

Khóa học Lãnh đạo Cloud (The Cloud Leader Course)

Từ con số không đến việc lãnh đạo một đội ngũ kỹ sư cloud — soạn riêng cho Fedly Bonneau

Ngày 13 tháng 8 năm 2026

Khóa học Lãnh đạo Cloud (The Cloud Leader Course)

Từ chỗ chưa biết gì đến việc lãnh đạo trôi chảy một đội ngũ kỹ sư cloud.

Khóa học này vận hành như thế nào

Bạn không học để *trở thành* một kỹ sư cloud. Bạn học để **lãnh đạo** các kỹ sư cloud: theo kịp mọi cuộc trò chuyện trong phòng họp, đặt những câu hỏi thực sự quan trọng, ra quyết định đúng đắn về tiền bạc, rủi ro và con người, và giành được sự tôn trọng của các chuyên gia mà không cần giả vờ mình là một người trong số họ. Đó là một kỹ năng khác — hoàn toàn có thể học được — và chính là kỹ năng mà khóa học này truyền dạy.

Khóa học gồm ba giai đoạn. **Giai đoạn 1 (Tuần 1–8): Nói được ngôn ngữ.** Bạn học cloud (điện toán đám mây) thực chất là gì và từ vựng của mọi khối xây dựng cốt lõi, để các cuộc họp không còn nghe như tiếng ồn. **Giai đoạn 2 (Tuần 9–16): Tư duy như cả phòng họp.** Kiến trúc, bảo mật và tiền bạc — ba cuộc trò chuyện mà mọi đội cloud đều có hằng tuần, và cũng là ba nơi mà một người lãnh đạo hoặc tạo ra giá trị, hoặc bị phớt lờ. **Giai đoạn 3 (Tháng 5–24): Lãnh đạo.** Chứng chỉ, tuyển dụng, điều hành các cuộc họp ra quyết định, và con đường hai năm trung thực để có thể đứng vững trước các kỹ sư senior.

Quy tắc cho toàn khóa học: học **một giờ mỗi ngày, sáu ngày mỗi tuần** — sự đều đặn thắng sự dồn dập. Mỗi module kết thúc bằng một **Fluency Drill** (bài luyện nói trôi chảy — những câu bạn nói thành tiếng cho đến khi chúng trở nên tự nhiên) và một **Milestone** (cột mốc — bằng chứng cho thấy bạn đã sẵn sàng đi tiếp). Hãy làm chúng; chỉ đọc thôi thì tạo ra sự nhận biết, chứ không tạo ra sự trôi chảy. Và ngay từ Tuần 1, hãy duy trì một **Decision Journal** (nhật ký quyết định): mỗi khi học một khái niệm, viết một câu về việc nó ảnh hưởng đến tiền bạc, rủi ro hay con người ra sao — vì chính công việc “phiên dịch” đó là toàn bộ vai trò của bạn.

GIẢI ĐOẠN 0 — TẦNG TRỆT (hãy bắt đầu từ đây nếu công nghệ còn mới với bạn)

Hai module này dành cho người đọc chưa từng làm việc trong ngành công nghệ. Chúng giống hệt Module 1 và 2 của khóa học tương tác tại b4lcilc.org. Nếu bạn đã biết web server và cơ sở dữ liệu là gì, hãy lướt nhanh qua; còn nếu chưa, đây là những trang quan trọng nhất của cuốn sách này — mọi thứ phía sau đều đứng trên nền của chúng.

Module A: Web vận hành như thế nào — điều gì xảy ra khi bạn gõ một địa chỉ

Ý tưởng lớn: Một trang web không phải là một bức ảnh được phát thẳng đến điện thoại của bạn. Nó là một tập hợp các **chỉ dẫn viết sẵn** được gửi từ một máy tính ở xa, và chính trình duyệt của bạn làm theo các chỉ dẫn đó để dựng nên trang web trên màn hình. Học thuộc sáu bước từ lúc gõ một địa chỉ đến lúc thấy trang hiện ra, và mọi cuộc họp công nghệ trong suốt phần còn lại của sự nghiệp sẽ trở nên dễ theo dõi hơn.

Hành trình, gói trong sáu bước mà bạn nên nói được thành tiếng:

1. Bạn gõ một cái tên — `b4lcilc.org` — vào một **trình duyệt** (Chrome, Safari, Edge, Firefox).
2. **DNS**, cuốn danh bạ điện thoại của internet, biến cái tên đó thành một **địa chỉ IP** — địa chỉ dạng số của một máy tính, chẳng hạn `142.250.66.14`.
3. Trình duyệt của bạn — **client**, bên đặt câu hỏi — gửi một **HTTP request** (yêu cầu HTTP) đến máy tính ở địa chỉ đó: “GET — hãy đưa tôi trang này.”
4. Yêu cầu ấy di chuyển dưới dạng các **packet** (gói tin) nhỏ theo luật chơi của **TCP/IP**, hệ thống ông nước của internet, vốn đánh số từng gói tin, xác nhận đã giao đến nơi, và gửi lại bất cứ gói nào bị thất lạc.
5. Máy tính ở xa — **server** (máy chủ), bên trả lời — gửi về một **response** (phản hồi): các chỉ dẫn của trang, hoặc một lời cáo lỗi được đánh số như **404** (“không có trang này”).
6. Trình duyệt của bạn làm theo các chỉ dẫn đó và dựng nên trang web. Các chỉ dẫn được viết bằng ba ngôn ngữ: **HTML** (cấu trúc — tiêu đề, đoạn văn, nút bấm), **CSS** (hình thức — màu sắc, kích thước, bố cục), và **JavaScript** (hành vi — điều gì xảy ra khi bạn nhấn chuột).

Client-side và server-side — khung tư duy sắp xếp mọi thứ: công việc được làm trên thiết bị của bạn, trong trình duyệt, là *client-side* (**front end** — phía giao diện); công việc được làm trên máy tính ở xa là *server-side* (**back end** — phía sau). Quy tắc nằm lòng đáng ghi nhớ: **bất cứ điều gì liên quan đến lòng tin hay trí nhớ đều thuộc về server; bất cứ điều gì liên quan đến cảm giác nhanh nhạy đều thuộc về client**. Kiểm tra mật khẩu, lưu một bản ghi, trừ tiền thẻ: server. Vẽ trang, phản hồi tức thì khi bạn gõ phím: client. Đứng giữa hai bên thường có một **proxy** — một server trung gian đảm nhiệm việc lọc, che chắn, hoặc phân tán lưu lượng; các “load balancer” và “CDN” ở những module sau chính là các proxy có chuyên môn riêng.

Vì sao DNS quan trọng với một người lãnh đạo: DNS hỏng theo một kiểu rất đặc biệt — khi nó gãy, mọi trang web “biến mất” cùng một lúc dù mọi server vẫn đang chạy bình thường. Cuốn danh bạ bị sập, chứ không phải các máy điện thoại. Khi một kỹ sư nói “có khi là do DNS,” họ muốn nói đúng điều đó — và một số sự cố nổi tiếng nhất của internet chính xác là chuyện này.

Và chiếc ổ khóa: HTTPS là HTTP cộng thêm mã hóa — toàn bộ cuộc trò chuyện được xáo trộn để chống lại bất kỳ ai đứng giữa, và đó chính là điều mà chiếc ổ khóa trên thanh địa chỉ chứng nhận. Quy tắc thời hiện đại: mọi trang web, mọi lúc.

Từ vựng:

Thuật ngữ	Định nghĩa
Web browser (trình duyệt)	Chương trình đi lấy các trang web về và vẽ chúng lên màn hình: Chrome, Safari, Edge, Firefox.
URL	Địa chỉ đầy đủ mà bạn gõ: giao thức + tên máy tính + tài liệu bạn muốn lấy.
Địa chỉ IP	Địa chỉ dạng số của một máy tính trên mạng, chẳng hạn 142.250.66.14.
DNS	Cuốn danh bạ điện thoại của internet: biến tên (google.com) thành địa chỉ IP.
HTTP / HTTPS	Bộ luật hỏi-và-đáp mà trình duyệt và server dùng để nói chuyện; chữ S thêm mã hóa (chiếc ổ khóa).
TCP/IP	Hệ thống ống nước của internet: chuyển các gói tin được đánh số một cách đáng tin cậy giữa các địa chỉ.
Client / server	Bên hỏi và bên trả lời. Đây là một mối quan hệ, không phải một loại máy.
Client-side (front end)	Công việc làm trong trình duyệt của người dùng: vẽ giao diện, phản hồi tức thì.
Server-side (back end)	Công việc làm trên chiếc máy ở xa: lòng tin, luật lệ, hồ sơ lưu vĩnh viễn.
Proxy	Một server trung gian xử lý lưu lượng thay mặt cho một bên.
HTML / CSS / JavaScript	Ba ngôn ngữ của trang web: cấu trúc / hình thức / hành vi.

Fluency drill: nói thành tiếng hành trình sáu bước, một mình, cho đến khi bạn nói xong trong vòng dưới ba mươi giây và không bao giờ phải ngập ngừng tìm từ. Sau đó nói lại cho một người khác nghe.

Milestone: bạn có thể trả lời, ngay trên bàn ăn tối, không cần chuẩn bị: “Thực sự thì điều gì xảy ra khi tôi gõ tên một trang web vào điện thoại?” — như kể một câu chuyện, chứ không phải liệt kê một tràng thuật ngữ.

Module B: Website được xây và lưu trữ như thế nào — từ một chiếc máy tính đến cloud

Ý tưởng lớn: Mọi website thực thụ đều được xây theo **ba tầng** — một front end nơi con người chạm vào, một tầng giữa nơi các luật lệ của tổ chức vận hành, và một tầng dữ liệu không bao giờ quên — và suốt hai mươi lăm năm qua, gần như tất cả đều được lắp ráp từ một trong hai gia đình công cụ nổi tiếng. Học thuộc ba tầng và hai gia đình ấy, và các cuộc họp với nhà cung cấp sẽ chuyển từ một thứ tiếng nước ngoài thành một tâm thực đơn.

Nhà hàng — phép so sánh bạn sẽ dùng suốt phần còn lại của khóa học: front end (UI — giao diện người dùng) là phòng ăn — mọi thứ khách hàng nhìn thấy, được xây bằng HTML, CSS và JavaScript. **Tầng giữa (middle tier — business logic, logic nghiệp vụ)** là căn bếp — code chạy trên server áp dụng các luật của nhà: “một học sinh không được ghi danh hai lần,” “đơn hàng vượt hạn mức cần được phê duyệt.” **Tầng dữ liệu (data layer)** là kho thực phẩm và cuốn sổ công thức — **cơ sở dữ liệu (database)**, nơi cất giữ mọi thứ không được phép quên. Giới trong ngành gọi hình dạng này là **kiến trúc N-tier** (nhiều tầng). Vì sao phải chia tầng? Để mỗi tầng có thể được tu sửa mà không phải xây lại các tầng khác — và khi một kỹ sư nói “điểm nghẽn nằm ở tầng dữ liệu,” họ đang trao cho bạn đúng tâm bản đồ này với một vòng tròn khoanh sẵn.

Chiếc máy và chương trình: một **server** (máy chủ) là một chiếc máy tính có thật — một hộp kim loại không màn hình, xếp giá cùng hàng trăm chiếc khác trong một **data center** (trung tâm dữ liệu), được chế tạo để chạy từng giây trong suốt cả năm. Trên đó chạy một *chương trình* cũng được gọi là web server, đảm nhận cuộc trò chuyện HTTP. Hãy biết mặt gọi tên hai cái nổi tiếng nhất: **IIS** (của Microsoft, chạy trên Windows) và **Apache** (miễn phí, chạy chủ yếu trên Linux). Giữ cho các từ khỏi lẫn lộn: *trình duyệt* hỏi; *web server* trả lời. Apache là một web **server** — bên trả lời — chứ không phải một trình duyệt. **Web hosting** là thuê chỗ trên server của người khác để chiếc máy của họ trả lời dưới tên của bạn.

Cơ sở dữ liệu và SQL, một cách nhẹ nhàng: một **cơ sở dữ liệu (database)** là một chương trình lưu trữ các bản ghi và trả lời các câu hỏi về chúng — một nhân viên lưu trữ ngăn nắp đến mức cuống tin và không bao giờ ngủ. **SQL** (đọc là “sequel”) đơn giản là **ngôn ngữ để đặt câu hỏi cho cơ sở dữ liệu**. Một *query* (truy vấn) là một câu hỏi. Hãy đọc dòng này — bạn thực ra đã đọc được rồi: `SELECT name FROM students WHERE village = 'Mae La'` — *từ bảng students, cho tôi tên của tất cả những ai đến từ Mae La*. Hai thương hiệu thông trị: **SQL Server** (của Microsoft, phải mua giấy phép) và **MySQL** (miễn phí, mã nguồn mở). Cả hai đều nói SQL. Điểm dành cho người lãnh đạo: **cơ sở dữ liệu là nơi tổ chức thực sự sống** — các trang web có thể dừng lại được, nhưng mật hồ sơ ghi danh thì không còn trường học nữa. Đó là lý do cơ sở dữ liệu nhận được sự bảo vệ kỹ thuật kỹ lưỡng nhất trong cả tòa nhà.

Hai gia đình:

Vai trò trong cỗ máy	Gia đình Microsoft	Gia đình mã nguồn mở LAMP
Hệ điều hành	Windows Server	Linux
Web server (trả lời HTTP)	IIS	Apache
Ngôn ngữ logic nghiệp vụ	C# trên .NET	PHP
Cơ sở dữ liệu	SQL Server	MySQL
Mô hình chi phí	Phí giấy phép, một nhà cung cấp	Miễn phí và mã nguồn mở; bạn trả tiền cho con người
Lãnh địa quen thuộc	Ngân hàng, chính phủ, doanh nghiệp lớn	Web đại chúng — riêng WordPress đã chạy một phần ba tổng số website

Mã nguồn mở (open source) nghĩa là công thức của phần mềm được công khai — bất kỳ ai cũng có thể xem xét, cải tiến và sử dụng mà không phải trả phí giấy phép. Đó là lý do một tổ chức eo hẹp vẫn luôn có thể xây một website đẳng cấp thế giới với chi phí chỉ bằng một chiếc máy đi thuê.

Con đường đến với cloud — câu văn mà cả cuốn sách này đứng trên đó: trong nhiều thập kỷ, tất cả những thứ trên chạy **on-premises** (tại chỗ): trong một phòng server do tổ chức sở hữu, được mua sắm cho cái ngày bận rộn nhất có thể tưởng tượng ra, năm không mọi ngày còn lại, và được giữ mát bằng tiền của chính tổ chức. Rồi Amazon và Microsoft đề nghị cho thuê năng lực từ các nhà kho của chính họ, tính theo giờ. Các bộ công cụ không hề thay đổi — vẫn chính Linux, Apache, IIS, SQL Server ấy đang chạy trên cloud ngày nay. **Điều thay đổi là các chiếc máy nằm ở đâu và ai sở hữu chúng.** Hãy giữ chặt câu đó: Module 1 của Giai đoạn 1 mở ra đúng từ chỗ ấy.

Từ vựng:

Thuật ngữ	Định nghĩa
Server / data center	Chiếc máy chạy liên tục không nghỉ, và tòa nhà chứa đầy các giá xếp những chiếc máy như thế.
Web server (chương trình)	Phần mềm trả lời các HTTP request: IIS (Microsoft), Apache (mã nguồn mở).
Web hosting	Thuê chỗ trên server của người khác cho website của bạn.
Kiến trúc N-tier	Xây theo tầng: front end (UI), tầng giữa (logic), tầng dữ liệu.
Database (cơ sở dữ liệu)	Chương trình lưu trữ các bản ghi và trả lời các câu hỏi — nơi tổ chức thực sự sống.
SQL / query	Ngôn ngữ gần như tiếng Anh để đặt câu hỏi cho cơ sở dữ liệu; một query là một câu hỏi.
SQL Server / MySQL	Cơ sở dữ liệu có giấy phép của Microsoft / gười mở mã nguồn mở được ưa chuộng. Cả hai đều nói SQL.
Microsoft stack	Windows Server + IIS + C#/.NET + SQL Server.
LAMP stack	Linux + Apache + MySQL + PHP — tất cả đều mã nguồn mở.
Open source (mã nguồn mở)	Công thức công khai: tự do xem xét, cải tiến và sử dụng.
On-premises	Máy do bạn sở hữu, trong căn phòng do bạn sở hữu — thế giới trước khi có cloud.

Fluency drill: đếm trên bốn ngón tay để giải nghĩa LAMP, nói thành tiếng, ba ngày liên tiếp. Sau đó giải thích phép so sánh nhà hàng — phòng ăn, căn bếp, kho thực phẩm — cho một người chưa từng nghe đến server.

Milestone: một nhà thầu nói: “Bên anh đang chạy một hệ thống Microsoft on-prem cũ — IIS và SQL Server. Bọn tôi là một LAMP shop, và bọn tôi sẽ chuyển anh lên cloud.” Bạn có thể dịch từng từ một cho ban lãnh đạo của mình — và bạn biết phần rủi ro cần xoay vào chính là việc di chuyển cơ sở dữ liệu.

GIAI ĐOẠN 1 — NÓI ĐƯỢC NGÔN NGỮ (Tuần 1-8)

Module 1 (Tuần 1-2): Cloud thực chất là gì

Ý tưởng lớn: Cloud là *máy tính của người khác, thuê theo giờ, được quản lý bằng phần mềm*. Trước khi có cloud, một công ty phải mua server (máy chủ) vật lý, đặt chúng trong một căn phòng, và trả tiền cho người bảo trì — chậm chạp, tốn kém, thiếu linh hoạt. AWS, Microsoft Azure và Google Cloud đã xây những nhà kho chứa hàng triệu máy tính (data center — trung tâm dữ liệu) và cho phép bất kỳ ai thuê từng “lát” của chúng theo từng giây, từ bất cứ đâu, thông qua một trang web hoặc một dòng lệnh. Chỉ vậy thôi. Mọi thứ còn lại trong khóa học này đều là chi tiết xây trên một ý tưởng duy nhất đó.

Ba nhà cung cấp lớn nhất, định nghĩa rõ ràng:

Nhà cung cấp	Đó là gì
AWS (Amazon Web Services)	Bộ phận cloud của Amazon và là nhà cung cấp cloud lớn nhất thế giới (~30% thị trường toàn cầu). Khởi đầu năm 2006 khi Amazon bắt đầu cho thuê các hệ thống máy tính vốn được xây cho cửa hàng của chính mình. Ngày nay AWS cung cấp hơn 200 dịch vụ — server, lưu trữ, cơ sở dữ liệu, AI và nhiều hơn nữa — cho thuê theo từng giây. Khi khóa học này nói “cloud”, AWS là ví dụ mặc định, và AWS đã khai trương Region (vùng) Thái Lan của riêng mình tại Bangkok vào tháng 1 năm 2025.
Microsoft Azure	Cloud của Microsoft, đứng thứ 2 toàn cầu và mạnh nhất bên trong các doanh nghiệp lớn vì nó kết nối tự nhiên với các công cụ Microsoft mà doanh nghiệp đã dùng sẵn (Windows, Office, Active Directory). Hiện đang xây dựng datacenter region đầu tiên tại Thái Lan.
Google Cloud (GCP)	Cloud của Google, đứng thứ 3 — mạnh nhất về phân tích dữ liệu và công cụ AI. Đã cam kết 1 tỷ USD cho một trung tâm dữ liệu tại Thái Lan và một cloud region ở Bangkok.

Tài khoản AWS miễn phí của bạn — hãy thiết lập ngay trong Tuần 1. Truy cập <https://aws.amazon.com/free> và nhấn “Create a Free Account” (trang đăng ký trực tiếp là https://signin.aws.amazon.com/signup?request_type=register). Bạn cần một địa chỉ email, một số điện thoại và một thẻ tín dụng/ghi nợ để xác minh danh tính — **Free Tier** (gói miễn phí) cho bạn hạn mức hàng tháng miễn phí với các dịch vụ cơ bản (bao gồm 750 giờ/tháng cho một server EC2 nhỏ trong năm đầu tiên), và toàn bộ bài tập của khóa học này nằm gọn trong hạn mức đó. Hai thói quen an toàn ngay từ ngày đầu: bật MFA (Module 6 sẽ giải thích) và đặt **billing alert** (cảnh báo chi phí) ở mức 5 USD (Module 7 hướng dẫn cách làm) để bạn không bao giờ bị bất ngờ. Sau khi đăng ký, bạn đăng nhập tại <https://console.aws.amazon.com> — “console” đơn giản là trang web điều khiển của AWS.

Vì sao các công ty dùng cloud: tốc độ (một server mới trong 60 giây thay vì 6 tuần), tính co giãn — elasticity (thuê 100 server cho ngày sale lớn, trả lại 90 chiếc vào hôm sau), không cần chi phí đầu tư ban đầu (chi phí vận hành thay vì chi phí vốn), và tầm với toàn cầu (đặt ứng dụng gần khách hàng của bạn ở Bangkok, Tokyo và Frankfurt mà không phải xây gì cả).

Ba tầng dịch vụ — mô hình tư duy hữu ích nhất trong cloud:

Tầng	Bạn thuê gì	Ví dụ về nhà bếp	Ví dụ
IaaS (Infrastructure as a Service — hạ tầng như một dịch vụ)	Máy tính “thô”, lưu trữ, mạng — bạn tự quản lý mọi thứ trên đó	Thuê một căn bếp trống: bạn mang theo đầu bếp, công thức, nguyên liệu	Amazon EC2, Azure Virtual Machines
PaaS (Platform as a Service — nền tảng như một dịch vụ)	Một nền tảng được quản lý sẵn — bạn chỉ mang ứng dụng của mình lên	Thuê một căn bếp có sẵn nhân viên: bạn chỉ mang công thức	AWS Elastic Beanstalk, Azure App Service
SaaS (Software as a Service — phần mềm như một dịch vụ)	Phần mềm hoàn chỉnh	Gọi món từ nhà hàng	Gmail, Salesforce, Canva

Region và Availability Zone: một **Region** (vùng) là một cụm trung tâm dữ liệu theo địa lý (Thái Lan có AWS Region riêng từ tháng 1 năm 2025, ở trong và quanh Bangkok). Một **Availability Zone (AZ — vùng khả dụng)** là một hoặc nhiều trung tâm dữ liệu tách biệt bên trong một region. Những ứng dụng quan trọng chạy trên ít nhất hai AZ, để sự cố của một tòa nhà không kéo sập chúng. Khi các kỹ sư nói “chúng ta chạy multi-AZ”, ý họ là “một trung tâm dữ liệu có thể cháy rụi mà chúng ta vẫn hoạt động.”

Từ vựng:

Thuật ngữ	Định nghĩa
Cloud provider / hyperscaler	Công ty sở hữu các trung tâm dữ liệu khổng lồ và cho thuê năng lực tính toán qua internet (AWS, Azure, Google Cloud). “Hyperscaler” = số ít công ty lớn nhất, được xây để mở rộng gần như vô hạn.
Data center	Một nhà kho được bảo vệ nghiêm ngặt

Thuật ngữ	Định nghĩa
	chứa hàng nghìn server với hệ thống điện và làm mát công nghiệp — nơi vật lý mà “cloud” thực sự tồn tại.
Region	Một cụm trung tâm dữ liệu theo địa lý được cung cấp như một lựa chọn vị trí, ví dụ “Asia Pacific (Bangkok)”. Bạn chọn region nơi hệ thống của mình chạy.
Availability Zone (AZ)	Một hoặc nhiều trung tâm dữ liệu tách biệt bên trong một region, có nguồn điện và mạng độc lập. Chạy trên hai AZ nghĩa là một tòa nhà có thể gặp sự cố mà bạn vẫn trực tuyến.
On-premises (“on-prem”)	Cách làm cũ: server do công ty bạn sở hữu, đặt trong tòa nhà của chính bạn. Đối lập với cloud.
Migration	Dự án di chuyển hệ thống từ on-prem lên cloud.
Workload	Bất kỳ ứng dụng hay hệ thống nào đang chạy — “workload tính lương”, “workload website”. Từ rất tiện: nó bao trùm mọi thứ.
Provision	Tạo/thiết lập một tài nguyên cloud (một server, một cơ sở dữ liệu). “Provision một server” = đưa nó vào tồn tại.
Scale up / scale out	Đáp ứng nhu cầu cao hơn bằng cách làm một máy lớn hơn (up) hoặc thêm nhiều máy hơn (out). Cloud ưu ái <i>out</i> .
Latency	Độ trễ trước khi dữ liệu đến nơi, đo bằng mili giây. Khoảng cách tạo ra latency — lý do một region ở Bangkok quan trọng với người dùng Thái.
Console	Trang điều khiển web của nhà cung cấp, nơi bạn xem và quản lý mọi thứ mình đang thuê.

Video cho module này (tất cả miễn phí, liên kết đã kiểm chứng):

Video	Kênh	Thời lượng	Liên kết
What is AWS?	Amazon Web Services (official)	~2 phút	https://www.youtube.com/watch?v=a9_D53WsUs
Top 50+ AWS Services Explained in	Fireship	~10 phút	https://www.youtube.com/

Video	Kênh	Thời lượng	Liên kết
10 Minutes			watch?v=JlbiYCM48to
What is Microsoft Azure? An Introduction	Eye on Tech	~3 phút	https://www.youtube.com/watch?v=l9JkLhvaKA8
รู้จัก AWS Cloud คืออะไร (Thai-language intro)	Aware Corporation	ngắn	https://www.youtube.com/watch?v=nrSpZKGxXd0

Fluency drill — nói cho đến khi tự nhiên: “Is that workload on-prem or in the cloud?” (Workload đó đang chạy on-prem hay trên cloud?) · “Which region are we in — and are we multi-AZ?” (Chúng ta đang ở region nào — và có chạy multi-AZ không?) · “Is this an IaaS approach or is there a managed service that removes the maintenance?” (Đây là cách tiếp cận IaaS, hay có một managed service nào giúp loại bỏ việc bảo trì?)

Bài tập: (1) Tạo tài khoản AWS miễn phí tại <https://aws.amazon.com/free> — bản thân quy trình đăng ký dạy bạn nhiều từ vựng hơn cả một chương sách. (2) Xem bốn video trong bảng trên (tổng cộng dưới 20 phút). (3) Trong nhật ký của bạn: viết bài “thuyết trình thang máy” giải thích cloud mà bạn sẽ nói với một hiệu trưởng trường học Thái Lan trong một hơi.

Milestone: bạn có thể giải thích IaaS vs PaaS vs SaaS bằng phép so sánh của riêng mình, và giải thích vì sao việc AWS mở region Bangkok lại quan trọng với các ngân hàng Thái (đáp án: latency + data residency — xem Module 6).

Module 2 (Tuần 3-4): Các khối xây dựng — compute, lưu trữ và cơ sở dữ liệu

Compute (năng lực tính toán) — những cỗ máy chạy code. Một **virtual machine (VM/instance — máy ảo)** là một “lát” của server vật lý hoạt động như một máy tính hoàn chỉnh; bạn chọn kích cỡ (CPU/RAM) và trả tiền theo từng giây nó chạy. Một **container** là cách đóng gói một ứng dụng nhẹ hơn, nhanh hơn để nó chạy giống hệt nhau ở mọi nơi; **Docker** đóng gói chúng và **Kubernetes (K8s)** điều phối cả hạm đội container — khi nghe “K8s”, hãy nghĩ “hệ thống tự động chạy và tự chữa lành hàng trăm container của chúng ta”. **Serverless** (không cần quản lý server — AWS Lambda) nghĩa là bạn chỉ tải lên một hàm code; cloud chạy nó khi được kích hoạt và bạn trả tiền theo mỗi lần gọi — hoàn toàn không có server nào phải quản lý. Quy luật cần chú ý: VM — container — serverless là một thanh trượt từ “nhiều quyền kiểm soát hơn, nhiều việc bảo trì hơn” đến “ít quyền kiểm soát hơn, gần như không phải bảo trì”. Đội giới chọn theo từng workload, không chọn theo mô-tơ.

Lưu trữ — ba hình dạng. **Object storage** (lưu trữ đối tượng — Amazon S3): một chiếc “xô” không đáy cho file — hình ảnh, video, bản sao lưu, bộ dữ liệu; rẻ, độ bền “mười một số chín”, câu trả lời mặc định cho “chúng ta để file ở đâu?” **Block storage** (lưu trữ khối — EBS): ổ cứng ảo gắn vào một VM. **File storage** (lưu trữ tệp — EFS): một ổ đĩa dùng chung mà nhiều máy cùng gắn kết một lúc. Rồi đến **tier** (bậc lưu trữ): nóng (truy cập thường xuyên, đắt) và

lạnh/lưu trữ dài hạn (Glacier — rẻ, chậm) — chuyển dữ liệu cũ xuống bậc lạnh là một trong những chiến thắng chi phí dễ nhất mà bất kỳ đội nào cũng có thể thực hiện.

Cơ sở dữ liệu — hai họ. Relational/SQL (quan hệ — MySQL, PostgreSQL; được quản lý dưới dạng Amazon RDS/Aurora): dữ liệu trong các bảng có cấu trúc chặt chẽ; lựa chọn mặc định cho tiền bạc, đơn hàng, người dùng — bất cứ thứ gì mà tính chính xác là bất khả xâm phạm. **NoSQL** (DynamoDB, MongoDB): linh hoạt, mở rộng cực lớn; lựa chọn mặc định cho dữ liệu khổng lồ, nhanh, cấu trúc đơn giản hơn (session, danh mục, bảng tin). “Managed database” (cơ sở dữ liệu được quản lý) nghĩa là nhà cung cấp lo việc sao lưu, vá lỗi và failover — đội của bạn cần một lý do thật mạnh mới nên tự vận hành cơ sở dữ liệu riêng.

Từ vựng:

Thuật ngữ	Định nghĩa
Instance	Một máy ảo (VM) đang thuê. “Spin up một instance” = khởi động một server.
Instance type / size	Cấu hình bạn chọn cho nó — bao nhiêu CPU, bao nhiêu bộ nhớ. Loại lớn hơn = giá theo giờ cao hơn.
Container	Một gói nhẹ chứa một ứng dụng cùng mọi thứ nó cần, để nó chạy giống hệt nhau trên bất kỳ máy nào. Nhanh hơn và rẻ hơn một VM đầy đủ.
Docker	Công cụ tiêu chuẩn để xây và chạy container.
Kubernetes (K8s)	Bộ điều phối tự động chạy và “chữa lành” các hạm đội container — khởi động lại cái bị lỗi, thêm cái mới khi tải tăng. “K8s” là biệt danh trong ngành.
Cluster / node	Cluster là một nhóm máy hoạt động như một hệ thống; mỗi máy trong đó là một node.
Serverless	Chạy code mà không quản lý bất kỳ server nào — cloud chạy hàm của bạn khi được kích hoạt và tính tiền theo mỗi lần chạy.
Lambda / function	Sản phẩm serverless của AWS; một “function” là mẫu code nhỏ mà nó chạy.
S3 / bucket	Dịch vụ object storage của AWS cho file; một bucket là một “thùng chứa” file có tên. Câu trả lời mặc định cho “chúng ta để file ở đâu?”
EBS volume	Ổ cứng ảo gắn vào một instance.
Durability	Xác suất dữ liệu đã lưu còn nguyên vẹn. “Mười một số chín” của S3 (99,999999999%)

Thuật ngữ	Định nghĩa
Storage tier	nghĩa là gần như không bao giờ mất dữ liệu. Bậc giá/tốc độ cho dữ liệu: nóng (tức thời, đắt) – lạnh/lưu trữ dài hạn (Glacier — rẻ, mất vài phút đến vài giờ để lấy lại).
RDS	Dịch vụ cơ sở dữ liệu quan hệ được quản lý của AWS — AWS lo sao lưu, vá lỗi và failover cho bạn.
SQL vs NoSQL	SQL: băng chặt chẽ, hoàn hảo cho tiền bạc và đơn hàng. NoSQL: linh hoạt và mở rộng cực lớn, cho session, danh mục, bảng tin.
Backup / snapshot	Một bản sao dữ liệu đã lưu (snapshot là bản sao tại-một-thời-điểm của ổ đĩa hoặc cơ sở dữ liệu) mà bạn có thể khôi phục từ đó.
Failover	Việc tự động chuyển sang bản sao dự phòng khi bản chính gặp sự cố — lý do managed database sống sót qua những đêm tồi tệ.

Video cho module này:

Video	Kênh	Thời lượng	Liên kết
Getting Started with EC2	AWS Developers (official)	27 phút	https://www.youtube.com/watch?v=nJ-djerESW0
Introduction to Amazon S3	Amazon Web Services (official)	~5 phút	https://www.youtube.com/watch?v=ecv-19sYL3w
Docker in 100 Seconds	Fireship	2 phút	https://www.youtube.com/watch?v=Gjnup-PuquQ
Kubernetes explained in 15 mins	TechWorld with Nana	~16 phút	https://www.youtube.com/watch?v=VnvRFRk_51k
Serverless Computing in 100 Seconds	Fireship	~2 phút	https://www.youtube.com/watch?v=W_VV2Fx32_Y

Fluency drill: “Should this run on VMs, containers, or serverless — and what’s the operational cost of each choice?” (Cái này nên chạy trên VM, container hay serverless — và chi phí vận

hành của mỗi lựa chọn là gì?) · “Is that data hot or can it go to a cheaper tier?” (Dữ liệu đó có “nóng” không, hay có thể chuyển xuống bậc rẻ hơn?) · “Why are we self-managing that database instead of using RDS?” (Tại sao chúng ta tự quản lý cơ sở dữ liệu đó thay vì dùng RDS?)

Bài tập: (1) Trong tài khoản miễn phí của bạn: khởi chạy instance EC2 nhỏ nhất, rồi terminate (chấm dứt) nó. Tải một file lên S3. Bạn vừa đích thân sử dụng IaaS. (2) Nhờ một trợ lý AI kiểm tra bạn: “Cho tôi 10 tình huống; tôi sẽ trả lời VM, container hay serverless, và bạn chấm điểm.”

Milestone: với bất kỳ ứng dụng đơn giản nào được mô tả trong một câu (“một website nơi học sinh đặt cơm trưa”), bạn có thể nêu tên các thành phần compute, lưu trữ và cơ sở dữ liệu của nó thành tiếng trong 60 giây.

Module 3 (Tuần 5–6): Mạng, và cách các mảnh ghép nói chuyện với nhau

VPC — khu phố riêng của bạn. Một **Virtual Private Cloud** (đám mây riêng ảo) là khu vực được rào chắn riêng của bạn bên trong mạng của nhà cung cấp. Bên trong nó là các **subnet** (mạng con) — loại public (truy cập được từ internet, ví dụ web server) và loại private (không truy cập được, ví dụ cơ sở dữ liệu). Câu nói về bảo mật phổ biến nhất mà bạn sẽ nghe: “*cơ sở dữ liệu nằm trong private subnet.*” Nếu bạn hiểu vì sao — kẻ tấn công không thể chạm vào thứ không có đường dẫn nào từ internet — bạn đã hiểu một nửa về bảo mật mạng.

Tầng lưu lượng: một **load balancer** (bộ cân bằng tải) phân phối lưu lượng truy cập đến trên nhiều server (và lặn lẽ loại bỏ những chiếc “ôm yếu”); **DNS** (Route 53) dịch những cái tên như yourcompany.com thành địa chỉ; một **CDN** (CloudFront) lưu bản sao nội dung của bạn tại hàng trăm thành phố để nó tải nhanh ở mọi nơi; một **API** là cánh cửa mà một phần mềm mở ra cho phần mềm khác — khi kỹ sư nói “we’ll expose an API”, ý họ là “chúng ta sẽ cho phần mềm khác một cách có kiểm soát để dùng phần mềm của mình”; một **API gateway** là quầy lễ tân quản lý những cánh cửa đó.

Kết nối các thế giới: **VPN** (đường hầm mã hóa qua internet) hoặc **Direct Connect** (đường truyền vật lý riêng) nối văn phòng/hệ thống on-prem của công ty với cloud của họ. **Hybrid cloud** (đám mây lai) = chạy cả on-prem lẫn cloud (đa số doanh nghiệp Thái); **multi-cloud** = dùng nhiều hơn một nhà cung cấp.

Từ vựng:

Thuật ngữ	Định nghĩa
VPC (Virtual Private Cloud)	Khu vực riêng, được rào chắn của bạn bên trong mạng của nhà cung cấp, nơi hệ thống của bạn “sinh sống”.
Subnet (public / private)	Một phân khu của VPC. Public subnet truy cập được từ internet (web server); private subnet thì không (cơ sở dữ liệu).
IP address	Địa chỉ dạng số của một máy trên mạng — cách các máy tính tìm thấy nhau.
Firewall / security group	Danh sách quy tắc quy định lưu lượng nào

Thuật ngữ	Định nghĩa
	được phép đến một máy (“cho web traffic vào, chặn mọi thứ khác”). Security group là tường lửa theo-từng-server của AWS.
Load balancer	“Người điều phối giao thông” phân phối các request đến trên nhiều server và ngừng gửi đến những server không khỏe.
DNS	Danh bạ điện thoại của internet — dịch yourcompany.com thành địa chỉ IP. Dịch vụ DNS của AWS là Route 53.
CDN (Content Delivery Network)	Các bản sao nội dung của bạn được lưu đệm tại hàng trăm thành phố để tải nhanh ở mọi nơi. CDN của AWS là CloudFront.
Edge location	Một trong những điểm CDN cấp thành phố đó — “the edge” (biên) nghĩa là gần người dùng.
API	Cánh cửa mà một phần mềm mở ra cho phần mềm khác. “We’ll expose an API” = chúng ta sẽ cho phần mềm khác một cách có kiểm soát để dùng phần mềm của mình.
API gateway	Quầy lễ tân quản lý những cánh cửa đó — xác thực, giới hạn tần suất, ghi log.
VPN	Đường hầm mã hóa qua internet công cộng nối hai mạng với nhau (ví dụ văn phòng của bạn với VPC của bạn).
Direct Connect	Đường truyền vật lý riêng đến cloud — nhanh hơn và ổn định hơn VPN, với một mức giá tương xứng.
Hybrid / multi-cloud	Hybrid: chạy on-prem và cloud cùng lúc (đa số doanh nghiệp Thái). Multi-cloud: dùng nhiều hơn một nhà cung cấp.
Ingress / egress	Lưu lượng đi vào / đi ra khỏi cloud. Egress tốn tiền — hãy nhớ từ này cho Module 7.

Video cho module này: AWS Networking Basics — VPC & Subnets for Beginners — KodeKloud — https://www.youtube.com/watch?v=QM63dyA_4Pc (xem nửa đầu ngay bây giờ; quay lại xem phần còn lại sau Module 5).

Fluency drill: “Is the database in a private subnet?” (Cơ sở dữ liệu có nằm trong private subnet không?) · “What happens when one web server dies — is the load balancer health-checking?” (Điều gì xảy ra khi một web server chết — load balancer có đang kiểm tra sức khỏe không?) · “Are we exposing that as an API or is it internal only?” (Chúng ta mở cái đó ra thành API hay chỉ dùng nội bộ?)

Bài tập: (1) Nhờ một AI hướng dẫn bạn vẽ, trên giấy, mạng của một ứng dụng giao đồ ăn: người dùng – CDN – load balancer – web server (public subnet) – cơ sở dữ liệu (private subnet). Vẽ ba lần cho đến khi bạn vẽ được từ trí nhớ. (2) Tìm sơ đồ kiến trúc của công ty bạn hoặc bất kỳ sơ đồ mẫu nào và khoanh tròn mọi ô mà giờ đây bạn gọi được tên.

Milestone: bạn có thể phác thảo sơ đồ ba tầng tiêu chuẩn đó lên bảng trắng và thuật lại hành trình một cú nhấp chuột của khách hàng đi qua nó.

Module 4 (Tuần 7–8): Cách các đội cloud làm việc — DevOps, IaC, và đọc vị phòng họp

DevOps là văn hóa hợp nhất “những người viết phần mềm” (Dev) và “những người vận hành nó” (Ops) thành một đội duy nhất, phát hành những thay đổi nhỏ một cách thường xuyên và an toàn, với tự động hóa gánh phần việc nặng. Nhịp tim của nó là **CI/CD pipeline** (đường ống tích hợp/triển khai liên tục): mỗi thay đổi code được tự động build, kiểm thử và triển khai (Continuous Integration / Continuous Delivery). Khi một đội nói “it’s in the pipeline”, ý họ là con robot đang kiểm thử và phát hành nó.

Infrastructure as Code (IaC — hạ tầng dưới dạng code) — ý tưởng đã thay đổi tất cả: thay vì nhấp chuột loay hoay trong console để tạo server, kỹ sư viết các *file văn bản* khai báo hạ tầng (“hai server, một load balancer, một cơ sở dữ liệu, những quy tắc tường lửa này”), và một công cụ (**Terraform**, CloudFormation) làm cho thực tế khớp với file. Vì sao lãnh đạo quan tâm: các file đó sống trong **Git** (hệ thống quản lý phiên bản), nên mọi thay đổi hạ tầng đều được review, có thể đảo ngược và có thể kiểm toán — sự khác biệt giữa một xưởng thủ công và một nhà máy.

Những nghi thức bạn sẽ ngồi dự: stand-up (15 phút mỗi ngày: đã xong gì, tiếp theo là gì, đang vướng gì — hãy lắng nghe các blocker; gỡ chúng là việc của bạn), **sprint** (một đơn vị công việc có kế hoạch kéo dài 1–2 tuần), **retro** (cân nhắc cải thiện gì), **post-mortem/incident review** (sau một sự cố ngừng hoạt động: phân tích *không đổ lỗi* về cái gì đã hỏng và cái gì sẽ ngăn nó lặp lại — sức khỏe của một đội thể hiện ở chỗ những buổi này có trung thực hay không), **on-call** (vòng xoay trực — ai bị đánh thức lúc 3 giờ sáng; nếu on-call khổ sở, những người giỏi nhất của bạn sẽ ra đi — hãy hỏi về nó hằng tháng).

Các chỉ số sức khỏe quan trọng: uptime/availability (thời gian hoạt động/tính khả dụng — “ba số chín” = 99,9% ≈ 8,8 giờ ngừng hoạt động/năm; mỗi số chín thêm vào nhân chi phí lên nhiều lần), **SLA** (lời hứa với khách hàng, kèm điều khoản phạt), **SLO** (mục tiêu nội bộ), **MTTR** (bạn phục hồi nhanh đến đâu — các đội trưởng thành tối ưu khả năng phục hồi, chứ không theo đuổi ảo tưởng không-bao-giờ-hỏng), **deployment frequency** (tần suất triển khai — đội khỏe mạnh phát hành thay đổi nhỏ thường xuyên; nỗi sợ deploy là một dấu hiệu xấu).

Từ vựng:

Thuật ngữ	Định nghĩa
DevOps	Văn hóa một đội vừa xây vừa vận hành phần mềm của mình, phát hành thay đổi nhỏ thường xuyên với tự động hóa.
CI/CD pipeline	Băng chuyền tự động build, kiểm thử và

Thuật ngữ	Định nghĩa
	triển khai mọi thay đổi code (Continuous Integration / Continuous Delivery).
Deploy / rollback	Deploy: phát hành một thay đổi lên hệ thống thật. Rollback: hoàn tác nhanh khi nó gây sự cố.
Git / repository / pull request	Git: hệ thống quản lý phiên bản ghi lại mọi thay đổi. Repository (“repo”): ngôi nhà chứa code của một dự án. Pull request (PR): một thay đổi được đề xuất để kỹ sư khác review trước khi hợp nhất.
IaC (Infrastructure as Code)	Khai báo hạ tầng trong các file văn bản mà một công cụ biến thành hiện thực — được review, có thể đảo ngược, có thể kiểm toán.
Terraform	Công cụ IaC phổ biến nhất (chạy trên mọi cloud). Công cụ riêng của AWS là CloudFormation.
Staging vs production (“prod”)	Staging: bản “diễn tập” của hệ thống. Prod: bản thật mà khách hàng chạm vào. “Broke prod” (làm sập prod) = một ngày tồi tệ.
Stand-up	Buổi đồng bộ 15 phút mỗi ngày: đã xong / tiếp theo / đang vướng. Lắng nghe các blocker — gỡ chúng là việc của bạn.
Sprint / backlog	Sprint: một đơn vị công việc có kế hoạch 1–2 tuần. Backlog: danh sách việc-cần-làm đã sắp thứ tự để “nuôi” các sprint.
Retro	Cuộc họp cuối sprint về cách làm việc tốt hơn lần sau.
Post-mortem	Buổi rà soát không đổ lỗi sau sự cố: cái gì hỏng, vì sao, cái gì ngăn nó lặp lại.
On-call	Vòng xoay trực — ai trả lời báo động lúc 3 giờ sáng. Nếu on-call khổ sở, người giỏi nhất của bạn sẽ ra đi.
Incident / sev-1	Một gián đoạn ngoài kế hoạch; “sev-1” (mức nghiêm trọng một) = loại tồi tệ nhất, cả đội vào cuộc.
SLA / SLO	SLA: lời hứa với khách hàng kèm điều khoản phạt (ví dụ 99,9% uptime). SLO: mục tiêu nội bộ nghiêm ngặt hơn để bảo vệ SLA.
MTTR	Mean Time To Recovery — thời gian trung bình để hoạt động trở lại sau sự cố. Các đội trưởng thành tồi ưu chỉ số này, không theo đuổi ảo tưởng không bao giờ hỏng.

Thuật ngữ	Định nghĩa
Monitoring / observability	Theo dõi sức khỏe hệ thống trực tiếp qua log (bản ghi sự kiện), metric (con số theo thời gian) và alert (báo động tự động khi vượt ngưỡng).

Video cho module này:

Video	Kênh	Thời lượng	Liên kết
What is DevOps? REALLY understand it	TechWorld with Nana	~15 phút	https://www.youtube.com/watch?v=0yWAtQ6wYNM
DevOps CI/CD Explained in 100 Seconds	Fireship	2 phút	https://www.youtube.com/watch?v=scEDHsr3APg
Terraform explained in 15 mins	TechWorld with Nana	18 phút	https://www.youtube.com/watch?v=l5klai_GBDE

Fluency drill: “Is that change through the pipeline or was it manual?” (Thay đổi đó đi qua pipeline hay làm thủ công?) · “What did the post-mortem conclude, and what’s the prevention item?” (Post-mortem kết luận gì, và hạng mục phòng ngừa là gì?) · “What’s our MTTR trending like?” (MTTR của chúng ta đang có xu hướng thế nào?) · “Is this in Terraform, or did someone click it into existence?” (Cái này có trong Terraform không, hay ai đó đã nhấp chuột tạo ra nó?) (kiểu sau được gọi là “ClickOps”, nói kèm một cái cau mày).

Bài tập: (1) Xem một bản viết post-mortem sự cố thực tế được thảo luận (nhiều bản công khai — các báo cáo sự cố của Cloudflare và AWS rất nổi tiếng) và tóm tắt vào nhật ký trong năm câu. (2) Ngồi dự (hoặc xem bản ghi hình) một buổi stand-up bất kỳ và viết ra ba blocker bạn nghe được.

Milestone — kết thúc Giai đoạn 1: bạn có thể ngồi suốt một cuộc họp lập kế hoạch kỹ thuật 30 phút và theo kịp ≥80% nội dung, và nhật ký của bạn chứng minh điều đó: ghi chú từ một cuộc họp thật hoặc mô phỏng với mọi từ viết tắt được mở rộng chính xác. **Đây cũng là lúc bạn nên đặt lịch thi AWS Cloud Practitioner (CLF-C02)** — 2–4 tuần ôn tập tập trung trên nền các module này là chuẩn được công bố, và việc thi đầu là bằng chứng bên ngoài đầu tiên của bạn.

GIAI ĐOẠN 2 — TƯ DUY NHƯ CẢ PHÒNG HỌP (Tuần 9–16)

Module 5 (Tuần 9–10): Kiến trúc — đánh giá các bản thiết kế mà không cần tự vẽ chúng

Vai trò của bạn trong một buổi design review không phải là thiết kế — mà là chất vấn. Khung tham chiếu mà cả ngành sử dụng là **Well-Architected Framework** của AWS, sáu trụ cột mà mọi bản thiết kế phải trả lời được: vận hành xuất sắc, **bảo mật**, **độ tin cậy**, hiệu quả hiệu năng, **tối ưu chi phí**, tính bền vững. Hãy học sâu ba trụ cột in đậm; đó là nơi tiền bạc và rủi ro trú ngụ.

Độ tin cậy, nói bằng ngôn ngữ thường: mọi thứ rồi sẽ hỏng, nên hệ thống tốt mặc định điều đó. **Redundancy** (dự phòng — không có điểm hỏng đơn lẻ nào; mọi thứ quan trọng đều có hai bản), **multi-AZ** (sóng sót khi mất một trung tâm dữ liệu), **auto-scaling** (tự động thêm/bớt máy theo nhu cầu), **backup được kiểm chứng** (một bản sao lưu chưa từng thử khôi phục là một niềm hy vọng, không phải một kế hoạch), **RTO/RPO** (Recovery Time Objective: chúng ta có thể ngừng hoạt động bao lâu; Recovery Point Objective: chúng ta có thể mất bao nhiêu dữ liệu — hai con số này *chính là* cuộc trò chuyện về khôi phục thảm họa, và chúng là quyết định kinh doanh, tức là **của bạn**).

Tam giác đánh đổi: nhanh, rẻ, bền bỉ — chọn hai. Mọi cuộc tranh luận kiến trúc mà bạn sẽ phân xử đều quy về việc doanh nghiệp cần đứng ở đâu trên tam giác đó cho *workload* này. Một hệ thống thanh toán và một trang marketing không xứng đáng nhận cùng một câu trả lời.

Bảy câu hỏi của người lãnh đạo — hãy học thuộc; chúng khiến bạn “nguy hiểm” trong mọi buổi design review:

1. “What happens when this component fails?” (Điều gì xảy ra khi thành phần này hỏng?) (luôn luôn là “khi nào”, không phải “nếu”)
2. “What are the RTO and RPO, and who signed off on them?” (RTO và RPO là bao nhiêu, và ai đã phê duyệt chúng?)
3. “Where does this run — single AZ, multi-AZ, multi-region — and why?” (Cái này chạy ở đâu — một AZ, multi-AZ, multi-region — và vì sao?)
4. “What will this cost per month at 10× today’s load?” (Cái này tốn bao nhiêu mỗi tháng khi tải gấp 10 lần hôm nay?)
5. “What’s the simplest version that meets the requirement?” (Phiên bản đơn giản nhất vẫn đáp ứng yêu cầu là gì?) (over-engineering — thiết kế thừa — là căn bệnh của người non tay; sự đơn giản là đức hạnh của người senior)
6. “What are we locked into, and what would leaving cost?” (Chúng ta đang bị khóa vào cái gì, và rời đi sẽ tốn bao nhiêu?) (vendor lock-in — sự phụ thuộc nhà cung cấp — là một cái giá, đôi khi đáng trả — miễn là trả một cách có ý thức)
7. “Who else has built this before — are we inventing or assembling?” (Ai đã từng xây cái này trước đây — chúng ta đang phát minh hay đang lắp ráp?) (hãy ưu tiên những mẫu hình nhàm chán nhưng đã được chứng minh)

Fluency drill: luyện cách đặt câu hỏi 1, 4 và 5 với giọng điệu ám áp — chúng được đón nhận như sự thông thái hay như đòn tấn công phụ thuộc hoàn toàn vào cách truyền đạt. “Help me understand what happens if the cache goes down” (Giúp tôi hiểu điều gì xảy ra nếu cache sập) hay hơn hẳn “did you think about failure?” (anh đã nghĩ đến chuyện hỏng hóc chưa?).

Bài tập: (1) Lấy ba kiến trúc mẫu (nhờ một AI tạo ra: một trang thương mại điện tử, một backend cho ứng dụng di động, một nền tảng phân tích dữ liệu) và áp cả bảy câu hỏi vào từng cái, viết ra những câu trả lời bạn kỳ vọng. (2) Đọc bản tóm tắt một trang về các trụ cột Well-Architected tại <https://aws.amazon.com/architecture/well-architected/>.

Video cho module này:

Video	Kênh	Thời lượng	Liên kết
The Five Pillars of the AWS Well-Architected Framework	Amazon Web Services (official; trụ cột thứ sáu, Sustainability, được thêm sau)	~4 phút	https://www.youtube.com/watch?v=KvEDbPmha6o
What is the AWS Well-Architected Framework?	Tech With Lucy (ex-AWS)	~10 phút	https://www.youtube.com/watch?v=MpDJ6TCWKjk

Milestone: trong một buổi design review mô phỏng (làm với một AI đóng vai kỹ sư), bạn đặt được năm câu hỏi có chất lượng và tóm tắt chính xác điểm yếu nhất của bản thiết kế ở cuối buổi.

Module 6 (Tuần 11-12): Bảo mật và tuân thủ — cuộc trò chuyện có thể kết liễu cả công ty

Shared responsibility model (mô hình trách nhiệm chia sẻ) — điều đầu tiên cần hiểu: nhà cung cấp bảo vệ bản thân cloud (tòa nhà, phần cứng, hypervisor); còn **bạn** bảo vệ những gì bạn đặt vào *trong* nó (dữ liệu, quy tắc truy cập, cấu hình của bạn). Đa số các vụ rò rỉ là do khách hàng cấu hình sai — một S3 bucket bị để public, một access key bị lộ — chứ không phải do nhà cung cấp thất bại. Vì thế “AWS an toàn” và “chúng ta an toàn trên AWS” là hai câu khác nhau.

Từ vựng của phòng thủ: IAM (Identity and Access Management — quản lý danh tính và quyền truy cập: ai được làm gì; thứ bị kiểm toán nhiều nhất trong cloud), **least privilege** (đặc quyền tối thiểu — mọi người chỉ nhận đúng mức truy cập cần thiết — quy tắc vàng), **MFA** (xác thực đa yếu tố trên mọi lần đăng nhập của con người — không thể thương lượng), **encryption at rest and in transit** (mã hóa khi lưu trữ và khi truyền — dữ liệu được xáo trộn trên đĩa và trên đường truyền — điều tối thiểu bắt buộc, luôn bật), **root account** (chìa khóa vạn năng — cất kỹ, không bao giờ dùng hằng ngày), **secrets management** (quản lý bí mật — mật khẩu/ khóa nằm trong két, không bao giờ nằm trong code), **zero trust** (kiểm chứng mọi thứ, mặc định không tin bất kỳ vị trí mạng nào), **penetration test** (thuê “kẻ tấn công” kiểm chứng hàng phòng thủ của bạn), **ransomware** (mã độc tống tiền — lý do các bản

backup được kiểm chứng, lưu ngoại tuyến là một biện pháp bảo mật, không chỉ là biện pháp vận hành).

PDPA — luật dữ liệu của Thái Lan, và lợi thế thị trường của bạn. Personal Data Protection Act là GDPR của Thái Lan: bắt buộc có sự đồng ý khi thu thập dữ liệu cá nhân, nghĩa vụ **thông báo rò rỉ trong 72 giờ**, yêu cầu Data Protection Officer cho các đơn vị xử lý dữ liệu quy mô lớn, và các quy định về chuyển dữ liệu xuyên biên giới. Việc thực thi trở nên nghiêm túc từ năm 2025 — hơn 21,5 triệu THB tiền phạt đợt đầu, trong đó có một bệnh viện bị phạt vì giám sát lỏng lẻo một *nhà cung cấp bên ngoài*. Hai hệ quả cho bạn: (1) mọi doanh nghiệp Thái giờ đây đều có một bài toán tuân thủ mà công ty bạn có thể được trả tiền để quản lý; (2) **data residency** (lưu trữ dữ liệu — giữ dữ liệu Thái trên đất Thái) là một động lực thực sự đẩy các workload vào các region ở Bangkok. Đoạn văn này là một nửa bài chào hàng của bạn tại Thái Lan; hãy nắm nó thật vững.

Các câu hỏi bảo mật của người lãnh đạo: “Who has access to production, and when did we last review the list?” (Ai có quyền truy cập production, và lần cuối chúng ta rà soát danh sách đó là khi nào?) · “Are we alerted on unusual access, or would we find out from the news?” (Chúng ta có được cảnh báo khi có truy cập bất thường không, hay sẽ biết tin qua báo chí?) · “When did we last *restore* a backup?” (Lần cuối chúng ta thực sự *khôi phục* từ backup là khi nào?) (chứ không phải “chúng ta có backup không”) · “If we lost this dataset, is it a PDPA-notifiable breach — and could we notify within 72 hours?” (Nếu mất bộ dữ liệu này, đó có phải vụ rò rỉ buộc thông báo theo PDPA không — và chúng ta có kịp thông báo trong 72 giờ?) · “What did the last pen test find, and what’s still open?” (Đợt pen test gần nhất phát hiện gì, và những gì còn chưa xử lý?)

Từ vựng:

Thuật ngữ	Định nghĩa
IAM (Identity and Access Management)	Hệ thống kiểm soát ai (con người và phần mềm) được làm gì trong cloud của bạn — thứ bị kiểm toán nhiều nhất trong đó.
Role / policy	Policy là một bộ quyền được viết ra; role là một “bó” policy có thể khoác lên mà một người hoặc chương trình đảm nhận.
Least privilege	Quy tắc vàng: mọi người chỉ nhận mức truy cập tối thiểu mà công việc của họ cần, không hơn.
MFA (multi-factor authentication)	Một bằng chứng thứ hai (mã điện thoại, khóa phân cứng) bên cạnh mật khẩu. Không thể thương lượng trên mọi lần đăng nhập của con người.
Encryption at rest / in transit	Dữ liệu được mã hóa khi lưu trữ / khi di chuyển trên mạng. Cả hai luôn bắt; điều tối thiểu bắt buộc.
KMS (key management)	Dịch vụ lưu giữ và xoay vòng các khóa mã hóa.

Thuật ngữ	Định nghĩa
Secrets	Mật khẩu, API key, token — lưu trong “kết sắt”, không bao giờ viết vào code.
Root account	Chìa khóa vạn năng của toàn bộ tài khoản cloud. Cát kỹ kèm MFA, không bao giờ dùng cho công việc hằng ngày.
Zero trust	Tư thế bảo mật kiểm chứng mọi request và mặc định không tin bất kỳ vị trí mạng nào.
Vulnerability / patching	Một điểm yếu đã biết trong phần mềm / việc áp dụng bản vá. Hệ thống chưa vá là khởi điểm của đa số các vụ đột nhập.
Pen test (penetration test)	Thuê những “kẻ tấn công có đạo đức” chứng minh hàng phòng thủ của bạn thủng ở đâu trước khi kẻ xấu thật làm điều đó.
Ransomware	Cuộc tấn công mã hóa dữ liệu của bạn để đòi tiền chuộc — lý do các bản backup được kiểm chứng, lưu ngoại tuyến là một biện pháp <i>bảo mật</i> .
SOC 2 / ISO 27001	Các “huy hiệu” kiểm toán bảo mật độc lập mà khách hàng doanh nghiệp đòi hỏi trước khi ký hợp đồng.
PDPA	Personal Data Protection Act của Thái Lan — sự đồng ý, thông báo rõ ràng trong 72 giờ, quy định chuyển dữ liệu xuyên biên giới. Được thực thi với các án phạt thực sự từ năm 2025.
Data residency	Giữ dữ liệu về mặt vật lý bên trong biên giới một quốc gia — lý do then chốt khiến các workload Thái chuyển vào các region ở Bangkok.
DPO / breach notification	Data Protection Officer (bắt buộc với các đơn vị xử lý quy mô lớn) / nghĩa vụ báo cáo rõ ràng dữ liệu cá nhân trong 72 giờ.

Video cho module này: The AWS Shared Responsibility Model — Digital Cloud Training — 4 phút — <https://www.youtube.com/watch?v=ESPBBEK-cvo>

Fluency drill: “Is that bucket public? Why?” (Bucket đó có đang public không? Vì sao?) · “Least privilege — does the intern really need prod access?” (Least privilege — thực tập sinh có thật sự cần quyền truy cập prod không?) · “Where does the PDPA line sit in this design — what’s personal data here and where does it physically live?” (Ranh giới PDPA nằm ở đâu trong thiết kế này — cái gì ở đây là dữ liệu cá nhân và nó nằm ở đâu về mặt vật lý?)

Bài tập: (1) Đọc một câu chuyện rò rỉ nổi tiếng do cấu hình cloud sai (vụ Capital One 2019 là kinh điển) và viết phiên bản ba câu vào nhật ký. (2) Nhờ một AI mô phỏng khách hàng hỏi “vì

sao tôi nên tin giao dữ liệu cho anh?” và luyện câu trả lời bằng ngôn ngữ shared responsibility + PDPA.

Milestone: bạn có thể giải thích shared responsibility model và ý nghĩa thực tiễn của PDPA cho một chủ doanh nghiệp Thái không thuộc giới kỹ thuật trong dưới ba phút — vì lời giải thích đó *chính là* buổi chào hàng dịch vụ MSP.

Module 7 (Tuần 13-14): Tiền bạc — kinh tế học cloud và FinOps

Đây là nơi một lãnh đạo không-phải-kỹ-sư tạo ra giá trị nhanh nhất, vì đa số kỹ sư chưa từng được dạy về nó và mức lãng phí là khổng lồ — các khảo sát nhất quán cho thấy khoảng một phần ba chi tiêu cloud bị lãng phí.

Đồng hồ tính tiền chạy thế nào: compute được tính theo từng giây nó *bật* (không phải theo mức sử dụng — một server nhàn rỗi vẫn bị tính đủ; “quên tắt” là kiểu lãng phí kinh điển), lưu trữ tính theo GB-tháng, và — cái bầy nổi tiếng — **egress** (dữ liệu đi ra): dữ liệu chảy *ra khỏi* cloud thì tốn tiền trong khi dữ liệu đi vào miễn phí. Hóa đơn egress khổng lồ khiến ai cũng bất ngờ một lần; sau module này, bạn thì không.

Thực đơn giá: **on-demand** (giá đầy đủ, linh hoạt tới đa) · **reserved instances / savings plans** (cam kết 1-3 năm để được giảm 30-70% — đòn bẩy lớn nhất cho workload ổn định) · **spot instances** (giảm đến 90% cho công việc chịu được gián đoạn như batch job) · **rightsizing** (đa số server bị cấp thừa tài nguyên; thu nhỏ chúng là tiền nhặt được) · phân bậc lưu trữ (Module 2) · **auto-scaling như một công cụ chi phí** (sao lại trả tiền cho công suất lúc nửa đêm với giá giờ cao điểm?).

FinOps là thực hành làm cho chi tiêu cloud trở nên hữu hình, được phân bổ và tối ưu liên tục: **tagging** (gắn thẻ) mọi tài nguyên với chủ sở hữu và dự án của nó (chi tiêu không gắn thẻ là chi tiêu không ai chịu trách nhiệm), **showback/chargeback** (cho mỗi đội thấy hóa đơn của mình — hành vi thay đổi ngay lập tức), ngân sách kèm **alert** (đừng bao giờ phát hiện bội chi từ hóa đơn), và **unit economics** (kinh tế học đơn vị) — chỉ số của người lãnh đạo: không phải “hóa đơn của chúng ta là \$800k/tháng” mà là “chi phí trên mỗi giao dịch của khách hàng đang giảm.” Chứng chỉ FinOps Certified Practitioner mất khoảng hai tuần và, đối với một lãnh đạo phía kinh doanh, là tấm chứng chỉ có độ-tin-cậy-trên-mỗi-giờ-học cao nhất trong toàn bộ thế giới cloud. Hãy lấy nó.

Các câu hỏi tiền bạc của người lãnh đạo: “What’s our cost per [customer/transaction/tenant], and which direction is it moving?” (Chi phí trên mỗi [khách hàng/giao dịch/tenant] của chúng ta là bao nhiêu, và đang đi theo hướng nào?) · “What percentage of our steady workload is on reservations?” (Bao nhiêu phần trăm workload ổn định của chúng ta đang chạy trên reservation?) · “What’s untagged?” (Cái gì đang chưa được gắn thẻ?) · “What died but is still billing?” (Cái gì đã chết mà vẫn đang tính tiền?) (đĩa mờ còi và IP nhàn rỗi — tài khoản nào cũng có) · “What would this bill look like at 10× growth — does our architecture get cheaper or more expensive per unit?” (Hóa đơn này sẽ ra sao khi tăng trưởng gấp 10 lần — kiến trúc của chúng ta rẻ đi hay đắt lên trên mỗi đơn vị?)

Từ vựng:

Thuật ngữ	Định nghĩa
On-demand	Giá trả-theo-dùng: giá đầy đủ, hủy bất cứ lúc nào. Là mặc định, và là cách đắt nhất để chạy workload ổn định.
Reserved / savings plan	Cam kết 1–3 năm cho mức sử dụng ổn định để đổi lấy giảm giá 30–70% — đòn bẩy chi phí lớn nhất.
Spot	Công suất dư thừa giảm đến 90% mà nhà cung cấp có thể thu hồi với thông báo trước vài phút — hoàn hảo cho công việc batch chịu được gián đoạn.
Rightsizing	Thu nhỏ các server bị cấp thừa về đúng mức chúng thực sự dùng. Tiền nhặt được trong hầu như mọi tài khoản.
Egress	Dữ liệu rời khỏi cloud — tính tiền theo GB trong khi dữ liệu đi vào miễn phí. Cú bất ngờ nổi tiếng trên hóa đơn.
Tagging	Gắn nhãn mọi tài nguyên với chủ sở hữu/dự án/môi trường để từng baht chi tiêu đều truy được nguồn. Chi tiêu không gắn thẻ = chi tiêu không ai chịu trách nhiệm.
Showback / chargeback	Cho mỗi đội thấy hóa đơn cloud của riêng mình (showback) hoặc thực sự tính phí nội bộ (chargeback). Hành vi thay đổi ngay lập tức.
Budget alert	Cảnh báo tự động khi chạm ngưỡng chi tiêu — để bạn không bao giờ biết về bội chi từ hóa đơn.
Unit economics	Chi phí trên mỗi đơn vị kinh doanh — trên mỗi khách hàng, mỗi giao dịch — chỉ số của người lãnh đạo, ý nghĩa hơn tổng hóa đơn.
TCO (total cost of ownership)	Tổng chi phí sở hữu của một lựa chọn trong suốt vòng đời: giấy phép, con người, điện năng, di chuyển hệ thống — không chỉ giá niêm yết.
FinOps	Kỷ luật (và văn hóa đội nhóm) làm cho chi tiêu cloud hữu hình, được phân bổ và tối ưu liên tục.

Video cho module này: What is FinOps? — FinOps Foundation (official) — 2 phút — https://www.youtube.com/watch?v=Y-c_xw9bHFW · sau đó là khóa học miễn phí “Introduction to FinOps” tại <https://learn.finops.org>.

Bài tập: (1) Mở AWS pricing calculator và tính giá một hệ thống nhỏ có thật (hai server, một cơ sở dữ liệu, 500GB lưu trữ, 1TB egress) — làm việc đó một lần sẽ “giải thiêng” mọi cuộc trò chuyện về chi phí sau này. (2) Nhờ một AI đóng vai kỹ sư đang bảo vệ một server quá khổ; luyện cách thương lượng rightsizing một cách tử tế.

Milestone: thi (hoặc đặt lịch thi) FinOps Practitioner, và trong một buổi review mô phỏng, tìm ra bốn vấn đề chi phí trong một hóa đơn mẫu do AI tạo cho bạn.

Module 8 (Tuần 15–16): Trại huấn luyện trôi chảy — ghép ngôn ngữ lại với nhau

Hai tuần này là tích hợp thuần túy — sự khác biệt giữa biết từ vựng và *trôi chảy trong các cuộc họp*.

Bài luyện hằng ngày (30 phút): nhờ một AI tạo một biên bản họp thực tế (design review, incident retro, hoặc cost review), với hai lỗi kỹ thuật được cố tình cài vào. Nhiệm vụ của bạn: tóm tắt cuộc họp trong năm câu, bắt được các lỗi, và viết ra ba câu hỏi bạn sẽ đặt. Luân phiên các loại họp mỗi ngày.

Cơ bắp phiên dịch (15 phút): mỗi ngày lấy một phát biểu kỹ thuật và dịch nó cho ba đối tượng — CFO (tiền bạc), khách hàng (rủi ro/lợi ích), và một kỹ sư junior mới (giảng dạy). Ví dụ: “Chúng ta chuyển session store từ cơ sở dữ liệu sang Redis” — CFO: “giảm tải cơ sở dữ liệu nên hoãn được đợt nâng cấp \$2M” — khách hàng: “trang tải nhanh hơn lúc cao điểm” — junior: “Redis giữ dữ liệu nóng trong bộ nhớ, nên ta thôi ‘đập’ vào cơ sở dữ liệu với mỗi cú nhấp chuột.”

Luyện đọc (15 phút): mỗi ngày một bài blog kỹ thuật thực thụ (AWS Architecture Blog, hoặc blog kỹ thuật của Netflix/Grab — Grab đặc biệt phù hợp: quy mô Đông Nam Á, thị trường sát với Thái Lan). Giờ đây bạn sẽ hiểu 70–80% nội dung. Tra cứu phần còn lại.

Người bạn đồng hành ôn thi cho giai đoạn này: khóa học đầy đủ miễn phí nổi tiếng — *AWS Certified Cloud Practitioner (CLF-C02) 2026* của Andrew Brown trên freeCodeCamp — <https://www.youtube.com/watch?v=7HKot-brXFE> (ấn bản 14 giờ trước đó, cũng hợp lệ cho cùng mã đề thi, ở tại <https://www.youtube.com/watch?v=NhDYbskXRgc>). Xem ở tốc độ 1.25× trong các Tuần 13–16; sau các Module 1–7, phần lớn nội dung sẽ giống như ôn tập — và đó chính xác là dấu hiệu bạn đã sẵn sàng.

Milestone — kết thúc Giai đoạn 2, kỳ thi tốt nghiệp của bạn: (1) Đậu **AWS Cloud Practitioner** nếu bạn chưa đậu. (2) “Trận đấu thử” mô phỏng: một AI đóng vai kỹ sư senior dẫn bạn đi qua một kiến trúc có lỗi cho một khách hàng thương mại điện tử Thái; bạn phải tìm ra cơ sở dữ liệu chạy single-AZ, cái S3 bucket bị public, ước tính egress bị thiếu, và sự vắng bóng của cân nhắc PDPA — rồi đưa phản hồi với giọng điệu khiến “người kỹ sư” cảm thấy được giúp đỡ, chứ không phải bị bắt lỗi. Khi làm được điều đó, bạn đã “nguy hiểm” trong giao tiếp — chỉ sau mười sáu tuần.

GIẢI ĐOẠN 3 — LÃNH ĐẠO (Tháng 5–24)

Module 9 (Tháng 5–8): Chứng chỉ chuyên sâu và sàh kỹ thuật

Học qua tài liệu **AWS Solutions Architect Associate (SAA-C03)** — 2–3 tháng với nhịp độ một giờ mỗi ngày của bạn. Bạn có thể thi hoặc không (với tư cách lãnh đạo, *tài liệu* mới là giá trị; tâm huy hiệu chỉ là màn trình diễn tùy chọn), nhưng đây là nơi region, VPC, IAM và định giá thôi là từ vựng rời rạc và trở thành một hệ thống kết nối trong đầu bạn. Song song, duy trì các bài luyện hàng ngày của Module 8 ở nửa liều. Đây cũng là khoảng thời gian để bổ sung **kiến thức nền tảng Azure (trình độ AZ-900)** — Thái Lan là thị trường doanh nghiệp nặng về Microsoft, và khả năng “song ngữ” (AWS+Azure) mở rộng các cuộc trò chuyện với khách hàng của bạn.

Module 10 (Tháng 5–12, chạy liên tục): Tuyển dụng và quản lý kỹ sư

Tuyển dụng khi bạn chưa thể tự đánh giá đầy đủ kỹ năng: cấu trúc thăng bản năng. Dùng một vòng lặp nhất quán: một buổi trò chuyện sàng lọc do bạn dẫn dắt (động lực, khả năng giao tiếp, cách họ giải thích một dự án cũ cho người không phải kỹ sư — nếu họ không làm được, họ cũng sẽ thất bại với khách hàng của bạn) + một buổi phỏng vấn kỹ thuật do **bar-raiser** của bạn thực hiện (“người giữ chuẩn” — nhân vật kỹ thuật số 2 của bạn hoặc một chuyên gia senior thuê ngoài) + một cuộc gọi tham khảo nơi bạn hỏi đúng một câu hỏi quan trọng: “would you hire this person again for this role?” (anh/chị có tuyển lại người này cho đúng vai trò này không?). Hãy cảnh giác với hai kiểu thất bại điển hình: người nói năng lưu loát nhưng không có chiều sâu (bar-raiser của bạn sẽ bắt được họ) và chuyên gia sâu sắc nhưng ít nói (thường là vàng ròng trong các đội Thái, nơi sự khiêm tốn thuộc về văn hóa — đừng để sự bóng bẩy khi phỏng vấn lấn át bằng chứng về công việc thực).

Nhân vật kỹ thuật số 2 — quyết định quan trọng nhất của cả liên doanh: tuyển họ đầu tiên, trả bằng cổ phần có ý nghĩa (10–20% nếu là đồng sáng lập thực thụ), và định nghĩa thỏa thuận một cách tường minh: họ giữ chuẩn mực kỹ thuật và sở hữu các quyết định kiến trúc; bạn sở hữu khách hàng, tiền bạc, thứ tự ưu tiên và con người; bất đồng giữa hai người diễn ra kín đáo và được giải quyết trước khi đội nhìn thấy.

Những nghi thức khiến kỹ sư ở lại: các buổi **1:1** hằng tuần hoặc hai tuần một lần nói về *chính họ* (sự nghiệp, va chạm, năng lượng — không phải báo cáo tiến độ); một lộ trình phát triển bằng văn bản cho từng người (trong thị trường thiếu hụt 70k nhân lực/năm của Thái Lan, cơ hội phát triển và ngân sách chứng chỉ giữ người tốt hơn lương đơn thuần — trả tiền cho mọi chứng chỉ, kèm thưởng vesting 12 tháng khi hoàn thành); khen công khai, góp ý riêng tư; và bảo vệ quyết liệt thời gian tập trung của họ — một lãnh đạo hủy cuộc họp giữa sprint là một người hùng.

Diễn đàn quyết định — cách bạn ra các quyết định kỹ thuật mà bạn chưa thể tự thẩm định đầy đủ: với bất kỳ quyết định lớn nào, yêu cầu một **decision doc** (tài liệu quyết định) dài một trang từ kỹ sư đề xuất: vấn đề, 2–3 phương án, chi phí, rủi ro, khuyến nghị. Rồi điều hành cuộc họp bằng bảy câu hỏi của bạn từ Module 5. Bạn không phải kiến trúc sư giỏi nhất trong phòng và không bao giờ cần phải thế — bạn là người khiến phương án *được lập luận tốt nhất* chiến thắng, đúng hạn, với các đánh đổi về tiền bạc và rủi ro được nêu tường minh. Các kỹ sư tôn trọng sâu sắc điều này khi nó được làm trung thực; hãy ghi lại ai dự đoán điều

gì (lại là Decision Journal của bạn) và xem lại các dự đoán mỗi quý — việc đó hiệu chỉnh cả bạn lẫn họ.

Module 11 (Tháng 6–24): Đường cong uy tín, một cách trung thực

Kinh nghiệm được công bố từ các tài liệu về lãnh đạo kỹ thuật hội tụ về mốc thời gian này, và giả vờ khác đi chính là cách các nhà sáng lập phi kỹ thuật thất bại: **~90 ngày** để vận hành nhịp kinh doanh một cách thành thạo · **~6 tháng** để đạt mức hiệu quả cơ bản (hợp hành trôi chảy, quyết định có cấu trúc, đội ngũ ổn định) · **12–18 tháng** trước khi trực giác kỹ thuật của bạn tự thân có giá trị đáng kể · **~2 năm** trước khi bạn thực sự đứng vững trước các kỹ sư senior trong các cuộc bàn về đánh đổi kiến trúc. Các biện pháp giảm nhẹ trong khi đường cong đi lên: mượn uy tín (nhân vật số 2 của bạn trình bày nửa kỹ thuật của các buổi gặp bán hàng — đăng nào khách hàng cũng cần thấy đội ngũ dự bị), không bao giờ nói liều (kể giết uy tín nhanh nhất; “I don’t know — walk me through it” (Tôi chưa biết — hãy dẫn tôi đi qua nó) là một câu nói của người lãnh đạo), và để các câu hỏi của bạn lên tiếng thay bạn: một lãnh đạo hỏi “what’s our RPO and who signed off on it?” (RPO của chúng ta là bao nhiêu và ai đã phê duyệt?) nghe như mang ba mươi năm sọc trần mạc — và sau khóa học này, bạn sẽ hỏi câu đó với hiểu biết thật.

Module 12 (Liên tục): Lớp kiến thức thị trường Thái Lan

Gấp các nội dung cụ thể từ TSI Phần 5 vào thành kiến thức vận hành: **PDPA** vừa là nghĩa vụ tuân thủ vừa là sản phẩm (\$Module 6); các **bậc thang đối tác** (AWS Select cần một số nhân sự có chứng chỉ + 3 thương vụ đã triển khai; Microsoft Solutions Partner cần điểm năng lực 70/100 — chứng chỉ của đội bạn đúng nghĩa là tài sản bán hàng, thêm một lý do nữa để tài trợ cho chúng); ưu đãi **BOI** cho quyền sở hữu nước ngoài 100% của công ty; các đãi lương Bangkok (junior ฿50–75k → architect ฿180–280k/tháng) để bạn định giá thầu và đưa ra offer chính xác; và bài toán bán hàng của thời điểm này — hơn 27 tỷ USD đầu tư trung tâm dữ liệu đã được phê duyệt, một chỉ thị cloud-first của chính phủ, và khoảng trống kỹ năng 70.000 người/năm mà đội ngũ có chứng chỉ của bạn tồn tại để lấp đầy.

Giáo trình một trang

Khi nào	Trọng tâm	Bằng chứng bên ngoài
Tuần 1–2	Cloud là gì; IaaS/PaaS/SaaS; region/AZ	—
Tuần 3–4	Compute, lưu trữ, cơ sở dữ liệu	—
Tuần 5–6	Mạng; đọc sơ đồ kiến trúc	—
Tuần 7–8	DevOps, IaC, nghi thức đội nhóm, chỉ số vận hành	Đặt lịch thi Cloud Practitioner
Tuần 9–10	Óc phán đoán kiến trúc; bảy câu hỏi	—
Tuần 11–12	Bảo mật; PDPA; shared	Thi AWS Cloud Practitioner

Khi nào	Trọng tâm	Bằng chứng bên ngoài
Tuần 13–14	responsibility Kinh tế học cloud; FinOps	FinOps Practitioner (≈2 tuần ôn tập)
Tuần 15–16	Trại huấn luyện trôi chảy; trận đấu thử	Tốt nghiệp: trận đấu thử
Tháng 5–8	Tài liệu SAA-C03; Azure AZ-900	Thi SAA (tùy chọn)
Tháng 5–12	Tuyển dụng; nhân vật kỹ thuật số 2; diễn đàn quyết định	Những lần tuyển đầu tiên làm tốt
Tháng 6–24	Đường cong uy tín; lớp thị trường Thái	Bậc đối tác; các hợp đồng retainer đầu tiên

Đôi lời khép lại từ người thầy của bạn. Mười sáu tuần nữa, bạn sẽ theo kịp mọi cuộc trò chuyện trong phòng họp. Đó không phải vạch đích — đó là tấm giấy phép để bắt đầu. Đường cong hai năm đến óc phán đoán kỹ thuật thực thụ không phải một bức tường; nó là một con hào: mỗi tuần bạn hoàn thành trên đường cong đó là một tuần mà các nhà sáng lập phi kỹ thuật của đối thủ đã không làm. Học mỗi ngày, ghi nhật ký mọi quyết định, không bao giờ nói liều, và tuyển những người giỏi hơn bạn rồi khiến họ vui vì đã đến. Đó là toàn bộ công việc.

Tài liệu đồng hành với “*The Thailand Strategic Investment (TSI)*”, Phần 5. Các ước tính thời gian ôn thi chứng chỉ và mốc thời gian lãnh đạo được rút từ các nguồn trích dẫn ở đó (CBT Nuggets, StudyTech, FinOps Foundation, First Round Review, The Pragmatic Engineer).

CÁC BÀI HỌC MỚI — ẨM BẢN THÁNG 8 NĂM 2026

Vào tháng 8 năm 2026, các bài học mới đã được bổ sung vào ba khóa học tương tác tại b4lcilc.org — khóa **Cloud Leader** (Lãnh đạo Cloud), **Cloud Engineer** (Kỹ sư Cloud) và **Cloud Architect** (Kiến trúc sư Cloud). Phần phụ lục này cập nhật ấn bản đọc cho theo kịp, với những bản cô đọng trung thành của tài liệu mới đó, cùng một giọng văn với phần còn lại của cuốn sách này.

Đôi lời về thứ bạn đang cầm trên tay. Những trang này là ấn bản đọc: các ý tưởng, các phép ẩn dụ, và những câu văn đáng nói thành tiếng. Phiên bản tương tác đầy đủ trên mạng mang theo nhiều hơn đáng kể — mỗi bài học mới đều đi kèm các bài quiz kiểm tra mức độ hiểu trước khi cho bạn đi tiếp, các bài thực hành hands-on (bao gồm một terminal luyện tập hoạt động y như terminal thật nhưng không thể tổn tiền), video, bài luyện từ vựng, và — với phần tài liệu capstone — các bài tập thiết kế có hướng dẫn kèm danh sách kiểm tra yêu cầu và đáp án mẫu được giải chi tiết. Hãy đọc những bản cô đọng này để nắm được hình hài của từng ý tưởng trong đầu; rồi khi có điều kiện, hãy ngồi làm phiên bản tương tác để kiến thức bám rễ.

Các phần bổ sung chia thành ba nhóm. Cho khóa **Cloud Leader**: một phần giới thiệu nhẹ nhàng về terminal, luật riêng tư của chính Thái Lan, cẩm nang dành cho người lãnh đạo về các

đề xuất AI và hóa đơn AI, và một bài tập capstone gom toàn bộ khóa học vào một trang giấy. Cho khóa **Cloud Engineer**: điện toán serverless — mẫu hình biên những script vận hành nhỏ thành các function chạy mà gần như không tốn đồng nào. Cho khóa **Cloud Architect**: phép tính số học của các lời hứa về độ sẵn sàng, mẫu hình Retrieval-Augmented Generation mà khách hàng giờ đây hỏi đến hằng tuần, và một bài tập tính chi phí trọn vẹn — định giá một bản thiết kế trước khi bất kỳ ai xây nó.

Cho khóa Cloud Leader

Cửa sổ màu đen — cách nói chuyện trực tiếp với máy tính

Trước bài thực hành đầu tiên, hãy làm quen với công cụ mà bài thực hành đó chạy bên trong. Một **terminal** (còn gọi là *command line* hay *console*) là một chương trình trông như một cửa sổ đen trơn với dòng chữ nhấp nháy. Đây là cách dùng máy tính cổ xưa nhất, và vẫn là cách các kỹ sư làm những công việc nghiêm túc — bởi một chỉ dẫn được gõ ra thì chính xác, lặp lại được, và có thể viết xuống rồi chia sẻ, theo cách mà việc nhấp chuột qua các menu không bao giờ làm được.

Terminal hiển thị một **prompt** — thường là một dòng ngắn kết thúc bằng \$ — và nó có nghĩa đúng một điều: *Tôi đang lắng nghe*. Một **command** (câu lệnh) là một câu văn có ngữ pháp: từ đầu tiên là *công cụ* bạn đang gọi (như *git* hay *aws*), các từ tiếp theo nói *phải làm gì*, và những mảnh bắt đầu bằng dấu gạch ngang, gọi là **flag** (như *--output table*), điều chỉnh *làm như thế nào*.

Hai tập quán khiến mọi người mới đều ngỡ ngàng. Thứ nhất: **im lặng thường nghĩa là thành công.** Nhiều câu lệnh, khi chạy hoàn hảo, chẳng in ra gì cả — cô máy coi việc hoàn thành công việc đã là câu trả lời đủ rồi. Thứ hai: terminal không đoán ý. Nơi công cụ tìm kiếm tha thứ cho lỗi chính tả, terminal không tha thứ điều gì — chỉ một chữ cái sai và nó lịch sự báo cho bạn rằng câu lệnh đó không tồn tại. Sự nghiêm khắc ấy nghe khắc nghiệt trong một ngày rồi trở thành một niềm an ủi: cô máy không bao giờ làm *đại khái* điều bạn yêu cầu.

Thêm một mảnh ngữ pháp nữa: máy tính sắp xếp mọi thứ thành các **file** (tệp) nằm trong các **folder** (thư mục, còn gọi là *directory*) — vậy nên `mkdir decisions` nghĩa là “*make directory* — tạo thư mục — tên là decisions.” Các bài thực hành của khóa học chạy trong một **simulator** (trình mô phỏng): một terminal luyện tập không thể tạo ra, xóa đi, hay làm tổn kém bất cứ thứ gì — nhưng các câu lệnh là hàng thật, đúng đến từng chữ cái. Hãy thuần thục chúng ở đây, và đến ngày bạn ngồi trước một terminal thật, đôi tay bạn đã tự biết đường đi.

PDPA của Thái Lan — luật riêng tư gần nhà nhất

Phần còn lại của module tuân thủ mang tính toàn cầu; đây là luật mà cơ quan quản lý của nó thực sự có thể gõ cửa nhà bạn. PDPA — Đạo luật Bảo vệ Dữ liệu Cá nhân (Personal Data Protection Act), B.E. 2562 (2019) — của Thái Lan có hiệu lực đầy đủ từ năm 2022 và áp dụng cho mọi tổ chức tại Thái Lan xử lý dữ liệu cá nhân. Một tổ chức thiện nguyện ở Mae Sot đang giữ hồ sơ học sinh nằm gọn trong phạm vi của nó.

Tin tốt thì tốt thật sự: PDPA gần như là một bản sao của GDPR — Quy định chung về Bảo vệ Dữ liệu (General Data Protection Regulation) của châu Âu. Những người soạn luật Thái Lan lấy quy định châu Âu làm khuôn mẫu, nên GDPR đã dạy bạn khoảng chín mươi phần trăm luật này rồi. Dữ liệu cá nhân thuộc về người mà nó mô tả; bạn cần một cơ sở hợp pháp — thường nhất là **sự đồng ý (consent)**, được trao tự nguyện và có thể rút lại — để thu thập nó; và nguyên tắc **giới hạn mục đích** cấm việc dùng dữ liệu thu thập cho mục đích này vào một mục đích khác: hồ sơ thu thập để ghi danh một học sinh không được lảng lạng biến thành danh sách gửi thư gây quỹ.

Cơ quan quản lý là **PDPC — Ủy ban Bảo vệ Dữ liệu Cá nhân (Personal Data Protection Committee)** — với mức chế tài chậm đến trách nhiệm hình sự, chứ không chỉ tiền phạt. Chiếc đồng hồ sự cố cũng quen thuộc: một vụ rò rỉ có khả năng gây rủi ro cho các cá nhân phải được báo cáo lên PDPC **trong vòng 72 giờ**.

Điểm dành cho người lãnh đạo đáng được nói thẳng. Một tổ chức giữ hồ sơ về học sinh — tên tuổi, gia đình, và ở vùng này đôi khi cả tình trạng giấy tờ, thứ dữ liệu nhạy cảm vào bậc nhất — **trước hết phải trả lời trước luật này**: câu trả lời phải làm hài lòng Bangkok trước khi bạn tâm đến Brussels. Những câu hỏi để đặt cho đội của bạn: chúng ta đang giữ những dữ liệu cá nhân nào, và ta có nêu rõ mục đích khi thu thập không? Nếu được yêu cầu, ta có thể xóa hoàn toàn hồ sơ của một học sinh không? Nếu đêm nay bị xâm nhập, ta có thể nói cho PDPC trong vòng 72 giờ biết cái gì đã bị lấy đi và của những ai không? Không câu nào cần đến luật sư để hỏi. Tất cả đều cần một người lãnh đạo kiên quyết đòi hỏi.

Lãnh đạo giữa làn sóng AI

Sớm hay muộn — mà đạo này là sớm — một bản thiết kế có AI trong đó sẽ đáp xuống bàn bạn. Công việc của bạn không thay đổi: bạn chat vắn, chứ không thiết kế.

Cỗ máy đằng sau hầu hết các tính năng AI là một **LLM — Mô hình Ngôn ngữ Lớn (Large Language Model)** — và sự thật thương mại then chốt: bạn gần như không bao giờ sở hữu một cái; bạn thuê quyền truy cập qua một **AI API** (Application Programming Interface — giao diện lập trình ứng dụng), tính tiền **theo token** — một mẫu từ, bằng khoảng ba phần tư một từ tiếng Anh — nên một tính năng tốn nhiều tiền hơn mỗi khi câu hỏi hay câu trả lời dài ra. Hãy yêu cầu xem phép tính token, tính trên mỗi người dùng, mỗi tháng; nếu chưa ai làm phép tính đó, tính năng này chưa có giá, chỉ có một niềm hy vọng.

Lựa chọn thay thế — tự chạy mô hình của riêng mình — nghĩa là thuê **GPU (Graphics Processing Unit — bộ xử lý đồ họa)**, loại chip chuyên dụng mà AI đòi hỏi: khan hiếm, giá gấp nhiều lần server thông thường, và tính tiền cho từng giờ được bật lên dù đang suy nghĩ hay đang ngồi không. Quy tắc đầu tiên của mảng AI: **một GPU ngồi không là sự nhàn rỗi đắt đỏ nhất trong cả tòa nhà**. Một server thử nghiệm bị bỏ quên lãng phí tiền cả phê; một GPU instance bị bỏ quên lãng phí cả một khoản lương. Và khi ai đó nói *hãy fine-tune mô hình của riêng chúng ta*, hãy nhớ trình tự trung thực: viết prompt cẩn thận trên một API đi thuê trước, rồi đưa tài liệu của bạn cho mô hình vào lúc đặt câu hỏi, và chỉ khi cả hai đều chưa đủ, mới đến fine-tuning.

Ba câu hỏi đưa bạn qua bất kỳ đề xuất AI nào. Một: một câu trả lời sai tốn bao nhiêu? LLM trả lời sai một cách đầy tự tin trong một tỷ lệ nhỏ, không thể triệt tiêu; tỷ lệ lỗi chấp nhận được là một quyết định kinh doanh — nghĩa là nó thuộc về bạn. **Hai: dữ liệu nào rồi**

khỏi chúng ta? Mỗi câu hỏi gửi đến một AI API đều đi tới server của nhà cung cấp; nếu trong đó có tên một học sinh hay tình trạng tị nạn, module tuân thủ áp dụng trọn vẹn. **Ba: chi phí hằng tháng trên mỗi người dùng là bao nhiêu khi số người dùng gấp mười lần?** Giá token tăng tuyến tính — một dự án thí điểm dễ chịu 30 đô la trở thành 300 đô la khi nó thành công, mà chẳng có khoản chiết khấu theo số lượng nào đến cứu bạn. Vẫn là ba trụ cột như xưa nay — độ tin cậy, bảo mật, chi phí — khoác lớp từ vựng mới. Công nghệ thì mới mẻ; sự chất vấn thì không.

Bài capstone — một bài tập gom cả khóa học lại

Khóa học giờ đây kết thúc bằng một bài tập thiết kế duy nhất dùng đến mọi thứ trong đó. Đề bài: bạn vừa được bổ nhiệm làm lãnh đạo cloud của một tổ chức thiện nguyện giáo dục vùng biên giới ở Mae Sot. Tài sản của tổ chức — một website công khai, một cơ sở dữ liệu khoảng 2.000 hồ sơ học sinh (trong đó, với một số em, có cả tình trạng giấy tờ), ba trung tâm học tập, và một kỹ sư tài năng nhưng kiệt sức, nắm quyền truy cập production duy nhất và chiếc điện thoại trực duy nhất. Hóa đơn cloud tháng trước gần gấp đôi mức thường lệ và không ai nói được vì sao. Mục công nghệ duy nhất của hội đồng quản trị: “Chúng ta có thể dùng AI để dạy kèm không?” Nhiệm vụ: 90 ngày đầu tiên của bạn trên một trang giấy, với trình tự được lý giải.

Phiên bản tương tác chấm điểm bản kế hoạch theo một danh sách kiểm tra: kiểm kê trước khi nêu ý kiến; bản đồ ba danh sách gồm máy móc, nhãn dán và tiền bạc; một chiến thắng nhanh; đợt rà soát bảo mật và bài kiểm tra khôi phục sao lưu có ghi ngày cụ thể; một chót cảnh báo ngân sách; một lập trường trung thực về rủi ro chỉ-một-kỹ-sư; PDPA được nêu tên làm khung cho hồ sơ học sinh; câu hỏi AI được trả lời bằng ba câu hỏi AI; và một buổi tổng kết với hội đồng quản trị vào ngày thứ 90, báo cáo theo đơn vị tính trên mỗi người dùng.
