Anti-Node Affinity ?

Kubernetes - Beginners | Intermediate | Advanced

View on GitHub Join Slack

Anti-Node Affinity ?

- Some scenarios require that you don't use one or more nodes except for particular pods. Think of the nodes that host your monitoring application.
- Those nodes shouldn't have many resources due to the nature of their role. Thus, if other pods than those which have the monitoring app are scheduled to those nodes, they hurt monitoring and also degrades the application they are hosting.
- In such a case, you need to use node anti-affinity to keep pods away from a set of nodes.

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-anti-node-affinity.yaml
```

Viewing Your Pods

```
[node1 Scheduler101]$ kubectl get pods --output=wide
NAME    READY   STATUS    RESTARTS   AGE     IP          NODE    NOMINATED N
nginx   1/1     Running   0          2m37s   10.44.0.1   node2   <none>
```

Get nodes label detail

```
[node1 Scheduler101]$ kubectl get nodes --show-labels | grep mynode
node2   Ready    <none>   166m   v1.14.9   beta.kubernetes.io/arch=amd64,bet
node3   Ready    <none>   165m   v1.14.9   beta.kubernetes.io/arch=amd64,bet
```

Get pod describe

```
[node1 Scheduler101]$ kubectl describe pods nginx
Name:               nginx
Namespace:          default
Priority:           0
PriorityClassName:  <none>
Node:               node2/192.168.0.17
Start Time:         Mon, 30 Dec 2019 19:02:46 +0000
Labels:             <none>
Annotations:        kubectl.kubernetes.io/last-applied-configuration:
                      {"apiVersion":"v1","kind":"Pod","metadata":{"annotatio
Status:             Running
IP:                 10.44.0.1
Containers:
  nginx:
    Container ID:   docker://2bdc20d79c360e1cd857eeb9bbb9424c726b2133e78f25b
    Image:          nginx
    Image ID:       docker-pullable://nginx@sha256:b2d89d0a210398b4d1120b3e3
    Port:           <none>
    Host Port:      <none>
```

```
    State:          Running
      Started:      Mon, 30 Dec 2019 19:03:07 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-qpgxq
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  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  60s   default-scheduler  Successfully assigned default/
  Normal  Pulling    56s   kubelet, node2     Pulling image "nginx"
  Normal  Pulled     54s   kubelet, node2     Successfully pulled image "ngi
  Normal  Created    40s   kubelet, node2     Created container nginx
  Normal  Started    39s   kubelet, node2     Started container nginx
```

- Adding another key to the matchExpressions with the operator NotIn will avoid scheduling the nginx pods on any node labelled worker-1.

Step Cleanup

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

```
kubectl delete -f pod-anti-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.