

Data Engineering

Filipe Nascimento
fgvn@cesar.school





C . E . S . A . R

Pessoas impulsionando inovação.
Inovação impulsionando negócios.

Everton Dias
etgdb@cesar.org.br

Janaína Branco
jcb@cesar.org.br

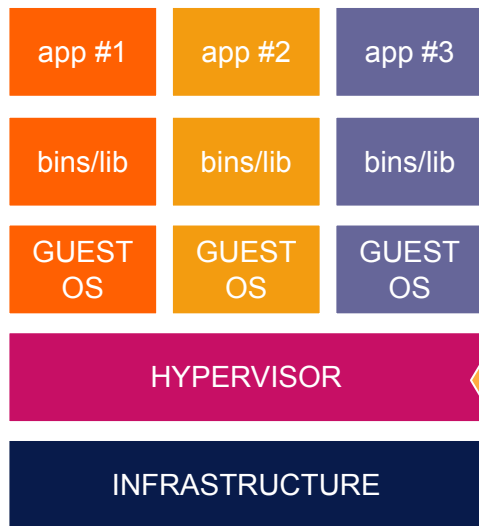


CON TÊINER ES

Nesta etapa faremos uma introdução rápida a tecnologia dos contêineres que nos ajudarão na configuração do nosso ambiente de trabalho e futuramente no deploy de nossas pipelines de dados.

DOCKER

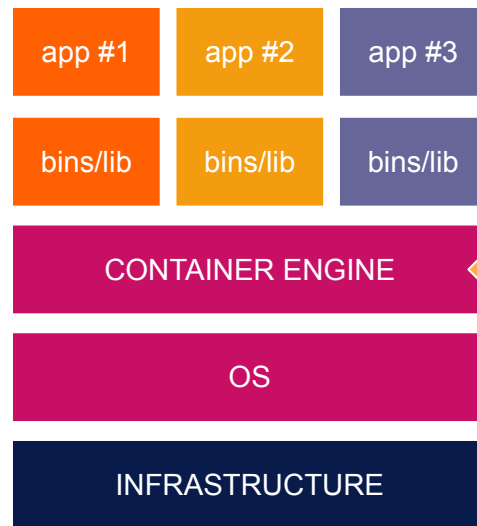
CONTÊINERES



VM's



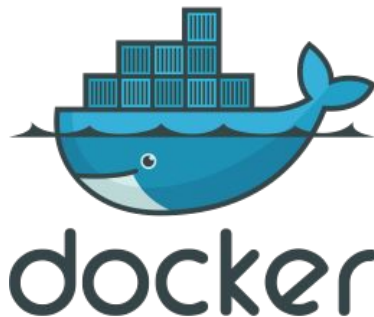
Microsoft Hyper-V



CONTAINER BASED



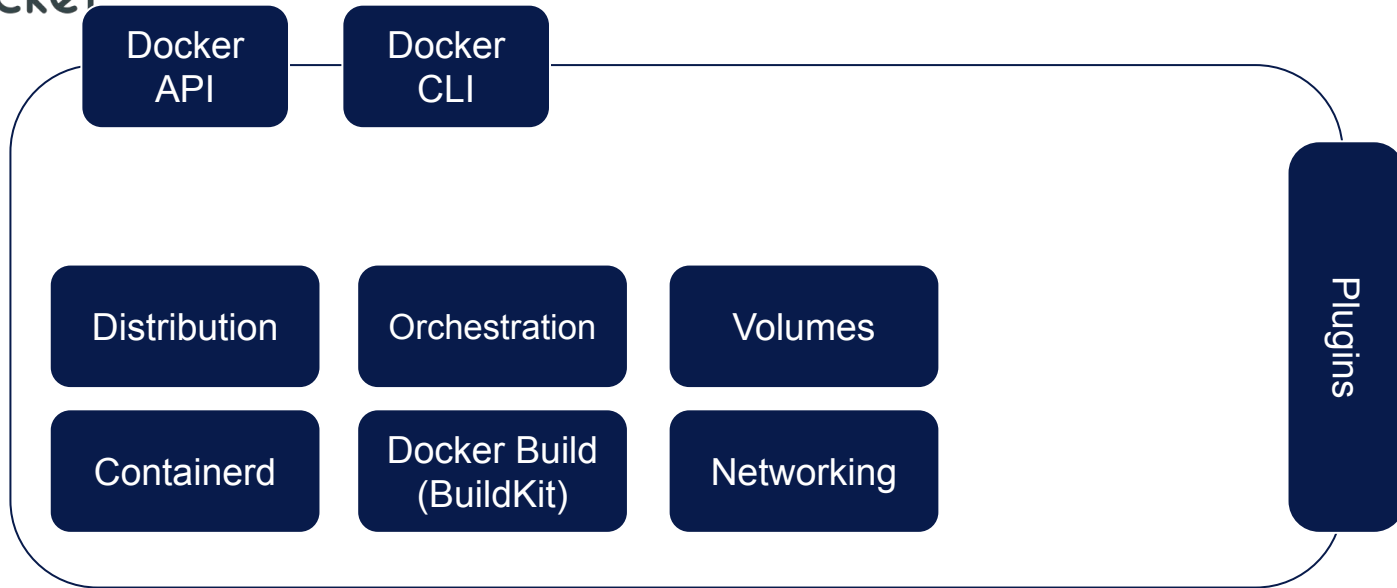
DOCKER



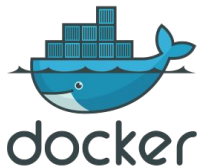
Surgiu da empresa dotCloud que depois passou a ser chamada de Docker, Inc - que era uma empresa PaaS, que utilizava a AWS para provisionamento de infraestrutura, onde realizava o deployment de aplicações dentro de um conceito novo chamado de contêineres, tornando possível hospedar várias aplicações em uma mesma máquina.

Sua criação partiu da necessidade de economizar nos gastos da empresa, permitindo subir várias aplicações em um mesmo IaaS na AWS.

DOCKER



DOCKER



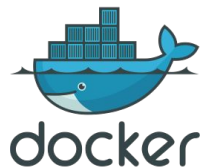
O **Docker** é uma coleção de tecnologias para facilitar o *deploy* e a execução das nossas aplicações. A sua principal tecnologia é a **Docker Engine**, a plataforma que segura os *containers*, fazendo o intermédio entre o sistema operacional.

Outras tecnologias importantes do Docker são:

- **Docker Compose** - um jeito fácil de definir e orquestrar múltiplos *containers*;
- **Docker Swarm** - uma ferramenta para colocar múltiplos *docker engines* para trabalharem juntos em um *cluster*;
- **Docker Hub**, um repositório com mais de 250 mil imagens diferentes para os nossos *containers*;
- **Docker Machine**, uma ferramenta que nos permite gerenciar o Docker em um *host* virtual.

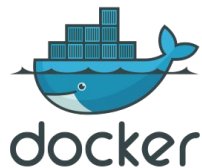
PRÁTICA

Exercício



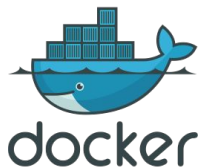
1. Crie uma conta no dockerhub - <https://hub.docker.com/>
2. Acesse a url : <https://labs.play-with-docker.com/>
3. Realize o login e vamos nos divertir!

COMANDOS DOCKER



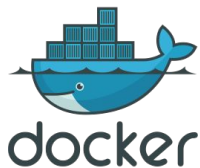
- `$docker --version`
 - Este comando retorna a versão do docker instalado;
- `$docker images`
 - Lista as imagens baixadas localmente;
- `$docker ps`
 - Lista os contêineres que estão em execução no momento;

COMANDOS DOCKER



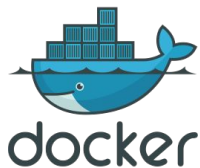
- **\$docker pull**
 - Este comando é utilizado para baixar imagens de um repositório docker como o docker hub por exemplo. (hub.docker.com);
- **\$docker run**
 - Este comando é utilizado para criar um container a partir de uma imagem, caso a imagem não exista ele irá baixá-la do seu repositório;
- **\$docker search**
 - Este comando permite que busquemos imagens compatíveis com os termos informados na busca em um repositório

COMANDOS DOCKER



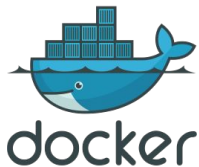
- **\$docker stats**
 - Este comando retornará informações acerca do percentual de uso de CPU, Memória, IO de internet entre outros processos do container;
- **\$docker inspect**
 - Permite inspecionar informações mais detalhadas acerca da sua imagem ou contêiner;
- **\$docker rmi**
 - Remove uma imagem do seu repositório local;

COMANDOS DOCKER



- **\$docker exec**
 - Permite executar comandos nos contêineres sem no entanto estar no console deles:
 - -i : permite interagir com o contêiner (modo interativo)
 - -t : associa seu terminal ao terminal do contêiner
 - --name <nome> : atribui um nome ao contêiner em execução;
 - -p 3060:3060 : mapeia a porta 3060 do contêiner para a porta 3060 do host
 - -d : executa o comando em modo detached (background) liberando o console
 - -v /pasta/host:/pasta/container : cria um volume '/pasta/container' dentro do container com o conteúdo da pasta '/pasta/host' do host

COMANDOS DOCKER



- `$docker logs`
 - Permite verificar os logs de um contêiner;
- `$docker start`
 - Permite inicializar um contêiner através de seu nome ou id;
- `$docker stop`
 - Permite parar um contêiner através de seu nome ou id;

PRÁTICA

Exercício



1. Acessando o ambiente do PWD
 - a. Crie duas instâncias e na primeira instância execute as seguintes ações;
 - i. Verifique a versão do docker disponibilizado através da linha de comando;
 - ii. Baixe a imagem do SGBD postgres;
 - iii. Crie dois contêineres a partir da imagem baixada do postgres;
 - iv. Crie um container a partir da imagem do NGINX e exponha na porta 8080
 - v. Pare os contêineres postgres;
 - vi. Instale um contêiner do Mysql colocando a senha de acesso root como 'rennerroot'
 - b. Na segunda instância crie um container do MariaDB colocando a senha de acesso como 'rootrenner'

DOCKER COMPOSE

O docker compose é uma ferramenta que permite automatizar a criação de vários contêineres onde com apenas um arquivo você pode subir um ecossistema com vários contêineres, com suas regras de configuração de ambiente e rede.



DOCKER COMPOSE

Sua construção pode ser divididas nos seguintes passos:

1. Dividir suas aplicações em **services**;
2. Baixar ou construir suas imagens;
3. Configurar variáveis de ambiente;
4. Configurar a comunicação de rede (networking);
5. Configurar os volumes;
6. Construir e executar o ambiente.



DOCKER COMPOSE

```
docker-compose.yml x
docker-compose.yml
1  version: '3.1'
2
3  services:
4    dbpg1:
5      image: postgres
6      networks:
7        - principal-renner
8      ports:
9        - "5432:5432"
10
11  networks:
12    principal-renner:
13      driver: bridge
```



DOCKER COMPOSE

```
docker-compose2.yml
1  version: '3.4'           # versão do docker-compose
2  services:
3    app:
4      image: renner/web      # imagem do container
5      container_name: app    # nome do container definido pelo usuario
6      ports:
7        - 8080:80           # portas <exposta>:<container>
8      depends_on:
9        - db                # container que também deve ser iniciado
10   db:
11     image: mysql
12     container_name: db
13     environment:           # variáveis de ambiente do mysql
14       - MYSQL_USER=root
15       - MYSQL_ALLOW_EMPTY_PASSWORD=yes
16       - MYSQL_DATABASE=loja
17     volumes:
18       - ./dados:/docker-entrypoint-initdb.d # mapeia o volume <exposta>:<container>
```

PRÁTICA

Exercício



1. Crie um arquivo docker-compose.yml que realize a subida de todos os serviços citados no exercício anterior, tomando cuidado para não haver conflito de portas.



DÚVIDAS