Calico部署文档

1. Calico部署服务的组成。

calico部署的组建包括，calico-node、calico-kube-controllers、cni插件三部分，期中calico node通过docker服务启动，calico-kube-controllers和cni插件通过yaml方式部署。

Calico-node以daemonset部署，每个节点上运行一个，主要包含了felix、bird、confd三个部分。

Felix:主要功能为生成路由、生成权限、接口管理、状态报告。Felix会监听etcd中心的存储，从他获取事件，比如用户创建一个pod后，felix负责将其网卡、ip、mac都设置好，然后在内核的路由表里面写一条，注明该ip应该用到这张网卡。如果用户设置了隔离政策，felix同样会将该策略创建到ACL中，以实现隔离。

Bird:bird是linux提供的一个bgp客户端，负责读取felix写入内核的路由信息，与其他节点的bird一起构建bgp网络。

Confd会监视etcd数据存储中对BGP配置进行更改。Confd根据对数据存储中数据的更新动态生成BIRD配置文件。当配置文件发生变化时，confd会触发BIRD的加载更新。

calico-kube-controller作用：policy controller: 监控网络策略，配置calico策略策略。监控pod标签的变化和更新calico workload endponts.监听k8s移除节点，移除calico相关的数据。

calico的cni插件，是calico和kubernetes对接的部分。Calico的cni插件会为每个容器设置一个veth pair,一端在宿主机上面。Calico的cni插件还需要在宿主机上面为每个容器设置一条路由规则，用于接收传入的ip包。有了veth pair设备之后，容器发出的ip包就会经过vp设备出现在宿主机上面。然后宿主机根据路由规则的下一跳ip地址，把他们转发给正确的网关。

1. 创建calico配置文件目录。

创建/etc/calico/目录拷贝文件。

1. 拷贝calico配置文件。

cp -a /etc/etcd/ssl/\* /etc/calico/.

1. 修改calico配置文件。

[Unit]

Description=calico node

After=docker.service

Requires=docker.service

[Service]

User=root

//calico连接k8s的etcd的集群信息根据需求进行修改。

Environment=ETCD\_ENDPOINTS=https://10.129.55.61:2379,https://10.129.55.65:2379,https://10.129.55.155:2379

PermissionsStartOnly=true

//docker启动后的calico node的docker name修改

ExecStart=/usr/bin/docker run --net=host --privileged --name=calico-node \

-e ETCD\_ENDPOINTS=${ETCD\_ENDPOINTS} \

-e ETCD\_CA\_CERT\_FILE=/etc/etcd/ssl/etcd-root-ca.pem \

-e ETCD\_CERT\_FILE=/etc/etcd/ssl/etcd.pem \

-e ETCD\_KEY\_FILE=/etc/etcd/ssl/etcd-key.pem \

-e NODENAME=master1 \

-e IP= \

//指定网卡名称。

-e IP\_AUTODETECTION\_METHOD=can-reach=10.129.55.61 \

-e AS=64512 \

-e CLUSTER\_TYPE=k8s,bgp \

-e CALICO\_IPV4POOL\_CIDR=172.17.0.0/16 \

-e CALICO\_IPV4POOL\_IPIP=always \

-e CALICO\_LIBNETWORK\_ENABLED=true \

-e CALICO\_NETWORKING\_BACKEND=bird \

-e CALICO\_DISABLE\_FILE\_LOGGING=true \

-e FELIX\_IPV6SUPPORT=false \

-e FELIX\_DEFAULTENDPOINTTOHOSTACTION=ACCEPT \

-e FELIX\_LOGSEVERITYSCREEN=info \

-e FELIX\_IPINIPMTU=1440 \

-e FELIX\_HEALTHENABLED=true \

-e CALICO\_K8S\_NODE\_REF=master1 \

-v /etc/calico/etcd-root-ca.pem:/etc/etcd/ssl/etcd-root-ca.pem \

-v /etc/calico/etcd.pem:/etc/etcd/ssl/etcd.pem \

-v /etc/calico/etcd-key.pem:/etc/etcd/ssl/etcd-key.pem \

-v /lib/modules:/lib/modules \

-v /var/lib/calico:/var/lib/calico \

-v /var/run/calico:/var/run/calico \

quay.io/calico/node:v3.1.0

ExecStop=/usr/bin/docker rm -f calico-node

Restart=always

RestartSec=10

[Install]

WantedBy=multi-user.target

1. 启动和检查calico-node服务。

systemctl start calico-node.service

systemctl status calico-node.service

1. 部署calico相关的权限文件：

for i in `kubectl get clusterrole|egrep system|awk -F'[ ]+' '{print $1}'`;do kubectl get clusterrole $i -o yaml > $i.yaml;done

Kubectl apply -f ./

1. 部署calico-kube-controllers和calico-cni插件部分。

calico的pod服务主要包含cni插件和kube-controllers服务。cni插件可以支持不同的容器runtime，对容器和网络插件起隔离作用，cni插件主要作用在容器到容器间的网络通信。由于前面已经启动calico-node，新部署的calico-node的pod中的实际内容为calico-cni.

ETCD\_CERT=`cat /etc/etcd/ssl/server.pem | base64 | tr -d '\n'`

ETCD\_KEY=`cat /etc/etcd/ssl/server-key.pem | base64 | tr -d '\n'`

ETCD\_CA=`cat /etc/etcd/ssl/ca.pem | base64 | tr -d '\n'`

ETCD\_ENDPOINTS="https://10.129.55.30:2379,https://10.129.55.61:2379,https://10.129.55.65:2379"

\cp calico.example.yaml calico.yaml

sed -i "s@.\*etcd\_endpoints:.\*@\ \ etcd\_endpoints:\ \"${ETCD\_ENDPOINTS}\"@gi" calico.yaml

sed -i "s@.\*etcd-cert:.\*@\ \ etcd-cert:\ ${ETCD\_CERT}@gi" calico.yaml

sed -i "s@.\*etcd-key:.\*@\ \ etcd-key:\ ${ETCD\_KEY}@gi" calico.yaml

sed -i "s@.\*etcd-ca:.\*@\ \ etcd-ca:\ ${ETCD\_CA}@gi" calico.yaml

sed -i 's@.\*etcd\_ca:.\*@\ \ etcd\_ca:\ "/calico-secrets/etcd-ca"@gi' calico.yaml

sed -i 's@.\*etcd\_cert:.\*@\ \ etcd\_cert:\ "/calico-secrets/etcd-cert"@gi' calico.yaml

sed -i 's@.\*etcd\_key:.\*@\ \ etcd\_key:\ "/calico-secrets/etcd-key"@gi' calico.yaml

# 注释掉 calico-node 部分(由 systemd 接管)

sed -i '123,219s@.\*@#&@gi' calico.yaml

部署calico服务。

kubectl create -f calico.yaml

1. 部署完成检查服务状态。确保calico-node和calico-kube-controllers服务状态运行正常.

[root@fp-web-11 ~]# kubectl get pod -n kube-system|egrep calico

calico-kube-controllers-98989846-whzmf 1/1 Running 0 41d

calico-node-lzs79 2/2 Running 0 41d

calico-node-n6ptq 2/2 Running 0 41d

calico-node-tklf6 2/2 Running 0 41d

查看日志检查状态：

kubectl logs -f calico-kube-controllers-98989846-whzmf -n kube-system