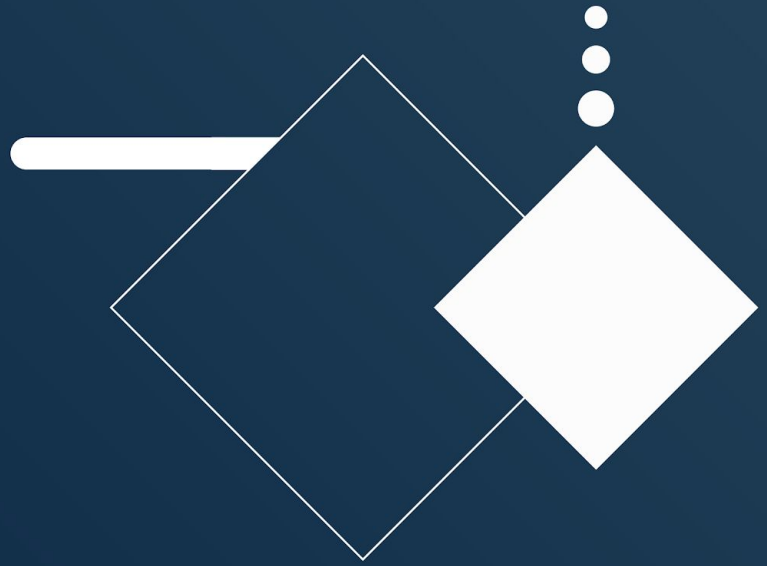
LAPPIS

Tais
Pipeline CI/CD

Quem sou eu?



Quem sou eu?

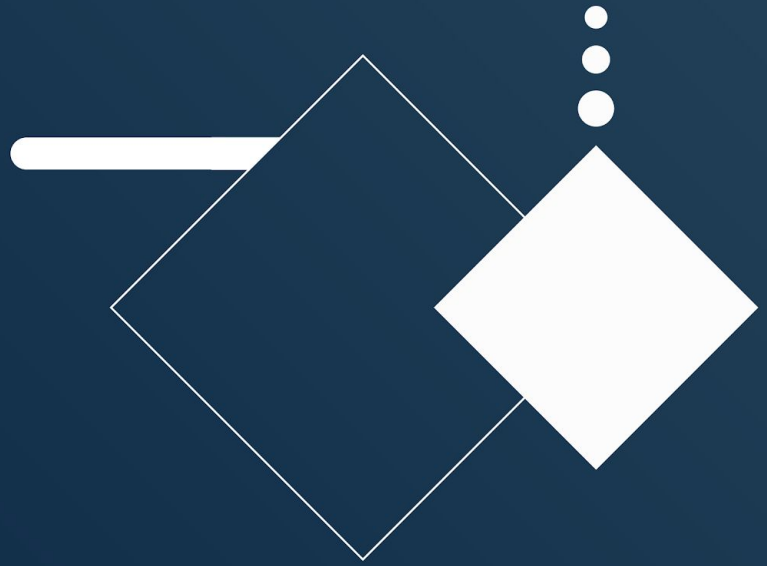
- Engenharia de Software (UnB)
- LAPPIS há 2 anos
  - Desenvolvimento Back-End
  - Infraestrutura
  - DevOps

LAPPIS + Ministério da Cidadania

- SALIC-ML
  - Apoio ao processo de análise de prestação de contas
  - Data Science em ambiente de produção
- Tais + BotFlow
  - Acesso à informação via assistente virtual
  - Validação e criação de modelos durante o pipeline CI

“Como levar, **continuamente**, um assistente virtual do markdown à produção?”

Ferramentas



Ferramentas

- Docker + DockerHub
  - Infraestrutura como código
  - Isolamento dos serviços
  - Distribuição facilitada
- GitLabCI
  - Integração Contínua
  - Registry Docker (temporário)

GitLabCI

- Plataforma de Integração Contínua
  - Jenkins, TravisCI, CircleCI...
- Configuração via **.gitlab-ci.yml**
  - **Stages**
  - **Jobs**
  - **Images**
  - **Scripts**

GitLabCI

- **Stages** ou Etapas
  - Abstração de processos sequenciais
  - Compostos por **jobs**
  - Exemplos:
    - 1) Test Stage
      - Linters (verificação da sintaxe)
      - Unit test
    - 2) Deploy Stage
      - Push to Heroku
      - Push to AWS

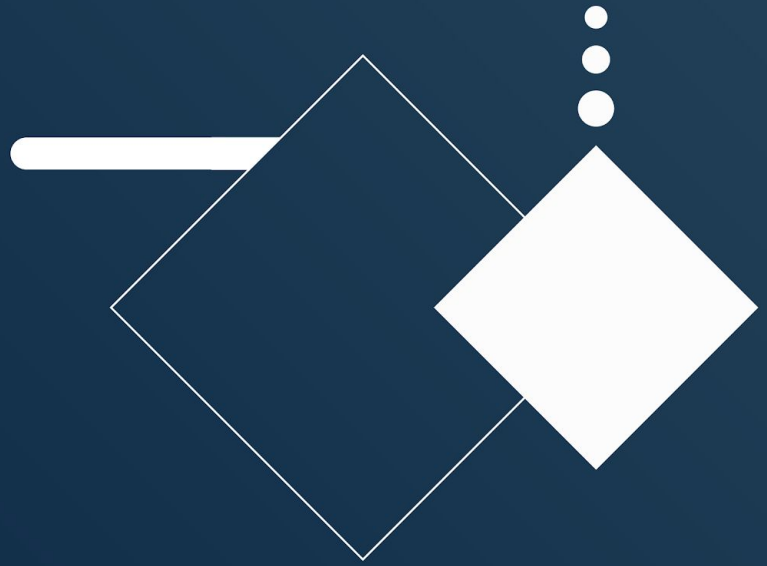
GitLabCI

- **Jobs** ou Tarefas
  - Abstração de processos paralelos
  - Ambiente definido por uma **image**
  - Ações definidas por **scripts**
  - Exemplos:
    - Push to Heroku
      - Image: ubuntu:latest
      - Script:
        - git push heroku master

GitLabCI

- **Test Stage**:
  - **check syntax**:
    - Image: python:3.7
    - Script:
      - cd /myproject
      - flake8
- **Deploy Stage**:
  - **deploy to heroku**:
    - Image: ubuntu:latest
    - Script:
      - cd /myproject
      - ./deploy_heroku.sh

Pipeline Tais V1



Pipeline Tais V1

- Stages e Jobs
  - build
    - build stable
    - build latest
  - deploy
    - deploy to development
    - deploy to production

```
1  stages:
2      - build
3      - deploy
```

```
5  variables:
6    BOT_LATEST_IMAGE: $CI_REGISTRY_IMAGE/bot:latest
7    ACTIONS_LATEST_IMAGE: $CI_REGISTRY_IMAGE/actions:latest
8    WEB_LATEST_IMAGE: $CI_REGISTRY_IMAGE/web:latest
9
10   BOT_STABLE_IMAGE: $CI_REGISTRY_IMAGE/bot:stable
11   BOT_LATEST_IMAGE: $CI_REGISTRY_IMAGE/actions:latest
12   WEB_STABLE_IMAGE: $CI_REGISTRY_IMAGE/web:stable
```

Boa prática: uso de variáveis de ambiente

```
14  build stable:
15    image: docker:latest
16    stage: build
17    services:
18      - docker:dind
19    script:
20      - docker login -u "gitlab-ci-token" -p "$CI_JOB_TOKEN" $CI_REGISTRY
21
22      - docker build -f docker/bot.Dockerfile -t $BOT_STABLE_IMAGE .
23      - docker push $BOT_STABLE_IMAGE
24
25      - docker build -f docker/actions.Dockerfile -t $ACTIONS_STABLE_IMAGE .
26      - docker push $ACTIONS_STABLE_IMAGE
27
28      - docker build -f docker/web/Dockerfile -t $WEB_STABLE_IMAGE .
29      - docker push $WEB_STABLE_IMAGE
30    only:
31      - tags
32    environment: production
33    tags:
34      - docker
```

build stable

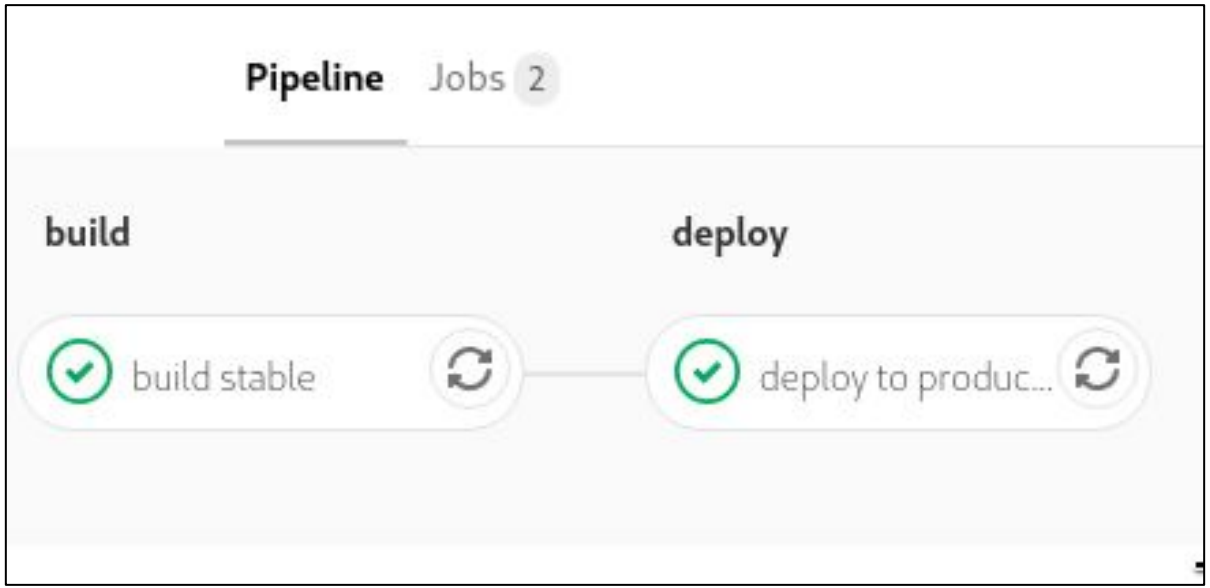
“Como rodar docker dentro de um container docker?”

```
14  build stable:
15    image: docker:latest
16    stage: build
17    services:
18      - docker:dind
19    script:
20      - docker login -u "gitlab-ci-token" -p "$CI_JOB_TOKEN" $CI_REGISTRY
21
22      - docker build -f docker/bot.Dockerfile -t $BOT_STABLE_IMAGE .
23      - docker push $BOT_STABLE_IMAGE
24
25      - docker build -f docker/actions.Dockerfile -t $ACTIONS_STABLE_IMAGE .
26      - docker push $ACTIONS_STABLE_IMAGE
27
28      - docker build -f docker/web/Dockerfile -t $WEB_STABLE_IMAGE .
29      - docker push $WEB_STABLE_IMAGE
30    only:
31      - tags
32    environment: production
33    tags:
34      - docker
```

build stable

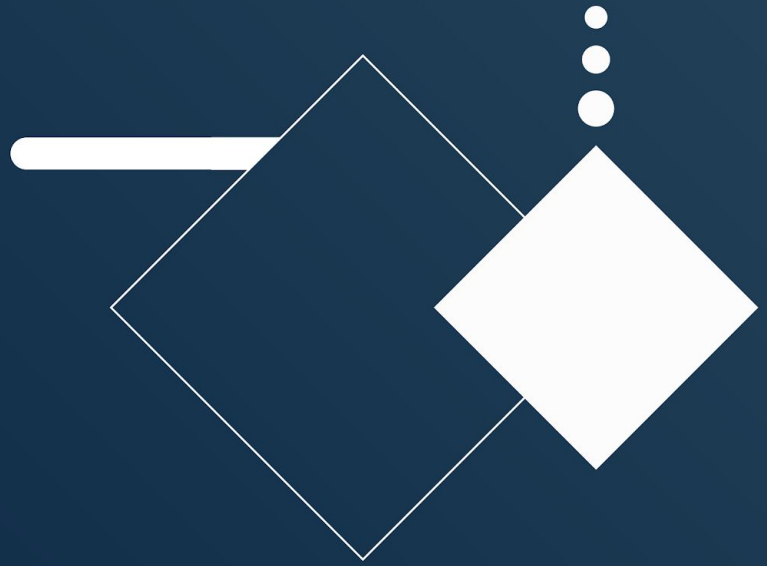
```
74  deploy_to_development:
75    image: cdrx/rancher-gitlab-deploy
76    stage: deploy
77    services:
78      - docker:dind
79    script:
80      - upgrade --environment Development --stack TAIS --service bot --new-image $BOT_LATEST_IMAGE --debug
81      - upgrade --environment Development --stack TAIS --service actions --new-image $actions_LATEST_IMAGE --debug
82      - upgrade --environment Development --stack TAIS --service web --new-image $WEB_LATEST_IMAGE --debug
83    only:
84      - /master/
85    environment: development
86    tags:
87      - docker
```

deploy to development



pipeline branch tags

Pipeline Tais V2



Pipeline Tais V2

- Mudanças arquiteturais
  - Separação das imagens Docker
    - Requirements
      - pip install … && apt install ...
    - Coach
      - FROM requirements:latest
      - Modelo treinado empacotado
    - Bot
      - FROM coach
      - Configurações de conexão do bot

Pipeline Tais V2

- Mudanças no processo
  - Testes automatizados
    - Para testar, é preciso treinar
  - Validação do formato das utters, intents e stories
    - Similar a um linter

Pipeline Tais V2

- Stages e Jobs
  - test style
    - Linter
  - validate format
    - Validação do formato do conteúdo
  - test stories
    - Testes automatizados
  - build requirements
  - build coach
  - build
  - deploy

```
3   stages:
4     - test style
5     - validate format
6     # - test stories
7     - build requirements
8     - build coach
9     - build
10    - deploy
```

```
17  test style:
18    stage: test style
19    script:
20      - pip -V
21      - python -V
22      - pip install -r dev.requirements.txt
23      - flake8 --exclude venv
```

test style

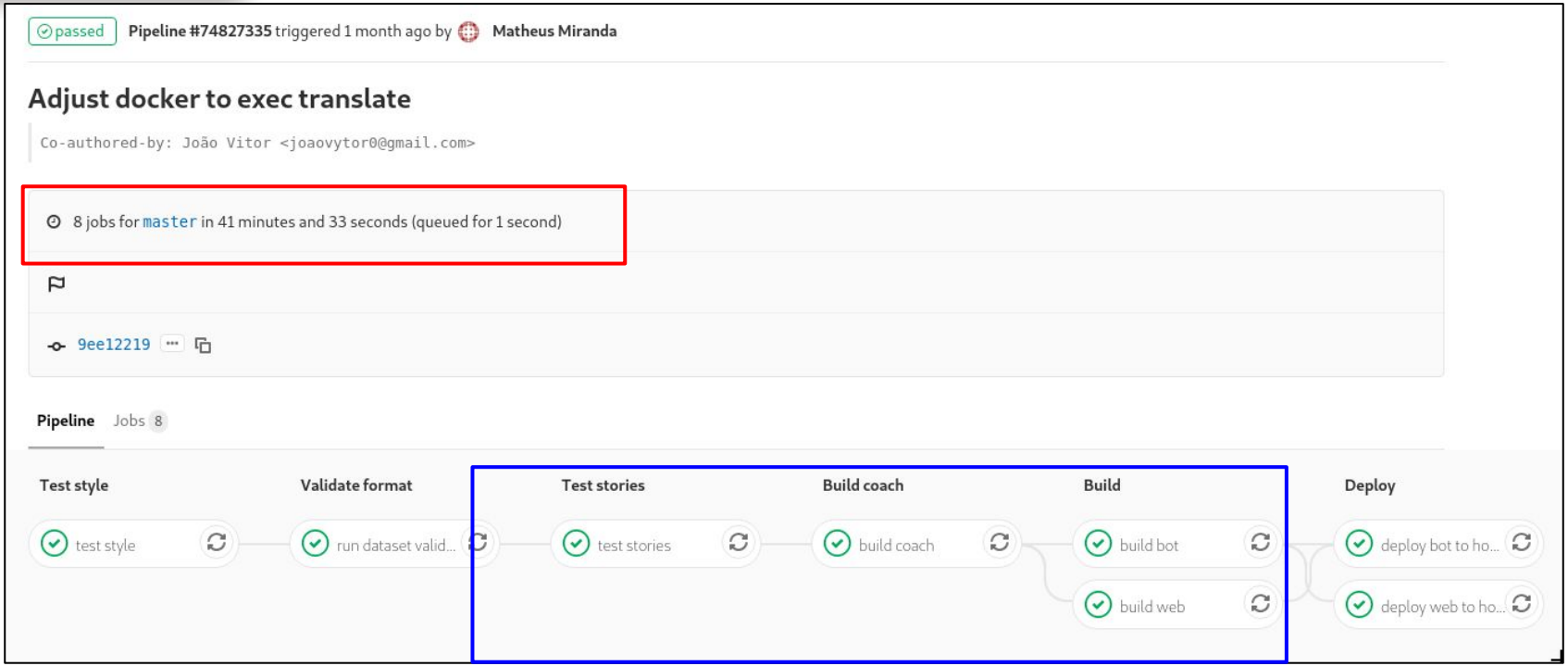
```
25  run dataset validator:
26    stage: validate format
27    image: lappis/coach:latest
28    script:
29      - cd coach/
30      - make run-validator
```

run dataset validator

```
32  #test stories:
33  #  stage: test stories
34  #  image: docker
35  #  tags:
36  #    - docker
37  #  services:
38  #    - docker:dind
39  #  script:
40  #    - docker build . -f docker/bot/coach.Dockerfile -t lappis/coach:latest > /dev/null 2>&1
41  #    - docker build -f docker/bot/bot.Dockerfile -t lappis/bot:latest .
42  #    - docker run --rm lappis/bot:latest make test-stories
```

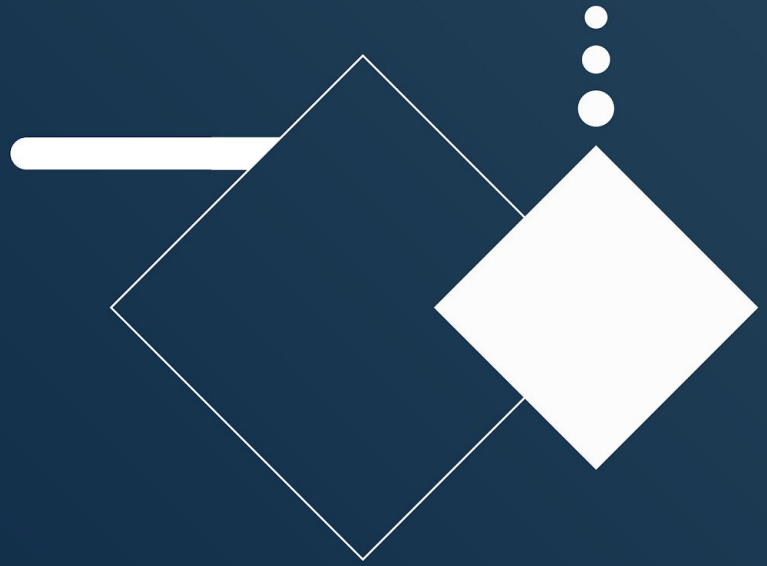
test stories

Para **testar**, é necessário **buildar**!



pipeline com os testes

Pipeline Tais V3



Pipeline Tais V3

- Otimização de pipeline
  - Start and Scaling Devops in the Enterprise (Gary Gruver)
    - Problemas no processo
      - Gastos desnecessários
    - Problemas técnicos
      - Implementação inadequada

Pipeline Tais V3

- Otimização de pipeline
  - Start and Scaling Devops in the Enterprise (Gary Gruver)
    - Problemas no processo
      - Gastos desnecessários
    - Problemas técnicos
      - Implementação inadequada*

* Possível implementação inadequada durante a resolução do problema

Pipeline Tais V3

- Problema
  - As builds do coach e do bot estão sendo feitas duas vezes
- Pergunta
  - Como reaproveitar uma build em outro processo do pipeline?

Pipeline Tais V3

- Problema
  - As builds do coach e do bot estão sendo feitas duas vezes
- Pergunta
  - Como reaproveitar uma build em outro processo do pipeline?
- Solução
  - Docker Registry

Pipeline Tais V3

- Stages e jobs
  - build requirements
  - build temp images
  - test style
  - validate format
  - test stories
  - push
    - push coach
    - push bot
    - push web
  - deploy

```
8   stages:
9     - build requirements
10    - build temp images
11    - test style
12    - validate format
13    - test stories
14    - push
15    - deploy
```

```
37  build requirements:
38    stage: build requirements
39    image: docker
40    tags:
41      - docker
42    services:
43      - docker:dind
44    script:
45      - docker login -u $DOCKERHUB_USER -p $DOCKERHUB_PASSWORD
46      - cd ./docker/bot
47      - ./build-base.sh publish
48    only:
49      refs:
50        - master
51      changes:
52        - ./docker/bot/requirements.txt
53    environment: homolog
```

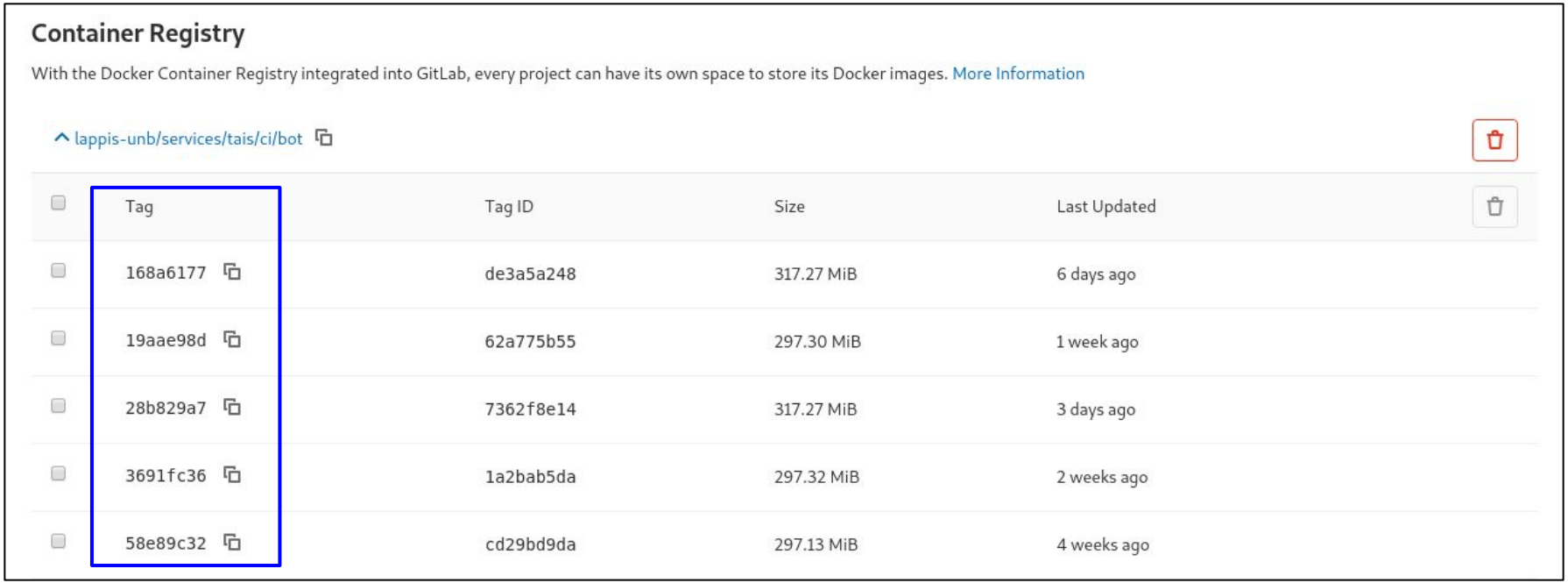
build requirements

```
55  build temp images:
56    stage: build temp images
57    image: docker
58    tags:
59      - docker
60    services:
61      - docker:dind
62    script:
63     - docker build . -f docker/bot/coach.Dockerfile -t lappis/coach:tais > /dev/null 2>&1 # Builds lappis/coach:tais
64     - docker build -f docker/bot/bot.Dockerfile -t $BOT_TEMP_IMAGE . # It uses lappis/coach:tais built locally
65     - docker login -u "gitlab-ci-token" -p "$CI_JOB_TOKEN" $CI_REGISTRY
66     - docker tag lappis/coach:tais $COACH_TEMP_IMAGE # Retags lappis/coach:tag to temp image
67     - docker push $COACH_TEMP_IMAGE # Pushes temp image to the gitlab registry
68     - docker push $BOT_TEMP_IMAGE
69    only:
70      refs:
71        - pipeline_refactoring
72        - master
73    environment: homolog
```

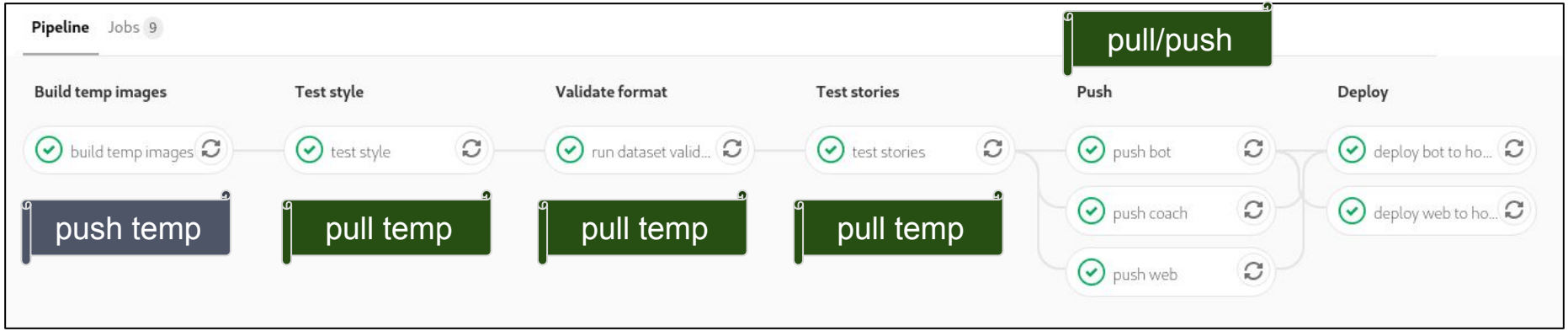
build temp images

```
3  variables:
4    COACH_TEMP_IMAGE: $CI_REGISTRY_IMAGE/ci/coach:$CI_COMMIT_SHORT_SHA
5    BOT_TEMP_IMAGE: $CI_REGISTRY_IMAGE/ci/bot:$CI_COMMIT_SHORT_SHA
```

temp variables



temp registry com tags por commit



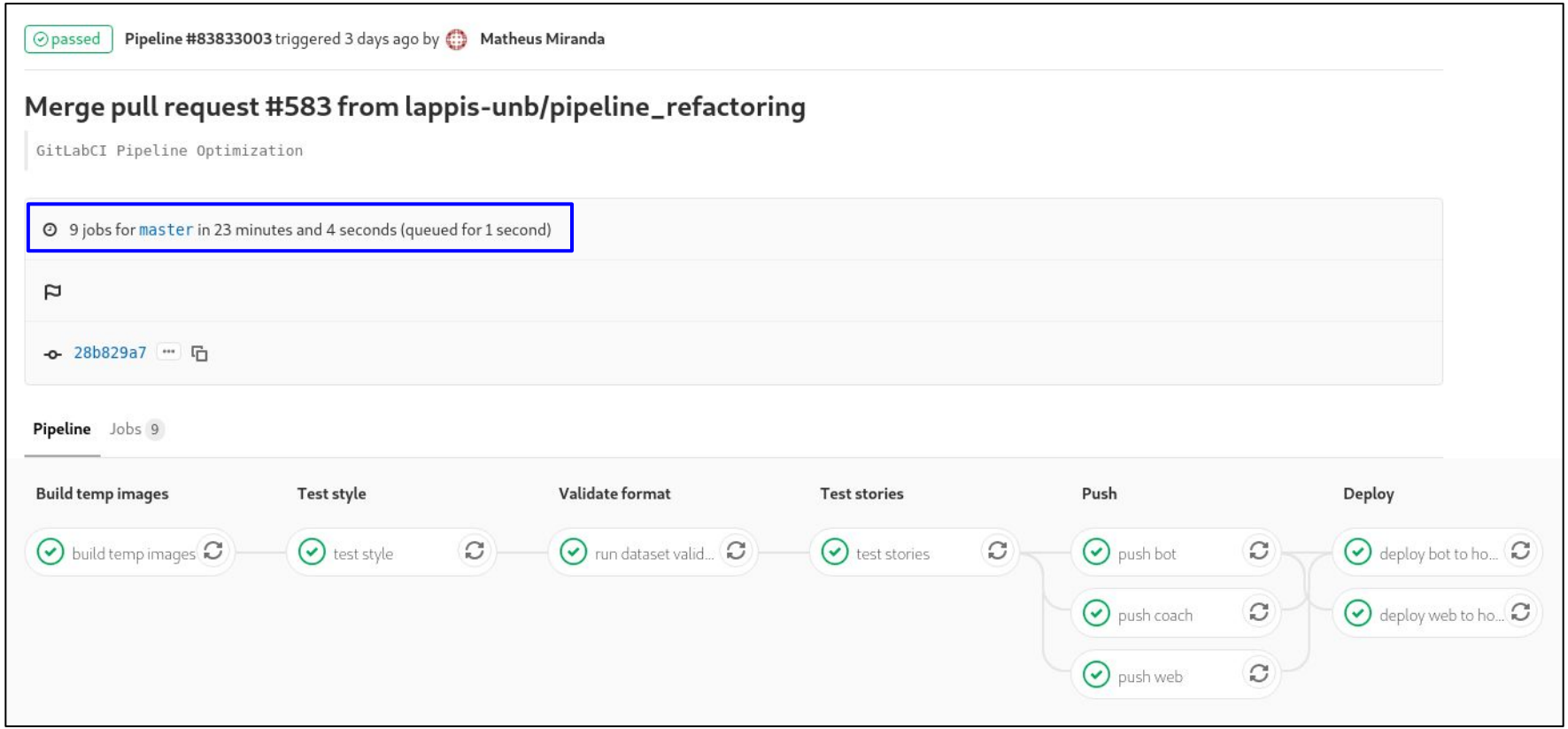
pipeline otimizado

```
75  test stories:
76   stage: test stories
77   image: docker
78   tags:
79     - docker
80   services:
81     - docker:dind
82   script:
83     - docker pull $BOT_TEMP_IMAGE
84     - docker run --rm $BOT_TEMP_IMAGE make test-stories
85
```

test stories

```
86  push coach:
87    stage: push
88    image: docker
89    tags:
90      - docker
91    services:
92      - docker:dind
93    script:
94      - docker login -u $DOCKERHUB_USER -p $DOCKERHUB_PASSWORD
95      - docker pull $COACH_TEMP_IMAGE
96      - docker tag $COACH_TEMP_IMAGE lappis/coach:tais
97      - docker push lappis/coach:tais
98    only:
99      - master
100   environment: homolog
```

push coach



redução do tempo de pipeline

Obrigado!
Dúvidas?

Victor Moura

<Contato>

mouracvictor@gmail.com

<Github>

https://github.com/victorcmoura

<Medium>

http://medium.com/@lappisunbfga