18_Pods in YAML :

YAML in Kubernetes

pod-definition.yml

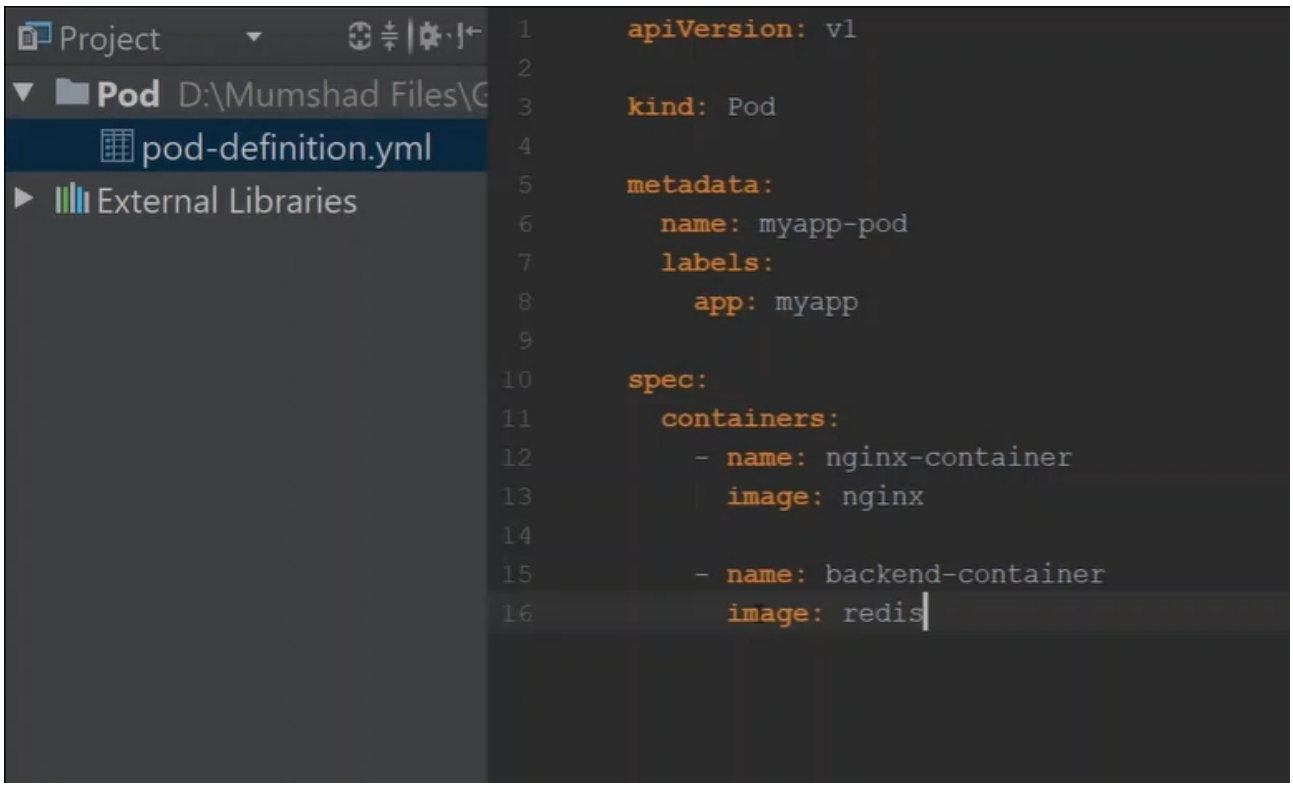
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
      app: myapp
      type: front-end
spec:
  containers:
    - name: nginx-container
      image: nginx
```

Kind	Version
POD	v1
Service	v1
ReplicaSet	apps/v1
Deployment	apps/v1

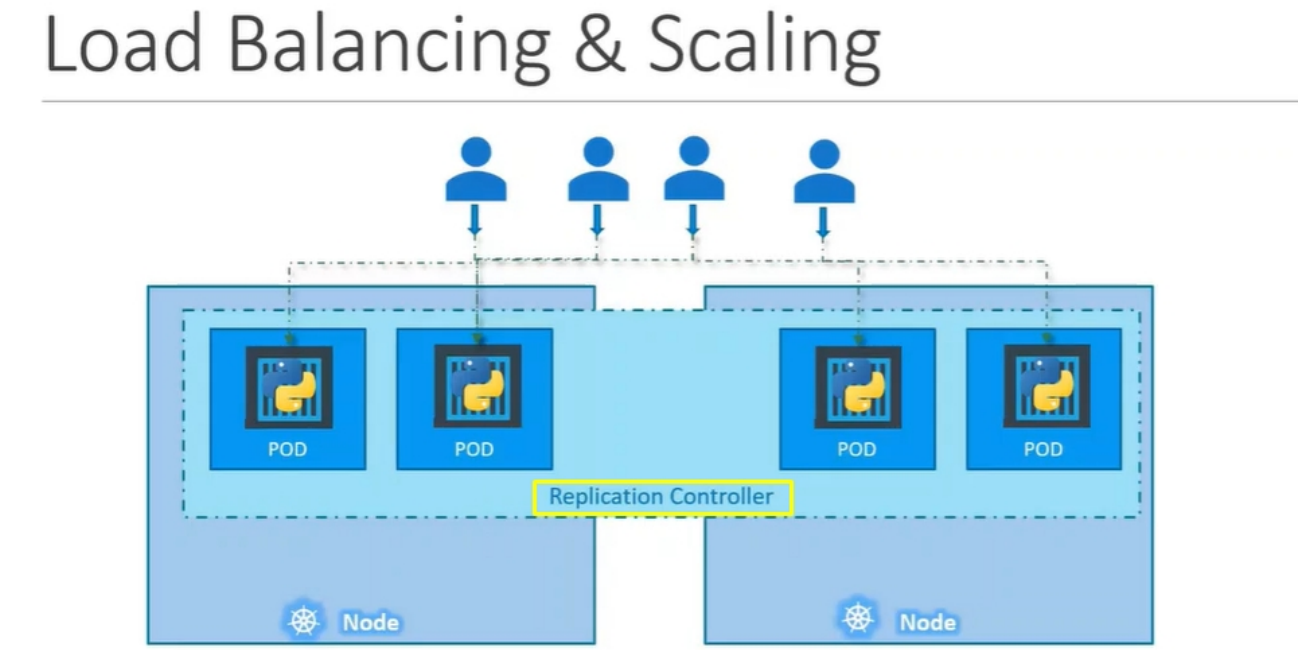
```
kubectl create -f pod-definition.yml
```

- These are the 4 basic attributes of the POD while defining it in YAML.

Demo :



24_Replication Controller



- It spans the cluster across nodes.
- Replication controller is older technology, ReplicaSet is the newer one.
- ReplicaSet is recommended to be used.

- Defining a ReplicationController.

rc-definition.yml

```yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: myapp-rc
  labels:
      app: myapp
      type: front-end
spec:
  template:
    metadata:
      name: myapp-pod
      labels:
        app: myapp
        type: front-end
    spec:
      containers:
      - name: nginx-container
        image: nginx

  replicas: 3
```

pod-definition.yml

```yaml
apiVersion: v1
kind: Pod
```

```
> kubectl create -f rc-definition.yml
replicationcontroller "myapp-rc" created

> kubectl get replicationcontroller
NAME       DESIRED   CURRENT   READY     AGE
myapp-rc   3         3         3         19s

> kubectl get pods
NAME             READY     STATUS    RESTARTS   AGE
myapp-rc-4lvk9   1/1       Running   0          20s
myapp-rc-mc2mf   1/1       Running   0          20s
myapp-rc-px9pz   1/1       Running   0          20s
```

- Defining a ReplicaSet.

replicaset-definition.yml

```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: myapp-replicaset
  labels:
      app: myapp
      type: front-end
spec:
  template:
    metadata:
      name: myapp-pod
      labels:
        app: myapp
        type: front-end
    spec:
      containers:
      - name: nginx-container
        image: nginx

  replicas: 3
  selector:
    matchLabels:
      type: front-end
```

pod-definition.yml

```yaml
apiVersion: v1
kind: Pod
```

```
> kubectl create -f replicaset-definition.yml
replicaset "myapp-replicaset" deleted

> kubectl get replicaset
NAME              DESIRED   CURRENT   READY     AGE
myapp-replicaset  3         3         3         19s

> kubectl get pods
NAME                     READY     STATUS    RESTARTS   AGE
myapp-replicaset-9dd19   1/1       Running   0          45s
myapp-replicaset-9jtpx   1/1       Running   0          45s
myapp-replicaset-hq84m   1/1       Running   0          45s
```

- ReplicaSet monitors the existing pods in case they are created before creating a ReplicaSet.
- It uses selector -> matchLabels filter to detect and monitor the existing pod.

- Scaling the ReplicaSet

Scale

```
> kubectl replace -f replicaset-definition.yml

> kubectl scale --replicas=6 -f replicaset-definition.yml

> kubectl scale --replicas=6 replicaset myapp-replicaset
```

TYPE NAME

replicaset-definition.yml

```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: myapp-replicaset
  labels:
      app: myapp
      type: front-end
spec:
  template:
    metadata:
      name: myapp-pod
      labels:
        app: myapp
        type: front-end
    spec:
      containers:
      - name: nginx-container
        image: nginx
  replicas: 6
  selector:
    matchLabels:
      type: front-end
```

- Commands Review.

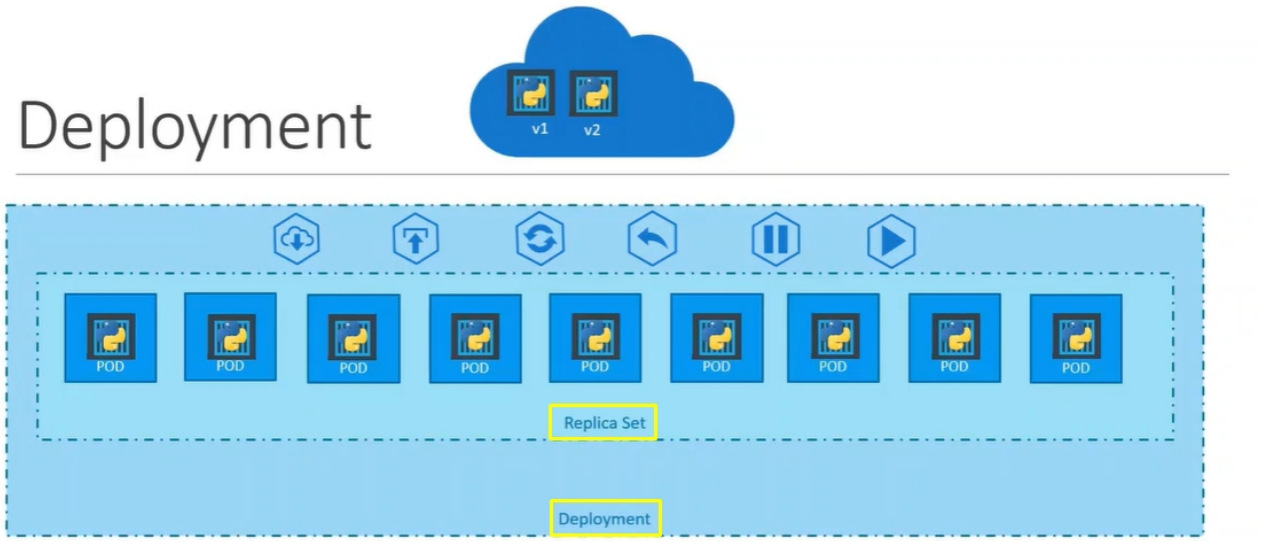
commands

```
> kubectl create -f replicaset-definition.yml

> kubectl get replicaset

> kubectl delete replicaset myapp-replicaset
```

*Also deletes all underlying PODs

```
> kubectl replace -f replicaset-definition.yml

> kubectl scale -replicas=6 -f replicaset-definition.yml
```

27_Deployments



- Definition of Deployment :

Definition

```
> kubectl create -f deployment-definition.yml
deployment "myapp-deployment" created
```

```
> kubectl get deployments
NAME              DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
myapp-deployment  3         3         3            3           21s
```

```
> kubectl get replicaset
NAME                         DESIRED   CURRENT   READY   AGE
myapp-deployment-6795844b58  3         3         3       2m
```

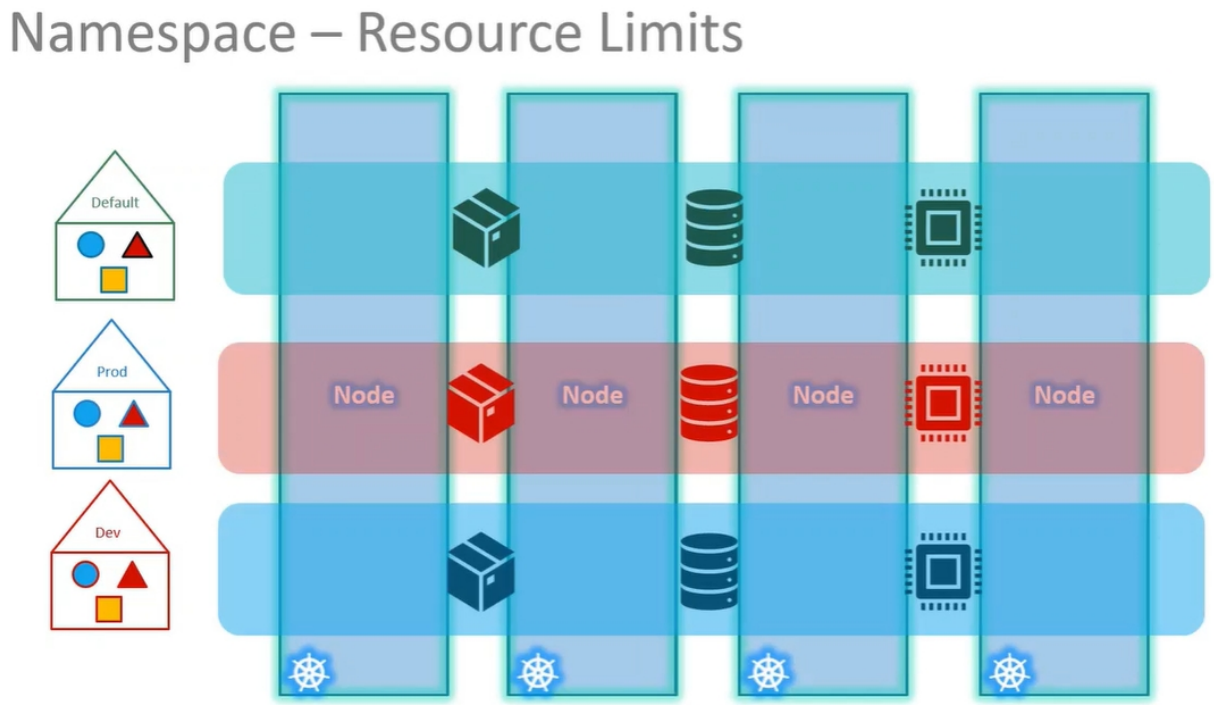
```
> kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
myapp-deployment-6795844b58-5rbjl  1/1     Running   0          2m
myapp-deployment-6795844b58-h4w55  1/1     Running   0          2m
myapp-deployment-6795844b58-lfjhv  1/1     Running   0          2m
```

deployment-definition.yml

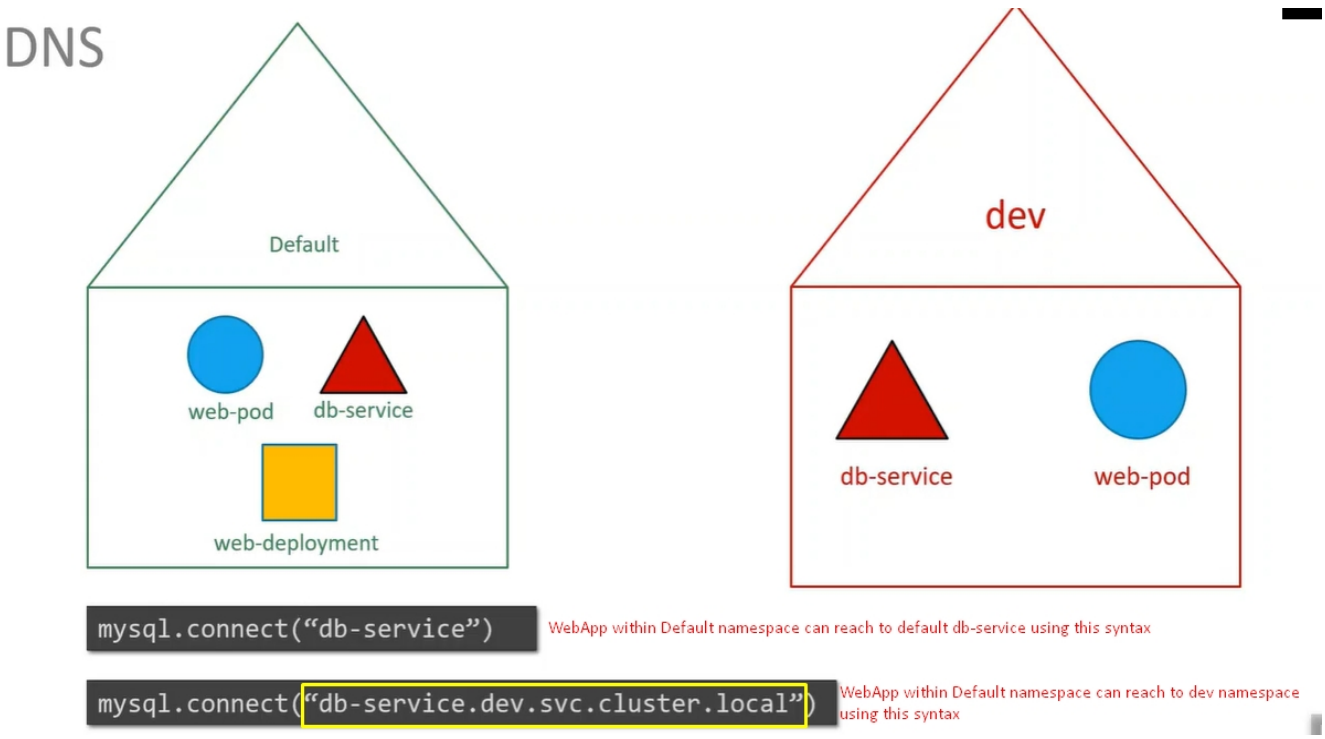
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: myapp-deployment
  labels:
      app: myapp
      type: front-end
spec:
  template:
    metadata:
      name: myapp-pod
      labels:
        app: myapp
        type: front-end
    spec:
      containers:
      - name: nginx-container
        image: nginx
  replicas: 3
  selector:
    matchLabels:
      type: front-end
```

30_NameSpaces

- We can assign a Quota to namespaces.



- Namespace reachability.



```
mysql.connect("db-service.dev.svc.cluster.local")
```


```
> kubectl get pods
NAME      READY     STATUS    RESTARTS   AGE
Pod-1     1/1       Running   0          3d
Pod-2     1/1       Running   0          3d
```

```
> kubectl get pods --namespace=kube-system
NAME                              READY     STATUS    RESTAR
coredns-78fcdf6894-92d52          1/1       Running   7
coredns-78fcdf6894-jx25g          1/1       Running   7
etcd-master                       1/1       Running   7
kube-apiserver-master             1/1       Running   7
kube-controller-manager-master    1/1       Running   7
kube-flannel-ds-amd64-hz4cf       1/1       Running   14
kube-proxy-4b8tn                  1/1       Running   7
kube-proxy-98db4                  1/1       Running   7
kube-proxy-jjrbs                  1/1       Running   7
kube-scheduler-master             1/1       Running   7
```

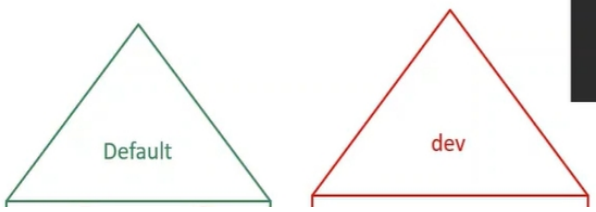
- Creating a Pod in different NameSpace.

```
> kubectl create -f pod-definition.yml
pod/myapp-pod created
```

```
> kubectl create -f pod-definition.yml --namespace=dev
pod/myapp-pod created
```

pod-definition.yml

```yaml
apiVersion: v1
kind: Pod

metadata:
 name: myapp-pod
 labels:
    app: myapp
    type: front-end
spec:
  containers:
  - name: nginx-container
    image: nginx
```



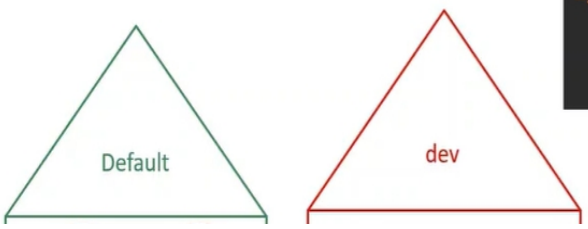
- Below is recommended way.

```
> kubectl create -f pod-definition.yml
pod/myapp-pod created
```

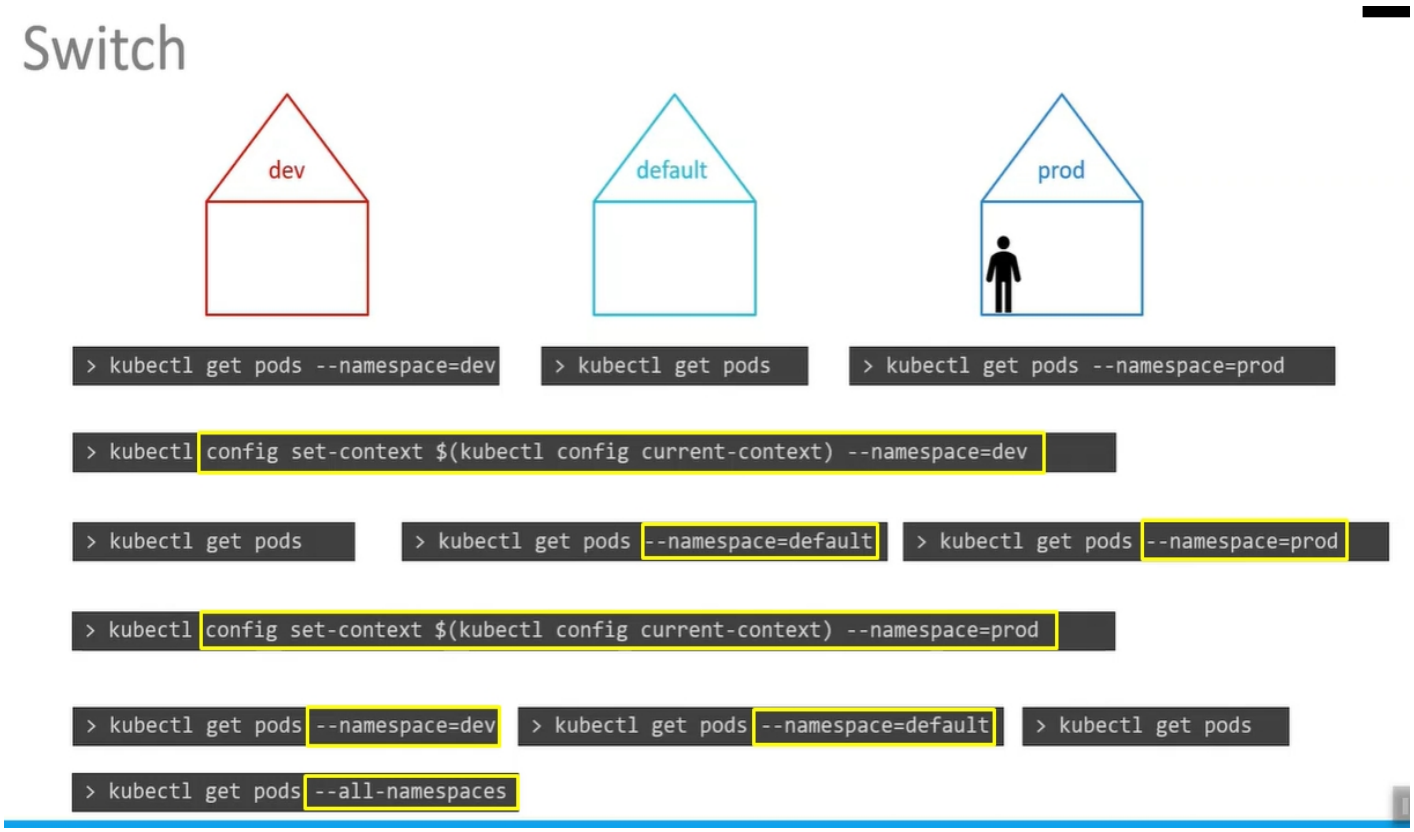
```
> kubectl create -f pod-definition.yml
pod/myapp-pod created
```

pod-definition.yml

```yaml
apiVersion: v1
kind: Pod

metadata:
 name: myapp-pod
 namespace: dev
 labels:
    app: myapp
    type: front-end
spec:
  containers:
  - name: nginx-container
    image: nginx
```



- Switching to different Namespace permanently.

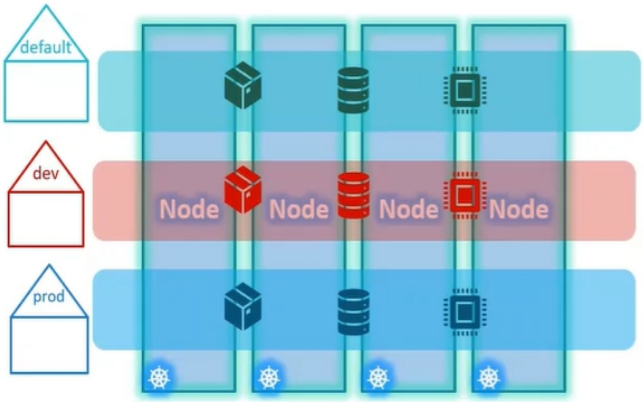


- Commands to switch the namespace.

```
> kubectl config set-context $(kubectl config current-context) --namespace=dev
```

- Assign a quota to namespaces.

Resource Quota



Compute-quota.yaml

```yaml
apiVersion: v1
kind: ResourceQuota
metadata:
    name: compute-quota
    namespace: dev
spec:
  hard:
    pods: "10"
    requests.cpu: "4"
    requests.memory: 5Gi
    limits.cpu: "10"
    limits.memory: 10Gi
```

```
> kubectl create -f compute-quota.yaml
```