

1. 제목 미정

2. EC2 인스턴스 생성

□ 목적

EC2 인스턴스를 생성하고 apache2를 설치하는 과정을 테라폼으로 작성한다.

□ 목표

- 테라폼으로 EC2 인스턴스를 생성할 수 있다.
- EC2인스턴스 원격접속을 위해 공개키 인증방식을 적용할 수 있다.
- 테라폼으로 EC2 인스턴스 user-data를 설정할 수 있다.

□ **aws_instance 리소스**

테라폼에서는 aws_instance 리소스를 이용하여 EC2 인스턴스를 생성한다. ami와 instance_type필드가 필수로 설정해야 한다.

- ami: aws ami ID
- instance_type: EC2 인스턴스 타입(예: t2.micro)

아래 예제는 타입이 t2.nano타입이고 AMI가 ubuntu18.04인 EC2 인스턴스를 생성하는 예제이다.

```
resource "aws_instance" "web" {  
  ami = "ami-0b50511490117e709" # ubuntu 18.04LTS  
  instance_type = "t2.nano"  
}
```

□ user data 설정

user data는 EC2 인스턴스가 생성되고 사용자가 입력한 스크립트를 실행하는 기능이다. 테라폼은 user data 작업을 실행하기 위해 EC2 인스턴스에 원격접속이 필요하다.

① key-pair 설정

공개키 인증방식을 사용하기 위한 설정과정을 설명한다.

공개키-비밀키 키쌍 생성

EC2 인스턴스에 등록할 공개키와 원격 접속에 사용하는 비밀키를 생성한다. -f인자에 생성할 키 이름을 설정한다. 아래 예는 공개키는 test.pub, 비밀키는 test이름으로 생성된다.

```
ssh-keygen -t rsa -b 4096 -N "" -f test
```

aws_key_pair 리소스 생성

생성한 키 쌍에서 공개키를 file함수를 사용하여 aws_key_pair 리소스를 등록한다.

```
resource "aws_key_pair" "demo-keypair" {
  key_name     = "deployer-key"
  public_key   = file("./test.pub")
}
```

aws_instance 리소스에 키쌍 등록

aws_key_pair 리소스 key_name필드를 활용하여 EC2 인스턴스에 공개키를 등록한다.

```
resource "aws_instance" "web" {
  ami = "ami-0b50511490117e709"
  instance_type = "t2.nano"
  key_name = aws_key_pair.demo-keypair.key_name
  ...
}
```

② user data 스크립트 설정

EC2 인스턴스 원격접속 후에 스크립트를 실행하는 내용을 설정한다. 스크립트 실행은 EC2 인스턴스 생성 후에 실행되어야 하기 때문에 실행 순서 의존성을 설정한다.

의존성 설정

EC2 인스턴스 생성 전에 스크립트 실행을 한다면 오류가 발생할 수 있으므로 EC2 인스턴스 생성 후에 스크립트를 실행한다. 의존성 설정을 위해 빈 리소스(null_resource)를 생성한다.

```
resource "null_resource" "name" {
  depends_on = [aws_instance.web]
  ...
}
```

EC2 인스턴스 원격접속

connection 리소스를 사용해서 EC2 인스턴스 ssh 원격접속 설정을 한다. 인증방법은 등록된 공개 키에 해당하는 비밀키를 사용한다.

```
connection {
  type = "ssh"
  user = "ubuntu"
  private_key = file("./test")
  host = aws_instance.web.public_ip
  timeout = "1m"
}
```

스크립트 설정

provisioner 리소스의 remote-exec를 사용하여 스크립트를 실행한다. 각 스크립트는 쉼표(,)로 구분한다.

```
provisioner "remote-exec" {
  inline = [
    "sudo apt update",
    "sudo apt install apache2 -y && sudo systemctl start apache2",
    "sudo chown -R ubuntu:ubuntu /var/www/html",
    "echo 'hello world' > /var/www/html/index.html"
  ]
}
```

□ 태그설정(옵션)

EC2 인스턴스를 구분하기 태그를 지정한다. 대표적으로 EC2인스턴스 이름인 Name필드를 입력한다. 아래 예제는 EC2 인스턴스가 생성되면 demo-instance이름이 설정된다.

```
resource "aws_instance" "web" {
  ami = "ami-0b50511490117e709"
  instance_type = "t2.nano"
  key_name = aws_key_pair.demo-keypair.key_name
```

```
...
tags = {
    Name = "demo-instance"
}
}
```

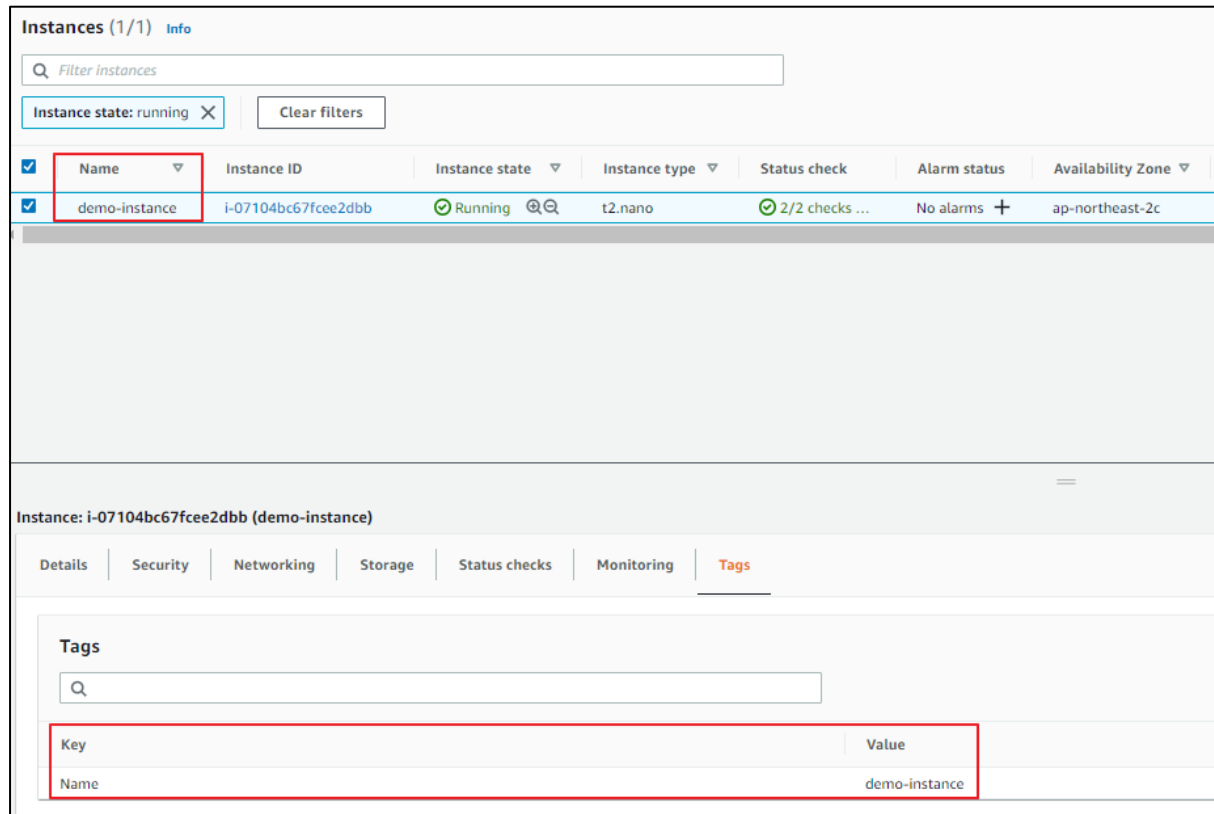


그림 1 EC2 Instance 태그 Name설정 예

□ user data 실행: 초기화 스크립트

provisioners "remote-exec" 리소스를 사용하여 스크립트를 실행할 수 있다. apt-get install 등 리눅스 패키지로 다운로드 가능하다.

③ 테라폼 실행 중 원격접속

remote-exec 리소스를 사용하기 위해서 원격접속이 필요하다. aws_key_pair 리소스를 사용해서 EC2 instance에 공개키를 등록한 후 비밀키를 사용하여 원격접속 한다. 비밀키는 file함수를 사용해서 로드한다.

```

connection {
  type = "ssh"
  user = "ubuntu"
  private_key = file("./test") # 비밀키
  host = self.public_ip
  timeout = "1m"
}

```

④ 초기화 스크립트 실행

remote-exec리소스를 사용하여 초기화 스크립트를 실행한다. 원격 접속 계정은 일반 권한을 갖고 있으므로 sudo 명령어를 사용해야 한다.

```

provisioner "remote-exec" {
  inline = [
    "apt update",
    "sudo apt install apache2 -y && sudo systemctl start apache2",
    "sudo echo 'hello world' > /var/www/html/index.html"
  ]
}

```

□ 최종코드

```

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

resource "aws_key_pair" "demo-keypair" {
  key_name     = "deployer-key"
  public_key = file("./test.pub")
}

resource "aws_instance" "web" {
  ami = "ami-0b50511490117e709"
  instance_type = "t2.nano"
}

```

```
key_name = aws_key_pair.demo-keypair.key_name

connection {
  type = "ssh"
  user = "ubuntu"
  private_key = file("./test")
  host = self.public_ip
  timeout = "1m"
}

provisioner "remote-exec" {
  inline = [
    "sudo su",
    "sudo apt update",
    "sudo apt install apache2 -y && sudo systemctl start apache2",
    "sudo echo 'hello world' > /var/www/html/index.html"
  ]
}

tags = {
  Name = "demo-instance"
}
}
```

3. 참고자료

블로그-terraform provisioners apt command: <https://blog.outsider.ne.kr/1344>

공식문서-terraform provisioners manual

공식문서-terraform

aws-instance

manual:

<https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/instance>