Juniper+zebra的BGP方案

撰稿人:秋明  
邮箱:tyumen@zhoufengjie.cn  
url:http://www.zhoufengjie.cn/?p=407  
日期:2019 年 04 月 30 日星期二  
口号:开源、分享、共进 注解:任何想通过本平台分享文档的，可以随时联系，文档里面注明出稿人，方 便文档使用者认识撰稿的兄弟;

目录

[1 前言 3](#_Toc7507814)

[1.1 软件及需求描述 3](#_Toc7507815)

[1.2 环境说明 3](#_Toc7507816)

[2 交换机配置 4](#_Toc7507817)

[2.1 配置BGP 4](#_Toc7507818)

[2.2 配置限制策略 4](#_Toc7507819)

[2.2.1 配置发布的路由策略 5](#_Toc7507820)

[2.2.2 配置接收的路由策略 6](#_Toc7507821)

[3 服务器配置 7](#_Toc7507822)

[3.1 准备工作 7](#_Toc7507823)

[3.1.1 安装软件包 7](#_Toc7507824)

[3.1.2 调整zebra权限 7](#_Toc7507825)

[3.1.3 启动zebra 7](#_Toc7507826)

[3.1.4 开启日志 8](#_Toc7507827)

[3.1.5 调整内核 8](#_Toc7507828)

[3.1.6 网卡注意细节[附注] 8](#_Toc7507829)

[3.2 服务器配置 8](#_Toc7507830)

[3.2.1 配置zebra 8](#_Toc7507831)

[3.2.2 配置bgp 9](#_Toc7507832)

[3.2.2.1 启动bgp 9](#_Toc7507833)

[3.2.2.2 配置bgp 9](#_Toc7507834)

[3.2.2.3 查看bgp状态 10](#_Toc7507835)

[3.2.2.4 配置宣告网络【使用主要关注】 10](#_Toc7507836)

# 前言

# 软件及需求描述

Quagga是一个开源路由软件套件。在这个教程中，我将会重点讲讲如何把一个Linux系统变成一个BGP路由器，还是使用Quagga，演示如何建立BGP与其它BGP路由器对等。

在我们进入细节之前，一些BGP的背景知识还是必要的。边界网关协议（即BGP）是互联网的域间路由协议的实际标准。在BGP术语中，全球互联网是由成千上万相关联的自治系统(AS)组成，其中每一个AS代表每一个特定运营商提供的一个网络管理域。

为了使其网络在全球范围内路由可达，每一个AS需要知道如何在因特网中到达其它的AS。这时候就需要BGP出来扮演这个角色了。BGP是一个AS去与相邻的AS交换路由信息的语言。这些路由信息通常被称为BGP线路或者BGP前缀。包括AS号(ASN；全球唯一号码)以及相关的IP地址块。一旦所有的BGP线路被当地的BGP路由表学习和记录，每一个AS将会知道如何到达互联网的任何公网IP。

在不同域(AS)之间路由的能力是BGP被称为外部网关协议(EGP)或者域间协议的主要原因。就如一些路由协议，例如OSPF、IS-IS、RIP和EIGRP都是内部网关协议(IGPs)或者域内路由协议，用于处理一个域内的路由.

# 环境说明

核心交换机：

1. 接服务器点对点Ip地址：172.17.3.1【所有的流量都经过核心交换机】。
2. AS号：100
3. 路由ID：172.17.3.1

核心交换机接线：

1. 接出口网络。
2. 接测试环境服务器

测试环境服务器：

1. 接核心交换机点对点ip地址：172.17.3.3
2. 接外网ip地址：192.168.1.3
3. AS号：103
4. 路由ID：172.17.3.3

# 交换机配置

# 配置BGP

set routing-options autonomous-system 65500

set routing-options router-id 172.17.3.1

set protocols bgp group ebgp local-as 100

set protocols bgp group ebgp type external

set protocols bgp group ebgp neighbor 172.17.3.3 multihop

set protocols bgp group ebgp neighbor 172.17.3.3 peer-as 103

set protocols bgp group ebgp neighbor 172.17.3.3 local-address 172.17.3.1

查看邻居信息以及发布过来或者出去的路由：

run show bgp neighbor

run show route receive-protocol bgp 172.17.3.3

run show route advertising-protocol bgp 172.17.3.3

# 配置限制策略

对于测试环境此为非必须的策略，但是生产环境一定要配置做控制。

编写需要需要引入的路由(import)策略限制，以及需要发布/不发布的路由(export)策略限制。做限制的目的主要是为了把内部的地址段不发布出去，以及外部的一些路由不想收的时候不收。默认配置上上面的策略就已经可以收Router的路由表了。

# 配置发布的路由策略

这条在本测试中没有任何价值，放在这只为了演示，对外发布的路由也仅仅是核心上的直连地址，下部的地址并未进行宣告【可以把静态路由、ospf路由等发布给bgp】。

配置拒绝发布的路由：

set policy-options policy-statement export\_policy term deny\_net from route-filter 192.168.0.0/16 orlonger

set policy-options policy-statement export\_policy term deny\_net from route-filter 172.16.0/12 orlonger

set policy-options policy-statement export\_policy term deny\_net from route-filter 10.10.10.0/24 orlonger

set policy-options policy-statement export\_policy term deny\_net then reject

配置发布的路由范围：

set policy-options policy-statement export\_policy term all\_local from protocol direct

set policy-options policy-statement export\_policy term all\_local from route-filter 10.0.0.0/8 upto /32

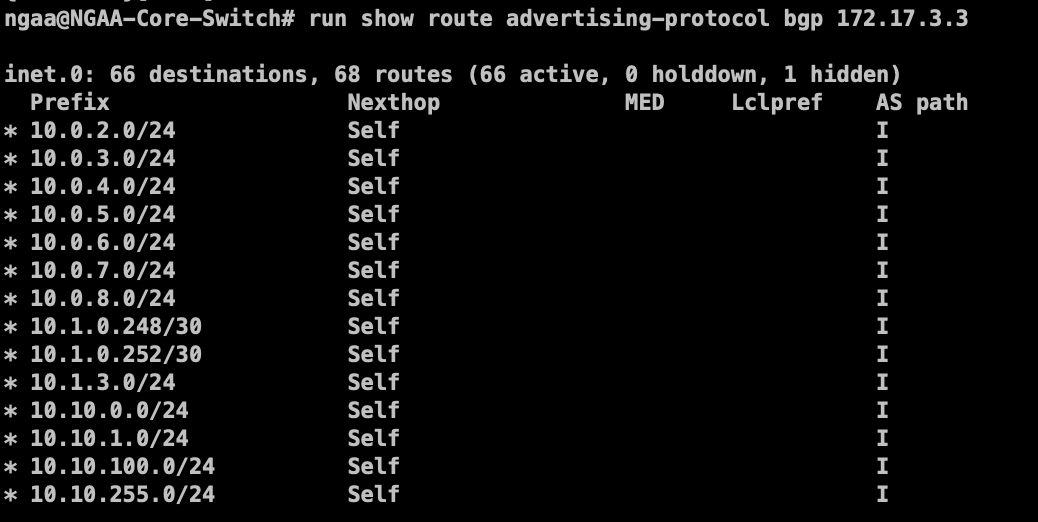
set policy-options policy-statement export\_policy term all\_local then accept

set policy-options policy-statement export\_policy then reject

应用策略：

set protocols bgp group ebgp neighbor 172.17.3.3 export export\_policy

查看发布的路由策略：show route advertising-protocol bgp 172.17.3.3



# 配置接收的路由策略

配置拒绝接受的路由：

set policy-options policy-statement import\_policy term deny\_net from route-filter 10.0.0.0/8 upto /32

set policy-options policy-statement import\_policy term deny\_net from route-filter 172.16.0.0/12 upto /32

set policy-options policy-statement import\_policy term deny\_net from route-filter 192.168.0.0/16 upto /32

set policy-options policy-statement import\_policy term deny\_net then reject

配置允许接收的路由:

set policy-options policy-statement import\_policy term allow\_net then accept

应用策略：

set protocols bgp group ebgp neighbor 172.17.3.3 import import\_policy

查看接收的路由策略：show route receive-protocol bgp 172.17.3.3



# 服务器配置

服务器的软件安装、内核调整以及注意事项

# 准备工作

服务器的软件安装调整

# 安装软件包

yum install quagga

配置一下vtysh管理终端的主机名显示为Router：

echo "hostname Router" > /etc/quagga/vtysh.conf

# 调整zebra权限

如果在设备CentOS-7上面开着selinx，那么执行给zebra加权限：

setsebool -P zebra\_write\_config 1

# 启动zebra

cp /usr/share/doc/quagga-**XXXXX**/zebra.conf.sample /etc/quagga/zebra.conf

systemctl start zebra

systemctl enable zebra

# 开启日志

vtysh

Router# configure terminal

Router (config)# log file /var/log/quagga/quagga.log

Router (config)# exit

Router # write

# 调整内核

编辑：/etc/sysctl.conf

net.ipv4.ip\_forward=1

net.ipv4.conf.all.rp\_filter=0

net.ipv4.conf.default.rp\_filter=0

然后执行：sysctl -p

# 网卡注意细节[附注]

如果使用单网卡trunk做单臂路由，那么需要注意几个细节：

1. vlan：在网卡配置需要加上VLAN=yes参数。
2. 网卡模式：需要删掉网卡配置里面的TYPE=Ethernet

# 服务器配置

# 配置zebra

vtysh

上述命令进入zebra的配置中

如果想查看系统物理网卡信息等情况的话，可以执行：show interface命令。

Router # configure terminal

Router (config)# interface eth0.13

Router (config-if)# ip address 192.168.1.3/24

Router (config-if)# description "to Wan"

Router (config-if)# no shutdown

Router (config-if)# exit

Router (config)# interface eth0.2003

Router (config-if)# ip address 172.17.3.3/29

Router (config-if)# description "to Lan"

Router (config-if)# no shutdown

Router (config-if)# exit

Router (config)#exit

Router # write

# 配置bgp

# 启动bgp

cp /usr/share/doc/quagga-XXXXXXX/bgpd.conf.sample /etc/quagga/bgpd.conf

systemctl start bgpd

systemctl enable bgpd

# 配置bgp

vtysh

Router # configure terminal

Router (config)# no router bgp 7675

Router (config)# router bgp 103

Router (config)# no auto-summary

Router (config)# no synchronizaiton

Router (config-router)# neighbor 172.17.3.1 remote-as 100

Router (config-router)# neighbor 172.17.3.1 description "Switch Lan provider"

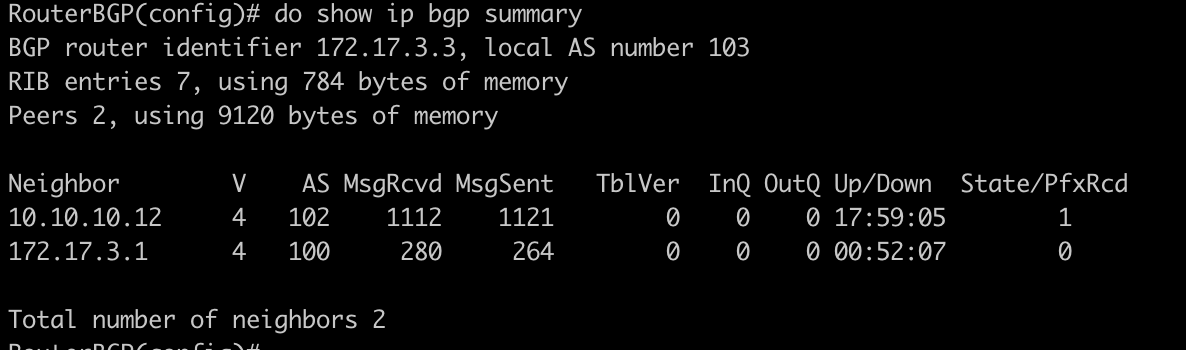
Router (config-router)# exit

Router (config)# exit

Router # write

# 查看bgp状态

Router-A# show ip bgp summary



# 配置宣告网络【使用主要关注】

vtysh

Router # configure terminal

Router (config)# router bgp 103

Router (config-router)#network 47.95.14.251/32

Router (config-router)# exit

Router (config)# exit

Router # write

查看bgp路由状态：

Router # show ip bgp

