昵称： SammyLiu
园龄： 6年4个月
荣誉： 推荐博客
粉丝： 1134
关注： 31
+加关注

< 2021年4月 >

日	一	二	三	四	五	六
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

Open vSwitch(1)
GRE(1)
Neutron(1)
OpenStack(1)

世民谈云计算（微信公众号ID：SammyTalksAboutCloud）

（声明：本站文章仅代表作者个人观点，与作者所在公司无关。若对我的文章感兴趣，请关注我的微信公众号【ID：SammyTalksAboutCloud】，接收我的更新通知。）

博客园 首页 新随笔 订阅 管理

随笔 - 187 文章 - 49 评论 - 727 阅读 - 213万

目录导航

Netruon 理解（11）：使用 NAT 将 Linux network namespace 连接外网

学习 Neutron 系列文章：

(1) Neutron 所实现的虚拟化网络

(2) Neutron OpenvSwitch + VLAN 虚拟网络

(3) Neutron OpenvSwitch + GRE/VxLAN 虚拟网络

(4) Neutron OVS OpenFlow 流表 和 L2 Population

(5) Neutron DHCP Agent

(6) Neutron L3 Agent

(7) Neutron LBaas

(8) Neutron Security Group

(9) Neutron FWaas 和 Nova Security Group

(10) Neutron VPNaas

(11) Neutron DVR

（12）Neutron VRRP

（13）High Availability （HA）

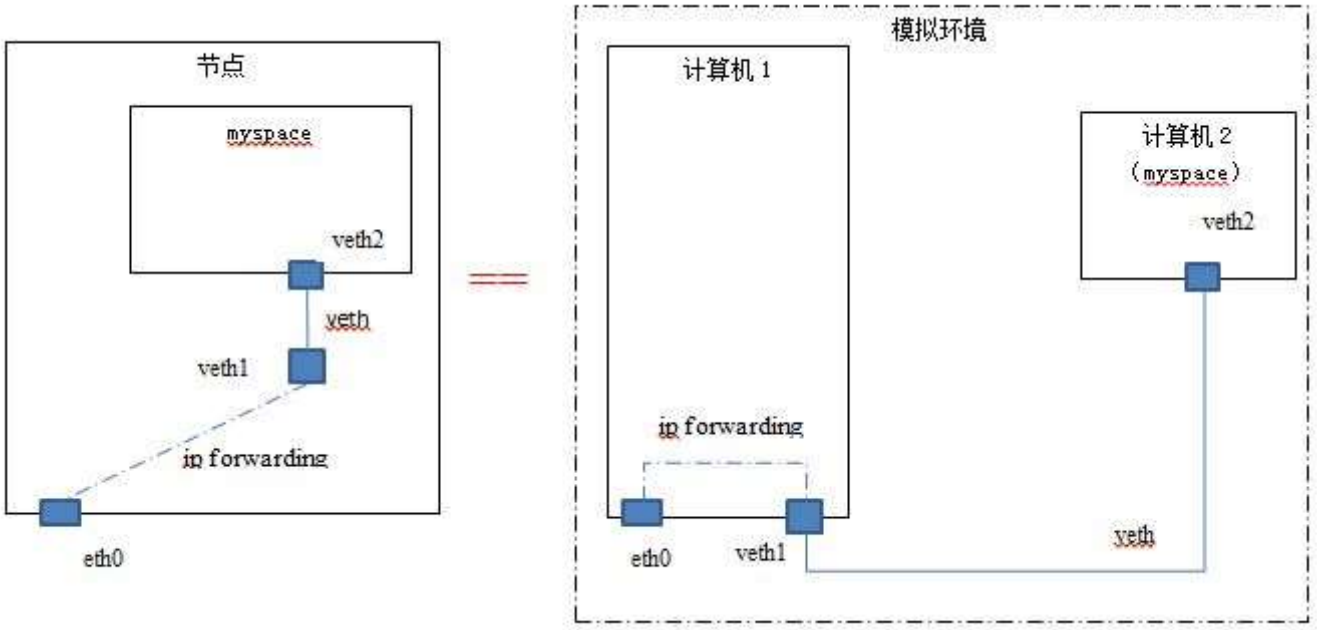
（14）使用 NAT 将 Linux network namespace 连接外网

（15）使用 Linux bridge 将 Linux network namespace 连接外网

Linux network namespace 连接外网从大类上来讲主要有两种方法：网络地址转换（NAT） 和 桥接（bridging），而桥接根据使用的网桥又可以分为使用 linux bridge 和 Open vSwitch 网络等。本文将说明 NAT 具体配置过程以及原理。

1. 环境及配置

我们可以把一个 linux network namespace 看作另一个计算机，这样看起来会更加直观：



节点 host1 的 IP 地址为 192.168.1.32. 实验使用的另一个机器 host2 的 IP 为 192.168.1.15.

为了能从在 host1 上的 netns myspace 上能 ping 通 host2，你需要做的配置及说明：

步骤#	命令	说明

2019年6月(2)
2019年3月(2)
2019年2月(1)
2019年1月(3)
2018年12月(7)
2018年11月(2)
2018年10月(2)
更多

文章分类 (24)

Ceph(1)
Web 服务器(2)
操作系统(1)
大数据(2)
服务器(1)
日常操作(2)
网络(11)
虚拟化(3)
云(1)

文章档案 (46)

2019年8月(1)
2018年8月(1)
2018年6月(2)
2016年10月(2)
2016年9月(1)
2016年6月(1)
2016年5月(3)
2015年12月(4)
2015年10月(5)

1	ip netns add myspace	创建名称为 'myspace' 的 linux network namespace
2	ip link add veth1 type veth peer name veth2	创建一个 veth 设备，一头为 veth1，另一头为 veth2
3	ip link set veth2 netns myspace	将 veth2 加入 myspace 作为其一个 network interface
4	ifconfig veth1 192.168.45.2 netmask 255.255.255.0 up	配置 veth1 的 IP 地址
5	ip netns exec myspace ifconfig veth2 192.168.45.3 netmask 255.255.255.0 up	配置 veth2 的 IP 地址，它和 veth1 需要在同一个网段上
6	ip netns exec myspace route add default gw 192.168.45.2	将 myspace 的默认路由设为 veth1 的 IP 地址
7	echo 1 > /proc/sys/net/ipv4/ip_forward	开启 linux kernel ip forwarding
8	iptables -t nat -A POSTROUTING -s 192.168.45.0/24 -o eth0 -j MASQUERADE	配置 SNAT，将从 myspace 发出的网络包的 soruce IP address 替换为 eth0 的 IP 地址
9	iptables -t filter -A FORWARD -i eth0 -o veth1 -j ACCEPT iptables -t filter -A FORWARD -o eth0 -i veth1 -j ACCEPT	在默认 FORWARD 规则为 DROP 时显式地允许 veth1 和 eth0 之间的 forwarding

这些配置之后，host 上的 route 表中自动添加了一条路由规则：

目录导航

2015年9月(2)
2015年6月(1)
2015年4月(23)

最新评论

1. Re:【译文连载】 理解Istio服务网格（第一章 概述）

翻译的很棒！！！ 加油

--MrSonKo

2. Re:从大公司到创业公司，技术人转型怎样转变思路与处事之道?

程序员真是一个缺乏安全感的职业

--等不到的口琴

3. Re:年终盘点 | 2020年，国内私有云正式进入3.0时代

不管到了几点零，争取云计算，而不是云存储，
硬盘CPU内存有钱就能买，软件(操作系统，数据库，中间件平台,企业应用）才是未来云计算核心

--信息化建设

4. Re:理解Docker（2）：Docker 镜像

这篇文章很老么？ ubuntu:14.04 都有了，很多年的东西，现在都用 ubuntu 20.04 了吧？ 开篇没写好. "所有 Linux 发行版都采用相同的 Linux 内核（kernel），然...

--Jacklondon Chen

5. Re:Istio服务网格原理与实践

@SammyLiu 好的 谢谢博主 这片文章我有看到，实际实践起来 效果并没有太理想，没有把demo的案例给

```
root@compute2:/home/s1# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.45.0    0.0.0.0         255.255.255.0   U     0      0        0 veth1
```

myspace 的路由表：

```
root@compute2:/home/s1# ip netns exec myspace route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.45.2    0.0.0.0         UG    0      0        0 veth2
192.168.45.0    0.0.0.0         255.255.255.0   U     0      0        0 veth2
```

其中第一条是显式地被创建的，第二条是自动被创建的。

现在你就可以从 myspace 中 ping 外网的地址了。

2 原理

2.1 关于第八条 SNAT

如果没有设置第八条 SNAT，那么 ICMP Request 能够到达对方计算机，但是 echo reply 消息回不来，因为其目的地址为一个内部地址。

```
root@compute1:/home/s1# tcpdump -eni bridge1 -p icmp -v
tcpdump: listening on bridge1, link-type EN10MB (Ethernet), capture size 65535 bytes
07:40:03.827852 08:00:27:4f:56:17 > 08:00:27:c7:cf:ca, ethertype IPv4 (0x0800), length 98: (tos 0x0
    192.168.45.3 > 192.168.1.15: ICMP echo request, id 26569, seq 1, length 64
07:40:04.829779 08:00:27:4f:56:17 > 08:00:27:c7:cf:ca, ethertype IPv4 (0x0800), length 98: (tos 0x0
```

加上第八条之后，ping 能成功，也就是 ICMP echo request 能发出，echo reply 能返回。

在 host2 的网卡 eth0 上，能看到 ICMP echo request 网络包的源 IP 为 host1 的 IP：

```
07:44:19.360519 08:00:27:c7:cf:ca > 08:00:27:4f:56:17, ethertype IPv4 (0x0800), length 98: (tos 0x0
    192.168.1.15 > 192.168.1.32: ICMP echo reply, id 28534, seq 7, length 64
```

目录导航

代理出去，可能我哪里有点不对，谢谢博主哈...

--紫色飞猪

阅读排行榜

1. Neutron 理解 (1): Neutron 所实现的网络虚拟化 [How Neutron Virtualizes Network](60783)
2. KVM 介绍（1）：简介及安装(60729)
3. 理解Docker（8）：Docker 存储之卷（Volume）(59907)
4. 探索 OpenStack 之（9）：深入块存储服务Cinder （功能篇）(50876)
5. KVM 介绍（2）：CPU 和内存虚拟化(49153)

评论排行榜

1. Neutron 理解 (1): Neutron 所实现的网络虚拟化 [How Neutron Virtualizes Network](68)
2. Neutron 理解（14）：Neutron ML2 + Linux bridge + VxLAN 组网(57)
3. Neutron 理解 (8): Neutron 是如何实现虚机防火墙的 [How Neutron Implements Security Group](34)
4. Neutron 理解（5）：Neutron 是如何向 Nova 虚机分配固定IP地址的（How Neutron Allocates Fixed IPs to Nova Instance）(26)
5. Neutron 理解 (3): Open vSwitch + GRE/VxLAN 组网 [Netruon Open vSwitch + GRE/VxLAN Virutal Network](25)

推荐排行榜

```
07:44:20.358360 08:00:27:4f:56:17 > 08:00:27:c7:cf:ca, ethertype IPv4 (0x0800), length 98: (tos 0x0
    192.168.1.32 > 192.168.1.15: ICMP echo request, id 28534, seq 8, length 64
```

在 host1 的网卡 eth0 上，能看到来回网络包使用的是 host1 和 host2 的 IP 地址：

```
root@compute2:/home/s1# tcpdump -envi eth0 -p icmp -v
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
06:31:27.285150 08:00:27:4f:56:17 > 08:00:27:c7:cf:ca, ethertype IPv4 (0x0800), length 98: (tos 0x0
    192.168.1.32 > 192.168.1.15: ICMP echo request, id 29610, seq 158, length 64
06:31:27.285777 08:00:27:c7:cf:ca > 08:00:27:4f:56:17, ethertype IPv4 (0x0800), length 98: (tos 0x0
    192.168.1.15 > 192.168.1.32: ICMP echo reply, id 29610, seq 158, length 64
```

在 host1 的 veth1 上，能看到发出的网络包的源 IP 和收到的网络包的目的 IP 皆为内部网段的 IP 地址：

```
root@compute2:/home/s1# tcpdump -envi veth1 -p icmp -v
tcpdump: listening on veth1, link-type EN10MB (Ethernet), capture size 65535 bytes
06:33:13.355956 b2:52:7e:b6:e9:4e > ee:53:ae:dd:6f:7f, ethertype IPv4 (0x0800), length 98: (tos 0x0
    192.168.45.3 > 192.168.1.15: ICMP echo request, id 29610, seq 264, length 64
06:33:13.356391 ee:53:ae:dd:6f:7f > b2:52:7e:b6:e9:4e, ethertype IPv4 (0x0800), length 98: (tos 0x0
    192.168.1.15 > 192.168.45.3: ICMP echo reply, id 29610, seq 264, length 64
```

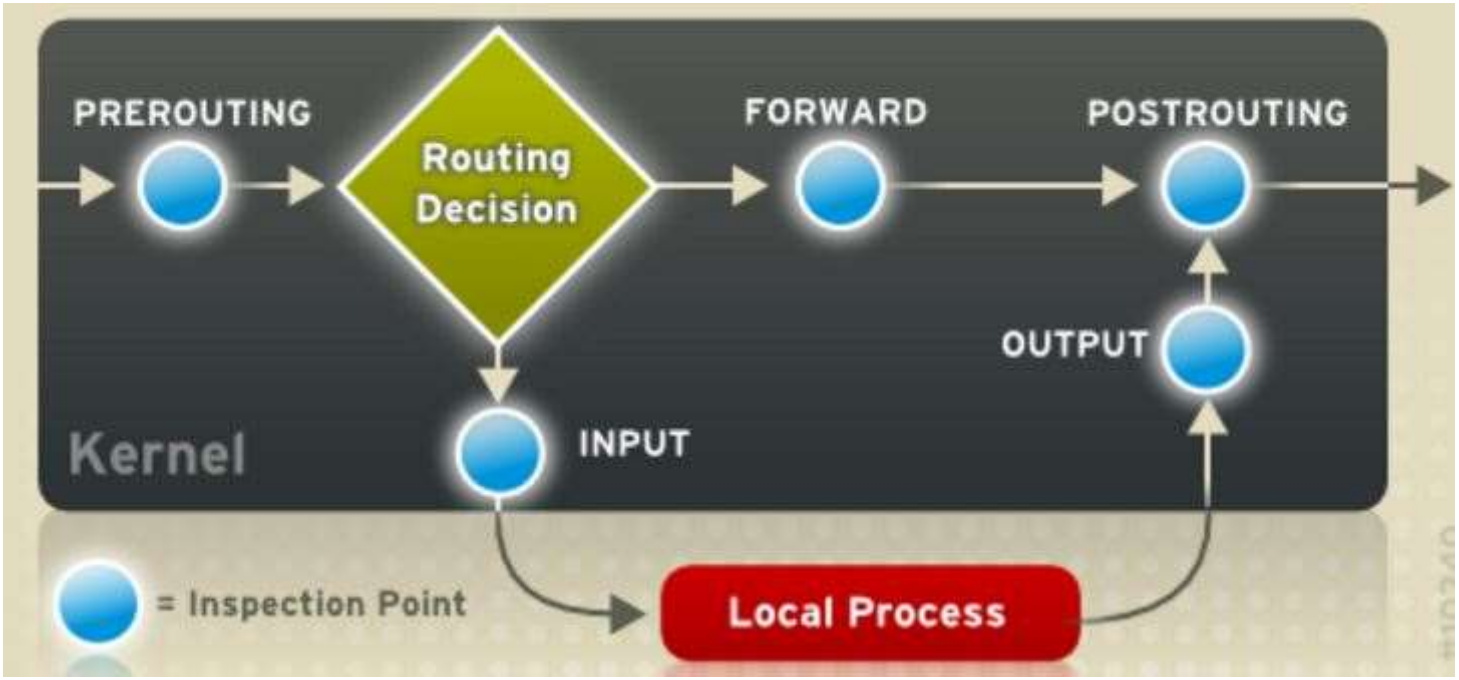
那为什么在 forwarding 发生之前，在 iptables nat 表中并没有显式做 DNAT 的情况下 veth1 和 eth0 之间有了 DNAT 呢?

原因是 ICMP 使用了 Query ID，而 NAT 会自动根据 ICMP Query ID 对 ICMP echo reply 做 DNAT。根据 https://tools.ietf.org/html/rfc5508 的 3.1. ICMP Query Mapping 章节，当内部的 host1 发一个 ICMP Query 给外部 host2 时，linux 内核的 NAT 模块会针对 NAT 的外部地址分配一个匹配的 query ID（上面例子中的 id 29610）；然后当收到 ICMP echo reply 时，NAT模块会根据 ICMP Query ID 以及 ICMP header checksum 将外部 IP 转化为内部 IP，然后再做 forwarding。也可以看出，ICMP Query ID 类似于 TCP 和 UDP 使用的端口号（port number），两者的区别在于 NAT 为 ICMP 自动做了 DNAT，而 TCP 和 UDP 则需要显式添加 DNAT 规则。

目录导航

1. 与朋友谈心，也是与自己谈心(18)
2. 理解Docker（8）：Docker 存储之卷（Volume）(13)
3. 理解Docker（2）：Docker 镜像(12)
4. Neutron 理解 (1): Neutron 所实现的网络虚拟化 [How Neutron Virtualizes Network](11)
5. KVM 介绍（2）：CPU 和内存虚拟化(10)

2.2 关于 IP forwarding



（图片来源）

Linux 内核在从 veth1 上收到 myspace 发过来的 ICMP 包以后，

1. 执行 PREROTING 规则，本例不需要此配置此规则。
2. 执行 Routing decision。它会检查网络包的目的IP地址，发现它不在本机上，说明需要进行 routing （FORWARD）。因为 Linux 上默认的 IP forwarding 是关闭的，因此需要执行第七条命令来开启它；然后再检查 iptable 规则中的这种 forwarding （ICMP 包从veth1 出再进入 eth0）。通常为了安全起见，管理员会将 FORWARD 的默认规则设置为 DROP，此时则需要执行第九条命令显式地允许（ACCEPT）所需要的 forwarding。
3. 执行 ip forwarding。查找 host1 上的路由表，网络包会被路由到 eth0。
4. 执行 POSTROUTING 规则。因为此时的网络包的源 IP 地址仍然为内部地址，为了避免 ICMP 网络包有去无回，需要通过 SNAT 将内部地址转换为外部地址。这就是第八条的作用。
5. 从 eth0 发出

（3）关于 myspace 的默认路由

因为 myspace 只有一根网线（veth）连接到 veth1，因此，必须将默认的路由器地址设置为 veth1 的 IP 地址。

2.3 DNAT

上面的配置只是为了能从 myspace 中访问外网。要使得外面网络能访问 myspace 中的应用的话，则需要在 host1 上添加 DNAT 规则，比如将 8080 端口受到的 TCP 转到内部 IP 上的 80 端口；同时还需要配置 forward 规则，允许从 eth0 出到 veth1 进。基本过程为：

对方计算机使用 host1 的 IP 地址和特定端口访问 mysapce 中的 TCP 应用 （192.168.1.32：8080），

1. Linux 内核在从 eth0 上收到发过来的TCP包（IP 为 192.168.1.32，端口为 8080）
2. 执行 PREROTING 规则，将目的 IP 及端口修改为 192.168.45.3 和 80
3. 执行 Routing decision。它会检查网络包的目的IP地址，发现它不在本机上，说明需要进行 routing （FORWARD）。检查 Linux 的 IP forwarding 是否打开；然后再检查 iptable 规则中的这种 forwarding （TCP 包从 eth0 出再进入 veth1）。通常为了安全起见，管理员会将 FORWARD 的默认规则设置为 DROP，此时则需要执行类似第九条命令显式地允许（ACCEPT）所需要的 forwarding。
4. 执行 ip forwarding。查找 host1 上的路由表，网络包会被路由到 veth1。
5. 执行 POSTROUTING 规则。尽管有第八条规则，但是它要求源地址在内部网段，因此不会执行。
6. 从 veth1 发出的包通过 veth 设备进入 myspace 的 veth0 网卡。
7. 被 80 端口上的应用接收到。

分类: Neutron, 网络, 原理

SammyLiu
关注 - 31
粉丝 - 1134

3 0

推荐博客

+加关注

« 上一篇： 在 KVM 上安装 Win7 虚拟机
» 下一篇： Netruon 理解（12）：使用 Linux bridge 将 Linux network namespace 连接外网

posted on 2016-08-11 11:14 SammyLiu 阅读(5564) 评论(2) 编辑 收藏