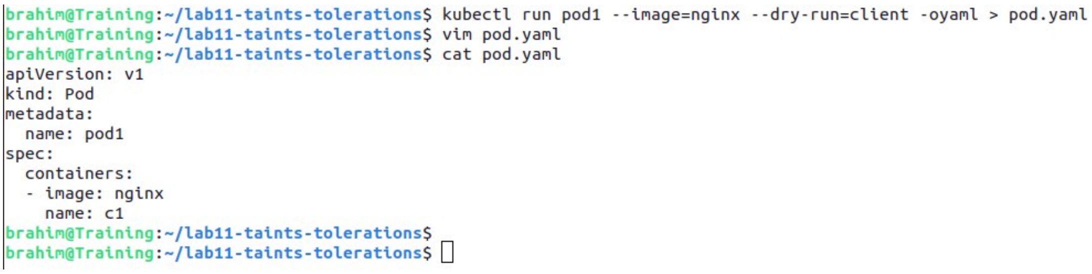
Lab11 – Configuring a Node to Only Accept Specific Pods

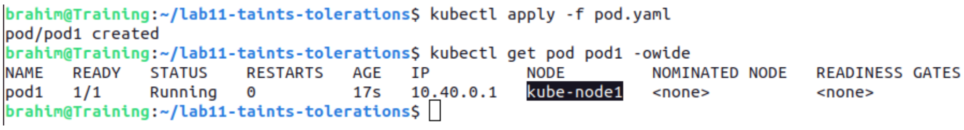
In this exercise, you will use the concept of taints and tolerations. First, you'll create a Pod. This Pod will be scheduled on one of the nodes. Next, you will add a taint to the node the Pod is running on and set a toleration effect that evicts the Pod from the node.

1. Define a Pod with the image `nginx` in the YAML manifest file `pod.yaml`



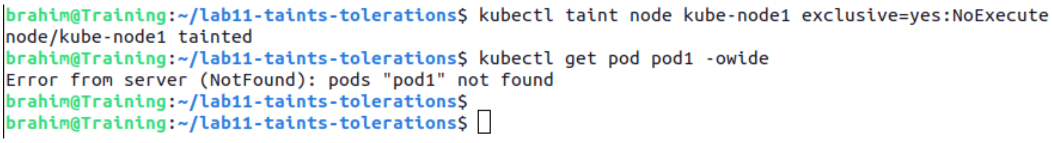
```
brahim@Training:~/lab11-taints-tolerations$ kubectl run pod1 --image=nginx --dry-run=client -oyaml > pod.yaml
brahim@Training:~/lab11-taints-tolerations$ vim pod.yaml
brahim@Training:~/lab11-taints-tolerations$ cat pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod1
spec:
  containers:
  - image: nginx
    name: c1
brahim@Training:~/lab11-taints-tolerations$
brahim@Training:~/lab11-taints-tolerations$
```

2. Create the Pod and check which node the Pod is running on.



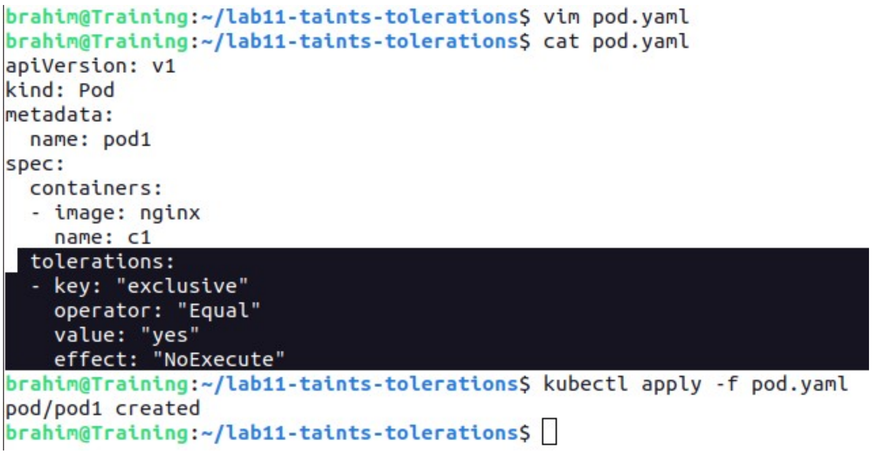
```
brahim@Training:~/lab11-taints-tolerations$ kubectl apply -f pod.yaml
pod/pod1 created
brahim@Training:~/lab11-taints-tolerations$ kubectl get pod pod1 -owide
NAME   READY   STATUS    RESTARTS   AGE   IP          NODE         NOMINATED NODE   READINESS GATES
pod1   1/1     Running   0          17s   10.40.0.1   kube-node1   <none>           <none>
brahim@Training:~/lab11-taints-tolerations$
```

3. Add a taint to the node. Set it to `exclusive: yes`.



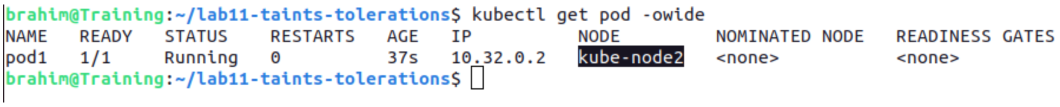
```
brahim@Training:~/lab11-taints-tolerations$ kubectl taint node kube-node1 exclusive=yes:NoExecute
node/kube-node1 tainted
brahim@Training:~/lab11-taints-tolerations$ kubectl get pod pod1 -owide
Error from server (NotFound): pods "pod1" not found
brahim@Training:~/lab11-taints-tolerations$
brahim@Training:~/lab11-taints-tolerations$
```

4. Modify the live Pod object by adding the following toleration: It should be equal to taint key-value pair and have the effect `NoExecute`.



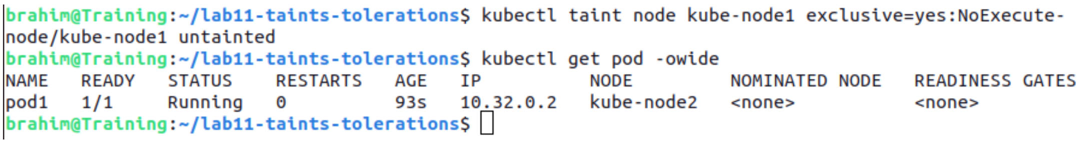
```
brahim@Training:~/lab11-taints-tolerations$ vim pod.yaml
brahim@Training:~/lab11-taints-tolerations$ cat pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod1
spec:
  containers:
  - image: nginx
    name: c1
  tolerations:
  - key: "exclusive"
    operator: "Equal"
    value: "yes"
    effect: "NoExecute"
brahim@Training:~/lab11-taints-tolerations$ kubectl apply -f pod.yaml
pod/pod1 created
brahim@Training:~/lab11-taints-tolerations$
```

5. Observe the running behavior of the Pod. If your cluster has more than a single node where do you expect the Pod to run?



```
brahim@Training:~/lab11-taints-tolerations$ kubectl get pod -owide
NAME    READY   STATUS    RESTARTS   AGE   IP          NODE         NOMINATED NODE   READINESS GATES
pod1    1/1     Running   0          37s   10.32.0.2   kube-node2   <none>           <none>
brahim@Training:~/lab11-taints-tolerations$
```

6. Remove the taint from the node. Do you expect the Pod to still run on the node?



```
brahim@Training:~/lab11-taints-tolerations$ kubectl taint node kube-node1 exclusive=yes:NoExecute-
node/kube-node1 untainted
brahim@Training:~/lab11-taints-tolerations$ kubectl get pod -owide
NAME    READY   STATUS    RESTARTS   AGE   IP          NODE         NOMINATED NODE   READINESS GATES
pod1    1/1     Running   0          93s   10.32.0.2   kube-node2   <none>           <none>
brahim@Training:~/lab11-taints-tolerations$
```