lxc and cgroups in practice

sesja linuksowa 2012

wojciech wirkijowski
wojciech /at/ wirkijowski /dot/ pl

agenda

introducion
cgroups
lxc
examples

about me

sysadmin at tieto
home page: reconlab.com
in spare time working on remashine - web app hosting platform

the idea

to build hosting platform

the problem

what if customer deploy cpu consuming task?

constrain resources

how about security?

isolate processes

how to achieve this?

full virtualization and paravirtualization - vmware, kvm, xen, etc.

full virtualization

pros: mature, proven functionality, well tested, separate kernels, different oses
cons: heavy-weight, expensive, difficult/impossible to limit some resources

lightweight virtualization (OS-level)

container concept

containers

pros: lightweight, performance gains, improved security when compared to regular hosting
cons: same OS for host and guests, lower security level when compared to regular virtualization

solutions

jail (freebsd), OpenVZ (linux), Linux-Vserver

lxc – linux containers

"LXC is the userspace control package for Linux Containers, a lightweight virtual system mechanism sometimes described as "chroot on steroids"."

limiting resources and prioritization

control groups - cgroups

control groups

out of the box in recent linux distributions
needed by lxc

why lxc and cgroups?

no kernel patching
simple management
tools are provided by most linux
distributions/packaging systems

you can focus on your goals*

*there is no time to configure and build everything

cgroups

hierarchical
children cgroups inherit certain attributes from their parents

hierarchies are attached to subsystems

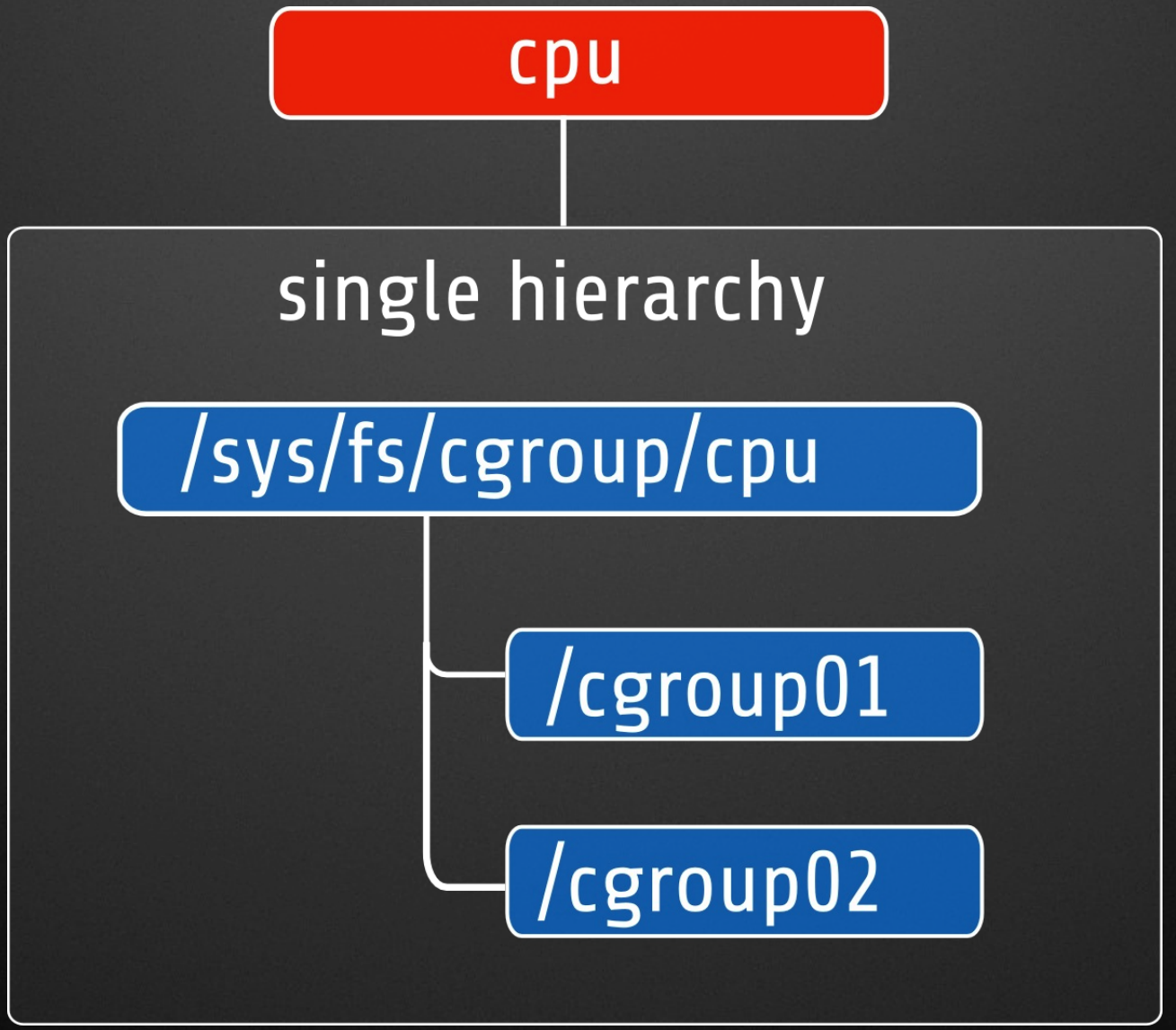
subsystem (resource controller) represents single resource such as cpu time or memory

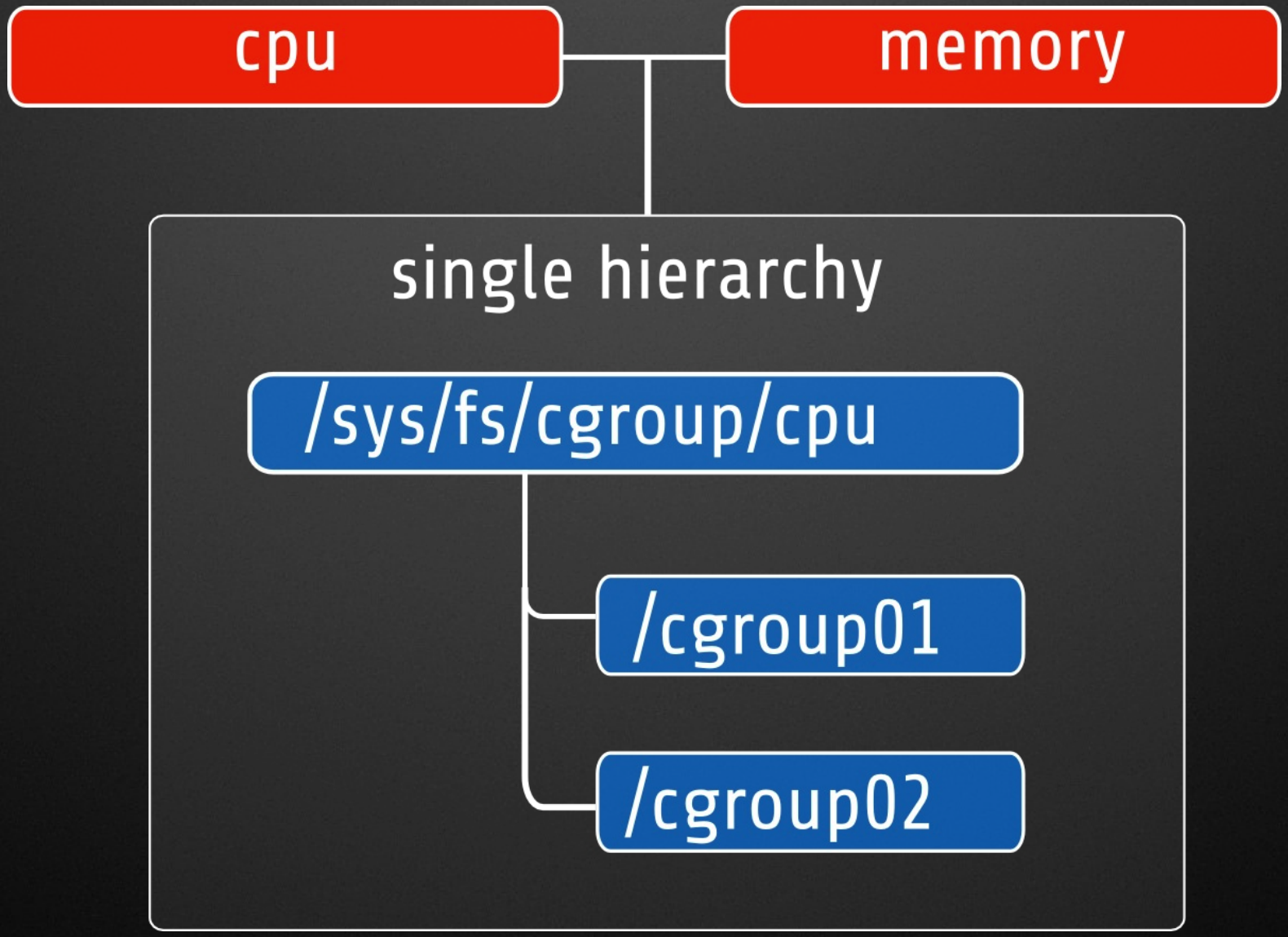
available subsystems

memory, cpu, cpuset, cpuacct, blkio, freezer, ns, net_cls

relationships

a single hierarchy can have one or more subsystems attached to it


cpu
single hierarchy
/sys/fs/cgroup/cpu
/cgroup01
/cgroup02

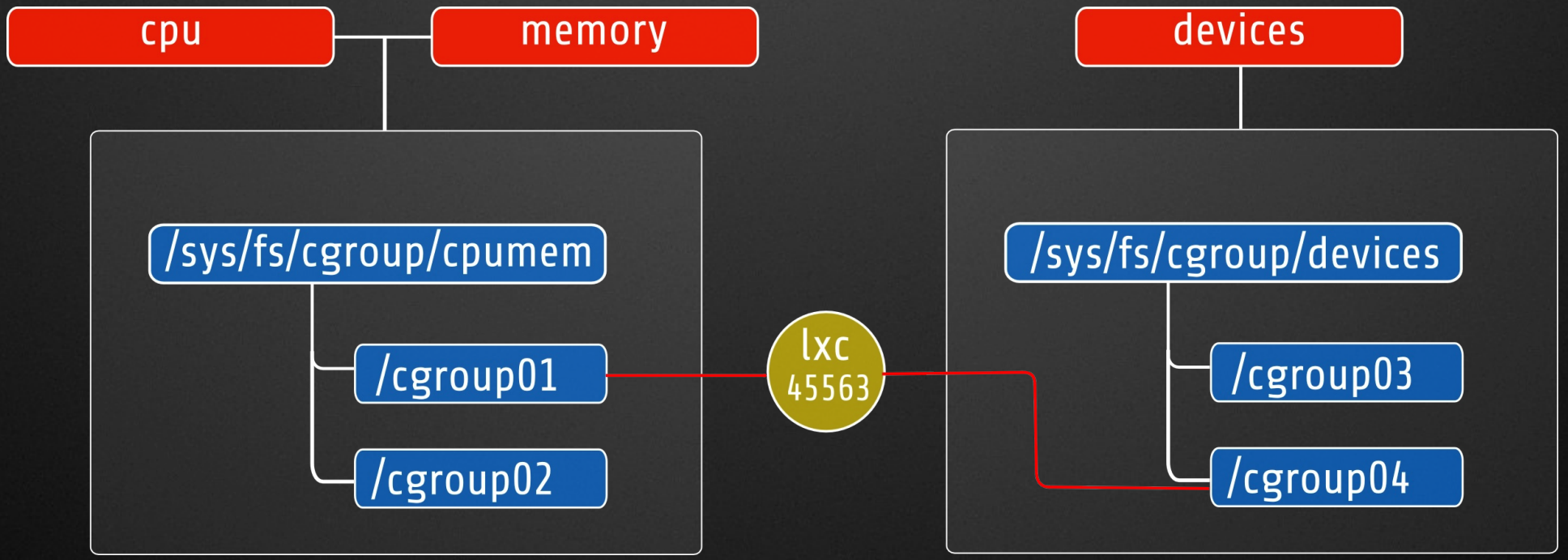

cpu
memory
single hierarchy
/sys/fs/cgroup/cpu
/cgroup01
/cgroup02

any single subsystem (such as cpu) cannot be attached to more than one hierarchy if one of those hierarchies has a different subsystem attached to it already

every hierarchy has default (root) cgroup and all processes are initially processes of that cgroup

any single task can be member of exactly one cgroup in particular hierachy

a task can be member of multiple cgroups, as long as each of those cgroups is in a different hierarchy


cpu
memory
devices
/sys/fs/cgroup/cpumem
/cgroup01
/cgroup02
lxc
45563
/sys/fs/cgroup/devices
/cgroup03
/cgroup04

forked process inherits cgroup membership from its parent

forked process can be moved to a different cgroups

managing hierarchies

/etc/cgconfig.conf
or
mount cgroup:<subsystem> -t cgroup -o
<subsystem|subsystems> /mount/point

creating cgroups

cd /mount/point/subsystem && mkdir cgroupname
or
cgcreate -t uid:gid -a uid:gid -g subsystems:path

setting parameters

```
cgset -r parameter=value path_to_cgroup
```

or

```
echo 0-1 > /cgroup/cpuset/group1/cpuset.cpus
```

moving process to a control group

cgclassify -g subsystems:path_to_cgroup pidlist
or
echo PID > /cgroup/group1/tasks

starting a process in a control group

cgexec -g subsystems:path_to_cgroup command arguments

debian

kernel param -> cgroup_enable=memory
some features are not available for vanilla debian kernel - you have to compile your own

memory controller

memory.limit_in_bytes
memory.memsw.limit_in_bytes

memory.memsw.limit_in_bytes = mem + swap

cpu controller

cpu.shares

cpu controller – since kernel 3.2

cpu.cfs_period_us
cpu.cfs_quota

limit a group to 20% of 1 CPU

```
# echo 10000 > cpu.cfs_quota_us /* quota = 10ms */
# echo 50000 > cpu.cfs_period_us /* period = 50ms */
```

cpu/→ cpu.shares = 4

cpu/cage01 → cpu.shares = 4
cpu/cage02 → cpu.shares = 4

33% + 33% + 33%
33% + 33% + … +33% + 33% + 33%

cpu/tasks → null
cpu/sysdefault→ cpu.shares = 4

cpu/cage01 → cpu.shares = 4
cpu/cage02 → cpu.shares = 4

0 * 33% + 33% + 33% + 33%

cpu/tasks → null
cpu/sysdefault→ cpu.shares = 4

cpu/cage01 → cpu.shares = 4
cpu/cage02 → cpu.shares = 2

40% + 40% + 20%

blkio controller

Proportional weight division
or
I/O throttling (Upper limit)

blkio - Proportional weight division

blkio.weight
blkio.weight_device (overrides blkio.weight)

I/O throttling (Upper limit)

blkio.throttle.{write|read}_{bps|iops}_device

echo ”major:minor value” > blkio.throttle_...

value '-1' deletes limits

linux containers

liblxc, libvirt

on debian/ubuntu you can use debootstrap to create container
or
use lxc-create script

lxc.conf

fstab

rootfs

network setup

tools for managing containers

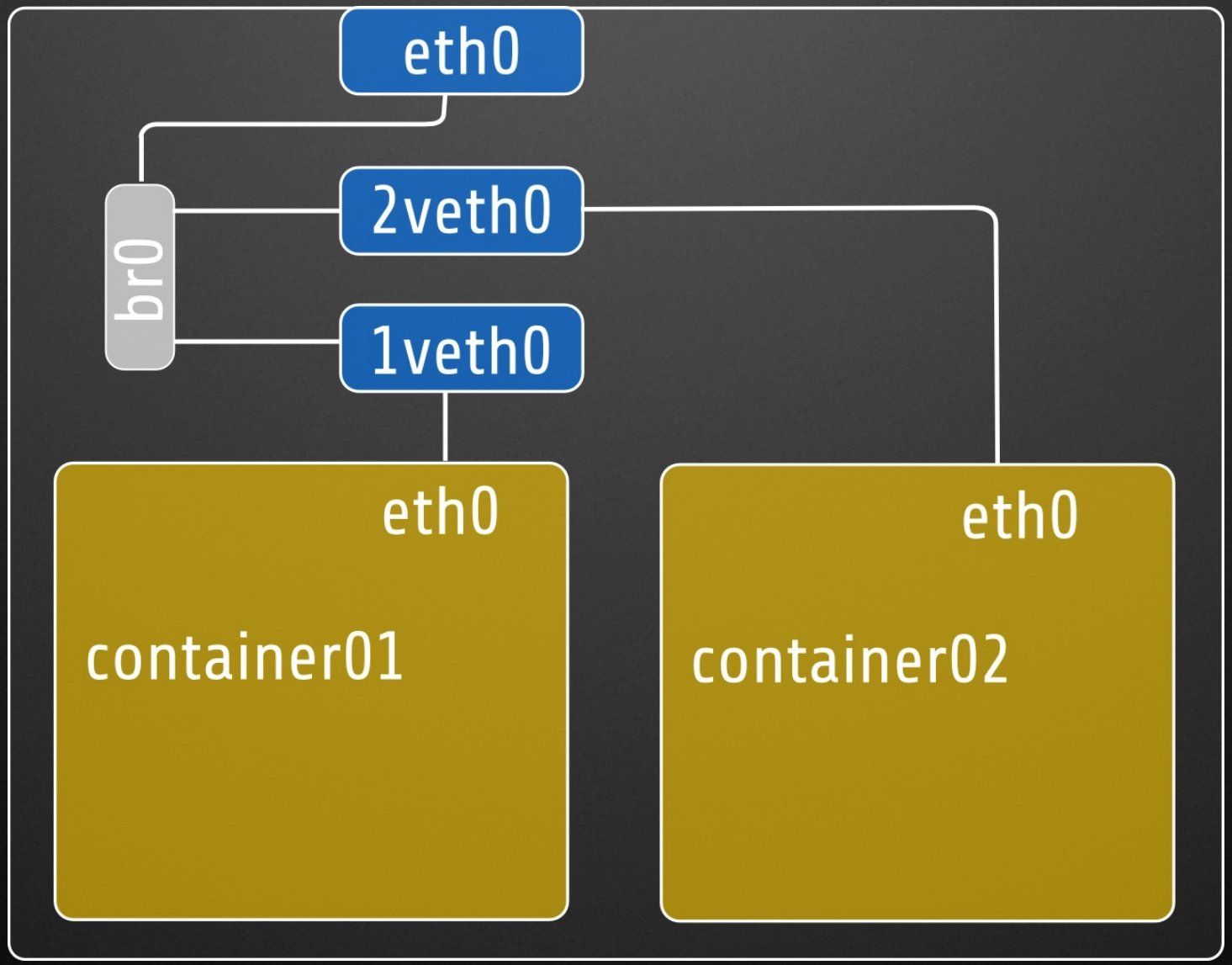
lxc-start, lxc-stop, lxc-console, lxc-info, ...

example

customer deploys their applications in a dedicated container
one container for one application

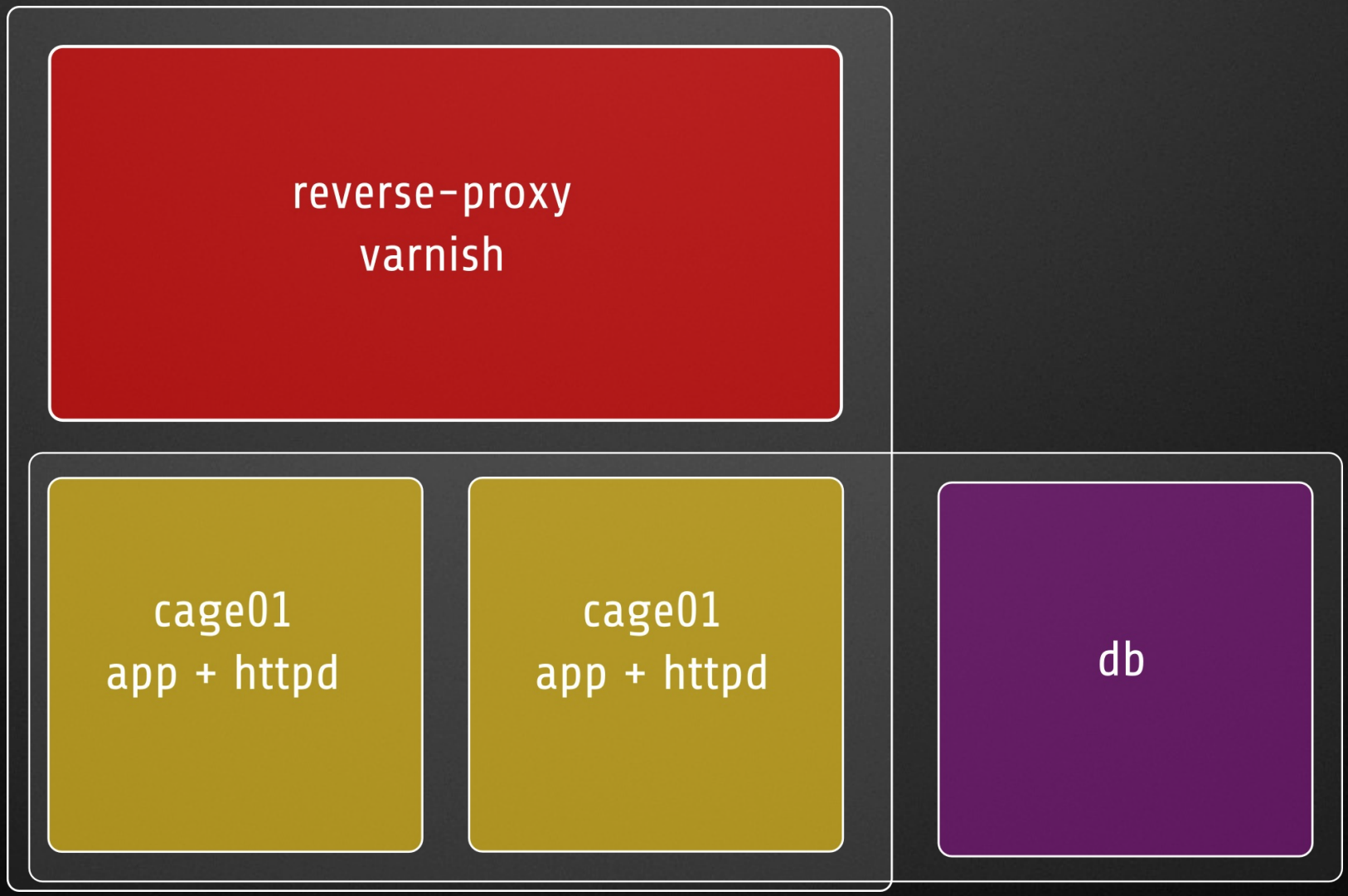
container is configured with specific environment – libraries, frameworks, application servers

container connect to private network behind reverse-proxies, load balancers, caches


eth0
2veth0
br0
1veth0
eth0
container01
eth0
container02

container has assigned resources and can be modified by user via dedicated api or webinterface

one application can’t influence other one


reverse-proxy
varnish
cage01
app + httpd
cage01
app + httpd
db

containers migration?

lxc-checkpoint
stores container state into a file
not implemented yet

orchestration

custom scripts, configuration management tools (puppet, cfengine, etc.)

summary

with cgroups you can limit and prioritize system resources
linux containers deliver OS-level virtualization
with these you can build powerful and secure hosting (much more than shared one)

http://lwn.net
http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Resource_Management_Guide/index.html
http://www.kernel.org/doc/Documentation/cgroups/
http://hydra.geht.net/tino/english/faq/debian/squeeze/cgroups/
http://lxc.teegra.net/#_keychain_behavior_when_running_in_a_private_pid_namespace
http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6.git;a=blob;f=Documentation/scheduler/sched-bwc.txt;hb=HEAD

questions?

thank you