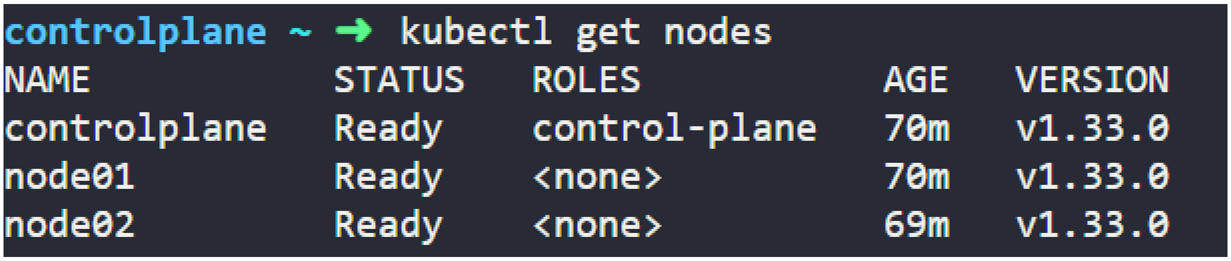
Taints & Tolerations

Taints: Node-level "repellents" that repel pods unless they tolerate the taint.

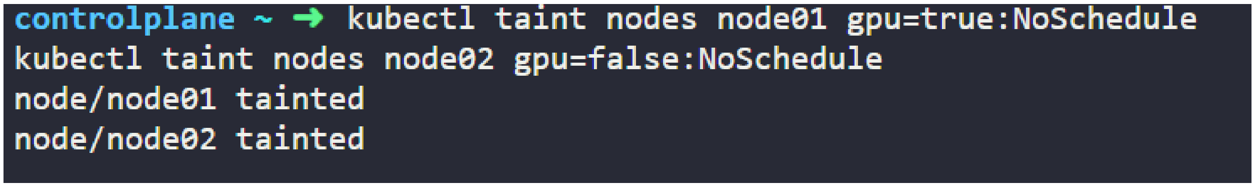
Tolerations: Pod-level "acceptance" rules that allow pods to be scheduled on tainted nodes.

Step 1: Apply Taints to Worker Nodes

These taints will prevent pods from scheduling unless they tolerate the taint.

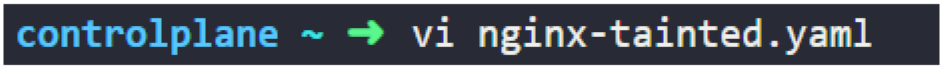


```
controlplane ~ ➜  kubectl get nodes
NAME           STATUS   ROLES           AGE   VERSION
controlplane   Ready    control-plane   70m   v1.33.0
node01         Ready    <none>          70m   v1.33.0
node02         Ready    <none>          69m   v1.33.0
```

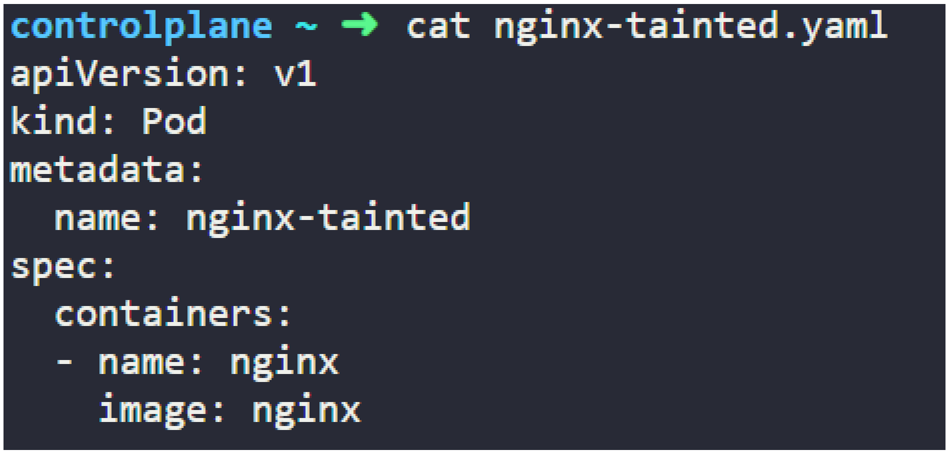


```
controlplane ~ ➜  kubectl taint nodes node01 gpu=true:NoSchedule
kubectl taint nodes node02 gpu=false:NoSchedule
node/node01 tainted
node/node02 tainted
```

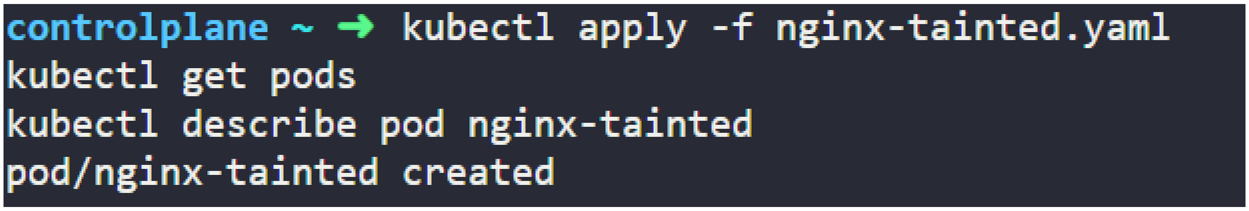
Step 2: Deploy a Pod Without Toleration



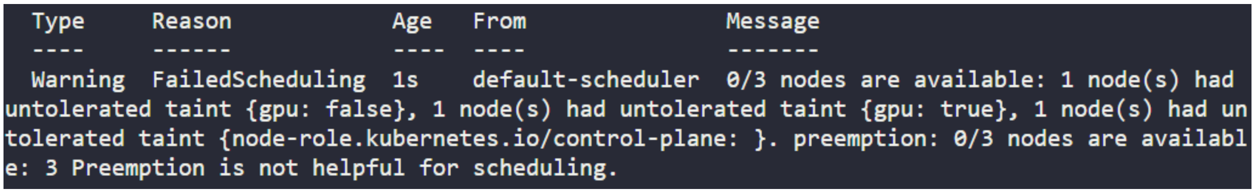
```
controlplane ~ ➜  vi nginx-tainted.yaml
```



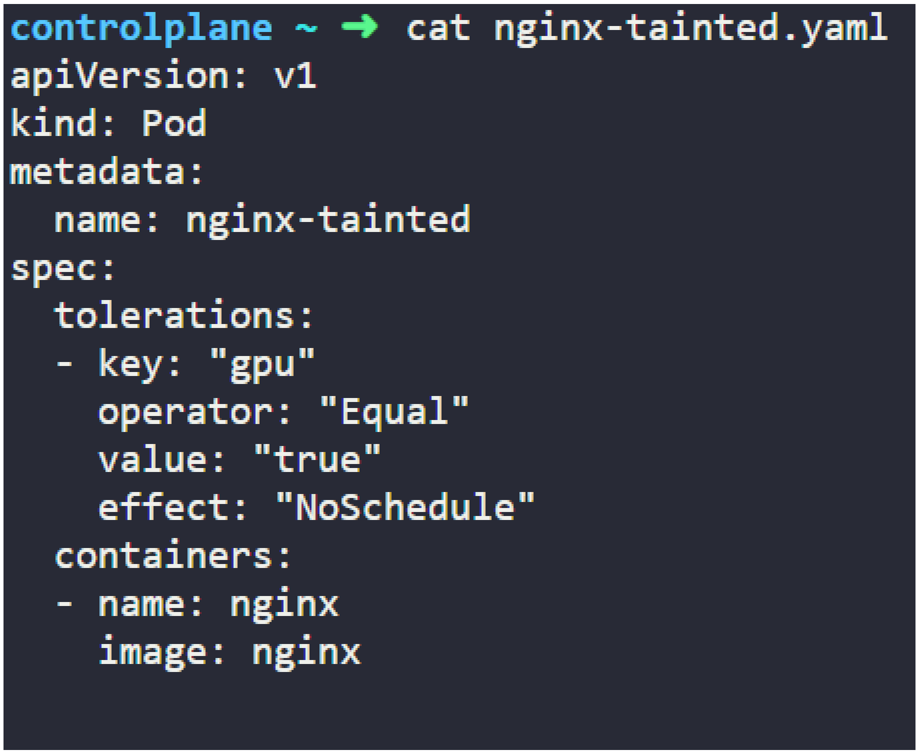
```
controlplane ~ ➜  cat nginx-tainted.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-tainted
spec:
  containers:
  - name: nginx
    image: nginx
```



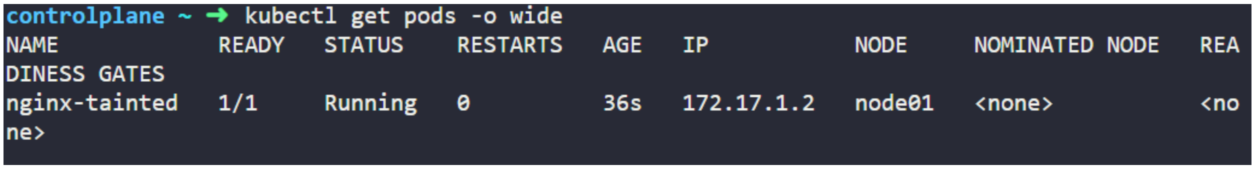
```
controlplane ~ ➜  kubectl apply -f nginx-tainted.yaml
kubectl get pods
kubectl describe pod nginx-tainted
pod/nginx-tainted created
```



```
  Type     Reason            Age   From               Message
  ----     ------            ----  ----               -------
  Warning  FailedScheduling  1s    default-scheduler  0/3 nodes are available: 1 node(s) had
untolerated taint {gpu: false}, 1 node(s) had untolerated taint {gpu: true}, 1 node(s) had un
tolerated taint {node-role.kubernetes.io/control-plane: }. preemption: 0/3 nodes are availabl
e: 3 Preemption is not helpful for scheduling.
```

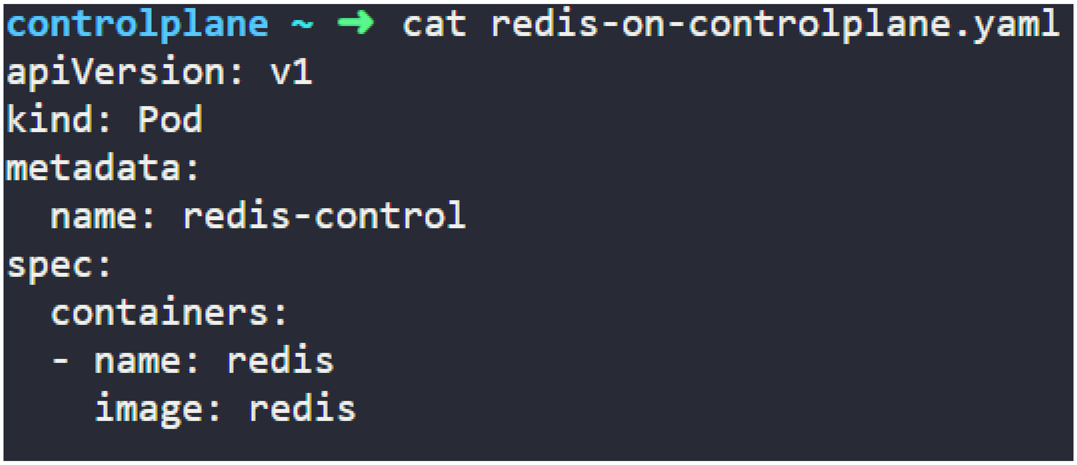


```
controlplane ~ ➜  cat nginx-tainted.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-tainted
spec:
  tolerations:
  - key: "gpu"
    operator: "Equal"
    value: "true"
    effect: "NoSchedule"
  containers:
  - name: nginx
    image: nginx
```

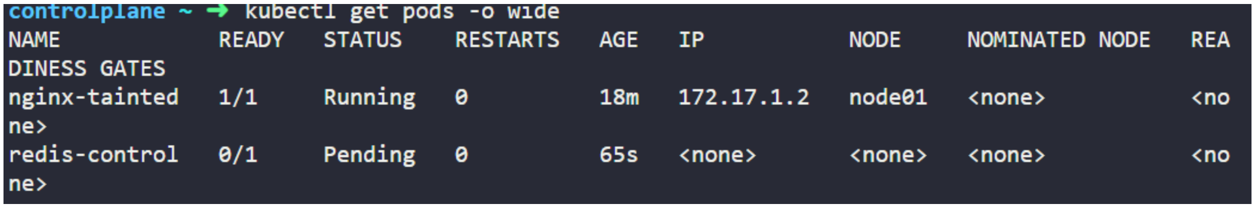


```
controlplane ~ ➜  kubectl get pods -o wide
NAME            READY   STATUS    RESTARTS   AGE   IP           NODE     NOMINATED NODE   REA
DINESS GATES
nginx-tainted   1/1     Running   0          36s   172.17.1.2   node01   <none>           <no
ne>
```

Step 3: Deploy Redis Pod to Control Plane



```
controlplane ~ ➜  cat redis-on-controlplane.yaml
apiVersion: v1
kind: Pod
metadata:
  name: redis-control
spec:
  containers:
  - name: redis
    image: redis
```



```
controlplane ~ ➜ kubectl get pods -o wide
NAME            READY   STATUS    RESTARTS   AGE   IP           NODE     NOMINATED NODE   REA
DINESS GATES
nginx-tainted   1/1     Running   0          18m   172.17.1.2   node01   <none>           <no
ne>
redis-control   0/1     Pending   0          65s   <none>       <none>   <none>           <no
ne>
```