

kube-vip DaemonSet als Ersatz für MetalLB

kube-vip DaemonSet

Worker Nodes markieren.

```
kubectl label node worker1 node-role.kubernetes.io/worker=""  
kubectl label node worker2 node-role.kubernetes.io/worker=""
```

RBAC.yml

```
kubectl apply -f https://kube-vip.io/manifests/rbac.yaml
```

daemonset.yml

```
apiVersion: apps/v1  
kind: DaemonSet  
metadata:  
  annotations:  
    name: kube-vip-lb  
    namespace: kube-system  
spec:  
  selector:  
    matchLabels:  
      name: kube-vip-lb  
  template:  
    metadata:  
      creationTimestamp: null  
      labels:  
        name: kube-vip-lb  
    spec:  
      affinity:  
        nodeAffinity:  
          requiredDuringSchedulingIgnoredDuringExecution:  
            nodeSelectorTerms:  
              - matchExpressions:  
                - key: node-role.kubernetes.io/worker  
                  operator: Exists  
      containers:  
        - args:  
            - manager  
          env:  
            - name: vip_arp  
              value: "true"  
            - name: vip_nodename  
              valueFrom:  
                fieldRef:
```

```
      fieldPath: spec.nodeName
-   name: vip_interface
    value: ens3
-   name: vip_cidr
    value: "32"
-   name: cp_enable
    value: "true"
-   name: vip_ddns
    value: "false"
-   name: svc_enable
    value: "true"
-   name: vip_leaderelection
    value: "true"
-   name: vip_leaseduration
    value: "5"
-   name: vip_renewdeadline
    value: "3"
-   name: vip_retryperiod
    value: "1"
-   name: lb_enable
    value: "true"
image: ghcr.io/kube-vip/kube-vip:v1.0.1
imagePullPolicy: Always
name: kube-vip
resources: {}
securityContext:
  capabilities:
    add:
      - NET_ADMIN
      - NET_RAW
      - SYS_TIME
  terminationMessagePath: /dev/termination-log
  terminationMessagePolicy: File
dnsPolicy: ClusterFirst
hostNetwork: true
restartPolicy: Always
schedulerName: default-scheduler
securityContext: {}
serviceAccount: kube-vip
serviceAccountName: kube-vip
terminationGracePeriodSeconds: 30
updateStrategy:
  rollingUpdate:
    maxSurge: 0
    maxUnavailable: 1
  type: RollingUpdate
```

Service Manifest als Beispiel

svc-homepage.yml

```
apiVersion: v1
kind: Service
metadata:
  creationTimestamp: null
  labels:
    app: svc-homepage
  name: svc-homepage
  namespace: webserver
spec:
  allocateLoadBalancerNodePorts: false
  loadBalancerIP: <dig ingress-<userid>.training.lab>
  ports:
    - name: "80"
      port: 80
      protocol: TCP
      targetPort: 80
  selector:
    run: nginx
  type: LoadBalancer
status:
  loadBalancer: {}
```

From:

<https://www.cooltux.net/> - TuxNet DokuWiki

Permanent link:

https://www.cooltux.net/doku.php?id=it-wiki:kubernetes:kube-vip_daemonset_ersatz_fuer_metallb

Last update: 2025/10/16 09:44

