

Taint and tolerations (1 and 2 session)

Scheduler puts container/pod, where it gets suitable for pods which doesn't define where it has to run in which node. In this case, nodeselector doesn't work, that's why taint and toleration came into the light.

If node got tainted, which has toleration token only will be able to run pod.

```
ter
3) Taint and tolerations Based Scheduling:
fdf
;559
;559      1) it is giving dedicate tency.
;559
;559
```

Why pods don't go on master nodes?

```
root@master01:~# kubectl describe node master01 | grep -i taint
Taints:          node-role.kubernetes.io/control-plane:NoSchedule
root@master01:~#
```

```
root@master01:~#
root@master01:~# kubectl taint node worker01 payment:NoSchedule
```

now, it is not normal node

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up with email

Sign up with Google

Sign up with GitHub

Sign up with Facebook

Toleration

```
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: krnetwork
  name: krnetwork
spec:
  replicas: 1
  selector:
    matchLabels:
      app: krnetwork
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: krnetwork
    spec:
      tolerations:
        - key: payment
          effect: NoSchedule
          operator: Exists
      containers:
        - image: nginx
          name: nginx
          resources: {}
status: {}
```

:wc

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

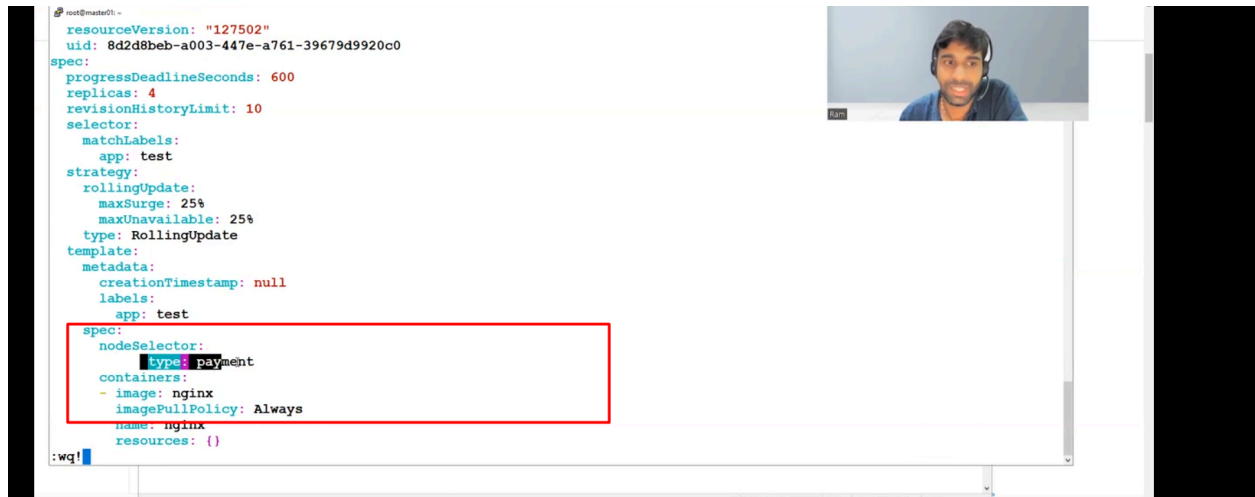
Sign up with email

Sign up with Google

Sign up with GitHub

Sign up with Facebook

If you replicate the pods, pods will run on any nodes. It isn't mandatory for pods to run on tainted nodes, then you have to add the nodeSelector parameter in the manifest.



```

root@master01:~# cat deployment.yaml
resourceVersion: "127502"
uid: 8d2d8beb-a003-447e-a761-39679d9920c0
spec:
  progressDeadlineSeconds: 600
  replicas: 4
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: test
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
    labels:
      app: test
    spec:
      nodeSelector:
        type: payment
      containers:
      - image: nginx
        imagePullPolicy: Always
        name: nginx
        resources: {}

```

2nd session



```

root@master01:~# kubectl get pod -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE     NOMINATED NODE   READINESS GATES
test-5746d4c59f-d7wb7              1/1     Running   0           84s   10.244.19.80    worker03 <none>         <none>

root@master01:~# kubectl taint node worker03 blue:NoSchedule
node/worker03 tainted
root@master01:~# kubectl get pod -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE     NOMINATED NODE   READINESS GATES
test-5746d4c59f-d7wb7              1/1     Running   0           84s   10.244.19.80    worker03 <none>         <none>

root@master01:~# kubectl taint node worker03 blue:NoExecute
node/worker03 tainted
root@master01:~# kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
test-5746d4c59f-grqb4              1/1     Running   0           5s
root@master01:~# kubectl get pod -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE     NOMINATED NODE   READINESS GATES
test-5746d4c59f-grqb4              1/1     Running   0           9s   10.244.30.76    worker02 <none>         <none>

```

Taint effects:

- 1) NoSchedule: it does not evict existing pod even they don't have tolerations
- 2) NoExecute: it evicts the pod if they don't have tolerations.

```
root@master01:~# kubectl taint node worker03 blue:NoExecute-
node/worker03 untainted
root@master01:~#
```

untaint

```
    app: kubernetes
spec:
  tolerations:
    - key: tier
      value: db
      effect: NoSchedule
      operator: Equal
  containers:
    - image: nginx
      name: nginx
      resources: {}
status: {}
worker03
```

tolerations:

Exists: Key/effect match

Equal= key=value match

```
root@master01: ~  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  creationTimestamp: null  
  labels:  
    app: krnetwork  
  name: krnetwork  
spec:  
  replicas: 1  
  selector:  
    matchLabels:  
      app: krnetwork  
  strategy: {}  
  template:  
    metadata:  
      creationTimestamp: null  
      labels:  
        app: krnetwork  
    spec:  
      tolerations:  
        - key: node-role.kubernetes.io/control-plane  
          effect: NoSchedule  
          operator: Exists  
      containers:  
        - image: nginx  
          name: nginx  
          resources: {}  
    status: {}  
~  
:wq!
```

