ANÁLISE DE SEGURANÇA DE RECURSOS VIRTUALIZADOS EM NUVENS COMPUTACIONAIS IAAS BASEADAS EM OPENSTACK[1]

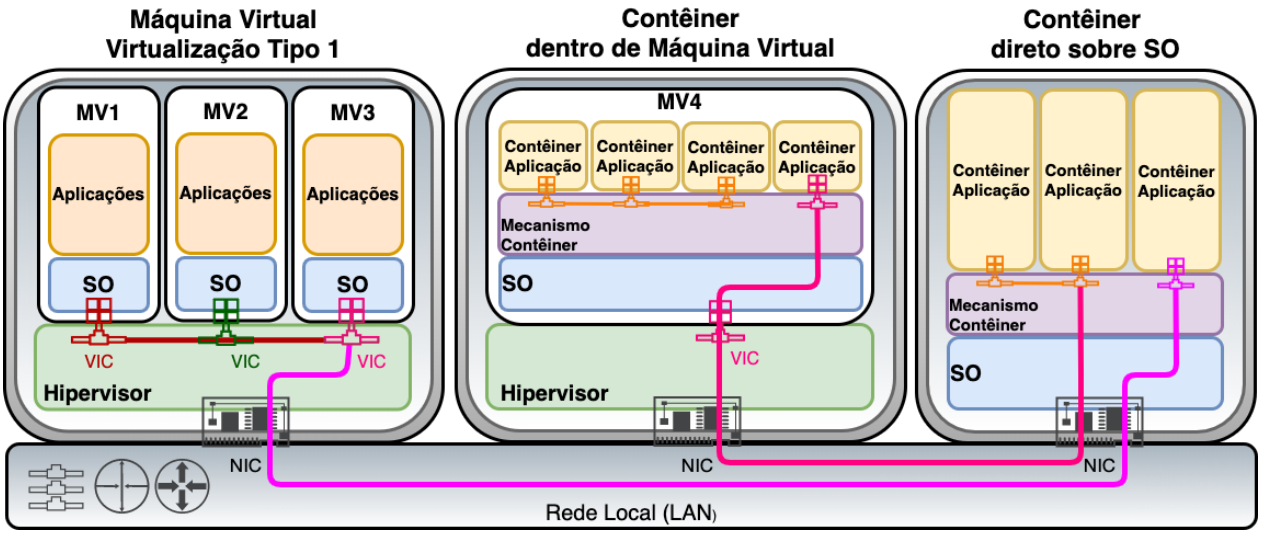
Pedro Kenzo Kawasaki Alves[2], Charles Christian Miers[3]

[1] Vinculado ao projeto "Análise de segurança de recursos virtualizados em nuvens computacionais IaaS baseadas em OpenStack usando honeypots de baixa interatividade"
[2]Acadêmica do Curso de Bacharelado em Ciência da Computação – CCT– Bolsista PROBITI/UDESC
[3] Orientador, Departamento de Computação – CCT – charles.miers@udesc.br

É fato afirmar que a virtualização colaborou significativamente na área da computação, mais especificamente na computação em nuvem. Com a evolução da tecnologia, e a quantidade de informações que transitam na rede a todo momento, faz se necessário o uso da nuvem, quer seja para processamento e/ou armazenamento. Atualmente, diversos serviços dependem de recursos computacionais disponibilizados por nuvens.

Existem diversas formas de virtualização, sendo que a máquina virtual (MV) tornou-se uma das abordagens mais populares para fornecer recursos computacionais devidos aos seus aspectos de compatibilidade, isolamento e desempenho. Contudo, no decorrer do tempo, a necessidade de otimizar recursos para obter maior eficiência fez com que fossem desenvolvidos novos meios de virtualização, sendo a conteinerização um destes meios. A Figura 1 ilustra, em uma visão geral, os componentes de máquina virtual (MV), contêiner baseado em MV e contêiner nativo.

Figura 1: Visão geral de máquina virtual, contêiner em máquina virtual e contêiner nativo.

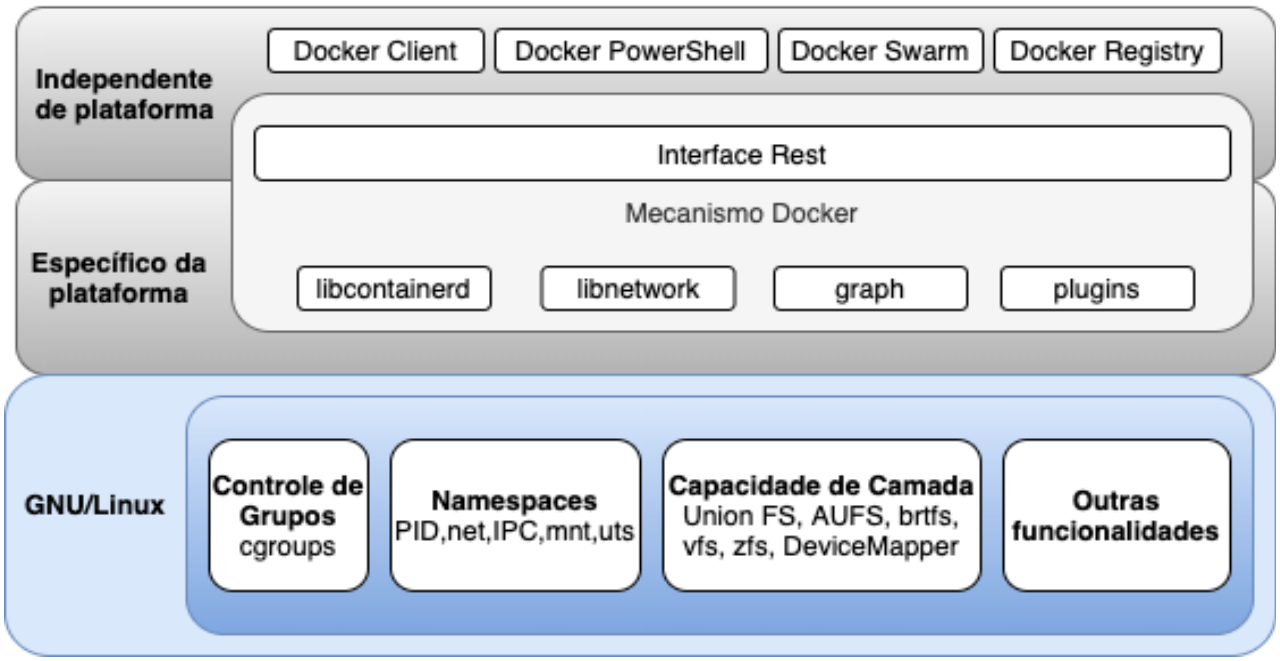


Visando uma redução no consumo de recursos surgiram os contêineres. Os contêineres (diferentemente da MVs) não necessitam de apenas um sistema operacional (SO), propiciando economia de recursos no computador hospedeiro. Ao contrário da MV, que requer um hipervisor, a conteinerização é realizada diretamente com o auxílio de softwares e bibliotecas que permitem controlar o acesso ao núcleo hospedeiro. Cada contêiner possui seu próprio *chroot*, *cgroups*, *namespace* e outras funcionalidades fornecidas pelo núcleo do SO hospedeiro.

Atualmente há diversas aplicações que utilizam a metodologia de conteinerização, dentre elas a Docker, sendo uma das mais populares. Há versões do Docker para GNU/Linux e MS-Windows, sendo o foco da pesquisa a GNU/Linux. O Docker reúne as principais partes de um software completo e as separa (encapsulamento). Sendo assim, uma aplicação conteinerizada utiliza as bibliotecas do núcleo e não necessitam de um SO por contêiner.

A Figura 2 indica que o Docker possui um conjunto de elementos que são independentes de plataforma (MS-Windows e GNU/Linux são suportados atualmente de modo nativo, identificado na cor cinza na Figura 2) e outros que são específico da plataforma. Na cor azul estão listados os principais elementos que o Docker emprega na sua versão para GNU/Linux, foco desta pesquisa.

Figura 2: Arquitetura geral do contêiner Docker.



Uma breve explicação dos principais elementos do Docker na plataforma GNU/Linux (Figura 2):

- *cgroups*: limita e restringe a quantidade de recursos de cada contêiner;
- *namespaces*: fornece o isolamento, de forma que nenhum outro contêiner interfira no outro;
- *chroot*: altera o diretório aparente do *root* para criar um novo ambiente do diretório *root* do sistema principal, o usuário que está operando nesse novo diretório não pode ver ou acessar arquivos fora dele;
- Capacidade de camada: permite que arquivos e diretórios de sistemas diferentes (ramificações) sejam sobrepostos, formando um único sistema de arquivos; e
- Outras funcionalidades: versionamento, compartilhamento, reutilização de componentes, entre outros.

Palavras-chave: Virtualização e contêineres, Segurança.