

Infraestructura como Código (IaC)



Infraestrutura como Código (IaC)

Definição

É o processo de criação e gerenciamento de uma infraestrutura qualquer através de codificação.

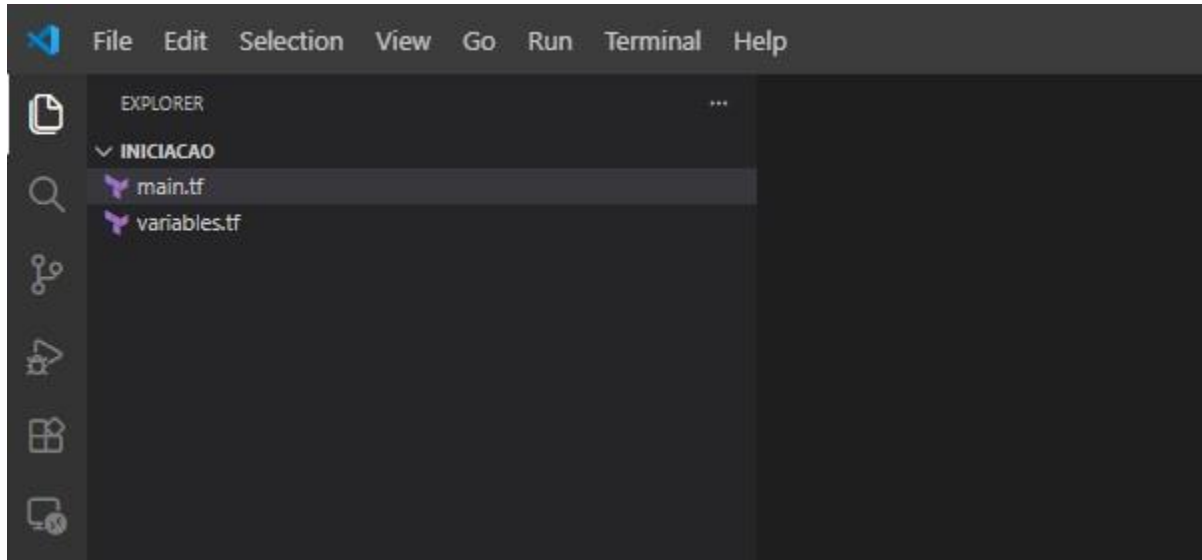
Depois de implementada a configuração em código, o processo de criação e gerenciamento da infraestrutura se torna mais rápido, permitindo ainda a replicação da mesma agilizando os processos.

Infraestrutura como Código (IaC)

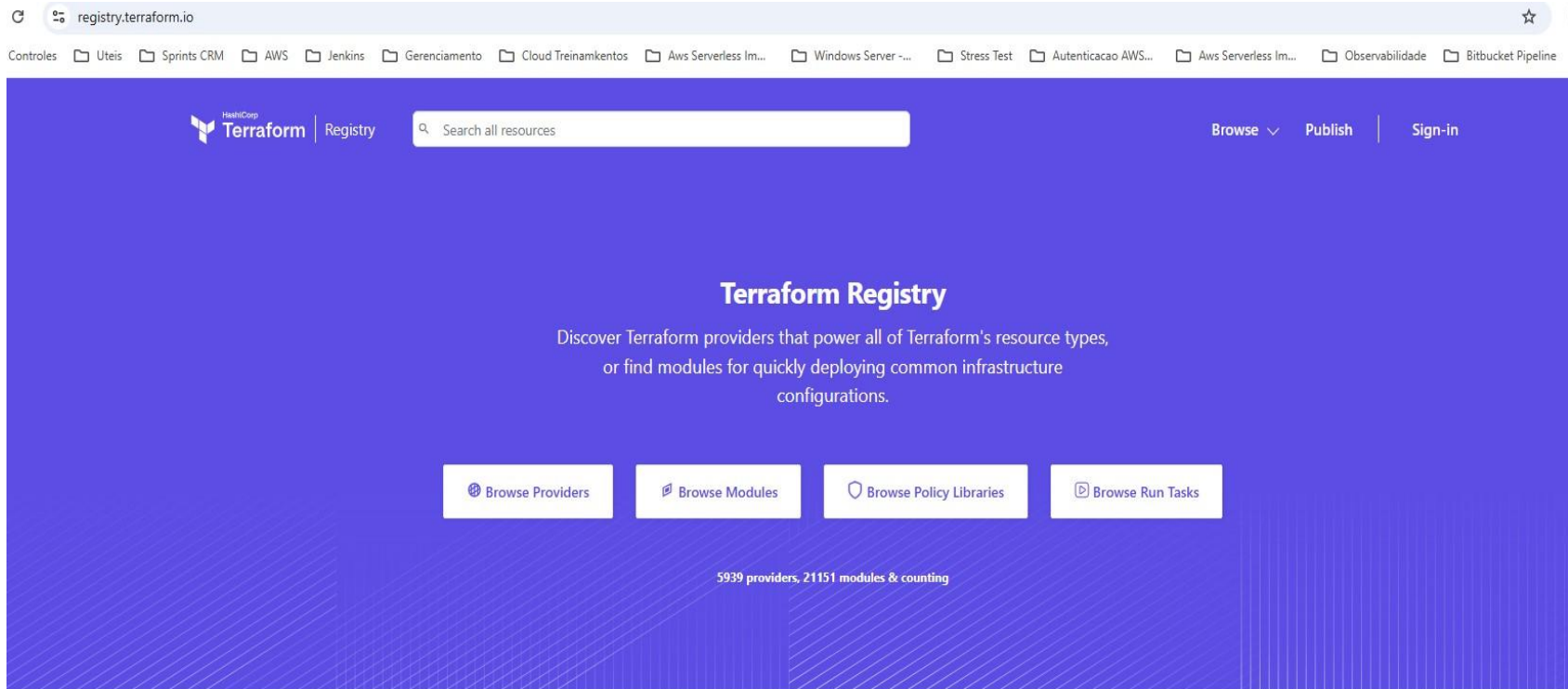
Ferramentas

- Terraform (Hashicorp)
- CloudFormation (AWS)
- Pulumi
- Ansible

Infraestrutura como Código (IaC)



Infraestrutura como Código (IaC)



The image shows a browser window with the URL `registry.terraform.io`. The browser's address bar and tabs are visible at the top. The page content is on a dark blue background. At the top left is the Terraform logo with the text "HashiCorp Terraform Registry". To its right is a search bar with the placeholder text "Search all resources". Further right are links for "Browse", "Publish", and "Sign-in". In the center, the heading "Terraform Registry" is followed by a descriptive paragraph: "Discover Terraform providers that power all of Terraform's resource types, or find modules for quickly deploying common infrastructure configurations." Below this are four white buttons with icons and text: "Browse Providers", "Browse Modules", "Browse Policy Libraries", and "Browse Run Tasks". At the bottom center, the text "5939 providers, 21151 modules & counting" is displayed.

registry.terraform.io

Controles Uteis Sprints CRM AWS Jenkins Gerenciamento Cloud Trainamkentos Aws Serverless Im... Windows Server -... Stress Test Autenticacao AWS... Aws Serverless Im... Observabilidade Bitbucket Pipeline

HashiCorp Terraform Registry

Search all resources

Browse Publish Sign-in

Terraform Registry

Discover Terraform providers that power all of Terraform's resource types, or find modules for quickly deploying common infrastructure configurations.

Browse Providers Browse Modules Browse Policy Libraries Browse Run Tasks

5939 providers, 21151 modules & counting

Infraestructura como Código (IaC)

The screenshot displays the Terraform Providers page. At the top, there are navigation tabs for Providers, Modules, Policy Libraries, and Run Tasks. On the left, a 'Filters' sidebar is visible, with 'Tier' set to 'Official', 'Partner', and 'Community', and 'Category' with various options like 'HashiCorp Platform', 'Infrastructure Management', etc. The main content area is titled 'Providers' and includes a description: 'Providers are a logical abstraction of an upstream API. They are responsible for understanding API interactions and exposing resources.' Below this, a grid of provider cards is shown, each with a logo and name: AWS, Azure, Google Cloud Platform, Kubernetes, Alibaba Cloud, and Oracle Cloud Infrastructure. At the bottom, three smaller cards represent modules: 'Aap' by ansible, 'Active Directory' by hashicorp, and 'Archive' by hashicorp.

Providers

Providers are a logical abstraction of an upstream API. They are responsible for understanding API interactions and exposing resources.

- AWS
- Azure
- Google Cloud Platform
- Kubernetes
- Alibaba Cloud
- Oracle Cloud Infrastructure

Aap by: ansible

Active Directory by: hashicorp

Archive by: hashicorp

Infraestructura como Código (IaC)

The screenshot displays the Terraform Registry page for the AWS provider. The header includes the Terraform logo, a search bar, and navigation links for 'Browse', 'Publish', and 'Sign-in'. The breadcrumb trail shows 'Providers / hashicorp / aws / Version 6.28.0 / Latest Version'. The main content area features the AWS logo, the provider name 'aws', and a 'Public Cloud' badge. A description states: 'Lifecycle management of AWS resources, including EC2, Lambda, EKS, ECS, VPC, S3, RDS, DynamoDB, and more. This provider is maintained internally by the HashiCorp AWS Provider team.' Below this, there are links for 'VERSION 6.28.0', 'PUBLISHED 11 days ago', and 'SOURCE CODE hashicorp/terraform-provider-aws'. On the right side, a 'Documentation' link is highlighted with a red box, and a 'USE PROVIDER' button is visible. A table titled 'Provider Downloads' shows the following data:

Downloads	All versions
Downloads this week	36.1M
Downloads this month	78.7M
Downloads this year	78.7M
Downloads over all time	5.5B

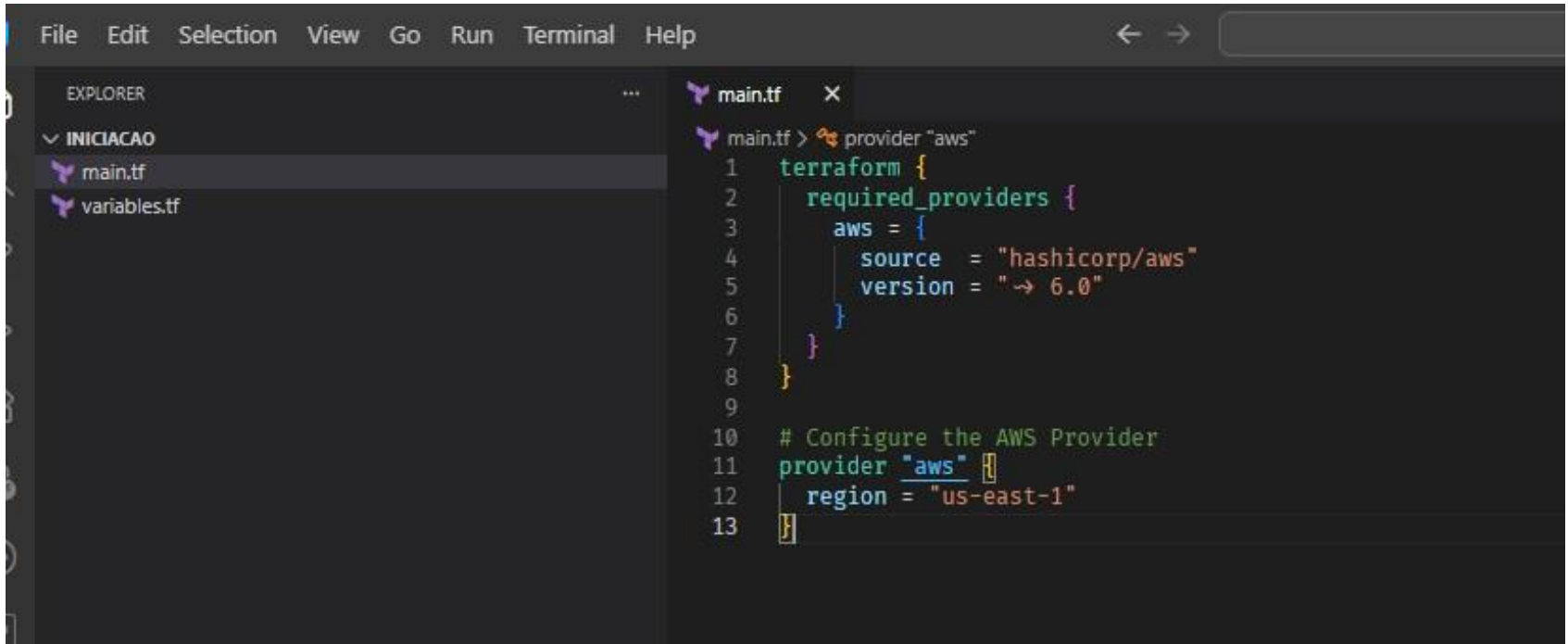
Below the table, there are 'HELPFUL LINKS' including: [Using providers](#), [Try HCP Terraform](#), [View tutorials](#), [Register for a workshop](#), [Post a forum question](#), and [Report an issue](#).

Infraestructura como Código (IaC)

```
terraform {  
  required_providers {  
    aws = {  
      source = "hashicorp/aws"  
      version = "~> 6.0"  
    }  
  }  
}  
  
# Configure the AWS Provider  
provider "aws" {  
  region = "us-east-1"  
}
```

Copy

Infraestrutura como Código (IaC)



The image shows a code editor window with a dark theme. The Explorer sidebar on the left shows a folder named 'INICIACAO' containing two files: 'main.tf' and 'variables.tf'. The main editor area displays the content of 'main.tf', which is a Terraform configuration for the AWS provider. The code is as follows:

```
main.tf > provider "aws"
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "→ 6.0"
6     }
7   }
8 }
9
10 # Configure the AWS Provider
11 provider "aws" {
12   region = "us-east-1"
13 }
```

Infraestructura como Código (IaC)

Provider Configuration

Warning:

Hard-coded credentials are not recommended in any Terraform configuration and risks secret leakage should this file ever be committed to a public version control system.

Credentials can be provided by adding an `access_key`, `secret_key`, and optionally `token`, to the `aws` provider block.

Usage:

```
provider "aws" {  
  region      = "us-west-2"  
  access_key = "my-access-key"  
  secret_key = "my-secret-key"  
}
```

Copy

Infraestructura como Código (IaC)

```
provider "aws" {  
  profile = "customprofile"  
}
```

Copy

Infraestructura como Código (IaC)

C:\Windows\system32\cmd.exe - aws configure

```
C:\Users\Futura>aws configure
AWS Access Key ID [None]: abc
AWS Secret Access Key [None]: xyz
Default region name [us-east-1]:
Default output format [json]:
```

Infraestructura como Código (IaC)

```
C:\Users\Futura>aws configure
```

```
AWS Access Key ID [None]: abc
```

```
AWS Secret Access Key [None]: xyz
```

```
Default region name [us-east-1]:
```

```
Default output format [json]:
```

```
C:\Users\Futura>aws configure --profile customprofile
```

```
AWS Access Key ID [None]: abc
```



```
AWS Secret Access Key [None]: xyz
```

```
Default region name [None]:
```

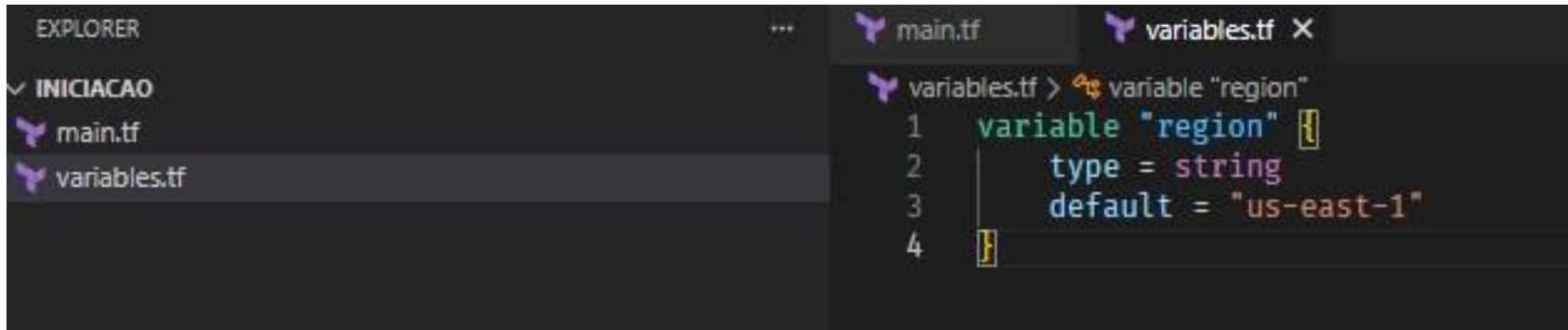
```
Default output format [None]:
```

Infraestrutura como Código (IaC)

> Disco Local (C:) > Usuários > Futura > .aws

Nome	Data de modificação	Tipo	Tamanho
 config	23/10/2025 15:06	Arquivo	1 KB
 credentials	29/12/2025 11:45	Arquivo	1 KB

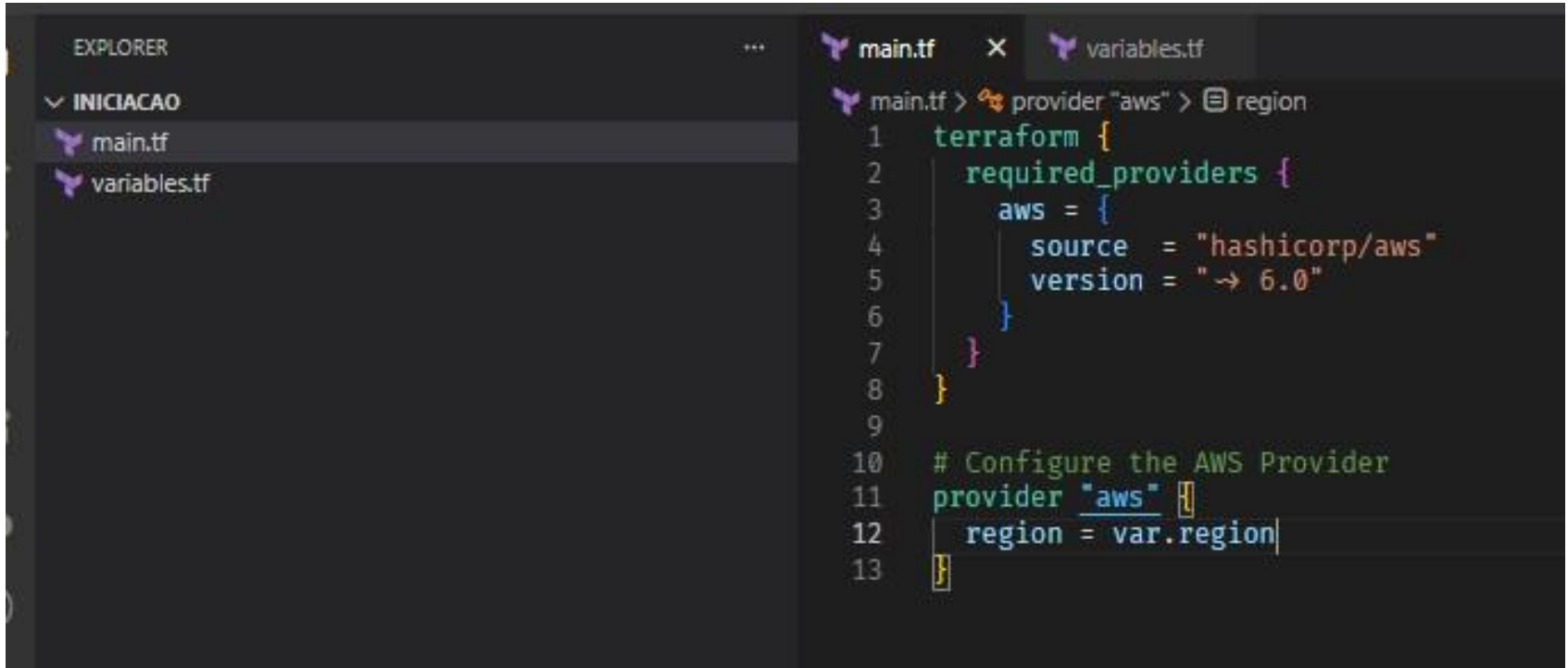
Infraestrutura como Código (IaC)



The image shows a code editor interface with a dark theme. On the left, the 'EXPLORER' sidebar is open, showing a folder named 'INICIACAO' containing two files: 'main.tf' and 'variables.tf'. The 'variables.tf' file is selected. The main editor area shows the content of 'variables.tf', which includes a variable definition for 'region'.

```
variables.tf > variable "region"  
1  variable "region" {  
2      type = string  
3      default = "us-east-1"  
4  }
```

Infraestrutura como Código (IaC)



The image shows a code editor interface with a dark theme. On the left, the 'EXPLORER' sidebar shows a folder named 'INICIACAO' containing two files: 'main.tf' and 'variables.tf'. The main editor area displays the content of 'main.tf'. The code is as follows:

```
main.tf > provider "aws" > region
1 terraform {
2     required_providers {
3         aws = {
4             source = "hashicorp/aws"
5             version = "~> 6.0"
6         }
7     }
8 }
9
10 # Configure the AWS Provider
11 provider "aws" {
12     region = var.region
13 }
```

Infraestructura como Código (IaC)

The screenshot shows the Terraform Registry website. At the top, there is a navigation bar with the HashiCorp Terraform logo, a search bar containing 'account', and links for 'Browse', 'Publish', and 'Sign-in'. Below the navigation bar, the breadcrumb path is 'Providers / hashicorp / aws / Version 6.28.0', with a 'Latest Version' button. The main content area is titled 'aws' and includes a search bar with 'vpc' entered, showing 169 matching results. The 'aws_vpc' resource is selected and highlighted. The main content area displays the 'Resource: aws_vpc' section, which includes a description: 'Provides a VPC resource.' Below this is the 'Example Usage' section, which shows two code snippets. The first is 'Basic usage:' with a code block:

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

 and a 'Copy' button. The second is 'Basic usage with tags:' with a code block:

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

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

 and a 'Copy' button. On the right side, there is a 'ON THIS PAGE' section with links for 'Example Usage', 'Argument Reference', 'Attribute Reference', and 'Import', along with a 'Report an issue' link.

HashiCorp Terraform Registry | account

Browse | Publish | Sign-in

Providers / hashicorp / aws / Version 6.28.0 | Latest Version

aws | Overview | Documentation | USE PROVIDER

AWS DOCUMENTATION

vpc

169 matching results

- aws_route
- aws_route_table
- aws_route_table_association
- aws_security_group
- aws_security_group_rule
- aws_subnet
- aws_vpc**
- aws_vpc_block_public_access_exclusion
- aws_vpc_block_public_access_options
- aws_vpc_dhcp_options
- aws_vpc_dhcp_options_association
- aws_vpc_encryption_control
- aws_vpc_endpoint
- aws_vpc_endpoint_connection_accepter

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"
  }
}
```

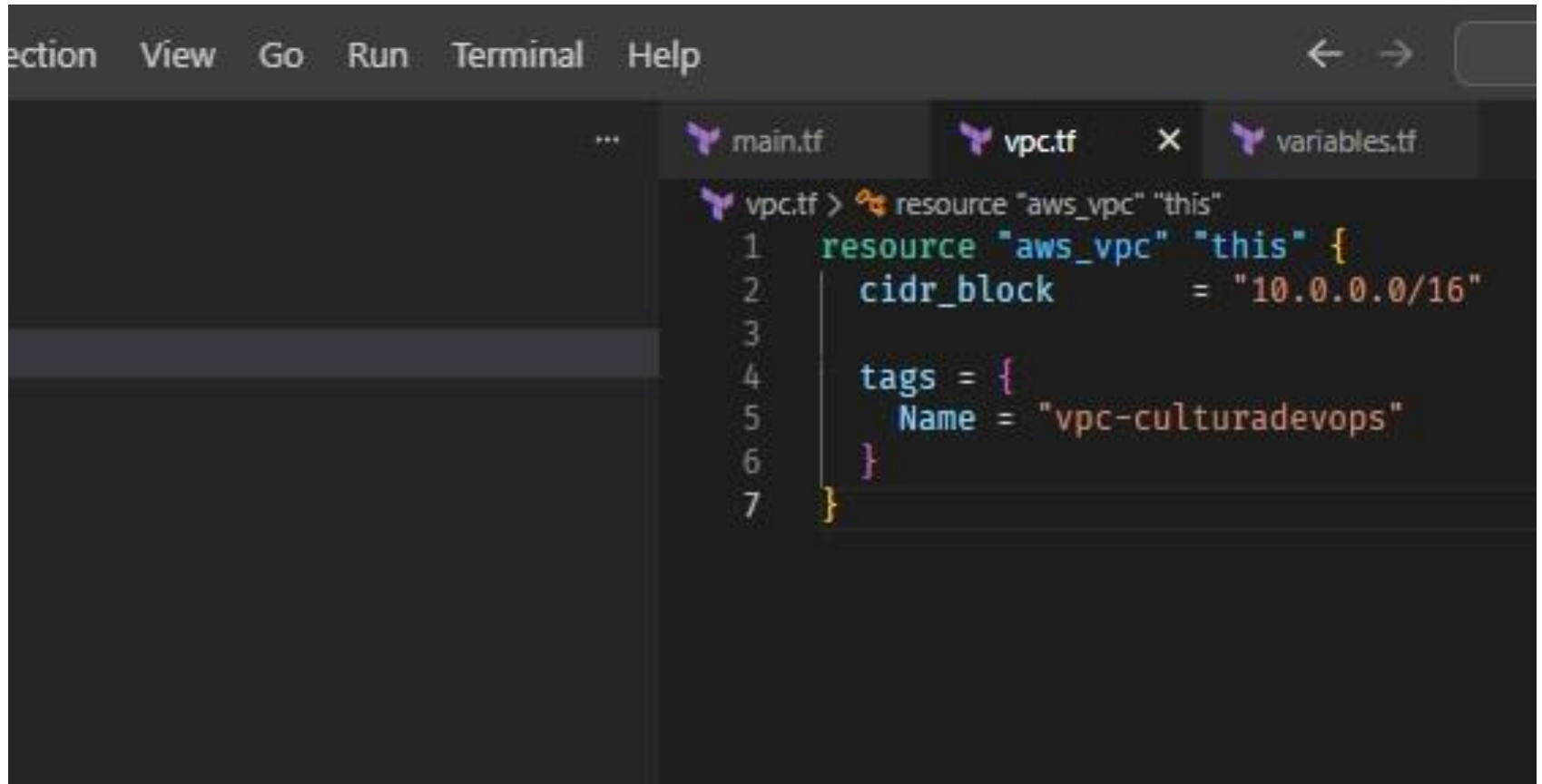
Copy

ON THIS PAGE

- Example Usage
- Argument Reference
- Attribute Reference
- Import

Report an issue

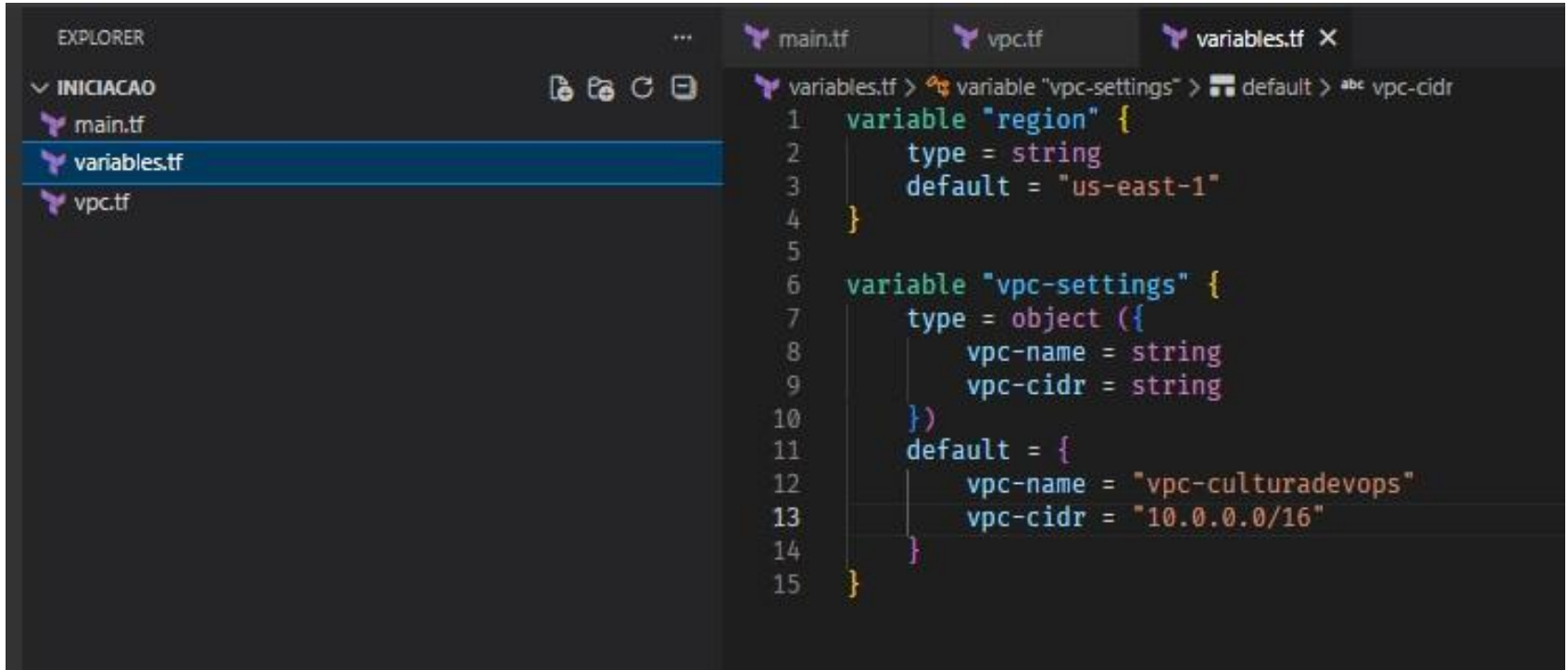
Infraestructura como Código (IaC)



The image shows a screenshot of a code editor with a dark theme. The top menu bar includes 'Action', 'View', 'Go', 'Run', 'Terminal', and 'Help'. There are navigation arrows and a search box on the right. The editor has three tabs: 'main.tf', 'vpc.tf', and 'variables.tf'. The 'vpc.tf' tab is active and shows the following Terraform code:

```
vpc.tf > resource "aws_vpc" "this"
1  resource "aws_vpc" "this" {
2      cidr_block      = "10.0.0.0/16"
3
4      tags = {
5          Name = "vpc-culturadevops"
6      }
7  }
```

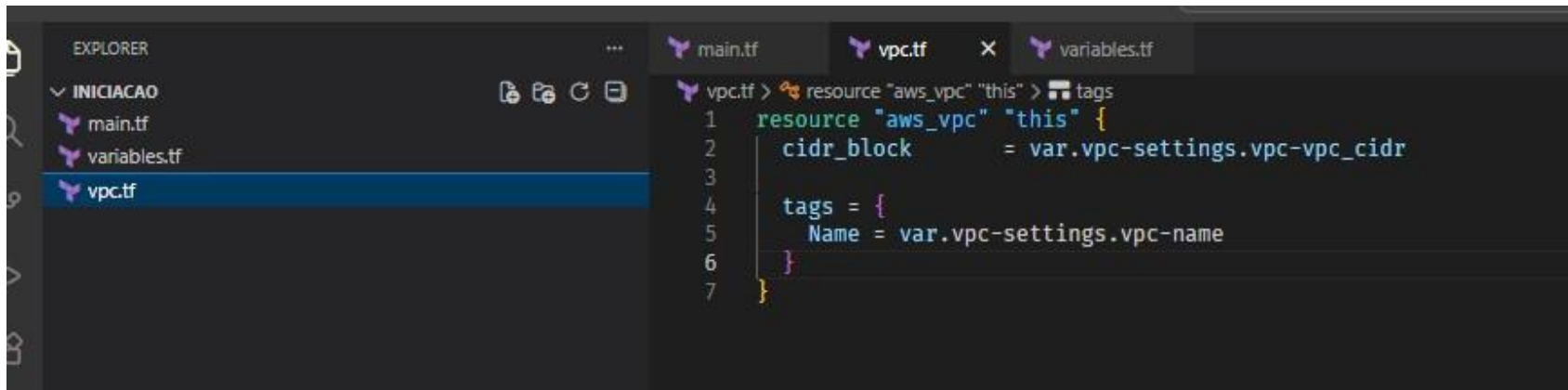
Infraestrutura como Código (IaC)



The image shows a code editor interface with a dark theme. On the left, the 'EXPLORER' sidebar shows a folder named 'INICIACAO' containing three files: 'main.tf', 'variables.tf', and 'vpc.tf'. The 'variables.tf' file is selected and highlighted in blue. The main editor area shows the content of 'variables.tf' with the following code:

```
1  variable "region" {
2      type = string
3      default = "us-east-1"
4  }
5
6  variable "vpc-settings" {
7      type = object ({
8          vpc-name = string
9          vpc-cidr = string
10     })
11     default = {
12         vpc-name = "vpc-culturadevops"
13         vpc-cidr = "10.0.0.0/16"
14     }
15 }
```

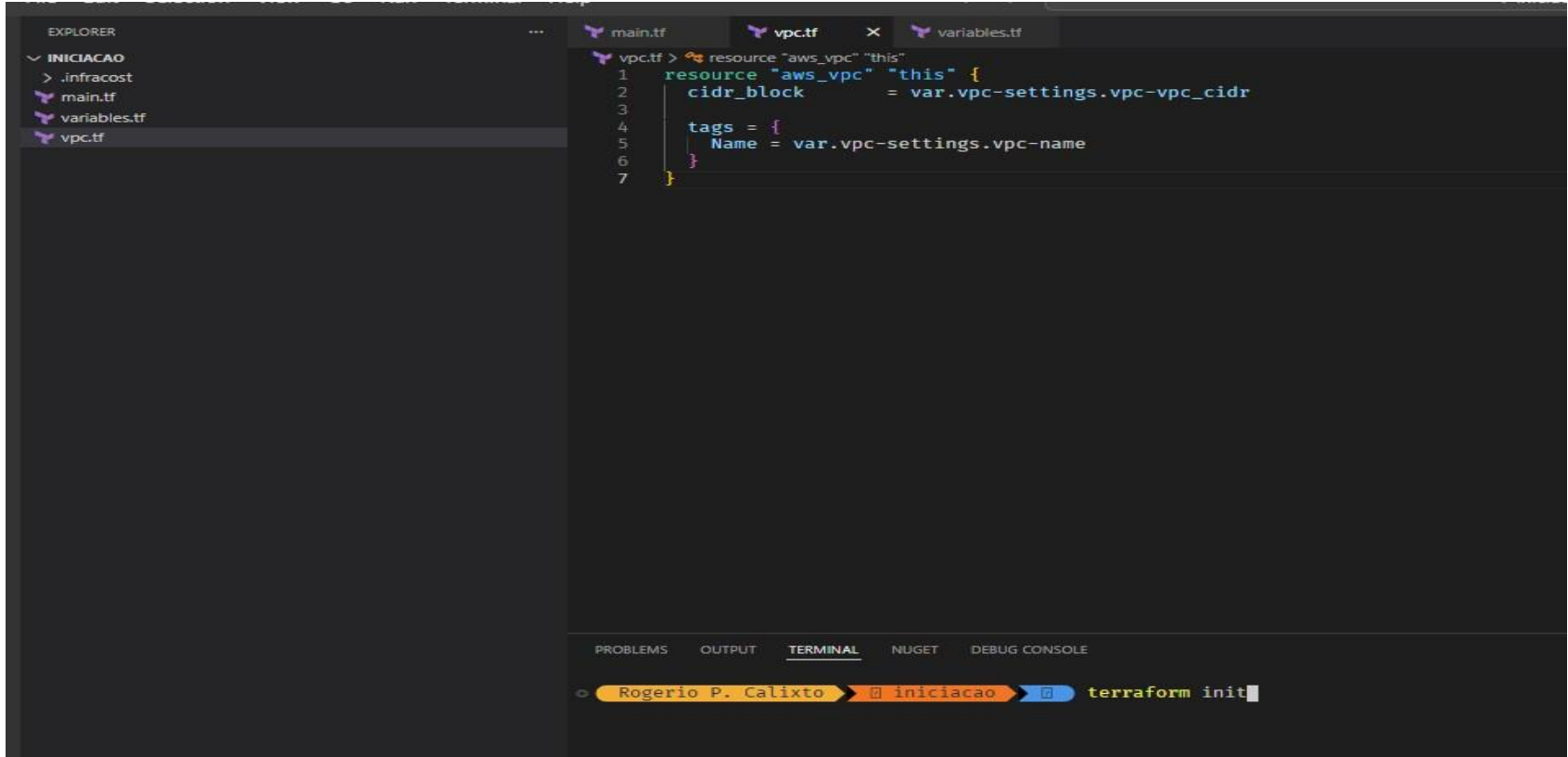
Infraestructura como Código (IaC)



The image shows a code editor interface with a dark theme. On the left, the 'EXPLORER' sidebar shows a folder named 'INICIACAO' containing three files: 'main.tf', 'variables.tf', and 'vpc.tf'. The 'vpc.tf' file is selected and highlighted in blue. The main editor area displays the contents of 'vpc.tf', which is a Terraform configuration for an AWS VPC resource. The code is as follows:

```
vpc.tf > resource "aws_vpc" "this" > tags
1 resource "aws_vpc" "this" {
2     cidr_block      = var.vpc-settings.vpc-vpc_cidr
3
4     tags = {
5         Name = var.vpc-settings.vpc-name
6     }
7 }
```

Infraestrutura como Código (IaC)



The image shows a screenshot of the Visual Studio Code editor interface. On the left, the Explorer sidebar shows a project structure with folders and files: INICIACAO, .infracost, main.tf, variables.tf, and vpc.tf. The main editor area displays the content of vpc.tf, which contains Terraform code for creating an AWS VPC. The code is as follows:

```
vpc.tf > resource "aws_vpc" "this"
1 resource "aws_vpc" "this" {
2   cidr_block      = var.vpc-settings.vpc-vpc_cidr
3
4   tags = {
5     Name = var.vpc-settings.vpc-name
6   }
7 }
```

At the bottom of the editor, the Terminal panel is active, showing a shell prompt with the user name 'Rogerio P. Calixto' and the current directory 'iniciacao'. The command 'terraform init' is entered and ready to be executed.

Infraestrutura como Código (IaC)

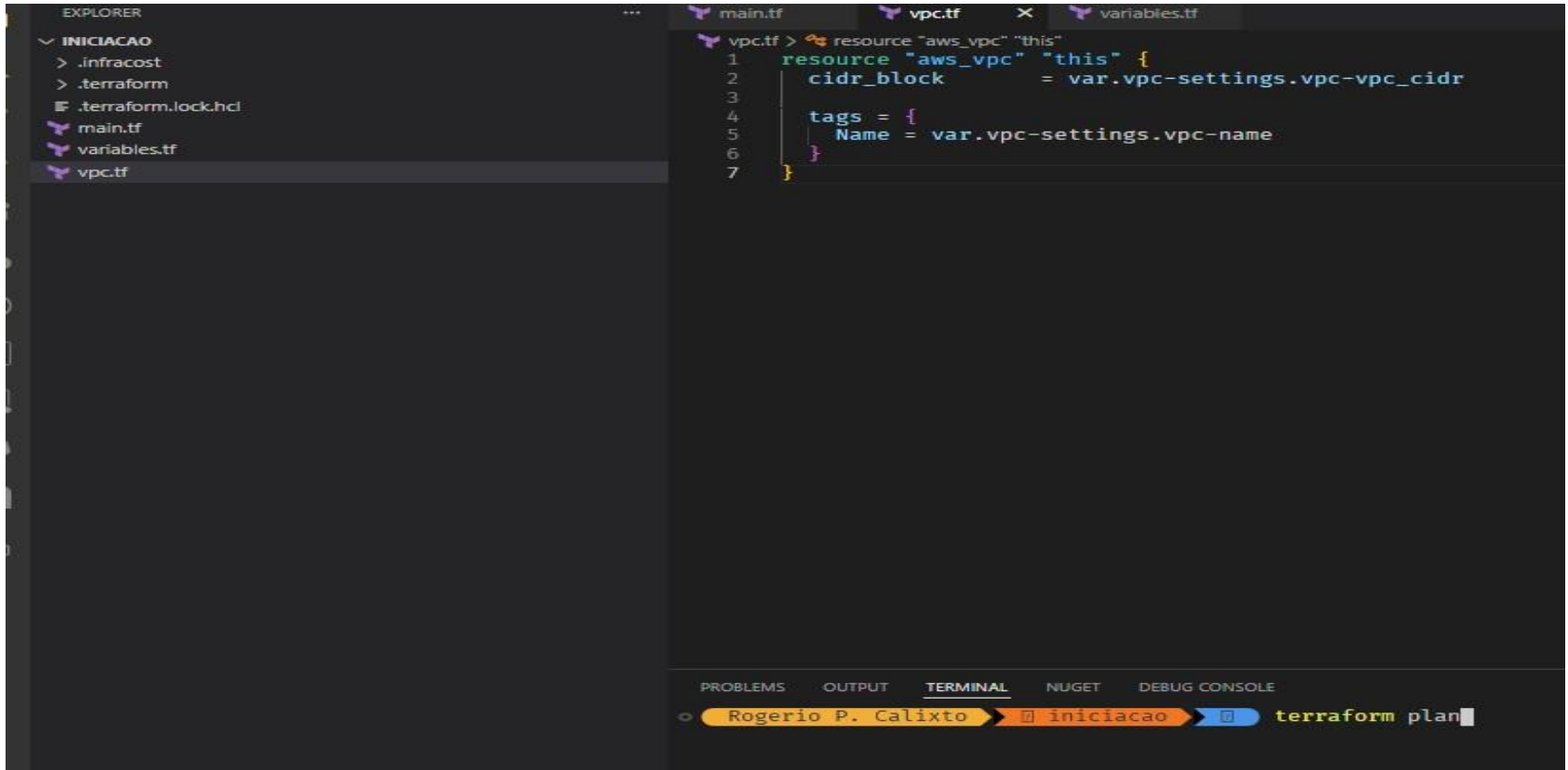
```
iniciacao terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 6.0"...
- Installing hashicorp/aws v6.28.0...
- Installed hashicorp/aws v6.28.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record 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.
Rogerio P. Calixto iniciacao
```

Infraestrutura como Código (IaC)



The image shows a code editor interface with a dark theme. On the left, the Explorer sidebar shows a project structure under the name 'INICIACAO'. The files listed are: '.infracost', '.terraform', '.terraform.lock.hcl', 'main.tf', 'variables.tf', and 'vpc.tf'. The 'vpc.tf' file is selected and its content is displayed in the main editor area. The code defines an AWS VPC resource named 'this' with a 'cidr_block' and a 'tags' block containing a 'Name' tag. The bottom of the editor shows a terminal window with the command 'terraform plan' being executed in a shell session for the user 'Rogerio P. Calixto' in the 'iniciacao' directory.

```
resource "aws_vpc" "this" {
  cidr_block = var.vpc-settings.vpc-vpc_cidr

  tags = {
    Name = var.vpc-settings.vpc-name
  }
}
```

PROBLEMS OUTPUT **TERMINAL** NUGET DEBUG CONSOLE

Rogerio P. Calixto iniciacao terraform plan

Infraestrutura como Código (IaC)

iniciacao terraform plan

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

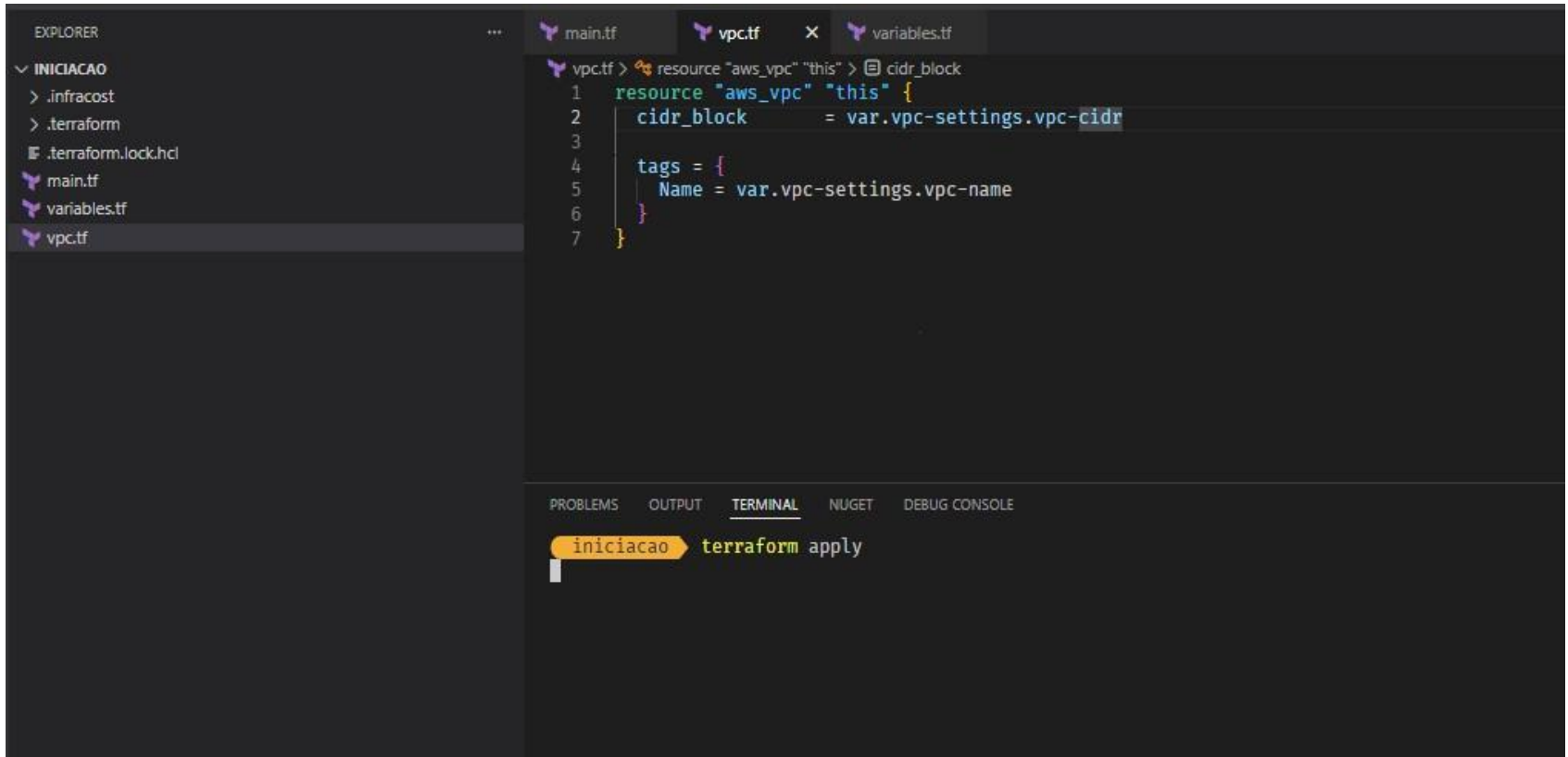
Terraform will perform the following actions:

```
# aws_vpc.this will be created
+ resource "aws_vpc" "this" {
  + arn                = (known after apply)
  + cidr_block         = "10.0.0.0/16"
  + 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_dns_hostnames = (known after apply)
  + enable_dns_support   = true
  + enable_network_address_usage_metrics = (known after apply)
  + id                  = (known after apply)
  + instance_tenancy    = "default"
  + ipv6_association_id = (known after apply)
  + ipv6_cidr_block     = (known after apply)
  + ipv6_cidr_block_network_border_group = (known after apply)
  + main_route_table_id = (known after apply)
  + owner_id            = (known after apply)
  + region              = "us-east-1"
  + tags                = {
    + "Name" = "vpc-culturadevops"
  }
  + tags_all           = {
    + "Name" = "vpc-culturadevops"
  }
}
```

Plan: 1 to add, 0 to change, 0 to destroy.

Note: You didn't use the `-out` option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

Infraestrutura como Código (IaC)



The image shows a screenshot of the Visual Studio Code editor interface. On the left, the Explorer sidebar shows a project structure under the name 'INICIACAO'. The files listed are: .infracost, .terraform, .terraform.lock.hcl, main.tf, variables.tf, and vpc.tf. The vpc.tf file is currently selected and open in the main editor window. The code in vpc.tf defines an AWS VPC resource named 'this' with a cidr_block attribute set to var.vpc-settings.vpc-cidr and a tags block containing Name = var.vpc-settings.vpc-name. The terminal at the bottom shows the command terraform apply being executed in the 'iniciacao' directory.

```
EXPLORER
INICIACAO
  > .infracost
  > .terraform
  .terraform.lock.hcl
  main.tf
  variables.tf
  vpc.tf

vpc.tf
resource "aws_vpc" "this" > cidr_block
1 resource "aws_vpc" "this" {
2   cidr_block = var.vpc-settings.vpc-cidr
3
4   tags = {
5     Name = var.vpc-settings.vpc-name
6   }
7 }
```

PROBLEMS OUTPUT TERMINAL NUGET DEBUG CONSOLE

```
iniciacao terraform apply
```

Infraestructura como Código (IaC)

Terraform will perform the following actions:

```
# aws_vpc.this will be created
+ resource "aws_vpc" "this" {
  + arn = (known after apply)
  + cidr_block = "10.0.0.0/16"
  + 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_dns_hostnames = (known after apply)
  + enable_dns_support = true
  + enable_network_address_usage_metrics = (known after apply)
  + id = (known after apply)
  + instance_tenancy = "default"
  + ipv6_association_id = (known after apply)
  + ipv6_cidr_block = (known after apply)
  + ipv6_cidr_block_network_border_group = (known after apply)
  + main_route_table_id = (known after apply)
  + owner_id = (known after apply)
  + region = "us-east-1"
  + tags = {
    + "Name" = "vpc-culturadevops"
  }
  + tags_all = {
    + "Name" = "vpc-culturadevops"
  }
}
```

Plan: 1 to add, 0 to change, 0 to destroy.

Infraestructura como Código (IaC)

```
...
+ arn = (known after apply)
+ cidr_block = "10.0.0.0/16"
+ 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_dns_hostnames = (known after apply)
+ enable_dns_support = true
+ enable_network_address_usage_metrics = (known after apply)
+ id = (known after apply)
+ instance_tenancy = "default"
+ ipv6_association_id = (known after apply)
+ ipv6_cidr_block = (known after apply)
+ ipv6_cidr_block_network_border_group = (known after apply)
+ main_route_table_id = (known after apply)
+ owner_id = (known after apply)
+ region = "us-east-1"
+ tags = {
  + "Name" = "vpc-culturadevops"
}
+ tags_all = {
  + "Name" = "vpc-culturadevops"
}
}
```

Plan: 1 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

Infraestrutura como Código (IaC)

```
PROBLEMS  OUTPUT  TERMINAL  NUGET  DEBUG CONSOLE
iniciacao terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create
...
+ arn = (known after apply)
+ cidr_block = "10.0.0.0/16"
+ 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_dns_hostnames = (known after apply)
+ enable_dns_support = true
+ enable_network_address_usage_metrics = (known after apply)
+ id = (known after apply)
+ instance_tenancy = "default"
+ ipv6_association_id = (known after apply)
+ ipv6_cidr_block = (known after apply)
+ ipv6_cidr_block_network_border_group = (known after apply)
+ main_route_table_id = (known after apply)
+ owner_id = (known after apply)
+ region = "us-east-1"
+ tags = {
+   + "Name" = "vpc-culturadevops"
}
+ tags_all = {
+   + "Name" = "vpc-culturadevops"
}
}

Plan: 1 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_vpc.this: Creating...
aws_vpc.this: Creation complete after 4s [id=vpc-0fd13af36f2baaf56]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
o Rogerio P. Calixto > iniciacao >
```

Infraestructura como Código (IaC)

Your VPCs

VPCs

VPC encryption controls

Your VPCs (1/1) [info](#)

Last updated
less than a minute ago

Actions

Create VPC

Find VPCs by attribute or tag

culturadevops

Clear filters

< 1 >

<input checked="" type="checkbox"/>	Name	VPC ID	State	Encryption c...	Encryption control ...	Block Public...	IPv4 CIDR	IPv6 CIDR
<input checked="" type="checkbox"/>	vpc-culturadevops	vpc-0fd13af36f2baaf56	Available	-	-	Off	10.0.0.0/16	-

vpc-0fd13af36f2baaf56 / vpc-culturadevops

Details

Resource map

CIDRs

Flow logs

Tags

Integrations

Details

VPC ID

[vpc-0fd13af36f2baaf56](#)

DNS resolution

Enabled

Main network ACL

[acl-05ad2a546e2d5d5b3](#)

IPv6 CIDR (Network border group)

-

State

Available

Tenancy

default

Default VPC

No

Network Address Usage metrics

Disabled

Block Public Access

Off

DHCP option set

[dopt-2722355d](#)

IPv4 CIDR

10.0.0.0/16

Route 53 Resolver DNS Firewall rule groups

-

DNS hostnames

Disabled

Main route table

[rtb-0a667096c46dde164](#)

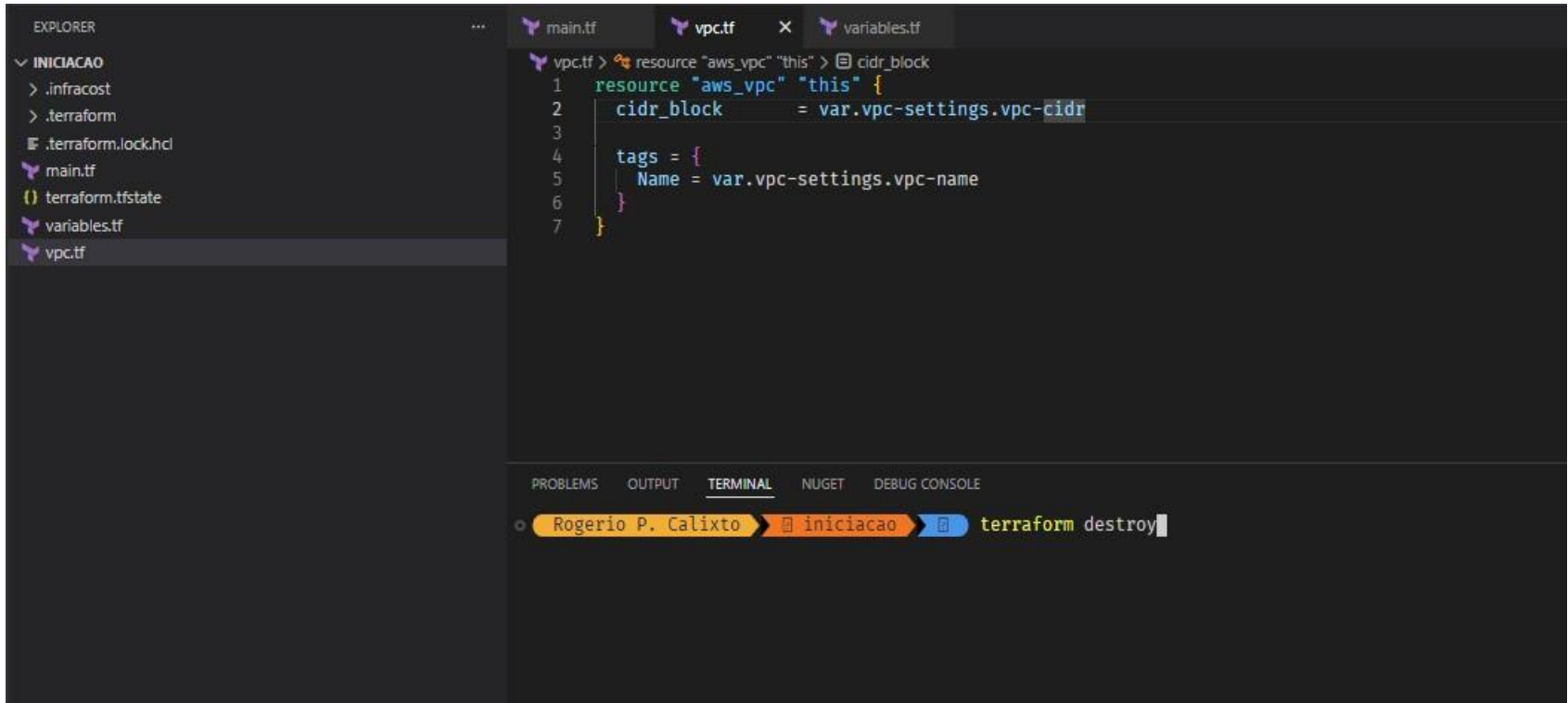
IPv6 pool

-

Owner ID

[791659331772](#)

Infraestrutura como Código (IaC)



The image shows a screenshot of the Visual Studio Code editor interface. On the left, the Explorer sidebar shows a project structure under the name 'INICIACAO'. The files listed are: `./infracost`, `./terraform`, `./terraform.lock.hcl`, `main.tf`, `terraform.tfstate`, `variables.tf`, and `vpc.tf`. The `vpc.tf` file is selected and open in the main editor. The code in `vpc.tf` is as follows:

```
resource "aws_vpc" "this" > cidr_block
1 resource "aws_vpc" "this" {
2   cidr_block      = var.vpc-settings.vpc-cidr
3
4   tags = {
5     Name = var.vpc-settings.vpc-name
6   }
7 }
```

At the bottom of the editor, the Terminal panel is active, showing the command `terraform destroy` being executed. The terminal output shows the name of the user, 'Rogerio P. Calixto', and the command being run.

Infraestrutura como Código (IaC)

```
PROBLEMS OUTPUT TERMINAL NUGET DEBUG CONSOLE
iniciacao terraform destroy
aws_vpc.this: Refreshing state... [id=vpc-0fd13af36f2baaf56]

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

Terraform will perform the following actions:

# aws_vpc.this will be destroyed
- resource "aws_vpc" "this" {
  - arn = "arn:aws:ec2:us-east-1:791659331772:vpc/vpc-0fd13af36f2baaf56" -> null
  - assign_generated_ipv6_cidr_block = false -> null
  - cidr_block = "10.0.0.0/16" -> null
  - default_network_acl_id = "acl-05ad2a546e2d5d5b3" -> null
  - default_route_table_id = "rtb-0a667096c46dde164" -> null
  - default_security_group_id = "sg-0b1dc0b492d8dd4d2" -> null
  - dhcp_options_id = "dopt-2722355d" -> null
  - enable_dns_hostnames = false -> null
  - enable_dns_support = true -> null
  - enable_network_address_usage_metrics = false -> null
  - id = "vpc-0fd13af36f2baaf56" -> null
  - instance_tenancy = "default" -> null
  - ipv6_netmask_length = 0 -> null
  - main_route_table_id = "rtb-0a667096c46dde164" -> null
  - owner_id = "791659331772" -> null
  - region = "us-east-1" -> null
  - tags = {
    - "Name" = "vpc-culturadevops"
  } -> null
  - tags_all = {
    - "Name" = "vpc-culturadevops"
  } -> null
  # (4 unchanged attributes hidden)
}

Plan: 0 to add, 0 to change, 1 to destroy.

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes
```

Infraestrutura como Código (IaC)

```
PROBLEMS OUTPUT TERMINAL NUGET DEBUG CONSOLE
iniciacao terraform destroy
aws_vpc.this: Refreshing state... [id=vpc-0fd13af36f2baaf56]

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

- destroy ...
  - arn = "arn:aws:ec2:us-east-1:791659331772:vpc/vpc-0fd13af36f2baaf56" -> null
  - assign_generated_ipv6_cidr_block = false -> null
  - cidr_block = "10.0.0.0/16" -> null
  - default_network_acl_id = "acl-05ad2a546e2d5d5b3" -> null
  - default_route_table_id = "rtb-0a667096c46dde164" -> null
  - default_security_group_id = "sg-0b1dc0b492d8dd4d2" -> null
  - dhcp_options_id = "dopt-2722355d" -> null
  - enable_dns_hostnames = false -> null
  - enable_dns_support = true -> null
  - enable_network_address_usage_metrics = false -> null
  - id = "vpc-0fd13af36f2baaf56" -> null
  - instance_tenancy = "default" -> null
  - ipv6_netmask_length = 0 -> null
  - main_route_table_id = "rtb-0a667096c46dde164" -> null
  - owner_id = "791659331772" -> null
  - region = "us-east-1" -> null
  - tags = {
    - "Name" = "vpc-culturadevops"
  } -> null
  - tags_all = {
    - "Name" = "vpc-culturadevops"
  } -> null
  # (4 unchanged attributes hidden)
}

Plan: 0 to add, 0 to change, 1 to destroy.

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes
aws_vpc.this: Destroying... [id=vpc-0fd13af36f2baaf56]
aws_vpc.this: Destruction complete after 2s

Destroy complete! Resources: 1 destroyed.
Rogerio P. Calixto iniciacao
```

Infraestructura como Código (IaC)

```
variable "vpc-settings" {  
  type = object ({  
    vpc-name = string  
    vpc-cidr = string  
    subnet_pvt_config = list(object({  
      name = string  
      cidr_block = string  
      available_zone = string  
      map_public_ip_on_launch = bool  
    }))  
    subnet_pub_config = list(object({  
      name = string  
      cidr_block = string  
      available_zone = string  
      map_public_ip_on_launch = bool  
    }))  
  })  
}
```

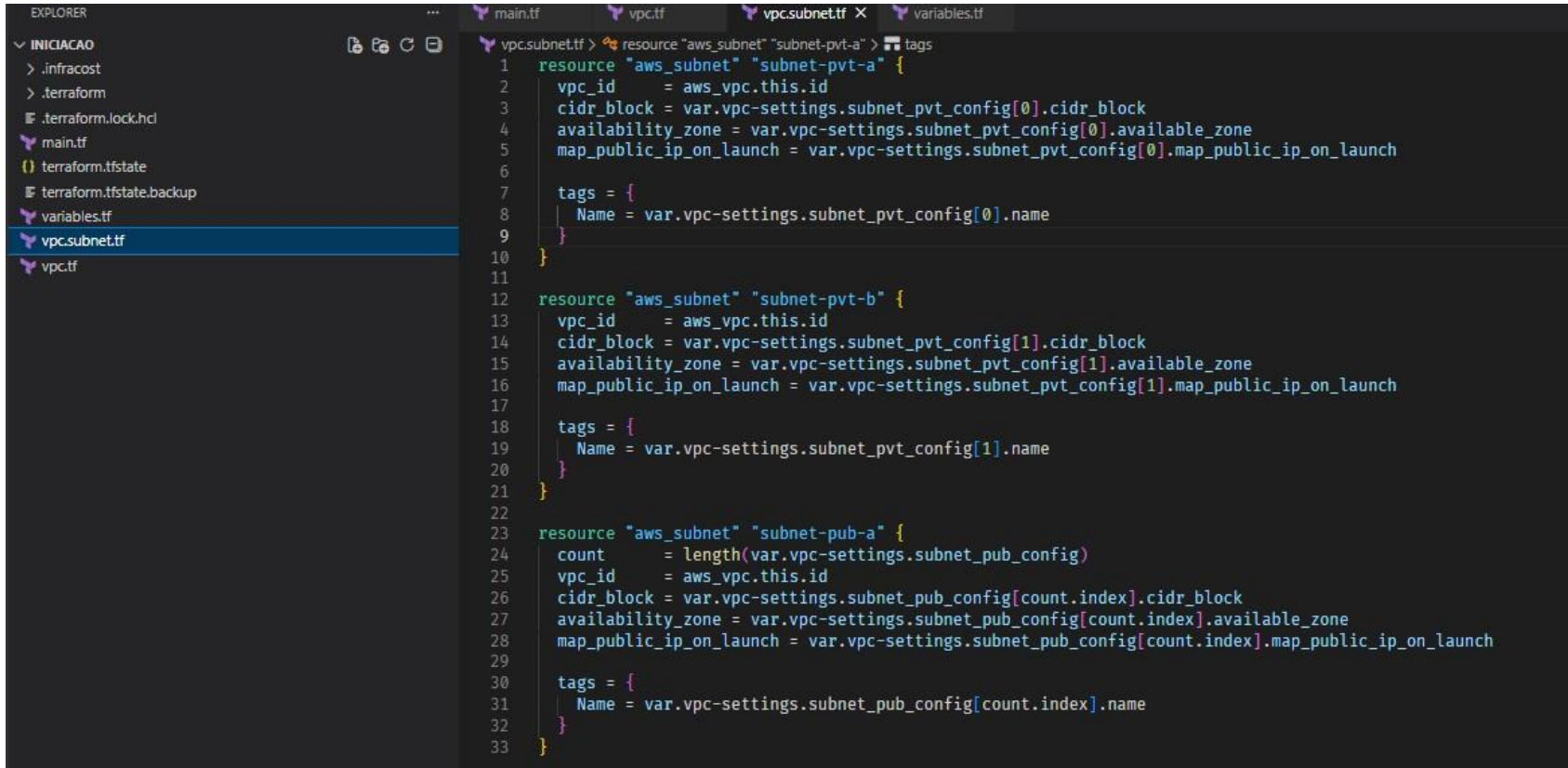
Infraestructura como Código (IaC)

```
default = {  
  vpc-name = "vpc-culturadevops"  
  vpc-cidr = "10.0.0.0/24"  
  subnet_pvt_config = [  
    {  
      name = "subnet-prv-a"  
      cidr_block = "10.0.0.0/27"  
      available_zone = "us-east-1a"  
      map_public_ip_on_launch = false  
    },  
    {  
      name = "subnet-prv-b"  
      cidr_block = "10.0.0.32/27"  
      available_zone = "us-east-1b"  
      map_public_ip_on_launch = false  
    }  
  ]  
}
```

Infraestructura como Código (IaC)

```
]
subnet_pub_config = [
  {
    name = "subnet-pub-a"
    cidr_block = "10.0.0.64/27"
    available_zone = "us-east-1a"
    map_public_ip_on_launch = true
  },
  {
    name = "subnet-pub-b"
    cidr_block = "10.0.0.96/27"
    available_zone = "us-east-1b"
    map_public_ip_on_launch = true
  }
]
```

Infraestrutura como Código (IaC)



The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like .infra, .terraform, and files like main.tf, terraform.tfstate, terraform.tfstate.backup, variables.tf, vpc.subnet.tf, and vpc.tf. The code editor shows the content of vpc.subnet.tf, which defines three AWS subnets: subnet-pvt-a, subnet-pvt-b, and subnet-pub-a. Each subnet resource is defined with its vpc_id, cidr_block, availability_zone, map_public_ip_on_launch, and tags.

```
1 resource "aws_subnet" "subnet-pvt-a" {
2     vpc_id      = aws_vpc.this.id
3     cidr_block  = var.vpc-settings.subnet_pvt_config[0].cidr_block
4     availability_zone = var.vpc-settings.subnet_pvt_config[0].available_zone
5     map_public_ip_on_launch = var.vpc-settings.subnet_pvt_config[0].map_public_ip_on_launch
6
7     tags = {
8         Name = var.vpc-settings.subnet_pvt_config[0].name
9     }
10 }
11
12 resource "aws_subnet" "subnet-pvt-b" {
13     vpc_id      = aws_vpc.this.id
14     cidr_block  = var.vpc-settings.subnet_pvt_config[1].cidr_block
15     availability_zone = var.vpc-settings.subnet_pvt_config[1].available_zone
16     map_public_ip_on_launch = var.vpc-settings.subnet_pvt_config[1].map_public_ip_on_launch
17
18     tags = {
19         Name = var.vpc-settings.subnet_pvt_config[1].name
20     }
21 }
22
23 resource "aws_subnet" "subnet-pub-a" {
24     count      = length(var.vpc-settings.subnet_pub_config)
25     vpc_id     = aws_vpc.this.id
26     cidr_block = var.vpc-settings.subnet_pub_config[count.index].cidr_block
27     availability_zone = var.vpc-settings.subnet_pub_config[count.index].available_zone
28     map_public_ip_on_launch = var.vpc-settings.subnet_pub_config[count.index].map_public_ip_on_launch
29
30     tags = {
31         Name = var.vpc-settings.subnet_pub_config[count.index].name
32     }
33 }
```

Infraestructura como Código (IaC)

```
# aws_subnet.subnet-pvt-a will be created
+ resource "aws_subnet" "subnet-pvt-a" {
+   arn                                = (known after apply)
+   assign_ipv6_address_on_creation    = false
+   availability_zone                  = "us-east-1a"
+   availability_zone_id                = (known after apply)
+   cidr_block                          = "10.0.0.0/27"
+   enable_dns64                        = false
+   enable_resource_name_dns_a_record_on_launch = false
+   enable_resource_name_dns_aaaa_record_on_launch = false
+   id                                  = (known after apply)
+   ipv6_cidr_block_association_id      = (known after apply)
+   ipv6_native                          = false
+   map_public_ip_on_launch             = false
+   owner_id                             = (known after apply)
+   private_dns_hostname_type_on_launch = (known after apply)
+   region                               = "us-east-1"
+   tags                                 = {
+     "Name" = "subnet-prv-a"
+   }
+   tags_all                             = {
+     "Name" = "subnet-prv-a"
+   }
+   vpc_id                               = (known after apply)
}

# aws_subnet.subnet-pvt-b will be created
+ resource "aws_subnet" "subnet-pvt-b" {
+   arn                                = (known after apply)
+   assign_ipv6_address_on_creation    = false
+   availability_zone                  = "us-east-1b"
+   availability_zone_id                = (known after apply)
+   cidr_block                          = "10.0.0.32/27"
+   enable_dns64                        = false
+   enable_resource_name_dns_a_record_on_launch = false
+   enable_resource_name_dns_aaaa_record_on_launch = false
+   id                                  = (known after apply)
+   ipv6_cidr_block_association_id      = (known after apply)
+   ipv6_native                          = false
+   map_public_ip_on_launch             = false
+   owner_id                             = (known after apply)
+   private_dns_hostname_type_on_launch = (known after apply)
+   region                               = "us-east-1"
+   tags                                 = {
+     "Name" = "subnet-prv-b"
+   }
+   tags_all                             = {
+     "Name" = "subnet-prv-b"
+   }
+   vpc_id                               = (known after apply)
}
```


Infraestrutura como Código (IaC)

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_vpc.this: Creating...

aws_vpc.this: Creation complete after 4s [id=vpc-018345c1a19af160c]

aws_subnet.subnet-pvt-b: Creating...

aws_subnet.subnet-pub-a[1]: Creating...

aws_subnet.subnet-pub-a[0]: Creating...

aws_subnet.subnet-pvt-a: Creating...

aws_subnet.subnet-pvt-b: Creation complete after 1s [id=subnet-04cb7d9281565ce19]

aws_subnet.subnet-pvt-a: Creation complete after 1s [id=subnet-08ace1fe93bf16723]

aws_subnet.subnet-pub-a[1]: Still creating... [00m10s elapsed]

aws_subnet.subnet-pub-a[0]: Still creating... [00m10s elapsed]


aws_subnet.subnet-pub-a[1]: Creation complete after 11s [id=subnet-0bafdb03b4c854c6b]

aws_subnet.subnet-pub-a[0]: Creation complete after 12s [id=subnet-0abf0e1a5e82b575d]


Apply complete! Resources: 5 added, 0 changed, 0 destroyed.


Infraestructura como Código (IaC)

Subnets (4) [Info](#)

Last updated 4 minutes ago  [Actions](#) [Create subnet](#)

Find subnets by attribute or tag

vpc-culturadevops  [Clear filters](#)

< 1 > 

<input type="checkbox"/>	Name ▲	Subnet ID ▼	VPC ▼	IPv4 CIDR ▼	Availability Zone ▼	Auto-assign public IPv4 address
<input type="checkbox"/>	subnet-prv-a	subnet-08ace1fe93bf16723	vpc-018345c1a19af160c vpc-culturadevops	10.0.0/27	use1-az6 (us-east-1a)	No
<input type="checkbox"/>	subnet-prv-b	subnet-04cb7d9281565ce19	vpc-018345c1a19af160c vpc-culturadevops	10.0.32/27	use1-az1 (us-east-1b)	No
<input type="checkbox"/>	subnet-pub-a	subnet-0abf0e1a5e82b575d	vpc-018345c1a19af160c vpc-culturadevops	10.0.64/27	use1-az6 (us-east-1a)	Yes
<input type="checkbox"/>	subnet-pub-b	subnet-0bafdb03b4c854c6b	vpc-018345c1a19af160c vpc-culturadevops	10.0.96/27	use1-az1 (us-east-1b)	Yes

Infraestrutura como Código (IaC)

```
You, 13 months ago | 2 authors (logand and one other)
module "vpc" {
  source           = "git::https://github.com/meu-repositorio/template-network.git"
  profile          = local.profile
  arn_assumerole   = var.arn_assumerole
  regioao         = var.regiao
  projeto         = local.projeto
  ambiente        = var.ambiente
  tipo_criacao    = local.tipo_criacao
  cidr_block      = local.cidr_block
  subnet_privada_config = local.subnet_privada_config
  subnet_publica_config = local.subnet_publica_config
}
```

```
You, 13 months ago | 2 authors (You and one other)
module "servidor" {
  source           = "git::https://github.com/meu-repositorio/template-instance.git"
  regioao         = var.regiao
  projeto         = local.projeto
  ambiente        = var.ambiente
  tipo_criacao    = local.tipo_criacao
  ami             = lookup(var.amis, var.regiao)
  instance_type   = var.instance_type
  keypair_name    = var.keypair_projetobillingtags
  vpc_id          = module.vpc.vpc_id
  subnet_id       = module.vpc.subnet_publica_ids[0]
  instance_name   = "Servidor 1"
  disable_api_termination = true
  root_block_device_size = var.root-block-device-size
  associate_public_ip = true
}
You, 13 months ago • Uncommitted changes
```

Infraestrutura como Código (IaC)

