

From Zero to Useful - Terraform para Sysadmins

**Como começar a usar o Terraform
agora para automatizar seu dia a dia**



Olá!

Eu sou o VAL

*E eu ajudo pessoas a crescerem e evoluirem
em suas carreiras através de comunidades!*

SOLUTION ARCHITECT @ UHG BRASIL

NERD · MARIDO · PAI DA MARI E DO JOÃO

BLOGGER · PODCASTER · SPEAKER
COMMUNITY LOVER · FOTÓGRAFO APOSENTADO
OLDSCHOOL GAMER · CERVEJEIRO

 valdecir.me

 [@homelaber](https://twitter.com/@homelaber)

 vbrownbagbrasil.com.br

 [/valdecircarvalho/](https://github.com/valdecircarvalho/)

Disclaimer:

*As opiniões e informações
contidas nessa apresentação
são pessoais e não representam
o meu empregador.*

From Zero to Useful - Terraform para Sysadmins

**Como começar a usar o Terraform
agora para automatizar seu dia a dia**

Você pode fazer o download da apresentação & demos no link abaixo:

<http://bit.ly/terraform-vugsp>

“
Ao final dessa apresentação, você
irá conseguir começar a usar o
Terraform e se aprofundar nos
estudos da ferramenta!

AGENDA

- INFRAESTRUTURA COMO CÓDIGO
- CONHECENDO O TERRAFORM
- INSTALANDO & CONFIGURANDO
- PRINCIPAIS COMANDOS
- COLOCANDO A MÃO NA MASSA (AWS & vSphere)
- DEMO
- PRÓXIMO PASSOS

INFRAESTRUTURA COMO CÓDIGO (IaC)



CALLS
REMS:

- All calls for tracking
- 20% worse than

INFRAESTRUTURA COMO CÓDIGO (IaC)

Escrever e executar um código para definir, provisionar, atualizar (e destruir) uma infraestrutura

INFRAESTRUTURA COMO CÓDIGO (IaC)

- **Servidores**
- **Networks**
- **Firewalls**
- **LoadBalancers**
- **Storage**
- **IPs**
- **DNS**
- **ETC**

O QUE É TERRAFORM?

TERRAFORM

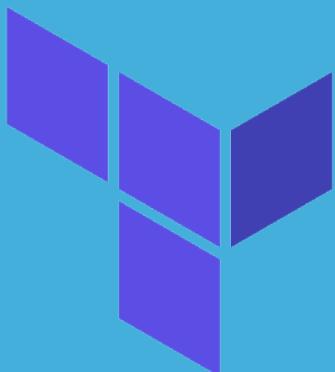
A palavra **terraform** significa modelar um planeta para que fique similar ao ambiente do planeta Terra.



TERRAFORM

TERRAFORM

É uma ferramenta para **CRIAR, ALTERAR, VERSIONAR**
e **DOCUMENTAR** uma infraestrutura de uma maneira
SEGURA e **EFICIENTE**.



HashiCorp
Terraform

HashiCorp Suite

ANY APPLICATION

RUN

SECURE

PROVISION

ANY INFRASTRUCTURE

OPEN SOURCE



HashiCorp
Nomad



HashiCorp
Vault



HashiCorp
Terraform



HashiCorp
Packer



HashiCorp
Vagrant



HashiCorp
Consul



HashiCorp

TERRAFORM

- Desenvolvida pela HashiCorp – hashicorp.org
- Free e Open Source, mas existe uma versão Ent.
- Escrita em GO
- Multi Cloud + 160 providers
- Uma maneira de gerenciar recursos na AWS
- Fácil de escrever e ler (HCL)
- Roda em Windows, Mac e Linux

TERRAFORM

- + 16.500 stars no Github
- Atualmente na versão 0.11.13 (Mar/2019)
- Versão 0.1.0 Julho/2014 – 6 anos



Google Cloud Platform

vmware®

ORACLE®
CLOUD PLATFORM



IBM Cloud

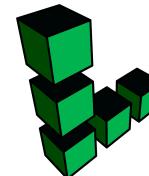


Alibaba Cloud

DigitalOcean



openstack®



linode



HEROKU



OVH

ACME

Alicloud

Archive

Arukas

AWS

Azure

Azure Active Directory

Azure Stack

Bitbucket

Brightbox

CenturyLinkCloud

Chef

Circonus

Cisco ASA

Cloudflare

CloudScale.ch

CloudStack

Cobbler

Consul

Datadog

DigitalOcean

DNS

DNSimple

DNSMadeEasy

Docker

Dyn

External

F5 BIG-IP

Fastly

FlexibleEngine

Fortios

GitHub

Gitlab

Google Cloud Platform

Grafana

Hedvig

Helm

Heroku

Hetzner Cloud

HTTP

HuaweiCloud

Icinga2

Ignition

InfluxDB

JDCloud

Kubernetes

Librato

Linode

Local

Logentries

LogicMonitor

Mailgun

MySQL

Naver Cloud

Netlify

New Relic

Nomad

NS1

Null

Nutanix

1&1

OpenStack

OpenTelekomCloud

OpsGenie

Oracle Cloud Infrastructure

Oracle Cloud Platform

Oracle Public Cloud

OVH

Packet

PagerDuty

Palo Alto Networks

PostgreSQL

PowerDNS

ProfitBricks

RabbitMQ

Rancher

Random

RightScale

Rundeck

RunScope

Scaleway

Selectel

Skytap

SoftLayer

Spotinst

StatusCake

TelefonicaOpenCloud

Template

TencentCloud

Terraform

Terraform Enterprise

TLS

Triton

UCloud

UltraDNS

Vault

VMware NSX-T

VMware vCloud Director

VMware vSphere

Yandex

ALTERNATIVAS AO TERRAFORM



CHEF



ANSIBLE



puppet



SALTSTACK

■ QUANDO USAR TERRAFORM

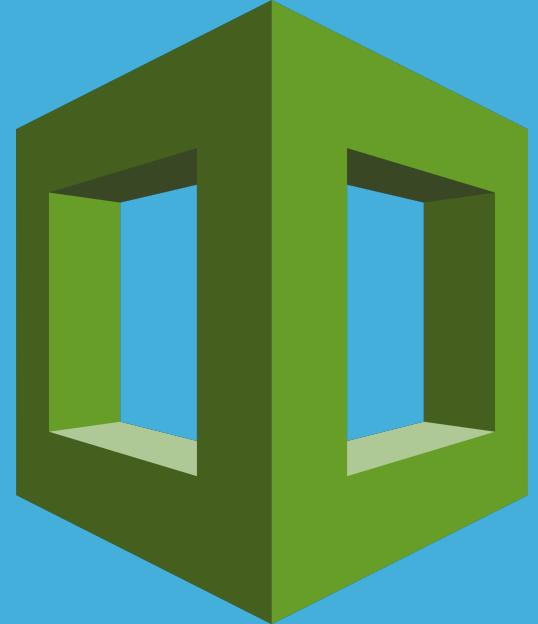
- CRIAR, MODIFICAR e DESTRUIR a sua INFRAESTRUTURA

■ QUANDO NÃO USAR TERRAFORM

- Configurações de SO
- Configurações de Sistema
- Puppet, Chef, Ansible são ferramentas focadas em CONFIGURAÇÃO.
- Devem ser utilizadas em conjunto

AWS CLOUDFORMATION

- Ferramenta nativa da AWS
- Desenvolvida e mantida pela AWS
- Funciona somente para AWS
- Usa JSON como linguagem
- Oferecido pela AWS for FREE



INSTALANDO O TERRAFORM



3 PASSOS

<https://www.terraform.io/downloads.html>

1

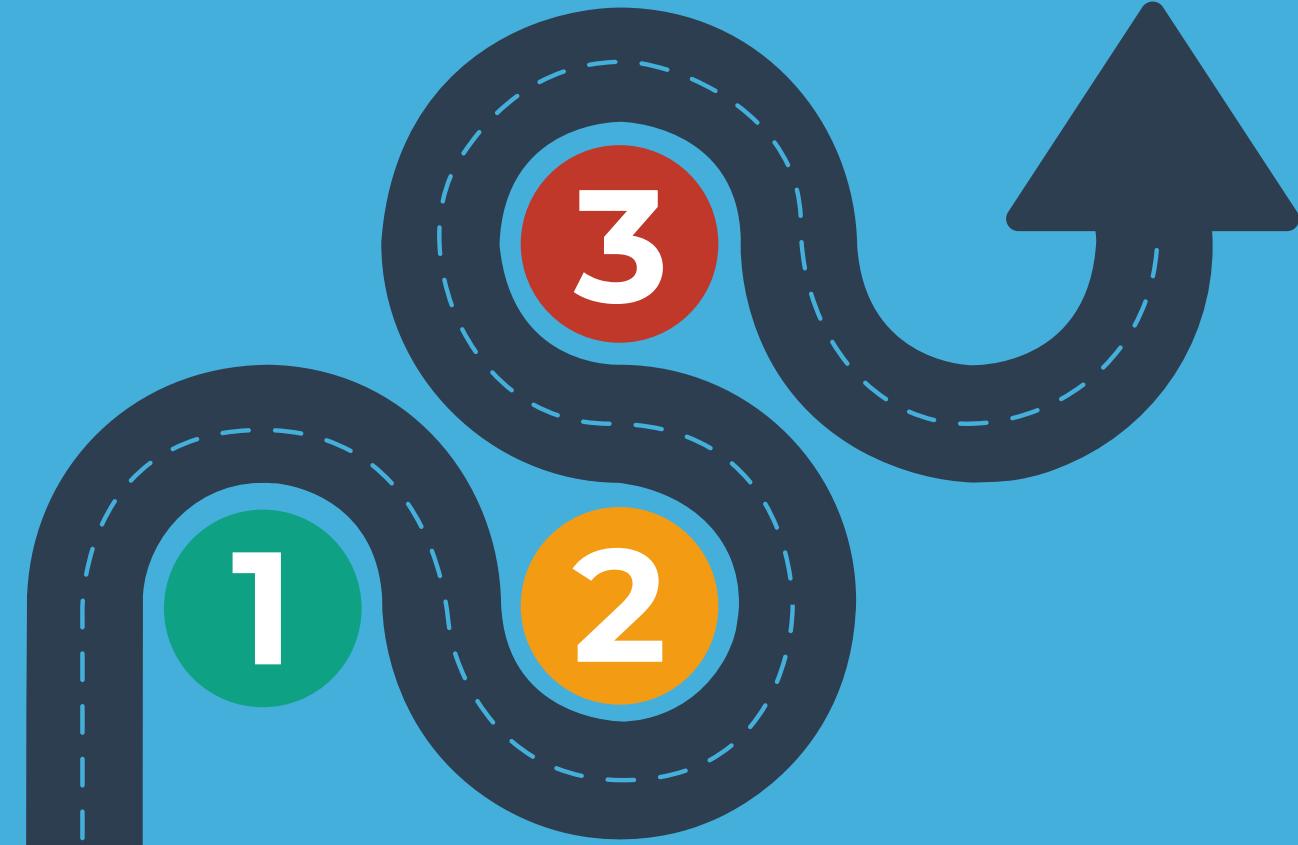
FAÇA O DOWNLOAD
DO BINÁRIO

2

EXECUTE O BINÁRIO

3

NÃO EXISTE
TERCEIRO PASSO ☺



AWS CLI



INSTALANDO AWS CLI

- Pré-Requisito para usar o Terraform com AWS
- Ferramenta de linha de comando para gerenciamento AWS
- Linux, Mac e Windows

https://docs.aws.amazon.com/pt_br/cli/latest/userguide/cli-chap-install.html

CONFIGURANDO AWS CLI

- Usuário IAM
- AWS Access Key ID
- AWS Secret Access Key

https://docs.aws.amazon.com/pt_br/cli/latest/userguide/cli-chap-configure.html#cli-quick-configuration

```
$ aws configure
```

AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE

AWS Secret Access Key [None]:wJalrXUtnFEMI/K7MDENG/bPxRfICYEXAMPLEKEY

Default region name [None]: us-west-2

Default output format [None]: json

PRINCIPAIS COMANDOS TERRAFORM

```
$ terraform
```

init
plan
apply
destroy
refresh
help

```
$ terraform -h
```

```
$ terraform -h  
Usage: terraform [-version] [-help] <command> [args]
```

The available commands for execution are listed below.
The most common, useful commands are shown first, followed by less common or more advanced commands. If you're just getting started with Terraform, stick with the common commands. For the other commands, please read the help and docs before usage.

Common commands:

apply	Builds or changes infrastructure
console	Interactive console for Terraform interpolations
destroy	Destroy Terraform-managed infrastructure
env	Workspace management
fmt	Rewrites config files to canonical format
get	Download and install modules for the configuration
graph	Create a visual graph of Terraform resources
import	Import existing infrastructure into Terraform
init	Initialize a Terraform working directory
output	Read an output from a state file
plan	Generate and show an execution plan
providers	Prints a tree of the providers used in the configuration
push	Upload this Terraform module to Atlas to run
refresh	Update local state file against real resources
show	Inspect Terraform state or plan
taint	Manually mark a resource for recreation
untaint	Manually unmark a resource as tainted
validate	Validates the Terraform files
version	Prints the Terraform version
workspace	Workspace management

All other commands:

debug	Debug output management (experimental)
force-unlock	Manually unlock the terraform state
state	Advanced state management

```
$ terraform -h
```

```
$ terraform -h
```

```
Usage: terraform [-version] [-help] <command> [args]
```

The available commands for execution are listed below.

The most common, useful commands are shown first, followed by less common or more advanced commands. If you're just getting started with Terraform, stick with the common commands. For the other commands, please read the help and docs before usage.

Common commands:

apply	Builds or changes infrastructure
console	Interactive console for Terraform interpolations
destroy	Destroy Terraform-managed infrastructure
env	Workspace management
fmt	Rewrites config files to canonical format
get	Download and install modules for the configuration
graph	Create a visual graph of Terraform resources
import	Import existing infrastructure into Terraform
init	Initialize a Terraform working directory
output	Read an output from a state file
plan	Generate and show an execution plan
providers	Prints a tree of the providers used in the configuration
push	Upload this Terraform module to Atlas to run
refresh	Update local state file against real resources
show	Inspect Terraform state or plan
taint	Manually mark a resource for recreation
untaint	Manually unmark a resource as tainted
validate	Validates the Terraform files
version	Prints the Terraform version
workspace	Workspace management

All other commands:

debug	Debug output management (experimental)
force-unlock	Manually unlock the terraform state
state	Advanced state management

TERRAFORM WORKFLOW



TERAFORM WORKFLOW



ARQUIVOS TERRAFORM



main.tf

variables.tfvars

outputs.tf

bozo.tf

xorумelas.tfvars

supermario.tf

ARQUIVOS TERRAFORM

- Escritos em HCL – Hashicorp Configuration Language
- .tf – arquivo de instruções do terraform
- terraform.tfvars – arquivos de variáveis
- .tfstate - arquivos com o “estado” da sua infra
- Dividido em blocos de recursos

<https://www.terraform.io/docs/configuration/index.html>

<https://github.com/hashicorp/hcl2/blob/master/hcl/hclsyntax/spec.md>

COLOCANDO A MÃO NA MASSA

```
provider "aws" {  
  DEFINIR O PROVIDER (main.tf)  
  access_key = "chave_de_acesso_aws"  
  secret_key = "chave_secreta_aws"  
  region     = "região_aws"  
}
```

```
resource "aws_instance" "web" {
  DEFINITION RESOURCE (main.tf)
  ami           = "ami-0a313d6098716f372" #ubuntu
  instance_type = "t2.micro"

  tags {
    Name = "web-demo-aws-ug-sp"
  }
}
```

DEFINIR O RESOURCE (main.tf)

```
resource "aws_instance" "web" {
    ami = "ami-0a313d6098716f372" #ubuntu
    instance_type = "t2.micro"

    tags {
        Name = "web-demo-aws-ug-sp"
    }
}
```

DEFINIR O RESOURCE (main.tf)

```
resource "aws_instance" "web" {
    ami = "ami-0a313d6098716f372" #ubuntu
    instance_type = "t2.micro"

    tags {
        Name = "web-demo-aws-ug-sp"
    }
}
```

DEFINIR O RESOURCE (main.tf)

```
resource "aws_instance" "web" {
    ami = "ami-0a313d6098716f372" #ubuntu
    instance_type = "t2.micro"

    tags {
        Name = "web-demo-aws-ug-sp"
    }
}
```

DEFINIR O RESOURCE (main.tf)

```
resource "aws_instance" "web" {
    ami = "ami-0a313d6098716f372" #ubuntu
    instance_type = "t2.micro"

    tags {
        Name = "web-demo-aws-ug-sp"
    }
}
```

DEFINIR O RESOURCE (main.tf)

```
resource "aws_instance" "web" {
    ami = "ami-0a313d6098716f372" #ubuntu
    instance_type = "t2.micro"

    tags {
        Name = "web-demo-aws-ug-sp"
    }
}
```

```
provider "aws" {
  access_key = "AKIA3FAKEKEYHUEBRDUL"
  secret_key = "fAkEke3yHhTlghuehUEBrh7Ryvp+00J2S/awSuu0d"
  region = "região_aws"
}

resource "aws_instance" "web" {
  ami = "ami-0a313d6098716f372" #ubuntu
  instance_type = "t2.micro"

  tags {
    Name = "web-demo-aws-ug-sp"
  }
}
```

```
$ terraform init
```

Initializing provider plugins...
\$terraform init

- Checking for available provider plugins on <https://releases.hashicorp.com>...
- Downloading plugin for provider "aws" (2.6.0)...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add `version = "..."` constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: `version = "~> 2.6"`

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "`terraform plan`" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

```
$ terraform plan
```

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

```
+ aws_instance.web
  id: <computed>
  ami: "ami-0a313d6098716f372"
  associate_public_ip_address: <computed>
  instance_type: "t2.micro"
  subnet_id: <computed>
  tags.%: "1"
  tags.name: "web-demo-aws-ug-sp"
```

Plan: 1 to add, 0 to change, 0 to destroy.

```
$ terraform apply tfplan

aws_instance.web: Creating
  ami:          "" => "ami-0a313d6098716f372"
  instance_type:      "" => "t2.micro"
  tags.%:           "" => "1"
  tags.name:        "" => "demo-aws-ug-sp"
  tenancy:          "" => "<computed>"
  volume_tags.%:    "" => "<computed>"
  vpc_security_group_ids.#: "" => "<computed>"

aws_instance.web: Still creating... (10s elapsed)
aws_instance.web: Still creating... (20s elapsed)
aws_instance.web: Still creating... (30s elapsed)
aws_instance.web: Still creating... (40s elapsed)
aws_instance.web: Creation complete after 42s (ID: i-0cf574f78aa2b7224)
```

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

AWS Services Resource Groups N. Virginia Support

EC2 Dashboard Events Tags Reports Limits

INSTANCES

Instances Launch Templates Spot Requests Reserved Instances Dedicated Hosts Scheduled Instances Capacity Reservations

IMAGES

AMIs Bundle Tasks

ELASTIC BLOCK STORE

Volumes Snapshots Lifecycle Manager

NETWORK & SECURITY

Security Groups Elastic IPs Placement Groups Key Pairs Network Interfaces

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

1 to 5 of 5

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring
web-demo-a...	i-00c07e9cf04e95903	t2.micro	us-east-1c	terminated	None	None	-	-	-	disabled	
	i-030e356db47d23a...	t2.micro	us-east-1a	terminated	None	None	-	-	-	disabled	
	i-09e618a450a462dc7	t2.micro	us-east-1c	terminated	None	None	-	-	-	disabled	
web-demo-a...	i-0d0e8ebdf1a2fc79b	t2.micro	us-east-1a	running	Initializing	None	ec2-3-87-189-111.compute-1.amazonaws.com	3.87.189.111	-	disabled	
	i-09e2a4d0ddb5636d	t2.micro	us-east-1a	terminated	None	None	-	-	-	disabled	

Instance: i-0d0e8ebdf1a2fc79b (web-demo-aws-ug-sp) Public DNS: ec2-3-87-189-111.compute-1.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID	i-0d0e8ebdf1a2fc79b	Public DNS (IPv4)	ec2-3-87-189-111.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	3.87.189.111
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-72-46.ec2.internal
Availability zone	us-east-1a	Private IPs	172.31.72.46
Security groups	default, view inbound rules, view outbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-fec57a87
AMI ID	ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20190212.1 (ami-0a313d6098716f372)	Subnet ID	subnet-3a982716
Platform	-	Network interfaces	eth0

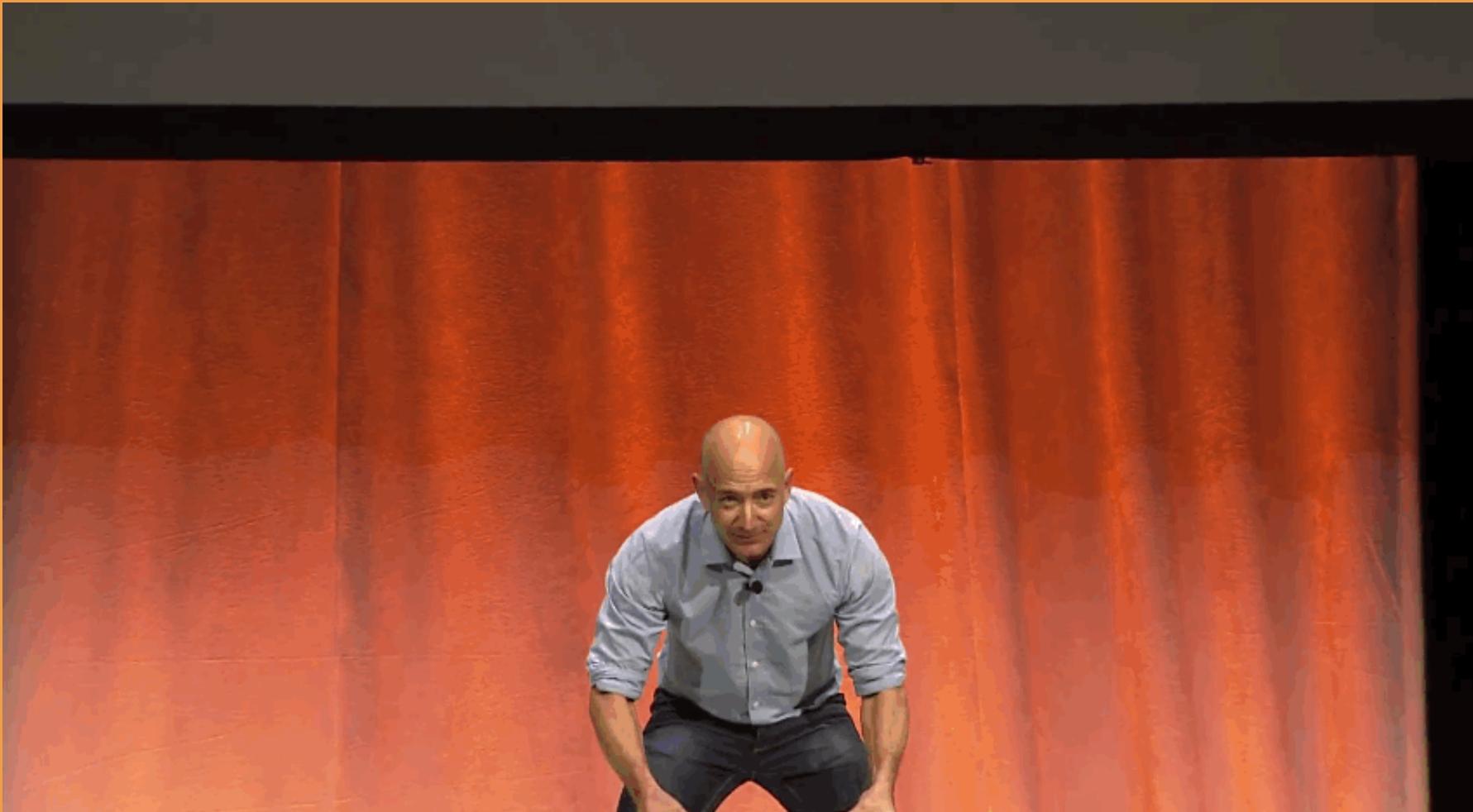
Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

AGORA VOCÊS
PODEM FAZER
OHHHHHHHH!



DEMO

**Provisionar uma instância EC2 simples
na AWS**



É assim que me sinto sempre que vou fazer uma demo ao vivo!



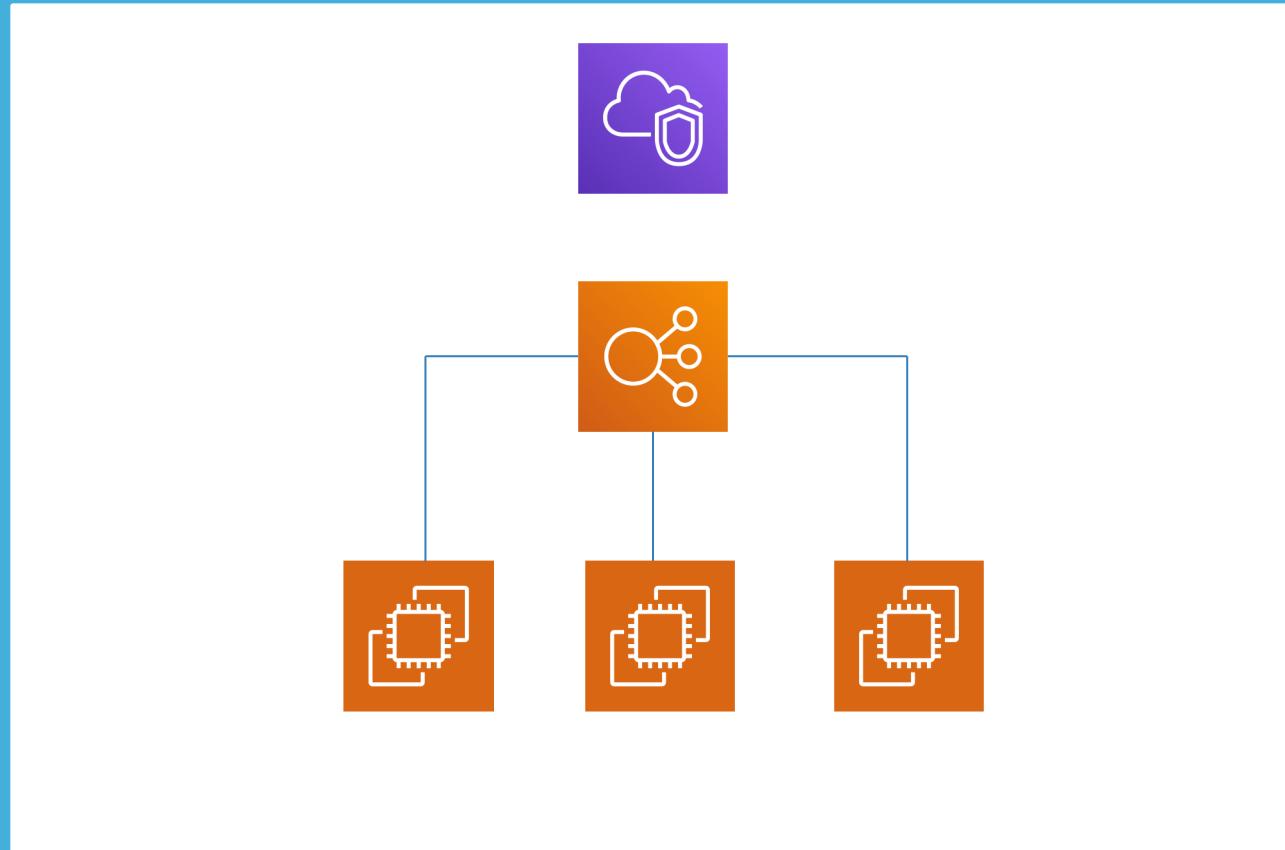
AGORA VOCÊS
PODEM FAZER
OHHHHHHHH
E APLAUDIR!



DEMO

**Provisionar uma infraestrutura
completa na AWS**

- VPC
- Subnet
- Internet Gateway
- Route Table
- Security Group
- Load Balance
- Instâncias





Essa é a plateia torcendo para o demo dar certo!



DEMO



Provisionar VMs no vSphere

```
provider "vsphere" {
  main.tf - VMware vSphere
  user = "${var.vsphere_user}"
  password = "${var.vsphere_password}"
  vsphere_server = "${var.vsphere_server}"

  # If you have a self-signed cert
  allow_unverified_ssl = true
}
```



**É PARA
APLAUDIR
DE PÉ!**



BONUS



TERRAFORMING

TERRAFORMING

- **Exporta** sua infraestrutura **existente na AWS no formato Terraform** (.tf, tfstate).
- Free e Opensource
- Criado por Daisuke Fujita (JP) -  /dtan4
- <http://terraform.dtan4.net/>

TERRAFORMING != CHEATING
TERRAFORMING = LEARNING

\$ terraforming

```
$ terraforming
Commands:
terraforming alb          # ALB
terraforming asg          # AutoScaling Group
terraforming cwa          # CloudWatch Alarm
terraforming dbng          # Database Parameter Group
terraforming dbsg          # Database Security Group
terraforming dbsh          # Database Subnet Group
terraforming ec2          # EC2
terraforming ecc          # ElasticCache Cluster
terraforming ecdn          # ElasticCache Subnet Group
terraforming efs          # EFS File System
terraforming eip          # EIP
terraforming elb          # ELB
terraforming help [COMMAND] # Describe available commands or one specific command
terraforming iamg          # IAM Group
terraforming iamgm         # IAM Group Membership
terraforming iamgp         # IAM Group Policy
terraforming iamip         # IAM Instance Profile
terraforming iamp          # IAM Policy
terraforming iampa         # IAM Policy Attachment
terraforming iamr          # IAM Role
terraforming iamrp         # IAM Role Policy
terraforming iamu          # IAM User
terraforming iamup         # IAM User Policy
terraforming igw           # Internet Gateway
terraforming kmsa          # KMS Key Alias
terraforming kmsk          # KMS Key
terraforming lc            # Launch Configuration
terraforming nacl          # Network ACL
terraforming nat           # NAT Gateway
terraforming nif            # Network Interface
terraforming r53r          # Route53 Record
terraforming r53z          # Route53 Hosted Zone
terraforming rds           # RDS
terraforming rs             # Redshift
terraforming rt            # Route Table
terraforming rta           # Route Table Association
terraforming s3             # S3
terraforming sg             # Security Group
terraforming sn             # Subnet
terraforming sns            # SNS Subscription
terraforming snst           # SNS Topic
terraforming sqs            # SQS
terraforming vgw           # VPN Gateway
terraforming vpc           # VPC
```

Options:

```
[--merge=MERGE]          # tfstate file to merge
[--overwrite], [--no-overwrite] # Overwrite existing tfstate
[--tfstate], [--no-tfstate]   # Generate tfstate
[--profile=PROFILE]         # AWS credentials profile
[--region=REGION]          # AWS region
[--assume=ASSUME]           # Role ARN to assume
[--use-bundled-cert], [--no-use-bundled-cert] # Use the bundled CA certificate from AWS SDK
```

AGORA É COM VOCÊ

**DICAS & RECURSOS PARA VOCÊ
SEGUIR EM FREnte COM TERRAFORM**

veeAM



HashiCorp
Terraform

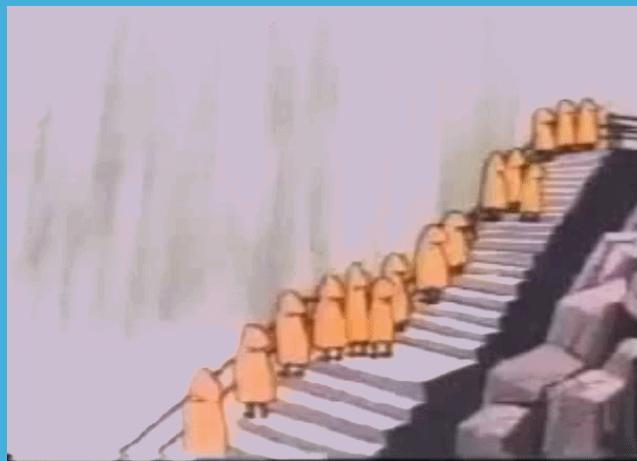
https://github.com/exospheredata/veeam_terraform

PRÓXIMOS PASSOS

- Crie uma conta free na AWS se ainda não tiver uma.
- Instale o terraform no seu computador
- <https://learn.hashicorp.com/terraform/> - Get Started Terraform
- Leia a documentação do Terraform –
- Comece devagar... baby steps e vá evoluindo
- Procure exemplos no Github (terraform aws)
- Comece um projeto de IaC na sua empresa (prática)
- User o terraforming em uma estrutura já criada para aprender como montar o seu .tf file

“

SE JOGA! VOCÊ SÓ FICA MELHOR FAZENDO!



JABÁ



#vBrownBag



Podcast vBrownBag Brasil
vbrownbagbrasil.com.br

Toda segunda-feira às 21:00hs
youtube.com/vbrownbagbrasil



Homelaber Brasil 
Homelab, Virtualização & infraestrutura de TI

Blog Homelaber Brasil
homelaber.com.br



**FOLLOW
@TARSIO**



Download da apresentação & demos:

<http://bit.ly/terraform-vugsp>

OBRIGADO



Valdecir Carvalho

+55 11 95259-3615 (📱 mobile)

<http://valdecir.me> | @homelaber

❤️ Community Lover | Blogger| Podcaster | vExpert | Veeam Vanguard