

Terraform 빠른 참조

프로바이더, 리소스, 변수, 상태, 모듈

기본

핵심 워크플로

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

필수 명령어

terraform init	작업 디렉터리 초기화, 프로바이더 다운로드
terraform plan	적용 없이 실행 계획 표시
terraform apply	인프라에 변경 사항 적용
terraform destroy	관리 중인 모든 리소스 삭제
terraform fmt	.tf 파일을 표준 스타일로 포맷
terraform validate	구성 구문 검사
terraform show	현재 상태 또는 계획 표시
terraform output	출력 값 표시

프로바이더

프로바이더 구성

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

프로바이더 참고사항

source	레지스트리 주소 (hashicorp/aws , hashicorp/google)
version	버전 제약 (~> 5.0, >= 3.0, < 4.0)
.terraform.lock.hcl	잠금 파일 — 버전 관리에 커밋
alias	동일 프로바이더에 여러 구성 사용

리소스

리소스 블록

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

리소스 메타 인수

depends_on	다른 리소스에 대한 명시적 의존성
count	여러 인스턴스 생성 (count = 3)
for_each	맵 또는 집합에서 인스턴스 생성
provider	기본값이 아닌 프로바이더 별칭 선택
lifecycle	생성/업데이트/삭제 동작 커스터마이징

리소스 참조

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

변수

변수 선언

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

변수 값 설정

-var 'region=us-west-2'	CLI 플래그
-var-file=prod.tfvars	.tfvars 파일에서 로드
terraform.tfvars	존재하는 경우 자동 로드
TF_VAR_region	환경 변수
Interactive prompt	기본값 없을 때 plan/apply에서 입력 요청

변수 타입

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

출력

출력 정의

```
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
}
```

출력 명령어

terraform output	모든 출력 표시
terraform output instance_ip	특정 출력 표시
terraform output -json	스크립팅용 JSON 형식
sensitive = true	CLI 출력에서 값 숨김
module.vpc.vpc_id	자식 모듈 출력 접근

상태

원격 백엔드

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

상태 명령어

terraform state list	상태 내 모든 리소스 목록
terraform state show <addr>	리소스 속성 표시
terraform state mv <src> <dst>	상태 내 리소스 이름 변경 / 이동
terraform state rm <addr>	상태에서 리소스 제거 (인프라 유지)
terraform state pull	원격 상태를 stdout으로 다운로드
terraform import <addr> <id>	기존 인프라를 상태로 임포트

모듈

모듈 사용

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

모듈 소스

"/modules/vpc"	로컬 경로
"terraform-aws-modules/vpc/aws"	Terraform 레지스트리
"github.com/org/repo/module"	GitHub 리포지터리
"s3::https://bucket/module.zip"	S3 버킷

모듈 구조

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

데이터 소스

기존 리소스 읽기

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

주요 데이터 소스

data.aws_ami	필터로 AMI 조회
data.aws_vpc	기존 VPC 조회
data.aws_caller_identity	현재 AWS 계정 ID
data.aws_region	현재 AWS 리전
data.terraform_remote_state	다른 상태 파일의 출력 읽기
data.external	데이터를 위해 외부 프로그램 실행

라이프사이클

라이프사이클 규칙

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

라이프사이클 옵션

create_before_destroy	기존 것 삭제 전 대체 리소스 생성
prevent_destroy	terraform destroy 로 이 리소스 대상 시 오류
ignore_changes	나열된 속성의 드리프트 감지 안 함
replace_triggered_by	참조 리소스 변경 시 강제 교체
precondition	apply 전 전제 조건 검증
postcondition	apply 후 결과 검증

Terraform 빠른 참조

자주 쓰는 패턴

루프 및 조건

```
# 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
```

유용한 함수

file("key.pub")	파일 내용 읽기
join(", ", list)	리스트를 문자열로 결합
lookup(map, key, default)	폴백이 있는 맵 조회
length(list)	요소 수
toset(["a", "b"])	리스트를 집합으로 변환 (for_each용)
try(expr, fallback)	expr 오류 시 폴백 반환
templatefile(path, vars)	템플릿 파일 렌더링