



# Engenharia de Software para Nuvem - Aula 3

Jemerson Fernando Maia - [jfnandopr@gmail.com](mailto:jfnandopr@gmail.com)

Pós-graduação em Engenharia de Software para Modernização de Sistemas

BIOPARK EDUCAÇÃO

---

# Agenda

- Terraform
- Ansible
- CI/CD
- Atividade





# 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.

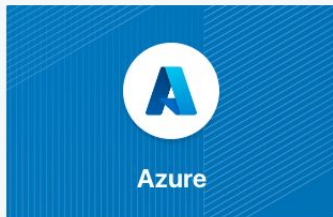
<https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli>

# 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

---



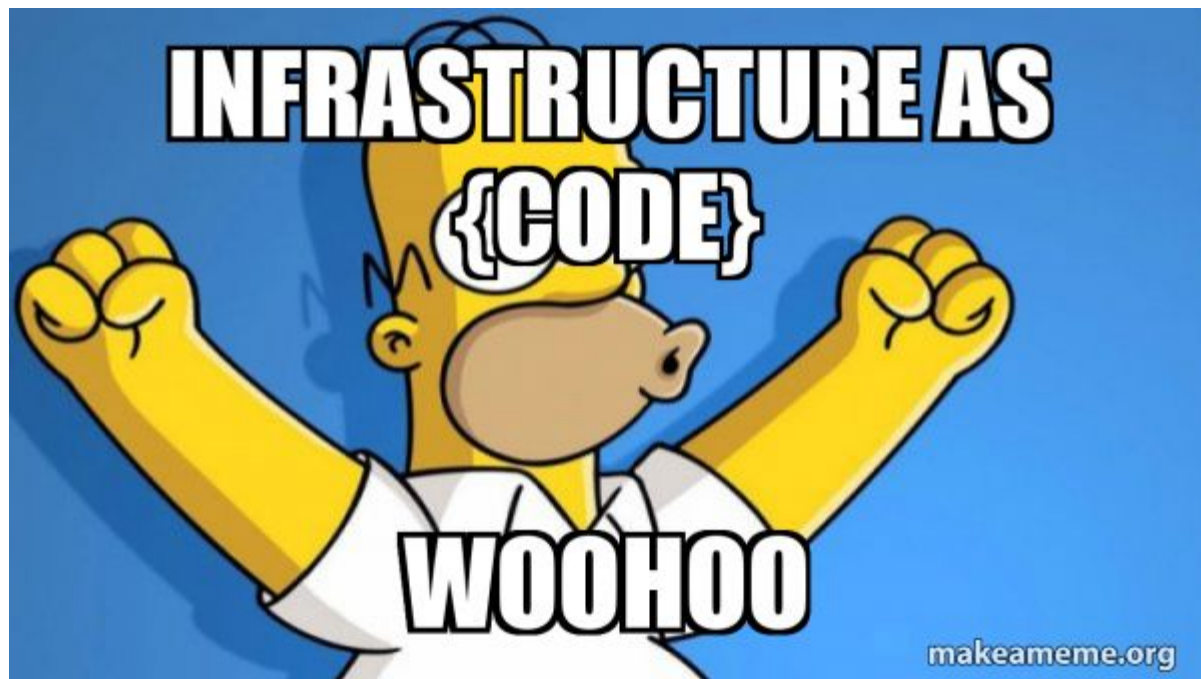
## Hands On

- Acessar CodeSpace
- Clonar o repositório *biopark-iac*
- Instalar Terraform
- Acessar pasta **terraform/docker**
- Criar container com terraform
  - `terraform init`
  - `terraform plan`
  - `terraform apply`



## Hands On

- Instalar aws-cli
- Configurar aws-cli
  - `aws configure`
- Acessar pasta **terraform/aws**
- Criar a infraestrutura na AWS utilizando Terraform
  - `terraform init`
  - `terraform plan`
  - `terraform apply`





**Ansible**

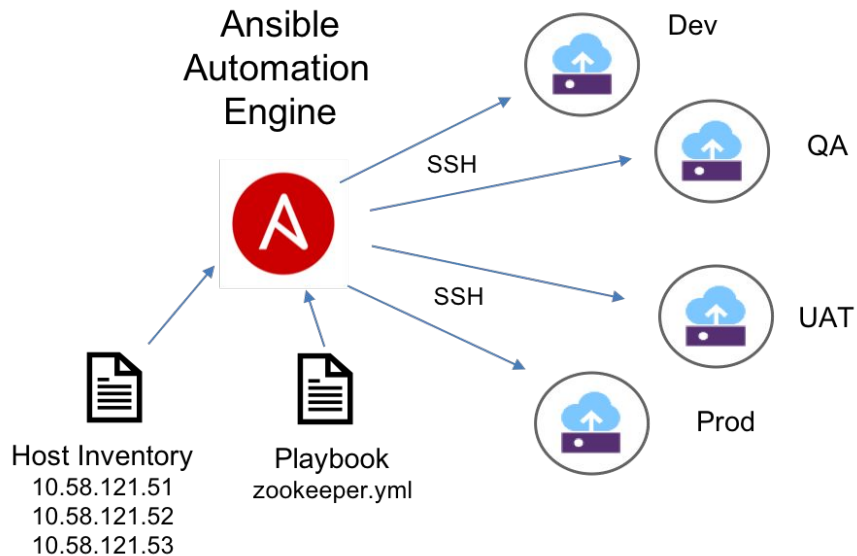


## O que é?

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



# 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`

**I SHOULD**



**DO THIS WITH ANSIBLE**

memegenerator.net

---

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?

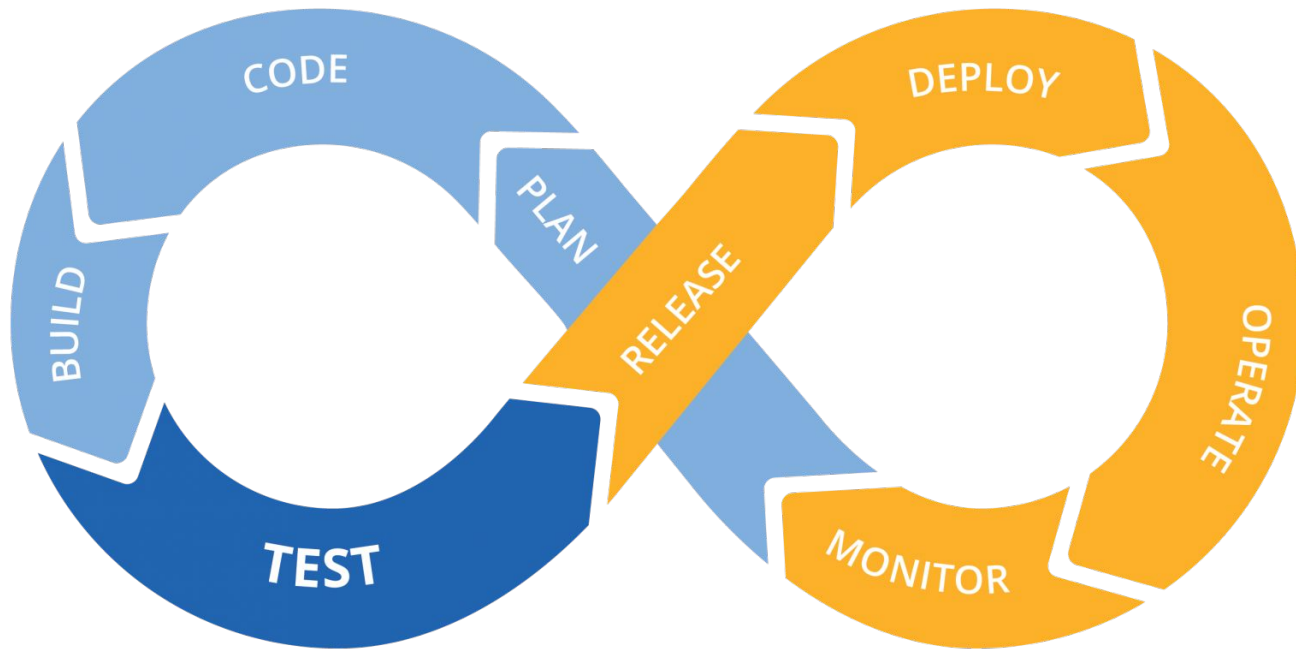
Continuous Integration (CI)

Continuous Delivery (CD)

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

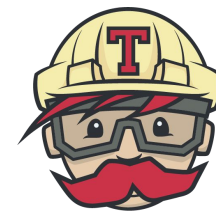
Prevê a implementação da automação nas etapas de desenvolvimento de aplicações.

## 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

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

- Localizar repositório **biopark-cicd**
- Realizar *fork* do repositório
- Acessar CodeSpace
- Clonar o repositório **biopark-cicd**
- Preparar a infra estrutura na AWS
  - Make



# Hands On

## Execução do CICD

- Criar as chaves de acesso no repositório
- Atualizar nome da instância no arquivo de cicd.yml
- Renomear a pasta \_workflows para workflows
- Commitar

Ao executar o pipeline será feito:

- Build da aplicação e execução de testes
- Criação da imagem docker
- Deploy da imagem no Docker Hub
- Execução da aplicação na instância EC2

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 is a blurred interior with wood paneling and a flag with a red star.

**HEY DEVOPS TEAM**

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

makeameme.org

# Trabalho

---



## Atividade Final

Em grupo:

- Utilizando um projeto da lista disponibilizada no próximo slide
  - Criar CI/CD via GitHub Actions
    - CI deverá executar testes
    - CD deverá empacotar a aplicação em um container e enviar para o Docker Hub
    - CD deverá rodar a aplicação no host EC2 da AWS



## Sugestão de Projetos

[https://github.com/rayed/django\\_crud](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](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