

namespaces

Antonio Espín Herranz

Contenidos

- namespaces

namespace

En Kubernetes, un **namespace** es una manera de organizar y dividir los recursos dentro de un cluster. Es como crear "compartimentos" o "espacios" separados para que diferentes grupos, proyectos o aplicaciones puedan trabajar independientemente sin interferencias entre sí.

Usos principales de los namespaces:

- 1. Aislamiento lógico:** Permite separar recursos como Pods, Servicios, Deployments, etc., de diferentes equipos o aplicaciones en un cluster compartido.
- 2. Gestión simplificada:** Puedes aplicar configuraciones específicas (como políticas de acceso) a un namespace en lugar de a cada recurso individualmente.
- 3. Escalabilidad:** Ayuda a administrar mejor los clusters grandes, dividiendo los recursos en espacios más manejables.

namespace

Namespace predeterminados en Kubernetes:

- **default:** Se usa para los recursos que no tienen un namespace especificado.
- **kube-system:** Contiene los recursos críticos del sistema, como los componentes principales de Kubernetes (por ejemplo, el controlador de DNS).
- **kube-public:** Un namespace accesible públicamente, útil para recursos compartidos en el cluster.
- **kube-node-lease:** Contiene objetos relacionados con el mecanismo de heartbeats de los nodos.

namespaces

- Kubernetes soporta crear varios clústeres virtuales en un mismo clúster físico
 - Estos clústeres virtuales se llaman espacios de nombres (namespaces)
- Están pensados para casos de uso con muchos usuarios distribuidos entre múltiples equipos o proyectos
 - En clústeres pequeños (como los de esta asignatura) no tienen mucho sentido
- Proporcionan aislamiento lógico entre clústeres
 - Se soporta cualquier número de namespaces en un mismo clúster
 - Los nombres de los recursos deben ser únicos dentro de un namespace
 - Dos namespaces distintos pueden nombrar a un recurso de la misma forma.
- A pesar de este aislamiento, como veremos, recursos en distintos namespaces pueden comunicarse entre sí

Ejemplo

- Crear recurso con un namespace específico:

```
apiVersion: v1
kind: Pod
metadata:
  name: mi-pod
  namespace: mi-namespace
spec:
  containers:
  - name: mi-contenedor
    image: nginx:latest
```