

Terraform 기본문법

1. Terraform 기본 문법

Terraform 공급자 요건지정

```
terraform {  
  # Terraform 버전 지정  
  required_version = ">= 1.0.0, < 2.0.0"  
  
  # 공급자 지정  
  required_providers {  
    aws = {  
      source  = "hashicorp/aws"  
      version = "~> 4.0"  
    }  
  }  
}
```

required_version : 테라폼 버전

required_providers : 각 인수는 하나의 공급자를 활성화한다.

source : 사용하려는 공급자의 주소

version : 사용가능한 공급자의 버전

공급자 지정

```
provider "aws" {  
  region = "ap-northeast-2"  
}
```

provider : 공급자 지정

region : 리전

여러 공급자 지정

```
provider "aws" {  
  region = "ap-northeast-2"  
}  
  
provider "aws" {  
  alias = "virginia"  
  region = "us-east-1"  
}  
  
...  
  
resource "aws_acm_certificate" "demo_sideeffect_kr" {  
  provider      = "aws.virginia"  
  
  domain_name    = "demo.sideeffect.kr"  
  validation_method = "EMAIL"  
}
```

Argument

argument는 특정 이름에 값을 할당한다.

```
image_id = "abc123"
```

Block

Block은 다른 콘텐츠에 대한 컨테이너 이다.

```
resource "aws_instance" "example" {  
  ami = "abc123"  
  
  network_interface {  
    # ...  
  }  
}
```

Resource Blocks

```
resource "aws_vpc" "main" {
  cidr_block = var.base_cidr_block
}

<BLOCK TYPE> "<BLOCK LABEL>" "<BLOCK LABEL>" {
  # Block body
  <IDENTIFIER> = <EXPRESSION> # Argument
}
```

파일 확장자

Terraform 언어의 코드는 .tf 파일 확장자를 가진 일반 텍스트 파일에 저장됩니다. .tf.json 파일 확장자로 명명된 언어의 JSON 기반 변형도 있다.

Terraform 코드가 포함된 파일을 종종 구성 파일이라고 한다.



init / plan / apply 명령어에서 .tf나 .tf.json 파일을 찾는데 nested로 하위 디렉토리에 대해서는 탐색을 하지 않는다.

텍스트 인코딩

구성 파일은 항상 UTF-8 인코딩을 사용해야 하며 일반적으로 Windows 스타일 줄 끝 (CRLF)보다는 Unix 스타일 줄 끝(LF)을 사용하지만 둘 다 허용됩니다.

디렉토리 및 모듈

모듈은 디렉토리에 함께 보관되는 .tf 및/또는 .tf.json 파일의 모음이다.

루트 모듈

Terraform은 항상 단일 루트 모듈의 컨텍스트에서 실행된다. 완전한 Terraform 구성은 루트 모듈과 하위 모듈의 트리(루트 모듈에서 호출한 모듈, 해당 모듈에서 호출한 모듈 등 포함)로 구성된다.

Terraform CLI에서 루트 모듈은 Terraform이 호출되는 작업 디렉토리이다. (명령행 옵션을 사용하여 작업 디렉토리 외부에 루트 모듈을 지정할 수 있지만 실제로는 드물다.)

식별자

인수 이름, 블록 유형 이름 및 리소스, 입력 변수 등과 같은 대부분의 Terraform 특정 구성의 이름은 모두 식별자이다.

식별자에는 문자, 숫자, 밑줄(_) 및 하이픈(-)이 포함될 수 있다. 리터럴 숫자의 모호성을 피하기 위해 식별자의 첫 번째 문자는 숫자가 아니어야 한다.

주석

- # 한 줄 주석.
- // 한 줄 주석.
- /* */ 여러 줄 주석.

스타일 규칙

```
# tab을 사용하지 않고 two space를 사용한다.
# keys와 values를 정렬시킨다.
terraform {
  required_version = ">= 1.0.0, < 2.0.0"

  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "~> 4.0"
    }
  }
}

provider "aws" {
  region = "ap-northeast-2"
}

# 인자들을 논리적으로 구분해서 줄로 구분
# meta-argument의 순서, 앞(머리), 뒤(꼬리)의 어디에 둘지 결정
resource "aws_instance" "example" {
  ami          = "ami-0ab04b3ccbadfae1f"
  instance_type = "t2.micro"

  tags = {
    Name = "terraform-example"
  }
}
```

```
}  
}
```

```
# 스타일 규칙에 맞게 자동으로 포맷팅하는 기능  
$ terraform fmt -diff
```