

FIAP

# PLATAFORMAS E SERVIÇOS COGNITIVOS

PROF. DR. FERNANDO TIMOTEO FERNANDES

# Agenda

---

- Introdução a contêineres
- Cloud9
- AWS ECS
- AWS ECR

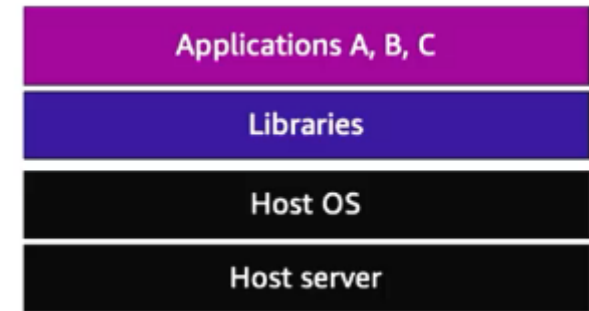


Antes do surgimento dos contêineres ...

# Antes dos contêineres... – Bare Metal

---

- Aplicações locais competem pelos recursos do servidor
- Diferentes SOs
- Diferentes formas de instalação
- Demora na instalação



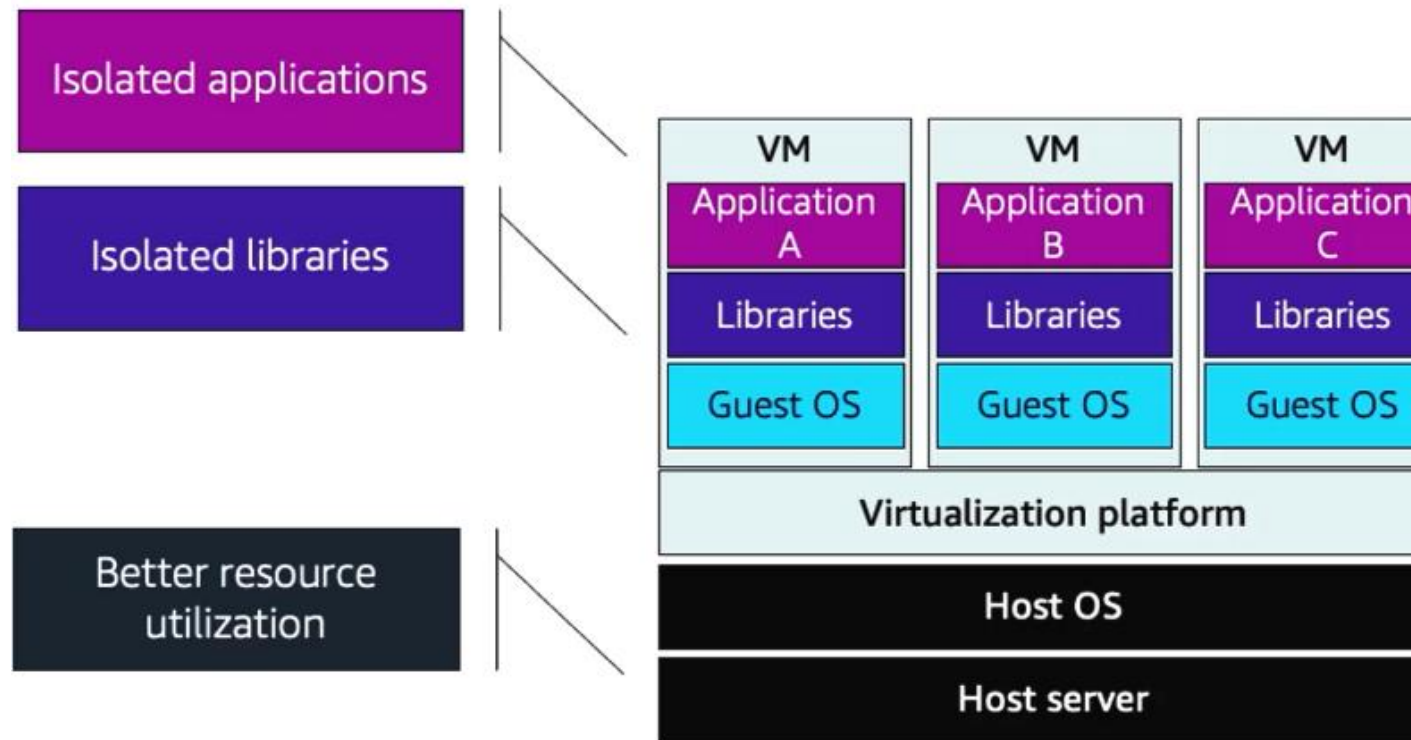
Fonte: AWS





# Antes dos contêineres... – VMs

---



Fonte: AWS

# Antes dos contêineres...

---

Equipe de desenvolvimento



Instruções de  
como instalar  
o servidor web  
e o banco de  
dados da  
aplicação

Equipe de Infra  
(Operações)



Problemas de  
dependências e  
configuração





Depois dos contêineres ...



# O qué um Contêiner ?

---

O que é um contêiner ?

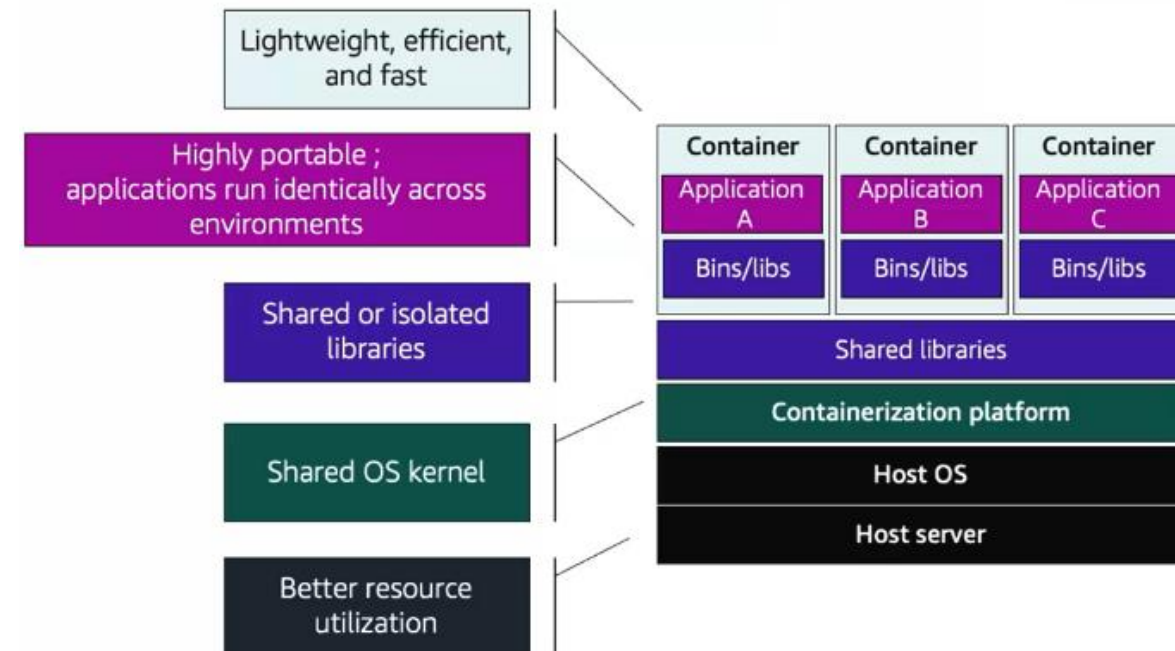
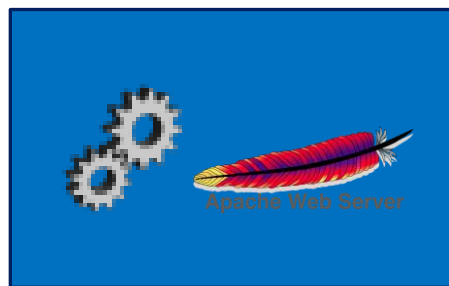
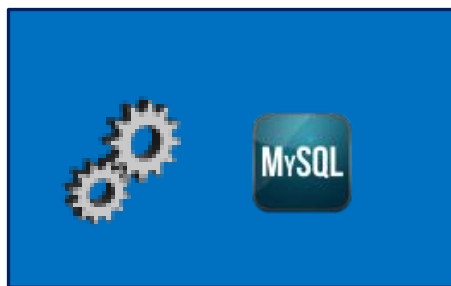
- Contêineres permitem empacotar uma aplicação com todas suas dependências e configurações
- Portátil
- Pode ser compartilhado facilmente





# Depois dos contêineres...

- Ambiente isolado
- Pacotes com todas as dependências
- Um único comando para instalação
- Diferentes versões de apps

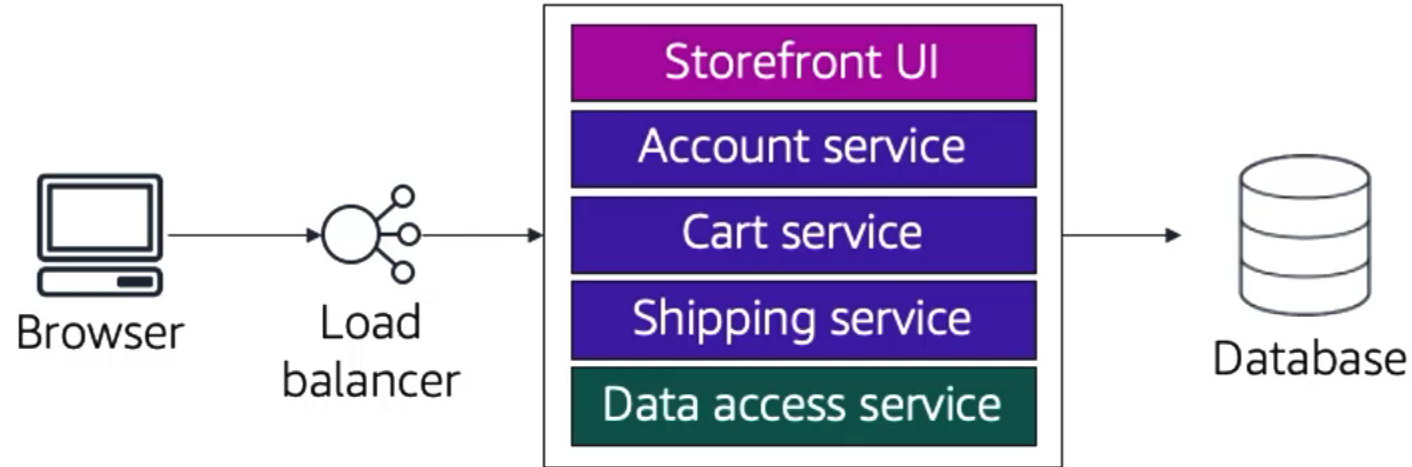


Fonte: AWS

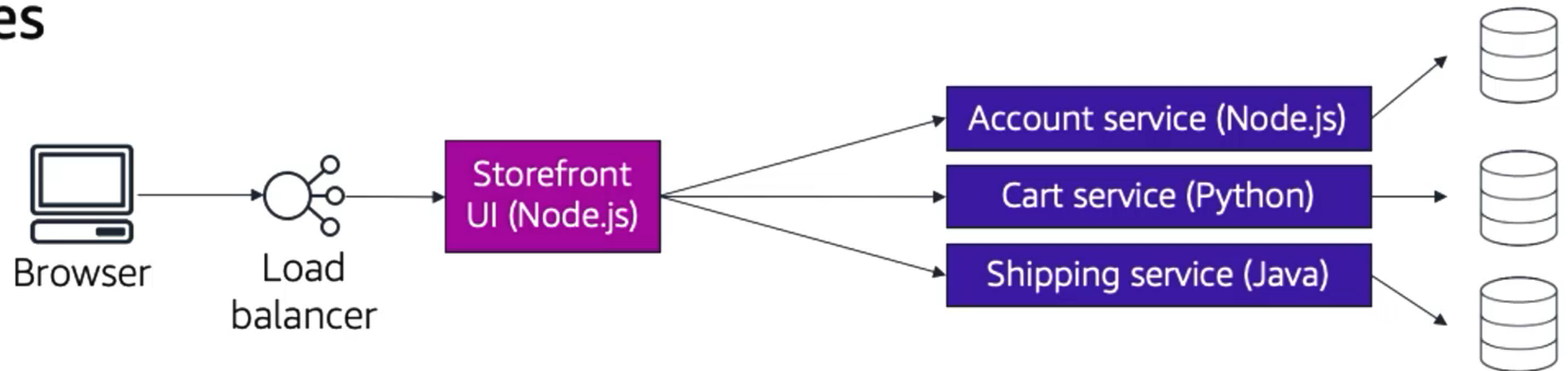
# Aplicações Monolíticas vs Microserviços

---

## Monolithic

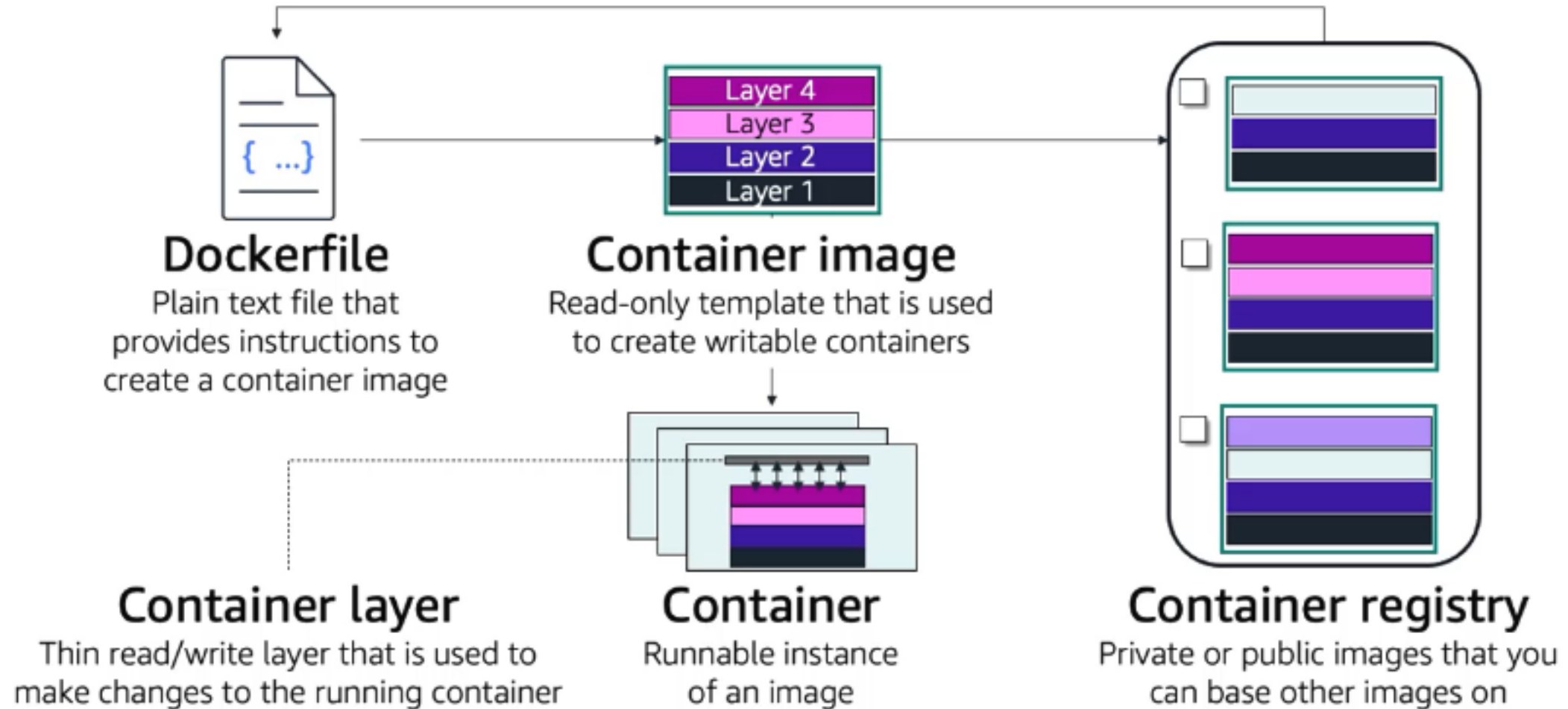


## Microservices





# Docker Containers



# Contêineres - Vantagens

---

## Vantagens


- Menor Tamanho
- Mais rápido para iniciar/parar
- Compatibilidade

## Desvantagens/Limitações

- Roda qualquer contêiner somente a partir do win 10
- Falta de algumas funcionalidades:
  - Ex: transferir arquivos entre o host e o container
- Sem interface gráfica
  - Exceto o Docker desktop para algumas opções




# Instalação - Docker

 [docker hub](#)

[Explore](#) [Pricing](#) [Sign In](#) [Sign Up](#)

[Explore](#) [Verified Publishers](#) [Docker Desktop for Windows](#)



## Docker Desktop for Windows

By [Docker](#) Verified Publisher

The fastest and easiest way to get started with Docker on Windows

[Edition](#) [Windows](#) [x86-64](#)

### Get Docker Desktop for Windows

Requires Microsoft Windows 10 Professional or Enterprise 64-bit, or Windows 10 Home 64-bit with WSL 2.

We updated the [Docker Subscription Service Agreement](#) on August 31, 2021. Please read the announcement and FAQs to learn more.

[Subscription Service Agreement](#) | [Data Processing Agreement](#) | [Data Privacy Policy](#)

[↓ Get Docker Desktop](#)

<https://hub.docker.com/editions/community/docker-ce-desktop-windows>

# Instalação – Baixar e executar primeiro container

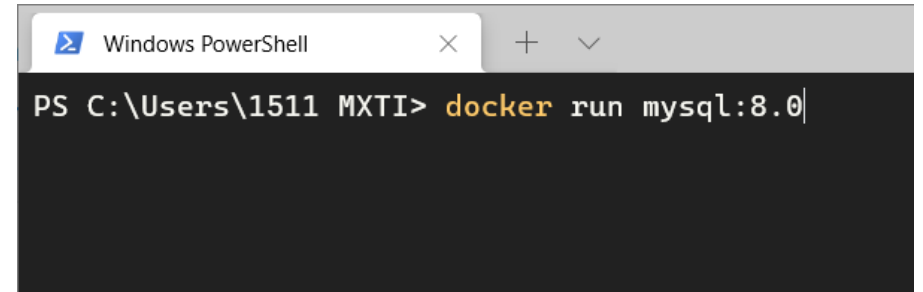
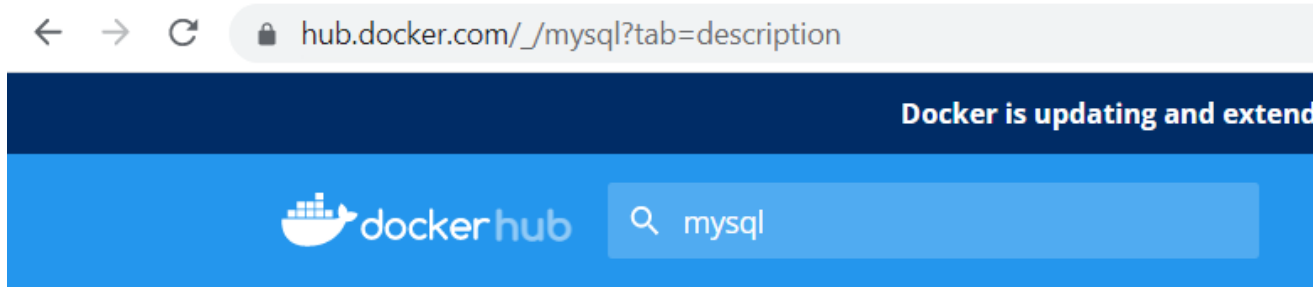
---

```
C:\WINDOWS\system32\cmd.exe  
C:\>docker run hello-world
```

```
Select C:\WINDOWS\system32\cmd.exe  
C:\>docker run hello-world  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world  
2db29710123e: Pull complete  
Digest: sha256:cc15c5b292d8525effc0f89cb299f1804f3a725c8d05e158653a563f15e4f685  
Status: Downloaded newer image for hello-world:latest  
  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
   (amd64)  
3. The Docker daemon created a new container from that image which runs the  
   executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent it  
   to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/
```



# Baixar e executar uma imagem



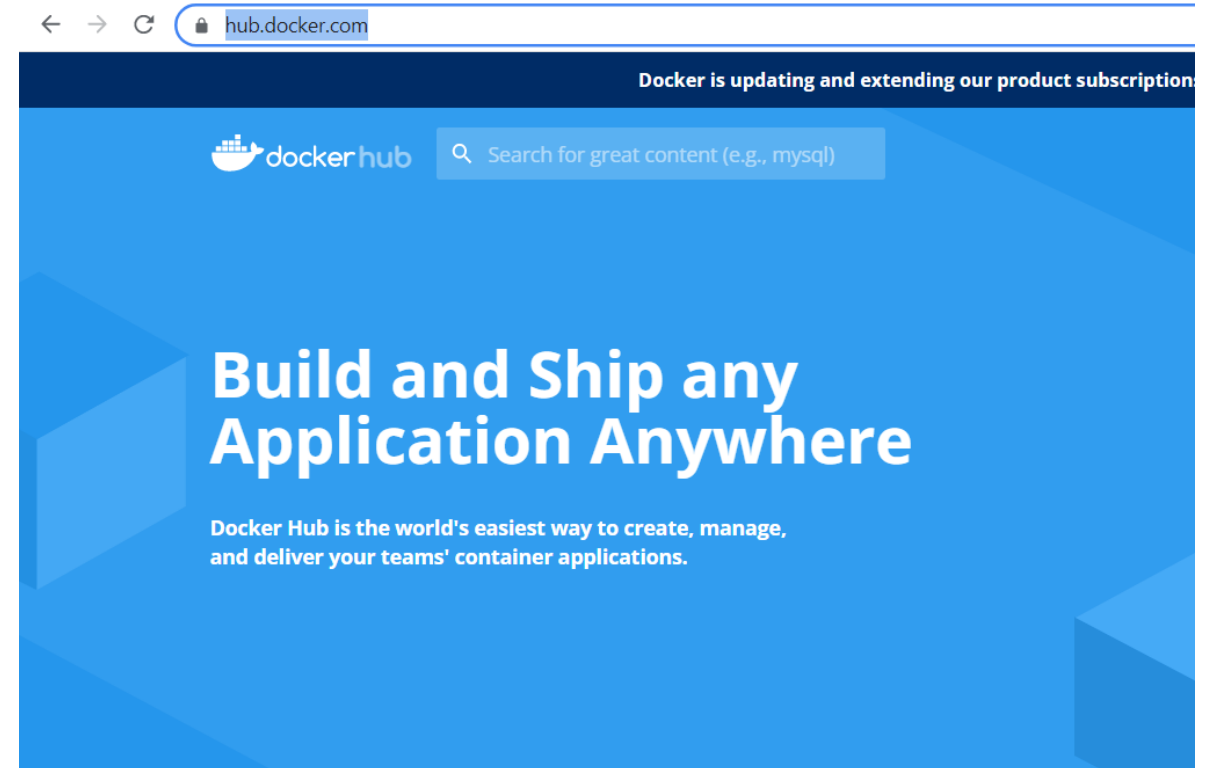
Description			
Reviews			
Tags			
Sort by Newest Filter Tags			
TAG			
latest			
Last pushed 10 hours ago by doijanky			
DIGEST			
1ea233722275			
OS/ARCH			
linux/amd64			
COMPRESSION SIZE			
144.43 MB			
docker pull mysql:latest			
TAG			
8.0.27			
Last pushed 10 hours ago by doijanky			
DIGEST			
1ea233722275			
OS/ARCH			
linux/amd64			
COMPRESSION SIZE			
144.43 MB			
docker pull mysql:8.0.27			

# Contêineres – Repositórios

---

Onde são instalados os contêineres?

- Repositórios privados
- Repositórios públicos
  - Ex: DockerHub
    - <https://hub.docker.com/>



# Docker Containers – Docker files

---



```
# Start with the Ubuntu latest image  
FROM ubuntu:latest
```

```
# Output hello world message  
CMD echo "Hello world!"
```

```
# Start with open JDK version 8 image  
FROM openjdk:8
```

```
# Copy the .jar file that contains your  
code from your system to the container  
COPY /hello.jar /usr/src/hello.jar
```

```
# Call Java to run your code  
CMD java -cp /usr/src/hello.jar  
Org.example.App
```



# Docker Containers – Docker files



```
# Start with CentOS 7 image
FROM centos:7

# Update the OS and install Apache
RUN yum -y update && yum -y install httpd

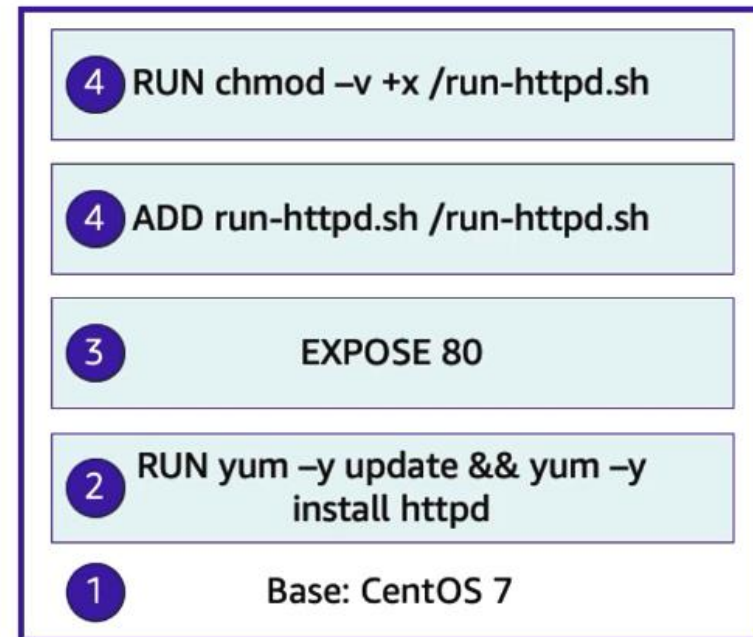
# Expose port 80—the port that the web server
"listens to"
EXPOSE Port 80

# Copy shell script and give it run permissions
ADD run-httpd.sh /run-httpd.sh
RUN chmod -v +x /run-httpd.sh

# Run shell script
CMD ["/run-httpd.sh"]
```

Subir um servidor web apache

Image layers (read-only)



# Docker CLI Comandos



Command	Description
<code>docker build</code>	Build an image from a Dockerfile.
<code>docker images</code>	List images on the Docker host.
<code>docker run</code>	Launch a container from an image.
<code>docker ps</code>	List the running containers.
<code>docker stop</code>	Stop a running container.
<code>docker start</code>	Start a container.
<code>docker push</code>	Push the image to a registry.
<code>docker tag</code>	Tag an image.

Command	Description
<code>docker logs</code>	View container log output.
<code>docker port</code>	List container port mappings.
<code>docker inspect</code>	Inspect container information.
<code>docker exec</code>	Run a command in a container.
<code>docker rm</code>	Remove one or more containers.
<code>docker rmi</code>	Remove one or more images from the host.
<code>docker update</code>	Dynamically update the container configuration.
<code>docker commit</code>	Create a new image from a container's changes.

# Docker CLI Comandos

---



```
docker build --tag node_app .
```

## Example output

```
Sending build context to Docker daemon 9.007MB
```

```
Step 1/7 : FROM node:11-alpine
```

```
11-alpine: Pulling from library/node
```

```
...
```

```
Successfully built a5886f101e12
```

```
Successfully tagged node_app:latest
```



# Docker CLI Comandos



```
docker images
```

## Example output

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
<none>	node_app:latest	a5886f101e12	18 seconds ago	82.7MB

# Docker CLI Comandos



## Docker command

```
docker run -d --name node_app_1 -p 8000:80 node_app
```

## Example output

```
5ed1ea04bcb58194100f71b2e7cd0aecab182313692ed833a6a700664994785f
```

```
docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
5ed1ea04bcb5	node_app	"docker- entrypoint.s ..."	9 seconds ago	Up 7 seconds	0.0.0.0:8000- >80/tcp

# Docker CLI Comandos



```
docker exec -it node_app_1 sh
```

## Example output

```
/usr/src/app #
```

```
/usr/src/app # ls
```

Dockerfile	README.md	app	
index.js	network.template	node_modules	
package-lock.json	package.json	public	views

```
/usr/src/app # exit
```



# Docker CLI Comandos

---

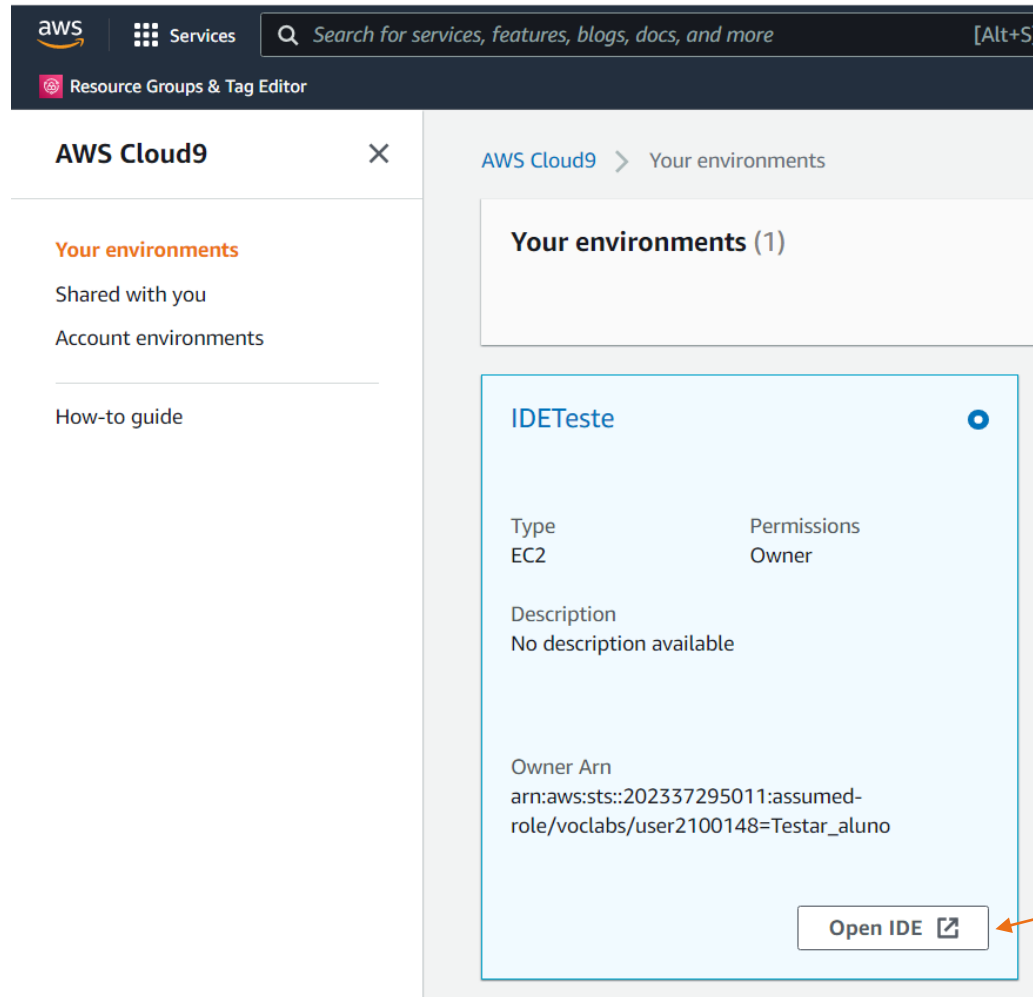


```
docker stop node_app_1 && docker rm node_app_1
```

Example output

```
node_app_1  
node_app_1
```

# Cloud9 - IDE

A screenshot of the AWS Cloud9 console. The top navigation bar includes the AWS logo, a "Services" menu, and a search bar. Below this is a "Resource Groups & Tag Editor" section. The main content area is titled "AWS Cloud9" and shows "Your environments (1)". A single environment named "IDETeste" is listed with a status of "Owner". The environment details include its type (EC2), a description (No description available), and the owner's ARN. An "Open IDE" button with an external link icon is at the bottom right of the environment card, highlighted by an orange arrow.

aws Services Search for services, features, blogs, docs, and more [Alt+S]

Resource Groups & Tag Editor

**AWS Cloud9** ✕

**Your environments**

Shared with you

Account environments

How-to guide

**AWS Cloud9** > Your environments

**Your environments (1)**

**IDETeste**

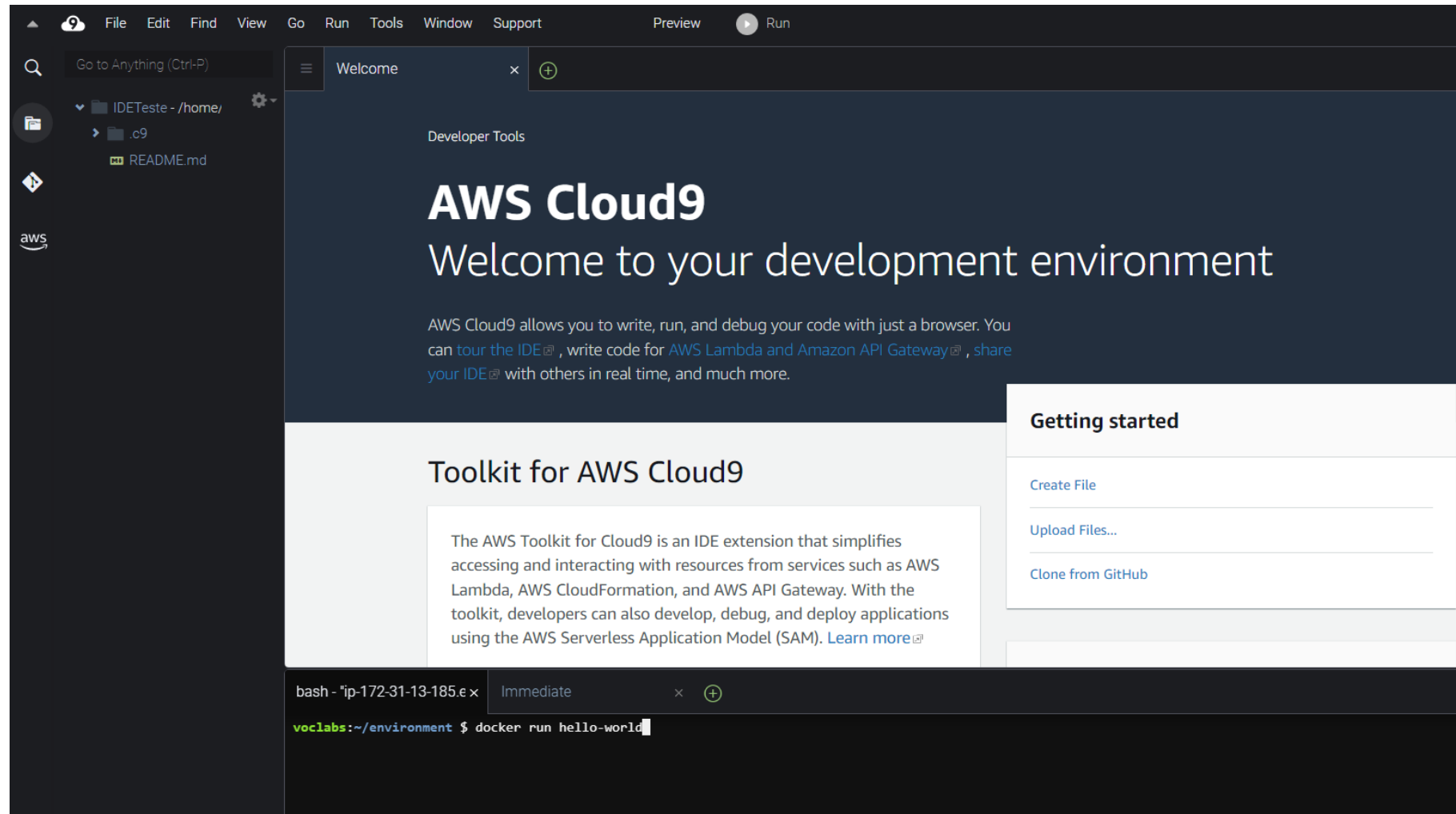
Type EC2 Permissions Owner

Description No description available

Owner Arn  
arn:aws:sts::202337295011:assumed-role/voclabs/user2100148=Testar\_aluno

Open IDE ↗

# Cloud9 - IDE





# Cloud9 - IDE

---



```
voclabs:~/environment/containers $ ls
Dockerfile
voclabs:~/environment/containers $ vim Dockerfile
```

```
vim - "ip-10-16-10-4" x
```

```
FROM mysql:8.0.23
EXPOSE 3306
```

# Cloud9 – Subir imagem para o ECR



Fazer o build de uma imagem de acordo com o Dockerfile na pasta atual

```
voclabs:~/environment/containers $ docker build -t ftf-meu-ecr .
```

“Taggear” a imagem

```
voclabs:~/environment/containers $ docker tag ftf-meu-ecr:latest 095180850434.dkr.ecr.us-east-1.amazonaws.com/ftf-meu-ecr:latest
```

Fazer o push da imagem tagueada

```
voclabs:~/environment/containers $ docker push 095180850434.dkr.ecr.us-east-1.amazonaws.com/ftf-meu-ecr:latest
```

# Cloud9 – Executar imagem



```
bash ip 10.10.10.4 x immediate
00e20000901c: Pull complete
e3a5e171c2f8: Pull complete
c2cedd8aa061: Pull complete
d6a485af4cc9: Pull complete
ee16a57baf60: Pull complete
64bab9180d2a: Pull complete
c3aceb7e4f48: Pull complete
269002e5cf58: Pull complete
d5abeb1bd18e: Pull complete
cbd79da5fab6: Pull complete
Digest: sha256:cdf3b62d78d1bbb1d2bd6716895a84014e00716177cbb7e90f6c6a37a21dc796
Status: Downloaded newer image for mysql:latest
fcb7e059795e40570de714ed3e46fe750a2b20b2e5bc673cb7a0a506ef9a509a
voclabs:~/environment/containers $ docker run --name meuservermysql -e MYSQL_ROOT_PASSWORD=1234 -d mysql:latest
```

```
docker run --name meuservermysql -e MYSQL_ROOT_PASSWORD=1234 -d mysql:latest
```

# Cloud9 – Acessar o bash da imagem



```
voclabs:~/environment/containers $ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
fcb7e059795e   mysql:latest   "docker-entrypoint.s..." About a minute ago Up About a minute   3306/tcp, 33060/tcp     meuservermysql
voclabs:~/environment/containers $ docker exec -it fcb bash
bash-4.4# ls
bin boot dev docker-entrypoint-initdb.d entrypoint.sh etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
bash-4.4#
```

# Cloud9 - IDE

---





# AWS ECS

---

- Orquestração de contêineres
- Serviço gerenciado que escala automaticamente

# AWS ECR

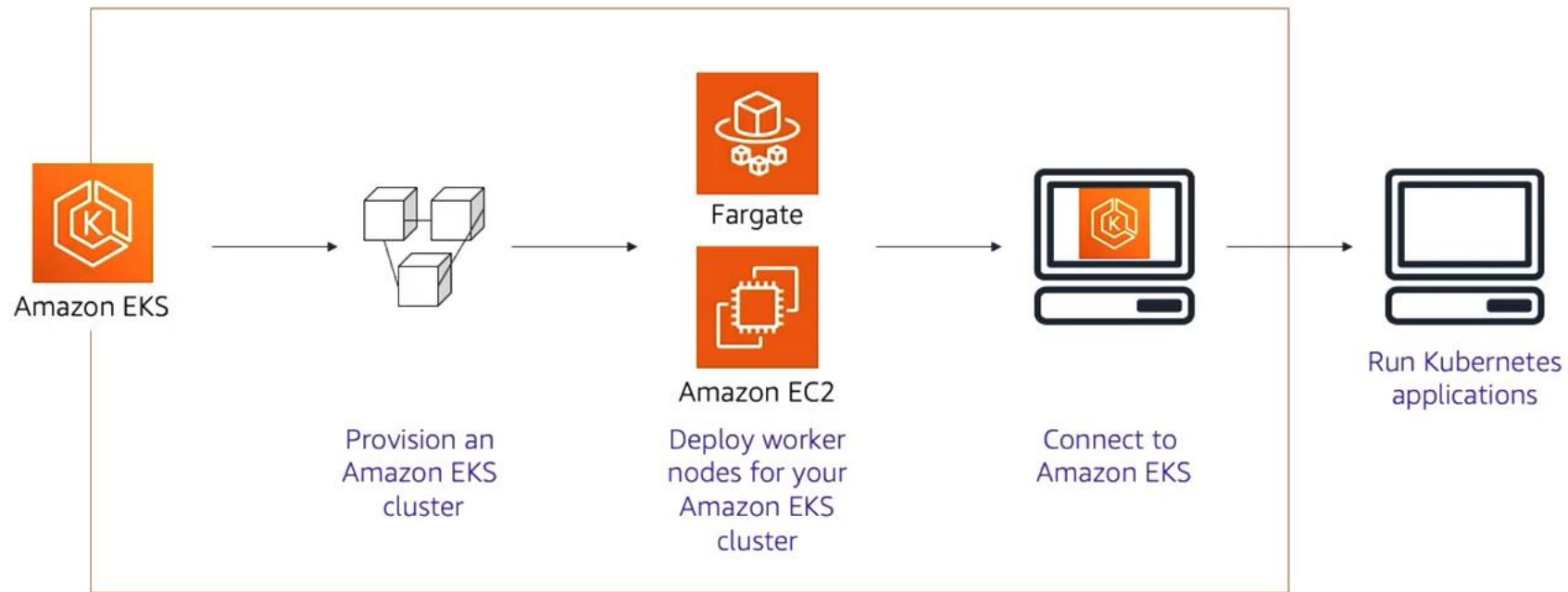
---

- Armazenar imagens de contêineres na AWS
- Integrado com o CLI do docker

# AWS EKS

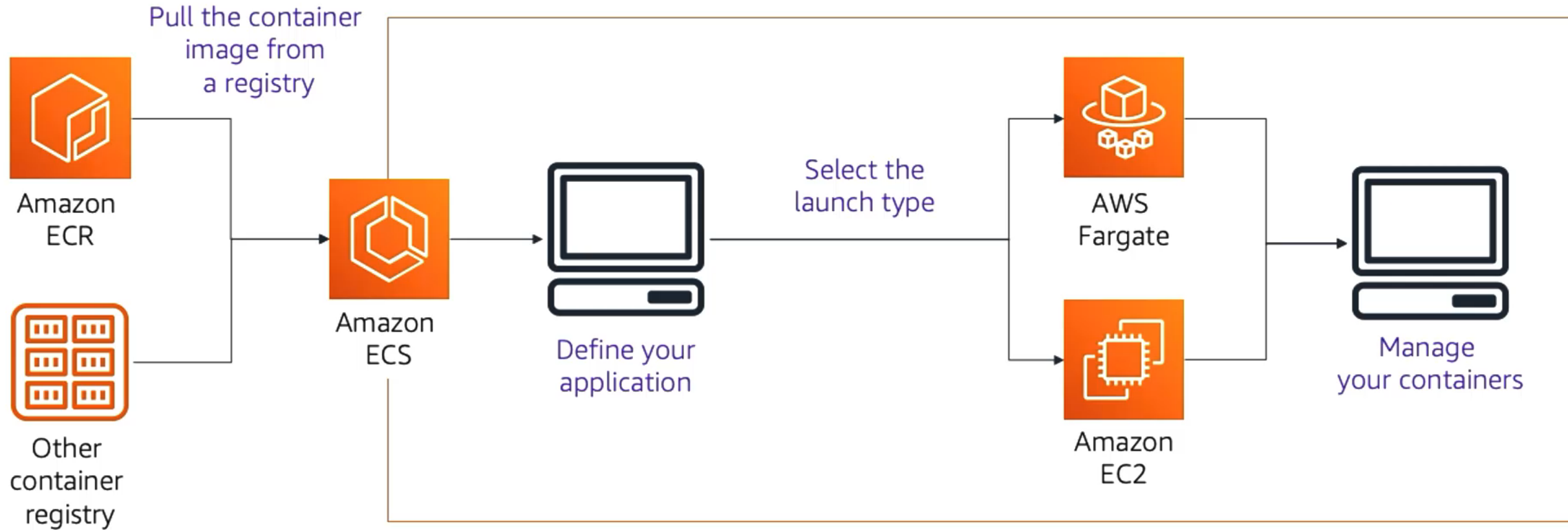
---

- Suporte a Kubernetes



# AWS ECR

---



# AWS ECR

---

```
# Create a repository called hello-world
> aws ecr create-repository \
    --repository-name hello-world \
    --region us-east-1

# Build and tag an image
> docker build -t hello-world .
> docker tag hello-world:latest aws_account_id.dkr.ecr.us-east-1.amazonaws.com/hello-world:latest

# Authenticate Docker to your Amazon ECR registry
# You can skip the `docker login` step if you have amazon-ecr-credential-helper set up
> aws ecr get-login-password --region region | docker login --username AWS --password-stdin
aws_account_id.dkr.ecr.region.amazonaws.com

# Push an image to your repository
> docker push aws_account_id.dkr.ecr.us-east-1.amazonaws.com/hello-world:latest
```



# AWS ECR

## Amazon Elastic Container Registry



Private registry

Public registry

**Repositories**

Getting started

Documentation

Public gallery

Amazon ECR > Repositórios

**Private**

Public

### Repositórios privados



Visualizar comandos push

Excluir

Actions ▼

**Criar repositório**

🔍 Localizar repositórios

< 1 > ⚙️

Nome do  
repositório ▲

URI

Criado  
em ▼

Imutabilidade de  
etiquetas

Frequência de  
verificação

Tipo de  
criptografia

Cache de  
espelhamento

**Nenhum repositório**

Nenhum repositório foi encontrado

# AWS ECR

---

## Configurações gerais

### Configurações de visibilidade [Info](#)

Escolha a configuração de visibilidade para o repositório.

- ☒ Privado  
O acesso é gerenciado pelas permissões da política do IAM e do repositório.
- ☐ Público  
Visível publicamente e acessível para extrações de imagens.






### Nome do repositório

Forneça um nome conciso. Um desenvolvedor deve ser capaz de identificar o conteúdo do repositório pelo nome.

202337295011.dkr.ecr.us-east-1.amazonaws.com/

11 de no máximo 256 caracteres (mínimo de 2). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, periods and forward slashes.

# AWS ECR

Repositórios privados (1)					Visualizar comandos push	Excluir	Actions ▼	Criar repositório
<input type="text" value="Localizar repositórios"/>				 1  				
	Nome do repositório ▲	URI	Criado em ▼	Imutabilidade de etiquetas	Frequência de verificação	Tipo de criptografia	Cache de espelhamento	
<input type="radio"/>	ftf-meu-ecr	 202337295011.dkr.ecr.us-east-1.amazonaws.com/ftf-meu-ecr	08 de setembro de 2022, 09:32:14 (UTC-03)	Desabilitado	Manual	AES-256	Inativo	

# AWS ECR

---

macOS / Linux

Windows

Verifique se você tem a versão mais recente do AWS CLI e do Docker instalada. Para obter mais informações, consulte [Primeiros passos com o Amazon ECR](#).

Use as etapas a seguir para autenticar e enviar uma imagem para o repositório. Para obter outros métodos de autenticação do registro, incluindo o auxiliar de credenciais do Amazon ECR, consulte [Autenticação do registro](#).

1. Recupere um token de autenticação e autentique seu cliente Docker em seu registro.

Use o AWS CLI:

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 202337295011.dkr.ecr.us-east-1.amazonaws.com
```

Observação: se você receber um erro ao usar o AWS CLI, verifique se você tem a versão mais recente do AWS CLI e do Docker instalada.

2. Crie sua imagem do Docker usando o comando a seguir. Para obter informações sobre como criar um arquivo Docker do zero, consulte as instruções [aqui](#). Você pode pular esta etapa se sua imagem já estiver construída:

```
docker build -t ftf-meu-ecr .
```

3. Depois que a compilação for concluída, marque sua imagem para que você possa enviá-la para este repositório:

```
docker tag ftf-meu-ecr:latest 202337295011.dkr.ecr.us-east-1.amazonaws.com/ftf-meu-ecr:latest
```

4. Execute o seguinte comando para enviar essa imagem para o repositório da AWS recém-criado:

```
docker push 202337295011.dkr.ecr.us-east-1.amazonaws.com/ftf-meu-ecr:latest
```

# AWS ECR

## Amazon Elastic Container Registry



Private registry

Public registry

Repositories

Summary

**Images**

Permissions

Lifecycle Policy

Tags

Getting started

Documentation

Public gallery

Amazon ECR > Repositórios > ftf-meu-ecr

## ftf-meu-ecr

Visualizar comandos push

Editar

### Imagens (1)



Excluir

Verificar

Encontrar imagens

< 1 >

<input type="checkbox"/>	Etiqueta de imagem	Tipo de artefato	Enviado por push em ▼	Tamanho (MB) ▼	URI de imagem	Resumo	Status da verificação	Vulnerabi
<input type="checkbox"/>	latest	Image	08 de setembro de 2022, 17:15:59 (UTC-03)	159.33	Copiar URI	sha256:08ed803c5f0a35...	-	-



# Tarefa – Configurar Cloud9 e rodar um container

- ☐ Configure um ambiente do Cloud9

  - ☐ Obs: Demora uns 5 min

- ☐ Rode um container hello-world

- ☐ **Encerrar o LAB**

# Obrigado!

Contato: [proffernando.fernandes@fiap.com.br](mailto:proffernando.fernandes@fiap.com.br)