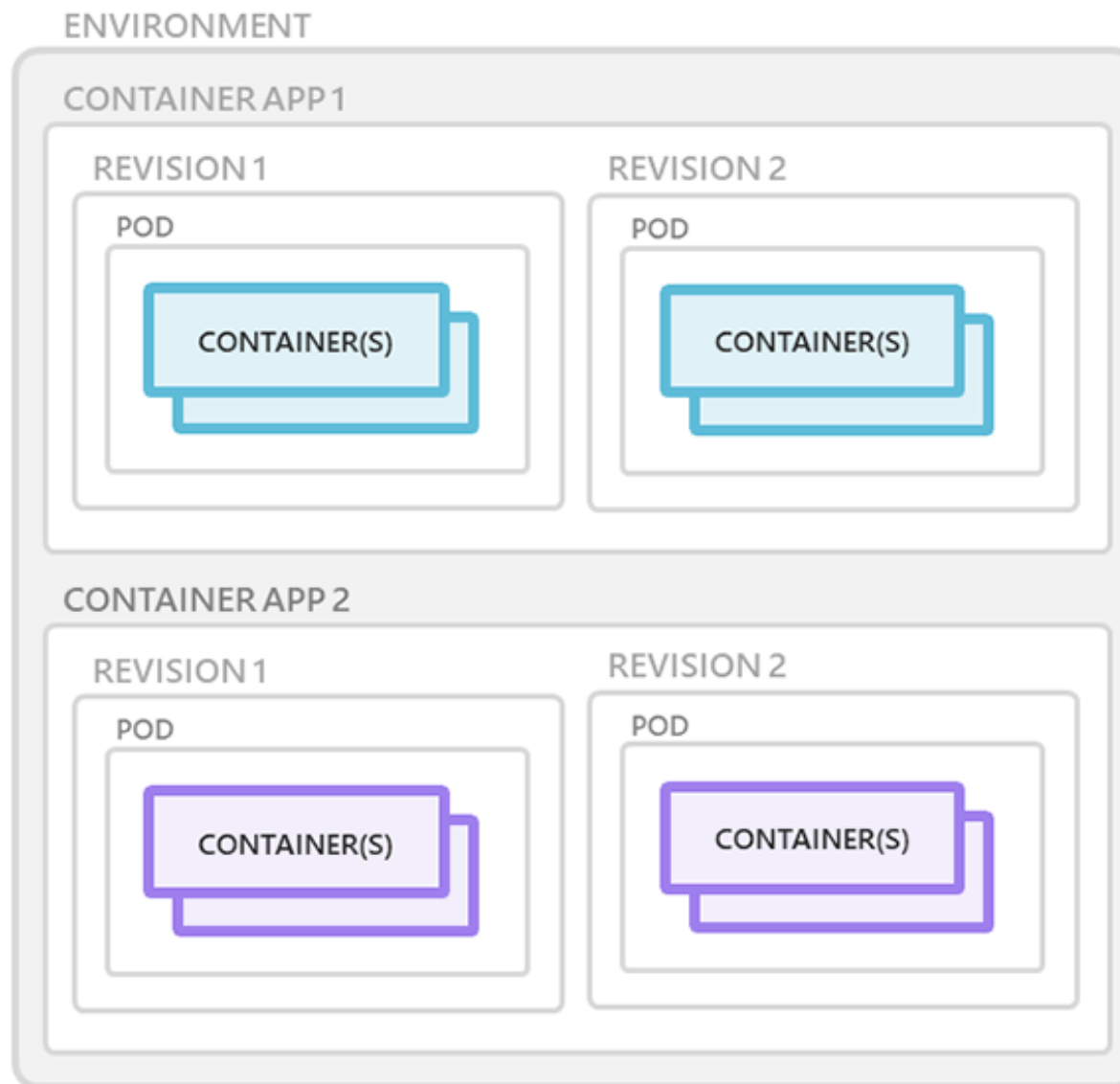


Container apps are
deployed as
microservices.



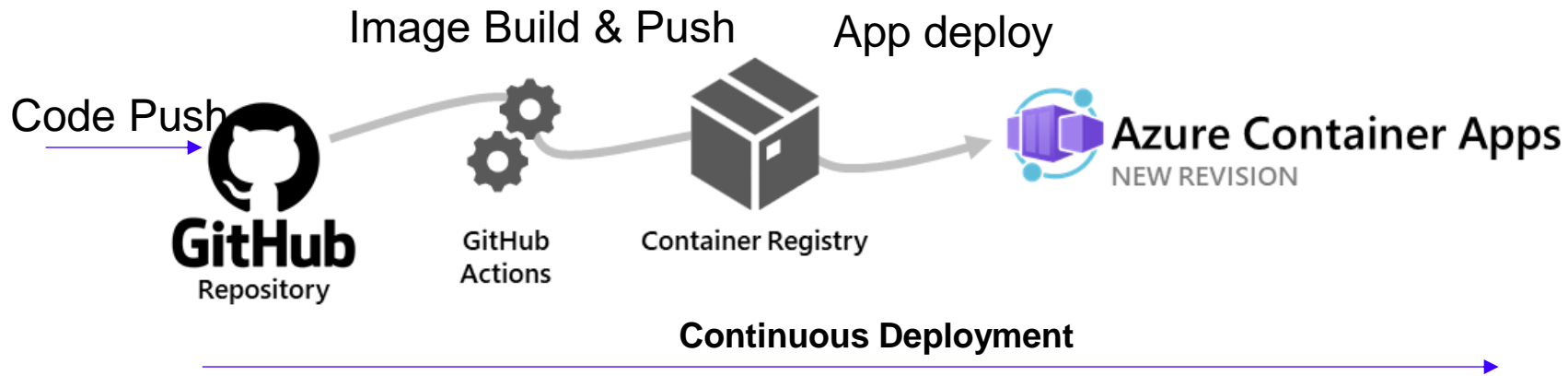
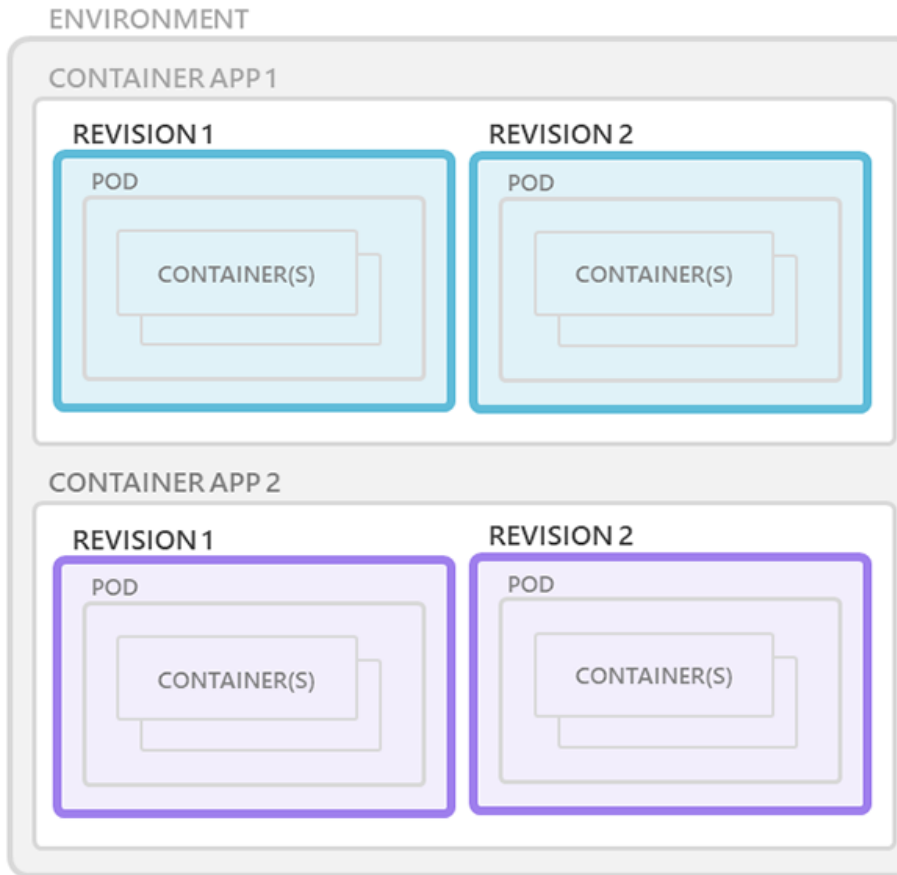


Containers for an Azure Container App are grouped together in pods inside revision snapshots.



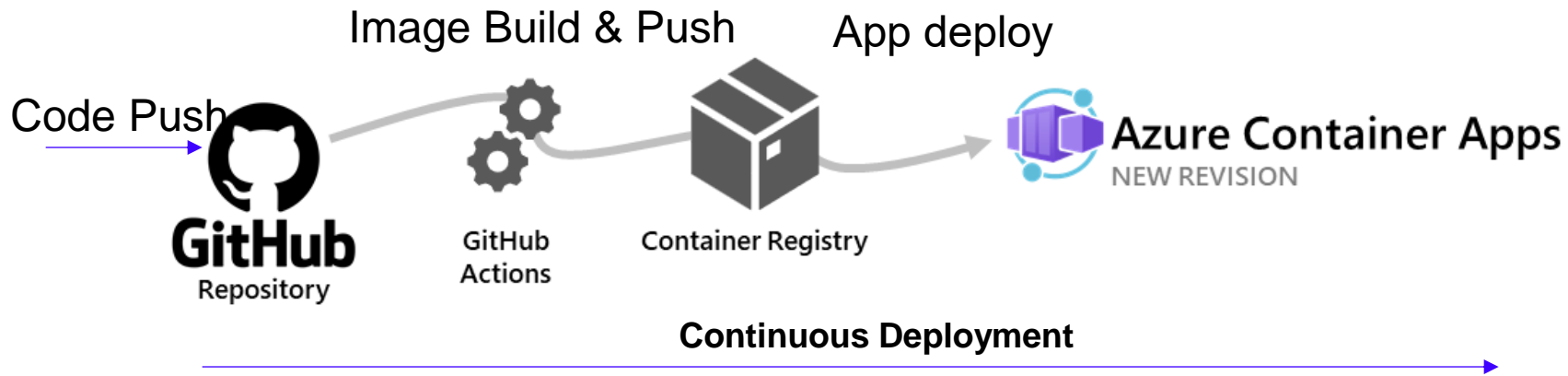
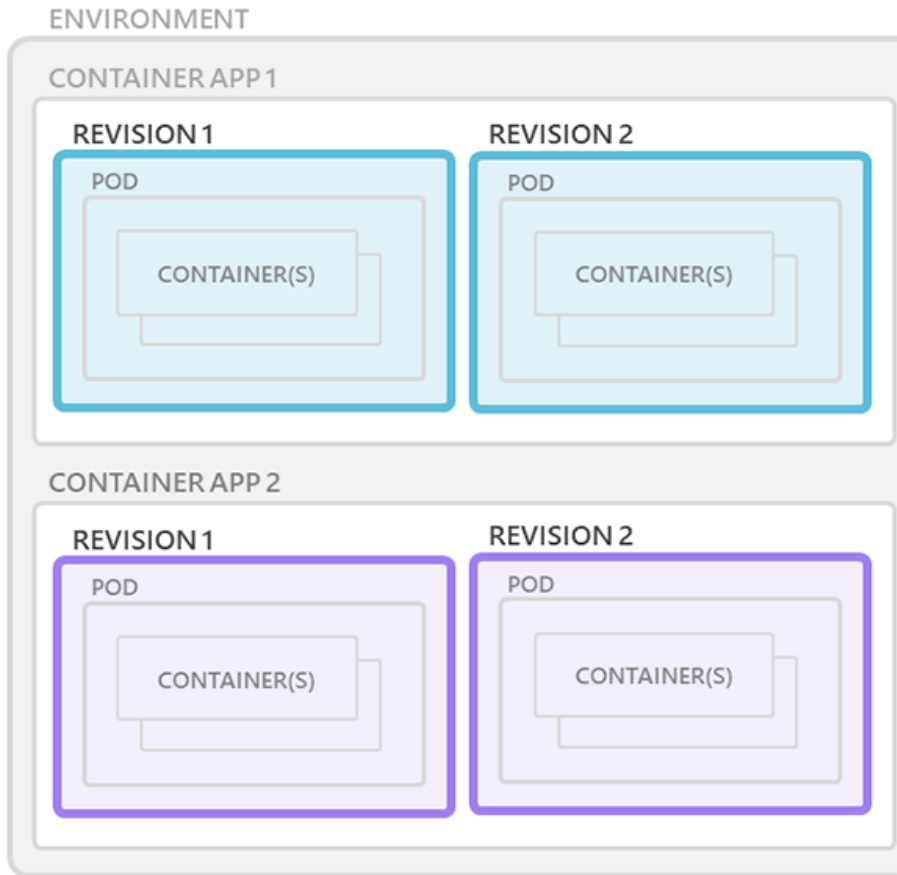


Revisions are immutable snapshots of a container app.





Revisions are immutable snapshots of a container app.





Once a revision is no longer needed, you can **deactivate** individual revisions, or choose to automatically deactivate old revisions.

Active Revisions

REVISION 2

POD

CONTAINER(S)

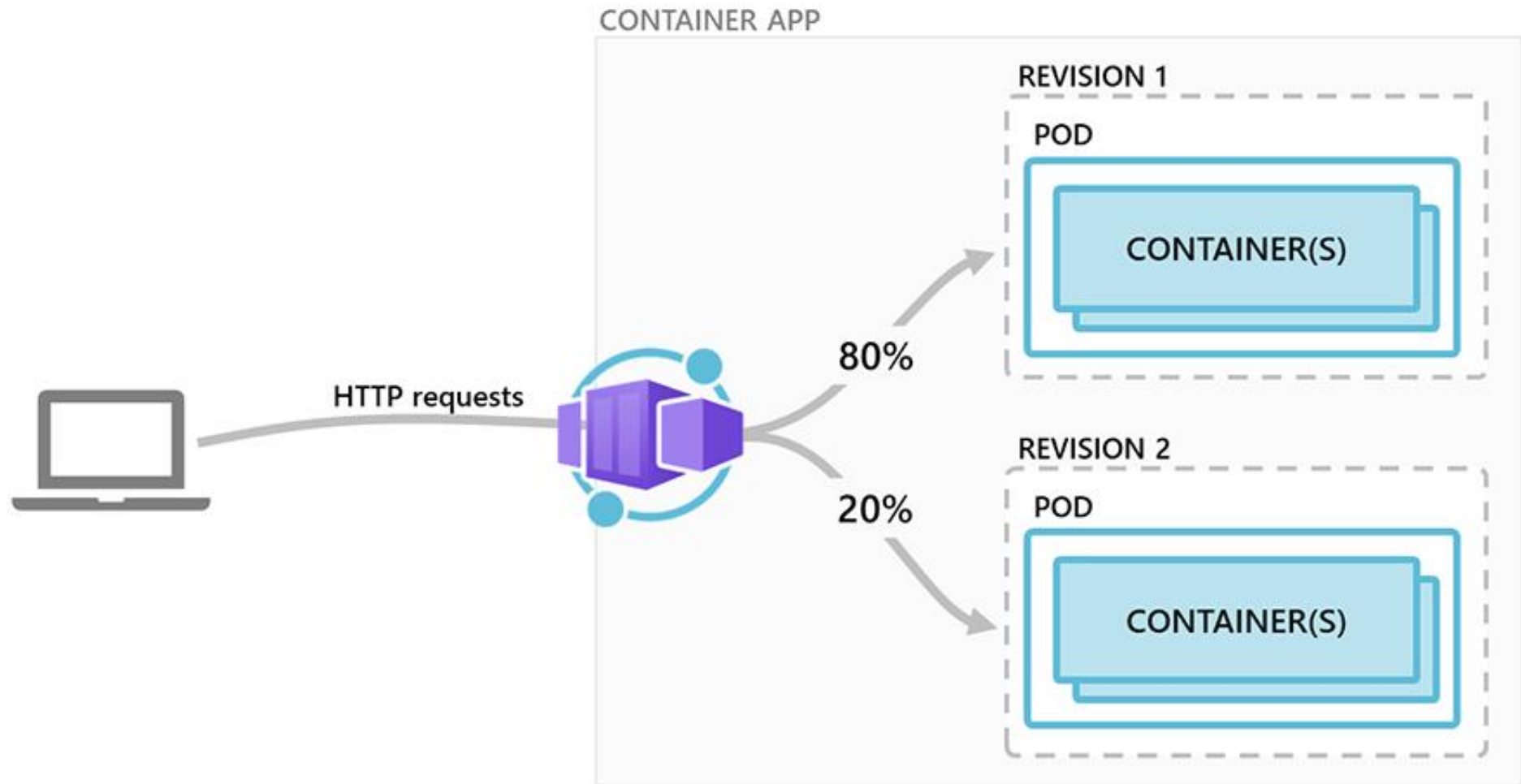
Inactive Revisions

REVISION 1

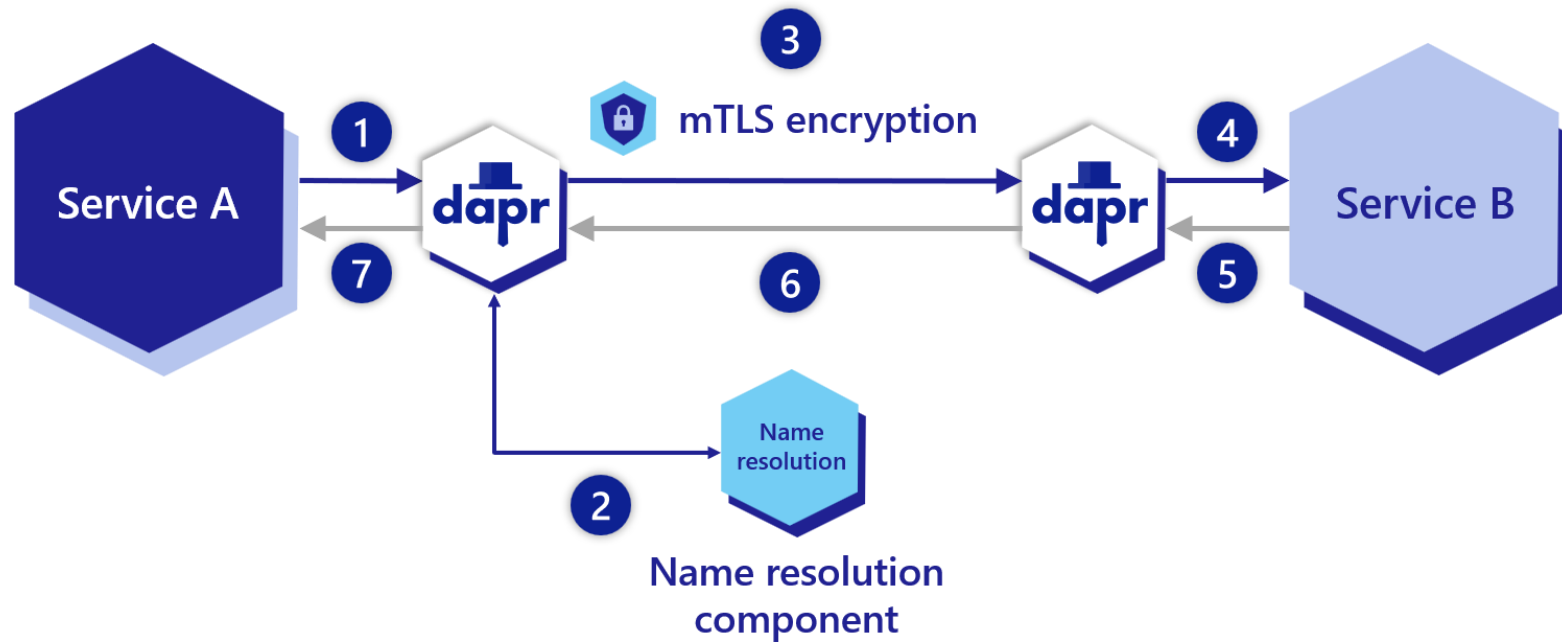
POD

CONTAINER(S)

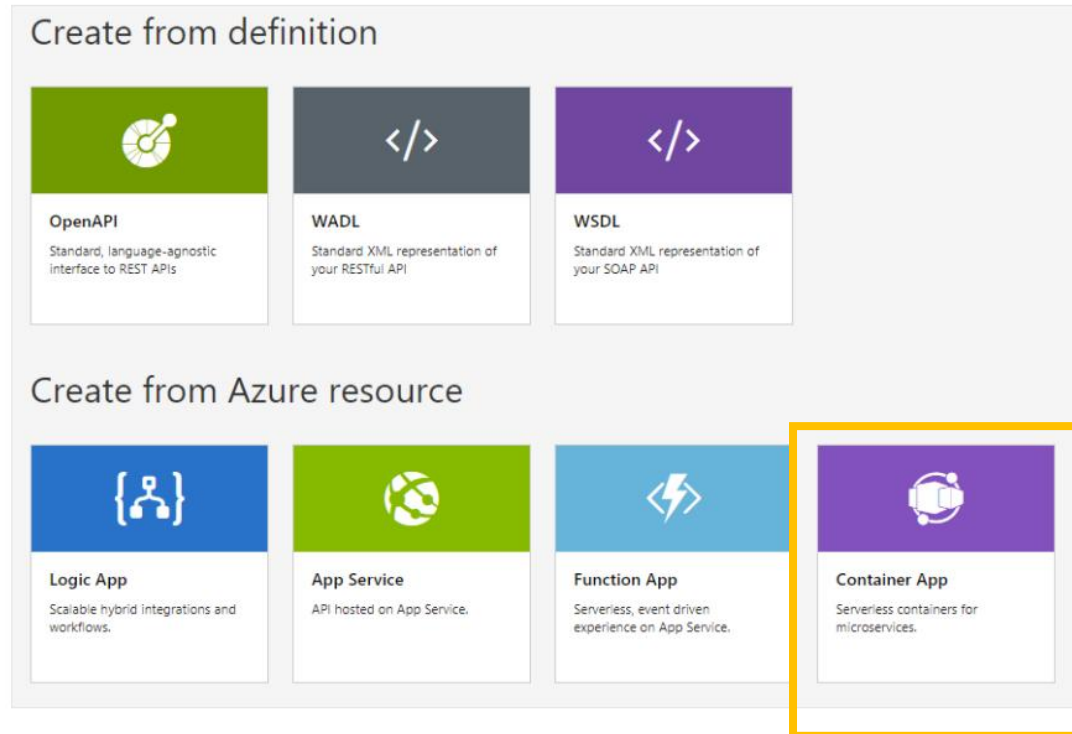
Ingress traffic splitting



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



API Management Import

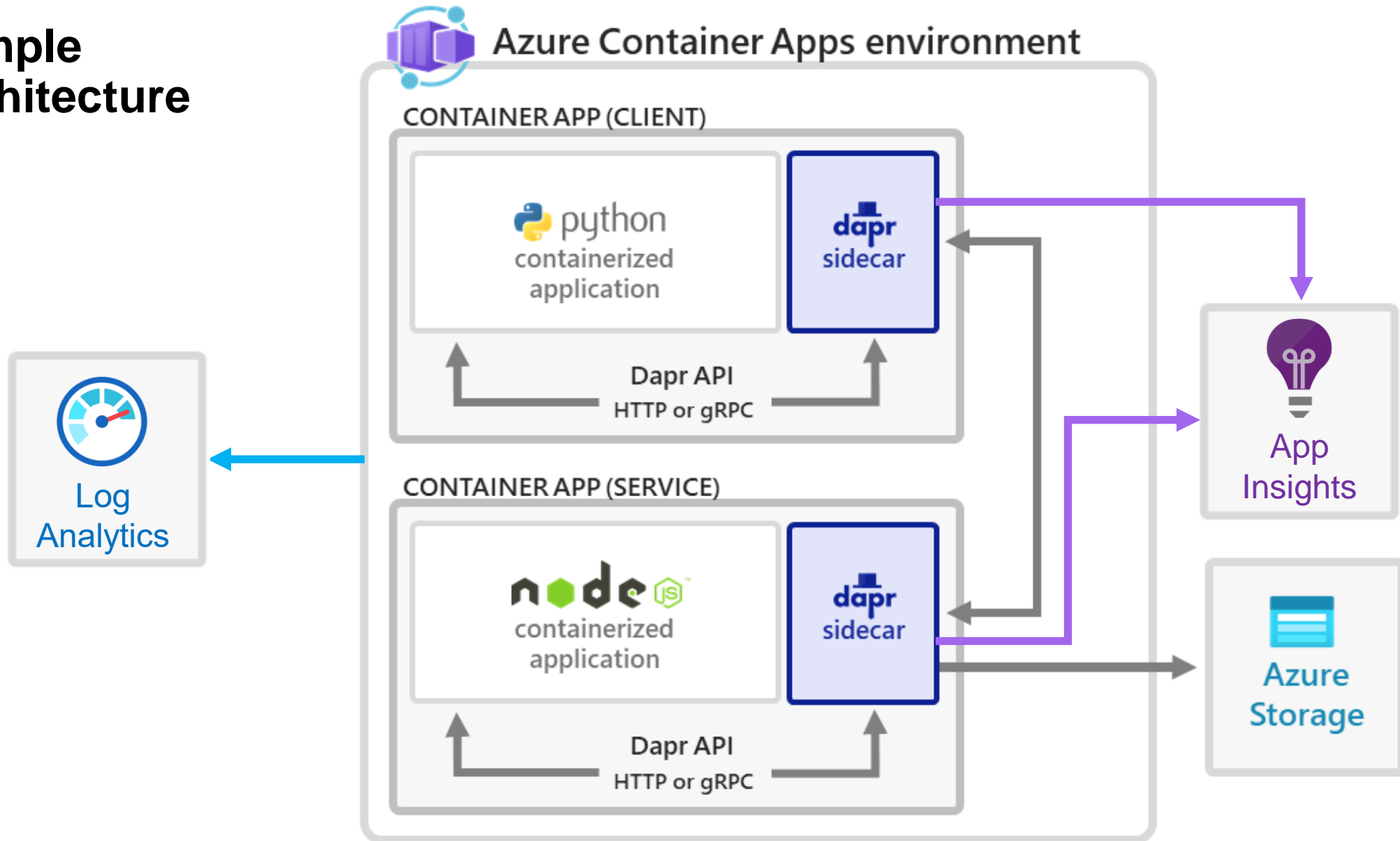


API Management will look in several locations for an OpenAPI Specification:

- 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>

Sample Architecture



<https://github.com/clarenceb/tutorial-dapr-cli>

Grazie!

Q&A

Contattateci:

davide.pizzanelli@softwareone.com

lorenzo.barbieri@softwareone.com

