

Cloud Native Application Development with Azure Kubernetes Service

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

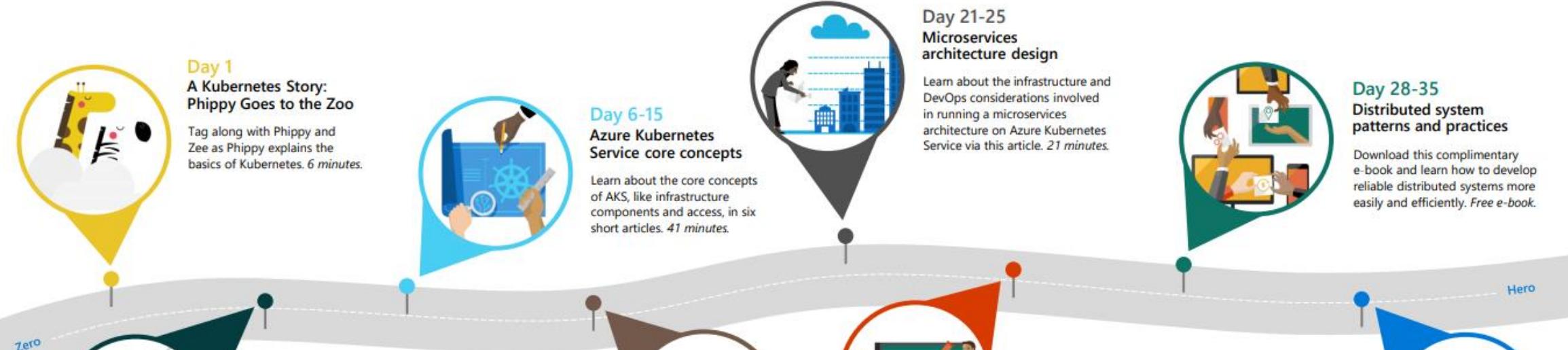
Disclaimer

- 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:

50 days from zero to hero with Kubernetes



Click the bubbles to access that resource



<https://aka.ms/LearnKubernetes>

Agenda

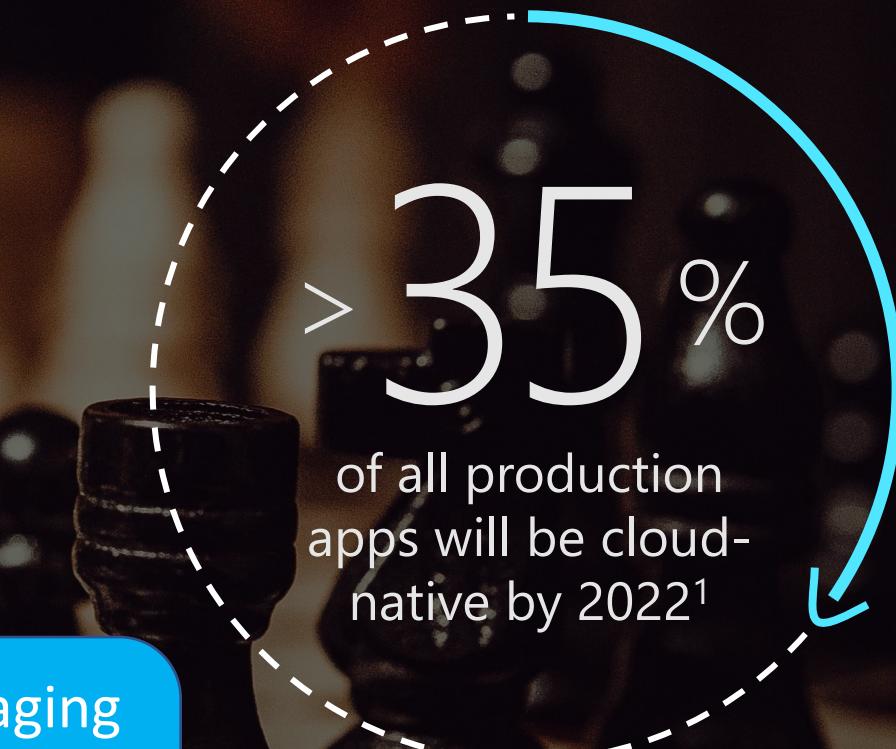
- Demystifying Cloud Native
- Crash course: Containers, Docker, and Kubernetes
- Kubernetes on Azure: top scenarios and reference architecture
- Best practices
- Scaling apps on AKS (Azure Kubernetes Services)
- What's new in AKS?
- What to expect in the next session
- Reference and learn more

Demystify Cloud Native



Cloud native
is the new
paradigm
of software
development

Though still leveraging
the existing good
application practices
such as 12factor-app



> 35%
of all production
apps will be cloud-
native by 2022¹

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

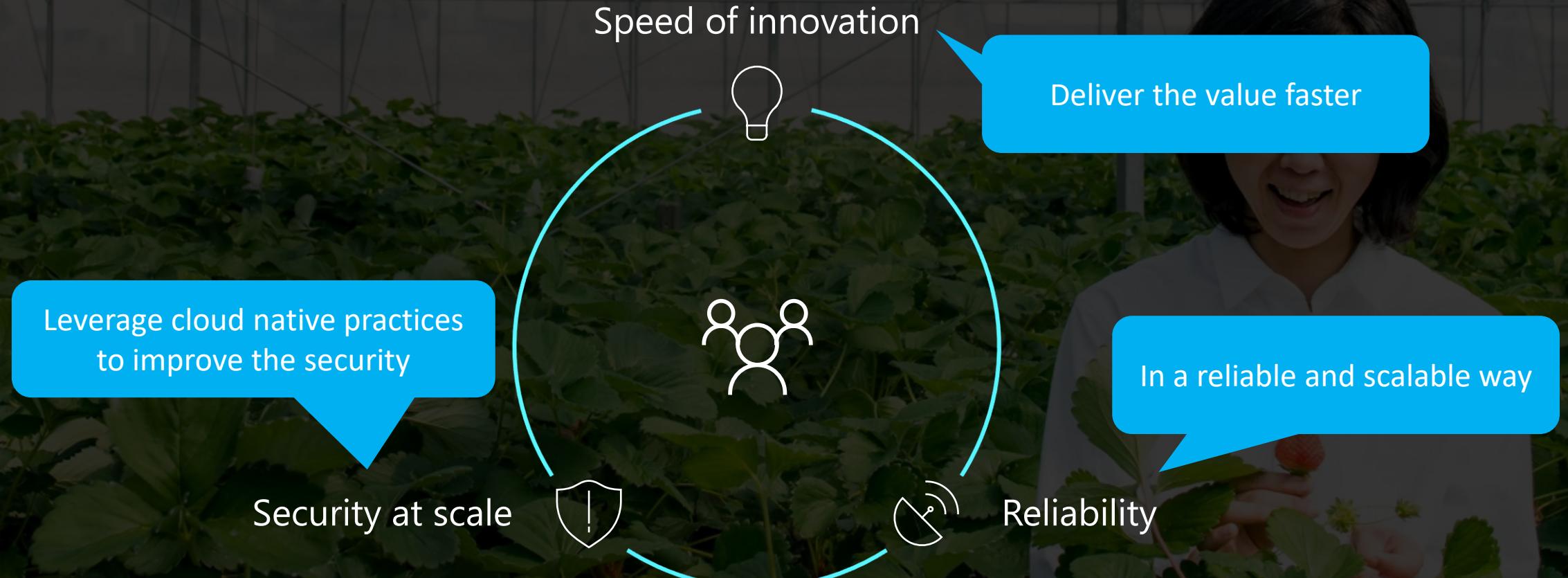


DevOps

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 experience¹
- 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

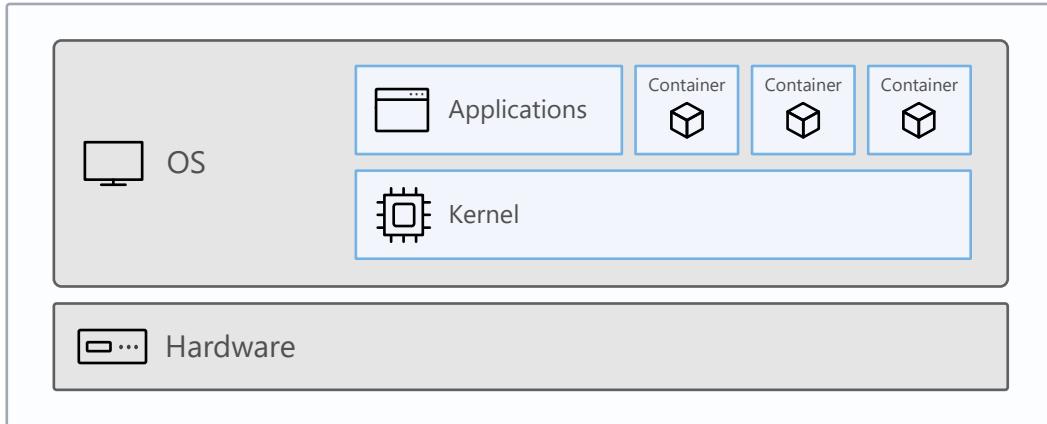
- \$1BN investment every year in security
- >90 compliance certifications
- Out of the box integration with Azure Policy, Active Directory and Security Center

Crash Course: Containers, Docker, and Kubernetes 101

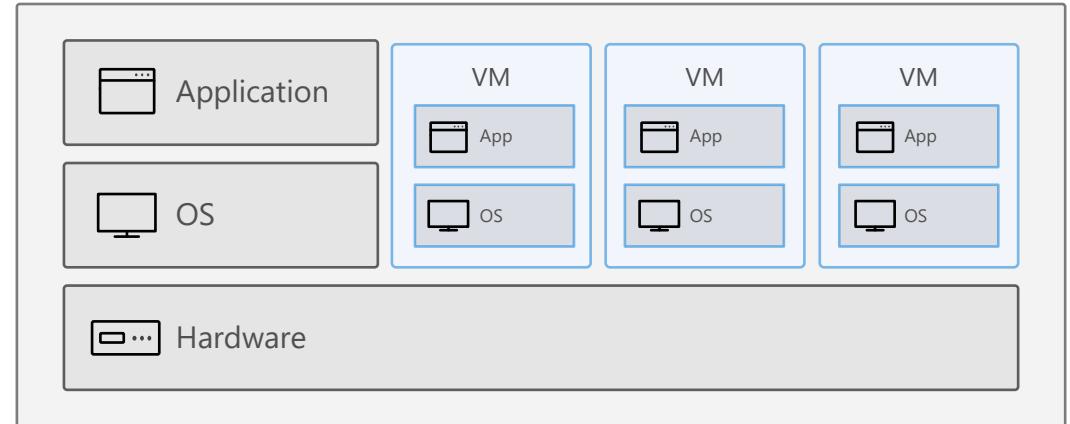
What is a container? (container vs VM)

- Container is an application packaging and deployment mechanism

Containers = operating system virtualization



Traditional virtual machines = hardware virtualization



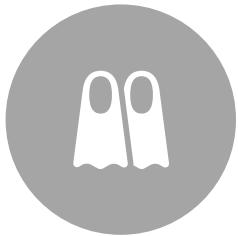
- Shared Host OS
- Near instant start-up
- Dependent app services and libraries are tied to container (layers)

- 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



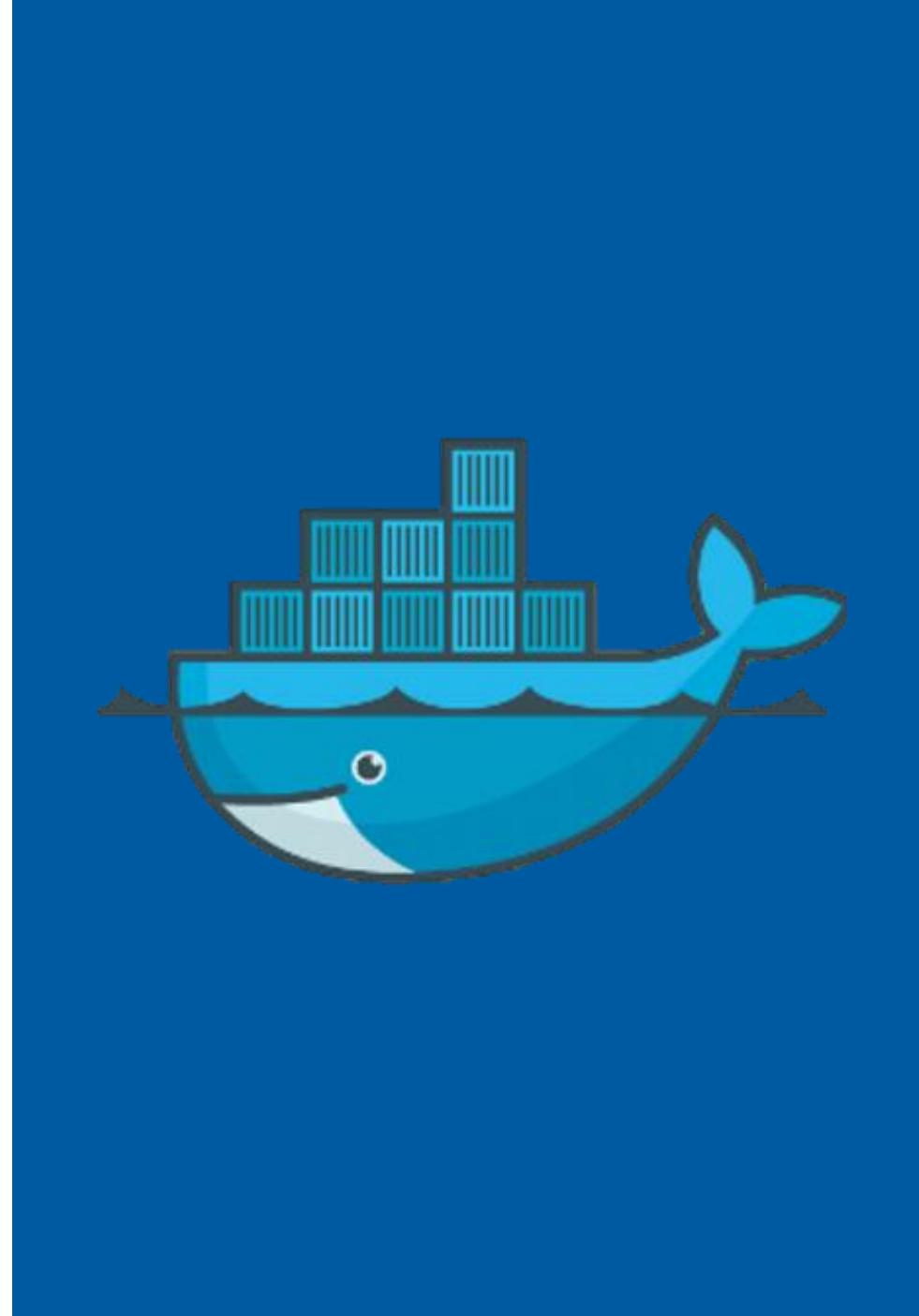
CONSISTENCY ACROSS
DEVELOPMENT, TEST, &
PRODUCTION



EASE OF SCALING
THROUGH CLOUD
ELASTICITY

Docker

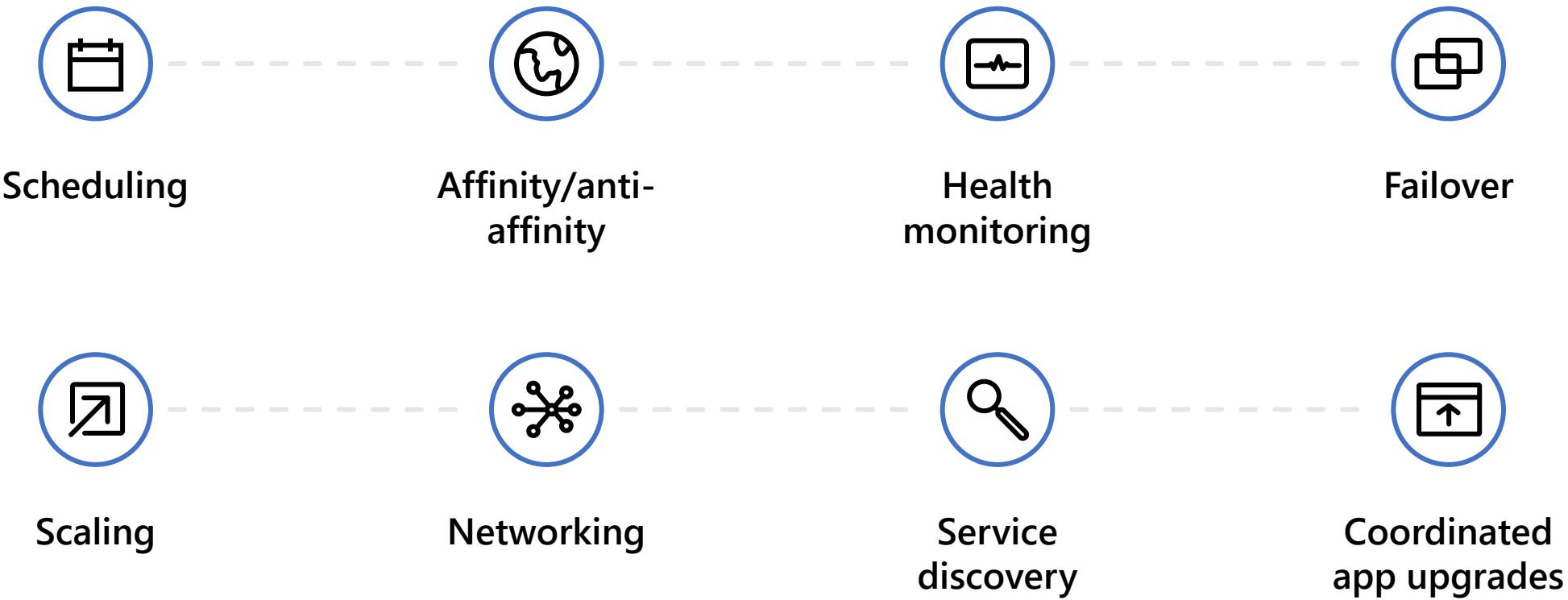
- Docker is the “de-facto” container format and sets of tools / APIs, opened-source, and managed by Docker Inc.
- Docker tools:
 - Docker CLI, Docker Engine, Docker Swarm, Docker Compose, etc.
- Can run on-premise or cloud
- Started in Linux, but later in Windows (Win 10 or Win Server 2016)
- Comes in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



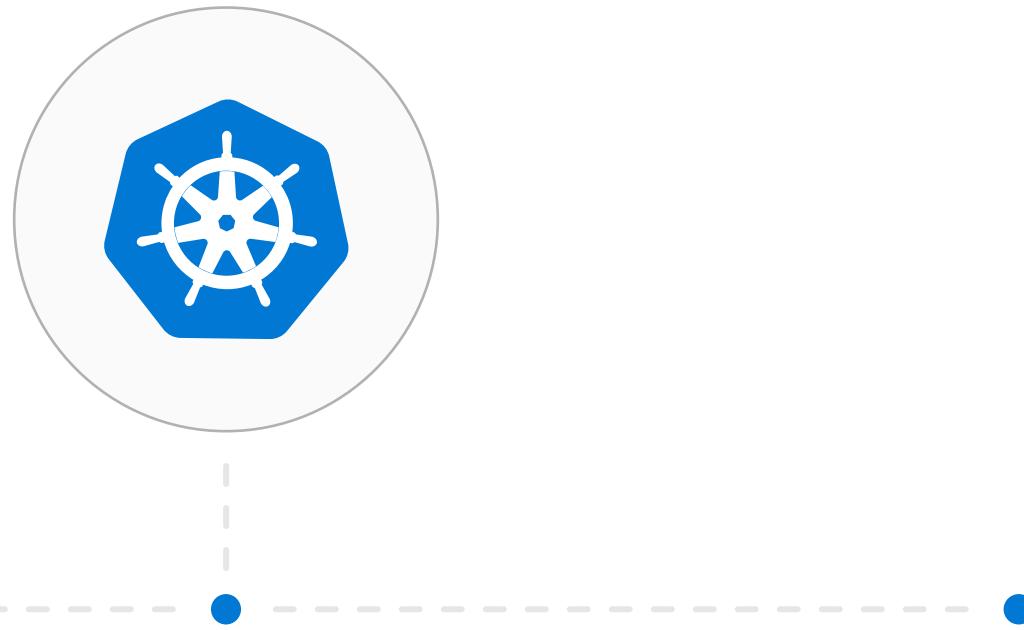
Question:

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

The elements of **orchestration**



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



Contributors 2,905

+ 2,894 contributors

<https://github.com/kubernetes/kubernetes>

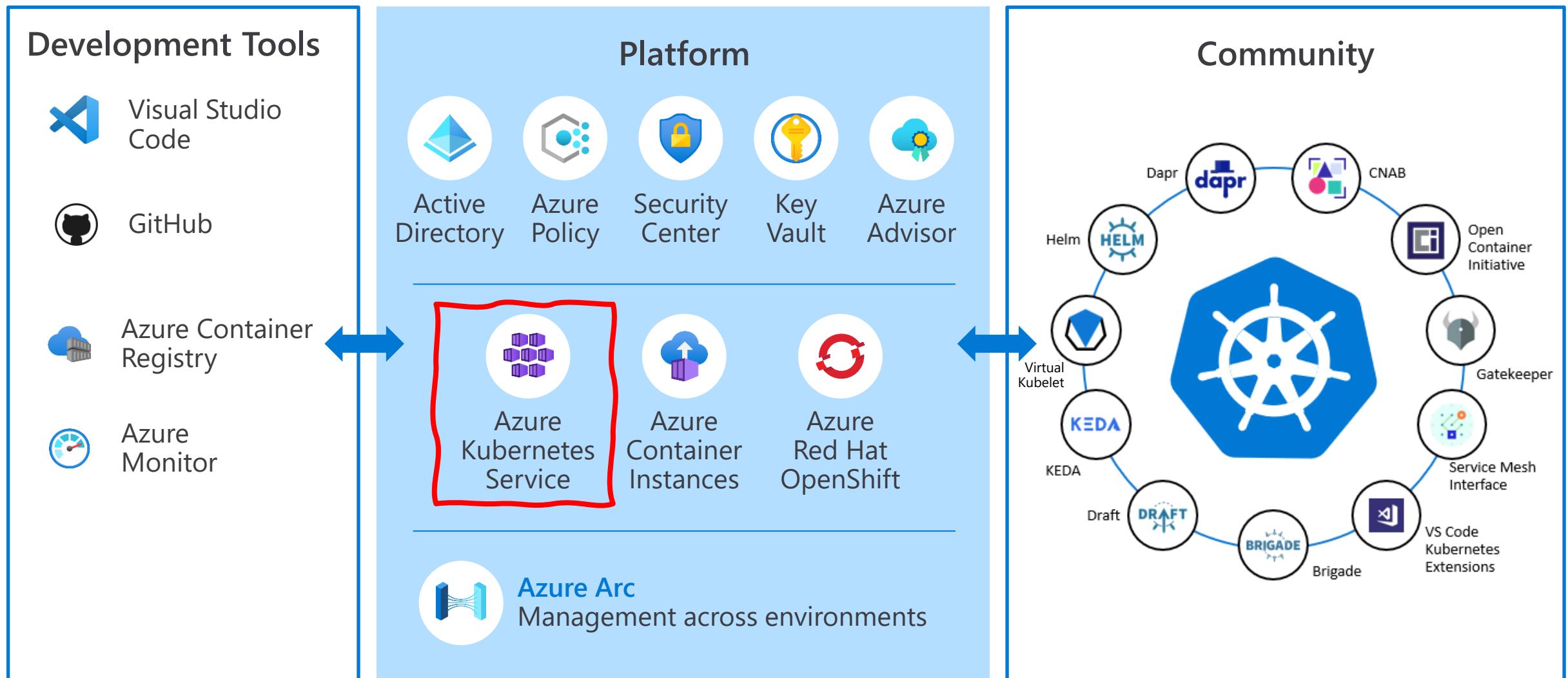
A circular GitHub contributor profile for the Kubernetes repository. It shows a grid of 12 small profile pictures of contributors, with a larger grid of 12 more below it. A blue callout bubble in the top right corner indicates there are 2,894 more contributors. Below the grid is the repository URL.

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

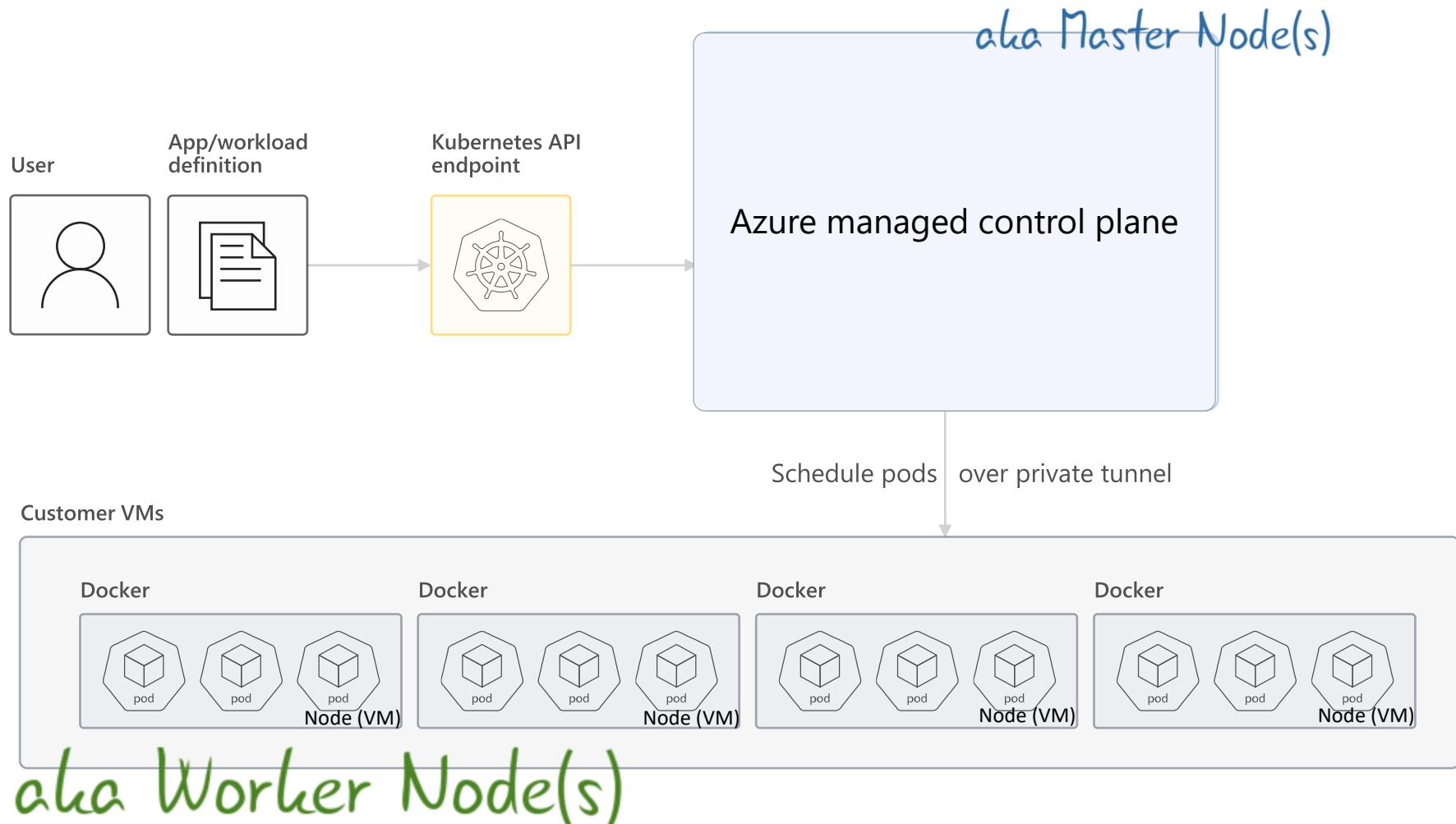
Kubernetes on Azure | Enterprise-grade by design



Azure Kubernetes Services: Scenario and Architecture

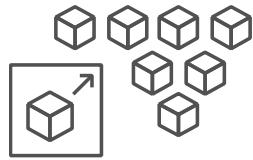
How Azure Kubernetes Services works

- 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

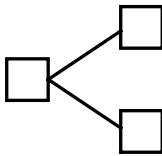
Lift and shift
to containers



Cost saving

without refactoring your app

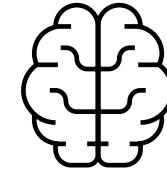
Microservices



Agility

Faster application
development

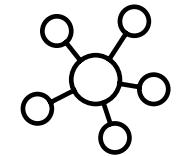
Machine
learning



Performance

Low latency processing

IoT

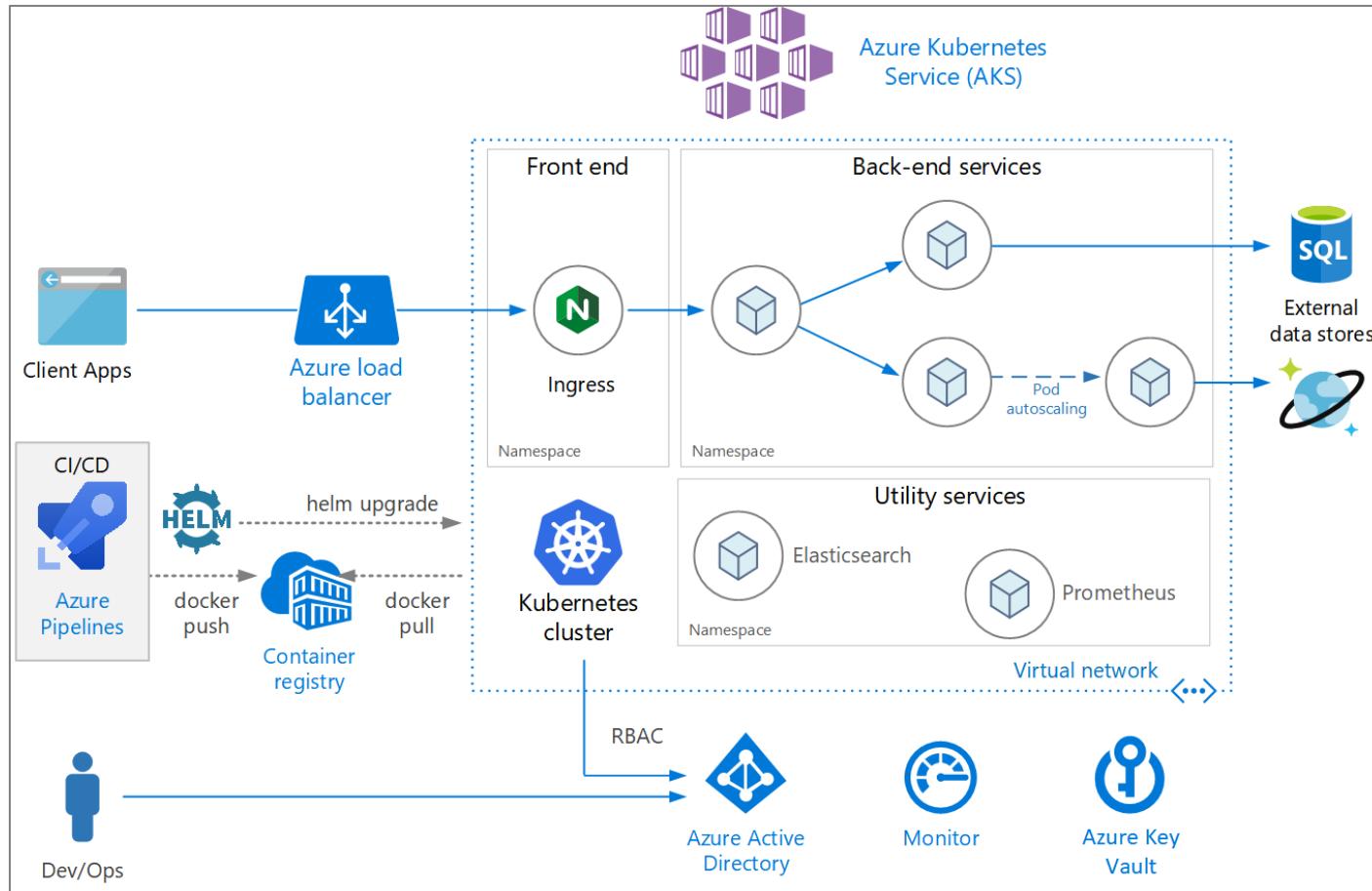


Portability

Build once, run
anywhere



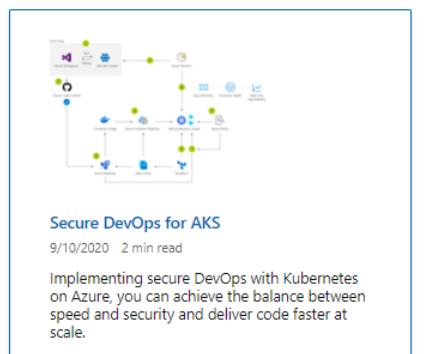
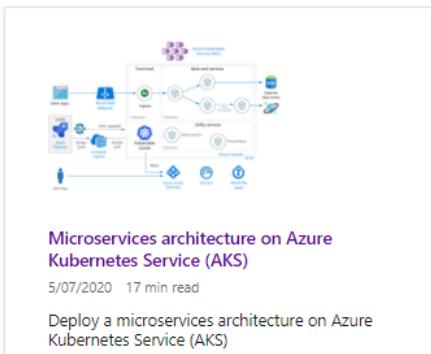
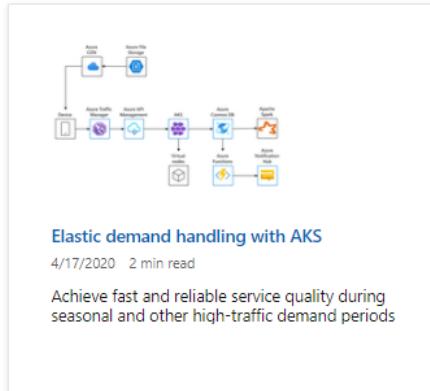
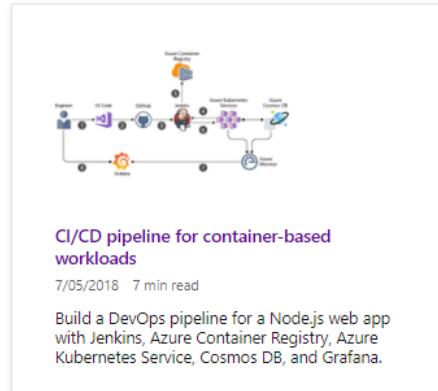
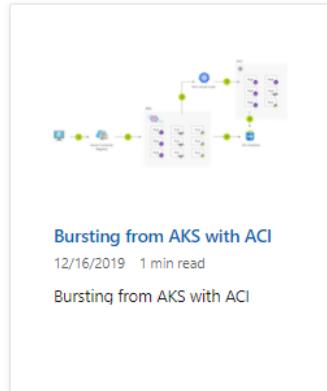
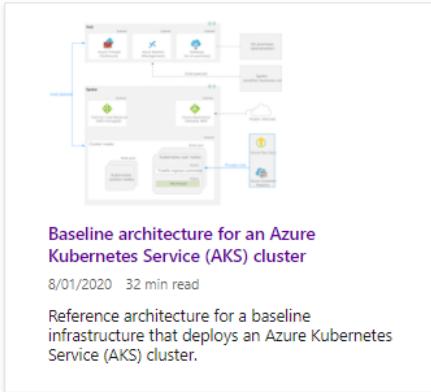
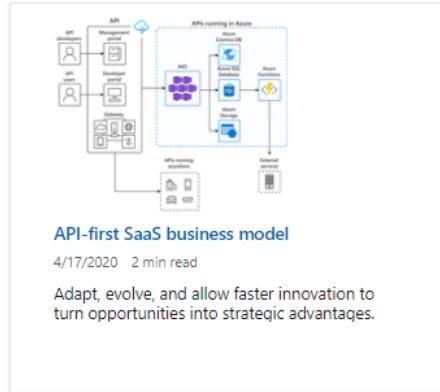
Microservices architecture on Azure Kubernetes Service (AKS)



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

Reference Architectures for Containers on Azure



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.

Don't forget: General best practices still apply

Cluster operator best practices

- Kubernetes best practices applies

The screenshot shows the 'Configuration Best Practices' page from the Kubernetes Documentation. The left sidebar includes links for Home, Getting started, Concepts, Overview, Cluster Architecture, Containers, Workloads, Services, Load Balancing, and Networking, Storage, Configuration, Configuration Best Practices, ConfigMaps, Secrets, and Managing Resources for Containers. The main content area is titled 'Configuration Best Practices' and discusses highlights and consolidates configuration best practices. It includes a note about being a living document and a section on 'General Configuration Tips' with a bulleted list of best practices.

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

Developer best practices

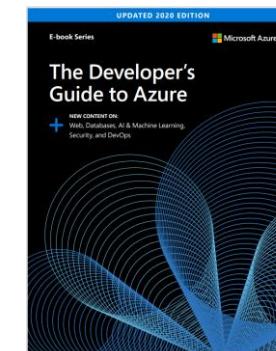
- Common application development best practices still apply
- Writing secured apps, resource-efficient apps

The screenshot shows the 'Secure development best practices on Azure' page from the Azure Documentation. The left sidebar includes links for Azure, Product documentation, Architecture, Learn Azure, Develop, and Resources. Under 'Develop', there are sub-links for Filter by title, Concepts, Best practices, Overview, Design secure apps, Develop secure apps, and Deploy secure apps. The main content area is titled 'Secure development best practices on Azure' and discusses security activities and controls for cloud development, mentioning the Microsoft Security Development Lifecycle (SDL) and security questions and concepts.

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

The screenshot shows the 'Best practices' page from the Kubernetes Documentation. The left sidebar includes links for Home, Getting started, Release notes and version skew, Learning environment, Production environment, Best practices, Concepts, and Feedback. The main content area is titled 'Best practices' and lists several sections: Building large clusters, Running in multiple zones, Validate node setup, PKI certificates and requirements, and Feedback.

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



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

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

- Analogy:



Me: How many people and would you like to reserve for your table?

Guest: I am not telling you.

Me: WHAT!!!

- 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

```
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.

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

- Detection

- Kube-Advisor:

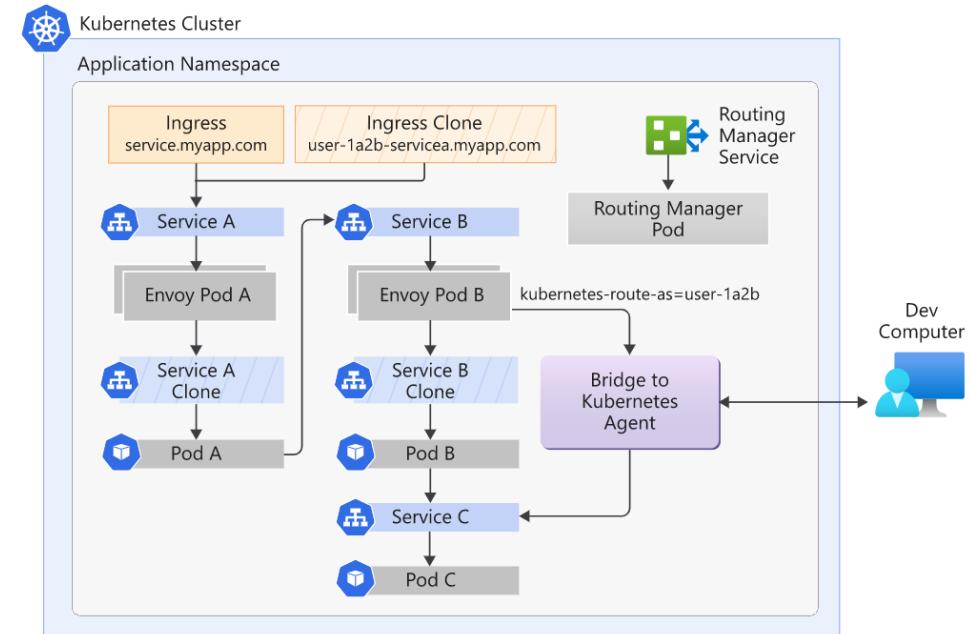
- AKS open-source project that scans a Kubernetes cluster and reports on issues that it finds.
- One useful check is to identify pods that don't have resource requests and limits in place.

NAMESPACE	POD NAME	POD CPU/MEMORY	CONTAINER	ISSUE
bikeapp	bikes-86dfbbcd9c7-2b8vs	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-containers/
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





Kubernetes

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

Develop, deploy and debug Kubernetes applications

[Install](#)[Trouble Installing?](#)[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.

Demo

Kubernetes Extension in VS Code

Tags

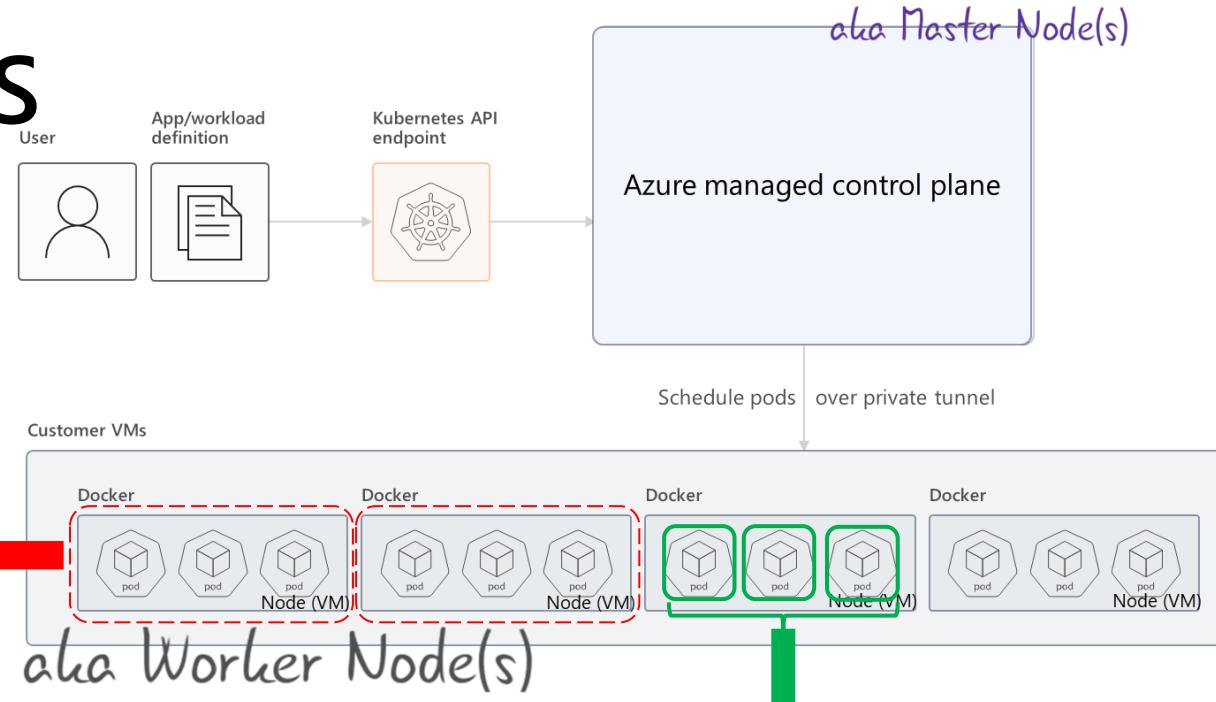
aks aws debuggers gke helm helm-template
ignore keybindings kubernetes snippet yaml

Resources

[Repository](#)[License](#)[Changelog](#)

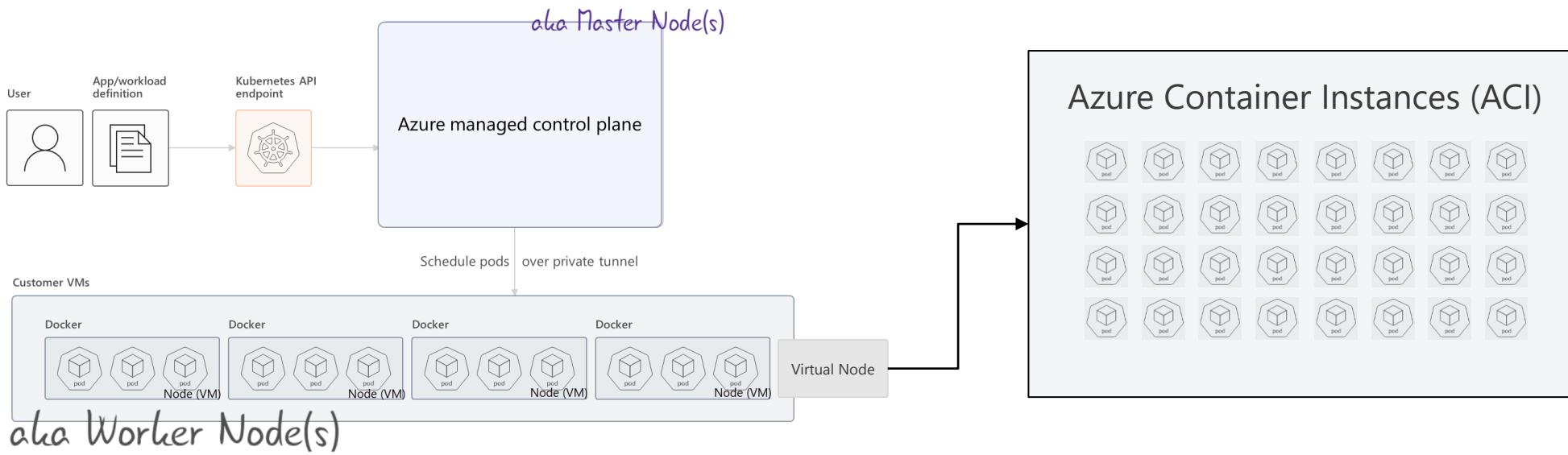
Scaling apps in AKS

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



Create Kubernetes cluster

Basics **Scale** Authentication Networking Monitoring T

Enable scaling features to allow flexible capacity and burstable scaling opti

- **Virtual nodes** allow burstable scaling backed by serverless Azure C
- **VM scale sets** are required for a variety of scenarios including auto [VM scale sets in AKS](#)

Virtual nodes ⓘ

Disabled **Enabled**

VM scale sets ⓘ

Disabled **Enabled**

Enable Virtual Nodes on existing cluster

Azure CLI

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

Console

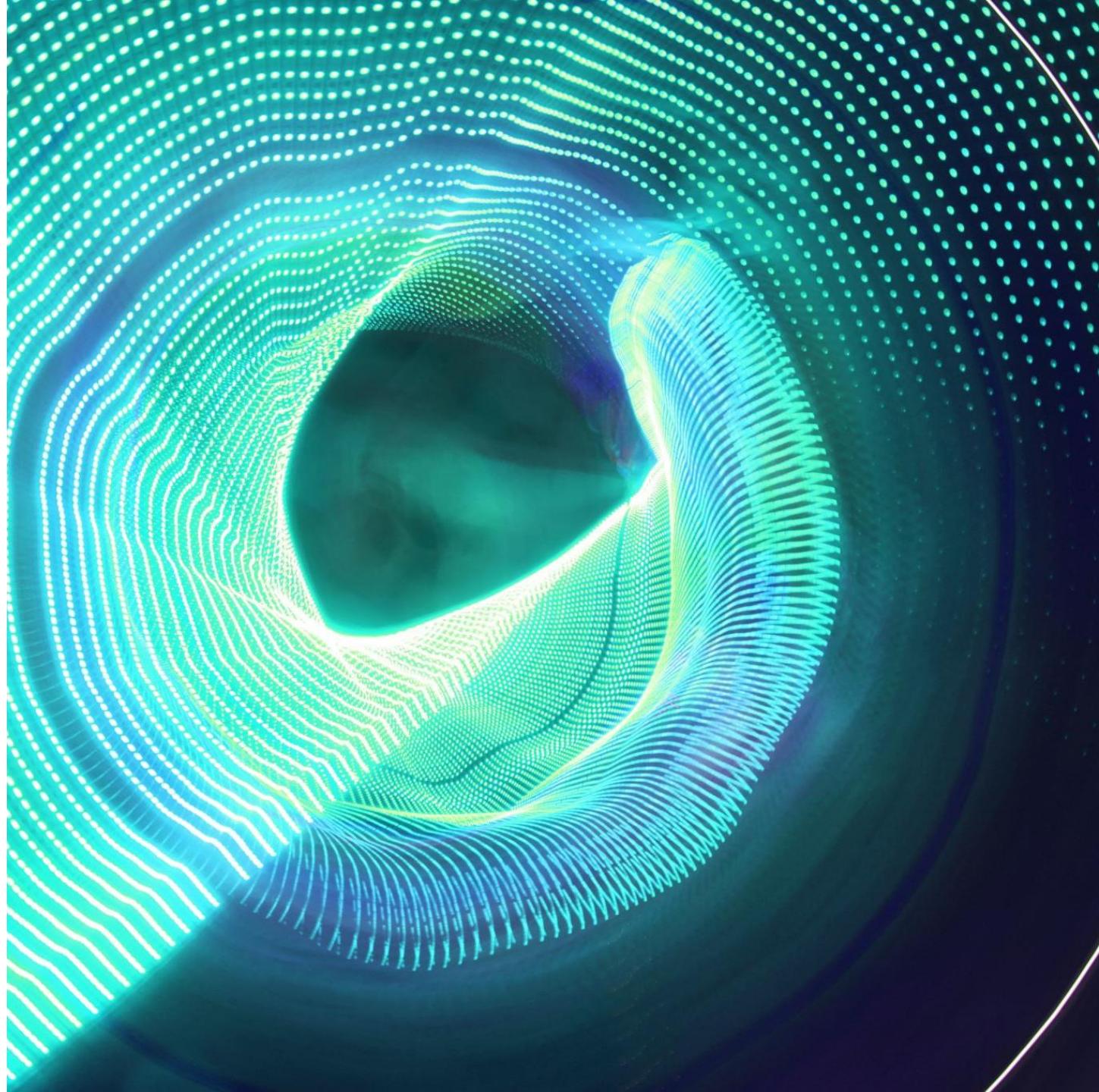
```
kubectl get nodes
```

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

welyaks2 | Workloads (preview)

Kubernetes service | Directory: Microsoft

Search (Ctrl+ /) < + Add Delete Refresh Show labels

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Security

Kubernetes resources

- Namespaces (preview)
- Workloads (preview)**
- Services and ingresses (preview)
- Storage (preview)
- Configuration (preview)

Settings

- Node pools
- Configuration
- Scale

Deployments Pods Replica sets Stateful sets Daemon sets Jobs Cron jobs

Filter by deployment name Filter by label selector ⓘ Filter by name

Name	Namespace
coredns	kube-system
coredns-autoscaler	kube-system
kubernetes-dashboard	kube-system
metrics-server	kube-system
omsagent-rs	kube-system
nginxdeploy	default
bikesharing-traefik	bikeapp
bikes	bikeapp
bikesharingweb	bikeapp
billing	bikeapp

azure-vote-front | YAML

Service

Search (Ctrl+ /) < Refresh

Overview YAML Events

YAML JSON

```
1 kind: Service
2 apiVersion: v1
3 metadata:
4   name: azure-vote-front
5   namespace: default
6   selfLink: /api/v1/namespaces/default/services/azure-vote-front
7   uid:
8   resourceVersion: '857494'
9   creationTimestamp: '2020-08-04T21:27:12Z'
10  finalizers:
11    - service.kubernetes.io/load-balancer-cleanup
12  managedFields:
13    - manager: Mozilla
14      operation: Update
15      apiVersion: v1
16      time: '2020-08-04T21:27:12Z'
17      fieldsType: FieldsV1
18      fieldsV1:
19        'f:spec':
20          'f:externalTrafficPolicy': {}
21          'f:ports':
22            .: {}
23            'k:{"port":80,"protocol":"TCP"}':
24              .: {}
25              'f:port': {}
26              'f:protocol': {}
```

Review + save Discard

Deployment Center in AKS

Deployment Center

Deployment center in Azure DevOps simplifies setting up a robust DevOps pipeline for your application. By default, this configures a DevOps pipeline to deploy your application updates to this Kubernetes cluster. You can extend the default configured DevOps pipeline and add richer DevOps capabilities - approvals before deploying, provisioning additional Azure resources, running scripts, upgrading your application, or even running

Source Repository Application Resources

Select the code location

Azure Repos GitHub

ed free private repos

Almost there!

Azure DevOps project

* Organization name: v-omkar

* Project name: Testing

Container Registry

* Container registry name:

Create new Use existing

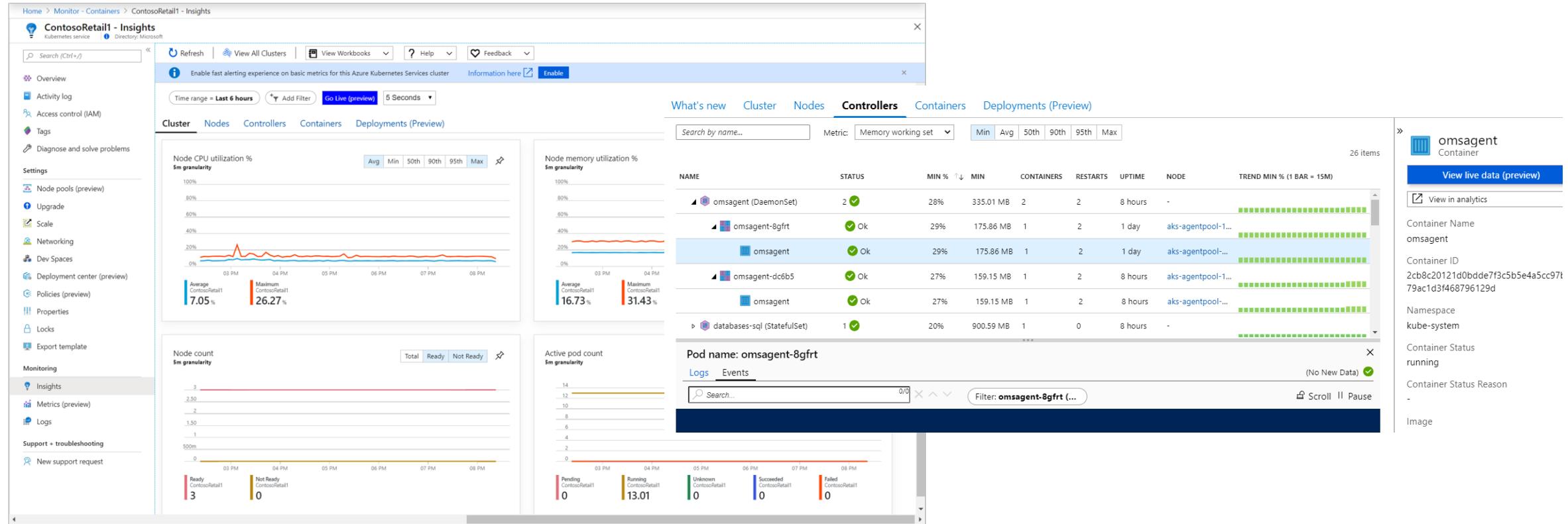
aksactionacr

Previous Next

Previous Finish

Azure Monitor for Containers

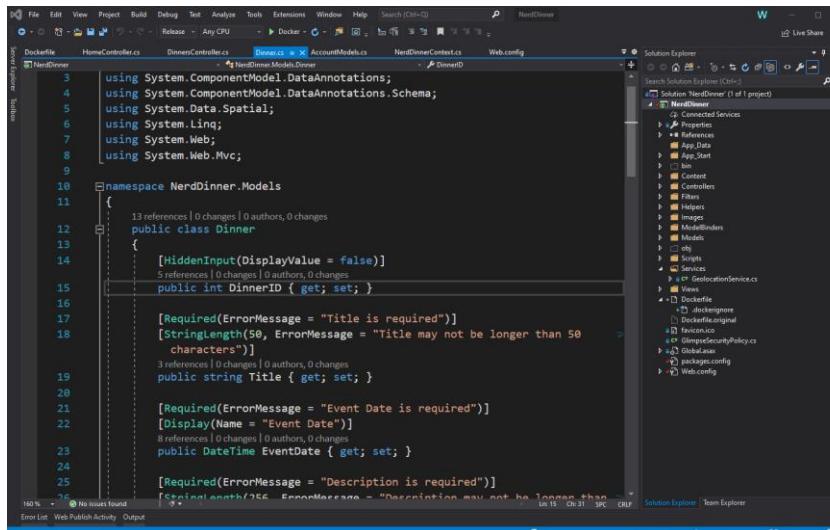
- Infrastructure and Cluster Level Logs / Metrics



- 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



A screenshot of the Visual Studio IDE showing the code for a legacy ASP.NET application. The code is in C# and defines a class named Dinner with properties like DinnerID, Title, EventDate, and Description. The code uses attributes from System.ComponentModel.DataAnnotations and System.Web.Mvc.

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using System.Data.Linq;
using System.Web;
using System.Web.Mvc;

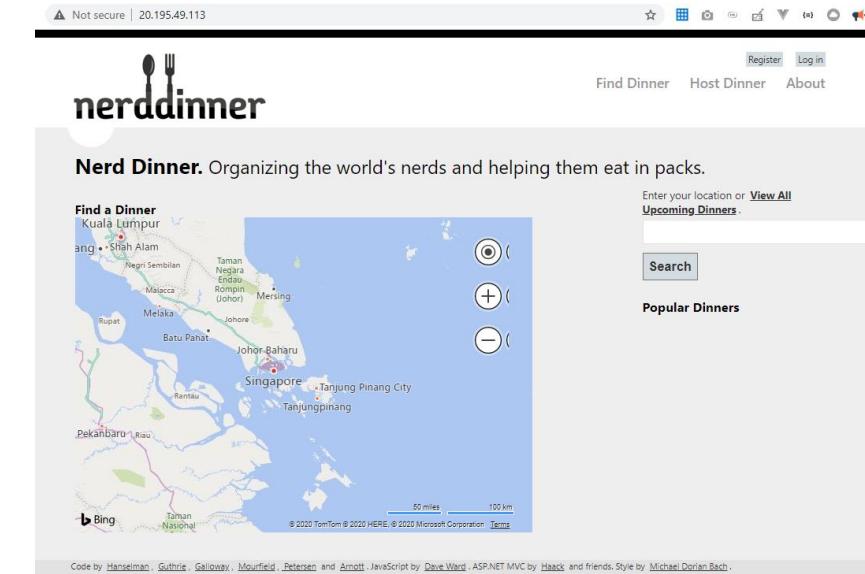
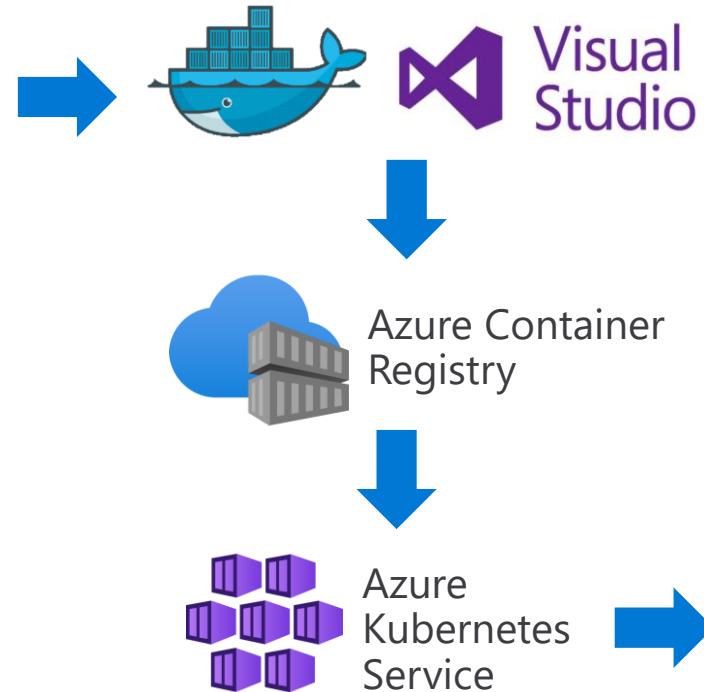
namespace NerdDinner.Models
{
    public class Dinner
    {
        [HiddenInput(DisplayValue = false)]
        public int DinnerID { get; set; }

        [Required(ErrorMessage = "Title is required")]
        [StringLength(50, ErrorMessage = "title may not be longer than 50 characters")]
        public string Title { get; set; }

        [Required(ErrorMessage = "Event Date is required")]
        [Display(Name = "Event Date")]
        public DateTime EventDate { get; set; }

        [Required(ErrorMessage = "Description is required")]
        [StringLength(1000, ErrorMessage = "Description must not be longer than 1000 characters")]
        public string Description { get; set; }
    }
}
```

Legacy ASP.NET Apps



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

github.com/Azure/AKS/projects/1

Why GitHub? Team Enterprise Explore Marketplace Pricing

Azure / AKS

Code Issues 222 Pull requests Actions Projects 2 Security Insights

Azure Kubernetes Service Roadmap (Public)
Updated 4 days ago

Backlog (21) Planned (Committed) (11) In Progress (Development) (18) Public Preview (Shipped & Improving) (24) Generally Available (Done) (28) Archive (GA older than 6 months) (7)

Planned (Committed): AKS in VNET behind company HTTP proxy, [Feature] Automatic Node Image Upgrade for node versions, AKS allows creation of NodePools in different Subnets (Kubenet), [Feature Request] AKS API to enable retrieval of supported SKUs on AKS, Enable AKS to use ACR with an image signing solution (ex: Notaryv2, Content Trust), Add Support for a KMS provider for Encrypting Secrets, AKS support of Azure NAT Gateway as outboundType option, Standard SKU Loadbalancer does not use own static public ip as outbound IP.

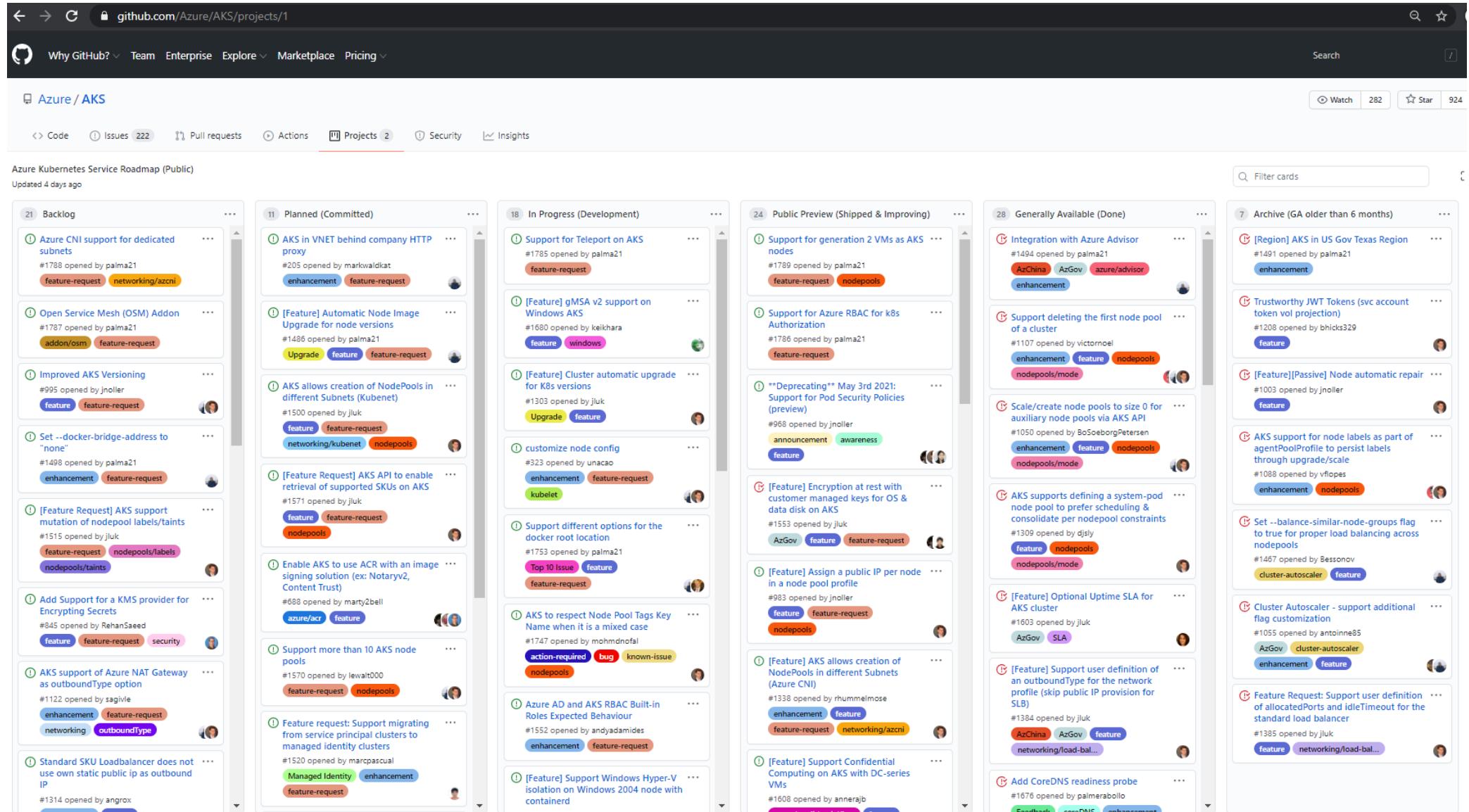
In Progress (Development): Support for Teleport on AKS, [Feature] gMSA v2 support on Windows AKS, [Feature] Cluster automatic upgrade for K8s versions, customize node config, Support different options for the docker root location, AKS to respect Node Pool Tags Key Name when it's a mixed case, Support more than 10 AKS node pools, Feature request: Support migrating from service principal clusters to managed identity clusters.

Public Preview (Shipped & Improving): Support for generation 2 VMs as AKS nodes, Support for Azure RBAC for k8s Authorization, **Deprecating** May 3rd 2021: Support for Pod Security Policies (preview), [Feature] Encryption at rest with customer managed keys for OS & data disk on AKS, Assign a public IP per node in a node pool profile, [Feature] Optional Uptime SLA for AKS cluster, [Feature] AKS allows creation of NodePools in different Subnets (Azure CNI), [Feature] Support user definition of an outboundType for the network profile (skip public IP provision for SLB), [Feature] Support Confidential Computing on AKS with DC-series VMs, Add CoreDNS readiness probe.

Generally Available (Done): Integration with Azure Advisor, Support deleting the first node pool of a cluster, Scale/create node pools to size 0 for auxiliary node pools via AKS API, AKS supports defining a system-pod node pool to prefer scheduling & consolidate per nodepool constraints, AKS supports defining a system-pod node pool to prefer scheduling & consolidate per nodepool constraints, [Feature] Assign a public IP per node in a node pool profile, [Feature] Optional Uptime SLA for AKS cluster, [Feature] Support user definition of an outboundType for the network profile (skip public IP provision for SLB), Feature Request: Support user definition of allocatedPorts and idleTimeout for the standard load balancer, Add CoreDNS readiness probe.

Archive (GA older than 6 months): [Region] AKS in US Gov Texas Region, Trustworthy JWT Tokens (svc account token vol projection), [Feature][Passive] Node automatic repair, AKS support for node labels as part of agentPoolProfile to persist labels through upgrade/scale, Set --balance-similar-node-groups flag to true for proper load balancing across nodepools, Cluster Autoscaler - support additional flag customization.

Filter cards



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>

