

## 发展历史

emergence { 1979 : Unix v7 — chroot system (first glimpse)  
2000 : FreeBSD Jails , introduce jail system

development { 2006 : cgroups were introduced  
2008 : cgroups built in LXC

golden age { 2013 : Docker  
2014 : Kubernetes  
2016 : Linux containers can be natively run on Windows computers

背景: 物理机时代 → 虚拟机时代 → 容器化时代

概念: code + dependencies + operating environment.  
↓  
真实物理机      ↓      操作系统级虚拟/隔离      ↓      进程级虚拟/隔离

特点: 1. 轻量化: 只打包了必要的 Bin/Lib

2. 部署快: 毫秒级到秒级部署速度

与虚拟机  
也可以形成  
一定的对比  
3. 易于移植: "Build once, run anywhere"

4. 弹性伸缩: 根据业务需求和策略, 经济地自动调整弹性计算资源.

## 技术. 相关概念

- (1) LXC : ① cgroups , ② namespace (Container 共有的两个基本技术, LXC 最早).
- (2) Docker : 一开始用了 LXC, 后来发展了自己的 libcontainer ,  
①  
② image 镜像  
③ container 容器  
④ repository 仓库  
⑤ Docker 架构  
} "Build, ship and run."  
三板斧  
→ Runc
- (3) Kubernetes (K8S) : 编排、管理和调度多个容器

## 标准化

OCI 组织 (Open Container Initiative)

2016.4 : 第一个开放容器标准

{ 运行时标准 runtime spec  
镜像标准 image spec  
分发标准 distribution spec

## 应用

- (1) 云原生 : (定义) Applications adopting principles of microservices packaged as containers ... developed using practices such as CI/CD and DevOps.

Microservices : 微服务架构

演进 : 单体 (monolithic) → 微服务 (micro services)

## ② CI/CD 的构建

CI 通常是完全自动化的，通过 `git` 命令推送并触发测试，测试成功后自动构建新镜像，然后推送至 Docker 镜像库。通过后续的预演脚本，可以将新镜像的容器部署到预演环境，从而进行进一步测试。

## ③ IT 基础设施优化。

展望：

???

可以改进的不足之处，

可能还会用在哪些领域（如移动通信等），

.....