

DevOps Módulo 6 - IT Talent

Escrito por: [Gabriel Oliveira dos Santos](#)

Github: <https://github.com/Hypothesis>

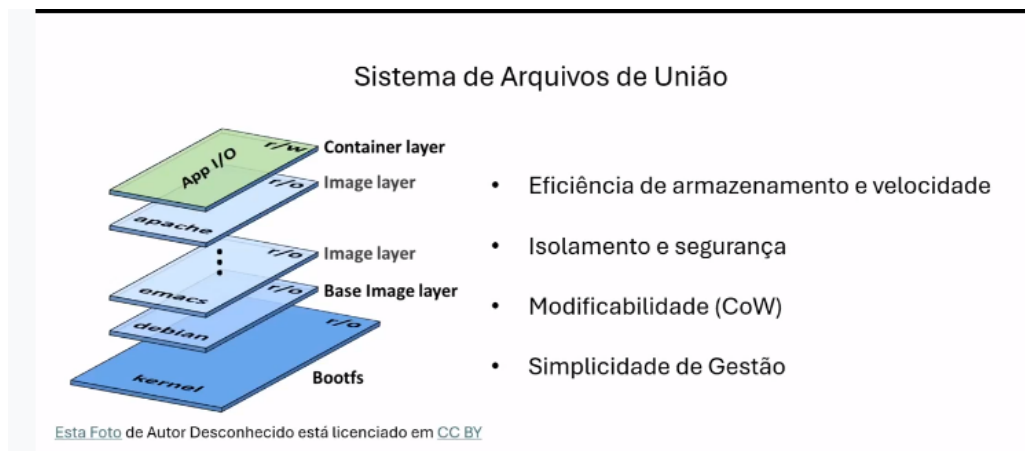
Docker



Docker

Executar um sua aplicação empacotada, em que é possível executar em qualquer lugar sem erros ou dependências, assim é feito o docker, a sua principal função. Esses pacotes se chamam containers, porém com o tempo, cada um podia fazer sua aplicação e tipo de container, assim foi feita uma criação de estrutura para as padronizações dos containers, assim essa tecnologia podia ser usada onde você quiser, pois o container não era só um arquivo que pudesse ser executado só no Docker, agora é open execute.

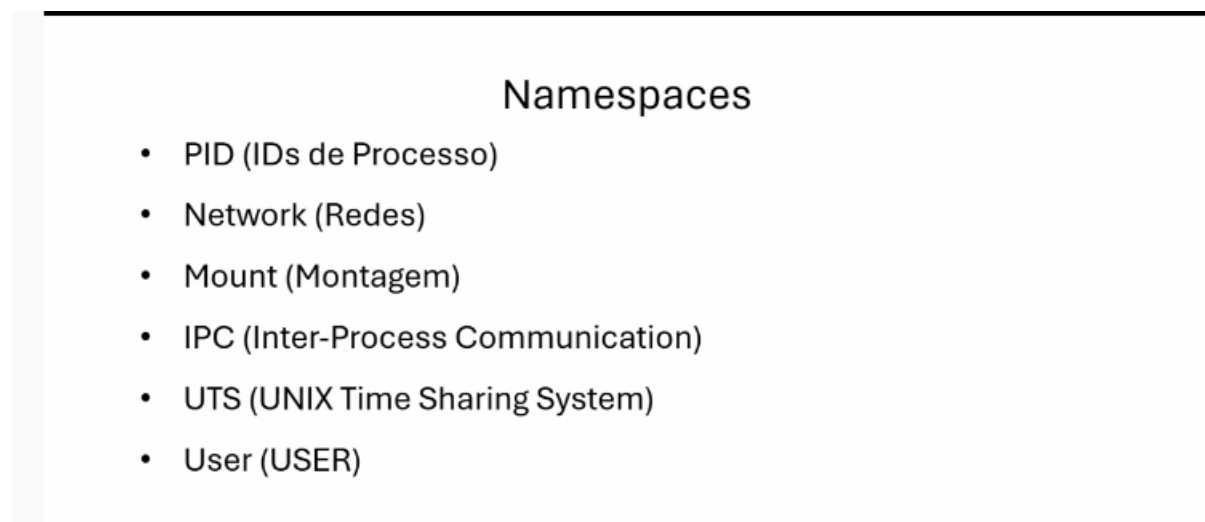
Sistema de Arquivos de União



Arquivos de Sistemas de Uniao facilitam a gestao das imagens Dockers, assim produzindo uma arquitetura de composição robusta e eficaz. Sao montadas para fazer um arquivo completo de execução, permitindo que os containers mais rápidos e completos.

Se é possível modificar uma parte de imagem, como o App I/O só é modificado e compilado essa parte, sem ter que compilar todo o sistema de imagem por completo.

Namespaces



Fornecem uma nomenclatura em que cada parte do Docker é executado difetente como Kernel do SO, assim é executado suas partições no sistema hospedeiro (host) de forma separada, eficaz e segura.

- PID (ID do processo), assim cada container tem seu processo executado na máquina, assim eles são executados de formas separadas no SO do Host.
- Network (Redes), cada container tem seu IP e sua identificação de se conectar com a rede de forma unitária. Containers podem se comunicar entre si ou com o mundo externo, assim são mais flexíveis com as redes.
- Mount (Montagem), isola cada ponto de montagem do container com o Host. As alterações de sistemas de arquivos de um container, não afeta outro.
- IPC (Inter-Process Communication), não faz a comunicação com outros processos de outros containers de forma NÃO segura.
- UTS (UNIX Time Sharing System), namespace que permite que cada container possa ter seu próprio nome e domínio NIS (Network Inter Service)
- User , namespace que isola o nome do usuário de um container do outro. Um Host de um container que é root, não significa que ele tem acesso privilegiado com outros containers.

Artefatos de Software

Artefatos de Software

Um artefato é um subproduto do desenvolvimento de software

- Design de Projeto: casos de uso, diagramas de classe, etc.
- **Ferramentas de Build:** código-fonte liberado (biblioteca) ou executável (programa)

Artefato de Softwares podem ser bibliotecas liberadas para acesso, ou até programas executáveis

Serviço de Registro

Serviços de Registro

Um serviço de registro é um sistema centralizado onde os artefatos de software são armazenados e gerenciados.

Serviços de Software tem o objetivo de que os containeres possam tem seus artefatos de softwares implementados, executados, compartilhados e baixados para com outros ambientes de softwares.

Serviços de Registro

Armazenamento e Gerenciamento:

- Armazena artefatos de software
- Gerencia versões

Acesso

- Público e Privado
- Controle de Acesso

Serviços de Registro

Integração e Automação:

- Integração com ferramentas de CI/CD
- Webhooks e APIs

Benefícios

- Centralização
- Reprodutibilidade
- Segurança
- Facilidade

Colaboração entres os desenvolvedores, sistema integrado, e multi-serviços para varios ambientes de Softwares fazem parte do serviço de serviço.

Serviços de Registro

Exemplos

- Maven Central
- NPM Registry
- DockerHub

Maven Central serviço de serviço para Java

NPM Registry serviço de resgistro para Node.js

DockerHub armazena e disponibiliza imagens dockers e containeres

DockerHub



Docker Hub

Nuvem para distribuição de imagens dockers, é uma rede social para a comunidade docker e a dsponibilização da integração para com os usuários do Docker.