

Lab 3 K8s

1. How many Namespaces exist on the system?

```
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl get ns
NAME                STATUS    AGE
default             Active    3d4h
kube-node-lease     Active    3d4h
kube-public         Active    3d4h
kube-system         Active    3d4h
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$
```

2. How many pods exist in the kube-system namespace?

```
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$ kubectl get pods -n kube-system
NAME                                READY   STATUS    RESTARTS   AGE
coredns-668d6bf9bc-6z9sg           1/1     Running   3 (5m53s ago)    3d4h
etcd-minikube                      1/1     Running   3 (5m53s ago)    3d4h
kube-apiserver-minikube            1/1     Running   3 (5m53s ago)    3d4h
kube-controller-manager-minikube   1/1     Running   3 (5m53s ago)    3d4h
kube-proxy-gqlxm                   1/1     Running   3 (5m53s ago)    3d4h
kube-scheduler-minikube            1/1     Running   3 (5m53s ago)    3d4h
storage-provisioner                 1/1     Running   8 (3m48s ago)    3d4h
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs$
```

3. Create a deployment with:

→Name: beta

→Image: redis

→Replicas: 2

→Namespace: finance

→Resources Requests:

→CPU: 500m

→Mem: 1G

→Resources Limits:

→CPU: 1

→Mem: 2G

```
! finance.yml x
lab3 > ! finance.yml > {} spec > {} template > {} spec >
io.k8s.api.apps.v1.Deployment (v1@deployment.json)
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: beta
5    namespace: finance
6  spec:
7    replicas: 2
8    selector:
9      matchLabels:
10       app: beta
11  template:
12    metadata:
13      labels:
14       app: beta
15    spec:
16      containers:
17       - name: redis
18         image: redis
19         resources:
20           requests:
21             cpu: "500m"
22             memory: "1Gi"
23           limits:
24             cpu: "1"
25             memory: "2Gi"
```

```

• shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs/lab3$ kubectl create namespace finance
namespace/finance created
• shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs/lab3$ kubectl apply -f finance.yml
deployment.apps/beta created
• shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs/lab3$ █

```

4. Apply a label color=blue to the master node

```

• shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs/lab3$ kubectl label node minikube color=blue
node/minikube labeled

```

```

• shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs/lab3$ kubectl get nodes --show-labels | grep color
minikube Ready control-plane 3d5h v1.32.0 beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,color=blue,kubernetes
kubernetes.io/os=linux,minikube.k8s.io/commit=dd5d320e41b5451cdf3c01891bc4e13d189586ed-dirty,minikube.k8s.io/name=minikube,mini
pdated_at=2025_02_09T09_14_48_0700,minikube.k8s.io/version=v1.35.0,node-role.kubernetes.io/control-plane=,node.kubernetes.io/exclude
• shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs/lab3$ █

```

5. Create a new deployment named blue with the nginx image and 2 replicas

→Set Node Affinity to the deployment to place the pods on master only

→NodeAffinity: requiredDuringSchedulingIgnoredDuringExecution

→Key: color

→values: blue

```

lab3 > ! blue.yml > {} spec > {} template > {} spec > [ ] containers > {} 0 > image
io.k8s.api.apps.v1.Deployment (v1@deployment.json)
1  apiVersion: apps/v1
2  kind: Deployment
3  ∨ metadata:
4    | name: blue
5  ∨ spec:
6    | replicas: 2
7    ∨ selector:
8    |   matchLabels:
9    |     | app: blue
10   ∨ template:
11   ∨   metadata:
12   ∨     labels:
13   |     | app: blue
14   ∨   spec:
15   ∨     affinity:
16   ∨       nodeAffinity:
17   ∨         requiredDuringSchedulingIgnoredDuringExecution:
18   ∨           nodeSelectorTerms:
19   ∨             - matchExpressions:
20   ∨               - key: "color"
21   |               operator: In
22   ∨               values:
23   |               - "blue"
24   ∨     containers:
25   ∨       - name: nginx
26   |       image: nginx

```

6. Create a namespace named “iti” with a resource quota on pods “2”

```
lab3 > ! iti.yml > {} spec > {} hard > pods
io.k8s.api.core.v1.ResourceQuota (v1@resourcequota.json)
1  apiVersion: v1
2  kind: ResourceQuota
3  metadata:
4    name: pod-quota
5    namespace: iti
6  spec:
7    hard:
8      pods: "2"
```

7. Create a deployment named “nginx” with image “nginx”, replicas 3 on the “iti” namespace

```
lab3 > ! nginx-deployment.yml > {} spec > {} template >
io.k8s.api.apps.v1.Deployment (v1@deployment.json)
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: nginx
5    namespace: iti
6  spec:
7    replicas: 3
8    selector:
9      matchLabels:
10       app: nginx
11   template:
12     metadata:
13       labels:
14         app: nginx
15     spec:
16       containers:
17       - name: nginx
18         image: nginx
```

8. How many pods have been created within the nginx deployment and why?

```
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs/lab3$ kubectl get pods -n iti
NAME                                READY   STATUS    RESTARTS   AGE
nginx-5869d7778c-2xwnz             1/1     Running   0           90s
nginx-5869d7778c-lbbhg             1/1     Running   0           90s
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs/lab3$
```

Only 2 pods were created even though 3 replicas were used because the ResourceQuota restriction limits the number of pods to 2 only so the 3rd only remained in a pending state.

9. How many DaemonSets are created in the cluster in all namespaces?

```
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs/lab3$ kubectl get daemonsets --all-namespaces
NAMESPACE   NAME           DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
default      elasticsearch   1         1         1       1            1           <none>          2d
default      nginx           1         1         1       1            1           <none>          2d
kube-system  kube-proxy      1         1         1       1            1           kubernetes.io/os=linux 3d6h
```

10. what DaemonSets exist on the kube-system namespace?

```
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs/lab3$ kubectl get daemonsets -n kube-system
NAME          DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
kube-proxy    1         1         1       1            1           kubernetes.io/os=linux 3d6h
```

11. What is the image used by the POD deployed by the kube-proxy DaemonSet?

```
shehab-gamal@shehab-gamal-Lenovo-ideapad-520-15IKB:~/Kubernetes-labs/lab3$ kubectl describe daemonset kube-proxy -n kube-system | grep Image
Image: registry.k8s.io/kube-proxy:v1.32.0
```

12. Taint node01, the taint should have:

- Key: special-node
- Value: true
- Effect: NoSchedule

```
controlplane $ kubectl get nodes
NAME          STATUS    ROLES          AGE   VERSION
controlplane  Ready     control-plane  21h   v1.31.0
node01        Ready     <none>         20h   v1.31.0
controlplane $ kubectl taint nodes node01 special-node=true:NoSchedule
node/node01 tainted
controlplane $
```

13. Create a pod named tolerant-pod that runs nginx.

```
lab3 > ! tolerant-pod.yml > {} spec > [ ] c
io.k8s.api.core.v1.Pod (v1@pod.json)
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: tolerant-pod
5  spec:
6    containers:
7      - name: nginx
8        image: nginx
```

14. On which node this pod scheduled & why?

It will run on any available nodes other than node01 because the pod doesn't have the tolerance to run on NoSchedule taint.

15. Tolerate pod tolerant-pod with the same taint that is on node01

```
lab3 > ! tolerant-pod.yml > {} spec > [ ] containers > {} 0 > image
io.k8s.api.core.v1.Pod (v1@pod.json)
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: tolerant-pod
5  spec:
6    tolerations:
7      - key: "special-node"
8        operator: "Equal"
9        effect: "NoSchedule"
10       value: "true"
11    containers:
12      - name: nginx
13        image: nginx
```

16. Now, on which node this pod scheduled & why?

The pod can now run normally on node01 because it has a matching toleration for the taint or can run any other nodes.