

Referensi Cepat Terraform

Provider, resource, variabel, state, modul

Dasar

Alur Kerja Inti

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

Perintah Esensial

terraform init	Inisialisasi direktori kerja, unduh provider
terraform plan	Tampilkan rencana eksekusi tanpa menerapkan
terraform apply	Terapkan perubahan ke infrastruktur
terraform destroy	Hancurkan semua resource yang dikelola
terraform fmt	Format file .tf ke gaya kanonik
terraform validate	Periksa sintaks konfigurasi
terraform show	Tampilkan state atau rencana saat ini
terraform output	Cetak nilai output

Provider

Konfigurasi Provider

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

Catatan Provider

source	Alamat registry (hashicorp/aws , hashicorp/google)
version	Batasan versi (~> 5.0, >= 3.0, < 4.0)
.terraform.lock.hcl	Lock file — commit ke version control
alias	Gunakan beberapa konfigurasi untuk provider yang sama

Resource

Blok Resource

```
resource "aws_instance" "web" {
  ami           = "ami-0c55b159cbfafa1f0"
  instance_type = "t3.micro"
  tags = { Name = "web-server" }
}
```

Meta-Argumen Resource

depends_on	Dependensi eksplisit pada resource lain
count	Buat beberapa instance (count = 3)
for_each	Buat instance dari map atau set
provider	Pilih alias provider non-default
lifecycle	Sesuaikan perilaku buat/perbarui/hancurkan

Merujuk Resource

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

Variabel

Deklarasi Variabel

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

Mengatur Nilai Variabel

-var 'region=us-west-2'	Flag CLI
-var-file=prod.tfvars	Muat dari file .tfvars
terraform.tfvars	Dimuat otomatis jika ada
TF_VAR_region	Variabel environment
Interactive prompt	Ditanyakan saat plan/apply jika tidak ada default

Tipe Variabel

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

Output

Mendefinisikan Output

```
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
}
```

Perintah Output

terraform output	Cetak semua output
terraform output instance_ip	Cetak output tertentu
terraform output -json	Format JSON untuk scripting
sensitive = true	Sembunyikan nilai dari output CLI
module.vpc.vpc_id	Akses output modul anak

State

Remote Backend

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

Perintah State

terraform state list	Tampilkan semua resource dalam state
terraform state show <addr>	Tampilkan atribut resource dalam state
terraform state mv <src> <dst>	Ganti nama / pindahkan resource dalam state
terraform state rm <addr>	Hapus resource dari state (infra tetap ada)
terraform state pull	Unduh state remote ke stdout
terraform import <addr> <id>	Impor infra yang sudah ada ke state

Modul

Menggunakan Modul

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

Sumber Modul

./modules/vpc	Path lokal
"terraform-aws-modules/vpc/aws"	Terraform Registry
"github.com/org/repo/module"	Repository GitHub
"s3::https://bucket/module.zip"	S3 bucket

Struktur Modul

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

Data Source

Membaca Resource yang Ada

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

Data Source Umum

data.aws_ami	Cari AMI berdasarkan filter
data.aws_vpc	Cari VPC yang sudah ada
data.aws_caller_identity	ID akun AWS saat ini
data.aws_region	Region AWS saat ini
data.terraform_remote_state	Baca output dari file state lain
data.external	Jalankan program eksternal untuk data

Lifecycle

Aturan Lifecycle

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

Referensi Cepat Terraform

Opsi Lifecycle

create_before_destroy	Buat pengganti sebelum menghancurkan yang lama
prevent_destroy	Error jika terraform destroy menarget ini
ignore_changes	Jangan deteksi drift pada atribut yang terdaftar
replace_triggered_by	Paksa penggantian ketika resource yang dirujuk berubah
precondition	Validasi asumsi sebelum apply
postcondition	Validasi hasil setelah apply

Pola Umum

Loop & Kondisional

```
# 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
```

Fungsi Berguna

file("key.pub")	Baca isi file
join(" ", list)	Gabungkan list menjadi string
lookup(map, key, default)	Pencarian map dengan fallback
length(list)	Jumlah elemen
toset(["a", "b"])	Ubah list ke set (untuk for_each)
try(expr, fallback)	Kembalikan fallback jika expr error
templatefile(path, vars)	Render file template