Subject: domain names / virtual machines
Posted by Thomasd on Mon, 07 Jan 2008 09:42:01 GMT
View Forum Message <> Reply to Message

here's a scenario:

two domain names: a.com and b.com
one ip for the HW node: 1.1.1.1
two VMs, 100 and 101

if I want to have any traffic from a.com go to VM 100 and any traffic from b.com go to VM 101, what are the options?

Should I run an apache reverse proxy from the HW node or another VM? or are there any other solutions?

Subject: Re: domain names / virtual machines
Posted by Thomasd on Tue, 15 Jan 2008 00:50:28 GMT
View Forum Message <> Reply to Message

for the curious, I managed to get this working using a reverse proxy (pound) and the setup is very straightforward

Subject: Re: domain names / virtual machines
Posted by DiSH on Thu, 28 Feb 2008 09:41:21 GMT
View Forum Message <> Reply to Message

Well... And what to do, if I have mail servers for theese domains as well?

Subject: Re: domain names / virtual machines
Posted by illc0mm on Thu, 28 Feb 2008 17:43:01 GMT
View Forum Message <> Reply to Message

You can setup postfix/sendmail on the Host node to relay for the 2 domains. It's more of a postfix/sendmail question than anything else.

You would then setup postfix/sendmail on the VE's as normal.

You could also setup another VE which would be your Mail Relay server and port forward port 25 to it, that's probably the more preferred approach since you can constrain the resources it uses inside that VE.

Subject: Re: domain names / virtual machines
Posted by DiSH on Thu, 28 Feb 2008 20:14:12 GMT

View Forum Message <> Reply to Message

Heh.... Well, and what if I want much more services, like LDAP, eJabberd, SIP (asterisk) and some specific a-la handmade?

Subject: Re: domain names / virtual machines
Posted by illc0mm on Thu, 28 Feb 2008 21:04:57 GMT

View Forum Message <> Reply to Message

Quote:Heh.... Well, and what if I want much more services, like LDAP, eJabberd, SIP (asterisk) and some specific a-la handmade?

It's Linux, just do what you normally do.

In the case of having only one public IP, Think of the host node as your Firewall/NAT router. You can make iptables rules to redirect traffic just like you would in a NAT situation with physical machines. See:

http://wiki.openvz.org/Using_NAT_for_VE_with_private_IPs

and

http://wiki.openvz.org/Shared_webhosting

The key is to not install hosted apps on your HN. Anything an outsider touches should really be in a VE, otherwise it kind of defeats the purpose in using OpenVZ.

I've done Asterisk in OpenVZ, it works pretty good as long as you're not using any hardware for like (like T1/Analog cards) since it can get a little more complicated.

You can also create a "private" network for things like LDAP and such if they're for "internal" use only.

Setting a VPN with OpenVPN is pretty straightforward as well. See:

http://wiki.openvz.org/VPN_via_the_TUN/TAP_device

The wiki has most of this stuff documented already. If you're having a particular problem, I'd suggest creating a post for it as this is now off topic.

-illc0mm

Subject: Re: domain names / virtual machines

Posted by DiSH on Fri, 29 Feb 2008 09:05:30 GMT
View Forum Message <> Reply to Message

Thanks for links.

Uhhh...
I meant I have 2 domains, which point to 1 white IP, and every domain has a lot of services (http, imap, smtp, jabber, svn, ssh, sybase, etc.)

What should I use in this case, if I want to use default ports for every service?
Is there a solution for such task? Maybe there is some kind of TCP/IP proxy, which can route packets depending on destination hostname?

Subject: Re: domain names / virtual machines
Posted by illc0mm on Fri, 29 Feb 2008 15:38:04 GMT
View Forum Message <> Reply to Message

If you want them in separate environments, I'd suggest something like the following. This is an over simplified setup but it should get the idea across.

Host Node:
IP 1.1.1.1 (external IP)
IP 192.168.0.1 (internal IP)

Firewall Rules:

```
#Port Forward TCP 25,21,80,110,143,443,whatever else from 1.1.1.1 to 192.168.0.101 (VE 101)
iptables -t nat -A PREROUTING -p tcp -d 1.1.1.1 --dport 25 -i eth0 -j DNAT --to-destination 192.168.0.101:25
iptables -t nat -A PREROUTING -p tcp -d 1.1.1.1 --dport 80 -i eth0 -j DNAT --to-destination 192.168.0.101:80
# etc...

# SNAT statement to allow VE nodes Internet access through the Host Node
iptables -A POSTROUTING -s 192.168.0.0/255.255.255.0 -o eth0 -j SNAT --to-source 1.1.1.1
```

VE Node 101 (your "proxy" node):
IP 192.168.0.101

apps:
 Perdition POP/IMAP proxy
 sshproxy-project
 pound proxy (http)
 postfix (relay)

VE Node 102 ("servera"):
IP 192.168.0.102

apps:
 apache
 ssh
 postfix (mta)
 courier-imap (pop/imap)
 whatever else...

VE Node 103 ("serverb"):
IP 192.168.0.103

apps:
 apache
 ssh
 postfix (mta)
 courier-imap (pop/imap)
 whatever else...

The only thing that is truly "name" aware will be your http connections since HTTP v1.1 provides a name based virtual facility. Nothing else, yet, can identify a server by name during the initial connection so you have to be more creative for those. IP packets don't contain name information so you have to resort to an application aware proxy. Usually you can do some sort of lookup on the user name or something to that effect to find out where to proxy the connection.

More about what runs on VE 101 (proxy node):

POP3/IMAP:
Perdition Mail Retrieval Proxy
http://www.vergenet.net/linux/perdition/

This will handle SSL and NON SSL connections and keys of the user account to find which server the user is going to. User accounts usually become the mailbox's full email address "user@domain.com". Perdition can then use several lookup methods (sql, hash, nis, regex, etc) to then proxy the connection to the correct server.

SMTP:
Postfix MTA
http://postfix.org

Postfix can be setup to accept messages for multiple domains, then relay those messages to the appropraite VE. Like perdition it can get that information from sql, hash, nis, regex, you name it. Makes it very flexiable. It's one of the more secure MTAs out there and it's pretty easy to configure.

HTTP/HTTPS:
http://www.apsis.ch/pound/

This can proxy both http and https (the back end communication will be http since it has to be decrypted in pound)

DNS:
Since everything is going to resolve to the same IP, this is pretty simple. I'd highly recommend maradns for this task, since it can use templates and make provisioning a snap, and it has an awesome security record. It has a low resource overhead and is easy to configure.

http://www.maradns.org/

SSH:
For this, you could use sshproxy-project

http://sshproxy-project.org/about/

You'd need to change the default port it runs on (port 2242 is the default) to port 22. At that point I'd also change the port my Host Node runs on to something else (like 22222 or whatever) so it will not conflict. That way, your users will have a seamless experience. sshproxy-project wasn't really designed for this purpose, but it should work fine.

Jabber:
Not sure on this one, never done it but I'm sure the people on Jabber support forum might have a clue on that one.

Sybase:
I would assume this would be internal and not exposed to the Internet. I'm not sure what sybase web tools exist out there (like phpmyadmin) but I would imagine that's what you'd want to do. Otherwise, this is where a VPN or port tunneling over SSH would come into play.

SVN:
You'll have to research that on your own, I imagine it's possible.

As far as what you run on server a and b, it doesn't matter since the proxy node just needs to hand it off to something.

Hope that helps.

-illc0mm

Subject: Re: domain names / virtual machines
Posted by DiSH on Sun, 02 Mar 2008 11:07:56 GMT

View Forum Message <> Reply to Message

Thanx, illc0mm!
Your answer is great, I've got the idea.

Subject: Re: domain names / virtual machines

Posted by mlowrie on Mon, 19 May 2008 23:58:53 GMT

View Forum Message <> Reply to Message

Hi,

I'm interesting in doing something similar in that each domain would be hosted on a separate VE node with one node being the master node that directs the request to the appropriate VE. This is due to only having a single puplic IP available for the multiple domains I host.

Have you had any success implementing this? I'm hoping someone has this working before I attempt it. Did you use the suggested proxying software or something else?

Thanks for the help.
Mike

Subject: Re: domain names / virtual machines
Posted by Thomasd on Tue, 20 May 2008 00:06:19 GMT

View Forum Message <> Reply to Message

yes, it was pretty easy in the end:

I used pound (http://www.apsis.ch/pound/) on the main node: it will accept the connection and based on a list of rules, redirect it to the VM of your choice.

so each VM is set with Apache listening on Port 80 like usual and Pound will match domain names and redirect.

a couple caveats:
- It doesn't seem to work with HTTPS (but it could be my mistake)
- Apache log options have to be changed if you want to log IPs; otherwise you'll only get the ip from the host node.

This is an example:

```
ListenHTTP

  Address your.host.ip.address
  Port    80
  xHTTP   2

  # mydomain1.com
  Service
    HeadRequire "Host:.*mydomain1.com.*"
    BackEnd
      Address my.virtualmachine1.ip.address
      Port    80
  End
 End

 # mydomain2.com
 Service
  HeadRequire "Host:.*mydomain2.com.*"
  BackEnd
   Address my.virtualmachine2.ip.address
   Port  80
  End
 End

End
```