ANÁLISE DE DESEMPENHO E SEGURANÇA DOS ORQUESTRADORES APACHE MESOS, DOCKER SWARM E KUBERNETES EM NUVENS BASEADAS EM OPENSTACK [1]

Kerolayne de Souza Vieira de Oliveira[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 Ciência da Computação – CCT – charles.miers@udesc.br

As nuvens computacionais baseadas em OpenStack são a opção preferencial para implantar nuvens *Infrastructure-as-a-Service* (IaaS) privadas baseadas em software livre. Na primeira década de sua existência, o OpenStack consolidou-se como solução IaaS oferecendo virtualização através de máquinas virtuais (MVs) em uma quantidade considerável de plataformas de hipervisores. Os contêineres podem ser disponibilizados na plataforma OpenStack de diversas formas, assim como maneiras distintas de gerenciamento. Neste contexto, os contêineres são considerados uma solução atraente por possuírem implantação com poucos recursos computacionais e proporcionarem um ambiente básico. O Magnum é um serviço do OpenStack que possui compatibilidade nativa com as soluções de orquestração de contêineres Apache Mesos, Docker Swarm e Kubernetes (Tabela 1).

Os orquestradores atuam como facilitadores na gestão de *clusters* de contêineres e são compostos por três camadas básicas, entre elas o gerenciamento de recursos é responsável por gerenciar os recursos de baixo nível, e.g., espaço em disco, volume, e CPU/GPU. Outra camada básica é a camada de agendamento que visa o uso eficiente dos recursos do *cluster*, e.g., replicação e dimensionamento do número de réplicas. Por último a camada de gerenciamento de serviços é responsável por disponibilizar recursos para possibilitam criar e implantar aplicativos corporativos, e.g., cgroups e namespaces ou dependência entre microsserviços.

Tabela 1. Comparativo dos orquestradores sobre a implantação e escalabilidade de contêineres.

Orquestrador	Escalabilidade	Implantação de contêineres
Kubernetes	O componente Replication Controller é o responsável pela função	Os *pods* são responsáveis pelo controle de distribuição dos contêineres
Docker Swarm	Além da escalabilidade padrão, permite a escalabilidade de serviços	Permite implantar varias imagens a partir de uma única imagem já implantada através de *slaves*
Apache Mesos	Kernel é escalável e resistente para permitir que estruturas compartilhem *clusters*	Os masters implementam o compartilhamento entre estruturas, e lista os recursos disponíveis nos *slaves*

Como critérios de análise de segurança foram utilizados os seguintes critérios: (i) controle e limitação de recursos e comunicação entre contêineres; (ii) controle e limitação de recursos evidenciou mais três critérios: (iia) configuração indevida do AppArmor, (iib) permissão indevida de capacidades do núcleo; e (iic) não filtragem de chamadas Seccomp. Estes critérios permitiram analisar a segurança com um viés bem delimitado. Baseado nestes critérios, foi possível identificar algumas vulnerabilidades e possíveis formas de reduzir o seu impacto em grandes nuvens de contêineres (Tabela 2).

Tabela 2. Resultados obtidos na análise de segurança.

Ataques/Vulnerabilidades	Possíveis soluções
Escalação de privilégios	Verificar se o mecanismo AppArmor está ativado, e ou utilizar o *profile* padrão do Docker
Negação de serviço; Execução de código arbitrário	Utilizar apenas as capacidades de *profile* padrão
Escalação de privilégios	Utilizar o *profile* padrão do Docker com as permissões limitadas a entorno de 311 chamadas de sistema
ARP Spoofing, *Man-in-the-Middle*	Cifrar o canal de comunicação, e ou uso de redes definidas pelo usuário

Em relação ao desempenho da orquestração de contêineres as literaturas apontaram a análise do tempo de implantação de um *cluster* e o tempo de *failover* como critérios importantes para analisar o desempenho de contêineres. Esta análise foi conduzida na nuvem OpenStack do LabP2D/UDESC, a Nuvem Tche. Foram criados ambientes de contêineres com orquestradores usando o sistema operacional GNU/Linux Ubuntu, e auxilio da ferramenta arcabouço nativa do Docker. Os resultados obtidos evidenciaram que o Kubernetes obteve o melhor resultado geral em relação aos outros dois orquestradores analisados, mas, isso não significa que seus concorrentes não tem pontos positivos. O Docker Swarm, por exemplo, forneceu o melhor tempo de *failover* e o Apache Mesos apresentou pontos positivos em relação a implantação do *cluster*.

Palavras-chave: Orquestração de contêineres, Segurança.