

Containerizing .NET Core Applications with Microsoft Azure and AKS

Manoj Ganapathi
Chief Architect, CodeOps

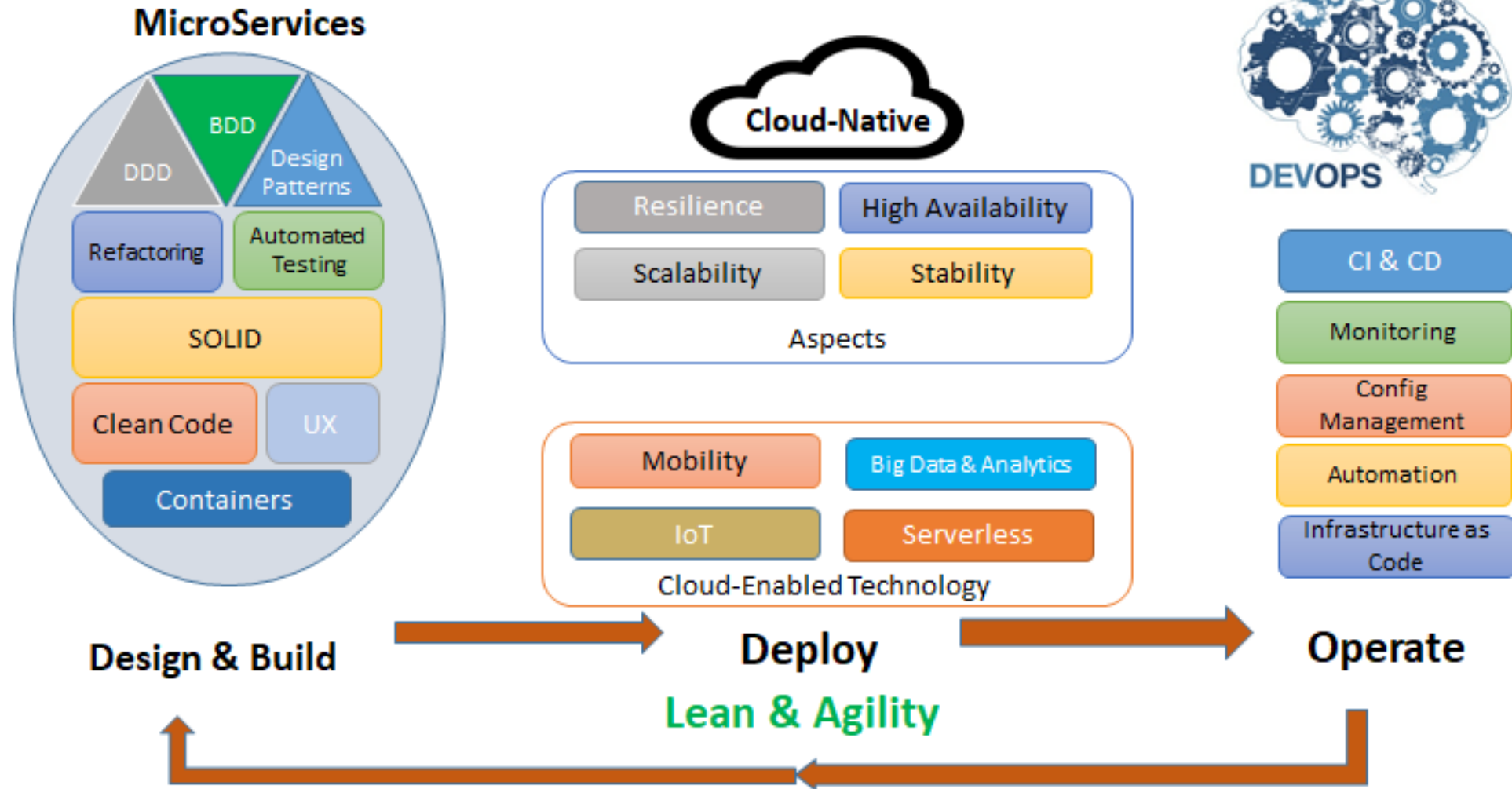


About Me



- Manoj is a seasoned IT professional with more than 20 years of experience. He has extensive experience in enterprise & solution architecture, design and implementation of large & complex enterprise systems. As an architect and technology consultant, he has consulted with several large, fortune 500 enterprises and worked with ISVs and startups. In his career, he has worked in multiple technology-oriented and leadership roles across all phases of software development life cycle. He is experienced in building and running technical communities and has been a speaker in several technology conferences.
- Over the last seven years, he has worked extensively on consulting, architecture and implementation of Cloud-based solutions, specializing on building highly scalable, resilient systems and DevOps practices.
- Currently, he is the Chief Architect at CodeOps Technologies (<http://codeops.tech/>) and a Digital Technology Consultant.
- LinkedIn profile: [@manojg](https://www.linkedin.com/in/manojg)
- @manojgr, manoj@codeops.tech

Perspective on Modern Software Delivery



Containers v/s VMs

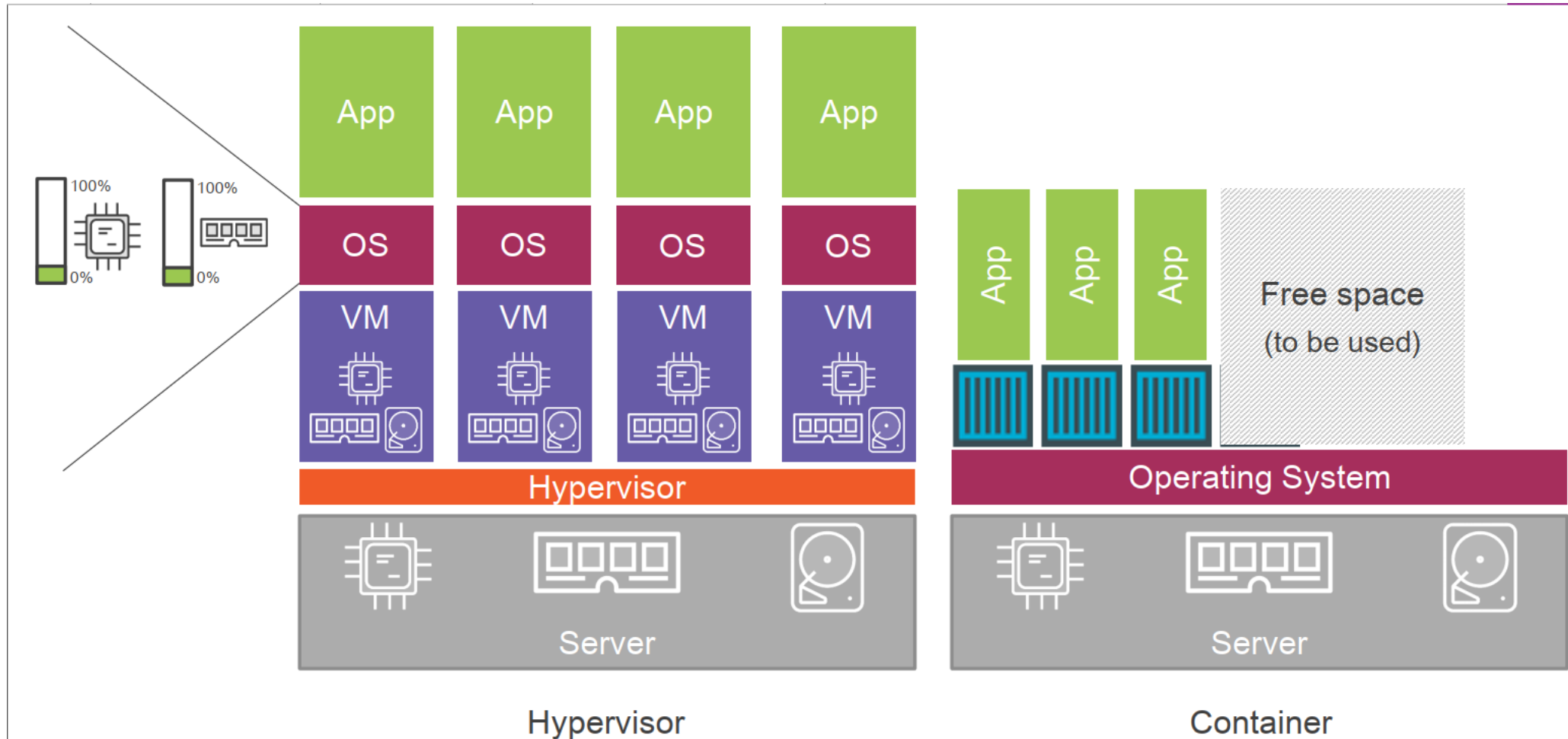
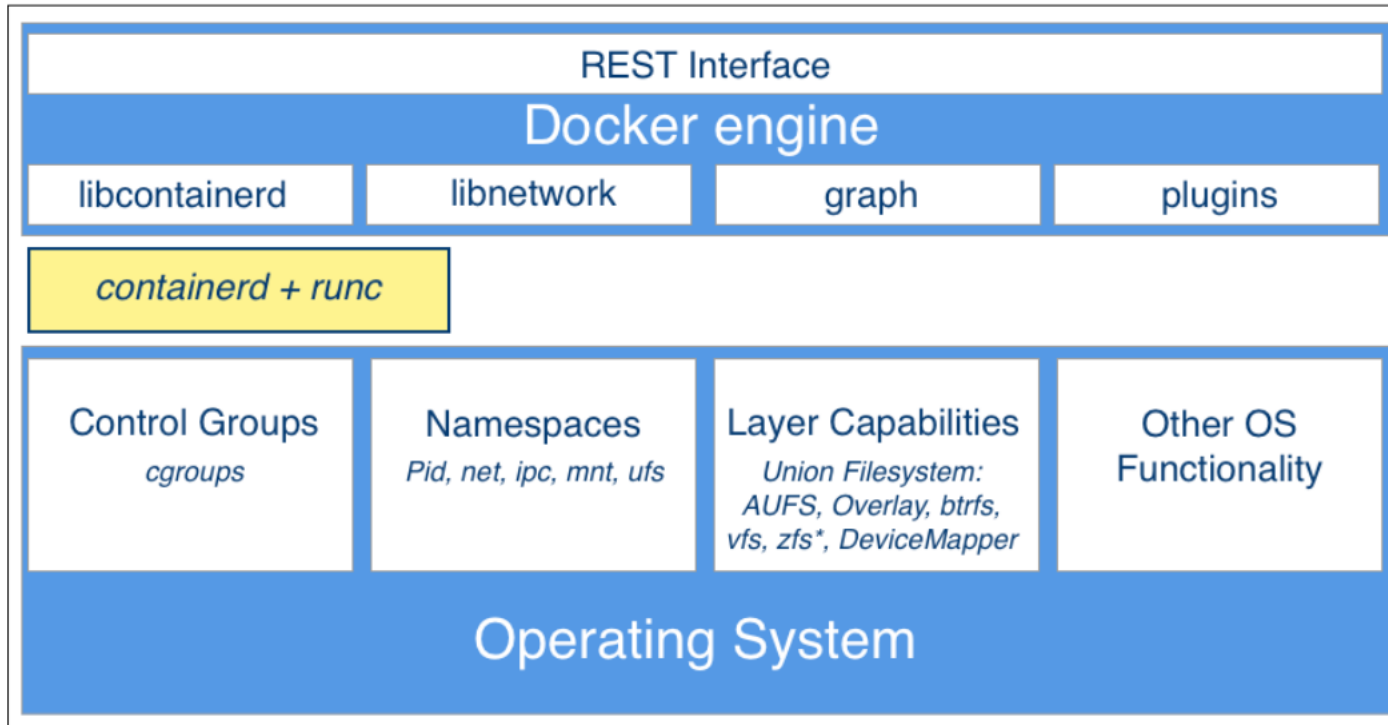


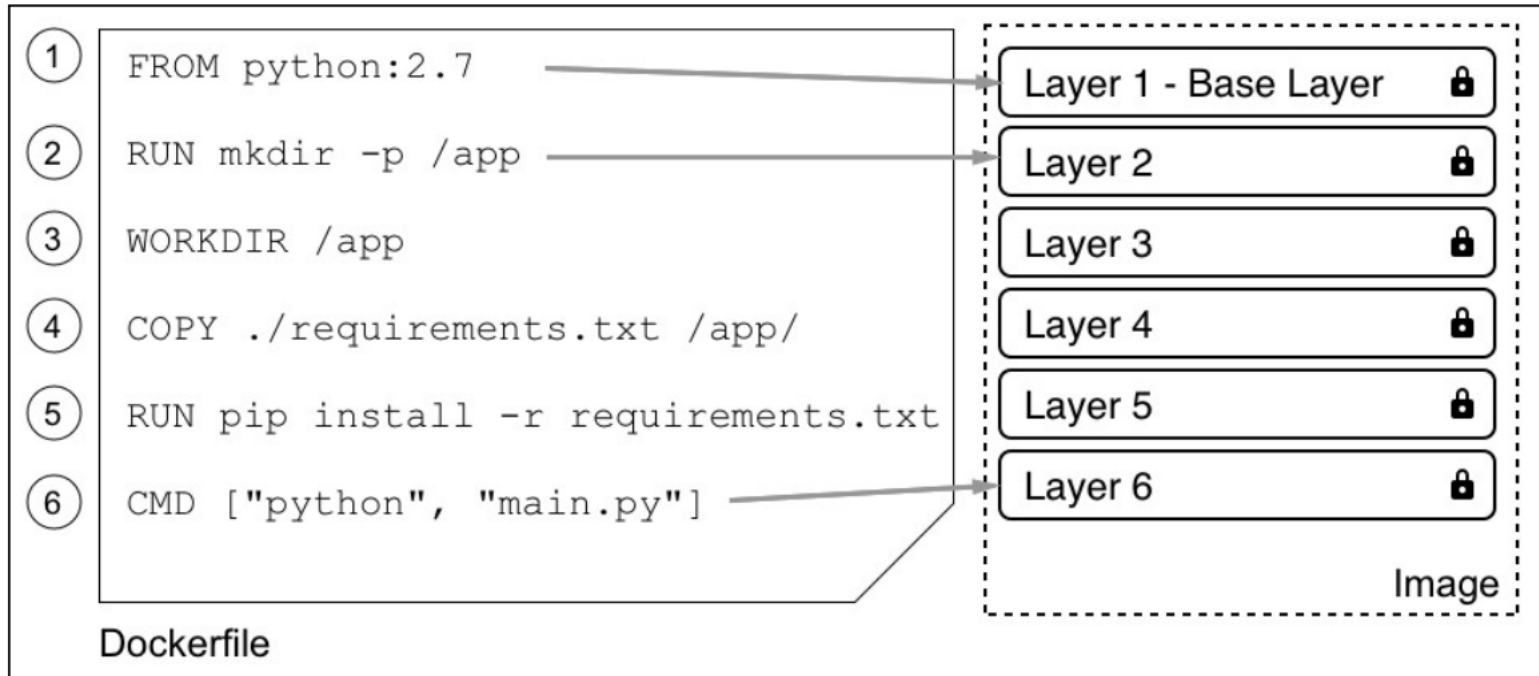
Image Ref: Nigel Poulton

Docker Architecture



- Containers are encapsulated, secure processes running on the host platform
- Benefits
 - Security
 - Isolation
 - Standardized Infra

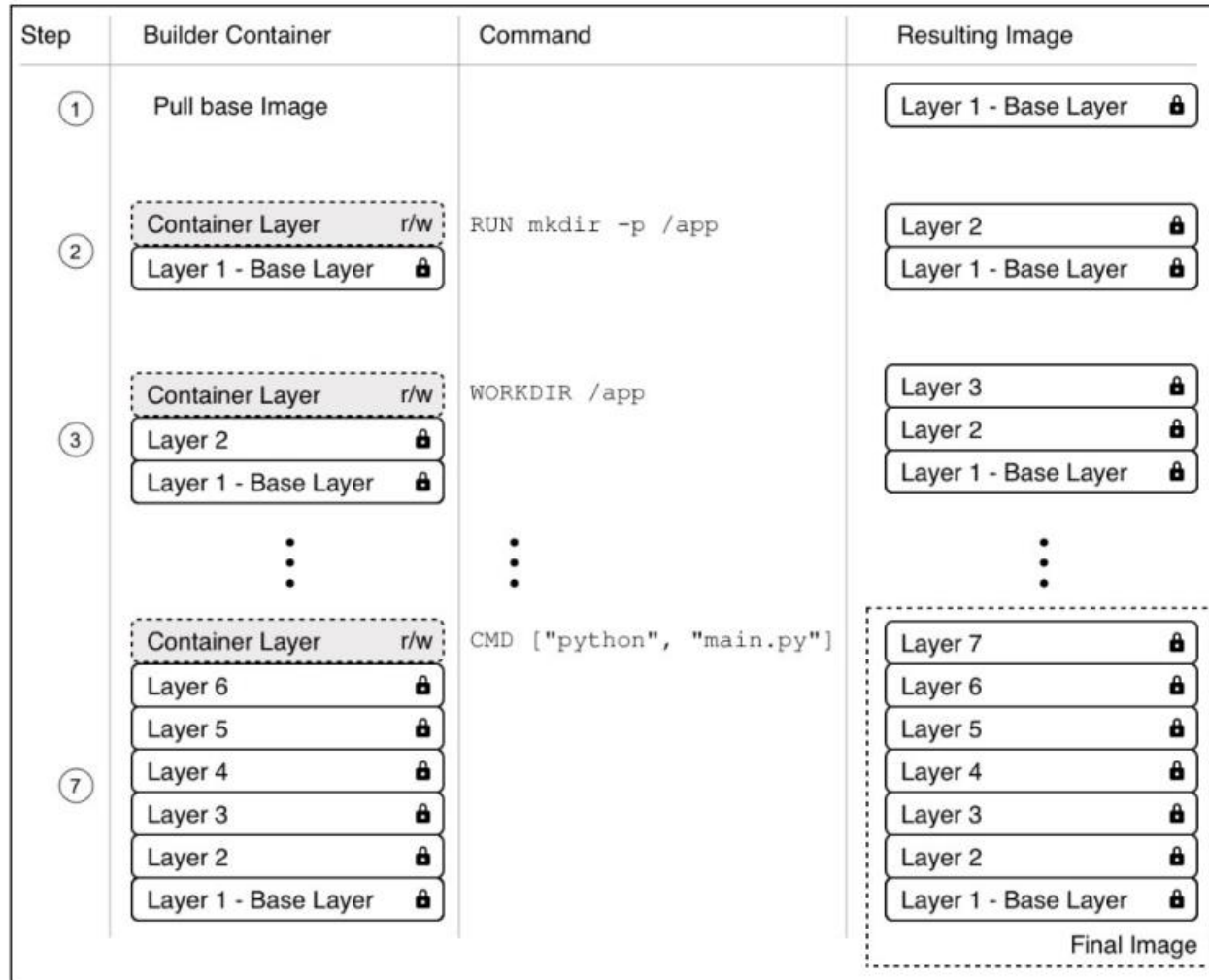
Container Images & Dockerfile



The relation of Dockerfile and layers in an image

- Templates for Containers
- Starts with a base layer, usually the OS
- All layers immutable, only top layer is writable

Best Practices



The image build process visualized

- Keep Containers ephemeral
- Order commands to leverage caching
- Avoid installing multiple packages
- Use .dockerignore
- Use multi-stage builds

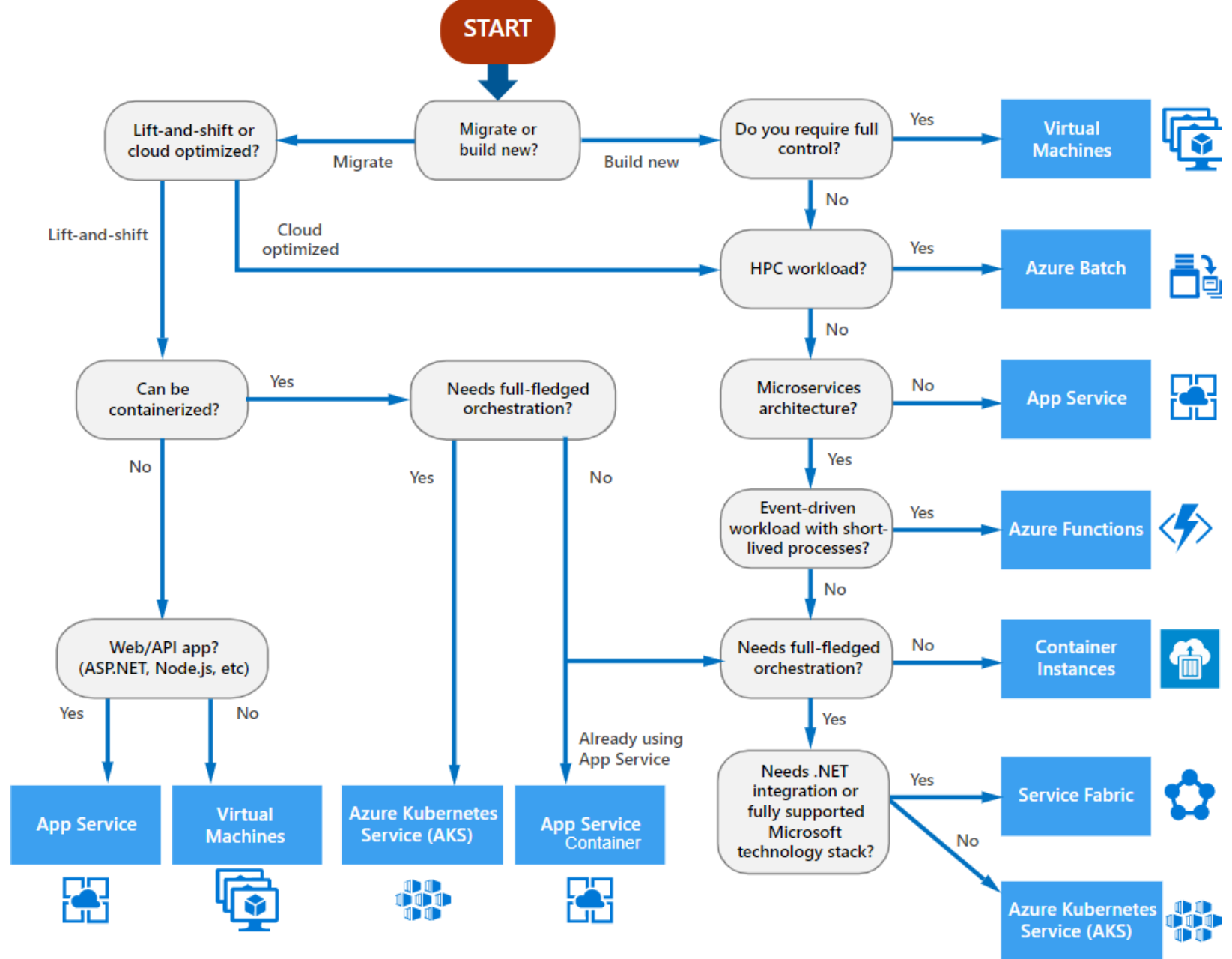
Advantages of Containers

- Very lightweight
- Scales quickly
- Accelerates developer onboarding
- Easy to package apps, dependencies and version
- Eliminate environment inconsistencies
- Easier distribution
- Support CI/CD DevOps more seamlessly
- Better oriented towards Cloud-Native and distributed architecture patterns

Container Registry

- Single place for your team to manage Docker images
- Public and Private registries
 - Azure Container Registry - Private
 - Docker Hub - Public
- Provide features like:
 - Vulnerability analysis
 - Geo-replication
 - Role-based access control.
- Docker Registry API supported by all registry providers (like Azure)
- Important element of CI/CD pipelines

Choosing the right compute option



Azure Container Instances

- Run containers without managing servers or having to learn new tools
- With the Virtual Kubelet, use ACI to elastically burst from your Azure Kubernetes Service (AKS) cluster when traffic comes in spikes.
- Gain the security of virtual machines for your container workloads, while preserving the efficiency of lightweight containers
- Combine ACI with the ACI Logic Apps connector, Azure queues and Azure Functions to build robust infrastructure which can elastically scale out containers on demand
- Use Azure Container Instances for data processing where source data is ingested, processed and placed in a durable store such as Azure Blob storage

Need for Container Orchestrators



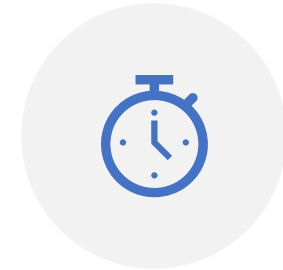
VELOCITY



SCALING (SOFTWARE
& TEAMS)



ABSTRACTING
INFRASTRUCTURE



EFFICIENCY

Achieving Velocity



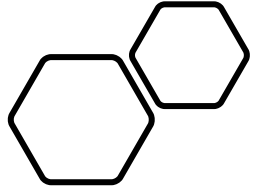
IMMUTABILITY



DECLARATIVE
CONFIGURATION



SELF-HEALING



Other benefits



DECOUPLED
ARCHITECTURE



HIGH UTILIZATION



EFFICIENCY

Why Azure Kubernetes Service (AKS)?

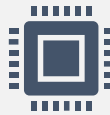


Highly available, secure and fully managed Kubernetes service

Fully managed control plane – you focus on your applications/workloads



Elastic provisioning of capacity without the need to manage the infrastructure



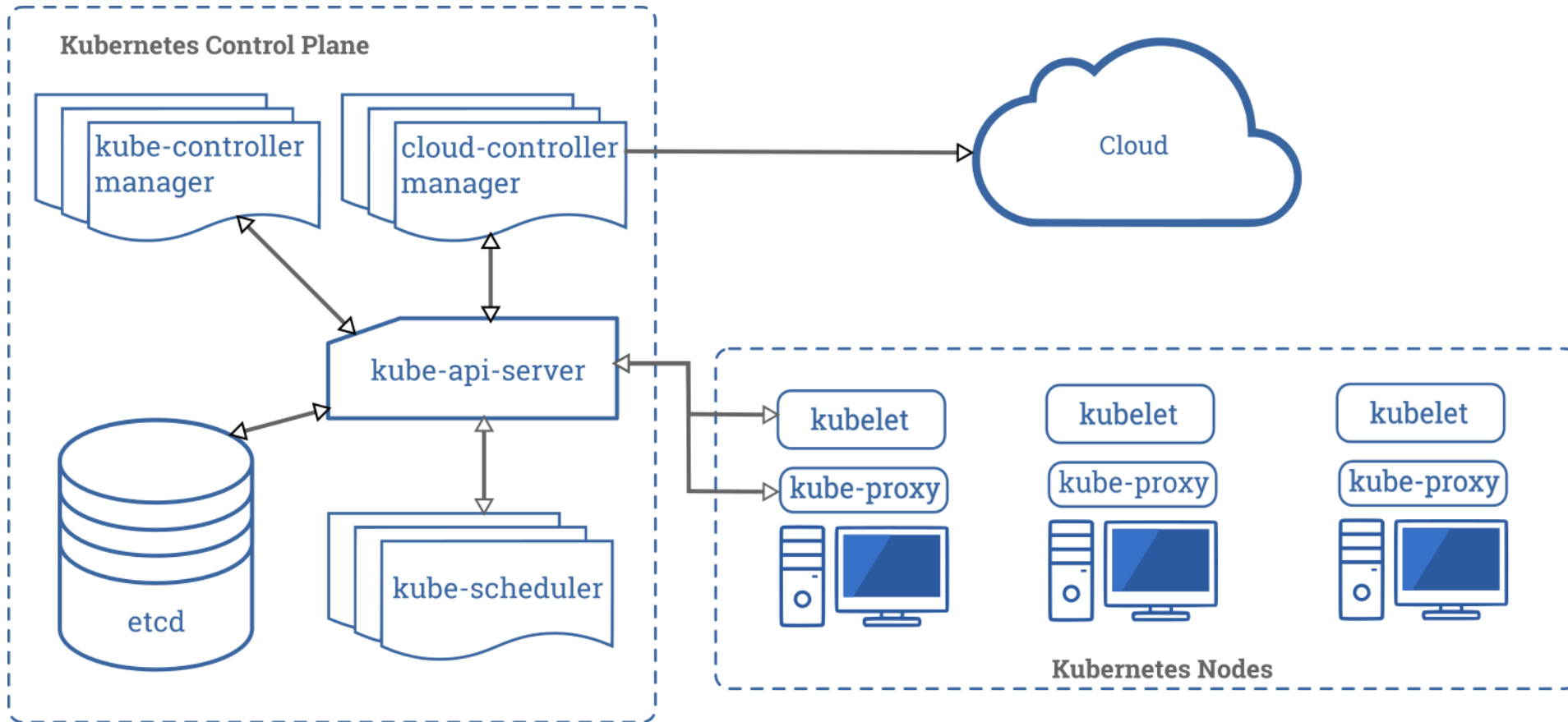
Faster end-to-end development experience with seamless integration with Visual Studio/VS Code, Dev Spaces, Azure DevOps and Azure Monitor



Advanced identity and access management using Azure Active Directory and dynamic rules enforcement across multiple clusters with Azure Policy

<https://azure.microsoft.com/en-in/services/kubernetes-service/>

K8S Architecture



<https://kubernetes.io/docs/concepts/overview/components/>

K8S objects

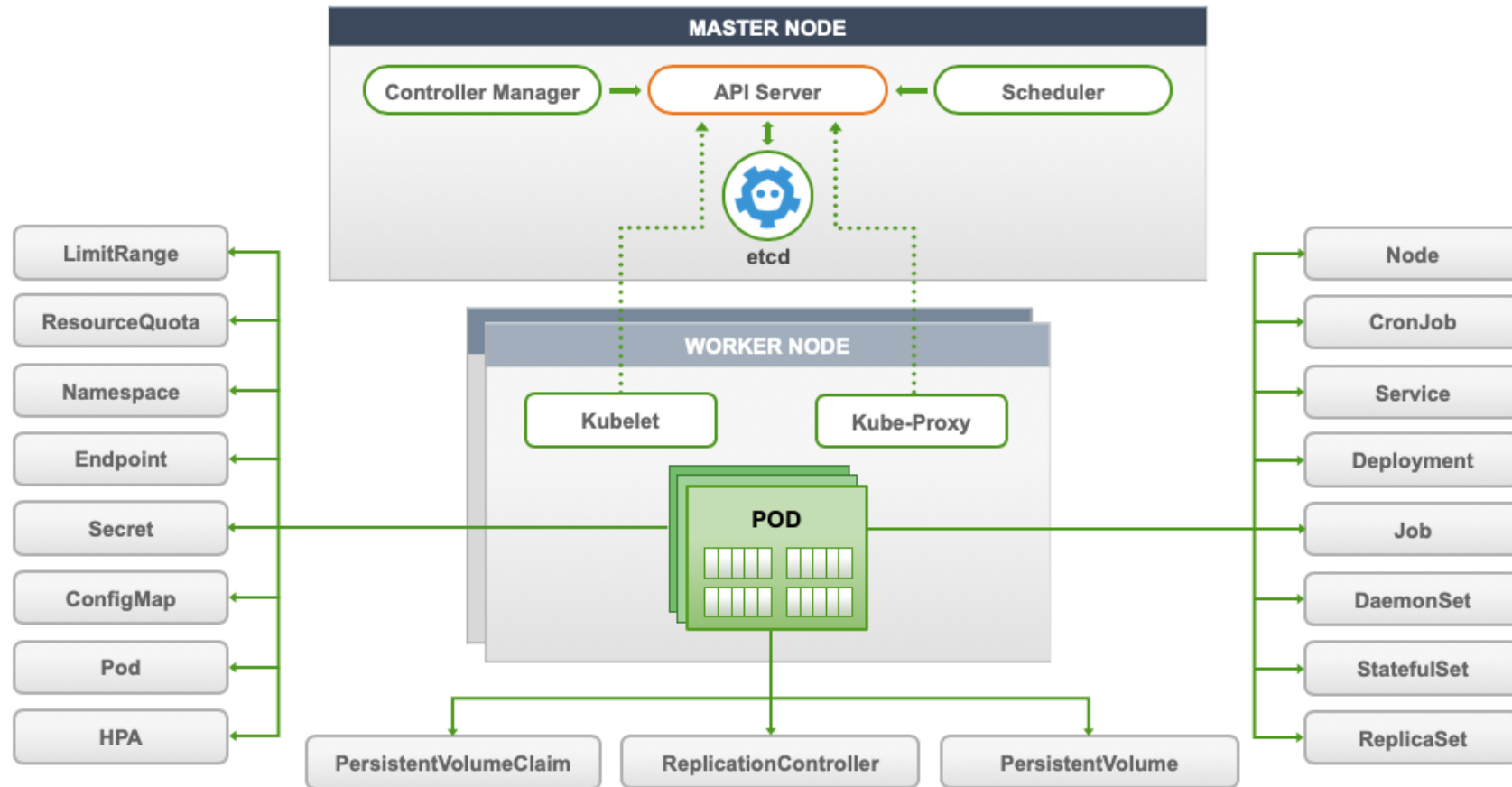


Image Ref: https://www.splunk.com/en_us/blog/it/monitoring-kubernetes.html

Pod Lifecycle

Phases

Pending

Running

Succeeded

Failed

Unknown

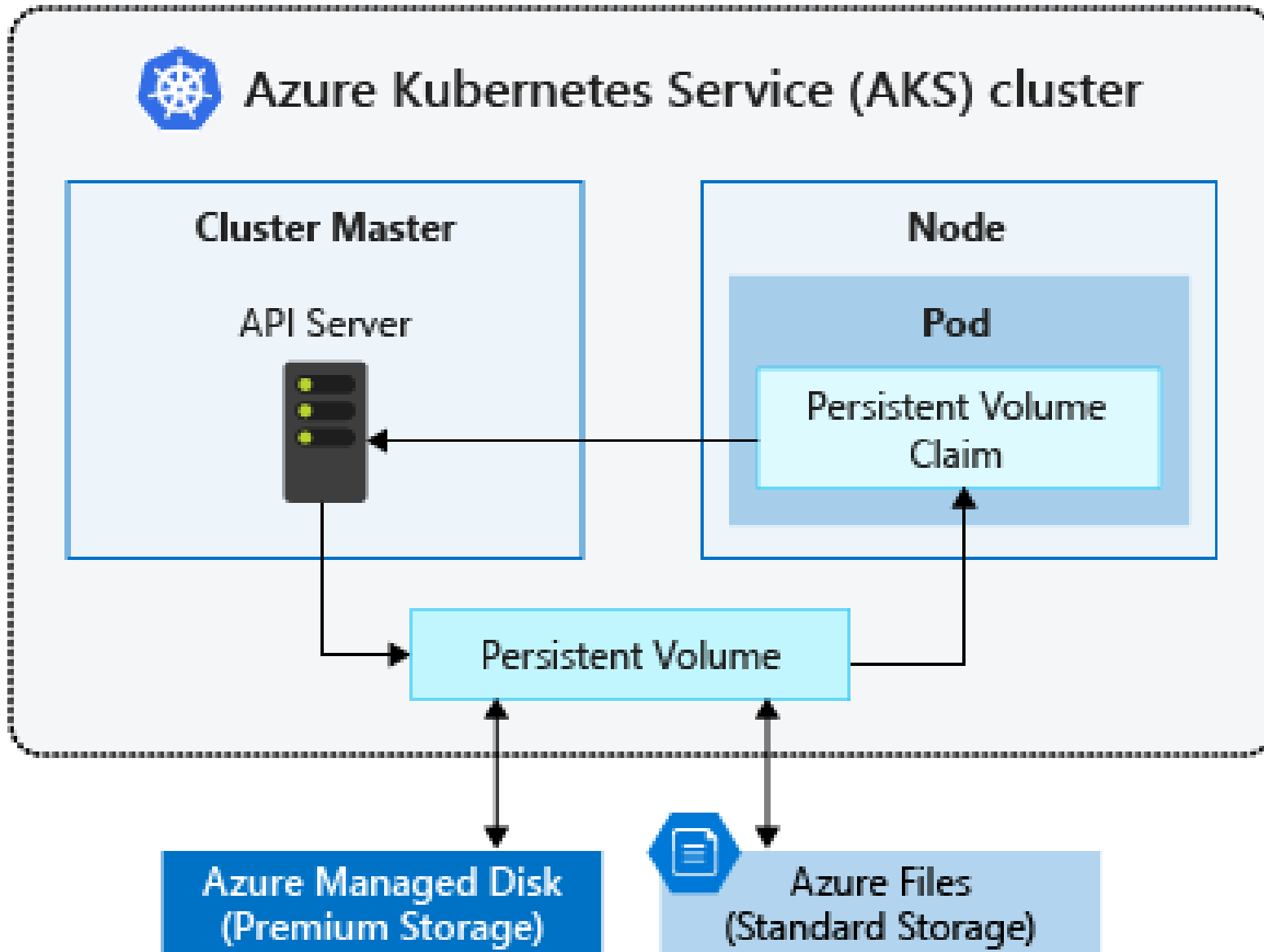
Status

PodScheduled

ContainersReady

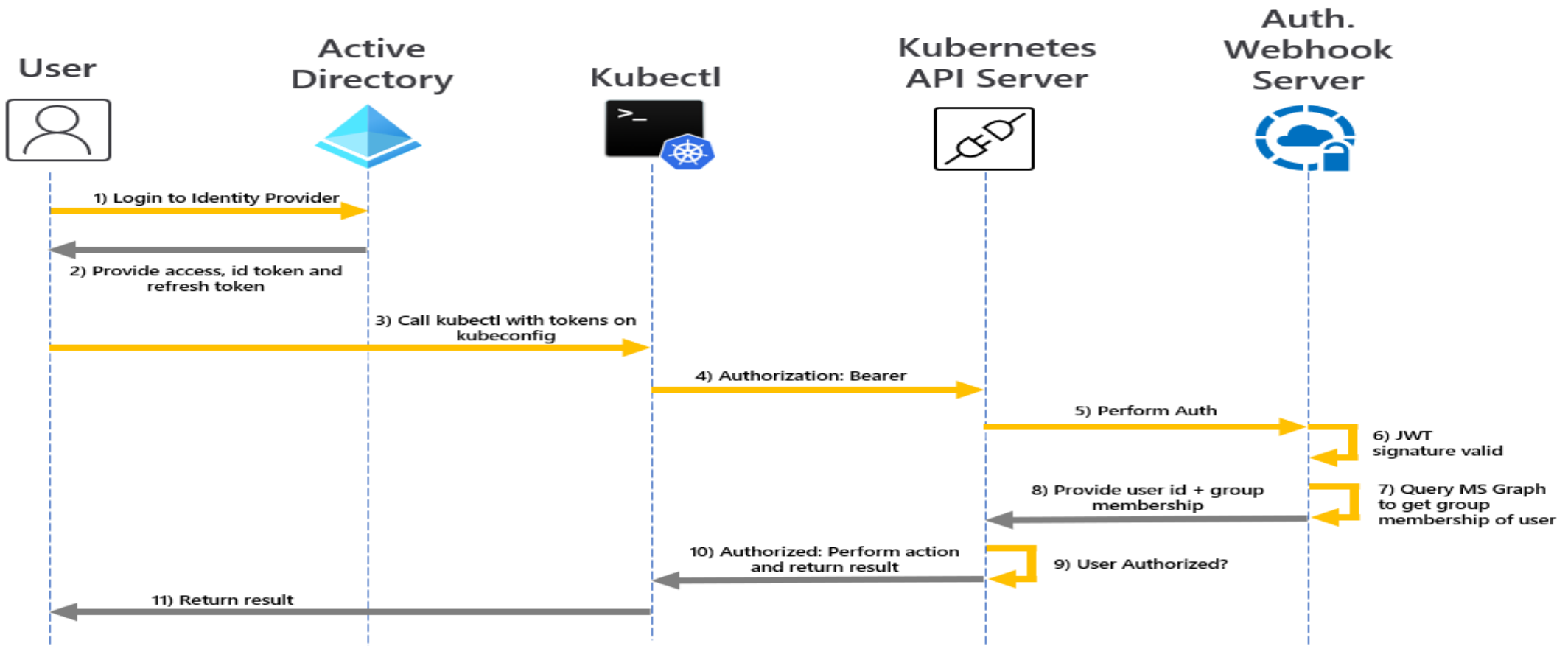
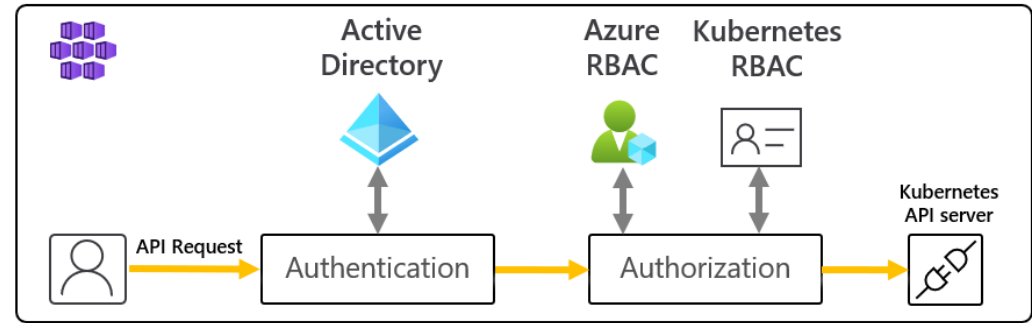
Initiatized

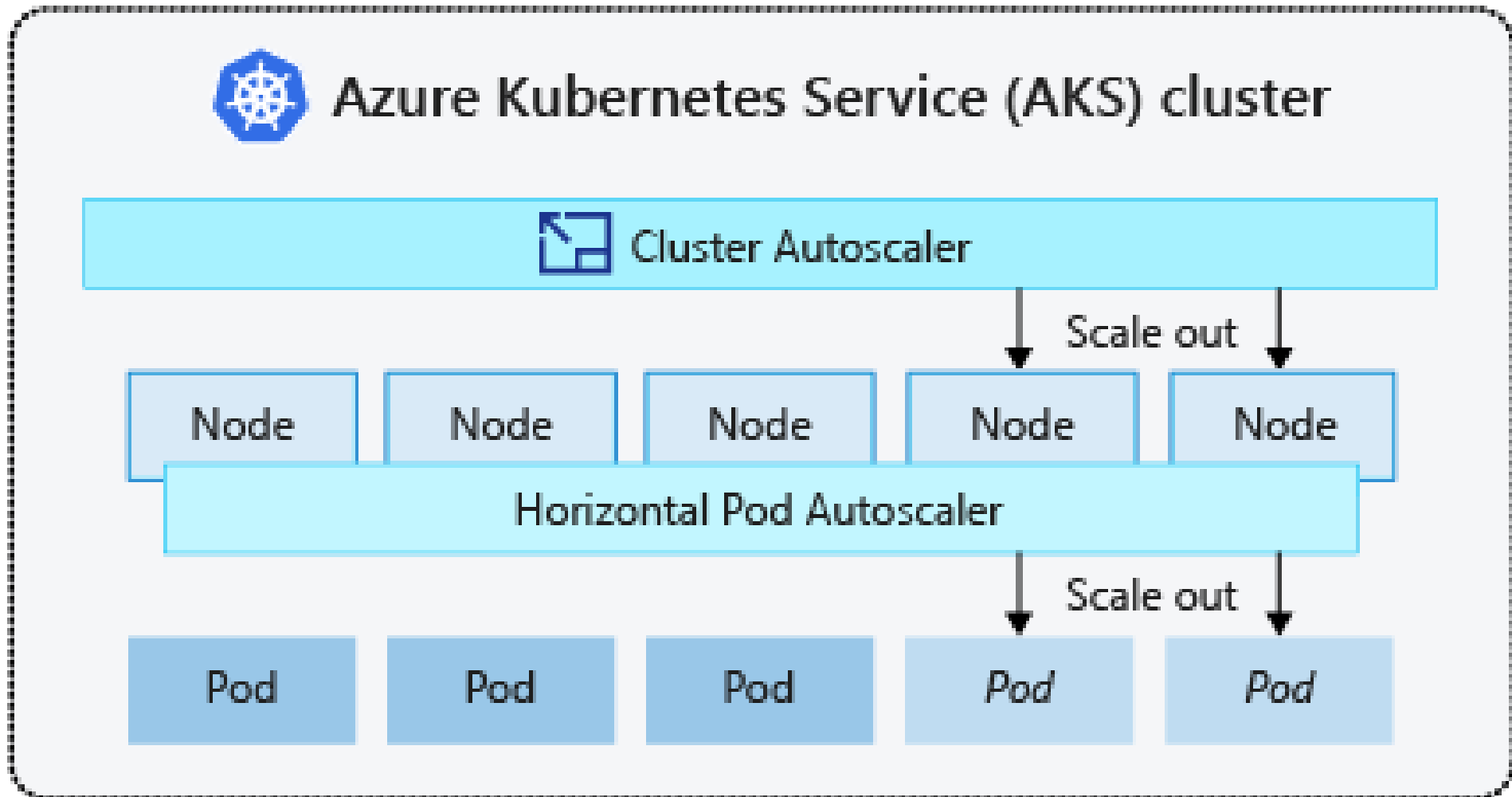
Ready



Storage in
K8S/AKS

Authentication & Authorization

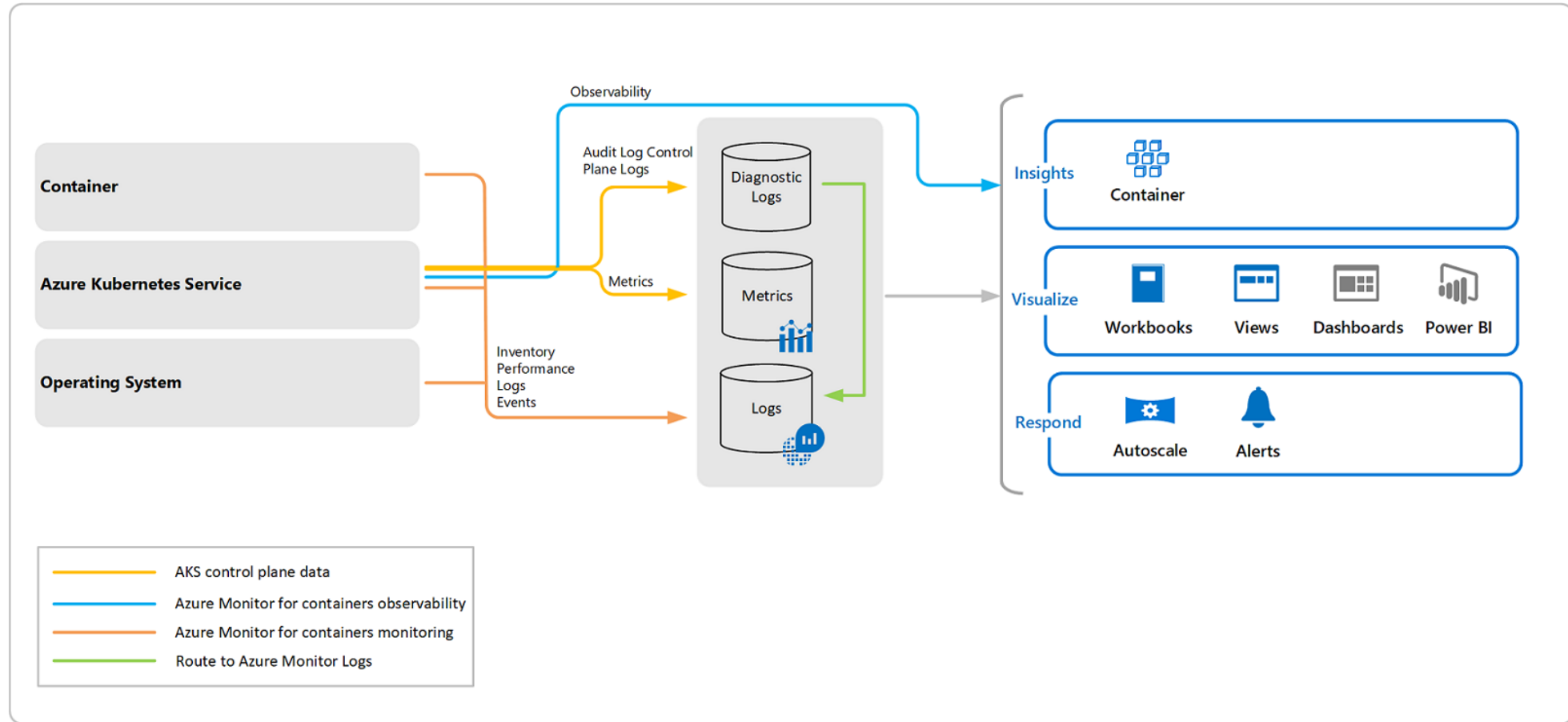




Scaling options for applications in Azure Kubernetes Service (AKS)

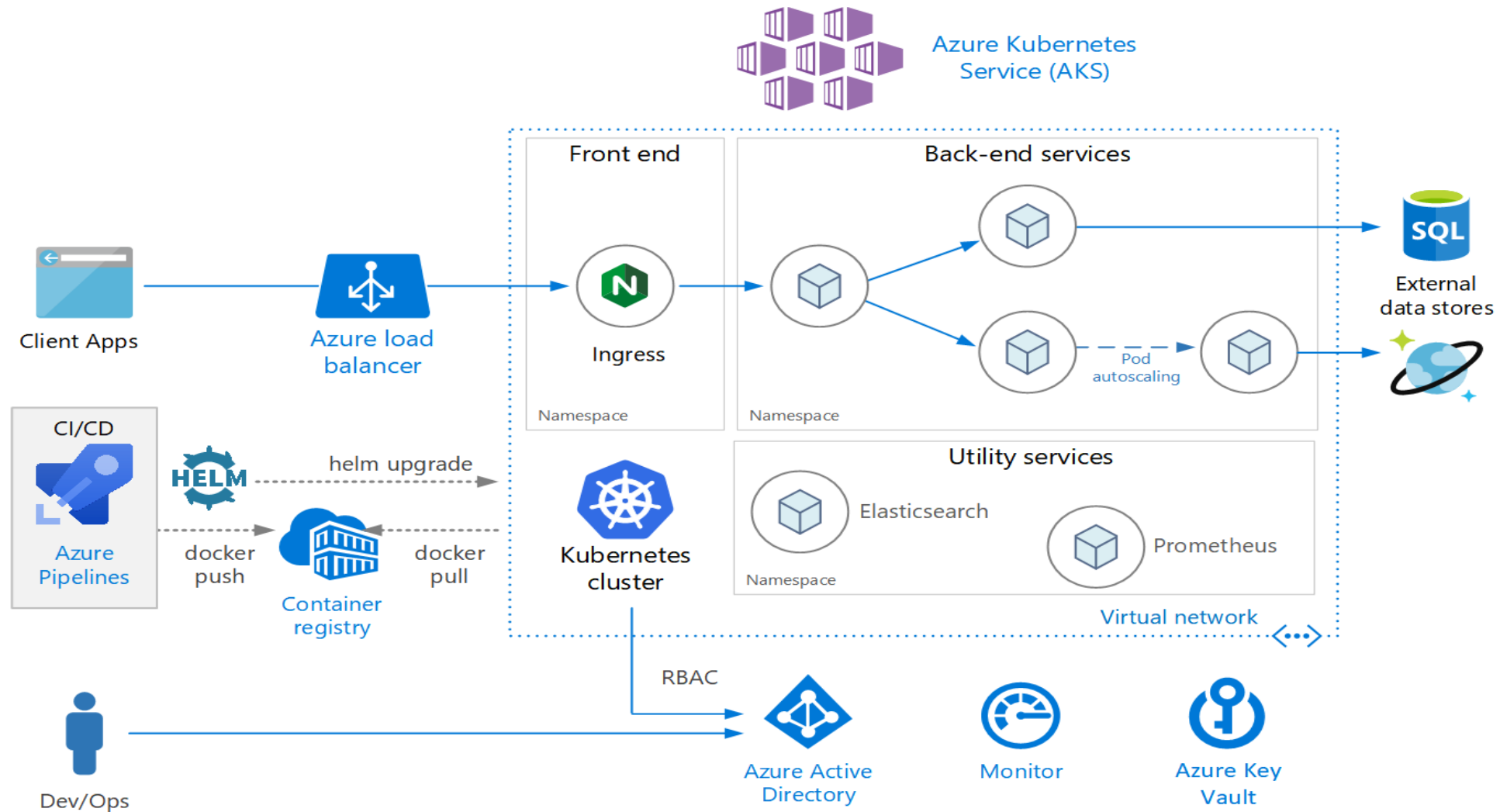
<https://docs.microsoft.com/en-us/azure/aks/concepts-scale>

Monitoring in AKS (Azure Monitor)



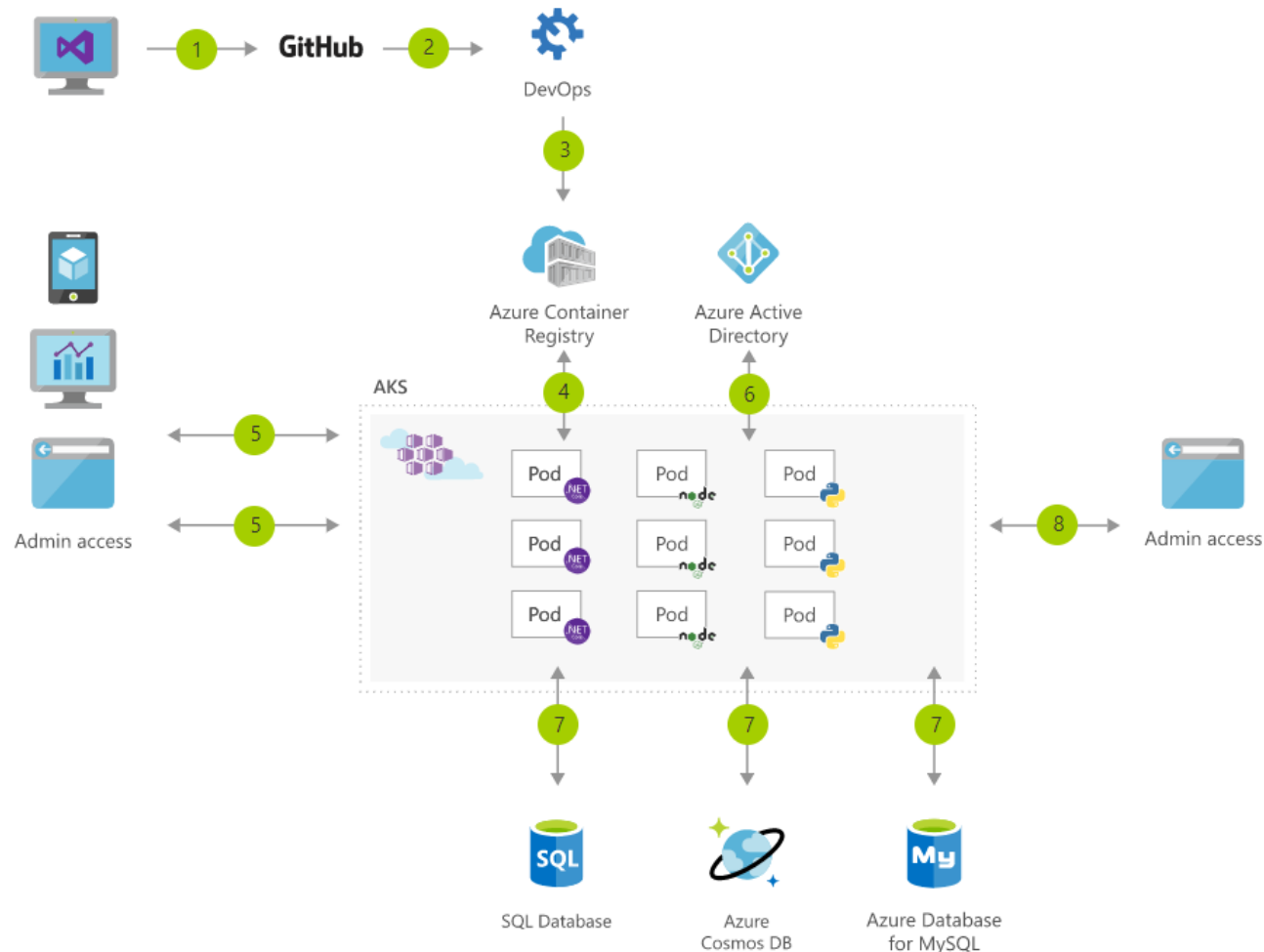
<https://docs.microsoft.com/en-us/azure/azure-monitor/insights/container-insights-overview>

Microservices architecture on Azure Kubernetes Service (AKS)



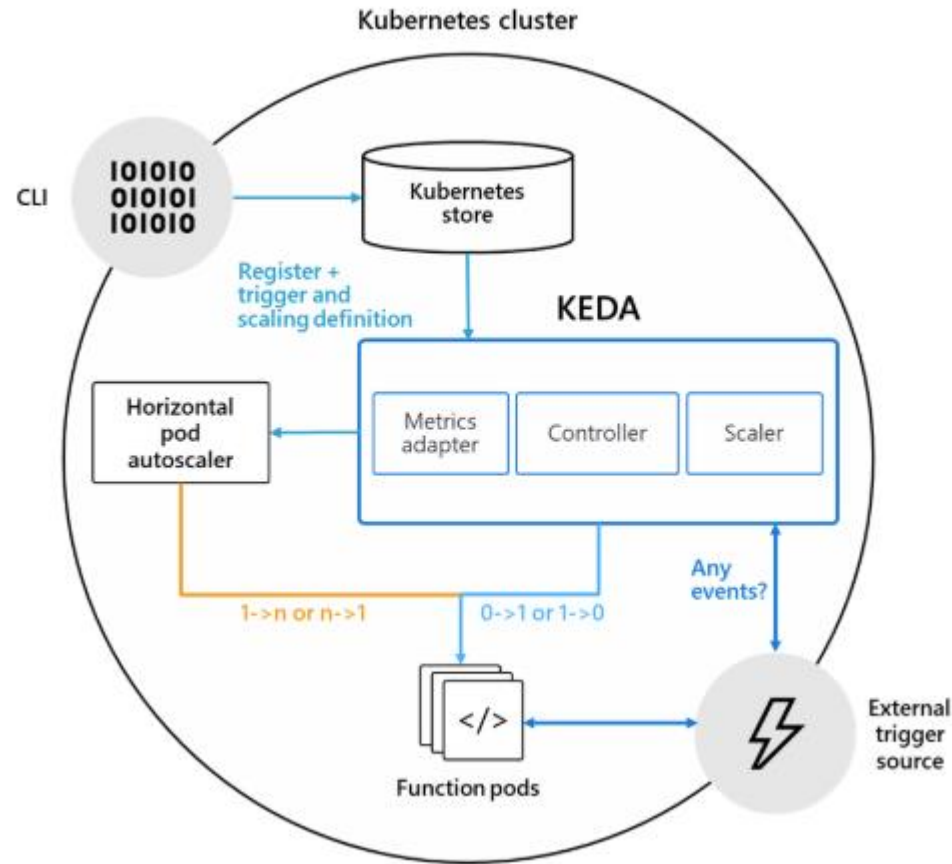
<https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/containers/aks-microservices/aks-microservices>

Deploying with AKS (DevOps)



- 1 Use an IDE, such as Visual Studio, to commit changes to GitHub.
- 2 GitHub triggers a new build on Azure DevOps
- 3 Azure DevOps packages microservices as containers and pushes them to the Azure Container Registry
- 4 Containers are deployed to AKS cluster
- 5 Azure Active Directory is used to secure access to the resources
- 6 Users access services via apps and websites
- 7 Administrators access the apps via a separate admin portal
- 8 Microservices use databases to store and retrieve information

KEDA (Kubernetes Event Driven Autoscaling)



<https://cloudblogs.microsoft.com/opensource/2019/05/06/announcing-keda-kubernetes-event-driven-autoscaling-containers/>

Resources

- [K8s Learning Path](#)
- [Docker on Azure](#) -
- [Containerize your apps with Docker & Kubernetes \(Book\)](#)
- [Learning path for Containers & Kubernetes on MS Learn](#)
- [Git Hub Repo \(Labs\)](#)
- [K8S Cheat Sheet](#)
- [AKS Best Practices](#)
- [Kubernetes Learning Path](#)
- [Kubernetes Up and Running Book](#)