Using veth and brctl for protecting HN and saving IP addresses

Configuration described below has been suggested by Ugo123. Thank you.

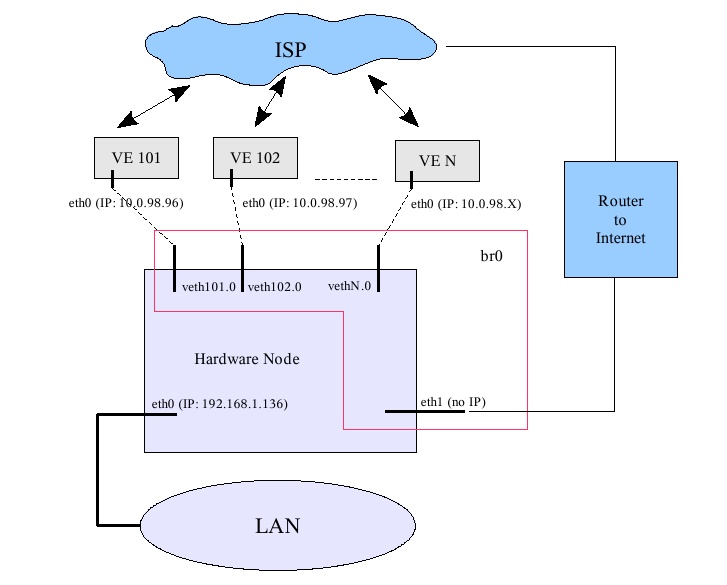
Consider we are facing the following task:

1. We have limited range of IP addresses granted by ISP. We want to assign as many granted IPs to containers as possible. We do not want to protect containers from Internet.
2. We want to protect the [HN](http://wiki.openvz.org/HN) OS ([CT](http://wiki.openvz.org/CT)0) from Internet and make it possible to manage containers from [CT0](http://wiki.openvz.org/CT0) within local area network.

Assume we have a [HN](http://wiki.openvz.org/HN) with 2 Ethernet cards (interfaces eth0 and eth1), OpenVZ kernel 2.6.18-028stab033, vzctl version 3.0.16, bridge-utils version 1.1. OpenVZ installation process is covered in [quick installation](http://wiki.openvz.org/Quick_installation).

This task can be effectively performed by setting up the configuration presented in Figure 1.

Figure 1: Effective configuration. 10.0.98.96-10.0.98.X - range of IP addresses granted by ISP, 192.168.1.136 - IP address from LAN

[](http://wiki.openvz.org/File:Fig.jpg)

Initial ifconfig output of HN is the following:

[HN]# ifconfig

eth0 Link encap:Ethernet HWaddr XX:XX:XX:XX:XX:34

inet addr:192.168.1.136 Bcast:192.168.3.255 Mask:255.255.252.0

inet6 addr: fe80::230:48ff:fe5b:ab34/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:3122 errors:0 dropped:0 overruns:0 frame:0

TX packets:246 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:325879 (318.2 KiB) TX bytes:57278 (55.9 KiB)

Interrupt:20

eth1 Link encap:Ethernet HWaddr XX:XX:XX:XX:XX:35

inet addr:192.168.0.32 Bcast:192.168.3.255 Mask:255.255.252.0

inet6 addr: fe80::213:d4ff:fe90:4d50/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:603734 errors:0 dropped:0 overruns:0 frame:0

TX packets:36627 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)

Interrupt:21

lo Link encap:Local Loopback

inet addr:127.0.0.1 Mask:255.0.0.0

inet6 addr: ::1/128 Scope:Host

UP LOOPBACK RUNNING MTU:16436 Metric:1

RX packets:1376 errors:0 dropped:0 overruns:0 frame:0

TX packets:1376 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:2078718 (1.9 MiB) TX bytes:2078718 (1.9 MiB)

Let us step through the setup process.

1) Create 2 containers on the HN as described in <http://download.openvz.org/doc/OpenVZ-Users-Guide.pdf>. For testing purposes I've used opensuse-10 precreated template from openvz.org:

[HN]# cd /vz/template/cache

[HN]# wget http://download.openvz.org/template/precreated/opensuse-10-i386-default.tar.gz

Create container 101 and assign it one of the IP addresses obtained from ISP:

[HN]# vzctl create 101 --ostemplate opensuse-10-i386-default --ipadd 10.0.98.96

[HN]# vzctl set 101 --userpasswd root:XXX --save

And do the same for CT 102 ... CT N. When ready - start containers:

[HN]# vzctl start 101

[HN]# vzlist -a

CTID NPROC STATUS IP\_ADDR HOSTNAME

101 4 running 10.0.98.96 -

102 4 running 10.0.98.97 -

2) By default containers use venet device for networking (see [venet](http://wiki.openvz.org/Venet)). But current configuration requires using alternative networking - through veth devices (see [Virtual Ethernet device](http://wiki.openvz.org/Virtual_Ethernet_device)). Switch CT 101 to veth by doing the following:

MAC address needed by eth0 of CT 101 and veth101.0 should be generated by easymac:

[HN]# wget http://www.easyvmx.com/software/easymac.sh

[HN]# chmod 0777 easymac.sh

[HN]# ./easymac.sh -R

00:0C:29:70:BB:34

[HN]# ./easymac.sh -R

00:0C:29:C0:2E:07

Replace venet by veth device on HN:

[HN]# ifconfig venet0:0 down

[HN]# vzctl set 101 --netif\_add eth0,00:0C:29:70:BB:34,veth101.0,00:0C:29:C0:2E:07 --save

[HN]# ifconfig veth101.0 0

[HN]# echo 0 > /proc/sys/net/ipv4/conf/veth101.0/forwarding

[HN]# echo 0 > /proc/sys/net/ipv4/conf/veth101.0/proxy\_arp

Enter the container and tune ifconfig within the container:

[CT 101]# vzctl enter 101

[CT 101]# ifconfig venet0:0 down

[CT 101]# ifconfig venet0 down

[CT 101]# ifconfig eth0 0

[CT 101]# ip addr add 10.0.98.96 dev eth0

[CT 101]# ip route add default dev eth0

The same (whole item 2) should be done for CT 102 ... CT N. 3) Now we should eliminate the IP address on eth1:

[HN]# vim /etc/sysconfig/network-scripts/ifcfg-eth1

Edit like this:

DEVICE=eth1

#BOOTPROTO=dhcp <<== comment out

HWADDR=XX:XX:XX:XX:XX:XX

ONBOOT=yes

and save changes (:wq).

[HN]# /etc/init.d/network restart

And turn off forwarding and proxy\_arp for eth1.

[HN]# ifconfig eth1 0

[HN]# echo 0 > /proc/sys/net/ipv4/conf/eth1/forwarding

[HN]# echo 0 > /proc/sys/net/ipv4/conf/eth1/proxy\_arp

</pre> 4) Create br0 bridge uniting eth1, veth101.0, ..., vethN.0:

[HN]# brctl addbr br0

[HN]# brctl addif br0 eth1

[HN]# brctl addif br0 veth101.0

..., veth102.0, vethN.0 etc.

[HN]# ifconfig br0 0

And turn off frowarding and proxy\_arp for br0:

[HN]# echo 0 > /proc/sys/net/ipv4/conf/br0/forwarding

[HN]# echo 0 > /proc/sys/net/ipv4/conf/br0/proxy\_arp

This is very important action. If skipped, network can be broken on further steps due to incoming arp-requests provoked storm.

As a result of above listed actions the ifconfig output like the following should be listed:

[HN]# ifconfig

br0 Link encap:Ethernet HWaddr 00:0C:29:A7:A9:D9

inet6 addr: fe80::20c:29ff:fea7:a9d9/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:79 errors:0 dropped:0 overruns:0 frame:0

TX packets:17 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:2972 (2.9 KiB) TX bytes:4390 (4.2 KiB)

eth0 Link encap:Ethernet HWaddr 00:30:48:5B:AB:34

inet addr:192.168.1.136 Bcast:192.168.3.255 Mask:255.255.252.0

inet6 addr: fe80::230:48ff:fe5b:ab34/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:347855 errors:0 dropped:0 overruns:0 frame:0

TX packets:4778 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:35964081 (34.2 MiB) TX bytes:698801 (682.4 KiB)

Interrupt:20

eth1 Link encap:Ethernet HWaddr 00:30:48:5B:AB:35

inet6 addr: fe80::230:48ff:fe5b:ab35/64 Scope:Link

UP BROADCAST MULTICAST MTU:1500 Metric:1

RX packets:322 errors:0 dropped:0 overruns:0 frame:0

TX packets:182 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:41943 (40.9 KiB) TX bytes:21338 (20.8 KiB)

Interrupt:21

lo Link encap:Local Loopback

inet addr:127.0.0.1 Mask:255.0.0.0

inet6 addr: ::1/128 Scope:Host

UP LOOPBACK RUNNING MTU:16436 Metric:1

RX packets:1376 errors:0 dropped:0 overruns:0 frame:0

TX packets:1376 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:2078718 (1.9 MiB) TX bytes:2078718 (1.9 MiB)

veth101.0 Link encap:Ethernet HWaddr 00:0C:29:C0:2E:07

inet6 addr: fe80::20c:29ff:fec0:2e07/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:363 errors:0 dropped:0 overruns:0 frame:0

TX packets:397 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:31134 (30.4 KiB) TX bytes:31440 (30.7 KiB)

veth102.0 Link encap:Ethernet HWaddr 00:0C:29:A7:A9:D9

inet6 addr: fe80::20c:29ff:fea7:a9d9/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:36 errors:0 dropped:0 overruns:0 frame:0

TX packets:60 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:1840 (1.7 KiB) TX bytes:2350 (2.2 KiB)

5) That is all. It's time to test the obtained configuration. Now plug eth1 of HN into network wall outlet provided by ISP and carry out the following testing:

- It should be tested that containers are accessible from Internet:

[INET]# ssh root@10.0.98.96

[CT 101]# ...

- HN is not accessible from Internet:

[INET]# ssh root@192.168.1.136

inaccessible

- containers can be managed from HN:

[HN]# vzctl enter 101

[CT 101]# ...

- containers CT 101, CT 102 .. CT N "see" each other (ping).

If all the steps are done as written, it should work. Enjoy.