搭建Kubernetes集群

建立日期：2020年7月30日

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 变更原因 | 变更内容简述 | 编制/修订者 | 适用范围 | 发布日期 |
| V1.0 | 建立 |  | 陈文华 | 开发、测试 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## 环境准备

以下操作在所有服务器上进行

### 服务器配置信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IP | 操作系统 | 角色 | Hostname | Docker版本 |
| 192.168.84.49 | Centos7.8 | master | master | 19.03.12 |
| 192.168.84.48 | Centos7.8 | worker | node1 | 19.03.12 |
| 192.168.84.47 | Centos7.8 | worker | node2 | 19.03.12 |

服务器时间校准

yum -y install ntp

ntpdate ntp1.aliyun.com

timedatectl set-timezone Asia/Shanghai

使用timedatectl命令查看当前时间

### 2、修改hostname

hostnamectl set-hostname your-new-host-name

hostnamectl status

重启后生效

### 3、安装系统工具

如果系统已有，跳过这一步

yum install -y net-tools.x86\_64 vim wget gcc

### 4、添加host解析



### 5、关闭防火墙

目的是防止kubernetes的Master节点和Node节点间的通信出现问题，正式环境的做法是在防火墙打开配置互相通信的端口，比如6443等

systemctl disable firewalld

systemctl stop firewalld

### 6、关闭SeLinux

setenforce 0

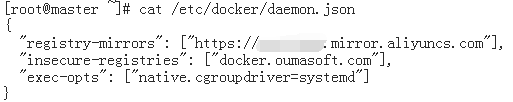
sed -i 's/^SELINUX=enforcing$/SELINUX=disabled/' /etc/selinux/config

### 7、配置Docker

vim /etc/docker/daemon.json

添加如下内容

"exec-opts": ["native.cgroupdriver=systemd"]



然后重启docker

systemctl daemon-reload

systemctl restart docker

### 8、流量转发

cat > /etc/sysctl.d/k8s.conf << EOF

net.bridge.bridge-nf-call-ip6tables = 1

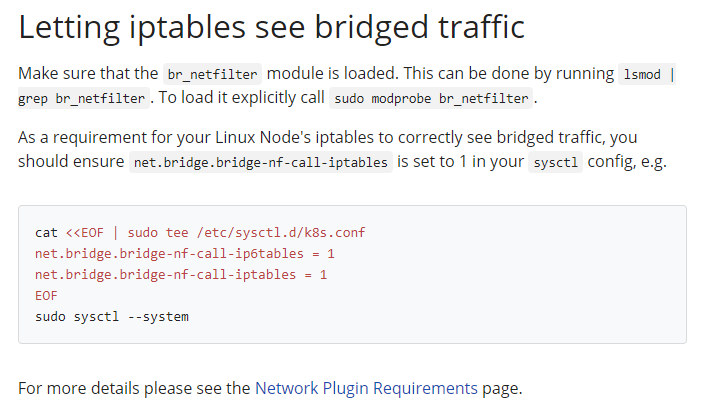
net.bridge.bridge-nf-call-iptables = 1

EOF

sysctl --system

官网的解释

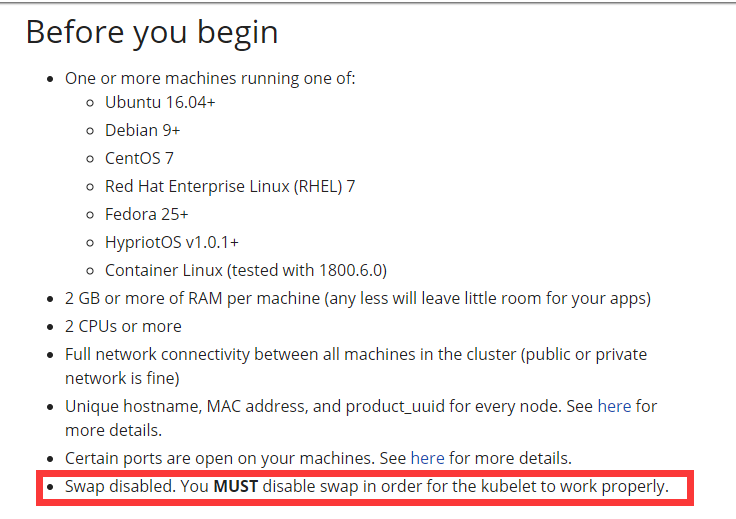
<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>



### 9、关闭swap

swapoff –a

官网的配置要求



### 10、安装kubeadm

1. 添加阿里kubernetes源

cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86\_64/

enabled=1

gpgcheck=1

repo\_gpgcheck=1

gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg

EOF

1. 安装kubelet、kubeadm、kubectl

yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes

以上命令默认安装最新稳定版，如需安装指定版本，如1.18.0

yum install -y kubelet-1.18.0 kubeadm-1.18.0 kubectl-1.18.0 --disableexcludes=kubernetes

1. 启动kubelet并设置开机自启

systemctl enable kubelet && systemctl start kubelet

## 二、安装master

以下操作只在master上进行

1. 执行kubeadm init命令初始化master

kubeadm init --kubernetes-version=v1.18.6 \

--pod-network-cidr=10.244.0.0/16 \

--service-cidr=10.1.0.0/16 \

--apiserver-advertise-address=192.168.84.49 \

--image-repository registry.aliyuncs.com/google\_containers

kubernetes-version: 指定kubernetes的版本，这里指定为最新的1.18.6版本；

pod-network-cidr：因为后面我们选择flannel作为Pod的网络插件，所以这里需要指定Pod的网络范围为10.244.0.0/16

service-cidr：用于指定SVC的网络范围；

apiserver-advertise-address：用于指定kube-apiserver监听的ip地址，就是master的IP地址。

image-repository: 其中默认的镜像仓库k8s.gcr.io没有科学上网的话无法访问，修改为国内的阿里镜像仓库

值得一提的是，截止到2020-7-30，阿里云镜像还没有同步1.18.6版本的kubernetes，目前同步过来的最新版为1.18.3，初始化操作如果出错，请将版本修改为1.18.3，另外kubernetes每个版本支持的docker版本不一样，kubernetes1.17版本支持docker19.03

<https://github.com/kubernetes/kubernetes/tree/master/CHANGELOG>

<https://github.com/kubernetes/kubernetes/releases/tag/v1.18.6>

1. 执行kubeadm init之前可以先查看所需镜像

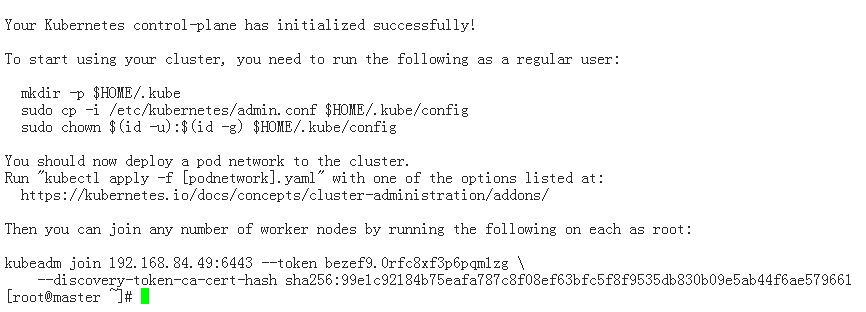
kubeadm config images list

可以先拉取镜像

kubeadm config images pull --image-repository=registry.aliyuncs.com/google\_containers --kubernetes-version=v1.18.6

安装过程出错时，执行 kubeadm reset 重置master

1. 安装成功后，有如下信息



根据提示，执行命令

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

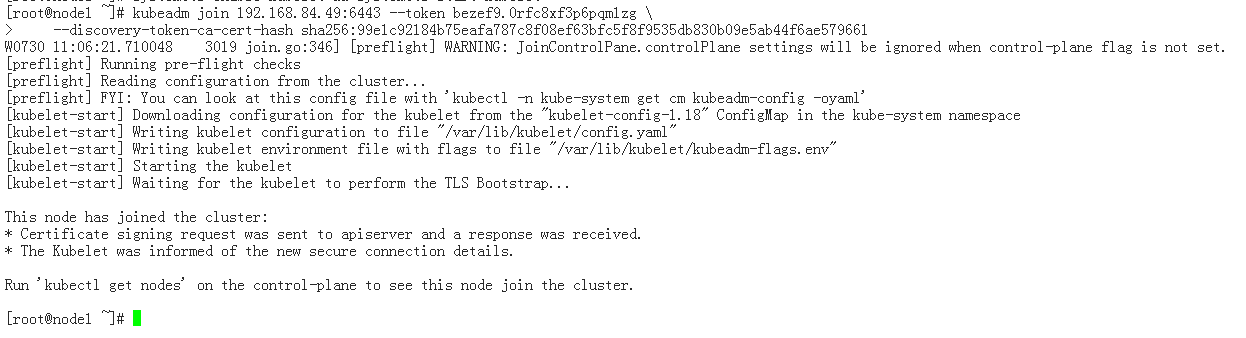
kubeadm join后面的一长串是用于node节点加入master节点的

## 三、安装node

在node1和node2上执行master节点安装成功后kubeadm join那一长串



加入集群成功后输入如下内容



## 四、安装网络插件

1、**查看master节点状态**

kubectl get nodes

显示NotReady状态，需要安装网络插件，kubernete支持多种网络插件：

Calico链接 https://docs.projectcalico.org/introduction/

Flannel链接 https://github.com/coreos/flannel/

Weave链接 https://www.weave.works/oss/net/

Canal 链接 https://github.com/projectcalico/canal

Calico使用较多，因为支持网络策略、支持服务网格Istio集成

安装文档<https://docs.projectcalico.org/getting-started/kubernetes/quickstart>

如果使用Calico，master启动命令修改为

kubeadm init --kubernetes-version=1.18.6 \

--apiserver-advertise-address=192.168.84.49 \

--image-repository registry.aliyuncs.com/google\_containers \

--service-cidr=10.10.0.0/16 --pod-network-cidr=10.122.0.0/16

2、**安装网络组件，以flannel为例**

wget https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

kubectl apply -f kube-flannel.yml

如果安装calico网络插件

kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml

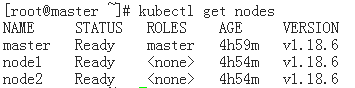
3、再次查看节点状态

kubectl get nodes

kubectl get pods --all-namespaces

如果STATUS提示NotReady，可以通过 kubectl describe node master 命令

查看具体的描述信息，性能差的服务器到达Ready状态时间会长些



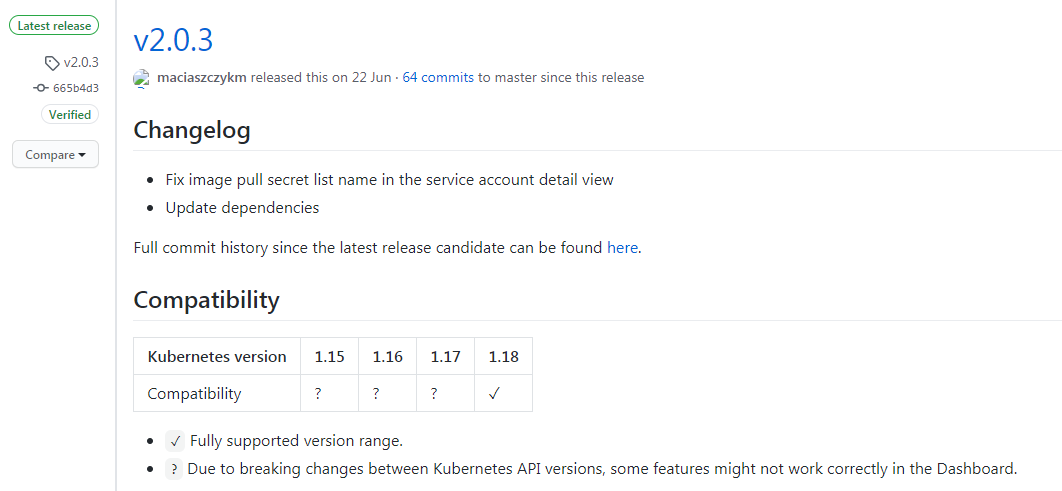
至此，集群成功搭建完成

## 五、安装Dashboard

### 1、下载Dashboard

Dashboard的github地址

<https://github.com/kubernetes/dashboard/releases>

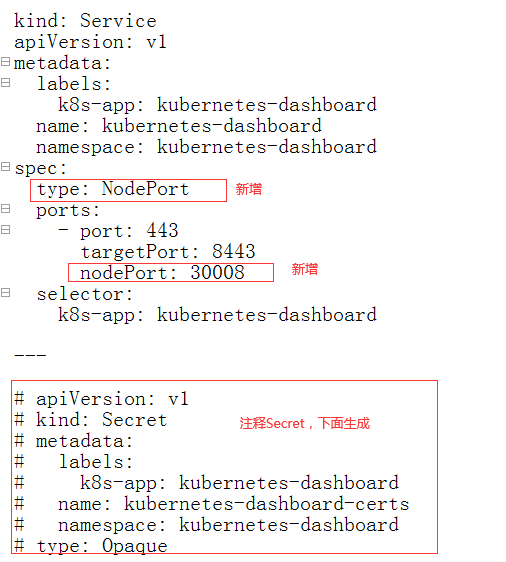


下载最新版2.0.3，可能需要科学上网

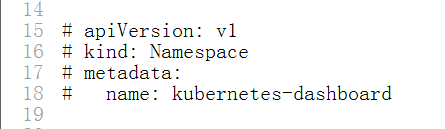
wget https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.3/aio/deploy/recommended.yaml

### 2、修改配置文件

vim recommended.yaml



文件开始创建Namespace的这段也注释



### 3、创建证书

mkdir dashboard-certs

cd dashboard-certs/

#创建命名空间

kubectl create namespace kubernetes-dashboard

#创建key文件

openssl genrsa -out dashboard.key 2048

#证书请求

openssl req -days 36000 -new -out dashboard.csr -key dashboard.key -subj '/CN=dashboard-cert'

#自签证书

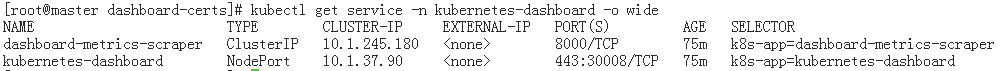
openssl x509 -req -in dashboard.csr -signkey dashboard.key -out dashboard.crt

#创建kubernetes-dashboard-certs对象

kubectl create secret generic kubernetes-dashboard-certs --from-file=dashboard.key --from-file=dashboard.crt -n kubernetes-dashboard

然后执行kubectl create -f ../recommended.yaml命令安装Dashboard。

使用kubectl get service -n kubernetes-dashboard -o wide命令查看是否部署成功：



### 4、创建账号与授权

Dashboard部署好后，接着创建账号：

vim dashboard-admin.yaml

内容如下

apiVersion: v1

kind: ServiceAccount

metadata:

labels:

k8s-app: kubernetes-dashboard

name: dashboard-admin

namespace: kubernetes-dashboard

创建该账号：

kubectl create -f dashboard-admin.yaml

账号创建好后，接着为其授权：

vim dashboard-admin-bind-cluster-role.yaml

内容如下所示：

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: dashboard-admin-bind-cluster-role

labels:

k8s-app: kubernetes-dashboard

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: ClusterRole

name: cluster-admin

subjects:

- kind: ServiceAccount

name: dashboard-admin

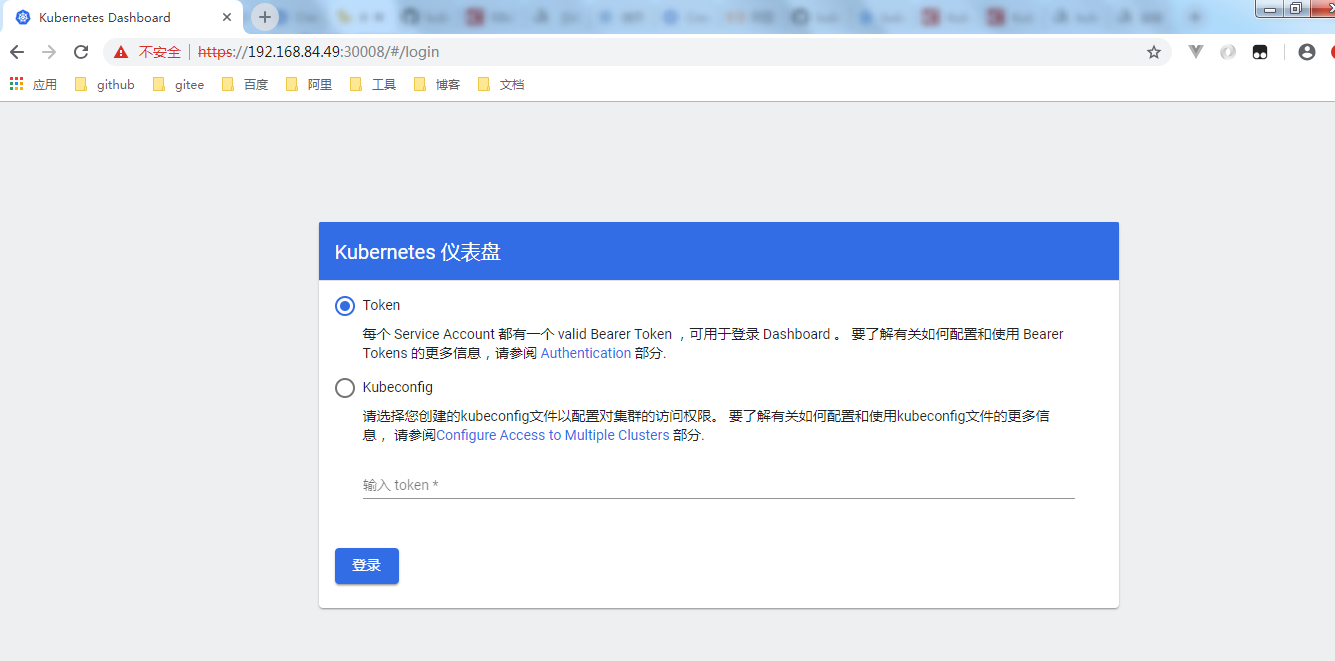
namespace: kubernetes-dashboard

授权：

kubectl create -f dashboard-admin-bind-cluster-role.yaml

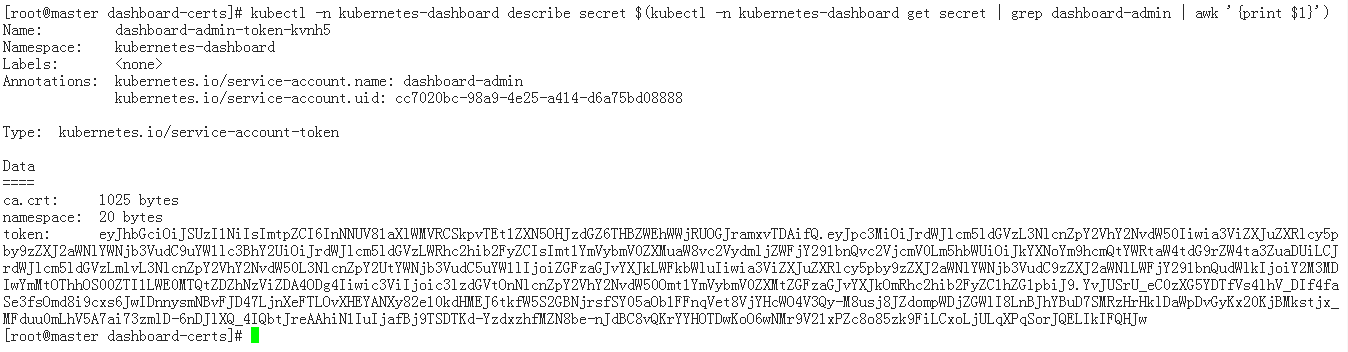
浏览器访问

<https://192.168.84.49:30008/#/login>

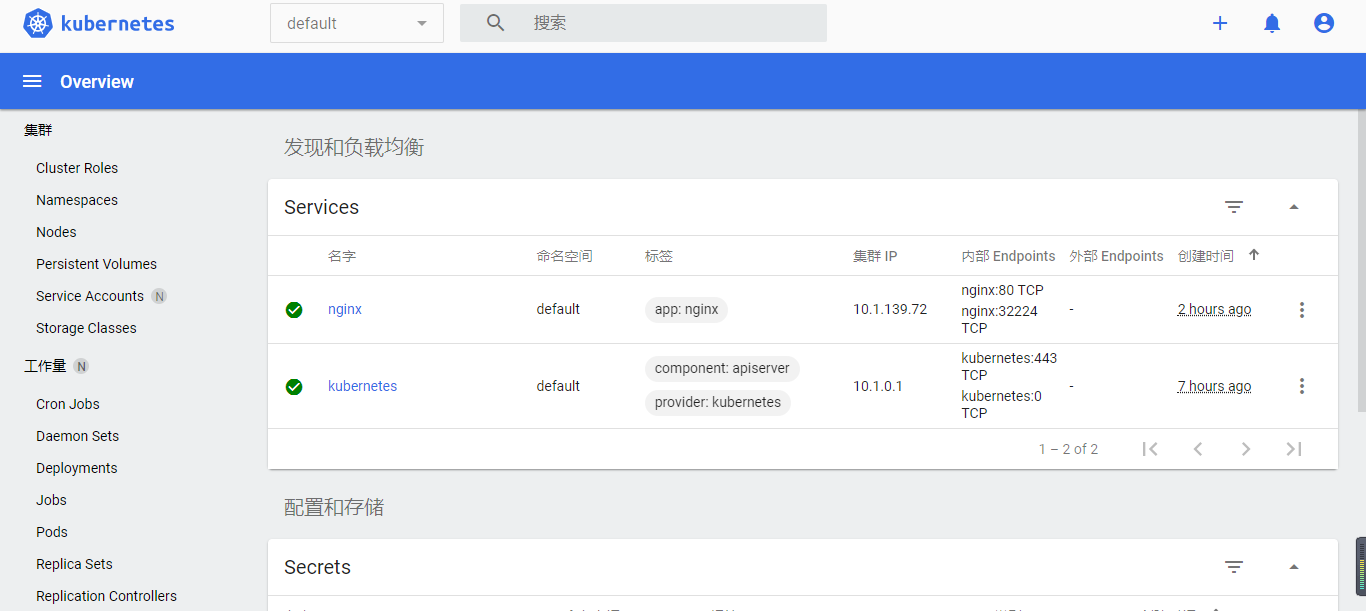


选择Token，Token的值可以用下面的命令获取：

kubectl -n kubernetes-dashboard describe secret $(kubectl -n kubernetes-dashboard get secret | grep dashboard-admin | awk '{print $1}')



复制togen的值，登录



## 六、搭建NFS服务器

Kubernetes支持多种存储模式

<https://kubernetes.io/docs/concepts/storage/persistent-volumes/#types-of-persistent-volumes>

以NFS为例

由于服务器资源有限，我们把NFS服务搭建在node1节点上，正式环境最好使用单独的服务器节点搭建NFS

1. 在node1节点上执行命令

mkdir /nfs

chmod 777 /nfs

touch /etc/exports

exports内容如下所示

/nfs \*(rw,insecure,sync,no\_subtree\_check,no\_root\_squash)

1. 使配置生效

exportfs –r

1. 启动NFS

systemctl enable nfs

systemctl enable rpcbind

systemctl restart nfs

systemctl restart rpcbind

1. 测试NFS

在master节点的/home/test目录下新建test-nfs.yml

apiVersion: v1

kind: Pod

metadata:

name: test-nfs-pod

spec:

containers:

- name: busybox

image: busybox

command:

- sh

- -c

- 'echo hello world > /mnt/hello'

imagePullPolicy: IfNotPresent

volumeMounts:

- mountPath: "/mnt"

name: nfs

volumes:

- name: nfs

nfs:

path: /nfs

server: 192.168.84.48

创建文件之后，执行kubectl apply -f test-nfs.yml

1. 切换回node1服务器，查看/nfs目录下是否有hello文件
2. 切换回master服务器，删除测试

kubectl delete -f test-nfs.yml