

部署k8s-v1.20.15

环境准备

- 部署k8s-v1.20.15 <https://io.vhm.com/book/books/k8s/page/kube-vipk8s>
- 部署keepalived <https://io.vhm.com/book/books/42e7a/page/docker-composekeepalived>

kube-vip部署keepalived部署IP

部署

```
# 部署环境

hostnamectl set-hostname 5-10.vpclub.io

# 部署hosts部署master部署kube-api-server

vi /etc/hosts
192.168.5.10    5-10.vpclub.io
192.168.5.10   kube-api-server

# 部署k8s部署
# 部署k8s部署 https://dl.k8s.io/release/stable.txt
# 部署google部署
# curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

# 部署环境

wget qq829.cn/uploads/software/k8s/v1.20.15/kubeadm
wget qq829.cn/uploads/software/k8s/v1.20.15/kubelet
wget qq829.cn/uploads/software/k8s/v1.20.15/kubectl

# docker-compose
```

```
wget qq829.cn/uploads/software/k8s/docker-compose
```

```
# 创建目录
```

```
wget qq829.cn/uploads/software/k8s/v1.20.15/10-kubeadm.conf
```

```
wget qq829.cn/uploads/software/k8s/v1.20.15/kubelet.service
```

```
# 安装
```

```
wget qq829.cn/uploads/software/k8s/v1.20.15/calico.yaml
```

```
wget qq829.cn/uploads/software/k8s/v1.20.15/traefik-ingress.tar
```

```
# 配置cni
```

```
wget qq829.cn/uploads/software/k8s/v1.20.15/calico-image-vp-whdev.yaml
```

```
# 创建目录并安装到本地目录
```

```
chmod 777 kubeadm kubelet kubectl docker-compose
```

```
cp kubeadm kubelet kubectl docker-compose /usr/local/bin
```

```
# 安装 socat 和 conntrack
```

```
yum install -y socat
```

```
yum install -y conntrack
```

```
# 安装 docker
```

```
vi /etc/docker/daemon.json
```

```
{  
    "graph": "/data/docker",  
    "exec-opts": ["native.cgroupdriver=systemd"]  
}
```

```
# 安装 kubelet
```

```
# 配置 kubelet
```

```
cp kubelet.service /usr/lib/systemd/system/kubelet.service
mkdir -p /etc/systemd/system/kubelet.service.d
cp 10-kubeadm.conf /etc/systemd/system/kubelet.service.d

# 配置kubelet 配置
systemctl enable kubelet.service
systemctl status kubelet
```

配置master节点

```
# 配置kubeadm 配置
# 1.24.0 配置docker
# 配置apiserver 配置kube-api-server: 6443

kubeadm init --control-plane-endpoint "kube-api-server:6443" --image-repository
registry.aliyuncs.com/google_containers --upload-certs --kubernetes-version v1.20.15 --pod-
network-cidr 10.244.0.0/16 --v 5

# 配置master 配置
echo "$(kubeadm token create --print-join-command) --control-plane --certificate-key $(kubeadm
init phase upload-certs --upload-certs | tail -1)"

# 配置node 配置

kubeadm token create --print-join-command
```

配置master节点<https://ioyhm.com/book/books/k8s/page/k8s-ZsW>

```
# 配置
kubeadm init --image-repository registry.aliyuncs.com/google_containers --apiserver-
advertise-address 192.168.5.10 --kubernetes-version v1.20.15 --pod-network-cidr
10.244.0.0/16 --v 5

# 配置
kubeadm reset
```

```
# 安装ipvs
# 40s , 安装54s mode:"ipvs"
kubectl edit cm -n kube-system kube-proxy

# 安装cni

kubectl apply -f calico.yaml
```

安装ingress-controller

安装traefik-ingress ~~nginx-ingress~~ 安装

```
# 安装nginx-ingress
wget qq829.cn/uploads/software/k8s/v1.20.15/nginx-ingress/nginx-ingress.yaml

kubectl apply -f nginx-ingress.yaml
```

```
# 安装traefik-controller

kubectl apply -f 00-account.yaml -n kube-system
kubectl apply -f 01-role.yaml -n kube-system
kubectl apply -f 02-role-binding.yaml -n kube-system
kubectl apply -f 03-traefik.yaml -n kube-system
kubectl apply -f 04-traefik-services.yaml -n kube-system
kubectl apply -f 05-traefik-default-tls.yaml -n kube-system

# 生成token
# 生成token
# kubeadm token create --print-join-command
```

k8s <https://iovhm.com/book/books/k8s/page/k3srancher>



```

# []
# vi /var/lib/kubelet/kubeadm-flags.env
# --max-pods=50

# pod[]ping[]
# kubectl edit cm -n kube-system kube-proxy
# 40[]mode: "ipvs"
# []k8s[]

k8s[]pod ping[]kube-proxy[]ipvs[]iptables[]kube-proxy

# {} [] controller-manager []
# {} [] scheduler []
# []
vi /etc/kubernetes/manifests/kube-controller-manager.yaml
vi /etc/kubernetes/manifests/kube-scheduler.yaml
# [] --port=0

# calico-node [] calico/node is not ready: BIRD is not ready: BGP not established
# calico-node []
# [], []eth0[]
IP_AUTODETECTION_METHOD =[]interface=eth0

# traefik-ingress []
# traefik []ingressendpoint.hostname[]ingressendpoint.hostname[]
# [] --providers.kubernetesingress.ingressendpoint.publishedService

--providers.kubernetesingress.ingressendpoint.hostname=park.vpclub.io
--providers.kubernetesingress.ingressendpoint.ip=0.0.0.0

# calico-kube-controllers []
# []Failed to write status error=open /status/status.json: permission denied
# []
# []/var/run/calico/status [] 777[] /status
# []

```

```
# load balance L4 负载均衡 pedding
# load balance 负载均衡k8s 负载均衡 nodeport 负载均衡 hostport
# 负载均衡traefik 负载均衡DaemonSet 负载均衡 host负载均衡

# 负载均衡kube-proxy负载均衡coredns负载均衡
kubeadm init phase addon kube-proxy --kubeconfig ~/.kube/config --control-plane-endpoint
"kube-api-server:6443" --image-repository registry.aliyuncs.com/google_containers --
kubernetes-version v1.28.0 --pod-network-cidr 10.244.0.0/16
```



```
# 负载均衡master负载均衡
kubectl get cs

# 负载均衡
kubectl get nodes

# 负载均衡pod负载均衡
kubectl get pods

# 负载均衡
kubectl get deployments

# 负载均衡
kubectl cluster-info

# 负载均衡
kubectl delete node node42.vpclub.io

# 负载均衡pod
kubectl delete pod nginx-3654852276-2dt73

# 负载均衡deployment
kubectl delete deployment nginx
```

```
# 查看pod
```

```
kubectl describe pods
```

```
# 进入
```

```
kubectl exec -it <nginx-webapp-2067515279-1z0lb> /bin/bash
```

```
# 查看yaml文件
```

```
kubectl get deploy NAME -o yaml
```

```
# 删除
```

```
kubectl delete pod <pod> --grace-period=0 --force
```

```
# 给node打label
```

```
kubectl label nodes 190.vpclub.io <label>
```

```
# 给node打label, 覆盖label -
```

```
kubectl label nodes 190.vpclub.io <label>-
```

```
# 给label , 覆盖 --overwrite
```

```
kubectl label nodes 190.vpclub.io role=apache --overwrite
```

```
# 给node打label -
```

```
# 给node打label POD
```

```
kubectl taint nodes 190.vpclub.io node-role.kubernetes.io/master=:NoExecute
```

```
# 给node打label
```

```
kubectl taint nodes 190.vpclub.io node-role.kubernetes.io/master=:NoSchedule
```

```
# 删除POD
```

```
kubectl get pod -n kube-system | grep kube-proxy | awk '{system("kubectl delete pod \"$1\" -n kube-system")}'
```

```
# 隔离
```

```
kubectl cordon 17.vpclub.io
```

```
# 解除隔离
```

```
kubectl uncordon 5-16.vpclub.io
```

```
# 查看nginx-ingress
```

```
kubectl delete -A ValidatingWebhookConfiguration ingress-nginx-admission
```

☐ ☐ ☐ ☐ ☐ ☐ POD

```
kubectl get pods --all-namespaces| grep
```

```
"Terminating\| OutOfpods\| CrashLoopBackOff\| Evicted\| ContainerStatusUnknown\| Error"
```

[] [] [] [] [] [] [] POD

```
kubectl get pods -n trade | grep
```

```
"Terminating\| OutOfpods\| CrashLoopBackOff\| Evicted\| ContainerStatusUnknown\| Error" | awk
```

```
'{print $1}' | xargs kubectl delete pod -n trade --force --grace-period=0
```

[] [] [] [] [] [] [] [] [] [] P0

```
_opns=kube-system && kubectl get pods -n ${_opns} | grep
```

```
"Terminating\| OutOfpods\| CrashLoopBackOff\| Evicted\| ContainerStatusUnknown\| Error" | awk
```

```
'{print $1}' | xargs kubectl delete pod -n ${_opns} --force --grace-period=0
```


--	--	--	--

```
kubectl scale -n devops-default --replicas=0 deployment/devops-admin-api
```

[illegible]

```
kubectl scale deploy --replicas=0 --all -n park-zjy
```

d

```
kubectl get deploy --all-namespaces --sort-by=.metadata.creationTimestamp
```

```
# [ ] [ ] [ ] [ ]cpu[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]metrics-server
```

```
kubectl top node --sort-by memory
```

```
# pod[ ]metrics-server
```

```
#kubectl top pods --sort-by memory
```

```
❏ master ❏
```

```
# [ ] [ ] [ ] [ ] node [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
```

```
kubectl delete node 01.vpclub.io
```

```
# [ ] [ ] [ ] [ ] [ ] etcd [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
```

```
kubectl exec -it etcd-00 sh -n kube-system
```

#	Pod	ETCD
1	Pod 1	ETCD 1
2	Pod 2	ETCD 2
3	Pod 3	ETCD 3
4	Pod 4	ETCD 4
5	Pod 5	ETCD 5
6	Pod 6	ETCD 6
7	Pod 7	ETCD 7
8	Pod 8	ETCD 8
9	Pod 9	ETCD 9
10	Pod 10	ETCD 10
11	Pod 11	ETCD 11
12	Pod 12	ETCD 12
13	Pod 13	ETCD 13
14	Pod 14	ETCD 14
15	Pod 15	ETCD 15
16	Pod 16	ETCD 16
17	Pod 17	ETCD 17
18	Pod 18	ETCD 18
19	Pod 19	ETCD 19
20	Pod 20	ETCD 20
21	Pod 21	ETCD 21
22	Pod 22	ETCD 22
23	Pod 23	ETCD 23
24	Pod 24	ETCD 24
25	Pod 25	ETCD 25
26	Pod 26	ETCD 26
27	Pod 27	ETCD 27
28	Pod 28	ETCD 28
29	Pod 29	ETCD 29
30	Pod 30	ETCD 30
31	Pod 31	ETCD 31
32	Pod 32	ETCD 32
33	Pod 33	ETCD 33
34	Pod 34	ETCD 34
35	Pod 35	ETCD 35
36	Pod 36	ETCD 36
37	Pod 37	ETCD 37
38	Pod 38	ETCD 38
39	Pod 39	ETCD 39
40	Pod 40	ETCD 40
41	Pod 41	ETCD 41
42	Pod 42	ETCD 42
43	Pod 43	ETCD 43
44	Pod 44	ETCD 44
45	Pod 45	ETCD 45
46	Pod 46	ETCD 46
47	Pod 47	ETCD 47
48	Pod 48	ETCD 48
49	Pod 49	ETCD 49
50	Pod 50	ETCD 50


```
export ETCDCTL_API=3
alias etcdctl='etcdctl --endpoints=https://127.0.0.1:2379 --
cacert=/etc/kubernetes/pki/etcd/ca.crt --cert=/etc/kubernetes/pki/etcd/server.crt --
key=/etc/kubernetes/pki/etcd/server.key'
```

```
# 查看成员
```

```
etcdctl member list
```

```
# 查看
```

```
669bc6472fb13679, started, master1, https://192.168.1.19:2380, https://192.168.1.19:2379,
false
```

```
959c93e3261aadcb, started, master2, https://192.168.1.20:2380, https://192.168.1.20:2379,
false
```

```
ca5f1f6f780545ba, started, master3, https://192.168.1.23:2380, https://192.168.1.23:2379,
false
```

```
# 删除master3
```

```
etcdctl member remove ca5f1f6f780545ba
```

```
查看 #39
```

```
查看 21 2023 04:58:11
```

```
查看 6 2024 10:10:16
```