

# Different Parts in a Worker Node of Kubernetes

# PODs

# Assumptions

---



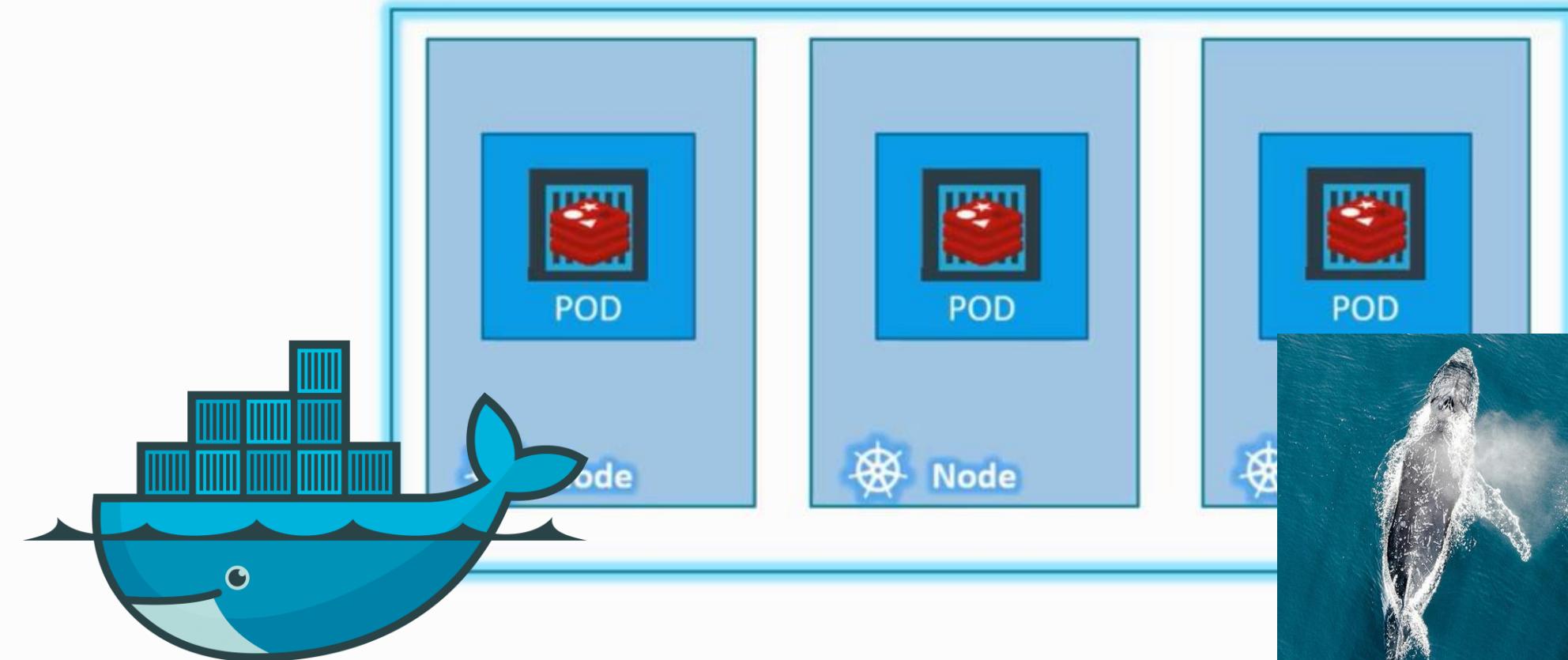
Docker Image



Kubernetes Cluster

# POD

---

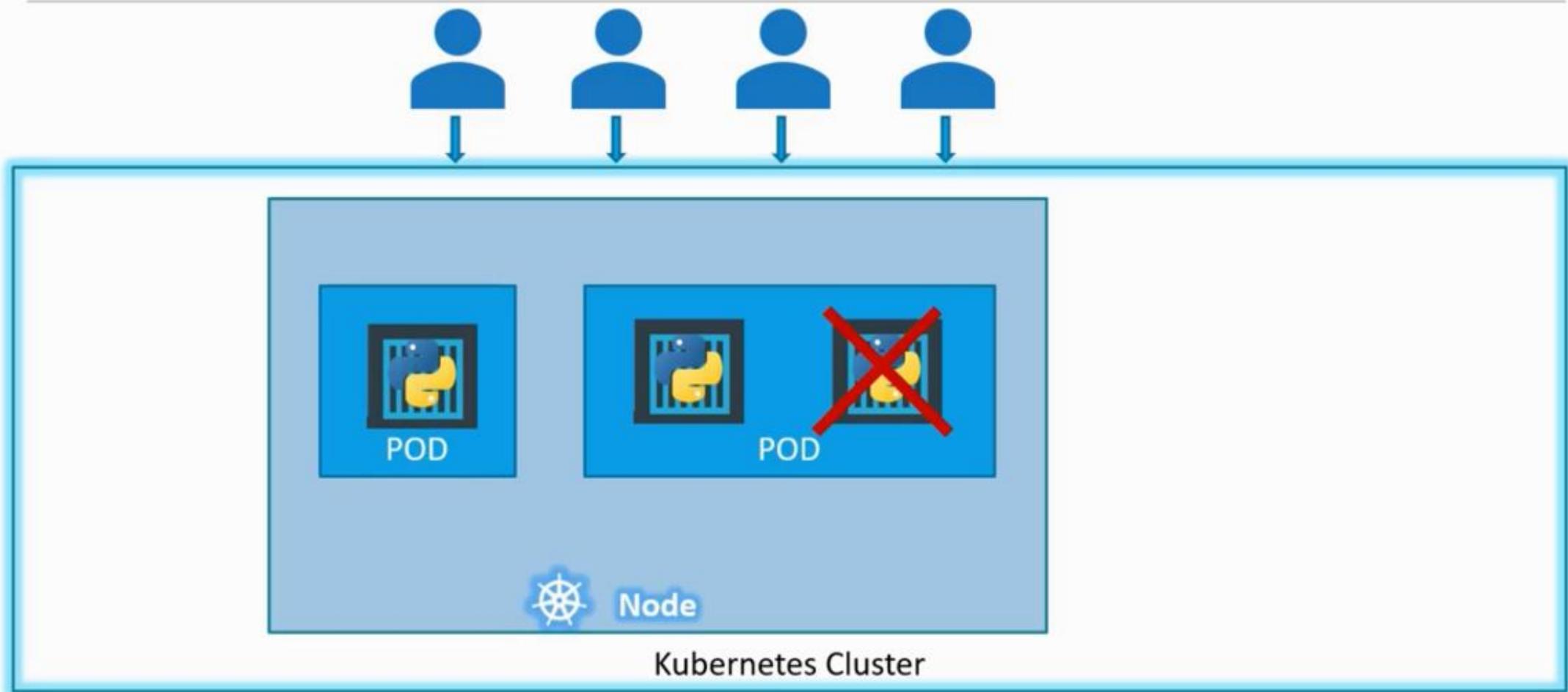


docker



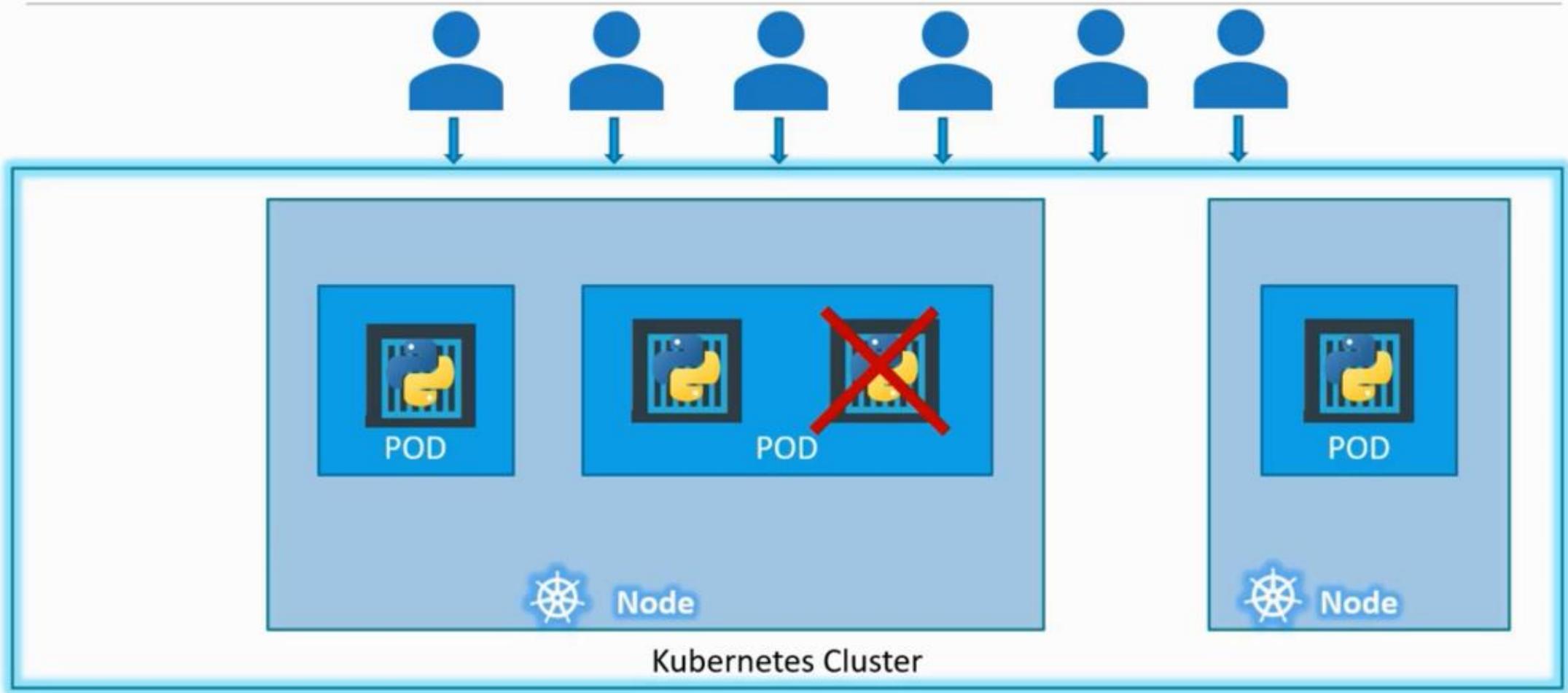
# POD

---



# POD

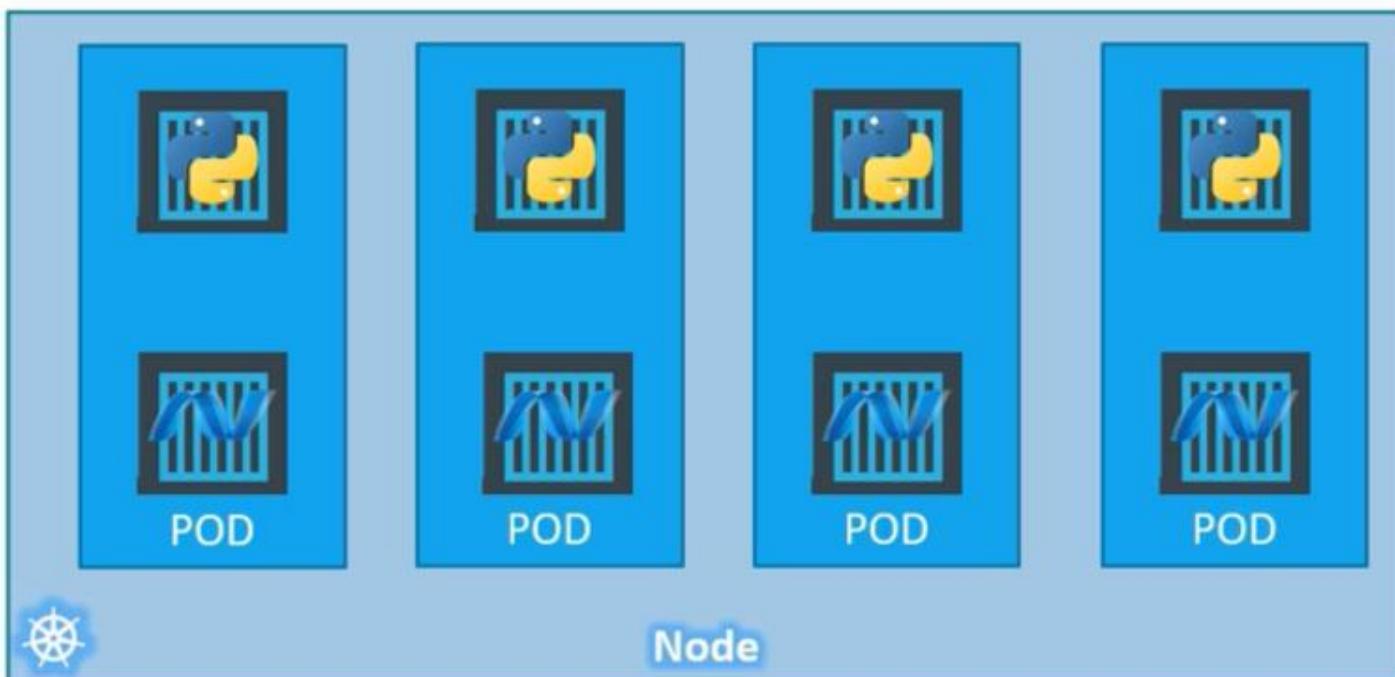
---



# PODs Again!

```
docker run python-app  
docker run python-app  
docker run python-app  
docker run python-app  
  
docker run helper -link app1  
docker run helper -link app2  
docker run helper -link app3  
  
docker run helper -link app4
```

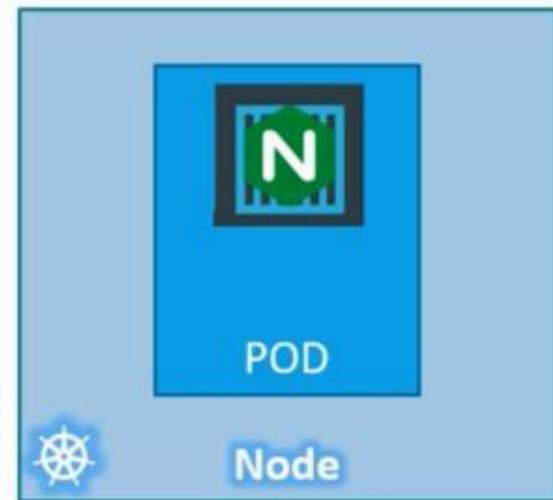
App	Helper	Volume
Python1	App1	Vol1
Python2	App2	Vol2



# kubectl

---

```
kubectl run nginx --image nginx
```



# kubectl

---

```
kubectl run nginx --image nginx
```

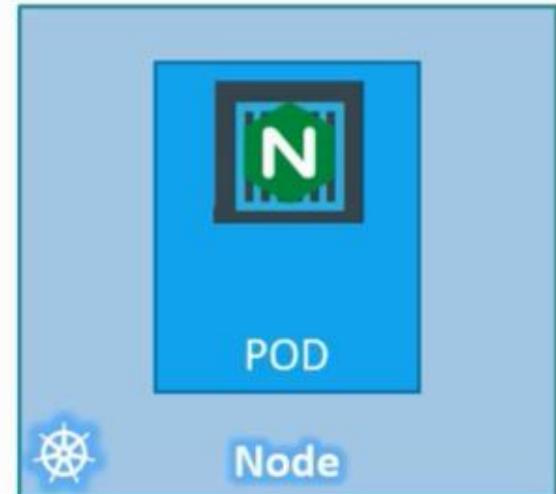
```
kubectl get pods
```

```
C:\Kubernetes>kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-8586cf59-whssr	0/1	ContainerCreating	0	3s

```
C:\Kubernetes>kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-8586cf59-whssr	1/1	Running	0	8s



# YAML in Kubernetes

---

```
pod-definition.yml
```

```
apiVersion:  
kind:  
metadata:
```

```
spec:
```

# YAML in Kubernetes

---

```
pod-definition.yml
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
spec:
```

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

# YAML in Kubernetes

---

```
pod-definition.yml
```

```
apiVersion: v1
kind: Pod

metadata:
  name: myapp-pod
  labels:
    app: myapp

spec:
```



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

# YAML in Kubernetes

---

```
pod-definition.yml
```

```
apiVersion: v1
kind: Pod

metadata:
  name: myapp-pod
  labels:
    app: myapp
```

```
spec:
```



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

# YAML in Kubernetes

---

pod-definition.yml

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

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

# YAML in Kubernetes

---

pod-definition.yml

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

1<sup>st</sup> Item in List

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

# YAML in Kubernetes

---

pod-definition.yml

```
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

# Commands

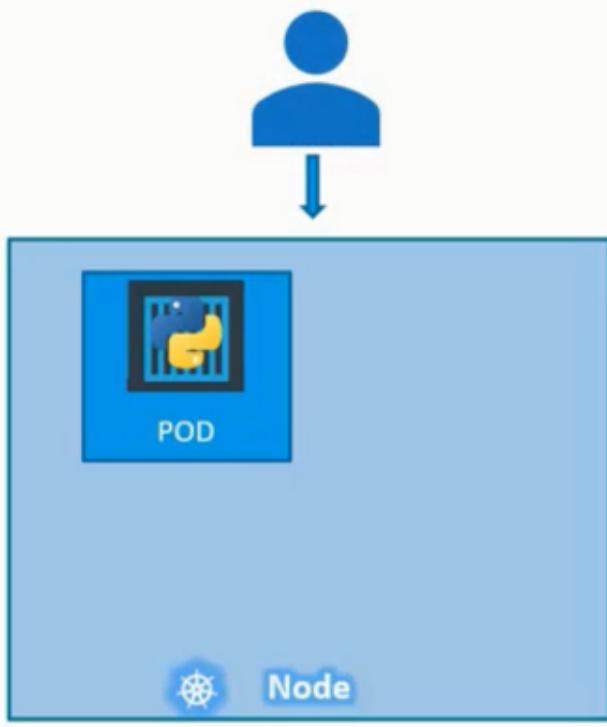
```
> kubectl get pods
```

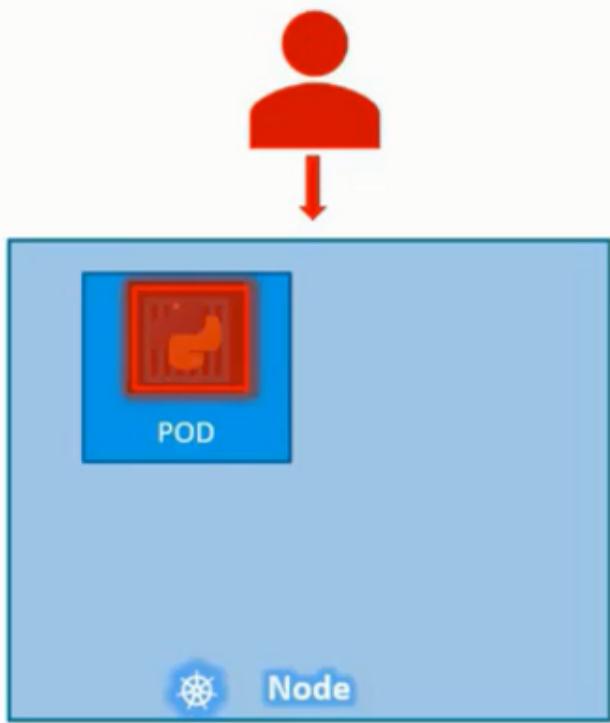
NAME	READY	STATUS	RESTARTS	AGE
myapp-pod	1/1	Running	0	20s

```
> kubectl describe pod myapp-pod
```

```
Name:           myapp-pod
Namespace:      default
Node:          minikube/192.168.99.100
Start Time:    Sat, 03 Mar 2018 14:26:14 +0800
Labels:         app=myapp
                name=myapp-pod
Annotations:   <none>
Status:        Running
IP:            172.17.0.24
Containers:
  nginx:
    Container ID:  docker://830bb56c8c42a86b4bb70e9c1488fae1bc38663e4918b6c2f5a783e7688b8c9d
    Image:          nginx
    Image ID:      docker-pullable://nginx@sha256:4771d09578c7c6a65299e110b3ee1c0a2592f5ea2618d23e4ffe7a4cab1ce5de
    Port:          <none>
    State:         Running
      Started:     Sat, 03 Mar 2018 14:26:21 +0800
    Ready:         True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-x95w7 (ro)
Conditions:
  Type      Status
  Initialized  True
  Ready       True
  PodScheduled  True
Events:
  Type  Reason          Age   From           Message
  ----  ----          ----  ----          -----
  Normal Scheduled      34s   default-scheduler  Successfully assigned myapp-pod to minikube
  Normal SuccessfulMountVolume 33s   kubelet, minikube  MountVolume.SetUp succeeded for volume "default-token-x95w7"
  Normal Pulling        33s   kubelet, minikube  pulling image "nginx"
  Normal Pulled         27s   kubelet, minikube  Successfully pulled image "nginx"
  Normal Created        27s   kubelet, minikube  Created container
  Normal Started        27s   kubelet, minikube  Started container
```

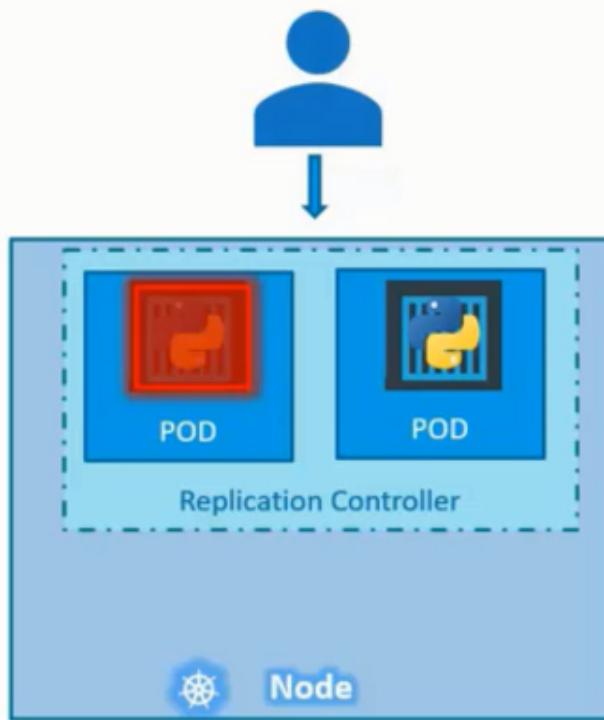
# ReplicaSet





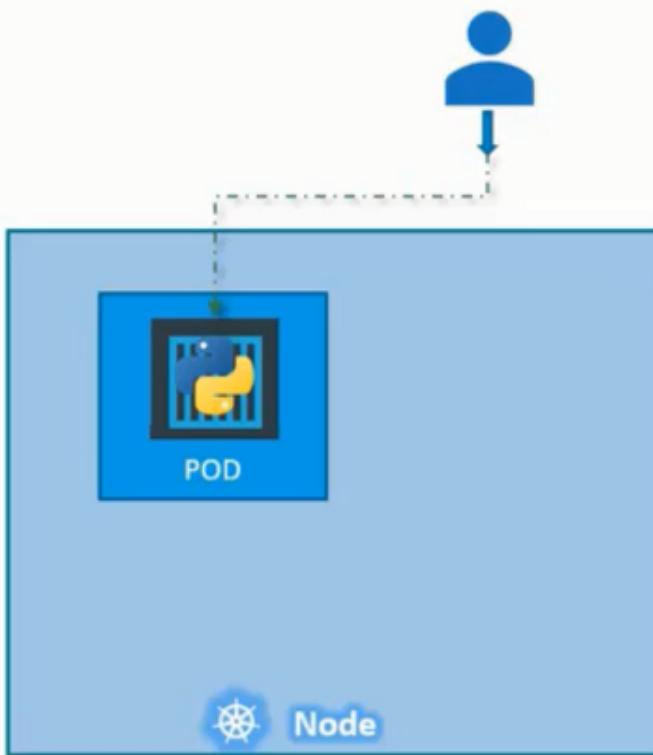
# High Availability

---



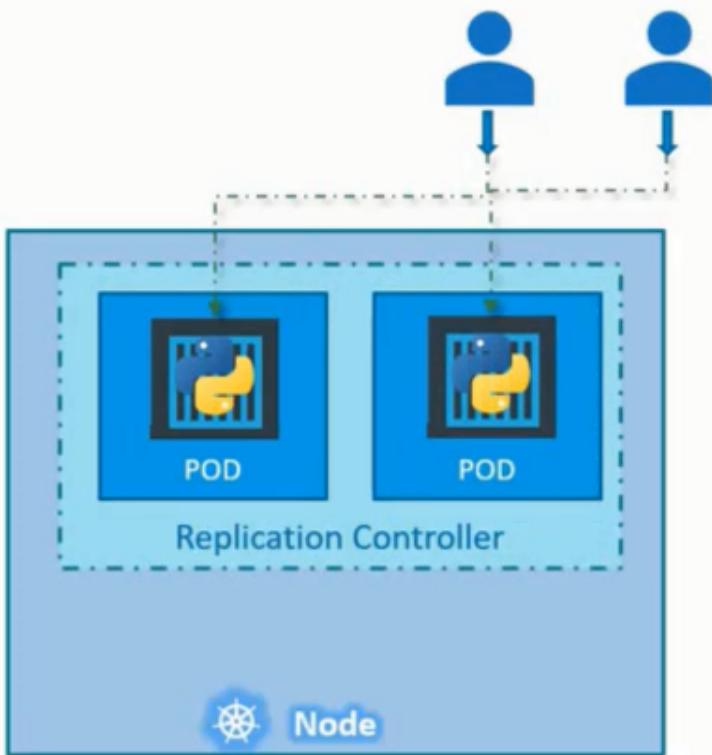
# Load Balancing & Scaling

---



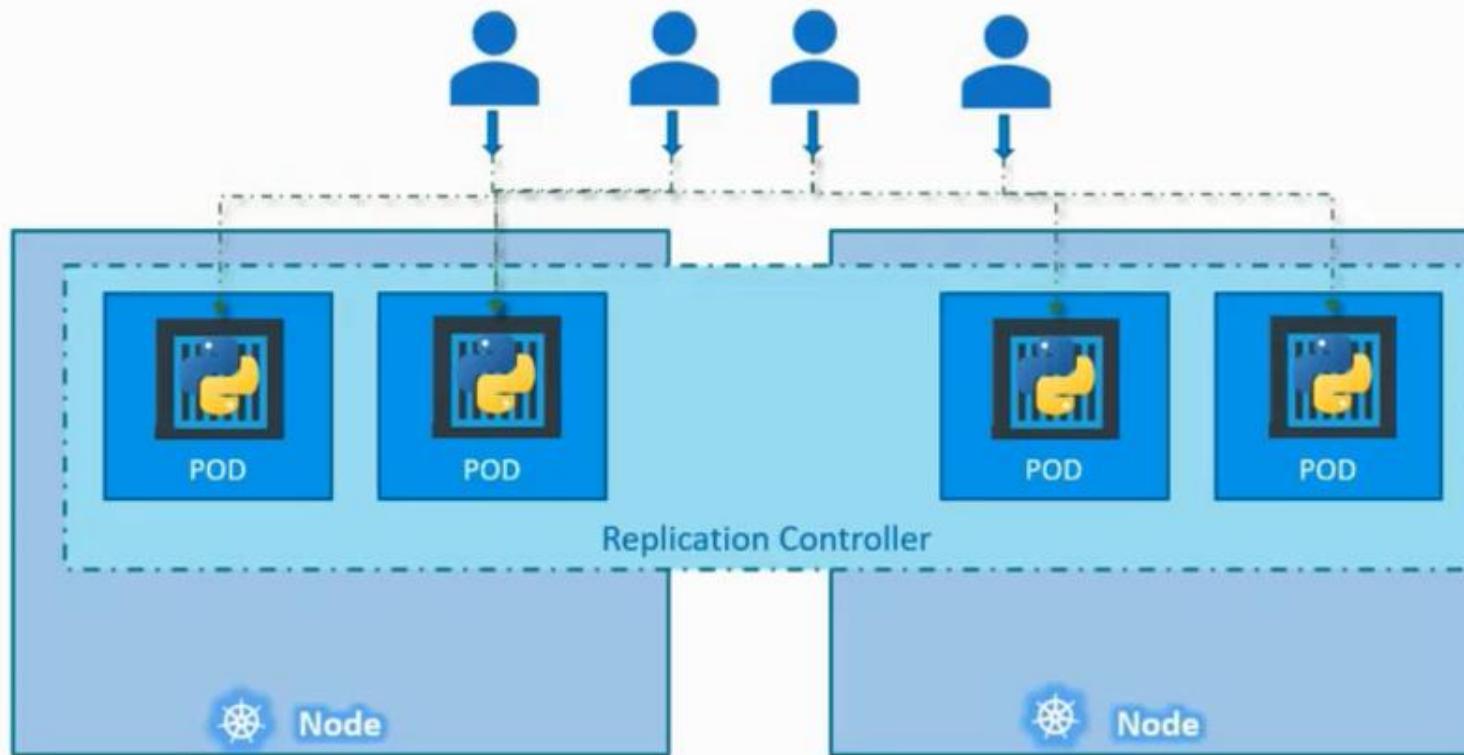
# Load Balancing & Scaling

---



# Load Balancing & Scaling

---



rc-definition.yml

apiVersion:

kind:

metadata:

spec:

```
rc-definition.yml
```

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: myapp-rc
  labels:
    app: myapp
    type: front-end
spec:
  template:
```



POD

```
rc-definition.yml
```

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: myapp-rc
  labels:
    app: myapp
    type: front-end
spec:
  template:
```

POD

```
pod-definition.yml
```

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

```
rc-definition.yml
```

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

```
pod-definition.yml
```

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

rc-definition.yml

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

pod-definition.yml

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

`rc-definition.yml`

```
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`

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

### rc-definition.yml

```
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

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

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

```
rc-definition.yml
```

```
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
```

```
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
```

### rc-definition.yml

```
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

```
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

### replicaset-definition.yml

```
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
```

### pod-definition.yml

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

### replicaset-definition.yml

```
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

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

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

### replicaset-definition.yml

```
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

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

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

```
> kubectl get replicaset
```

NAME	DESIRED	CURRENT	READY	AGE
myapp-replicaset	3	3	3	19s

### replicaset-definition.yml

```
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

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

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

```
> kubectl get replicaset
```

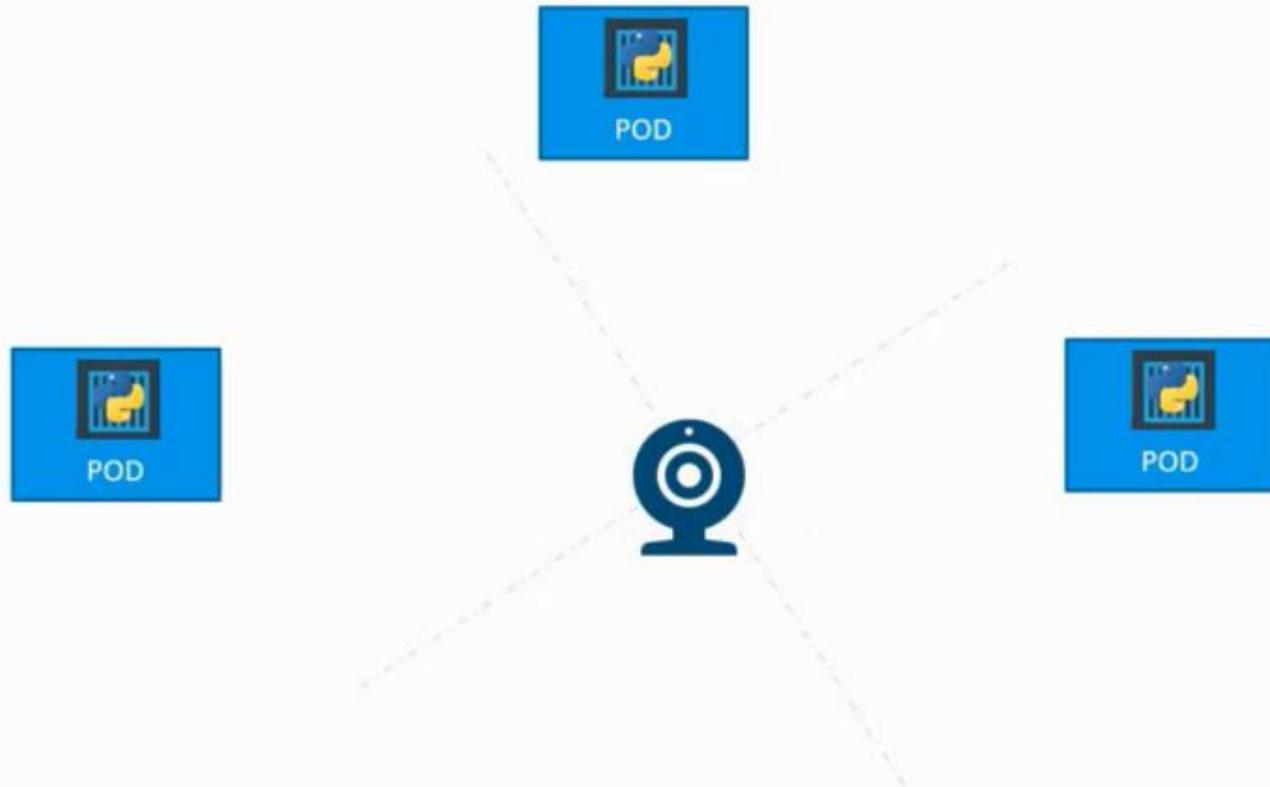
NAME	DESIRED	CURRENT	READY	AGE
myapp-replicaset	3	3	3	19s

```
> 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

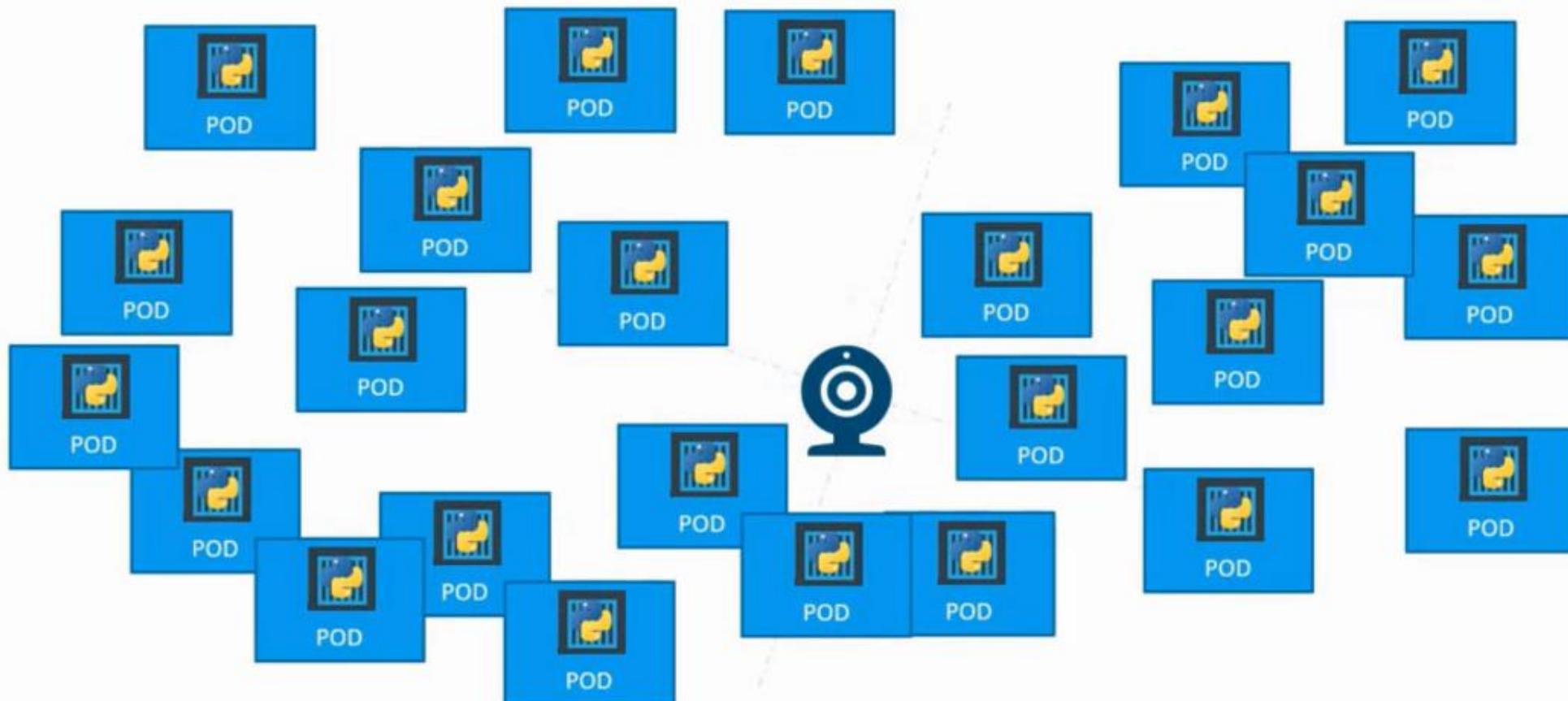
# Labels and Selectors

---

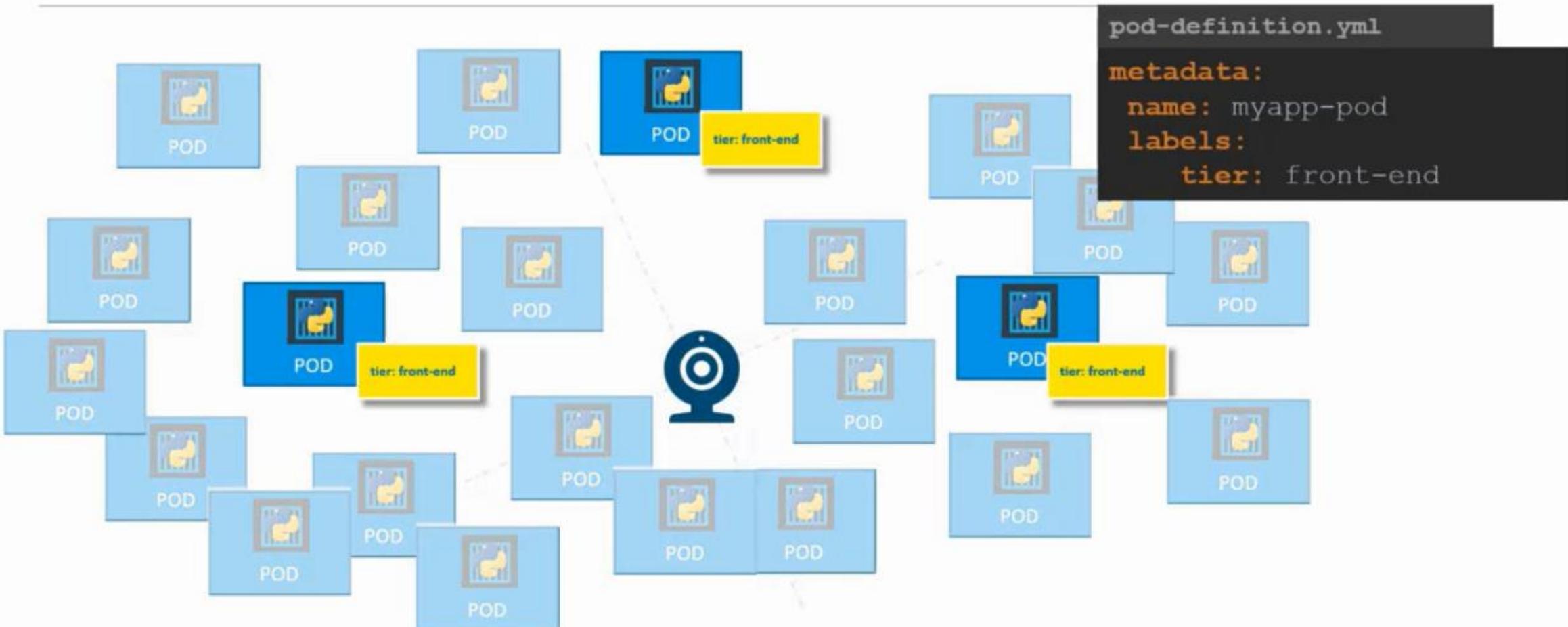


# Labels and Selectors

---



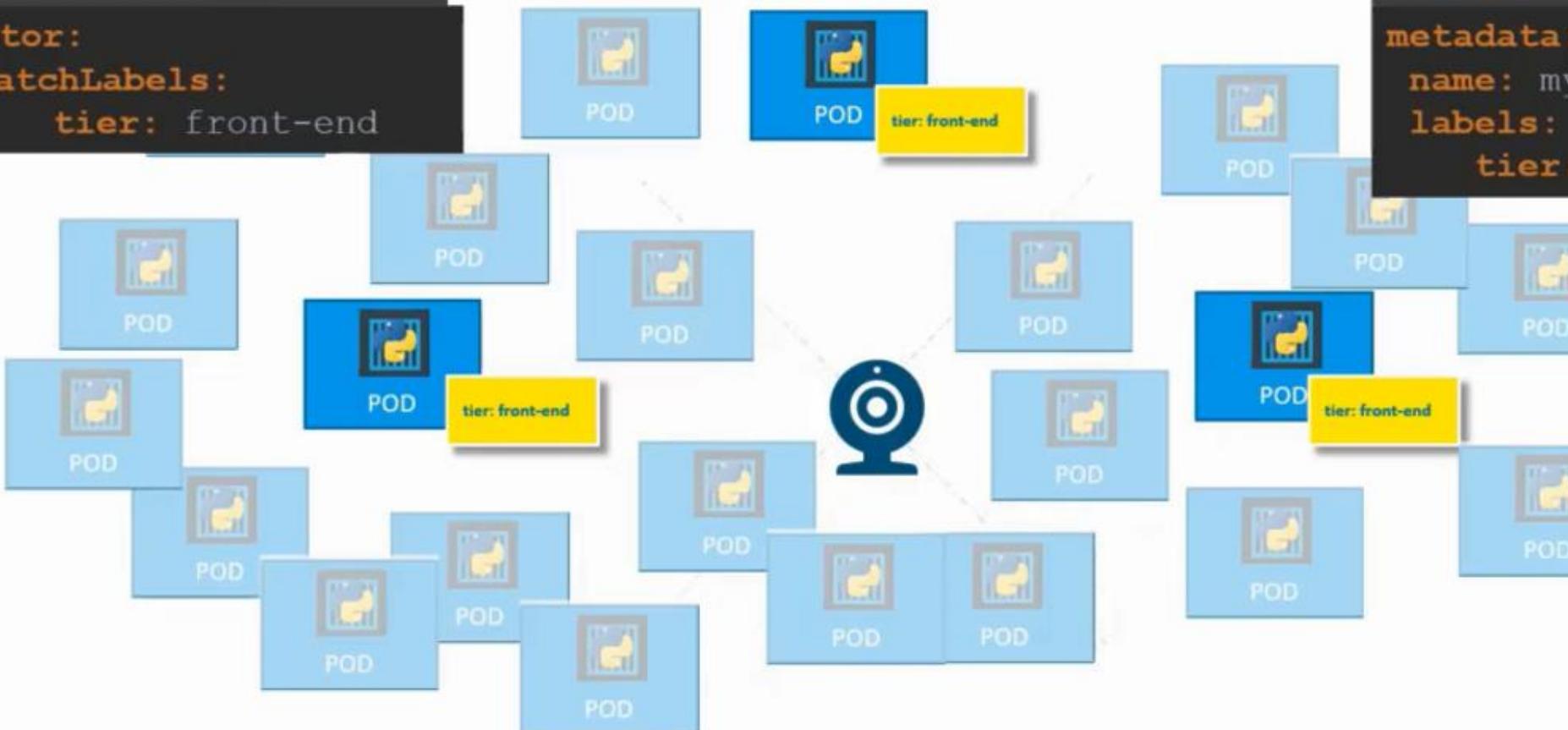
# Labels and Selectors



# Labels and Selectors

```
replicaset-definition.yml
```

```
selector:  
  matchLabels:  
    tier: front-end
```



```
replicaset-definition.yml
```

```
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
```



# Scale

```
replicaset-definition.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
```

# Scale

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

```
replicaset-definition.yml
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
```

# Scale

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

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

```
replicaset-definition.yml
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
```

# Scale

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

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

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



```
replicaset-definition.yml
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

---

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

```
> kubectl get replicaset
```

```
> kubectl delete replicaset myapp-replicaset
```

\*Also deletes all underlying PODs

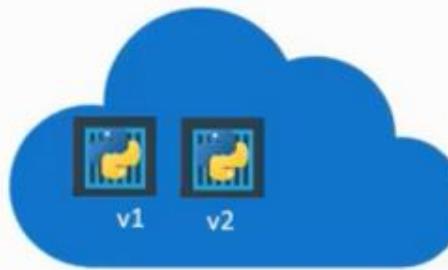
```
> kubectl replace -f replicaset-definition.yml
```

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

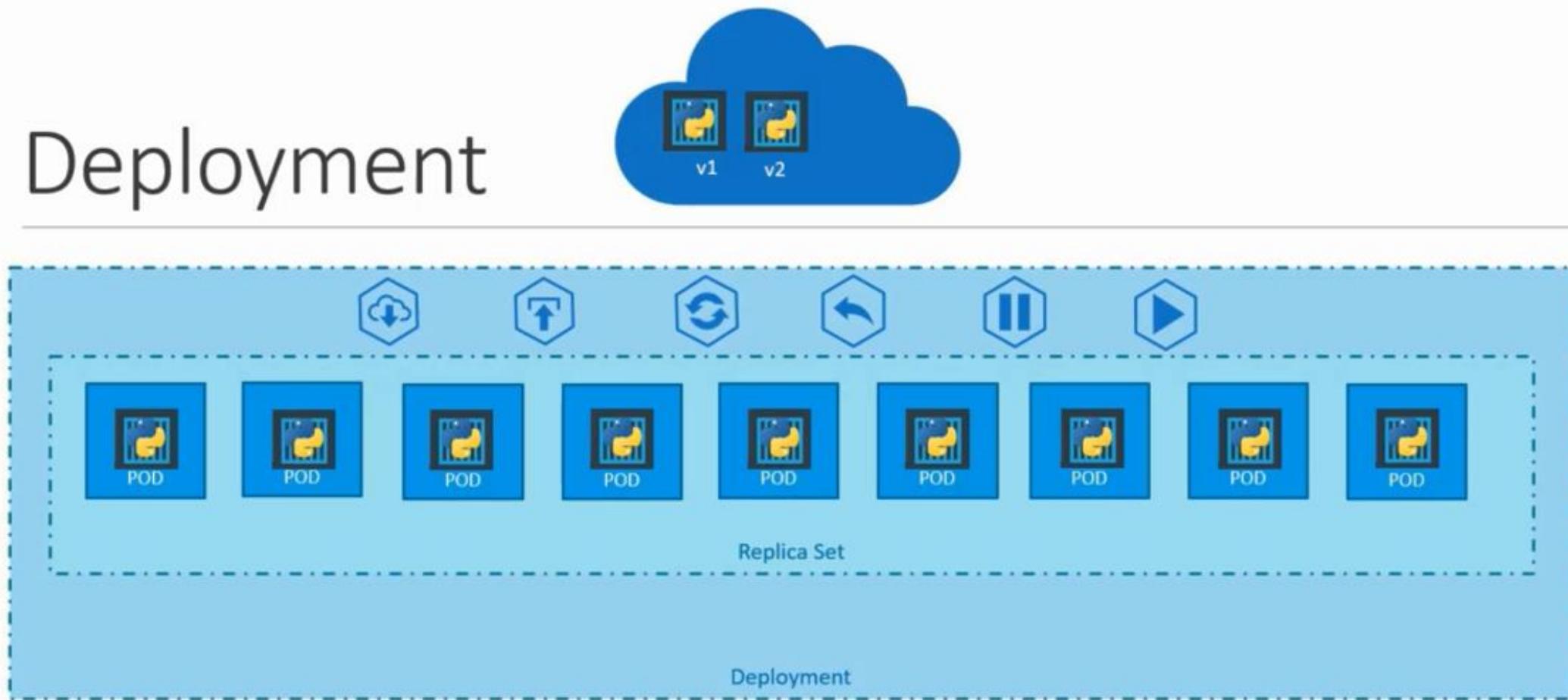
# Deployment

# Deployment

---



# Deployment



# Definition

---

```
deployment-definition.yml
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
```

# 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
```

```
deployment-definition.yml  
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
```

# 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
```

```
deployment-definition.yml  
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
```

# 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
```

```
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
```

# commands

---

```
> kubectl get all
```

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deploy/myapp-deployment	3	3	3	3	9h
NAME	DESIRED	CURRENT	READY	AGE	
rs/myapp-deployment-6795844b58	3	3	3	9h	
NAME	READY	STATUS	RESTARTS	AGE	
po/myapp-deployment-6795844b58-5rbjl	1/1	Running	0	9h	
po/myapp-deployment-6795844b58-h4w55	1/1	Running	0	9h	
po/myapp-deployment-6795844b58-1fjhv	1/1	Running	0	9h	

# NameSpaces

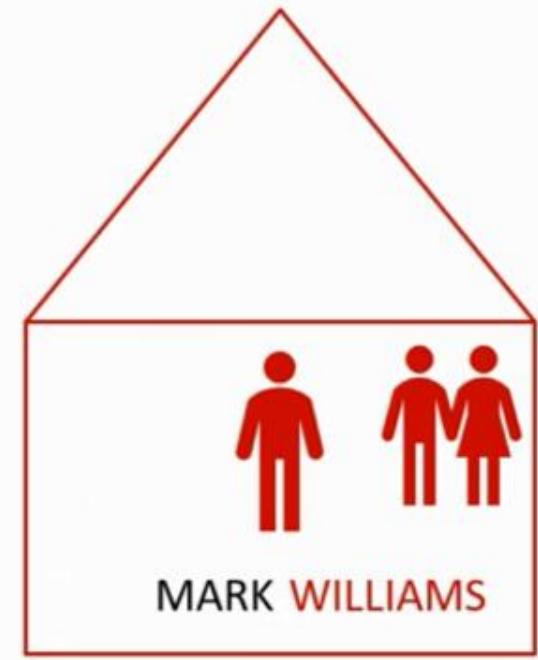
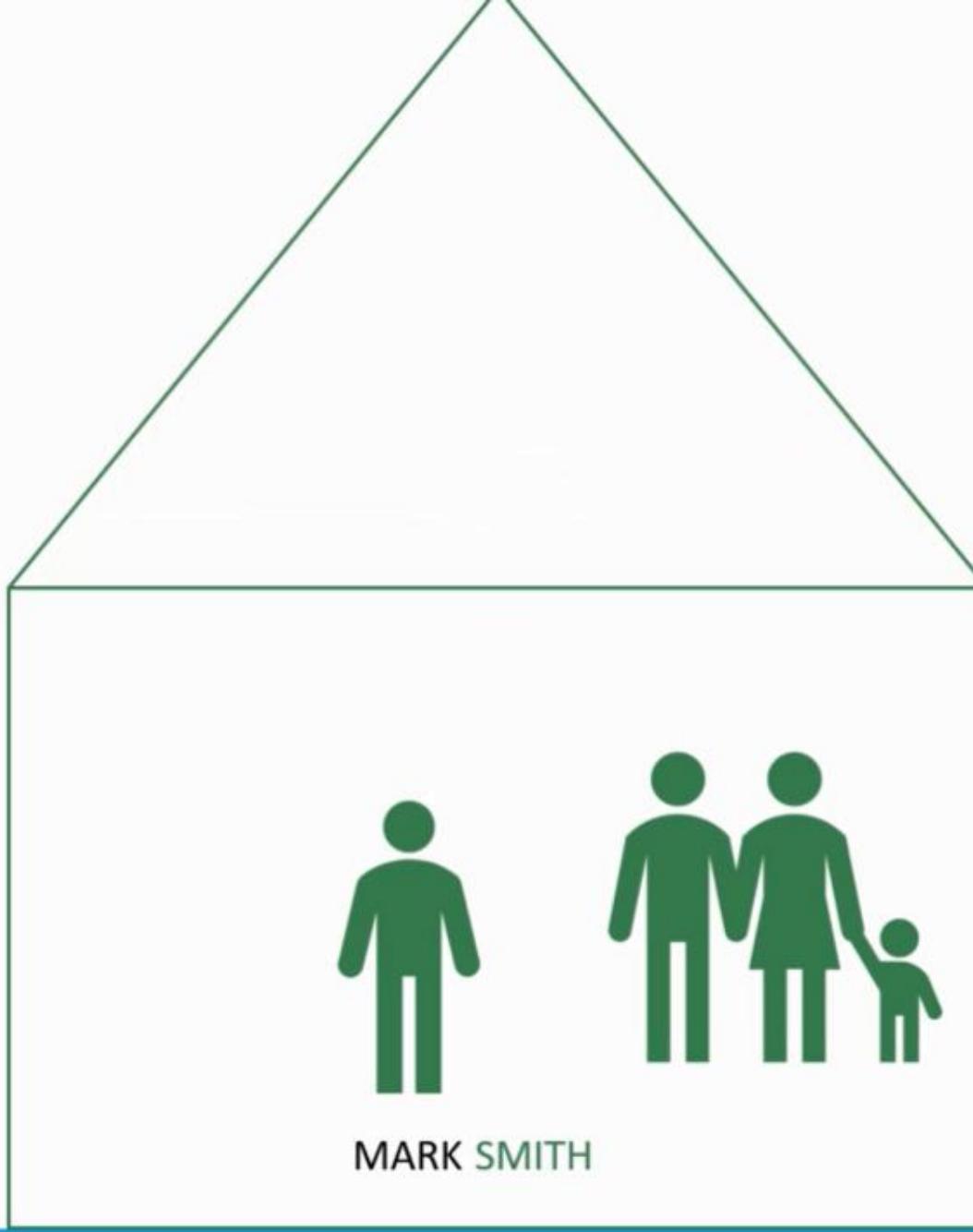


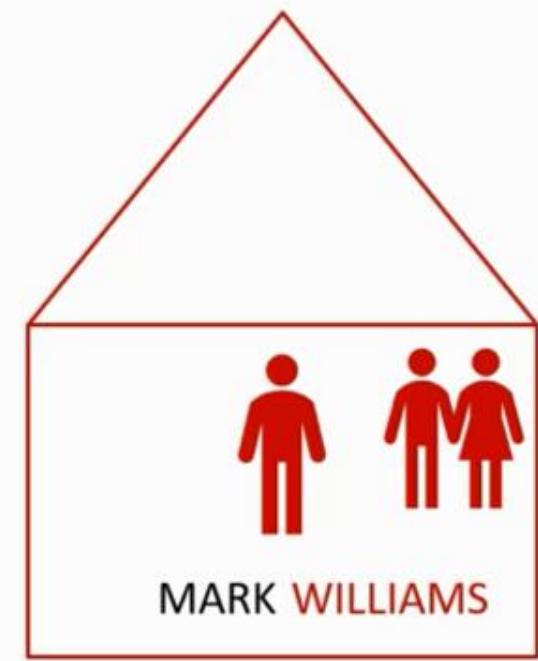
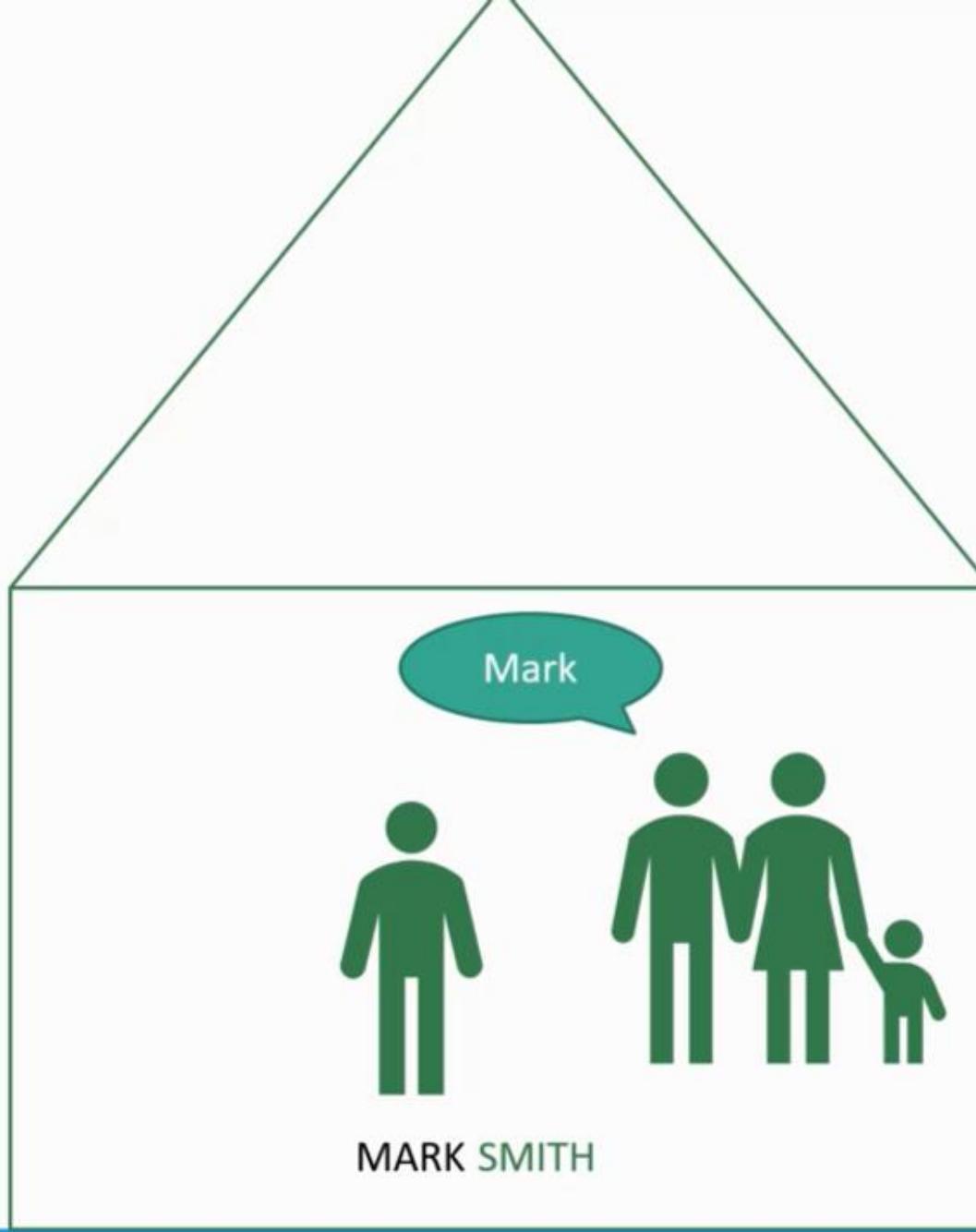
MARK SMITH

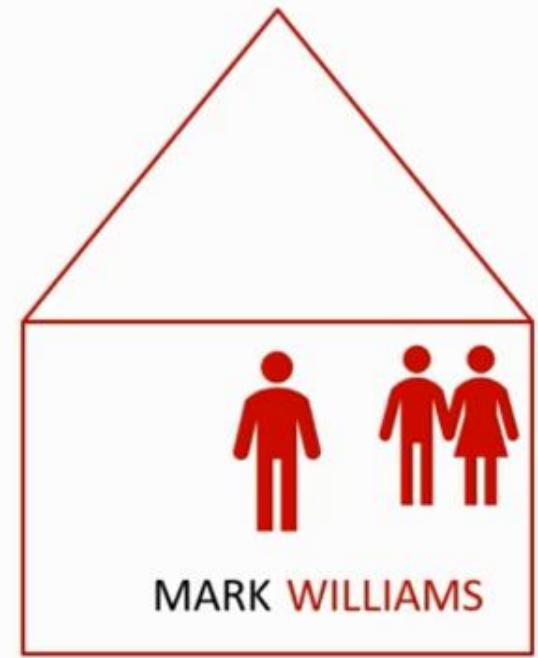
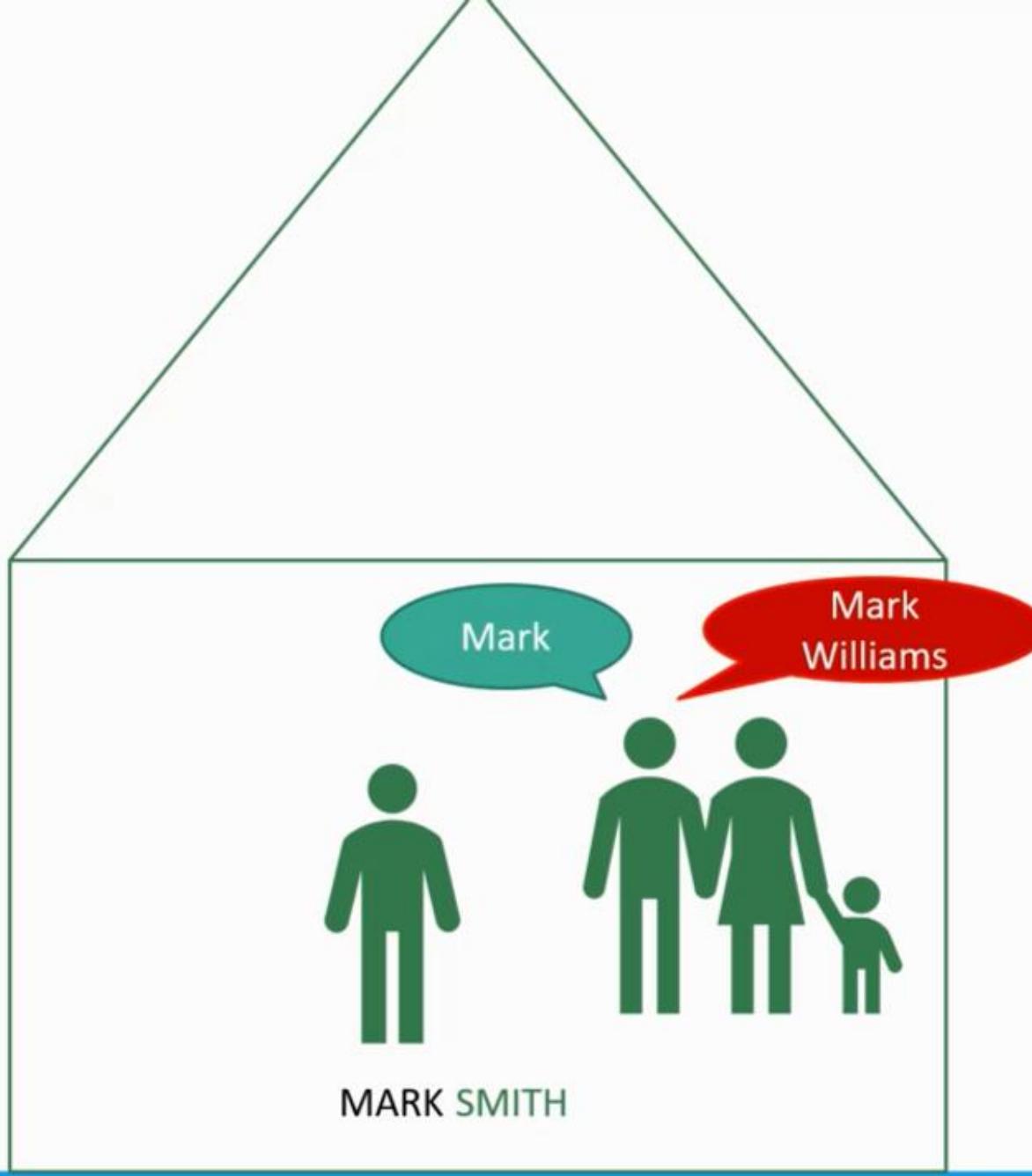


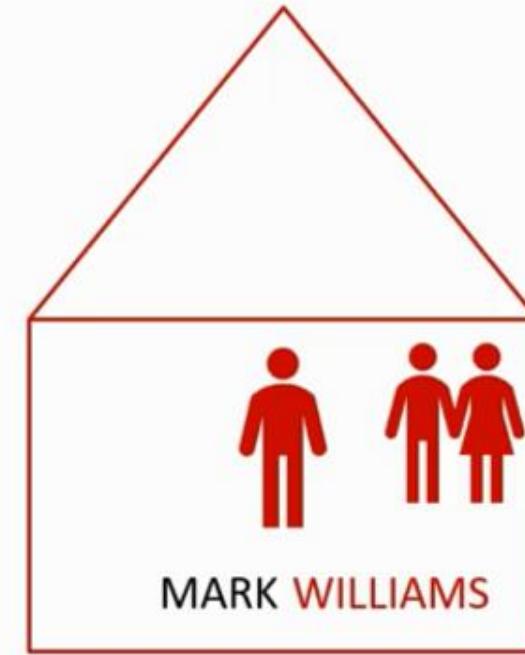
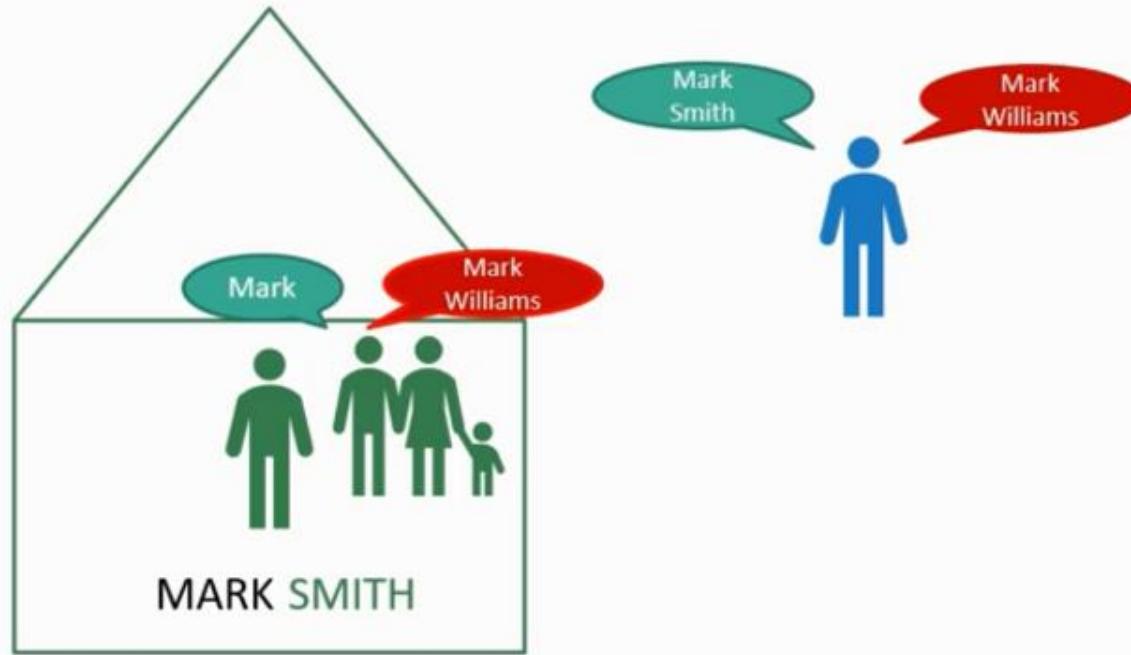
MARK WILLIAMS

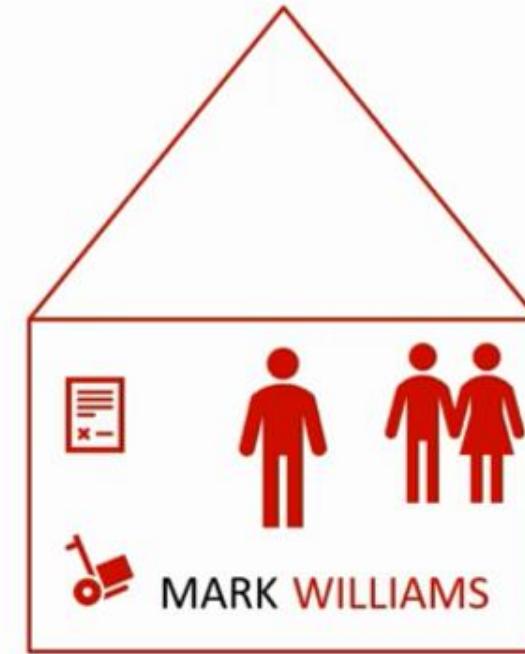
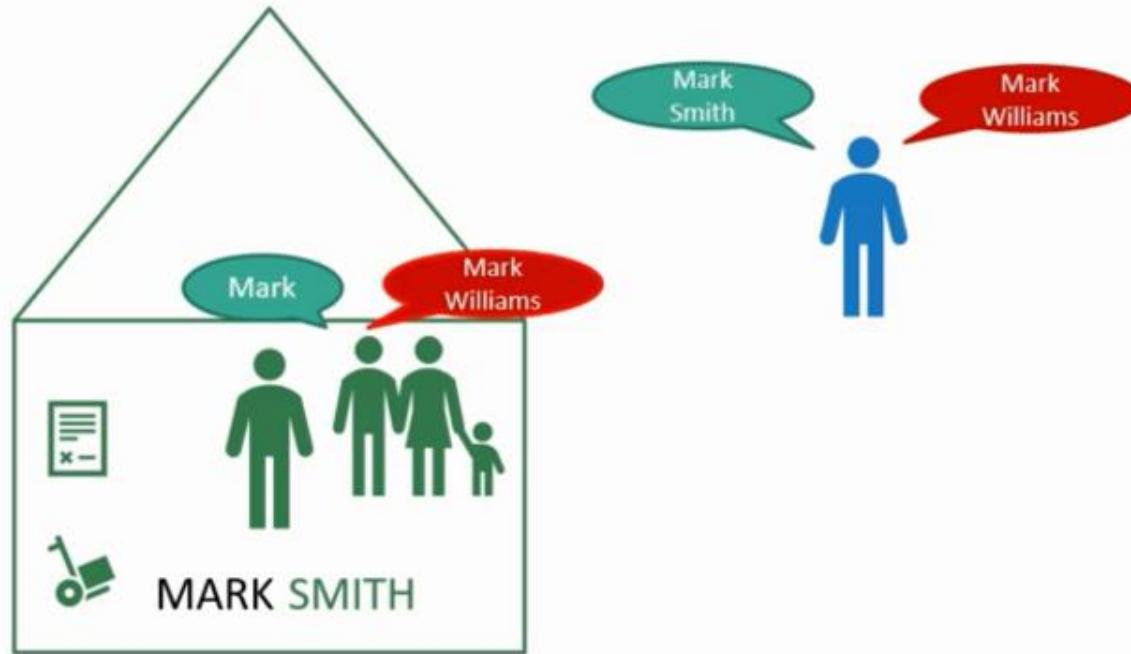


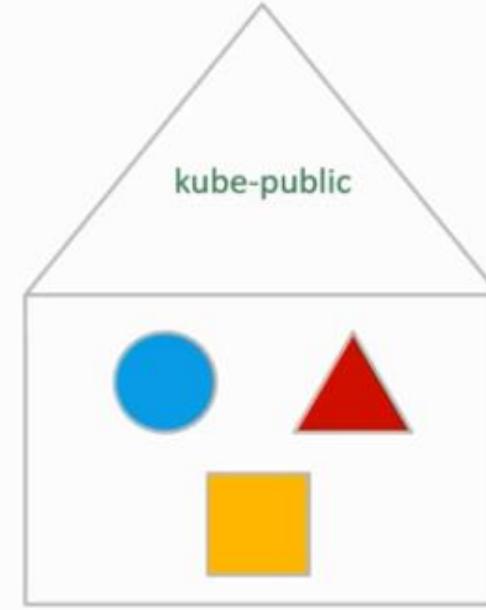
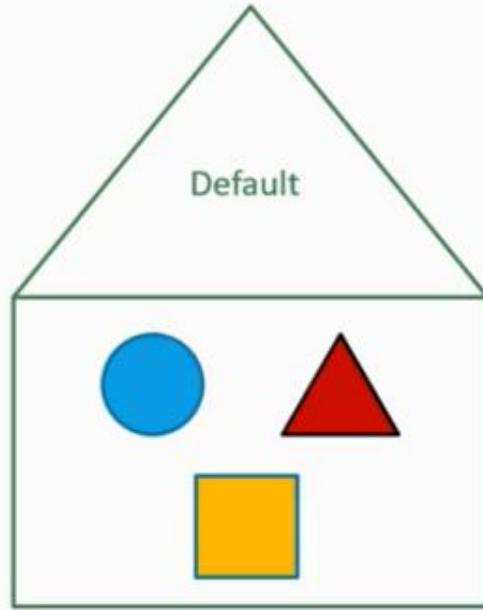
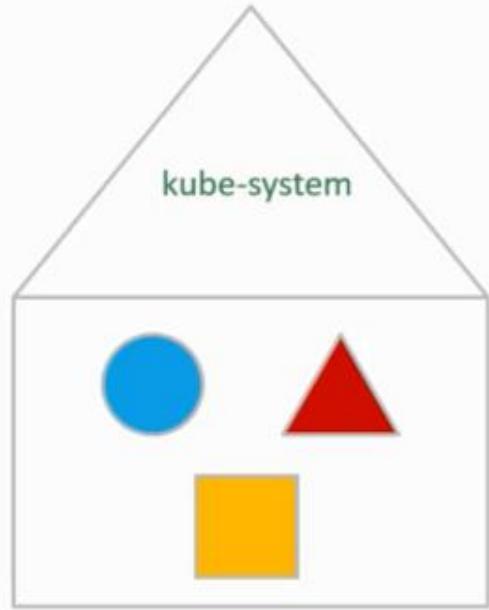




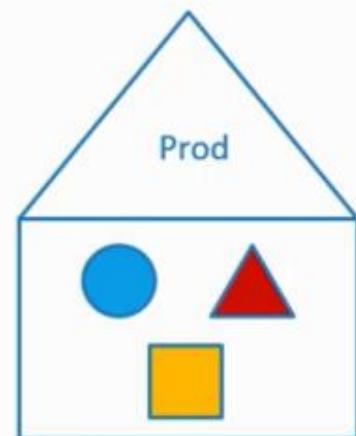
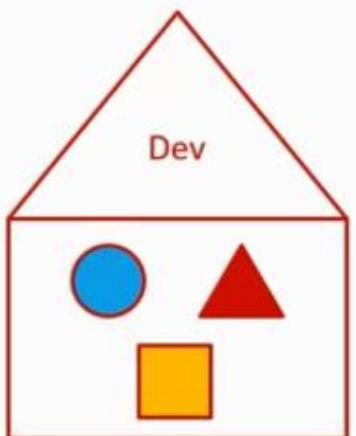
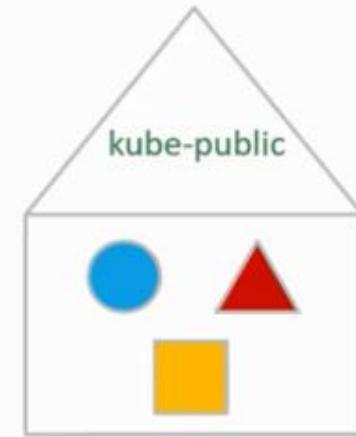
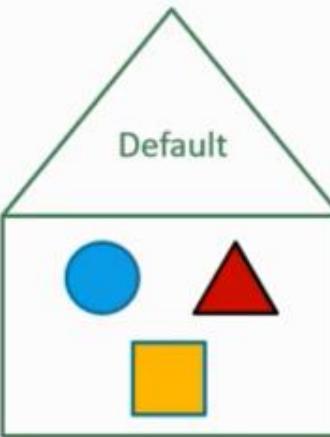
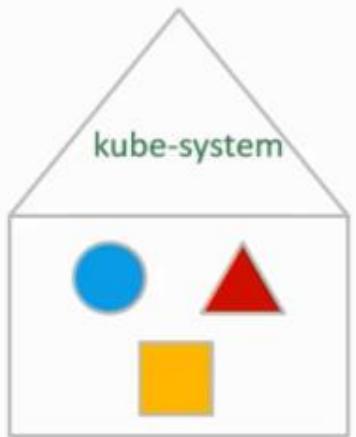




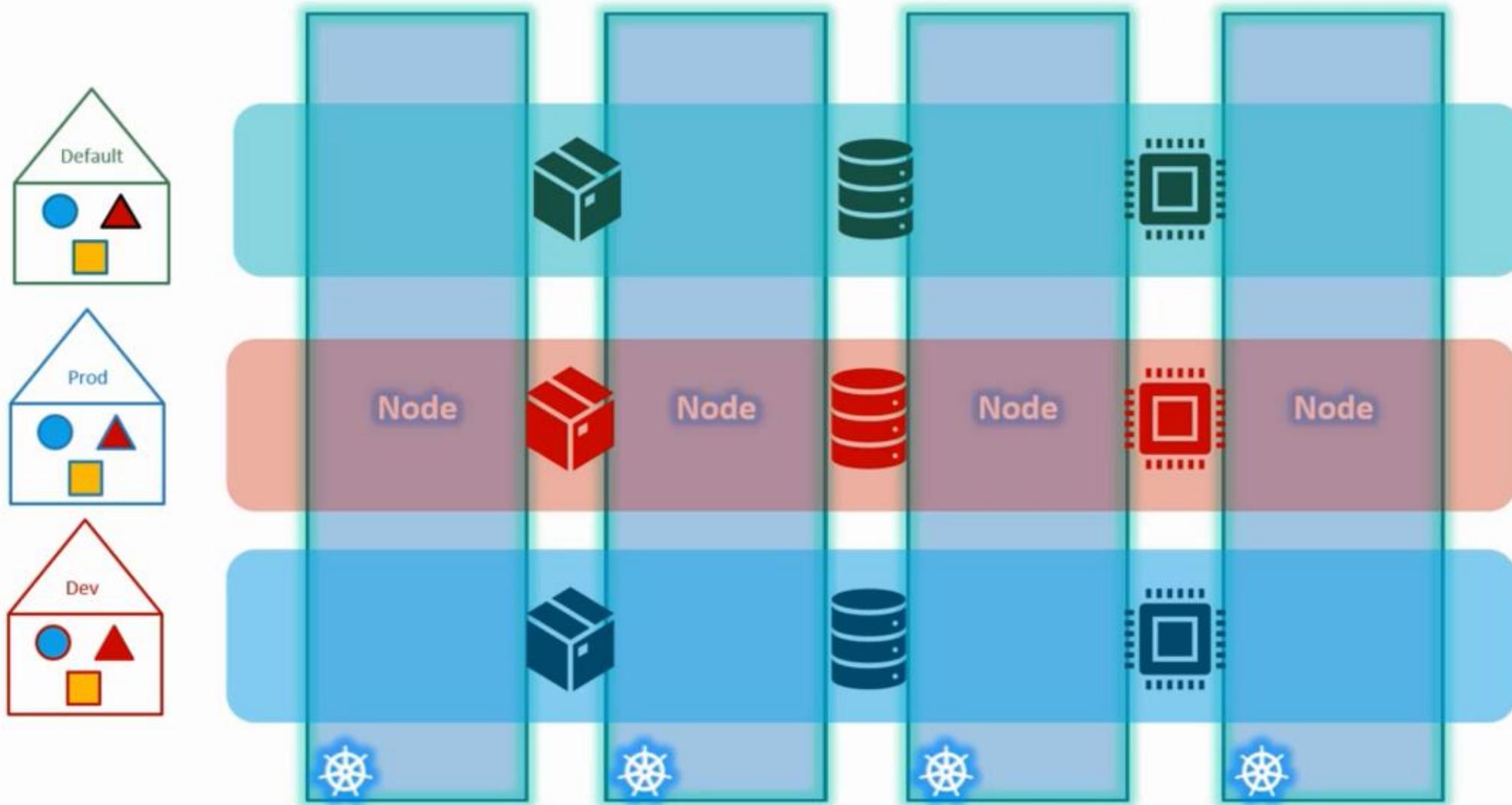




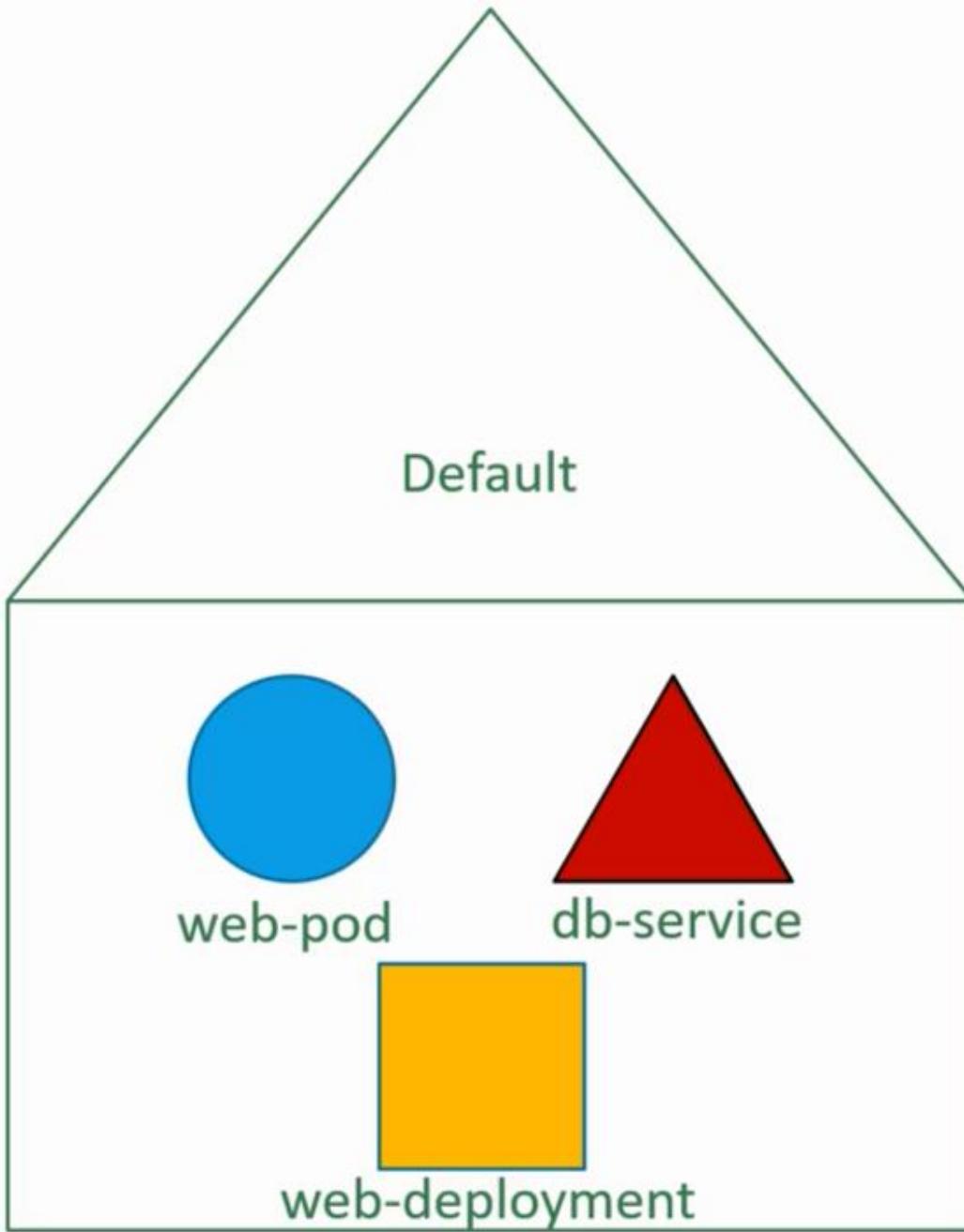
# Namespace - Isolation



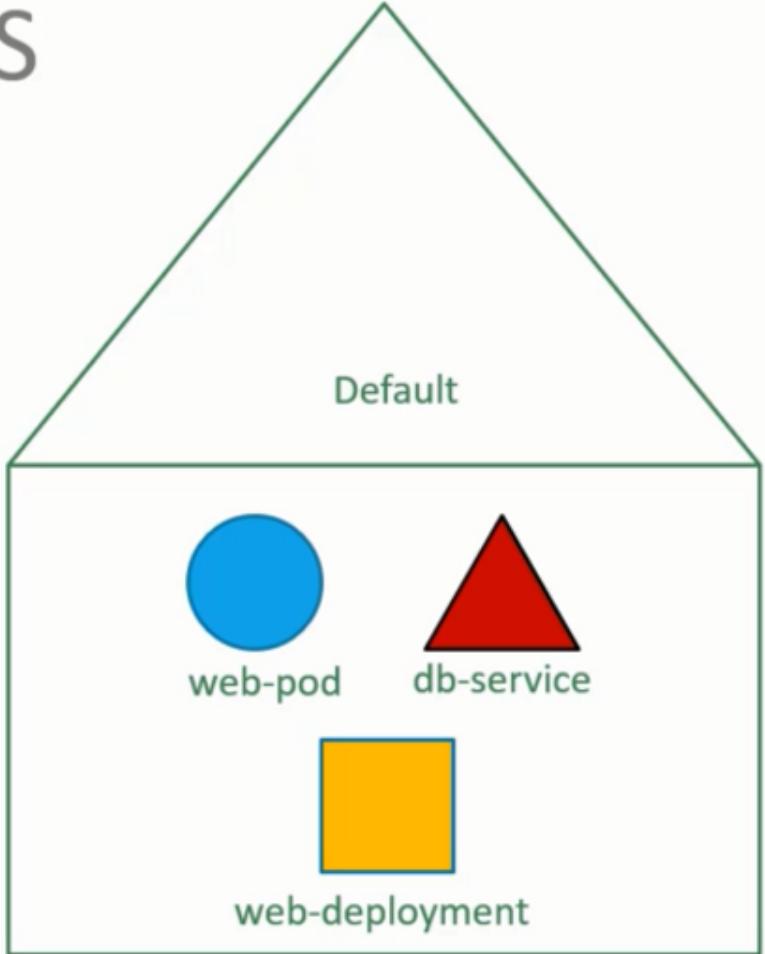
# Namespace – Resource Limits



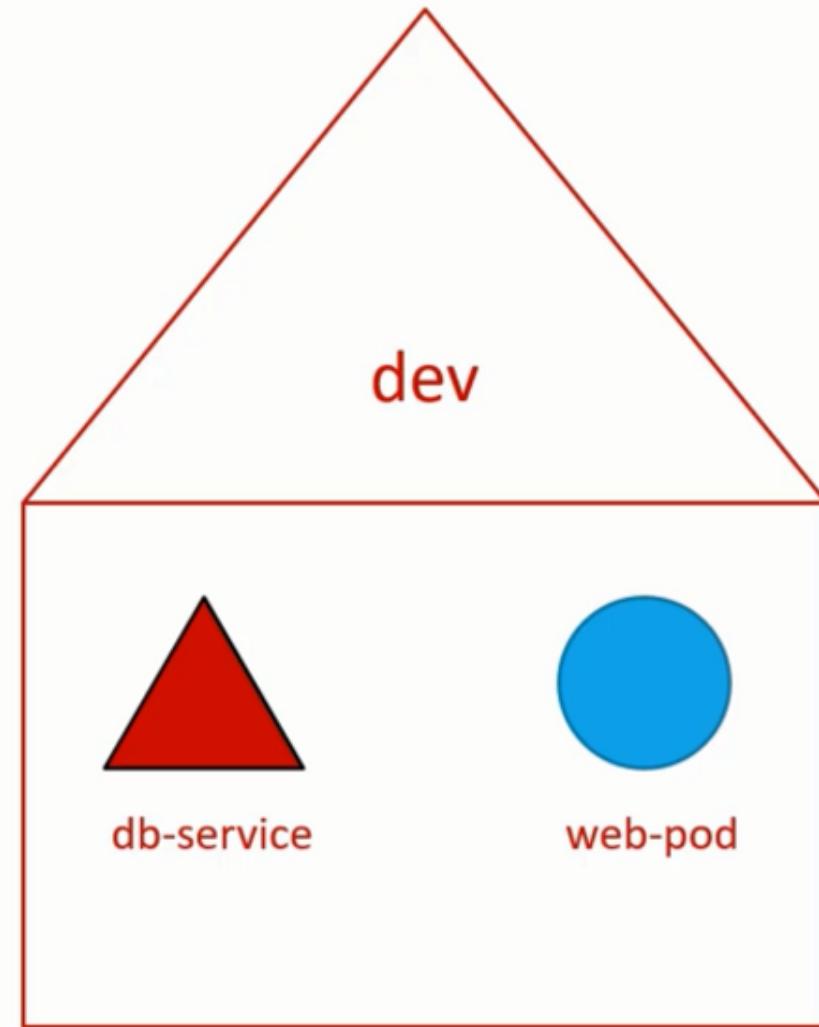
# DNS



# DNS



```
mysql.connect("db-service")
```



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

# DNS

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

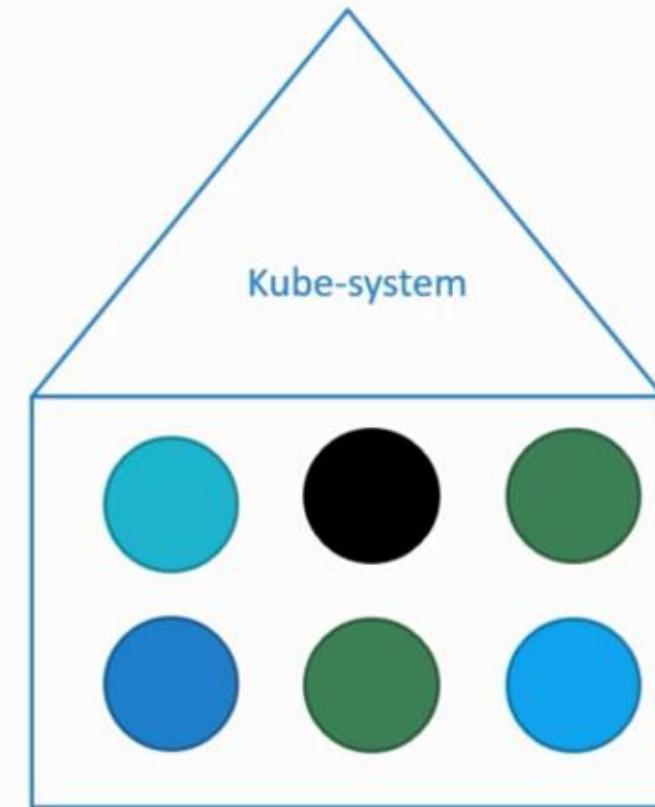
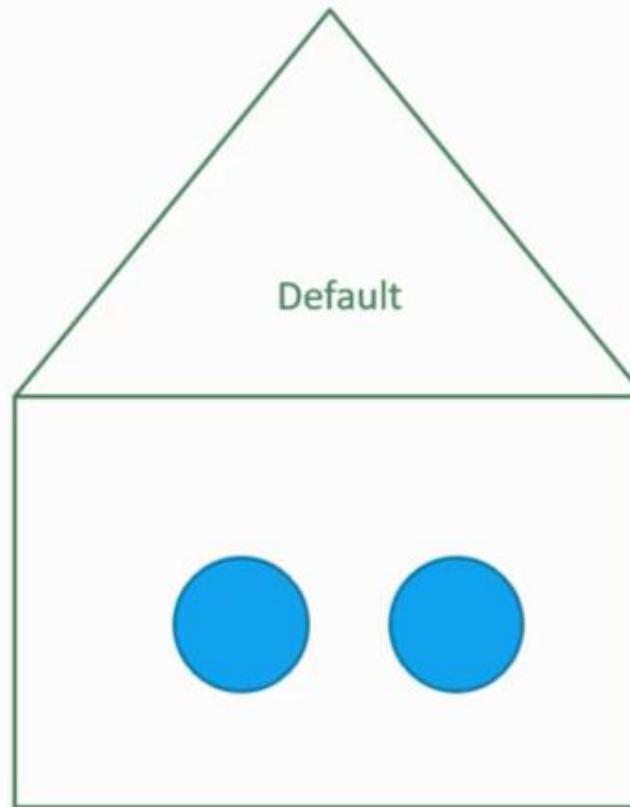


```
> kubectl get pods
```

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
```

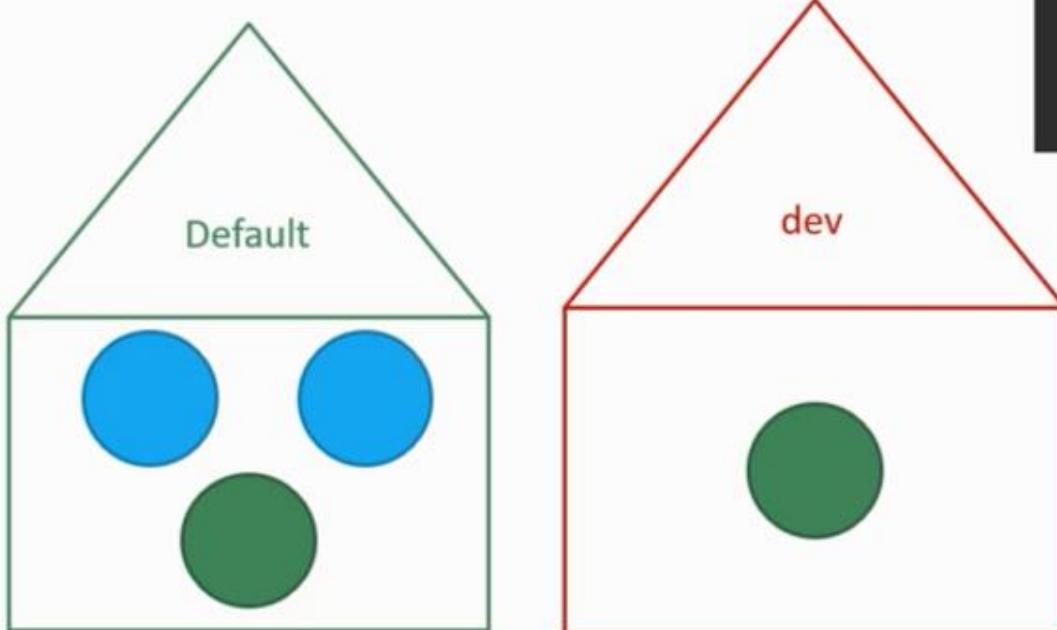
NAME	READY	STATUS	RESTAR
coredns-78fcdf6894-92d52	1/1	Running	7
coredns-78fcdf6894-jx25g	1/1	Running	7
etcd-master	1/1	Running	7
kube-apiserver-master	1/1	Running	7
kube-controller-manager-master	1/1	Running	7
kube-flannel-ds-amd64-hz4cf	1/1	Running	14
kube-proxy-4b8tn	1/1	Running	7
kube-proxy-98db4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



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

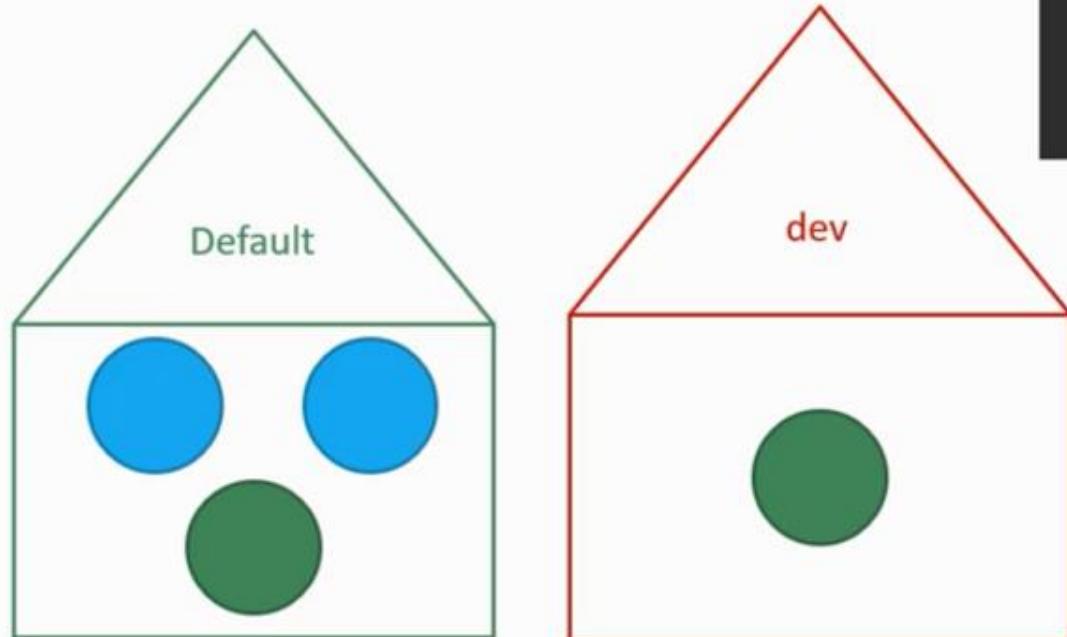
```
> kubectl create -f pod-definition.yml --namespace=dev  
pod/myapp-pod created
```

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



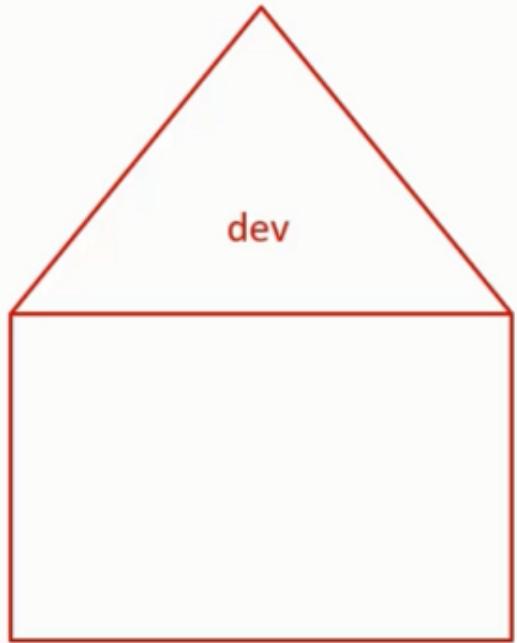
```
> kubectl create -f pod-definition.yml  
pod/myapp-pod created
```

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



```
pod-definition.yml  
apiVersion: v1  
kind: Pod  
metadata:  
  name: myapp-pod  
  namespace: dev  
  labels:  
    app: myapp  
    type: front-end  
spec:  
  containers:  
  - name: nginx-container  
    image: nginx
```

# Create Namespace



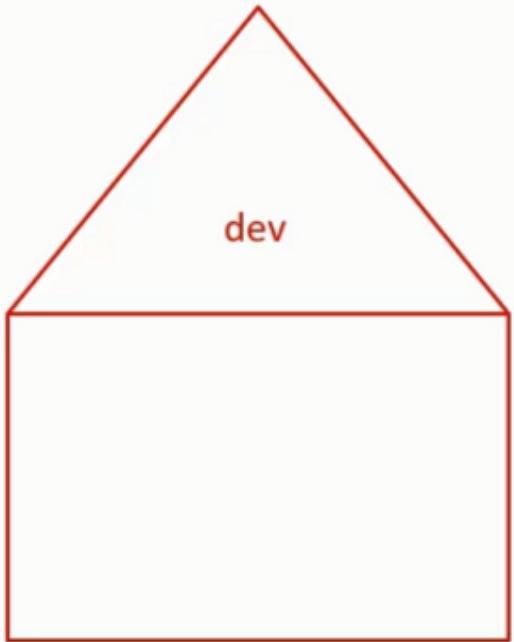
```
namespace-dev.yml
```

```
apiVersion: v1
kind: Namespace
metadata:
  name: dev
```

```
> kubectl create -f namespace-dev.yml
```

```
namespace/dev created
```

# Create Namespace



```
namespace-dev.yml
```

```
apiVersion: v1
kind: Namespace
metadata:
  name: dev
```

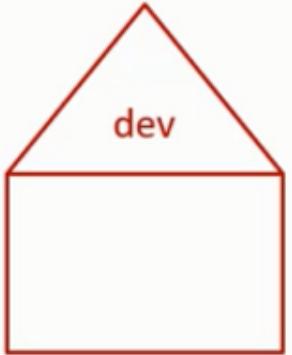
```
> kubectl create -f namespace-dev.yml
```

```
namespace/dev created
```

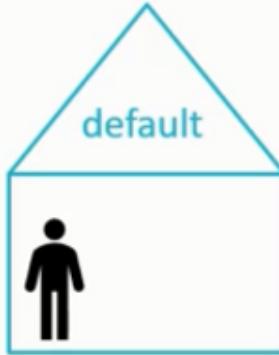
```
> kubectl create namespace dev
```

```
namespace/dev created
```

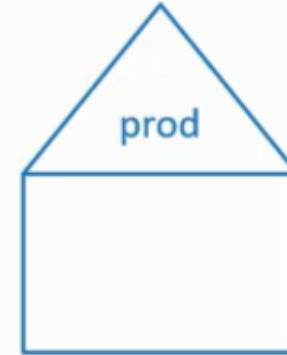
# Switch



```
> kubectl get pods --namespace=dev
```

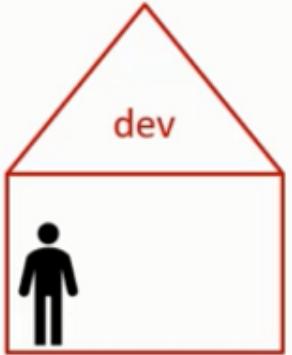


```
> kubectl get pods
```

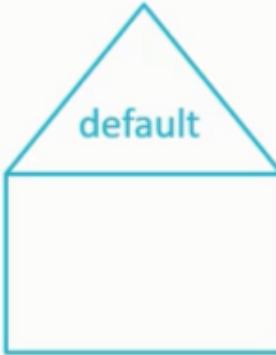


```
> kubectl get pods --namespace=prod
```

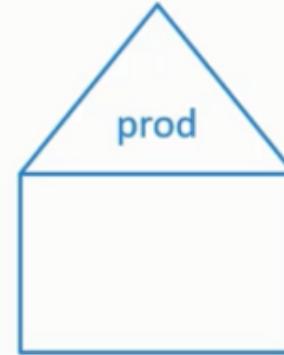
# Switch



```
> kubectl get pods --namespace=dev
```



```
> kubectl get pods
```

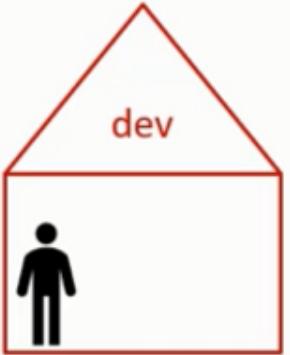


```
> kubectl get pods --namespace=prod
```

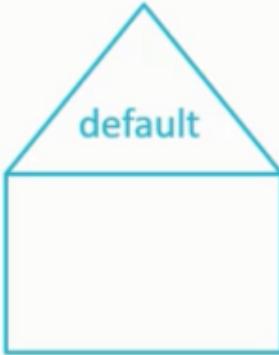
```
> kubectl config set-context $(kubectl config current-context) --namespace=dev
```

```
> kubectl get pod
```

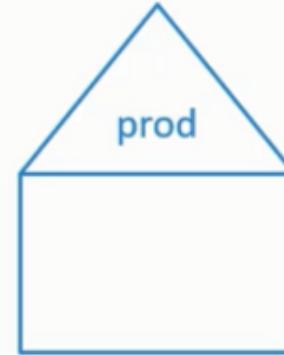
# Switch



```
> kubectl get pods --namespace=dev
```



```
> kubectl get pods
```



```
> kubectl get pods --namespace=prod
```

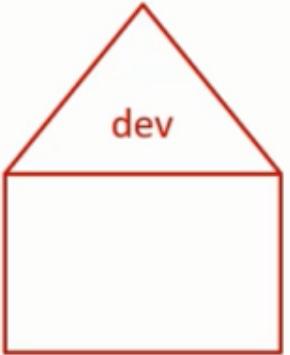
```
> kubectl config set-context $(kubectl config current-context) --namespace=dev
```

```
> kubectl get pods
```

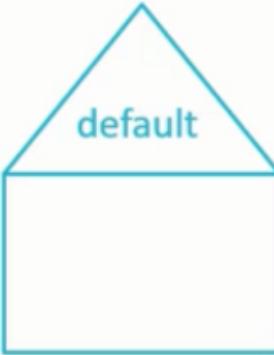
```
> kubectl get pods --namespace=default
```

```
> kubectl get pods --namespace=prod
```

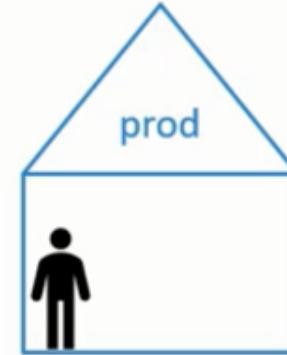
# Switch



```
> kubectl get pods --namespace=dev
```



```
> kubectl get pods
```



```
> kubectl get pods --namespace=prod
```

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

```
> kubectl get pods
```

```
> kubectl get pods --namespace=default
```

```
> kubectl get pods --namespace=prod
```

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

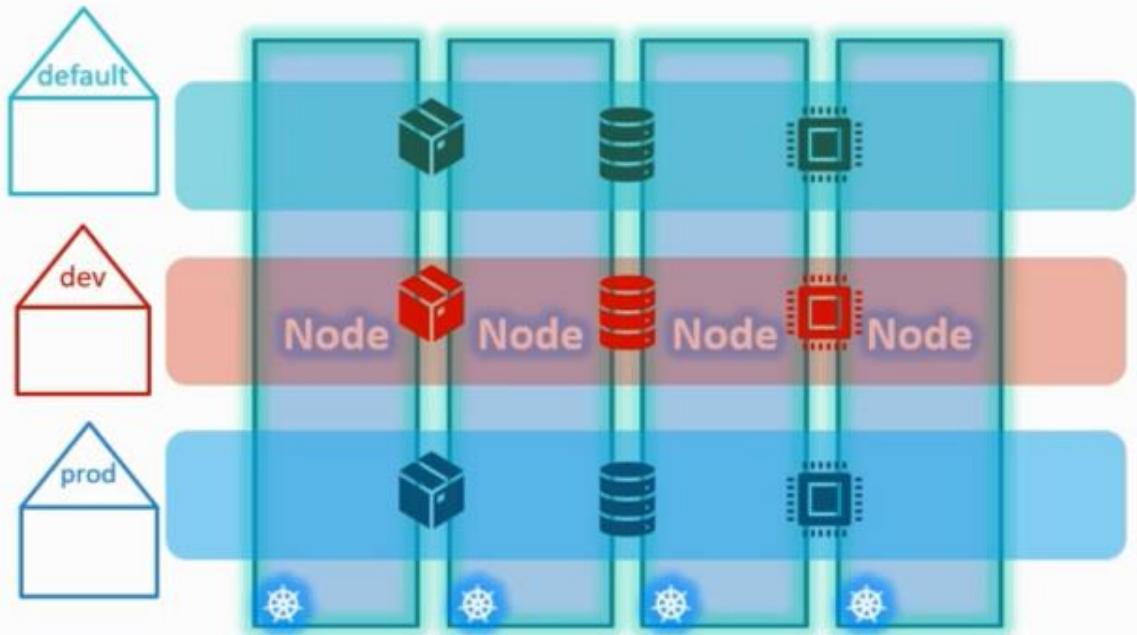
```
> kubectl get pods --namespace=dev
```

```
> kubectl get pods --namespace=default
```

```
> kubectl get pods
```

```
> kubectl get pods --all-namespaces
```

# Resource Quota



Compute-quota.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
```

# Services

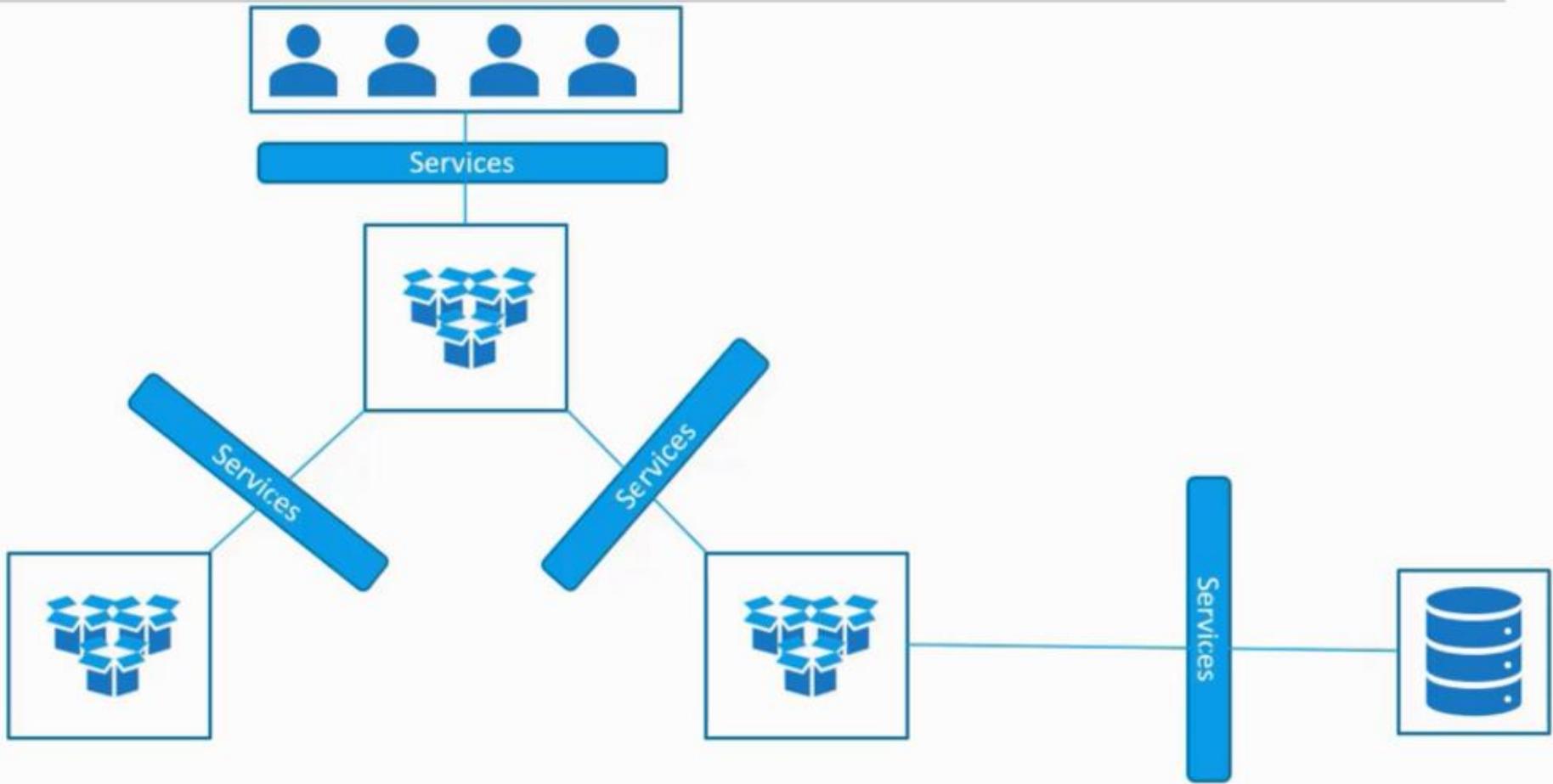
# Services

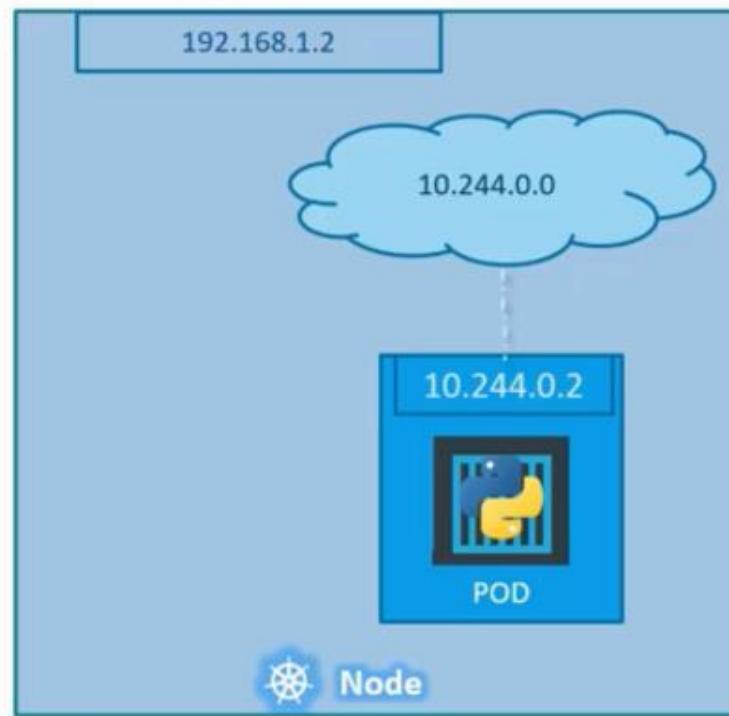
---

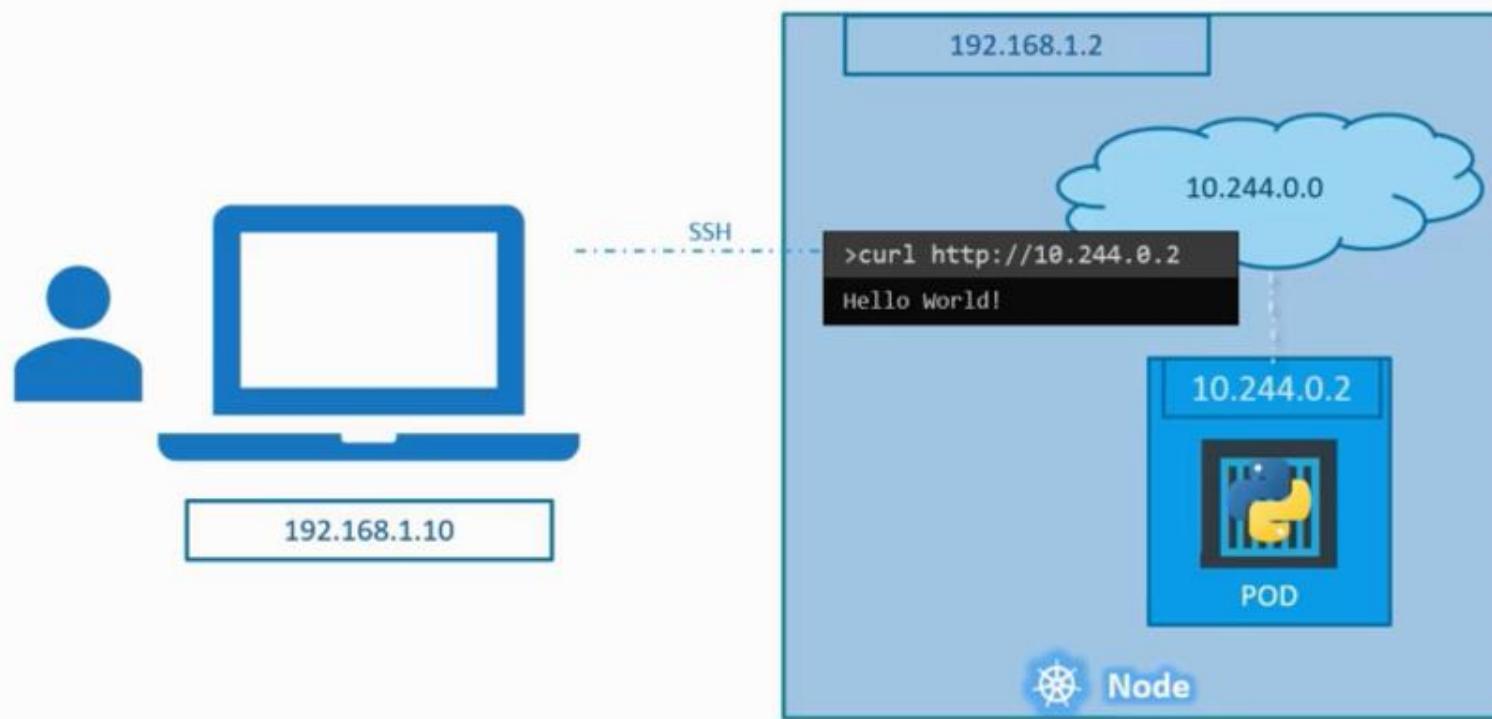


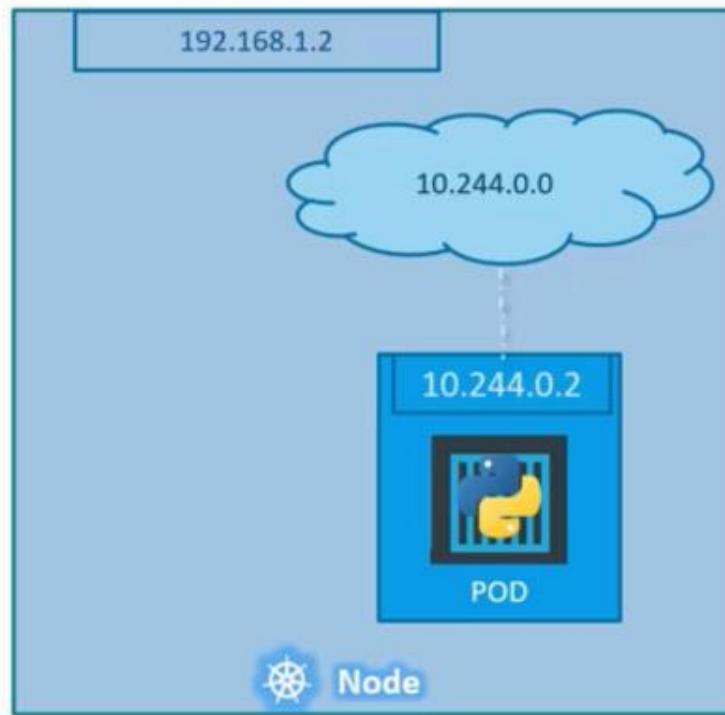
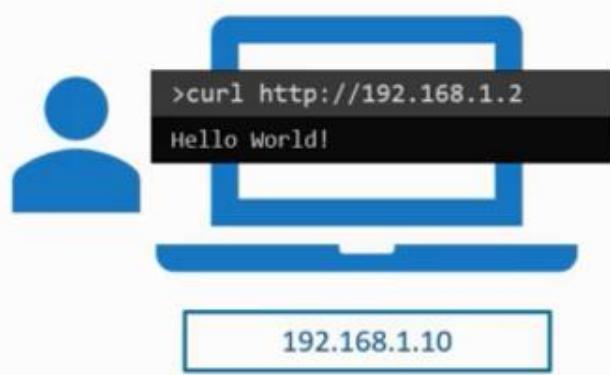
# Services

---



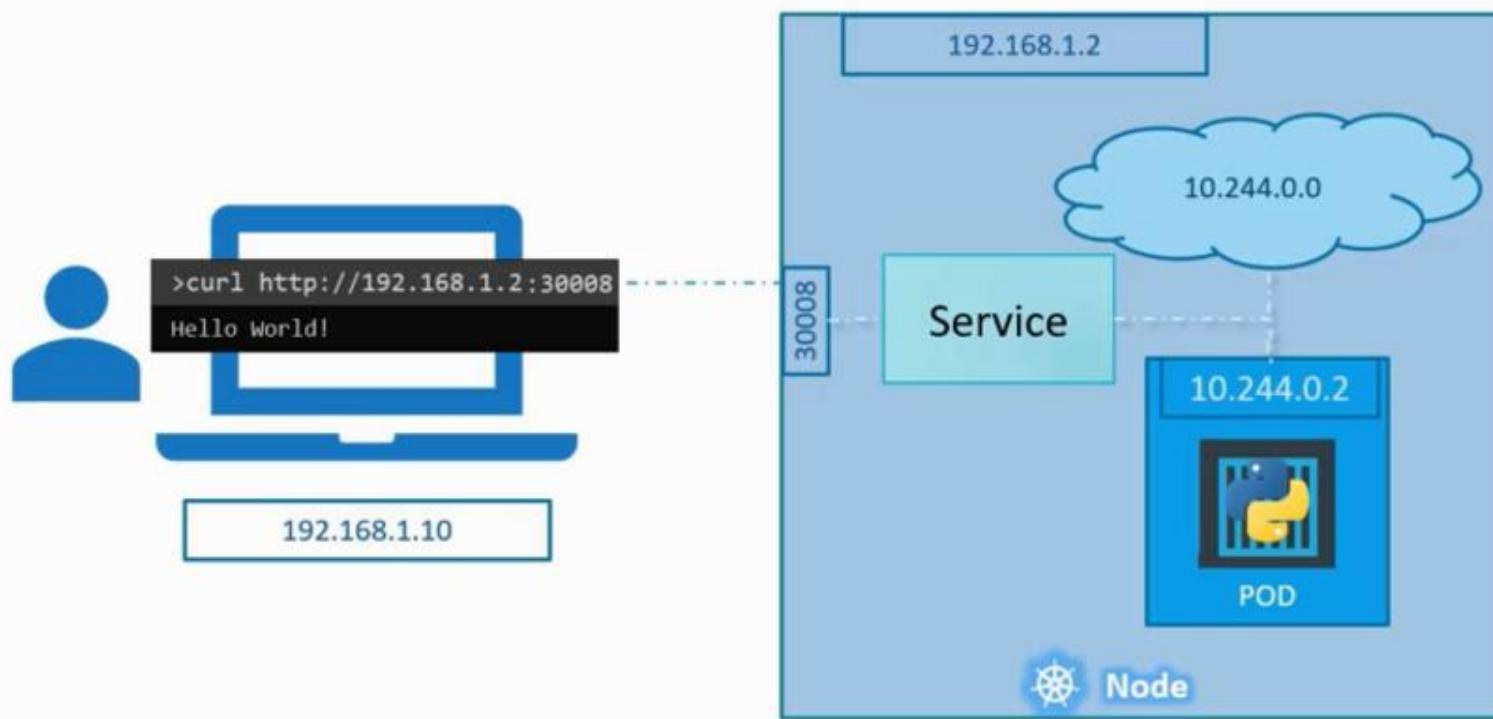






# Service

---



# Services Types

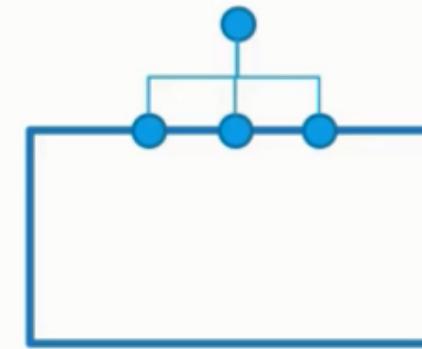
---



NodePort



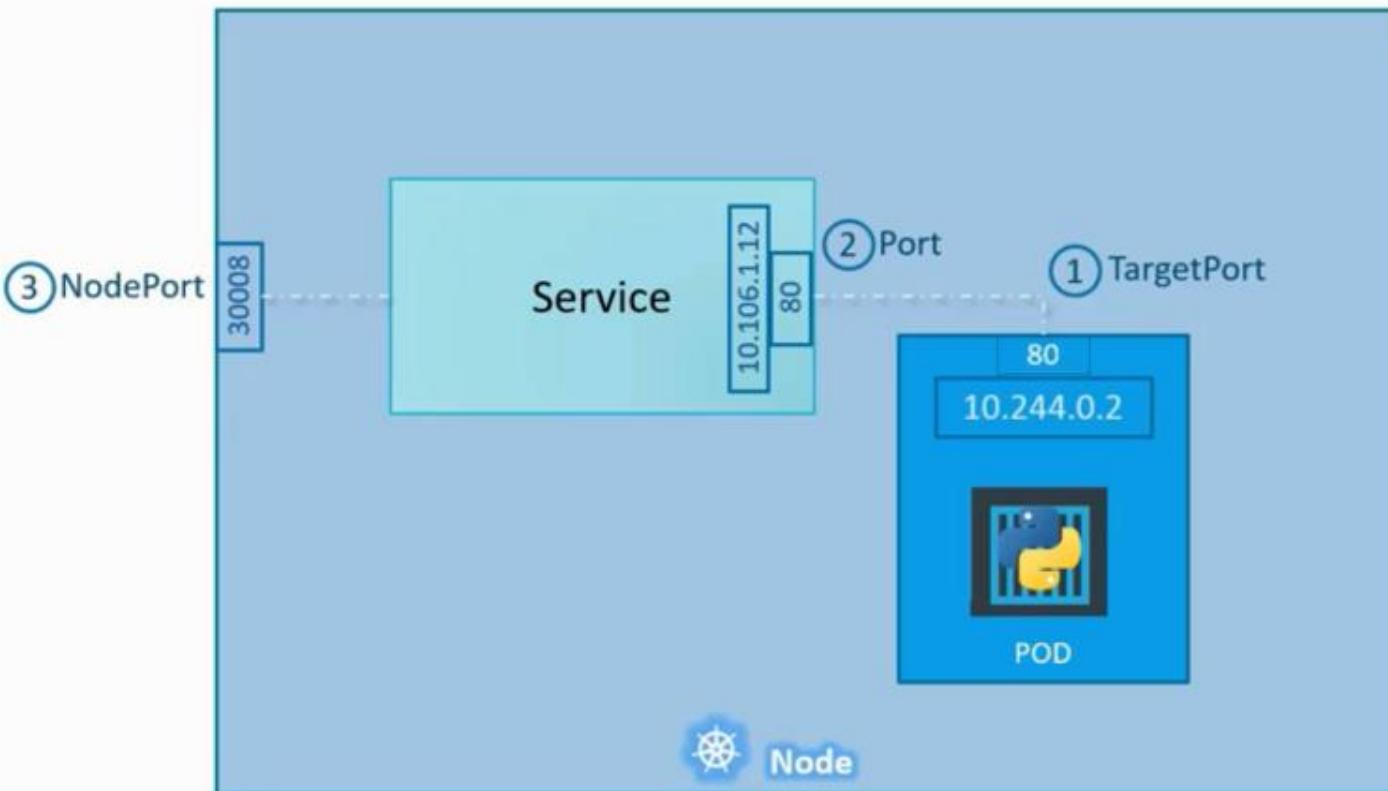
ClusterIP



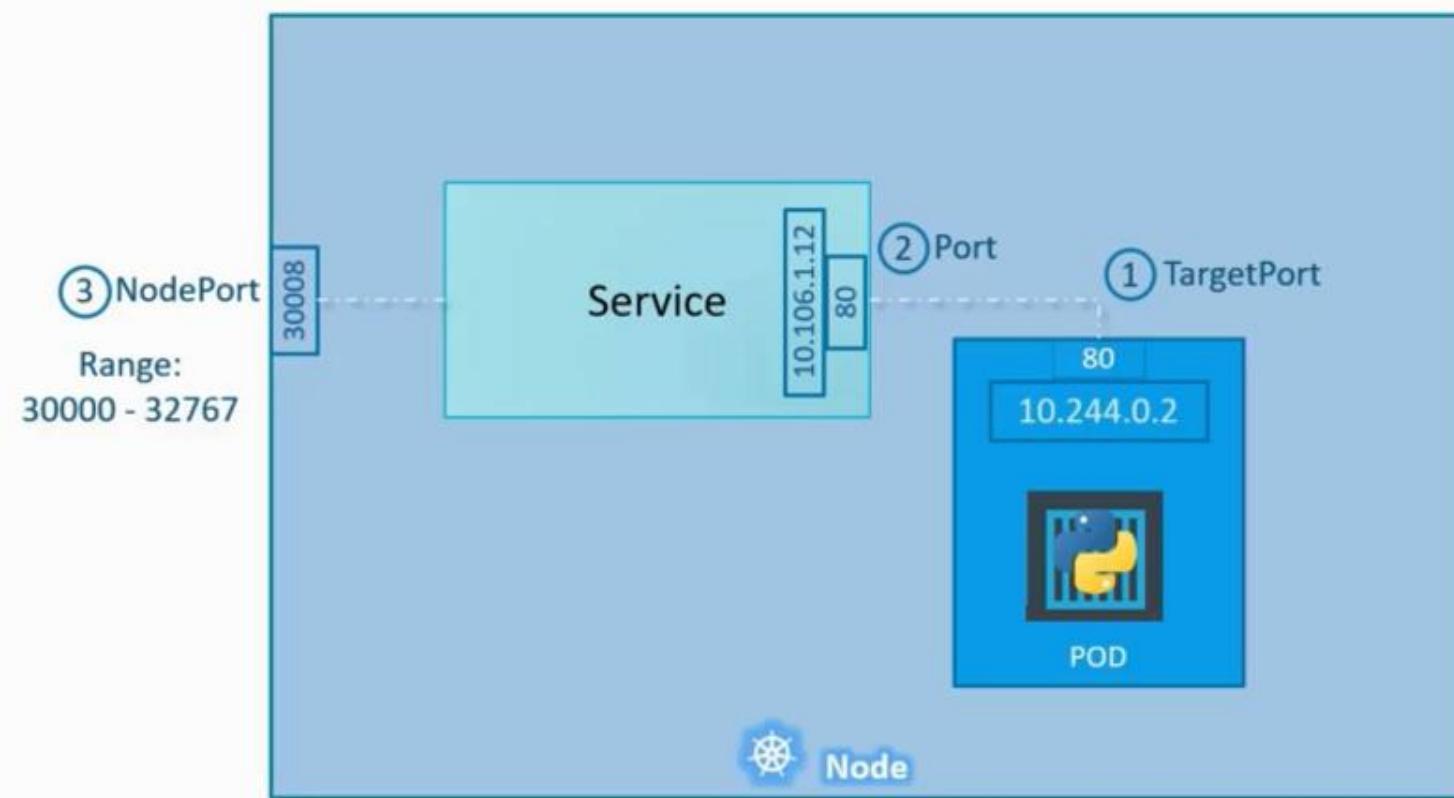
LoadBalancer

# Service - NodePort

---



# Service - NodePort



```
service-definition.yml
```

```
apiVersion: v1
kind: Service
metadata:
spec:
```

# Service - NodePort

---

service-definition.yml

```
apiVersion: v1
kind: Service
metadata:
  name: myapp-service
spec:
  type: NodePort
  ports:
    - targetPort: 80
      port: 80
      nodePort: 30008
  selector:
```

pod-definition.yml

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

# Service - NodePort

---

service-definition.yml

```
apiVersion: v1
kind: Service
metadata:
  name: myapp-service
spec:
  type: NodePort
  ports:
    - targetPort: 80
      port: 80
      nodePort: 30008
  selector:
    app: myapp
    type: front-end
```

pod-definition.yml

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
spec:
  containers:
    - name: nginx-container
      image: nginx
```

# Service - NodePort

---

```
service-definition.yml
```

```
apiVersion: v1
kind: Service
metadata:
  name: myapp-service

spec:
  type: NodePort
  ports:
    - targetPort: 80
      port: 80
      nodePort: 30008
  selector:
    app: myapp
    type: front-end
```

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

# Service - NodePort

---

```
service-definition.yml
```

```
apiVersion: v1
kind: Service
metadata:
  name: myapp-service
spec:
  type: NodePort
  ports:
    - targetPort: 80
      port: 80
      nodePort: 30008
  selector:
    app: myapp
    type: front-end
```

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

```
service "myapp-service" created
```

```
> kubectl get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	16d
myapp-service	NodePort	10.106.127.123	<none>	80:30008/TCP	5m

# Service - NodePort

```
service-definition.yml
```

```
apiVersion: v1
kind: Service
metadata:
  name: myapp-service
spec:
  type: NodePort
  ports:
    - targetPort: 80
      port: 80
      nodePort: 30008
  selector:
    app: myapp
    type: front-end
```

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

```
> kubectl get services
```

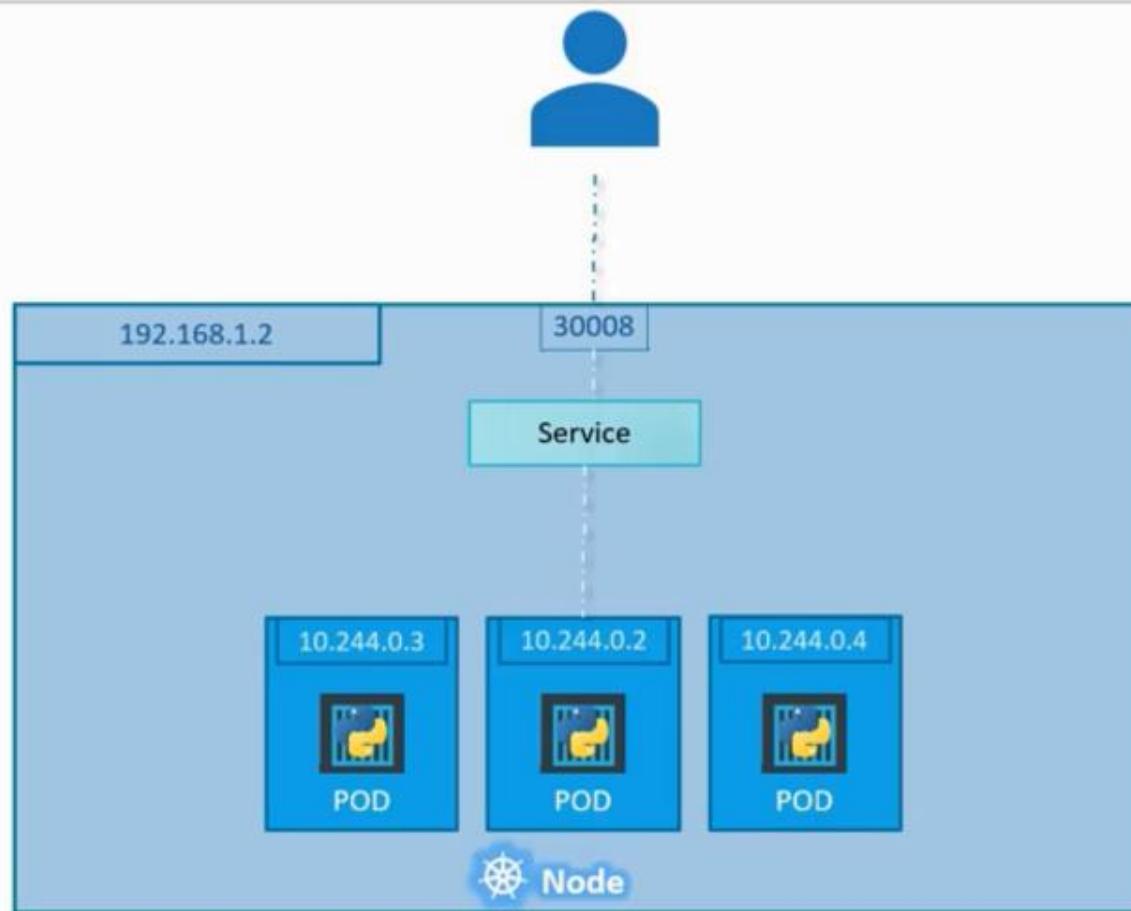
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	16d
myapp-service	NodePort	10.106.127.123	<none>	80:30008/TCP	5m

```
> curl http://192.168.1.2:30008
```

```
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
```

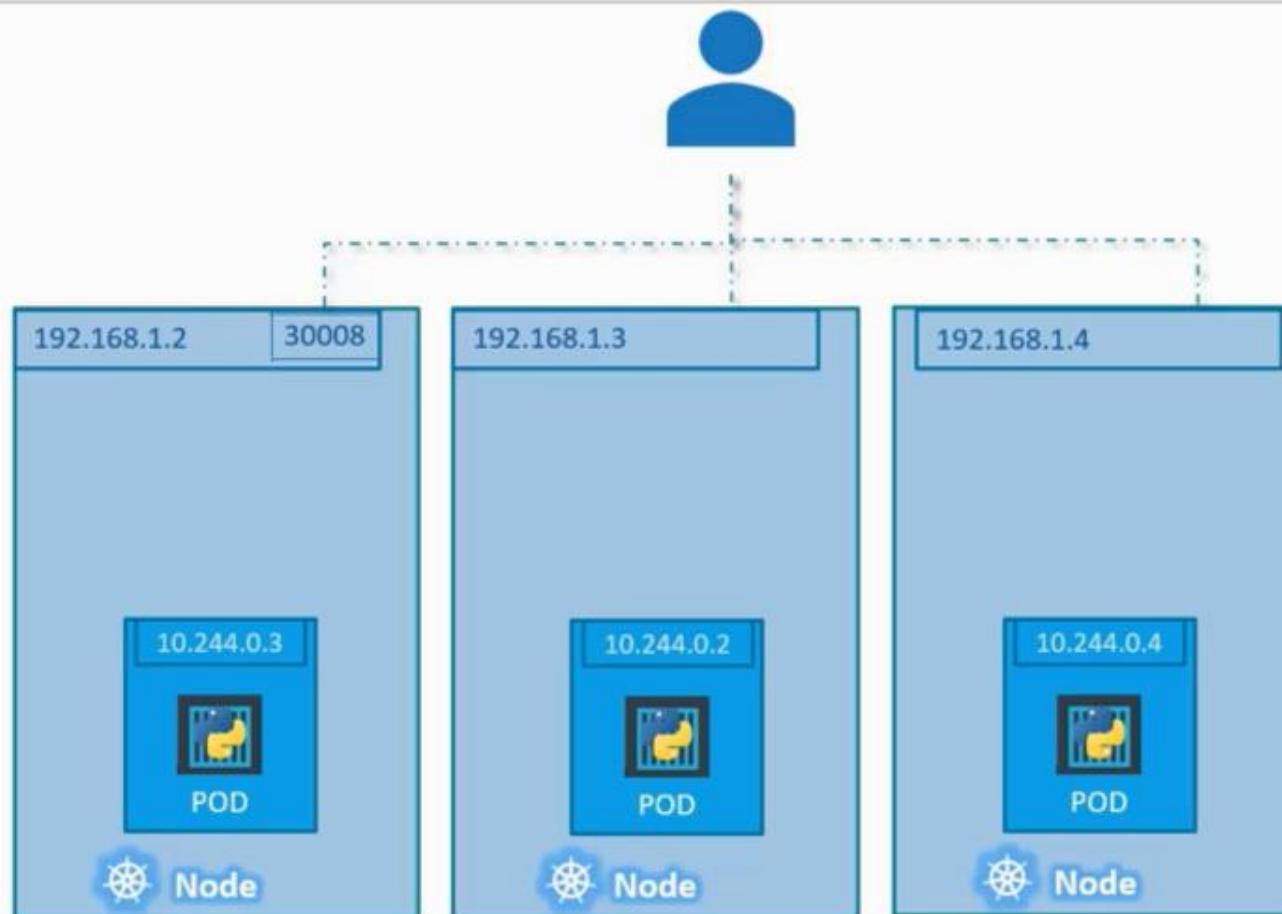
# Service - NodePort

---



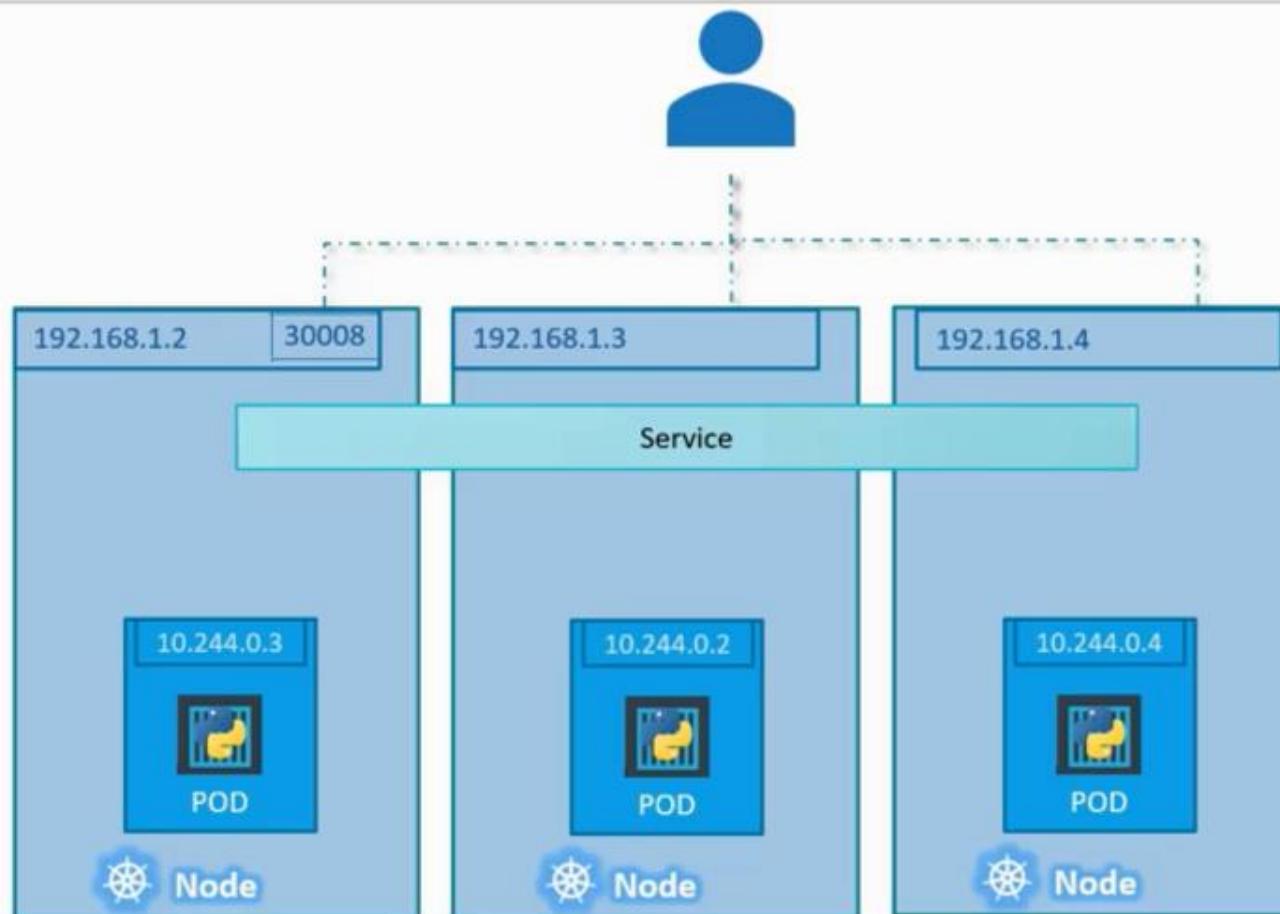
# Service - NodePort

---



# Service - NodePort

---



# Service - NodePort

