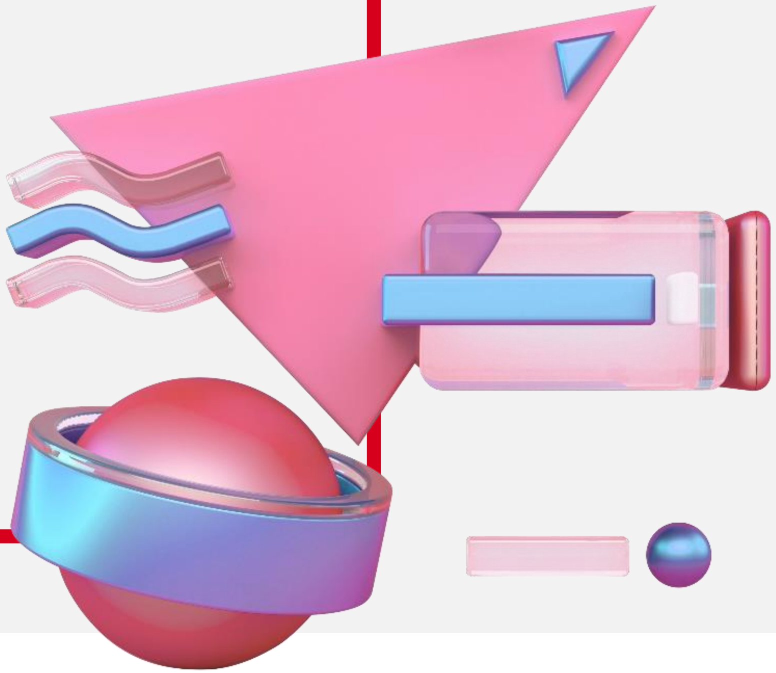
Azure Kubernetes Service (AKS) and Azure Container Registry (ACR)



Tran Ngoc Son

Cao Thi Hoang Le

July - 2023

Nash
Tech.

Agenda

Azure Kubernetes Service (AKS)

- Introduction to AKS
- AKS Architecture
- AKS Features
- AKS Networking
- AKS Storage
- AWS Security
- AKS Scaling

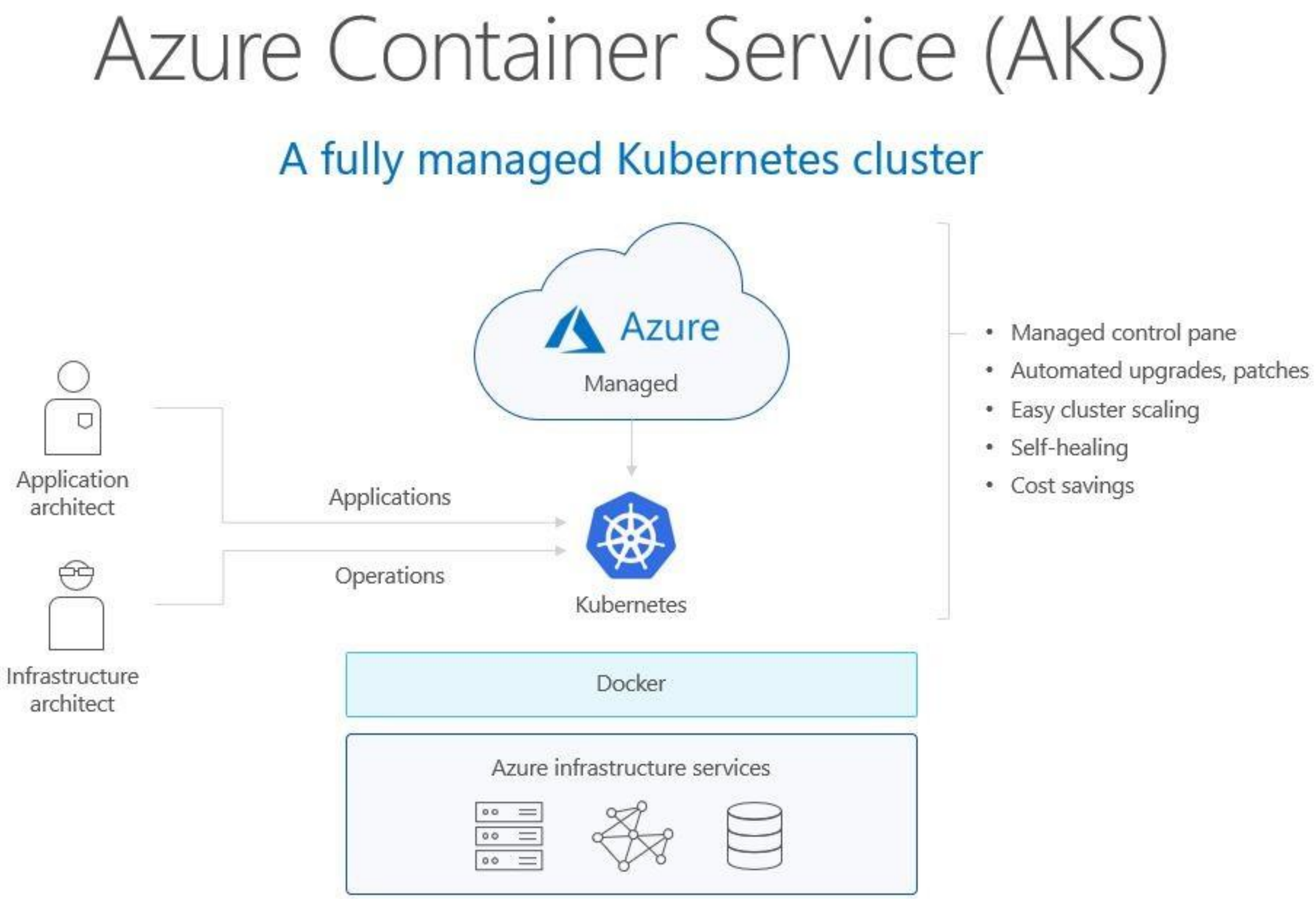
ACR

- What is ACR?
- Security

Demo

Azure Kubernetes Service (AKS)

Introduction to AKS



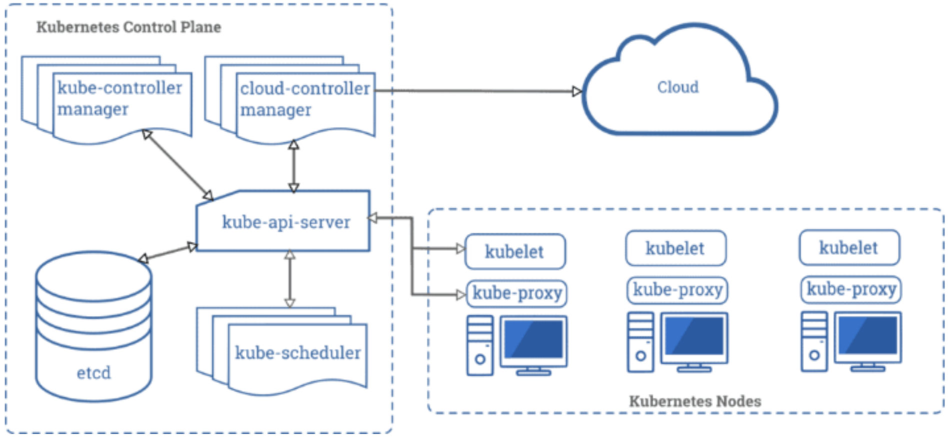
Azure Container Service (AKS)

A fully managed Kubernetes cluster

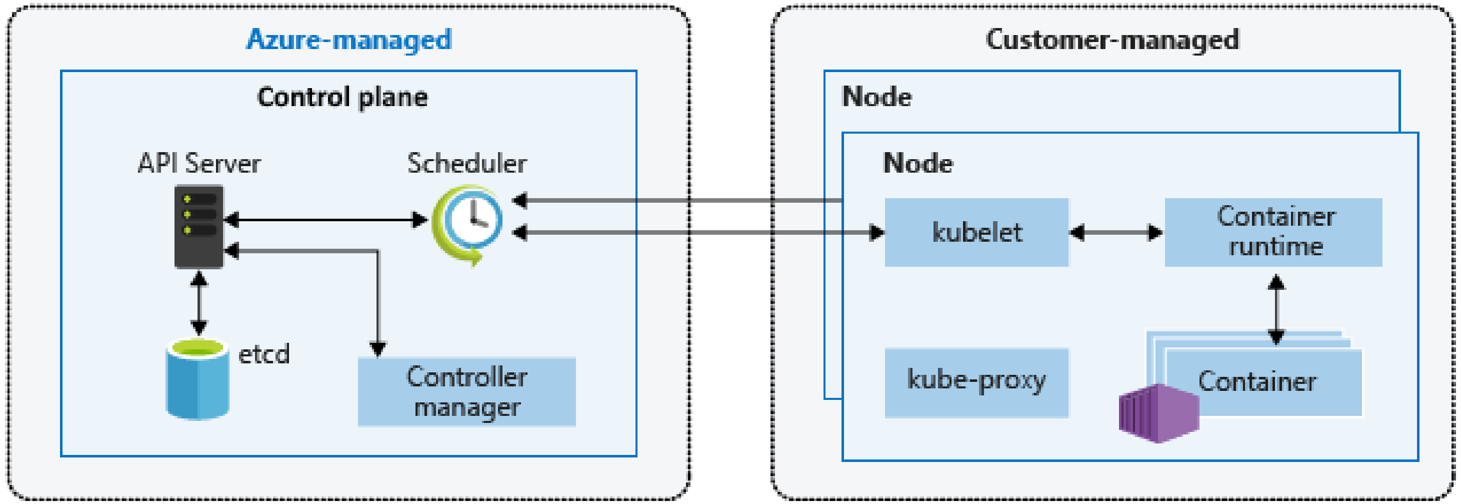
- Managed control pane
- Automated upgrades, patches
- Easy cluster scaling
- Self-healing
- Cost savings

AKS Architecture

Kubernetes Architecture Components



AKS Architecture



AKS Features

Control Plane

- Installation and management for free

Upgrades

- Kubernetes version upgrades

Patching

- Automated security patching

Self Healing

- Control plane auto health management

Networking

- Basic and Advanced networking options

Scaling

- Automated node scaling

Encryption

- TLS encrypted connection everywhere

AD RBAC

- Installation and management for free

AKS Networking (1/2)

AKS kubenet (Basic Networking)

By default, Azure Kubernetes Service (AKS) clusters use kubenet, and this will create an Azure virtual network and subnet for you. Using kubenet, only the nodes receive an IP address in the virtual network subnet and pods can't communicate directly with each other. Instead, User Defined Routing (UDR) and IP forwarding are used for connectivity between pods across nodes. In basic networking pod, IP natted inside subnet. Network address translation (NAT) is used so that the pods can reach resources on the Azure virtual network

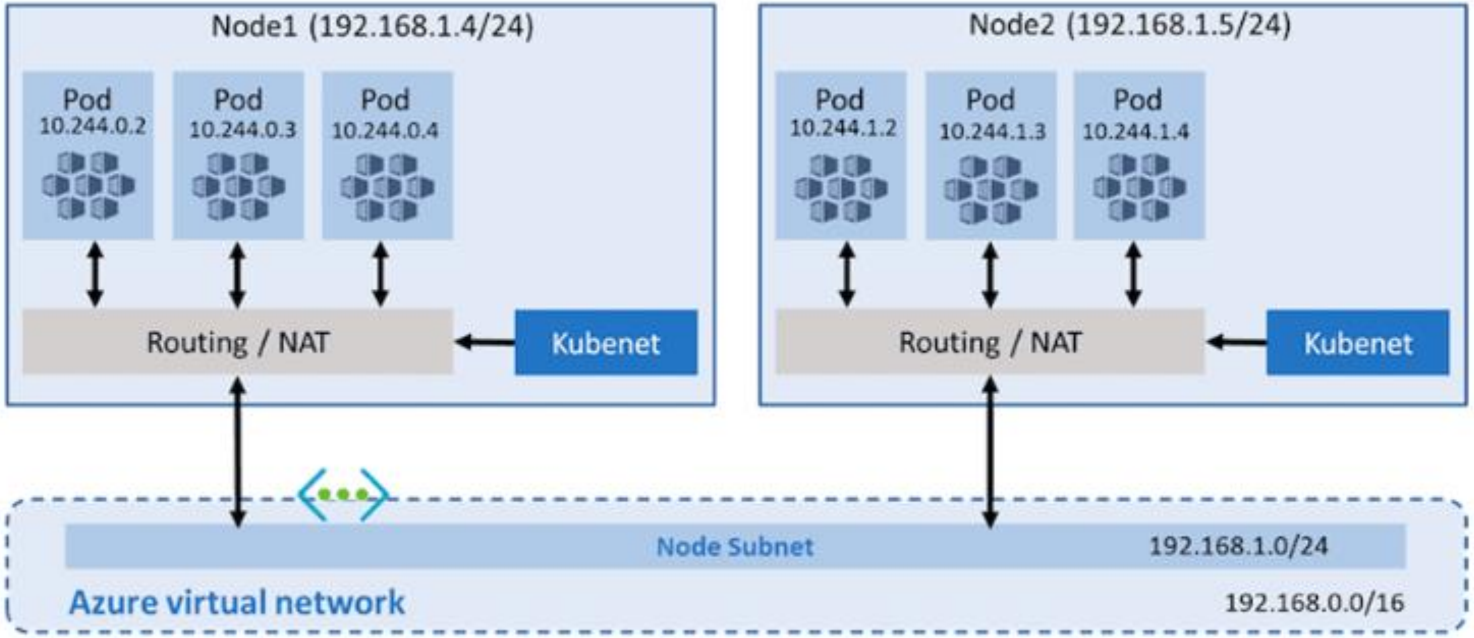


Image Courtesy: Microsoft

AKS Networking (2/2)

AKS CNI (Advanced Networking)

In Azure Container Networking Interface (CNI) every pod gets an IP address from the subnet and can be accessed directly via their private IP address from connected networks. These IP addresses must be unique across your network space. These IP's must be planned in advance. Advance networking requires more planning if all IP addresses used then we need to rebuild clusters in a larger subnet as your application demands

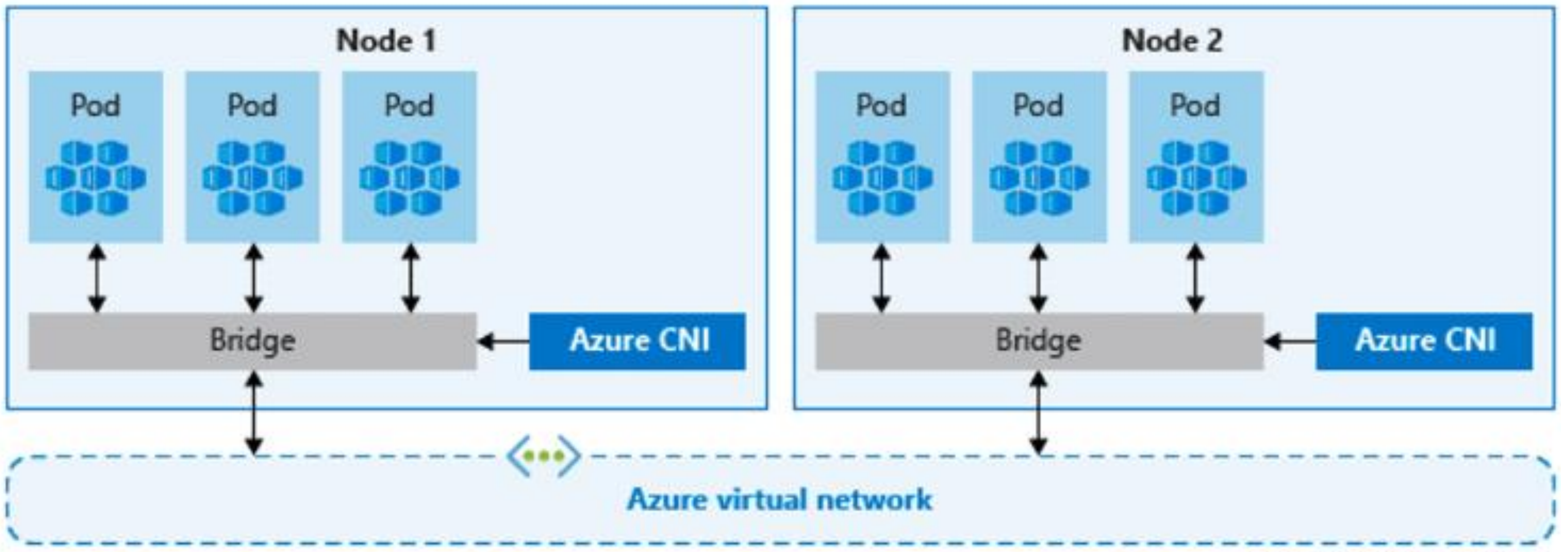


Image Courtesy: Microsoft

AKS Storage

- In AKS, There are two types of storage options available:

- Azure Disk

- These stores are used to provide persistent storage to store data and images persistently. Azure disks can be used to create a Kubernetes DataDisk resource and mounted as ReadWrite, so the disks are only available to a single pod. we can't share this with multiple pods

- Azure Files

Azure Files are SMB based shared file system mounted across different machines. Using Azure files, we can share data across multiple nodes and pods.

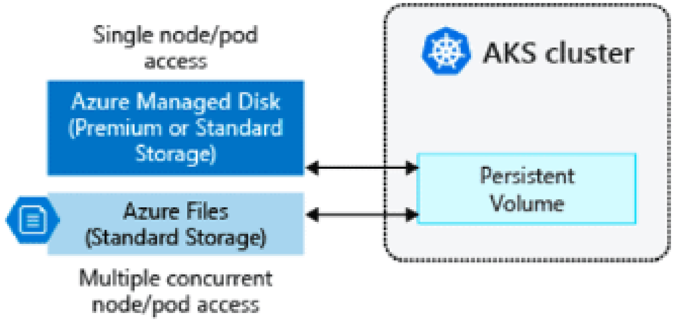
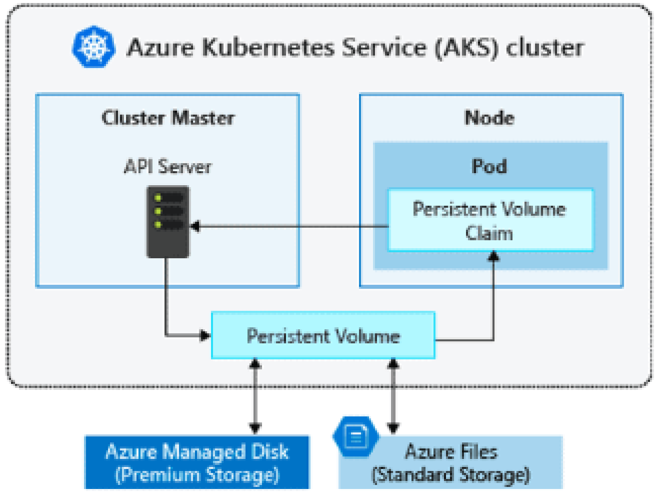

Single node/pod access
Azure Managed Disk (Premium or Standard Storage)
Azure Files (Standard Storage)
Multiple concurrent node/pod access
AKS cluster
Persistent Volume

Image Courtesy: Microsoft


Azure Kubernetes Service (AKS) cluster
Cluster Master
API Server
Node
Pod
Persistent Volume Claim
Persistent Volume
Azure Managed Disk (Premium Storage)
Azure Files (Standard Storage)

AKS Security

- Azure Active Directory with AKS, We can integrate Azure Kubernetes with Azure Active Directory so the users in Azure Active Directory for user authentication.
- Using this user in Azure Active Directory can access the AKS cluster using an Azure AD authentication token. we can also configure Kubernetes role-based access control (RBAC) based on a user’s identity. Azure AD legacy integration can only be enabled during cluster creation.

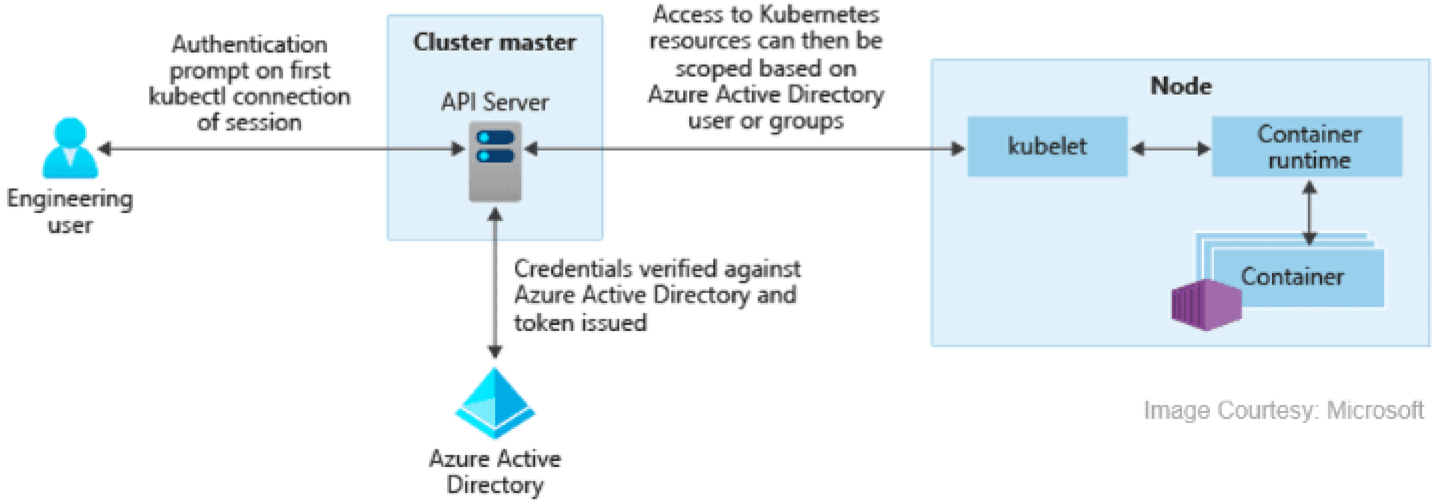

Authentication prompt on first kubectl connection of session
Cluster master
API Server
Access to Kubernetes resources can then be scoped based on Azure Active Directory user or groups
Node
kubelet
Container runtime
Container
Engineering user
Credentials verified against Azure Active Directory and token issued
Azure Active Directory

Image Courtesy: Microsoft

AKS Scaling (1/3)

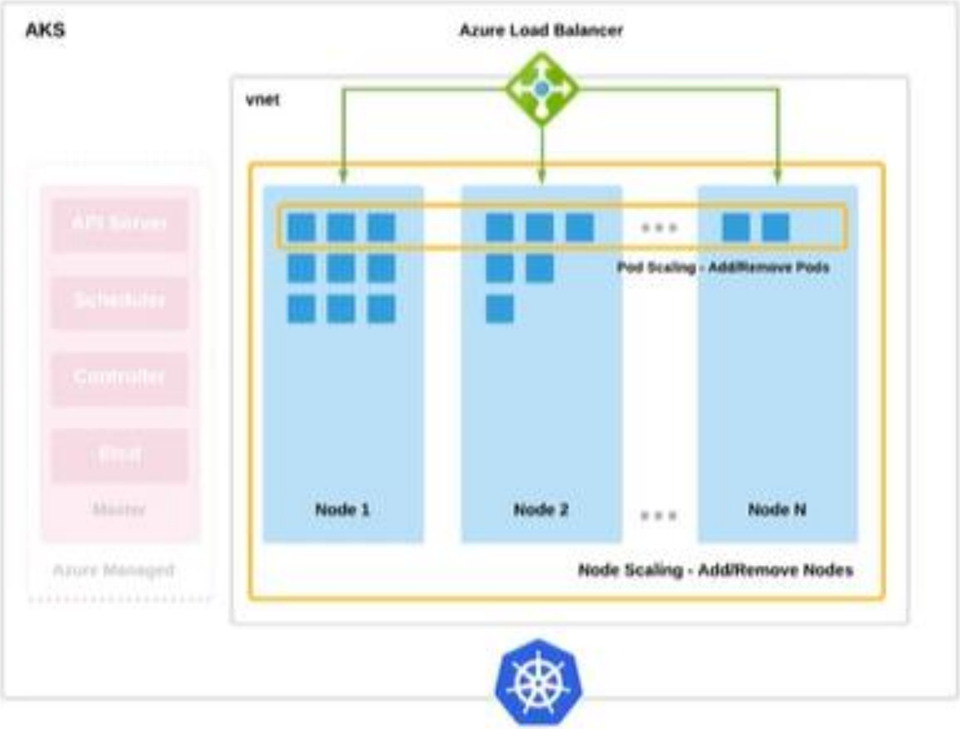
Manual Scaling

- Manual scaling can be performed either at the node level, or at the pod level by simply increasing or decreasing the respective count property
- For example, the following Azure CLI command is used to scale an existing nodepool up to 10 nodes.

```
 az aks scale \
      --name democluster \
      --resource-group practical-devops\
      --node-count 10 \
      --node-pool-name workerpool11
```

Or the kubectl command is used to manually scale up the webapp deployment to have 10 pods.

```
kubectl scale deployment webapp \
      --name democluster \
      --repliccas=10
```

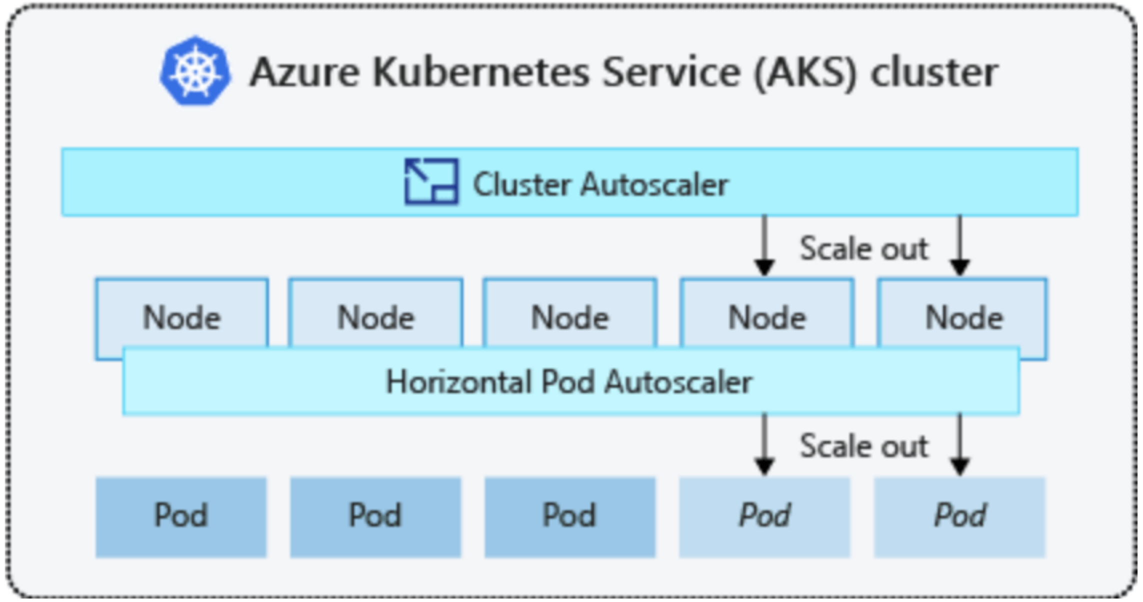


AKS Scaling (2/3)

Auto scaling

- Auto scaling can be performed within an AKS cluster automatically by two different scalers.
  - The **cluster autoscaler** can be configured to automatically add or remove worker node vm's to and from the cluster. This helps to ensure that your team does not have to manually intervene when there are changes in workloads
    - For example, here we are using CLI to create the cluster. We need to use the flag –enable-cluster-autoscaler

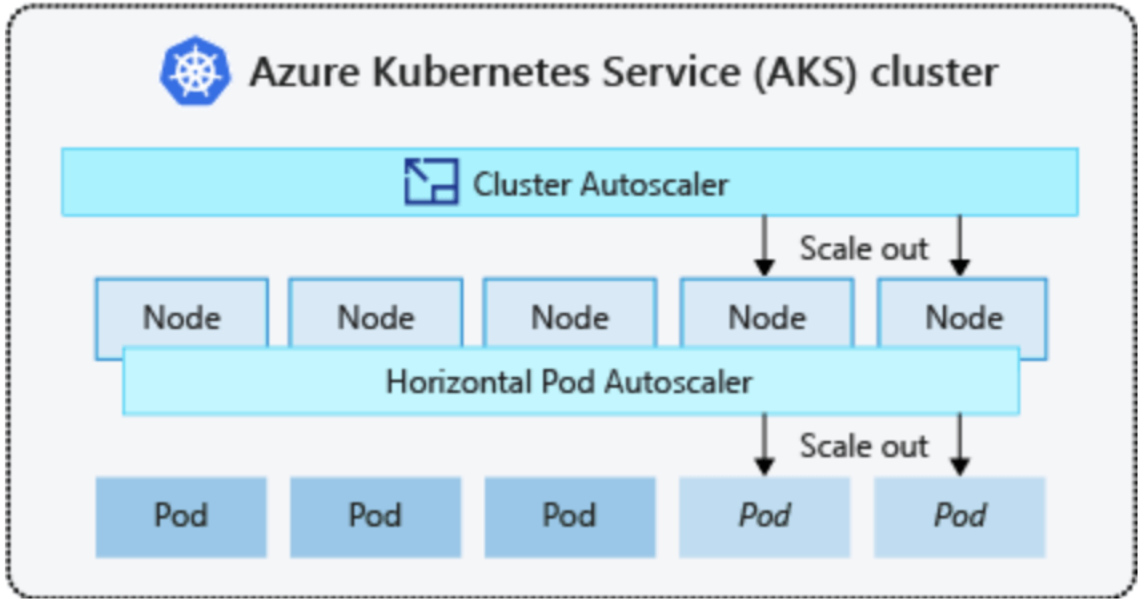
```
az aks create \
  --resource-group myResourceGroup \
  --name myAKSCluster \
  --node-count 1 \
  --enable-cluster-autoscaler \
  --min-count 1 \
  --max-count 3 \
  --cluster-autoscaler-profile scan-interval=30s
```



AKS Scaling (3/3)

Auto scaling

- Kubernetes allows for **horizontal pod autoscaling**, which can adjust the number of pods in a deployment based on CPU utilization and other select metrics. The Metrics Server provides resource usage data to Kubernetes and is automatically included in version 1.10 and higher AKS clusters
- We can implement the Autoscaler:
  - Create a manifest file to define Autoscaler behavior.
  - Enable horizontal pod autoscaling via kubectl
    - kubectl autoscale deployment myapp --cpu-percent=50 --min=3 --max=10

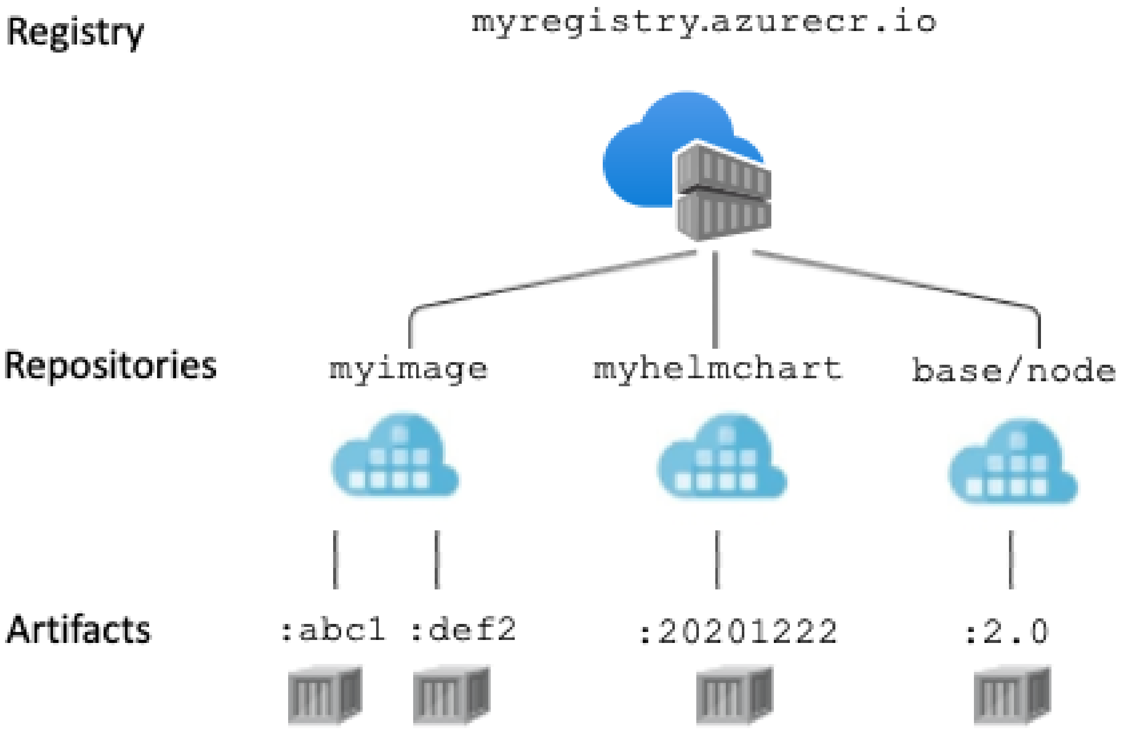


02

Azure Container Registry (ACR)

What is ACR?

- Azure Container Registry permits you to create, store, and manage container images in a registry for all types of container deployments.
- ACR enables you to create private registries, which are only accessible by you and your team members, or public registries, which can be accessed by anyone with the registry's name and a valid subscription ID.



Azure Container Registry Features

Registry Service Tiers

- The Basic tier provides an image repository with limited storage capacity and retention time.
- The Standard tier provides an image repository with unlimited storage capacity and retention time

Security and Access

- Access a registry with the Azure CLI or with the usual docker login command. Using TLS encryption, Azure Container Registry encrypts connections to clients and sends container images over HTTPS.

Versioned Storage

- Azure Container Registry stores your container images as a collection of layers that can be versioned independently. This allows you to control access by tagging layers with access control lists (ACLs) or to use permissions on specific tags.

Automated Image Builds

- Azure Container Registry provides Automated Image Builds feature that allows you to build container images from source code on a schedule. The built images are stored in the same registries as they were built. This feature helps you to avoid manual steps of pushing images to the registry and enables you to have a single source of truth for your container images

03

Demo

AKS Demo

- Refer to AKS_HandsOn

ACR Demo

- Refer to ACR_HandsOn

Thank you

Nash
Tech.