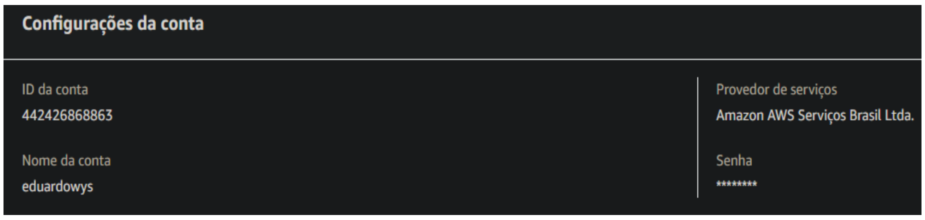
Desafio AWS + Terraform

Aluno: Eduardo Wesley Andrade Silva

O desafio consiste em realizar o processo de provisionamento da infra na AWS via Terraform, onde será criado o security group, a EC2 e por último o deploy do site (via o arquivo script.sh) que resultará na imagem abaixo (acessando via browser).

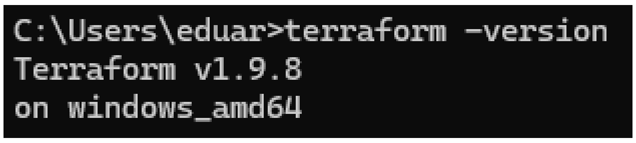
- Conta criada free no provedor de nuvem (AWS):



Configurações da conta

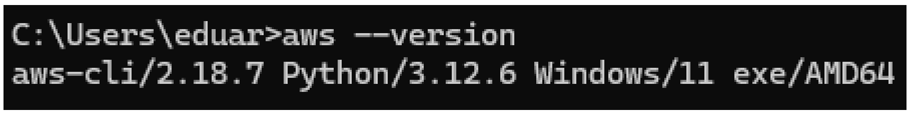
ID da conta 442426868863	Provedor de serviços Amazon AWS Serviços Brasil Ltda.
Nome da conta eduardowys	Senha ********

- Instale o Terraform localmente:



```
C:\Users\eduar>terraform -version
Terraform v1.9.8
on windows_amd64
```

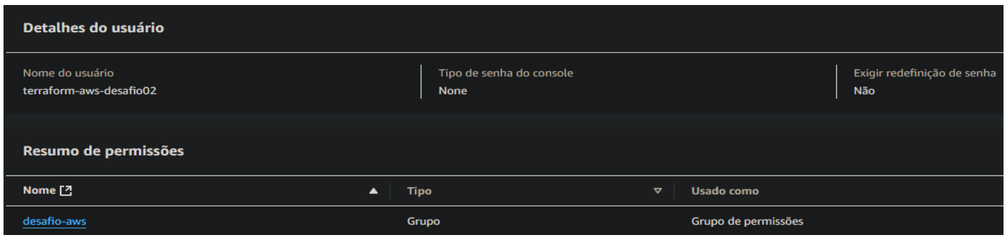
- Adicione um provedor (AWS) localmente:



```
C:\Users\eduar>aws --version
aws-cli/2.18.7 Python/3.12.6 Windows/11 exe/AMD64
```

- Escreva os arquivos de configuração. (receitinha de bolo):

- Criado o grupo e o usuário IAM:



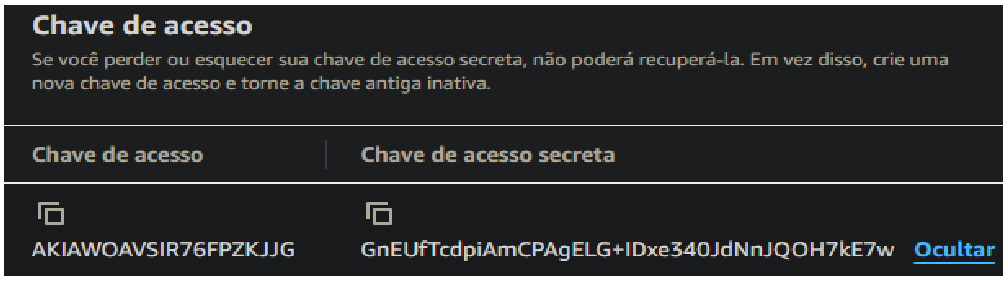
Detalhes do usuário

Nome do usuário	Tipo de senha do console	Exigir redefinição de senha
terraform-aws-desafio02	None	Não

Resumo de permissões

Nome	Tipo	Usado como
desafio-aws	Grupo	Grupo de permissões

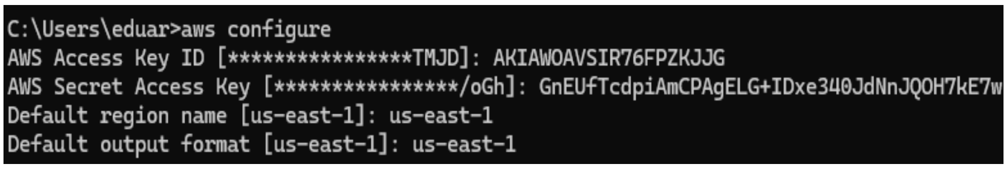
- Chave de acesso para vincular ao AWS CLI na máquina:



Chave de acesso

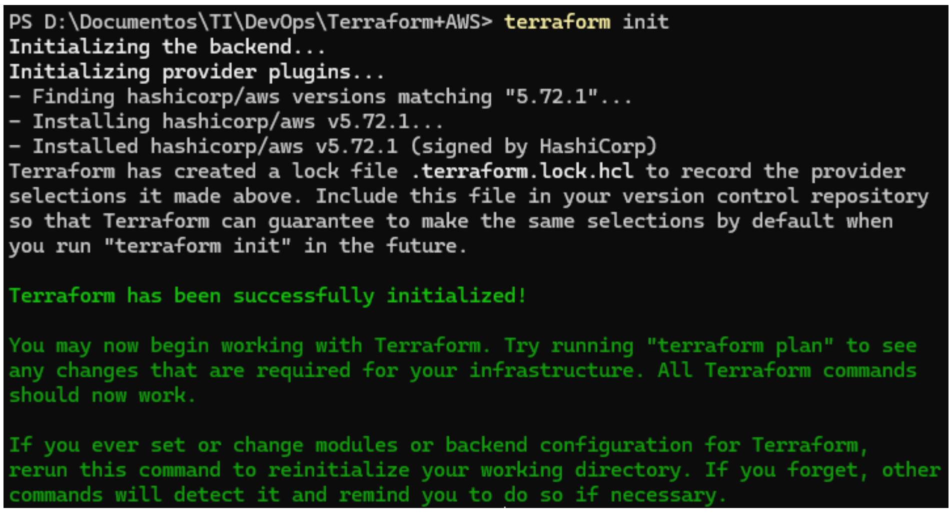
Se você perder ou esquecer sua chave de acesso secreta, não poderá recuperá-la. Em vez disso, crie uma nova chave de acesso e torne a chave antiga inativa.

Chave de acesso	Chave de acesso secreta
AKIAWOAVSIR76FPZKJJG	GnEUfTcdpiAmCPAgELG+IDxe340JdNnJQOH7kE7w Ocultar



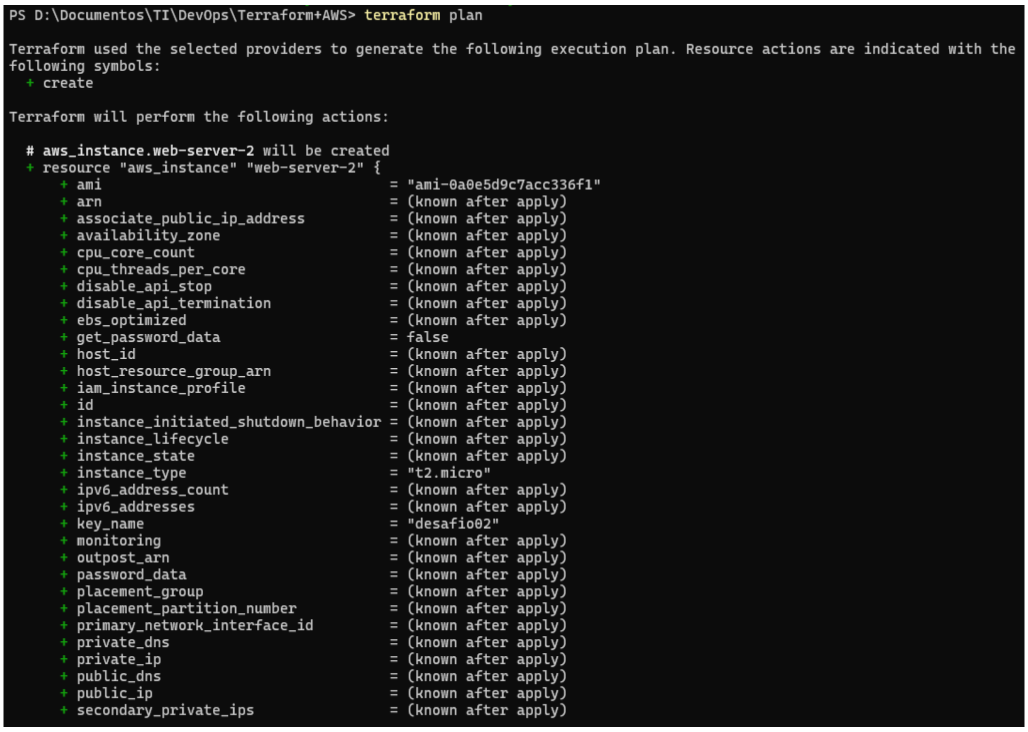
```
C:\Users\eduar>aws configure
AWS Access Key ID [****************TMJD]: AKIAWOAVSIR76FPZKJJG
AWS Secret Access Key [****************/oGh]: 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
Default region name [us-east-1]: us-east-1
Default output format [us-east-1]: us-east-1
```

- Criada instância EC2 Ubuntu 24.04 LTS - Terraform:
    - Inicialize o Terraform. (# **terraform init**):



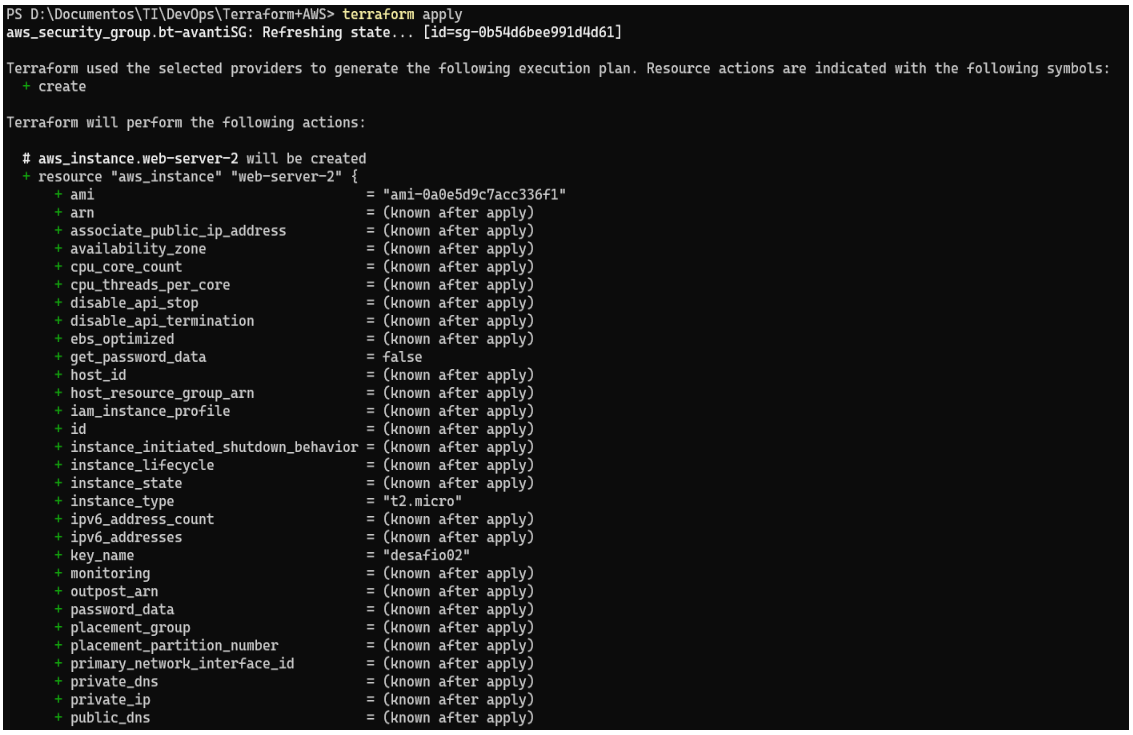
```
PS D:\Documentos\TI\DevOps\Terraform+AWS> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.72.1"...
- Installing hashicorp/aws v5.72.1...
- Installed hashicorp/aws v5.72.1 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

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.
```

    - Visualizar a infraestrutura a ser criada. **(# terraform plan**):

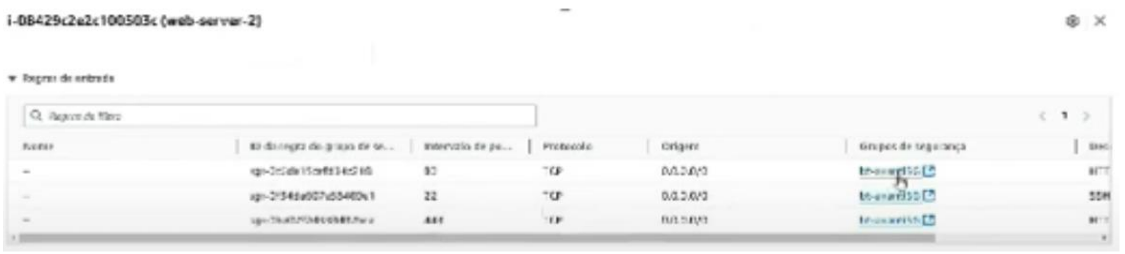


```
PS D:\Documentos\TI\DevOps\Terraform+AWS> terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.web-server-2 will be created
  + resource "aws_instance" "web-server-2" {
      + ami                                  = "ami-0a0e5d9c7acc336f1"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "desafio02"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
```

- O provisionamento na AWS. **(# terraform apply):**



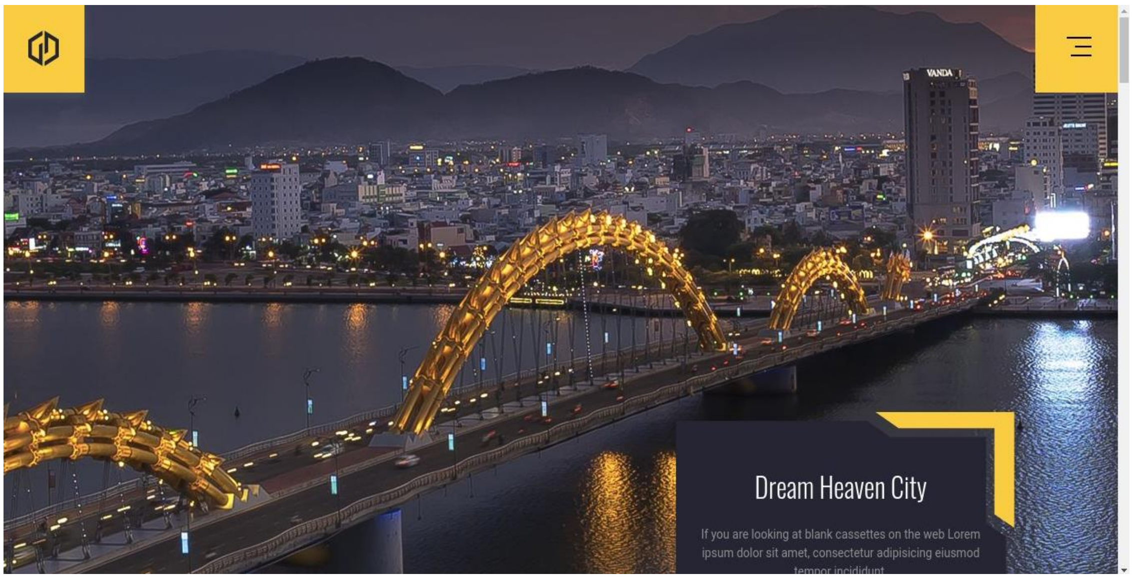
```
PS D:\Documentos\TI\DevOps\Terraform+AWS> terraform apply
aws_security_group.bt-avantiSG: Refreshing state... [id=sg-0b54d6bee991d4d61]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.web-server-2 will be created
  + resource "aws_instance" "web-server-2" {
      + ami                                  = "ami-0a0e5d9c7acc336f1"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "desafio02"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
```



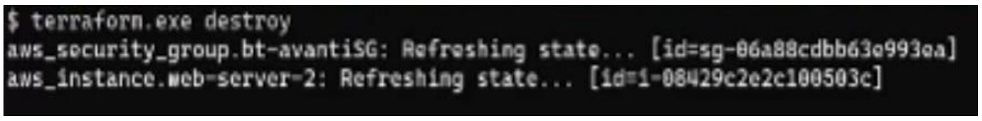
i-08429c2e2c100503c (web-server-2)

▼ Regras de entrada

Nome	ID da regra do grupo de se...	Intervalo de po...	Protocolo	Origem	Grupos de segurança	Desc
–	[illegible]	80	TCP	0.0.0.0/0	bt-avantiSG	HTTP
–	[illegible]	22	TCP	0.0.0.0/0	bt-avantiSG	SSH
–	[illegible]	443	TCP	0.0.0.0/0	bt-avantiSG	HTTPS


Dream Heaven City
If you are looking at blank cassettes on the web Lorem ipsum dolor sit amet, consectetur adipisicing eiusmod tempor incididunt

- Para eliminar os recursos provisionados. (**# terraform destroy**)



```
$ terraform.exe destroy
aws_security_group.bt-avantiSG: Refreshing state... [id=sg-06a88cdbb63e993ea]
aws_instance.web-server-2: Refreshing state... [id=i-08429c2e2c100503c]
```