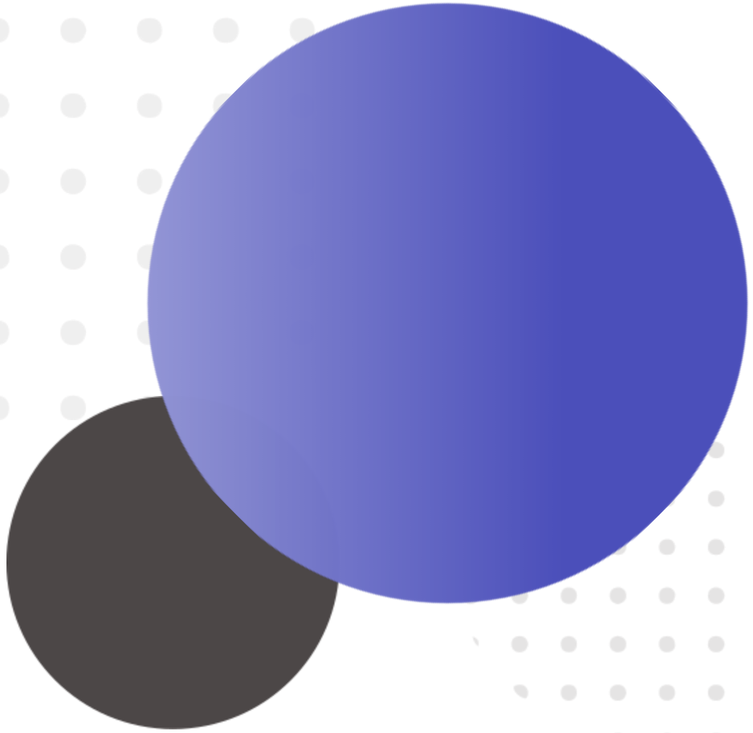


2024년 널카말카 세미나

IaC

인프라를 코드로

일시 2024년 06월 17일

CONTENTS

01

IaC란 무엇인가?

특징 및 장점들 살펴보기

Infrastructure as Code

Infrastructure as Code (IaC)는 인프라 자원을 코드로 정의하고 자동화하여 관리하는 접근 방식입니다.
이는 수동으로 인프라를 구축하고 관리하는 대신, 코드를 사용하여 일관성 있게 인프라를 배포하고
변경 관리할 수 있게 합니다.

장점 및 단점

IaC는 code이기 때문에 code가 갖고 있는 장점을 갖고 있습니다.

- 재사용성

  한번 작성한 코드를 여러 프로젝트에서 다시 사용할 수 있습니다. 이로 인해 시간과 비용이 감소 합니다.

- 가시성 향상

  인프라를 코드로 구성함으로서 인프라 구성에 대한 가시성이 향상됩니다. 이로 인해 문제를 더 빠르게 찾을 수 있습니다.

- 공유 및 버전관리

  코드로 정의 되어 있기 때문에 공유가 간편하고 버전관리에 용이 합니다.

장점 및 단점

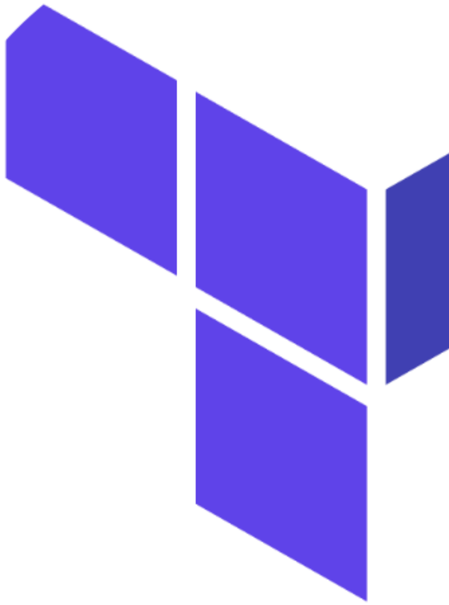
장점이 있으면 단점이 있기 마련

01 초기 학습에 필요 시간

IaC 도구와 개념을 익히는 데 시간이 걸리며, 이를 효과적으로 사용하기 위해서는 개발자와 운영팀이 추가적인 학습과 훈련을 필요로 합니다.

02 동가화 문제

만약 빠르게 대처해야 하는 문제가 생기면 빠르게 대처하기 위해 수동으로 고쳐야 되기에 코드와 실제 인프라가 일치하지 않아 동기화가 깨지는 문제가 생길 수 있습니다.



02

Terraform? Ansible?

절차적 언어 vs 선언적 언어

Terraform? Ansible?

그뭔씹?

테라폼

테라폼은 하시코프에서 오픈소스로 개발중인IaC도구입니다.

클라우드 및 온프레미스 인프라 자원을 코드로 정의하고 자동화하는 오픈소스 도구입니다. HCL을 사용하여 인프라 리소스를 선언적으로 관리하며, 다양한 클라우드 서비스와 호환됩니다.

앤서블

앤서블(Ansible)은 리눅스와 유닉스 기반 시스템의 설정 및 배포 작업을 자동화하기 위한 환경 구성 자동화 도구입니다.

앤서블은 에이전트(agent)가 필요하지 않으며 SSH를 통해 리모트 노드에 명령을 전달하고 YAML 형식으로 작성된 Playbook을 사용하여 간단하게 배포 작업을 수행할 수 있습니다.

절차적 vs 선언적

- 절차적 언어는 코드의 위에서 아래로 순차적으로 살행되는 언어입니다.
- 사용자가 목표상태에 도달하기 위한 작업일체를 기술 합니다.
- 대표적인 언어로는 Ansible이 있으며 어떤 라이브러리를 구동하기 위한 일련의 활동이 전체를 사용자가 기술해야 합니다.

- 선언적 언어는 목표상태 그 자체를 기술 합니다.
- 예를 들면 특정 인스턴스가 특정 서브넷에 속하고 특정 보안그룹에 속하고 등등... 인프라의 상태를 코드로 기술합니다.
- 테라폼이 대표적인 언어이고 목표상태가 변경되면 기존 테라폼이 가지고 있던 state와 비교하여 그 차이 만큼을 인프라에 반영하게 됩니다.
- 절차적 언어와 달리 목표상태에 달하는 세부동작을 사용자가 알 필요가 없습니다. 왜냐면 IaC도구가 알아서 그 차이를 인식하고 세부동작을 수행합니다.

03

Terraform 맛보기

Terraform으로 AWS 사용하기

맛있게 먹어보자!

Window 환경에서 구축

test.tf

```hcl
provider "aws" {
  region = "us-east-1"
}

# VPC 생성
resource "aws_vpc" "main_vpc" {
  cidr_block = "10.0.0.0/16"

  tags = {
    Name = "main-vpc"
  }
}

# 인터넷 게이트웨이 생성
resource "aws_internet_gateway" "main_igw" {
  vpc_id = aws_vpc.main_vpc.id

  tags = {
    Name = "main-igw"
  }
}

# 퍼블릭 서브넷 생성
resource "aws_subnet" "public_subnet" {
  vpc_id                  = aws_vpc.main_vpc.id
  cidr_block              = "10.0.1.0/24"
  availability_zone       = "us-east-1a"
  map_public_ip_on_launch = true

  tags = {
    Name = "public-subnet"
  }
}

# 퍼블릭 라우팅 테이블 생성
resource "aws_route_table" "public_rt" {
  vpc_id = aws_vpc.main_vpc.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.main_igw.id
  }

  tags = {
    Name = "public-rt"
  }
}
```

```hcl
# 퍼블릭 서브넷과 라우팅 테이블 연결
resource "aws_route_table_association" "public_rta" {
  subnet_id      = aws_subnet.public_subnet.id
  route_table_id = aws_route_table.public_rt.id
}

# 보안 그룹 생성 (HTTP 및 SSH 접근 허용)
resource "aws_security_group" "web_sg" {
  vpc_id = aws_vpc.main_vpc.id
  name   = "web-sg"

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "web-sg"
  }
}

# 웹 서버 인스턴스 생성
resource "aws_instance" "web" {
  ami           = "ami-0c55b159cbfafe1f0" # Amazon Linux 2 AMI
  instance_type = "t2.micro"
  subnet_id     = aws_subnet.public_subnet.id
  security_groups = [aws_security_group.web_sg.name]

  tags = {
    Name = "web-server"
  }

  user_data = <<-EOF
              #!/bin/bash
              yum update -y
              yum install -y httpd
              systemctl start httpd
              systemctl enable httpd
              echo "Hello, World!" > /var/www/html/index.html
              EOF
}

# 출력 정보 (인스턴스의 퍼블릭 IP)
output "web_public_ip" {
  value = aws_instance.web.public_ip
}
```

맛있게 먹어보자!

Window 환경에서 구축

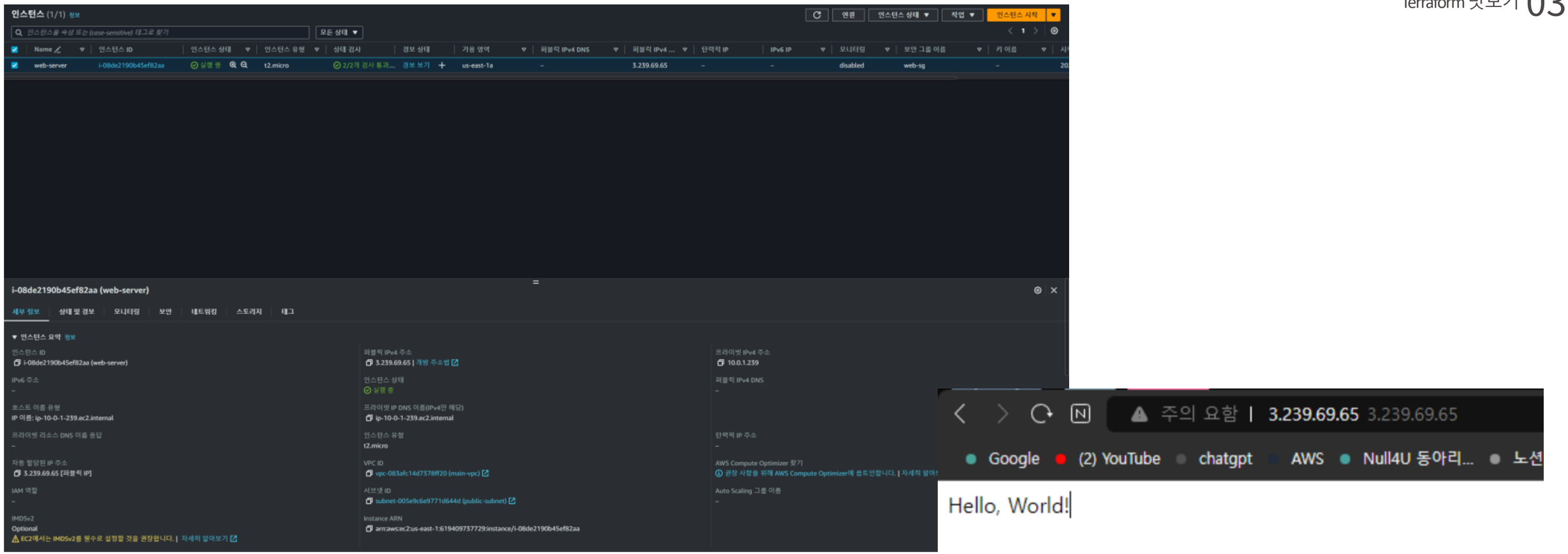
```
Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure
```

- init: 다른 명령을 실행하기 위해 작업 디렉토리를 준비합니다.
- validate: 구성 파일이 유효한지 확인합니다.
- plan: 현재 구성에서 요구하는 변경 사항을 보여줍니다.
- apply: 인프라를 생성하거나 업데이트합니다.
- destroy: 이전에 생성된 인프라를 제거합니다.

```
PS C:\[redacted]\semina> terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v3.76.1

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

```
Apply complete! Resources: 7 added, 0 changed, 0 destroyed.

Outputs:

web_public_ip = "3.239.69.65"
```

에러 없이 생성되면 이렇게 성공 메세지가 출력


인스턴스 (1/1) 정보
연결
인스턴스 상태
작업
인스턴스 시작
모든 상태
Name
인스턴스 ID
인스턴스 상태
인스턴스 유형
상태 검사
경보 상태
가용 영역
퍼블릭 IPv4 DNS
퍼블릭 IPv4 ...
탄력적 IP
IPv6 IP
모니터링
보안 그룹 이름
키 이름
web-server
i-08de2190b45ef82aa
실행 중
t2.micro
2/2개 검사 통과...
경보 보기
us-east-1a
3.239.69.65
disabled
web-sg
i-08de2190b45ef82aa (web-server)
세부 정보
상태 및 경보
모니터링
보안
네트워킹
스토리지
태그
인스턴스 요약 정보
인스턴스 ID
i-08de2190b45ef82aa (web-server)
퍼블릭 IPv4 주소
3.239.69.65 | 개방 주소법
프라이빗 IPv4 주소
10.0.1.239
IPv6 주소
인스턴스 상태
실행 중
퍼블릭 IPv4 DNS
호스트 이름 유형
IP 이름: ip-10-0-1-239.ec2.internal
프라이빗 IP DNS 이름(IPv4만 해당)
ip-10-0-1-239.ec2.internal
프라이빗 리소스 DNS 이름 응답
인스턴스 유형
t2.micro
탄력적 IP 주소
자동 할당된 IP 주소
3.239.69.65 [퍼블릭 IP]
VPC ID
vpc-083afc14d7378ff20 (main-vpc)
AWS Compute Optimizer 찾기
IAM 역할
서브넷 ID
subnet-005e9c6e9771d644d (public-subnet)
Auto Scaling 그룹 이름
IMDSv2
Optional
EC2에서는 IMDSv2를 필수로 설정할 것을 권장합니다. | 자세히 알아보기
Instance ARN
arn:aws:ec2:us-east-1:619409737729:instance/i-08de2190b45ef82aa
주의 요함 | 3.239.69.65 3.239.69.65
Google
(2) YouTube
chatgpt
AWS
Null4U 동아리...
Hello, World!

성공!

2024 널카말카 세미나

THANK YOU

현성

Null4U