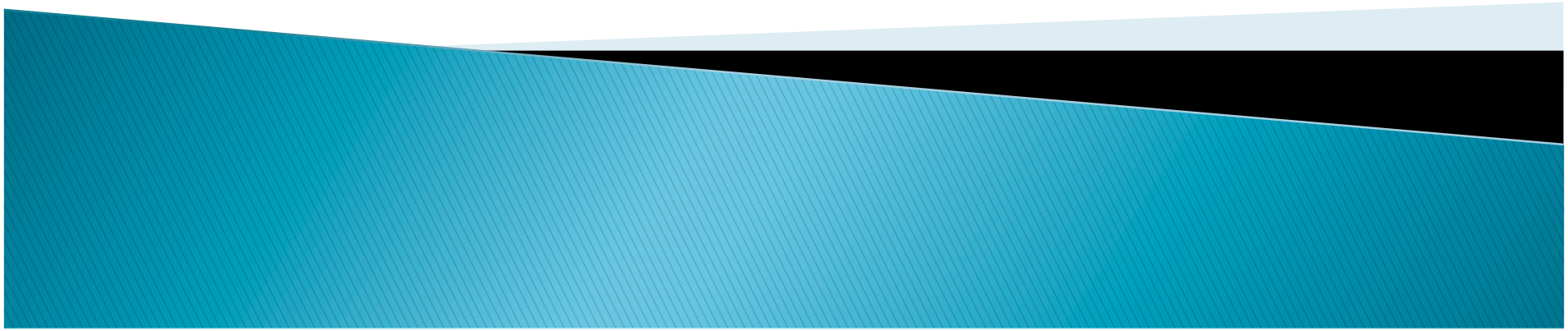
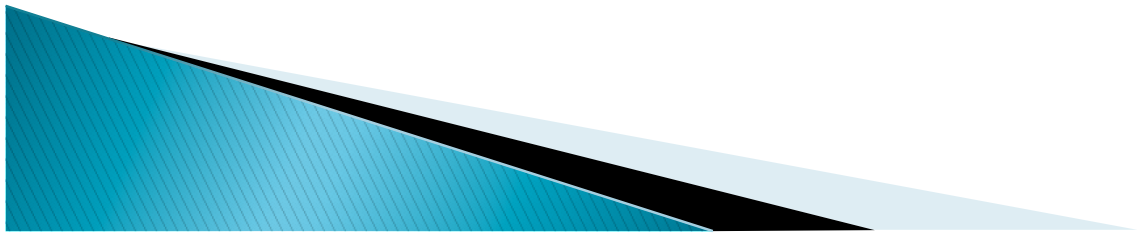
LXC – Linux Containers

Tom Eastep
LinuxfestNW 2011
April 30 – May 1, 2011
Bellingham, Washington
http://www.shorewall.net/LInuxfest2011.pdf
http://www.shorewall.net/LXC.html



Agenda

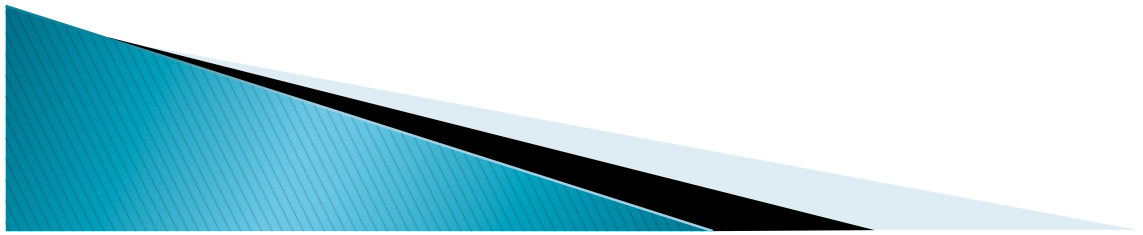
- Overview of LXC
- Getting Started
- Experience with a Linux-Vserver -> LXC Migration
- Issues
- Q & A



About the Presenter

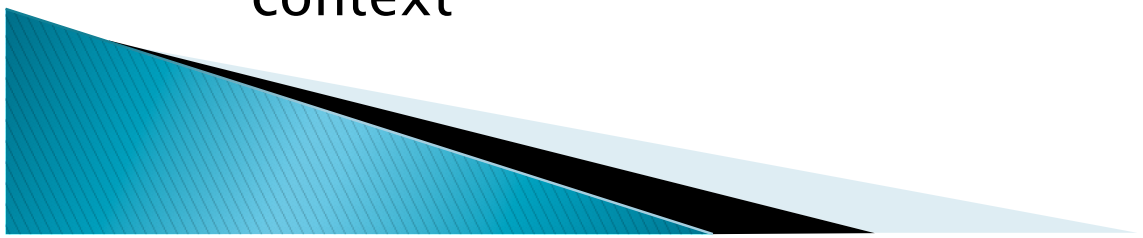
- System Software Architect with HP
- Creator/maintainer of Shorewall
- This presentation does not represent HP

- My bias is toward Debian
- Approximately 6 weeks Experience with LXC on Debian Squeeze
- Not involved in LXC development
  - I do subscribe to lxc-devel



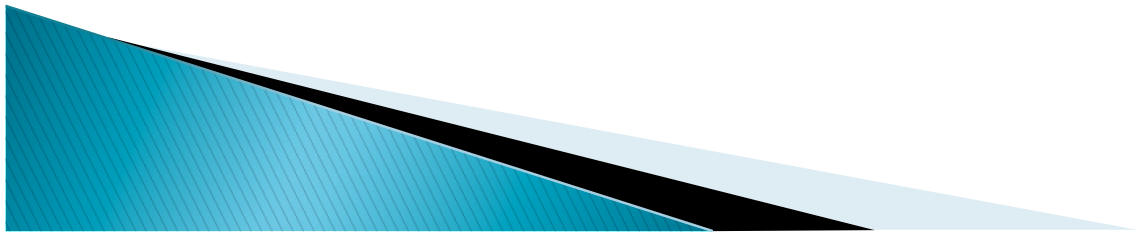
Terminology

- System-based Virtualization
  - Emulates hardware (virtual machine)
  - Examples: VMware, VirtualBox, Xen, KVM
- Kernel-based Virtualization
  - Partitions kernel resources into isolated security *contexts*
  - Examples: Virtuoso (OpenVZ), Linux-vserver, LXC
- Host
  - OS environment running outside of Virtual Machines and Security Contexts
- Guest
  - OS environment running in a virtual machine or security context



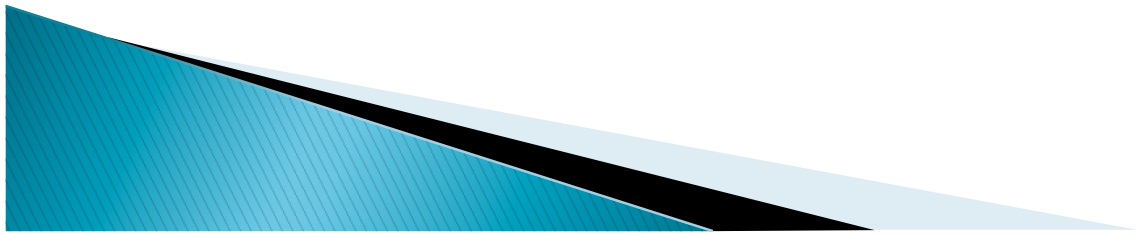
My Path to LXC

- First kernel-based virtualization solution was OpenVZ
- Wanted to Implement IPv6 support in Server
  - IPv6 didn't work in OpenVZ containers on Debian Lenny
- Switched to Linux-vserver
  - Stopping a vserver with an IPv6 configuration usually wiped out the IPv6 local routing table which killed IPv6 on the host and other vservers
- A Shorewall user recently introduced me to LXC



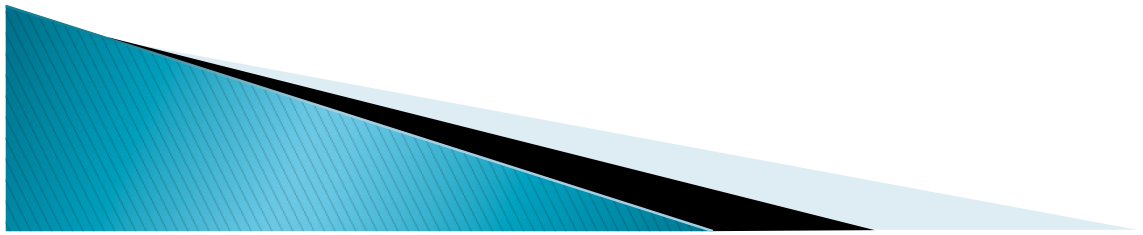
LXC Overview

- LXC is a set of user-space tools that complements the 'chroot on steroids' container capability introduced in Kernel 2.6.27 and made fully usable in 2.6.29
- Like OpenVZ, it is *container*-based
  - Single instance of Linux kernel on the box
  - Filesystem on a guest is a sub-tree of the host filesystem hierarchy
- Documentation is skimpy and oriented toward Arch Linux



LXC Overview – Continued

- CLI takes the form of many individual programs
  - lxc-start
  - lxc-info
  - lxc-console (like vserver .. enter)
  - ...
- To specify the target container, the '*-n*' or '*—name*' option is required
  - *lxc-start -n www*
- A *porcelain* with syntax similar to vserver was recently submitted to lxc-devel
  - *lxc www start*



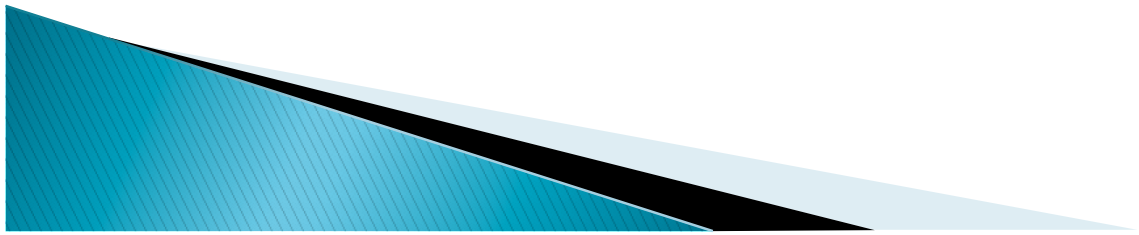
LXC Overview – Continued

- Good Network Virtualization
  - veth pair with optional host bridge (my choice)
  - vlan
  - macvlan
  - physical device
- Other areas not as well virtualized
  - /dev – but access can be restricted in containers
  - Only one kernel ring buffer so all kernel logging is done in the host
    - Shows up if Shorewall is run in a container
- Work in progress

LXC Overview – cgroup pseudo-filesystem

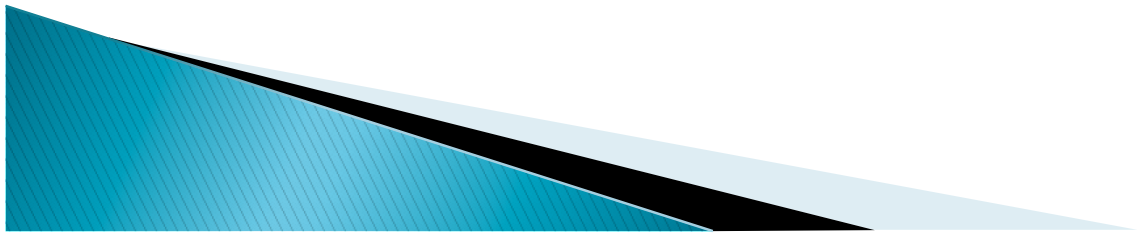
- Requires the '*cgroup*' (control group) pseudo-filesystem to be mounted
  - **`mkdir /cgroup`**
  - In /etc/fstab:

    `cgroup  /cgroup  cgroup defaults  0  0`
  - **`mount -a`**
- Interface for resource control and isolation of containers



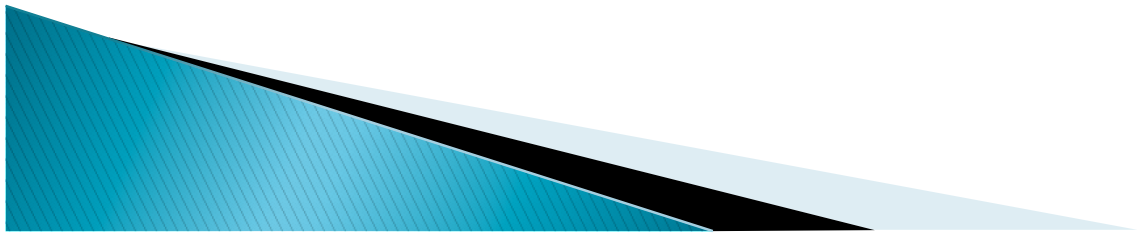
LXC Overview – cgroup pseudo-filesystem

```
root@gateway:/etc# ls /cgroup
cgroup.procs            cpuset.memory_pressure           devices.deny
cpuacct.stat            cpuset.memory_pressure_enables   devices.list
cpuacct.usage           cpuset.memory_spread_page        mail
cpuacct.usage_percpu    cpuset.memory_spread_slab        net_cls.classid
cpuset.cpu_exclusive    cpuset.mems                      notify_on_release
cpuset.cpus             cpuset.sched_load_balance        release_agent
cpuset.mem_exclusive    cpuset.sched_relax_domain_level  server
cpuset.mem_hardwall     cpu.shares                       tasks
cpuset.memory_migrate   devices.allow
root@gateway:/etc#
```



LXC Overview – cgroup pseudo-filesystem

```
root@gateway:/etc# ls /cgroup/mail
cgroup.procs            cpuset.memory_migrate           devices.allow
cpuacct.stat            cpuset.memory_pressure          devices.deny
cpuacct.usage           cpuset.memory_spread_page       devices.list
cpuacct.usage_percpu    cpuset.memory_spread_slab       freezer.state
cpuset.cpu_exclusive    cpuset.mems                     net_cls.classid
cpuset.cpus             cpuset.sched_load_balance       notify_on_release
cpuset.mem_exclusive    cpuset.sched_relax_domain_level tasks
cpuset.mem_hardwall     cpu.shares
root@gateway:/etc#
```

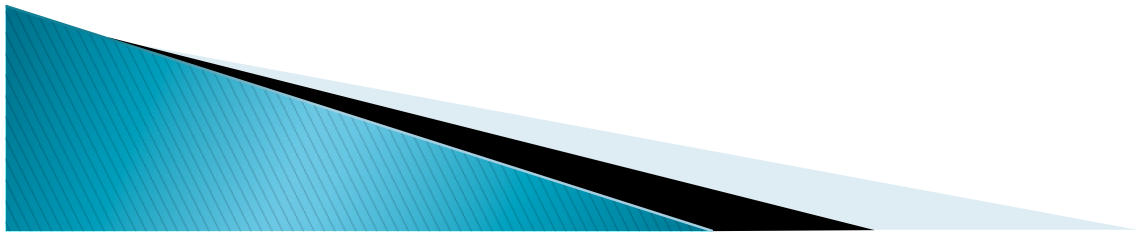


Getting Started

- Standard Distro Kernel – I use **linux-image-2.6.32-6-amd64**
- Install the **lxc** user-space tools package
  - Includes '*templates*'
    - Debian
    - Fedora
    - Ubuntu
- Unfortunately, the Debian template is for Debian *Lenny*!
  - http://jtrancas.wordpress.com/2011/02/10/debian-squeeze-lxc-template/

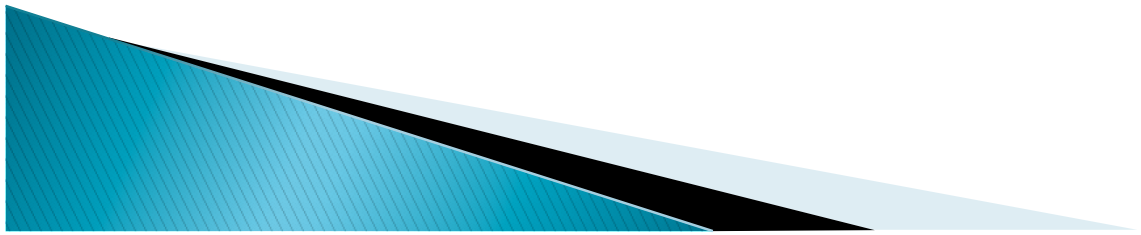
Getting Started

- Mount cgroup filesystem
- Create a container
  - You need a .conf file
  - Examples are rarer than hen's teeth (but there is one on the next slide ☺)
- lxc-create -c <conf file> -t <template>
  - `lxc-create –n <name> –c mylxc.conf –t squeeze` ← From link on preceding slide
- Creates:
  - /var/lib/lxc/<name>/config ← Subsequent changes must be made here
  - /var/lib/lxc/<name>/rootfs



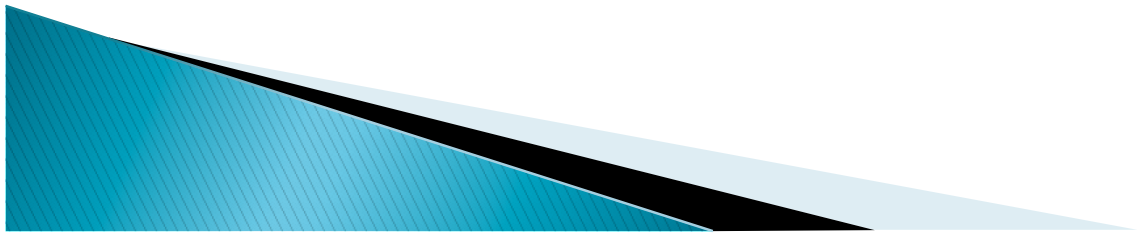
Getting Started – Sample Config File

```
lxc.utsname=mail ← Host name of container

lxc.network.type=veth ← Use veth pair
lxc.network.link=br0 ← Automatically add host veth to this bridge
lxc.network.veth.pair=veth0 ← Name of the host veth interface
lxc.network.flags=up ← Bring up host veth automatically

lxc.network.ipv4=70.90.191.124/29 ← IPv4 Address of container veth
lxc.network.ipv6=2001:470:b:227::42/124 ← IPv6 Address of container veth

lxc.cap.drop=sys_module sys_resource sys_chroot ← Capabilities not
   available in container

lxc.console=/dev/tty11 <- Host device to use as container console

lxc.tty = 4 ← Number of ttys available for lxc-console sessions
lxc.pts = 1024 ← Enables private PTS space in the container
```



Getting Started – Host Bridge Configuration

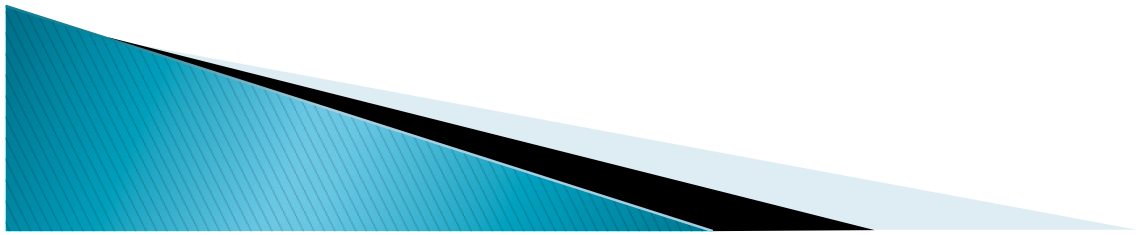
```
auto br0
iface br0 inet static
      address 70.90.191.121 ← Host's external IP address
      broadcast 0.0.0.0
      netmask 255.255.255.255
      bridge_ports none
      bridge_fd 0 ← Makes interfaces come up faster
      post-up ... ← Depending on your setup, may need to add routes

iface br0 inet6 static
      address 2001:470:b:227::41
      netmask 124
```



Getting Started – IPv6 Default Routes in Containers

- Need to modify /etc/radvd.conf on the host

```
interface br0{
  AdvSendAdvert on;
  AdvDefaultLifetime 9000;

  route ::/0 {
     AdvRouteLifetime infinity;
  };
};
```

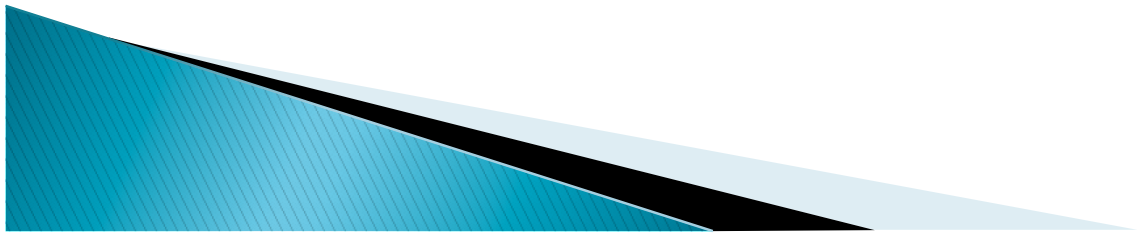


Getting Started – Container Startup at Boot

- Starting containers at boot
- /etc/default/lxc

```
# Comment out to run the lxc init script
RUN=yes
# Directory containing the container configurations
CONF_DIR=/etc/lxc# Start /etc/lxc/example.conf, /etc/lxc/
  autostart.conf, etc.
CONTAINERS="www mail"
```

- **ln –s /var/lib/lxc/www1/config /etc/lxc/www.conf**
- **ln –s /var/lib/lxc/mail/config /etc/lxc/mail.conf**



Comparison with Linux-vserver

Linux-vserver

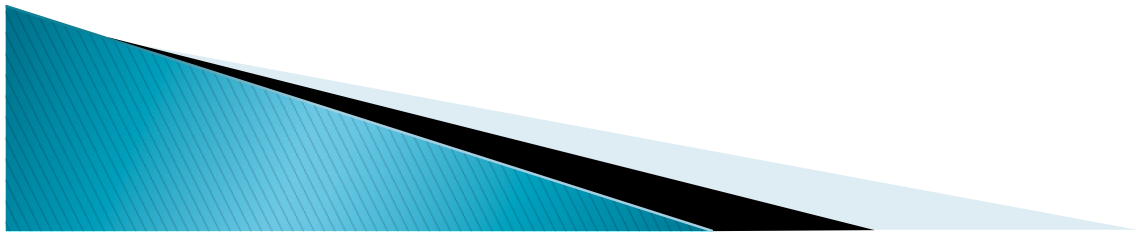
- Requires kernel patches
- Uses udevd
- Minimal /etc/inittab
- Minimal Network Virtualization
  - Firewall must be in the host
  - Unconventional configuration
- */etc/init.d/vserver stop* on the host does an orderly shutdown of the vservers

LXC

- Uses standard kernel
- Cannot use udev
- Needs /etc/inittab if you want consoles (including lxc-console)
- Fuller Network Virtualization (more like OpenVZ)
- Firewall possible in each container
  - Logging occurs on the host!
- */etc/init.d/lxc stop* takes a big stick to the processes in the containers

Migrate a Vserver (www) to LXC

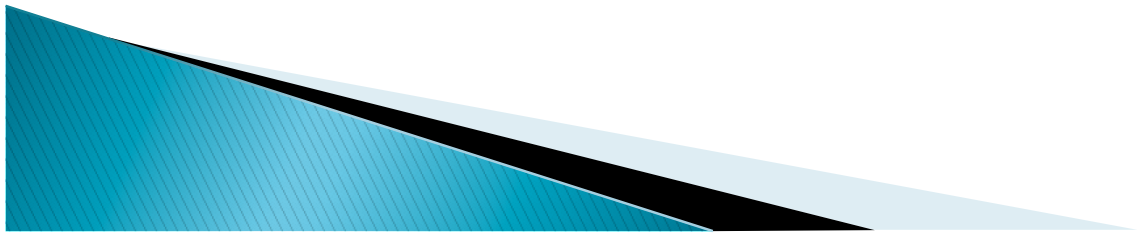
- Create and configure bridge
- Mount cgroup
- Generate a *test* LXC container with lxc-create
- *lxc-start -n test* and be sure that it works properly
- mkdir /var/lib/lxc/www
- cp /var/lib/lxc/test/config /var/lib/lxc/www/



Migrate a Vserver (www) to LXC

- Modify settings of at least:
  - lxc.network.ipv4
  - lxc.network.ipv6
  - lxc.rootfs ← point to vserver fs root
  - lxc.console
- Backup vserver
- Disable startup of udev in vserver

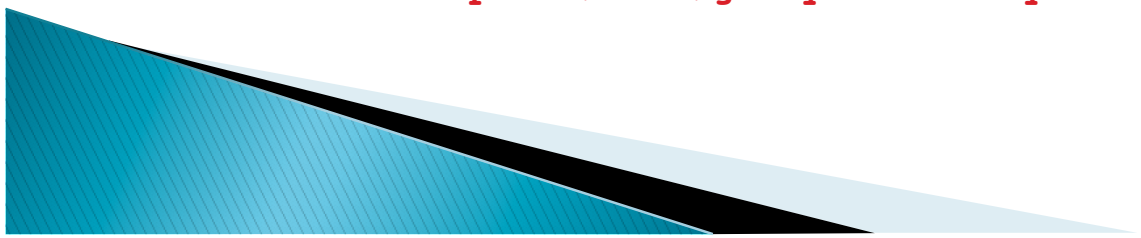
```
update-rc.d udev-mtab remove
update-rc.d udev remove
```



Migration – continued

- Copy /etc/inittab from test root to vserver root's /etc/ directory
- Modify copied inittab to start login processes on ttys

```
id:3:initdefault:
si::sysinit:/etc/init.d/rcS
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
# Normally not reached, but fallthrough in case of emergency.
z6:6:respawn:/sbin/sulogin
1:2345:respawn:/sbin/getty 38400 console
c1:12345:respawn:/sbin/getty 38400 tty1 linux
c2:12345:respawn:/sbin/getty 38400 tty2 linux
c2:12345:respawn:/sbin/getty 38400 tty3 linux
c2:12345:respawn:/sbin/getty 38400 tty4 linux
```



Issues

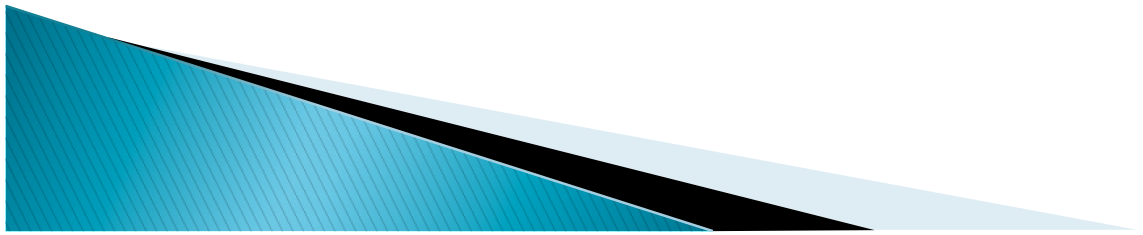
- IPv6 configuration of container's eth0 is slow causing bind9 startup to fail.
  - Added this in /etc/rc.local:

```
if [ -z "$(ps aux | fgrep –v fgrep | fgrep 'named ')" ]; then
  /etc/init.d/bind9 start
fi
```

- Container restart problems when some chrooted services are run in a container.
  - Switched from vsftpd to pure-ftpd *virtual chroot* as a workaround.
  - Still cannot disable the chroot capability in the container.

Issues – Continued

- lxc-console –n *container* always uses the container's /dev/tty1, even if there is already a shell running on that tty
  - Workaround: lxc-console –n *continer* –t *ttynum*
- lxc-stop does not perform a clean shutdown of the container but rather just kills all of its processes
  - Workaround: See http://www1.shorewall.net/pub/shorewall/contrib/lxc



Q&A