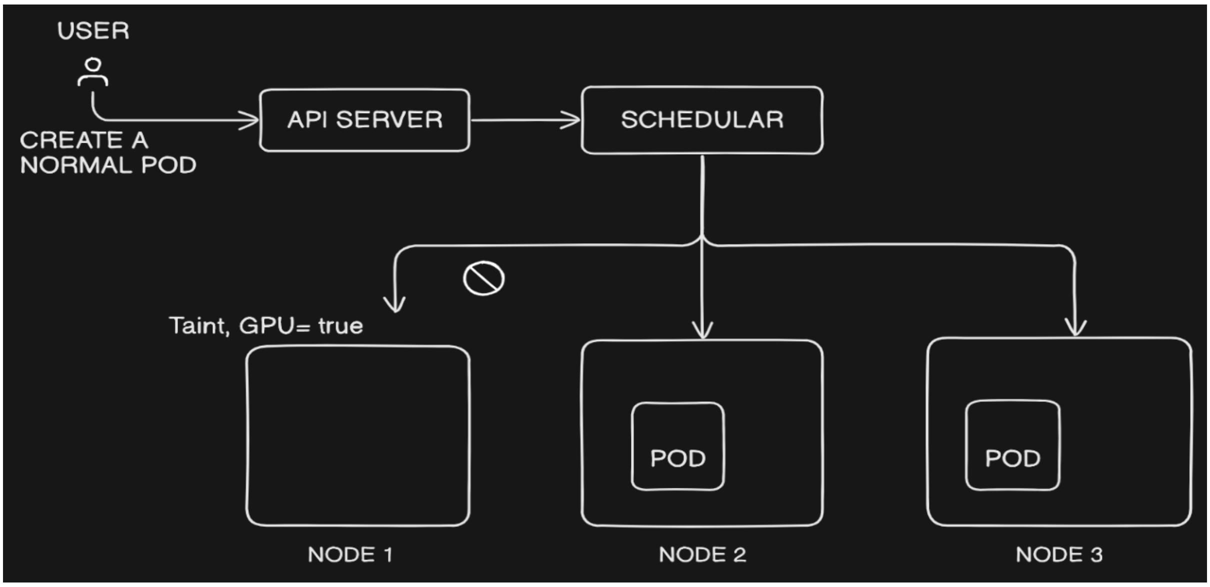
TAINTS AND TOLERATIONS IN K8S

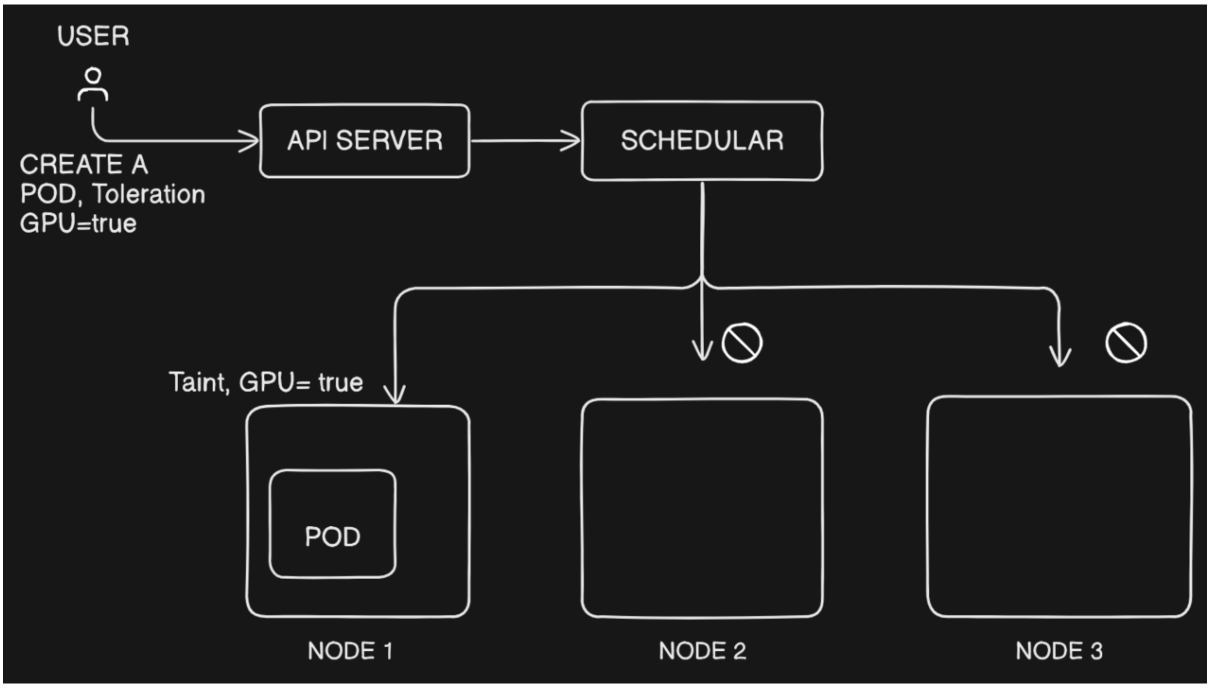
Taints and **Tolerations** work together to control which nodes can run specific pods. This mechanism helps ensure that certain pods are only scheduled on designated nodes, or conversely, that certain nodes repel specific pods.

Let's say we have 3 node, node 1 should contains the pod which only works on AI and remaining 2 nodes will be normal node.

Now we created a normal POD this POD won't create on node 1 because it is specified that node 1 should work only for AI related work, so POD get rejected and move to other nodes.



When POD is specified by toleration then it will check node 1 is accepting it or not. If it accepts it will be created on node 1.



Taint we do it on NODE, Toleration we do it on POD

TAINTS AND TOLERATIONS IN K8S

Taints consist of three parts:

1. **Key**: An identifier for the taint (e.g., key=value:effect).
2. **Value**: An optional value associated with the key.
3. **Effect**: Defines the action to be taken when a pod doesn't tolerate the taint.

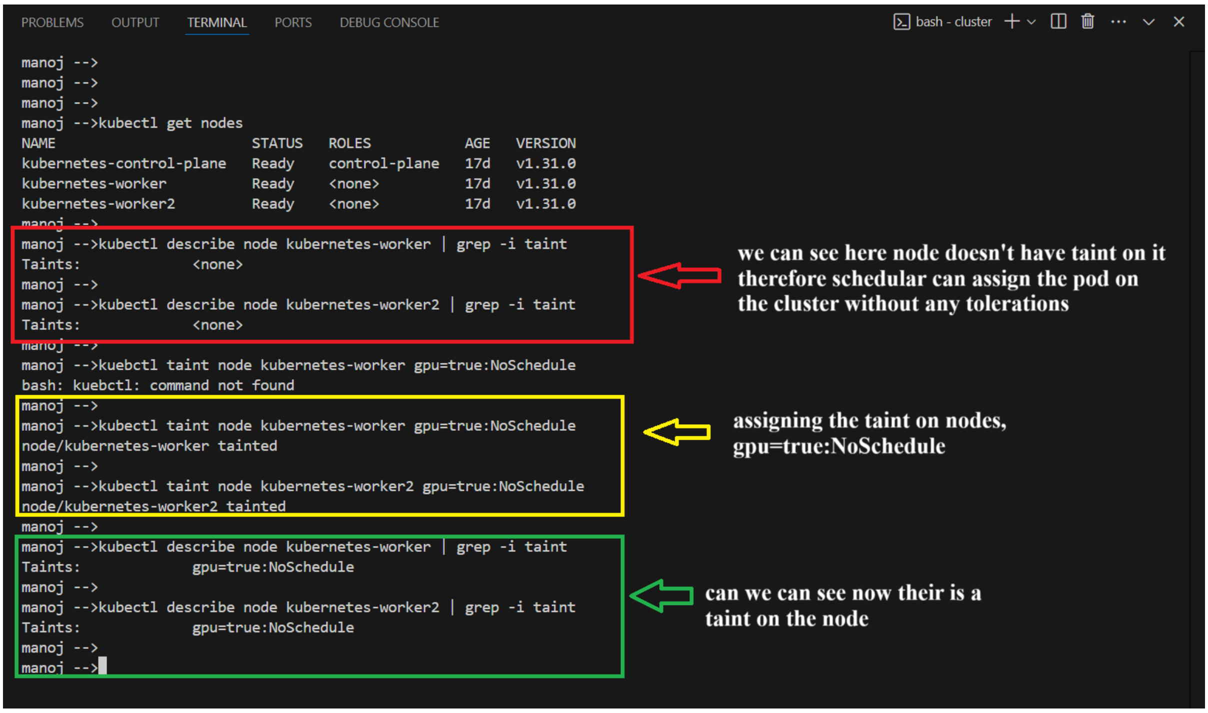
   ➔ **NoSchedule**: it will work on newer PODS

   ➔ **preferNOSchedule**: Kubernetes avoids scheduling pods that don't tolerate the taint, but it's not guaranteed.

   ➔ **noExecute**: on existing or newer PODS

why we use Taint?

- If we have specialized node for to run specific type of work load
- Control plane has taint on default because only control plane component and system components should be scheduled on control plane node that why It automatically added taint on the node

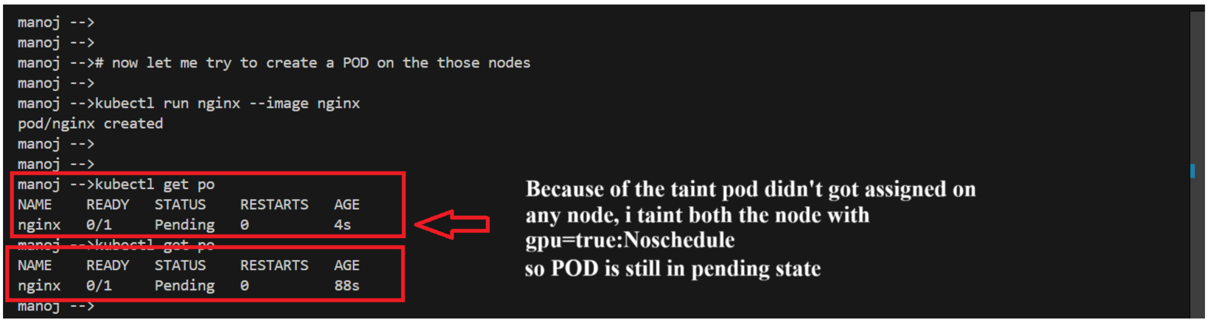


```
manoj -->
manoj -->
manoj -->
manoj -->kubectl get nodes
NAME                      STATUS   ROLES           AGE   VERSION
kubernetes-control-plane  Ready    control-plane   17d   v1.31.0
kubernetes-worker         Ready    <none>          17d   v1.31.0
kubernetes-worker2        Ready    <none>          17d   v1.31.0
manoj -->
manoj -->kubectl describe node kubernetes-worker | grep -i taint
Taints:             <none>
manoj -->
manoj -->kubectl describe node kubernetes-worker2 | grep -i taint
Taints:             <none>
manoj -->
manoj -->kuebctl taint node kubernetes-worker gpu=true:NoSchedule
bash: kuebctl: command not found
manoj -->
manoj -->kubectl taint node kubernetes-worker gpu=true:NoSchedule
node/kubernetes-worker tainted
manoj -->
manoj -->kubectl taint node kubernetes-worker2 gpu=true:NoSchedule
node/kubernetes-worker2 tainted
manoj -->
manoj -->kubectl describe node kubernetes-worker | grep -i taint
Taints:             gpu=true:NoSchedule
manoj -->
manoj -->kubectl describe node kubernetes-worker2 | grep -i taint
Taints:             gpu=true:NoSchedule
manoj -->
manoj -->
```

we can see here node doesn't have taint on it therefore schedular can assign the pod on the cluster without any tolerations

assigning the taint on nodes, gpu=true:NoSchedule

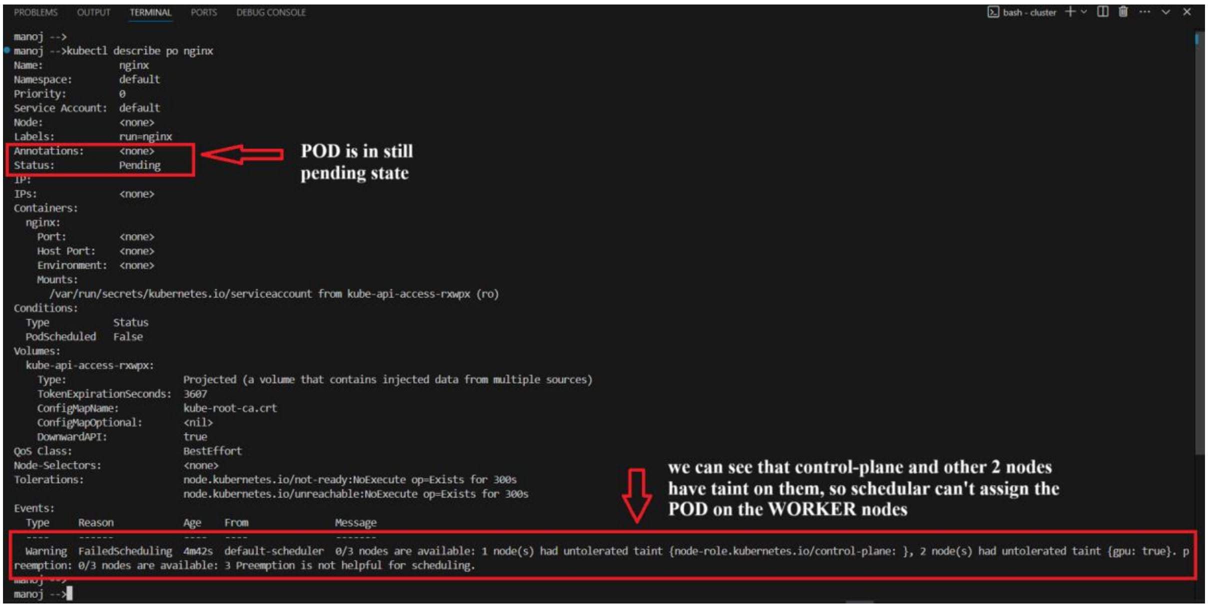
can we can see now their is a taint on the node



```
manoj -->
manoj -->
manoj --># now let me try to create a POD on the those nodes
manoj -->
manoj -->kubectl run nginx --image nginx
pod/nginx created
manoj -->
manoj -->
manoj -->kubectl get po
NAME    READY   STATUS    RESTARTS   AGE
nginx   0/1     Pending   0          4s
manoj -->kubectl get po
NAME    READY   STATUS    RESTARTS   AGE
nginx   0/1     Pending   0          88s
manoj -->
```

Because of the taint pod didn't got assigned on any node, i taint both the node with gpu=true:Noschedule so POD is still in pending state

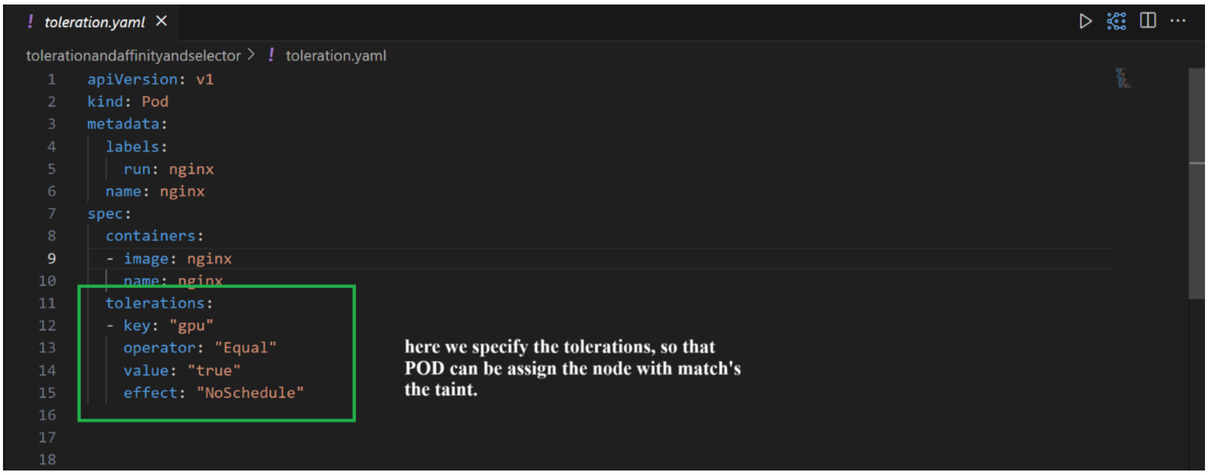
TAINTS AND TOLERATIONS IN K8S



```
manoj -->
manoj -->kubectl describe po nginx
Name:             nginx
Namespace:        default
Priority:         0
Service Account:  default
Node:             <none>
Labels:           run=nginx
Annotations:      <none>
Status:           Pending
IP:
IPs:              <none>
Containers:
  nginx:
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-rxwpx (ro)
Conditions:
  Type           Status
  PodScheduled   False
Volumes:
  kube-api-access-rxwpx:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 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
  ----     ------            ----   ----               -------
  Warning  FailedScheduling  4m42s  default-scheduler  0/3 nodes are available: 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane: }, 2 node(s) had untolerated taint {gpu: true}. p
reemption: 0/3 nodes are available: 3 Preemption is not helpful for scheduling.
manoj -->
manoj -->
```

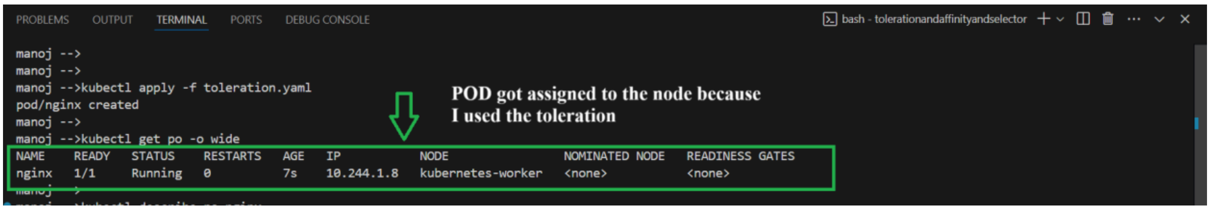
POD is in still pending state

we can see that control-plane and other 2 nodes have taint on them, so schedular can't assign the POD on the WORKER nodes



```yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx
  name: nginx
spec:
  containers:
  - image: nginx
    name: nginx
  tolerations:
  - key: "gpu"
    operator: "Equal"
    value: "true"
    effect: "NoSchedule"
```

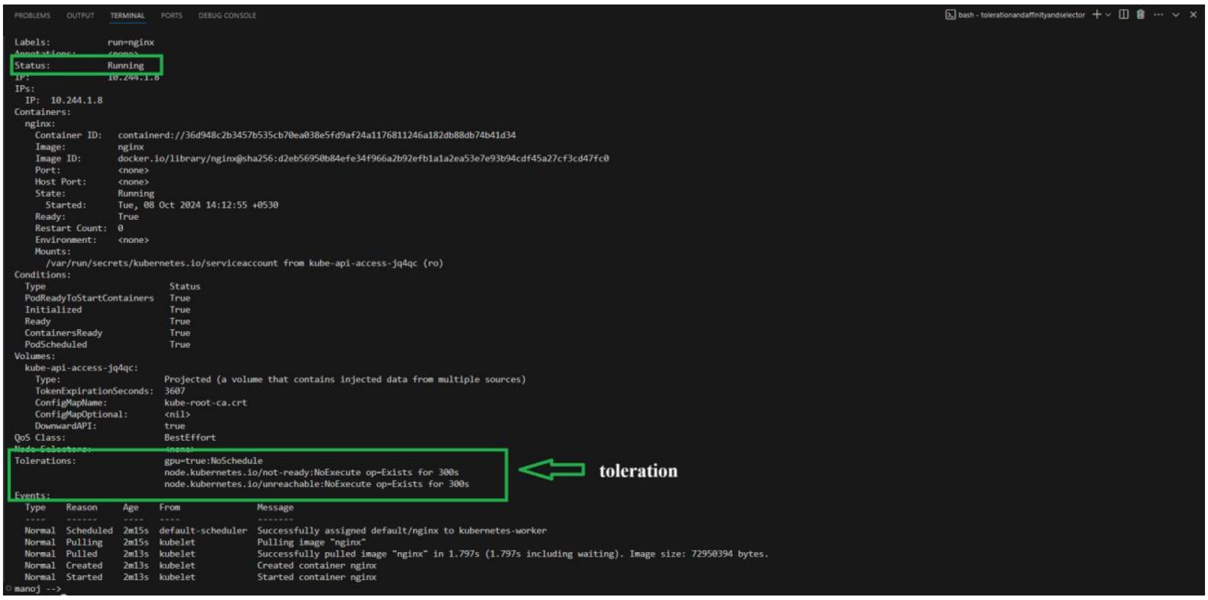
here we specify the tolerations, so that POD can be assign the node with match's the taint.



```
manoj -->
manoj -->
manoj -->kubectl apply -f toleration.yaml
pod/nginx created
manoj -->
manoj -->kubectl get po -o wide
NAME    READY   STATUS    RESTARTS   AGE   IP           NODE                NOMINATED NODE   READINESS GATES
nginx   1/1     Running   0          7s    10.244.1.8   kubernetes-worker   <none>           <none>
manoj -->
```

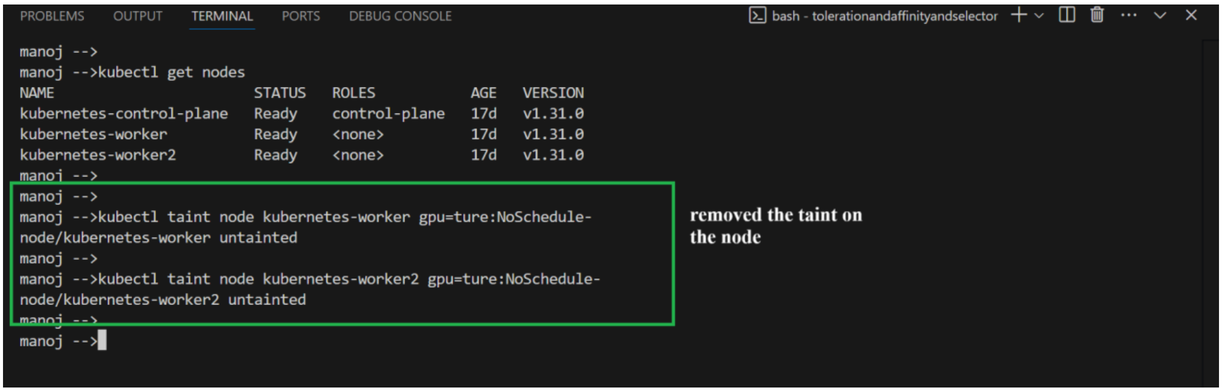
POD got assigned to the node because I used the toleration

TAINTS AND TOLERATIONS IN K8S



```
Labels:           run=nginx
Annotations:      <none>
Status:           Running
IP:               10.244.1.8
IPs:
  IP:  10.244.1.8
Containers:
  nginx:
    Container ID:   containerd://36d948c2b3457b535cb70ea038e5fd9af24a1176811246a182db88db74b41d34
    Image:          nginx
    Image ID:       docker.io/library/nginx@sha256:d2eb56950b84efe34f966a2b92efb1a1a2ea53e7e93b94cdf45a27cf3cd47fc0
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Tue, 08 Oct 2024 14:12:55 +0530
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-jq4qc (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-jq4qc:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 gpu-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  2m15s  default-scheduler  Successfully assigned default/nginx to kubernetes-worker
  Normal  Pulling    2m15s  kubelet            Pulling image "nginx"
  Normal  Pulled     2m13s  kubelet            Successfully pulled image "nginx" in 1.797s (1.797s including waiting). Image size: 72950394 bytes.
  Normal  Created    2m13s  kubelet            Created container nginx
  Normal  Started    2m13s  kubelet            Started container nginx
manoj -->
```

toleration



```
manoj -->
manoj -->kubectl get nodes
NAME                        STATUS   ROLES           AGE   VERSION
kubernetes-control-plane    Ready    control-plane   17d   v1.31.0
kubernetes-worker           Ready    <none>          17d   v1.31.0
kubernetes-worker2          Ready    <none>          17d   v1.31.0
manoj -->
manoj -->
manoj -->kubectl taint node kubernetes-worker gpu=ture:NoSchedule-
node/kubernetes-worker untainted
manoj -->
manoj -->kubectl taint node kubernetes-worker2 gpu=ture:NoSchedule-
node/kubernetes-worker2 untainted
manoj -->
manoj -->
```

removed the taint on the node