

1) O que é um Dockerfile?

Um Dockerfile é um arquivo de texto usado para definir a configuração de um contêiner Docker. Ele contém uma série de instruções que são interpretadas pelo Docker para criar uma imagem do contêiner. O Dockerfile permite que você defina todos os componentes necessários para criar um ambiente de execução específico dentro do contêiner, como o sistema operacional base, pacotes, bibliotecas, variáveis de ambiente e comandos de inicialização. Com o Dockerfile, é possível criar imagens personalizadas e repetíveis para aplicativos ou ambientes de desenvolvimento, garantindo a consistência e a portabilidade das aplicações.

2) Qual comando é usado para iniciar um Dockerfile?

`docker build -t nome-da-imagem caminho-do-dockerfile`

3) Qual comando é usado para definir a imagem base no Dockerfile?

`FROM nome-da-imagem[:tag]`

4) Como você adiciona arquivos locais ao sistema de arquivos do contêiner no Dockerfile?

`COPY app.js /usr/src/app/`

5) Como você especifica o diretório de trabalho no Dockerfile?

`WORKDIR <caminho-do-diretorio>`

6) Qual comando é usado para executar comandos durante a criação da imagem Docker no Dockerfile?

`RUN apt-get update && apt-get install -y nginx`

7) Como você expõe portas no Dockerfile?

`EXPOSE <porta> [<porta>/<protocolo>...]`

8) Qual comando é usado para definir variáveis de ambiente no Dockerfile?

`ENV <nome>=<valor>`

9) Como você comenta linhas no Dockerfile?

`# Este é um comentário sobre o comando abaixo`

`RUN apt-get update && apt-get install -y nginx`

`# Outro comentário`

`EXPOSE 80/tcp`

10) Qual comando é usado para executar a aplicação principal quando um contêiner é iniciado no Dockerfile?

Resposta: O Docker usa uma interface CLI para gerenciar os seus vários objetos através de comandos. O que faz os comandos abaixo:

1)docker run:

O comando "docker run" é usado para criar e executar um novo contêiner a partir de uma imagem. Ele cria uma instância em execução do contêiner com base nas configurações especificadas, como portas expostas, variáveis de ambiente e volumes montados.

2)docker build:

O comando "docker build" é usado para criar uma nova imagem Docker a partir de um Dockerfile. Ele automatiza o processo de construção da imagem, executando as etapas definidas no Dockerfile e gerando uma nova imagem resultante.

3)docker pull:

O comando "docker pull" é usado para baixar uma imagem Docker de um registro ou repositório, como o Docker Hub. Ele faz o download da imagem especificada para a máquina local, tornando-a disponível para uso posteriormente.

4)docker push:

O comando "docker push" é usado para enviar uma imagem Docker local para um registro ou repositório, como o Docker Hub. Ele envia a imagem especificada para o registro, tornando-a disponível para uso por outros desenvolvedores ou em outras máquinas.

5)docker ps:

O comando "docker ps" é usado para listar os contêineres em execução no momento. Ele exibe informações como o ID do contêiner, a imagem usada, o status de execução, as portas expostas e outros detalhes relevantes.

6)docker images:

O comando "docker images" é usado para listar as imagens Docker disponíveis localmente na máquina. Ele exibe informações como o nome da imagem, o tamanho, a versão e outros detalhes relacionados.

7)docker stop:

O comando "docker stop" é usado para interromper a execução de um ou mais contêineres em execução. Ele envia um sinal de interrupção (SIGTERM) para o contêiner, permitindo que ele faça uma finalização adequada antes de ser encerrado.

8)

docker start:

O comando "docker start" é usado para iniciar um ou mais contêineres previamente criados que estão parados. Ele inicia a execução do contêiner, mantendo as configurações e o estado anteriores.

9) docker rm:

O comando "docker rm" é usado para remover um ou mais contêineres que estão parados. Ele remove completamente o contêiner, eliminando todos os dados associados a ele, como sistema de arquivos, rede e configurações.

10) docker rmi:

O comando "docker rmi" é usado para remover uma ou mais imagens Docker locais. Ele remove a imagem especificada da máquina local, liberando espaço em disco.

11) ndocker exec:

O comando "docker exec" é usado para executar um comando dentro de um contêiner em execução. Ele permite a execução de comandos específicos dentro do ambiente do contêiner, como entrar em um shell interativo.

12) docker logs:

O comando "docker logs" é usado para visualizar os logs de um contêiner em execução. Ele exibe a saída do console (stdout/stderr) do contêiner, permitindo que você acompanhe e depure o funcionamento da aplicação.

13) docker network:

O comando "docker network" é usado para gerenciar redes Docker. Ele permite a criação, listagem e remoção de redes, bem como a conexão de contêineres a redes específicas.

14) docker volume:

O comando "docker volume" é usado para gerenciar volumes Docker. Ele permite a criação, listagem e remoção de volumes, que são usados para armazenar e compartilhar dados persistentes entre contêineres.