

# Docker

## Aprendendo o Básico

#1 Meetup Mansão Tech  
28/02/2020

Eduardo Daltro

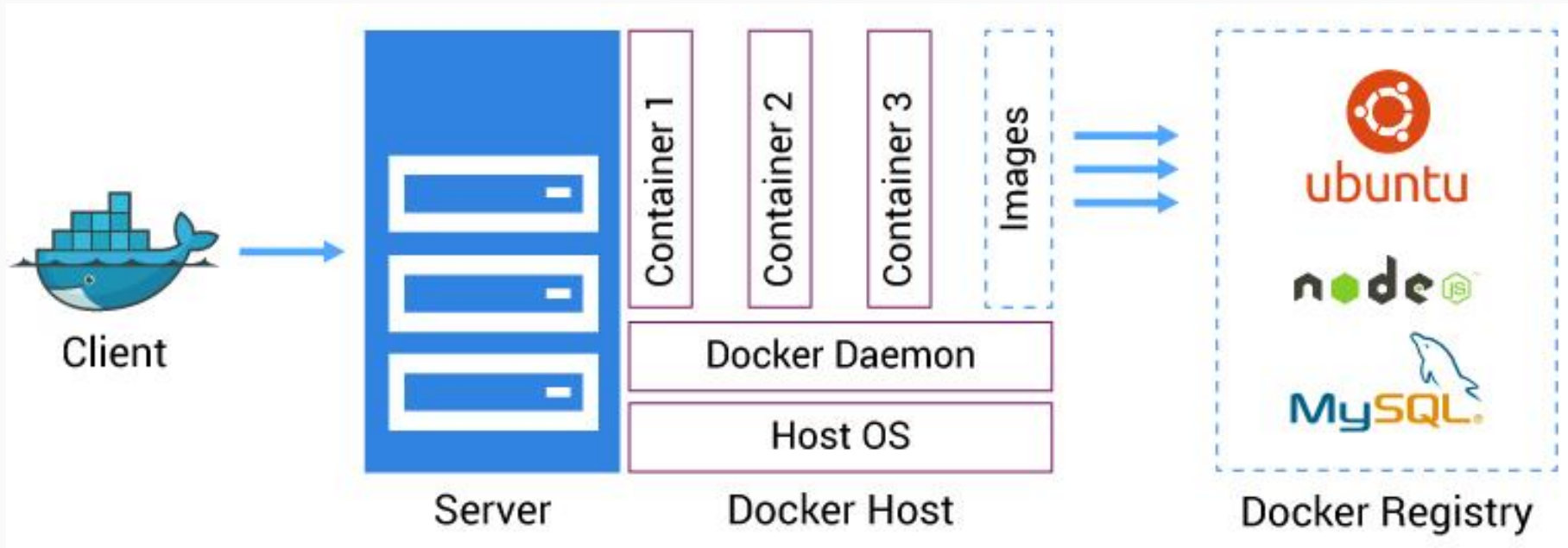
# O que Esperar da Talk

1. Público-alvo: devs que sabem pouco ou nada do assunto
2. Abordagem básica da parte teórica
3. Demonstração prática dos comandos e fluxos de trabalho
4. Conseguir aplicar o conteúdo da talk em projetos iniciais e testes de emprego

# Por que Usar Docker?

1. Aplicação completa em um só lugar
  - Ex: menos dores de cabeça com dependências
2. Ambientes diferentes com códigos iguais
  - Ex: código no seu notebook = código do servidor
3. Times de infra e dev em sincronia
  - Ex: mais facilidade para implantações (deploys)
4. Exigido em muitas vagas (principalmente back-end)

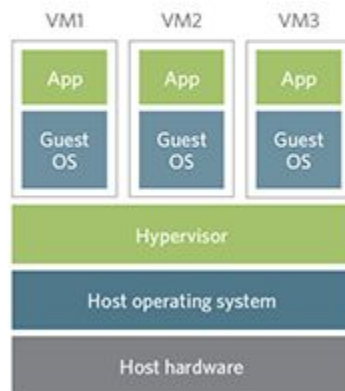
# A Solução



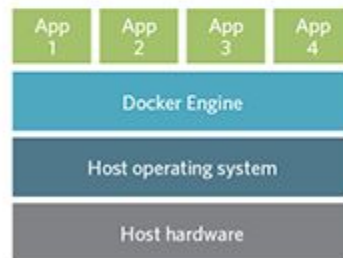
# Docker Não é VM!

## Virtual machines versus containers

### VIRTUAL MACHINES



### CONTAINERS



# Onde Consigo Utilizar?

## 1. Multiplataforma

- a. Linux
- b. Windows
- c. Mac

## 2. Cloud

- a. Heroku
- b. AWS
- c. Digital Ocean
- d. Etc.

# Exemplo Cloud: Deploy na Heroku

1. Instale o Heroku CLI: <https://devcenter.heroku.com/articles/heroku-cli>

2. Login: `heroku login`

3. Login no container: `heroku container:login`

4. Crie a aplicação: `heroku create my_application`

5. Envie a aplicação: `heroku container:push web --app my_application`

6. Coloque pra rodar: `heroku container:release web --app my_application`

É só isso mesmo ;)

# Termos Importantes

1. Dockerfile: arquivo onde são escritas as instruções da imagem
2. Docker Compose: possibilita criar múltiplas imagens
3. Imagem: instruções da aplicação, baseadas em containers originais
4. Container: criado a partir da imagem, executa a aplicação real
5. Docker Engine: serviço que gerencia os recursos do Docker
6. Docker Hub: hospedagem de imagens e documentação  
<https://hub.docker.com/>



# Alguns Comandos

Listar imagens na máquina: `docker image list`

Atualizar imagem: `docker image pull <imagem>`

Inspecionar imagem: `docker image inspect <imagem>`

Executar:

`docker container run <parâmetros> <imagem> <CMD> <argumentos>`

Listar containers: `docker ps <parâmetros>`

Parar um container: `docker kill <nome/id_processo>`

# Próximos Passos

- Aprender mais:
  - Docker para Desenvolvedores  
<https://leanpub.com/dockerparadesenvolvedores>
  - Docker Curriculum  
<https://docker-curriculum.com>
- Entrar no grupo: <https://t.me/devfsa>
- Me add: @daltroedu 