

## Part 1: Namespaces

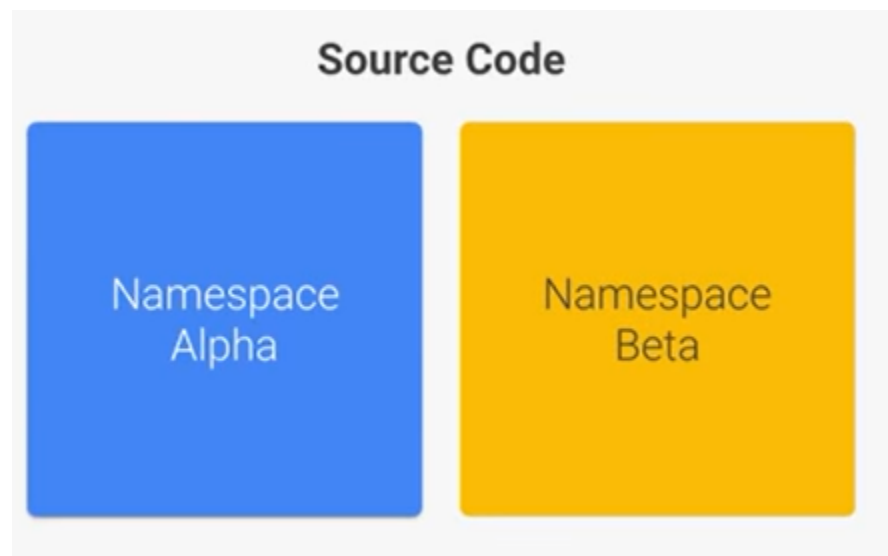
### What is a Namespace?

Everything we've done up to this point is great for individual use if we want to simply deploy a web server. However, from a business perspective that's not scaleable. However, as you start to deploy new resources to kubernetes, small tasks start to get more complicated.

If you have multiple teams of a business creating multiple deployments, those deployments cannot have the same name.

If you have 1000's of pods between many teams, just managing them could take time, let alone administering them.

Namespaces are essentially clusters within our Minikube cluster. You can have multiple namespaces and they're all isolated from one another. This gives us better control over which deployments and services belong to who.



There is no “right” way to implement namespaces in a business, they're only purpose is to have strict separation between resources. If you have multiple teams using kubernetes in a business, you don't want the sales department to have access to accounting data.

We use separate namespaces to keep apart services that should be prevented from sharing that should be isolated from other services.

## Cross Communication

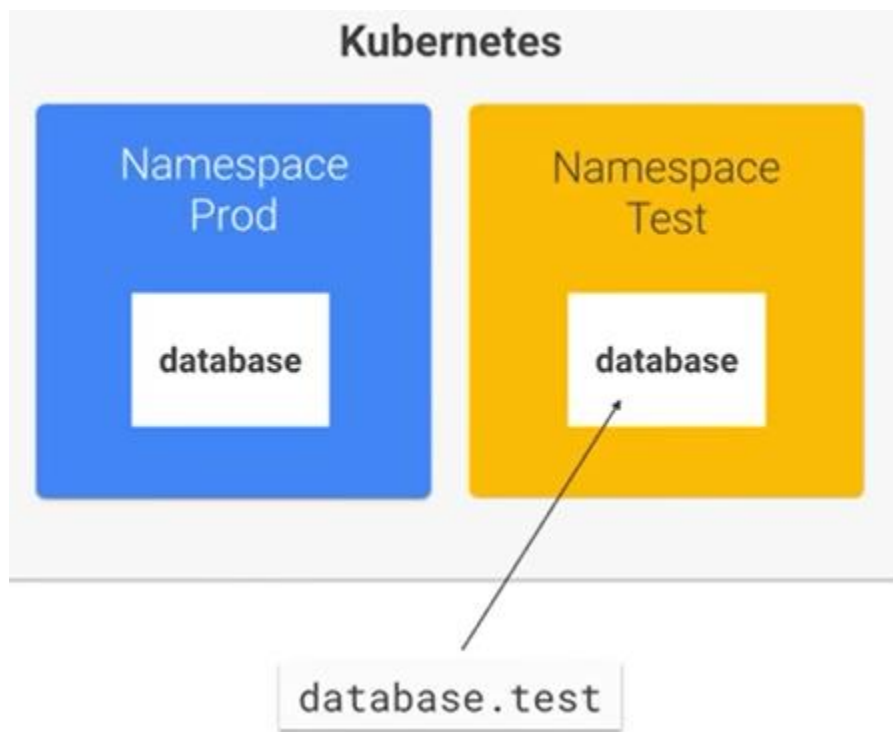
Although they are isolated, namespaces can access services in a different namespace with proper access. If two database teams are in separate namespaces, but need access to the same big data, they can do so through **cross namespace communication**.

CNC uses Services in Kubernetes, specifically DNS, to associate with a specific naming pattern. Essentially, you need the **service name plus the namespace** you're wanting to reach.

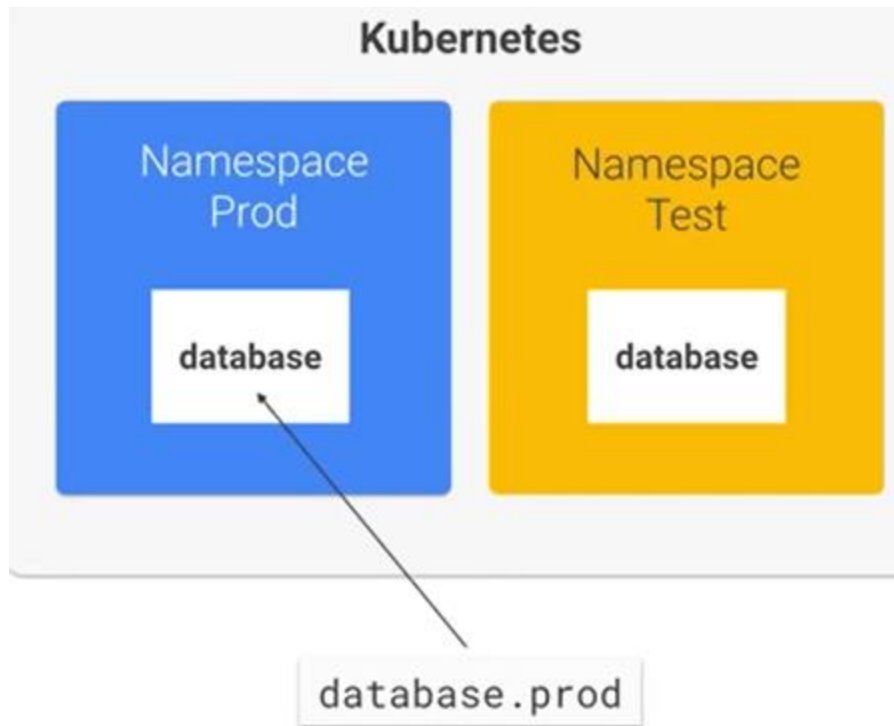
`<Service Name>.<Namespace Name>`

For example if you're going from Namespace Prod to Namespace Test, your address would be:

*database.test*

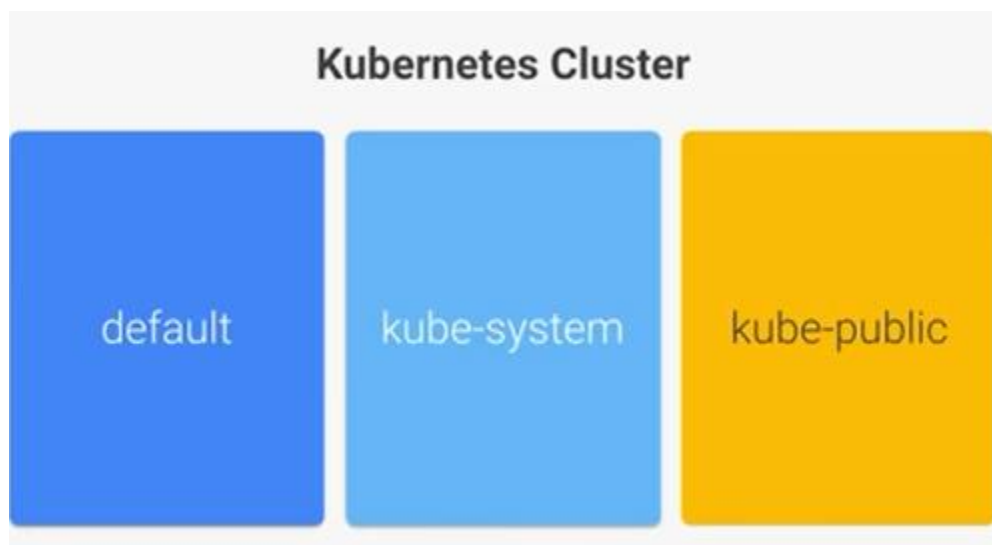


and vice versa...



### 3 Primary Namespaces

kubectl get ns



Default - The default namespace is for objects with no other namespace. If you create a deployment or service and don't specify a namespace, it goes here. In a professional environment you should stay away from using the default namespace and instead you should create multiple namespaces for each

team. You can use these to segment services into manageable chunks.

Kube-public - This is created automatically and is readable by all users (including those not authenticated). This namespace is mostly reserved for cluster usage for resources that should be readable publicly throughout the whole cluster. Naming this "Kube-public" is only a convention, not a requirement.

kube-system - The namespace for objects created by the Kubernetes system.

## Hands On - Create a Namespace

`kubectl create namespace <name>`

```
dominickhrndz314@minikube-c1-training:~$ kubectl create namespace doms-test1
namespace/doms-test1 created
dominickhrndz314@minikube-c1-training:~$ kubectl get ns
NAME                STATUS   AGE
default             Active   87m
doms-test1          Active   12s
kube-node-lease     Active   87m
kube-public         Active   87m
kube-system         Active   87m
dominickhrndz314@minikube-c1-training:~$
```

Or you can do it via YAML:

`nano doms-test2.yaml`

```
apiVersion: v1

kind: Namespace

metadata:
  name: doms-test2
```

`kubectl apply -f namespaces.yaml`

## Hands on - Changing Namespaces

Because namespaces are isolated, you may not see deployments in your current namespace because they're in another one.

**Kubens** is an installable Kubernetes tool that allows us to view our namespaces and then switch to the correct one.

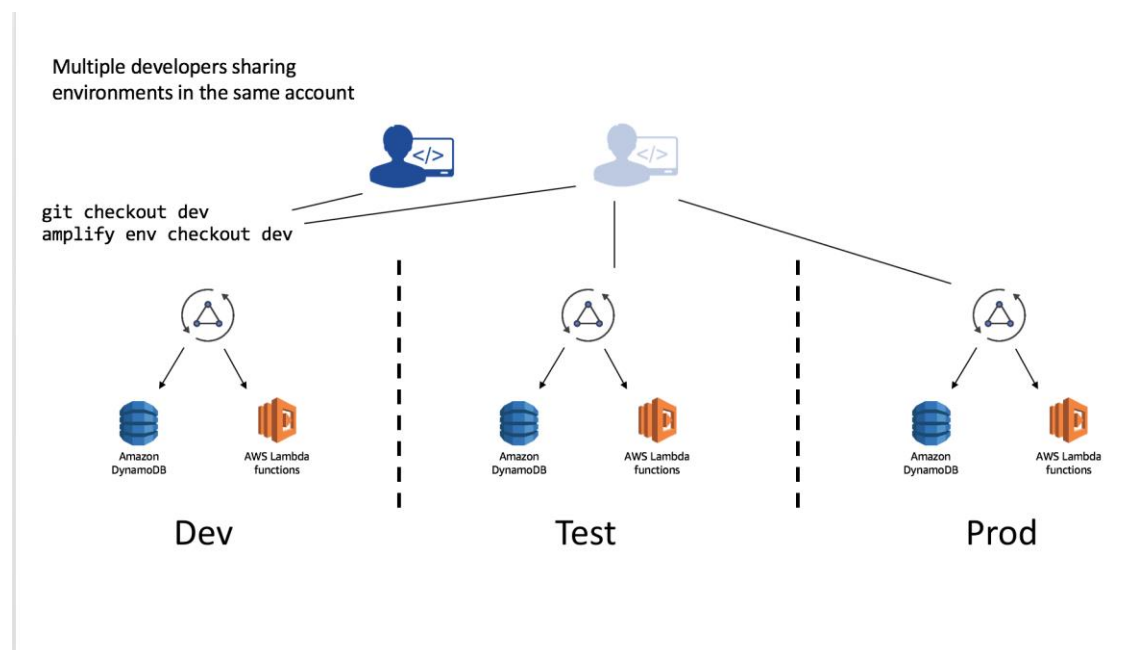
You should see the active namespace highlighted after you run the command.

```
dominickhrndz314@minikube-c1-training:~$ kubens
default
doms-test1
doms-test2
kube-node-lease
kube-public
kube-system
```

You can run `kubens kube-system` to switch between different namespaces.

```
dominickhrndz314@minikube-c1-training:~$ kubens doms-test2
Context "minikube" modified.
Active namespace is "doms-test2".
dominickhrndz314@minikube-c1-training:~$
```

## When Should Businesses Use Namespaces?



A new phone app may have developers using multiple namespaces for different objectives. A dev namespace for developing new features on the phone app. A test namespace for determining quality and possible bugs. Last, a prod namespace that serves that app over the internet to mobile users.

There are 4 major stages that most businesses go through to determine when to isolate clusters:

### Small teams

No more 5-10 projects

Use the default namespace and divide that between dev and prod environments. They're most likely using something like minikube for testing their development.



### Medium sized teams

More than 10 projects

These businesses will create a dev/prod environment

...And also create namespaces for smaller teams.

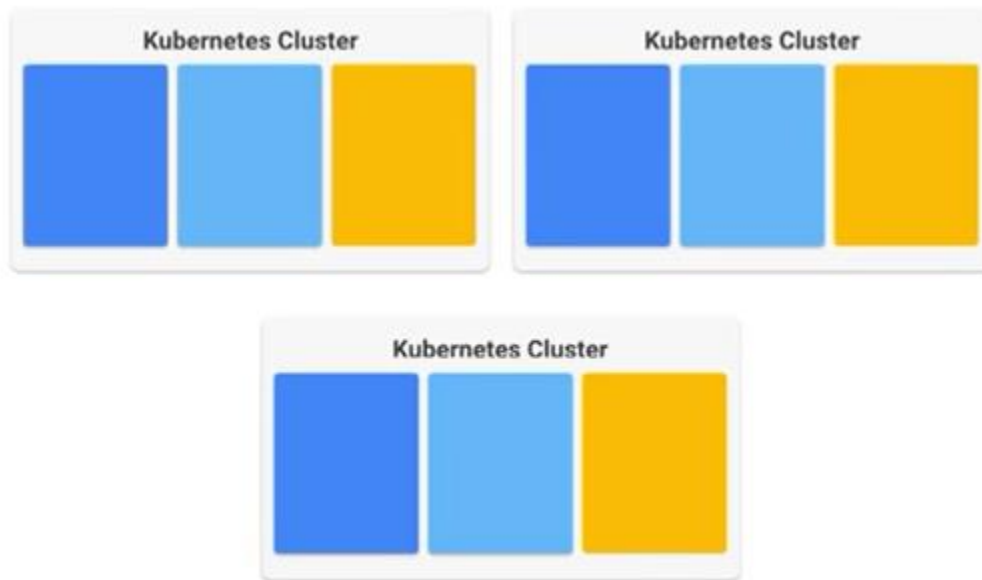


### Large Companies/Enterprises

Teams are working on multiple features and service contracts, no idea what other teams do.

At this level each team definitely needs their own namespaces and possibly separate clusters. Multicluster deployments make billing easier and reduces the blast radius for poorly managed applications.

Locks for namespaces should be used to keep people out who shouldn't have access to.



## Hands on - Assign a Pod to Your Namespace

```
apiVersion: v1
kind: Pod
metadata:
  name: doms-pod
  namespace: doms-test1
  labels:
    name: doms-pod
spec:
  containers:
  - name: doms-pod
    image: nginx
```

nano doms-pod.yaml

apiVersion: v1

kind: Pod

metadata:

```
name: doms-pod

namespace: doms-test1

labels:

  name: doms-pod

spec:

  containers:

  - name: doms-pod

    image: nginx
```

`kubectl apply -f doms-pod.yaml`

If you look for your pod, you may not find it. You need to find the pod in the correct namespace.

```
dominickhrndz314@minikube-c1-training:~$ kubectl get pods
No resources found in doms-test2 namespace.
dominickhrndz314@minikube-c1-training:~$
```

You can either switch to the correct one or run the following ad-hoc command.

`kubectl get pods --namespace=doms-test1`

```
dominickhrndz314@minikube-c1-training:~$ kubectl get pods --namespace=doms-test1
NAME        READY   STATUS    RESTARTS   AGE
doms-pod    1/1     Running   0           3m28s
dominickhrndz314@minikube-c1-training:~$
```