Linux kernel infrastructure for Containers

Srivatsa S. Bhat
VMware
srivatsa@csail.mit.edu

University of Washington
20 Feb 2018
Virtual Machines vs Containers
What is a container?

Containers:
• Provide a virtual Operating System environment
• Are processes with enhanced grouping and isolation
• Share an underlying kernel
• Don’t need special hardware support (eg: VT-x etc)
Building blocks of containers

- Namespaces
- Control Groups (cgroups)
- And the rest of the traditional OS abstraction (processes, files, networking, IPC, users etc)
Namespaces

What is a namespace?
• A collection of names identifying objects/entities
• A technique to partition a global resource into smaller scope.

Namespaces in the Linux kernel:
• UTS namespace
• mnt namespace
• PID namespace
• User namespace
• Network namespace
• IPC namespace
Namespace API

- clone(function, stack, CLONE_NEW*, args);
  - UTS namespace - CLONE_NEWUTS
  - mnt namespace - CLONE_NEWNS
  - PID namespace - CLONE_NEWPID
  - User namespace - CLONE_NEWUSER
  - Network namespace - CLONE_NEWNET
  - IPC namespace - CLONE_NEWIPC

- setns(fd, nstype);  fd refers to one of /proc/PID/ns/**

- unshare(flags);
Abstracts:
• sethostname()
• setdomainname()
• uname()
mnt namespace

Abstracts:
• mount points
• File system hierarchy

Features:
• Supports shared subtrees via mount-event propagation
  • MS_SHARED
  • MS_PRIVATE
  • MS_SLAVE
  • MS_UNBINDABLE
PID namespace

Abstracts:
• Process ID numbers

Features:
• Hierarchical
• Special semantics for the ‘init’ process in each PID namespace
  • Reaping orphan tasks
  • Restrictions on sending signals to the init process
User namespace

Abstracts:
• User IDs, Group IDs and Capabilities

Features:
• Hierarchical

• Unprivileged process can create user namespaces
  • Gets full capabilities in new user namespace
  • Root privileges inside namespace; unprivileged outside.

• UID/GID mappings defined using:
  • /proc/PID/uid_map
  • /proc/PID/gid_map

• Used in conjunction with other namespaces
Other namespaces

Network namespace:
• Abstracts network devices, IP addresses, port numbers etc.
• Eg:
  • ip netns add mynetns
  • ip netns exec mynetns <command>
  • ip netns delete mynetns

IPC namespace:
• Abstracts Sys V IPC (shared memory etc), POSIX message queues
Control groups

Cgroups provide resource control for various system resources
  • Eg: CPU time, memory consumption, I/O bandwidth etc.

/sys/fs/cgroup:
  • cpuset
  • cpu,cpuacct
  • blkio
  • memory
  • ...

• Cgroups are mostly orthogonal to namespaces
  • Resource limits can be applied to any group(s) of processes
  • Offers flexibility in applying resource limits on containers
Putting it all together

**Docker**
- Dockerfile – used to build container images
- Container images – layered using copy-on-write filesystem overlay
- Container registries – reusable container image layers

**Kubernetes**
- Provides container orchestration and management
- Microservices – a new paradigm to deploy containerized apps

**Containers in the cloud**
- Containers as a service, as opposed to virtual machines

Tip: Check out the **contain** tool, a bare-bones container runtime: