Terraform 이해하기

David Hyeshik Yoon
david@2miles.co.kr

코드형 인프라의 장점

- 셀프 서비스
- 속도와 안정성
- 문서화
- 버전 관리
- 확인 및 감사
- 재사용성
- 운영 관리의 편안함과 행복

코드형 인프라 비교 사항

- 구성 관리 vs 배포 도구
- 가변적인 인프라 vs 변하지 않는 인프라
- 절차적 언어 vs 선언적 언어
- 마스터 유무
- 에이전트 유무
- 커뮤니티 규모
- 성숙한 기술 vs 신규 기술

코드형 인프라 비교 사항 – 언어 형태

▪ 절차적 언어

```
- ec2
    count : 10
    image : ami-40d34289
    instance_type : t2.micro
```

```
- ec2
    count : 5
    image : ami-40d34289
    instance_type : t2.micro
```

15대로 늘리는 경우

▪ 선언적 언어

```
resource “aws_instance” “example” {
        count           = 10
        ami             = ” ami-40d34289”
        instance_typ    = “t2.micro”
}
```

```
resource “aws_instance” “example” {
        count           = 15
        ami             = ” ami-40d34289”
        instance_typ    = “t2.micro”
}
```

코드형 인프라 비교 사항 – 언어 형태

- 절차적인 코드들은 인프라의 모든 상태를 저장하기 어려움
- 선언적인 코드들은 최신 인프라 상태를 유지

코드형 인프라 비교 사항 – 마스터 유무

- 마스터 서버가 필요한 툴
  - Chef , Puppet, Salt stack

- 마스터 서버가 없는 툴
  -Ansible, CloudFormation, Heat, Terraform

코드형 인프라 비교 사항 –Agent유무

- Agent가 필요한 툴
  - Chef , Puppet, Salt stack
    (Agent 없이 사용이 가능하지만 제약이 따름)

- Agent가 필요 없는 툴
  -Ansible, CloudFormation, Heat, Terraform

코드형 인프라 비교 사항 – 커뮤니티

	Chef	Puppet	Ansible	SaltStack	CloudFormation	Terraform
Code	Open source	Open source	Open source	Open source	Closed source	Open source
Cloud	All	All	All	All	AWS only	All
Type	Config Mgmt	Config Mgmt	Config Mgmt	Config Mgmt	Orchestration	Orchestration
Infrastructure	Mutable	Mutable	Mutable	Mutable	Immutable	Immutable
Language	Procedural	Declarative	Procedural	Declarative	Declarative	Declarative
Architecture	Client/Server	Client/Server	Client-Only	Client/Server	Client-Only	Client-Only

참조) https://bit.ly/2D27cWA

코드형 인프라 비교 사항 – 성숙도

	제공되기 시작한 시기	현재 버전
Chef	2009	14.11.21
Puppet	2005	6.0
Ansible	2012	2.3
SaltStack	2011	2019.2
CloudFormation	2011	?
Heat	2012	12.0.0
Terraform	2014	0.11.13

Terraform 설치하기

- 다운로드 : www.terraform.io 에서 파일 다운로드
- 운영체제에 맞는 패키지 선택 후 zip 파일 압축 풀기
- terraform 단일 파일로 실행
- 환경 변수 설정

예) 아마존 접속을 위한 환경 변수 등록

$export AWS_ACCESS_KEY_ID=액세스키 ID

$export AWS_SECRET_ACCESS_KEY=비밀 액세스 키

Terraform 명령어

init

테라폼을 수행하기 위한 공급자의 플러그인들을 초기 설정하는 명령어

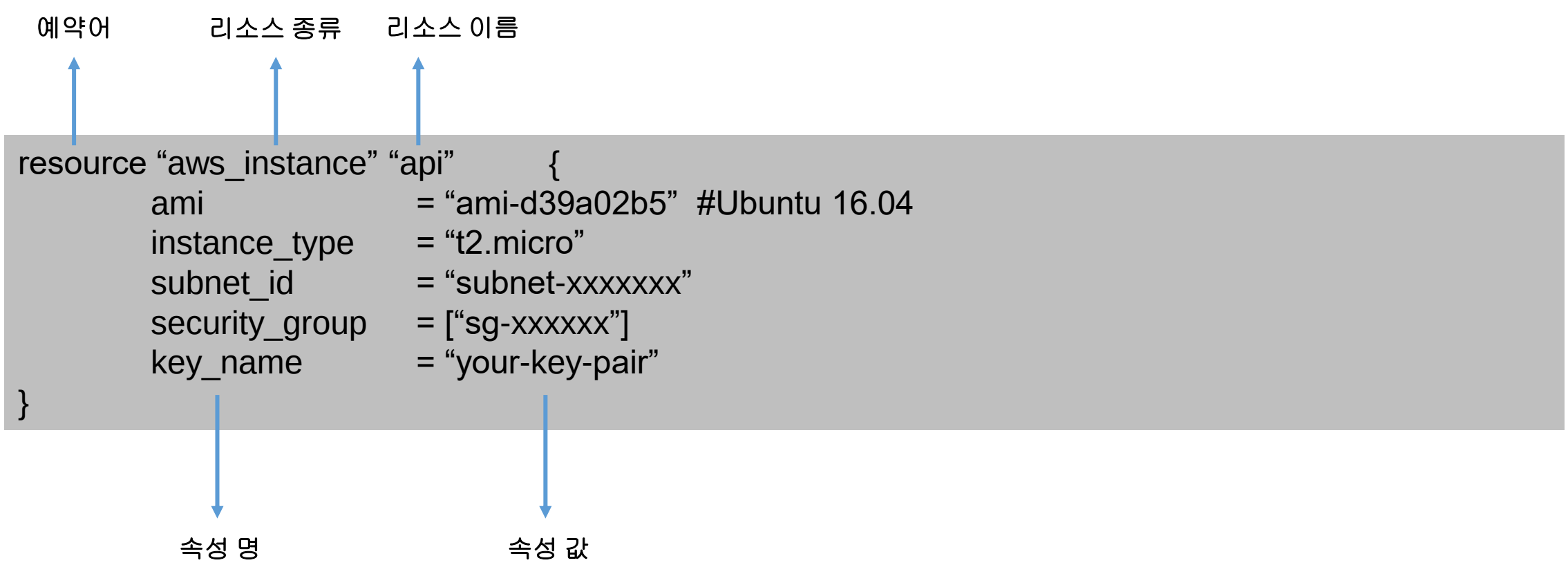
plan

테라폼을 통해 실제로 생성되고 변경되는 내역을 보여줌, 실제 환경에 적용하기 전에 검증할 수 있게 해 주는 수단. plan의 결과는 유닉스, 리눅스와 Git의 diff 명령어와 비슷함
(+) 기호는 생성한다는 의미이고, (-)기호는 제거한다는 의미

apply

destroy

Terraform 문법



Terraform Style

```
resource "azurerm_resource_group" "user01-rg" {
    name                = "user01resourcegroup"
    location            = "koreacentral"

    tags = {
        environment  = "Terraform Demo"
    }
}
```

```
resource "azurerm_resource_group" "user01-rg" {
name       = "user01resourcegroup"
location   = "koreacentral"
tags = {
environment  = "Terraform Demo"
}
}
```

- Indent two spaces
- Single meta-arguments first
- Block meta-arguments last
- Blank lines for clarity
- Group single arguments
- Think about readability
- Line up the equal signs

Terraform Output

```
Output “NAME” {

        value = VALUE
}
```

Terraform Variable

```
variable "NAME" {

        default = "VALUE"
}
```

<예제>

```
variable "resource_group" {

        default = "user01gresourcegroup"
}
```

```
resource "azurerm_resource_group" "user01-rg" {
    name     = var.resource_group
    location = "koreacentral"

    tags = {
        environment = "Terraform Demo"
    }
}
```

```
variable “user01region" {

        default = “koreacentral"
}
```

```
variable “vnetname" {

        default = “user01vnet"
}
```

```
resource "azurerm_resource_group" "user01-rg" {
    name     = var.resource_group
    location = var.user01region

    tags = {
        environment = "Terraform Demo"
    }
}
```

Terraform Graph

$terraform graph

```
davids-iMac:dev_vpc david$ terraform graph
digraph {
        compound = "true"
        newrank = "true"
        subgraph "root" {
                "[root] aws_default_network_acl.dev_default" [label = "aws_default_network_acl.dev_default", shape = "box"]
                "[root] aws_default_security_group.dev_default" [label = "aws_default_security_group.dev_default", shape = "box"]
                "[root] aws_eip.bastion_1a" [label = "aws_eip.bastion_1a", shape = "box"]
                "[root] aws_eip.nat_dev_1a" [label = "aws_eip.nat_dev_1a", shape = "box"]
                "[root] aws_eip.nat_dev_1c" [label = "aws_eip.nat_dev_1c", shape = "box"]
                "[root] aws_instance.bastion_1a" [label = "aws_instance.bastion_1a", shape = "box"]
                "[root] aws_internet_gateway.dev" [label = "aws_internet_gateway.dev", shape = "box"]
                "[root] aws_nat_gateway.dev_1a" [label = "aws_nat_gateway.dev_1a", shape = "box"]
                "[root] aws_nat_gateway.dev_1c" [label = "aws_nat_gateway.dev_1c", shape = "box"]
                "[root] aws_route_table.dev_private_1a" [label = "aws_route_table.dev_private_1a", shape = "box"]
                "[root] aws_route_table.dev_private_1c" [label = "aws_route_table.dev_private_1c", shape = "box"]
                "[root] aws_route_table.dev_public" [label = "aws_route_table.dev_public", shape = "box"]
```

DOT 라는 그래프 설명 언어로 결과 표시
GraphvizOnline (http://bit.ly/2mPbxmg) 에서 리소스간 의존성 확인

Terraform GraphvizOnline 결과

Not Secure | dreampuf.github.io/GraphvizOnline/#digraph%20%7B%0A%09compound%20%3D%20"true"%0A%09newrank%20%3D%20"true"%0A%09subgraph%20"root"%20%7B%0A%09%0...

Click to go back, hold to see history

```
    compound = "true"
    newrank = "true"
    subgraph "root" {
        "[root] aws_default_network_acl.dev_default" [label = "aws_default_network_acl.dev_default
        "[root] aws_default_security_group.dev_default" [label = "aws_default_security_group.dev_d
        "[root] aws_eip.bastion_1a" [label = "aws_eip.bastion_1a", shape = "box"]
        "[root] aws_eip.nat_dev_1a" [label = "aws_eip.nat_dev_1a", shape = "box"]
        "[root] aws_eip.nat_dev_1c" [label = "aws_eip.nat_dev_1c", shape = "box"]
        "[root] aws_instance.bastion_1a" [label = "aws_instance.bastion_1a", shape = "box"]
        "[root] aws_internet_gateway.dev" [label = "aws_internet_gateway.dev", shape = "box"]
        "[root] aws_nat_gateway.dev_1a" [label = "aws_nat_gateway.dev_1a", shape = "box"]
        "[root] aws_nat_gateway.dev_1c" [label = "aws_nat_gateway.dev_1c", shape = "box"]
        "[root] aws_route_table.dev_private_1a" [label = "aws_route_table.dev_private_1a", shape =
        "[root] aws_route_table.dev_private_1c" [label = "aws_route_table.dev_private_1c", shape =
        "[root] aws_route_table.dev_public" [label = "aws_route_table.dev_public", shape = "box"]
        "[root] aws_route_table_association.dev_private_1a" [label = "aws_route_table_association.
        "[root] aws_route_table_association.dev_private_1c" [label = "aws_route_table_association.
        "[root] aws_route_table_association.dev_public_1a" [label = "aws_route_table_association.d
        "[root] aws_route_table_association.dev_public_1c" [label = "aws_route_table_association.d
        "[root] aws_security_group.bastion" [label = "aws_security_group.bastion", shape = "box"]
        "[root] aws_subnet.private_1a" [label = "aws_subnet.private_1a", shape = "box"]
        "[root] aws_subnet.private_1c" [label = "aws_subnet.private_1c", shape = "box"]
        "[root] aws_subnet.public_1a" [label = "aws_subnet.public_1a", shape = "box"]
        "[root] aws_subnet.public_1c" [label = "aws_subnet.public_1c", shape = "box"]
        "[root] aws_vpc.dev" [label = "aws_vpc.dev", shape = "box"]
        "[root] provider.aws" [label = "provider.aws", shape = "diamond"]
        "[root] aws_default_network_acl.dev_default" -> "[root] aws_subnet.private_1a"
        "[root] aws_default_network_acl.dev_default" -> "[root] aws_subnet.private_1c"
        "[root] aws_default_network_acl.dev_default" -> "[root] aws_subnet.public_1a"
        "[root] aws_default_network_acl.dev_default" -> "[root] aws_subnet.public_1c"
        "[root] aws_default_security_group.dev_default" -> "[root] aws_vpc.dev"
        "[root] aws_eip.bastion_1a" -> "[root] aws_instance.bastion_1a"
        "[root] aws_eip.nat_dev_1a" -> "[root] provider.aws"
        "[root] aws_eip.nat_dev_1c" -> "[root] provider.aws"
        "[root] aws_instance.bastion_1a" -> "[root] aws_default_security_group.dev_default"
        "[root] aws_instance.bastion_1a" -> "[root] aws_security_group.bastion"
        "[root] aws_instance.bastion_1a" -> "[root] aws_subnet.public_1a"
        "[root] aws_instance.bastion_1a" -> "[root] var.amazon_linux"
        "[root] aws_instance.bastion_1a" -> "[root] var.dev_keyname"
        "[root] aws_internet_gateway.dev" -> "[root] aws_vpc.dev"
        "[root] aws_nat_gateway.dev_1a" -> "[root] aws_eip.nat_dev_1a"
        "[root] aws_nat_gateway.dev_1a" -> "[root] aws_subnet.public_1a"
        "[root] aws_nat_gateway.dev_1c" -> "[root] aws_eip.nat_dev_1c"
        "[root] aws_nat_gateway.dev_1c" -> "[root] aws_subnet.public_1c"
```

Engine: dot Format: svg Show raw output Share