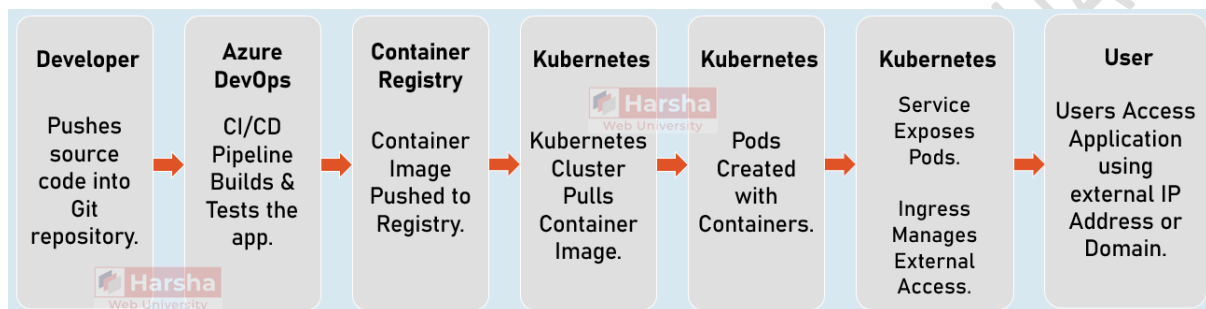


.NET Core Microservices – True Ultimate Guide

Section 13 – AKS (Kubernetes) – Cheat Sheet

Introduction to Kubernetes and AKS

Kubernetes, or K8s, is an open-source platform for deploying and managing containerized applications. It handles cluster management, scaling, and rolling updates, ensuring your applications run smoothly with minimal disruption.



Pros and Cons of AKS

Pros

Scalability

Easily scales microservices based on demand, ensuring efficient resource use.

Management

Azure handles the Kubernetes control plane, reducing your operational burden.

Integration

Seamlessly integrates with other Azure services for monitoring, logging, and security.

Flexibility

Supports complex, multi-container applications and orchestrates intricate deployments.

Cons

Complexity

Kubernetes can be complex to manage and configure, especially if you're new to container orchestration.

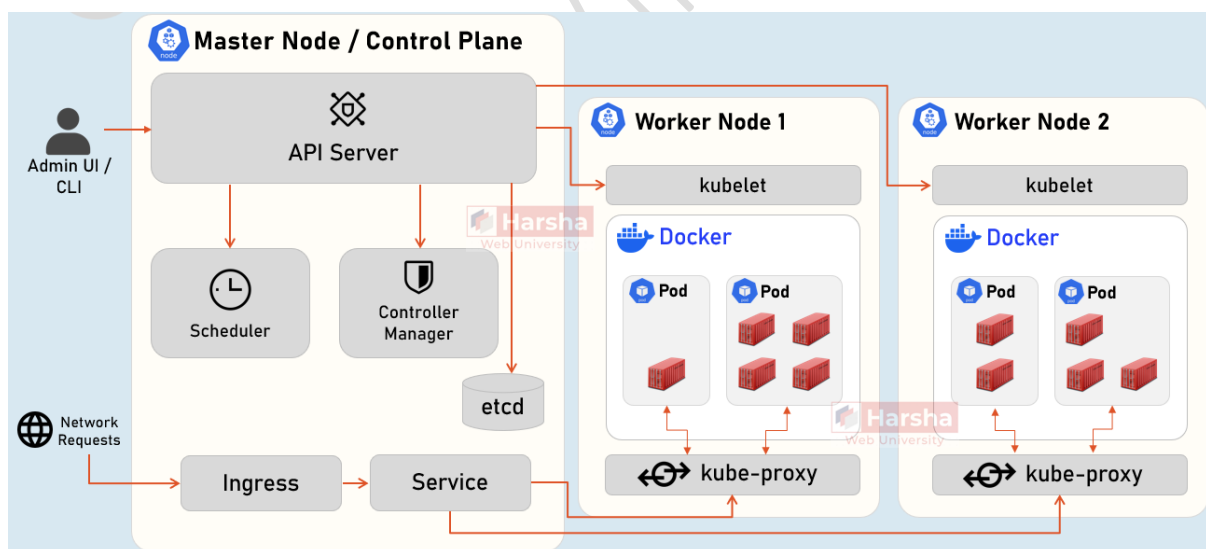
Learning Curve

There's a significant learning curve associated with Kubernetes and AKS.

Cost

While AKS itself doesn't have a direct cost, you pay for the underlying virtual machines and other resources.

Kubernetes Architecture



Cluster

The overall system that consists of all the Kubernetes components, including nodes, control plane, and the applications they manage.

Node

A machine (virtual or physical) in the Kubernetes cluster that runs containerized applications.

Each node contains the necessary components to run pods and is managed by the control plane.

Master Node / Control Plane

The central management layer that controls the entire Kubernetes cluster.

API Server

The core component that exposes the Kubernetes API. It's the entry point for all administrative tasks and the communication hub for all other components.

etcd

A distributed key-value store that holds all the cluster's data, such as configuration details and the state of the cluster.

Controller Manager

Manages controllers that handle routine tasks, such as ensuring the desired number of pod replicas are running.

Scheduler

Assigns pods to nodes based on resource availability and other constraints.

Service

An abstraction that defines a logical set of pods and a policy for accessing them, typically used to expose applications to other components within the cluster or to external clients.

Ingress

Manages external access to services, typically HTTP/S, providing load balancing, SSL termination, and name-based virtual hosting.

Pod

The smallest deployable unit in Kubernetes, typically containing one or more containers that share resources and network.

Kubelet

An agent running on each node that ensures containers are running in a pod as specified in the PodSpec in the deployment manifest.

Kube-Proxy

A network proxy running on each node that helps with routing traffic to the appropriate pod.

Volume

A storage abstraction that allows data to persist across pod restarts, supporting various storage backends.

ConfigMap

A Kubernetes object used to store non-sensitive configuration data as key-value pairs.

It allows you to decouple configuration artifacts from the application code, making it easier to manage and update configurations without rebuilding the container images.

Secret

A Kubernetes object designed to store sensitive data such as passwords, tokens, or keys.

Like ConfigMap, it stores data as key-value pairs but with added encryption for security.

Secrets keep sensitive information separate from the application code and configurations, protecting it from exposure.

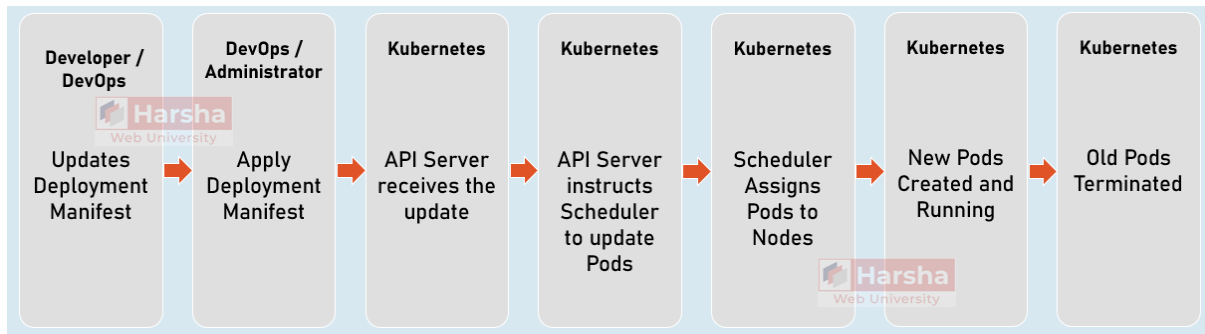
Namespace

A way to divide cluster resources between multiple users or teams, providing a scope for resources like pods and services.

Kubernetes Deployments

A Deployment in Kubernetes manages the creation, scaling, and updating of Pods, ensuring the application runs as specified.

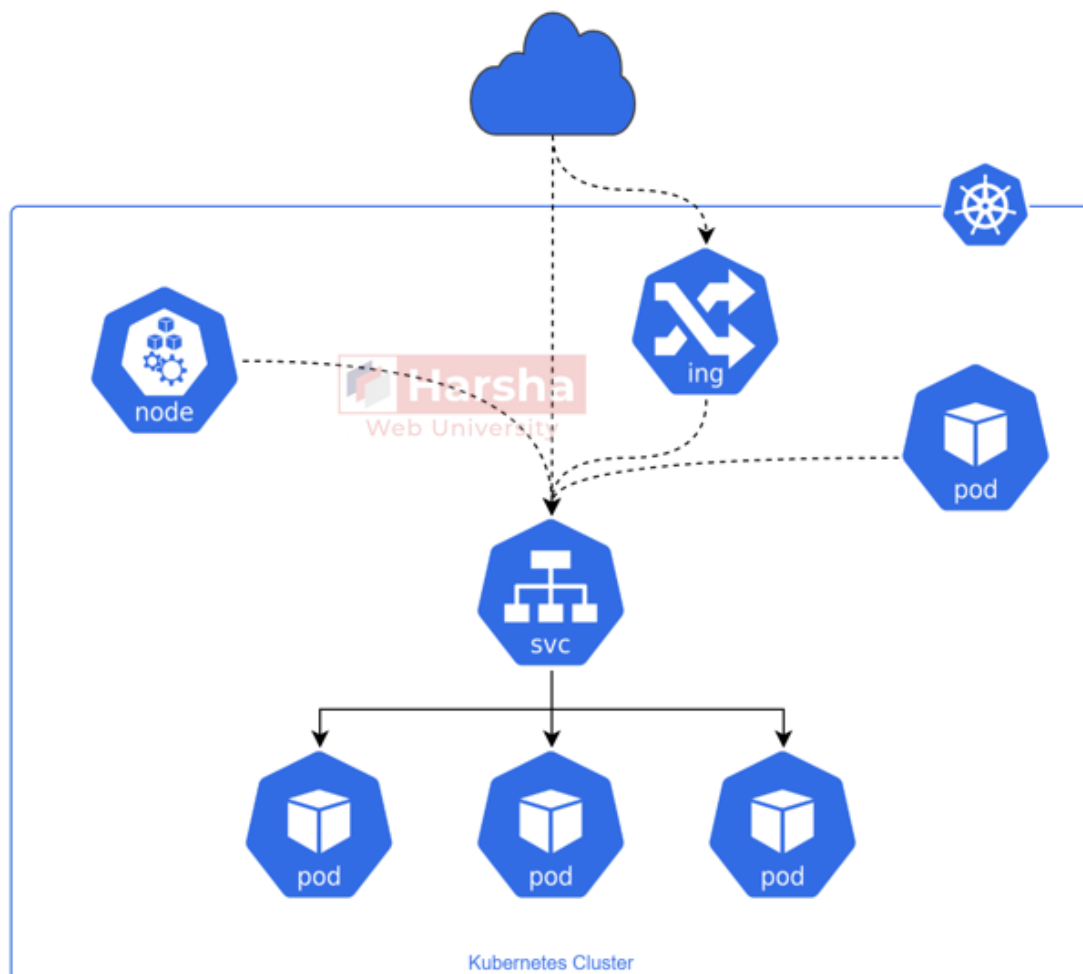
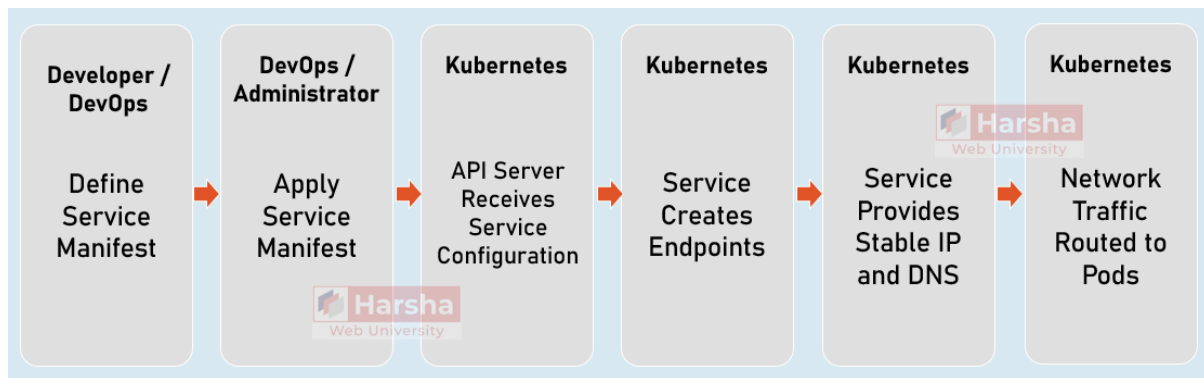
It automates rolling updates, rollbacks, and scaling to maintain the desired state of the application.



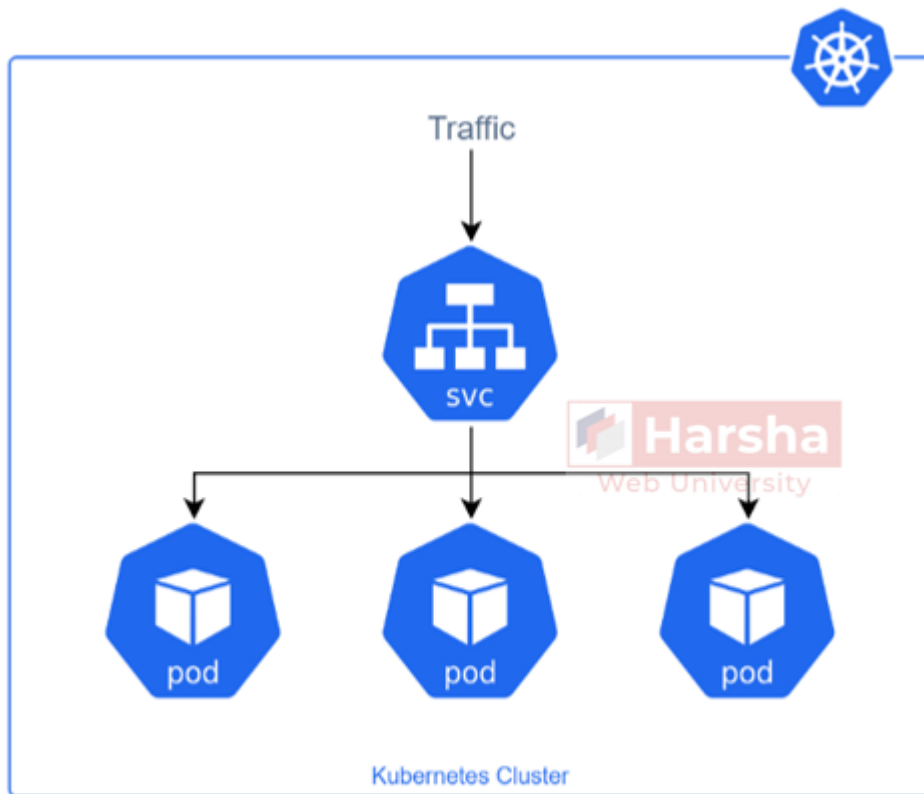
Kubernetes Services

A Service in Kubernetes provides a stable network endpoint for a set of Pods, allowing consistent access through a fixed IP and DNS name.

It handles load balancing and can be configured for internal or external access.

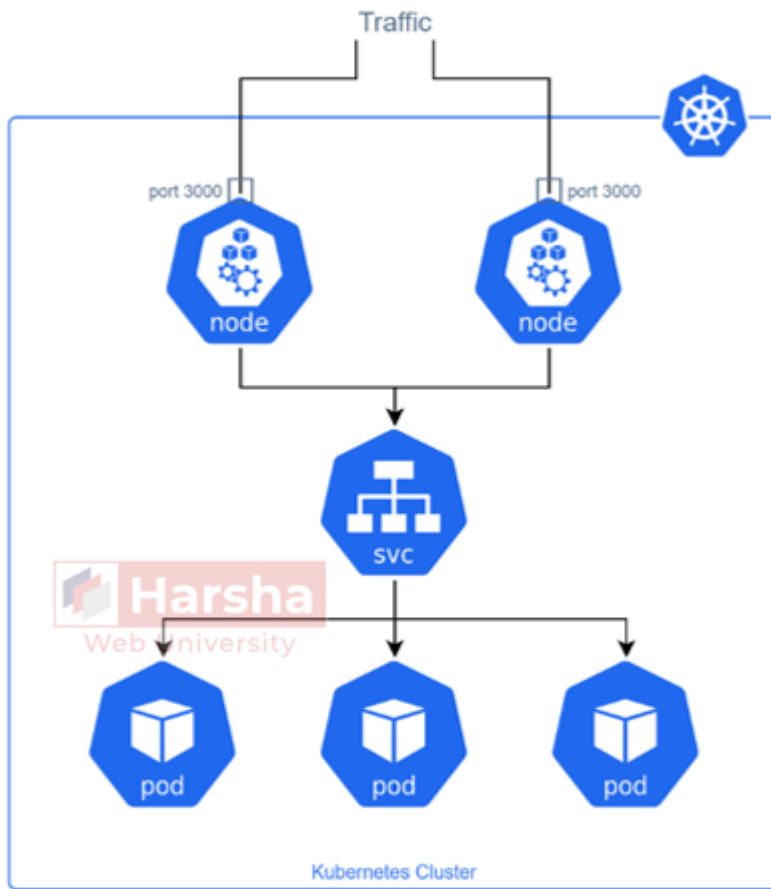


Kubernetes - Services - Cluster



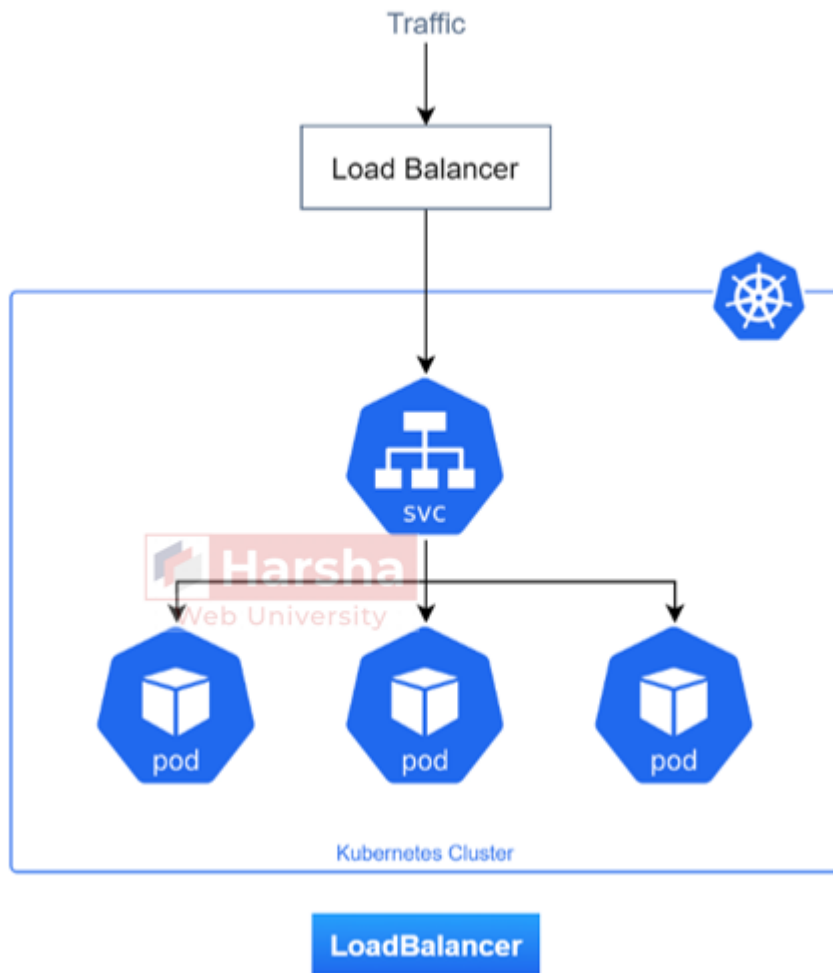
ClusterIP

Kubernetes - Services - NodePort

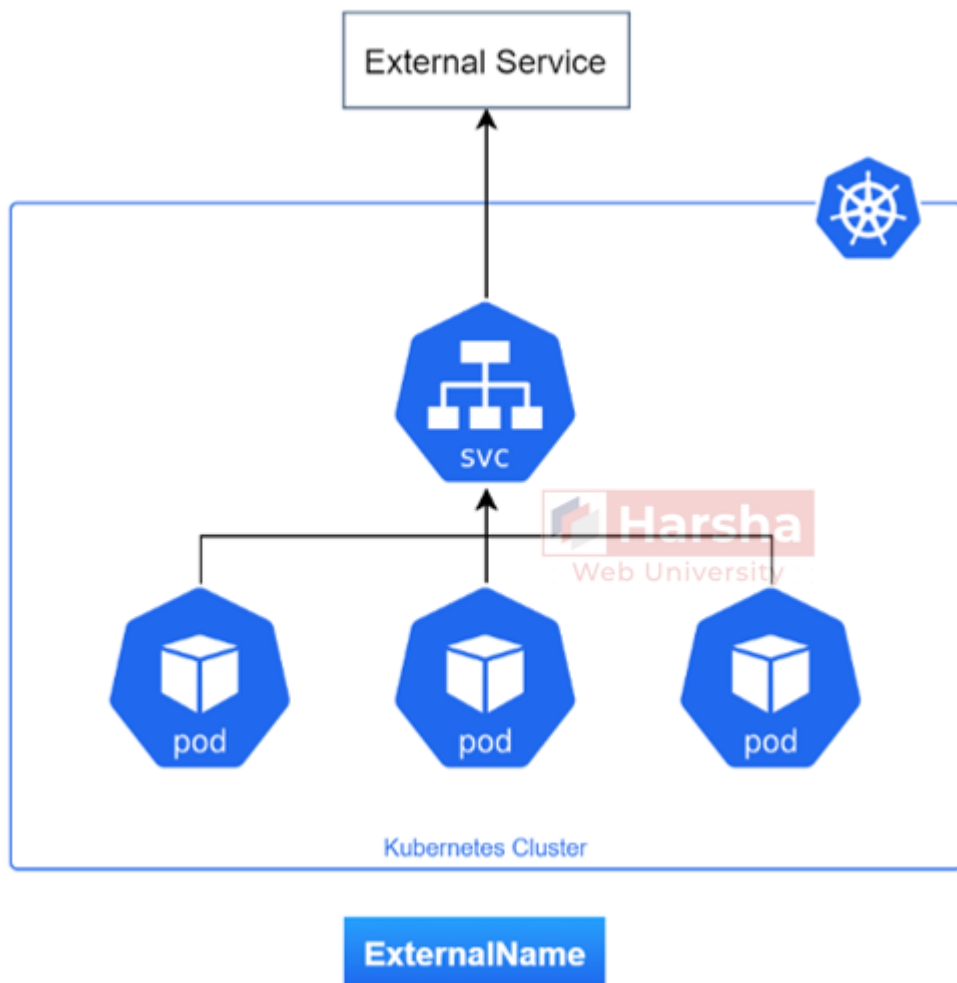


NodePort

Kubernetes - Services - LoadBalancer



Kubernetes - Services - ExternalName



Kubernetes Secrets

Secrets in Kubernetes are objects designed to store and manage sensitive data like passwords, tokens, and keys securely.

They ensure that sensitive information is protected and can be accessed by applications within the cluster without exposing it in plaintext.

Secrets in Kubernetes are stored in etcd (the cluster's distributed key-value store).

They are encoded in base64 and can be encrypted at rest, depending on the cluster's configuration.