Alper Can

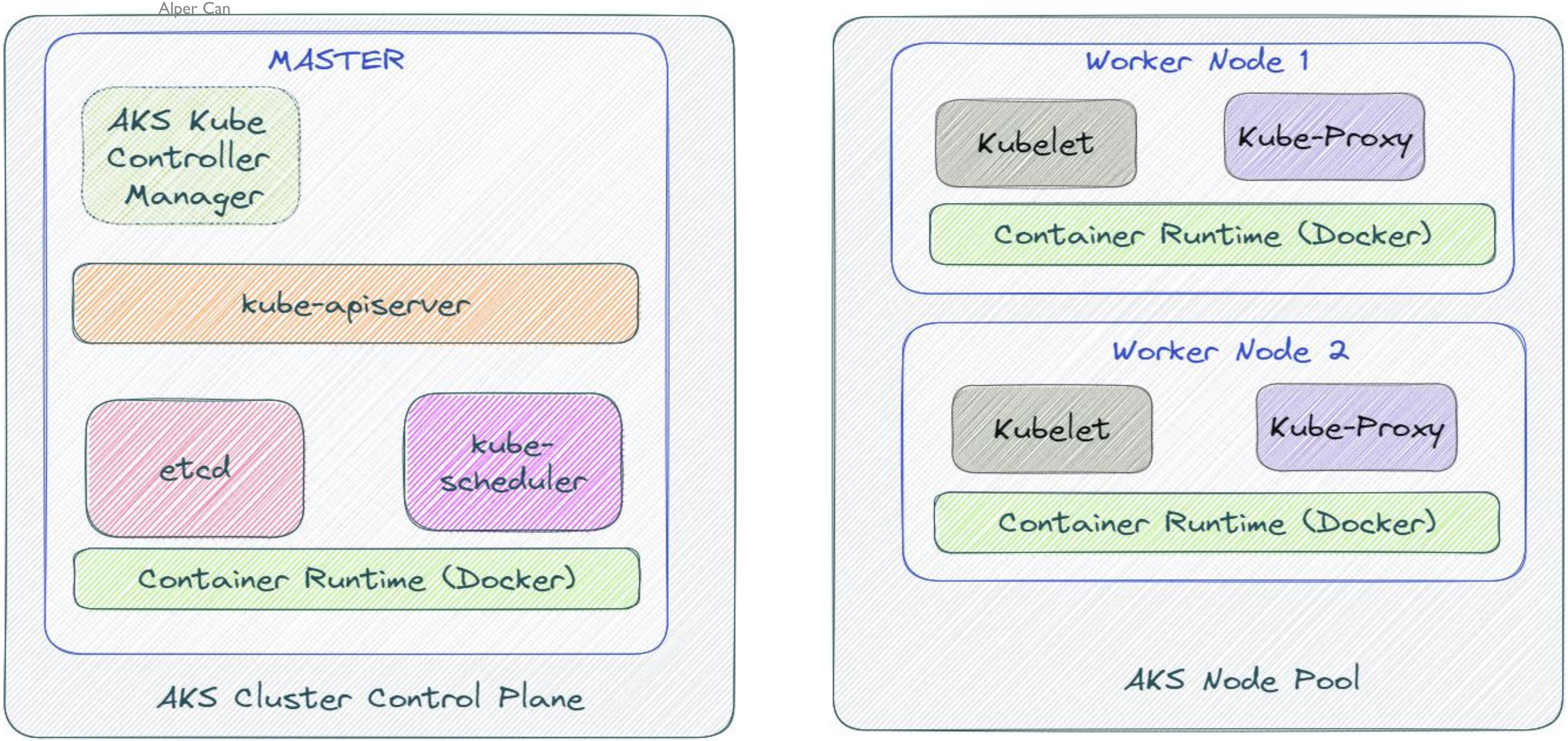
AZURE AKS Create Cluster Introduction

Alper Can

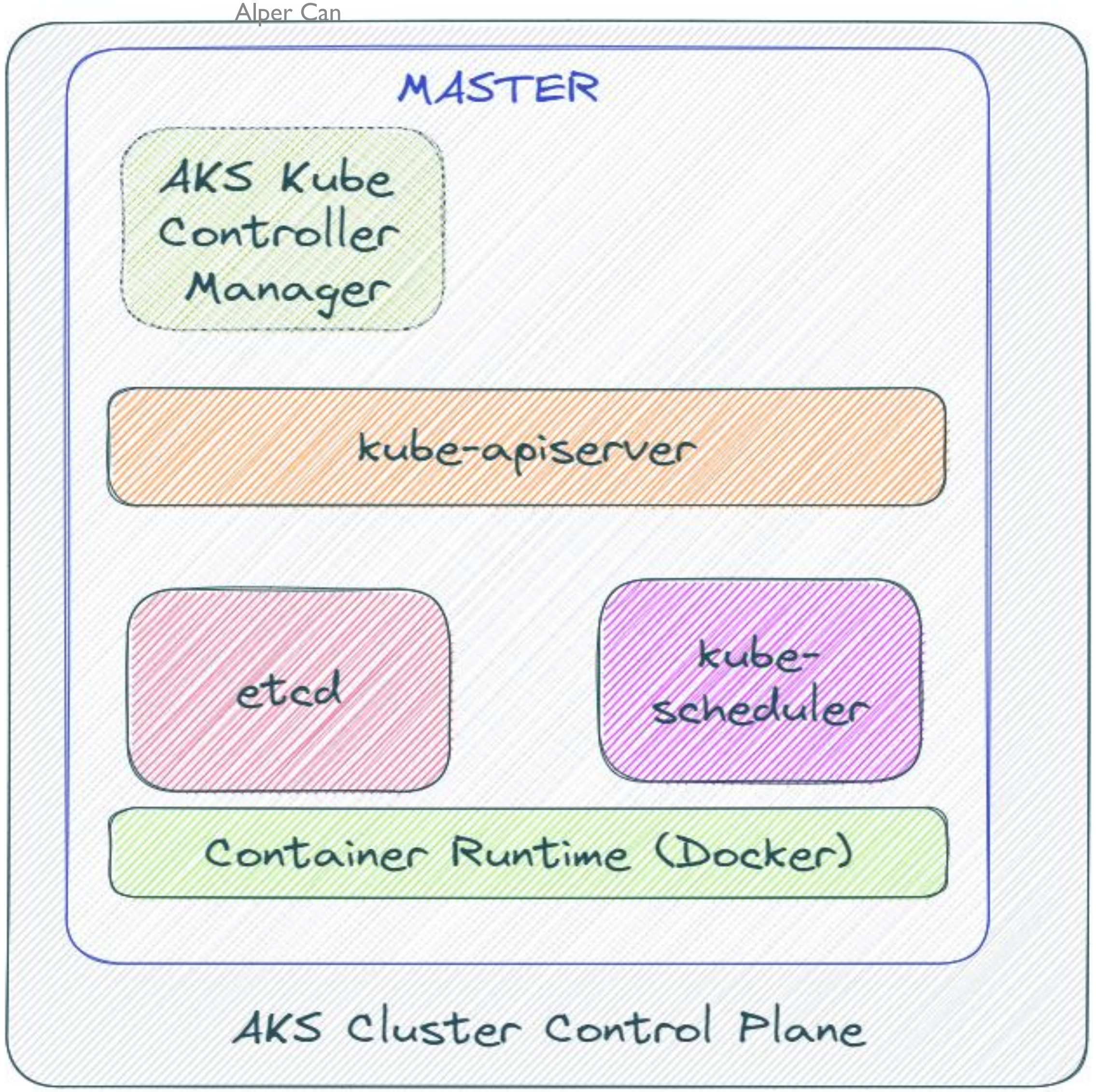
AKS - Introduction

- AKS – Azure Kubernetes Service
- AKS is highly available, secure and fully managed Service
- When compared to other cloud providers, AKS is the one which is available in highest number of regions
- Will be able to run any type of workloads
  - Windows based applications like .Net Apps
  - Linux supported applications like Java
  - IOT device deployment and management on demand
  - Machine Learning Model training with AKS
- Able to run in Hybrid Platforms
  - Azure StacK HCI
  - Windows Servers with Linux Distros
  - Planing for Vmware Platform
- Able to use Azure services without additional infra and admin effort
  - You can deploy and integrate azure services with your AKS easly
  - Azure Storage, Azure Key Vault, Azure Devops, Azure LB, etc.

Alper Can

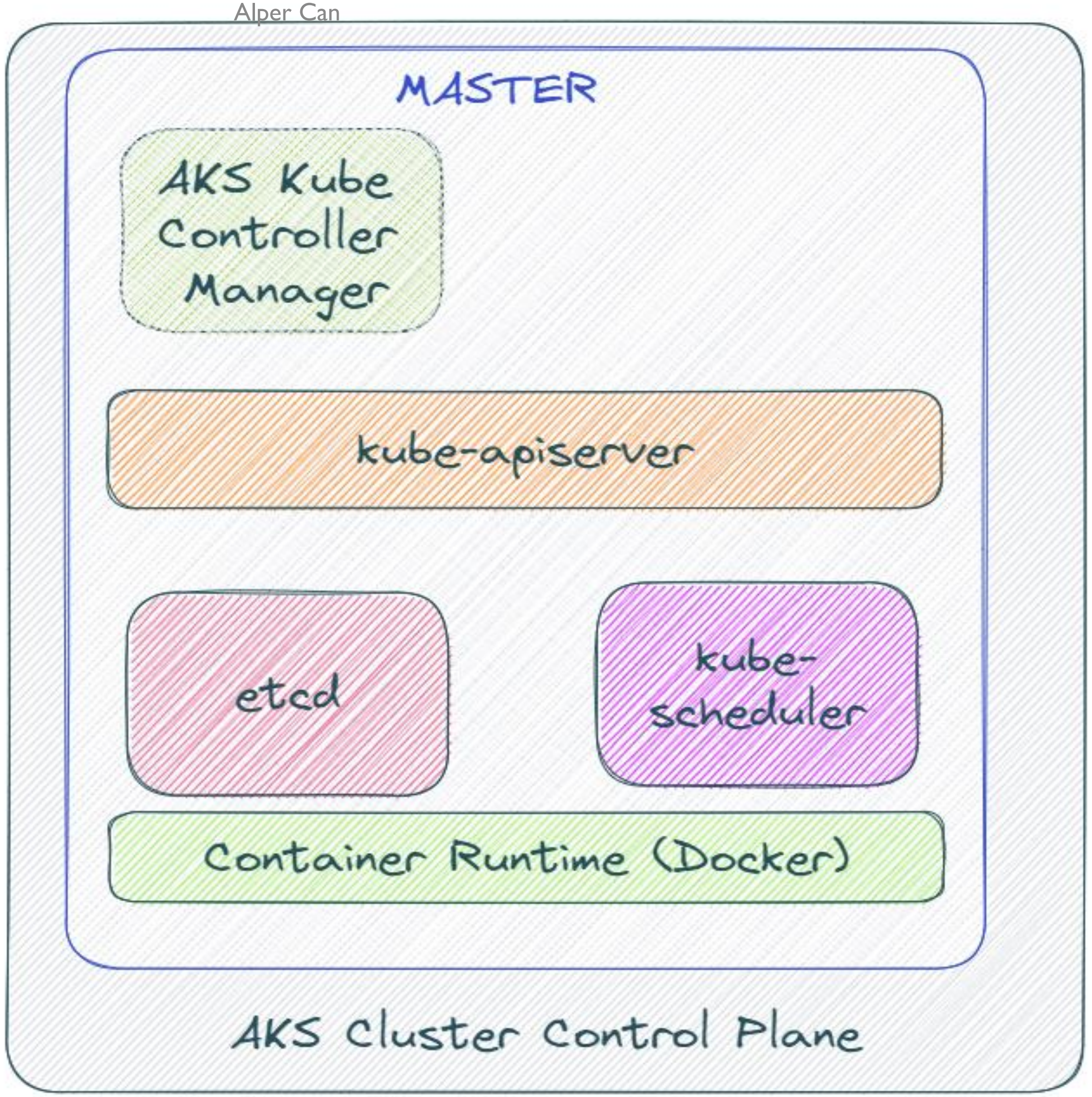

MASTER
AKS Kube Controller Manager
kube-apiserver
etcd
kube-scheduler
Container Runtime (Docker)
AKS Cluster Control Plane
Worker Node 1
Kubelet
Kube-Proxy
Container Runtime (Docker)
Worker Node 2
Kubelet
Kube-Proxy
Container Runtime (Docker)
AKS Node Pool

Alper Can


MASTER
AKS Kube Controller Manager
kube-apiserver
etcd
kube-scheduler
Container Runtime (Docker)
AKS Cluster Control Plane

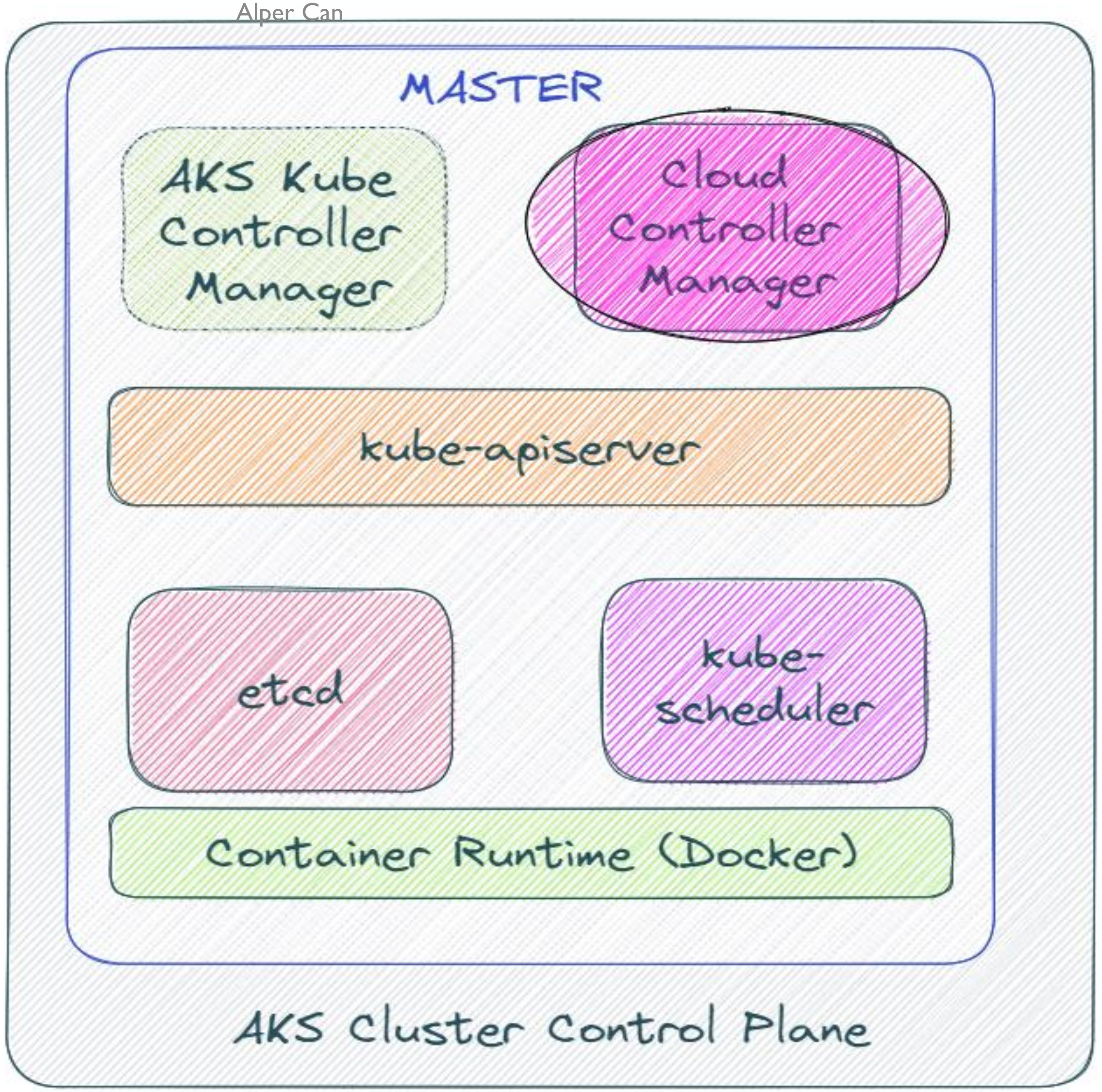
- kube-apiserver
  - It acts as front end for the Kubernetes control plane. It exposes the Kubernetes API
  - Command line tools (like kubectl), Users and even Master components (scheduler, controller manager, etcd) and Worker node components like (Kubelet) everything talk with API Server.
- etcd
  - Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data.
  - It stores all the masters and worker node information.
- kube-scheduler
  - Scheduler is responsible for distributing containers across multiple nodes.
  - It watches for newly created Pods with no assigned node, and selects a node for them to run on.

Alper Can


MASTER
AKS Kube Controller Manager
kube-apiserver
etcd
kube-scheduler
Container Runtime (Docker)
AKS Cluster Control Plane

- kube-controller-manager
  - Controllers are responsible for noticing and responding when nodes, containers or endpoints go down. They make decisions to bring up new containers in such cases.
  - Node Controller: Responsible for noticing and responding when nodes go down.
  - Replication Controller: Responsible for maintaining the correct number of pods for every replication controller object in the system.
  - Endpoints Controller: Populates the Endpoints object (that is, joins Services & Pods)
  - Service Account & Token Controller: Creates default accounts and API Access for new namespaces.

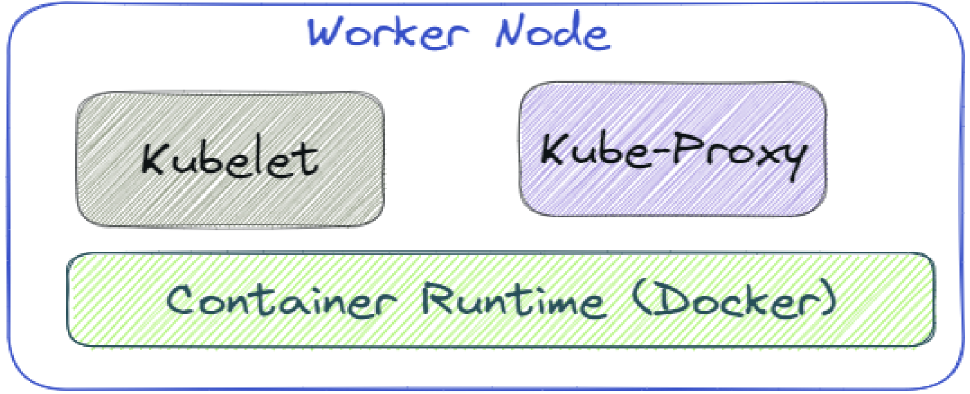
Alper Can


MASTER
AKS Kube Controller Manager
Cloud Controller Manager
kube-apiserver
etcd
kube-scheduler
Container Runtime (Docker)
AKS Cluster Control Plane

○ cloud-controller-manager

- A Kubernetes control plane component that embeds cloud-specific control logic.
- It only runs controllers that are specific to your cloud provider.
- On-Premise Kubernetes clusters will not have this component.
- Node controller: For checking the cloud provider to determine if a node has been deleted in the cloud after it stops responding.
- Route controller: For setting up routes in the underlying cloud infrastructure.
- Service controller: For creating, updating and deleting cloud provider load balancer.

Alper Can



○ Container Runtime

- Container Runtime is the underlying software where we run all these
- We are using Docker, but we have other runtime options like rkt, container-d etc.

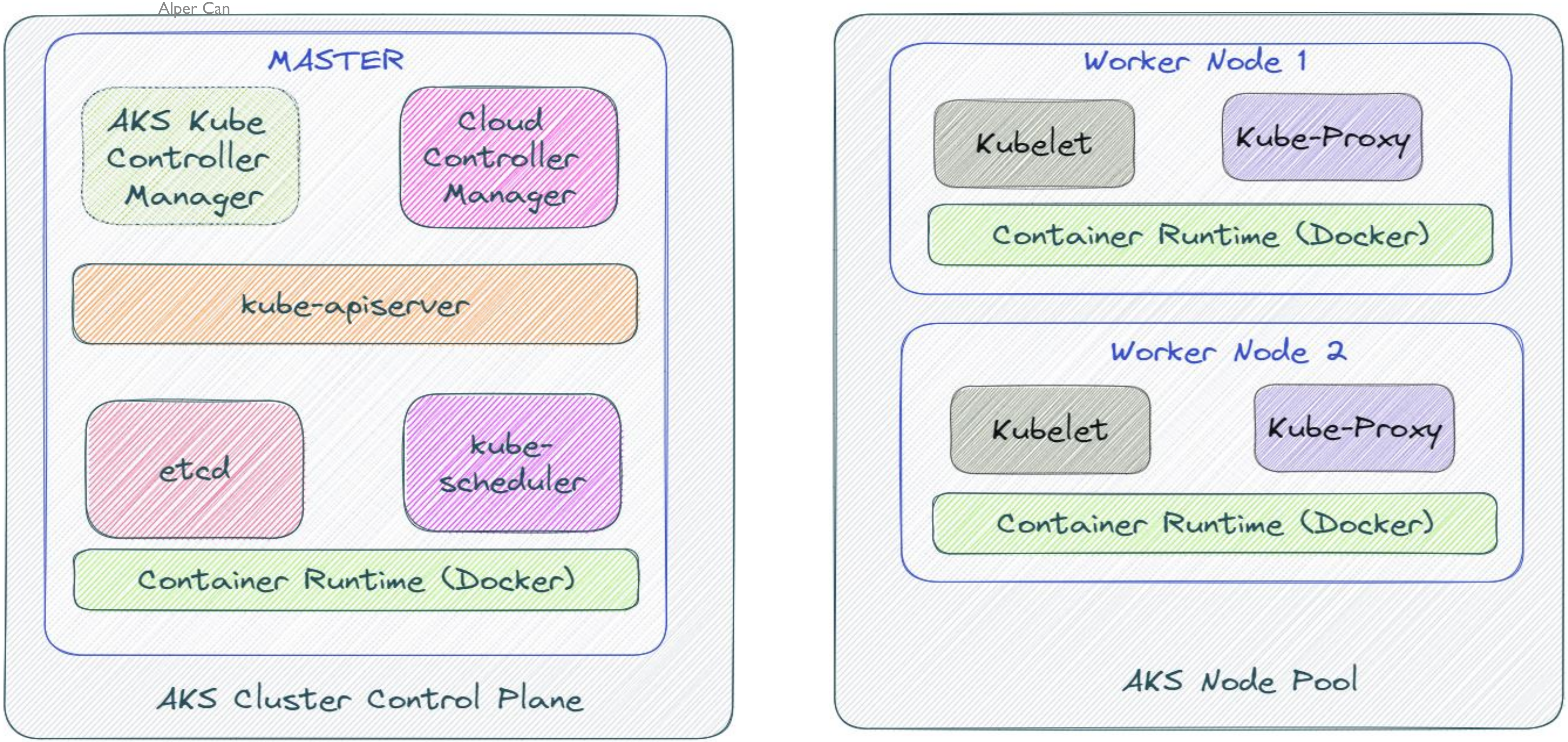
○ Kubelet

- Kubelet is the agent that runs on every node in the cluster
- This agent is responsible for making sure that containers are running in a Pod on a node.

○ Kube-Proxy

- It is a network proxy that runs on each node in your cluster.
- It maintains network rules on nodes
- In short, these network rules allow network communication to your Pods from network sessions inside or outside of your cluster.

Alper Can


MASTER
AKS Kube Controller Manager
Cloud Controller Manager
kube-apiserver
etcd
kube-scheduler
Container Runtime (Docker)
AKS Cluster Control Plane
Worker Node 1
Kubelet
Kube-Proxy
Container Runtime (Docker)
Worker Node 2
Kubelet
Kube-Proxy
Container Runtime (Docker)
AKS Node Pool

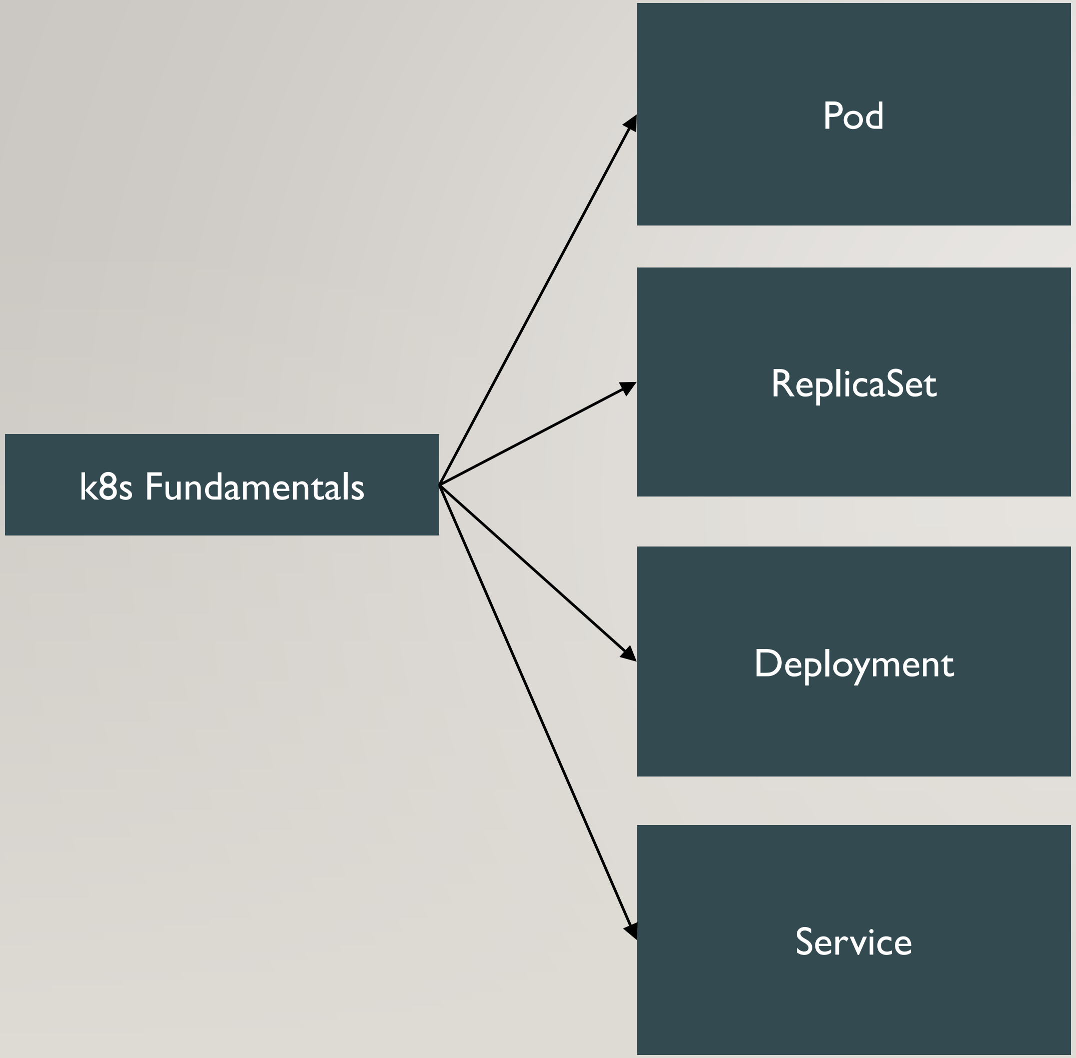
Kubernetes

Fundamentals

Pod, ReplicaSet, Deployment & Service

Alper Can

KUBERNETES - FUNDAMENTALS



k8s Fundamentals

Pod

ReplicaSet

Deployment

Service

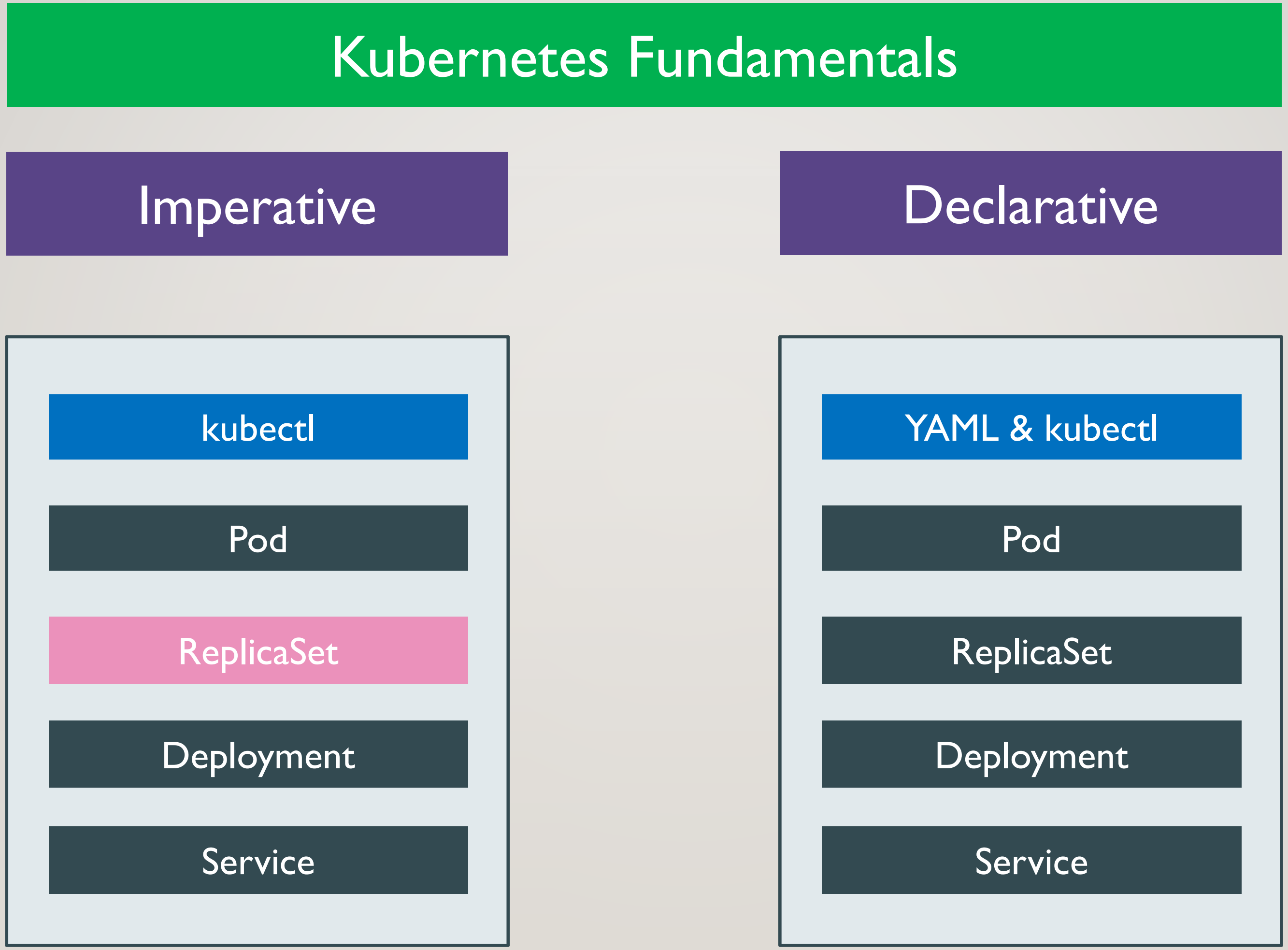
A POD is a single instance of an Application.
A POD is the smallest object, that you can create in Kubernetes.

A ReplicaSet will maintain a stable set of replica Pods running at any given time.
In short, it is often used to guarantee the availability of a specified number of identical Pods

A Deployment runs multiple replicas of your application and automatically replaces any instances that fail or become unresponsive. Rollout & rollback changes to applications. Deployments are well-suited for stateless applications.

A service is an abstraction for pods, providing a stable, so called virtual IP (VIP) address.
In simple terms, service sits Infront of a POD and acts as a load balancer.

KUBERNETES - IMPERATIVE & DECLARATIVE



Kubernetes Fundamentals

Imperative

kubectl

Pod

ReplicaSet

Deployment

Service

Declarative

YAML & kubectl

Pod

ReplicaSet

Deployment

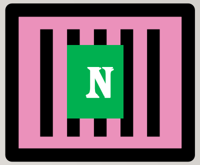
Service

Kubernetes

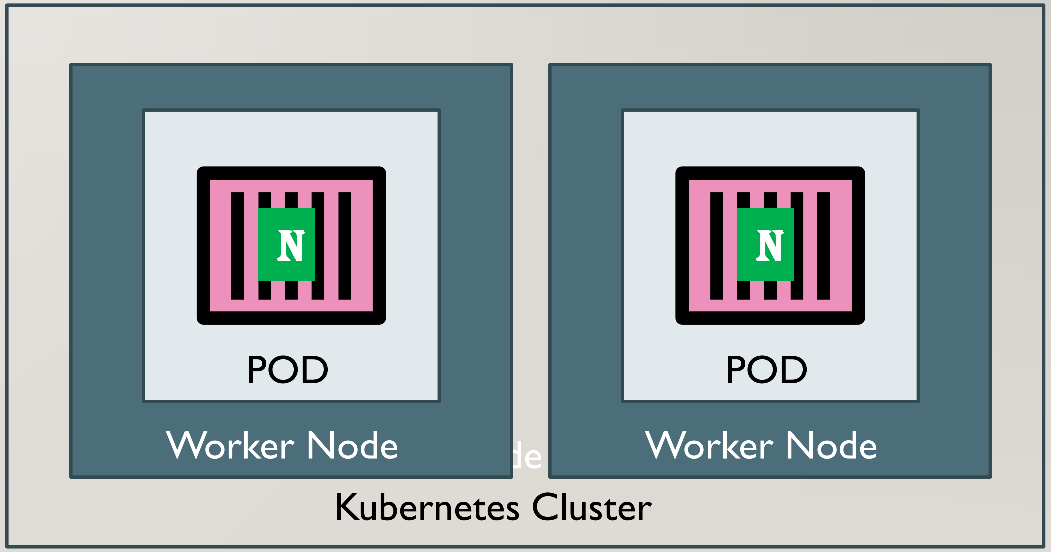
POD

KUBERNETES - POD

- With Kubernetes our core goal will be to deploy our applications in the form of containers on worker nodes in a k8s cluster.
- Kubernetes does not deploy containers directly on the worker nodes.
- Container is encapsulated in to a Kubernetes Object named POD.
- A POD is a single instance of an application.
- A POD is the smallest object that we can create in Kubernetes.

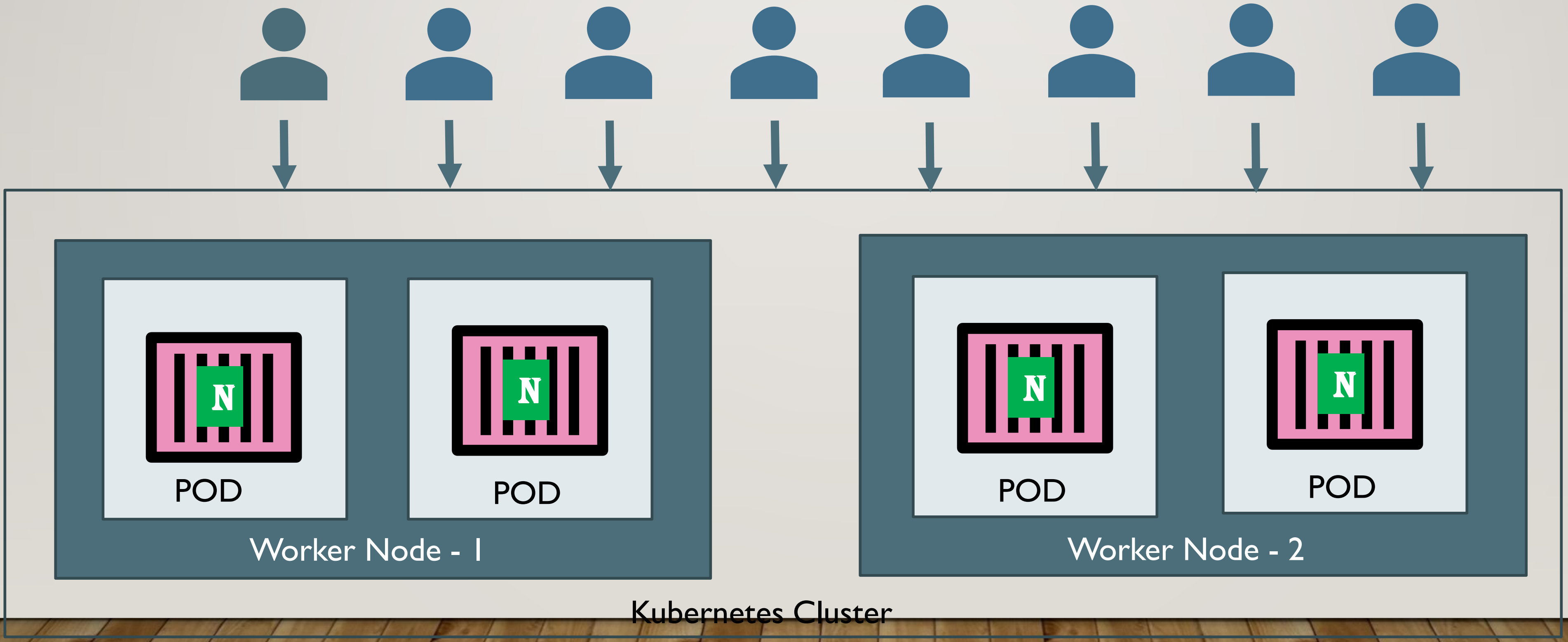


Nginx Container Image



KUBERNETES - POD

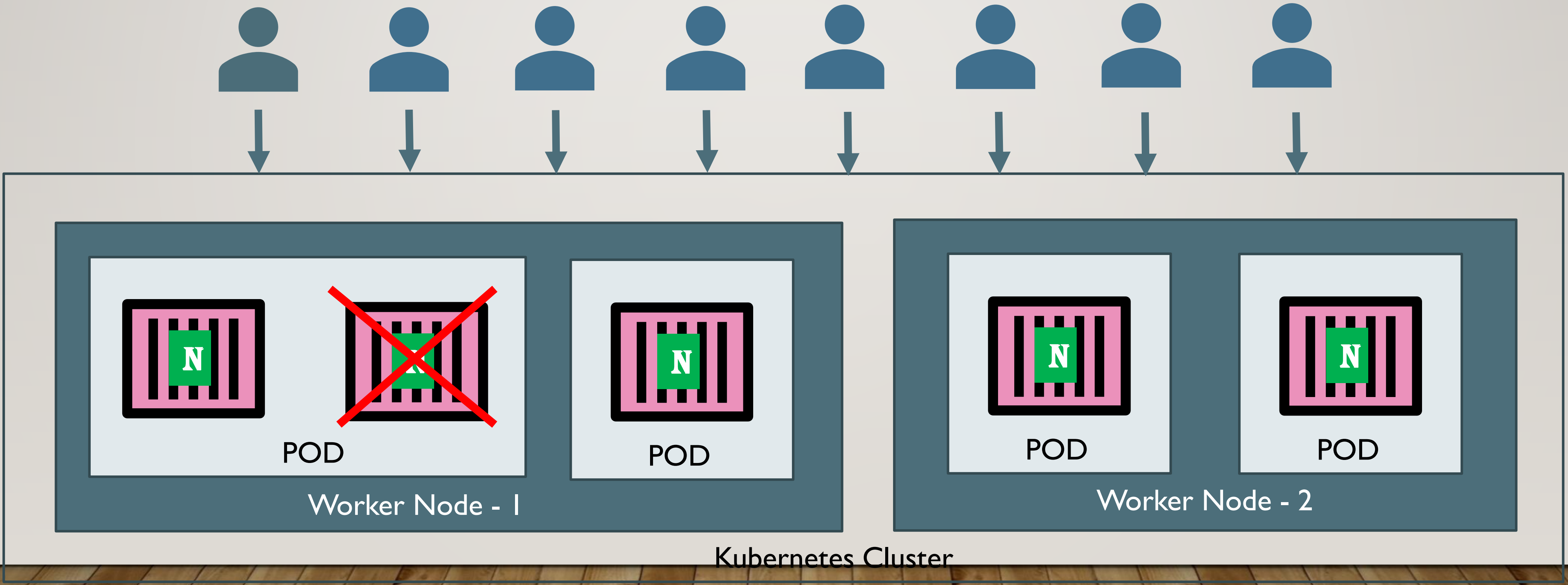
- PODs generally have one to one relationship with containers.
- To scale up we create new POD and to scale down we delete the POD.



Alper Can

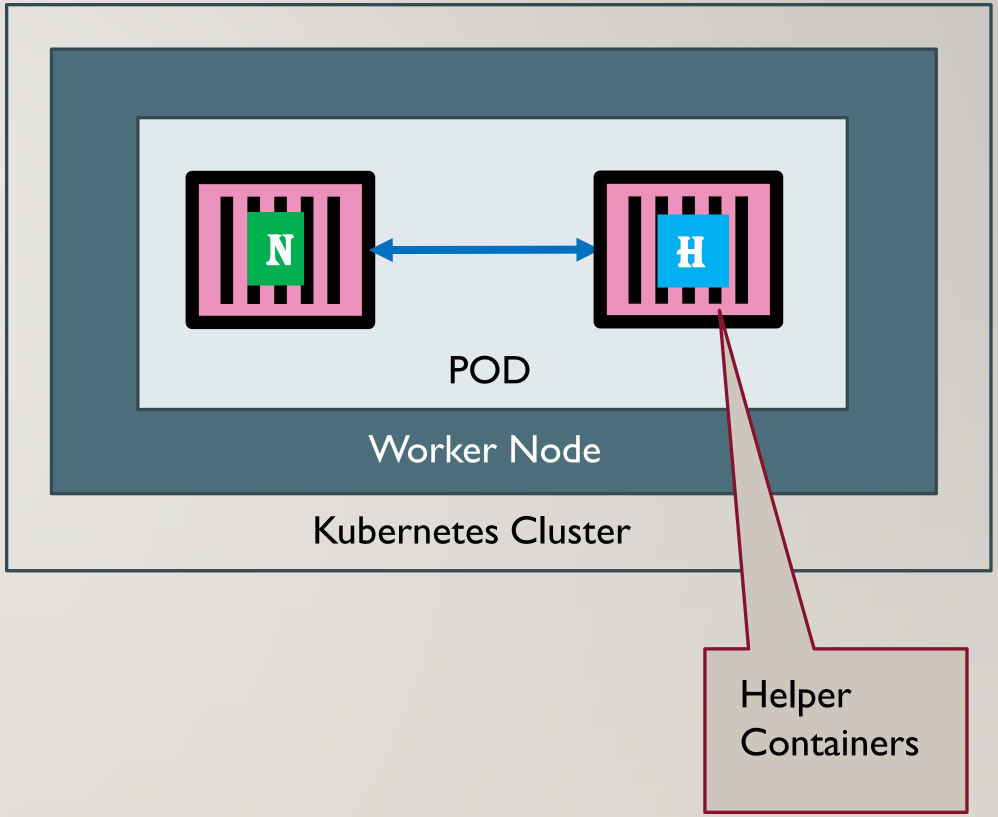
KUBERNETES – PODS

- We cannot have multiple containers of same kind in a single POD.
- Example: Two NGINX containers in single POD serving same purpose is not recommended.



KUBERNETES – MULTI-CONTAINER PODS

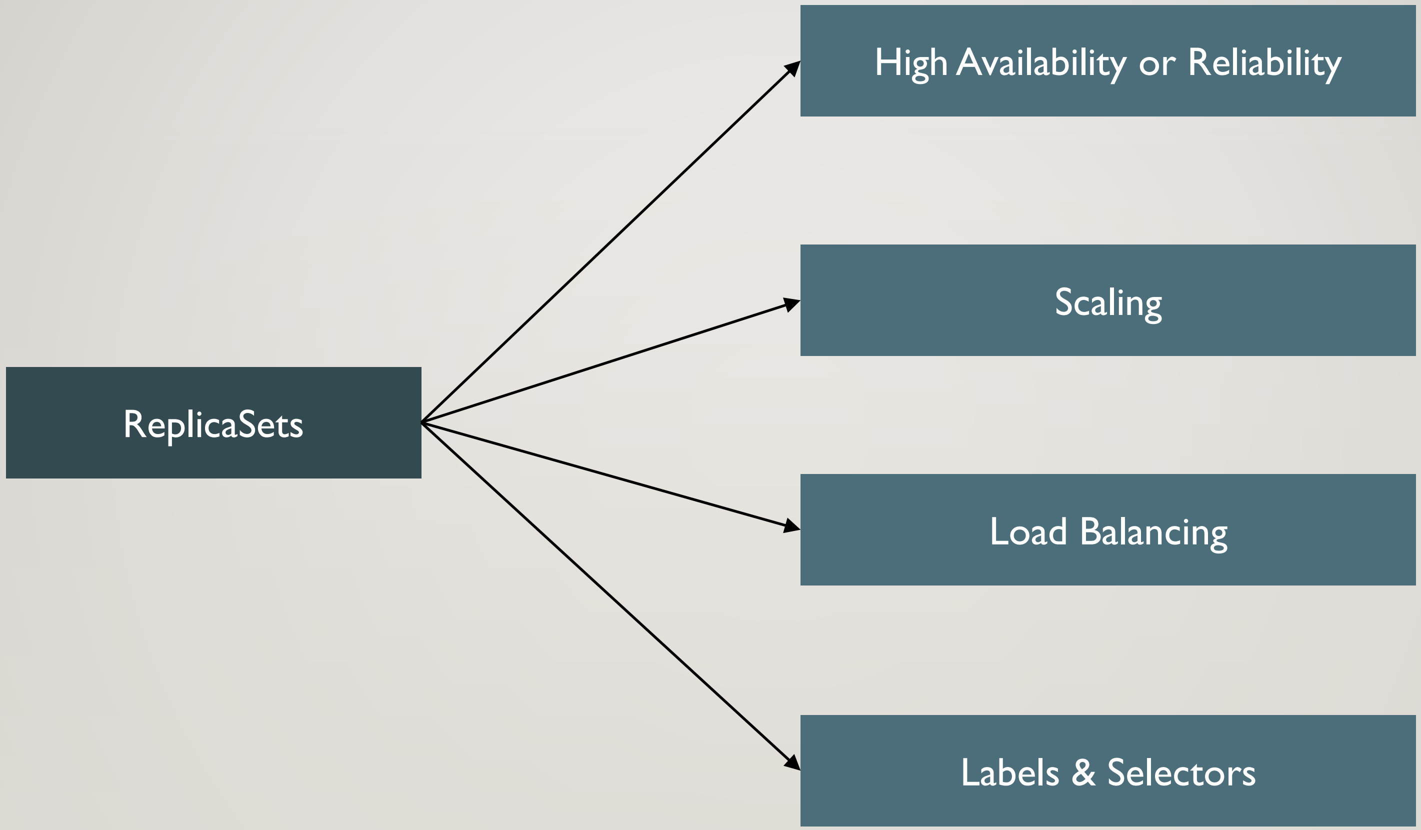
- We can have multiple containers in a single POD, provided they are not of same kind.
- Helper Containers (Side-car)
  - Data Pullers: Pull data required by Main Container
  - Data pushers: Push data by collecting from main container (logs)
  - Proxies: Writes static data to html files using Helper container and Reads using Main Container.
- Communication
  - The two containers can easily communicate with each other Node easily as they share same network space.
  - They can also easily share same storage space.
- Multi-Container Pods is a rare use-case and we will try to focus on core fundamentals.



Kubernetes

ReplicaSets

KUBERNETES - REPLICASETS



ReplicaSets

High Availability or Reliability

Scaling

Load Balancing

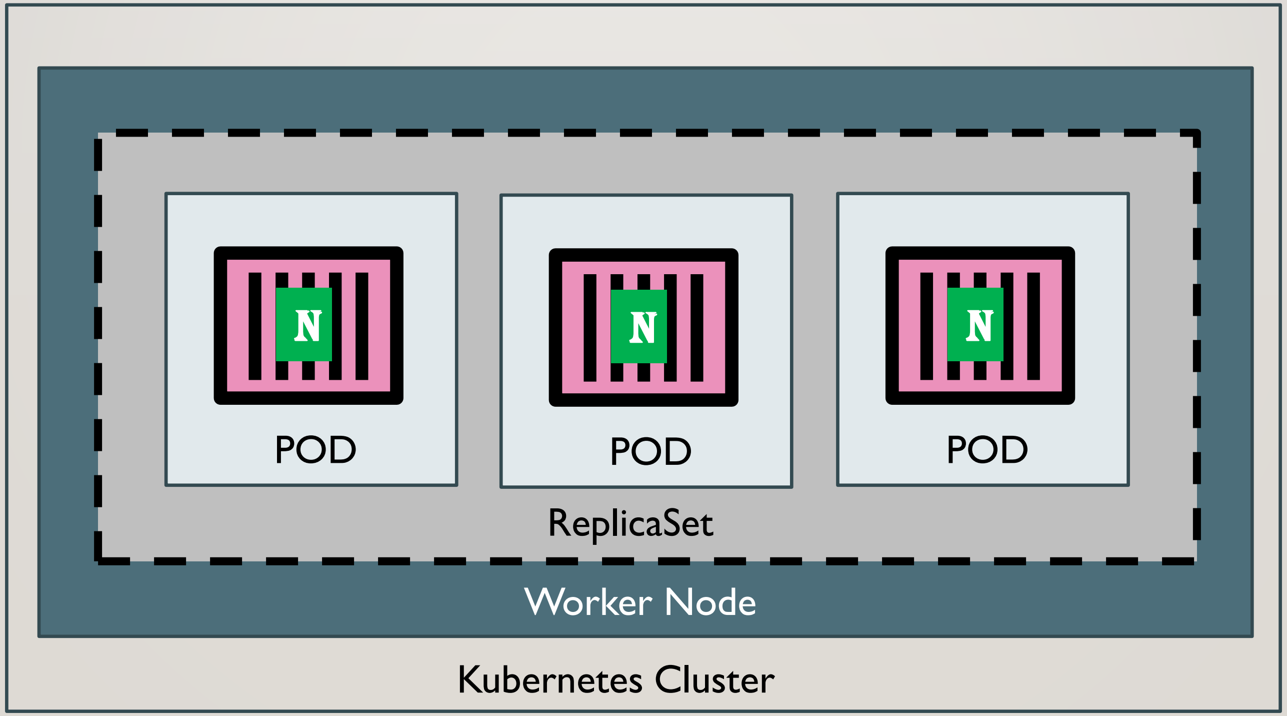
Labels & Selectors

KUBERNETES – REPLICASET

- A ReplicaSet's purpose is to maintain a stable set of replica Pods running at any given time.

- If our application crashes (any pod dies), replicaset will recreate the pod immediately to ensure the configured number of pods running at any given time.

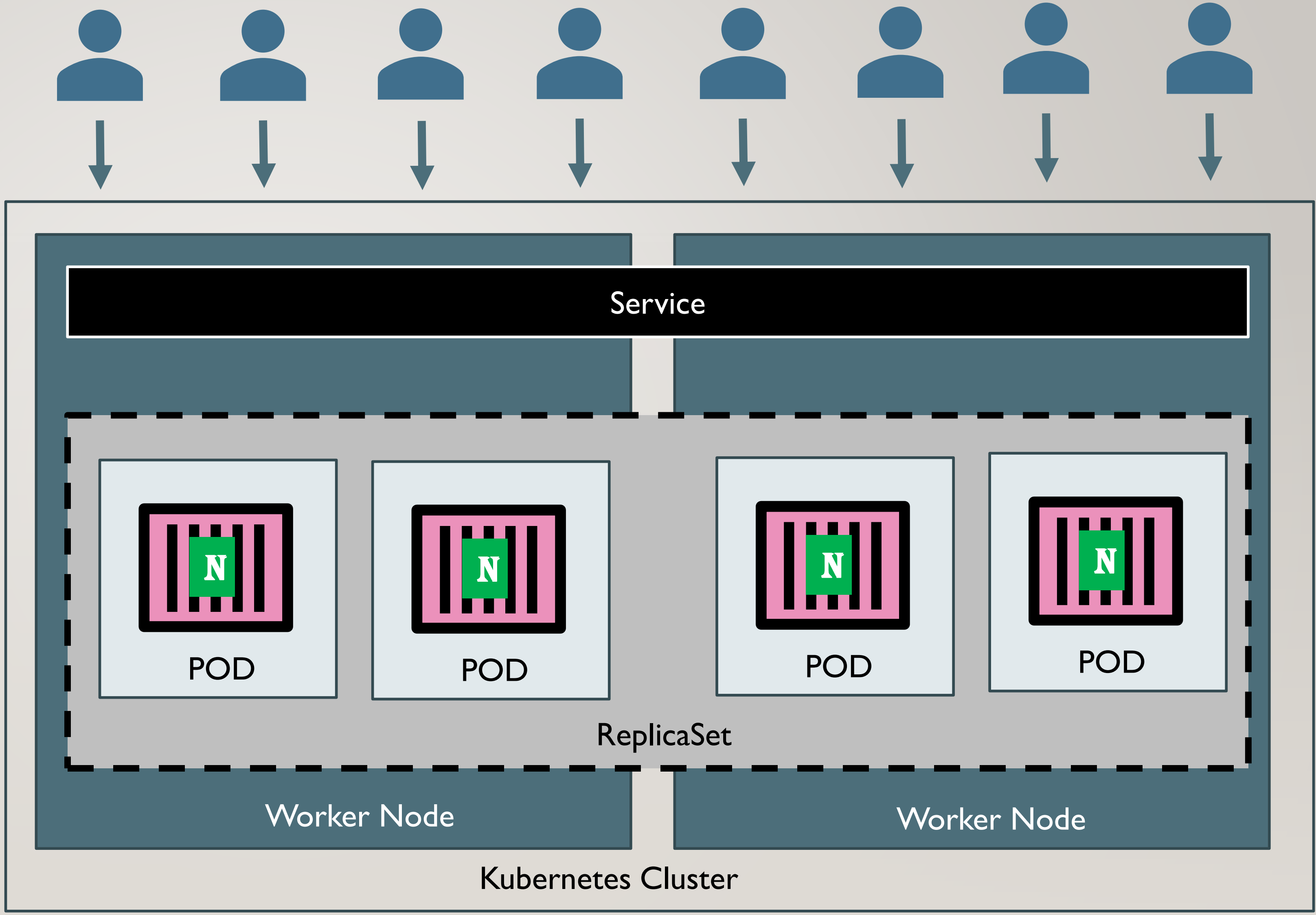
Reliability
Or
High Availability



Alper Can

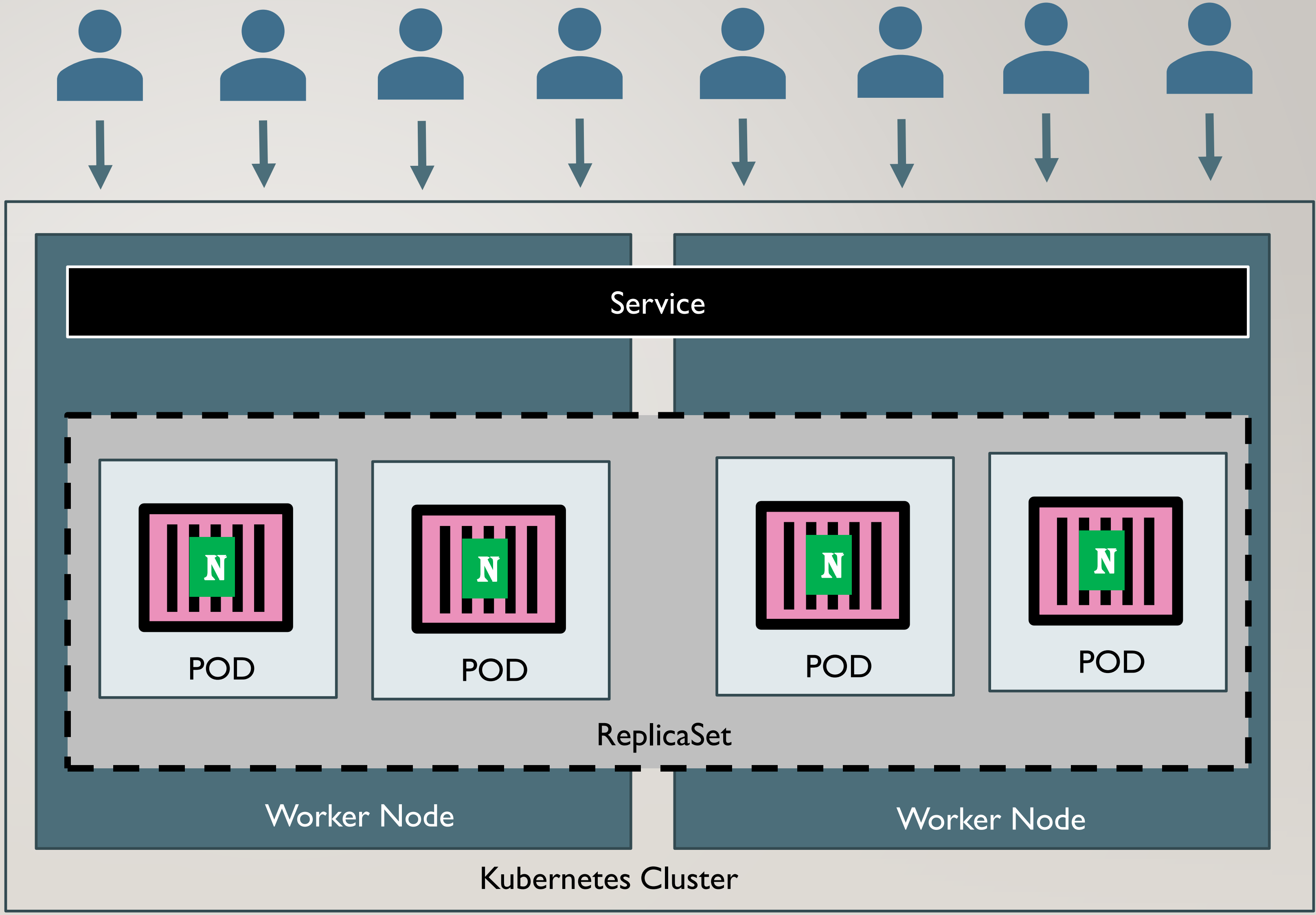
KUBERNETES – REPLICASET

- Load Balancing
- To avoid overloading of traffic to single pod we can use load balancing.
- Kubernetes provides pod load balancing out of the box using Services for the pods which are part of a ReplicaSet
- Labels & Selectors are the key items which ties all 3 together (Pod, ReplicaSet & Service), we will know in detail when we are writing YAML manifests for these objects



KUBERNETES – REPLICASET

- Scaling
- When load become too much for the number of existing pods, Kubernetes enables us to easily scale up our application, adding additional pods as needed.
- This is going to be seamless and super quick.



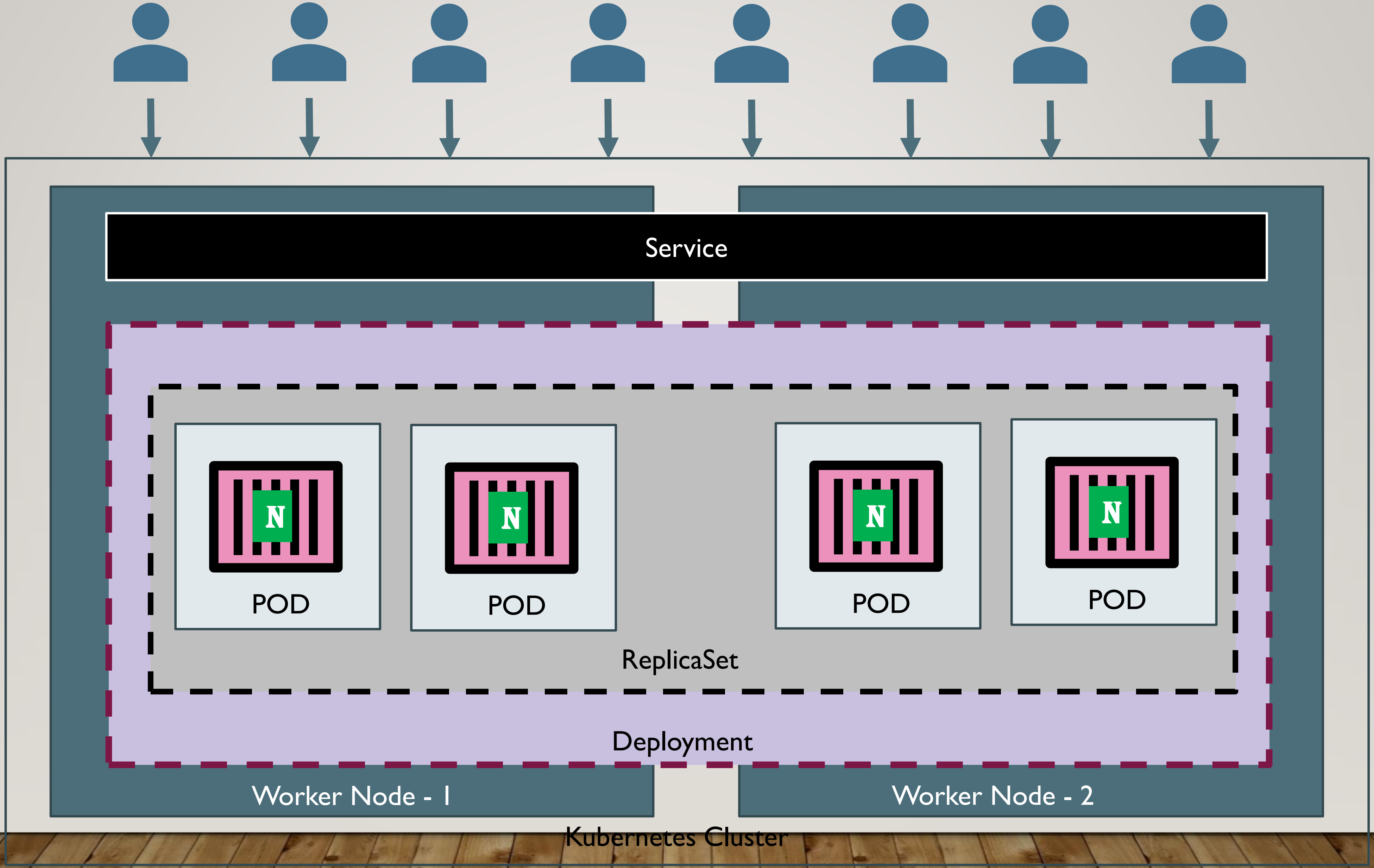
Kubernetes ReplicaSets Demo

Kubernetes

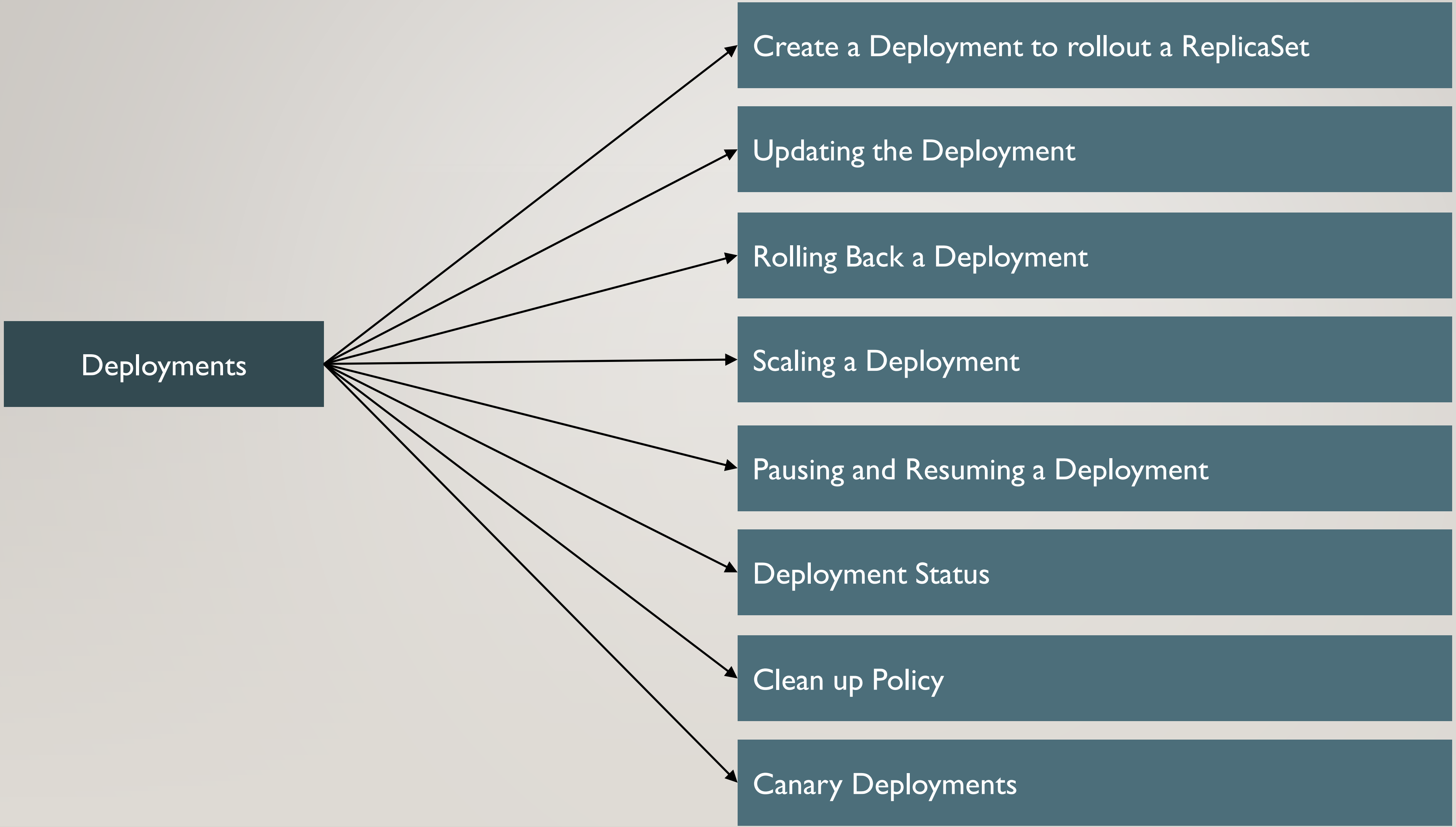
Deployments

Kubernetes
Deployments

KUBERNETES – DEPLOYMENTS



KUBERNETES - DEPLOYMENT



Deployments

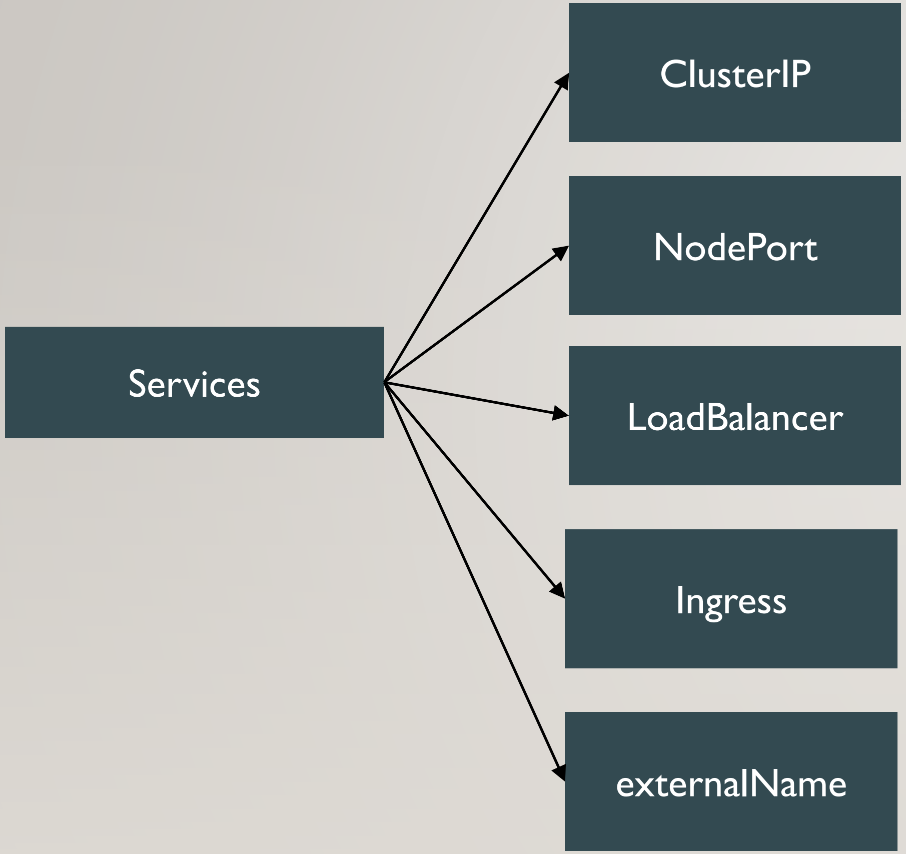
- Create a Deployment to rollout a ReplicaSet
- Updating the Deployment
- Rolling Back a Deployment
- Scaling a Deployment
- Pausing and Resuming a Deployment
- Deployment Status
- Clean up Policy
- Canary Deployments

Kubernetes Deployments Demo

Kubernetes

Services

KUBERNETES - SERVICES



Services

ClusterIP

Used for communication between applications inside k8s cluster (Example: Frontend application accessing backend application)

NodePort

Used for accessing applications outside of of k8s cluster using Worker Node Ports (Example: Accessing Frontend application on browser)

LoadBalancer

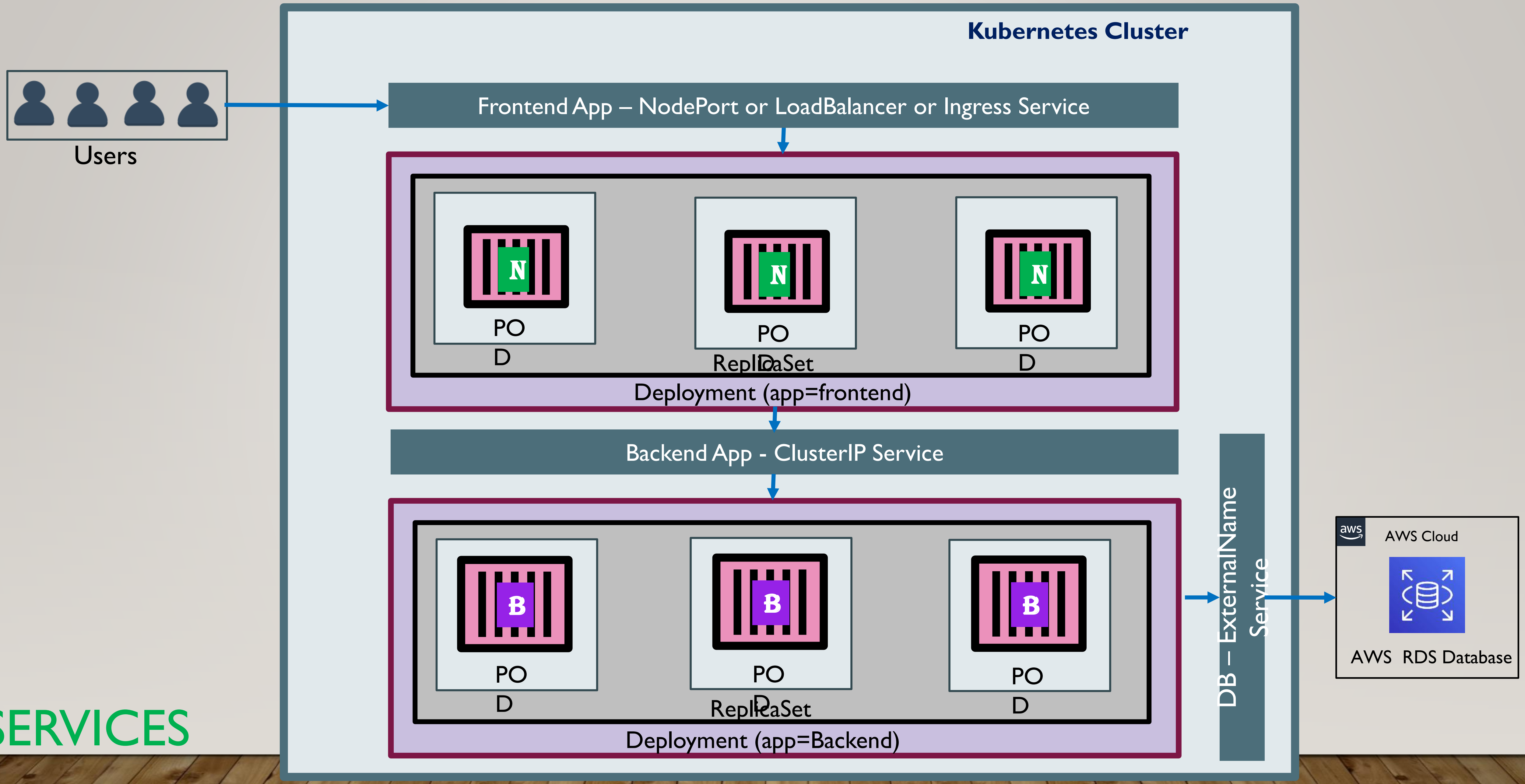
Primarily for Cloud Providers to integrate with their Load Balancer services (Example: AWS Elastic Load Balancer)

Ingress

Ingress is an advanced load balancer which provides Context path based routing, SSL, SSL Redirect and many more (Example: AWS ALB)

externalName

To access externally hosted apps in k8s cluster (Example: Access AWS RDS Database endpoint by application present inside k8s cluster)



Kubernetes Cluster

Users

Frontend App – NodePort or LoadBalancer or Ingress Service

PO
D

PO
D

PO
D

ReplicaSet

Deployment (app=frontend)

Backend App - ClusterIP Service

PO
D

PO
D

PO
D

ReplicaSet

Deployment (app=Backend)

DB – ExternalName Service

aws

AWS Cloud

AWS RDS Database

SERVICES

Kubernetes

Services

Demo

Kubernetes

YAML Basics

YAML BASICS

- YAML is used to store information about different things
- We can use YAML to define key, Value pairs like variables, lists and objects
- YAML is very similar to JSON (Javascript Object Notation)
- YAML primarily focuses on readability and user friendliness
- YAML is designed to be clean and easy to read
- We can define YAML files with two different extensions
  - abc.yml
  - abc.yaml

YAML BASICS

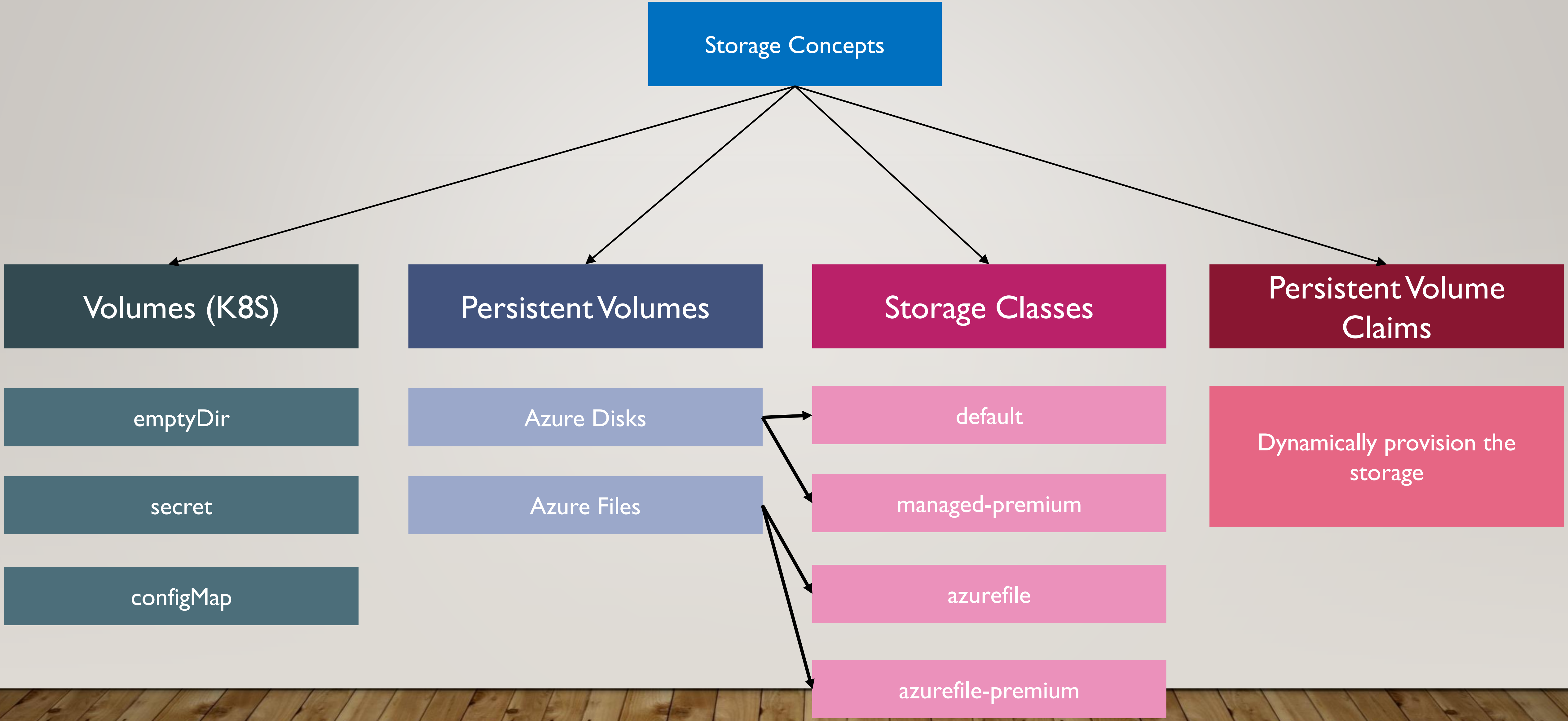
- YAML Comments
- YAML Key Value Pairs
- YAML Dictionary or Map
- YAML Array / Lists
- YAML Spaces
- YAML Document Separator

Azure AKS Storage

with

Azure Disks & Files

AKS STORAGE CONCEPTS



Storage Concepts

Volumes (K8S)

- emptyDir
- secret
- configMap

Persistent Volumes

- Azure Disks
- Azure Files

Storage Classes

- default
- managed-premium
- azurefile
- azurefile-premium

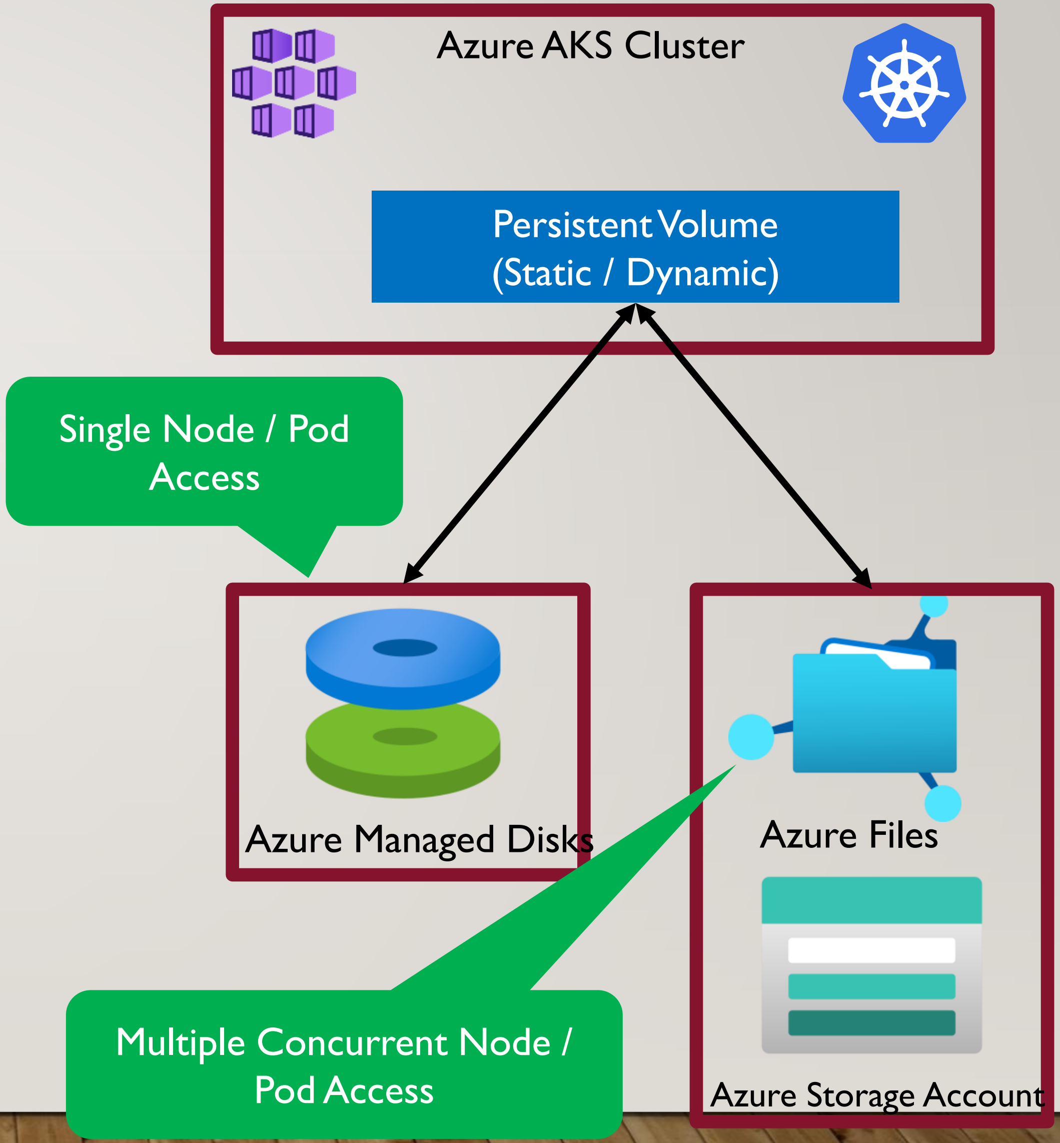
Persistent Volume Claims

- Dynamically provision the storage

Alper Can

WHY DO WE NEED PERSISTENT STORAGE IN KUBERNETES?

- Volumes that are defined and created as part of the pod lifecycle only exist until the pod is deleted.
- Pods often expect their storage to remain if a pod is rescheduled on a different host during a maintenance event, especially in StatefulSets.
- A *persistent volume* (PV) is a storage resource created and managed by the Kubernetes API that can exist beyond the lifetime of an individual pod.
- For AKS, Azure Disks or Azure Files are used to provide the PersistentVolume.



Alper Can

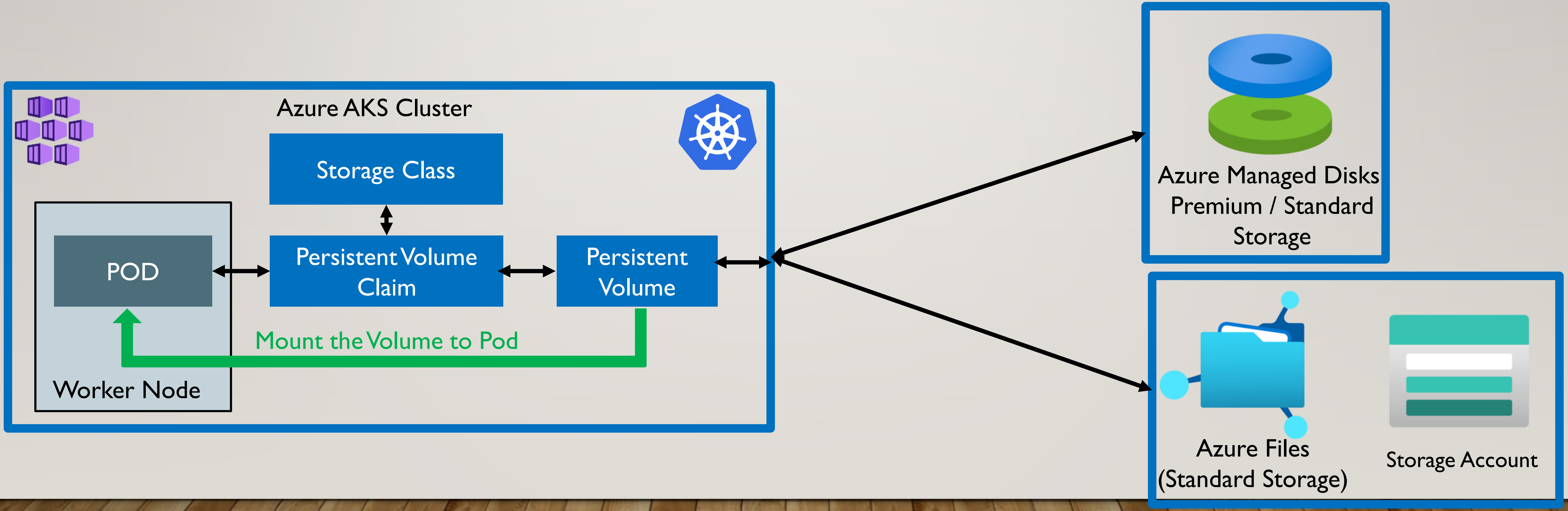
PERSISTENT VOLUME CLAIMS – HOW IT WORKS?

create a pod

PVC will request storage based on specifics from Storage class (Disk or File)

Kube API Server will dynamically provision the Persistent Volume PV requested

Once the PV is available it will get mounted to Pod for usage.



Alper Can

Azure AKS Storage

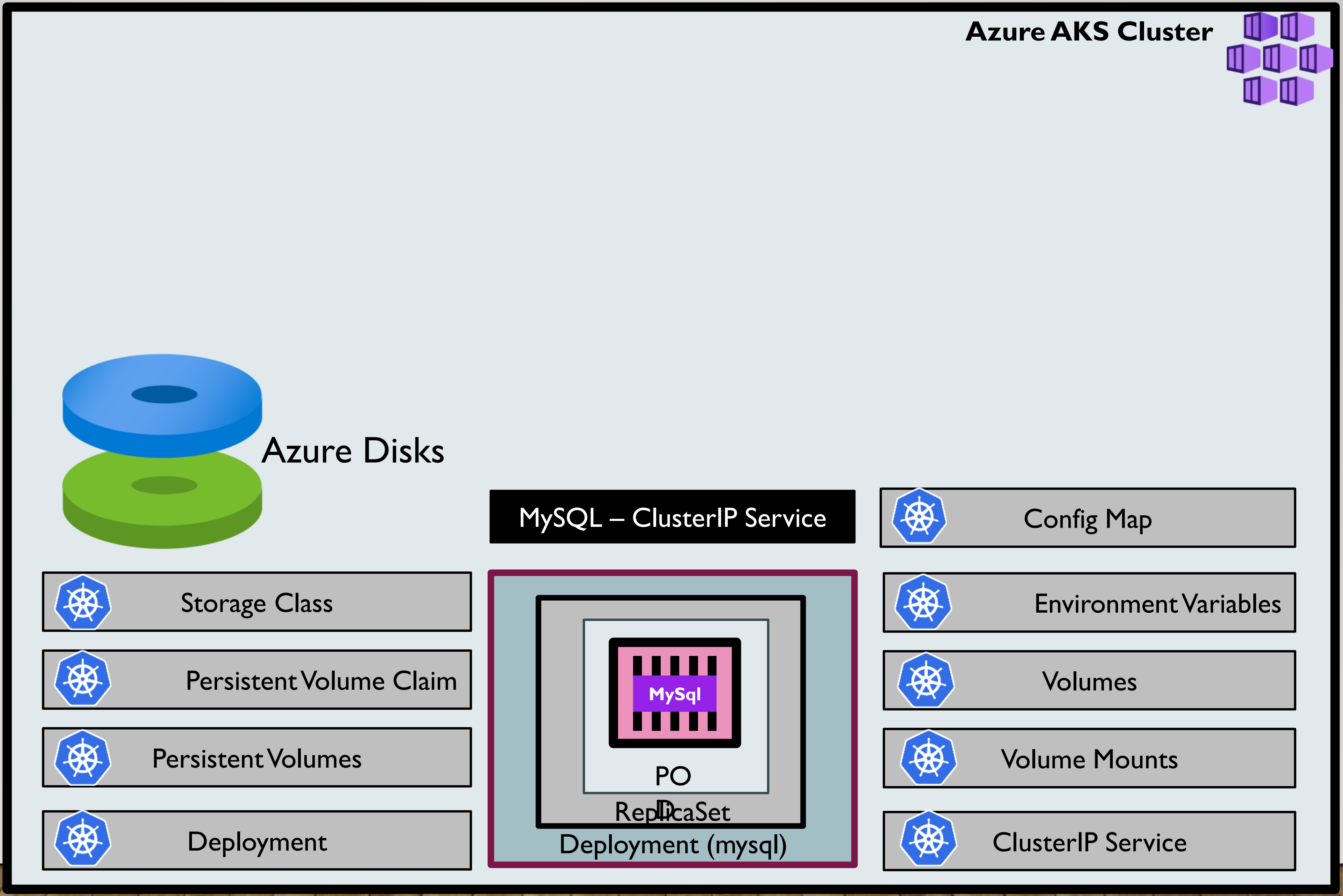
with

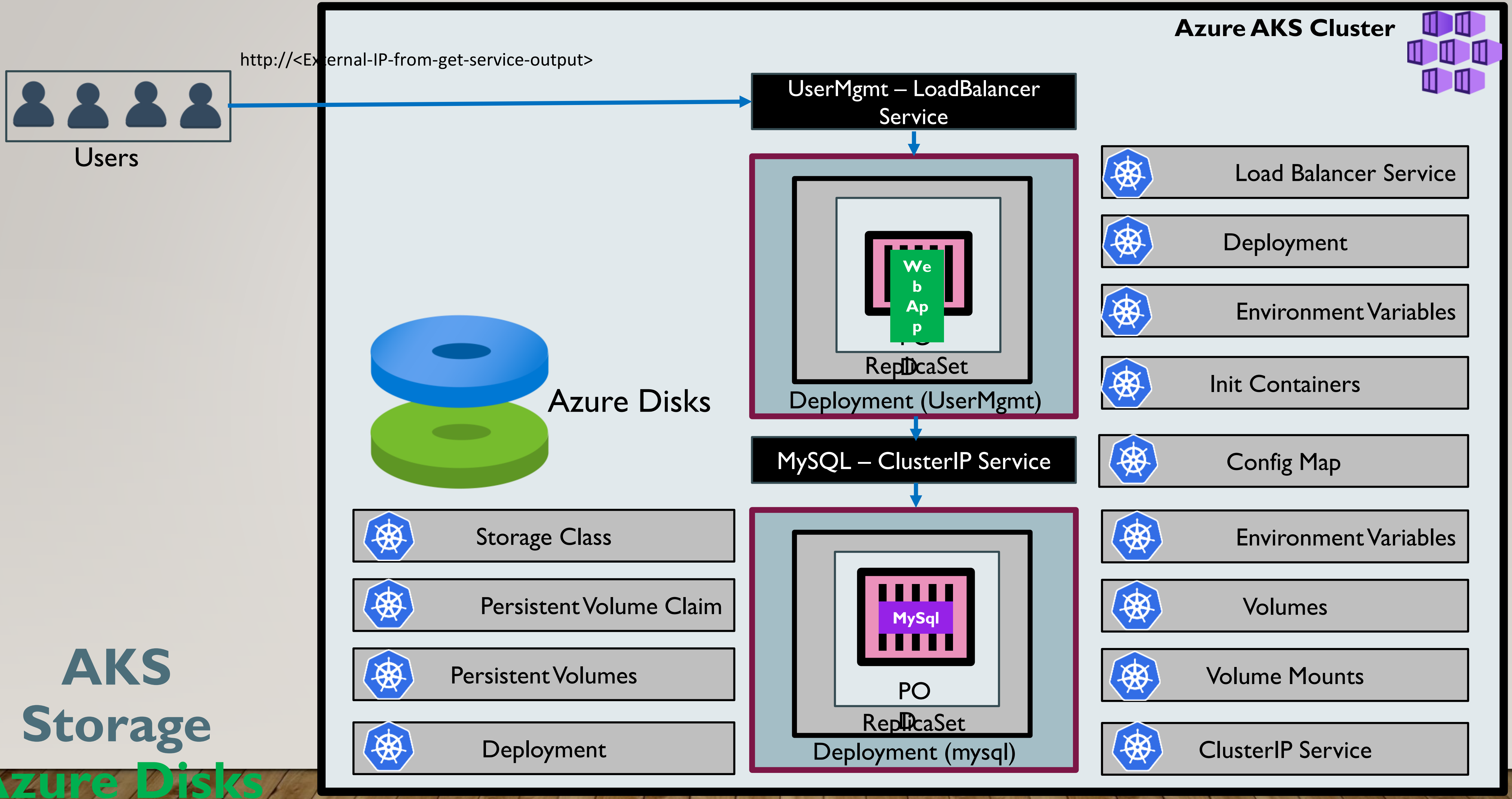
Azure Disks

AZURE DISKS - INTRODUCTION

- Azure Disk Storage offers high-performance, highly durable block storage for our mission- and business-critical workloads
- We can mount these volumes as devices on our Virtual Machines & Container instances.
- **Cost-effective storage**
  - Built-in bursting capabilities to handle unexpected traffic and process batch jobs cost-effectively
- **Unmatched resiliency**
  - 0 percent annual failure rate for consistent enterprise-grade durability
- **Seamless scalability**
  - Dynamic scaling of disk performance on Ultra Disk Storage without disruption
- **Built-in security**
  - Automatic encryption to help protect your data using Microsoft-managed keys or your own

AKS STORAGE AZURE DISKS





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

Users

Azure AKS Cluster

UserMgmt – LoadBalancer Service

We
b
Ap
p

PO
D

ReplicaSet

Deployment (UserMgmt)

Azure Disks

MySQL – ClusterIP Service

MySql

PO
D

ReplicaSet

Deployment (mysql)

Storage Class

Persistent Volume Claim

Persistent Volumes

Deployment

Load Balancer Service

Deployment

Environment Variables

Init Containers

Config Map

Environment Variables

Volumes

Volume Mounts

ClusterIP Service

AKS Storage Azure Disks

Alper Can

Azure AKS Storage

with

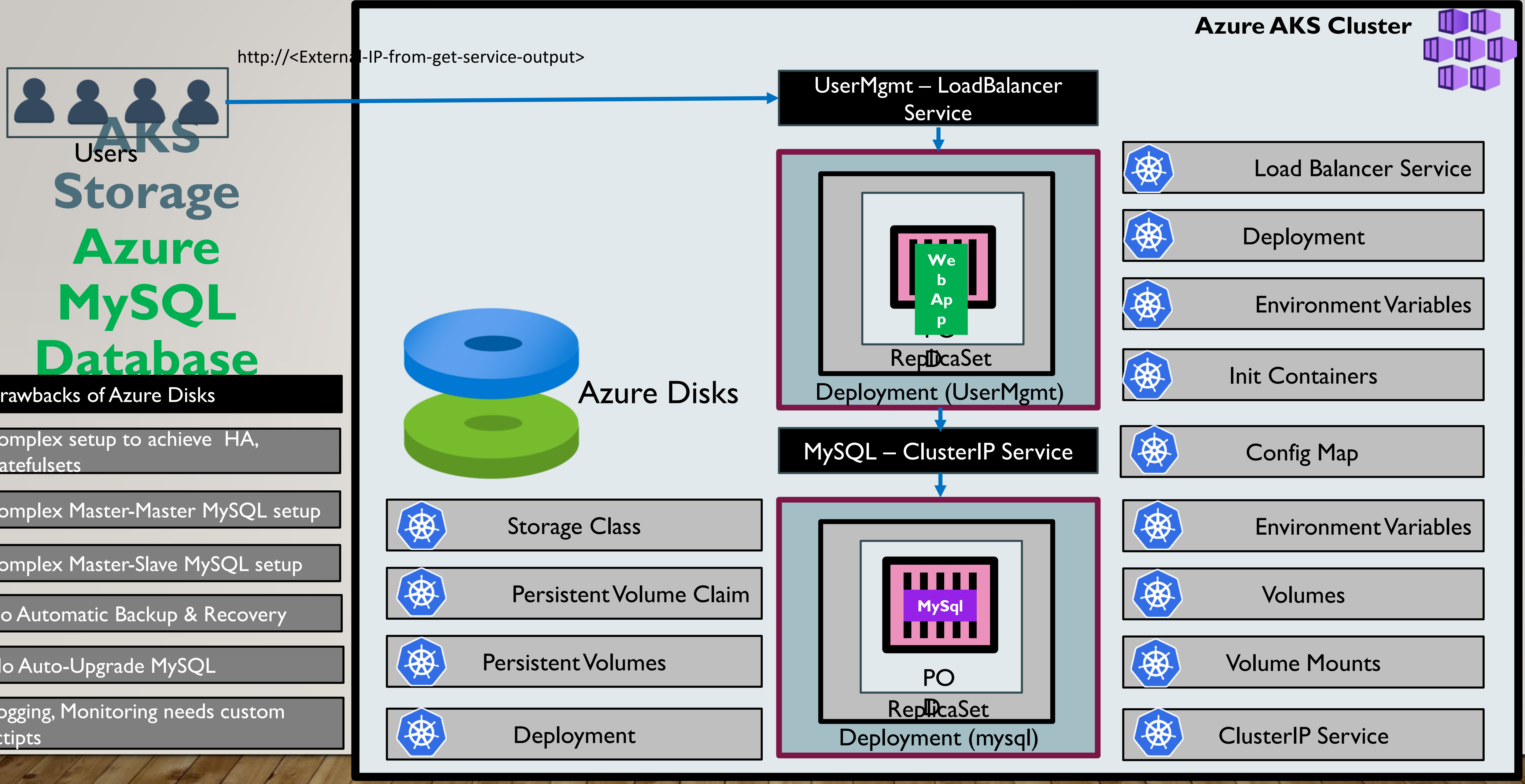
Azure MySQL Database

Alper Can

AKS Storage Azure MySQL Database

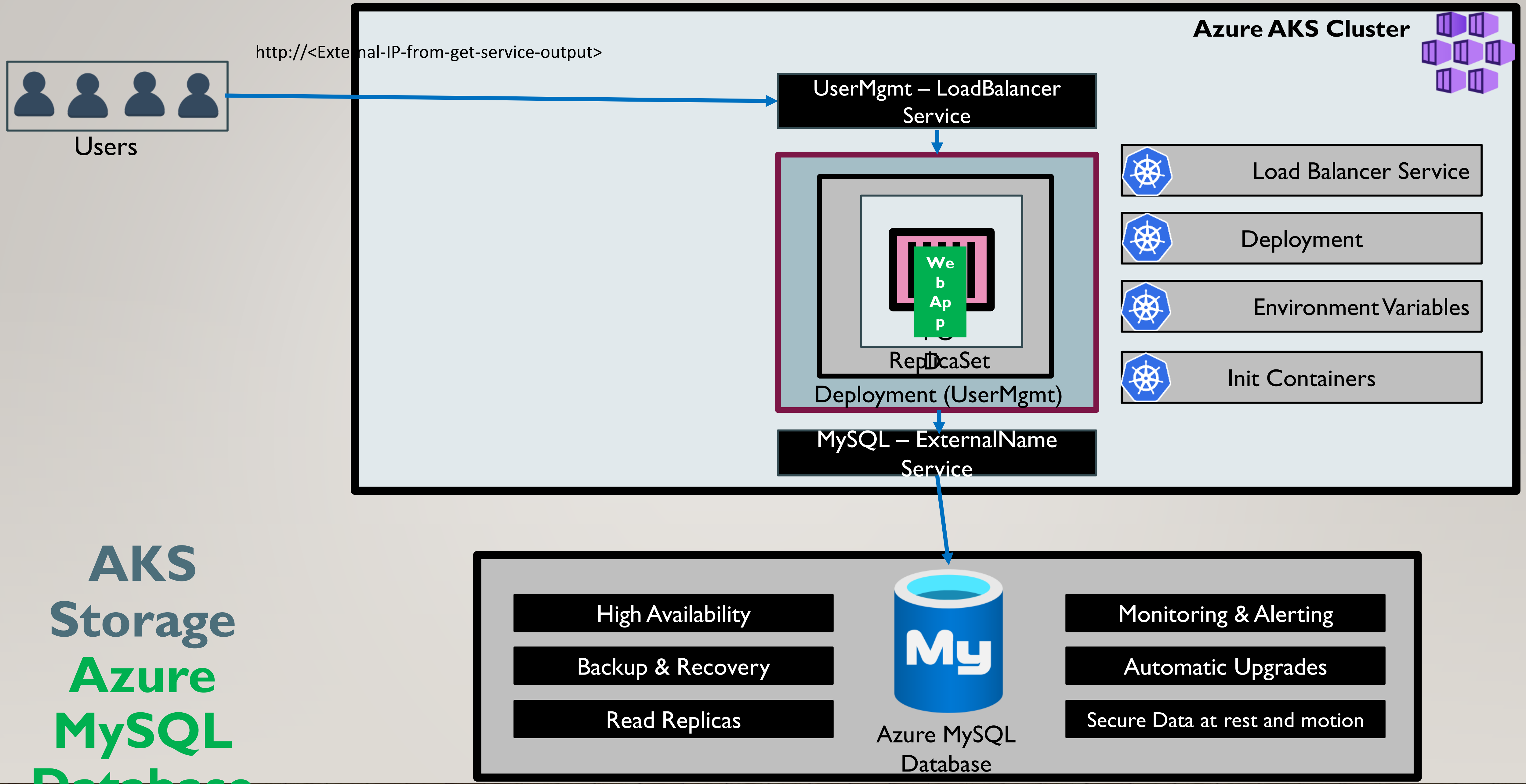
Drawbacks of Azure Disks

- Complex setup to achieve HA, Statefulsets
- Complex Master-Master MySQL setup
- Complex Master-Slave MySQL setup
- No Automatic Backup & Recovery
- No Auto-Upgrade MySQL
- Logging, Monitoring needs custom sctipts



AZURE MYSQL DATABASE

- **Features**
- Built-in high availability with no additional cost.
- Predictable performance, using inclusive pay-as-you-go pricing.
- Scale as needed within seconds.
- Secured to protect sensitive data at-rest and in-motion.
- Automatic backups and point-in-time-restore for up to 35 days.
- Enterprise-grade security and compliance.



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

Users

Azure AKS Cluster

UserMgmt – LoadBalancer Service

Web App

ReplicaSet

Deployment (UserMgmt)

MySQL – ExternalName Service

Load Balancer Service

Deployment

Environment Variables

Init Containers

High Availability

Backup & Recovery

Read Replicas

Azure MySQL Database

Monitoring & Alerting

Automatic Upgrades

Secure Data at rest and motion

AKS Storage Azure MySQL Database

Alper Can

Azure AKS Storage

with

Azure Files

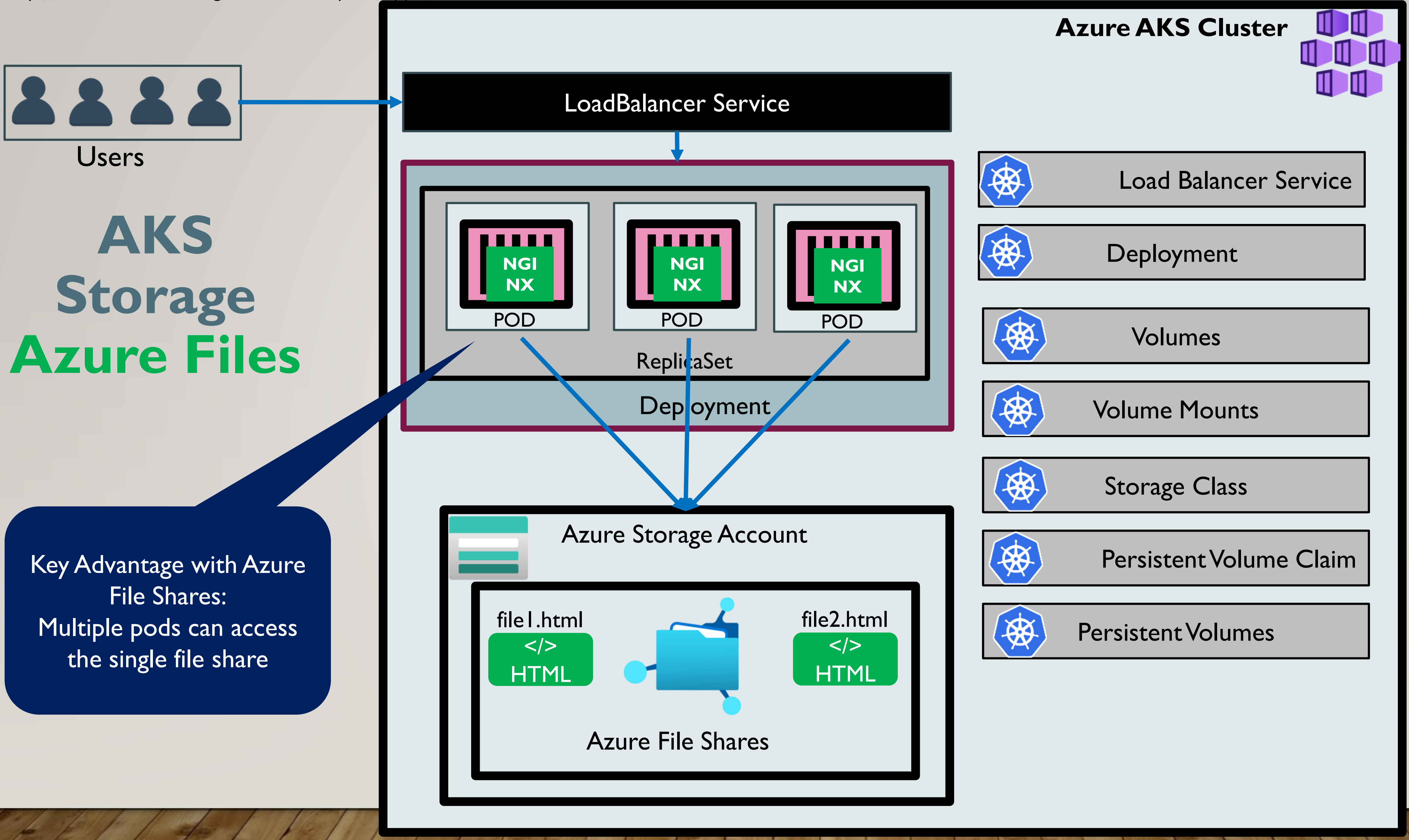
Alper Can

AZURE FILES

- These are simple, secure and fully managed cloud file shares
- We can secure data at rest and in-transit using SMB 3.0 and HTTPS
- We can create high-performance file shares using the Premium Files storage tier
- We can replace or supplement on-premises file servers
- Scripting and tooling: PowerShell cmdlets and Azure CLI can be used to create, mount, and manage Azure file shares as part of the administration of Azure applications.
- We can create and manage Azure file shares using Azure portal and Azure Storage Explorer.
- For Application workloads we can use for use cases like Static Content storage, shared configuration access to multiple JVMs etc.

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

AKS Storage Azure Files



Users

Azure AKS Cluster

LoadBalancer Service

Azure Storage Account

Azure File Shares

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

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