To create Deployment file, node affinity files please use https://kubernetes.io/docs.

Q1) Create a new deployment named test-deploy with the nginx image and 6 replicas.

Ans:

```
kubectl create deployment test-deploy --image=nginx
kubectl scale deployment test-deploy --replicas=6
```

Q2) Apply a label color=blue to node worker.

Ans:

```
kubectl label node worker color=blue
```

Q3) Set **Node Affinity** to the deployment test-deploy to place the pods on worker node only.

Ans:

```
vim blue.yaml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: blue
spec:
  replicas: 6
  selector:
    matchLabels:
      run: nginx
  template:
    metadata:
      labels:
        run: nginx
    spec:
      containers:
      - image: nginx
        name: nginx
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: color
                operator: In
```

```
            values:
            - blue
```

```
kubectl create -f blue.yaml
```

Q4) Remove taint from the master node and verify node is untaint.

Ans:

```
kubectl taint node master node-role.kubernetes.io/master-
kubectl describe nodes | egrep "Name:|Taints:"
```

Q5) Create a create pod with node affinity rule.

Ans:

```
vim nginx-deploy.yaml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: myKey
                operator: In
                values:
                - label1
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
```

```
kubectl create -f nginx-deploy.yaml
```

```
root@master:/home/ubuntu# vim nginx-deploy.yaml
root@master:/home/ubuntu# kubectl create -f nginx-deploy.yaml
deployment.apps/nginx-deployment created
```

Q6) check pod status if it's pending check error.

Ans:

```
kubectl get pod -o wide        // pod Should be in pending state
check error by kubectl describe pod <pod error> and check for error
```

```
NAME                               READY   STATUS    RESTARTS   AGE     IP          NODE     NOMINATED NODE   READINESS
GATES
nginx                              1/1     Running   2          3d      10.44.0.1   worker   <none>           <none>
nginx-deployment-df7f96746-pm9xk   0/1     Pending   0          2m48s   <none>      <none>   <none>           <none>
nginx-deployment-df7f96746-qxbpt   0/1     Pending   0          2m48s   <none>      <none>   <none>           <none>
root@master:/home/ubuntu#
```

```
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-r6p8j (ro)
Conditions:
  Type           Status
  PodScheduled   False
Volumes:
  default-token-r6p8j:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-r6p8j
    Optional:    false
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  7s (x8 over 3m34s)  default-scheduler  0/2 nodes are available: 2 node(s) didn't match node
 selector.
root@master:/home/ubuntu#
```

Q7) create same label on master node.

Ans:

```
kubectl label nodes master myKey=label1
```

Q8) check status of pods again

Ans:

```
kubectl get pods -o wide
```

Q9) Add **vip=true** on worker node.

Ans:

```
kubectl label nodes worker vip=true
```

Q10) Run PODs with a Node Selector on worker node with image **busybox** and name **vip1**.

Ans:

```
vim nodeselector.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: vip1
spec:
  containers:
  - name: sleep
    image: busybox
  nodeSelector:
    vip: "true"
```

```
kubectl get pod -o wide
```