LINUX PITER

Container Anatomy

Virtuozzo

by Pavel Tikhomirov ptikhomirov@virtuozzo.com

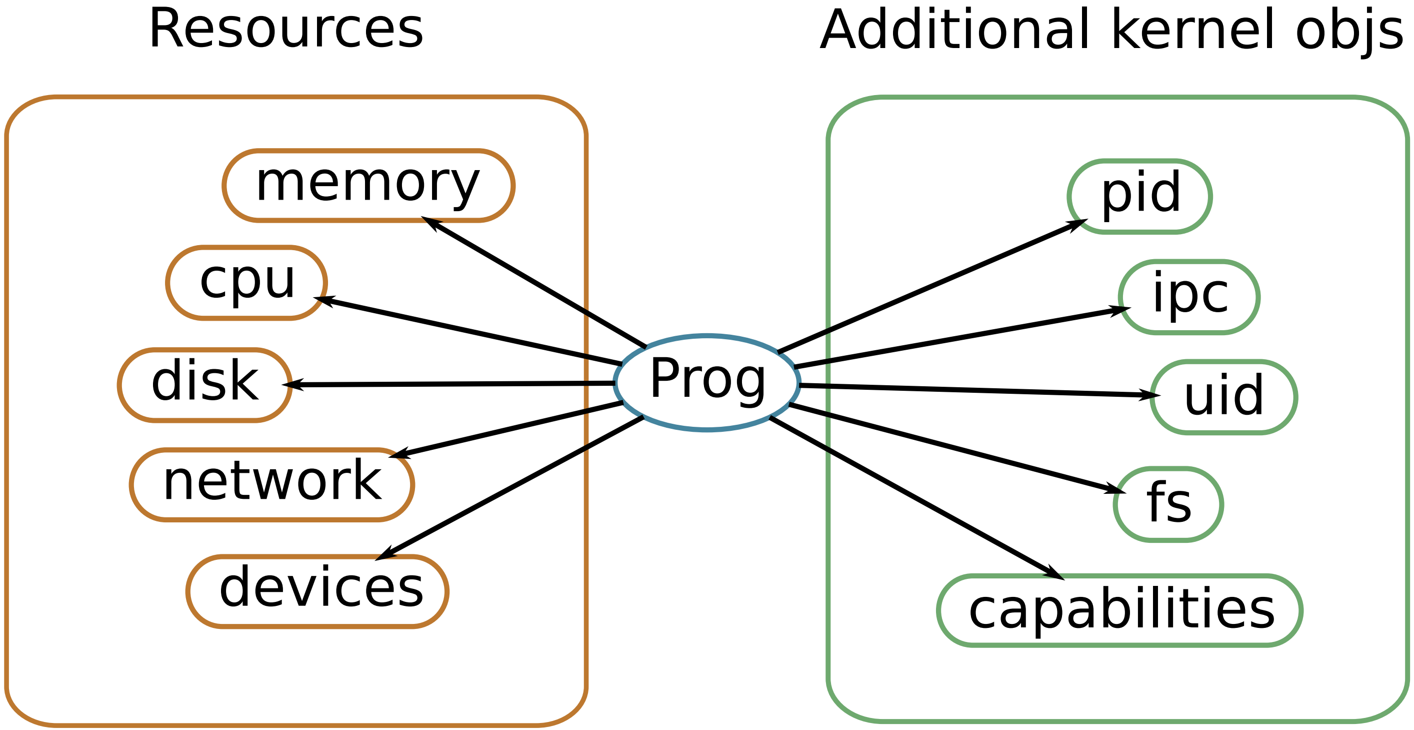
Agenda

- Brief history of containers
- Container skeleton
- Cgroups
- Namespaces
- Root fs
- Network
- Unikernel

Brief history of containers

- CT appeared as a replacement for VM
  - Relatively fast
  - High density
- But at what cost?
  - Security is a problem
    - Wide attack surface
  - Endless run to virtualize every new kernel feature or object
- Downshift to microservices

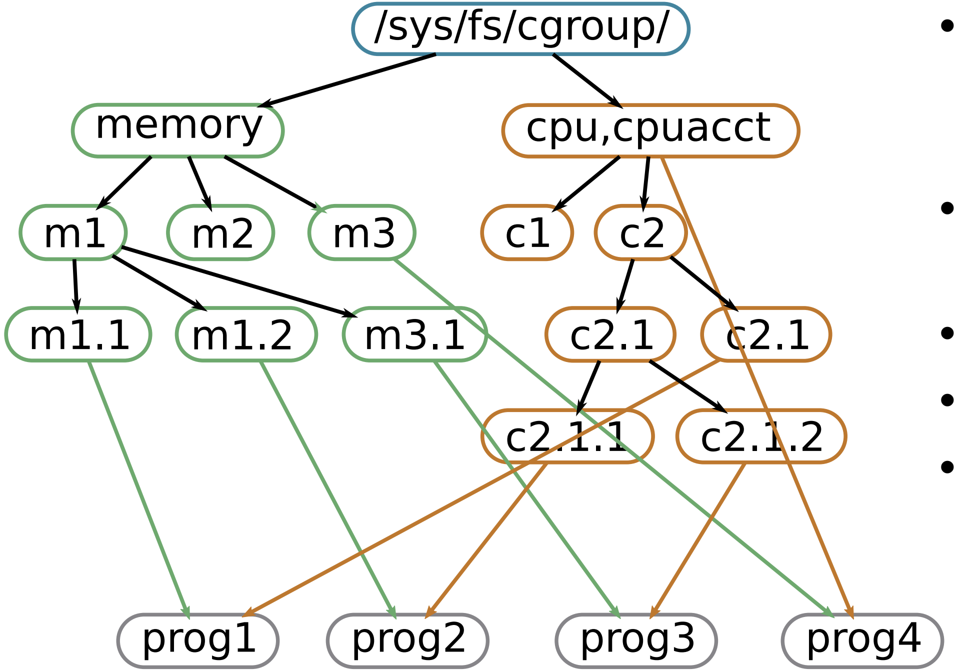
Container skeleton: what?



Container skeleton: how?

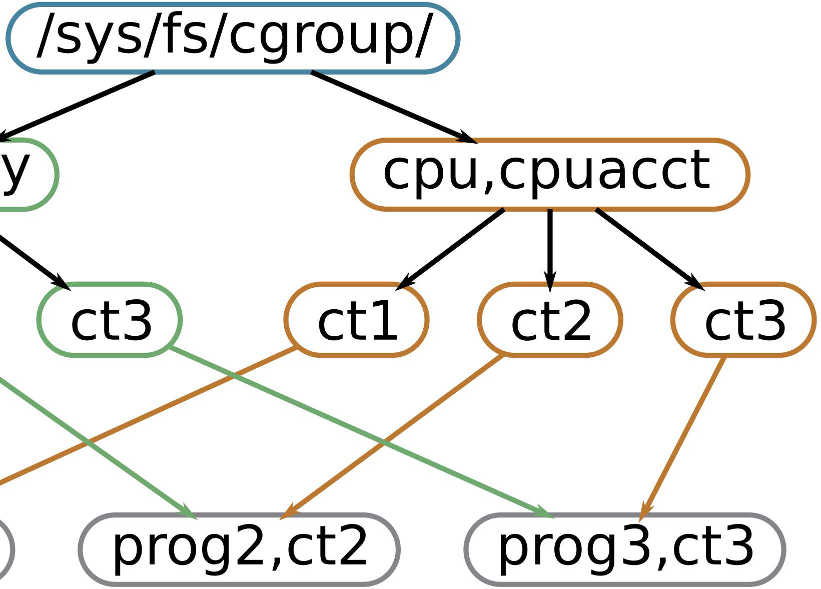
- cgroups + namespaces = limit + isolate
- root file system
- network setup
- container manager

Cgroups – control groups



- Any process is in cgroup (directory) in each hierarchy
- Limits relative or absolute amount of some resource
- Nested
- Inherited on fork
- Confusing configurations…

Cgroups unified hierarchy



Cgroups problems

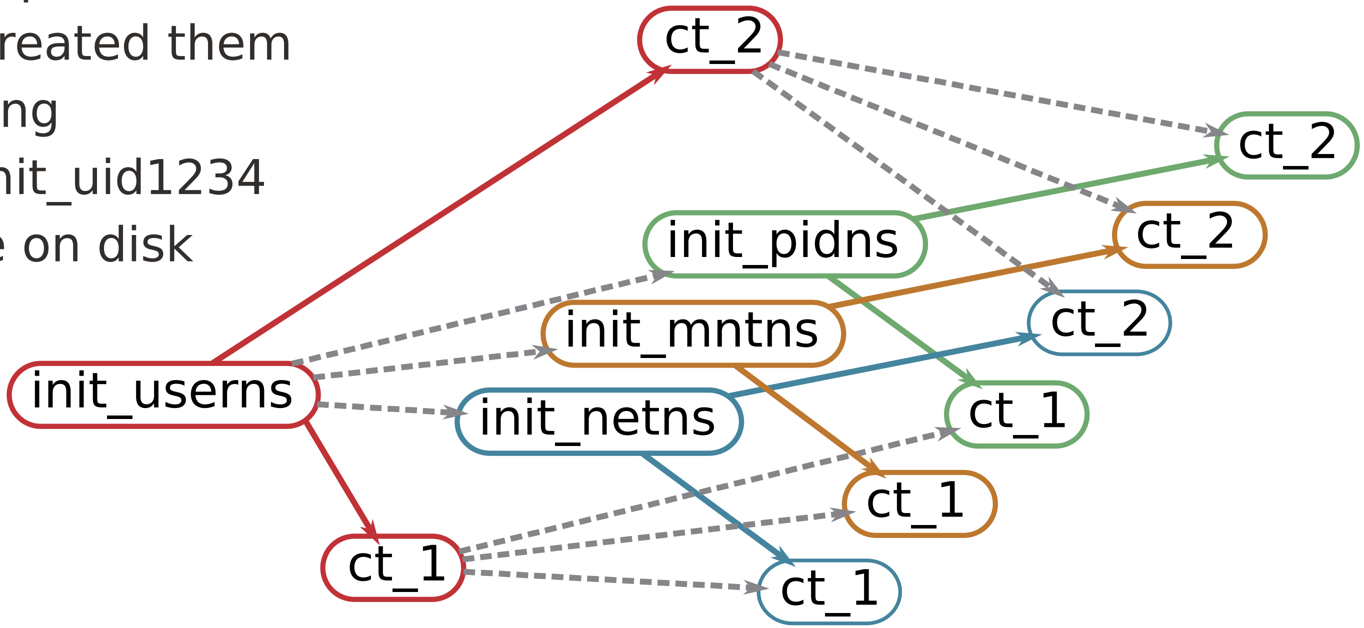
- Memcg no soft limit for kmem
- Memcg does not protect from single kernel object spamming
  - Separate kmem.tcp limits introdused
- How to choose limits?
- Performance slow down when accounting every allocation
- Cgroups in container =) ? - see in next slides
- “free” utility works strange in memory limited container – memory namespace?

Namespaces

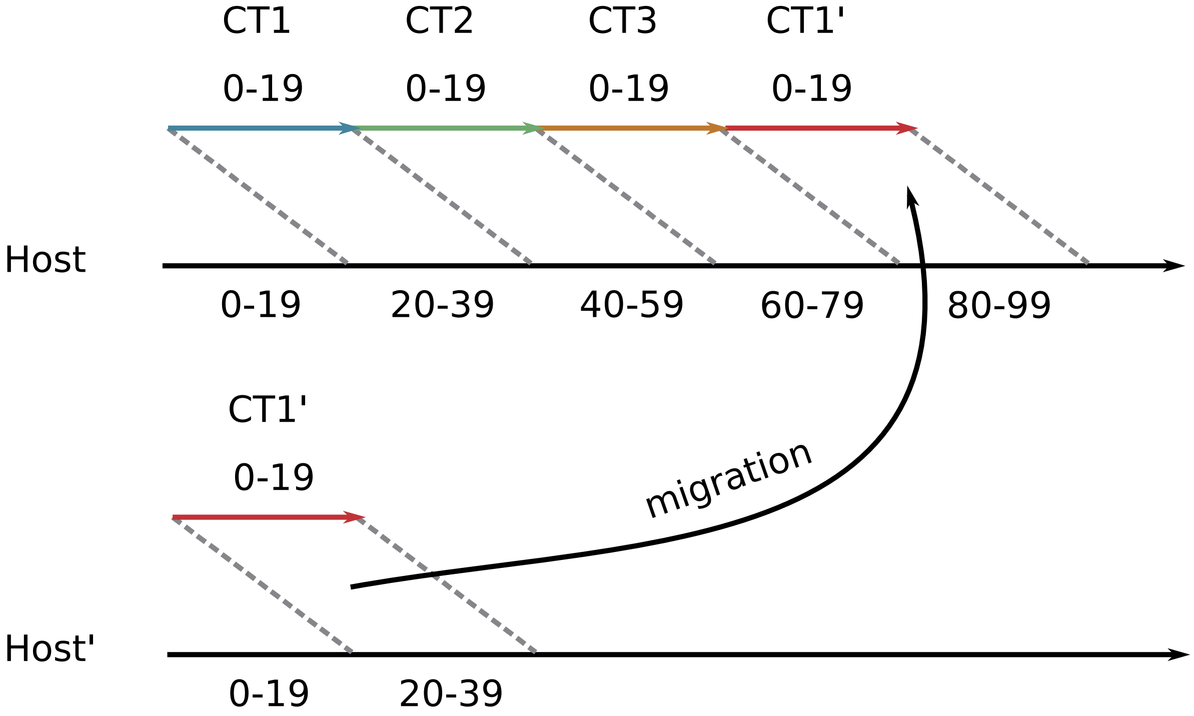
- Own isolated view on some kind of resource
- Every process is in one of each kind of namespaces
- Namespaces are inherited on fork
- Why no cpu namespace and own view on cpus(virtual) exists? - no idea...

Namespaces hierarchy & userns

- All other namespaces are linked to userns which created them
- User ids mapping
  - ct1_uid0 → init_uid1234
  - Real uids are on disk



Userns uid mapping



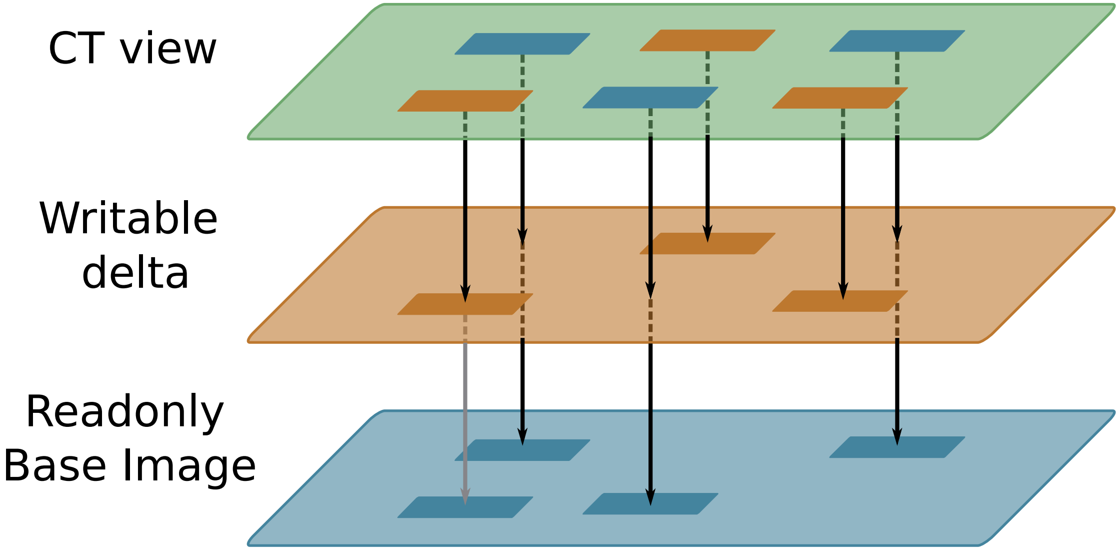
Cgroup namespace

- Need cgroups in container
  - Some usefull stats
  - Nested containers
- Writing to host cgroups from CT can escape their restrictions
- CT should feel like it is in root cgroup

Root file system

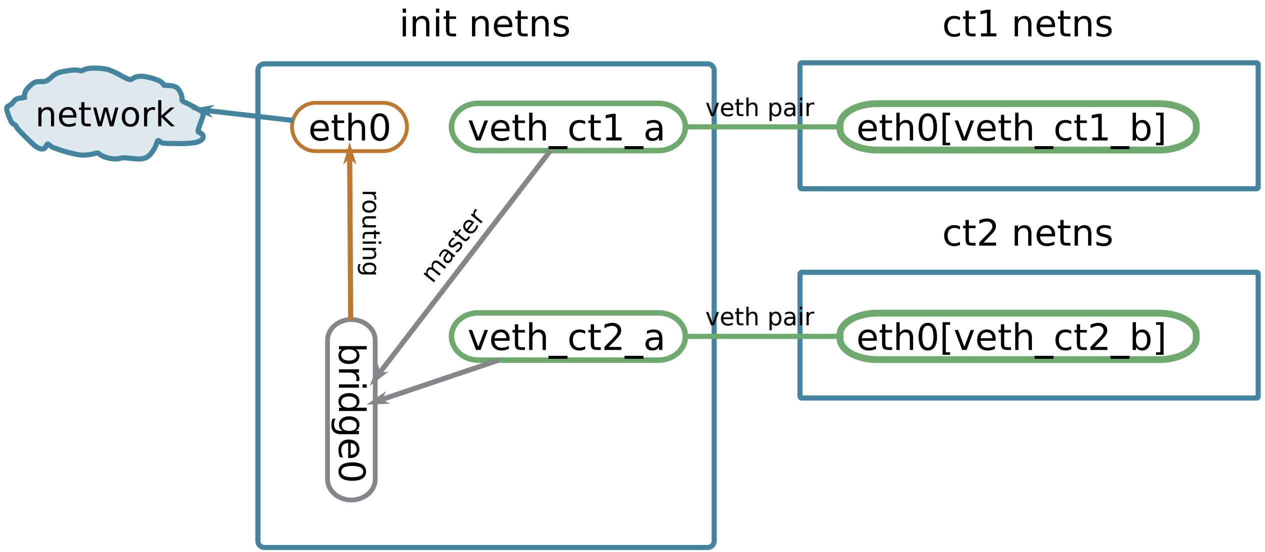
1. Enter userns+mntns
2. Copy container files to /path/to/container_roots/ct1_root/
   - Binaries, configurations and libraries
3. Optional – bindmount external directories
4. Change root to it with pivot_root
5. Exec the binary of your app

No copy – overlay/dm-thin

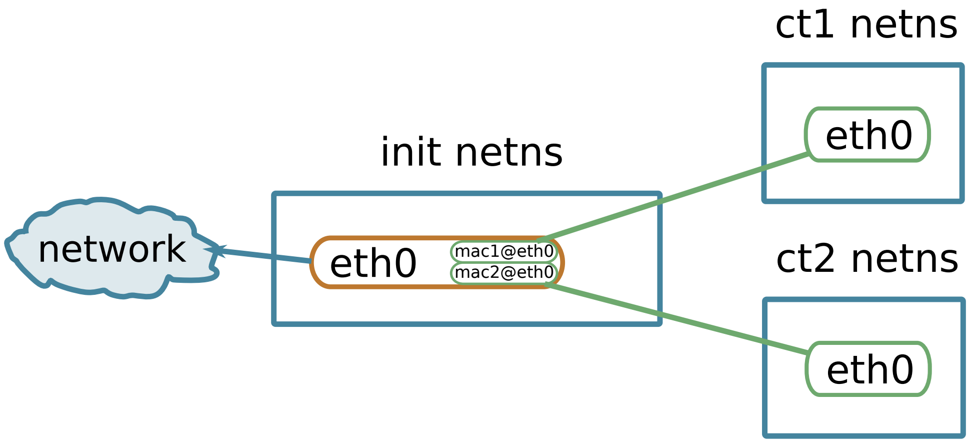


- Advantages:
  - Fast start
  - Base files/blocks are shared
  - Shared also in memory for overlay

Network veth



Network mac(ip)vlan



Container manager

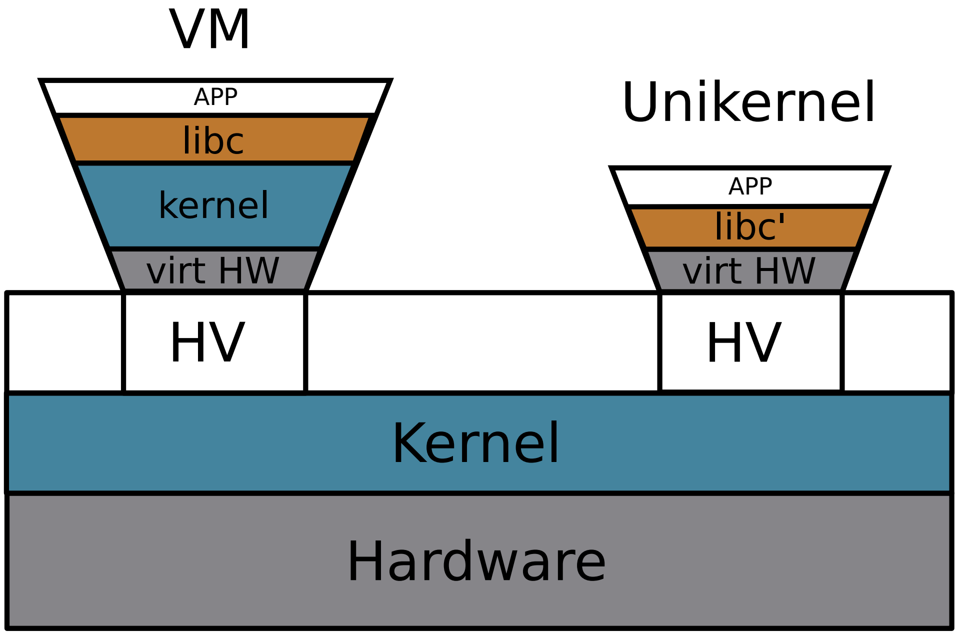
- Start, stop, configure
- Bring together all “container” parts
- Monitor container
  - Show stats
  - When container had stopped?
- “Make containers kernel objects”, David Howels, RedHat

  https://lkml.org/lkml/2017/5/22/645

Unikernel

- Single APP
- Minimal kernel
- Drivers for hypervisor
- Network stack support
- Physical address space
- Reduce context switches

- Fast and small as CT
- Secure as VM



Copyright © 2016 Virtuozzo. All Right Reserved.

Open Container Initiative

- Creates standards how container should look like
- On level higher than kernel

(cgroups and namespaces are almost standard everywhere)

- Unify images
- Unify runtime
- opencontainers.org

Any questions?

Virtuozzo

by Pavel Tikhomirov ptikhomirov@virtuozzo.com