The Perfect Xen Setup For Debian And Ubuntu - Page 6

5 Create A Virtual Local Network From The Virtual Machines (Optional)

(This chapter is optional, and what is described here can only be done if you installed Xen from the sources and compiled a *dom0* kernel with *iptables* and the *dummy* network driver as **modules** (see chapter 3.2).)

Now let's say you got a dedicated server in some data center that has one network card and only one IP address. Now you want to set up a web server (*vm01*) and a mail server (*vm02*) as virtual machines. If you would do it the way described above, you would need three public IP addresses (one for *dom0*, one for *vm01*, one for *vm02*), but you only got one. The solution is to set up a virtual local network on your server which means the *dom0* has the public IP address and acts as a router (doing *NAT*, network address translation), and behind that router we have a local network (in this example it is the network *192.168.3.0*).

This is how you do it (all these steps have to be made on *dom0*!):

First, we need a second network interface; it is for the local network. Since we have only one real network card (*eth0*) which has the public IP address, we use the *dummy* network driver to set up the

network interface *dummy0*.

```
echo dummy >> /etc/modules
```

Append the following part to */etc/network/interfaces*:

```
auto dummy0
iface dummy0 inet static
       address 192.168.3.1
       netmask 255.255.255.0
```

This will give *dummy0* the IP address *192.168.3.1*.

Then we have to tell Xen that it should bind the Xen bridge *xen-br0* to *dummy0*. Therefore you have to edit */etc/xen/scripts/network*. Change the line

```
netdev=${netdev:-eth0}
```

to

```
netdev=${netdev:-dummy0}
```

Of course, we have to change the network settings in */etc/xen/vm01-config.sxp* and */etc/xen/vm02-config.sxp*. *vm01* will have the IP address *192.168.3.2*, so its configuration file looks like this:

```
name ="vm01"
kernel ="/boot/vmlinuz-2.6.11.12-xenU
"
root ="/dev/hda1"
memory =128
disk = ['file:/vserver/images/vm01.im
g,hda1,w','file:/vserver/images/vm01-
swap.img,hda2,w']

# network
nics=1
dhcp ="off"
ip="192.168.3.2"
netmask="255.255.255.0"
gateway="192.168.3.1"
hostname="vm01.example.com"

extra="3"
```

Now we have to tell *dom0* that it should do *NAT* so that the virtual machines have internet access. We also have to tell *dom0* which ports it should forward to

which IP address. Therefore we create the file */etc/network/if-up.d/iptables*:

```
#!/bin/sh

echo "1" > /proc/sys/net/ipv4/ip_forw
ard
iptables -t nat -A POSTROUTING -s 192
.168.0.0/16 -j MASQUERADE

### Port Forwarding ###
iptables -A PREROUTING -t nat -p tcp
-i eth0 --dport 80 -j DNAT --to 192.1
68.3.2:80
iptables -A PREROUTING -t nat -p tcp
-i eth0 --dport 25 -j DNAT --to 192.1
68.3.3:25
iptables -A PREROUTING -t nat -p tcp
-i eth0 --dport 110 -j DNAT --to 192.
168.3.3:110
```

The first two commands enable *Nat*'ing on *dom0*. In the section after *### Port Forwarding ###* you put as many rules as you need. This tells *dom0* to forward certain ports to certain destination ports on certain destination IP addresses. For example, the first rule tells *dom0* to forward requests on port 80 (*http*) to port 80 on *192.168.3.2*. So if you have a web server running on *vm01* (*192.168.3.2*), then all requests on port 80 on *dom0* will be forwarded to this web server. The last two rules forward ports 25 (*smtp*) and 110 (*pop3*) to our mail server *vm02* (*192.168.3.3*).

Now we have to make that script executable:

chmod 755 /etc/network/if-up.d/iptables

Finally, we reboot the server:

shutdown -r now

After the reboot, you should have a virtual local network on your Xen system!

Whenever you need new port forwarding rules, put them at the end of */etc/network/if-up.d /iptables*. And because you do not want to reboot your system whenever you need new port forwarding rules, you can run the same rule on the shell. For example, if you want to forward port 21 (*ftp*) to *vm01*, you put the rule

iptables -A PREROUTING -t nat -p tcp -i eth0

```
--dport 21 -j DNAT --to 192.168.3.2:21
```

at the end of */etc/network/if-up.d/iptables*. Plus, you run this rule on the shell so that it becomes valid immediately:

```
iptables -A PREROUTING -t nat -p tcp -i eth0
--dport 21 -j DNAT --to 192.168.3.2:21
```

Links

Xen: http://www.cl.cam.ac.uk/Research/SRG/netos/xen/

Debian: http://www.debian.org/

Ubuntu: http://www.ubuntu.com/

<< Prev

Sub pages

5 Comment(s)

Add comment

Name * Email *

p

Submit comment

Comments

From: Anonymous Reply

Thx for this nice in depth article. Fine work. I would like to point out 2 things. As of Nov 13. Intel releases its VT processors[1], so you can run even an unmodifyed windows within a Xen 3.0 domain. There is another Xen related debian project called Xenophilia that is worth a look.
[1] http://www.nordichardware.com/news,2253.html
[2] http://cosi.clarkson.edu/xen/

the polarizer
http://www.codixx.de/polarizer.html

From: Anonymous Reply

needed for iptables:
apt-get install modconf module-init-tools

From: falko Reply

Yes, as soon as anybody cares to answer this question:
http://lists.xensource.com/archives/html/xen-users/2006-01/msg00432.html
It seems I'm having the same problem than you... :-(

From: Anonymous Reply

Hi there,
first of all, thanks for the really great setup tutorial. It worked out great for me.
Could you also provide help on how to set up the virtual local network in Xen 3.0, as this page seems not applicable to that version.

Kind regards
Nico

From: Anonymous Reply

Instead of dummy driver you could just use the IP alias feature (eth0:1)! correct?