

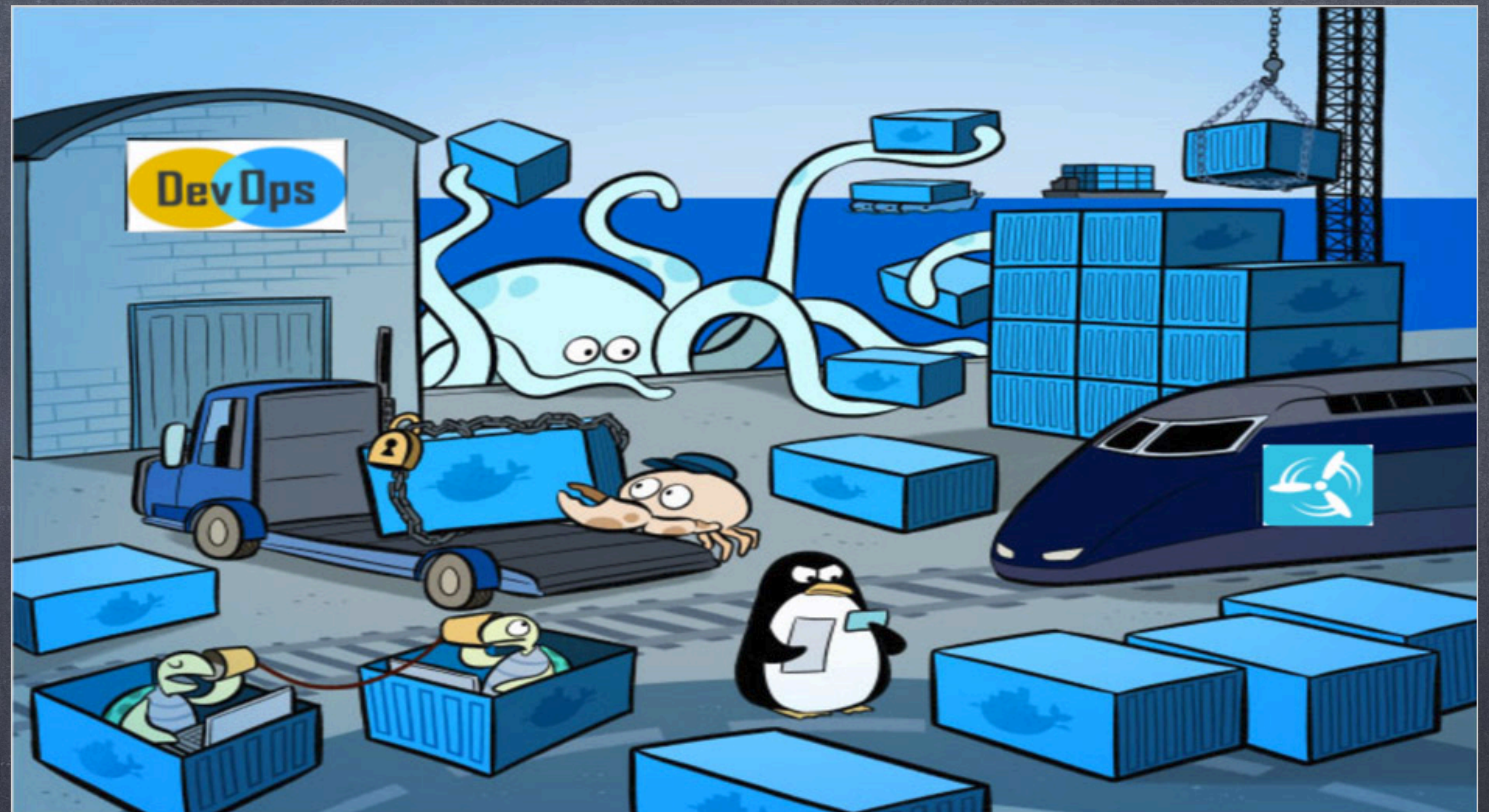
微服务入门与Docker实战

京程一灯

<http://www.yidengxuetang.com>

Docker入门与实战

- 微服务的核心思想
- Docker vs VM
- Docker的相关概念
- 安装与Hello world
- 常用命令与Dockerfile
- Docker Hub
- 实战：打包一个WEB服务器

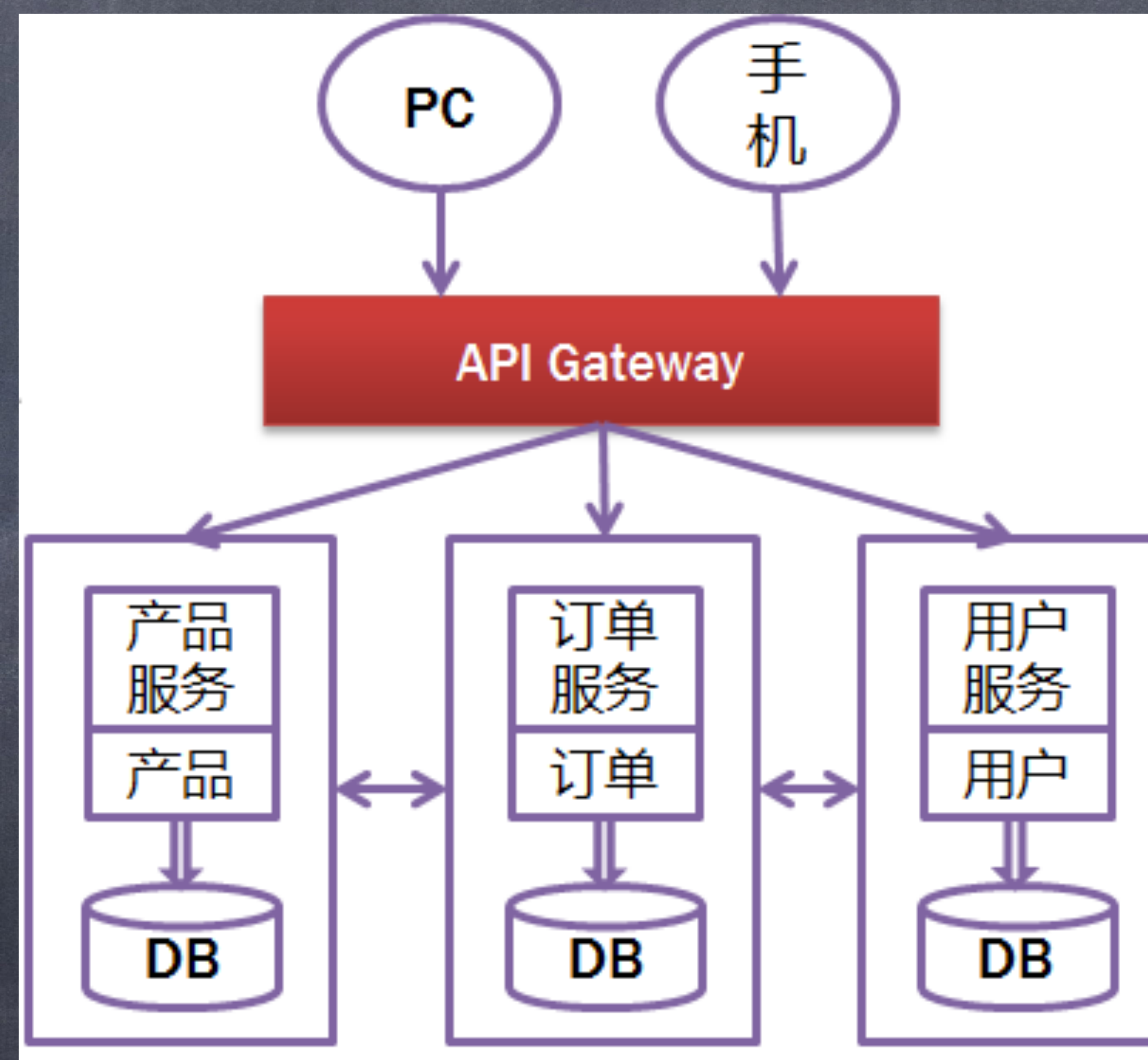
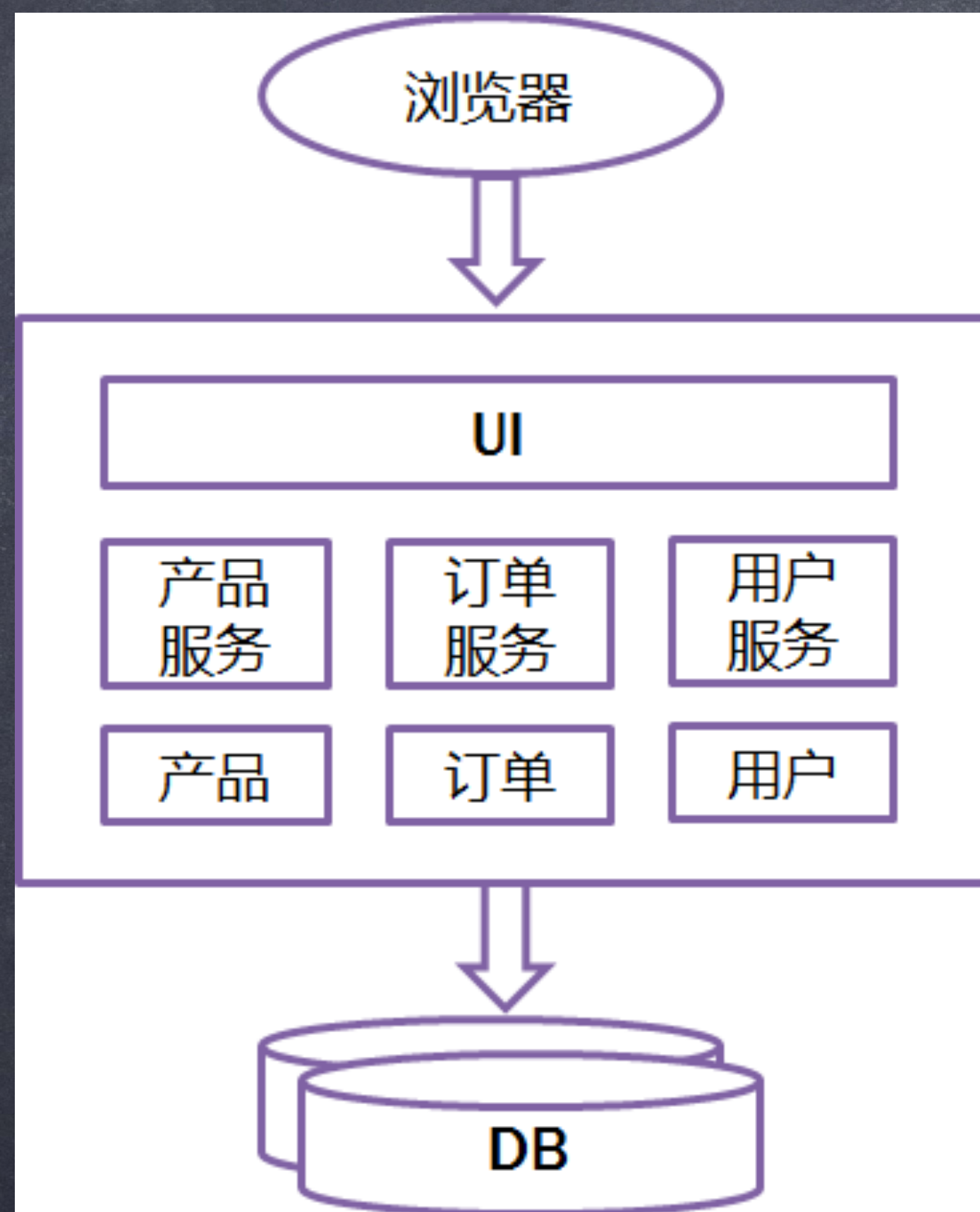


微服务到底是什么

- 微服务属于架构层面的设计模式
- 微服务的设计概念以业务功能为主
- 微服务独立提供对应的业务功能
- 微服务不拘泥于具体的实现语言
- 微服务架构 \approx 模块化开发 + 分布式计算

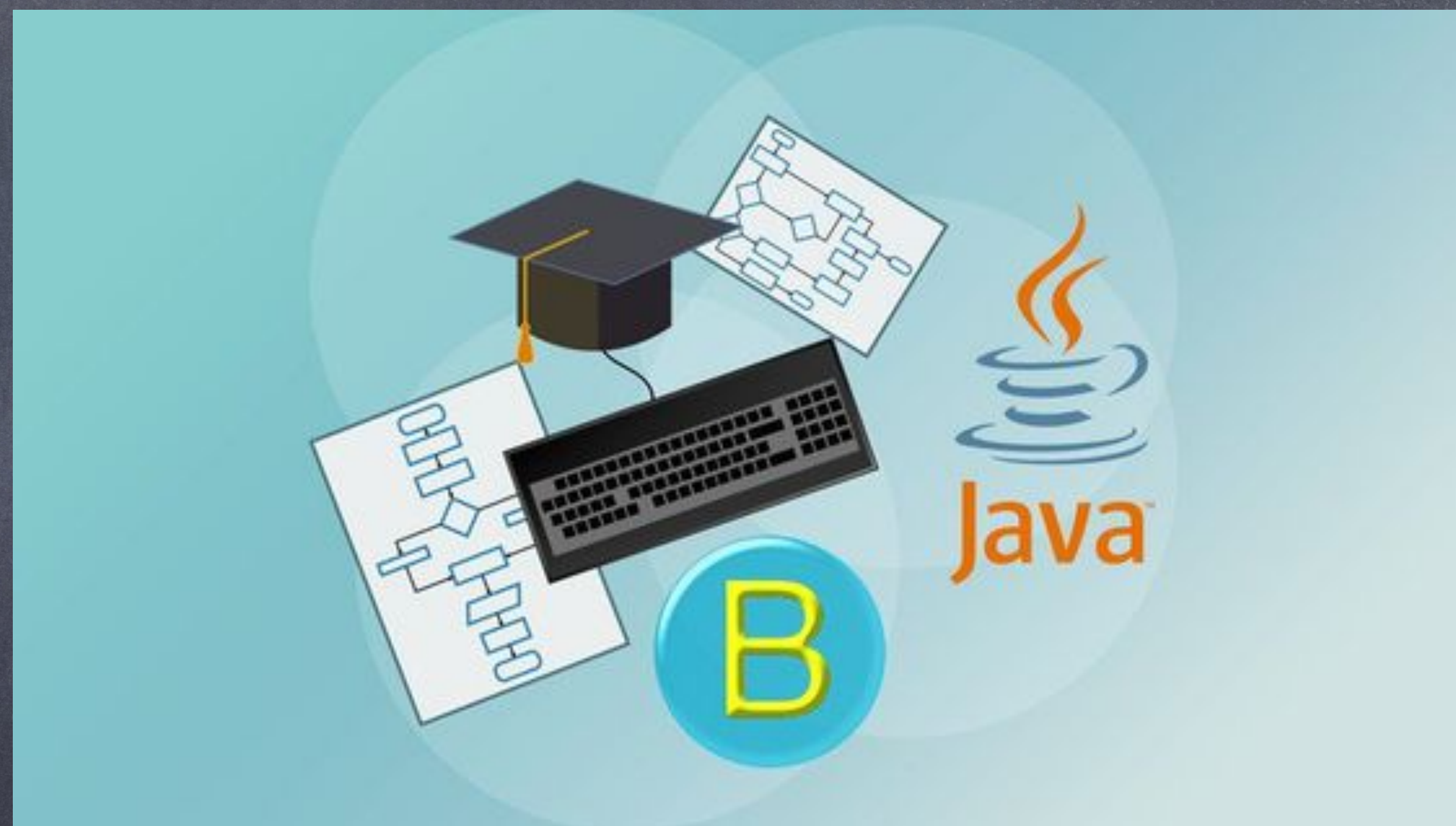


总揽微服务



微服务的特点

- 小, 且专注于做一件事情
- 处于独立的进程中
- 轻量级的通信机制
- 松耦合、独立部署



合理使用微服务

- 业务复杂度高
- 团队规模大
- 业务需要长期演进
- 最后——没有银弹



微服务——集成与部署

- 持续集成——jenkins
- 虚拟化——虚拟机
- 容器——Docker



Docker简介

- Docker是一个开源的引擎，可以轻松地为任何应用创建一个轻量级的、可移植的、自给自足的容器。开发者在笔记本上编译测试通过的容器可以批量地在生产环境中部署，包括VMs（虚拟机）、bare metal、OpenStack 集群和其他的基础应用平台。
- Docker通常用于如下场景：
 - web应用的自动化打包和发布；
 - 自动化测试和持续集成、发布；
 - 在服务型环境中部署和调整数据库或其他的后台应用；
 - 从头编译或者扩展现有的OpenShift或Cloud Foundry平台来搭建自己的PaaS环境。

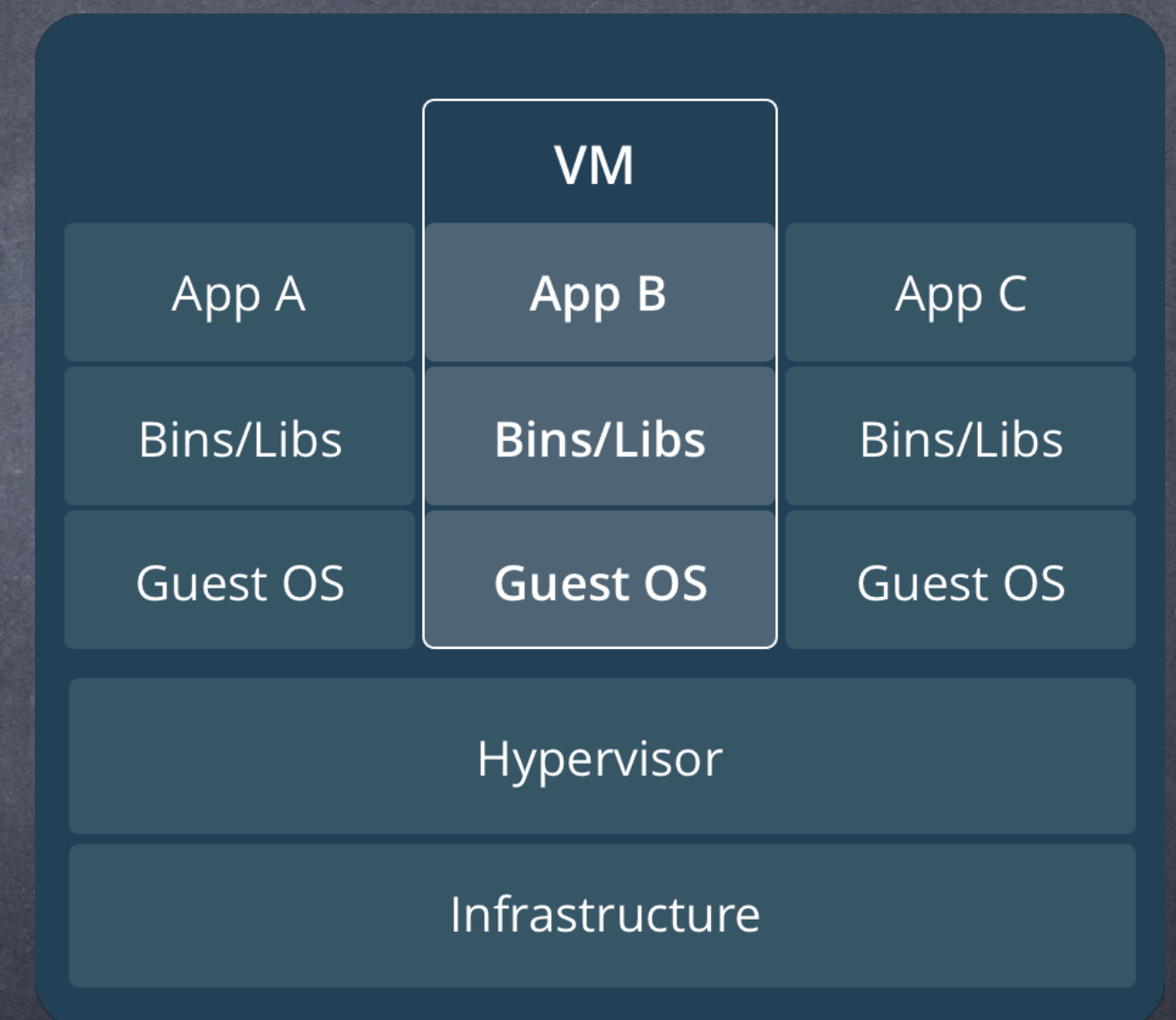
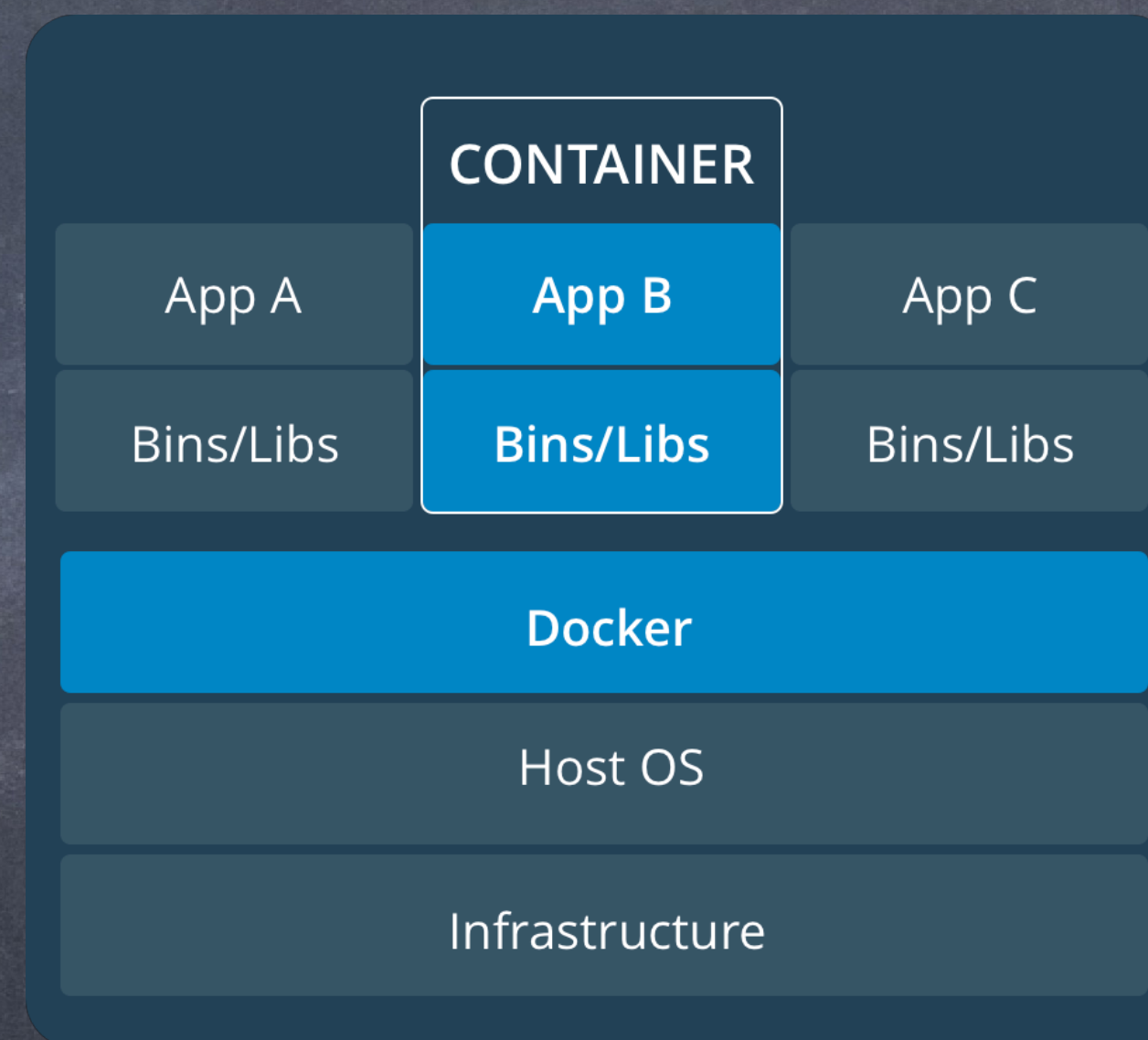
Docker vs. VM

- VM :

- 运行在宿主机之上的完整的操作系统
- 运行自身操作系统会占用较多的资源

- Docker :

- Docker更加轻量高效
- 对系统资源的利用率很高
- 比虚拟机技术更为轻便、快捷
- 隔离效果不如VM

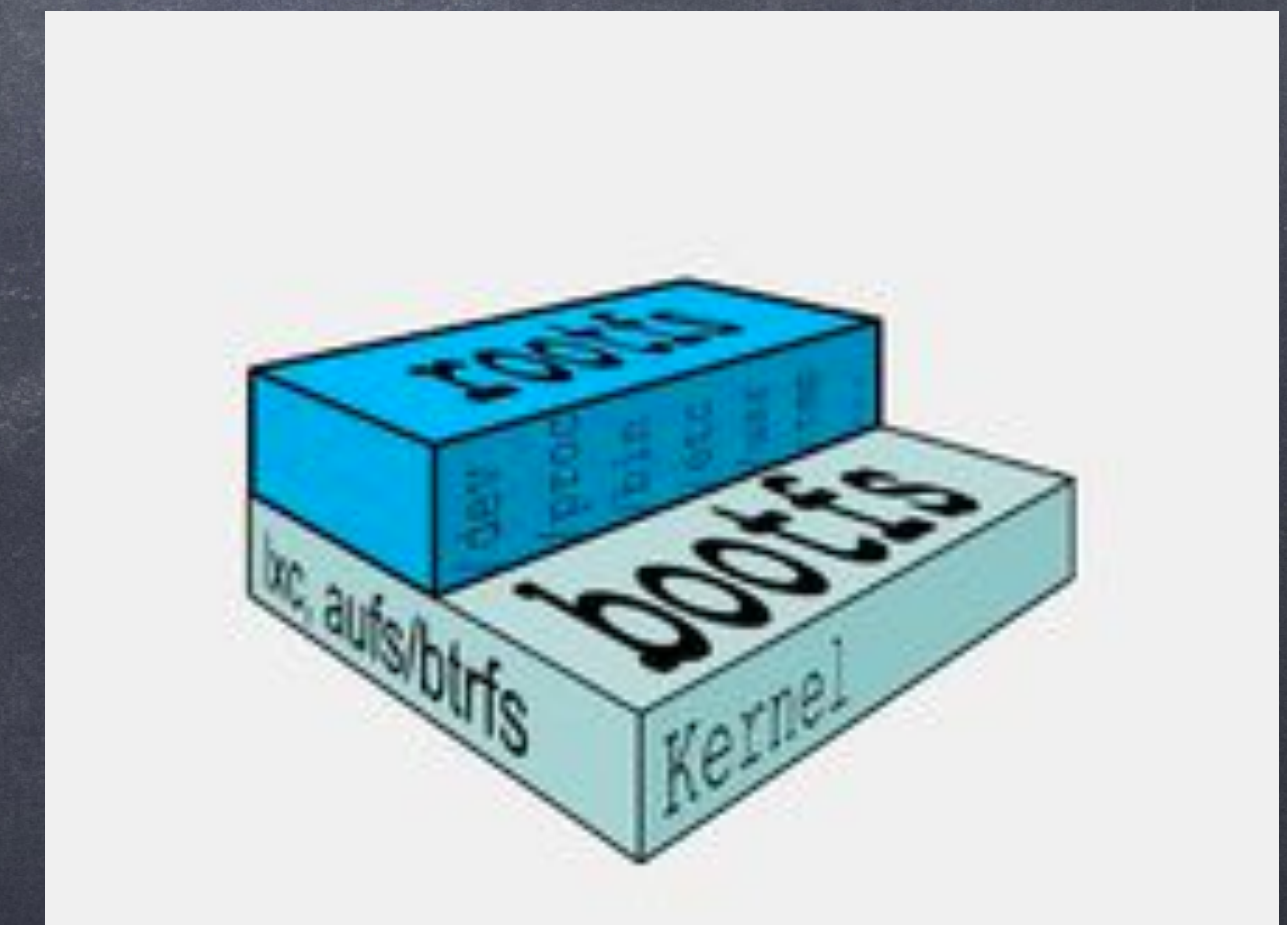
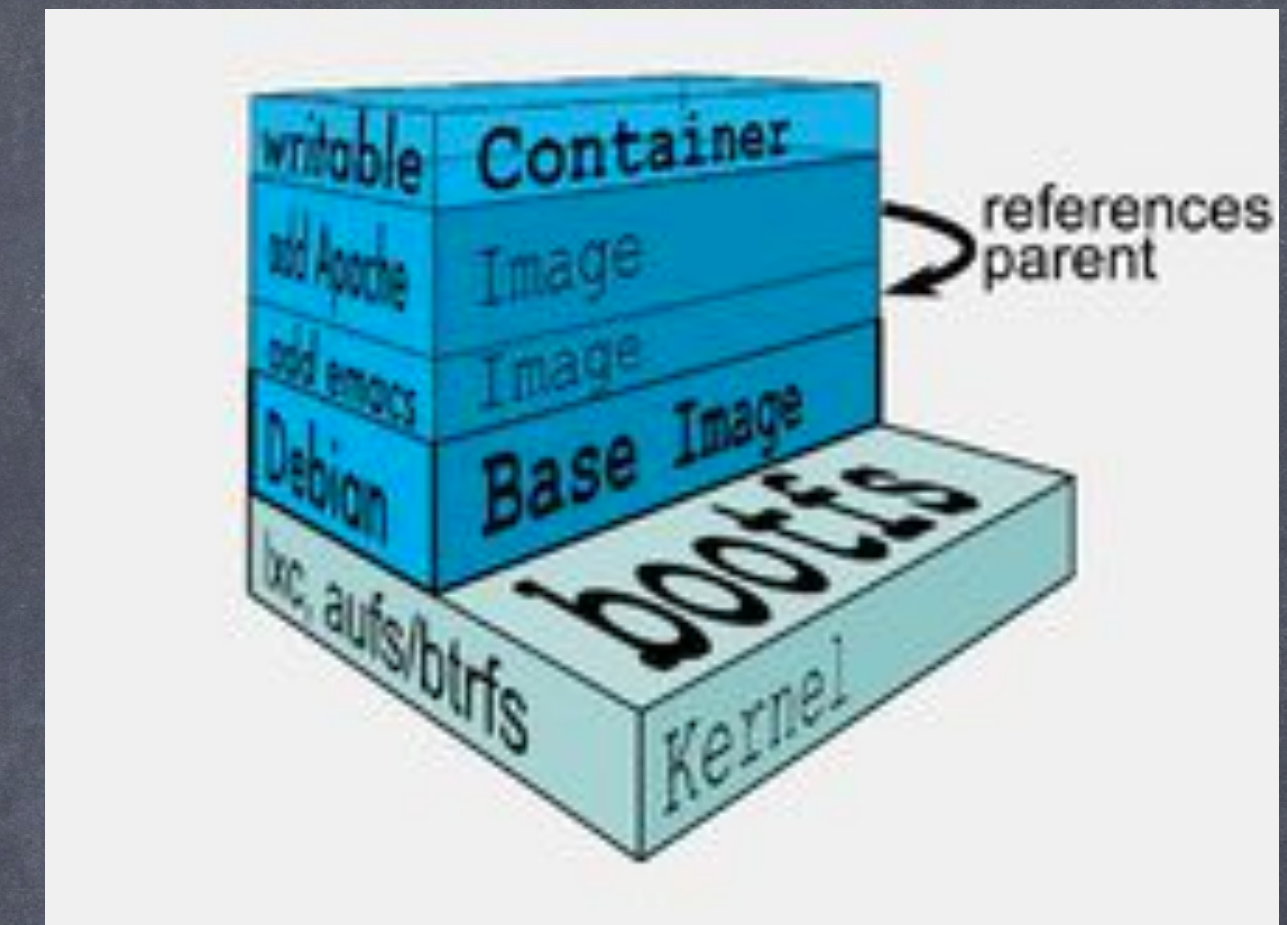


Docker相关概念

- Docker是CS架构，主要有两个概念：
- Docker daemon:
 - 运行在宿主机上
 - Docker守护进程
 - 用户通过Docker client(Docker命令)与Docker daemon交互
- Docker client:
 - Docker 命令行工具，是用户使用Docker的主要方式
 - Docker client与Docker daemon通信并将结果返回给用户
 - Docker client也可以通过socket或者RESTful api访问远程的Docker daemon

相关概念

- Docker是CS架构，主要有两个概念：
- Docker daemon:
 - 运行在宿主机上
 - Docker守护进程
 - 用户通过Docker client(Docker命令)与Docker daemon交互
- Docker client:
 - Docker 命令行工具，是用户使用Docker的主要方式
 - Docker client与Docker daemon通信并将结果返回给用户
 - Docker client也可以通过socket或者RESTful api访问远程的Docker daemon



Dockerfile

- Dockerfile 概念
- Dockerfile 文件格式
- 构建镜像
- 镜像标签
- 修改容器内容

