

compass.uol

AI R

Agentic AI
Software
Engineering



Projeto Github Actions



Github Actions.

Onde se encontra?

Github Actions é uma ferramenta para CI/CD dentro do próprio Github.

Para que serve?

Essa ferramenta serve para criar pipelines CI/CD



Uma **pipeline CI/CD** é como uma **linha de produção de fábrica**, mas para software. Ela garante que cada mudança feita no código passe por etapas automáticas para **testar, preparar e entregar** o software de forma segura e rápida.

No repositório da aplicação encontra-se o arquivo de configuração do Github Actions, aplicação em python e o Dockerfile (documento que serve como base para a criação da imagem docker)

```
FROM python:alpine3.22

WORKDIR /app

RUN pip install "fastapi" uvicorn

COPY main.py .

EXPOSE 8080

CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8080"]
```

Os manifestos são arquivos de instrução para o Kubernetes, dizendo como rodar e expor a aplicação.

Manifestos Kubernetes

Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: app-python-deployment
  labels:
    app: app-python
spec:
  replicas: 2
  selector:
    matchLabels:
      app: app-python
  template:
    metadata:
      labels:
        app: app-python
    spec:
      containers:
        - name: app-python
          image: yuribrigido/app-python:7576d7f324dfb86579351f010eb53af7e220d0f9
          ports:
            - containerPort: 8080
```

Service

```
apiVersion: v1
kind: Service
metadata:
  name: app-python-service
spec:
  selector:
    app: app-python
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
  type: ClusterIP
```

Os manifestos são arquivos de instrução para o Kubernetes, dizendo como rodar e expor a aplicação.

Manifestos Kubernetes

Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: app-python-deployment
  labels:
    app: app-python
spec:
  replicas: 2
  selector:
    matchLabels:
      app: app-python
  template:
    metadata:
      labels:
        app: app-python
    spec:
      containers:
        - name: app-python
          image: yuribrigido/app-python:7576d7f324dfb86579351f010eb53af7e220d0f9
          ports:
            - containerPort: 8080
```

Service

```
apiVersion: v1
kind: Service
metadata:
  name: app-python-servico
spec:
  selector:
    app: app-python
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
  type: ClusterIP
```

Qual objetivo?

Pipelines:

Automatizam ciclo de atualização da aplicação.

Antes, um desenvolvedor precisaria fazer tudo manualmente. Construir a imagem, enviar pro Docker Hub, atualizar os arquivos Kubernetes, aplicar no cluster...

Github Actions(CI/CD)

Para o funcionamento do GitHub Actions criou-se um arquivo YAML e com ele será feita a configuração de todo o CI/CD. Esse arquivo de configuração deve estar localizado na estruturação de pastas corretas (Nesse caso estava no primeiro repositório e seguindo as normas de estruturação de pastas)

```
name: CI/CD

on:
  push:

jobs:
  buildar-e-deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Login Docker Hub
        uses: docker/login-action@v3
        with:
          username: ${secrets.DOCKER_NOMEUSUARIO}
          password: ${secrets.DOCKER_SENHA}

      - name: Buildar e push da imagem Docker
        uses: docker/build-push-action@v5
        with:
          context: .
          push: true
          tags: |
            ${secrets.DOCKER_NOMEUSUARIO}/app-python:latest
            ${secrets.DOCKER_NOMEUSUARIO}/app-python:${github.sha}

      - name: Checkout Repositório dos manifestos ArgoCD
        uses: actions/checkout@v4
        with:
          repository: yuri-ferreira/app-manifests
          token: ${secrets.PAT}}
          path: app-manifests

      - name: Atualizar tag da imagem no manifesto Kubernetes
        run: |
          cd app-manifests
          sed -i "s|image: ./app-python:.*|image: ${secrets.DOCKER_NOMEUSUARIO}/app-python:${github.sha}|g" deployment.yaml
          git config user.name "Github Actions Bot"
          git config user.email "actions@github.com"
          git add deployment.yaml
          git commit -m "Atualizando a imagem do app-pyhton para ${github.sha} 🚀" || echo "Nenhuma mudança 🙄"

      - name: Criar Pull Request para o manifesto ArgoCD
        uses: peter-evans/create-pull-request@v6
        with:
          token: ${secrets.PAT}}
          commit-message: "Atualizar a imagem do python-app para ${github.sha}"
          title: "Atualizar a imagem do python-app para ${github.sha}"
          body: "Esse Pull Request serve para atualizar a imagem Docker do python-app no manifesto Kubernetes :)"
          branch: atualizar-tag-da-imagem${github.sha}
          base: main
          delete-branch: true
          path: app-manifests
```

Com o uso da ação "login-action@v3", é feito a autenticação no Docker Hub

Aqui é utilizado um recurso do Github chamado "Secrets"

Github Actions(CI/CD)

```
name: CI/CD

on:
  push:

jobs:
  buildar-e-deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Login Docker Hub
        uses: docker/login-action@v3
        with:
          username: ${secrets.DOCKER_NOMEUSUARIO}
          password: ${secrets.DOCKER_SENHA}

      - name: Buildar e push da imagem Docker
        uses: docker/build-push-action@v5
        with:
          context: .
          push: true
          tags: |
            ${secrets.DOCKER_NOMEUSUARIO}/app-python:latest
            ${secrets.DOCKER_NOMEUSUARIO}/app-python:${github.sha}

      - name: Checkout Repositório dos manifestos ArgoCD
        uses: actions/checkout@v4
        with:
          repository: yuri-ferreira/app-manifests
          token: ${secrets.PAT}
          path: app-manifests

      - name: Atualizar tag da imagem no manifesto Kubernetes
        run: |
          cd app-manifests
          sed -i "s|image: ./app-python:.*|image: ${secrets.DOCKER_NOMEUSUARIO}/app-python:${github.sha}|g" deployment.yaml
          git config user.name "Github Actions Bot"
          git config user.email "actions@github.com"
          git add deployment.yaml
          git commit -m "Atualizando a imagem do app-pyhton para ${github.sha} 🐍" || echo "Nenhuma mudança 🐍"

      - name: Criar Pull Request para o manifesto ArgoCD
        uses: peter-evans/create-pull-request@v6
        with:
          token: ${secrets.PAT}
          commit-message: "Atualizar a imagem do python-app para ${github.sha}"
          title: "Atualizar a imagem do python-app para ${github.sha}"
          body: "Esse Pull Request serve para atualizar a imagem Docker do python-app no manifesto Kubernetes :)"
          branch: atualizar-tag-da-imagem${github.sha}
          base: main
          delete-branch: true
          path: app-manifests
```

Com a ação "build-push-action@v5" é possível construir a imagem docker usando o Dockerfile e após é enviado ao Docker Hub.

São criadas duas tags para a imagem, a latest e o SHA.

Github Actions(CI/CD)

```
name: CI/CD

on:
  push:

jobs:
  buildar-e-deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Login Docker Hub
        uses: docker/login-action@v3
        with:
          username: ${secrets.DOCKER_NOMEUSUARIO}
          password: ${secrets.DOCKER_SENHA}

      - name: Buildar e push da imagem Docker
        uses: docker/build-push-action@v5
        with:
          context: .
          push: true
          tags: |
            ${secrets.DOCKER_NOMEUSUARIO}/app-python:latest
            ${secrets.DOCKER_NOMEUSUARIO}/app-python:${github.sha}

      - name: Checkout Repositório dos manifestos ArgoCD
        uses: actions/checkout@v4
        with:
          repository: yuri-ferreira/app-manifests
          token: ${secrets.PAT}}
          path: app-manifests

      - name: Atualizar tag da imagem no manifesto Kubernetes
        run: |
          cd app-manifests
          sed -i "s|image: ./app-python:.*|image: ${secrets.DOCKER_NOMEUSUARIO}/app-python:${github.sha}|g" deployment.yaml
          git config user.name "Github Actions Bot"
          git config user.email "actions@github.com"
          git add deployment.yaml
          git commit -m "Atualizando a imagem do app-pyhton para ${github.sha} 🐳" || echo "Nenhuma mudança 🐳"

      - name: Criar Pull Request para o manifesto ArgoCD
        uses: peter-evans/create-pull-request@v6
        with:
          token: ${secrets.PAT}}
          commit-message: "Atualizar a imagem do python-app para ${github.sha}"
          title: "Atualizar a imagem do python-app para ${github.sha}"
          body: "Esse Pull Request serve para atualizar a imagem Docker do python-app no manifesto Kubernetes :)"
          branch: atualizar-tag-da-imagem${github.sha}
          base: main
          delete-branch: true
          path: app-manifests
```

Aqui é uma parte muito importante do CI/CD, é feito uma série de comandos a serem executados no ubuntu que tem como os seguintes objetivos:

- Entrar na pasta do repositório clonado em um dos jobs anteriores.
- É feito a troca da imagem no manifesto kubernetes que será mostrado nas etapas seguintes.
- Também é feito algumas configurações como o nome do usuário, email e é feito o commit da alteração.

Github Actions(CI/CD)

```
name: CI/CD

on:
  push:

jobs:
  buildar-e-deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Login Docker Hub
        uses: docker/login-action@v3
        with:
          username: ${secrets.DOCKER_NOMEUSUARIO}
          password: ${secrets.DOCKER_SENHA}

      - name: Buildar e push da imagem Docker
        uses: docker/build-push-action@v5
        with:
          context: .
          push: true
          tags: |
            ${secrets.DOCKER_NOMEUSUARIO}/app-python:latest
            ${secrets.DOCKER_NOMEUSUARIO}/app-python:${github.sha}

      - name: Checkout Repositório dos manifestos ArgoCD
        uses: actions/checkout@v4
        with:
          repository: yuri-ferreira/app-manifests
          token: ${secrets.PAT}
          path: app-manifests

      - name: Atualizar tag da imagem no manifesto Kubernetes
        run: |
          cd app-manifests
          sed -i "s|image: ./app-python:.*|image: ${secrets.DOCKER_NOMEUSUARIO}/app-python:${github.sha}|g" deployment.yaml
          git config user.name "Github Actions Bot"
          git config user.email "actions@github.com"
          git add deployment.yaml
          git commit -m "Atualizando a imagem do app-pyhton para ${github.sha} 🔄" || echo "Nenhuma mudança 🙌"

      - name: Criar Pull Request para o manifesto ArgoCD
        uses: peter-evans/create-pull-request@v6
        with:
          token: ${secrets.PAT}
          commit-message: "Atualizar a imagem do python-app para ${github.sha}"
          title: "Atualizar a imagem do python-app para ${github.sha}"
          body: "Esse Pull Request serve para atualizar a imagem Docker do python-app no manifesto Kubernetes :)"
          branch: atualizar-tag-da-imagem${github.sha}
          base: main
          delete-branch: true
          path: app-manifests
```


No final, tudo vira um pull request para atualizar o repositório de forma segura.

Github Actions(CI/CD)

```
name: CI/CD

on:
  push:

jobs:
  buildar-e-deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Login Docker Hub
        uses: docker/login-action@v3
        with:
          username: ${secrets.DOCKER_NOMEUSUARIO}
          password: ${secrets.DOCKER_SENHA}

      - name: Buildar e push da imagem Docker
        uses: docker/build-push-action@v5
        with:
          context: .
          push: true
          tags: |
            ${secrets.DOCKER_NOMEUSUARIO}/app-python:latest
            ${secrets.DOCKER_NOMEUSUARIO}/app-python:${github.sha}

      - name: Checkout Repositório dos manifestos ArgoCD
        uses: actions/checkout@v4
        with:
          repository: yuri-ferreira/app-manifests
          token: ${secrets.PAT}
          path: app-manifests

      - name: Atualizar tag da imagem no manifesto Kubernetes
        run: |
          cd app-manifests
          sed -i "s|image: ./app-python:.*|image: ${secrets.DOCKER_NOMEUSUARIO}/app-python:${github.sha}|g" deployment.yaml
          git config user.name "Github Actions Bot"
          git config user.email "actions@github.com"
          git add deployment.yaml
          git commit -m "Atualizando a imagem do app-python para ${github.sha} 🔄" || echo "Nenhuma mudança 🙄"

      - name: Criar Pull Request para o manifesto ArgoCD
        uses: peter-evans/create-pull-request@v6
        with:
          token: ${secrets.PAT}
          commit-message: "Atualizar a imagem do python-app para ${github.sha}"
          title: "Atualizar a imagem do python-app para ${github.sha}"
          body: "Esse Pull Request serve para atualizar a imagem Docker do python-app no manifesto Kubernetes :)"
          branch: atualizar-tag-da-imagem${github.sha}
          base: main
          delete-branch: true
          path: app-manifests
```


O ArgoCD tem como papel "observar" o repositório dos manifestos Kubernetes e caso ocorra alguma mudança cabe a ele atualizar a aplicação.

Dessa maneira pode-se automatizar o deploy da aplicação sem se preocupar quando ocorre alguma mudança nela.



CREATE

CANCEL

×

SOURCE

Repository URL

https://github.com/yuri-ferreira/app-manifests

GIT ▼

Revision

main

Branches ▼

Path

.

DESTINATION

Cluster Name

in-cluster

NAME ▼

Namespace

app-python

Directory ▼

DIRECTORY

CREATE

CANCEL

×

SOURCE

Repository URL

https://github.com/yuri-ferreira/app-manifests

GIT ▾

Revision

main

Branches ▾

Path

.

DESTINATION

Cluster Name

in-cluster


NAME ▾

Namespace

app-python

Directory ▾

DIRECTORY

argo
v3.0.6+db93798

Applications

Settings


User Info


Documentation


Application filters

☐ ★ Favorites Only


SYNC STATUS


☐  Unknown 0


☐  Synced 1


☐  OutOfSync 0


HEALTH STATUS


☐  Progressing 0

☐  Suspended 0

☐  Healthy 1

☐  Degraded 0

☐  Missing 0

☐  Unknown 0

LABELS

LABELS

PROJECTS

PROJECTS

Applications


+ NEW APP

↻ SYNC APPS

↻ REFRESH APPS

🔍 Search applications...

📄



APPLICATIONS TILES


📊

📋

📈



Log out

Sort: name ▼ Items per page: 10 ▼

app-python☆

Project: default

Labels:

Status:  Healthy  Synced

Reposito... https://github.com/yuri-ferreira/app-ma...

Target R... main

Path: .

Destinat... in-cluster

Namesp... app-python

Created ... 07/24/2025 16:44:08 (a few seconds a...


Last Sync: 07/24/2025 16:44:08 (a few seconds a...

↻ SYNC

↻ REFRESH

🗑 DELETE

Interna

argo
v3.0.6+db93798

Applications

Settings

User Info

Documentation

Application filters

☐ ★ Favorites Only

SYNC STATUS

☐ ⏸ Unknown0

☐ ✅ Synced1

☐ ⚡ OutOfSync0

HEALTH STATUS

☐ 🔄 Progressing0

☐ ⏸ Suspended0

☐ ❤️ Healthy1

☐ ❤️ Degraded0

☐ 🧑 Missing0

☐ ❓ Unknown0

LABELS

LABELS

PROJECTS

PROJECTS

Applications

+ NEW APP

↻ SYNC APPS

🔄 REFRESH APPS

🔍 Search applications...

📄

🔧

APPLICATIONS TILES


📊

📋

📈

Log out

Sort: name ▼ Items per page: 10 ▼

app-python

☆

Project: default

Labels:

Status: ❤️ Healthy ✅ Synced

Reposito... https://github.com/yuri-ferreira/app-ma...

Target R... main

Path: .

Destinat... in-cluster

Namesp... app-python

Created ... 07/24/2025 16:44:08 (a few seconds a...

Last Sync: 07/24/2025 16:44:08 (a few seconds a...

↻ SYNC

🔄 REFRESH

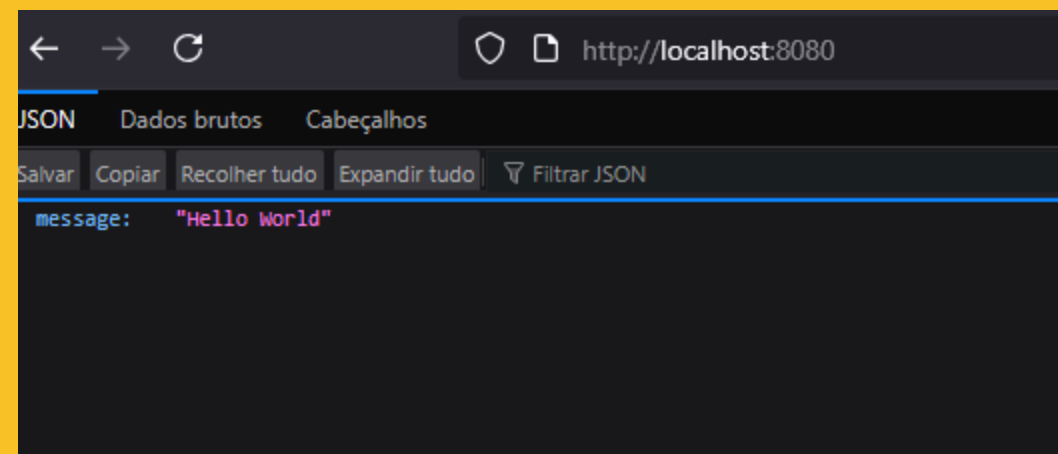
🗑 DELETE



Interna

O ArgoCD tem como papel "observar" o repositório dos manifestos kubernetes e caso ocorra alguma mudança cabe a ele atualizar a aplicação.

Dessa maneira pode-se automatizar o deploy da aplicação sem se preocupar quando ocorre alguma mudança nela.



Acessando aplicação localmente



 **app-python** 

Project: default

Labels:

Status:  Progressing  Synced

Reposito... <https://github.com/yuri-ferreira/app-ma...>

Target R... main




Path: .

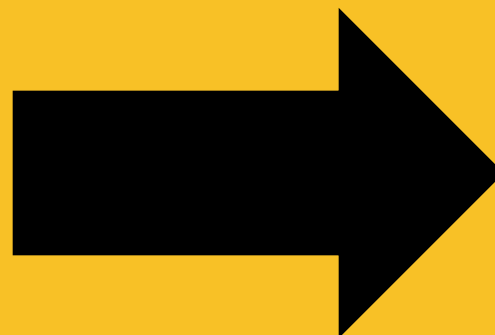
Destinat... in-cluster



Namesp... app-python

Created ... 07/24/2025 16:44:08 (12 minutes ago)

Last Sync: 07/24/2025 16:55:42 (a few seconds a...)



 SYNC  REFRESH  DELETE



 **app-python** 

Project: default

Labels:

Status:  Healthy  Synced

Reposito... <https://github.com/yuri-ferreira/app-ma...>

Target R... main




Path: .

Destinat... in-cluster

Namesp... app-python

Created ... 07/24/2025 16:44:08 (12 minutes ago)

Last Sync: 07/24/2025 16:55:42 (a few seconds a...)

 SYNC  REFRESH  DELETE

