1. 什么是容器：

硬件抽象层基于hypervisor的虚拟化方式可以最大程度上提供虚拟化管理的灵活性。各种不同操作系统的虚拟机都能通过hypervisor（KVM、XEN等）来衍生、运行、销毁。然而对于hypervisor环境来说，每个虚拟机都需要运行一个完整的操作系统以及其中安装好的大量应用程序。但实际生产开发环境里，我们更关注的是自己部署的应用程序，如果每次部署发布我都得搞一个完整操作系统和附带的依赖环境，这让任务和性能变得很重和很低下。

基于上述情况，为了能让人更加的关注应用程序本身，底层多余的操作系统和环境可以共享和复用，诞生了容器技术，它是一种内核轻量级的操作系统层虚拟化技术。Linux Container主要由Namespace和Cgroup两大机制来保证实现。Namespace做隔离，Cgroup就负责资源管理控制作用，比如进程组使用CPU/MEM的限制，进程组的优先级控制，进程组的挂起和恢复等等。

1. 特点：

* 极其轻量：只打包了必要的Bin/Lib；
* 秒级部署：根据镜像的不同，容器的部署大概在毫秒与秒之间（比虚拟机强很多）；
* 易于移植：一次构建，随处部署；
* 弹性伸缩：Kubernetes、Swam、Mesos这类开源、方便、好使的容器管理平台有着非常强大的弹性管理能力。

1. 标准化

OCI组织推出，标准主要包括runtime运行时标准和image镜像标准。

1. 应用https://www.cnblogs.com/qcloud1001/p/9273549.html
   1. 传统应用：提高安全性、可移植性、节约成本
   2. 持续集成和持续部署 (CI/CD)的提速

现代化开发流程快速、持续且具备自动执行能力，最终目标是开发出更加可靠的软件。通过持续集成 (CI) 和持续部署 (CD)，每次开发人员签入代码并顺利测试之后，IT 团队都能够集成新代码。作为开发运维方法的基础，CI/CD 创造了一种实时反馈回路机制，持续地传输小型迭代更改，从而加速更改，提高质量。CI 环境通常是完全自动化的，通过 git 推送命令触发测试，测试成功时自动构建新镜像，然后推送到 Docker 镜像库。通过后续的自动化和脚本，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。

* 1. 微架构
  2. IT基础设施优化

1. 历史

应用部署演化：物理机时代——虚拟机时代——容器化时代

其各有特点 <https://blog.csdn.net/LiushaoMr/article/details/104326217>

各时代局限性： https://zhuanlan.zhihu.com/p/133514354

1. 容器发展趋势：<https://zhuanlan.zhihu.com/p/345100482>
2. 架构：https://www.zhihu.com/question/268294770/answer/2347813311

Docker 是 PaaS 提供商 DotCloud 开源的一个高级容器引擎，源代码托管在 Github 上，基于Go语言并遵从Apache2.0协议开源。Docker相当于物理行业的集装箱对物流的影响一样，成为Container上运行镜象的统一打包和交换的标准。

Docker使用了容器的环境隔离和资源限制技术，把镜像和运行环境打包到Image中。Register支持容器上传和下载功能。Docker同时提供了Build，Ship和Run，运维只需要在环境重配置好Docker，剩下的工作就是部署容器，实现Build Once Run Anywhere和Configure Once Run Anything；从而促进了容器技术的爆发。

在架构上，Docker采用Client Server模式和插件式架构设计，Docker的后端采用非常松耦合的架构，模块之间相互独立，用户通过Docker Client与Docker Daemon建立通信，并发送请求给Docker Daemon。Docker Daemon提供Server功能接受Docker Client的请求；随后通过Engine执行Docker内部的一系列工作，每项工作都是以一个Job的形式的存在。

Docker讲底层容器运行时剥离出来，实现更好的平台无关性。LibContainer是对各种容器的抽象，发展为RunC，并贡献给OCP组织作为定义容器环境的标准。Docker容器的三大编排工具，Compose、Swarm和Machine。Compose是服务编排工具，是定义和运行Docker主机上多容器应用的工具，通过单独文件，定义多容器应用并运行容器。

Docker的网络技术和能力一直是容器技术中最难、也是最不看好的技术之一，Libnetwork是Docker公司正在开发的新的网络底层架构，由[libcontainer](https://www.zhihu.com/search?q=libcontainer&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A2347813311%7D" \t "_blank)和Docker Engine中的网络相关的代码合并而成。Libnetwork的目标是引入了容器网络模型(CNM)，并为应用程序提供一致的编程API接口以及网络抽象。 Libnetwork的容器网络模型包含了三个重要概念，Network Sandbox，Endpoint和Network。

Weave创建了Networking Plugin技术，目前成熟的有Networking Plugin和Volume Plugin。Weave方案包含两大组件，用户态Shell脚本和Weave虚拟路由容器。Weave虚拟路由容器需要在每个宿主机上布置第三方插件，把不同宿主机的Route容器连接起来，使得Docker工具生态无缝集成到Docker。