



CI/CD

Python + Rancher + GitLab

Dados

Bruno Emanuel - Analista de Infra IFMA

Autodidata desde 1990

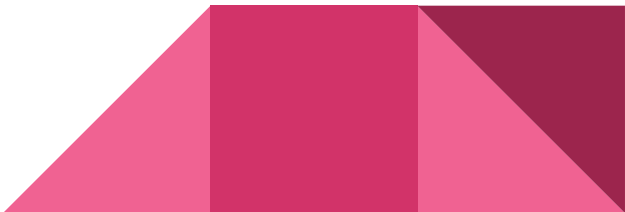
Formado em Sistemas de Informação - CEUMA

Pós em Gerência de Projetos - Fac. Pitágoras

Foco em FOSS

Cultura DevOPS

 @jwalker_pe





ERP Suap

IFRN / IFMA



Python + Django

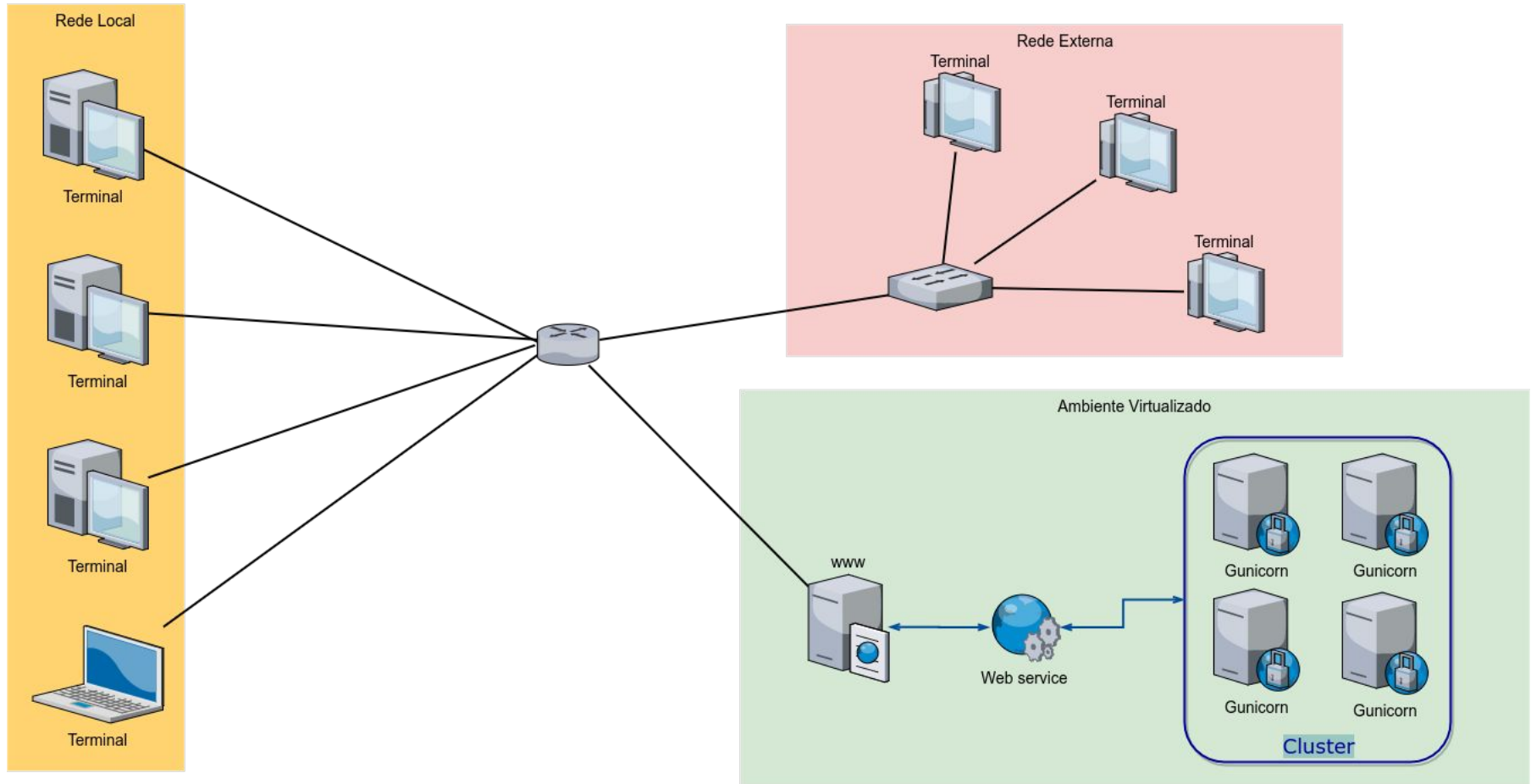
PostgreSQL

Gunicorn / uWSGI

NGinx

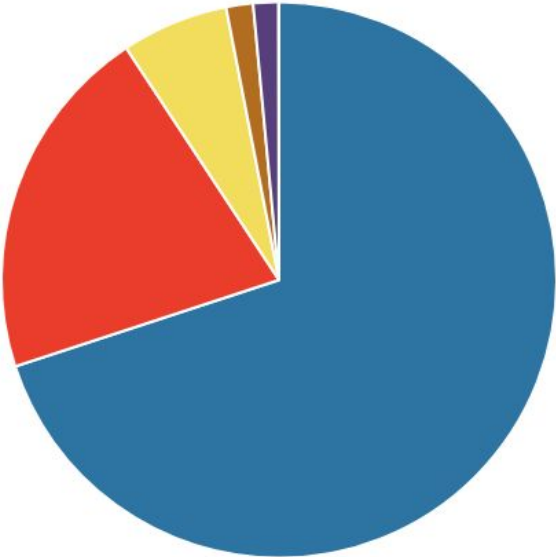
> 40 módulos (dentre eles: edu, processo eletrônico, central de serviços)

Estrutura Anterior



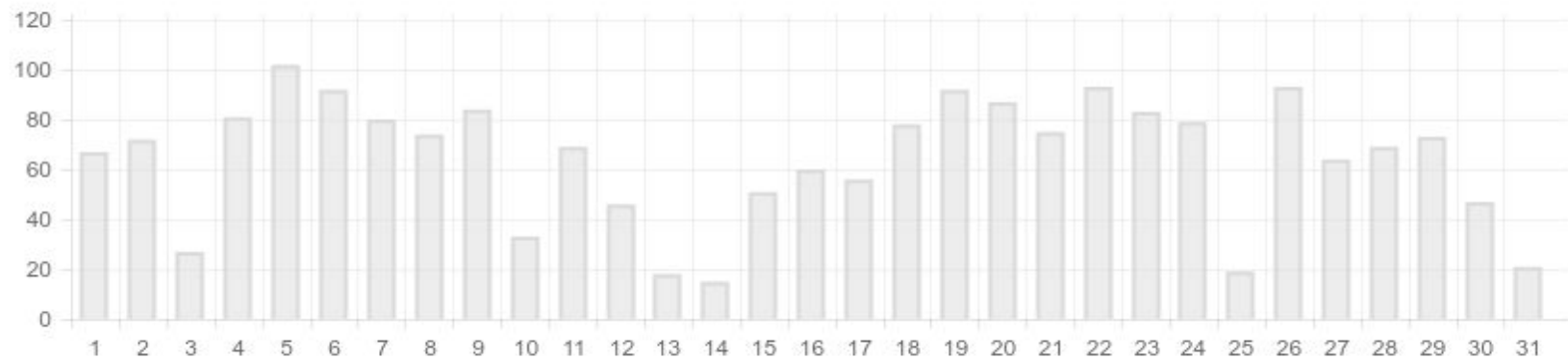
Programming languages used in this repository

Python	69.97 %
HTML	20.76 %
JavaScript	6.22 %
Java	1.57 %
CSS	1.47 %
Shell	0.01 %
PHP	0.01 %

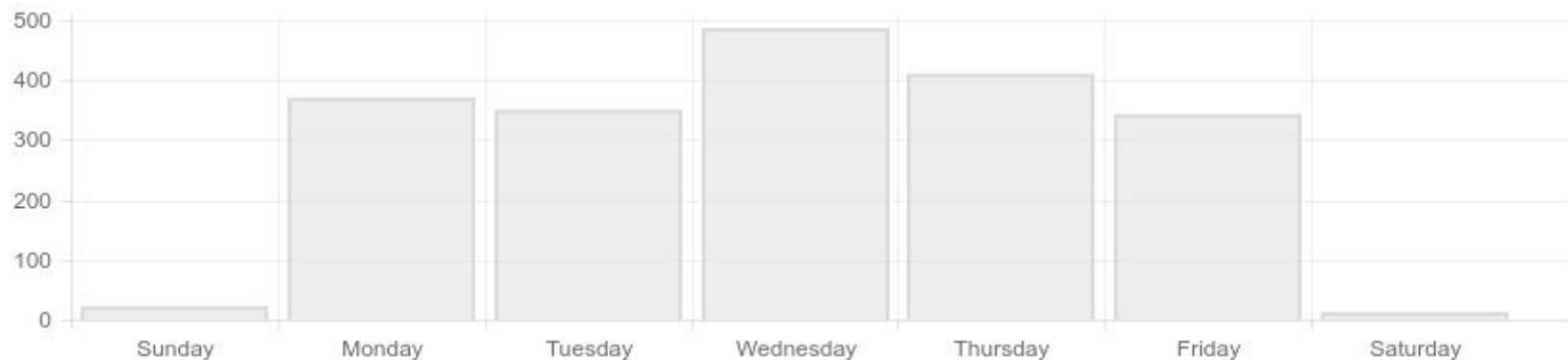


suap

Commits per day of month



Commits per weekday







CI/CD ??

Continuous Integration

Metodologia moderna de desenvolvimento, em que os desenvolvedores envolvidos em um projeto integram seus trabalhos continuamente.

Cada integração é consolidada por uma ferramenta, que inclusive pode executar diversos testes para identificar erros de digitação de códigos, incompatibilidades entre comandos dados por desenvolvedores diferentes etc.



Continuous Deployment



Uma extensão da CI, com o objetivo de minimizar o *lead time* - ou seja, entrega em produção.

Ideia implementada a partir de 2009 e vem se popularizando nos últimos anos.



Como está hoje a estrutura

Atualização Antes

Atualização



GitLab
Merge



```
akshay@akshay-UbPC:~/Mytest$ git add README
akshay@akshay-UbPC:~/Mytest$ git add sample.c
akshay@akshay-UbPC:~/Mytest$ git commit -m "first edit"
[master (root-commit) 82c032f] first edit
 2 files changed, 7 insertions(+)
 create mode 100644 README
 create mode 100644 sample.c
akshay@akshay-UbPC:~/Mytest$ git remote add origin https://github.com/akshaypai/Mytest.git
akshay@akshay-UbPC:~/Mytest$ git push origin master
Username for 'https://github.com': akshaypai
Password for 'https://github.com/akshaypai/Mytest.git':
Counting objects: 4, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 327 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/akshaypai/Mytest.git
 * [new branch]      master -> master
akshay@akshay-UbPC:~/Mytest$
```



Migration



```
akshay@akshay-UbPC:~/Mytest$ git add README
akshay@akshay-UbPC:~/Mytest$ git add sample.c
akshay@akshay-UbPC:~/Mytest$ git commit -m "first edit"
[master (root-commit) 82c032f] first edit
 2 files changed, 7 insertions(+)
 create mode 100644 README
 create mode 100644 sample.c
akshay@akshay-UbPC:~/Mytest$ git remote add origin https://github.com/akshaypai/Mytest.git
akshay@akshay-UbPC:~/Mytest$ git push origin master
Username for 'https://github.com': akshaypai
Password for 'https://github.com/akshaypai/Mytest.git':
Counting objects: 4, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 327 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/akshaypai/Mytest.git
 * [new branch]      master -> master
akshay@akshay-UbPC:~/Mytest$
```



x 4



Problemas

Desvantagens dessa estrutura

- Processo mais demorado
 - Caso de erros desfazer tudo demora mais
 - Manutenção de servidores exponencial
 - Falta de controle das dependências
-

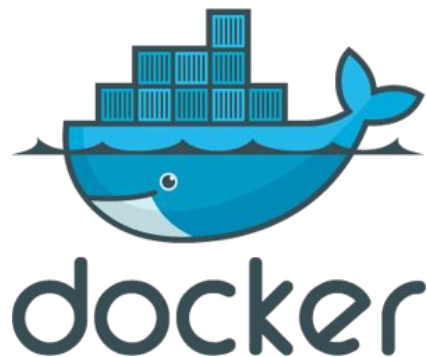
Tecnologias

Tecnologias adotadas

Visando melhorar esse fluxo, toda a estrutura e as tecnologias utilizadas foram analisadas

Docker

O Docker possibilita o empacotamento de uma aplicação ou ambiente inteiro dentro de um container, e a partir desse momento o ambiente inteiro torna-se portátil para qualquer outro Host que contenha o Docker instalado - [Mundo Docker] <https://goo.gl/aqDtgo>



Vantagens do Docker

Ambiente sempre atualizado

Ambiente volátil

Documentação espontânea da estrutura

Dockerfile

```
1 FROM python:3
2 ENV PYTHONUNBUFFERED 1
3 RUN mkdir /code
4 WORKDIR /code
5 ADD requirements.txt /code/
6 RUN pip install -r requirements.txt
7 ADD . /code/
```

Adaptações na aplicação

Uso de getenv para configurações de sistema

Criação de containers a parte para agilizar o processo

Organização diferente de diretórios, mudanças básicas

Falta de registro de determinadas dependências, ponto normal

Alteração de algumas rotinas (escrita em diretório temporário...)



Rancher

Orquestrador de containers

- OpenSource
- Possibilita a criação de containers via interface de administração, Docker-Compose e APIs





Default ▾

STACKS ▾

CATALOG

INFRASTRUCTURE ▾

ADMIN ▾ !

API

Hosts

Add Host

ACTIVE



instance-1

10.132.0.2

1.12.3

Ubuntu 16.04.1 LTS (4.4.0-47-generic)

2.6 GHz

3.61 GiB

9.63 GiB

cloud=Google



ACTIVE



ip-10-0-0-199.eu-west-1.compute.i...

10.0.0.199

1.12.3

RancherOS v0.7.1 (4.4.24-rancher)

2x2.4 GHz

3.86 GiB

14.7 GiB

cloud=AWS



ACTIVE



ip-10-0-1-135.eu-west-1.compute.i...

10.0.1.135

1.12.3

RancherOS v0.7.1 (4.4.24-rancher)

2x2.4 GHz

3.86 GiB

14.7 GiB

cloud=AWS





Default ▾

STACKS ▾

CATALOG ▾

INFRASTRUCTURE ▾

ADMIN ▾

API ▾



User Stacks

Add Stack

Add from Catalog

Sort By: State Name

JamesKolce-com The Brain of James Kolce			Add Service ▾	1 Services	3 Containers	
Active	application + 2 Sidekicks ⓘ	Image: jameskolce/brain		Service	3 Containers	
LetsEncrypt			Up to date Add Service ▾	1 Service	1 Container	
Active	letsencrypt ⓘ	Image: janeczku/rancher-letsencrypt:v0.5.0		Service	1 Container	
Proxy			Add Service ▾	2 Services	2 Containers	
Active	https-redirect ⓘ	Image: hope/redirect-http-to-https Ports: 80		Service	1 Container	
Active	loadbalancer ⓘ	To: JamesKolce-com/application Ports: 443/tcp		Load Balancer	1 Container	



Features

Gerenciamento de Tarefas

GitLab Pages

Wiki

CI/CD

Pipeline

Integração Kubernetes

Integração Telegram



```
test:
  script:
    # this configures Django application to use attached postgres database that is run on `postgres` host
    - export DATABASE_URL=postgres://postgres:@postgres:5432/python-test-app
    - apt-get update -qy
    - apt-get install -y python-dev python-pip
    - pip install -r requirements.txt
    - python manage.py test

staging:
  type: deploy
  script:
    - apt-get update -qy
    - apt-get install -y ruby-dev
    - gem install dpl
    - dpl --provider=heroku --app=gitleab-ci-python-test-staging --api-key=$HEROKU_STAGING_API_KEY
  only:
    - master

production:
  type: deploy
  script:
    - apt-get update -qy
    - apt-get install -y ruby-dev
    - gem install dpl
    - dpl --provider=heroku --app=gitleab-ci-python-test-prod --api-key=$HEROKU_PRODUCTION_API_KEY
  only:
    - tags
```

```
1 image: "docker:latest"
2
3 services:
4   - docker:dind
5
6 before_script:
7   - docker login -u gitlab-ci-token -p $CI_BUILD_TOKEN registry.gitlab.ifma.edu.br
8   - 'curl -s -u "${HCCD_ACCESS_KEY}:${HCCD_SECRET_KEY}"
9     -X POST -H "Accept: application/json" -H "Content-Type: application/json" -d "{}"
10     "http://${CCD_SERVER_URL}:${CCD_SERVER_URL_PORT}/v1/projects/${HCCD_PROJECT_ID}/services/${HCCD_SERVICE_ID}/?action=finishupgrade" > /dev/null'
11   - 'curl -s -u "${CCD_ACCESS_KEY}:${CCD_SECRET_KEY}"
12     -X POST -H "Accept: application/json" -H "Content-Type: application/json" -d "{}"
13     "http://${CCD_SERVER_URL}:${CCD_SERVER_URL_PORT}/v1/projects/${CCD_PROJECT_ID}/services/${HCCD_CRON_ID}/?action=finishupgrade" > /dev/null'
14   - 'curl -s -u "${CCD_ACCESS_KEY}:${CCD_SECRET_KEY}"
15     -X POST -H "Accept: application/json" -H "Content-Type: application/json" -d "{}"
16     "http://${CCD_SERVER_URL}:${CCD_SERVER_URL_PORT}/v1/projects/${CCD_PROJECT_ID}/services/${CCD_SERVICE_ID}/?action=finishupgrade" > /dev/null'
17   - 'curl -s -u "${CCD_ACCESS_KEY}:${CCD_SECRET_KEY}"
18     -X POST -H "Accept: application/json" -H "Content-Type: application/json" -d "{}"
19     "http://${CCD_SERVER_URL}:${CCD_SERVER_URL_PORT}/v1/projects/${CCD_PROJECT_ID}/services/${CCD_CRON_ID}/?action=finishupgrade" > /dev/null'
20   - git checkout tags/$CI_BUILD_REF_NAME
21
22 stages:
23   - build
24   - release
25   - deploy
26   - build_cron
27   - release_cron
28   - deploy_cron
29
30
31 variables:
32   CONTAINER_TEST_IMAGE: $CI_REGISTRY_IMAGE:$CI_BUILD_REF_NAME
33   CT_BASE_TEST_IMAGE: $CI_REGISTRY_IMAGE:base_$CI_BUILD_REF_NAME
34   CT_BASE_IMAGE: $CI_REGISTRY_IMAGE:base
35   CT_CRON_IMAGE: $CI_REGISTRY_IMAGE/cron:$CI_BUILD_REF_NAME
36   CONTAINER_RELEASE_IMAGE: $CI_REGISTRY_IMAGE:latest
37   CT_CRON_RELEASE_IMAGE: $CI_REGISTRY_IMAGE/cron:latest
```

✓ passed

#1228 by 
latest

 
 Merge 



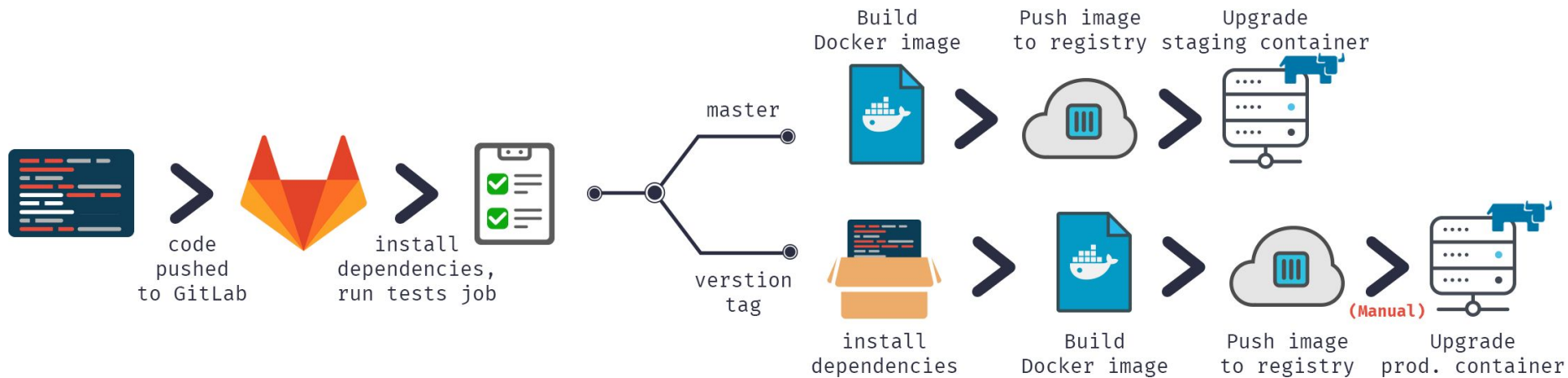
🕒 00:10:59
📅 a day ago

Criando uma tag

🔑 18.01.25-CRON

```
build crontab to homologacao:
  stage: build
  script:
    - 'sed -e "s/<\!--xxVERSAOxx-->/v.$CI_BUILD_REF_NAME/" -i docker/crontab'
    - echo $CI_BUILD_REF_NAME > versao.txt
    - docker build --no-cache -t $CT_CRON_IMAGE -f docker/Dockerfile_cron .
    - docker push $CT_CRON_IMAGE
  only:
    - /^(?:(\d+)\.){1,3}(?:(\d+)\.){1,3}(\*|\d+)-CRON-RC\d+$/
  environment: homologacao
```

























Fluxo



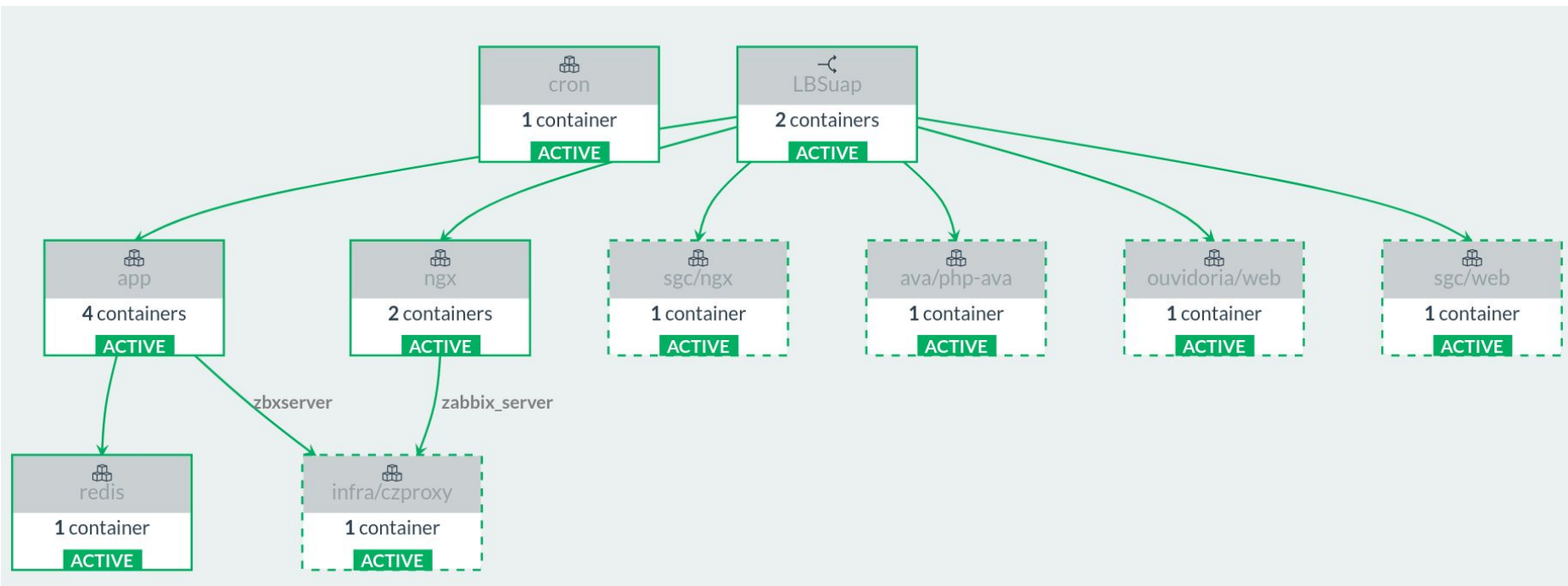
gitlab-ci.yml / deploy

```
deploy cron to homologacao:
  stage: deploy
  script:
    - 'curl -s -u "${HCCD_ACCESS_KEY}:${HCCD_SECRET_KEY}"
      -X POST -H "Accept: application/json" -H "Content-Type: application/json"
      -d "{\"inServiceStrategy\": {\"launchConfig\": {\"imageUid\": \"docker:${CT_CRON_IMAGE}\", \"dataVolumes\": [\"${HCCD_SUAP_VOL}\", \"env
      \"http://${CCD_SERVER_URL}:${CCD_SERVER_URL_PORT}/v1/projects/${HCCD_PROJECT_ID}/services/${HCCD_CRON_ID}/?action=upgrade\" > /dev/null'
  only:
    - /^(?:(\d+)\.){0,3}?(?:(\d+)\.){0,3}?(.*)-CRON-RC\d+$/
  environment:
    name: homologacao
```

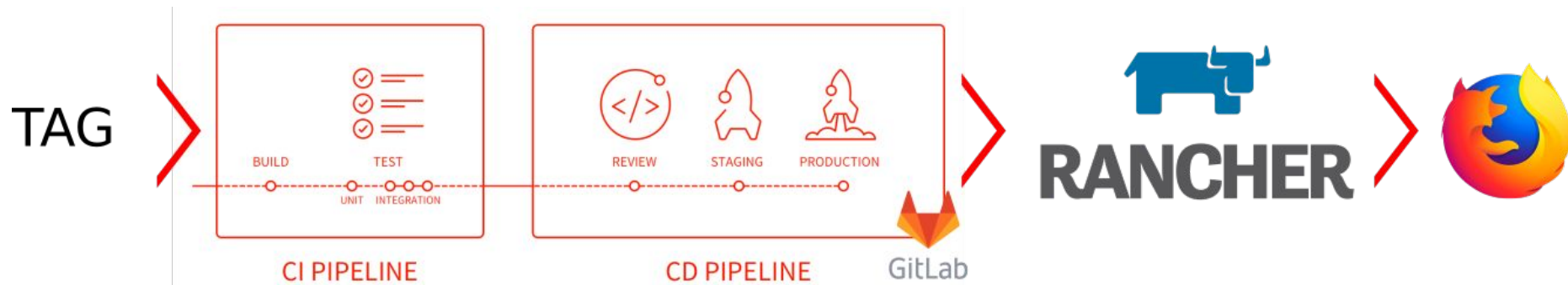
Service no Rancher

Ports	Containers	Labels	Links	Log					
State	Name	IP Address	Host	Image	Stats				
Running	suap-app-1	10.42.48.15	svml00rch006.ifma.edu	registry.gitlab.ifma.edu.br/ndsis/suap:1...					 
Running	suap-app-2	10.42.90.118	cloud004.ifma.edu	registry.gitlab.ifma.edu.br/ndsis/suap:1...					 
Running	suap-app-3	10.42.111.129	cloud004.ifma.edu	registry.gitlab.ifma.edu.br/ndsis/suap:1...					 
Running	suap-app-4	10.42.72.108	cloud02.ifma.edu	registry.gitlab.ifma.edu.br/ndsis/suap:1...					 

Fluxograma de serviços no Rancher



Atualização Hoje





Comparando

Atualização



```
akshay@akshay-UBPC:~/Mytest$ git add README
akshay@akshay-UBPC:~/Mytest$ git add sample.c
akshay@akshay-UBPC:~/Mytest$ git commit -m "first edit"
[master (root-commit) 82c032f] first edit
 2 files changed, 7 insertions(+)
 create mode 100644 README
 create mode 100644 sample.c
akshay@akshay-UBPC:~/Mytest$ git remote add origin https://github.com/akshaypai/Mytest.git
akshay@akshay-UBPC:~/Mytest$ git push origin master
Username for 'https://github.com': akshaypai
Password for 'https://akshaypai@github.com':
Counting objects: 4, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 337 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
to https://github.com/akshaypai/Mytest.git
 * [new branch] master -> master
akshay@akshay-UBPC:~/Mytest$
```



```
akshay@akshay-UBPC:~/Mytest$ git add README
akshay@akshay-UBPC:~/Mytest$ git add sample.c
akshay@akshay-UBPC:~/Mytest$ git commit -m "first edit"
[master (root-commit) 82c032f] first edit
 2 files changed, 7 insertions(+)
 create mode 100644 README
 create mode 100644 sample.c
akshay@akshay-UBPC:~/Mytest$ git remote add origin https://github.com/akshaypai/Mytest.git
akshay@akshay-UBPC:~/Mytest$ git push origin master
Username for 'https://github.com': akshaypai
Password for 'https://akshaypai@github.com':
Counting objects: 4, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 337 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
to https://github.com/akshaypai/Mytest.git
 * [new branch] master -> master
akshay@akshay-UBPC:~/Mytest$
```



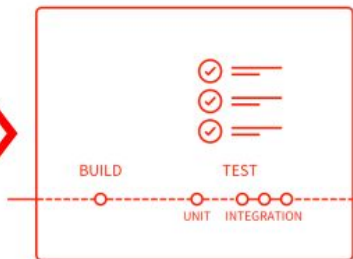
Migration



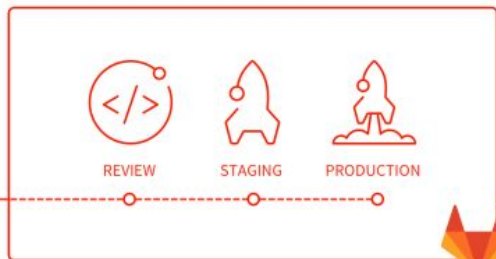
x 4



TAG



CI PIPELINE



CD PIPELINE

GitLab



Problemas / Dificuldades / Desafios

Problemas e barreiras encontradas
Novos desafios e implementações

Duvidas???

