Docker Internals

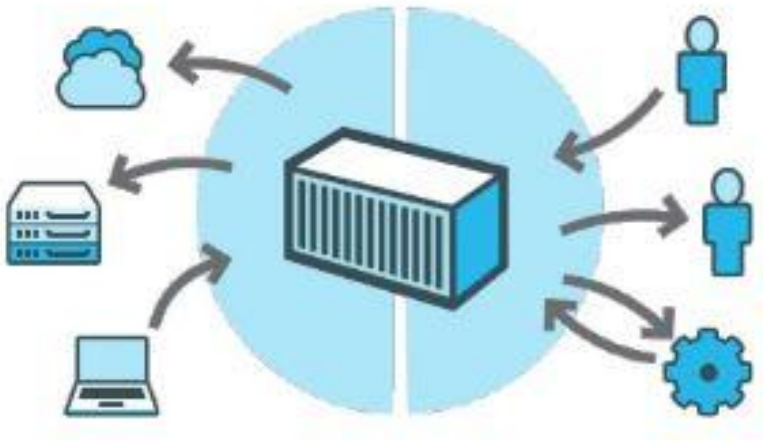
Rajesh Kumar

DevOps Architect

@RajeshKumarIN | www.RajeshKumar.xyz

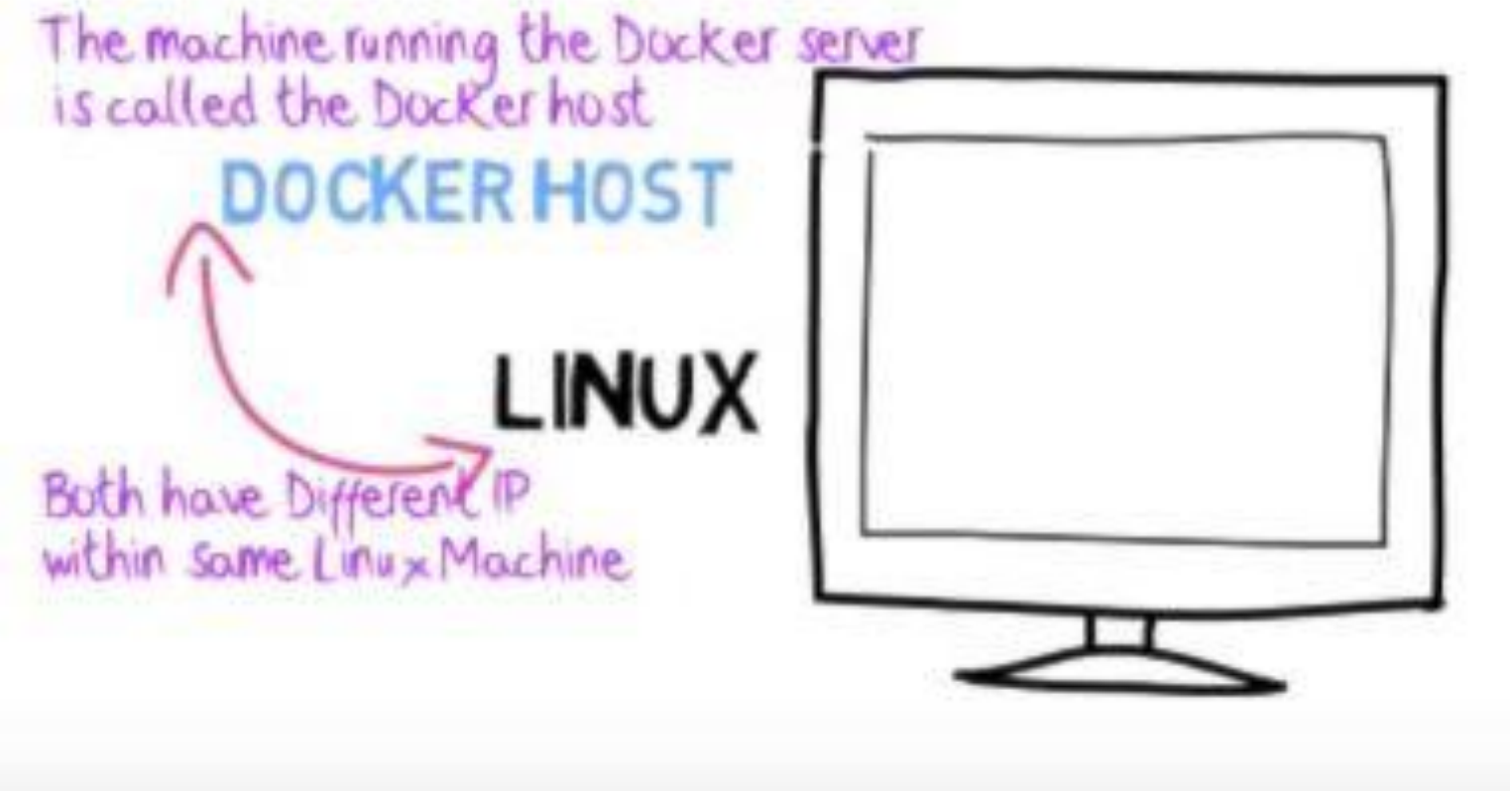
Build Once, Configure Once.

Deploy Everything*
Everywhere*
Reliably & Consistently
Efficiently
Cheaply



How Docker Works!

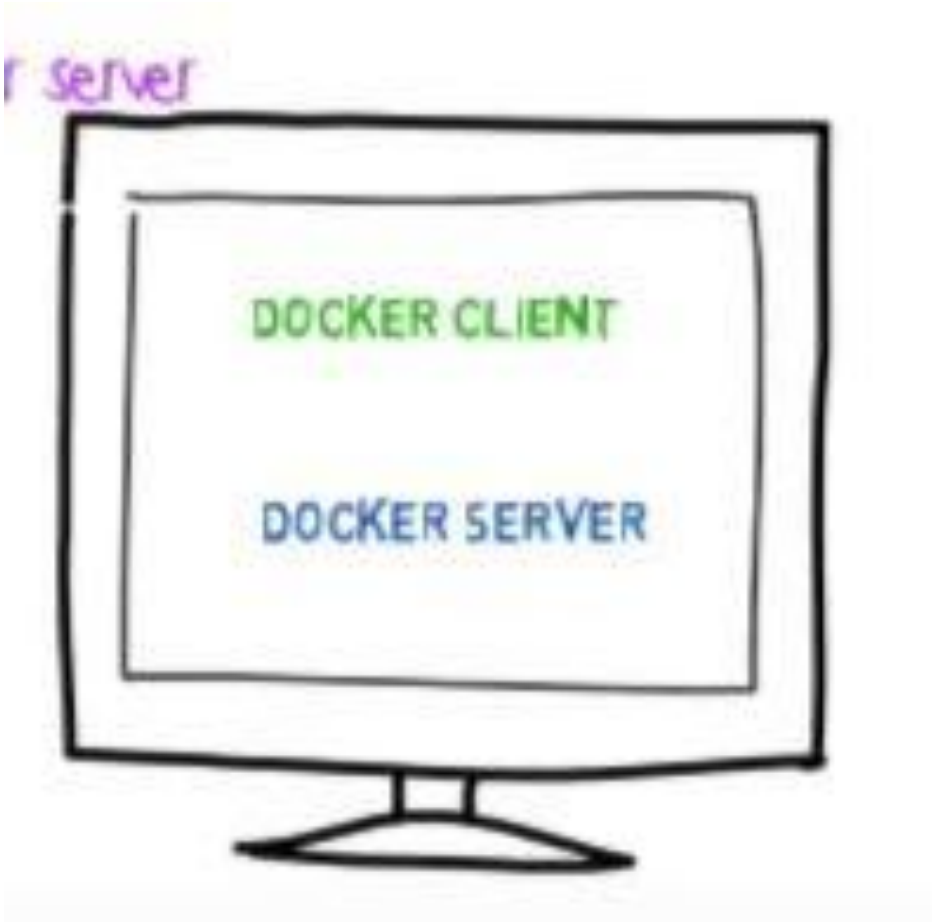
How Docker Works!



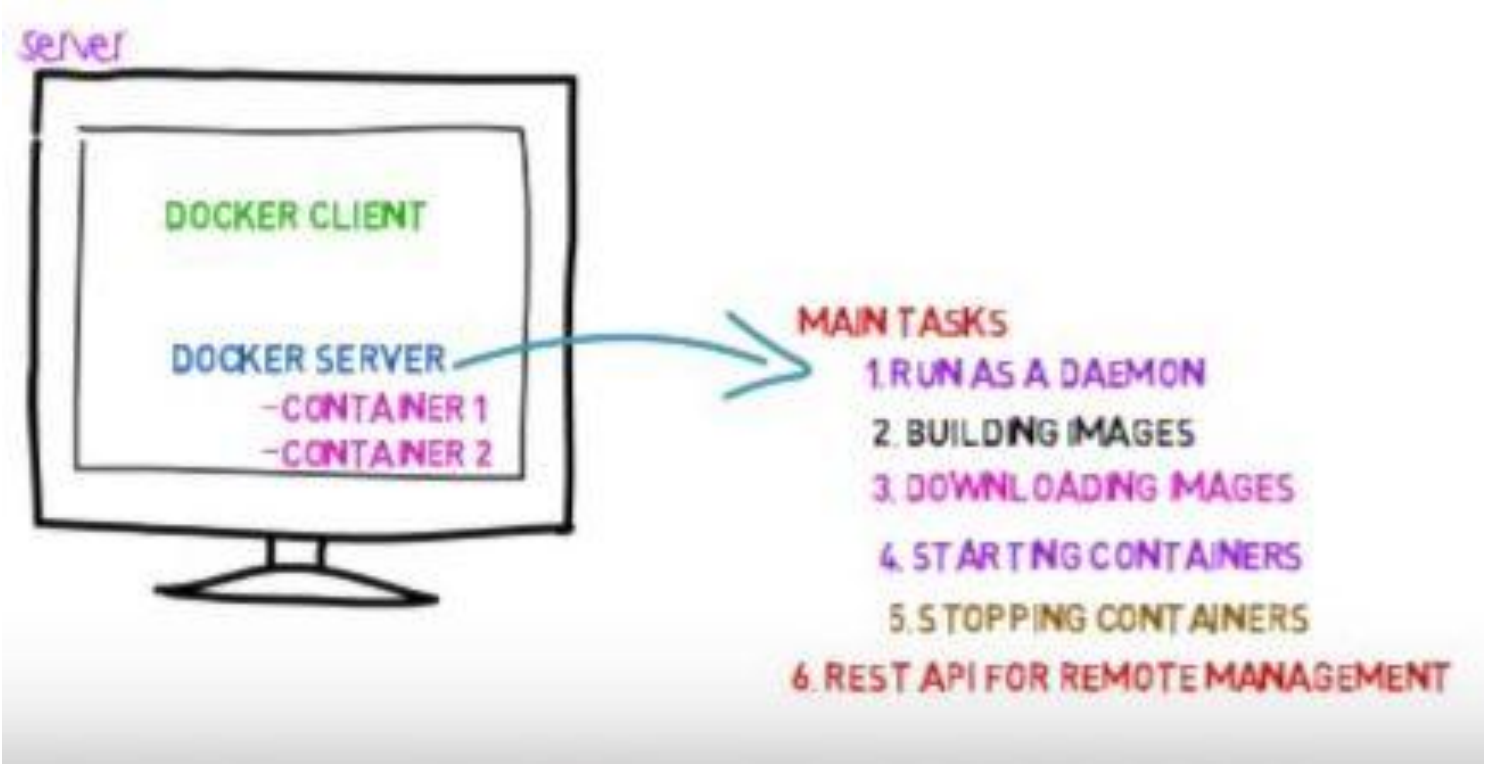
How Docker Works!

DOCKER IS CLIENT SERVER APPLICATION

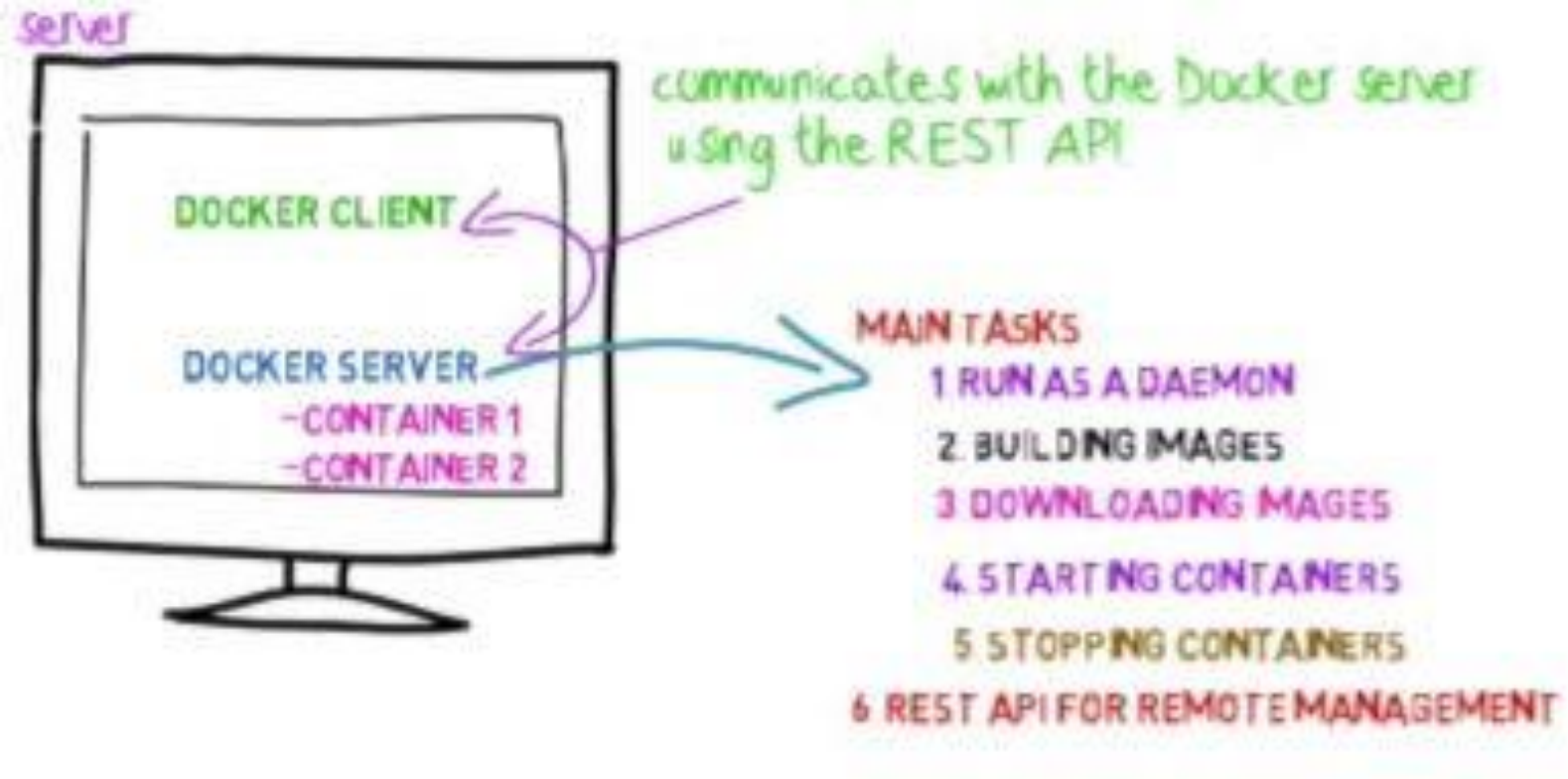
How Docker Works!

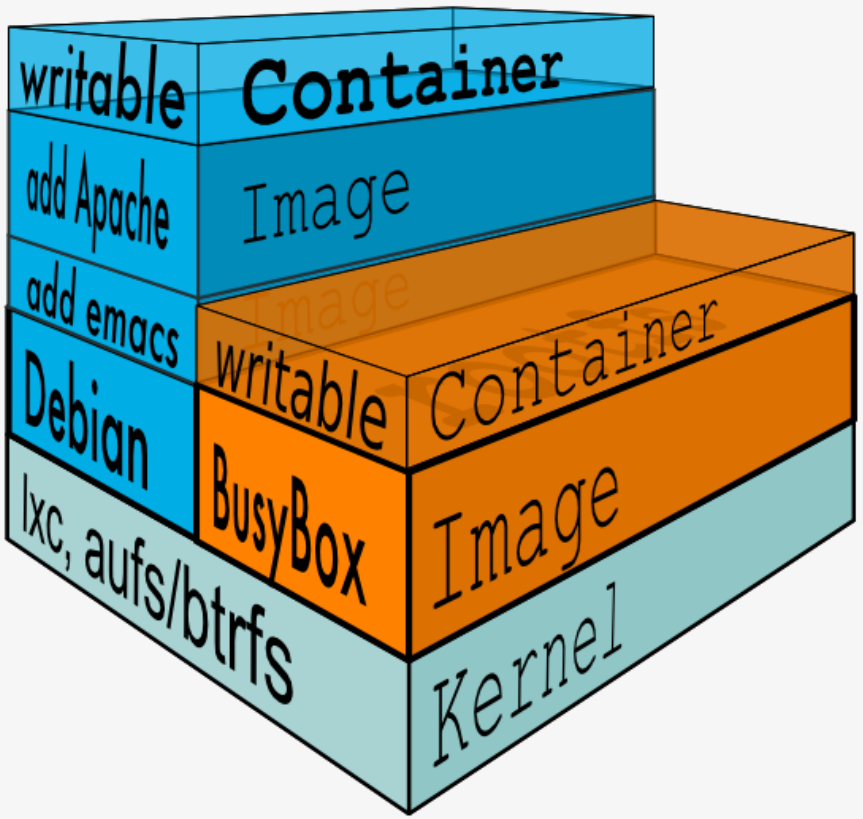


How Docker Works!

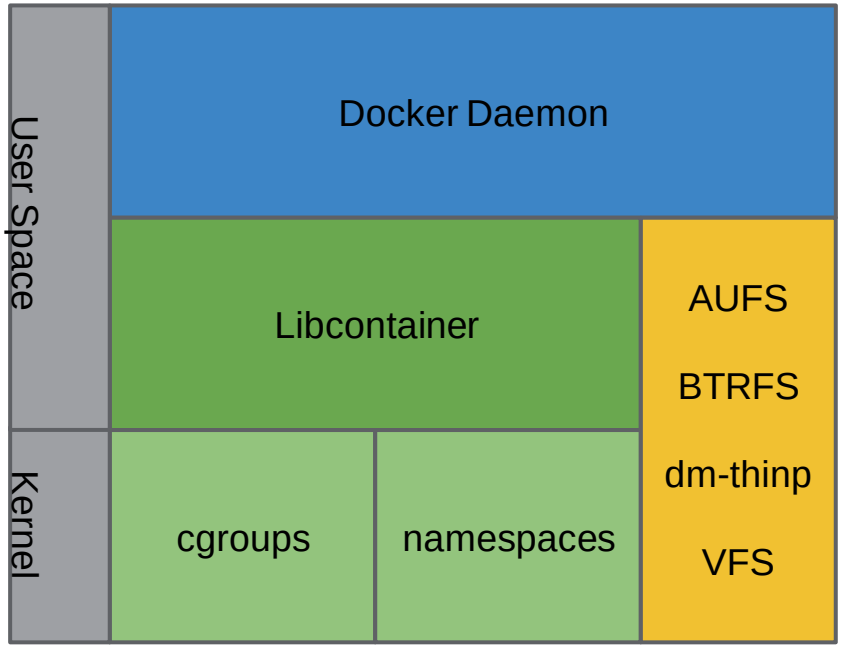

server
DOCKER CLIENT
DOCKER SERVER
-CONTAINER 1
-CONTAINER 2
MAIN TASKS
1. RUN AS A DAEMON
2. BUILDING IMAGES
3. DOWNLOADING IMAGES
4. STARTING CONTAINERS
5. STOPPING CONTAINERS
6. REST API FOR REMOTE MANAGEMENT

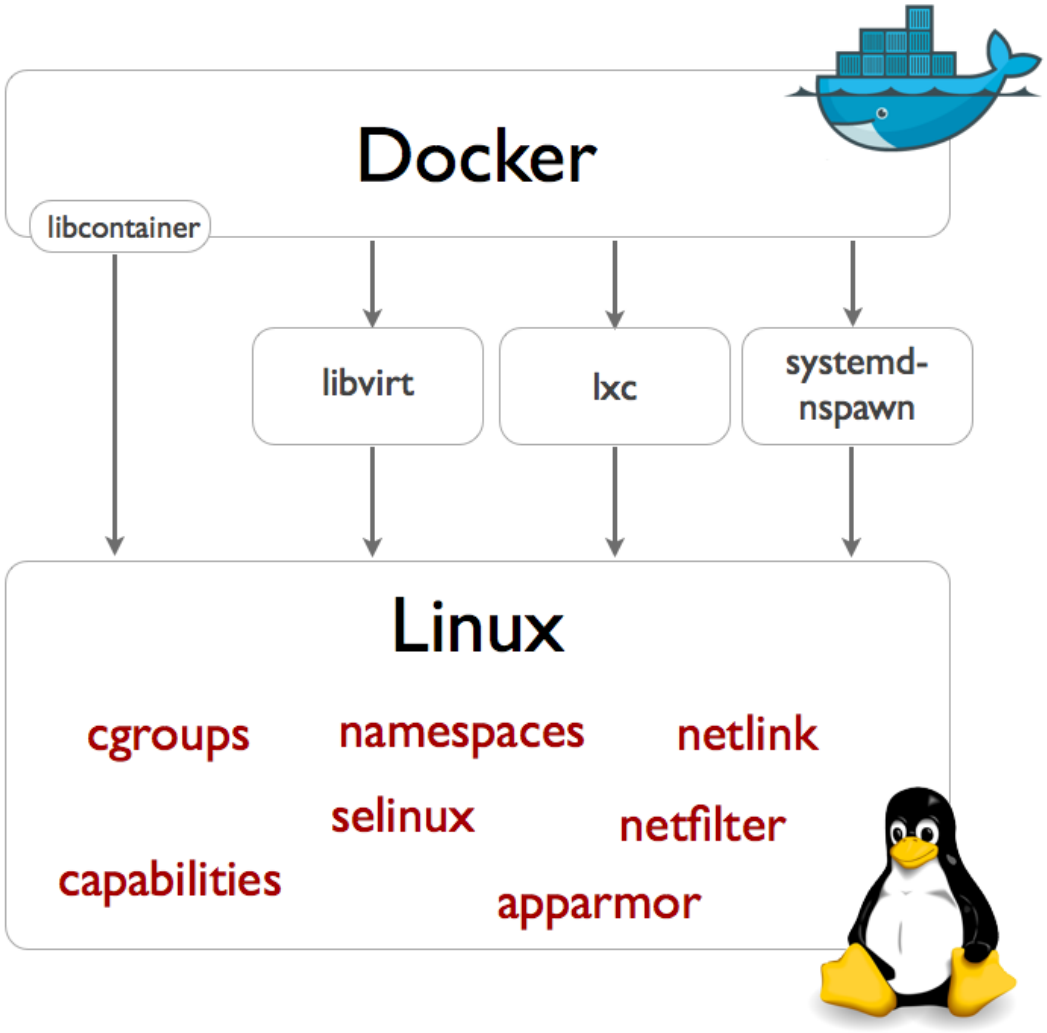
How Docker Works!

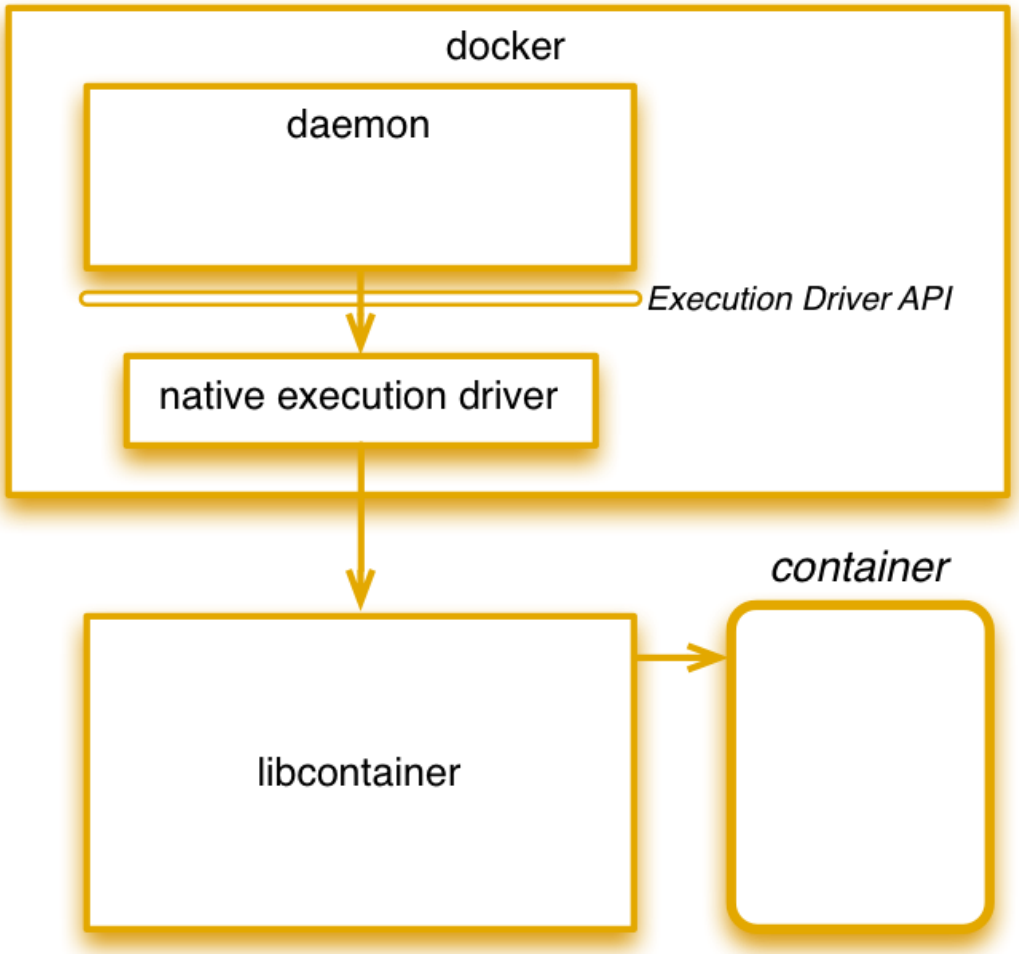



writable
Container
add Apache
Image
add emacs
writable
Container
Debian
BusyBox
Image
lxc, aufs/btrfs
Kernel

Docker Components




Docker
libcontainer
libvirt
lxc
systemd-
nspawn
Linux
cgroups
namespaces
netlink
selinux
netfilter
capabilities
apparmor


docker
daemon
Execution Driver API
native execution driver
container
libcontainer

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. Changes to the global resource are visible to other processes that are members of the namespace, but are invisible to other processes.
- One use of namespaces is to implement containers.

Linux provides the following namespaces:

Namespace	Constant	Isolates
Cgroup	CLONE_NEWCGROUP	Cgroup root directory
IPC	CLONE_NEWIPC	System V IPC, POSIX message queues
Network	CLONE_NEWNET	Network devices, stacks, ports, etc.
Mount	CLONE_NEWNS	Mount points
PID	CLONE_NEWPID	Process IDs
User	CLONE_NEWUSER	User and group IDs
UTS	CLONE_NEWUTS	Hostname and NIS domain name

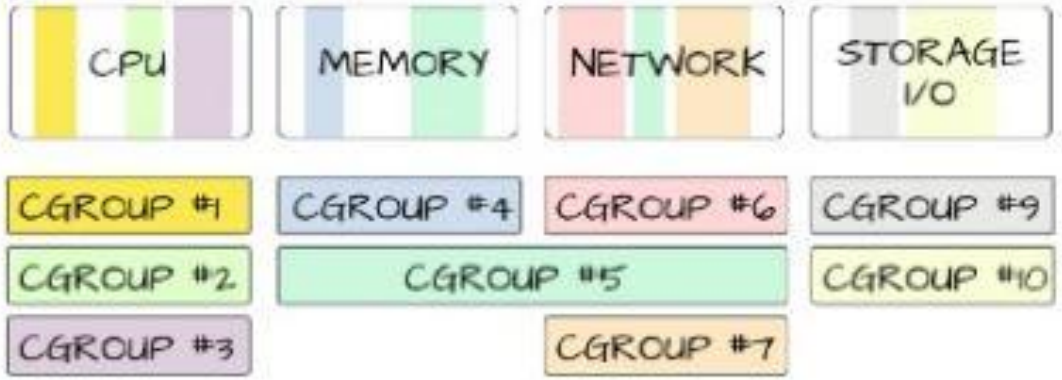
CGROUPS

cgroups (abbreviated from control groups) is a Linux kernel feature that limits, accounts for, and isolates the resource usage (CPU, memory, disk I/O, network, etc.) of a collection of processes.

Docker: Isolations

Cgroups : Isolation and accounting

- cpu
- memory
- block i/o
- devices
- network
- numa
- freezer



LXC & libcontainer

- Linux Containers (LXC) was used before docker 1.8 as one execution driver by docker, and offered a userspace interface for the Linux kernel containment features. It is very specific to Linux

- libcontainer (now opencontainers/runc) is an abstraction, in order to support a wider range of isolation technologies

LibContainer Overview

- Libcontainer is now the default docker execution environment. It is driver (named native) and a library.
- In other words, it is a replacement (since version 0.9) for formerly LXC execution environment

More Reading

http://jancorg.github.io/blog/2015/01/03/libcontainer-overview/

http://www.zdnet.com/article/docker-libcontainer-unifies-linux-container-powers/

Docker: Filesystems

File-system Isolation:

Building a rootfs dir and chroot into it.

With mount namespace, use pivot-root.

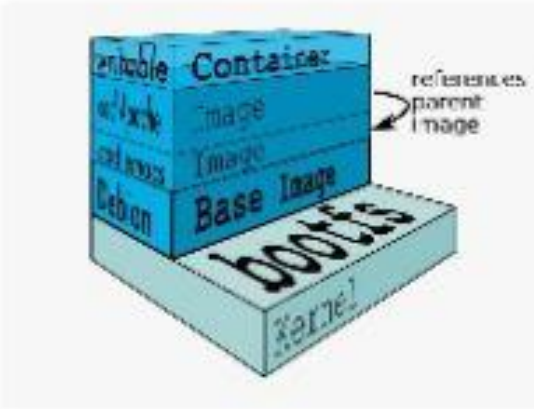
Features:

Layering, CoW, Caching, Diffing

Solutions:

UnionFS, Snapshotting FS, VFS

AUFS in action



Docker: Security

Security Layers

- Linux Capabilities.
- User namespaces: Unprivileged users.
- nosuid & ro mounts.
- Seccomp-bpf
- GRSEC and PAX
- Device cgroups
- Access Control: SELinux & AppArmor
- Future: Namespace aware sys/proc

More Reading

Google.com

Stackoverflow.com

Docker.com