Bom dia! Vamos por partes — esse código é um exemplo de configuração básica em Terraform usando o provedor da AWS. Vou te explicar **bloco por bloco** o que cada parte faz:

**1. Bloco terraform**

terraform {

required\_providers {

aws = {

source = "hashicorp/aws"

version = "5.37.0"

}

}

}

Esse bloco define o **provedor** necessário para o projeto Terraform funcionar.

* required\_providers: lista os provedores que o Terraform vai usar.
* aws: é o nome do provedor.
* source = "hashicorp/aws": especifica o repositório do provedor.
* version = "5.37.0": trava a versão do provedor para evitar problemas de compatibilidade.

**2. Bloco resource "aws\_s3\_bucket" "my\_bucket"**

resource "aws\_s3\_bucket" "my\_bucket" {

bucket = "my-sample-bucket"

}

Esse bloco cria um recurso **gerenciado pelo Terraform**, no caso, um bucket S3 chamado my-sample-bucket.

* aws\_s3\_bucket: tipo do recurso (bucket S3).
* "my\_bucket": nome interno (apelido) do recurso no Terraform.
* bucket: nome real do bucket que será criado na AWS.

**3. Bloco data "aws\_s3\_bucket" "my\_external\_bucket"**

data "aws\_s3\_bucket" "my\_external\_bucket" {

bucket = "not-managed-by-us"

}

Esse bloco usa um **data source** para referenciar um bucket que **já existe** na AWS, mas **não é gerenciado** por este código Terraform.

* data: indica que é apenas leitura (não será criado).
* "aws\_s3\_bucket": tipo do recurso que está sendo consultado.
* "my\_external\_bucket": apelido interno.
* bucket = "not-managed-by-us": nome real do bucket existente.

**4. Bloco variable "bucket\_name"**

variable "bucket\_name" {

type = string

description = "My variable used to set bucket name"

default = "my\_default\_bucket\_name"

}

Define uma **variável** que pode ser usada em qualquer lugar do código.

* type = string: o tipo da variável.
* description: descrição da variável.
* default: valor padrão caso o usuário não passe outro valor.

**5. Bloco output "bucket\_id"**

output "bucket\_id" {

value = aws\_s3\_bucket.my\_bucket.id

}

Esse bloco **exporta um valor** ao final da execução do terraform apply.

* "bucket\_id": nome da saída.
* value = aws\_s3\_bucket.my\_bucket.id: retorna o ID do bucket criado no recurso my\_bucket.

**6. Bloco locals**

locals {

local\_example = "This is a local variable"

}

Define uma **variável local**, útil para armazenar valores reutilizáveis no código.

* local\_example: nome da variável local.
* "This is a local variable": valor.

Essas variáveis não são visíveis externamente, apenas dentro do próprio Terraform.

**7. Bloco module "my\_module"**

module "my\_module" {

source = "./module-example"

}

Esse bloco **chama um módulo externo**. Módulos são como **blocos reutilizáveis** de Terraform.

* module "my\_module": nome do módulo.
* source = "./module-example": caminho local de onde o módulo está localizado.

**Resumo:**

| **Bloco** | **Função** |
| --- | --- |
| terraform | Define provedor da AWS e versão |
| resource | Cria um bucket S3 |
| data | Consulta um bucket S3 já existente |
| variable | Cria variável reutilizável |
| output | Exibe o ID do bucket criado |
| locals | Define uma variável local interna |
| module | Usa um módulo Terraform reutilizável |

Se quiser, posso te ajudar a modificar esse código para criar buckets com nomes dinâmicos, aplicar políticas, versionamento, ou qualquer outra funcionalidade. Deseja fazer alguma melhoria nele?