tags: worker, containerd

06-3. 部署 containerd 组件

- 06-3. 部署 containerd 组件
  - 下载和分发二进制文件
  - 创建和分发 containerd 配置文件
  - 创建 containerd systemd unit 文件
  - 分发 systemd unit 文件，启动 containerd 服务
  - 创建和分发 crictl 配置文件

containerd 实现了 kubernetes 的 Container Runtime Interface (CRI) 接口，提供容器运行时核心功能，如镜像管理、容器管理等，相比 dockerd 更加简单、健壮和可移植。

注意：

1. 如果没有特殊指明，本文档的所有操作均在 zhangjun-k8s01 节点上执行。
2. 如果想使用 docker，请参考附件 F.部署docker.md；
3. docker 需要与 flannel 配合使用，且先安装 flannel；

下载和分发二进制文件

下载二进制文件：

```
cd /opt/k8s/work
wget https://github.com/kubernetes-sigs/cri-tools/releases/download/v1.17.0/crictl-v1.17.0-linux-amd64.tar.gz \
  https://github.com/opencontainers/runc/releases/download/v1.0.0-rc10/runc.amd64 \
  https://github.com/containernetworking/plugins/releases/download/v0.8.5/cni-plugins-linux-amd64-v0.8.5.tgz \
  https://github.com/containerd/containerd/releases/download/v1.3.3/containerd-1.3.3.linux-amd64.tar.gz
```

解压：

```
cd /opt/k8s/work
mkdir containerd
tar -xvf containerd-1.3.3.linux-amd64.tar.gz -C containerd
tar -xvf crictl-v1.17.0-linux-amd64.tar.gz

mkdir cni-plugins
sudo tar -xvf cni-plugins-linux-amd64-v0.8.5.tgz -C cni-plugins

sudo mv runc.amd64 runc
```

分发二进制文件到所有 worker 节点：

```
cd /opt/k8s/work
source /opt/k8s/bin/environment.sh
for node_ip in ${NODE_IPS[@]}
  do
    echo ">>> ${node_ip}"
    scp containerd/bin/*  crictl  cni-plugins/*  runc  root@${node_ip}:/opt/k8s/bin
    ssh root@${node_ip} "chmod a+x /opt/k8s/bin/* && mkdir -p /etc/cni/net.d"
  done
```

创建和分发 containerd 配置文件

```
cd /opt/k8s/work
source /opt/k8s/bin/environment.sh
cat << EOF | sudo tee containerd-config.toml
version = 2
root = "${CONTAINERD_DIR}/root"
state = "${CONTAINERD_DIR}/state"

[plugins]
  [plugins."io.containerd.grpc.v1.cri"]
    sandbox_image = "registry.cn-beijing.aliyuncs.com/images_k8s/pause-amd64:3.1"
    [plugins."io.containerd.grpc.v1.cri".cni]
      bin_dir = "/opt/k8s/bin"
      conf_dir = "/etc/cni/net.d"
  [plugins."io.containerd.runtime.v1.linux"]
    shim = "containerd-shim"
    runtime = "runc"
    runtime_root = ""
    no_shim = false
    shim_debug = false
EOF
```

```
cd /opt/k8s/work
source /opt/k8s/bin/environment.sh
for node_ip in ${NODE_IPS[@]}
  do
    echo ">>> ${node_ip}"
    ssh root@${node_ip} "mkdir -p /etc/containerd/ ${CONTAINERD_DIR}/{root,state}"
    scp containerd-config.toml root@${node_ip}:/etc/containerd/config.toml
  done
```

创建 containerd systemd unit 文件

```
cd /opt/k8s/work
cat <<EOF | sudo tee containerd.service
[Unit]
Description=containerd container runtime
Documentation=https://containerd.io
After=network.target

[Service]
Environment="PATH=/opt/k8s/bin:/bin:/sbin:/usr/bin:/usr/sbin"
ExecStartPre=/sbin/modprobe overlay
ExecStart=/opt/k8s/bin/containerd
Restart=always
RestartSec=5
Delegate=yes
KillMode=process
OOMScoreAdjust=-999
LimitNOFILE=1048576
LimitNPROC=infinity
LimitCORE=infinity

[Install]
WantedBy=multi-user.target
EOF
```

分发 systemd unit 文件，启动 containerd 服务

```
cd /opt/k8s/work
source /opt/k8s/bin/environment.sh
for node_ip in ${NODE_IPS[@]}
  do
    echo ">>> ${node_ip}"
    scp containerd.service root@${node_ip}:/etc/systemd/system
    ssh root@${node_ip} "systemctl enable containerd && systemctl restart containerd"
  done
```

创建和分发 crictl 配置文件

crictl 是兼容 CRI 容器运行时的命令行工具，提供类似于 docker 命令的功能。具体参考官方文档。

```
cd /opt/k8s/work
cat << EOF | sudo tee crictl.yaml
runtime-endpoint: unix:///run/containerd/containerd.sock
image-endpoint: unix:///run/containerd/containerd.sock
timeout: 10
debug: false
EOF
```

分发到所有 worker 节点：

```
cd /opt/k8s/work
source /opt/k8s/bin/environment.sh
for node_ip in ${NODE_IPS[@]}
  do
    echo ">>> ${node_ip}"
    scp crictl.yaml root@${node_ip}:/etc/crictl.yaml
  done
```