

Linguagem de Programação Visual

RICARDO HENDGES

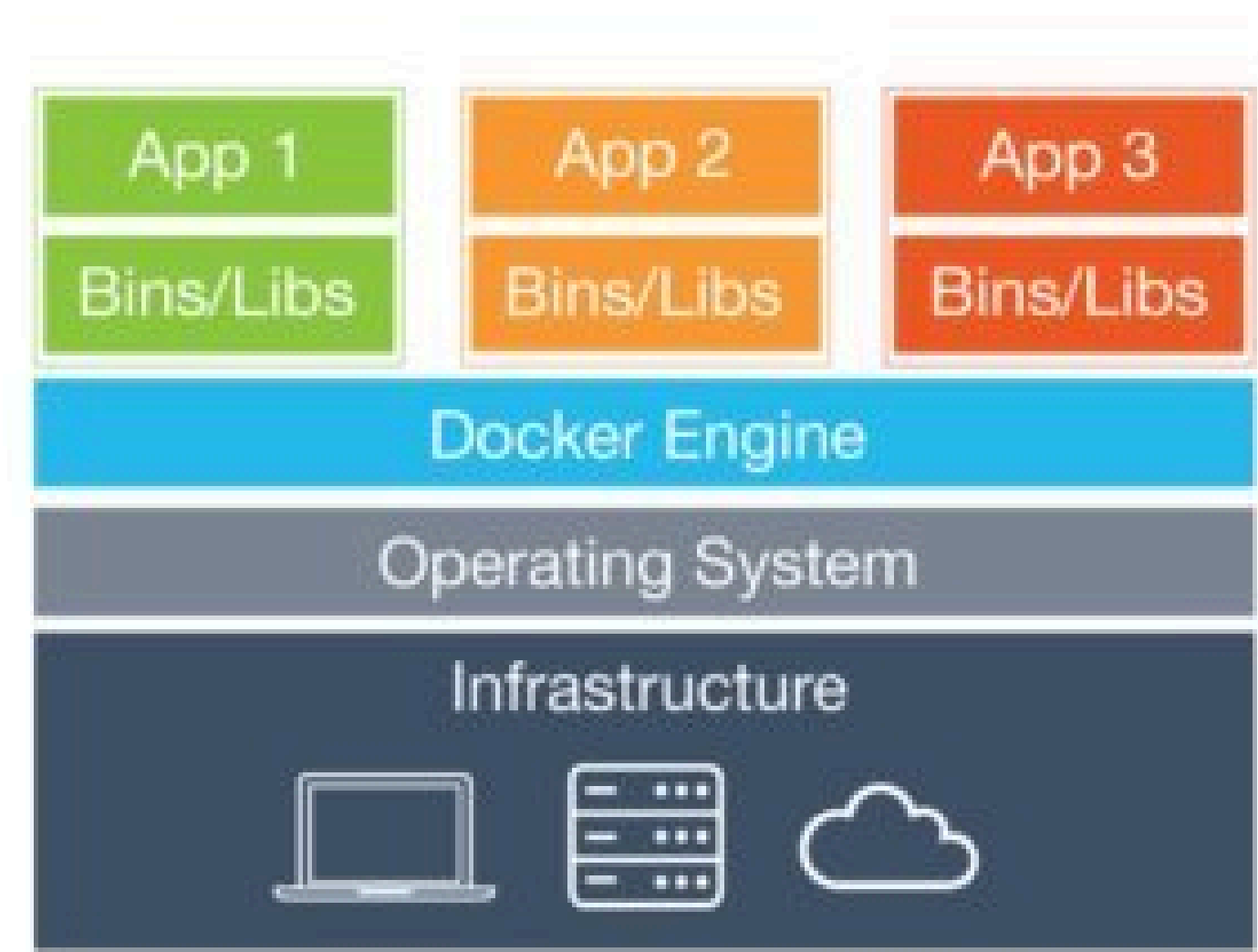
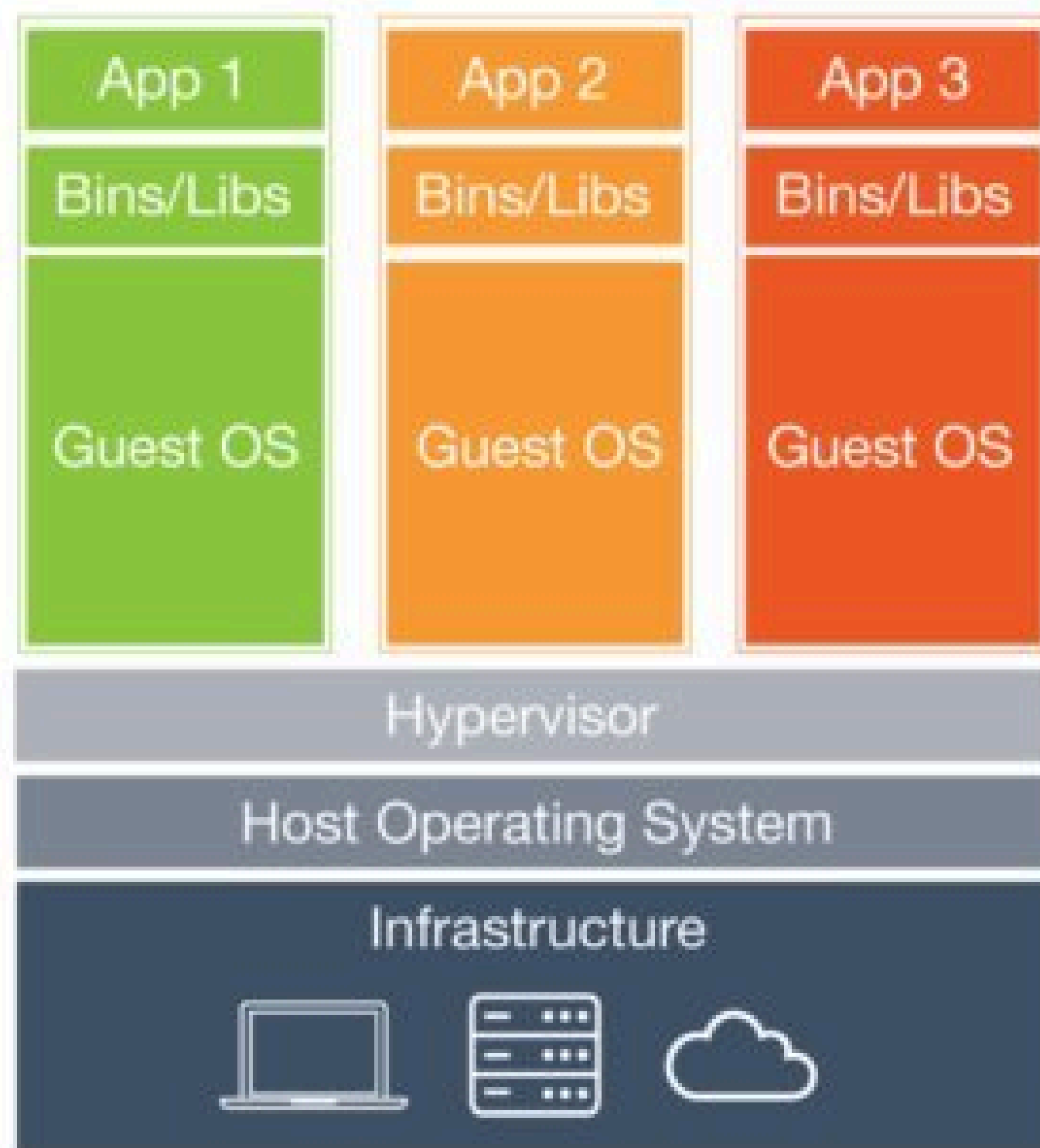




O Docker é uma plataforma open source que facilita a criação e administração de ambientes isolados. Ele possibilita o empacotamento de uma aplicação ou ambiente dentro de um container, se tornando portátil para qualquer outro host que contenha o Docker instalado. Então, você consegue criar, implantar, copiar e migrar de um ambiente para outro com maior flexibilidade. A ideia do Docker é subir apenas uma máquina, ao invés de várias. E, nessa única máquina, você pode rodar várias aplicações sem que haja conflitos entre elas.

Docker é uma máquina virtual?

O Docker é algo parecido com uma máquina virtual extremamente leve, mas não se trata de fato de uma máquina virtual. O Docker utiliza containers que possuem uma arquitetura diferente, permitindo maior portabilidade e eficiência. O container exclui a virtualização e muda o processo para o Docker. Então, não podemos dizer que o Docker é uma máquina virtual.





O que são esses containers?

Um container é um ambiente isolado. Um container contém um conjunto de processos que são executados a partir de uma imagem, imagem esta que fornece todos os arquivos necessários. Os containers compartilham o mesmo kernel e isolam os processos da aplicação do restante do sistema.

Como o container possui uma imagem que contém todas as dependências de um aplicativo, ele é portátil e consistente em todas as etapas de desenvolvimento. Essa imagem é um modelo de somente leitura que é utilizada para subir um container. O Docker nos permite construir nossas próprias imagens e utilizá-las como base para os containers.



Por que utilizar o Docker?

- Economia de recursos
- Melhor disponibilidade do sistema (pelo compartilhamento do SO e de outros componentes),
- Possibilidades de compartilhamento
- Simplicidade de criação e alteração da infraestrutura
- Manutenção simplificada (reduzindo o esforço e o risco de problemas com as dependências do aplicativo)
- Entre muitas outras.

Entendendo a estrutura

FROM

Instrução obrigatória que indica qual imagem vai ser utilizada como ponto de partida. Podemos criar uma imagem a partir de uma imagem do Node.js ou um sistema Linux usando uma imagem do Ubuntu e assim por diante.

from node:14-alpine

The bottom right corner of the slide features several thin, light blue wavy lines that sweep upwards and to the right, creating a modern, abstract graphic element.

Entendendo a estrutura

ENV

Serve para você setar variáveis de ambiente, você pode tanto deixar essas variáveis setadas de forma fixa dentro do Dockerfile quanto passá-las dinamicamente na hora que você instanciar o container.

ENV TZ=America/Sao_Paulo



Entendendo a estrutura

RUN

Serve para executar comandos no processo de montagem da imagem que estamos construindo no Dockerfile, ele é executado durante o build (construção da imagem) e não durante a construção do container. Em um Dockerfile é possível ter mais de um comando RUN.

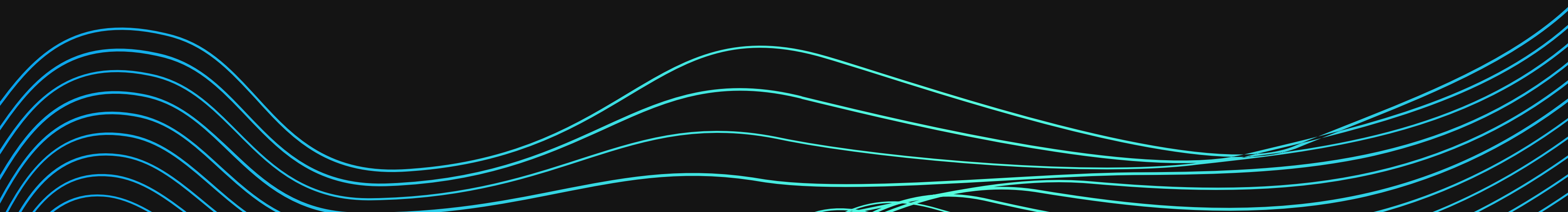
```
RUN mkdir -p /home/node/horus
```

Entendendo a estrutura

WORKDIR

Define uma pasta dentro do container onde os comandos serão executados.

WORKDIR /home/node/horus



Entendendo a estrutura

COPY

Comando para copiar arquivos e pastas de um lugar específico na máquina local para uma pasta no container.

```
COPY ./package.json ./  
COPY ./src ./src
```



RUN npm install

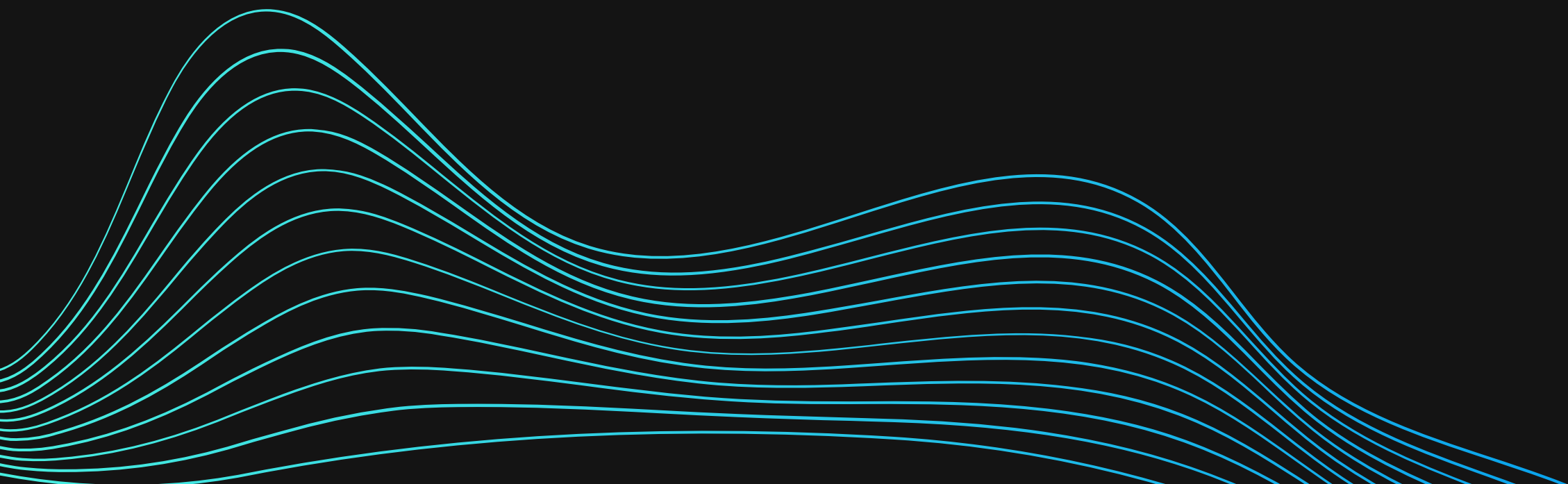
Entendendo a estrutura

EXPOSE

Instrução que informa qual porta deverá ser liberada. EXPOSE não é um comando que libera a porta, ele serve para ficar documentado no Dockerfile quais portas deverão ser liberadas ao criar o container.

Para efetivamente liberar a porta é preciso passar o parâmetro `-p` no comando `docker run`, vamos ver a seguir.

EXPOSE 3000



Entendendo a estrutura CMD

Diferente do RUN, o CMD executa apenas na criação do container e não no build da imagem. Deve ser único no Dockerfile.

```
CMD [ "node", "./src/index.js" ]
```



Build

docker build -t horus .

**docker run -d -p
3000:3000 horus**

```
FROM node:14-alpine
```

```
ENV TZ=America/Sao_Paulo
```

```
RUN mkdir -p /home/node/horus
```

```
WORKDIR /home/node/horus
```

```
COPY ./package.json ./
```

```
COPY ./src ./src
```

```
RUN npm install
```

```
EXPOSE 3000
```

```
CMD [ "node", "./src/index.js" ]
```