# Projeto VExpenses - Infraestrutura AWS com Terraform

## Descrição

Este repositório contém os arquivos necessários para criar uma infraestrutura básica na AWS utilizando o Terraform. A infraestrutura inclui:

- Uma instância EC2 rodando \*\*Debian 12\*\*.

- Configuração de rede utilizando \*\*VPC\*\*, \*\*Subnets\*\*, \*\*Internet Gateway\*\* e \*\*Route Tables\*\*.

- Configuração de segurança com \*\*Security Groups\*\*.

- Automatização da instalação do servidor \*\*Nginx\*\*.

\*Descrição Técnica\*

1. \*Provider AWS\*:

Define o provedor AWS e a região (us-east-1).

2. \*Variáveis (variable)\*:

- projeto: Nome do projeto (padrão: "VExpenses").

- candidato: Nome do candidato (padrão: "SeuNome").

3. \*Chave SSH\*:

- Gera uma chave privada (tls\_private\_key).

- Cria um par de chaves na AWS (aws\_key\_pair).

4. \*VPC e Sub-rede\*:

- Cria uma VPC (aws\_vpc) com o CIDR 10.0.0.0/16.

- Configura uma sub-rede (aws\_subnet) na zona de disponibilidade us-east-1a com o CIDR 10.0.1.0/24.

5. \*Gateway de Internet e Roteamento\*:

- Cria um Gateway de Internet (aws\_internet\_gateway).

- Define uma tabela de rotas (aws\_route\_table) para permitir o tráfego externo (0.0.0.0/0).

- Associa a tabela de rotas à sub-rede (aws\_route\_table\_association).

6. \*Grupo de Segurança\*:

- Permite o tráfego SSH (porta 22) de qualquer lugar (0.0.0.0/0).

- Libera todo o tráfego de saída.

7. \*Imagem da EC2 (aws\_ami)\*:

- Busca a AMI mais recente do Debian 12 na AWS.

8. \*Instância EC2\*:

- Cria uma instância t2.micro com a AMI Debian 12.

- Usa a chave SSH criada e o grupo de segurança configurado.

- Anexa um volume de 20 GB.

- Executa um script de inicialização (user\_data) para atualizar o sistema.

9. \*Outputs\*:

- private\_key: Exibe a chave privada gerada (sensível).

- ec2\_public\_ip: Mostra o IP público da instância EC2.

---

### 💡 \*Observações\*

- \*Credenciais da AWS\*: Certifique-se de que suas credenciais (aws configure) estejam configuradas corretamente no seu ambiente.

- \*Permissões\*: A conta da AWS precisa ter permissões suficientes para criar todos esses recursos.

- \*Custo\*: Mesmo recursos pequenos, como t2.micro, podem gerar custos na AWS.

## Alterações Implementadas

### 1. \*\*Melhorias de Segurança\*\*

- \*\*Regras de Segurança para SSH:\*\*

- O acesso SSH à instância EC2 foi restrito para permitir conexões apenas de um bloco de IPs específico. Isso evita acessos indesejados de qualquer lugar, aumentando a segurança do ambiente.

- A regra de segurança para SSH foi configurada da seguinte forma:

```hcl

ingress {

description = "Permitir SSH de um IP específico (exemplo: 203.0.113.0/24)"

from\_port = 22

to\_port = 22

protocol = "tcp"

cidr\_blocks = ["203.0.113.0/24"] # Substitua pelo seu IP ou bloco de IPs

}

```

- \*\*Chave Privada Sensível:\*\*

- A chave privada gerada para o acesso à instância EC2 foi marcada como \*\*sensível\*\* no Terraform, o que impede que o valor seja exibido durante a execução do Terraform.

```hcl

output "private\_key" {

description = "Chave privada para acessar a instância EC2"

value = tls\_private\_key.ec2\_key.private\_key\_pem

sensitive = true

}

```

### 2. \*\*Automação da Instalação do Nginx\*\*

- O script de inicialização da instância (`user\_data`) foi configurado para:

- Atualizar o sistema (`apt-get update`).

- Instalar o servidor \*\*Nginx\*\* automaticamente.

- Iniciar o serviço \*\*Nginx\*\* e garantir que ele seja reiniciado após a reinicialização do servidor.

O script de automação da instalação do Nginx é o seguinte:

```bash

#!/bin/bash

apt-get update -y

apt-get upgrade -y

apt-get install -y nginx

systemctl start nginx

systemctl enable nginx

echo "Servidor Nginx rodando com sucesso!" > /var/www/html/index.html

```

- \*\*Resultado esperado:\*\* A instância EC2 será criada com o Nginx instalado e rodando automaticamente. A página inicial do servidor exibirá a mensagem "Servidor Nginx rodando com sucesso!".

### 3. \*\*Outras Melhorias\*\*

- \*\*Perfil IAM para a Instância EC2:\*\*

- A instância EC2 agora possui um \*\*IAM Role\*\* associado, permitindo que ela tenha permissões específicas e controladas.

- Embora não tenha sido configurada uma política IAM detalhada neste exemplo, é recomendável adicionar permissões mínimas de acordo com as necessidades específicas da aplicação.

O código do perfil IAM:

```hcl

resource "aws\_iam\_role" "ec2\_role" {

name = "${var.projeto}-${var.candidato}-ec2-role"

assume\_role\_policy = jsonencode({

Version = "2012-10-17"

Statement = [

{

Action = "sts:AssumeRole"

Effect = "Allow"

Principal = {

Service = "ec2.amazonaws.com"

}

},

]

})

}

```

- \*\*Grupo de Segurança Mais Restrito:\*\*

- O grupo de segurança foi configurado para permitir somente o tráfego necessário, com ênfase no acesso SSH restrito a um conjunto específico de IPs.

## Como Usar

### Pré-requisitos

- [Terraform](https://www.terraform.io/downloads.html) instalado na sua máquina.

- Conta na AWS configurada com as credenciais apropriadas.

### Passos para Rodar o Código

1. \*\*Clone o repositório:\*\*

```bash

git clone https://github.com/seu-usuario/projeto-vexpenses.git

cd projeto-vexpenses

```

2. \*\*Inicialize o Terraform:\*\*

O comando a seguir irá baixar os provedores necessários e preparar o ambiente para execução.

```bash

terraform init

```

3. \*\*Valide a configuração do Terraform:\*\*

Verifique se a configuração está correta antes de aplicar as mudanças.

```bash

terraform validate

```

4. \*\*Veja o que será criado:\*\*

O comando `plan` mostra um resumo dos recursos que serão criados, alterados ou destruídos.

```bash

terraform plan

```

5. \*\*Aplique as mudanças:\*\*

Crie os recursos na AWS com o comando abaixo. O Terraform irá criar a infraestrutura definida no arquivo `main.tf`.

```bash

terraform apply

```

6. \*\*Obtenha a chave privada para acessar a instância EC2:\*\*

Após a execução do `terraform apply`, a chave privada gerada será fornecida como um \*\*output sensível\*\*. Lembre-se de armazená-la de forma segura.

7. \*\*Acesse sua instância EC2:\*\*

Com a chave privada salva, você pode acessar a instância EC2 através de SSH. Substitua o valor de `YOUR\_PUBLIC\_IP` pelo IP público da instância EC2 (fornecido como output do Terraform).

```bash

ssh -i path/to/your-private-key.pem ubuntu@YOUR\_PUBLIC\_IP

```

### Limpeza

Para destruir todos os recursos criados na AWS, execute:

```bash

terraform destroy