Kubernetes高可用 & Rancher & KubeOperator

Kubernetes高可用架构：

部署

基本要求:

```
关闭selinux（非必须）
关闭防火墙（生产不推荐）
修改hosts文件
关闭swap（必须）
echo "swapoff -a" >> /etc/profile && source /etc/profile
修改内核参数（必须）
docker部署完成（必须）
ipvsadm安装，keepalived安装（本次部署使用ipvs）
    ipvs vs iptables:
        ipvs工作在内核空间，效率更高，iptables规则随pod数量增加，效率会降低
```

安装组件`yum install -y ipvsadm keepalived`

加载内核模块`vim /etc/sysconfig/modules/ipvs.modules`

```
modprobe -- ip_vs
modprobe -- ip_vs_rr
modprobe -- ip_vs_wrr
modprobe -- ip_vs_sh
modprobe -- nf_conntrack_ipv4


chmod +x /etc/sysconfig/modules/ipvs.modules
. /etc/sysconfig/modules/ipvs.modules
```

编辑keepalived`vim /etc/keepalived/keepalived.conf`

```
! Configuration File for keepalived

global_defs {
    router_id LVS_DEVEL
}

vrrp_instance VI_1 {
    state MASTER
```

```
    interface ens33
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        10.10.100.10
    }
}

virtual_server 10.10.100.10 6443 {
    delay_loop 6
    lb_algo rr
    lb_kind DR
    persistence_timeout 50
    protocol TCP

    real_server 10.10.100.130 6443 {
        weight 1
        TCP_CHECK {
            connect_timeout 5
            nb_get_retry 3
            delay_before_retry 3
            connect_port 6443
        }
    }
    real_server 10.10.100.131 6443 {
        weight 1
        TCP_CHECK {
            connect_timeout 5
            nb_get_retry 3
            delay_before_retry 3
            connect_port 6443
        }
    }
    real_server 10.10.100.132 6443 {
        weight 1
        TCP_CHECK {
            connect_timeout 5
            nb_get_retry 3
            delay_before_retry 3
            connect_port 6443
        }
    }
}
```

编写kube-admin.yaml

ClusterConfiguration

kube-proxy Configuration

```yaml
apiVersion: kubeadm.k8s.io/v1beta2
kind: ClusterConfiguration
kubernetesVersion: 1.21.0
imageRepository: registry.cn-hangzhou.aliyuncs.com/google_containers
clusterName: kubecamp
controlPlaneEndpoint: "10.10.100.10:6443"
networking:
    dnsDomain: kubecmap.com
    podSubnet: 192.168.150.0/24
    serviceSubnet: 192.168.151.0/24
dns:
    type: CoreDNS
    imageRepository: coredns
    imageTag: 1.8.0

---
apiVersion: kubeproxy.config.k8s.io/v1alpha1
kind: KubeProxyConfiguration
mode: ipvs
```

执行命令：初始化集群`kubeadm init --config=kubeadm-config.yaml --upload-certs`

实质：

- etcd的高可用，建议生产使用独立集群（不与controller部署于相同节点）
- apiserver 高可用，需要外部负载均衡的支持，单独使用keepalived实际上是仅仅是伪高可用
- controller-manager 控制平面副本
- kube-scheduler 控制平面副本
- kubelet， container生命周期管理all nodes
- kube-proxy， 集群级分布式负载均衡

Rancher

Rancher 是为使用容器的公司打造的容器***管理平台***。Rancher 简化了使用 Kubernetes 的流程，开发者可以随处运行 Kubernetes（Run Kubernetes Everywhere），满足 IT 需求规范，赋能 DevOps 团队。

Rancher所能管理的集群类型：

- Rancher托管集群（可以管理集群全部生命周期及资源）
- 外部导入集群（管理能力受限）

特性：

- RBAC访问代理
- monitor and alert
- log
- 基于Helm的应用商店

- CI/CD Pipeline集成
- 跨集群应用支持
- 全局DNS
- services mesh
- 安全扫描
- 集群模板
- 策略管理（基于OPA）

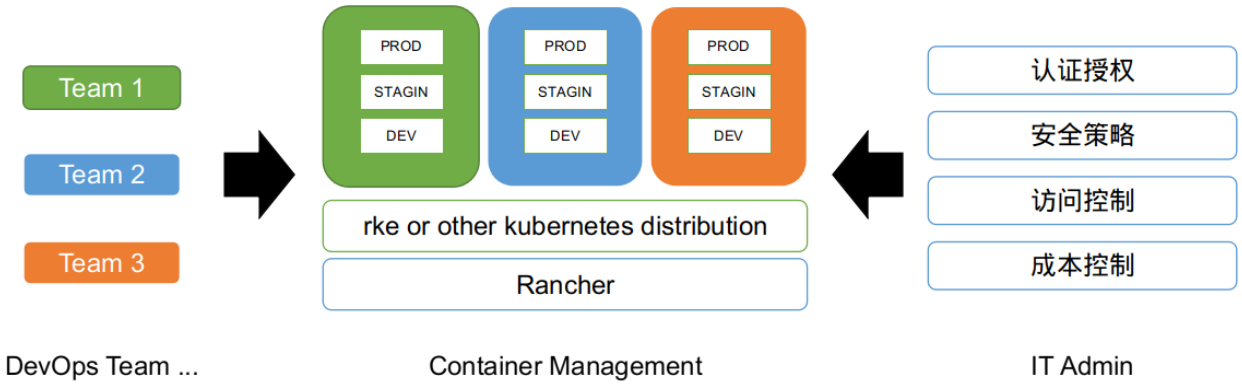
总而言之，Rancher 是一个全栈式的 Kubernetes 容器管理平台，也是一个可以在任何地方都能成功运行 Kubernetes 的工具。

内嵌IT规范

Rancher 支持集中化认证、权限控制、监控和管理所有 Kubernetes 集群。您可以使用 Rancher 完成以下操作：

- 使用活动目录（Active Directory）的认证信息访问云端 Kubernetes 集群，如 GKE、AKS、EKS 等
- 设置用户、用户组、项目组、集群、云服务的权限控制策略和安全策略
- 一站式监控您名下所有集群的健康状态

IT效率赋能

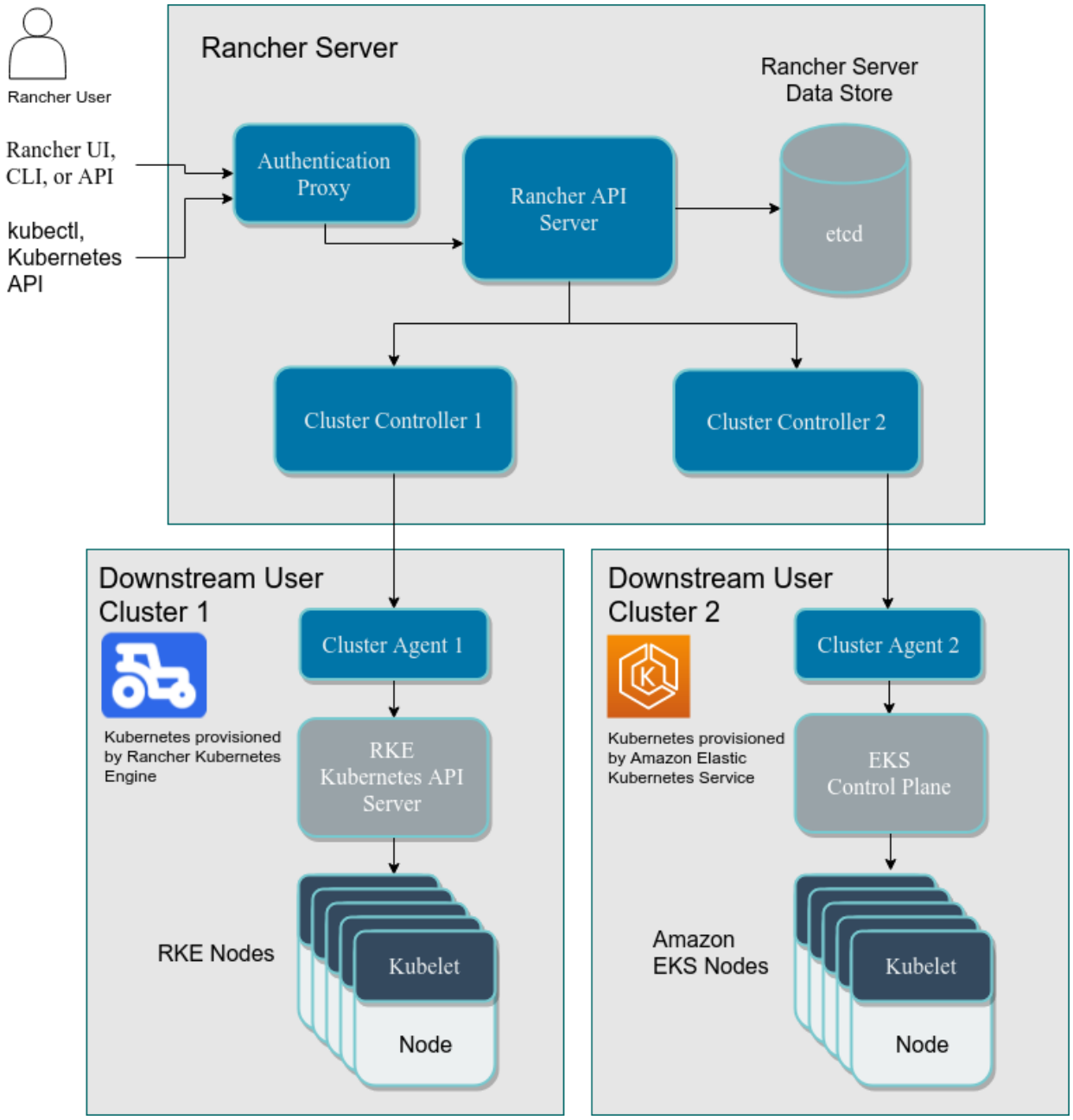


Rancher管理范围

功能	Rancher 启动的 Kubernetes 集群 (RKE 集群)	托管的 Kubernetes 集群	导入的 Kubernetes 集群
使用 kubectl 和 kubeconfig 文件访问集群	✓	✓	✓
添加集群成员	✓	✓	✓
编辑集群	✓	✓	*

功能	Rancher 启动的 Kubernetes 集群 (RKE 集群)	托管的 Kubernetes 集群	导入的 Kubernetes 集群
管理节点	✓	✓	✓
管理持久卷和存储类	✓	✓	✓
管理项目和命名空间	✓	✓	✓
使用应用商店	✓	✓	✓
使用配置工具（告警、通知、日志、监控和 Istio）	✓	✓	✓
克隆集群	✓	✓	
证书轮换的能力	✓		
备份您的 Kubernetes 集群的能力	✓		
恢复和还原 etcd 的能力	✓		
当集群不再能从 Rancher 访问时，清理 Kubernetes 组件	✓		
配置 Pod 安全策略	✓		
运行安全扫描	✓		

架构



部署架构

推荐使用高可用的Kuberentes集群（K3S）部署rancher

节点角色建议：

对于Rancher Server的集群，建议使用两个Server节点，不需要agnet节点

部署规模	集群	节点	vCPUs	内存	数据库规模
小	<=150	<=1500	2	8G	2cores,4GB+,1000IOPS
中	<=300	<=3000	4	16G	2cores,4GB+,1000IOPS

部署规模	集群	节点	vCPUs	内存	数据库规模
大	<=500	<=5000	8	32G	2cores,4GB+,1000IOPS
特大	<=1000	<=10000	16	64G	2cores,4GB+,1000IOPS
超大	<=2000	<=20000	32	128G	2cores,4GB+,1000IOPS

高可用的k3s

部署一台Mysql服务器

```
docker run --name mysql -d \
--network host \
-v "$PWD/mysql":/etc/mysql \
-v "$PWD/data":/var/lib/mysql \
-e MYSQL_ROOT_PASSWORD=devops \
mysql:5.7 \
--character-set-server=utf8mb4 \
--collation-server=utf8mb4_unicode_ci
```

该集群root未授权，无法远程连接，修改配置文件`/etc/mysql.conf.d/mysqld.cnf`添加如下内容：

```
skip-grant-tables
```

登录mysql，执行授权命令

```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'devops' WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED BY 'devops' WITH GRANT
OPTION;
```

删除`skip-grant-tables`选项，重启mysql

在其他节点执行命令：

```
curl -sfL http://rancher-mirror.cnrancher.com/k3s/k3s-install.sh | INSTALL_K3S_MIRROR=cn
sh -s - server --datastore-
endpoint="mysql://root:devops@tcp(10.10.100.201:3306)/kubernetes"
```

安装下载helm

添加rancher仓库

```
helm repo add rancher-stable http://rancher-mirror.oss-cn-beijing.aliyuncs.com/server-charts/stable
```

添加Namespace

```
kubectl create namespace cattle-system
```

获取Chart选择ssl选项，部署cert-manager支持

```
# 安装 CustomResourceDefinition 资源

kubectl apply --validate=false -f https://github.com/jetstack/cert-manager/releases/download/v1.0.4/cert-manager.crds.yaml

# **重要：**
# 如果您正在运行 Kubernetes v1.15 或更低版本，
# 则需要在上方的 kubectl apply 命令中添加 --validate=false`标志，
# 否则您将在 cert-manager 的 CustomResourceDefinition 资源中收到与
# x-kubernetes-preserve-unknown-fields 字段有关的验证错误。
# 这是一个良性错误，是由于 kubectl 执行资源验证的方式造成的。

# 为 cert-manager 创建命名空间

kubectl create namespace cert-manager

# 添加 Jetstack Helm 仓库

helm repo add jetstack https://charts.jetstack.io

# 更新本地 Helm chart 仓库缓存

helm repo update

# 安装 cert-manager Helm chart

helm install \
  cert-manager jetstack/cert-manager \
  --namespace cert-manager \
  --version v1.0.4
```

部署rancher

```
helm install rancher ./rancher \
 --namespace cattle-system \
 --set hostname=rancher.liuzhi.com \    # 这里注意虚拟化环境，宿主机配置hosts文件
 --set replicas=3
```

KubeOperator

扬言自己是kubernetes的发行版，我觉得他们对发行版是什么有点误解。

部署机要求

需求项	具体要求	参考（以CentOS7.6为例）
操作系统		支持 Docker 的 Linux OS
CPU架构	支持 x86_64 和 aarch64	uname -m
kernel版本	>=Linux 3.10.0-957.el7.x86_64	uname -sr
swap	关闭	swapoff -a && sed -i '/ swap / s/^(.*)$/#\1/g' /etc/fstab
防火墙	关闭	systemctl stop firewalld && systemctl disable firewalld
端口	所有节点防火墙必须放通 SSH（默认22）、80、8081-8083端口	firewall-cmd --zone=public --add-port=80/tcp --permanent
SELinux	关闭	setenforce 0 && sed -i "s/SELINUX=enforcing/SELINUX=disabled/g" /etc/selinux/config

集群节点要求

需求项	具体要求	参考（以CentOS7.6为例）

操作系统	CentOS/RHEL 7.4 - 7.9 Minimal or EulerOS 2.5（x86_64）or EulerOS 2.8（arm64）	cat /etc/redhat-release
CPU架构	支持 x86_64 和 aarch64	uname -m
kernel版本	>=Linux 3.10.0-957.el7.x86_64	uname -sr
swap	关闭。如果不满足，系统会有一定几率出现 io 飙升，造成 docker 卡死。kubelet 会启动失败 (可以设置 kubelet 启动参数 --fail-swap-on 为 false 关闭 swap 检查)	swapoff -a && sed -i '/ swap / s/^(.*)$/#\1/g' /etc/fstab
防火墙	关闭。Kubernetes 官方要求	systemctl stop firewalld && systemctl disable firewalld
SELinux	关闭	setenforce 0 && sed -i "s/SELINUX=enforcing/SELINUX=disabled/g" /etc/selinux/config
时区	所有服务器时区必须统一，建议设置为 Asia/Shanghai	timedatectl set-timezone Asia/Shanghai

在线安装部署：

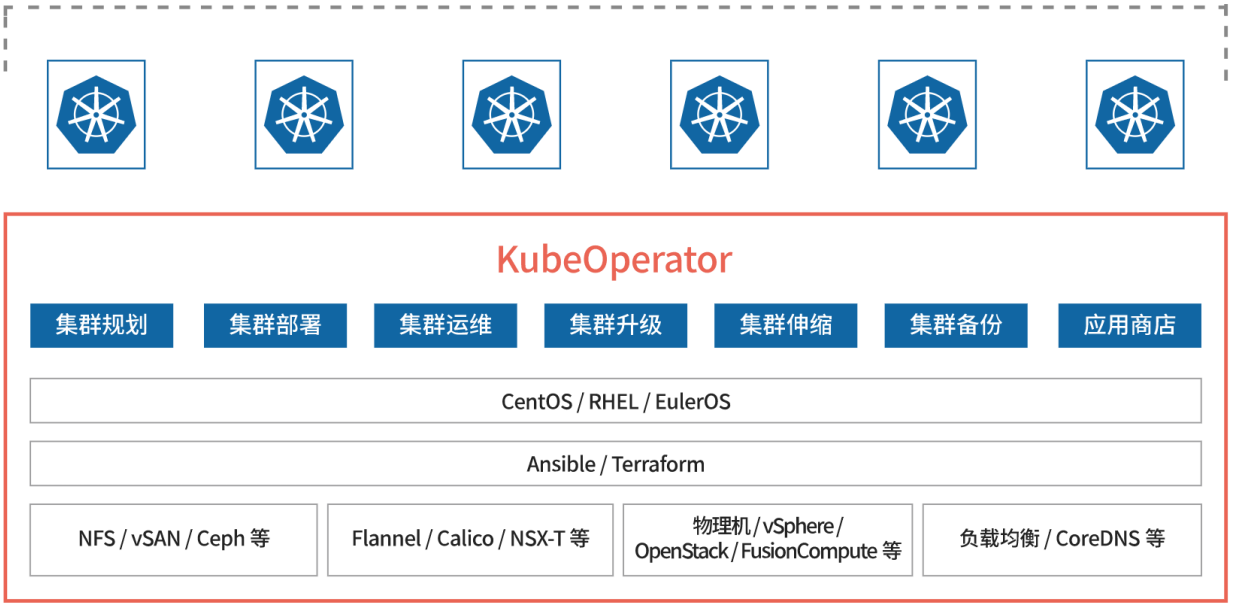
```
# 以 root 用户 ssh 登录目标服务器，执行如下命令
curl -sSL
https://github.com/KubeOperator/KubeOperator/releases/latest/download/quick_start.sh -o
quick_start.sh
bash quick_start.sh

# 查看部署状态
koctl status
```

访问方式：

```
地址：http://<ko服务器_ip>:80
用户名：admin
密码：kubeoperator@admin123
```

系统架构



组件关系