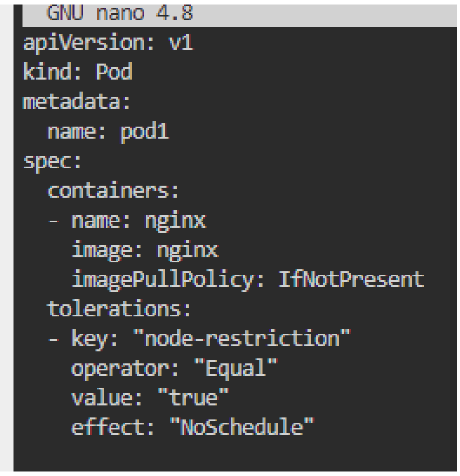
Exercise - Taints and Tolerations

Name: Haffiz Hissham

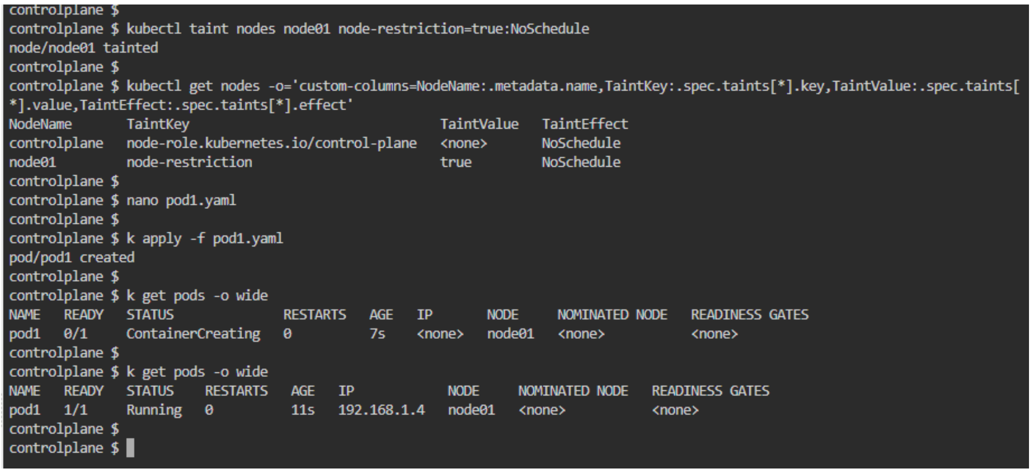
Date: 21 September 2024

Q1

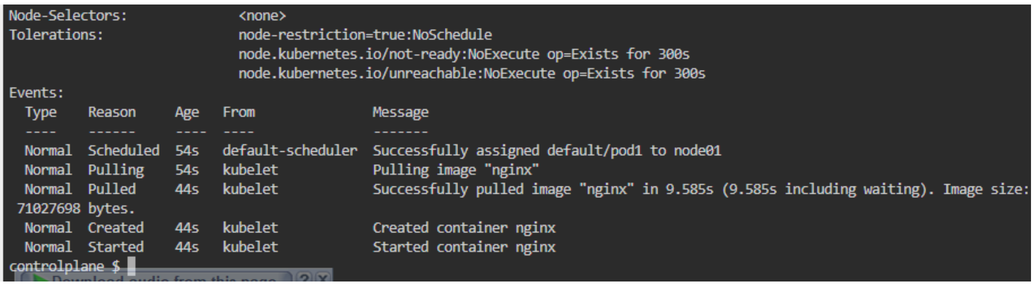
create a taint on node01 that not allow any new pod schedule on node.
node01 is the name of the node you want to taint.
key=value is the key-value pair for the taint.
 in this case use key=node-restriction and value=true.
Run a pod pod1 with image=nginx that use the toleration.
Make sure pod is running .



```
  GNU nano 4.8
apiVersion: v1
kind: Pod
metadata:
  name: pod1
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  tolerations:
  - key: "node-restriction"
    operator: "Equal"
    value: "true"
    effect: "NoSchedule"
```



```
controlplane $
controlplane $ kubectl taint nodes node01 node-restriction=true:NoSchedule
node/node01 tainted
controlplane $
controlplane $ kubectl get nodes -o='custom-columns=NodeName:.metadata.name,TaintKey:.spec.taints[*].key,TaintValue:.spec.taints[
*].value,TaintEffect:.spec.taints[*].effect'
NodeName       TaintKey                                 TaintValue   TaintEffect
controlplane   node-role.kubernetes.io/control-plane   <none>       NoSchedule
node01         node-restriction                         true         NoSchedule
controlplane $
controlplane $ nano pod1.yaml
controlplane $
controlplane $ k apply -f pod1.yaml
pod/pod1 created
controlplane $
controlplane $ k get pods -o wide
NAME   READY   STATUS              RESTARTS   AGE   IP       NODE     NOMINATED NODE   READINESS GATES
pod1   0/1     ContainerCreating   0          7s    <none>   node01   <none>           <none>
controlplane $
controlplane $ k get pods -o wide
NAME   READY   STATUS    RESTARTS   AGE   IP            NODE     NOMINATED NODE   READINESS GATES
pod1   1/1     Running   0          11s   192.168.1.4   node01   <none>           <none>
controlplane $
controlplane $ 
```



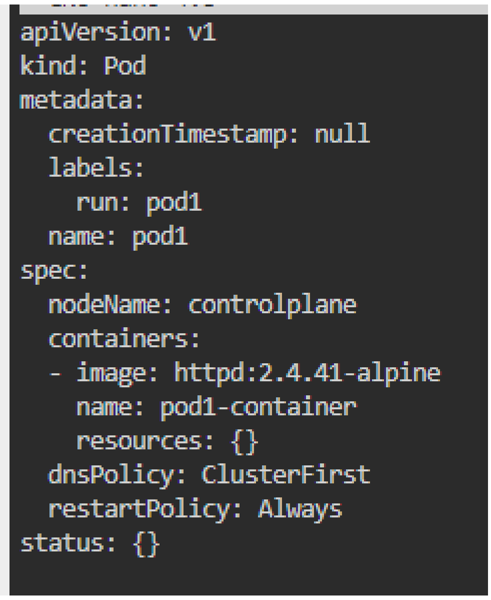
```
Node-Selectors:              <none>
Tolerations:                 node-restriction=true:NoSchedule
                             node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  54s   default-scheduler  Successfully assigned default/pod1 to node01
  Normal  Pulling    54s   kubelet            Pulling image "nginx"
  Normal  Pulled     44s   kubelet            Successfully pulled image "nginx" in 9.585s (9.585s including waiting). Image size:
 71027698 bytes.
  Normal  Created    44s   kubelet            Created container nginx
  Normal  Started    44s   kubelet            Started container nginx
controlplane $ 
```

Q2

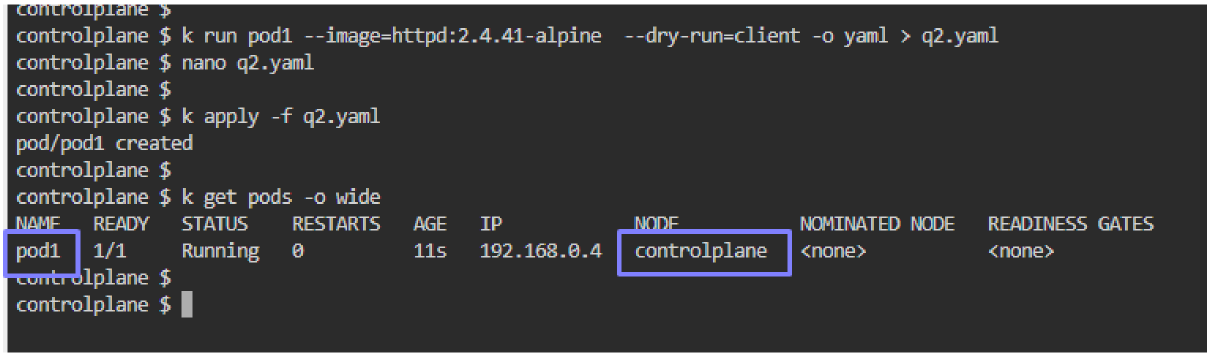
Create a single Pod of image httpd:2.4.41-alpine in Namespace default.

The Pod should be named pod1 and the container should be named pod1-container. This Pod should only be scheduled on controlplane nodes.

Do not add new labels to any nodes.



```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: pod1
  name: pod1
spec:
  nodeName: controlplane
  containers:
  - image: httpd:2.4.41-alpine
    name: pod1-container
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
```



```
controlplane $
controlplane $ k run pod1 --image=httpd:2.4.41-alpine  --dry-run=client -o yaml > q2.yaml
controlplane $ nano q2.yaml
controlplane $
controlplane $ k apply -f q2.yaml
pod/pod1 created
controlplane $
controlplane $ k get pods -o wide
NAME   READY   STATUS    RESTARTS   AGE   IP            NODE           NOMINATED NODE   READINESS GATES
pod1   1/1     Running   0          11s   192.168.0.4   controlplane   <none>           <none>
controlplane $
controlplane $ 
```