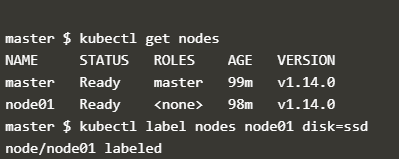
1. Add label to a node
2. Remove label from a node
3. Add label to a pod
4. Remove label to a pod
5. Add label to a deployment
6. Remove label from a deployment
7. Use nodeSelector to schedule a pod on a particular node
8. Use nodeName to schedule a pod on a particular node
9. Use taints to prevent pods from being scheduled on a particular node
10. Use tolerations to ignore taints
11. Use nodeAffinity to schedule a pod on a particular node
12. Use podAntiAffinity to make sure that pods in the same deployments are not scheduled on the same node
13. Use podAffinity to make sure that pods from separate deployments are scheduled on the same node

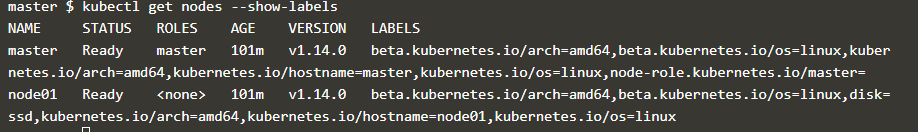
# Add label to a node

kubectl label nodes <node-name> <label-key>=<label-value>

kubectl label nodes node01 disk=ssd



kubectl get nodes --show-labels



apiVersion: v1

kind: Pod

metadata:

name: nginx

labels:

env: test

spec:

containers:

- name: nginx

image: nginx

imagePullPolicy: IfNotPresent

nodeSelector:

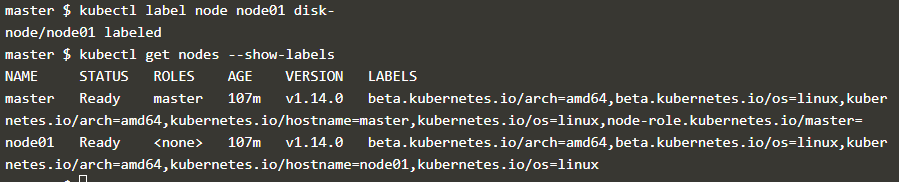
dis: ssd

The Pod will get scheduled on the node that you attached the label to

2. Remove Label from Node

kubectl label node <nodename> <label>-

kubectl label node node01 disk-



3. Add Label to Pod

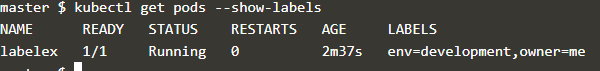
kubectl apply -f <https://raw.githubusercontent.com/openshift-evangelists/kbe/master/specs/labels/pod.yaml>

kubectl label pods <podname> <key>= <value>

kubectl label pods labelex owner=me

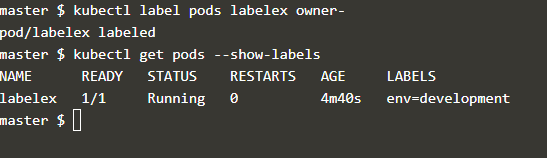


kubectl get pods --show-labels



4. Remove pod label

kubectl label pods labelex owner-



1. Add label to a deployment
2. Remove label from a deployment

$ kubectl patch deployment myDeployment --patch '{"spec": {"template": {"metadata": {"labels": {"myLabelKey": "myLabelValue}}}}}'

$ kubectl patch deployment myDeployment --patch '{"spec": {"template": {"metadata": {"labels": {"myLabelKey": "myLabelValue}}}}}'

Use nodeSelector to schedule a pod on a particular node

Use nodeName to schedule a pod on a particular node