cri-o 安装

author: Uncle Dragon
date: 2021-06-28

安装
安装工具

安装

配置前置环境

```
# 创建 .conf 文件以在启动时加载模块
cat <<EOF | sudo tee /etc/modules-load.d/crio.conf
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter

# 配置 sysctl 参数，这些配置在重启之后仍然起作用
cat <<EOF | sudo tee /etc/sysctl.d/99-kubernetes-cri.conf
net.bridge.bridge-nf-call-iptables  = 1
net.ipv4.ip_forward                 = 1
net.bridge.bridge-nf-call-ip6tables = 1
EOF

sudo sysctl --system
```

安装:

操作系统	$OS
Ubuntu 20.04	`xUbuntu_20.04`
Ubuntu 18.04	`xUbuntu_18.04`

```
OS=xUbuntu_18.04
VERSION=1.20:1.20.0

cat <<EOF | sudo tee
/etc/apt/sources.list.d/devel:kubic:libcontainers:stable.list
deb
https://download.opensuse.org/repositories/devel:/kubic:/libcontai
ners:/stable/$OS/ /
EOF
cat <<EOF | sudo tee
/etc/apt/sources.list.d/devel:kubic:libcontainers:stable:cri-
o:$VERSION.list
deb
http://download.opensuse.org/repositories/devel:/kubic:/libcontain
ers:/stable:/cri-o:/$VERSION/$OS/ /
EOF

```

```
curl -L https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/$OS/Release.key | sudo apt-key --keyring /etc/apt/trusted.gpg.d/libcontainers.gpg add -
curl -L https://download.opensuse.org/repositories/devel:kubic:libcontainers:stable:cri-o:$VERSION/$OS/Release.key | sudo apt-key --keyring /etc/apt/trusted.gpg.d/libcontainers-cri-o.gpg add -

sudo apt-get update
sudo apt-get install cri-o cri-o-runc
```

启动

```
sudo systemctl daemon-reload
sudo systemctl enable crio --now
```

默认情况下，CRI-O 使用 systemd cgroup 驱动程序。要切换到 `cgroupfs` 驱动程序，或者编辑 `/ etc / crio / crio.conf` 或放置一个插件 在 `/etc/crio/crio.conf.d/02-cgroup-manager.conf` 中的配置：

```
[crio.runtime]
conmon_cgroup = "pod"
cgroup_manager = "cgroupfs"
```

另请注意更改后的 `conmon_cgroup`，将 CRI-O 与 `cgroupfs` 一起使用时， 必须将其设置为 `pod`。通常有必要保持 kubelet 的 cgroup 驱动程序配置 （通常透过 kubeadm 完成）和 CRI-O 一致

安装工具

- 使用 wget

```
VERSION="v1.21.0"
wget https://github.com/kubernetes-sigs/cri-tools/releases/download/$VERSION/crictl-$VERSION-linux-amd64.tar.gz
sudo tar zxvf crictl-$VERSION-linux-amd64.tar.gz -C /usr/local/bin
rm -f crictl-$VERSION-linux-amd64.tar.gz
```

- 使用 curl

```
VERSION="v1.21.0"
curl -L https://github.com/kubernetes-sigs/cri-tools/releases/download/$VERSION/crictl-${VERSION}-linux-amd64.tar.gz --output crictl-${VERSION}-linux-amd64.tar.gz
sudo tar zxvf crictl-$VERSION-linux-amd64.tar.gz -C /usr/local/bin
rm -f crictl-$VERSION-linux-amd64.tar.gz
```

`crictl` 默认连接到 `unix:///var/run/dockershim.sock`

```
$ sudo cat <<EOF | sudo tee /etc/crictl.yaml
> runtime-endpoint: unix:///run/containerd/containerd.sock
> image-endpoint: unix:///run/containerd/containerd.sock
> timeout: 10
> debug: false
> EOF

```

这个是为了 k8s 使用containerd 而做的，后面配合k8s 再做具体的说明

下面是docker 和 containerd 的cli 工具使用常用命令对比

id	docker	ctr	crictl	备注
1	docker images	ctr images ls	crictl images	查看本地镜像
2	docker pull	ctr images pull	crictl pull	拉取镜像
3	docker run	ctr container run	-	运行容器
4	docker ps	ctr task ls	crictl ps	查看运行的人容器
5	docker rm	ctr container del	crictl rm	移除容器
6	docker exec	ctr task exec	crictl exec	进入容器
7	docker logs		crictl logs	查看容器日志