

# 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

<b>terraform init</b>	Inicializar directorio de trabajo, descargar proveedores
<b>terraform plan</b>	Mostrar plan de ejecución sin aplicar
<b>terraform apply</b>	Aplicar cambios a la infraestructura
<b>terraform destroy</b>	Destruir todos los recursos gestionados
<b>terraform fmt</b>	Formatear archivos <b>.tf</b> al estilo canónico
<b>terraform validate</b>	Verificar sintaxis de la configuración
<b>terraform show</b>	Mostrar estado actual o plan
<b>terraform output</b>	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

<b>source</b>	Dirección del registro ( <b>hashicorp/aws</b> , <b>hashicorp/google</b> )
<b>version</b>	Restricción de versión (~> <b>5.0</b> , >= <b>3.0</b> , < <b>4.0</b> )
<b>.terraform.lock.hcl</b>	Archivo de bloqueo — confirmar al control de versiones
<b>alias</b>	Usar múltiples configuraciones para el mismo proveedor

## Recursos

### Bloques de Recursos

```
resource "aws_instance" "web" {
  ami           = "ami-0c55b159cbfaffe1f0"
  instance_type = "t3.micro"
  tags = { Name = "web-server" }
}
```

### Meta-Argumentos de Recursos

<b>depends_on</b>	Dependencia explícita de otro recurso
<b>count</b>	Crear múltiples instancias ( <b>count = 3</b> )
<b>for_each</b>	Crear instancias desde un mapa o conjunto
<b>provider</b>	Seleccionar un alias de proveedor no predeterminado
<b>lifecycle</b>	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

<b>-var 'region=us-west-2'</b>	Opción de línea de comandos
<b>-var-file=prod.tfvars</b>	Cargar desde un archivo <b>.tfvars</b>
<b>terraform.tfvars</b>	Se carga automáticamente si existe
<b>TF_VAR_region</b>	Variable de entorno
<b>Prompt interactivo</b>	Se solicita en plan/apply si no hay valor por defecto

### Tipos de Variables

<b>string</b>	"us-east-1"
<b>number</b>	42
<b>bool</b>	true / false
<b>list(string)</b>	["a", "b"]
<b>map(string)</b>	{ key = "val" }
<b>object({...})</b>	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

<b>terraform output</b>	Imprimir todas las salidas
<b>terraform output instance_ip</b>	Imprimir una salida específica
<b>terraform output -json</b>	Formato JSON para scripting
<b>sensitive = true</b>	Ocultar valor de la salida CLI
<b>module.vpc.vpc_id</b>	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

<b>terraform state list</b>	Listar todos los recursos en estado
<b>terraform state show &lt;addr&gt;</b>	Mostrar atributos de un recurso
<b>terraform state mv &lt;src&gt; &lt;dst&gt;</b>	Renombrar / mover un recurso en estado
<b>terraform state rm &lt;addr&gt;</b>	Eliminar recurso del estado (conservar infraestructura)
<b>terraform state pull</b>	Descargar estado remoto a stdout
<b>terraform import &lt;addr&gt; &lt;id&gt;</b>	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

<b>./modules/vpc</b>	Ruta local
<b>"terraform-aws-modules/vpc/aws"</b>	Registro de Terraform
<b>"github.com/org/repo/module"</b>	Repositorio de GitHub
<b>"s3://https://bucket/module.zip"</b>	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

<b>data.aws_ami</b>	Buscar una AMI por filtros
<b>data.aws_vpc</b>	Buscar VPC existente
<b>data.aws_caller_identity</b>	ID de cuenta AWS actual
<b>data.aws_region</b>	Región AWS actual
<b>data.terraform_remote_state</b>	Leer salidas de otro archivo de estado
<b>data.external</b>	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]
  }
}
```

# Referencia Rápida de Terraform

---

## Opciones de Ciclo de Vida

<b>create_before_destroy</b>	Crear reemplazo antes de destruir el antiguo
<b>prevent_destroy</b>	Error si <b>terraform destroy</b> apunta a este recurso
<b>ignore_changes</b>	No detectar desviaciones en atributos listados
<b>replace_triggered_by</b>	Forzar reemplazo cuando el recurso referenciado cambia
<b>precondition</b>	Validar supuestos antes de aplicar
<b>postcondition</b>	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

<b>file("key.pub")</b>	Leer contenido de archivo
<b>join(" ", list)</b>	Unir lista en cadena
<b>lookup(map, key, default)</b>	Búsqueda en mapa con valor por defecto
<b>length(list)</b>	Número de elementos
<b>toset(["a", "b"])</b>	Convertir lista a conjunto (para for_each)
<b>try(expr, fallback)</b>	Devolver fallback si expr produce error
<b>templatefile(path, vars)</b>	Renderizar un archivo de plantilla