Laboratório 17 – Validação

# Cenário

Nesse laboratório iremos ver como realizar validações.

# Objetivos:

Nesse laboratório, você irá:

* Atividade 01: Usar validação de entreda
* Atividade 02: Usar precondition
* Atividade 03: Usar postcondition

# Instruções

## Atividade 01: Usar validação de entrada

1. Crie um diretório chamado lab17 no diretório laboratórios.
2. Copie os arquivos do zip desse laboratório para dentro do diretório criado.
3. Abra o VS Code no diretório lab17.
4. Crie um arquivo chamado aws\_instance.tf e adicione o seguinte conteúdo.

resource "aws\_instance" "foo" {

ami = var.image\_id

instance\_type = "t2.micro"

associate\_public\_ip\_address = true

tags = {

"Name" = var.ec2\_name

"Environment" = "MyEnv"

}

}

1. Crie um arquivo chamado variables.tf e adicione o seguinte conteúdo.

variable "image\_id" {

type = string

description = "The id of the machine image (AMI) to use for the server."

validation {

# regex(...) fails if it cannot find a match

condition = can(regex("^ami-", var.image\_id))

error\_message = "The image\_id value must be a valid AMI id, starting with \"ami-\"."

}

}

variable "ec2\_name" {

description = "Nome da instância do EC2"

type = string

}

variable "vpc\_name" {

description = "Nome da VPC"

type = string

}

variable "subnet\_name" {

description = "Nome da subnet"

type = string

}

variable "env" {

description = "Ambiente que será provisionado"

type = string

}

1. Verifique que a variável image\_id possui uma validação.
2. Acesse o arquivo provider.tf e adicione o access key e secret key do seu usuário.
3. Execute o comando terraform init.
4. Execute o comando terraform apply --auto-approve.
5. Depois de ser executado com sucesso, execute o comando terraform destroy --auto-approve.

## Atividade 02: Usar precondition

1. Acesse o arquivo aws\_instance.tf e adicione o seguinte conteúdo dentro do bloco que cria a instância.

lifecycle {

# The AMI ID must refer to an AMI that contains an operating system

# for the `x86\_64` architecture.

precondition {

condition = data.aws\_ami.example.architecture == "x86\_64"

error\_message = "The selected AMI must be for the x86\_64 architecture."

}

}

1. Verifique que o bloco lifecycle possui um bloco precondition com uma validação da arquitetura da imagem utilizada na criação da instância.
2. Execute o comando terraform apply --auto-approve.
3. Depois de ser executado com sucesso, execute o comando terraform destroy --auto-approve.

## Atividade 02: Usar postcondition

1. Acesse o arquivo aws\_instance.tf e adicione o seguinte conteúdo dentro do bloco lifecycle.

The EC2 instance must be allocated a public DNS hostname.

postcondition {

condition = self.public\_ip != ""

error\_message = "EC2 instance must be in a VPC that has public DNS hostnames enabled."

}

1. Verifique que o bloco lifecycle possui um bloco postcondition com uma validação do dns público da instância.
2. Execute o comando terraform apply --auto-approve.
3. Depois de ser executado com sucesso, execute o comando terraform destroy --auto-approve.