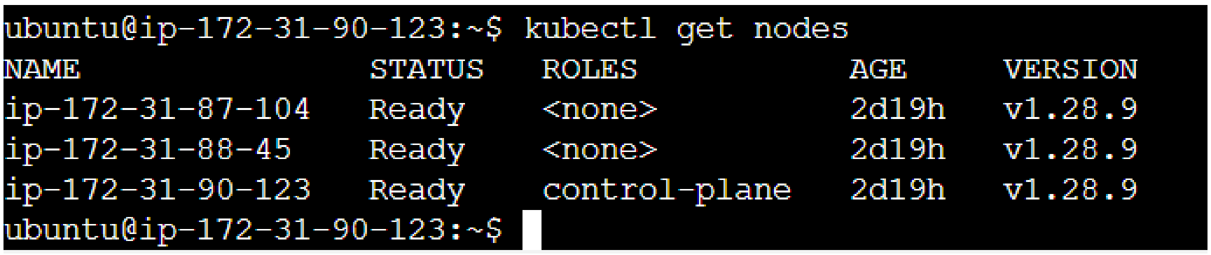
Creating a Taint on a node then a pod with tolerations but not to the taint

Operation 1: get the node name by running the command

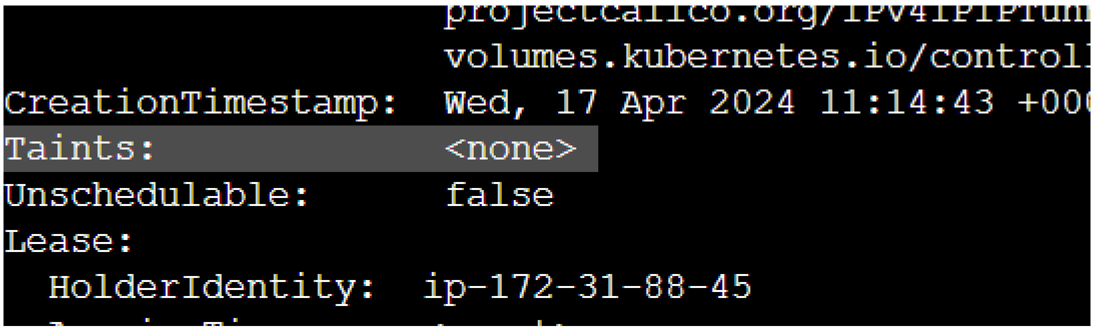
kubectl get nodes



```
ubuntu@ip-172-31-90-123:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE     VERSION
ip-172-31-87-104   Ready    <none>          2d19h   v1.28.9
ip-172-31-88-45    Ready    <none>          2d19h   v1.28.9
ip-172-31-90-123   Ready    control-plane   2d19h   v1.28.9
ubuntu@ip-172-31-90-123:~$
```

Let me show you that we don't have any taints

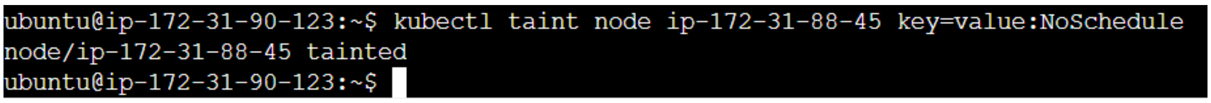
kubectl describe node <node name>



```
                    volumes.kubernetes.io/control
CreationTimestamp:  Wed, 17 Apr 2024 11:14:43 +00
Taints:             <none>
Unschedulable:      false
Lease:
  HolderIdentity:  ip-172-31-88-45
```

Operation 2: add a taint to a node using the following command

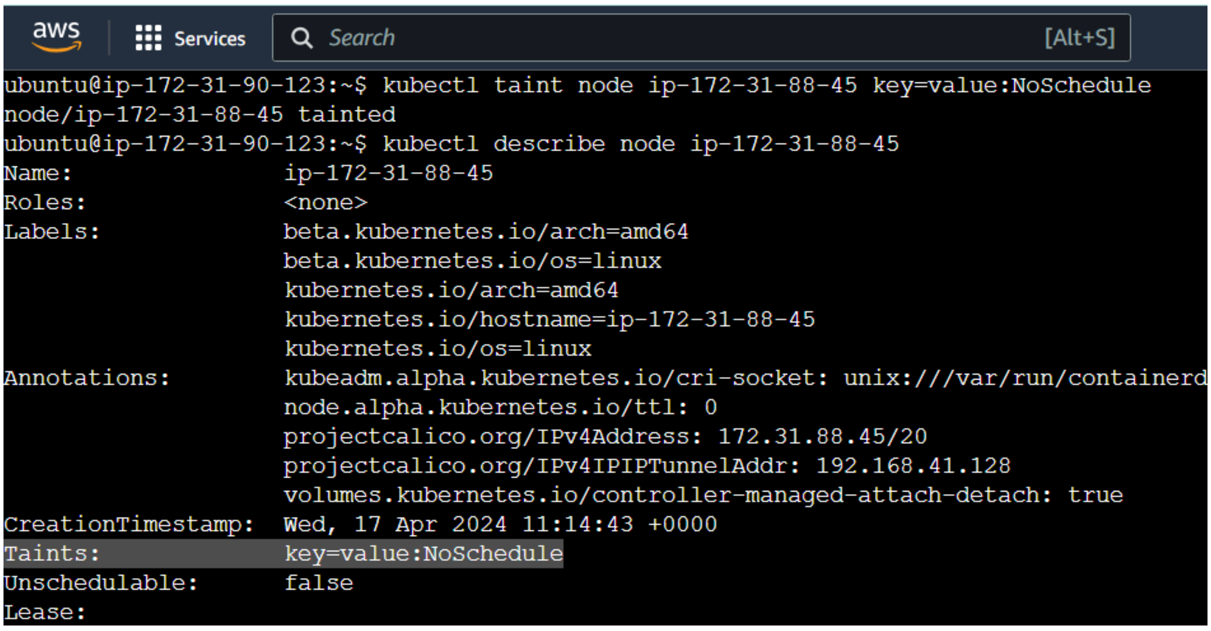
kubectl taint nodes <node name> key=value:NoSchedule



```
ubuntu@ip-172-31-90-123:~$ kubectl taint node ip-172-31-88-45 key=value:NoSchedule
node/ip-172-31-88-45 tainted
ubuntu@ip-172-31-90-123:~$
```

Now let's verify it

kubectl describe node <node name>



```
ubuntu@ip-172-31-90-123:~$ kubectl taint node ip-172-31-88-45 key=value:NoSchedule
node/ip-172-31-88-45 tainted
ubuntu@ip-172-31-90-123:~$ kubectl describe node ip-172-31-88-45
Name:               ip-172-31-88-45
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=ip-172-31-88-45
                    kubernetes.io/os=linux
Annotations:        kubeadm.alpha.kubernetes.io/cri-socket: unix:///var/run/containerd
                    node.alpha.kubernetes.io/ttl: 0
                    projectcalico.org/IPv4Address: 172.31.88.45/20
                    projectcalico.org/IPv4IPIPTunnelAddr: 192.168.41.128
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Wed, 17 Apr 2024 11:14:43 +0000
Taints:             key=value:NoSchedule
Unschedulable:      false
Lease:
```

So what taints does not allow is scheduling of new pods

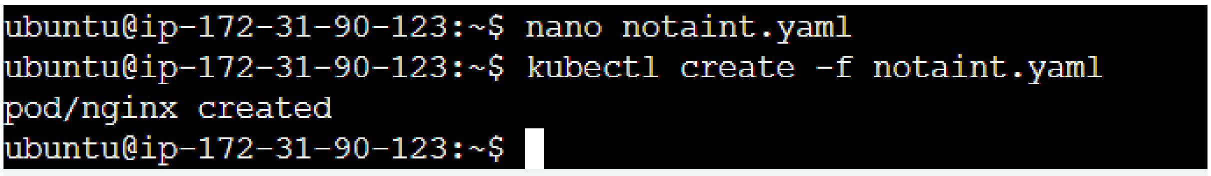
Operation 3: next thing to do is to create the yaml file.

First will create a pod which does not have tolerations

```
apiVersion: v1
kind: Pod
metadata:
name: nginx
labels:
env: test
spec:
containers:
- name: nginx
image: nginx
imagePullPolicy: IfNotPresent
```

To create pod use the below command

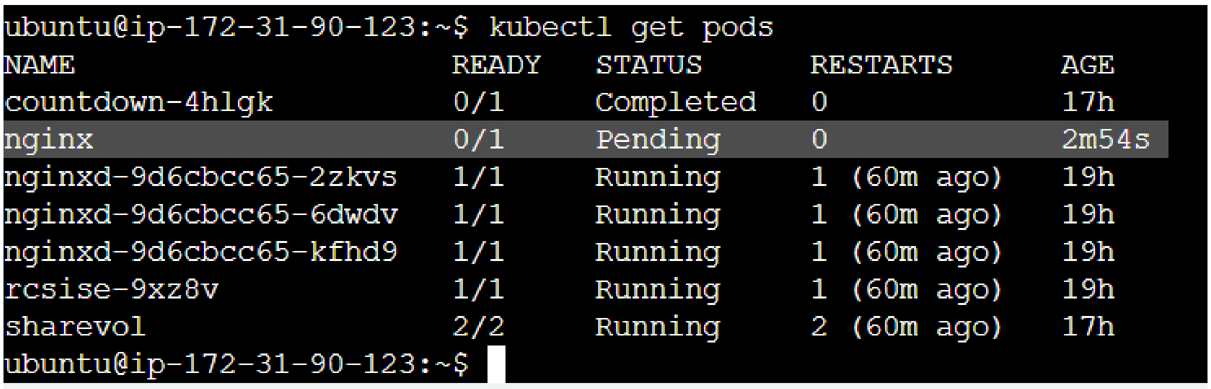
kubectl create -f <file_name>.yaml



```
ubuntu@ip-172-31-90-123:~$ nano notaint.yaml
ubuntu@ip-172-31-90-123:~$ kubectl create -f notaint.yaml
pod/nginx created
ubuntu@ip-172-31-90-123:~$
```

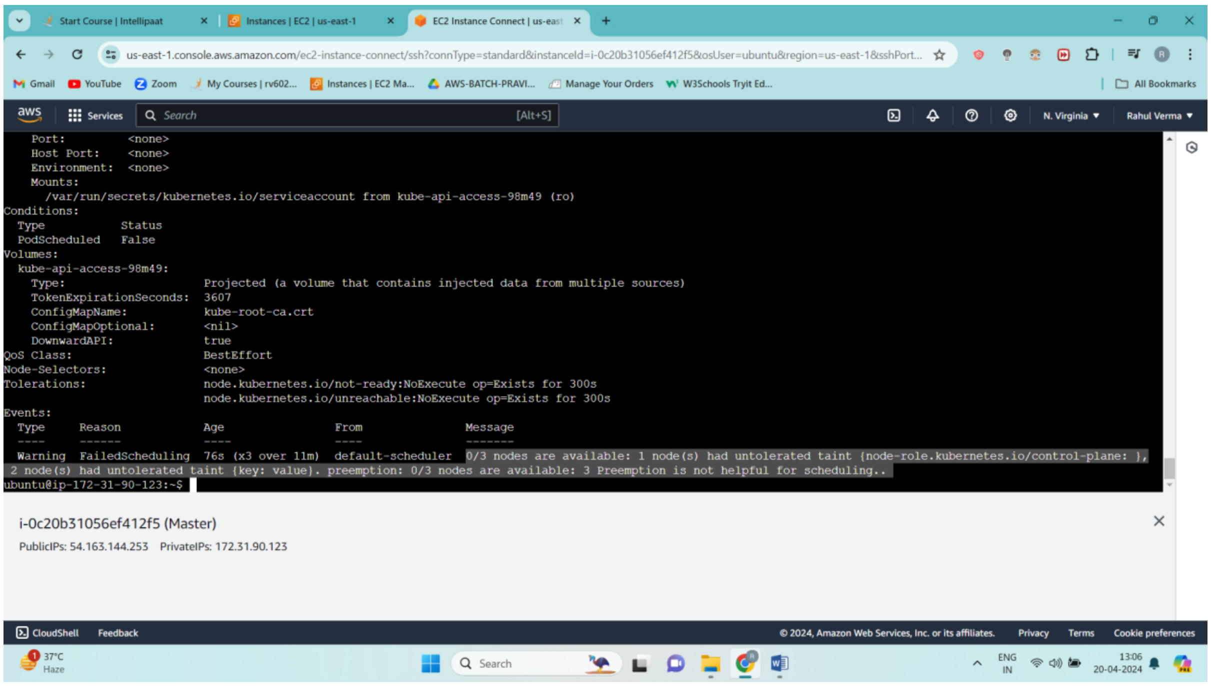
Now if we will check the pod status it should be in pending status

kubectl get pods



```
ubuntu@ip-172-31-90-123:~$ kubectl get pods
NAME                     READY   STATUS      RESTARTS       AGE
countdown-4hlgk          0/1     Completed   0              17h
nginx                    0/1     Pending     0              2m54s
nginxd-9d6cbcc65-2zkvs   1/1     Running     1 (60m ago)    19h
nginxd-9d6cbcc65-6dwdv   1/1     Running     1 (60m ago)    19h
nginxd-9d6cbcc65-kfhd9   1/1     Running     1 (60m ago)    19h
rcsise-9xz8v             1/1     Running     1 (60m ago)    19h
sharevol                 2/2     Running     2 (60m ago)    17h
ubuntu@ip-172-31-90-123:~$
```

So our taint is working properly



```
    Port:          <none>
    Host Port:     <none>
    Environment:   <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-98m49 (ro)
Conditions:
  Type           Status
  PodScheduled   False
Volumes:
  kube-api-access-98m49:
    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  76s (x3 over 11m)  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 {key: value}. preemption: 0/3 nodes are available: 3 Preemption is not helpful for scheduling..
ubuntu@ip-172-31-90-123:~$
```

i-0c20b31056ef412f5 (Master)

PublicIPs: 54.163.144.253 PrivateIPs: 172.31.90.123

Start Course | Intellipaat
Instances | EC2 | us-east-1
EC2 Instance Connect | us-east
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0c20b31056ef412f5&osUser=ubuntu&region=us-east-1&sshPort...
Gmail
YouTube
Zoom
My Courses | rv602...
Instances | EC2 Ma...
AWS-BATCH-PRAVI...
Manage Your Orders
W3Schools Tryit Ed...
All Bookmarks
Services
Search
[Alt+S]
N. Virginia
Rahul Verma
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
37°C
Haze
Search
ENG
IN
13:06
20-04-2024

Now let’s try to create a pod that uses tolleration

Operation 4: For that let’s create a yaml file

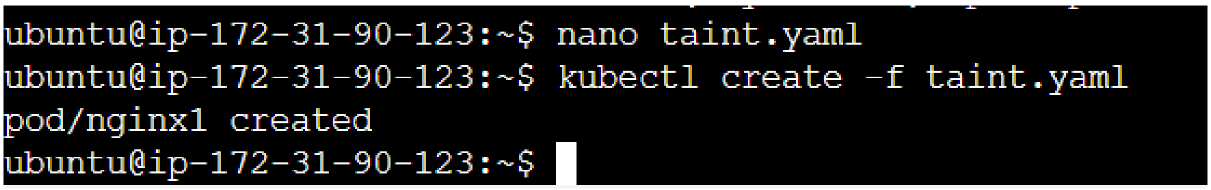
nano <file name>.yaml

copy the below code

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx1
  labels:
    env: test
spec:
  containers:
  - name: nginx1
    image: nginx
    imagePullPolicy: IfNotPresent
  tolerations:
  - key: "key"
    operator: "Equal"
    value: "value"
    effect: "NoSchedule"
```

And now create pod using that yaml file

kubectl create –f <file name>.yaml

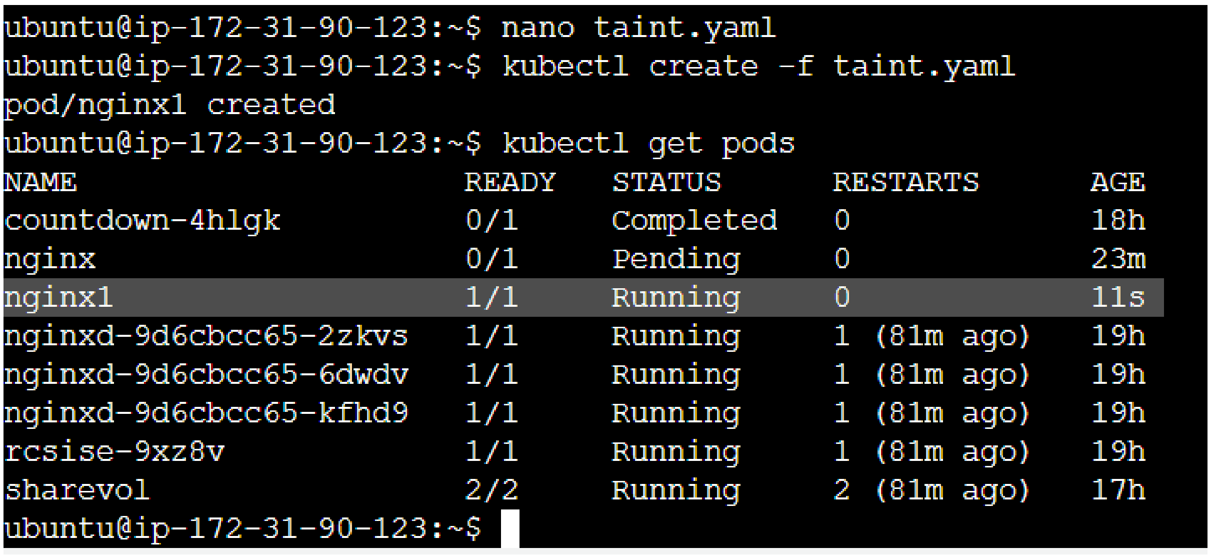


```
ubuntu@ip-172-31-90-123:~$ nano taint.yaml
ubuntu@ip-172-31-90-123:~$ kubectl create -f taint.yaml
pod/nginx1 created
ubuntu@ip-172-31-90-123:~$
```

Now this pod should be scheduled because it uses toleration

Let’s check

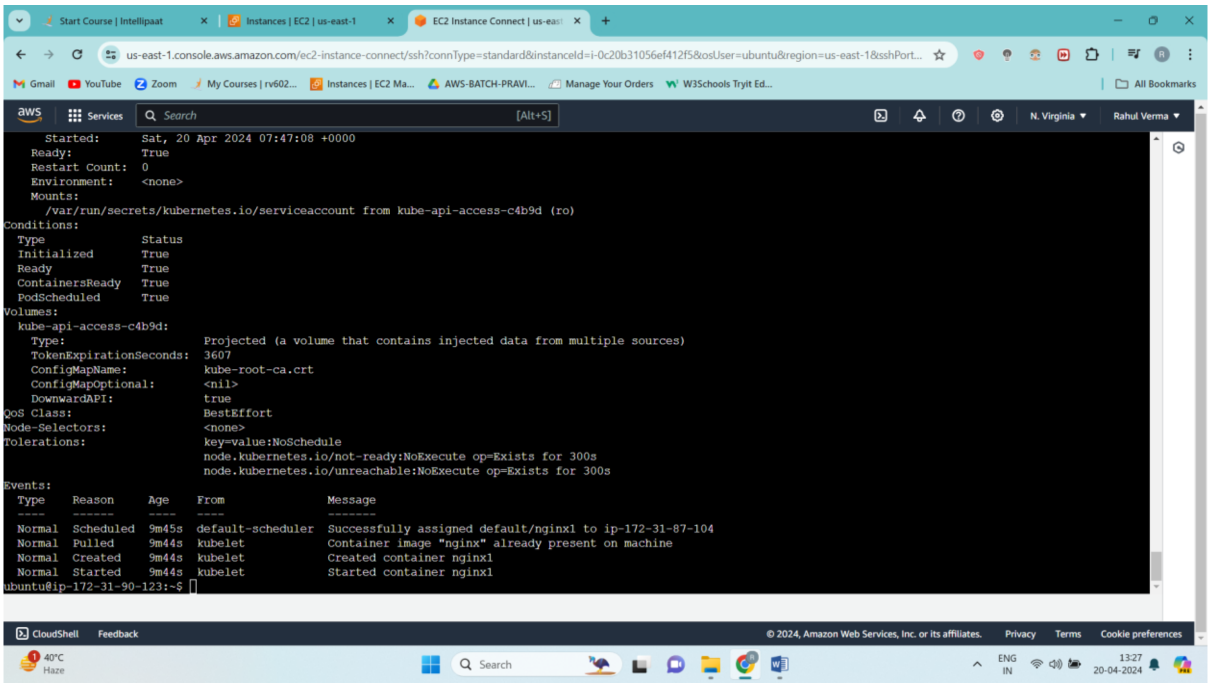
kubectl get pods



```
ubuntu@ip-172-31-90-123:~$ nano taint.yaml
ubuntu@ip-172-31-90-123:~$ kubectl create -f taint.yaml
pod/nginx1 created
ubuntu@ip-172-31-90-123:~$ kubectl get pods
NAME                     READY   STATUS      RESTARTS      AGE
countdown-4hlgk          0/1     Completed   0             18h
nginx                    0/1     Pending     0             23m
nginx1                   1/1     Running     0             11s
nginxd-9d6cbcc65-2zkvs   1/1     Running     1 (81m ago)   19h
nginxd-9d6cbcc65-6dwdv   1/1     Running     1 (81m ago)   19h
nginxd-9d6cbcc65-kfhd9   1/1     Running     1 (81m ago)   19h
rcsise-9xz8v             1/1     Running     1 (81m ago)   19h
sharevol                 2/2     Running     2 (81m ago)   17h
ubuntu@ip-172-31-90-123:~$
```

And if we do

kubectl describe nginx1



```
      Started:        Sat, 20 Apr 2024 07:47:08 +0000
    Ready:            True
    Restart Count:    0
    Environment:      <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-c4b9d (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  kube-api-access-c4b9d:
    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:                 key=value: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  9m45s  default-scheduler  Successfully assigned default/nginx1 to ip-172-31-87-104
  Normal  Pulled     9m44s  kubelet            Container image "nginx" already present on machine
  Normal  Created    9m44s  kubelet            Created container nginx1
  Normal  Started    9m44s  kubelet            Started container nginx1
ubuntu@ip-172-31-90-123:~$
```