5.1 LXC Networking Additional Considerations

Introduction

By default, your containers are accessible only from the host. For serious use you will want to expose some containers to the outside world. There are various ways of doing this. Currently I have settled on the following,

Port Forwarding using Static IPs with IP Tables - allows you to leverage your host's IP address (assuming it is public).

macvlan with Additional IP - allows you to have, a dedicated network interfaces (to the outside world) but actually only use one real physical network card. Unlike using a bridge this will not have the cpu overhead and need for your network card to work in promiscuous mode. Also, if you go with a hosting provider, bridge is probably not available as you're already inside of virtualization via something like KVM (explain this more for those not familiar with kvm later). This article builds on the work done in the introductory LXC article.

I actually use both techniques together.

> Make sure to change the password or better remove the default ubuntu account generated by the lxc creation script before making the container accessible to the Internet.

UFW

UFW in the Host

UFW is a great simple firewall, but at this point I do not recommend installing on your host if you intend to use port forwarding as there may be conflicts. Second, port forwarding using UFW is overly complex and seems like a hack versus it being very simple with IP Tables.

If you insist on using UFW, make sure to change the setting to not drop forwarded packets. I will revisit this later as I do like UFW. Perhaps I can ask the developers to make port forwarding more straight-forward.

UFW in a Container

Also, firewalls work at the kernel level. So you should not be installing UFW or even IP Tables inside of a container.

I will revisit this topic but believe it is due to modules not loading inside of containers /etc/modules and the container not being able to modify it.

```
sudo ufw allow 22
ERROR: initcaps
[Errno 2] modprobe: ERROR: ../libkmod/libkmod.c:556 kmod_search_moddep()
could not open moddep file '/lib/modules/3.13.0-57-generic/modules.dep.bin'
ip6tables v1.4.21: can't initialize ip6tables table `filter': Table does
not exist (do you need to insmod?)
Perhaps ip6tables or your kernel needs to be upgraded.
```

Trying to enable UFW inside of a container results in a a kernel needs to be upgraded error.

macvtap

This looks promising... The most prominent user of macvtap interfaces seems to be libvirt/KVM, which allows guests to be connected to macvtap interfaces. Doing so allows for (almost) bridged-like behaviour of guests but without the need to have a real bridge on the host, as a regular ethernet interface can be used as the macvtap's lower device.

References

Networking - https://help.ubuntu.com/lts/serverguide/lxc.html#lxc-network

Networking LXD More Detail - https://www.flockport.com/lxc-networking-guide/

INotes issue about bridge mode promiscuous mode - http://wiki.alpinelinux.org/wiki/LXC#Creating_a_LXC_container_without_modifying_your_network_interfaces

Getting macvlan working - https://www.flockport.com/lxc-macvlan-networking/

Learning macvlan and good background info - http://backreference.org/2014/03/20/some-notes-on-macvlanmacvtap/

Oracle article on details of container creation - http://docs.oracle.com/cd/E37670_01/E37355/html/ol_config_os_containers.html#ol_setup_fs_containers

Looks to be most comprehensive yet of what I want (multiple network cards with macvlan) - http://containerops.org/2013/11/19/lxc-networking/

How I figured out to create a macvlan - http://cyberiantiger.livejournal.com/24104.html

Not sure I can use comments here... need to investigate if it causes issues.

Generate mac address same way lxc does - http://giantdorks.org/alain/how-to-generate-a-unique-mac-address/

More in depth and discusses outbound NAT so containers can communicate to other container public IPs -http://blog.codeaholics.org/2013/giving-dockerlxc-containers-a-routable-ip-address/