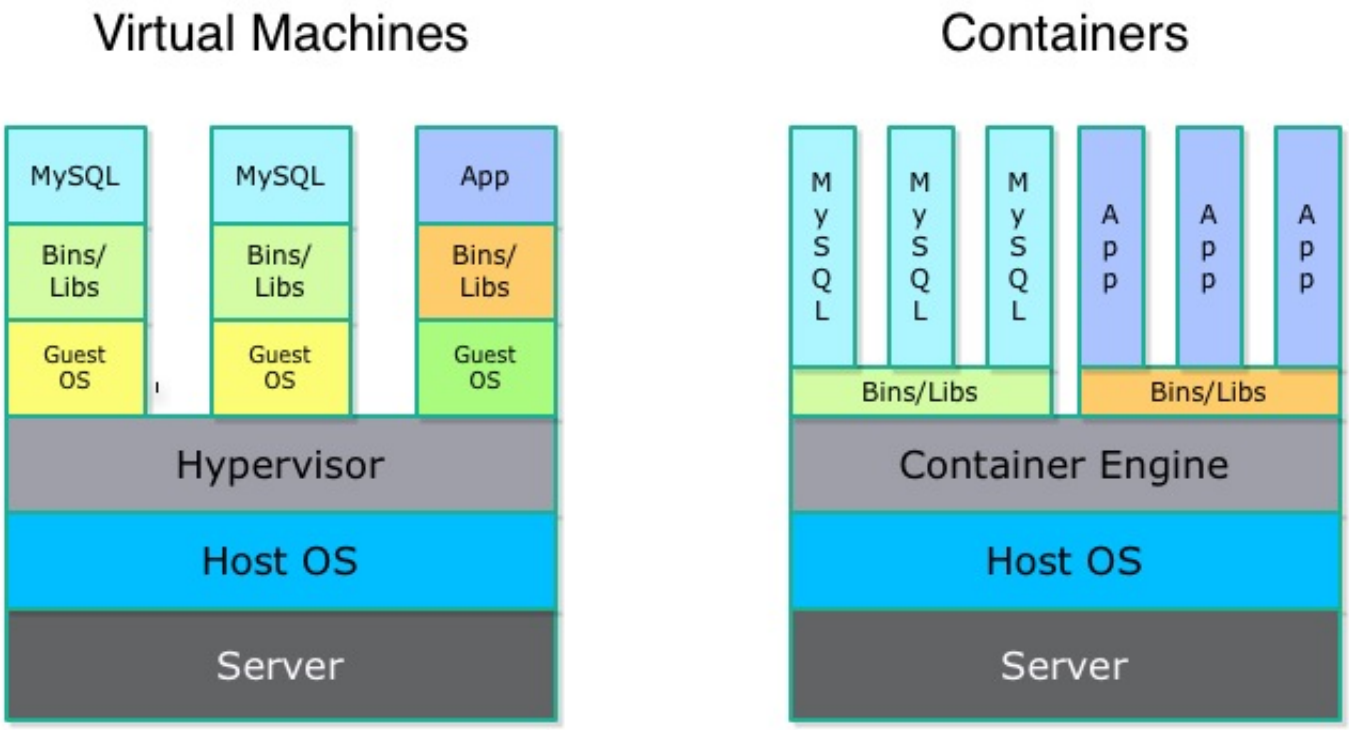
Problem 3

1352978 施闻轩

资源隔离和管理 / 轻量级虚拟化

容器技术是一种轻量级虚拟化。与全虚拟化不同的是，轻量级虚拟化是利用操作系统提供的功能，在操作系统以上为应用程序建立隔离环境。



Linux

在 Linux 操作系统中，诸如 LXC（Docker 早期使用的轻量级虚拟化方案）、libcontainer（目前 Docker 使用的轻量级虚拟化方案）、OpenVZ 等轻量级虚拟化均以 Linux 内核提供的 namespace 和 cgroup 等功能为核心。

1. namespaces（资源隔离）

Linux 内核的 namespace 功能提供了内核级别资源隔离功能，可以隔离的资源包括进程、网络、用户等。具体来说 Linux 有以下六种 namespace：

Namespace	隔离内容
UTS	主机名与域名
IPC	进程间通信如信号量、消息队列和共享内存
PID	进程
Network	网络设备、网络栈、端口等
Mount	挂载点（文件系统）
User	用户和用户组

namespace 技术被 Docker 和 OpenVZ 所使用。其中，Docker 没有完全利用 User Namespace，而仅仅使用了其提供的 capabilities 机制，对 `root` 用户进行限制，使得隔离环境下的 `root` 用户无法操作内核或修改 namespace 参数以保障安全。

2. cgroups（资源管理）

Linux 内核的 cgroups (Control Groups) 功能提供了内核级别资源管理功能，可以限制、记录、隔离进程组所使用的物理资源（包括 CPU、Memory、IO 等）。具体来说 cgroups 提供了以下四大功能：

功能	描述
资源限制	对进程组使用的资源总额进行限制。如设定应用运行时使用内存的上限
优先级分配	控制分配的 CPU 时间片数量及硬盘 IO 带宽大小
资源统计	统计系统的资源使用量，如 CPU 使用时长、内存用量等
进程控制	对进程组执行挂起、恢复等操作

LXC 和 libcontainer 使用 cgroups 进行资源管理。

3. OpenVZ 资源管理

OpenVZ 通过内核补丁的方式自己实现了适用于容器的资源管理，包括容器级别资源管理和计费。可被管理的资源主要有：

资源	涵盖
CPU	可用的 CPU、可用的 CPU 时间等
Disk	磁盘空间、优先级、带宽等
Memory	内存空间、进程数等

Windows NT

Windows NT 操作系统中虽然没有容器的成熟时限，但内核已包含实现容器所需的各项技术，可以实现基本的安全隔离和资源限制（即沙盒）。注意，Windows 10 中新引入的面向 UWP 应用程序的 Container 技术由于太新不在本文讨论之列。

1. 特权令牌（权限控制）

Windows 操作系统实现了 ACL，可以控制各个用户对各个对象（进程、线程、桌面、桌面站、互斥体、计时器、管道、窗口、菜单等）的权限。

用户权限包含：

- 登录本地系统
- 登录为服务
- 修改时间
- 关机
- 修改配额
- 管理进程权限
- 其他

进程权限包含：

- 创建进程
- 创建线程
- 查询某个进程的信息
- 暂停或终止进程
- 虚拟内存控制
- 等待
- 其他

2. Desktop & Window station（窗口隔离）

Windows 图形窗口是由内核提供的，因此有特别的技术进行窗口隔离。Windows 下不同桌面和不同工作站之间是天然隔离的。注，每个登录用户都有自己的桌面。

3. Job Objects（资源管理）

Windows 中每一个进程都可以被关联 Job Objects，可以实现资源管理和计费。可被限制和计费的资源有：

类别	涵盖
Basic	CPU 时间限制、内存使用限制、优先级、进程数等
UI	桌面对象、窗口对象、剪贴板、显示器等
CPU Rate	CPU 占用率
Extended	虚拟内存使用限制等
Notification	I/O 字节数等

4. NTFS based ACL & Quota（文件系统资源管理）

Windows NTFS 文件系统提供了针对用户的访问控制和配额限制，访问控制包括各个用户对各个文件的读写权限，配额限制则是各个用户对文件系统的空间配额限制。

文件系统隔离 (Linux)

1. chroot

chroot 可以将一个目录作为根目录，从而实现文件系统访问隔离。这是最简单的隔离。

2. AUFS

AUFS 可以将不同物理位置的目录合并挂载到一个目录中，在 Docker 中实现跨容器的文件共享并隔离。

3. OverlayFS

OverlayFS 与 AUFS 类似，可以将不同目录合并挂载到一个目录下。

热迁移 (Linux)

1. CRIU

CRIU 可冻结（建立 Checkpoint）一个应用程序执行，保存执行状态到文件中，并在随后从文件中恢复程序执行。Docker 和 OpenVZ 利用 CRIU 实现容器热迁移。

Docker 容器管理

Kubernetes

Kuberenetes 是由 Google 开源的一个适用于集群的 Docker containers 管理工具。用户可以将一组 containers 以 “POD” 形式通过 Kubernetes 部署到集群之中。Kubernetes 以 “POD” 为单位管理一系列彼此联系的 Containers，这些 Containers 被部署在同一台物理主机中、拥有同样地网络地址并共享存储配额。[1]

flannel

flannel (rudder) 是 CoreOS 团队针对 Kubernetes 设计的一个覆盖网络 (overlay network) 工具，其目的在于帮助每一个使用 Kuberentes 的 CoreOS 主机拥有一个完整的子网。Kubernetes 会为每一个 POD 分配一个独立的 IP 地址，这样便于同一个 POD 中的 Containers 彼此连接，而之前的 CoreOS 并不具备这种能力。为了解决这一问题，flannel 通过在集群中创建一个覆盖网格网络 (overlay mesh network) 为主机设定一个子网。[1]

基于容器的交付部署

Docker 可以解决运行环境的复用问题，实现运行环境的复用和自动化交付部署。

通过结合轻量型应用程序隔离和基于映像的部署方法，Linux 软件容器将应用程序和它们的运行时组件保持在一起。 容器将应用程序和库及其运行所需的其它二进制文件打包在一起，从而为应用程序提供了一个独立的操作系统环境。这样有效避免了那些依靠底层宿主机操作系统关键组件的应用程序之间存在的冲突。由于软件容器不包含 (OS) 内核，这使得它们比虚拟机更加快速和灵活。[2]

应用开发人员

- 版本质量更高
- 应用可扩展性更强
- 应用隔离效果更好

（来源于 [2]）

IT 架构师

- 横向扩展速度更快
- 测试周期更短
- 部署错误更少

（来源于 [2]）

IT 运营人员

- 版本质量更高
- 更高效地替换生产环境中的全部虚拟机
- 应用管理更便捷

（来源于 [2]）

Reference

[1] http://www.infoq.com/cn/articles/what-is-coreos

[2] https://www.redhat.com/zh/insights/containers