Task 1: Execute all yaml files shown in video.

vi firstpod.yml

apiVersion: v1

kind: Pod

metadata:

name: firstpod

spec:

containers:

- name: firstcontainer

image: nginx

imagePullPolicy: Never

tolerations:

- key: "special"

operator: "Equal"

value: "true"

effect: "NoSchedule"

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: firstpod
spec:
  containers:
    - name: firstcontainer
      image: nginx
      imagePullPolicy: Never
  tolerations:
    - key: "special"
      operator: "Equal"
      value: "true"
      effect: "NoSchedule"
```

- **kubectl get nodes**
- **kubectl taint nodes worker-02 special=true:NoSchedule**
- **kubectl describe node worker-02 | grep -i taint**

```
[root@master ~]# vi firstpod.yml
[root@master ~]# kubectl get nodes
NAME         STATUS   ROLES           AGE   VERSION
master       Ready    control-plane   10d   v1.34.3
worker-01    Ready    <none>          10d   v1.34.3
worker-02    Ready    <none>          10d   v1.34.3
[root@master ~]# kubectl taint nodes worker-02 special=true:NoSchedule
node/worker-02 tainted
[root@master ~]# kubectl describe node worker-02 | grep -i taint
Taints:             special=true:NoSchedule
```

- **kubectl apply -f firstpod.yml**
- **kubectl describe pod firstpod**

```
[root@master ~]# kubectl get pods -o wide
NAME       READY   STATUS    RESTARTS   AGE   IP           NODE        NOMINATED NODE   READINESS GATES
firstpod   1/1     Running   0          42s   10.244.2.56   worker-02   <none>           <none>
[root@master ~]# kubectl describe pod firstpod
Name:             firstpod
Namespace:        default
Priority:         0
Service Account:  default
Node:             worker-02/172.31.72.26
Start Time:       Sat, 27 Dec 2025 06:52:53 +0000
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.2.56
IPs:
  IP:  10.244.2.56
Containers:
  firstcontainer:
    Container ID:   containerd://2c994c77634a29c1764c84af171dd72cefc6c4f123f5e36700020ef9b4eac1a4
    Image:          nginx
    Image ID:       docker.io/library/nginx@sha256:fb01117203ff38c2f9af91db1a7409459182a37c87cced5cb442d
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Sat, 27 Dec 2025 06:52:53 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
```

<u>NoExecute</u>:

apiVersion: v1

kind: Pod

metadata:

name: firstpod

spec:

containers:

- name: firstcontainer

image: nginx

imagePullPolicy: Never

tolerations:

- key: "special"

operator: "Exists"

effect: "NoExecute"

tolerationSeconds: 60

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: firstpod
spec:
  containers:
    - name: firstcontainer
      image: nginx
      imagePullPolicy: Never
  tolerations:
    - key: "special"
      operator: "Exists"
      effect: "NoExecute"
      tolerationSeconds: 60
```

- **kubectl get nodes**
- **docker pull nginx**
- **kubectl apply -f firstpod.yml**

```
[root@master ~]# kubectl get nodes
NAME        STATUS   ROLES           AGE   VERSION
master      Ready    control-plane   10d   v1.34.3
worker-01   Ready    <none>          10d   v1.34.3
worker-02   Ready    <none>          10d   v1.34.3
[root@master ~]# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
1733a4cd5954: Pull complete
5b219a92f92a: Pull complete
ee3a09d2248a: Pull complete
7382b41547b8: Pull complete
9ee60c6c0558: Pull complete
114e699da838: Pull complete
5b5fa0b64d74: Pull complete
Digest: sha256:fb01117203ff38c2f9af91db1a7409459182a37c87cced5cb442d1d8fcc66d19
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
[root@master ~]# docker images | grep nginx
nginx        latest    576306625d79   2 weeks ago   152MB
[root@master ~]# kubectl apply -f firstpod.yml
pod/firstpod created
```

- **kubectl taint nodes worker-02 special=true:NoExecute**
- **kubectl get pods -o wide**

```
[root@master ~]# kubectl get pods
NAME       READY   STATUS    RESTARTS   AGE
firstpod   1/1     Running   0          21s
[root@master ~]# kubectl taint nodes worker-02 special=true:NoExecute
node/worker-02 tainted
[root@master ~]# kubectl get pods -w
NAME       READY   STATUS    RESTARTS   AGE
firstpod   1/1     Running   0          77s
^C[root@master ~]# kubectl get pods -o wide
NAME       READY   STATUS    RESTARTS   AGE     IP            NODE        NOMINATED NODE   READINESS GA
firstpod   1/1     Running   0          3m17s   10.244.1.58   worker-01   <none>           <none>
```

NodeSelector:

apiVersion: v1

kind: Pod

metadata:

name: firstpod

spec:

containers:

- name: firstcontainer

image: nginx

imagePullPolicy: Never

nodeSelector:

env: prod

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: firstpod
spec:
  containers:
    - name: firstcontainer
      image: nginx
      imagePullPolicy: Never
  nodeSelector:
    env: prod
```

- **kubectl get nodes --show-labels**
- **kubectl label node worker-02 env=prod**

```
[root@master ~]# kubectl get nodes --show-labels
NAME        STATUS   ROLES           AGE   VERSION   LABELS
master      Ready    control-plane   10d   v1.34.3   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/
,kubernetes.io/hostname=master,kubernetes.io/os=linux,node-role.kubernetes.io/control-plane=,node.kubernetes.io/exclude-from-
oad-balancers=
worker-01   Ready    <none>          10d   v1.34.3   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/
,kubernetes.io/hostname=worker-01,kubernetes.io/os=linux
worker-02   Ready    <none>          10d   v1.34.3   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/
,kubernetes.io/hostname=worker-02,kubernetes.io/os=linux
[root@master ~]# kubectl label node worker-02 env=prod
node/worker-02 labeled
[root@master ~]# kubectl get nodes --show-labels | grep prod
worker-02   Ready    <none>          10d   v1.34.3   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,env=prod,kuber
rch=amd64,kubernetes.io/hostname=worker-02,kubernetes.io/os=linux
```

- **kubectl apply -f firstpod.yml**
- **kubectl get pods -o wide**

```
[root@master ~]# kubectl apply -f firstpod.yml
pod/firstpod created
[root@master ~]# kubectl get pods -o wide
NAME       READY   STATUS    RESTARTS   AGE   IP       NODE     NOMINATED NODE   READINESS GA
firstpod   0/1     Pending   0          11s   <none>   <none>   <none>           <none>
[root@master ~]#
```

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.

- **kubectl get nodes**
- **kubectl taint nodes worker-02 special=true:NoSchedule**
- **kubectl describe node worker-02 | grep -i taint**

```
[root@master ~]# kubectl get nodes
NAME        STATUS   ROLES           AGE   VERSION
master      Ready    control-plane   10d   v1.34.3
worker-01   Ready    <none>          10d   v1.34.3
worker-02   Ready    <none>          10d   v1.34.3
[root@master ~]# kubectl taint nodes worker-02 special=true:NoSchedule
node/worker-02 tainted
[root@master ~]# kubectl describe node <node-name> | grep -i taint
-bash: syntax error near unexpected token `|'
[root@master ~]# kubectl describe node worker-02 | grep -i taint
Taints:             special=true:NoExecute
```

- **vi tolerant-pod.yml**

apiVersion: v1

kind: Pod

metadata:

name: tolerant-pod

spec:

containers:

- name: nginx-container

image: nginx

imagePullPolicy: IfNotPresent

tolerations:

- key: "special"

operator: "Equal"

value: "true"

effect: "NoSchedule"

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: tolerant-pod
spec:
  containers:
    - name: nginx-container
      image: nginx
      imagePullPolicy: IfNotPresent
  tolerations:
    - key: "special"
      operator: "Equal"
      value: "true"
      effect: "NoSchedule"
```

- **kubectl apply -f tolerant-pod.yml**
- **kubectl get pods -o wide**

```
[root@master ~]# vi tolerant-pod.yml
[root@master ~]# kubectl apply -f tolerant-pod.yml
pod/tolerant-pod created
[root@master ~]# kubectl get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP            NODE        NOMINATED
tolerant-pod  1/1     Running   0          12s   10.244.1.59   worker-01   <none>
[root@master ~]#
[root@master ~]#
```

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.

- **kubectl get nodes**
- **kubectl label node worker-01 env=dev**

```
[root@master ~]# kubectl get nodes
NAME         STATUS   ROLES           AGE   VERSION
master       Ready    control-plane   10d   v1.34.3
worker-01    Ready    <none>          10d   v1.34.3
worker-02    Ready    <none>          10d   v1.34.3
[root@master ~]# kubectl label node worker-01 env=dev
node/worker-01 labeled
```

- **kubectl get nodes --show-labels**

```
[root@master ~]# kubectl get nodes --show-labels
NAME        STATUS   ROLES           AGE   VERSION   LABELS
master      Ready    control-plane   10d   v1.34.3   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kuberne
tes.io/arch=amd64,kubernetes.io/hostname=master,kubernetes.io/os=linux,node-role.kubernetes.io/control-plane=,node.kub
ernetes.io/exclude-from-external-load-balancers=
worker-01   Ready    <none>          10d   v1.34.3   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,env=dev
,kubernetes.io/arch=amd64,kubernetes.io/hostname=worker-01,kubernetes.io/os=linux
worker-02   Ready    <none>          10d   v1.34.3   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,env=pro
d,kubernetes.io/arch=amd64,kubernetes.io/hostname=worker-02,kubernetes.io/os=linux
```

vi dev-pod.yml

apiVersion: v1

kind: Pod

metadata:

name: dev-pod

spec:

nodeSelector:

env: dev

containers:

- name: nginx

  image: nginx

```
apiVersion: v1
kind: Pod
metadata:
  name: dev-pod
spec:
  nodeSelector:
    env: dev
  containers:
  - name: nginx
    image: nginx
```

- kubectl apply -f dev-pod.yml
- kubectl get pod dev-pod -o wide

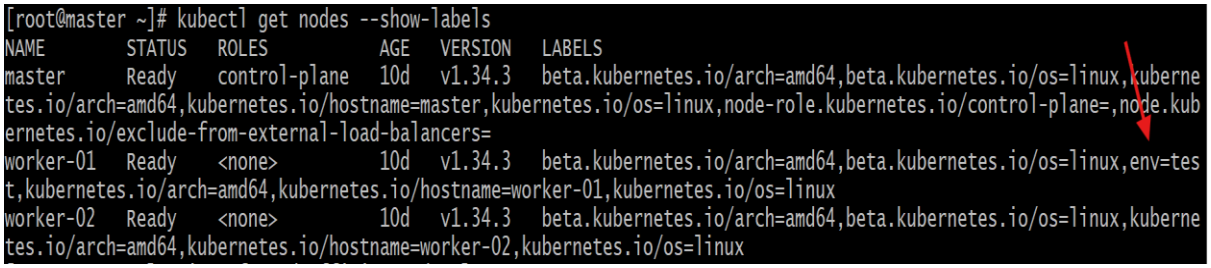
```
[root@master ~]# vi dev-pod.yml
[root@master ~]# kubectl apply -f dev-pod.yml
pod/dev-pod created
[root@master ~]# kubectl get pod dev-pod -o wide
NAME      READY   STATUS    RESTARTS   AGE   IP            NODE        NOMINAT
dev-pod   1/1     Running   0          14s   10.244.1.60   worker-01   <none>
[root@master ~]#
```

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.

- kubectl get nodes
- kubectl label node worker-01 env=test

```
[root@master ~]# kubectl get nodes
NAME          STATUS   ROLES           AGE   VERSION
master        Ready    control-plane   10d   v1.34.3
worker-01     Ready    <none>          10d   v1.34.3
worker-02     Ready    <none>          10d   v1.34.3
[root@master ~]# kubectl label node worker-01 env=test
node/worker-01 labeled
```

- **kubectl get nodes --show-labels**



vi preferred-affinity-pod.yml

apiVersion: v1

kind: Pod

metadata:

name: affinity-pod

spec:

affinity:

nodeAffinity:

preferredDuringSchedulingIgnoredDuringExecution:

- weight: 100

preference:

matchExpressions:

- key: env

operator: In

values:

- test

containers:

- name: nginx

image: nginx

```
apiVersion: v1
kind: Pod
metadata:
  name: affinity-pod
spec:
  affinity:
    nodeAffinity:
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 100
        preference:
          matchExpressions:
          - key: env
            operator: In
            values:
            - test
  containers:
  - name: nginx
    image: nginx
```

- **kubectl apply -f preferred-affinity-pod.yml**
- **kubectl get pod affinity-pod -o wide**

```
[root@master ~]# vi preferred-affinity-pod.yml
[root@master ~]# kubectl apply -f preferred-affinity-pod.yml
pod/affinity-pod created
[root@master ~]# kubectl get pod affinity-pod -o wide
NAME           READY   STATUS    RESTARTS   AGE   IP            NODE        NOMINATED
affinity-pod   1/1     Running   0          10s   10.244.1.61   worker-01   <none>
```

- **kubectl label node worker-01 env-**
- **kubectl get nodes --show-labels**

- **kubectl get pod affinity-pod**

```
[root@master ~]# kubectl label node worker1 env-
Error from server (NotFound): nodes "worker1" not found
[root@master ~]# kubectl label node worker-01 env-
node/worker-01 unlabeled
[root@master ~]# kubectl get nodes --show-labels
NAME        STATUS   ROLES           AGE   VERSION   LABELS
master      Ready    control-plane   10d   v1.34.3   beta.kubernetes.io/arch=amd64,bet
tes.io/arch=amd64,kubernetes.io/hostname=master,kubernetes.io/os=linux,node-role.kuber
ernetes.io/exclude-from-external-load-balancers=
worker-01   Ready    <none>          10d   v1.34.3   beta.kubernetes.io/arch=amd64,bet
tes.io/arch=amd64,kubernetes.io/hostname=worker-01,kubernetes.io/os=linux
worker-02   Ready    <none>          10d   v1.34.3   beta.kubernetes.io/arch=amd64,bet
tes.io/arch=amd64,kubernetes.io/hostname=worker-02,kubernetes.io/os=linux
[root@master ~]# kubectl get pod affinity-pod
NAME           READY   STATUS    RESTARTS   AGE
affinity-pod   1/1     Running   0          74s
[root@master ~]#
```

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.

vi required-affinity-pod.yml

apiVersion: v1

kind: Pod

metadata:

name: prod-affinity-pod

spec:

affinity:

nodeAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

nodeSelectorTerms:

- matchExpressions:

- key: env

operator: In

values:

- prod

containers:

- name: nginx

image: nginx

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

- **kubectl apply -f required-affinity-pod.yml**
- **kubectl get pod prod-affinity-pod**

```
[root@master ~]# vi required-affinity-pod.yml
[root@master ~]# kubectl apply -f required-affinity-pod.yml
pod/prod-affinity-pod created
[root@master ~]# kubectl get pod prod-affinity-pod
NAME                READY   STATUS    RESTARTS   AGE
prod-affinity-pod   0/1     Pending   0          11s
```

- **kubectl describe pod prod-affinity-pod**

```
Node-Selectors:              <none>
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
  ----     ------            ----  ----              -------
  Warning  FailedScheduling  21s   default-scheduler  0/3 nodes are available: 1 node(s) didn't match Pod's node affin
ity/selector, 2 node(s) had untolerated taint(s). no new claims to deallocate, preemption: 0/3 nodes are available: 3
Preemption is not helpful for scheduling.
```

- **kubectl label node worker1 env=prod**
- **kubectl get pod prod-affinity-pod -o wide**

```
[root@master ~]# kubectl label node worker-01 env=prod
node/worker-01 labeled
[root@master ~]# kubectl get pod prod-affinity-pod -o wide
NAME                READY   STATUS    RESTARTS   AGE   IP            NODE        NO
prod-affinity-pod   1/1     Running   0          58s   10.244.1.62   worker-01   <n
[root@master ~]#
```

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.

- **kubectl get nodes**
- **kubectl taint node worker-01 special=true:NoExecute**

```
[root@master ~]# kubectl get nodes
NAME        STATUS   ROLES           AGE   VERSION
master      Ready    control-plane   10d   v1.34.3
worker-01   Ready    <none>          10d   v1.34.3
worker-02   Ready    <none>          10d   v1.34.3
[root@master ~]# kubectl taint node worker-01 special=true:NoExecute
node/worker-01 tainted
```

- **kubectl describe node worker-01 | grep -i taint**
- **vi noexecute-pod.yml**

```
apiVersion: v1
kind: Pod
metadata:
  name: noexecute-pod
spec:
  tolerations:
  - key: "special"
    operator: "Equal"
    value: "true"
    effect: "NoExecute"
    tolerationSeconds: 60
  containers:
  - name: nginx
    image: nginx
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: noexecute-pod
spec:
  tolerations:
  - key: "special"
    operator: "Equal"
    value: "true"
    effect: "NoExecute"
    tolerationSeconds: 60
  containers:
  - name: nginx
    image: nginx
```

- **kubectl apply -f noexecute-pod.yml**

```
[root@master ~]# vi noexecute-pod.yml
[root@master ~]# kubectl apply -f noexecute-pod.yml
pod/noexecute-pod created
[root@master ~]# kubectl get pod noexecute-pod -o wide
NAME           READY   STATUS    RESTARTS   AGE   IP            NODE        NOMINA
noexecute-pod   1/1     Running   0          10s   10.244.1.63   worker-01   <none>
```

After 60 sec see the pod it will be deleted

- **kubectl get pod noexecute-pod**

```
[root@master ~]# kubectl get pod noexecute-pod
Error from server (NotFound): pods "noexecute-pod" not found
```