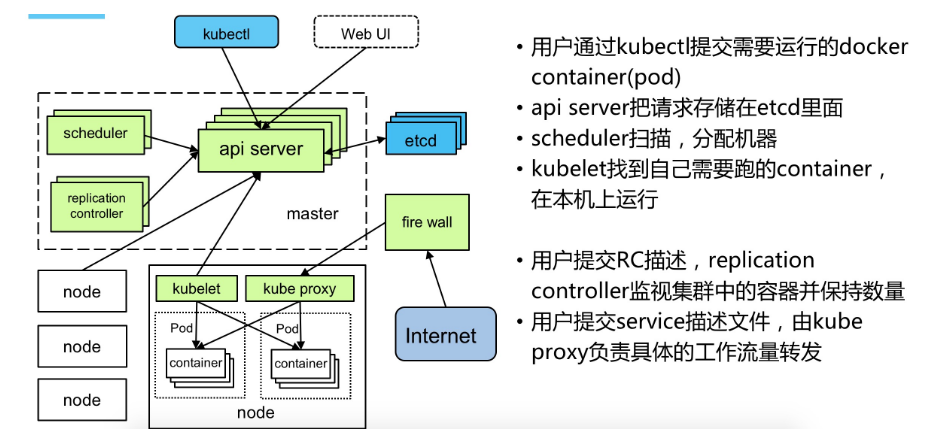
**docker k8s(kubernetes)集群搭建**



centos7.2

docker1.2

docker-master: 172.22.3.207

docker-node1: 172.22.3.111

**一 docker-master上安装k8s/docker**

**主机hosts文件**

echo "172.22.3.207 docker-master

172.22.3.111 docker-node1" >> /etc/hosts

参考 https://kubernetes.io/docs/getting-started-guides/centos/centos\_manual\_config/

所有主机master node创建/etc/yum.repos.d/virt7-docker-common-release.repo文件其中包含以下信息。

**[virt7-docker-common-release]**

**name=virt7-docker-common-release**

**baseurl=http://cbs.centos.org/repos/virt7-docker-common-release/x86\_64/os/**

**gpgcheck=0**

在所有主机上安装Kubernetes/etc/docker

**yum -y install --enablerepo=virt7-docker-common-release kubernetes etcd**

**~~yum install docker\* -y~~**

编辑/ etc / kubernetes / config将在所有主机上相同以包含

***# logging to stderr means we get it in the systemd journal***

**KUBE\_LOGTOSTDERR="--logtostderr=true"**

***# journal message level, 0 is debug***

**KUBE\_LOG\_LEVEL="--v=0"**

***# Should this cluster be allowed to run privileged docker containers***

**KUBE\_ALLOW\_PRIV="--allow-privileged=false"**

***# How the replication controller and scheduler find the kube-apiserver***

**KUBE\_MASTER="--master=http://docker-master:8080"**

禁用主节点和所有节点上的防火墙

**setenforce 0**

**systemctl disable firewalld**

**systemctl stop firewalld**

编辑/etc/etcd/etcd.conf显示为：

***# [member]***

**ETCD\_NAME=default**

**ETCD\_DATA\_DIR="/var/lib/etcd/default.etcd"**

**ETCD\_LISTEN\_CLIENT\_URLS="http://0.0.0.0:2379"**

***#[cluster]***

**ETCD\_ADVERTISE\_CLIENT\_URLS="http://0.0.0.0:2379"**

编辑/ etc / kubernetes / apiserver出现如下：

***# The address on the local server to listen to.***

**KUBE\_API\_ADDRESS="--insecure-bind-address=0.0.0.0"**

***# The port on the local server to listen on.***

**KUBE\_API\_PORT="--port=8080"**

***# Port kubelets listen on***

**KUBELET\_PORT="--kubelet-port=10250"**

***# Comma separated list of nodes in the etcd cluster***

**KUBE\_ETCD\_SERVERS="--etcd-servers=http://docker-master:2379"**

***# Address range to use for services***

**KUBE\_SERVICE\_ADDRESSES="--service-cluster-ip-range=10.254.0.0/16"**

***# Add your own!***

**KUBE\_API\_ARGS=""**

10.254.0.0/16网段为创建service时分配的地址

在主机上启动相应的服务

**for SERVICES in etcd kube-apiserver kube-controller-manager kube-scheduler docker; do**

**systemctl restart $SERVICES**

**systemctl enable $SERVICES**

**systemctl status $SERVICES**

**done**

service docker start 启动docker

二 **在节点上配置Kubernetes服务。**

编辑/ etc / kubernetes / kubelet出现如下

***# The address for the info server to serve on***

**KUBELET\_ADDRESS="--address=0.0.0.0"**

***# The port for the info server to serve on***

**KUBELET\_PORT="--port=10250"**

***# You may leave this blank to use the actual hostname***

***# Check the node number!***

**KUBELET\_HOSTNAME="--hostname-override=docker-node1"**

***# Location of the api-server***

**KUBELET\_API\_SERVER="--api-servers=http://docker-master:8080"**

***# Add your own!***

**KUBELET\_ARGS=""**

在节点（docker-node1）上启动适当的服务

**for SERVICES in kube-proxy kubelet docker; do**

**systemctl restart $SERVICES**

**systemctl enable $SERVICES**

**systemctl status $SERVICES**

**done**

配置kubectl

**kubectl config set-cluster default-cluster --server=http://docker-master:8080**

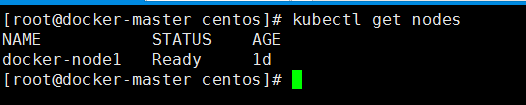
**kubectl config set-context default-context --cluster=default-cluster --user=default-admin**

**kubectl config use-context default-context**

配置完成！

在docker-master查看节点状态

#kubectl get nodes



**三 docker-master上安装网桥供容器使用**

yum install unzip wget -y

wget https://github.com/jpetazzo/pipework/archive/master.zip

unzip pipework-master.zip

mv pipework-master pipework

cp -rp pipework/pipework /usr/local/bin/

安装bridge-utils，次插件可以给容器分配固定ip地址

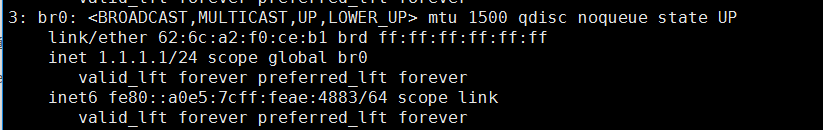
yum -y install bridge-utils

创建网络

brctl addbr br0

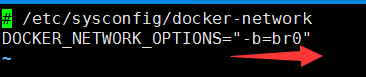
ip link set dev br0 up

ip addr add 1.1.1.0/24 dev br0



让网桥使用自定义的网络

/etc/sysconfig/docker-network



**四 集群docker-master上创建rc、pod、service**

创建rc之前先解决身份认证 否则创建pod失败

[**kubenetes无法创建pod/创建RC时无法自动创建pod的问题**](http://blog.csdn.net/jinzhencs/article/details/51435020)

**1 跳过认证**

<http://www.sunmite.com/linux/installing-kubernetes-cluster-on-centos7-to-manage-pods-and-services/>

# kubectl create -f nginx.yaml

此时有如下报错：

Error from server: error when creating "nginx.yaml": Pod "nginx" is forbidden: no API token found for service account default/default, retry after the token is automatically created and added to the service account

解决办法是编辑/etc/kubernetes/apiserver 去除 KUBE\_ADMISSION\_CONTROL中的SecurityContextDeny,ServiceAccount，并重启kube-apiserver.service服务：

#vim /etc/kubernetes/apiserver

KUBE\_ADMISSION\_CONTROL="--admission\_control=NamespaceLifecycle,NamespaceExists,LimitRanger,ResourceQuota"

#systemctl restart kube-apiserver.service

之后重新创建pod:

**2 解决认证**

出处：   
<http://stackoverflow.com/questions/34464779/pod-mysql-is-forbidden-no-api-token-found-for-service-account-default-default>

To get your setup working, you can do the same thing local-up-cluster.sh is doing:

Generate a signing key:   
openssl genrsa -out /tmp/serviceaccount.key 2048

Update /etc/kubernetes/apiserver:   
KUBE\_API\_ARGS=”–service\_account\_key\_file=/tmp/serviceaccount.key”

Update /etc/kubernetes/controller-manager:   
KUBE\_CONTROLLER\_MANAGER\_ARGS=”–service\_account\_private\_key\_file=/tmp/serviceaccount.key”

From <https://github.com/kubernetes/kubernetes/issues/11355#issuecomment-127378691>

**创建rc之后 会自动创建pod**

RC的作用是保证service的服务能力和服务质量始终处于预期的标准

RC是主要功能之一就是自动部署一个容器应用的多个副本，以及持续监控副本的数量，在集群内始终维护用户指定的副本数量

可在rc文件中手动指定pod的IP地址，如果不指定则使用docker网桥的网段

mysq-rc.yaml 文件

apiVersion: v1

kind: ReplicationController ------副本控制器RC

metadata:

name: mysql ------RC的名称 全局唯一

spec:

replicas: 1 --------Pod副本期待数量

selector:

app: mysql ------符合目标的pod拥有此标签

template: ------根据此模板创建Pod的副本（实例）

metadata:

labels:

app: mysql ------Pod副本拥有的标签，对应RC的Selector

spec:

containers: ------Pod内容器的定义部分

- name: mysql ------容器名称

images: docker.io/mysql ------容器对应的Docker Images，可以提前pull下来mysql镜像

imagePullPolicy: IfNotPresent

ports:

- containerPort: 3306 ------容器暴露的端口号

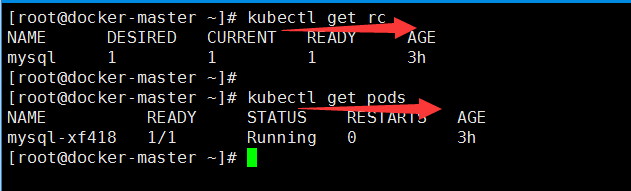
env: ------注入到容器内的环境变量

- name: MYSQL\_ROOT\_PASSWORD

value: "123456" ------此处设置mysql密码123456

#kubectl create -f mysql-rc.yaml

查看 rc和pod



然后再创建与之对应的service服务

kubectl expose rc service名称 快速创建service

可在service文件中手动指定serviceip地址，使用spec.clusterIP参数

vi mysql-svc.yaml

apiVersion: v1

kind: Service

metadata:

name: mysql ----------service全局唯一名称

spec:

type: NodePort

ports:

- port: 3306

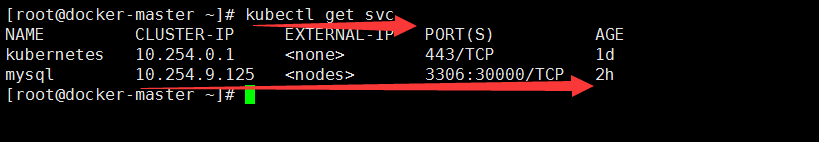
nodePort: 30000 --------------对外映射30000端口

selector:

app: mysql -------------service对应的Pod拥有这里定义的标签

#kubectl create -f mysql-rsvc.yaml

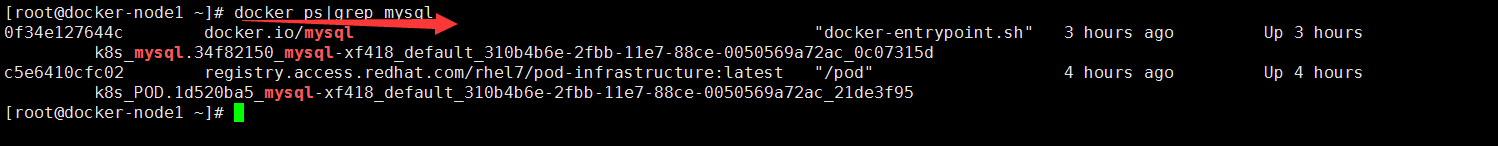
查看mysql服务



#外部访问mysql可通过mysql服务所在节点的IP地址加30000端口访问，内部可以直接使用

10.254.9.125 3306访问

在docker-node1节点上查看mysql服务进程



#kubectl describe node查看所有节点信息

http://节点IP:4194 查看节点资源使用情况

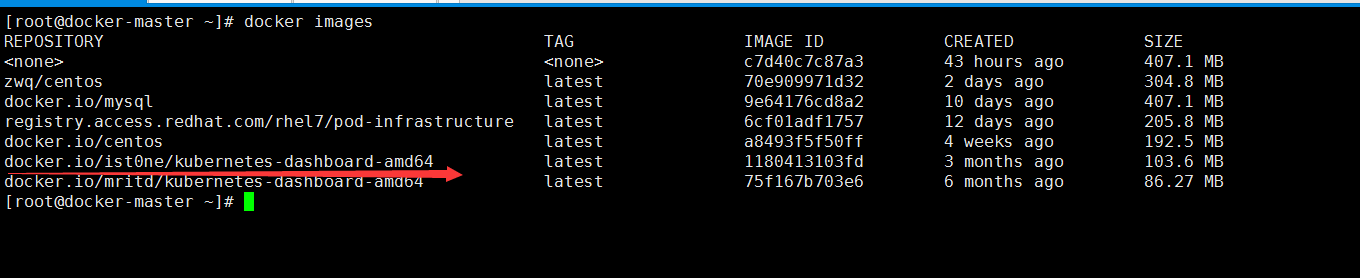


**四 k8s安装dashboard图形界面管理工具（master安装）**

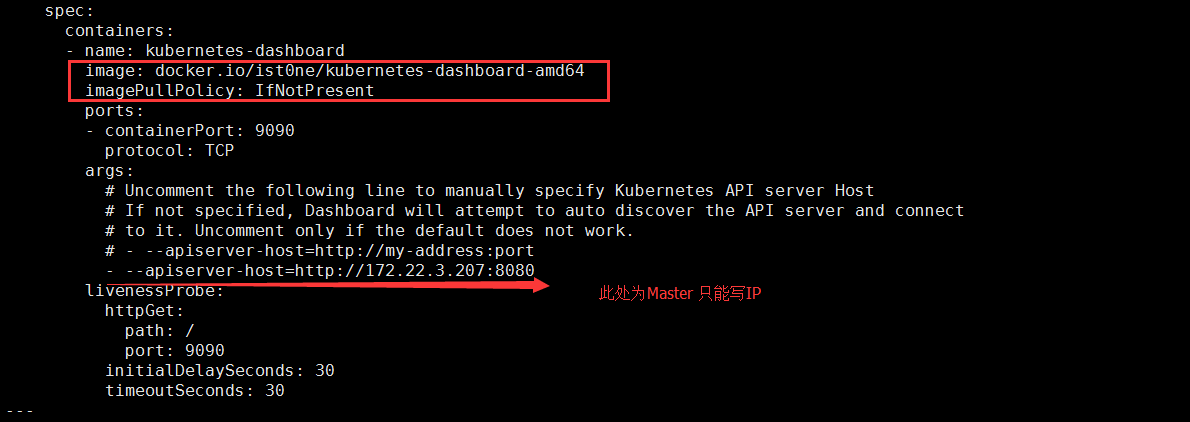
下载images镜像 以免创建过程中等待去下载image

#docker pull docker.io/ist0ne/kubernetes-dashboard-amd64 或者

# docker pull docker.io/mritd/kubernetes-dashboard-amd64

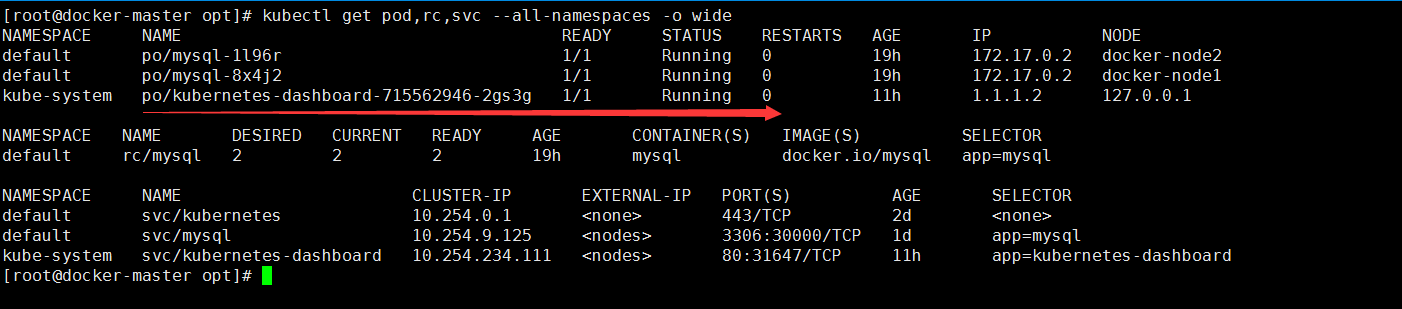


使用kubernetes-dashboard.yaml文件创建service

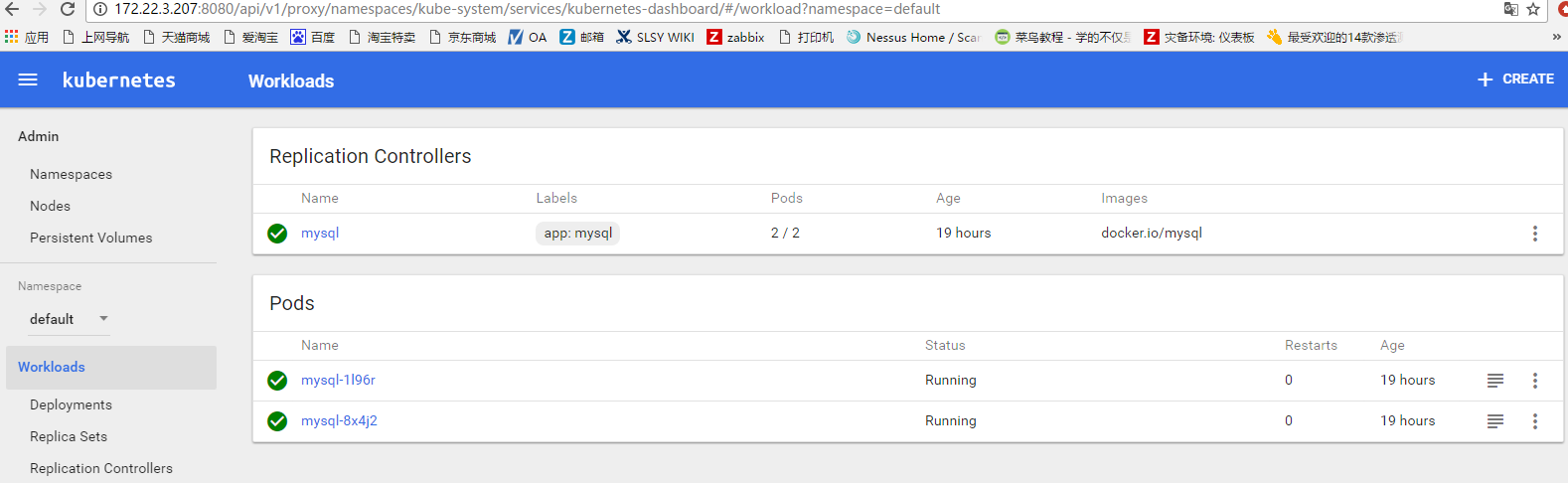


# kubectl create -f kubernetes-dashboard.yaml

查看dashboard进程

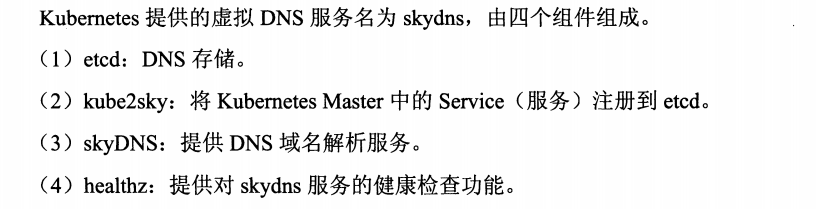


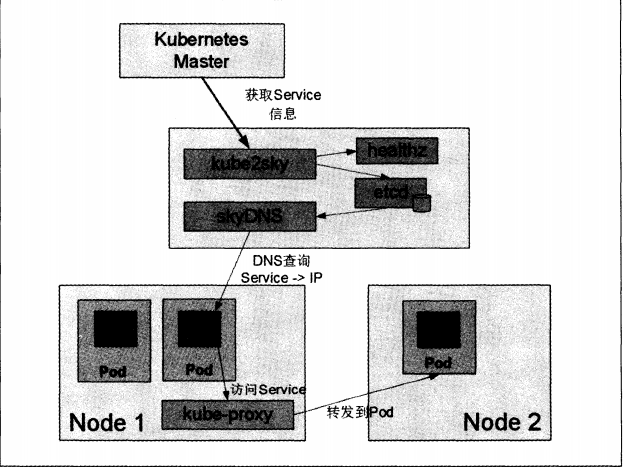
http://172.22.3.207:8080/ui访问



# kubectl delete -f kubernetes-dashboard.yaml 删除dashboard服务

**五 k8s安装DNS服务**





**参考地址：http://blog.csdn.net/luckytanggu/article/details/70807398**

**1.加速pull下载镜像**

curl -sSL https://get.daocloud.io/daotools/set\_mirror.sh | sh -s http://d3989bee.m.daocloud.ioCopy

该脚本可以将 --registry-mirror 加入到你的 Docker 配置文件 /etc/default/docker 中。适用于 Ubuntu14.04、Debian、CentOS6 、CentOS7、Fedora、Arch Linux、openSUSE Leap 42.1，其他版本可能有细微不同。更多详情请[访问文档](http://guide.daocloud.io/dcs/daocloud-9153151.html)。