Security of Linux containers in the cloud

Dobrica Pavlinušić
http://blog.rot13.org

FSEC, FOI Varaždin, 2012-09-20

http://www.flickr.com/photos/21644167@N04/5879497600/

Security of Linux containers in the cloud

Linux container (LXC) seems to be preferred technology for deployment of Platform as a service (PaaS) in cloud. Partly because it's easy to install on top of existing visualization platforms (KVM, VMware, VirtualBox), partly because it is lightweight solution to provide separation and process allocations between separate containers running under single kernel.

In this talk we will take a look at LXC and try to explain how to combine it with mandatory access control (MAC) mechanisms within Linux kernel to provide secure separation between different users of applications.

Presentation overview

- Containers
  - namespaces
  - cgroups resource management
- Why we need MAC to make LXC secure?
- Overview of security mechanisms
  - SELinux, pam_namespace
  - libvirt-lxc = API + LXC + SELinux
  - libvirt-sandbox, virt-sandbox-service
  - systemd
  - AppArmor
  - Smack
  - OpenVZ
- stackato PaaS with free micro cloud

Linux Namespaces

- Mount - mounting/unmounting filesystems
- UTS - hostname, domainname
- IPC - SysV message queues, semaphore/shared memory segments
- Network - IPv4/IPv6 stacks, routing, firewall, proc/net /sys/class/net directory trees, sock
- Pid - Own set of pids
- UID - Not implemented yet.

chroot is filesystem namespace

cgroups resource management

- limits
  - CPU (sched, cpu account, cpuset) - NUMA
  - Memory
  - Block I/O scheduling, limits
- Used to isolate different chunks of processes
  - Android - application isolation
  - Google Chrome - tabs isolation
  - systemd - services isolation
  - LXC - whole system (devices, network interfaces)
  - libvirt-lxc - incompatible with LXC

http://wiki.debian.org/LXC

GRUB_CMDLINE_LINUX="cgroup_enable=memory"

LXC security problems

- Root in container with /proc and /sys access
  - echo s > /proc/sysrq-trigger
  - removing capabilities from container
  - Mandatory access control (MAC) based on paths
- filesystem
  - devpts remount, securityfs, debugfs, bitfmt_misc
  - remount / on host if shared with container
- udev storm on host
  - udevadm trigger action=add

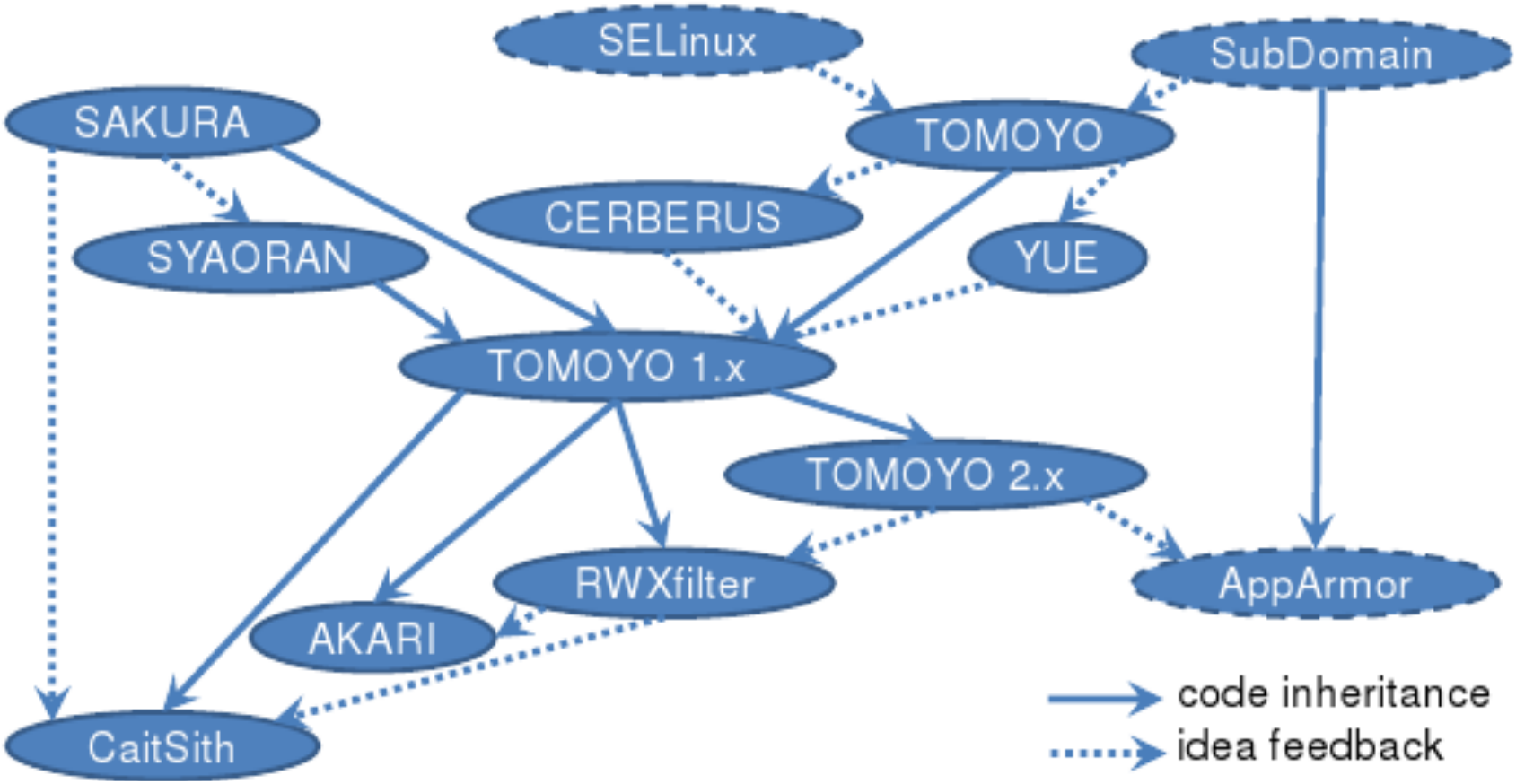
https://wiki.ubuntu.com/LxcSecurity

https://bugs.launchpad.net/ubuntu/+source/lxc/+bug/645625

How to make containers secure?

http://kernsec.org/wiki/index.php/Linux_Security_Summit_2012/Schedule

Which MAC to use?
And this are not all options!



http://kernsec.org/files/CaitSith-en.pdf
http://kernsec.org/wiki/index.php/Linux_Security_Summit_2012/Abstracts/Handa

Namespaces in RHEL/Fedora

- pam_namespace - RHEL5/Fedora 6
- SELinux sandbox - RHEL6/Fedora 8
- Systemd - Fedora 17
  - UnitFile: PrivateTmp, PrivateNetwork
- OpenShift - RHEL6
  - Pam_namespace : Private /tmp

OPENSHIFT

http://kernsec.org/wiki/index.php/Linux_Security_Summit_2012/Abstracts/Walsh
http://kernsec.org/files/securelinuxcontainers.pdf

libvirt-lxc = API + LXC + SELinux

- Container virtualization
- Boot “init” binary
- sVirt SELinux TE + MCS
- Firewall ebtables/ip[6]tables
- Host FS passthrough bind mounts
- CGroups resource control

RadHat based distributions

libvirt-sandbox

virt-sandbox-service
isolated virtual http server

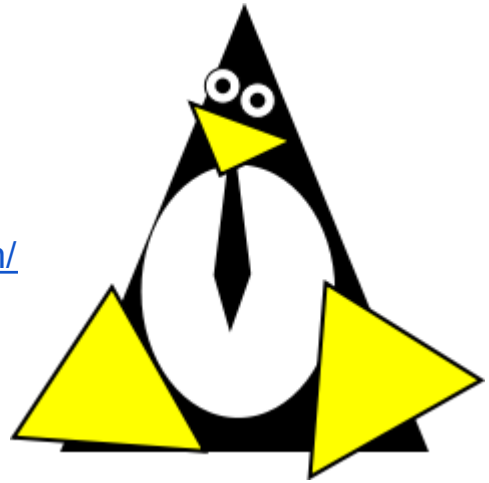
virt-sandbox-service create -C -u httpd.service apache1

- Config /etc/libvirt-sandbox/service/apache1.sandbox
- Multiple unit files allowed
- SystemD unit file
  - /etc/systemd/system/httpd@apache1.service
- Create state directories or image
  - /var/lib/libvirt/filesystem/apache1
    - Chroot type directory
  - Examines rpm payload
    - Clone - /var and /etc config
    - Share /usr
- Allocate unique MCS security label

Systemd

- systemctl start httpd@apache1.service
- systemctl reload httpd.service
  - Should trigger reload in all httpd@services
  - ReloadPropagatedFrom=httpd.service
- systemctl start httpd@.service
  - Should start all httpd services

- Systemd for the User Session
  - systemctl --user
  - http://linuxplumbers.ubicast.tv/videos/systemd-for-the-user-session/

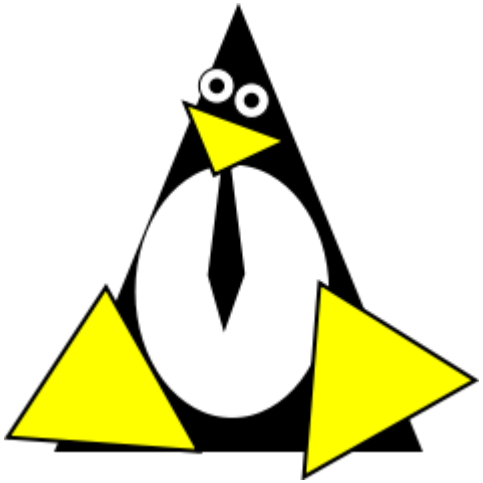


AppArmor LSM Update

- basic lxc integration
- lot of userspace work
  - porting to python 3
  - simple policy language improvements/consistency
  - policy compiler improvements
  - refactoring

http://kernsec.org/files/apparmor-update.odp

Smack Veers Mobile

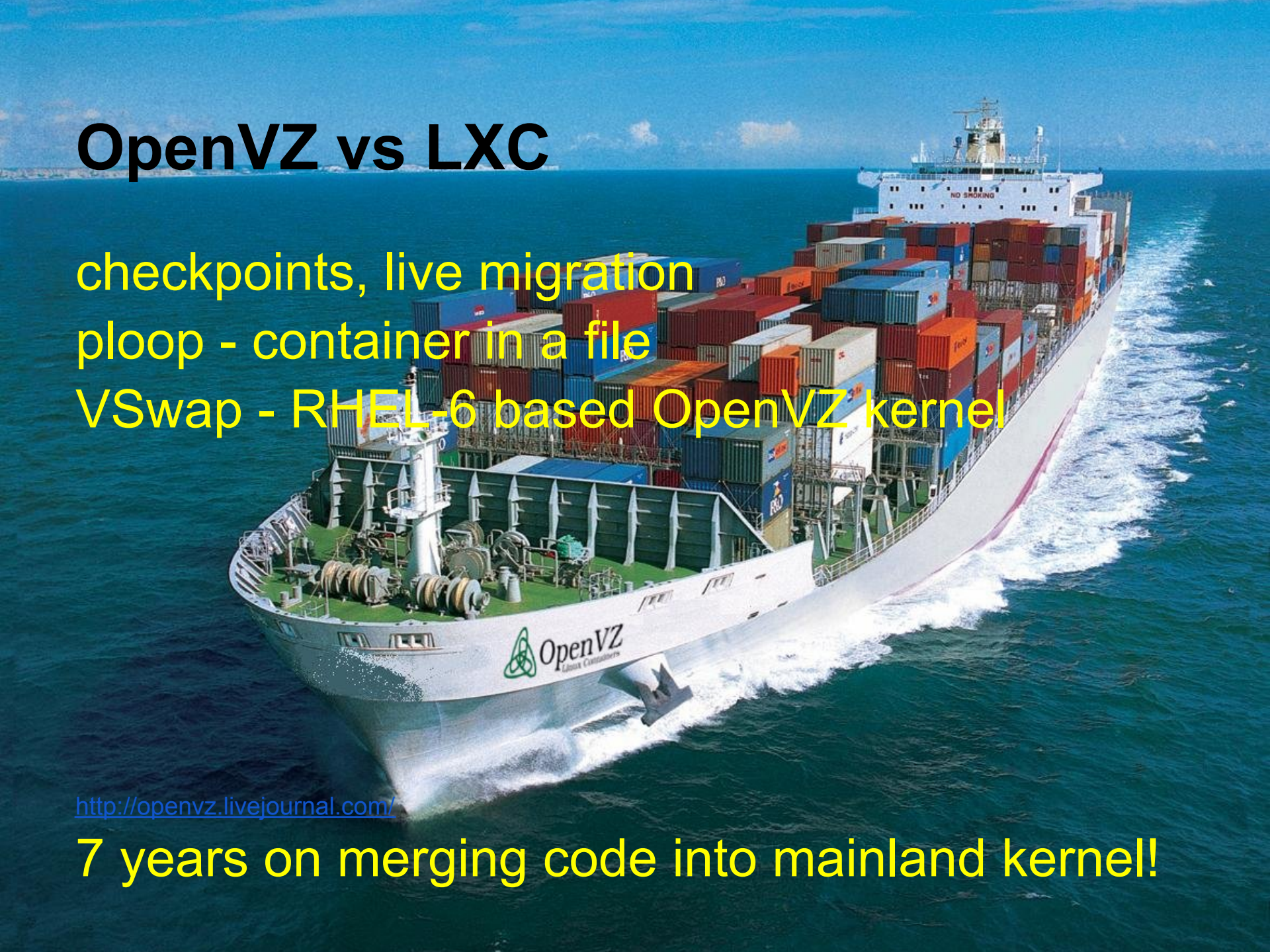


Tizen focused

```
# smack_label.py -w -r /srv/lxc/lxc1 lxc1
# echo "lxc1" > /proc/self/current/attr
# lxc-start -n lxc1
# echo "_" > /proc/self/current/attr
```

http://kernsec.org/files/SmackLinuxSecuritySummit2012.pdf
http://osdir.com/ml/lxc-chroot-linux-containers/2011-08/msg00004.html

OpenVZ vs LXC



checkpoints, live migration

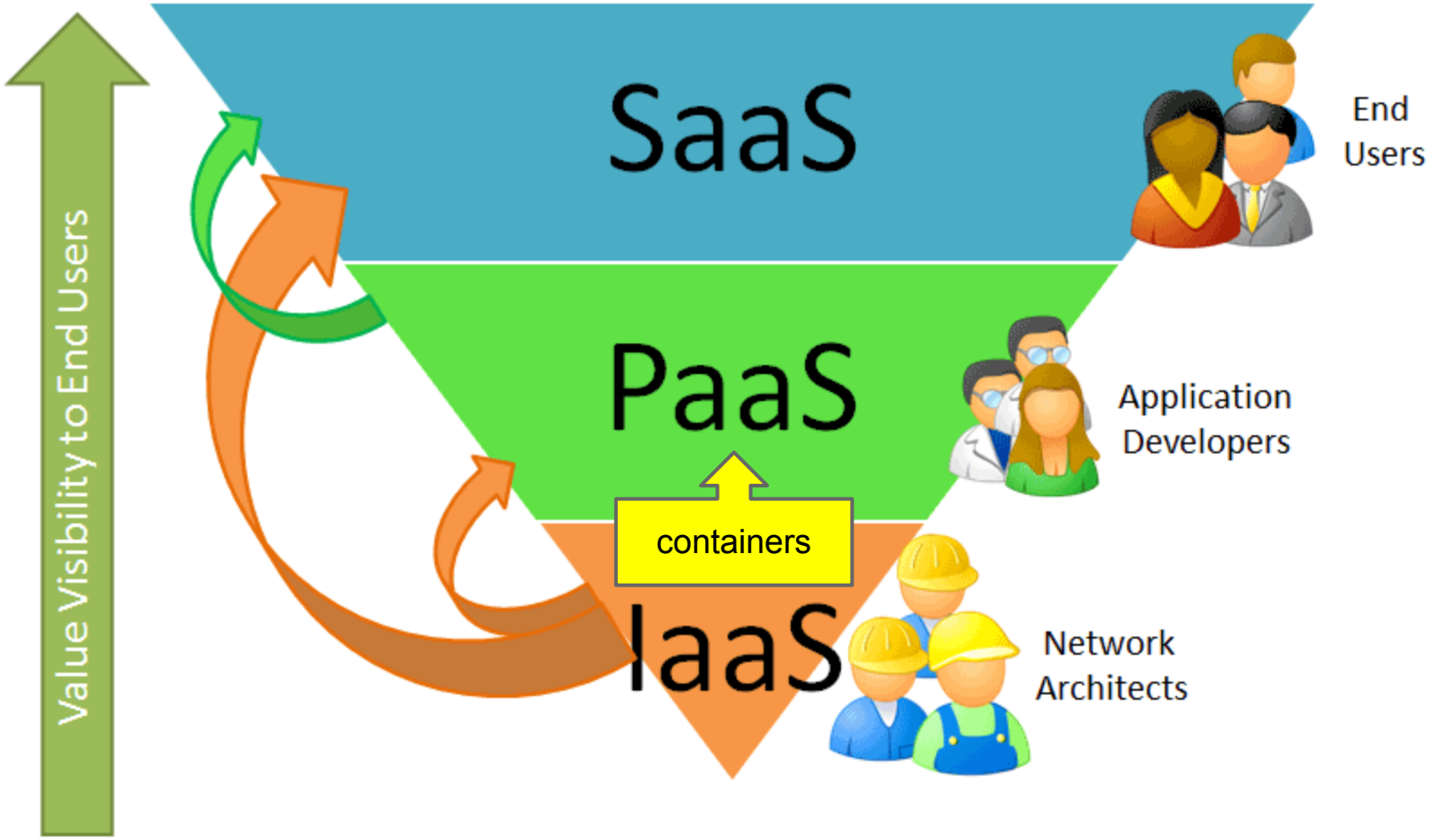
ploop - container in a file

VSwap - RHEL-6 based OpenVZ kernel

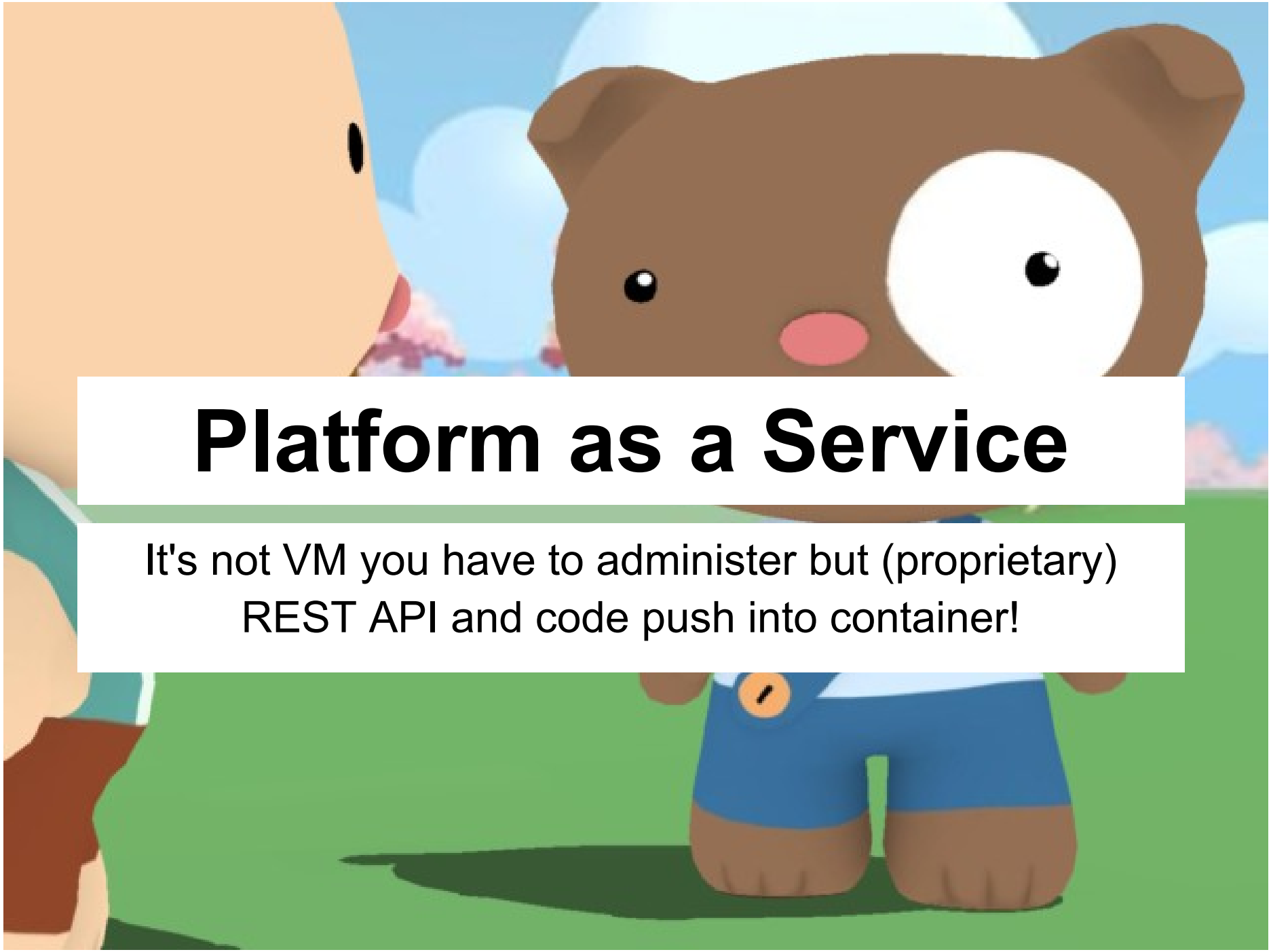
http://openvz.livejournal.com/

7 years on merging code into mainland kernel!

It's too complicated!
* as a service - buzzword bingo



Infrastructure -> Platform -> Service



Platform as a Service

It's not VM you have to administer but (proprietary) REST API and code push into container!

Stackato PaaS

- appliance PaaS, Ubuntu lucid VM
- deploy through web interface from git repo
- Ingy döt Net - Stackato Bringing the Cloud Back Home http://youtu.be/fiGurzIzfvY
- LXC (using liblxc directly)
- aufs for re-use of base installation
- collectd for stats (nothing about containers!)
- no-pay Micro Cloud download
  - http://www.activestate.com/stackato/download_vm
- **Commercial licence required for public deployment!**


Questions?
Are we there yet?
containers suffers from too many different incompoatible deployment options
http://bit.ly/fsec2012-lxc
http://blog.rot13.org
http://www.flickr.com/photos/bcnbits/2859519269/
http://www.flickr.com/photos/bcnbits

Virtualizations vs Containers

different kernels/OS	single kernel
emulation of devices	**acl+syscall**
many fs caches	**single fs cache**
limits per machine	**limits per process**
legacy consolidation	service deployment