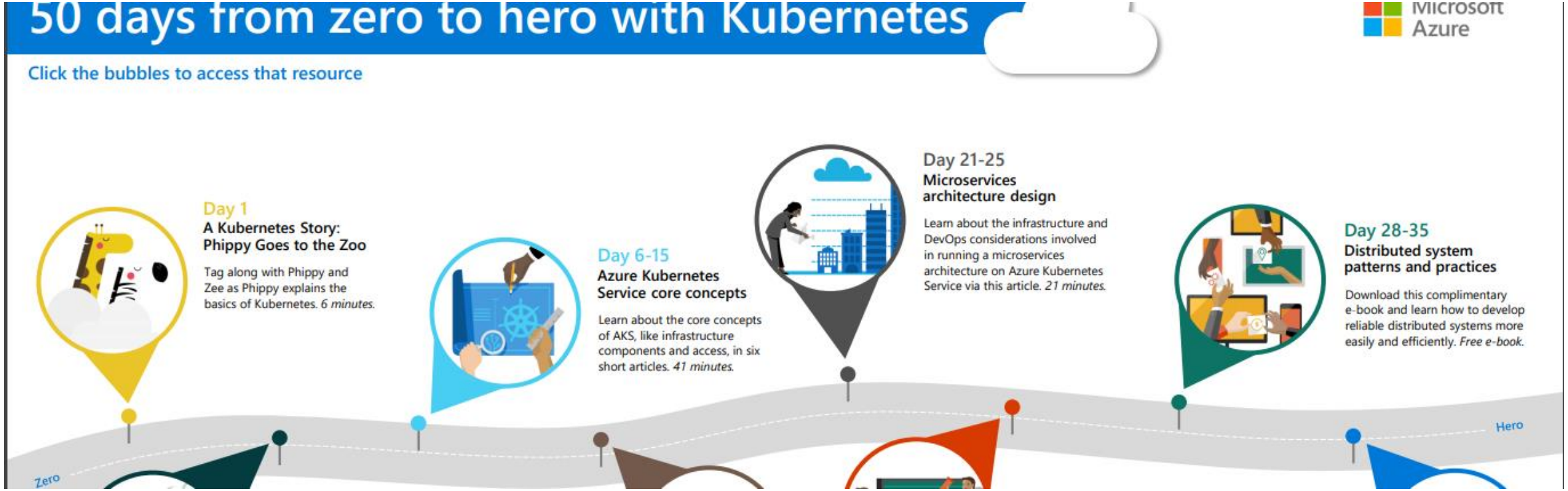
Cloud Native Application Development with Azure Kubernetes Service

Wely Lau
Sr Cloud Solution Architect
One Commercial Partner, Microsoft APAC

Disclaimer

- This is not Containers, Docker, Kubernetes 101 Session
- But if you are new to Containers and Kubernetes,
  - I'll be doing a quick crash course
  - check out:



https://aka.ms/LearnKubernetes

Agenda

- Demystifying Cloud Native
- Crash course: Containers, Docker, and Kubernetes
- Kubernetes on Azure: top scenarios and reference architecture
- Best practices
- Scaling apps on AKS (Azure Kubernetes Services)
- What’s new in AKS?
- What to expect in the next session
- Reference and learn more

Demystify Cloud Native

[1]IDC FutureScape, 2018

What is cloud native?

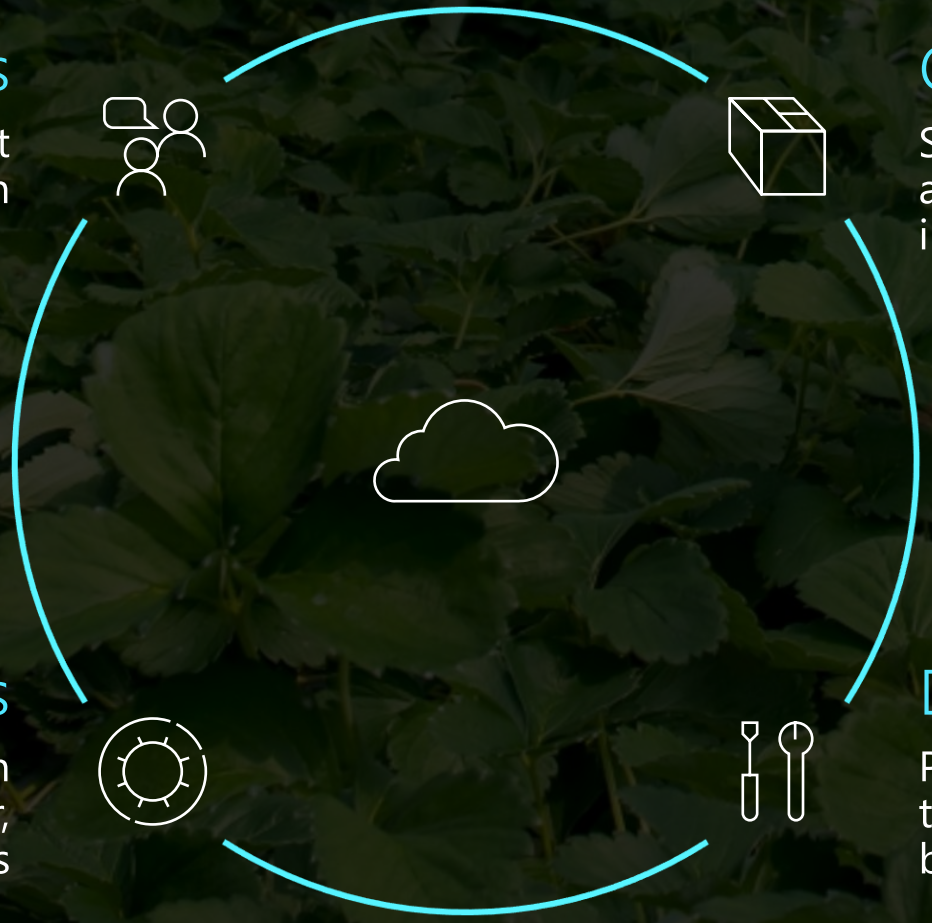
Package application code & dependencies in containers, deploy as microservices and manage them using DevOps processes & tools

APIs

Expose services as light-weight APIs for easier integration

Containers

Standard deployment format to abstract code from underlying infrastructure differences



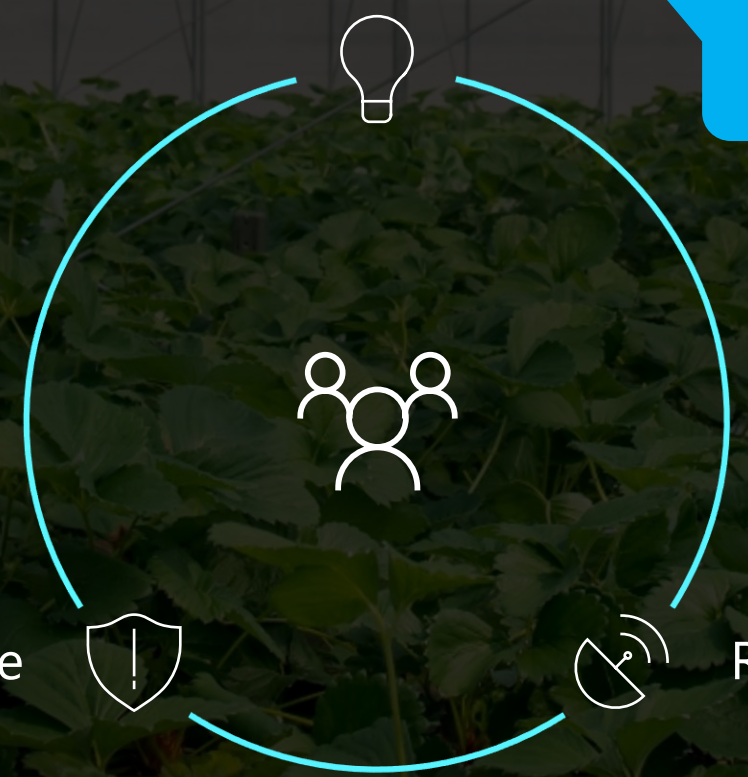
Microservices

Architectural approach to developing an application as a collection of modular, loosely coupled services

DevOps

People, processes and technology that promote collaborative building and delivery

Why cloud native?



Cloud Native IS NOT for EVERYONE!

Customers adopting cloud native need to be aware about

Platform and component typically change fast

Rapid development and deployment

Higher frequency of service decommissioning or breaking change

Why Azure for cloud native apps?

Speed of innovation

- Strongest developer experience[1]
- Most complete tool chain from Git to production
- Industry-leading MLOps and most comprehensive AI portfolio

Reliability

- Fully managed database services with >99.99-percent high availability
- Single-digit millisecond latencies on reads and writes
- Available in more regions than any other cloud providers

Security at scale

- $1BN investment every year in security
- >90 compliance certifications
- Out of the box integration with Azure Policy, Active Directory and Security Center

[1]Forrester New Wave for Public Cloud Enterprise Container Platforms

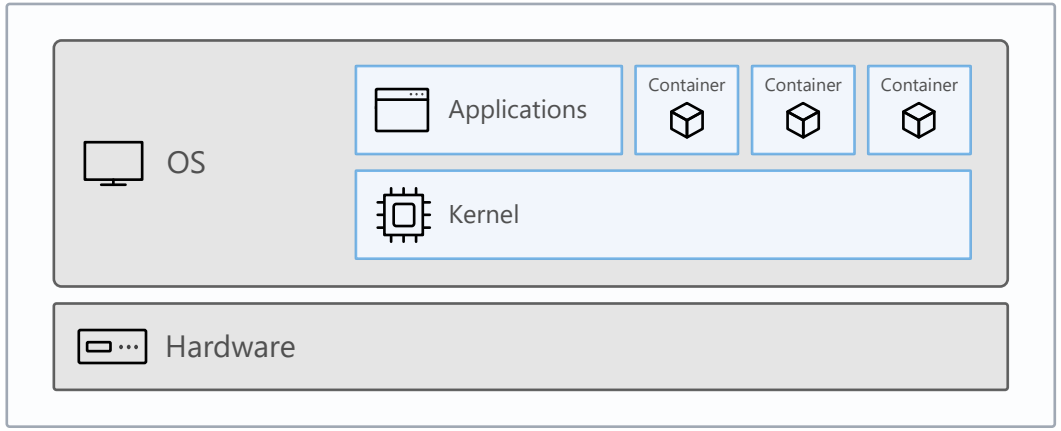
Crash Course:

Containers, Docker, and Kubernetes 101

What is a container? (container vs VM)

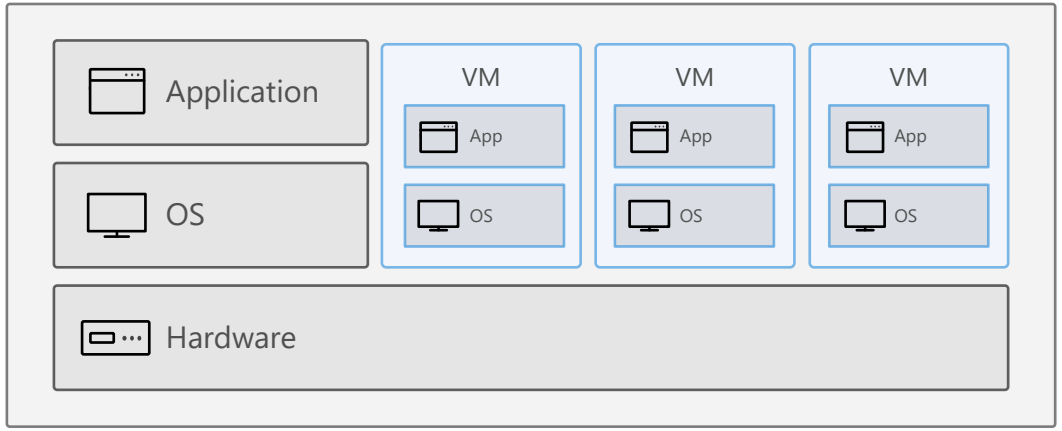
- Container is an application packaging and deployment mechanism

Containers = operating system virtualization



- Shared Host OS
- Near instant start-up
- Dependent app services and libraries are tied to container (layers)

Traditional virtual machines = hardware virtualization



- Each VM has independent, full OS
- Full isolation
- Separate app frameworks

Benefits of container

EFFICIENCY – HIGHER DENSITY THAN VM

LIGHTWEIGHT – SHIP SOFTWARE FASTER

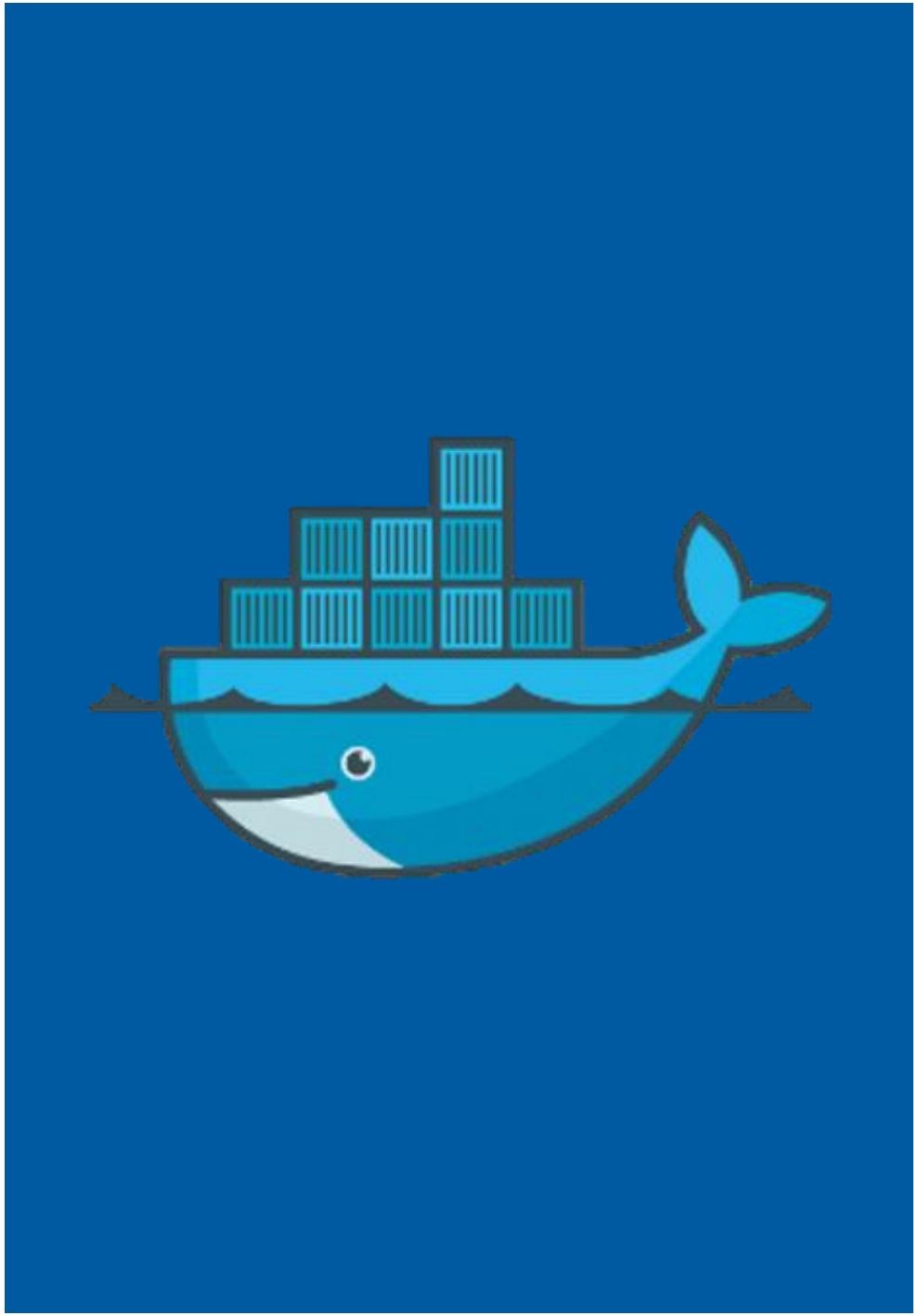
LIFT AND SHIFT CONTAINER IMAGES – AVOIDANCE OF LOCK-IN

CONSISTENCY ACROSS DEVELOPMENT, TEST, & PRODUCTION

EASE OF SCALING THROUGH CLOUD ELASTICITY

Docker

- Docker is the “de-facto” container format and sets of tools / APIs, opened-source, and managed by Docker Inc.
- Docker tools:
  - Docker CLI, Docker Engine, Docker Swarm, Docker Compose, etc.
- Can run on-premise or cloud
- Started in Linux, but later in Windows (Win 10 or Win Server 2016)
- Comes in 2 edition:
  - Community Edition
  - Enterprise Edition
- A docker image instantiate running container(s)



Question:

Is Docker alone enough for us to build and run containerized apps?

The elements of **orchestration**

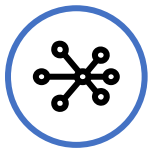
Scheduling

Affinity/anti-affinity

Health monitoring

Failover

Scaling



Networking

Service discovery

Coordinated app upgrades

Kubernetes: the industry leading orchestrator

Portable

Public, private, hybrid, multi-cloud

Extensible

Modular, pluggable, hookable, composable

Self-healing

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

https://github.com/kubernetes/kubernetes

About Kubernetes

8 characters

- Also referred to as **k8s**
- Kubernetes comes from the Greek word **κυβερνήτης**:, which means helmsman or ship pilot.
- Kubernetes is highly inspired by the **Google Borg system**.
  - v1.0 released on July 21, 2015 by Joe Beda, Brendan Burns, & Craig McLuckie
  - It is an open source project written in the Go language
  - Google donated it to the **Cloud Native Computing Foundation (CNCF)** and licensed under the Apache License Version 2.0.
  - Actively contributed by Google, Red Hat, Microsoft, IBM, etc.
- **kubectl** is the primary way to interact with Kubernetes but GUI (dashboard) and APIs exists

Kubernetes on Azure | Enterprise-grade by design

Development Tools

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

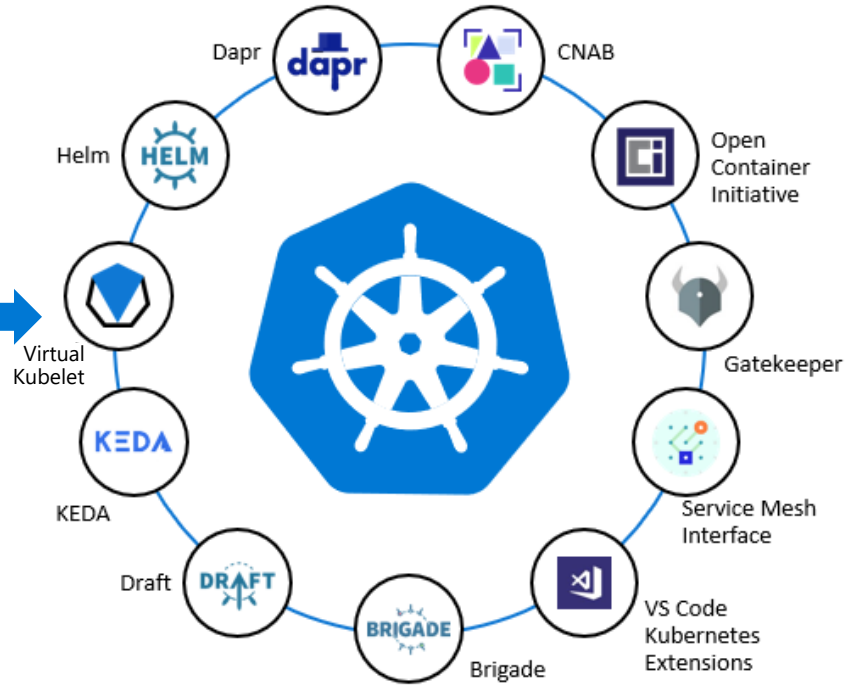
Platform

- 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



Azure Kubernetes Services: Scenario and Architecture

How Azure Kubernetes Services works

Automated upgrades, patches

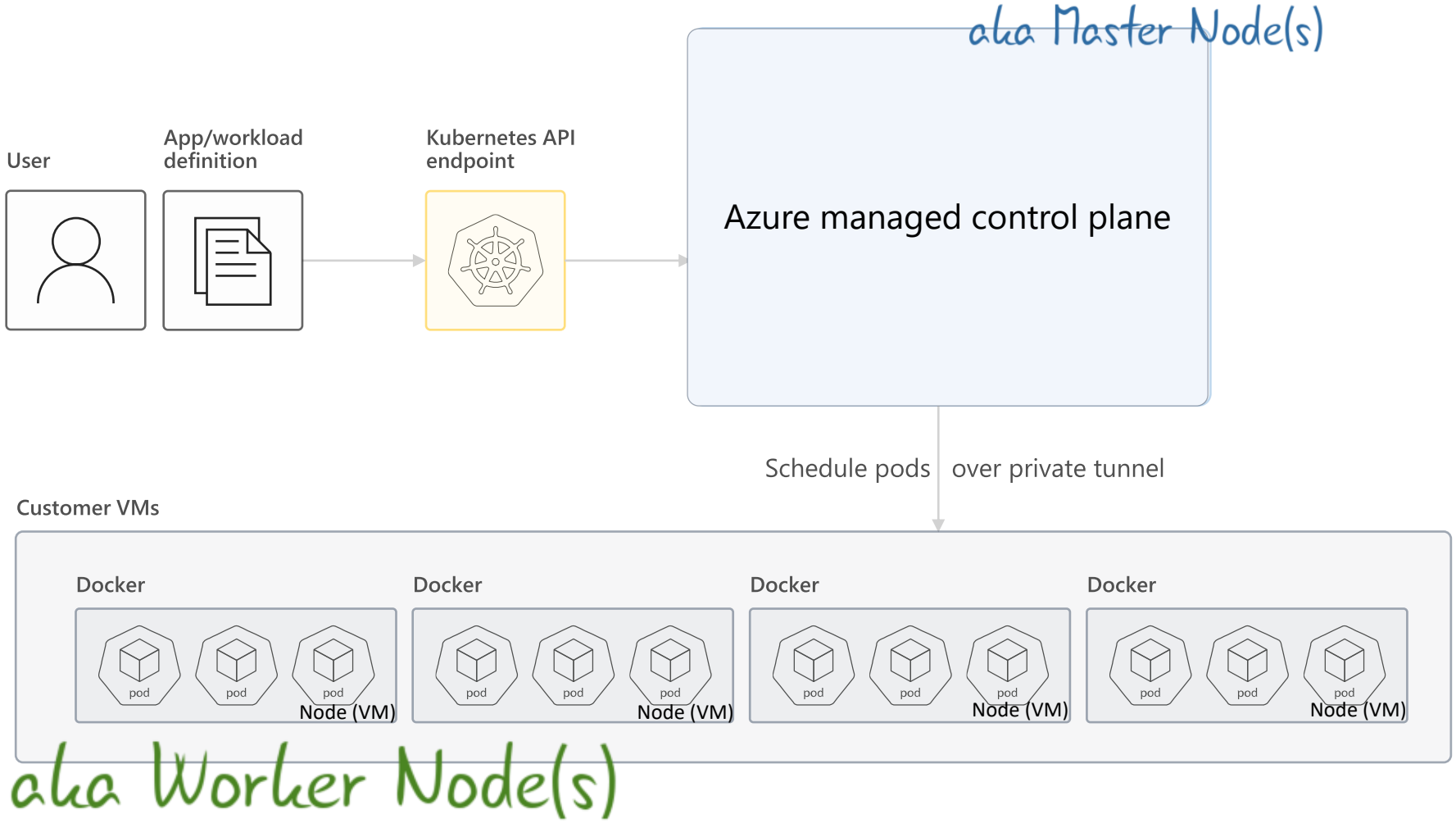
High reliability and availability

Easy and secure cluster scaling

Self-healing

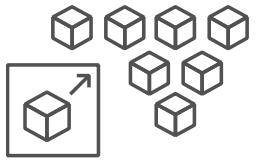
API server monitoring

Control plane at no charge



Top scenarios for Kubernetes on Azure

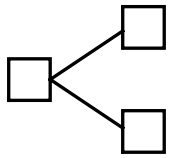
Lift and shift to containers



Cost saving

without refactoring your app

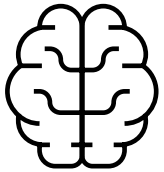
Microservices



Agility

Faster application development

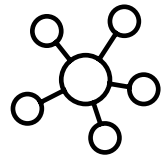
Machine learning



Performance

Low latency processing

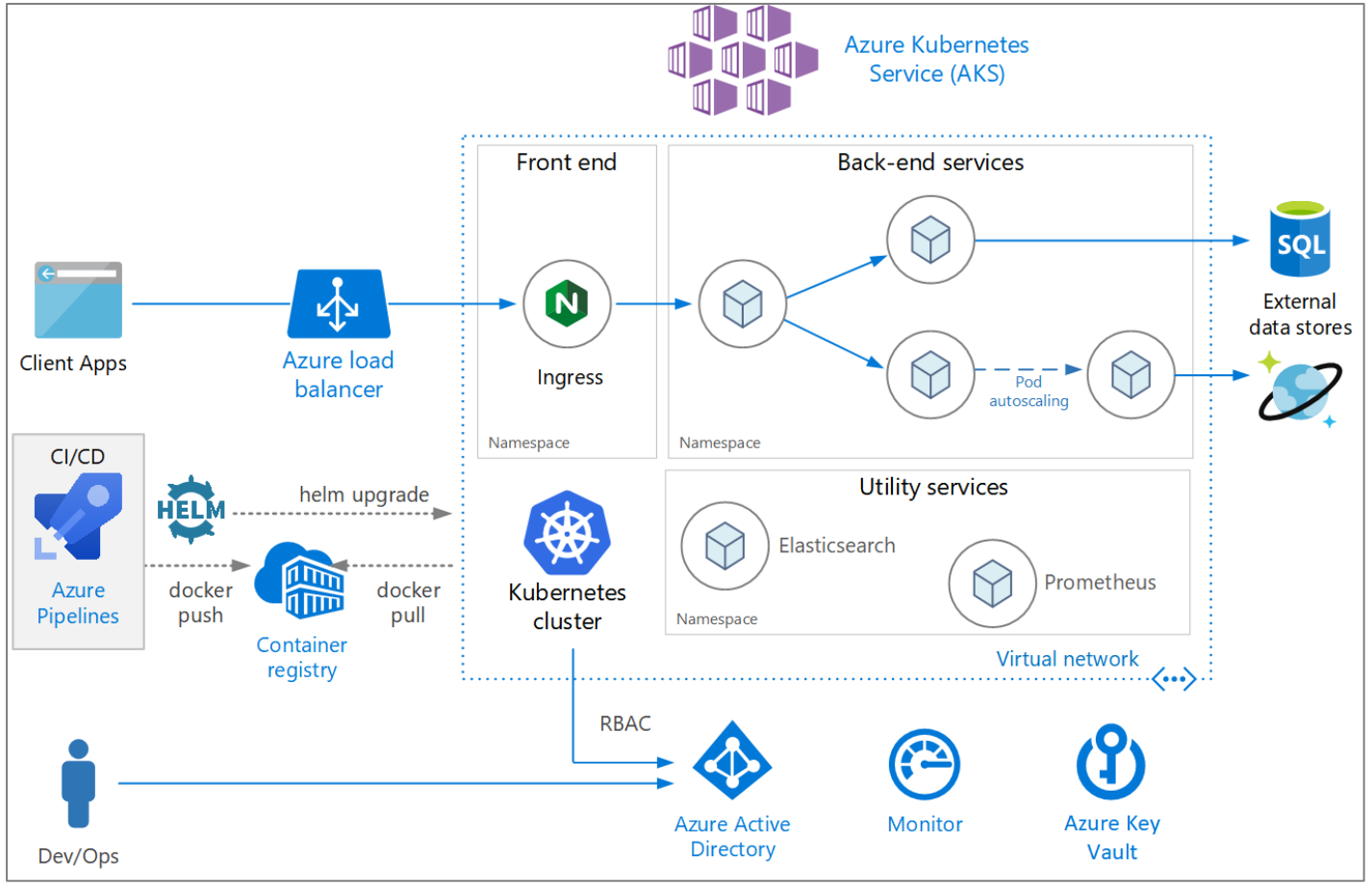
IoT



Portability

Build once, run anywhere

Microservices architecture on Azure Kubernetes Service (AKS)

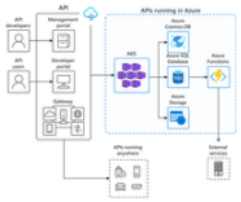


Main component

- **AKS** for hosting microservice apps
- Different **database engines** are used (Cosmos DB, SQL Database)
- **Nginx** is used as Ingress Controller
- **Azure Pipelines** is used as CI/CD
- **Azure Container Registry** is for storing container images
- **AAD** is used for authn / authz
- **Azure Key Vault** is used for storing sensitive info
- **Azure Monitor and Prometheus** for monitoring

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

Reference Architectures for Containers on Azure



API-first SaaS business model

4/17/2020 2 min read

Adapt, evolve, and allow faster innovation to turn opportunities into strategic advantages.

Baseline architecture for an Azure Kubernetes Service (AKS) cluster

8/01/2020 32 min read

Reference architecture for a baseline infrastructure that deploys an Azure Kubernetes Service (AKS) cluster.

Building a telehealth system on Azure

5/25/2019 7 min read

A healthcare system connecting users, devices, and providers built using Kubernetes and PostgreSQL

Bursting from AKS with ACI

12/16/2019 1 min read

Bursting from AKS with ACI

CI/CD pipeline for container-based workloads

7/05/2018 7 min read

Build a DevOps pipeline for a Node.js web app with Jenkins, Azure Container Registry, Azure Kubernetes Service, Cosmos DB, and Grafana.

Elastic demand handling with AKS

4/17/2020 2 min read

Achieve fast and reliable service quality during seasonal and other high-traffic demand periods

Instant IoT data streaming with AKS

4/17/2020 2 min read

Ingest and analyze high volumes of IoT data and generate real-time recommendations

Microservices architecture on Azure Kubernetes Service (AKS)

5/07/2020 17 min read

Deploy a microservices architecture on Azure Kubernetes Service (AKS)

Secure DevOps for AKS

9/10/2020 2 min read

Implementing secure DevOps with Kubernetes on Azure, you can achieve the balance between speed and security and deliver code faster at scale.

Azure Architecture Center:

- Explain the data flow, network topology, architecture and design consideration, component used, etc.
- https://docs.microsoft.com/en-us/azure/architecture/browse/#containers

AKS Best Practices

Cluster operator and developer best practices to build and manage applications on Azure Kubernetes Service

Cluster operator best practices

- **Multi-tenancy**
  - Best practices for cluster isolation
  - Best practices for basic scheduler features
  - Best practices for advanced scheduler features
  - Best practices for authentication and authorization
- **Security**
  - Best practices for cluster security and upgrades
  - Best practices for container image management and security
  - Best practices for pod security
- **Network and storage**
  - Best practices for network connectivity
  - Best practices for storage and backups
- **Running enterprise-ready workloads**
  - Best practices for business continuity and disaster recovery

Developer best practices

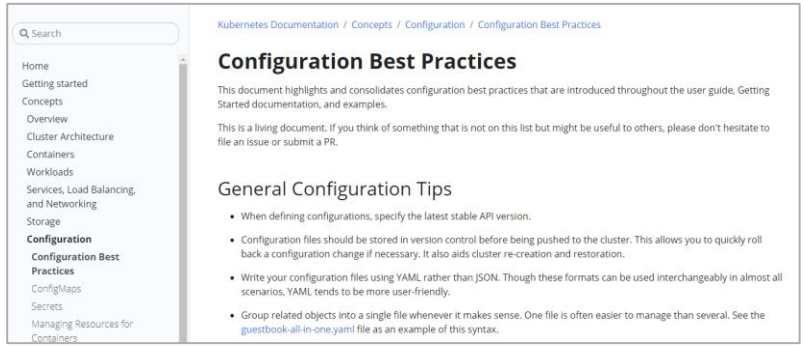
- Best practices for application developers to manage resources
  - Includes defining pod resource requests and limits, configuring development tools, and checking for application issues.
- Best practices for pod security
  - Includes securing access to resources, limiting credential exposure, and using pod identities and digital key vaults.

https://docs.microsoft.com/en-us/azure/aks/best-practices

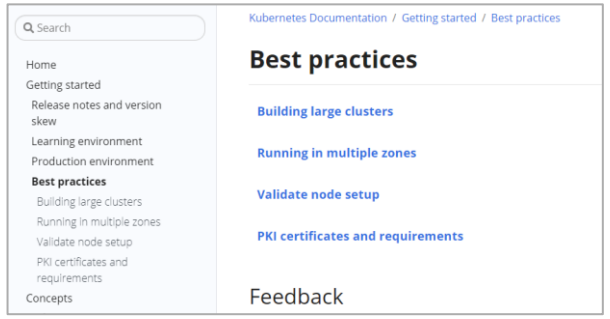
Don’t forget: General best practices still apply

Cluster operator best practices

- Kubernetes best practices applies



https://kubernetes.io/docs/concepts/configuration/overview/

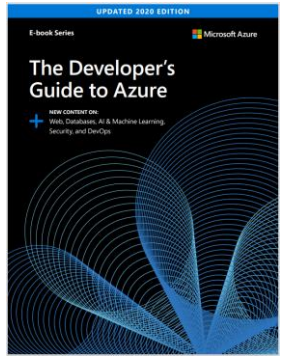


https://kubernetes.io/docs/setup/best-practices/

Developer best practices

- Common application development best practices still apply
  - Writing secured apps, resource-efficient apps

https://docs.microsoft.com/en-us/azure/security/develop/secure-dev-overview



https://azure.microsoft.com/en-us/campaigns/developer-guide/

Best practices #1: Set resource requests and limits of the pods

- Analogy:

- What happens if you don't?
  - Default requests and limits will be used
  - Reduced efficiency (over/under-utilized)
  - Performance might degrade

- Define pod resource requests and limits

YAML

```yaml
kind: Pod
apiVersion: v1
metadata:
  name: mypod
spec:
  containers:
  - name: mypod
    image: mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine
    resources:
      requests:
        cpu: 100m
        memory: 128Mi
      limits:
        cpu: 250m
        memory: 256Mi
```

Define a set amount of CPU and memory that pod needs on a regular basis.

The maximum of CPU and memory that a pod can use.

Best practices #2: Enforcement and Detection for #1

- Enforcement
  - Leverage LimitRange object in Kubernetes
    - It enforces defaults & limits for all Pods does not exceed resource minimum, maximum and ratio defined
    - When a LimitRange is activated resources, users must specify requests or limits. Otherwise, the system may reject Pod creation.

```yaml
apiVersion: v1
kind: LimitRange
metadata:
  name: mem-limit-range
spec:
  limits:
  - default:
      memory: 512Mi
    defaultRequest:
      memory: 256Mi
    type: Container
```

- Detection
  - Kube-Advisor:
    - AKS open-source project that scans a Kubernetes cluster and reports on issues that it finds.
    - One useful check is to identify pods that don't have resource requests and limits in place.

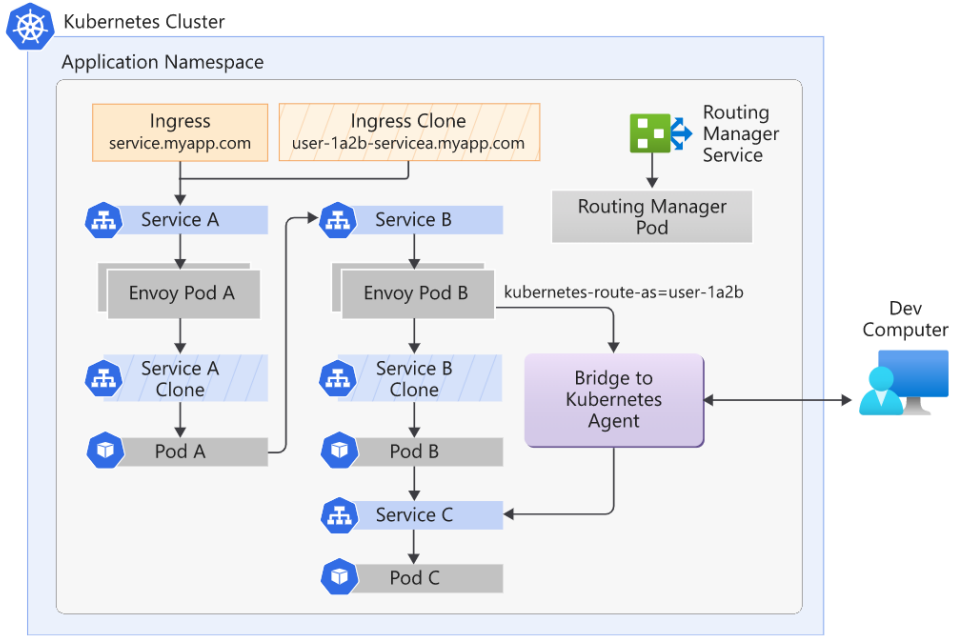
NAMESPACE	POD NAME	POD CPU/MEMORY	CONTAINER	ISSUE
bikeapp	bikes-86dfbbd9c7-2b8vs	3822936n / 43100Ki	bikes	Memory Resource Limits Missing
				CPU Request Limits Missing
				Memory Request Limits Missing
				CPU Resource Limits Missing

ISSUE	REMEDIATION
CPU Request Limits Missing	Consider setting resource and request limits to prevent resource starvation: https://kubernetes.io/docs/concepts/configuration/manage-compute-resources-container/
Memory Request Limits Missing	
CPU Resource Limits Missing	
Memory Resource Limits Missing	

Best practices #3: Use dev tools for AKS

- Kubernetes Extension for VS Code
  - Works with any Kubernetes anywhere (Azure, Minikube, AWS, GCP and more!).
  - Open source
  - Dependencies: kubectl, docker / buildah
- Key capabilities:
  - View your clusters in an **explorer tree view**, and drill into objects.
  - **Browse** Helm repos and **install** charts into your cluster.
  - **Intellisense** for Kubernetes resources, Helm charts, & templates.
  - Edit Kubernetes **resource manifests** and apply them to your cluster.
  - Get or follow logs and events from your clusters.
  - Forward local ports to your application's pods.

- Develop and debug against an AKS cluster with Bridge to Kubernetes



Visual Studio | Marketplace

Sign in

Visual Studio Code > Snippets > Kubernetes

New to Visual Studio Code? Get it now.

Kubernetes

Microsoft | 750,828 installs | ★★★★★ (21) | Free

Develop, deploy and debug Kubernetes applications

Install

Trouble Installing?

Demo

Kubernetes Extension in VS Code

Overview | Version History | Q & A | Rating & Review

Visual Studio Code Kubernetes Tools

build passing

The extension for developers building applications to run in Kubernetes clusters and for DevOps staff troubleshooting Kubernetes applications.

Works with any Kubernetes anywhere (Azure, Minikube, AWS, GCP and more!).

Features include:

- View your clusters in an explorer tree view, and drill into workloads, services, pods and nodes.
- Browse Helm repos and install charts into your Kubernetes cluster.
- Intellisense for Kubernetes resources and Helm charts and templates.

Snippets | Linters | Debuggers | Azure

Tags

aks | aws | debuggers | gke | helm | helm-template | ignore | keybindings | kubernetes | snippet | yaml

Resources

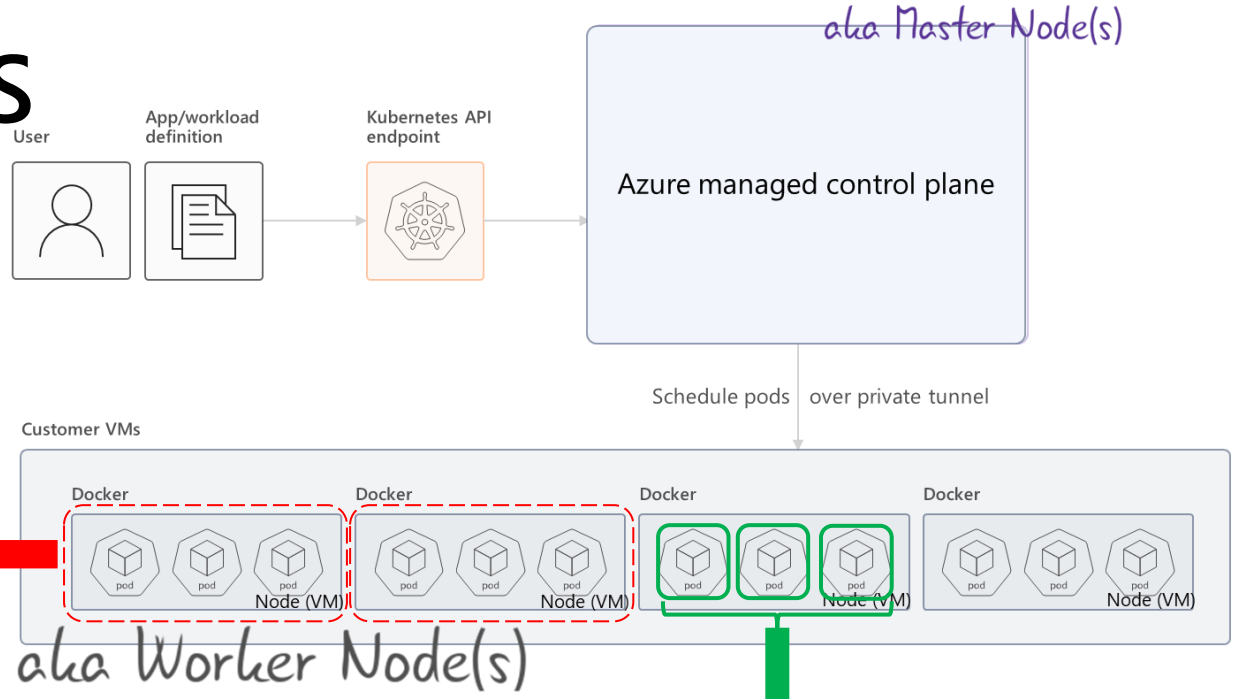
Repository

License

Changelog

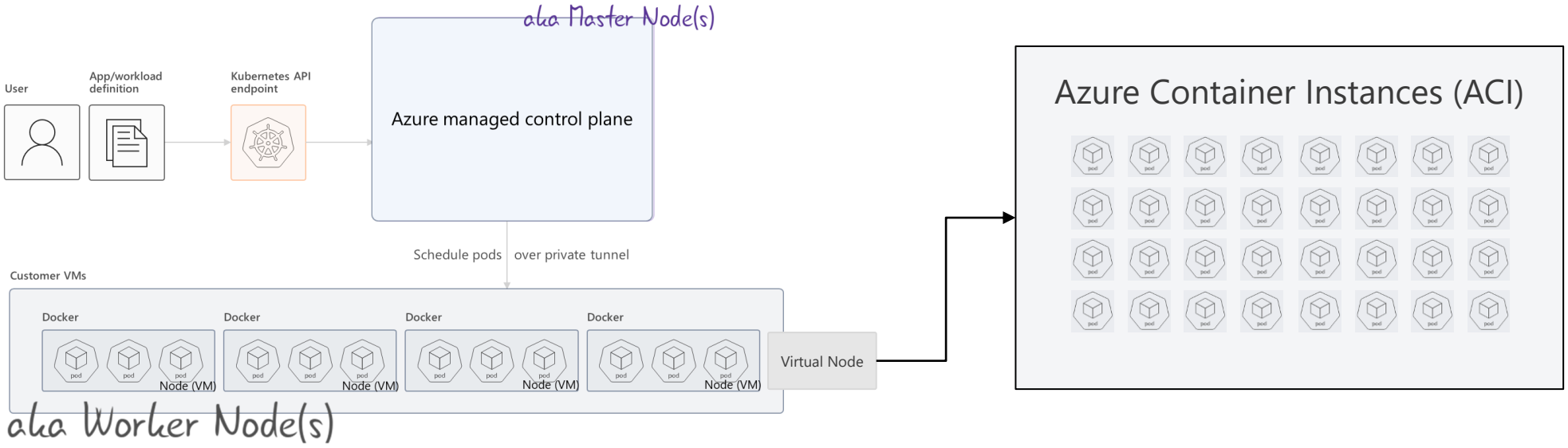
Scaling apps in AKS

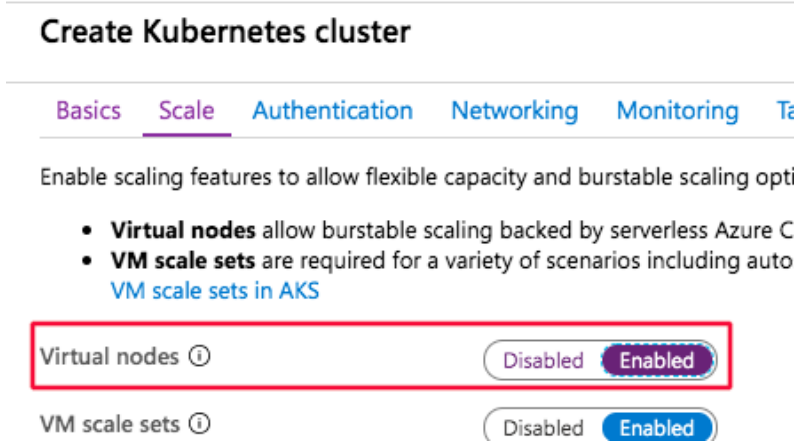
Scaling apps in AKS



	(Worker) Node Level	Pod Level
Manually	Thru CLI or Azure Portal **az aks scale --name MyAKSCluster --node-count 3**	Thru kubectl or Azure Portal **kubectl scale --replicas=3 -f foo.yaml** **kubectl scale deploy/nginx --replicas=3**
Automatically	Cluster Auto-scaler **az aks nodepool update --cluster-name myAKSCluster --name nodepool1** **--min-count 1 --max-count 5**	Horizontal Pod Auto-scaler **kubectl autoscale deployment foo --min=2 --max=10**

Scaling in AKS with Virtual Nodes





Enable Virtual Nodes on existing cluster

Azure CLI

```
az aks enable-addons \
    --resource-group myResourceGroup \
    --name myAKSCluster \
    --addons virtual-node \
    --subnet-name myVirtualNodeSubnet
```

Console

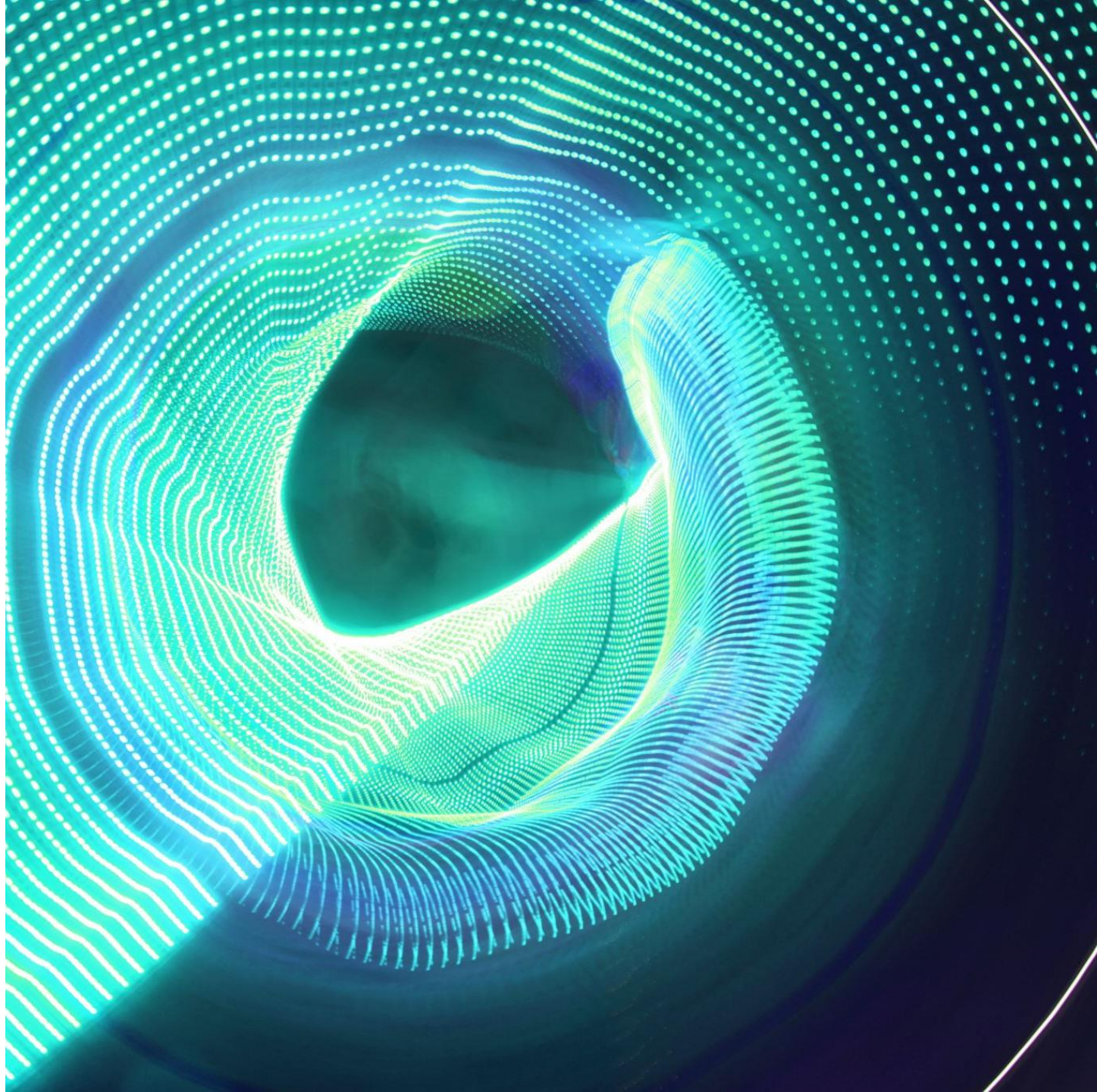
```
kubectl get nodes
```

Output

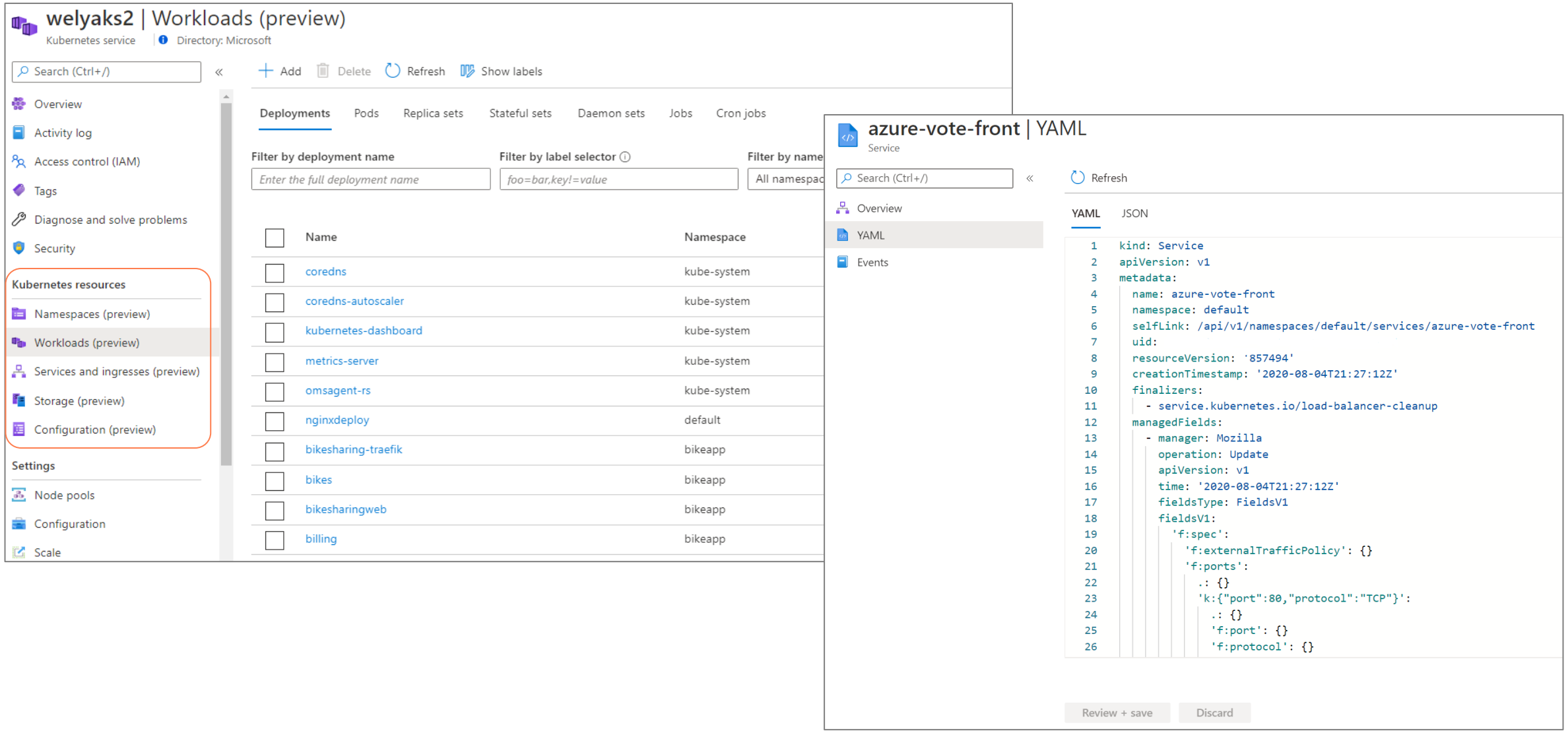
```
NAME                          STATUS   ROLES   AGE   VERSION
virtual-node-aci-linux        Ready    agent   28m   v1.11.2
aks-agentpool-14693408-0      Ready    agent   32m   v1.11.2
```

What’s new in AKS?

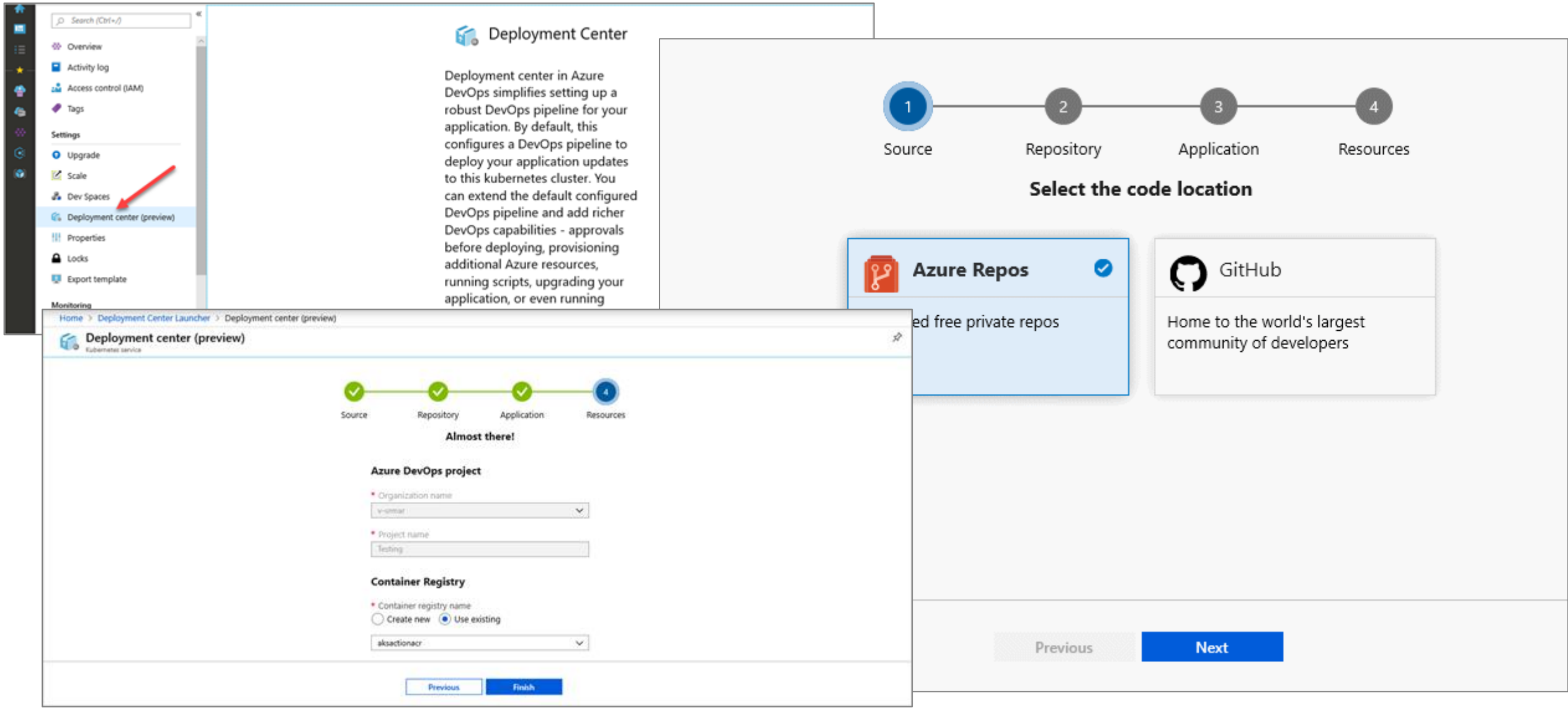
- Kubernetes resources view in Azure Portal (Preview Sept 2020)
- Deployment Center (July 2019)
- Azure Monitor for Containers (improved)
- Windows Server Container General Available (April 2020)



Kubernetes resources view in Azure Portal

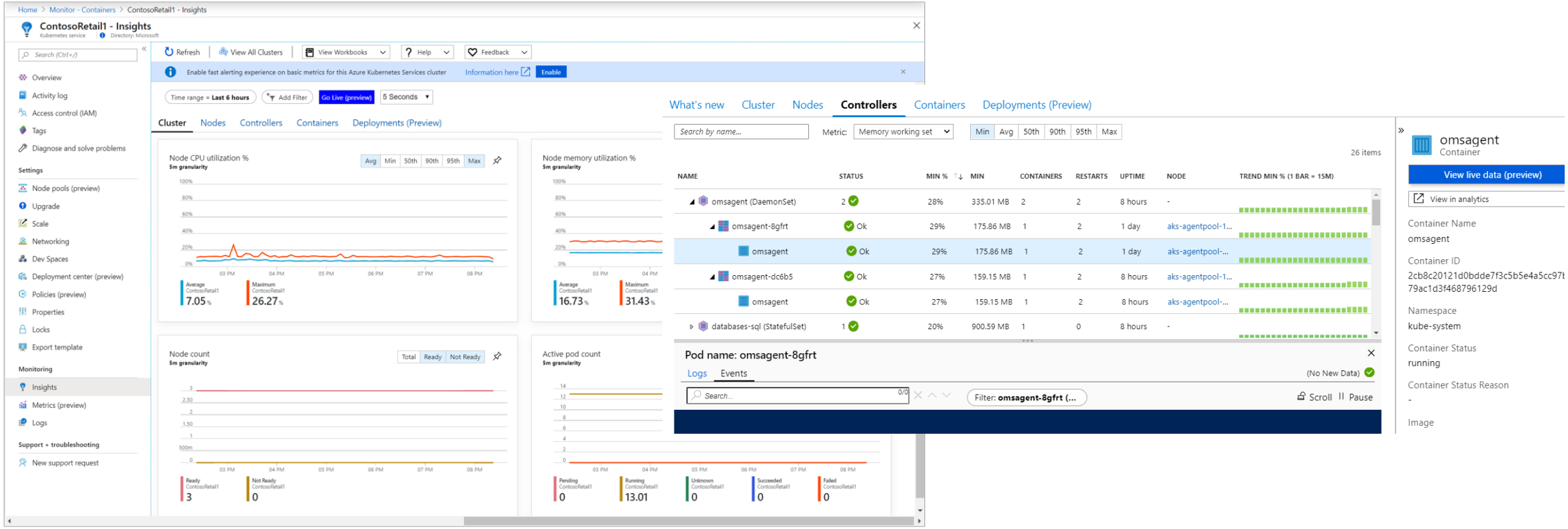


Deployment Center in AKS



Azure Monitor for Containers

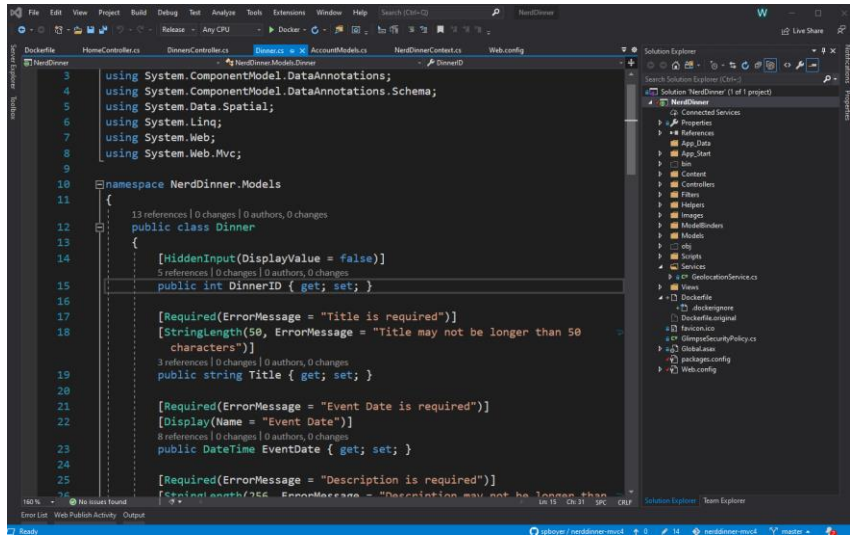
- Infrastructure and Cluster Level Logs / Metrics



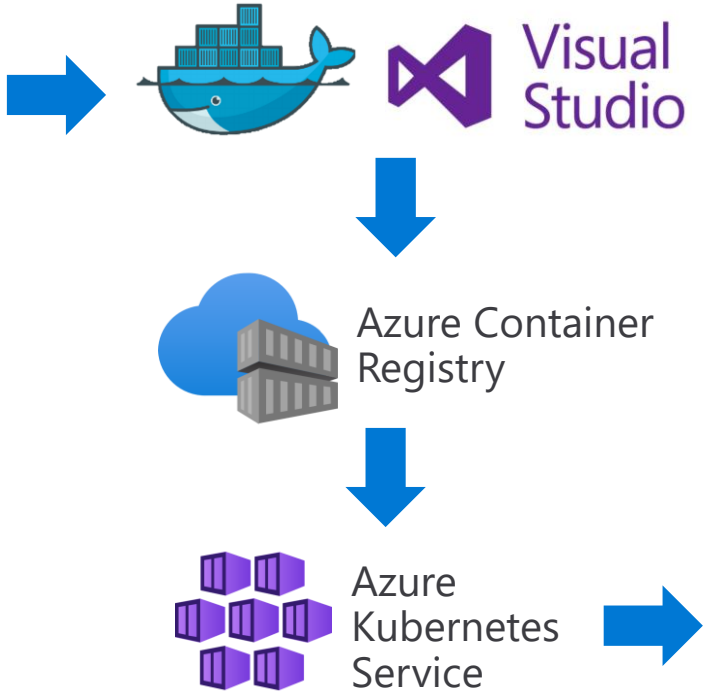
- Application-Level Monitor with App Insight

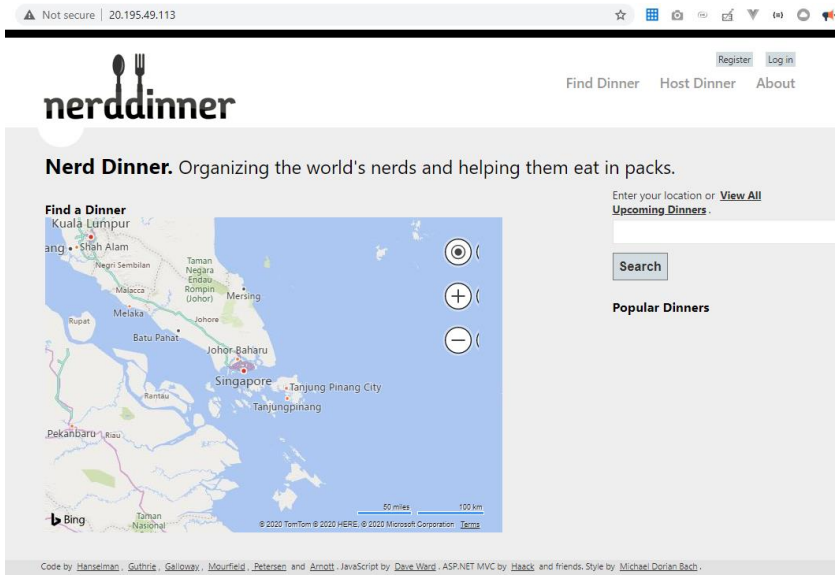
Windows Server Container General Available on AKS

- Enables the scenario of lift-and-shift Windows apps (particularly .NET Framework)
- Add Node Pool with Windows as the OS

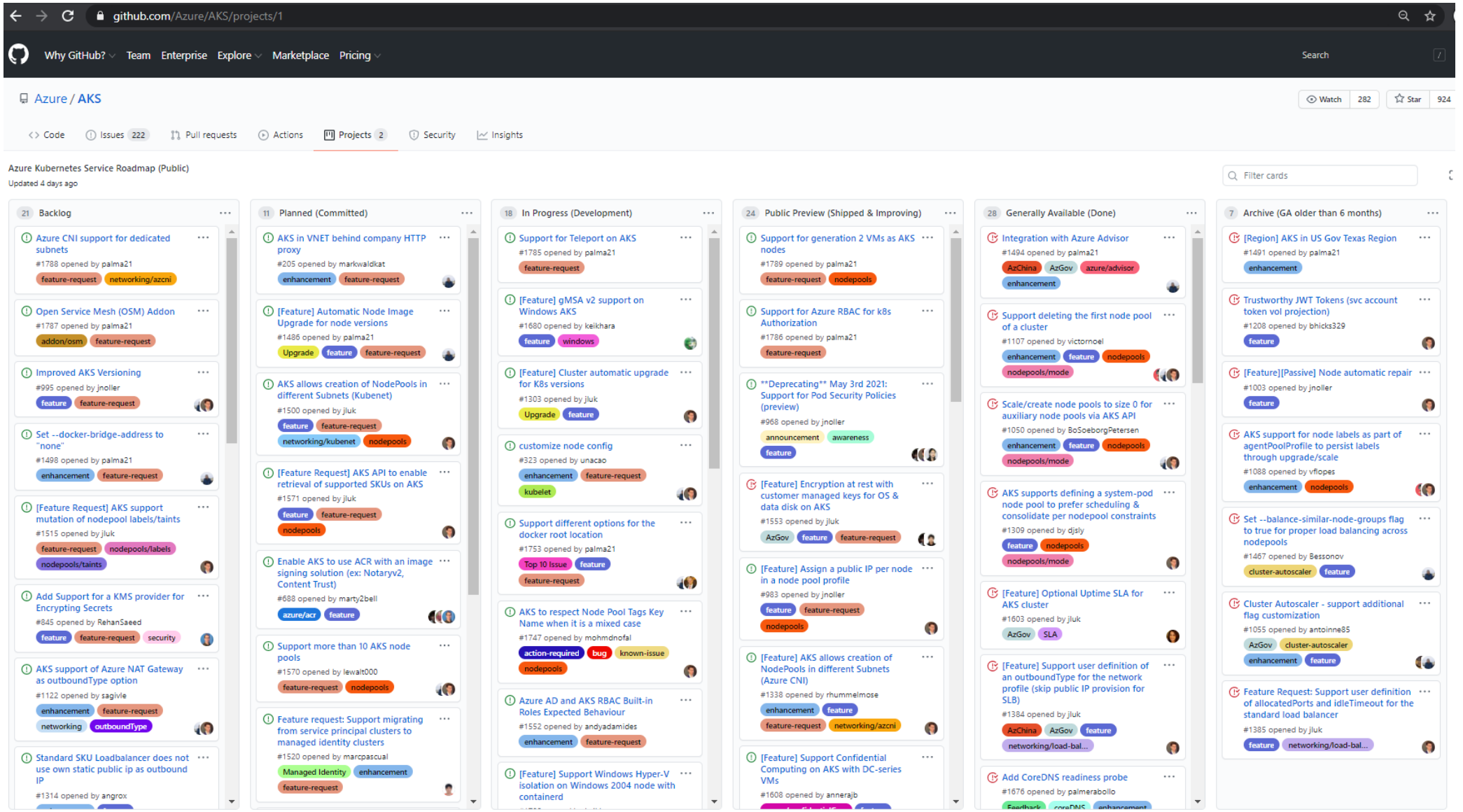


Legacy ASP.NET Apps





Public Roadmap: https://github.com/Azure/AKS/projects/1



What to expect in the next session

- Developing Microservices Architecture on Azure
- 10th November 2020 @ 10:15 Singapore Time

- We'll be covering:
  - Microservices architecture concepts
  - Critical success criteria
  - Azure services for building Microservices architecture
  - Developing Microservices on Azure with Bridge to Kubernetes

References and Learn More

- https://aka.ms/LearnKubernetes
- https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/microservices/aks
- https://docs.microsoft.com/en-us/azure/architecture/browse/#containers
- https://docs.microsoft.com/en-us/azure/security/develop/secure-dev-overview
- https://azure.microsoft.com/en-us/campaigns/developer-guide/
- https://kubernetes.io/docs/concepts/configuration/overview/
- https://kubernetes.io/docs/setup/best-practices/
- https://docs.microsoft.com/en-us/azure/aks/best-practices
- https://github.com/Azure/kube-advisor
- Public Roadmap:
  - https://github.com/Azure/AKS/projects/1
- Auto-scaling with Virtual Nodes in AKS
  - https://www.youtube.com/watch?v=vZp71F3xDNc
- How to monitor your Kubernetes cluster
  - https://www.youtube.com/watch?v=RjsNmapggPU