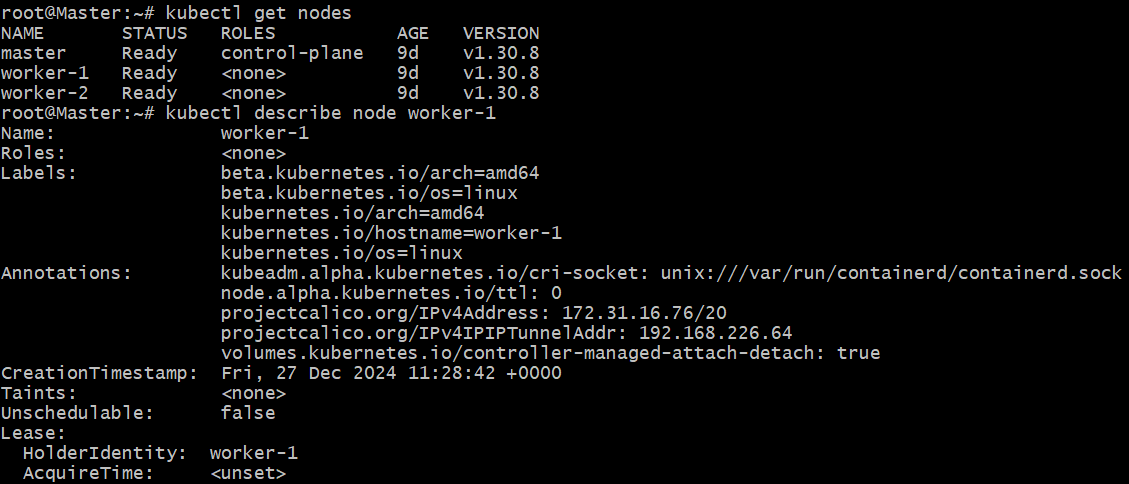
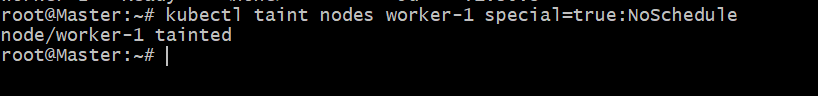
KUBERNETES[TASK-07]

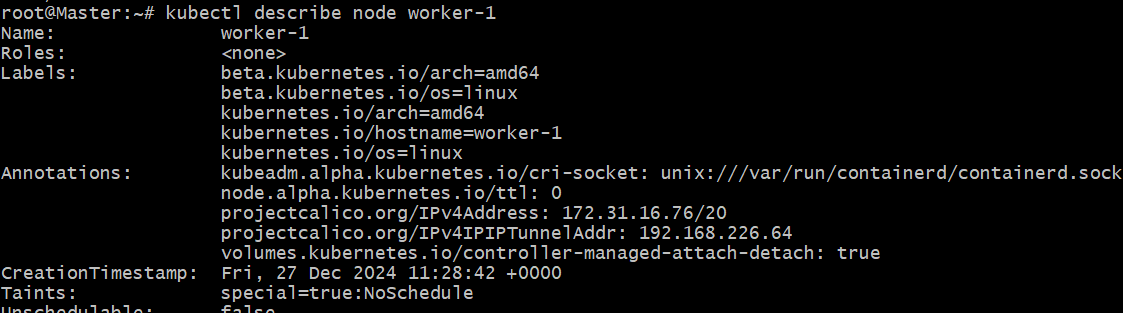
**Task 1: Execute all yaml files shown in video.**

Taint and tolerations

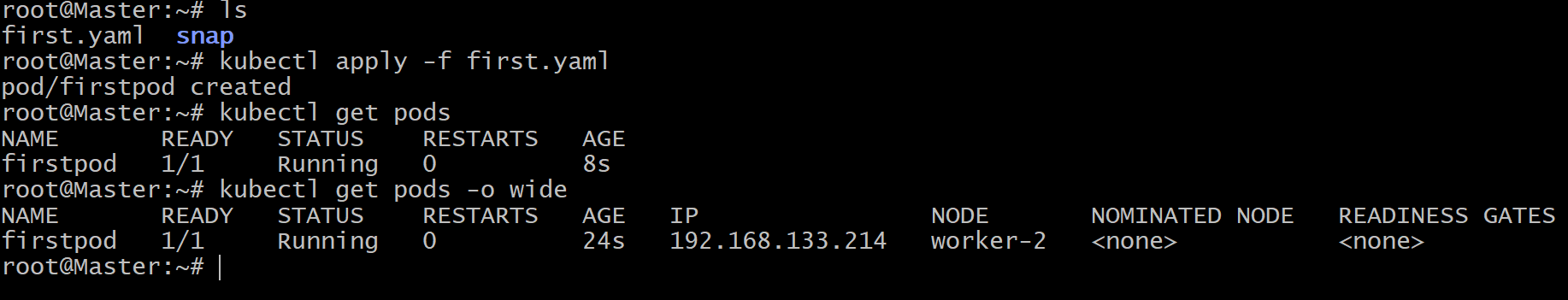


Adding Taint to the worker-1

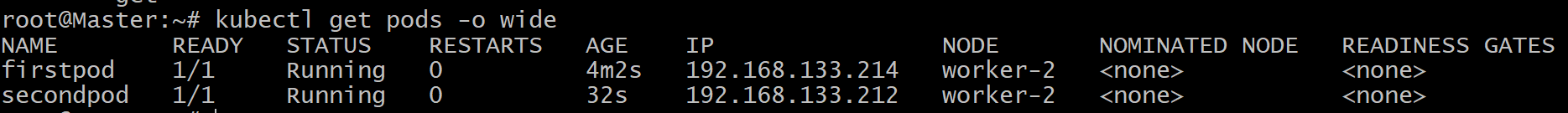




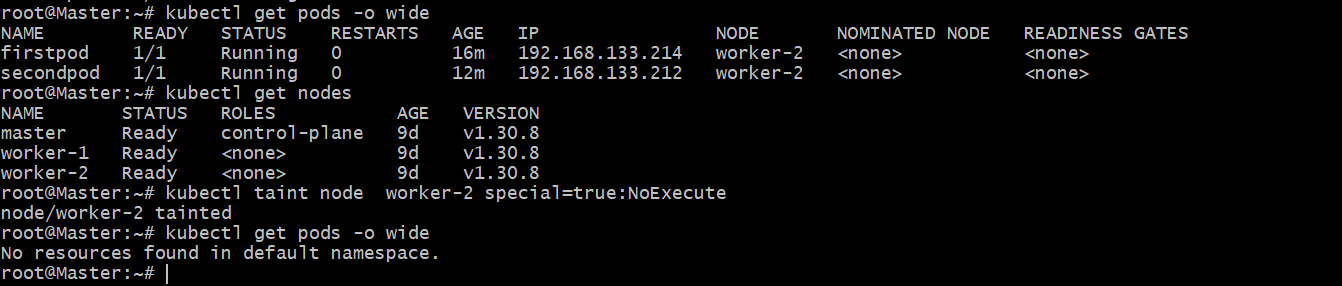
Creating a pod,to check on which node pod is getting created

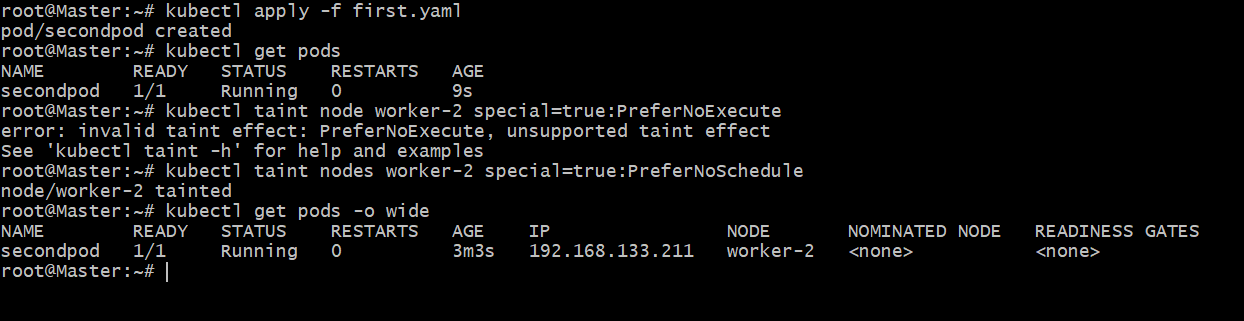


Even if we created another pod,

As taint is added to worker-1 node,

Pods will be created in the worker-2





Applying the tolerations:

apiVersion: v1

kind: Pod

metadata:

name: firstpod

spec:

containers:

- name: firstcontainer

image: nginx

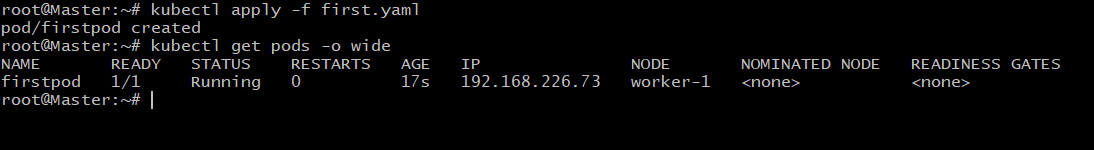
imagePullPolicy: Never

tolerations:

- key: "special"

operator: "Exists”

effect: "NoSchedule"



We can use toleration seconds in tolerations of yaml if the effect is "NoExecute"

apiVersion: v1

kind: Pod

metadata:

name: firstpod

spec:

containers:

- name: firstcontainer

image: nginx

imagePullPolicy: Never

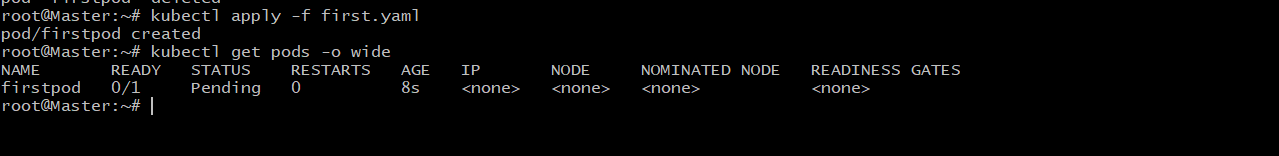
tolerations:

- key: "special"

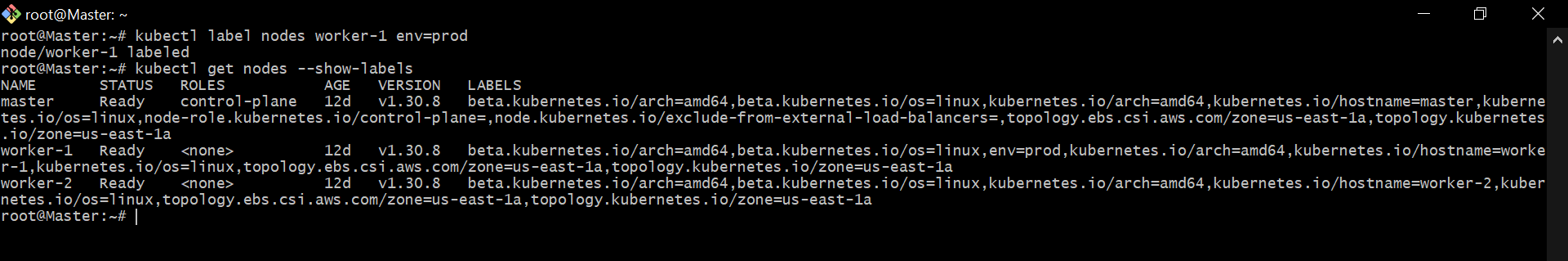
operator: "Exists" #Operator value can be Equal or Exists

effect: "NoExecute"

tolerationSeconds: 60



NODE SELECTORS:



Created a yaml file

Vi nodes.yaml

apiVersion: v1

kind: Pod

metadata:

name: firstpod

spec:

containers:

- name: firstcontainer

image: nginx

imagePullPolicy: Never

nodeSelector:

env: prod

-kubectl apply -f nodes.yaml



**NODE-AFFINITY**

Node affinity is a set of rules used by the scheduler to determine where a pod can be placed.

Two types of Node Affinity is available:

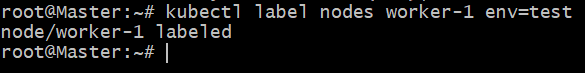
1) PreferredDuringSchedulingIgnoredDuringExecution (Soft Scheduling):

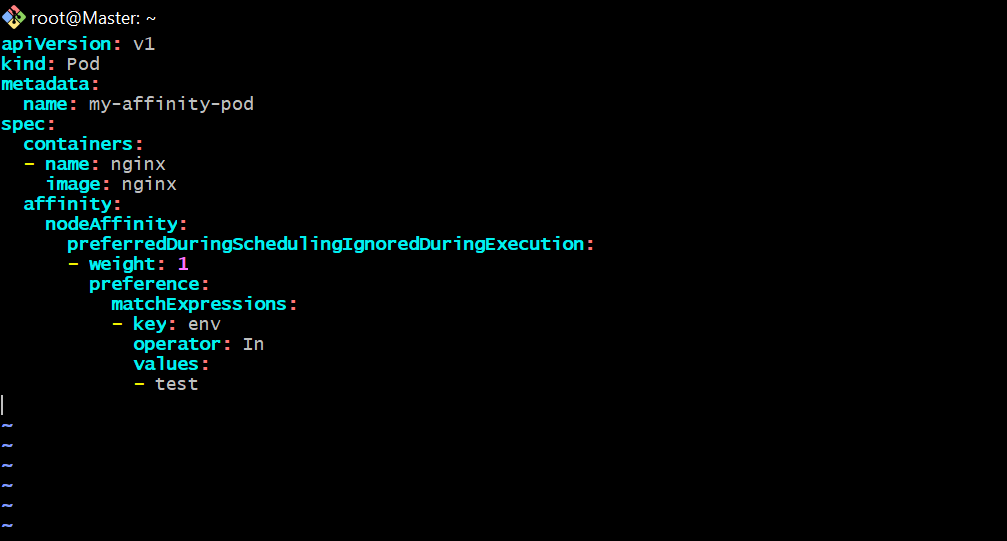
Means we are asking scheduler to check if the values/label is matching then schedule pod on the node and

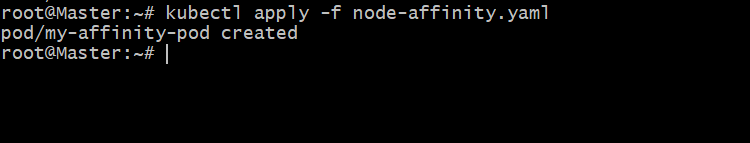
if node with same label is not available then it will schedule on any available node.

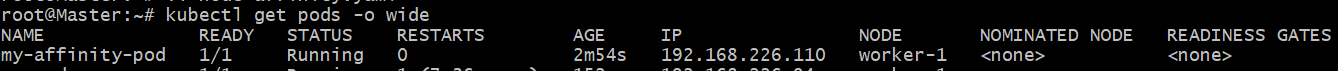
Ignored During Execution means if the pod is already scheduled on node and later on we have removed the label then

the pod will be ingored and continue running on the same node**.**

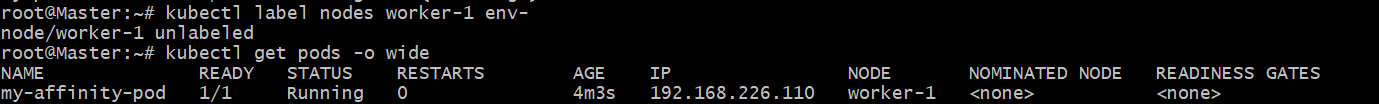








Remove the label and verify the pod continues to run.



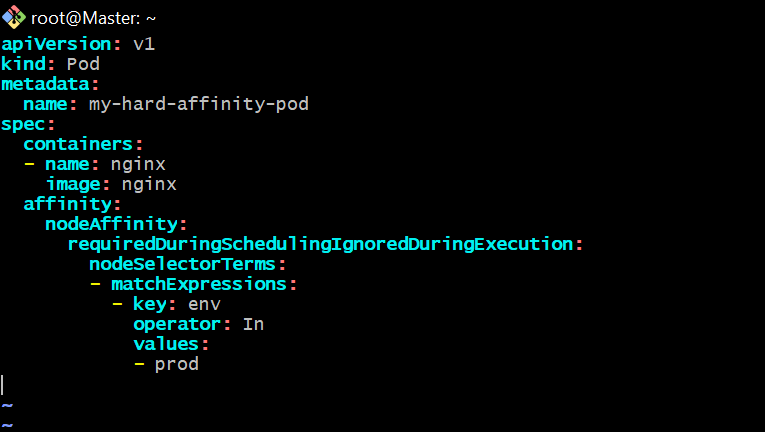
2) RequiredDuringSchedulingIgnoredDuringExecution (Hard Scheduling):

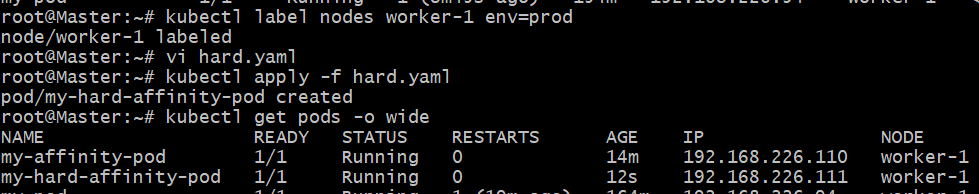
Means we are asking scheduler to check if the values/label is matching then schedule pod on the node.

If not matched then pod will not be scheduled in other nodes.

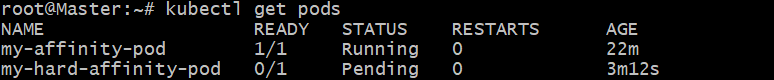
Ignored During Execution means if the pod is already scheduled on node and later on we have removed the label then

the pod will be ingored and continue running on the same node.



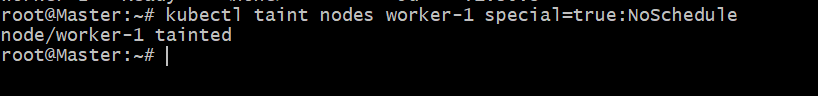


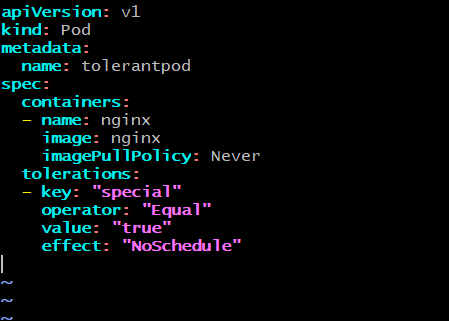
If no node has the env=prod label

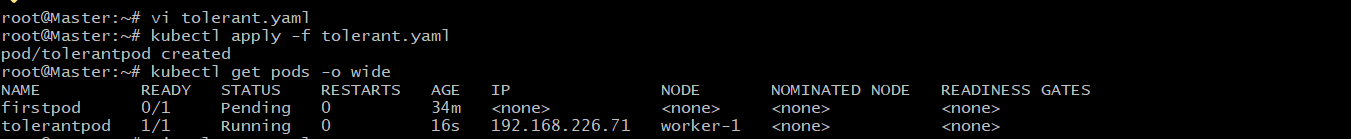


**Task 2: Taint a Node and Schedule a Tolerant Pod Taint a node with special=true:NoSchedule. Create a pod with a toleration that matches the taint, allowing it to be scheduled on the tainted node.**

Tainted a node

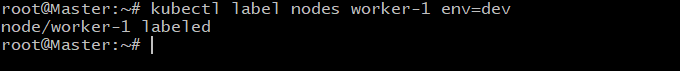


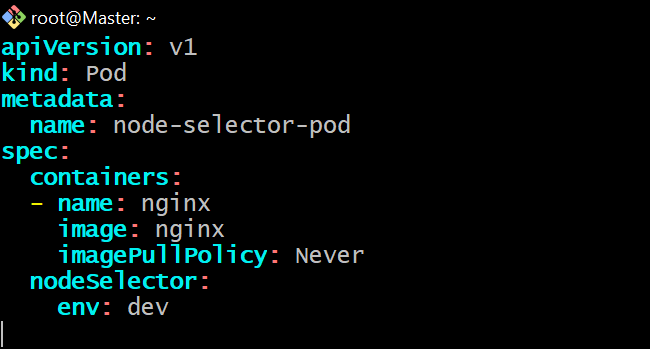




**Task 3: Use NodeSelector to Schedule a Pod on a Specific Node Label a node with env=dev. Create a pod with a nodeSelector that schedules it only on the node labeled env=dev.**

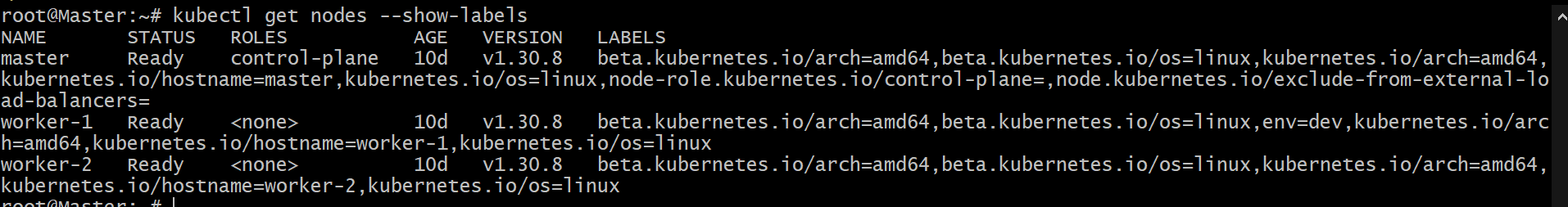
1)label the node





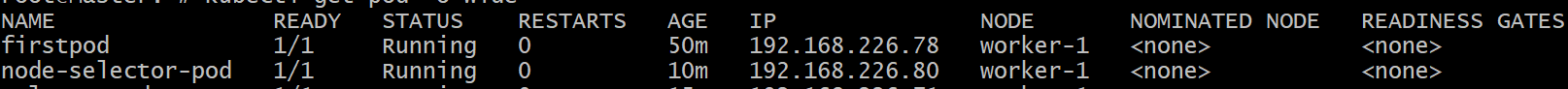
Apply the changes

Specific Node Label a node with env=dev.

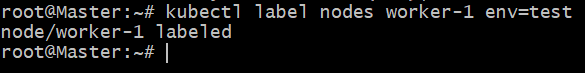


And check the pods

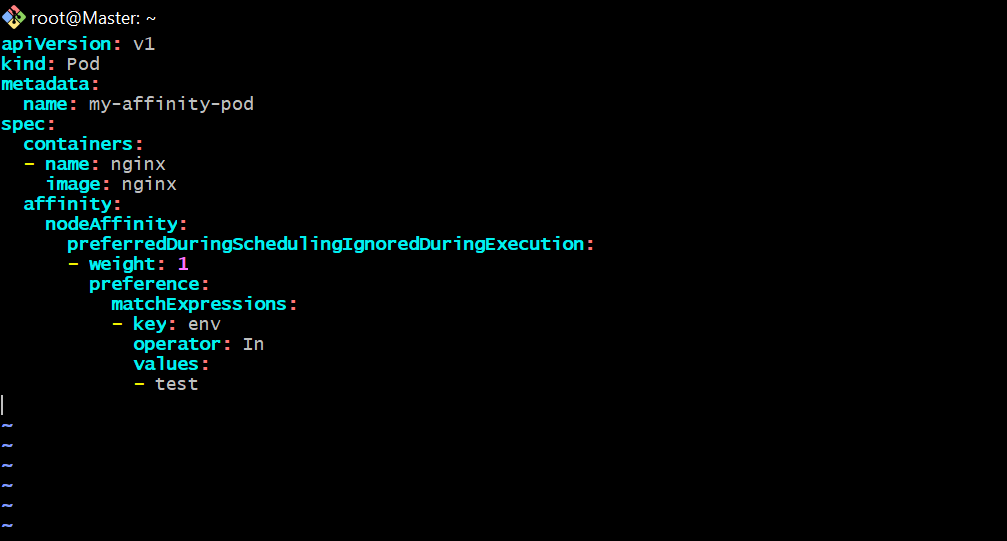
“kubectl get pods -o wide”

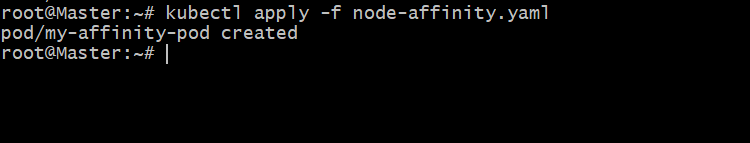


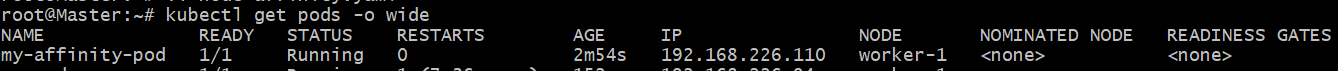
**Task 4: Use Node Affinity with Soft Scheduling Label a node with env=test. Create a pod with PreferredDuringSchedulingIgnoredDuringExecution node affinity, preferring to schedule it on a node labeled env=test. Remove the label and verify the pod continues to run.**



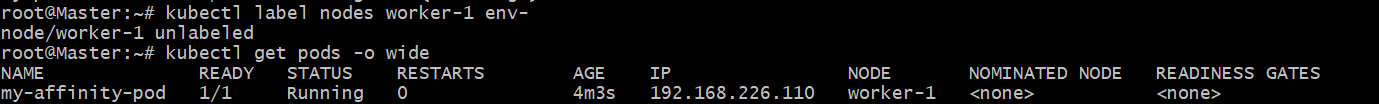
Created a node-affinity yaml file



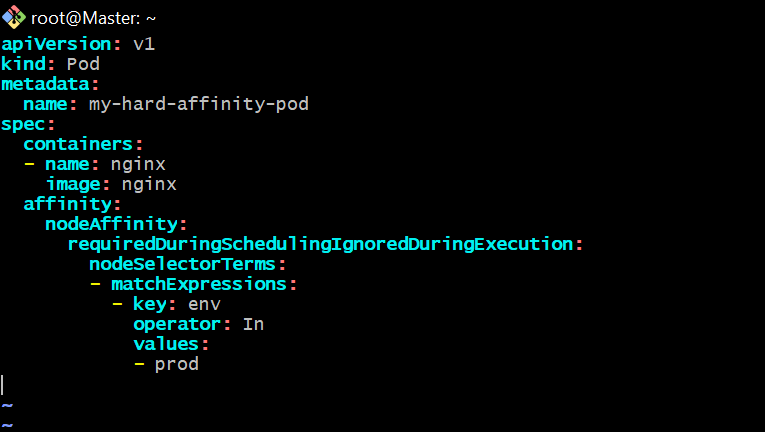


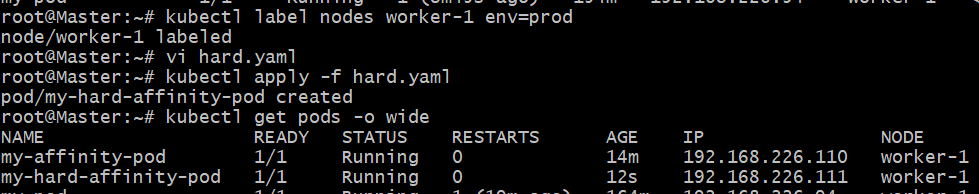


Remove the label and verify the pod continues to run.

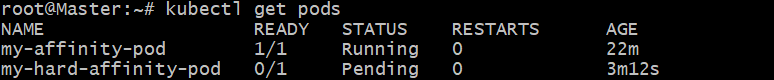


**Task 5: Implement Node Affinity with Hard Scheduling Create a pod with RequiredDuringSchedulingIgnoredDuringExecution node affinity, ensuring it will only be scheduled on a node labeled env=prod. Verify the pod cannot be scheduled if no node has the env=prod label.**





If no node has the env=prod label.



**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.**

toleration.yaml file

