**Análise e Desenvolvimento de Sistemas**

**Construção de uma imagem Docker de uma aplicação baseada em Django**

Maria Luiza Vasconcellos Zanin

Taubaté, 01 de março de 2024

Faculdade Senai de Taubaté

**O que é virtualização?**

A virtualização é um procedimento que estabelece um ambiente simulado em um servidor existente, permitindo a execução de um programa específico sem interferir nos demais serviços oferecidos pelo servidor para outros usuários.

Esse ambiente virtual pode consistir em uma única instância ou em uma combinação de diversos elementos, tais como sistemas operacionais, servidores de rede, ambientes de computação, dispositivos de armazenamento, entre outros.

**Tipos de virtualização**

A **virtualização de hardware**, também conhecida como virtualização assistida por hardware ou virtualização de servidor, é baseada na ideia de que um único hardware ou servidor físico pode ser dividido em vários segmentos menores, consolidando vários servidores físicos em servidores virtuais executados em um único servidor principal. Cada segmento menor pode hospedar uma máquina virtual, enquanto todo o conjunto de servidores é tratado como uma única entidade por qualquer processo que requeira hardware.

O hypervisor é responsável pela alocação de recursos de hardware. Isso resulta em maior poder de processamento devido à utilização otimizada do hardware e aumento do tempo de atividade do aplicativo.

Existem três subtipos principais:

- Virtualização Total: O software convidado não requer modificações, pois o hardware subjacente é totalmente simulado.

- Virtualização de Emulação: A máquina virtual simula o hardware e se torna independente dele. O sistema operacional convidado não requer modificações.

- Paravirtualização: O hardware não é simulado e o software convidado executa em domínios isolados.

A **virtualização de software** cria ambientes virtuais na máquina host, permitindo a execução de sistemas operacionais convidados.

Isso inclui:

-Virtualização do sistema operacional: Execução de vários sistemas operacionais -dentro do sistema operacional nativo.

-Virtualização de aplicativos: Execução de aplicativos em ambientes virtuais separados do sistema operacional nativo.

Virtualização de serviços: Hospedagem de processos e serviços específicos relacionados a um aplicativo em particular.

A virtualização de memória envolve a agregação de memória física de diferentes servidores em um único conjunto de memória virtualizada. Esse processo resulta em uma memória de trabalho contígua ampliada, oferecendo benefícios significativos. Por exemplo, em alguns sistemas operacionais, como o Microsoft Windows OS, uma parte do disco de armazenamento pode ser utilizada como extensão da RAM.

Os subtipos da virtualização de memória incluem o controle em nível de aplicativo, onde os aplicativos acessam diretamente o pool de memória, e o controle em nível do sistema operacional, onde o acesso ao pool de memória é intermediado pelo sistema operacional.

**Qual a diferença entre contêineres e máquinas virtuais?**

Contêineres e máquinas virtuais são tecnologias que desvinculam as aplicações de sua infraestrutura de TI, proporcionando independência e flexibilidade.

Um contêiner, um pacote de código de software completo com suas dependências, facilita a portabilidade das aplicações para diferentes dispositivos. Por outro lado, uma máquina virtual é uma réplica digital de uma máquina física, permitindo a execução de múltiplos sistemas operacionais no mesmo sistema hospedeiro. Ambas as tecnologias oferecem soluções eficazes para a implantação e execução de aplicações em ambientes diversos.

**Como o Docker adicionou dinâmica a esse processo?**

O Docker é uma plataforma de software que permite aos desenvolvedores empacotar, distribuir e executar aplicativos em contêineres. Ele fornece uma maneira consistente e portátil de empacotar aplicativos e suas dependências em contêineres isolados. Inclui ferramentas para criar, gerenciar e distribuir imagens de contêineres. Ele é usado principalmente no desenvolvimento de aplicativos, permitindo que os desenvolvedores criem e executem aplicativos em ambientes consistentes, independentemente do ambiente de execução.

O Docker é considerado uma ferramenta vantajosa para desenvolvedores devido à sua capacidade de permitir a seleção independente de linguagem de programação, sistema de gerenciamento de banco de dados e distribuição preferida. Para os sysadmins, suas vantagens são ainda mais significativas, visto que dispensa a configuração manual do servidor com todas as dependências da aplicação e não requer distinção entre máquinas físicas e virtuais, já que o Docker oferece suporte para ambas as plataformas.

O uso do Docker traz benefícios para a empresa como um todo, como a melhoria da agilidade no desenvolvimento de aplicações, graças à reutilização da mesma imagem nos ambientes de QA, STAGING e PROD, reduzindo assim o tempo de transição entre eles. Além disso, há uma economia de custos com hardware devido ao melhor gerenciamento e utilização dos recursos, e o overhead é minimizado em comparação com outras soluções, como a virtualização.

Com o Docker, a implementação de microservices torna-se mais viável, um conceito em ascensão na indústria de tecnologia, que envolve a divisão de uma aplicação em partes menores e independentes, cada uma executando tarefas específicas. Este tópico será abordado com mais detalhes posteriormente neste texto.

**Quais os serviços de containers que podemos utilizar em diferentes provedores cloud?**

**(Existem vários serviços, serão citados os principais)**

**Azure Kubernetes Service (AKS)**, uma oferta de Kubernetes gerenciado que simplifica ainda mais a implantação e o gerenciamento de aplicativos baseados em contêiner.

O AKS é uma solução gerenciada disponibilizada pelo Microsoft Azure para simplificar a implantação, a administração e a escalabilidade de aplicativos em contêineres utilizando o Kubernetes, que é reconhecido como o sistema de orquestração de contêineres mais utilizado globalmente. Ao optar pelo AKS, é possível usufruir dos recursos do Kubernetes sem lidar com a complexidade e a demanda de gerenciar sua própria infraestrutura Kubernetes.

O **Google Cloud Run** é uma plataforma de computação gerenciada que permite a execução de contêineres diretamente na infraestrutura escalável do Google.

É possível pode implantar qualquer código, independentemente da linguagem de programação utilizada, no Cloud Run, desde que seja possível criar uma imagem de contêiner baseada nele.

O **AWS Fargate** é uma tecnologia que pode ser utilizada em conjunto com o Amazon ECS para executar contêineres sem a necessidade de gerenciar servidores ou clusters de instâncias do Amazon EC2. Com o AWS Fargate, não é mais necessário realizar provisionamento, configuração ou escalonamento de clusters de máquinas virtuais para executar contêineres. Isso elimina a necessidade de selecionar tipos de servidor, determinar o momento adequado para dimensionar os clusters ou otimizar o agrupamento de clusters.

Ao executar suas tarefas e serviços utilizando o tipo de inicialização do Fargate, você empacota sua aplicação em contêineres, define os requisitos de CPU e memória, estabelece as políticas de rede e IAM, e inicia a aplicação. Cada tarefa do Fargate possui seu próprio limite de isolamento e não compartilha o kernel subjacente, recursos de CPU, recursos de memória ou interface de rede elástica com outra tarefa.

**Quais critérios para tornar uma aplicação baseada em container?**

Os desenvolvedores adotam a estruturação em contêineres para criar e implantar aplicações modernas devido a várias vantagens. Primeiramente, a portabilidade é uma característica essencial, permitindo que as aplicações sejam implantadas em diversos ambientes sem a necessidade de reescrever o código do programa.

Isso significa que uma aplicação pode ser desenvolvida uma única vez e implantada em diferentes sistemas operacionais, como Linux e Windows, usando os mesmos contêineres. Além disso, a escalabilidade é facilitada pelos contêineres, que são componentes de software leves e eficientes. Diferentemente das máquinas virtuais, os contêineres podem iniciar aplicações mais rapidamente, uma vez que não precisam inicializar um sistema operacional separado. Isso permite que os desenvolvedores adicionem facilmente vários contêineres para diferentes aplicações em uma única máquina, sem que um interfira no funcionamento do outro.

A tolerância a falhas é outra vantagem importante dos contêineres. As equipes de desenvolvimento podem usar múltiplos contêineres para executar microsserviços na nuvem, garantindo que um contêiner com defeito não afete o desempenho dos demais. Isso aumenta a resiliência e a disponibilidade da aplicação.

Por fim, a agilidade é favorecida pelo ambiente isolado proporcionado pelos contêineres. Os desenvolvedores podem solucionar problemas e alterar o código da aplicação sem interferir no sistema operacional, no hardware ou em outros serviços de aplicações. Isso resulta em ciclos de lançamento de software mais curtos e na capacidade de trabalhar em atualizações rapidamente, graças ao modelo de contêiner.

Referências bibliográficas:

<https://www.qostecnologia.com.br/tipos-de-virtualizacao/>

<https://aws.amazon.com/pt/compare/the-difference-between-containers-and-virtual-machines/>

<https://waldyrfelix.com.br/docker-a-revolu%C3%A7%C3%A3o-dos-containers-e-o-que-voc%C3%AA-deve-saber-para-come%C3%A7ar-a-entender-dockerfile-imagens-3deaceaeafe0>

<https://livro.descomplicandodocker.com.br/chapters/chapter_02.html>

<https://learn.microsoft.com/pt-br/azure/aks/concepts-clusters-workloads>

<https://pt.linkedin.com/pulse/o-que-%C3%A9-azure-kubernetes-service-aks-wellington-ag%C3%A1pto>

<https://cloud.google.com/run/docs/overview/what-is-cloud-run?hl=pt-br#:~:text=O%20Cloud%20Run%20%C3%A9%20uma,de%20cont%C3%AAiner%20com%20base%20nele>.

<https://docs.aws.amazon.com/pt_br/AmazonECS/latest/developerguide/AWS_Fargate.html>

https://aws.amazon.com/pt/what-is/containerization/