tags: worker, kube-nginx

06-2. apiserver 高可用

本文档讲解使用 nginx 4 层透明代理功能实现 Kubernetes worker 节点组件高可用访问 kube-apiserver 集群的步骤。

注意：如果没有特殊指明，本文档的所有操作均在 **zhangjun-k8s-01** 节点上执行。

基于 nginx 代理的 kube-apiserver 高可用方案

- 控制节点的 kube-controller-manager、kube-scheduler 是多实例部署且连接本机的 kube-apiserver，所以只要有一个实例正常，就可以保证高可用；
- 集群内的 Pod 使用 K8S 服务域名 kubernetes 访问 kube-apiserver， kube-dns 会自动解析出多个 kube-apiserver 节点的 IP，所以也是高可用的；
- 在每个节点起一个 nginx 进程，后端对接多个 apiserver 实例，nginx 对它们做健康检查和负载均衡；
- kubelet、kube-proxy 通过本地的 nginx（监听 127.0.0.1）访问 kube-apiserver，从而实现 kube-apiserver 的高可用；

下载和编译 nginx

下载源码：

```
cd /opt/k8s/work
wget http://nginx.org/download/nginx-1.15.3.tar.gz
tar -xzvf nginx-1.15.3.tar.gz
```

配置编译参数：

```
cd /opt/k8s/work/nginx-1.15.3
mkdir nginx-prefix
yum install -y gcc make
./configure --with-stream --without-http --prefix=$(pwd)/nginx-prefix --without-
http_uwsgi_module --without-http_scgi_module --without-http_fastcgi_module
```

- `--with-stream`：开启 4 层透明转发(TCP Proxy)功能；
- `--without-xxx`：关闭所有其他功能，这样生成的动态链接二进制程序依赖最小；

输出：

```
Configuration summary
  + PCRE library is not used
  + OpenSSL library is not used
  + zlib library is not used

  nginx path prefix: "/root/tmp/nginx-1.15.3/nginx-prefix"
  nginx binary file: "/root/tmp/nginx-1.15.3/nginx-prefix/sbin/nginx"
  nginx modules path: "/root/tmp/nginx-1.15.3/nginx-prefix/modules"
  nginx configuration prefix: "/root/tmp/nginx-1.15.3/nginx-prefix/conf"
  nginx configuration file: "/root/tmp/nginx-1.15.3/nginx-prefix/conf/nginx.conf"
  nginx pid file: "/root/tmp/nginx-1.15.3/nginx-prefix/logs/nginx.pid"
  nginx error log file: "/root/tmp/nginx-1.15.3/nginx-prefix/logs/error.log"
  nginx http access log file: "/root/tmp/nginx-1.15.3/nginx-prefix/logs/access.log"
  nginx http client request body temporary files: "client_body_temp"
  nginx http proxy temporary files: "proxy_temp"
```

编译和安装：

```
cd /opt/k8s/work/nginx-1.15.3
make && make install
```

验证编译的 nginx

```
cd /opt/k8s/work/nginx-1.15.3
./nginx-prefix/sbin/nginx -v
```

输出：

```
nginx version: nginx/1.15.3
```

安装和部署 nginx

创建目录结构：

```
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 /opt/k8s/kube-nginx/{conf,logs,sbin}"
  done
```

拷贝二进制程序：

```
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 /opt/k8s/kube-nginx/{conf,logs,sbin}"
    scp /opt/k8s/work/nginx-1.15.3/nginx-prefix/sbin/nginx
root@${node_ip}:/opt/k8s/kube-nginx/sbin/kube-nginx
    ssh root@${node_ip} "chmod a+x /opt/k8s/kube-nginx/sbin/*"
  done
```

- 重命名二进制文件为 kube-nginx；

配置 nginx，开启 4 层透明转发功能：

```
cd /opt/k8s/work
cat > kube-nginx.conf << \EOF
worker_processes 1;

events {
    worker_connections  1024;
}

stream {
    upstream backend {
        hash $remote_addr consistent;
        server 172.27.138.251:6443        max_fails=3 fail_timeout=30s;
        server 172.27.137.229:6443        max_fails=3 fail_timeout=30s;
        server 172.27.138.239:6443        max_fails=3 fail_timeout=30s;
    }

    server {
        listen 127.0.0.1:8443;
        proxy_connect_timeout 1s;
        proxy_pass backend;
    }
}
EOF
```

- `upstream backend` 中的 server 列表为集群中各 kube-apiserver 的节点 IP，**需要根据实际情况修改**；

分发配置文件：

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

配置 systemd unit 文件，启动服务

配置 kube-nginx systemd unit 文件：

```
cd /opt/k8s/work
cat > kube-nginx.service <<EOF
[Unit]
Description=kube-apiserver nginx proxy
After=network.target
After=network-online.target
Wants=network-online.target

[Service]
Type=forking
ExecStartPre=/opt/k8s/kube-nginx/sbin/kube-nginx -c /opt/k8s/kube-nginx/conf/kube-
nginx.conf -p /opt/k8s/kube-nginx -t
ExecStart=/opt/k8s/kube-nginx/sbin/kube-nginx -c /opt/k8s/kube-nginx/conf/kube-
nginx.conf -p /opt/k8s/kube-nginx
ExecReload=/opt/k8s/kube-nginx/sbin/kube-nginx -c /opt/k8s/kube-nginx/conf/kube-
nginx.conf -p /opt/k8s/kube-nginx -s reload
PrivateTmp=true
Restart=always
RestartSec=5
StartLimitInterval=0
LimitNOFILE=65536

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

分发 systemd unit 文件：

```
cd /opt/k8s/work
source /opt/k8s/bin/environment.sh
for node_ip in ${NODE_IPS[@]}
  do
    echo ">>> ${node_ip}"
    scp kube-nginx.service  root@${node_ip}:/etc/systemd/system/
  done
```

启动 kube-nginx 服务：

```
cd /opt/k8s/work
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 kube-nginx && systemctl restart kube-nginx"
  done
```

检查 kube-nginx 服务运行状态

```
cd /opt/k8s/work
source /opt/k8s/bin/environment.sh
for node_ip in ${NODE_IPS[@]}
  do
    echo ">>> ${node_ip}"
    ssh root@${node_ip} "systemctl status kube-nginx |grep 'Active:'"
  done
```

确保状态为 `active (running)`，否则查看日志，确认原因：

```
journalctl -u kube-nginx
```