

Terraform 모듈 제작

정지훈

목차

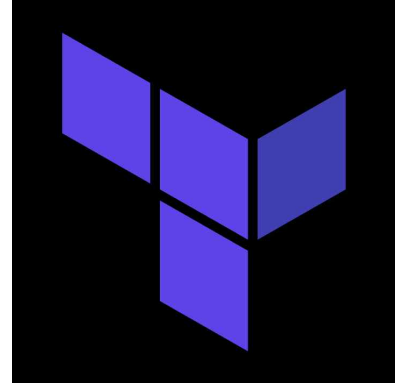
- 개요
- 초기 구성
- terraform 모듈 제작

- 개요

테라폼(Terraform)은 클라우드 및 온프레미스 리소스를 안전하고 효율적으로 빌드, 변경 및 버전화할 수 있는 코드형 인프라 도구입니다.

* Terraform 장점

- 장점 버전 컨트롤 가능
- 손쉬운 공유와 협업
- 문서로서의 코드
- 재사용과 복제

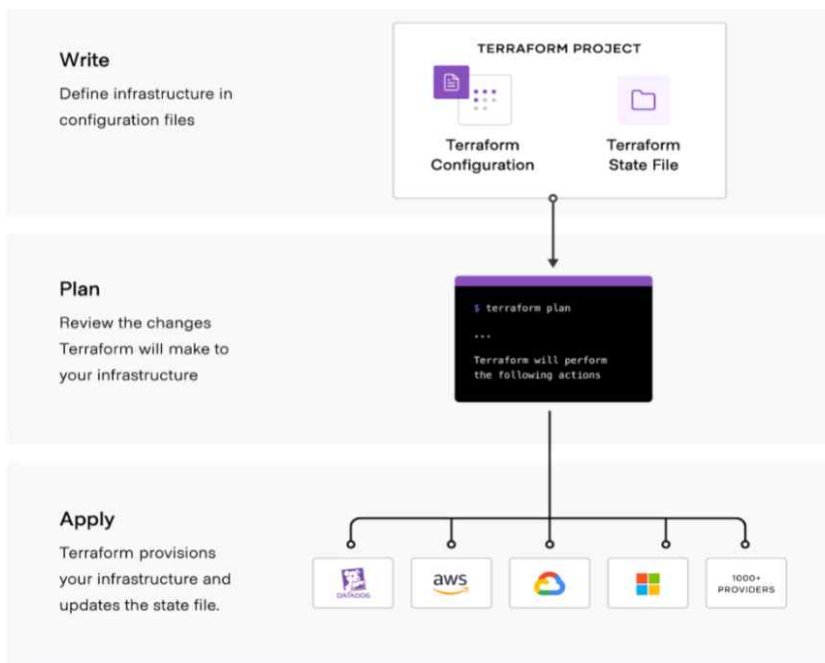


* Terraform 작동방식(3단계로 구성)

쓰기(write): 여러 클라우드 공급자 및 서비스에 걸쳐 있을 수 있는 리소스를 정의합니다. 예를 들어 보안 그룹 및 로드 밸런서가 있는 Virtual Private Cloud(VPC) 네트워크의 가상 머신에 애플리케이션을 배포하는 구성을 생성할 수 있습니다.

계획(plan): Terraform은 기존 인프라 및 구성을 기반으로 생성, 업데이트 또는 파괴할 인프라를 설명하는 실행 계획을 생성합니다.

적용(apply): 승인 시 Terraform은 모든 리소스 종속성을 고려하여 올바른 순서로 제안된 작업을 수행합니다. 예를 들어 VPC의 속성을 업데이트하고 해당 VPC의 가상 머신 수를 변경하면 Terraform은 가상 머신을 확장하기 전에 VPC를 다시 생성합니다.



- 초기 구성

```
# vpc
# aws_subnets
# *security group
# *instance -> auto
# ASG
# * lb
# * lb_listener
# * lb_target_group
# * lb_listener_rule
# * security group
# * ASG
```

* 디렉토리 구조

```
● [tf@terraform ~/tf/project2]$ tree
```

```
├── main.tf
├── modules
│   ├── main.tf
│   ├── outputs.tf
│   └── variables.tf
├── terraform.tfstate
├── terraform.tfstate.backup
└── user_data.tftpl
```

1 directory, 7 files

—

- terraform 모듈 제작

\$ mkdir -p project2 # 폴더를 먼저 생성, 그 후에 순차적으로 파일들을 생성

\$ cd ~/tf/project2

\$ mkdir -p modules <- 모듈 생성

\$ touch main.tf, user_data.tftpl <- 파일 생성

\$ vi main.tf

```
data "aws_vpc" "default" {
    default = true
}

data "aws_subnets" "default" {
    filter {
        name = "vpc-id"
        values = [data.aws_vpc.default.id]
    }
}

# 보안 그룹 인스턴스 설정
resource "aws_security_group" "instance" {
    name = var.instance_security_group_name
}

# 보안그룹 룰 적용
ingress {
    from_port      = var.backend_server_port
    to_port        = var.backend_server_port
    protocol       = "tcp"
    cidr_blocks    = ["0.0.0.0/0"]
}

# 이미지 데이터 설정
data "aws_ami" "ubuntu2004" {
    most_recent = true

    filter {
        name = "name"
```

```

        values = ["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-*"]
    }

    filter {
        name    = "virtualization-type"
        values = ["hvm"]
    }
    owners = ["099720109477"]
}

resource "aws_launch_configuration" "default" {
    image_id = data.aws_ami.ubuntu2004.id
    instance_type = "t2.micro"
    security_groups = [aws_security_group.instance.id]
    user_data = file("user_data.tftpl")

    lifecycle {
        create_before_destroy = true
    }
}

#보안그룹 설정
resource "aws_security_group" "default" {
    name = var.alb_security_group_name

    # Allow inbound HTTP requests
    ingress {
        from_port    = var.frontend_server_port
        to_port      = var.backend_server_port
        protocol     = "tcp"
        cidr_blocks = ["0.0.0.0/0"]
    }

    # Allow all outbound requests
    egress {
        from_port    = 0
        to_port      = 0
        protocol     = "-1"
        cidr_blocks = ["0.0.0.0/0"]
    }
}

```

```

resource "aws_lb" "default" {
  name = var.alb_name

  load_balancer_type = "application"
  subnets = data.aws_subnets.default.ids
  security_groups = [aws_security_group.default.id]
}

resource "aws_lb_listener" "http" {
  load_balancer_arn = aws_lb.default.arn
  port              = var.frontend_server_port
  protocol          = "HTTP"

  # By default, return a simple 404 page
  default_action {
    type = "fixed-response"

    fixed_response {
      content_type = "text/plain"
      message_body = "404: page not found"
      status_code  = 404
    }
  }
}

# lb(=Load Balance) 타겟 그룹 설정
resource "aws_lb_target_group" "asg" {
  name = var.alb_name

  port      = var.backend_server_port
  protocol  = "HTTP"
  vpc_id    = data.aws_vpc.default.id

  health_check {
    path            = "/"
    protocol        = "HTTP"
    matcher         = "200"
    interval        = 15
    timeout         = 3
    healthy_threshold = 2
  }
}

```

```

        unhealthy_threshold = 2
    }
}

resource "aws_lb_listener_rule" "asg" {
    listener_arn = aws_lb_listener.http.arn
    priority     = 100

    condition {
        path_pattern {
            values = ["*"]
        }
    }

    action {
        type                = "forward"
        target_group_arn = aws_lb_target_group.asg.arn
    }
}

# 오토스케일링 그룹 설정
resource "aws_autoscaling_group" "default" {
    launch_configuration = aws_launch_configuration.default.name
    vpc_zone_identifier = data.aws_subnets.default.ids

    target_group_arns = [aws_lb_target_group.asg.arn]
    health_check_type = "ELB"

    min_size = 2
    max_size = 3

    tag {
        key           = "Name"
        value         = "jjh_asg"
        propagate_at_launch = true
    }
}

```

\$ vi variables.tf

```

variable "instance_security_group_name" {
    description = "EC2 Instance Security Group Name"
}

```



```

    type = string
  }
  # 백엔드 사이트 포트 8080에 대한 변수 설정
  variable "backend_server_port" {
    description = "Backend Web Server Port Number"
    type = number
  }
  # ALB 보안그룹 변수 설정
  variable "alb_security_group_name" {
    description = "ALB Security Group Name"
    type = string
  }
  # 프론트엔드 서버 포트 80에 대한 변수 설정
  variable "frontend_server_port" {
    description = "Frontend Web Server Port Number"
    type = number
  }
  # alb 변수설정
  variable "alb_name" {
    description = "Application Load Balancer"
    type = string
  }
}

```

\$ vi output.tf

```

output "alb_dns_name" {
  value      = aws_lb.default.dns_name
  description = "The domain name of the load balancer"
}

```

\$ terraform plan (구축 계획 확인)

\$ terraform apply -auto-approve (=ccc) - 리소스 생성

* 보안그룹 생성 확인

Name	보안 그룹 ID	보안 그룹 이름	VPC ID	설명	소유자	인바운드 규칙 수	아웃바운드 규칙 수
-	sg-040c05b4d099e293	instance-sg	vpc-0b738530032f0ba3	Managed by Terraform	870290872477	1 권한 항목	0 권한 항목
-	sg-0bf9d81ce6e72ee8a	default	vpc-0b738530032f0ba3	default VPC security gr...	870290872477	1 권한 항목	1 권한 항목
-	sg-0bc00e703b685c662	alb-sg	vpc-0b738530032f0ba3	Managed by Terraform	870290872477	1 권한 항목	1 권한 항목

* EC2

Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	정보 상태	가용 영역	퍼블릭 IPv4 DNS	퍼블릭 IPv4 ...	탄력적 IP	IF
bisc_asg	i-00d2b0db0a311aaa	종료됨	t2.micro	-	정보 없음	us-east-2a	-	-	-	-
bisc_asg	i-0cd7cc7e795fcb5e3	종료됨	t2.micro	-	정보 없음	us-east-2a	-	-	-	-
bisc_asg	i-0e4b17a77d2a52855	실행 중	t2.micro	2/2개 검사 통과...	정보 없음	us-east-2a	ec2-18-117-186-93.us-...	18.117.186.93	-	-
bisc_asg	i-0562370403a0032ae	실행 중	t2.micro	2/2개 검사 통과...	정보 없음	us-east-2a	ec2-18-117-745.us-east...	18.117.745	-	-

\$ terraform destroy(=ddd) <- 정상적으로 삭제가 되었는지 aws 사이트
에 들어가서 꼭 확인하기

```
[*] terraform ~/tf/project2$ ddd
module.jjh-module.data.aws_vpc.default: Reading...
module.jjh-module.data.aws_security_group.instance: Refreshing state... [id=sg-040c05b84d99e293]
module.jjh-module.data.aws_security_group.default: Refreshing state... [id=sg-0bc00e703b685c662]
module.jjh-module.data.aws_ami.ubuntu2004: Reading...
module.jjh-module.data.aws_ami.ubuntu2004: Read complete after 1s [id=ami-0ada6d94f396377f2]
module.jjh-module.aws_launch_configuration.default: Refreshing state... [id=terraform-20221205110550003200000001]
module.jjh-module.data.aws_vpc.default: Read complete after 2s [id=vpc-0b738530032f0bac3]
module.jjh-module.data.aws_subnets.default: Reading...
module.jjh-module.aws_lb_target_group.asg: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-2:870290872477:targetgroup/alb/1841639ac2b25976]
module.jjh-module.data.aws_subnets.default: Read complete after 0s [id=us-east-2]
module.jjh-module.aws_autoscaling_group.default: Refreshing state... [id=terraform-20221205110551655600000002]

Terraform used the selected providers to generate the following execution plan. Resource
actions are indicated with the following symbols:
- destroy

Terraform will perform the following actions:

# module.jjh-module.aws_autoscaling_group.default will be destroyed
- resource "aws_autoscaling_group" "default" {
  - arn = "arn:aws:autoscaling:us-east-2:870290872477:autoScalingGroup:fc04ad6b-776c-4a62-b199-bece2db892fe:autoScalingGroupName/terraform-20221205110551655600000002" -> null
  - availability_zones = [
    - "us-east-2a",
  ] -> null
  - capacity_rebalance = false -> null
  - default_cooldown = 300 -> null
  - default_instance_warmup = 0 -> null
  - desired_capacity = 2 -> null
  - enabled_metrics = [] -> null
  - force_delete = false -> null
  - force_delete_warm_pool = false -> null
  - health_check_grace_period = 300 -> null
  - health_check_type = "ELB" -> null
  - id = "terraform-20221205110551655600000002" -> null
  - launch_configuration = "terraform-20221205110550003200000001" -> null
  - load_balancers = [] -> null
}
```

* VPC

Name	VPC ID	상태	IPv4 CIDR	IPv6 CIDR	DHCP 옵션 세트	기본 라우팅 테이블	기본
-	vpc-0b738530032f0bac3	Available	172.31.0.0/16	-	dopt-0ca342ae5af950...	rtb-0ad334dce1d28b881	ad-

* EC2

Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	정보 상태	가용 영역	퍼블릭 IPv4 DNS	퍼블릭 IPv4 ...	탄력적 IP	IF
bisc_asg	i-0e4b17a77d2a52855	종료됨	t2.micro	-	정보 없음	us-east-2a	-	-	-	-
bisc_asg	i-0562370403a0032ae	종료됨	t2.micro	-	정보 없음	us-east-2a	-	-	-	-