

# k8s



```
# master
kubectl get cs

# 
kubectl get nodes

# pod
kubectl get pods

# 
kubectl get deployments

# 
kubectl cluster-info

# 
kubectl delete node node42.vpclub.io

# pod
kubectl delete pod nginx-3654852276-2dt73

# deployment
kubectl delete deployment nginx

# 
kubectl describe pods

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

# nginx.yaml
```



```

"Terminating\| OutOfpods\| CrashLoopBackOff\| Evicted\| ContainerStatusUnknown\| Error" | awk
'{print $1}' | xargs kubectl delete pod -n trade --force --grace-period=0

# 000000000000P0
_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

# 0000
kubectl scale -n devops-default --replicas=0 deployment/devops-admin-api

# 000NS0000000000000
kubectl scale deploy --replicas=0 --all -n park-zjy

# 0000d00
kubectl get deploy --all-namespaces --sort-by=.metadata.creationTimestamp

# 0000cpu000000000000metrics-server
kubectl top node --sort-by memory

# 00pod000000000000metrics-server
#kubectl top pods --sort-by memory

```

## 00master00

```

# 0000node0000000000
kubectl delete node 01.vpclub.io

# 0000etcd0000000000
kubectl exec -it etcd-00 sh -n kube-system

# Pod 0000 ETCD 00000000000000000000
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'

# 00000000
etcdctl member list

```

```
# []
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

# []master3[]
etcdctl member remove ca5f1f6f780545ba
```

## docker

```
# []
# - a[]
# - f[]
docker system prune -af

# []
docker volume prune

# []
docker network prune

# []
docker volume prune

# []
docker image prune

# []tag[]
docker rmi $(docker images -f "dangling=true" -q)

# []tag[]
docker images -f "dangling=true"
```



---

□□ #8

□ □□ □□ 16 □ 2025 04:27:12

□ □□ □□ 30 □ 2025 04:38:42