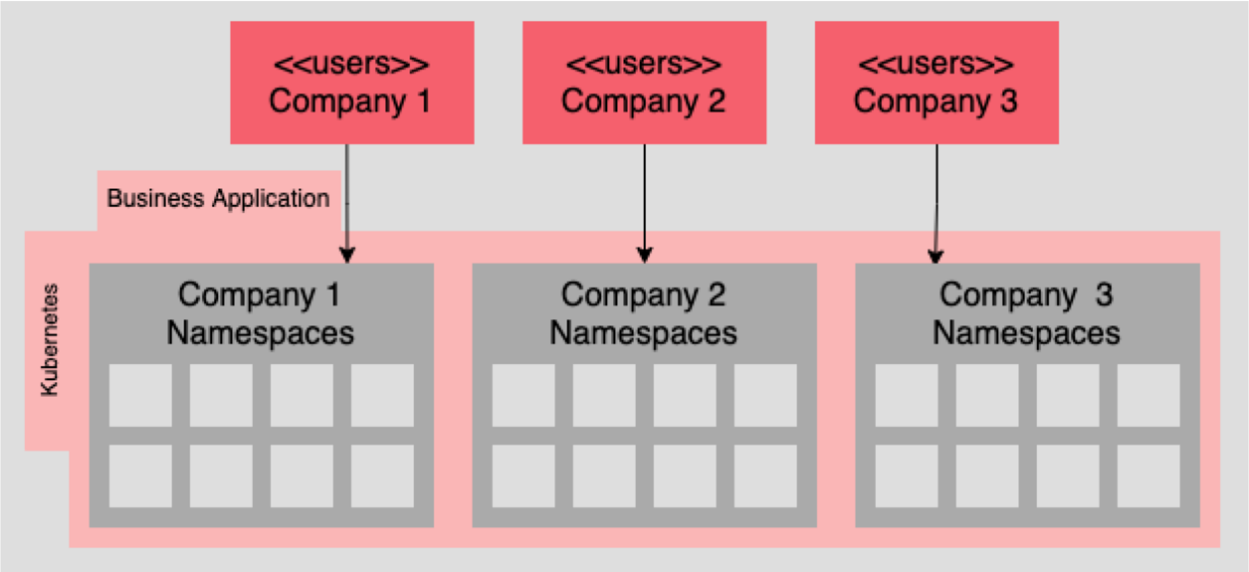
NameSpace in Kubernetes

Internet definition : Namespaces are a way to organize clusters into virtual sub-clusters — they can be helpful when different teams or projects share a Kubernetes cluster. Any number of namespaces are supported within a cluster, each logically separated from others but with the ability to communicate with each other.

As you can see above the definition is absolutely correct but hard to understand .
So I will simplify namespace in kubernetes in simple terms:



Namespace is provided in kubernetes because suppose in your company there are two departments dev and testing . An application is deployed in kubernetes cluster within a default namespace so if the testing team makes some changes in the application it will overwrite the changes for the dev team's application also because that application is present in the same namespace and with the same name . Namespace is like a folder in a computer , if two files present in the same folder then it will overwrite the changes. That's if we want a file with the same name and working then we have to make two different folders. So it will not cause the problem.

It is better to make a new namespace for new application deployment so every application has its own space. You can create namespace by the writing following command:

```
kubectl create namespace <insert-namespace-name-here>
```

Or you can write yml file for that like this :

Create a new YAML file called `my-namespace.yaml` with the contents:

```
apiVersion: v1

kind: Namespace

metadata:

 name: <insert-namespace-name-here>
```

Then run:

```
kubectl create -f ./my-namespace.yaml
```

See In kubernetes , by default probably 4 namespace allocated in a cluster and you can see those namespace by the entering the following command :

```
kubectl get ns      // —> ns represent the namespace
```

And this command will show those 4 namespaces which are pre-allocated

```
NAME              STATUS   AGE
default           Active   17d
Kube-node-lease   Active   17d
kube-public       Active   17d
kube-system       Active   17d
```

All of your deployments are created in default namespace unless you create your own namespace which is the best practice .