Rancher最新版本rancher-v2.2.2的HA部署文档

本文预计阅读时间 36 分钟
rancher2.1版本的功能介绍：
https://www.cnrancher.com/docs/rancher/v2.x/cn/overview/feature/

1，准备工作。

1，主机准备。

本次部署所用机器均为 `CentOS Linux release 7.6.1810 (Core)` 。

节点名称	IP	安装组件
nginx	192.168.111.6	nginx
node1	192.168.111.3	etcd, docker, k8s
node2	192.168.111.4	etcd, docker, k8s
node3	192.168.111.5	etcd, docker, k8s

2，软件准备。

因为软件版本可能中有变更，所以我把这次部署的包都放在百度网盘，下载之后部署，以保证部署过程的流畅。

- 链接: https://pan.baidu.com/s/1sdLPuRTDBbd9UrzyMQCTeA
- 提取码: `7ete`

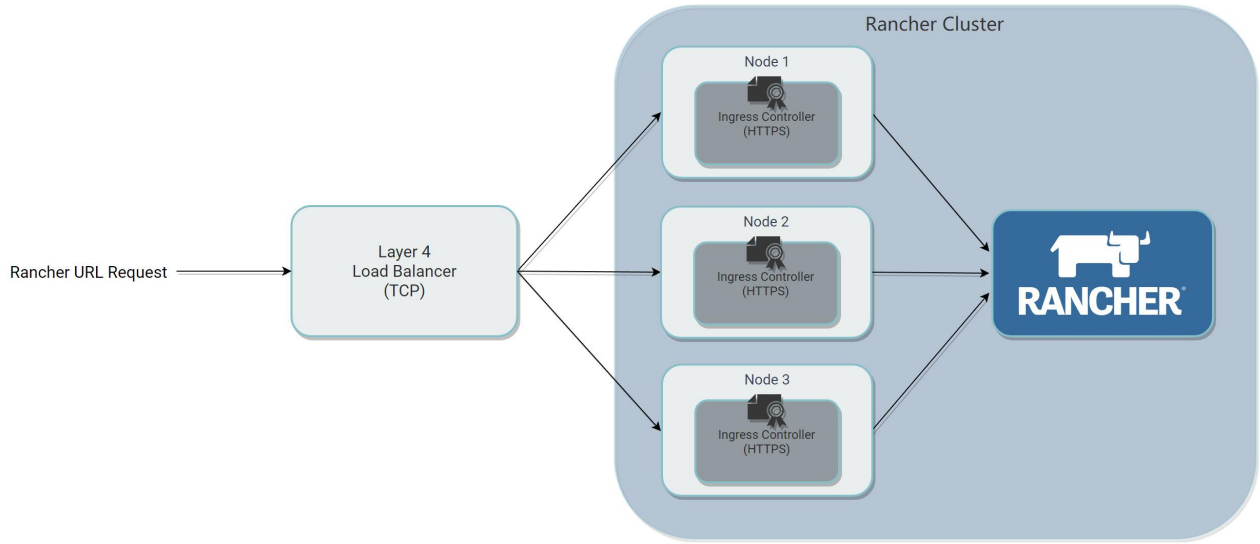
文中相关部署软件的命令，可做相对应的调整。

3，软件版本。

- rancher-2.2.2
- kubernetes-1.13.5
- rke-v0.2.2

- kubectl-v1.13.5
- helm-v2.13.1
- tiller-v2.13.1

4，架构示意。



2，初始化环境。

初始化部分，三台node机器都要操作。

1，关闭相关服务

关闭防火墙

```
systemctl stop firewalld
systemctl disable firewalld
```

关闭setlinx

```
$ sudo setenforce 0
$ grep SELINUX /etc/selinux/config
SELINUX=disabled
```

关闭swap

```
swapoff -a && sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab
```

2，主机名等设置。

设置永久主机名称，然后重新登录

```
$ sudo hostnamectl set-hostname node1
$ sudo hostnamectl set-hostname node2
$ sudo hostnamectl set-hostname node3
```

设置的主机名保存在 /etc/hosts 文件中

```
$ cat > /etc/hosts << EOF
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.111.3 node1
192.168.111.4 node2
192.168.111.5 node3
EOF
```

3，操作系统及kernel调优

文件打开数调优。

```
echo -e  "root soft nofile 65535\nroot hard nofile 65535\n* soft nofile 65535\n*
hard nofile 65535\n"     >> /etc/security/limits.conf
sed -i 's#4096#65535#g' /etc/security/limits.d/20-nproc.conf
```

kernel调优

```
cat >> /etc/sysctl.conf<<EOF
net.ipv4.ip_forward=1
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-ip6tables=1
vm.swappiness=0
vm.max_map_count=655360
EOF
```

4，安装一些基础软件。

```
yum -y install wget ntpdate lrzsz curl yum-utils device-mapper-persistent-data
lvm2 bash-completion && ntpdate -u cn.pool.ntp.org
```

5，创建用户等

创建用户并且添加到docker组

```
groupadd docker
useradd rancher -G docker
echo "123456" | passwd --stdin rancher
```

这一步非常重要，如果没有操作，则后边可能会有报错等问题。

ssh免密登录

在`node1`服务器上执行下面命令：

```
su - rancher
ssh-keygen
ssh-copy-id rancher@192.168.111.3
ssh-copy-id rancher@192.168.111.4
ssh-copy-id rancher@192.168.111.5
```

> 通过授权node1主机对三台主机的免密码登陆，为后边安装k8s的步骤做好准备工作。

3，安装docker。

需要在三台主机上一起安装docker。

`rke工具目前只支持docker v17.03.2，请务必保持版本一致，否则后续安装会报错。`

1、安装repo源：

```
yum-config-manager --add-repo http://mirrors.aliyun.com/docker-
ce/linux/centos/docker-ce.repo
```

2，卸载旧docker版本

```
yum remove -y docker \
              docker-client \
              docker-client-latest \
              docker-common \
              docker-latest \
              docker-latest-logrotate \
              docker-logrotate \
              docker-selinux \
              docker-engine-selinux \
              docker-engine \
              container*
```

3、安装docker-ce-17.03.2

```
yum -y install --setopt=obsoletes=0 docker-ce-17.03.2.ce-1.el7.centos docker-ce-
selinux-17.03.2.ce-1.el7.centos
```

4、启动docker

```
$ systemctl enable docker
$ systemctl start docker
$ systemctl status docker
```

5、添加国内加速代理，设置storage-driver

```
cat > /etc/docker/daemon.json << EOF
{
  "registry-mirrors":
["https://7bezldxe.mirror.aliyuncs.com/","https://kw88y6eh.mirror.aliyuncs.com"],
  "insecure-registries":["192.168.112.69"],
    "storage-driver": "overlay2",
    "storage-opts": [
    "overlay2.override_kernel_check=true"
    ]
}
EOF
```

- `registry-mirrors`：表示公网的加速器地址，可设置多个，`地址需要添加协议头(https或者http)`。
- `insecure-registries`：表示内网的私服地址，`地址不能添加协议头(http)`。
- `storage-driver`：表示使用OverlayFS的overlay2存储驱动。
- 6、重启docker

```
systemctl daemon-reload
systemctl restart docker
```

4，安装nginx。

在192.168.111.6服务器上安装nginx，用于rancher-server负载均衡。

安装nginx：

```
sudo rpm -Uvh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.ngx.noarch.rpm
yum install nginx -y
sudo systemctl enable nginx.service
```

修改配置文件：vi /etc/nginx/nginx.conf

```
user nginx;
worker_processes 4;
worker_rlimit_nofile 40000;

events {
    worker_connections 8192;
}

http {
    # Gzip Settings
    gzip on;
    gzip_disable "msie6";
    gzip_disable "MSIE [1-6]\.(?!.*SV1)";
    gzip_vary on;
    gzip_static on;
    gzip_proxied any;
    gzip_min_length 0;
    gzip_comp_level 8;
    gzip_buffers 16 8k;
    gzip_http_version 1.1;
    gzip_types text/xml application/xml application/atom+xml application/rss+xml application/xhtml+xml image/svg+xml application/font-woff text/javascript application/javascript application/x-javascript text/x-json application/json application/x-web-app-manifest+json text/css text/plain text/x-component font/opentype application/x-font-ttf application/vnd.ms-fontobjectfont/woff2 image/x-icon image/png image/jpeg;

    server {
        listen         80;
        return 301 https://$host$request_uri;
    }
}

stream {
    upstream rancher_servers {
        least_conn;
        server 192.168.111.3:443 max_fails=3 fail_timeout=5s;
        server 192.168.111.4:443 max_fails=3 fail_timeout=5s;
        server 192.168.111.5:443 max_fails=3 fail_timeout=5s;
    }
    server {
        listen     443;
        proxy_pass rancher_servers;
    }
}
```

启动nginx：

```
sudo systemctl restart nginx.service
```

5，Rancher集群部署

1，安装必要工具

以下操作只需在`192.168.111.3`这一台上操作即可。

安装rke:

```
su root
wget https://www.cnrancher.com/download/rke/rke_linux-amd64
chmod +x rke_linux-amd64
mv rke_linux-amd64 /usr/bin/rke
```

安装kubectl：

```
wget https://www.cnrancher.com/download/kubectl/kubectl_amd64-linux
chmod +x kubectl_amd64-linux
mv kubectl_amd64-linux /usr/bin/kubectl
```

安装helm：

```
wget https://www.cnrancher.com/download/helm/helm-linux.tar.gz
tar xf helm-linux.tar.gz
mv linux-amd64/helm /usr/bin/helm
mv linux-amd64/tiller /usr/bin/tiller
rm -rf helm-linux.tar.gz linux-amd64/
```

其它工具下载地址：https://www.cnrancher.com/docs/rancher/v2.x/cn/install-prepare/download/

2，安装k8s

1、切换到rancher用户

```
su - rancher
```

> 注意：必须使用普通用户操作，否则后边的操作会报下边的错：

```
Please check if the configured user can execute `docker ps` on the node, and if
the SSH server version is at least version 6.7 or higher. If youare using
RedHat/CentOS, you can't use the user `root`. Please refer to the documentation
for more instructions. Error: ssh: rejected: administratively prohibited (open
failed)
```

2、创建rancher集群配置文件：

```
cat > rancher-cluster.yml << EOF
nodes:
  - address: 192.168.111.3
    user: rancher
    role: [controlplane,worker,etcd]
  - address: 192.168.111.4
    user: rancher
    role: [controlplane,worker,etcd]
  - address: 192.168.111.5
    user: rancher
    role: [controlplane,worker,etcd]
services:
  etcd:
    snapshot: true
    creation: 6h
    retention: 24h
EOF
```

1. `address`：公共域名或IP地址
2. `user`：可以运行docker命令的用户，需要是普通用户。
3. `role`：分配给节点的Kubernetes角色列表
4. `ssh_key_path`：用于对节点进行身份验证的SSH私钥的路径（默认为~/.ssh/id_rsa）

3、启动集群

```
$ rke up --config ./rancher-cluster.yml
```

如果这一步报错下边的内容：

```
if the SSH server version is at least version 6.7 or higher. If you are using
RedHat/CentOS, you can't use the user `root`. Please refer to the documentation
for more instructions
```

则可能是系统的 `openssh` 版本太低，只需执行如下命令升级即可：

```
[rancher@localhost ~]$ ssh -V
OpenSSH_6.6.1p1, OpenSSL 1.0.1e-fips 11 Feb 2013  #低于上边要求的6.7

[rancher@localhost ~]$ exit

[root@localhost ~]$ yum -y update openssh

[root@localhost ~]$ssh -V
OpenSSH_7.4p1, OpenSSL 1.0.2k-fips  26 Jan 2017
```

然后再切回rancher用户执行安装即可！

完成后，它应显示：`Finished building Kubernetes cluster successfully.`
并且已经创建了一个文件kube_config_rancher-cluster.yml。这个文件包含kubectl和helm访问K8S的凭据。

4、配置环境变量：

切换到root用户

```
su - root

vi /etc/profile

export KUBECONFIG=/home/rancher/kube_config_rancher-cluster.yml
```

保存，并执行：

```
source /etc/profile
```

保存`kube_config_rancher-cluster.yml`和`rancher-cluster.yml`文件的副本,后期维护和升级Rancher实例时将会用到。

5、通过kubectl测试您的连接，并查看您的所有节点是否处于Ready状态

先配置一下kubectl的命令补全功能。

```
$ echo "source <(kubectl completion bash)" >> ~/.bashrc
$ source ~/.bashrc
$ su - rancher
$ echo "source <(kubectl completion bash)" >> ~/.bashrc
$ source ~/.bashrc
```

然后查看节点状态。

```
[root@node1 ~]$ su - rancher
[rancher@node1 ~]$ kubectl get node
NAME            STATUS   ROLES                      AGE   VERSION
192.168.111.3   Ready    controlplane,etcd,worker   10m   v1.13.5
192.168.111.4   Ready    controlplane,etcd,worker   10m   v1.13.5
192.168.111.5   Ready    controlplane,etcd,worker   10m   v1.13.5
```

由于需要联网下载docker镜像文件，所以需要一段时间才能安装好，10-30分钟左右。

6、检查集群Pod的运行状况

```
[rancher@node1 ~]$ kubectl get pods --all-namespaces
NAMESPACE       NAME                                      READY   STATUS
RESTARTS   AGE
ingress-nginx   default-http-backend-78fccfc5d9-j8v5h     1/1     Running     0
9m45s
ingress-nginx   nginx-ingress-controller-cpb9t            1/1     Running     0
9m16s
ingress-nginx   nginx-ingress-controller-fzcdl            1/1     Running     0
9m16s
ingress-nginx   nginx-ingress-controller-n2f5b            1/1     Running     0
9m36s
kube-system     canal-9vzxn                               2/2     Running     0
9m59s
kube-system     canal-p8t59                               2/2     Running     0
9m59s
kube-system     canal-v8nhz                               2/2     Running     0
9m59s
kube-system     kube-dns-58bd5b8dd7-dp8nk                 3/3     Running     0
9m55s
kube-system     kube-dns-autoscaler-77bc5fd84-t2jht       1/1     Running     0
9m54s
kube-system     metrics-server-58bd5dd8d7-pr6nh           1/1     Running     0
9m50s
kube-system     rke-ingress-controller-deploy-job-qh82s   0/1     Completed   0
9m46s
kube-system     rke-kube-dns-addon-deploy-job-g95sp       0/1     Completed   0
9m56s
kube-system     rke-metrics-addon-deploy-job-mmk57        0/1     Completed   0
9m51s
kube-system     rke-network-plugin-deploy-job-b75ds       0/1     Completed   0
10m
```

保存kube_config_rancher-cluster.yml和rancher-cluster.yml文件的副本,以后将需要这些文件来维护和升级Rancher实例。

3，Helm

Helm有两个部分：Helm客户端(helm)和Helm服务端(Tiller)。

使用Helm在集群上安装tiller服务以管理charts，由于RKE默认启用RBAC, 因此我们需要使用kubectl来创建一个serviceaccount，clusterrolebinding才能让tiller具有部署到集群的权限。

1、在kube-system命名空间中创建ServiceAccount：

```
kubectl -n kube-system create serviceaccount tiller
```

2、创建ClusterRoleBinding以授予tiller帐户对集群的访问权限：

```
kubectl create clusterrolebinding tiller --clusterrole cluster-admin --
serviceaccount=kube-system:tiller
```

3、安装Helm Server(Tiller)

```
helm init --service-account tiller   --tiller-image registry.cn-
hangzhou.aliyuncs.com/eryajf/tiller:v2.13.1 --stable-repo-url
https://kubernetes.oss-cn-hangzhou.aliyuncs.com/charts
```

4、安装Tiller金丝雀版本

```
helm init --service-account tiller --canary-image
```

需要修改成国内镜像（可能需要delete再重新init）

```
kubectl --namespace=kube-system set image deployments/tiller-deploy
tiller=registry.cn-hangzhou.aliyuncs.com/eryajf/tiller:v2.13.1
```

4，helm安装rancher

1，添加Chart仓库地址

使用helm repo add命令添加Rancher chart仓库地址,访问Rancher tag和Chart版本
替换为您要使用的Helm仓库分支(即latest或stable）。

```
helm repo add rancher-stable https://releases.rancher.com/server-charts/stable
```

2，安装证书管理器

1、只有Rancher自动生成的证书和LetsEncrypt颁发的证书才需要cert-manager。如果是你自己的证书，可使用ingress.tls.source=secret参数指定证书，并跳过此步骤。

```
helm install stable/cert-manager \
  --name cert-manager \
  --namespace kube-system
```

2、Rancher自动生成证书

默认情况下，Rancher会自动生成CA根证书并使用cert-manager颁发证书以访问Rancher server界面。

唯一的要求是将hostname配置为访问Rancher的域名地址，使用这种SSL证书配置方式需提前安装证书管理器。

```
helm install rancher-stable/rancher \
  --name rancher \
  --namespace cattle-system \
  --set hostname=rancher.com
```

rancher.com就是后面访问rancher的域名，需要在/etc/hosts文件中添加关联（所有主机）：

```
[root@node1 ~]$ echo "192.168.111.6 rancher.com" >> /etc/hosts
[root@node2 ~]$ echo "192.168.111.6 rancher.com" >> /etc/hosts
[root@node3 ~]$ echo "192.168.111.6 rancher.com" >> /etc/hosts
[root@nginx ~]$ echo "192.168.111.6 rancher.com" >> /etc/hosts
```

由于我们通过hosts文件来添加映射，所以需要为Agent Pod添加主机别名(/etc/hosts)：

```
kubectl -n cattle-system patch  deployments cattle-cluster-agent --patch '{
    "spec": {
        "template": {
            "spec": {
                "hostAliases": [
                    {
                        "hostnames":
                        [
                            "rancher.com"
                        ],
                            "ip": "192.168.111.6"
                    }
                ]
            }
        }
    }
}'
```

这一步如果马上执行，可能会报错：`Error from server (NotFound): deployments.extensions "cattle-cluster-agent" not found`，这个deployment是上一步install时创建的，比较慢，耐心等待一下，这个时候也可以先跳过这里，去到后边，简单配置一下，访问一下rancher的界面。

```
kubectl -n cattle-system patch  daemonsets cattle-node-agent --patch '{
    "spec": {
        "template": {
            "spec": {
                "hostAliases": [
                    {
                        "hostnames":
                        [
                            "rancher.com"
                        ],
                            "ip": "192.168.111.6"
                    }
                ]
            }
        }
    }
}'
```

5，登录rancher管理端

1、同样将刚刚的域名映射关系写入到Windows主机的hosts文件。

```
192.168.111.6 rancher.com
```

2、使用域名访问https://rancher.com

输入：admin/admin，进入首页界面。

刚进入，会看到一个问题。

根这个问题的原因就是刚刚创建的`cattle-cluster-agent`还没有被创建成功，同样耐心等待即可。这个时候可以随便点点看看先。这个过程与自己的网络有关，这时也可以在node1主机上，通过如下命令进行一个监控。

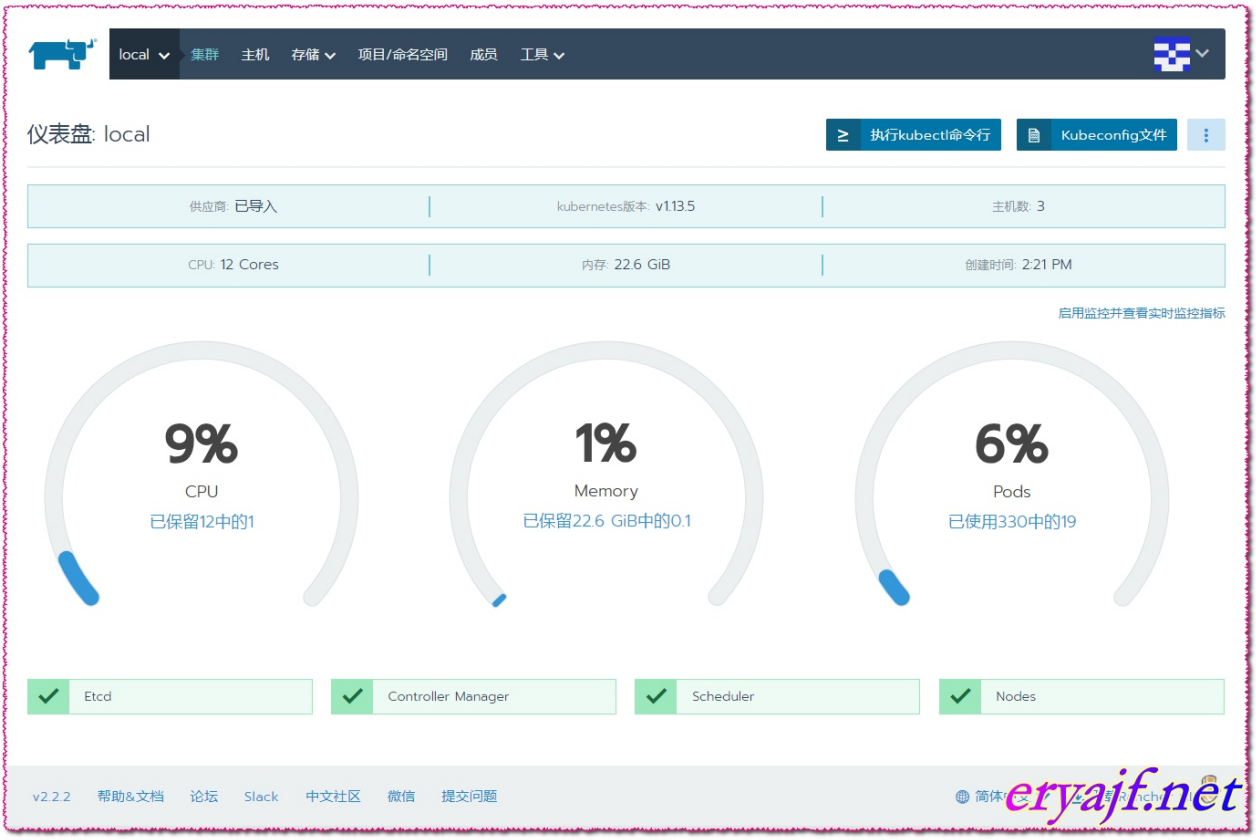
```
[rancher@node1 ~]$ kubectl get -n cattle-system pod -w
NAME                       READY   STATUS    RESTARTS   AGE
rancher-bdf49fb9-7qhgp     1/1     Running   1          12m
rancher-bdf49fb9-hf6tm     1/1     Running   0          12m
rancher-bdf49fb9-xmbv7     1/1     Running   1          12m



cattle-cluster-agent-7b54db4bc8-r4blg   0/1   Pending   0     0s
cattle-cluster-agent-7b54db4bc8-r4blg   0/1   Pending   0     0s
cattle-cluster-agent-7b54db4bc8-r4blg   0/1   ContainerCreating   0     0s
cattle-node-agent-mskmb   0/1   Pending   0     0s
cattle-node-agent-2cmww   0/1   Pending   0     0s
cattle-node-agent-kkpvn   0/1   Pending   0     0s
cattle-node-agent-mskmb   0/1   ContainerCreating   0     0s
cattle-node-agent-kkpvn   0/1   ContainerCreating   0     0s
cattle-node-agent-2cmww   0/1   ContainerCreating   0     0s
```

在我这里，等了十分钟左右，才开始正式的部署。这个时候，可以返回到上边，将那两条命令导入进去。

操作之后，然后再看rancher，就不会报连接问题了。

到这里，基本上安装步骤也就完成了，可以随便点点看看界面里边的各项功能什么的。

最后再来个整体界面的定妆照：


local
集群
主机
存储
项目/命名空间
成员
工具
仪表盘: local
执行kubectl命令行
Kubeconfig文件
供应商: 已导入
kubernetes版本: v1.13.5
主机数: 3
CPU: 12 Cores
内存: 22.6 GiB
创建时间: 2:21 PM
启用监控并查看实时监控指标
9%
CPU
已保留12中的1
1%
Memory
已保留22.6 GiB中的0.1
6%
Pods
已使用330中的19
Etcd
Controller Manager
Scheduler
Nodes
v2.2.2
帮助&文档
论坛
Slack
中文社区
微信
提交问题
eryajf.net