Referência Rápida do Terraform

Providers, recursos, variáveis, estado, módulos

Básico

Fluxo de Trabalho Principal

```
terraform init      # instalar providers e módulos
terraform plan      # pré-visualizar alterações
terraform apply     # aplicar alterações
terraform destroy   # destruir todos os recursos
```

Comandos Essenciais

`terraform init`	Inicializar diretório de trabalho, baixar providers
`terraform plan`	Exibir plano de execução sem aplicar
`terraform apply`	Aplicar alterações à infraestrutura
`terraform destroy`	Destruir todos os recursos gerenciados
`terraform fmt`	Formatar arquivos **`.tf`** no estilo canônico
`terraform validate`	Verificar sintaxe da configuração
`terraform show`	Exibir estado atual ou plano
`terraform output`	Imprimir valores de saída

Providers

Configuração de Provider

```
terraform {
  required_providers {
    aws = { source = "hashicorp/aws", version = "~> 5.0" }
  }
}
provider "aws" {
  region = "us-east-1"
}
```

Notas sobre Provider

`source`	Endereço do registro (**`hashicorp/aws`**, **`hashicorp/google`**)
`version`	Restrição de versão (**`~> 5.0`**, **`>= 3.0, < 4.0`**)
`.terraform.lock.hcl`	Arquivo de lock — confirmar no controle de versão
`alias`	Usar múltiplas configs para o mesmo provider

Recursos

Blocos de Recurso

```
resource "aws_instance" "web" {
  ami           = "ami-0c55b159cbfafe1f0"
  instance_type = "t3.micro"
  tags = { Name = "web-server" }
}
```

Meta-Argumentos de Recurso

`depends_on`	Dependência explícita de outro recurso
`count`	Criar múltiplas instâncias (**`count = 3`**)
`for_each`	Criar instâncias a partir de um mapa ou conjunto
`provider`	Selecionar um alias de provider não padrão
`lifecycle`	Personalizar comportamento de criação/atualização/destruição

Referenciando Recursos

```
# tipo.nome.atributo
aws_instance.web.id
aws_instance.web.public_ip
aws_vpc.main.cidr_block
```

Variáveis

Declarando Variáveis

```
variable "region" {
  type    = string
  default = "us-east-1"
}
variable "instance_count" {
  type        = number
  description = "Número de instâncias"
}
```

Definindo Valores de Variável

`-var 'region=us-west-2'`	Flag de CLI
`-var-file=prod.tfvars`	Carregar de um arquivo **`.tfvars`**
`terraform.tfvars`	Carregado automaticamente se presente
`TF_VAR_region`	Variável de ambiente
`Prompt interativo`	Solicitado no plan/apply se não houver padrão

Tipos de Variável

`string`	**`"us-east-1"`**
`number`	**`42`**
`bool`	**`true`** / **`false`**
`list(string)`	**`["a", "b"]`**
`map(string)`	**`{ key = "val" }`**
`object({...})`	Tipo estruturado com atributos nomeados

Saídas

Definindo Saídas

```
output "instance_ip" {
  value       = aws_instance.web.public_ip
  description = "IP público do servidor web"
}
output "db_password" {
  value     = random_password.db.result
  sensitive = true
}
```

Comandos de Saída

`terraform output`	Imprimir todas as saídas
`terraform output instance_ip`	Imprimir uma saída específica
`terraform output -json`	Formato JSON para scripts
`sensitive = true`	Ocultar valor da saída de CLI
`module.vpc.vpc_id`	Acessar saídas de módulo filho

Estado

Backend Remoto

```
terraform {
  backend "s3" {
    bucket = "my-tf-state"
    key    = "prod/terraform.tfstate"
    region = "us-east-1"
  }
}
```

Comandos de Estado

`terraform state list`	Listar todos os recursos no estado
`terraform state show <end>`	Exibir atributos de um recurso
`terraform state mv <src> <dst>`	Renomear / mover um recurso no estado
`terraform state rm <end>`	Remover recurso do estado (manter infraestrutura)
`terraform state pull`	Baixar estado remoto para stdout
`terraform import <end> <id>`	Importar infraestrutura existente para o estado

Módulos

Usando Módulos

```
module "vpc" {
  source  = "terraform-aws-modules/vpc/aws"
  version = "~> 5.0"
  cidr    = "10.0.0.0/16"
}
```

Fontes de Módulo

`"./modules/vpc"`	Caminho local
`"terraform-aws-modules/vpc/aws"`	Terraform Registry
`"github.com/org/repo//module"`	Repositório GitHub
`"s3::https://bucket/module.zip"`	Bucket S3

Estrutura do Módulo

```
modules/vpc/
  main.tf          # recursos
  variables.tf     # variáveis de entrada
  outputs.tf       # valores de saída
```

Fontes de Dados

Lendo Recursos Existentes

```
data "aws_ami" "ubuntu" {
  most_recent = true
  filter {
    name   = "name"
    values = ["ubuntu/images/hvm-ssd/*"]
  }
  owners = ["099720109477"]
}
```

Fontes de Dados Comuns

`data.aws_ami`	Buscar uma AMI por filtros
`data.aws_vpc`	Buscar VPC existente
`data.aws_caller_identity`	ID da conta AWS atual
`data.aws_region`	Região AWS atual
`data.terraform_remote_state`	Ler saídas de outro arquivo de estado
`data.external`	Executar programa externo para obter dados

Ciclo de Vida

Regras de Ciclo de Vida

```
resource "aws_instance" "web" {
  lifecycle {
    create_before_destroy = true
    prevent_destroy       = true
    ignore_changes        = [tags]
  }
}
```

Referência Rápida do Terraform

Opções de Ciclo de Vida

`create_before_destroy`	Criar substituto antes de destruir o antigo
`prevent_destroy`	Erro se **`terraform destroy`** visar este recurso
`ignore_changes`	Não detectar drift nos atributos listados
`replace_triggered_by`	Forçar substituição quando recurso referenciado muda
`precondition`	Validar suposições antes de aplicar
`postcondition`	Validar resultados após aplicar

Padrões Comuns

Loops e Condicionais

```
# for_each com um mapa
resource "aws_iam_user" "users" {
  for_each = toset(["alice", "bob"])
  name     = each.value
}
# recurso condicional
count = var.create_db ? 1 : 0
```

Funções Úteis

`file("key.pub")`	Ler conteúdo de arquivo
`join(", ", list)`	Juntar lista em string
`lookup(map, key, default)`	Pesquisa em mapa com fallback
`length(list)`	Número de elementos
`toset(["a", "b"])`	Converter lista para conjunto (para for_each)
`try(expr, fallback)`	Retornar fallback se expr gerar erro
`templatefile(path, vars)`	Renderizar um arquivo de template