DESAFIO KRONTON

Criado o desafio em Terraform que está no github:

Comandos executados:

- terraform init
- terraform plan -out=”tfplan.out”

```
aws_volume_attachment.ebs-volume-1-attachment: Refreshing state... [id=vai-1327941778]

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_ebs_volume.ebs-volume-1 will be created
  + resource "aws_ebs_volume" "ebs-volume-1" {
      + arn               = (known after apply)
      + availability_zone = "us-east-1a"
      + encrypted         = (known after apply)
      + id                = (known after apply)
      + iops              = (known after apply)
      + kms_key_id        = (known after apply)
      + size              = 10
      + snapshot_id       = (known after apply)
      + tags              = {
          + "Name" = "volume para log nginx"
        }
      + tags_all          = {
          + "Name" = "volume para log nginx"
        }
      + throughput        = (known after apply)
      + type              = "gp2"
    }

  # aws_eip.jvn-nat-eip will be created
  + resource "aws_eip" "jvn-nat-eip" {
      + allocation_id        = (known after apply)
      + association_id       = (known after apply)
      + carrier_ip           = (known after apply)
      + customer_owned_ip    = (known after apply)
      + domain               = (known after apply)
      + id                   = (known after apply)
      + instance             = (known after apply)
      + network_border_group = (known after apply)
      + network_interface    = (known after apply)
      + private_dns          = (known after apply)
      + private_ip           = (known after apply)
      + public_dns           = (known after apply)
      + public_ip            = (known after apply)
      + public_ipv4_pool     = (known after apply)
      + tags_all             = (known after apply)
      + vpc                  = true
    }

  # aws_instance.desafio_kroton will be created
  + resource "aws_instance" "desafio_kroton" {
      + ami                                  = "ami-0747bdcabd34c712a"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_termination              = false
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (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                             = "acesso_miami"
```

```
+ outpost_arn                  = (known after apply)
+ password_data                = (known after apply)
+ placement_group              = (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)
+ security_groups              = (known after apply)
+ source_dest_check            = true
+ subnet_id                    = (known after apply)
+ tags                         = {
    + "Name" = "Desafio_Kroton"
  }
+ tags_all                     = {
    + "Name" = "Desafio_Kroton"
  }
+ tenancy                      = (known after apply)
+ user_data                    = "8f0a60a5d72fecd08c9742d685fffb079e6b889f"
+ vpc_security_group_ids       = (known after apply)

+ capacity_reservation_specification {
    + capacity_reservation_preference = (known after apply)

    + capacity_reservation_target {
        + capacity_reservation_id = (known after apply)
      }
  }

+ ebs_block_device {
    + delete_on_termination = (known after apply)
    + device_name           = (known after apply)
    + encrypted             = (known after apply)
    + iops                  = (known after apply)
    + kms_key_id            = (known after apply)
    + snapshot_id           = (known after apply)
    + tags                  = (known after apply)
    + throughput            = (known after apply)
    + volume_id             = (known after apply)
    + volume_size           = (known after apply)
    + volume_type           = (known after apply)
  }

+ enclave_options {
    + enabled = (known after apply)
  }

+ ephemeral_block_device {
    + device_name  = (known after apply)
    + no_device    = (known after apply)
    + virtual_name = (known after apply)
  }

+ metadata_options {
    + http_endpoint               = (known after apply)
    + http_put_response_hop_limit = (known after apply)
    + http_tokens                 = (known after apply)
  }

+ network_interface {
    + delete_on_termination = (known after apply)
    + device_index          = (known after apply)
    + network_interface_id  = (known after apply)
  }

+ root_block_device {
    + delete_on_termination = (known after apply)
    + device_name           = (known after apply)
    + encrypted             = (known after apply)
    + iops                  = (known after apply)
    + kms_key_id            = (known after apply)
    + tags                  = (known after apply)
    + throughput            = (known after apply)
    + volume_id             = (known after apply)
    + volume_size           = (known after apply)
    + volume_type           = (known after apply)
```

```
    }
  }

# aws_internet_gateway.terragatwey will be created
+ resource "aws_internet_gateway" "terragatwey" {
    + arn      = (known after apply)
    + id       = (known after apply)
    + owner_id = (known after apply)
    + tags     = {
        + "Name" = "terragatwey"
      }
    + tags_all = {
        + "Name" = "terragatwey"
      }
    + vpc_id   = (known after apply)
  }

# aws_nat_gateway.jvnGW-nat will be created
+ resource "aws_nat_gateway" "jvnGW-nat" {
    + allocation_id        = (known after apply)
    + connectivity_type    = "public"
    + id                   = (known after apply)
    + network_interface_id = (known after apply)
    + private_ip           = (known after apply)
    + public_ip            = (known after apply)
    + subnet_id            = (known after apply)
    + tags_all             = (known after apply)
  }

# aws_route_table.jvn-private-rt will be created
+ resource "aws_route_table" "jvn-private-rt" {
    + arn              = (known after apply)
    + id               = (known after apply)
    + owner_id         = (known after apply)
    + propagating_vgws = (known after apply)
    + route            = (known after apply)
    + tags             = {
        + "Name" = "jvn-private-rt"
      }
    + tags_all         = {
        + "Name" = "jvn-private-rt"
      }
    + vpc_id           = (known after apply)
  }

# aws_route_table.jvn-public-rt will be created
+ resource "aws_route_table" "jvn-public-rt" {
    + arn              = (known after apply)
    + id               = (known after apply)
    + owner_id         = (known after apply)
    + propagating_vgws = (known after apply)
    + route            = [
        + {
            + carrier_gateway_id         = ""
            + cidr_block                 = "0.0.0.0/0"
            + destination_prefix_list_id = ""
            + egress_only_gateway_id     = ""
            + gateway_id                 = (known after apply)
            + instance_id                = ""
            + ipv6_cidr_block            = ""
            + local_gateway_id           = ""
            + nat_gateway_id             = ""
            + network_interface_id       = ""
            + transit_gateway_id         = ""
            + vpc_endpoint_id            = ""
            + vpc_peering_connection_id  = ""
          },
      ]
    + tags             = {
        + "Name" = "jvn-public-rt"
      }
    + tags_all         = {
        + "Name" = "jvn-public-rt"
      }
    + vpc_id           = (known after apply)
  }
```

```
# aws_route_table_association.jvnr-private-rta will be created
+ resource "aws_route_table_association" "jvnr-private-rta" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# aws_route_table_association.jvnr-public-rta will be created
+ resource "aws_route_table_association" "jvnr-public-rta" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# aws_security_group.jvnSG will be created
+ resource "aws_security_group" "jvnSG" {
    + arn                    = (known after apply)
    + description            = "Managed by Terraform"
    + egress                 = [
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
              ]
            + description      = ""
            + from_port        = 0
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "-1"
            + security_groups  = []
            + self             = false
            + to_port          = 0
          },
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
              ]
            + description      = ""
            + from_port        = 22
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 22
          },
      ]
    + id                     = (known after apply)
    + ingress                = [
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
              ]
            + description      = ""
            + from_port        = 22
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 22
          },
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
              ]
            + description      = ""
            + from_port        = 443
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 443
          },
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
```

```
            ]
          + description      = ""
          + from_port        = 80
          + ipv6_cidr_blocks = []
          + prefix_list_ids  = []
          + protocol         = "tcp"
          + security_groups  = []
          + self             = false
          + to_port          = 80
        },
    ]
  + name                   = "jvnSG"
  + name_prefix            = (known after apply)
  + owner_id               = (known after apply)
  + revoke_rules_on_delete = false
  + tags                   = {
      + "Name" = "jvnSG"
    }
  + tags_all               = {
      + "Name" = "jvnSG"
    }
  + vpc_id                 = (known after apply)
}

# aws_subnet.jvnsn-private will be created
+ resource "aws_subnet" "jvnsn-private" {
  + arn                                            = (known after apply)
  + assign_ipv6_address_on_creation                = false
  + availability_zone                              = "us-east-1b"
  + availability_zone_id                           = (known after apply)
  + cidr_block                                     = "10.3.100.0/24"
  + id                                             = (known after apply)
  + ipv6_cidr_block_association_id                 = (known after apply)
  + map_public_ip_on_launch                        = false
  + owner_id                                       = (known after apply)
  + tags                                           = {
      + "Name" = "jvnsn-private"
    }
  + tags_all                                       = {
      + "Name" = "jvnsn-private"
    }
  + vpc_id                                         = (known after apply)
}

# aws_subnet.jvnsn-public will be created
+ resource "aws_subnet" "jvnsn-public" {
  + arn                                            = (known after apply)
  + assign_ipv6_address_on_creation                = false
  + availability_zone                              = "us-east-1a"
  + availability_zone_id                           = (known after apply)
  + cidr_block                                     = "10.3.10.0/24"
  + id                                             = (known after apply)
  + ipv6_cidr_block_association_id                 = (known after apply)
  + map_public_ip_on_launch                        = false
  + owner_id                                       = (known after apply)
  + tags                                           = {
      + "Name" = "jvnsn-public"
    }
  + tags_all                                       = {
      + "Name" = "jvnsn-public"
    }
  + vpc_id                                         = (known after apply)
}

# aws_volume_attachment.ebs-volume-1-attachment will be created
+ resource "aws_volume_attachment" "ebs-volume-1-attachment" {
  + device_name = "/dev/sdh"
  + id          = (known after apply)
  + instance_id = (known after apply)
  + volume_id   = (known after apply)
}

# aws_vpc.jvnVPC will be created
+ resource "aws_vpc" "jvnVPC" {
  + arn                                            = (known after apply)
  + assign_generated_ipv6_cidr_block               = false
  + cidr_block                                     = "10.3.0.0/16"
```

```
    + default_network_acl_id          = (known after apply)
    + default_route_table_id          = (known after apply)
    + default_security_group_id       = (known after apply)
    + dhcp_options_id                 = (known after apply)
    + enable_classiclink              = (known after apply)
    + enable_classiclink_dns_support  = (known after apply)
    + enable_dns_hostnames            = true
    + enable_dns_support              = true
    + id                              = (known after apply)
    + instance_tenancy                = "default"
    + ipv6_association_id             = (known after apply)
    + ipv6_cidr_block                 = (known after apply)
    + main_route_table_id             = (known after apply)
    + owner_id                        = (known after apply)
    + tags                            = {
        + "Name" = "jvnVPC"
      }
    + tags_all                        = {
        + "Name" = "jvnVPC"
      }
  }

Plan: 14 to add, 0 to change, 0 to destroy.

------------------------------------------------------------------------

This plan was saved to: tfplan.out

To perform exactly these actions, run the following command to apply:
    terraform apply "tfplan.out"
```

Após execução do “plan” executamos o comando:

- terraform apply "tfplan.out" (iniciamos a criação da infra e de todo o processo)

```
aws_ebs_volume.ebs-volume-1: Creating...
aws_vpc.jvnVPC: Creating...
aws_vpc.jvnVPC: Still creating... [10s elapsed]
aws_ebs_volume.ebs-volume-1: Still creating... [10s elapsed]
aws_ebs_volume.ebs-volume-1: Creation complete after 12s [id=vol-0a86f4d98eb774195]
aws_vpc.jvnVPC: Creation complete after 19s [id=vpc-0d97c38768f969c89]
aws_internet_gateway.terragatwey: Creating...
aws_subnet.jvnsn-public: Creating...
aws_route_table.jvn-private-rt: Creating...
aws_subnet.jvnsn-private: Creating...
aws_security_group.jvnSG: Creating...
aws_route_table.jvn-private-rt: Creation complete after 3s [id=rtb-04808c3ae23e3ef5f]
aws_subnet.jvnsn-public: Creation complete after 3s [id=subnet-095545e17e53ed0b1]
aws_subnet.jvnsn-private: Creation complete after 3s [id=subnet-096f783aaf713b6de]
aws_route_table_association.jvnr-private-rta: Creating...
aws_route_table_association.jvnr-private-rta: Creation complete after 1s [id=rtbassoc-02404c6f243fb0cd6]
aws_internet_gateway.terragatwey: Creation complete after 4s [id=igw-08c857ed0a290fdbf]
aws_eip.jvn-nat-eip: Creating...
aws_route_table.jvn-public-rt: Creating...
aws_eip.jvn-nat-eip: Creation complete after 2s [id=eipalloc-0889f77bac0387728]
aws_nat_gateway.jvnGW-nat: Creating...
aws_security_group.jvnSG: Creation complete after 7s [id=sg-0143aca31264bdf2d]
aws_instance.desafio_kroton: Creating...
aws_route_table.jvn-public-rt: Creation complete after 3s [id=rtb-0c202210706747931]
aws_route_table_association.jvnr-public-rta: Creating...
aws_route_table_association.jvnr-public-rta: Creation complete after 1s [id=rtbassoc-0d77be1559cb45976]
aws_nat_gateway.jvnGW-nat: Still creating... [10s elapsed]
aws_instance.desafio_kroton: Still creating... [10s elapsed]
aws_nat_gateway.jvnGW-nat: Still creating... [20s elapsed]
aws_instance.desafio_kroton: Still creating... [20s elapsed]
aws_nat_gateway.jvnGW-nat: Still creating... [30s elapsed]
aws_instance.desafio_kroton: Still creating... [30s elapsed]
aws_nat_gateway.jvnGW-nat: Still creating... [40s elapsed]
aws_instance.desafio_kroton: Still creating... [40s elapsed]
aws_nat_gateway.jvnGW-nat: Still creating... [50s elapsed]
aws_instance.desafio_kroton: Still creating... [50s elapsed]
aws_instance.desafio_kroton: Creation complete after 50s [id=i-08fdaea240d09381a]
aws_volume_attachment.ebs-volume-1-attachment: Creating...
aws_nat_gateway.jvnGW-nat: Still creating... [1m0s elapsed]
aws_volume_attachment.ebs-volume-1-attachment: Still creating... [10s elapsed]
aws_nat_gateway.jvnGW-nat: Still creating... [1m10s elapsed]
aws_volume_attachment.ebs-volume-1-attachment: Still creating... [20s elapsed]
aws_volume_attachment.ebs-volume-1-attachment: Creation complete after 24s [id=vai-1939173982]
aws_nat_gateway.jvnGW-nat: Still creating... [1m20s elapsed]
aws_nat_gateway.jvnGW-nat: Still creating... [1m30s elapsed]
aws_nat_gateway.jvnGW-nat: Creation complete after 1m34s [id=nat-0c19f96cf9d4d6b31]

Apply complete! Resources: 14 added, 0 changed, 0 destroyed.

The state of your infrastructure has been saved to the path
below. This state is required to modify and destroy your
infrastructure, so keep it safe. To inspect the complete state
use the `terraform show` command.
```

Processo com terraform finalizado com SUCESSO!

Evidencias do Nginx com IP Publico funcionando corretamente!

Deployed via Terraform - Desafio Kronton - by Luiz Carlos Nascimento Junior

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

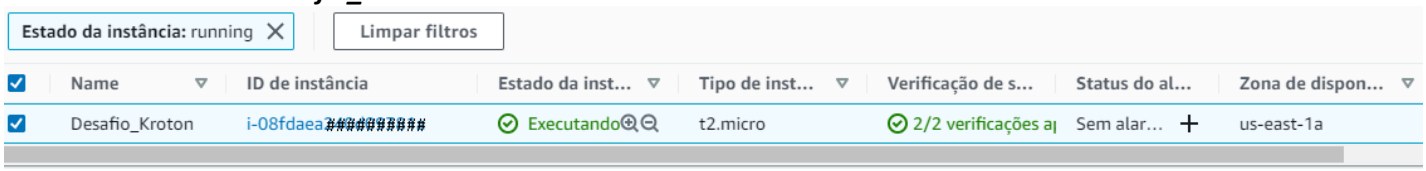
Storage/Filesystem montado.

```
root@ip-10-3-10-162:/home/ubuntu# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            476M     0  476M   0% /dev
tmpfs            98M  780K   98M   1% /run
/dev/xvda1      7.7G  1.4G  6.4G  18% /
tmpfs           490M     0  490M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           490M     0  490M   0% /sys/fs/cgroup
/dev/loop0       33M   33M     0 100% /snap/snapd/11588
/dev/loop1       56M   56M     0 100% /snap/core18/1997
/dev/loop2       34M   34M     0 100% /snap/amazon-ssm-agent/3552
/dev/xvdh       9.8G   37M  9.3G   1% /var/log/nginx
tmpfs            98M     0   98M   0% /run/user/1000
root@ip-10-3-10-162:/home/ubuntu# cd /var/log/nginx/
root@ip-10-3-10-162:/var/log/nginx# ls -ltr
total 20
drwx------ 2 root root 16384 Jul 23 04:13 lost+found
-rw-r--r-- 1 root root     0 Jul 23 04:13 error.log
-rw-r--r-- 1 root root  2620 Jul 23 04:33 access.log
```

Arquivo “index.html” que é feito um echo no script “user-data.sh”

```
root@ip-10-3-10-162:/var/www/html# ls -ltr
total 8
-rw-r--r-- 1 root root 612 Jul 23 04:13 index.nginx-debian.html
-rw-r--r-- 1 root root  85 Jul 23 05:07 index.html
root@ip-10-3-10-162:/var/www/html# cat index.html
<h1>Deployed via Terraform - Desafio Kronton - by Luiz Carlos Nascimento Junior</h1>
root@ip-10-3-10-162:/var/www/html#
```

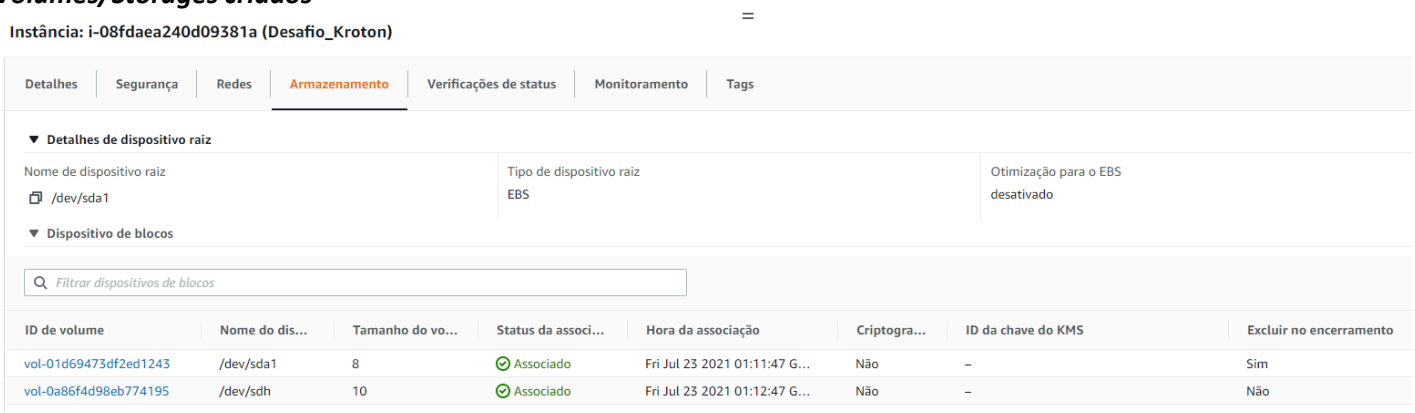
EC2 Criado na AWS "Desafio_Kroton"



Estado da instância: running | Limpar filtros

Name	ID de instância	Estado da inst...	Tipo de inst...	Verificação de s...	Status do al...	Zona de dispon...
Desafio_Kroton	i-08fdaea##########	Executando	t2.micro	2/2 verificações a	Sem alar...	us-east-1a

Volumes/Storages criados



Instância: i-08fdaea240d09381a (Desafio_Kroton)

Detalhes | Segurança | Redes | Armazenamento | Verificações de status | Monitoramento | Tags

▼ Detalhes de dispositivo raiz

Nome de dispositivo raiz	Tipo de dispositivo raiz	Otimização para o EBS
/dev/sda1	EBS	desativado

▼ Dispositivo de blocos

Filtrar dispositivos de blocos

ID de volume	Nome do dis...	Tamanho do vo...	Status da associ...	Hora da associação	Criptogra...	ID da chave do KMS	Excluir no encerramento
vol-01d69473df2ed1243	/dev/sda1	8	Associado	Fri Jul 23 2021 01:11:47 G...	Não	–	Sim
vol-0a86f4d98eb774195	/dev/sdh	10	Associado	Fri Jul 23 2021 01:12:47 G...	Não	–	Não

Volume "/dev/sda1" 8Gb para o "/"
Volume "/dev/sdh" 10Gb para "/var/log/nginx"

VPC "jvnVPC" criado

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR (Network border group)	IPv6 pool	DHCP options set
jvnVPC	vpc-0d97c38768f969c89	Available	10.3.0.0/16	–	–	dopt-c081aaba

Subnets/Router tables/Internet gateways/Security Groups/Inbound rules

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresses	Availabi
jvnsn-public	subnet-095545e17e53ed0b1	Available	vpc-0d97c38768f969c89 | jvn...	10.3.10.0/24	–	249	us-east-1
jvnsn-private	subnet-096f783aaf713b6de	Available	vpc-0d97c38768f969c89 | jvn...	10.3.100.0/24	–	251	us-east-1

Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC	Owner ID
jvn-public-rt	rtb-0c202210706747931	subnet-095545e17e53e...	–	No	vpc-0d97c38768f969c89 | jvn...	013579297658
jvn-private-rt	rtb-04808c3ae23e3ef5f	subnet-096f783aaf713b...	–	No	vpc-0d97c38768f969c89 | jvn...	013579297658

Name	Internet gateway ID	State	VPC ID	Owner
terragatwey	igw-08c857ed0a290fdbf	Attached	vpc-0d97c38768f969c89 | jvnVPC	013579297658

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules count
jvnSG	sg-0143aca31264bdf2d	jvnSG	vpc-0d97c38768f969c89	Managed by Terraform	013579297658	3 Permission entries	2 Permission entries

Inbound rules (3)

Filter security group rules

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
–	sgr-0303fd3585aca2d78	IPv4	SSH	TCP	22	0.0.0.0/0
–	sgr-07bb4a06fc41ddcf8	IPv4	HTTPS	TCP	443	0.0.0.0/0
–	sgr-06a32161ea22b64f7	IPv4	HTTP	TCP	80	0.0.0.0/0

Para replicar este ambiente:

- Fazer o "git clone" do projeto.
- Acessar a pasta “ec2” do projeto dentro de "desafio_kroton_luiz"

Obs.: Lembrando que você tem que fazer o "export" da sua:
"AWS_ACCESS_KEY_ID", "AWS_SECRET_ACCESS_KEY" e também "AWS_DEFAULT_REGION"

```
export AWS_ACCESS_KEY_ID=(cole aqui)
    ex.: export AWS_ACCESS_KEY_ID=lksjda35lL61JASsjaljndla124sjhda
export AWS_SECRET_ACCESS_KEY=(cole aqui)
export AWS_DEFAULT_REGION=us-west-2
```

- Após os "export" executar os comandos abaixo:

```
$ terraform init
$ terraform plan -out=”tfplan.out”
$ terraform apply "tfplan.out"
```