

Atividade Prática: Explorando a Virtualização na Nuvem

Na engenharia de software moderna, a **Virtualização** é o coração da computação em nuvem (Cloud Computing). Antigamente, para testar softwares em diferentes sistemas operacionais, as empresas precisavam comprar computadores físicos caríssimos e realizar o chamado 'upgrade vertical' (comprar hardwares cada vez mais potentes). Hoje, usamos a virtualização para dividir um único supercomputador físico em várias Máquinas Virtuais (Guest/Convidado) ou em Containers de isolamento, economizando espaço, energia e custos.

Nesta atividade prática de **23 passos**, nós vamos usar apenas o navegador de internet para provisionar uma infraestrutura virtual na nuvem do GitHub. Vamos explorar conceitos vitais de virtualização: Hospedeiro (Host) vs. Convidado (Guest), Virtualização por Container (OS-based), SDN (Rede Definida por Software) e SDS (Armazenamento Definido por Software). Siga o passo a passo atentamente.

Parte 1: Provisionando o seu Sistema Convidado (Guest) na Nuvem

Nesta etapa, usaremos um serviço em nuvem que criará uma máquina virtualizada do tipo OS-based (Container) rodando em cima de um servidor físico Hospedeiro (Host) da Microsoft.

1. **1. Acesse o GitHub:** Abra uma nova aba no seu navegador e entre no site github.com. Caso não tenha uma conta, clique em 'Sign up' para criar uma gratuitamente. Se já tiver, faça o seu login ('Sign in').
2. **2. Crie um novo repositório:** No canto superior direito, clique no botão azul ou no ícone de '+' e selecione 'New repository' (Novo repositório).
3. **3. Configure o seu projeto:** No campo 'Repository name', digite: estudo-de-virtualizacao-proz. Certifique-se de que a opção 'Public' está marcada.
4. **4. Adicione a base de arquivos:** Marque a caixinha 'Add a README file'. Este arquivo texto servirá de base para criarmos nosso disco virtualizado.
5. **5. Confirme a criação:** Vá até o final da página e clique no botão verde 'Create repository' (Criar repositório).
6. **6. Inicie a Máquina Virtualizada (Codespace):** Na página inicial do seu repositório criado, procure pelo botão verde escrito 'Code', clique nele, selecione a aba 'Codespaces' e clique no botão verde 'Create codespace on main'.
7. **7. Aguarde o provisionamento:** Uma nova tela com um terminal escuro será carregada diretamente no seu navegador. O que está acontecendo por trás é puro conceito de virtualização: a nuvem do GitHub localizou um servidor físico (Hospedeiro) e, por meio de um software de gerenciamento, isolou uma 'fatia' desse computador para criar um sistema operacional Convidado (Guest) Linux leve (um container) especialmente para você em segundos!

Parte 2: Investigando o Sistema Operacional Convidado (Guest)

Agora que o editor de código e o terminal abriram no seu navegador, vamos investigar essa máquina virtualizada para provar que ela se comporta como um computador real e isolado.

8. **8. Abra o Terminal de Comandos:** Caso não haja uma aba preta na parte inferior escrita 'Terminal', clique no ícone de menu (três linhas horizontais no canto superior esquerdo), vá até Terminal e clique em 'New Terminal'.
9. **9. Descubra o Sistema Operacional (Guest):** No terminal (onde tem uma linha esperando você digitar), digite o comando abaixo e aperte Enter para descobrir as informações da máquina virtualizada em que estamos:

```
uname -a
```

Nota do Professor: Você verá na saída a palavra Linux e a versão do Kernel. Mesmo que o seu computador físico real use Windows ou macOS, a virtualização permite que você rode um Linux completo dentro do seu navegador!

10. **10. Verifique os recursos de hardware emulados:** Digite o comando abaixo e aperte Enter para visualizar a memória RAM emulada que o seu Hospedeiro separou para esta máquina:

```
free -h
```

11. **11. Veja o espaço de armazenamento virtual:** Digite o comando abaixo e aperte Enter para listar os discos de armazenamento:

```
df -h
```

Parte 3: Experimentando Armazenamento Definido por Software (SDS)

O SDS (Software Defined Storage) é uma camada de software que permite gerenciar discos físicos como se fossem discos virtuais flexíveis. Vamos criar um diretório que simula um volume de dados persistente.

12. **12. Crie um arquivo de dados local:** No menu lateral esquerdo do editor (área de arquivos), clique com o botão direito e selecione 'New File' (Novo Arquivo). Nomeie-o exatamente como meu-disco-virtual.txt.
13. **13. Escreva no disco virtual:** Abra esse arquivo recém-criado e digite a seguinte frase:

```
Este arquivo simula dados salvos em um disco virtualizado gerenciado por software (SDS).
```

14. **14. Salve o arquivo:** Pressione as teclas Ctrl + S (ou Cmd + S no Mac) para garantir que as alterações no arquivo foram gravadas no disco virtualizado.

Parte 4: Trabalhando com Virtualização por Containers (OS-based)

No mercado, a virtualização por container (OS-based) é extremamente popular porque ela não precisa emular um hardware inteiro como as Máquinas Virtuais clássicas (H-based). Os containers compartilham

o sistema operacional Hospedeiro, tornando tudo extremamente leve e rápido. Vamos subir um container secundário dentro da nossa máquina virtual!

15. **15. Crie a receita do container (Dockerfile):** No menu esquerdo, clique com o botão direito, selecione 'New File' e nomeie-o exatamente como Dockerfile (com 'D' maiúsculo e sem extensão).
16. **16. Adicione as instruções de virtualização:** Abra o arquivo Dockerfile e cole o código abaixo. Ele usará uma imagem ultra-leve do sistema Linux Alpine (apenas 5MB) para criar um container isolado:

```
# Usando uma imagem base Linux minimalista compartilhando o Kernel do Host
FROM alpine

# Criando uma pasta interna
WORKDIR /app

# Comando padrão executado ao iniciar o container
CMD ["echo", "Olá, aluno PROZ! Este texto vem de dentro de um Container virtualizado (OS-based)!"]
```

17. **17. Construa a imagem virtualizada:** No seu terminal (parte inferior), digite o comando abaixo para compilar o container e aperte Enter:

```
docker build -t meu-container-teste .
```

18. **18. Execute o container:** Agora, vamos dar partida nessa mini máquina virtualizada de forma isolada (como em uma caixa de areia ou sandbox). Digite no terminal e aperte Enter:

```
docker run --rm meu-container-teste
```

Nota do Professor: Veja que a mensagem definida no passo 16 foi impressa instantaneamente na tela! O container ligou, executou a tarefa de forma totalmente isolada e desligou. Isso é virtualização de alta eficiência!

Parte 5: Simulando Rede Definida por Software (SDN)

A SDN (Software Defined Networking) permite criar infraestruturas de redes lógicas (virtuais) por cima da rede física real. Vamos criar uma rede isolada no Docker para ver os containers conversando entre si de forma segura.

19. **19. Crie uma rede virtualizada local:** No terminal, digite o seguinte comando e aperte Enter:

```
docker network create minha-rede-virtual
```

20. **20. Verifique a rede virtual criada:** Digite o comando abaixo para listar as redes lógicas ativas no ambiente virtualizado:

```
docker network ls
```

Nota do Professor: A rede minha-rede-virtual que você acabou de criar agora existe de forma totalmente lógica dentro do seu computador de testes!

Parte 6: Encerramento e Rastreabilidade de Aprendizado

21. **21. Documente os testes da atividade:** No menu esquerdo, abra o arquivo README.md que foi criado no início da prática.
22. **22. Escreva as conclusões no README:** Apague o conteúdo padrão e cole o seguinte relatório de laboratório estruturado para registrar seu aprendizado:

```
# Relatório de Laboratório - Prática de Virtualização (Tema 7)

## 1. Conceitos Vivenciados
* Hospedeiro (Host): Servidores em nuvem do GitHub que sustentaram o ambiente.
* Convidado (Guest): O Codespace Linux acessado diretamente no navegador.
* Virtualização OS-based: Criação de uma imagem leve usando Docker e o sistema Alpine.
* SDN & SDS: Configuração de diretórios simulando discos e criação de redes virtuais via terminal.

## 2. Status da Prática
* Configuração do Codespace: [OK]
* Criação e Execução de Container: [OK]
* Criação de Rede Virtual (SDN): [OK]
```

23. **23. Salve e encerre a máquina virtual:** Salve o arquivo README.md (Ctrl + S). No menu do Codespace (canto inferior esquerdo, na barra azul), você pode clicar sobre o nome 'Codespace...' e escolher a opção de parar a execução do ambiente. Parabéns, você concluiu com maestria a sua prática de virtualização!