172.25.254.11 rke(master) centos7.3

172.25.254.12 node1 centos7.3

172.25.254.13 node2 centos7.3

RKE是一个用Golang编写的Kubernetes安装程序，极为简单易用，用户不再需要做大量的准备工作，即可拥有闪电般快速的Kubernetes安装部署体验。

rke,node1,node2都安装docker-ce：

yum install -y docker-ce

SELinux：

必须关闭！CentOS7下可修改配置文件/etc/sysconfig/selinux，设置SELINUX=disabled ，重启后永久关闭。关闭防火墙

主机文件：

172.25.254.11 rke

172.25.254.12 node1

172.25.254.13 node2

172.25.254.14 node3

禁用SWAP：

一定要禁用swap，否则kubelet组件无法运行，永久禁用swap可以直接修改/etc/fstab文件，注释掉swap项。swapoff -a 只是临时禁用

启用cgroup内存和Swap限额，修改/etc/default/grub 配置文件，修改/增加两项：

GRUB\_CMDLINE\_LINUX\_DEFAULT="cgroup\_enable=memory swapaccount=1"

GRUB\_CMDLINE\_LINUX="cgroup\_enable=memory swapaccount=1"

IPV4转发：

vim /etc/sysctl.conf

net.ipv4.ip\_forward = 1

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

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

执行sysctl -p 立刻生效。

modprobe br\_netfilter

**设置SSH**

RKE通过SSH tunnel进行安装部署，需要事先建立RKE到各节点的SSH免密登录。如果集群中有3个节点，需要在RKE机器上执行1遍秘钥生成命令ssh-keygen，并将生成侧公钥通过分发命令：ssh-copy-id {user}@{ip}。

1）在各个node节点(172.25.254.12、172.25.254.13)上创建ssh用户，并将其添加至docker组中：

#useradd docker\_user

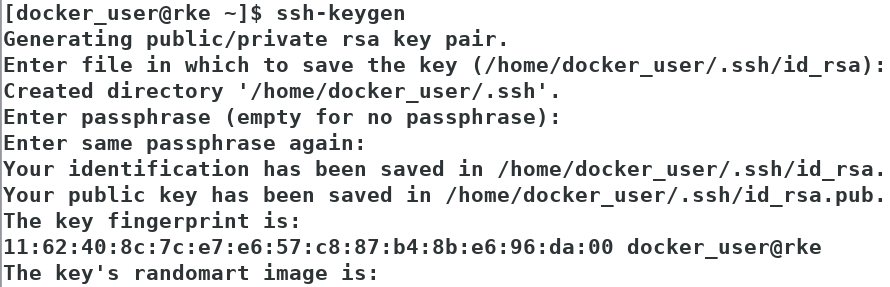
#usermod -aG docker docker\_user

注意：重启系统以后才能生效，只重启Docker服务是不行的！重启后，docker\_user用户也可以直接使用docker run命令。

1. 在rke所在主机上创建密钥：

[root@rke ~]# su - docker\_user

[docker\_user@rke ~]$ssh-keygen



1. 将所生成的密钥的公钥分发到各个node节点：

vi /etc/ssh/sshd\_config

PermitRootLogin yes #打开

systemctl restart sshd

[docker\_user@rke ~]$ssh-copy-id docker\_user@172.25.254.12

[docker\_user@rke ~]$ssh-copy-id docker\_user@172.25.254.13

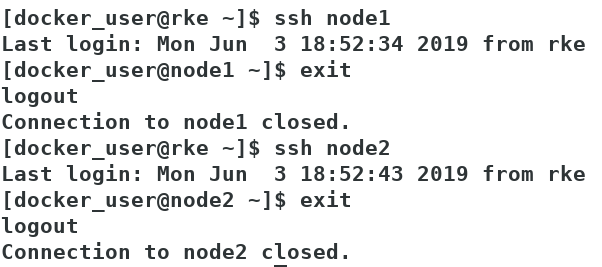
**注意：此处也要使用ssh-copy-id docker\_user@本机IP，否则也会报错，另在centos系统下不能使用root用户进行免密设置**

测试连接

使用ssh连接各主机看是否可以成功免密码登录，如果成功即可继续后续操作，具体如下：

[docker\_user@rke ~]$ ssh node1

[docker\_user@rke ~]$ ssh node2



三个节点启动docker：

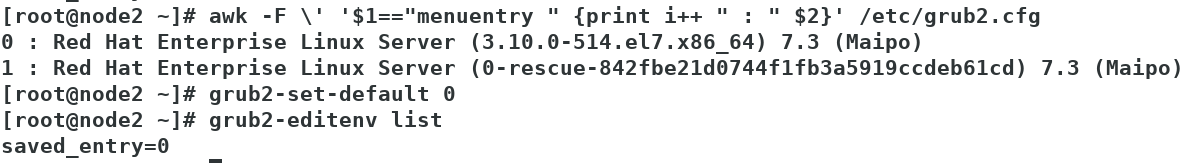
[root@ ~]# systemctl start docker

[root@ ~]# systemctl enable docker

查看内核默认启动顺序

awk -F \' '$1=="menuentry " {print i++ " : " $2}' /etc/grub2.cfg

#设置默认启动的内核，顺序index 分别是 0,1,2,3，每个人机器不一样，看清楚选择自己的index



**升级系统内核**

运行docker的node节点需要升级到4.x内核才支持overlay2驱动，docker 存储驱动很多默认用devicemapper，存在很多问题，但目前最好建议使用overlay2，内核版本小于 3.10.0-693 的不要使用 overlay2 驱动。两个node节点升级内核：

创建升级脚本

[root@node ~]# vim update-kernel.sh

#!/bin/bash

set -o errexit

set -o nounset

set -o pipefail

#检查内核

uname -sr

#添加升级内核的第三方库 www.elrepo.org 上有方法

rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org

#列出内核相关包

rpm -Uvh http://www.elrepo.org/elrepo-release-7.0-3.el7.elrepo.noarch.rpm

#安装最新稳定版

yum --disablerepo="\*" --enablerepo="elrepo-kernel" list available

#查看内核默认启动顺序

yum --enablerepo=elrepo-kernel install kernel-ml -y

#结果显示

awk -F\' '$1=="menuentry " {print $2}' /etc/grub2.cfg

#设置默认启动的内核，顺序index 分别是 0,1,2,3，每个人机器不一样，看清楚选择自己的index， 执行以下代码选择内核

grub2-set-default 0

echo "update Centos7 success!"

[root@node ~]# chmod +x update-kernel.sh

[root@node ~]# sh update-kernel.sh

修改内核默认启动顺序

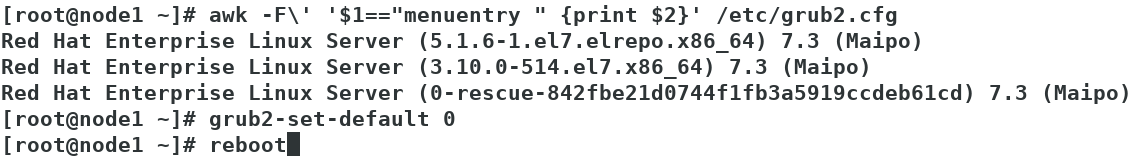
[root@node1 ~]# awk -F\' '$1=="menuentry " {print $2}' /etc/grub2.cfg

Red Hat Enterprise Linux Server (5.1.6-1.el7.elrepo.x86\_64) 7.3 (Maipo)

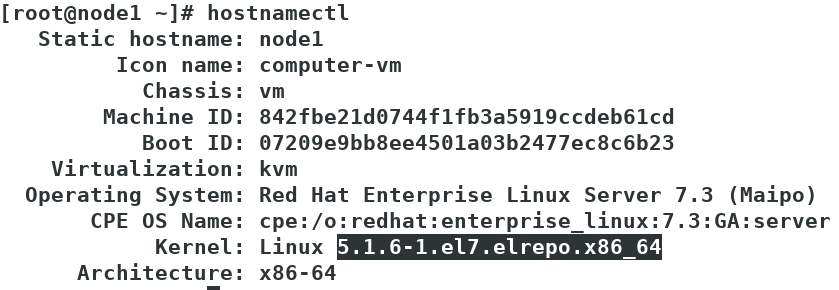
Red Hat Enterprise Linux Server (3.10.0-514.el7.x86\_64) 7.3 (Maipo)

Red Hat Enterprise Linux Server (0-rescue-842fbe21d0744f1fb3a5919ccdeb61cd) 7.3 (Maipo)

[root@node1 ~]# grub2-set-default 0



[root@node1 ~]# hostnamectl



使用overlay2驱动

[root@node1 ~]# yum install -y yum-plugin-ovl

关闭docker

[root@node1 ~]#systemctl stop docker

配置 docker 的存储驱动

# 如该文件没有，可通过如下命令直接新建

[root@node1 ~]#vi /etc/docker/daemon.json

{

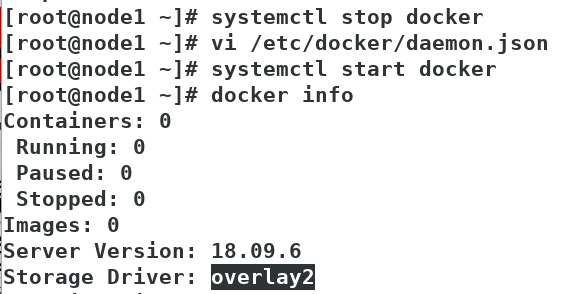
"storage-driver": "overlay2"

}

启动 docker

systemctl start docker

[root@node1 ~]# docker info



RKE的安装

下载RKE二进制文件

[root@rke~]#wget https://github.com/rancher/rke/releases/download/v0.1.11/rke\_linux-amd64

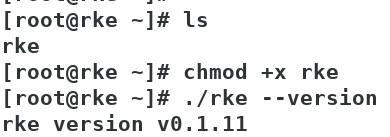
[root@rke ~]# mv rke\_linux-amd64 rke

[root@rke ~]# ./rke --version #通过运行以下命令确认RKE现在是可执行的：

[root@rke ~]# mv rke /home/docker\_user

[root@rke ~]# su - docker\_user

[root@rke ~]# chmod 777 /home/docker\_user/rke



**创建群集配置文件**

RKE使用群集配置文件，称为cluster.yml确定群集中的节点以及如何部署Kubernetes。有许多配置选项，可以在设置cluster.yml。在本示例中，将为Kubernetes集群假设最少一个节点。

有两种简单的方法可以创建cluster.yml：

使用我们的最小值cluster.yml并根据您将使用的节点更新它。

使用rke config来查询所需的所有信息。

运用 rke config

使用默认方式创建cluster.yml

[root@rke ~]# su - docker\_user

[docker\_user@rke ~]$ ./rke config #要创建新的cluster.yml，运行rke config并且此命令会提示您输入构建群集所需的所有信息。查看群集配置选项以了解每个问题的含义。

[docker\_user@rke ~]$ vim cluster.yml

# If you intened to deploy Kubernetes in an air-gapped environment,

# please consult the documentation on how to configure custom RKE images.

nodes:

- address: "172.25.254.11"

port: "22"

internal\_address: ""

role:

- controlplane

- etcd

hostname\_override: "rke"

user: docker\_user

docker\_socket: /var/run/docker.sock

ssh\_key: ""

ssh\_key\_path: /home/docker\_user/.ssh/id\_rsa

labels: {}

- address: "172.25.254.12"

port: "22"

role:

- worker

hostname\_override: node1

user: docker\_user

docker\_socket: /var/run/docker.sock

ssh\_key: ""

ssh\_key\_path: /home/docker\_user/.ssh/id\_rsa

labels: {}

- address: "172.25.254.13"

port: "22"

role:

- worker

hostname\_override: node2

user: docker\_user

docker\_socket: /var/run/docker.sock

ssh\_key: ""

ssh\_key\_path: /home/docker\_user/.ssh/id\_rsa

labels: {}

services:

etcd:

image: ""

extra\_args: {}

extra\_binds: []

extra\_env: []

external\_urls: []

ca\_cert: ""

cert: ""

key: ""

path: ""

snapshot: false

retention: ""

creation: ""

kube-api:

image: ""

extra\_args: {}

extra\_binds: []

extra\_env: []

service\_cluster\_ip\_range: 10.43.0.0/16

service\_node\_port\_range: ""

pod\_security\_policy: false

kube-controller:

image: ""

extra\_args: {}

extra\_binds: []

extra\_env: []

cluster\_cidr: 10.42.0.0/16

service\_cluster\_ip\_range: 10.43.0.0/16

scheduler:

image: ""

extra\_args: {}

extra\_binds: []

extra\_env: []

kubelet:

image: ""

extra\_args: {}

extra\_binds: []

extra\_env: []

cluster\_domain: cluster.local

infra\_container\_image: ""

cluster\_dns\_server: 10.43.0.10

fail\_swap\_on: false

kubeproxy:

image: ""

extra\_args: {}

extra\_binds: []

extra\_env: []

network:

plugin: canal

options: {}

authentication:

strategy: x509

options: {}

sans: []

addons: ""

addons\_include: []

system\_images:

etcd: rancher/coreos-etcd:v3.2.18

alpine: rancher/rke-tools:v0.1.15

nginx\_proxy: rancher/rke-tools:v0.1.15

cert\_downloader: rancher/rke-tools:v0.1.15

kubernetes\_services\_sidecar: rancher/rke-tools:v0.1.15

kubedns: rancher/k8s-dns-kube-dns-amd64:1.14.10

dnsmasq: rancher/k8s-dns-dnsmasq-nanny-amd64:1.14.10

kubedns\_sidecar: rancher/k8s-dns-sidecar-amd64:1.14.10

kubedns\_autoscaler: rancher/cluster-proportional-autoscaler-amd64:1.0.0

kubernetes: rancher/hyperkube:v1.11.3-rancher1

flannel: rancher/coreos-flannel:v0.10.0

flannel\_cni: rancher/coreos-flannel-cni:v0.3.0

calico\_node: rancher/calico-node:v3.1.3

calico\_cni: rancher/calico-cni:v3.1.3

calico\_controllers: ""

calico\_ctl: rancher/calico-ctl:v2.0.0

canal\_node: rancher/calico-node:v3.1.3

canal\_cni: rancher/calico-cni:v3.1.3

canal\_flannel: rancher/coreos-flannel:v0.10.0

wave\_node: weaveworks/weave-kube:2.1.2

weave\_cni: weaveworks/weave-npc:2.1.2

pod\_infra\_container: rancher/pause-amd64:3.1

ingress: rancher/nginx-ingress-controller:0.16.2-rancher1

ingress\_backend: rancher/nginx-ingress-controller-defaultbackend:1.4

metrics\_server: rancher/metrics-server-amd64:v0.2.1

ssh\_key\_path: ~/.ssh/id\_rsa

ssh\_agent\_auth: false

authorization:

mode: rbac

options: {}

ignore\_docker\_version: false

kubernetes\_version: ""

private\_registries: []

ingress:

provider: ""

options: {}

node\_selector: {}

extra\_args: {}

cluster\_name: ""

cloud\_provider:

name: ""

prefix\_path: ""

addon\_job\_timeout: 0

bastion\_host:

address: ""

port: ""

user: ""

ssh\_key: ""

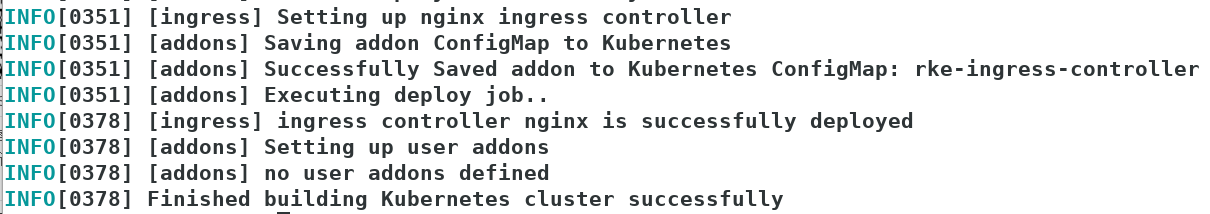
ssh\_key\_path: ""

monitoring:

provider: ""

options: {}

[docker\_user@rke ~]$ ./rke up --config cluster.yml



最后一行Finished building Kubernetes cluster successfully以指示您的群集已准备好使用。作为Kubernetes创建过程的一部分，kubeconfig已创建并编写了一个文件，该文件kube\_config\_cluster.yml可用于开始交互您的Kubernetes集群。

安装kubectl

[root@rke docker\_user]# su - docker\_user

[docker\_user@rke~]$ wget <https://centos-k8s.oss-cn-hangzhou.aliyuncs.com/rpm/1.8.4/kubectl>

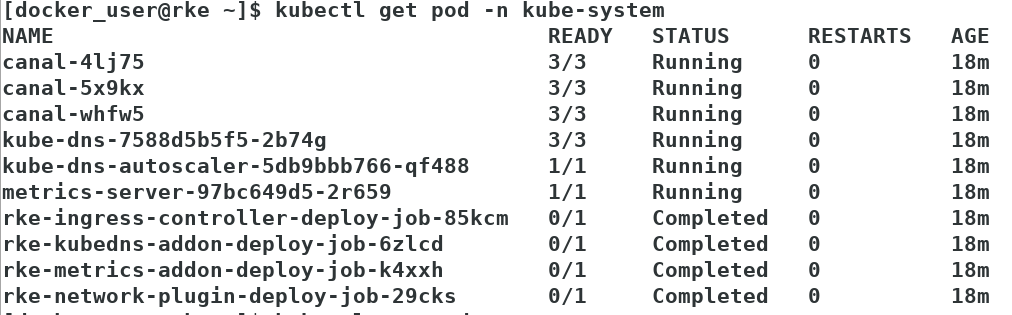
[docker\_user@rke ~]$ chmod a+x kubectl

[docker\_user@rke ~]$ mv kubectl /usr/bin

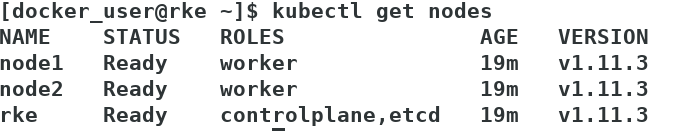
[docker\_user@rke ~]$ mkdir .kube

[docker\_user@rke ~]$ cp kube\_config\_cluster.yml .kube/config

[docker\_user@rke ~]$ kubectl get pod -n kube-system

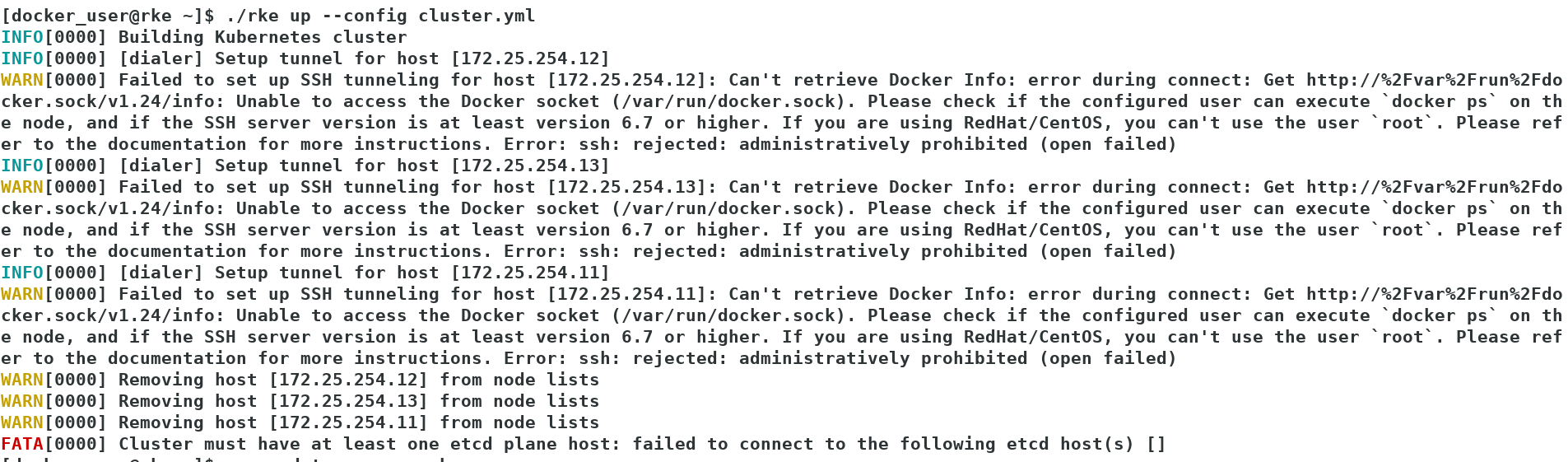


[docker\_user@rke ~]$ kubectl get nodes



**问题（坑）：**

[root@rke ~]# yum update -y openssh-server #升级



yum install -y --setopt=obsoletes=0 docker-ce-17.03.2.ce-1.el7.centos.x86\_64 docker-ce-selinux-17.03.2.ce-1.el7.centos.noarch

版本要是17.03.x

