



Container, o novo passo para a virtualização.

João Marcelo Barros de Almeida

Linux Container é uma tecnologia que permite a criação de múltiplas instâncias isoladas de um determinado sistema operacional dentro de um único hospedeiro ou, em outras palavras, é uma maneira de virtualizar aplicações dentro de um servidor Linux.

O conceito é simples e antigo, sendo o comando *chroot* seu precursor mais famoso que foi lançado em 1979 pelo Unix V7 com o intuito de segregar acessos de diretórios e evitar que o usuário possa ter acesso à estrutura raiz (“/” ou *root*). Esse conceito evoluiu alguns anos mais tarde com o lançamento do comando *jail*, no sistema operacional FreeBSD 4. Essa implementação já introduzia a ideia de segregação de rede e limitação dos acessos de superusuários aos processos, que passou a ser adotada, com maiores funcionalidades, pelas distribuições Linux. Posteriormente foi melhor definido em alguns sistemas como o AIX WPAR e o Solaris Containers. Nesses dois sistemas já havia o conceito de virtualização de sistema operacional, mas não o conceito de *container*.

Nas distribuições Linux o *chroot* era uma maneira fácil de criar uma *jail* para as conexões dos servidores FTP, mas acabou ficando mais conhecido pela sua vulnerabilidade do que propriamente pela segurança que seu nome indicava implicitamente, mas acabou ajudando a cunhar um termo *jailbreak* (procedimento que permite a execução de aplicativos não-autorizados pelas regras de segurança), que é utilizado inclusive em sistemas operacionais de celulares, por exemplo.

A grande diferença entre o *chroot* e o Linux Container é o nível de segurança que se pode alcançar. Com relação à virtualização, a diferença está no fato do Linux Container não necessitar de uma camada de sistema operacional para cada aplicação, como pode ser visto na figura. Ao comparar com a virtualização tradicional, fica mais claro que uma aplicação sendo executada em um Linux Container demanda muito menos recursos, consumindo menos espaço em disco e com um nível de portabilidade difícil de ser alcançado por outras plataformas.

Tais características tornaram o Linux Container uma ferramenta poderosa para criação e teste de aplicações por desenvolvedores, conquistando também os administradores Linux pela facilidade de implementar as aplicações “empacotadas” nesse modelo de virtualização.

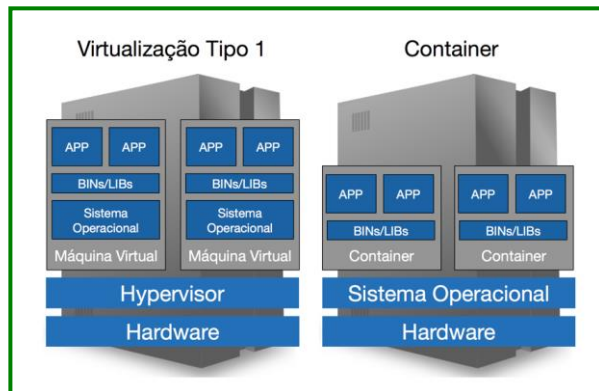
Mas não foi só a adoção de desenvolvedores e administradores que tornou essa tecnologia recentemente tão popular. A consolidação da virtualização no mercado e a crescente demanda por computação em nuvem criaram o ambiente perfeito para o Linux Container se espalhar rapidamente. Aplicações podem ser portadas direto do notebook do desenvolvedor para o servidor de produção, ou ainda para uma instância virtual em uma nuvem pública. Suas facilidades de uso ficaram evidentes ao mercado com a rápida adoção dos principais competidores de computação em nuvem, que perceberam o valor do Linux Container e adequaram seu catálogo para suportar essa implementação.

Hoje um dos mais conhecidos Linux Containers do mercado é o Docker, que nasceu como um projeto *open source* da DotCloud, uma empresa de PaaS (*Platform as a Service*) que apesar de estar mais interessada em utilizar Linux Containers apenas em suas aplicações, acabou desenvolvendo um produto que foi muito bem aceito pelo mercado. O Docker foi ganhando espaço como uma maneira de gerenciar *containers* no ambiente Linux e, depois de um ano de sua primeira versão, surpreendeu novamente lançando sua própria

biblioteca deixando de lado as bibliotecas nativas que eram utilizadas para gerenciar o Linux Container e assumindo o controle dos *drivers* diretamente com o *kernel*.

O que para muitos especialistas seria um erro, curiosamente para o mercado foi a criação de um padrão. Essa opção agradou aos principais mantenedores de Linux e Containers, como por exemplo a Canonical (Ubuntu), a Parallels e a RedHat, que passaram inclusive a contribuir com o código do Docker.

Ainda é cedo para se dizer que é o fim da virtualização de servidores como a conhecemos hoje. Muitas aplicações serão reescritas para suportar o uso de *containers* de forma adequada. Alguns sistemas operacionais ainda serão repensados para suportar o Linux Container nativamente. E também alguns paradigmas deverão ser quebrados para que essa tecnologia seja amplamente utilizada pelo mercado, exatamente como aconteceu há alguns anos com a virtualização de servidores.



Para saber mais

[http://domino.research.ibm.com/library/cyberdig.nsf/papers/0929052195DD819C85257D2300681E7B/\\$File/rc25482.pdf](http://domino.research.ibm.com/library/cyberdig.nsf/papers/0929052195DD819C85257D2300681E7B/$File/rc25482.pdf)

<https://blog.docker.com/>