



PUC Minas

Plataforma Node.js

Porque Node.js

Porque Node se chama Node? 😊

Inicialmente criado para substituir servidores Web bloqueantes como o Apache, o Node foi além e acabou sendo uma plataforma para criar aplicações distribuídas de diversas naturezas, um nó em uma estrutura maior
→ daí ... node.

Originally, Dahl called his project web.js. It was merely a webserver, an alternative to Apache and other "blocking" servers. But the project soon grew beyond his initial webserver library, expanding into a framework that could be used to build, well, almost anything. So he rechristened it node.js.

Fonte: [The Node Ahead: JavaScript leaps from browser into future • The Register](#)



276



Node Package Manager (NPM)

Prof. Rommel Vieira Carneiro





PUC Minas

Plataforma Node.js

Node.js Shebang

Node.js shebang (unix-like OSs)

Ao invés de executar seu script por meio da linha de comando `node script.js`, é possível o arquivo do script executável (`chmod a+x script.js`) e permitir a execução diretamente pela linha de comando.

Shebang (`#!/`) → linha colocada no início do arquivo que informa ao SO qual é o interpretador do script que está no arquivo. Veja o exemplo:

```
#!/usr/bin/env node  
console.log ('Arquivo auto executável')
```

Fonte: [Node.js shebang](#)

Node.js shebang (unix-like OSs)

Começar o shebang com `/usr/bin/env` permite que o SO inicie um novo shell com o PATH atual para executar o comando que se segue.

Torna o script mais portátil. No exemplo, o SO vai procurar o Node.js em qualquer lugar do PATH.

```
#!/usr/bin/env node  
console.log ('Arquivo auto executável')
```

Fonte: [Node.js shebang](#)



279



Node – Tópicos Avançados

Prof. Rommel Vieira Carneiro





PUC Minas

Plataforma Node.js

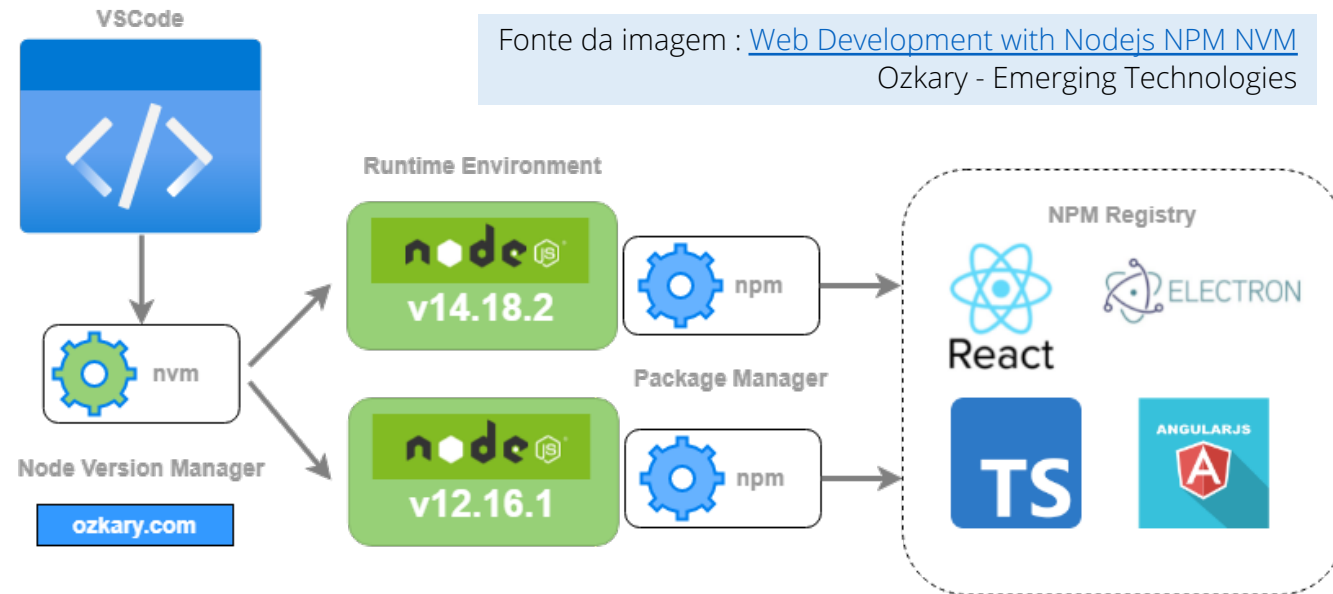
Node Version Manager (NVM)

Node.js Versions (nvm)



Em ambientes profissionais, muitas vezes se torna necessário utilizar diversas versões da plataforma Node.js para projetos diferentes.

Com o **Node Version Manager (NVM)**, conseguimos gerenciar diversas versões do Node.js no mesmo computador.



Node.js Versions (nvm)



1) Instalação a partir do [repositório do GitHub](#) (Linux e MacOS)

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash  
  
# or  
  
wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash
```

2) Configuração do PATH no shell do OS (Linux e MacOS)

→ Acrescente o script de configuração para o NVM no ~/.zshrc ou ~/.bashrc

```
export NVM_DIR="$([ -z "${XDG_CONFIG_HOME-}" ] && printf %s "${HOME}/.nvm" || printf %s "${XDG_CONFIG_HOME}/nvm")"  
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"
```


Node.js Versions (nvm)



3) Resumo dos principais comandos do NVM

Exemplos de Comandos	Descrição
<code>nvm ls</code>	// Lista as versões instaladas
<code>nvm install latest</code>	// Instala a última versão
<code>nvm install --lts</code>	// Instala a versão LTS
<code>nvm install 14.17.6</code>	// Instala uma versão específica
<code>nvm uninstall 14.17.6</code>	// Remove uma versão específica
<code>nvm alias default 14.17.6</code>	// Cria um alias para uma versão instalada
<code>nvm use 14.17.6</code> <code>nvm use default</code>	// Seleciona uma versão para ser utilizada
<code>nvm run default script.js</code>	// Executa o script.js em versão específica

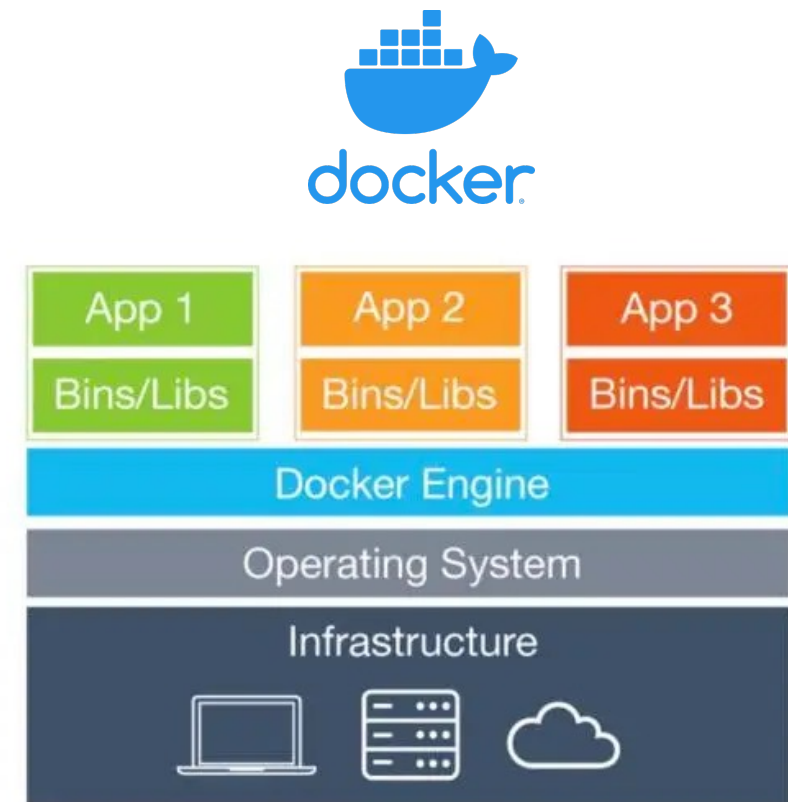


PUC Minas

Plataforma Node.js

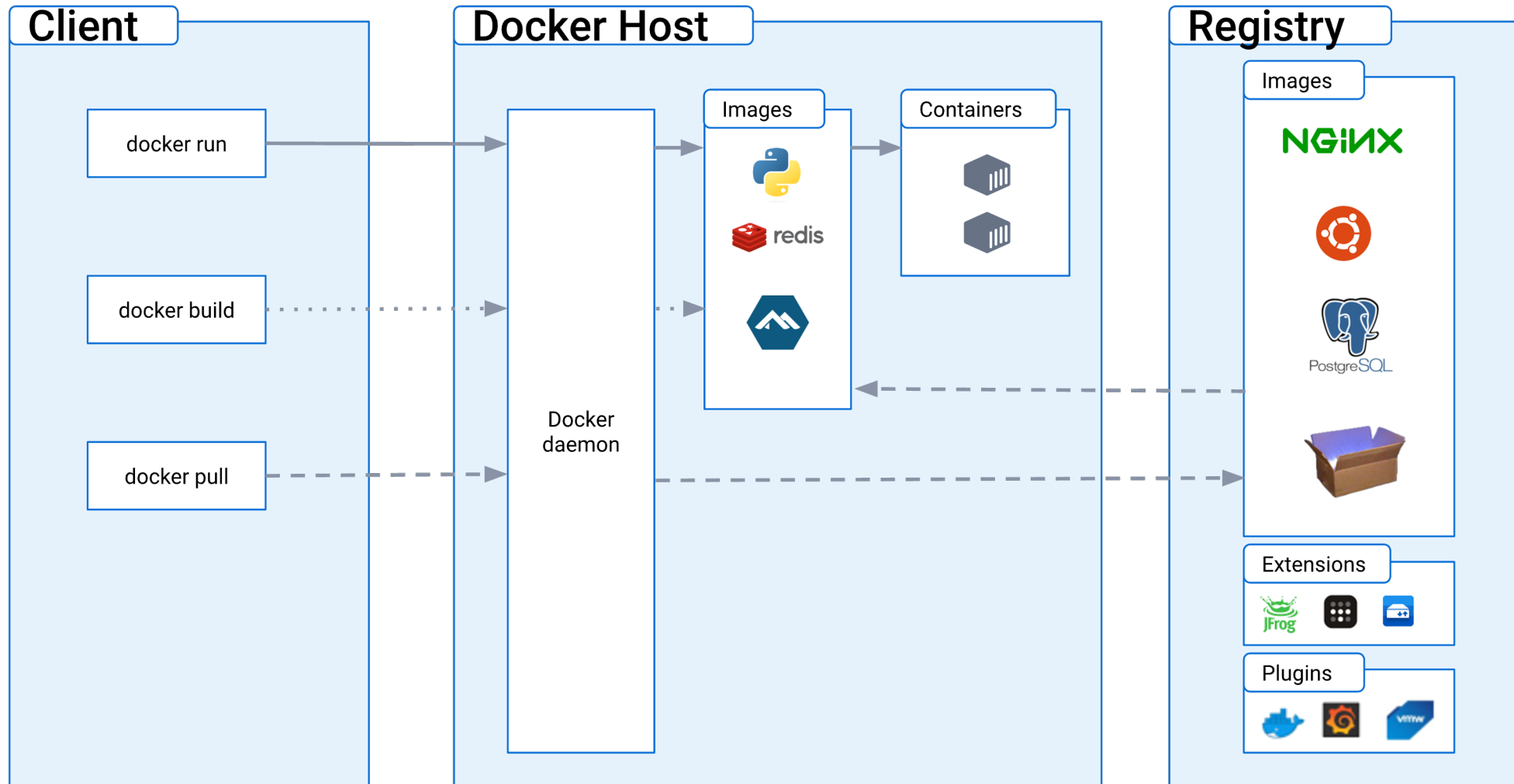
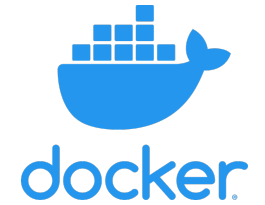
Containers Docker

Máquinas virtuais vs Containers



Fonte: [What is Docker? The spark for the container revolution](#) (InfoWorld)

Containers e Docker



Fonte: [Docker overview](#) | [Docker Documentation](#)

Dockerfile para Node



docker

```
# Informa a imagem a ser utilizada como base
FROM node:lts-alpine

# Cria o diretório para a aplicação
WORKDIR /app

# Copia o package.json para o diretório da aplicação e instala as dependências
COPY package*.json ./
RUN npm install

# Copia o código fonte para o diretório da aplicação
COPY . .

# Define porta de execução da aplicação
EXPOSE 80

# Executa a aplicação
CMD [ "npm", "start" ]
```



289



Node – Tópicos Avançados

Prof. Rommel Vieira Carneiro



Docker – Principais Comandos



docker

Exemplos de Comandos	Descrição
<code>docker pull node:lts-alpine</code>	Faz download de uma imagem do Docker Hub. No exemplo a imagem node com tag lts-alpine
<code>docker build . -t node-app</code>	Cria uma imagem a partir de um Dockerfile . Especifica o caminho do Dockerfile (pasta corrente) -t Nomeia a imagem (tag) com node-app
<code>docker images</code>	Lista todas as imagens disponíveis localmente.
<code>docker run -d -p 80:3000 -name app node-app</code>	Cria um container a partir de uma imagem. -d Executa em segundo plano -p 80:3000 Associa portas 80 (host):3000 (container) -name app Nomeia o container como app node-app Informa a imagem a ser usada (node-app)
<code>docker logs app</code>	Exibe os logs de um container.
<code>docker exec -it app /bin/bash</code>	Executa um terminal interativo no container -i Executa em modo interativo -t Aloca um pseudo-terminal (tty)



290



Node – Tópicos Avançados

Prof. Rommel Vieira Carneiro



Docker – Principais Comandos



docker

Exemplos de Comandos	Descrição
<code>docker run -it alpine</code>	Cria um container de nome aleatório a partir da imagem do Alpine Linux (alpine) e abre um terminal (shell) no container para interação -i Executa em modo interativo -t Aloca um pseudo-terminal (tty)
<code>docker ps -a</code>	Lista os containers -a inclui todos os containers, inclusive os parados
<code>docker start <container></code>	Inicia um container que está parado
<code>docker pause <container></code>	Pausa um container em execução
<code>docker stop <container></code>	Finaliza um container em execução
<code>docker rm <container></code>	Remove um container
<code>docker rmi <image></code>	Remove uma imagem



291



Obrigado!