

THAM KHẢO NHANH TERRAFORM

Providers, resources, biến, state, modules

Cơ bản

Quy trình cốt lõi

```
terraform init # install providers & modules
terraform plan # preview changes
terraform apply # apply changes
terraform destroy # tear down all resources
```

Lệnh cần thiết

terraform init Khởi tạo thư mục làm việc, tải providers
terraform plan Hiện thị kế hoạch thực thi mà không áp dụng
terraform apply Áp dụng thay đổi vào hạ tầng
terraform destroy Phá hủy tất cả resources được quản lý
terraform fmt Định dạng file *.tf theo phong cách chuẩn
terraform validate Kiểm tra cú pháp cấu hình
terraform show Hiện thị state hoặc plan hiện tại
terraform output In các giá trị output

Providers

Cấu hình Provider

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

Ghi chú về Provider

source Địa chỉ registry (`hashicorp/aws`, `hashicorp/google`)
version Ràng buộc phiên bản (`>= 5.0`, `>= 3.0, < 4.0`)
terraform.lock.hcl File lock — commit vào version control
alias Dùng nhiều cấu hình cho cùng một provider

Resources

Resource Blocks

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

Meta-Arguments của Resource

depends_on Phụ thuộc rõ ràng vào resource khác
count Tạo nhiều instances (`count = 3`)
for_each Tạo instances từ map hoặc set
provider Chọn alias provider không mặc định
lifecycle Tùy chỉnh hành vi tạo/cập nhật/phá hủy

Tham chiếu Resources

```
# type.name.attribute
aws_instance.web.id
aws_instance.web.public_ip
aws_vpc.main.cidr_block
```

Biến số

Khai báo biến

```
variable "region" {
  type = string
  default = "us-east-1"
}
variable "instance_count" {
  type = number
  description = "Number of instances"
}
```

Thiết lập giá trị biến

-var `region=us-west-2` Cờ CLI
-var-file=prod.tfvars Tải từ file *.tfvars
terraform.tfvars Tự động tải nếu tồn tại
TF_VAR_region Biến môi trường
interactive prompt Hỏi khi plan/apply nếu không có giá trị mặc định

Các loại biến

string `"us-east-1"`
number `42`
bool `true` / `false`
list(string) `["a", "b"]`
map(string) `{key="val"}`
object {...} Kiểu có cấu trúc với các thuộc tính đặt tên

Outputs

Định nghĩa Outputs

```
output "instance_ip" {
  value = aws_instance.web.public_ip
  description = "Public IP of the web server"
}
output "db_password" {
  value = random_password.db.result
  sensitive = true
}
```

Lệnh Output

terraform output In tất cả outputs
terraform output instance_ip In một output cụ thể
terraform output -json Định dạng JSON để scripting
sensitive = true Ẩn giá trị khỏi output CLI
module.vpc.vpc_id Truy cập outputs của module con

State

Remote Backend

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

Lệnh State

Lệnh terraform state list

terraform state show <addr> Liệt kê tất cả resources trong state
Hiện thị thuộc tính của một resource

terraform state mv <src> <dst> Đổi tên / di chuyển resource trong state

terraform state rm <addr> Xóa resource khỏi state (giữ hạ tầng)

terraform state pull Tải remote state về stdout

terraform import <addr> <id> Import hạ tầng hiện có vào state

Modules

Dùng Modules

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

Nguồn Module

"/modules/vpc" Đường dẫn cục bộ
"terraform-aws-modules/vpc/aws" Terraform Registry
"github.com/org/repo/module" GitHub repository
"s3:https://bucket/module.zip" S3 bucket

Cấu trúc Module

```
modules/vpc/
main.tf # resources
variables.tf # input variables
outputs.tf # output values
```

Data Sources

Đọc resources hiện có

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

Data Sources phổ biến

data.aws_ami Tìm AMI theo bộ lọc
data.aws_vpc Tìm VPC hiện có
data.aws_caller_identity ID tài khoản AWS hiện tại
data.aws_region Region AWS hiện tại
data.terraform_remote_state Đọc outputs từ file state khác
data.external Chạy chương trình bên ngoài để lấy dữ liệu

Lifecycle

Quy tắc Lifecycle

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

Tùy chọn Lifecycle

create_before_destroy Tạo replacement trước khi phá hủy cái cũ
prevent_destroy Bảo lỗi nếu `terraform destroy` nhằm vào resource này
ignore_changes Không phát hiện drift trên các thuộc tính được liệt kê

replace_triggered_by Bắt buộc thay thế khi resource được tham chiếu thay đổi

precondition Xác thực giả định trước khi apply

postcondition Xác thực kết quả sau khi apply

Các mẫu thường dùng

Vòng lặp & Điều kiện

```
# for each with a map
resource "aws_iam_user" "users" {
  for_each = toset(["alice", "bob"])
  name = each.value
}
# conditional resource
count = var.create_db ? 1 : 0
```

Hàm hữu ích

file("key.pub") Đọc nội dung file
join(" ", list) Nối list thành chuỗi
lookup(map, key, default) Tìm trong map với giá trị dự phòng
length(list) Số phần tử
toset(["a", "b"]) Chuyển list thành set (cho for_each)
try(expr, fallback) Trả về dự phòng nếu expr lỗi
templatefile(path, vars) Render một file template