Docker：容器时...

Docker 的发展历史

大家好，我是张晋涛，自 Docker 0.9 版本开始了自己的 Docker 之路后，到现在 Docker 19.03 版本也已经发布，大大小小的坑踩过无数个，也在实践和参与社区贡献的过程中积累了很多经验。

2013 年 Docker 第一次在 PyCon 上亮相，随后在 Hacker News 上引起了强烈的反响，之后在 GitHub 上开源，从此它正式开启了 Docker 的时代。

到现在无论我们在谈 Container、Docker、Kubernetes 或者这整个生态中的其他产品或技术，不可避免的都需要或多或少的 Docker 相关知识。

在近些年，也出来过不少的类 Docker 产品，比如说 rkt，曾经也有人说类似 rkt 这样的产品出来，将取代 Docker 的地位。但几年过去了 Docker 仍然是容器技术的首选，而 rkt 则已经因为项目失去活力被 CNCF 归档（当然不可否认 rkt 也曾为容器技术的发展做出了不少贡献）。

Docker 的优势

Docker 一开始能吸引众多用户，其中一个很重要的因素就在于它上手使用很简单。在安装完 Docker 之后，一行命令语句 docker run hello-world 便已经运行了一个容器，一切看着都及其简单。

然而在实际的生产使用中，我们所运行的容器包含着业务代码，生产环境的网络、系统等因素也都很复杂。一旦出现问题，如何才能最高效地定位并解决问题，并且保证之后不再有类似的问题发生？

要做到这些，就需要对 Docker 技术有足够的了解。在大型公司中，都有专职负责基础支撑技术的开发或运维人员，而在中小型企业可能就很少有这类团队或者压根没有，需要开发人员自行解决。如果对 Docker 技术不甚了解，解决问题需要花费大量的时间，而且可能并不能彻底解决问题。

另一方面，不一定只是在容器运行过程时才会遇到问题，我们可能在构建镜像的时候就已经遇到了各种各样的问题：

- 构建的镜像体积为什么这么大
- Pull 镜像为什么这么慢
- 新构建的镜像为什么不生效
- 构建镜像为什么这么久

我们往往在遇到问题时才会想着怎么解决问题，但如果不成体系的学习，终究很难做到游刃有余。而且**成体系的学习，有助于巩固知识，将其吸收并转化为自己内在的能力。**

现在很多岗位的招聘需求上都有写需要了解或者掌握 Docker，或者将掌握 Docker 等容器技术作为加分项目，并且这个要求不局限于运维或者后端开发等岗位。

专栏大纲

我希望借由这个专栏，**将 Docker 容器技术的本质和思想与我在开发和运维 Docker 过程中对其原理和实践经验的总结讲清楚，并将结合着实践和核心特性的原理，加深对 Docker 容器技术的理解。**

因此，我把专栏划分成了三大模块：

（一）**Docker 入门**：这个模块分成了三篇内容，通过第一篇，带你了解 Docker 容器技术生态的发展脉络；第二篇，是为刚入门 Docker 的读者准备的，也是为后续章节进行铺垫；第三篇是很多读者或公司都常会困惑的问题，Docker 与 Linux 内核兼容性如何，要上生产环境该选择哪个版本？我会在这一篇中与你分享，让你不再困惑。

（二）**Docker 核心特性**：这个大模块围绕 Docker 的核心知识点，拆分成了 7 大部分，分别是容器、镜像、CI/CD、架构、存储、安全和网络。这些是该专栏的核心内容，在这部分内容中，我将基本按照从实践到原理的方式进行组织，让你从根儿上知道如何用，以及为什么这么用。

- 在容器篇，我会先给你介绍容器生命周期管理相关的内容，那你对容器的使用有个基本认识；之后会对容器资源进行管理；并对容器的核心进行深入剖析；最后动手来自己写容器。

- 在镜像篇，将介绍镜像完整的生命周期管理；镜像是如何构建与分发的；如何使用 Dockerfile 进行镜像构建；并介绍 Docker 的下一代构建系统是如何提速近 10 倍的；接下来结合我的实际镜像为你介绍 Dockerfile 的优化和最佳实践；最后分别深入源码为你介绍镜像构建、分发的原理，以及认证流程和原理。

- 在 CI/CD 篇，会介绍如何将 Docker 与 CI/CD 结合，同时介绍适用于生产环境使用的 CI/CD pipeline，希望能为读者建设 CI/CD 提供一些启发。

- 架构篇中我会结合源码介绍 Docker 的核心架构，以及其是如何协作的；Docker 提供了一种可扩展的 Plugin 机制，在特定场景下使用 Plugin 扩展 Docker 也是一种不错的选择；接下来会结合实际经验介绍容器监控和日志方面的具体实践方案，希望能为读者在实际使用中提供一个参考；最后会**与读者分享我所总结的 Docker 相关问题的定位及调试手段** 正确的方法能让你排查问题的效率翻倍。

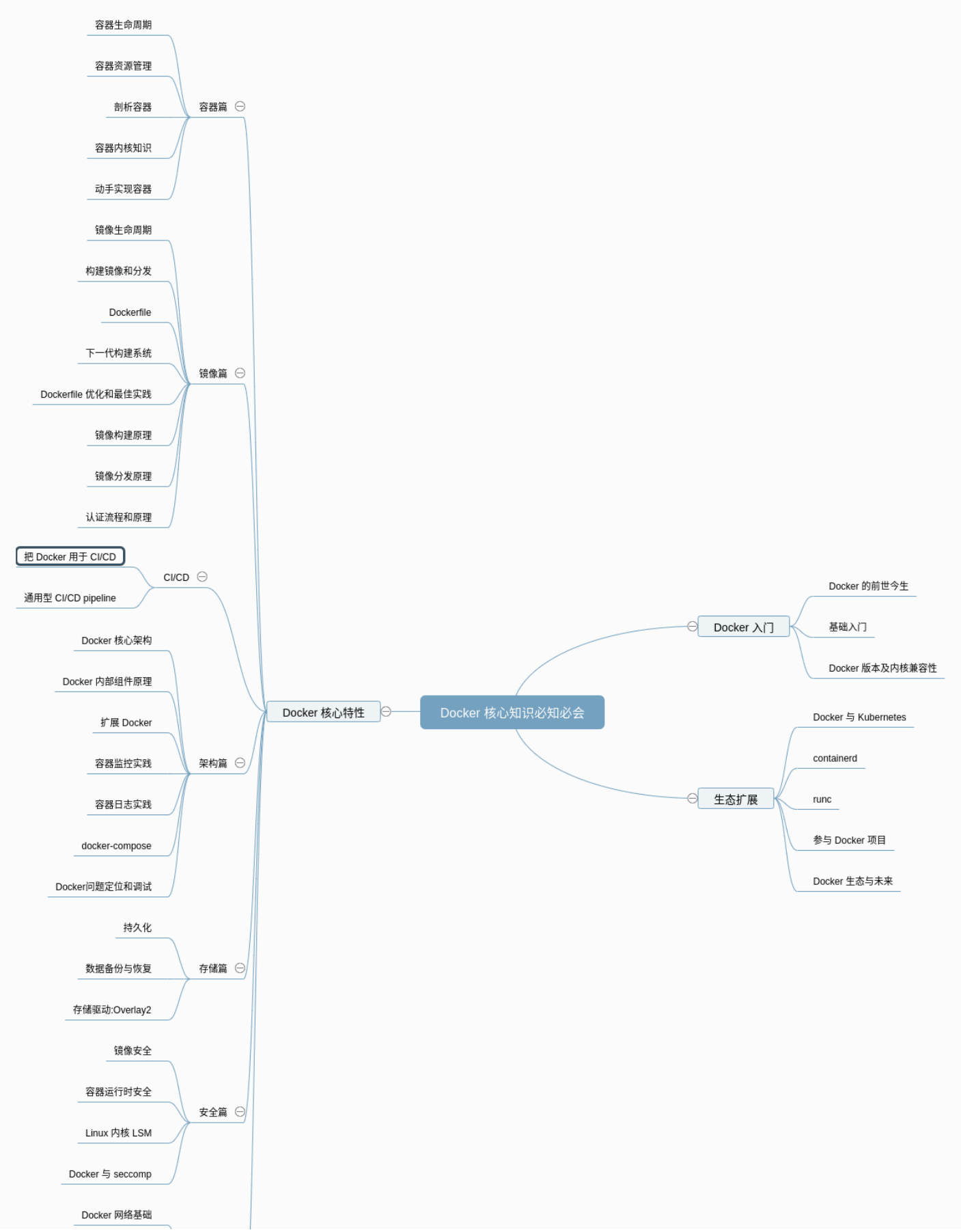
- 存储篇主要介绍 Docker 中 volume 的使用；以及如何进行数据备份和恢复；最后会深入内部介绍现在 Docker 最推荐的 Overlay2 存储驱动的工作原理。

- 安全篇会涉及镜像和容器运行时的安全；以及会涉及一些的 Linux 内核安全相关的知识，详细介绍如何利用 Linux 内核的安全模块为 Docker 保驾护航。

- 网络篇除了介绍基础网络知识外，还会介绍如何定制 bridge 网络；iptables 始终是一个很核心的知识点，我会为读者将 Docker 与 iptables 梳理清楚，以及如何自定义的进行网络管理；最后详细介绍了 docker-proxy，Docker 内部 DNS 以及 Docker 的核心网络知

识，让网络不再成为一个拦路虎。

（三）**生态扩展**：不得不说“开源”是 Docker 成长迅速的关键，在这个模块中，我将详细介绍 Docker 与 Kubernetes 间的联系，以及容器生态中的其他组件；与读者分享如何参与到 Docker 容器生态内，当然这里也会涉及到 Docker 现在的代码组织相关的内容；最后将与读者探讨 Docker 生态未来的走向。



专栏寄语

专栏会采用循序渐进的方式，从基础实践到原理分析，我希望能通过此专栏，让你对 Docker 容器技术做到“知其然，知其所以然”，将 Docker 容器技术作为自己的一技之长。

Docker 等容器技术正处于技术的热潮中，将其内化为自己的技术实力，不仅可以提升自己日常的开发部署效率，更可以作为面试的一个加分项。同时，深入原理后，相信你也会从中学习到很多不只局限于 Docker 自身的知识。