INSTITUTO FEDERAL GOIANO – *CAMPUS CERES*
BACHARELADO EM SISTEMAS DE INFORMAÇÃO
GUILHERME HENRIQUE GONTIJO ALENCAR

Como o docker pode ajudar a diminuir custos das suas implantações

Ceres, GO
2025

GUILHERME HENRIQUE GONTIJO ALENCAR

COMO O DOCKER PODE AJUDAR A DIMINUIR CUSTOS DAS SUAS IMPLANTAÇÕES

Trabalho de curso apresentado ao curso de Sistemas de Informação do Instituto Federal Goiano – Campus Ceres, como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação, sob orientação do Prof. Roitier Campos Gonçalves.

Ceres, GO
2025

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO

PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO

Com base no disposto na Lei Federal nº 9.610, de 19 de fevereiro de 1998, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano a disponibilizar gratuitamente o documento em formato digital no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

IDENTIFICAÇÃO DA PRODUÇÃO TÉCNICO-CIENTÍFICA

- ☐ Tese (doutorado)
- ☐ Dissertação (mestrado)
- ☐ Monografia (especialização)
- ☑ TCC (graduação)
- ☐ Artigo científico
- ☐ Capítulo de livro
- ☐ Livro
- ☐ Trabalho apresentado em evento
- ☐ Produto técnico e educacional - Tipo:

Nome completo do autor:
Guilherme Henrique Gontijo Alencar

Matrícula:
2019103202030407

Título do trabalho:
COMO O DOCKER PODE AJUDAR A DIMINUIR CUSTOS DAS SUAS IMPLANTAÇÕES

RESTRIÇÕES DE ACESSO AO DOCUMENTO

Documento confidencial: ☑ Não ☐ Sim, justifique:

Informe a data que poderá ser disponibilizado no RIIF Goiano: 02 /06 /2025

O documento está sujeito a registro de patente? ☐ Sim ☑ Não

O documento pode vir a ser publicado como livro? ☐ Sim ☑ Não

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O(a) referido(a) autor(a) declara:

• Que o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;

• Que obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autoria, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;

• Que cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Ceres
Local

02 /06 /2025
Data

gov.br Documento assinado digitalmente
GUILHERME HENRIQUE GONTIJO ALENCAR
Data: 02/06/2025 21:30:24-0300
Verifique em https://validar.iti.gov.br

Assinatura do autor e/ou detentor dos direitos autorais

Ciente e de acordo:

gov.br Documento assinado digitalmente
ROITIER CAMPOS GONCALVES
Data: 09/06/2025 10:55:11-0300
Verifique em https://validar.iti.gov.br

REVISTA CONTEMPORÂNEA

e-ISSN 2447-0961

Qualis CAPES 2017-2020: B1

DECLARAÇÃO

Revista Contemporânea, ISSN 2447-0961, declara para os devidos fins, que o artigo intitulado COMO O DOCKER PODE AJUDAR A DIMINUIR CUSTOS DAS SUAS IMPLANTAÇÕES de autoria de Guilherme Henrique Gontijo Alencar, foi publicado no v.5, n.5, de 2025.

A revista é on-line, e os artigos podem ser encontrados ao acessar o link:

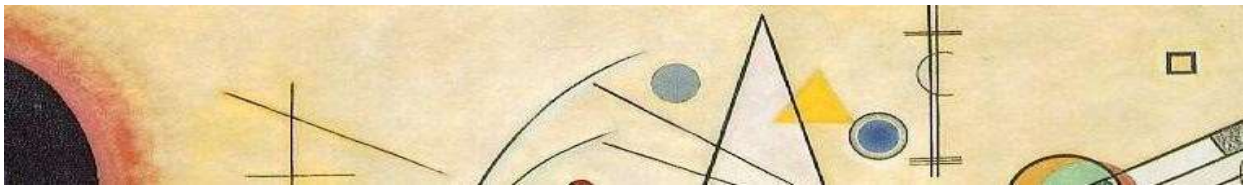
https://ojs.revistacontemporanea.com/ojs/index.php/home/issue/view/40

DOI: https://doi.org/10.56083/RCV5N5-067

Por ser a expressão da verdade, firmamos a presente declaração.

Curitiba, 3 junho 2025.

Equipe Editorial



Contemporânea
Contemporary Journal
Vol. 5 N°. 5: p. 01-12, 2025
ISSN: 2447-0961

Artigo

COMO O DOCKER PODE AJUDAR A DIMINUIR CUSTOS DAS SUAS IMPLANTAÇÕES

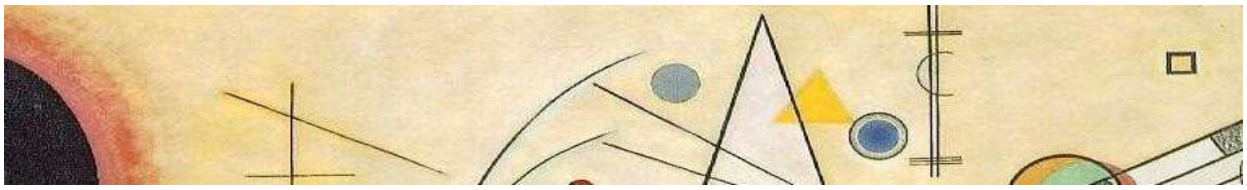
HOW DOCKER CAN HELP REDUCE DEPLOYMENT COSTS

CÓMO DOCKER PUEDE AYUDAR A REDUCIR LOS COSTOS DE TUS IMPLEMENTACIONES

DOI: 10.56083/RCV5N5-067
Receipt of originals: 4/18/2025
Acceptance for publication: 5/9/2025

Guilherme Henrique Gontijo Alencar
Graduando em Sistemas de Informação
Instituição: Instituto Federal Goiano - campus Ceres
Endereço: Ceres, Goiás, Brasil
E-mail: gabinete.ce@ifgoiano.edu.br

RESUMO: O Docker é uma ferramenta que permite criar e gerenciar contêineres de software que contém todo o ambiente necessário para que uma aplicação possa ser executada de forma consistente em diferentes sistemas operacionais. Com o uso do Docker, é possível padronizar o ambiente de desenvolvimento, garantindo que todos os desenvolvedores trabalhem com a mesma configuração e evitando problemas de compatibilidade entre diferentes versões de bibliotecas e ferramentas. Isso acelera o processo de desenvolvimento, tornando-o mais eficiente e facilitando a colaboração entre os membros da equipe. Além disso, o Docker também ajuda a simplificar o processo de implantação de uma aplicação em produção, permitindo que ela seja executada em qualquer sistema que tenha suporte ao Docker, sem precisar de configurações adicionais. Outro benefício importante do Docker é a facilidade de escalar aplicações, especialmente em ambientes de nuvem e orquestração com ferramentas como o Kubernetes. Ele também contribui para o uso eficiente de recursos do sistema, já que os contêineres compartilham o mesmo kernel do sistema operacional, tornando-se mais leves que máquinas virtuais.



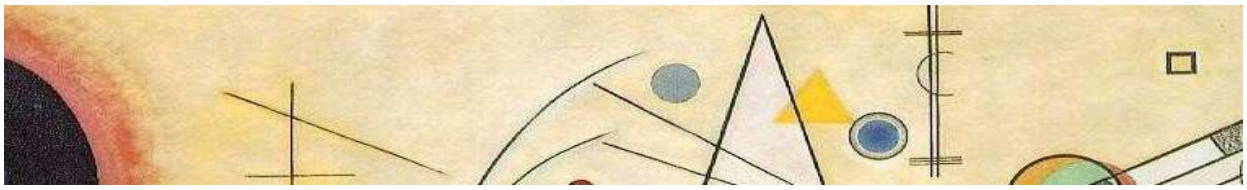
PALAVRAS-CHAVE: Docker, container, configurações, sistemas operacionais.

ABSTRACT: Docker is a tool that allows you to create and manage software containers that contain all the necessary environment for an application to be executed consistently across different operating systems. By using Docker, it is possible to standardize the development environment, ensuring that all developers work with the same configuration and avoiding compatibility issues between different versions of libraries and tools. This accelerates the development process, making it more efficient and facilitating collaboration between team members. In addition, Docker also helps simplify the deployment process of an application in production, allowing it to be executed on any system that supports Docker without requiring additional configurations. Another important benefit of Docker is the ease of scaling applications, especially in cloud environments and orchestration platforms like Kubernetes. It also contributes to the efficient use of system resources, as containers share the same operating system kernel, making them lighter than virtual machines.

KEYWORDS: Docker, container, configurations, operating systems.

RESUMEN: Docker es una herramienta que permite crear y gestionar contenedores de software que incluyen todo el entorno necesario para que una aplicación pueda ejecutarse de forma consistente en diferentes sistemas operativos. Con el uso de Docker, es posible estandarizar el entorno de desarrollo, garantizando que todos los desarrolladores trabajen con la misma configuración y evitando problemas de compatibilidad entre diferentes versiones de bibliotecas y herramientas. Esto acelera el proceso de desarrollo, haciéndolo más eficiente y facilitando la colaboración entre los miembros del equipo. Además, Docker también ayuda a simplificar el proceso de implementación de una aplicación en producción, permitiendo que se ejecute en cualquier sistema que tenga soporte para Docker, sin necesidad de configuraciones adicionales. Otro beneficio importante de Docker es la facilidad para escalar aplicaciones, especialmente en entornos de nube y orquestación con herramientas como Kubernetes. También contribuye al uso eficiente de los recursos del sistema, ya que los contenedores comparten el mismo núcleo del sistema operativo, lo que los hace más livianos que las máquinas virtuales.

PALABRAS CLAVE: Docker, contenedores, configuraciones, sistemas operativos



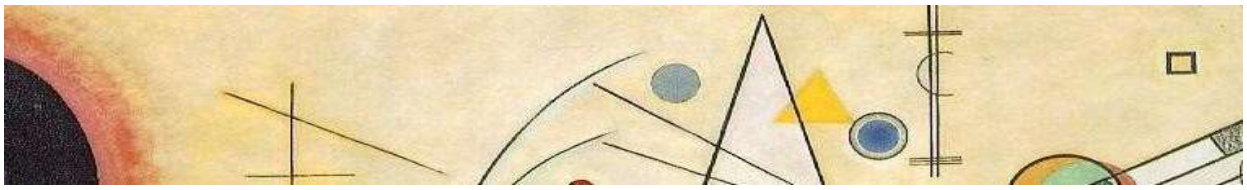
1. Introdução

O Docker é uma plataforma de software que permite que os desenvolvedores empacotem, distribuam e executem aplicativos em contêineres isolados. Esses contêineres são ambientes independentes que contêm todos os elementos necessários para executar um aplicativo, incluindo o código, as bibliotecas, as dependências e as configurações. Com essa abordagem, é possível garantir a consistência e a portabilidade do ambiente de desenvolvimento, facilitando a colaboração e a integração contínua.

Uma das principais vantagens do Docker é a sua capacidade de simplificar a implantação e a escalabilidade de aplicativos em diferentes ambientes, como nuvens públicas, privadas ou híbridas. Isso ocorre porque os contêineres são altamente portáteis e podem ser executados em diferentes sistemas operacionais e infraestruturas, sem a necessidade de reescrever o código ou fazer grandes modificações. Além disso, o Docker oferece recursos avançados de gerenciamento de recursos e segurança, tornando-se uma opção popular para empresas que desejam adotar uma abordagem de DevOps ágil e eficiente.

Em resumo, o Docker é uma ferramenta essencial para os desenvolvedores modernos que desejam construir aplicativos escaláveis e portáteis em diferentes ambientes de computação. Com a sua capacidade de empacotar, distribuir e executar aplicativos em contêineres isolados, o Docker permite que os desenvolvedores se concentrem no desenvolvimento de software, enquanto a plataforma cuida da infraestrutura. Com a crescente adoção da computação em nuvem e da abordagem DevOps, é provável que o Docker continue a ser uma tecnologia fundamental para a criação de aplicativos inovadores e escaláveis no futuro.

Este estudo tem como objetivo demonstrar como a tecnologia Docker pode reduzir custos em implantações em nuvem. Para isso, será utilizada a



infraestrutura da AWS, comparando-se os custos do serviços ECS (Elastic Container Service) com o EC2 (Elastic Compute Cloud), de forma a evidenciar as diferenças de preços entre essas duas abordagens.

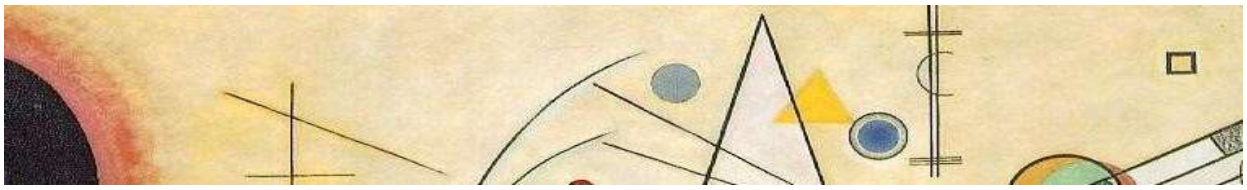
2. Revisão de Literatura

Embora o uso de contêineres seja feito desde 2006 com os Linux Containers, que estavam presentes em datacenters do Google, essa tecnologia só foi popularizada com o surgimento de containers Docker em 2013. A popularização ocorreu devido à simplicidade com que containers Docker são executados, além da maior padronização envolvida, se comparado com os Linux Containers (STALLINGS, 2018).

A empresa por trás dos containers Docker, Docker Inc., define o Docker como "uma plataforma aberta para desenvolvimento, entrega e execução de aplicações" (2019a), agindo como uma interface entre a infraestrutura de rede e as aplicações, permitindo de modo simples e ágil, entrega, gerência e monitoramento de aplicações através de ferramentas de orquestração de contêineres. Desse modo, a plataforma oferece alta escalabilidade e alta disponibilidade dos serviços. Devido a essas características, a Plataforma Docker é vendida e amplamente utilizada como uma boa solução para softwares construídos com a Arquitetura de Microsserviços (DOCKER INC., 2019c), os quais têm tais características de agilidade e eficiência.

Inicialmente o Docker implementou a solução de contêineres através de uma

tecnologia chamada Linux Container (LXC), entretanto, com o passar do tempo, a plataforma customizou suas tecnologias e ferramentas numa ferramenta chamada containerd, que é um container daemon, ou seja, uma interface de execução de contêiner que implementa o funcionamento de contêineres em diferentes Sistemas Operacionais, permitindo a interoperabilidade da plataforma (CROSBY, 2017).



Visando a padronização do mercado, a Docker Inc. doou o containerd para a Cloud Native Computing Foundation (CNCF) e em 28 de fevereiro de 2018, o projeto containerd foi considerado maduro para o mercado (DOCKER INC., 2019b; CLOUD NATIVE COMPUTING FOUNDATION, 2018).

Contêineres Docker são definidos como uma instância de execução isolada, independente e efêmera de uma Imagem Docker (em inglês, Docker Images), que são armazenados em um registro chamado Docker Registry (TORRE, 2019; DOCKER INC., 2019a; STALLINGS, 2018, p. 641; ELDRIDGE, 2018). Uma Docker Image especifica as tarefas, o comportamento e os processos que serão executados e é definido por um template chamado Docker Image (DOCKER INC., 2019c).

Tais contêineres são criados e gerenciados por meio de uma Interface de Linha de Comando (em inglês, Command Line Interface – CLI) chamada Docker CLI, a qual utiliza uma API REST para se comunicar com a interface de execução de contêineres chamada de Docker Daemon ou dockerd. Esse funcionamento caracteriza o Docker Engine (DOCKER INC., 2019a).

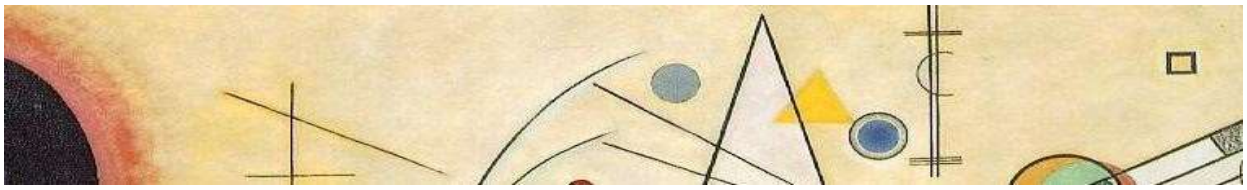
3. Metodologia

Neste trabalho, utilizamos o Docker para ilustrar o processo de conteinerização de uma aplicação web baseada em microsserviços, composta por duas partes principais: a aplicação em Node.js e um banco de dados (MySQL). Assim, temos processo de criação e configuração dos containers para cada serviço, bem como a comunicação entre eles usando network no próprio docker compose.

3.1 Estrutura da Aplicação

Para este exemplo, a aplicação foi estruturada da seguinte forma:

- **Backend**: Um container Node.js que processa as requisições.



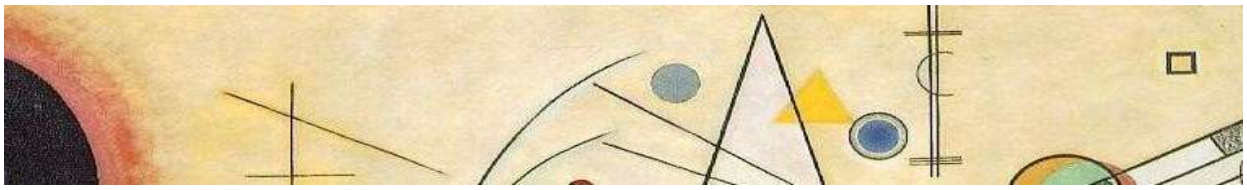
- **Banco de Dados:** Um container MySQL que armazena as informações necessárias para a aplicação.

Cada um desses serviços foi configurado para rodar em um container separado, permitindo a escalabilidade independente de cada parte da aplicação.

3.2 Criação do Dockerfile

Para o backend em Node.js, criamos um Dockerfile específico para garantir que todas as dependências e configurações sejam reproduzíveis. O arquivo Dockerfile foi construído da seguinte forma:

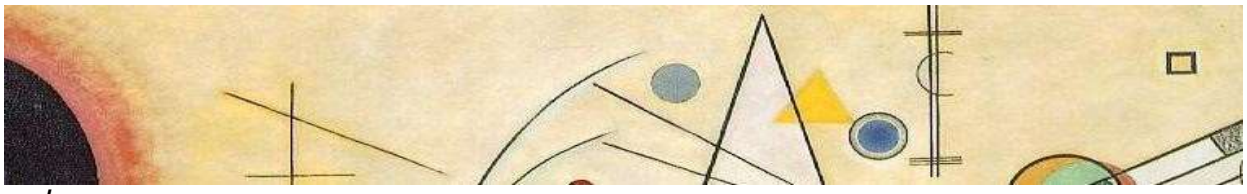
```
FROM node:18 AS build

WORKDIR /usr/src/app

COPY package.json yarn.lock ./

RUN yarn

COPY . .

RUN yarn run build

RUN npm prune --production

FROM node:18-alpine3.19

WORKDIR /usr/src/app

COPY --from=build /usr/src/app/package.json ./package.json
COPY --from=build /usr/src/app/dist ./dist
COPY --from=build /usr/src/app/node_modules ./node_modules

EXPOSE 3000

CMD [ "yarn", "run", "start:prod" ]
```



3.3 Configuração do docker-compose.yml

Para orquestrar os containers e definir as redes, utilizamos o docker-compose.yml, que facilita o processo de subir múltiplos containers simultaneamente. Abaixo está o arquivo de configuração:

```
version: '3.7'

services:
mysql:
image: mysql:8
container_name: mysql
volumes:
- db:/var/lib/mysql
ports:
- 3306:3306
environment:
- MYSQL_ROOT_PASSWORD=root
- MYSQL_DATABASE=rocketseat-db
- MYSQL_USER=admin
- MYSQL_PASSWORD=root
networks:
- primeira-network

api-rocket:
build:
context: .
container_name: api-rocket
ports:
- 3000:3000
depends_on:
- mysql
networks:
- primeira-network

networks:
primeira-network:
name: primeira-network
external: true
driver: bridge
```



volumes:
db:

4. Explicação do Processo de Implantação

A execução do comando docker-compose up inicia todos os serviços. O docker-compose cria redes para que os containers se comuniquem internamente e define as dependências entre eles, garantindo que o banco de dados seja inicializado antes do backend, por exemplo. Através dessa configuração, a aplicação está pronta para ser executada em um ambiente controlado, onde cada serviço está isolado, mas interligado, facilitando a manutenção e o escalonamento.

5. Máquinas Virtuais

Uma máquina virtual (VM) é um ambiente computacional virtualizado que simula um computador físico dentro de um sistema operacional. Ela permite rodar múltiplos sistemas operacionais independentes em um único hardware físico, isolando cada um deles como se fossem máquinas separadas.

A VM é criada e gerenciada por um hypervisor, que é um software responsável por virtualizar os recursos físicos (CPU, memória, disco, rede) e distribuí-los entre as máquinas virtuais. Existem dois tipos principais de hypervisors:

- **Hypervisor Tipo 1 (Bare Metal)**: Roda diretamente sobre o hardware físico (exemplo: VMware ESXi, Microsoft Hyper-V, KVM).
- **Hypervisor Tipo 2 (Hosted)**: Roda sobre um sistema operacional já existente (exemplo: VirtualBox, VMware Workstation).

5.1 Principais Características

Isolamento: Cada VM possui seu próprio sistema operacional, arquivos e aplicações, funcionando de forma independente.

Flexibilidade: Permite rodar diferentes sistemas operacionais no mesmo hardware, como Linux e Windows simultaneamente.

Segurança: Como as VMs são isoladas, um problema em uma delas não afeta as demais.

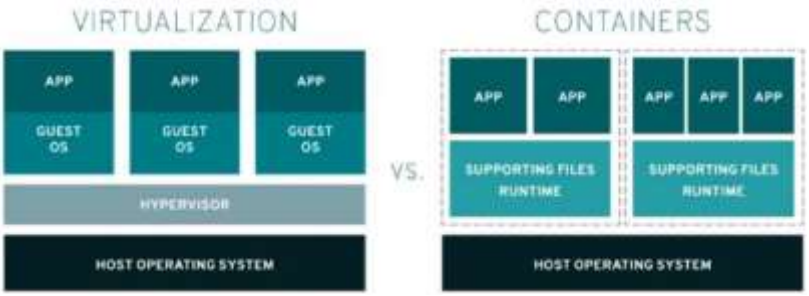
Gerenciamento Centralizado: Em ambientes corporativos, as VMs podem ser gerenciadas remotamente através de plataformas como VMware vSphere ou Microsoft Azure.

5.2 Casos de Uso

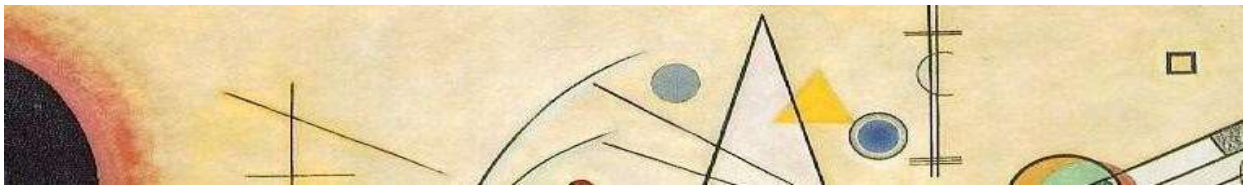
- Hospedagem de servidores sem a necessidade de hardware físico adicional.
- Testes de software em diferentes sistemas operacionais.
- Execução de aplicações legadas sem afetar o sistema principal.
- Ambientes de desenvolvimento e testes em nuvem.

6. Containers vs. Máquinas Virtuais

Figura 1. Containers vs. Máquinas Virtuais



Fonte:https://www.redhat.com/pt-br/topics/containers/containers-vs-vms (acessado em 08/05/2025)



7. AWS EC2 vs. AWS ECS

O Amazon Elastic Compute Cloud (EC2) e o Amazon Elastic Container Service (ECS) são serviços da AWS que oferecem diferentes abordagens para hospedar e gerenciar aplicações na nuvem. O EC2 permite a criação de instâncias de máquinas virtuais, proporcionando controle total sobre o ambiente de execução. Já o ECS é um serviço de orquestração de contêineres que facilita a execução de aplicações em contêineres Docker, podendo ser utilizado com o AWS Fargate, uma opção que elimina a necessidade de gerenciar servidores.

7.1 Comparação de Custos: EC2 vs. ECS com Fargate

Amazon EC2: As instâncias EC2 oferecem flexibilidade nos modelos de pagamento:

- *Sob Demanda:* Pague apenas pelo que usar, sem compromissos de longo prazo. Por exemplo, uma instância t3.medium (2 vCPUs, 4 GB de RAM) na região de São Paulo custa aproximadamente USD 0,0416 por hora.
- *Instâncias Reservadas:* Compromisso de 1 ou 3 anos com descontos significativos. A mesma instância t3.medium, com contrato de 1 ano e pagamento antecipado total, custa cerca de USD 0,026 por hora.
- *Instâncias Spot:* Utilização de capacidade ociosa da AWS com descontos de até 90%. A instância t3.medium pode custar aproximadamente USD 0,010 por hora nesse modelo.

Amazon ECS com AWS Fargate: O Fargate permite executar contêineres sem gerenciar servidores, cobrando pelos recursos provisionados:

- *CPU e Memória:* Por exemplo, na região Leste dos EUA (Norte da Virgínia), o custo é de USD 0,000011244 por segundo por vCPU e USD

0,000001235 por segundo por GB de memória. Uma tarefa utilizando 1 vCPU e 2 GB de memória por 10 minutos diariamente durante 30 dias resultaria em um custo mensal de aproximadamente USD 1,26.

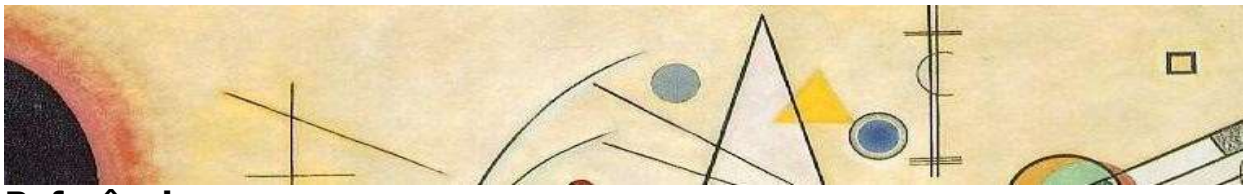
Armazenamento Temporário: 20 GB estão incluídos por padrão; armazenamento adicional é cobrado separadamente.

8. Considerações Finais

A escolha entre EC2 e ECS com Fargate depende das necessidades específicas da sua aplicação. O ECS com Fargate pode ser mais conveniente e simplificado para workloads baseados em contêineres, especialmente por eliminar o esforço de gerenciamento de servidores. No entanto, para aplicações que requerem controle total sobre o ambiente, ou que podem se beneficiar de modelos de preços como Instâncias Spot ou Reservadas, o EC2 pode ser mais vantajoso.

No exemplo analisado, uma instância EC2 t3.medium no modelo Reservado por 1 ano custa cerca de USD 0,13 por mês (equivalente a aproximadamente R$ 0,74, considerando o dólar a R$ 5,66), sendo mais barata do que o uso do ECS com Fargate, que custaria cerca de USD 1,26 por mês (aproximadamente R$ 7,13).

Com isso, podemos ver que, ao usar Docker no dia a dia, temos diversas opções de serviços em nuvem, e a escolha ideal deve considerar tanto o custo quanto a flexibilidade e o nível de controle necessário para sua aplicação.



Referências

AMAZON WEB SERVICES. *Definição de preço do Amazon ECS*. Disponível em: https://aws.amazon.com/pt/ecs/pricing/. Acesso em: 12 abr. 2025.
AMAZON WEB SERVICES. *Definição de preço do AWS Fargate*. Disponível em: https://aws.amazon.com/pt/fargate/pricing/. Acesso em: 12 abr. 2025.

AWS. The difference between containers and virtual machines. Amazon Web Services, 2025. Disponível em: https://aws.amazon.com/pt/compare/the-difference-between-containers-and-virtual-machines/. Acesso em: 22 fev. 2025.

CHELES, Paulo. *Custos e Preços do AWS EC2: guia completo para otimizar sua infraestrutura na nuvem*. UDS Tecnologia, 13 dez. 2024. Disponível em: https://uds.com.br/blog/custos-precos-ec2-aws-como-otimizar-cloud. Acesso em: 12 abr. 2025.

GOMES, Rafael. **Docker para desenvolvedores**. Salvador: Instruct, 2017. 175 p.

RED HAT. Containers vs. VMs. Red Hat, 2025. Disponível em: https://www.redhat.com/pt-br/topics/containers/containers-vs-vms. Acesso em: 22 fev. 2025.

RESENDE, Pedro Augusto. **ADOÇÃO DE ARQUITETURA DE CONTÊINERES DOCKER**: um estudo de caso. 2019. 97 f. TCC (Graduação) - Curso de Ciência da Computação, Escola de Educação, Tecnologia e Comunicação Curso de Ciência da Computação, Brasília, 2019.

SILVA FILHO, Antonio Joaquim e. **ANÁLISE DE MODELO DE APLICAÇÕES DE ALTA DISPONIBILIDADE**: um estudo da aplicabilidade das tecnologias docker e kubernetes. 2020. 40 f. Monografia (Especialização) - Curso de Sistemas de Informação, Centro Universitário Unidade de Ensino Superior Dom Bosco Curso Sistemas de Informação, São Luís, 2020.