

PODS with Labels

Labels:-

- Labels helps in grouping
- Kubernetes arranges pods with the help of labels

Front-End
POD 1

Front End
POD 2

Backend
POD-3

Backend
POD-4

Think of a situation where we have hundred of Pods with multiple Replicas.

- Multiple labels can be assigned to any resource
- Labels can be assigned at the time of resource creation
OR
- You can assign or change label value after creation of resource.

To show labels:-

- `kubectl get pods --show-labels`
- `kubectl get pods -L env`
- `kubectl get pods -L env, type`
- `kubectl get pods -l type=frontend --show-labels`

Demo = ??

➤ How to assign label when resource already created?

✱ kubectl label pods first-pod
(label) (name)
type=middleware env=azure

➤ How to change the existing label?
use the above command with
--overwrite

✱ kubectl label pods first-pod
type: front --overwrite.

How to remove label = ?

✱ kubectl label pods first-pod type-
(delete)

Label Selection Criteria:

```
kubectl get pods -l type = frontend  
--show-labels
```

✗ Not Equal to
type != frontend ← Criteria

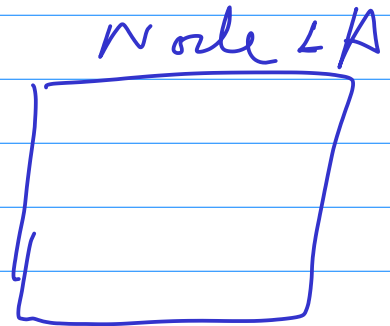
✗ -l '!env' --show-labels

✗ -l 'type in (frontend, backend)'
--show-labels

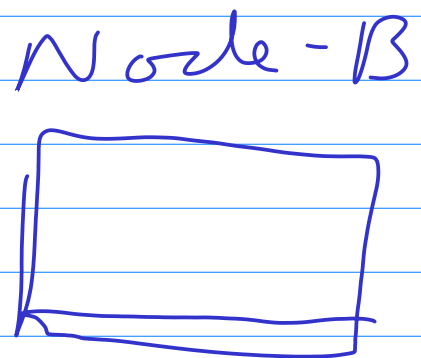
✗ -l 'type not in (frontend, backend)'
--show-labels

POD with Node Selector:

Label: Ram=High



Label: CPU=High



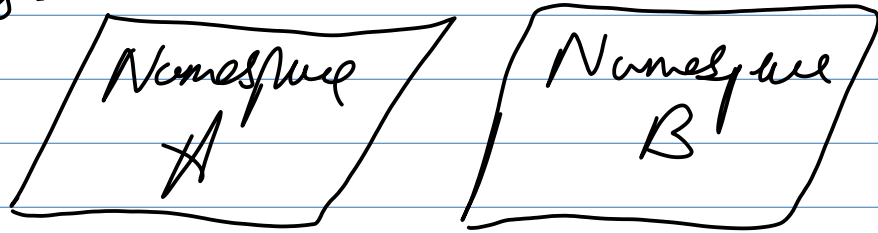
* We want a pod to run on a specific Node -

* kubectl label node minikube ram=high

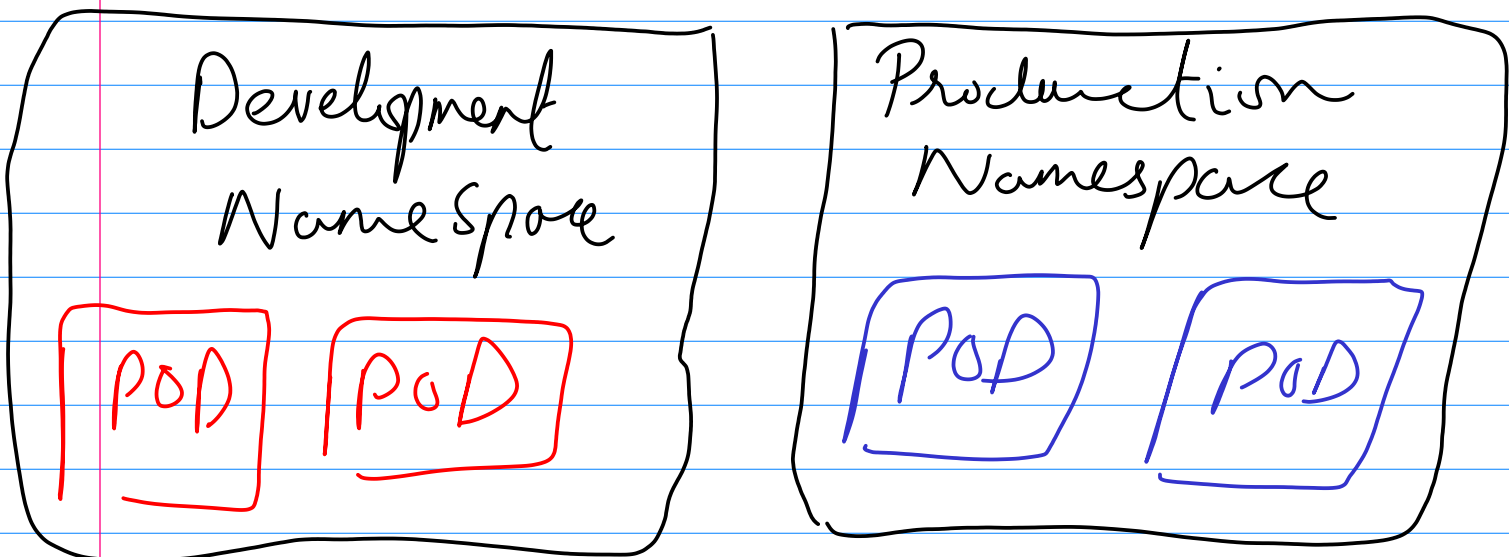
Namespaces :

• Kubernetes also group objects into namespaces.

- Namespace is a kind of virtual box which isolate resources from other namespaces.



- You can separate the scope of resource using namespace



Project-A

Namespace

Project-B

Namespace

Boundary of Project
is Namespace

Command:-

- kubectl create namespace development
- kubectl create namespace production

-
- ❖ kubectl get namespace
 - ❖ kubectl get pods --namespace=development
 - ❖ kubectl get pods --all-namespaces
 - ❖ kubectl delete namespace development
 - ❖ kubectl delete pods -l type=backend
-