

Containers on Azure an Overview

Patrick Gryczka
Cloud Solution Architect
April 2019

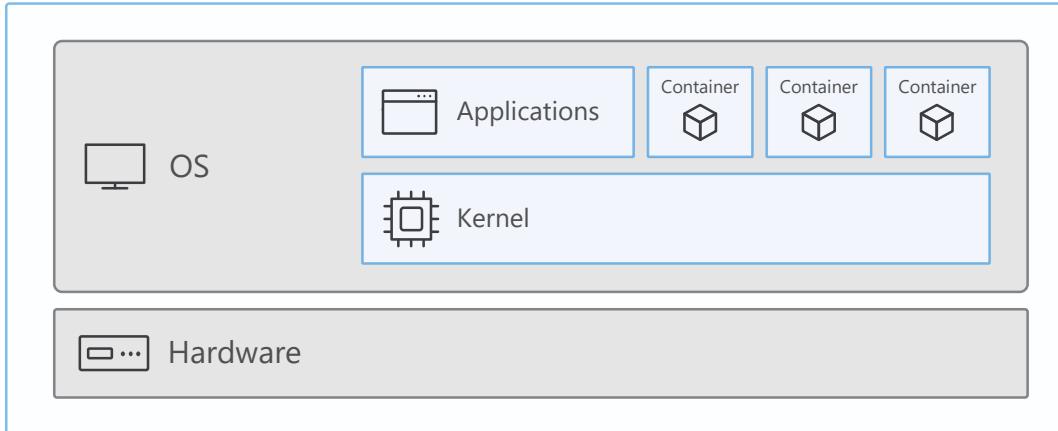
Contents

- [**## Container 101**](#)
- [**## Container Advantages**](#)
- [**## Orchestration**](#)
- [**## Containers on Azure**](#)
- [**## AKS**](#)
- [**## ACI**](#)
- [**## App Service**](#)
- [**## Service Fabric**](#)
- [**## Azure Batch**](#)
- [**## Azure Container Registry**](#)
- [**## AKS Roadmap**](#)

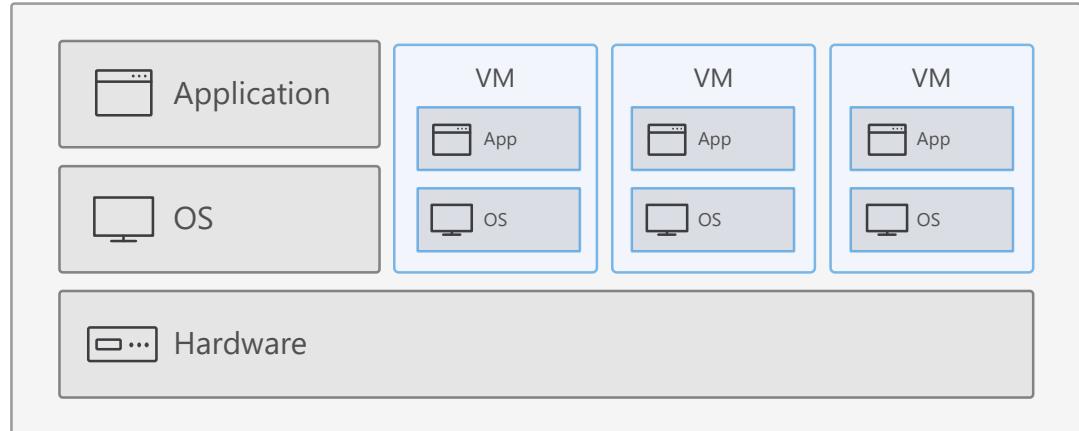
Container 101

What is a container?

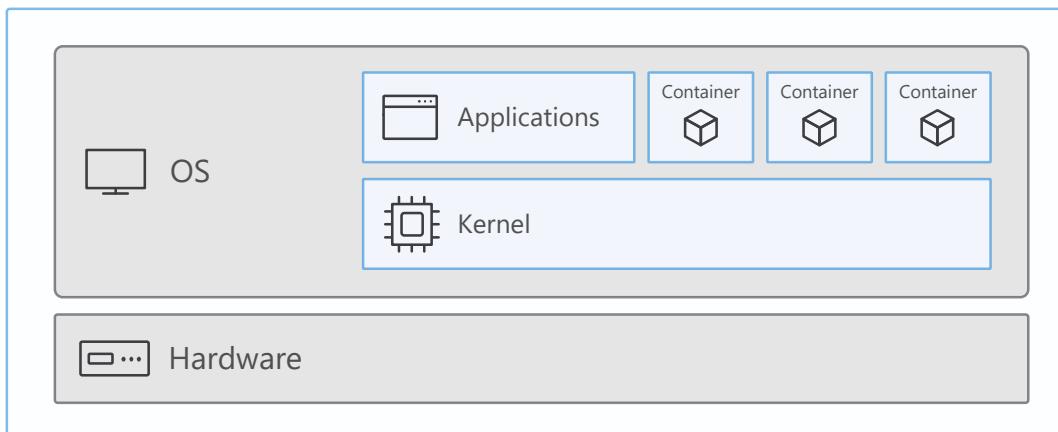
Containers = operating system virtualization



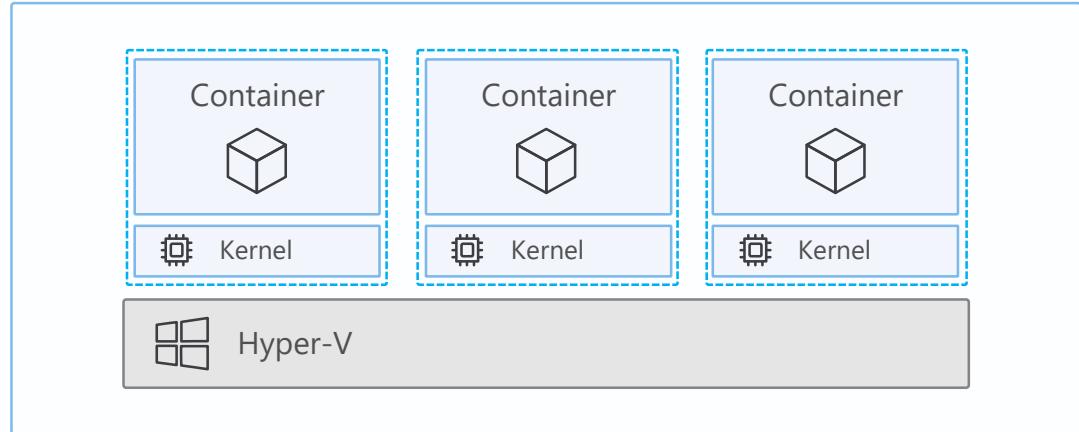
Traditional virtual machines = hardware virtualization



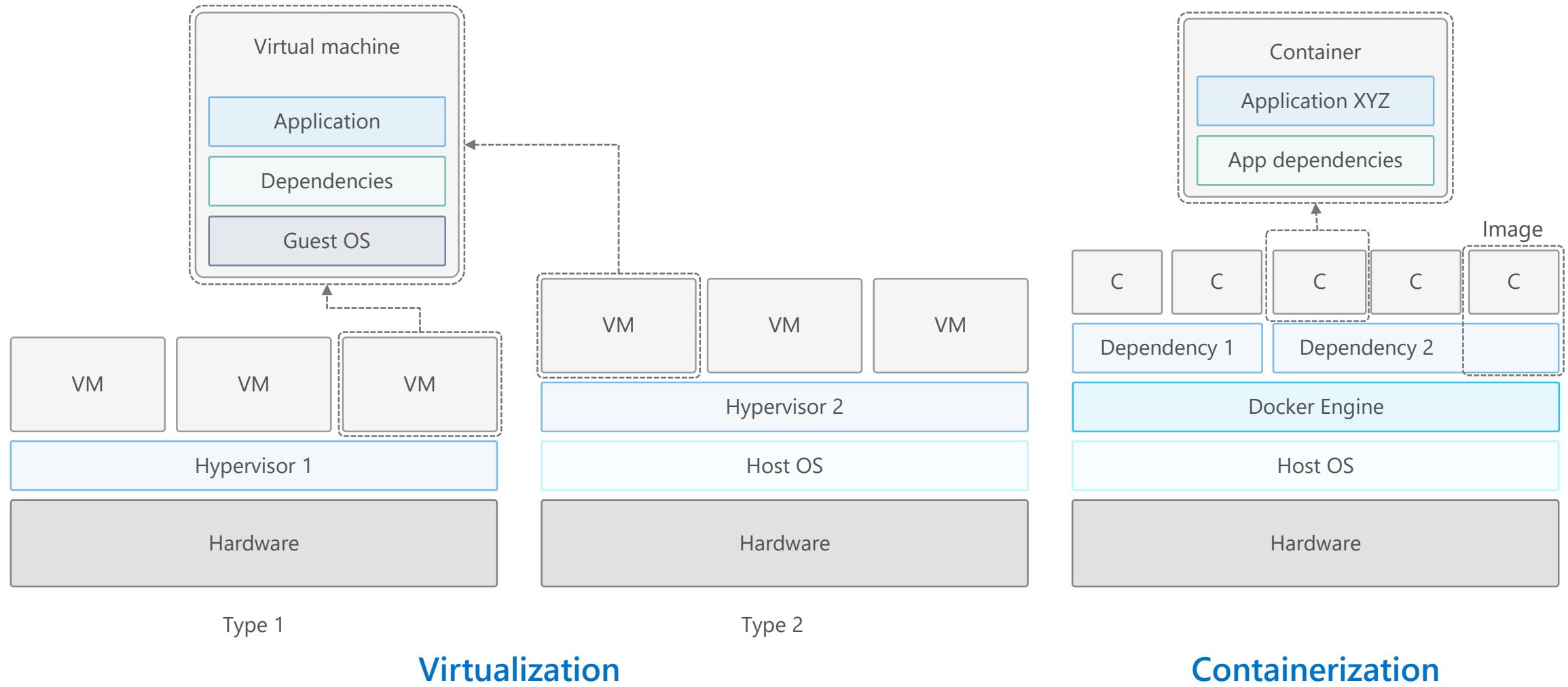
Windows Server containers: maximum speed and density



Hyper-V containers: isolation plus performance



Virtualization versus containerization?



What Advantages does this bring to our Processes?

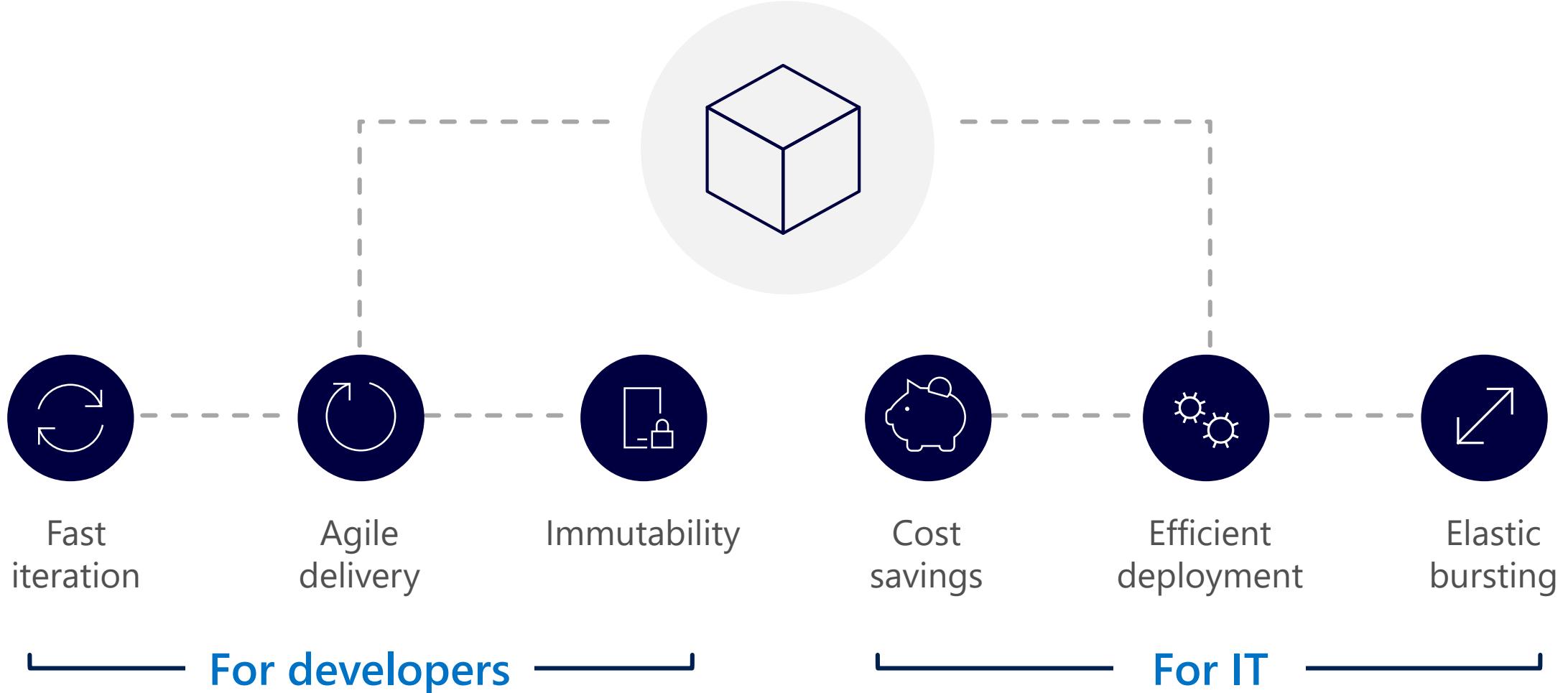
Can you Spot the differences?



MAIL: BEST CREDIT PHOTOS BY REX

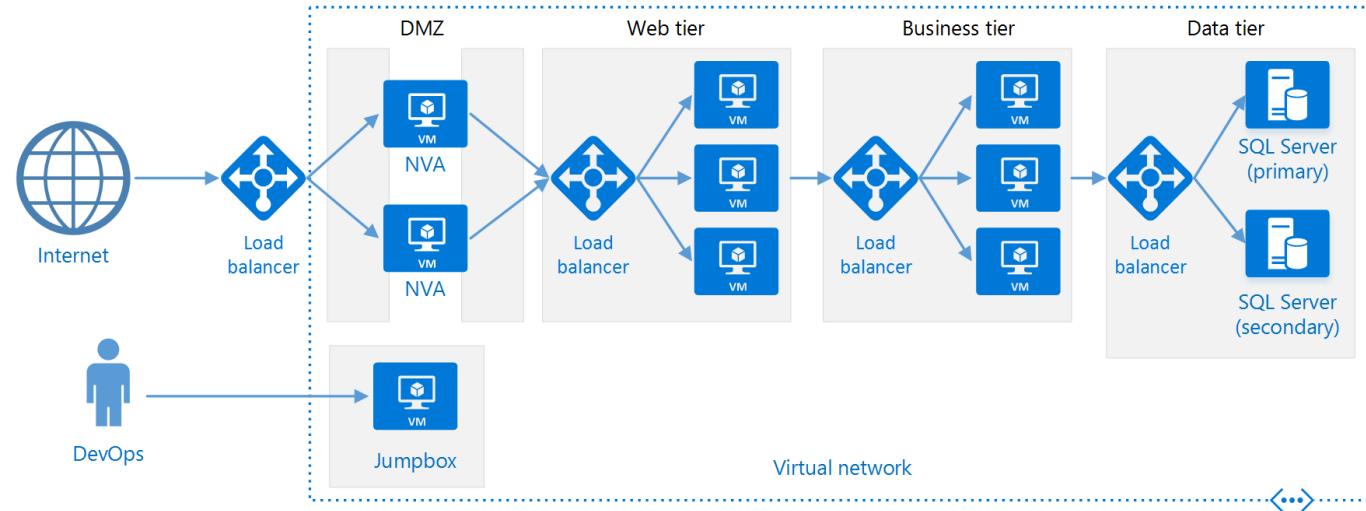


DevOps view of the container advantage

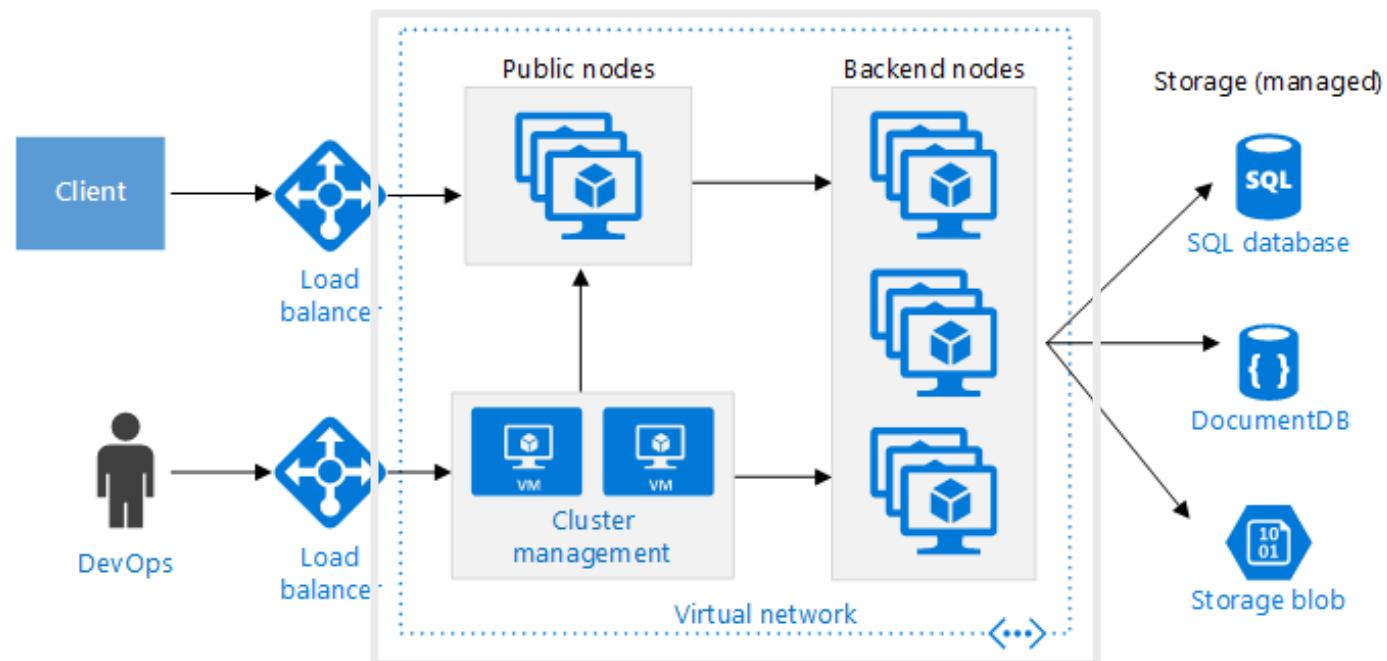


Impact in Software Architecture

N-Tier Architecture

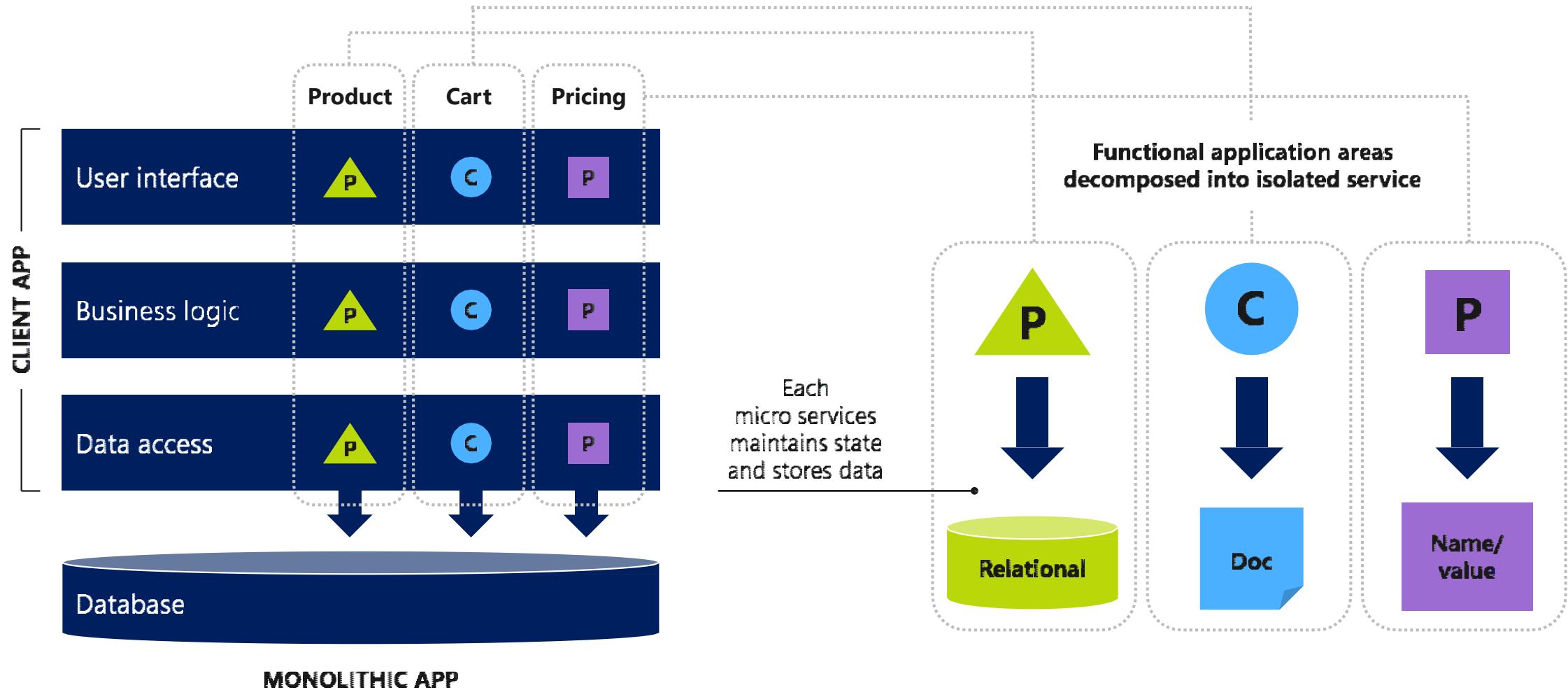


Microservices Architecture



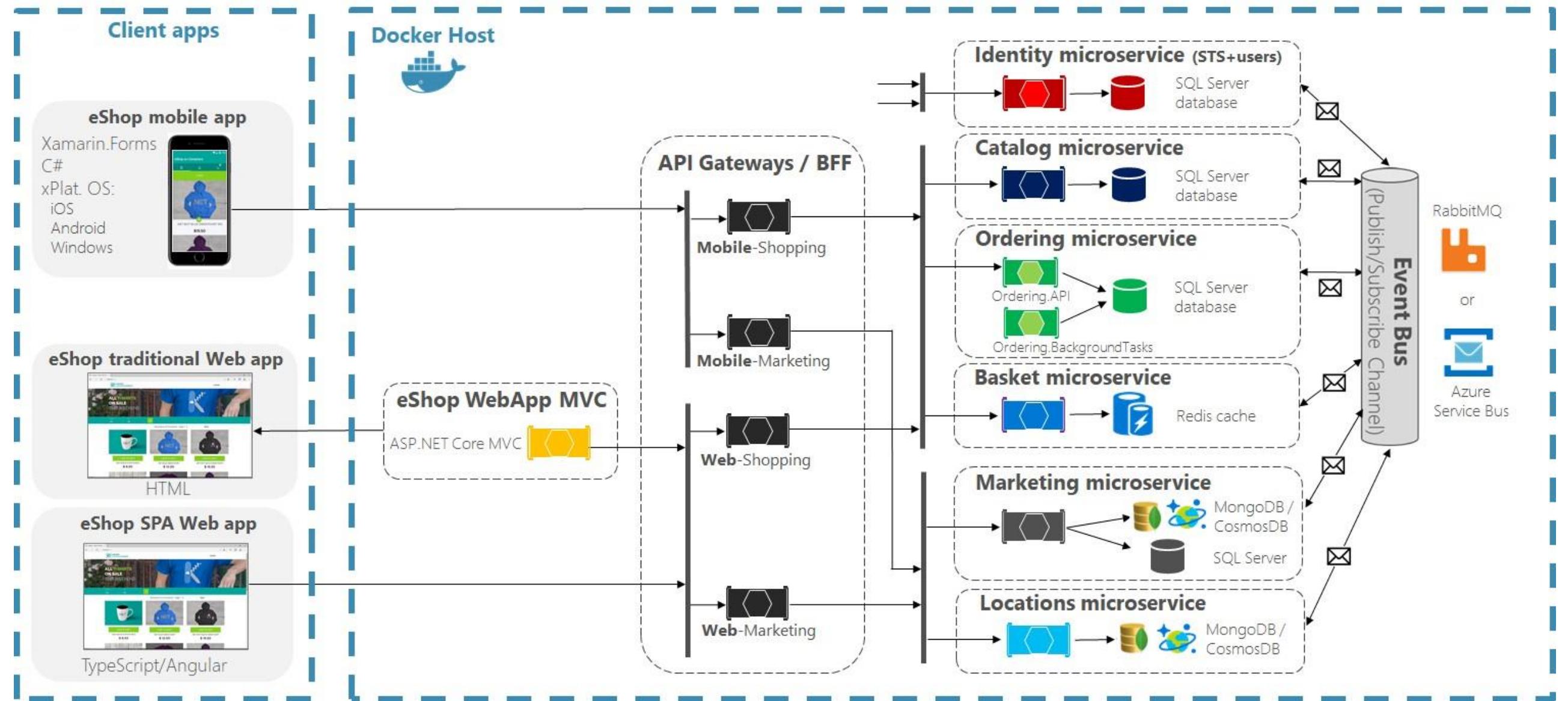
Micro service architecture

Refactoring a monolith to a micro service based architecture



eShopOnContainers reference application

(Development environment architecture)



Container Orchestration

The elements of orchestration



Scheduling



Affinity/anti-affinity



Health monitoring



Failover



Scaling



Networking



Service discovery



Coordinated app upgrades

Kubernetes: the de-facto orchestrator



Portable

Public, private, hybrid,
multi-cloud

Extensible

Modular, pluggable,
hookable, composable

Self-healing

Auto-placement, auto-restart,
auto-replication, auto-scaling

Kubernetes: empowering you to do more



Deploy your
applications quickly
and predictably

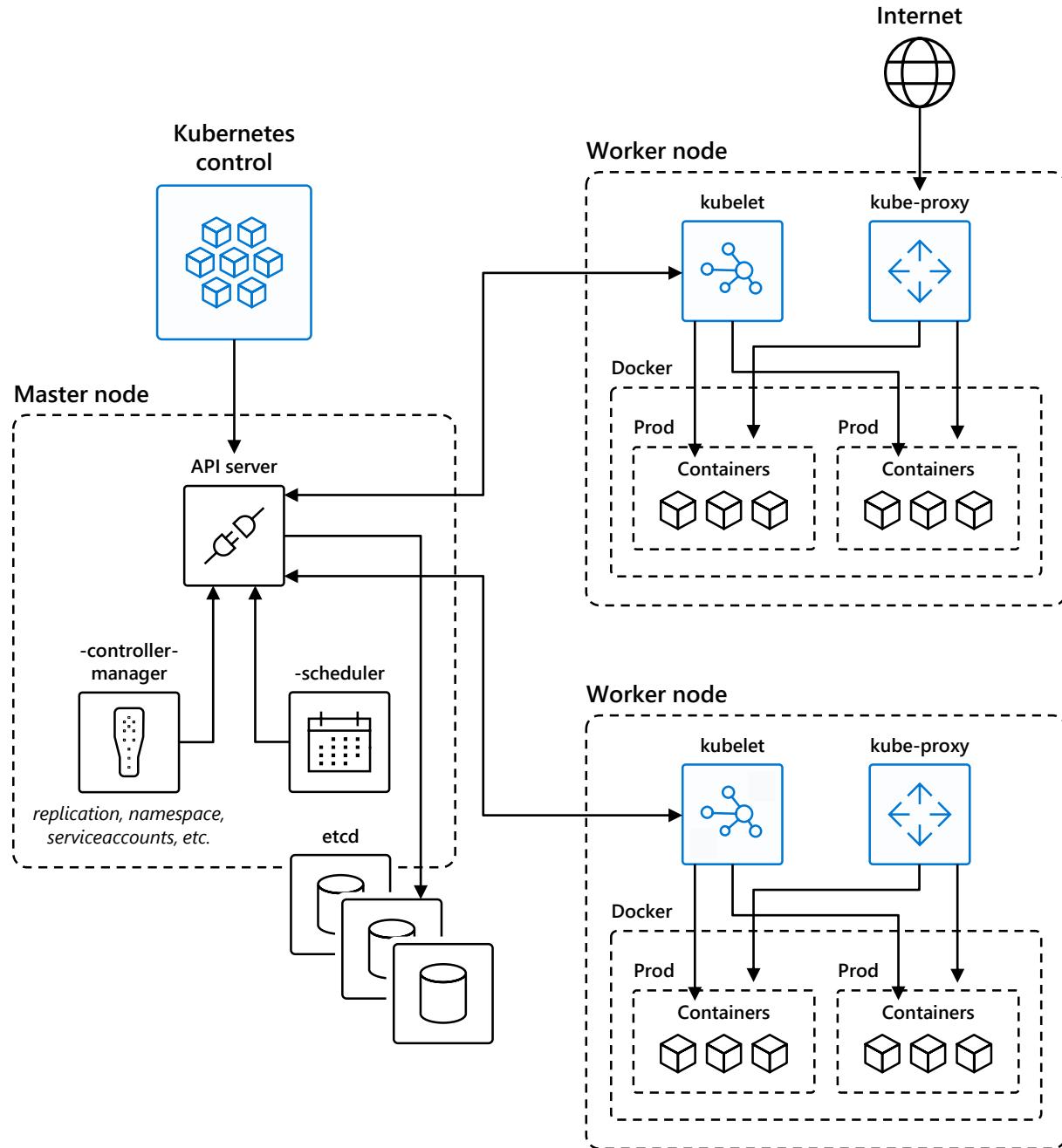
Scale your
applications on
the fly

Roll out
new features
seamlessly

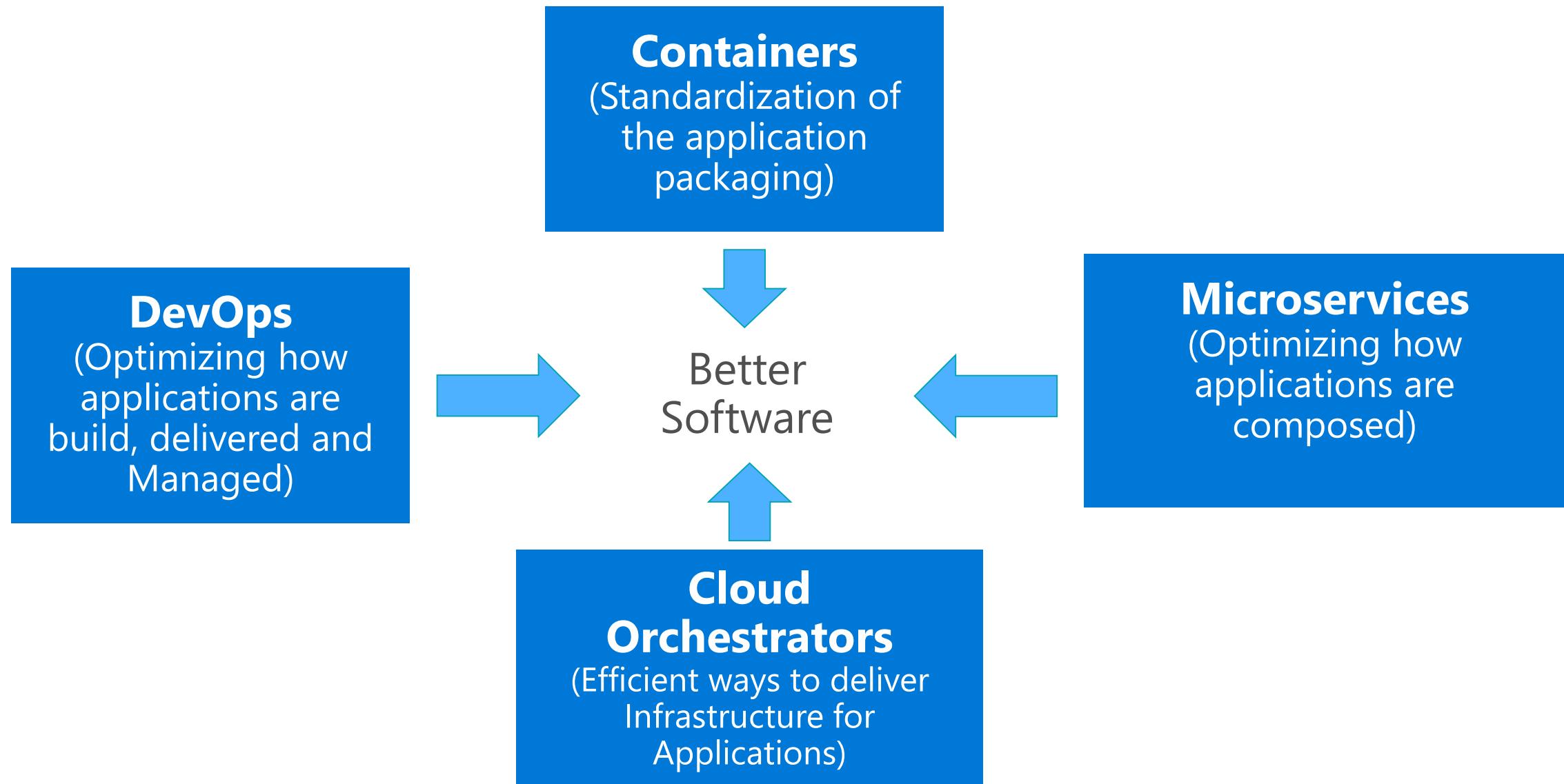
Limit hardware
usage to required
resources only

Kubernetes 101

1. Kubernetes users communicate with API server and apply desired state
2. Master nodes actively enforce desired state on worker nodes
3. Worker nodes support communication between containers
4. Worker nodes support communication from the Internet

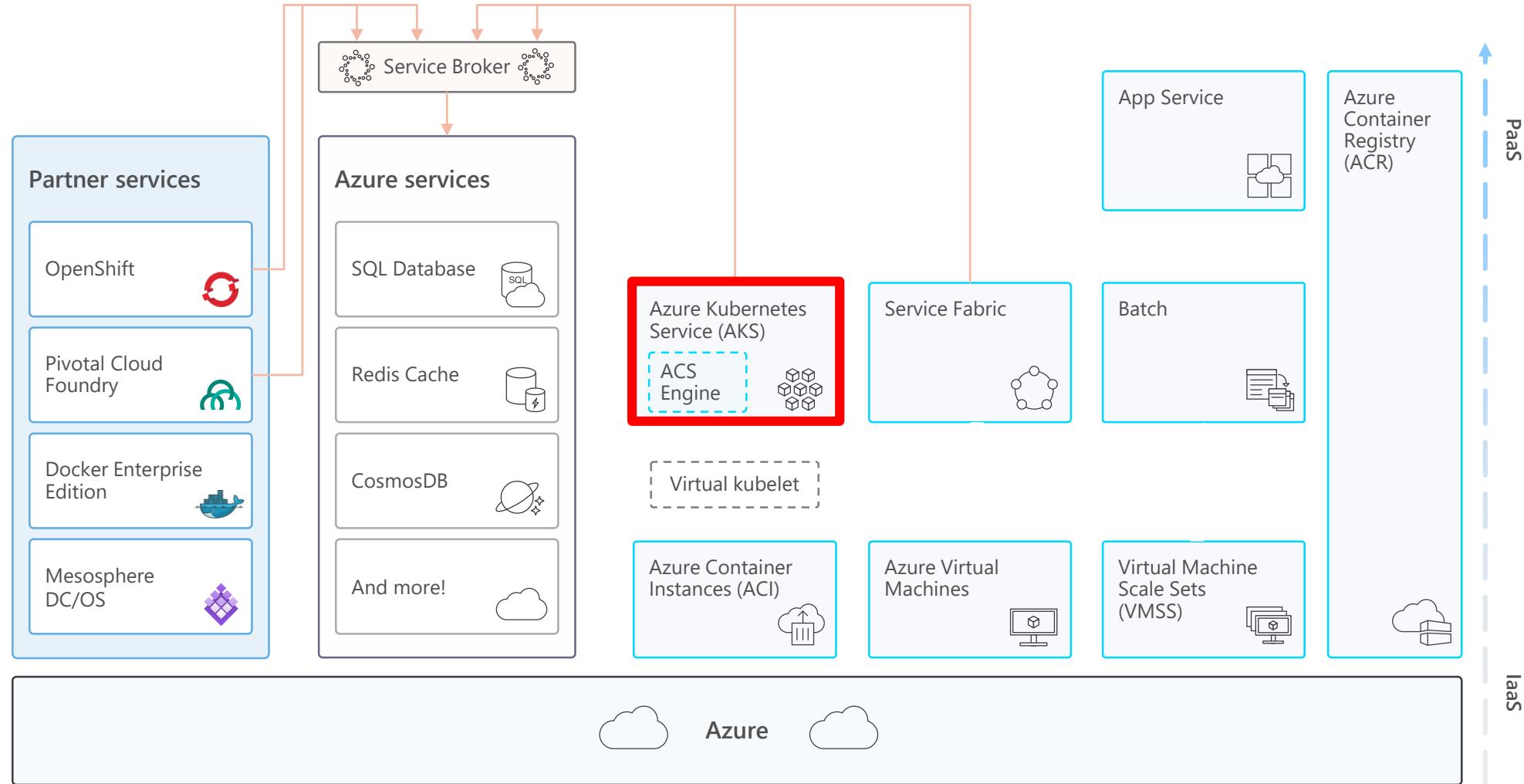


Software Innovation Collision



Containers on Azure

Azure container ecosystem



AKS:

Azure Kubernetes Service

Azure Kubernetes Service (AKS)

Simplify the deployment, management, and operations of Kubernetes



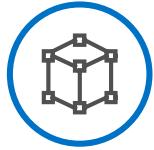
Deploy and
manage Kubernetes
with ease



Scale and run
applications with
confidence



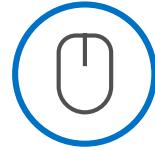
Secure your
Kubernetes
environment



Accelerate
containerized application
development



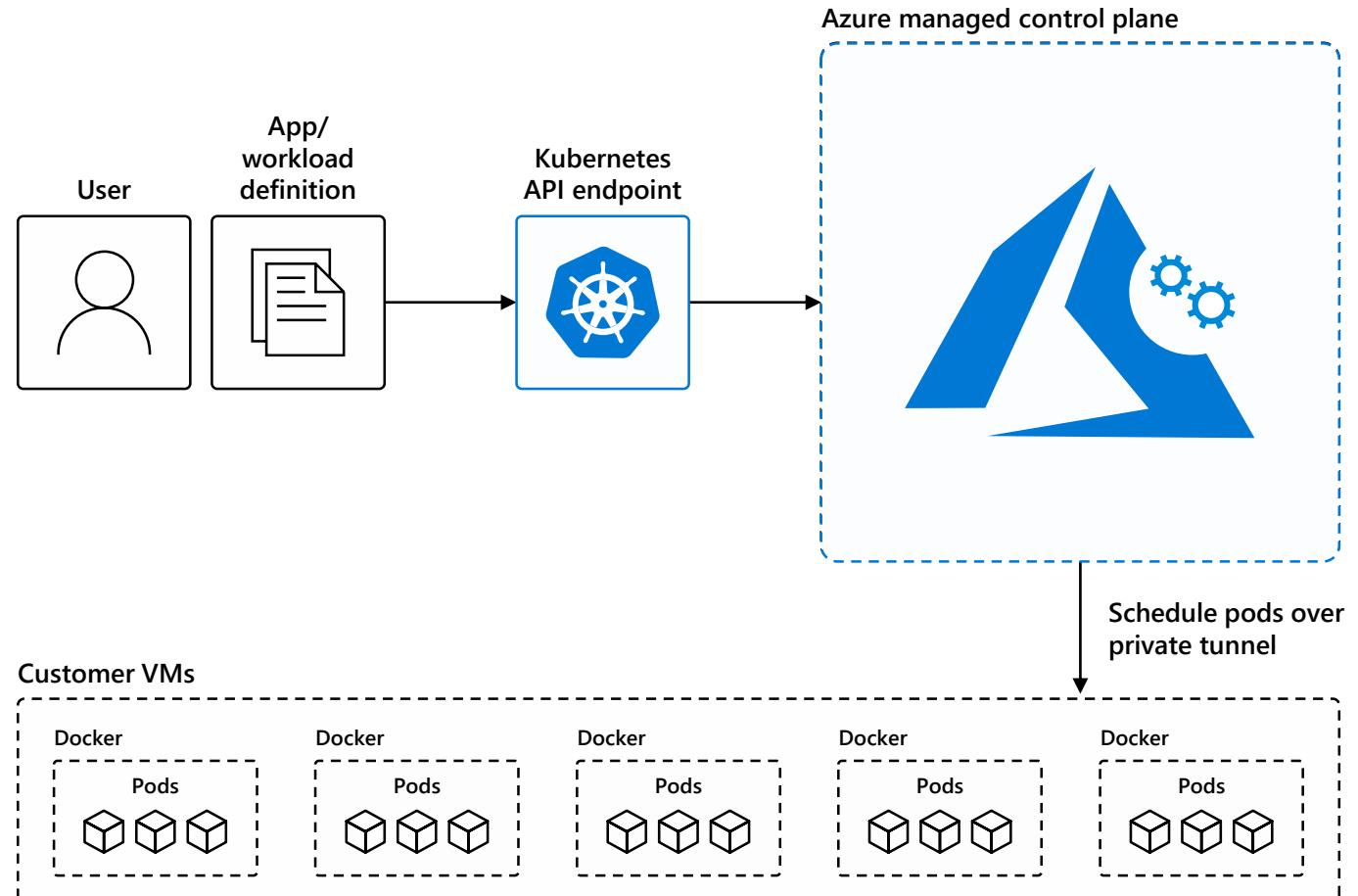
Work how you want
with open-source
tools & APIs



Set up
CI/CD in a
few clicks

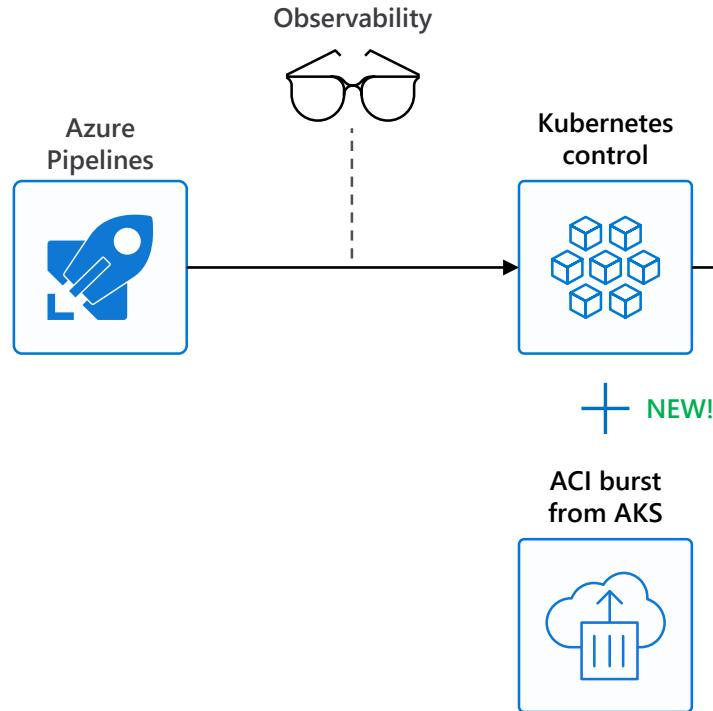
How managed Kubernetes on Azure works

- Automated upgrades, patches
- High reliability, availability
- Easy, secure cluster scaling
- Self-healing
- API server monitoring
- At no charge



Monitoring AKS

Azure Monitor for containers



Visualization

Visualize overall health and performance from clusters to containers with drill downs and filters

Insights

Provide insights with multi-cluster health roll up view

Monitor & Analyze

Monitor and analyze Kubernetes and container deployment performance, events, health, and logs

Response

Native alerting with integration to issue management and ITSM tools

Observability

Observe live container logs on container deployment status

Azure Monitor for containers

1. Get detailed insights about your workloads with Azure Monitor
2. See graphical insights about clusters
3. Filter for details about nodes, controllers, and containers
4. Pull events and logs for detailed activity analysis

4

The screenshot displays the Microsoft Azure portal interface, specifically the 'Containers' tab for a Kubernetes service named 'ContosoSH360KubCluster'. The left sidebar contains navigation links such as Overview, Activity log, Access control (IAM), Tags, Settings, Upgrade, Scale, Properties, Locks, Automation script, Monitoring, Insights (which is highlighted in blue), Metrics (preview), Logs, Support + troubleshooting, and New support request. The main content area shows a table with columns: NAME, STATUS, AVG %, AVERAGE, POD, NODE, RESTARTS, UPTIME, and TREND AVG % (1 BAR =). The table lists several containers: addon-http-application-routing-d..., main, tunnel-front, sh360-sql-data, addon-http-application-routing-n..., kube-proxy, and another kube-proxy entry. Below the table, a search bar shows log entries starting with 'ssh':

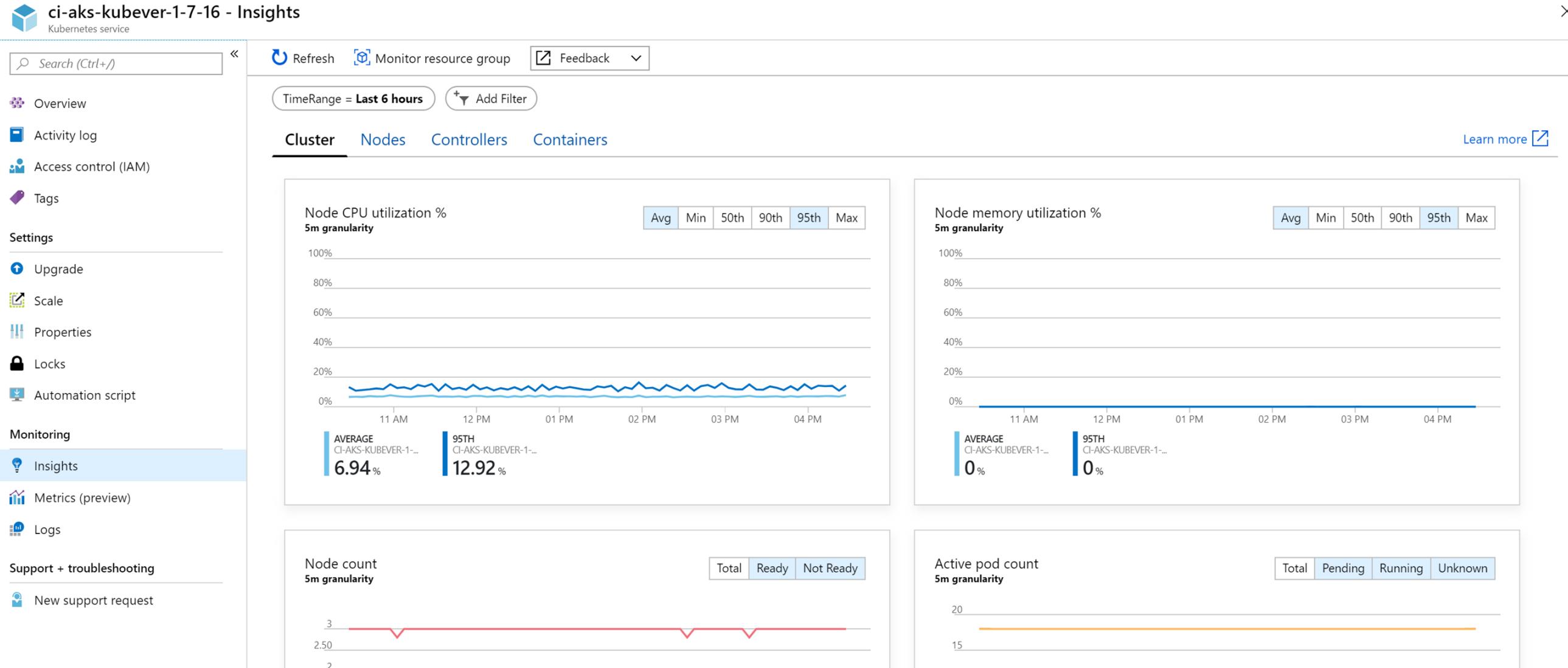
```
2018-12-04T01:44:20.835838006Z [0144:20] INF: Tunnel front and end both server.version == local.version, no rotation needed
2018-12-04T01:44:20.842472017Z [0144:20] INF: Ssh to tunnelEnd is connected with pid: 14934
2018-12-04T01:44:20.843429147Z [0144:20] INF: going to sleep for[30] Seconds
2018-12-04T01:44:51.350804807Z [0144:51] INF: Tunnel front and end both server.version == local.version, no rotation needed
2018-12-04T01:44:51.356066073Z [0144:51] INF: Ssh to tunnelEnd is connected with pid: 14934
2018-12-04T01:44:51.357071505Z [0144:51] INF: going to sleep for[30] Seconds
2018-12-04T01:45:21.855248168Z [0145:21] INF: Tunnel front and end both server.version == local.version, no rotation needed
2018-12-04T01:45:21.86081045Z [0145:21] INF: Ssh to tunnelEnd is connected with pid: 14934
2018-12-04T01:45:21.86182277Z [0145:21] INF: going to sleep for[30] Seconds
2018-12-04T01:45:52.369483588Z [0145:52] INF: Tunnel front and end both server.version == local.version, no rotation needed
2018-12-04T01:45:52.374858059Z [0145:52] INF: Ssh to tunnelEnd is connected with pid: 14934
2018-12-04T01:45:52.375955993Z [0145:52] INF: going to sleep for[30] Seconds
2018-12-04T01:46:22.888319192Z [0146:22] INF: Tunnel front and end both server.version == local.version, no rotation needed
2018-12-04T01:46:22.888879068Z [0146:22] INF: Ssh to tunnelEnd is connected with pid: 14934
2018-12-04T01:46:22.889973032Z [0146:22] INF: going to sleep for[30] Seconds
```

Easy onboarding experience

- Creating new AKS Cluster
- az aks create --resource-group myAKSCluster --name myAKSCluster --node-count 1 **--enable-addons monitoring** --generate-ssh-keys
 - Enable monitoring on existing AKS Cluster
- az aks **enable-addons -a monitoring** -n MyExistingManagedCluster -g MyExistingManagedClusterRG

Overview of AKS cluster health

Home > Monitor - Containers > ci-aks-kubever-1-7-16 - Insights



Node event logs

Home > Kubernetes services > monclu - Insights (preview) > Logs

Logs

defaultworkspace-5abfd9c4-ec8c-4db9-acd4-c762dce93508-cca



New Query 1*



Help Settings Query explorer

Run

Time range: Set in query

Save

Copy link

Export

Set alert Pin

```
»
let startDateTime = datetime('2018-09-15T08:30:00.000Z');
let endDateTime = datetime('2018-09-26T14:31:46.659Z');
let EmptyKubeEvents_CLTable = datatable(TimeGenerated: datetime, Name_s: string, ObjectKind_s: string,
                                         Type_s: string, Reason_s: string, Message: string, Namespace_s: string) [];
let KubeEvents_CLTable = union isfuzzy = true EmptyKubeEvents_CLTable, KubeEvents_CL
| where TimeGenerated >= startDateTime and TimeGenerated < endDateTime
| where ObjectKind_s == 'Node'
| where Name_s == 'aks-agentpool-41197944-0'
| project TimeGenerated, Name_s , ObjectKind_s , Type_s, Reason_s , Message , Namespace_s
| order by TimeGenerated desc;
KubeEvents_CLTable
...
```

Completed

⌚ 00:00:01.608 ⏱ 7 records

TABLE CHART Columns ▾

Display time (UTC+00:00) ▾

Drag a column header and drop it here to group by that column

TimeGenerated [UTC]	Name_s	ObjectKind_s	Type_s	Reason_s	Message	Namespace_s
> 2018-09-17T19:19:28.000	aks-agentpool-41197944-0	Node	Normal	Starting	Starting kubelet.	
> 2018-09-17T19:19:28.000	aks-agentpool-41197944-0	Node	Normal	NodeHasSufficientDisk	Node aks-agentpool-41197944-0 status is now: NodeHasSufficientDisk	
> 2018-09-17T19:19:28.000	aks-agentpool-41197944-0	Node	Normal	NodeHasSufficientMemory	Node aks-agentpool-41197944-0 status is now: NodeHasSufficientMe...	
> 2018-09-17T19:19:28.000	aks-agentpool-41197944-0	Node	Normal	NodeHasNoDiskPressure	Node aks-agentpool-41197944-0 status is now: NodeHasNoDiskPressure	
> 2018-09-17T19:19:28.000	aks-agentpool-41197944-0	Node	Normal	NodeHasSufficientPID	Node aks-agentpool-41197944-0 status is now: NodeHasSufficientPID	

Pod usage and details

Home > Kubernetes services > monclu - Insights (preview)

monclu - Insights (preview)

Kubernetes service

TimeRange = Last 7 days Add Filter

Cluster Nodes Controllers Containers

Monitor resource group Learn more Feedback

All 44 item(s)

NAME	STATUS	95TH %	95TH	POD	NODE	RESTARTS	UPTIME	TREND 95TH % (1 BAR = 8H)
cpu-demo-ctr	Ok	50%	1001 mc	cpu-demo-679b7...	aks-agentpool-4...	0	5 days	
omsagent	Ok	11%	16 mc	omsagent-h4v6l	aks-agentpool-4...	0	8 days	
omsagent	Ok	4%	6 mc	omsagent-rs-57f...	aks-agentpool-4...	0	8 days	
tunnel-front	Ok	2%	47 mc	tunnelfront-85c6...	aks-agentpool-4...	0	8 days	
memory-demo-ctr	Ok	2%	41 mc	memory-demo	aks-agentpool-4...	0	6 days	
addon-http-application-routi...	Ok	1%	0.1 mc	addon-http-appli...	aks-agentpool-4...	0	8 days	
heapster-nanny	Ok	0.9%	0.4 mc	heapster-5457df...	aks-agentpool-4...	0	8 days	
heapster	Ok	0.8%	0.7 mc	heapster-5457df...	aks-agentpool-4...	0	8 days	
main	Ok	0.5%	0.5 mc	kubernetes-dash...	aks-agentpool-4...	0	8 days	
redirector	Ok	0.2%	4 mc	kube-svc-redirect...	aks-agentpool-4...	0	8 days	

View container logs

Container Name: cpu-demo-ctr

Container ID: ac13d52d72eb0d348006804787e42d6dffe6ce08ef9d99131836274974d23b61

Container Status: running

Image stress

Image Tag: latest

Container Creation Time Stamp: 9/20/2018, 1:48:58 PM

Start Time: 9/20/2018, 1:48:58 PM

Finish Time: -

CPU Limit: 2000 mc

CPU Request: 1000 mc

Memory Limit:

Live container logs

tunnelfront-1777119743-tvnmc (tunnel-front) (3 New Logs)  

ssh 28/84   Scroll  Pause Clear  

```
2018-12-04T16:16:57.16908281Z [ 16:16:57 ] INF: going to sleep for:[5] Seconds
2018-12-04T16:17:02.835523435Z [ 16:17:02 ] INF: Tunnel front and end both server.version == local.version, no rotation needed
2018-12-04T16:17:02.911547049Z [ 16:17:02 ] INF: Ssh to tunnelEnd is connected with pid: 31184
2018-12-04T16:17:02.912713087Z [ 16:17:02 ] INF: going to sleep for:[5] Seconds
2018-12-04T16:17:08.580647777Z [ 16:17:08 ] INF: Tunnel front and end both server.version == local.version, no rotation needed
2018-12-04T16:17:08.655039736Z [ 16:17:08 ] INF: Ssh to tunnelEnd is connected with pid: 31184
2018-12-04T16:17:08.65636918Z [ 16:17:08 ] INF: going to sleep for:[5] Seconds
2018-12-04T16:17:14.261892521Z [ 16:17:14 ] INF: Tunnel front and end both server.version == local.version, no rotation needed
2018-12-04T16:17:14.358894828Z [ 16:17:14 ] INF: Ssh to tunnelEnd is connected with pid: 31184
2018-12-04T16:17:14.360003365Z [ 16:17:14 ] INF: going to sleep for:[5] Seconds
2018-12-04T16:17:19.985445879Z [ 16:17:19 ] INF: Tunnel front and end both server.version == local.version, no rotation needed
2018-12-04T16:17:20.068811836Z [ 16:17:20 ] INF: Ssh to tunnelEnd is connected with pid: 31184
2018-12-04T16:17:20.07015748Z [ 16:17:20 ] INF: going to sleep for:[5] Seconds
2018-12-04T16:17:25.697537373Z [ 16:17:25 ] INF: Tunnel front and end both server.version == local.version, no rotation needed
2018-12-04T16:17:25.780151505Z [ 16:17:25 ] INF: Ssh to tunnelEnd is connected with pid: 31184
2018-12-04T16:17:25.784029933Z [ 16:17:25 ] INF: going to sleep for:[5] Seconds
```

Multi-cluster monitoring

Home > Monitor - Containers

Monitor - Containers

Microsoft

 Search (Ctrl+)

 Refresh

 Feedback

 Overview

 Activity log

 Alerts

 Metrics

 Logs

 Service Health

Insights

 Applications

 Virtual Machines (preview)

 Containers

 Network

 More

Settings

 Diagnostics settings

 Autoscale

Cluster Status Summary

53
Total

6  Critical

0  Warning

11  Unknown

AKS Healthy 23 

AKS-engine Healthy 0 

Non-monitored 13

Monitored clusters(40)

Non-monitored clusters(13)

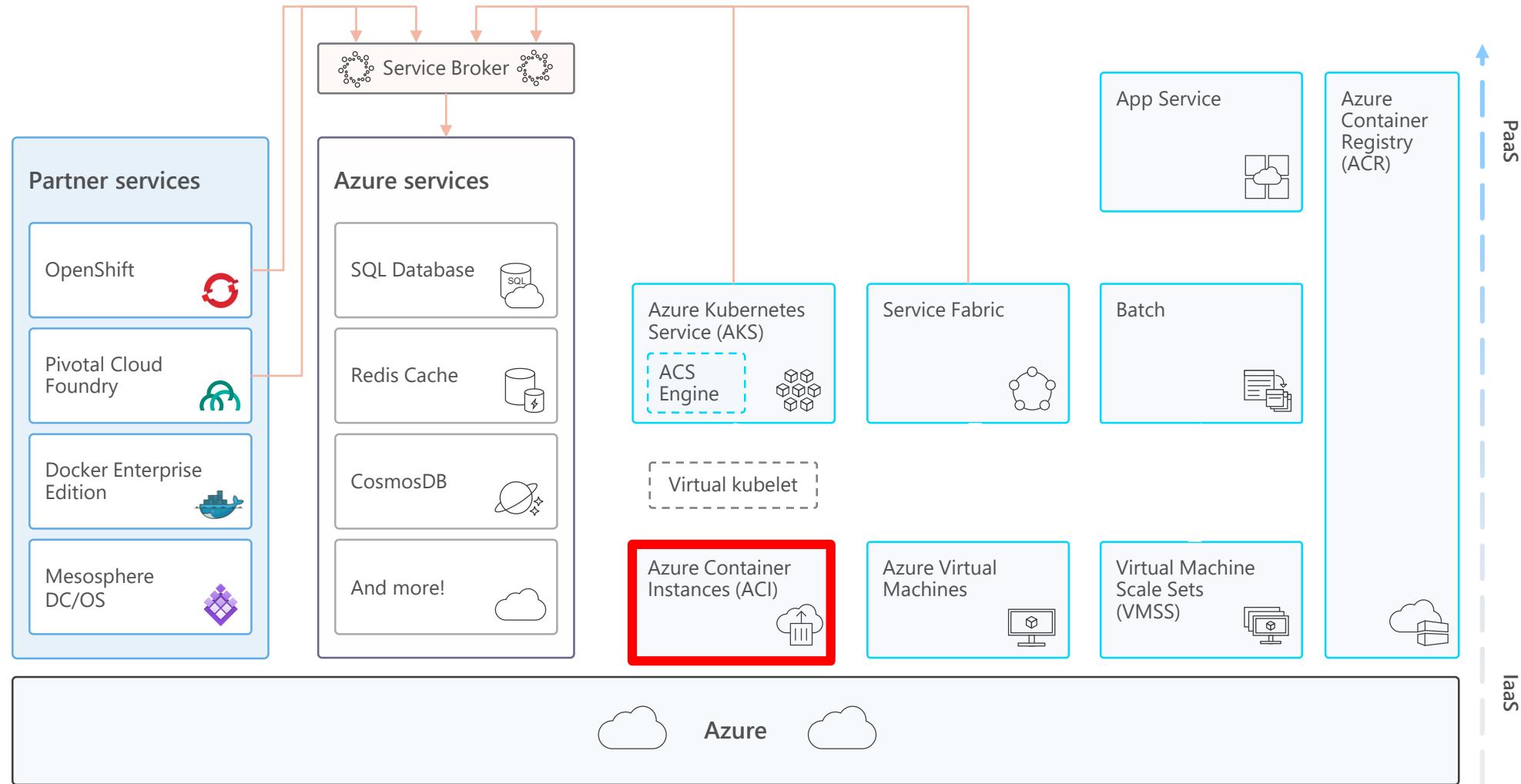
Learn more 

Search by name...

40 item:

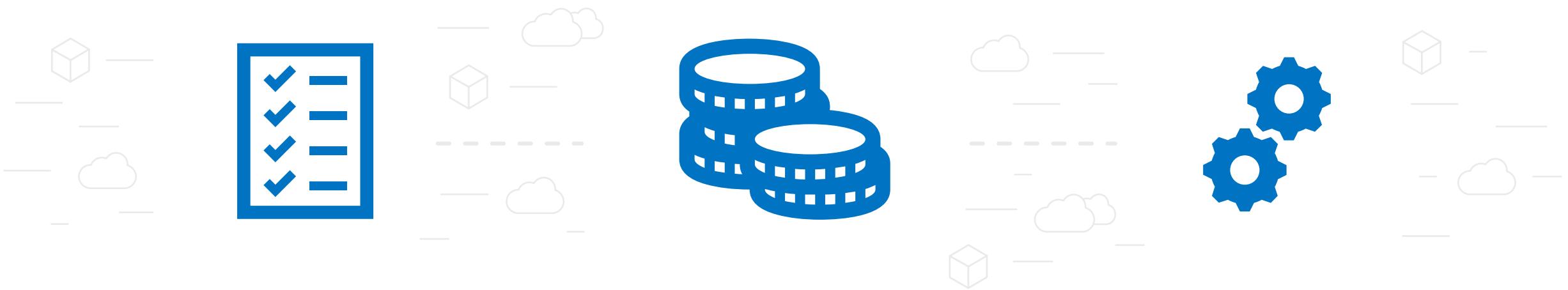
CLUSTER NAME	CLUSTER TYPE	STATUS	↑↓ NODES	USER PODS	SYSTEM PODS
ci-aks-kubever-1-7-9	AKS	 Healthy	 3 / 3	 6 / 6	 15 / 15
ci-aks-kubever-1-8-7	AKS	 Healthy	 3 / 3	 3 / 3	 15 / 15
ci-aks-kubever-1-9-1	AKS	 Healthy	 3 / 3	 3 / 3	 15 / 15
ci-aks-kubever-1-9-2	AKS	 Healthy	 3 / 3	 3 / 3	 15 / 15
ci-aks-kubever-1-9-6	AKS	 Healthy	 3 / 3	 3 / 3	 15 / 15
ci-aks-kubever-1-7-16	AKS	 Healthy	 3 / 3	 3 / 3	 15 / 15

Azure container ecosystem



ACI: Azure Container Instances

What's a serverless container?



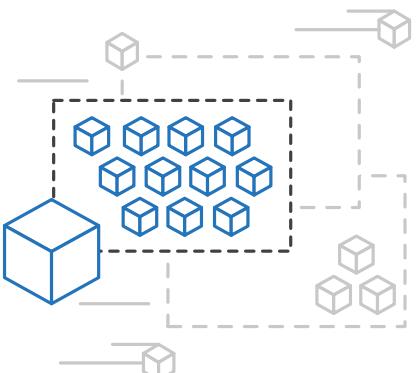
Invisible
infrastructure

Micro-billing

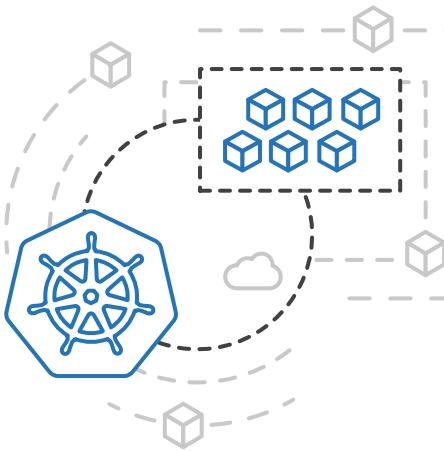
Event-based programming
models

Azure Container Instances (ACI)

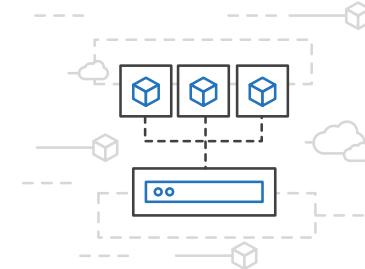
Easily run serverless containers



Run containers
without managing
servers

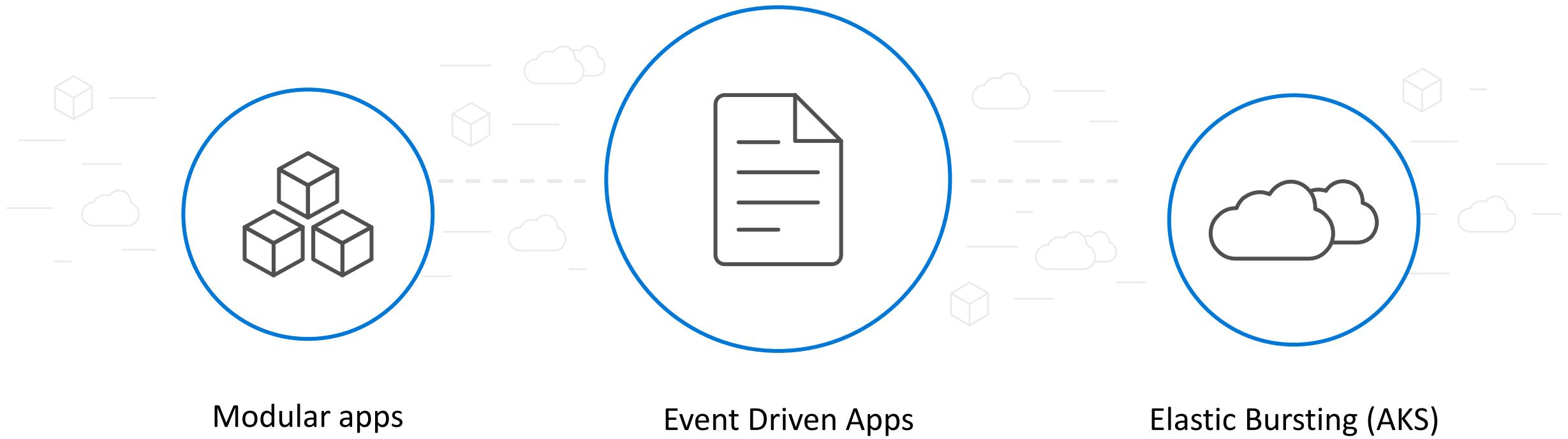


Containers as a primitive
billed per second



Secure applications with
hypervisor isolation

What can you build with ACI today

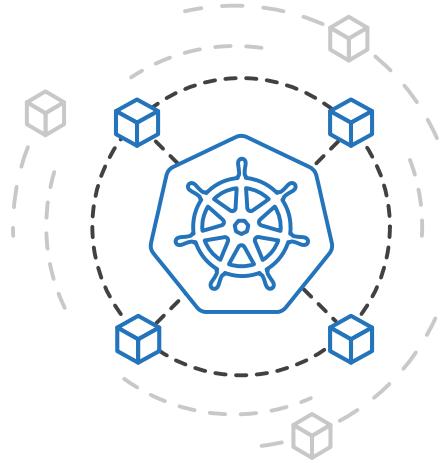


ACI Features

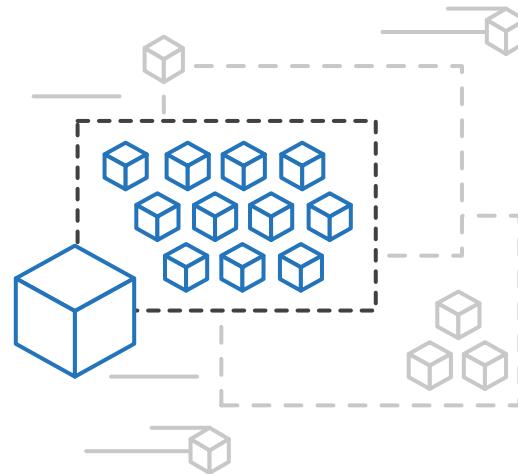
- ✓ Create & manage in Azure portal
- ✓ CPU / Memory metrics in the portal
- ✓ Execute (EXEC) commands in running containers
- ✓ DNS name labels for containers
- ✓ Stream container logs
- ✓ Container restart policies
- ✓ Various volume types for mounting to Linux
- ✓ Uptime SLA of 99.9%

ACI Connector for Kubernetes

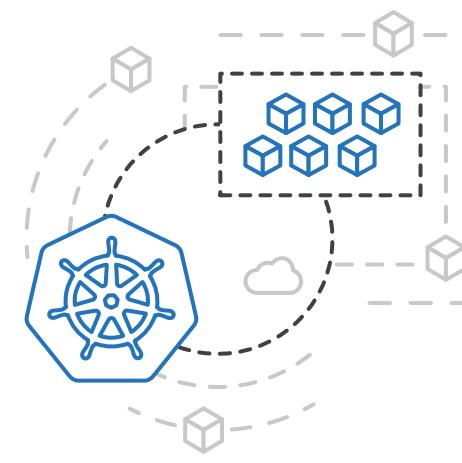
aka "Virtual Kubelet"



Kubernetes provides rich
orchestration capabilities

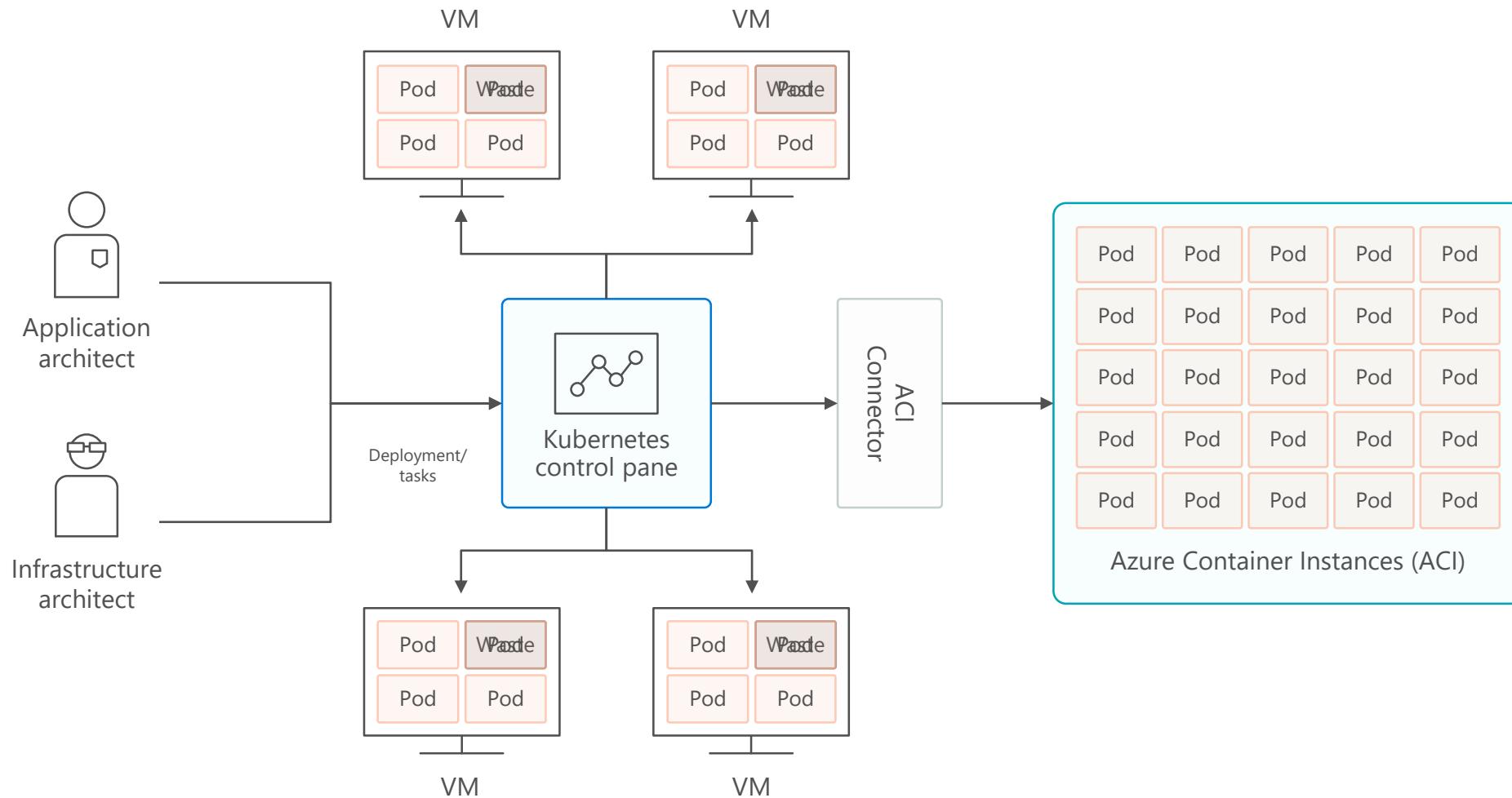


ACI provides infinite
container-based scale

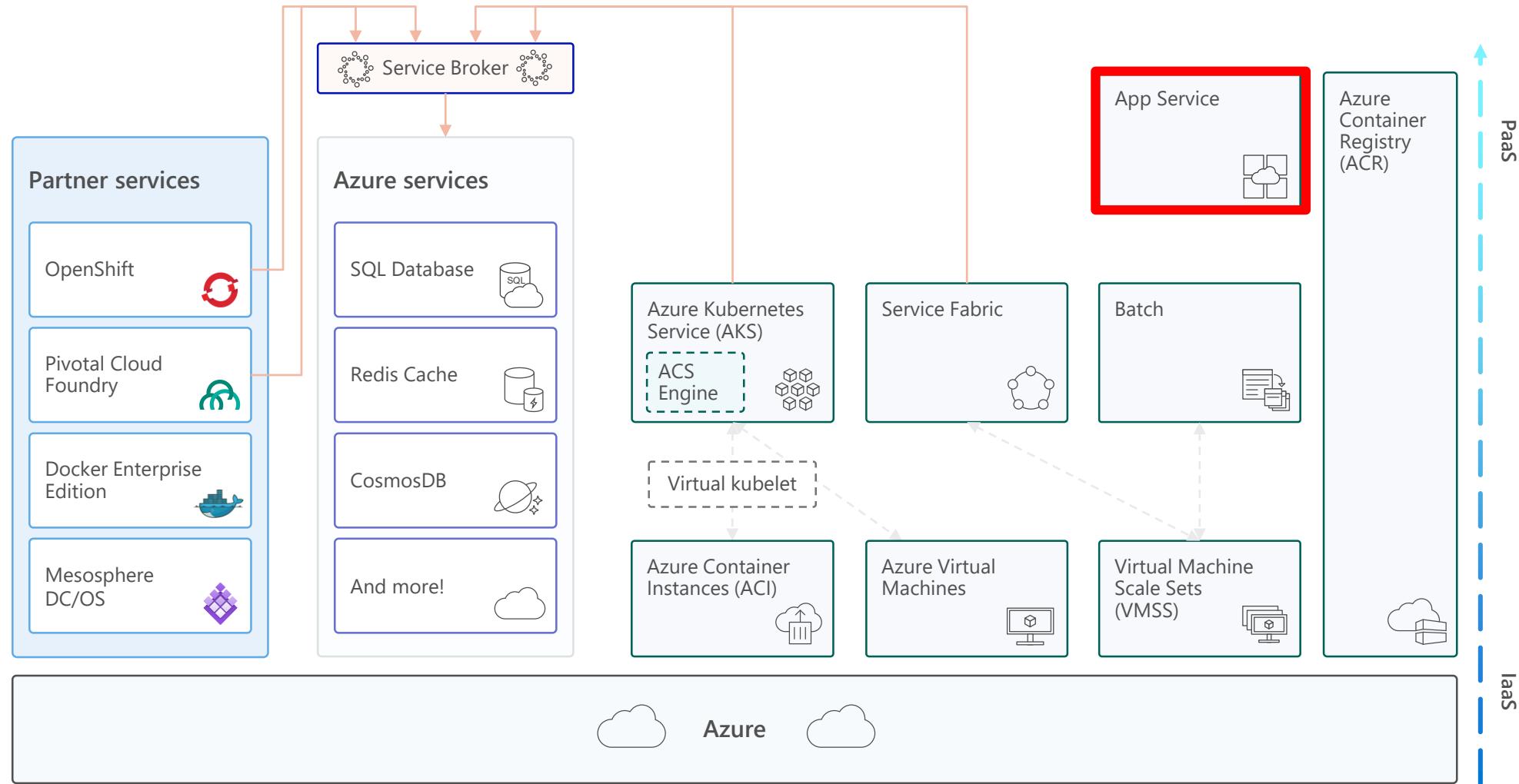


The ACI Connector for
K8s brings them
together

Bursting with the ACI Connector



Azure container ecosystem



Azure App Service

App Service

Easily deploy and run container-based web apps at scale

Accelerated outer loop



Tight integration w/ Docker Hub, Azure Container Registry



Built-in CI/CD w/
Deployment Slots



Intelligent diagnostics &
troubleshooting, remote debugging

Fully managed platform



Automatic scaling
and load balancing



High availability
w/ auto-patching



Backup &
recovery

Flexibility & choices



From CLI, portal, or
ARM template



Single Docker image, multi
container w/ Docker Compose



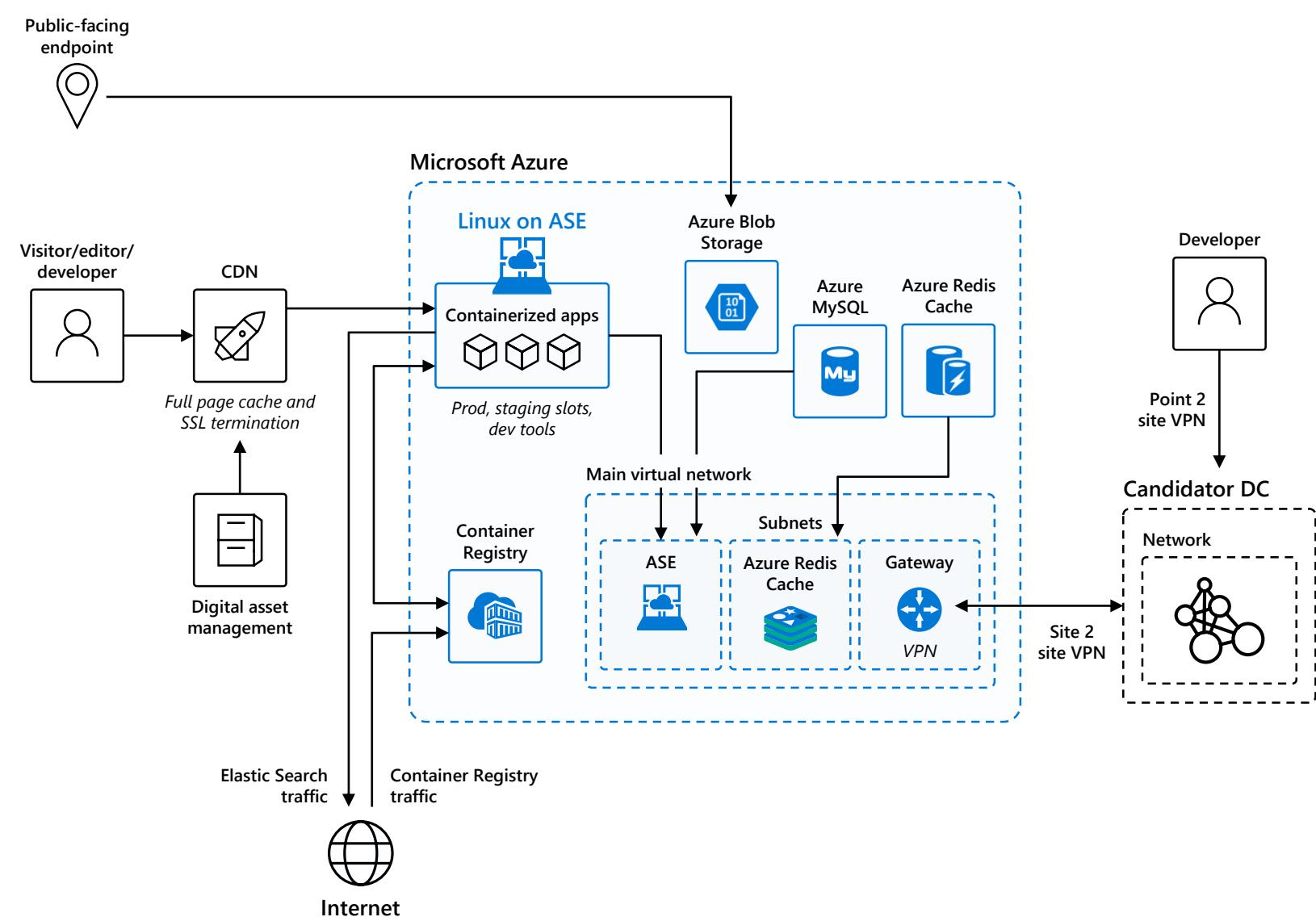
IntelliJ, Jenkins, Maven
Visual Studio family

Nobel Prize website

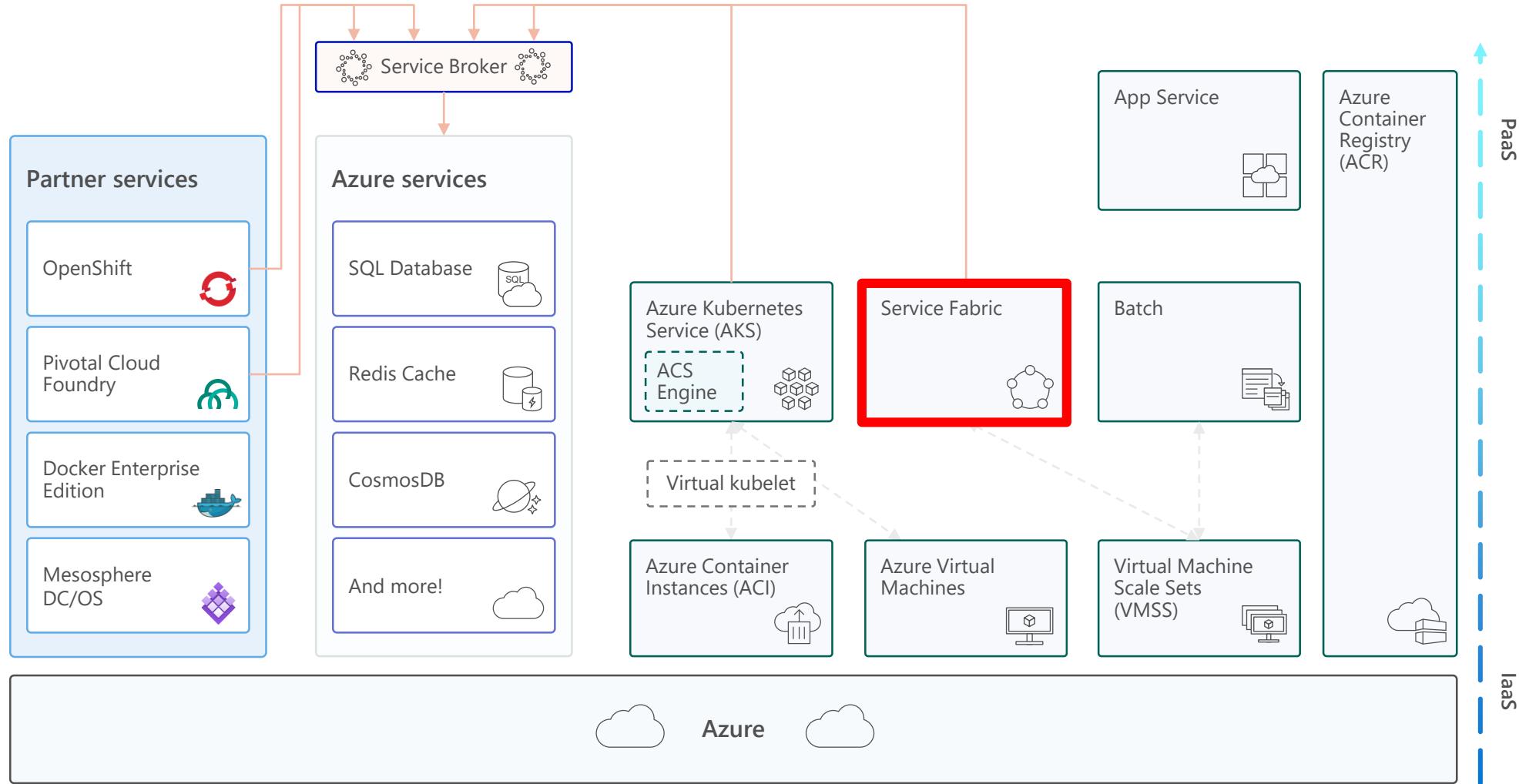
Deploy a global website using Linux containers in a PaaS environment

- Run containerized applications without worrying about the infrastructure
- Leave the scaling orchestration to our PaaS platform for hassle-free scaling for higher traffic loads
- Secure your applications in an Azure Virtual Network to meet security requirements

Simple to get started, but also robust to handle global scale with little maintenance



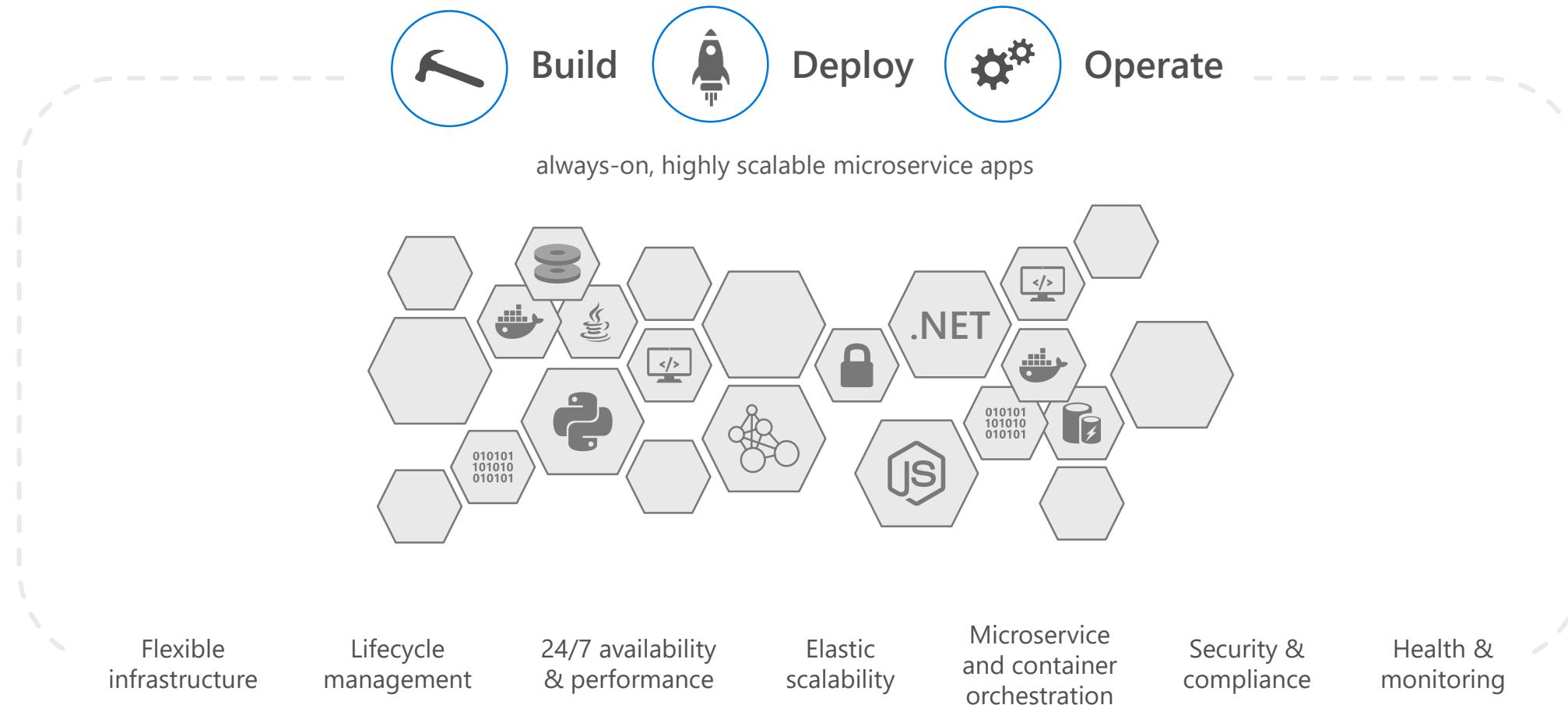
Azure container ecosystem



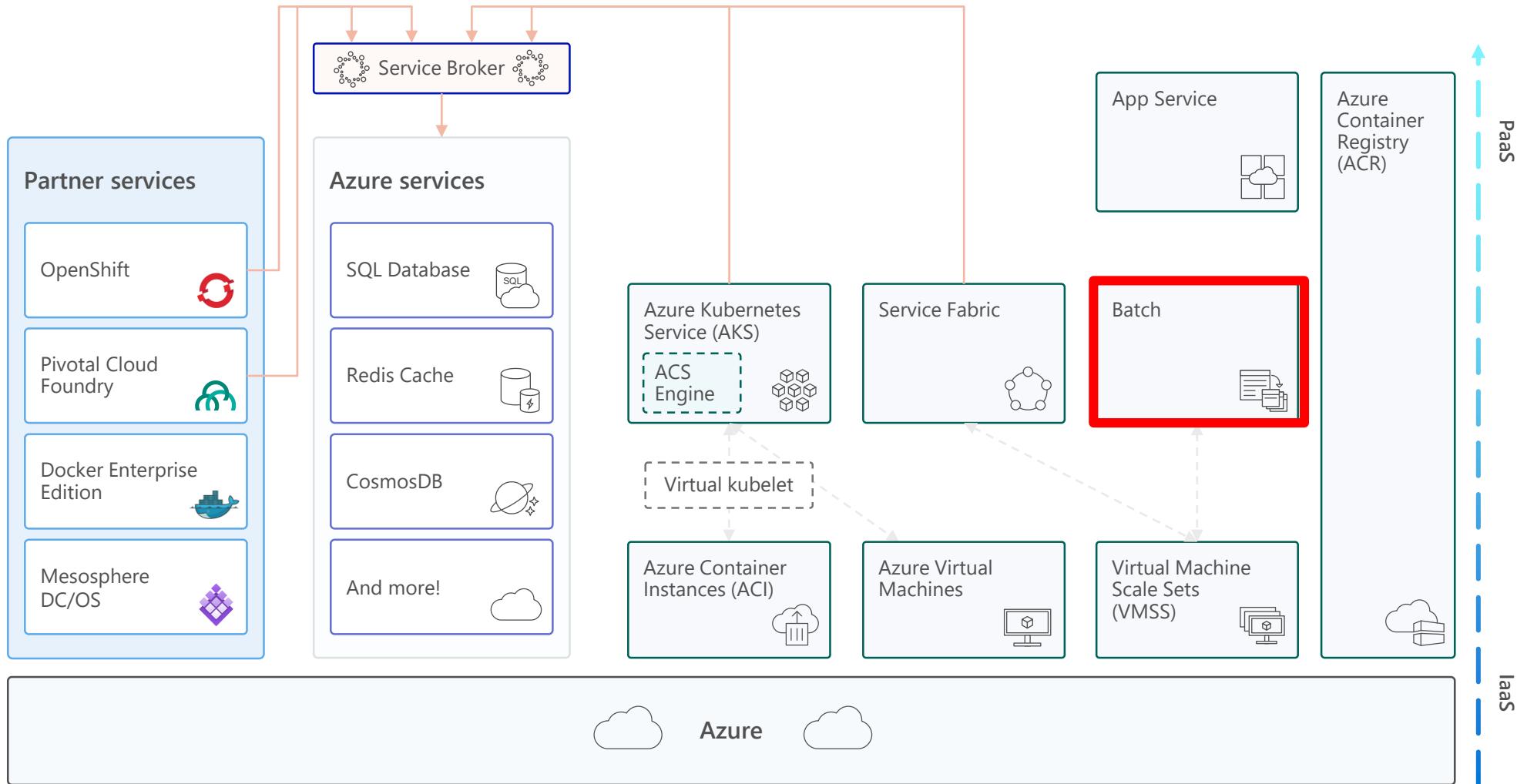
Azure Service Fabric

Azure Service Fabric

A microservices platform for business critical applications



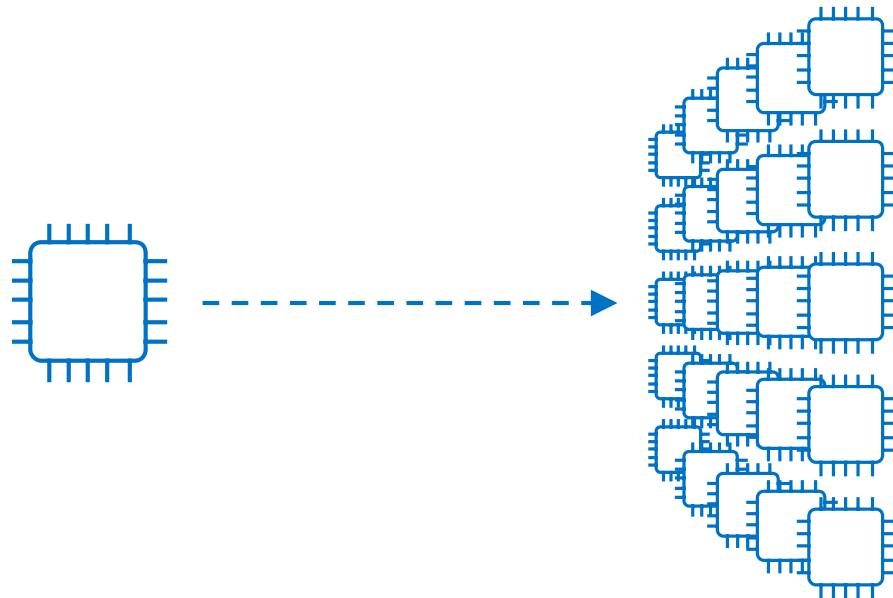
Azure container ecosystem



Azure Batch

Azure Batch

Run repetitive compute jobs using containers

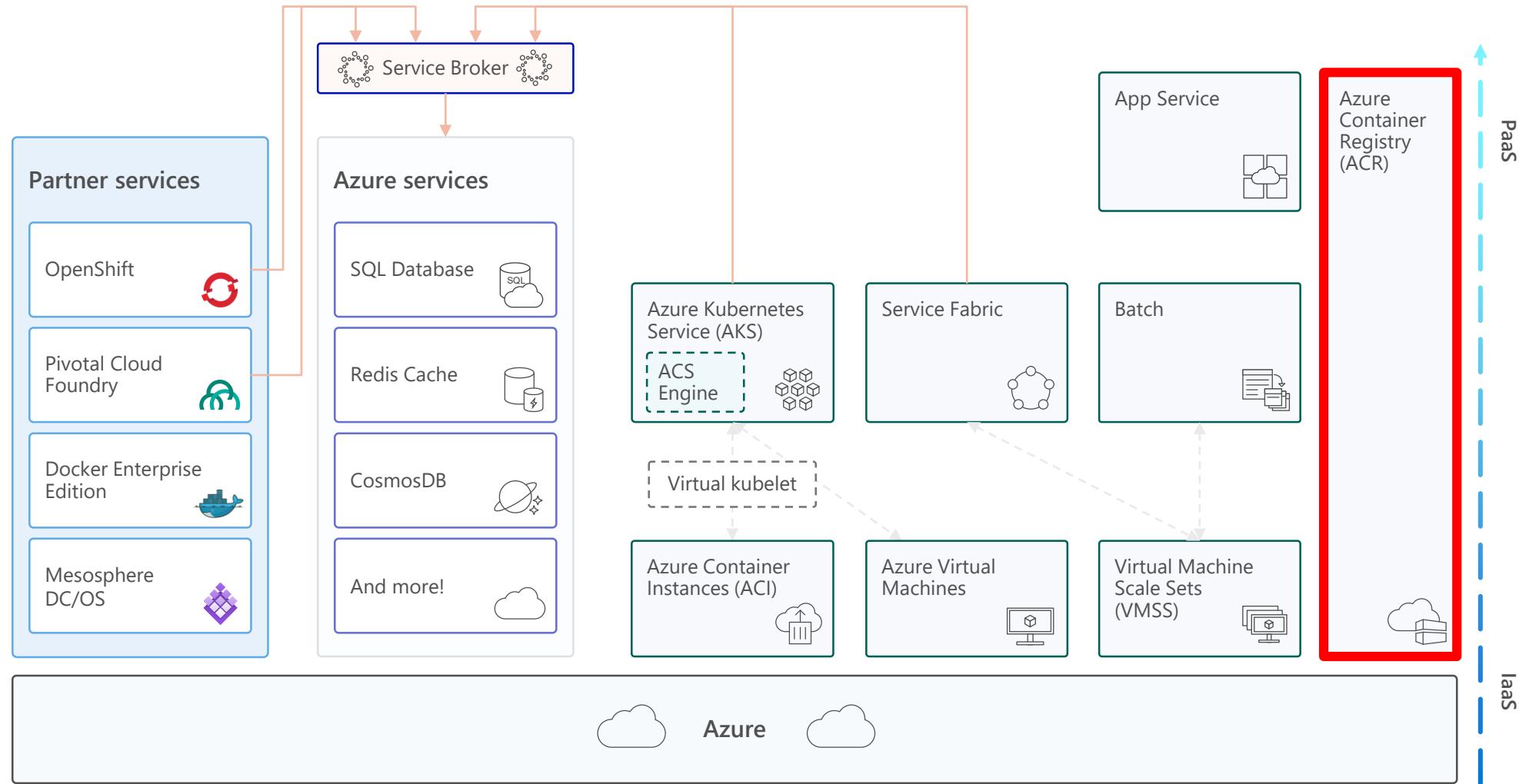


Enable applications and algorithms to easily and efficiently run in parallel at scale.

Run Batch tasks without having to manage an environment and dependencies.

Package, execute, and scale your High Performance Computing applications and batch workloads in a consistent, reproducible manner.

Azure container ecosystem

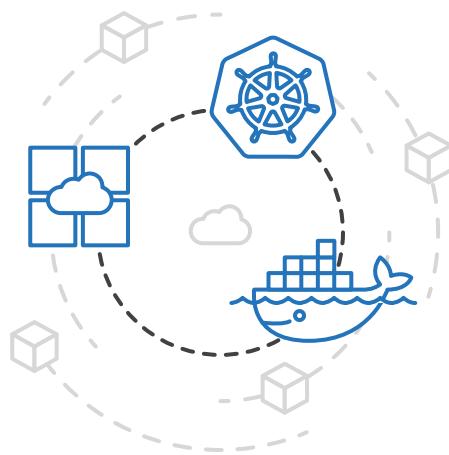


ACR:

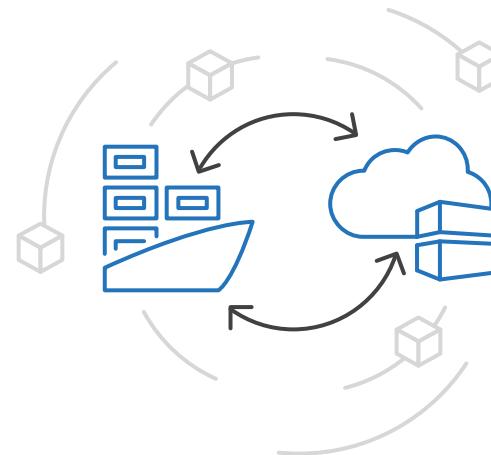
Azure Container Registry

Azure Container Registry

Manage a Docker private registry as a first-class Azure resource



Manage images for all
types of containers

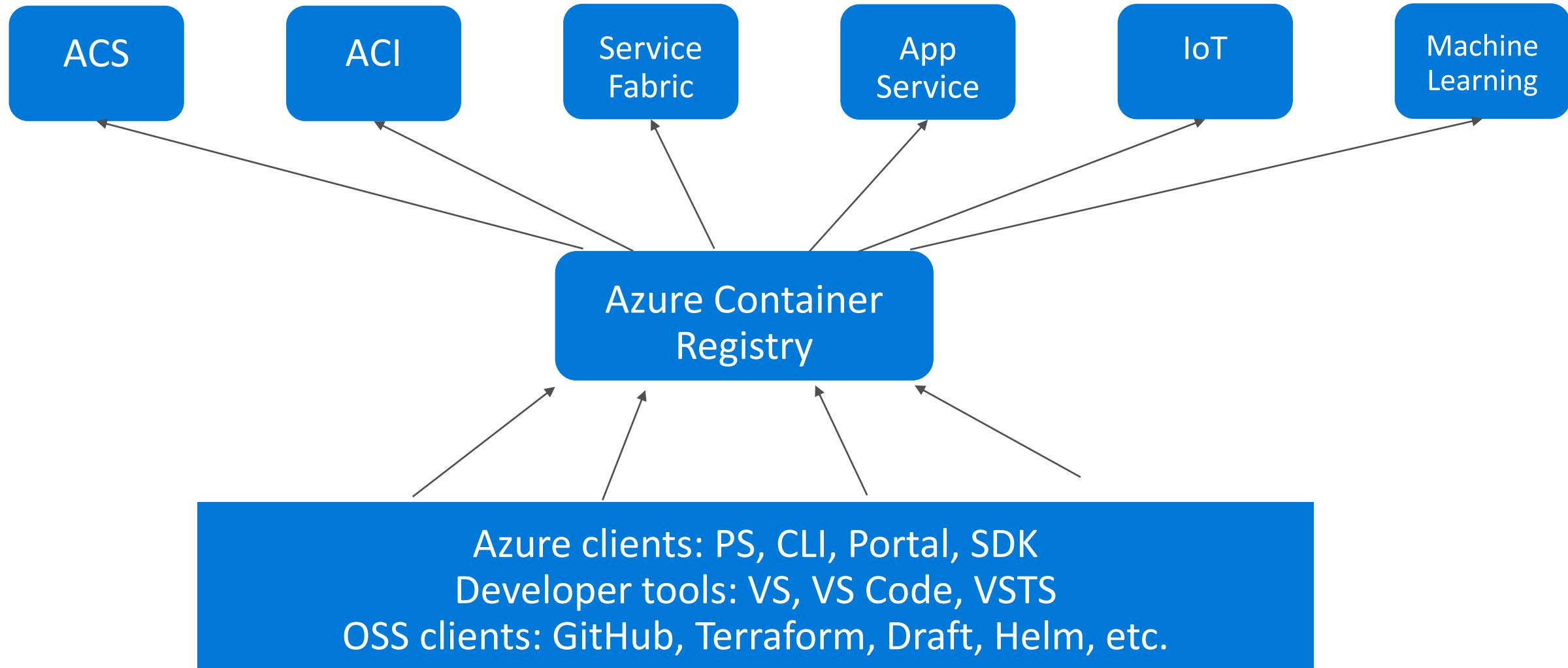


Use familiar, open-
source Docker CLI tools



Azure Container Registry
geo-replication

Containers as the App Packaging Format



ACR Geo-replication

Push Built Images

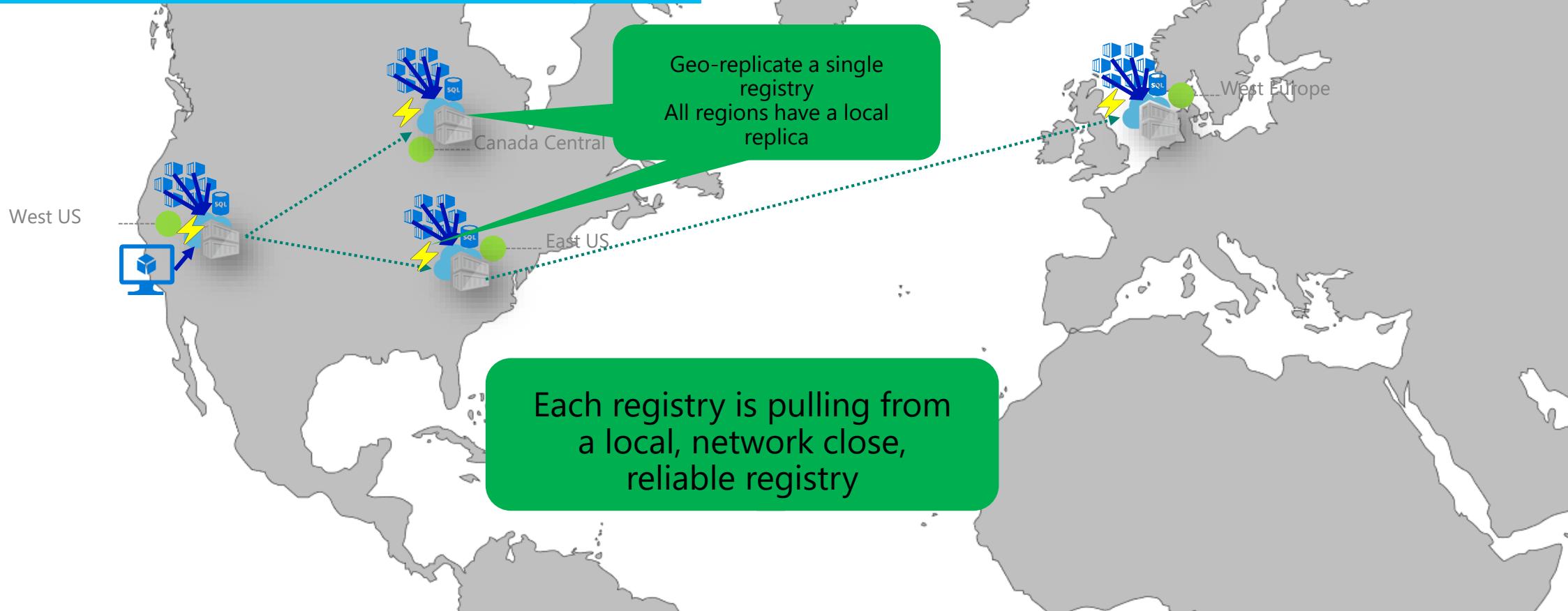
```
docker push contoso.azurecr.io/web:1234
```

ACR Syncs Image Blobs

Regional Web Hooks fire for local deployments

Nodes Pull Images

```
docker pull contoso.azurecr.io/web:1234
```



AKS Roadmap

Developer Updates

Developer productivity

[Serverless for Azure Kubernetes Service \(AKS\)](#): Public Preview since Dec 2018

Elastically provision additional compute capacity in seconds without worrying about managing the infrastructure

[Cluster autoscaling](#): Public Preview Q1 CY19

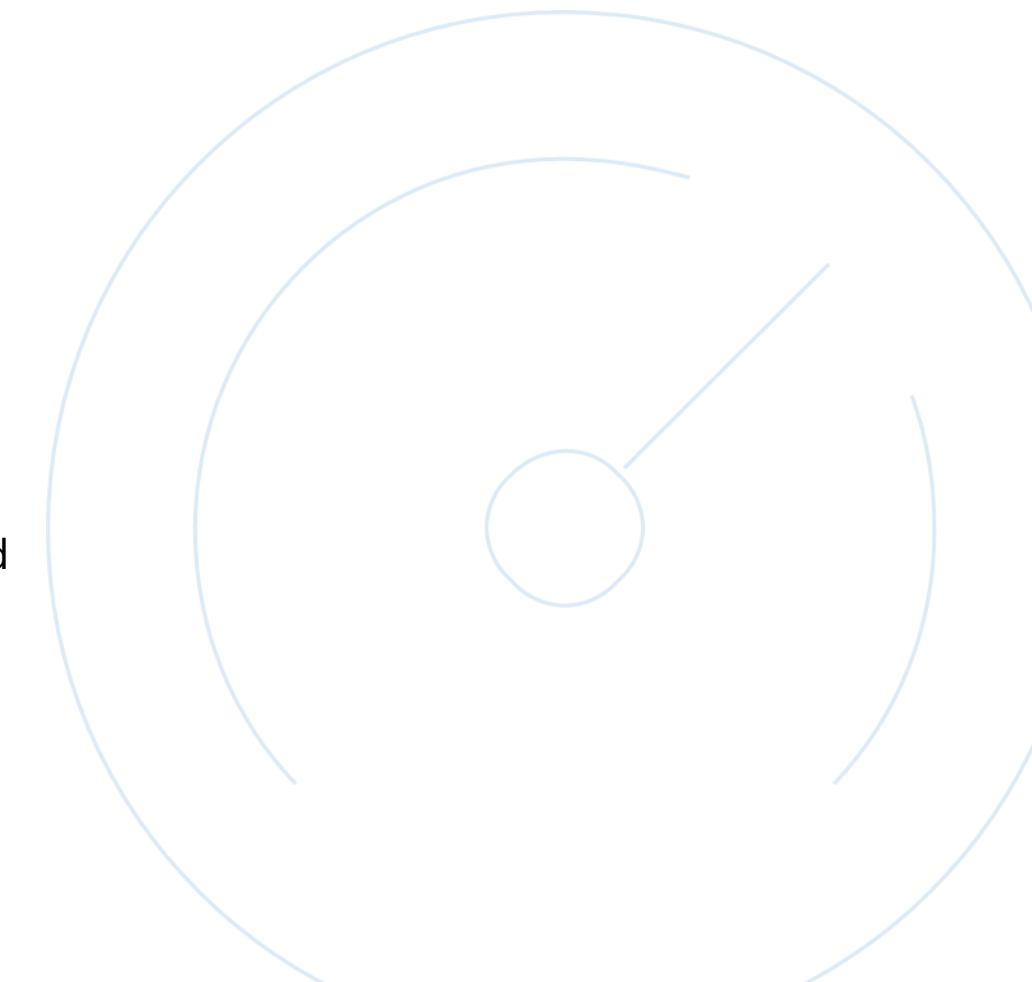
Automatically adds new instances when more capacity is required and removes them when they are no longer needed

[Node auto-repair](#): Public Preview Q2 CY19

Achieve always-on state with self-healing clusters. AKS will initiate a repair process automatically if a health check fails

[Cluster auto-upgrade](#): Public Preview Q2 CY19

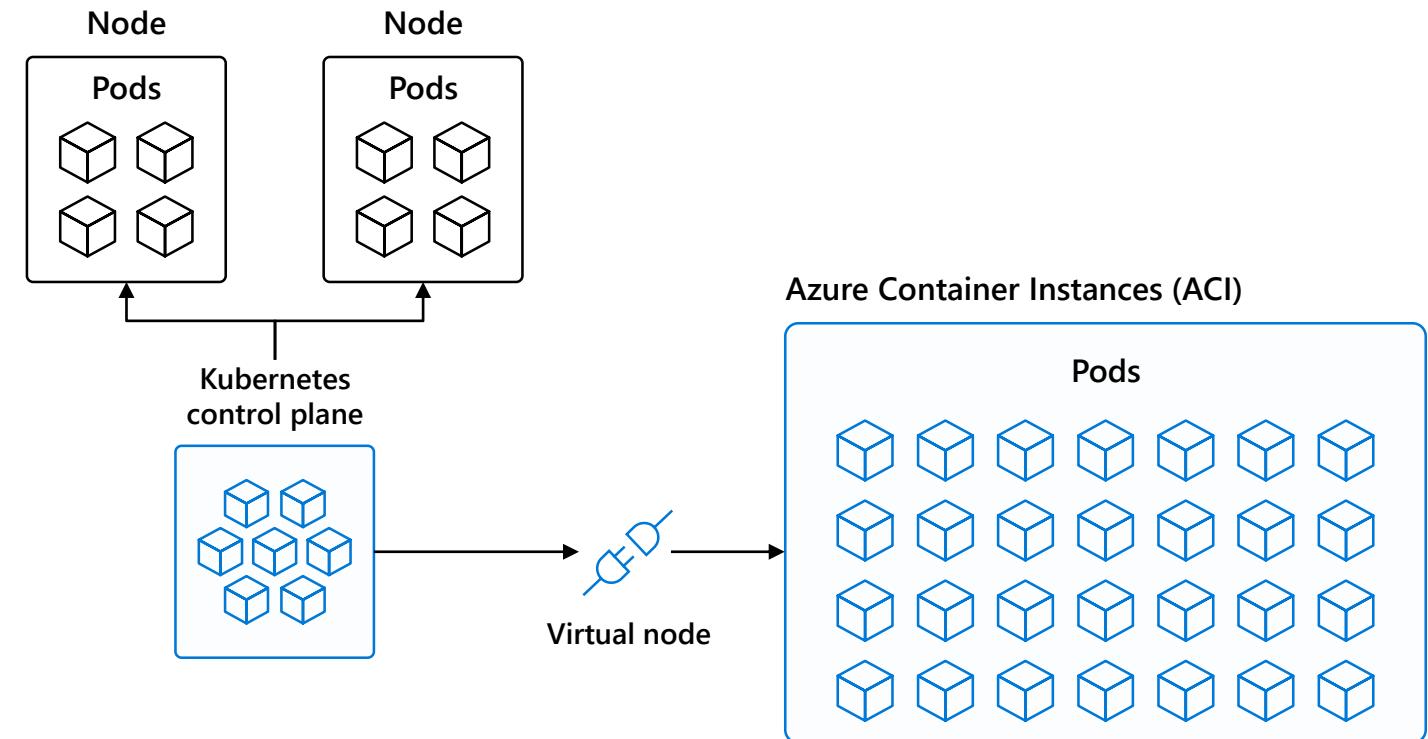
Stay up to date with the latest and greatest of Kubernetes and get patching and security updates automatically



Serverless for AKS

Elastically provision compute capacity
in seconds

No infrastructure to manage



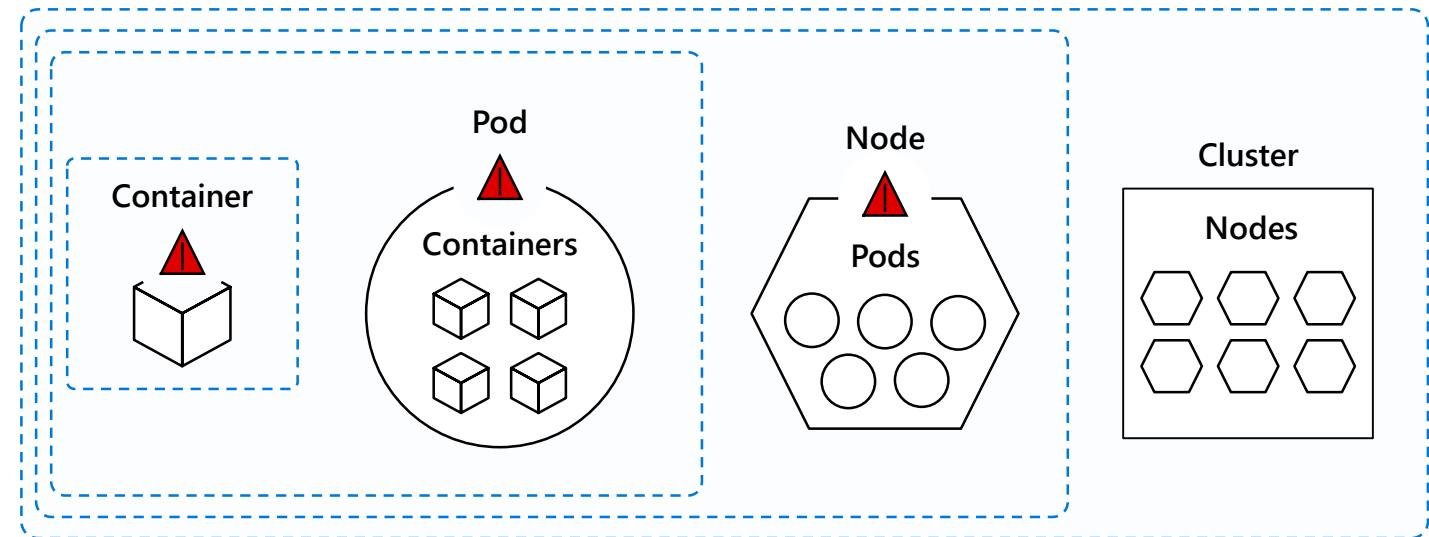
Public Preview since Dec 2018

Cluster autoscaling

Efficiently scale and run apps without downtime —all out of the box

Automatically add or remove instances based on resource utilization

Out-of-the-box cluster autoscaling



Automatically spin up more... → Pods → Containers → Clusters

 = exhausted

Public Preview Q1 CY19

Node auto-repair

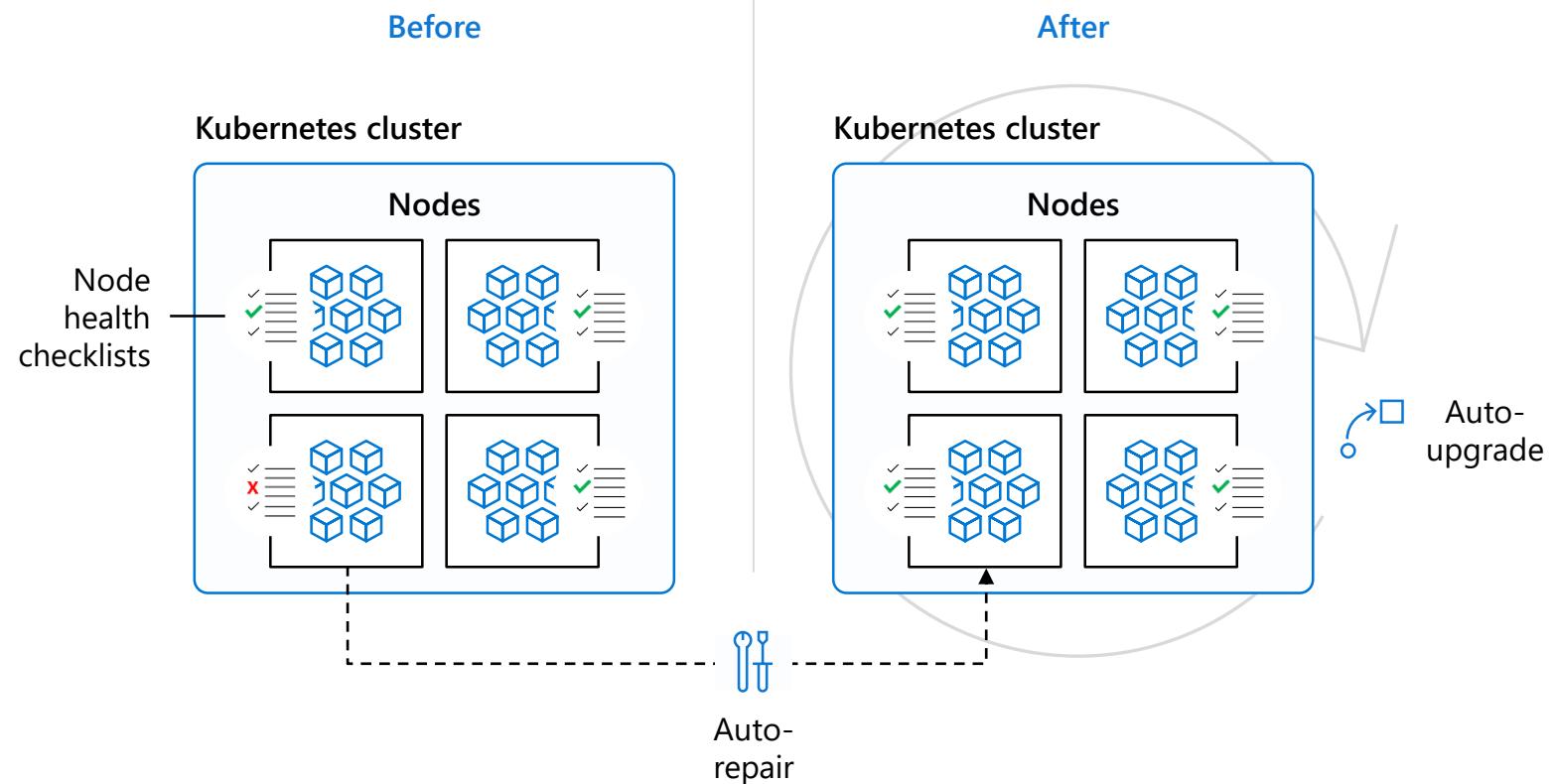
Achieve always-on and healthy state with self-healing clusters

AKS automatically initiates repair process if a node health check fails

Cluster auto-upgrade

Stay up to date with the latest and greatest Kubernetes API

Options to schedule automatic patching and security updates

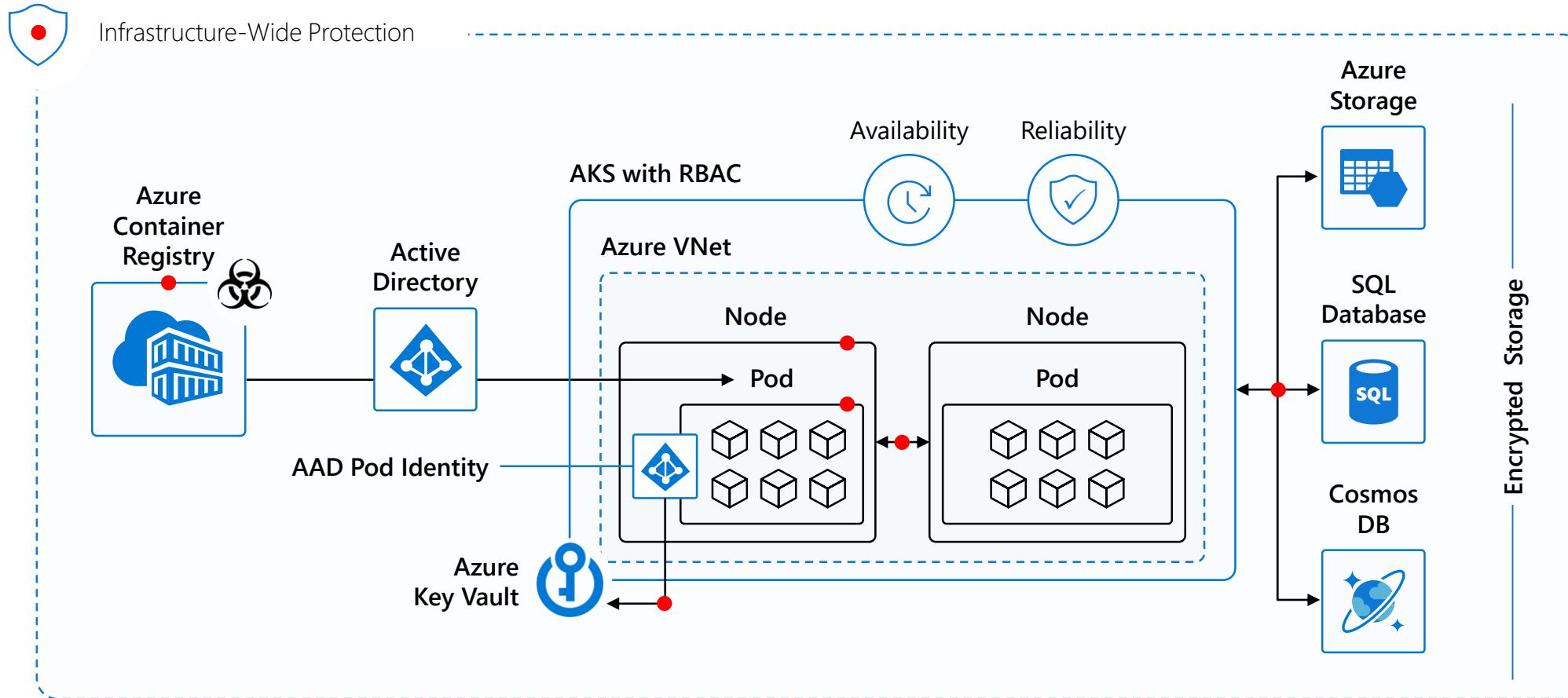


Public Preview Q2 CY19

Enterprise-grade Updates

Enterprise-grade platform

Highly available, reliable service with hardened security and layers of isolation



Enterprise-grade platform

[Control plane audit logs](#): Public Preview released Q1 CY19

Easily investigate suspicious API requests, collect statistics, and create monitoring alerts

[AKS network policy](#): Public Preview released Q1 CY19

Enable better security controls with user-defined network policy

[Pod security policy](#): Public Preview Q1 CY19

Protect sensitive pod specifications by setting up policies to validate requests to pods

[Private cluster](#): Public Preview Q2 CY19

Limit access to the Kubernetes API server to your Azure virtual network

[Multiple node pods](#): Public Preview Q2 CY19

Use different VM sizes per node pool to run a variety of workloads in a single cluster

[Availability Zones](#): Public Preview Q2 CY19

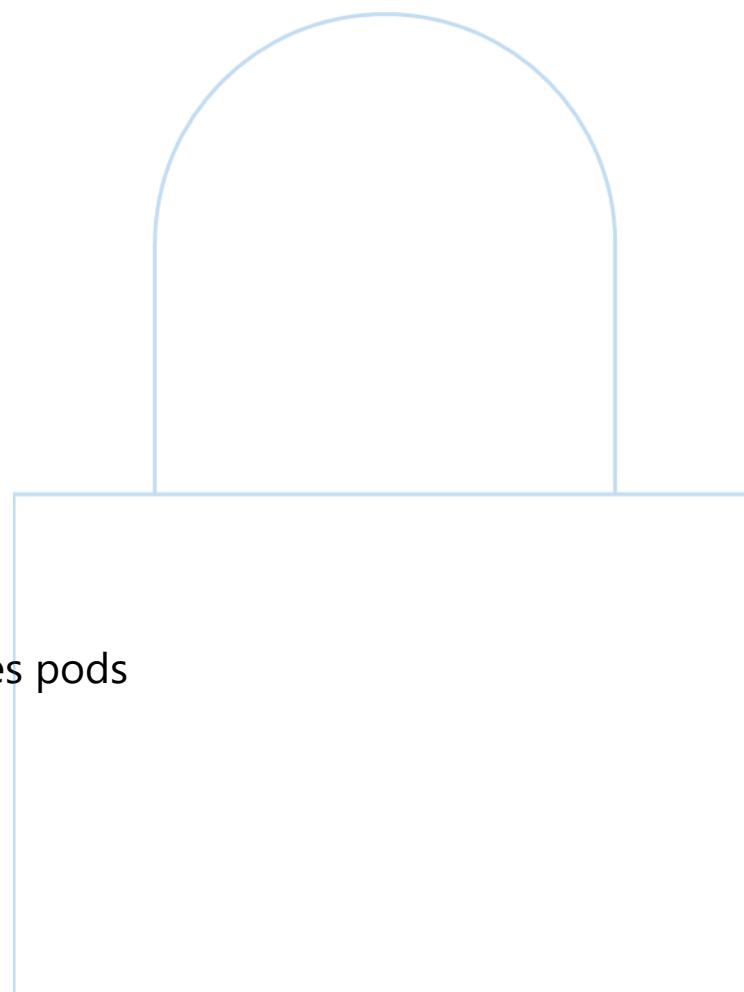
Achieve higher availability and resiliency for both the control plane and worker nodes

[Pod identity](#): Public Preview CY19

Securely communicate with Azure services like Key Vault and Storage by giving Kubernetes pods their own first-class identity in Azure Active Directory

[Key Vault FlexVolume for Kubernetes](#): Public Preview CY19

Centrally store secrets outside of clusters



Control plane audit logs

Easily troubleshoot by reviewing log of API calls made to the Kubernetes control plane

Easily investigate suspicious API requests, collect statistics, and create monitoring alerts for unwanted API calls

The screenshot shows the Azure Log Analytics workspace titled "DefaultWorkspace-19da35d3-9a1a-4f3b-9b9c-3c56ef409565-EUS - Logs". A search query is displayed in the center:

```
defaultworkspace-19da35d3-9a1a-4f3b-9b9c-3c56ef409565-EUS | where Category == "kube-apiserver" | where log_s contains "pods/nginx" | project log_s
```

The results pane shows a table of log entries from the last 24 hours. One entry is visible:

log_s
1025 22:47:33.141060 1 wwp.go:42] GET /api/v1/namespaces/default/pods/nginx: [3.024572ms] 200 [[kubelite/v1.9.11 (linux/amd64) kubernetes/12feeb6] 172.31...]

Below the table, there are navigation controls for pages and items per page.

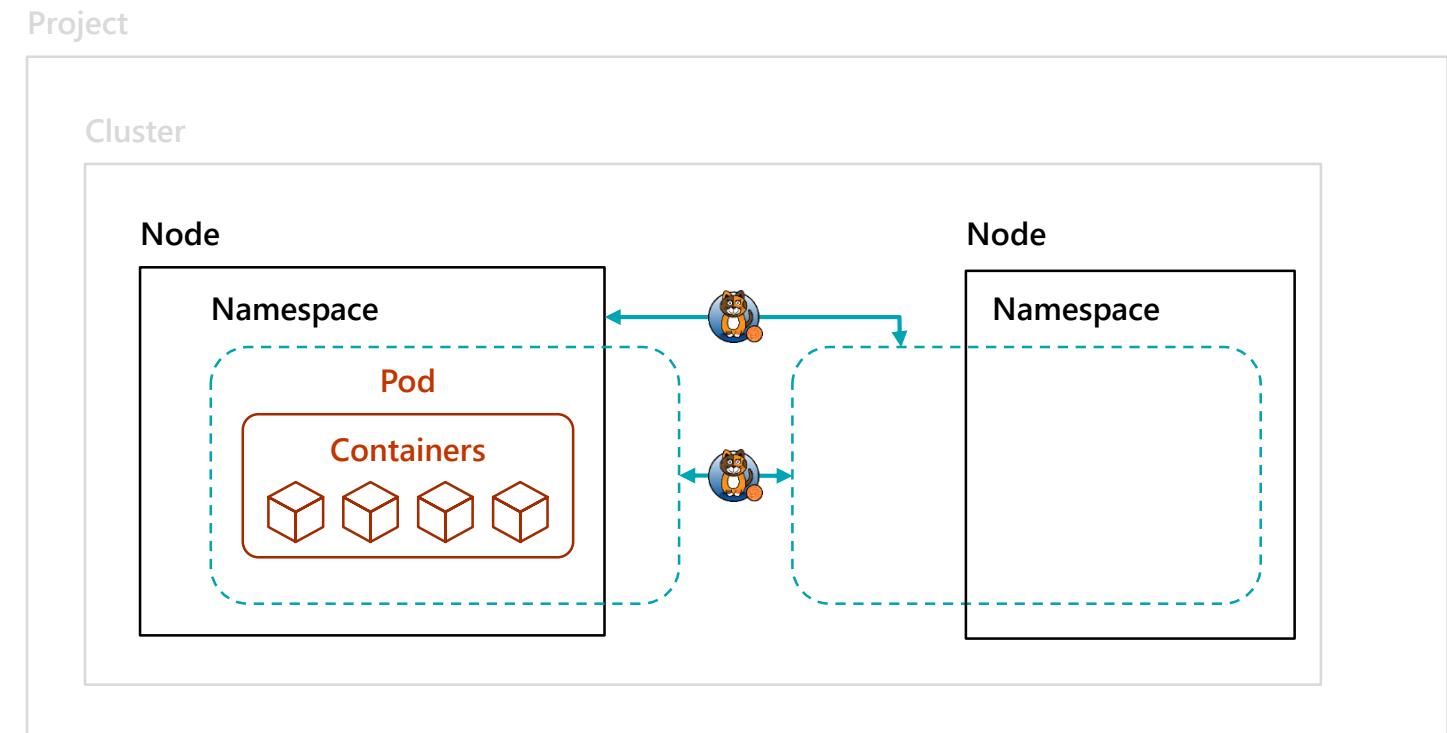
Public Preview released Q1 CY19

AKS network policy

Secure communication paths between namespaces and nodes

Better controls with user-defined network policy

All powered by Calico , an open source project



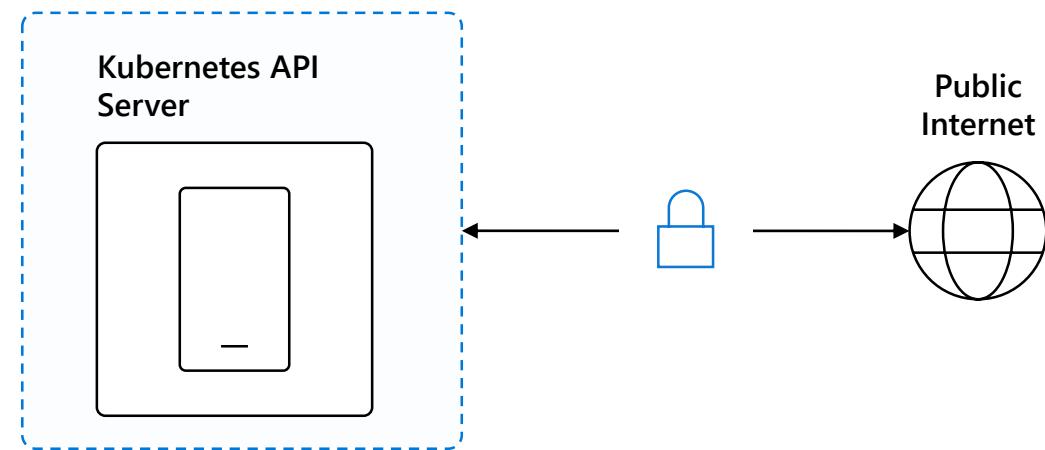
Public Preview released Q1 CY19

Private cluster

Isolate your workload from the Internet

Limit access to the Kubernetes API server, also known as the control plane, to your Azure virtual network

Azure private network



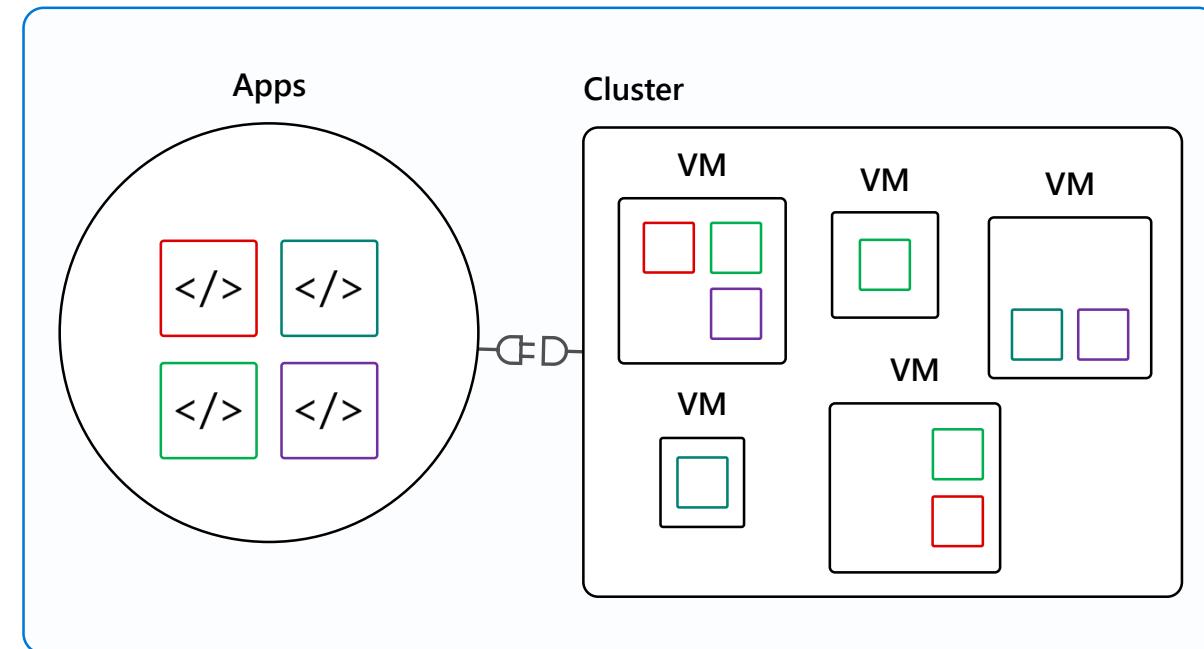
Public Preview Q2 CY19

Multiple node pools

Use different VM(s) within a cluster based on application workload needs

Enhanced compute resource management and efficient resource utilization

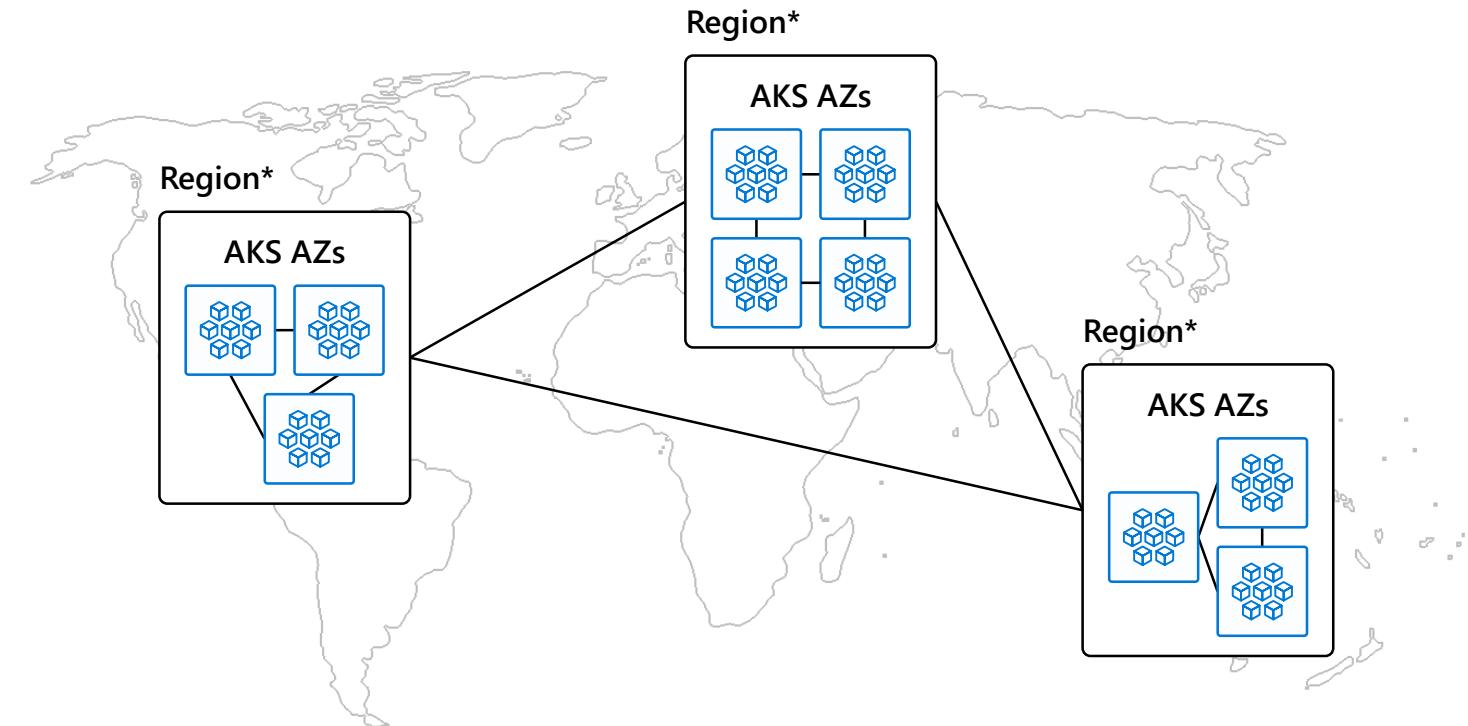
AKS



Availability Zones

Achieve higher availability and resiliency for both the control plane and worker nodes

Protect applications and data from datacenter failures with redundancies across availability zones



Public Preview Q2 CY19

*Regions are for illustrative purposes only. Actual AKS AZ regions may vary.

Pod security policy

Protect sensitive pod specifications and validate pod requests with policies

Pod identity

Securely communicate with a variety of Azure services, like Key Vault and Storage

Give Kubernetes pods their own first-class identity in Azure Active Directory

No additional keys or secrets to manage

[Pod security policy](#): Public Preview Q1 CY19

[Pod identity](#): Public Preview CY19

Project

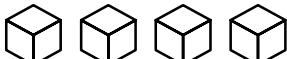
Cluster

Node

Namespace

Pod

Containers



Secure communication
between Pods and
other Azure services



Azure
Key Vault



Azure
Storage

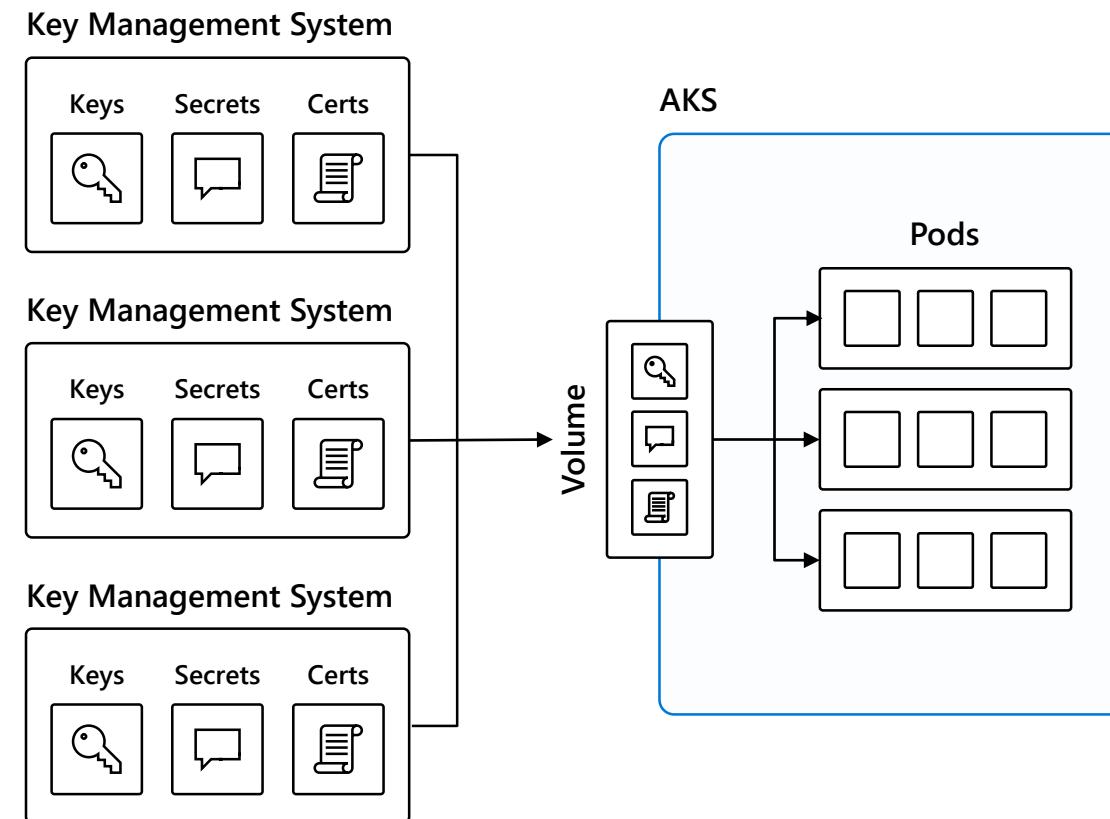
Validate requests
with policies

Key Vault FlexVolume for Kubernetes

Meet rigorous compliance requirements with secrets centrally stored outside of clusters

Access application-specific keys, secrets, and certificates natively within Kubernetes from Azure Key Vault

Mount Azure Key Vault stores as flex volumes to Kubernetes pods



Public Preview CY19

