

# 程序员练级攻略（2018）：容器化和自动化运维

2018-07-24 陈皓



程序员练级攻略（2018）：容器化和自动化运维

朗读人：柴巍 06'42" | 3.07M

这篇文章重点学习 Docker 和 Kubernetes，它们已经是分布式架构和自动化运维的必需品了，也是你必需要学习的。对于这两个东西，你千万不要害怕，因为技术方面都不算复杂，只是它们的玩法和传统运维不一样，所以不用担心，只要你花上一点时间，一定会学好的。

## Docker

- 你可以先看一下 Docker 的官方介绍 [Docker Overview](#)。
- 然后再去一个 Web 在线的 Playground 上体验一下，[Katacoda Docker Playground](#) 或者是 [Play With Docker](#)。
- 接下来，跟着 [Learn Docker](#) 这个文档中的教程自己安装一个 Docker 的环境，实操一把。
- 然后跟着 [Docker Curriculum](#) 这个超详细的教程玩一下 Docker。

有了上述的一些感性体会之后，你就可以阅读 Docker 官方文档 [Docker Documentation](#) 了，这是学习 Docker 最好的方式。

如果你想了解一下 Docker 的底层技术细节，你可以参看我的文章。

- [Docker 基础技术：Linux Namespace \(上\)](#)
- [Docker 基础技术：Linux Namespace \(下\)](#)
- [Docker 基础技术：Cgroup](#)
- [Docker 基础技术：AUFS](#)
- [Docker 基础技术：DeviceMapper](#)

还有一些不错的与 Docker 网络有关的文章你需要阅读及实践一下。

- [A container networking overview](#)
- [Docker networking 101 - User defined networks](#)
- [Understanding CNI \(Container Networking Interface\)](#)
- [Using CNI with Docker](#)

Docker 有下面几种网络解决方案：[Calico](#)、[Flannel](#) 和 [Weave](#)，你需要学习一下。另外，还需要学习一下 [netshoot](#)。这是一个很不错的用来诊断 Docker 网络问题的工具集。

关于这几个容器网络解决方案的性能对比，你可以看一下下面这几篇文章或报告。

- [Battlefield: Calico, Flannel, Weave and Docker Overlay Network](#)
- [Comparison of Networking Solutions for Kubernetes](#)
- [Docker Overlay Networks: Performance analysis in high-latency environments](#)

如果你对 Docker 的性能有什么问题的话，你可以看一下下面这些文章。

- [IBM Research Report: An Updated Performance Comparison of Virtual Machines and Linux Containers](#)
- [An Introduction to Docker and Analysis of its Performance](#)

下面是一些和存储相关的文章。

- [Storage Concepts in Docker: Network and Cloud Storage](#)
- [Storage Concepts in Docker: Persistent Storage](#)
- [Storage Concepts in Docker: Shared Storage and the VOLUME directive](#)

然后是跟运维相关的文章。

- [Docker Monitoring with the ELK Stack: A Step-by-Step Guide](#)

最后，推荐看看 [Valuable Docker Links](#)，其中收集并罗列了一系列非常不错的 Docker 文章。

## 最佳实践

下面分享一些与 Docker 相关的最佳实践。

- [Best Practices for Dockerfile](#)，Docker 官方文档里的 Dockerfile 的最佳实践。
- [Docker Best Practices](#)，这里收集汇总了存在于各个地方的使用 Docker 的建议和实践。
- [Container Best Practices](#)，来自 Atomic 项目，是一个介绍容器化应用程序的架构、创建和管理的协作型文档项目。
- [Eight Docker Development Patterns](#)，八个 Docker 的开发模式：共享基础容器、共享同一个卷的多个开发容器、开发工具专用容器、测试环境容器、编译构建容器、防手误的安装容器、默认服务容器、胶黏容器。

## Kubernetes

Kubernetes 是 Google 开源的容器集群管理系统，是 Google 多年大规模容器管理技术 Borg 的开源版本，也是 CNCF 最重要的项目之一，主要功能包括：

- 基于容器的应用部署、维护和滚动升级；
- 负载均衡和服务发现；
- 跨机器和跨地区的集群调度；
- 自动伸缩；
- 无状态服务和有状态服务；
- 广泛的 Volume 支持；
- 插件机制保证扩展性。

Kubernetes 发展非常迅速，已经成为容器编排领域的领导者。

首先，我推荐你阅读 Kubernetes 前世今生的一篇论文。

- [Borg, Omega, and Kubernetes](#)，看看 Google 这十几年来从这三个容器管理系统中得到的经验教训。

学习 kubernetes，有两个免费的开源电子书。

- 《[Kubernetes Handbook](#)》，这本书记录了作者从零开始学习和使用 Kubernetes 的心路历程，着重于经验分享和总结，同时也会有相关的概念解析。希望能够帮助你少踩坑，少走

弯路，还会指引你关注 kubernetes 生态周边，如微服务构建、DevOps、大数据应用、Service Mesh、Cloud Native 等领域。

- 《[Kubernetes 指南](#)》，这本书旨在整理平时在开发和使用 Kubernetes 时的参考指南和实践总结，形成一个系统化的参考指南以方便查阅。

这两本电子书都不错，前者更像是一本学习教程，而且面明显广一些，还包括 Cloud Natvie、Service Mesh 以及微服务相关的东西。而后者聚焦于 Kubernetes 本身，更像一本参考书。

但是也别忘了 Kubernetes 的官方网站：[Kubernetes.io](#)，上面不但有[全面的文档](#)，也包括一个很不错的[官方教程](#)。

此外，还有一些交互式教程，帮助你理解掌握，以及一些很不错的文章推荐你阅读。

一些交互式教程

- [Katacoda](#)
- [Kubernetes Bootcamp](#)

一些文章

这里还有一些不错的文档，你应该去读一下。

- [Kubernetes tips & tricks](#)
- [Achieving CI/CD with Kubernetes](#)
- [How to Set Up Scalable Jenkins on Top of a Kubernetes Cluster](#)
- 10 Most Common Reasons Kubernetes Deployments Fail [Part I](#) 和 [Part II](#)
- [How to Monitor Kubernetes](#)，一共有 4 个篇章
- [Logging in Kubernetes with Fluentd and Elasticsearch](#)
- [Kubernetes Monitoring: Best Practices, Methods, and Existing Solutions](#)

网络相关的文章

要学习 Kubernetes，你只需要读一下，下面这个 Kubernetes 101 系列的文章。

- [Kubernetes 101 - Networking](#)
- [Kubernetes networking 101 - Pods](#)
- [Kubernetes networking 101 - Services](#)
- [Kubernetes networking 101 - \(Basic\) External access into the cluster](#)
- [Kubernetes Networking 101 - Ingress resources](#)
- [Getting started with Calico on Kubernetes](#)

## CI/CD 相关的文章

- [Automated Image Builds with Jenkins, Packer, and Kubernetes](#)
- [Jenkins setups for Kubernetes and Docker Workflow](#)
- [Lab: Build a Continuous Deployment Pipeline with Jenkins and Kubernetes](#)

## 最佳实践

- [Kubernetes Best Practices](#) by [Sachin Arote](#) , AWS 工程师总结的最佳实践。
- [Kubernetes Best Practices](#) by [Sandeep Dinesh](#) , Google 云平台工程师总结的最佳实践。

## Docker 和 Kubernetes 资源汇总

下面是 Github 上和 Docker & Kubernetes 相关的 Awesome 系列。

- [Awesome Docker](#)。
- [Awesome Kubernetes](#)。

虽然上面的这些系列非常全的罗列了很多资源，但是我觉得很不系统。对于系统的说明 Docker 和 Kubernetes 生态圈，我非常推荐大家看一下 The New Stack 为 Kubernetes 出的一系列的电子书或报告。

- [The New Stack eBook Series](#) , 非常完整和详实的 Docker 和 Kubernetes 生态圈的所有东西。
  - Book 01: [The Docker Container Ecosystem](#)
  - Book 02: [Applications & Microservices with Docker & Containers](#)
  - Book 03: [Automation & Orchestration with Docker & Containers](#)
  - Book 04: [Network, Security & Storage with Docker & Containers](#)
  - Book 05: [Monitoring & Management with Docker & Containers](#)
  - Book 06: [Use Cases for Kubernetes](#)
  - Book 07: [State of the Kubernetes Ecosystem](#)
  - Book 08: [Kubernetes Deployment & Security Patterns](#)
  - Book 09: [CI/CD with Kubernetes](#)
  - Book 10: [Kubernetes solutions Directory](#)
  - Book 11: [Guid to Cloud-Native Microservices](#)

## 小结

总结一下今天的内容。Docker 和 Kubernetes 已经成为分布式架构和自动化运维方面的不可或缺的两大基本构成，是你必需要学习的。虽然它们的玩法跟传统运维不一样，但技术方面并不算

复杂，只要你花上一点时间，一定会学好的。

在这篇文章中，我推荐了 Docker 和 Kubernetes 基础技术方面的学习资料，并给出了存储、运维、网络、CI/CD 等多方面的资料，同时列出了与之相关的最佳实践。相信认真学习和消化这些知识，你一定可以掌握 Docker 和 Kubernetes 两大利器。

下篇文章，我们将学习机器学习和人工智能方面的内容。敬请期待。

下面是《程序员练级攻略（2018）》系列文章的目录（持续更新中）。

- [开篇词](#)
- 入门篇
  - [零基础启蒙](#)
  - [正式入门](#)
- 修养篇
  - [程序员修养](#)
- 专业基础篇
  - [编程语言](#)
  - [理论学科](#)
  - [系统知识](#)
- 软件设计篇
  - [软件设计](#)
- 高手成长篇
  - [Linux 系统、内存和网络（系统底层知识）](#)
  - [异步 I/O 模型和 Lock-Free 编程（系统底层知识）](#)
  - [Java 底层知识](#)
  - [数据库](#)
  - [分布式架构入门（分布式架构）](#)
  - [分布式架构经典图书和论文（分布式架构）](#)
  - [分布式架构工程设计（分布式架构）](#)
  - [微服务](#)
  - [分布式架构工程设计](#)
  - [容器化和自动化运维](#)
  - .....





版权归极客邦科技所有，未经许可不得转载

#### 精选留言



dilei

👍 8

耗子叔 这几期感觉像是在贴链接，能不能找一两个重点细细解读下，还是想听听你的东西。

2018-07-24



刘强

👍 2

在二叉树视频里，看到耗子的一头白发，顿生感动和佩服。没有人能随随便便成功，耗子的几百行代码值20万，背后付出了多少。

2018-07-24



小薛薛

👍 1

做个标记，看几年后能不能走到这里。

2018-07-24



Joran

👍 1

耗子叔你的知识广度和深度，我十分敬佩，但是这么多我感觉够我学一辈子了

2018-07-24



机械师种太阳

👍 1

非常期待人下一期人工智能~~

2018-07-24



李志博

👍 0

非常喜欢，太感谢分享这些宝贵的资源

2018-07-24



Abyssal  
感谢皓叔！  
2018-07-24

👍 0



\_CountingStars  
kubernetes in action 不错  
2018-07-24

👍 0



manx00  
看到目录最后的三个点，感觉路还长...  
2018-07-24

👍 0