Fundamentos de DevOps

Aula 12 - Automação de Builds e Contêineres na Nuvem

- Laboratório

Prof. Esp. Guilherme Jorge Aragão da Cruz

guilherme.cruz@alumni.usp.br

linkedin.com/in/guijac

Laboratório

1. Criar um novo projeto em seu Gitlab (ou utilizar o projeto da ADO-03) para desenvolvimento da **etapa de build** em um pipeline;
2. Criar ou editar o arquivo .gitlab-ci.yml com um stage de build;
3. Implementar o before_script para autenticação segura no Gitlab Container Registry;
4. Aplicar boas práticas de versionamento no build da imagem Docker;
5. Publicar a imagem no GitLab Registry.

.gitlab-ci.yml com um stage de build

- Criar o arquivo .gitlab-ci.yml com a stage build, imagem docker:24.0.5 e o serviço docker:dind;
- O modo Docker-in-Docker (DinD) permite que o runner execute comandos Docker dentro de um contêiner;
- Exemplo:


```yaml
build:
  stage: build
  image: docker:24.0.5
  services:
    - docker:24.0.5-dind
```

Implementar before_script

- O before_script executa comandos antes do build.
- O login deve utilizar variáveis protegidas do GitLab, evitando expor senhas em linha de comando;
- Isto permite um *push* seguro das imagens geradas;
- Exemplo:


```
before_script:
      - echo "$CI_REGISTRY_PASSWORD" | docker login -u "$CI_REGISTRY_USER" --password-stdin "$CI_REGISTRY"
```

Aplicar boas práticas de versionamento

- Estas variáveis automatizam o controle de versões:
  - $CI_COMMIT_REF_SLUG → identifica a branch ou tag atual;
  - $CI_COMMIT_SHA → hash único do commit, garantindo que cada build seja identificado de forma imutável e auditável.
- Tais práticas facilitam rastreabilidade e *rollbacks*.
- Exemplo:


```
script:
  - docker build -t "$CI_REGISTRY_IMAGE/$CI_COMMIT_REF_SLUG:$CI_COMMIT_SHA" .
```

Publicar a imagem no GitLab Registry

- Ao publicar a imagem, o pipeline entrega um artefato versionado e pronto para deploy;
- O uso das tags latest e $CI_COMMIT_SHA garante histórico e consistência entre builds;
- Após o push, valide o resultado em seu Gitlab Registry;
- Exemplo:


```
- docker push "$CI_REGISTRY_IMAGE/$CI_COMMIT_REF_SLUG:$CI_COMMIT_SHA"
- docker push "$CI_REGISTRY_IMAGE/$CI_COMMIT_REF_SLUG:latest"
```

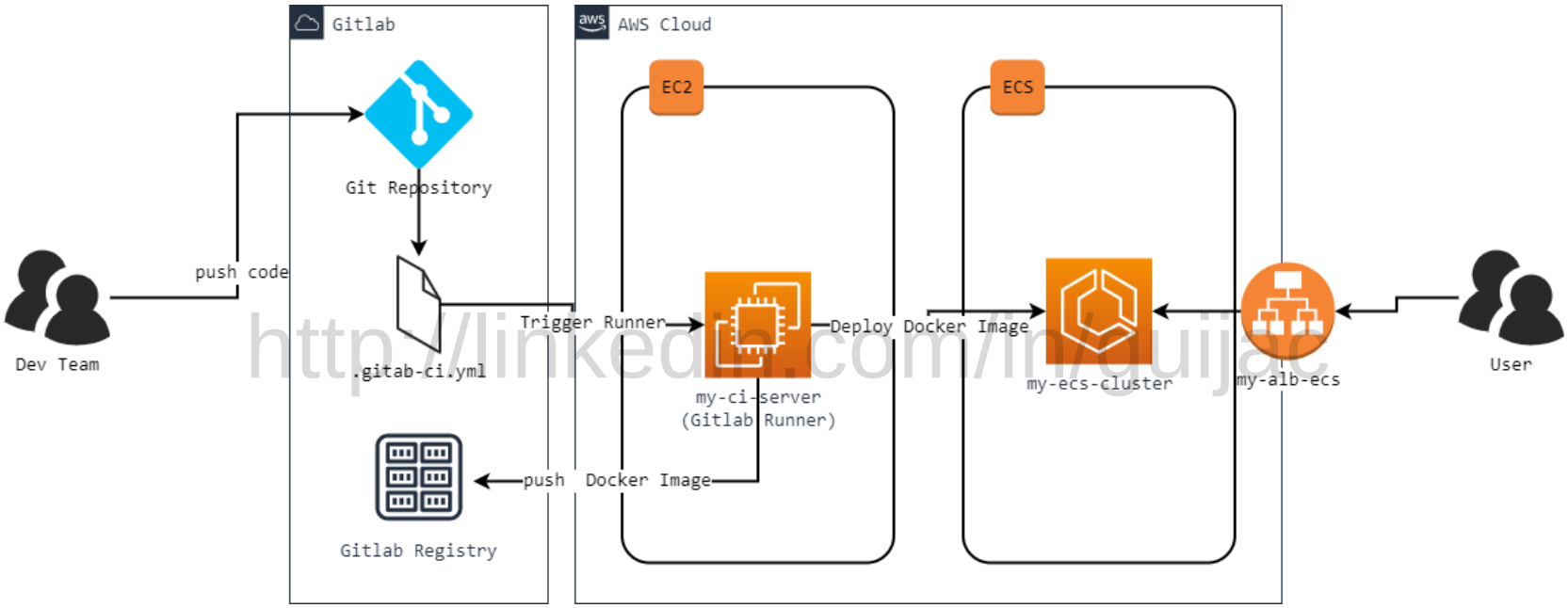
Referências Bibliográficas

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 (agora sim!) é só!



Fonte: Elaboração própria.

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-CompartilhaIgual 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

linkedin.com/in/guijac