k8s常用命令

常用命令行

```
# 查看master组件状态
kubectl get cs

# 查看集群节点
kubectl get nodes

# 查看pod信息
kubectl get pods

# 获取发布信息
kubectl get deployments

# 查看集群信息
kubectl cluster-info

# 删除节点
kubectl delete node node42.vpclub.io

# 删除 pod
kubectl delete pod nginx-3654852276-2dt73

# 删除deployment
kubectl delete deployment nginx

# 详细日志调试工具
kubectl describe pods

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

# 查看已经部署的yml配置信息
```

```
kubectl get deploy NAME -o yaml

# □□□□
kubectl delete pod <pod□> --grace-period=0 --force

# □node□□label
kubectl label nodes 190.vpclub.io <label>

# □□node□label,□□label□□□ -
kubectl label nodes 190.vpclub.io <label>-

# □□□□label ,□□□□□□ --overwrite
kubectl label nodes 190.vpclub.io role=apache --overwrite

# □□□□□□□□□□□□□□□□□□ label□□□ -
# □□□□□□□□□□□□□□□□□□POD
kubectl taint nodes 190.vpclub.io node-role.kubernetes.io/master=:NoExecute

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

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

# □□□□□
kubectl cordon 17.vpclub.io

# □□□□□□
kubectl uncordon 5-16.vpclub.io

# □□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

# □□□□□□□□□□□□□□PO
_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

# □□□NS□□□□□□□□□□0
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 □□□□□□□□□□□□□□□□□□
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'

# □□□□□□□□
etcdctl member list
```

```
# 示例:
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节点
etcdctl member remove ca5f1f6f780545ba
```

清除未使用的docker资源

```
# 清理所有未被使用的资源（包括镜像、容器、网络等）
# -a：清理所有未被使用的资源，包括镜像、容器、网络等。
# -f：强制执行清理操作，不进行确认提示。
docker system prune -af

# 清理未被使用的容器
docker volume prune

# 清理未被使用的网络
docker network prune

# 清理未被使用的卷
docker volume prune

# 清理未被使用的镜像
docker image prune

# 清理tag为空的镜像
docker rmi $(docker images -f "dangling=true" -q)

# 列出tag为空的镜像
docker images -f "dangling=true"
```

□pod□□□□□□□□□□□□□□□

```
# □□□volumes□□
      tolerations:
        - effect: NoSchedule
          operator: Exists
        - key: CriticalAddonsOnly
          operator: Exists
        - effect: NoExecute
          operator: Exists
```

□□□□□□□□□IP□□

```
# □□□□
apiVersion: v1
kind: Service
metadata:
  name: mysql # □□□□□□
spec:
  clusterIP:  None # headless
  ports:
    - name: default
      protocol: TCP
      port: 42
      targetPort: 42

---
# □□□□
apiVersion: v1
kind: Endpoints
metadata:
  name: mysql # □□□□□□
subsets:
  - addresses:
      - ip: 192.168.0.46  # □□□□
    ports:
      - name: default
        port: 42
        protocol: TCP
```

□□□ #8
□ □□□ □□□ 16 □□ 2025 04:27:12
□ □□□ □□□ 30 □□ 2025 04:38:42