IPv4 only环境下通过Wireguard获取全局IPv6地址

http://www.jeepxie.net/article/60803.html

None Thu Oct, 22 00:10

wireguard作为一个微皮恩的后起之秀，具备配置简单，上手快速的特点，而且由于其太过于先进，对IPv6的支持非常好，可以给每一个加入进来的节点分配一个本地组播IPv6+内网IPv4地址。而且其还有一个非常厉害的优点，就是可以轻松假设一个非星形的微皮恩网络，这在Open微皮恩上比较难以实现的。

由于wireguard这个协议过于先进，所以linux内核版本如果低了还装不上，这里使用Ubuntu 18.04为例介绍安装和配置过程。本文所实现的效果就是你的一台在IPv4 Only环境下的服务器或PC，能够同时访问IPv4网络和IPv6网络上的站点，并且具备一个全局单播IPv6地址，这样其他人如果有IPv6网络就可以直接通过IPv6地址来访问你这台IPv4环境中的服务器，化普通服务器为双栈服务器。但是有一个前提要求，就是你需要一台IPv6+IPv4的双栈服务器来作为wireguard的Server端可以，这台Server端仅仅是做网络转发用，所以配置不需要太高，带宽足够大就行。可惜目前国内较火的云主机提供商腾讯云，阿里云，华为云等均没有双栈服务器可以选择，所以此处我选择一家美国的云主机服务商提供的双栈服务器，1核512M内存来做示范。

首先要在我们的服务端双栈服务器上安装wireguard，操作系统为Ubuntu 18.04，其他版本可能会碰到内核版本太低的问题

```
sudo add-apt-repository ppa:wireguard/wireguard
dudo apt-get update
dudo apt-get install wireguard
```

这样wireguard就安装完成了

然后我们编写一个配置文件，由于wireguard协议没有server端和客户端之分，大家都是平等的节点，只是在网络路由上有区别，互相之间通过一对密钥来认证，所以我们首先来生成服务端的一对密钥和客户端的一对密钥

生成密钥使用wg genkey命令可以生成一个私钥，然后wireguard有一个方法能够根据私钥来算出公钥，密钥并没有server和client之分（这点不同于Open微皮恩的SSL密钥），于是我们这样来生成

```
# wg genkey
YIFnXXCCt1H1KD/UcRZ7dr91dXjdKSFnUEZVBmkHdFs=


# echo 'YIFnXXCCt1H1KD/UcRZ7dr91dXjdKSFnUEZVBmkHdFs=' | wg pubkey
T0hbccLqGqfapjoQ5ijgxQDRhQ4KPuqeYQIkKxWRiHM=
```

如上所示，先生成一个私钥，然后通过管道传给 wg pubkey方法就生成了一个公钥，于时我们就有了一个密钥对

私钥：YIFnXXCCt1H1KD/UcRZ7dr91dXjdKSFnUEZVBmkHdFs=

公钥：T0hbccLqGqfapjoQ5ijgxQDRhQ4KPuqeYQIkKxWRiHM=

顾名思义，私钥就是每个节点自己知道，别人不知道，用来解密流量的。公钥就是自己可以不知道但是和你相连的节点必须知道，用来认证你并且给你发流量的。

由于我们有两个节点，所以我们用上面的方法再生成一对密钥

私钥2：YDPB08gE6pxmSsUOEYmPcl6AovjJdWw8cvkkXUErclg=

公钥2：bgbcIhmmoxsC4PPo6627A6Md+/OyHF224Rd8zQjzd1U=

然后我们开始编写服务端的配置文件，此处我们把先生成的那对密钥交给服务端

vi /etc/wireguard/wg0.conf

```
[Interface]
ListenPort = 36889
PrivateKey = YIFnXXCCt1H1KD/UcRZ7dr91dXjdKSFnUEZVBmkHdFs=
Address = 10.0.20.1/24, fd86::1/64
PostUp = ip6tables -t nat -A POSTROUTING -o ens3 -j MASQUERADE; iptables
-t nat -A POSTROUTING -o ens3 -j MASQUERADE
PostDown = ip6tables -t nat -D POSTROUTING 1; iptables -t nat -D
POSTROUTING 1


[Peer]
Endpoint = 10.0.20.2:36889
PublicKey = bgbcIhmmoxsC4PPo6627A6Md+/OyHF224Rd8zQjzd1U=
AllowedIPs = 10.0.20.2/32, fd86::2/64
```

由于wireguard是类似于点对点的微皮恩，所以每个节点上的配置文件要写上自己的私钥和别的节点的公钥，再[interface]节点上写自己的私钥，该节点只有一个，代表自己。再[peer]节点上写上其他节点的公钥，[peer]节点可以写一个或多个。Address就是自身配置的地址，两个节点之间要统一起来，尽量使用内网地址，如192.168.0.0/16，10.0.0.0/8，172.17.0.0/16这样的地址，IPv6也不要用2000::以上的地址，以免和公网上的冲突，我这里使用的是fd86::。

AloowedIPs不是说允许哪些IP连接进入，而是说哪些目的地址的IP要路由到该节点去，此处我们就把发送到10.0.20.2的数据包发送到我们的客户端服务器去

PostUP和PostDown是wireguard再启动前和启动后执行的脚本，可以直接写脚本，用;分割，也可以用sh '脚本文件路径' 的方式。这里我写的两条是为该服务器增加路由器的功能，即根据目标地址选择相应的网卡进行NAT，我这台服务器上配有公网IP的网卡为ens3，要根据实际情况来设置。

注意，由于我们的双栈服务器要当成路由器来使用，所以要打开内核的相关模块，并且要打开防火墙

vi /etc/sysctl.conf 重点是下面这两行

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
net.ipv6.conf.all.forwarding=1
```

然后使用下面语句来生效

```
sysctl -p
```

防火墙配置属于基础知识，这里不再赘述
然后我们启动服务端的wireguard服务

```
wg setconf wg0 /etc/wireguard/wg0.conf
wg-quick up wg0
```

其中，第一句有可能会报Address的异常，我们不用管，只要第二局能启动wg0网卡即可，然后可以通过ifconfig命令查看当前的网卡配置

```
root@vultr:~# ifconfig
ens3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 45.66.229.137  netmask 255.255.254.0  broadcast
45.77.229.255
        inet6 fe80::5400:1ff:fee9:1cb8  prefixlen 64  scopeid 0x20<link>
        inet6 2401:19f0:7400:84fe:5400:1ff:fee9:1cb8  prefixlen 64
scopeid 0x0<global>
        ether 56:00:01:e9:1c:b8  txqueuelen 1000  (Ethernet)
        RX packets 60027  bytes 29461061 (29.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 65477  bytes 29978573 (29.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 226  bytes 20481 (20.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 226  bytes 20481 (20.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wg0: flags=209<UP,POINTOPOINT,RUNNING,NOARP>  mtu 1420
        inet 10.0.20.1  netmask 255.255.255.0  destination 10.0.20.1
        inet6 fd86::1  prefixlen 64  scopeid 0x0<global>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
txqueuelen 1000  (UNSPEC)
        RX packets 20216  bytes 3230204 (3.2 MB)
        RX errors 92  dropped 42  overruns 0  frame 92
        TX packets 20985  bytes 19153392 (19.1 MB)
        TX errors 42  dropped 163 overruns 0  carrier 0  collisions 0
```

然后开始写客户端节点，也就是IPv4环境的服务器的配置文件，和服务端节点的语法一样，不同的就是路由策略和密钥

```
[Interface]
PrivateKey = YDPB08gE6pxmSsUOEYmPcl6AovjJdWw8cvkkXUErclg=
Address = 10.0.20.2/24, fd86::2/64

[Peer]
PublicKey = T0hbccLqGqfapjoQ5ijgxQDRhQ4KPuqeYQIkKxWRiHM=
Endpoint = 45.66.229.137:36889
AllowedIPs = 0.0.0.0/0, ::/0
```

客户端的配置文件相当于把服务端的调换一下位置，然后把各自的弓腰换乘私钥，私钥换成公钥，这里的[Peer]的Endpoint要填写服务端wireguard连接的IP和端口，AllowedIPs 这个比较有误导性，不是说限制哪些IP可以连接的意思，而是说哪些目的地址的IP需要路由到该节点的意思。这里就是把所有的IPv4地址和IPv6地址都转发到服务端去。在windows机器上0.0.0.0/0这种写法可能会导致IPv4不通，还需要我们单独把服务端的IP写一条路由指向公网网卡，然后写成0.0.0.0/1,128.0.0.0/1的形式。

然后客户端也通过一样的方式启动wireguard，此时我们在服务端ping 客户端的IPv4 和IPv6地址，就可以通了

ping 10.0.20.2

ping fd86::2

此时客户端也可以访问IPv6站点了，看到自己在外网的IP就是wireguard服务端在外网的IPv6地址。

此时客户端只有一个内网的IPv6地址，没有公网的全局单播地址，那么要怎样才能从公网通过IPv6直接访问我们的客户端呢。

因为我们在购买云主机的时候，云主机厂商往往会给你一个/64的地址，也就是说你的双栈服务器可以配置2^64个全局单播的IPv6地址，一辈子都用不完，我们只需在服务端拿出一个IPv6地址，然后做DNAT转发，就可以把发送到该IPv6地址的数据包全部通过wireguard转发给客户端了，实现方法如下

首先给服务端双栈服务器配置两个IPv6地址

vi /etc/netplan/10-ens3.yaml

```
network:
  version: 2
  renderer: networkd
  ethernets:
    ens3:
      dhcp4: no
      addresses: [45.66.229.137/23,'2401:19f0:7400:84fe:
5400:01ff:fee9:1cb8/64','2401:19f0:7400:84fe::
2/64','2401:19f0:7400:84fe::3/64','2401:19f0:7400:84fe::4/64']
      gateway4: 45.66.1.1
      nameservers:
        addresses: [108.61.10.10]
      routes:
      - to: 169.254.0.0/16
        via: 45.66.1.1
        metric: 100
```

注意网关和DNS要根据实际来设置

2401:19f0:7400:84fe:5400:01ff:fee9:1cb8/64是运营商通过DHCP分配的地址，2001:19f0:7400:84fe::2/64是我要分配给客户端的地址，剩下那几个::3 和::4是留着备用的地址

然后使用下面命令来生效

```
netplan apply
```

然后使用ifconfig就会发现这几个地址都配置上了，在外网也能ping通这几个新加的IPv6地址

```
ens3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 45.66.229.137  netmask 255.255.254.0  broadcast
45.66.229.255
        inet6 fe80::5400:1ff:fee9:1cb8  prefixlen 64  scopeid 0x20<link>
        inet6 2401:19f0:7400:84fe::2  prefixlen 64  scopeid 0x0<global>
        inet6 2401:19f0:7400:84fe::3  prefixlen 64  scopeid 0x0<global>
        inet6 2401:19f0:7400:84fe::4  prefixlen 64  scopeid 0x0<global>
        inet6 2401:19f0:7400:84fe:5400:1ff:fee9:1cb8  prefixlen 64
scopeid 0x0<global>
        ether 56:00:01:e9:1c:b8  txqueuelen 1000  (Ethernet)
        RX packets 52116  bytes 27323543 (27.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 57379  bytes 27749047 (27.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

然后我们只需要使用ip6tables做一条DNAT策略，就可以把某个IPv6地址的数据包全都转发给客户端了（服务端不需要做SNAT，因为客户端把服务端当成网关了，我们上面的命令会让服务端自动做SNAT）

```
 ip6tables -t nat -A PREROUTING -d 2401:19f0:7400:84fe::2 -j DNAT --to-destination  fd86::2
```

这样在客户端连接好wireguard服务端之后，如果有人在IPv6网络访问2401:19f0:7400:84fe::2这个地址的时候，就相当于访问我们的客户端服务器了，我们IPv4 Only的客户端也就变成传说中的双栈服务器了。你需要做的只是把服务端的带宽尽可能提高，延迟尽可能的降低。因为很多云主机他的IPv4路由路径和IPv6的路径是不一样的。有时候IPv6的路径和延迟要远远大于IPv4