

REFERENCIA RÁPIDA DE TERRAFORM

Proveedores, recursos, variables, estado, módulos

Fundamentos

Flujo de Trabajo Principal

```
terraform init # instalar proveedores y módulos
terraform plan # previsualizar cambios
terraform apply # aplicar cambios
terraform destroy # desmantelar todos los recursos
```

Comandos Esenciales

terraform init Inicializar directorio de trabajo, descargar proveedores

terraform plan Mostrar plan de ejecución sin aplicar

terraform apply Aplicar cambios a la infraestructura

terraform destroy Destruir todos los recursos gestionados

terraform fmt Formatear archivos `.tf` al estilo canónico

terraform validate Verificar sintaxis de la configuración

terraform show Mostrar estado actual o plan

terraform output Imprimir valores de salida

Proveedores

Configuración del Proveedor

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

Notas sobre Proveedores

source Dirección del registro (`hashicorp/aws`, `hashicorp/google`)

version Restricción de versión (`>= 5.0`, `>= 3.0, < 4.0`)

terraform.lock.hcl Archivo de bloqueo — confirmar al control de versiones

alias Usar múltiples configuraciones para el mismo proveedor

Recursos

Bloques de Recursos

```
resource "aws_instance" "web" {
  ami           = "ami-9c55b199cbfafef1f0"
  instance_type = "t3.micro"
  tags = { Name = "web-server" }
}
```

Meta-Argumentos de Recursos

depends_on Dependencia explícita de otro recurso

count Crear múltiples instancias (`count = 3`)

for_each Crear instancias desde un mapa o conjunto

provider Seleccionar un alias de proveedor no predeterminado

lifecycle Personalizar comportamiento de crear/actualizar/destruir

Referenciar Recursos

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

Variables

Declarar Variables

```
variable "region" {
  type = string
  default = "us-east-1"
}
variable "instance count" {
  type = number
  description = "Número de instancias"
}
```

Establecer Valores de Variables

-var region=us-west-2 Opción de línea de comandos

-var-file=prod.tfvars Cargar desde un archivo `.tfvars`

terraform.tfvars Se carga automáticamente si existe

TF_VAR_region Variable de entorno

Prompt interactivo Se solicita en `plan/apply` si no hay valor por defecto

Tipos de Variables

string `"us-east-1"`

number `42`

bool `true` / `false`

list(string) `["a", "b"]`

map(string) `{ key = "val" }`

object({...}) Tipo estructurado con atributos con nombre

Salidas

Definir Salidas

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

Comandos de Salida

terraform output Imprimir todas las salidas

terraform output instance_ip Imprimir una salida específica

terraform output -json Formato JSON para scripting

sensitive = true Ocultar valor de la salida CLI

module.vpc.vpc_id Acceder a salidas de módulos hijo

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 los recursos en estado

terraform state show <addr> Mostrar atributos de un recurso

terraform state mv <src> <dst> Renombrar / mover un recurso en estado

terraform state rm <addr> Eliminar recurso del estado (conservar infraestructura)

terraform state pull Descargar estado remoto a `stdout`

terraform import <addr> <id> Importar infraestructura existente al estado

Módulos

Usar Módulos

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

Fuentes de Módulos

"/modules/vpc" Ruta local

"terraform-aws-modules/vpc/aws" Registro de Terraform

"github.com/org/repo/module" Repositorio de GitHub

"s3:https://bucket/module.zip" Bucket S3

Estructura del Módulo

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

Fuentes de Datos

Leer Recursos Existentes

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

Fuentes de Datos Comunes

data.aws_ami Buscar una AMI por filtros

data.aws_vpc Buscar VPC existente

data.aws_caller_identity ID de cuenta AWS actual

data.aws_region Región AWS actual

data.terraform_remote_state Leer salidas de otro archivo de estado

data.external Ejecutar un programa externo para obtener datos

Ciclo de Vida

Reglas de Ciclo de Vida

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

Opciones de Ciclo de Vida

create_before_destroy Crear reemplazo antes de destruir el antiguo

prevent_destroy Error si `terraform destroy` apunta a este recurso

ignore_changes No detectar desviaciones en atributos listados

replace_triggered_by Forzar reemplazo cuando el recurso referenciado cambia

precondition Validar supuestos antes de aplicar

postcondition Validar resultados después de aplicar

Patrones Comunes

Bucles y Condicionales

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

Funciones Útiles

file("key.pub") Leer contenido de archivo

join(" ", list) Unir lista en cadena

lookup(map, key, default) Búsqueda en mapa con valor por defecto

length(list) Número de elementos

toset(["a", "b"]) Convertir lista a conjunto (para `for_each`)

try(expr, fallback) Devolver `fallback` si `expr` produce error

templatefile(path, vars) Renderizar un archivo de plantilla