

DevOps

Jemerson Fernando Maia

**Pós-graduação em Desenvolvimento de Aplicações Web e
Mobile**

UNIPAR - Universidade Paranaense



O que é DevOps?

O Conflito



o problema não
são as máquinas,
é o código

o problema não é
o código, são as
máquinas

DevOps

DevOps (Development + Operations) é um processo de desenvolvimento e entrega de software que enfatiza a comunicação entre os profissionais de desenvolvimento e operações.

É um conjunto de práticas que automatizam os processos entre equipes de desenvolvimento de software e de TI para que possam criar, testar e liberar softwares de maneira mais rápida e confiável.

A combinação de filosofias culturais, práticas e ferramentas que aumentam a capacidade de uma empresa de distribuir aplicativos e serviços em alta velocidade.



Benefícios

- Melhorar a frequência dos deploys;
- Automatizar processos;
- Diminuir a ocorrência de erros em novas versões;
- Curtos períodos de tempo para mudanças e melhorias;
- Recuperação rápida em caso de falhas no ambiente;
- Padronização nos processos de configuração e servidores;
- Racionalização de recursos humanos e físicos;

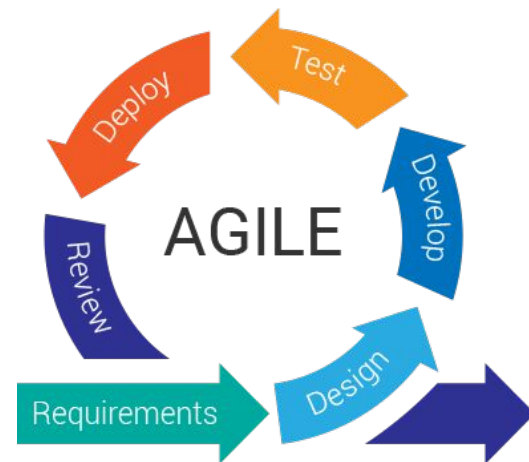




Como tudo
começou

Manifesto Ágil

- Indivíduos e interações, mais que processos e ferramentas
- Software em funcionamento, mais que documentação abrangente
- Responder a mudanças, mais que seguir um plano



História

- Agile Conference - 2008
 - Infraestrutura Ágil de Andrew Schafer para Patrick Debois
- Velocity Conference da O'Reilly - 2009
 - John Allspaw e Paul Hammond apresentaram a famosa palestra chamada “10+ Deploys per Day: Dev and Ops Cooperation at Flickr” no qual Patrick Debois não estava
- DevOpsDays - 2009
 - Patrick Debois decidiu criar sua própria conferência na Bélgica, que ficou conhecida como DevOpsDays
 - Para lembrar o dia foi criada a tag #DevOps no Twitter

<https://devopsdays.org/about>

<https://devopsdays.org/>



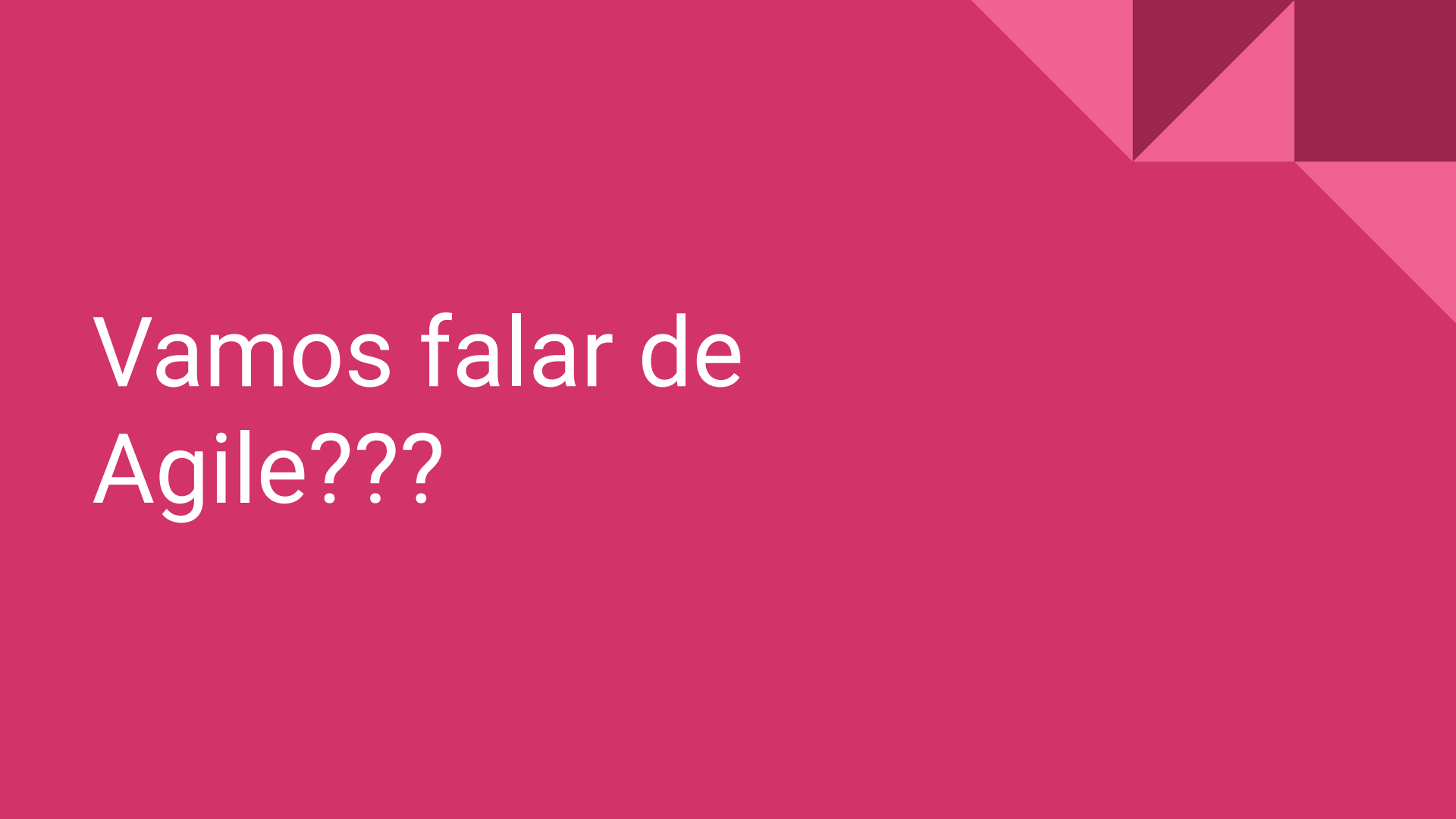
DevSecOps

O DevSecOps incorpora automaticamente a segurança em todas as fases do ciclo de vida de desenvolvimento de software, permitindo o desenvolvimento de software seguro na velocidade do Agile e do DevOps.



C.A.M.S. (Culture, Automation, Measure, Sharing)

- Culture
 - É preciso colaborar, compartilhar e entender a importância de manter uma relação saudável entre todas as áreas.
- Automation
 - Identificar os processos que sejam repetitivos ou que levam bastante tempo e buscar resolver o quanto antes.
- Measure
 - Deve-se medir tudo que é possível: performance, processos, interações e até mesmo pessoas. O processo de melhoria contínua é o coração do DevOps!
- Sharing
 - Ter uma boa comunicação entre as equipes, incentivar as pessoas a se comunicarem e compartilharem ideias e problemas (Blameless).



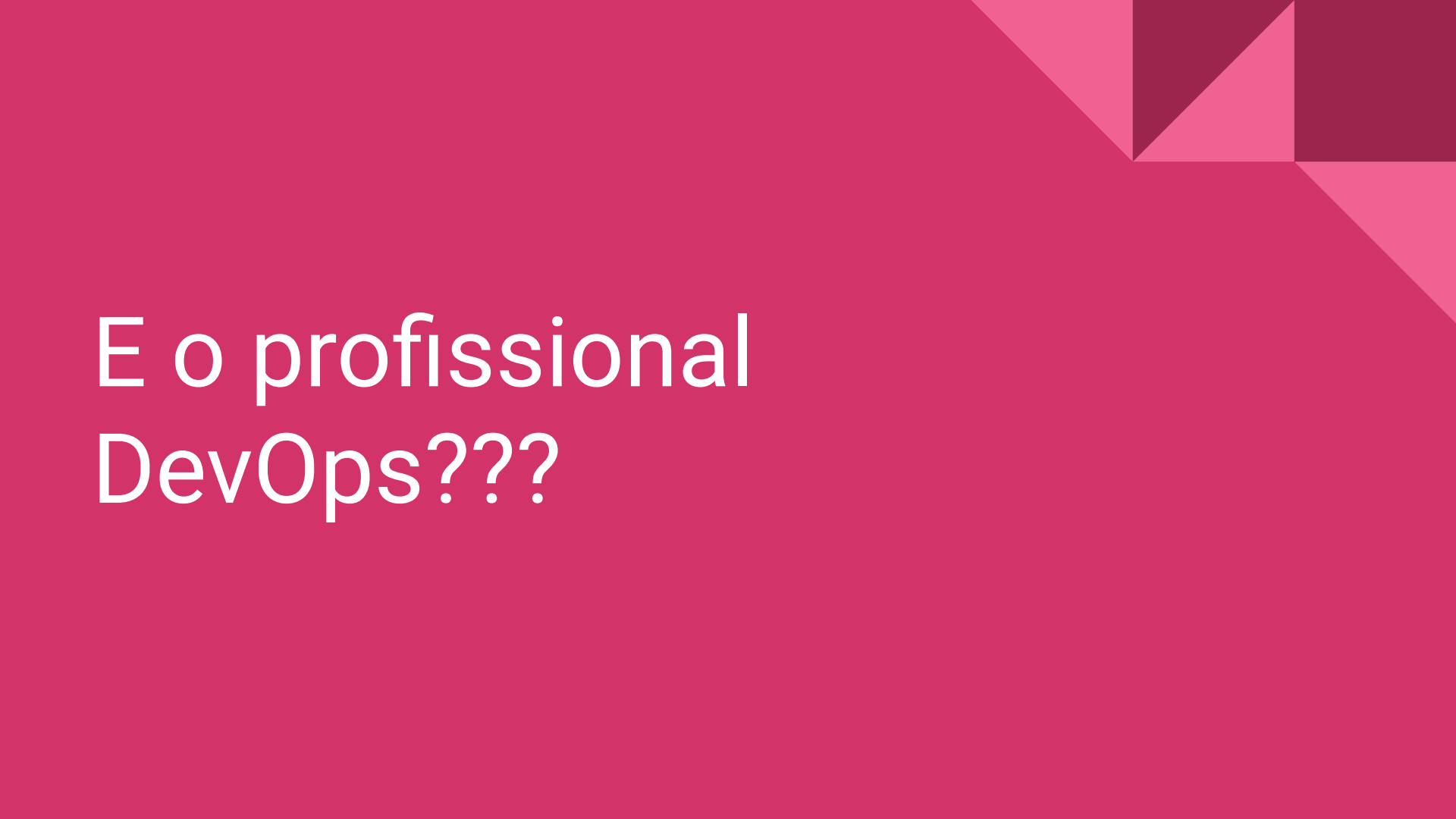
Vamos falar de
Agile???

Agile

Agile é um processo de desenvolvimento de software que foca em pequenos entregáveis e um processo cíclico que sempre mobiliza melhorias.

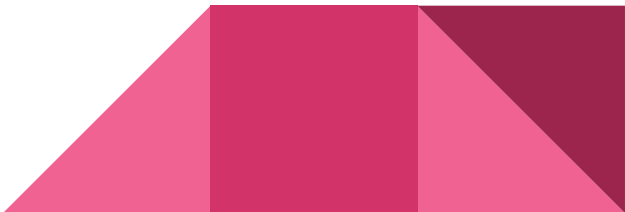
- Scrum
- Sprint
- Daily Standup
- Retrospectiva
- Review
- Planning
- Backlog
- Scrum Master
- Product Owner





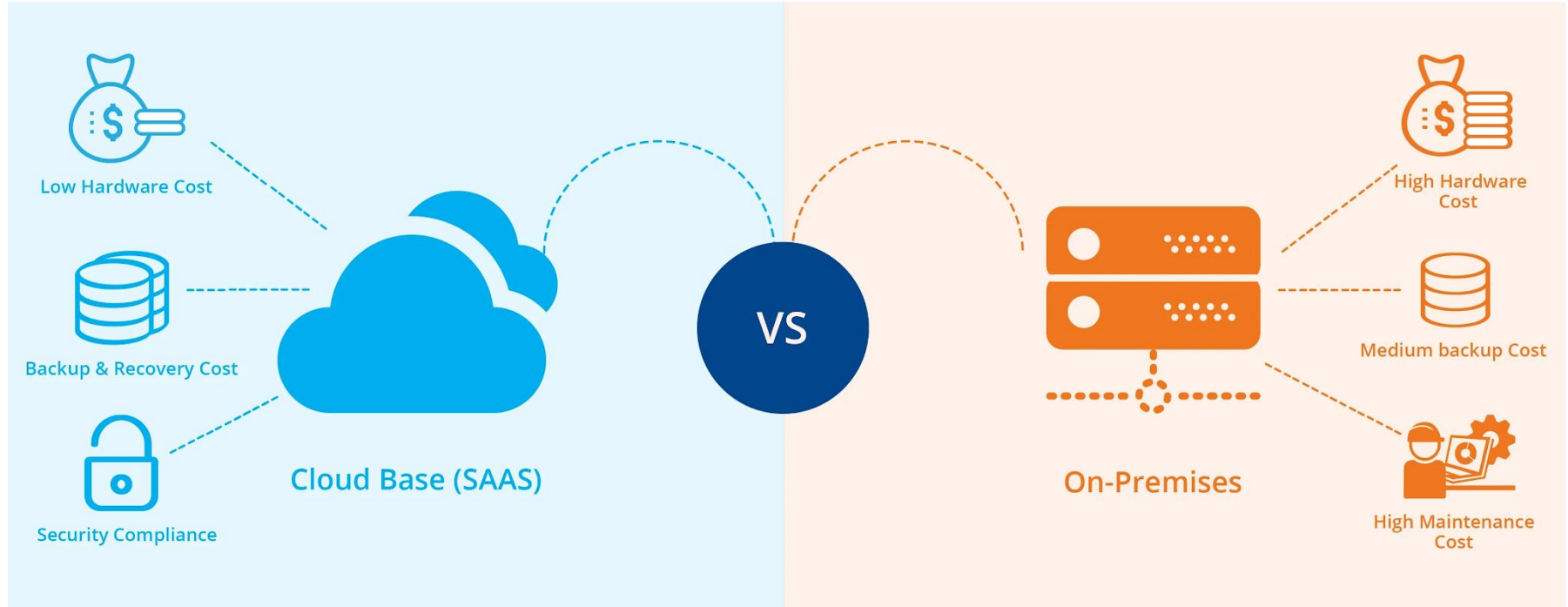
E o profissional
DevOps???

O que é preciso aprender???

- Cultura;
 - Programação;
 - Gerenciar Servidores, Virtualização, Redes e Segurança;
 - Criação de Scripts;
 - Instalar e configurar middlewares (apache, nginx, haproxy);
 - Instalar softwares;
 - Versionar arquivos com GIT;
 - Automatizar tarefas e gerenciar configurações;
 - Infraestrutura como Código;
 - Observabilidade;
 - Orquestração de Containers;
 - Compartilhar
- 



Cloud
vs
On premises





Mercado

Mais de 40 estatísticas de DevOps que você deve conhecer em 2022

- Em 2021, 83% dos tomadores de decisão de TI relataram implementar práticas de DevOps para liberar maior valor comercial.
- A IDC prevê que o mercado de DevOps deve crescer de US\$ 2,9 bilhões para US\$ 8 bilhões até 2022.
- O segmento de grandes empresas no mercado de DevOps do Brasil deve atingir um CAGR (Compound Annual Growth Rate) de cerca de 20% até 2028.
- Pesquisa Atlassian
 - 99% dos entrevistados disseram que o DevOps teve um impacto positivo em sua organização.
 - 78% tiveram que aprender uma nova habilidade.
 - 48% dos entrevistados disseram que isso os ajudou a obter um aumento.
 - 61% disseram que isso os ajudou a produzir produtos de alta qualidade.
 - 69% têm uma equipe com DevOps no nome e/ou funcionários com DevOps em seus títulos.
 - 46% das organizações são relativamente novas no DevOps (menos de 3 anos de experiência).

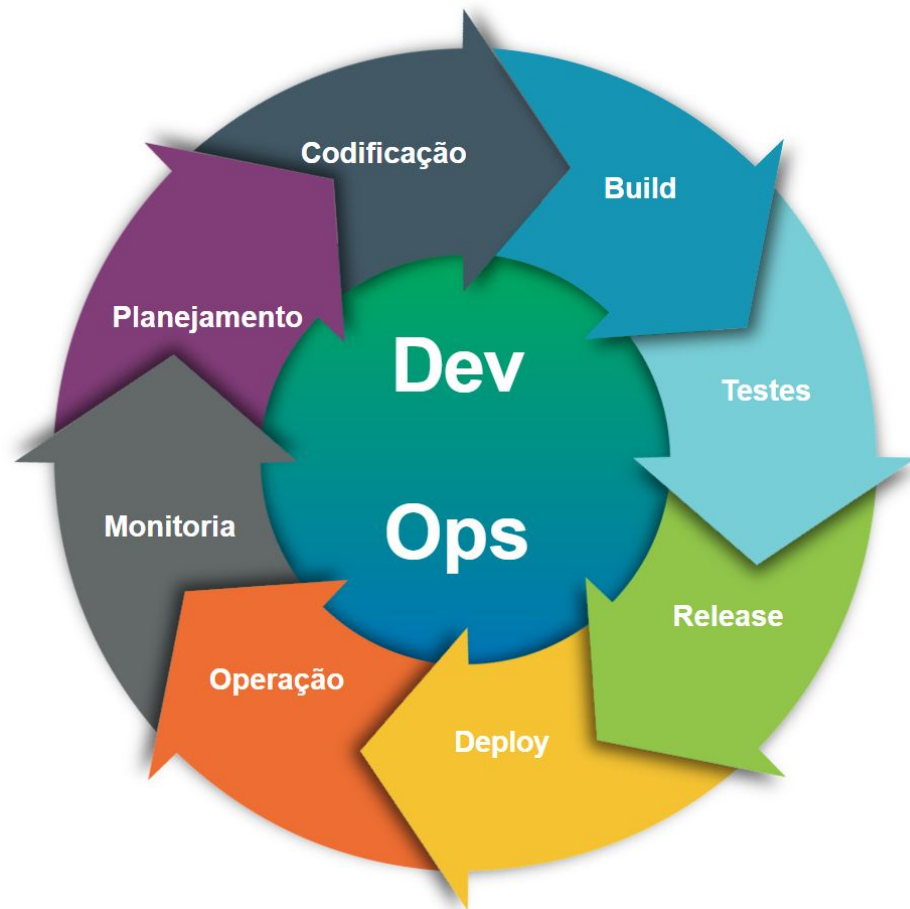
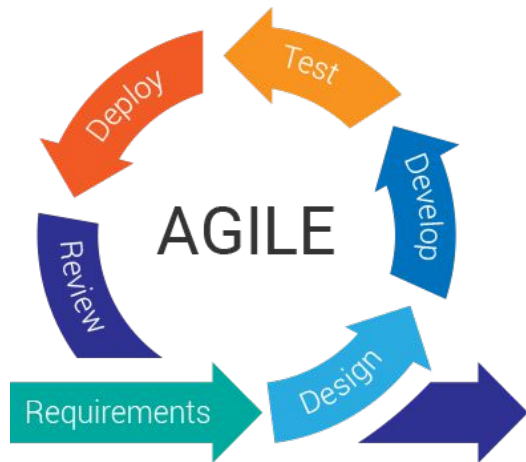
Certificações

Técnicas

- LPI DevOps Tools Engineer (LPIC-OT)
 - Testa o candidato com relação a todos os pilares de DevOps
 - Docker Certified Associate (Docker DCA)
 - Testa o conhecimento do candidato em tecnologias de Containers, focado em Docker e Kubernetes
 - Kubernetes Certified Administrator (CKA)
 - Testa o conhecimento e prática em Kubernetes
 - Kubernetes Certified Application Developer (CKAD)
 - Testa o conhecimento e prática em Kubernetes
 - Certificações Cloud
 - Certificações de Cloud: GCP, AWS, Azure
- 

Pipeline DevOps

Pipeline DevOps





Docker & Container

O que é Docker?

É uma tecnologia de containerização que permite a criação e o uso de containers Linux

Containers são como se fossem máquinas virtuais modulares e extremamente leves



O que isso quer dizer?

O Docker possibilita o empacotamento de uma aplicação ou ambiente inteiro dentro de um **container**, e a partir desse momento o ambiente inteiro torna-se **portável** para qualquer outro Host que contenha o Docker instalado



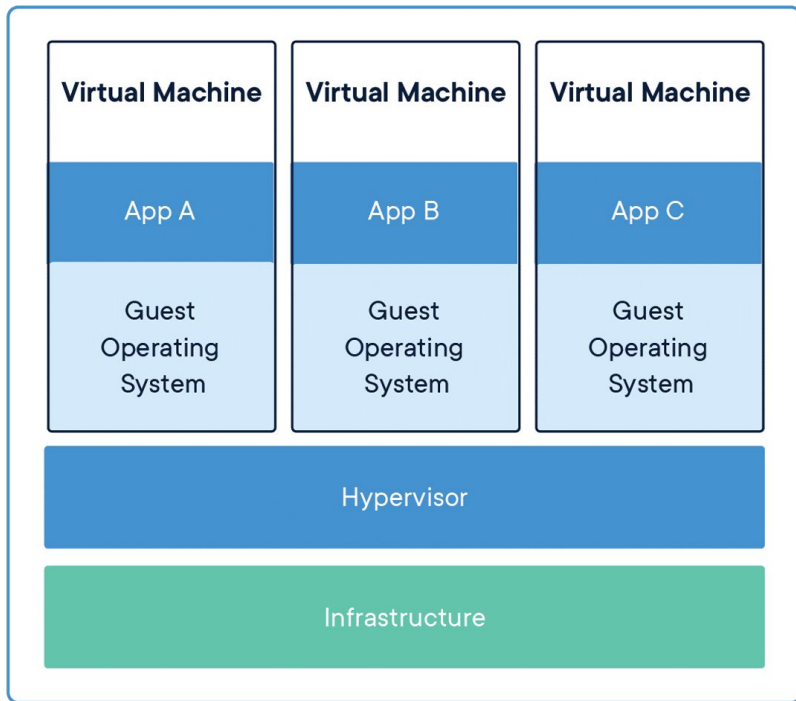


O que tenta resolver?



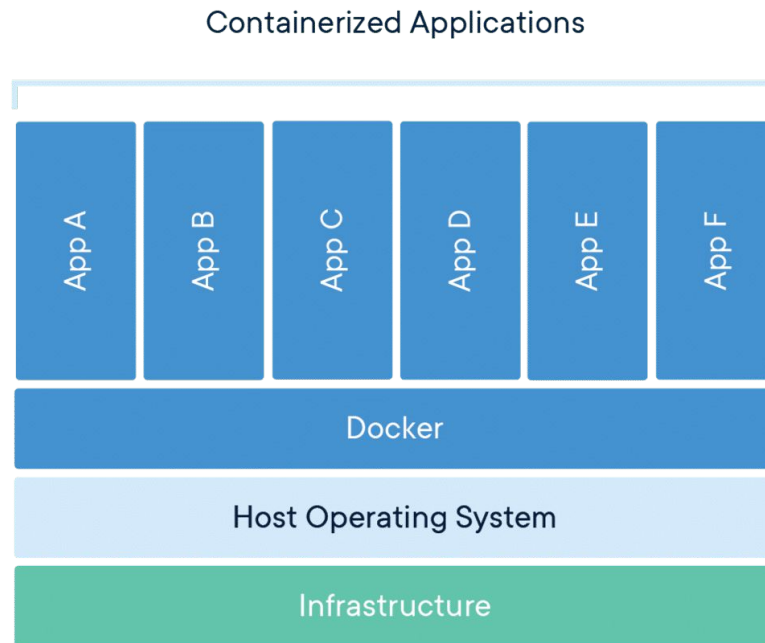
Container

Virtualização

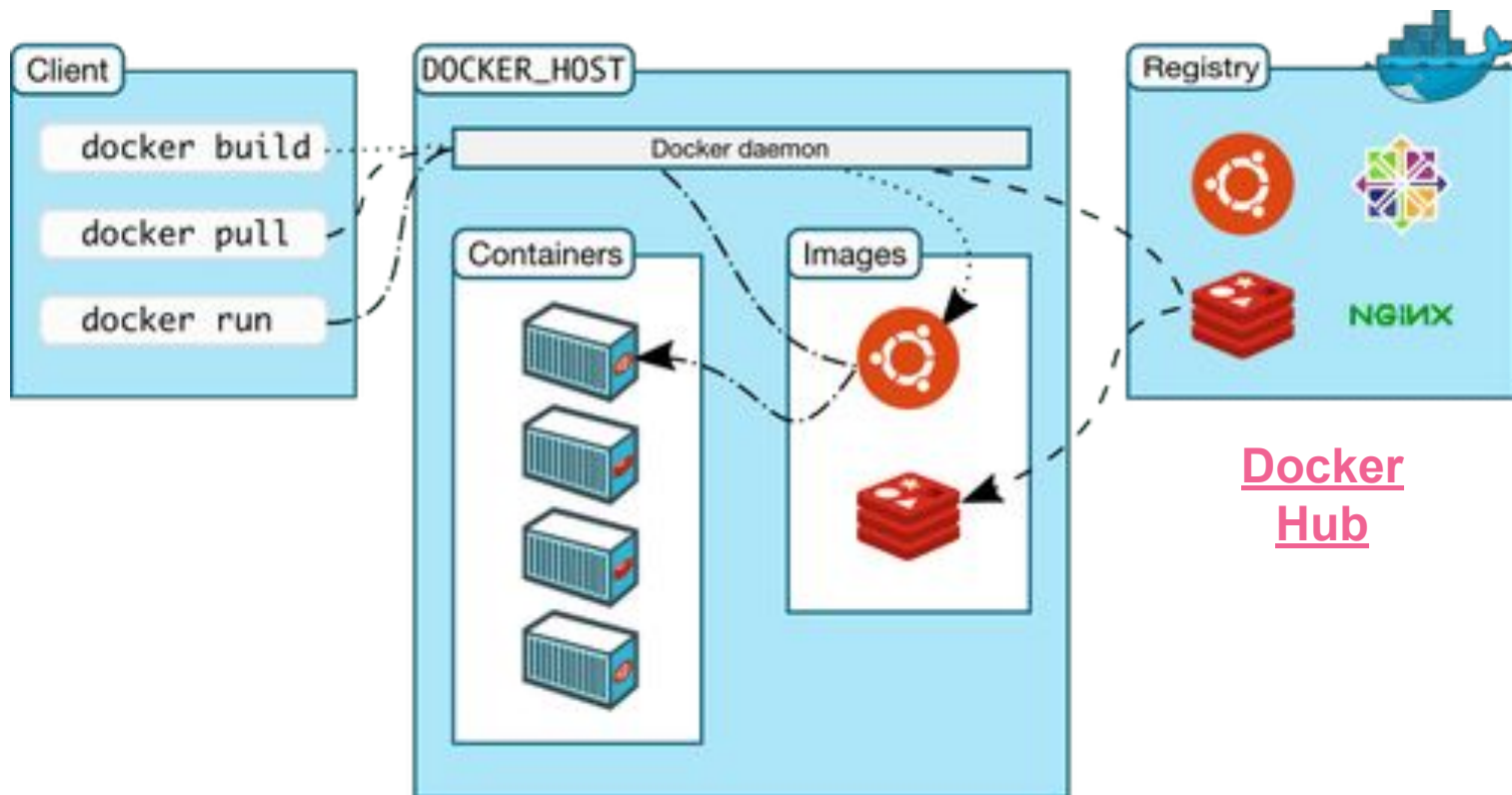


X

Containerização



Arquitetura



Docker in Action

Instalação

<https://docs.docker.com/get-docker/>

You can download and install Docker on multiple platforms. Refer to the following section and choose the best installation path for you.



Docker Desktop for Mac

A native application using the macOS sandbox security model which delivers all Docker tools to your Mac.



Docker Desktop for Windows

A native Windows application which delivers all Docker tools to your Windows computer.



Docker Desktop for Linux

A native Linux application which delivers all Docker tools to your Linux computer.

DockerHub

Build and Ship any Application Anywhere

Docker Hub is the world's easiest way to create, manage,
and deliver your team's container applications.

Get Started Today for Free

Already have an account? [Sign In](#)



- ☐ Send me occasional product updates and announcements.
- ☐ I agree to the [Subscription Service Agreement](#), [Privacy Policy](#)
and [Data Processing Terms](#).



Não sou um robô



hCAPTCHA
Privacy Policy - Terms

Sign Up

Docker in action

```
$ docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
```

```
$ docker run -it --name server1 ubuntu
```

```
$ docker ps -a
```

```
$ docker start server1
```


```
$ docker exec -it server1 bash
```

```
$ docker restart server1
```

```
$ docker stop server1
```

```
$ docker rm server1
```

Alguns comandos

- **docker images**: lista imagens baixadas.
 - **docker search**: procura e lista imagens do docker hub.
 - **docker pull**: baixa uma imagem do docker hub.
 - **docker ps**: lista containers que estão rodando.
 - **docker rm**: remove um container.
 - **docker rmi**: remove uma imagem.
 - **docker run**: cria e inicia um container.
 - **docker start/stop/restart**: inicia, para ou reinicia um container.
- 

Network

```
$ docker network ls
```

- Usada principalmente para estabelecer comunicação entre os contêineres do Docker

NETWORK ID	NAME	DRIVER	SCOPE
fe86fbbee631	bridge	bridge	local
e5f51b3c4df8	host	host	local
13c3474d8d0f	none	null	local

Docker in action

```
$ docker network COMMAND
```

```
$ docker network create site-net
```

```
$ docker run --name db -d --net site-net -e MYSQL_DATABASE=wordpress  
-e MYSQL_ROOT_PASSWORD=123456 mysql
```

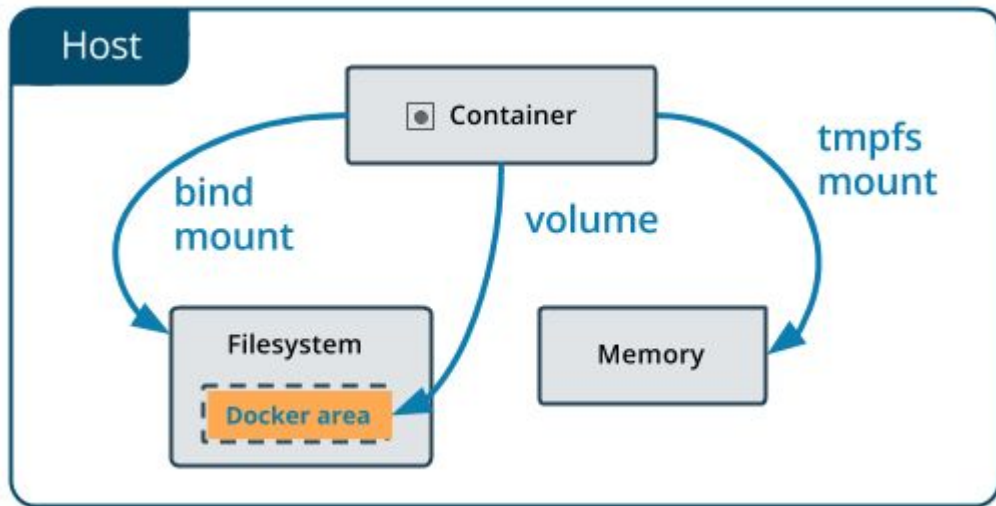
```
$ docker run --name site -d -p 80:80 --net site-net wordpress
```

```
$ docker run --name adminer -d -p 8080:8080 --net site-net adminer
```

Volume

```
$ docker volume ls
```

- Usado para manter dados persistentes, compartilhar arquivos/configurações



Docker in action

```
$ docker volume COMMAND
```

```
$ docker volume create site-db
```

```
$ docker run --name db -d --net site-net -e MYSQL_DATABASE=wordpress  
-e MYSQL_ROOT_PASSWORD=123456 -v site-db:/var/lib/mysql mysql
```

Dockerfile

Dockerfile

Serve como a receita para construir um container, permitindo definir um ambiente personalizado e próprio para seu projeto pessoal ou empresarial.



Dockerfile

```
FROM ubuntu

RUN apt-get update -y; \
    apt-get install -y nginx curl; \
    rm -rf /var/lib/apt/lists/*

VOLUME ["/var/www/html", "/var/log/nginx"]

EXPOSE 80/tcp

WORKDIR /var/www/

COPY entrypoint.sh /entrypoint.sh
RUN chmod +x /entrypoint.sh

ADD index.tar.gz /var/www/html/

ENTRYPOINT ["/entrypoint.sh"]
CMD ["/usr/sbin/nginx", "-g", "daemon off;"]
```

Construindo a imagem e executado o container

```
$ docker build -t unipar/nginx ". "
```

```
$ docker run --name proxy -d -p 80:80  
unipar/nginx
```

Disponibilizando a imagem no docker hub

```
$ docker login
```

```
Authenticating with existing credentials...  
Login Succeeded
```

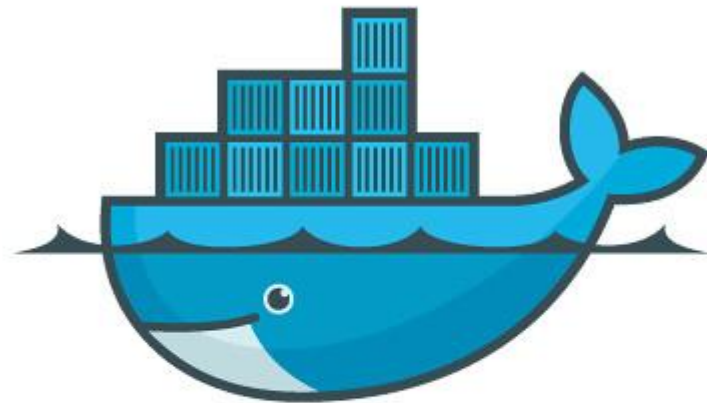
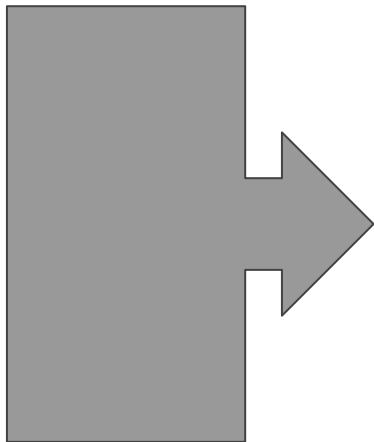
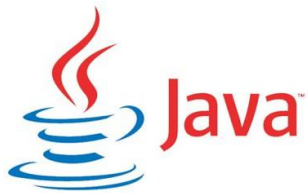
```
Logging in with your password grants your terminal complete access to your account.  
For better security, log in with a limited-privilege personal access token. Learn more at  
https://docs.docker.com/go/access-tokens/
```

```
$ docker tag unipar/nginx account/image-name
```

```
$ docker push account/image-name
```



Containerizando a aplicação



docker



Grails

- Baixar projeto

```
docker run --rm -it -v $(pwd):/workspace -w/workspace -p  
8080:8080 openjdk:8 bash
```

```
java -version
```

```
./grailsw run-app
```

```
./grailsw war
```

```
./grailsw prod war
```



Grails - Dockerfile

```
FROM openjdk:8 as builder
```

```
COPY . /app
```

```
WORKDIR /app
```

```
RUN ./grailsw prod war
```

```
FROM tomcat:9.0.65-jre8
```

```
COPY --from=builder /app/build/libs/*.war  
/usr/local/tomcat/webapps/ROOT.war
```


Grails - Dockernize

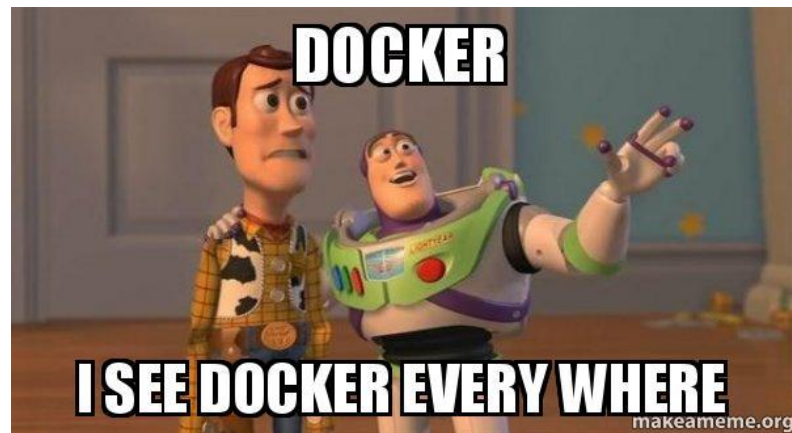
```
$ docker build -t unipar/grails-app "."
```

```
$ docker network create rails-app-net
```

```
$ docker run --name rails-app-db -d --net rails-app-net -e  
MYSQL_DATABASE=myapp -e MYSQL_ROOT_PASSWORD=123456 mysql
```

```
$ docker run --name rails-app -p 8082:8080 -net  
rails-app-net -e DB_SERVER=rails-app-db -e DB_USER=root -e  
DB_PASS=123456 unipar/grails-app
```

```
$ docker run --name adminer -d -p 8080:8080 --net site-net adminer
```




Atividade em grupo

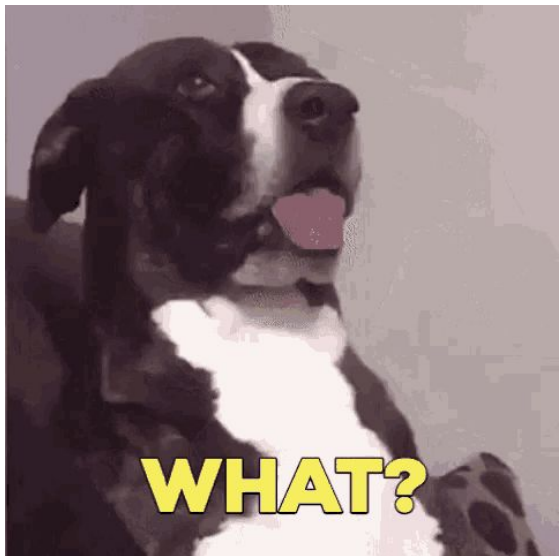
- Utilizar uma aplicação qualquer, de qualquer linguagem, mas uma aplicação Web.
 - Dê preferência com alguma conectividade com algum banco de dados
- Criar um Dockerfile que construa e gere uma imagem da aplicação



Sugestão de Projetos

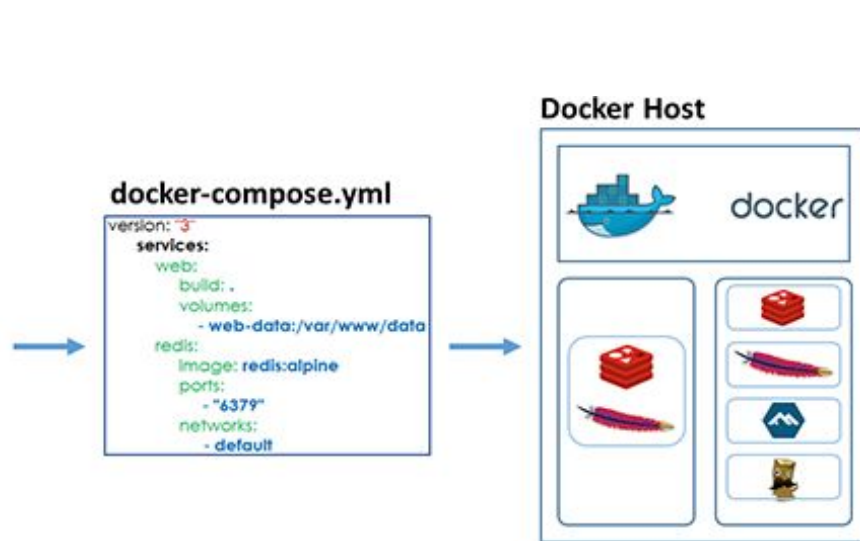
- https://github.com/rayed/django_crud Python
 - <https://github.com/bhiapp4/crudApp> Java
 - <https://github.com/zianwar/springboot-crud-demo> Java
 - <https://github.com/LisandroVillarroel/proyectoSCPC> JavaScript
 - <https://github.com/Bunkermaster/crud-webp2020-g1> PHP
 - <https://github.com/rubydocs/app> Ruby
 - https://github.com/herusdianto/gorm_crud_example Go
 - <https://github.com/miolivc/grails-product-crud> Grails
- 

Docker-Compose



Docker Compose é o orquestrador
de containers da Docker.

Docker-Compose



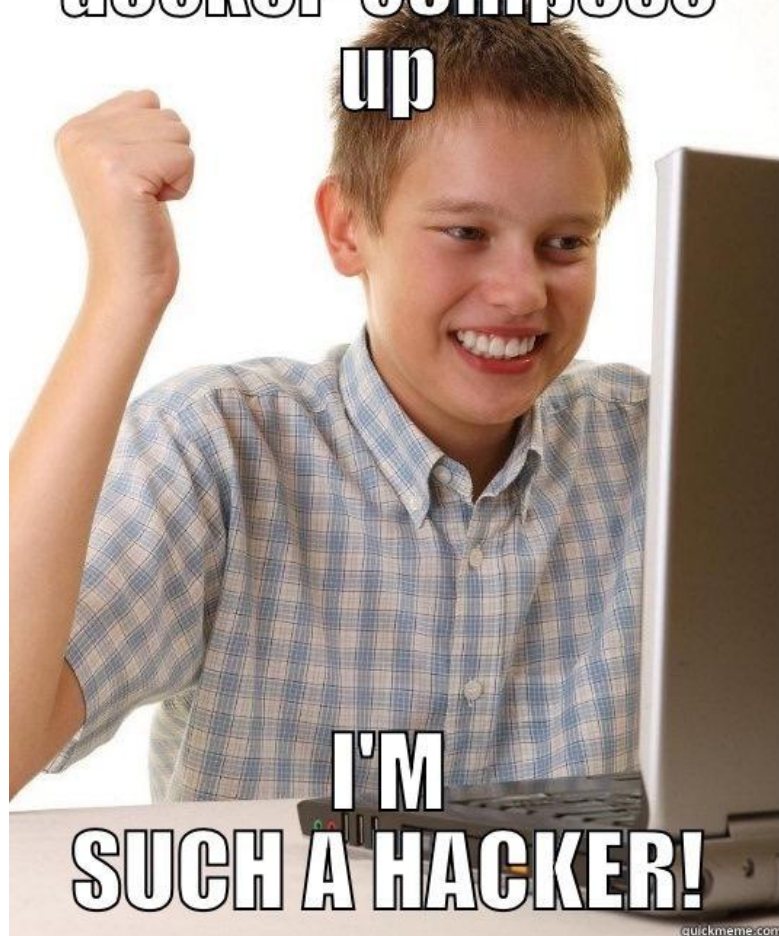
Docker-Compose



<https://github.com/jfnandopr/mini-curso-docker/blob/main/docker-compose.yml>

```
$ docker-compose up -d
$ docker-compose start SERVICE_NAME
$ docker-compose stop SERVICE_NAME
$ docker-compose restart SERVICE_NAME
$ docker-compose down
$ docker-compose ps
$ docker-compose build
$ docker-compose up -d -build
```


**docker-compose
up**



**I'M
SUCH A HACKER!**

