

Vinculando recursos criados via console AWS no projeto terraform



Vinculando recursos criados via console AWS no projeto terraform

Terraform Import

Comando do Terraform que permite que fazer o projeto assumir a propriedade sobre um recurso que foi criado diretamente via console AWS.

A lógica é a gente criar o recurso no Terraform, informar as suas configurações e vincular à versão que está configurada na AWS sem a necessidade de excluir ou inativá-lo.

Para isso usamos a documentação do Terraform para localizar o recurso que queremos importar e identificar o comando de importação que devemos anexar ao nosso projeto.

Vinculando recursos criados via console AWS no projeto terraform

Proposta

Serão importados os seguintes recursos de uma rede criada manualmente via console AWS

- VPC
- Subnet
- Route Table

The screenshot displays the AWS Management Console interface for a VPC named 'DougBrazil-vpc' (ID: vpc-0b7cab1ae3ae0ee3f). The 'Resource map' tab is active, showing a visual overview of the network resources. The VPC is connected to two subnets: 'us-east-1a' (DougBrazil-subnet-private1-us-east-1a) and 'us-east-1b' (DougBrazil-subnet-private2-us-east-1b). Three route tables are associated with the VPC: 'DougBrazil-rtb-private2-us-east-1b', 'rtb-0330c84d83c49fdb0', and 'DougBrazil-rtb-private1-us-east-1a'. The 'Network Connections' section shows zero connections. A red arrow points to the VPC box, and blue and red lines connect the subnets to their respective route tables.

vpc-0b7cab1ae3ae0ee3f / DougBrazil-vpc

Details **Resource map** CIDRs Flow logs Tags Integrations

Resource map Info Show all details

VPC
Your AWS virtual network
DougBrazil-vpc

Subnets (2)
Subnets within this VPC

- us-east-1a**
DougBrazil-subnet-private1-us-east-1a
- us-east-1b**
DougBrazil-subnet-private2-us-east-1b

Route tables (3)
Route network traffic to resources

- DougBrazil-rtb-private2-us-east-1b
- rtb-0330c84d83c49fdb0
- DougBrazil-rtb-private1-us-east-1a

Network Connections (0)
Connections to other networks

Vinculando recursos criados via console AWS no projeto terraform

Para começar

Devemos criar um projeto Terraform e fazer a codificação de todos os recursos que vamos importar.

Nesse exemplo:

- 1 VPC
- 2 Subnets
- 2 Route Tables + Associações com as subnets

Vinculando recursos criados via console AWS no projeto terraform

Para começar

```
() terraform.tfstate > ...
1  {
2    "version": 4,
3    "terraform_version": "1.13.5",
4    "serial": 11,
5    "lineage": "62896651-7a1c-92ff-a89d-66e68229cdb5",
6    "outputs": {},
7    "resources": [],
8    "check_results": null
9  }
10
```

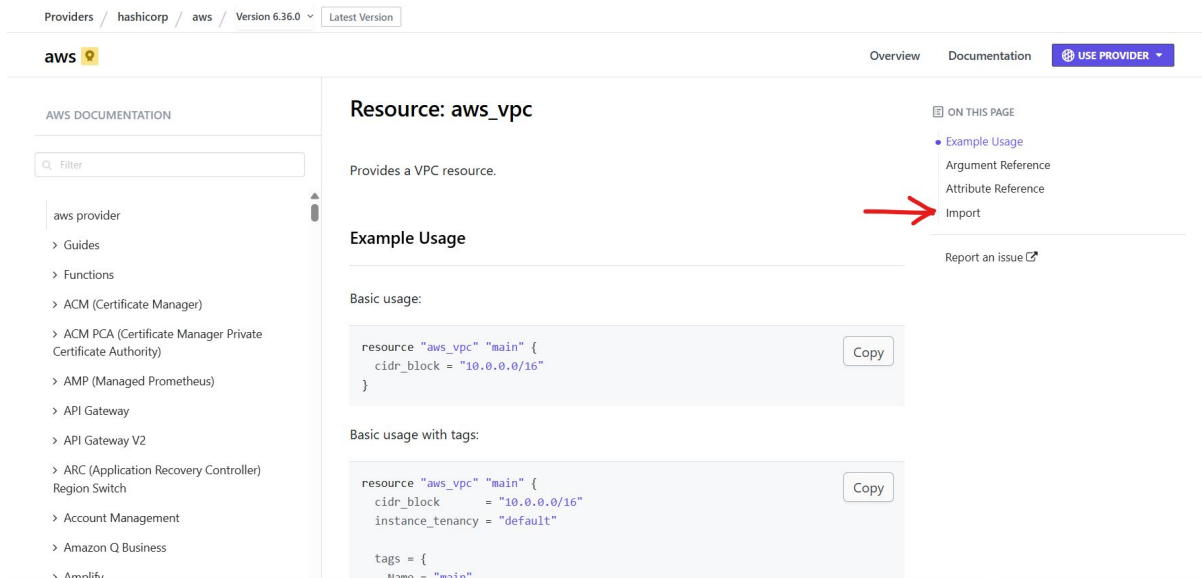
Verificamos que no momento o nosso arquivo TFState está vazio.

Ao final dessa apresentação ele vai conter as referências de todos os recursos importados.


Vinculando recursos criados via console AWS no projeto terraform

Importando VPC

Para iniciar, acessamos a página de [documentação do terraform](#) e buscamos o recurso que iremos importar.



Providers / hashicorp / aws / Version 6.36.0 ▾ Latest Version

aws 

Overview Documentation [USE PROVIDER ▾](#)

AWS DOCUMENTATION

Q Filter

- aws provider
 - > Guides
 - > Functions
 - > ACM (Certificate Manager)
 - > ACM PCA (Certificate Manager Private Certificate Authority)
 - > AMP (Managed Prometheus)
 - > API Gateway
 - > API Gateway V2
 - > ARC (Application Recovery Controller) Region Switch
 - > Account Management
 - > Amazon Q Business
 - > Amplify

Resource: aws_vpc

Provides a VPC resource.

Example Usage

Basic usage:

```
resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}
```

Copy

Basic usage with tags:

```
resource "aws_vpc" "main" {
  cidr_block      = "10.0.0.0/16"
  instance_tenancy = "default"

  tags = {
    Name = "main"
  }
}
```

Copy

ON THIS PAGE

- Example Usage
- Argument Reference
- Attribute Reference
- Import

Report an issue [↗](#)

Note que do lado direito é exibido link explicando como fazer o seu import.

Vinculando recursos criados via console AWS no projeto terraform

Importando VPC

Instruções de importação



The screenshot shows the Terraform documentation page for importing VPCs. On the left is a navigation sidebar with categories like 'aws provider', 'Guides', 'Functions', and various AWS services. The main content area is titled 'Import' and includes a list of arguments: 'owner_id' (the ID of the AWS account) and 'tags_all' (a map of tags). Below this, there are two code blocks: one for Terraform v1.5.0 and later using the 'import' block, and another for older versions using the 'terraform import' command. A right-hand sidebar contains 'Argument Reference' and 'Attribute Reference' sections, with 'Import' selected.

- `owner_id` - The ID of the AWS account that owns the VPC.
- `tags_all` - A map of tags assigned to the resource, including those inherited from the provider `default_tags` configuration block.

Import

In Terraform v1.5.0 and later, use an `import` block to import VPCs using the VPC `id`. For example:

```
import {
  to = aws_vpc.test_vpc
  id = "vpc-a01106c2"
}
```

Using `terraform import`, import VPCs using the VPC `id`. For example:

```
% terraform import aws_vpc.test_vpc vpc-a01106c2
```

Podemos observar 2 maneiras de importar o recurso:

- Linha de comando
- Bloco `import { }` (Terraform \geq 1.5.0)

Vinculando recursos criados via console AWS no projeto terraform

Importando VPC

Instruções de importação

The screenshot displays the AWS Management Console interface for VPCs. At the top, there's a search bar and a table listing VPCs. The table has columns for Name, VPC ID, State, Encryption controls, Block Public Access, IPv4 CIDR, and IPv6 CIDR. One VPC, 'DougBrazil-vpc' with ID 'vpc-0b7cab1ae3ae0ee3f', is highlighted. Below the table, the details for this VPC are shown in a grid format. Red arrows point to the VPC ID and IPv4 CIDR fields in both the table and the details view.

Your VPCs

VPCs | VPC encryption controls

Your VPCs (1/1) Info

Find VPCs by attribute or tag

DougBrazil-vpc X Clear filters

| Name | VPC ID | State | Encryption c... | Encryption control ... | Block Public... | IPv4 CIDR | IPv6 CIDR |
|----------------|-----------------------|-----------|-----------------|------------------------|-----------------|----------------|-----------|
| DougBrazil-vpc | vpc-0b7cab1ae3ae0ee3f | Available | - | - | Off | 192.168.0.0/24 | - |

vpc-0b7cab1ae3ae0ee3f / DougBrazil-vpc

Details | Resource map | CIDRs | Flow logs | Tags | Integrations

Details

| | | | |
|--|--|--|---------------------------------|
| VPC ID vpc-0b7cab1ae3ae0ee3f | State Available | Block Public Access Off | DNS hostnames Enabled |
| DNS resolution Enabled | Tenancy default | DHCP option set dopt-27223355d | Main route table - |
| Main network ACL - | Default VPC No | IPv4 CIDR 192.168.0.0/24 | IPv6 pool - |
| IPv6 CIDR (Network border group) - | Network Address Usage metrics Disabled | Route 53 Resolver DNS Firewall rule groups - | Owner ID 791659331772 |
| Encryption control ID - | Encryption control mode - | | |

Identificamos no console as informações que vamos precisar.

Vinculando recursos criados via console AWS no projeto terraform

Importando VPC

Instruções de importação (Usando Linha de comando)

```
1 resource "aws_vpc" "doughbr_vpc" {
2   cidr_block      = "192.168.0.0/24"
3
4   tags = {
5     Name = "DougBrazil-VPC"
6   }
7 }
```

iniciacao terraform import aws_vpc.doughbr_vpc vpc-0b7cab1ae3ae0ee3f
aws_vpc.doughbr_vpc: Importing from ID "vpc-0b7cab1ae3ae0ee3f" ...
aws_vpc.doughbr_vpc: Import prepared!
Prepared aws_vpc for import
aws_vpc.doughbr_vpc: Refreshing state... [id=vpc-0b7cab1ae3ae0ee3f]
import successful!
The resources that were imported are shown above. These resources are now in
your Terraform state and will henceforth be managed by Terraform.

Rogério P. Calixto iniciacao

Codificamos as informações obtidas no projeto Terraform e executamos na linha de comando o import para vincular o recurso que vamos importar com a sua referência na AWS.

```
terraform import aws_vpc.doughbr_vpc vpc-0b7cab1ae3ae0ee3f
```

O retorno “import successful!” indica que a importação foi bem sucedida.

Vinculando recursos criados via console AWS no projeto terraform

Importando Subnets

Neste caso a nossa VPC possui 2 subnets privadas que serão vinculadas ao nosso projeto.

Subnets (2) [Info](#)

Last upc
less than a minut

Find subnets by attribute or tag

DougBrazil ×

Clear filters

| <input type="checkbox"/> | Name | Subnet ID | State | VPC | Block Public... | IPv4 CIDR |
|--------------------------|---------------------------------------|--|-----------|---|-----------------|------------------|
| <input type="checkbox"/> | DougBrazil-subnet-private2-us-east-1b | subnet-0308c2df2ca7fb196 | Available | vpc-0b7cab1ae3ae0ee3f Doug... | Off | 192.168.0.144/28 |
| <input type="checkbox"/> | DougBrazil-subnet-private1-us-east-1a | subnet-0ce87a45d909b87ff | Available | vpc-0b7cab1ae3ae0ee3f Doug... | Off | 192.168.0.128/28 |

Vinculando recursos criados via console AWS no projeto terraform

Importando Subnets

Instruções de importação

In Terraform v1.5.0 and later, use an `import` block to import subnets using the subnet `id`.

For example:

```
import {  
  to = aws_subnet.example  
  id = "subnet-9d4a7b6c"  
}
```

Copy

Using `terraform import`, import subnets using the subnet `id`. For example:

```
% terraform import aws_subnet.example subnet-9d4a7b6c
```

Copy

Dessa vez usaremos o recurso do bloco `import`.

Vinculando recursos criados via console AWS no projeto terraform

Importando Subnets

Instruções de importação

```
vpc.subnet.tf > import > abc id
1 import {
2   to = aws_subnet.subnet_private_1
3   id = "subnet-0ce87a45d909b87ff"
4 }
5
6 resource "aws_subnet" "subnet_private_1" {
7   vpc_id     = aws_vpc.douibr_vpc.id
8   cidr_block = "192.168.0.128/28"
9   availability_zone = "us-east-1a"
10
11   tags = {
12     Name = "DougBr-subnet-private1-us-east-1a"
13   }
14 }
15
16 import {
17   to = aws_subnet.subnet_private_2
18   id = "subnet-0308c2df2ca7fb196"
19 }
20
21 resource "aws_subnet" "subnet_private_2" {
22   vpc_id     = aws_vpc.douibr_vpc.id
23   cidr_block = "192.168.0.144/28"
24   availability_zone = "us-east-1b"
25
26   tags = {
27     Name = "DougBr-subnet-private2-us-east-1b"
28   }
29 }
```

Configuramos nossas 2 subnets privadas com as informações obtidas na AWS, vinculamos elas ao VPC que havíamos importado anteriormente e por fim incluímos um bloco import para cada uma vinculando os IDs correspondentes na AWS.

Vinculando recursos criados via console AWS no projeto terraform

Importando Subnets

Instruções de importação

```
# aws_subnet_subnet_private_1 will be imported
resource "aws_subnet" "subnet_private_1" {
  arn                = "arn:aws:ec2:us-east-1:791659331772:subnet/subnet-8ce87a5d999b87ff"
  assign_ipv6_address_on_creation = false
  availability_zone  = "us-east-1a"
  availability_zone_id = "us-east-1a"
  cidr_block        = "192.168.0.128/28"
  customer_owned_ipv4_pool = null
  enable_dns64     = false
  enable_lni_at_device_index = 0
  enable_resource_name_dns_a_record_on_launch = false
  enable_resource_name_dns_aaaa_record_on_launch = false
  id                = "subnet-8ce87a5d999b87ff"
  ipv6_cidr_block   = null
  ipv6_cidr_block_association_id = null
  ipv6_native       = false
  map_customer_owned_ip_on_launch = false
  map_public_ip_on_launch = false
  outpost_arn       = null
  owner_id          = "791659331772"
  private_dns_hostname_type_on_launch = "ip-name"
  region            = "us-east-1"
  tags              = {
    "Name" = "DougBrazil-subnet-private1-us-east-1a"
  }
  tags_all          = {
    "Name" = "DougBrazil-subnet-private1-us-east-1a"
  }
  vpc_id            = "vpc-8b7c0ab3ae3ae3f3"
}

# aws_subnet_subnet_private_2 will be imported
resource "aws_subnet" "subnet_private_2" {
  arn                = "arn:aws:ec2:us-east-1:791659331772:subnet/subnet-8388c2df2ca77f6196"
  assign_ipv6_address_on_creation = false
  availability_zone  = "us-east-1b"
  availability_zone_id = "us-east-1b"
  cidr_block        = "192.168.0.144/28"
  customer_owned_ipv4_pool = null
  enable_dns64     = false
  enable_lni_at_device_index = 0
  enable_resource_name_dns_a_record_on_launch = false
  enable_resource_name_dns_aaaa_record_on_launch = false
  id                = "subnet-8388c2df2ca77f6196"
  ipv6_cidr_block   = null
  ipv6_cidr_block_association_id = null
  ipv6_native       = false
  map_customer_owned_ip_on_launch = false
  map_public_ip_on_launch = false
  outpost_arn       = null
  owner_id          = "791659331772"
  private_dns_hostname_type_on_launch = "ip-name"
  region            = "us-east-1"
  tags              = {
    "Name" = "DougBrazil-subnet-private2-us-east-1b"
  }
  tags_all          = {
    "Name" = "DougBrazil-subnet-private2-us-east-1b"
  }
  vpc_id            = "vpc-8b7c0ab3ae3ae3f3"
}

# Plan: 2 to import, 0 to add, 0 to change, 0 to destroy.
```

Ao executar o comando terraform plan, são identificados os 2 novos recursos para importar.

Vinculando recursos criados via console AWS no projeto terraform

Importando Subnets

Instruções de importação

```
Plan: 2 to import, 0 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_subnet.subnet_private_1: Importing... [id=subnet-0ce87a45d909b87ff]
aws_subnet.subnet_private_1: Import complete [id=subnet-0ce87a45d909b87ff]
aws_subnet.subnet_private_2: Importing... [id=subnet-0308c2df2ca7fb196]
aws_subnet.subnet_private_2: Import complete [id=subnet-0308c2df2ca7fb196]

Apply complete! Resources: 2 imported, 0 added, 0 changed, 0 destroyed.
```

E com o comando terraform apply a gente concretiza a importação.

Vinculando recursos criados via console AWS no projeto terraform

Importando Route Tables

Route tables (3) [Info](#)

Last updated
24 minutes ago 

Actions 

Cre...

Find route tables by attribute or tag

DougBrazil 

Clear filters

| <input type="checkbox"/> | Name | Route table ID | Explicit subnet associations | Edge... | Main | VPC |
|--------------------------|------------------------------------|---------------------------------------|--|---------|------|--|
| <input type="checkbox"/> | - | rtb-0330c84d83c49fdb0 | - | - | Yes | vpc-0b7cab1ae3ae0ee3f DougBrazil-vpc |
| <input type="checkbox"/> | DougBrazil-rtb-private1-us-east-1a | rtb-0a3779ea2238bf5a9 |  subnet-0ce87a45d909b87ff / DougBrazil-subnet-private1-us-east-1a | - | No | vpc-0b7cab1ae3ae0ee3f DougBrazil-vpc |
| <input type="checkbox"/> | DougBrazil-rtb-private2-us-east-1b | rtb-0373fa4dce1947d97 |  subnet-0308c2df2ca7fb196 / DougBrazil-subnet-private2-us-east-1b | - | No | vpc-0b7cab1ae3ae0ee3f DougBrazil-vpc |

Verificamos 2 tabelas de rotas, associadas uma a cada subnet.

Vamos colocá-las em nosso projeto, efetuar as associações com as subnets e aplicar os blocos import.

Vinculando recursos criados via console AWS no projeto terraform

Importando Route Tables

Instruções de importação

```
import {
  to = aws_route_table.rt_private_1
  id = "rtb-0a3779ea2238bf5a9"
}

resource "aws_route_table" "rt_private_1" {
  vpc_id = aws_vpc.douibr_vpc.id

  tags = {
    Name = "DougBrazil-rtb-private1-us-east-1a"
  }
}

import {
  to = aws_route_table.rt_private_2
  id = "rtb-0373fa4dce1947d97"
}

resource "aws_route_table" "rt_private_2" {
  vpc_id = aws_vpc.douibr_vpc.id

  tags = {
    Name = "DougBrazil-rtb-private2-us-east-1b"
  }
}
```

```
import {
  to = aws_route_table_association.rt_private_association_subnet1
  id = "subnet-0ce87a45d909b87ff /rtb-0a3779ea2238bf5a9"
}

resource "aws_route_table_association" "rt_private_association_subnet1" {
  route_table_id = aws_route_table.rt_private_1.id
  subnet_id      = aws_subnet.subnet_private_1.id
}

import {
  to = aws_route_table_association.rt_private_association_subnet2
  id = "subnet-0308c2df2ca7fb196/rtb-0373fa4dce1947d97"
}

resource "aws_route_table_association" "rt_private_association_subnet2" {
  route_table_id = aws_route_table.rt_private_2.id
  subnet_id      = aws_subnet.subnet_private_2.id
}
```

Vinculando recursos criados via console AWS no projeto terraform

Importando Route Tables

Instruções de importação

```
terraform will perform the following actions:
# aws_route_table.rt_private_1 will be imported
resource "aws_route_table" "rt_private_1" {
  arn          = "arn:aws:ec2:us-east-1:791659331772:route-table/rtb-0a3779ea2238bf5a9"
  id          = "rtb-0a3779ea2238bf5a9"
  owner_id    = "791659331772"
  propagating_vgws = []
  region      = "us-east-1"
  routes      = []
  tags        = {
    "Name" = "DougBrazil-rtb-privatel-us-east-1a"
  }
  tags_all    = {
    "Name" = "DougBrazil-rtb-privatel-us-east-1a"
  }
  vpc_id      = "vpc-007c81a3a80e3f"
}
# aws_route_table.rt_private_2 will be imported
resource "aws_route_table" "rt_private_2" {
  arn          = "arn:aws:ec2:us-east-1:791659331772:route-table/rtb-0373fa1dce1947d977"
  id          = "rtb-0373fa1dce1947d977"
  owner_id    = "791659331772"
  propagating_vgws = []
  region      = "us-east-1"
  routes      = []
  tags        = {
    "Name" = "DougBrazil-rtb-private2-us-east-1b"
  }
  tags_all    = {
    "Name" = "DougBrazil-rtb-private2-us-east-1b"
  }
  vpc_id      = "vpc-007c81a3a80e3f"
}
# aws_route_table_association.rt_private_association_subnet1 will be imported
resource "aws_route_table_association" "rt_private_association_subnet1" {
  gateway_id = null
  id         = "rtbassoc-0fbc4536af7b5867"
  region     = "us-east-1"
  route_table_id = "rtb-0a3779ea2238bf5a9"
  subnet_id  = "subnet-0ce87a45d9998bfff"
}
# aws_route_table_association.rt_private_association_subnet2 will be imported
resource "aws_route_table_association" "rt_private_association_subnet2" {
  gateway_id = null
  id         = "rtbassoc-03e92362b3ccea7a13"
  region     = "us-east-1"
  route_table_id = "rtb-0373fa1dce1947d977"
  subnet_id  = "subnet-0388c2d2f2ca7fb196"
}
Plan: 4 to import, 0 to add, 0 to change, 0 to destroy.
```

Ao executar o comando terraform plan, são identificados os novos recursos para importar.

Vinculando recursos criados via console AWS no projeto terraform

Importando Route Tables

Instruções de importação

```
Do you want to perform these actions?  
Terraform will perform the actions described above.  
Only 'yes' will be accepted to approve.  
  
Enter a value: yes  
  
aws_route_table.rt_private_2: Importing... [id=rtb-0373fa4dce1947d97]  
aws_route_table.rt_private_2: Import complete [id=rtb-0373fa4dce1947d97]  
aws_route_table.rt_private_1: Importing... [id=rtb-0a3779ea2238bf5a9]  
aws_route_table.rt_private_1: Import complete [id=rtb-0a3779ea2238bf5a9]  
aws_route_table_association.rt_private_association_subnet2: Importing... [id=subnet-0308c2df2ca7fb196/rtb-0373fa4dce1947d97]  
aws_route_table_association.rt_private_association_subnet2: Import complete [id=subnet-0308c2df2ca7fb196/rtb-0373fa4dce1947d97]  
aws_route_table_association.rt_private_association_subnet1: Importing... [id=subnet-0ce87a45d909b87ff/rtb-0a3779ea2238bf5a9]  
aws_route_table_association.rt_private_association_subnet1: Import complete [id=subnet-0ce87a45d909b87ff/rtb-0a3779ea2238bf5a9]  
  
Apply complete! Resources: 4 imported, 0 added, 0 changed, 0 destroyed.
```

E com o comando terraform apply a gente concretiza a importação.

Vinculando recursos criados via console AWS no projeto terraform

Resultado

Instruções de importação

Agora podemos verificar nosso [arquivo tfstate](#).

A partir desse momento os recursos criados no console aws, passam a ser gerenciados pelo terraform

Vinculando recursos criados via console AWS no projeto terraform

Compensa ?

Você pode se perguntar: “Se tenho que informar todos os recursos, não compensa criar do zero ?”

A resposta é: “**Depende**”

Se os recursos a importar possuem dependências com outros recursos e precisam ser mantidos operacionais, sem dúvida, a solução é importar.

Agora caso você possa recriar o recurso, aí o import não se torna necessário.

Vinculando recursos criados via console AWS no projeto terraform

Observações

- Com base na documentação do Terraform, podemos identificar tudo o que precisamos para efetuar a importação dos nossos recursos.
- Importante salientar que para manter o recurso na AWS inalterado temos que incluir na definição do recurso Terraform todas as configurações já existentes. Caso contrário seu código irá sobrescrever os recursos na AWS.
- Muito cuidado neste processo para que o código esteja correto para execução não correndo assim riscos de possíveis inconvenientes.

Vinculando recursos criados via console AWS no projeto terraform

