

What is a Namespace?

A Namespace in Kubernetes is like a folder or workspace inside the cluster.

It is used to organize and separate resources (Pods, Services, Deployments, ConfigMaps, etc.).

Namespaces allow:

- Different teams or environments (like dev, test, prod) to share the same cluster without interfering with each other.
- Applying access control (who can access what).
- Setting resource limits (so one team can't consume all cluster resources).

Example: You can have a pod called web in both dev namespace and prod namespace, and they won't conflict.

Simple Example

Imagine you have a development team and a production team using the same cluster.

- You create a dev namespace → Developers deploy apps here.
- You create a prod namespace → Production apps run here, isolated from dev.

So, the same cluster can safely host both environments.

Useful Commands

1. List all namespaces `kubectl get namespaces` Shows all namespaces in the cluster (default ones: default, kube-system, kube-public, etc.).
2. Create a new namespace

```
kubectl create namespace dev
```

Creates a namespace named dev.

3. Run a Pod inside a namespace

```
kubectl run nginx --image=nginx -n dev
```

Starts an nginx pod inside the dev namespace.

4. See resources inside a namespace

```
kubectl get pods -n dev  
kubectl get all -n dev
```

Shows only the resources running inside the dev namespace.

5. Set default namespace for current context

```
kubectl config set-context --current --namespace=dev
```

After this, you don't need `-n dev` in every command (kubectl will default to dev).

6. Describe a namespace

```
kubectl describe namespace dev
```

Shows details like labels, resource quotas, and status.

7. Delete a namespace

```
kubectl delete namespace dev
```

Deletes the namespace and everything inside it. (Be careful!)

NOTE

Can we rename a namespace? No, we cannot rename a namespace. Instead, create a new namespace and move resources there.
