Linux Containers

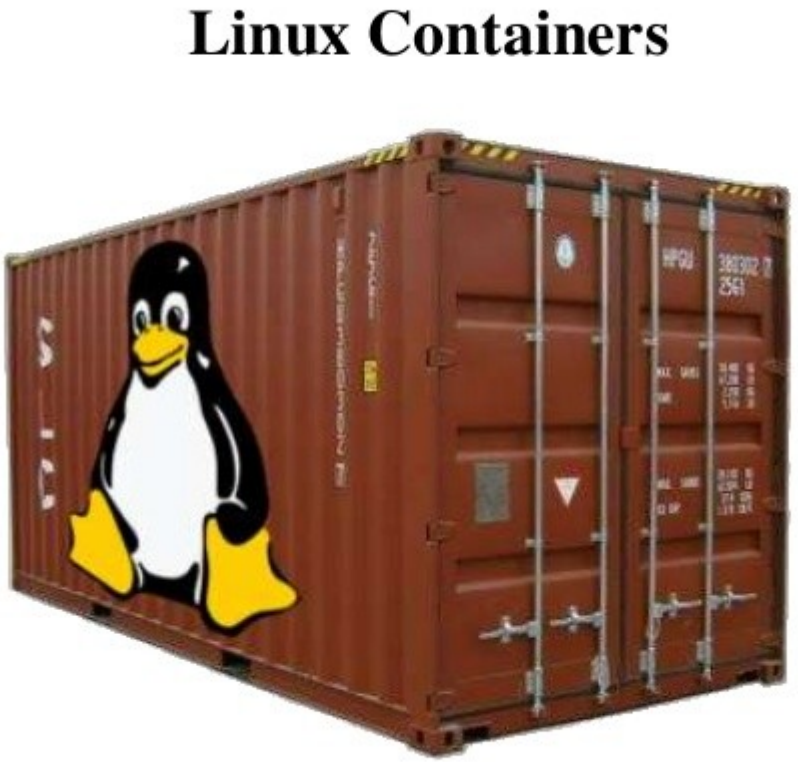
Algiers Tech Meetup
2016

Djalal Harouni

Linux Containers

- Introduction
- systemd
- Linux Containers
- Docker
- systemd-nspawn
- Linux Namespaces and cgroups
- Conclusion

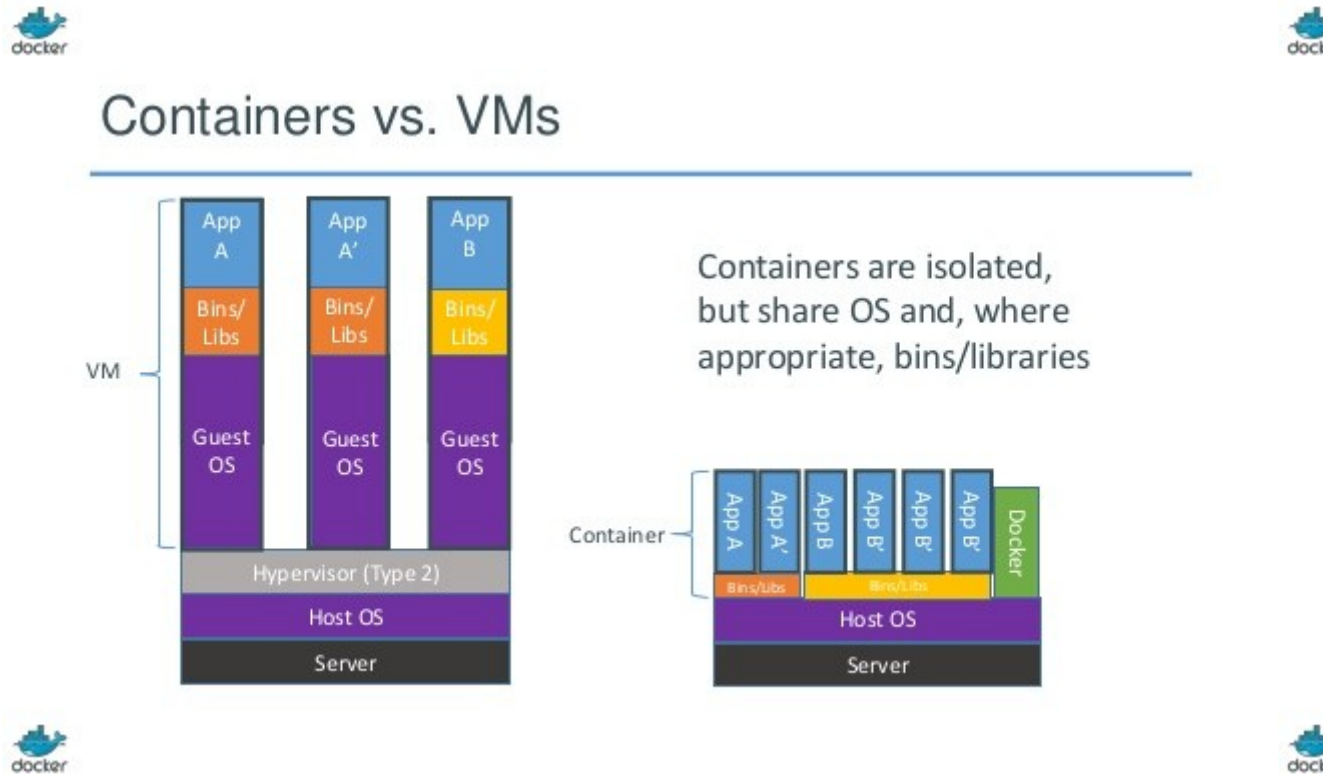
Introduction



Linux Containers [1]

Introduction

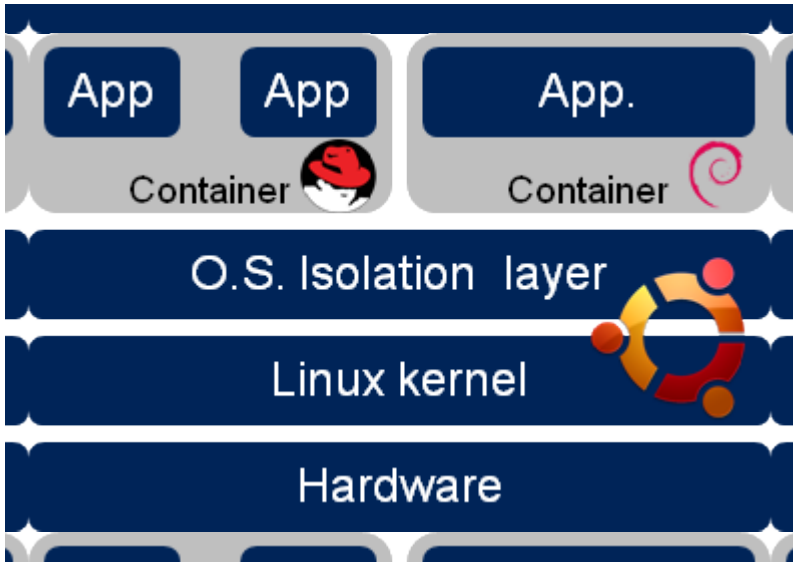
Introduction

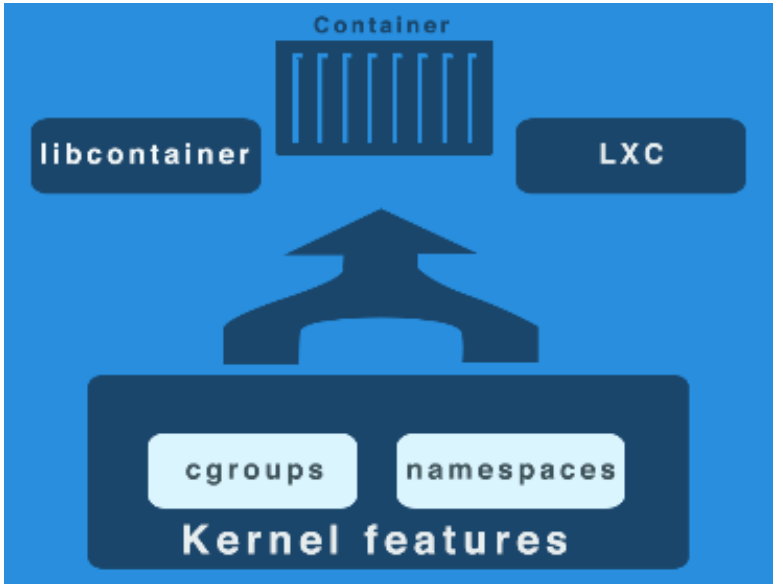


Containers vs. VMs [2]

Introduction

Linux Container API == Isolation layer
==
Namespaces + cgroups + ...





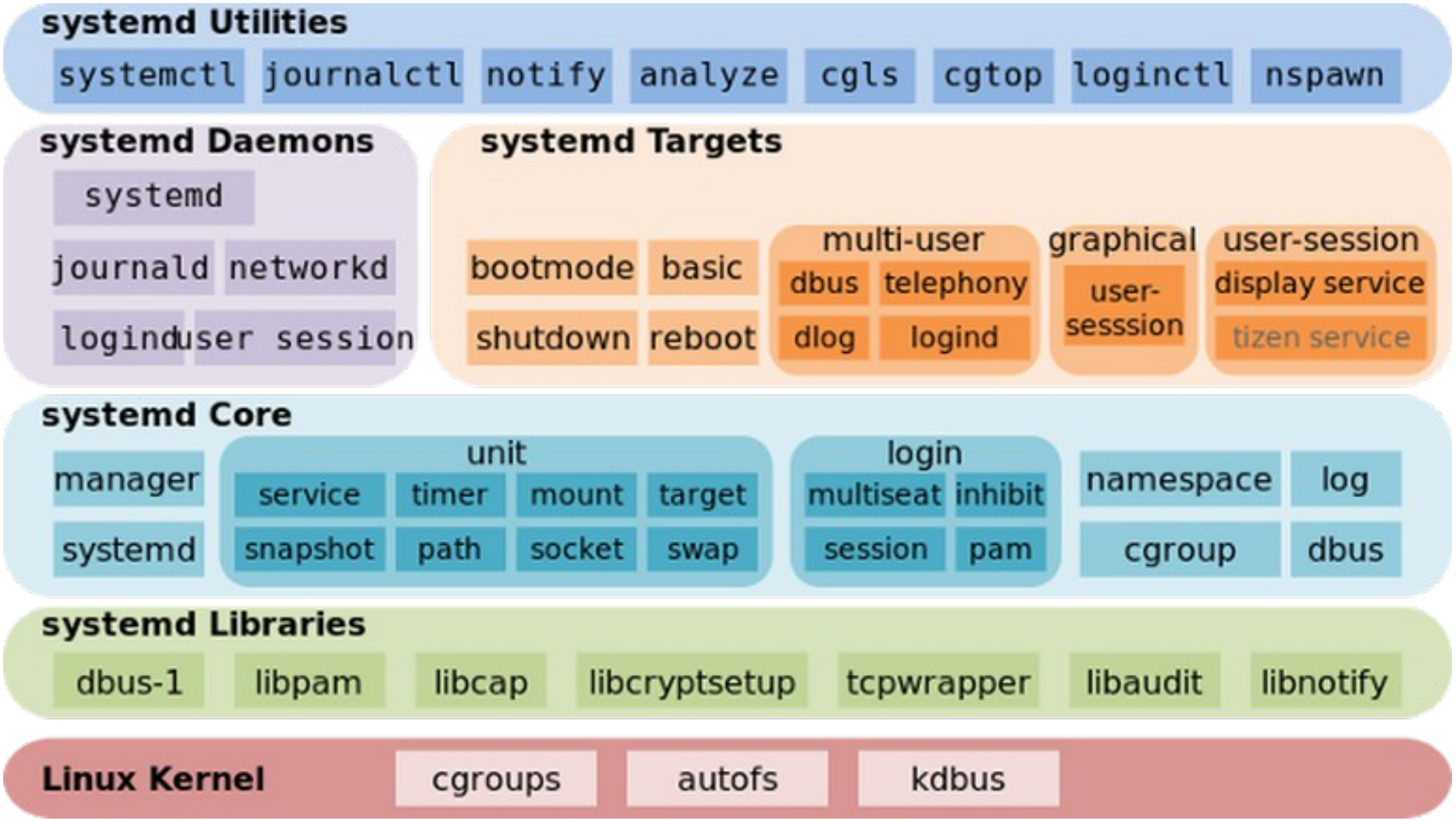
systemd

- systemd is a system manager
- systemd is a service manager
  - => systemd in this case is just PID 1
  - => uses units, mount, device units
  - => socket activation

- systemd is a project: systemd, systemd-journald, systemd-logind, systemd-machined, …
- systemdctl, journalctl, loginctl, machinectl, …

Demo unit, systemctl, journalctl, loginctl, systemd-cgls

systemd


systemd Utilities
systemctl
journalctl
notify
analyze
cgls
cgtop
loginctl
nspawn
systemd Daemons
systemd
journald
networkd
logind
user session
systemd Targets
bootmode
basic
shutdown
reboot
multi-user
dbus
telephony
dlog
logind
graphical
user-sesssion
user-session
display service
tizen service
systemd Core
manager
systemd
unit
service
timer
mount
target
snapshot
path
socket
swap
login
multiseat
inhibit
session
pam
namespace
log
cgroup
dbus
systemd Libraries
dbus-1
libpam
libcap
libcryptsetup
tcpwrapper
libaudit
libnotify
Linux Kernel
cgroups
autofs
kdbus

Security

- PrivateTmp=yes|no
- PrivateDevices=yes|no “access disk /dev/sda but not /dev/sdb”
- PrivateNetwork=yes|no
- ProtectSystem=yes|no|full
- ProtectHome=yes|no|read-only
- ReadOnlyDirectories=
- NoNewPrivileges= “Can not gain privileges anymore”
- CapabilityBoundingSet= “no **CAP_SYS_BOOT, CAP_SYS_MODULE**”
- SystemCallArchitectures=x86_64
- SystemCallFilter=
- User=X , Group=Y , SupplementaryGroups=Z
- RootDirectory= “old chroot”
- …

Linux Containers

Docker, Rkt, LXC, libvirt-lxc, OpenVZ, . . .
systemd-nspawn + systemd-machined +
systemd-importd...

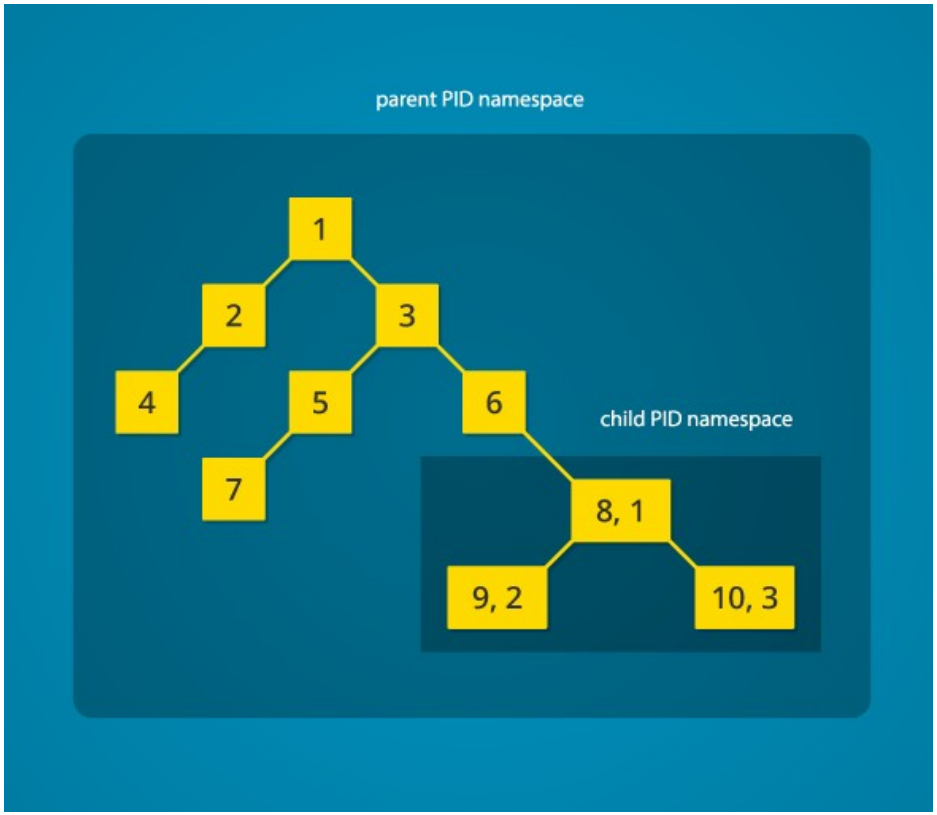
Linux Containers

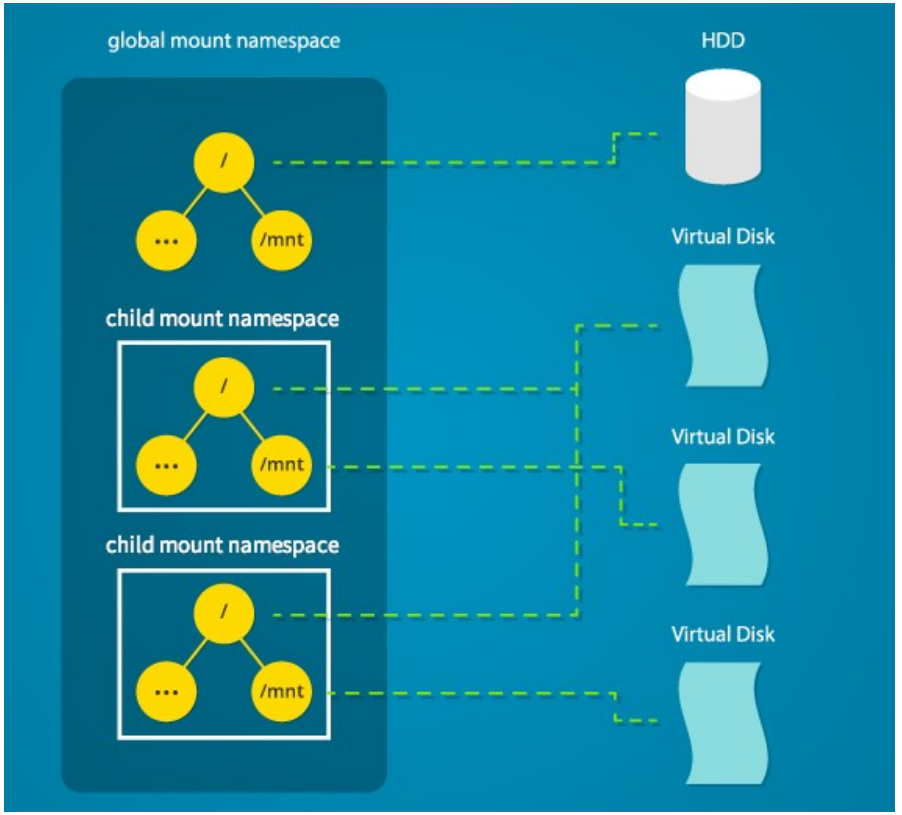
Containers are part of Linux now.

- Inspiration: Solaris Zones
- OS running inside the container similar to OS outside of the container
- Future package managers (disk space)

Linux Containers

Kernel API: **Namespaces** and **cgroups**

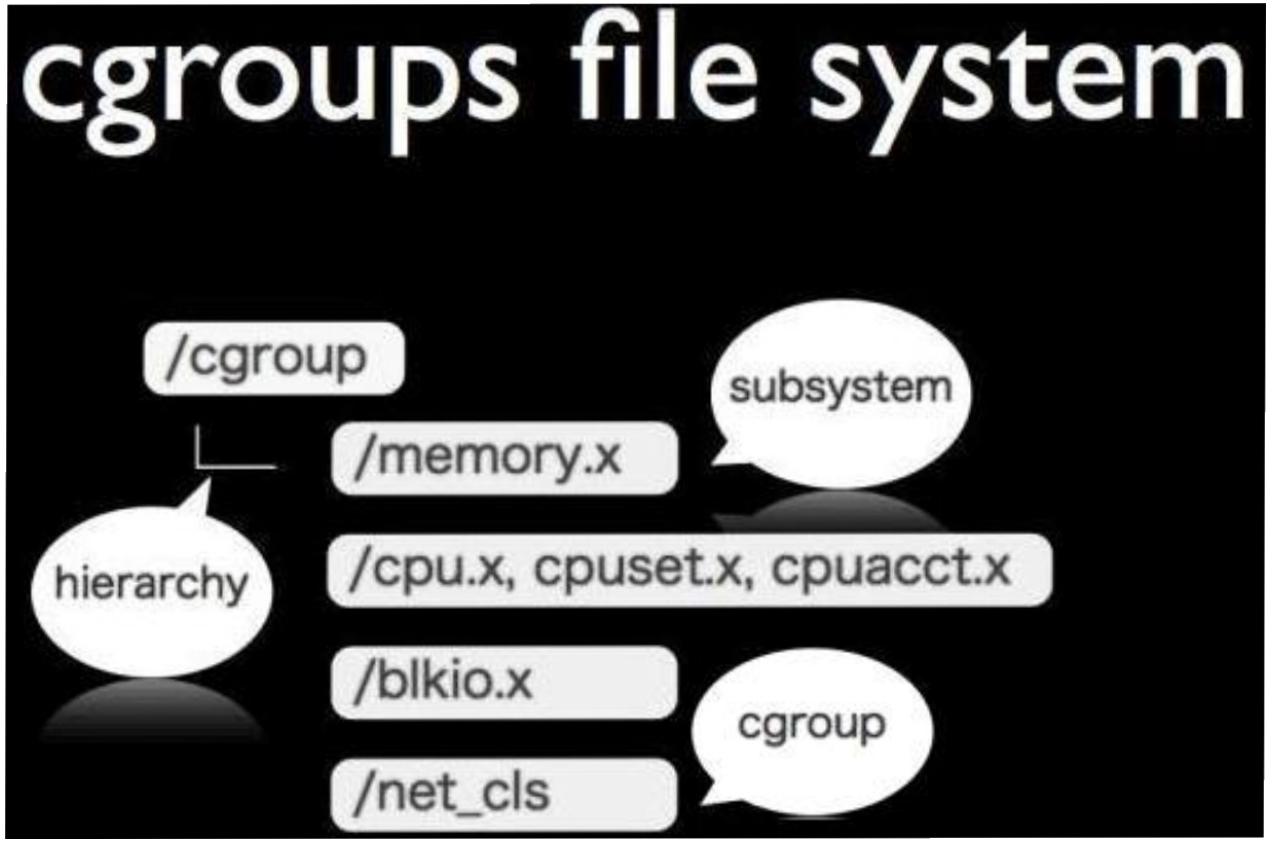




Pid and mount namespaces [3]

Linux Containers

Kernel API: **Namespaces** and **cgroups**



Docker

- Docker made Linux containers easy to use.
- Allows to create and share container images
- Docker a daemon to manage and talk to containers

Docker

Docker hub to download already images

- **docker run -it debian**
- **docker run -it nginx**
- **docker run -p 8080:80 -d -i -t nginx**

  docker attach

  docker ps , docker logs

  docker commit <id> newname

Systemd-nspawn

Systemd-nspawn: lightweight container

- Boot from directory:

  systemd-nspawn -bD ~/fedora-tree/ 3

- Boot from image:

  systemd-nspawn -M Fedora-Cloud-Base

Systemd-nspawn

- **machinectl + systemd-machined**
  Register containers or VMs with machined

- **Systemd-networkd: minimal**

- **machinectl pull-raw url**
- **machinectl export-tar fedora myfedora.tar.xz**

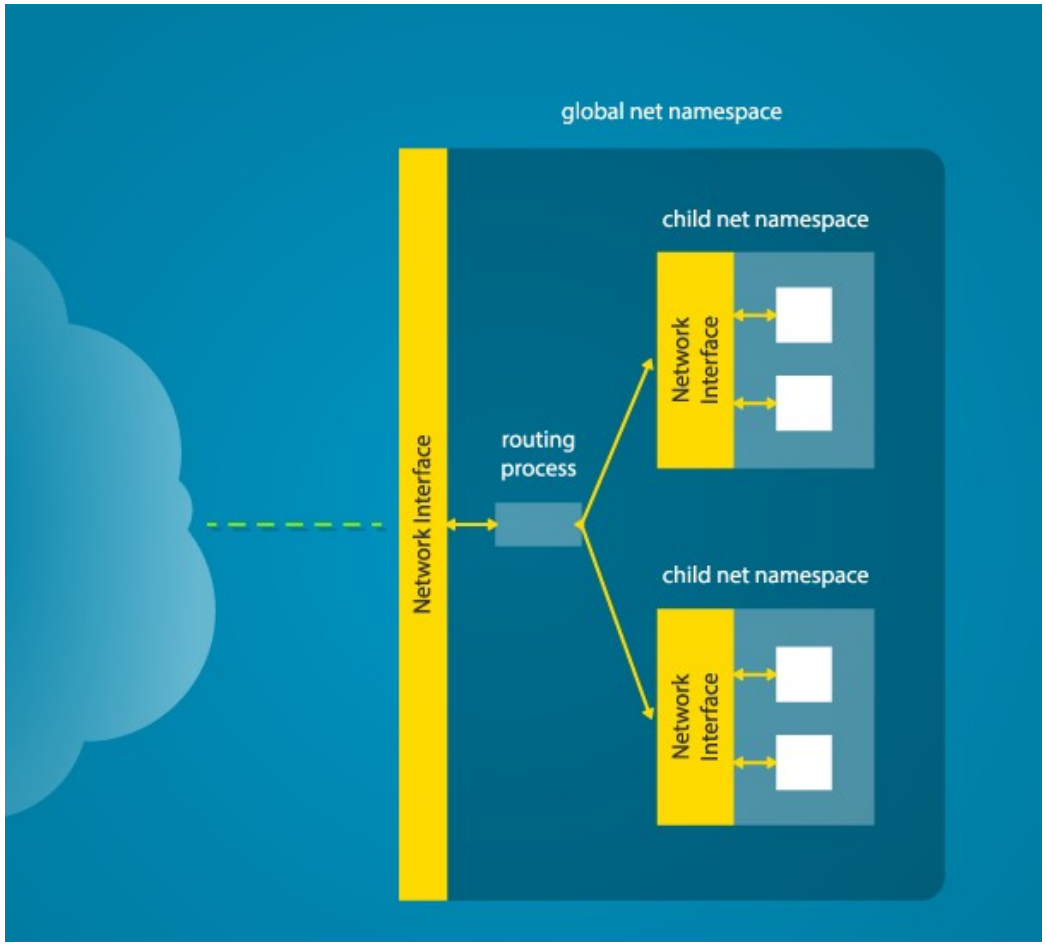
  Demo: machinectl, start and stop container

Linux Namespaces

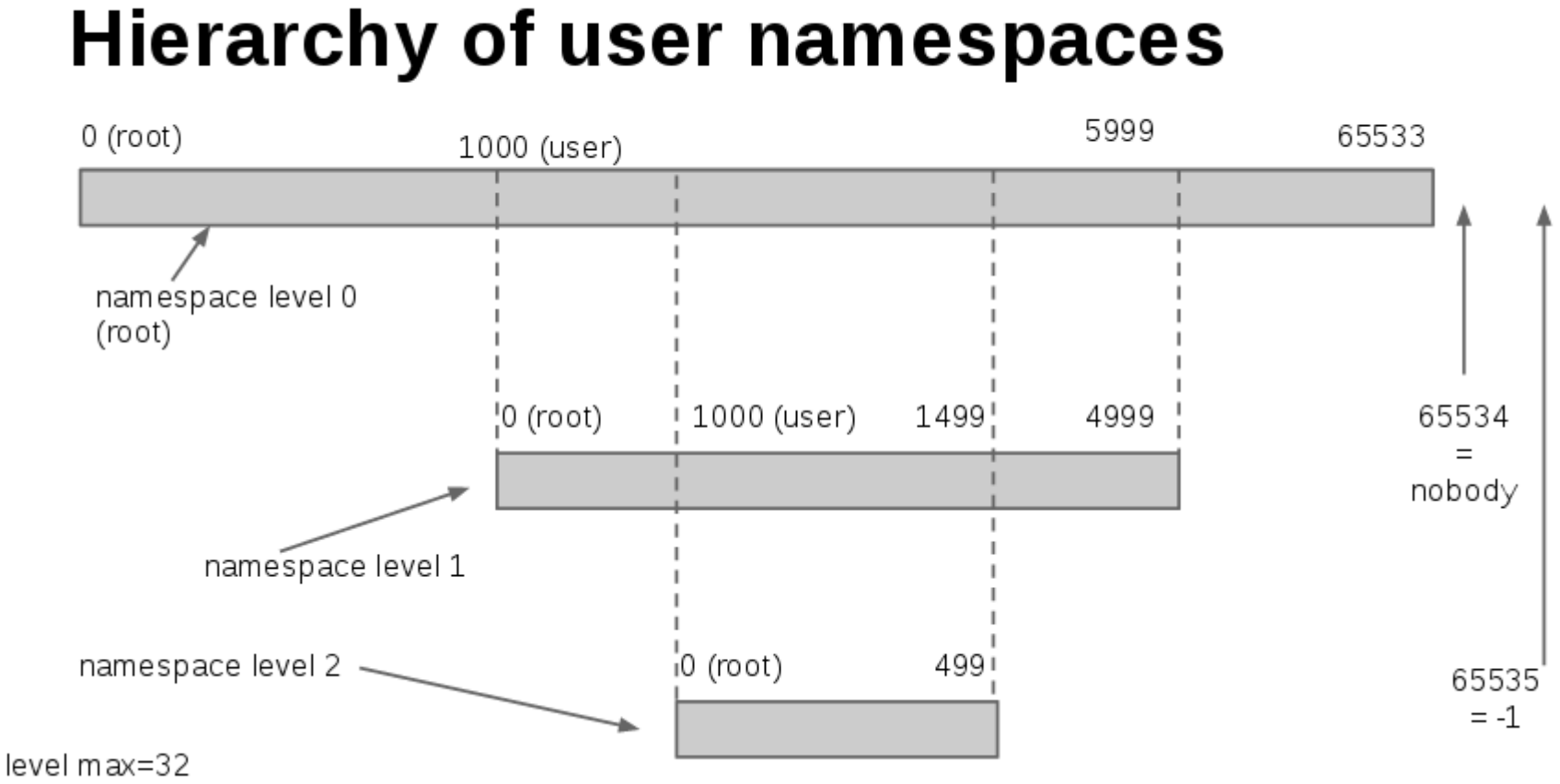
Linux Namespaces API used by containers:

- **UTS (Unix Timesharing System) namespace**
- **Mount namespace**
- **Pid namespace**
- **Network namespace**
- **User namespace**

Network namespace



User namespace



Linux cgroups

- /sys/fs/cgroup/
- ├─ cpu
- ├─ devices
- ├─ freezer
- ├─ memory
- ├─ ...
- └─ systemd

Demo: systemd-cgls, limit resources

Linux file systems

Copy-on-write

- **Union mount: Overlay filesystem**
  lower and upper layers
- **Snapshotting: btrfs**

Conclusion

- Linux Containers: next package managers ?
- systemd –-user session, seats ...
- Wayland and GUI apps sandboxed and running inside containers by default.
- Every one with his own cloud and containers

Linux Containers

References:

- [1] http://www.slideshare.net/sssooraj/introduction-to-linux-containers
- [2] http://www.slideshare.net/fasgoncalves/hypervisor-versus-linux-containers
- [3] https://www.toptal.com/linux/separation-anxiety-isolating-your-system-with-linux-namespaces
-
- Security Features in systemd - NLUUG Najaarsconferentie 2014
- Containers and systemd - Berlin systemd Meetup 2015

Linux containers

Thanks!