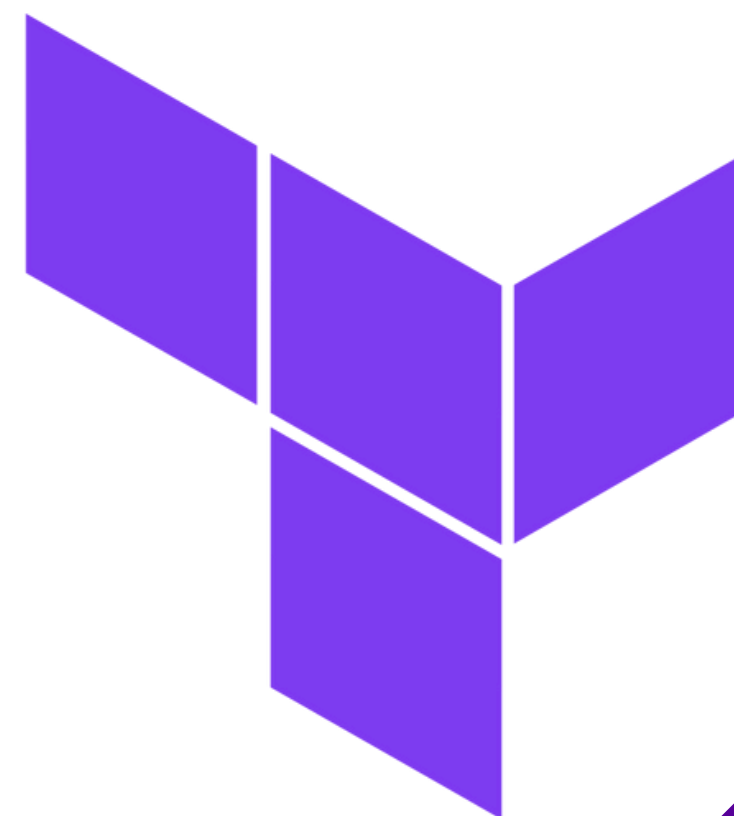




APRESENTANDO O

TERRAFORM



Daniel Mantovani

Gabriel Santos

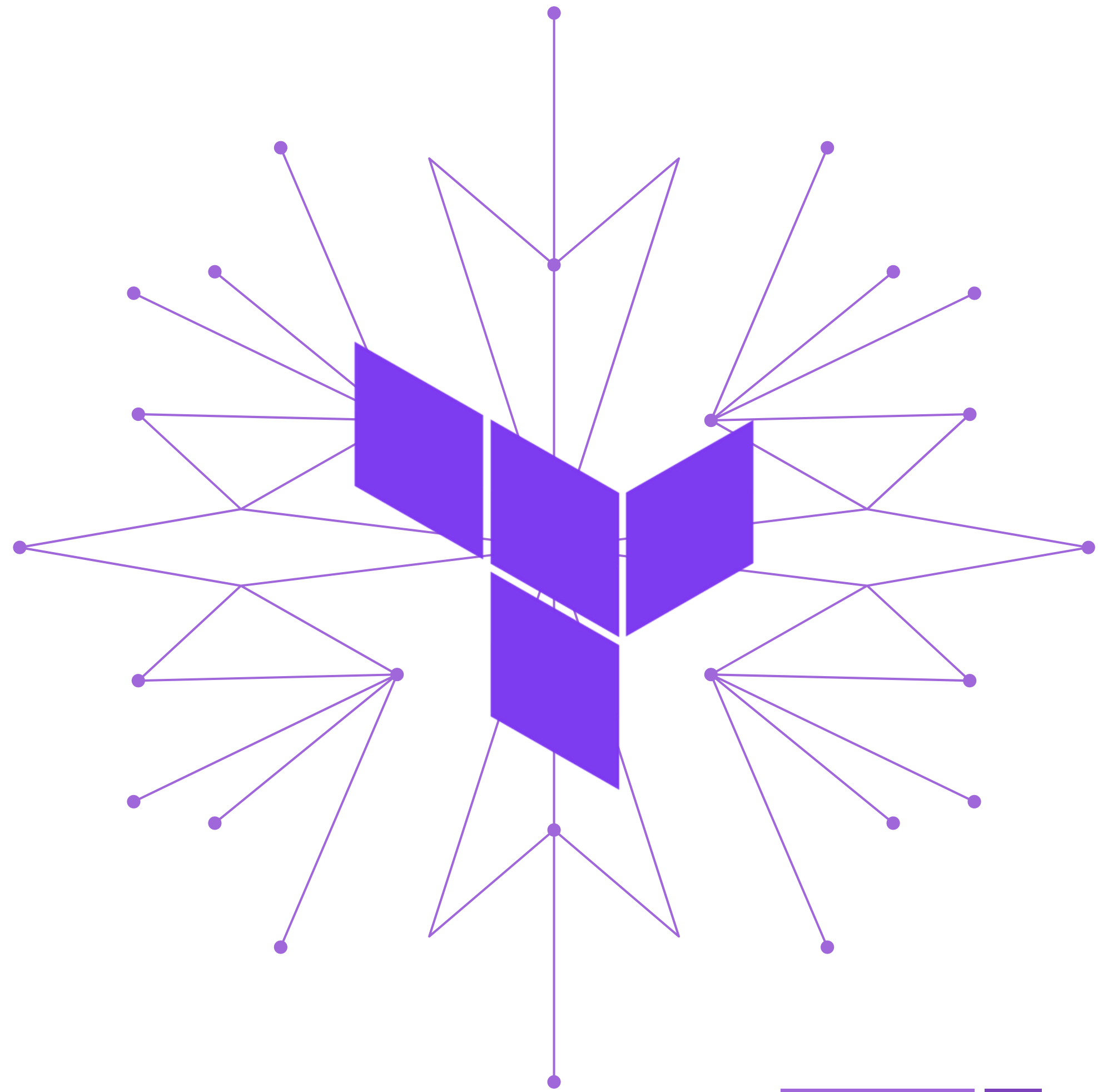
Iury Barbosa

José Vitor

A series of parallel, wavy purple lines that flow from the bottom right corner towards the center of the page, creating a dynamic, organic feel.

SUMÁRIO

1. Descrição da ferramenta
2. Vantagens e limitações
3. Instalação e configuração
4. Exemplos práticos



O QUE É TERRAFOM

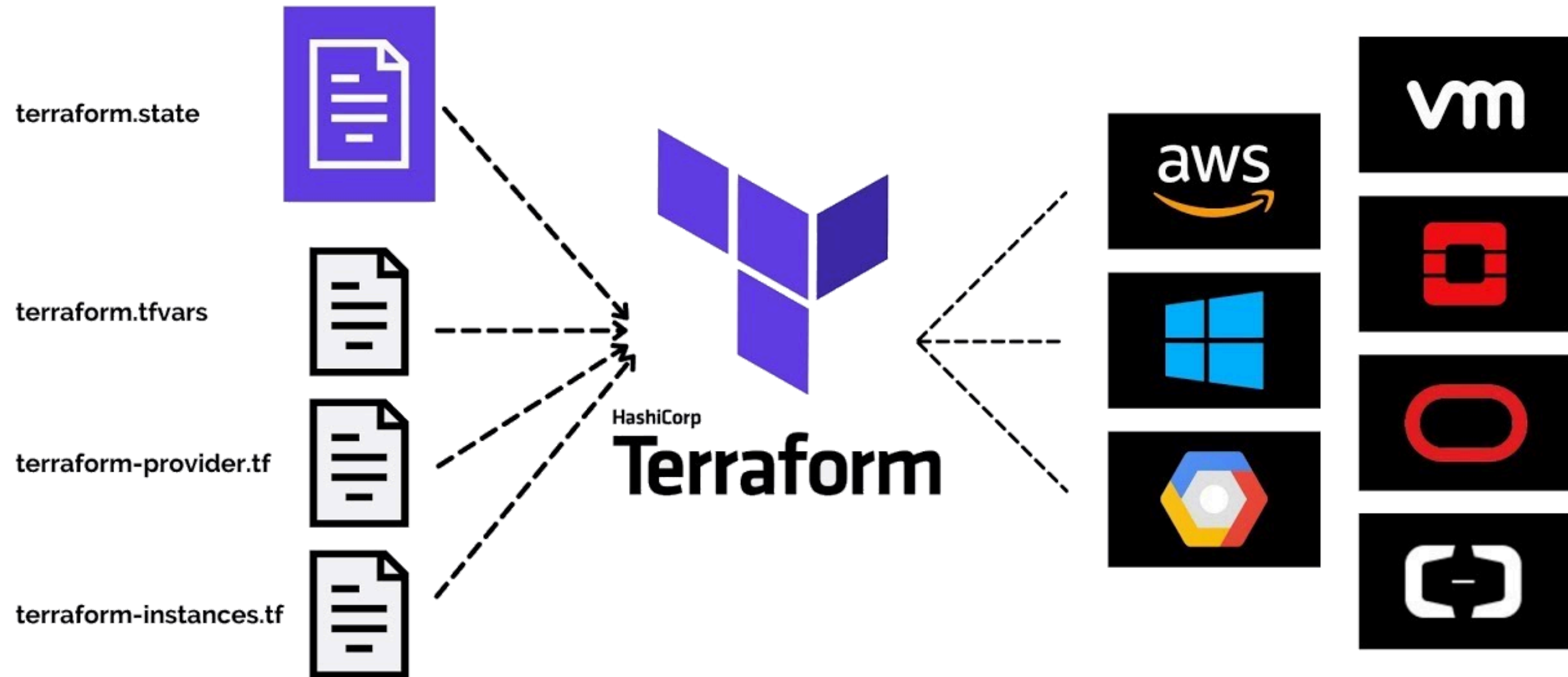
Terraform é uma ferramenta de **Infrastructure as Code (IaC)**

Gerenciar infraestruturas por meio de arquivos **.tf**
(como servidores, bancos de dados, etc.)

HCL (HashiCorp Configuration Language)



```
terraform {  
  required_providers {  
    google = {  
      source = "hashicorp/google"  
      version = "3.5.0"  
    }  
  }  
}  
  
provider "google" {  
  credentials = file("<NAME>.json")  
  
  project = "<PROJECT_ID>"  
  region  = "us-central1"  
  zone    = "us-central1-c"  
}  
  
resource "google_compute_network" "vpc_network" {  
  name = "terraform-network"  
}
```



TL;DR

Configurar diversas infraestruturas é complicado. O Terraform faz isso para você por meio de simples arquivos de configuração.

TERRAFOM REGISTRY



Registry

Browse ▾

Publish ▾

Sign-in



Use HCP Terraform for free ↗

Providers Modules Policy Libraries Run Tasks

Filters

[Clear Filters](#)

Tier ⓘ

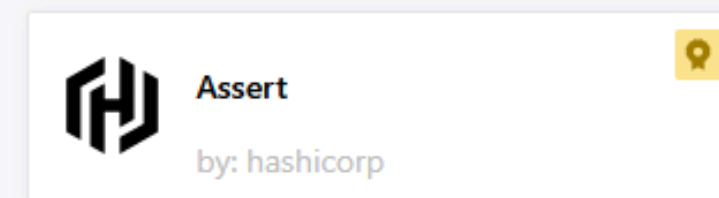
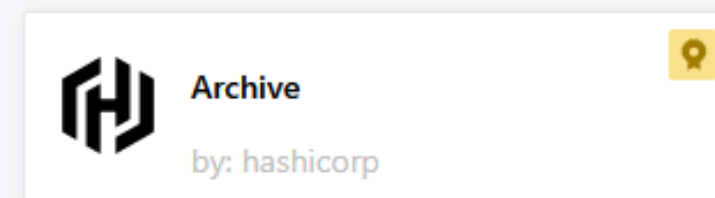
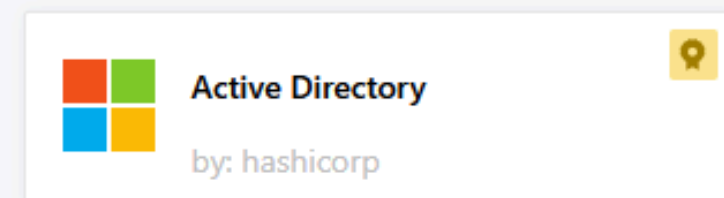
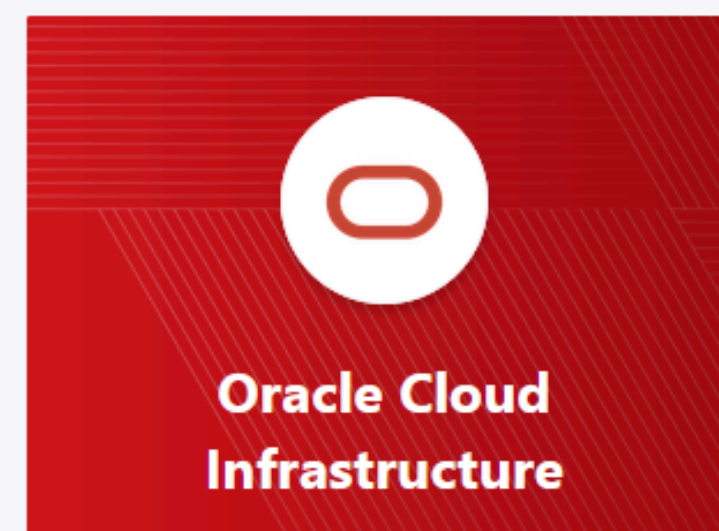
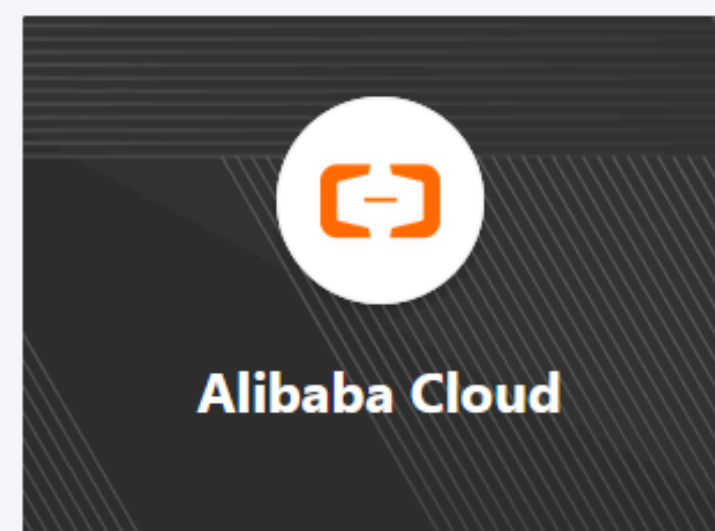
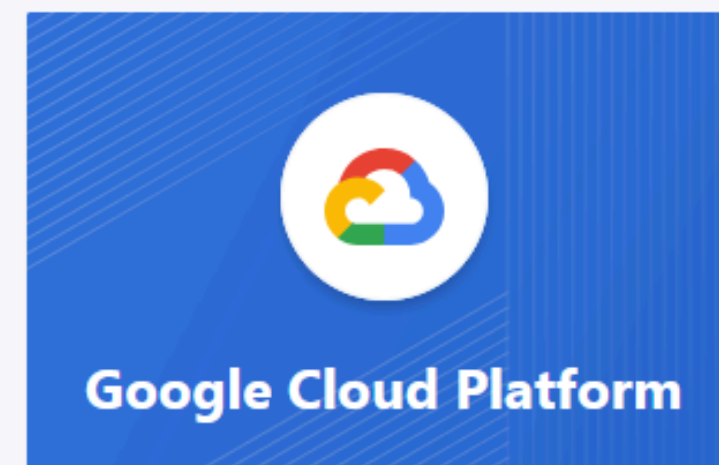
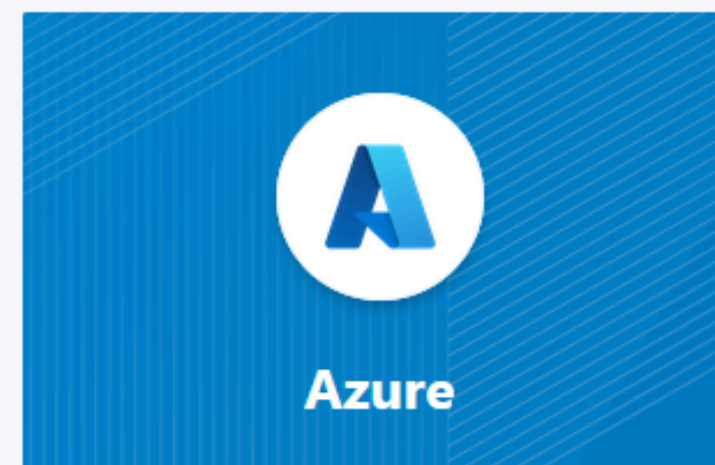
- ☒ Official
- ☒ Partner
- ☒ Community

Category

- ☐ HashiCorp Platform
- ☐ Infrastructure Management
- ☐ Public Cloud
- ☐ Asset Management
- ☐ Cloud Automation
- ☐ Communication & Messaging
- ☐ Container Orchestration
- ☐ Continuous Integration/Deployment (CI/CD)
- ☐ Data Management
- ☐ Database
- ☐ Infrastructure (IaaS)
- ☐ Logging & Monitoring
- ☐ Networking
- ☐ Platform (PaaS)
- ☐ Security

Providers

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



TERRAFOM REGISTRY



Registry

Browse ▾

Publish ▾

Sign-in



Use HCP Terraform for free ↗

Providers / hashicorp / aws / Version 5.94.1 ▾ Latest Version

aws

Overview

Documentation

USE PROVIDER ▾



aws

Official

by: HashiCorp

Public Cloud

Lifecycle management of AWS resources, including EC2, Lambda, EKS, ECS, VPC, S3, RDS, DynamoDB, and more. This provider is maintained internally by the HashiCorp AWS Provider team.

VERSION

5.94.1

PUBLISHED

4 days ago

SOURCE CODE

hashicorp/terraform-provider-aws

How to use this provider

To install this provider, copy and paste this code into your Terraform configuration. Then, run `terraform init`.


Terraform 0.13+

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.94.1"
    }
  }
}

provider "aws" {
  # Configuration options
}
```

Visão geral do Terraform no Azure - o que é Terraform?

Artigo • 14/10/2024 • 4 colaboradores

 Comentários

Neste artigo

[Provedores do Terraform para infraestrutura do Azure](#)

[Benefícios do Terraform com o Azure](#)


[Próximas etapas](#)

[Documentação](#) > [AWS Service Catalog](#) > [Guia do administrador](#)

Conceitos básicos de um produto Terraform

 PDF

☐ Modo de foco

 As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.

TERRAFORM WORKFLOW

Init - Inicializa os plugins necessários

Plan - Visualize as alterações antes de aplicá-las

Apply - Aplique a infraestrutura reproduzível

Destroy - Remove completamente os recursos criados



PROPÓSITO

- **Automatização**
- **Reprodutibilidade**
- **Controle de versão**
- **Redução de erros**
- **Colaboração**



Terraform

The Terraform logo is a white, stylized, isometric icon consisting of three rectangular blocks arranged in a 3D structure. The word "Terraform" is written in a large, white, sans-serif font below the logo.

VANTAGENS

- Portabilidade entre plataformas
- Documentação detalhada
- Utilização da HCL
- Automação e facilidade de uso
- Módulos Terraform

bash

Exemplo 1: Provedor de Nuvem (AWS - EC2)

Linguagem: TypeScript com CDKTF

```
import { Construct } from 'constructs';
import { App, TerraformStack } from 'cdktf';
import { AwsProvider, Instance } from '@cdktf/provider-aws';

class AwsStack extends TerraformStack {
  constructor(scope: Construct, id: string) {
    super(scope, id);

    new AwsProvider(this, 'AWS', {
      region: 'us-east-1',
    });

    new Instance(this, 'MyEC2Instance', {
      ami: 'ami-0c55b159cbfafa1f0',
      instanceType: 't2.micro',
    });
  }
}

const app = new App();
new AwsStack(app, 'aws-example');
app.synth();
```

Esse código cria uma instância EC2 na AWS

VANTAGENS

- Portabilidade entre plataformas
- Documentação detalhada
- Utilização da HCL
- Automação e facilidade de uso
- Módulos Terraform

bash

Exemplo 2: Criando uma VM local no VSphere

Linguagem: Python com CDKTF

```
class VsphereStack(TerraformStack):
    def __init__(self, scope: Construct, ns: str):
        super().__init__(scope, ns)
        # Instanciando o provedor vSphere
        VsphereProvider(self, "vSphere",
            user="administrator@vsphere.local",
            password="senha-super-secreta",
            vsphere_server="192.168.0.10",
            allow_unverified_ssl=True
        )
        # Definição de uma VM
        VsphereVirtualMachine(self, "MinhaVmBasica",
            name="vm-simples",
            resource_pool_id="resgroup-1",
            datastore_id="datastore-1",
            num_cpus=1,
            memory=512,
            guest_id="otherGuest",
            disk=[VsphereVirtualMachineDisk(
                label="disco-vm",
                size=10
            )],
            network_interface=[VsphereVirtualMachineNetworkInterface(
                network_id="network-1"
            )]
        )
        app = App()
        VsphereStack(app, "vsphere-exemplo-simples")
        app.synth()
```

Esse código provisiona uma máquina virtual local em uma instância do VMware

VANTAGENS

- Portabilidade entre plataformas
- Documentação detalhada
- Utilização da HCL
- Automação e facilidade de uso
- Módulos Terraform

About Terraform

Define cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share.

Featured docs

[Introduction to Terraform](#)

[Configuration Language](#)

[Terraform CLI](#)

[HCP Terraform](#)

[Provider Use](#)

VANTAGENS

- Portabilidade entre plataformas
- Documentação detalhada
- **Utilização da HCL**
- Automação e facilidade de uso
- Módulos Terraform

```
bash      Exemplo 1: Provedor de Nuvem (AWS - EC2)
          Linguagem: HCL

terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
  required_version = ">= 1.0.0"
}

provider "aws" {
  region = var.region
}

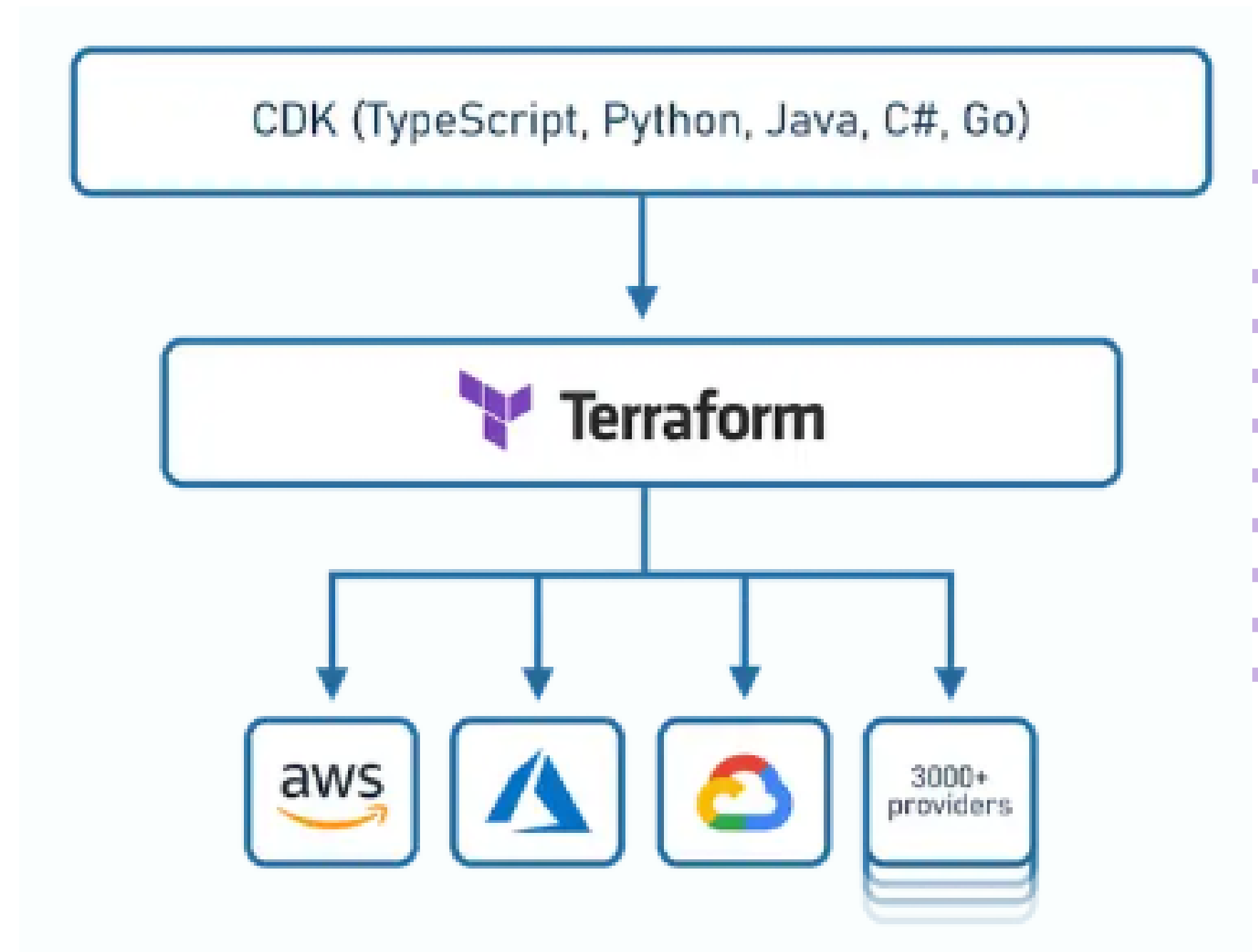
resource "aws_instance" "MyEC2Instance" {
  ami          = var.ami
  instance_type = var.instance_type

  tags = {
    Name = "CDKTF-to-HCL-Instance"
  }
}
```

Esse código cria uma instância EC2 na AWS

VANTAGENS

- Portabilidade entre plataformas
- Documentação detalhada
- **Utilização da HCL**
- Automação e facilidade de uso
- Módulos Terraform



VANTAGENS

- Portabilidade entre plataformas
- Documentação detalhada
- Utilização da HCL
- Automação e facilidade de uso
- Módulos Terraform

```
bash      Exemplo 1: Provedor de Nuvem (AWS - EC2)
          Linguagem: TypeScript com CDKTF

import { Construct } from 'constructs';
import { App, TerraformStack } from 'cdktf';
import { AwsProvider, Instance } from '@cdktf/provider-aws';

class AwsStack extends TerraformStack {
  constructor(scope: Construct, id: string) {
    super(scope, id);

    new AwsProvider(this, 'AWS', {
      region: 'us-east-1',
    });

    new Instance(this, 'MyEC2Instance', {
      ami: 'ami-0c55b159cbfafa1f0',
      instanceType: 't2.micro',
    });
  }
}

const app = new App();
new AwsStack(app, 'aws-example');
app.synth();
```

Esse código cria uma instância EC2 simples na AWS


VANTAGENS

- Portabilidade entre plataformas
- Documentação detalhada
- Utilização da HCL
- Automação e facilidade de uso
- Módulos Terraform

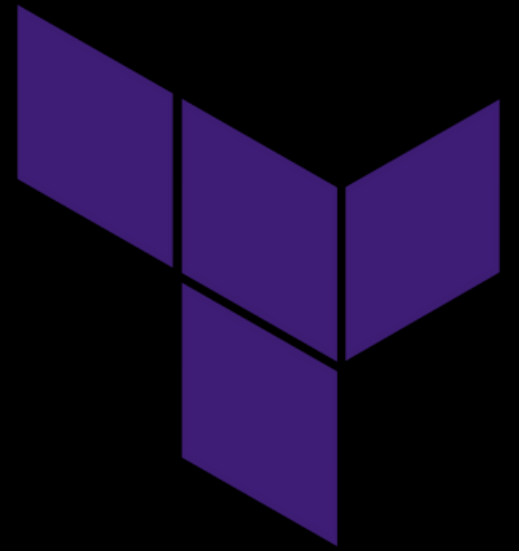
**Imperativo**

1. Criar rede
2. Criar sub-rede
3. Criar VM
4. Associar IP
5. Instalar pacote
6. Configurar DNS
- ...

VS.

**Declarativo**

```
resource "aws_instance" "web" {  
  ami      = "ami-123"      # image SO  
  instance_type = "t2.micro" # config da VM  
}
```



Você descreve o que quer. O Terraform descobre como chegar lá.

VANTAGENS

- Portabilidade entre plataformas
- Documentação detalhada
- Utilização da HCL
- Automação e facilidade de uso
- Módulos Terraform

```
bash

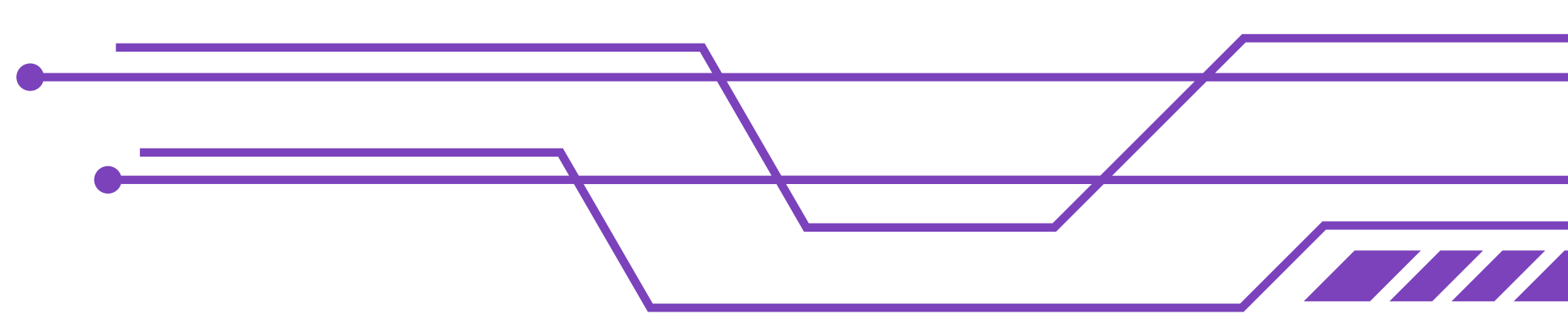
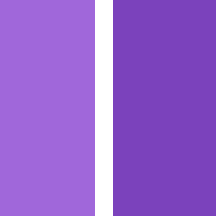
infra/
├─ main.tf                # Chama os módulos e conecta os recursos
├─ variables.tf
├─ outputs.tf
├─ modules/
│   ├─ networking/        # Módulo de rede
│   │   └─ main.tf         # Cria VPC, sub-redes, etc.
│   │   └─ variables.tf
│   ├─ application/       # Módulo da aplicação web
│   │   └─ main.tf         # Cria EC2, ASG, ELB
│   │   └─ variables.tf
│   └─ database/          # Módulo de banco de dados
│       └─ main.tf         # Cria instância RDS, SGs, etc.
│       └─ variables.tf
```

Padronização de componentes, melhorando manutenção e a consistência entre ambientes.

LIMITAÇÕES

- Curva de aprendizagem difícil
- Gestão complexa de Estado
- Gerenciamento de Versão
- Problemas com plugins de parceiros

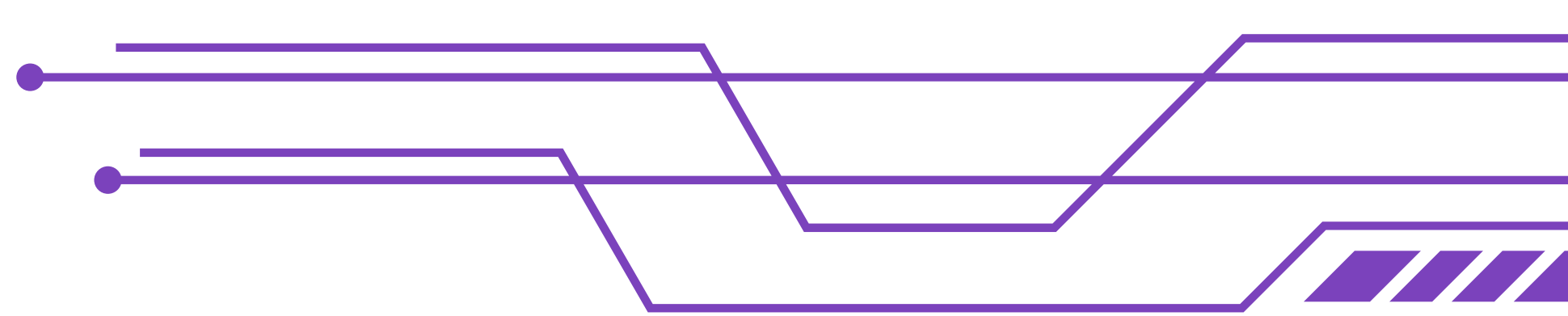
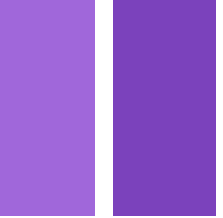




INSTALAÇÃO

Sistemas operacionais disponiveis:

- Windows
- macOS
- Linux(Ubuntu/Debian, CentOS/RHEL, fedora, amazon linux)
- FreeBSD
- OpenBSD
- Solaris

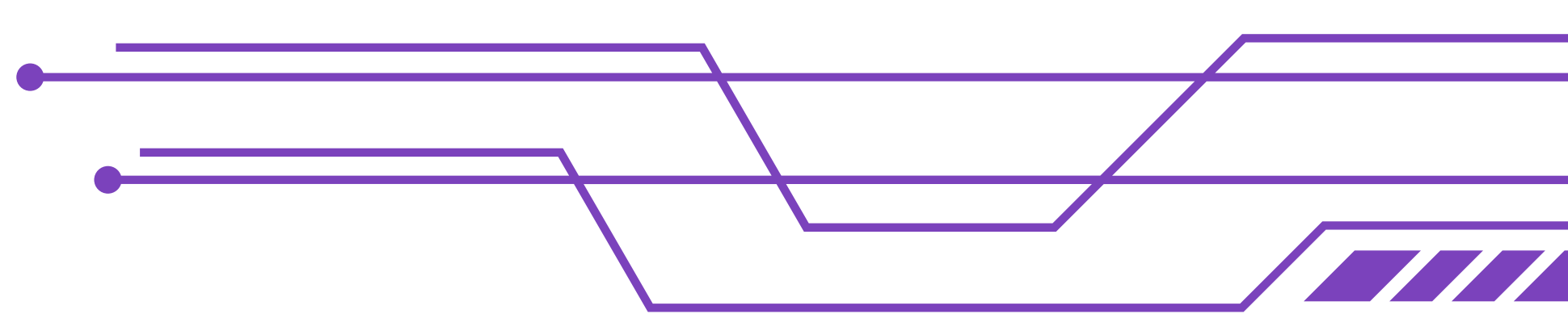
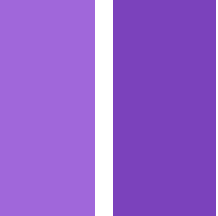


INSTALAÇÃO - LINUX (FEDORA 41)

```
sudo dnf install -y dnf-plugins-core
```

```
sudo dnf config-manager addrepo --from-repofile=https://rpm.releases.hashicorp.com/fedora/  
hashicorp.repo
```

```
sudo dnf -y install terraform
```

DEMONSTRAÇÃO DE APLICAÇÃO

Nosso cenário:

- Empresa de médio porte
- Burocracia >> código
- Equipe de QA
- Aplicações complexas (custo alto para correção de bugs)



OBRIGADO

Fonte: <https://developer.hashicorp.com/terraform>

Material disponível em:

github.com/vitorbarbosa123/terraform-sysadmin