

# Rancher

---

## Rancher

### 前言

- 1.Rancher是什么
- 2.Rancher可以做什么
- 3.Rancher和Kubernetes的关系?
- 4.Rancher类似平台对比

### 架构图

#### Rancher 部署指南

##### ※ 软硬件要求

- 1.基础环境搭建
- 2.Rancher安装
- 3.WEB访问

#### kubeadm 部署 kubernetes

##### ※ 软硬件要求

- 1.基础环境搭建
- 2.安装containerd
- 3.安装kubeadm
  - 3.1 Master节点初始化
  - 3.2 Work节点加入集群
  - 3.3 为Master节点配置kubectl
  - 3.4 安装Addons 安装
  - 3.5 查看集群
- 4.将K8S 集群导入到Rancher中
  - 4.1 在Master拥有kubectl节点执行
  - 4.2如果下载镜像失败，可以修改镜像仓库

#### Rancher 部署 kubernetes

##### ※ 软硬件要求

- 1.基础环境搭建
- 2.WEB 界面创建集群
- 3.Master注册到集群中
  - 3.1 查看服务状态
  - 3.2 查看日志
- 4.二进制ctl命令
  - 4.1 临时生效
  - 4.2 当前用户永久生效
  - 4.3 所有用户生效

### FAQ

## 前言

---

### 1.Rancher是什么

Rancher是一个Kubernetes管理工具，可以在任何地方部署、运行和管理K8s集群。

Rancher可以为集群提供更精细的监控和告警，并且可以将告警信息发送到外部提供商。同时Rancher可以对接已有的CI/CD系统，也可以使用Rancher提供的Fleet工具，自动部署和升级工作负载。

## 2.Rancher可以做什么

支持部署集群和管理现有集群

支持K8s多集群进行集中认证、访问控制和监控

支持对接AD（Active Directory）等域认证

支持一站式查看所有集群的运行状态和容量

支持Helm仓库

支持日志、监控、告警、服务网格管理

## 3.Rancher和Kubernetes的关系？

Kubernetes是一个开源的容器编排系统，用于管理多个主机上的容器化应用。

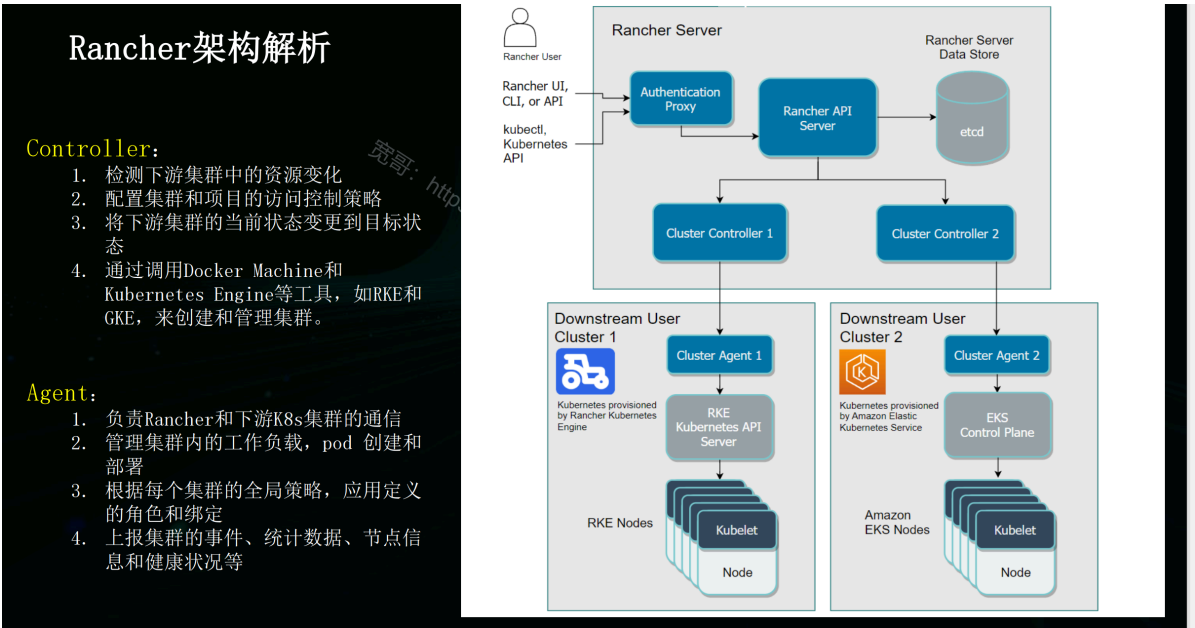
Rancher是一个企业级多集群Kubernetes管理平台，可以对存在于本地、私有云、公有云上的Kubernetes做统一管理。

Rancher是建于Kubernetes之上的，并且提供了一个友好的管理页面，帮助技术人员轻松部署、管理和监控K8s集群。同时Rancher还提供了很多附加的功能和工具，让技术人员更简单的管理和维护Kubernetes集群

## 4.Rancher类似平台对比

 <b>Rancher</b> 在集群管理方面表现出色，提供了强大的多集群管理能力，使得用户可以轻松管理和维护多个Kubernetes集群。同时Rancher具备多种基础设施的整合能力，Rancher在需要跨多个环境或云平台的场景中表现尤为出色，适用于大型企业和复杂的应用场景	 <b>KubeSphere</b> 是以应用为中心的容器平台，更注重对应用的管理和部署。对于开发团队来说，KubeSphere提供了一个简单易用的平台，并且提供了一个应用商店，用户可以选择需要的应用程序，并通过简单的操作可以快速构建和部署应用。	 <b>自研平台</b> 可以根据具体的业务需求或者操作习惯进行定制开发，不仅能满足特定的功能需求，还更加贴近业务的实际需要。
--	---	---

# 架构图



## Rancher 部署指南

### ※ 软硬件要求

VMWare	17 PRO	
机器配置	4C/4G/100G+	生产环境8C/16G
操作系统	Rocky8.10	<a href="https://download.rockylinux.org/pub/rocky/8/isos/x86_64/Rocky-8.10-x86_64-dvd1.iso">https://download.rockylinux.org/pub/rocky/8/isos/x86_64/Rocky-8.10-x86_64-dvd1.iso</a>
基础环境	Docker/K8s	
rancher	v2.10.1	<code>docker pull registry.cn-hangzhou.aliyuncs.com/rancher/rancher:v2.10.1</code>

### 1.基础环境搭建

```
1 systemctl disable --now firewalld
2 setenforce 0
3
4 sed -i 's#SELINUX=enforcing#SELINUX=disabled#g' /etc/sysconfig/selinux
5 sed -i 's#SELINUX=enforcing#SELINUX=disabled#g' /etc/selinux/config
6
7 swapoff -a && sysctl -w vm.swappiness=0
8 sed -ri '/^[^#]*swap/s@^@#/' /etc/fstab
9
10 yum install wget jq psmisc vim net-tools telnet yum-utils device-mapper-
    persistent-data lvm2 git -y
11 yum-config-manager --add-repo https://mirrors.aliyun.com/docker-
    ce/linux/centos/docker-ce.repo
12
13 sudo modprobe overlay
14 sudo modprobe br_netfilter
15
```

```

16 cat > /etc/modules-load.d/rancher-modules.conf <<EOF
17 overlay
18 br_netfilter
19 EOF
20
21 cat > /etc/sysctl.d/99-kubernetes-cri.conf <<EOF
22 net.bridge.bridge-nf-call-iptables = 1
23 net.ipv4.ip_forward = 1
24 net.bridge.bridge-nf-call-ip6tables = 1
25 EOF
26 sudo sysctl --system
27
28 yum install docker-ce -y
29 systemctl enable docker --now
30
31 cat > /etc/docker/daemon.json <<EOF
32 {
33     "registry-mirrors": [
34         "https://docker.nastool.de",
35         "https://docker.lms.run",
36         "https://docker.lpanel.live",
37         "https://hub1.nat.tf",
38         "https://docker.lpanel.top",
39         "https://dockerpull.org",
40         "https://docker.13140521.xyz"
41     ]
42 }
43 EOF
44 systemctl daemon-reload
45 systemctl restart docker

```

## 2.Rancher安装

```

1 mkdir -p /data/rancher
2 docker pull registry.cn-hangzhou.aliyuncs.com/rancher/rancher:v2.10.1
3 docker run -d \
4     --name rancher \
5     --restart=unless-stopped \
6     --privileged \
7     -v /data/rancher:/var/lib/rancher \
8     -p 443:443 \
9     -p 80:80 \
10    registry.cn-beijing.aliyuncs.com/dotballo/rancher:v2.10.1

```

### ⚠ Caution

注意 rancher 容器的状态为运行并不代表他现在就可以 WEB 访问,需要等待 集群状态初始化完成

使用 `docker logs -f rancher` 可以进行查看, 这个时间可能会持续10~20分钟

### 3.WEB访问

浏览器访问 `https://192.168.0.104:443`

```
1 # 查看密码
2 [root@Rancher data]# docker logs rancher 2>&1 | grep "Bootstrap Password:"
3 2025/01/03 10:56:13 [INFO] Bootstrap Password:
    7z7k6dcdctz7s99dqw6gpf9kpr42htqd42wlvhk2nvkwmfncld5x2q
```

## kubeadm 部署 kubernetes

### ※ 软硬件要求

Rocky8.10	<code>https://download.rockylinux.org/pub/rocky/8/isos/x86_64/Rocky-8.10-x86_64-dvd1.iso</code>	
K8S-Master	192.168.0.105	
K8S-Works	192.168.0.106	
kubeadm	vversion--1.31.4	
kubelet	vversion--1.31.4	
kubectl	vversion--1.31.4	

### 修改主机名称

```
1 hostnamectl set-hostname K8S-Master
2 hostnamectl set-hostname K8S-works
```

## 1.基础环境搭建

```
1 systemctl disable --now firewalld
2 systemctl disable --now dnsmasq
3
4 setenforce 0
5 sed -i 's#SELINUX=enforcing#SELINUX=disabled#g' /etc/sysconfig/selinux
6 sed -i 's#SELINUX=enforcing#SELINUX=disabled#g' /etc/selinux/config
7
8 swapoff -a && sysctl -w vm.swappiness=0
9 sed -ri '/^[^#]*swap/s@^#@#' /etc/fstab
10
11 sed -e 's|^mirrorlist=|#mirrorlist=|g' \
12     -e
    's|^#baseurl=http://dl.rockylinux.org/$contentdir|baseurl=https://mirrors.aliyun.com/rockylinux|g' \
13     -i.bak \
14     /etc/yum.repos.d/Rocky-*.repo
15 dnf makecache
16
```

```

17 # -e 's|^mirrorlist=|#mirrorlist=|g': 注释掉所有 mirrorlist 行。
18 # -e
   's|^#baseurl=http://dl.rockylinux.org/$contentdir|baseurl=https://mirrors.aliyun.com/rockylinux|g': 取消注释并替换 baseurl 地址为阿里云镜像源地址。
19 # -i.bak: 对文件进行原地修改, 同时备份 .bak 文件。
20
21 yum install wget jq psmisc vim net-tools telnet yum-utils device-mapper-
persistent-data lvm2 git -y
22
23 #注意版本号的更换, 需要安装什么版本的k8s, 按照实际情况进行修改即可
24 cat > /etc/yum.repos.d/kubernetes.repo <<EOF
25 [kubernetes]
26 name=Kubernetes
27 baseurl=https://mirrors.aliyun.com/kubernetes-new/core/stable/v1.31/rpm/
28 enabled=1
29 gpgcheck=1
30 gpgkey=https://mirrors.aliyun.com/kubernetes-
new/core/stable/v1.31/rpm/repodata/repomd.xml.key
31 EOF
32
33 yum-config-manager --add-repo https://mirrors.aliyun.com/docker-
ce/linux/centos/docker-ce.repo

```

## 2.安装containerd

```

1 yum install containerd.io -y
2 cat > /etc/modules-load.d/containerd.conf <<EOF
3 overlay
4 br_netfilter
5 EOF
6
7 sudo modprobe overlay
8 sudo modprobe br_netfilter
9
10 cat > /etc/sysctl.d/99-kubernetes-cri.conf <<EOF
11 net.bridge.bridge-nf-call-iptables = 1
12 net.ipv4.ip_forward = 1
13 net.bridge.bridge-nf-call-ip6tables = 1
14 EOF
15 sudo sysctl --system
16
17 sudo mkdir -p /etc/containerd
18 containerd config default | sudo cat > /etc/containerd/config.toml
19
20 sed -i 's#SystemdCgroup = false#SystemdCgroup =true#g'
/etc/containerd/config.toml
21 sed -i 's#k8s.gcr.io/pause#registry.cn-
hangzhou.aliyuncs.com/google_containers/pause#g' /etc/containerd/config.toml
22 sed -i 's#registry.gcr.io/pause#registry.cn-
hangzhou.aliyuncs.com/google_containers/pause#g' /etc/containerd/config.toml
23 sed -i 's#registry.k8s.io/pause#registry.cn-
hangzhou.aliyuncs.com/google_containers/pause#g' /etc/containerd/config.toml
24
25 # 启动Containerd
26 systemctl daemon-reload

```

### 3.安装kubeadm

```

1 yum install kubeadm-1.31.* kubelet-1.31.* kubectl-1.31.* -y
2 systemctl enable --now kubelet
3
4 # 下载镜像
5 kubeadm config images pull --image-repository registry.cn-
  hangzhou.aliyuncs.com/google_containers --kubernetes-version 1.31.4
6
7 [root@localhost ~]# kubeadm config images pull \
8 > --image-repository registry.cn-hangzhou.aliyuncs.com/google_containers --
  kubernetes-version 1.31.4
9 [config/images] Pulled registry.cn-
  hangzhou.aliyuncs.com/google_containers/kube-apiserver:v1.31.4
10 [config/images] Pulled registry.cn-
  hangzhou.aliyuncs.com/google_containers/kube-controller-manager:v1.31.4
11 [config/images] Pulled registry.cn-
  hangzhou.aliyuncs.com/google_containers/kube-scheduler:v1.31.4
12 [config/images] Pulled registry.cn-
  hangzhou.aliyuncs.com/google_containers/kube-proxy:v1.31.4
13 [config/images] Pulled registry.cn-
  hangzhou.aliyuncs.com/google_containers/coredns:v1.11.3
14 [config/images] Pulled registry.cn-
  hangzhou.aliyuncs.com/google_containers/pause:3.10
15 [config/images] Pulled registry.cn-
  hangzhou.aliyuncs.com/google_containers/etcd:3.5.15-0

```

```

[root@localhost ~]# yum install kubeadm-1.31.* kubelet-1.31.* kubectl-1.31.* -y
Last metadata expiration check: 6:15:04 ago on Sun 25 Jan 2025 10:23:31 PM CST.
Package kubeadm-1.31.0-1.el9.x86_64 is already installed.
Package kubelet-1.31.0-1.el9.x86_64 is already installed.
Package kubectl-1.31.0-1.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@localhost ~]#

```

#### 3.1 Master节点初始化

```

1 kubeadm init --apiserver-advertise-address 192.168.0.105 --image-repository
  registry.cn-hangzhou.aliyuncs.com/google_containers --cri-socket
  "unix:///var/run/containerd/containerd.sock" --kubernetes-version 1.31.4
2 # 版本需要替换为实际版本号
3
4
5 Your Kubernetes control-plane has initialized successfully!
6
7 To start using your cluster, you need to run the following as a regular
  user:
8
9   mkdir -p $HOME/.kube
10   sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
11   sudo chown $(id -u):$(id -g) $HOME/.kube/config
12
13 Alternatively, if you are the root user, you can run:
14
15   export KUBECONFIG=/etc/kubernetes/admin.conf
16
17 You should now deploy a pod network to the cluster.

```

```

18 Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
19   https://kubernetes.io/docs/concepts/cluster-administration/addons/
20
21 Then you can join any number of worker nodes by running the following on
22 each as root:
23
24 kubectl join 192.168.0.105:6443 --token su6wcn.5qw19ib51c9vb3xm \
25   --discovery-token-ca-cert-hash
   sha256:c7d4d20d51f4cc24d8745c3561f40d060ba31d1bb205c9b4f02923b7457aadde

```

## 3.2 Work节点加入集群

```

1 kubectl join 192.168.0.105:6443 --token su6wcn.5qw19ib51c9vb3xm \
2   --discovery-token-ca-cert-hash
   sha256:c7d4d20d51f4cc24d8745c3561f40d060ba31d1bb205c9b4f02923b7457aadde

```

## 3.3 为Master节点配置kubectl

```

1 mkdir -p $HOME/.kube
2 sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
3 sudo chown $(id -u):$(id -g) $HOME/.kube/config

```

## 3.4 安装Addons 安装

```

1 git clone https://gitee.com/dukuan/k8s-ha-install.git
2 cd k8s-ha-install
3 kubectl create -f calico.yaml
4 kubectl create -f krm.yaml

```

## 3.5 查看集群

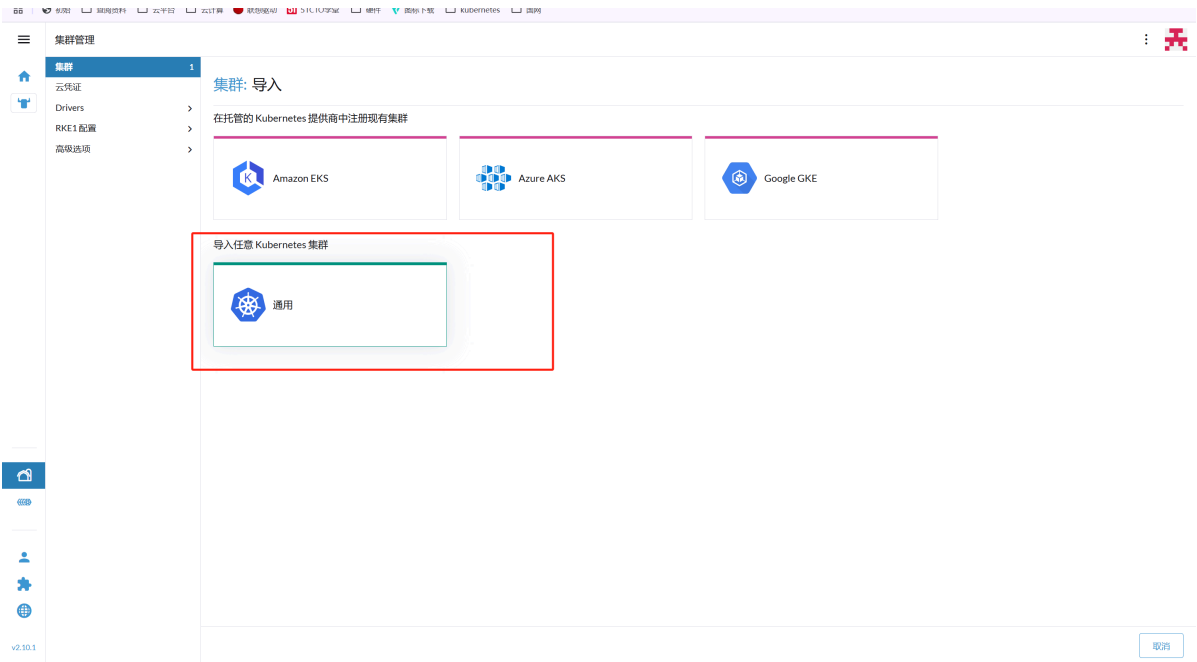
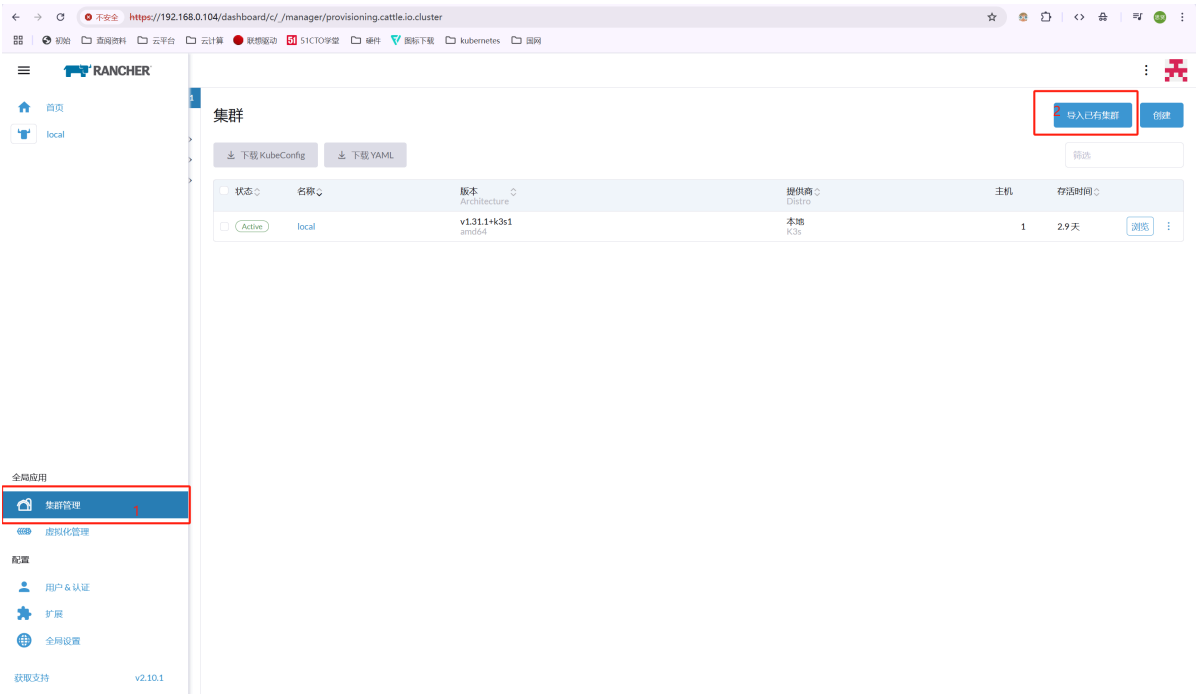
```

1 [root@K8S-Master k8s-ha-install]# kubectl get node -owide
2 NAME          STATUS    ROLES          AGE    VERSION    INTERNAL-IP    EXTERNAL-IP    OS-IMAGE          KERNEL-VERSION    CONTAINER-RUNTIME
3 k8s-master    Ready    control-plane   32h    v1.31.4    192.168.0.105  <none>         Rocky Linux 8.10 (Green Obsidian) 4.18.0-553.el8_10.x86_64 containerd://1.6.32
4 k8s-works     Ready    <none>          32h    v1.31.4    192.168.0.106  <none>         Rocky Linux 8.10 (Green Obsidian) 4.18.0-553.el8_10.x86_64 containerd://1.6.32

```



# 4.将K8S 集群导入到Rancher中

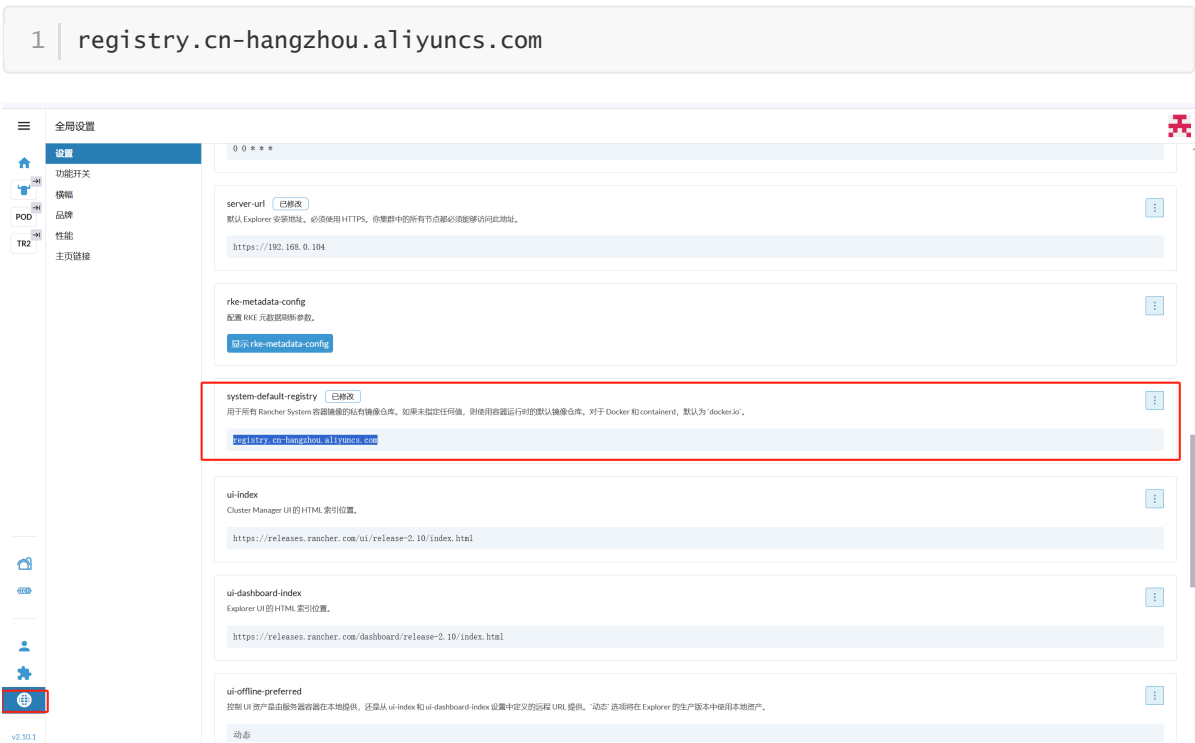


The screenshot displays the Rancher UI interface for cluster management. The left sidebar shows the navigation menu with options like '集群' (Clusters), '云凭证' (Cloud Credentials), 'Drivers', 'RKE1 配置' (RKE1 Configuration), and '高级选项' (Advanced Options). The main content area is titled '集群: 导入通用' (Cluster: Import General). It shows a form for importing a cluster, with fields for '集群名称' (Cluster Name) set to 'Prod' and '集群描述' (Cluster Description) set to '生产环境' (Production Environment). Below this, there's a section for '成员角色' (Member Roles) with a table showing 'Default Admin (admin)' and 'Cluster Owner'. At the bottom right, there are buttons for '取消' (Cancel), '以 YAML 文件编辑' (Edit with YAML file), and '创建' (Create). The second part of the screenshot shows the '集群: prod' (Cluster: prod) page in a 'Pending' state. It includes a warning message about the cluster being in a pending state and a section for '配置日志' (Configuration Log) with instructions on how to import the cluster using kubectl and curl commands. The curl command is highlighted with a red box.

## 4.1 在Master拥有kubectI节点执行

```
1 #将K8S集群导入到rancher中
2 curl --insecure -sfl
https://192.168.0.104/v3/import/51tkl1tkg2fb4ftm54ksfmhm81412tv7jjglnb52q5q
dhbm8smhfc_c-m-kwr645z9.yaml | kubectl apply -f -
3
4 #查看所有cattle-system资源
5 kubectl get all -n cattle-system
6
7 #可能会存在下载失败情况，因为这个rancher-agent镜像需要连接外网才可以下载,可以修工修改
deploy-image，替换为以下镜像 registry.cn-beijing.aliyuncs.com/dotballo/rancher-
agent:v2.10.1
8 kubectl edit deploy cattle-cluster-agent -n cattle-system
9
10 %s/rancher\ /rancher-agent:v2.10.1/registry.cn-
beijing.aliyuncs.com\ /dotballo\ /rancher-agent:v2.10.1/g
```

## 4.2如果下载镜像失败，可以修改镜像仓库



## Rancher 部署 kubernetes

### ※ 软硬件要求

Rocky8.10	<a href="https://download.rockylinux.org/pub/rocky/8/isos/x86_64/Rocky-8.10-x86_64-dvd1.iso">https://download.rockylinux.org/pub/rocky/8/isos/x86_64/Rocky-8.10-x86_64-dvd1.iso</a>	
Rke2-k8s-Master	192.168.0.107	
基础环境	docker-ce	
Rancher-server-agent	rancher-system-agent version v0.3.11	
rke2	containerd	

### 修改主机名称

```
1 | hostnamectl set-hostname Rke2-K8s-Master
```

### 1.基础环境搭建

```
1 | systemctl disable --now firewalld
2 | setenforce 0
3 |
4 | sed -i 's#SELINUX=enforcing#SELINUX=disabled#g' /etc/sysconfig/selinux
5 | sed -i 's#SELINUX=enforcing#SELINUX=disabled#g' /etc/selinux/config
6 |
7 | swapoff -a && systemctl -w vm.swappiness=0
8 | sed -ri '/^[^#]*swap/s@^#@#' /etc/fstab
```

```

9
10 yum install wget jq psmisc vim net-tools telnet yum-utils device-mapper-
    persistent-data lvm2 git -y
11 yum-config-manager --add-repo https://mirrors.aliyun.com/docker-
    ce/linux/centos/docker-ce.repo
12
13 sudo modprobe overlay
14 sudo modprobe br_netfilter
15
16 cat > /etc/modules-load.d/rancher-modules.conf <<EOF
17 overlay
18 br_netfilter
19 EOF
20
21 cat > /etc/sysctl.d/99-kubernetes-cri.conf <<EOF
22 net.bridge.bridge-nf-call-iptables = 1
23 net.ipv4.ip_forward = 1
24 net.bridge.bridge-nf-call-ip6tables = 1
25 EOF
26 sudo sysctl --system
27
28 yum install docker-ce -y
29 systemctl enable docker --now

```

## 2.WEB 界面创建集群

集群: 自定义 - test-rke Error

命名空间: fleet-default 存活时间: 35 分钟

failed to create fleet-default/test-rke cluster: x-k8s.io/v1beta1, Kind=Cluster for rke-cluster fleet-default/test-rke: Internal error occurred: failed calling webhook "default.cluster.cluster.x-k8s.io": failed to call webhook: Post "https://capi-webhook-service.cattle.io/provisioning-capi-system.svc:443/mutate-cluster.x-k8s.io/v1beta1-cluster?timeout=10s": service "capi-webhook-service" not found

描述: 使用rke2模拟创建kubernetes集群  
提供商: RKE2

主机 配置日志 注册 Snapshots 状态 最近事件 相关资源

**Step 1**

节点角色  
选择节点在集群中的角色。在集群中，每个角色都需要至少一个节点。

☒ ETCD ☒ Control Plane ☒ Worker

展开高级选项

**Step 2**

注册命令  
在每台要注册的 Linux 主机上运行此命令。

```
curl --insecure -fL https://192.168.0.104/system-agent-install.sh | sudo sh -s -- --server https://192.168.0.104 --label 'cattle.io/os=linux' --token v4tkn1zzh4sgz5419q62mcr8wnf9121n8n6gj148xqf8g8h1ps7ft2 --ca-checksum b78b33cadd74d182851ff32ed3f6438acc392a1c175bcff0db452b87567f0153 --etcd --controlplane --worker
```

☒ 不安全: 如果你的服务器具有自签名证书, 选择此项以跳过 TLS 验证。

在每台要注册的 Windows 主机中的 PowerShell 上运行此命令。Windows 节点只能充当 worker。

在用于添加 Windows worker 的注册命令显示之前, 集群必须已启动并运行 Linux ETCD, Control Plane, 以及 worker 节点。

## 3.Master注册到集群中

如果是添加worker节点，需要把ETCD和ControlPlane去掉勾选，如果是高可用需要至少两台master

```

1 curl --insecure -fL https://192.168.0.104/system-agent-install.sh | sudo sh -s
  - --server https://192.168.0.104 --label 'cattle.io/os=linux' --token
    v4tkn1zzh4sgz5419q62mcr8wnf9121n8n6gj148xqf8g8h1ps7ft2 --ca-checksum
    b78b33cadd74d182851ff32ed3f6438acc392a1c175bcff0db452b87567f0153 --etcd --
    controlplane --worker

```

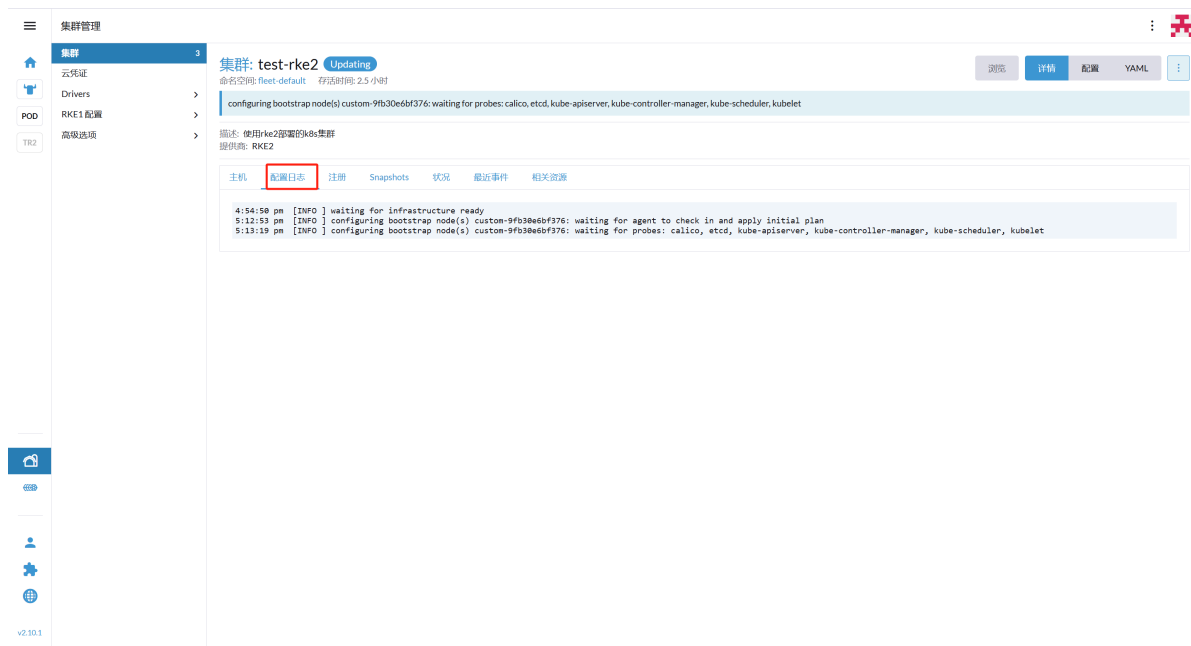
## 3.1 查看服务状态

```
1 | systemctl status rancher-system-agent
```

## 3.2 查看日志

```
1 | tail -f /var/log/message
```

### WEB可以在这里看到部署日志



The screenshot shows the Rancher web interface. On the left is a sidebar with navigation options: 集群管理 (Cluster Management), 云凭证 (Cloud Credentials), Drivers, POD, RKE1 配置 (RKE1 Configuration), 高级选项 (Advanced Options), and TR2. The main panel displays details for a cluster named 'test-rke2' with a status of 'Updating'. Below the cluster name, there's a description: '命名空间: fleet-default 存活时间: 2.5 小时'. A message bar indicates 'configuring bootstrap node(s) custom-9fb30e6bf376: waiting for probes: calico, etcd, kube-apiserver, kube-controller-manager, kube-scheduler, kubelet'. Below this, there's a table with tabs: 主机 (Hosts), 部署日志 (Deployment Logs) - which is highlighted with a red box, 注册 (Registration), Snapshots, 状况 (Status), 最近事件 (Recent Events), and 相关资源 (Related Resources). The '部署日志' tab shows a log entry: '4:54:58 pm [INFO] waiting for infrastructure ready' and '5:12:53 pm [INFO] configuring bootstrap node(s) custom-9fb30e6bf376: waiting for agent to check in and apply initial plan'. At the bottom of the log entry, it says '5:13:19 pm [INFO] configuring bootstrap node(s) custom-9fb30e6bf376: waiting for probes: calico, etcd, kube-apiserver, kube-controller-manager, kube-scheduler, kubelet'.

## 4.二进制ctl命令

### 4.1 临时生效

```
1 | export CRI_CONFIG_FILE=/var/lib/rancher/rke2/agent/etc/crictl.yaml
2 | ln -s /var/lib/rancher/rke2/bin/crictl /usr/local/bin
3 | crictl ps
4 |
5 | export KUBECONFIG=/etc/rancher/rke2/rke2.yaml
6 | ln -s /var/lib/rancher/rke2/bin/kubect1 /usr/local/bin
7 | kubect1 get nodes
```

### 4.2 当前用户永久生效

```
1 | cat >> ~/.bashrc <<EOF
2 | export CRI_CONFIG_FILE=/var/lib/rancher/rke2/agent/etc/crictl.yaml
3 | export KUBECONFIG=/etc/rancher/rke2/rke2.yaml
4 | EOF
5 | source ~/.bashrc
6 |
7 | ln -s /var/lib/rancher/rke2/bin/crictl /usr/local/bin
8 | ln -s /var/lib/rancher/rke2/bin/kubect1 /usr/local/bin
```

## 4.3 所有用户生效

```
1 cat >> /etc/profile <<EOF
2 export CRI_CONFIG_FILE=/var/lib/rancher/rke2/agent/etc/crictl.yaml
3 export KUBECONFIG=/etc/rancher/rke2/rke2.yaml
4 EOF
5 source /etc/profile
6
7 ln -s /var/lib/rancher/rke2/bin/crictl /usr/local/bin
8 ln -s /var/lib/rancher/rke2/bin/kubectl /usr/local/bin
```

## FAQ

如果集群时间中提示pause镜像问题，请进入rancher容器里面下载一下pause镜像

```
1 ctr --address /run/k3s/containerd/containerd.sock -n k8s.io i pull
  registry.cn-hangzhou.aliyuncs.com/rancher/mirrored-pause:3.6 ; ctr --address
  /run/k3s/containerd/containerd.sock -n k8s.io i tag registry.cn-
  hangzhou.aliyuncs.com/rancher/mirrored-pause:3.6 docker.io/rancher/mirrored-
  pause:3.6
```