云计算技术与实践

第三章 云计算关键技术
—轻量级虚拟化

云南大学软件学院 梁宇
E_mail：yuliang@ynu.edu.cn
Tel：0871-65931534

目　录

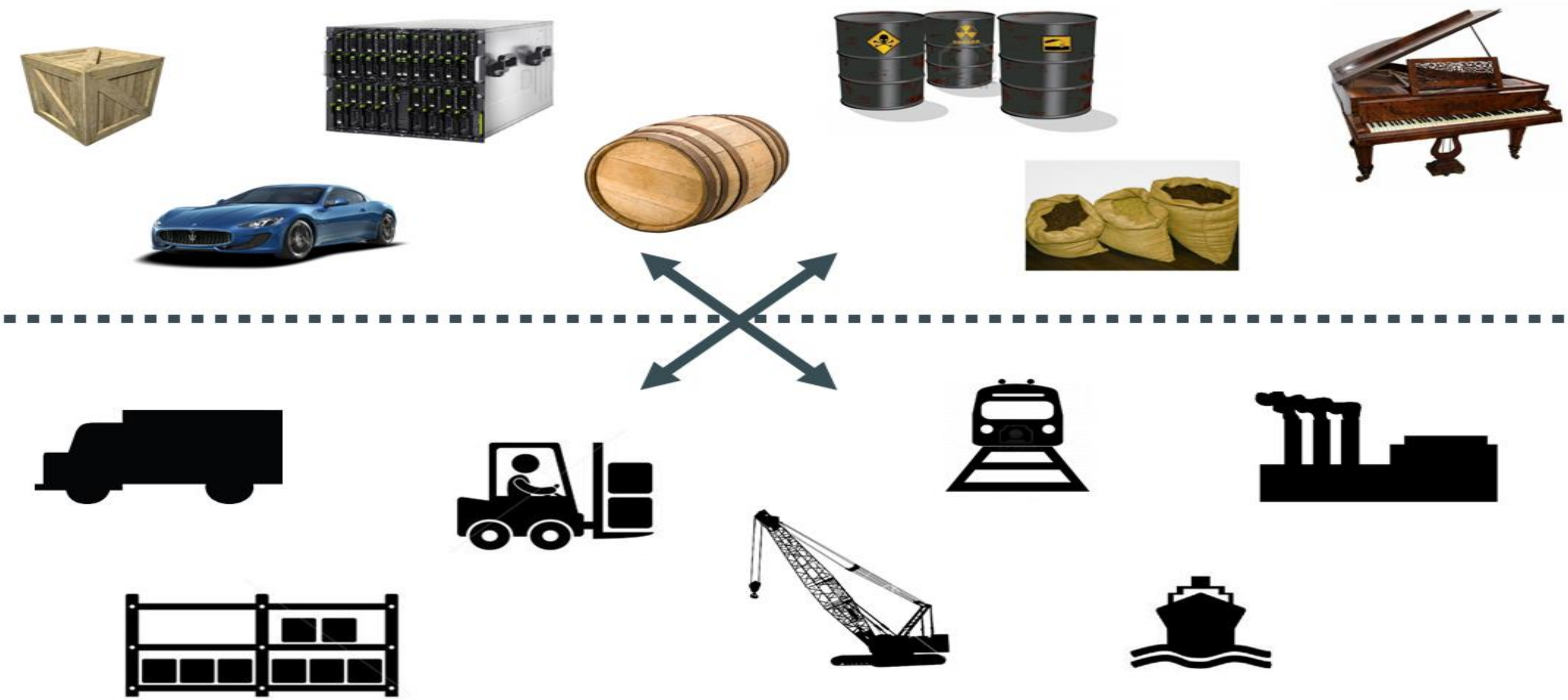
1、计算（服务器）虚拟化存在的问题

1) 环境管理复杂：从各种OS到各种中间件再到各种App，产品能够成功发布，作为开发者需要关心的东西太多，且难于管理，这个问题在软件行业中普遍存在并需要直接面对。
2) 云计算时代的到来： AWS的成功, 引导开发者将应用转移到云上, 解决了硬件管理的问题，然而软件配置和管理相关的问题依然存在（AWS cloudformation 是这个方向的业界标准, 样例模板可参考这里）。
3) 虚拟化手段的变化：云时代采用标配硬件来降低成本，采用虚拟化手段来满足用户按需分配的资源需求，以及保证可用性和隔离性。然而无论是KVM还是Xen，都存在浪费资源，因为用户需要的是高效运行环境而非OS，GuestOS既浪费资源又难于管理。

2、轻量级计算虚拟化

Docker – 轻量级虚拟化技术

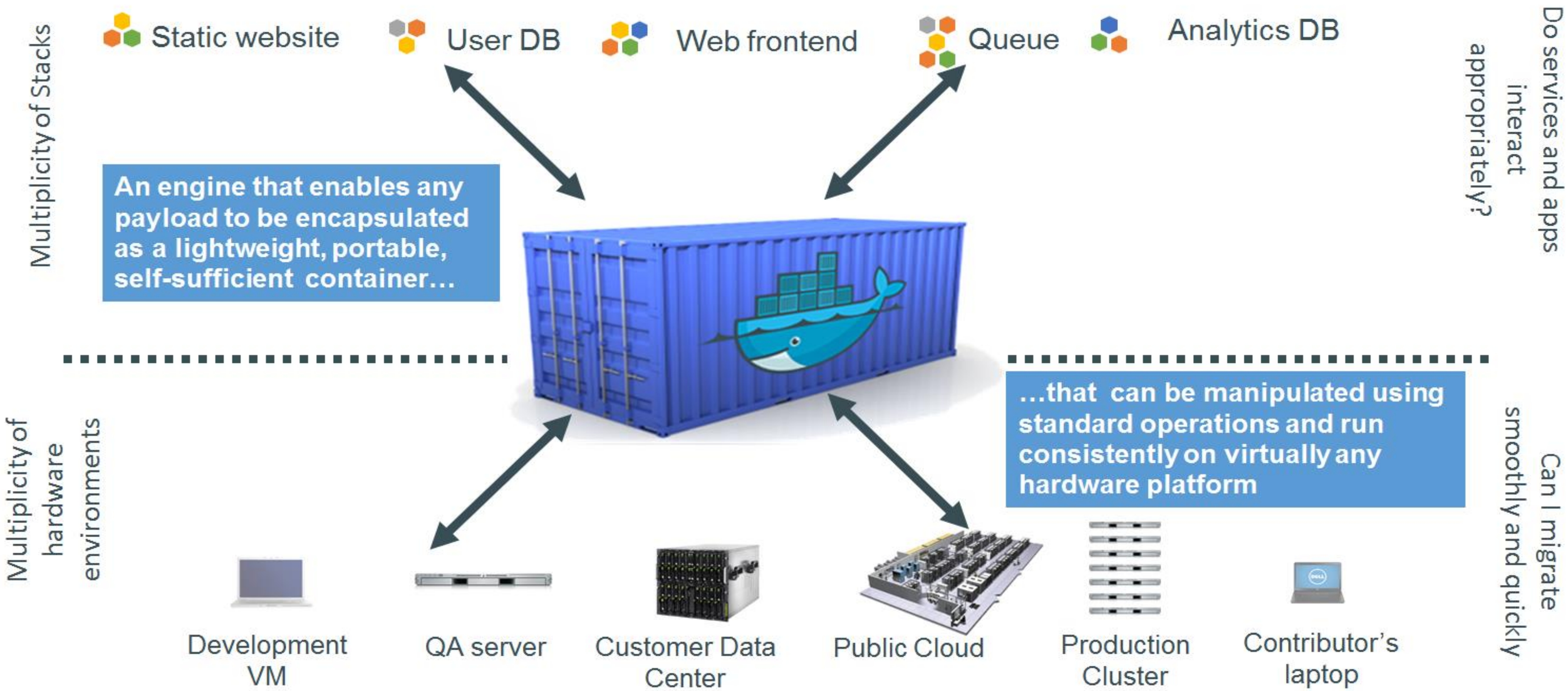
Docker是一个构建在LXC之上的，基于进程容器(Process Container)的轻量级VM解决方案。



2、轻量级计算虚拟化

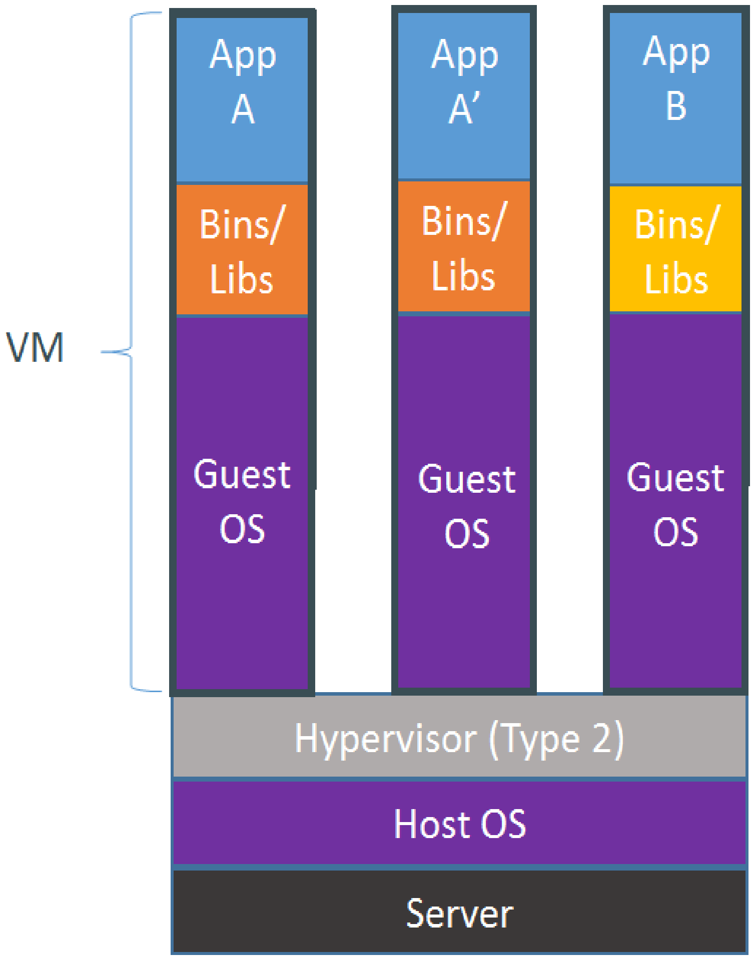
Docker – 轻量级虚拟化技术

Docker是一个构建在LXC之上的，基于进程容器(Process Container)的轻量级VM解决方案。



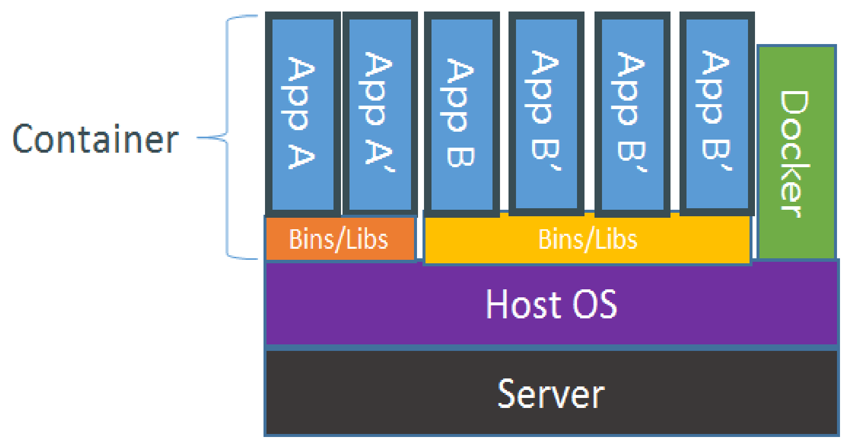
2、轻量级计算虚拟化

轻量级的LXC（ Linux 容器）更加灵活和快速：



Containers are isolated, but share OS and, where appropriate, bins/libraries

...result is significantly faster deployment, much less overhead, easier migration, faster restart



2、轻量级计算虚拟化

轻量级的LXC（ Linux 容器）

- Linux进程运行的共享环境和资源：
  - Linux内核
  - 文件系统
  - 网络系统
  - PID、UID、IPC等资源
  - 内存、CPU、磁盘等资源
  - 其他

2、轻量级计算虚拟化

轻量级的LXC（ Linux 容器）：

- 系统进程的管理要求：
  - 资源隔离
  - 资源限制
- 进程的进一步要求：
  - 能对一组进程进行隔离和资源限制

2、轻量级计算虚拟化

轻量级的LXC（ Linux 容器）更加灵活和快速：

需求总结

- 为一组进程分配独立的运行环境
  - 文件系统
  - 网络系统
  - PID、UID、UTS、mount、IPC空间
- 能对他们能使用的资源从整体上进行限制
  - 内存
  - CPU
  - 网络
  - 存储空间
- 防止进程组之间互相干扰

2、轻量级计算虚拟化

轻量级的LXC（ Linux 容器）更加灵活和快速：

轻量级虚拟化提出和定义

- 用来满足上述限制条件的进程组称为轻量级虚拟机，或者容器（Container）
- 进程容器（Process Container）的概念是由Google工程师在2006年推出来的。
  - http://lwn.net/Articles/199643/
  - http://lwn.net/Articles/236038/
- Wikipedia定义
  - http://en.wikipedia.org/wiki/Operating_system-level_virtualization

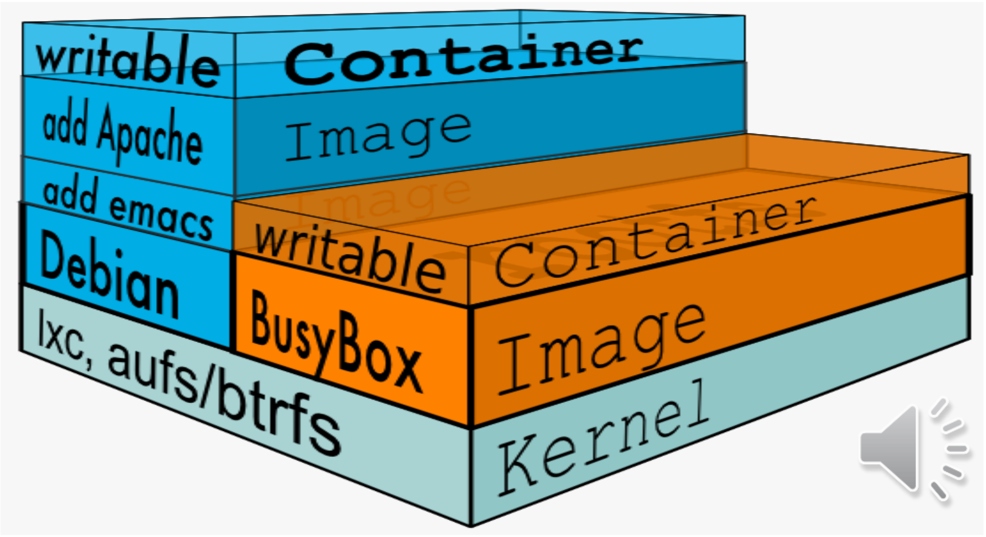
3、轻量级虚拟化原理与架构

- **轻量级虚拟化原理**

➢ 轻量级虚拟化本质上是在底层使用LXC启动一个Linux Container，通过cgroup等机制对不同的container内运行的应用程序进行隔离,权限管理和quota分配等

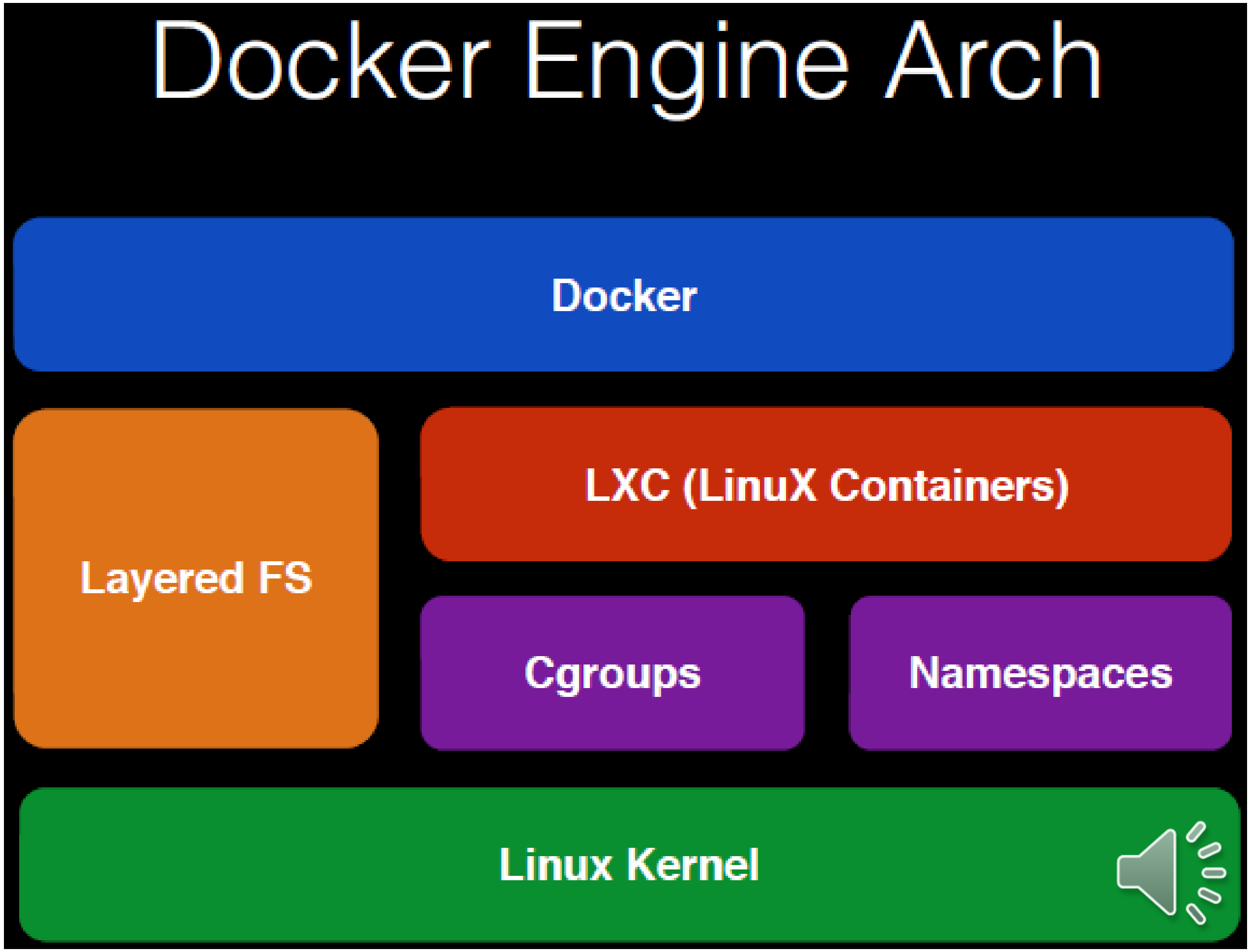
➢ 每个container拥有自己独立的各种命名空间（亦即资源）包括：PID 进程, MNT文件系统, NET网络, IPC , UTS主机名等

➢ 在LXC的基础上，轻量级虚拟化额外提供的功能包括：标准统一的打包部署运行方案， 历史版本控制， Image的重用，Image共享发布等。



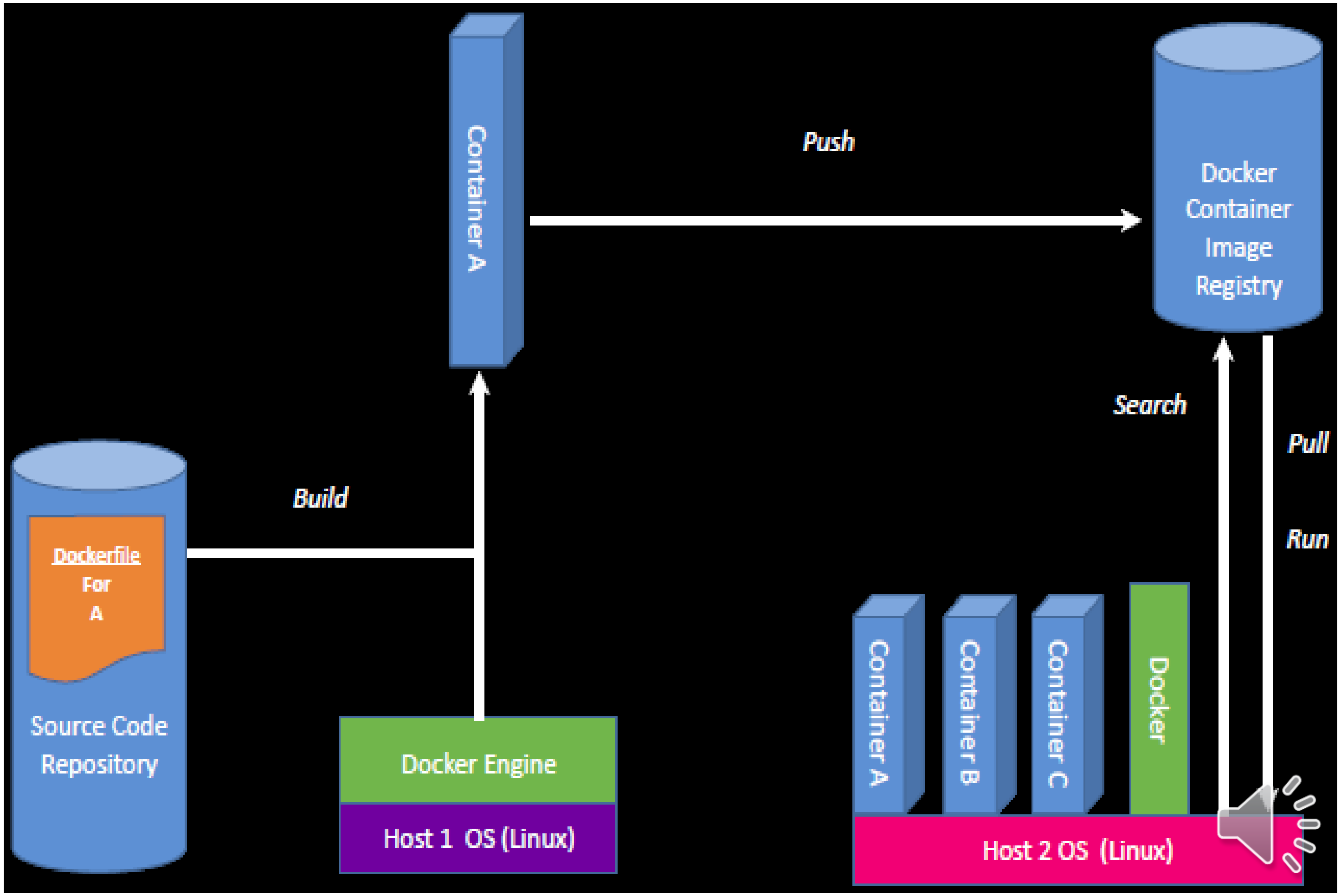
3、轻量级虚拟化原理与架构

- 轻量级虚拟化Docker架构



3、轻量级虚拟化原理与架构

- 轻量级虚拟化Docker架构



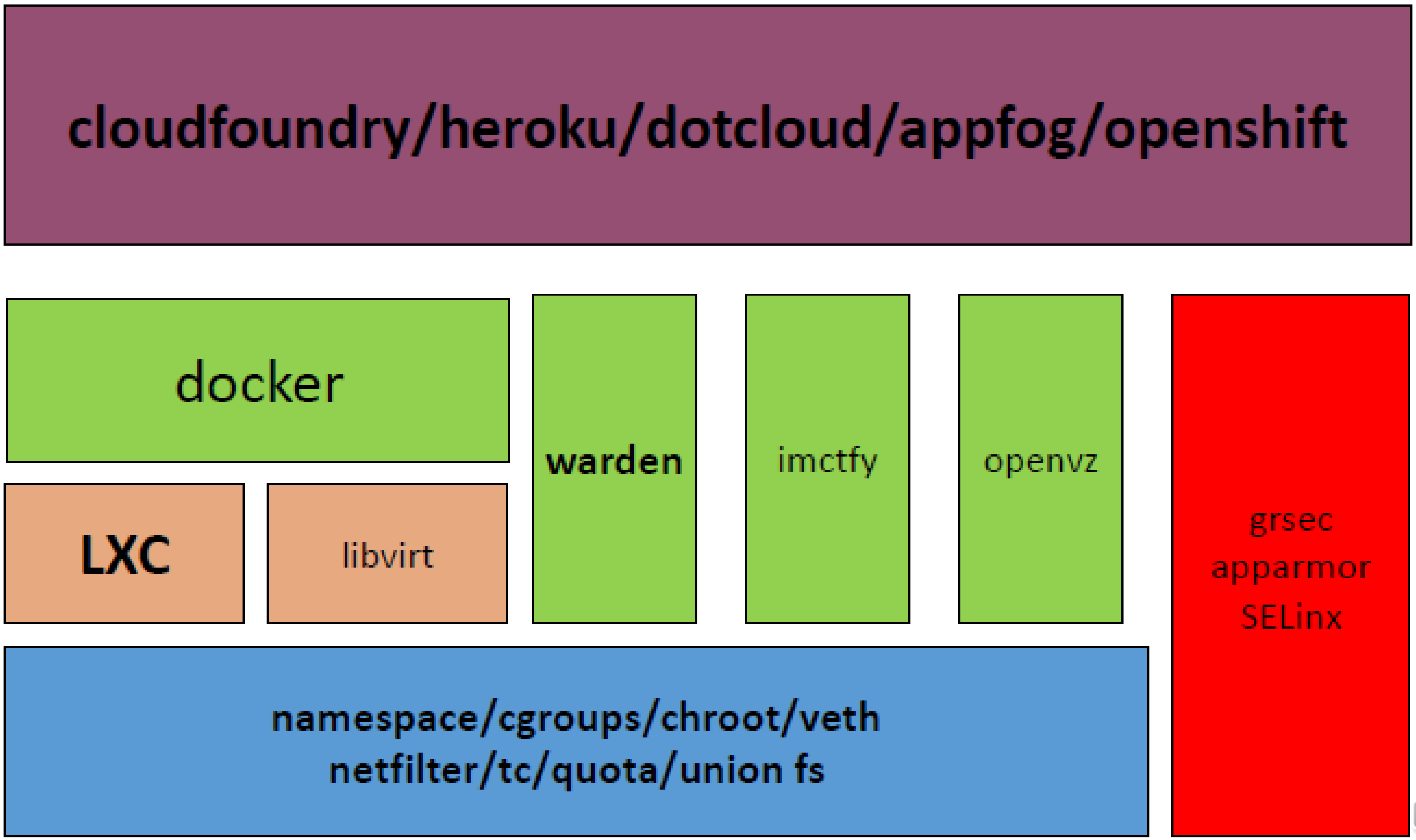
3、轻量级虚拟化原理与架构

轻量级虚拟化技术—相关开源技术和开源项目

- 底层技术：
  - namespace/cgroups
  - chroot/veth
  - union fs(AUFS)
  - netfilter/tc/quota
- container管理
  - LXC/libvirt
- 安全相关
  - grsec/apparmor/SELinux
- 高级
  - docker/warden/lmctfy/openVZ

3、轻量级虚拟化原理与架构

- 轻量级虚拟化技术—相关开源技术和开源项目



4、轻量级虚拟化特征与价值

●虚拟化特征

从虚拟化方法的四个方面：隔离性、可配额/可度量、便携性、安全性来详细介绍Docker的技术特征。

4.1. 隔离性: Linux Namespace (ns)

每个用户实例之间相互隔离, 互不影响。 一般的硬件虚拟化方法给出的方法是VM，而LXC给出的方法是container，更细一点讲就是kernel namespace。其中pid、net、ipc、mnt、uts、user等namespace将container的进程、网络、消息、文件系统、UTS("UNIX Time-sharing System")和用户空间隔离开。

4、轻量级虚拟化特征与价值

●虚拟化特征

从虚拟化方法的四个方面：隔离性、可配额/可度量、便携性、安全性来详细介绍Docker的技术特征。

4.2 可配额/可度量 - Control Groups (cgroups)

cgroups 实现了对资源的配额和度量。 cgroups 的使用非常简单，提供类似文件的接口，在 /cgroup目录下新建一个文件夹即可新建一个group，在此文件夹中新建task文件，并将pid写入该文件，即可实现对该进程的资源控制。groups可以限制blkio、cpu、cpuacct、cpuset、devices、freezer、memory、net_cls、ns九大子系统的资源

4、轻量级虚拟化特征与价值

●虚拟化特征

从虚拟化方法的四个方面：隔离性、可配额/可度量、便携性、安全性来详细介绍Docker的技术特征。

4.3 便携性：AUFS

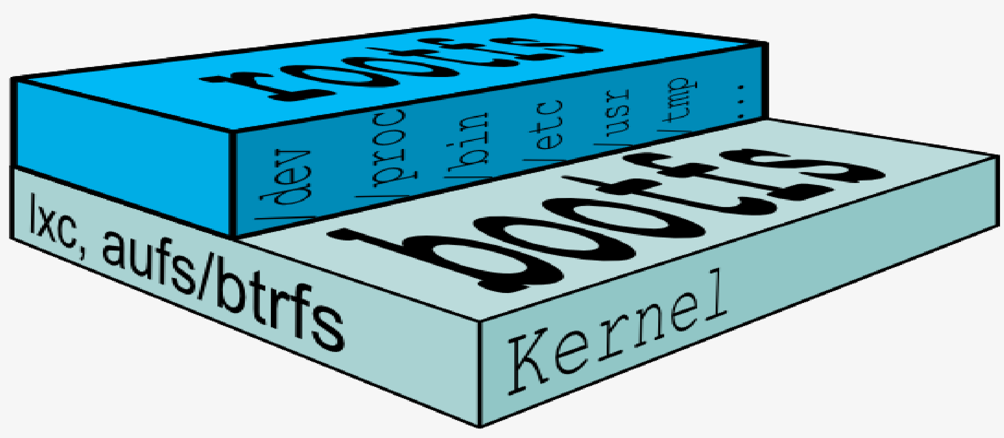
AUFS (AnotherUnionFS) 是一种 Union FS, 简单来说就是支持将不同目录挂载到同一个虚拟文件系统下(unite several directories into a single virtual filesystem)的文件系统, 更进一步的理解, AUFS支持为每一个成员目录(类似Git Branch)设定readonly、readwrite 和 whiteout-able 权限, 同时 AUFS 里有一个类似分层的概念, 对 readonly 权限的 branch 可以逻辑上进行修改(增量地, 不影响 readonly 部分的)。

4、轻量级虚拟化特征与价值

●虚拟化特征

从虚拟化方法的四个方面：隔离性、可配额/可度量、便携性、安全性来详细介绍Docker的技术特征。

典型的启动Linux运行需要两个FS：bootfs + rootfs：



4、轻量级虚拟化特征与价值

●虚拟化特征

从虚拟化方法的四个方面：隔离性、可配额/可度量、便携性、安全性来详细介绍Docker的技术特征。

4.4 安全性：AppArmor，SELinux，GRSEC

安全永远是相对的，这里有三个方面可以考虑Docker的安全特性：

1.由kernel namespaces和cgroups实现的Linux系统固有的安全标准;

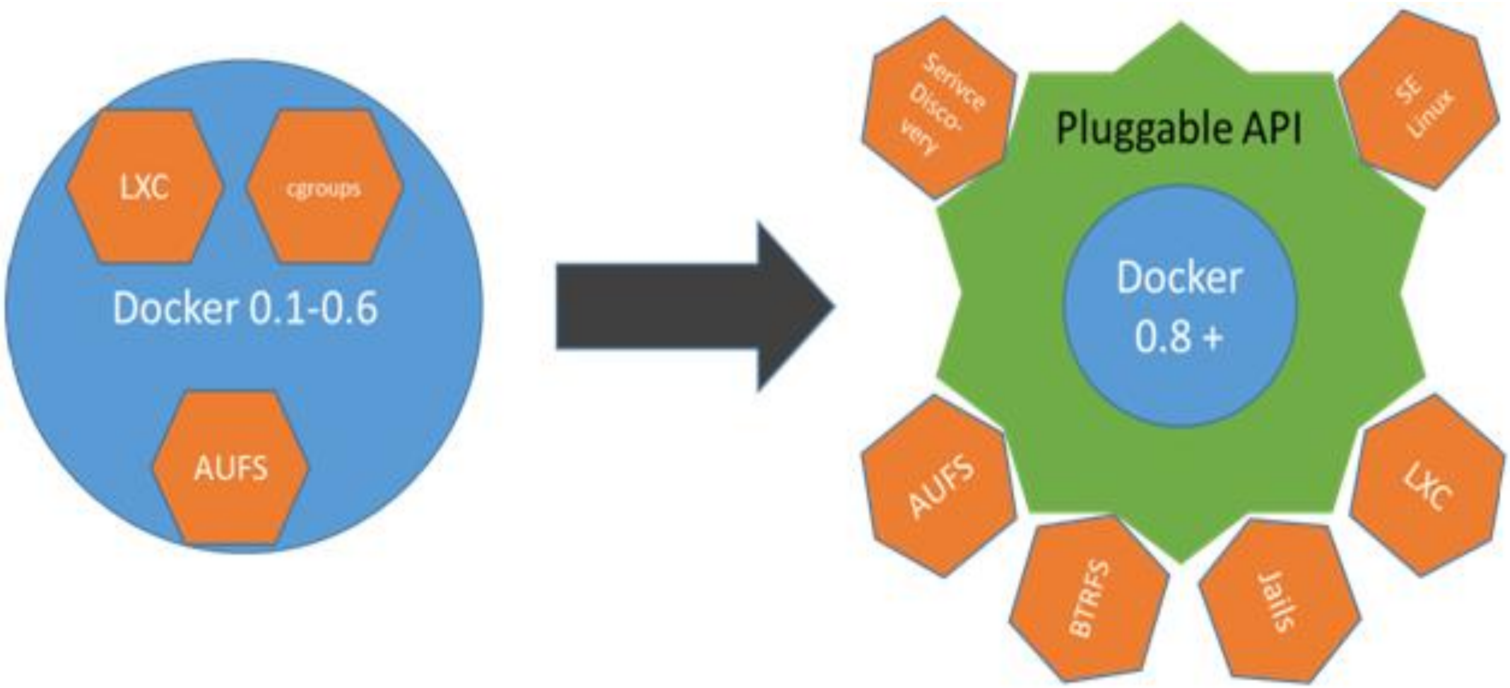
2.Docker Deamon的安全接口;

3.Linux本身的安全加固解决方案,类如AppArmor，SELinux;

5、轻量级虚拟化发展与应用

- 未来发展

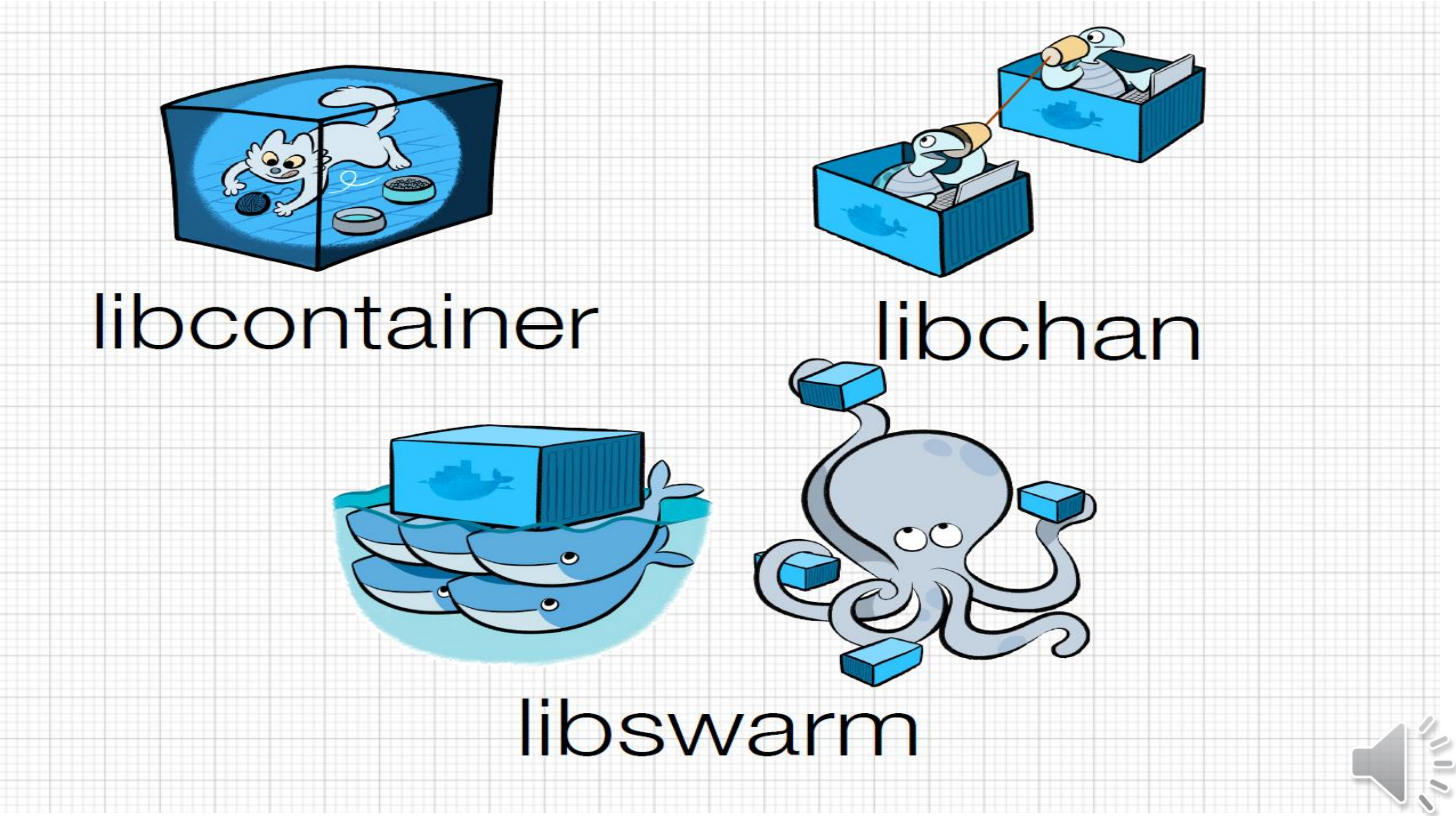
Docker 框架改成了插件式的架构，便于添加替换各个功能模块



5、轻量级虚拟化发展与应用

- 未来发展

核心项目：Libswarm、Libchan、Libcontainer



5、轻量级虚拟化发展与应用

- 未来发展

➢ docker将成为这个领域的领头羊

➢ Cloudfoundry, Openshift 将支持docker

➢ 基于docker的开源PaaS平台将诞生，且使用Go语言开发

➢ 越来越多的新兴开源项目将基于docker展开

5、轻量级虚拟化发展与应用

- 未来发展—存在的主要问题

➢ 安全

✓user namespace来解决root权限问题

✓SELinux增强安全性

➢ 支持更多的Linux发行版

➢ 由于AUFS的限制，目前仅支持ubuntu

➢ 支持更多的架构

➢ 目前仅支持x86-64

轻量级虚拟化总结

- 源于 LXC，但是比比 LXC 更便于重建、复制、移动，以及管理状态
- 不运行行时是软件包，运行行时是容器，比 VM 虚拟机更弹性更灵活可伸缩
- 过去部署程序是：装运行行环境，上传代码，初始化配置，运行行程序
- 现在部署程序是：把程序和配置及其上下文文环境整体打包，分发软件包，运行行软件包（容器）

参考资料

http://www.infoq.com/articles/docker-containers

http://en.wikipedia.org/wiki/Operating_system-level_virtualization

http://en.wikipedia.org/wiki/LXC

http://en.wikipedia.org/wiki/Cgroups

http://en.wikipedia.org/wiki/Linux_Containers

http://en.wikipedia.org/wiki/Aufs

http://libvirt.org/

http://linuxcontainers.org/

https://wiki.ubuntu.com/LxcSecurity

http://lwn.net/Articles/236038/