



Devops on *AWS* for beginner

Instructor – Linh Nguyen

Engineering Consultant, AWS Cloud Solution Architect

Infra as Code (IaC) sử dụng Terraform

“Không có việc gì khó, chỉ sợ không biết làm!”

Copyright@Linh Nguyen on Udemyl
All right reserved

Mục tiêu

- Hiểu được khai niệm Infra as Code (IaC) là gì?
- Tại sao lại cần sử dụng IaC?
- Các công cụ IaC thường được sử dụng.
- Giới thiệu về Terraform.
- Các thành phần và cơ chế hoạt động của Terraform.
- Cài đặt các công cụ cần thiết để làm việc với Terraform.
- Quản lý resource và version như thế nào?
- Làm quen với Terraform thông qua các bài lab từ dễ đến khó.

Copyright@Linh Nguyen on Udemy
All right reserved

Infra as Code (IaC) là gì?

Infra as Code (IAC), hay còn được biết đến với tên gọi "Infrastructure as Code", là một phương pháp quản lý và cấu hình cơ sở hạ tầng công nghệ thông tin (IT) tự động hóa thông qua mã nguồn (code). Thay vì thực hiện cấu hình và quản lý cơ sở hạ tầng một cách thủ công và trực tiếp, người ta có thể sử dụng mã nguồn (script, template) để tự động hóa các công việc này.

IaC ra đời để giải quyết các vấn đề của phương pháp triển khai Infra truyền thống:

- Tốc độ triển khai chậm, đặc biệt khi hệ thống có nhiều môi trường, regions.
- Không nhất quán, có thể có sai sót do trình độ người thực thi.
- Không thể tái sử dụng.

Copyright@Linh Nguyen on Udemy
All right reserved

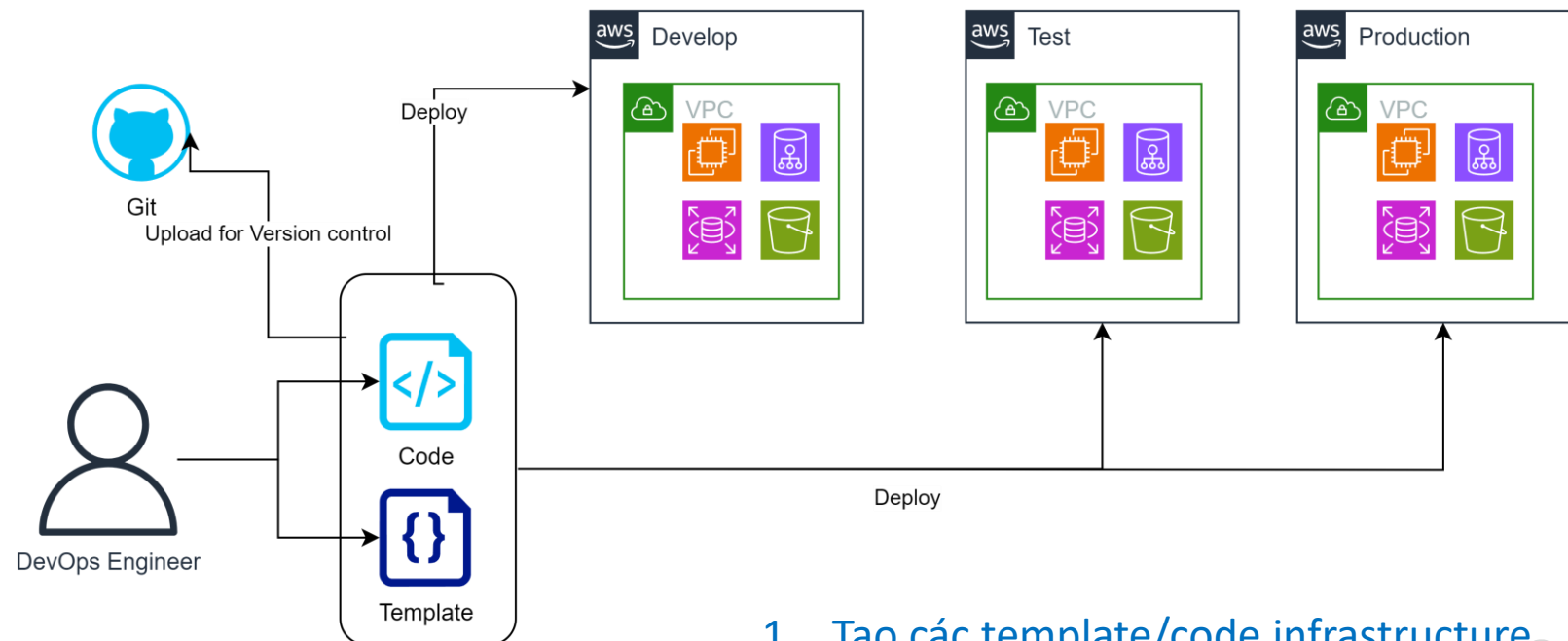
Infra as Code (IaC) là gì?

Ưu điểm khi triển khai Infrastructure as Code

- **Tự động hóa:** giúp giảm thiểu rủi ro và tăng tốc độ triển khai.
- **Nhất quán và Dễ dàng kiểm soát:**
 - Đảm bảo rằng môi trường được cấu hình một cách nhất quán và đúng đắn.
 - Kiểm soát phiên bản cho phép quản lý, theo dõi và phục hồi phiên bản cấu hình.
- **Tích hợp và Phối hợp:**
 - **Tích hợp với DevOps** tạo điều kiện cho việc phát triển và vận hành liền mạch.
 - **Phối hợp với các công cụ khác:** Dễ dàng tích hợp và phối hợp với các công cụ và dịch vụ khác như CI/CD, monitor, và log.
- **Tối ưu hoá tài nguyên và chi phí:** Thông qua việc tái sử dụng, giảm bớt công sức và thời gian của đội ngũ kỹ sư và nhân viên IT, giúp tiết kiệm chi phí.
- **An toàn và Bảo mật:**
 - **Quản lý quyền truy cập:** Dễ dàng quản lý và kiểm soát quyền truy cập.
 - **Tích hợp bảo mật:** Có thể tích hợp các chuẩn và quy tắc bảo mật vào trong quá trình cấu hình và triển khai.

Copyright@Linh Nguyen on Udemy
All right reserved

Các bước triển khai infra as code



1. Tạo các template/code infrastructure
2. Kiểm thử trên môi trường develop
3. Upload lên Version control để tracking
4. Triển khai hàng loạt
5. Update/Maintenance infrastructure

Copyright @ Linh Nguyen on Udemy
All right reserved

Tại sao sử dụng Infra as Code?

Câu hỏi 1: Tôi (KH của tôi) chỉ có nhu cầu triển khai 1 môi trường, vậy có cần sử dụng Infrastructure as Code (IaC) không?

Trả lời: Trừ khi thời gian quá gấp rút, tác vụ quá đơn giản và bạn không muốn tốn nhiều thời gian vào viết template/code, bạn vẫn nên sử dụng IaC. Những infrastructure template có thể không được tái sử dụng ngay trong project nhưng về lâu dài rất có lợi cho bạn và công ty.

Copyright@Linh Nguyen on Udemy
All right reserved

Các công cụ Infra as Code thường dùng

CloudFormation

- **Provider:** Amazon Web Services (AWS)
- **Mô tả:** AWS CloudFormation là một dịch vụ của AWS cho phép người dùng định nghĩa và triển khai cấu trúc và tài nguyên dịch vụ dưới dạng template (Yaml hoặc Json)

*Do là service native của AWS nên không hỗ trợ cloud khác.



AWS CloudFormation

Copyright@Linh Nguyen on Udemy
All right reserved

Các công cụ Infra as Code thường dùng

AWS Cloud Development Kit (CDK)

- **Provider:** AWS
- **Mô tả:** một công cụ do AWS phát triển, cho phép bạn định nghĩa **cloud infrastructure** bằng cách sử dụng các ngôn ngữ lập trình phổ biến như TypeScript, JavaScript, Python, C#, và Java.
- Phù hợp với những người đã quen với việc lập trình (code) và muốn tận dụng để provision resource một cách tự động.
 - *Chỉ hỗ trợ AWS.
 - *Khi sử dụng CDK, sẽ có 1 phase code được compile thành các CloudFormation template nên về cơ bản phía sau của CDK vẫn là các **CloudFormation** stack.



AWS Cloud Development Kit
(AWS CDK)

Copyright@Linh Nguyen on Udemy
All right reserved

Các công cụ Infra as Code thường dùng

Terraform

- **Provider:** HashiCorp
- **Mô tả:** Terraform là một công cụ IaC mã nguồn mở cho phép người dùng định nghĩa và cung cấp cơ sở hạ tầng dưới dạng mã. Terraform rất linh hoạt và có thể tương tác với nhiều dịch vụ và nhà cung cấp khác nhau (AWS, GCP, Azure).

*Điểm khác biệt cơ bản lớn nhất giữa Terraform và CloudFormation là nó không tạo ra các CloudFormation stack trong quá trình deploy. Devops có thể giữ template cho riêng mình và không release (black-box).



Copyright@Linh Nguyen on Udemy
All right reserved

Các công cụ Infra as Code thường dùng

Azure Resource Manager (ARM) Templates

- **Provider:** Microsoft
- **Mô tả:** ARM Templates cho phép người dùng định nghĩa và cấu hình cơ sở hạ tầng Azure của mình thông qua các mẫu JSON.



Copyright@Linh Nguyen on Udemy
All right reserved

Các công cụ Infra as Code thường dùng

Google Cloud Deployment Manager

- **Provider:** Google Cloud
- **Mô tả:** Dịch vụ này giúp người dùng tự động hóa việc tạo và quản lý tài nguyên trên Google Cloud Platform thông qua các file cấu hình YAML.



Copyright@Linh Nguyen on Udemy
All right reserved

Các công cụ Infra as Code thường dùng

Các công cụ khác



- Mô hình agentless, chỉ cần cài đặt Ansible trên máy quản lý.
- Sử dụng SSH để kết nối và quản lý các máy chủ.
- Sử dụng YAML để viết playbook và roles.



- Mô hình agent-master, cần cài đặt agent trên các máy chủ được quản lý.
- Sử dụng SSL để kết nối và quản lý các máy chủ.
- Sử dụng DSL (Domain Specific Language) riêng của Puppet và cũng hỗ trợ Ruby.



- Mô hình client-server, cần cài đặt client trên các máy chủ được quản lý.
- Sử dụng SSL để kết nối và quản lý các máy chủ.
- Sử dụng Ruby DSL để viết recipes và cookbooks.

Copyright@Linh Nguyen on Udemy
All right reserved

Lựa chọn công cụ IaC nào là tốt nhất?

Câu hỏi 2: Trong các tool IaC tool nào là tốt nhất?

Trả lời: Các tool IaC được thiết kế đa dạng mục đích và có ưu nhược điểm riêng. Tùy vào nhu cầu dự án và khả năng của member mà lựa chọn công nghệ phù hợp để mang lại hiệu quả cao nhất.

Vd: những người đi lên từ developer thích dùng **CDK**, những người đi lên từ system admin thích dùng **Terraform**.

Copyright@Linh Nguyen on Udemy
All right reserved

Giới thiệu Terraform

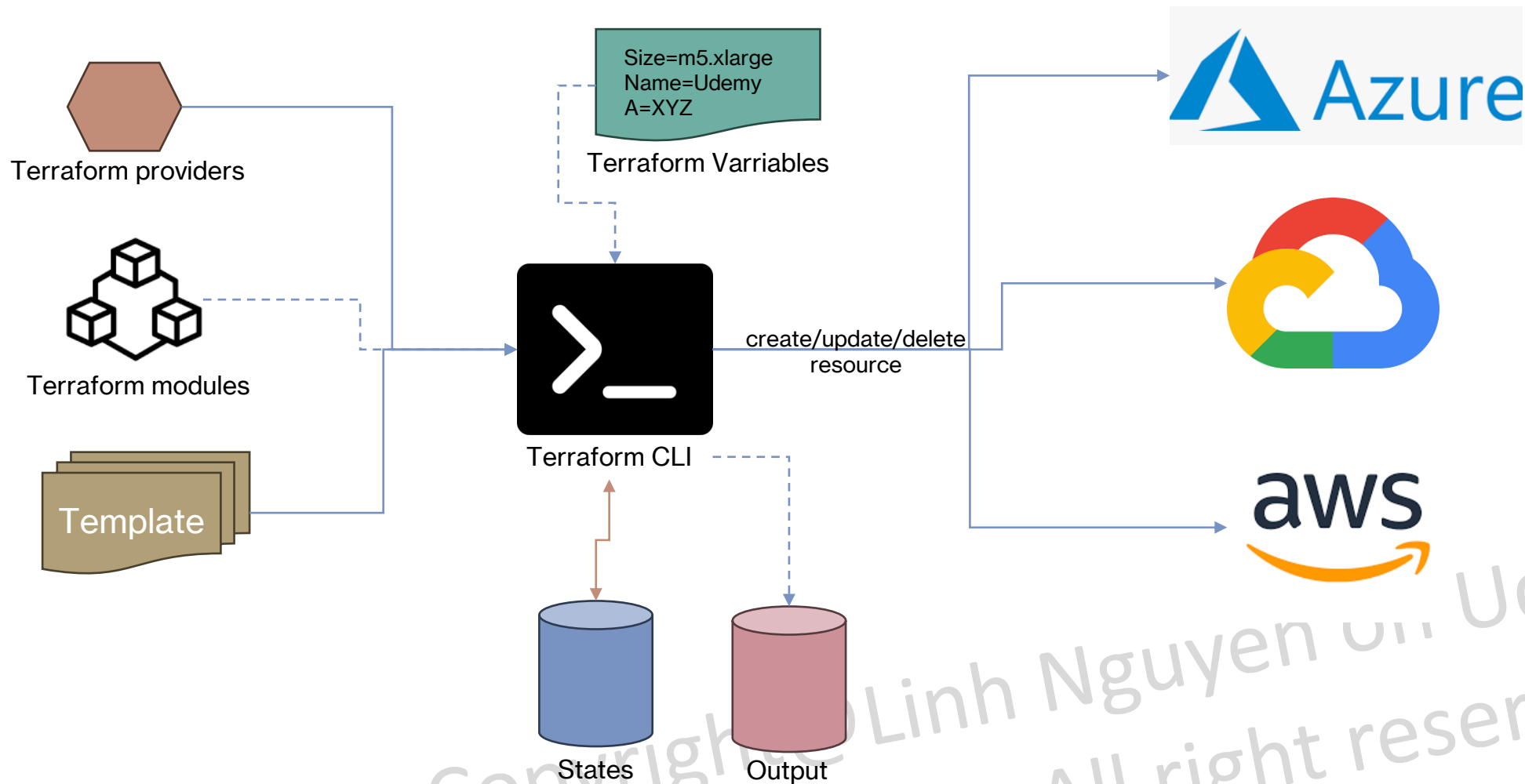
- **Provider:** HashiCorp
- **Ngôn ngữ:** HashiCorp Configuration Language (HCL).
- **Hỗ trợ cloud:** AWS, GCP, Azure,...



Copyright@Linh Nguyen on Udemy
All right reserved

Giới thiệu Terraform

Cơ chế hoạt động của Terraform



Giới thiệu Terraform

Các thành phần cơ bản của Terraform

- 1. Template Files (.tf):** Các tệp cấu hình được viết bằng HCL, mô tả cơ sở hạ tầng mong muốn.
- 2. Providers:** Providers là plugins được Terraform sử dụng để tương tác với các APIs của các dịch vụ đám mây, dịch vụ nền tảng hoặc sản phẩm phần mềm.
- 3. Resources:** Resources là các thành phần cơ bản của cơ sở hạ tầng, bao gồm, nhưng không giới hạn ở, máy ảo, mạng, và dịch vụ.
- 4. Terraform CLI:** CLI của Terraform cung cấp các lệnh để quản lý và kiểm soát các hành động của Terraform, bao gồm khởi tạo, áp dụng, và hủy bỏ cơ sở hạ tầng.
- 5. State Files:** Terraform dùng các tệp trạng thái để lưu trữ thông tin về cơ sở hạ tầng đã được tạo và quản lý.
- 6. Variables and Outputs:** Variables được sử dụng để tùy chỉnh cấu hình, và outputs được sử dụng để lấy giá trị từ cơ sở hạ tầng đã được tạo.

Copyright@Linh Nguyen on Udemy
All right reserved

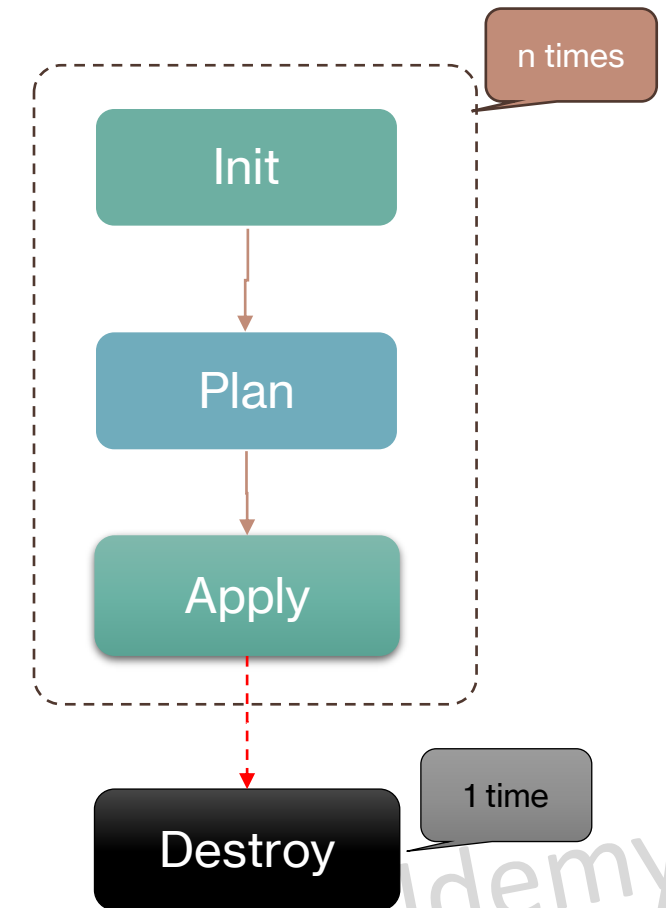
Giới thiệu Terraform

Các bước để triển khai Terraform

1. **Initialize (terraform init):** Chuẩn bị môi trường làm việc bằng cách khởi tạo Terraform.
2. **Planning (terraform plan):** Xác định & lên kế hoạch những gì sẽ được tạo, thay đổi, hoặc xóa bỏ.
3. **Apply (terraform apply):** Áp dụng các thay đổi để tạo hoặc chỉnh sửa Infrastructure.
4. **Destroy (terraform destroy):** Hủy bỏ và xóa bỏ Infrastructure.

*Lưu ý (1): trong phase “**apply**”, tùy vào sự thay đổi của template mà resource có thể được **thêm vào**, bị **chỉnh sửa** hoặc **xóa** đi.

*Lưu ý (2): thao tác “**destroy**” sẽ xóa toàn bộ resource, thường chỉ áp dụng khi muốn xóa toàn bộ môi trường (lab/dev/test).



Copyright @ Linh Nguyen on Udemy
All right reserved

Lab 1 – Cài đặt Terraform trên máy local.

1. Download và cài đặt AWS CLI, kiểm tra version.
2. Download và cài đặt Terraform, kiểm tra version.
3. Cài đặt Terraform plugin cho VS Code.
4. Giới thiệu document chính thức của Terraform, cách tra cứu.

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 2 – Tạo một template Terraform đơn giản và deploy lên AWS.

1. Chuẩn bị 1 keypair (1 private + 1 public key)
2. Tạo một Terraform template với resource là 1 EC2 instance, 1 Security Group với cấu hình cơ bản.
*Lưu ý: Account phải có **default VPC**.
3. Giới thiệu các step của Terraform.
4. Giới thiệu các thành phần của Terraform.
5. Thực hành thay đổi thông số resource (vd instance size)
6. Apply thay đổi.
7. Destroy resources.

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 3 – Làm quen với Parameter, Mapping, Variable file.

1. Giới thiệu về Parameter, Mapping.
2. Cách chỉ định parameter trong câu lệnh Terraform.
3. Cách sử dụng Variable file.
4. Deploy 1 resource đơn giản sử dụng Parameter, Mapping, Variable file.

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 4 – Terraform Output

1. Giới thiệu về Terraform Output.
2. Thực hành deploy 1 Terraform có sử dụng output.

Copyright@Linh Nguyen on Udemy
All right reserved

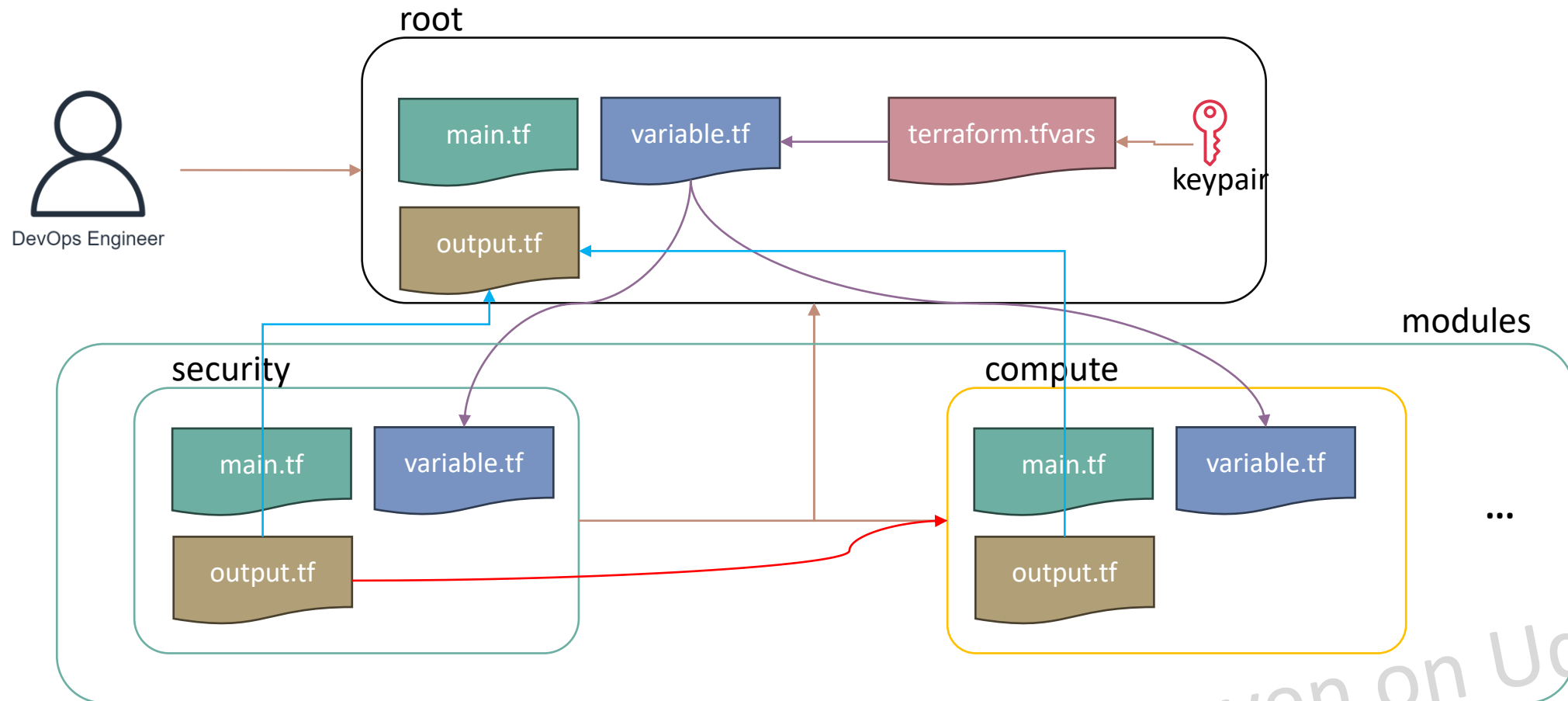
Lab 5 –Sử dụng Terraform Output để liên kết sub-modules

1. Giới thiệu về cách phân chia module, tổ chức thư mục.
2. Phân chia resource ở bài lab trước thành 2 module là **security** và **compute**.
3. Sử dụng Output làm liên kết giữa 2 module.
4. Deploy AWS resources sử dụng modules

*Lưu ý: bài lab này vẫn sử dụng **default -vpc**

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 5 – Terraform Output – Sử dụng output để liên kết sub-modules



Copyright@Linh Nguyen on Udemy
All right reserved

Lab 6 – Thực hành deploy 1 VPC hoàn chỉnh + resources

1. Thực hành tạo và deploy 1 VPC hoàn chỉnh với các thành phần:

- 2 public subnet, 2 private subnet
- Public route table, Private route table
- 1 Internet Gateway, 1 NAT Gateway
- Kiểm tra resource được tạo ra.

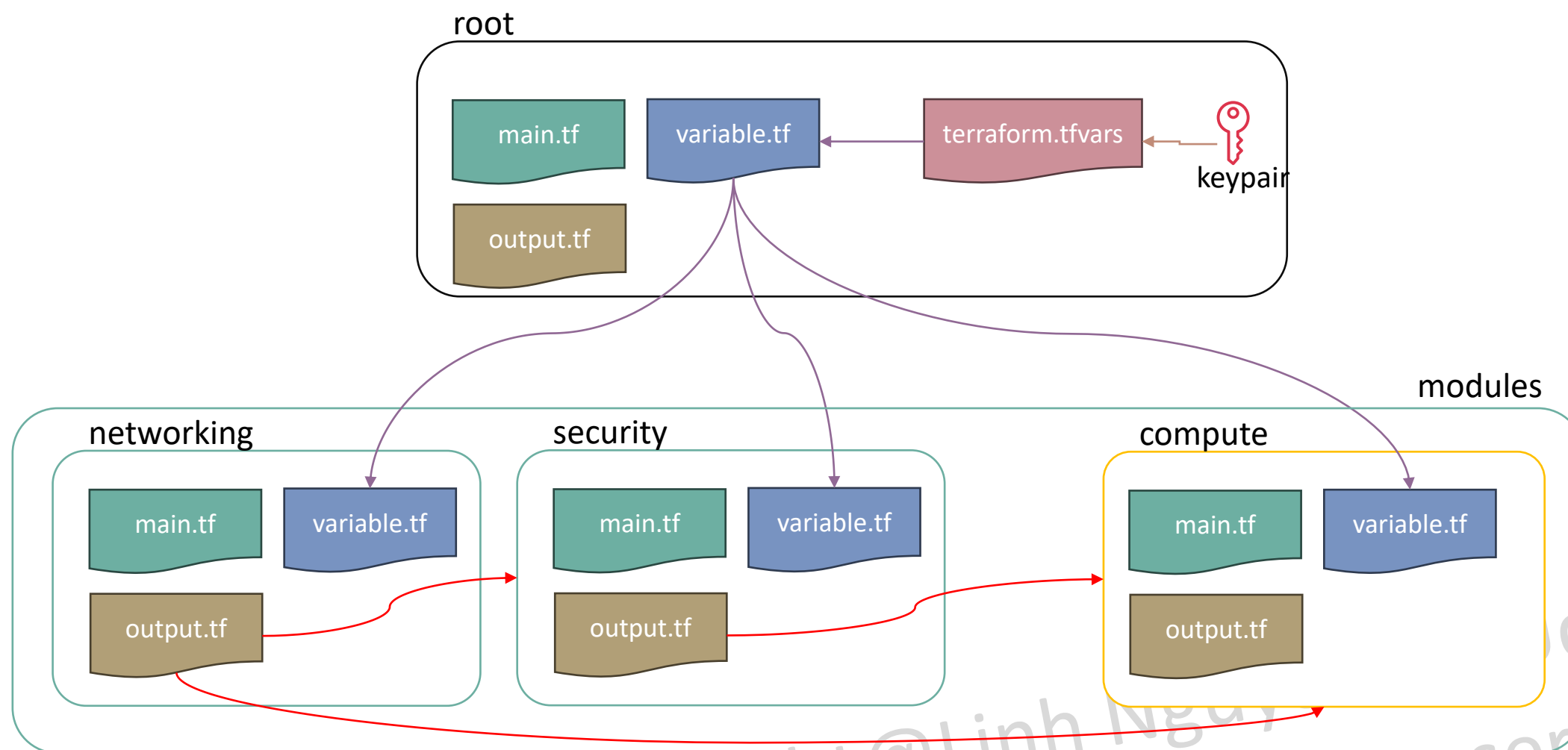
**Lưu ý sử dụng module để tách thành 1 module networking riêng*

2. Deploy thêm **security** + **compute** module sau khi module **networking** deploy thành công.

- Security Group nhận vpc id lấy từ module **networking** output
- EC2 nhận Security Group ID lấy từ module **security** output

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 6 – Thực hành deploy 1 VPC hoàn chỉnh



Copyright @ Linh Nguyen
All right reserved

Lab 7 – Sử dụng module có sẵn của Terraform.

1. Giới thiệu về Terraform module (built-in).
2. Thực hành sử dụng 1 số built-in module cơ bản (vd VPC kèm subnets).
3. Deploy resource lên AWS sử dụng Terraform built-in module.

Copyright@Linh Nguyen on Udemy
All right reserved

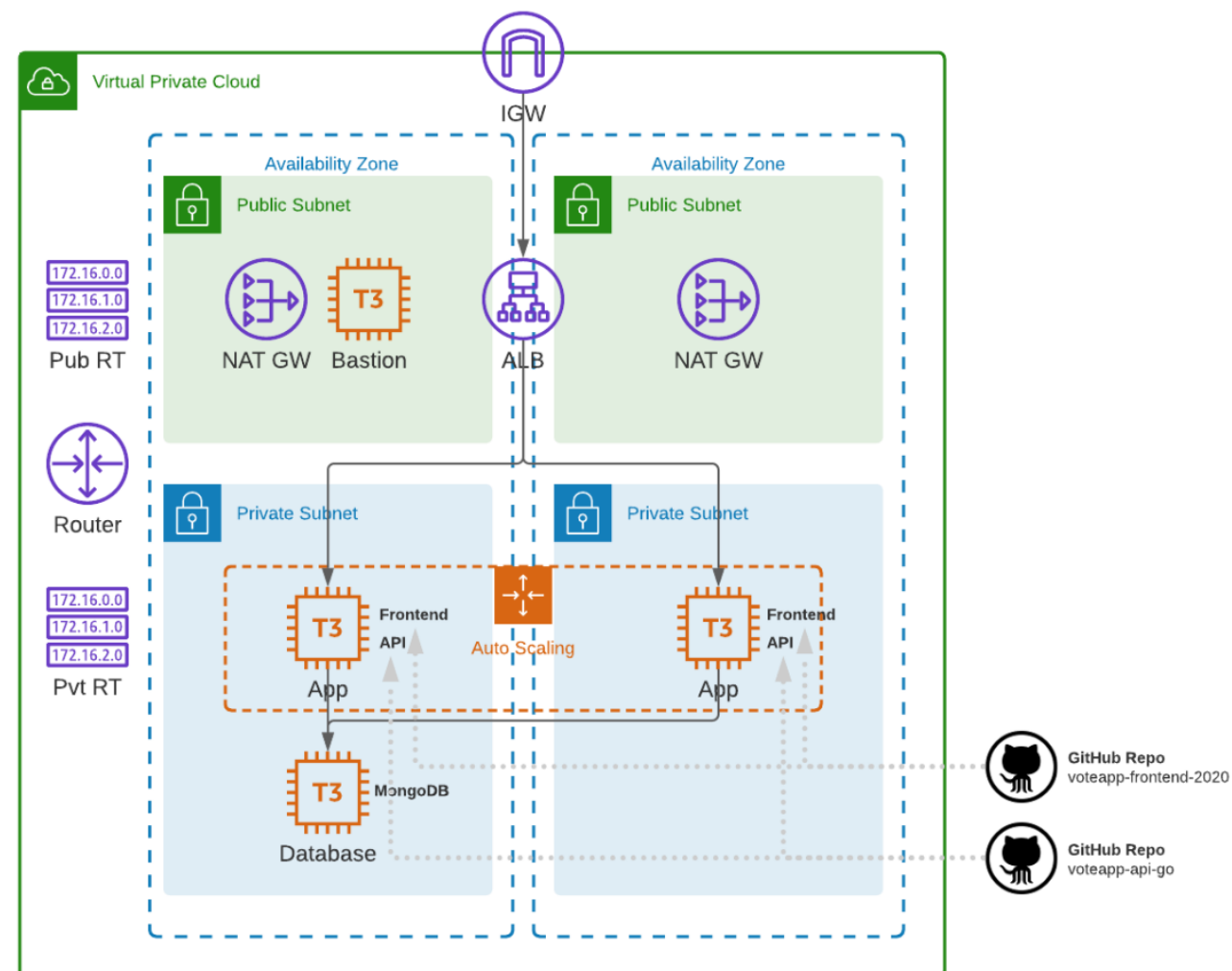
Lab 8 – Bài tập cuối chương.

1. Phân tích & tạo một cấu trúc Terraform có phân chia module.
2. Thực hành implement Terraform template theo thiết kế có sẵn.
3. Deploy một stack cơ bản lên AWS bao gồm
 - VPC, Subnet, Security Group,
 - EC2, Application Load Balancer, Target Group, Autoscaling Group.
 - MongoDB database

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 8 – Bài tập cuối chương.

Sơ đồ hệ thống của đề bài.



Copyright @ Linh Nguyen All right reserved

Các phương án quản lý state của Terraform

1. Giới thiệu về State trong Terraform
2. Quản lý State sử dụng Github.
3. Quản lý State sử dụng S3 Bucket.

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 9: Sử dụng S3 để quản lý state của Terraform

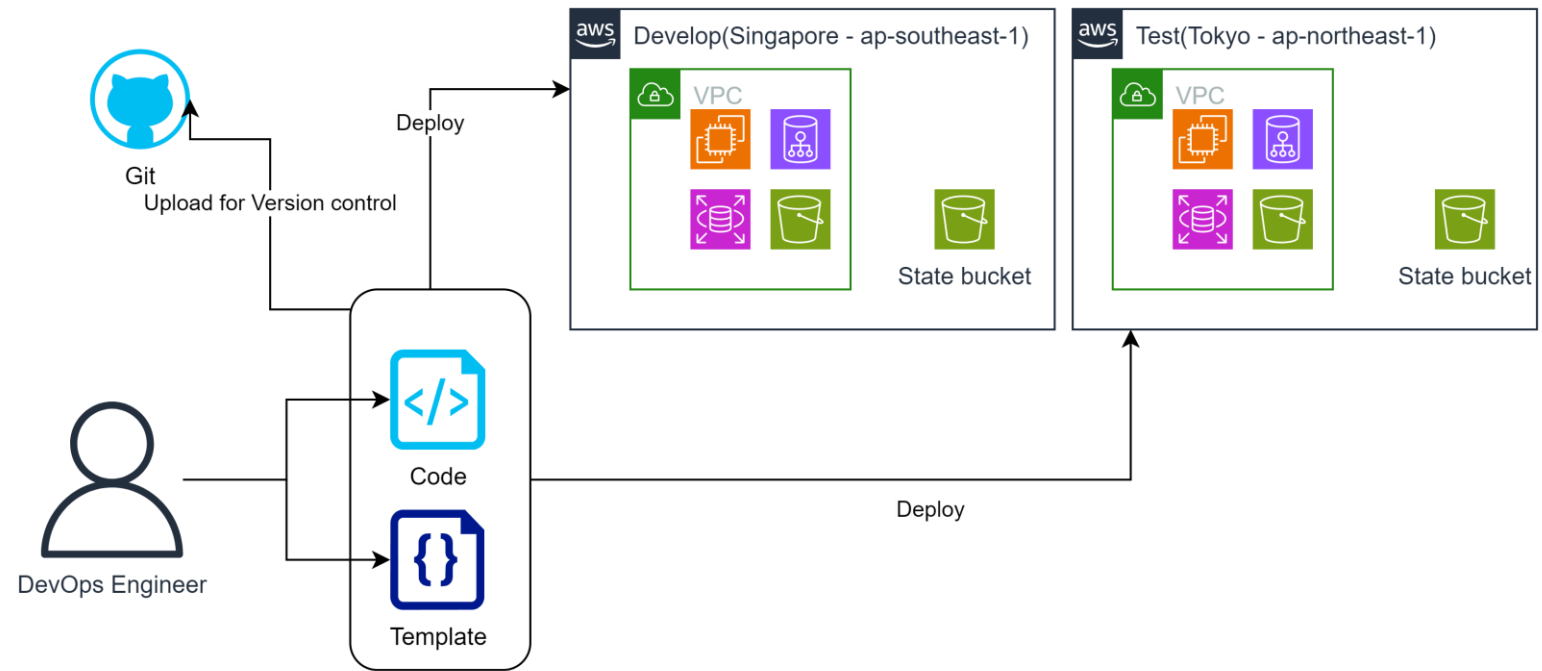
1. Chuẩn bị 1 S3 bucket.
2. Tạo 1 terraform template với cấu hình chỉ định S3 làm nơi lưu state.
3. Deploy terraform template.
4. Kiểm tra resource được tạo ra và trạng thái của state S3 bucket.

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 10 – Thực hành triển khai multi environment

Sơ đồ hệ thống

Yêu cầu: Triển khai hệ thống lên 2 môi trường dev, test nằm ở 2 region khác nhau, chỉ sử dụng một bộ modules chung, không duplicate code.



Copyright@Linh Nguyen on Udemy
All right reserved

Tổng kết & Clear resources

Nội dung đã học:

- IaC là gì? Công dụng của IaC.
- Cách setup môi trường làm việc Terraform.
- Cách thức hoạt động và các thành phần cơ bản của Terraform.
- Các bước để triển khai Infrastructure lên AWS sử dụng Terraform.
- Vận dụng linh hoạt Terraform để triển khai Infrastructure theo System Design.
- Biết cách quy hoạch Terraform theo module để tái sử dụng cũng như triển khai Multi-environment, Multi-region.

Copyright@Linh Nguyen on Udemy
All right reserved

Thanks you and see you in the next chapter!

Copyright@Linh Nguyen on Udemy
All right reserved