# Azure-AKS Concepts

# Azure Services Integrated with AKS

Azure Disks

Azure Public IP

Azure AD

Azure Container Registry

Azure Kubernetes Service


Azure MySQL

Standard Load Balancer

AD Groups

Azure DevOps

AKS Linux & Windows Nodepools

Azure Files

Virtual Network

AD Users

Azure Build & Release Pipelines

Azure VM ScaleSets

Storage Account

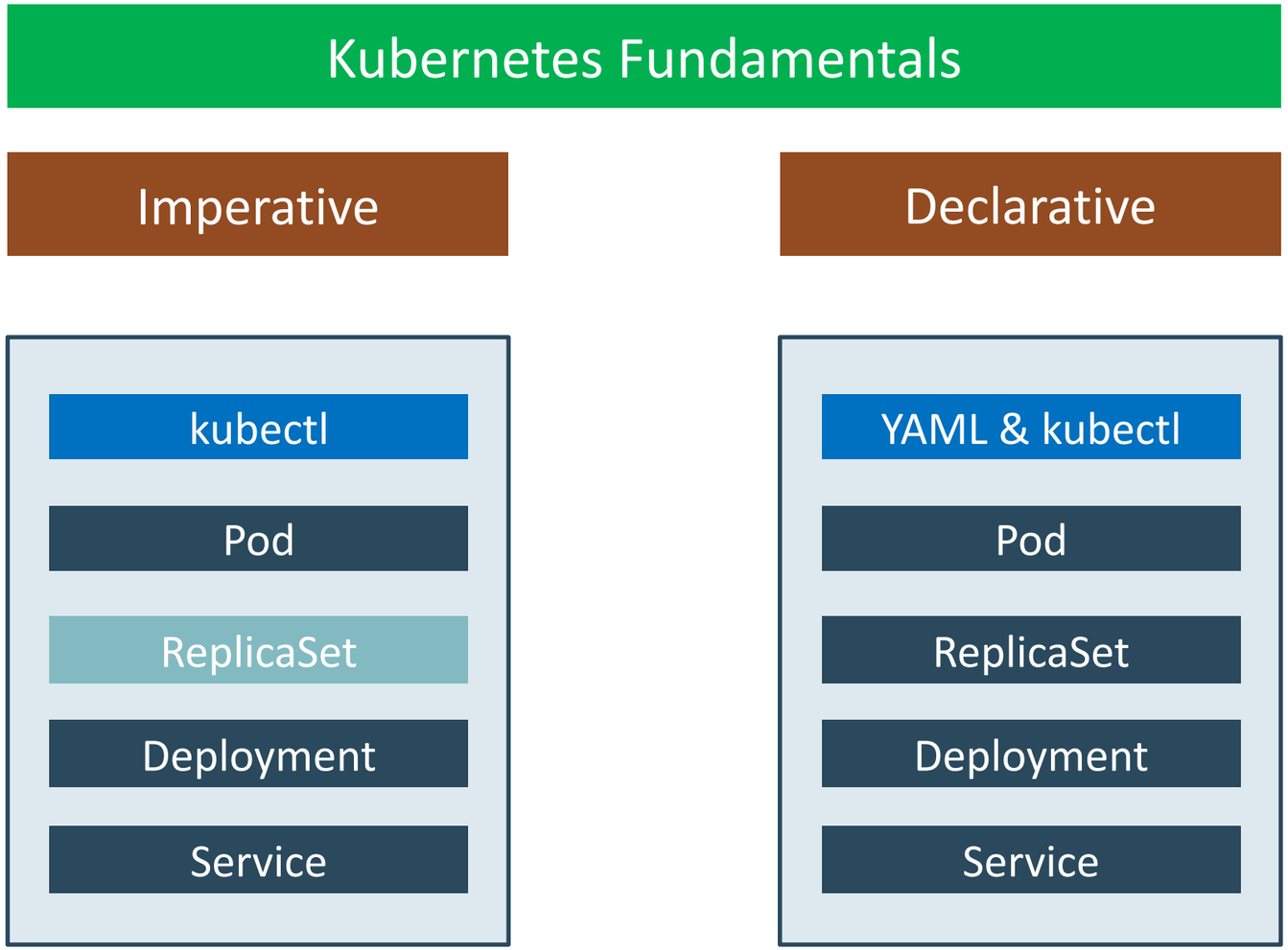
Subnets

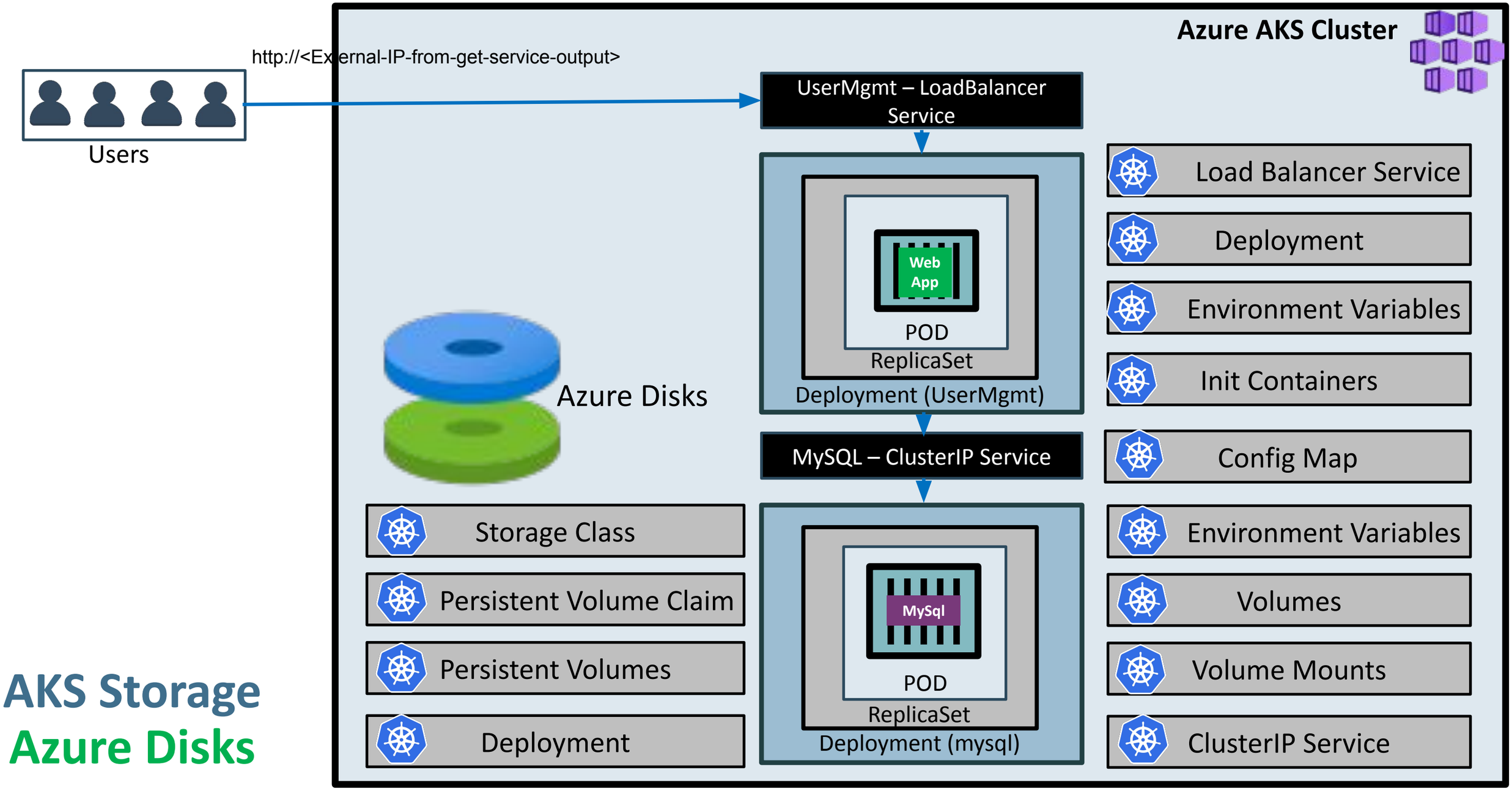

Azure DNS Zones

Azure Container Instances Virtual Nodes

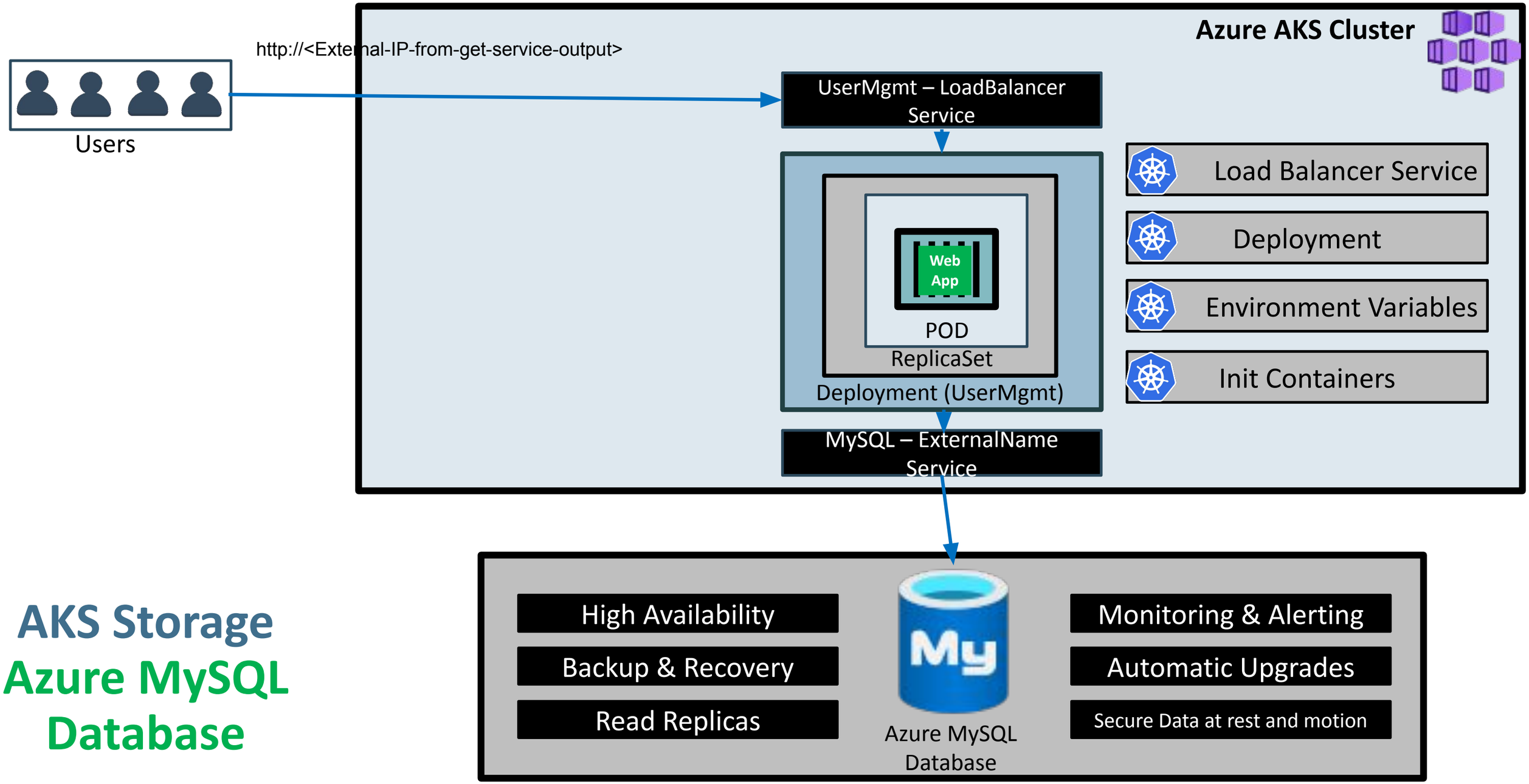
Azure Managed Service Identity

# Kubernetes - Imperative & Declarative



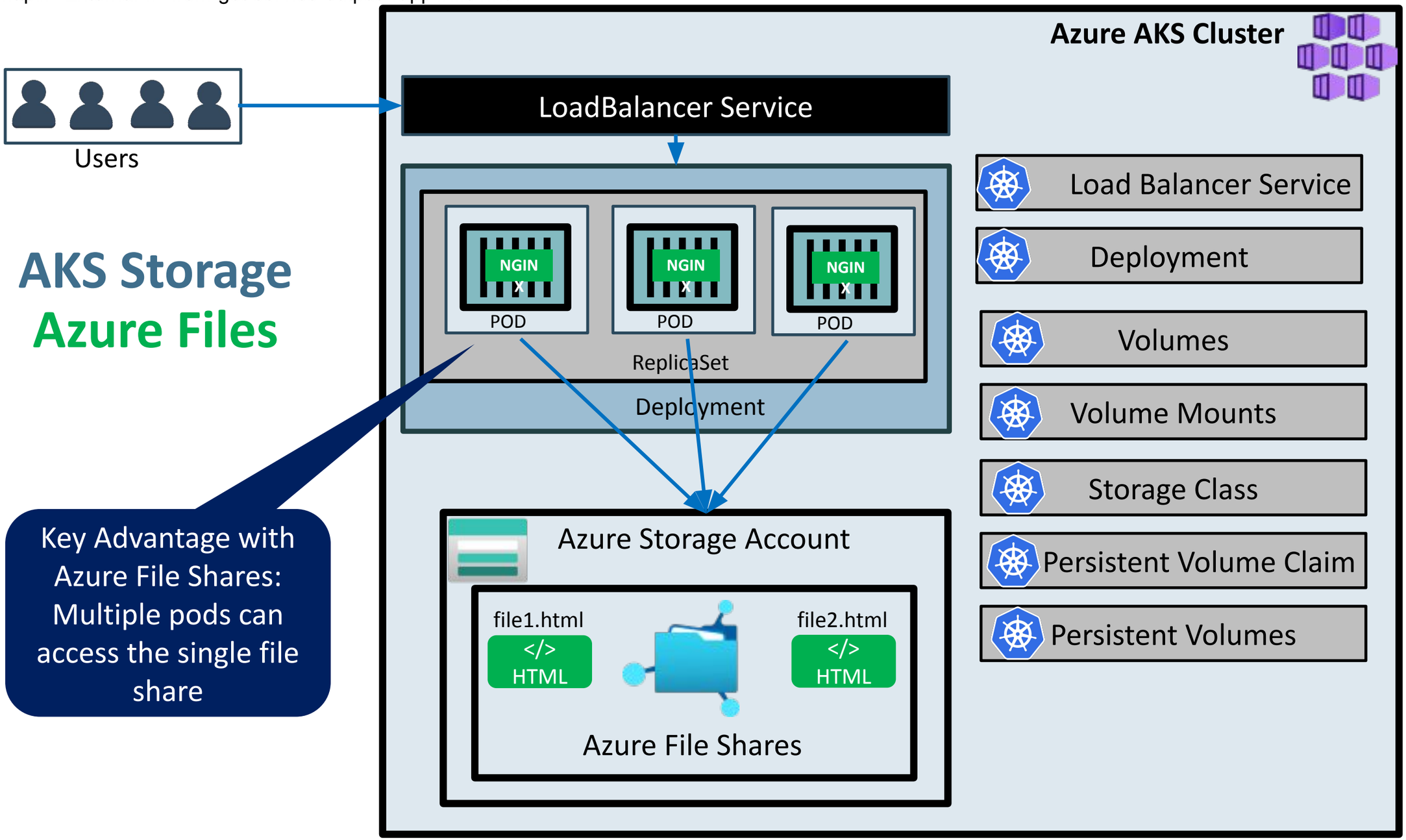



# AKS Storage
# Azure Disks




# AKS Storage
# Azure MySQL Database

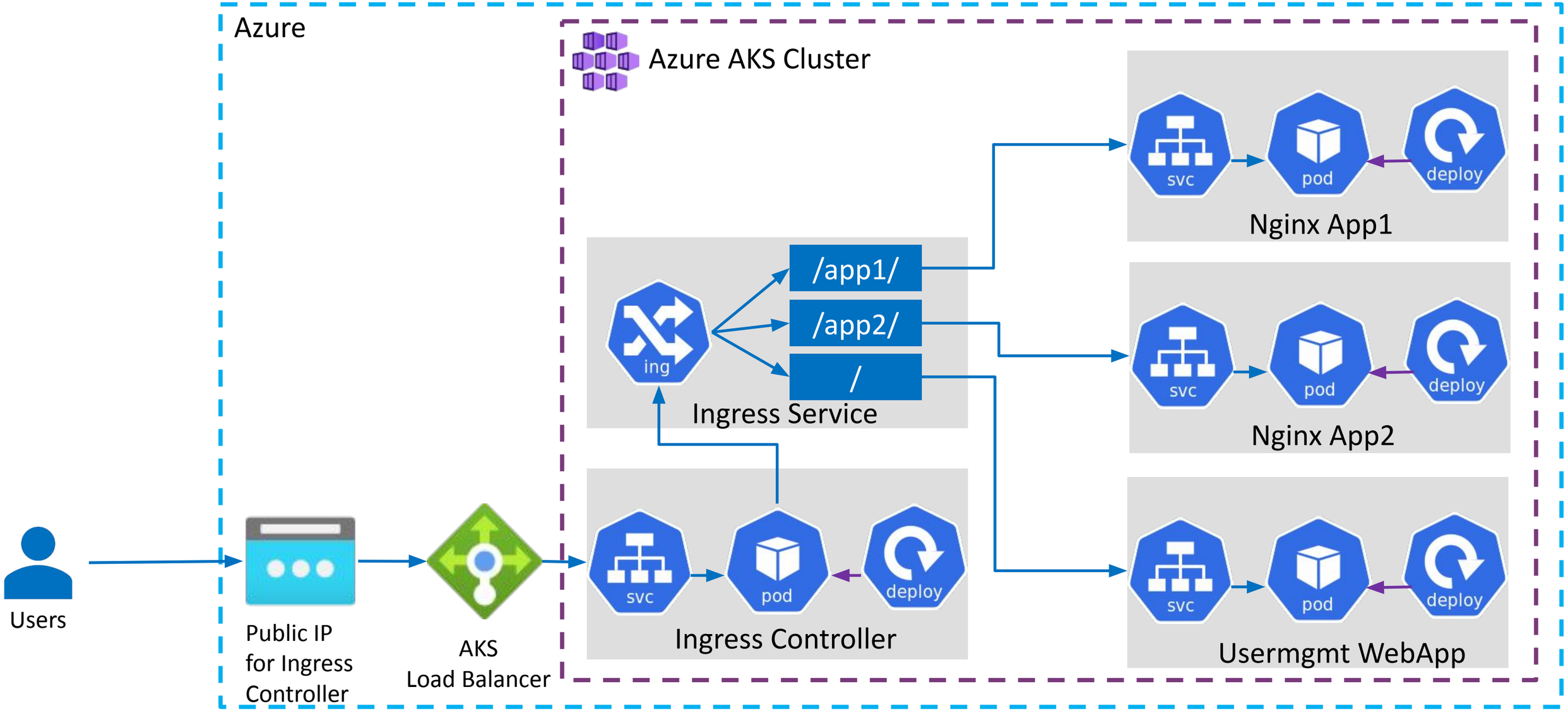
http://<External-IP-from-get-service-output>/app1/file1.html



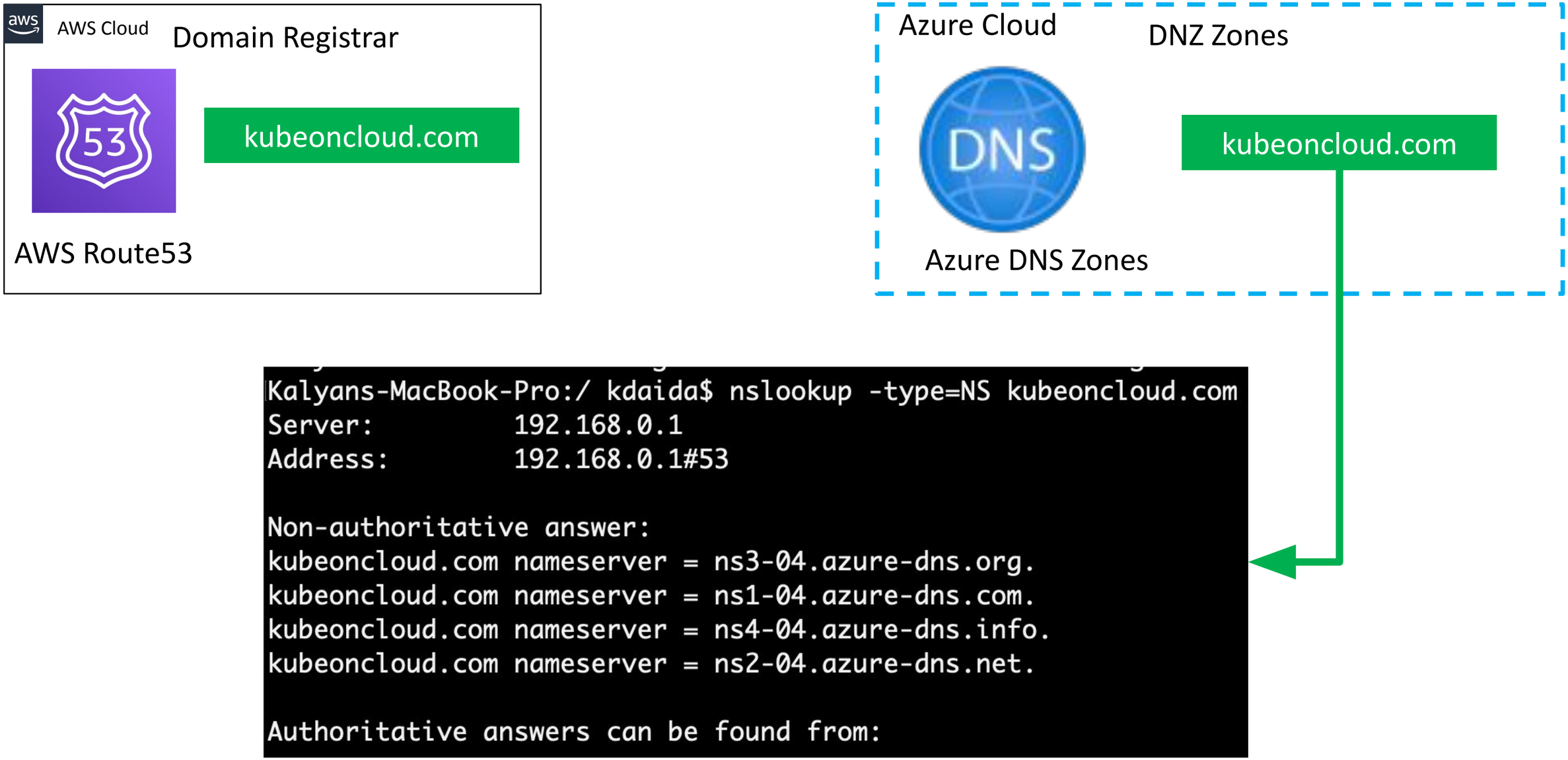

# AKS Storage
# Azure Files

Key Advantage with Azure File Shares: Multiple pods can access the single file share

# Azure AKS & Nginx Ingress – Context Path Based Routing

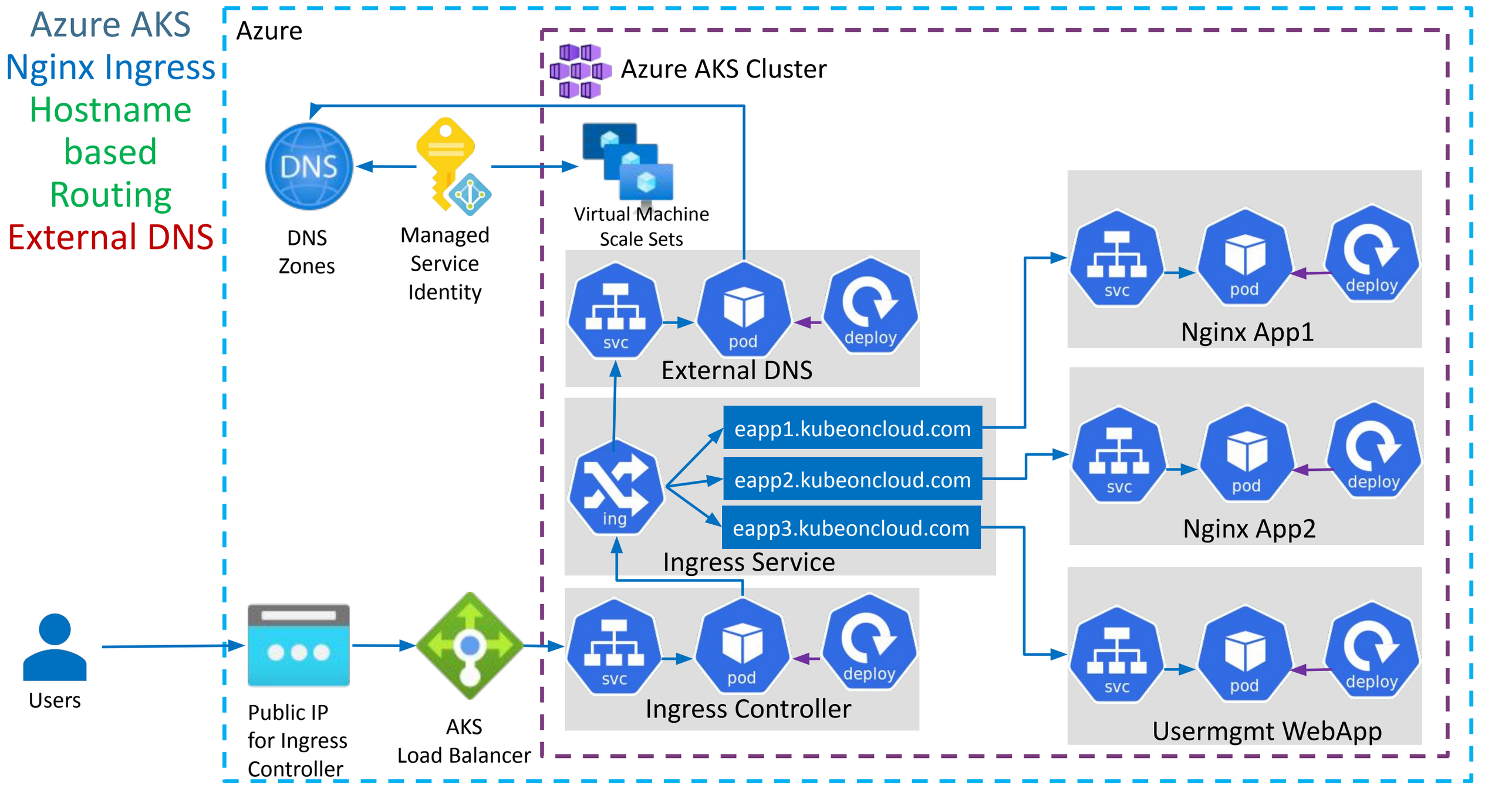


# Delegate Domain to Azure DNS

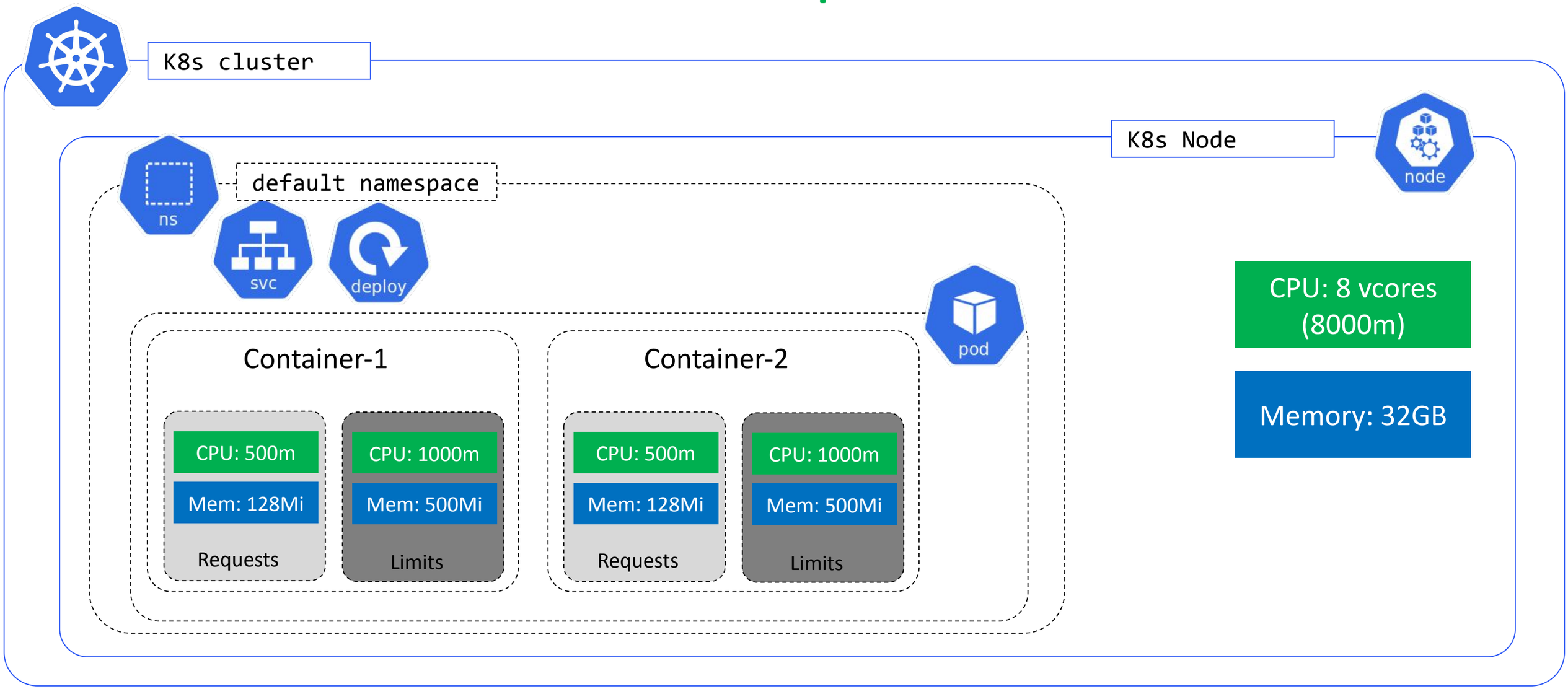



```
Kalyans-MacBook-Pro:/ kdaida$ nslookup -type=NS kubeoncloud.com
Server:         192.168.0.1
Address:        192.168.0.1#53

Non-authoritative answer:
kubeoncloud.com nameserver = ns3-04.azure-dns.org.
kubeoncloud.com nameserver = ns1-04.azure-dns.com.
kubeoncloud.com nameserver = ns4-04.azure-dns.info.
kubeoncloud.com nameserver = ns2-04.azure-dns.net.

Authoritative answers can be found from:
```

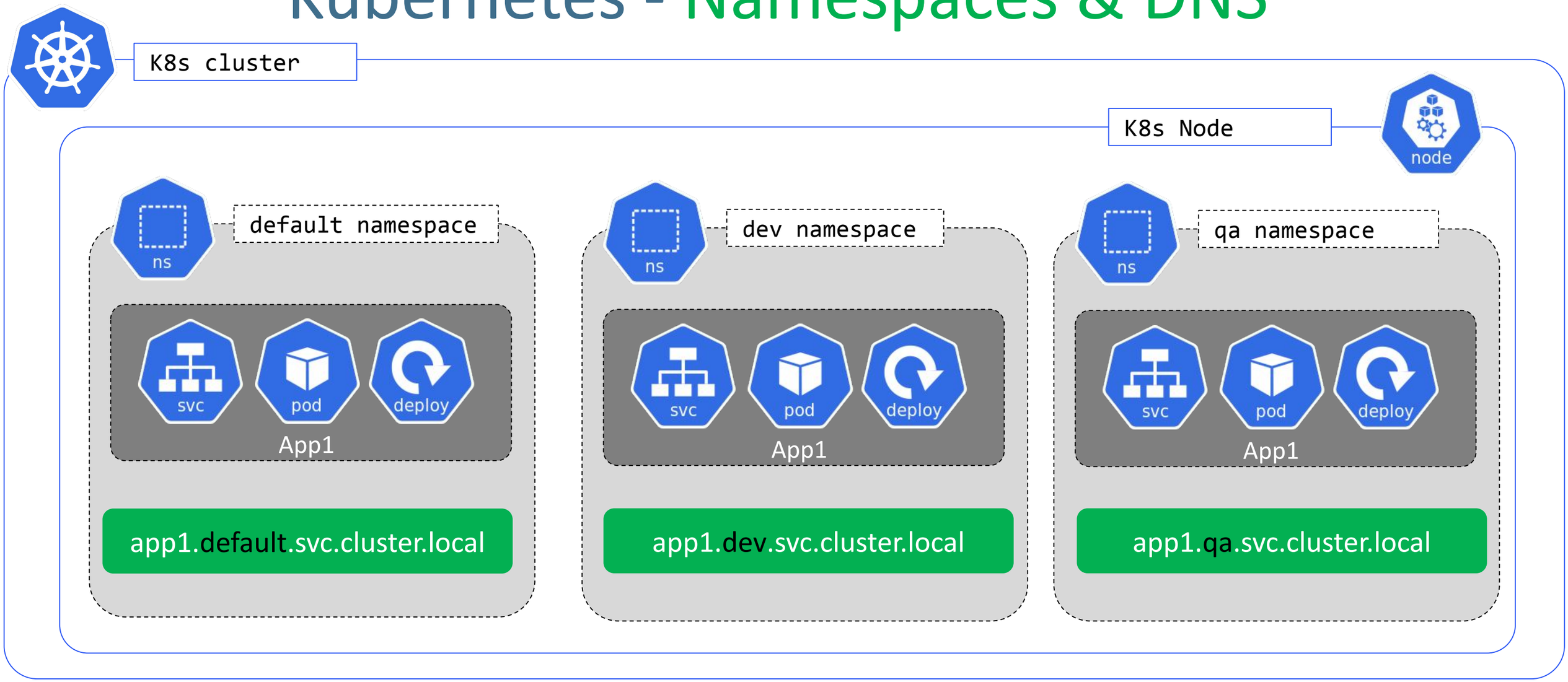
# Azure AKS Nginx Ingress Hostname based Routing External DNS



# Kubernetes – Requests & Limits

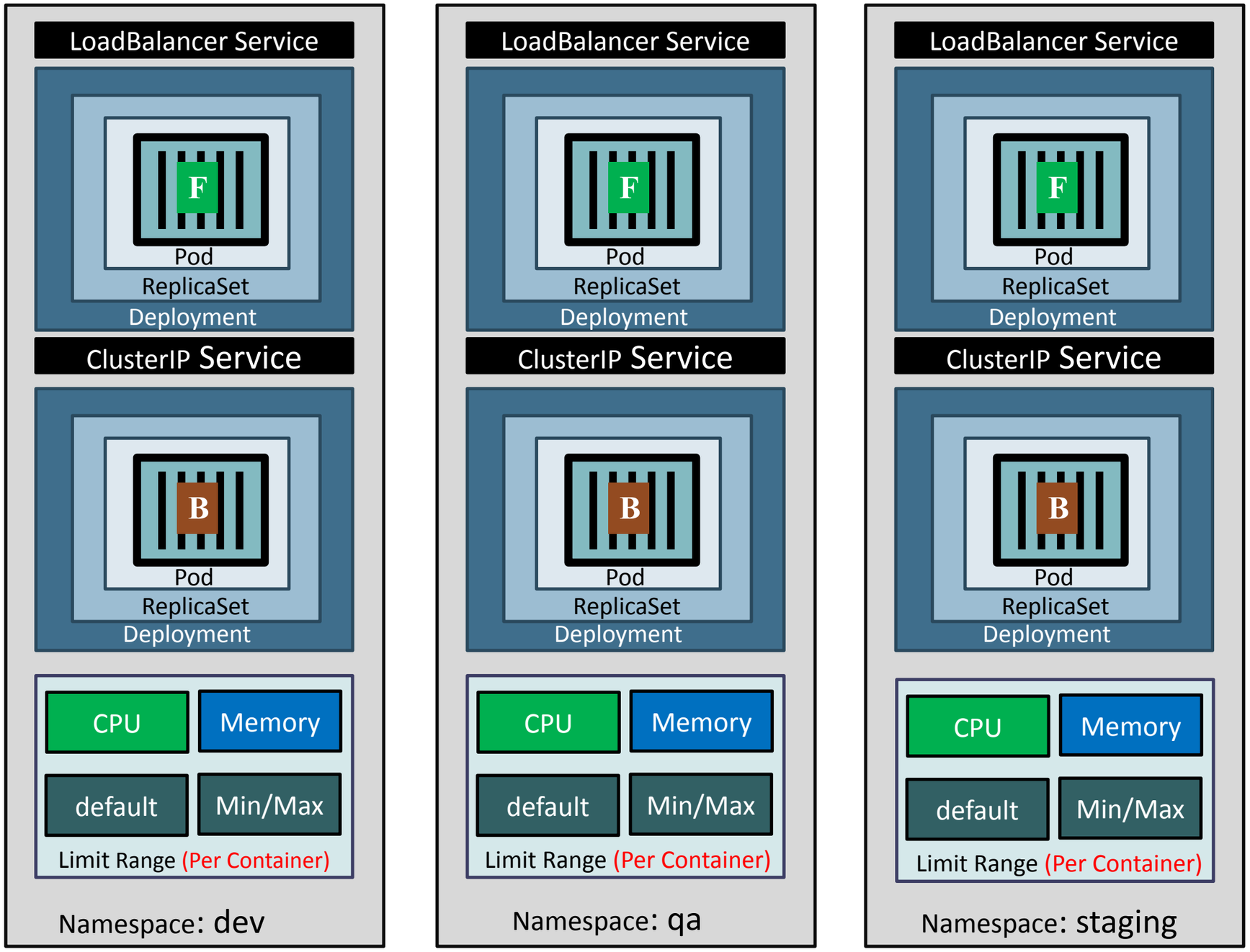


# Kubernetes - Namespaces & DNS




<service-name>.<namespace-name>.svc.cluster.local

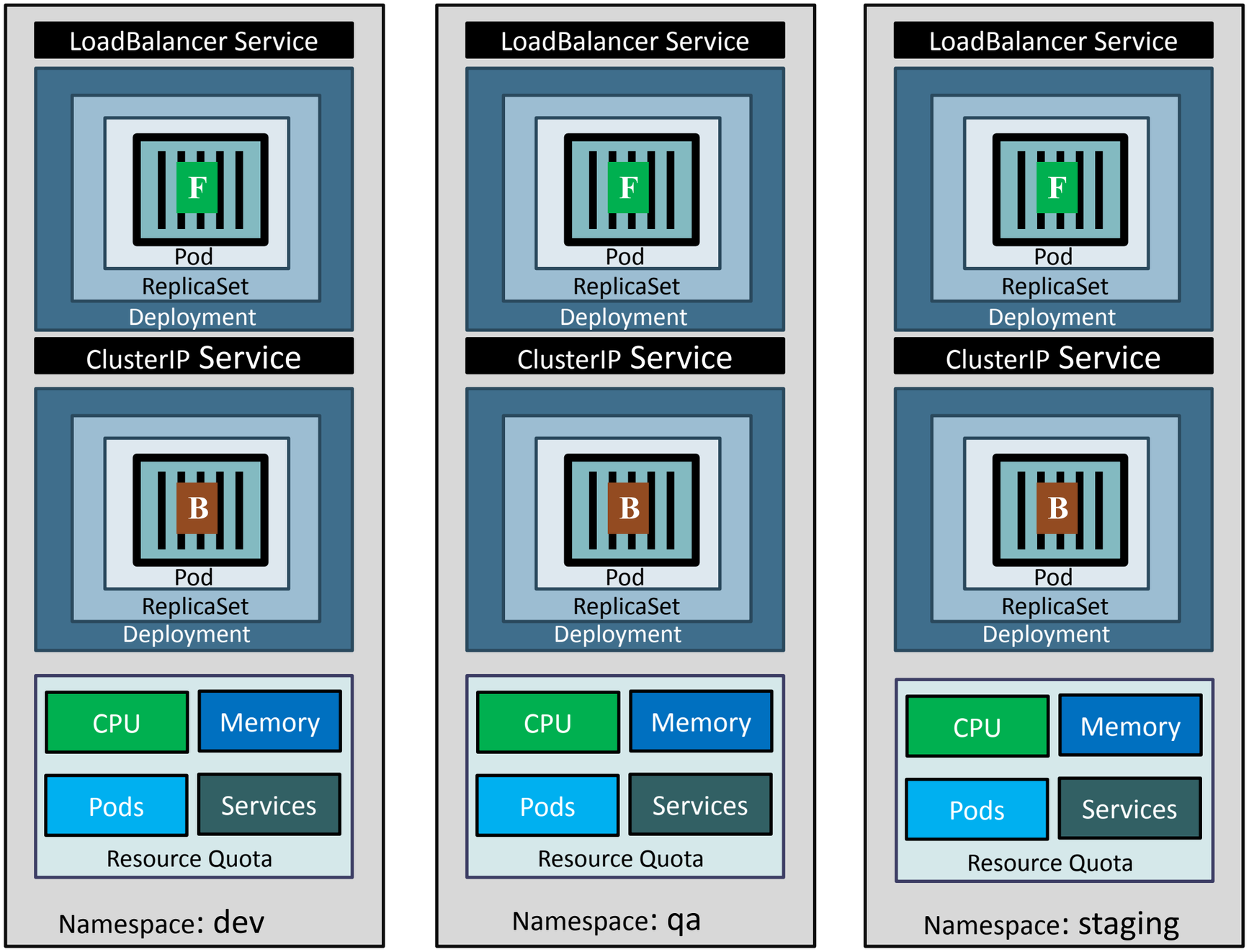
# Limit Range




**Limit Range Manifest**

```yaml
apiVersion: v1
kind: LimitRange
metadata:
  name: default-cpu-mem-limit-range
  namespace: dev3
spec:
  limits:
    - default:
        memory: "512Mi"
        cpu: "500m"
      defaultRequest:
        memory: "256Mi"
        cpu: "300m"
      type: Container
```

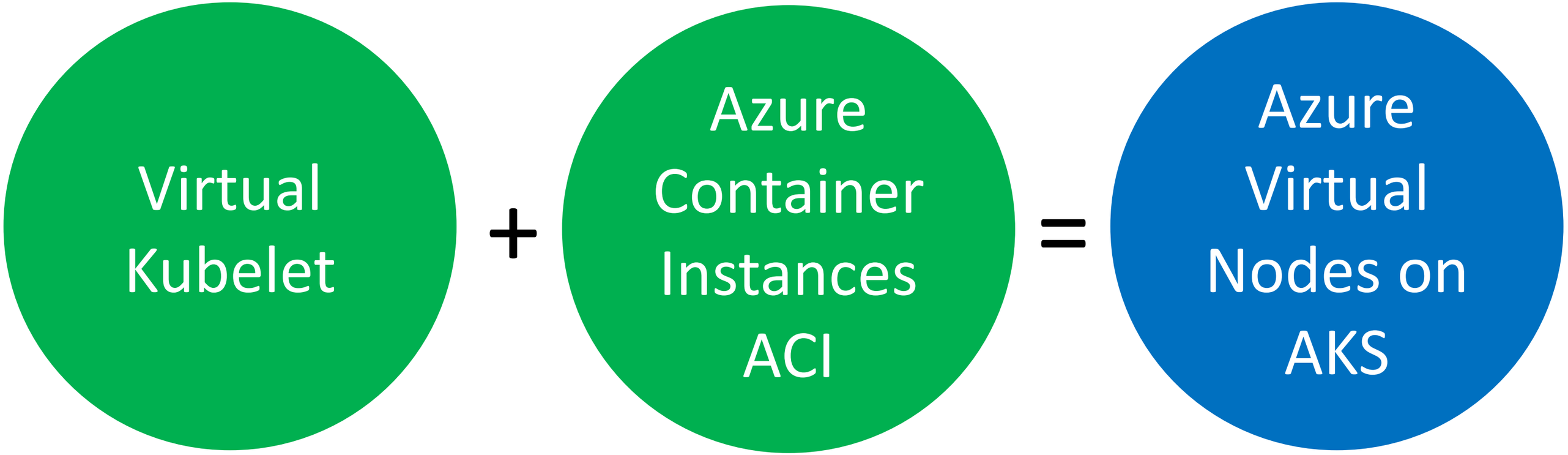
# Resource Quota



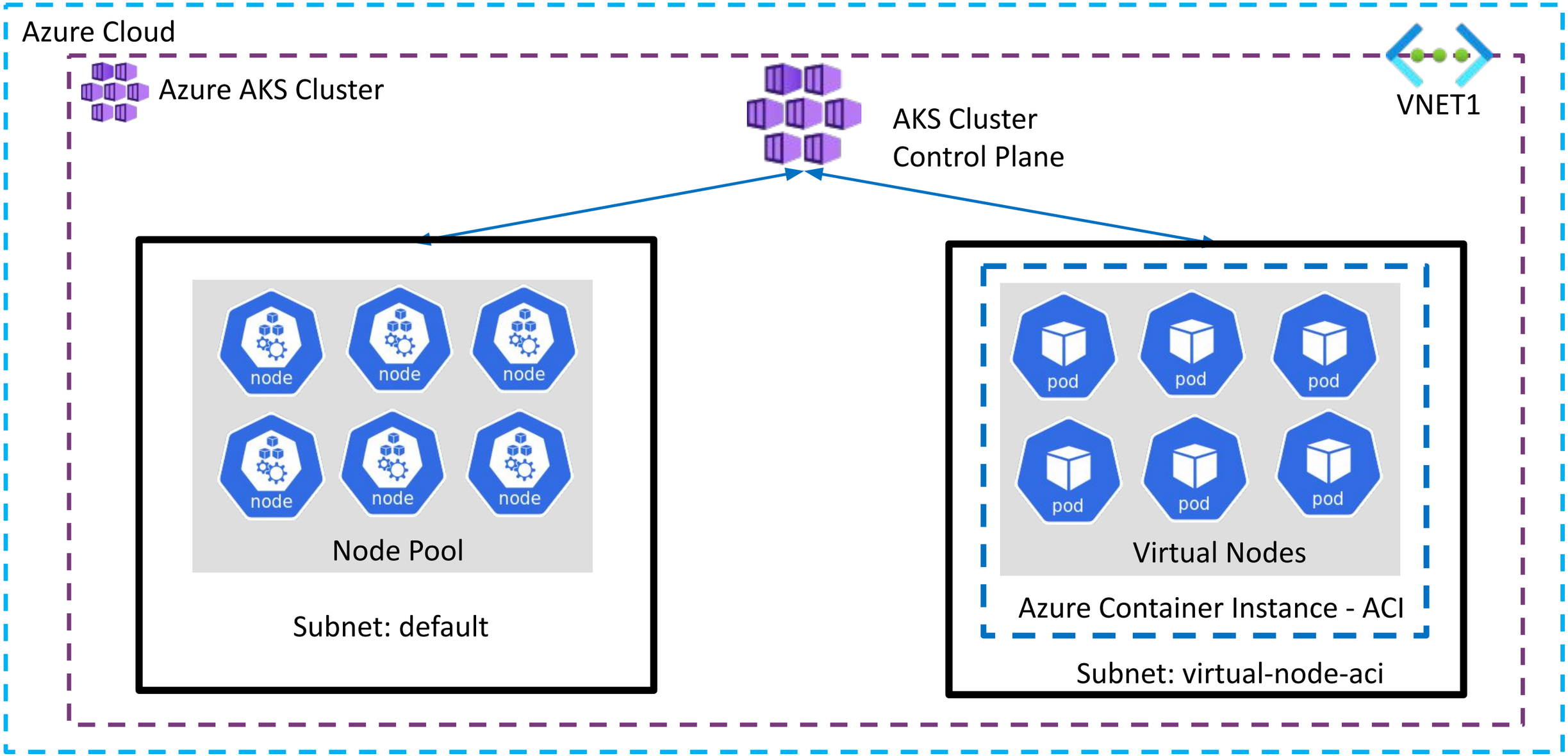

**Resource Quota Manifest**

```yaml
apiVersion: v1
kind: ResourceQuota
metadata:
  name: ns-resource-quota
  namespace: dev3
spec:
  hard:
    requests.cpu: "1"
    requests.memory: 1Gi
    limits.cpu: "2"
    limits.memory: 2Gi
    pods: "5"
    configmaps: "5"
    persistentvolumeclaims: "5"
    secrets: "5"
    services: "5"
```

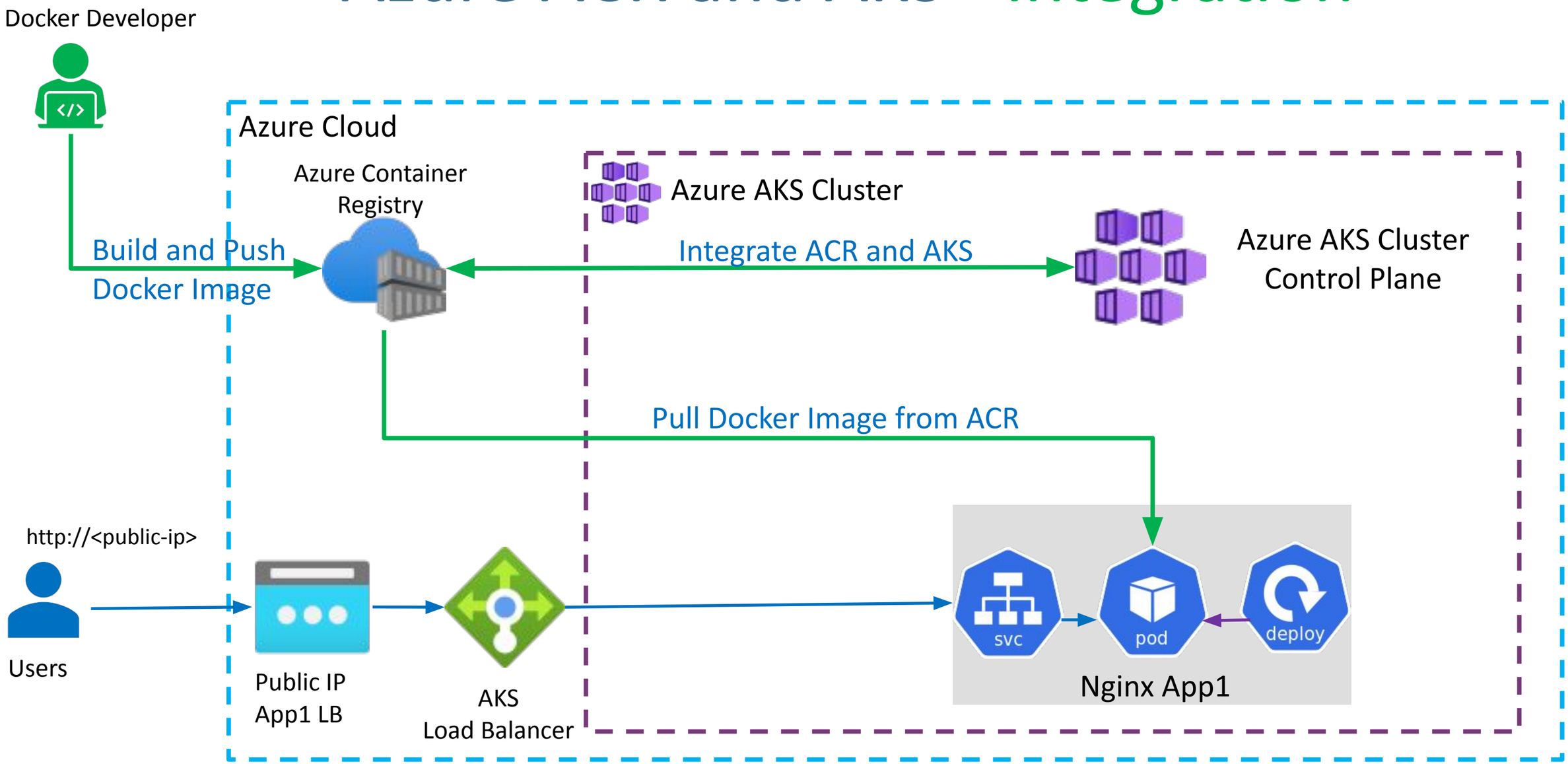
# Azure AKS Virtual Nodes



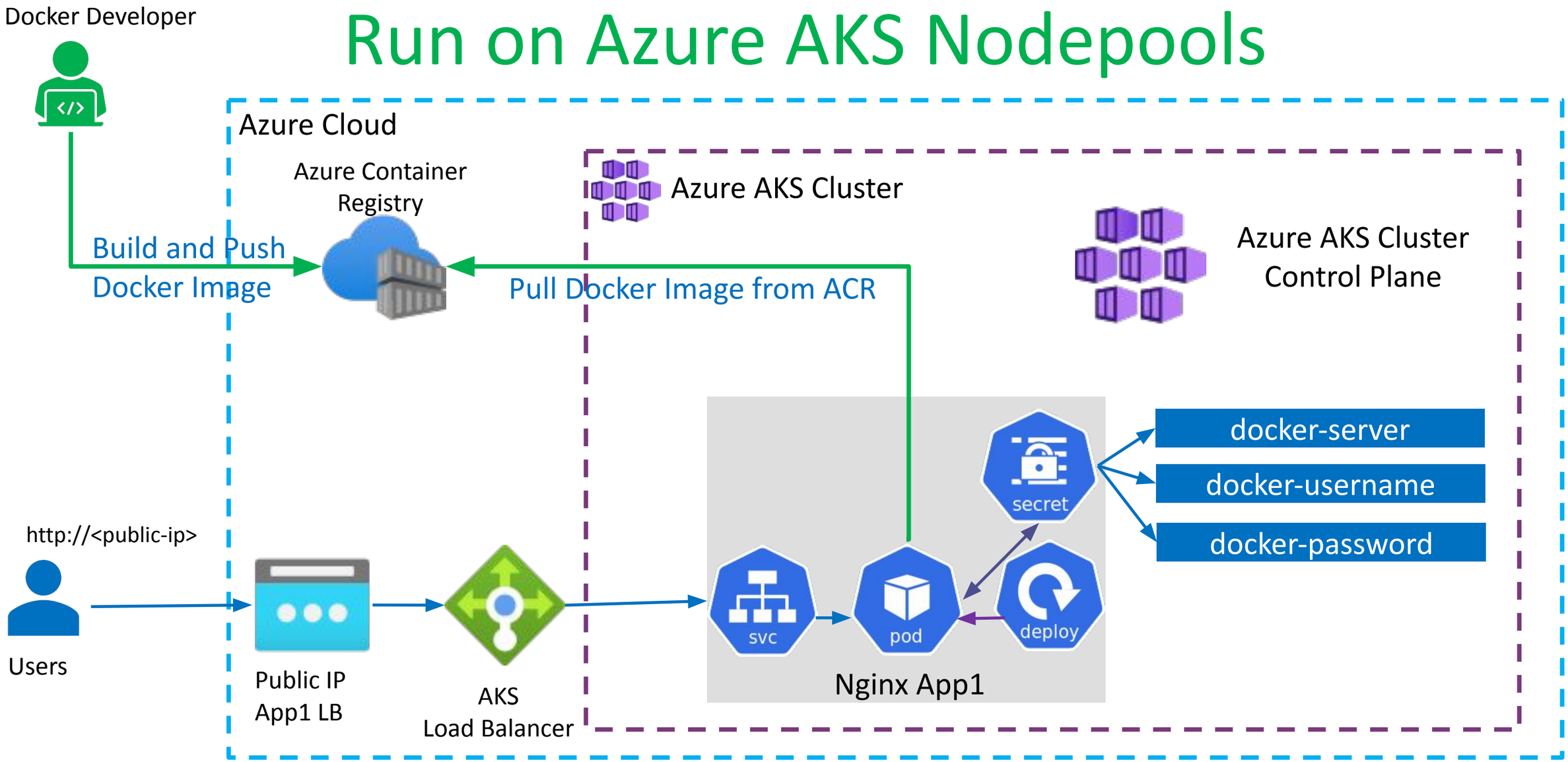
# Azure AKS - Virtual Nodes



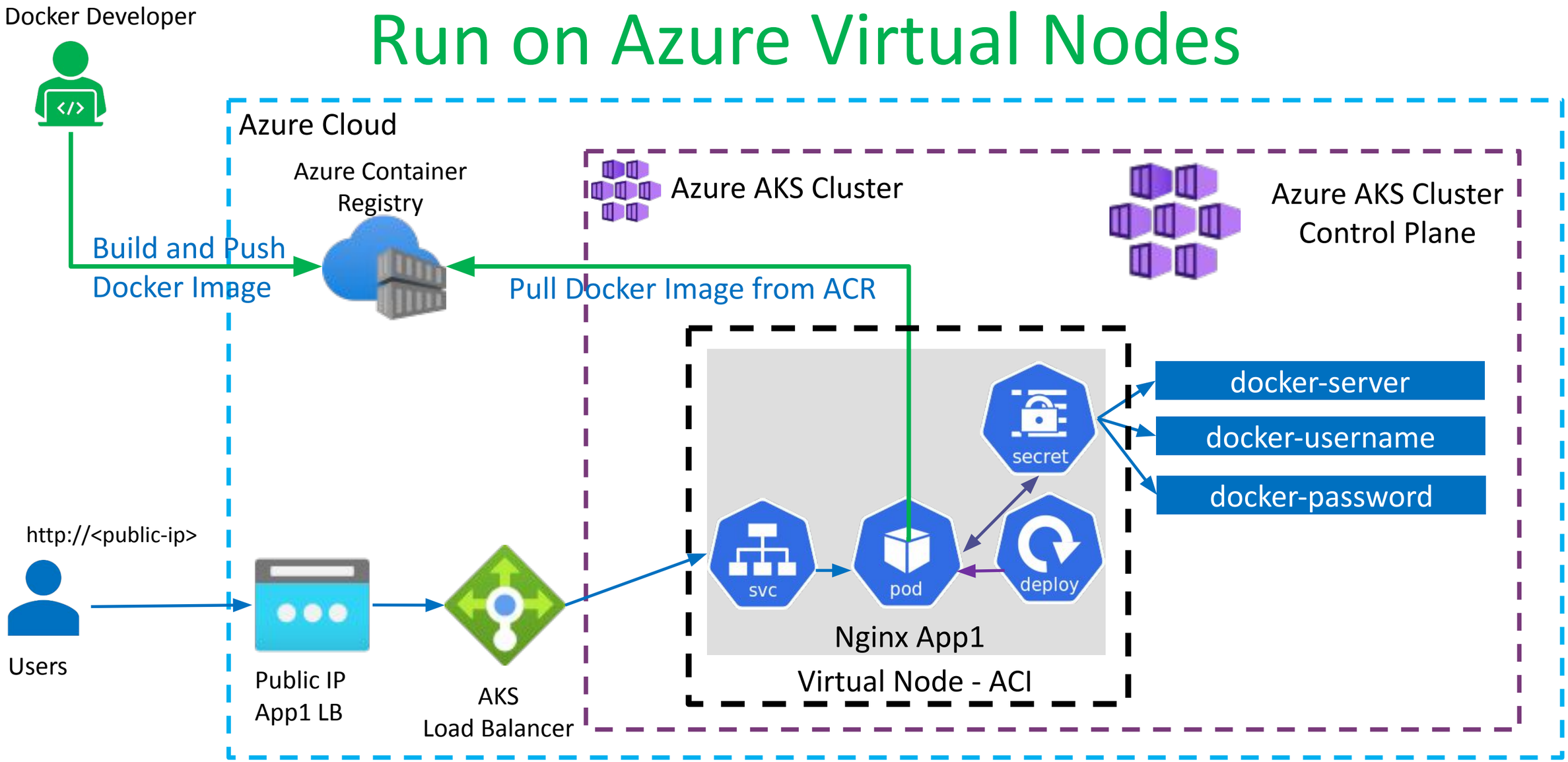
# Azure ACR and AKS - Integration



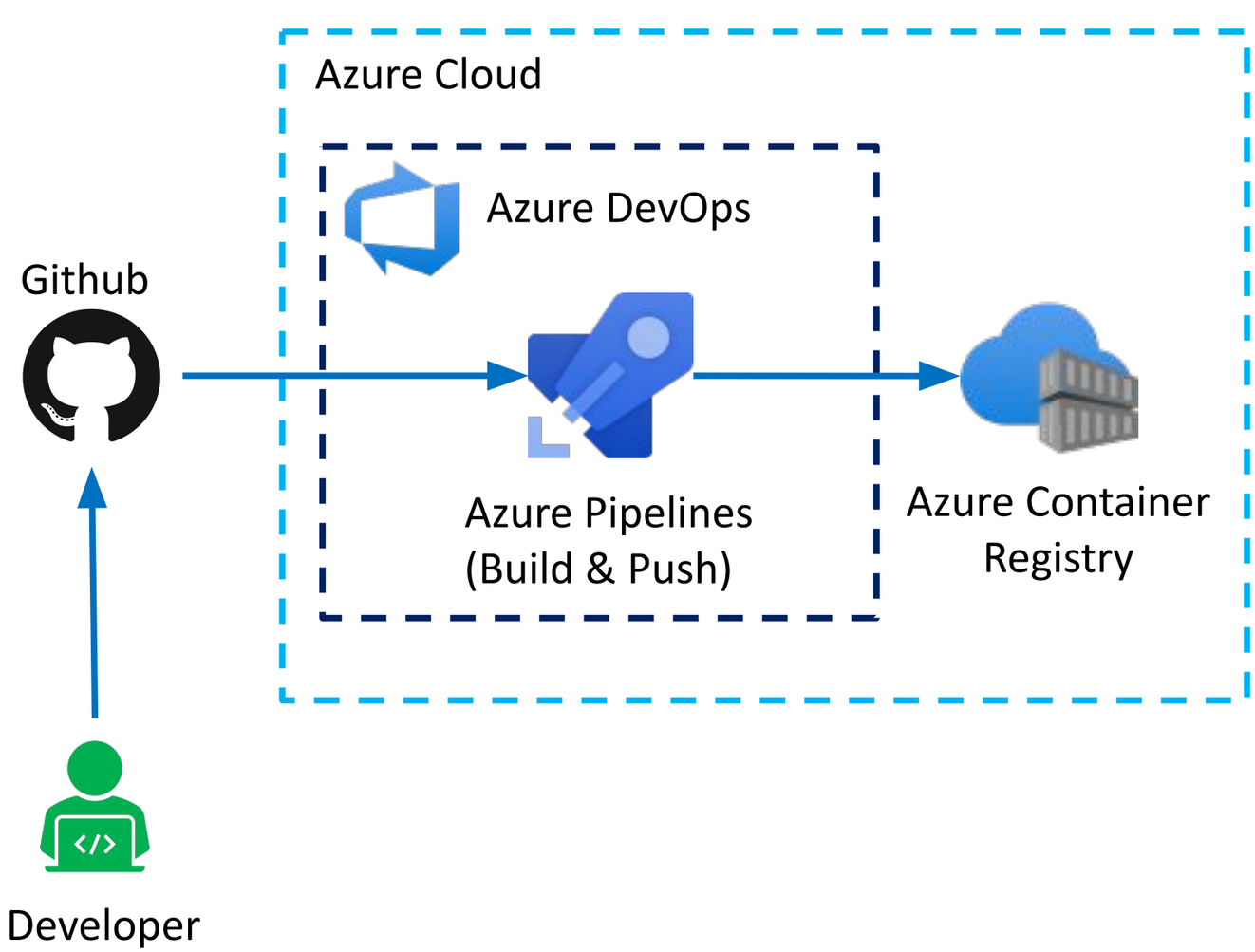
# Pull Docker Images from ACR using Service Principal Run on Azure AKS Nodepools

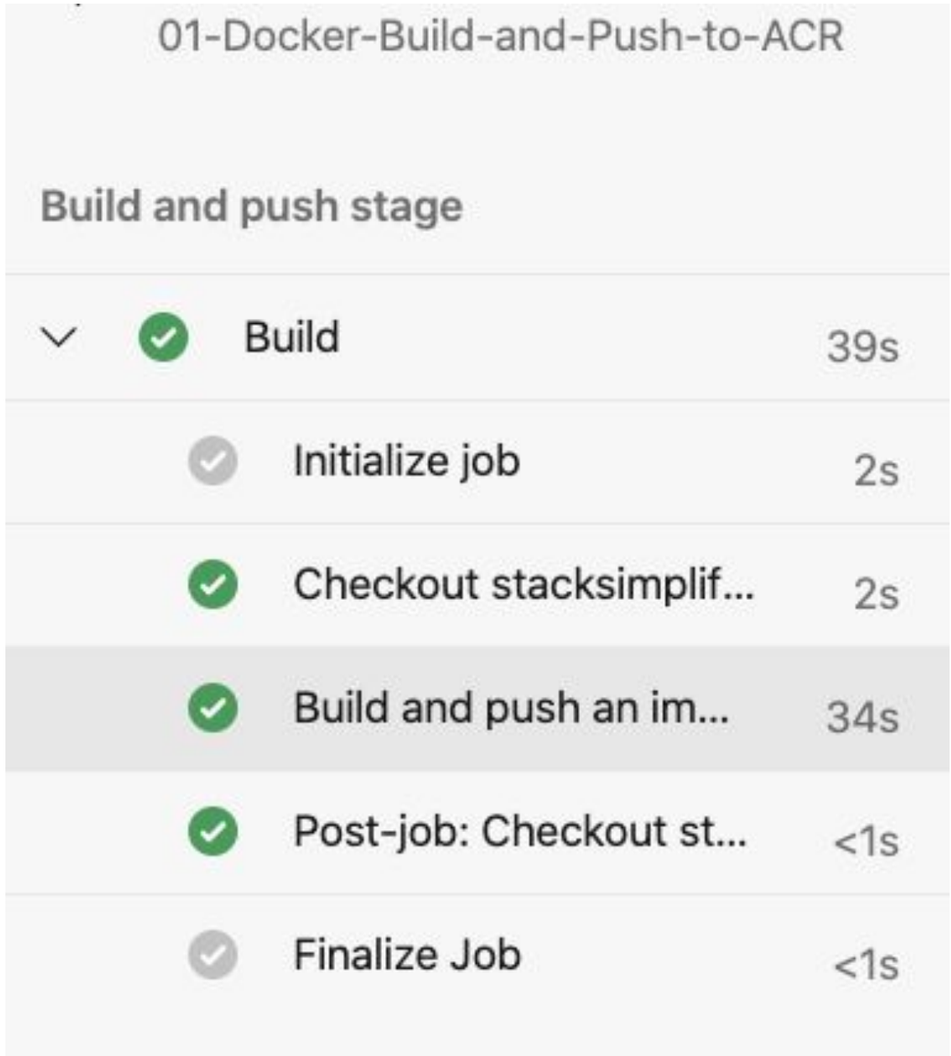


# Pull Docker Images from ACR using Service Principal Run on Azure Virtual Nodes

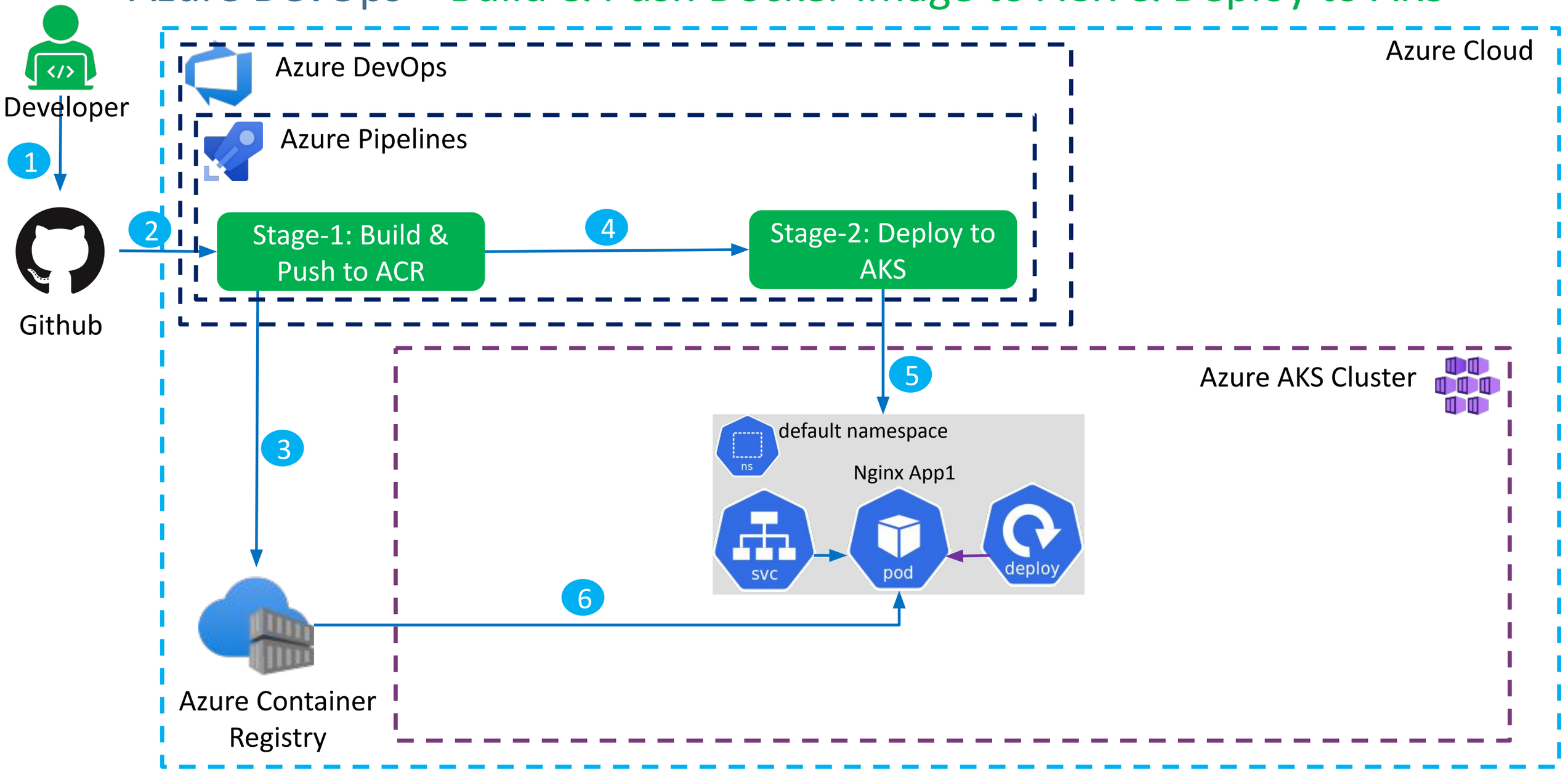


# Azure DevOps Pipelines – Build & Push Docker Image to ACR

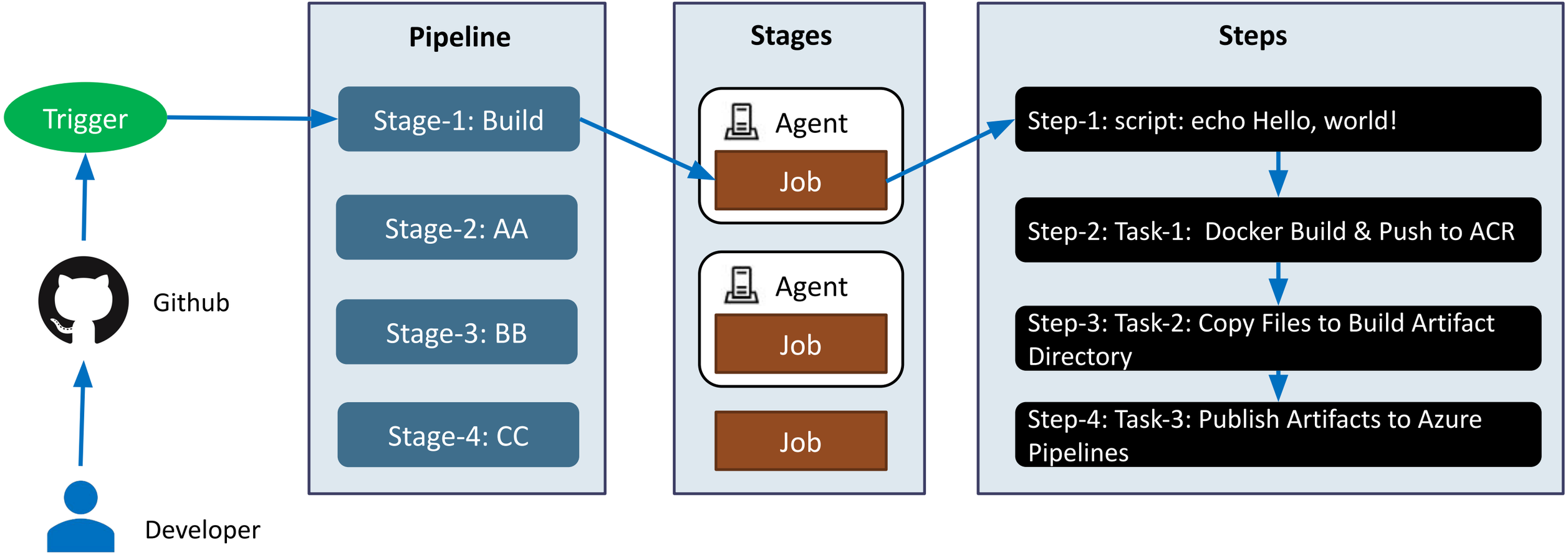


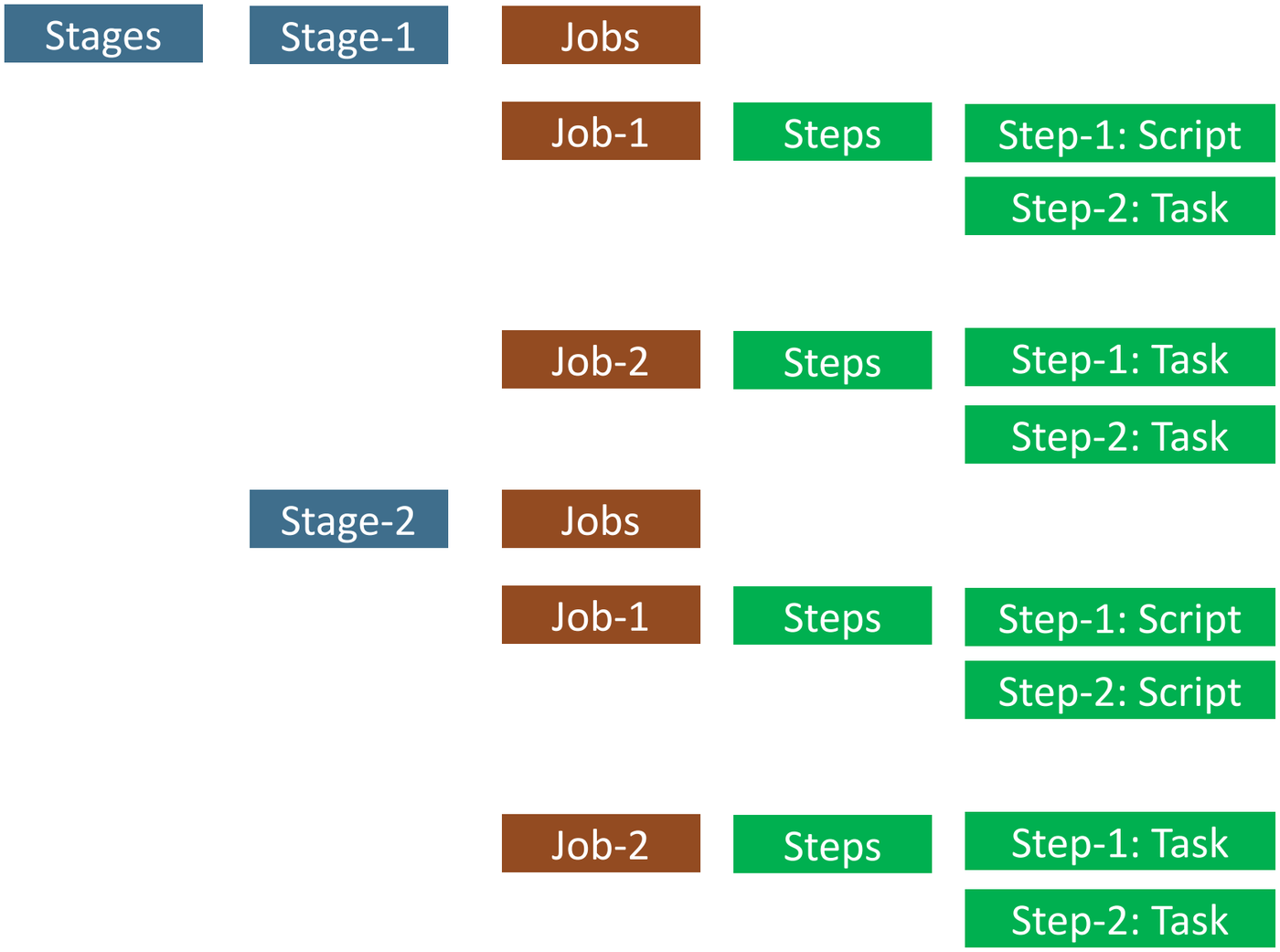
# Azure DevOps – Build & Push Docker Image to ACR & Deploy to AKS



# Azure Pipelines – Key Concepts



# Azure Pipelines – Key Concepts




```yaml
stages:
- stage: Stage-1
  jobs:
  - job: Job-1
    steps:
    - script: echo Step-1
    - script: echo Step-2
  - job: Job-2
    steps:
    - task: some task step-1
    - task: some task step-2
- stage: Stage-2
  jobs:
  - job: Job-1
    steps:
    - task: some task step-1
    - task: some task step-2
  - job: Job-2
    steps:
    - script: echo Step-1
    - script: echo Step-2
```

# Azure Pipelines – Starter Pipeline

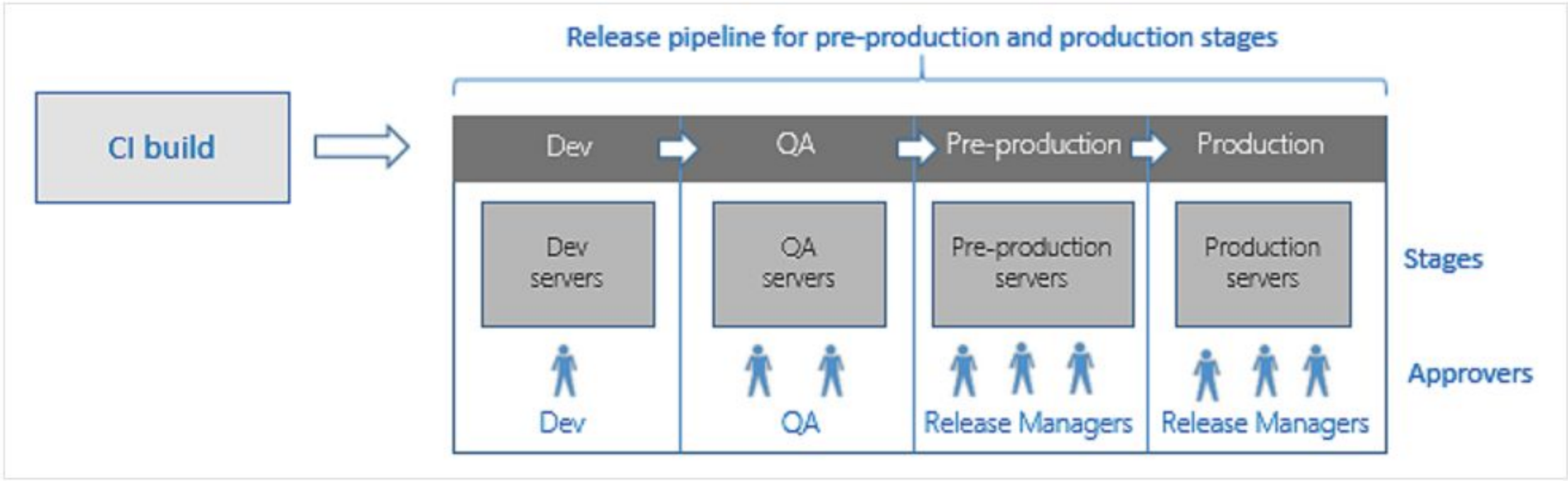
| Goal | Create a Pipeline that will build docker images, push them to Azure Container Registry and Publish Kubernetes Manifests to Azure Pipelines |
|---|---|

| Part-1 | Semi Customized |
|---|---|
| Task-1 | Use pre-defined Docker Build & Push Pipeline |
| Task-2 | Customize Pipeline to Use Copy Files Task |
| Task-3 | Customize Pipeline to Use Publish Build Artifacts Task |

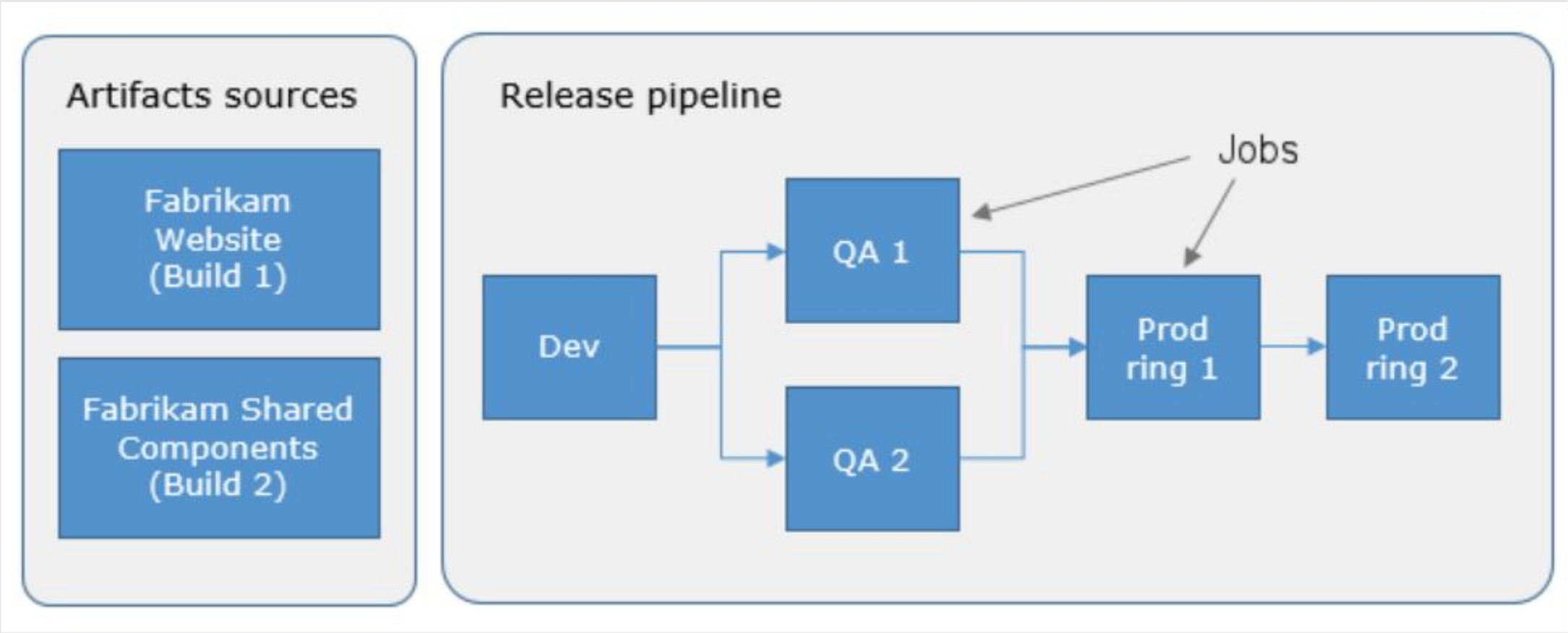
| Part-2 | Fully Customized using Starter Pipeline |
|---|---|
| Task-1 | Start using Starter pipeline and use Docker Build or Push Docker Images Task |
| Task-2 | Customize Pipeline to Use Copy Files Task |
| Task-3 | Customize Pipeline to Use Publish Build Artifacts Task |

# Azure DevOps – Release Pipelines

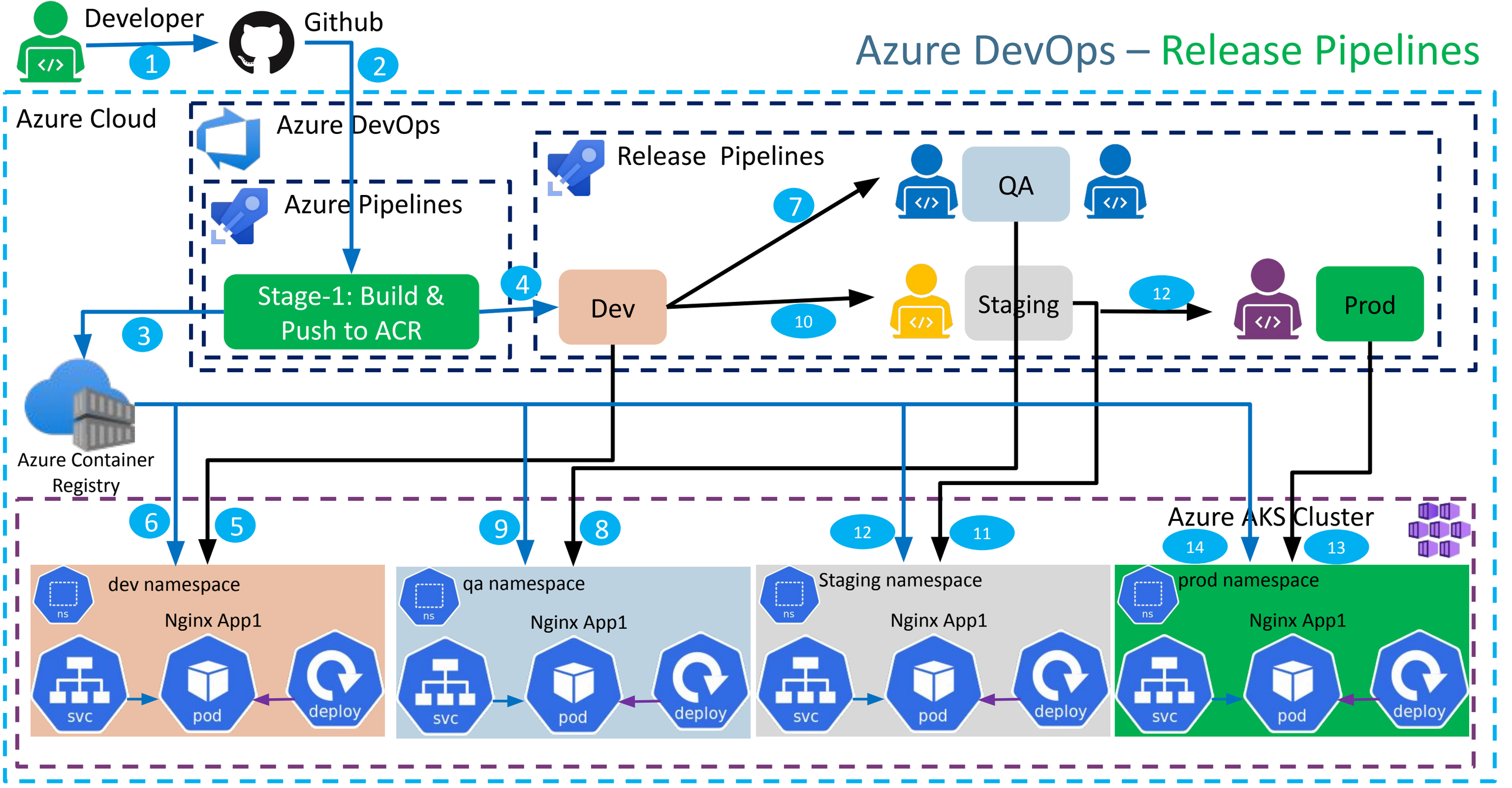
To achieve Continuous Delivery we use Release Pipelines



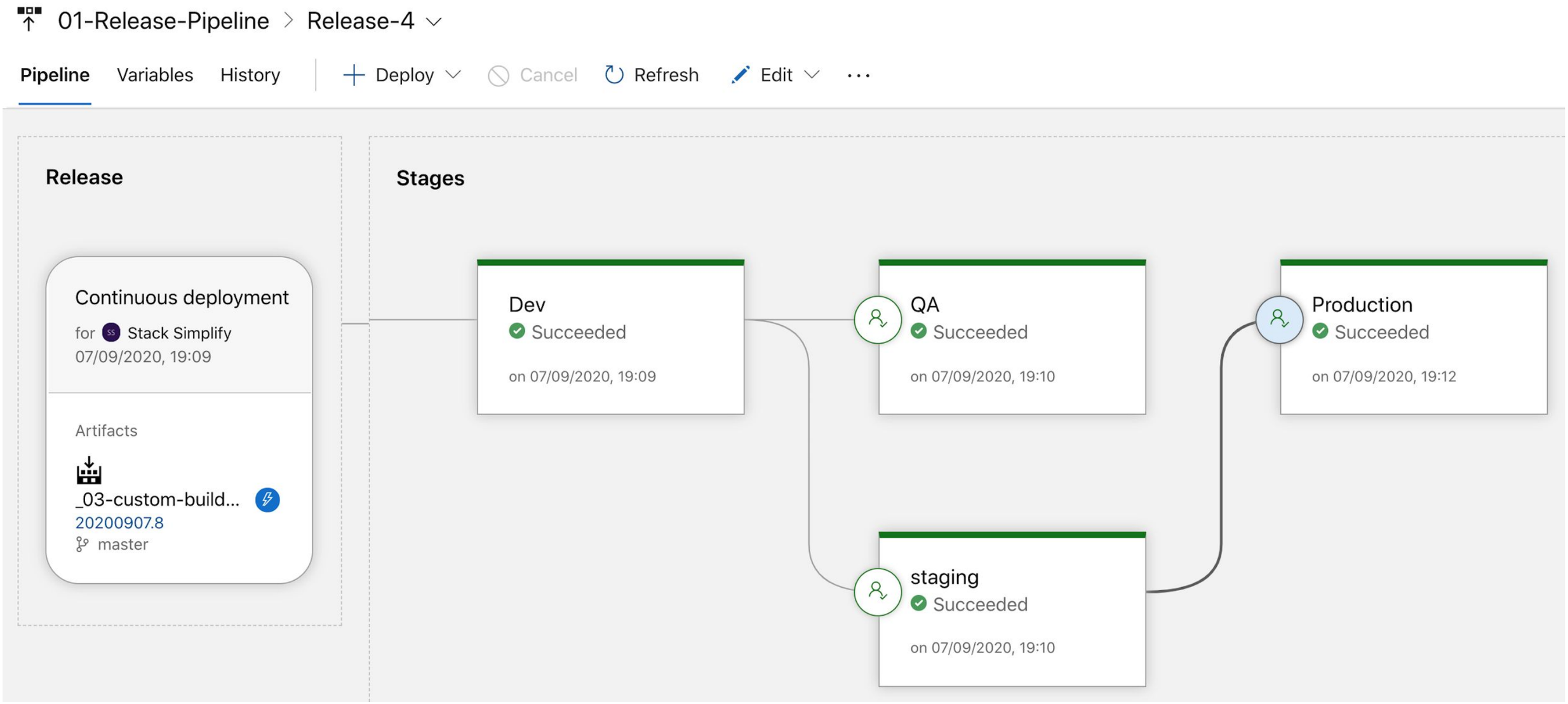
# Azure DevOps – Release Pipelines



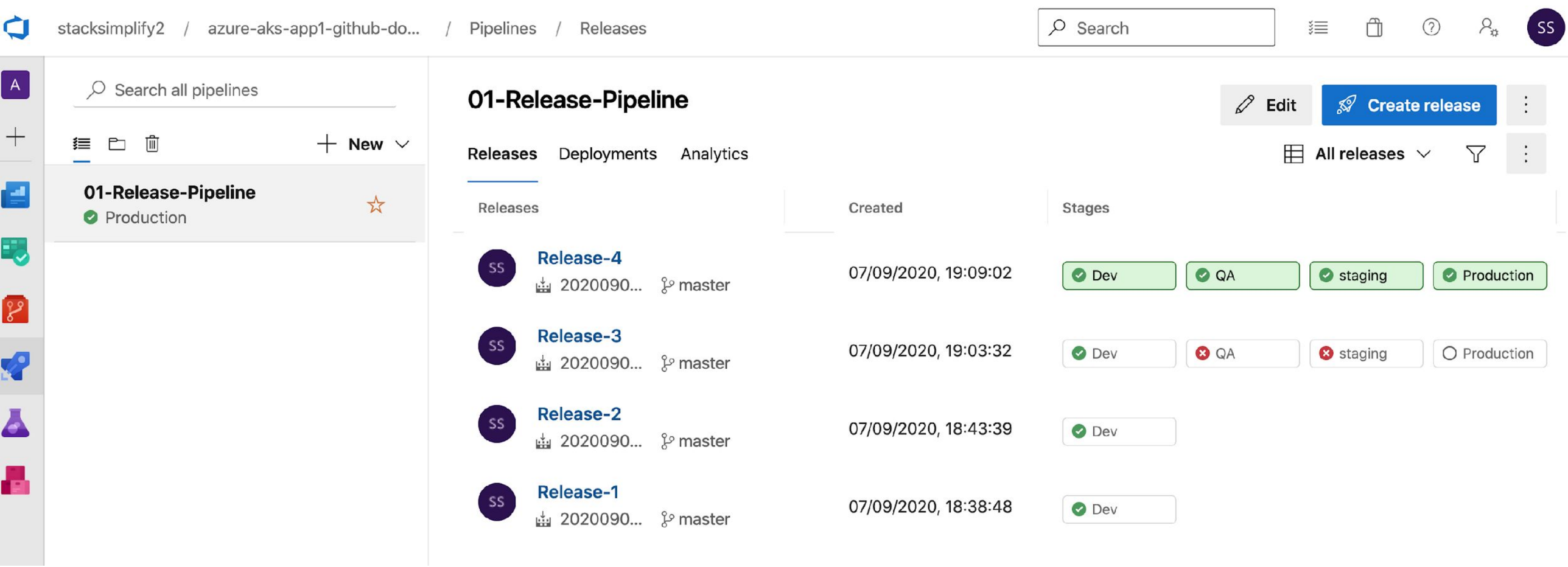
# Azure DevOps – Release Pipelines



# Azure Release Pipelines



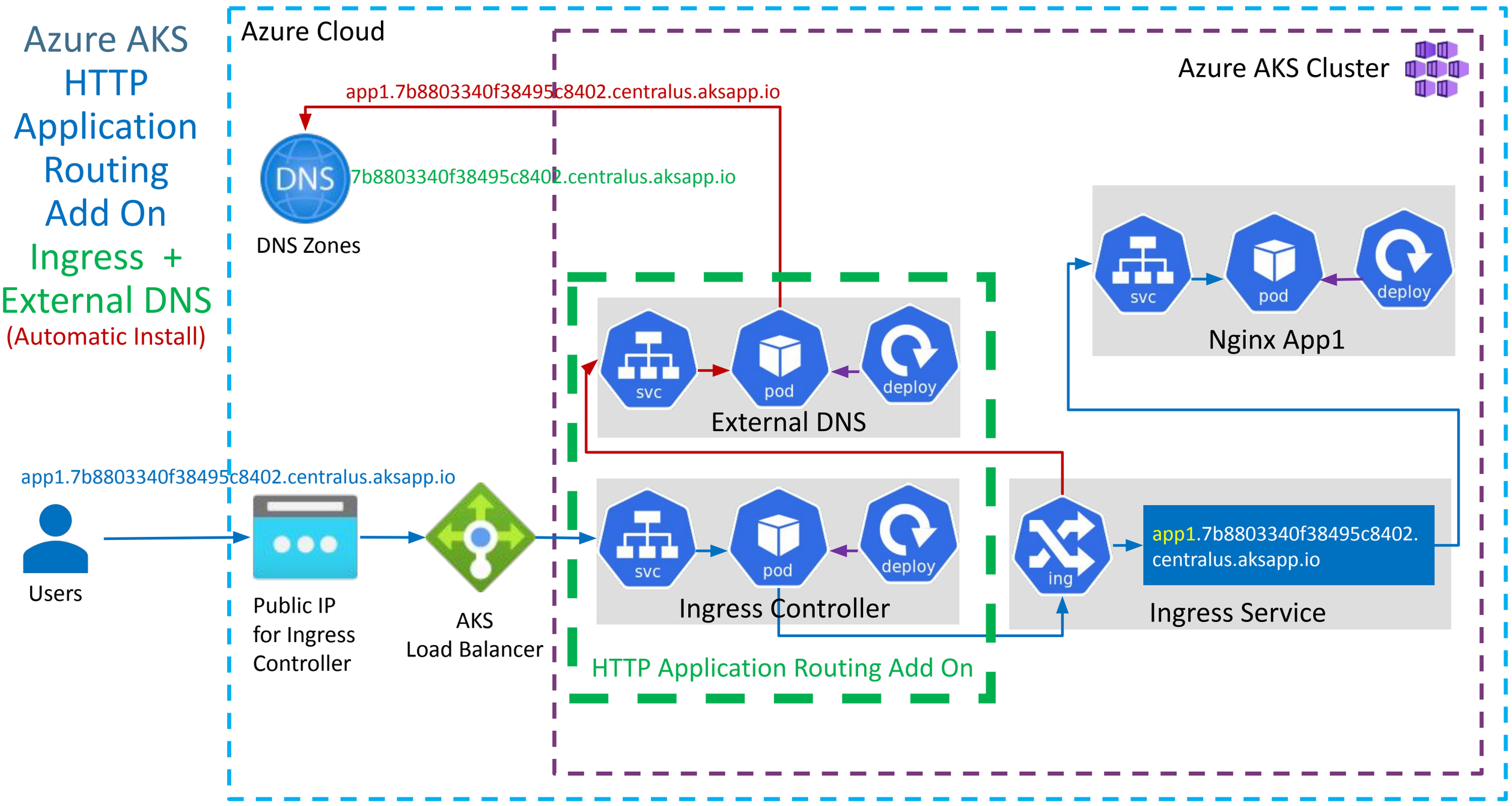
# Azure Release Pipelines - Releases



# Azure AKS HTTP Application Routing Add On

## Ingress + External DNS

(Automatic Install)



# Azure AKS Cluster Access

```
# Configure AKSDEMO3 & 4 Cluster Access for kubectl
az aks get-credentials --resource-group aks-rg3 --name aksdemo3
az aks get-credentials --resource-group aks-rg4 --name aksdemo4
```

```
# View kubeconfig
kubectl config view
```

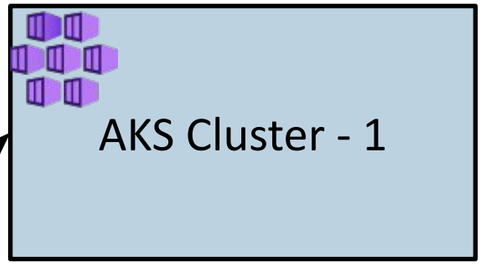
AKS Admin

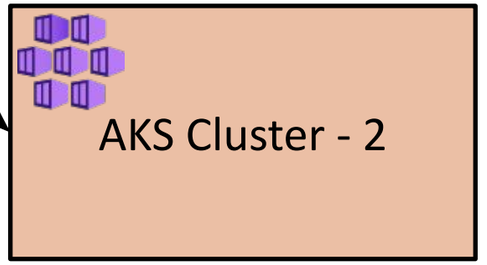
kubectl

Current Context

AKS Cluster-1

AKS Cluster-2

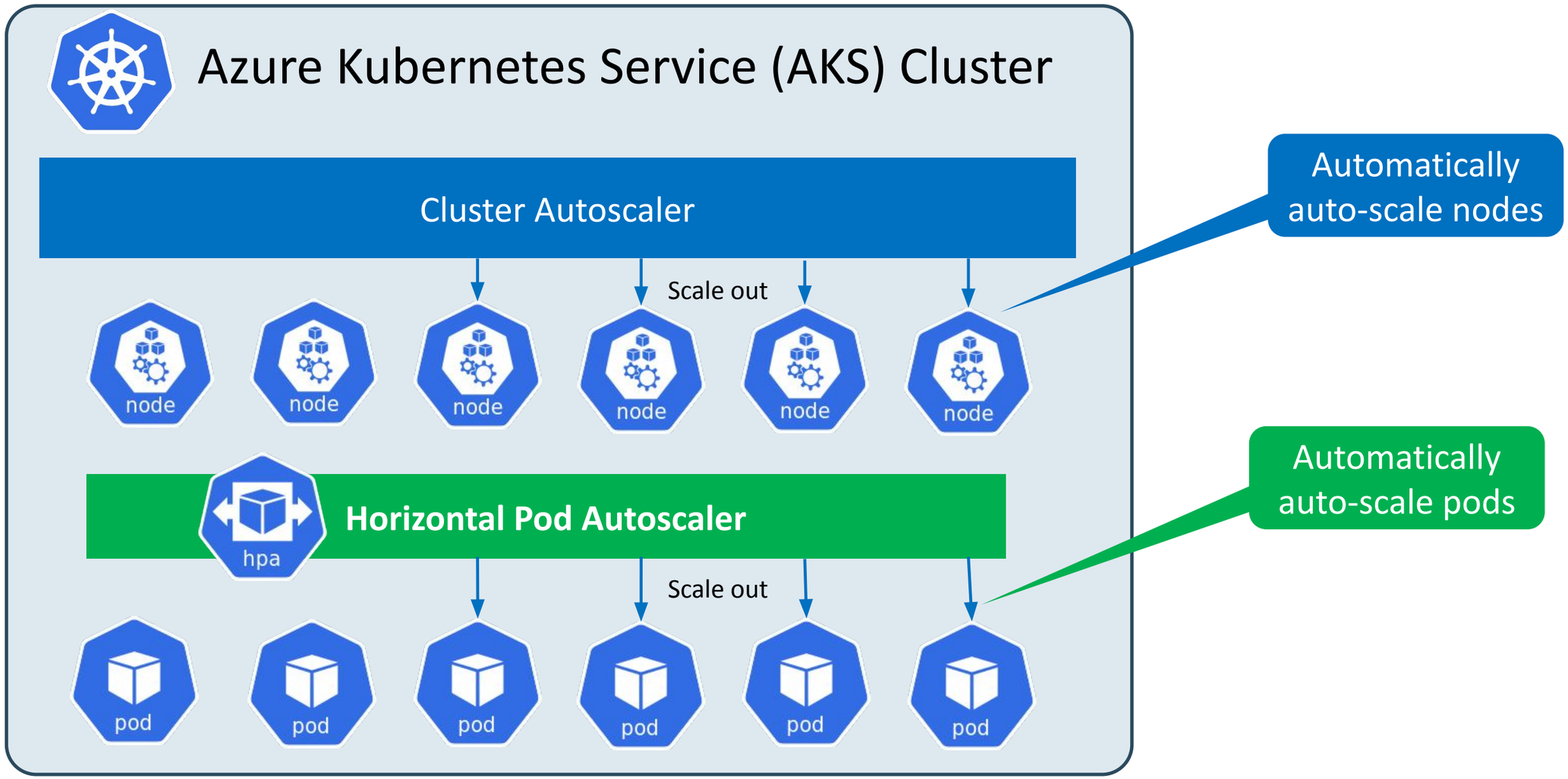



```
# View the current context for kubectl
kubectl config current-context
```

```
# Switch Context
kubectl config use-context aksdemo3
```

# Azure AKS – Autoscaling Nodes & Pods



# Thank You