

Disciplina 2023/2

ENGENHARIA DE SOFTWARE

Professor: Thiago Bussola da Silva

thiago.bsilva@unicesumar.edu.br

Gerenciamento de Configuração

- Docker é uma plataforma de software que simplifica o processo de construção, execução, gerenciamento e distribuição de aplicativos.
- Ele faz isso virtualizando o sistema operacional do computador no qual está instalado e sendo executado
- Ele permite que os usuários criem ambientes independentes e isolados para iniciar e implantar seus aplicativos. Esses ambientes são chamados de contêineres.
- Isso permitirá que o desenvolvedor execute um contêiner em qualquer máquina.

- Com o Docker, não há mais problemas de dependência ou compilação. Tudo que você precisa fazer é iniciar seu contêiner e seu aplicativo será iniciado imediatamente.
- O Docker é um programa open source desenvolvido pela Docker Inc. com a linguagem de programação GO.
- Sua primeira edição foi lançada em 13 de março de 2013 e, desde seu lançamento, se tornou um software importantíssimo no mundo do desenvolvimento de tecnologia.
- Podemos dizer que as palavras chaves para o Docker são: construir, entregar e rodar em qualquer ambiente (build, ship and run anywhere).

- Docker não é um sistema operacional completo numa máquina virtual.
- O Docker só compartilhará os recursos da máquina host para executar seus ambientes.

- O Docker é rápido. Ao contrário de uma máquina virtual, seu aplicativo inicializa em alguns segundos e para com a mesma rapidez.
- O Docker é multiplataforma. Você pode iniciar seu contêiner em qualquer sistema.
- Os contêineres podem ser construídos e excluídos mais rápido do que em uma máquina virtual.

- Não há mais dificuldades em configurar seu ambiente de trabalho. Depois que seu Docker estiver configurado, você nunca mais terá que reinstalar suas dependências manualmente mesmo se mudar seu computador.
- Você mantém seu espaço de trabalho limpo, pois cada um de seus ambientes será isolado e você pode excluí-los a qualquer momento, sem impactar o resto.
- É mais fácil implantar seu projeto no servidor para colocá-lo online

- Há uma tonelada de solicitações de recursos para upgrade que ainda estão em andamento (como capacidade de autorregistro e autoinspeção de contêineres, cópia de arquivos do host para o contêiner e muito mais).
- Há momentos em que um container fica inativo, então depois disso, ele precisa de uma estratégia de backup e recuperação, embora existam várias soluções, mas que não são automatizadas ou nem muito escaláveis ainda.

- Em comparação com as máquinas virtuais, os containers Docker oferecem menos sobrecarga, mas não sobrecarga zero.
- O principal problema é que se um aplicativo projetado para ser executado em um contêiner do Docker no Windows, ele não pode ser executado no Linux ou vice-versa. No entanto, as máquinas virtuais não estão sujeitas a essa limitação.
- Muitos contêineres podem prejudicar o desempenho da máquina na hora do desenvolvimento e aumentar o custo de infra da sua aplicação

Docker para Mac, Linux e Windows – versões que permitem instalar e executar containers nos sistemas operacionais de forma isolada.

Docker Daemon – Software que roda na máquina onde o Docker está instalado. Usuário não interage diretamente com o daemon.

Docker Client – CLI ou REST API que aceita comandos do usuário e repassa estes comandos ao Docker daemon.

Docker Image – É um template. Uma imagem contém todos os dados e metadados necessários para executar containers a partir de uma imagem.

Docker Container – Detém tudo que é necessário para uma aplicação ser executada. Cada container é criado a partir de uma imagem. Cada container é uma aplicação isolada independente.

Docker Engine – Usado para criar imagens e containers.

Docker Registry – Uma coleção de imagens hospedadas e rotuladas que juntas permitem a criação do sistema de arquivos de um container. Um registro pode ser público ou privado.

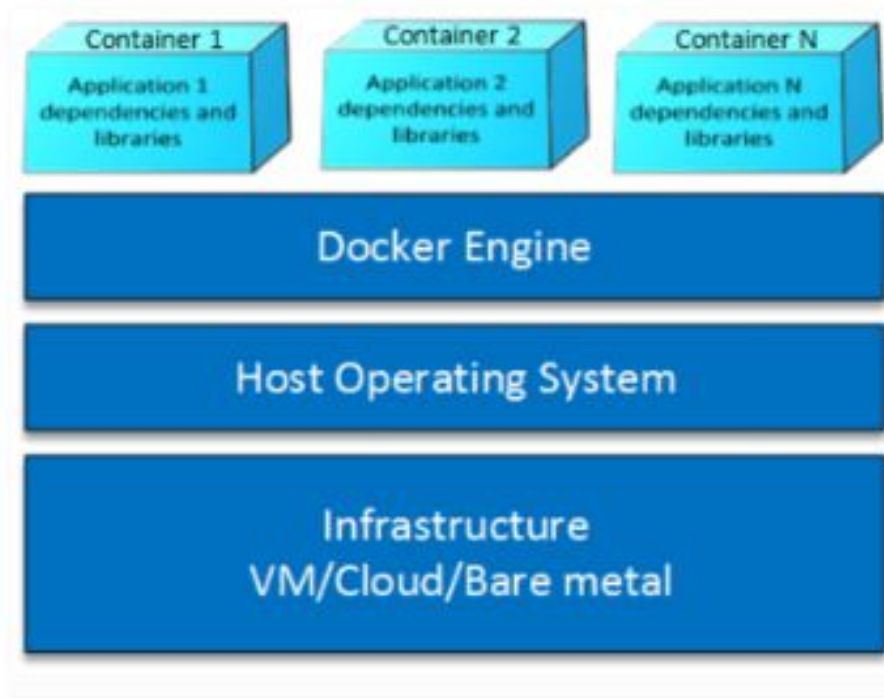
Docker Hub – Este é um registro usado para hospedar e baixar diversas imagens. Pode ser visto como uma plataforma SAAS de compartilhamento e gerenciamento de imagens.

Dockerfile – Um arquivo texto contendo uma syntax simples para criação de novas imagens.

Docker Compose – Usado para definir aplicações usando diversos containers.

Docker Swarm – É uma ferramenta que permite o agrupamento (clustering) de Containers Docker.

Arquitetura do Docker



Nesta etapa, você escreve um Dockerfile que cria uma imagem do Docker. A imagem contém todas as dependências que o aplicativo Node.js precisa, incluindo o próprio Node.

Docker Image

FROM node:18-alpine

WORKDIR /home/node/app

COPY package*.json ./

RUN npm install

COPY --chown=node:node . .

RUN npm run build

EXPOSE 3001

CMD ["npm", "run", "start:prod"]

1- A imagem alpine é derivada do projeto Alpine Linux e ajudará a manter o tamanho da imagem baixo:

FROM node:18-alpine

Esta imagem inclui Node.js e npm. Cada Dockerfile deve começar com uma instrução FROM.

2 - Em seguida, defina o diretório de trabalho do aplicativo como `/home/node/app`:
WORKDIR /home/node/app

Se um WORKDIR não estiver definido, o Docker criará um por padrão, portanto é uma boa ideia defini-lo explicitamente.

3 - Em seguida, copie os arquivos package.json e package-lock.json:

COPY package*.json ./

Dica: Adicionar esta instrução COPY antes de executar npm install ou copiar o código do aplicativo permite que você aproveite as vantagens do mecanismo de cache do Docker. Em cada estágio da construção, o Docker verificará se há uma camada armazenada em cache para aquela instrução específica. Se você alterar o package.json, esta camada será reconstruída, mas se você não fizer isso, esta instrução permitirá que o Docker use a camada de imagem existente e pule a reinstalação dos módulos do nó.

4 - Após copiar as dependências do projeto e trocar o usuário, execute npm install:

RUN npm install

5 - Em seguida, copie o código do seu aplicativo com as permissões apropriadas para o diretório do aplicativo no contêiner: **COPY --chown=node:node . .**

Isso garantirá que os arquivos do aplicativo sejam de propriedade do usuário do nó não raiz.

6 - Como estamos rodando uma aplicação em typescript é preciso fazer o processo de transpilação primeiro:

RUN npm run build

7 e 8 - Por fim, exponha a porta 8080 no contêiner e inicie o aplicativo:

EXPOSE 3001

CMD ["npm", "run", "start:prod"]

EXPOSE não publica a porta, mas funciona como uma forma de documentar quais portas do contêiner serão publicadas em tempo de execução. O **CMD** executa o comando para iniciar o aplicativo – neste caso, **start:prod** esse script de inicialização está presente no package.json

build	Build an image from a Dockerfile
cp	Copy files/folders between a container and the local filesystem
create	Create a new container
exec	Run a command in a running container
images	List images
inspect	Return low-level information on a container or image
kill	Kill a running container
logs	Fetch the logs of a container
port	List port mappings or a specific mapping for the CONTAINER
ps	List containers
pull	Pull an image or a repository from a registry
push	Push an image or a repository to a registry

rename	Rename a container
restart	Restart a container
rm	Remove one or more containers
rmi	Remove one or more images
run	Run a command in a new container
save	Save one or more images to a tar archive
start	Start one or more stopped containers
stop	Stop a running container
top	Display the running processes of a container
volume	Manage Docker volumes

Para aplicativos que dependem de vários serviços, orquestrar todos os contêineres para inicialização, comunicação e encerramento juntos pode se tornar rapidamente complicado. Docker Compose é uma ferramenta que permite executar ambientes de aplicativos com vários contêineres com base em definições definidas em um arquivo YAML. Ele usa definições de serviço para criar ambientes totalmente personalizáveis com vários contêineres que podem compartilhar redes e volumes de dados.

- Clone o projeto base no Github
- Crie o Docker File
- Pesquise como criar e utilizar o docker-compose
- Crie um arquivo docker-compose que baixe uma imagem do mongodb e faça a conexão entre o container do banco com o seu container da API
- Crie um texto explicando e relacionando Gerenciamento de Configuração e uso do docker
- Crie um texto explicando e relacionando Gerenciamento de Configuração e certificações como o CMMI e a MPS.BR
- Criar um docker file em uma pasta separada que baixe a imagem de alguma linguagem de programação. E execute comandos dentro do container para garantir a instalação