Docker容器的原理、特征、基本架构、与应用场景

题目难度：★★★

知识点标签：Docker

学习时长：15分钟

题目描述

面试官：说一下你对docker的了解

什么是容器

一句话概括容器：容器就是将软件打包成标准化单元，以用于开发、交付和部署。

- 容器镜像是轻量的、可执行的独立软件包 ，包含软件运行所需的所有内容：代码、运行时环境、系统工具、系统库和设置。
- 容器化软件适用于基于Linux和Windows的应用，在任何环境中都能够始终如一地运行。
- 容器赋予了软件独立性，使其免受外在环境差异（例如，开发和预演环境的差异）的影响，从而有助于减少团队间在相同基础设施上运行不同软件时的冲突。

再来看看容器较为通俗的解释：

如果需要通俗的描述容器的话，我觉得容器就是一个存放东西的地方，就像书包可以装各种文具、衣柜可以放各种衣服、鞋架可以放各种鞋子一样。我们现在所说的容器存放的东西可能更偏向于应用比如网站、程序甚至是系统环境。

什么是Docker

Docker 是一个开源的应用容器引擎，基于 Go 语言 并遵从 Apache2.0 协议开源。Docker 可以让开发者打包他们的应用以及依赖包到一个轻量级、可移植的容器中，然后发布到任何流行的 Linux 机器上，也可以实现虚拟化。

容器是完全使用沙箱机制，相互之间不会有任何接口（类似 iPhone 的 app）,更重要的是容器性能开销极低

***为什么使用docker?**

1.快速，一致地交付您的应用程序

Docker 允许开发人员使用您提供的应用程序或服务的本地容器在标准化环境中工作，从而简化了开发的生命周期。

容器非常适合持续集成和持续交付（CI / CD）工作流程，请考虑以下示例方案：

- 您的开发人员在本地编写代码，并使用 Docker 容器与同事共享他们的工作。
- 他们使用 Docker 将其应用程序推送到测试环境中，并执行自动或手动测试。
- 当开发人员发现错误时，他们可以在开发环境中对其进行修复，然后将其重新部署到测试环境中，以进行测试和验证。
- 测试完成后，将修补程序推送给生产环境，就像将更新的镜像推送到生产环境一样简单。

2.响应式部署和扩展

Docker 是基于容器的平台，允许高度可移植的工作负载。Docker 容器可以在开发人员的本机上，数据中心的物理或虚拟机上，云服务上或混合环境中运行。

Docker 的可移植性和轻量级的特性，还可以使您轻松地完成动态管理的工作负担，并根据业务需求指示，实时扩展或拆除应用程序和服务。

3.在同一硬件上运行更多工作负载

Docker 轻巧快速。它为基于虚拟机管理程序的虚拟机提供了可行、经济、高效的替代方案，因此您可以利用更多的计算能力来实现业务目标。Docker 非常适合于高密度环境以及中小型部署，而您可以用更少的资源做更多的事情。

Docker 基本架构

Docker 使用客户端-服务器 (C/S) 架构模式，使用远程API来管理和创建Docker容器。

Docker 主要有以下几部分组成：

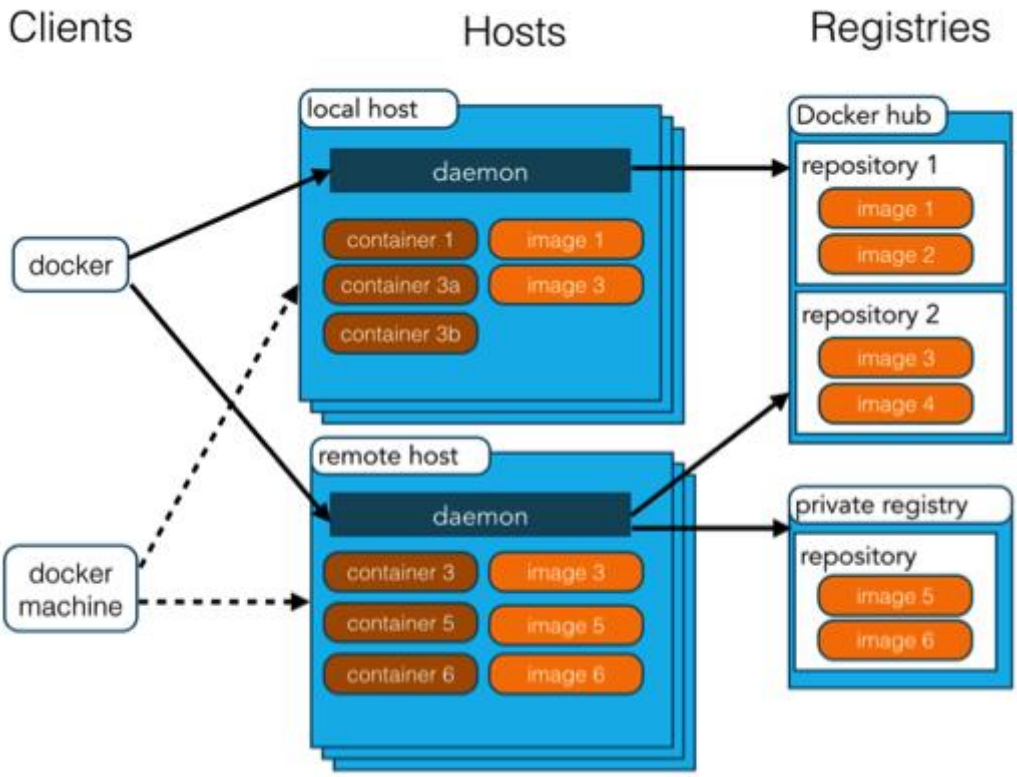
1. Docker Client 客户端
2. Docker daemon 守护进程
3. Docker Image 镜像
4. Docker Container 容器
5. Docker Registry 仓库

客户端和守护进程：

- Docker是C/S（客户端client-服务器server）架构模式。
- docker通过客户端连接守护进程，通过命令向守护进程发出请求，守护进程通过一系列的操作返回结果。
- docker客户端可以连接本地或者远程的守护进程。
- docker客户端和服务器通过socket或RESTful API进行通信。

Docker 容器通过 Docker 镜像来创建，容器与镜像的关系类似于面向对象编程中的对象与类。

如图所示基本架构：



1.Docker 镜像(Images)

Docker 镜像是用于创建 Docker 容器的模板。

2.Docker 容器(Container)

容器是独立运行的一个或一组应用。

3.Docker 客户端(Client)

Docker 客户端通过命令行或者其他工具使用 Docker API

4.Docker 主机(Host)

一个物理或者虚拟的机器用于执行 Docker 守护进程和容器。

5.Docker 仓库(Registry)

Docker 仓库用来保存镜像，可以理解为代码控制中的代码仓库。

6.Docker Hub

提供了庞大的镜像集合供使用。

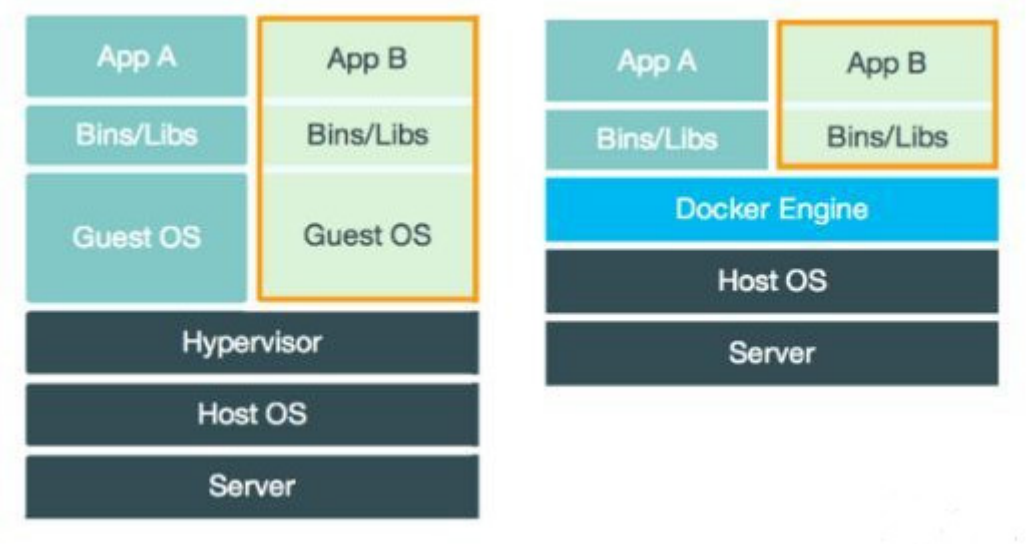
7.Docker Machine

Docker Machine是一个简化Docker安装的命令行工具，通过一个简单的命令行即可在相应的平台上安装Docker，比如VirtualBox、 Digital Ocean、Microsoft Azure。

总之，Docker是一种轻量虚拟化的容器技术，提供类似虚拟机的隔离功能，并使用了一种分层的联合文件系统技术管理镜像，能极大简化环境运维过程，最后看看对应的应用场景。

docker与虚拟机的区别

虚拟机也是一种虚拟化技术，它与Docker最大的区别在于它是通过模拟硬件，并在硬件上安装操作系统来实现。



Docker 是一个能把开发的应用程序自动部署到容器的开源引擎

虚拟机（Virtual Machine）指通过软件模拟的具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统。在实体计算机中能够完成的工作在虚拟机中都能够实现。在计算机中创建虚拟机时，需要将实体机的部分硬盘和内存容量作为虚拟机的硬盘和内存容量。每个虚拟机都有独立的CMOS、硬盘和操作系统，可以像使用实体机一样对虚拟机进行操作

特性	启动	磁盘使用	性能	系统支持量
容器	秒级	一般为MB	接近原生	单机支持上千个容器
虚拟机	分钟级	一般为GB	弱于	一般是几十个

(1) 虚拟机是在一台物理机上，利用虚拟化技术，虚拟出来多个操作系统，每个操作系统之间是隔离的。
docker是开源的应用容器引擎，依然需要先在电脑上安装操作系统，然后安装Docker容器管理器。

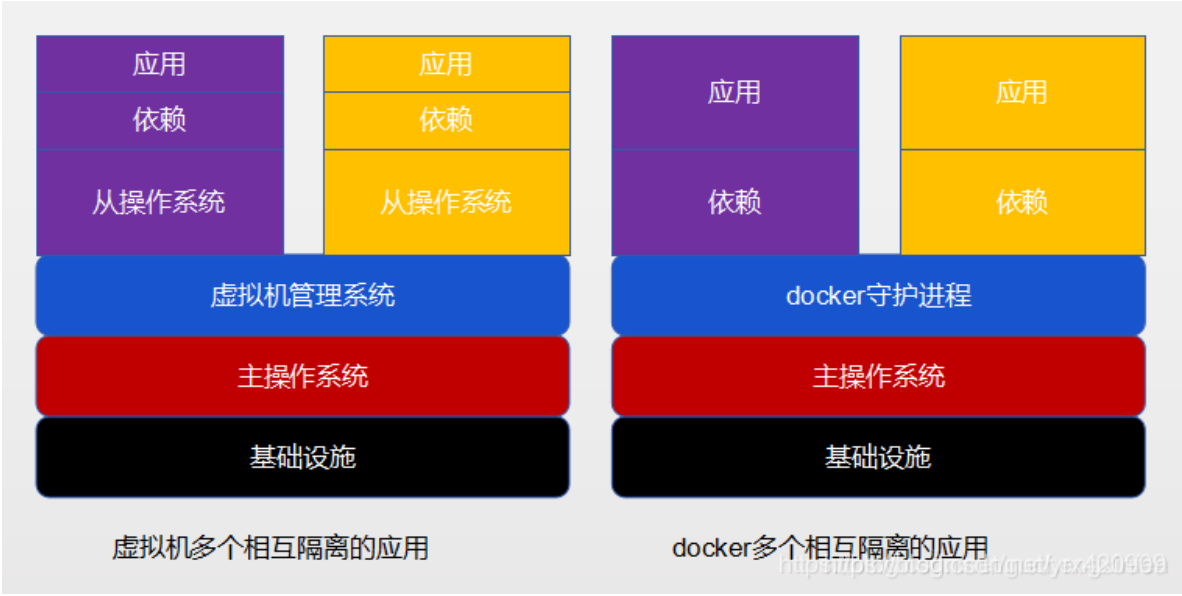
(2) 虚拟机是在硬件级别进行虚拟化,而docker是在操作系统的层面虚拟化

(3) 虚拟机是通过模拟硬件搭建操作系统,而docker则是复用操作系统

(4) 虚拟机实现了操作系统之间的隔离,docker只是进程之间的隔离,所以虚拟机的隔离级别更高,安全性更强

(5) docker的运行速度更快

(6) docker的文件要小的多,虚拟机要大



Docker守护进程可以直接与主操作系统进行通信，为各个Docker容器分配资源；它还可以将容器与主操作系统隔离，并将各个容器互相隔离。虚拟机启动需要数分钟，而Docker容器可以在数毫秒内启动。由于没有臃肿的从操作系统，Docker可以节省大量的磁盘空间以及其他系统资源。

比较点	Docker	虚拟机
操作系统	与宿主机共享OS	宿主机OS上运行虚拟机OS
存储大小	镜像小，利于存储和传输	镜像大
运行性能	几乎没有额外的性能损失	操作系统额外的CPU和内存消耗
移植性	轻便，灵活，适应于Linux	笨重，和虚拟机技术的耦合度高
硬件亲和性	面向软件开发人员	面向硬件运维人员

总结：

Docker的应用场景

1. 作为云主机使用

相比虚拟机来说，容器使用的是一系列非常轻量级的虚拟化技术，使得其启动、部署、升级跟管理进程一样迅速，用起来灵活又感觉跟虚拟机一样没什么区别，所以有些人直接使用Docker的Ubuntu等镜像创建容器，当作轻量的虚拟机来使用。

2.作为服务使用

如果你仅仅把Docker容器当作一个轻量的固定虚拟机用，那其实只能算是另类用法，Docker容器最重要价值在于提供一整套平台无关的标准化技术，简化服务的部署、升级、维护，只要把需要运维的各种服务打包成标准的集装箱，就可以在任何能运行Docker的环境下跑起来，达到开箱即用的效果，这个特点才是Docker容器风靡全球的根本原因。

- **Web应用服务**
- **持续集成和持续部署**

3. 微服务架构使用

如果说上面两种应用场景还不足以体现出与传统的PaaS平台相比的巨大优势的话，那么对微服务的架构这种复杂又灵活的使用场景的无缝支持绝对具有革命意义。

微服务架构将传统分布式服务继续拆分解耦，形成一些更小服务模块，服务模块之间独立部署升级，这些特性与容器的轻量、高效部署不谋而合。