Using Containers Without Risking Your *aas

Serge Hallyn, Scott Moser

Canonical, Inc

serge.hallyn@ubuntu.com, scott.moser@canonical.com

November 3, 2014

Overview

Linux Containers?

- operating system-level virtualization method for running multiple isolated Linux systems (containers) on a single control host.
- "chroot on steroids"
- "it's like bsd jails" (or solaris zones)
- from the inside looks like a vm
- from the outside looks like processes

Containers prior to user namespaces

Namespaces

- *id* $\rightarrow$ *resource* mapping
  - Prevent resource access by not providing a handle
  - i.e. pid 1 is not global init
  - `/etc/shadow` not accessible
- Many leaks (`/proc/pid/fd/N`)

Control groups

1. Resource limits and accounting
2. Limit device access
3. If root, re-mount cgroups and change/escape limits.

Capabilities bounding set

1. Limit privs of root in container
2. Root still owns most host files
3. http://www.sevagas.com/IMG/pdf/exploiting_capabilities_the_dark_side.pdf
4. Prevents useful things like tmpfs mounts

LSMs

1. Paper over the (huge) remaining holes
2. i.e. prevent `/proc/sys/*` writing, etc
3. "Safe from accidental damage by container root"
4. People always want unsafe exceptions
5. Lack of policy nesting limits use *in* containers

Seccomp

1. Prevent use of some syscalls
2. Reduce exposed kernel surface
3. Hard to do generally

1. Nevertheless
    1. Root in container is still root on host
    2. Owns many system files as uid 0
    3. Any leak = game over
    4. Answer: "Wait for user namespaces"
2. User Namespace
    1. First patch in 2007
    2. Separate accounting only
    3. No isolation
    4. Final design upstreamed in December 2012

User namespaces

Goals

1. Uid separation
   1. c1.500 != c2.500
   2. Separate access controls (kill, open, etc)
   3. Separate accounting, limits
2. Container root privileged over container
   1. uids
   2. network
   3. etc
3. Container root has no privilege outside of container
   1. Root in container as safe as unpriv user on host
   2. Safe for use by untrusted users
4. Able to be nested

User namespace design

1. Uids map 1-1 to kuids
    1. Translated at kernel-user boundary
    2. Default mapping 0-4294967295:0-4294967295
    3. Unmapped userids show up as -1, has 'o' perms
    4. Unpriv user can only map own host uid
2. Other namespaces owned by a user ns
    1. Root in ns has full privilege over what it owns

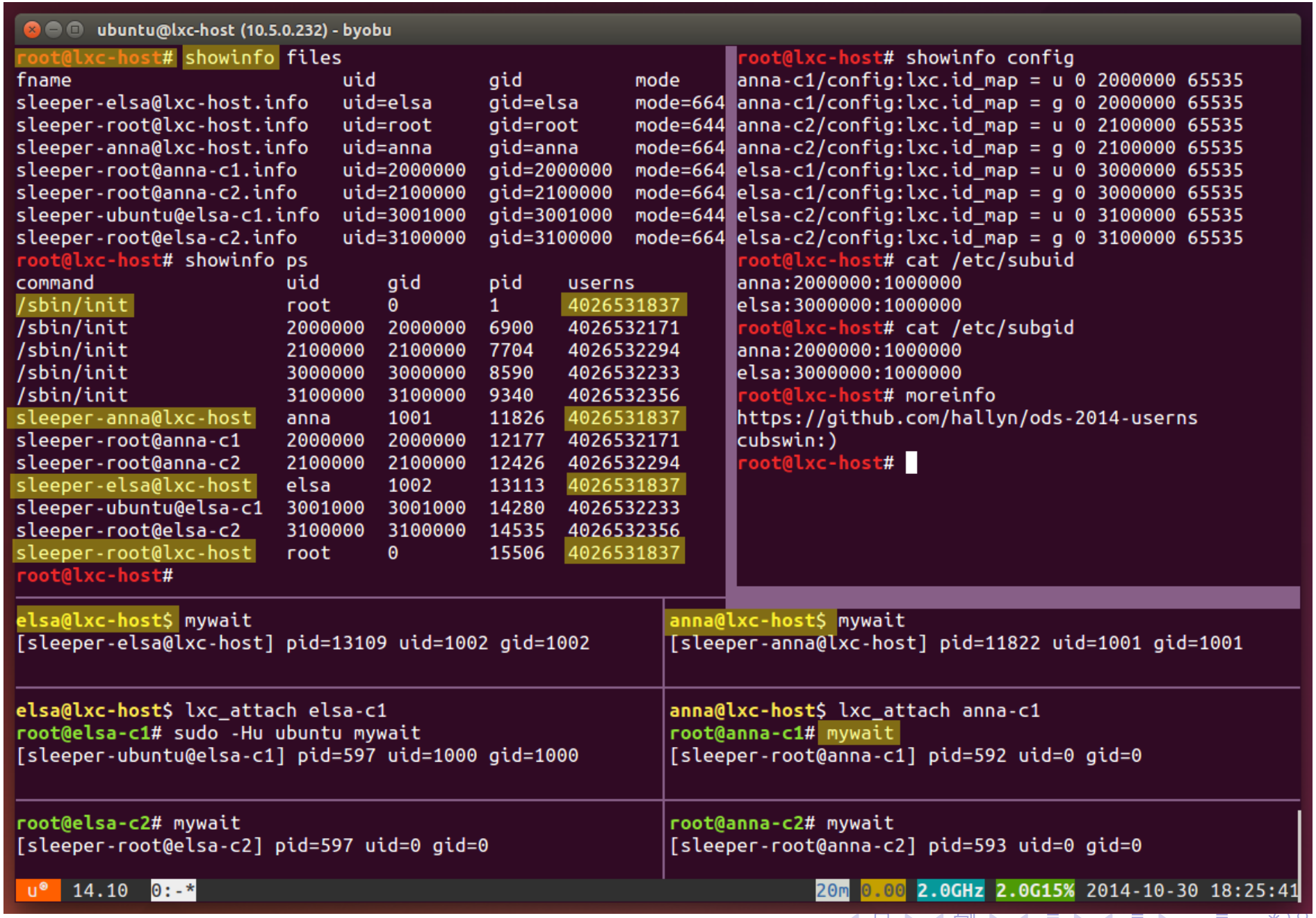
Uid delegation

1. Root delegates *subuids* to users
   1. `/etc/subuid` and `/etc/subgid`: serge:100000:65536
   2. Set using `usermod`: usermod -v 100000-200000 -w 100000-200000 serge
2. Setuid-root programs write to `/proc/self/{ug}id_map`
3. Each user may be delegated a set of subuids and subgids

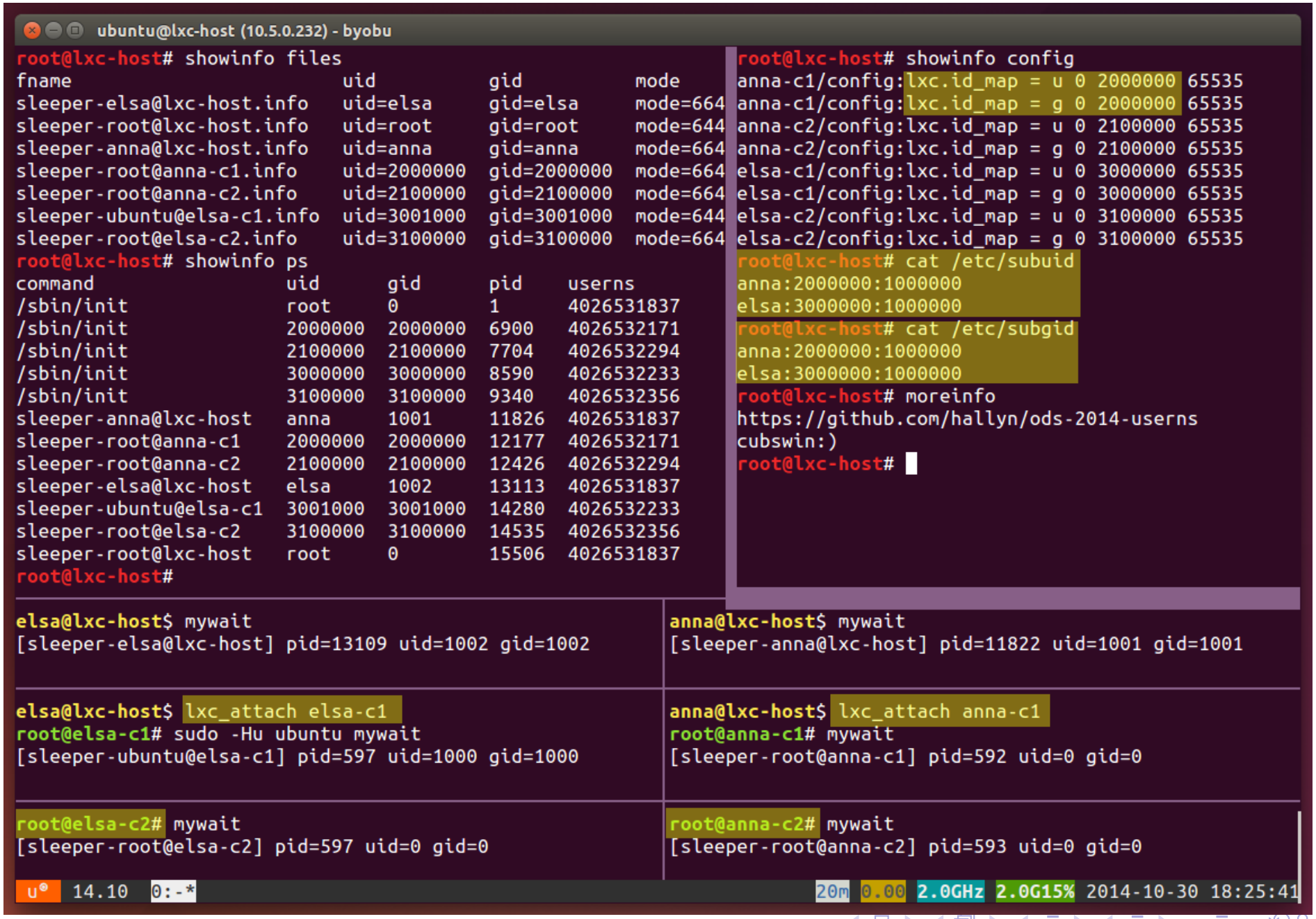
Demo Time [sort of].

1. Ubuntu 14.10 instance with hostname 'lxc-host'.
2. 2 users (elsa, anna) are each configured to run lxc unprivileged.
3. 'showinfo': simple shell filter to 'find' or 'ps' or 'grep'.
4. 'mywait': Very Exciting. Run it, it prints its pid, uid, gid. Then creates a file named 'sleeper-user@hostname' and sleeps forever. copied into each container's /usr/local/bin.

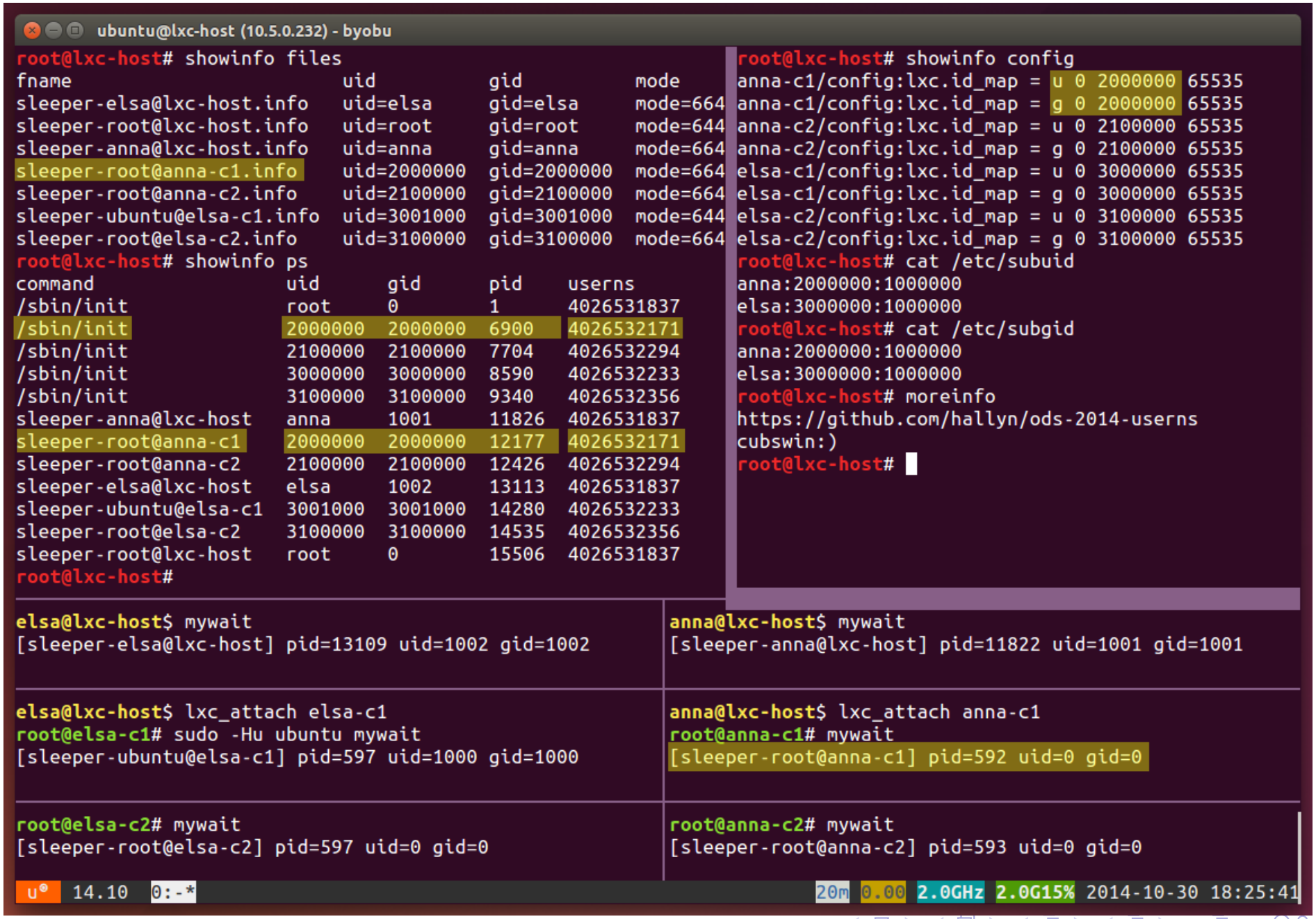
Host Processes / Users.



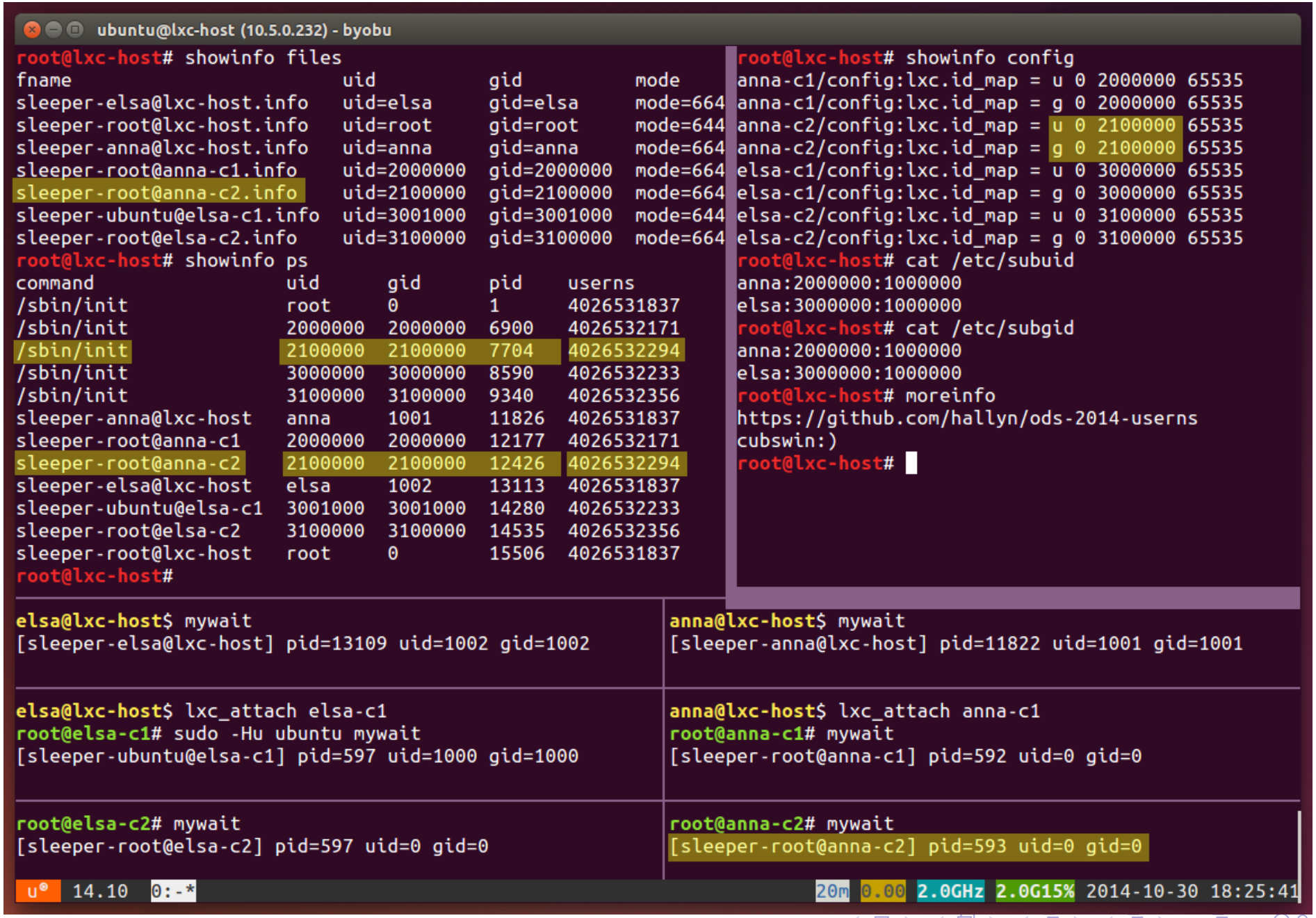
LXC Containers and Configuration



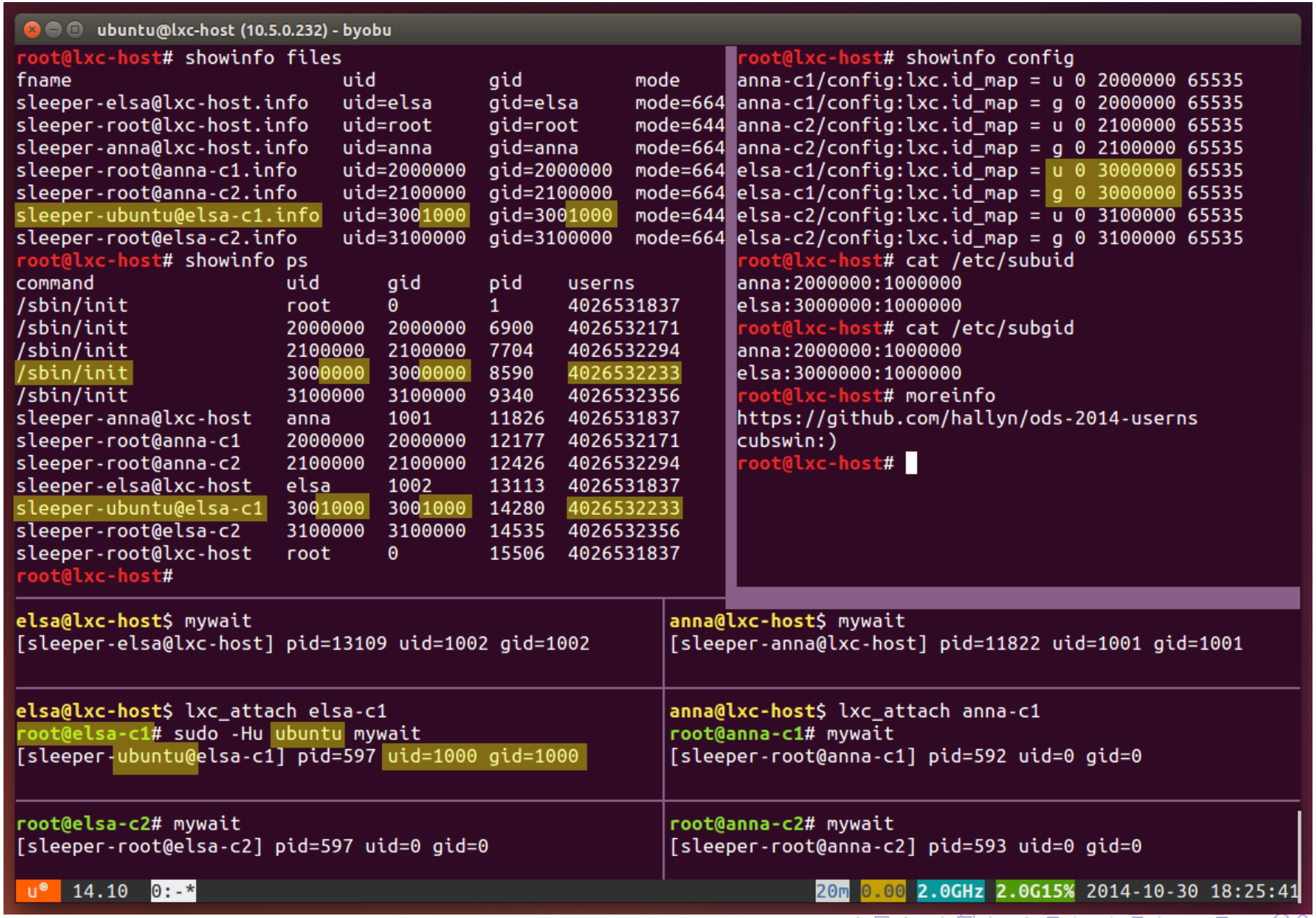
Anna's containers: **anna-c1**, anna-c2



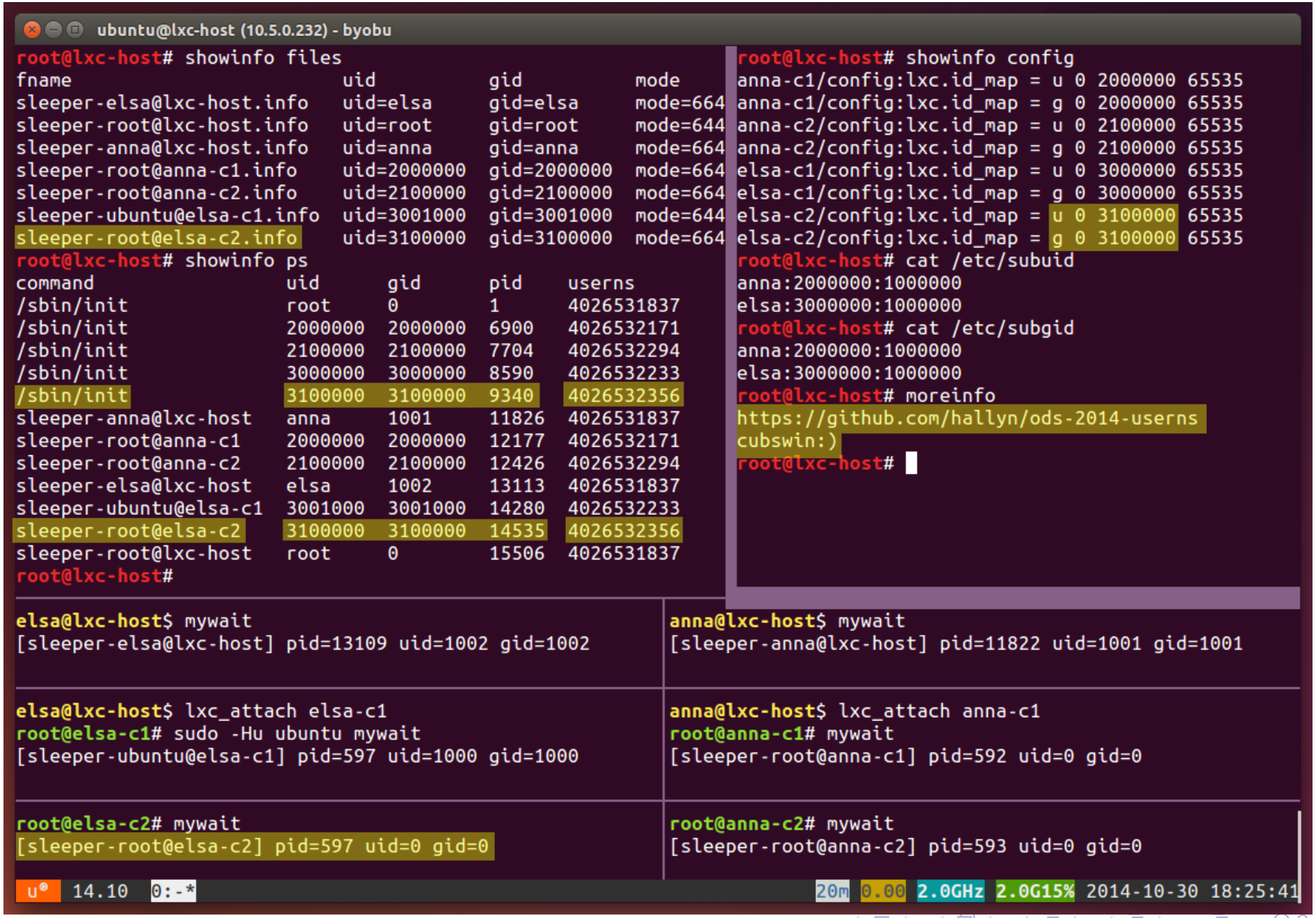
Anna's containers: anna-c1, **anna-c2**



Elsa's Containers: **elsa-c1**, elsa-c2



Elsa's Containers: elsa-c1, **elsa-c2**



How safe is this?

- users including root in container are *unpriviledged* users, but are local users.
- Linux Kernel CVEs
  - As of 2014-11-02, per `http://cvedetails.com`
  - 101 Total
  - 14 Priviledge Gain (some of these exploitable from kvm)

Use Cases

- run existing service in container: straightforward win
- move kvm / vm workload into container: performance win
- provide trusted users "root" access: yes
- provide untrusted users "root" access: maybe (if you provide shell access, same vulnerability level)

Distribution / Toolkit Support

- Linux kernel: added in 3.8, necessary improvements in 3.10
- Distros:
  - Ubuntu 14.04+
  - Red Hat Enterprise Linux: 7 (3.10.0-123 kernel)
  - SUSE Linux Enterprise Server: 12 (3.12.28-4.6)
- Tools:
  - lxc: in 1.0 improvements in 1.1
  - libvirt: yes, current versions
  - nova libvirt-lxc driver: Juno
  - nova-compute-flex: yes
  - Parallels: early 2015

More Info

- These Slides: `https://github.com/hallyn/ods-2014-userns`
- Serge Hallyn <serge.hallyn@canonical.com> [freenode: 'hallyn']
- Scott Moser <scott.moser@canonical.com> [freenode: 'smoser']
- 'Namespaces In Operation' [lwn.net]
- 10 part series on LXC 1.0 [stgraber.net]
- LWN.net Secure Linux containers [lwn.net]