

Infra as a Code + Go = <3

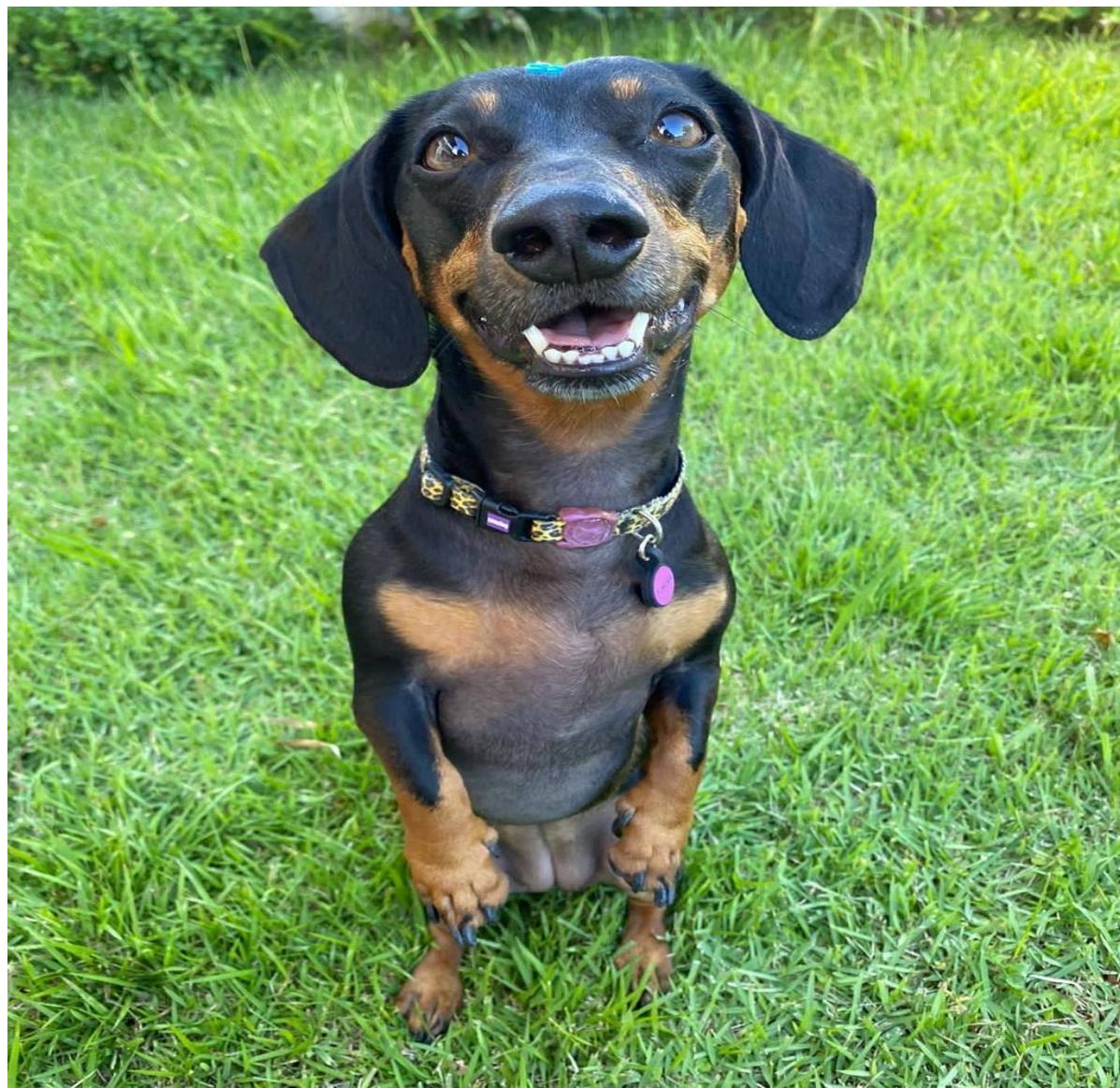
22 Set 2023

Pery Lemke

\$whoami

- Pery Lemke mas pode me chamar de Pery mesmo!
- Já fui de tudo um pouco, (SysAdmin, SRE, Platform Engineer) mas agora estou em transição para Dev :)
- Baterista bem mais ou menos
- N00b de Jiu Jitsu
- Colecionador de camisas de futebol
- Fascinado por computação
- Pai da Aurora ;)

Aurora <3



DevOpsDays Florianópolis - GOPHERCON10



O que é Infra as Code?

De maneira simples, é uma abordagem que permite automatizar a criação, configuração e gerenciamento de uma infraestrutura usando código.

5

Algumas ferramentas

- CFEngine
- Puppet
- Chef
- Ansible
- [OpenTofu](https://spacelift.io/blog/what-is-opentofu) (<https://spacelift.io/blog/what-is-opentofu>)
- Terraform (HCL)
- Terraform CDK
- Pulumi

Por que adotar uma linguagem de programação para IaC?

- Flexibilidade
- Curva de aprendizado
- Escolha da equipe

Por que não adotar uma linguagem de programação para IaC?

- Flexibilidade
- Curva de aprendizado
- Escolha da equipe

8

Terraform CDK

Possibilidade de criar infraestrutura com uma linguagem de programação com os providers do Terraform.

Linguagens suportadas:

- TypeScript
- Python
- Java
- C#
- Go

A partir deste momento irei chamar de cdktf para não confundir ;)

9

Pulumi

Ferramenta open source que utiliza as mesmas linguagens que o cdktf, mas pode utilizar linguagens de marcação como yaml e CUE.

Você pode conferir as diferenças entre Terraform e Pulumi [aqui](https://www.pulumi.com/docs/concepts/vs/terraform/#differences) (<https://www.pulumi.com/docs/concepts/vs/terraform/#differences>) .

10

Estruturas de pastas

Terraform

```
modules
  vpc
    main.go
    outputs.go
    variables.go
    README.md
  s3
    main.go
    outputs.go
    variables.go
    README.md
  ...Another modules
env
  production.tfvars
  dev.tfvars
  staging.tfvars
tests
  vpc_test.go
  s3_test.go
  README.md
main.go
outputs.go
variables.go
README.md
```

cdktf

```
$ cdktf init --template="go" --providers="aws@~>4.0" --project-name="gophercon-iac-cdk" --local
```

```
|__ gophercon-iac-cdk
|  __ .gitignore
|  __ cdktf.json
|  __ go.mod
|  __ go.sum
|  __ help
|  __ main_test.go
|  __ main.go
```

Pulumi

```
$ pulumi new aws-go --name gophercon-iac-pulumi
```

```
|__ gophercon-iac-pulumi
|  |__ go.mod
|  |__ go.sum
|  |__ pulumi.dev.yaml
|  |__ pulumi.yaml
|  |__ main.go
```

14

Arquivo de configuração

Terraform

provider.tf

```
terraform {  
    required_providers {  
        aws = {  
            source  = "hashicorp/aws"  
            version = "5.5.0"  
        }  
    }  
}  
  
# Configure the AWS Provider  
provider "aws" {  
    default_tags {  
        tags = {  
            environment = terraform.workspace  
            owner       = "pery"  
            created_by  = "pery"  
        }  
    }  
}
```

cdktf

cdktf.json

```
{  
  "language": "go",  
  "app": "go run main.go",  
  "codeMakerOutput": "generated",  
  "projectId": "330b3036-f12c-491e-ab5b-cc7d610d16c6",  
  "sendCrashReports": "true",  
  "terraformProviders": [  
    "hashicorp/aws@~>4.0"  
,  
  "terraformModules": [],  
  "context": {}  
}
```

Pulumi

Pulumi.yaml

```
name: gophercon2-iac-pulumi
runtime: go
description: A minimal AWS Go Pulumi program
```

Pulumi.dev.yaml

```
config:
  aws:region: us-east-1
```

Armazenamento de estados

Terraform

backend.tf

```
terraform {  
    required_version = ">= 0.12"  
  
    backend "s3" {  
        bucket = "my-infra"  
        key    = "states/envs"  
        region = "us-east-1"  
    }  
}
```

cdktf

```
package main

import (
    "github.com/aws/constructs-go/constructs/v10"
    "github.com/aws/jsii-runtime-go"
    "github.com/hashicorp/terraform-cdk-go/cdktf"
)

func NewCloudBackendStack(scope constructs.Construct, name string) cdktf.TerraformStack {
    stack := cdktf.NewTerraformStack(scope, &name)

    cdktf.NewCloudBackend(stack, &cdktf.CloudBackendConfig{
        Hostname: jsii.String("app.terraform.io"),
        Organization: jsii.String("company"),
        Workspaces: cdktf.NewNamedCloudWorkspace(jsii.String("my-app-prod")),
    })

    cdktf.NewTerraformOutput(stack, jsii.String("dns-server"), &cdktf.TerraformOutputConfig{
        Value: "hello-world",
    })

    return stack
}
```

Pulumi

- Pulumi Cloud:

```
$ pulumi login
```

- S3:

```
$ pulumi login s3://<bucket-name>
```

- Azure Blob Storage:

```
$ pulumi login azblob://<container-path>
```

- GCS

```
$ pulumi login gs://<my-pulumi-state-bucket>
```

- Local

```
$ pulumi login --local
```

Código

Terraform

Criando um bucket privado:

```
resource "aws_s3_bucket" "test" {
  bucket = "my-bucket-test"

  tags = {
    Name      = "My test bucket"
    Environment = "Dev"
  }
}
```

24

cdktf

```
package main

import (
    "github.com/hashicorp/terraform-cdk-go/tfcdk"
    "github.com/zclconf/go-cty/cty"
)

func NewS3Bucket(stack tfcdk.TfcdkStack, name string) {
    stack.Resource(name, "aws_s3_bucket", "my_test_bucket",
        &map[string]cty.Value{
            "bucket": cty.StringVal("my-bucket-test"),
        })
}

func main() {
    app := tfcdk.NewApp(nil)

    stack := tfcdk.NewStack(app, "MyStack")

    NewS3Bucket(stack, "MyS3Bucket")

    app.Synth(nil)
}
```

Pulumi

```
package main

import (
    "github.com/pulumi/pulumi-aws/sdk/v6/go/aws/s3"
    "github.com/pulumi/pulumi/sdk/v3/go/pulumi"
)

func main() {
    pulumi.Run(func(ctx *pulumi.Context) error {
        // Create an AWS resource (S3 Bucket)
        bucket, err := s3.NewBucket(ctx, "my-bucket-test", nil)
        if err != nil {
            return err
        }

        // Export the name of the bucket
        ctx.Export("bucketName", bucket.ID())
        return nil
    })
}
```

Testes

Terraform

No Terraform geralmente utilizamos o terratest que usa **Go** como linguagem para os testes.

```
func TestTerraformAwsS3Example(t *testing.T) {
    t.Parallel()

    expectedName := fmt.Sprintf("my-bucket-test-%s", strings.ToLower(random.UniqueId()))
    expectedEnvironment := "dev"

    awsRegion := aws.GetRandomStableRegion(t, nil, nil)

    terraformOptions := terraform.WithDefaultRetryableErrors(t, &terraform.Options{
        TerraformDir: "../",
        Vars: map[string]interface{}{
            "tag_bucket_name":           expectedName,
            "tag_bucket_environment":   expectedEnvironment,
            "with_policy":              "true",
            "region":                   awsRegion,
        },
    })
    defer terraform.Destroy(t, terraformOptions)

    terraform.InitAndApply(t, terraformOptions)
```

```
bucketID := terraform.Output(t, terraformOptions, "bucket_id")

actualStatus := aws.GetS3BucketVersioning(t, awsRegion, bucketID)
expectedStatus := "Enabled"
assert.Equal(t, expectedStatus, actualStatus)

}
```

cdktf

```
package main

import (
    "testing"

    "cdk.tf/go/stack/generated/kreuzwerker/docker/container"
    "cdk.tf/go/stack/generated/kreuzwerker/docker/image"
    "github.com/hashicorp/terraform-cdk-go/cdktf"
)

var stack = NewMyApplicationsAbstraction(cdktf.Testing_App(nil), "stack")
var synth = cdktf.Testing_Synth(stack)

func TestShouldContainContainer(t *testing.T) {
    assertion := cdktf.Testing_ToHaveResource(synth, container.Container_TfResourceType())

    if !*assertion {
        t.Error("Assertion Failed")
    }
}
```

Pulumi

```
package main

import (
    "testing"

    "github.com/pulumi/pulumi/sdk/v4/go/pulumi"
    "github.com/stretchr/testify/assert"
)

func TestCreateS3Bucket(t *testing.T) {
    pulumi.Run(func(ctx *pulumi.Context) error {
        bucket, err := createS3Bucket(ctx, "my-test-bucket")
        if err != nil {
            t.Fatalf("Error on created the S3 bucket: %v", err)
        }

        assert.NotNil(t, bucket)
        assert.Equal(t, "my-test-bucket", bucket.ID())

        return nil
    })
}
```

Conclusão

Bom, essa talk serve como uma introdução da introdução sobre o tema.

Minha opinião pessoal: Se for adotar alguma ferramenta de IaC com Go, adote o **Pulumi**, o Terraform CDK funciona melhor em outras linguagens.

Mas não façam nada na pressa, testem, façam POCs e aplique somente se for algo de fácil manutenção para você e sua equipe.

31

Links Úteis

- [Repo da Talk](https://github.com/perylemke/talks/tree/main/go-iac)
- [Docs do Pulumi](https://www.pulumi.com/docs/)
- [Docs do cdktf](https://developer.hashicorp.com/terraform/cdktf)
- [CDKTF — Terraform's Go CDK](https://medium.com/geekculture/cdktf-terraforms-go-cdk-c7be1a216d67)
- [Deploy Infrastructure using CDK for Terraform with Go](https://michaellin.me/deploy-infrastructure-using-cdktf-with-go/)
- [Build AWS infrastructure with CDK for Terraform](https://developer.hashicorp.com/terraform/tutorials/cdktf/cdktf-build)
- [Should you use Terraform or Pulumi?](https://mkdev.me/posts/should-you-use-terraform-or-pulumi)

Thank you

Pery Lemke

pery.lemke@gmail.com (<mailto:pery.lemke@gmail.com>)

[@perylemke](https://twitter.com/perylemke) ([http://twitter.com/perylemke](https://twitter.com/perylemke))