

# Curso DevOps

Aula 09 - Introdução ao CI/CD

---

**Prof. Esp. Guilherme Jorge Aragão da Cruz**

 [guilherme.cruz@alumni.usp.br](mailto:guilherme.cruz@alumni.usp.br)

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

# Roteiro

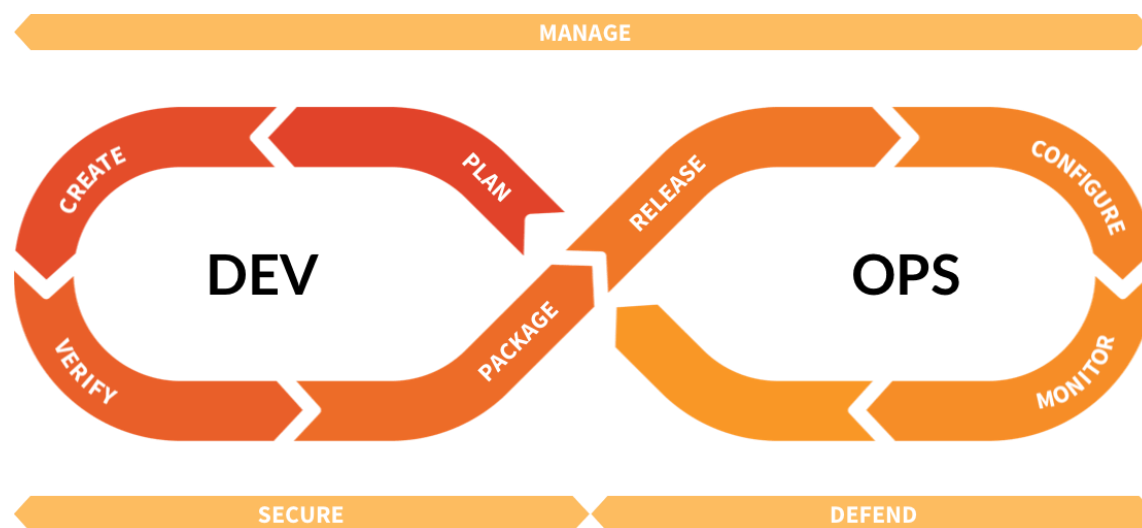
---

- Revisitando DevOps/DevSecOps;
- Revisitando Estrutura Gitlab;
- *Continuous Integration* (CI);
- *Continuous Delivery e Deployment* (CD);
- Juntando as Peças: CI+CD;
- Gitlab Runner:
  - Definição;
  - O arquivo .gitlab-ci.yml.
- Github Actions:
  - O arquivo YAML.
- Referências Bibliográficas;

# Revisitando DevOps/DevSecOps

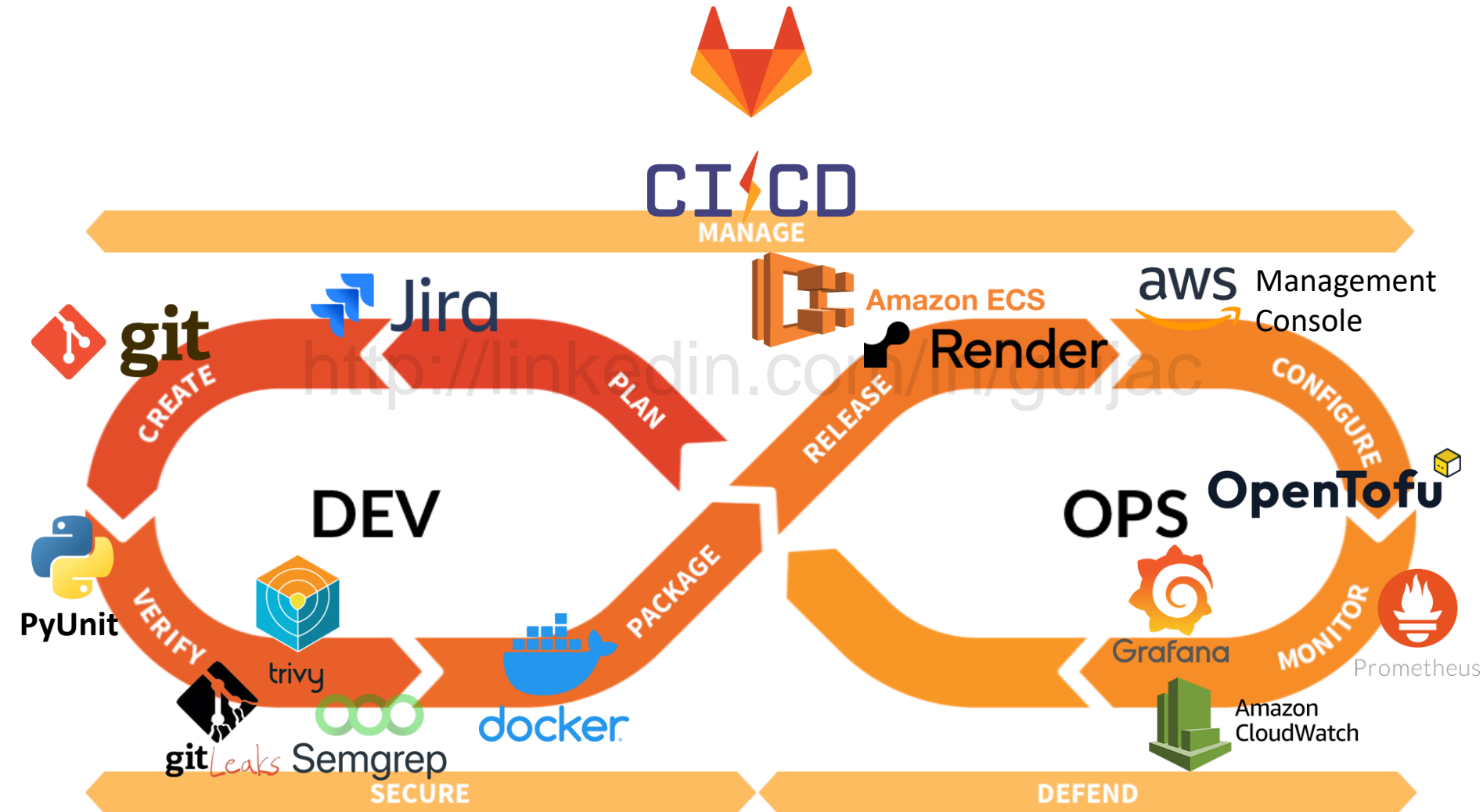
- **Cultura** na Engenharia de Software que aproxima as equipes de desenvolvimento (Dev) e operadoras/administradoras de sistemas (Ops);
- Visa melhorar a comunicação dos dois papéis dentro de um projeto e defender a **automação** e monitoramento de um software.

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



Fonte: [The industry is moving towards a single application for the DevOps lifecycle | GitLab](#)

# Revisitando DevOps/DevSecOps

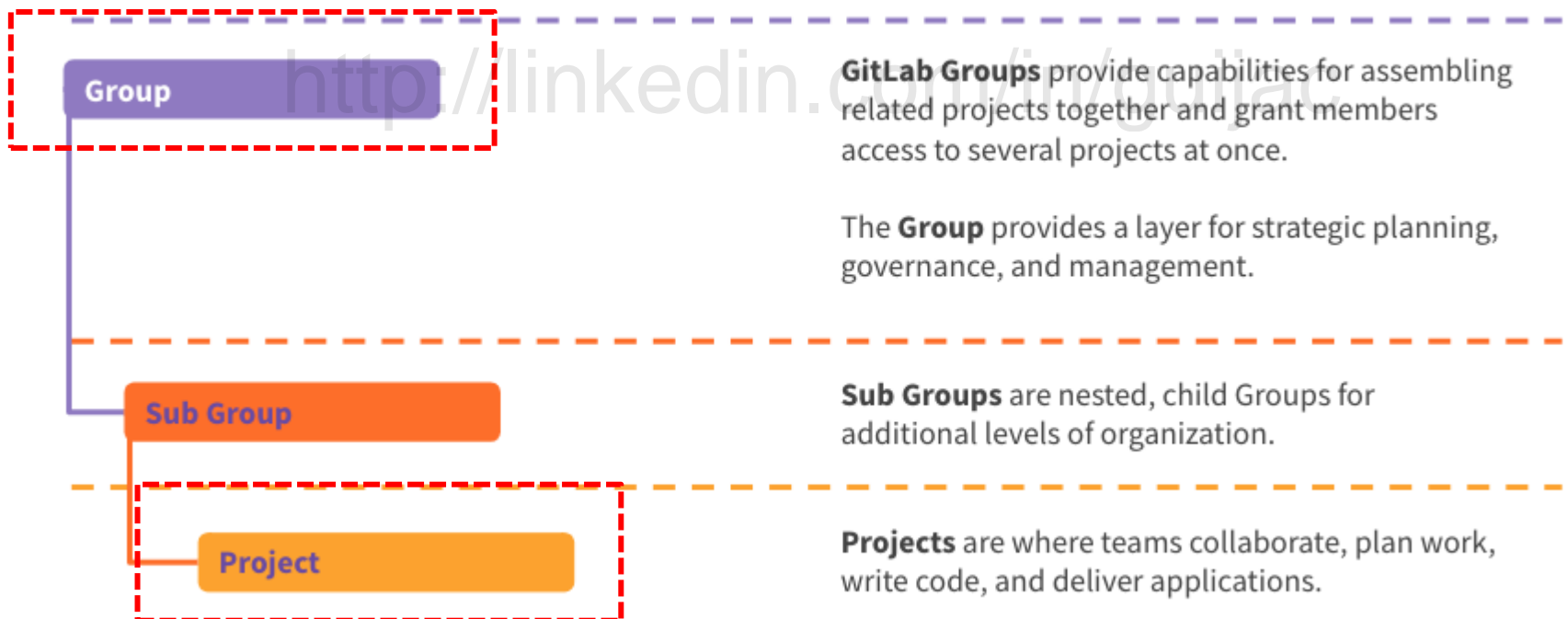


Fonte: [The industry is moving towards a single application for the DevOps lifecycle | GitLab](https://www.gitlab.com/topics/devops/devops-lifecycle)

# Revisitando Estrutura Gitlab

- Um recurso criado em um grupo do Gitlab será herdado pelos projetos abaixo dele.

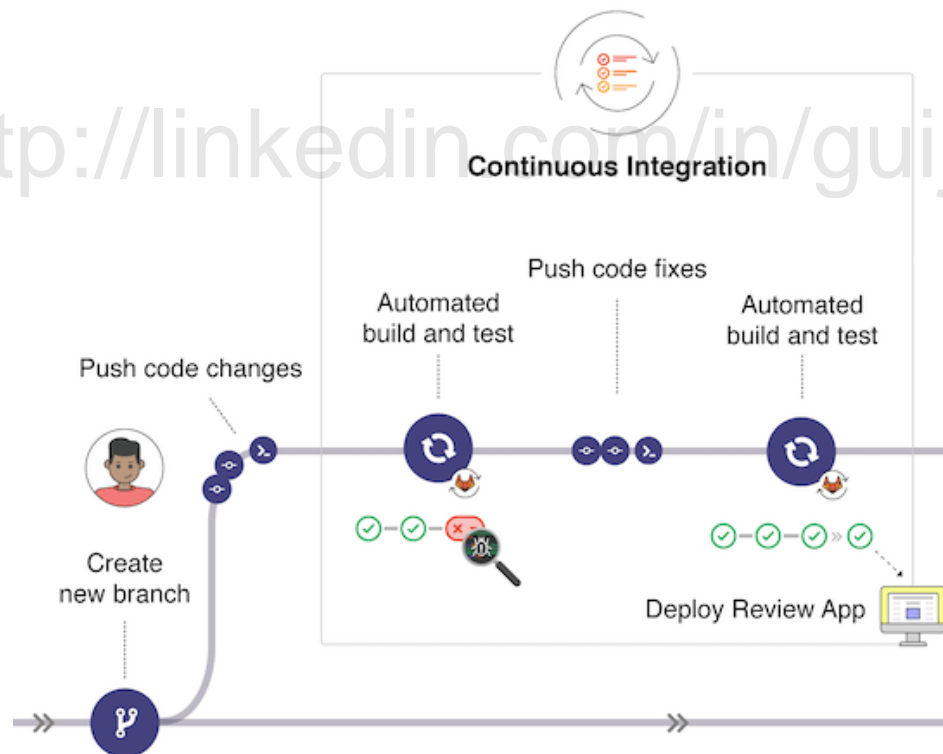
Trabalharemos com grupos e projetos.



Fonte: [How to use GitLab for Agile portfolio planning and project management](https://about.gitlab.com/topics/agile-project-management/) | GitLab

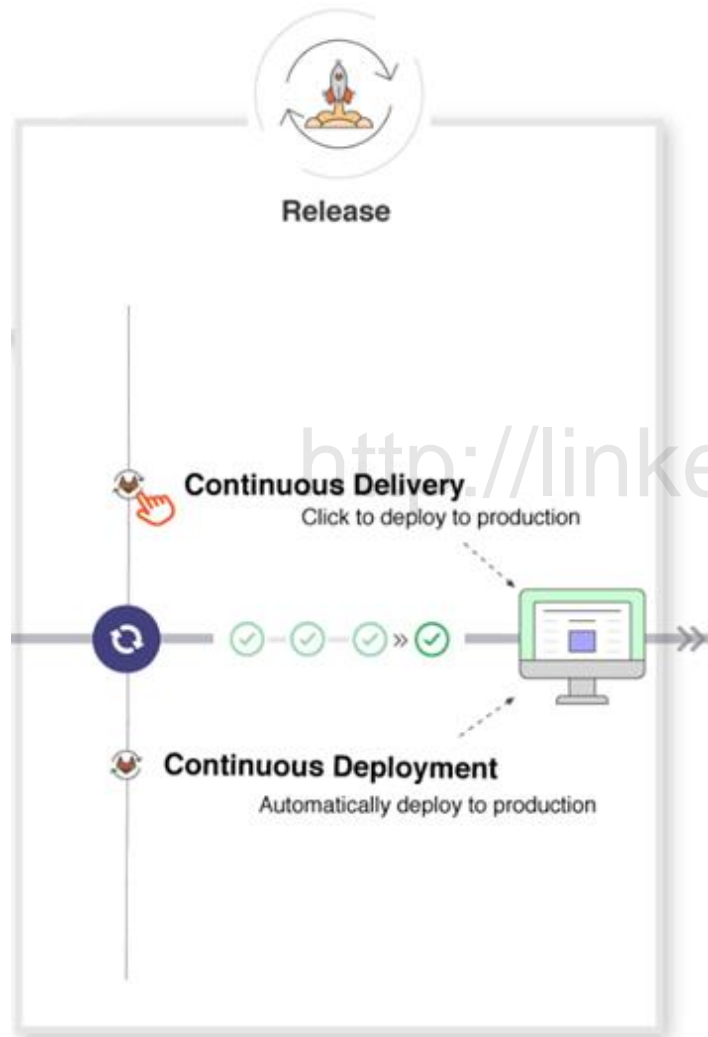
# Continuous Integration (CI)

- A partir da entrega de um novo código em um repositório (**git push**) ocorre a execução de **testes automatizados**, além de outras automações, por exemplo
- Os testes e a geração do pacote (*build*) são realizados através de um “**Servidor de CI**”, visando a **melhoria na qualidade** do produto.



Fonte: [CI/CD concepts](#) | [GitLab](#)

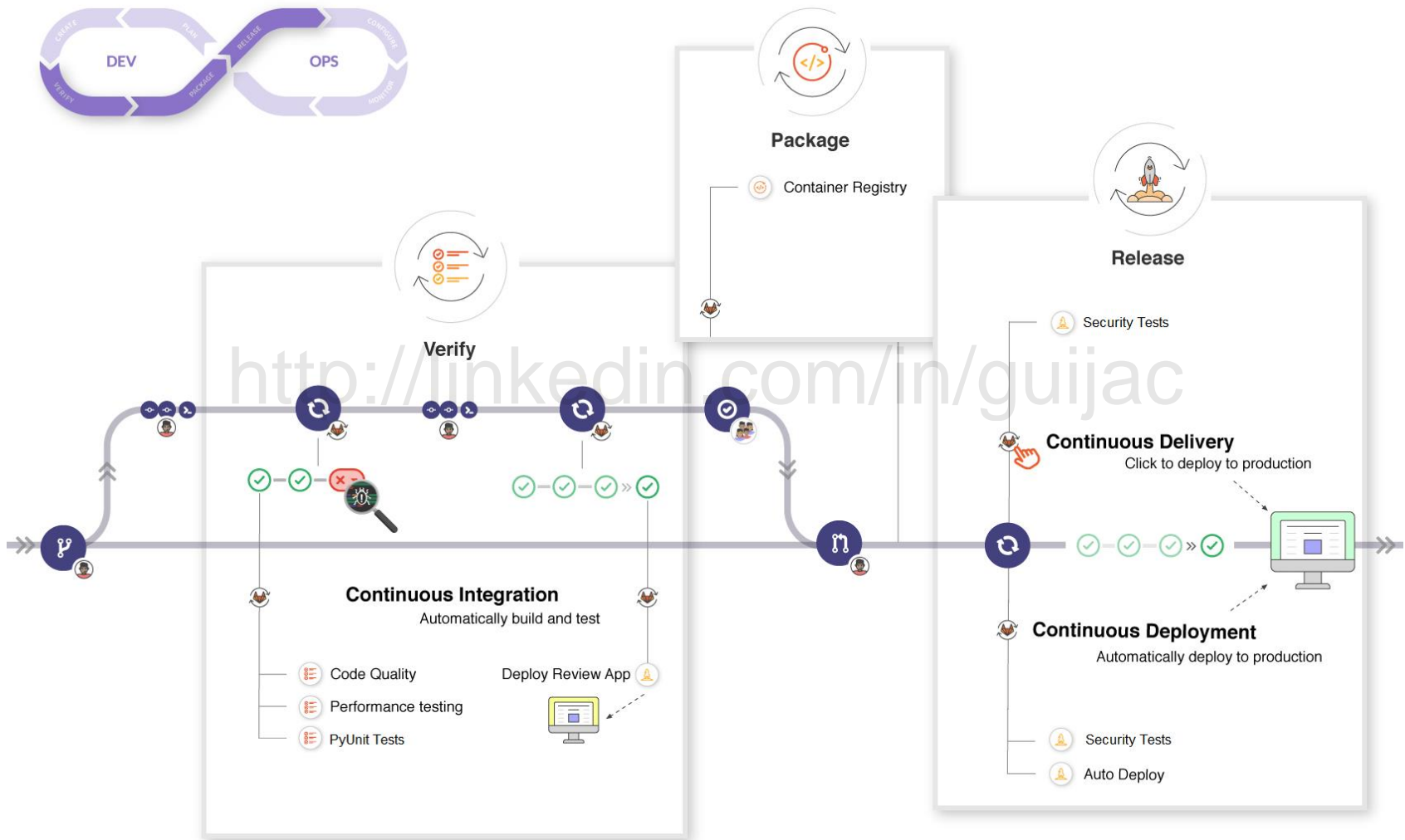
# Continuous Delivery e Deployment (CD)



- Nesta etapa o código alterado é **implantado nos ambientes** do produto, geralmente divididos em Desenvolvimento, Homologação e Produção:
- Implantação Manual = Continuous Delivery;
- Implantação Automática = Continuous Deployment.
- Podem ser executados testes mais abrangentes (análise estática e/ou dinâmica com foco em segurança, por exemplo).

Fonte: Adaptado de [CI/CD concepts](#) | [GitLab](#)

# Juntando as Peças: CI+CD

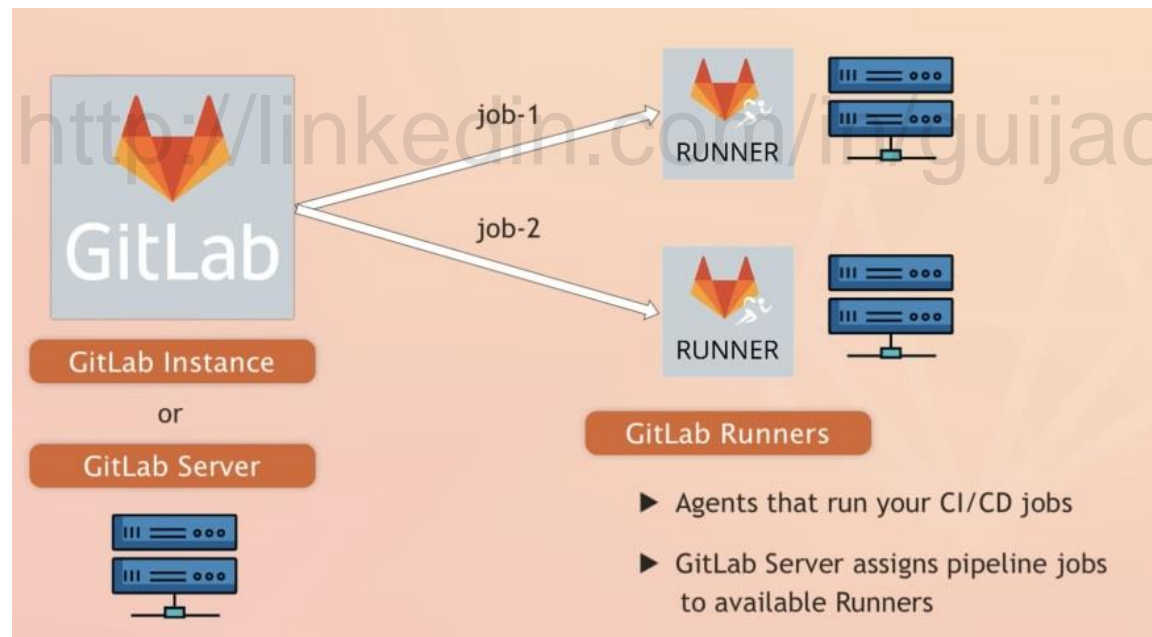


Fonte: Adaptado de [CI/CD concepts](#) | [GitLab](#)



# Gitlab Runner: Definição

- Recurso do Gitlab que permite a execução de *pipelines*<sup>1</sup> para um CI/CD da sua própria aplicação;
- Permite a **orquestração** de todas as etapas para a implantação da sua aplicação: build, testes e deploy.



Fonte: [GitLab CI/CD for Beginners \[FREE Course\] - DEV Community](#)

<sup>1</sup> Termo derivado de Arquitetura de Computadores, consiste na técnica de *hardware* que permite que a CPU realize a busca de uma ou mais instruções além da próxima a ser executada, criando uma fila de atividades, agrupadas por estágios (*stages*).

# Gitlab Runner: O Arquivo .gitlab-ci.yml

- Arquivo YAML (YAML Ain't Markup Language) que especifica as instruções para a execução do *pipeline* através do Gitlab Runner;
- O arquivo deve ser salvo na raiz do projeto sempre com o nome .gitlab-ci.yml;
- Possui uma série de palavras-chave para diversas situações de uso, como acionamento em *branches* específicas;
- A sintaxe completa do arquivo pode ser consultada em sua [documentação de referência](#) e o preenchimento pode ser validado no [Editor de Pipelines do Gitlab](#) ou através de uma extensão do [VSCode](#).

```
test-job:
  stage: test
  script:
    - echo "This job tests something"
```

Arquivo .gitlab-ci.yml que executa um stage chamado "test-job" exibindo o valor informado a partir do comando "echo" dentro da interface gráfica do Gitlab (menu CI/CD > Pipelines > Jobs).

# Gitlab Runner: O Arquivo .gitlab-ci.yml

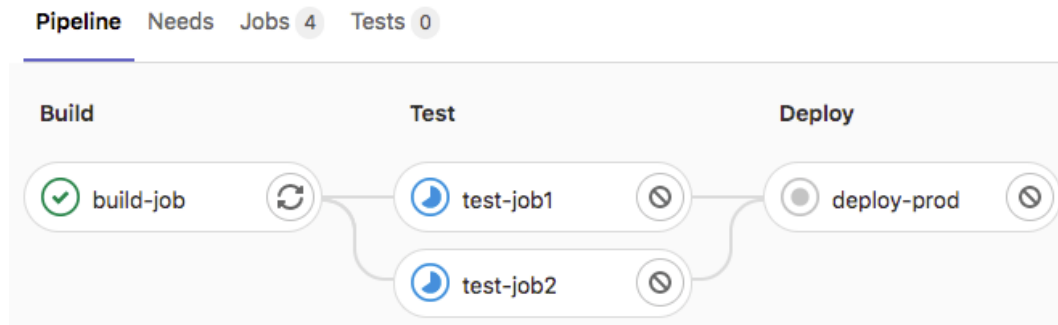
```
build-job:
  stage: build
  script:
    - echo "Hello, $GITLAB_USER_LOGIN!"

test-job1:
  stage: test
  script:
    - echo "This job tests something"

test-job2:
  stage: test
  script:
    - echo "This job tests something, but takes more time than test-job1."
    - echo "After the echo commands complete, it runs the sleep command for 20 seconds"
    - echo "which simulates a test that runs 20 seconds longer than test-job1"
    - sleep 20

deploy-prod:
  stage: deploy
  script:
    - echo "This job deploys something from the $CI_COMMIT_BRANCH branch."
```

Arquivo .gitlab-ci.yml com quatro *stages*, também realizando apenas a impressão dos comandos definidos dentro do bloco "script" na interface gráfica do Gitlab.



# Github Actions – O Arquivo YAML

```
name: GitHub Actions Demo

run-name: ${{GITHUB_ACTOR}} is testing out GitHub Actions 🚀

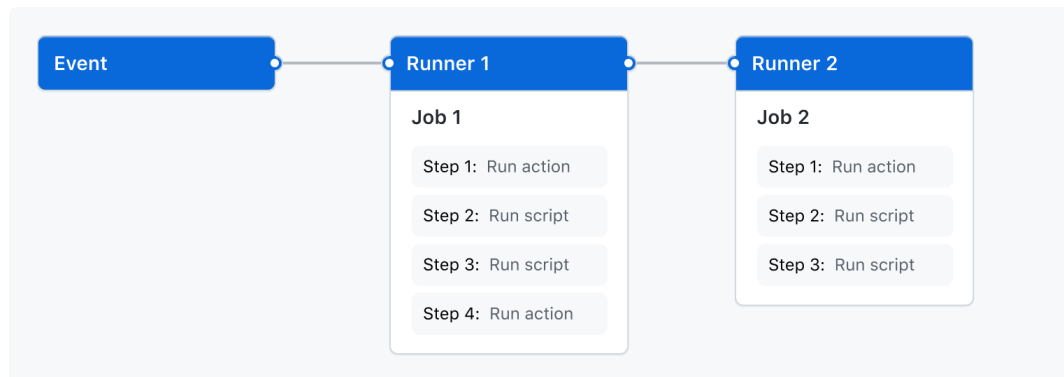
on: [push]

jobs:
  build-job:
    runs-on: ubuntu-latest
    steps:
      - run: echo "Hello, ${{GITHUB_ACTOR}}"

  test-job1:
    runs-on: ubuntu-latest
    steps:
      - run: echo "This job tests something"

  deploy:
    runs-on: ubuntu-latest
    steps:
      - run: echo "This job deploys something from the ${{GITHUB_REF}} branch."
```

Arquivo YAML para Github Actions, o arquivo deve ser salvo sempre em um diretório chamado `.github/workflows`.



# Referências Bibliográficas

---

FOWLER, Martin. **Continuous Integration**. Disponível em <https://martinfowler.com/articles/continuousIntegration.html>. Acesso em 08 mar 2025;

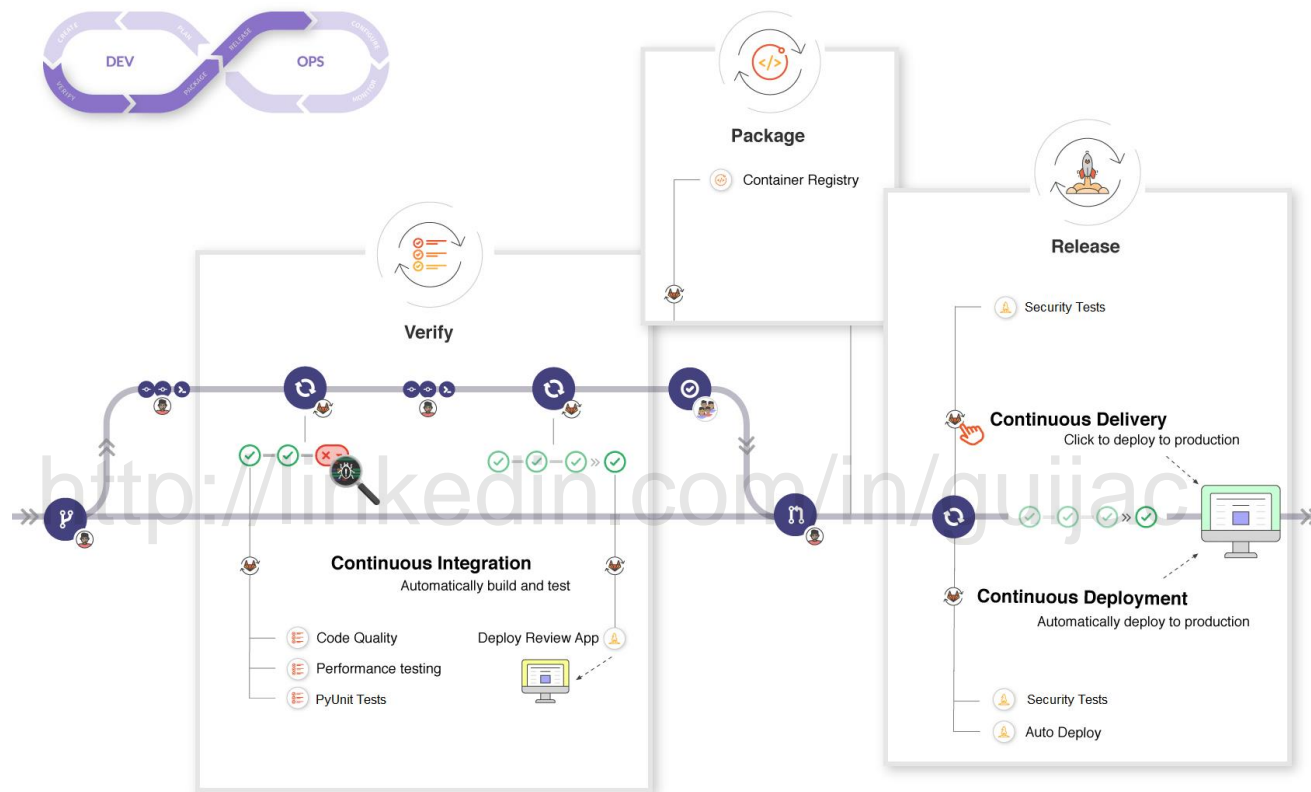
GITHUB. **Workflow syntax for GitHub Actions**. Disponível em <https://docs.github.com/en/actions/using-workflows/workflow-syntax-for-github-actions>. Acesso em 08 mar 2025;

GITLAB. **CI/CD concepts**. Disponível em <https://docs.gitlab.com/ee/ci/introduction/>. Acesso em 08 mar 2025;

HUMBLE, Jez; FARLEY, David. **Continuous delivery: reliable software releases through build, test, and deployment automation**. Pearson Education, 2010;

KIM, Gene et al. **The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations**. IT Revolution, 2021.

# Por hoje (de teoria!) é só!



Fonte: Adaptado de [CI/CD concepts](#) | [GitLab](#)

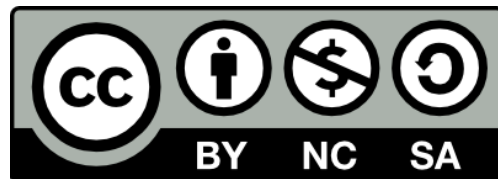
**Prof. Esp. Guilherme Jorge Aragão da Cruz**

 guilherme.cruz@alumni.usp.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.cruz@alumni.usp.br](mailto:guilherme.cruz@alumni.usp.br)

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