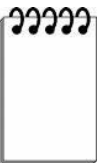# Linux Namespaces

Martin Leischner
October 11, 2022

Namespaces + cgroups

Namespaces: Lightweight <mark>process virtualization</mark>.

- **Different processes should have different views of the system resources.**

- **This allows different processes on a system to be isolated from each other. (<mark>process virtualization</mark>)**

- **Namespaces are built directly into Linux starting with <mark>kernel version 2.4.19</mark> (year 2002). Therefore no hypervisor is necessary for the isolation of processes.**

cgroups: control groups is a resource management solution.

- **Resources are assigned to a group of processes. The use of resources is monitored. In particular, it can be limited and prioritized.**

- **First approaches starting with Linux <mark>kernel version 2.6.24</mark>. Starting with <mark>Ubuntu 12.10</mark>.**

Applications: Linux Containers (LXC), Docker, Podman.

Important namespaces

● **PID namespace**

Groups processes that see each other.
No view to the outside.
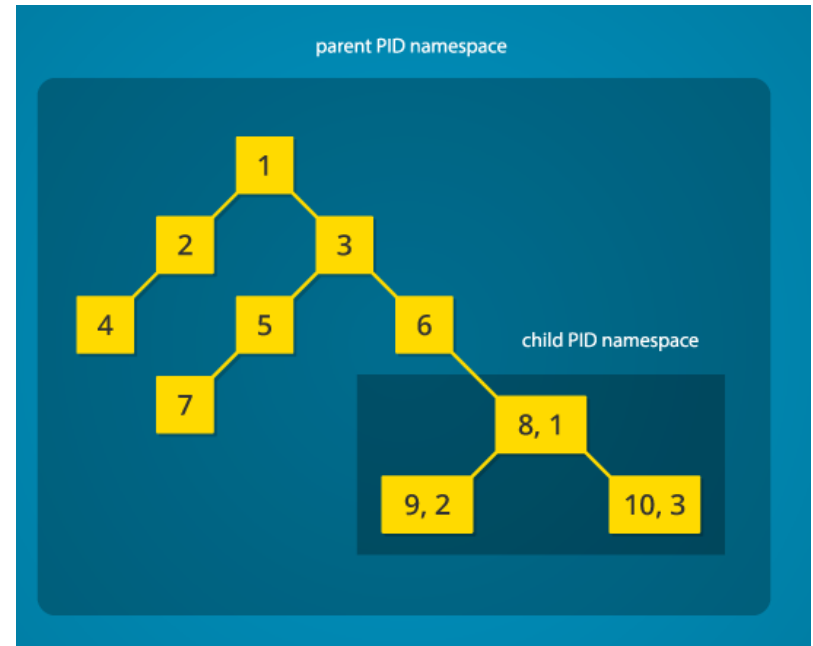First process in the respective namespace
gets the PID 1.

Fig.: Mahmud Ridwan, https://www.toptal.com/linux/separation-anxiety-isolating-your-system-with-linux-namespaces

● **Network namespace**

The interfaces in the namespace can be fully used in the namespace without conflicting
with the interfaces outside the namespace.

● **Mount namespace**

File systems can be mounted without affecting the host system.

● **… + Cgroup / IPC / User / UTS Namespace**

Short demo

Create a new PID namespace and switch to it.

For this the Linux command `unshare` is used. (http://manpages.ubuntu.com/manpages/xenial/man1/unshare.1.html)

`unshare`

`--fork`	Split off the specified program as a child.
`--pid`	Split off the PID namespace. remark: With --net the network namespace is split off.
`--mount-proc`	Mount the proc filesystem (= interface to the kernel) on the mountpoint /proc
`program`	Program that is started first in the new PID namespace.

End