tags: worker, docker

F. 部署 docker 组件

docker 运行和管理容器，kubelet 通过 Container Runtime Interface (CRI) 与它进行交互。

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，然后远程分发文件和执行命令；
2. 需要先安装 flannel，请参考附件 E.部署flannel网络.md；

安装依赖包

参考 07-0.部署worker节点.md。

下载和分发 docker 二进制文件

到 docker 下载页面 下载最新发布包：

```
cd /opt/k8s/work
wget https://download.docker.com/linux/static/stable/x86_64/docker-18.09.6.tgz
tar -xvf docker-18.09.6.tgz
```

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 ">>> ${node_ip}"
    scp docker/*  root@${node_ip}:/opt/k8s/bin/
    ssh root@${node_ip} "chmod +x /opt/k8s/bin/*"
  done
```

创建和分发 systemd unit 文件

```
cd /opt/k8s/work
cat > docker.service <<"EOF"
[Unit]
Description=Docker Application Container Engine
Documentation=http://docs.docker.io

[Service]
WorkingDirectory=##DOCKER_DIR##
Environment="PATH=/opt/k8s/bin:/bin:/sbin:/usr/bin:/usr/sbin"
EnvironmentFile=-/run/flannel/docker
ExecStart=/opt/k8s/bin/dockerd $DOCKER_NETWORK_OPTIONS
ExecReload=/bin/kill -s HUP $MAINPID
Restart=on-failure
RestartSec=5
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
Delegate=yes
KillMode=process

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

- EOF 前后有双引号，这样 bash 不会替换文档中的变量，如 `$DOCKER_NETWORK_OPTIONS` (这些环境变量是 systemd 负责替换的。)；
- dockerd 运行时会调用其它 docker 命令，如 docker-proxy，所以需要将 docker 命令所在的目录加到 PATH 环境变量中；
- flanneld 启动时将网络配置写入 `/run/flannel/docker` 文件中，dockerd 启动前读取该文件中的环境变量 `DOCKER_NETWORK_OPTIONS` ，然后设置 docker0 网桥网段；
- 如果指定了多个 `EnvironmentFile` 选项，则必须将 `/run/flannel/docker` 放在最后(确保 docker0 使用 flanneld 生成的 bip 参数)；
- docker 需要以 root 用于运行；

- docker 从 1.13 版本开始，可能将 **iptables FORWARD chain的默认策略设置为DROP**，从而导致 ping 其它 Node 上的 Pod IP 失败，遇到这种情况时，需要手动设置策略为 `ACCEPT`：

  ```
  $ sudo iptables -P FORWARD ACCEPT
  ```

  并且把以下命令写入 `/etc/rc.local` 文件中，防止节点重启**iptables FORWARD chain的默认策略又还原为DROP**

  ```
  /sbin/iptables -P FORWARD ACCEPT
  ```

分发 systemd unit 文件到所有 worker 机器：

```
cd /opt/k8s/work
source /opt/k8s/bin/environment.sh
sed -i -e "s|##DOCKER_DIR##|${DOCKER_DIR}|" docker.service
for node_ip in ${NODE_IPS[@]}
  do
    echo ">>> ${node_ip}"
    scp docker.service root@${node_ip}:/etc/systemd/system/
  done
```

配置和分发 docker 配置文件

使用国内的仓库镜像服务器以加快 pull image 的速度，同时增加下载的并发数 (需要重启 dockerd 生效)：

```
cd /opt/k8s/work
source /opt/k8s/bin/environment.sh
cat > docker-daemon.json <<EOF
{
    "registry-mirrors": ["https://docker.mirrors.ustc.edu.cn","https://hub-mirror.c.163.com"],
    "insecure-registries": ["docker02:35000"],
    "max-concurrent-downloads": 20,
    "live-restore": true,
    "max-concurrent-uploads": 10,
    "debug": true,
    "data-root": "${DOCKER_DIR}/data",
    "exec-root": "${DOCKER_DIR}/exec",
    "log-opts": {
      "max-size": "100m",
      "max-file": "5"
    }
}
EOF
```

分发 docker 配置文件到所有 worker 节点：

```
cd /opt/k8s/work
source /opt/k8s/bin/environment.sh
for node_ip in ${NODE_IPS[@]}
  do
    echo ">>> ${node_ip}"
    ssh root@${node_ip} "mkdir -p  /etc/docker/ ${DOCKER_DIR}/{data,exec}"
    scp docker-daemon.json root@${node_ip}:/etc/docker/daemon.json
  done
```

启动 docker 服务

```
source /opt/k8s/bin/environment.sh
for node_ip in ${NODE_IPS[@]}
  do
    echo ">>> ${node_ip}"
    ssh root@${node_ip} "systemctl daemon-reload && systemctl enable docker &&
systemctl restart docker"
  done
```

检查服务运行状态

```
source /opt/k8s/bin/environment.sh
for node_ip in ${NODE_IPS[@]}
  do
    echo ">>> ${node_ip}"
    ssh root@${node_ip} "systemctl status docker|grep Active"
  done
```

确保状态为 active (running)，否则查看日志，确认原因：

```
journalctl -u docker
```

检查 docker0 网桥

```
source /opt/k8s/bin/environment.sh
for node_ip in ${NODE_IPS[@]}
  do
    echo ">>> ${node_ip}"
    ssh root@${node_ip} "/usr/sbin/ip addr show flannel.1 && /usr/sbin/ip addr show
docker0"
  done
```

确认各 worker 节点的 docker0 网桥和 flannel.1 接口的 IP 处于同一个网段中(如下 172.30.80.0/32 位于 172.30.80.1/21 中):

```
>>> 172.27.137.240
3: flannel.1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc noqueue state UNKNOWN
group default
    link/ether ce:9c:a9:08:50:03 brd ff:ff:ff:ff:ff:ff
    inet 172.30.80.0/32 scope global flannel.1
       valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
group default
    link/ether 02:42:5c:c1:77:03 brd ff:ff:ff:ff:ff:ff
    inet 172.30.80.1/21 brd 172.30.87.255 scope global docker0
       valid_lft forever preferred_lft forever
```

注意: 如果您的服务安装顺序不对或者机器环境比较复杂, docker服务早于flanneld服务安装，此时 worker 节点的 docker0 网桥和 flannel.1 接口的 IP可能不会同处同一个网段下，这个时候请先停止docker服务, 手工删除docker0网卡，重新启动docker服务后即可修复:

```
systemctl stop docker
ip link delete docker0
systemctl start docker
```

查看 docker 的状态信息

```
$ ps -elfH|grep docker
4 S root     116590      1  0  80   0 - 131420 futex_ 11:22 ?       00:00:01
/opt/k8s/bin/dockerd --bip=172.30.80.1/21 --ip-masq=false --mtu=1450
4 S root     116668 116590  1  80   0 - 161643 futex_ 11:22 ?       00:00:03
containerd --config /data/k8s/docker/exec/containerd/containerd.toml --log-level
debug
```

```
$ docker info
Containers: 0
 Running: 0
 Paused: 0
 Stopped: 0
Images: 0
Server Version: 18.09.6
Storage Driver: overlay2
 Backing Filesystem: extfs
 Supports d_type: true
 Native Overlay Diff: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
 Volume: local
```

```
 Network: bridge host macvlan null overlay
 Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Init Binary: docker-init
containerd version: bb71b10fd8f58240ca47fbb579b9d1028eea7c84
runc version: 2b18fe1d885ee5083ef9f0838fee39b62d653e30
init version: fec3683
Security Options:
 apparmor
 seccomp
  Profile: default
Kernel Version: 4.14.110-0.el7.4pd.x86_64
Operating System: CentOS Linux 7 (Core)
OSType: linux
Architecture: x86_64
CPUs: 8
Total Memory: 15.64GiB
Name: zhangjun-k8s01
ID: VJYK:3T6T:EPHU:65SM:3OZD:DMNE:MT5J:O22I:TCG2:F3JR:MZ76:B3EF
Docker Root Dir: /data/k8s/docker/data
Debug Mode (client): false
Debug Mode (server): true
 File Descriptors: 22
 Goroutines: 43
 System Time: 2019-05-26T11:26:21.2494815+08:00
 EventsListeners: 0
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
 docker02:35000
 127.0.0.0/8
Registry Mirrors:
 https://docker.mirrors.ustc.edu.cn/
 https://hub-mirror.c.163.com/
Live Restore Enabled: true
Product License: Community Engine

WARNING: No swap limit support
```

更新 kubelet 配置并重启服务（每个节点上都操作）

需要删除 kubelet 的 systemd unit 文件(/etc/systemd/system/kubelet.service)，删除下面 4 行：

```
  --network-plugin=cni \\
  --cni-conf-dir=/etc/cni/net.d \\
  --container-runtime=remote \\
  --container-runtime-endpoint=unix:///var/run/containerd/containerd.sock \\
```

然后重启 kubelet 服务：

```
systemctl restart kubelet
```