

# Introduction to Azure Kubernetes Service

Championing Azure – Cloud Native

**Jakob Ehn**

@jakobehn

<https://blog.ehn.nu>

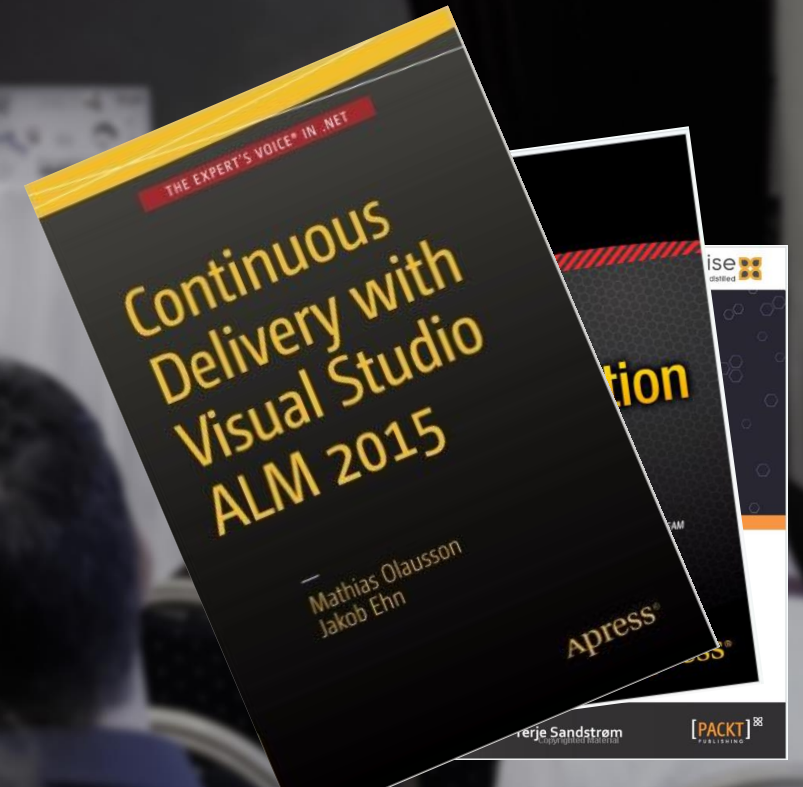




Jakob Ehn  
Microsoft Azure MVP

<https://blog.ehn.nu>

@jakobehn

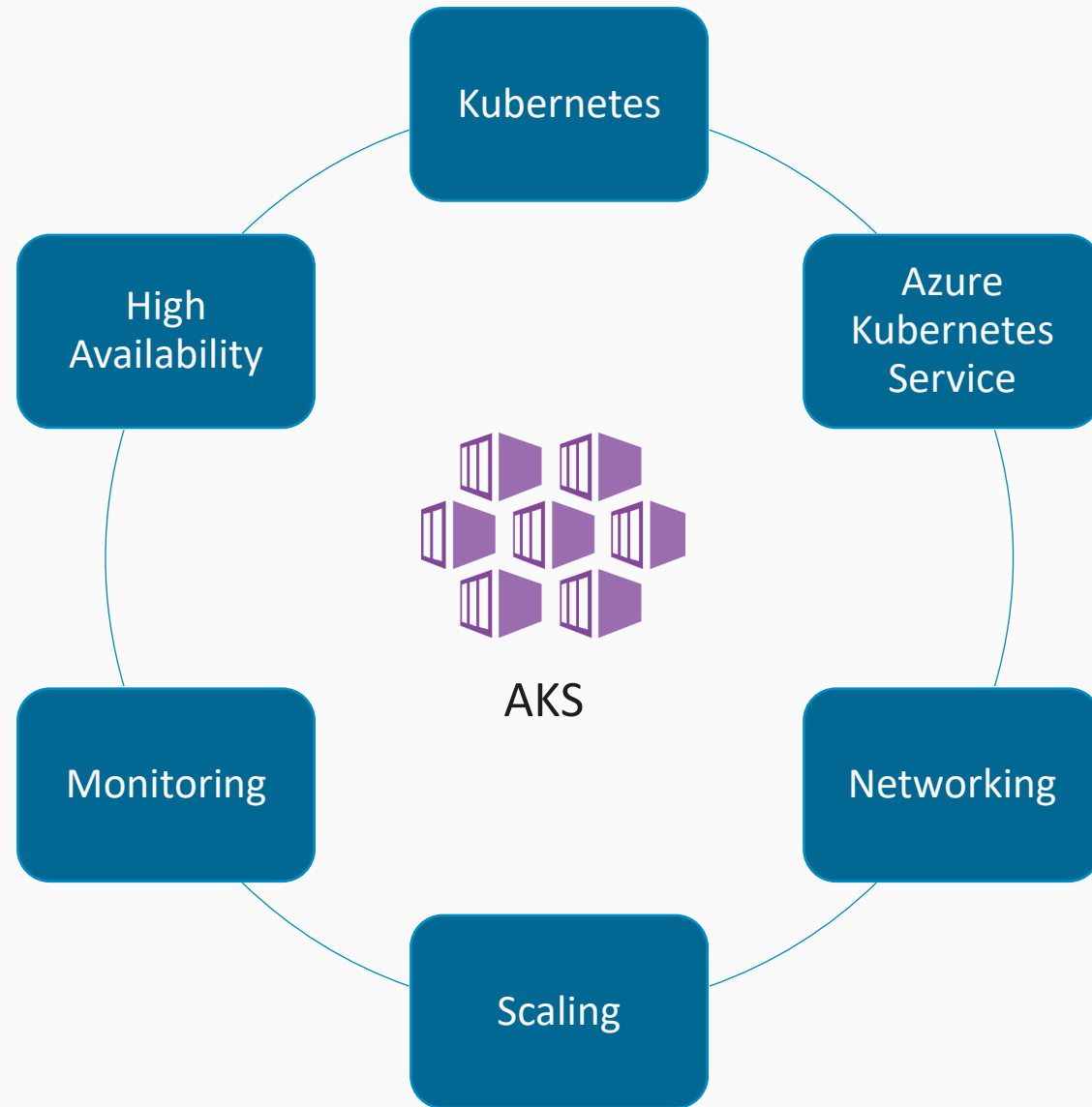




# Software Developers in Sweden!

A large group of approximately 30 people, mostly men, are posing for a group photo on a wooden pier. They are arranged in several rows, some standing and some sitting on the pier's edge. The pier is decorated with a row of white tires along its front edge. In the background, there are traditional Swedish wooden houses with gabled roofs and a large white boat with a cabin and a Swedish flag on its mast. The sky is overcast and the water is calm.

**active**  
SOLUTION



# Options for running Containers in Azure



Azure Container Instances



Azure App Service



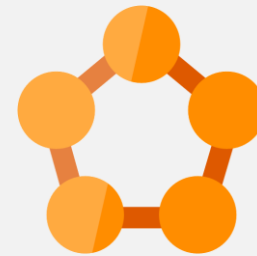
Azure Container Apps



Azure Kubernetes Service

# Container Orchestration

- Rolling Upgrades
- Health Monitoring & Self-Healing
- Load Balancing & Service Discovery
- Dynamic scaling
- Resource Governance



# Kubernetes (k8s)

- Container cluster management, open sourced by Google
- Portable
- Extensible
- Self healing
- Supported by all cloud vendors (+ on premises)
- Huge community support

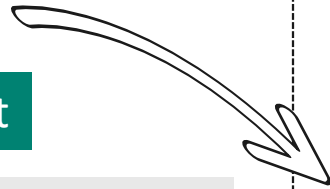


# Kubernetes Overview

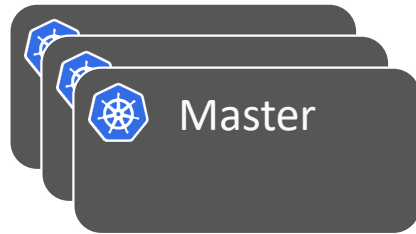


Deployment

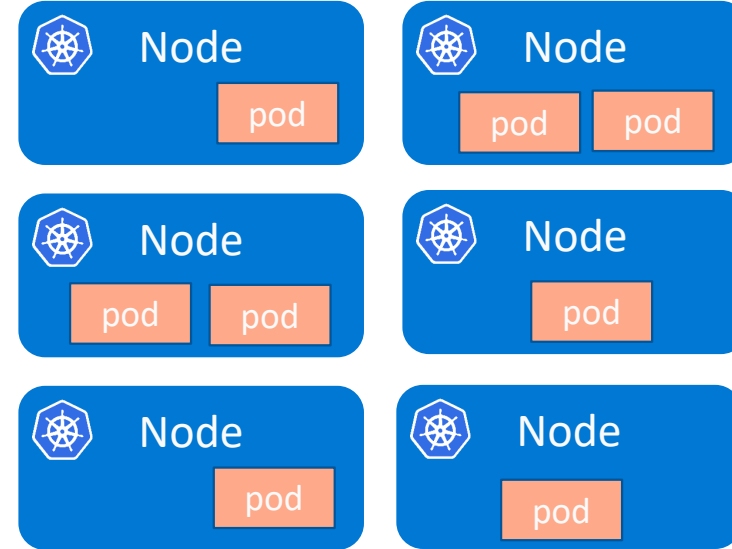
*kubectl apply*



```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: <deployment-name>
spec:
  replicas: 8
  template:
    metadata:
      labels:
        <key>: <value>
    spec:
      containers:
        image: <container-image>
      ports:
        - containerPort: 80
  env:
    - name: <key>
      value: <value>
```



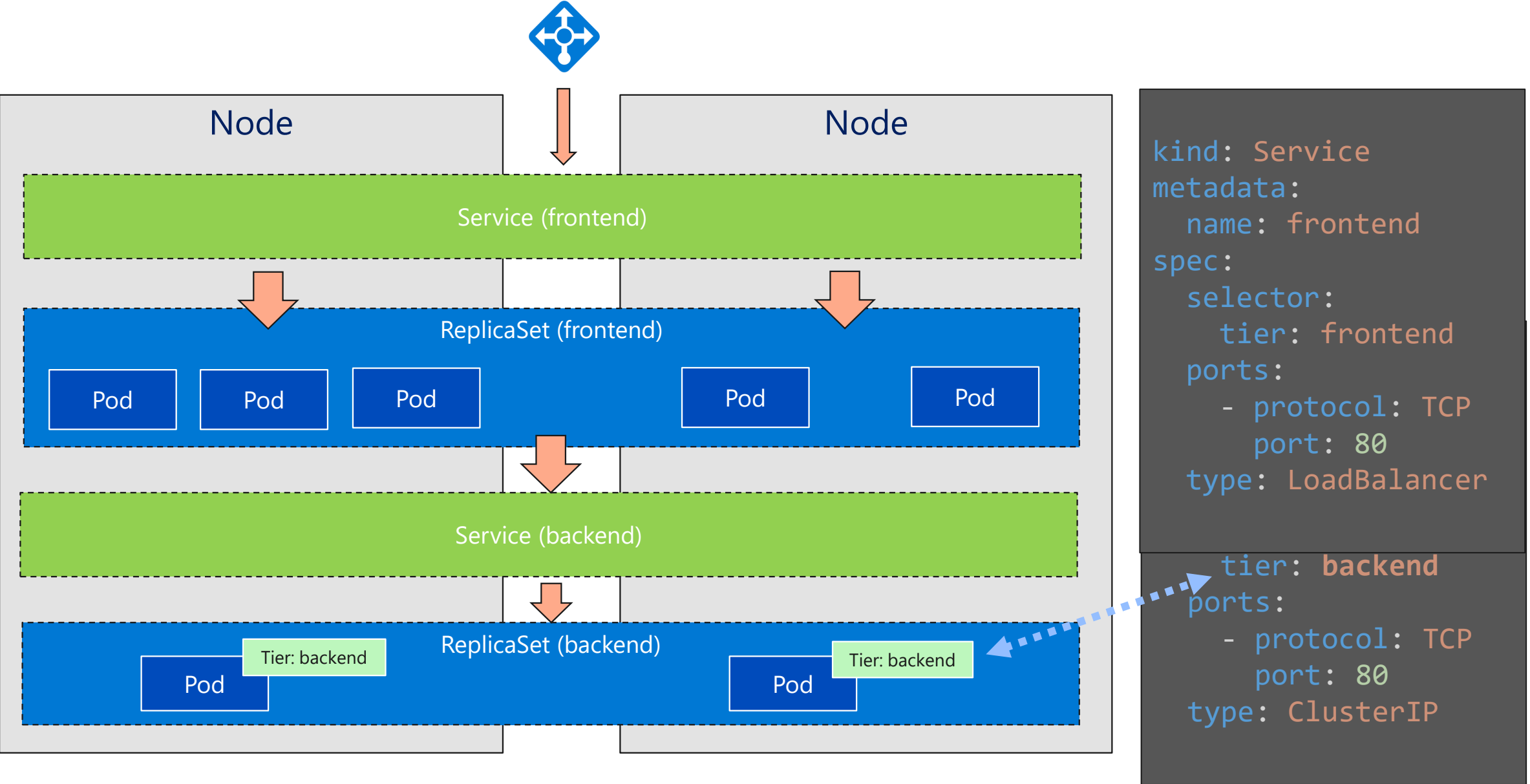
Master

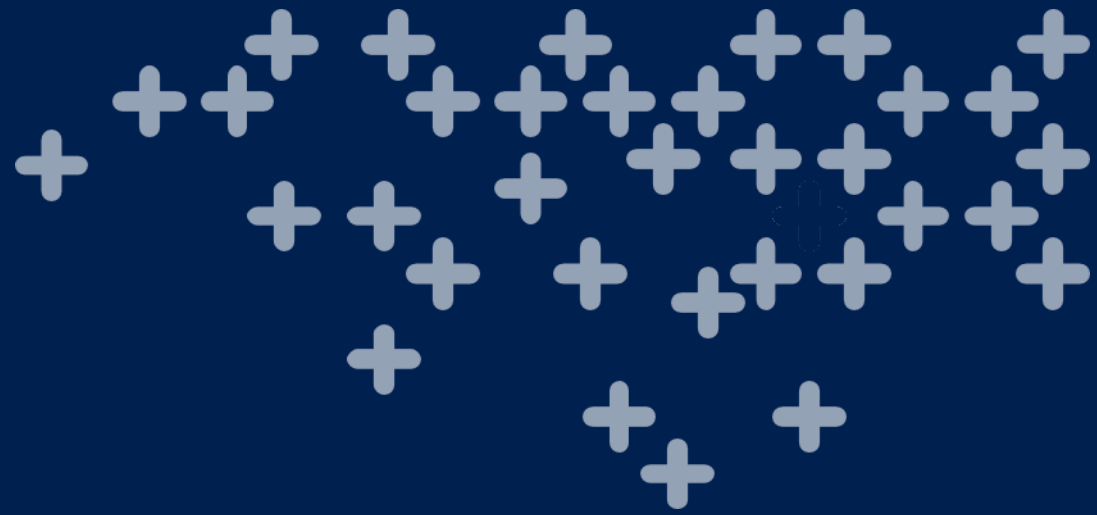


K8s cluster



# Kubernetes Concepts





# Azure Kubernetes Service

# Azure Kubernetes Services (AKS)

- Managed Kubernetes cluster
- 100% upstream (currently supports  $\leq 1.22.6$ )
- GA June 2018
- "one of the fastest growing services in the history of Azure."

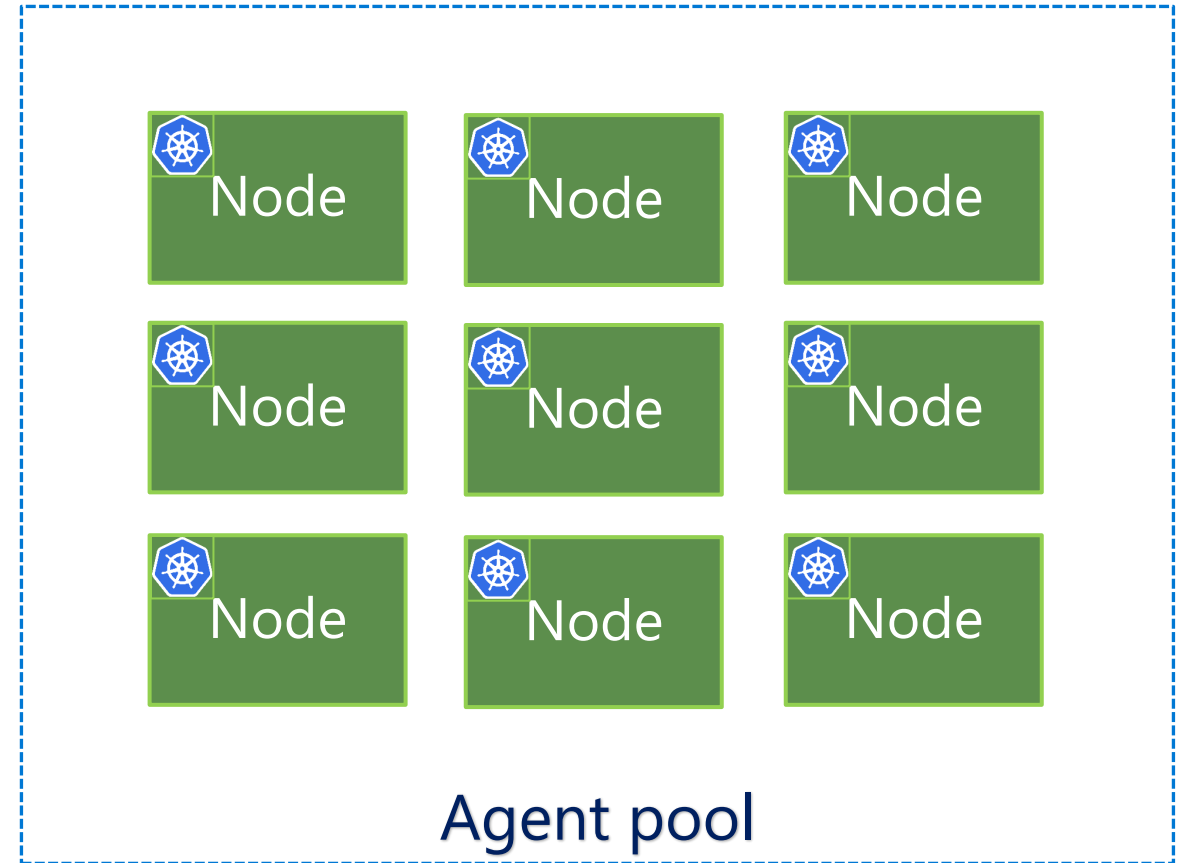


<https://azure.microsoft.com/en-au/blog/bringing-serverless-to-azure-kubernetes-service/>

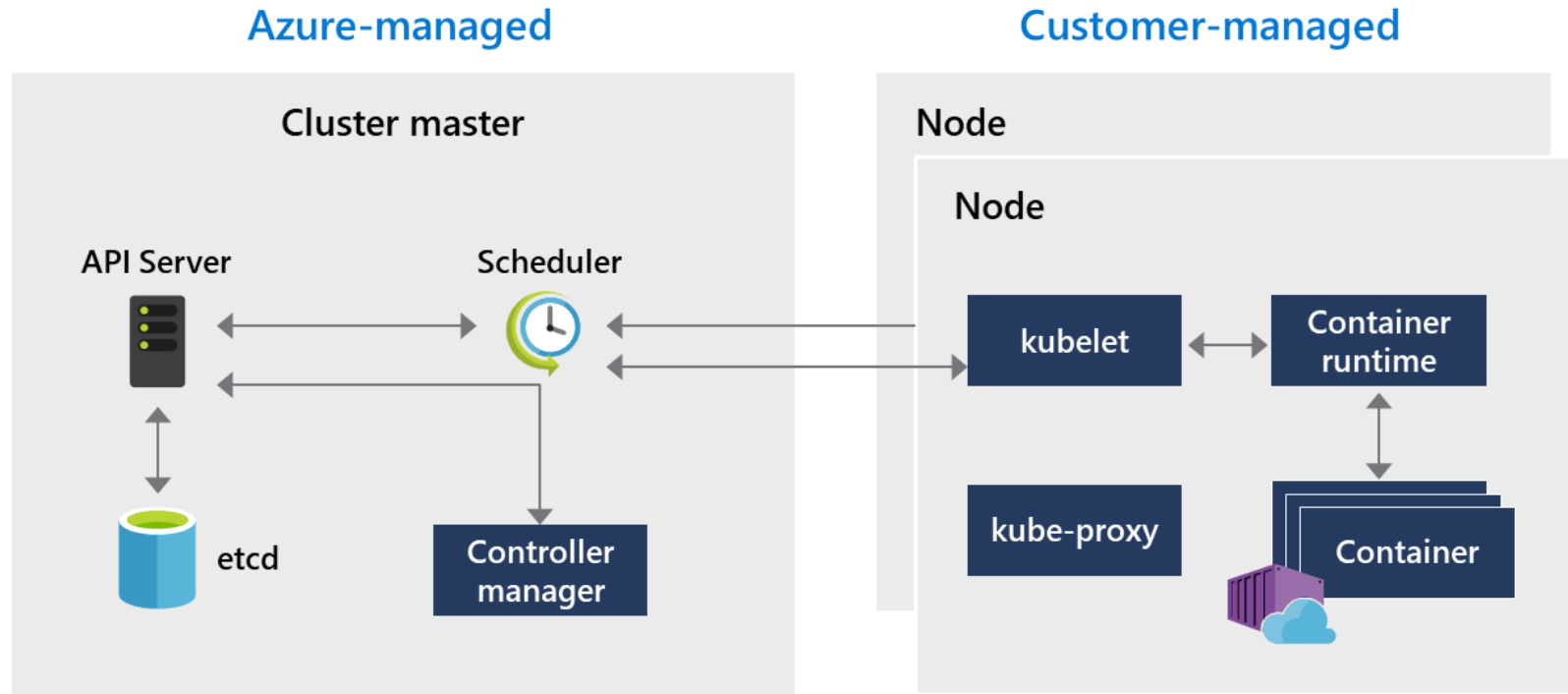
# Azure Kubernetes Services



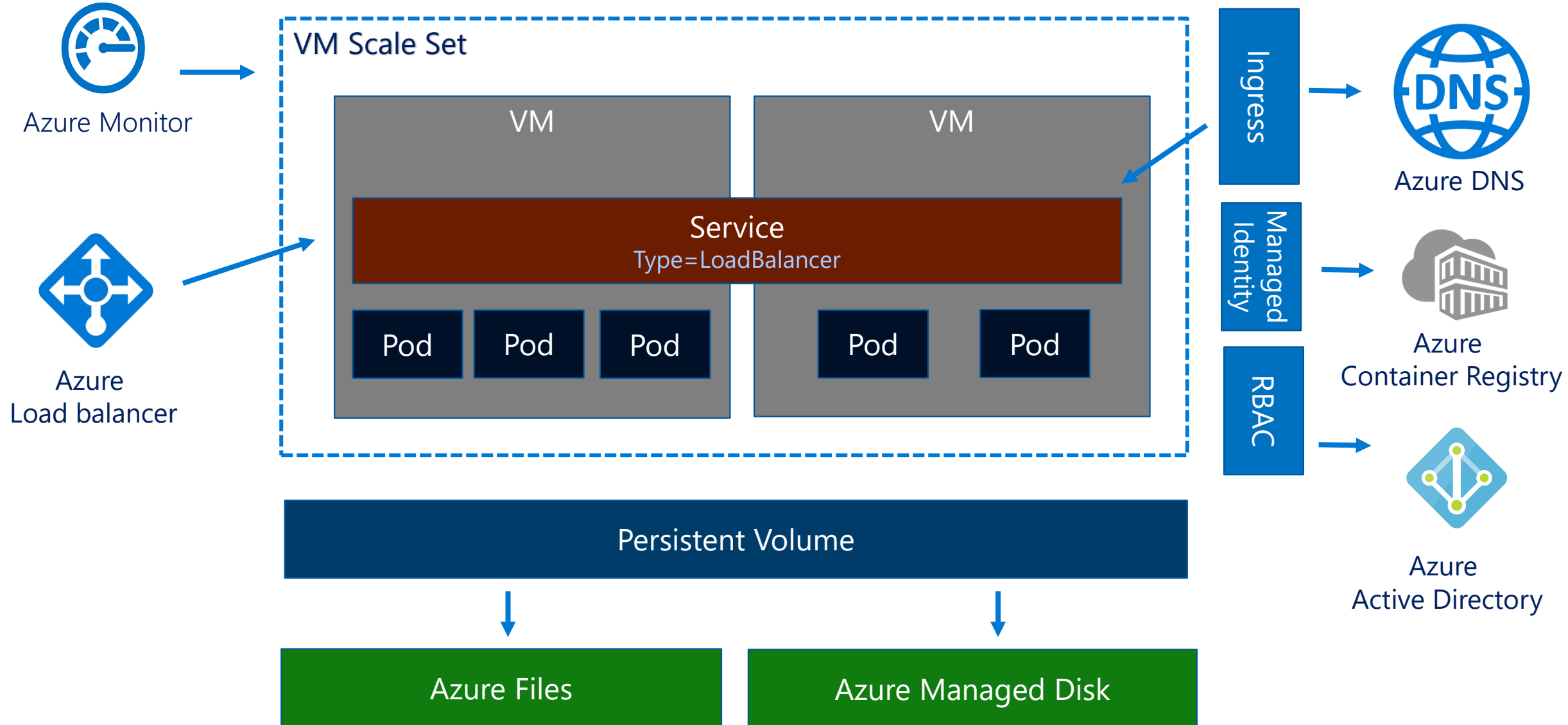
- Automated upgrades and patching
- Cluster (Auto)Scaling
- Self-healing control plane
- Pay for agent nodes only



# AKS Architectural Components



# Kubernetes in Azure





# Creating an AKS Cluster - Portal

## Create Kubernetes cluster

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

\* Subscription ⓘ

Visual Studio Enterprise



\* Resource group ⓘ

aks



[Create new](#)

### CLUSTER DETAILS

\* Kubernetes cluster name ⓘ

\* Region ⓘ

North Europe



\* Kubernetes version ⓘ

1.11.3



\* DNS name prefix ⓘ

### SCALE

The number and size of nodes in your cluster. For production workloads, at least 3 nodes are recommended for resiliency. For development or test workloads, only one node is required. You will not be able to change the node size after cluster creation, but you will be able to change the number of nodes in your cluster after creation. [Learn more about scaling in Azure Kubernetes Service](#)

\* Node size ⓘ

**Standard DS2 v2**

2 vcpus, 7 GB memory

[Change size](#)

\* Node count ⓘ



3

[Review + create](#)

[Previous](#)

[Next : Authentication >](#)

# Creating an AKS Cluster - CLI

```
//Create AKS cluster
```

```
> az aks create -g myResourceGroup -n myCluster --generate-ssh-keys
```

```
\ Running ..
```

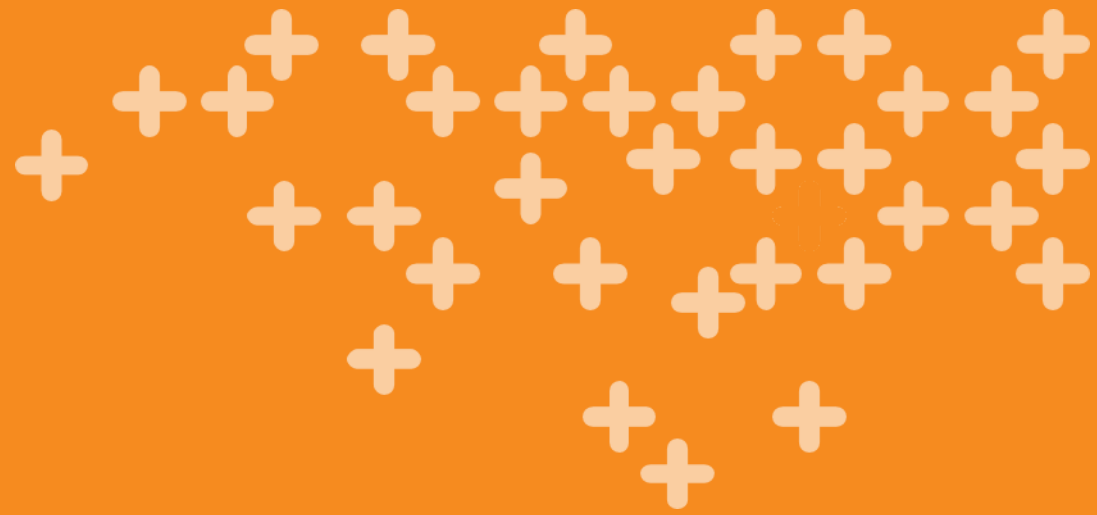
```
//Get access credentials for the cluster
```

```
> az aks get-credentials -g myResourceGroup -n myCluster
```

```
Merged "myCluster" as current context ..
```

# Creating an AKS Cluster – ARM Template

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": { ...
  },
  "resources": [
    {
      "apiVersion": "2018-03-31",
      "type": "Microsoft.ContainerService/managedClusters",
      "location": "[parameters('location')]",
      "name": "[parameters('resourceName')]",
      "properties": {
        "kubernetesVersion": "[parameters('kubernetesVersion')]",
        "dnsPrefix": "[parameters('dnsPrefix')]",
        "agentPoolProfiles": [
          {
            "name": "agentpool",
            "osDiskSizeGB": "[parameters('osDiskSizeGB')]",
            "count": "[parameters('agentCount')]",
            "vmSize": "[parameters('agentVMSize')]",
            "osType": "[parameters('osType')]",
            "storageProfile": "ManagedDisks"
          }
        ],
        "linuxProfile": {
          "adminUsername": "[parameters('linuxAdminUsername')]",
          "ssh": {
            "publicKeys": [
              {
                "keyData": "[parameters('sshRSAPublicKey')]"
              }
            ]
          }
        }
      }
    }
  ]
}
```



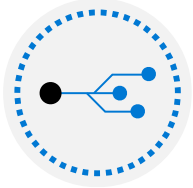
# DEMO

Creating an AKS cluster



# Networking

# Kubernetes Networking



Kubernetes uses Services to logically group a set of pods together to provide network connectivity

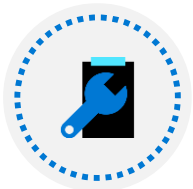
---



## **ClusterIP:**

Creates an internal IP address for use within the AKS cluster

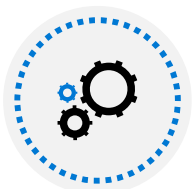
---



## **NodePort:**

Creates a port mapping on the underlying node, so an application can be accessed directly with the node IP address and port

---



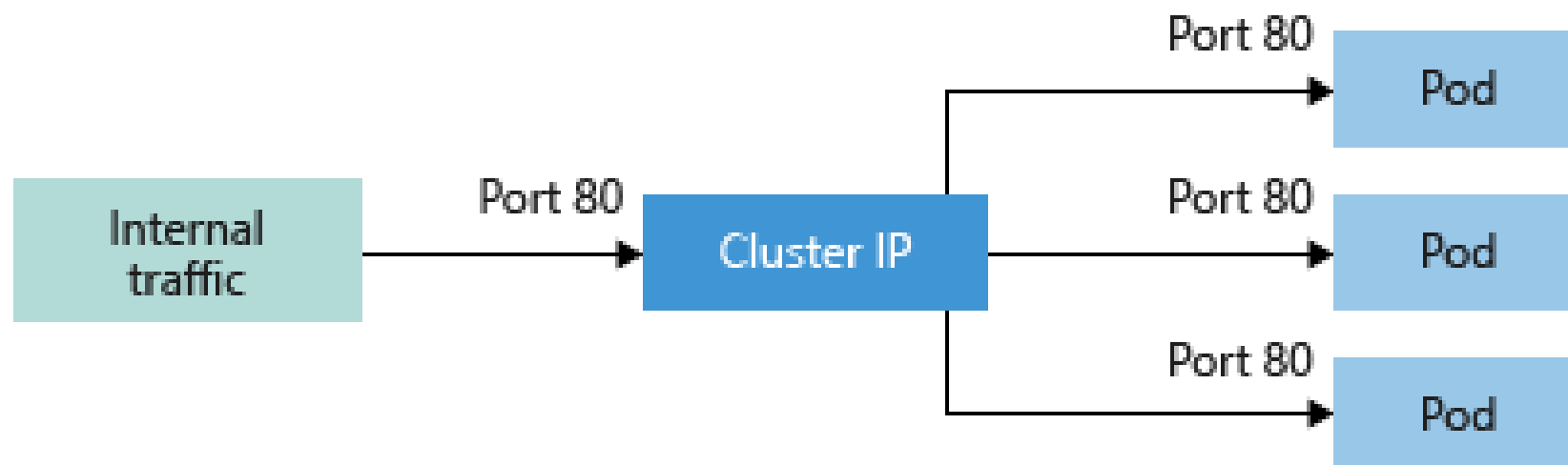
## **LoadBalancer:**

Creates an Azure LB resource, configures an external IP address, and connects the pods to the load balancer backend pool

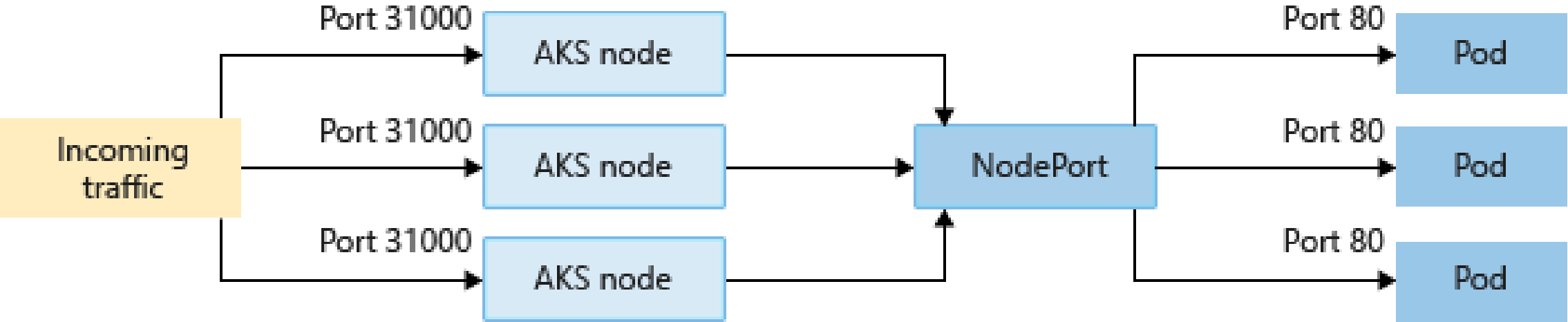
Customer traffic allowed through load balancing rules on the desired ports



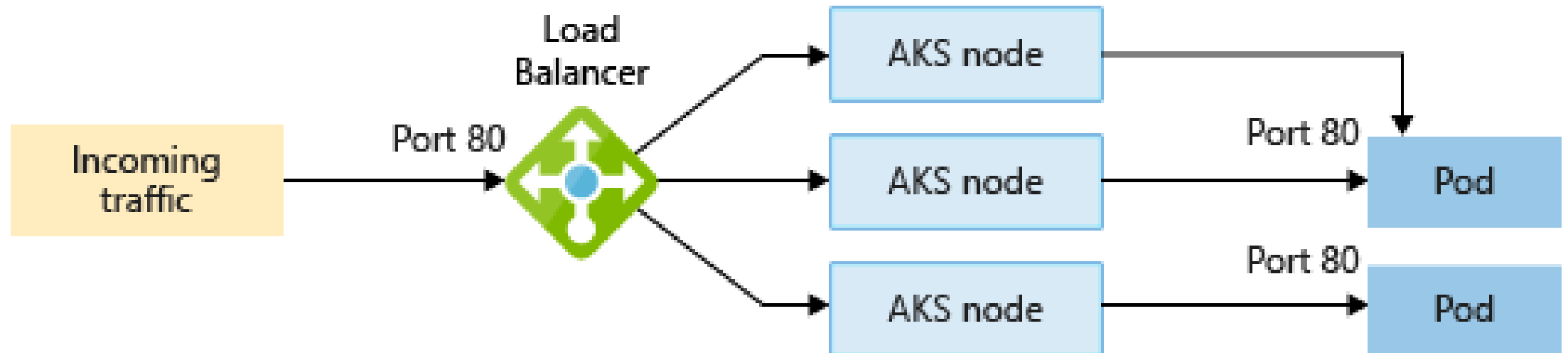
## ServiceType: ClusterIP



# ServiceType: NodePort

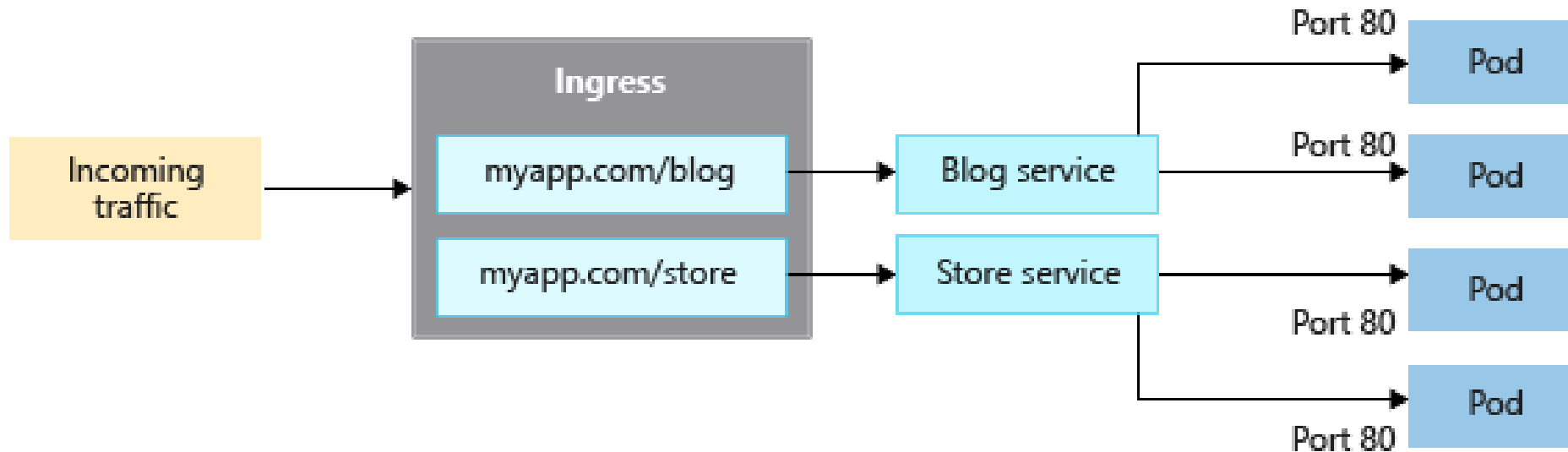


## ServiceType: LoadBalancer



# Ingress

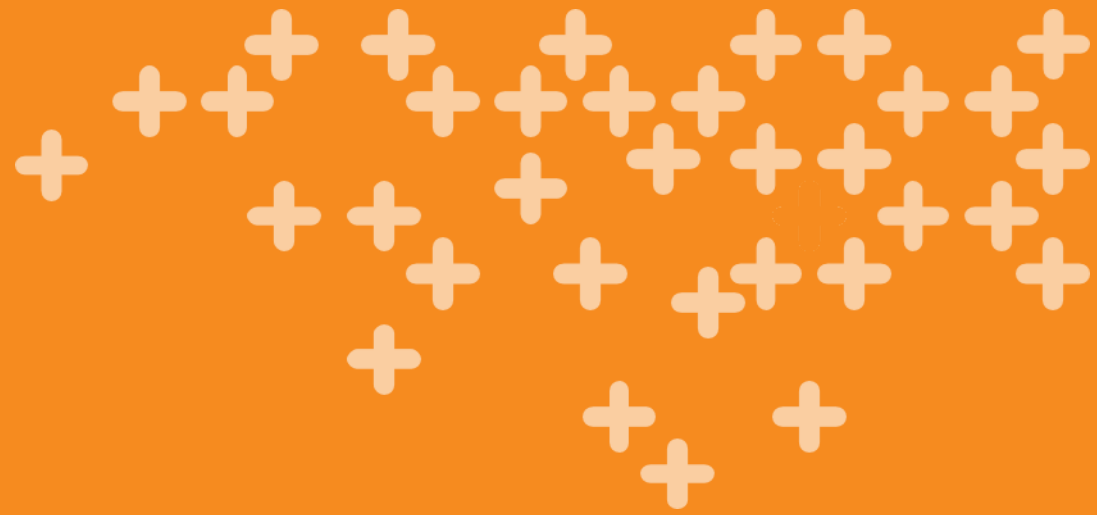
*Ingress controllers* work at Layer 7 so can use more intelligent rules to distribute traffic  
For example, route HTTP traffic to different applications based on the inbound URL



# Kubectl

Command-line tool for running commands against Kubernetes clusters. Deploy applications, manage cluster resources.

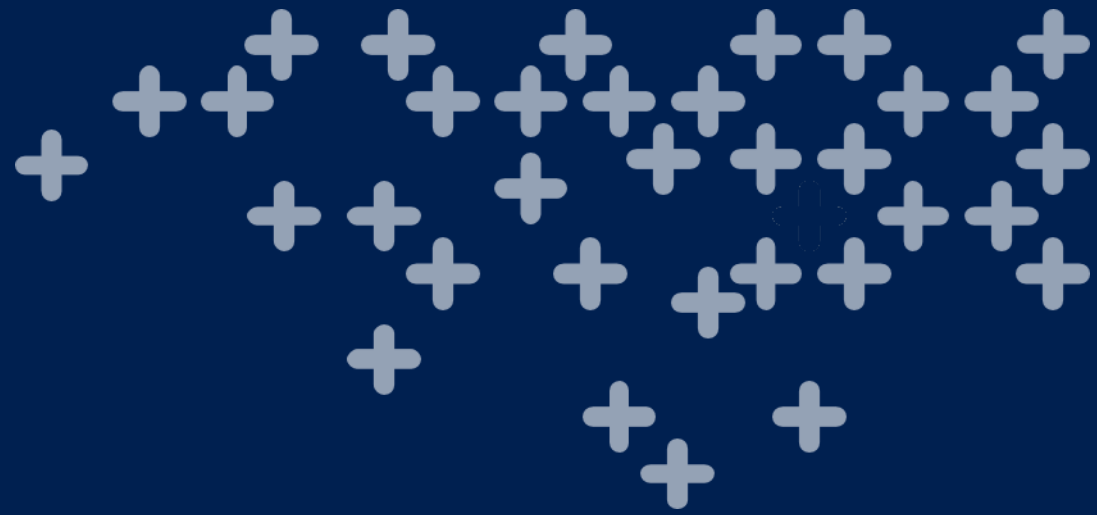
Common Commands	
annotate	Add/update annotations for resources
apply	Apply configuration changes
autoscale	Scale pods managed by a replication controller
certificate	Modify certificate resources
cluster-info	Display endpoint information about master and services
config	Modify kubeconfig files
cp	Copies files to/from containers
describe	Show detailed state about resources
exec	Execute a command against a container
label	Add/update labels for resources
logs	Print the logs for a container
run	Run an image on a cluster



# DEMO

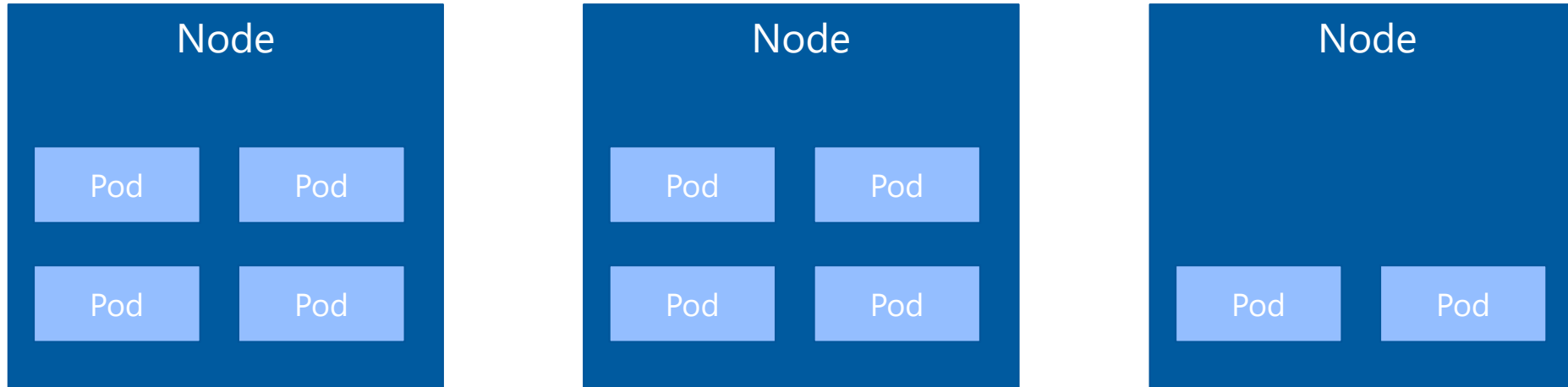
Deploy Applications to AKS





# Scaling Azure Kubernetes Service

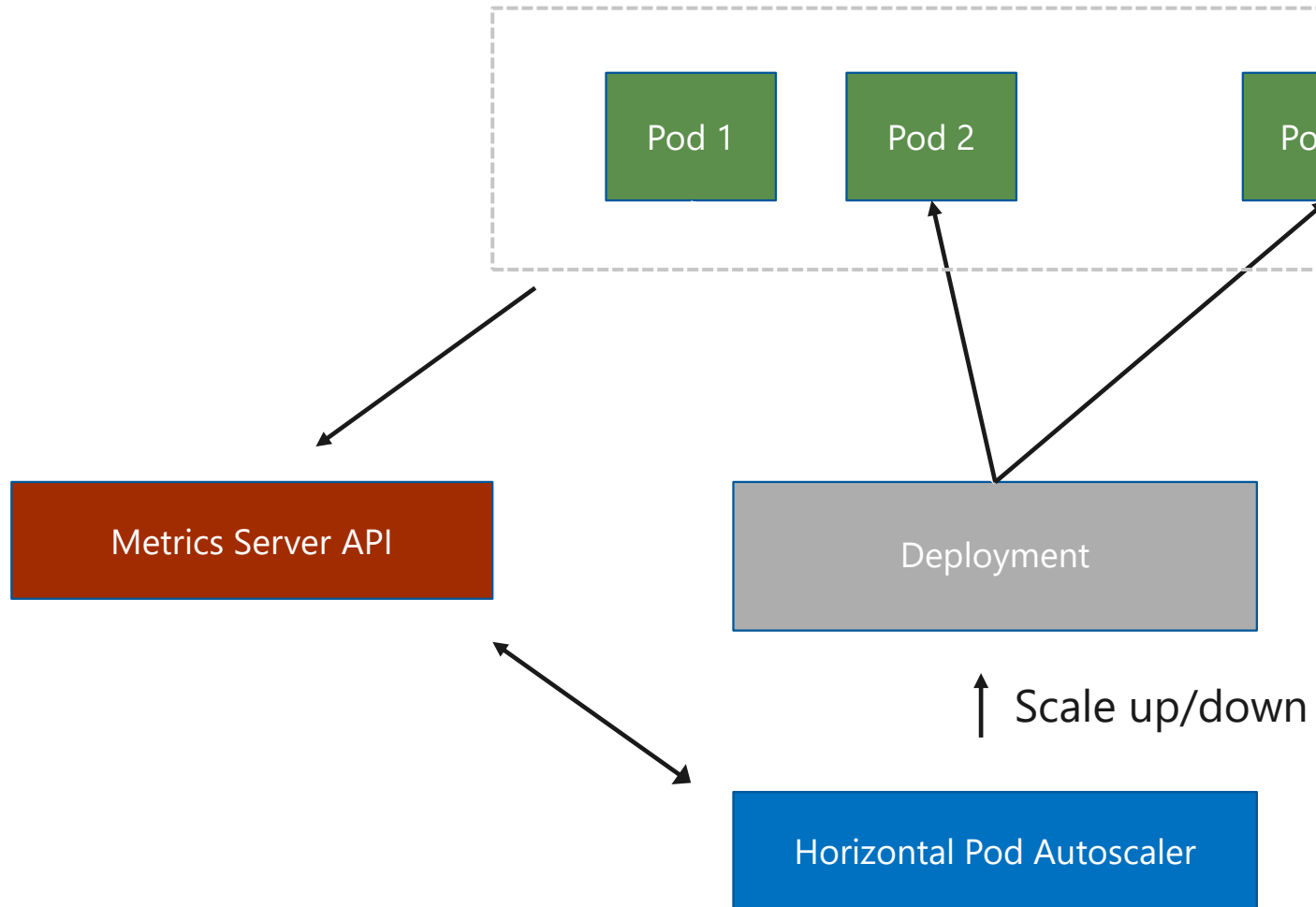
# Scaling with AKS



Horizontal Pod Autoscaling (HPA)

Cluster Autoscaler

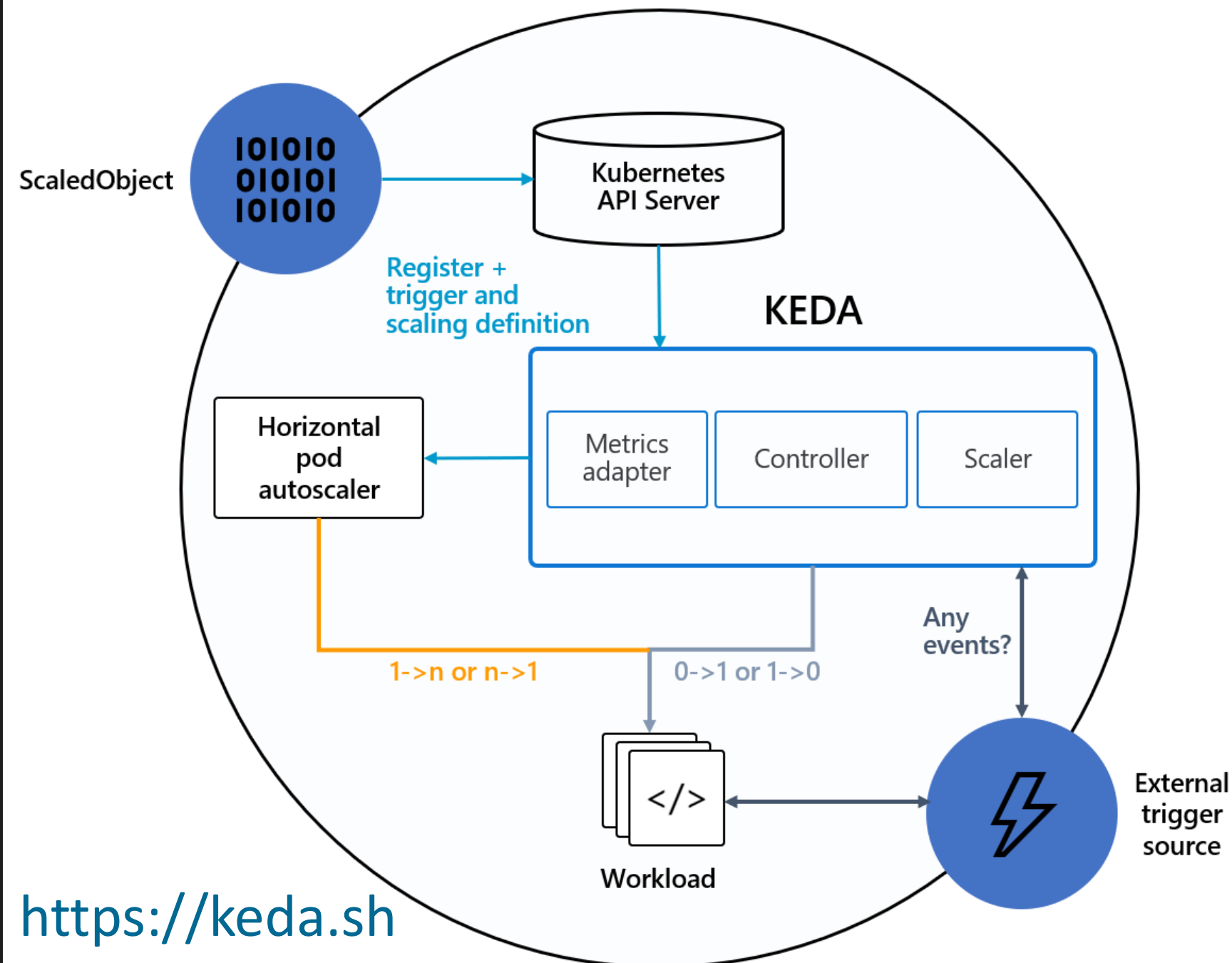
# Horizontal Pod Autoscaler



```
app: qbox
tier: frontend
spec:
  containers:
  - name: frontend
    image: qbox.web:1.0
    ports:
    - containerPort: 80
    resources:
      requests:
        cpu: 250m
      limits:
        cpu: 500m
```

```
kubectl autoscale deployment xyz
--cpu-percent=10
--min=1
--max=10
```

## Kubernetes cluster



## KEDA Scalers

- Apache Kafka
- AWS CloudWatch
- AWS Simple Queue Service
- Azure Event Hub
- Azure Monitor
- Azure Service Bus Queues & Topics
- Azure Storage Queues
- GCP PubSub
- IBM MQ
- Influx DB
- Kafka
- Liiklus
- MongoDB
- MySQL
- Nats Streaming
- Prometheus
- RabbitMQ
- Redis Lists, Streams
- ...

# Cluster Autoscaling

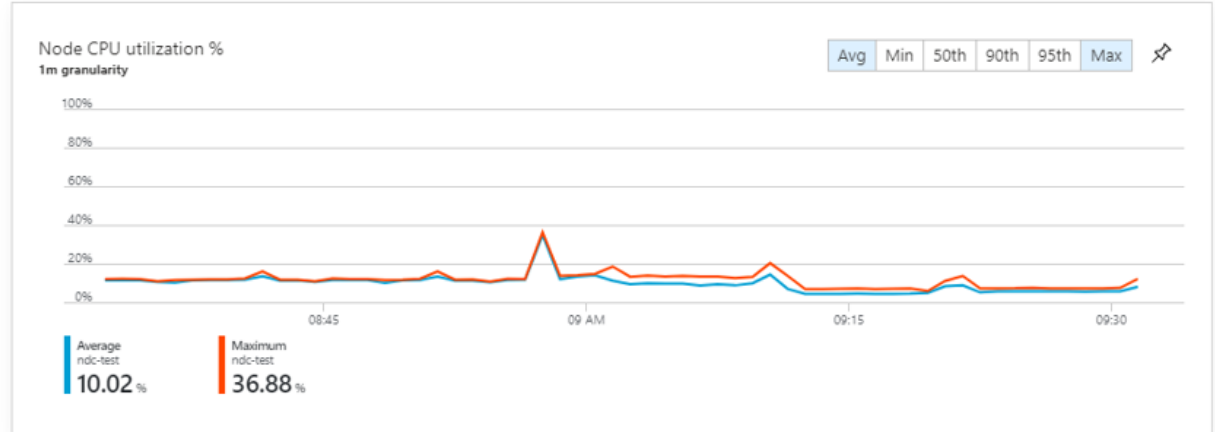
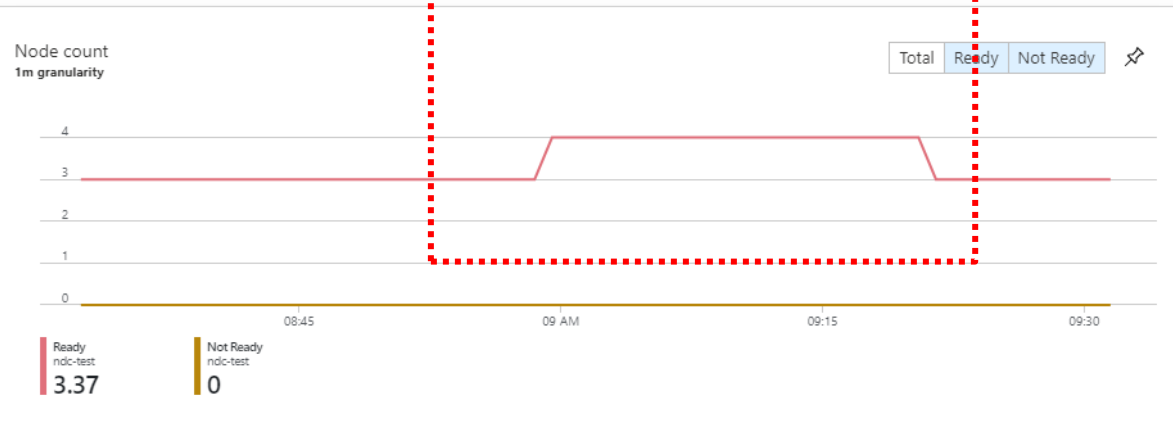
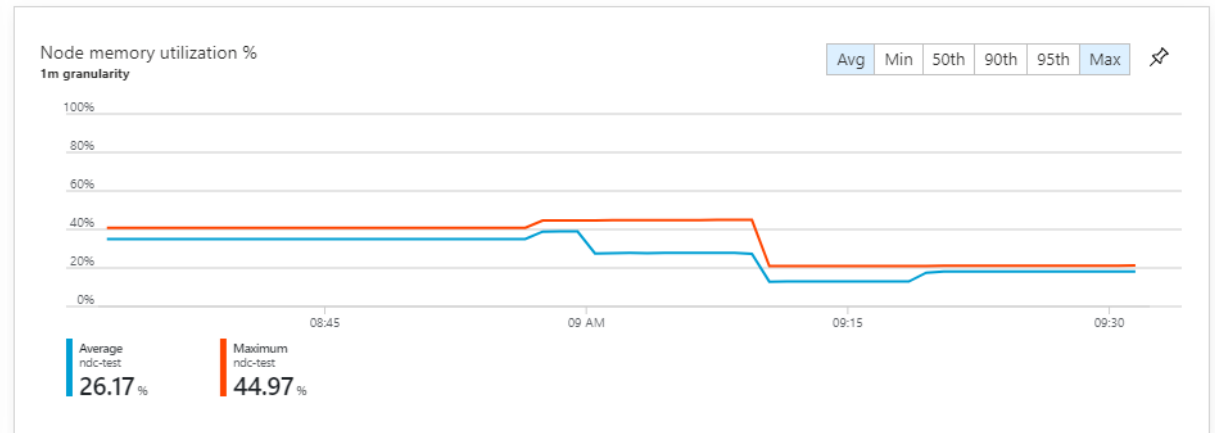
```
"copy": [
  {
    "name": "agentPoolProfiles",
    "count": "[length(parameters('agentPoolProfiles'))]",
    "input": {
      "name": "[if(equals(parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].osType, 'Linux'), if(lessOrEquals(length(para
      "orchestratorVersion": "[parameters('kubernetesVersion')]",
      "maxPods": 250,
      "osDiskSizeGB": 128,
      "count": "[parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].nodeCount]",
      "vmSize": "[parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].nodeVmSize]",
      "osType": "[parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].osType]",
      "vnetSubnetID": "[variables('agentPoolProfiles').vnetSubnetId]",
      "enableAutoScaling": true,
      "maxCount": 2,
      "minCount": 4,
      "type": "VirtualMachineScaleSets",
      "availabilityZones": "[parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].availabilityZones]",
      "mode": "[parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].mode]",
      "enableNodePublicIP": false,
      "nodeLabels": "[parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].nodeLabels]",
      "nodeTaints": "[parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].nodeTaints]"
    }
  }
],
"networkProfile": {
```

# Cluster Autoscaling

Enable fast alerting experience on basic metrics for this Azure Kubernetes Services cluster. Learn more [here](#) Enable

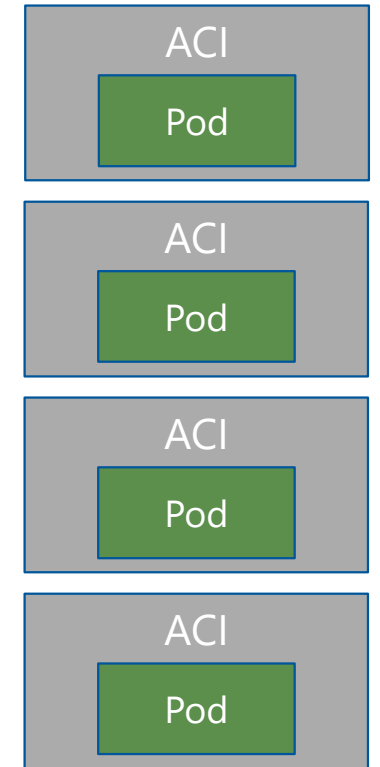
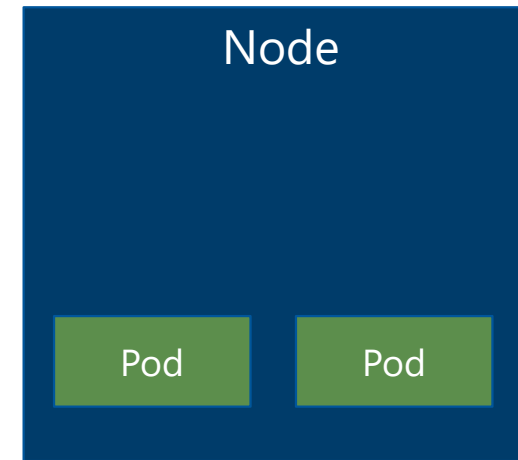
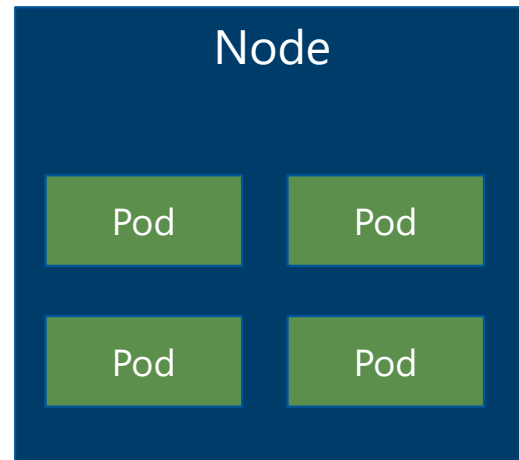
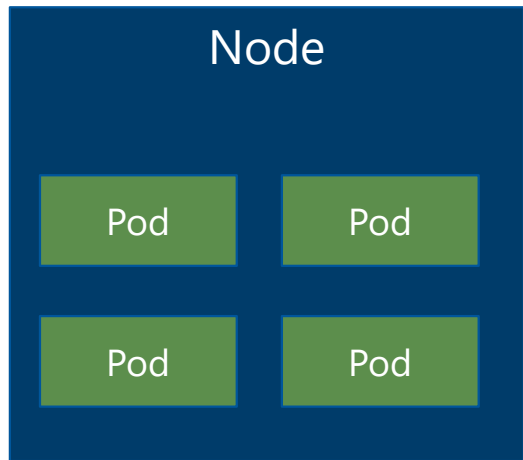
Time range = **Last hour** + Add Filter Live: ☒ Off

What's new Cluster Reports (Preview) Nodes Controllers Containers Deployments (Preview)





# Serverless scaling with AKS

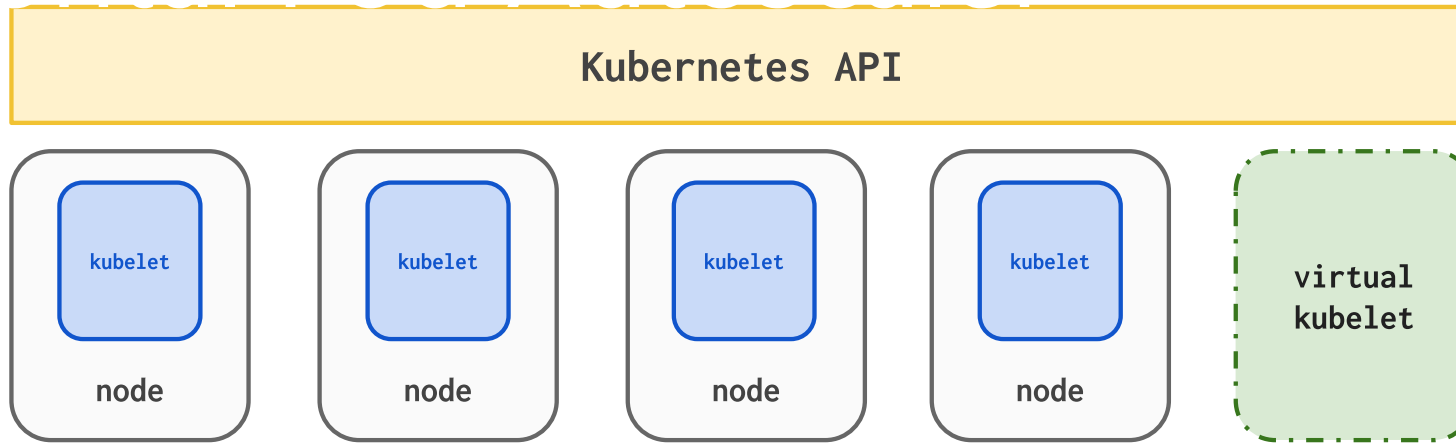


Horizontal Pod Autoscaling (HPA)

Cluster Autoscaler

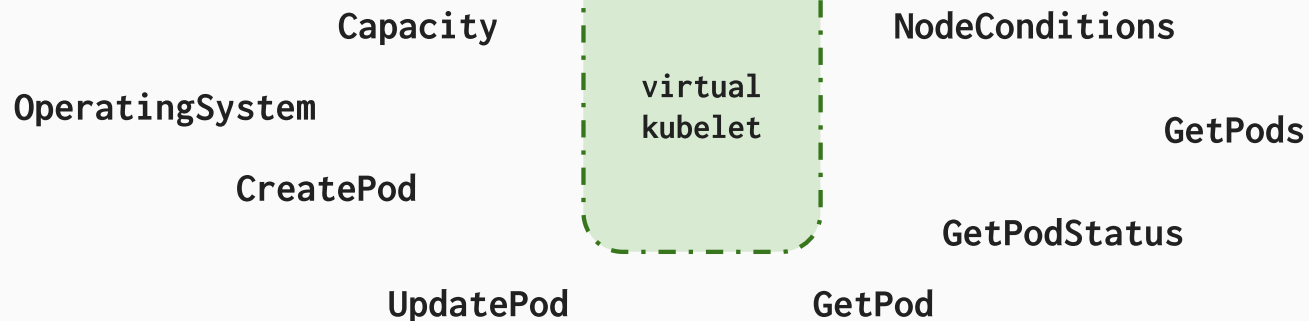
Virtual Nodes

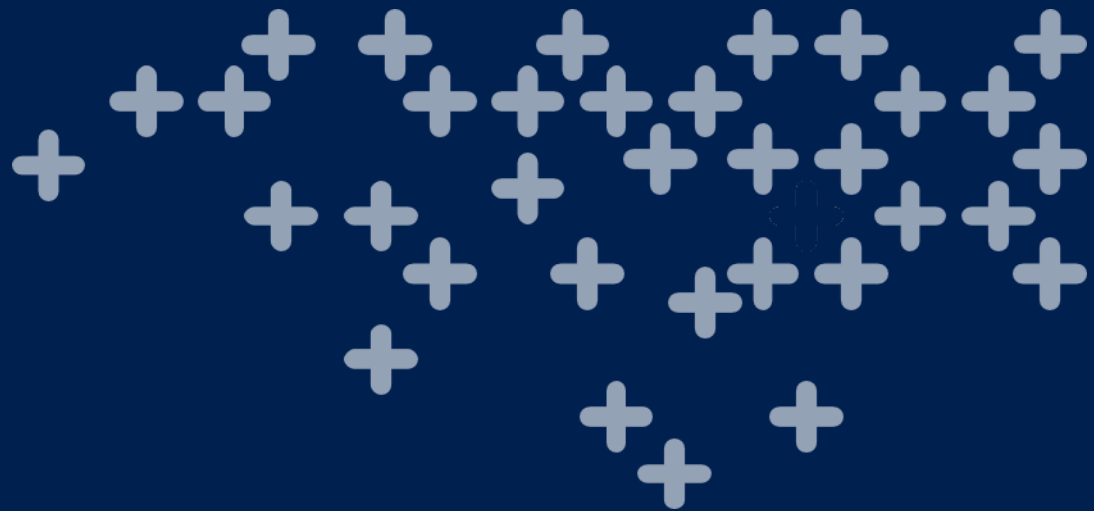
# Virtual Node Architecture



Typical kubelets implement the pod and container operations for each node as usual.

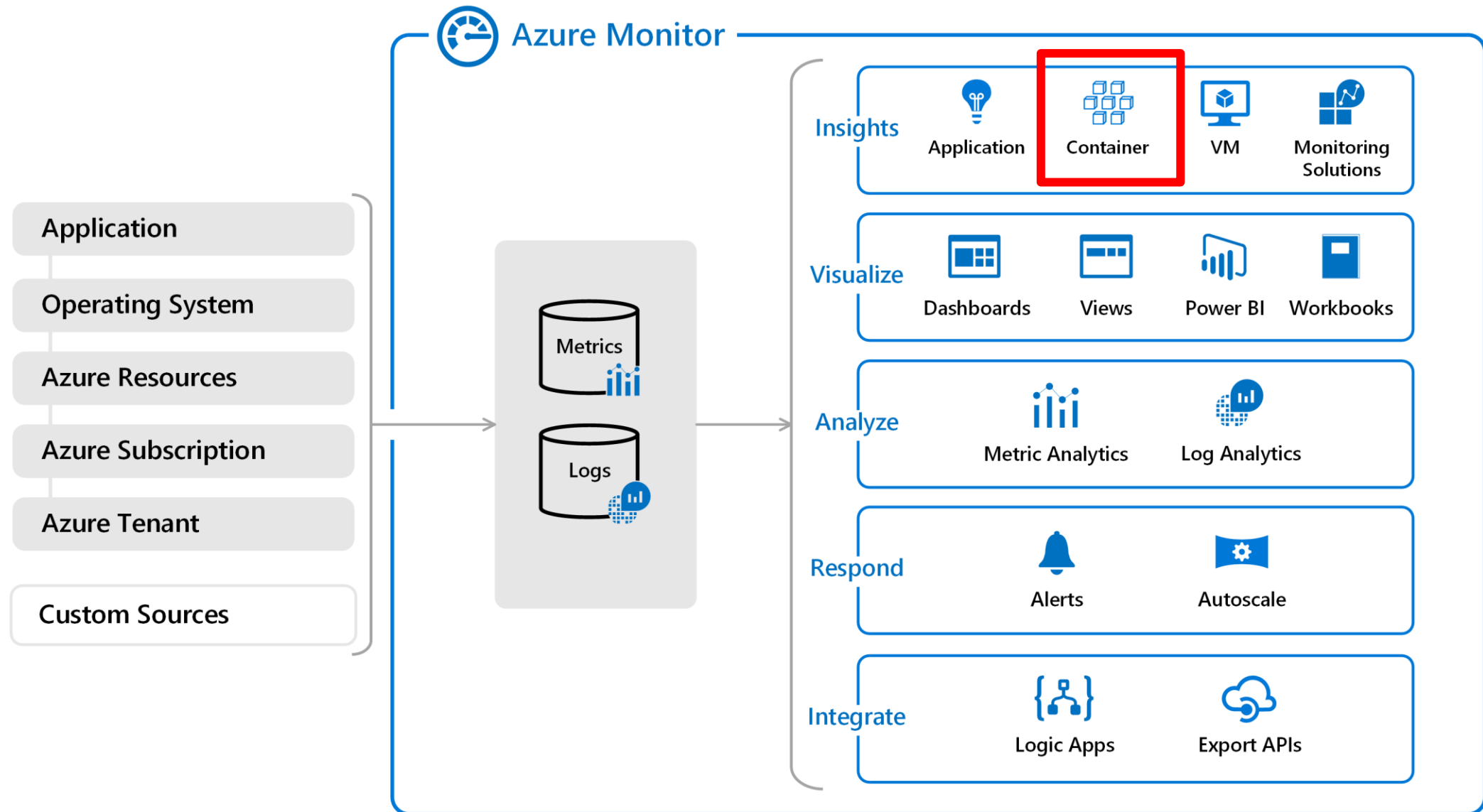
Virtual kubelet registers itself as a “node” and allows developers to deploy pods and containers with their own APIs.



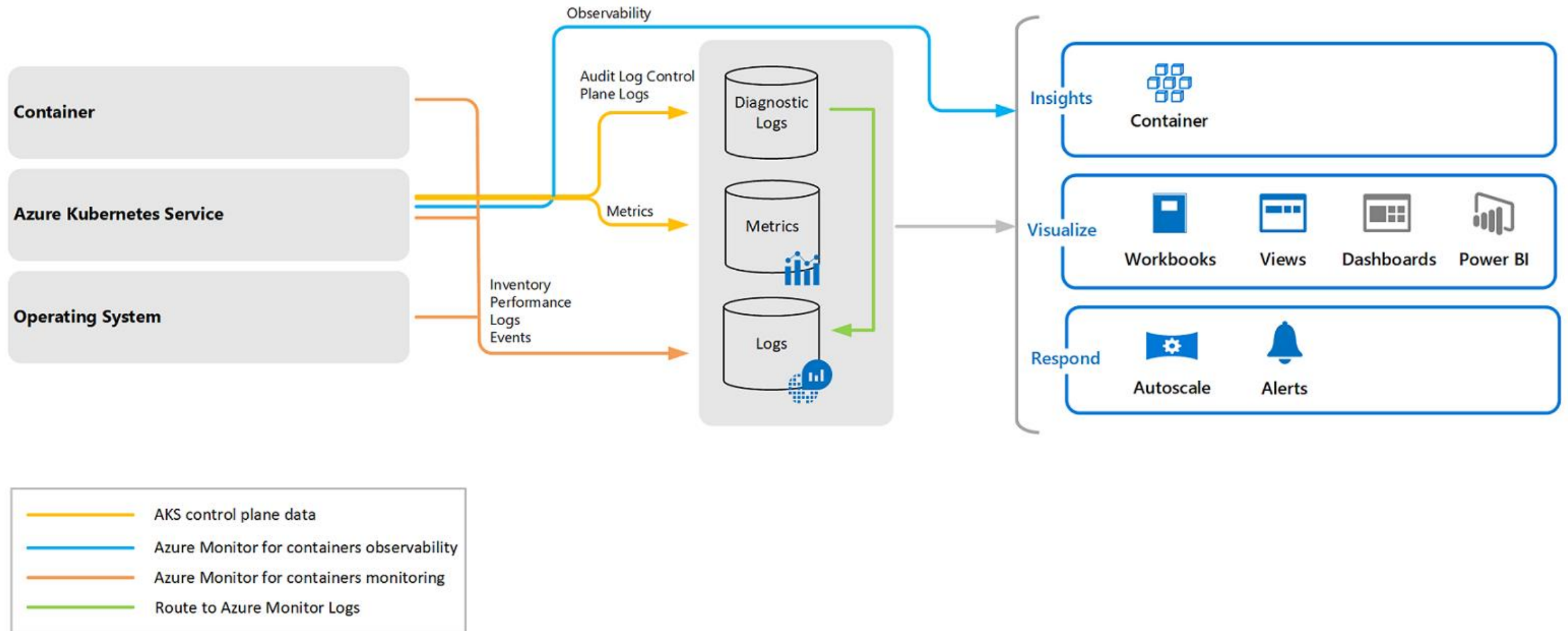


# AKS Monitoring

# Azure Monitor



# Azure Monitor for AKS



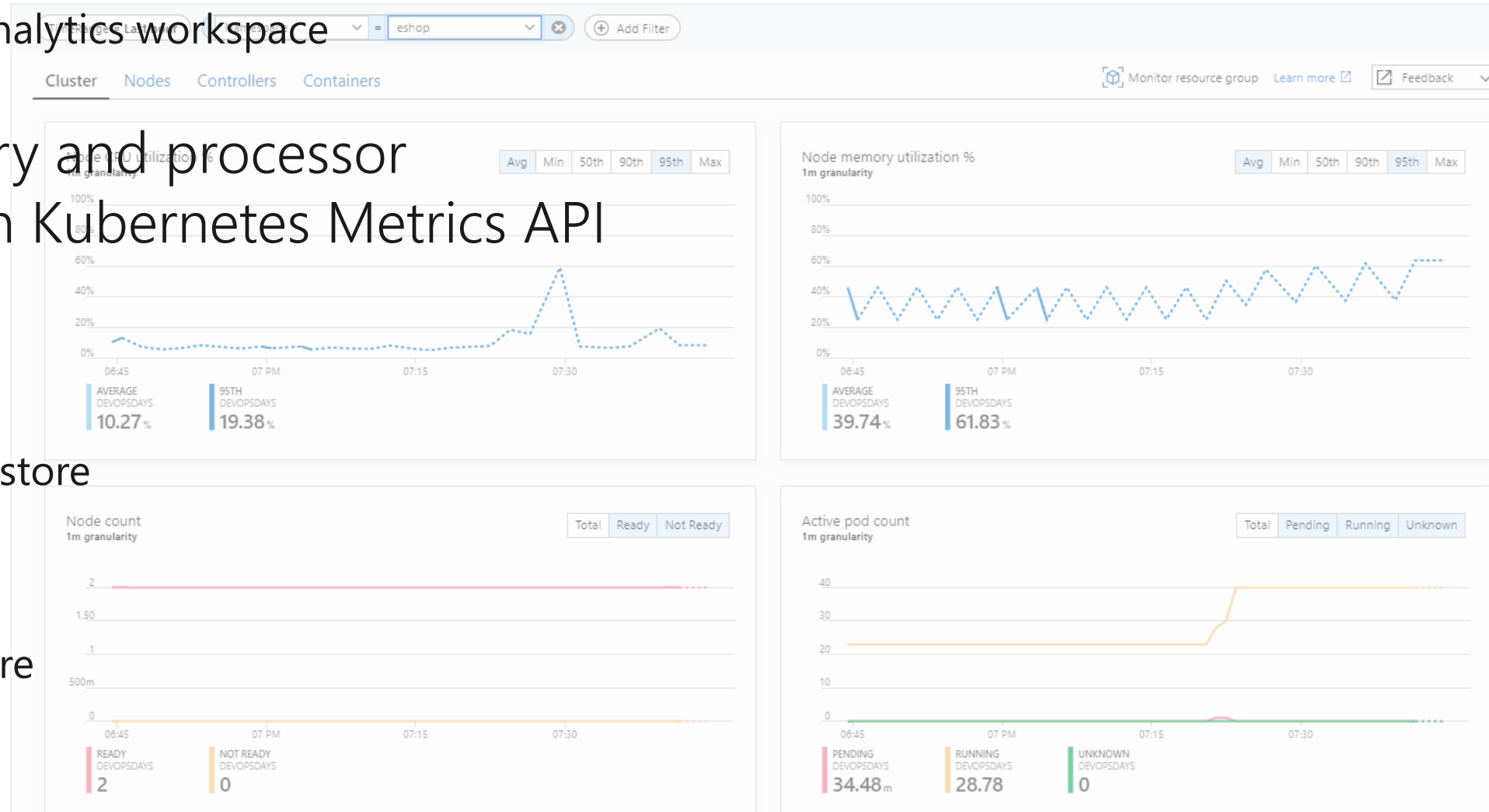
# Azure Monitor for AKS

- Deploys Log Analytics agent in cluster
  - Connects to Log analytics workspace

- Collects memory and processor metrics through Kubernetes Metrics API

- Controllers
- Nodes
- Containers
- Written to metrics store

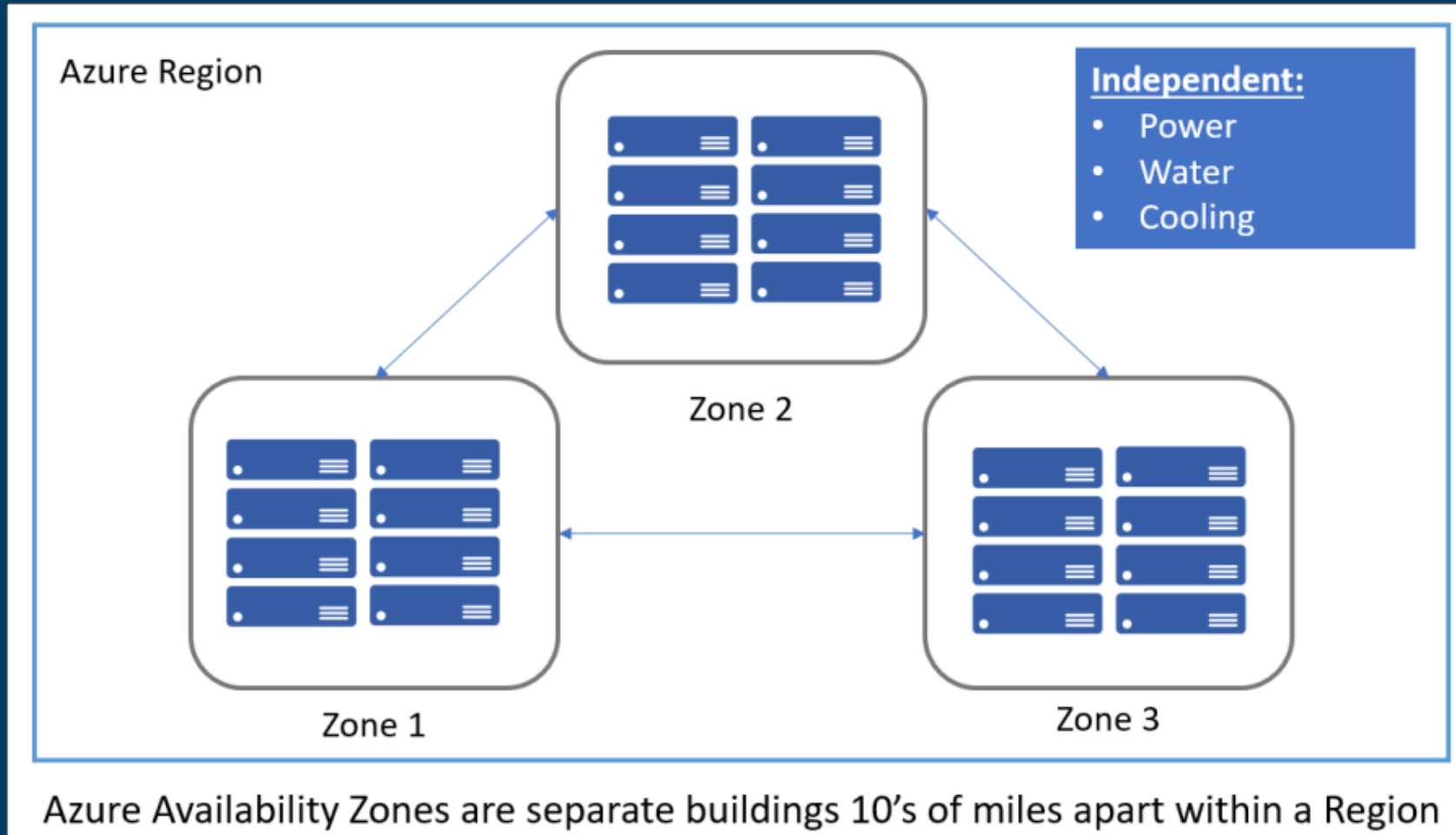
- Container logs
  - Written to logs store





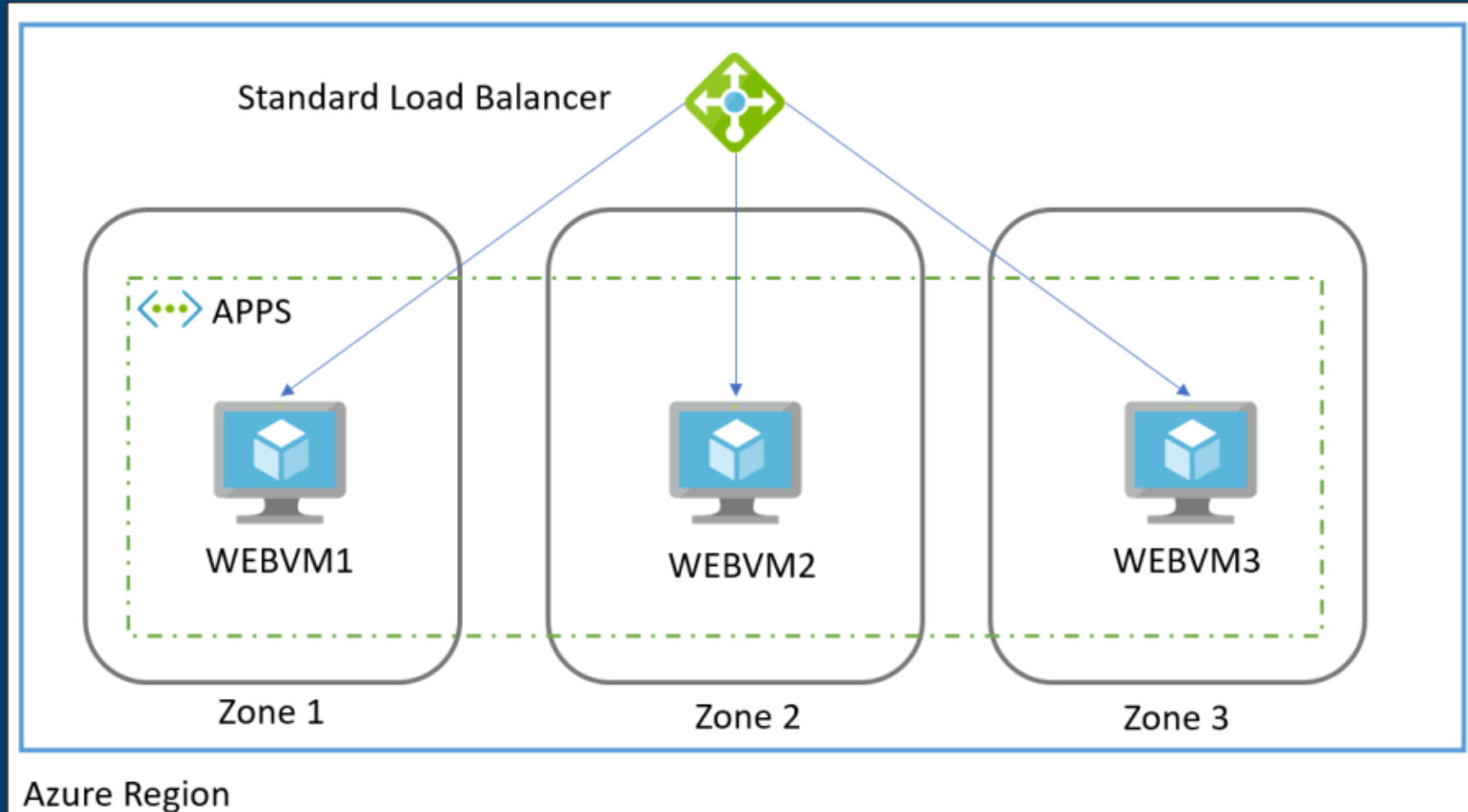
# High Availability

# Azure Regions and Availability Zones

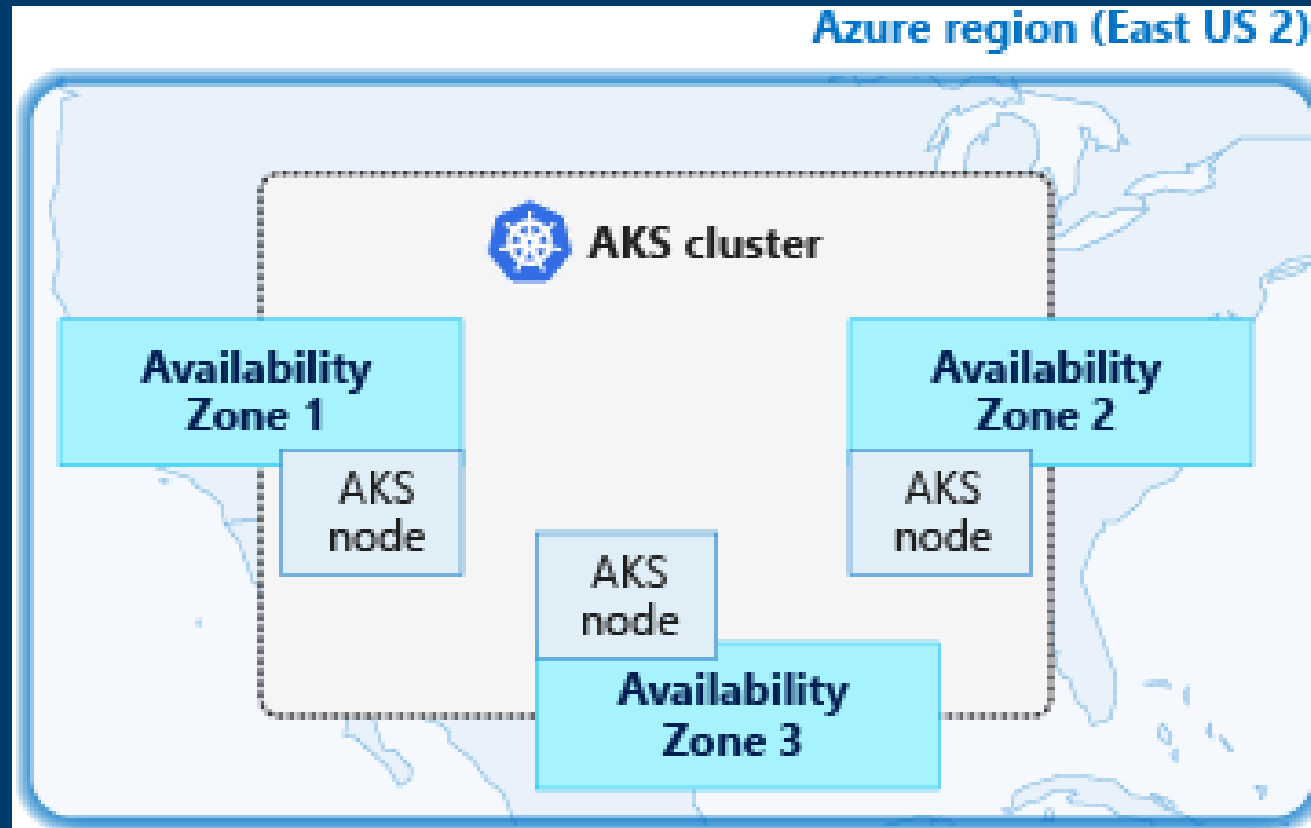




# Azure Regions and Availability Zones



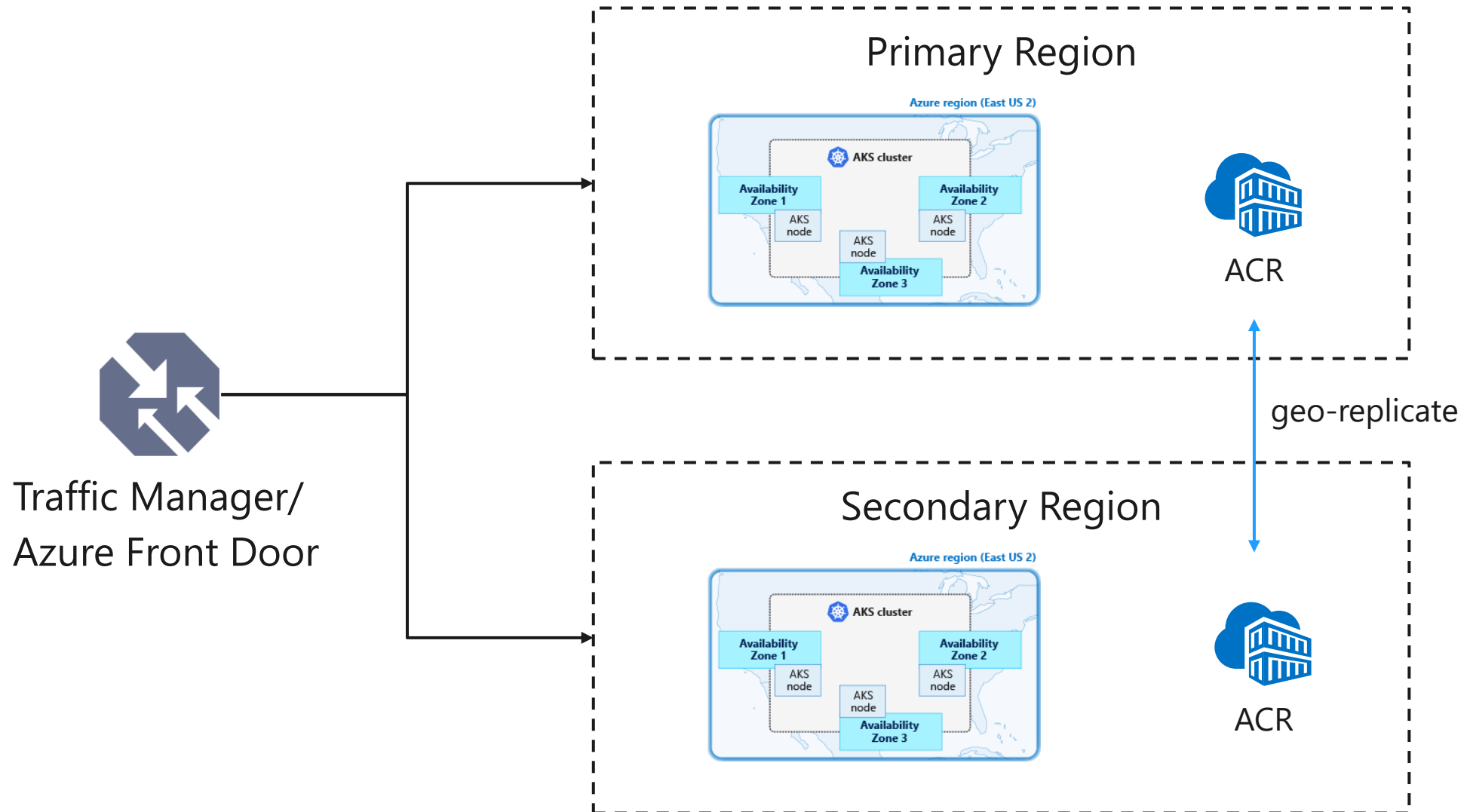
# Cluster Availability and Failover



# Enabling Availability Zones for AKS

```
copy : [
{
  "name": "agentPoolProfiles",
  "count": "[length(parameters('agentPoolProfiles'))]",
  "input": {
    "name": "[if(equals(parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].osType, 'Linux'), parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].name, 'agentPoolProfiles')]",
    "orchestratorVersion": "[parameters('kubernetesVersion')]",
    "maxPods": 250,
    "osDiskSizeGB": 128,
    "count": "[parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].nodeCount]",
    "vmSize": "[parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].nodeVmSize]",
    "osType": "[parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].osType]",
    "vnetSubnetID": "[variables('agentPoolProfiles').vnetSubnetId]",
    "enableAutoScaling": "[if(parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].enableAutoScaling, 'true', 'false')]",
    "maxCount": "[if(parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].enableAutoScaling, '[parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].maxCount]', '0')]",
    "minCount": "[if(parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].enableAutoScaling, '[parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].minCount]', '0')]",
    "type": "VirtualMachineScaleSets",
    "availabilityZones": ["1","2","3"],
    "mode": "[parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].mode]",
    "enableNodePublicIP": false,
    "nodeLabels": "[parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].nodeLabels]",
    "nodeTaints": "[parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].nodeTaints]"
  }
}]
```

# Cluster Availability and Failover





Azure Kubernetes Docs

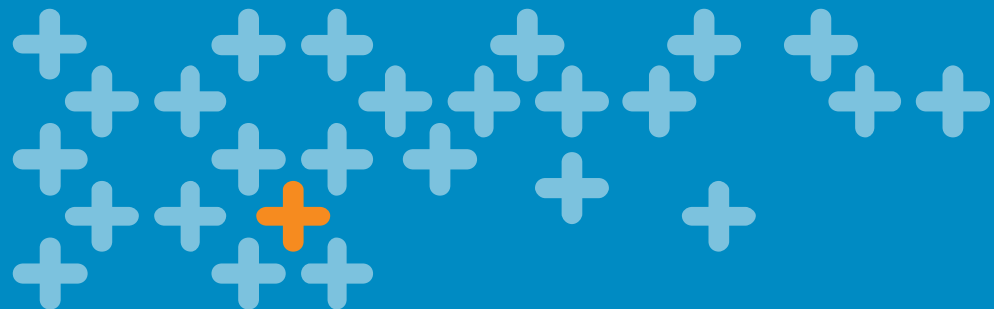
<https://docs.microsoft.com/en-us/azure/aks>

Kubernetes Learning Path

<https://azure.microsoft.com/en-us/resources/kubernetes-learning-path/>

AKS Learning Module

<https://docs.microsoft.com/en-us/learn/modules/intro-to-azure-kubernetes-service/>



# Thank you!

**Jakob Ehn**

@jakobehn

<https://blog.ehn.nu>



Living in Sweden?  
Sign up for some free stickers!

<https://activesolution.se/stickers>





## Championing Azure: AI & Machine Learning

Jan 18th	AI & ML Intro – Alan Smith & Peter Örneholm
Jan 25th	Democratizing AI – Peter Örneholm
Feb 1st	Machine Learning Theory – Alan Smith
Feb 8th	Reinforcement Learning in Gaming – Alan Smith & Eve Pardi
Feb 15th	Azure Custom Vision – Alan Smith
Feb 22nd	Azure Machine Learning – Robert Folkesson

## Championing Cloud Native

Jan 19th	Introduction to Cloud Native Options - Chris Klug
Jan 26th	Event Driven Architecture - Alan Smith
Feb 2nd	Serverless Architectures in Microsoft Azure - Alan Smith
Feb 9th	Containers and Dockers - Chris Klug
Feb 16th	Azure Container Apps - Jakob Ehn
Feb 23rd	Azure Kubernetes Service - Jakob Ehn

