

Sử dụng Terraform tạo Route 53 Records (Alias) để định tuyến traffic truy cập vào ALB

Giới thiệu

Trong bài thực hành này, bạn sẽ tạo một Route 53 alias record để định tuyến các traffic truy cập từ publicly hosted zone trong Route 53 (môi trường của bài thực hành này đã được tạo sử dụng file Terraform-Lab2.json) tới một cân bằng tải ALB sử dụng Terraform templates.

✓ Chúng ta click vào đường link :

<https://raw.githubusercontent.com/phuongluuho/TerraformAnsible/main/Terraform-Lab2.json>

Sau đó sử dụng CloudFormation Stack. **Để tạo ra một môi trường cho bài thực hành gồm:** 01 máy ảo (EC2 instance & installed Terraform), 02 máy ảo (EC2 instance & Installed Apache webserver), SecurityGroups, 01 VPC & Subnets, Internet Gateway, 01 EC2InstanceProfile.

Giải pháp

Log in vào Linux Terraform Controller Node của bạn

```
ssh cloud_user@<PUBLIC_IP_ADDRESS>
```

Note: Khi bạn copying và pasting code bên trong Vim từ lab guide, đầu tiên enter `:set paste` (và sau đó nhấn phím `i` để enter vào chế độ soạn thảo insert mode) để tránh thêm vào các khoảng trắng và dấu thăng không cần thiết. Để lưu lại nội dung và thoát ra khỏi file, nhấn **Escape** tiếp theo là `:wq`. để thoát khỏi the file mà không lưu lại nội dung, nhấn **Escape** tiếp theo là `:q!`.

Log in vào Linux Terraform Controller Node EC2 Instance

1. Xác nhận Terraform version đã được cài đặt là 1.8.4:

```
terraform version
```

Clone the GitHub Repo cho Terraform Code

1. Sử dụng lệnh `git` để clone GitHub repo:

```
git clone https://github.com/phuongluuho/terraform-labs.git
```

2. Xem `resource_ids.txt`:

```
cat resource_ids.txt
```

3. Copy tất cả các giá trị của file trên vào notepad file để sử dụng cho bước tiếp theo trong lab này.
4. Chuyển vào thư mục Terraform code của lab này :

```
cd content-deploying-to-aws-ansible-terraform/lab_deploying_dns_acm
```

5. Liệt kê nội dung của thư mục:

```
ls
```

Paste vào các giá trị của các tài nguyên được cung cấp vào `import_resources.tf`

1. Edit `import_resources.tf` file và paste vào file này các giá trị đã được copy vào notepad của bạn trong bước trước từ `resource_ids.txt` file:

```
vim import_resources.tf
```

2. Sử dụng các giá trị đã được copy từ `resource_ids.txt` file, thay đổi "<INSERT-VALUE-HERE>" text cho mỗi một giá trị tương ứng.
3. Lưu lại nội dung và thoát ra khỏi file:

```
:wq
```

Lấy Public Hosted Route 53 Zone và nhập nó vào trong `variables.tf`

1. Sử dụng AWS CLI call:

```
aws route53 list-hosted-zones
```

2. Copy tên DNS, đảm bảo phải copy bao gồm cả dấu chấm . tại cuối cùng của tên miền.
3. Edit the `variables.tf` file và nhập vào tên DNS đã được copy:

```
vim variables.tf
```

4. Sử dụng tên DNS vừa được copy, thay đổi "<insert-domain-name-ending-with-dot>" text.
5. Lưu lại nội dung và thoát ra khỏi file:

```
:wq
```

Triển khai (Deploy) Terraform Code

1. Khởi tạo thư mục Terraform:

```
terraform init
```

2. Đảm bảo Terraform code đúng với định dạng chuẩn:

```
terraform fmt
```

3. Đảm bảo code đúng cú pháp và không có lỗi gì:

```
terraform validate
```

4. Xem kế hoạch thực thi và note số resources sẽ được tạo ra:

```
terraform plan
```

5. Triển khai resources (*NOTE: Sau khi `terraform apply` đã chạy thành công, bạn có thể sử dụng AWS CLI trong Linux Controller node để liệt kê các tài nguyên (resources) hoặc bạn có thể log in vào AWS Console để xác nhận và kiểm tra các resources đã được tạo ra.*):

```
terraform apply
```

6. Enter `yes` when prompted.

7. Copy the URL và paste nó vào trong trình duyệt web của bạn để test ứng dụng (application).

8. Xóa tất cả các resources đã được tạo ra:

```
terraform destroy
```

Conclusion

Chúc mừng bạn đã hoàn thành bài hands-on lab này!