# 显示最近创建的容器

docker ps -l

docker logs

# 查看容器日志信息

docker ps

# 打印pod中容器的的日志信息

docker logs -f -t --tail 100 960581f1f696

960581f1f696--容器ID

#查看集群节点状态

kubectl get nodes -n

kubectl -n {$nameSpace} -o yaml

kubectl get {$sourceType} --all-namespaces

通过kubectl api-versions 查看所有的api版本

## **通过kubectl explain查看api字段**

可以通过--recursive 查看所有的字段（来列出所有可能的字段）

**kubectl explain svc --recursive**

## 创建对象

通过yaml文件创建：

kubectl create -f xxx.yaml （不建议使用，无法更新，必须先delete）

kubectl apply -f xxx.yaml （创建+更新，可以重复使用）

##查看集群信息（列出使用到的组件以及暴露的Api方法）

kubectl cluster-info

# k8s获取命名空间

kubectl get secret -n research-management

#编辑dockerfile后，执行docker build file的命令构建一个镜像

  vi Dockerfile

  docker build -t 172.17.2.213:8090/devops/docker:proxy-enabled .

pstree -a

###查看kubectl各个节点的信息

## 查看单独节点的信息

kubectl describe node [节点名称]

kubectl describe node 172.17.64.154

每个k8s node节点上运行着：

* Kubelet，管理Kubernetes Master和Node之间的通信; 管理机器上运行的Pods和containers容器。
* container runtime（如Docker，rkt）。

ReplicationController：

kubectl get namespace , 查询所有namespace

**kubectl get namespaces**

**kubectl delete namespace -name**

**kubectl create namespace -name**

node

#用户保存配置数据的键值对！

kubectl create configmap

#删除

#查看configmap的信息

kubectl describe configmap nacos-cm

#查看pod服务运行在哪个具体的节点上

kubectl get pod -n csmc-mangement -o -wide

-n 是命名空间的意思。

加参数 -o wide 可显示运行节点

#获取集群的状态

kubectl get cluster-status