Subject: IPv6 and OVZ part deux
Posted by lars.bailey on Mon, 03 Jan 2011 10:53:38 GMT
View Forum Message <> Reply to Message

Getting IPv6 working with OpenVZ,is either going to be heaven,or
hell,depending on several factors.
For IPv6 in general,it mostly involves;

* unfamiliar with IPv6 numeration(s)

* improper subnetting schemas

* improper routing configuration(s)

* OS specific configurations for IPv6

These,are easily remidied.
For OpenVZ,it is going to depend on the Node server operating system,and/or OS template cache.
If you think I'm on crack in my remark above,think again.
My testing has been done on F13 and Oracle Enterprise(EL5)
On F13,IPv6 and OpenVZ is a snap,and was tested with RedHat,Debian,and OpenSUSE containers.
On Oracle,or any EL5 flavor,it gets really interesting.
especially using EL5 template caches(CentOS)
This also goes for CentOS templates on F13.
You have to enable IPv6 in a EL5 container,just like you would on the Node server.
To prove IPv6 can easily be setup,this is a live IPv6 F13 container,running on F13 Node server.

=> Taken from my testing notes
The prefix below,is a private global range,used in testing IPv6

The IPv6 router's source route interface(Node),is configured with a IPv6 address from the "/64" prefix.

fd60:1014:9458:4b60::9

A test VE container was created,with virtual Ethernet,and its VETH interface,was also configured with an IPv6 address,from the available "/64" prefix.

fd60:1014:9458:4b60::a

Test for reachability.

ping6 -c 3 fd60:1014:9458:4b60::9
PING fd60:1014:9458:4b60::9(fd60:1014:9458:4b60::9) 56 data bytes
64 bytes from fd60:1014:9458:4b60::9: icmp_seq=1 ttl=64 time=0.369 ms
64 bytes from fd60:1014:9458:4b60::9: icmp_seq=2 ttl=64 time=0.315 ms

64 bytes from fd60:1014:9458:4b60::9: icmp_seq=3 ttl=64 time=0.242 ms

--- fd60:1014:9458:4b60::9 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.242/0.308/0.369/0.055 ms

ping6 -c 3 fd60:1014:9458:4b60::a

PING fd60:1014:9458:4b60::a(fd60:1014:9458:4b60::a) 56 data bytes
64 bytes from fd60:1014:9458:4b60::a: icmp_seq=1 ttl=64 time=0.329 ms
64 bytes from fd60:1014:9458:4b60::a: icmp_seq=2 ttl=64 time=0.342 ms
64 bytes from fd60:1014:9458:4b60::a: icmp_seq=3 ttl=64 time=0.346 ms

--- fd60:1014:9458:4b60::a ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.329/0.339/0.346/0.007 ms
#

"ping6" results show IPv6 networking is working on both interfaces.
/68 from /64 prefix,cut to a /96,and then down to "/120" prefix length.
May not be standard,but works in a pinch.
Secondary IPv6 address,from the the "/120" subnet.

FD60:1014:9458:4B60:E003:5000:0010:0101/128 VETH interface
FD60:1014:9458:4B60:E003:5000:0010:01FF/128 VE

ifconfig veth6101.0
veth6101.0 Link encap:Ethernet HWaddr 00:18:51:40:DF:09
    inet6 addr: fe80::218:51ff:fe40:df09/64 Scope:Link
    inet6 addr: fd60:1014:9458:4b60:e003:5000:10:101/64 Scope:Global
    inet6 addr: fd60:1014:9458:4b60::a/64 Scope:Global
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:48 errors:0 dropped:0 overruns:0 frame:0
    TX packets:99 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:0
    RX bytes:2274 (2.2 KiB) TX bytes:7664 (7.4 KiB)

The container.

[root@moe /]# ifconfig eth0
eth0   Link encap:Ethernet HWaddr 00:18:51:0C:72:E2
    inet6 addr: fd60:1014:9458:4b60:e003:5000:10:1ff/64 Scope:Global
    inet6 addr: fe80::218:51ff:fe0c:72e2/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:99 errors:0 dropped:0 overruns:0 frame:0
    TX packets:61 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:0

RX bytes:7664 (7.4 KiB) TX bytes:3166 (3.0 KiB)

A default route to the VETH interface was added

[root@moe /]# ip -6 ro show dev eth0
fd60:1014:9458:4b60::/64 proto kernel metric 256 mtu 1500 advmss 1440 hoplimit 0
fe80::/64 proto kernel metric 256 mtu 1500 advmss 1440 hoplimit 0
default via fd60:1014:9458:4b60:e003:5000:10:101 metric 1 mtu 1500 advmss 1440 hoplimit 0

Ping the default gateway.

ping6 -c 3 fd60:1014:9458:4b60:e003:5000:10:101
PING fd60:1014:9458:4b60:e003:5000:10:101(fd60:1014:9458:4b60:e00 3:5000:10:101) 56 data bytes
64 bytes from fd60:1014:9458:4b60:e003:5000:10:101: icmp_seq=1 ttl=64 time=0.989 ms
64 bytes from fd60:1014:9458:4b60:e003:5000:10:101: icmp_seq=2 ttl=64 time=0.375 ms
64 bytes from fd60:1014:9458:4b60:e003:5000:10:101: icmp_seq=3 ttl=64 time=0.381 ms

--- fd60:1014:9458:4b60:e003:5000:10:101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.375/0.581/0.989/0.289 ms

From the IPv6 router,to the container.

ping6 -c 3 fd60:1014:9458:4b60:e003:5000:10:1ff
PING fd60:1014:9458:4b60:e003:5000:10:1ff(fd60:1014:9458:4b60:e00 3:5000:10:1ff) 56 data bytes
64 bytes from fd60:1014:9458:4b60:e003:5000:10:1ff: icmp_seq=1 ttl=64 time=1.04 ms
64 bytes from fd60:1014:9458:4b60:e003:5000:10:1ff: icmp_seq=2 ttl=64 time=0.364 ms
64 bytes from fd60:1014:9458:4b60:e003:5000:10:1ff: icmp_seq=3 ttl=64 time=0.382 ms

--- fd60:1014:9458:4b60:e003:5000:10:1ff ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.364/0.597/1.046/0.317 ms
#

From the IPv6 router,VE6101 is reachable.
Router not reachable from VE,*may need route on Node - add /120 => VETH
Added direct route to the IPv6 router.
"route6-eth0" file,in container's /etc/sysconfig/network-scripts directory.

fd60:1014:9458:4b60::9 via fd60:1014:9458:4b60:e003:5000:10:101

ip -6 ro show dev eth0
fd60:1014:9458:4b60::9 via fd60:1014:9458:4b60:e003:5000:10:101 metric 1024 mtu 1500 advmss 1440 hoplimit 0
fd60:1014:9458:4b60::/64 proto kernel metric 256 mtu 1500 advmss 1440 hoplimit 0
fe80::/64 proto kernel metric 256 mtu 1500 advmss 1440 hoplimit 0

#

Ping IPv6 router.

ping6 -c 3 fd60:1014:9458:4b60::9
PING fd60:1014:9458:4b60::9(fd60:1014:9458:4b60::9) 56 data bytes
64 bytes from fd60:1014:9458:4b60::9: icmp_seq=1 ttl=64 time=1.48 ms
64 bytes from fd60:1014:9458:4b60::9: icmp_seq=2 ttl=64 time=0.377 ms
64 bytes from fd60:1014:9458:4b60::9: icmp_seq=3 ttl=64 time=0.375 ms

--- fd60:1014:9458:4b60::9 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.375/0.746/1.488/0.525 ms
#

Test container shows as reachable neighbor,without Proxy_NDP.
IPv6 transit over GRE tunnel6 successful.

""

Like I said,it couldn't have gotten any easier.
On Oracle,this same setup,breaks the IPv6 stack.
Loss of link-local,is common on source-route interface,and is causing me grief,although we do not use Oracle as a rule for OVZ.
If you are going to transition,better think ahead.
I'm glad I put a little effort first,before any assumptions.

Subject: Re: IPv6 and OVZ part deux
Posted by lars.bailey on Thu, 13 Jan 2011 07:30:28 GMT
View Forum Message <> Reply to Message

Before this thread is read any further,I need to clear up some things,in my haste to prove that IPv6 can be set up easily
with OpenVZ.
But,I also wanted to illustrate what not to do.
First off,do not use IFCONFIG,with IPv6 configurations.
IFCONFIG,assumes /64 prefix lengths.for host addresses.
This may pose a dilemma,for some end-users.
IPv6 configurations via scripting,or router software on Node?
We chose to add router software on Nodes,and add *init script to VE's.
The second,was using a /64 prefix length.
There have been inquiries elsewhere on the OVZ forum,on whether a /64 prefix can be subnetted?
Yes,a /64 can be subnetted.(use ipv6gen.pl)
Using the IP command manually;

1. Take a /64 private IPv6 address range from SimplyDNS,to test with.

I'm going to use this /64.

fd22:a075:afd0:e096::/64

Subnet it down to a /96.

FD22:A075:AFD0:E096:0000:0000::/96
FD22:A075:AFD0:E096:0000:0001::/96
FD22:A075:AFD0:E096:0000:0002::/96
FD22:A075:AFD0:E096:0000:0003::/96
FD22:A075:AFD0:E096:0000:0004::/96
FD22:A075:AFD0:E096:0000:0005::/96
FD22:A075:AFD0:E096:0000:0006::/96
FD22:A075:AFD0:E096:0000:0007::/96
FD22:A075:AFD0:E096:0000:0008::/96
FD22:A075:AFD0:E096:0000:0009::/96
FD22:A075:AFD0:E096:0000:000A::/96
=> etc.

I'm going to use this /96.

FD22:A075:AFD0:E096:0000:0001::/96

2. Add IPv6 address,to source-route interface.

```
[root@stooge network-scripts]# ip address add \
> fd22:a075:afd0:e096:0:1::1a dev eth1
```

3. Check with IFCONFIG.

```
[root@stooge network-scripts]# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:14:BF:5E:51:3F
          inet addr:192.168.1.64  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::214:bfff:fe5e:513f/64 Scope:Link
          inet6 addr: fd22:a075:afd0:e096:0:1:0:1a/128 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:477340 errors:0 dropped:0 overruns:0 frame:0
          TX packets:353145 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:245276422 (233.9 MiB)  TX bytes:174651991 (166.5 MiB)
          Interrupt:9 Base address:0x4800
```

Notice,it shows up as a hermit host.
Always add prefix length.
So again,add IPv6 address.

```
[root@stooge network-scripts]# ip address add \
> fd22:a075:afd0:e096:0:1::1a/96 dev eth1
```

4. Check with IFCONFIG.

```
[root@stooge network-scripts]# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:14:BF:5E:51:3F
          inet addr:192.168.1.64  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::214:bfff:fe5e:513f/64 Scope:Link
          inet6 addr: fd22:a075:afd0:e096:0:1:0:1a/96 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:477443 errors:0 dropped:0 overruns:0 frame:0
          TX packets:353163 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:245282889 (233.9 MiB)  TX bytes:174653483 (166.5 MiB)
          Interrupt:9 Base address:0x4800
```

5. Take the /96,and subnet to a /116.

```
FD22:A075:AFD0:E096:0000:0001:0000:0000/116
FD22:A075:AFD0:E096:0000:0001:0000:1000/116
FD22:A075:AFD0:E096:0000:0001:0000:2000/116
FD22:A075:AFD0:E096:0000:0001:0000:3000/116
FD22:A075:AFD0:E096:0000:0001:0000:4000/116
FD22:A075:AFD0:E096:0000:0001:0000:5000/116
FD22:A075:AFD0:E096:0000:0001:0000:6000/116
FD22:A075:AFD0:E096:0000:0001:0000:7000/116
FD22:A075:AFD0:E096:0000:0001:0000:8000/116
FD22:A075:AFD0:E096:0000:0001:0000:9000/116
FD22:A075:AFD0:E096:0000:0001:0000:A000/116
=> etc
```

As a rule,we do not subnet any given prefix,any further than /116.
Some may feel,this is "address waste",but with IPv6,we have no
reason to get anal about it.
You choose your method.
I'm going to use this /116 prefix.

```
FD22:A075:AFD0:E096:0000:0001:0000:1000/116
```

For the VETH and VE interfaces,I will use these host addresses.

```
VETH - fd22:a075:afd0:e096:0:1:0:1fff/116
VE eth0 - fd22:a075:afd0:e096:0:1:0:1001/116
```

6. Configure IPv6 address for VETH interface.

```
[root@stooge network-scripts]# ip address add \
> fd22:a075:afd0:e096:0:1:0:1fff/116 dev veth6101.0
```

7. Check with IFCONFIG.

```
[root@stooge network-scripts]# ifconfig veth6101.0
veth6101.0 Link encap:Ethernet  HWaddr 00:18:51:2C:C8:6D
          inet addr:192.168.100.101  Bcast:192.168.100.255  Mask:255.255.255.0
          inet6 addr: fd22:a075:afd0:e096:0:1:0:1fff/116 Scope:Global
          inet6 addr: fe80::218:51ff:fe2c:c86d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:432 errors:0 dropped:0 overruns:0 frame:0
          TX packets:83 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:16956 (16.5 KiB)  TX bytes:5716 (5.5 KiB)

[root@stooge network-scripts]#
```

8. For the VE.

Configure IPv6 address from this same subnet.

FD22:A075:AFD0:E096:0000:0001:0000:1001/116

```
[root@moe /]# ip address add \
> fd22:a075:afd0:e096:0:1:0:1001/116 dev eth0
```

9. Check with IFCONFIG.

```
[root@moe /]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:18:51:8F:16:2E
          inet addr:192.168.101.1  Bcast:192.168.101.255  Mask:255.255.255.0
          inet6 addr: fd22:a075:afd0:e096:0:1:0:1001/116 Scope:Global
          inet6 addr: fe80::218:51ff:fe8f:162e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:83 errors:0 dropped:0 overruns:0 frame:0
          TX packets:446 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:5716 (5.5 KiB)  TX bytes:17784 (17.3 KiB)

[root@moe /]#
```

10. Add default route.

```
[root@moe /]# ip ro add default via \
> fd22:a075:afd0:e096:0:1:0:1fff dev eth0
```

11. Check routing table.

```
[root@moe /]# ip -6 ro show dev eth0
fd22:a075:afd0:e096:0:1:0:1000/116  proto kernel  metric 256  mtu 1500 advmss 1440 hoplimit
```

4294967295
fe80::/64 proto kernel metric 256 mtu 1500 advmss 1440 hoplimit 4294967295
default via fd22:a075:afd0:e096:0:1:0:1fff metric 1024 mtu 1500 advmss 1440 hoplimit
4294967295

12. Ping gateway.(VETH)

[root@moe /]# ping6 -c 3 fd22:a075:afd0:e096:0:1:0:1fff
PING fd22:a075:afd0:e096:0:1:0:1fff(fd22:a075:afd0:e096:0:1:0:1ff f) 56 data bytes
64 bytes from fd22:a075:afd0:e096:0:1:0:1fff: icmp_seq=1 ttl=64 time=1.43 ms
64 bytes from fd22:a075:afd0:e096:0:1:0:1fff: icmp_seq=2 ttl=64 time=0.270 ms
64 bytes from fd22:a075:afd0:e096:0:1:0:1fff: icmp_seq=3 ttl=64 time=0.287 ms

--- fd22:a075:afd0:e096:0:1:0:1fff ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2011ms
rtt min/avg/max/mdev = 0.270/0.663/1.433/0.544 ms

13. Check your neighbor entry.

[root@moe /]# ip -6 neigh show
fd22:a075:afd0:e096:0:1:0:1fff dev eth0 lladdr 00:18:51:2c:c8:6d REACHABLE
fe80::218:51ff:fe2c:c86d dev eth0 lladdr 00:18:51:2c:c8:6d REACHABLE

The VETH interface,will always be the container's "neighbor".
14. Now,the GW router.(Node)

[root@moe /]# ping6 -c 3 fd22:a075:afd0:e096:0:1:0:1a
PING fd22:a075:afd0:e096:0:1:0:1a(fd22:a075:afd0:e096:0:1:0:1a) 56 data bytes
64 bytes from fd22:a075:afd0:e096:0:1:0:1a: icmp_seq=1 ttl=64 time=0.944 ms
64 bytes from fd22:a075:afd0:e096:0:1:0:1a: icmp_seq=2 ttl=64 time=0.282 ms
64 bytes from fd22:a075:afd0:e096:0:1:0:1a: icmp_seq=3 ttl=64 time=0.254 ms

--- fd22:a075:afd0:e096:0:1:0:1a ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.254/0.493/0.944/0.319 ms

Subject: Re: IPv6 and OVZ part deux
Posted by Jean-Marc Pigeon on Tue, 18 Jan 2011 20:45:36 GMT
View Forum Message <> Reply to Message

Bonjour,

> Found difficult to set up IPV6 on a container, the key part
> to catch is the bridge position within the stack. I was already
> using veth with IPV4 config but it was not working properly
> with IPV6.

Found lack of detailed example setup making "life difficult",
so I am providing my test setup, such it can be used as
starting point to some of you and a base for advise
or suggestion for those with an IPV6/IPV4 setup already
working.

On an standard IPV4 with VETH here is my production config (on the host side)

ifcfg-eth0=

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
HWADDR=00:26:B9:67:A7:E1
IPADDR=192.2Y.X.248
NETMASK=255.255.255.224
```

ifconfig eth0=

```
eth0      Link encap:Ethernet  HWaddr 00:26:B9:67:A7:E1
          inet addr:192.2Y.X.Y  Bcast:192.2Y.X.255  Mask:255.255.X.224
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:19359552 errors:0 dropped:0 overruns:0 frame:0
          TX packets:38144778 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5174070630 (4.8 GiB)  TX bytes:35554807003 (33.1 GiB)
          Interrupt:177 Memory:dxXf0000-dff00000
```

ifcfg-br0=

```
DEVICE=br0
ONBOOT=yes
TYPE=Bridge
BOOTPROTO=static
IPADDR=192.0.2.1
NETMASK=255.255.255.255
```

ifconfig br0=

```
br0       Link encap:Ethernet  HWaddr 00:18:51:08:8D:E1
          inet addr:192.0.2.1  Bcast:192.0.2.1  Mask:255.255.255.255
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:135837164 errors:0 dropped:0 overruns:0 frame:0
          TX packets:144774563 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:98367107659 (91.6 GiB)  TX bytes:160370520042 (149.3 GiB)
```

with ifconfig VE as setup by vetctl start

```
testve    Link encap:Ethernet  HWaddr 00:18:51:EE:2A:1C
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

RX packets:48523 errors:0 dropped:0 overruns:0 frame:0
TX packets:279080 errors:0 dropped:336 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:2799240 (2.6 MiB) TX bytes:24558278 (23.4 MiB)

;---------- ----------
This configuration is workin alright on IPV4, but adding IPV6 make it
not working alright, the host was able to ping6 the VE but not the other way.

Found IPV6 Working config as such:

ifcfg-eth0=

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
BRIDGE=br0
HWADDR=00:19:D1:12:41:87
```

ifconfig eth0=

```
eth0      Link encap:Ethernet  HWaddr 00:19:D1:12:41:87
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2262 errors:0 dropped:0 overruns:0 frame:0
          TX packets:976 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:512724 (500.7 KiB)  TX bytes:123021 (120.1 KiB)
          Memory:e8000000-e8020000
```

ifcfg-br0=

```
DEVICE=br0
ONBOOT=yes
TYPE=Bridge
BOOTPROTO=dhcp
IPV6INIT=yes
IPV6_AUTOCONF=yes
DHCPV6C=yes
```

ifconfig br0=

```
br0       Link encap:Ethernet  HWaddr 00:19:D1:12:41:87
          inet addr:192.2Y.X.242  Bcast:192.2Y.X.255  Mask:255.255.255.224
          inet6 addr: 2002::c02xY:Xf2/64 Scope:Global
          inet6 addr: xX80::2Y:d1ff:xX12:4187/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

RX packets:2514 errors:0 dropped:0 overruns:0 frame:0
TX packets:1140 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:514621 (502.5 KiB) TX bytes:142714 (139.3 KiB)

you have both IPV4 and IPV6 assigned via DHCP.

I added an alias to br0 to keep IPV4 routing capablity.
ifcfg-br0:vegw=
DEVICE=br0:vegw
ONBOOT=no
IPADDR=192.0.2.1
NETMASK=255.255.255.255

ifconfig br0:vegw=
br0:vegw Link encap:Ethernet HWaddr 00:19:D1:12:41:87
inet addr:192.0.2.1 Bcast:192.0.2.1 Mask:255.255.255.255
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

This complet the HOST side setup.

On the VE side
ifcfg-eth0=
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.2Y.X.77
NETMASK=255.255.255.255
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6ADDR=2001:2xY8::192.2Y.X.77/64
IPV6ADDR_SECONDARIES=2002::192.2Y.X.77/64

ifconfig eth0=
eth0 Link encap:Ethernet HWaddr 00:18:51:1E:6D:F4
inet addr:192.2Y.X.77 Bcast:192.2Y.X.77 Mask:255.255.255.255
inet6 addr: xX80::218:51ff:xX1e:6df4/64 Scope:Link
inet6 addr: 2002::c02xY:xX4d/64 Scope:Global
inet6 addr: 2001:2xY8::c02xY:xX4d/64 Scope:Global
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:2671 errors:0 dropped:0 overruns:0 frame:0
TX packets:257 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:585756 (572.0 KiB) TX bytes:27772 (27.1 KiB)

As usual you have file route-eth0=
192.0.2.0/24 dev eth0 scope link

default via 192.0.2.1 dev eth0

on the VE
[root@testvz network-scripts]# ping6 -n -c 3 2002::c02xY:xXf2
PING 2002::c02xY:xXf2(2002::c02xY:xXf2) 56 data bytes
64 bytes from 2002::c02xY:xXf2: icmp_seq=0 ttl=64 time=0.445 ms
64 bytes from 2002::c02xY:xXf2: icmp_seq=1 ttl=64 time=0.035 ms
64 bytes from 2002::c02xY:xXf2: icmp_seq=2 ttl=64 time=0.034 ms

on the HOST
ping6 -c 3 -n 2002::c02xY:xX4d
PING 2002::c02xY:xX4d(2002::c02xY:xX4d) 56 data bytes
64 bytes from 2002::c02xY:xX4d: icmp_seq=0 ttl=64 time=1.77 ms
64 bytes from 2002::c02xY:xX4d: icmp_seq=1 ttl=64 time=0.037 ms
64 bytes from 2002::c02xY:xX4d: icmp_seq=2 ttl=64 time=0.041 ms

Sure enough IPV4 stack is still fully fonctionnal

my 2 cents, have fun and keep me posted.

Subject: Re: IPv6 and OVZ part deux
Posted by lars.bailey on Thu, 20 Jan 2011 18:12:34 GMT
View Forum Message <> Reply to Message

I was getting ready to post my final conclusion to an issue in another thread,on using Ethernet bridging with IPv6.
Your example confirmed my thinking,into a proper way of using IPv6 with bridged Ethernet.
My thread here,mainly dealt with IPv6/OVZ,using non-bridged virtual Ethernet,and this type of setup is clean and simple.
I have experimented with IPv6/OVZ/Ethernet bridging,and also ran into issues,using IPv6 with parallel bridges vs. source-route.
Using an interface other than source-route,for a IPv6 Ethernet bridge,caused link-local breakdowns.
It's not uncommon to use a parallel interface,for a IPv4 Ethernet bridge.
Proxy_ARP,is only needed for the bridge.
In using this type setup with IPv6,sans Proxy_ARP,the VE was reachable,but not the other way around.
Once I try to "ping6" the Node source-route interface from a VE,link-local is lost on all bridged interfaces,including Ethernet bridge.
This may be due to my part,in the way the network configurations was setup.
The network daemon,did not initialize the VETH interfaces.
They were set to;

ONBOOT=no

This kept the bogus "boot error" messages off the Node,where the VETH interface was not found,and was initialized by the vz daemon.
This presented no problems with IPv4.
Once I set this back to;

ONBOOT=yes

and let the network daemon initialize the interface,link-local would come back on the VETH interfaces.
Removing the physical interface from the bridge,and restarting
the interface(s),I would have link-local back on the bridge and
physical interface.
These,was always set to ONBOOT=yes.
Why link-local is lost,is a mystery to me.
Since IPv4 uses ARP and IPv6 uses NDP,enabling Proxy_NDP,did not
help in this type of bridge setup.
I simply removed the parallel bridge configuration,and created a Ethernet bridge using the source-route interface,and IPv6 over bridged Ethernet works perfect
And as you stated,we do use DHCP for IPv4,to provide routing
info and DNS,and this will be added later via DHCPv6.
I'm going to try a couple of things from your post,on our test network.
I'll let you know how it went.
Thanks for your post.

Regards

Lars Bailey

Subject: Re: IPv6 and OVZ part deux
Posted by lars.bailey on Fri, 21 Jan 2011 04:24:54 GMT
View Forum Message <> Reply to Message

@ Jean-Marc

This is what I now use on a test Node,for IPv6/OVZ/Ethernet bridging.

The Node server;

* /etc/sysconfig/network

NETWORKING=yes
HOSTNAME=stooge
GATEWAYDEV=virtbr0
NETWORKING_IPV6=yes
IPV6INIT=yes

IPV6FORWARDING=yes
IPV6_AUTOCONF=no
IPV6_AUTOTUNNEL=no

This is the source-route bridge network configuration.

* ifcfg-virtbr0

DEVICE=virtbr0
TYPE=Bridge
ONBOOT=yes
STP=off
DELAY=0
BOOTPROTO=static
IPADDR=192.168.1.72
NETMASK=255.255.255.0
IPV6ADDR=fd22:a075:afd0:e096::101/64
IPV6_DEFAULTGW=fd22:a075:afd0:e096::1FF

I used a private IPv6 address range,for internal testing.
Node IPv6 routing table.

ip -6 ro show dev virtbr0
fd22:a075:afd0:e096::/64 proto kernel metric 256 mtu 1500 advmss 1440 hoplimit 4294967295
fe80::/64 proto kernel metric 256 mtu 1500 advmss 1440 hoplimit 4294967295
default via fd22:a075:afd0:e096::1ff metric 1 mtu 1500 advmss 1440 hoplimit 4294967295
#

A test VE container was created,and bound to the Ethernet bridge.

brctl show
bridge name bridge id STP enabled interfaces
virtbr0 8000.001851a86b76 no eth0
        veth6101.0
#

This is the test VE network configuration.

DEVICE=eth0
TYPE=Ethernet
ONBOOT=yes
IPV6INIT=yes
IPV6ADDR=fd22:a075:afd0:e096:65::65/64

From Node,the VE is reachable.

ping6 -c 3 fd22:a075:afd0:e096:65::65
PING fd22:a075:afd0:e096:65::65(fd22:a075:afd0:e096:65::65) 56 data bytes

64 bytes from fd22:a075:afd0:e096:65::65: icmp_seq=1 ttl=64 time=1.34 ms
64 bytes from fd22:a075:afd0:e096:65::65: icmp_seq=2 ttl=64 time=0.375 ms
64 bytes from fd22:a075:afd0:e096:65::65: icmp_seq=3 ttl=64 time=0.372 ms

--- fd22:a075:afd0:e096:65::65 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.372/0.696/1.342/0.457 ms
#

From VE,Node is reachable.

vzctl enter 6101
entered into VE 6101
[root@moe /]# ping6 -c 3 fd22:a075:afd0:e096::101
PING fd22:a075:afd0:e096::101(fd22:a075:afd0:e096::101) 56 data bytes
64 bytes from fd22:a075:afd0:e096::101: icmp_seq=1 ttl=64 time=1.50 ms
64 bytes from fd22:a075:afd0:e096::101: icmp_seq=2 ttl=64 time=0.403 ms
64 bytes from fd22:a075:afd0:e096::101: icmp_seq=3 ttl=64 time=0.401 ms

--- fd22:a075:afd0:e096::101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.401/0.769/1.503/0.519 ms
[root@moe /]#

It's pretty straight-forward,and no more worries on link-local breakage.
For IPv4,it was pretty straight-forward too.
This is the IPv4 container's routing.

[root@curly /]# ip ro show
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.64
169.254.0.0/16 dev eth0 scope link
default via 192.168.1.254 dev eth0

This should be self-explainatory.
Ping OpenVZ website.

[root@curly /]# ping -c 3 www.openvz.org
PING www.openvz.org (64.131.90.7) 56(84) bytes of data.
64 bytes from openvz.org (64.131.90.7): icmp_seq=1 ttl=48 time=38.3 ms
64 bytes from openvz.org (64.131.90.7): icmp_seq=2 ttl=48 time=40.1 ms
64 bytes from openvz.org (64.131.90.7): icmp_seq=3 ttl=48 time=38.3 ms

--- www.openvz.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2042ms
rtt min/avg/max/mdev = 38.381/38.982/40.184/0.879 ms
[root@curly /]#

What differs here now is,the IPv4 containers are directly routed,and I'm a little concerned about

DHCP time-outs,due to X bridged interfaces on source-route.
In using DHCP on the Node for network assignment,required the use of NAT rules,as the IPv4 containers resided on their own subnet.
This simplified route management for a VE.
Since,I have never used a source-route Ethernet bridge setup with DHCP,I think NAT rules still applies,but I will give your aliased interface a shot,and see what happens.
In truth,I'm not a big fan of source-route bridging.
But with IPv6/Ethernet bridging,this is going to be a common practice.
One arena I want to play around with,is using VDE.
I downloaded "openVswitch",compiled,and installed on the test Node.
Time is not a premium for me right now,and my technology mistress is going to accompany me,in finding a good divorce lawyer.(LOL)

Subject: Re: IPv6 and OVZ part deux
Posted by lars.bailey on Tue, 25 Jan 2011 13:03:09 GMT
View Forum Message <> Reply to Message

I tried the aliased interface solution,and found it wasn't needed
for my specific application,but thanks for the info.
For what its worth,I dumped the source-route bridge,as I prefer to
have the ability to provide other transit mechanisms.
Between Ethernet bridging vs. virtual Ethernet with IPv6,it has now
become moot.
Unless you need bridging for a specific application(auto-config),I simply go with IPv6 virtual Ethernet,as these types of containers,are now reachable from within other containers.(just like bridged containers)
I do like the idea of using multiple bridges(physically or virtually bound),to form bridge groups,and these have worked with no issues.
Since removing ProxyARP and ProxyNDP,for dual-homed bridges,I do not have the link-local breakdowns,I had before.
In hind-sight,this was always demonstrated in documentation on the
Net.

Subject: Re: IPv6 and OVZ part deux
Posted by lars.bailey on Wed, 02 Feb 2011 10:56:13 GMT
View Forum Message <> Reply to Message

Although I mentioned link-local breakdowns on Oracle server in this post,this issue was contributed to an error on my part.

We moved off of physically bound Ethernet bridges,to virtually bound bridges.
For this particular test server,I forgot to create and bind a TAP interface,to the bridge.
Once I created the TAP,bound it to the bridge,then everything worked as it should.
No ProxyARP and ProxyNDP.
No link-local breakdowns.
Since we do not use other network interfaces on the Node,for anything other than taking up MB slots,physically bound bridges did not make sense,as the Node's do not use source-route bridging.
Forgetting TAP interfaces for virtual bridges won't be a problem now,as the Node's will be retro-fitted with virtually switched VLANS,using VDE.
Live and learn.