

# 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

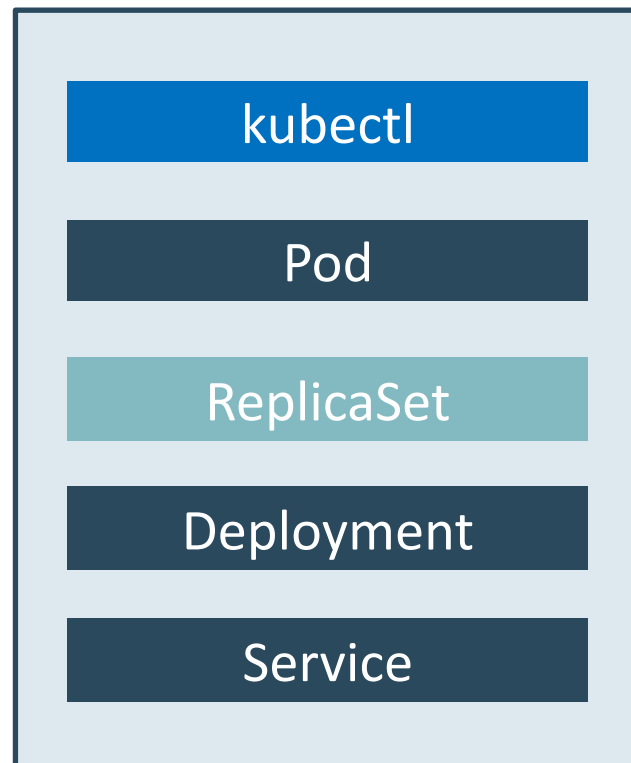


Azure  
Managed  
Service  
Identity

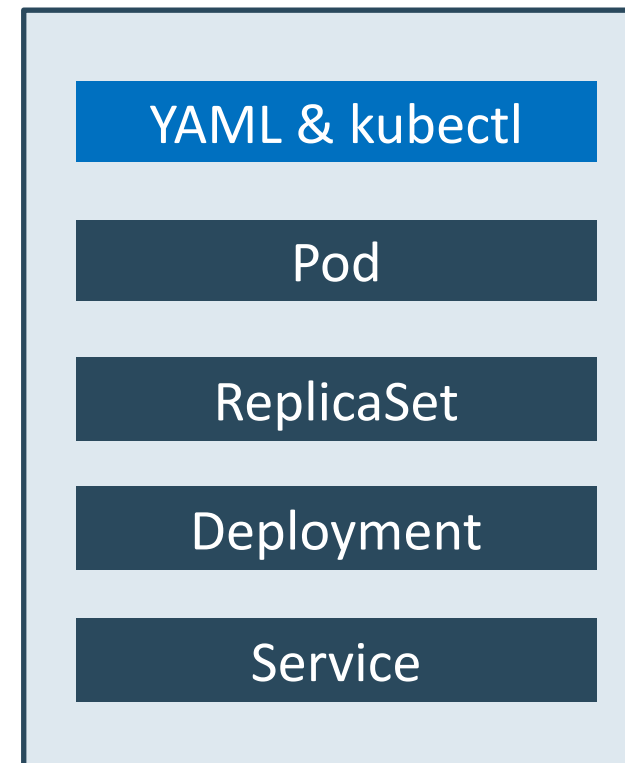
# Kubernetes - Imperative & Declarative

## Kubernetes Fundamentals

Imperative



Declarative





Users

<http://<External-IP-from-get-service-output>>



Azure Disks



Storage Class



Persistent Volume Claim

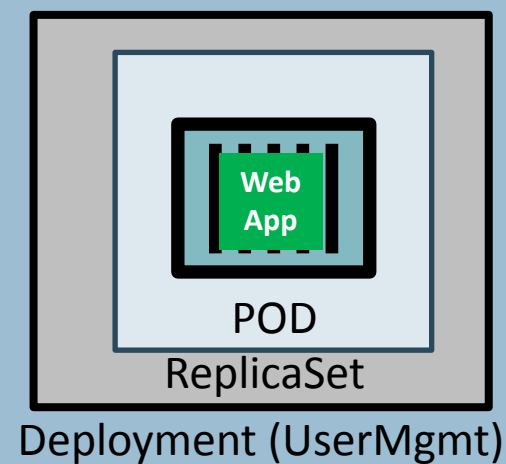


Persistent Volumes

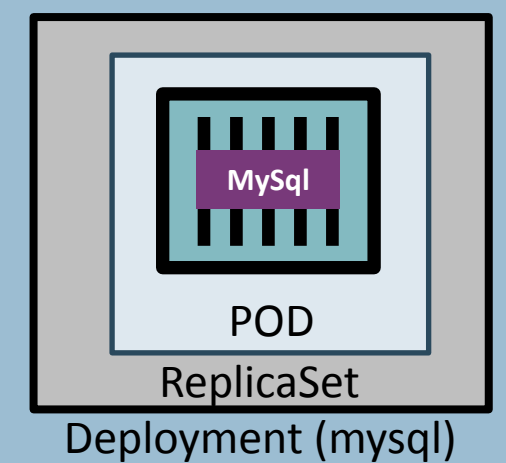


Deployment

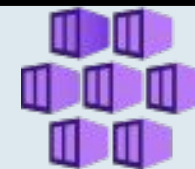
UserMgmt – LoadBalancer Service



MySQL – ClusterIP Service



Azure AKS Cluster



Load Balancer Service



Deployment



Environment Variables



Init Containers



Config Map



Environment Variables



Volumes

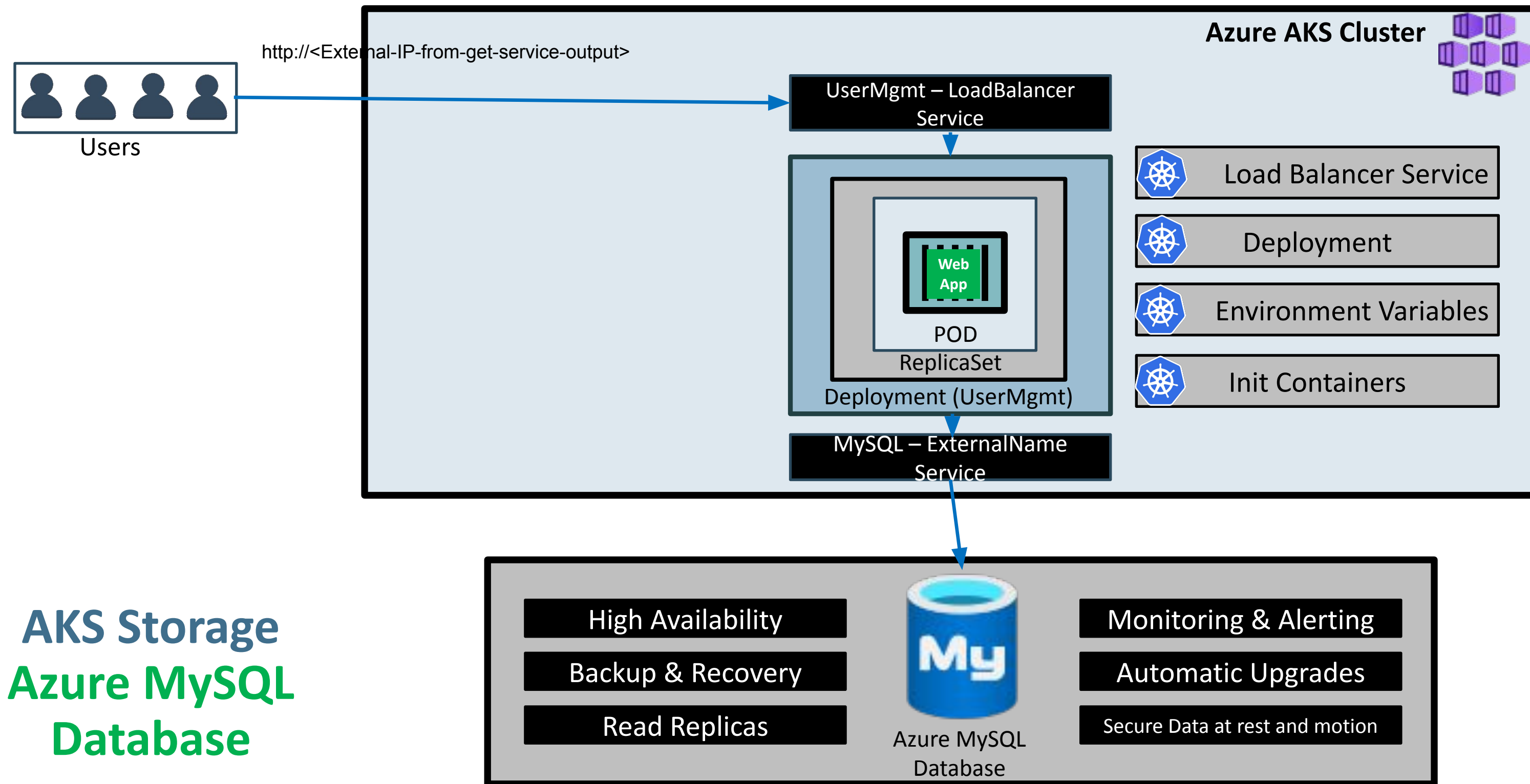


Volume Mounts



ClusterIP Service

AKS Storage  
Azure Disks



**AKS Storage**  
**Azure MySQL**  
**Database**

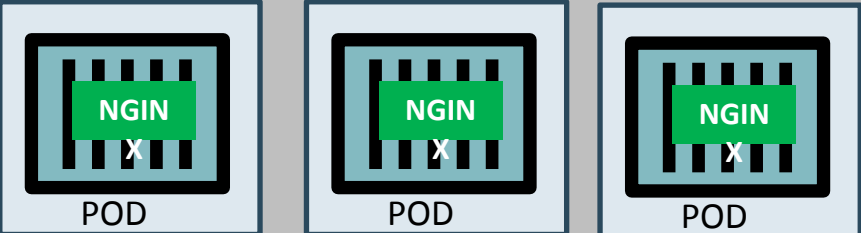


Users

LoadBalancer Service

AKS Storage  
Azure Files

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



ReplicaSet  
Deployment

Azure Storage Account



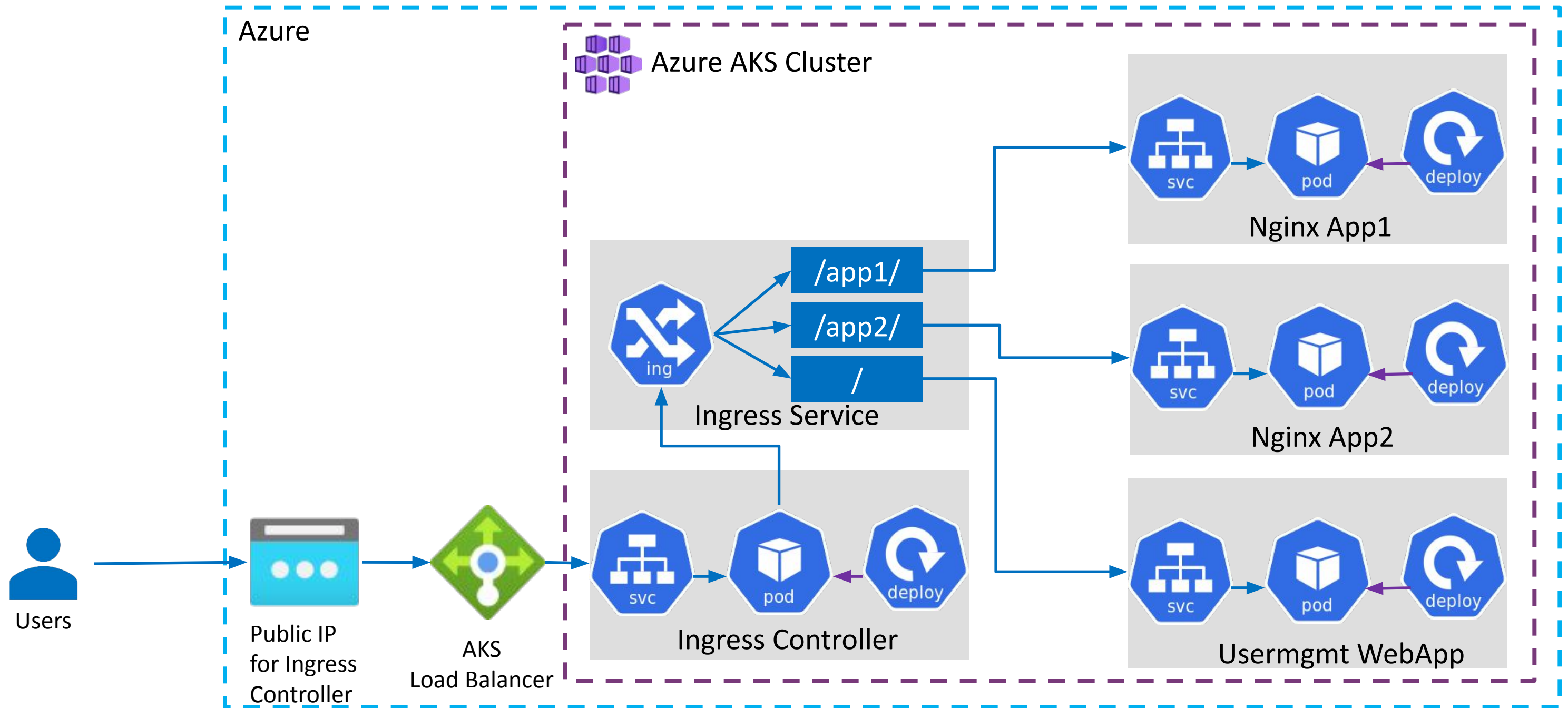
Azure File Shares

Azure AKS Cluster



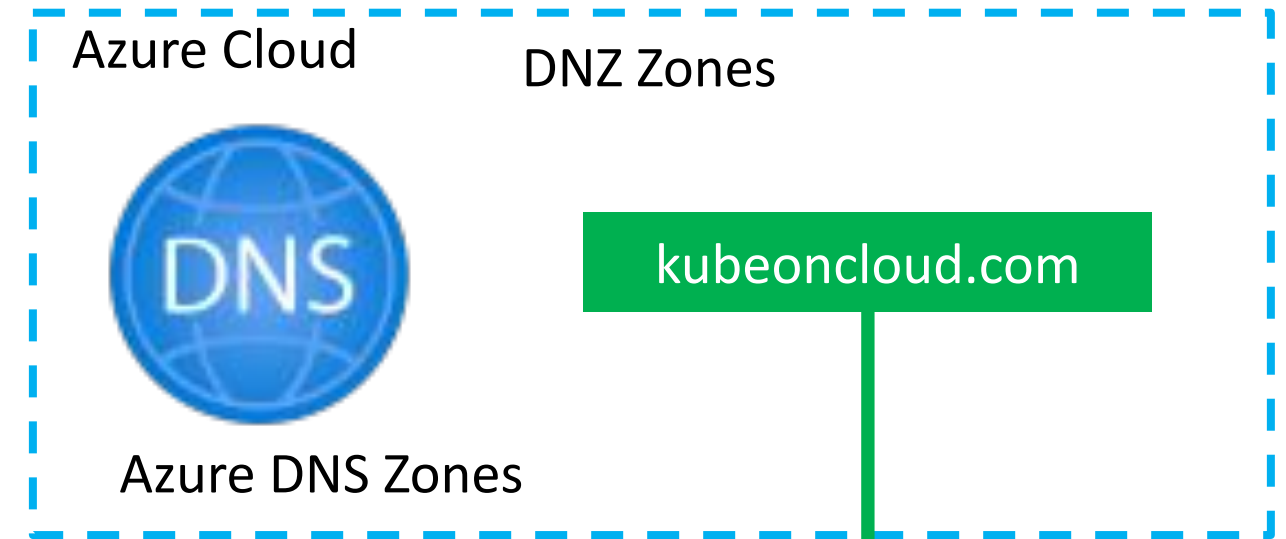
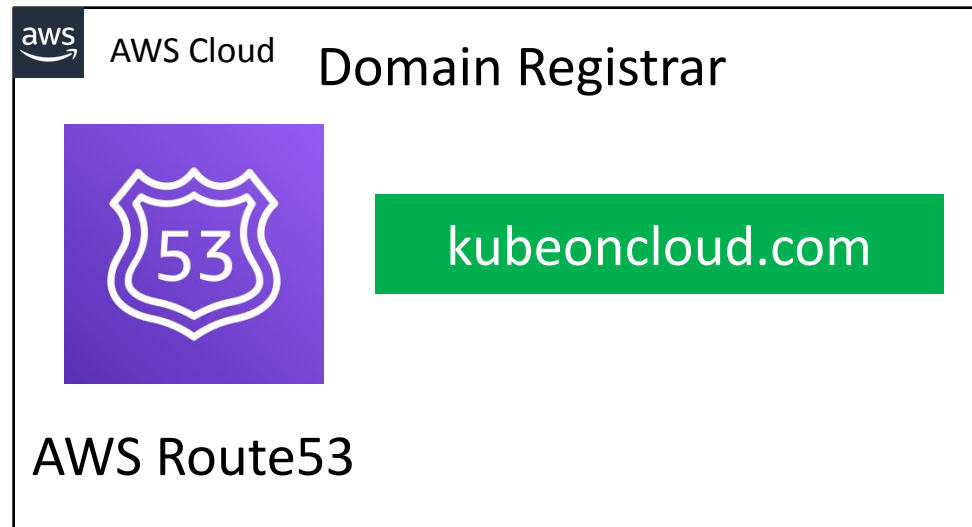
- Load Balancer Service
- Deployment
- Volumes
- Volume Mounts
- Storage Class
- Persistent Volume Claim
- Persistent Volumes

# Azure AKS & Nginx Ingress – Context Path Based Routing





# Delegate Domain to Azure DNS



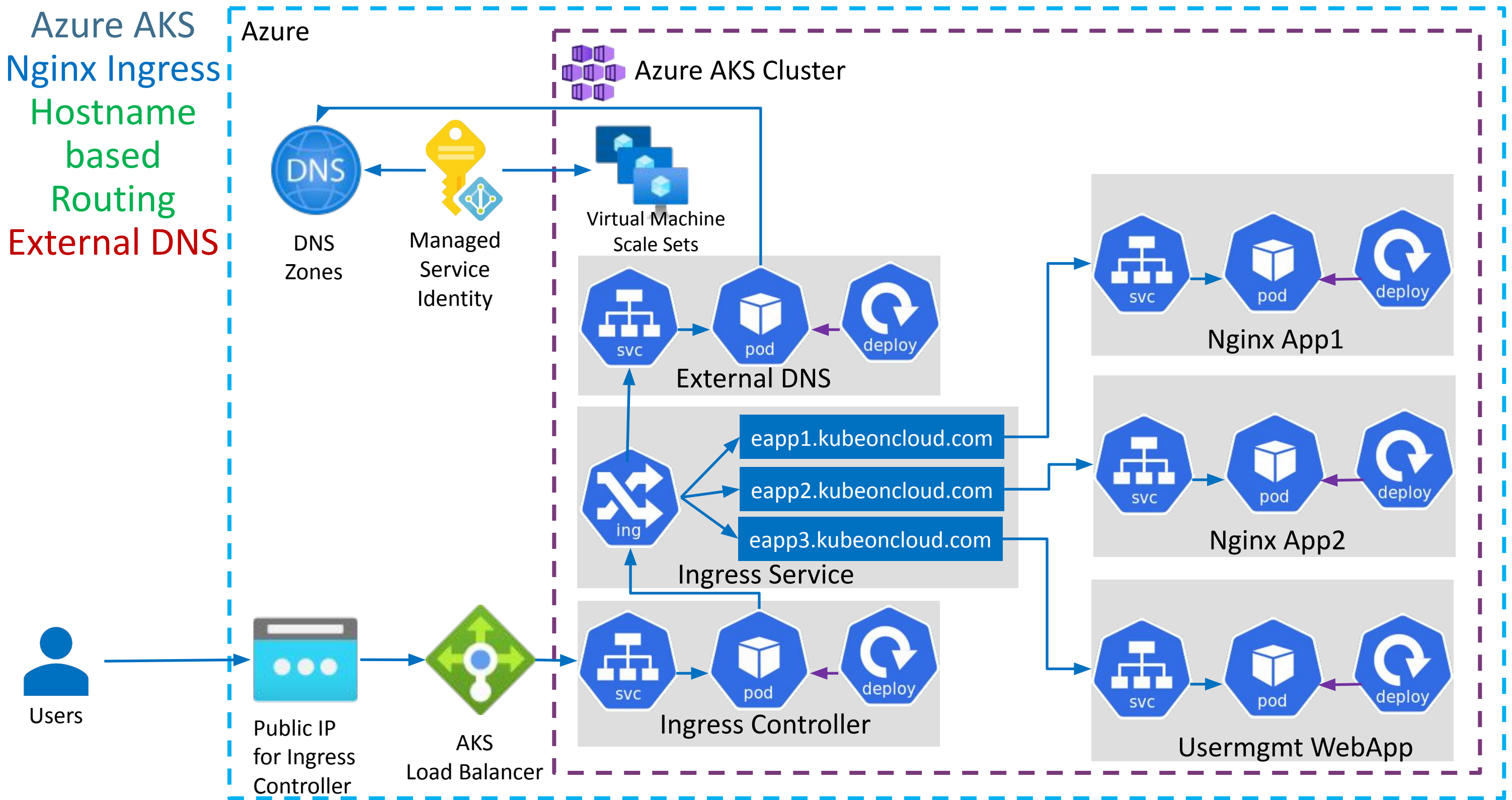
```
Kalyans-MacBook-Pro:/ kdaida$ nslookup -type=NS kuboncloud.com
Server:          192.168.0.1
Address:         192.168.0.1#53

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

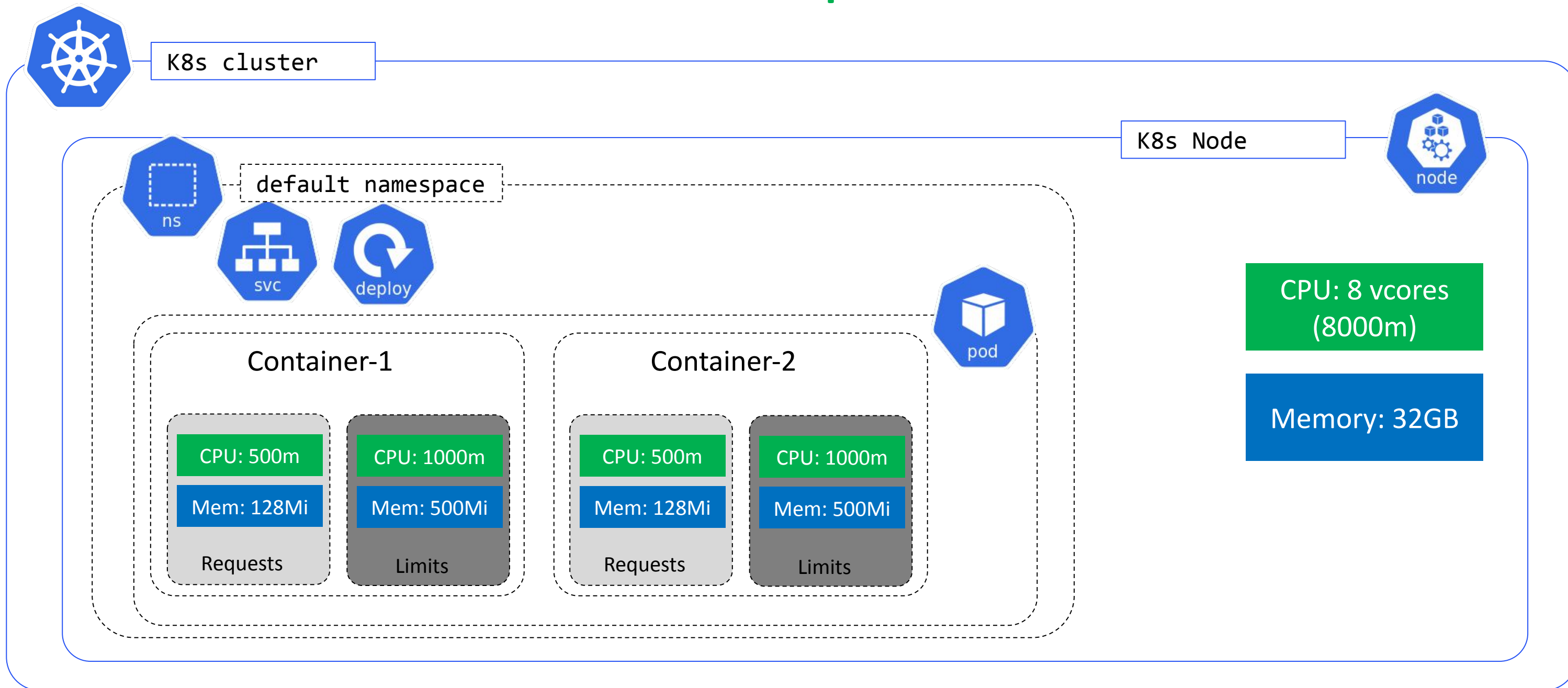
Authoritative answers can be found from:
```



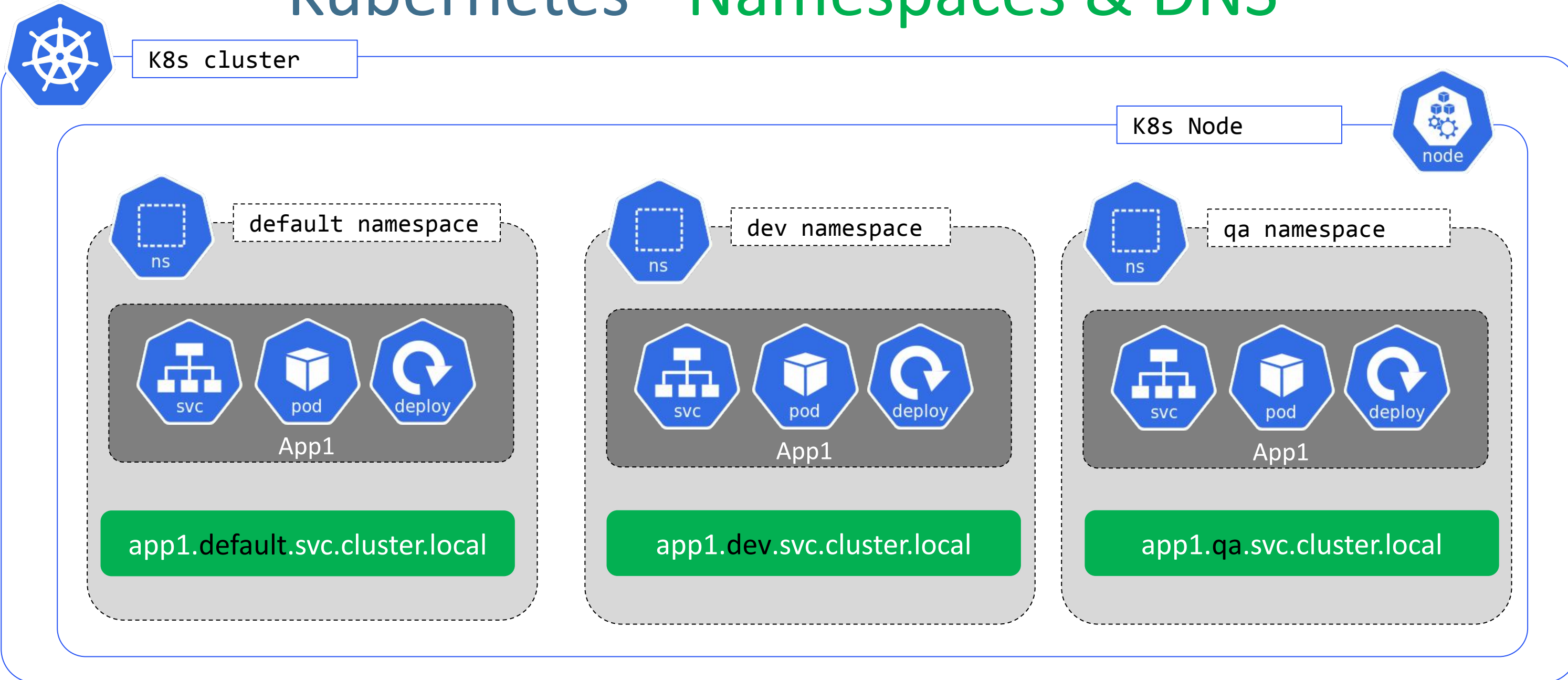




# Kubernetes – Requests & Limits



# Kubernetes - Namespaces & DNS



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

# Limit Range

## Limit Range Manifest

```
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
```

LoadBalancer Service

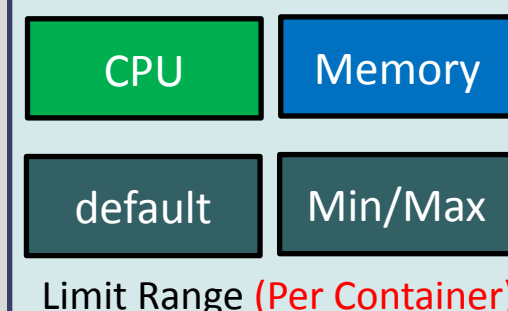
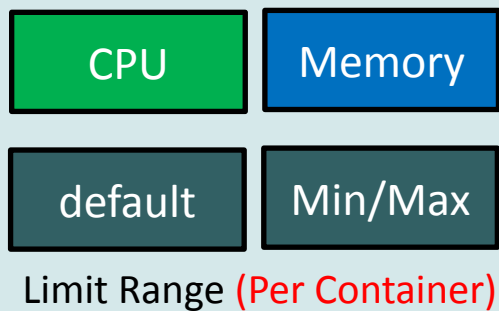
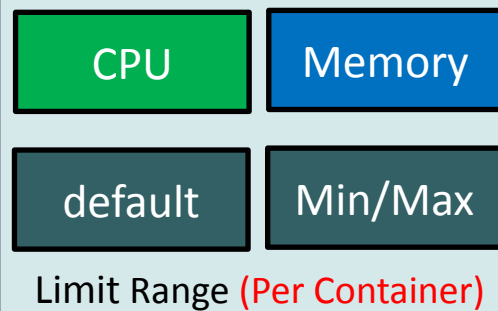
LoadBalancer Service

LoadBalancer Service

ClusterIP Service

ClusterIP Service

ClusterIP Service



Namespace: dev

Namespace: qa

Namespace: staging

Pod  
ReplicaSet  
Deployment

Pod  
ReplicaSet  
Deployment

Pod  
ReplicaSet  
Deployment

Pod  
ReplicaSet  
Deployment

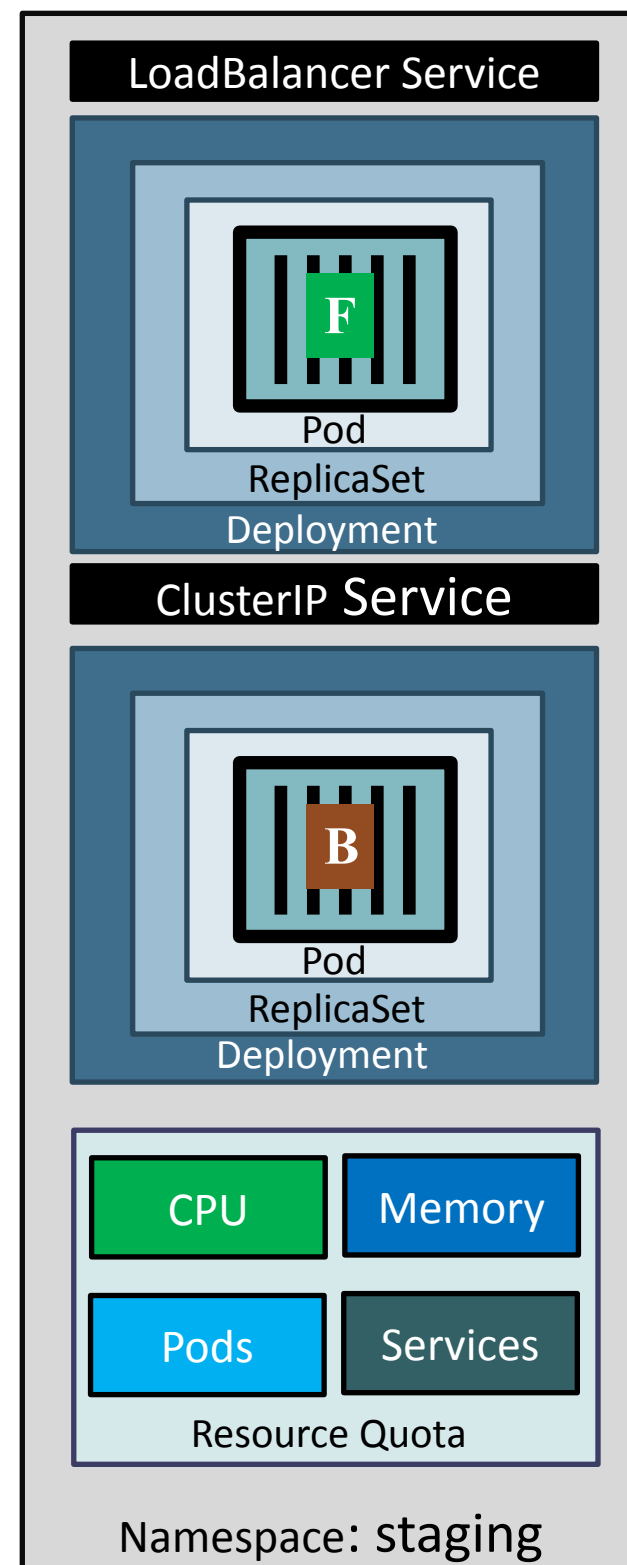
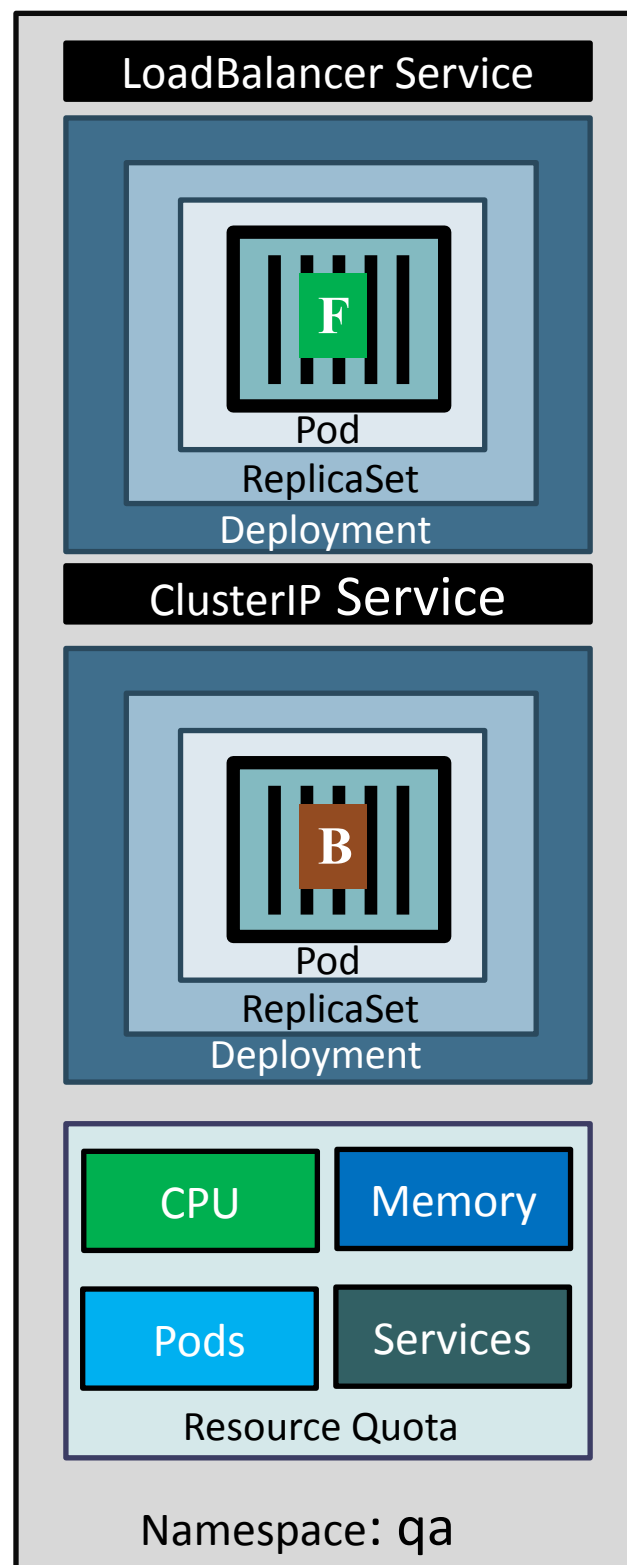
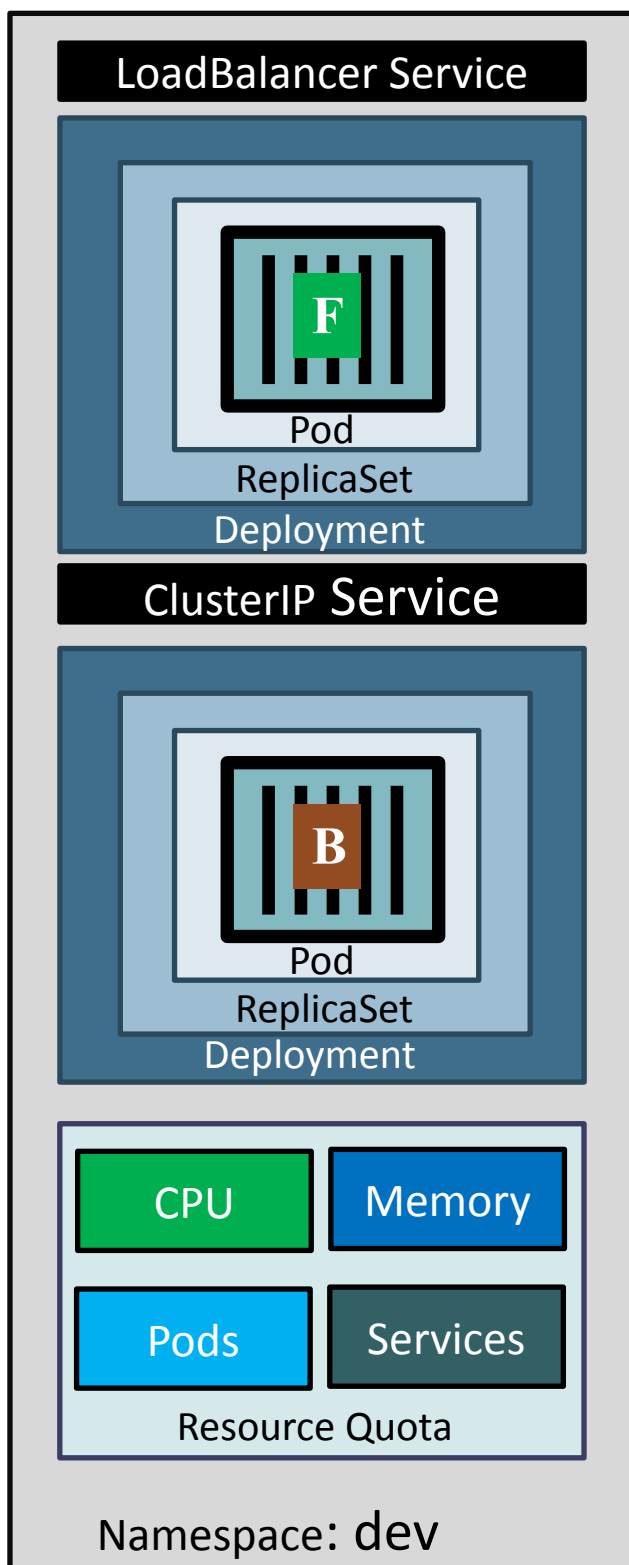
Pod  
ReplicaSet  
Deployment

Pod  
ReplicaSet  
Deployment

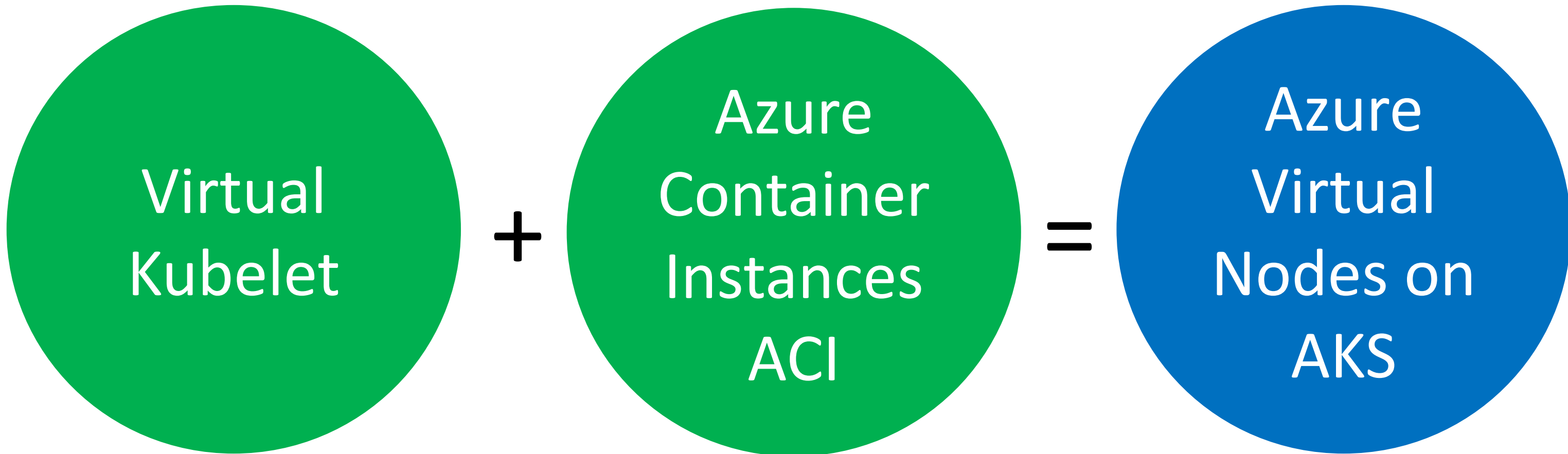
# Resource Quota

## Resource Quota Manifest

```
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

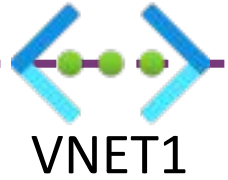
Azure Cloud



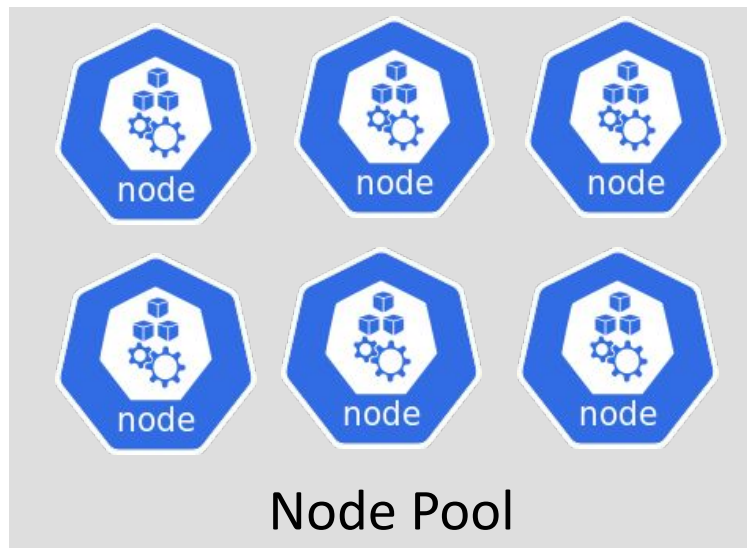
Azure AKS Cluster



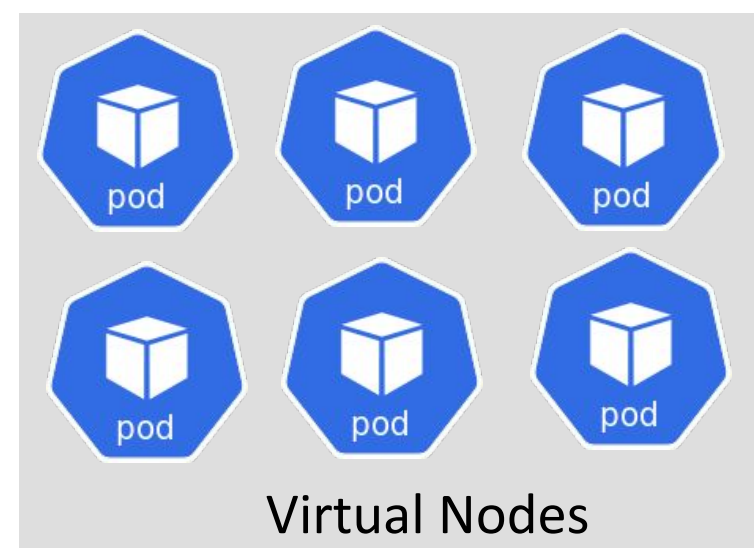
AKS Cluster  
Control Plane



VNET1



Subnet: default



Azure Container Instance - ACI

Subnet: virtual-node-aci



# Azure ACR and AKS - Integration

Docker Developer



Build and Push  
Docker Image

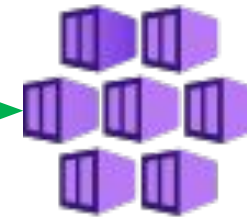
Azure Cloud

Azure Container  
Registry



Azure AKS Cluster

Integrate ACR and AKS



Azure AKS Cluster  
Control Plane

Pull Docker Image from ACR

http://<public-ip>



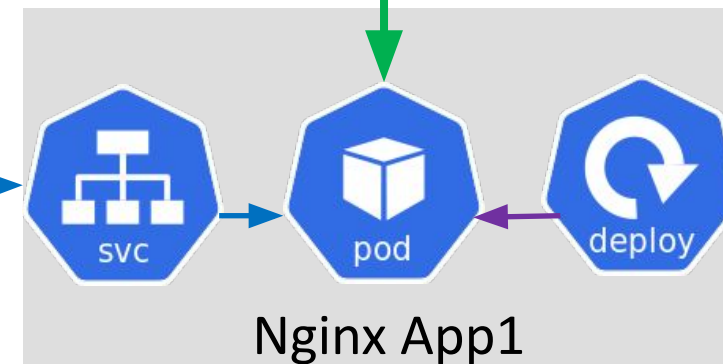
Users



Public IP  
App1 LB



AKS  
Load Balancer



Nginx App1

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

Docker Developer



Build and Push  
Docker Image

Azure Cloud

Azure Container  
Registry



Azure AKS Cluster



Azure AKS Cluster  
Control Plane

Pull Docker Image from ACR

http://<public-ip>



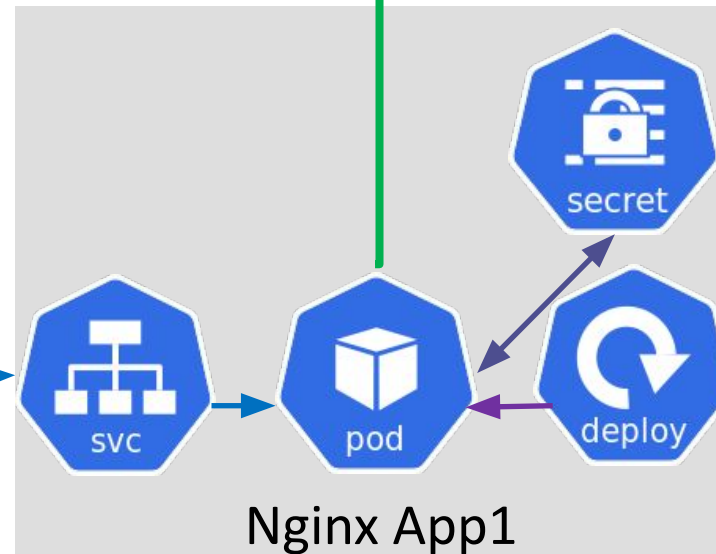
Users



Public IP  
App1 LB



AKS  
Load Balancer



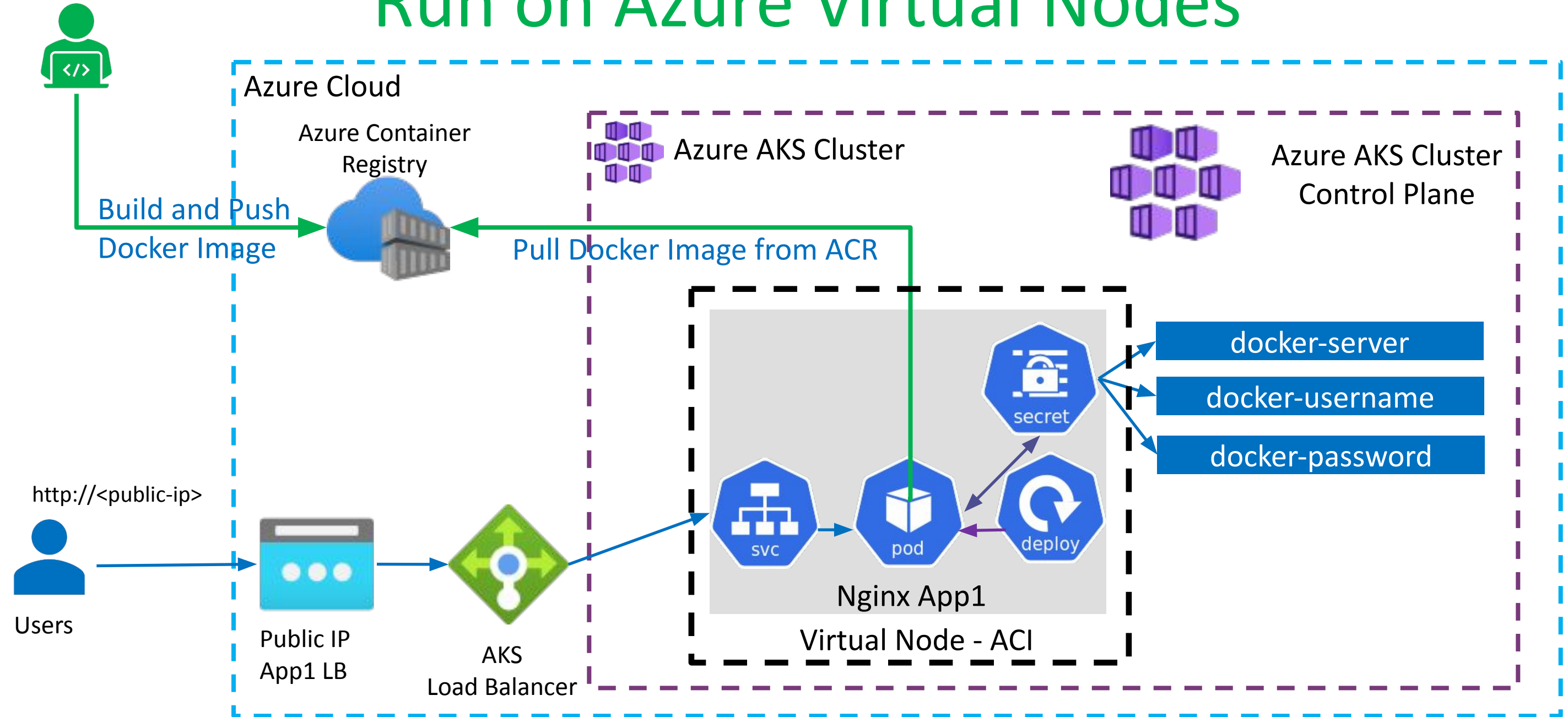
docker-server

docker-username

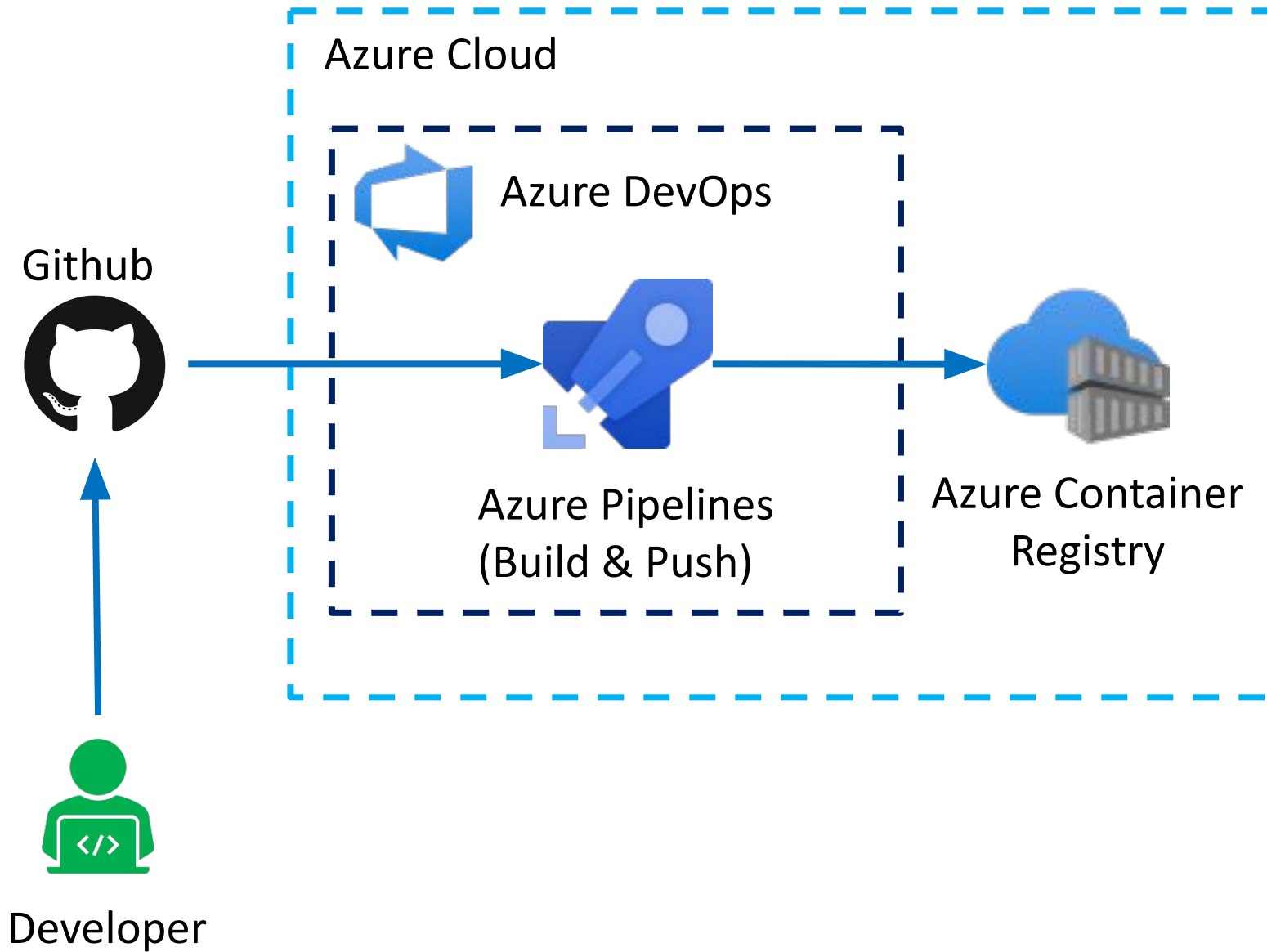
docker-password

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

Docker Developer

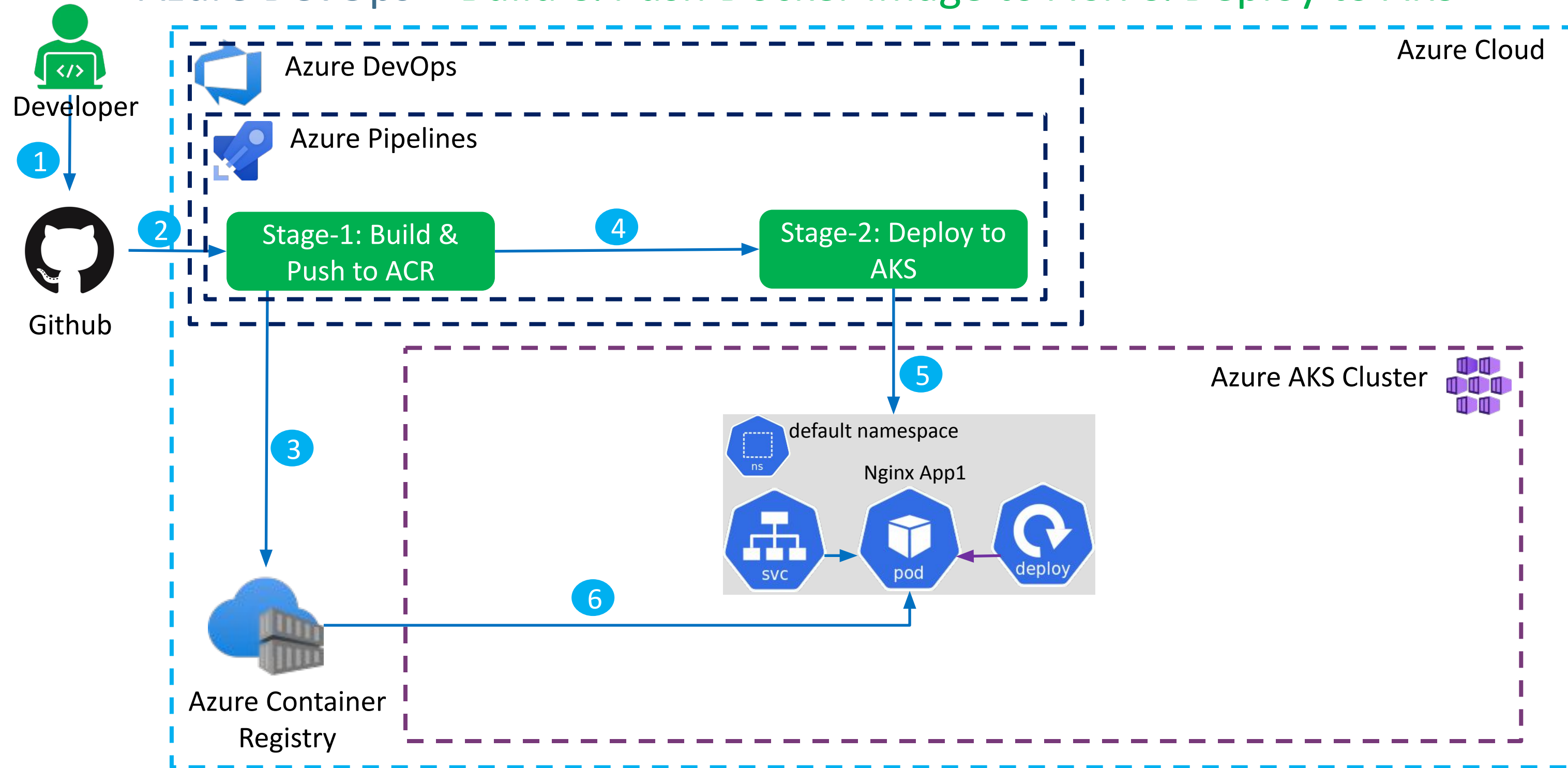


# Azure DevOps Pipelines – Build & Push Docker Image to ACR

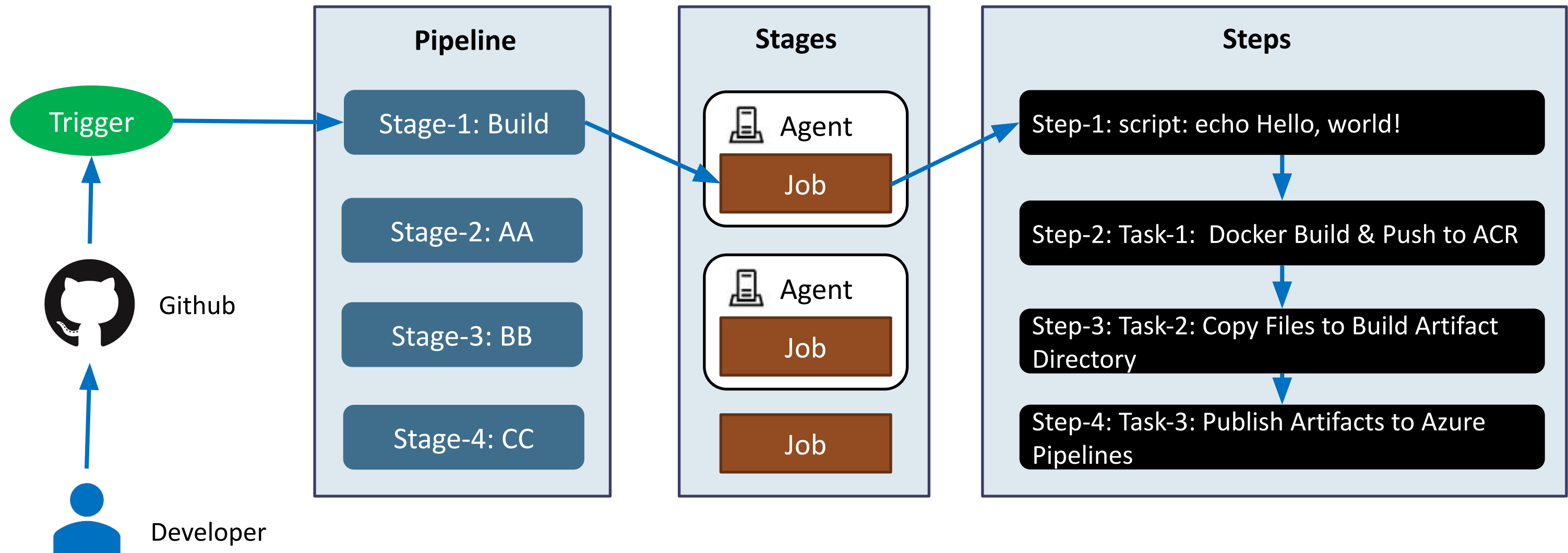


01-Docker-Build-and-Push-to-ACR			
Build and push stage			
▼	✓	Build	39s
	✓	Initialize job	2s
	✓	Checkout stacksimplif...	2s
	✓	Build and push an im...	34s
	✓	Post-job: Checkout st...	<1s
	✓	Finalize Job	<1s

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



# Azure Pipelines – Key Concepts



# Azure Pipelines – Key Concepts



Stages

Stage-1

Jobs

Job-1

Steps

Step-1: Script

Step-2: Task

Job-2

Steps

Step-1: Task

Step-2: Task

Stage-2

Jobs

Job-1

Steps

Step-1: Script

Step-2: Script

Job-2

Steps

Step-1: Task

Step-2: Task

```
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

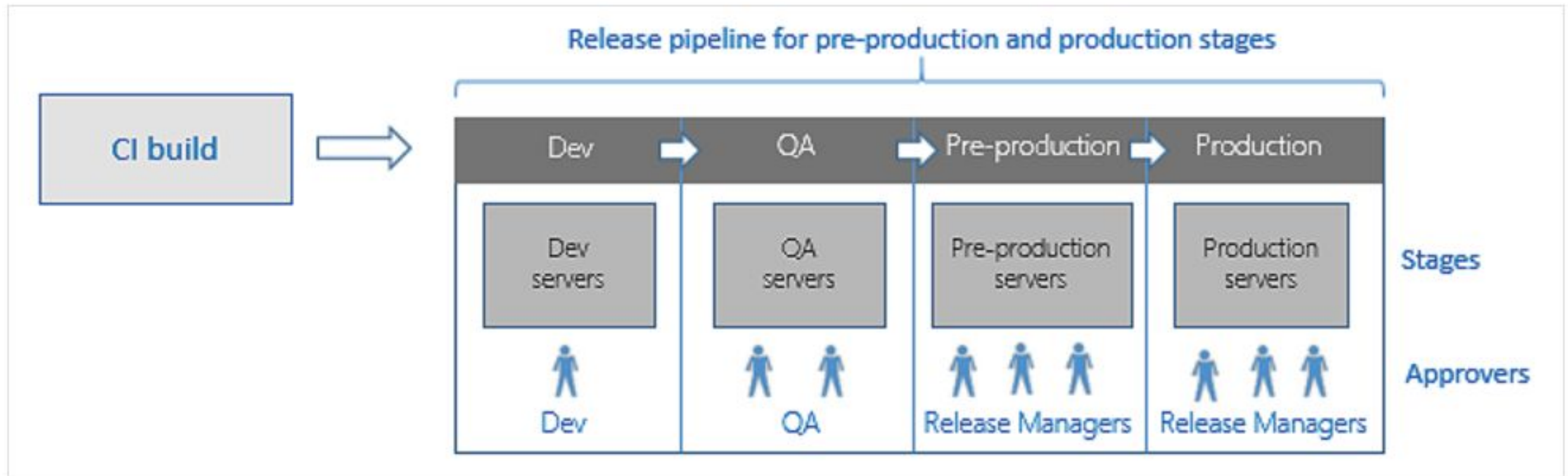
Customize Pipeline to Use **Copy Files Task**

#### Task-3

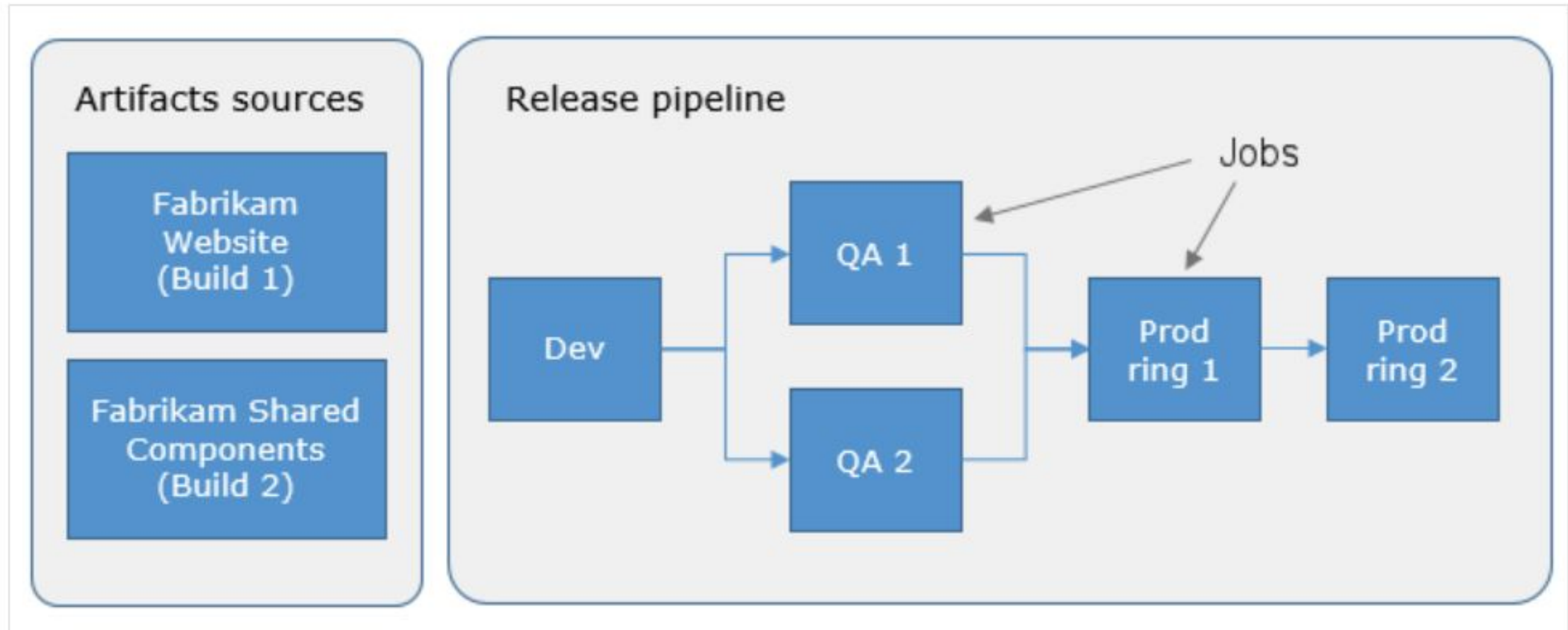
Customize Pipeline to Use **Publish Build Artifacts Task**

# Azure DevOps – Release Pipelines

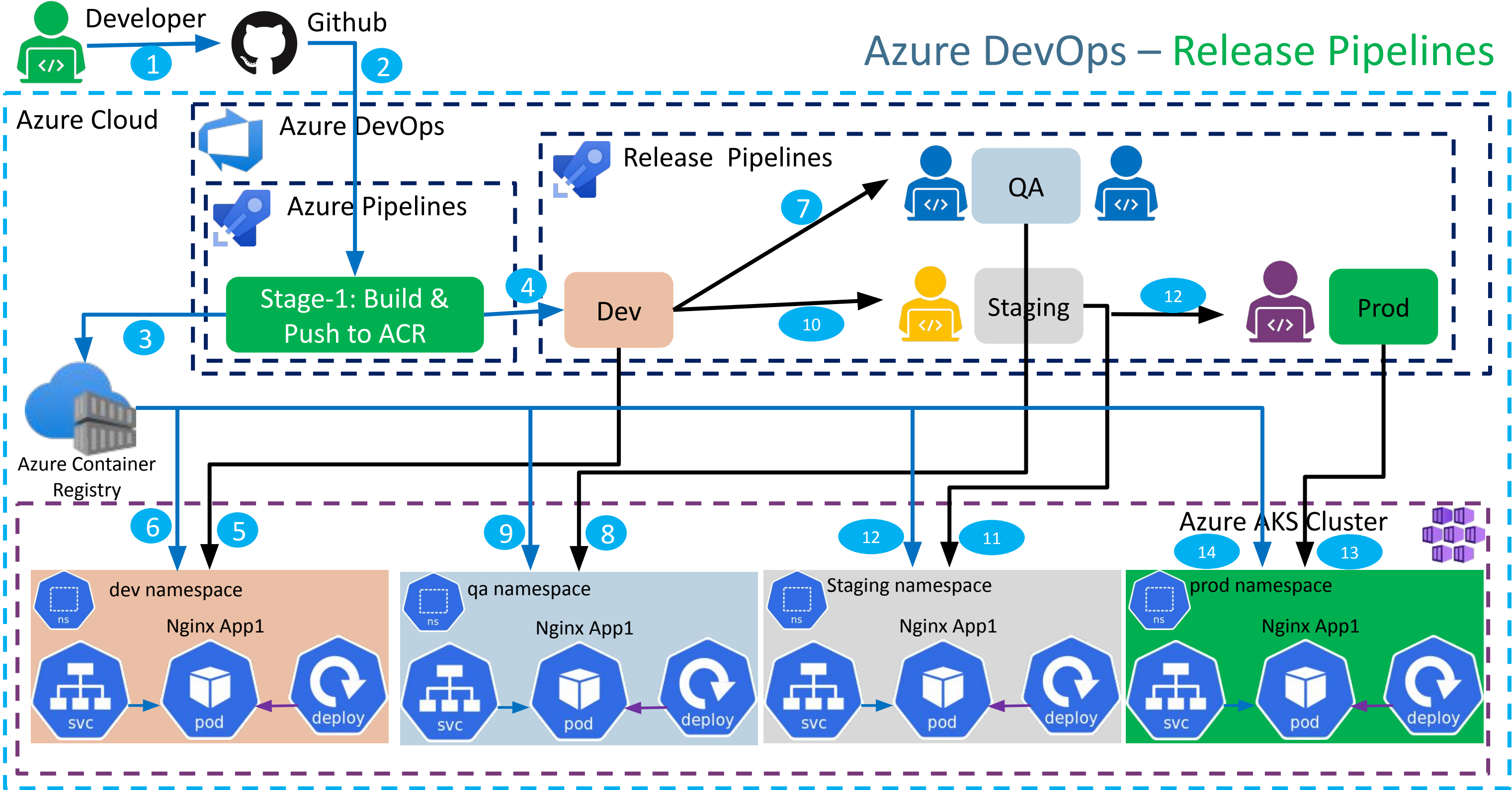
To achieve **Continuous Delivery** we use Release Pipelines



# Azure DevOps – Release Pipelines



# Azure DevOps – Release Pipelines



# Azure Release Pipelines

01-Release-Pipeline > Release-4

Pipeline Variables History | + Deploy Cancel Refresh Edit ...

## Release

### Continuous deployment

for Stack Simplify  
07/09/2020, 19:09

### Artifacts



\_03-custom-build...

20200907.8

master

## Stages

### Dev

✓ Succeeded

on 07/09/2020, 19:09

### QA

✓ Succeeded

on 07/09/2020, 19:10

### Production

✓ Succeeded


on 07/09/2020, 19:12






### staging









✓ Succeeded




on 07/09/2020, 19:10



# Azure Release Pipelines - Releases

 stacksimplify2 / azure-aks-app1-github-do... / Pipelines / Releases
















 New 























01-Release-Pipeline

 Production 

## 01-Release-Pipeline

 Edit  Create release 

 All releases 

	Releases	Deployments	Analytics
	Releases	Created	Stages
 <b>Release-4</b>  2020090...  master	07/09/2020, 19:09:02	 Dev  QA  staging  Production	
 <b>Release-3</b>  2020090...  master	07/09/2020, 19:03:32	 Dev  QA  staging  Production	
 <b>Release-2</b>  2020090...  master	07/09/2020, 18:43:39	 Dev	
 <b>Release-1</b>  2020090...  master	07/09/2020, 18:38:48	 Dev	



Azure AKS  
HTTP  
Application  
Routing  
Add On  
Ingress +  
External DNS  
(Automatic Install)

app1.7b8803340f38495c8402.centralus.aksapp.io



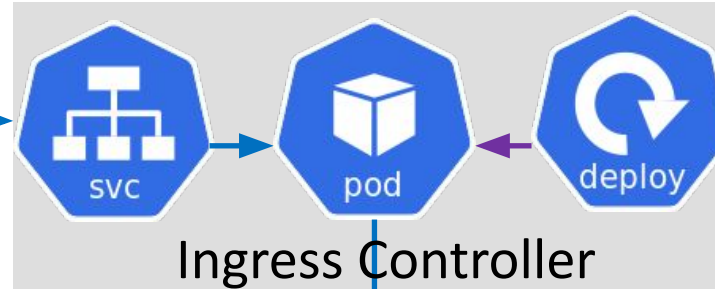
Users



Public IP  
for Ingress  
Controller



AKS  
Load Balancer



Ingress Controller

HTTP Application Routing Add On



app1.7b8803340f38495c8402.  
centralus.aksapp.io

Ingress Service

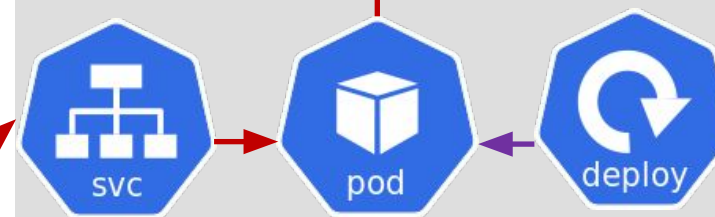
Azure Cloud



DNS Zones

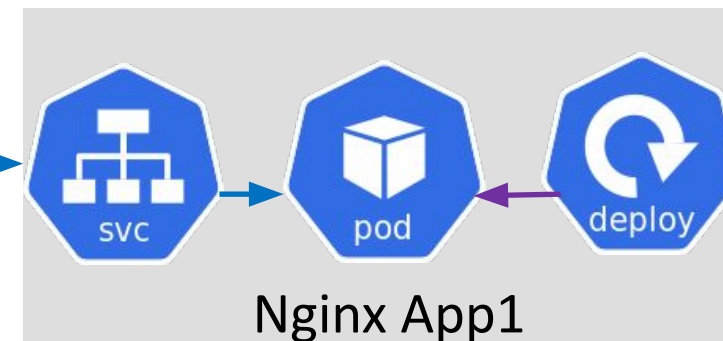
app1.7b8803340f38495c8402.centralus.aksapp.io

7b8803340f38495c8402.centralus.aksapp.io



External DNS

Azure AKS Cluster



Nginx App1



# 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

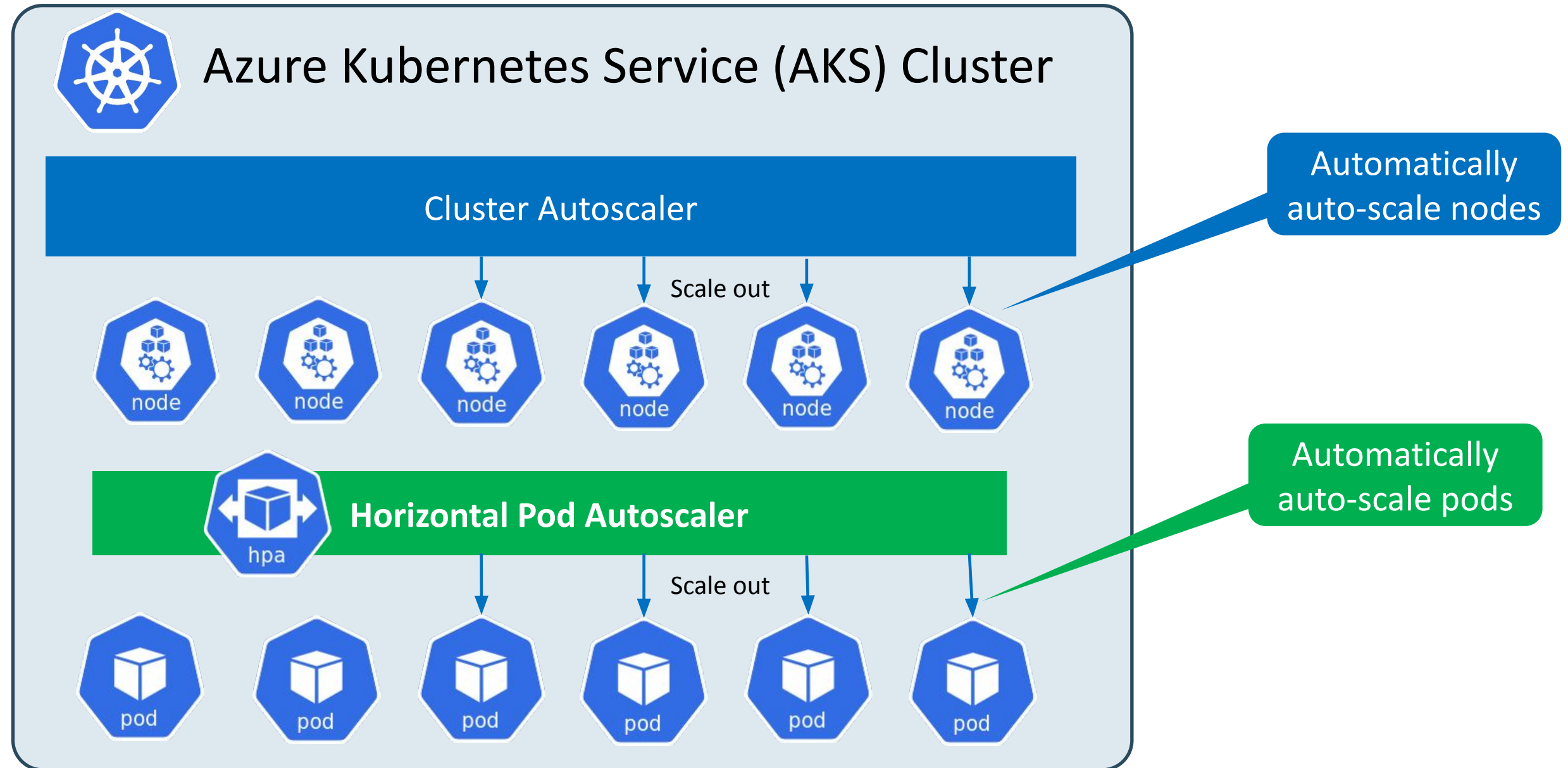
AKS Cluster - 1

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