

K8s 07

Task 1: Execute all yaml files shown in video.

-- Executed all yaml files in below tasks and remaining also executed which are showed in the video.

---Yaml to add toleration to pod using operator Exists:

```
root@master: ~# cat first.yaml
apiVersion: v1
kind: Pod
metadata:
  name: firstpod
spec:
  containers:
  - name: firstcontainer
    image: nginx
    imagePullPolicy: Never
  tolerations:
  - key: "special"
    operator: "Exists"      #if we haven't specified operator then it will pick default as Equal only.
    effect: "NoSchedule"
```

-run the yaml

```
root@master:~# kubectl apply -f first.yaml
pod/firstpod created
```

-check pods

```
root@master:~# kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
firstpod      1/1     Running   0           8s
```

```
root@master:~# kubectl get pod -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP              NODE              NOMINATED NODE   READINESS GATES
firstpod      1/1     Running   0           8m20s  192.168.201.226  ip-172-31-13-158  <none>           <none>
```

-describe the firstpod

In Exists it will check if any key with value special is there not and it will apply toleration.

Here Tolerations operator as exists means it Waits 5 mins before evicting

```
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                  node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
                  special:NoSchedule op=Exists
```

```
Events:
  Type     Reason      Age   From              Message
  ----     -
  Normal   Scheduled   5m49s  default-scheduler  Successfully assigned default/firstpod to ip-172-31-13-158
  Normal   Pulled      5m48s  kubelet           Container image "nginx" already present on machine
  Normal   Created     5m48s  kubelet           Created container: firstcontainer
  Normal   Started     5m48s  kubelet           Started container firstcontainer
```

Task 2: Taint a Node and Schedule a Tolerant Pod

Taint a node with special=true:NoSchedule.

Create a pod with a toleration that matches the taint, allowing it to be scheduled on the tainted node.

--Tainted a Node (worker-01=ip-172-31-13-158)

```
root@master:~# kubectl taint nodes ip-172-31-13-158 special=true:NoSchedule
node/ip-172-31-13-158 tainted
```

--yaml to Create a pod with a toleration that matches the taint, allowing it to be scheduled on the tainted node

```
root@master: ~
apiVersion: v1
kind: Pod
metadata:
  name: tolerant-pod
spec:
  containers:
  - name: nginx
    image: nginx
  tolerations:
  - key: "special"
    operator: "Equal"
    value: "true"
    effect: "NoSchedule"
```

-run yaml

```
kubectl apply -f tolerant-pod.yaml
```

--verified that the pod got scheduled on the tainted node only

```
root@master:~# kubectl get pod tolerant-pod -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP            NODE          NOMINATED NODE   READINESS GATES
tolerant-pod  1/1     Running   0          22s   192.168.201.219  ip-172-31-13-158  <none>           <none>
```

- describe the pod

```
Events:
  Type     Reason      Age   From          Message
  ----     -
Normal    Scheduled   88s   default-scheduler   Successfully assigned default/tolerant-pod to ip-172-31-13-158
Normal    Pulling     88s   kubelet          Pulling image "nginx"
Normal    Pulled      87s   kubelet          Successfully pulled image "nginx" in 254ms (254ms including waiting)
Normal    Created     87s   kubelet          Created container: nginx
Normal    Started     87s   kubelet          Started container nginx
```

Task 3: Use NodeSelector to Schedule a Pod on a Specific Node

Label a node with env=dev.

Create a pod with a nodeSelector that schedules it only on the node labeled env=dev.

-- Labeled the Node (worker-02=ip-172-31-4-112) with env=dev

```
root@master:~# kubectl label nodes ip-172-31-4-112 env=dev
node/ip-172-31-4-112 labeled
```

--yaml file to Create a Pod with nodeSelector

```
root@master: ~
apiVersion: v1
kind: Pod
metadata:
  name: dev-node-pod
spec:
  containers:
  - name: nginx
    image: nginx
  nodeSelector:
    env: dev
```

--run yaml

```
kubectl apply -f dev-node-pod.yaml
```

pod created

```
root@master:~# kubectl apply -f dev-node-pod.yaml
pod/dev-node-pod created
```

-- check if the pod is running on the right node

```
kubectl get pod dev-node-pod -o wide
```

Created a pod with a nodeSelector that schedules pod only on the node labeled env=dev.

```
root@master:~# kubectl get pod dev-node-pod -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP              NODE          NOMINATED NODE   READINESS GATES
dev-node-pod  1/1     Running   0           96s   192.168.123.130 ip-172-31-4-112 <none>          <none>
```

--describe pod

kubectl describe pod dev-node-pod

```
Node-Selectors:      env=dev
Tolerations:         node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                    node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason      Age   From          Message
  ----     -
Normal    Scheduled   3m21s default-scheduler Successfully assigned default/dev-node-pod to ip-172-31-4-112
Normal    Pulling     3m21s kubelet       Pulling image "nginx"
Normal    Pulled      3m21s kubelet       Successfully pulled image "nginx" in 232ms (232ms including waiting)
Normal    Created     3m21s kubelet       Created container: nginx
Normal    Started     3m21s kubelet       Started container nginx
```

Task 4: Use Node Affinity with Soft Scheduling

Label a node with env=test.

Create a pod with PreferredDuringSchedulingIgnoredDuringExecution node affinity, preferring to schedule it on a node labeled env=test.

Remove the label and verify the pod continues to run.

-- Labeled a Node(worker-01=ip-172-31-13-158) with env=test

```
root@master:~# kubectl label nodes ip-172-31-13-158 env=test
node/ip-172-31-13-158 labeled
```

```
root@master:~# kubectl get nodes --show-labels
NAME                STATUS    ROLES    AGE   VERSION   LABELS
ip-172-31-13-158    Ready     <none>    6d23h v1.29.15  beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,env=test,kubernetes.io/arch=amd64,kubernetes.io/hostname=ip-172-31-13-158,kubernetes.io/os=linux
ip-172-31-4-112     Ready     <none>    6d23h v1.29.15  beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,env=dev,kubernetes.io/arch=amd64,kubernetes.io/hostname=ip-172-31-4-112,kubernetes.io/os=linux
master              Ready     control-plane 7d    v1.29.15  beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=master,kubernetes.io/os=linux,node-role.kubernetes.io/control-plane=node.kubernetes.io/exclude-from-external-load-balancers=
```

--yaml file to Create a Pod with Soft Node Affinity

```
root@master: ~
apiVersion: v1
kind: Pod
metadata:
  name: soft-affinity-pod
spec:
  containers:
  - name: nginx
    image: nginx
  affinity:
    nodeAffinity:
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 1
        preference:
          matchExpressions:
          - key: env
            operator: In
            values:
            - test
```

-run the yaml

```
kubectl apply -f soft-affinity-pod.yaml
```

```
root@master:~# kubectl apply -f soft-affinity-pod.yaml
pod/soft-affinity-pod created
```

-- Checked where the pod was scheduled:

```
kubectl get pod soft-affinity-pod -o wide
```

showing it running on the node labeled env=test.

```
root@master:~# kubectl get pod soft-affinity-pod -o wide
NAME                READY   STATUS    RESTARTS   AGE   IP            NODE                NOMINATED NODE   READINESS GATES
soft-affinity-pod   1/1     Running   0          13s   192.168.201.221   ip-172-31-13-158   <none>           <none>
```

--describe it

```
kubectl describe pod soft-affinity-pod
```

```
Events:
  Type     Reason      Age   From              Message
  ----     -
Normal    Scheduled   2m40s    default-scheduler   Successfully assigned default/soft-affinity-pod to ip-172-31-13-158
Normal    Pulling     2m40s    kubelet             Pulling image "nginx"
Normal    Pulled      2m40s    kubelet             Successfully pulled image "nginx" in 200ms (200ms including waiting)
Normal    Created     2m40s    kubelet             Created container: nginx
Normal    Started     2m39s    kubelet             Started container nginx
```

-- Removed the Label and Check Pod Status

```
kubectl label nodes <node-name> env-
```

```
root@master:~# kubectl label nodes ip-172-31-13-158 env-
node/ip-172-31-13-158 unlabeled
```

--then checked that the pod is still running or not

```
root@master:~# kubectl get pod soft-affinity-pod
NAME                READY   STATUS    RESTARTS   AGE
soft-affinity-pod   1/1     Running   0          9m2s
```

It will continue running because PreferredDuringSchedulingIgnoredDuringExecution affects only scheduling, not execution.

Task 5: Implement Node Affinity with Hard Scheduling

Create a pod with RequiredDuringSchedulingIgnoredDuringExecution node affinity, ensuring it will only be scheduled on a node labeled env=prod.

Verify the pod cannot be scheduled if no node has the env=prod label.

--yaml to Create a Pod with Hard Node Affinity

```
root@master: ~  
apiVersion: v1  
kind: Pod  
metadata:  
  name: hard-affinity-pod  
spec:  
  containers:  
  - name: nginx  
    image: nginx  
  affinity:  
    nodeAffinity:  
      requiredDuringSchedulingIgnoredDuringExecution:  
        nodeSelectorTerms:  
        - matchExpressions:  
          - key: env  
            operator: In  
            values:  
            - prod
```

--run the yaml

```
kubectl apply -f hard-affinity-pod.yaml
```

```
root@master:~# kubectl apply -f hard-affinity-pod.yaml  
pod/hard-affinity-pod created
```

-- Verified Pod Is Not Scheduled it stuck in Pending state only and the pod cannot be scheduled if no node has the env=prod label.

```
root@master:~# kubectl get pods  
NAME                READY   STATUS    RESTARTS   AGE  
hard-affinity-pod   0/1     Pending   0           46s  
soft-affinity-pod   1/1     Running   0           21m
```

--describe pod

```
kubectl describe pod hard-affinity-pod
```

```
Events:
  Type     Reason          Age    From          Message
  ----     -
  Warning  FailedScheduling  2m34s  default-scheduler  0/3 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane: }, 2 node(s) didn't match Pod's node affinity/selector.
preemption: 0/3 nodes are available: 3 Preemption is not helpful for scheduling.
```

It confirmed the pod can't be scheduled due to the hard affinity rule.

--now Add the Required Label to node

```
root@master:~# kubectl label node ip-172-31-13-158 env=prod  
node/ip-172-31-13-158 labeled
```

Now again run the yaml

```
root@master:~# kubectl apply -f hard-affinity-pod.yaml
pod/hard-affinity-pod unchanged
```

Check status of pod now it is running

```
root@master:~# kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
hard-affinity-pod	1/1	Running	0	8m39s
soft-affinity-pod	1/1	Running	0	29m


```
root@master:~# kubectl get pod hard-affinity-pod -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED	NODE	READINESS	GATES
hard-affinity-pod	1/1	Running	0	9m44s	192.168.201.220	ip-172-31-13-158	<none>		<none>	

--describe it

```
node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
```

Events:	Type	Reason	Age	From	Message
	Warning	FailedScheduling	10m	default-scheduler	0/3 nodes are available: 1 node(s) had intolerated taint {node-role.kubernetes.io/control-plane: }, 2 node(s) didn't match Pod's node affinity/selector, preemption: 0/3 nodes are available: 3 Preemption is not helpful for scheduling.
	Warning	FailedScheduling	4m43s	default-scheduler	0/3 nodes are available: 1 node(s) had intolerated taint {node-role.kubernetes.io/control-plane: }, 2 node(s) didn't match Pod's node affinity/selector, preemption: 0/3 nodes are available: 3 Preemption is not helpful for scheduling.
Normal	Scheduled		3m35s	default-scheduler	Successfully assigned default/hard-affinity-pod to ip-172-31-13-158
Normal	Pulling		3m34s	kubelet	Pulling image "nginx"
Normal	Pulled		3m34s	kubelet	Successfully pulled image "nginx" in 230ms (230ms including waiting)
Normal	Created		3m34s	kubelet	created container: nginx
Normal	Started		3m34s	kubelet	started container nginx

Task 6: Taint a Node and Use NoExecute with Toleration Seconds

Taint a node with special=true:NoExecute.

Create a pod with a tolerationSeconds field (e.g., 60 seconds) and observe it gets evicted after 60 seconds on the tainted node.

-- Taint the Node with special=true:NoExecute

```
root@master:~# kubectl taint nodes ip-172-31-4-112 special=true:NoExecute
node/ip-172-31-4-112 tainted
```

--yaml file to Create a Pod with tolerationSeconds

```

root@master: ~
apiVersion: v1
kind: Pod
metadata:
  name: timeout-toleration-pod
spec:
  containers:
  - name: nginx
    image: nginx
  tolerations:
  - key: "special"
    operator: "Equal"
    value: "true"
    effect: "NoExecute"
    tolerationSeconds: 60

```

--run the yaml

```
kubectl apply -f timeout-toleration-pod.yaml
```

```

root@master:~# kubectl apply -f timeout-toleration-pod.yaml
pod/timeout-toleration-pod created

```

--now Observe Behavior

Confirmed the pod schedules to the tainted node

```

root@master:~# kubectl get pod timeout-toleration-pod -o wide
NAME                READY   STATUS    RESTARTS   AGE   IP              NODE                NOMINATED NODE   READINESS GATES
timeout-toleration-pod 1/1     Running   0           17s   192.168.123.135  ip-172-31-4-112    <none>           <none>

```

-- Then wait 60 seconds Check the pod status again

```
kubectl get pod timeout-toleration-pod
```

```

root@master:~# kubectl get pod timeout-toleration-pod -o wide
NAME                READY   STATUS    RESTARTS   AGE   IP              NODE                NOMINATED NODE   READINESS GATES
timeout-toleration-pod 1/1     Terminating   0           60s   192.168.123.133  ip-172-31-4-112    <none>           <none>
root@master:~# kubectl get pod timeout-toleration-pod
Error from server (NotFound): pods "timeout-toleration-pod" not found
root@master:~# kubectl get pod timeout-toleration-pod -o wide
Error from server (NotFound): pods "timeout-toleration-pod" not found

```

At 60 seconds pod starts terminating

observed it gets evicted after 60 seconds on the tainted node.

```

root@master:~# kubectl get pod timeout-toleration-pod
Error from server (NotFound): pods "timeout-toleration-pod" not found
root@master:~# kubectl get pods
No resources found in default namespace.

```