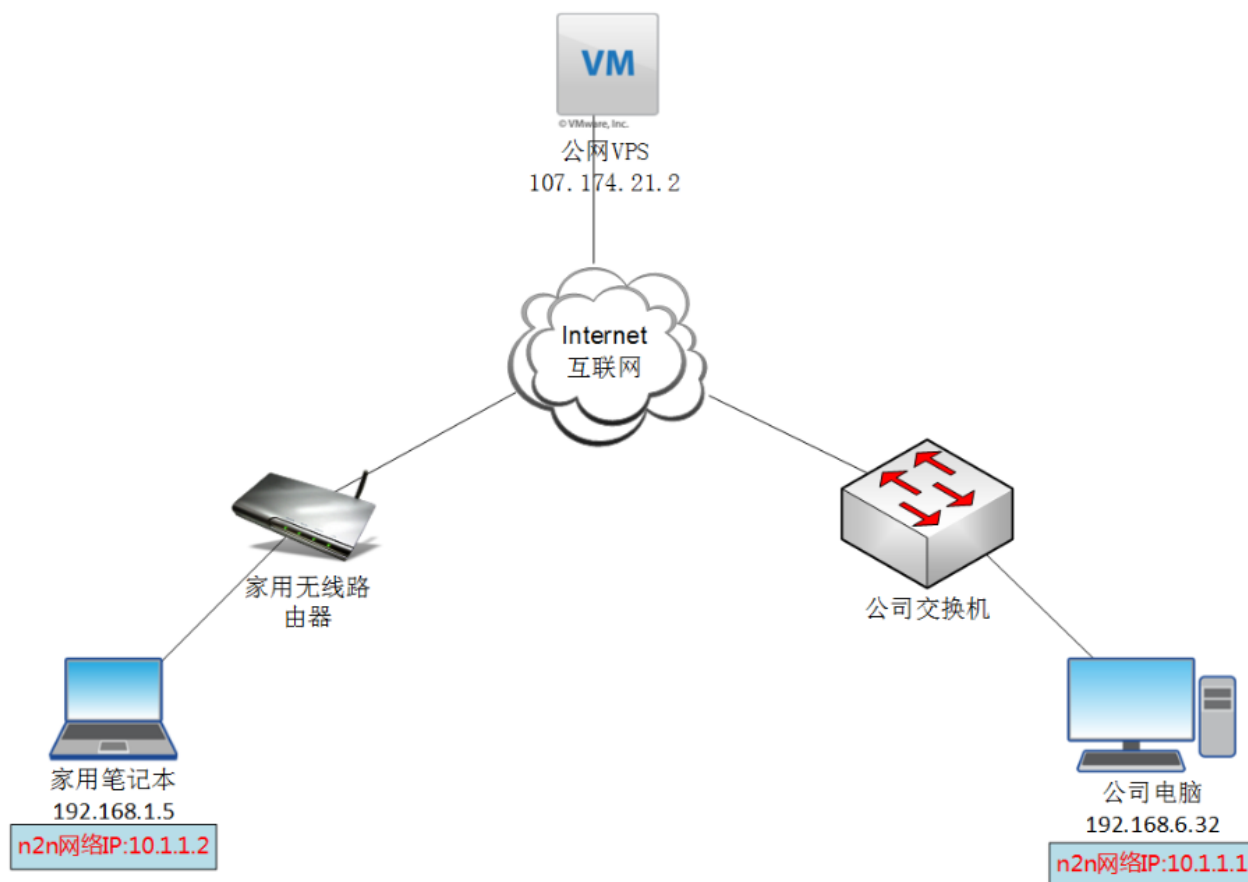


组建局域网

在几个服务器之间搭一个私有的局域网

网络拓扑：



公网服务器的配置

一、安装相关环境依赖

```
sudo yum groupinstall "Development tools" -y
```

二、下载我们的主角软件

```
[soft]$ git clone https://github.com/meyerd/n2n
Cloning into 'n2n'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 1190 (delta 0), reused 3 (delta 0), pack-reused 1187
Receiving objects: 100% (1190/1190), 507.64 KiB | 62.00 KiB/s, done.
Resolving deltas: 100% (761/761), done.
```

三、编译安装 cmake

```
wget https://cmake.org/files/v3.5/cmake-3.5.2.tar.gz
tar -zxvf cmake-3.5.2.tar.gz
cd cmake-3.5.2
./bootstrap # ./bootstrap --prefix=/usr
gmake
sudo gmake install
cmake --version
```

四、编译安装我们的主角 n2n

[安装教程](#)

```
cd n2n/n2n_v2/
mkdir build/
cd build/
cmake ..
make
sudo make install
```

在目录下生成了 supernode 和 edge 两个可执行程序

五、服务器端启动

```
supernode -l UDP监听端口
```

#参数说明

- f 前台运行
- u 指定运行所用的UID
- g 指定运行所用的GID
- v 输出比较详细的log

使用事例：

```
supernode -l 8080 >/dev/null &
nohup supernode -l 9000 &
```

六、客户端启动

```
edge -d 虚拟网卡名 -a 10.0.0.1 -c testnet -k senname -l 1.2.3.4:1234
```

#参数说明

- d 虚拟网卡名
- a [static:|dhcp:]虚拟网段(IP), static模式其实可以不用加那个static: 直接写IP就行
- c 用于区分节点的社区(组)名
- k 用于加密的字符串

-K 用于加密的key文件，和-k不能共存
-s 子网掩码
-l supernode的IP:端口，可以指定多个supernode的
-i NAT打洞间隔
-b 当使用DHCP时定期刷新IP
-p 指定本地端口
-u 指定运行所用的UID
-g 指定运行所用的GID
-f 前台运行
-m 为虚拟网卡指定MAC地址
-r 启用包转发，当-a指定DHCP时需要启用
-E 接收组播MAC地址
-v 输出比较详细的log
-t 指定用于管理的UDP端口

常用事例：

```
edge -d vircard -a 10.0.0.112 -c testnet -k abcd1234ABCD -l 10.0.0.1:2222
edge -d n2n0 -a 10.1.12.2 -c mynetwork -k encryptme -l 1.2.3.4:8080
edge -d n2n0 -a 10.1.12.1 -c mynetwork -k encryptme -l 1.2.3.4:8080 >/dev/null &
edge -d n2nvpn -c vpn -k password -a 10.1.1.1 -u 1000 -l 107.174.21.2:9000 >/dev/null &
edge -d n2nedge -a 172.10.2.210 -c community -u 500 -g 500 -k passwod -l
222.222.222.222:10088 -m 8B:2A:76:3F:C0:7B -r -L 192.168.1.110
```

如果创建虚拟网卡出问题了，可以尝试手动创建:

```
yum install tunctl -y
```

```
tunctl -t 虚拟网卡名
```

七、常见问题

- 怎么知道自己的 edge 是否已经与中心节点 supernode 连接成功了？

针对 N2N_V2 每两分钟显示一条，V2s 则每 25s 一条下面的信息 关键字：Rx REGISTER_SUPER_ACK myMAC

```
RX REGISTER_SUPER_ACK myMAC=EE:DC:38:99:C9:42 [123.249.20.254:10086] (external
218.89.10.162:50021). Attempts 2
```

- 如果中心节点 supernode 失效了会是什么样子呢？

针对 edge_v1，则每分钟显示

```
Registering with supernode
```

针对 edge_v2，则每 40 秒显示，针对 edge_v2s，则每 20 秒显示

```
WARNING: Supernode not responding - moving to 0 of 1
```

- 如果 edge 与 supernode 的版本对应错了，会是什么样子的？

edge_v1 试图与 supernode_v2 或 v2s 连接 (每分钟出现一条下面的信息)

```
Registering with supernode
```

edge_v2 或 v2s 试图与 supernode_v1 连接时, 反复出现如下信息 (V2 每 40 秒, v2s 每 20 秒)

```
WARNING: Supernode not responding - moving to 0 of 1
```

edge_v2 试图与 supernode_v2s (v2s 试图与 V2) 连接时, 反复出现如下信息 (V2 每 10 秒, v2s 每 5 秒)

```
RX REGISTER_SUPER_ACK myMAC=1D:C9:00:00:00:00 [123.249.20.254:10088] (external
218.89.10.162:42858). Attempts 2
WARNING: RX REGISTER_SUPER_ACK with wrong or old cookie.
```

八、网络测试

iperf的github

```
https://github.com/esnet/iperf/releases
```

解压

```
sudo tar -zxvf iperf-3.6.tar.gz -C /usr/local
```

进入目录, 编译

```
sudo ./configure
```

安装

```
sudo make
sudo make install
```

报错

```
iperf3: error while loading shared libraries: libiperf.so.0: cannot open shared object
file: No such file or directory
```

手动执行

```
sudo /sbin/ldconfig
```

可以访问了

```
# iperf3 -v
iperf 3.6 (cJSON 1.5.2)
Linux JQ-Mac 4.15.0-45-generic #48~16.04.1-Ubuntu SMP Tue Jan 29 18:03:48 UTC 2019 x86_64
optional features available: CPU affinity setting, IPV6 flow label, TCP congestion
algorithm setting, sendfile / zerocopy, socket pacing, authentication
```

开启服务

```
# iperf3 -s
-----
Server listening on 5201
-----
```

确保服务端开启5201端口

```
vim /etc/sysconfig/iptables
/etc/init.d/iptables restart
```

测试访问

```
# iperf3 -c 192.168.1.200
Connecting to host 192.168.1.200, port 5201
[ 5] local 192.168.1.228 port 58092 connected to 192.168.1.200 port 5201
[ ID] Interval           Transfer     Bitrate      Retr  Cwnd
[ 5]  0.00-1.00    sec   11.3 MBytes  94.5 Mbits/sec    0   24.0 KBytes
[ 5]  1.00-2.00    sec   11.2 MBytes  94.1 Mbits/sec    0   26.9 KBytes
[ 5]  2.00-3.00    sec   11.2 MBytes  94.1 Mbits/sec    0   28.3 KBytes
[ 5]  3.00-4.00    sec   11.2 MBytes  94.2 Mbits/sec    0   43.8 KBytes
[ 5]  4.00-5.00    sec   11.2 MBytes  94.2 Mbits/sec    0   43.8 KBytes
[ 5]  5.00-6.00    sec   11.2 MBytes  93.7 Mbits/sec    0   43.8 KBytes
[ 5]  6.00-7.00    sec   11.1 MBytes  93.3 Mbits/sec    0   43.8 KBytes
[ 5]  7.00-8.00    sec   11.3 MBytes  94.4 Mbits/sec    0   66.5 KBytes
[ 5]  8.00-9.00    sec   11.3 MBytes  94.8 Mbits/sec    0   100 KBytes
[ 5]  9.00-10.00   sec   11.1 MBytes  93.2 Mbits/sec    0   100 KBytes
- - - - -
[ ID] Interval           Transfer     Bitrate      Retr
[ 5]  0.00-10.00   sec   112 MBytes  94.0 Mbits/sec    0             sender
[ 5]  0.00-10.04   sec   112 MBytes  93.6 Mbits/sec                  receiver

iperf Done.
```