What is node affinity ?

Kubernetes - Beginners | Intermediate | Advanced

What is node affinity ?

- In simple words this allows you to tell Kubernetes to schedule pods only to specific subsets of nodes.
- The initial node affinity mechanism in early versions of Kubernetes was the nodeSelector field in the pod specification. The node had to include all the labels specified in that field to be eligible to become the target for the pod.

nodeSelector

Steps

```
git clone https://github.com/collabnix/dockerlabs
cd dockerlabs/kubernetes/workshop/Scheduler101/
kubectl label nodes node2 mynode=worker-1
kubectl apply -f pod-nginx.yaml
```

- We have label on the node with node name,in this case i have given node2 as

mynode=worker-1 label.

Viewing Your Pods

```
kubectl get pods --output=wide
```

```
[node1 Scheduler101]$ kubectl describe po nginx
Name:               nginx
Namespace:          default
Priority:           0
PriorityClassName:  <none>
Node:               node2/192.168.0.17
Start Time:         Mon, 30 Dec 2019 16:40:53 +0000
Labels:             env=test
Annotations:        kubectl.kubernetes.io/last-applied-configuration:
                      {"apiVersion":"v1","kind":"Pod","metadata":{"annotatio
Status:             Pending
IP:
Containers:
  nginx:
    Container ID:
    Image:          nginx
    Image ID:
    Port:           <none>
    Host Port:      <none>
    State:          Waiting
      Reason:       ContainerCreating
    Ready:          False
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-qpgxq
Conditions:
  Type              Status
  Initialized       True
  Ready             False
  ContainersReady   False
  PodScheduled      True
```

```
Volumes:
  default-token-qpgxq:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-qpgxq
    Optional:    false
QoS Class:       BestEffort
Node-Selectors:  mynode=worker-1
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  7s    default-scheduler  Successfully assigned default/
  Normal  Pulling    3s    kubelet, node2     Pulling image "nginx"
[node1 Scheduler101]$
```

- You can check in above output Node-Selectors: mynode=worker-1

Deleting the Pod

```
kubectl delete -f pod-nginx.yaml
pod "nginx" deleted
```

Node affinity

- 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.
- There are currently two types of node affinity.
    1. requiredDuringSchedulingIgnoredDuringExecution (Preferred during scheduling, ignored during execution; we are also known as “hard” requirements)
    2. preferredDuringSchedulingIgnoredDuringExecution (Required during scheduling, ignored during execution; we are also known as “soft”

requirements)

Steps

```
git clone https://github.com/collabnix/dockerlabs
cd dockerlabs/kubernetes/workshop/Scheduler101/
kubectl label nodes node2 mynode=worker-1
kubectl label nodes node3 mynode=worker-3
kubectl apply -f pod-with-node-affinity.yaml
```

Viewing Your Pods

```
kubectl get pods --output=wide
NAME                READY   STATUS    RESTARTS   AGE     IP          NODE
with-node-affinity   1/1     Running   0          9m46s   10.44.0.1   kube-s
```

```
[node1 Scheduler101]$ kubectl describe po
Name:               with-node-affinity
Namespace:          default
Priority:           0
PriorityClassName:  <none>
Node:               node3/192.168.0.16
Start Time:         Mon, 30 Dec 2019 19:28:33 +0000
Labels:             <none>
Annotations:        kubectl.kubernetes.io/last-applied-configuration:
                      {"apiVersion":"v1","kind":"Pod","metadata":{"annotatio
Status:             Pending
IP:
Containers:
  nginx:
    Container ID:
    Image:          nginx
    Image ID:
    Port:           <none>
```

```
    Host Port:      <none>
    State:          Waiting
      Reason:       ContainerCreating
    Ready:          False
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-qpgxq
Conditions:
  Type              Status
  Initialized       True
  Ready             False
  ContainersReady   False
  PodScheduled      True
Volumes:
  default-token-qpgxq:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-qpgxq
    Optional:    false
QoS Class:       BestEffort
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  26s   default-scheduler  Successfully assigned default/
  Normal  Pulling    22s   kubelet, node3     Pulling image "nginx"
  Normal  Pulled     20s   kubelet, node3     Successfully pulled image "ngi
  Normal  Created    2s    kubelet, node3     Created container nginx
  Normal  Started    0s    kubelet, node3     Started container nginx
```

Step Cleanup

Finally you can clean up the resources you created in your cluster:

```
kubectl delete -f pod-with-node-affinity.yaml
```

Next»

Join KubeDaily

9 Members Online

Support

MEMBERS ONLINE

- Aman Manapure
- h3ll_boy
- MEE6
- Nitinkashyap — Apex Legends
- ojaswa
- Parmeshwar
- prasad
- trimankaur
- vikas027

Free voice chat from Discord

Connect

Tweets by collabnix

kubelabs is maintained by collabnix.

This page was generated by GitHub Pages.