

Fundamentos de DevOps

<http://linkedin.com/in/guijac>

Aula 12 - Automação de Build e Contêineres na Nuvem

Prof. Esp. Guilherme Jorge Aragão da Cruz

 guilherme.cruz@alumni.usp.br

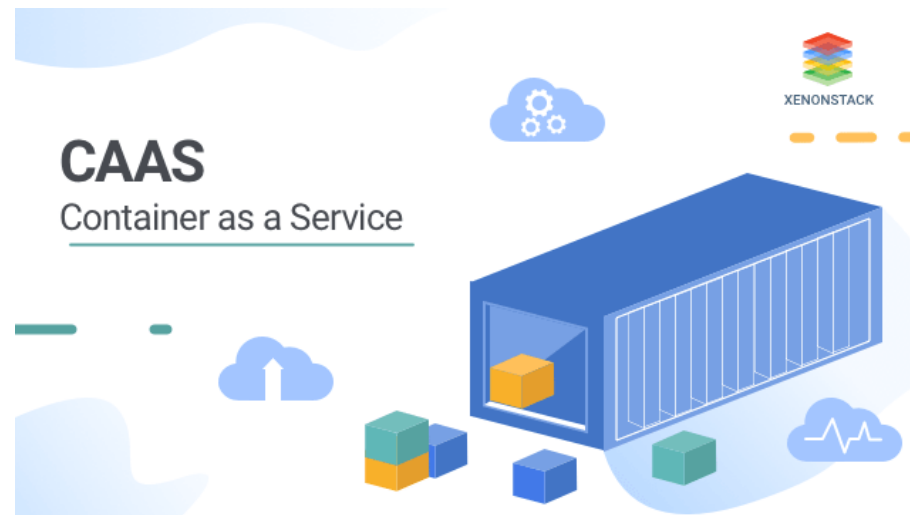
 linkedin.com/in/guijac

Roteiro

- Novos Modelos de Computação em Nuvem:
 - *Container as a Service*;
 - *Function as a Service*.
- Revisitando Redes de Computadores:
 - *Internet Protocol address (IP)*;
 - *Classes Inter-Domain Routing (CIDR)*;
 - *Virtual Private Cloud (VPC)*;
 - *Elastic Load Balancer (ELB)*.
- GitLab Container Registry;
- Amazon ECR;
- Automação de Build no Gitlab-CI;
- Amazon ECS e Principais Conceitos:
 - *Task Definition*;
 - *Service*;
 - *Cluster*.
- Referências Bibliográficas.

Novos Modelos de Computação em Nuvem

- *Container as a Service (CaaS):*
 - Permite a implantação de aplicações baseadas em contêineres sem uma preocupação com outros componentes, como o sistema operacional.
- *Function as a Service (FaaS):*
 - Possibilita a criação e implantação de pequenos códigos, com funções especializadas para a execução de uma única tarefa, tal como o envio de um e-mail.



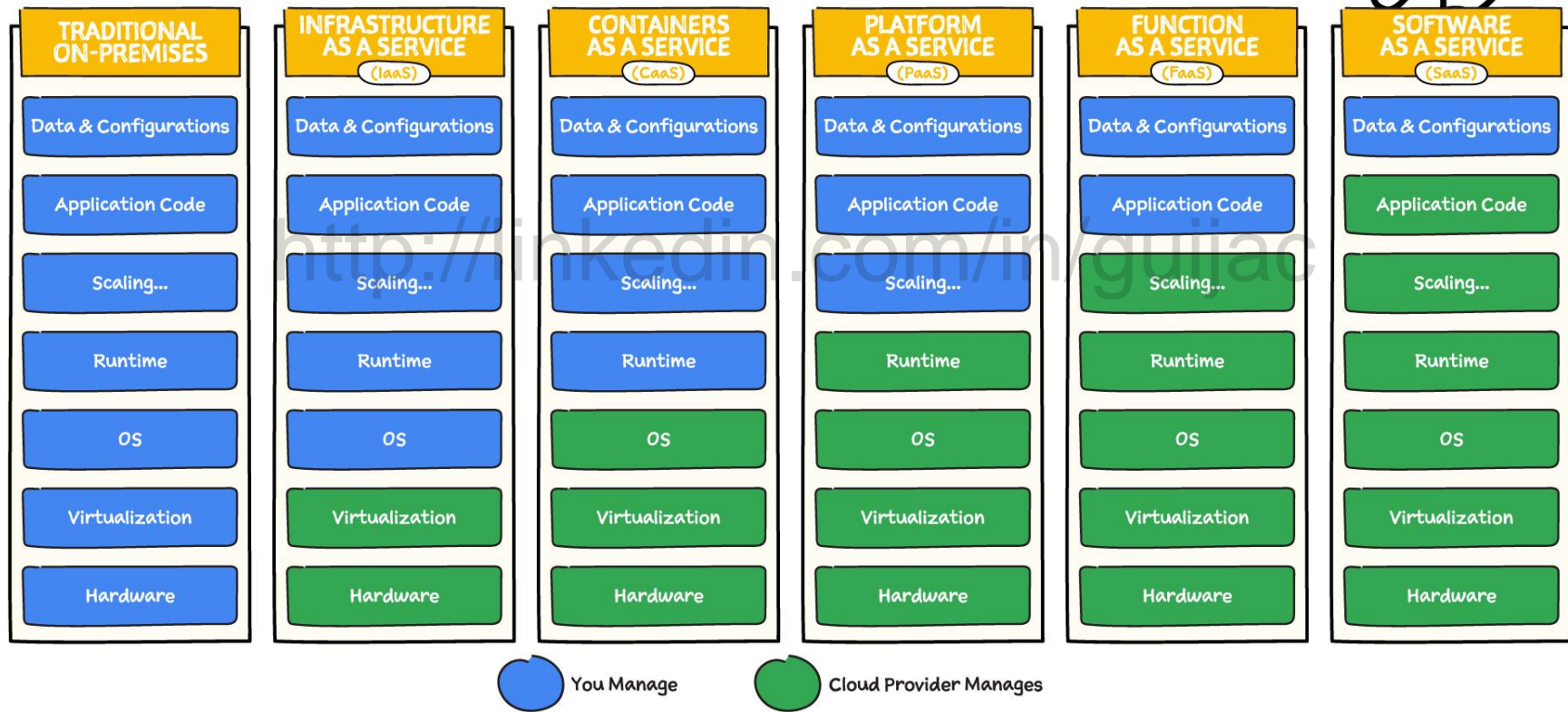
Fonte: [Containers as a Service \(CaaS\) Architecture and Management Solutions](#)

Novos Modelos de Computação em Nuvem

@pvergadia



Wait... what is Cloud again?



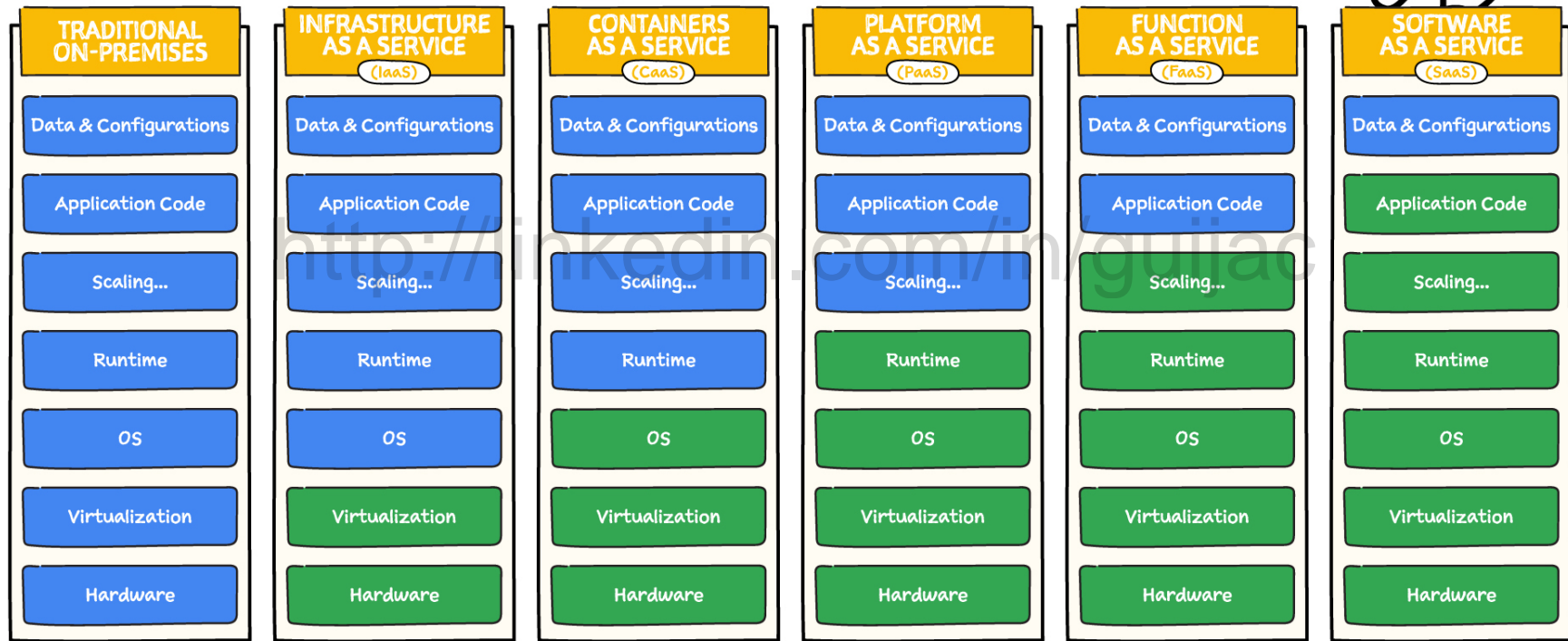
Fonte: [PaaS x IaaS e SaaS: qual é a diferença?](#) | [Google Cloud](#)

Novos Modelos de Computação em Nuvem

@pvergadia



Wait... what is Cloud again?



Amazon EC2



AWS ECS

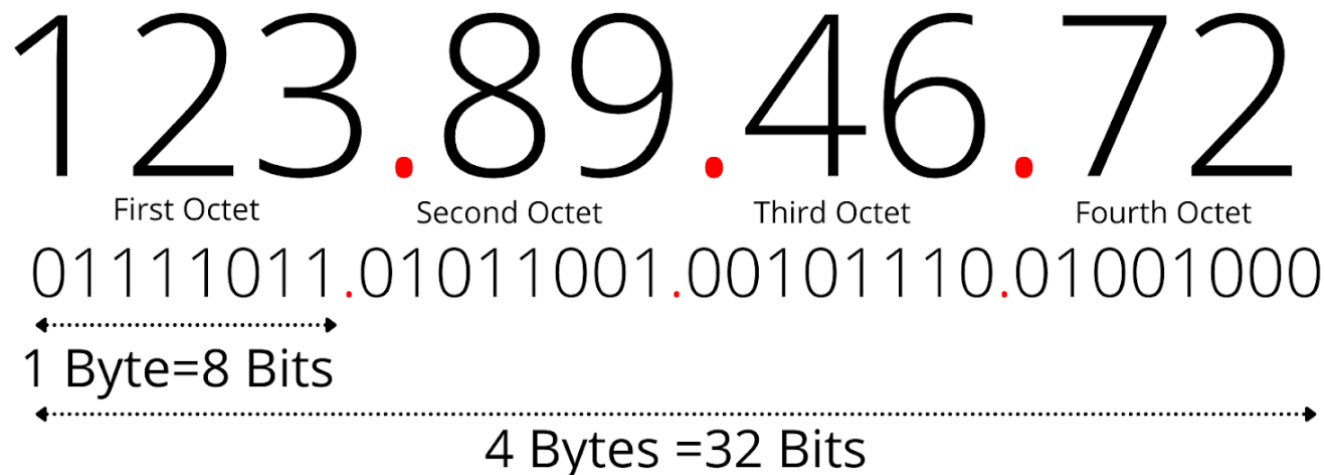


Fonte: [PaaS x IaaS e SaaS: qual é a diferença?](#) | [Google Cloud](#)

Internet Protocol Address (IP)

- Cada máquina, servidor e dispositivo de usuário final que se conecta à Internet tem um número exclusivo, chamado endereço IP, associado a ele;
- Um endereço IPv4 consiste em 32 bits. Cada sequência de números separados pelo ponto consiste em 8 bits, representados por 0 a 255 em formas numéricas.

IPv4 Address Format (Dotted Decimal Notation)



Fonte: [What is an IP Address? - zenarmor.com](https://zenarmor.com/what-is-an-ip-address/)

Classes Inter-Domain Routing (CIDR)

- A notação Classes Inter-Domain Routing (CIDR) representa um endereço IP e um sufixo que indica bits reservados para uma rede. Por exemplo, 10.0.0.0/16 é um endereço IPv4 no qual os primeiros 16 bits, ou 10.0, são o endereço da rede;
- Traz uma maior flexibilidade para provisionar endereços IP, organizando-os em sub-redes.

<http://linkedin.com/in/guijac>

Bloqueio de endereço de rede	Máscara de sub-rede	Número de hosts/sub-redes	Número de sub-redes
10.0.0.0/16	255.255.0.0/16	65536	1
Intervalo de endereço de host	Endereço de difusão	Máscara curinga	Notações de CIDR
10.0.0.1 - 10.0.255.254	10.0.255.255	0.0.255.255	10.0.0.0/16

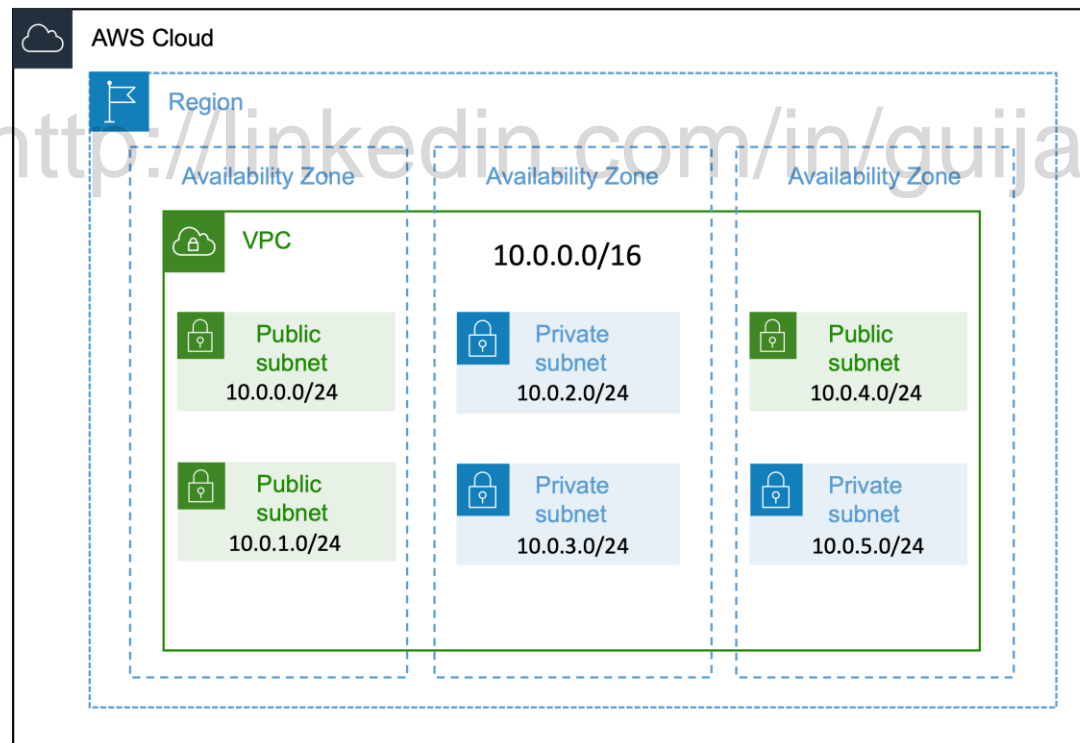
Detalhes da sub-rede

Subnet ID	Subnet Address	Host Address Range	Broadcast Address
1	10.0.0.0	10.0.0.1 - 10.0.255.254	10.0.255.255

Fonte: [Calculadora de sub-redes de IP para IPv4](#) | [Calculadora de máscara de sub-rede online](#) - Site24x7

Virtual Private Cloud (VPC)

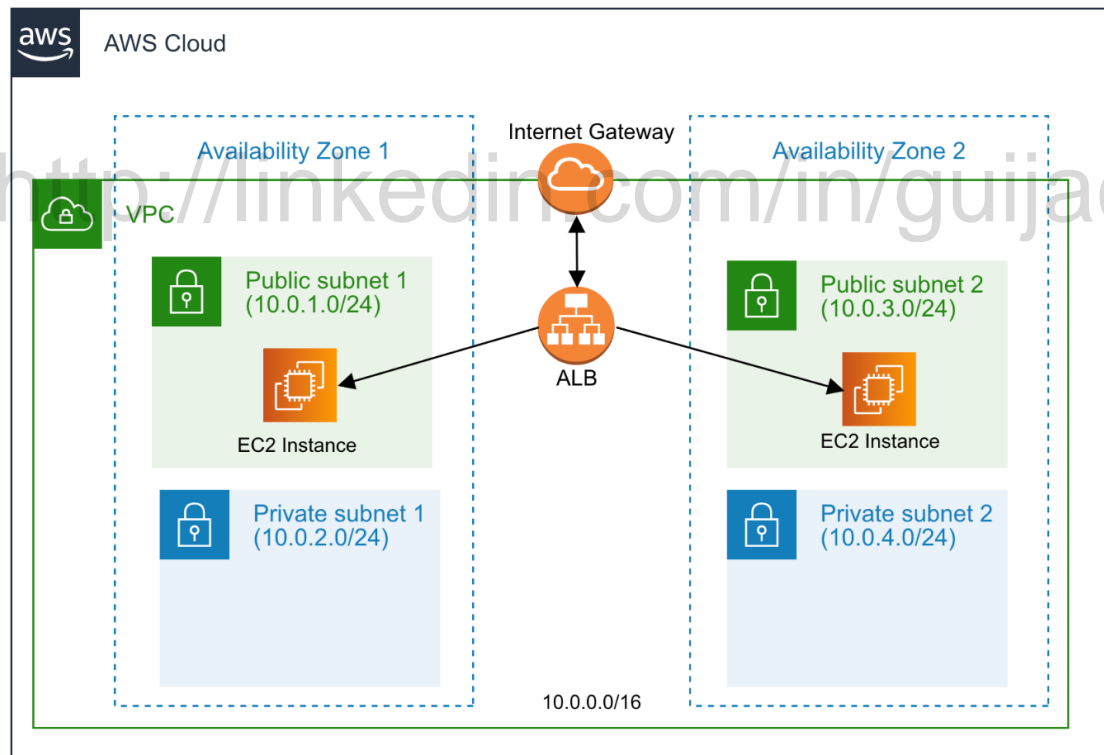
- Uma rede virtual que assemelha-se a uma rede tradicional de um datacenter, mas com o benefício de utilizar-se da infraestrutura da nuvem.



Fonte: [Simon Holdorf no Twitter](#)

Elastic Load Balancer (ELB)

- Distribui automaticamente o tráfego de entrada entre vários destinos, como instâncias EC2 e contêineres, em uma ou mais zonas de disponibilidade.



Fonte: [CloudFormation Template for VPC with EC2 and ALB | DevelopersIO \(classmethod.jp\)](https://www.developers.io/cloudformation-template-for-vpc-with-ec2-and-elb)

GitLab Container Registry

- Serviço para repositório de contêineres, sendo totalmente gerenciado pela Gitlab;
- **Simplificando: um Docker Hub da Gitlab.**



The screenshot shows the GitLab Container Registry interface for the project 'omnibus-gitlab'. The left sidebar contains navigation options: Project, Activity, Files, Commits, Pipelines (16), Builds (7), Container Registry (selected), Graphs, and Milestones. The main content area displays the project name and a search bar. Below this, there is a description: 'A 'container image' is a snapshot of a container. You can host your container images with GitLab. To start using container images hosted on GitLab you first need to login:'. A code block shows the command: `docker login registry.gitlab.com`. Below that, it says 'Then you are free to create and upload a container image with build and push commands:' followed by another code block: `docker build -t registry.gitlab.com/gitlab/omnibus-gitlab .` and `docker push registry.gitlab.com/gitlab/omnibus-gitlab`. At the bottom, there is a table with the following data:

Name	Image ID	Size	Created
latest 	6d28225f8	46.1 MB · 5 layers	19 days 

Fonte: [GitLab Container Registry | GitLab](https://gitlab.com/gitlab-org/container-registry)

Amazon ECR

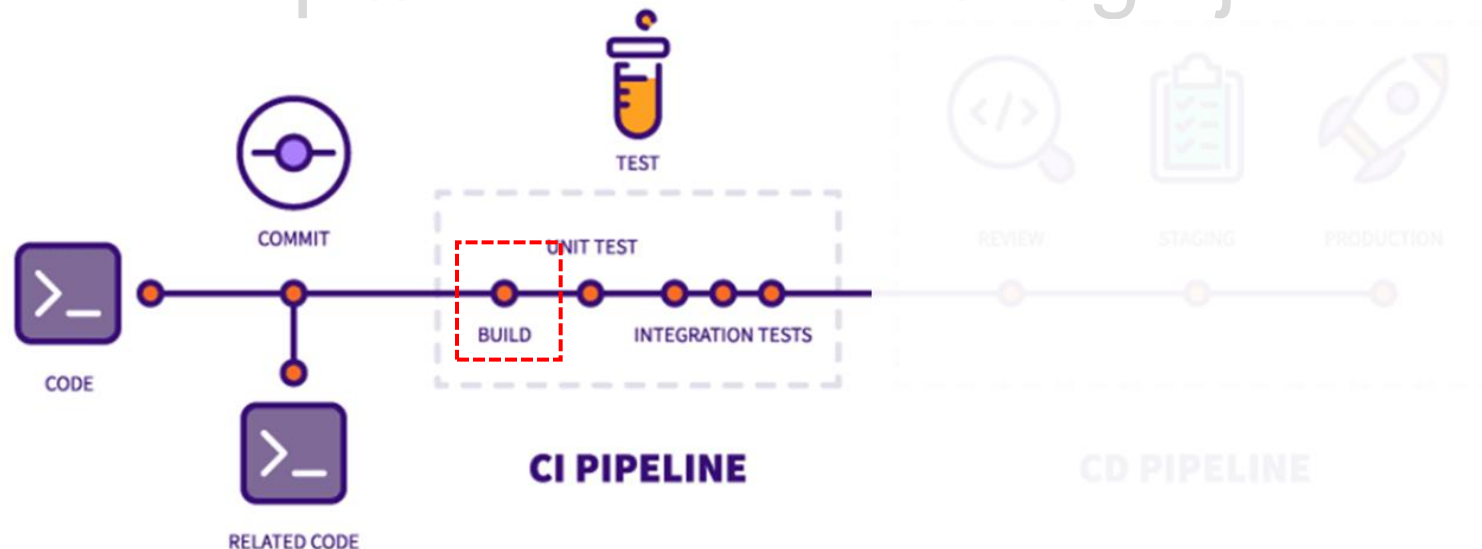
- *Amazon Elastic Container Registry*
 - Serviço para repositório de contêineres, sendo totalmente gerenciado pela AWS;
 - **Simplificando: um Docker Hub da AWS.**



Fonte: [Registro de contêiner totalmente gerenciado – Amazon Elastic Container Registry – Amazon Web Services](#)

Automação de Build no Gitlab

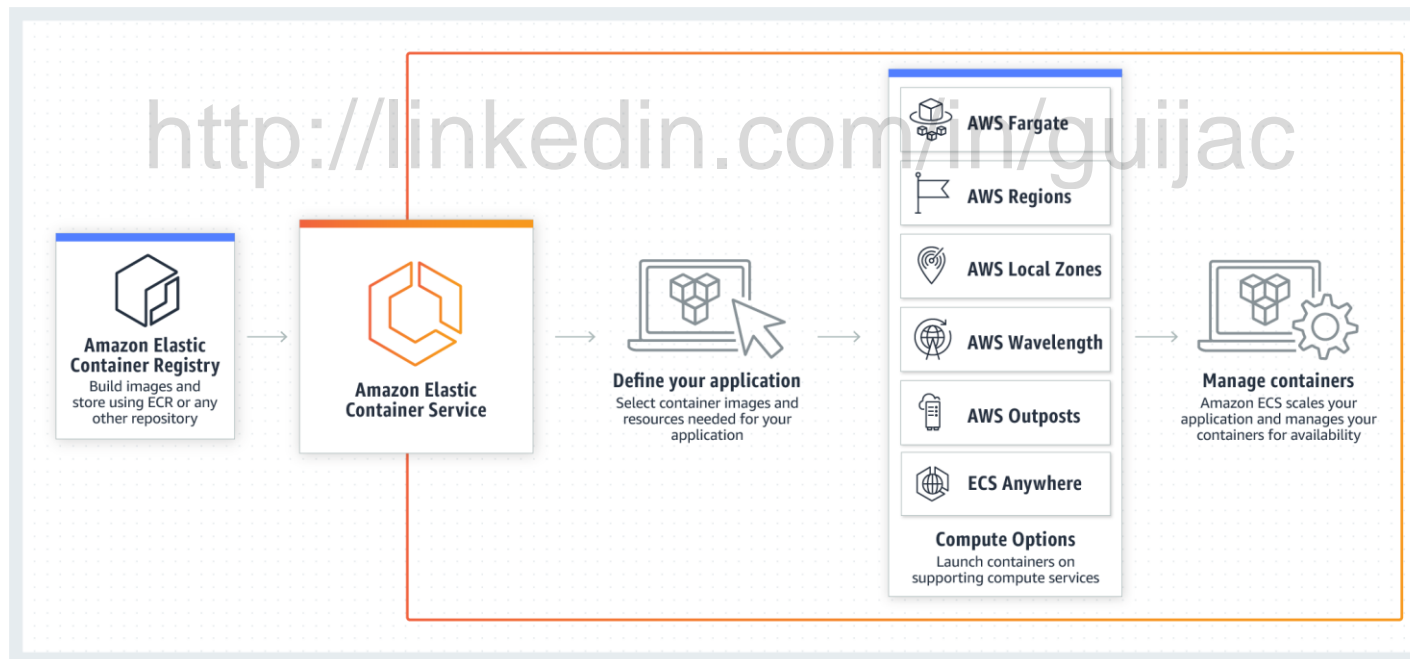
- O GitLab CI/CD pode automatizar builds de imagens Docker:
 - Etapa docker build → gerar a imagem a partir de um Dockerfile.
 - Etapa docker push → enviar a imagem para o GitLab Container Registry ou ECR.
- Uso de variáveis de CI/CD para evitar expor credenciais (CI_REGISTRY_USER, CI_REGISTRY_PASSWORD).



Fonte: Adaptado de [Ultimate guide to CI/CD: Fundamentals to advanced implementation](#)

Amazon ECS

- *Amazon Elastic Container Service*
 - Serviço de orquestração de contêineres, sendo totalmente gerenciado pela AWS;
 - **Simplificando: um orquestrador de contêineres da AWS.**

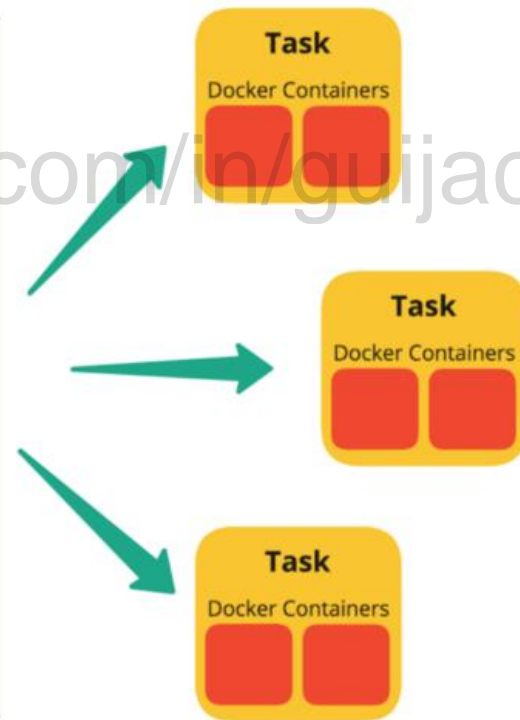


Fonte: [Amazon EC2 Container Service - AWS](#)

Amazon ECS: *Task Definition*

- Prepara uma aplicação para execução.
- É um arquivo de texto, no formato JSON, que descreve um ou mais containers.

```
taskDefinition:
  containerDefinitions:
    - name: "my-app"
      image: "xxx.ecr.{{region}}.amazonaws.com/xxx:{{tag}}"
      cpu: 300
      memory: 750
      portMappings:
        - containerPort: 5000
          protocol: "tcp"
      links:
        - statsd:statsd
      environment:
        - name: 'NODE_ENV'
          value: '{{ environment }}'
        - name: 'AWS_REGION'
          value: '{{ region }}'
        - name: 'UV_THREADPOOL_SIZE'
          value: '128'
    - name: "statsd"
      image: "xxx.ecr.{{region}}.amazonaws.com/xxx-statsd:4.5.0"
      cpu: 64
      memory: 64
      environment:
        - name: "TARGET"
          value: '{{ statsd_target }}'
      startupTimeout: 20
      taskRoleArn: {{ task_role_arn }}
```



Fonte: [A beginner's guide to Amazon's Elastic Container Service \(freecodecamp.org\)](https://www.freecodecamp.org/learn/amazon-aws/certification/amazon-aws-ecs-task-definition/)

Amazon ECS: *Service*

- Permite a execução e manutenção de um número especificado de instâncias a partir de uma definição de tarefa em um *cluster*.
- Se alguma tarefa falhar o Container Agent irá inicializar outra instância de sua definição de tarefa.

<http://linkedin.com/in/guijac>

```

service:
  minCapacity: {{ desired_count }}
  maxCapacity: {{ max_capacity }}
  policies:
    - adjustmentType: ChangeInCapacity
      stepAdjustments:
        - scalingAdjustment: 1
          metricIntervalLowerBound: 8.0
      cooldown: 10
      metricAggregationType: Average
  alarms:
    - namespace: AWS/ECS
      metricName: CPUUtilization
      dimensions:
        - name: ClusterName
        - name: ServiceName
      statistic: Average
      comparisonOperator: GreaterThanOrEqualToThreshold
      threshold: 80.0
      evaluationPeriods: 3
      period: 60
  
```

Service : Update Delete

Cluster	Desired count	4
Status	Pending count	0
Task definition	Running count	4
Launch type	EC2	
Service role		

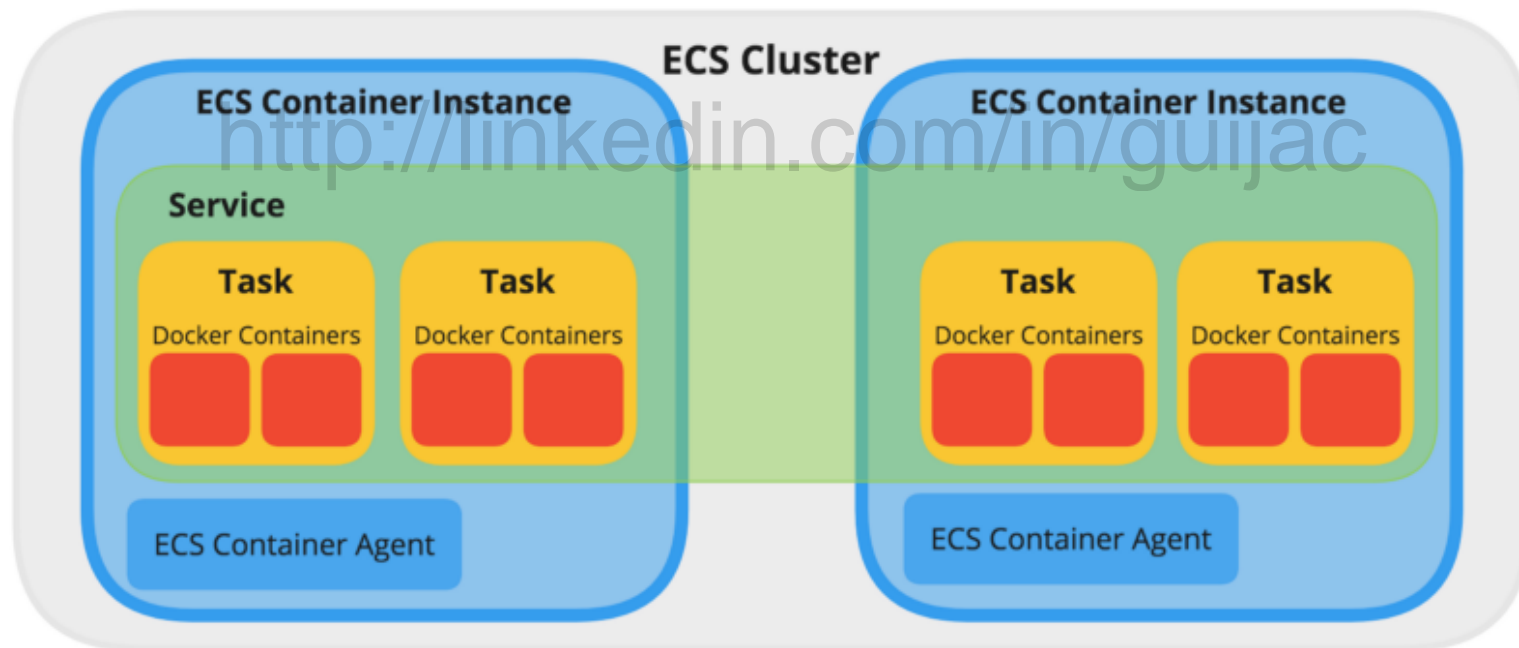
Details Tasks Events **Auto Scaling** Deployments Metrics

Minimum tasks: 4	Maximum tasks: 10
-scaling-policy-2: CPUUtilization < 20	-scaling-policy-1: CPUUtilization > 80
Policy type: Step scaling	Policy type: Step scaling
For alarms:	For alarms:
Take the action: Remove 1 tasks when 20 >= CPUUtilization	Take the action: Add 1 tasks when 80 <= CPUUtilization

Fonte: [A beginner's guide to Amazon's Elastic Container Service \(freecodecamp.org\)](https://www.freecodecamp.org/learn/amazon-aws/certification-preparation-guide/amazon-aws-certification-preparation-guide-10/)

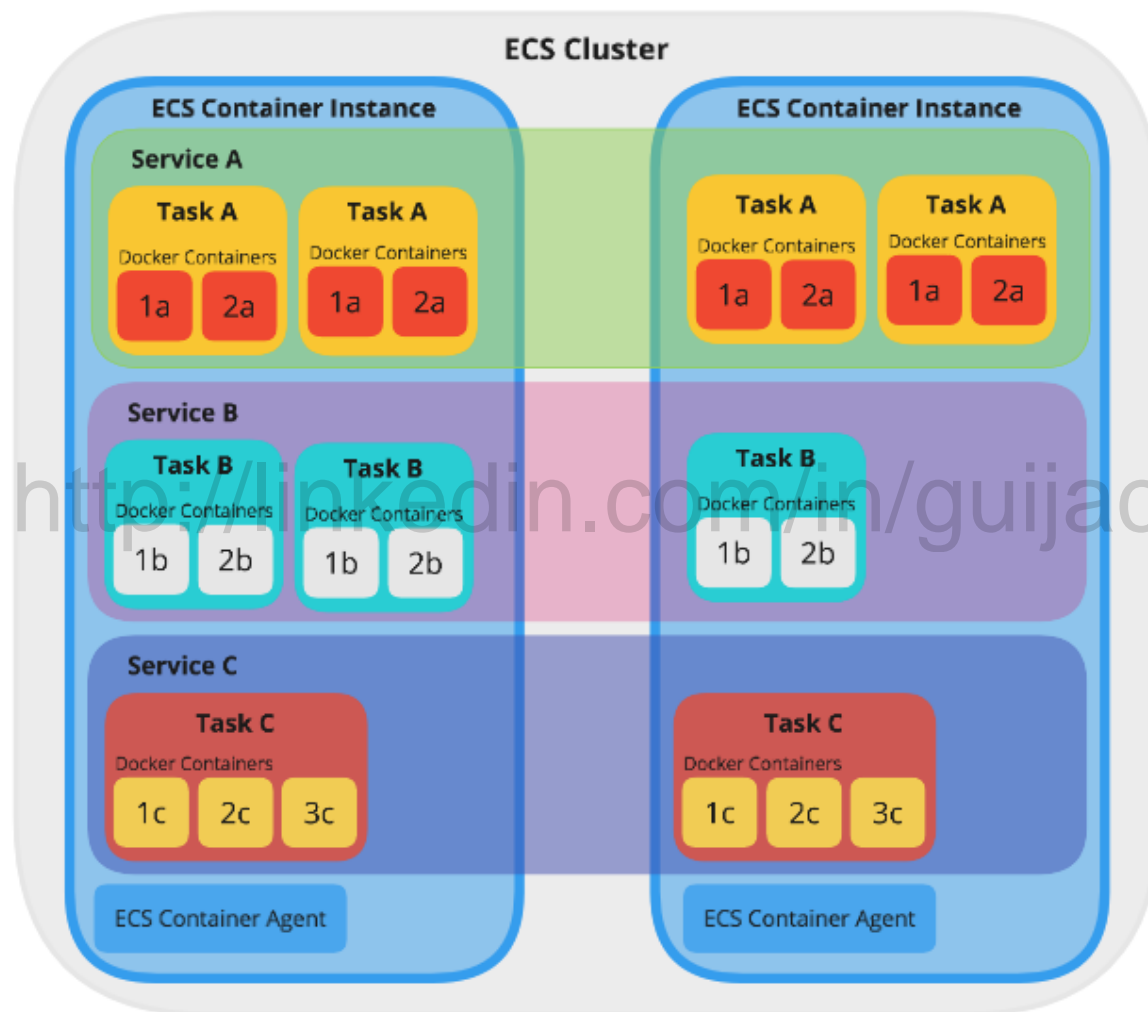
Amazon ECS: *Cluster*

- Um grupo de instâncias ECS;
- Permite agrupar múltiplos serviços e tarefas, minimizando o tempo de configuração e uso mais eficiente de recursos.



Fonte: [A beginner's guide to Amazon's Elastic Container Service \(freecodecamp.org\)](https://www.freecodecamp.org)

Amazon ECS: *Visão Geral*



Cluster ECS executando três serviços, cada um definido por suas próprias tarefas, distribuídas entre duas instâncias de contêiner.

Fonte: [A beginner's guide to Amazon's Elastic Container Service \(freecodecamp.org\)](https://www.freecodecamp.org)

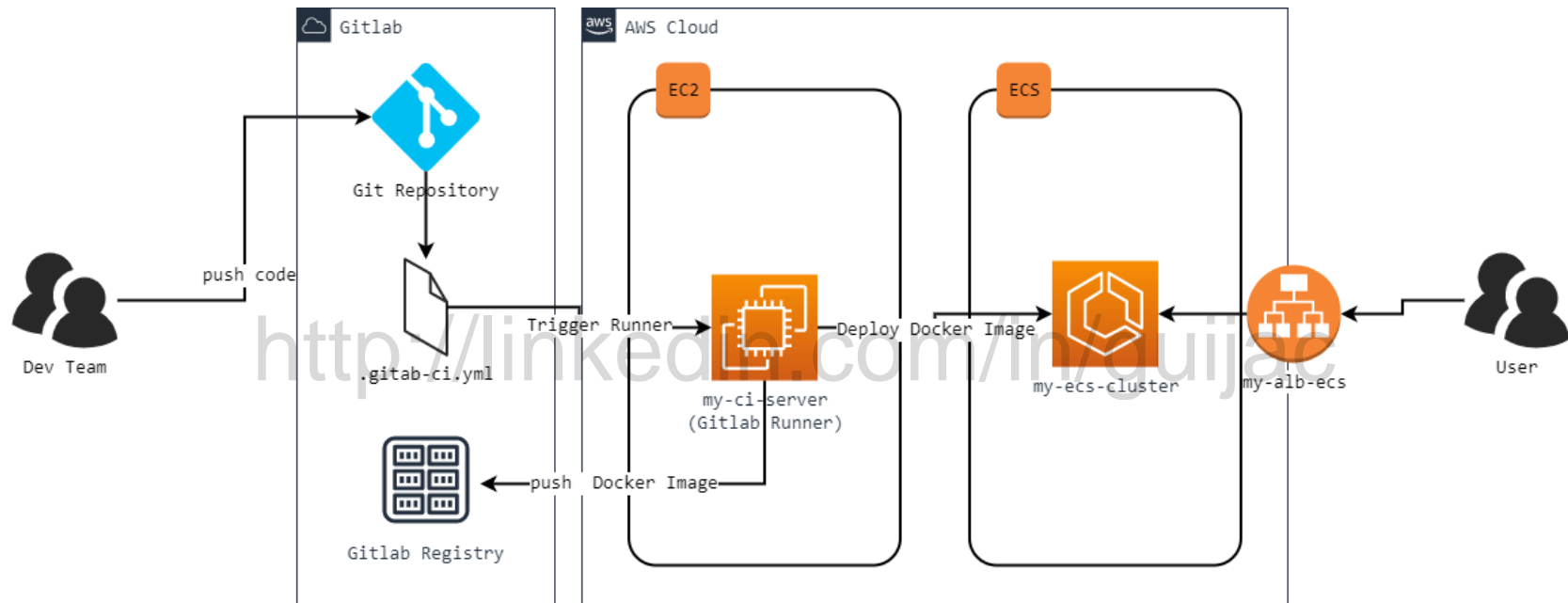
Referências Bibliográficas

AWS. Amazon Elastic Container Registry. Disponível em <https://aws.amazon.com/pt/ecr/>. Acesso em 15 set 2025;

AWS. Amazon Elastic Container Service. Disponível em <https://aws.amazon.com/pt/ecs/>. Acesso em 15 set 2025;

GITLAB. Deploy to Amazon Elastic Container Service. Disponível em https://docs.gitlab.com/ci/cloud_deployment/ecs/deploy_to_aws_ecs/. Acesso em 15 set 2025.

Por hoje (de teoria!) é só!



Fonte: Elaboração própria.

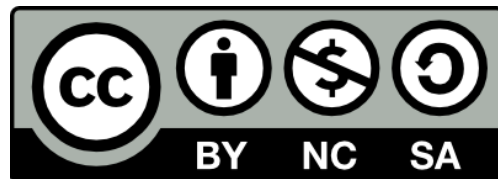
Prof. Esp. Guilherme Jorge Aragão da Cruz

 guilherme.jacruz@sp.senac.br

 linkedin.com/in/guijac

Licença

- Este conteúdo está licenciado sob a Licença Creative Commons Atribuição-NãoComercial-Compartilhalgual 4.0 Internacional (CC BY-NC-SA 4.0).
- Todos os direitos autorais sobre este conteúdo pertencem ao autor, e este material não pode ser usado comercialmente sem autorização expressa.
- Para ver o texto completo da licença, acesse o <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>.



Prof. Esp. Guilherme Jorge Aragão da Cruz

 guilherme.jacruz@sp.senac.br

 [linkedin.com/in/guijac](https://www.linkedin.com/in/guijac)

