

El Curso del Líder Cloud (The Cloud Leader Course)

De cero conocimiento a liderar un equipo de ingenieros cloud — preparado para Fedly Bonneau

13 de agosto de 2026

El Curso del Líder Cloud (The Cloud Leader Course)

De cero conocimiento a liderar con fluidez un equipo de ingenieros cloud.

Cómo funciona este curso

Usted no se está formando para *ser* ingeniero cloud. Se está formando para **liderar** ingenieros cloud: para seguir cada conversación en la sala, hacer las preguntas que importan, tomar buenas decisiones sobre dinero, riesgo y personas, y ganarse el respeto de los expertos sin fingir ser uno de ellos. Esa es una habilidad distinta y totalmente aprendible — y es la habilidad que este curso enseña.

El curso tiene tres etapas. **Etapas 1 (Semanas 1–8): Hable el idioma.** Aprende qué es realmente la nube (cloud) y el vocabulario de cada componente básico, para que las reuniones dejen de sonar a ruido. **Etapas 2 (Semanas 9–16): Piense como la sala.** Arquitectura, seguridad y dinero — las tres conversaciones que todo equipo cloud tiene cada semana, y las tres donde un líder aporta valor o queda ignorado. **Etapas 3 (Meses 5–24): Lidere.** Certificaciones, contratación, dirigir reuniones de decisión, y el camino honesto de dos años para sostenerse por sí mismo frente a ingenieros senior.

Reglas para todo el curso: estudie **una hora al día, seis días a la semana** — la constancia vence a la intensidad. Cada módulo termina con un **Drill de fluidez** (frases que usted dice en voz alta hasta que le salgan con naturalidad) y un **Hito** (la prueba de que está listo para avanzar). Hágalos; leer solamente construye reconocimiento, no fluidez. Y desde la Semana 1, lleve un **Diario de Decisiones**: cada vez que aprenda un concepto, escriba una frase sobre cómo afecta al dinero, al riesgo o a las personas — porque esa traducción es todo su trabajo.

ETAPA 0 — LA PLANTA BAJA (empiece aquí si la tecnología es nueva para usted)

Estos dos módulos son para el lector que nunca ha trabajado en tecnología en absoluto. Son los mismos que los Módulos 1 y 2 del curso interactivo en b4lcilc.org. Si usted ya sabe qué son un servidor web y una base de datos, hojéelos; si no lo sabe, son las páginas más importantes de este libro — todo lo que viene después se apoya en ellas.

Módulo A: Cómo funciona la web — qué pasa cuando usted escribe una dirección

La gran idea: una página web no es una imagen transmitida a su teléfono. Es un conjunto de **instrucciones escritas** enviadas desde una computadora lejana, que su propio navegador sigue para construir la página en su pantalla. Aprenda los seis pasos entre escribir una dirección y ver una página, y cada reunión de tecnología por el resto de su carrera se volverá más fácil de seguir.

El viaje, en seis pasos que usted debería poder decir en voz alta:

1. Usted escribe un nombre — `b4lcilc.org` — en un **navegador** (Chrome, Safari, Edge, Firefox).
2. El **DNS**, la guía telefónica de internet, convierte ese nombre en una **dirección IP** — la dirección numérica de una computadora, como `142.250.66.14`.
3. Su navegador — el **cliente**, el que pregunta — envía una **petición HTTP** a la computadora en esa dirección: “TRÁEME esta página.”
4. La petición viaja en pequeños **paquetes** bajo las reglas de **TCP/IP**, la fontanería de internet, que numera cada paquete, confirma la entrega y reenvía cualquier cosa que se pierda.
5. La computadora lejana — el **servidor**, el que responde — devuelve una **respuesta**: las instrucciones de la página, o una excusa numerada como **404** (“no existe tal página”).
6. Su navegador sigue esas instrucciones y construye la página. Las instrucciones vienen en tres lenguajes: **HTML** (estructura — encabezados, párrafos, botones), **CSS** (apariencia — colores, tamaños, disposición) y **JavaScript** (comportamiento — qué pasa cuando usted hace clic).

Del lado del cliente y del lado del servidor — el marco que lo organiza todo: el trabajo hecho en su dispositivo, en el navegador, es *del lado del cliente* (el **front end**); el trabajo hecho en la computadora lejana es *del lado del servidor* (el **back end**). La regla práctica que vale la pena memorizar: **todo lo que involucra confianza o memoria pertenece al servidor; todo lo que involucra sensación de rapidez pertenece al cliente**. Verificar una contraseña, guardar un registro, cobrar una tarjeta: servidor. Dibujar la página, respuesta instantánea mientras usted escribe: cliente. Entre los dos suele estar un **proxy** — un servidor intermedio que filtra, protege o distribuye el tráfico; los “load balancers” (balanceadores de carga) y “CDN” de módulos posteriores son proxies con especialidades.

Por qué el DNS le importa a un líder: el DNS falla de una manera especial — cuando se rompe, todos los sitios “desaparecen” a la vez aunque todos los servidores sigan funcionando. La guía telefónica está caída, no los teléfonos. Cuando un ingeniero dice “it might be DNS” (podría ser el DNS), eso es lo que quiere decir, y algunas de las caídas más famosas de internet fueron exactamente esto.

Y el candado: HTTPS es HTTP con cifrado — toda la conversación va codificada contra cualquiera que esté en medio, que es lo que certifica el candado en la barra de direcciones. La regla moderna: todos los sitios, siempre.

Vocabulario:

Término	Definición
Navegador web	El programa que trae las páginas y las dibuja: Chrome, Safari, Edge, Firefox.
URL	La dirección completa que usted escribe: protocolo + nombre de la computadora + qué documento quiere.
Dirección IP	La dirección numérica de una computadora en una red, como 142.250.66.14.
DNS	La guía telefónica de internet: convierte nombres (google.com) en direcciones IP.
HTTP / HTTPS	Las reglas de petición-y-respuesta que hablan navegadores y servidores; la S añade cifrado (el candado).
TCP/IP	La fontanería de internet: mueve paquetes numerados de forma confiable entre direcciones.
Cliente / servidor	El que pregunta y el que responde. Una relación, no un tipo de máquina.
Lado del cliente (front end)	Trabajo hecho en el navegador del usuario: dibujar, respuesta instantánea.
Lado del servidor (back end)	Trabajo hecho en la máquina lejana: confianza, reglas, registros permanentes.
Proxy	Un servidor intermediario que maneja el tráfico en nombre de una de las partes.
HTML / CSS / JavaScript	Los tres lenguajes de página: estructura / apariencia / comportamiento.

Drill de fluidez: diga el viaje de seis pasos en voz alta, a solas, hasta que le tome menos de treinta segundos y nunca le falte una palabra. Después dígaselo a otra persona.

Hito: usted puede responder, en una cena, sin preparación: “¿Qué pasa realmente cuando escribo el nombre de un sitio web en mi teléfono?” — como una historia, no como una lista de jerga.

Módulo B: Cómo se construyen y alojan los sitios web — de una computadora a la nube

La gran idea: todo sitio web real se construye en **tres capas** — un front end que la gente toca, una capa intermedia donde corren las reglas de la organización, y una capa de datos que nunca olvida — y durante veinticinco años casi todo se ha armado con una de dos famosas familias de herramientas. Aprenda las tres capas y las dos familias, y las reuniones con proveedores pasarán de ser un idioma extranjero a ser un menú.

El restaurante, que usted usará durante el resto de este curso: el **front end (UI)** es el comedor — todo lo que el cliente ve, construido en HTML, CSS y JavaScript. La **capa intermedia (lógica de negocio)** es la cocina — código en el servidor que aplica las reglas de la casa: “un estudiante no puede inscribirse dos veces,” “los pedidos por encima del límite necesitan aprobación.” La **capa de datos** es la despensa y el recetario — la **base de datos**, donde se guarda todo lo que no debe olvidarse. La industria llama a esta forma **arquitectura N-tier** (de N capas). ¿Por qué capas? Para que cada una pueda renovarse sin reconstruir las demás — y cuando un ingeniero dice “el cuello de botella está en la capa de datos,” le está entregando a usted exactamente este mapa con un círculo dibujado encima.

La máquina y el programa: un **servidor** es una computadora real — una caja metálica sin pantalla, montada en un rack junto a cientos más en un **centro de datos (data center)**, construida para funcionar cada segundo del año. En ella corre un *programa* que también se llama servidor web, y que conduce la conversación HTTP. Conozca por nombre a los dos más famosos: **IIS** (de Microsoft, corre en Windows) y **Apache** (gratuito, corre principalmente en Linux). Mantenga las palabras en orden: *un navegador pregunta; un servidor web responde*. Apache es un **servidor** web — el lado que responde — no un navegador. El **web hosting** (alojamiento web) es alquilar espacio en el servidor de otro para que su máquina responda bajo el nombre de usted.

La base de datos y SQL, con calma: una **base de datos** es un programa que almacena registros y responde preguntas sobre ellos — un archivista fanáticamente organizado que nunca duerme. **SQL** (pronunciado “sequel”) es simplemente **el lenguaje para hacerle preguntas a una base de datos**. Una *query* (consulta) es una pregunta. Lea esta línea — usted ya puede: `SELECT name FROM students WHERE village = 'Mae La'` — *de la tabla de estudiantes, dame los nombres de todos los que son de Mae La*. Dos marcas dominan: **SQL Server** (de Microsoft, con licencia) y **MySQL** (gratuito, open source). Ambos hablan SQL. El punto de liderazgo: **la base de datos es donde la organización realmente vive** — las páginas pueden reconstruirse, pero pierda los registros de inscripción y no hay escuela. Por eso las bases de datos reciben la ingeniería más protectora del edificio.

Las dos familias:

Función en la máquina	Familia Microsoft	Familia open source LAMP
Sistema operativo	Windows Server	Linux
Servidor web (responde HTTP)	IIS	Apache
Lenguaje de lógica de negocio	C# en .NET	PHP
Base de datos	SQL Server	MySQL
Modelo de costos	Cuotas de licencia, un solo proveedor	Gratuito y open source; usted paga por la gente
Terreno clásico	Bancos, gobiernos, grandes empresas	La web abierta — WordPress por sí solo corre un tercio de todos los sitios web

Open source (código abierto) significa que la receta del software es pública — cualquiera puede inspeccionarla, mejorarla y usarla sin cuotas de licencia. Es la razón por la que una organización con pocos recursos siempre pudo construir un sitio web de clase mundial por el precio de una máquina alquilada.

El camino hacia la nube — la frase sobre la que se sostiene todo este libro: durante décadas todo esto corrió **on-premises**: en una sala de servidores que la organización poseía, comprada para el día más ocupado imaginable, ociosa todos los demás días, mantenida fría a costa de la organización. Entonces Amazon y Microsoft ofrecieron alquilar capacidad de sus propios almacenes por hora. Los stacks no cambiaron — los mismos Linux, Apache, IIS y SQL Server corren hoy en la nube. **Lo que cambió es dónde viven las máquinas y quién las posee.** Guarde esa frase: el Módulo 1 de la Etapa 1 abre exactamente ahí.

Vocabulario:

Término	Definición
Servidor / centro de datos	La máquina siempre encendida, y el edificio lleno de racks con ellas.
Servidor web (programa)	Software que responde peticiones HTTP: IIS (Microsoft), Apache (open source).
Web hosting	Alquilar espacio en el servidor de otro para su sitio web.
Arquitectura N-tier	Construir en capas: front end (UI), capa intermedia (lógica), capa de datos.
Base de datos	El programa que almacena registros y responde preguntas — donde vive la organización.
SQL / query	El lenguaje casi en inglés para hacerle preguntas a una base de datos; una query es una pregunta.
SQL Server / MySQL	La base de datos con licencia de Microsoft / la favorita open source. Ambas hablan SQL.
Stack de Microsoft	Windows Server + IIS + C#/.NET + SQL Server.
Stack LAMP	Linux + Apache + MySQL + PHP — todo open source.
Open source	Receta pública: libre de inspeccionar, mejorar y usar.
On-premises	Máquinas que usted posee, en una sala que usted posee — el mundo antes de la nube.

Drill de fluidez: desarme LAMP con cuatro dedos, en voz alta, tres días seguidos. Después explique el restaurante — comedor, cocina, despensa — a alguien que nunca ha oído hablar de un servidor.

Hito: un contratista le dice: “Ustedes están en un viejo montaje Microsoft on-prem — IIS y SQL Server. Nosotros somos una casa LAMP, y los moveríamos a la nube.” Usted puede traducir cada palabra para su junta directiva — y sabe que la parte riesgosa que hay que sondear es la migración de la base de datos.

ETAPA 1 — HABLE EL IDIOMA (Semanas 1-8)

Módulo 1 (Semanas 1-2): Qué es realmente la nube

La gran idea: la nube es *las computadoras de otro, alquiladas por hora, gestionadas por software*. Antes de la nube, una empresa compraba servidores físicos, los ponía en una sala y pagaba a gente para mantenerlos — lento, caro, inflexible. AWS, Microsoft Azure y Google Cloud construyeron almacenes con millones de computadoras (data centers, centros de datos) y permiten que cualquiera alquile porciones de ellas por segundo, desde cualquier lugar, a través de una página web o una línea de código. Eso es todo. Todo lo demás en este curso es detalle sobre esa única idea.

Los tres grandes proveedores, definidos:

Proveedor	Qué es
AWS (Amazon Web Services)	La división cloud de Amazon y el mayor proveedor de nube del mundo (~30% del mercado global). Comenzó en 2006, cuando Amazon empezó a alquilar los sistemas de cómputo que había construido para su propia tienda. Hoy ofrece más de 200 servicios — servidores, almacenamiento, bases de datos, IA y más — alquilados por segundo. Cuando este curso dice “la nube”, AWS es el ejemplo por defecto, y abrió su propia Región de Tailandia (Bangkok) en enero de 2025.
Microsoft Azure	La nube de Microsoft, la #2 a nivel mundial y la más fuerte dentro de las grandes empresas porque se conecta de forma natural con las herramientas de Microsoft que las compañías ya usan (Windows, Office, Active Directory). Actualmente construye su primera región de datacenters en Tailandia.
Google Cloud (GCP)	La nube de Google, la #3 — la más fuerte en analítica de datos y herramientas de IA. Comprometió \$1,000 millones para un centro de datos tailandés y una región cloud en Bangkok.

Su cuenta gratuita de AWS — configúrela en la Semana 1. Vaya a <https://aws.amazon.com/free> y haga clic en “Create a Free Account” (la página directa de registro es https://signin.aws.amazon.com/signup?request_type=register). Necesitará una dirección de correo electrónico, un número de teléfono y una tarjeta de crédito/débito para verificación de identidad — el **Free Tier** (capa gratuita) le da una asignación mensual gratuita de los servicios básicos (incluidas 750 horas/mes de un servidor EC2 pequeño durante su primer año), y los ejercicios de este curso se mantienen dentro de ella. Dos hábitos de seguridad desde el primer día: active MFA (el Módulo 6 la explica) y configure una **alerta de facturación** en \$5 (el Módulo 7 explica cómo) para que nunca pueda llevarse una sorpresa. Tras el registro, usted inicia sesión en <https://console.aws.amazon.com> — la “console” (consola) es simplemente la página web de control de AWS.

Por qué las empresas usan la nube: velocidad (un servidor nuevo en 60 segundos en lugar de 6 semanas), elasticidad (alquile 100 servidores para el gran día de rebajas y devuelva 90 al día siguiente), sin costo inicial (gasto operativo en lugar de gasto de capital) y alcance global (ponga su aplicación cerca de sus clientes en Bangkok, Tokio y Fráncfort sin construir nada).

Las tres capas de servicio — el modelo mental más útil del cloud:

Capa	Qué alquila usted	Analogía de la cocina	Ejemplo
IaaS (Infrastructure as a Service, infraestructura como servicio)	Computadoras, almacenamiento y redes en bruto — usted gestiona todo lo que corre sobre ellos	Alquilar una cocina vacía: usted trae chefs, recetas, ingredientes	Amazon EC2, Azure Virtual Machines
PaaS (Platform as a Service, plataforma como servicio)	Una plataforma gestionada — usted solo trae su aplicación	Alquilar una cocina con personal: usted trae la receta	AWS Elastic Beanstalk, Azure App Service
SaaS (Software as a Service, software como servicio)	Software terminado	Pedir en un restaurante	Gmail, Salesforce, Canva

Regiones y zonas de disponibilidad: una **Región** es un grupo geográfico de centros de datos (Tailandia tiene su propia Región de AWS desde enero de 2025, en Bangkok y sus alrededores). Una **Availability Zone (AZ, zona de disponibilidad)** es uno o más centros de datos aislados dentro de una región. Las aplicaciones que importan corren en al menos dos AZ, para que la falla de un edificio no las tumbe. Cuando los ingenieros dicen “we’re multi-AZ” (estamos en multi-AZ), quieren decir “un centro de datos puede incendiarse y seguimos en línea”.

Vocabulario:

Término	Definición
Proveedor cloud / hyperscaler	Una empresa que posee enormes centros de datos y alquila cómputo por internet (AWS, Azure, Google Cloud). “Hyperscaler” = los pocos más grandes, contruidos para una

Término	Definición
	escala casi ilimitada.
Centro de datos (data center)	Un almacén asegurado, lleno de miles de servidores con energía y refrigeración industriales — el lugar físico donde “la nube” realmente vive.
Región	Un grupo geográfico de centros de datos ofrecido como una opción de ubicación, p. ej., “Asia Pacific (Bangkok)”. Usted elige la región donde corren sus sistemas.
Availability Zone (AZ)	Uno o más centros de datos aislados dentro de una región, con energía y redes independientes. Correr en dos AZ significa que un edificio puede fallar y usted sigue en línea.
On-premises (“on-prem”)	La vieja manera: servidores que su empresa posee, en su propio edificio. Lo contrario de la nube.
Migración	El proyecto de mover sistemas desde on-prem hacia la nube.
Workload (carga de trabajo)	Cualquier aplicación o sistema que corre — “el workload de nómina”, “el workload del sitio web”. Palabra práctica: abarca cualquier cosa.
Aprovisionar (provision)	Crear/configurar un recurso cloud (un servidor, una base de datos). “Aprovisionar un servidor” = traerlo a la existencia.
Escalar vertical / horizontal (scale up / scale out)	Atender más demanda haciendo una máquina más grande (up) o añadiendo más máquinas (out). La nube favorece <i>out</i> .
Latencia	El retraso antes de que lleguen los datos, medido en milisegundos. La distancia crea latencia — la razón por la que una región en Bangkok importa a los usuarios tailandeses.
Consola (console)	El panel web de control del proveedor donde usted ve y gestiona todo lo que está alquilando.

Videos de este módulo (todos gratuitos, enlaces verificados):

Video	Canal	Duración	Enlace
What is AWS?	Amazon Web Services (oficial)	~2 min	https://www.youtube.com/watch?

Video	Canal	Duración	Enlace
Top 50+ AWS Services Explained in 10 Minutes	Fireship	~10 min	v=a9_D53WsUs https://www.youtube.com/watch?v=JlbiYCM48to
What is Microsoft Azure? An Introduction	Eye on Tech	~3 min	https://www.youtube.com/watch?v=19JkLhvaKA8
รู้จัก AWS Cloud คืออะไร (introducción en tailandés)	Aware Corporation	corto	https://www.youtube.com/watch?v=nrSpZKGxXd0

Drill de fluidez — diga estas frases hasta que le salgan con naturalidad: “Is that workload on-prem or in the cloud?” (¿Ese workload está on-prem o en la nube?) · “Which region are we in — and are we multi-AZ?” (¿En qué región estamos — y somos multi-AZ?) · “Is this an IaaS approach or is there a managed service that removes the maintenance?” (¿Es este un enfoque IaaS o hay un servicio gestionado que elimina el mantenimiento?)

Ejercicios: (1) Cree su cuenta gratuita de AWS en <https://aws.amazon.com/free> — el propio flujo de registro le enseña más vocabulario que un capítulo de lectura. (2) Vea los cuatro videos de la tabla anterior (menos de 20 minutos en total). (3) En su diario: escriba la explicación de ascensor de la nube que le daría a un director de escuela tailandés de un solo aliento.

Hito: usted puede explicar IaaS vs PaaS vs SaaS con su propia analogía, y explicar por qué la apertura de la región de AWS en Bangkok importó a los bancos tailandeses (respuesta: latencia + residencia de datos — vea el Módulo 6).

Módulo 2 (Semanas 3-4): Los componentes básicos — cómputo, almacenamiento y bases de datos

Cómputo — los motores que ejecutan código. Una **máquina virtual (VM/instancia)** es una porción de un servidor físico que se comporta como una computadora completa; usted elige un tamaño (CPU/RAM) y paga por segundo que está encendida. Un **container** (contenedor) es una forma más ligera y rápida de empaquetar una aplicación para que corra de forma idéntica en cualquier lugar; **Docker** los empaqueta y **Kubernetes** (K8s) orquesta flotas de ellos — cuando escuche “K8s”, piense “el sistema que ejecuta y repara nuestros cientos de contenedores automáticamente”. **Serverless** (sin servidor; AWS Lambda) significa que usted sube solamente una función de código; la nube la ejecuta cuando se dispara y usted paga por invocación — sin servidores que gestionar en absoluto. El patrón a notar: VM → container → serverless es un control deslizante desde “más control, más mantenimiento” hasta “menos control, mantenimiento casi cero”. Los buenos equipos eligen por workload, no por moda.

Almacenamiento — tres formas. Almacenamiento de objetos (Amazon S3): un balde sin fondo para archivos — imágenes, videos, respaldos, datasets; barato, con durabilidad de once

nueves, la respuesta por defecto a “¿dónde ponemos los archivos?” **Almacenamiento de bloques** (EBS): el disco duro virtual conectado a una VM. **Almacenamiento de archivos** (EFS): una unidad compartida que muchas máquinas montan a la vez. Luego los **tiers** (niveles): caliente (acceso frecuente, costoso) vs frío/archivo (Glacier — barato, lento) — mover los datos viejos a niveles fríos es una de las victorias de costos más fáciles que cualquier equipo puede lograr.

Bases de datos — dos familias. Relacional/SQL (MySQL, PostgreSQL; gestionadas como Amazon RDS/Aurora): datos en tablas con estructura estricta; la opción por defecto para dinero, pedidos, usuarios — todo aquello donde la corrección es sagrada. **NoSQL** (DynamoDB, MongoDB): flexible, masivamente escalable; la opción por defecto para datos enormes, rápidos y de forma más simple (sesiones, catálogos, feeds). “Base de datos gestionada” significa que el proveedor se encarga de los respaldos, los parches y el failover — los equipos deberían tener una razón fuerte para operar la suya propia.

Vocabulario:

Término	Definición
Instancia	Una máquina virtual (VM) alquilada. “Spin up an instance” = arrancar un servidor.
Tipo / tamaño de instancia	La especificación que usted eligió — cuántas CPU, cuánta memoria. Tipo más grande = precio por hora más alto.
Container (contenedor)	Un paquete ligero que contiene una aplicación más todo lo que necesita, para que corra de forma idéntica en cualquier máquina. Más rápido y barato que una VM completa.
Docker	La herramienta estándar para construir y ejecutar contenedores.
Kubernetes (K8s)	El orquestador que ejecuta y repara flotas de contenedores automáticamente — reinicia los caídos, añade más bajo carga. “K8s” es el apodo de la industria.
Cluster / nodo	Un cluster es un grupo de máquinas trabajando como un solo sistema; cada máquina en él es un nodo.
Serverless	Ejecutar código sin gestionar ningún servidor — la nube ejecuta su función cuando se dispara y factura por ejecución.
Lambda / función	El producto serverless de AWS; una “función” es la pequeña pieza de código que ejecuta.
S3 / bucket	El almacenamiento de objetos de AWS para archivos; un bucket (balde) es un contenedor de archivos con nombre. La respuesta por

Término	Definición
Volumen EBS	defecto a “¿dónde ponemos los archivos?” El disco duro virtual conectado a una instancia.
Durabilidad	La probabilidad de que los datos almacenados sobrevivan. Los “once nueves” de S3 (99.999999999%) significan que la pérdida esencialmente nunca ocurre.
Tier de almacenamiento	Clase de precio/velocidad para los datos: caliente (instantáneo, costoso) — frío/archivo (Glacier — barato, de minutos a horas para recuperar).
RDS	El servicio de base de datos relacional gestionada de AWS — AWS se encarga de respaldos, parches y failover por usted.
SQL vs NoSQL	SQL: tablas estrictas, perfecto para dinero y pedidos. NoSQL: flexible y masivamente escalable, para sesiones, catálogos, feeds.
Respaldo / snapshot	Una copia guardada de los datos (un snapshot es una copia en un punto en el tiempo de un disco o base de datos) desde la cual usted puede restaurar.
Failover	El cambio automático a una copia en espera cuando falla la primaria — la razón por la que las bases de datos gestionadas sobreviven a las malas noches.

Videos de este módulo:

Video	Canal	Duración	Enlace
Getting Started with EC2	AWS Developers (oficial)	27 min	https://www.youtube.com/watch?v=nJ-djerESW0
Introduction to Amazon S3	Amazon Web Services (oficial)	~5 min	https://www.youtube.com/watch?v=ecv-19sYL3w
Docker in 100 Seconds	Fireship	2 min	https://www.youtube.com/watch?v=Gjnup-PuquQ
Kubernetes explained in 15 mins	TechWorld with Nana	~16 min	https://www.youtube.com/

Video	Canal	Duración	Enlace
Serverless Computing in 100 Seconds	Fireship	~2 min	watch?v=VnvRFRk_51k https://www.youtube.com/watch?v=W_VV2Fx32_Y

Drill de fluidez: “Should this run on VMs, containers, or serverless — and what’s the operational cost of each choice?” (¿Debería esto correr en VMs, contenedores o serverless — y cuál es el costo operativo de cada opción?) · “Is that data hot or can it go to a cheaper tier?” (¿Esos datos son calientes o pueden ir a un tier más barato?) · “Why are we self-managing that database instead of using RDS?” (¿Por qué estamos gestionando esa base de datos nosotros mismos en lugar de usar RDS?)

Ejercicios: (1) En su cuenta gratuita: lance la instancia EC2 más pequeña y luego termínela. Suba un archivo a S3. Acaba de usar IaaS personalmente. (2) Pida a un asistente de IA que lo examine: “Dame 10 escenarios; yo responderé VM, container o serverless, y tú me calificas.”

Hito: dada cualquier aplicación simple descrita en una frase (“un sitio web donde los estudiantes piden el almuerzo”), usted puede nombrar en voz alta sus piezas de cómputo, almacenamiento y base de datos en 60 segundos.

Módulo 3 (Semanas 5–6): Redes, y cómo se hablan las piezas

La VPC — su vecindario privado. Una **Virtual Private Cloud** (nube privada virtual) es su propia sección cercada de la red del proveedor. Dentro hay **subredes (subnets)** — públicas (alcanzables desde internet, p. ej., servidores web) y privadas (inalcanzables, p. ej., bases de datos). La frase de seguridad más común que escuchará: “*la base de datos está en una subred privada.*” Si entiende por qué — los atacantes no pueden tocar lo que no tiene camino desde internet — entiende la mitad de la seguridad de redes.

La capa de tráfico: un **balanceador de carga (load balancer)** reparte el tráfico entrante entre varios servidores (y retira silenciosamente a los enfermos); el **DNS** (Route 53) traduce nombres como suempresa.com en direcciones; una **CDN** (CloudFront) almacena en caché su contenido en cientos de ciudades para que cargue rápido en todas partes; una **API** es la puerta que una pieza de software ofrece a otra — cuando los ingenieros dicen “we’ll expose an API” (expondremos una API), quieren decir “le daremos a otro software una forma controlada de usar el nuestro”; un **API gateway** es la recepción que gestiona esas puertas.

Conectando mundos: una **VPN** (túnel cifrado sobre internet) o **Direct Connect** (una línea física privada) enlaza las oficinas/sistemas on-prem de una empresa con su nube. **Nube híbrida** = operar tanto on-prem como en la nube (la mayoría de las empresas tailandesas); **multi-cloud** = usar más de un proveedor.

Vocabulario:

Término	Definición
VPC (Virtual Private Cloud)	Su propia sección privada y cercada de la red del proveedor, donde viven sus sistemas.
Subred (pública / privada)	Una subdivisión de la VPC. Las subredes públicas son alcanzables desde internet (servidores web); las privadas no (bases de datos).
Dirección IP	La dirección numérica de una máquina en una red — cómo las computadoras se encuentran entre sí.
Firewall / security group	La lista de reglas que dice qué tráfico puede llegar a una máquina (“tráfico web entra, nada más”). Un security group (grupo de seguridad) es el firewall por servidor de AWS.
Balanceador de carga	El director de tráfico que reparte las solicitudes entrantes entre varios servidores y deja de enviar a los que no están sanos.
DNS	La guía telefónica de internet — traduce suempresa.com en una dirección IP. El servicio DNS de AWS es Route 53.
CDN (Content Delivery Network)	Copias de su contenido en caché en cientos de ciudades para que cargue rápido en todas partes. La de AWS es CloudFront.
Edge location	Uno de esos puntos de CDN a nivel de ciudad — “the edge” (el borde) significa cerca de los usuarios.
API	La puerta que una pieza de software ofrece a otra. “We’ll expose an API” = le daremos a otro software una forma controlada de usar el nuestro.
API gateway	La recepción que gestiona esas puertas — autenticación, límites de tasa, registro.
VPN	Un túnel cifrado sobre la internet pública que enlaza dos redes (p. ej., su oficina con su VPC).
Direct Connect	Una línea física privada hacia la nube — más rápida y estable que la VPN, a un precio.
Híbrida / multi-cloud	Híbrida: operar on-prem y nube juntos (la mayoría de las empresas tailandesas). Multi-cloud: usar más de un proveedor.
Ingress / egress	Tráfico que entra / sale de la nube. El egress cuesta dinero — recuerde esta palabra para

Término	Definición
	el Módulo 7.

Video de este módulo: AWS Networking Basics — VPC & Subnets for Beginners — KodeKloud — https://www.youtube.com/watch?v=QM63dyA_4Pc (vea la primera mitad ahora; regrese por el resto después del Módulo 5).

Drill de fluidez: “Is the database in a private subnet?” (¿La base de datos está en una subred privada?) · “What happens when one web server dies — is the load balancer health-checking?” (¿Qué pasa cuando muere un servidor web — el balanceador de carga está haciendo health checks?) · “Are we exposing that as an API or is it internal only?” (¿Estamos exponiendo eso como una API o es solo interno?)

Ejercicios: (1) Pida a una IA que lo guíe para dibujar, en papel, la red de una app de entrega de comida: usuarios → CDN → balanceador de carga → servidores web (subred pública) → base de datos (subred privada). Dibújela tres veces hasta que pueda hacerla de memoria. (2) Encuentre el diagrama de arquitectura de su empresa o cualquier diagrama de ejemplo y encierre en un círculo cada caja que ahora puede nombrar.

Hito: usted puede esbozar ese diagrama estándar de tres capas en una pizarra y narrar el camino del clic de un cliente a través de él.

Módulo 4 (Semanas 7-8): Cómo trabajan los equipos cloud — DevOps, IaC y leer la sala

DevOps es la cultura de fusionar “las personas que escriben el software” (Dev) y “las personas que lo operan” (Ops) en un solo equipo que entrega cambios pequeños con frecuencia y seguridad, con la automatización haciendo el trabajo pesado. Su latido es el **pipeline de CI/CD** (canalización de integración y entrega continuas): cada cambio de código se construye, prueba y despliega automáticamente (Continuous Integration / Continuous Delivery). Cuando un equipo dice “it’s in the pipeline” (está en el pipeline), quiere decir que el robot lo está probando y enviando.

Infrastructure as Code (IaC, infraestructura como código) — la idea que lo cambió todo: en lugar de hacer clics por una consola para crear servidores, los ingenieros escriben *archivos de texto* que declaran la infraestructura (“dos servidores, un balanceador de carga, una base de datos, estas reglas de firewall”), y una herramienta (**Terraform**, CloudFormation) hace que la realidad coincida con el archivo. Por qué le importa a los líderes: los archivos viven en **Git** (control de versiones), así que cada cambio de infraestructura es revisado, reversible y auditable — la diferencia entre un taller y una fábrica.

Los rituales en los que se sentará: stand-up (15 minutos diarios: qué está hecho, qué sigue, qué está bloqueado — escuche los bloqueos; eliminarlos es su trabajo), **sprint** (una unidad de trabajo planificado de 1-2 semanas), **retro** (qué mejorar), **post-mortem/revisión de incidente** (después de una caída: análisis *sin culpas* de qué falló y qué lo prevendrá — la salud de un equipo se ve en si estos son honestos), **on-call** (guardia; la rotación de quién es despertado a las 3 a.m.; si el on-call es miserable, su mejor gente se irá — pregunte por él mensualmente).

Las métricas de salud que importan: uptime/disponibilidad (“tres nueves” = 99.9% $\approx$ 8.8 horas caído/año; cada nueve extra multiplica el costo), **SLA** (la promesa, con penalizaciones), **SLO** (el objetivo interno), **MTTR** (qué tan rápido se recupera — los equipos maduros optimizan la recuperación, no la fantasía de cero fallas), **frecuencia de despliegue** (los equipos sanos entregan cambios pequeños a menudo; el miedo a desplegar es una mala señal).

Vocabulario:

Término	Definición
DevOps	La cultura de un solo equipo que construye y opera su software, entregando cambios pequeños con frecuencia mediante automatización.
Pipeline de CI/CD	La cinta transportadora automatizada que construye, prueba y despliega cada cambio de código (Continuous Integration / Continuous Delivery).
Deploy / rollback	Deploy (desplegar): liberar un cambio al sistema en vivo. Rollback: deshacerlo rápido cuando se porta mal.
Git / repositorio / pull request	Git: el sistema de control de versiones que registra cada cambio. Repositorio (“repo”): el hogar del código de un proyecto. Pull request (PR): un cambio propuesto que otro ingeniero revisa antes de fusionarse.
IaC (Infrastructure as Code)	Declarar la infraestructura en archivos de texto que una herramienta convierte en realidad — revisada, reversible, auditable.
Terraform	La herramienta de IaC más popular (funciona en todas las nubes). La propia de AWS es CloudFormation.
Staging vs producción (“prod”)	Staging: la copia de ensayo del sistema. Prod: la real que tocan los clientes. “Broke prod” (rompió prod) = el mal día.
Stand-up	Sincronización diaria de 15 minutos: hecho / siguiente / bloqueado. Escuche los bloqueos — eliminarlos es su trabajo.
Sprint / backlog	Sprint: una unidad de trabajo planificado de 1–2 semanas. Backlog: la lista de pendientes ordenada que alimenta los sprints.
Retro	Reunión de fin de sprint sobre cómo trabajar mejor la próxima vez.
Post-mortem	La revisión sin culpas después de una caída: qué falló, por qué, qué previene una

Término	Definición
On-call	repetición. La rotación de quién responde la alarma de las 3 a.m. Si el on-call es miserable, su mejor gente se va.
Incidente / sev-1	Una interrupción no planificada; “sev-1” (severidad uno) = la peor clase, todos a bordo.
SLA / SLO	SLA: la promesa al cliente con penalizaciones (p. ej., 99.9% de uptime). SLO: el objetivo interno más estricto que protege el SLA.
MTTR	Mean Time To Recovery (tiempo medio de recuperación) — qué tan rápido vuelve a estar en línea tras una falla. Los equipos maduros optimizan esto, no la fantasía de nunca fallar.
Monitoreo / observabilidad	Vigilar la salud en vivo mediante logs (registros de eventos), métricas (números en el tiempo) y alertas (avisos automáticos cuando se rompen umbrales).

Videos de este módulo:

Video	Canal	Duración	Enlace
What is DevOps? REALLY understand it	TechWorld with Nana	~15 min	https://www.youtube.com/watch?v=OyWAtQ6wYNM
DevOps CI/CD Explained in 100 Seconds	Fireship	2 min	https://www.youtube.com/watch?v=scEDHsr3APg
Terraform explained in 15 mins	TechWorld with Nana	18 min	https://www.youtube.com/watch?v=l5k1ai_GBDE

Drill de fluidez: “Is that change through the pipeline or was it manual?” (¿Ese cambio pasó por el pipeline o fue manual?) · “What did the post-mortem conclude, and what’s the prevention item?” (¿Qué concluyó el post-mortem, y cuál es la acción de prevención?) · “What’s our MTTR trending like?” (¿Cómo va la tendencia de nuestro MTTR?) · “Is this in Terraform, or did someone click it into existence?” (¿Esto está en Terraform, o alguien lo creó a punta de clics?) (esto último se llama “ClickOps”, dicho con el ceño fruncido).

Ejercicios: (1) Vea la discusión de un informe post-mortem real de un incidente (muchos son públicos — los informes de caídas de Cloudflare y AWS son famosos) y resúmalos en su diario en cinco frases. (2) Asista (o vea la grabación de) cualquier stand-up y anote los tres bloqueos que escuchó.

Hito — fin de la Etapa 1: usted puede sentarse en una reunión técnica de planificación de 30 minutos y seguir ≥80% de ella, y su diario lo demuestra: notas de una reunión real o simulada con cada acrónimo correctamente expandido. **Este es también el momento de agendar el examen AWS Cloud Practitioner (CLF-C02)** — 2–4 semanas de preparación enfocada sobre estos módulos es la norma publicada, y aprobarlo es su primera prueba externa.

ETAPA 2 — PIENSE COMO LA SALA (Semanas 9–16)

Módulo 5 (Semanas 9–10): Arquitectura — juzgar diseños sin dibujarlos

Su papel en una revisión de diseño no es diseñar — es interrogar. El marco que usa toda la industria es el **Well-Architected Framework** de AWS, seis pilares ante los cuales todo diseño debe responder: excelencia operativa, **seguridad**, **confiabilidad**, eficiencia de rendimiento, **optimización de costos**, sostenibilidad. Aprenda a fondo los tres en negrita; ahí es donde viven el dinero y el riesgo.

Confiabilidad, en palabras simples: todo falla tarde o temprano, así que los buenos sistemas lo asumen. **Redundancia** (ningún punto único de falla — dos de cada cosa importante), **multi-AZ** (sobrevivir la pérdida de un centro de datos), **auto-scaling** (escalado automático; máquinas añadidas/retiradas automáticamente según la demanda), **respaldos que se prueban** (un respaldo no probado es una esperanza, no un plan), **RTO/RPO** (Recovery Time Objective: cuánto tiempo podemos estar caídos; Recovery Point Objective: cuántos datos podemos perder — esos dos números *son* la conversación de recuperación ante desastres, y son decisiones de negocio, es decir, **suyas**).

El triángulo de los trade-offs (compromisos): rápido, barato, resiliente — elija dos. Cada discusión de arquitectura que usted arbitrará se reduce a en qué punto de ese triángulo necesita sentarse el negocio para *este* workload. Un sistema de pagos y un sitio de marketing no merecen la misma respuesta.

Las siete preguntas del líder — memorícelas; lo vuelven peligroso en cualquier revisión de diseño:

1. “What happens when this component fails?” (¿Qué pasa cuando falla este componente?) (siempre hay un “cuando”)
2. “What are the RTO and RPO, and who signed off on them?” (¿Cuáles son el RTO y el RPO, y quién los aprobó?)
3. “Where does this run — single AZ, multi-AZ, multi-region — and why?” (¿Dónde corre esto — una sola AZ, multi-AZ, multi-región — y por qué?)

4. “What will this cost per month at 10× today’s load?” (¿Cuánto costará esto al mes con 10× la carga de hoy?)
5. “What’s the simplest version that meets the requirement?” (¿Cuál es la versión más simple que cumple el requisito?) (la sobreingeniería es la enfermedad del junior; la simplicidad es la virtud del senior)
6. “What are we locked into, and what would leaving cost?” (¿A qué estamos atados, y cuánto costaría salir?) (el vendor lock-in es un precio, a veces vale la pena pagarlo — a sabiendas)
7. “Who else has built this before — are we inventing or assembling?” (¿Quién más ha construido esto antes — estamos inventando o ensamblando?) (prefiera los patrones aburridos y probados)

Drill de fluidez: practique entregar las preguntas 1, 4 y 5 en un tono cálido — aterrizan como sabiduría o como ataque dependiendo enteramente de la entrega. “Help me understand what happens if the cache goes down” (Ayúdeme a entender qué pasa si se cae la caché) vence a “did you think about failure?” (¿pensó en las fallas?).

Ejercicios: (1) Tome tres arquitecturas de ejemplo (pida a una IA que genere: un sitio de e-commerce, un backend de app móvil, una plataforma de analítica de datos) y aplique las siete preguntas a cada una, escribiendo las respuestas que esperaría. (2) Lea el resumen de una página de los pilares Well-Architected en <https://aws.amazon.com/architecture/well-architected/>.

Videos de este módulo:

Video	Canal	Duración	Enlace
The Five Pillars of the AWS Well-Architected Framework	Amazon Web Services (oficial; un sexto pilar, Sustainability, fue añadido después)	~4 min	https://www.youtube.com/watch?v=KvEDbPmha6o
What is the AWS Well-Architected Framework?	Tech With Lucy (ex-AWS)	~10 min	https://www.youtube.com/watch?v=MpDJ6TCWKjk

Hito: en una revisión de diseño simulada (hágala con una IA en el papel del ingeniero), usted hace cinco preguntas sustanciales y resume correctamente el punto más débil del diseño al final.

Módulo 6 (Semanas 11–12): Seguridad y cumplimiento — la conversación que puede acabar con empresas

El modelo de responsabilidad compartida — lo primero que hay que entender: el proveedor asegura la nube en sí (edificios, hardware, hipervisores); **usted** asegura lo que pone *dentro* de ella (sus datos, reglas de acceso, configuraciones). La mayoría de las brechas son errores de configuración del cliente — un bucket de S3 dejado público, una clave de acceso

filtrada — no fallas del proveedor. Así que “AWS is secure” (AWS es seguro) y “we are secure on AWS” (nosotros somos seguros en AWS) son frases distintas.

El vocabulario de la defensa: **IAM** (Identity and Access Management, gestión de identidades y accesos — quién puede hacer qué; lo más auditado del cloud), **mínimo privilegio (least privilege)** (todos reciben el acceso mínimo necesario — la regla de oro), **MFA** (autenticación multifactor; segundo factor en cada inicio de sesión humano — no negociable), **cifrado en reposo y en tránsito** (datos codificados en disco y en la red — lo mínimo esperable, siempre activo), **cuenta root** (la llave maestra — guardada bajo llave, nunca usada a diario), **gestión de secretos** (contraseñas/claves en una bóveda, nunca en el código), **zero trust** (confianza cero; verificar todo, no confiar en ninguna ubicación de red por defecto), **penetration test** (prueba de penetración; atacantes contratados que ponen a prueba sus defensas), **ransomware** (la razón por la que los respaldos probados y fuera de línea son un control de seguridad, no solo de operaciones).

PDPA — la ley de datos de Tailandia, y su ventaja de mercado. La Personal Data Protection Act es el GDPR de Tailandia: consentimiento requerido para recolectar datos personales, un deber de **notificación de brechas en 72 horas**, Data Protection Officers para procesadores a gran escala, y reglas sobre el envío de datos a través de fronteras. La aplicación se volvió real en 2025 — más de THB 21.5M en primeras multas, incluido un hospital multado por supervisar mal a un *proveedor*. Dos consecuencias para usted: (1) toda empresa tailandesa tiene ahora un problema de cumplimiento que a su empresa le pueden pagar por gestionar; (2) la **residencia de datos** — mantener los datos tailandeses en suelo tailandés — es un impulsor real que empuja workloads hacia las regiones de Bangkok. Este párrafo es la mitad de su discurso de ventas en Tailandia; apréndaselo al dedillo.

Las preguntas de seguridad del líder: “Who has access to production, and when did we last review the list?” (¿Quién tiene acceso a producción, y cuándo revisamos la lista por última vez?) · “Are we alerted on unusual access, or would we find out from the news?” (¿Recibimos alertas por accesos inusuales, o nos enteraríamos por las noticias?) · “When did we last *restore* a backup?” (¿Cuándo fue la última vez que *restauramos* un respaldo?) (no “¿tenemos respaldos?”) · “If we lost this dataset, is it a PDPA-notifiable breach — and could we notify within 72 hours?” (Si perdiéramos este dataset, ¿es una brecha notificable bajo PDPA — y podríamos notificar en 72 horas?) · “What did the last pen test find, and what’s still open?” (¿Qué encontró el último pen test, y qué sigue abierto?)

Vocabulario:

Término	Definición
IAM (Identity and Access Management)	El sistema que controla quién (personas y software) puede hacer qué en su nube — lo más auditado que hay en ella.
Rol / política	Una política (policy) es un conjunto de permisos por escrito; un rol es un paquete de políticas que una persona o programa asume, como quien se pone un uniforme.
Mínimo privilegio	La regla de oro: cada quien recibe el acceso

Término	Definición
MFA (autenticación multifactor)	mínimo que su trabajo necesita, nada más. Una segunda prueba (código en el teléfono, llave física) además de la contraseña. No negociable en todo inicio de sesión humano.
Cifrado en reposo / en tránsito	Datos codificados mientras se almacenan / mientras viajan por la red. Ambos siempre activos; lo mínimo esperable.
KMS (gestión de llaves)	El servicio que guarda y rota las llaves de cifrado.
Secretos	Contraseñas, claves de API, tokens — guardados en una bóveda, nunca escritos en el código.
Cuenta root	La llave maestra de toda la cuenta cloud. Guardada bajo llave con MFA, nunca usada para el trabajo diario.
Zero trust	Postura de seguridad que verifica cada solicitud y no confía en ninguna ubicación de red por defecto.
Vulnerabilidad / parcheo	Una debilidad conocida en el software / aplicar la corrección. Los sistemas sin parches son el origen de la mayoría de las intrusiones.
Pen test (prueba de penetración)	Atacantes éticos contratados que demuestran dónde fallan sus defensas antes de que lo hagan los reales.
Ransomware	Ataque que cifra sus datos a cambio de un rescate — la razón por la que los respaldos probados y fuera de línea son un control de <i>seguridad</i> .
SOC 2 / ISO 27001	Los sellos independientes de auditoría de seguridad que los clientes empresariales exigen antes de firmar.
PDPA	La Personal Data Protection Act de Tailandia — consentimiento, notificación de brechas en 72 horas, reglas de transferencia transfronteriza. Aplicada con multas reales desde 2025.
Residencia de datos	Mantener los datos físicamente dentro de las fronteras de un país — una razón clave por la que los workloads tailandeses se mueven a las regiones de Bangkok.
DPO / notificación de brechas	Data Protection Officer (requerido para

Término	Definición
	procesadores a gran escala) / el deber de reportar una brecha de datos personales en 72 horas.

Video de este módulo: The AWS Shared Responsibility Model — Digital Cloud Training — 4 min — <https://www.youtube.com/watch?v=ESPBBEK-cvo>

Drill de fluidez: “Is that bucket public? Why?” (¿Ese bucket es público? ¿Por qué?) · “Least privilege — does the intern really need prod access?” (Mínimo privilegio — ¿el practicante de verdad necesita acceso a prod?) · “Where does the PDPA line sit in this design — what’s personal data here and where does it physically live?” (¿Dónde queda la línea de la PDPA en este diseño — qué son datos personales aquí y dónde viven físicamente?)

Ejercicios: (1) Lea una historia famosa de brecha por mala configuración cloud (Capital One 2019 es el clásico) y escriba la versión de tres frases para su diario. (2) Haga que una IA simule a un cliente que pregunta “¿por qué debería confiarle mis datos?” y practique la respuesta usando el lenguaje de responsabilidad compartida + PDPA.

Hito: usted puede explicar el modelo de responsabilidad compartida y el significado práctico de la PDPA a un dueño de negocio tailandés no técnico en menos de tres minutos — porque esa explicación es la reunión de ventas del MSP.

Módulo 7 (Semanas 13–14): El dinero — economía del cloud y FinOps

Aquí es donde un líder no ingeniero aporta valor más rápido, porque a la mayoría de los ingenieros nunca se lo enseñaron y el desperdicio es enorme — las encuestas encuentran consistentemente que aproximadamente un tercio del gasto en cloud se desperdicia.

Cómo corre el medidor: el cómputo se factura por segundo que está *encendido* (no por uso — un servidor ocioso factura completo; “lo dejamos corriendo” es el desperdicio clásico), el almacenamiento por GB-mes, y — la trampa famosa — el **egress** (salida de datos): los datos que fluyen *hacia afuera* de la nube cuestan dinero mientras que los datos que entran son gratis. Las grandes facturas de egress sorprenden a todos una vez; después de este módulo, a usted no.

El menú de precios: **on-demand** (bajo demanda; precio completo, flexibilidad completa) · **instancias reservadas / savings plans** (comprométase 1–3 años por 30–70% de descuento — la palanca más grande sobre un workload estable) · **instancias spot** (hasta 90% de descuento para trabajo interrumpible como los jobs por lotes) · **rightsizing** (ajuste de tamaño; la mayoría de los servidores están sobredimensionados; encogerlos es dinero gratis) · niveles de almacenamiento (Módulo 2) · **auto-scaling como herramienta de costos** (¿por qué pagar la capacidad de medianoche a precios de mediodía?).

FinOps es la práctica de hacer el gasto cloud visible, asignado y optimizado continuamente: **etiquetado (tagging)** de cada recurso con su dueño y proyecto (el gasto sin etiquetar es gasto sin responsable), **showback/chargeback** (mostrar a cada equipo su factura — el comportamiento cambia al instante), presupuestos con **alertas** (nunca descubra el sobregasto por la factura), y **unit economics** (economía unitaria) — la métrica del líder: no “nuestra factura es \$800k/mes” sino “nuestro costo *por transacción de cliente* está bajando”. La

credencial FinOps Certified Practitioner toma unas dos semanas y es, para un líder del lado del negocio, el certificado con mayor credibilidad por hora de todo el mundo cloud. Obténgala.

Las preguntas de dinero del líder: “What’s our cost per [customer/transaction/tenant], and which direction is it moving?” (¿Cuál es nuestro costo por [cliente/transacción/tenant], y en qué dirección se mueve?) · “What percentage of our steady workload is on reservations?” (¿Qué porcentaje de nuestro workload estable está en reservas?) · “What’s untagged?” (¿Qué está sin etiquetar?) · “What died but is still billing?” (¿Qué murió pero sigue facturando?) (discos huérfanos e IPs ociosas — toda cuenta los tiene) · “What would this bill look like at 10× growth — does our architecture get cheaper or more expensive per unit?” (¿Cómo se vería esta factura con un crecimiento de 10× — nuestra arquitectura se vuelve más barata o más cara por unidad?)

Vocabulario:

Término	Definición
On-demand	Precio de pago por uso: precio completo, cancele cuando quiera. El predeterminado, y la forma más cara de correr workloads estables.
Reservada / savings plan	Un compromiso de 1–3 años de uso estable a cambio de 30–70% de descuento — la palanca de costos más grande.
Spot	Capacidad sobrante con hasta 90% de descuento que el proveedor puede reclamar con minutos de aviso — perfecta para trabajo por lotes interrumpible.
Rightsizing	Encoger servidores sobredimensionados a lo que realmente usan. Dinero gratis en casi toda cuenta.
Egress	Datos que salen de la nube — facturados por GB mientras que los datos entrantes son gratis. La sorpresa famosa en las facturas.
Tagging (etiquetado)	Etiquetar cada recurso con dueño/proyecto/entorno para que cada baht de gasto sea atribuible. Gasto sin etiquetar = gasto sin responsable.
Showback / chargeback	Mostrar a cada equipo su propia factura cloud (showback) o cobrársela internamente de verdad (chargeback). El comportamiento cambia al instante.
Alerta de presupuesto	Un aviso automático al cruzar un umbral de gasto — para que nunca se entere del sobregasto por la factura.
Unit economics	Costo por unidad de negocio — por cliente,

Término	Definición
	por transacción — la métrica del líder, más significativa que la factura total.
TCO (costo total de propiedad)	El costo completo de una decisión durante su vida: licencias, personas, energía, migración — no solo el precio de etiqueta.
FinOps	La disciplina (y cultura de equipo) de hacer el gasto cloud visible, asignado y continuamente optimizado.

Video de este módulo: What is FinOps? — FinOps Foundation (oficial) — 2 min — https://www.youtube.com/watch?v=Y-c_xw9bHFW · luego el curso gratuito “Introduction to FinOps” en <https://learn.finops.org>.

Ejercicios: (1) Abra la calculadora de precios de AWS y cotice un sistema real pequeño (dos servidores, una base de datos, 500GB de almacenamiento, 1TB de egress) — el ejercicio de hacerlo una vez desmitifica toda conversación futura de costos. (2) Pida a una IA que interprete a un ingeniero defendiendo un servidor sobredimensionado; practique negociar el rightsizing con amabilidad.

Hito: tome (o agende) el examen FinOps Practitioner, y en una revisión simulada, encuentre cuatro problemas de costos en una factura de ejemplo que una IA genere para usted.

Módulo 8 (Semanas 15–16): Bootcamp de fluidez — juntando todo el idioma

Esta quincena es pura integración — la diferencia entre conocer las palabras y *ser fluido en las reuniones*.

Drill diario (30 min): haga que una IA genere una transcripción realista de reunión (revisión de diseño, retro de incidente o revisión de costos), con dos errores técnicos deliberados escondidos en ella. Su trabajo: resumir la reunión en cinco frases, atrapar los errores y escribir las tres preguntas que habría hecho. Alterne los tipos de reunión a diario.

El músculo de la traducción (15 min): tome una afirmación técnica por día y tradúzcala para tres audiencias — el CFO (dinero), un cliente (riesgo/beneficio) y un ingeniero junior nuevo (enseñanza). Ejemplo: “We’re moving the session store from the database to Redis” (Estamos moviendo el almacén de sesiones de la base de datos a Redis) — CFO: “reduce la carga de la base de datos y así aplazamos una ampliación de \$2M” — cliente: “las páginas cargan más rápido en las horas pico” — junior: “Redis mantiene los datos calientes en memoria, así dejamos de martillar la base de datos con cada clic.”

Práctica de lectura (15 min): un artículo real de blog de ingeniería por día (el AWS Architecture Blog, o los blogs de ingeniería de Netflix/Grab — Grab es especialmente relevante: escala del sudeste asiático, mercado vecino al tailandés). Ahora entenderá el 70–80% de ellos. Busque el resto.

Su compañero de preparación de examen para esta etapa: el famoso curso completo gratuito — *AWS Certified Cloud Practitioner (CLF-C02) 2026* de Andrew Brown en freeCodeCamp

— <https://www.youtube.com/watch?v=7HKot-brXFE> (la edición anterior de 14 horas, también válida para el mismo código de examen, está en <https://www.youtube.com/watch?v=NhDYbskXRgc>). Véalo a 1.25× a lo largo de las Semanas 13–16; después de los Módulos 1–7 la mayor parte le parecerá repaso, que es exactamente la señal de que está listo.

Hito — fin de la Etapa 2, su examen de graduación: (1) Apruebe el **AWS Cloud Practitioner** si aún no lo ha hecho. (2) El desafío simulado: una IA interpreta a un ingeniero senior que le presenta una arquitectura defectuosa para un cliente tailandés de e-commerce; usted debe encontrar la base de datos en una sola AZ, el bucket de S3 público, la estimación de egress faltante y la consideración de PDPA ausente — y entregar la retroalimentación en un tono que haga que el “ingeniero” se sienta ayudado, no atrapado. Cuando pueda hacer eso, usted es conversacionalmente peligroso, a dieciséis semanas de haber empezado.

ETAPA 3 — LIDERE (Meses 5–24)

Módulo 9 (Meses 5–8): La certificación profunda y el piso técnico

Trabaje el material de **AWS Solutions Architect Associate (SAA-C03)** — 2–3 meses a su ritmo de una hora. Puede o no presentar el examen (como líder, el *material* es el valor; la insignia es teatro opcional), pero aquí es donde las regiones, las VPC, IAM y los precios dejan de ser vocabulario y se convierten en un sistema conectado en su cabeza. En paralelo, mantenga los drills diarios del Módulo 8 a media dosis. Esta es también la ventana para añadir **fundamentos de Azure (nivel AZ-900)** — Tailandia es un mercado empresarial muy orientado a Microsoft, y la alfabetización bilingüe (AWS+Azure) amplía sus conversaciones con clientes.

Módulo 10 (Meses 5–12, en curso continuo): Contratar y gestionar ingenieros

Contratar cuando no puede juzgar del todo la habilidad: la estructura vence al instinto. Use un circuito consistente: una conversación de filtrado que usted lidera (motivación, comunicación, cómo explican un proyecto pasado a un no ingeniero — si no pueden, también fallarán con sus clientes) + una entrevista técnica conducida por su **bar-raiser** (su #2 técnico o un contratista senior pagado) + una llamada de referencias donde usted hace exactamente la única pregunta que importa: “would you hire this person again for this role?” (¿volvería a contratar a esta persona para este puesto?). Vigile los dos arquetipos de fracaso: el conversador fluido sin profundidad (su bar-raiser los atrapa) y el experto profundo pero callado (a menudo oro en los equipos tailandeses, donde la modestia es cultural — no deje que el brillo en la entrevista pese más que la evidencia de trabajo real).

El #2 técnico — la decisión más importante de la aventura: contrátele primero, páguele con equity significativo (10–20% si es un verdadero cofundador), y defina el trato explícitamente: él sostiene el estándar técnico y es dueño de las decisiones de arquitectura; usted es dueño de los clientes, el dinero, las prioridades y las personas; los desacuerdos entre ustedes ocurren en privado y se resuelven antes de que el equipo los vea.

Los rituales que hacen que los ingenieros se queden: 1:1s semanales o quincenales que son sobre *ellos* (carrera, fricción, energía — no informes de estado); un plan de crecimiento por escrito por persona (en el mercado tailandés con déficit de 70k talentos/año, el crecimiento y los presupuestos de certificación retienen mejor que el salario solo — pague cada certificación, con un bono de permanencia de 12 meses al completarla); crédito en público, corrección en privado; y protección implacable de su tiempo de concentración — un líder que cancela una reunión a mitad de sprint es un héroe.

Foros de decisión — cómo tomar decisiones técnicas que no puede evaluar del todo: para toda decisión grande, exija un **documento de decisión** de una página al ingeniero que propone: el problema, 2–3 opciones, costos, riesgos, recomendación. Luego dirija la reunión con sus siete preguntas del Módulo 5. Usted no es el arquitecto más inteligente de la sala y nunca necesita serlo — usted es la persona que hace que gane la opción *mejor argumentada*, a tiempo, con los trade-offs de dinero y riesgo hechos explícitos. Los ingenieros respetan esto profundamente cuando se hace con honestidad; anote quién predijo qué (su Diario de Decisiones otra vez) y revise las predicciones trimestralmente — eso los calibra a usted y a ellos.

Módulo 11 (Meses 6–24): La curva de credibilidad, con honestidad

La experiencia publicada en la literatura de liderazgo de ingeniería converge en esta línea de tiempo, y fingir lo contrario es la forma en que fracasan los fundadores no técnicos: **~90 días** para dirigir el ritmo del negocio con competencia · **~6 meses** para una efectividad básica (reuniones fluidas, decisiones estructuradas, equipo estable) · **12–18 meses** antes de que sus instintos técnicos valgan mucho por sí solos · **~2 años** antes de sostenerse genuinamente frente a ingenieros senior en trade-offs de arquitectura. Las mitigaciones mientras la curva sube: pida credibilidad prestada (su #2 presenta la mitad técnica de las reuniones de ventas — los clientes necesitan ver la banca de todos modos), nunca fanfarronee (el asesino de credibilidad más rápido; “I don’t know — walk me through it” (No lo sé — explíquemelo) es una frase de liderazgo), y deje que sus preguntas hablen por usted: un líder que pregunta “what’s our RPO and who signed off on it?” (¿cuál es nuestro RPO y quién lo aprobó?) suena a treinta años de cicatrices, y después de este curso, usted lo dirá en serio.

Módulo 12 (En curso): La capa del mercado tailandés

Incorpore los detalles de la Parte 5 del TSI como conocimiento operativo: la **PDPA** como deber de cumplimiento y como producto (\$Módulo 6); las **escaleras de partners** (AWS Select necesita un puñado de personal certificado + 3 acuerdos lanzados; Microsoft Solutions Partner necesita un puntaje de capacidad de 70/100 — las certificaciones de su equipo son literalmente activos de ventas, otra razón para financiarlas); la **promoción BOI** para la propiedad extranjera del 100% de la empresa; las bandas salariales de Bangkok (junior ฿50–75k — arquitecto ฿180–280k/mes) para cotizar licitaciones y ofertas correctamente; y la matemática de ventas del momento — más de \$27B en inversión aprobada de centros de datos, un mandato gubernamental cloud-first, y una brecha de habilidades de 70,000/año que su banca certificada existe para llenar.

El plan de estudios en una página

Cuándo	Enfoque	Prueba externa
Semanas 1–2	Qué es el cloud; IaaS/PaaS/SaaS; regiones/AZs	—
Semanas 3–4	Cómputo, almacenamiento, bases de datos	—
Semanas 5–6	Redes; leer diagramas de arquitectura	—
Semanas 7–8	DevOps, IaC, rituales de equipo, métricas de ops	Agendar Cloud Practitioner
Semanas 9–10	Juicio de arquitectura; las siete preguntas	—
Semanas 11–12	Seguridad; PDPA; responsabilidad compartida	Examen AWS Cloud Practitioner
Semanas 13–14	Economía del cloud; FinOps	FinOps Practitioner (≈2 semanas de preparación)
Semanas 15–16	Bootcamp de fluidez; desafío simulado	Graduación: el desafío
Meses 5–8	Material SAA-C03; Azure AZ- 900	Examen SAA (opcional)
Meses 5–12	Contratación; #2 técnico; foros de decisión	Primeras contrataciones bien hechas
Meses 6–24	Curva de credibilidad; capa del mercado tailandés	Nivel de partner; primeros retainers

Unas palabras de cierre de su profesor. Dentro de dieciséis semanas usted seguirá cada conversación en la sala. Esa no es la línea de meta — es la licencia para empezar. La curva de dos años hasta el verdadero juicio técnico no es un muro; es un foso: cada semana que usted completa de ella es una semana que los fundadores no técnicos de sus competidores no completaron. Estudie a diario, registre cada decisión en su diario, nunca fanfarronee, y contrate a personas mejores que usted y haga que se alegren de haber venido. Ese es todo el trabajo.

Complemento de “The Thailand Strategic Investment (TSI)”, Parte 5. Las estimaciones de tiempo de preparación de certificaciones y las líneas de tiempo de liderazgo provienen de las fuentes allí citadas (CBT Nuggets, StudyTech, FinOps Foundation, First Round Review, The Pragmatic Engineer).

LECCIONES NUEVAS — EDICIÓN DE AGOSTO DE 2026

En agosto de 2026 se añadieron lecciones nuevas a los tres cursos interactivos de b4lcilc.org — los cursos de Líder Cloud, Ingeniero Cloud y Arquitecto Cloud. Este apéndice pone al día la edición de lectura con condensaciones fieles de ese material nuevo, en la misma voz que el resto de este libro.

Unas palabras sobre lo que usted tiene en las manos. Estas páginas son la edición de lectura: las ideas, las metáforas y las frases que vale la pena decir en voz alta. Las versiones interactivas completas en línea traen bastante más — cada lección nueva incluye quizzes que comprueban su comprensión antes de dejarlo avanzar, laboratorios prácticos (incluida una terminal de práctica que se comporta como la real pero no puede costar dinero), videos, drills de vocabulario y — para el material del capstone — ejercicios de diseño guiados con listas de requisitos y respuestas modelo resueltas. Lea estas condensaciones para meterse en la cabeza la forma de cada idea; después, cuando pueda, haga las versiones interactivas para que el conocimiento se quede.

*Las adiciones caen en tres grupos. Para el curso de **Líder Cloud**: una introducción amable a la terminal, la ley de privacidad propia de Tailandia, una guía del líder para propuestas de IA y facturas de IA, y un ejercicio capstone que ensambla el curso entero en una sola página. Para el curso de **Ingeniero Cloud**: la computación serverless — el patrón que convierte los pequeños scripts operativos en funciones que no cuestan nada ejecutar. Para el curso de **Arquitecto Cloud**: la aritmética de las promesas de disponibilidad, el patrón de Generación Aumentada por Recuperación (Retrieval-Augmented Generation) por el que los clientes ahora preguntan cada semana, y un ejercicio completo de costeo — ponerle precio a un diseño antes de que alguien lo construya.*

Para el curso de Líder Cloud

La ventana negra — cómo hablarle a una computadora directamente

Antes de su primer laboratorio, conozca la herramienta donde se ejecuta. Una **terminal** (también llamada *línea de comandos* o *consola*) es un programa que parece una simple ventana negra con texto parpadeante. Es la manera más antigua de usar una computadora, y sigue siendo como los ingenieros hacen el trabajo serio — porque una instrucción escrita es precisa, repetible y se puede anotar y compartir, de una manera que hacer clic por los menús nunca podrá ser.

La terminal muestra un **prompt** — normalmente una línea corta que termina en \$ — que significa exactamente una cosa: *estoy escuchando*. Un **comando** es una oración con gramática: la primera palabra es la *herramienta* a la que usted llama (como `git` o `aws`), las palabras siguientes dicen *qué hacer*, y las piezas que empiezan con un guion, llamadas **flags** (como `- -output table`), ajustan *cómo* hacerlo.

Dos costumbres sorprenden a todo recién llegado. Primera: **el silencio suele significar éxito**. Muchos comandos, cuando funcionan a la perfección, no imprimen absolutamente nada — la máquina considera que hacer el trabajo es respuesta suficiente. Segunda: la terminal no

adivina. Donde un buscador perdona la ortografía, la terminal no perdona nada — una sola letra equivocada y le dice educadamente que el comando no existe. Esa severidad se siente dura durante un día y luego se vuelve un consuelo: la máquina nunca hace *más o menos* lo que usted pidió.

Una pieza más de gramática: las computadoras organizan todo en **archivos** que viven dentro de **carpetas** (también llamadas *directorios*) — así que `mkdir decisions` significa “*make directory* (crea un directorio) llamado decisions”. Los laboratorios del curso corren en un **simulador**: una terminal de práctica que no puede crear, borrar ni costar nada — pero los comandos son los reales, letra por letra. Domínelos aquí, y el día que se siente frente a una terminal real, sus manos ya sabrán el camino.

La PDPA de Tailandia — la ley de privacidad más cercana a casa

El resto del módulo de cumplimiento es global; esta es la ley cuyo regulador de verdad puede tocar a la puerta. La PDPA de Tailandia — la Personal Data Protection Act (Ley de Protección de Datos Personales), B.E. 2562 (2019) — entró plenamente en vigor en 2022 y aplica a cualquier organización en Tailandia que maneje datos personales. Una organización benéfica de Mae Sot que guarda expedientes de estudiantes está de lleno dentro de ella.

La buena noticia es genuinamente buena: la PDPA es casi una copia del GDPR — el Reglamento General de Protección de Datos europeo. Los redactores de Tailandia tomaron el reglamento europeo como plantilla, así que el GDPR ya le ha enseñado aproximadamente el noventa por ciento de esta ley. Los datos personales pertenecen a la persona que describen; usted necesita una base legal — casi siempre el **consentimiento**, dado libremente y revocable — para recolectarlos; y la **limitación de propósito** prohíbe usar datos recogidos para un fin en otro distinto: los expedientes recolectados para inscribir a un estudiante no pueden convertirse calladamente en una lista de correo para recaudar fondos.

El regulador es el PDPC — el Personal Data Protection Committee (Comité de Protección de Datos Personales) — cuyas sanciones alcanzan la responsabilidad penal, no solo multas. El reloj de las brechas también le resultará familiar: una brecha que probablemente suponga un riesgo para las personas debe reportarse al PDPC **dentro de 72 horas**.

El punto de liderazgo merece decirse sin rodeos. Una organización que guarda expedientes de estudiantes — nombres, familias y, en esta región, a veces el estatus de documentación, datos tan sensibles como los que más — **responde primero ante esta ley**: la respuesta debe satisfacer a Bangkok antes de preocuparse por Bruselas. Las preguntas para hacerle a su equipo: ¿qué datos personales tenemos, y declaramos el propósito cuando los recolectamos? ¿Podríamos borrar por completo los expedientes de un estudiante si nos lo pidieran? Si nos vulneraran esta noche, ¿podríamos decirle al PDPC dentro de 72 horas qué se llevaron y de quién? Ninguna requiere un abogado para hacerla. Todas requieren un líder que insista en hacerlas.

Liderar en plena ola de la IA

Tarde o temprano — y últimamente es temprano — llegará a su escritorio un diseño con IA dentro. Su trabajo no cambia: usted interroga, no diseña.

El motor detrás de la mayoría de las funciones de IA es un **LLM — un Large Language Model (modelo de lenguaje grande)** — y el hecho comercial crucial: usted casi nunca es dueño de uno; alquila el acceso a través de una **API de IA** (Application Programming Interface, interfaz de programación de aplicaciones), facturada **por token** — un pedazo de palabra de aproximadamente tres cuartos de una palabra en inglés — así que una función cuesta más cada vez que las preguntas o las respuestas se alargan. Pida ver la matemática de tokens, por usuario, por mes; si nadie la ha hecho, la función todavía no tiene precio, solo una esperanza.

La alternativa — ejecutar su propio modelo — significa alquilar **GPUs (Graphics Processing Units, unidades de procesamiento gráfico)**, los chips especializados que la IA requiere: escasos, con precios muchas veces por encima de los servidores ordinarios, y facturando por cada hora encendidos, estén pensando o de brazos cruzados. La primera regla de la línea de IA: **una GPU ociosa es la ociosidad más cara del edificio**. Un servidor de pruebas olvidado desperdicia el dinero del café; una instancia de GPU olvidada desperdicia un salario. Y cuando alguien diga *hagamos fine-tuning de nuestro propio modelo*, recuerde la secuencia honesta: primero prompting cuidadoso sobre una API alquilada, luego darle al modelo sus documentos en el momento de la pregunta, y solo cuando ambos se queden cortos, el fine-tuning.

Tres preguntas lo llevan a través de cualquier propuesta de IA. Una: ¿cuánto cuesta una respuesta equivocada? Los LLM se equivocan con toda confianza una fracción pequeña e irreducible de las veces; la tasa de error aceptable es una decisión de negocio — lo que significa que es suya. **Dos: ¿qué datos salen de nosotros?** Cada pregunta enviada a una API de IA viaja a los servidores del proveedor; si contiene el nombre o el estatus de refugiado de un estudiante, el módulo de cumplimiento aplica en su totalidad. **Tres: ¿cuál es el costo mensual por usuario con diez veces los usuarios?** El precio por token escala linealmente — un piloto cómodo de \$30 se convierte en \$300 cuando tiene éxito, sin que llegue ningún descuento por volumen a salvarlo. Los mismos tres pilares de siempre — confiabilidad, seguridad, costo — con vocabulario nuevo. La tecnología es novedosa; el interrogatorio no lo es.

El capstone — un ejercicio que ensambla el curso entero

El curso ahora termina con un único ejercicio de diseño que usa todo lo que hay en él. El encargo: usted acaba de ser nombrado líder cloud de una organización benéfica educativa en la región fronteriza, en Mae Sot. Su patrimonio — un sitio web público, una base de datos de unos 2,000 expedientes de estudiantes (que incluyen, para algunos, el estatus de documentación), tres centros de aprendizaje, y un ingeniero talentoso y agotado que sostiene el único acceso a producción y el único teléfono de guardia. La factura cloud del mes pasado llegó casi al doble de lo habitual y nadie sabe decir por qué. El único punto de tecnología de la junta: “¿Podemos usar IA para tutorías?”. La tarea: sus primeros 90 días en una página, con la secuencia justificada.

La versión interactiva califica el plan contra una lista de verificación: inventario antes que opiniones; el mapa de tres listas de máquinas, etiquetas y dinero; una victoria rápida; el barrido de seguridad y la prueba de respaldo-y-restauración con fechas puestas; un cable trampa de presupuesto; una posición honesta sobre el riesgo del ingeniero único; la PDPA nombrada como el marco para los expedientes de estudiantes; la pregunta de la IA respondida con las tres preguntas de IA; y una revisión con la junta al día 90, reportada en términos unitarios.
