

DESVENDANDO A VIRTUALIZAÇÃO E CONTAINERS

Uma Jornada Essencial para o Estudante de Sistemas Operacionais da UFRR

INTRODUÇÃO

O ESSENCIAL DA VIRTUALIZAÇÃO E CONTAINERS

Nesta apresentação, exploraremos o projeto final da disciplina de Sistemas Operacionais na UFRR, focando na construção de uma rede de computadores virtuais utilizando Máquinas Virtuais (VMs) e Containers. Prepare-se para entender os fundamentos, as implementações e as ferramentas que impulsionam o futuro da computação distribuída.



CONCEITOS FUNDAMENTAIS

VMs, Containers e suas diferenças.



SISTEMA WEB ESCALÁVEL

Criando uma aplicação distribuída com Docker Compose.



IMPLEMENTAÇÃO PRÁTICA

Montando seu cluster de VMs e Docker Swarm.



GERENCIAMENTO E OTIMIZAÇÃO

Ferramentas e sistemas operacionais focados em virtualização.

MÃOS NA MASSA

INSTALAÇÃO DE SOFTWARES CHAVE

Para dar o primeiro passo em nosso projeto, é fundamental ter as ferramentas certas. Vamos começar com a instalação do VirtualBox e do Docker, as bases para a criação de ambientes virtuais e containerizados.

VIRTUALBOX

Uma plataforma robusta para a criação de máquinas virtuais, permitindo a execução de múltiplos sistemas operacionais em um único hardware.

```
# Ubuntu/Debian
sudo apt update
sudo apt install virtualbox

# Fedora
sudo dnf install VirtualBox

# Windows
Baixe em: https://www.virtualbox.org/wiki/Downloads
```

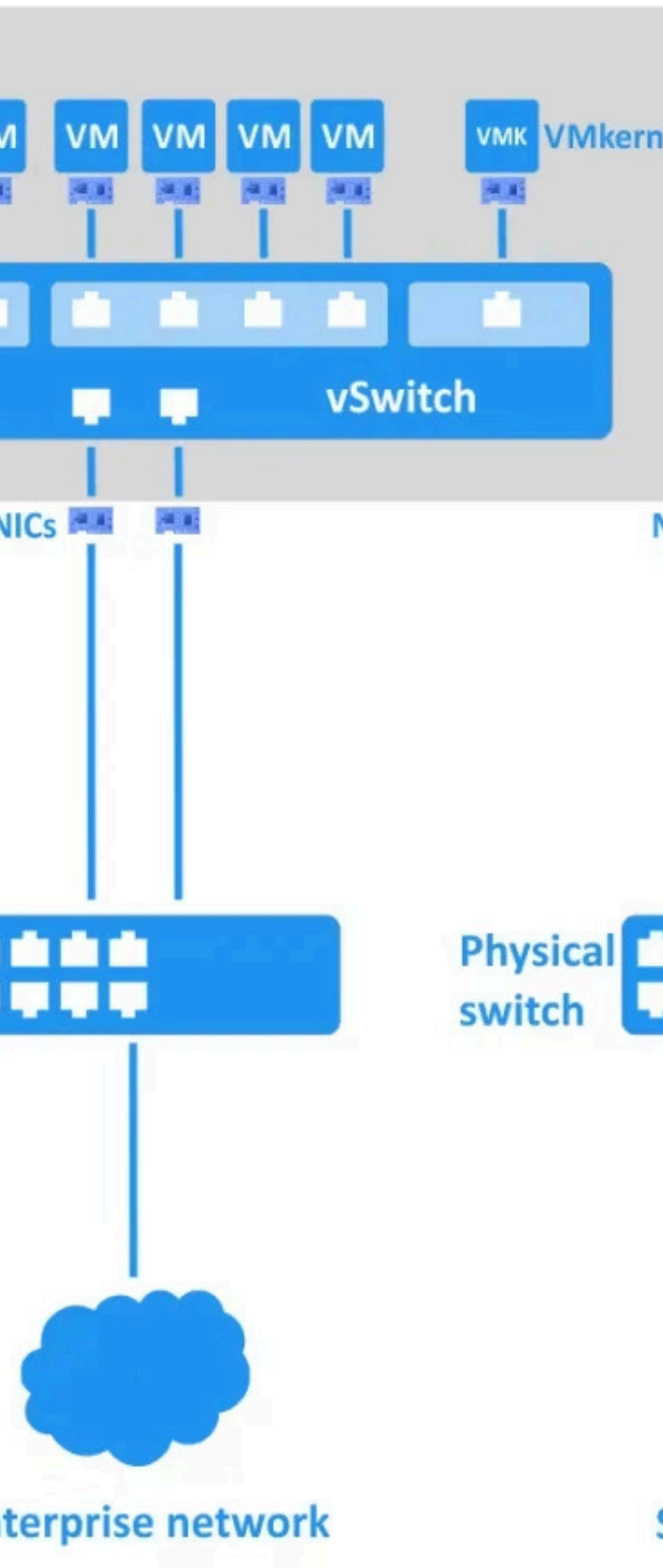
DOCKER

A ferramenta líder para containerização, que permite empacotar aplicações e suas dependências em ambientes leves e portáteis.

```
# Ubuntu
sudo apt update
sudo apt install docker.io
sudo systemctl start docker
sudo systemctl enable docker

# Fedora
sudo dnf install docker
sudo systemctl start docker
sudo systemctl enable docker

# Verificação
docker --version
```



FUNDAMENTOS

CLUSTER COM MÁQUINAS VIRTUAIS

O coração do nosso projeto começa com a criação de um cluster de VMs. Configuraremos três máquinas virtuais no VirtualBox, garantindo que elas possam se comunicar de forma isolada, simulando um ambiente de rede real.

1

Criação das VMs: Configure 3 VMs no VirtualBox (Ubuntu Server ou Fedora Server).

2

Modo de Rede: Defina como **Rede Interna** ou **Adaptador de Rede Paravirtualizado (VirtIO)** para comunicação interna.

3

IPs Estáticos: Atribua IPs fixos para cada VM (ex: 192.168.100.10, .11, .12).

4

Teste de Conectividade: Verifique a comunicação usando o comando `ping`.

CONSTRUINDO UM DOCKER SWARM

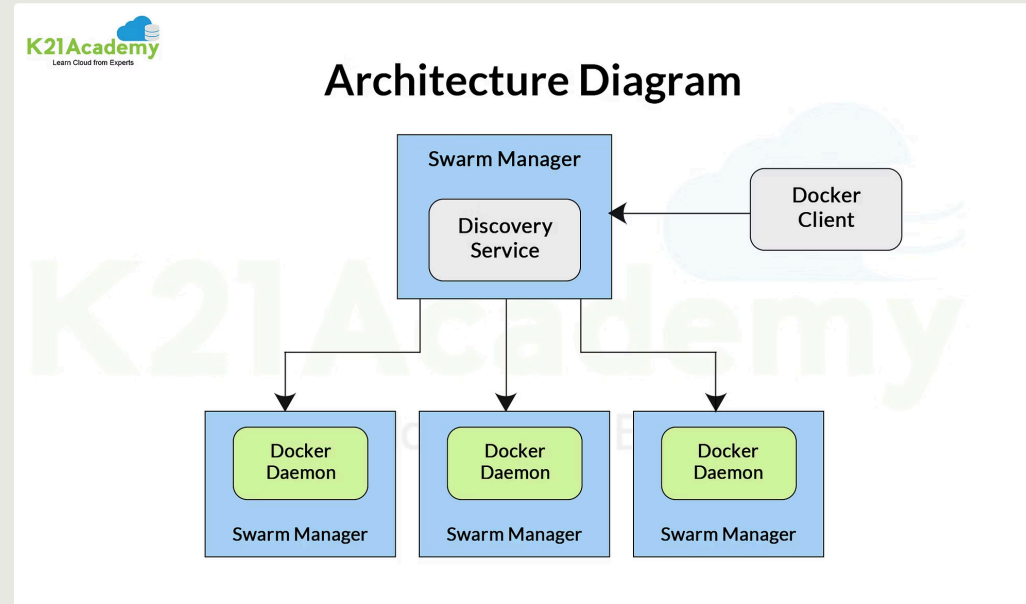
Após configurar as VMs, é hora de integrar os containers. Utilizaremos o Docker Swarm para transformar nossas máquinas virtuais em um cluster de containers, otimizando a orquestração e o gerenciamento de serviços.

O Docker Swarm permite que você crie um cluster de máquinas Docker, onde você pode implantar, gerenciar e escalar serviços.

```
# Em um dos nós (gerenciador)
docker swarm init --advertise-addr 192.168.100.10

# Nos outros nós (workers), use o token exibido:
docker swarm join --token 192.168.100.10:2377

# Verifique os nós no gerenciador:
docker node ls
```



ESCOLHAS ESTRATÉGICAS

VMS VS. CONTAINERS

A escolha entre Máquinas Virtuais e Containers é crucial para o desempenho e a flexibilidade do seu ambiente. Compreender suas vantagens e desvantagens é fundamental para o projeto.

Critério	Máquinas Virtuais	Containers
Desempenho	Mais pesado	Mais leve e rápido
Isolamento	Total (inclui kernel)	Parcial (compartilham kernel)
Portabilidade	Limitada	Alta (imagem leve)
Inicialização	Lento	Muito rápido
Uso de Recursos	Mais recursos	Menos recursos

SISTEMA WEB ESCALÁVEL COM CONTAINERS

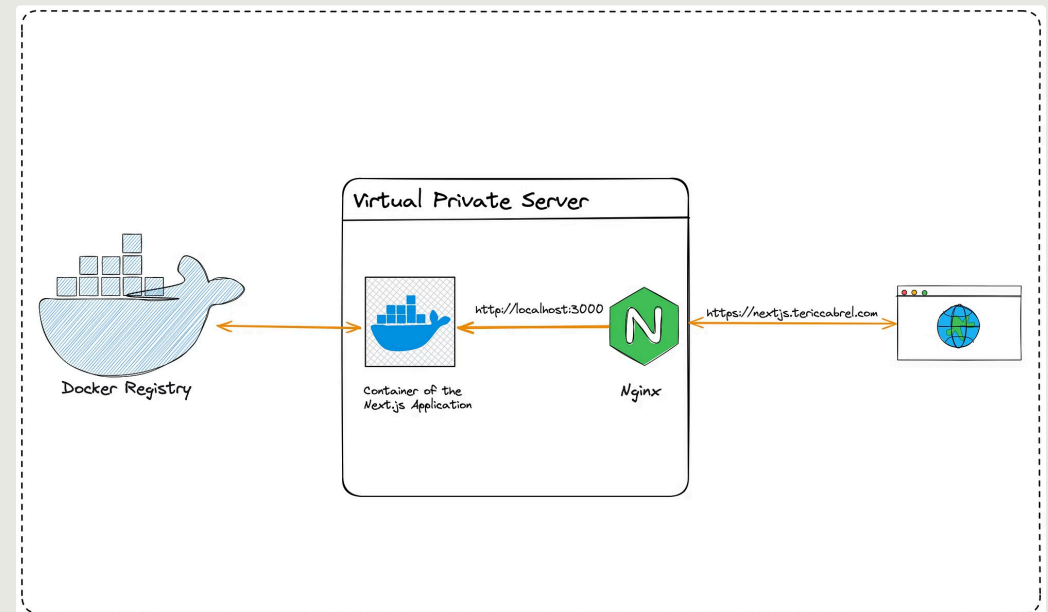
Vamos criar um sistema web completo e escalável, utilizando o Docker Compose para orquestrar um proxy reverso (NGINX), uma aplicação web (Node.js) e um banco de dados (PostgreSQL), tudo dentro de containers.

ESTRUTURA DO SERVIÇO

- **NGINX:** Como proxy reverso, direcionando requisições.
- **Aplicação Web:** Node.js (ou Flask) para a lógica de negócio.
- **Banco de Dados:** PostgreSQL para persistência de dados.

DOCKER-COMPOSE.YML

O Docker Compose atua como uma ferramenta que permite executar várias ações de forma independente do SO da máquina.



O arquivo `docker-compose.yml` define e orquestra os serviços, redes e volumes do nosso ambiente.

CONFIGURAÇÕES DO DOCKER COMPOSE

Detalhes dos arquivos `docker-compose.yml` e `nginx.conf` que orquestram a aplicação web escalável.

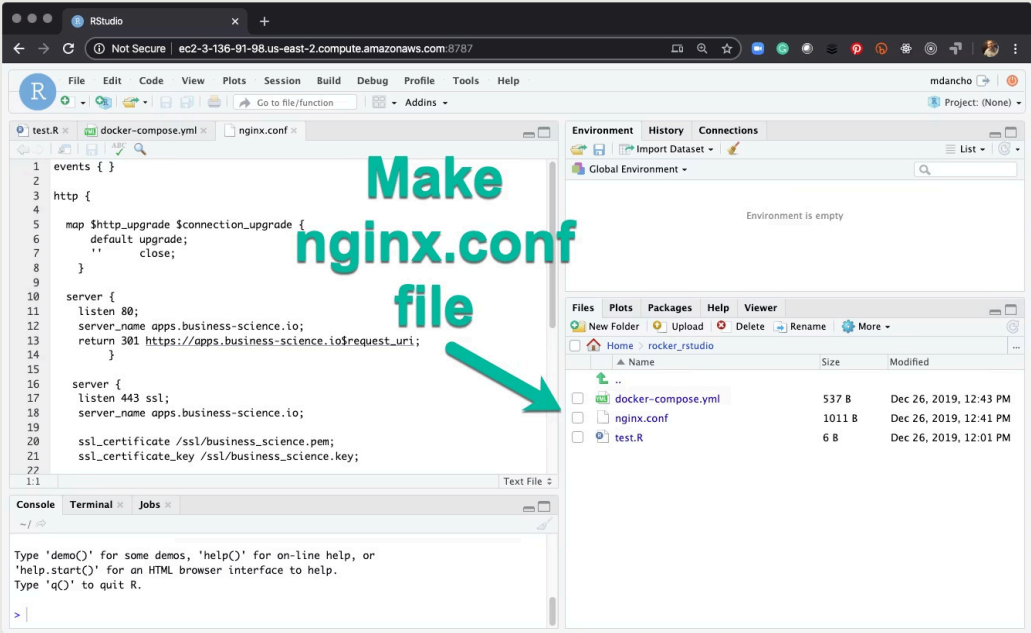
DOCKER-COMPOSE.YML

```
version: '3'
services:
  web:
    image: node:18
    volumes:
      - ./app:/app
    working_dir: /app
    command: node server.js
    ports:
      - "3000:3000"
  db:
    image: postgres:15
    environment:
      POSTGRES_USER: user
      POSTGRES_PASSWORD: pass
      POSTGRES_DB: appdb
    volumes:
      - pgdata:/var/lib/postgresql/data
  nginx:
    image: nginx:alpine
    ports:
      - "80:80"
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf
  volumes:
    pgdata:
```

NGINX.CONF

```
events {}
http {
  server {
    listen 80;
    location / {
      proxy_pass http://web:3000;
    }
  }
}
```

Esses arquivos definem a interconexão e o comportamento dos serviços. Eles são a base para a criação de um ambiente de microserviços eficiente e escalável.



FACILITANDO O GERENCIAMENTO

PORTAINER E SOS PARA VIRTUALIZAÇÃO

Para gerenciar nosso ambiente containerizado de forma eficiente, apresentamos o Portainer. Além disso, exploraremos sistemas operacionais otimizados para virtualização, que podem aprimorar ainda mais o desempenho e a administração dos seus projetos.

PORTAINER

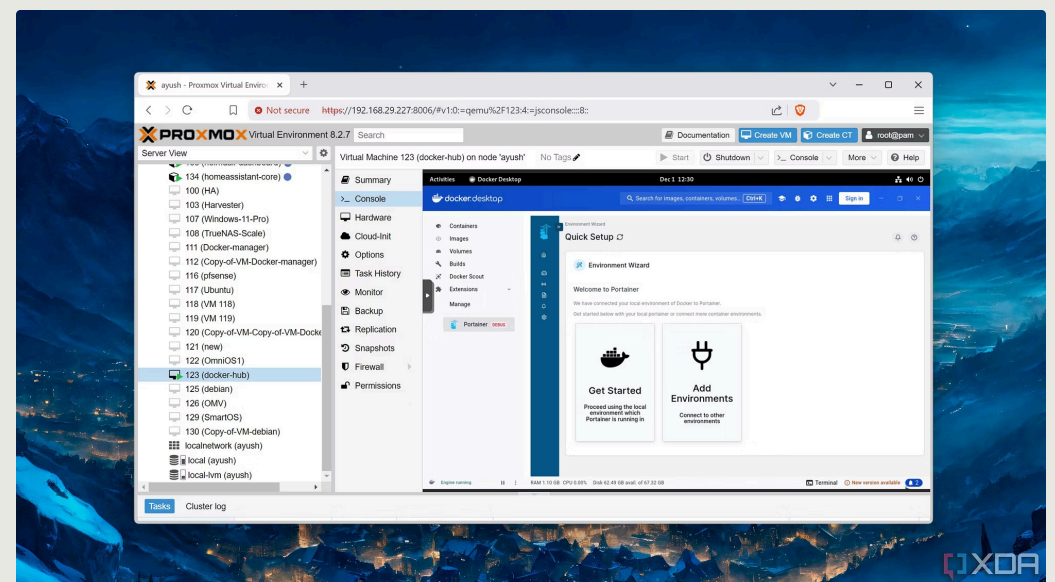
Uma poderosa ferramenta de gerenciamento para Docker, oferecendo uma interface gráfica intuitiva para orquestrar containers, volumes e redes.

```
docker volume create portainer_data
docker run -d -p 9000:9000 --name portainer --
restart=always -v
/var/run/docker.sock:/var/run/docker.sock -v
portainer_data:/data portainer/portainer-ce
```

Acesse: <http://localhost:9000>

SISTEMAS OPERACIONAIS FOCADOS

- **Proxmox VE:** Focado em virtualização KVM e containers LXC, com interface web integrada.
- **Ubuntu Server com KVM:** Suporte robusto a libvirt, KVM e virt-manager.
- **Fedora Server com Cockpit:** Ferramenta web para administração, incluindo máquinas virtuais.



CONCLUSÃO

SEU CAMINHO PARA A COMPUTAÇÃO DO FUTURO

Este projeto oferece uma visão prática e aprofundada das tecnologias de virtualização e containerização. Dominar essas ferramentas não é apenas uma exigência do mercado, mas uma habilidade fundamental para qualquer estudante de Sistemas Operacionais da UFRR que busca construir e gerenciar sistemas escaláveis e eficientes.



APRENDIZADO CONTÍNUO

A virtualização e os containers são campos em constante evolução. Mantenha-se atualizado!



INOVAÇÃO AO SEU ALCANCE

Utilize essas tecnologias para criar soluções inovadoras e resolver problemas complexos.



COLABORAÇÃO

Compartilhe seu conhecimento e colabore com outros estudantes e profissionais.

OBRIGADO!