Lesson 05 Demo 01

Configuring Pods with Nodename and Nodeselector Fields

Objective: To configure pods with nodename and nodeselector fields for efficient resource use, compliance, and specific application needs in a cluster

Tools required: kubeadm, kubectl, kubelet, and containerd

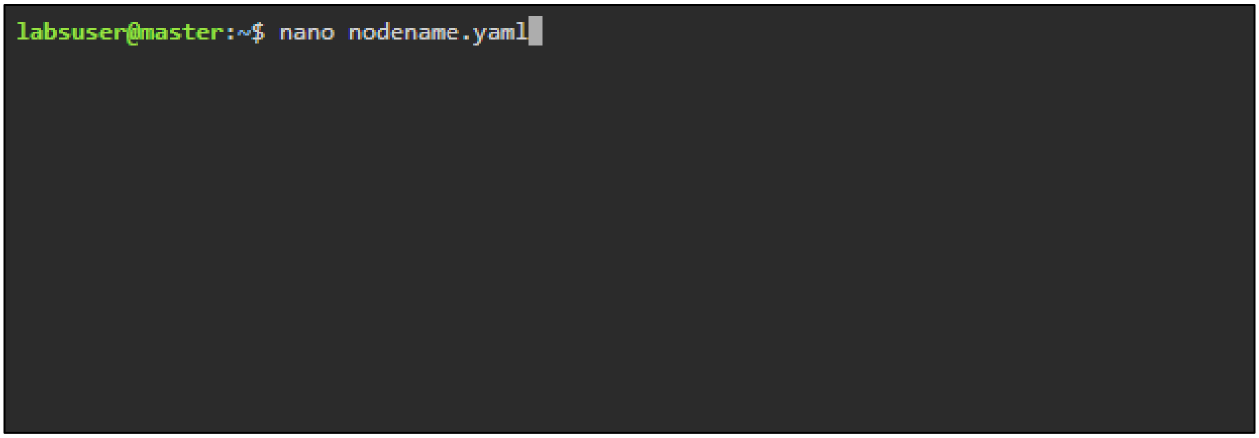
Prerequisites: A Kubernetes cluster should already be set up (refer to the steps provided in Lesson 02, Demo 01 for guidance).

Steps to be followed:

1. Create pods with the fields nodename and nodeselector
2. Assign labels to the nodes
3. Create a pod with the NotIn operator

Step 1: Create pods with the fields nodename and nodeselector

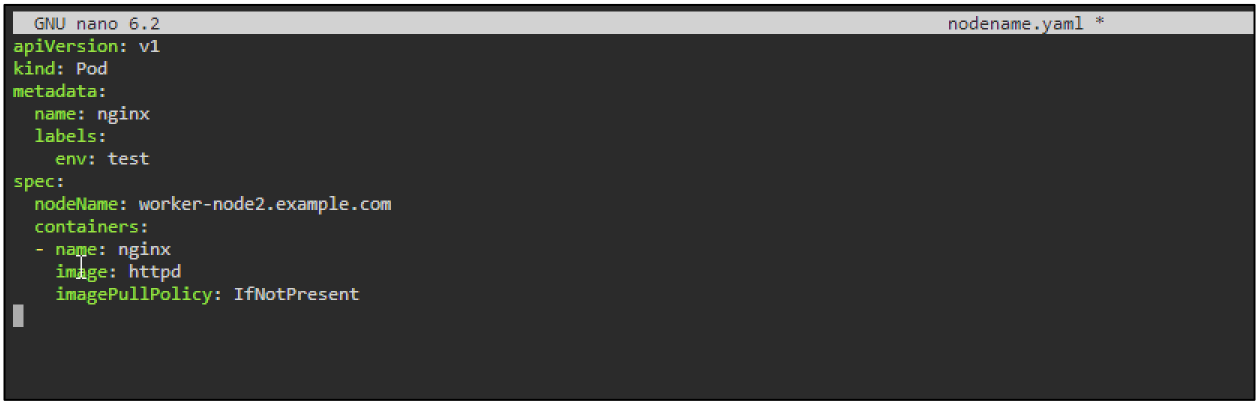
1.1 On the master node, create a YAML file using the following command:
nano nodename.yaml



```
labsuser@master:~$ nano nodename.yaml
```

1.2 Copy the following code in the YAML file:

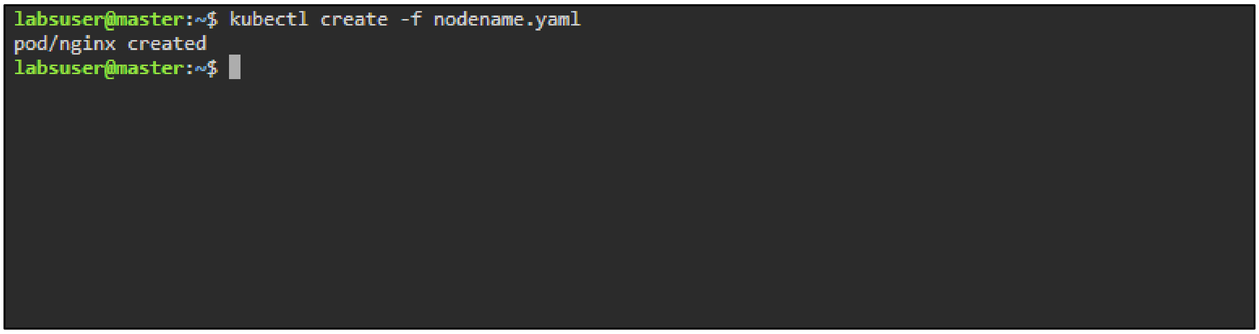
```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    env: test
spec:
  nodeName: worker-node2.example.com
  containers:
  - name: nginx
    image: httpd
    imagePullPolicy: IfNotPresent
```



```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    env: test
spec:
  nodeName: worker-node2.example.com
  containers:
  - name: nginx
    image: httpd
    imagePullPolicy: IfNotPresent
```

1.3 Create a pod by executing the following command:

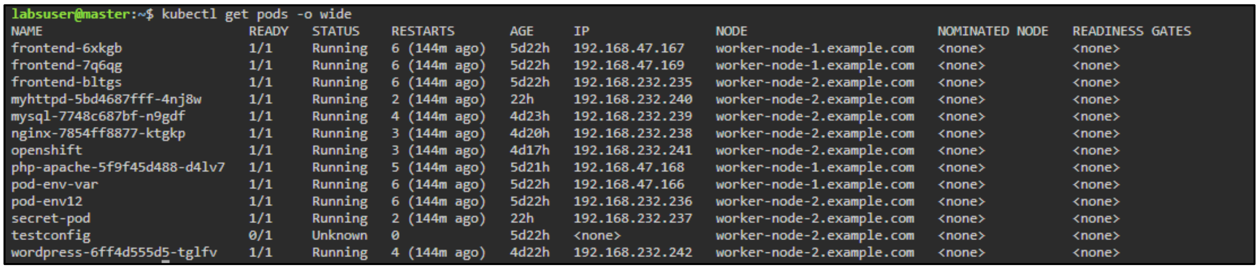
kubectl create -f nodename.yaml



```
labsuser@master:~$ kubectl create -f nodename.yaml
pod/nginx created
labsuser@master:~$
```

1.4 Verify the pod state by executing the following command:

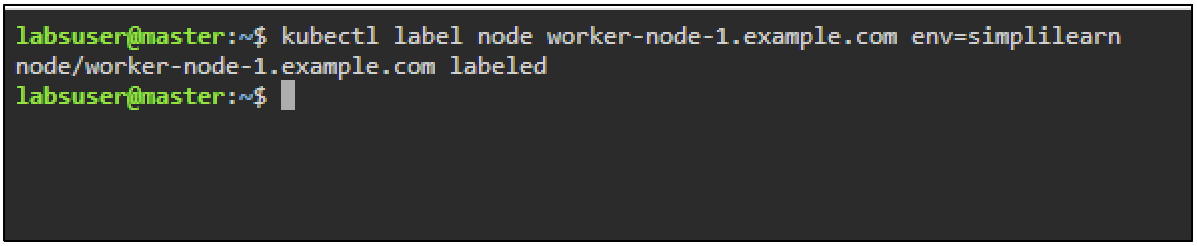
kubectl get pods -o wide



```
labsuser@master:~$ kubectl get pods -o wide
NAME                              READY   STATUS    RESTARTS       AGE     IP                NODE                         NOMINATED NODE   READINESS GATES
frontend-6xkgb                    1/1     Running   6 (144m ago)   5d22h   192.168.47.167    worker-node-1.example.com   <none>           <none>
frontend-7q6qg                    1/1     Running   6 (144m ago)   5d22h   192.168.47.169    worker-node-1.example.com   <none>           <none>
frontend-bltgs                    1/1     Running   6 (144m ago)   5d22h   192.168.232.235   worker-node-2.example.com   <none>           <none>
myhttpd-5bd4687fff-4nj8w          1/1     Running   2 (144m ago)   22h     192.168.232.240   worker-node-2.example.com   <none>           <none>
mysql-7748c687bf-n9gdf            1/1     Running   4 (144m ago)   4d23h   192.168.232.239   worker-node-2.example.com   <none>           <none>
nginx-7854ff8877-ktgkp            1/1     Running   3 (144m ago)   4d20h   192.168.232.238   worker-node-2.example.com   <none>           <none>
openshift                         1/1     Running   3 (144m ago)   4d17h   192.168.232.241   worker-node-2.example.com   <none>           <none>
php-apache-5f9f45d488-d4lv7       1/1     Running   5 (144m ago)   5d21h   192.168.47.168    worker-node-1.example.com   <none>           <none>
pod-env-var                       1/1     Running   6 (144m ago)   5d22h   192.168.47.166    worker-node-1.example.com   <none>           <none>
pod-env12                         1/1     Running   6 (144m ago)   5d22h   192.168.232.236   worker-node-2.example.com   <none>           <none>
secret-pod                        1/1     Running   2 (144m ago)   22h     192.168.232.237   worker-node-2.example.com   <none>           <none>
testconfig                        0/1     Unknown   0              5d22h   <none>            worker-node-2.example.com   <none>           <none>
wordpress-6ff4d555d5-tglfv        1/1     Running   4 (144m ago)   4d22h   192.168.232.242   worker-node-2.example.com   <none>           <none>
```

1.5 Run the following command to label the node:

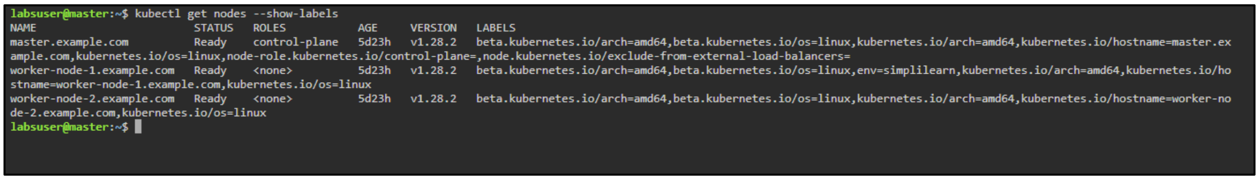
kubectl label node worker-node-1.example.com env=simplilearn



```
labsuser@master:~$ kubectl label node worker-node-1.example.com env=simplilearn
node/worker-node-1.example.com labeled
labsuser@master:~$
```

1.6 Verify the label by entering the following command:

kubectl get nodes –show labels

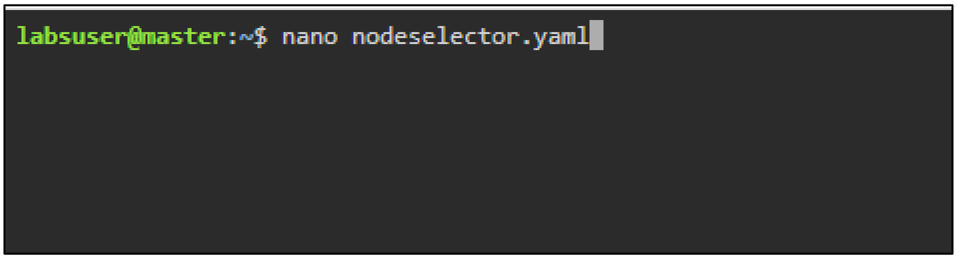


```
labsuser@master:~$ kubectl get nodes --show-labels
NAME                        STATUS   ROLES           AGE     VERSION   LABELS
master.example.com          Ready    control-plane   5d23h   v1.28.2   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=master.ex
ample.com,kubernetes.io/os=linux,node-role.kubernetes.io/control-plane=,node.kubernetes.io/exclude-from-external-load-balancers=
worker-node-1.example.com   Ready    <none>          5d23h   v1.28.2   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,env=simplilearn,kubernetes.io/arch=amd64,kubernetes.io/ho
stname=worker-node-1.example.com,kubernetes.io/os=linux
worker-node-2.example.com   Ready    <none>          5d23h   v1.28.2   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=worker-no
de-2.example.com,kubernetes.io/os=linux
labsuser@master:~$
```

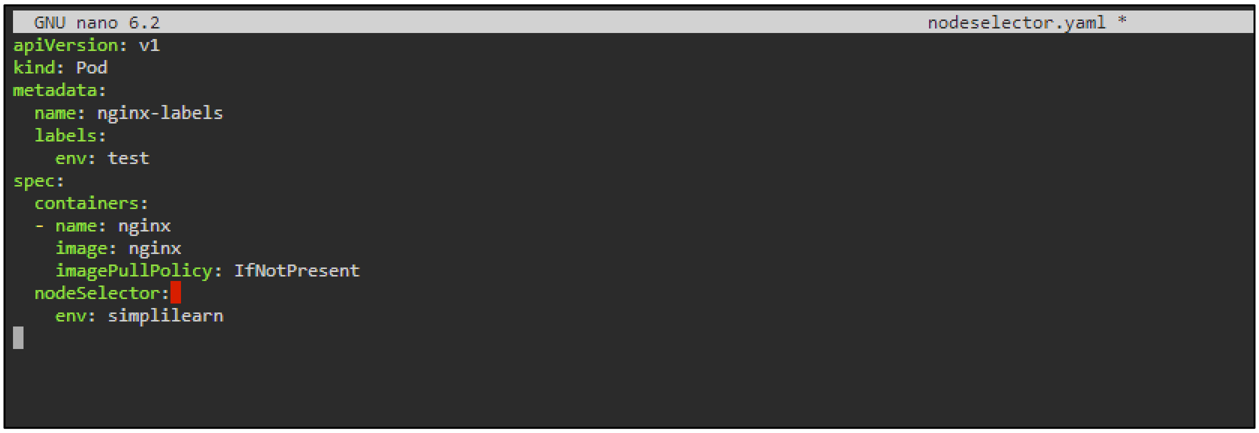
1.7 Enter the command **nano nodeselector.yaml** to create a YAML file and copy the following code in that file:

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

nodeSelector:
env: simplilearn

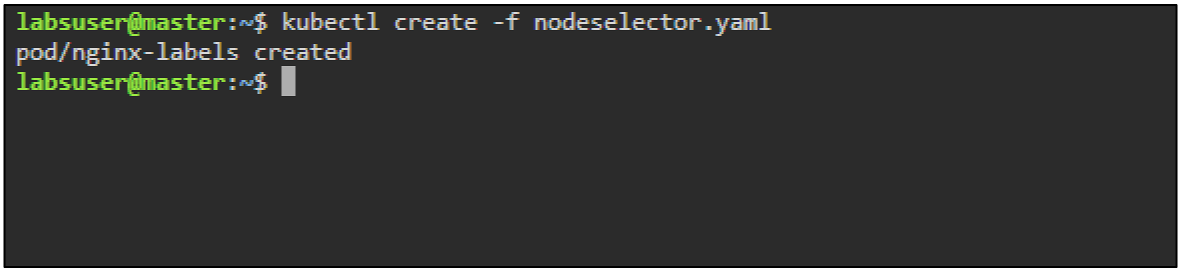


```
labsuser@master:~$ nano nodeselector.yaml
```



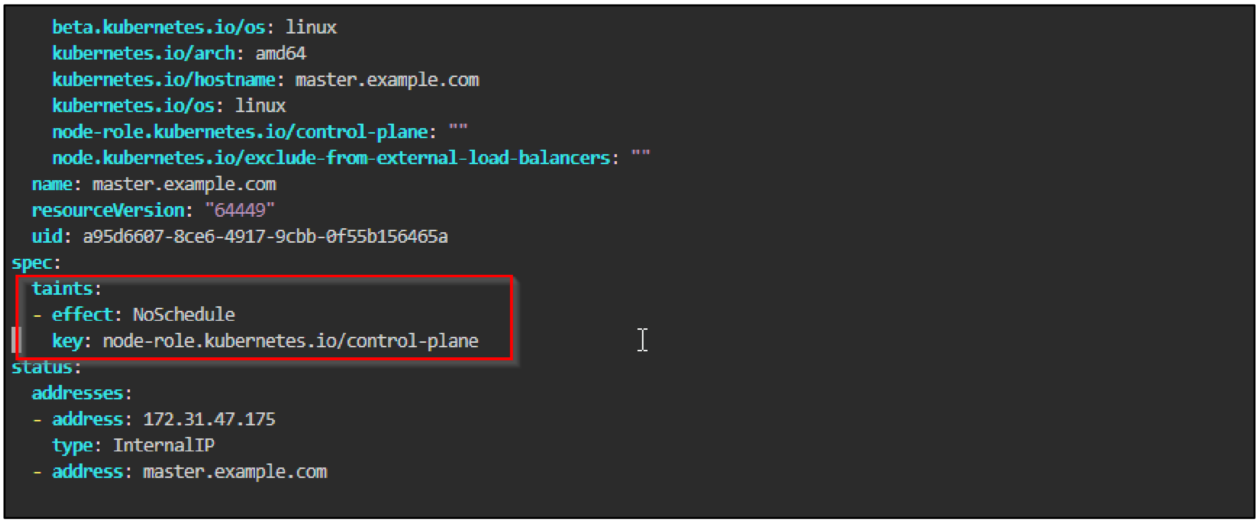
```
  GNU nano 6.2                                                                  nodeselector.yaml *
apiVersion: v1
kind: Pod
metadata:
  name: nginx-labels
  labels:
    env: test
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  nodeSelector:
    env: simplilearn
```

1.8 Run the following command to create a pod:
kubectl create -f nodeselector.yaml

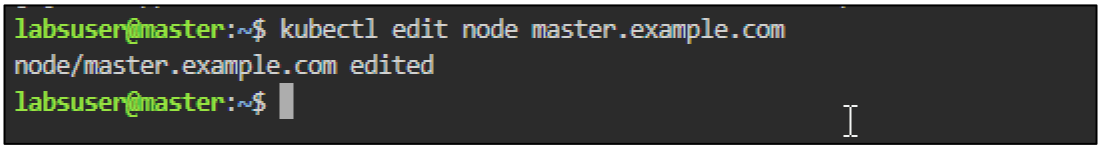


```
labsuser@master:~$ kubectl create -f nodeselector.yaml
pod/nginx-labels created
labsuser@master:~$
```

1.9 Remove the **taints** field from **master.example.com** by executing the following command:
kubectl edit node master.example.com



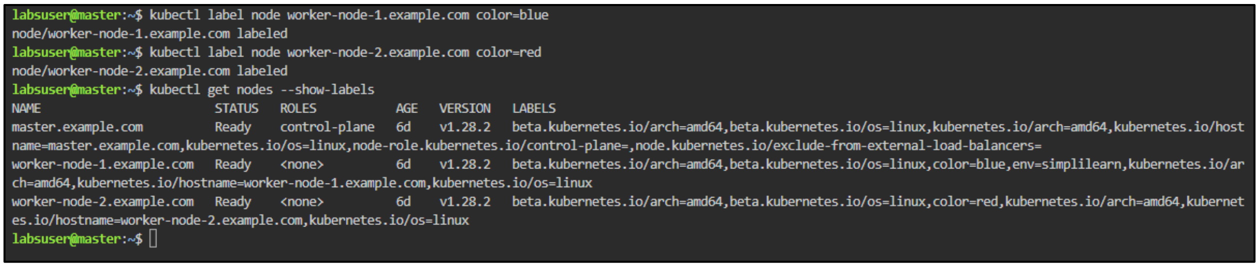
```
    beta.kubernetes.io/os: linux
    kubernetes.io/arch: amd64
    kubernetes.io/hostname: master.example.com
    kubernetes.io/os: linux
    node-role.kubernetes.io/control-plane: ""
    node.kubernetes.io/exclude-from-external-load-balancers: ""
  name: master.example.com
  resourceVersion: "64449"
  uid: a95d6607-8ce6-4917-9cbb-0f55b156465a
spec:
  taints:
  - effect: NoSchedule
    key: node-role.kubernetes.io/control-plane
status:
  addresses:
  - address: 172.31.47.175
    type: InternalIP
  - address: master.example.com
```



```
labsuser@master:~$ kubectl edit node master.example.com
node/master.example.com edited
labsuser@master:~$ 
```

Step 2: Assign labels to the nodes

2.1 Run the following commands to assign labels to worker 1 and worker 2 nodes for pod assignment:
kubectl label node worker-node-1.example.com color=blue
kubectl label node worker-node-2.example.com color=red
kubectl get nodes --show-labels

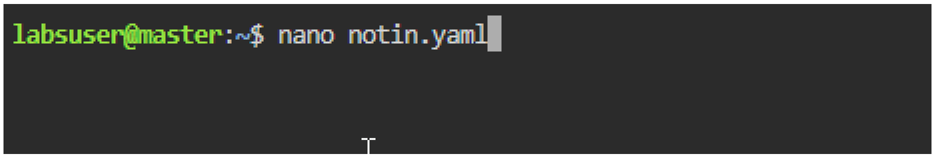


```
labsuser@master:~$ kubectl label node worker-node-1.example.com color=blue
node/worker-node-1.example.com labeled
labsuser@master:~$ kubectl label node worker-node-2.example.com color=red
node/worker-node-2.example.com labeled
labsuser@master:~$ kubectl get nodes --show-labels
NAME                        STATUS   ROLES           AGE   VERSION   LABELS
master.example.com          Ready    control-plane   6d    v1.28.2   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/host
name=master.example.com,kubernetes.io/os=linux,node-role.kubernetes.io/control-plane=,node.kubernetes.io/exclude-from-external-load-balancers=
worker-node-1.example.com   Ready    <none>          6d    v1.28.2   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,color=blue,env=simplilearn,kubernetes.io/ar
ch=amd64,kubernetes.io/hostname=worker-node-1.example.com,kubernetes.io/os=linux
worker-node-2.example.com   Ready    <none>          6d    v1.28.2   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,color=red,kubernetes.io/arch=amd64,kubernet
es.io/hostname=worker-node-2.example.com,kubernetes.io/os=linux
labsuser@master:~$ 
```

Step 3: Create a pod with the NotIn operator

3.1 Run the following command to create a YAML file:

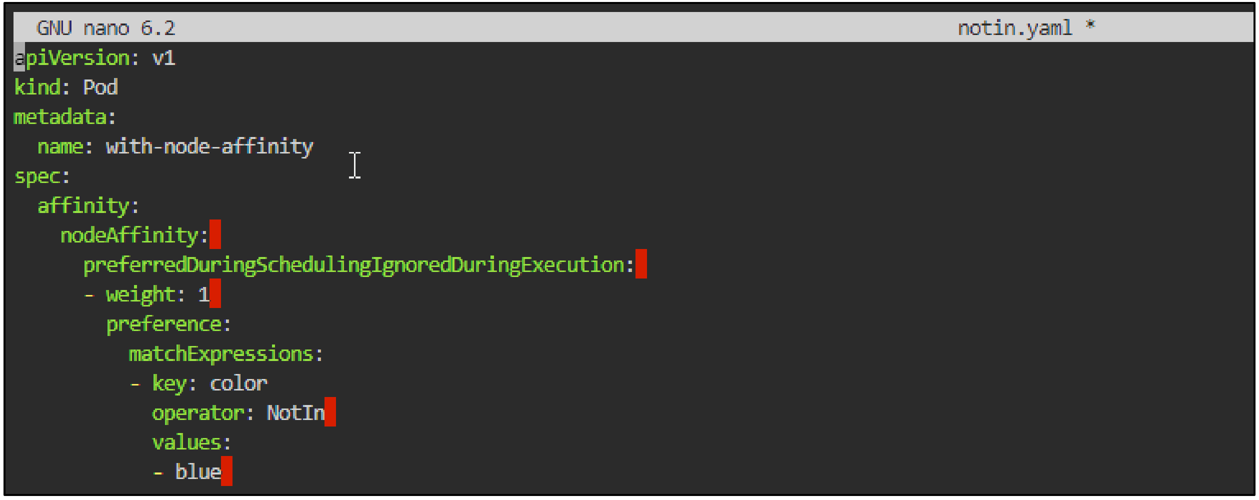
nano notin.yaml



```
labsuser@master:~$ nano notin.yaml
```

3.2 Copy the following code in the YAML file:

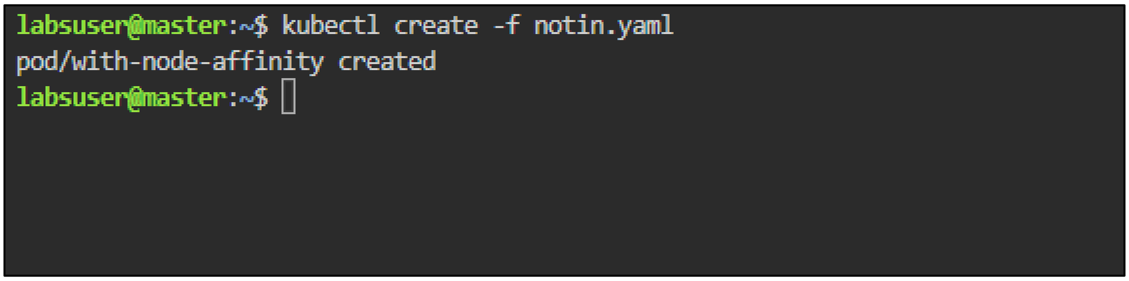
```
apiVersion: v1
kind: Pod
metadata:
name: with-node-affinity
spec:
affinity:
nodeAffinity:
preferredDuringSchedulingIgnoredDuringExecution:
- weight: 1
preference:
matchExpressions:
- key: color
operator: NotIn
values:
- blue
containers:
- name: httpd
image: docker.io/httpd
```



```
  GNU nano 6.2                                                              notin.yaml *
apiVersion: v1
kind: Pod
metadata:
  name: with-node-affinity
spec:
  affinity:
    nodeAffinity:
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 1
        preference:
          matchExpressions:
          - key: color
            operator: NotIn
            values:
            - blue
```

3.3 Run the following command to create a pod:

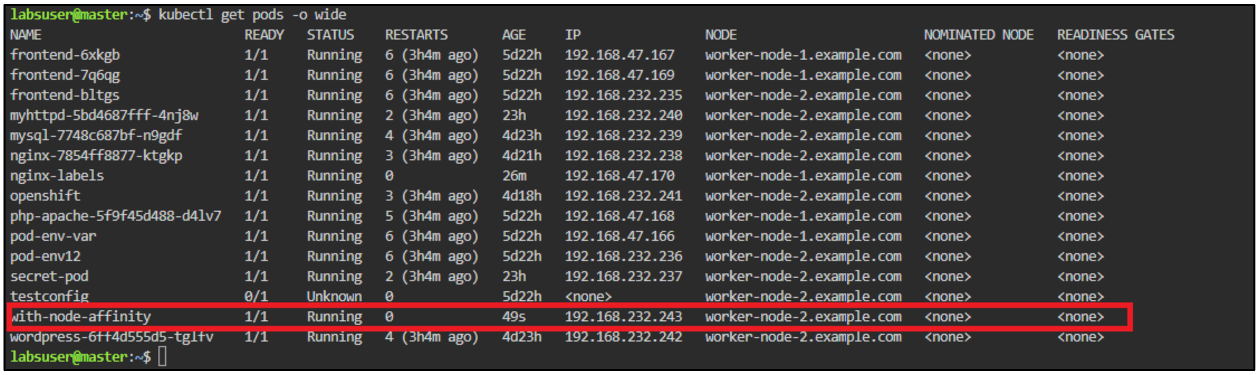
kubectl create -f notin.yaml



```
labsuser@master:~$ kubectl create -f notin.yaml
pod/with-node-affinity created
labsuser@master:~$ 
```

3.4 Verify the pod state by running the following command:

kubectl get pods -o wide



```
labsuser@master:~$ kubectl get pods -o wide
NAME                           READY   STATUS    RESTARTS       AGE     IP                NODE                       NOMINATED NODE   READINESS GATES
frontend-6xkgb                 1/1     Running   6 (3h4m ago)   5d22h   192.168.47.167    worker-node-1.example.com  <none>           <none>
frontend-7q6qg                 1/1     Running   6 (3h4m ago)   5d22h   192.168.47.169    worker-node-1.example.com  <none>           <none>
frontend-bltgs                 1/1     Running   6 (3h4m ago)   5d22h   192.168.232.235   worker-node-2.example.com  <none>           <none>
myhttpd-5bd4687fff-4nj8w       1/1     Running   2 (3h4m ago)   23h     192.168.232.240   worker-node-2.example.com  <none>           <none>
mysql-7748c687bf-n9gdf         1/1     Running   4 (3h4m ago)   4d23h   192.168.232.239   worker-node-2.example.com  <none>           <none>
nginx-7854ff8877-ktgkp         1/1     Running   3 (3h4m ago)   4d21h   192.168.232.238   worker-node-2.example.com  <none>           <none>
nginx-labels                   1/1     Running   0              26m     192.168.47.170    worker-node-1.example.com  <none>           <none>
openshift                      1/1     Running   3 (3h4m ago)   4d18h   192.168.232.241   worker-node-2.example.com  <none>           <none>
php-apache-5f9f45d488-d4lv7    1/1     Running   5 (3h4m ago)   5d22h   192.168.47.168    worker-node-1.example.com  <none>           <none>
pod-env-var                    1/1     Running   6 (3h4m ago)   5d22h   192.168.47.166    worker-node-1.example.com  <none>           <none>
pod-env12                      1/1     Running   6 (3h4m ago)   5d22h   192.168.232.236   worker-node-2.example.com  <none>           <none>
secret-pod                     1/1     Running   2 (3h4m ago)   23h     192.168.232.237   worker-node-2.example.com  <none>           <none>
testconfig                     0/1     Unknown   0              5d22h   <none>            worker-node-2.example.com  <none>           <none>
with-node-affinity             1/1     Running   0              49s     192.168.232.243   worker-node-2.example.com  <none>           <none>
wordpress-6ff4d555d5-tglfv     1/1     Running   4 (3h4m ago)   4d23h   192.168.232.242   worker-node-2.example.com  <none>           <none>
labsuser@master:~$ 
```

You can see that the node affinity pod is running on worker node 2.

By following these steps, you have successfully configured the pods with nodename and nodeselector fields.