

Infraestrutura para Sistemas de Software

Prof. Dr. Carlos Alberto da Silva



Módulo 5 - Arquitetura de micros serviços

Unidade 2 - Exemplos de provedores e seus serviços



Mapa da aula

Conhecer os provedores de serviços na nuvem

- *Container* como um Serviço (CaaS)
- Provedores de CaaS
- Testar uma plataforma *Docker*®
- *PodMan*®
- *E-books* sobre *Docker*®

***Container* como um Serviço (CaaS)**

Container como um Serviço (CaaS)

CaaS

- Deve ser uma plataforma de hospedagem baseada em nuvem, perfeita para gerenciar seus *containers Docker*®.
- Deve oferecer escalabilidade e confiabilidade robustas para qualquer tamanho de projeto.

Container como um Serviço (CaaS)

CaaS

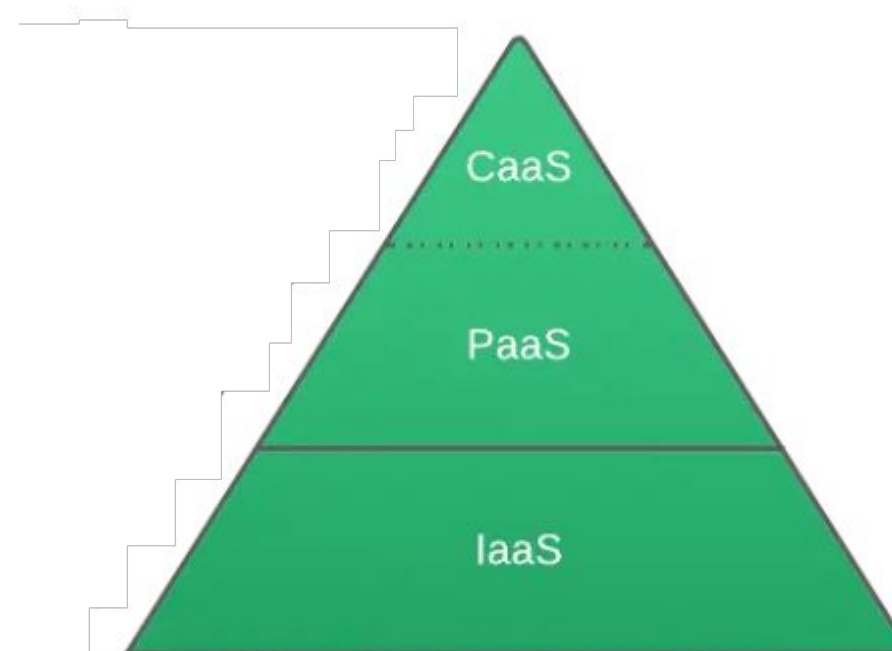
Com recursos avançados de gerenciamento como:

- implantação automatizada;
- funcionalidade de auto recuperação;
- e opções de escalonamento personalizadas.

Container como um Serviço (CaaS)

CaaS

Utiliza os IaaS e PaaS do mesmo provedor



Montagem: Carlos Alberto da Silva, 2023.

Provedores de CaaS

Provedores de CaaS

Você pode escolher entre uma variedade de sistemas operacionais e configurações de hardware:

- *Amazon Web Services (AWS): a AWS oferece o serviço EC2 (Elastic Compute Cloud)*
- *Google Cloud*
- *Microsoft Azure*
- *IBM Cloud*
- *Oracle Cloud*

Provedores de CaaS

Como acessar:

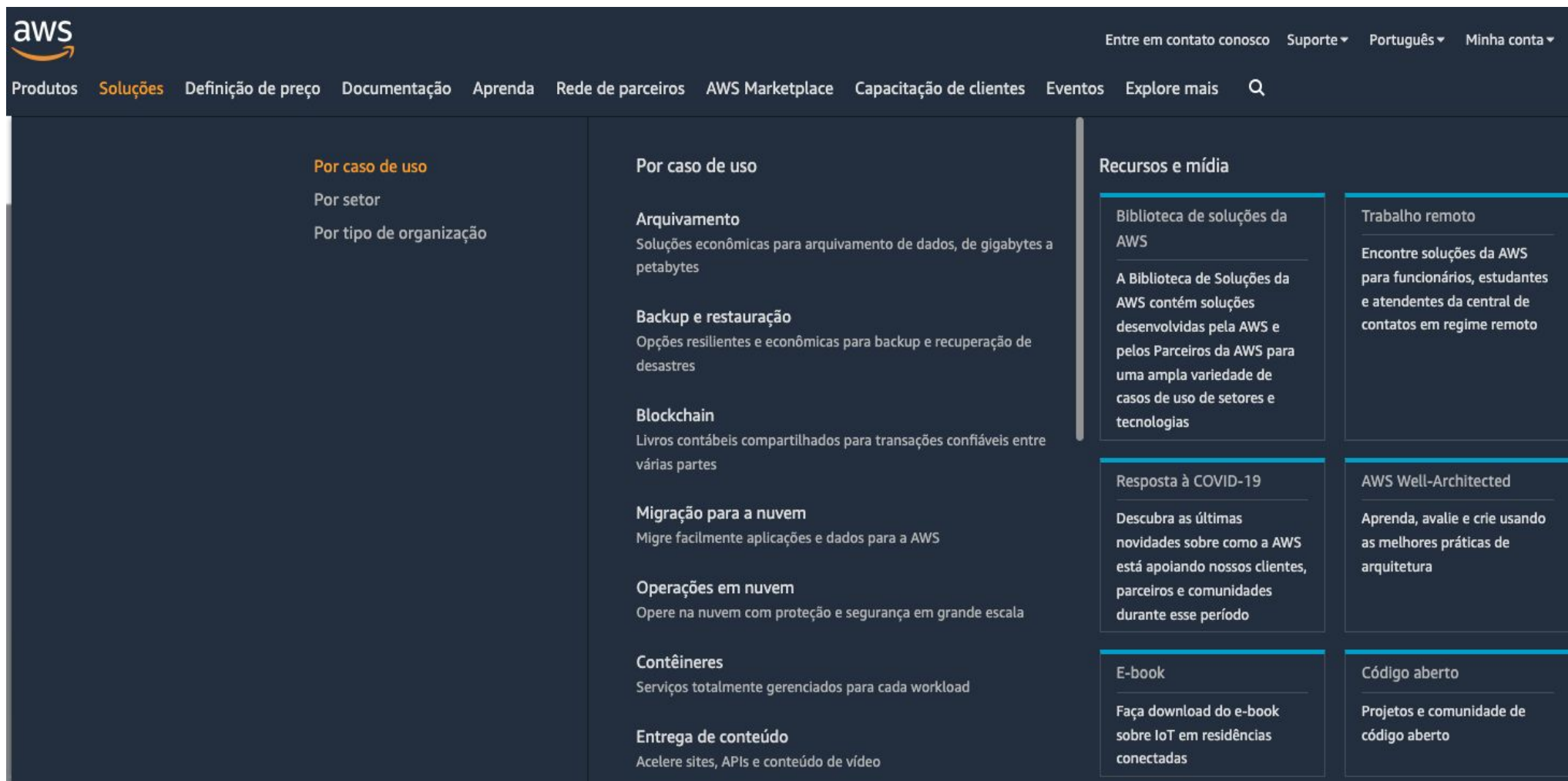
- *Amazon Web Services (AWS):* <https://aws.amazon.com/pt>
- *Google Cloud:* <https://cloud.google.com/>
- *Microsoft Azure:* <https://azure.microsoft.com/pt-br/>
- *IBM Cloud:* <https://cloud.ibm.com/login>
- *Oracle Cloud:* <https://www.oracle.com/br/cloud/>

Provedores de CaaS

Símbolos:



Provedores de CaaS



The screenshot shows the AWS website's 'Soluções' (Solutions) page. The header includes the AWS logo and navigation links: 'Produtos', 'Soluções' (highlighted), 'Definição de preço', 'Documentação', 'Aprenda', 'Rede de parceiros', 'AWS Marketplace', 'Capacitação de clientes', 'Eventos', 'Explore mais', and a search icon. The main content is organized into three columns. The left column, titled 'Por caso de uso' (By use case), lists 'Por setor' and 'Por tipo de organização'. The middle column, also titled 'Por caso de uso', lists several use cases: 'Arquivamento' (Archiving), 'Backup e restauração' (Backup and restoration), 'Blockchain', 'Migração para a nuvem' (Migration to the cloud), 'Operações em nuvem' (Cloud operations), 'Contêineres' (Containers), and 'Entrega de conteúdo' (Content delivery). The right column, titled 'Recursos e mídia' (Resources and media), lists 'Biblioteca de soluções da AWS' (AWS Solutions Library), 'Trabalho remoto' (Remote work), 'Resposta à COVID-19' (COVID-19 response), 'AWS Well-Architected' (AWS Well-Architected), 'E-book' (E-book), and 'Código aberto' (Open source). Each use case or resource has a brief description below its title.

aws

Entre em contato conosco Suporte ▼ Português ▼ Minha conta ▼

Produtos **Soluções** Definição de preço Documentação Aprenda Rede de parceiros AWS Marketplace Capacitação de clientes Eventos Explore mais 🔍

Por caso de uso

Por setor

Por tipo de organização

Por caso de uso

Arquivamento
Soluções econômicas para arquivamento de dados, de gigabytes a petabytes

Backup e restauração
Opções resilientes e econômicas para backup e recuperação de desastres

Blockchain
Livros contábeis compartilhados para transações confiáveis entre várias partes

Migração para a nuvem
Migre facilmente aplicações e dados para a AWS

Operações em nuvem
Opere na nuvem com proteção e segurança em grande escala

Contêineres
Serviços totalmente gerenciados para cada workload

Entrega de conteúdo
Acelere sites, APIs e conteúdo de vídeo

Recursos e mídia

Biblioteca de soluções da AWS
A Biblioteca de Soluções da AWS contém soluções desenvolvidas pela AWS e pelos Parceiros da AWS para uma ampla variedade de casos de uso de setores e tecnologias

Trabalho remoto
Encontre soluções da AWS para funcionários, estudantes e atendentes da central de contatos em regime remoto

Resposta à COVID-19
Descubra as últimas novidades sobre como a AWS está apoiando nossos clientes, parceiros e comunidades durante esse período

AWS Well-Architected
Aprenda, avalie e crie usando as melhores práticas de arquitetura

E-book
Faça download do e-book sobre IoT em residências conectadas

Código aberto
Projetos e comunidade de código aberto

Fonte: Amazon AWS, 2023.

Provedores de CaaS

Como tratar este serviço de CaaS

Passos para contratar um serviço na nuvem (CaaS):

- 1) Escolher o provedor;
- 2) Criar uma conta de acesso;
- 3) **Registrar um cartão de crédito** na sua conta;
- 4) Escolher o serviço e finalizar o contrato.

Testar uma plataforma *Docker*[®]

Comandos *Docker*®

Comandos para	Gerenciamento de
<i>builder</i>	<i>builds</i>
<i>config</i>	<i>Docker configs</i>
<i>container</i>	<i>contêineres</i>
<i>context</i>	<i>contextos</i>
<i>engine</i>	<i>docker engine</i>
<i>network</i>	<i>networks</i>
<i>node</i>	<i>Swarm nodes</i>

<i>plugin</i>	<i>plugins</i>
<i>secret</i>	<i>Docker secrets</i>
<i>service</i>	<i>serviços</i>
<i>stack</i>	<i>Docker stacks</i>
<i>swarm</i>	<i>swarm</i>
<i>system</i>	<i>system</i>
<i>trust</i>	<i>trust on Docker images</i>
<i>volume</i>	<i>volumes</i>

Comandos *Docker*[®]

Sintaxe do comando em *Docker*[®]

Docker (comando global) (comando específico) (parâmetro/s)

⇒ Por exemplo: Listar todos os *Containers* ativos e inativos
docker container ls -a

Comandos *Docker*[®]

Comandos gerais

- *docker version* (no cliente ou servidor);
- *docker login* (no *Docker Hub*[®]);
- *docker logout* (no *Docker Hub*[®]);
- *docker system prune* - eliminar os *containers*:
 - parados (*stopped*);
 - de redes inativas;
 - de imagens de teste.

Comandos para *containers*

- *create* – cria novo *container* baseado em uma imagem;
- *start* – inicializa um *container* existente;
- *run* – cria um *container* e o inicia;
- *ls* – lista todos os *containers* ativos;
- *inspect* – fornece informações mais completas sobre um *container*;
- *logs* – lista os *logs* de um *container* específico.

Comandos *Docker*[®]

Comandos para *containers*

- *stop* – interrompe um *container* ativo.
- *rm* – remove um *container* parado.

⇒ Por exemplo: Inspecionar um *container* com:
docker container inspect (*nome do container*)

Comandos *Docker*[®]

Comandos para *Imagem*

- *build* – cria uma *imagem* baseada em um *Dockerfile*[®];
- *push* – faz *upload* de uma *imagem* para o repositório;
- *tag* – permite atribuir uma *tag* à *imagem*;
- *ls* – lista todas as *imagens* baixadas atualmente;
- *inspect* – obtém mais informações de uma *imagem*;
- *rm* – remove uma *imagem* do sistema.

⇒ Por exemplo: ***docker rm*** (*nome da imagem*)

Comandos *Docker*®

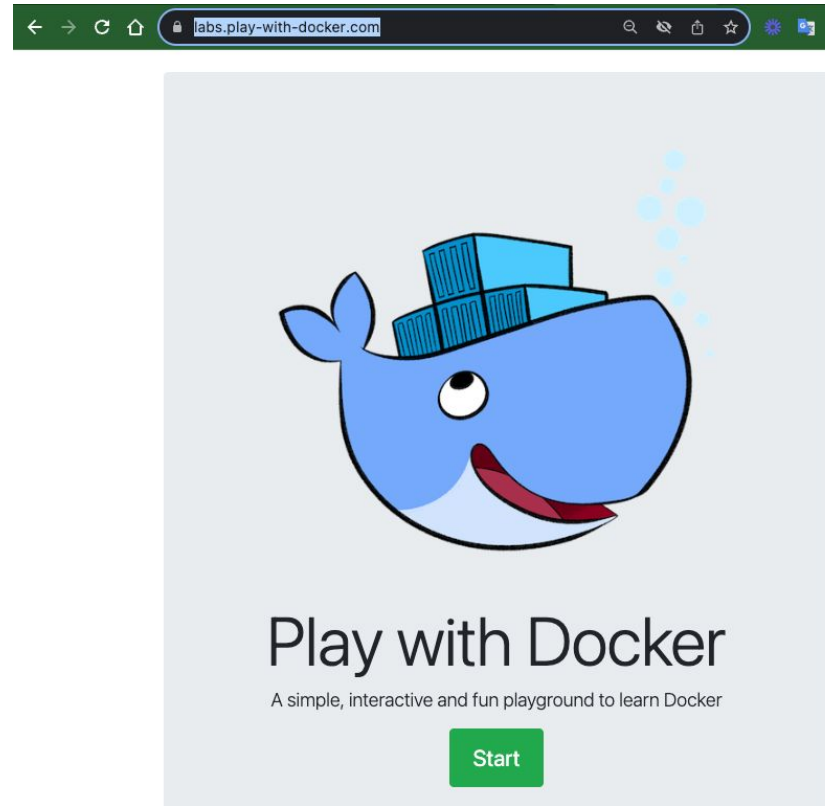
Comandos para redes e volumes

- *ls* – lista todas as redes ou volumes
- *rm* – remove uma rede ou volume
- *create* – cria nova rede ou volume
- *inspect* – fornece informações mais completas sobre uma rede ou volume
- *prune* - eliminar uma rede ou volume

⇒ Por exemplo: ***docker network create*** (*nome da rede*)

Testar uma plataforma *Docker*®

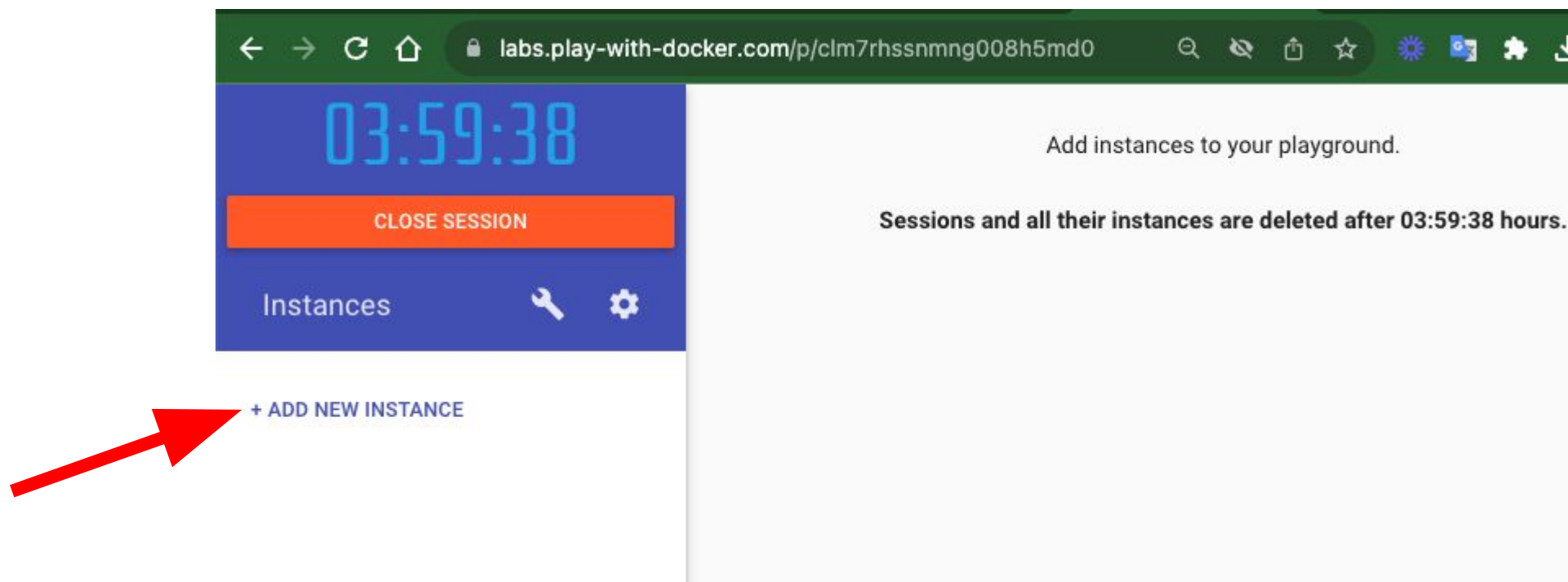
Site <https://labs.play-with-docker.com/>



Fonte: Labs.play-with-docker, 2023.

Testar uma plataforma *Docker*®

Site <https://labs.play-with-docker.com/>



Testar uma plataforma *Docker*®

Site <https://labs.play-with-docker.com/>

The screenshot displays the Labs.play-with-docker.com interface. On the left, a sidebar shows a digital clock at 03:58:47, a 'CLOSE SESSION' button, and an 'Instances' section with a '+ ADD NEW INSTANCE' button. Below this, a list of instances shows '192.168.0.28' and 'node1', which is highlighted by a red arrow. The main panel on the right shows details for the selected instance 'clm7rhss_clm7rvssnmng008h5mdg', including its IP (192.168.0.28), memory usage (0.97%), CPU usage (0.05%), and an SSH command. At the bottom, a terminal window shows a warning message and the prompt '[node1] (local) root@192.168.0.28 ~'.

Fonte: Labs.play-with-docker, 2023.

Testar uma plataforma *Docker*®

Site <https://labs.play-with-docker.com/>

Digite o comando:
docker version

```
[root@localhost ~]# docker version
Client:
 Version:           24.0.7
 API version:       1.43
 Go version:        go1.20.10
 Git commit:        afdd53b
 Built:             Thu Oct 26 09:04:00 2023
 OS/Arch:           linux/amd64
 Context:           default

Server: Docker Engine - Community
 Engine:
  Version:          24.0.7
  API version:      1.43 (minimum version 1.12)
  Go version:       go1.20.10
  Git commit:       311b9ff
  Built:            Thu Oct 26 09:05:28 2023
  OS/Arch:          linux/amd64
  Experimental:     true
 containerd:
  Version:          v1.7.6
  GitCommit:        091922f03c2762540fd057fba91260237ff86acb
 runc:
  Version:          1.1.9
  GitCommit:        v1.1.9-0-gccaecfc
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

Fonte: Labs.play-with-docker, 2023.

Testar uma plataforma *Docker*®

Site <https://labs.play-with-docker.com/>

Digite o comando:

docker container ls -a

```
[node1] (local) root@192.168.0.28 ~  
$ docker container ls -a  
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES  
[node1] (local) root@192.168.0.28 ~  
$
```

Testar uma plataforma *Docker®*

Site <https://labs.play-with-docker.com/>
Criar outro *container* (instância)

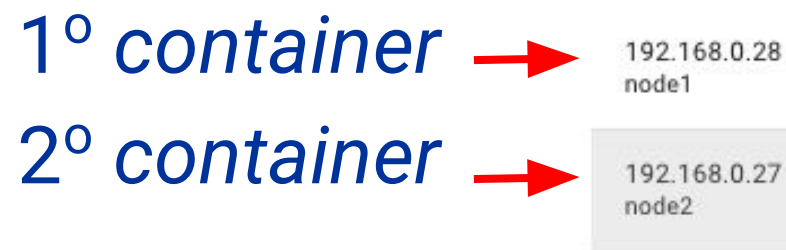
The screenshot displays the Labs.play-with-docker.com interface. On the left, a sidebar shows a list of instances: '192.168.0.28 node1' and '192.168.0.27 node2'. A red arrow points to the '+ ADD NEW INSTANCE' button. The main panel shows the details for the selected instance 'clm7rhss_clm7tt6fml8g0088st80'. It includes the IP address '192.168.0.27', memory usage '1.02% (40.68MiB / 3.906GiB)', and CPU usage '1.03%'. There are buttons for 'CLOSE SESSION', 'OPEN PORT', 'DELETE', and 'EDITOR'. The SSH command is 'ssh ip172-18-0-8-clm7rhssnmng008h5md0@direct.labs.p'. The terminal output shows a warning message: 'WARNING!!!! This is a sandbox environment. Using personal credentials is HIGHLY! discouraged. Any consequences of doing so are completely the user's responsibilities. The PWD team.'

Testar uma plataforma *Docker*®

Site <https://labs.play-with-docker.com/>

Temos 2 *containers*:

com números IP: **192.168.0.28** e **192.168.0.27**.



Testar uma plataforma *Docker*®

Site <https://labs.play-with-docker.com/>

Digitar o comando no 1º **container** (192.168.0.28):
curl -I 192.168.0.27

Falha de conexão com o servidor Web no IP: 192.168.0.27

```
[node1] (local) root@192.168.0.28 ~  
$ curl -I 192.168.0.27  
curl: (7) Failed to connect to 192.168.0.27 port 80 after 4 ms: Couldn't connect to server  
[node1] (local) root@192.168.0.28 ~  
$
```

Testar uma plataforma *Docker*®

Site <https://labs.play-with-docker.com/>

Digitar o comando no **2º container** (192.168.0.27):
docker pull nginx

```
[node2] (local) root@192.168.0.27 -  
$ docker pull nginx  
Using default tag: latest  
latest: Pulling from library/nginx  
1f7ce2fa46ab: Pull complete  
9b16c94bb686: Pull complete  
9a59d19f9c5b: Pull complete  
9ea27b074f71: Pull complete  
c6edf33e2524: Pull complete  
84b1ff10387b: Pull complete  
517357831967: Pull complete  
Digest: sha256:10d1f5b58f74683ad34eb29287e07dable90f10af243f151bb50aa5dbb4d62ee  
Status: Downloaded newer image for nginx:latest  
docker.io/library/nginx:latest  
[node2] (local) root@192.168.0.27 -  
$
```

Testar uma plataforma *Docker*®

Site <https://labs.play-with-docker.com/>

Digitar o comando no **2º container** (192.168.0.27):
docker run -d -p 80:80 nginx

```
[node2] (local) root@192.168.0.27 ~  
$ docker run -d -p 80:80 nginx  
295348b2a9e56bcd2df7152bf9ab765d81822233710e7494d54921158074c94  
[node2] (local) root@192.168.0.27 ~  
$
```

Fonte: Labs.play-with-docker, 2023.

Testar uma plataforma *Docker*®

Site <https://labs.play-with-docker.com/>

Digitar no 1º *container* (192.168.0.28):
curl -I 192.168.0.27

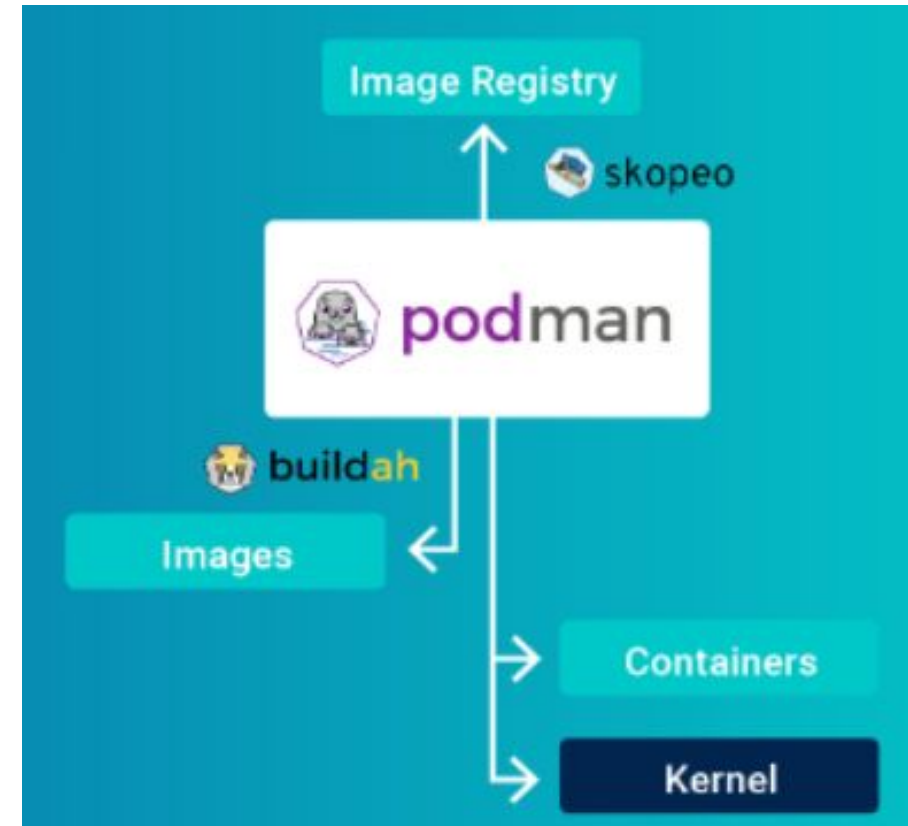
```
[node1] (local) root@192.168.0.28 ~  
$ curl -I 192.168.0.27  
HTTP/1.1 200 OK  
Server: nginx/1.25.3  
Date: Sun, 03 Dec 2023 13:23:50 GMT  
Content-Type: text/html  
Content-Length: 615  
Last-Modified: Tue, 24 Oct 2023 13:46:47 GMT  
Connection: keep-alive  
ETag: "6537cac7-267"  
Accept-Ranges: bytes  
  
[node1] (local) root@192.168.0.28 ~  
$
```


Podman[®]

Podman®

Podman® (POD MANager)

Documentação em:
<https://podman.io/>



Podman[®]

Podman[®] (POD MANager)

- Concorrente do Docker[®];
- *Podman*[®] foi criado sem *daemon*;

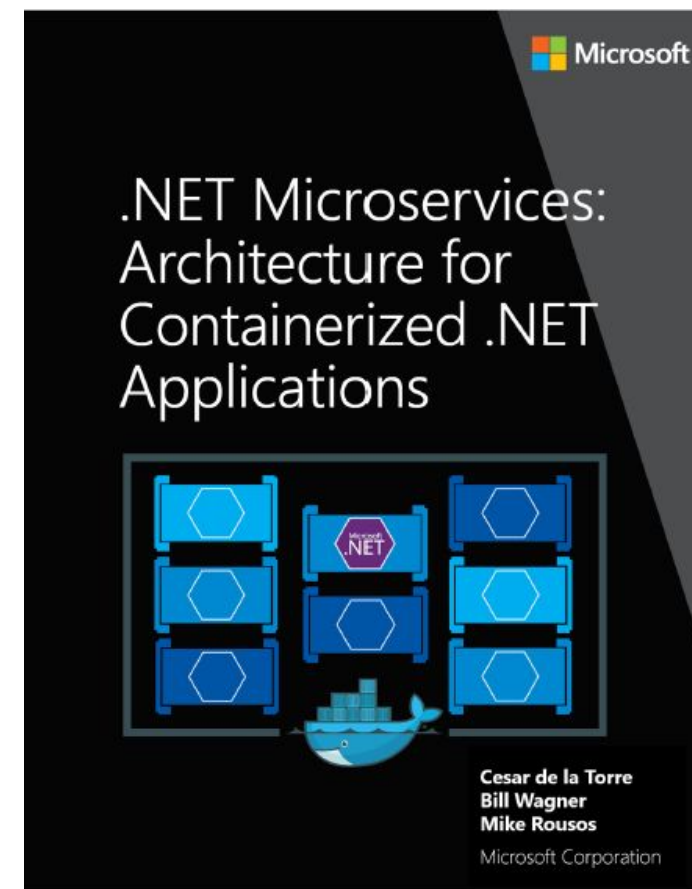
Podman® (POD MANager)

- Para desenvolver, gerenciar e executar *containers* OCI em seu sistema *Linux*;
- A biblioteca *libpod* fornece APIs para gerenciar *containers*, *pods*, imagens de *container* e volumes.

E-books sobre Docker®

E-book *Docker*®

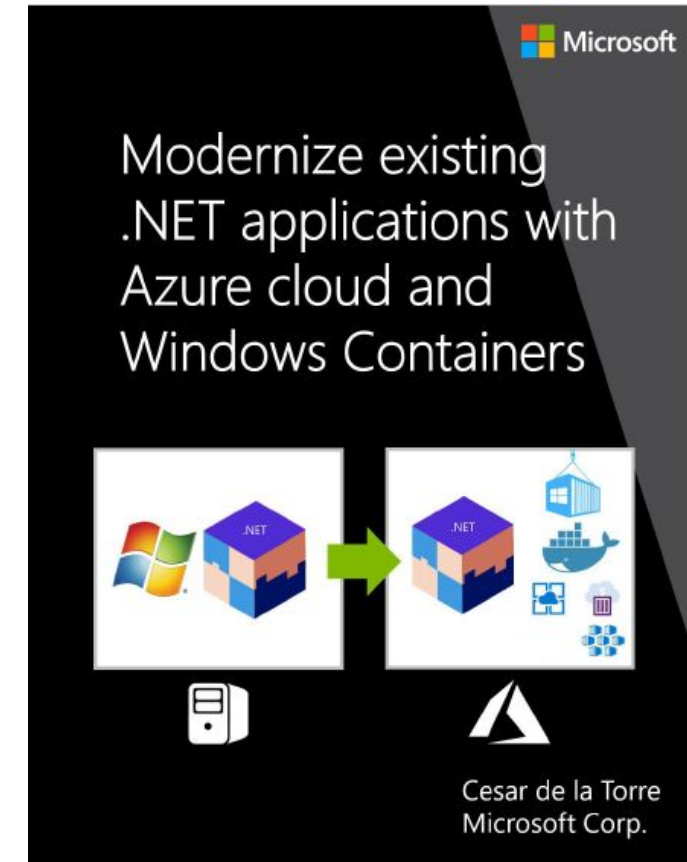
<https://dotnet.microsoft.com/pt-br/download/e-book/microservices-architecture/pdf>



Fonte: .NET microsoft Docker

E-book *Docker*®

<https://aka.ms/liftandshiftwithcontainersebook>



Fonte: .NET microsoft Docker

E-book *Docker*®

[https://pub.erudio.com.br/
guia-docker](https://pub.erudio.com.br/guia-docker)



Fonte: Docker um guia rápido, 2018.

E-book *Docker*®

https://docs.oracle.com/cd/F61410_01/docker/OL-Docker.pdf

Oracle Linux
Oracle Container Runtime for Docker User's
Guide



E87205-29
August 2022



Fonte: Oracle Linux Docker, 2022.

Referências

AMAZON AWS. [Online]. Disponível em: <https://www.aws.amazon.com/pt/>. Acessado em: 18 de novembro de 2023.

DOCKER. [Online]. Disponível em: <https://www.docker.com/>. Acessado em: 18 de novembro de 2023.

PLAY WITH DOCKER. [Online]. Disponível em: <https://labs.play-with-docker.com/>. Acessado em: 18 de novembro de 2023.

PODMAN. [Online]. Disponível em: <https://podman.io/>. Acessado em: 18 de novembro de 2023.

Licenciamento



Respeitadas as formas de citação formal de autores de acordo com as normas da ABNT NBR 6023 (2018), a não ser que esteja indicado de outra forma, todo material desta apresentação está licenciado sob uma [Licença Creative Commons - Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).