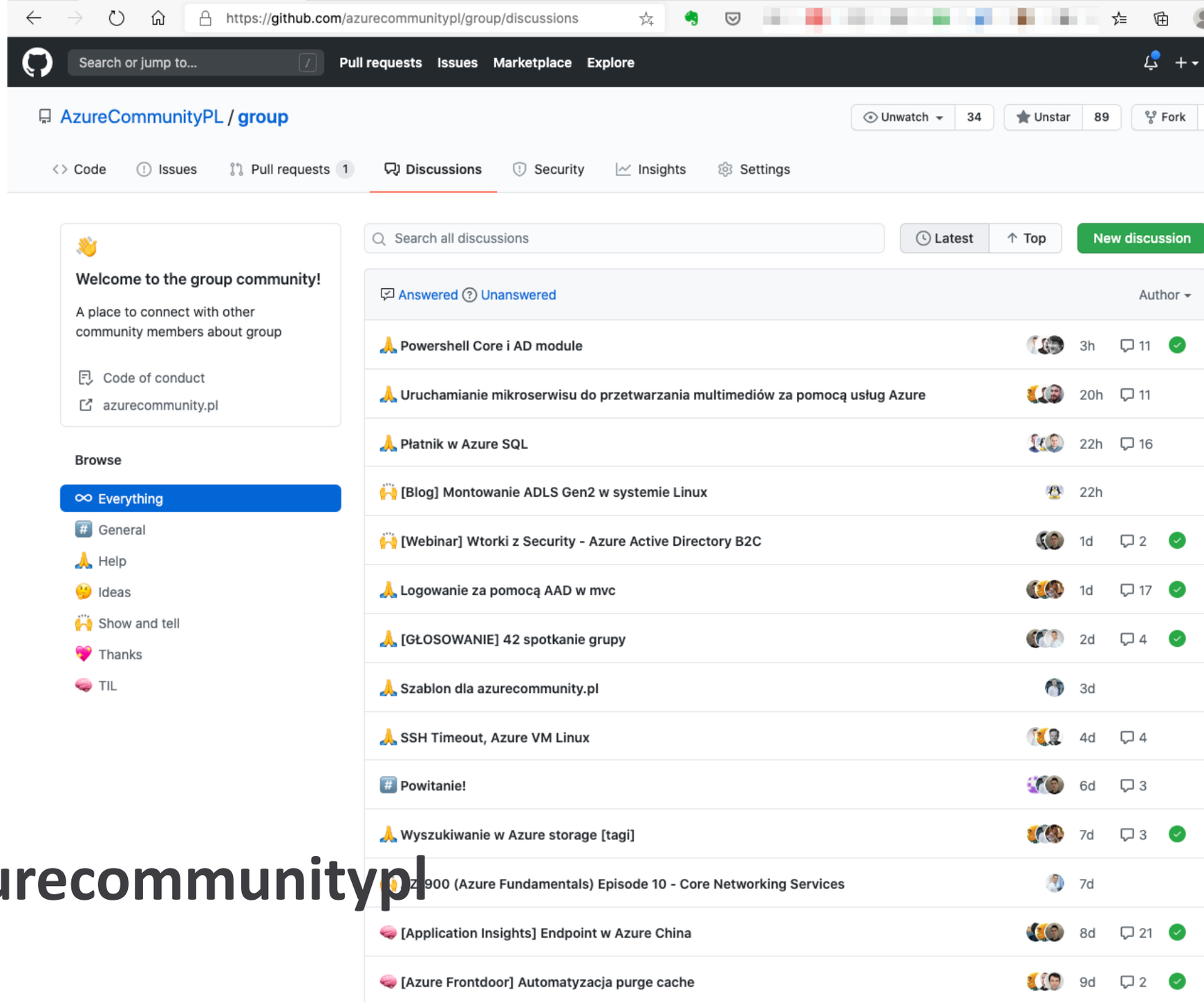
Kubernetes on Azure

Łukasz Kałużny, Microsoft Azure MVP

patoarchitekci.io, poznajkubernetes.pl

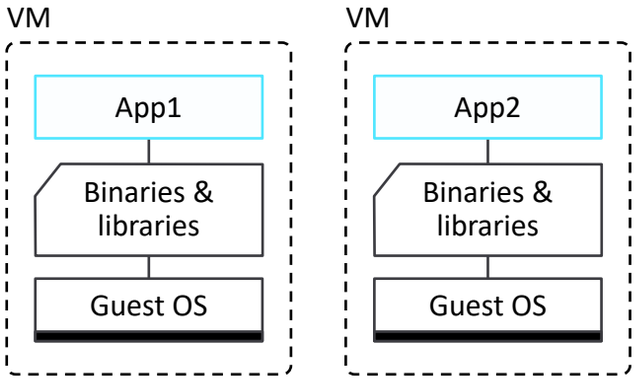
AzureCommunityPL



https://aka.ms/azurecommunitypl

Containers overview

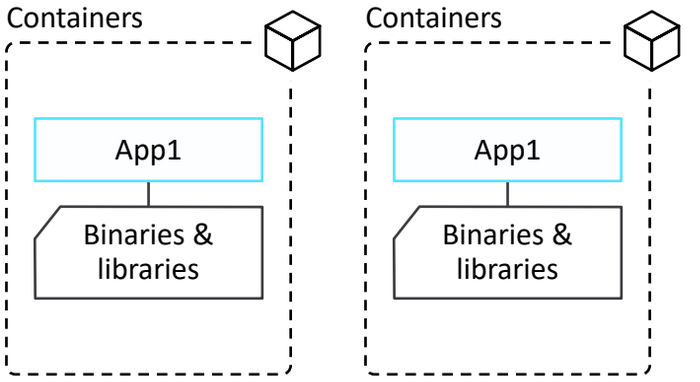
What is a container?



Virtual machines

Virtualize the hardware

VMs as units of scaling



Containers

Virtualize the operating system

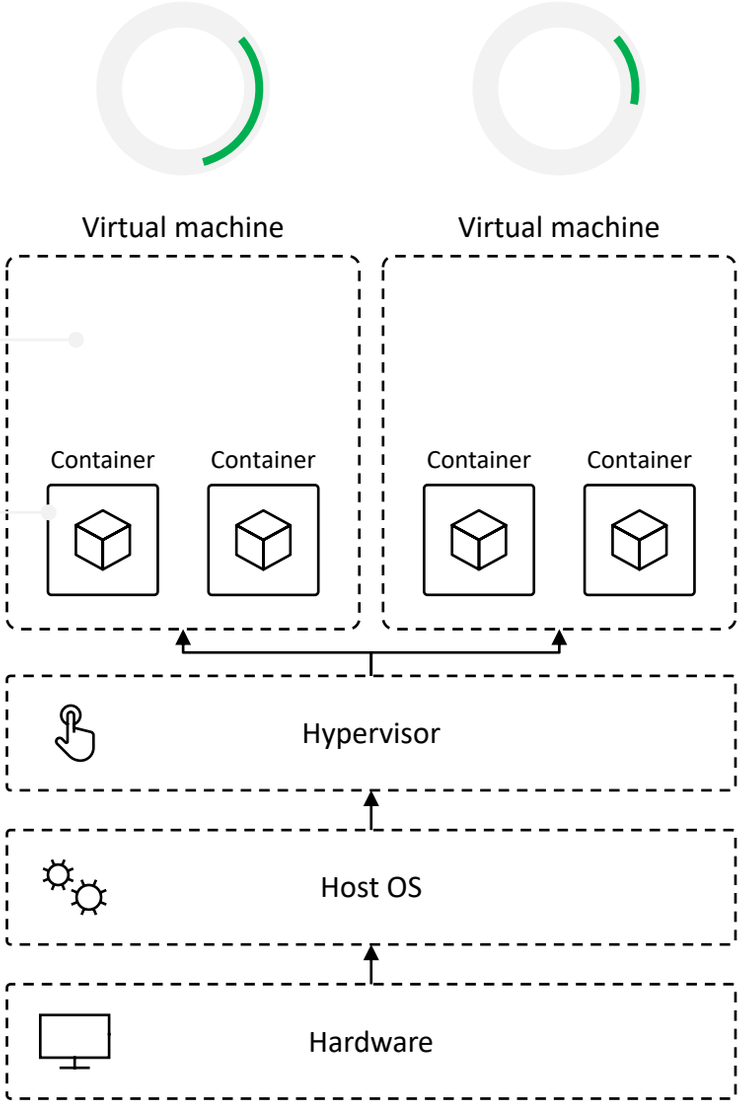
Applications as units of scaling

Traditional virtualized environment

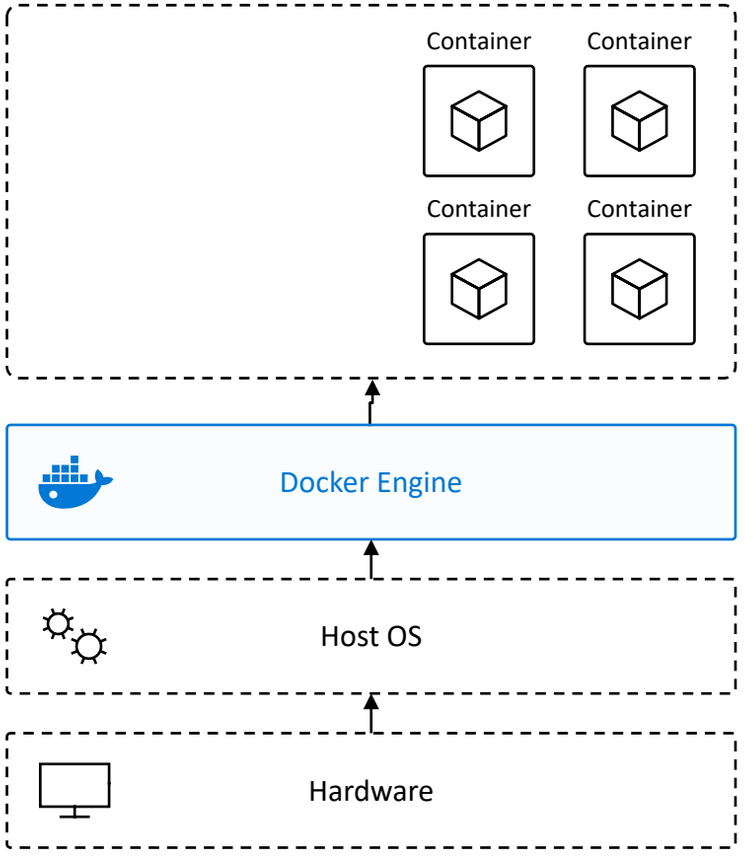
From dev to production agility across development and operations teams

Low utilization of resources

Containerization of applications and their dependencies for portability



Advantages of a containerized environment



The benefits of using containers

Agility

+

Ship apps faster

Portability

+

Easily move workloads

Density

+

Achieve resource efficiency

Rapid scale

+

Scale easily to meet demand

The elements of orchestration

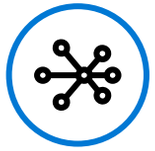
Scheduling

Affinity/anti-affinity

Health monitoring

Failover

Scaling

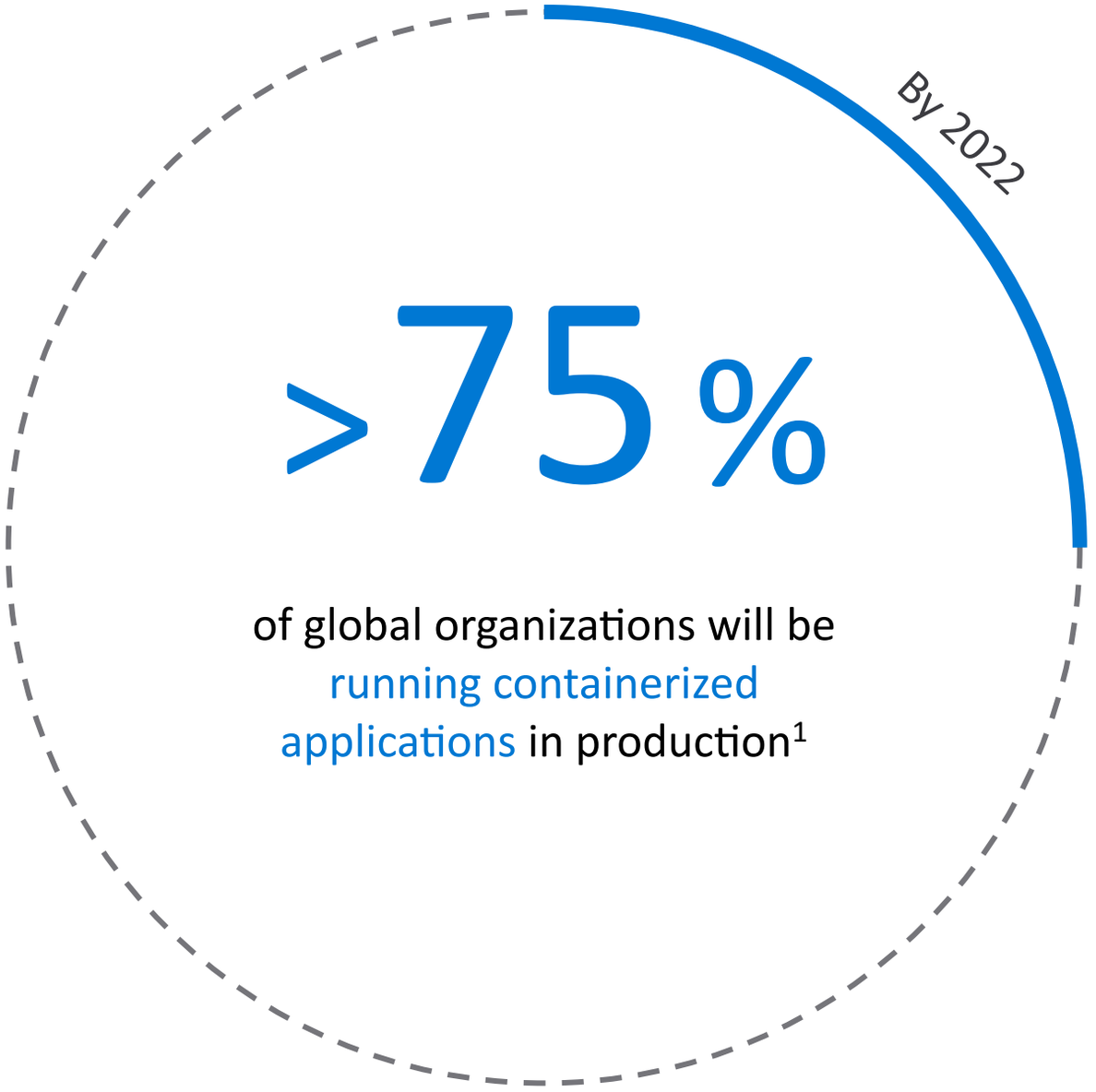


Networking

Service discovery

Coordinated app upgrades

Kubernetes momentum


BY 2022
>75%
of global organizations will be
running containerized
applications in production[1]

[1]Gartner.

What's behind the growth?

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

It's widely used

Kubernetes is in production for global companies across industries[1]

Capital One	eBay	SAP
New York Times	Pokémon Go	Spotify

It's vendor-neutral

A variety of cloud providers offer robust Kubernetes support

Azure	AWS
VMWare	Red Hat

It's community-supported

There's a huge community of active contributors supporting Kubernetes[3]

35,000 contributors since 2016	1.1 million contributions since 2016

[1]Kubernetes.io. "Kubernetes User Case Studies." [2]CNCF. "Kubernetes Is First…" [3]CNCF. K8s Project Journey

Kubernetes is built and maintained by the community

Kubernetes collects wisdom, code, and efforts from hundreds of corporate contributors and thousands of individual contributors

148,000
commits

35,000
contributors

#1
GitHub project

Microsoft is part of this vibrant community and leads in the associated committees to help shape the future of Kubernetes and its ecosystem

CNCF
platinum member

CNCF
technical oversight committee

CNCF
governing board

Kubernetes
steering committee

Linux Foundation
board member

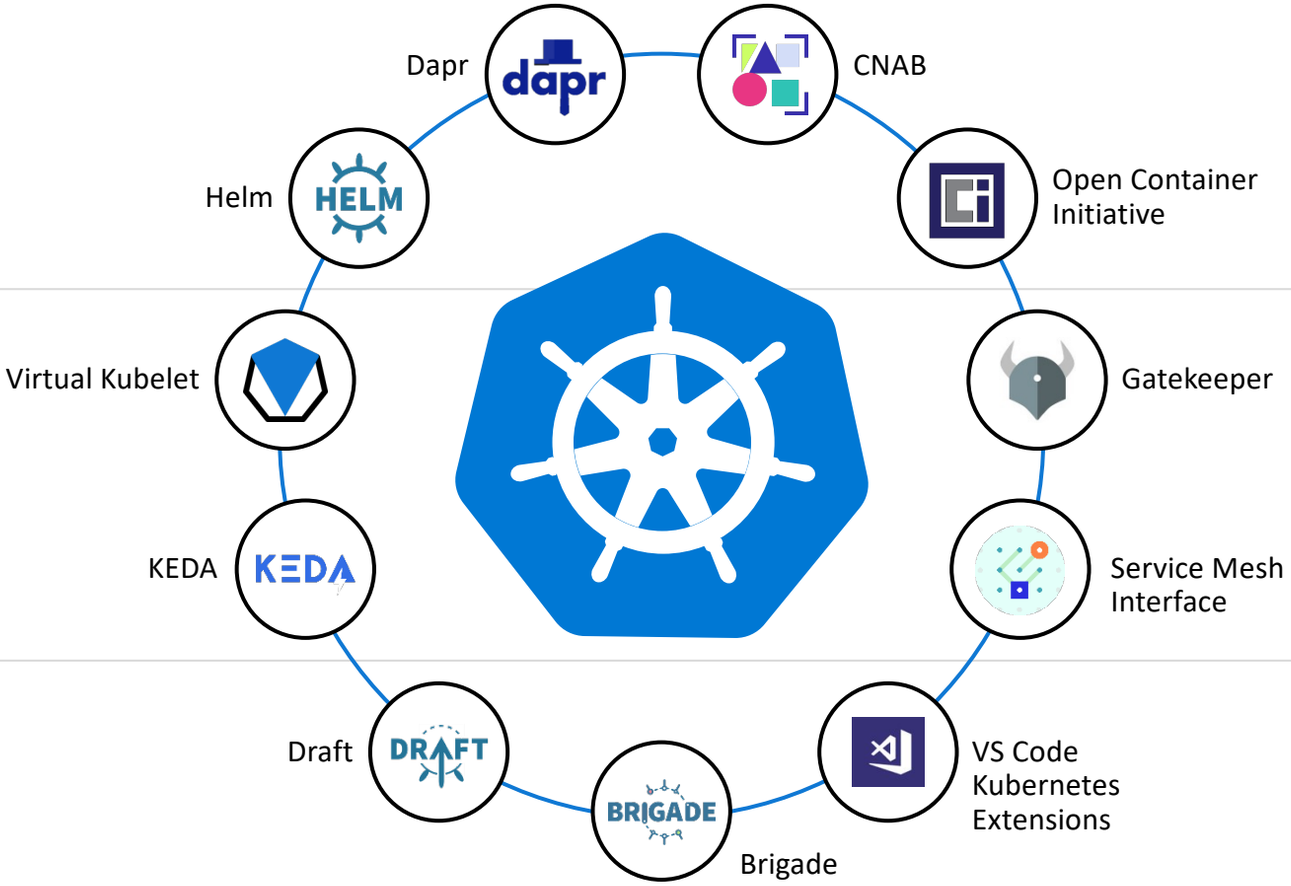
AKS is certified Kubernetes conformant, ensuring portability and interoperability of your container workloads

Microsoft contributions to the community

Packaging & distribution

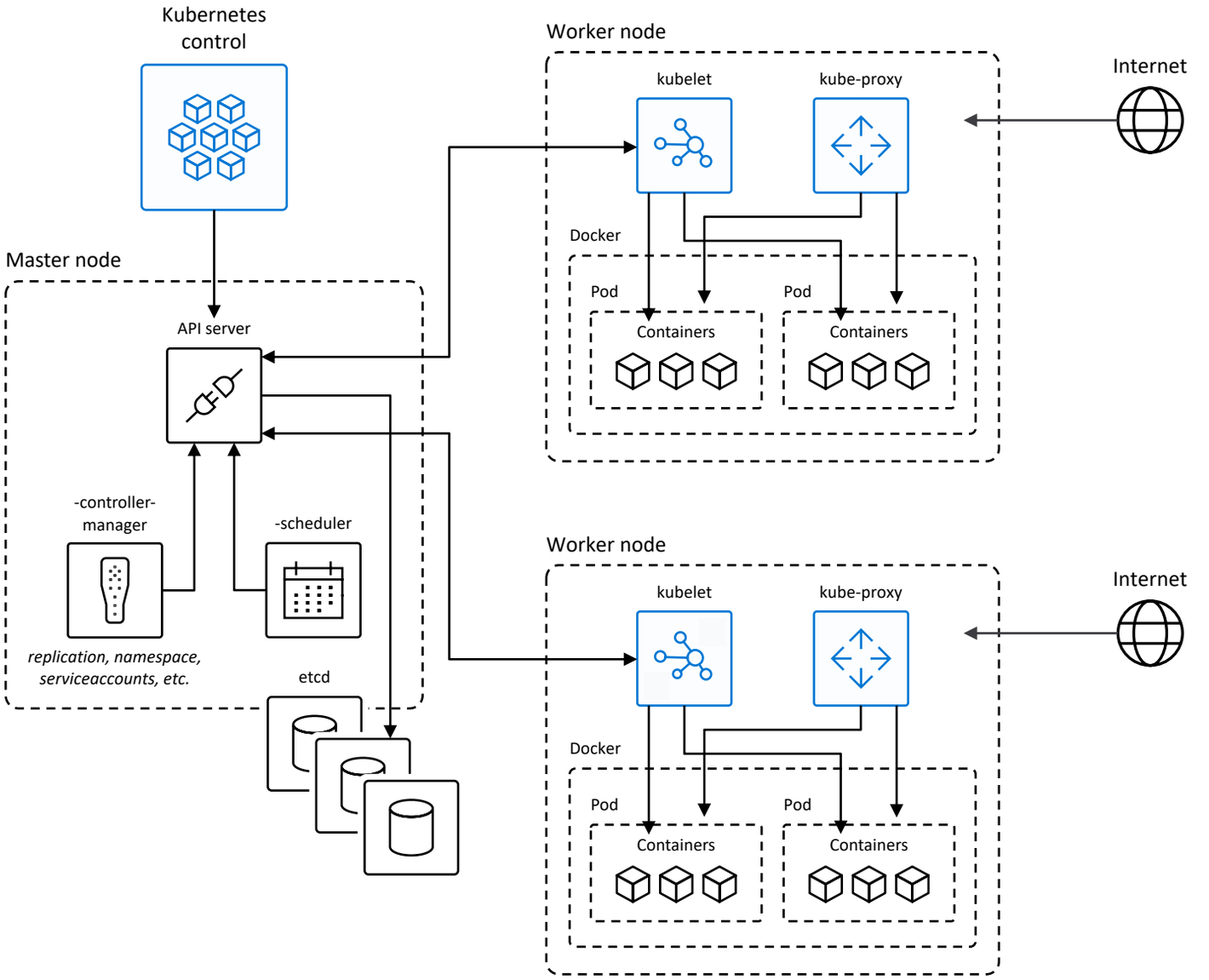
Scalability & control

Kubernetes developer tooling



How Kubernetes works

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

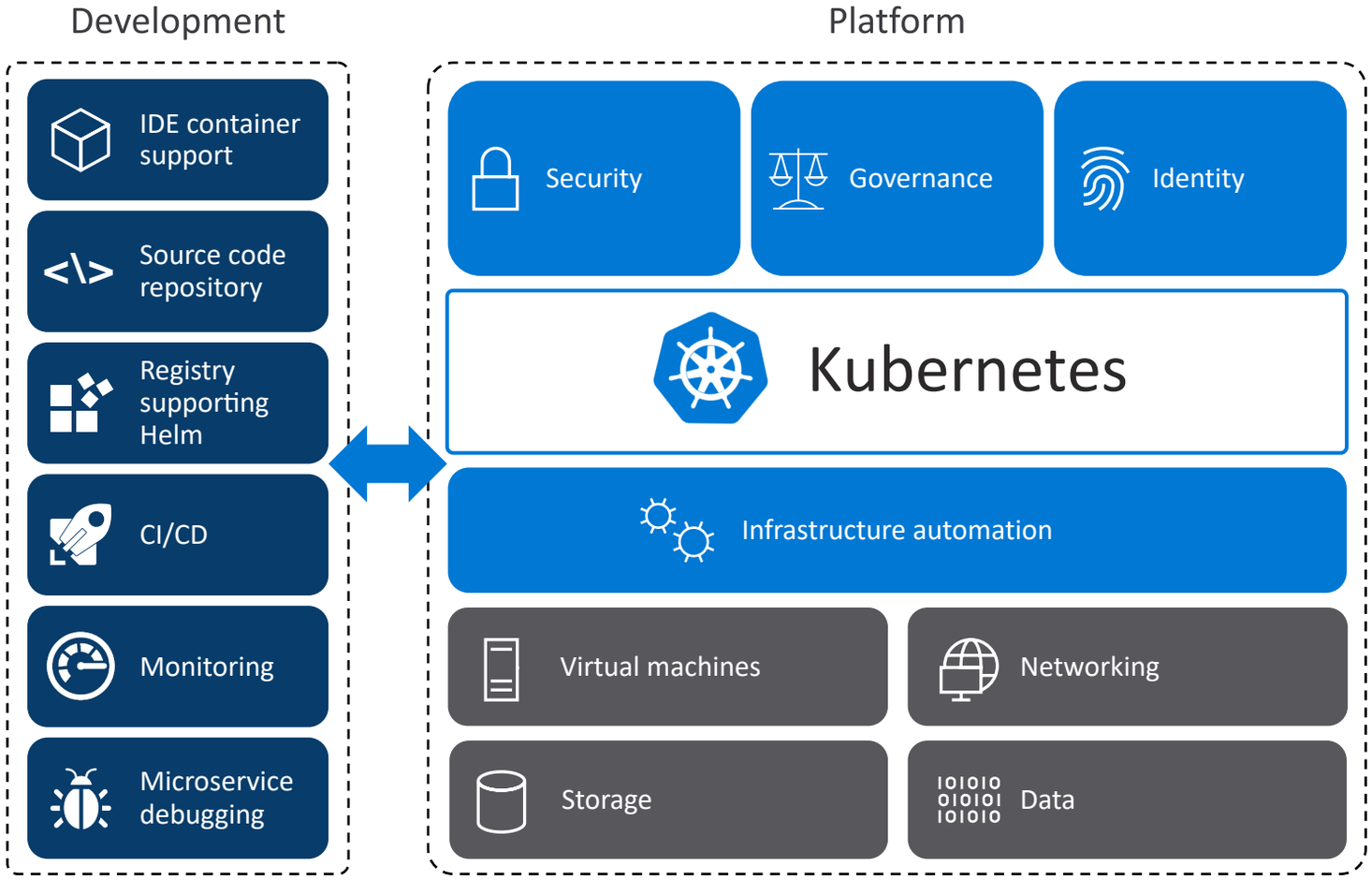


Kubernetes on its own is not enough

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

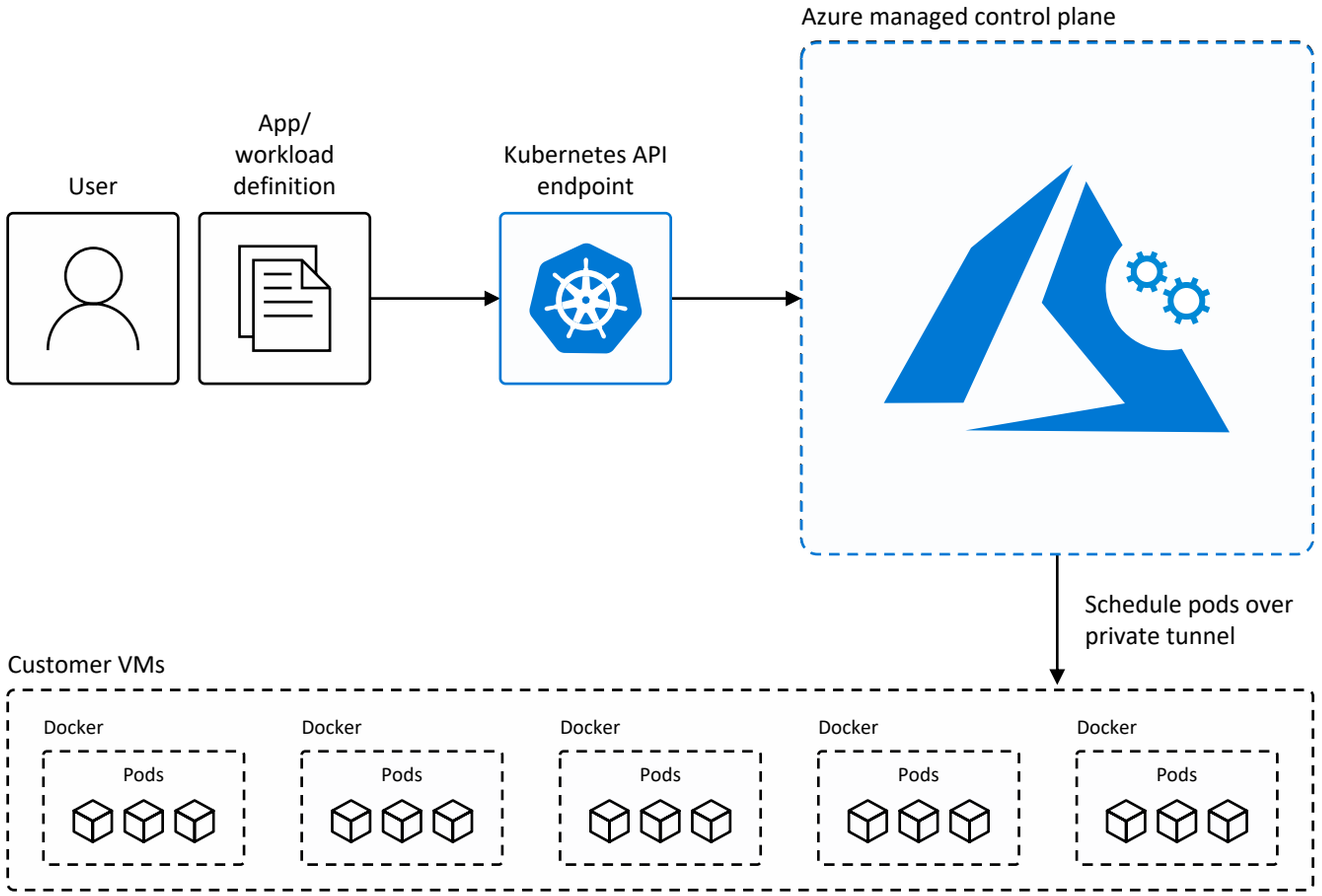


Increase operational efficiency

Focus on your containers and code, not the plumbing of them

Responsibilities	DIY with Kubernetes	Managed Kubernetes on Azure
Containerization		
Application iteration, debugging		
CI/CD		
Provisioning, upgrades, patches		
Reliability availability		
Scaling		
Monitoring and logging		

Customer Microsoft



Kubernetes on Azure

Enterprise-grade by design

Built-in best practices

Proactive and actionable recommendations, based on knowledge from thousands of enterprise engagements

Multi-layer security

Hardened security and layers of isolation across compute resources, data, and networking

Unified management

Consistent configuration and governance across environments

Kubernetes on Azure | Enterprise-grade by design

Development tools

- Visual Studio Code
- GitHub
- Azure Dev Spaces
- Azure Container Registry
- Azure Monitor

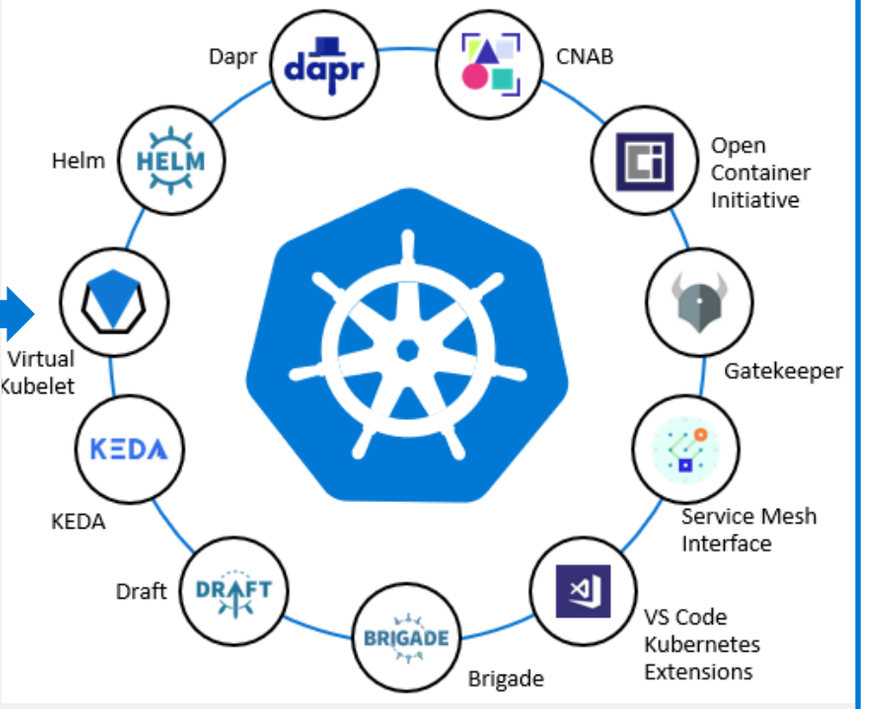
Platform

- Active Directory
- Azure Policy
- Security Center
- Key Vault
- Azure Advisor

- Azure Kubernetes Service
- Azure Container Instances
- Azure Red Hat OpenShift

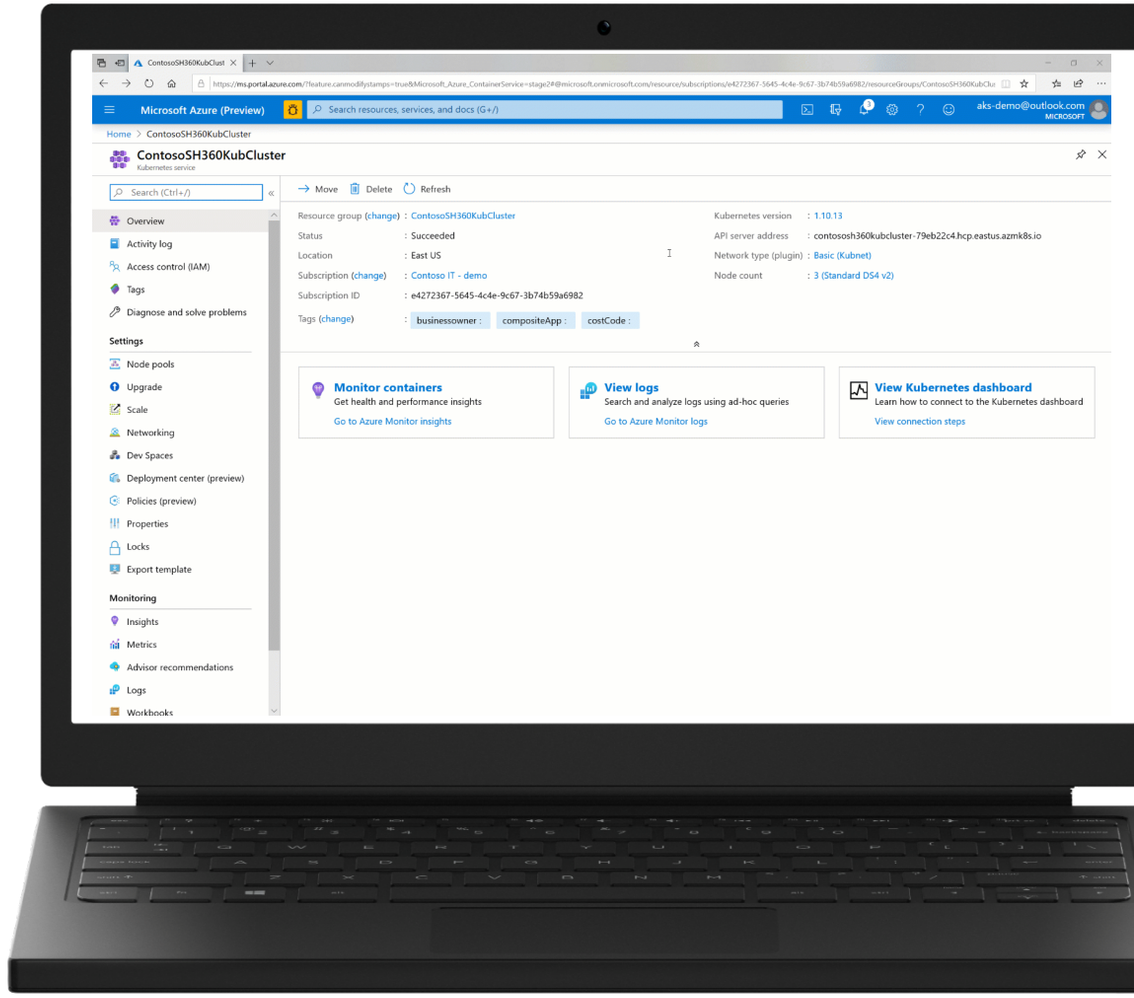
Azure Arc
Management across environments

Community



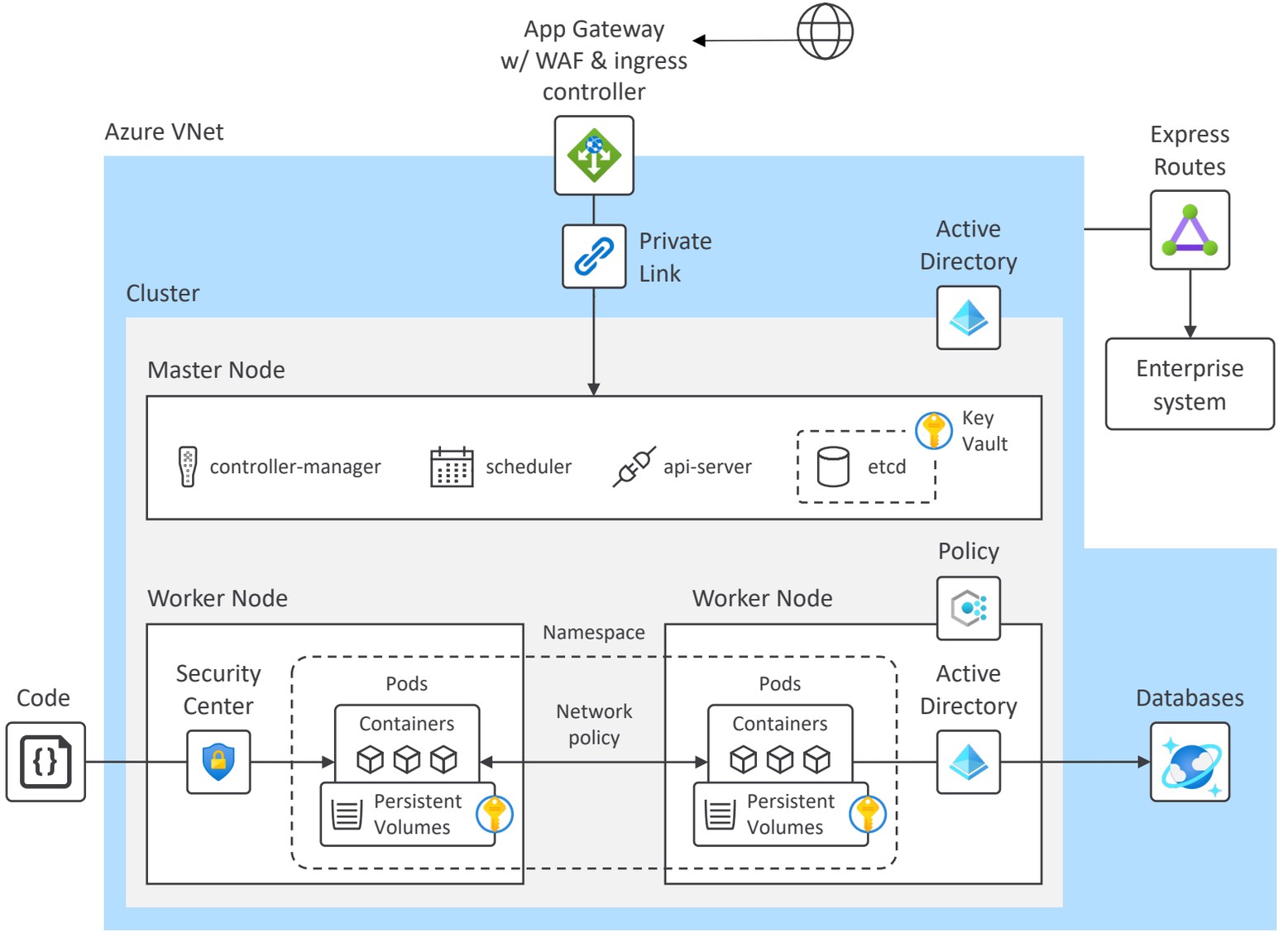
Built-in best practices

- Proactive and actionable recommendations from Azure Advisor based on your configuration and usage telemetry
- Grounded with knowledge from thousands of customer engagements
- Improve the performance, availability, and security of your cluster before there's a problem



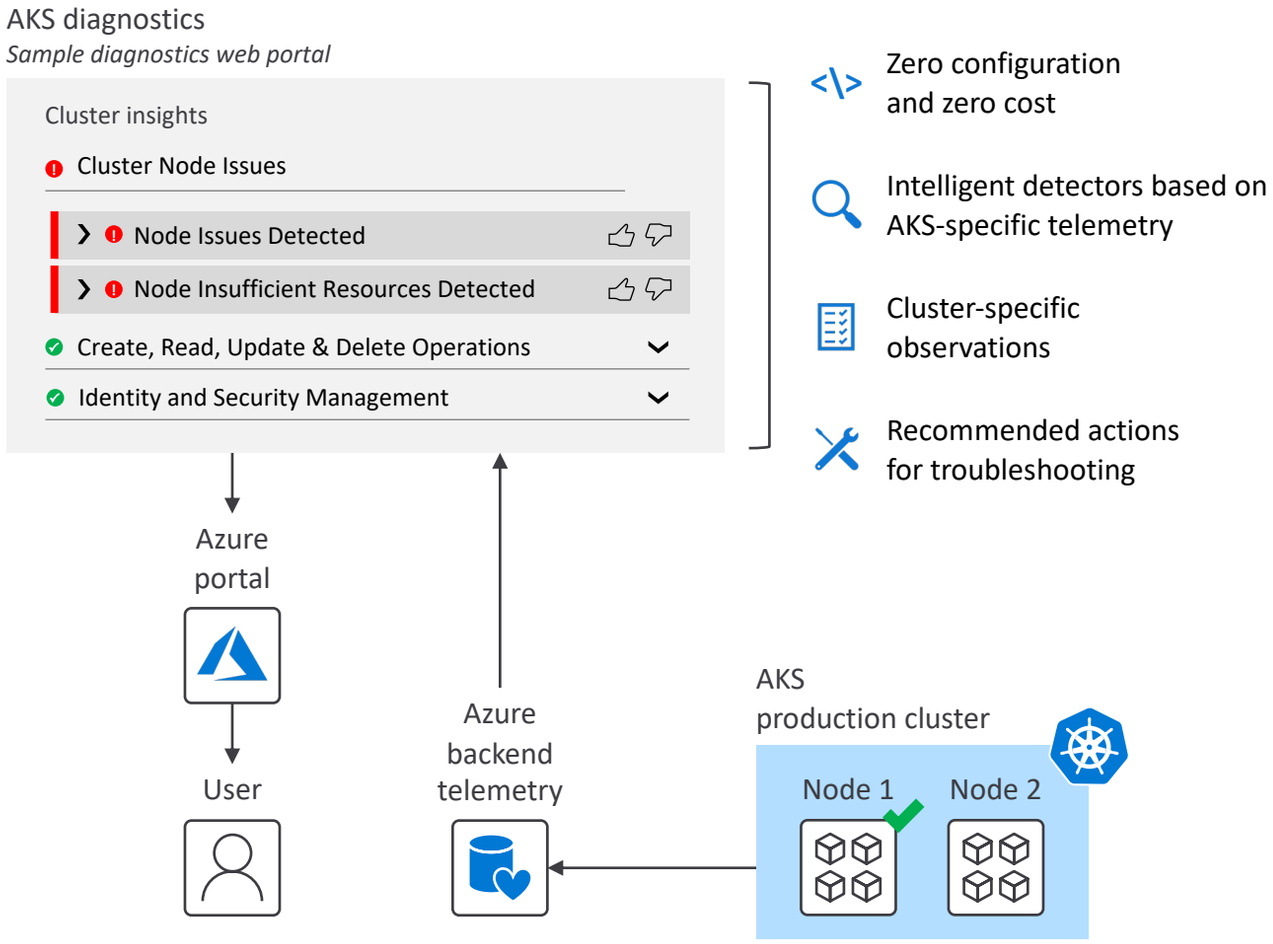
Multi-layer security

- Enforce compliance rules to your cluster and CI/CD pipeline consistently with Azure Policy
- Apply fine-grained identity and access control using Azure Active Directory
- Encrypt the disks used in your AKS cluster using your own keys, stored in Azure Key Vault
- Gain unmatched security management, intelligent threat detection, and actionable recommendations with Azure Security Center integration
- Interact with the Kubernetes API server as a private endpoint using Azure Private Link



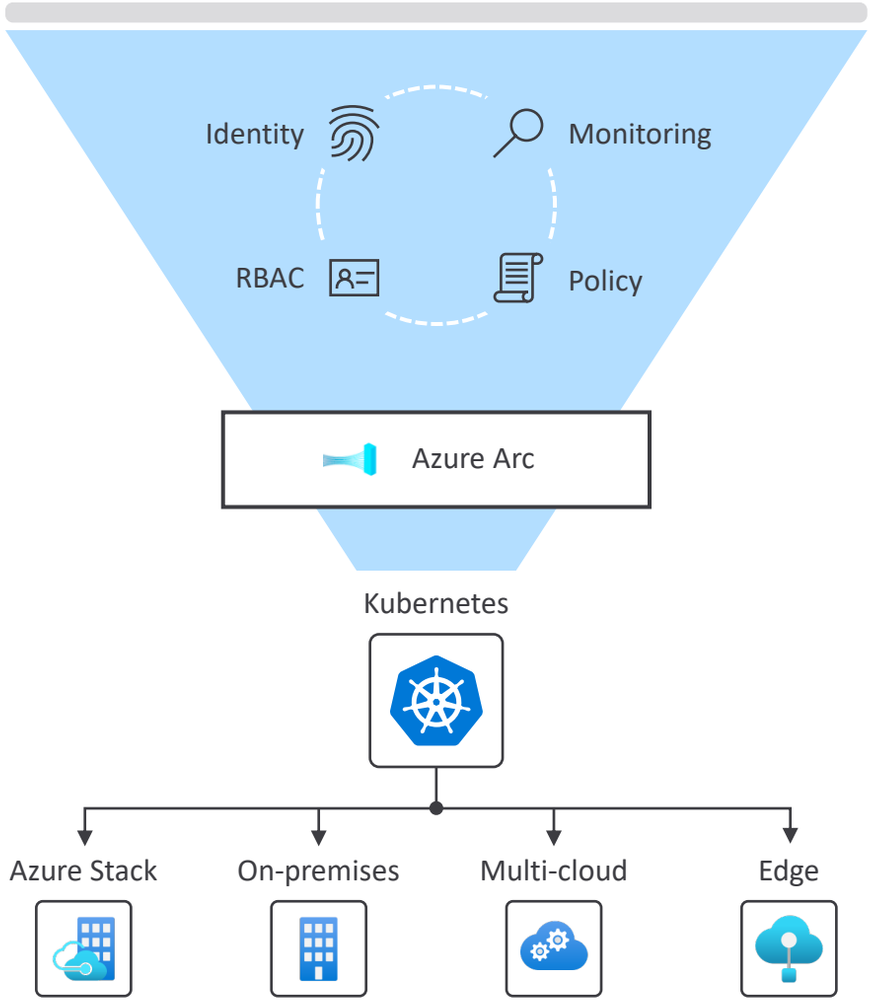
Enterprise support

- 24x7x365 on-call support backed by Kubernetes certified experts
- >100 partners worldwide offering Kubernetes best practices
- Self-service troubleshooting with tools used by Microsoft customer support team
- Faster resolution of common issues with an intelligent, self-diagnostic experience right in the portal



Unified management

- Central inventory and monitoring of the sprawling assets running anywhere from on-premises to edge
- Consistently apply policies, role-based-access-controls (RBAC) for at-scale governance
- Deploy Kubernetes resources to all clusters using a GitOps-based workflow



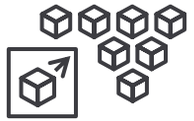
Work how you want with opensource tools and APIs

	Development	DevOps	Monitoring	Networking	Storage	Security
Take advantage of services and tools in the Kubernetes ecosystem	HELM DRAFT Virtual kubelet CNAB	JFrog HashiCorp BRIGADE	Prometheus fluentd Grafana DATADOG	CNI Networking TIGERA	MAPR portworx ceph OpenEBS GlusterFS STORAGEOS	Twistlock aqua Qualys.
Leverage 100+ turn-key Azure services	Visual Studio Code GitHub	Azure Pipelines ARM Container Registry	Azure Monitor	Azure VNET Service Mesh Interface	Azure Storage Azure Cosmos DB	Azure Policy AAD ASC Key Vault

Top scenarios

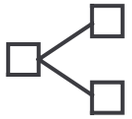
Top scenarios for Kubernetes on Azure

Lift and shift
to containers



Cost saving
without refactoring
your app

Microservices



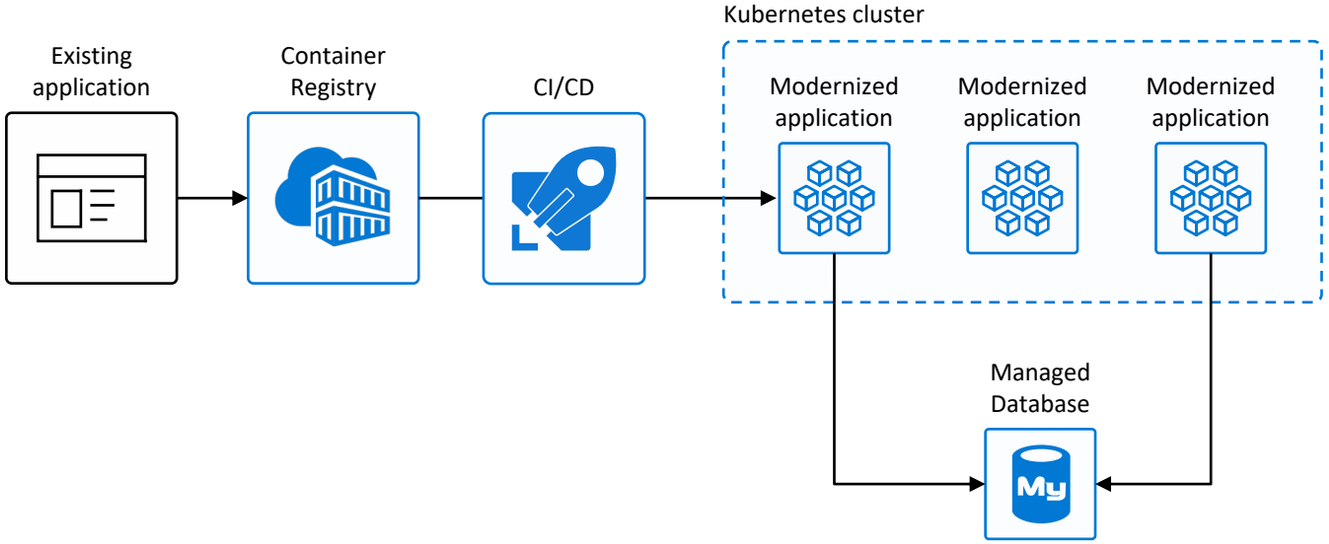
Agility
Faster application
development

Secure
DevOps

Automation
Deliver code faster and
securely at scale

App modernization without code changes

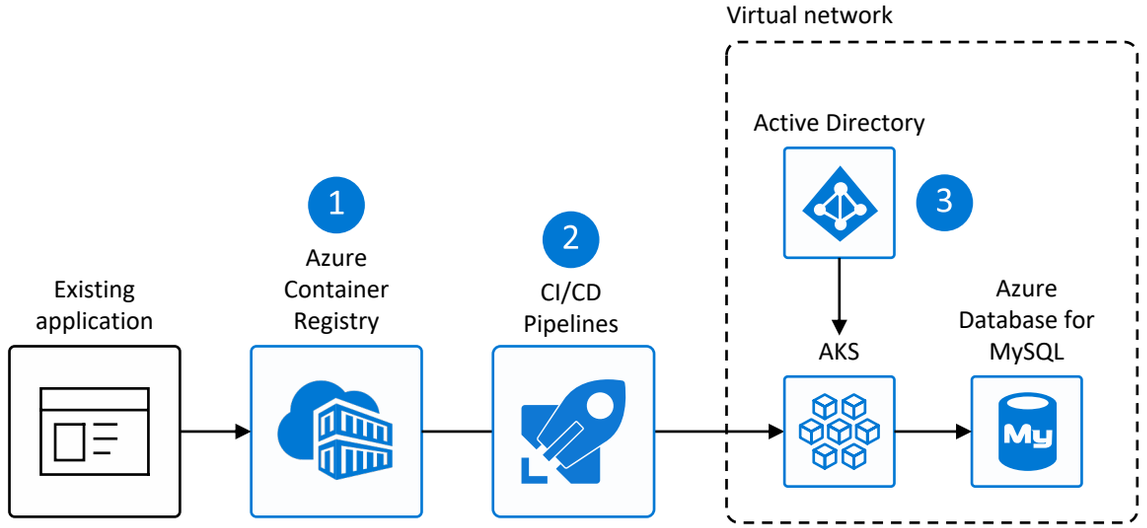
- Speed application deployments by using container technology
- Defend against infrastructure failures with container orchestration
- Increase agility with continuous integration and continuous delivery



App modernization without code changes

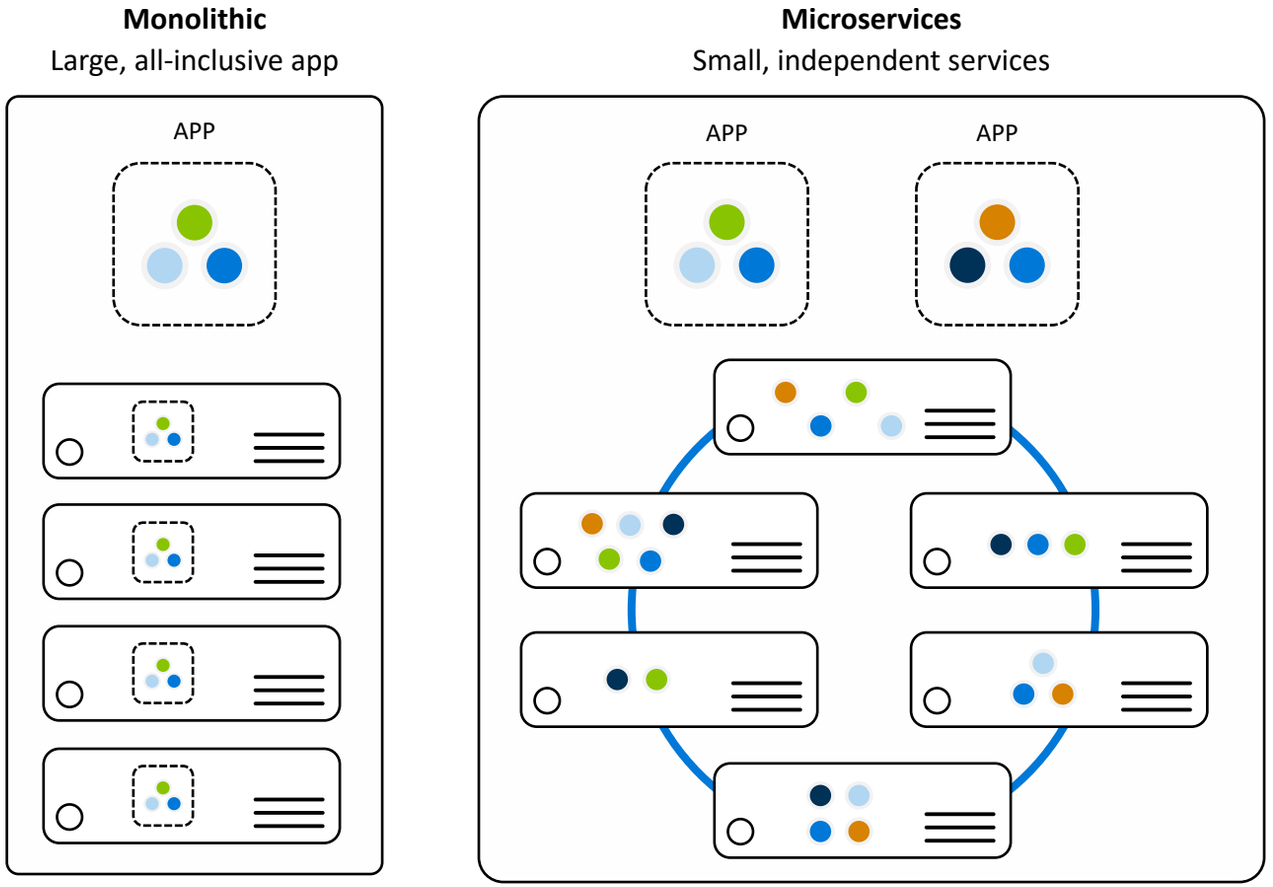
Capabilities

1. Use Azure Container Registry to store container images and Helm charts for your modernized applications, replicated globally for low latency image serving
2. Integrate AKS with Azure Pipelines or other Kubernetes ecosystem tooling to enable continuous integration/continuous delivery (CI/CD)
3. Enhance security with Azure Active Directory and RBAC to control access to AKS resources



Microservices: for faster app development

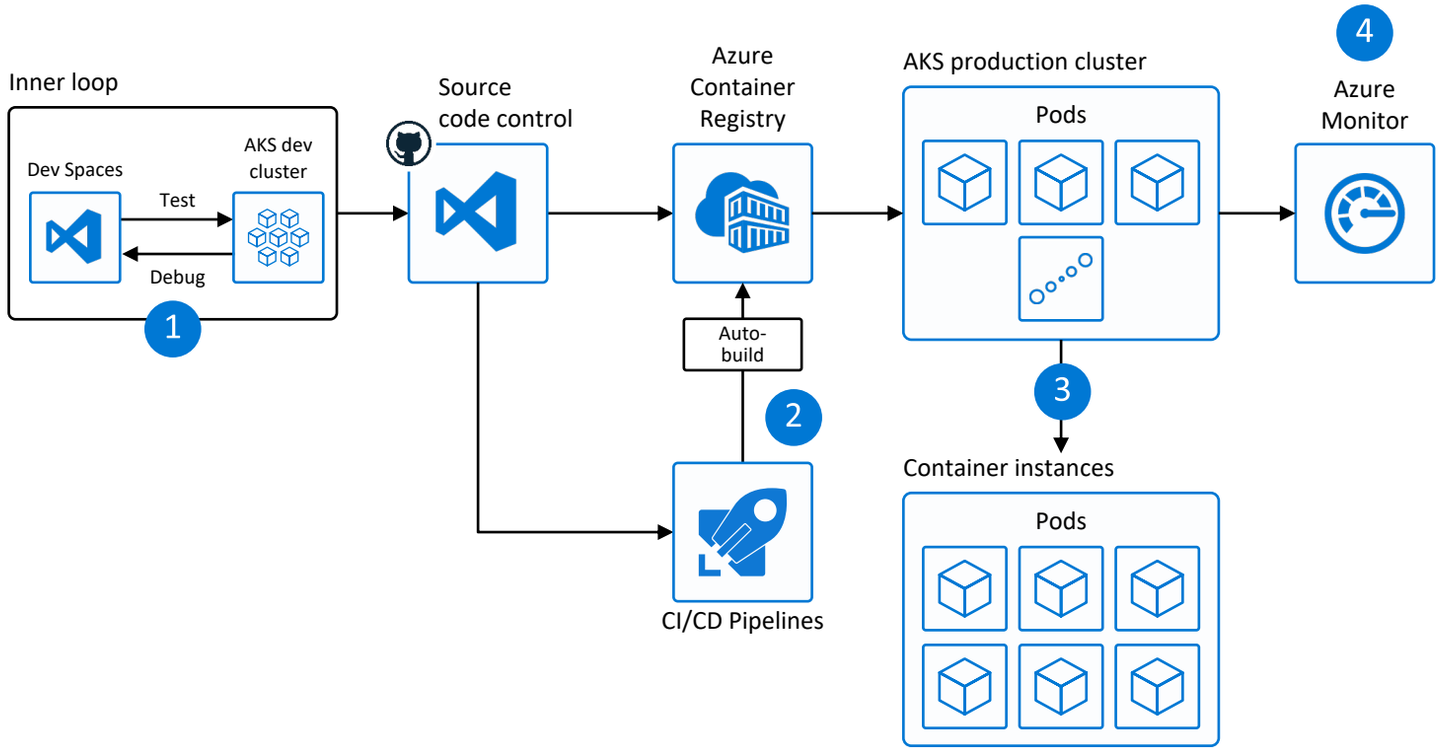
- Independent deployments
- Improved scale and resource utilization per service
- Smaller, focused teams



Microservices for faster app development

Capabilities

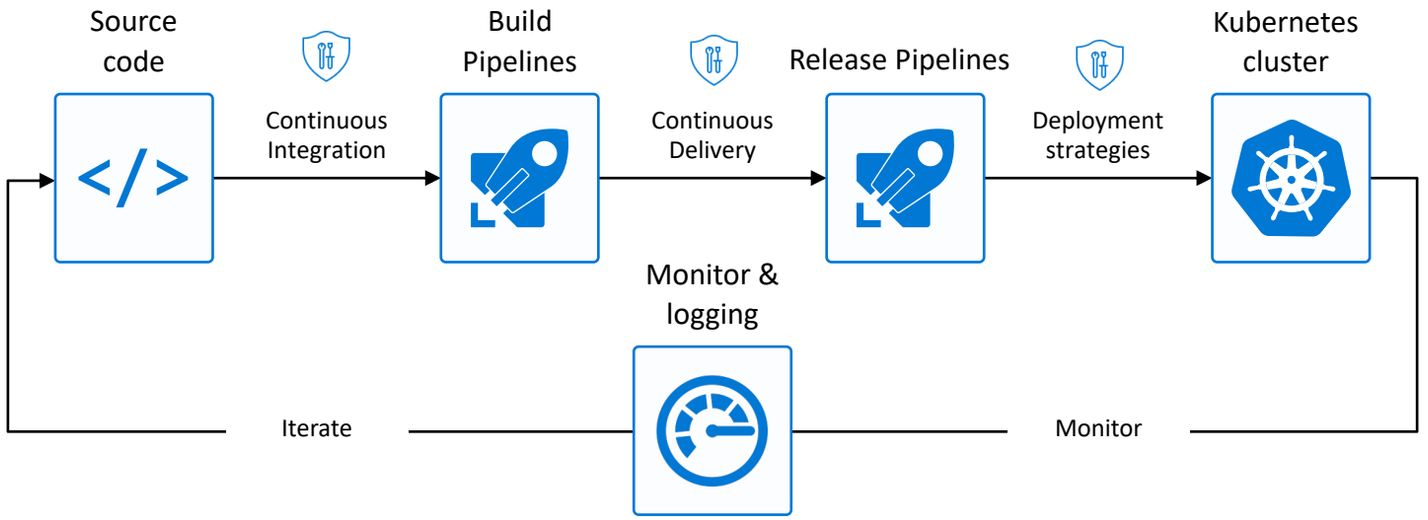
1. Use Azure Dev Spaces to iteratively develop, test, and debug microservices targeted for AKS clusters.
2. Azure Pipelines has native integration with Helm and helps simplifying continuous integration/continuous delivery (CI/CD)
3. Virtual node—a Virtual Kubelet implementation—allows fast scaling of services for unpredictable traffic.
4. Azure Monitor provides a single pane of glass for monitoring over app telemetry, cluster-to-container level health analytics.



https://github.com/Microsoft/SmartHotel360-AKS-DevSpaces-Demo

Secure DevOps

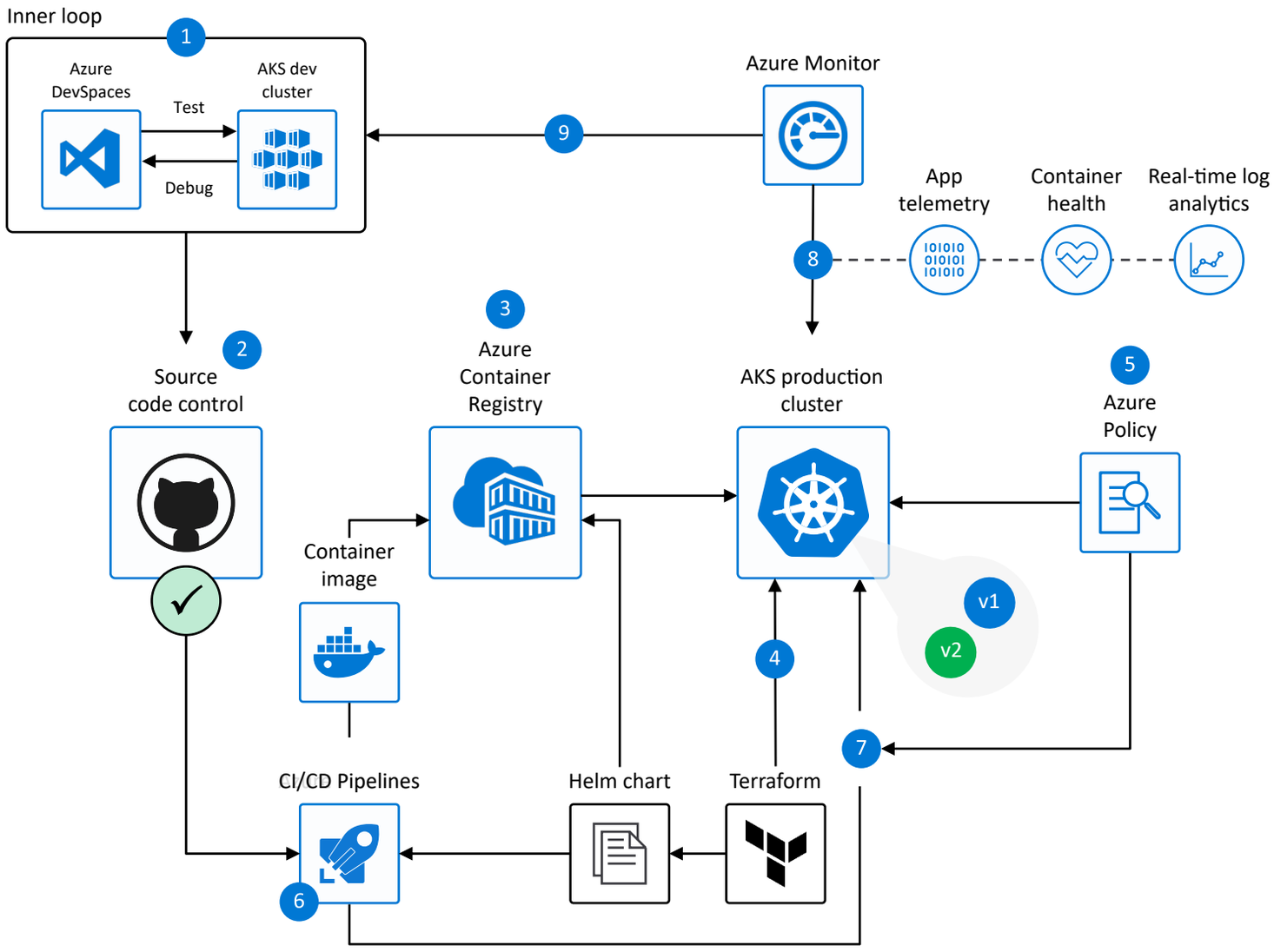
- Deliver code faster with Kubernetes and CI/CD
- Accelerate the feedback loop with constant monitoring
- Balance speed and security with continuous security and deep traceability



Secure DevOps

Capabilities

1. Developers rapidly iterate, test, and debug different parts of an application together in the same Kubernetes cluster
2. Code is merged into a GitHub repository, after which automated builds and tests are run by Azure Pipelines
3. Container image is pushed to Azure Container Registry
4. Kubernetes clusters are provisioned using tools like Terraform; Helm charts, installed by Terraform, define the desired state of app resources and configurations
5. Operators enforce policies to govern deployments to the AKS cluster
6. Release pipeline automatically executes pre-defined deployment strategy with each code change
7. Policy enforcement and auditing is added to CI/CD pipeline using Azure Policy
8. App telemetry, container health monitoring, and real-time log analytics are obtained using Azure Monitor
9. Insights used to address issues and fed into next sprint plans



Demo

github.com/kaluzaaa/czwartek-z-azure-aks

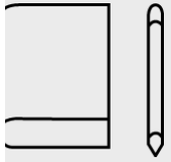
What next?

Learning path
aka.ms/LearnKubernetes

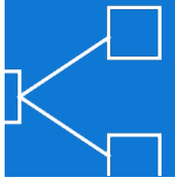
What is Kubernetes
aka.ms/k8sLearning

Hear from experts
aka.ms/AKS/videos

Case studies
aka.ms/aks/casestudy



See what's new
aka.ms/k8s/roadmap



Try for free
aka.ms/aks/trial