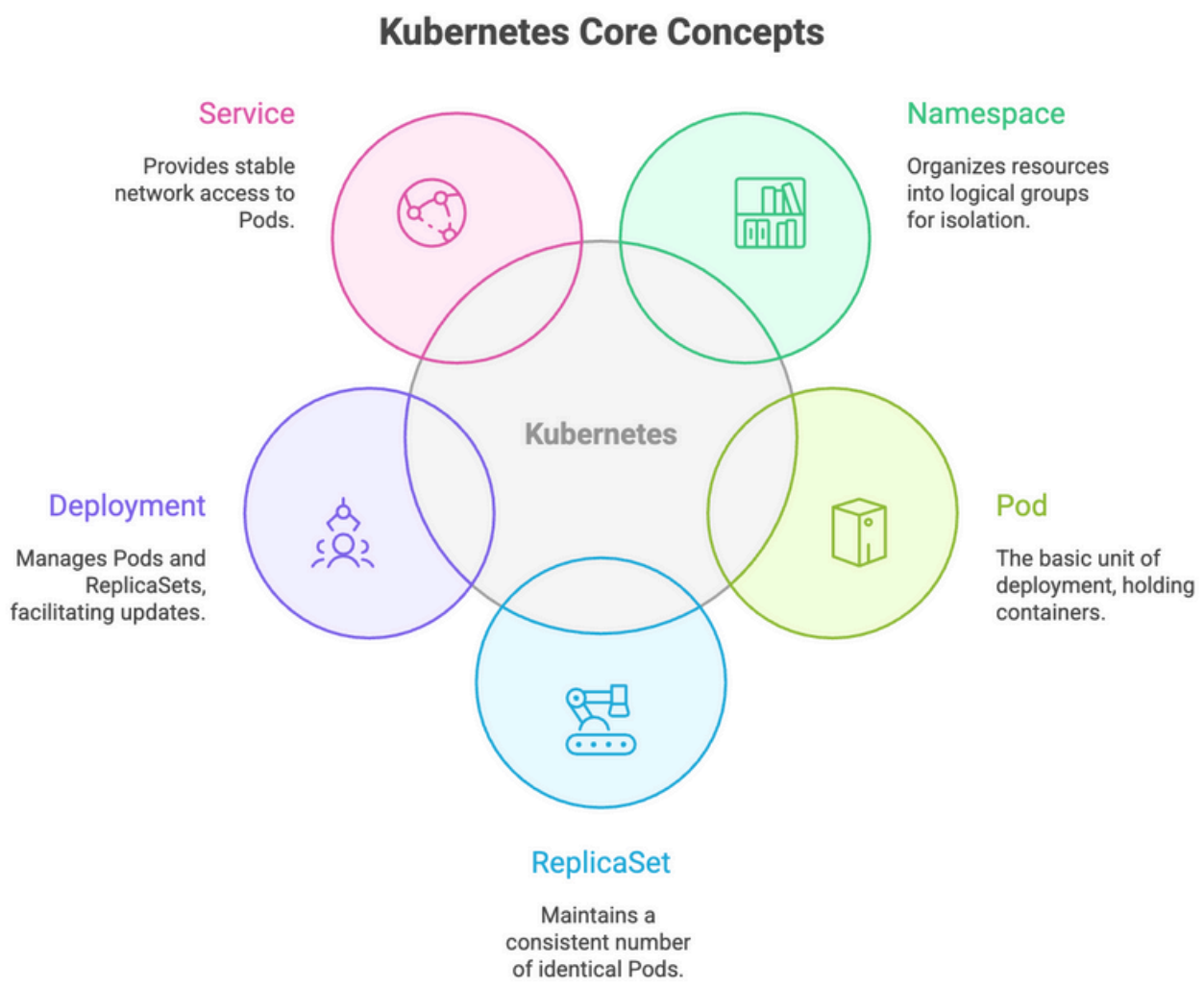
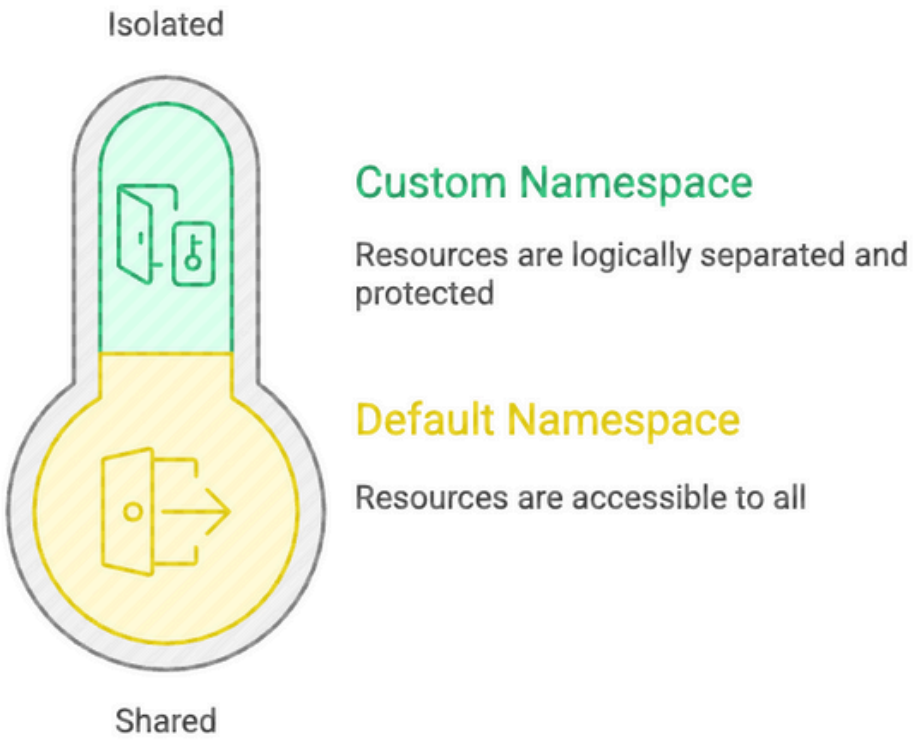
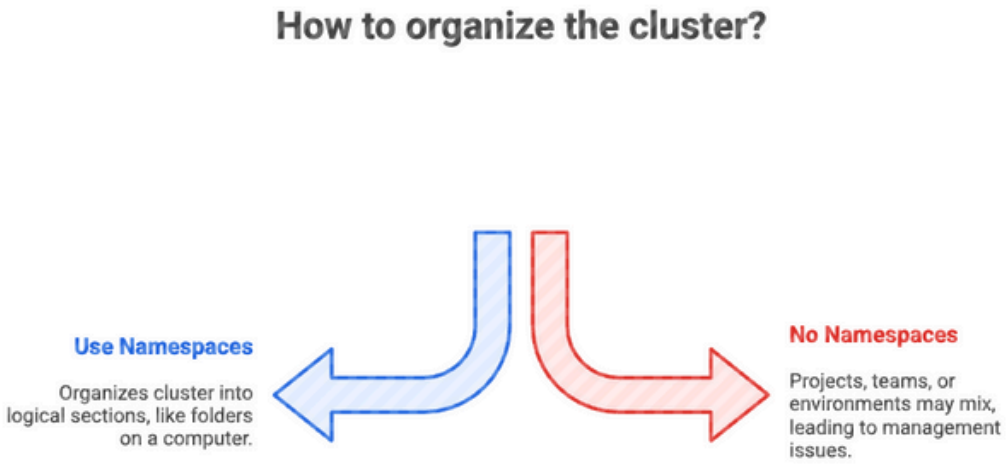


The main Kubernetes objects:

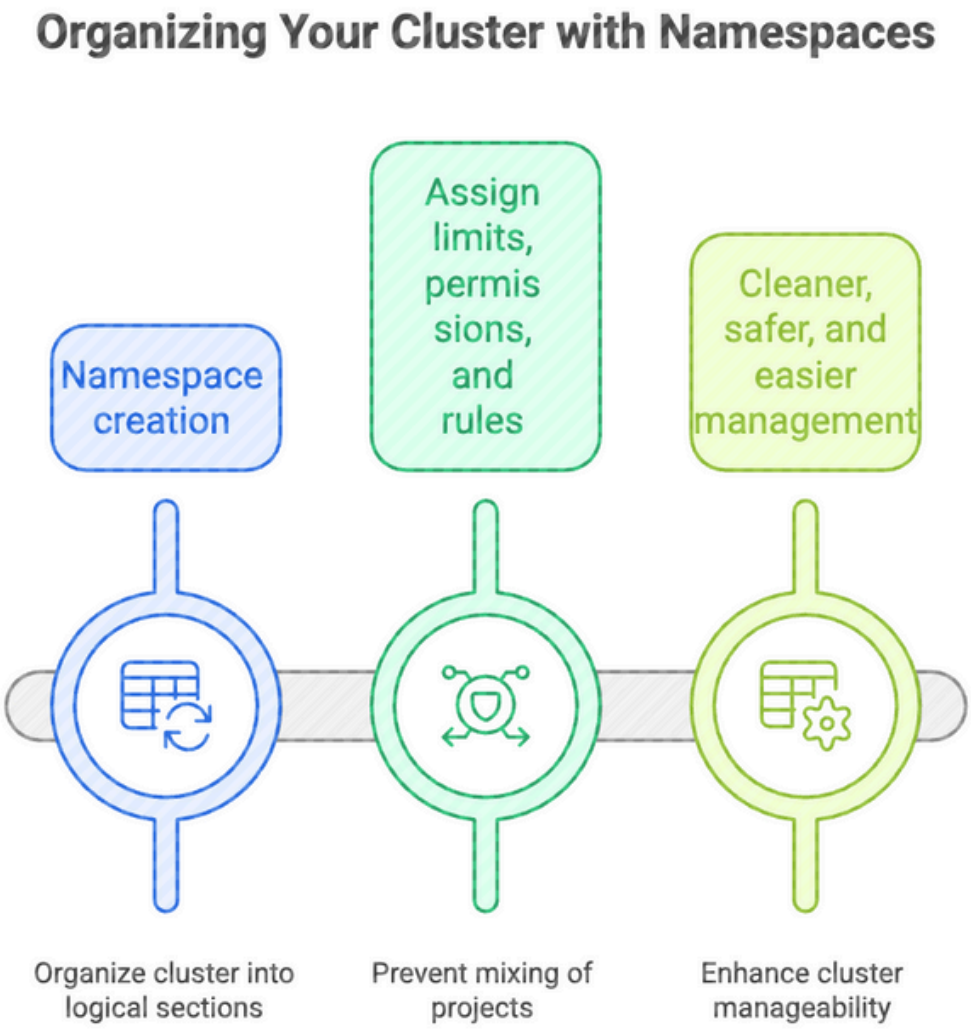
- **Namespace:** A way to group and separate resources inside Kubernetes.
- **Pod:** The smallest runnable unit that holds one or more containers.
- **ReplicaSet:** Ensures a fixed number of identical Pods are always running.
- **Deployment:** Manages Pods and ReplicaSets, enabling updates and rollbacks.
- **Service:** Gives Pods a stable way to be reached over the network.

1) Namespace — “The Organizer”





A Namespace helps you organize your cluster into logical sections.
Just like folders on a computer, it keeps different projects, teams, or environments separate.
You can assign limits, permissions, and rules per namespace to prevent everything from mixing.



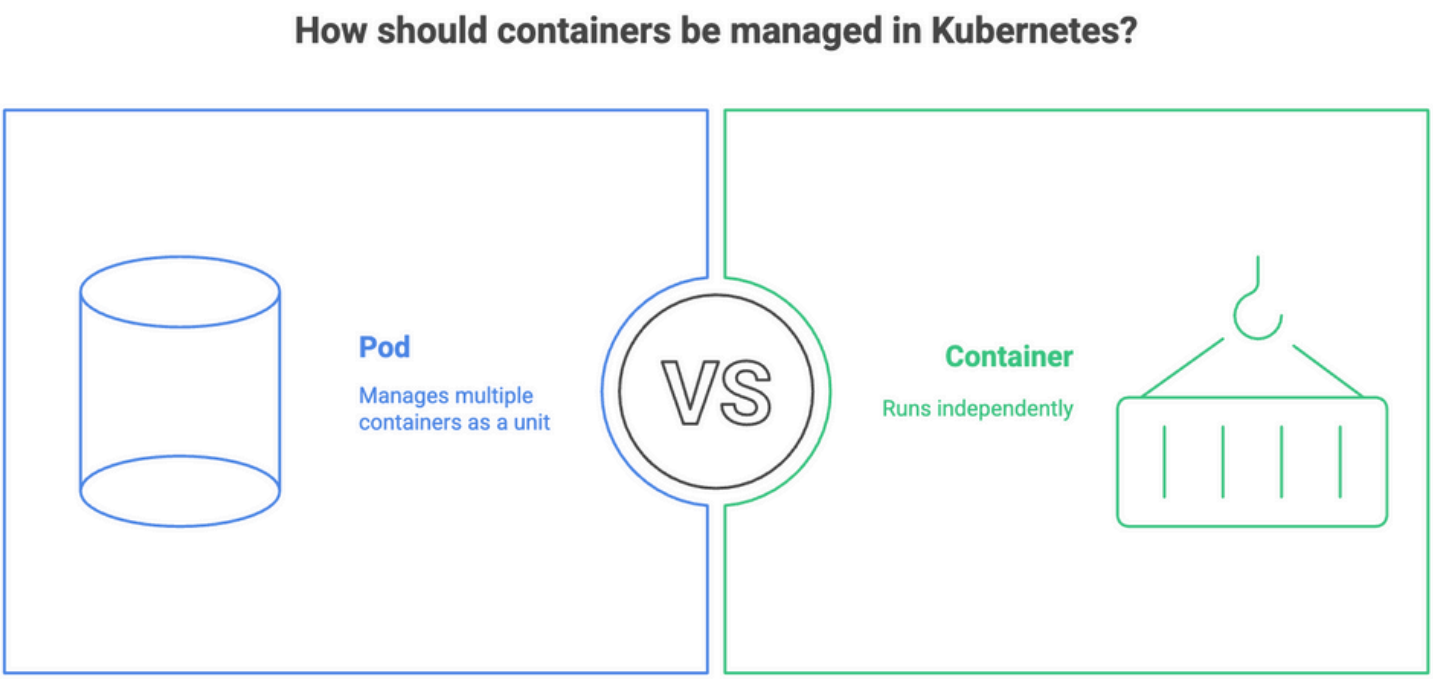
Why it matters:

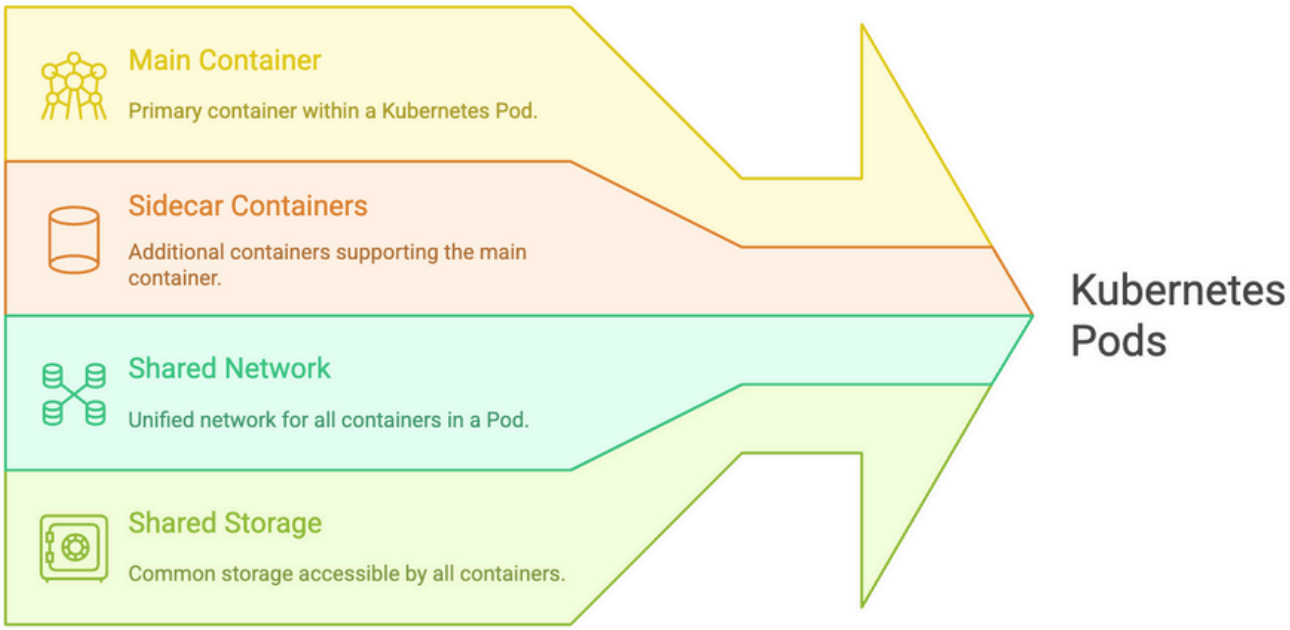
Makes large clusters cleaner, safer, and easier to manage.

Kubectl Namespace Commands

Command	`kubectl get namespaces`	`kubectl create namespace <namespace-name>`	`kubectl get all -n <namespace-name>`	`kubectl delete namespace <namespace-name>`
Description	Lists available namespaces	Creates a new namespace	Lists resources in a namespace	Deletes a namespace

2) Pod — “The Home for Containers”

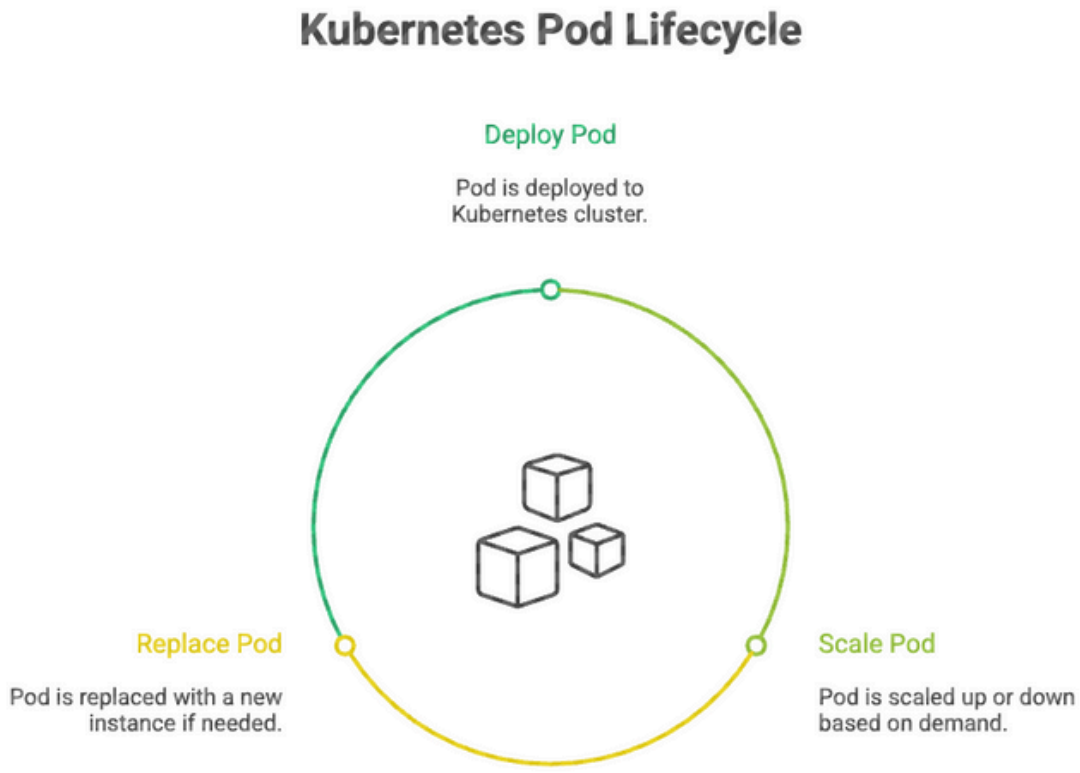




Kubernetes does not run containers directly.
It runs Pods, and a pod can contain:

- One main container
- Or multiple containers that must work closely together

Pods share the same network and storage, so the containers inside them act like a small team.

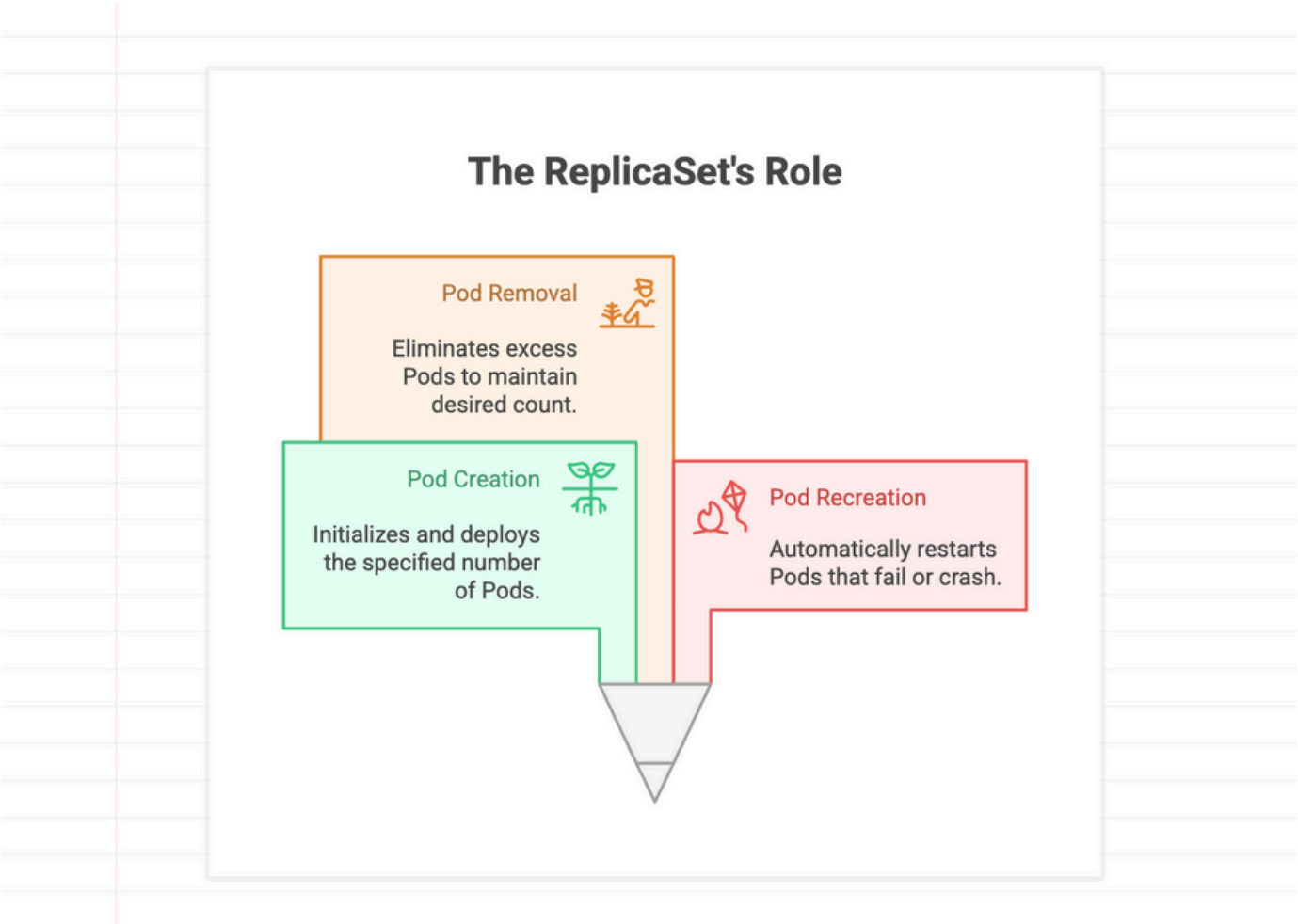


Why it matters:
It’s the smallest unit Kubernetes can deploy, scale, or replace

Basic PODS Commands

Command	`kubectl get pods`	`kubectl run <pod-name> –image=<image-name>`	`kubectl run <pod-name> –image=<image-name> -n <namespace-name>`	`kubectl get pods -n <namespace-name>`	`kubectl describe pod <pod-name>`	`kubectl delete pod <pod-name>`
Description	List pods in default namespace	Create pod in default namespace	Create pod in specified namespace	List pods in specified namespace	Describe pod	Delete pod

3) ReplicaSet — “The Pod Counter & Protector”



A ReplicaSet’s job is very simple and very important

Keep the desired number of Pods running.

If you say “I want 3 pods,” it:

- Creates them
- Recreates them when they crash
- Removes extras if more than 3 exist

Why it matters:
Guarantees availability and reliability.

How to ensure pod availability and reliability?

Pod Creation

Creates and maintains the desired number of pods

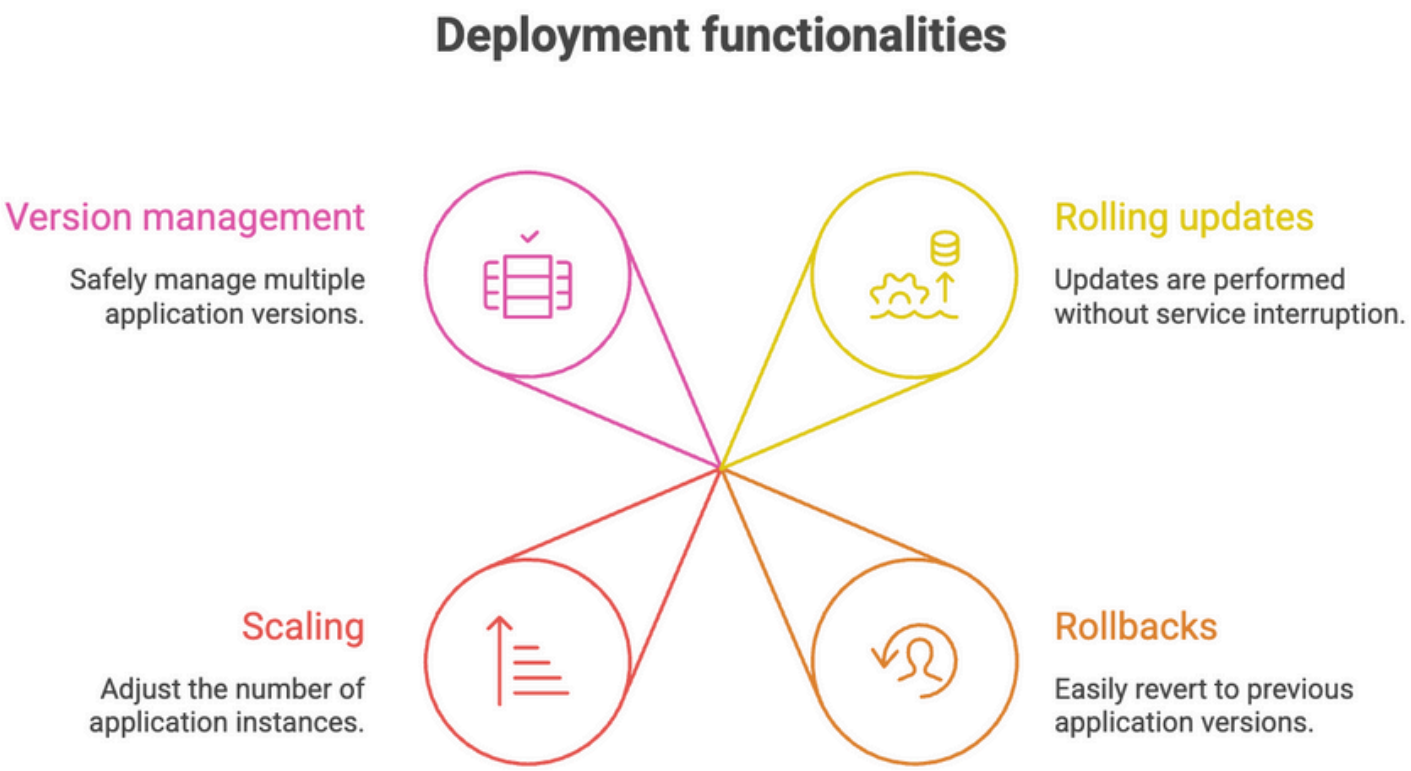
Pod Protection

Recreates crashed pods and removes extras

kubectl Commands for ReplicaSets

Command	`kubectl get rs`	`kubectl describe rs <rs-name>`	`kubectl delete rs <rs-name>`
Description	Lists all ReplicaSets	Describes a ReplicaSet	Deletes a ReplicaSet

4) Deployment — “The Manager with Upgrade Powers”



A Deployment sits above the ReplicaSet and acts as the real controller of your application.
It handles:

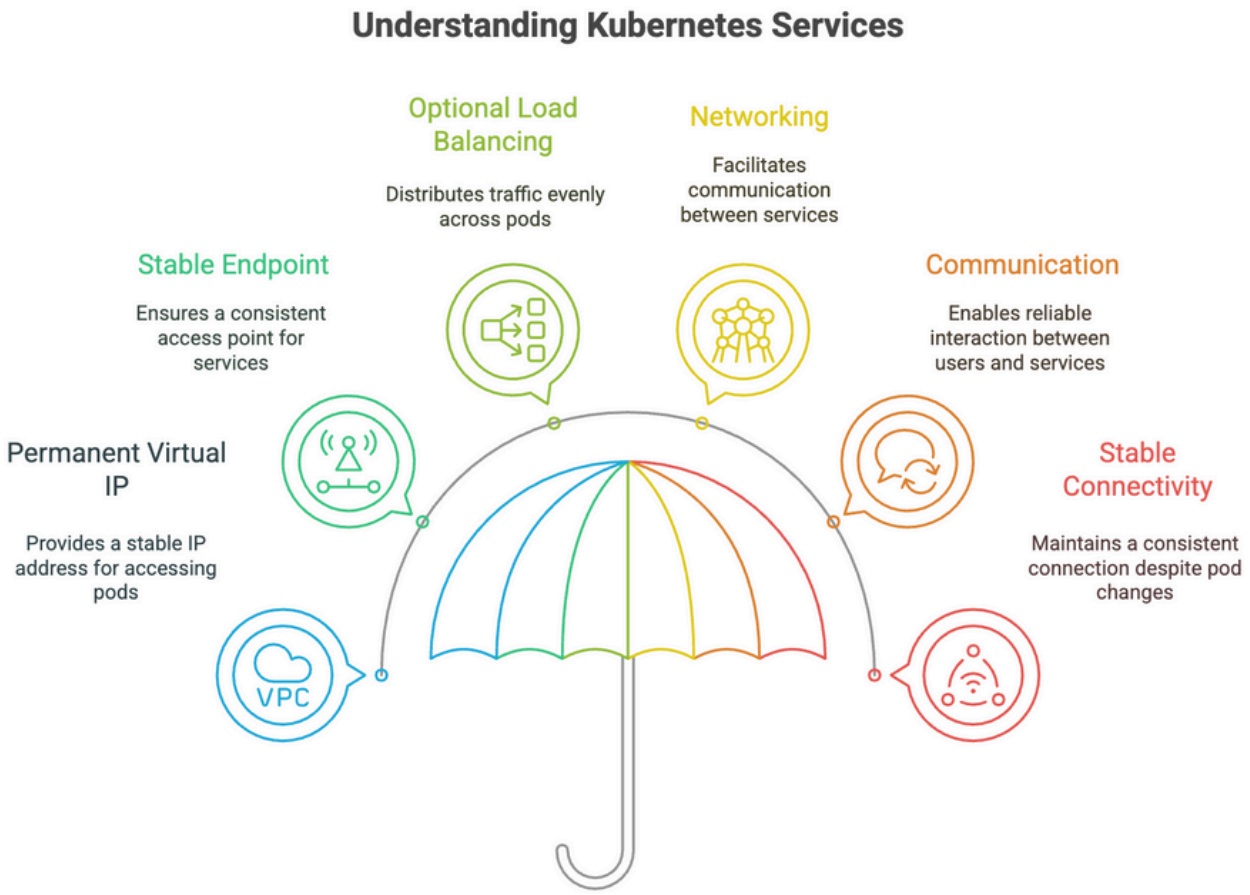
- Rolling updates
- Rollbacks
- Scaling
- Managing multiple app versions safely

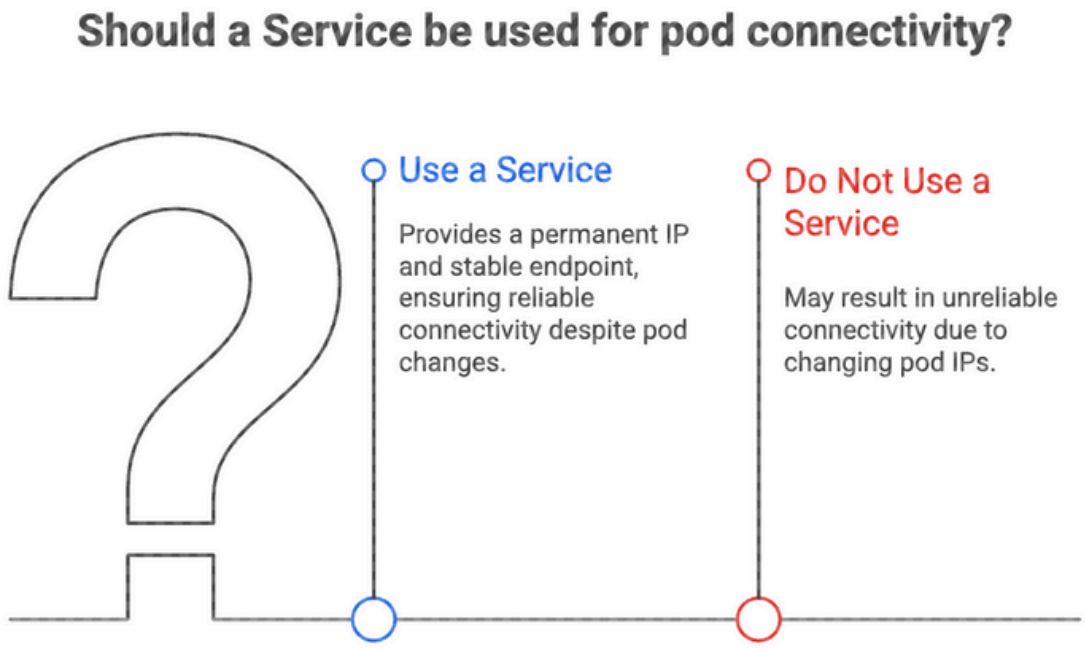
When you change a Deployment:

- It creates a new ReplicaSet
- Gradually replaces old pods with new ones
- Ensures zero downtime (if configured correctly)

Why it matters:
It’s the safest and most flexible way to run and update applications in Kubernetes.

5) Service — “The Permanent Address”





Pods come and go.
Their IPs constantly change.
A Service solves this by giving:

- A permanent virtual IP
- A stable endpoint
- Optional load balancing

It lets other services or users reach your pods reliably, even if pods keep changing behind the scenes.

Why it matters:
Enables networking, communication, and stable connectivity.