CLOUD COMPUTING APPLICATIONS

Containers: Namespaces

Prof. Reza Farivar

Namespaces

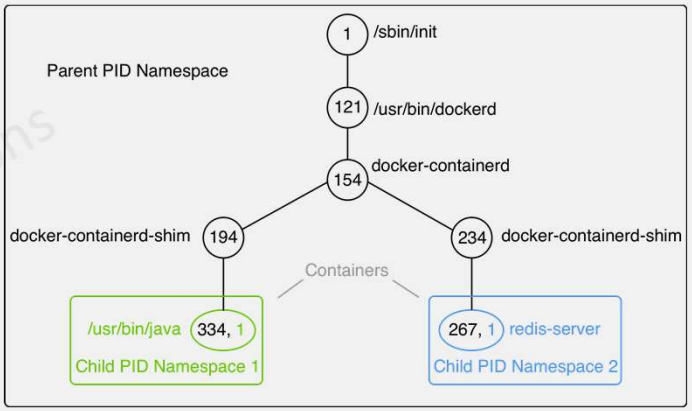
- A namespace wraps a global system resource in an abstraction that makes it appear to the processes within the namespace that they have their own isolated instance of the global resource
- Linux processes form a single hierarchy, with all processes rooting at init.
- Usually privileged processes in this tree can trace or kill other processes.
- Linux namespace enables us to have many hierarchies of processes with their own “subtrees” such that processes in one subtree can NOT access or even know of those in another.

Namespaces

Namespace	Description (This namespace isolates …)
Cgroup	Cgroup root directory
IPC	Isolates System V IPC, POSIX message queues. The common characteristic of these IPC mechanisms is that IPC objects are identified by mechanisms other than filesystem pathnames.
Network *	Network devices, stacks, ports, etc. each network namespace has its own network devices, IP addresses, IP routing tables, /proc/net directory, port numbers, etc.
Mount *	Mount points. processes in different mount namespaces can have different views of the filesystem hierarchy
PID *	Isolates Process IDs. In other words, processes in different PID namespaces can have the same PID.
Time	Boot and monotonic clocks
User *	User and group IDs. In other words, a process's user and group IDs can be different inside and outside a user namespace
UTS	Hostname and NIS domain name. Allows each container to have its own hostname and NIS domain name. Affects nodename and domainname—returned by the uname() system call.

PID Namespace example

- Without namespace, all processes descend hierarchically from PID 1(init).
- If we create a PID namespace and run a process in it, that first process becomes PID 1 in that namespace.
- The process that creates namespace still remains in parent namespace, but makes its child the root of new process tree.
- The processes within the new namespace can not see parent process but the parent process namespace can see the child namespace.
- The processes within new namespace have 2 PIDs: one for new namespace and one for global namespace.
- PID namespaces also allow each container to have its own `init` (PID 1), the "ancestor of all processes" that manages various system initialization tasks and reaps orphaned child processes when they terminate



Network Namespace

- Provide isolation of the system resources associated with networking
- each network namespace has its own network devices, IP addresses, IP routing tables, /proc/net directory, port numbers, and so on
- Network namespaces make containers useful from a networking perspective
  - Each container can have its own (virtual) network device and its own applications that bind to the per-namespace port number space
  - suitable routing rules in the host system can direct network packets to the network device associated with a specific container
- E.g. have multiple containerized web servers on the same host system, with each server bound to port 80 in its (per-container) network namespace

User Namespace

- process's user and group IDs can be different inside and outside a user namespace
- The most interesting case here is that a process can have a normal unprivileged user ID outside a user namespace while at the same time having a user ID of 0 inside the namespace.
  - This means that the process has full root privileges for operations inside the user namespace, but is unprivileged for operations outside the namespace.
- From a security perspective this is a great feature as it allows our containers to continue running with root privileges, but without actually having any root privilege on the host.
  - Docker since 2016

Mount Namespace

- Mount namespaces were the first type of namespace to be implemented on Linux, appearing in 2002
  - by contrast with the use of the chroot() system call, mount namespaces are a more secure and flexible tool for this tas
- Isolate the set of filesystem mount points seen by a group of processes
- Processes in different mount namespaces can have different views of the filesystem hierarchy