UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS
CIÊNCIA DA COMPUTAÇÃO

ANA BEATRIZ STAHL
LUIZA FERNANDA DOS REIS
PATRÍCIA NAGEL

ARQUITETURAS VIRTUALIZADAS: APROFUNDAMENTO EM MÁQUINAS VIRTUAIS

São Leopoldo, RS
2024

SUMÁRIO

1 **REFERENCIAL TEÓRICO**

1.1 MÁQUINAS VIRTUAIS

1.1.1 **Conceito**

Uma máquina virtual é uma emulação ou modelo digital de um sistema de computador físico. Essas são comumente referidas como guests, enquanto a máquina física em que operam é conhecida como host.

O processo de virtualização possibilita a geração de uma variedade de máquinas virtuais, cada um operando com seu próprio sistema operacional (OS) e conjunto de aplicativos, dentro de um único sistema físico. Uma máquina virtual não pode interagir diretamente com a máquina física e requer um software chamado hypervisor para coordenar a interação com o hardware subjacente. Este hypervisor distribui recursos computacionais físicos, como processadores, memória e espaço de armazenamento para cada máquina virtual e as mantém independentes uma das outras para evitar a interferência de operação.

Esta tecnologia é conhecida por vários nomes, incluindo servidor virtual, instância de servidor virtual (VSI) e servidor privado virtual (VPS), embora neste artigo ela seja simplesmente referida como máquinas virtuais.

A virtualização funciona quando um hypervisor é usado em um computador físico ou servidor, permitindo separar o sistema operacional e os aplicativos do hardware. Isso cria várias "máquinas virtuais" independentes. Cada máquina virtual pode executar seu próprio sistema operacional e aplicativos, compartilhando os recursos, como memória e armazenamento, do servidor físico.

Há dois tipos de hypervisors. O tipo 1 é executado diretamente no hardware físico e geralmente utiliza um produto de software para criar VMs. Você pode selecionar um sistema operacional guest para instalação na VM e até usar uma VM como modelo para criar outras.

O hypervisor do tipo 2 é executado como um aplicativo dentro do sistema operacional do host. É usado na criação manual de uma VM na qual pode ser instalado um sistema operacional guest. Esse tipo de hypervisor pode configurar recursos físicos para a VM, como núcleos do processador e memória, e até opções de aceleração 3D para gráficos.

1.2 CONTAINERS

1.2.1 **Conceito**

De acordo com a IBM, contêineres são unidades executáveis de software nas quais o código do aplicativo é empacotado junto com suas bibliotecas e dependências, de formas comuns, para que o código possa ser executado em qualquer lugar.

Eles utilizam uma forma de virtualização a nível de sistema operacional na qual os recursos desse kernel da máquina host são utilizados para isolar os processos dos containers e controlar a quantidade de recursos de hardware que esses processos podem acessar, não precisam incluir um sistema operacional próprio. O contêiner em si é um processo sendo executado no sistema operacional da máquina host.

Alguns dos recursos do kernel Linux utilizados para implementar a essência de um contêiner são as ferramentas: chroot, que cria um ambiente isolado de sistema de arquivos para o processo, que só pode ver e interagir com o novo diretório raiz, simulando um novo sistema de arquivos, porém sem fornecer isolamento total, pois aspectos como rede e processos ainda compartilham o mesmo host. Unshare que é responsável então por criar os namespaces que vão isolar os diferentes recursos (processos, rede, IPC etc.) para garantir que o contêiner opere independentemente do sistema host. Nsenter que permite acessar os namespaces de processos existentes sendo útil para a administração e finalmente o cgroup que gerencia e limita o uso de recursos do sistema, garantindo que cada container tenha recursos dedicados sem interferir em outros contêineres ou no sistema host. Ou seja, a partir de uma máquina Linux somos capazes de realizar uma implementação simples de contêineres.

Mas não temos que realizar essas implementações manualmente, existem ferramentas atuais como o Docker, Podman (Red Hat), e LXC Linux Containers que implementam contêineres quase de forma automatizada com muito mais recursos para o usuário e ainda possui repositório de imagens já conteinerizadas de aplicações.

1.2.2 **Vantagens e desvantagens**

Os contêineres oferecem diversas vantagens frente a outras opções. Eles são altamente portáteis e leves, podendo ser executados em qualquer plataforma que

suporte o runtime de contêineres, o que os torna ideais para aplicações que precisam ser implantadas em diferentes ambientes, como nuvens pública e privada. Além disso, são mais eficientes em termos de recursos do que as máquinas virtuais, pois compartilham o kernel do sistema operacional do host, resultando em economia de custos operacionais.

A escalabilidade também é uma grande vantagem dos contêineres, permitindo que sejam facilmente dimensionados para atender às demandas de carga variáveis de forma rápida e eficiente. Em termos de implantação e gerenciamento, os contêineres são mais simples e diretos, sendo ideais para organizações que precisam de soluções ágeis e que não dispõem de recursos de TI dedicados para operações complexas de infraestrutura.

Já no apontamento das desvantagens, em termos de segurança e isolamento, os contêineres são menos robustos que as máquinas virtuais, uma vez que compartilham o kernel do sistema operacional host. Isso significa que um contêiner comprometido pode potencialmente afetar outros contêineres ou até mesmo o sistema operacional host, o que pode comprometer a integridade e a segurança dos dados e da máquina.

Em termos de gerenciamento, contêineres podem se tornar complexos em ambientes grandes e distribuídos. A necessidade de implantar e gerenciar um sistema de orquestração de contêineres, pode adicionar camadas adicionais de complexidade operacional e demandar habilidades específicas de administração de sistemas.

1.2.3 **Ferramentas conhecidas**

Com relação a ferramentas conhecidas de contêineres de aplicações, existem diversas como Podman, Docker, Linux Containers (LXC), entre outras. A ideia básica do contêiner de aplicação é que cada aplicação rode em um contêiner, de forma a isolá-la do sistema operacional e das demais aplicações.

O Docker é uma das ferramentas mais populares do momento sendo também um produto multiplataforma compatível com Linux, Windows e MacOS. Ele possui o Docker Hub que é um repositório público gigante de aplicativos de software em contêiner populares. Esses contêineres no Docker Hub podem ser baixados e implementados na hora em um tempo de execução local do Docker. Já para realizar a construção de uma imagem Docker (transformar sua aplicação em um contêiner),

utiliza-se um arquivo de instruções chamado de Docker file e nele deve constar as especificações da aplicação.

Temos também Linux Containers (LXC) que é uma ferramenta de contêiner Linux de código aberto. Seu objetivo era aproveitar as funcionalidades de namespaces e cgroups do kernel Linux para oferecer isolamento de processos e recursos de forma mais eficiente que as máquinas virtuais tradicionais. Desde então, o LXC evoluiu significativamente, tornando-se uma ferramenta robusta para a gestão de contêineres. É mais adequado para usuários que precisam de contêineres que se comportam mais como máquinas virtuais leves

Já no que diz respeito ao gerenciamento de contêineres temos por exemplo o Kubernetes que gerencia contêineres de aplicação, realizando sua orquestração. Isto é, ela cria e destrói contêineres conforme a necessidade (demanda) levando em conta também a reinício de contêineres com falhas, bem como automatiza as configurações necessárias para a execução de uma aplicação, ele ainda pode implementar balanceadores de carga entre os containers e roteamento de tráfego.

O Kubernetes foi desenvolvido originalmente pela Google e se tornou o grande padrão de orquestração de contêineres. Entre seus principais componentes estão: o cluster em si que consiste em um conjunto de servidores de processamento, chamados nós, que executam aplicações conteinerizadas. Todo cluster irá possuir ao menos um servidor de processamento chamado de worker node. Esses worker nodes hospedam os Pods, que são os componentes de uma aplicação. Existe também os componentes da camada de gerenciamento que tomam decisões globais sobre o cluster (por exemplo, alocação de Pods). E por fim, o kubelet que é um agente executado em cada nó no cluster garantindo que os contêineres estejam sendo executados em um Pod.

2 **APROFUNDAMENTO SOBRE MÁQUINAS VIRTUAIS**

As máquinas virtuais (VMs) têm sido um componente crucial na evolução da tecnologia da informação, permitindo a criação de ambientes de execução independentes dentro de um único sistema físico. Esta técnica de virtualização é fundamental para melhorar a utilização de recursos, otimizar a infraestrutura de TI e proporcionar flexibilidade para operações e desenvolvimento. Vamos explorar a arquitetura das máquinas virtuais, os tipos de hypervisors e as ferramentas de gerenciamento com mais detalhes.

2.1 ARQUITETURA

Uma máquina virtual (VM) é uma emulação de um computador físico, conhecida como guest, enquanto o computador físico que hospeda essa VM é denominado host. A virtualização permite a criação de múltiplas VMs em um único host, cada uma operando com seu próprio sistema operacional e aplicativos. As VMs não interagem diretamente com o hardware físico; em vez disso, utilizam um software chamado hypervisor, que aloca recursos físicos como processadores, memória e armazenamento para cada VM. O hypervisor também garante o isolamento entre as VMs para prevenir interferências entre elas.

2.2 FUNCIONAMENTO DE HYPERVISORS

Quando um hypervisor é utilizado em um servidor físico (bare metal), ele permite que o sistema operacional e os aplicativos sejam separados do hardware. Isso possibilita a criação de várias máquinas virtuais independentes, cada uma com seus próprios sistemas operacionais e aplicativos, compartilhando os recursos do servidor físico, como memória, RAM e armazenamento. O hypervisor gerencia a alocação desses recursos para cada máquina virtual, garantindo o uso eficiente e evitando interrupções.

Hypervisors do tipo 1 são executados diretamente no hardware físico, substituindo o sistema operacional do servidor. Eles utilizam um software distinto para criar e manipular VMs. Ferramentas de gerenciamento, como o vSphere da VMware, permitem selecionar e instalar um sistema operacional guest nas VMs. É possível

utilizar uma VM como modelo e duplicá-la para criar outras, atendendo a diversas necessidades, como testes de software, bancos de dados de produção e ambientes de desenvolvimento.

Hypervisors do tipo 2, por outro lado, são executados como um aplicativo dentro de um sistema operacional de host, geralmente em plataformas de desktop ou notebooks de usuário único. Com um hypervisor do tipo 2, é possível criar manualmente uma VM e instalar nela um sistema operacional guest. O hypervisor permite alocar recursos físicos para a VM, como processadores e memória, e configurar opções adicionais, como a aceleração 3D para gráficos, dependendo dos recursos disponíveis.

2.3 TIPOS DE MÁQUINAS VIRTUAIS

Existem dois tipos de máquinas virtuais que são as de sistemas e de processos. As máquinas de sistemas permitem a execução de um Sistema operacional completo em cima de outro Sistema Operacional, emulando um computador físico. Estas são utilizadas para criar ambientes de desenvolvimento isolados para testar novos sistemas, consolidar servidores e permitir a execução de múltiplos sistemas operacionais em um único hardware físico, como exemplo podemos citar VMware ESXi, Microsoft Hyper-V, Oracle VM VirtualBox e KVM. Já as máquinas virtuais de processo fornecem uma Plataforma de execução para um único processo ou aplicação, criando um ambiente isolado para rodar um aplicativo. A sua utilidade é a execução de programas em diferentes sistemas operacionais sem modificação do Código fonte, além de fornecer uma camada adicional de segurança e gerenciamento de recursos, como exemplo deste tipo de VM podemos mencionar a Java Virtual Machine (JVM), .NET Common Language Runtime (CLR), Python Virtual Machine (PVM) e BEAM.

2.4 VANTAGENS DE MÁQUINAS VIRTUAIS

As máquinas virtuais (VMs) oferecem diversos benefícios em comparação com o hardware físico tradicional. Em termos de utilização de recursos e retorno sobre investimento (ROI), várias VMs podem ser executadas em um único computador físico, eliminando a necessidade de adquirir novos servidores para cada sistema

operacional adicional, aumentando assim o retorno de cada peça de hardware existente.

Em termos de escala, a computação em nuvem facilita a implementação de múltiplas cópias dela VM, permitindo uma melhor acomodação de aumentos na carga de trabalho. A portabilidade das VMs permite que elas sejam realocadas entre diferentes computadores físicos em uma rede, possibilitando a alocação de cargas de trabalho em servidores com capacidade ociosa. Além disso, as VMs podem se mover entre ambientes locais e na nuvem, sendo úteis para cenários de nuvem híbrida.

A flexibilidade é outro benefício importante, pois criar uma VM é mais rápido e fácil do que instalar um sistema operacional em um servidor físico, permitindo que desenvolvedores e testadores de software criem ambientes sob demanda. Em termos de segurança, as VMs oferecem vantagens significativas, como a possibilidade de serem escaneadas externamente para detectar malwares, criar snapshots para rápida recuperação em caso de infecção e a capacidade de excluir e recriar rapidamente VMs comprometidas, acelerando a recuperação de problemas de segurança.

2.5 DESVANTAGES DE MÁQUINAS VIRTUAIS

Mesmo apresentando suas vantagens como flexibilidade e isolamento, melas possuem também suas desvantagens, como desempenho reduzido em comparação com hardware físico, competição por recursos quando várias VMs rodam no mesmo host, complexidade de gerenciamento em grande escala, sobrecarga do hypervisor, custos significativos de licenciamento e suporte, latência adicional ao acessar recursos de hardware, complicações no backup e recuperação, riscos de segurança devido a vulnerabilidades no hypervisor e consumo significativo de recursos de hardware, que pode exigir investimentos adicionais em infraestrutura. Avaliar essas desvantagens é crucial em relação às necessidades específicas do ambiente de TI.

2.6 CASOS DE USO DE MÁQUINAS VIRTUAIS

As máquinas virtuais (VMs) podem ser utilizadas de diversas maneiras por administradores de TI e usuários corporativos. Na computação em nuvem, as VMs têm sido a base fundamental, permitindo a execução e escalabilidade de diferentes tipos de aplicativos e cargas de trabalho. No suporte ao DevOps, as VMs permitem

que desenvolvedores criem modelos com configurações específicas para desenvolvimento e teste de software, integrando essas etapas em um fluxo de trabalho automatizado, o que aprimora a cadeia de ferramentas de DevOps.

As VMs também são úteis para testar novos sistemas operacionais em uma área de trabalho sem afetar o sistema operacional principal, investigar malware, e executar software incompatível. Por exemplo, usuários que preferem um sistema operacional, mas precisam de um programa disponível apenas em outro, podem utilizar VMs para acessar esse software. Além disso, a navegação segura é facilitada pelo uso de VMs, permitindo que os usuários visitem sites sem risco de infecções, restaurando a máquina a um estado anterior após cada sessão de navegação. Isso pode ser configurado pelo próprio usuário com um hypervisor de área de trabalho do tipo 2 ou fornecido por um administrador como uma área de trabalho virtual temporária no servidor.

3 **ESTUDO DE CASO**

3.1 DESCRIÇÃO DO ESTUDO DE CASO

Este estudo visa demonstrar a utilização do VirtualBox da Oracle para criação e gerenciamento de máquinas virtuais. Abordaremos a configuração básica da VM e a instalação do sistema operacional Ubuntu.

3.2 PASSOS PARA IMPLEMENTAÇÃO

3.2.1 **Instalação da VirtualBox**

Para iniciar, faça o download do VirtualBox no site oficial da Oracle (https://www.virtualbox.org/) e siga as instruções de instalação específicas para o seu sistema operacional.

3.2.2 **Criação de uma nova máquina virtual**

Abra o VirtualBox e clique em "Novo" para iniciar o assistente de criação de VM. Defina um nome para a VM, selecione o tipo e versão do sistema operacional guest (por exemplo, Ubuntu, Windows), e aloque recursos como memória RAM e espaço em disco conforme necessário.

3.2.3 **Configuração da VM**

Após criar a VM, selecione-a na lista do VirtualBox e clique em "Configurações". Configure opções adicionais como quantidade de processadores, tamanho do disco virtual e configurações de rede conforme suas necessidades.

3.2.4 **Instalação do Sistema Operacional**

Inicie a VM clicando em "Iniciar" no VirtualBox. Instale o sistema operacional dentro da VM utilizando uma imagem ISO adequada ao sistema operacional guest escolhido.

3.3 RESULTADOS ESPERADOS

Ao seguir esses passos, espera-se configurar uma máquina virtual utilizando o VirtualBox da Oracle, instalar um sistema operacional guest e configurar o sistema operacional na VM. Este ambiente virtual proporcionará um espaço isolado e funcional para testes e desenvolvimento de aplicativos, demonstrando a flexibilidade e eficácia do VirtualBox como uma solução de virtualização.

4 **REFERÊNCIAS**

Containers. Disponível em: <https://www.redhat.com/pt-br/topics/containers>. Acesso em: 30 jun. 2024.

FILHO, F. V. **Máquinas Virtuais e Containers**. Disponível em: <https://prof.valiante.info/aulas/arquitetura-de-computadores/máquinas-virtuais-e-containers>. Acesso em: 30 jun. 2024.

O que é um container? Disponível em: <https://aws.amazon.com/pt/containers/>. Acesso em: 30 jun. 2024.

O que é uma Máquina Virtual (VM)? Disponível em: <https://www.oracle.com/br/cloud/compute/virtual-machines/what-is-virtual-machine/>. Acesso em: 25 jun. 2024.

O que é uma VM (máquina virtual)? Disponível em: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-a-virtual-machine>. Acesso em: 25 jun. 2024.

O que são contêineres? Disponível em: <https://cloud.google.com/learn/what-are-containers>. Acesso em: 25 jun. 2024a.

O que são contêineres? Disponível em: <https://www.ibm.com/br-pt/topics/containers>. Acesso em: 25 jun. 2024b.

O que são máquinas virtuais (VMs)? Disponível em: <https://www.ibm.com/br-pt/topics/virtual-machines>. Acesso em: 25 jun. 2024.