DevOps with Azure Kubernetes Service

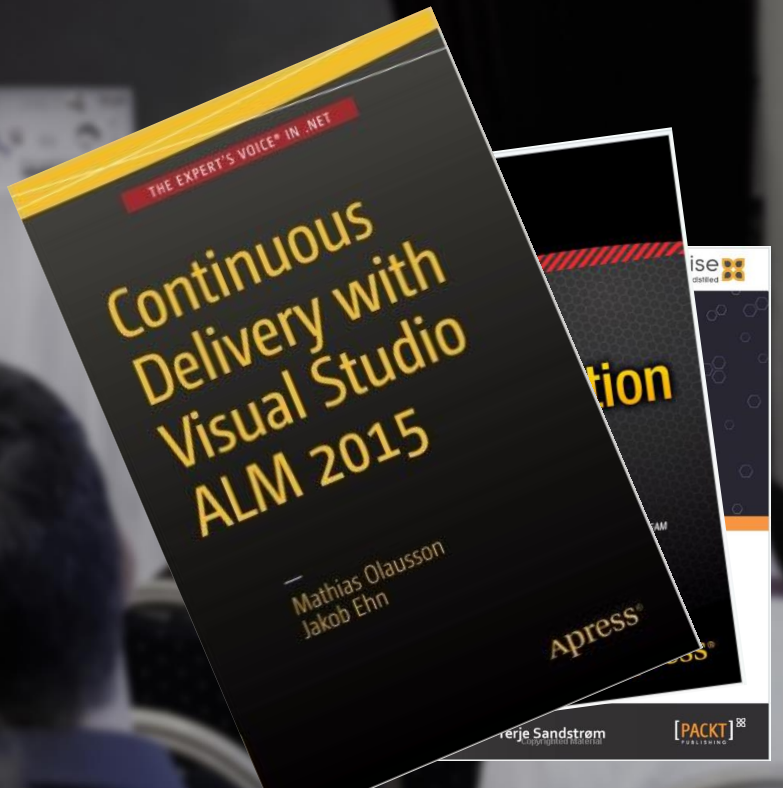
NDC London 2021

@jakobehn https://blog.ehn.nu jakob.ehn@activesolution.se

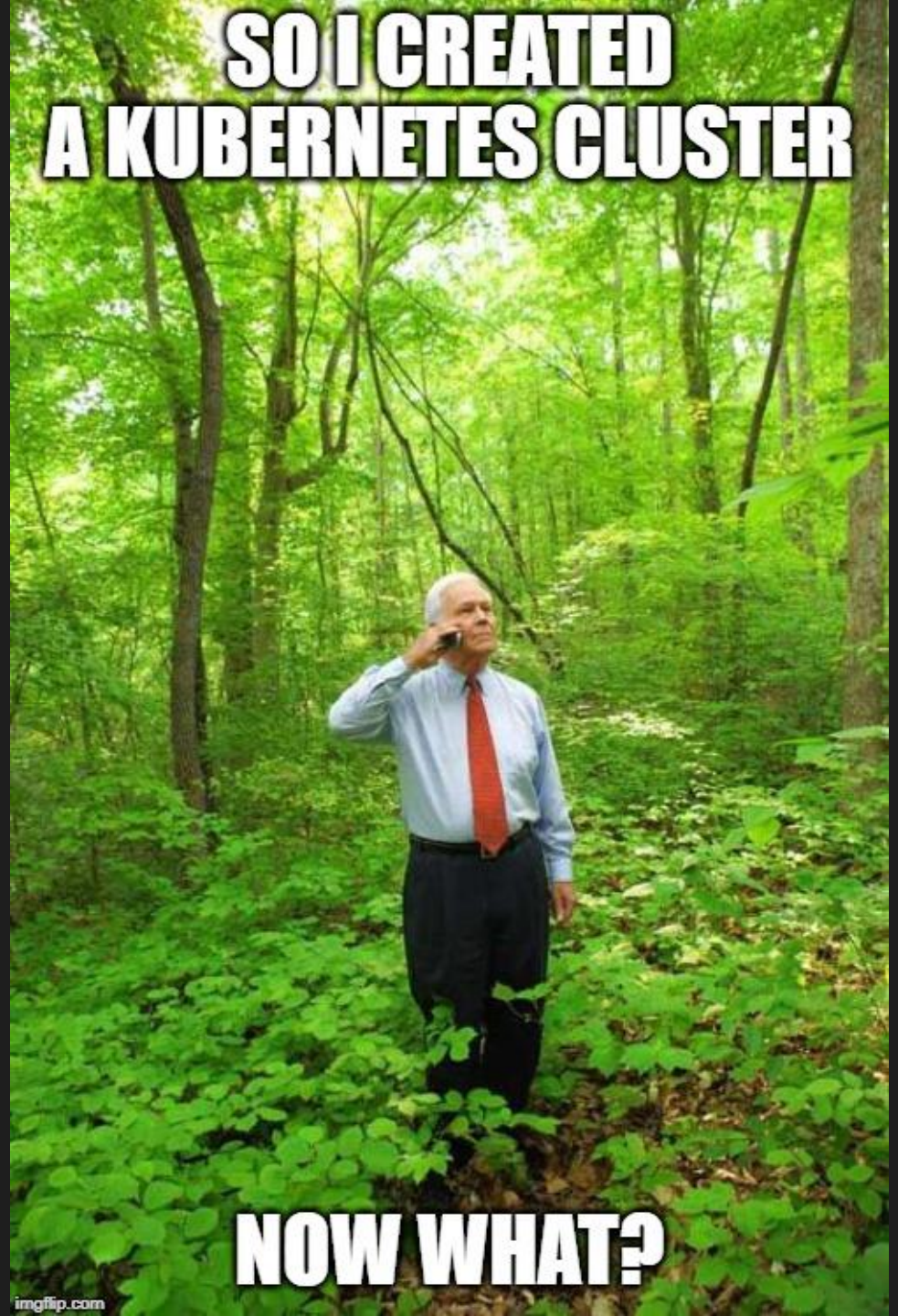
Jakob Ehn
Microsoft Azure MVP

https://blog.ehn.nu

@jakobehn



MVP Microsoft® Most Valuable Professional



Infrastructure as Code

Automated Deployments

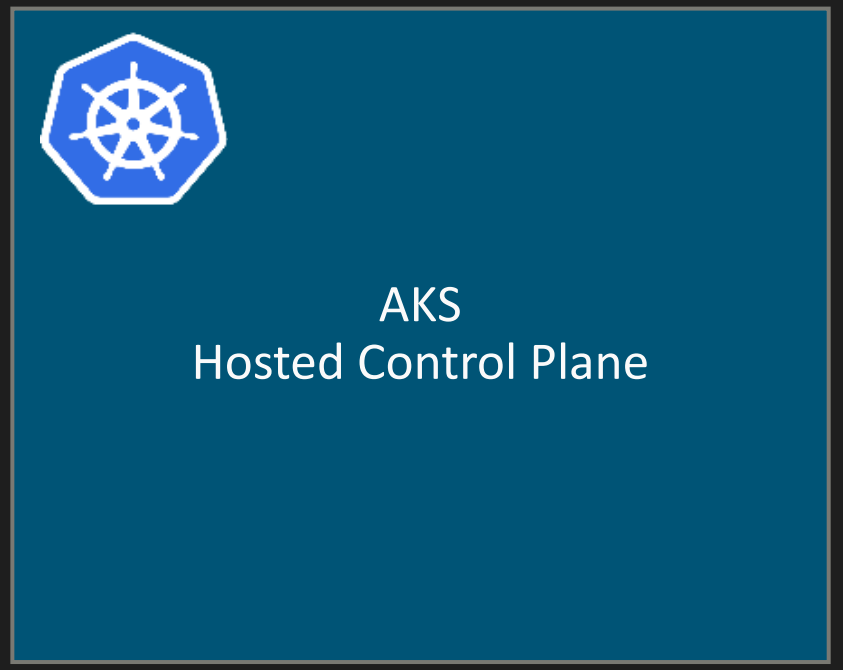
Monitoring

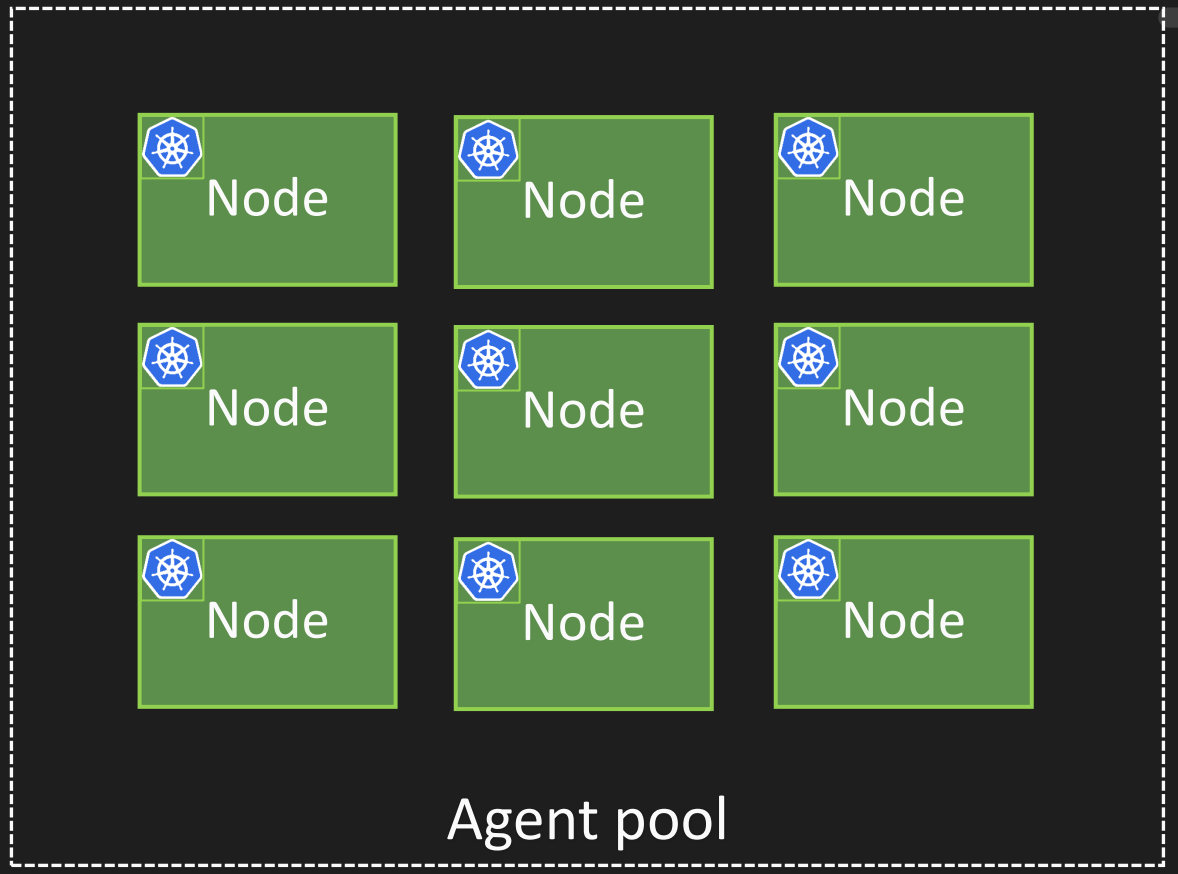
High Availability

Scalability

AKS

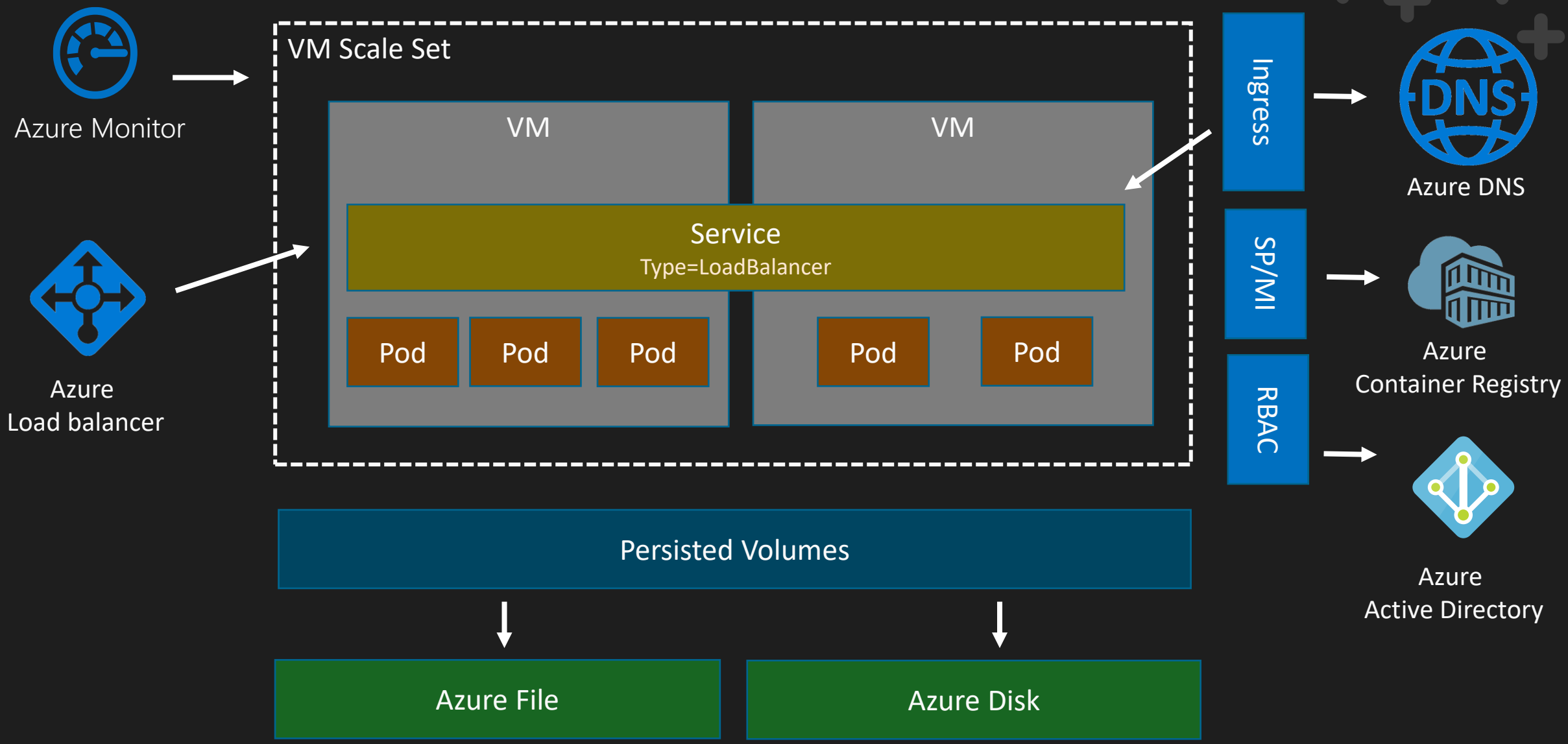
Azure Kubernetes Services

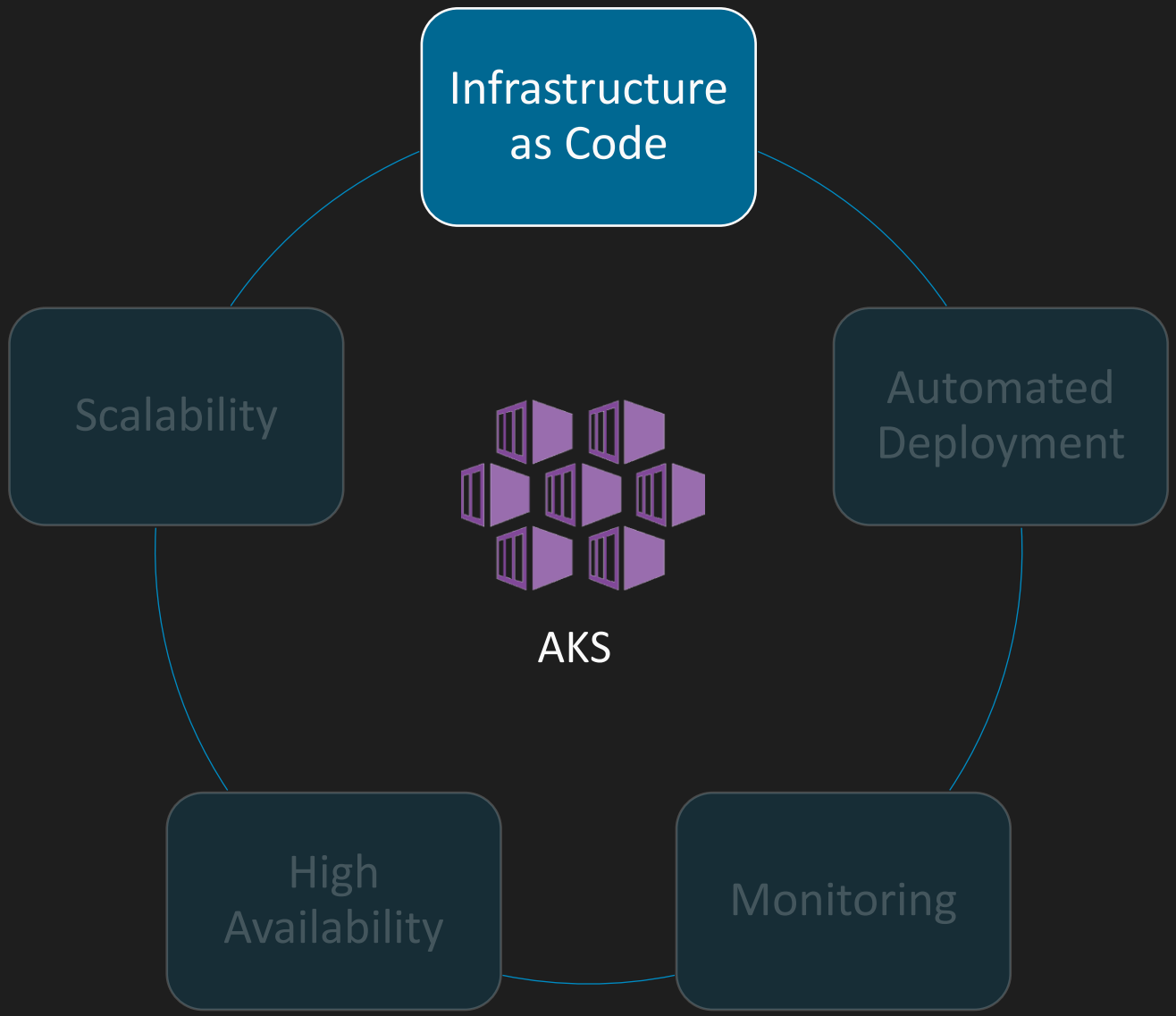




- Automated upgrades and patching
- Cluster (Auto)Scaling
- Self-healing control plane
- Pay for agent nodes only

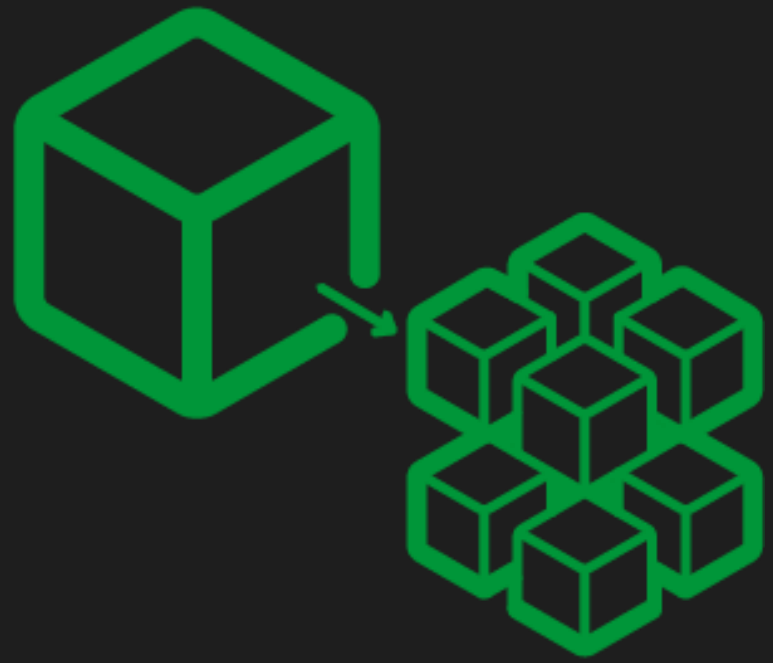
Kubernetes in Azure




Infrastructure as Code
Automated Deployment
Monitoring
High Availability
Scalability
AKS

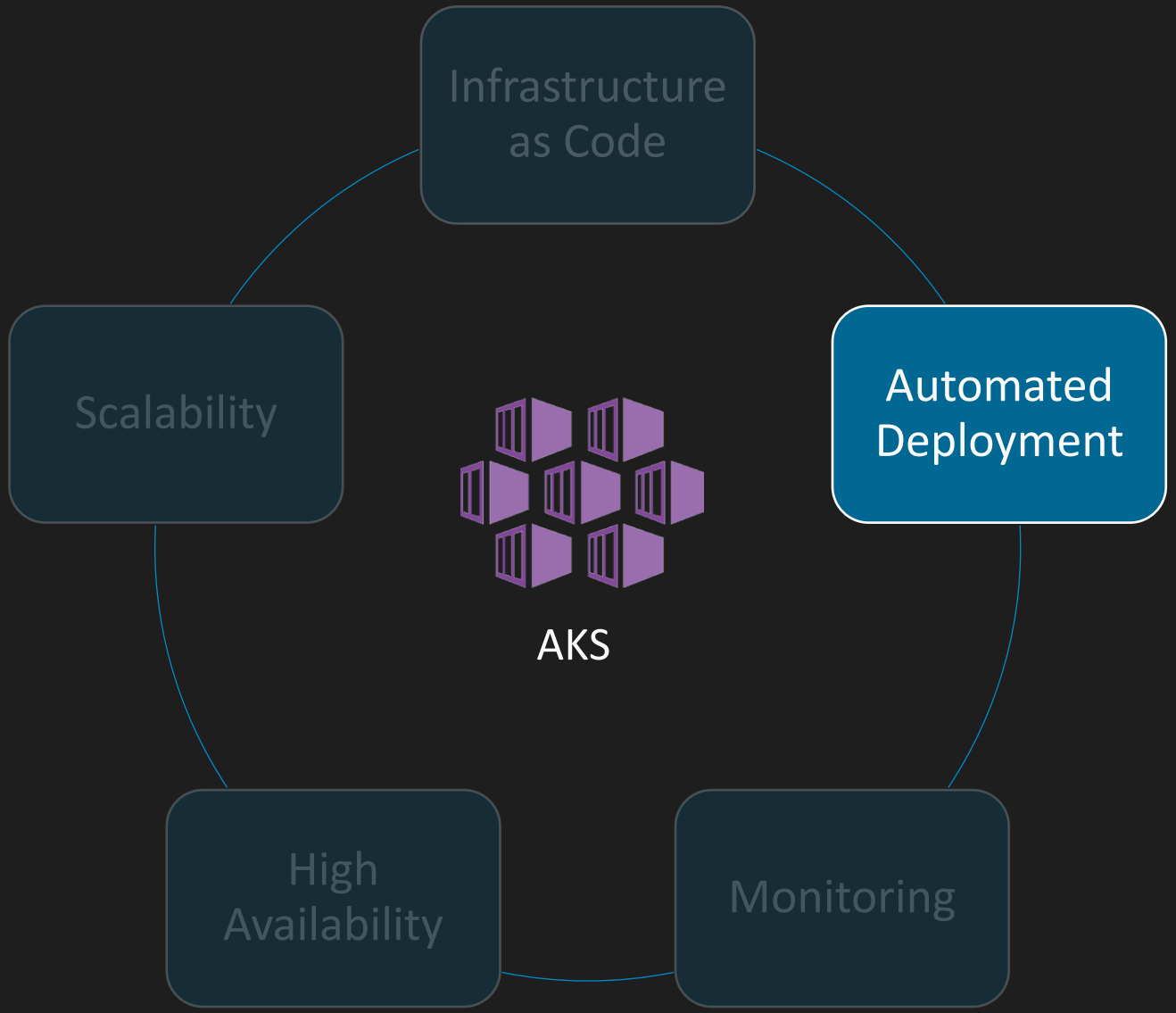
Infrastructure as Code with AKS

- Store cluster definition as code
- Consistent infrastructure
- Repeatable deployments
- Spin up cluster on-demand
- Several options available
  - ARM Templates
  - TerraForm
  - Pulumi
  - Azure CLI

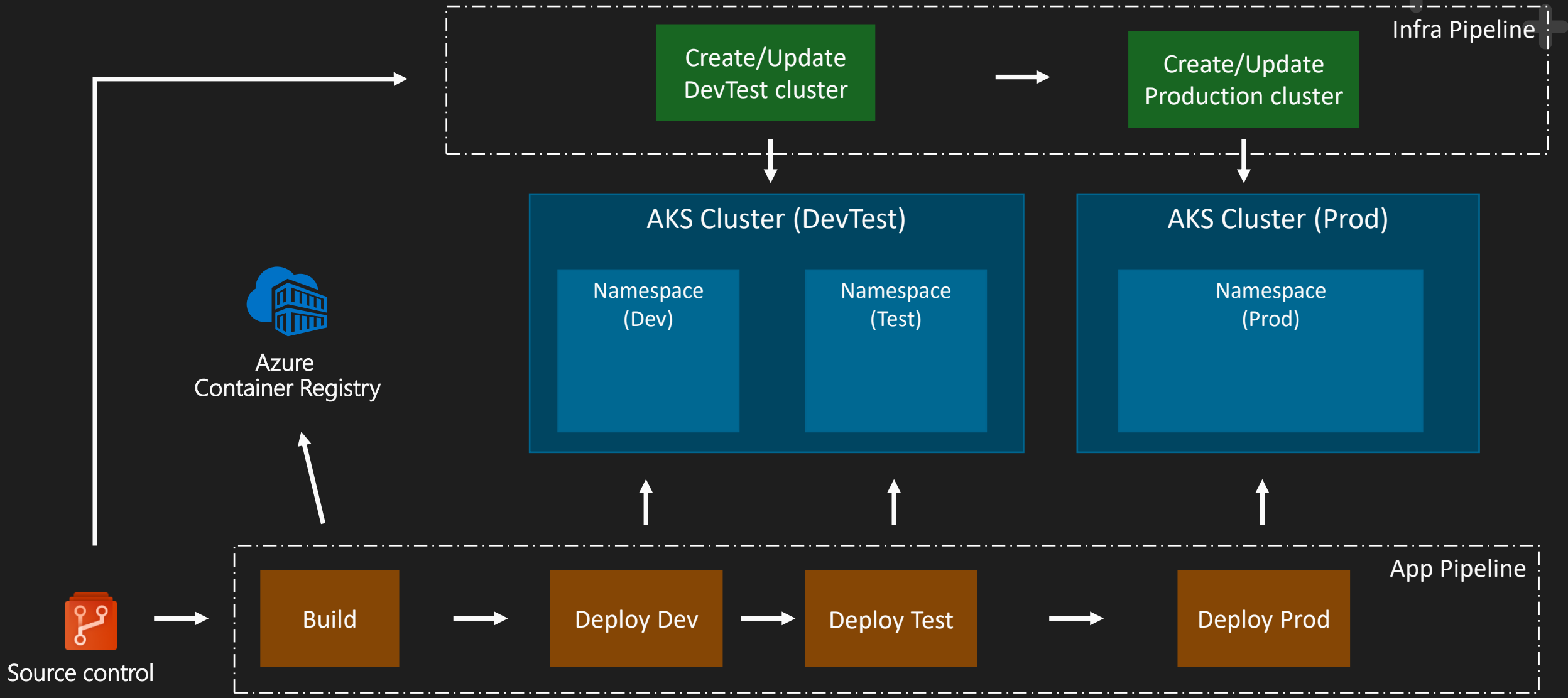


Deploying AKS Infrastructure

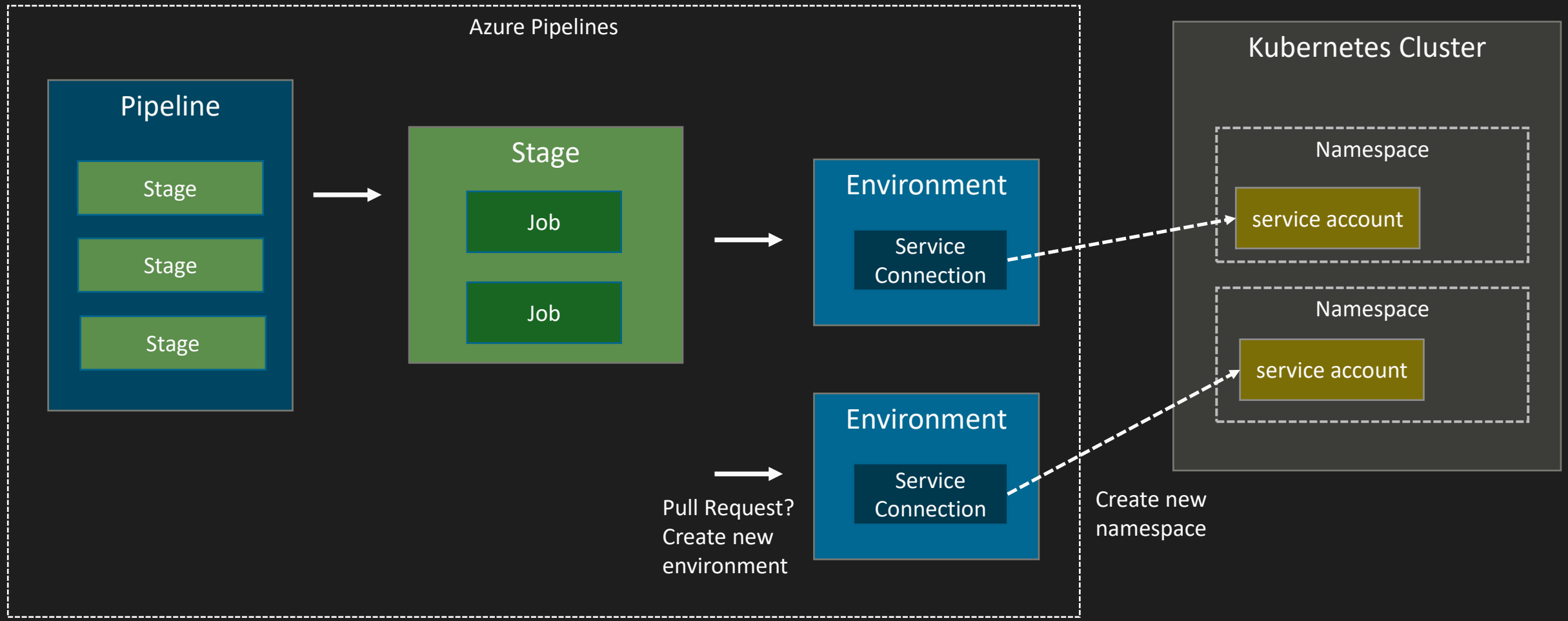
- Virtual Network
- AKS Cluster
- Namespaces
- Ingress controller / Cert Manager
- Permissions
  - Service accounts
  - Azure AD groups
  - k8s roles and role bindings
- …


Infrastructure as Code
Automated Deployment
Monitoring
High Availability
Scalability
AKS

CI/CD with AKS

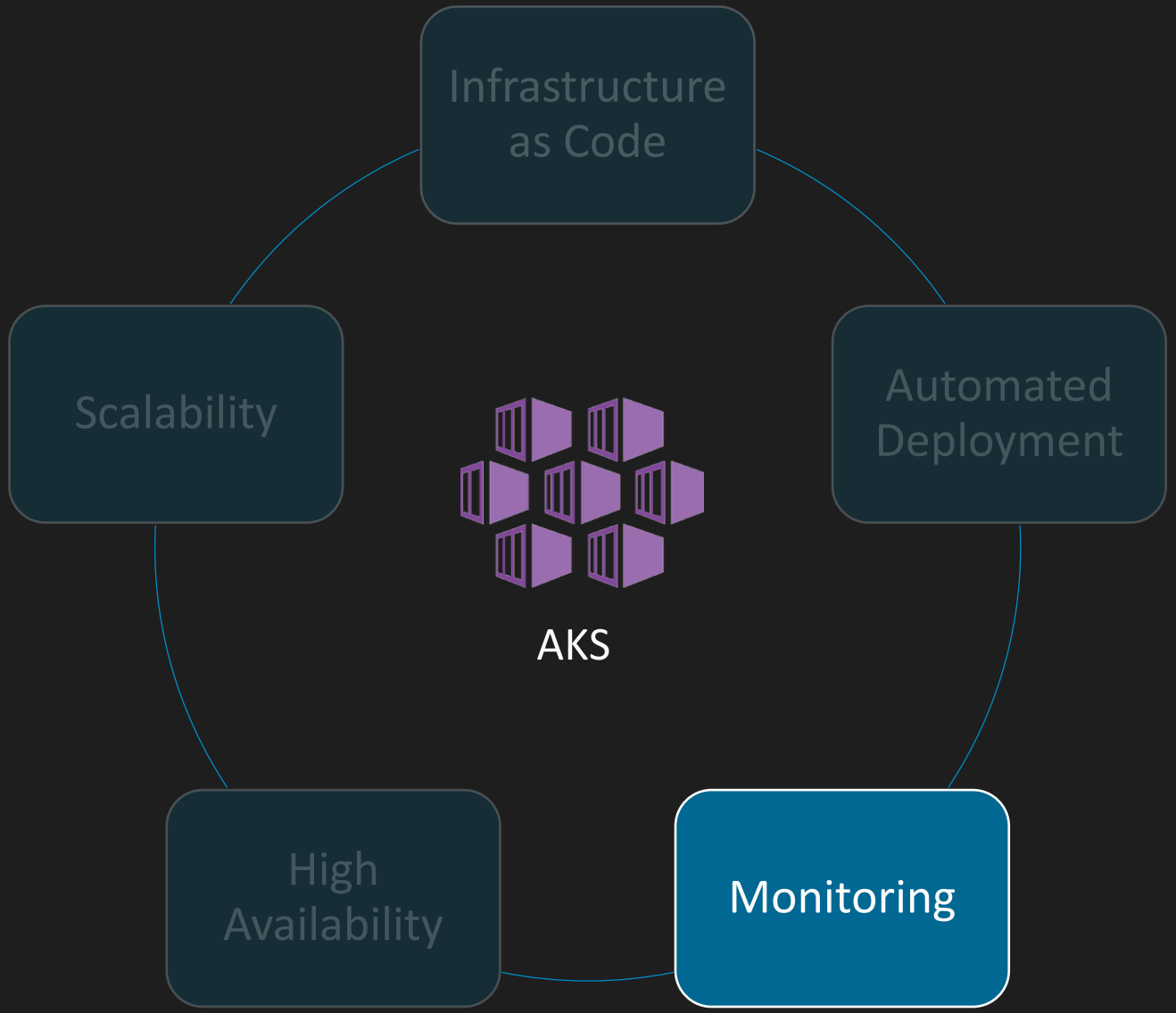


Kubernetes deployments with Azure Pipelines

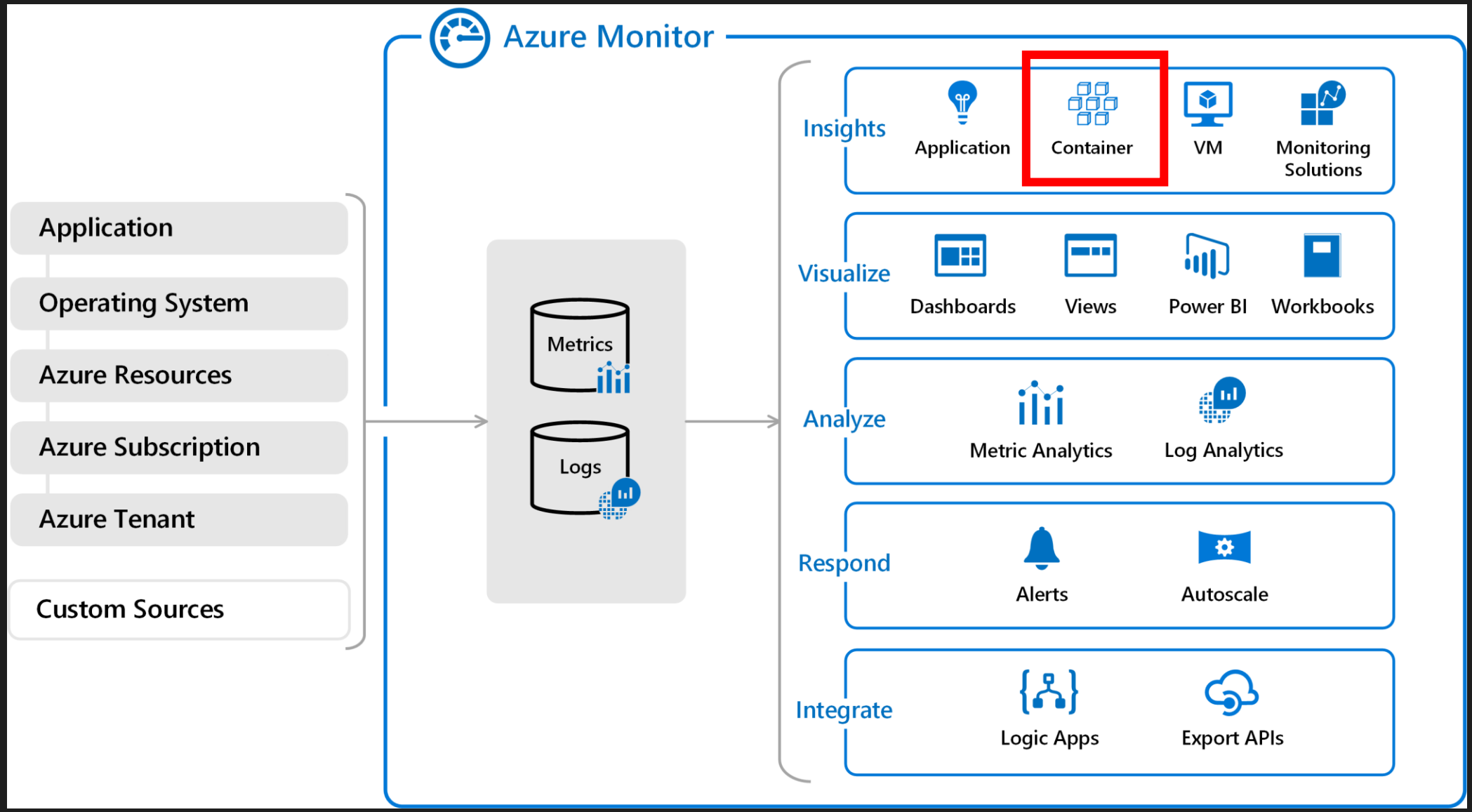


Demo

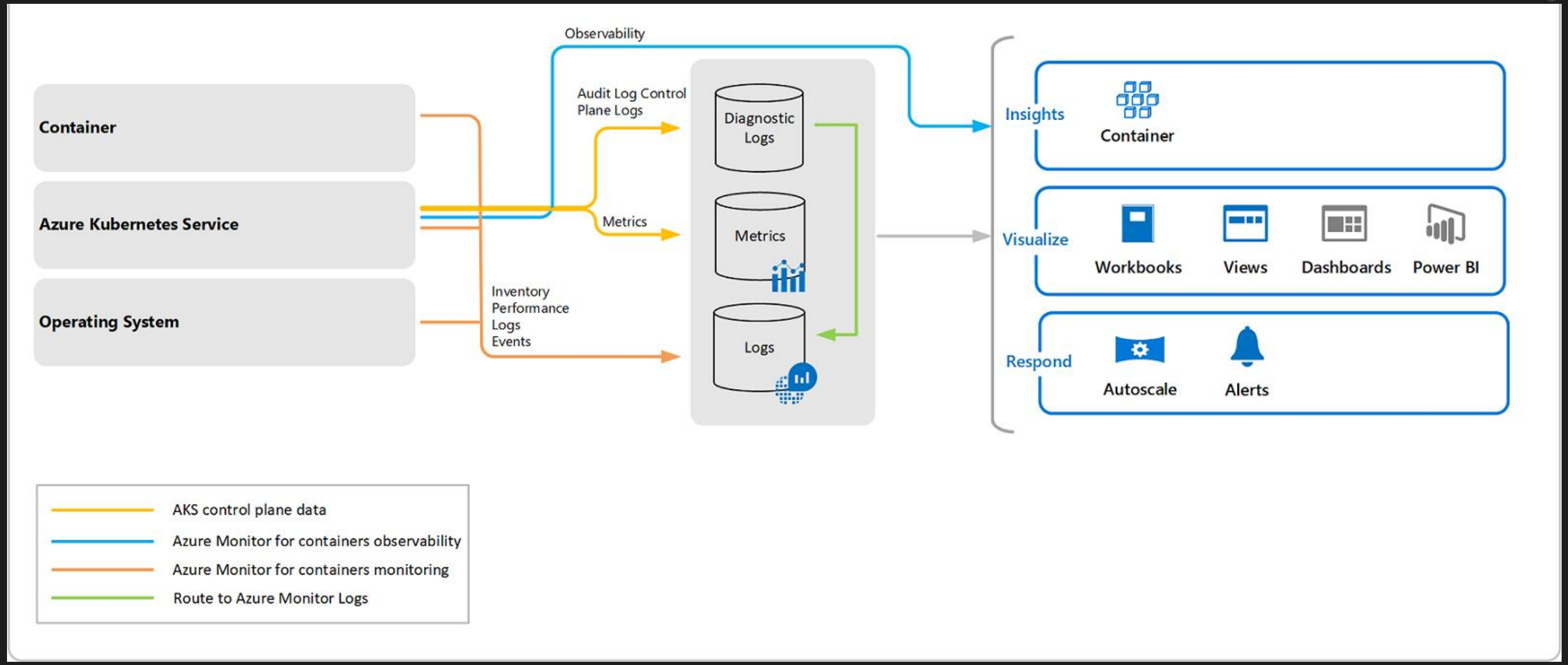
Infrastructure as Code
Automated Deployment


Infrastructure as Code
Automated Deployment
Monitoring
High Availability
Scalability
AKS

Azure Monitor



Azure Monitor for containers



Azure Monitor for AKS

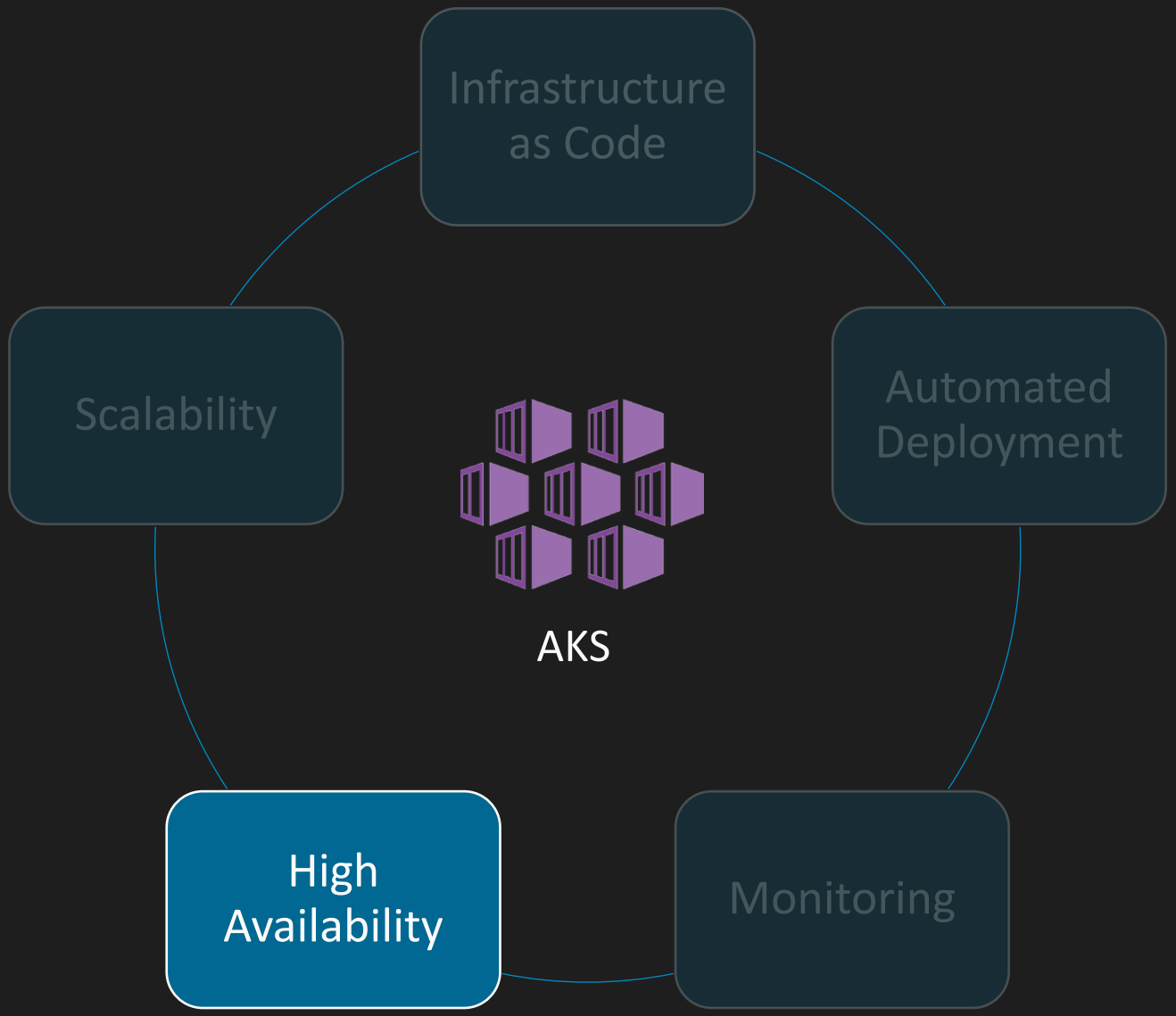
- Deploys Log Analytics agent in cluster
  - Connects to Log analytics workspace
- Collects memory and processor metrics through Kubernetes Metrics API
  - Controllers
  - Nodes
  - Containers
  - Written to metrics store
- Container logs
  - Written to logs store

Enabling Azure Monitor for AKS

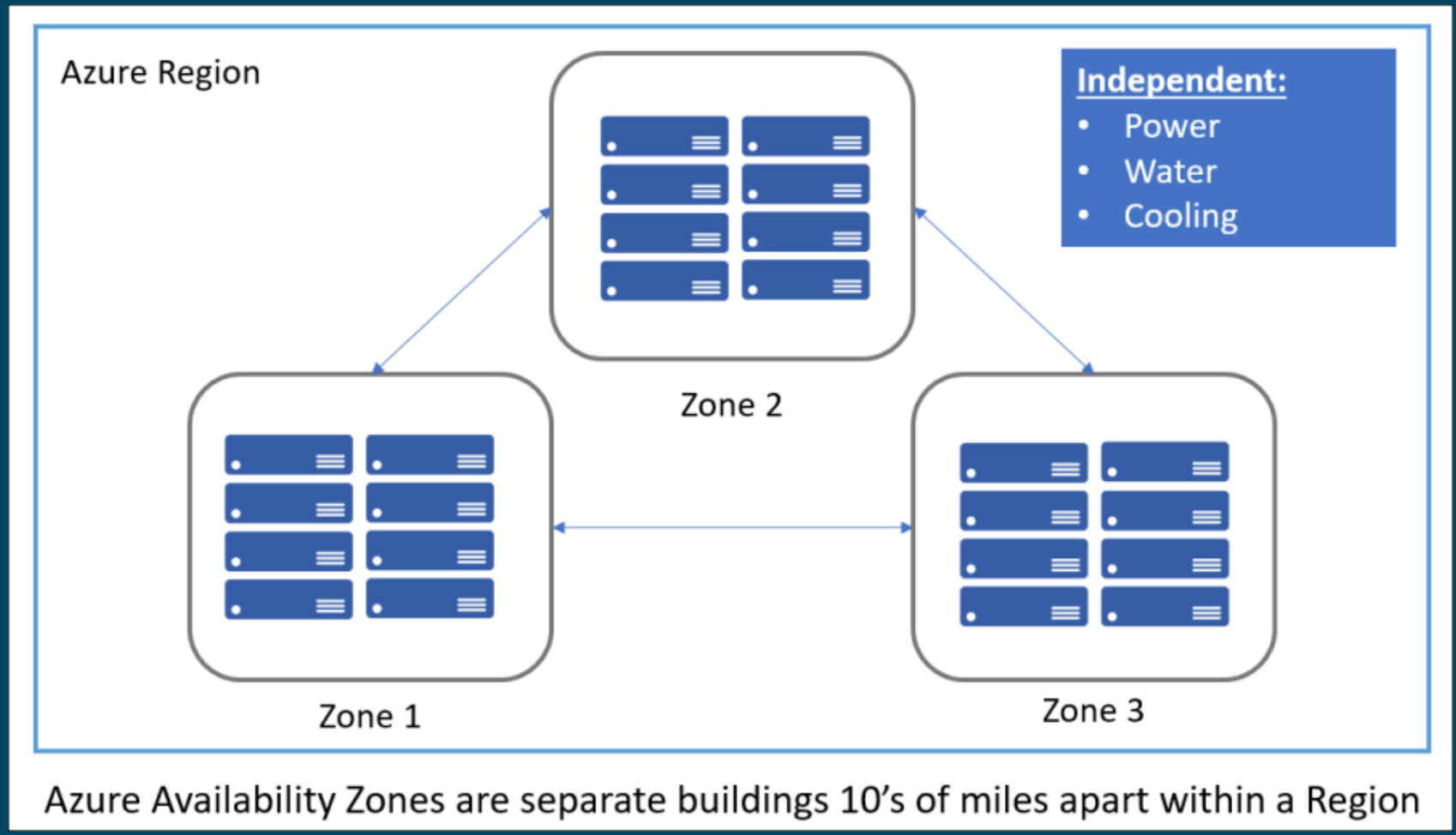
```
"resources": [
  {
    "apiVersion": "[variables('apiVersion').aks]",
    "type": "Microsoft.ContainerService/managedClusters",
    "name": "[parameters('clusterName')]",
    "location": "[resourceGroup().location]",
    "properties": {
      "nodeResourceGroup": "[concat(parameters('clusterName'), '-worker')]",
      "kubernetesVersion": "[parameters('kubernetesVersion')]",
      "enableRBAC": true,
      "dnsPrefix": "[parameters('clusterName')]",
      "addonProfiles": {
        "kubeDashboard": {
          "enabled": false
        },
        "omsagent": {
          "enabled": true,
          "config": {
            "logAnalyticsWorkspaceResourceID": "[variables('cluster').workspaceId]"
          }
        },
        "azurepolicy": {
```

Demo

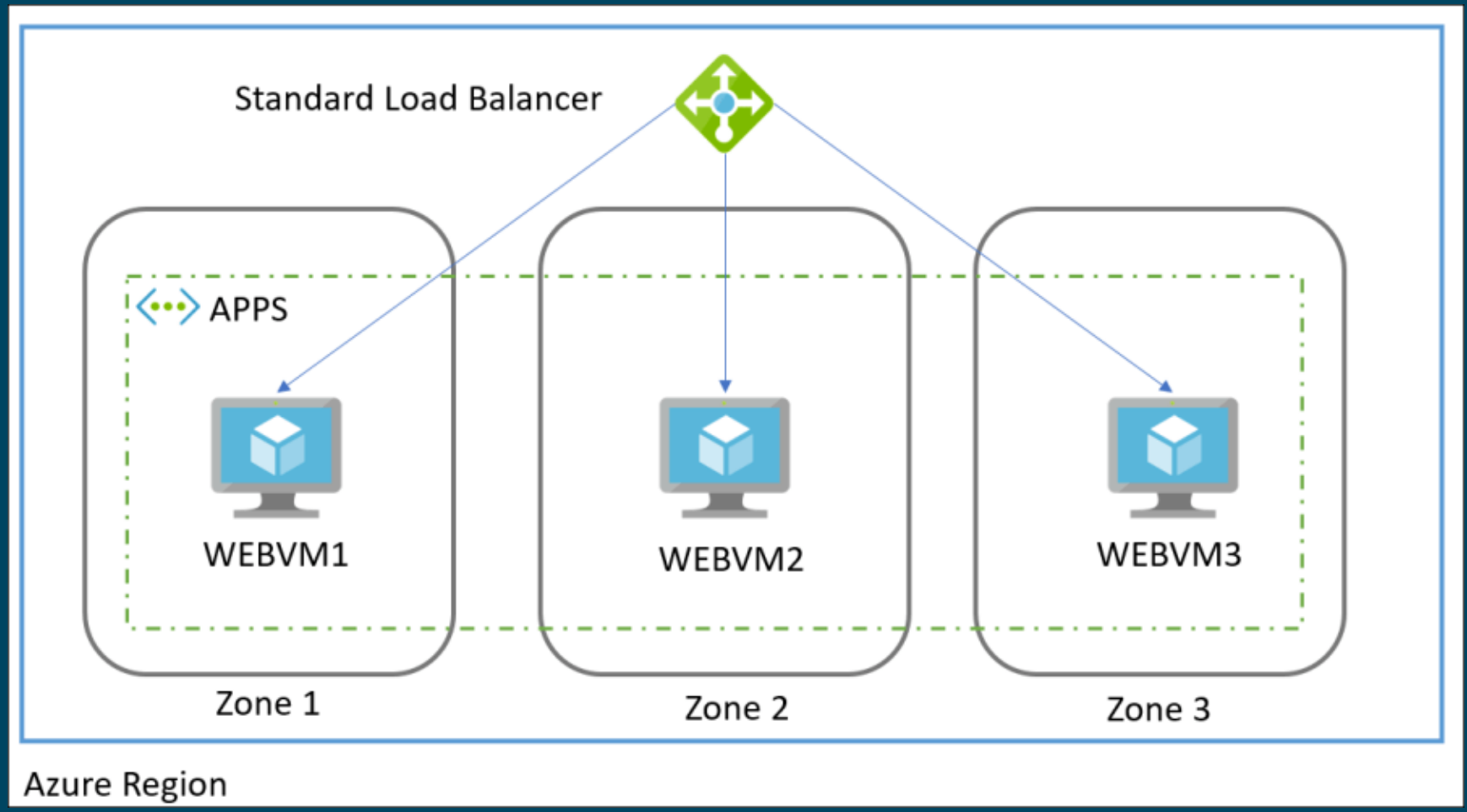
Monitoring & Feedback


Infrastructure as Code
Automated Deployment
Monitoring
High Availability
Scalability
AKS

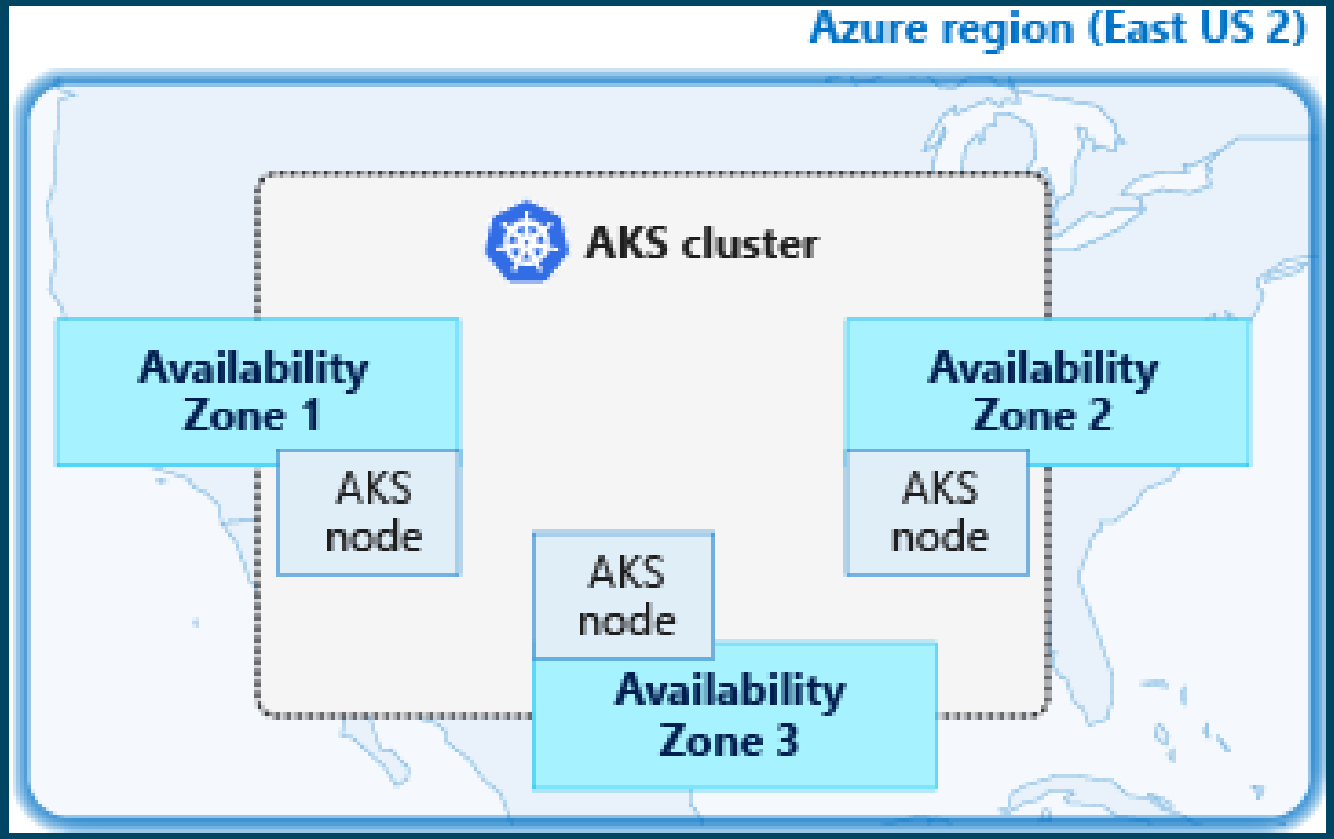
Azure Regions and Availability Zones



Azure Regions and Availability Zones



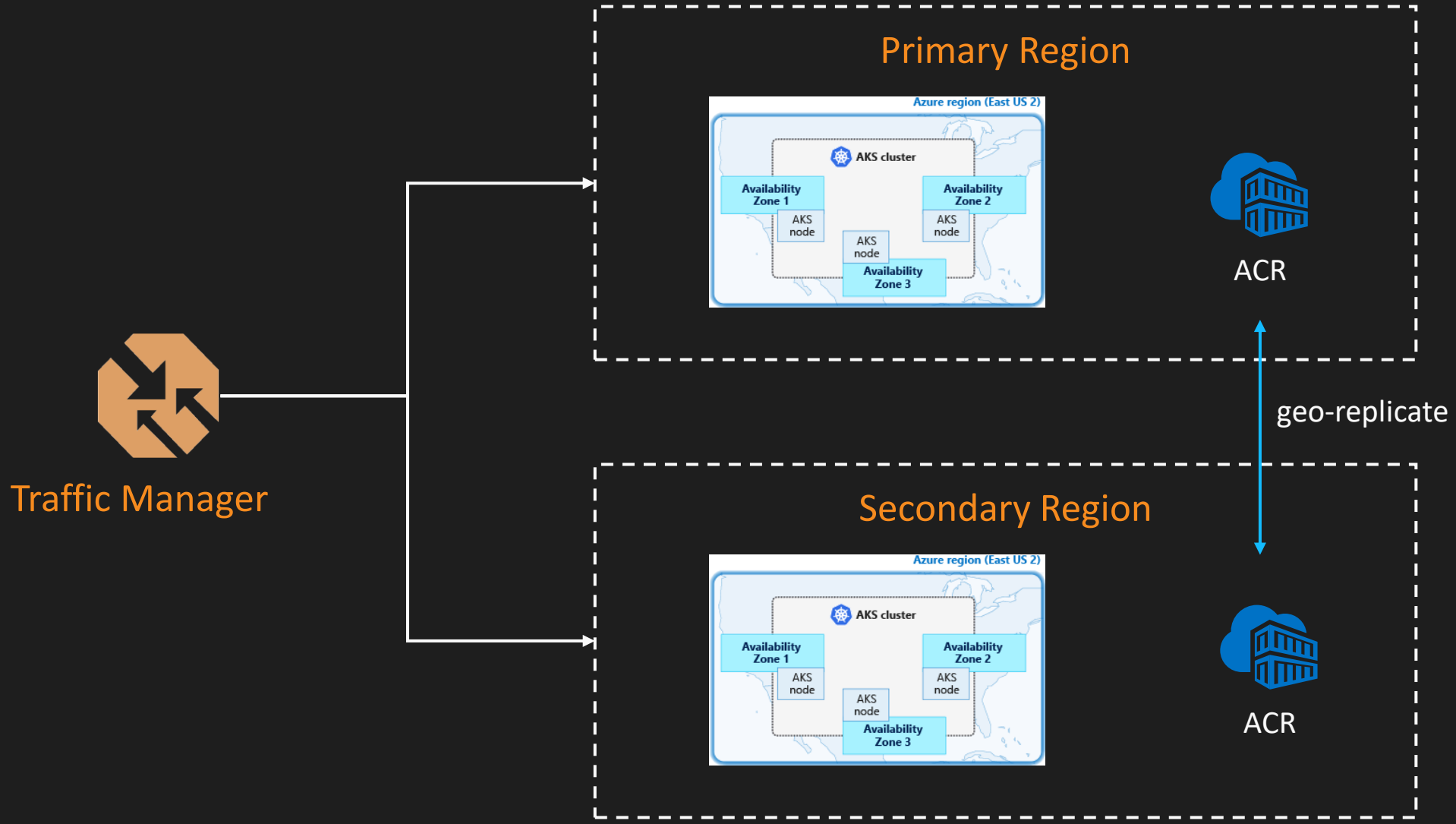
Cluster Availability and Failover

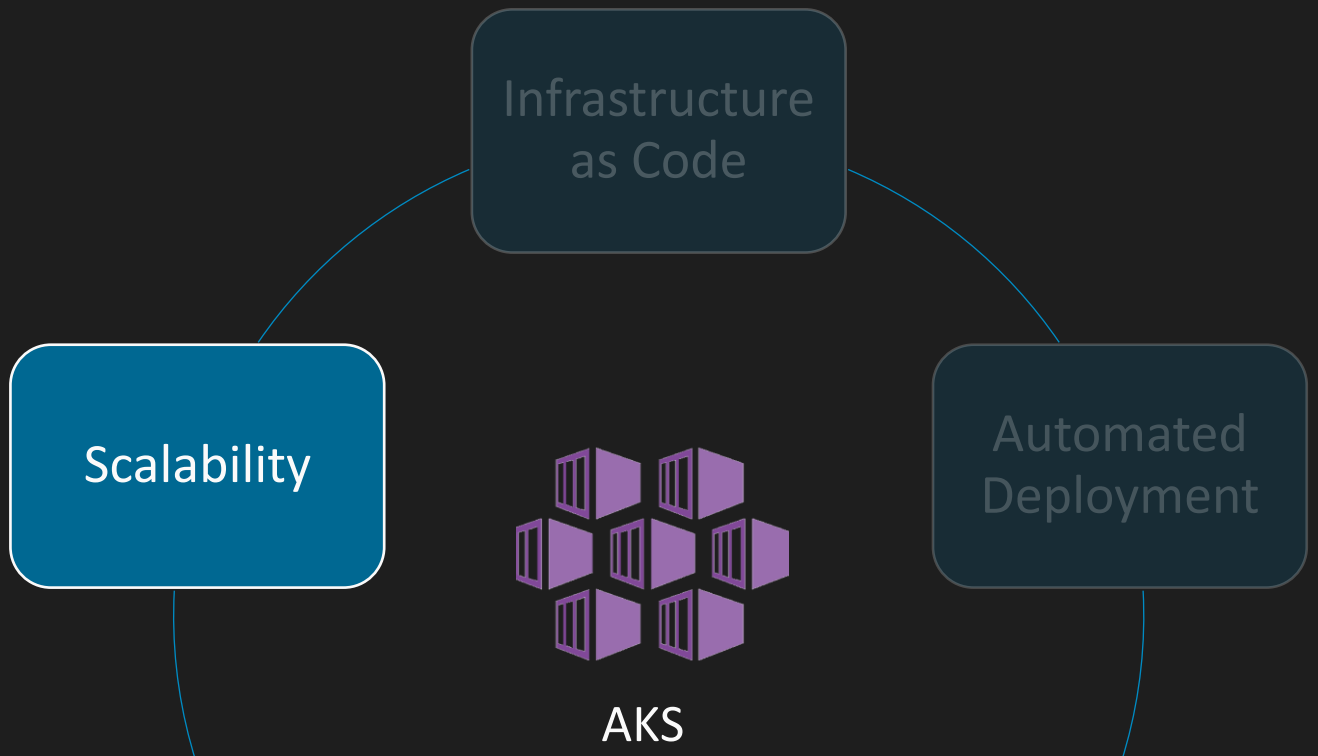


Enabling Availability Zones for AKS

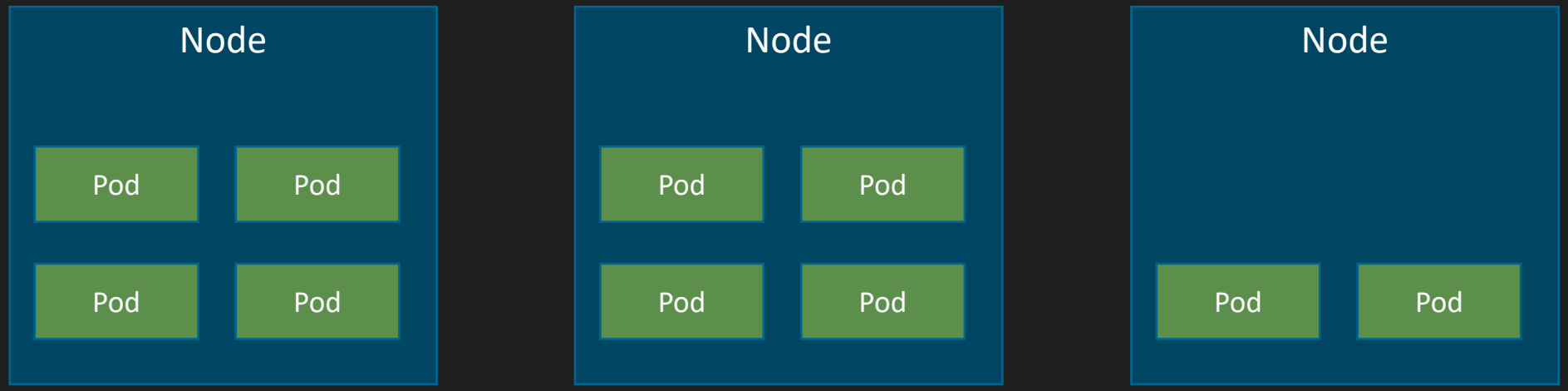
```
"copy": [
  {
    "name": "agentPoolProfiles",
    "count": "[length(parameters('agentPoolProfiles'))]",
    "input": {
      "name": "[if(equals(parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].osType, 'Lir
      "orchestratorVersion": "[parameters('kubernetesVersion')]",
      "maxPods": 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')].enable
      "maxCount": "[if(parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].enableAutoScali
      "minCount": "[if(parameters('agentPoolProfiles')[copyIndex('agentPoolProfiles')].enableAutoScali
      "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




Infrastructure as Code
Automated Deployment
Monitoring
High Availability
Scalability
AKS

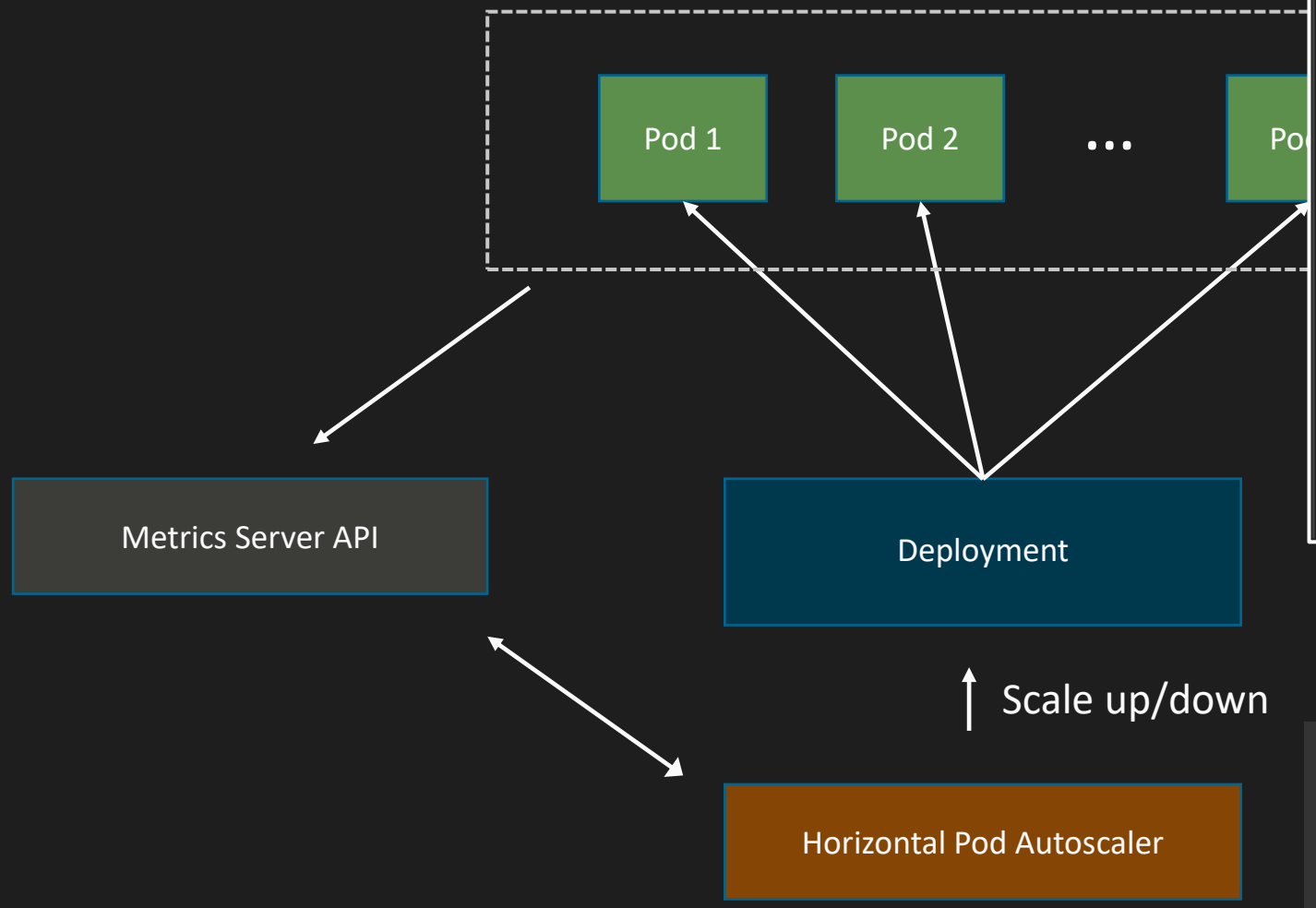
Scaling with AKS



Horizontal Pod Autoscaling (HPA)

Cluster Autoscaler

Horizontal Pod Autoscaler



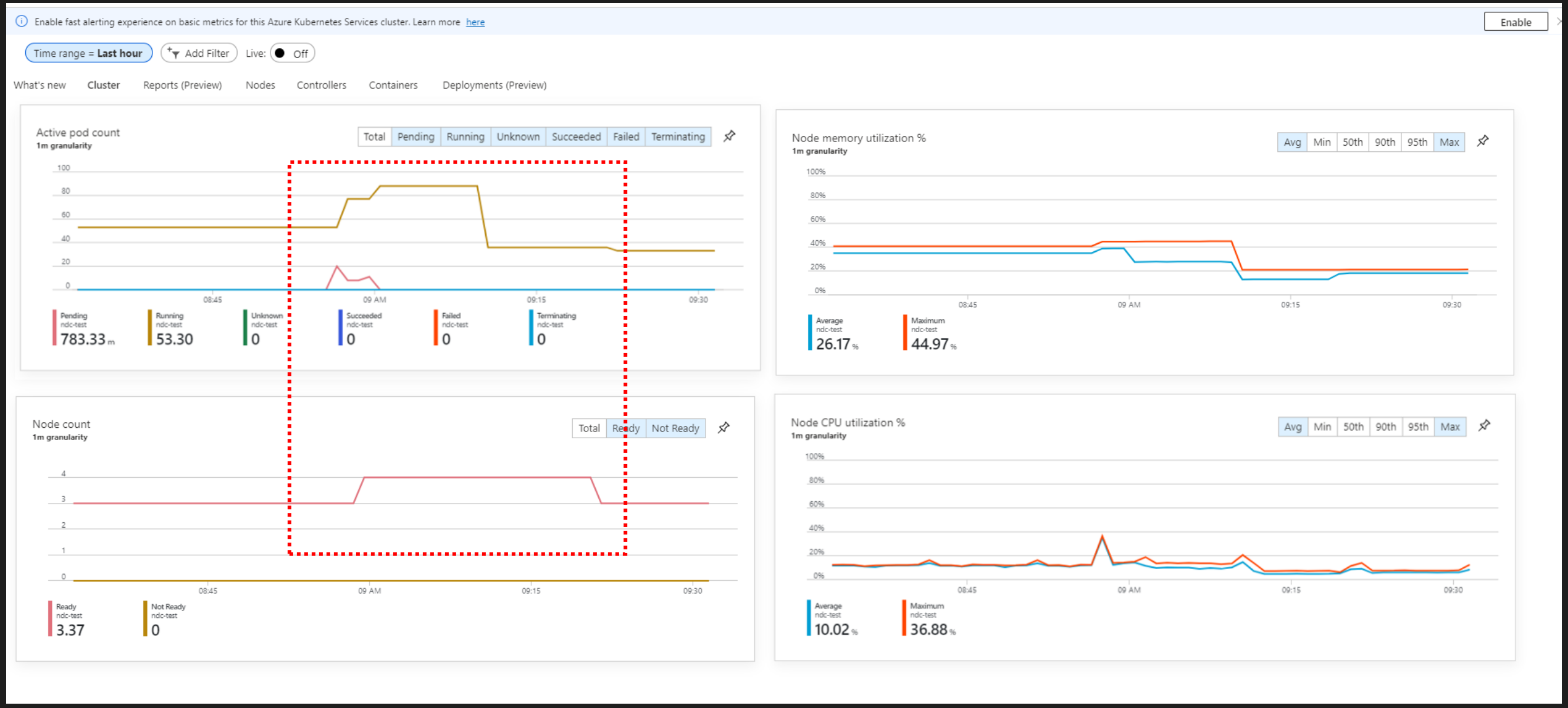

```
      app: qbox
      tier: frontend
  spec:
    containers:
    - name: frontend
      image: qbox.web:1.0
      ports:
      - containerPort: 80
      resources:
        requests:
          cpu: 250m
        limits:
          cpu: 500m
```

```
kubectl autoscale deployment xyz
  --cpu-percent=80
  --min=3
  --max=10
```

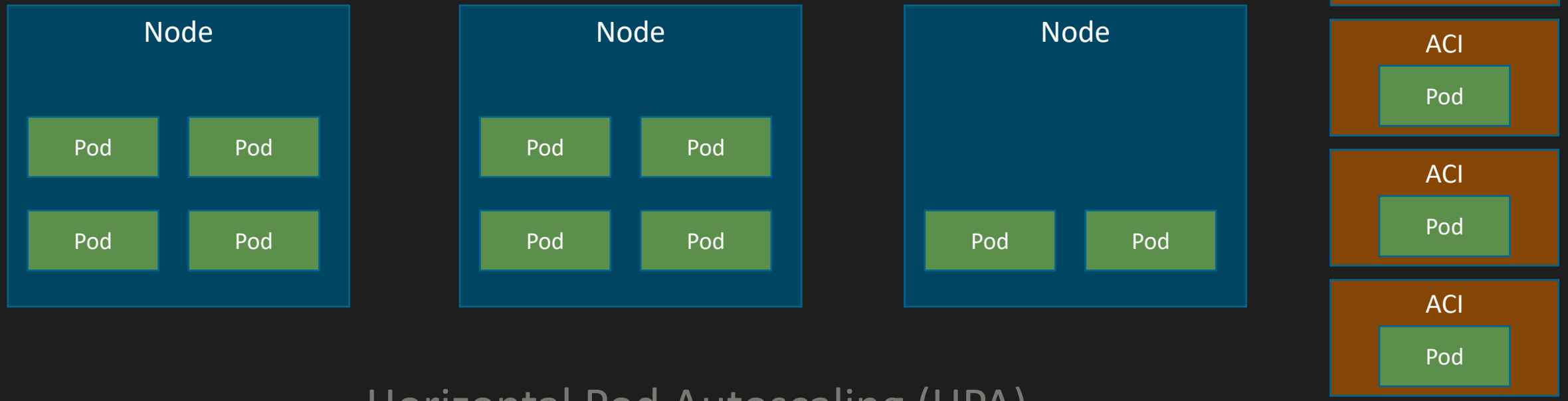
Cluster Autoscaling

```
"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



Serverless scaling with AKS

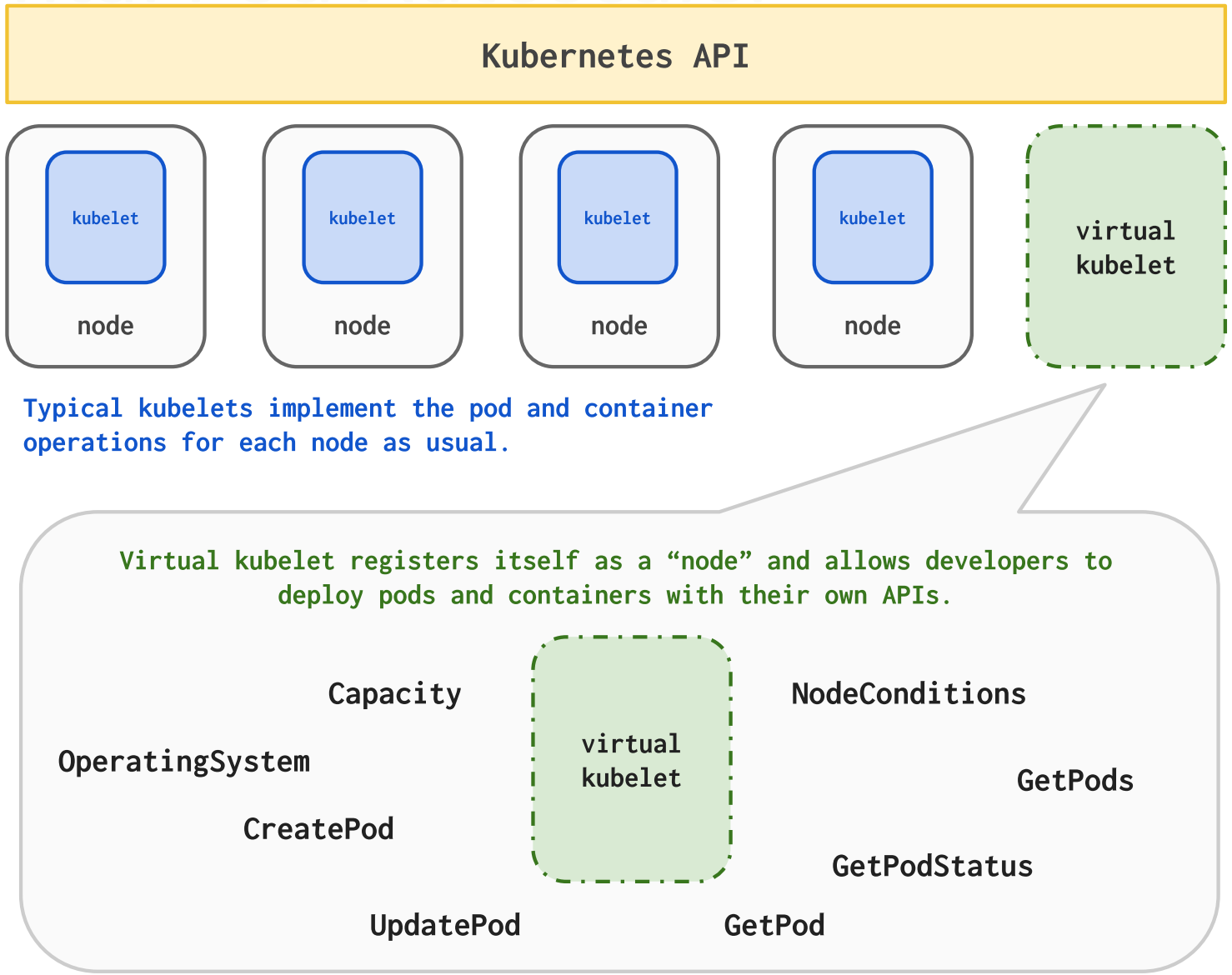


Horizontal Pod Autoscaling (HPA)

Cluster Autoscaler

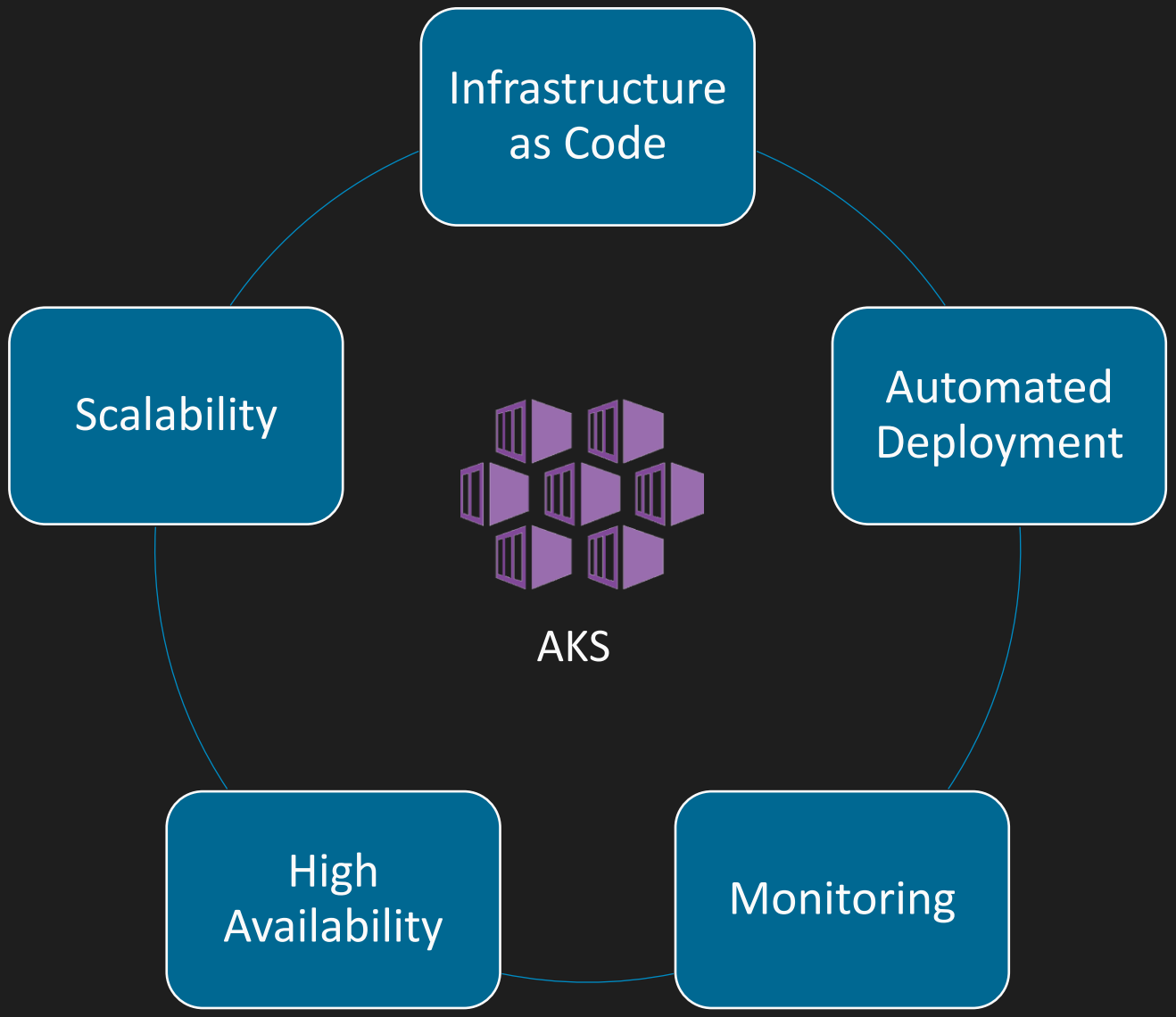
Virtual Nodes

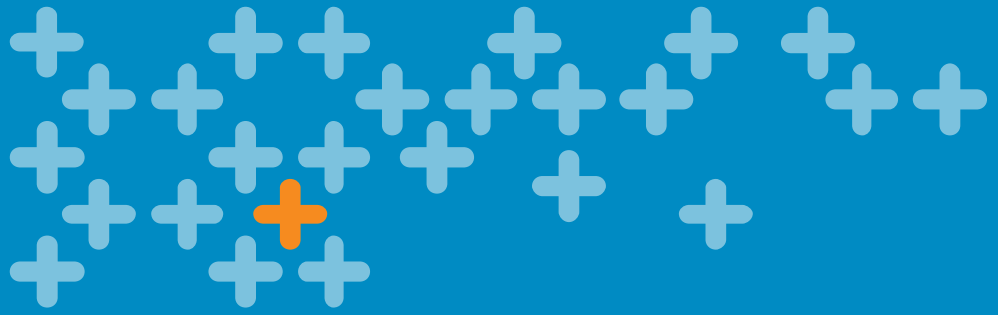
Virtual Node Architecture



Demo

High Availability
Autoscaling


Infrastructure as Code
Automated Deployment
Monitoring
High Availability
Scalability
AKS



Thank you

Jakob Ehn

@jakobehn

https://blog.ehn.nu