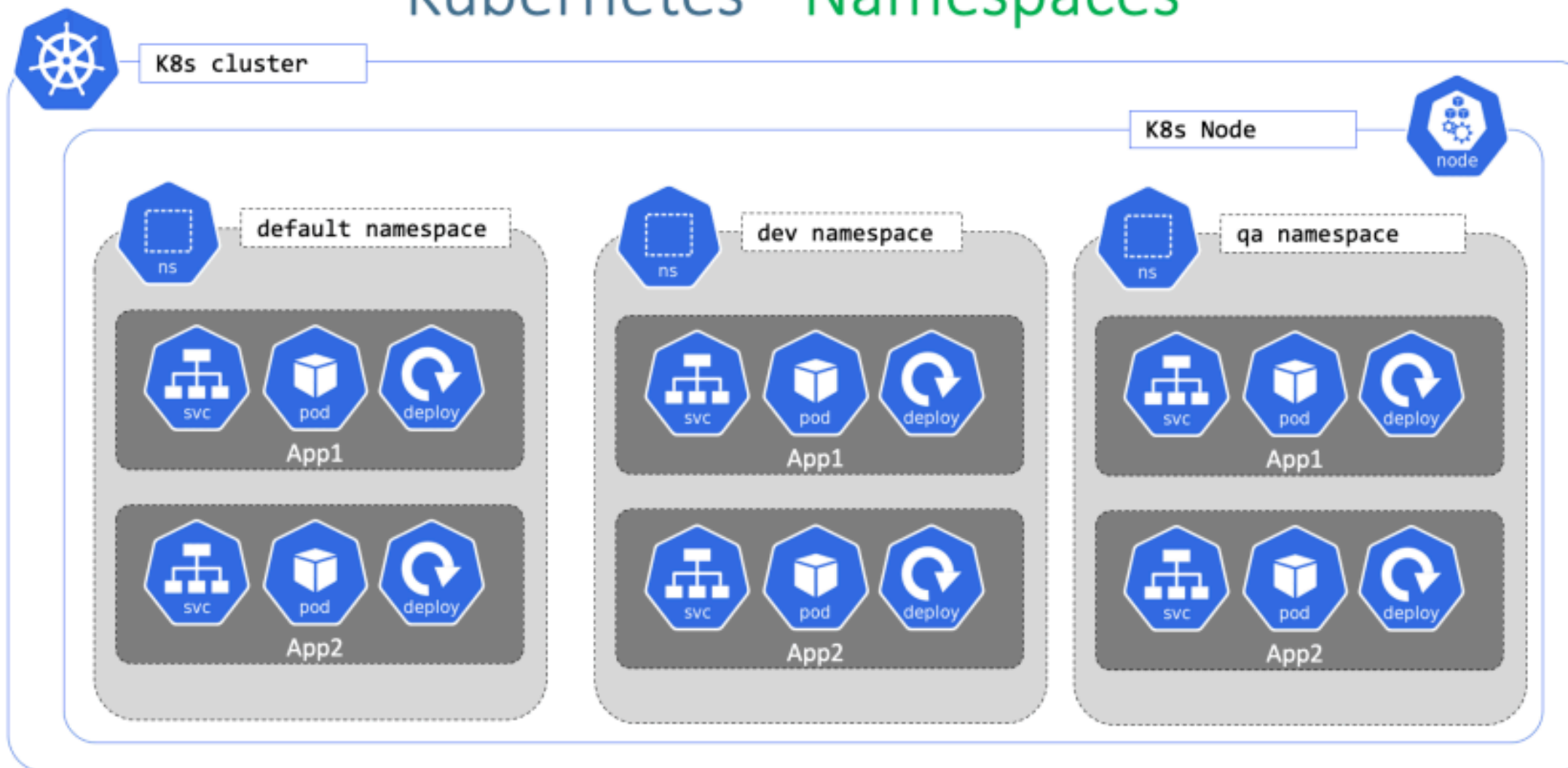
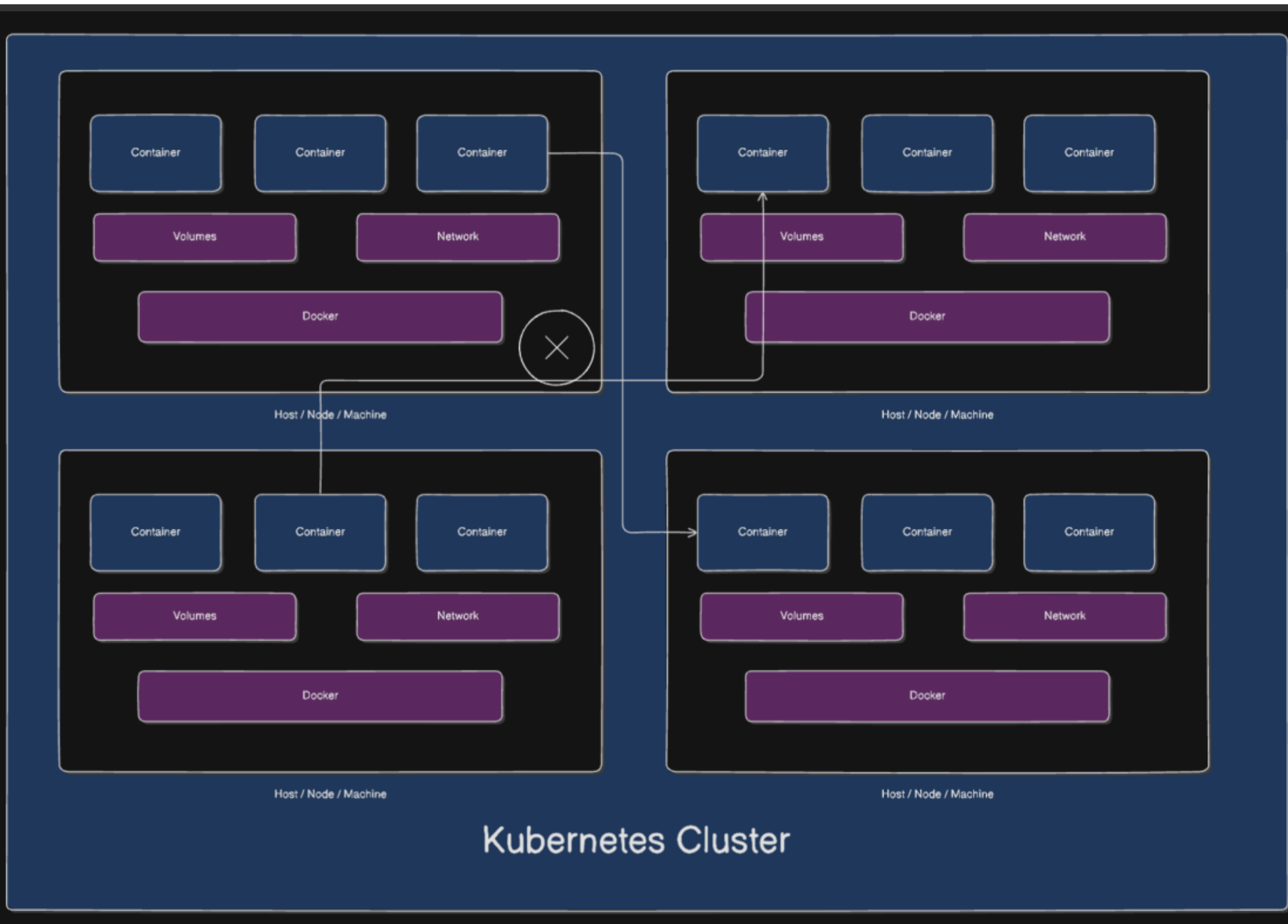


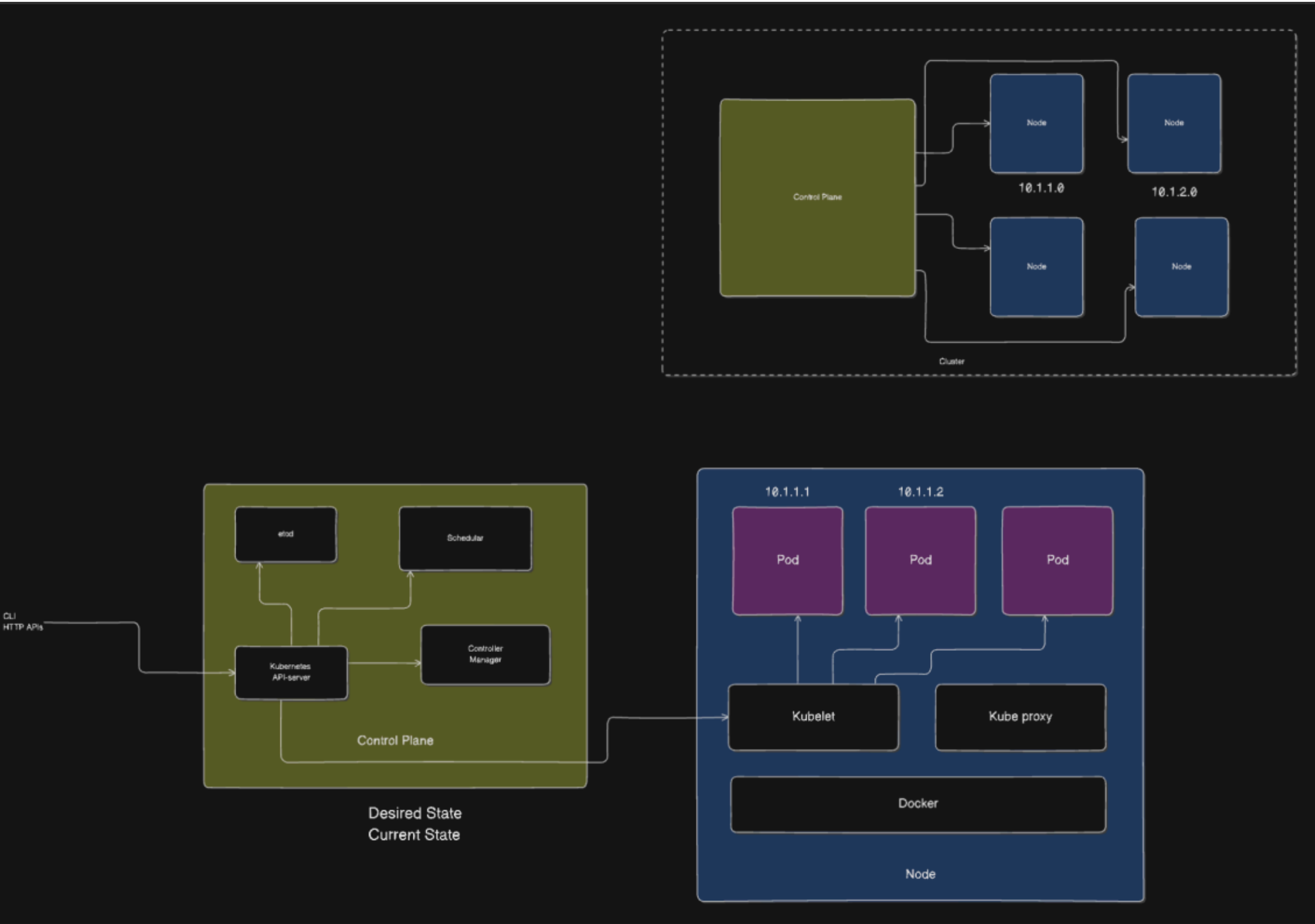
Kubernetes

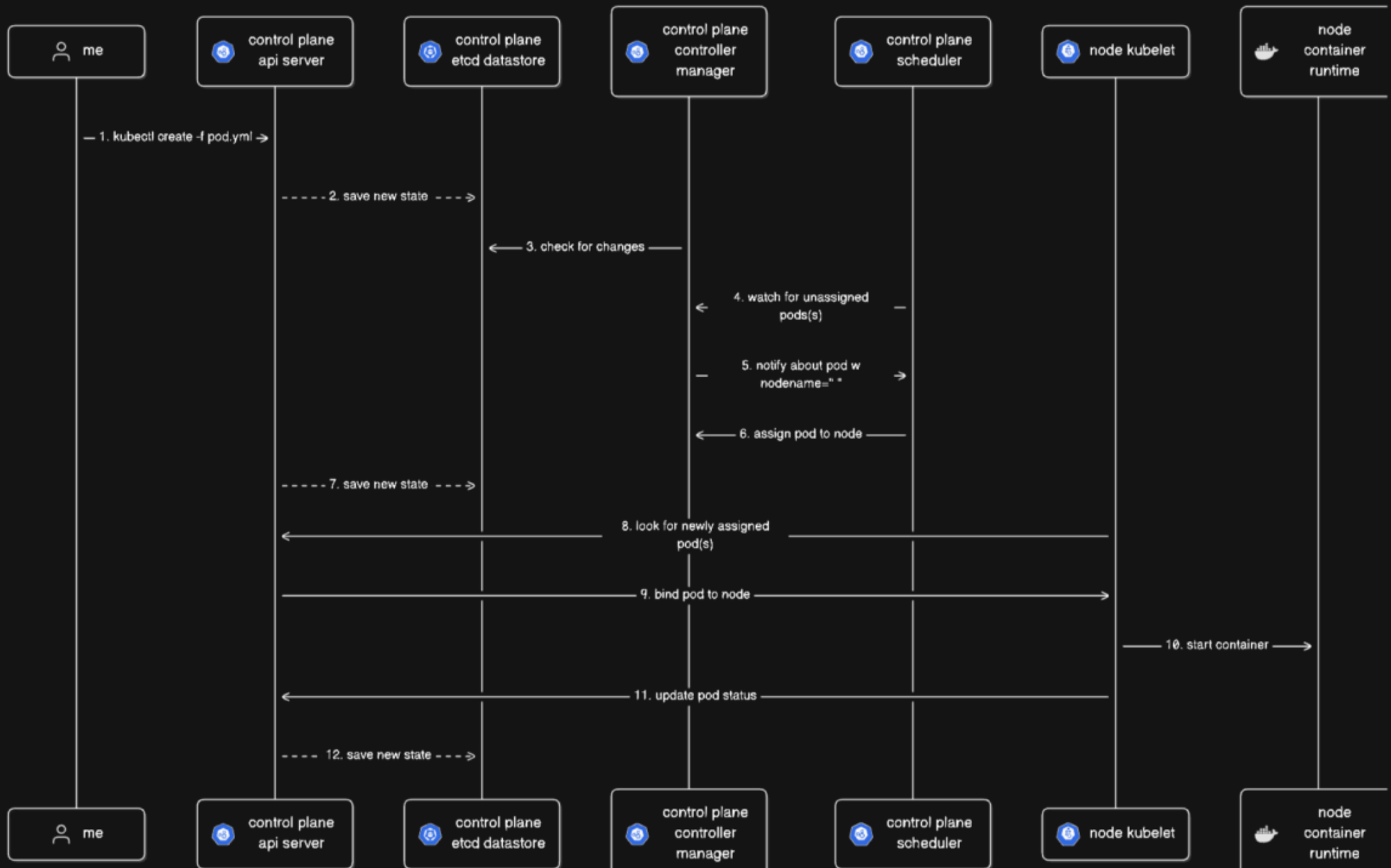
Pod, Deployment, ReplicaSet and Namespace

Kubernetes - Namespaces





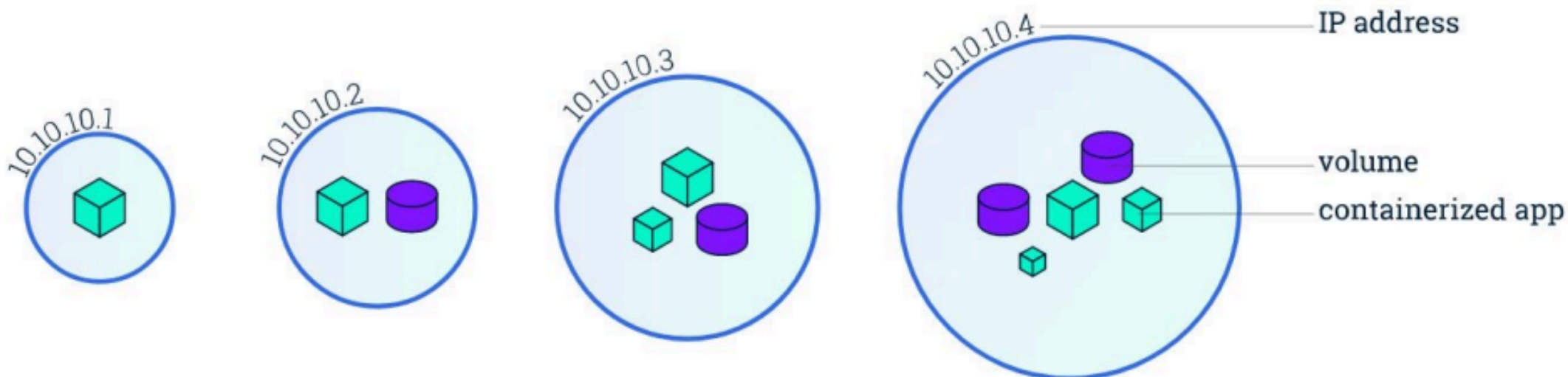




Namespace

A namespace is a logical partitioning of cluster resources that allows for the organization and separation of resources within the cluster. Namespaces provide a way to create multiple virtual clusters within a single physical cluster, allowing teams or applications to operate independently and avoid resource conflicts.

Pods



**A Pod is the smallest and simplest
Kubernetes object**

**. It represents a single instance of a
running process in your cluster.**

- A Pod can contain one or more containers,
which share storage and network resources.**
- All containers in a Pod are scheduled
on the same node and can communicate
with each other via localhost.**
- Pods are ephemeral and can be
created, destroyed, or replaced dynamically.**

ReplicaSet

replicas: 3

ReplicaSet



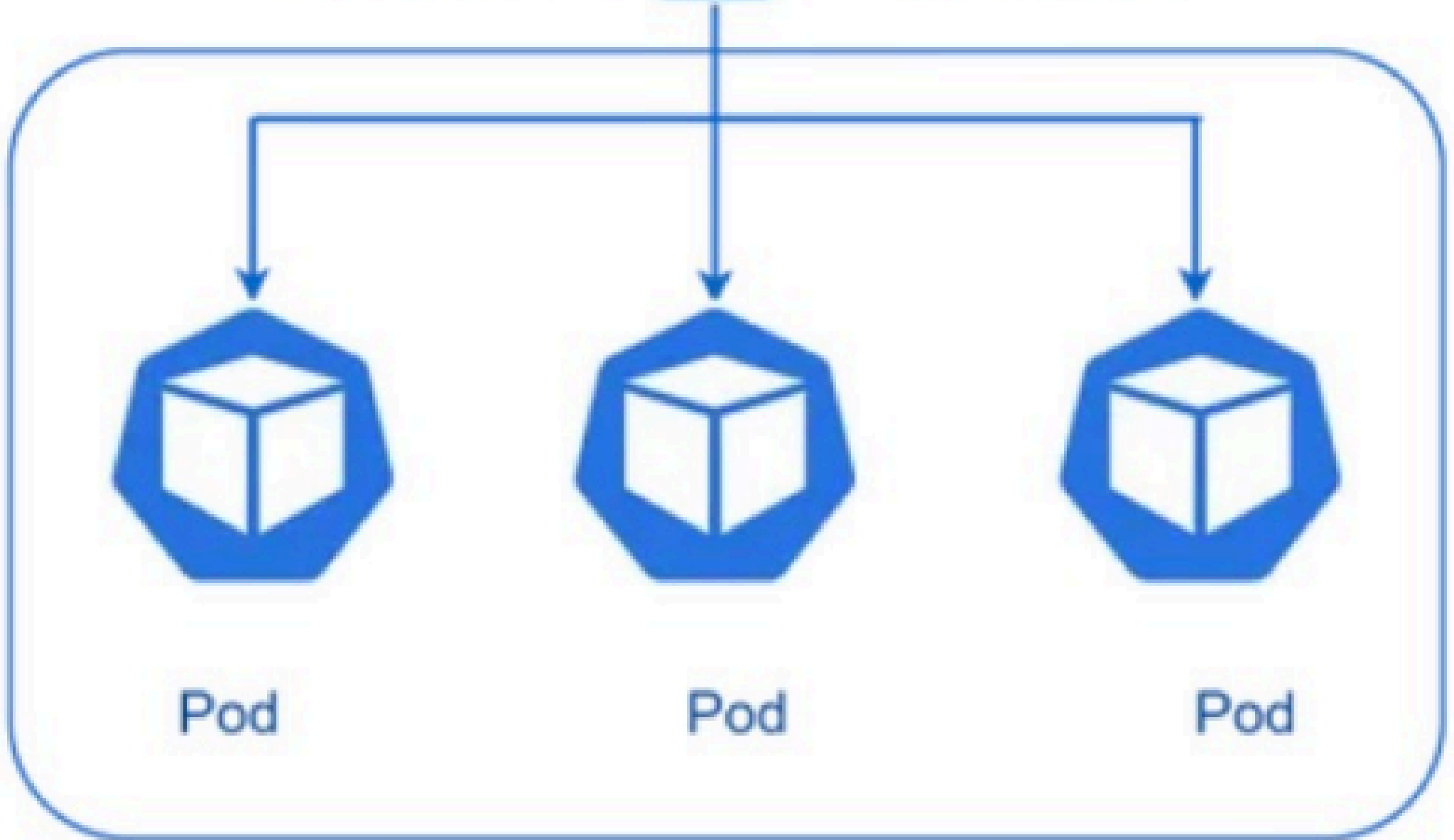
Pod



Pod



Pod



ReplicaSet

A ReplicaSet is responsible for maintaining a stable set of replica Pods running at any given time.

- It ensures that a specified number of identical Pods are running, providing high availability**
- If a Pod fails or is terminated, the ReplicaSet automatically creates a new Pod to replace it**
- A ReplicaSet is defined by a label selector that identifies the Pods it manages.**

Deployment

A Deployment provides declarative updates for Pods and ReplicaSets. It is a higher-level abstraction over ReplicaSets.

- It allows you to describe the desired state of your application (like which images to use, how many replicas, etc.).
- It manages the creation and scaling of ReplicaSets and Pods.
- It enables easy rollbacks to previous versions of your application and handles rolling updates without downtime.

Manage Pods

- `kubectl run my-pod --image=nginx`
- `kubectl get pods`: Lists all Pods in the default namespace.
- `kubectl get pods -n <namespace>`: Lists Pods in a specific namespace.
- `kubectl describe pod <pod-name>`: Shows detailed information about a specific Pod.
- `kubectl logs <pod-name>`: Retrieves logs for a specific Pod.
- `kubectl exec -it <pod-name> -- /bin/bash`: Opens a shell session inside a Pod (if it supports it).
- `kubectl delete pod <pod-name>`
- `kubectl apply -f pod.yaml`: Create a Pod (from a YAML file):

Manage Deployments

- `kubectl create deploy <deployment-name> --replicas=3 --image=<image-name>`
- `kubectl scale deployment <deployment-name> --replicas=5`
- `kubectl set image deployment/<deployment-name> <container-name>=<new-image>` : Update a Deployment
- `kubectl rollout undo deployment/<deployment-name>` : Rollback to Previous Revision:
- `kubectl rollout history deployment/<deployment-name>` : Check Rollout History:
- `kubectl rollout undo deployment/<deployment-name> --to-revision=<revision-number>` : To rollback to a specific revision

Manage Namespaces

```
kubectl get namespaces or kubectl get ns
kubectl create namespace <namespace-name>
kubectl delete namespace <namespace-name>
kubectl get pods -n <namespace-name>
kubectl config set-context --current --namespace=<namespace-name>
kubectl get pods --all-namespaces or kubectl get pods -A
kubectl describe namespace <namespace-name>
kubectl edit namespace <namespace-name>
kubectl get all --all-namespaces (to get all resources from all namespaces)
```