Kubernetes（k8s）

环境：centos7.4虚拟机5台，搭建好yum，关闭防火墙、SELinux，清空iptables规则

Yum源包括系统源和extras源

架构：

Node01 192.168.1.1 repo docker-distribution

Node10 192.168.1.10 kube\_master api scheduler controller etcd

Node11 192.168.1.11 kube\_node docker kubelet kube\_proxy..

Node12 192.168.1.12 kube\_node docker kubelet kube\_proxy..

Node13 192.168.1.13 kube\_node docker kubelet kube\_proxy..

#环境准备

[root@hostos ~]# virsh list --all

Id 名称 状态

----------------------------------------------------

5 node01 running

6 node10 running

7 node11 running

8 node12 running

9 node13 running

#构建repo

[root@repo ~]# ls

my\_imgs.tar.gz

[root@repo ~]# yum -y install docker docker-distribution

[root@repo ~]# vim /lib/systemd/system/docker.service

[root@repo ~]# sed -rn '30,31p' /lib/systemd/system/docker.service

ExecStartPost=/sbin/iptables -P FORWARD ACCEPT

ExecReload=/bin/kill -s HUP $MAINPID

[root@repo ~]# vim /etc/sysconfig/docker

[root@repo ~]# sed -rn '/5000/p' /etc/sysconfig/docker

ADD\_REGISTRY='--add-registry 192.168.1.1:5000'

INSECURE\_REGISTRY='--insecure-registry 192.168.1.1:5000'

[root@repo ~]# systemctl start docker-distribution.service

[root@repo ~]# systemctl enable docker-distribution.service

Created symlink from /etc/systemd/system/multi-user.target.wants/docker-distribution.service to /usr/lib/systemd/system/docker-distribution.service.

[root@repo ~]# systemctl enable docker

Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.

[root@repo ~]# systemctl start docker

[root@repo ~]# tar -xf my\_imgs.tar.gz

[root@repo ~]# curl 192.168.1.1:5000/v2/

{}[root@repo ~]# ls

centos.tar httpd.tar my\_imgs.tar.gz myos.tar nginx.tar

[root@repo ~]# for i in \*.tar

> do

> docker load -i $i

> done

[root@repo ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

myos nginx d83911892df2 2 days ago 394.7 MB

myos httpd 9932e1264487 4 days ago 316.5 MB

myos v2 4c1e5ce2f7eb 4 days ago 286.9 MB

docker.io/centos latest 76d6bc25b8a5 19 months ago 199.7 MB

[root@repo ~]# docker tag myos:v2 myos:latest

[root@repo ~]# docker rmi myos:v2

Untagged: myos:v2

[root@repo ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

myos nginx d83911892df2 2 days ago 394.7 MB

myos httpd 9932e1264487 4 days ago 316.5 MB

myos latest 4c1e5ce2f7eb 4 days ago 286.9 MB

docker.io/centos latest 76d6bc25b8a5 19 months ago 199.7 MB

[root@repo ~]# docker push myos:latest

The push refers to a repository [192.168.1.1:5000/myos]

fab4c623fcda: Pushed

4bf0f90eac8f: Pushed

bcc97fbfc9e1: Pushed

latest: digest: sha256:9f6f0a340d491fada4642bb6fe13a00fd181f68e1db8ef74cd871719bb266656 size: 948

[root@repo ~]# docker push myos:httpd

The push refers to a repository [192.168.1.1:5000/myos]

9141367f0f49: Pushed

96eabedd4b95: Pushed

7a04260e2e6e: Pushed

bcc97fbfc9e1: Layer already exists

httpd: digest: sha256:c406f2b788b282fc725a6aee5498894fac8f4e49371499717985bb28c37d7cf7 size: 1155

[root@repo ~]# docker push myos:nginx

The push refers to a repository [192.168.1.1:5000/myos]

9d1c47bd36c8: Pushed

70fb7813b7ad: Pushed

e57e86839b7e: Pushed

bea235577464: Pushed

bcc97fbfc9e1: Layer already exists

nginx: digest: sha256:d8df381424740685375fd6d9cea350f75337fe7a5d8f73c9cbeeab69e2b94656 size: 1366

[root@repo ~]#

[root@repo ~]# curl 192.168.1.1:5000/v2/\_catalog

{"repositories":["myos"]}

[root@repo ~]# curl 192.168.1.1:5000/v2/myos/tags/list

{"name":"myos","tags":["latest","httpd","nginx"]}

[root@repo ~]#

##################################################################

#node节点安装docker服务

[root@repo ~]# for i in 192.168.1.{10..13}

> do

> scp /etc/sysctl.conf $i:/etc/

> ssh $i sysctl -p

> ssh $i hostname

> ssh $i cat /proc/sys/net/ipv4/ip\_forward

> done

Warning: Permanently added '192.168.1.10' (ECDSA) to the list of known hosts.

sysctl.conf 100% 471 273.7KB/s 00:00

net.ipv4.ip\_forward = 1

kube\_master

1

sysctl.conf 100% 471 30.1KB/s 00:00

net.ipv4.ip\_forward = 1

kube\_node01

1

sysctl.conf 100% 471 278.4KB/s 00:00

net.ipv4.ip\_forward = 1

kube\_node02

1

sysctl.conf 100% 471 417.0KB/s 00:00

net.ipv4.ip\_forward = 1

kube\_node03

1

[root@repo ~]#

[root@repo ~]# for i in 192.168.1.{11..13}

>do

>ssh $i yum -y install docker; scp /lib/systemd/system/docker.service $i:/lib/systemd/system/

>scp /etc/sysconfig/docker $i:/etc/sysconfig/

>ssh $i systemctl enable docker

>ssh $i systemctl daemon-reload;

>ssh $i systemctl start docker

>done

[root@repo ~]# for i in 192.168.1.{11..13}

> do

> ssh $i hostname

> ssh $i systemctl is-active docker

> done

kube\_node01

active

kube\_node02

active

kube\_node03

active

[root@repo ~]#

##################################################################

#制作k8s软件源

[root@hostos ~]# mkdir /var/ftp/localrepo

[root@hostos ~]# unzip /data/soft/architecture/k8s/kubernetes.zip -d /var/ftp/localrepo/

Archive: /data/soft/architecture/k8s/kubernetes.zip

creating: /var/ftp/localrepo/kubernetes/

inflating: /var/ftp/localrepo/kubernetes/containernetworking-cni-0.6.0-3.el7.x86\_64.rpm

inflating: /var/ftp/localrepo/kubernetes/etcd-3.3.11-2.el7.centos.x86\_64.rpm

inflating: /var/ftp/localrepo/kubernetes/flannel-0.7.1-4.el7.x86\_64.rpm

inflating: /var/ftp/localrepo/kubernetes/kubernetes-1.10.3-0.el7.x86\_64.rpm

inflating: /var/ftp/localrepo/kubernetes/kubernetes-client-1.10.3-0.el7.x86\_64.rpm

inflating: /var/ftp/localrepo/kubernetes/kubernetes-kubeadm-1.10.3-0.el7.x86\_64.rpm

inflating: /var/ftp/localrepo/kubernetes/kubernetes-master-1.10.3-0.el7.x86\_64.rpm

inflating: /var/ftp/localrepo/kubernetes/kubernetes-node-1.10.3-0.el7.x86\_64.rpm

[root@hostos ~]# ls /var/ftp/localrepo/

kubernetes

[root@hostos ~]# cd /var/ftp/localrepo/

[root@hostos localrepo]# createrepo .

Spawning worker 0 with 1 pkgs

Spawning worker 1 with 1 pkgs

Spawning worker 2 with 1 pkgs

Spawning worker 3 with 1 pkgs

Spawning worker 4 with 1 pkgs

Spawning worker 5 with 1 pkgs

Spawning worker 6 with 1 pkgs

Spawning worker 7 with 1 pkgs

Workers Finished

Saving Primary metadata

Saving file lists metadata

Saving other metadata

Generating sqlite DBs

Sqlite DBs complete

[root@hostos localrepo]# ls

kubernetes repodata

[root@hostos localrepo]# cd

[root@hostos ~]#

[root@kube\_master ~]# vim /etc/yum.repos.d/development.repo

[root@kube\_master ~]# cat /etc/yum.repos.d/development.repo

[development]

name=dvd

baseurl=ftp://192.168.1.254/iso/centos/

enabled=1

gpgcheck=1

[extras]

name=extras

baseurl=ftp://192.168.1.254/iso/extras/

enabled=1

gpgcheck=0

[localrepo]

name=localrepo

baseurl=ftp://192.168.1.254/localrepo/

enabled=1

gpgcheck=0

[root@kube\_master ~]# yum clean all; yum repolist

repolist: 9,995

[root@kube\_master ~]#

[root@kube\_master ~]# for i in 192.168.1.{11..13}

> do

> scp /etc/yum.repos.d/development.repo $i:/etc/yum.repos.d/

> ssh $i yum clean all &> /dev/null

> ssh $i yum repolist

> done

#所有机器均添加localrepo源，共计8个软件包

###################################################################部署etcd服务

[root@kube\_master ~]# yum -y install etcd

[root@kube\_master ~]# vim /etc/etcd/etcd.conf

[root@kube\_master ~]# sed -rn '6p' /etc/etcd/etcd.conf

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

[root@kube\_master ~]# systemctl enable etcd.service --now

[root@kube\_master ~]# systemctl is-active etcd.service

active

[root@kube\_master ~]#

#操作etcd数据库

[root@kube\_master ~]# etcdctl ls /

[root@kube\_master ~]# etcdctl mkdir /abc

[root@kube\_master ~]# etcdctl ls /

/abc

[root@kube\_master ~]# etcdctl rmdir /abc

[root@kube\_master ~]# etcdctl ls /

[root@kube\_master ~]#

[root@kube\_master ~]# etcdctl mk /atomic.io/network/config '{"Network":"10.254.0.0/16","Backend":{"Type":"vxlan"}}'

{"Network":"10.254.0.0/16","Backend":{"Type":"vxlan"}}

[root@kube\_master ~]# etcdctl get /atomic.io/network/config

{"Network":"10.254.0.0/16","Backend":{"Type":"vxlan"}}

[root@kube\_master ~]#

###################################################################安装配置flannel

[root@kube\_master ~]# yum -y install flannel.x86\_64

[root@kube\_master ~]# sed -ri '/127.0.0.1/s/127.0.0.1/192.168.1.10/' /etc/sysconfig/flanneld

[root@kube\_master ~]# sed -rn '/192/p' /etc/sysconfig/flanneld

FLANNEL\_ETCD\_ENDPOINTS="http://192.168.1.10:2379"

[root@kube\_master ~]# systemctl enable flanneld.service

Created symlink from /etc/systemd/system/multi-user.target.wants/flanneld.service to /usr/lib/systemd/system/flanneld.service.

Created symlink from /etc/systemd/system/docker.service.wants/flanneld.service to /usr/lib/systemd/system/flanneld.service.

[root@kube\_master ~]# systemctl start flanneld.service

[root@kube\_master ~]# ifconfig | grep 10.254

inet 10.254.53.0 netmask 255.255.255.255 broadcast 0.0.0.0

[root@kube\_master ~]#

[root@kube\_master ~]# for i in 192.168.1.{11..13}

> do

> ssh $i yum -y install flannel

> scp /etc/sysconfig/flanneld $i:/etc/sysconfig/

> ssh $i systemctl enable flanneld

> ssh $i systemctl stop docker

> ssh $i systemctl start flanneld docker

> done

[root@kube\_master ~]# for i in 192.168.1.{11..13}

> do

> ssh $i hostname

> ssh $i ifconfig | grep "10.254"

> done

kube\_node01

inet 10.254.29.1 netmask 255.255.255.0 broadcast 0.0.0.0

inet 10.254.29.0 netmask 255.255.255.255 broadcast 0.0.0.0

kube\_node02

inet 10.254.100.1 netmask 255.255.255.0 broadcast 0.0.0.0

inet 10.254.100.0 netmask 255.255.255.255 broadcast 0.0.0.0

kube\_node03

inet 10.254.46.1 netmask 255.255.255.0 broadcast 0.0.0.0

inet 10.254.46.0 netmask 255.255.255.255 broadcast 0.0.0.0

[root@kube\_master ~]#

###################################################################测试节点之间容器通信

[root@kube\_node01 ~]# docker run -it myos:latest

Unable to find image 'myos:latest' locally

Trying to pull repository 192.168.1.1:5000/myos ...

latest: Pulling from 192.168.1.1:5000/myos

0ffa5ac9f3c5: Pull complete

89bb679e5804: Pull complete

e17c6d329068: Pull complete

Digest: sha256:9f6f0a340d491fada4642bb6fe13a00fd181f68e1db8ef74cd871719bb266656

[root@17afeae1c901 /]# ifconfig | head -2

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1450

inet 10.254.29.2 netmask 255.255.255.0 broadcast 0.0.0.0

[root@kube\_node03 ~]# docker run -it myos:latest

Unable to find image 'myos:latest' locally

Trying to pull repository 192.168.1.1:5000/myos ...

latest: Pulling from 192.168.1.1:5000/myos

0ffa5ac9f3c5: Pull complete

89bb679e5804: Pull complete

e17c6d329068: Pull complete

Digest: sha256:9f6f0a340d491fada4642bb6fe13a00fd181f68e1db8ef74cd871719bb266656

[root@f2cffa1b44cd /]# ifconfig | head -2

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1450

inet 10.254.46.2 netmask 255.255.255.0 broadcast 0.0.0.0

[root@f2cffa1b44cd /]#

[root@17afeae1c901 /]# ping 10.254.46.2

PING 10.254.46.2 (10.254.46.2) 56(84) bytes of data.

64 bytes from 10.254.46.2: icmp\_seq=1 ttl=62 time=1.62 ms

64 bytes from 10.254.46.2: icmp\_seq=2 ttl=62 time=0.451 ms

^C

--- 10.254.46.2 ping statistics ---

2 packets transmitted, 2 received, 0% packet loss, time 1002ms

rtt min/avg/max/mdev = 0.451/1.037/1.623/0.586 ms

[root@17afeae1c901 /]#

##################################################################

#配置kube\_master

[root@kube\_master ~]# yum -y install kubernetes-master.x86\_64 kubernetes-client.x86\_64

[root@kube\_master ~]# cd /etc/kubernetes/

[root@kube\_master kubernetes]# ls

apiserver config controller-manager scheduler

[root@kube\_master kubernetes]# vim config

[root@kube\_master kubernetes]# vim apiserver

[root@kube\_master kubernetes]# cat config | grep -v ^$ | grep -v ^#

KUBE\_LOGTOSTDERR="--logtostderr=true"

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

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

KUBE\_MASTER="--master=http://192.168.1.10:8080"

[root@kube\_master kubernetes]# cat apiserver | grep -v ^$ | grep -v ^#

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

KUBE\_ETCD\_SERVERS="--etcd-servers=http://192.168.1.10:2379"

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

KUBE\_ADMISSION\_CONTROL="--admission-control=NamespaceLifecycle,LimitRanger,SecurityContextDeny,ResourceQuota"

KUBE\_API\_ARGS=""

[root@kube\_master kubernetes]#

[root@kube\_master ~]# systemctl is-active etcd.service

active

[root@kube\_master ~]# systemctl enable kube-apiserver.service kube-controller-manager.service kube-scheduler.service

Created symlink from /etc/systemd/system/multi-user.target.wants/kube-apiserver.service to /usr/lib/systemd/system/kube-apiserver.service.

Created symlink from /etc/systemd/system/multi-user.target.wants/kube-controller-manager.service to /usr/lib/systemd/system/kube-controller-manager.service.

Created symlink from /etc/systemd/system/multi-user.target.wants/kube-scheduler.service to /usr/lib/systemd/system/kube-scheduler.service.

[root@kube\_master ~]# systemctl start kube-apiserver.service kube-controller-manager.service kube-scheduler.service

[root@kube\_master ~]# systemctl is-active kube-apiserver.service kube-controller-manager.service kube-scheduler.service

active

active

active

[root@kube\_master ~]#

[root@kube\_master ~]# kubectl get cs

NAME STATUS MESSAGE ERROR

scheduler Healthy ok

etcd-0 Healthy {"health":"true"}

controller-manager Healthy ok

[root@kube\_master ~]# kubectl get node

No resources found.

[root@kube\_master ~]#

###################################################################防止网络通信故障，添加内部hosts解析

[root@repo ~]# vim /etc/hosts

[root@repo ~]# tail -5 /etc/hosts

192.168.1.1 repo

192.168.1.10 kube\_master kube-master

192.168.1.11 kube\_node01 kube-node01

192.168.1.12 kube\_node02 kube-node02

192.168.1.13 kube\_node03 kube-node03

[root@repo ~]# for i in 192.168.1.{10..13}

> do

> scp /etc/hosts $i:/etc/

> done

hosts 100% 292 421.1KB/s 00:00

hosts 100% 292 456.3KB/s 00:00

hosts 100% 292 560.0KB/s 00:00

hosts 100% 292 634.4KB/s 00:00

[root@repo ~]#

#导入pod镜像到repo

[root@repo ~]# ls pod-infrastructure.tar

pod-infrastructure.tar

[root@repo ~]# docker load -i pod-infrastructure.tar

c1eac31e742f: Loading layer [==================================================>] 205.9 MB/205.9 MB

9161a60cc964: Loading layer [==================================================>] 10.24 kB/10.24 kB

6872307367a6: Loading layer [==================================================>] 12.74 MB/12.74 MB

Loaded image: pod-infrastructure:latest ] 131.1 kB/12.74 MB

[root@repo ~]# docker images | grep pod

pod-infrastructure latest 99965fb98423 2 years ago 208.6 MB

[root@repo ~]# docker push pod-infrastructure:latest

The push refers to a repository [192.168.1.1:5000/pod-infrastructure]

6872307367a6: Pushed

9161a60cc964: Pushed

c1eac31e742f: Pushed

latest: digest: sha256:60b52a2ba3d2d11e6639d747dc9799e88cd2a6df1a0eab0fcbd6e63aa27e554e size: 948

[root@repo ~]# curl localhost:5000/v2/\_catalog

{"repositories":["myos","pod-infrastructure"]}

[root@repo ~]# curl localhost:5000/v2/pod-infrastructure/tags/list

{"name":"pod-infrastructure","tags":["latest"]}

[root@repo ~]#

###################################################################部署kubernetes-node

[root@kube\_node01 ~]# yum -y install kubernetes-node.x86\_64

[root@kube\_node01 ~]# cd /etc/kubernetes/

[root@kube\_node01 kubernetes]# ls

config kubelet proxy

[root@kube\_node01 kubernetes]# vim config

[root@kube\_node01 kubernetes]# vim kubelet

[root@kube\_node01 kubernetes]# vim kubelet.kubeconfig

[root@kube\_node01 kubernetes]# cat config | grep -v ^$ | grep -v ^#

KUBE\_LOGTOSTDERR="--logtostderr=true"

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

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

KUBE\_MASTER="--master=http://192.168.1.10:8080"

[root@kube\_node01 kubernetes]# cat kubelet | grep -v ^$ | grep -v ^#

KUBELET\_ADDRESS="--address=0.0.0.0"

KUBELET\_HOSTNAME="--hostname-override=kube-node01"

KUBELET\_ARGS="--cgroup-driver=systemd --fail-swap-on=false --kubeconfig=/etc/kubernetes/kubelet.kubeconfig --pod-infra-container-image=192.168.1.1:5000/pod-infrastructure:latest"

#此处踩坑一个，hostname中不能出现\_，识别过程中识别[a-z0-9]([-a-z0-9]\*[a-z0-9])?(\.[a-z0-9]([-a-z0-9]\*[a-z0-9])?)\*

[root@kube\_node01 kubernetes]# cat kubelet.kubeconfig

kind: Config

clusters:

- cluster:

server: http://192.168.1.10:8080

name: local

contexts:

- context:

cluster: local

name: local

current-context: local

[root@kube\_node01 kubernetes]# systemctl enable kubelet.service kube-proxy.service

Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to /usr/lib/systemd/system/kubelet.service.

Created symlink from /etc/systemd/system/multi-user.target.wants/kube-proxy.service to /usr/lib/systemd/system/kube-proxy.service.

[root@kube\_node01 kubernetes]# systemctl start kubelet.service kube-proxy.service

[root@kube\_node01 kubernetes]# systemctl is-active kubelet.service kube-proxy.service

active

active

[root@kube\_node01 kubernetes]# cd

[root@kube\_node01 ~]#

[root@kube\_node02 ~]# yum -y install kubernetes-node.x86\_64

[root@kube\_node02 ~]# scp 192.168.1.11:/etc/kubernetes/\* /etc/kubernetes/

Warning: Permanently added '192.168.1.11' (ECDSA) to the list of known hosts.

config 100% 658 453.8KB/s 00:00

kubelet 100% 542 593.3KB/s 00:00

kubelet.kubeconfig 100% 166 183.5KB/s 00:00

proxy 100% 103 194.8KB/s 00:00

[root@kube\_node02 ~]# sed -ri '/node01/s/node01/node02/' /etc/kubernetes/kubelet

[root@kube\_node02 ~]# sed -rn '/node02/p' /etc/kubernetes/kubelet

KUBELET\_HOSTNAME="--hostname-override=kube-node02"

[root@kube\_node02 ~]# systemctl enable kubelet.service kube-proxy.service

Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to /usr/lib/systemd/system/kubelet.service.

Created symlink from /etc/systemd/system/multi-user.target.wants/kube-proxy.service to /usr/lib/systemd/system/kube-proxy.service.

[root@kube\_node02 ~]# systemctl start kubelet.service kube-proxy.service

[root@kube\_node02 ~]# systemctl is-active kubelet.service kube-proxy.service

active

active

[root@kube\_node02 ~]#

[root@kube\_node03 ~]# yum -y install kubernetes-node.x86\_64

[root@kube\_node03 ~]# scp 192.168.1.11:/etc/kubernetes/\* /etc/kubernetes/

Warning: Permanently added '192.168.1.11' (ECDSA) to the list of known hosts.

config 100% 658 590.6KB/s 00:00

kubelet 100% 542 671.6KB/s 00:00

kubelet.kubeconfig 100% 166 175.1KB/s 00:00

proxy 100% 103 212.4KB/s 00:00

[root@kube\_node03 ~]# sed -ri '/node01/s/node01/node03/' /etc/kubernetes/kubelet

[root@kube\_node03 ~]# sed -rn '/node03/p' /etc/kubernetes/kubelet

KUBELET\_HOSTNAME="--hostname-override=kube-node03"

[root@kube\_node03 ~]# systemctl enable kubelet.service kube-proxy.service

Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to /usr/lib/systemd/system/kubelet.service.

Created symlink from /etc/systemd/system/multi-user.target.wants/kube-proxy.service to /usr/lib/systemd/system/kube-proxy.service.

[root@kube\_node03 ~]# systemctl start kubelet.service kube-proxy.service

[root@kube\_node03 ~]# systemctl is-active kubelet.service kube-proxy.service

active

active

[root@kube\_node03 ~]#

#测试node节点是否正常

[root@kube\_master ~]# kubectl get node

NAME STATUS ROLES AGE VERSION

kube-node01 Ready <none> 10m v1.10.3

kube-node02 Ready <none> 4m v1.10.3

kube-node03 Ready <none> 38s v1.10.3

[root@kube\_master ~]#

###################################################################部署kube-dashboard

[root@repo ~]# docker load -i busybox.tar

a6d503001157: Loading layer [==================================================>] 1.437 MB/1.437 MB

Loaded image: busybox:latest[=> ] 32.77 kB/1.437 MB

[root@repo ~]# docker push busybox:latest

The push refers to a repository [192.168.1.1:5000/busybox]

a6d503001157: Pushed

latest: digest: sha256:43d5f7ca6c4c4e2e8fa5007494c676dc1c73c6726e4529315b7cd7a7cec79464 size: 527

[root@repo ~]# docker load -i kubernetes-dashboard.tar

23ddb8cbb75a: Loading layer [==================================================>] 102.8 MB/102.8 MB

Loaded image: kubernetes-dashboard-amd64:v1.8.3 ] 557.1 kB/102.8 MB

[root@repo ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

myos nginx d83911892df2 3 days ago 394.7 MB

myos httpd 9932e1264487 5 days ago 316.5 MB

myos latest 4c1e5ce2f7eb 5 days ago 286.9 MB

busybox latest 83aa35aa1c79 10 weeks ago 1.22 MB

docker.io/centos latest 76d6bc25b8a5 19 months ago 199.7 MB

kubernetes-dashboard-amd64 v1.8.3 0c60bcf89900 2 years ago 102.3 MB

pod-infrastructure latest 99965fb98423 2 years ago 208.6 MB

[root@repo ~]# docker push busybox:latest

The push refers to a repository [192.168.1.1:5000/busybox]

a6d503001157: Layer already exists

latest: digest: sha256:43d5f7ca6c4c4e2e8fa5007494c676dc1c73c6726e4529315b7cd7a7cec79464 size: 527

[root@repo ~]# docker push kubernetes-dashboard-amd64:v1.8.3

The push refers to a repository [192.168.1.1:5000/kubernetes-dashboard-amd64]

23ddb8cbb75a: Pushed

v1.8.3: digest: sha256:dc4026c1b595435ef5527ca598e1e9c4343076926d7d62b365c44831395adbd0 size: 529

[root@repo ~]# curl localhost:5000/v2/\_catalog

{"repositories":["busybox","kubernetes-dashboard-amd64","myos","pod-infrastructure"]}

[root@repo ~]# curl localhost:5000/v2/busybox/tags/list

{"name":"busybox","tags":["latest"]}

[root@repo ~]# curl localhost:5000/v2/myos/tags/list

{"name":"myos","tags":["latest","httpd","nginx"]}

[root@repo ~]# curl localhost:5000/v2/kubernetes-dashboard-amd64/tags/list

{"name":"kubernetes-dashboard-amd64","tags":["v1.8.3"]}

[root@repo ~]# curl localhost:5000/v2/pod-infrastructure/tags/list

{"name":"pod-infrastructure","tags":["latest"]}

[root@repo ~]#

#导入所有docker镜像，共计6个

###################################################################启动kube-dashboard

[root@kube\_master ~]# vim kube-dashboard.yaml

[root@kube\_master ~]# cat kube-dashboard.yaml | grep 192.168

image: 192.168.1.1:5000/kubernetes-dashboard-amd64:v1.8.3

- --apiserver-host=http://192.168.1.10:8080 ###修改为Master的IP

[root@kube\_master ~]# kubectl create -f kube-dashboard.yaml

deployment.apps "kubernetes-dashboard" created

service "kubernetes-dashboard" created

[root@kube\_master ~]# for i in 192.168.1.{11..13}

> do

> ssh $i hostname

> ssh $i docker ps

> done

kube\_node01

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

kube\_node02

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

d8f0aa413612 0c60bcf89900 "/dashboard --insecur" 22 seconds ago Up 17 seconds k8s\_kubernetes-dashboard\_kubernetes-dashboard-58fc4fd9b7-jpwtk\_kube-system\_54c4d744-9b24-11ea-8fdd-52540078c20b\_0

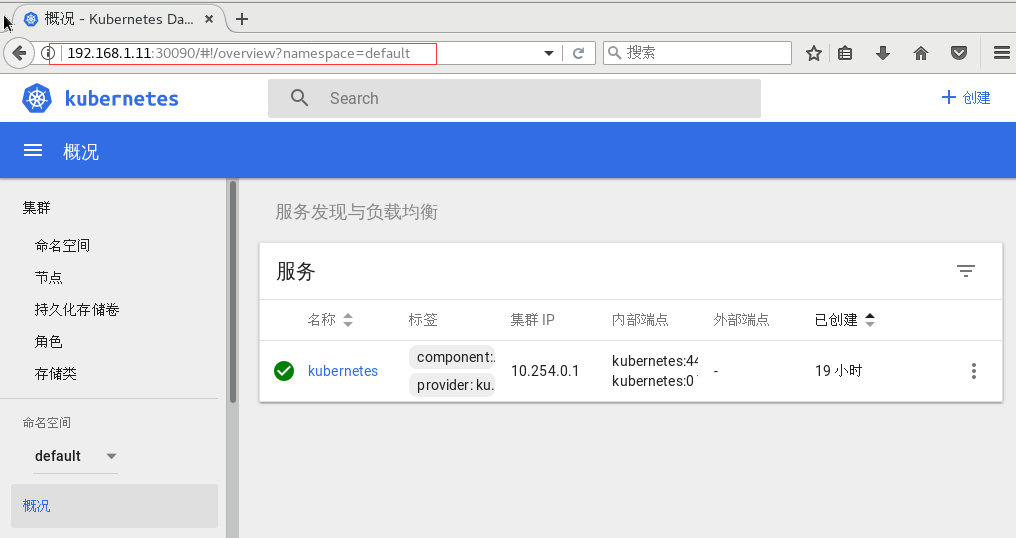
87635b993baa 192.168.1.1:5000/pod-infrastructure:latest "/usr/bin/pod" 26 seconds ago Up 22 seconds k8s\_POD\_kubernetes-dashboard-58fc4fd9b7-jpwtk\_kube-system\_54c4d744-9b24-11ea-8fdd-52540078c20b\_0

kube\_node03

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

[root@kube\_master ~]#

#测试访问任意node节点的30090端口



####################################################################################################################################

#使用kubectl命令

[root@kube\_master ~]# kubectl get node

NAME STATUS ROLES AGE VERSION

kube-node01 Ready <none> 4h v1.10.3

kube-node02 Ready <none> 4h v1.10.3

kube-node03 Ready <none> 4h v1.10.3

[root@kube\_master ~]# kubectl get pod

No resources found.

##########

#创建容器

[root@kube\_master ~]# kubectl run haha -i -t --image=192.168.1.1:5000/myos:latest

If you don't see a command prompt, try pressing enter.

[root@haha-6df4f7f6d7-8rw4t /]# ls

bin dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

[root@haha-6df4f7f6d7-8rw4t /]# ifconfig | head -2

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1450

inet 10.254.29.2 netmask 255.255.255.0 broadcast 0.0.0.0

[root@haha-6df4f7f6d7-8rw4t /]# Session ended, resume using 'kubectl attach haha-6df4f7f6d7-8rw4t -c haha -i -t' command when the pod is running

[root@kube\_master ~]#

#按ctrl+p+q退出，相当于attach

#########

#进入容器

[root@kube\_master ~]# kubectl exec -it haha-6df4f7f6d7-8rw4t /bin/bash

[root@haha-6df4f7f6d7-8rw4t /]# pstree

bash

[root@haha-6df4f7f6d7-8rw4t /]# pstree -p

bash(1)

[root@haha-6df4f7f6d7-8rw4t /]# exit

exit

[root@kube\_master ~]#

#推荐使用exec方式，对应docker命令

[root@kube\_master ~]# kubectl get pod

NAME READY STATUS RESTARTS AGE

haha-6df4f7f6d7-8rw4t 1/1 Running 0 7m

[root@kube\_master ~]# kubectl get replicaset

NAME DESIRED CURRENT READY AGE

haha-6df4f7f6d7 1 1 1 7m

[root@kube\_master ~]# kubectl get deployment

NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE

haha 1 1 1 1 7m

[root@kube\_master ~]#

######################  
#查看资源详细信息，用于排错

[root@kube\_master ~]# kubectl get deployment

NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE

haha 1 1 1 1 11m

[root@kube\_master ~]# kubectl describe deployment haha

Name: haha

Namespace: default

CreationTimestamp: Thu, 21 May 2020 15:34:05 +0800

Labels: run=haha

Annotations: deployment.kubernetes.io/revision=1

Selector: run=haha

Replicas: 1 desired | 1 updated | 1 total | 1 available | 0 unavailable

StrategyType: RollingUpdate

MinReadySeconds: 0

RollingUpdateStrategy: 1 max unavailable, 1 max surge

Pod Template:

Labels: run=haha

Containers:

haha:

Image: 192.168.1.1:5000/myos:latest

Port: <none>

Host Port: <none>

Environment: <none>

Mounts: <none>

Volumes: <none>

Conditions:

Type Status Reason

---- ------ ------

Available True MinimumReplicasAvailable

Progressing True NewReplicaSetAvailable

OldReplicaSets: <none>

NewReplicaSet: haha-6df4f7f6d7 (1/1 replicas created)

Events:

Type Reason Age From Message

---- ------ ---- ---- -------

Normal ScalingReplicaSet 11m deployment-controller Scaled up replica set haha-6df4f7f6d7 to 1

[root@kube\_master ~]# kubectl get pod

NAME READY STATUS RESTARTS AGE

haha-6df4f7f6d7-8rw4t 1/1 Running 0 11m

[root@kube\_master ~]# kubectl describe pod haha-6df4f7f6d7-8rw4t

Name: haha-6df4f7f6d7-8rw4t

Namespace: default

Node: kube-node01/192.168.1.11

Start Time: Thu, 21 May 2020 15:34:06 +0800

Labels: pod-template-hash=2890939283

run=haha

Annotations: <none>

Status: Running

IP: 10.254.29.2

Controlled By: ReplicaSet/haha-6df4f7f6d7

Containers:

haha:

Container ID: docker://a5ad9607eb587655172c42f3a4e8f526b6e245c3f3709e4ee04ec7e748566c60

Image: 192.168.1.1:5000/myos:latest

Image ID: docker-pullable://192.168.1.1:5000/myos@sha256:9f6f0a340d491fada4642bb6fe13a00fd181f68e1db8ef74cd871719bb266656

Port: <none>

Host Port: <none>

State: Running

Started: Thu, 21 May 2020 15:34:18 +0800

Ready: True

Restart Count: 0

Environment: <none>

Mounts: <none>

Conditions:

Type Status

Initialized True

Ready True

PodScheduled True

Volumes: <none>

QoS Class: BestEffort

Node-Selectors: <none>

Tolerations: <none>

Events:

Type Reason Age From Message

---- ------ ---- ---- -------

Normal Scheduled 12m default-scheduler Successfully assigned haha-6df4f7f6d7-8rw4t to kube-node01

Normal Pulling 11m kubelet, kube-node01 pulling image "192.168.1.1:5000/myos:latest"

Normal Pulled 11m kubelet, kube-node01 Successfully pulled image "192.168.1.1:5000/myos:latest"

Normal Created 11m kubelet, kube-node01 Created container

Normal Started 11m kubelet, kube-node01 Started container

Warning MissingClusterDNS 36s (x12 over 12m) kubelet, kube-node01 pod: "haha-6df4f7f6d7-8rw4t\_default(7360b804-9b35-11ea-8fdd-52540078c20b)". kubelet does not have ClusterDNS IP configured and cannot create Pod using "ClusterFirst" policy. Falling back to "Default" policy.

[root@kube\_master ~]#

##############

#查看容器内部执行的历史命令

[root@kube\_master ~]# kubectl logs haha-6df4f7f6d7-8rw4t

[root@haha-6df4f7f6d7-8rw4t /]# ls

bin dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

[root@haha-6df4f7f6d7-8rw4t /]# ifconfig | head -2

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1450

inet 10.254.29.2 netmask 255.255.255.0 broadcast 0.0.0.0

[root@kube\_master ~]#

###################

#删除资源，资源层次deployment>replicaset>pod，所以需要删除顶层资源

[root@kube\_master ~]# kubectl get deployment

NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE

haha 1 1 1 1 30m

[root@kube\_master ~]# kubectl get replicaset

NAME DESIRED CURRENT READY AGE

haha-6df4f7f6d7 1 1 1 30m

[root@kube\_master ~]# kubectl get pod

NAME READY STATUS RESTARTS AGE

haha-6df4f7f6d7-8rw4t 1/1 Running 0 30m

[root@kube\_master ~]# kubectl get pod -o wide

NAME READY STATUS RESTARTS AGE IP NODE

haha-6df4f7f6d7-8rw4t 1/1 Running 0 30m 10.254.29.2 kube-node01

[root@kube\_master ~]# kubectl delete pod haha-6df4f7f6d7-8rw4t

pod "haha-6df4f7f6d7-8rw4t" deleted

[root@kube\_master ~]# kubectl get pod -o wide

NAME READY STATUS RESTARTS AGE IP NODE

haha-6df4f7f6d7-8rw4t 1/1 Terminating 0 30m 10.254.29.2 kube-node01

haha-6df4f7f6d7-td9gm 0/1 ContainerCreating 0 6s <none> kube-node03

[root@kube\_master ~]# kubectl get pod -o wide

NAME READY STATUS RESTARTS AGE IP NODE

haha-6df4f7f6d7-8rw4t 1/1 Terminating 0 30m 10.254.29.2 kube-node01

haha-6df4f7f6d7-td9gm 0/1 ContainerCreating 0 15s <none> kube-node03

[root@kube\_master ~]# kubectl get pod -o wide

NAME READY STATUS RESTARTS AGE IP NODE

haha-6df4f7f6d7-8rw4t 1/1 Terminating 0 31m 10.254.29.2 kube-node01

haha-6df4f7f6d7-td9gm 1/1 Running 0 29s 10.254.46.2 kube-node03

[root@kube\_master ~]# kubectl get pod -o wide

NAME READY STATUS RESTARTS AGE IP NODE

haha-6df4f7f6d7-td9gm 1/1 Running 0 36s 10.254.46.2 kube-node03

[root@kube\_master ~]#

#直接删除pod无效，会删除当前pod但是会在其他节点启动一个新的pod，所以需要直接删除顶级资源deployment

[root@kube\_master ~]# kubectl delete deployment haha

deployment.extensions "haha" deleted

[root@kube\_master ~]# kubectl get deployment

No resources found.

[root@kube\_master ~]# kubectl get replicaset

No resources found.

[root@kube\_master ~]# kubectl get pod

NAME READY STATUS RESTARTS AGE

haha-6df4f7f6d7-td9gm 1/1 Terminating 0 2m

[root@kube\_master ~]# kubectl get pod

No resources found.

[root@kube\_master ~]#

###################################################################service管理

[root@kube\_master ~]# kubectl run web -r 2 --image=192.168.1.1:5000/myos:httpd

deployment.apps "web" created

[root@kube\_master ~]# kubectl get pod -o wide

NAME READY STATUS RESTARTS AGE IP NODE

web-8684bcd4f6-gb46f 1/1 Running 0 7m 10.254.29.2 kube-node01

web-8684bcd4f6-zrwdk 1/1 Running 0 7m 10.254.46.2 kube-node03

[root@kube\_master ~]#

#问题一，只能kube集群内部机器访问，集群外部机器无法访问服务

[root@kube\_master ~]# curl http://10.254.29.2/

docker http test page

[root@kube\_master ~]# curl http://10.254.46.2/

docker http test page

[root@kube\_master ~]#

[root@repo ~]# curl -m 2 http://10.254.46.2/

curl: (28) Connection timed out after 2001 milliseconds

[root@repo ~]# curl -m 2 http://10.254.29.2/

curl: (28) Connection timed out after 2001 milliseconds

[root@repo ~]#

#问题二，若一个pod损坏，自动启动一个新的pod，IP地址发生变化

[root@kube\_master ~]# kubectl delete pod web-8684bcd4f6-zrwdk

pod "web-8684bcd4f6-zrwdk" deleted

[root@kube\_master ~]# kubectl get pod -o wide

NAME READY STATUS RESTARTS AGE IP NODE

web-8684bcd4f6-gb46f 1/1 Running 0 16m 10.254.29.2 kube-node01

web-8684bcd4f6-lbxd4 1/1 Running 0 1m 10.254.100.3 kube-node02

[root@kube\_master ~]#

###################################################################创建service

[root@kube\_master ~]# kubectl expose deploy web --port=80 --target-port=80 --name=http-service

service "http-service" exposed

[root@kube\_master ~]# kubectl get service -o wide

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE SELECTOR

http-service ClusterIP 10.254.213.117 <none> 80/TCP 4m run=web

kubernetes ClusterIP 10.254.0.1 <none> 443/TCP 22h <none>

[root@kube\_master ~]# ping 10.254.213.117

PING 10.254.213.117 (10.254.213.117) 56(84) bytes of data.

^C

--- 10.254.213.117 ping statistics ---

2 packets transmitted, 0 received, 100% packet loss, time 999ms

[root@kube\_master ~]#

#启动一个容器测试

[root@kube\_master ~]# kubectl run testos -i -t --image=192.168.1.1:5000/myos:latest

If you don't see a command prompt, try pressing enter.

[root@testos-6fb6c97755-g2px6 /]# Session ended, resume using 'kubectl attach testos-6fb6c97755-g2px6 -c testos -i -t' command when the pod is running

[root@kube\_master ~]#

[root@kube\_master ~]# kubectl exec -it -c web web-8684bcd4f6-lbxd4 bash

[root@web-8684bcd4f6-lbxd4 html]# ls

index.html

[root@web-8684bcd4f6-lbxd4 html]# hostname >> index.html

[root@web-8684bcd4f6-lbxd4 html]# exit

exit

[root@kube\_master ~]# kubectl exec -it -c web web-8684bcd4f6-gb46f bash

[root@web-8684bcd4f6-gb46f html]# hostname >> index.html

[root@web-8684bcd4f6-gb46f html]# exit

exit

[root@kube\_master ~]# kubectl exec -it -c testos testos-6fb6c97755-g2px6 bash

[root@testos-6fb6c97755-g2px6 /]# curl http://10.254.213.117/

docker http test page

web-8684bcd4f6-lbxd4

[root@testos-6fb6c97755-g2px6 /]# curl http://10.254.213.117/

docker http test page

web-8684bcd4f6-gb46f

[root@testos-6fb6c97755-g2px6 /]# exit

exit

[root@kube\_master ~]#

##################################################################

#使用资源文件

[root@kube\_master ~]# ls

baseos.yaml kube-dashboard.yaml

[root@kube\_master ~]# vim baseos.yaml

[root@kube\_master ~]# cat baseos.yaml

---

apiVersion: extensions/v1beta1 #当前格式的版本

kind: Deployment #当前创建资源的类型， 当前类型是Deployment

metadata: #当前资源的元数据

name: test-os #当前资源的名字 是元数据必须的项

spec: #是当前Deployment的规格说明

replicas: 1 #指当前创建的副本数量 默认不填 默认值就为‘1’

template: #定义pod的模板

metadata: #当前pod的元数据

labels: #至少顶一个labels标签，可任意创建一个 key:value

app: test\_os

spec: #当前pod的规格说明

containers: #容器

- name: centos #是容器的名字容器名字是必须填写的

image: 192.168.1.1:5000/myos:latest #镜像 镜像的名字和版本

stdin: true

tty: true

[root@kube\_master ~]# kubectl create -f baseos.yaml

deployment.extensions "test-os" created

[root@kube\_master ~]# kubectl get deployment

NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE

test-os 1 1 1 0 7s

testos 1 1 1 1 17h

web 2 2 2 2 18h

[root@kube\_master ~]# kubectl get rs

NAME DESIRED CURRENT READY AGE

test-os-6dcd97fd5b 1 1 1 12s

testos-6fb6c97755 1 1 1 17h

web-8684bcd4f6 2 2 2 18h

[root@kube\_master ~]# kubectl get pod

NAME READY STATUS RESTARTS AGE

test-os-6dcd97fd5b-vcz8s 1/1 Running 0 20s

testos-6fb6c97755-g2px6 1/1 Running 1 17h

web-8684bcd4f6-gb46f 1/1 Running 1 18h

web-8684bcd4f6-lbxd4 1/1 Running 1 17h

[root@kube\_master ~]#

#清空原有web的service和deployment

[root@kube\_master ~]# kubectl get service

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

http-service ClusterIP 10.254.213.117 <none> 80/TCP 18h

kubernetes ClusterIP 10.254.0.1 <none> 443/TCP 1d

[root@kube\_master ~]# kubectl get deployment

NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE

test-os 1 1 1 1 39m

testos 1 1 1 1 17h

web 2 2 2 2 18h

[root@kube\_master ~]# kubectl delete service http-service

service "http-service" deleted

[root@kube\_master ~]# kubectl delete deployment web

deployment.extensions "web" deleted

[root@kube\_master ~]#

#编写service的资源文件

[root@kube\_master ~]# vim web.yaml

[root@kube\_master ~]# cat web.yaml

---

apiVersion: extensions/v1beta1

kind: Deployment

metadata:

name: web

spec:

replicas: 3

template:

metadata:

labels:

app: web

spec:

containers:

- name: web

image: 192.168.1.1:5000/myos:httpd

stdin: true

tty: true

---

apiVersion: v1

kind: Service

metadata:

name: web-service

namespace: default

spec:

clusterIP: 10.254.254.110

ports:

- port: 80

targetPort: 80

nodePort:

protocol: TCP

selector:

app: web

type: ClusterIP

[root@kube\_master ~]# kubectl create -f web.yaml

deployment.extensions "web" created

service "web-service" created

[root@kube\_master ~]# kubectl get service -o wide

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE SELECTOR

kubernetes ClusterIP 10.254.0.1 <none> 443/TCP 1d <none>

web-service ClusterIP 10.254.254.110 <none> 80/TCP 11s app=web

[root@kube\_master ~]# kubectl get deployment -o wide

NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

test-os 1 1 1 1 3h centos 192.168.1.1:5000/myos:latest app=test\_os

testos 1 1 1 1 20h testos 192.168.1.1:5000/myos:latest run=testos

web 3 3 3 0 19s web 192.168.1.1:5000/myos:httpd app=web

[root@kube\_master ~]# kubectl get pod -o wide

NAME READY STATUS RESTARTS AGE IP NODE

test-os-6dcd97fd5b-vcz8s 1/1 Running 0 3h 10.254.40.3 kube-node03

testos-6fb6c97755-g2px6 1/1 Running 1 20h 10.254.40.2 kube-node03

web-74d48979b-45qf4 1/1 Running 0 26s 10.254.22.2 kube-node02

web-74d48979b-9gm7g 1/1 Running 0 26s 10.254.30.2 kube-node01

web-74d48979b-cmjnb 1/1 Running 0 26s 10.254.40.4 kube-node03

[root@kube\_master ~]# kubectl exec -it test-os-6dcd97fd5b-vcz8s bash

[root@test-os-6dcd97fd5b-vcz8s /]# curl http://10.254.254.110/

docker http test page

[root@test-os-6dcd97fd5b-vcz8s /]# curl http://10.254.254.110/

docker http test page

[root@test-os-6dcd97fd5b-vcz8s /]# curl http://10.254.254.110/

docker http test page

[root@test-os-6dcd97fd5b-vcz8s /]# exit

exit

[root@kube\_master ~]# kubectl get service web-service -o yaml | tail -2

status:

loadBalancer: {}

[root@kube\_master ~]#

###################################################################配置configmap映射文件或目录

[root@kube\_master ~]# kubectl get pod -o wide

NAME READY STATUS RESTARTS AGE IP NODE

test-os-6dcd97fd5b-vcz8s 1/1 Running 0 4h 10.254.40.3 kube-node03

testos-6fb6c97755-g2px6 1/1 Running 1 21h 10.254.40.2 kube-node03

web-74d48979b-45qf4 1/1 Running 0 1h 10.254.22.2 kube-node02

web-74d48979b-9gm7g 1/1 Running 0 1h 10.254.30.2 kube-node01

web-74d48979b-cmjnb 1/1 Running 0 1h 10.254.40.4 kube-node03

[root@kube\_master ~]# kubectl exec -it web-74d48979b-45qf4 bash

[root@web-74d48979b-45qf4 html]# yum -y install openssh-clients

[root@web-74d48979b-45qf4 html]# scp /etc/httpd/conf/httpd.conf 192.168.1.10:/root

[root@web-74d48979b-45qf4 html]# exit

exit

[root@kube\_master ~]# ls

baseos.yaml httpd.conf kube-dashboard.yaml web1.yaml web-services.yaml web.yaml

[root@kube\_master ~]# mkdir files

[root@kube\_master ~]# mv httpd.conf files/

[root@kube\_master ~]# sed -rn '/80/p' files/httpd.conf

#Listen 12.34.56.78:80

Listen 80

#ServerName [www.example.com:80](http://www.example.com:80)

[root@kube\_master ~]# sed -ri '/^Listen/s/80/8080/' files/httpd.conf

[root@kube\_master ~]# sed -rn '/8080/p' files/httpd.conf

Listen 8080

[root@kube\_master ~]# kubectl create configmap my-httpd --from-file=files/httpd.conf

configmap "my-httpd" created

[root@kube\_master ~]# kubectl get configmap -o wide

NAME DATA AGE

my-httpd 1 11s

[root@kube\_master ~]# vim web1.yaml

[root@kube\_master ~]# cat web1.yaml

---

apiVersion: extensions/v1beta1 #格式版本

kind: Deployment #创建资源类型

metadata: #资源元数据

name: web1 #资源名字

spec: #规格说明

replicas: 1 #副本数量，默认为1

template: #定义pod的模板

metadata: #定义pod的元数据

labels: #定义pod的标签

run: web1

spec: #pod规格说明

containers: #容器

- name: web1 #容器名称

image: 192.168.1.1:5000/myos:httpd #镜像地址

ports: #端口类

- containerPort: 8080 #容器端口

stdin: true #标准输入

tty: true #交互终端

volumeMounts: #容器内部映射参数

- mountPath: /var/webroot/ #容器内映射目录

name: site-data #映射参数名(配置文件内使用)

- mountPath: /etc/httpd/conf/httpd.conf #容器内映射文件

name: my-config #映射参数名(配置文件内使用)

subPath: httpd.conf #映射文件名

volumes: #容器外部映射参数

- name: site-data #映射参数名(配置文件内使用)

emptyDir: {} #容器外映射目录

- name: my-config #映射参数名(配置文件内使用)

configMap: #调用映射参数(真实配置映射)

name: my-httpd #调用映射名(kubectl命令配置的真实映射名)

items: #真实映射参数

- key: httpd.conf #映射键值

path: httpd.conf #真实映射文件名

[root@kube\_master ~]# kubectl create -f web1.yaml

deployment.extensions "web1" created

[root@kube\_master ~]# kubectl get deployment web1 -o wide

NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

web1 1 1 1 1 11s web1 192.168.1.1:5000/myos:httpd run=web1

[root@kube\_master ~]# kubectl get pod web1-c6976dd7d-x5t99 -o wide

NAME READY STATUS RESTARTS AGE IP NODE

web1-c6976dd7d-x5t99 1/1 Running 0 21s 10.254.30.3 kube-node01

[root@kube\_master ~]# curl -m 2 http://10.254.30.3/

curl: (7) Failed connect to 10.254.30.3:80; 拒绝连接

[root@kube\_master ~]# curl -m 2 http://10.254.30.3:8080/

docker http test page

[root@kube\_master ~]#

###################################################################kube-dns插件

[root@kube\_master ~]# kubectl describe pod web1-c6976dd7d-x5t99 | tail -1

Warning MissingClusterDNS 16s (x18 over 18m) kubelet, kube-node01 pod: "web1-c6976dd7d-x5t99\_default(9c9bf8b6-9bf8-11ea-baf8-52540078c20b)". kubelet does not have ClusterDNS IP configured and cannot create Pod using "ClusterFirst" policy. Falling back to "Default" policy.

[root@kube\_master ~]#

#导入kube-dns镜像，共3个

[root@kube\_master ~]# curl http://192.168.1.1:5000/v2/\_catalog

{"repositories":["busybox","k8s-dns-dnsmasq-nanny-amd64","k8s-dns-kube-dns-amd64","k8s-dns-sidecar-amd64","kubernetes-dashboard-amd64","myos","pod-infrastructure"]}

[root@kube\_master ~]#

#修改kubelet服务配置文件，三个node节点都要修改

[root@kube\_node01 ~]# vim /etc/kubernetes/kubelet

[root@kube\_node01 ~]# tail -1 /etc/kubernetes/kubelet

KUBELET\_ARGS="--cgroup-driver=systemd --fail-swap-on=false --kubeconfig=/etc/kubernetes/kubelet.kubeconfig --pod-infra-container-image=192.168.1.1:5000/pod-infrastructure:latest --cluster-dns=10.254.254.253 --cluster-domain=tedu.local."

[root@kube\_node01 ~]# systemctl restart kubelet.service

[root@kube\_node02 ~]# vim /etc/kubernetes/kubelet

[root@kube\_node02 ~]# tail -1 /etc/kubernetes/kubelet

KUBELET\_ARGS="--cgroup-driver=systemd --fail-swap-on=false --kubeconfig=/etc/kubernetes/kubelet.kubeconfig --pod-infra-container-image=192.168.1.1:5000/pod-infrastructure:latest --cluster-dns=10.254.254.253 --cluster-domain=tedu.local."

[root@kube\_node02 ~]# systemctl restart kubelet.service

[root@kube\_node02 ~]#

[root@kube\_node03 ~]# vim /etc/kubernetes/kubelet

[root@kube\_node03 ~]# tail -1 /etc/kubernetes/kubelet

KUBELET\_ARGS="--cgroup-driver=systemd --fail-swap-on=false --kubeconfig=/etc/kubernetes/kubelet.kubeconfig --pod-infra-container-image=192.168.1.1:5000/pod-infrastructure:latest --cluster-dns=10.254.254.253 --cluster-domain=tedu.local."

[root@kube\_node03 ~]# systemctl restart kubelet.service

[root@kube\_node03 ~]#

#重启node节点的kubelet服务以后再master上确认节点状态

[root@kube\_master ~]# kubectl get node

NAME STATUS ROLES AGE VERSION

kube-node01 Ready <none> 1d v1.10.3

kube-node02 Ready <none> 1d v1.10.3

kube-node03 Ready <none> 1d v1.10.3

[root@kube\_master ~]#