

部署k8s-v1.20.15

环境准备

- 部署k8s-v1.20.15 <https://qq829.cn/book/books/k8s/page/kube-vipk8s>
- 部署keepalived <https://qq829.cn/book/books/42e7a/page/docker-composekeepalived>

kube-vip部署keepalived部署IP

部署

```
# 环境准备

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"

# 环境准备

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
```

```
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 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://qq829.cn/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
```

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

# 配置cni

kubectl apply -f calico.yaml
```

部署ingress-controller

traefik-ingress ~~nginx-ingress~~ 部署

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

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

```
# 部署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://qq829.cn/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")}'
```

```
# 查看
```

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

```
# 查看
```

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

```
# 查看nginx-ingress
```

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

```
# 删除POD
```

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

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

```
# 删除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
```

```
# 删除POD
```

```
_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
```

```
# 删除
```

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

```
# 删除NS
```

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

```
# 删除
```

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

```
# 查看cpu metrics-server
```

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

```
# 查看pod metrics-server
```

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

master

```
# 删除node
```

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

```
# 删除etcd
```

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

```
# Pod ETCD
```



```
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
```