Hands-On Lab (HOL): Understanding Kubernetes Namespace

Author: Sandeep Kaushik
Topic: Namespace in Kubernetes

Learning Objective

By the end of this HOL, learners will:

- Understand what a Namespace is in Kubernetes
- Know why namespaces are needed
- Learn how to create, view, and use namespaces
- Practice working with pods inside different namespaces
- Understand how to switch default namespace context

Learning Outcome

After completing this lab, participants will be able to:

- Explain namespace in simple terms
- Organize Kubernetes resources using namespaces
- Isolate environments like dev, test, and prod
- Use kubectl commands to manage namespaces
- Set and switch default namespaces in Kubernetes context

What is a Namespace? (Simple Explanation)

Imagine a **big building** 🏢 with many **rooms** inside it:

- The building = Kubernetes Cluster
- The rooms = Namespaces
- People in each room = Pods, Services, Deployments, etc.

Even though everyone is in the same building, each room is **separate and organized**.

In Simple Words:

A **Namespace** is a logical partition inside a Kubernetes cluster that helps to:

- Separate resources
- Avoid name conflicts
- Manage access control
- Organize environments (dev, test, prod)

Why Do We Need Namespace?

We need namespaces because:

1. Resource Isolation (dev resources don't affect prod)
2. Better Organization
3. Security and Access Control
4. Team-wise separation
5. Environment separation (dev / test / prod)

Example:

- Namespace: dev → developers use it
- Namespace: test → testers use it
- Namespace: prod → production workloads

Hands-On Lab (HOL)

Step 1: View Existing Namespaces

```
kubectl get namespace
```

Step 2: View Pods in kube-system Namespace

```
kubectl get pods --namespace=kube-system
```

Step 3: Create a New Namespace

```
kubectl create namespace dev
```

Step 4: Verify Namespace Creation

```
kubectl get namespace
```

Step 5: Create a Pod in dev Namespace

```
kubectl run pod1 --image=nginx --namespace=dev
```

Step 6: Check Pods in Default Namespace

```
kubectl get pods
```

You will not see pod1 here

Step 7: Check Pods in dev Namespace

```
kubectl get pods --namespace=dev
```

Now you will see pod1

Step 8: Set dev as Default Namespace

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

Step 9: Get Pods (Without Namespace Flag)

```
kubectl get pods
```

Now it shows pods from dev namespace by default

Step 10: Switch Back to Default Namespace

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

Step 11: View Command History

```
history
```

Real-Life Example

Real Life	Kubernetes
Building	Cluster
Room	Namespace
People	Pods
Cupboard	Resource limits

Summary

- Namespace = Logical separation inside cluster
- Helps in management, security, and organization
- Same cluster, multiple environments
- Default namespace = `default`
- System namespace = `kube-system`

One-Line Definition (For Interview)

"A namespace in Kubernetes is a logical isolation mechanism used to organize, secure, and manage resources within a single cluster."