



Engenharia de Software para Nuvem - Aula 2

Jemerson Fernando Maia - jfnandopr@gmail.com

Pós-graduação em Engenharia de Software para Modernização de Sistemas

BIOPARK EDUCAÇÃO

Agenda

- Git e GitHub
- CI/CD
- Docker
- Docker-Compose
- Terraform
- Atividade



Git

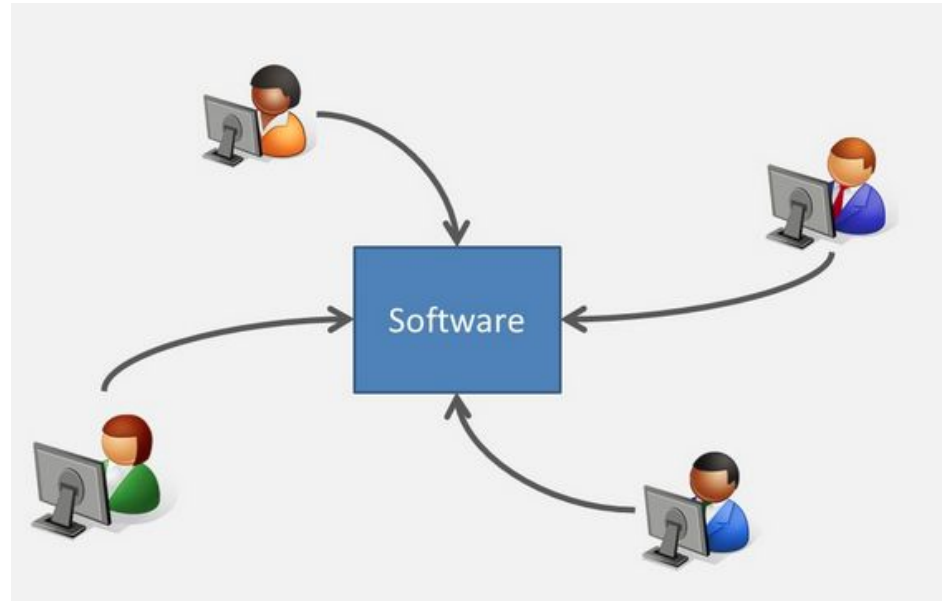
Controle de Fontes

Permite acompanhar as mudanças realizadas no código-fonte, garantindo a integridade do projeto e facilitando a colaboração entre os membros da equipe.



Controle de Fontes

- Benefícios
 - Histórico de alterações completo
 - Ramificação e mesclagem
 - Rastreabilidade



Controle de Fontes



TOP VERSION CONTROL SYSTEMS



Git

"Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency."

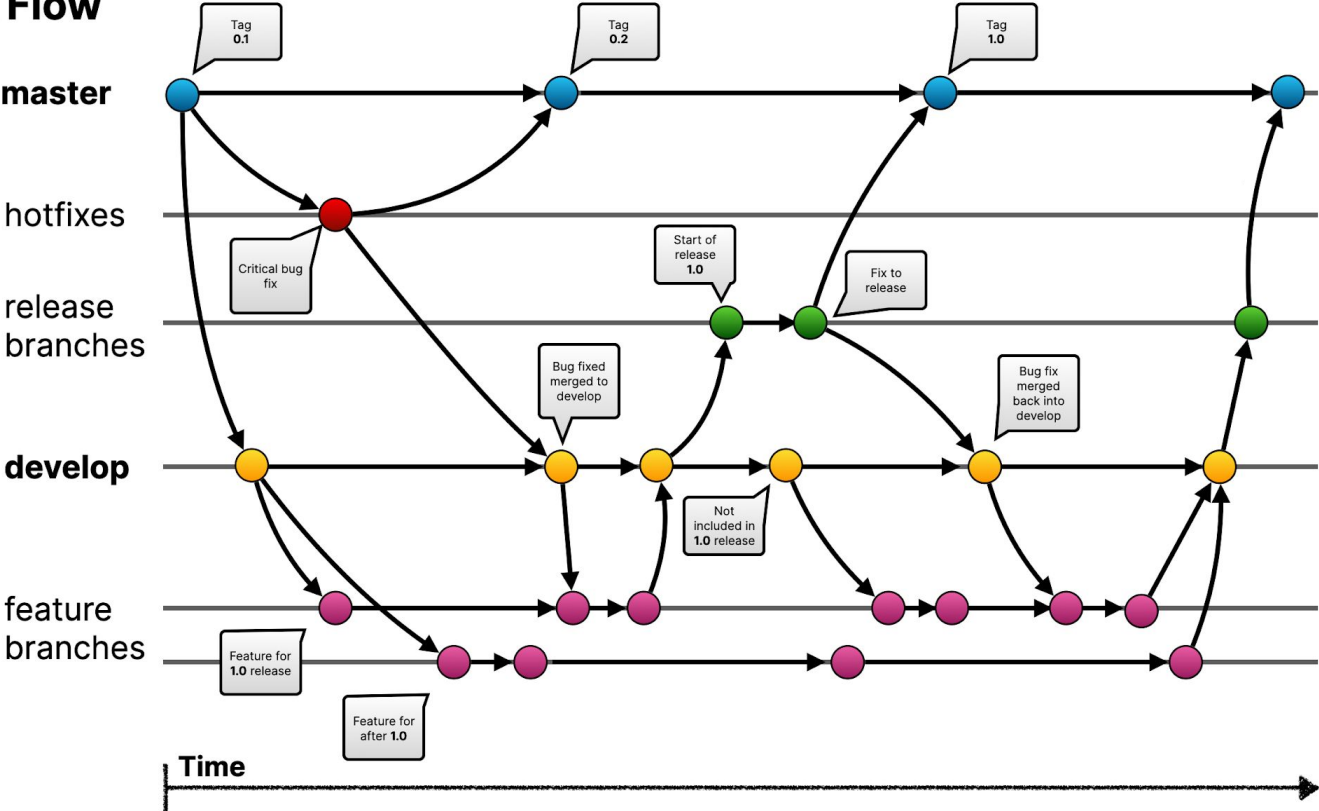
<https://git-scm.com/>



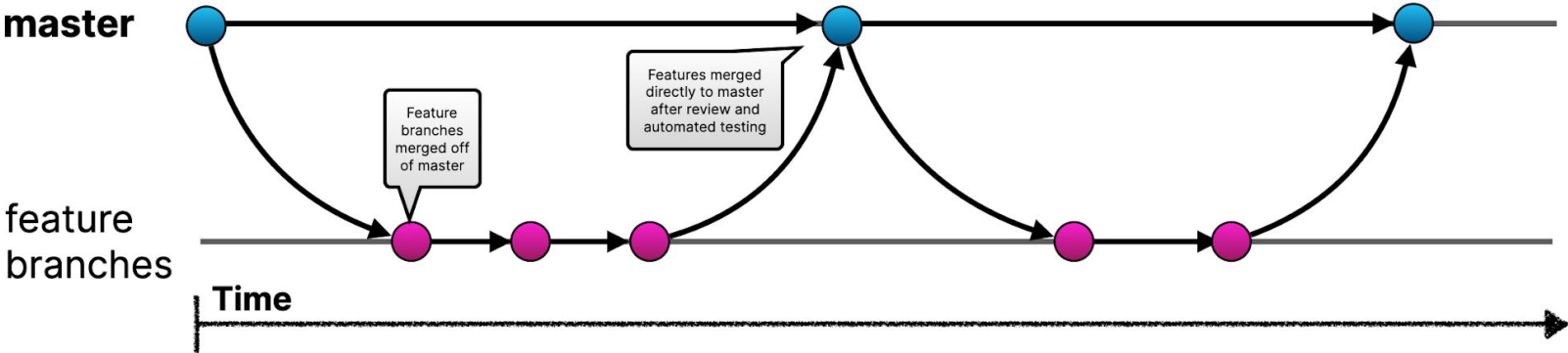
Branches

Branch (Ramificação) significa que você ramifica a linha principal de desenvolvimento e continua a trabalhar sem alterar essa linha principal.

Git Flow

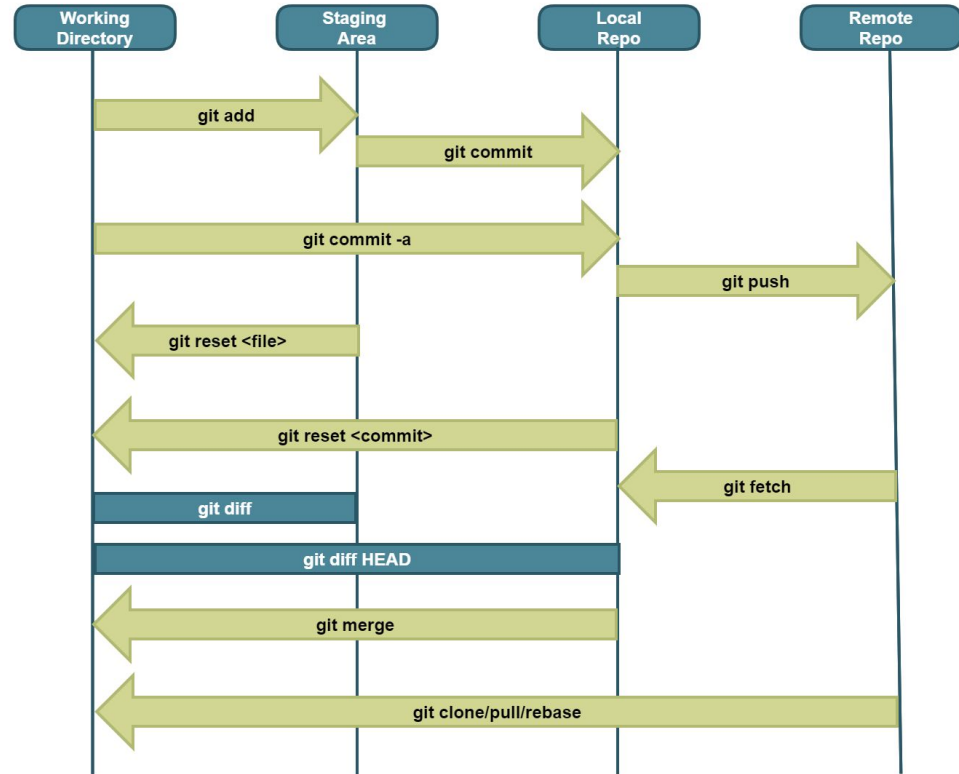


GitHub Flow



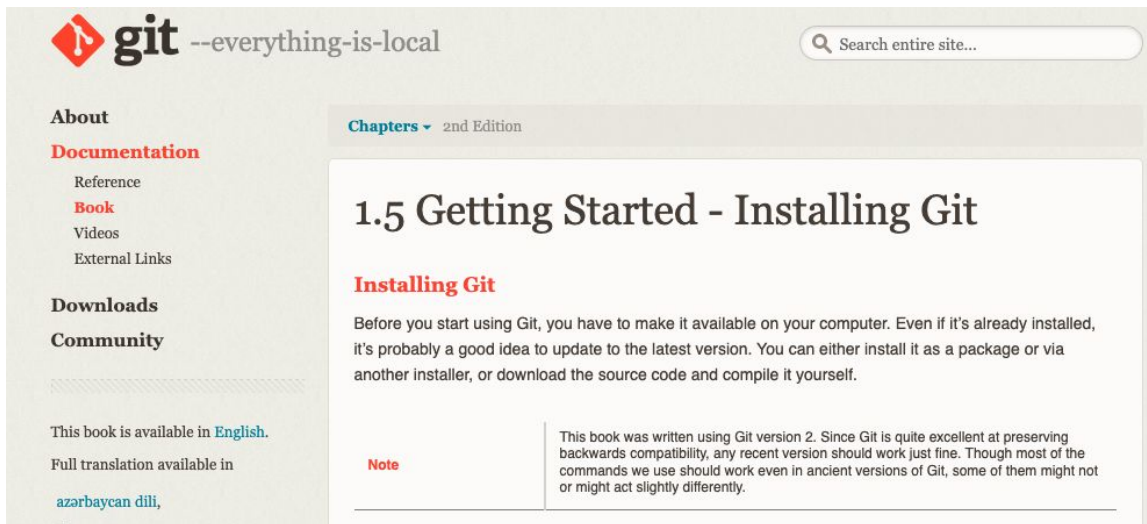
Comandos

GIT MERGE



Instalação

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>



The screenshot shows the Git documentation website. At the top left is the Git logo with the tagline "--everything-is-local". To the right is a search bar labeled "Search entire site...". On the left sidebar, under "About", the "Documentation" link is highlighted in red. Below it are links for "Reference", "Book", "Videos", and "External Links". Under "Downloads" and "Community" are also listed. At the bottom of the sidebar, it says "This book is available in [English](#). Full translation available in [azərbaycan dili](#),". The main content area shows the "Chapters" dropdown set to "2nd Edition". The current chapter is "1.5 Getting Started - Installing Git". Below the chapter title, the text reads: "Before you start using Git, you have to make it available on your computer. Even if it's already installed, it's probably a good idea to update to the latest version. You can either install it as a package or via another installer, or download the source code and compile it yourself." A "Note" box at the bottom states: "This book was written using Git version 2. Since Git is quite excellent at preserving backwards compatibility, any recent version should work just fine. Though most of the commands we use should work even in ancient versions of Git, some of them might not or might act slightly differently."

git --everything-is-local

Search entire site...

About

Documentation

Reference

Book

Videos

External Links

Downloads

Community

This book is available in [English](#).
Full translation available in [azərbaycan dili](#),

Chapters ▾ 2nd Edition

1.5 Getting Started - Installing Git

Installing Git

Before you start using Git, you have to make it available on your computer. Even if it's already installed, it's probably a good idea to update to the latest version. You can either install it as a package or via another installer, or download the source code and compile it yourself.

Note

This book was written using Git version 2. Since Git is quite excellent at preserving backwards compatibility, any recent version should work just fine. Though most of the commands we use should work even in ancient versions of Git, some of them might not or might act slightly differently.

Plataformas Git



Bitbucket



GitHub



Gitlab



GitHub



GitHub

GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git.

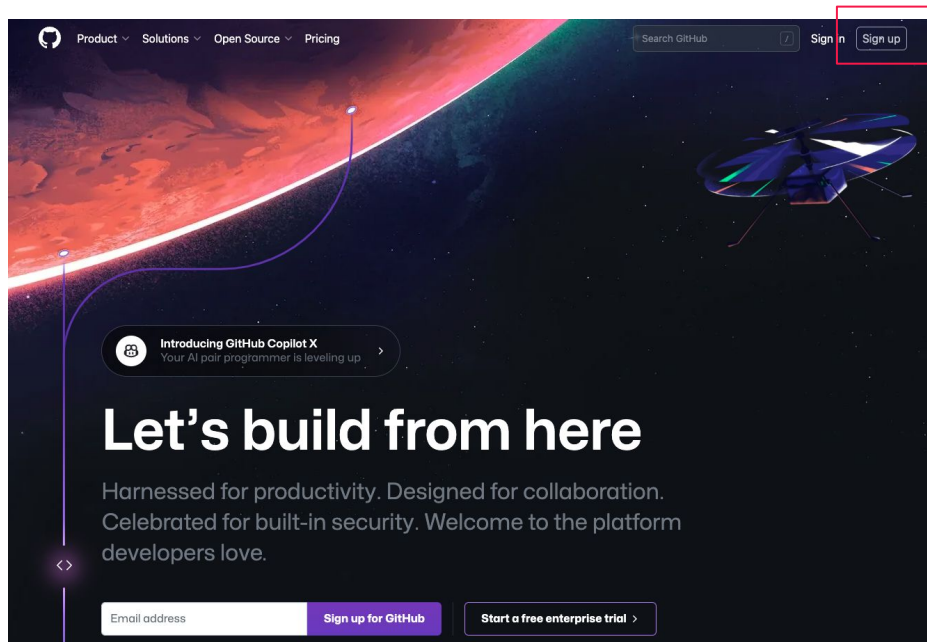
Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo.



Trabalhando em equipe

- Usuário/Conta
- Organizações e Repositórios
- Fluxo de desenvolvimento estabelecido
- Ramificação ou ramificações principais protegidas
- Criação de *Pull-Request*
 - Execução de actions
 - Revisão de código (*Code Review*)
- Registro de problemas (Issues) *

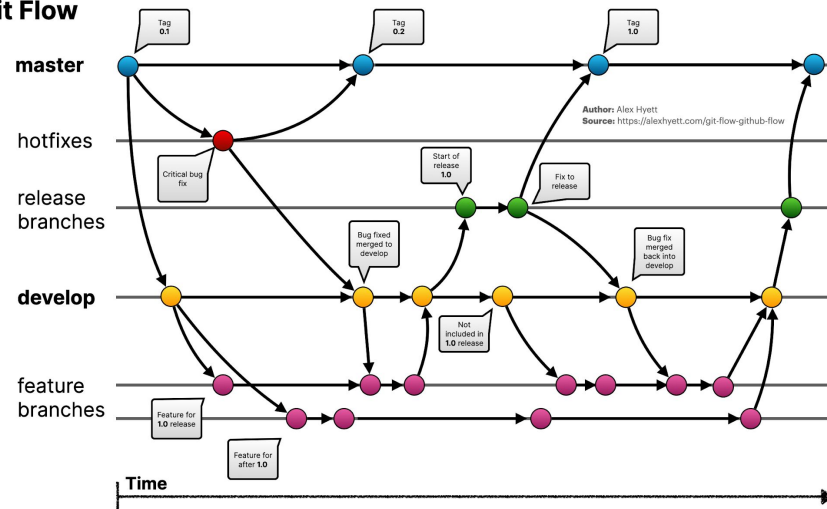
Usuário/Conta



<https://github.com/>

Fluxo de desenvolvimento

Git Flow



<> Code Issues Pull requests Actions Projects Security Insights Settings

Search branches...

Overview

Yours

Active

Stale

All branches

New branch

Default branch

main Updated 2 months ago by jfmandopr

Default



Your main branch isn't protected

Protect this branch from force pushing or deletion, or require status checks before merging. [Learn more](#)

Dismiss

Protect this branch

Ramificação(ões) Protegida(s)

<> Code Issues Pull requests Actions Projects Security Insights Settings

General

Access

Collaborators and teams

Code and automation

Branches

Tags

Rules

Beta

Actions

Webhooks

Pages

Branch protection rule



Protect your most important branches

[Branch protection rules](#) define whether collaborators can delete or force push to the branch and set requirements for any pushes to the branch, such as passing status checks or a linear commit history.



Your rules won't be enforced on this private repository until you [upgrade this organization account to GitHub Team or Enterprise](#).

Branch name pattern *

main

Criação de Pull-Request

<> Code Issues **Pull requests** Actions Projects Security Insights Settings

test had recent pushes less than a minute ago

Compare & pull request

Filters is:pr is:open

Labels 9

Milestones 0

New pull request

<> Code Issues **Pull requests 1** Actions Projects Security Insights Settings

Filters is:pr is:open

Labels 9

Milestones 0

New pull request

☐ 1 Open 0 Closed

Author

Label

Projects

Milestones

Reviews

Assignee

Sort

☐ Create README.md

#1 opened now by jfnandopr

Revisão de Código

Create README.md #1

Edit <> Code

 Open jfnandopr wants to merge 1 commit into `main` from `test`

Conversation 0 Commits 1 Checks 0 Files changed 1

+1 -0

Changes from all commits File filter Conversations Jump to

0 / 1 files viewed

Review in codespace

Finish your review 1

1 README.md

<> Viewed

@@ -0,0 +1 @@

1 + # poc-terraform-aws

jfnandopr Pending Author

Não precisa adicionar mais informações?



Reply...



Registro de problemas (Issues)

< > Code Issues 1 Pull requests 1 Actions Projects Security Insights Settings

Filters


Labels 9

Milestones 0

New issue

☐ 1 Open ✓ 0 Closed

Author ▾ Label ▾ Projects ▾ Milestones ▾ Assignee ▾ Sort ▾

☐  **Faltando archivo README**
#2 opened now by jfnandopr

Hands on



Hands On

- Criar conta GitHub
- Acessar Codespace no Github
 - Configurar chave ssh
 - `ssh-keygen -t ed25519 -C "email.da@conta"`
 - `cat ~/.ssh/id_ed25519.pub`
 - Configurar em Profile -> Settings -> SSH and GPG Keys
- Clonar o repositório **biopark-nodejs-api** (`git@github.com:jfnandopr/biopark-nodejs-api.git`)
- Criar nova branch
- Realizar alterações e commits
- Criar PR
- Mesclar (merge) o PR na branch principal



Docker

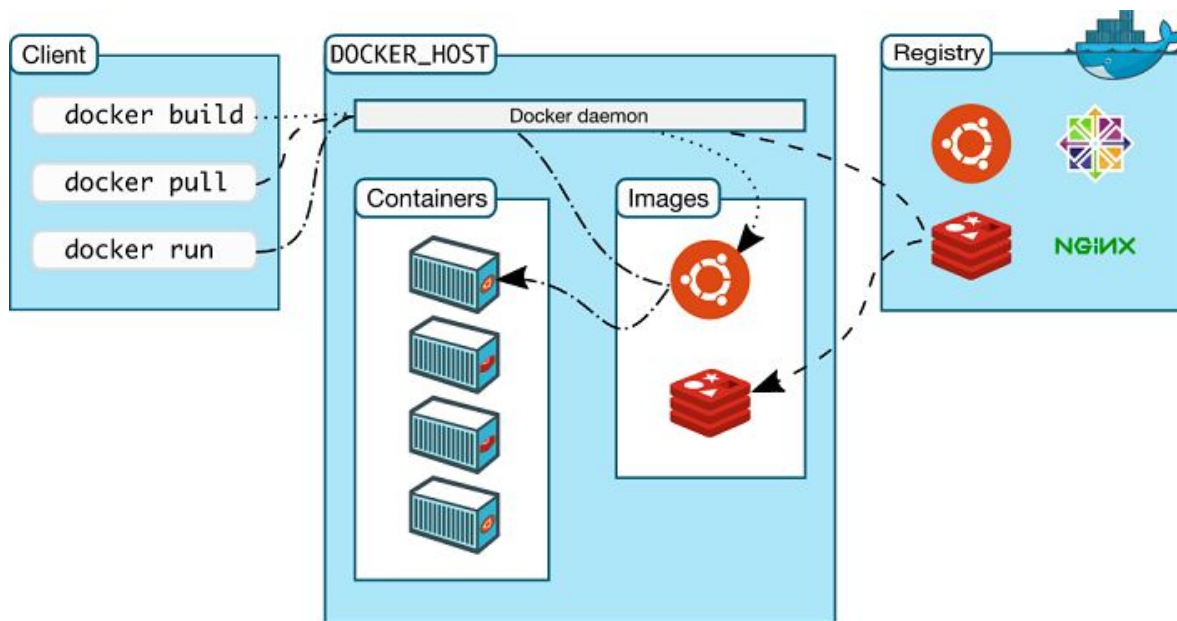


O que é?

É uma tecnologia de containerização que permite a criação e o uso de containers Linux

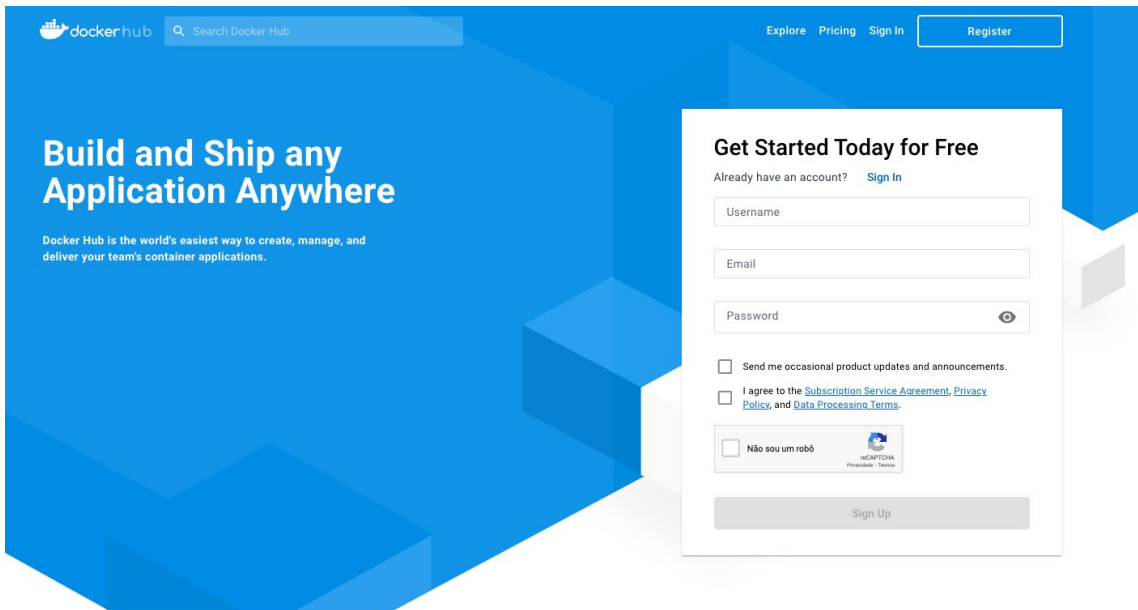
O Docker possibilita o empacotamento de uma aplicação ou ambiente inteiro dentro de um container, e a partir desse momento o ambiente inteiro torna-se portátil para qualquer outro Host que contenha o Docker instalado

Arquitetura





Docker Hub




The image shows the Docker Hub registration page. The background is blue with a geometric pattern of cubes. On the left, the text 'Build and Ship any Application Anywhere' is displayed in white, followed by a smaller line: 'Docker Hub is the world's easiest way to create, manage, and deliver your team's container applications.' The top navigation bar is blue and contains the Docker Hub logo, a search bar, and links for 'Explore', 'Pricing', 'Sign In', and a 'Register' button. A white registration form is overlaid on the right side of the page.

Build and Ship any Application Anywhere

Docker Hub is the world's easiest way to create, manage, and deliver your team's container applications.


Get Started Today for Free

Already have an account? [Sign In](#)




☐ Send me occasional product updates and announcements.


☐ I agree to the [Subscription Service Agreement](#), [Privacy Policy](#), and [Data Processing Terms](#).

☐ Não sou um robô 
reCAPTCHA
Privacidade Termos

<https://hub.docker.com/>

Instalação

 docker docs

 Search the docs

Home

Guides

Manuals

Reference

Samples

Contribute

[Home](#) / [Manuals](#) / [Docker Engine](#) / [Install](#) / [Overview](#)

Docker Desktop

Docker Extensions

Docker Engine

Overview

Install

Overview

CentOS

Debian

Fedora

RHEL

SLES

Ubuntu

Raspbian

Binaries

Post-installation steps

 **Docker Desktop for Linux**

<https://docs.docker.com/engine/install/>



Comandos

- **docker images:** lista imagens baixadas.
- **docker search:** procura e lista imagens do docker hub.
- **docker pull:** download da imagem do docker hub.
- **docker ps:** lista containers que estão rodando.
- **docker rm:** remove um container.
- **docker rmi:** remove uma imagem.
- **docker run:** cria e inicia um container.
- **docker start/stop/restart:** inicia, para ou reinicia um container.



Exemplo

```
$ docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
```

```
$ docker run -it --name server1 ubuntu
```

```
$ docker ps -a
```

```
$ docker start server1
```

```
$ docker exec -it server1 bash
```

```
$ docker restart server1
```

```
$ docker stop server1
```

```
$ docker rm server1
```



Network

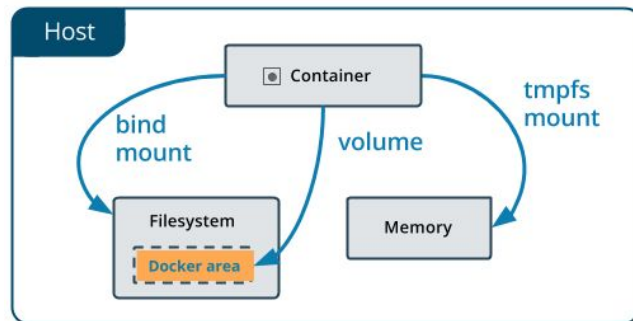
```
$ docker network ls
```

Usada principalmente para estabelecer comunicação entre os contêineres do Dock

Volume

```
$ docker volume ls
```

Usado para manter dados persistentes, compartilhar arquivos/configurações



Dockerfile

O que é?

Serve como a receita para construir um container, permitindo definir um ambiente personalizado e próprio para seu projeto pessoal ou empresarial.





Exemplo

```
FROM ubuntu

RUN apt-get update -y; \
    apt-get install -y nginx curl; \
    rm -rf /var/lib/apt/lists/*

VOLUME ["/var/www/html", "/var/log/nginx"]

EXPOSE 80/tcp

WORKDIR /var/www/

COPY entrypoint.sh /entrypoint.sh
RUN chmod +x /entrypoint.sh

ADD index.tar.gz /var/www/html/

ENTRYPOINT ["/entrypoint.sh"]
CMD ["/usr/sbin/nginx", "-g", "daemon off;"]
```



Criando a Imagem e Executando o Container

```
$ docker build -t account/image-name "."
```

```
$ docker run --name nginx -d -p 80:80  
account/image-name
```



Disponibilizando a imagem no docker hub

```
$ docker login -u username
```

```
Authenticating with existing credentials...  
Login Succeeded
```

```
Logging in with your password grants your terminal complete access to your account.  
For better security, log in with a limited-privilege personal access token. Learn more at  
https://docs.docker.com/go/access-tokens/
```

```
$ docker push account/image-name
```

```
# docker tag another/image-name account/image-name
```



Hands On

- Acessar Codespace no Github
- Acessar o projeto da aplicação nodejs
- Criar o arquivo Dockerfile para a aplicação nodejs
- Construir a imagem
- Enviar para Docker Hub

<https://docs.docker.com/language/nodejs/build-images/>



Dockerfile

```
FROM node:latest
```

```
WORKDIR /app
```

```
COPY . .
```

```
RUN npm i
```

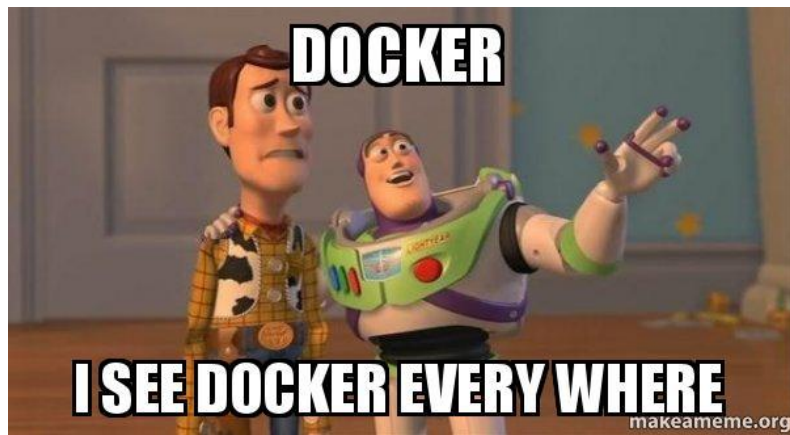
```
EXPOSE 3000
```

```
CMD ["node", "server.js"]
```




Criando a Imagem e Executando o Container

```
$ docker build -t biopark/api "."  
# docker run --name api -d -p  
3000:3000 biopark/api  
$ docker tag biopark/api  
account/image-name  
$ docker push account/image-name
```



CI/CD



O que é CI/CD?

Continuous Integration (CI)

Continuous Delivery (CD)

CI/CD é um método para entregar aplicações com frequência aos clientes.

Prevê a implementação da automação nas etapas de desenvolvimento de aplicações.



O que é CI/CD?

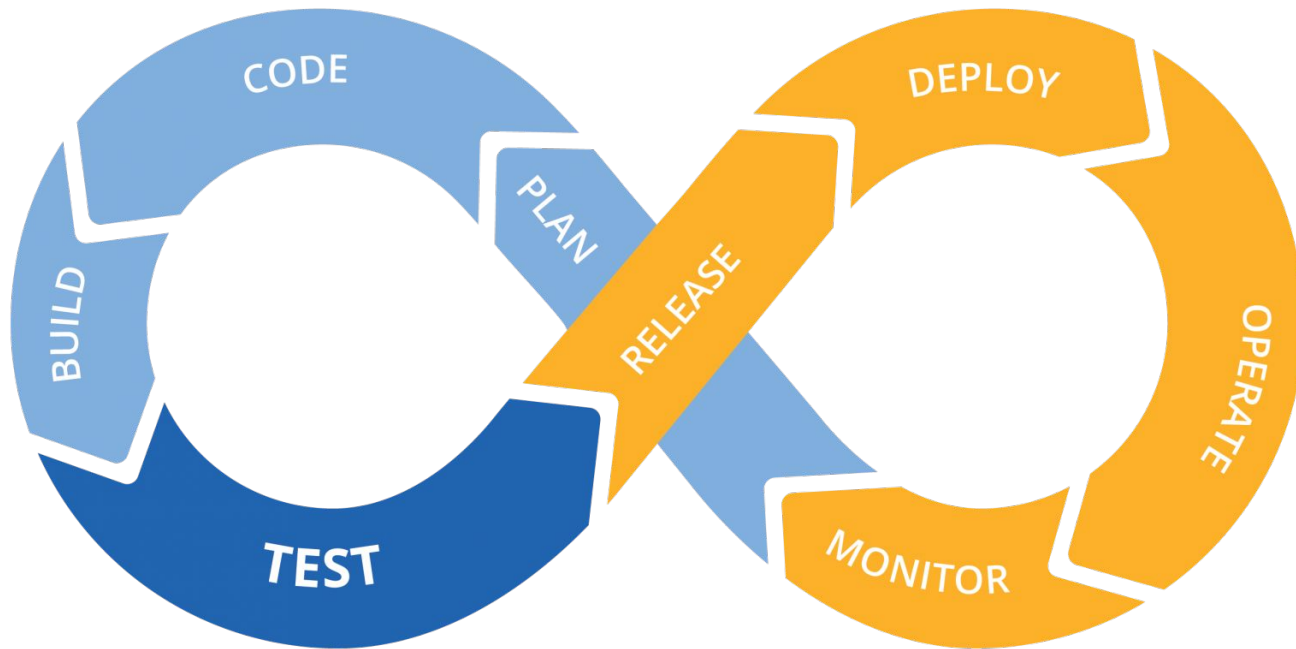
Continuous Integration (CI)

Continuous Delivery (CD)

CI/CD é um método para entregar aplicações com frequência aos clientes.

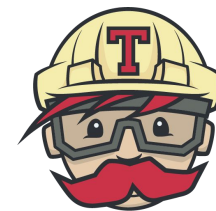
Prevê a implementação da automação nas etapas de desenvolvimento de aplicações.

O que é CI/CD?



Ferramentas de CI/CD

- AWS CodePipeline
- Azure Pipelines
- Google Cloud Build
- Bitbucket Pipelines
- GitLab CI/CD
- GitHub Actions
- CircleCI
- Travis CI
- Jenkins



GitHub Actions



O que é GitHub Actions?

GitHub Actions é uma plataforma de integração contínua e entrega contínua (CI/CD) que permite automatizar a sua compilação, testar e pipeline de implantação.

Possui disponíveis várias ações para realizar vários trabalhos (ações) ao projeto, bem como combinar ações em um fluxo de trabalho completamente personalizado.



O que é GitHub Actions?

GitHub fornece máquinas virtuais Linux, Windows e macOS para executar fluxos de trabalho.

É possível configurar no GitHub Actions um fluxo de trabalho para ser acionado quando um evento ocorre no repositório como, por exemplo, um pull request sendo aberto ou um problema sendo criado.

O seu fluxo de trabalho contém um ou mais trabalhos que podem ser executados em ordem sequencial ou em paralelo.



Estrutura Base

```
name: Java CI
```

```
on: [push]
```

```
jobs:
```

```
  build:
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

```
      - uses: actions/checkout@v3
```

```
      - name: Set up JDK 11
```

```
        uses: actions/setup-java@v3
```

```
        with:
```

```
          java-version: '11'
```

```
          distribution: 'adopt'
```

```
      - name: Build with Maven
```

```
        run: mvn package
```

[.github/workflows/my-ci.yml](#)

GitHub Actions Marketplace

Marketplace / Search results

Types

Apps

Actions

Categories

API management

Chat

Code quality

Code review

Continuous integration

Dependency management

Deployment

IDEs

Learning

Localization

Mobile

Monitoring

Q Search for apps and actions

Sort: Best Match

Actions

An entirely new way to automate your development workflow.

14596 results filtered by Actions

Actions



First interaction

By actions

Greet new contributors when they create their first issue or open their first pull request

☆ 140 stars



Setup .NET Core SDK

By actions

Used to build and publish .NET source. Set up a specific version of the .NET and authentication to private NuGet repository

☆ 501 stars



Upload a Build Artifact

By actions

Upload a build artifact that can be used by subsequent workflow steps

☆ 1.6k stars



Download a Build Artifact

By actions

Download a build artifact that was previously uploaded in the workflow by the upload-artifact action

☆ 539 stars



Setup Go environment

By actions

Setup a Go environment and add it to the PATH

☆ 809 stars



Setup Java JDK

By actions

Set up a specific version of the Java JDK and add the command-line tools to the PATH

☆ 784 stars

Hands on



Hands On

Preparação da Infra

- Realizar fork do projeto biopark-nodejs-api
- Criar novo Codespace com o projeto
- Criar um arquivo yaml para execução de actions que vão compilar e testar a aplicação, construir uma imagem docker e enviar para Docker Hub

A meme featuring a close-up of Donald Trump. He is wearing a dark suit, a white shirt, and a red striped tie. He is pointing his right index finger directly at the viewer with a serious, slightly open-mouthed expression. The background consists of dark wood paneling and a portion of a flag with a red star.

HEY DEVOPS TEAM

**PLEASE DEPLOY MY APPLICATION
TO DEV / PRODUCTION**

makeameme.org



Terraform



O que é?

"Terraform is an infrastructure as code (IaC) tool that allows you to build, change, and version infrastructure safely and efficiently."





Pode ser utilizada para gerenciar

Nuvens públicas:

- Amazon Web Services,
- Microsoft Azure e
- Google Cloud Platform

Nuvens privadas

- VMWare vSphere
- OpenStack
- CloudStack



Instalação

Install Terraform

Manual installation

Homebrew on OS X

Chocolatey on Windows

Linux

Retrieve the `terraform` binary by downloading a pre-compiled binary or compiling it from source.

Pre-compiled binary

Compile from source

To install Terraform, find the [appropriate package](#) for your system and download it as a zip archive.

After downloading Terraform, unzip the package. Terraform runs as a single binary named `terraform`. Any other files in the package can be safely removed and Terraform will still function.

Finally, make sure that the `terraform` binary is available on your `PATH`. This process will differ depending on your operating system.

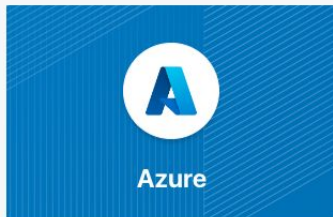
<https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli>

Provedores

<https://registry.terraform.io/browse/providers>

⌕ Providers

Providers are a logical abstraction of an upstream API. They are responsible for understanding API interactions and exposing resources.





Exemplo

```
terraform {
  required_providers {
    docker = {
      source  = "kreuzwerker/docker"
      version = "~> 2.13.0"
    }
  }
}

provider "docker" {}

resource "docker_image" "nginx" {
  name = "nginx:latest"
}

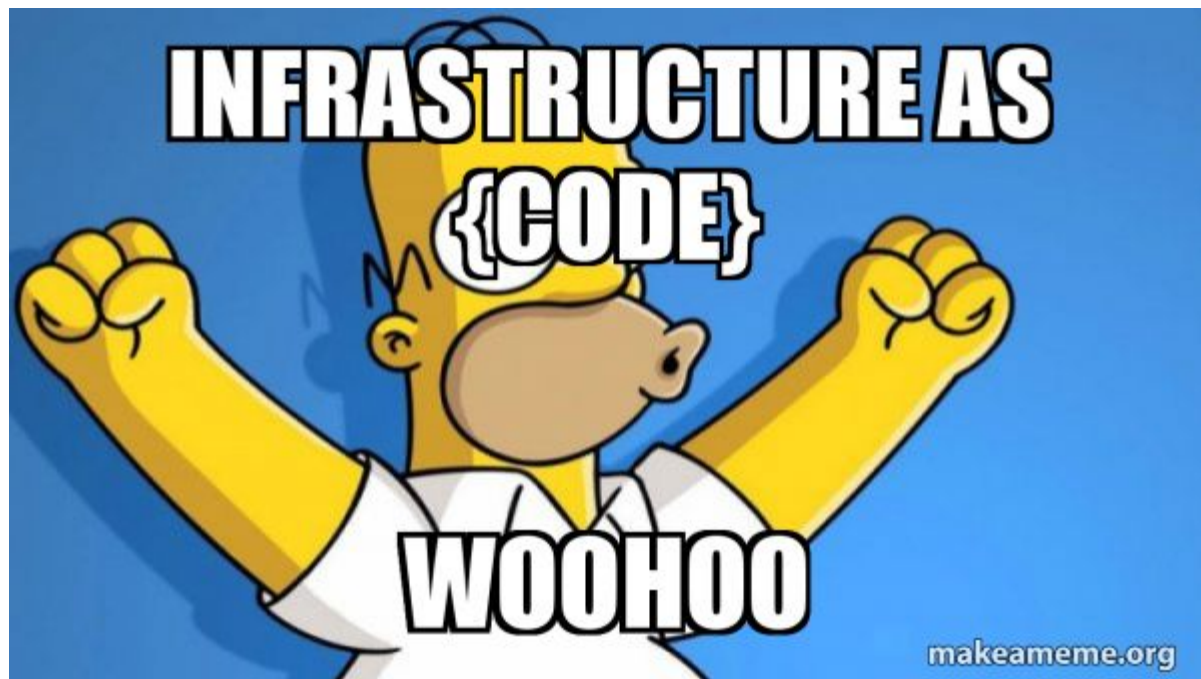
resource "docker_container" "nginx" {
  image = docker_image.nginx.latest
  name  = "proxy"
  ports {
    internal = 80
    external = 8000
  }
}
```

Hands on



Hands On

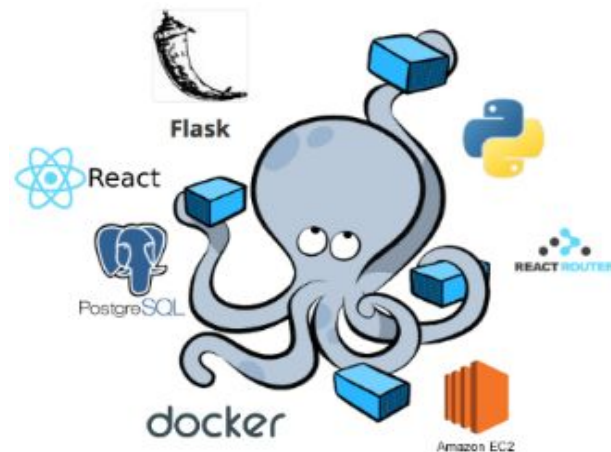
- Acessar Codespace no Github
- Clonar o repositório *biopark-iac* ([git@github.com:jfnandopr/biopark-iac.git](https://github.com/jfnandopr/biopark-iac))
- Instalar Terraform
- Instalar aws-cli
- Obter chaves na AWS e configurar aws-cli
 - `aws configure`
- Acessar pasta **terraform/aws**
- Criar a infraestrutura na AWS utilizando Terraform
 - `terraform init`
 - `terraform plan`
 - `terraform apply`



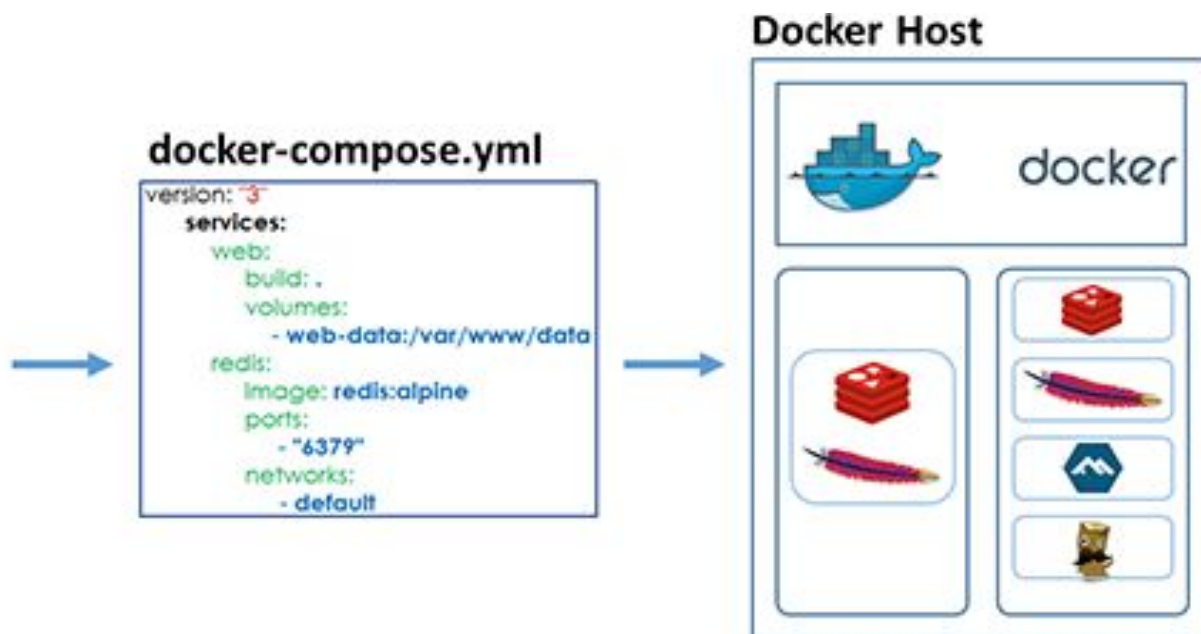
Docker Compose

O que é?

Docker Compose é o orquestrador de containers da Docker.



Estrutura





Estrutura

```
services:
  web:
    image: wordpress
    restart: always
    depends_on:
      - db
    ports:
      - 80:80
    networks:
      - frontend
      - backend
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: root
      WORDPRESS_DB_PASSWORD: 123456
      WORDPRESS_DB_NAME: wordpress
    volumes:
      - wordpress:/var/www/html

  db:
    image: mysql:5.7
    restart: always
    environment:
      MYSQL_DATABASE: wordpress
      MYSQL_ROOT_PASSWORD: '123456'
    volumes:
      - db:/var/lib/mysql
    networks:
      - backend
```

```
volumes:
  wordpress:
  db:

networks:
  frontend:
  backend:
```



Commandos

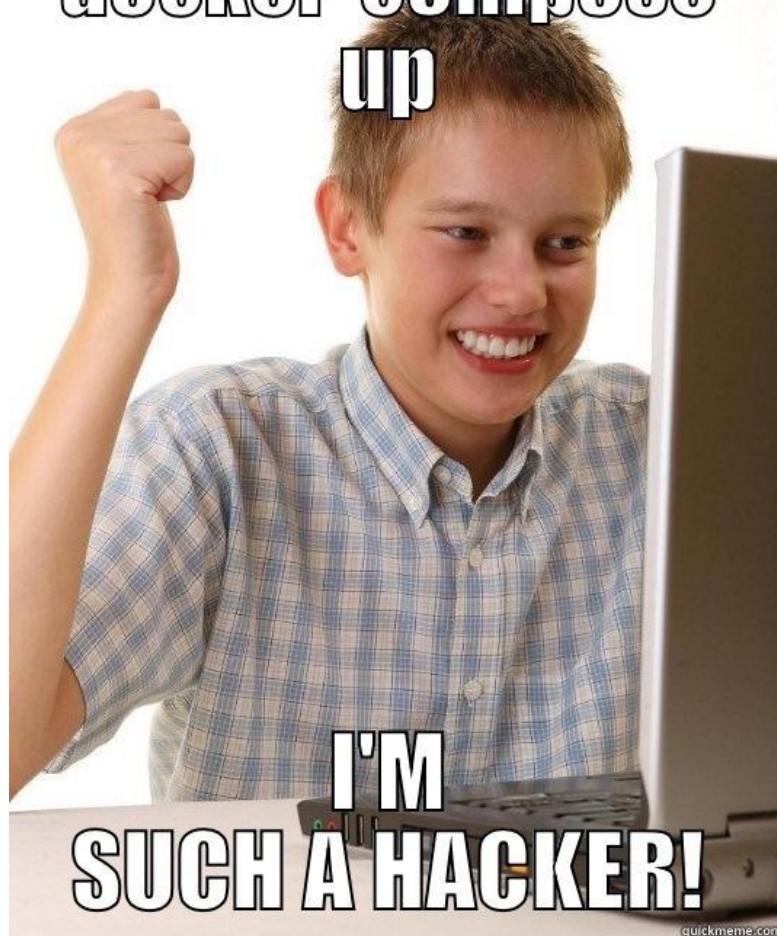
```
$ docker-compose up -d
$ docker-compose start SERVICE_NAME
$ docker-compose stop SERVICE_NAME
$ docker-compose restart SERVICE_NAME
$ docker-compose down
$ docker-compose ps
$ docker-compose build
$ docker-compose up -d -build
```



Hands On

- Acessar a instância EC2 na AWS
- Criar um arquivo docker-compose conforme exemplo
- Iniciar serviços

**docker-compose
up**



**I'M
SUCH A HACKER!**

Trabalho



Atividade

Executar um container da imagem com a aplicação biopark-nodejs-api



Atividade Final

Gravar um vídeo demonstrando a execução do Hands on do conteúdo abaixo

- Se necessário, buscar materiais de apoio, vídeos, aulas
- Enviar até 27/11



Ansible

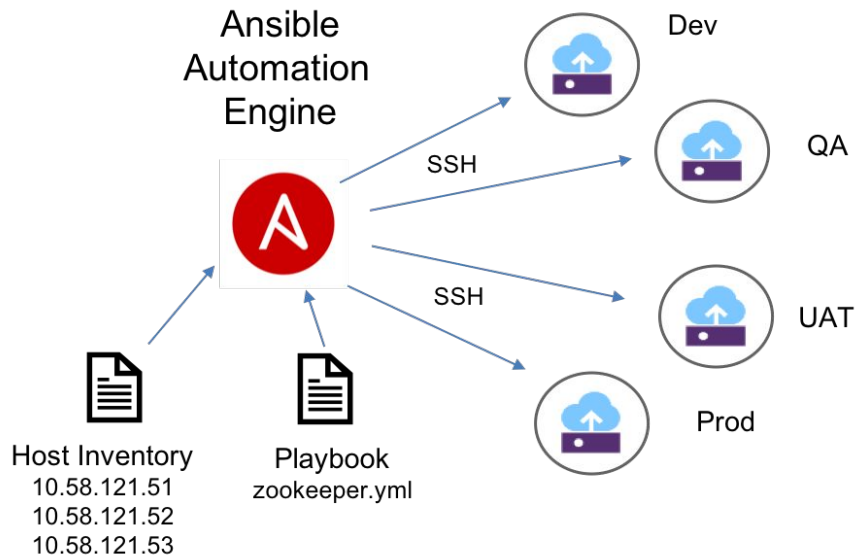


O que é?

O Ansible é um mecanismo de automação de TI open source para automação de processos como provisionamento, gerenciamento de configurações, implantação de aplicações, orquestração e muitos outros.

<https://docs.ansible.com/>

Como funciona o Ansible?





Inventory

Os inventários organizam os nós gerenciados em arquivos centralizados que fornecem ao Ansible informações do sistema e locais de rede.



Inventory

Os inventários organizam os nós gerenciados em arquivos centralizados que fornecem ao Ansible informações do sistema e locais de rede.

```
ansible-inventory -i inventory.yaml --list
```

```
ansible webserver01 -m ping -i inventory.yaml
```



Playbook

Os playbooks são blueprints de automação, em formato YAML, que o Ansible usa para implantar e configurar nós gerenciados.

```
ansible-inventory -i inventory.yaml --list
```

```
ansible webserver01 -m ping -i inventory.yaml
```




Algumas definições

- Tasks
 - menor unidade de trabalho, uma ação
 - exemplo: "Instalar docker"
- Plays
 - conjunto de tarefas
 - exemplo: "Preparar aplicação web"
- Playbook
 - conjunto de plays
 - exemplo: "Preparar servidor para a aplicação X"
 - "Preparar aplicação web"
 - "Preparar banco de dados"

Hands on



Hands On

- Acessar CodeSpace
- Clonar o repositório *biopark-iac*
- Instalar Ansible
- Acessar pasta **ansible**
- Atualizar o arquivo inventory.yaml com DNS IPv4 público do host criado
 - `ansible-inventory -i inventory.yaml --list`



Hands On

- Acessar CodeSpace
- Acessar pasta **ansible**
 - `ansible-playbook -i inventory.yaml aws/install-docker.yaml`
 - `ansible-playbook -i inventory.yaml aws/install-site.yaml`



Sugestão de Projetos

https://github.com/rayed/django_crud Python

<https://github.com/bhiapp4/crudApp> Java

<https://github.com/zianwar/springboot-crud-demo> Java

<https://github.com/LisandroVillarroel/proyectoSCPC> JavaScript

<https://github.com/Bunkermaster/crud-webp2020-g1> PHP

<https://github.com/rubydocs/app> Ruby

https://github.com/herusdianto/gorm_crud_example Go



Kwai

ID:casalcrossp...



**MAS É UM
PROCESSO
SOLITÁRIO E
DEMORADO.**

Obrigado!!!

O processo pode ser
lento, mas desistir não
acelera!

— Surama Jurdi