LXC for Devs

What is it?

Cheap virtualization

- No security (who needs it)
- No resource limits (unless you turn them on)

Like a slightly more polished chroot

How do I get going?

```
sudo lxc-create -n myvirtmachine -t ubuntu
# assumes you are on ubuntu

sudo lxc-start -n myvirtmachine
# use -d to daemonize
```

Some technical details

- Shares the same kernel
  - assuming same architecture
  - you can see the processes running in the main os
- Linux only
- Alternative architectures via qemu
- Uses debootstrap for minimal debian flavour os downloads
- File system stored in /var/lib/lxc or can use LVVM volumes
  - BTRFS recommended for quick snapshots

General use

- See commands starting lxc-*
- I avoid lxc-halt, run halt from the machine
- lxc-console attaches you to a console, useful if you can't ssh

Creating machines

- The general scripts for creating the machines are in /usr/share/lxc/templates. I've modified mine to do some extra wiring.
- Be careful about the caching of os images if you do modify them
- lxc-clone -o original -n new-machine is an alternative way to create machines

Networking

- On Ubuntu things setup to just work
- A NAT bridge is setup for the boxes to talk to the internet
- Your box can talk to them too since it's hosting the bridge
- I've never tried to make the boxes running accessible from outside, but there's no reason you can not

General advice

- Set the boxes up to use something like apt-cacher. That should save on downloading packages again and again

Ubuntu 13.04

I started using Ubuntu 12.10 and there are a few differences now.

- lxc-ls now requires --active to see running machines
- there appears to be inter-machine name resolution (although the host can't resolve the names)