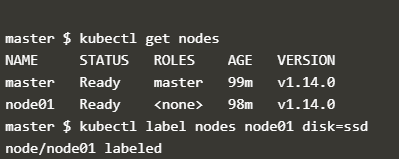
Labels in K8s

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

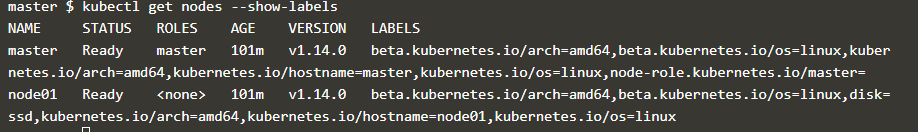
# Add label to a node

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

kubectl label nodes node01 disk=ssd



kubectl get nodes --show-labels



7 Use nodeSelector to schedule a pod on a particular node

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

7 Use node name to schedule a pod on a particular node

apiVersion: v1

kind: Pod

metadata:

name: nginx

spec:

containers:

- name: nginx

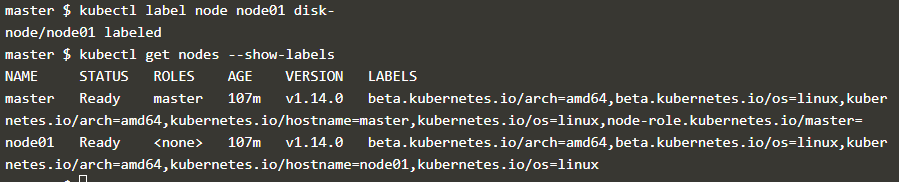
image: nginx

nodeName: kube-01

2. Remove Label from Node

kubectl label node <nodename> <label>-

kubectl label node node01 disk-



3. Add Label to Pod

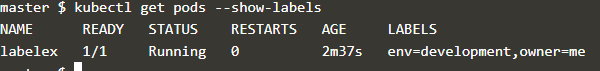
kubectl apply -f <https://raw.githubusercontent.com/openshift-evangelists/kbe/master/specs/labels/pod.yaml>

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

kubectl label pods labelex owner=me

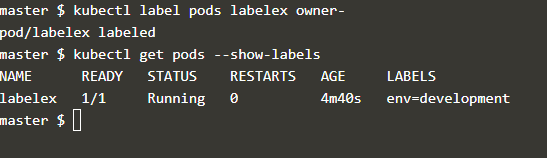


kubectl get pods --show-labels



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}}}}}'

1. Use taints to prevent pods from being scheduled on a particular node

Taints: they allow a node to **repel** a set of pods

Taints and tolerations work together to ensure that pods are not scheduled onto inappropriate nodes.

Tolerations are applied to pods, and allow (but do not require) the pods to schedule onto nodes with matching taints.

Add taints

kubectl taint nodes node1 key=value:NoSchedule

Remove Taints

kubectl taint nodes node1 key:NoSchedule-

apiVersion: v1

kind: Pod

metadata:

name: nginx

labels:

env: test

spec:

containers:

- name: nginx

image: nginx

imagePullPolicy: IfNotPresent

tolerations:

- key: "example-key"

operator: "Exists"

effect: "NoSchedule"

1. Use nodeAffinity to schedule a pod on a particular node

Node affinity is conceptually similar to nodeSelector – it allows you to constrain which nodes your pod is eligible to be scheduled on, based on labels on the node.

two types of node affinity

requiredDuringSchedulingIgnoredDuringExecution (hard)

and preferredDuringSchedulingIgnoredDuringExecution (soft)

Pod Affinity

apiVersion: apps/v1

kind: Deployment

metadata:

name: web-server

spec:

selector:

matchLabels:

app: web-store

replicas: 3

template:

metadata:

labels:

app: web-store

spec:

affinity:

podAntiAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

- labelSelector:

matchExpressions:

- key: app

operator: In

values:

- web-store

topologyKey: "kubernetes.io/hostname"

podAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

- labelSelector:

matchExpressions:

- key: app

operator: In

values:

- store

topologyKey: "kubernetes.io/hostname"

containers:

- name: web-app

image: nginx:1.16-alpine

Anti Affinitity

apiVersion: v1

kind: Pod

metadata:

name: with-pod-affinity

spec:

affinity:

podAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

- labelSelector:

matchExpressions:

- key: security

operator: In

values:

- S1

topologyKey: failure-domain.beta.kubernetes.io/zone

podAntiAffinity:

preferredDuringSchedulingIgnoredDuringExecution:

- weight: 100

podAffinityTerm:

labelSelector:

matchExpressions:

- key: security

operator: In

values:

- S2

topologyKey: failure-domain.beta.kubernetes.io/zone

containers:

- name: with-pod-affinity

image: k8s.gcr.io/pause:2.0

Node affinity

apiVersion: v1

kind: Pod

metadata:

name: with-node-affinity

spec:

affinity:

nodeAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

nodeSelectorTerms:

- matchExpressions:

- key: kubernetes.io/e2e-az-name

operator: In

values:

- e2e-az1

- e2e-az2

preferredDuringSchedulingIgnoredDuringExecution:

- weight: 1

preference:

matchExpressions:

- key: another-node-label-key

operator: In

values:

- another-node-label-value

containers:

- name: with-node-affinity

image: k8s.gcr.io/pause:2.0