

Docker e Docker Compose para Iniciantes

Gustavo G. Pereira

Introdução

- O que é Docker?
 - Docker é uma plataforma que permite criar, implantar e executar aplicações em containers.
 - Containers são unidades leves e portáteis que contêm tudo o que é necessário para executar um software: código, runtime, bibliotecas e configurações do sistema.
- O que é Docker Compose?
 - Docker Compose é uma ferramenta que permite definir e gerenciar aplicações multi-container.
 - Usando um arquivo YAML, você pode definir serviços, redes e volumes necessários para a sua aplicação.

Vantagens do Docker

- Portabilidade
 - Funciona em qualquer lugar: no seu laptop, em servidores locais ou na nuvem.
- Consistência
 - Ambientes de desenvolvimento e produção idênticos.
- Eficiência
 - Uso eficiente de recursos do sistema.
- Isolamento
 - Isola aplicações e suas dependências.

Conceitos Básicos do Docker

- Imagem
 - Um snapshot imutável de um container. Contém o sistema de arquivos e a configuração necessária para rodar uma aplicação.
- Container
 - Uma instância de uma imagem em execução. É isolado e pode ser facilmente iniciado, parado, movido ou deletado.
- Dockerfile
 - Um script que define como criar uma imagem Docker.
- Registry
 - Um repositório para armazenar imagens Docker. O mais popular é o Docker Hub.

Como o Docker Funciona

1. Dockerfile
 - Cria uma imagem base.
2. Imagem
 - Armazena a aplicação e suas dependências.
3. Container
 - Instância da imagem em execução.
4. Registry
 - Armazena e distribui imagens.

Docker na Prática

- Comandos Básicos
 - `docker build`: Cria uma imagem a partir de um Dockerfile.
 - `docker run`: Executa um container a partir de uma imagem.
 - `docker ps`: Lista containers em execução.
 - `docker stop`: Para um container em execução.
 - `docker rm`: Remove um container.

Introdução ao Docker Compose

- Arquivo `docker-compose.yml`
 - Define os serviços, redes e volumes para uma aplicação.
- Comandos Básicos
 - `docker-compose up`: Inicia e executa todos os serviços definidos no `docker-compose.yml`.
 - `docker-compose down`: Para e remove todos os containers, redes e volumes criados pelo `docker-compose up`.
 - `docker-compose ps`: Lista os containers em execução.

Exemplo de Arquivo docker-compose.yml

```
version: '3'
services:
  web:
    image: nginx
    ports:
      - "80:80"
  db:
    image: postgres
    environment:
      POSTGRES_PASSWORD: example
```


Usando Docker Compose

1. Criar `docker-compose.yml`
 - Defina seus serviços e configurações.
2. Executar `docker-compose up`
 - Inicia todos os serviços definidos.
3. Verificar status com `docker-compose ps`
 - Verifica quais serviços estão em execução.
4. Parar e remover com `docker-compose down`
 - Para e limpa o ambiente.

Benefícios do Docker Compose

- Simplificação
 - Gerencia múltiplos containers facilmente.
- Automação
 - Automação de setups de desenvolvimento e pipelines de CI/CD.
- Ambientes Reprodutíveis
 - Consistência entre ambientes de desenvolvimento, teste e produção.

Conclusão

- Recapitulando:
 - Docker facilita a criação, distribuição e execução de aplicações.
 - Docker Compose simplifica o gerenciamento de aplicações multi-container.
- Próximos Passos:
 - Explore a documentação oficial do Docker e Docker Compose.
 - Experimente criar seus próprios Dockerfiles e `docker-compose.yml`.
 - Pratique com projetos de exemplo.

Perguntas e Respostas

- Perguntas?
 - Abrir para perguntas do público.

