

Explorando a Eficiência e Escalabilidade com Containers e Máquinas Virtuais no Desenvolvimento de um Sistema Web

Fernando Souza Rodrigues¹, Paulo Ferreira da Silva Júnior¹, Thiago Vieira Camara¹

¹Departamento de Ciência da Computação – Universidade Federal de Roraima (UFRR)
Av. Capitão Ene Garcez, 2413 - Bairro Aeroporto, Boa Vista - Roraima, RR.

fernando.rodrigues@ufrr.br, juniorrkcm@outlook.com,
thiagocamarasp@gmail.com

Abstract. *Este artigo apresenta um projeto de Sistemas Operacionais para criar uma rede de computadores virtuais utilizando Máquinas Virtuais (VMs) e Containers. São realizadas avaliações comparativas entre VMs e Containers, considerando desempenho, isolamento, escalabilidade e recursos necessários. O processo de instalação de software para criação de VMs e Containers é abordado, assim como os procedimentos para criação de clusters. Devido a limitações do projeto, não foi possível implementar o sistema web escalável com Containers, optando-se por configurar um servidor Apache e desenvolver um sistema web básico com HTML, CSS e PHP. Além disso, são apresentados três sistemas operacionais especializados em virtualização. O artigo oferece uma visão abrangente sobre a criação de redes de computadores virtuais, abordando virtualização, opções de implementação do sistema web e sistemas operacionais voltados para virtualização.*

Resumo. *This article presents an Operating Systems project to create a network of virtual computers using Virtual Machines (VMs) and Containers. Comparative evaluations between VMs and Containers are conducted, considering performance, isolation, scalability, and resource requirements. The software installation process for VMs and Containers creation is addressed, as well as the procedures for creating clusters. Due to project limitations, it was not possible to implement the scalable web system using Containers. Instead, an Apache server was configured, and a basic web system was developed using HTML, CSS and PHP. Additionally, three operating systems specialized in virtualization are presented. The article provides a comprehensive overview of creating virtual computer networks, covering virtualization, options for implementing the web system, and operating systems focused on virtualization.*

1. Introdução

A virtualização é uma tecnologia que permite que vários sistemas operacionais sejam executados em um único servidor físico [STRONG 2019]. Já a containerização é uma tecnologia que permite que várias aplicações sejam executadas em um único sistema operacional [KATHRIN, 2022] permitindo que os desenvolvedores criem e implementem aplicativos de forma rápida e segura, seja no método tradicional (aplicativo de camada única) ou um microsserviço modular [STRONG 2019].

Em computação, Clusters são grupos de computadores que trabalham juntos para executar tarefas [PEDRO 2022]. Em relação a virtualização e containerização, um cluster pode ser composto por várias máquinas virtuais ou contêineres que trabalham juntos para executar uma tarefa específica [APOLINARIO et al. 2023].

Um cluster de VMs (Virtual Machine) é um grupo de máquinas virtuais que trabalham juntas para executar uma tarefa específica. Cada máquina virtual é executada em um servidor físico separado e é gerenciada por um software de gerenciamento de cluster [LYNCH 2023]. O software de gerenciamento de cluster coordena as atividades das máquinas virtuais e garante que elas trabalhem juntas para executar a tarefa [ORACLE 2023].

Um cluster de contêineres é semelhante a um cluster de VMs, mas em vez de máquinas virtuais, ele usa contêineres [BUCHANAN 2023]. Cada contêiner é executado em um servidor físico separado e é gerenciado por um software de gerenciamento de clusters. O software de gerenciamento de cluster coordena as atividades dos contêineres e garante que eles trabalhem juntos para executar a tarefa [APOLINARIO et al. 2023].

Este relatório apresenta uma rede de 3 (três) computadores virtuais interconectados por meio de Máquinas Virtuais (VMs) e Containers. Inicialmente, é realizada uma avaliação comparativa entre VMs e Containers, analisando suas vantagens e desvantagens em termos de desempenho, isolamento, escalabilidade e recursos necessários. Em seguida, são descritos os processos de instalação de software para criação de VMs e Containers, incluindo os requisitos e considerações relevantes. O relatório também aborda a criação de clusters tanto com VMs quanto com Containers, detalhando os procedimentos para interconexão. Nosso relatório também tentou implementar um sistema web escalável utilizando Containers, no entanto, devido a limitações e restrições do projeto, não foi possível desenvolvê-lo para objetivos específicos, como inicialmente planejado. Em vez disso, adotou-se uma abordagem alternativa, onde um servidor Apache foi configurado na máquina 1, e foram desenvolvidos arquivos HTML e CSS com um formulário e código em PHP para interagir com um banco de dados. O uso de um software de gerenciamento de Containers, como por exemplo Docker, é aplicado para facilitar a administração do sistema. Por fim, são apresentados três sistemas operacionais focados em virtualização de sistemas operacionais, destacando suas características e recursos específicos.

2. Comparativo entre virtualização e containers

Contêineres e máquinas virtuais são tecnologias de virtualização muito semelhantes em diferentes aspectos [BUCHANAN 2023].

Primeiramente, um contêiner é um pacote, leve e isolado para executar um aplicativo no sistema operacional do host. Os contêineres se baseiam no kernel do sistema operacional do computador hospedeiro (que pode ser considerado como o encanamento subterrâneo do sistema operacional) e contêm apenas aplicativos e algumas APIs e serviços leves do sistema operacional que são executados no modo de usuário, conforme imagem abaixo [STRONG 2019].

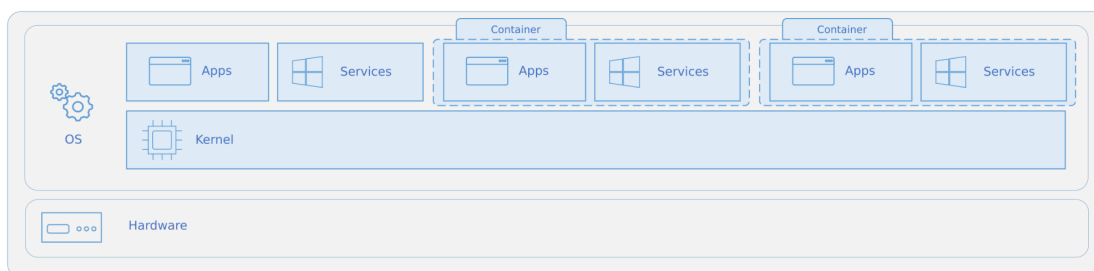


Figura 1. Diagrama de container

Ja máquinas virtuais são pacotes de software pesados que fornecem emulação completa de dispositivos de hardware de baixo nível, como CPU, disco e dispositivos de rede [STRONG 2019].

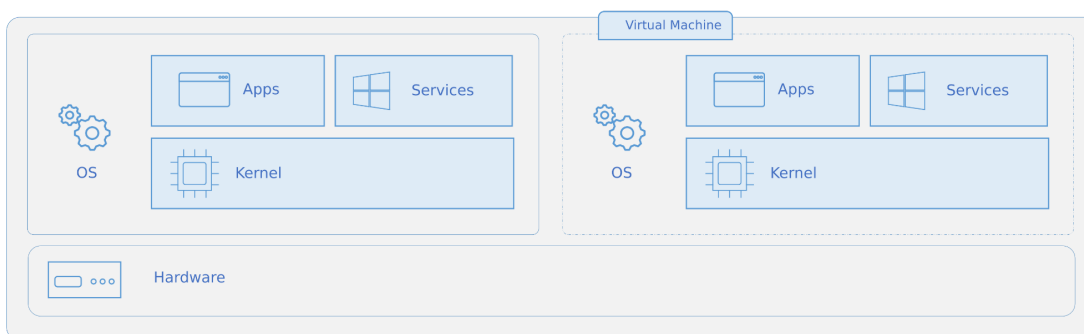


Figura 2. Diagrama de Virtual Machine

Fazendo um comparativo entre as duas funcionalidades, os containers oferecem uma rápida velocidade de iteração e um ecossistema robusto, com containers pré-construídos disponíveis em repositórios públicos. No entanto, eles apresentam o risco de explorações compartilhadas do host, pois compartilham o mesmo hardware subjacente. Por outro lado, as máquinas virtuais fornecem isolamento completo e segurança, permitindo o desenvolvimento interativo e configurações dinâmicas. No entanto, elas possuem uma velocidade de iteração mais lenta e consomem mais espaço de armazenamento em comparação com os contêineres. Em resumo, os contêineres se destacam em agilidade e ecossistema, enquanto as máquinas virtuais priorizam o isolamento e a segurança. A escolha entre os dois depende das necessidades específicas e dos compromissos da aplicação [BUCHANAN 2023].

3. Cluster

Em computação, cluster, que é uma palavra inglesa e significa “aglomeração”, denomina uma arquitetura de sistema capaz de combinar vários computadores como se fossem apenas um ou, também, ao grupo em si de computadores combinados [HOSTONE 2019].

Sua importância vai desde o aumento no poder de processamento, pois ele é igual ou superior a computadores mainframe (computadores projetados para processar um grande volume de dados simultaneamente [PEDRO 2022]) e outras formas de supercomputadores disponíveis no mercado, quanto ao custo-benefício pois implementar e manter um cluster, seja ele de qualquer tipo, é inferior aos gastos com computadores mainframe e outros. Outro ponto de importância para a implementação de clusters são a escalabilidade, onde a medida que o negócio cresce é possível adicionar nós (máquinas) para suprir as demandas, e a sua aplicabilidade pois é abordar de maneira eficiente os mais diversos problemas e objetivos, dos mais simples aos mais complexos [HOSTONE 2019].

3.1 Cluster de VMs

Para criação do cluster de máquinas virtuais utilizamos o Virtual Box da Oracle, um software de virtualização, ou seja, criação de um ou mais ambientes virtuais dentro de um mesmo computador ou servidor físico, desenvolvido pela Innotek e, posteriormente, foi comprada pela Oracle[LIRA 2021].

Primeiramente, tivemos que fazer download da ISO do sistema operacional Ubuntu 22.04 disponível neste [link](#). Após essa etapa, abrimos o software Virtual Box e clicamos em adicionar.

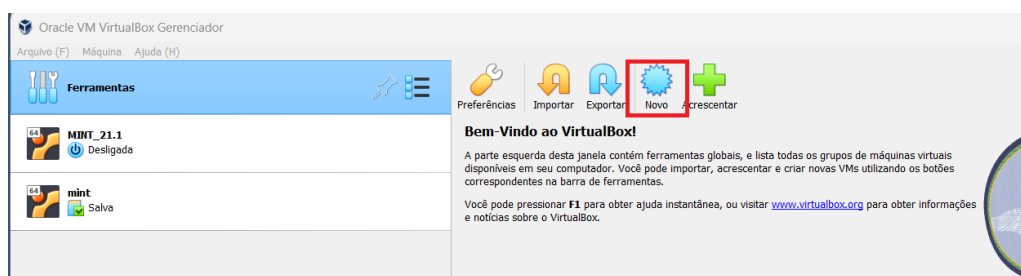


Figura 3. Criação de VM.

Após isso, devemos escolher o nome da ISO e selecionar a pasta onde foi previamente baixada.

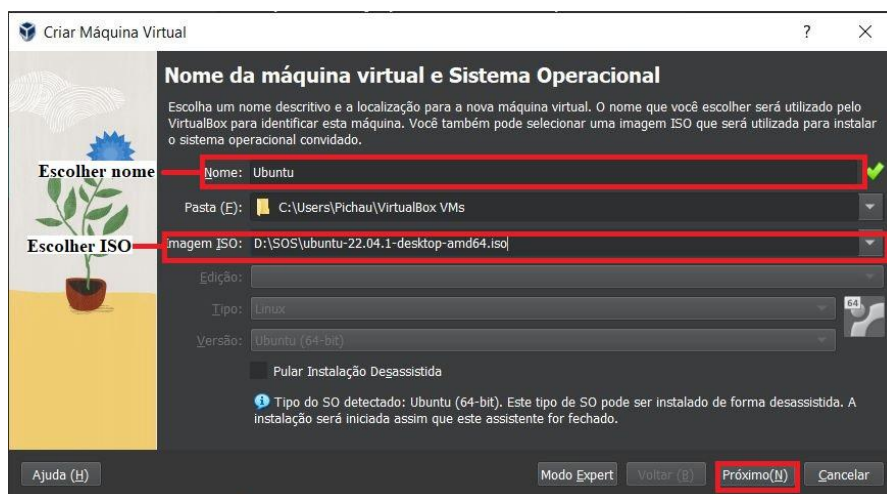


Figura 4. Escolha da ISO

Após clicar em Próximo é exibido a janela Hardware onde configuramos o hardware mínimo para rodar o nosso sistema operacional virtual.

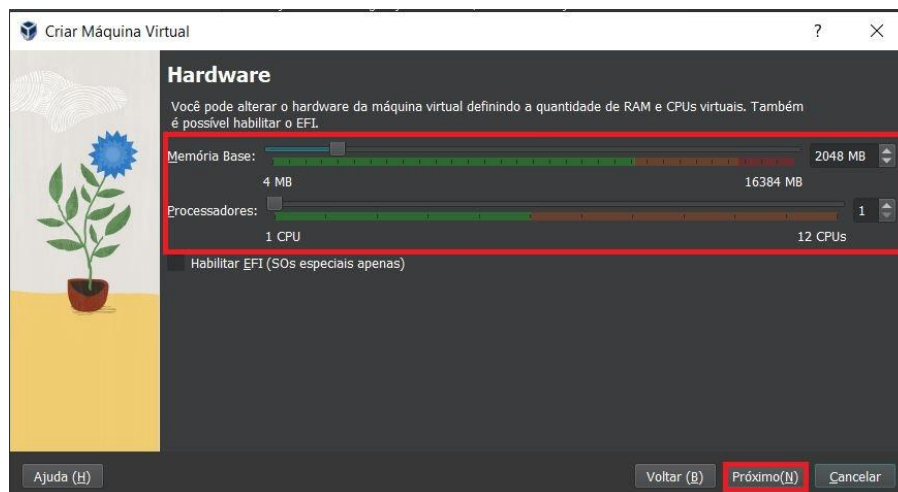


Figura 5. Hardware da máquina virtual

Ao clicar próximo temos uma outra janela chamada Disco Rígido Virtual onde vamos configurar o armazenamento mínimo do SO.

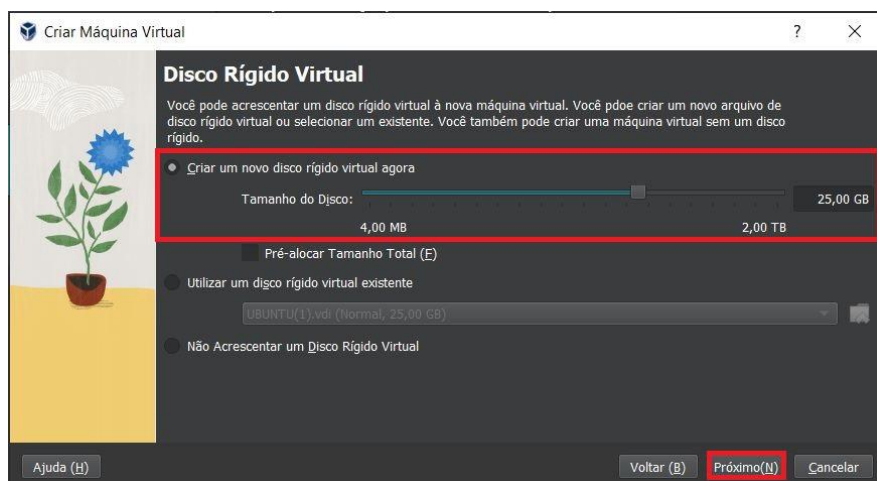


Figura 6. Configuração de armazenamento.

Finalmente temos uma tela com o resumo das configurações aplicadas ao SO, onde clicamos em Finalizar.

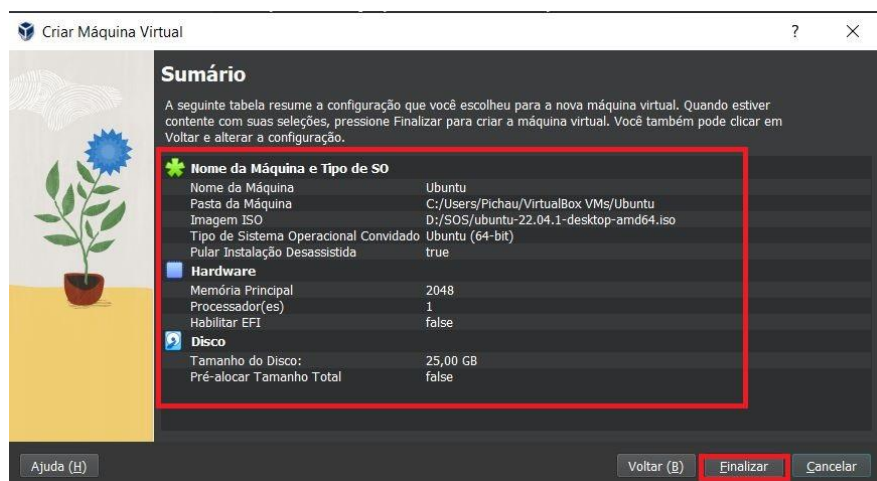


Figura 7. Finalização da configuração

Após isso, se todas configurações acima tiverem sido aplicadas corretamente, o SO inicializará normalmente.

Mas para a criação do cluster temos que modificar a configuração de rede do SO. Logo, após a inicialização do SO, procure o ícone de Configurações, no menu superior, e clique nele.



Figura 8. Configurações

Depois aparecerá uma janela com as principais configurações do SO, onde vamos procurar o ícone de Rede e, na aba do adaptador 1 na opção de conectado a escolhemos Placa em Modo Bridge.

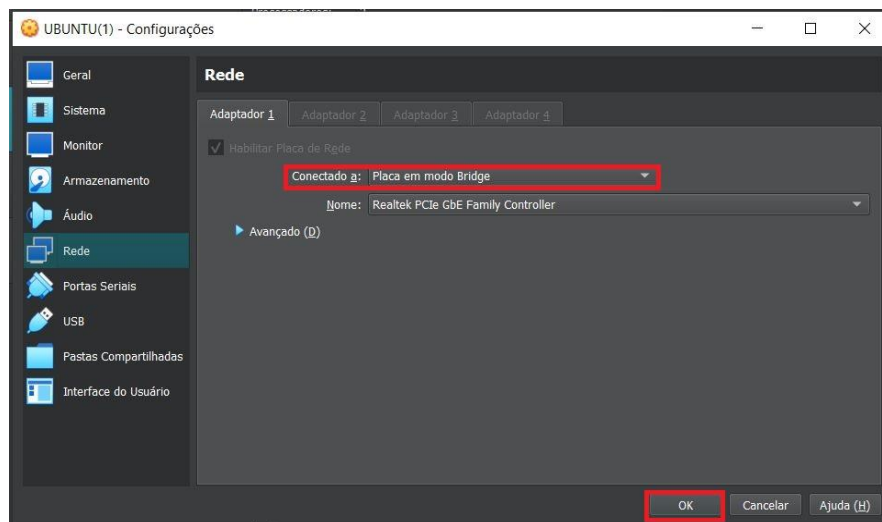


Figura 9, Placa de rede.

Criamos mais outras duas máquinas virtuais com a mesma ISO através da opção Novo no menu superior, seguindo os mesmos passos, tanto para instalação quanto para configuração da rede.

Após realizar todos os passos mencionados acima corretamente, podemos verificar a conexão entre as máquinas por meio do terminal usando o comando "ping" seguido do "endereço IP" da máquina adjacente. Dessa forma, podemos confirmar a conectividade entre elas. Ao concluir esse processo, teremos o cluster formado por três máquinas virtuais, como mostrado na imagem abaixo:

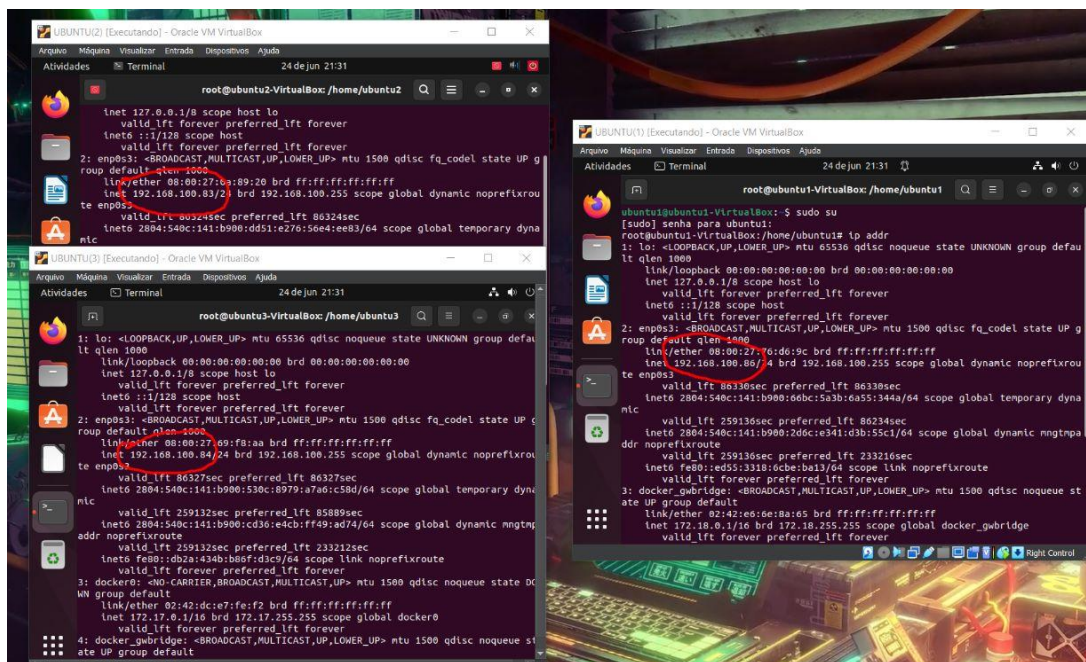


Figura 10. 3 VMs em cluster

3.2 Cluster de Containers

Para a criação do cluster de containers utilizamos o Docker que é uma plataforma lançada em 2013 e escrita em GO que possibilita o empacotamento de uma aplicação dentro de um container [SOARES 2023]. Utilizamos essa plataforma para fazer o gerenciamento dos containers da aplicação, pois é uma ferramenta já consolidada no mercado, através do Swarm, ferramenta nativa do Docker de clustering e agendamento para contêineres do Docker [FAWZY 2018].

Também utilizamos o VM VirtualBox, da Oracle, para criar as máquinas virtuais que serão utilizadas no cluster de containers através do Swarm.

Primeiro criamos 3 máquinas virtuais, 2 managers e 1 worker para simular o Cluster, com o sistema operacional Ubuntu 20.1 utilizando as configurações básicas para rodar os sistemas.

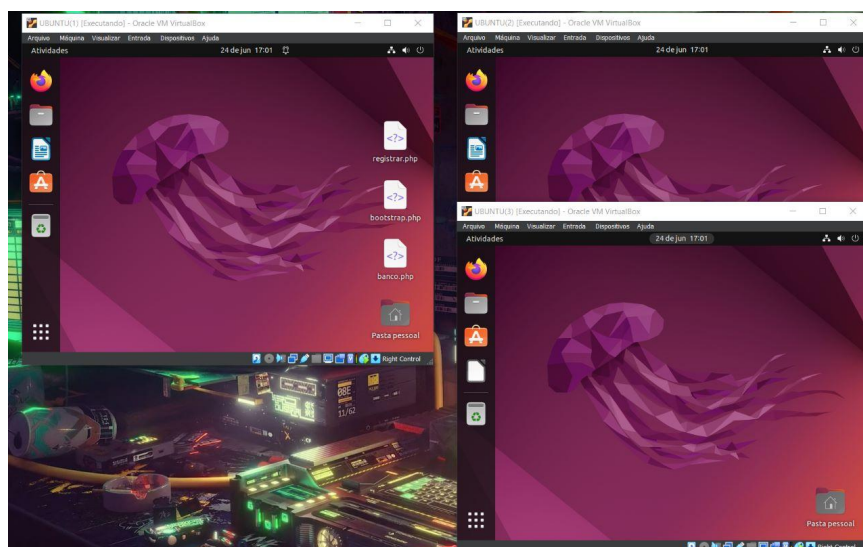


Figura 11. Imagem de VMs.

A primeira máquina será o manager do Swarm, e precisamos definir e dizer ao Swarm, no momento da inicialização do cluster, qual será a interface de rede que será usada quando existir mais de uma interface na máquina.

Depois, no terminal de comando de uma das máquinas rodamos o comando: *docker swarm init --advertise-addr "ip_da_instância"*, onde "ip_da_instância" é o ip da máquina que estamos rodando este comando. Esta máquina será o manager (leader) do Swarm, pois definimos no momento em que rodamos este código na inicialização do cluster.

O resultado deste comando será, como no exemplo a seguir: *docker swarm join --token"token_gerado"*. Esse token vai adicionar um worker (nó) ao cluster. Copie e cole esse comando no terminal de uma outra máquina virtual. O terminal vai retornar com a saída "This node joined a swarm as a worker". Aplique novamente este comando no terminal da outra máquina virtual e a mesma saída será exibida.

Para criar uma redundância para o manager (recomendado) devemos ter uma redundância para garantir a saúde do cluster e dos serviços que estão rodando. Para isso devemos rodar o comando *docker swarm join-token manager* na máquina leader do swarm. O comando anterior vai retornar um token para adicionar um manager, e esse token é diferente do que é usado no worker, como por exemplo: *docker swarm join --token "token_gerado"*. A saída do comando será essa: This node joined a swarm as a manager.

Para listar o nós do nosso cluster basta rodar o comando abaixo: *docker node ls*.

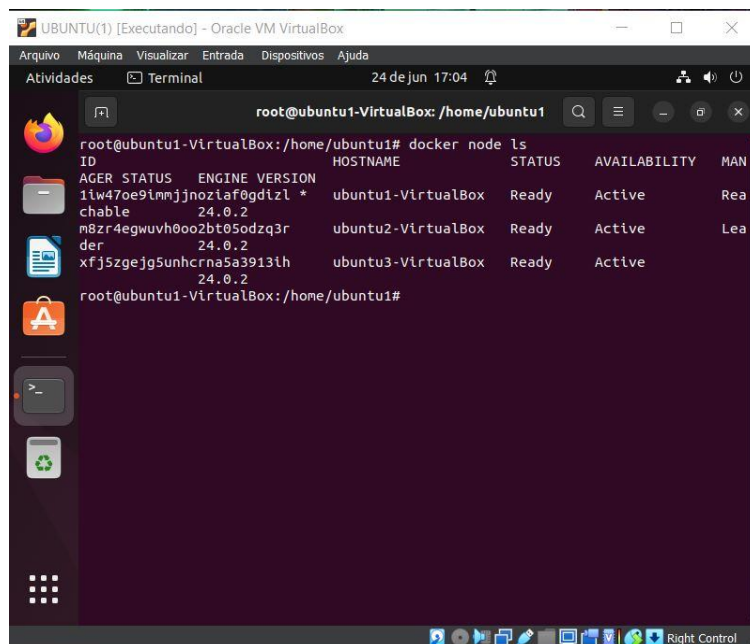


Figura 12. Terminal Linux.

Reparem que na última coluna, denominada MANAGER, é possível saber quais nós são managers, ou não. Quando em um nó não houver nenhuma informação na coluna MANAGER, é por que aquele nó é um worker.

4. Sistema Web

Este relatório descreve um sistema web baseado em cinco códigos principais, desenvolvidos utilizando PHP, HTML e CSS. O sistema é composto pelos seguintes arquivos:

- **banco.php:**

Este arquivo está localizado na pasta "include", e contém o código PHP responsável pela conexão com o banco de dados. Ele utiliza a função **mysqli_connect** para estabelecer a conexão com um servidor MySQL local, utilizando o usuário "root" e a senha vazia. O banco de dados utilizado é chamado "SO".

- **bootstrap.php:**

Este arquivo PHP inclui o arquivo "banco.php" utilizando a função **require_once**. Isso permite que outros arquivos tenham acesso à conexão com o banco de dados.

- **registrar_copy.php:**

Este arquivo contém códigos em HTML, CSS e PHP e representa um formulário de cadastro. O formulário coleta informações como nome completo, email, senha e confirmação de senha. Além disso, possui uma caixa de seleção onde o

usuário deve marcar a opção "Eu li e aceito os termos". Após o preenchimento do formulário, a senha é criptografada utilizando o algoritmo de hash SHA256 e os dados são enviados para serem armazenados em um banco de dados no phpMyAdmin.

- **login.php:**

Este arquivo contém códigos em PHP, HTML e CSS e é responsável pela autenticação de usuários cadastrados. Ele exibe um formulário onde o usuário pode inserir seu email e senha. Ao pressionar o botão de login, o sistema verifica as credenciais fornecidas pelo usuário e permite o acesso à página "dashboard.php" se forem válidas.

- **dashboard.php:**

Este arquivo possui uma mensagem simples que exibe "Login aprovado, bem-vindo ao sistema". Ele representa a página de destino para os usuários autenticados.

Além dos arquivos mencionados, foi configurado um servidor web Apache utilizando o XAMPP na máquina "ubuntu1". As máquinas "ubuntu2" e "ubuntu3" têm acesso às páginas do sistema, incluindo a base de dados. A máquina "ubuntu1" hospeda o banco de dados, a máquina "ubuntu2" tem a página "registrar_copy.php" aberta e a máquina "ubuntu3" tem a página "login.php" aberta.

Essa configuração permite que os usuários acessem o sistema web, realizem o registro de novos usuários através do formulário de cadastro e façam login utilizando as credenciais corretas. Uma vez autenticados, eles são redirecionados para a página "dashboard.php", onde uma mensagem de boas-vindas é exibida.

5. SOs dedicados para virtualização

- Microsoft Hyper-V: O hyper-v fornece a virtualização de hardware, isso significa que a máquina virtual é executada em hardware virtual. Fornece suporte a vários sistemas operacionais como Windows, FreeBSD e Linux [MICROSOFT 2022].
- KVM: É uma tecnologia de virtualização open source que foi baseada no Linux. Uma coisa interessante é que o KVM já está incluso dentro do Linux se for uma versão maior que a 2.6.20, e outra função interessante é que o KVM transforma o Linux em um hipervisor tipo 1 (que é chamado de "bare-metal") porque todos os hipervisores precisam de componentes no nível do sistema operacional [REDHAT 2023].
- Proxmox: software de gerenciamento de máquinas virtuais e contêineres que tem como base o Debian Linux e ainda utiliza o KVM com uma interface bem simples e capaz de rodar em um navegador na internet, além disso ele é "Open Source", ou seja, é de código aberto e não oferece custos para uso [TECNOAPP 2020].

6 Resultados

Com o objetivo de avaliar o desempenho e a funcionalidade do sistema web desenvolvido, foram realizados testes em diferentes áreas-chave. Os testes foram conduzidos com base nos requisitos funcionais do sistema, que incluem o registro de novos usuários, autenticação de usuários existentes e exibição da página de dashboard após o login bem-sucedido.

- **Teste de Registro de Usuários:**

O objetivo deste teste foi verificar se o formulário de registro estava funcionando corretamente. Foram inseridos diferentes conjuntos de dados de registro, incluindo nomes, endereços de e-mail, senhas e confirmações de senha. O teste verificou se os dados foram devidamente validados e se a criptografia da senha utilizando o algoritmo SHA256 foi aplicada corretamente. Além disso, foi verificado se os dados foram armazenados corretamente no banco de dados. O teste foi considerado bem-sucedido se os dados de registro foram validados corretamente e armazenados sem erros.

- **Teste de Autenticação de Usuários:**

Este teste teve como objetivo verificar se o formulário de login estava funcionando corretamente. Foram inseridas diferentes combinações de e-mail e senha, incluindo credenciais válidas e inválidas. O teste verificou se o sistema foi capaz de autenticar os usuários corretamente com base nas credenciais fornecidas. O teste foi considerado bem-sucedido se os usuários autenticados foram redirecionados corretamente para a página de dashboard.

- **Teste de Funcionalidade da Página de Dashboard:**

Este teste visou verificar se a página de dashboard estava sendo exibida corretamente após o login bem-sucedido. Após autenticar com sucesso um usuário válido, o teste verificou se a mensagem de boas-vindas "Login aprovado, bem-vindo ao sistema" era exibida na página de dashboard. O teste foi considerado bem-sucedido se a mensagem de boas-vindas foi exibida corretamente.

7. Conclusão

Em conclusão, este relatório abordou a configuração de um sistema web utilizando tecnologias como PHP, HTML, CSS e um banco de dados MySQL. Embora o objetivo inicial fosse implementar um sistema web escalável utilizando Containers, foram utilizadas máquinas virtuais e um servidor Apache para hospedar o sistema devido a limitações e restrições do projeto.

Ao longo do relatório, foram mencionadas algumas tecnologias importantes, como virtualização e containerização. A virtualização permite a execução de vários sistemas operacionais em um único servidor físico, enquanto a containerização permite a execução de várias aplicações em um único sistema operacional, oferecendo benefícios de rapidez e segurança no desenvolvimento e implantação de aplicativos.

Além disso, foi mencionada a importância dos clusters, que são grupos de computadores que trabalham juntos para executar tarefas específicas. No contexto da virtualização e containerização, um cluster pode ser composto por várias máquinas virtuais ou contêineres, coordenados por software de gerenciamento de clusters.

Embora o sistema desenvolvido não tenha explorado plenamente as capacidades de escalabilidade e distribuição de carga dos clusters, ele demonstrou o uso de tecnologias web comuns, como o Apache, PHP e MySQL, para criar um sistema de cadastro e login básico. O sistema permitiu que usuários se registrassem, criptografassem suas senhas e as armazenassem em um banco de dados, e também permitiu que usuários autenticados acessassem uma página de dashboard.

No geral, este relatório forneceu uma visão geral do sistema web desenvolvido, destacando as tecnologias e os processos envolvidos. Embora o sistema possa ser considerado simples em termos de recursos avançados de escalabilidade e distribuição, ele serve como um ponto de partida para futuras implementações e explorações de tecnologias mais avançadas no campo da virtualização, containerização e sistemas web.

8. Referências

- APOLINARIO, V.; ERICAWE, GRANT, J.; BOHER, S.; COULTER, D.; LOHR, H., ROSS, E.; Gerend, Jason (2023). “Contêineres vs. máquinas virtuais”. Disponível em: <<https://learn.microsoft.com/pt-br/virtualization/windowscontainers/about/containers-vs-vm>>. Acesso em: 23 de junho de 2023.
- BUCHANAN, I. (2023). “Containers vs. virtual machines”. Disponível em: <<https://www.atlassian.com/microservices/cloud-computing/containers-vs-vm#:~:text=The%20key%20differentiator%20between%20containers,above%20the%20operating%20system%20level>>. Acesso em: 23 de junho de 2023.
- FAWZY, M. (2018). “Create Cluster using docker swarm”. Disponível em: <<https://medium.com/tech-tajawal/create-cluster-using-docker-swarm-94d7b2a10c43>>. Acesso em: 25 de junho de 2023.
- HOSTONE (2019). “Cluster: o que é e qual sua importância”. Disponível em: <<https://blog.hostone.com.br/cluster-o-que-e-e-qual-sua-importancia/>>. Acesso em: 25 de junho de 2023.
- KATHRIN (2022). “Containerização de Aplicações: App Containerization”. Disponível em: <<https://leadcomm.com.br/2020/06/05/conteinerizacao-de-aplicacoes-app-containerization/>>. Acesso em: 23 de junho de 2023.
- LIRA, M. (2021). “VirtualBox: Saiba o que é, como funciona e como instalar!”. Disponível em: <<https://blog.b2bstack.com.br/virtualbox/>>. Acesso em: 25 de junho de 2023.
- LYNCH, M. (2023). “What is a virtual machine cluster?”. Disponível em: <<https://www.thetechadvocate.org/what-is-a-virtual-machine-cluster-vm-cluster/#:~:t>>

[ext=It%20is%20a%20group%20of,management%20than%20standalone%20virtual%20machines.>](#). Acesso em: 23 de abril de 2023.

MICROSOFT (2022). “Introdução ao Hyper-V no Windows 10”. Disponível em:
<<https://learn.microsoft.com/pt-br/virtualization/hyper-v-on-windows/about/>>.
Acesso em: 24 de junho de 2023.

ORACLE (2023). “Gerenciar Redes de Clusters de VMs”. Disponível em:
<<https://docs.oracle.com/pt-br/iaas/exadata/doc/ecc-manage-vm-cluster-network.html>
>. Acesso em: 23 de junho de 2023.

PEDRO, W. (2022). “O que é um cluster?”. Disponível em:
<<https://tecnoblog.net/responde/o-que-e-um-cluster-em-computacao/>>. Acesso em:
23 de junho de 2023.

PEDRO, W. (2022). “O que é um mainframe?”. Disponível em:
<<https://tecnoblog.net/responde/o-que-e-mainframe/>>. Acesso em: 25 de junho de
2023.

REDHAT (2023). “KVM | Máquina virtual baseada em Kernel”. Disponível em:
<<https://www.redhat.com/pt-br/topics/virtualization/what-is-KVM>>. Acesso em: 24
de Junho de 2023.

SOARES, J. V. (2023). “O que é cluster de container? Saiba a diferença entre Docker e Kubernetes!”. Disponível em:
<[https://blog.binario.cloud/o-que-e-cluster-de-container-saiba-a-diferenca-entre-dock
er-e-kubernetes/](https://blog.binario.cloud/o-que-e-cluster-de-container-saiba-a-diferenca-entre-docker-e-kubernetes/)>. Acesso em: 25 de junho de 2023.

STRONG, Aaron (2019). “Containerization vs. Virtualization: What's the Difference?”. Disponível em:
<[TECNOAPP \(2020\). “Conhecendo o Proxmox - Gerenciador de virtualização gratuito”. Disponível em:
<\[https://www.tecnoapp.com.br/Noticias/19/Conhecendo-o-Proxmox--Gerenciador-de
-virtualizacao-gratuito\]\(https://www.tecnoapp.com.br/Noticias/19/Conhecendo-o-Proxmox--Gerenciador-de-virtualizacao-gratuito\)>. Acesso em: 24 de junho de 2023.](https://www.burwood.com/blog-archive/containerization-vs-virtualization#:~:text=Virtualization%20enables%20you%20to%20run.single%20virtual%20machine%20on%20server.>>. Acesso em: 23 de junho de 2023.</p></div><div data-bbox=)