Utilização de contêineres para encapsulamento de aplicações

Bianca Carvalho Oliveira
19/0103329
Turma 02

I. Resumo

O relatório apresenta o desenvolvimento de um arquivo `docker-compose.yml` que orquestra quatro serviços complementares e/ou interdependentes, simulando um cenário comercial realista. Cada serviço foi escolhido e configurado cuidadosamente para se integrar aos demais, formando um sistema coeso e funcional. O documento detalha as razões por trás da escolha de cada serviço, explicando como foram configurados para operar em conjunto e como essa configuração pode ser aplicada em um contexto prático.

***Index Terms*—Streaming, Sockets, Redes**

II. Introdução

A tecnologia de contêineres revolucionou o mundo do desenvolvimento de *software* e da implantação de aplicações. Contêineres, uma forma de virtualização a nível de sistema operacional, permitem que os desenvolvedores encapsulem suas aplicações juntamente com todas as suas dependências em um pacote autônomo. Este pacote, ou contêiner, pode então ser executado consistentemente em qualquer ambiente que suporte a tecnologia de contêineres, independentemente das especificidades do sistema operacional ou do *hardware* subjacente.

Os contêineres oferecem várias vantagens significativas em relação aos métodos tradicionais de implantação de aplicações. Eles são leves, pois compartilham o kernel do sistema operacional host e não requerem um sistema operacional completo para cada aplicação. Além disso, eles são portáteis entre diferentes plataformas e provedores de nuvem, facilitam a orquestração e o escalonamento de aplicações e aumentam a eficiência e a utilização de recursos

Apesar de suas inúmeras vantagens, a adoção de contêineres traz consigo desafios notáveis. Esses desafios abrangem a demanda por novas ferramentas e habilidades, inquietações relacionadas à segurança e a complexidade associada ao gerenciamento de contêineres em ampla escala. Este serviço propõe uma análise abrangente das oportunidades e desafios inerentes ao emprego de contêineres para encapsular aplicações.

III. Fundamentação Teórica

A tecnologia de contêineres fundamenta-se no princípio da virtualização ao nível do sistema operacional. Ao contrário da abordagem convencional de virtualização, que emula um sistema operacional completo para cada instância, a virtualização ao nível do sistema operacional permite que múltiplas instâncias (ou contêineres) compartilhem o mesmo kernel do sistema operacional. Esse método resulta em contêineres significativamente mais leves e eficientes se comparados às máquinas virtuais tradicionais.

Os contêineres são como pequenos compartimentos isolados uns dos outros e do sistema operacional principal. Cada um tem seu próprio espaço separado para o usuário. Isso significa que o que acontece em um contêiner não afeta os outros ou o sistema principal. Essa maneira de fazer as coisas traz bastante segurança e liberdade, permitindo que os desenvolvedores rodem seus programas em ambientes isolados e controlados.

A portabilidade é outra característica chave dos contêineres. Uma vez que um contêiner é criado, ele pode ser executado em qualquer máquina que suporte a tecnologia de contêineres, independentemente do sistema operacional ou do hardware subjacente. Isso facilita a migração de aplicações entre diferentes ambientes e plataformas, desde um laptop de desenvolvedor até um cluster de produção na nuvem

Contêineres são fáceis de levar para qualquer lugar. Depois de criar um, pode ser usado em qualquer máquina que suporte contêineres, não importa o sistema operacional ou hardware. Isso torna simples mudar suas aplicações de um laptop para um servidor na nuvem.

Arquivo docker-compose.yml

A seguir está o conteúdo do arquivo `docker-compose.yml` utilizado para configurar os serviços:

```
version: '3.8'

services:
  mysql:
    image: mysql:latest
    container_name: my-mysql
    environment:
      MYSQL_ROOT_PASSWORD: root_password
      MYSQL_DATABASE: nextcloud
      MYSQL_USER: nextcloud_user
      MYSQL_PASSWORD: nextcloud_password
```

```
    volumes:
      - mysql_data:/var/lib/mysql

  mysql-plex:
    image: mysql:latest
    container_name: my-mysql-plex
    environment:
      MYSQL_ROOT_PASSWORD: root_password
      MYSQL_DATABASE: plex
      MYSQL_USER: plex_user
      MYSQL_PASSWORD: plex_password
    volumes:
      - mysql_plex_data:/var/lib/mysql

  nginx:
    image: nginx:latest
    container_name: my-nginx
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./nginx/conf.d:/etc/nginx/conf.d
      - nextcloud_data:/var/www/html

  nextcloud:
    image: nextcloud:latest
    container_name: my-nextcloud
    depends_on:
      - mysql
    ports:
      - "8080:80"
    volumes:
      - nextcloud_data:/var/www/html
    environment:
      - MYSQL_HOST=mysql
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud_user
      - MYSQL_PASSWORD=
        ↪ nextcloud_password


  plex:
    image: plexinc/pms-docker:latest
    container_name: my-plex
    ports:
      - "32400:32400"
      - "3005:3005"
      - "8324:8324"
      - "32469:32469"
      - "1900:1900/udp"
      - "32410:32410/udp"
      - "32412:32412/udp"
      - "32413:32413/udp"
      - "32414:32414/udp"
    environment:
      - PLEX_UID=1000
      - PLEX_GID=1000
      - PLEX_CLAIM=""
      - ADVERTISE_IP="http
        ↪ ://192.168.18.172:32400/"
      - MYSQL_HOST=mysql-plex
      - MYSQL_DATABASE=plex
      - MYSQL_USER=plex_user
      - MYSQL_PASSWORD=plex_password

volumes:
  mysql_data:
  mysql_plex_data:
  nextcloud_data:
```

Listing 1. docker-compose.yml

EXPLICAÇÕES DETALHADAS

A seguir, forneço explicações detalhadas para cada parte do arquivo '`docker-compose.yml`':

1) **mysql:** Este serviço utiliza a imagem `mysql:latest`. Esse código é uma configuração para iniciar um serviço MySQL em um ambiente de contêineres usando o Docker Compose. Ele define a imagem a ser usada, as configurações do MySQL (senha do root, nome do banco de dados, usuário e senha), além de especificar um volume para persistência de dados.
2) ***mysql-plex:*** *Utiliza a imagem `mysql:latest`. Esse código configura um serviço MySQL específico para o Plex, um popular servidor de mídia. Ele define as configurações do MySQL, incluindo senha do root, nome do banco de dados, usuário e senha específicos para o Plex, e também utiliza um volume para persistência de dados relacionados ao Plex.*
3) ***Nginx:*** *Este serviço utiliza a imagem `nginx:latest`. Este código configura um serviço NGINX, um servidor web conhecido por sua eficiência e flexibilidade. Ele define as configurações da imagem, o nome do contêiner, as portas a serem mapeadas e os volumes para configurações personalizadas do NGINX e para compartilhar dados com o serviço Nextcloud.*
4) ***nextcloud:*** *Este serviço utiliza a imagem `nextcloud:latest`.Esse código configura um serviço Nextcloud em um ambiente de contêineres, estabelecendo as dependências necessárias com o serviço MySQL, mapeando portas para acesso externo, definindo volumes para persistência de dados e configurando variáveis de ambiente para a conexão com o banco de dados MySQL.*
5) ***plex:*** *Este serviço utiliza a imagem `plexinc/pms-docker:latest` Esse código configura um serviço Plex em um ambiente de contêineres, especificando detalhes como a imagem a ser usada, portas a serem mapeadas, configurações de ambiente, e detalhes de conexão ao banco de dados MySQL específico para o Plex.*
6) ***volumes::*** *Essa seção é crucial para garantir a persistência de dados em ambientes de contêineres. Ao definir volumes dedicados para serviços específicos, os dados são armazenados fora dos contêineres, facilitando a gestão e evitando a perda de dados quando os contêineres são manipulados (iniciados, parados, removidos). Cada volume pode ser mapeado para diretórios dentro dos contêineres correspondentes, permitindo o acesso aos dados necessários para o funcionamento dos serviços.*

RAZÃO DA ESCOLHA DOS SERVIÇOS

Este Docker Compose foi configurado para fornecer uma solução integrada para os serviços Nextcloud e Plex, com foco na modularidade e persistência de dados. O MySQL é utilizado para ambos, com um banco dedicado para cada serviço. O NGINX atua como ponto central, facilitando a comunicação entre o usuário e os serviços. O Nextcloud depende do MySQL, garantindo a disponibilidade do banco de dados antes de iniciar. O Plex, por sua vez, possui portas abertas para facilitar o acesso e a transmissão de mídia. A configuração de volumes assegura a persistência dos dados em reinicializações. Em resumo, essa estrutura oferece uma solução completa para hospedagem de nuvem e streaming de mídia.

CONFIGURAÇÃO DOS SERVIÇOS

Os serviços são configurados para interconectar-se de maneira eficiente. Por exemplo, o serviço `phpmyadmin` depende do serviço `db`, garantindo que o PhpMyAdmin tenha acesso ao banco de dados MySQL. Além disso, o Rocket.Chat depende do serviço `mongodb`, assegurando a conectividade com o banco de dados NoSQL.

A exposição de portas é estrategicamente escolhida para facilitar o acesso externo aos serviços, permitindo a interação com a aplicação de diferentes pontos as imagens foram tiradas do https://hub.docker.com/[1]

UTILIZAÇÃO EM UM CENÁRIO DE NEGÓCIOS

No cenário dinâmico dos negócios contemporâneos, empresas buscam constantemente soluções eficientes para gerenciar arquivos e disponibilizar serviços internos de streaming para treinamentos e conteúdos multimídia. Nesse contexto, a utilização do Docker Compose se destaca como uma escolha ideal para estabelecer uma infraestrutura flexível e escalável.

Implementação da Infraestrutura: Para atender às necessidades específicas da empresa, foram adotadas as seguintes ferramentas e serviços:

Nextcloud para Colaboração: O Nextcloud é implementado para facilitar o compartilhamento seguro de arquivos e promover a colaboração eficiente entre os membros da equipe. Isso possibilita o acesso, edição e compartilhamento de documentos de maneira centralizada.

Plex para Treinamentos Multimídia: O Plex é empregado como plataforma interna de streaming de mídia, permitindo à empresa criar, organizar e distribuir treinamentos multimídia, webinars e outros conteúdos de forma eficiente para os colaboradores.

MySQL para Bancos de Dados Isolados: A estratégia inclui a implementação de dois serviços MySQL independentes para manter bancos de dados separados para o Nextcloud e o Plex. Essa abordagem garante o isolamento e a otimização dos dados, atendendo às necessidades distintas de cada serviço.

NGINX como Ponto de Entrada: O NGINX atua como servidor web reverso, simplificando o acesso aos serviços Nextcloud e Plex. Além de proporcionar uma camada adicional de segurança, o NGINX facilita a comunicação entre usuários e serviços.

Benefícios para o Cenário de Negócios: Eficiência Colaborativa: O Nextcloud oferece uma plataforma colaborativa robusta, melhorando o compartilhamento e a colaboração em documentos, resultando em maior eficiência da equipe.

Treinamento Multimídia: O Plex permite à empresa criar e distribuir treinamentos multimídia de maneira eficiente, promovendo um ambiente de aprendizado interno dinâmico.

Isolamento de Dados: A utilização de bancos de dados MySQL separados proporciona uma gestão eficaz dos dados, garantindo que cada serviço tenha seus próprios recursos otimizados.

Acesso Centralizado: O NGINX centraliza o acesso aos serviços, simplificando a administração e proporcionando uma experiência do usuário aprimorada.

Em resumo, essa configuração proporciona uma base sólida para um ambiente de negócios moderno, onde a colaboração, o armazenamento seguro e a comunicação eficaz são essenciais.

IV. Conclusão

A implementação do Docker Compose oferece uma solução holística, atendendo às demandas de colaboração e treinamento multimídia da empresa. Esta abordagem não apenas promove a eficiência operacional, mas também preserva a integridade e a segurança dos dados, destacando-se como uma estratégia eficaz para a modernização e otimização dos processos internos.

Link para o vídeo a apresentação: www.videodeapresentacao.com.

References

[1] Docker hub. repositório público de imagens de containers. https://hub.docker.com/, 2021. Acessado em 17 de dezembro de 2023.