



DevOps

Automação e Infraestrutura como Código (IaC)



O que é Infraestrutura como Código?

Infraestrutura como Código (IaC) é a prática de gerenciar e provisionar infraestrutura de TI por meio de código, em vez de processos manuais.

- Infraestrutura tratada como software, escrita em arquivos de configuração
- Automatiza provisionamento, configuração e gerenciamento de servidores
- Permite versão, reutilização e colaboração

Benefícios

- **Consistência** - Evita configurações manuais erradas
- **Automatização** - Reduz tempo de provisionamento
- **Escalabilidade** - Facilita crescimento da infraestrutura
- **Versionamento** - Infraestrutura rastreável como código-fonte



On-Premises vs IaC

On-Premises?

Infraestrutura gerenciada localmente dentro da empresa, com servidores físicos, armazenamento e redes próprias.

Vantagens

- Maior controle sobre hardware e segurança
- Baixa latência para sistemas internos
- Independência de provedores de nuvem

Desvantagens

- Alto custo e manutenção constante
- Escalabilidade limitada e lenta
- Atualizações e provisionamento manuais



On-Premises vs IaC

Característica	On-Premises	IaC
Custo	Alto (hardware próprio)	Baixo (paga conforme uso)
Escalabilidade	Limitada e demorada	Automática e rápida
Automação	Baixa	Alta
Gerenciamento	Manual	Automatizado via código
Tempo de Provisionamento	Dias/semanas	Minutos
Flexibilidade	Baixa	Alta



Ferramentas - Terraform

O **Terraform** é uma ferramenta de **orquestração de infraestrutura** criada pela HashiCorp. Ele permite definir, provisionar e gerenciar infraestrutura utilizando um código declarativo.

Principais Características:

- **Código Declarativo:** Define o estado desejado da infraestrutura, e o Terraform cuida do resto.
- **Multi-Cloud:** Suporta **AWS, Azure, Google Cloud, Kubernetes, VMware**, entre outros.
- **Versionamento e Controle de Estado:** Usa arquivos **.tfstate** para manter a infraestrutura sincronizada.
- **Infraestrutura Imutável:** Se precisar alterar algo, ele destrói e recria os recursos corretamente.



Ferramentas - Ansible (Red Hat)

O **Ansible** é uma ferramenta de **automação de configuração** e **gerenciamento de servidores**, desenvolvida pela Red Hat. Ele permite automatizar tarefas como instalação de pacotes, configuração de servidores e gerenciamento de aplicações.

Principais Características:

- **Baseado em YAML:** Usa **Playbooks** para definir configurações.
- **Agentless:** Não precisa instalar agentes nos servidores gerenciados.
- **Fácil Integração:** Funciona bem com AWS, Azure, Google Cloud, Kubernetes, Docker, etc.
- **Ideal para Configuração:** Enquanto o Terraform cria a infraestrutura, o Ansible configura os servidores dentro dela.



Ferramentas - AWS CloudFormation

O **AWS CloudFormation** é uma ferramenta da própria **Amazon Web Services (AWS)** para criar e gerenciar infraestrutura na nuvem **usando código**.

Principais Características:

- **Específico para AWS:** Diferente do Terraform, ele **só funciona na AWS**.
- **Baseado em JSON ou YAML:** Os arquivos descrevem a infraestrutura desejada.
- **Automação Completa:** Pode criar servidores, redes, bancos de dados e até aplicações inteiras.
- **Gerenciamento de Stack:** Permite definir toda a infraestrutura como um conjunto de recursos interligados.



Comparação



AWS CloudFormation

Resources:

MyEC2Instance:

Type: AWS::EC2::Instance

Properties:

ImageId: ami-12345678

InstanceType: t2.micro

Ansible

```
- name: Instalar Nginx
hosts: servidores
become: yes
tasks:
  - name: Atualizar pacotes
    apt:
      update_cache: yes
  - name: Instalar Nginx
    apt:
      name: nginx
      state: present
```

Terraform

```
provider "aws" {
  region = "us-east-1"
}

resource "aws_instance" "example" {
  ami           = "ami-12345678"
  instance_type = "t2.micro"
}
```




Prática - Terraform

Clonar repositório <https://github.com/paeeglee/devops-unisatc-iac>