Label:  
  
Imagine you have a POD that’s need to be host on a machine with certain specifications ( SSD HD, physical location , processing power , ..,etc )   
OR a imagine you want to search or group your PODs for easier administration   
what would you do ?  
then label is your way to go, in Kubernetes Label are Key/value pairs attached to an object   
let’s see how can we use label with node selector to make a POD is lunched on a certain machine

|  |  |
| --- | --- |
| Tip | * You can assign multiple labels per object but avoid using too much label or too little, too much would get you confused and too little won’t give the real benefits of grouping, selecting and searching * Best practice is to assign labels to indicate application/program ID use this POD owner (who manage this POD/application) stage (the POD/application in development/testing/ production as well version) resource requirements (SSD, CPU, storage) location (preferred location/zone/ Datacenter to run this POD/application) |

Let’s assign label stage: testing & zone: production to these two nodes respectively then try to lunch a POD with label zone: testing

kubectl get nodes --show-labels

NAME STATUS ROLES AGE VERSION LABELS

ip-172-25-1-216.us-west-1.compute.internal Ready <none> 2h v1.9.2 <none>

ip-172-25-1-56.us-west-1.compute.internal NotReady <none> 2h v1.9.2 <none>

ip-172-25-1-83.us-west-1.compute.internal Ready <none> 2h v1.9.2 <none>

kubectl label nodes ip-172-25-1-83.us-west-1.compute.internal stage=testing

kubectl label nodes ip-172-25-1-216.us-west-1.compute.internal stage=production

kubectl get nodes --show-labels

NAME STATUS ROLES AGE VERSION LABELS

ip-172-25-1-216.us-west-1.compute.internal Ready <none> 2h v1.9.2 stage=production

ip-172-25-1-56.us-west-1.compute.internal NotReady <none> 2h v1.9.2 <none>

ip-172-25-1-83.us-west-1.compute.internal Ready <none> 2h v1.9.2 stage=testing

now let’s lunch a basic Nginx POD tagged with stage =testing and confirm it will land on a node tagged with stage=testing

[root@ip-172-25-1-56 /]# cat > web-server.yaml

apiVersion: v1

kind: Pod

metadata:

name: nginx

labels:

app: webserver

spec:

containers:

- name: nginx

image: nginx

nodeSelector:

stage: testing

[root@ip-172-25-1-56 /]# kubectl create -f web-server.yaml

pod "wordpress" created  
  
[root@ip-172-25-1-56 /]# kubectl get pods --output=wide

NAME READY STATUS RESTARTS AGE IP NODE

wordpress 1/1 Running 0 48s 10.47.255.250 ip-172-25-1-83.us-west-1.compute.internal

|  |  |
| --- | --- |
| Tip | * You can assign POD to certain node without label by adding the argument nodeName: nodeX under spec in the YAML file where nodeX is the name of the node |

Another very important use of labels is linking PODs to Replica-set , Deployment , services …,etc and that’s what we will see next with services and label selector