

Jeferson Fernando Noronha Vitalino
Marcus André Nunes Castro



Descomplicando o Docker

✓ 2ª EDIÇÃO
ATUALIZADA



Prefácio de
Jérôme Petazzoni



Descomplicando o

Docker

Jeferson Fernando Noronha Vitalino
Marcus André Nunes Castro

Descomplicando o
Docker

2ª edição



Copyright© 2018 por Brasport Livros e Multimídia Ltda.

1ª edição: 2016

2ª edição: 2018

Todos os direitos reservados. Nenhuma parte deste livro poderá ser reproduzida, sob qualquer meio, especialmente em fotocópia (xerox), sem a permissão, por escrito, da Editora.

Para uma melhor visualização deste e-book sugerimos que mantenha seu software constantemente atualizado.

Editor: Sergio Martins de Oliveira

Diretora Editorial: Rosa Maria Oliveira de Queiroz

Gerente de Produção Editorial: Marina dos Anjos Martins de Oliveira

Editoração Eletrônica: SBNigri Artes e Textos Ltda.

Capa: Use Design

Produção de e-pub: SBNigri Artes e Textos Ltda.

Técnica e muita atenção foram empregadas na produção deste livro. Porém, erros de digitação e/ou impressão podem ocorrer. Qualquer dúvida, inclusive de conceito, solicitamos enviar mensagem para brasport@brasport.com.br, para que nossa equipe, juntamente com o autor, possa esclarecer. A Brasport e o(s) autor(es) não assumem qualquer responsabilidade por eventuais danos ou perdas a pessoas ou bens, originados do uso deste livro.

ISBN Digital: 978-85-7452-902-8

BRASPORT Livros e Multimídia Ltda.

Rua Teodoro da Silva, 536 A – Vila Isabel

20560-001 Rio de Janeiro-RJ

Tels. Fax: (21) 2568.1415/2568.1507

e-mails:

marketing@brasport.com.br

vendas@brasport.com.br

editorial@brasport.com.br

site: www.brasport.com.br

Filial

Av. Paulista, 807 – conj. 915
01311-100 – São Paulo-SP

Agradecimentos

Jeferson Fernando

Ao meu companheiro de longas jornadas de trampo, Marcus André, pessoa fundamental para o desenvolvimento deste livro. Além de ser um superamigo, orgulha-me muito acompanhar seu crescimento profissional e pessoal! *Ieiinii!*

Aos alunos que acumulei e com quem fiz amizades ao longo desses 15 anos ministrando treinamentos pelo Brasil, especialmente aos alunos do treinamento *Descomplicando o Docker* da LINUXtips! :D

Ao Carlos “do CNI” de’Villa, lá de Itanhaém, que foi o primeiro a acreditar no meu potencial em repassar o conhecimento, isso quando eu tinha apenas 16 anos.

A todos os amigos que fiz por onde passei, especialmente o Juliano “ncode” Martinez, João “orelhinhas” Gabriel, Jean Feltrin, Guilherme “Lero” Schroeder, Rodolfo “enetepe” Ponteado, Guilherme Almeida, Luiz Salvador Jr, Jhonatan Araujo, Francisco “chico” Freire, Adinan Paiva, Adilson Nunes, Pedro Vara, Eduardo Lipolis, Fábio “santa” Santiago, Rodrigo “birgui” Inacio, Giordano Fecho, Rafael Benvenuti, João Borges, Mateus Prado, Leo Martins, Guilherme Calcette e o Ricardo Iorio.

Com muito carinho, gostaria de agradecer a duas pessoas que foram muito importantes para mim nesses últimos anos, principalmente em relação a minha carreira: Eduardo Scarpellini e Leonardo Lorieri, muito obrigado! Como sempre digo, vocês mudaram a minha vida! <3

Meus chefes:

Rogério Lelis, que sempre me ajudou e me apoiou em tudo que precisei! André Brandão, que, além de ser o melhor gerente que já tive, é o meu *coach* e ajudou-me a quebrar algumas barreiras bem importantes em minha vida. Rodrigo Campos, que tenho como espelho pela sensacional habilidade de palestrar com maestria.

Quero agradecer a minha sogra, Suely, meu sogro, Mario Sergio, e minha cunhada, Camila Silvano, que sempre me apoiam e torcem por mim. Amo vocês!

Quero agradecer muito aos meus irmãos Magno Junior e Camila Vitalino, sempre ao meu lado e me apoiando em tudo. Amo vocês!

Meu filhote Eduardo, que hoje já está um homem! Como o tempo passa! Eu te amo, Brooo!

O agradecimento mais especial é para as mulheres da minha vida:

- Minha esposa e fiel companheira Aletheia, que sempre está ao meu lado em todos os momentos, tristes ou felizes, fáceis ou difíceis, e que sempre acreditou em mim. Eu te amo!
- Minhas princesas e as coisas mais lindas, Maria Fernanda e Maria Eduarda, que são meu tudo, meu ar, minha vida! Eu amo vocês, princesas!
- E finalmente minha rainha, minha mãe, Cidinha Tavares, que passou por diversas dificuldades para conseguir educar o homem que hoje sou! Eu te amo!

Marcus André

Eu agradeço, primeiramente, ao Jeferson Vitalino (quem?) pela ideia, oportunidade e parceria neste livro. Gostaria, também, de continuar agradecendo a ele pelo tempo, pela dedicação e, acima de tudo, por acreditar em mim. Esses últimos anos têm sido, sem dúvida, os melhores da minha carreira profissional e você tem uma bela parcela de culpa nisso.

Queria também agradecer a todas as pessoas que contribuíram direta ou indiretamente para a minha formação como profissional de TI: colegas de trabalho, professores da faculdade, chefes, ex-clientes, pessoas que me agregaram muito valor e às quais eu serei eternamente grato.

Um agradecimento especial aos meus amigos/irmãos: Mário-Anderson, Márcio Ângelo, Bruno Henrique e Manuela Guedes. Meus amigos Juciano Werllen, Rondineli Gomes, Renan Vicente e amigos de outras áreas: Jémina Diógenes, Juliana Mesquita, Ana Clara Rezende e Rodrigo Peixe.

À minha família: ao meu pai, José Monte, e à minha mãe, dona-Evani, muito obrigado por tudo e parabéns, seu projeto de homem feliz deu certo. :)

À minha mais nova família, Elisângela Oliveira. Agradeço imensamente pela compreensão nesses últimos tempos, pelo carinho, pela emoção, pelo cuidado. Muito obrigado por me mostrar a complexidade nas coisas simples e o universo que existe entre o 0 e o 1. Eu te amo.

Agradecimento especial:

Gostaríamos ainda de agradecer ao Vicente Marçal, ao Carlos Augusto Malucelli, ao Eduardo Fonseca e ao Henrique Serrat Guimarães por dar aquela conferida final no livro, para que

podéssemos ter a certeza de que você, leitor, teria uma experiência sensacional durante a leitura e execução dos exemplos. Bom aprendizado, divirtam-se!

Prefácio da primeira edição

If you are a developer, you probably have heard about Docker by now: Docker is the ideal platform to run applications in containers. Fine, but what are those containers? Containers are a lightweight virtualization technique providing us with lots of possibilities: thanks to them, we can ship applications faster; we can easily implement CI/CD (continuous integration and continuous delivery); we can set up development environments faster than ever; we can ensure parity between those development environments and our production servers; and much more. If you write code, if you deploy code, if you operate code, then I promise that containers are going to make your life easier.

But containers have been around for *decades*, and Docker was only released in 2013. So what's the big deal? Why is everybody so excited about Docker, if the technology behind it is more than ten years old?

Because the main innovation of Docker is not the technology. The main innovation of Docker is to make this technology available to every developer and sysadmin, very easily, without having to spend years of practice to become an expert, and without requiring to develop tons of custom tools.

I will share a secret with you: the really important thing is not Docker and containers. What is really important is to improve the quality of our software, reduce our development and deployment costs, release better, more reliable code, faster. It turns out that Docker is an insanely efficient way to achieve those goals. That's why it is so popular.

This book will teach you all you need to know to get started with Docker, and use it to build, ship, and run your applications. It will be your guide in the world of containers, and on the path to ship code better than you ever did before.

Jérôme Petazzoni

Docker Tinkerer Extraordinaire

Sobre o Livro

A proposta deste livro é ajudá-lo a construir ambientes utilizando *containers* com a ferramenta que está revolucionando as empresas de tecnologia ao redor do mundo: vamos aprender e brincar bastante com o sensacional Docker!

De maneira leve e totalmente prática, vamos aprender desde o que é o Docker até a criação de um *cluster* Docker com diversos *containers* em execução! Sempre de forma prática, vamos abordar temas importantes para que consiga administrar ambientes que utilizam ou pretendam utilizar o Docker.

Inclusive, vamos aprender a montar *dockerfiles* personalizados para construção de imagens de *containers*, além de conhecer como melhor administrá-las.

Também veremos na prática como utilizar o Docker Machine para criação de *hosts* Docker, seja local ou na nuvem. Vamos criar um *cluster* utilizando o Swarm e aprender como escalar o nosso ambiente através do Compose.

Nesta segunda edição adicionamos muitas novidades, como o Docker Secret e o Docker Stack, e atualizamos praticamente todo o restante do livro, visando torná-lo super atual e didático. Adicionamos mais exemplos práticos, para que sua experiência possa ser ainda mais agradável e proveitosa.

Sobre os Autores

Jeferson Fernando

Profissional com mais de 15 anos de experiência em administração de servidores Unix/Linux em ambientes críticos de grandes empresas como Motorola, TIVIT, Votorantim, Alog, Locaweb, Walmart.com e Santander. Possuindo mais de 12 anos de experiência como instrutor em grandes centros de treinamento, passou por Impacta Tecnologia, Utah Linux Center, Green, 4Linux e foi instrutor oficial da Red Hat. Já ministrou treinamentos como consultor em empresas como Petrobras, Vale, Ministério do Exército, Polícia Militar do Estado de SP e HP. Possui diversas certificações, como *Docker Certified Associate*, *Certified Kubernetes Administrator*, LPI, RHCE, RHCI, Solaris, entre outras. É um grande entusiasta do software livre e especialista na administração de ambientes críticos com cultura DevOps. Possui ainda um canal no YouTube chamado LinuxTips, onde aborda temas interessantes para todos os níveis de administradores de sistemas Linux.

Marcus André

Profissional com mais de 10 anos de experiência em administração de redes e sistemas, tendo passado por grandes empresas nos últimos anos, como Walmart.com, IBM, Santander,

atuando em diversas plataformas e ambiente homogêneo (*nix, Windows).

Focado em FOSS (*Free and Open Source Software*), é apaixonado e evangelista da linguagem Python e de toda a cultura DevOps.

Sumário

Capa

Copyright

Agradecimentos

Prefácio da primeira edição

Sobre o Livro

Sobre os Autores

Introdução

1. O que é container?

1.1. Então vamos lá, o que é um container?

1.2. E quando começou que eu não vi?

2. O que é o Docker?

2.1. Onde entra o Docker nessa história?

2.2. E esse negócio de camadas?

2.2.1. Copy-On-Write (COW) e Docker

2.3. Storage drivers

2.3.1. AUFS (Another Union File System)

2.3.2. Device Mapper

2.3.3. OverlayFS e OverlayFS2

2.3.4. BTRFS

2.4. Docker Internals

2.5. Namespaces

2.5.1. PID namespace

2.5.2. Net Namespace

2.5.3. Mnt Namespace

2.5.4. IPC namespace

2.5.5. UTS namespace

2.5.6. User namespace

2.6. Cgroups

2.7. Netfilter

2.8. Para quem ele é bom?

3. Instalando o Docker

3.1. Quero instalar, vamos lá?

3.2. Instalando no Debian/Centos/Ubuntu/Suse/Fedora

3.3. Instalando 'manualmente' no Debian

3.3.1. Dica importante

4. Executando e administrando containers Docker

4.1. Então vamos brincar com esse tal de container!

4.2. Legal, quero mais!

4.2.1. Modo interativo

4.2.2. Daemonizando o container

4.3. Entendi, agora vamos praticar um pouco?

4.4. Tá, agora quero sair...

4.5. Posso voltar ao container?

4.6. Continuando com a brincadeira...

4.7. Subindo e matando containers...

4.8. Visualizando o consumo de recursos pelo container...

4.9. Cansei de brincar de container, quero removê-lo!

5. Configurando CPU e memória para os meus containers

5.1. Especificando a quantidade de memória

5.2. Especificando a quantidade de CPU

5.3. Eu consigo alterar CPU e memória dos meus containers em execução?

6. Meu primeiro e tosco dockerfile...

7. Entendendo volumes

7.1. Introdução a volumes no Docker

7.2. Criando volumes

7.3. Localizando volumes

7.4. Criando e montando um data-only container

7.5. Sempre é bom um backup...

8. Criando e gerenciando imagens

8.1. Agora eu quero criar minha imagem, posso?

8.2. Vamos começar do começo então, dockerfile!

8.3. Vamos aprender um pouco mais sobre dockerfile?

8.4. Multi-stage

8.5. Vamos customizar uma imagem base agora?

9. Compartilhando as imagens

9.1. O que é o Docker Hub?

9.2. Vamos criar uma conta?

9.3. Agora vamos compartilhar essas imagens na interwebs!

9.4. Não confio na internet; posso criar o meu registry local?

10. Gerenciando a rede dos containers

10.1. Consigo fazer com que a porta do container responda na porta do host?

10.2. E como ele faz isso? Mágica?

11. Controlando o daemon do Docker

11.1. O Docker sempre utiliza 172.16.X.X ou posso configurar outro intervalo de IP?

11.2. Opções de sockets

11.2.1. Unix Domain Socket

11.2.2. TCP

11.3. Opções de storage

11.4. Opções de rede

11.5. Opções diversas

12. Docker Machine

12.1. Ouvi dizer que minha vida ficaria melhor com o Docker Machine!

12.1.1. Vamos instalar?

12.1.2. Vamos iniciar nosso primeiro projeto?

13. Docker Swarm

13.1. Criando o nosso cluster!

13.2. O sensacional services!

14. Docker Secrets

14.1. O comando docker secret

14.2. Tudo bem, mas como uso isso?

14.3. Acessando a secret

14.4. Atualizando a secret de um serviço

15. Docker Compose

15.1. O comando docker stack

15.2. E já acabou?

Introdução

Se você é um desenvolvedor que sempre sonhou em poder reproduzir o ambiente de produção em sua máquina ou a aplicação que você desenhou e desenvolveu poder rodar em produção facilmente, com o mesmo “binário” que foi utilizado nos momentos de testes...

...Imagine conseguir instalar e testar diversas aplicações rapidamente e ainda simular um ambiente completo, bem mais leve e inteligente do que as máquinas virtuais em questão de minutos.

Se você é um administrador de sistemas que sempre sonhou em não precisar mais preparar os servidores para suportar milhares de aplicações diferentes, feitas em linguagens diferentes, com bibliotecas diferentes, com versões diferentes (ufa!)

...Imagine economizar milhões de horas em instalações de novos servidores, atualizações de aplicações, *deploys*, acionamentos na madrugada por conta de serviços que “falharam” e não poderiam ser executados em outros servidores por incompatibilidades diversas.

Se você precisa escalar seu ambiente em questão de minutos para suportar uma demanda que o pegou de surpresa após uma campanha do departamento de marketing de sua empresa...

Se você é um gestor e está preocupado com a fatura do *datacenter* no final do mês por conta da quantidade de recursos necessários para seu ambiente funcionar....

E mesmo quando você possui todos os recursos computacionais para suportar a demanda, se você percebe que na maioria das vezes os seus recursos estão ociosos e sua equipe está com dificuldades para equalizá-los...

...Então este livro e o Docker foram feitos para você!

Boa leitura e divirta-se!

1. O que é *container*?

1.1. Então vamos lá, o que é um *container*?

Container é, em português claro, o agrupamento de uma aplicação junto com suas dependências, que compartilham o *kernel* do sistema operacional do *host*, ou seja, da máquina (virtual ou física) onde está rodando. Deu para entender?

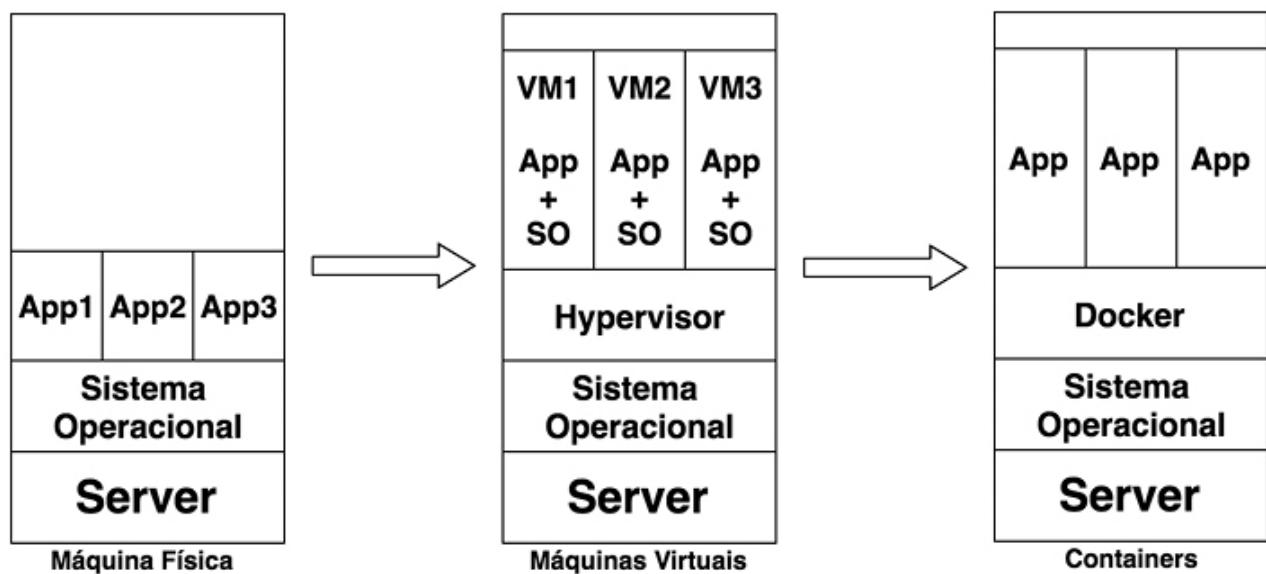
Containers são bem similares às máquinas virtuais, porém mais leves e mais integrados ao sistema operacional da máquina *host*, uma vez que, como já dissemos, compartilha o seu *kernel*, o que proporciona melhor desempenho por conta do gerenciamento único dos recursos.

Na maioria dos casos, a imagem de um *container* é bastante enxuta, havendo somente o necessário para o funcionamento da aplicação, que, quando em execução, possui um pequeno *overhead* se comparada à mesma aplicação rodando nativamente no sistema operacional, grande parte disso por conta do compartilhamento dos recursos com a máquina *host*.

Quando estamos utilizando máquinas virtuais, nós emulamos um novo sistema operacional e virtualizamos todo o seu hardware utilizando mais recursos da máquina *host*, o que não ocorre quando utilizamos *containers*, pois os recursos são compartilhados. O ganho

óbvio disso é a capacidade de rodar mais *containers* em um único *host*, se comparado com a quantidade que se conseguiria com máquinas virtuais.

A seguir, na figura, podemos notar as diferenças de quando temos aplicações sendo executadas nativamente, máquinas virtuais e por fim em *containers*. Repare que não é necessário emular um novo sistema operacional quando estamos utilizando *containers*, diferentemente das máquinas virtuais.



Outro ponto interessante na utilização de *containers* é a portabilidade. Não importa em qual ambiente você criou o seu *container*, ele irá rodar em qualquer outro que possua, no nosso caso, o Docker instalado, seja ele no Linux, MacOS ou Windows. Você não precisa se preocupar com suas dependências, está tudo dentro do *container*. :D

O desenvolvedor consegue, na sua própria máquina, criar uma aplicação em *container* e depois executá-la em um servidor de produção sem nenhum problema de dependência ou algo do tipo –

nem mesmo o bom e velho “engraçado, na minha máquina funciona” escapa, hein?

Lembre-se: na máquina virtual você emula um novo sistema operacional dentro do sistema operacional do *host*. Já no *container* você emula somente as aplicações e suas dependências e as torna portáteis.

1.2. E quando começou que eu não vi?

Apesar de o termo ter se tornado *hype* nos últimos anos, durante décadas já utilizávamos *containers* em sistemas Unix através do comando *chroot*. Sim, bem mais simplório, é verdade, pois era apenas uma forma de isolar o *filesystem*, mas já era o começo!

Em seguida vieram os *jails* do FreeBSD, que, além do isolamento do *filesystem*, permitiam também o isolamento de processos, seguidos de perto pela Sun, que desenvolveu o *Solaris Zones*, mais uma solução baseada em *containers*, porém somente para sistemas Solaris.

O grande passo rumo ao cenário que temos hoje foi a criação, pela Parallels do Virtuozzo, de um painel que permitia o fácil gerenciamento de *containers* e a disponibilização do *core* do Virtuozzo como *open source* com o nome de OpenVZ.

O OpenVZ foi uma ferramenta que ganhou bastante destaque no gerenciamento de *containers* e ajudou e muito na popularização do VPS (*Virtual Private Server*) e, conseqüentemente, na criação de centenas de empresas de *hosting* espalhadas pelo mundo. O principal ponto negativo do OpenVZ era a necessidade de aplicar um *patch* no *kernel* Linux.

Logo após surgir o OpenVZ, o Google iniciou o desenvolvimento do CGroups para o *kernel* do Linux e iniciou a utilização de *containers* em seus *datacenters*.

Em 2008, desenvolvedores de empresas como Virtuozzo, IBM e o próprio Google iniciaram o projeto LXC, que trazia consigo o CGroups, *namespaces* e *chroot* para prover uma completa e estável solução para a criação e o gerenciamento de *containers*.

Porém, foi no ano de 2013 que os *containers* conquistaram o *mainstream*, saíram do *underground* através da utilização massiva pelas empresas de internet e gigantes de tecnologia e invadiram os principais eventos de tecnologia ao redor do mundo, com palestras sobre o sucesso na utilização de *containers* e com o melhor aproveitamento dos recursos físicos como CPU e memória, maior agilidade no *deployment* de novas aplicações em fração de segundos e tudo isso com uma facilidade que impressiona. Amigo, estamos falando do simplesmente sensacional **Docker**.

2. O que é o Docker?

2.1. Onde entra o Docker nessa história?

Tudo começou em 2008, quando Solomon Hykes fundou a dotCloud, empresa especializada em PaaS com um grande diferencial: o seu *Platform-as-a-Service* não era atrelado a nenhuma linguagem de programação específica, como era o caso, por exemplo, da Heroku, que suportava somente aplicações desenvolvidas em Ruby.

A grande virada na história da dotCloud ocorreu em março de 2013, quando decidiram tornar *open source* o *core* de sua plataforma – assim nascia o Docker!

As primeiras versões do Docker nada mais eram do que um *wrapper* do LXC integrado ao *Union Filesystem*, mas o seu crescimento foi fantástico e muito rápido, tanto que em seis meses seu GitHub já possuía mais de seis mil *stars* e mais de 170 pessoas contribuindo para o projeto ao redor do mundo.

Com isso, a dotCloud passou a se chamar Docker e a versão 1.0 foi lançada apenas 15 meses após sua versão 0.1. A versão 1.0 do Docker trouxe muito mais estabilidade e foi considerada “production ready”, além de trazer o Docker Hub, um repositório público para *containers*.

PRÉVIA GRATUITA · OFERTA POR TEMPO LIMITADO

Você chegou ao fim desta prévia.

Continue lendo "Descomplicando o Docker 2a Edição" e mais de 1 milhão de livros — de graça por 30 dias.

★★★★★ Mais de 1 milhão de leitores já aproveitam

Com o Kindle Unlimited, sua leitura não tem fim:

- ✓ Leia à vontade — explore mais de 1 milhão de títulos sem pagar por livro.
- ✓ Leve para qualquer lugar — baixe o app gratuito e leia onde e quando quiser.
- ✓ Em qualquer tela — celular, tablet, computador ou Kindle — você escolhe.
- ✓ Grandes autores — best-sellers e novos talentos, inclusive títulos em inglês.

COMEÇAR MEUS 30 DIAS GRÁTIS

Cobrança só após o período grátis.

- ✓ Pagamento seguro
- ✓ Acesso imediato
- ✓ Cancele quando quiser

Não precisa ter um Kindle: baixe o app gratuito e comece a ler agora.

Se não quiser ler no aplicativo Kindle, compre o livro [clikando aqui](#).