

Introduction to Web Frameworks with Docker

F. Farias, L. Melo, M. d'Amorim
{fbf3,lhsm,damorim}@cin.ufpe.br

Objetivo

- Entendimento do funcionamento de novas tecnologias de desenvolvimento;
- Preparar para exercícios práticos na próxima aula.

Agenda

Parte 1:

1. Introdução à Web Frameworks

- a. O que é um Web Server?
- b. O que é um Web Framework?
- c. Motivação
- d. Exemplos práticos



Agenda

Parte 2:

1. Introdução ao Docker

- a. O que é o Docker?
- b. Motivação
- c. Utilização & Comunidade
- d. Comandos básicos
- e. Exemplos práticos



O que é um Web Server?

- É um programa que entende o protocolo HTTP e gerar respostas apropriadas para cada requisição.
 - **Cliente** faz uma solicitação ao **servidor**
 - **Servidor** recebe e interpreta a solicitação
 - **Servidor** envia um cabeçalho com a resposta
 - **Servidor** fecha a conexão
 - **Cliente** interpreta a resposta

Features de um Web Service

- Rotas/caminhos da aplicação;
- Páginas dinâmicas;
- Bancos de dados;
- etc...

Introdução à Web Frameworks







- Fornecem idiomas de programação (abstrações) para facilitar o desenvolvimento.

Exemplos práticos

- Flask - Python
- Express - Node.JS

Introdução ao Docker

- É uma tecnologia de código aberto que permite criar, executar, testar e implantar aplicações distribuídas dentro de containers de software.
- É uma ferramenta para criar e manter containers, ou seja, responsável por armazenar vários serviços de forma isolada do SO host

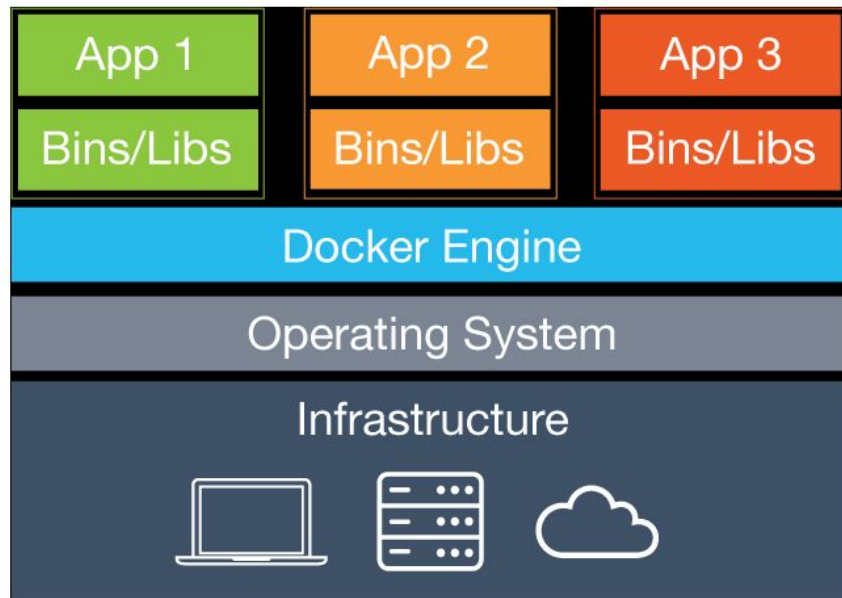
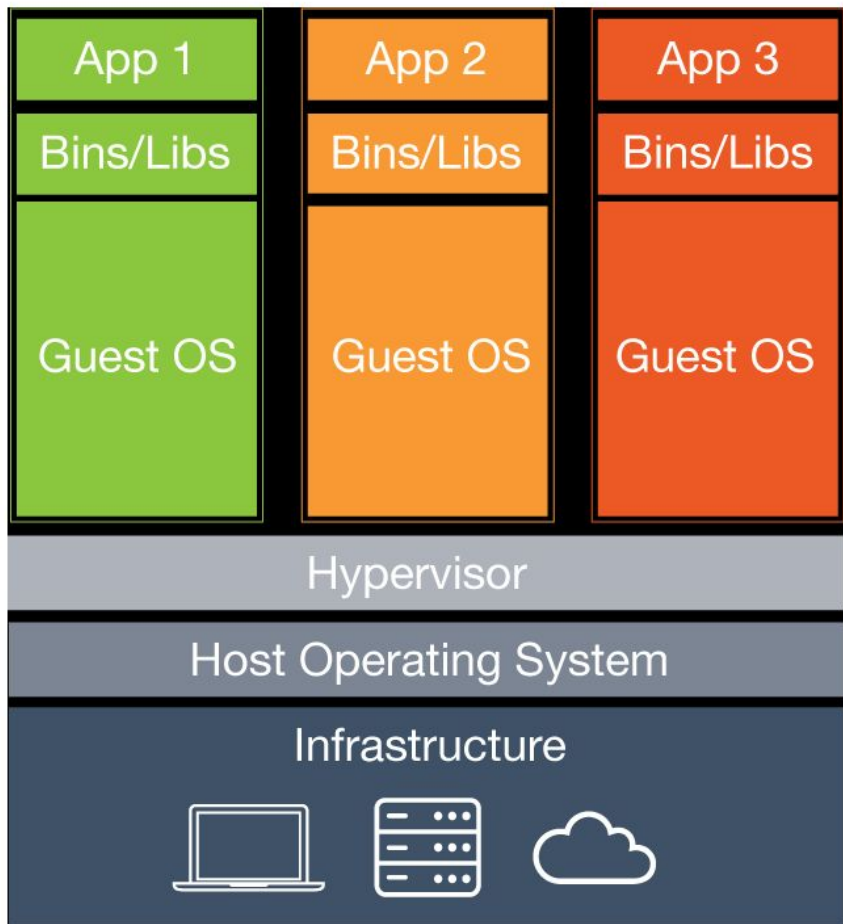
	Static website	?	?	?	?	?	?	?
	Web frontend	?	?	?	?	?	?	?
	Background workers	?	?	?	?	?	?	?
	User DB	?	?	?	?	?	?	?
	Analytics DB	?	?	?	?	?	?	?
	Queue	?	?	?	?	?	?	?
		Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers





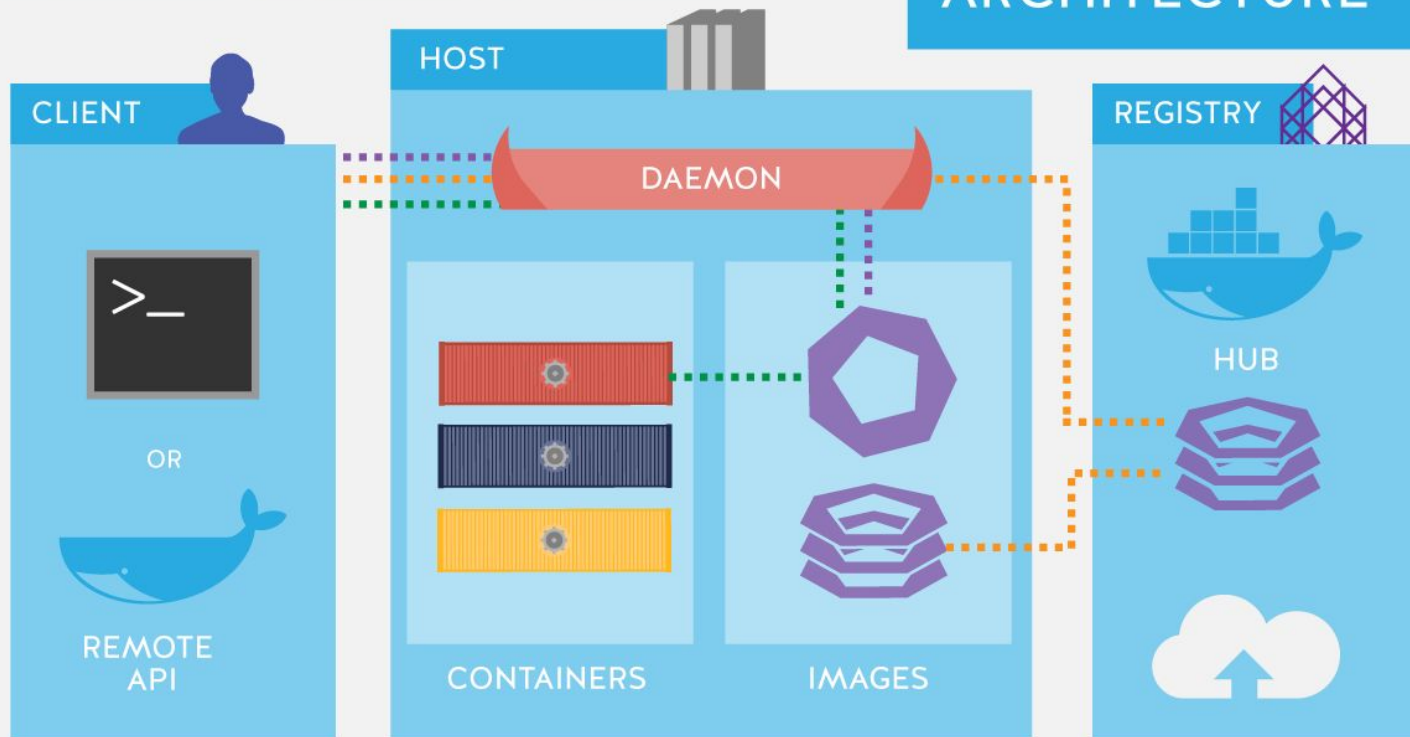
Motivação

- É uma ferramenta para criar e manter containers;
- Tornar containers facilmente portáteis;
- Evitar conflitos de dependências;
- Automatizar a implantação dentro de containers;
- etc...





DOCKER ARCHITECTURE



Explorando Docker

Instalação:

```
$> sudo apt-get update
```

```
$> sudo apt-get install docker-engine
```

Hello Docker:

```
$> sudo docker run ubuntu /bin/echo Hello, Cln!
```

Explorando Docker

Docker Hub:

<https://hub.docker.com/explore>

Contém milhares de imagens para baixar e usar na construção de containers.

Explore Official Repositories			
 nginx official	5.8K STARS	10M+ PULLS	> DETAILS
 redis official	3.6K STARS	10M+ PULLS	> DETAILS
 busybox official	990 STARS	10M+ PULLS	> DETAILS
 ubuntu official	5.9K STARS	10M+ PULLS	> DETAILS

Imagem x Container

Imagem:

- Imagem é uma coleção de arquivos;
- Imagens são feitas de várias camadas empilhadas;
- Cada camada pode adicionar e/ou remover arquivos;
- Imagens podem compartilhar camadas entre si;
- Imagens são read-only.

Imagem x Container

Container:

- Container é um conjunto de processos encapsulados sendo executados em modo read-write;
- **docker run** executa um container a partir de uma imagem.

Dockerfile & Comandos básicos

- Dockerfile é uma receita para uma imagem Docker;
- **docker build** compila um **Dockerfile** em uma **imagem**.

Dockerfile & Comandos básicos

- **FROM:** Indica a imagem base;
- **RUN:** Indica um comando a ser executado no Docker;
 - Nenhum **RUN** pode ser um comando interativo;
- **CMD:** Define o comando **padrão** ao ser executado quando nenhum é informado;
- **COPY:** Copia um ou vários arquivos para dentro da imagem;
- **WORKDIR:** Cria uma pasta e define como pasta padrão para a Imagem.