

Le Cours du Leader Cloud (The Cloud Leader Course)

De zéro connaissance à la direction d'une équipe d'ingénieurs cloud — préparé pour Fedly
Bonneau

13 août 2026

Le Cours du Leader Cloud

De zéro connaissance à la direction fluide d'une équipe d'ingénieurs cloud.

Comment fonctionne ce cours

Vous ne vous formez pas pour *devenir* ingénieur cloud. Vous vous formez pour **diriger** des ingénieurs cloud : suivre chaque conversation dans la salle, poser les questions qui comptent, prendre de bonnes décisions concernant l'argent, les risques et les personnes, et gagner le respect des experts sans prétendre en être un. C'est une compétence différente, entièrement apprenable — et c'est la compétence que ce cours enseigne.

Le cours comporte trois étapes. **Étape 1 (Semaines 1–8) : Parler la langue.** Vous apprenez ce qu'est réellement le cloud (l'informatique en nuage) et le vocabulaire de chaque brique de base, pour que les réunions cessent de ressembler à du bruit. **Étape 2 (Semaines 9–16) : Penser comme la salle.** Architecture, sécurité et argent — les trois conversations que toute équipe cloud a chaque semaine, et les trois où un leader apporte de la valeur ou se fait ignorer. **Étape 3 (Mois 5–24) : Diriger.** Certifications, recrutement, animation des réunions de décision, et le chemin honnête de deux ans pour tenir tête aux ingénieurs seniors.

Règles pour l'ensemble du cours : étudiez **une heure par jour, six jours par semaine** — la régularité l'emporte sur l'intensité. Chaque module se termine par un **Exercice de fluidité (Fluency Drill)** (des phrases à dire à voix haute jusqu'à ce qu'elles deviennent naturelles) et un **Jalon (Milestone)** (la preuve que vous êtes prêt à passer à la suite). Faites-les ; la lecture seule construit la reconnaissance, pas la fluidité. Et dès la Semaine 1, tenez un **Journal de décisions** : chaque fois que vous apprenez un concept, écrivez une phrase sur la façon dont il affecte l'argent, les risques ou les personnes — car cette traduction est tout votre métier.

ÉTAPE 0 — LE REZ-DE-CHAUSSÉE (commencez ici si la technologie est nouvelle pour vous)

Ces deux modules s'adressent au lecteur qui n'a jamais travaillé dans la technologie. Ils sont identiques aux Modules 1 et 2 du cours interactif sur b4lcilc.org. Si vous savez déjà ce que sont un serveur web et une base de données, parcourez-les rapidement ; sinon, ce sont les pages les plus importantes de ce livre — tout ce qui suit repose sur elles.

Module A : Comment fonctionne le web — ce qui se passe quand vous tapez une adresse

La grande idée : une page web n'est pas une image projetée vers votre téléphone. C'est un ensemble d'**instructions écrites**, envoyées par un ordinateur lointain, que votre propre navigateur suit pour construire la page sur votre écran. Apprenez les six étapes entre le moment où vous tapez une adresse et celui où vous voyez une page, et chaque réunion technique du reste de votre carrière deviendra plus facile à suivre.

Le voyage, en six étapes que vous devez savoir raconter à voix haute :

1. Vous tapez un nom — `b4lcilc.org` — dans un **navigateur** (Chrome, Safari, Edge, Firefox).
2. **DNS**, l'annuaire téléphonique d'internet, transforme ce nom en **adresse IP** — l'adresse numérique d'un ordinateur, comme `142.250.66.14`.
3. Votre navigateur — le **client**, celui qui demande — envoie une **requête HTTP** à l'ordinateur situé à cette adresse : « GET — donne-moi cette page. »
4. La requête voyage sous forme de petits **paquets** selon les règles de **TCP/IP**, la plomberie d'internet, qui numérote chaque paquet, confirme la livraison et renvoie tout ce qui s'est perdu en route.
5. L'ordinateur lointain — le **serveur**, celui qui répond — renvoie une **réponse** : les instructions de la page, ou une excuse numérotée comme **404** (« cette page n'existe pas »).
6. Votre navigateur suit ces instructions et construit la page. Les instructions arrivent en trois langages : **HTML** (la structure — titres, paragraphes, boutons), **CSS** (l'apparence — couleurs, tailles, mise en page) et **JavaScript** (le comportement — ce qui se passe quand vous cliquez).

Côté client et côté serveur — le cadre qui organize tout : le travail effectué sur votre appareil, dans le navigateur, est *côté client* (le **front end**) ; le travail effectué sur l'ordinateur lointain est *côté serveur* (le **back end**). La règle d'or à mémoriser : **tout ce qui touche à la confiance ou à la mémoire appartient au serveur ; tout ce qui touche à la sensation de rapidité appartient au client**. Vérifier un mot de passe, enregistrer un dossier, débiter une carte : serveur. Dessiner la page, réagir instantanément pendant que vous tapez : client. Entre les deux se trouve souvent un **proxy** — un serveur intermédiaire qui filtre, protège ou répartit le trafic ; les « load balancers » (répartiteurs de charge) et les « CDN » des modules suivants sont des proxys spécialisés.

Pourquoi le DNS compte pour un leader : le DNS tombe en panne d'une façon bien particulière — quand il casse, tous les sites « disparaissent » d'un coup alors que chaque serveur fonctionne toujours. C'est l'annuaire qui est en panne, pas les téléphones. Quand un ingénieur dit « c'est peut-être le DNS », c'est de cela qu'il parle — et certaines des pannes les plus célèbres d'internet étaient exactement cela.

Et le cadenas : HTTPS, c'est HTTP avec du chiffrement — toute la conversation est brouillée contre quiconque se trouverait entre les deux, et c'est ce que certifie le cadenas dans la barre d'adresse. La règle moderne : tous les sites, tout le temps.

Vocabulaire :

Terme	Définition
Navigateur web	Le programme qui va chercher les pages et les dessine : Chrome, Safari, Edge, Firefox.
URL	L'adresse complète que vous tapez : protocole + nom de l'ordinateur + document demandé.
Adresse IP	L'adresse numérique d'un ordinateur sur un réseau, comme 142.250.66.14.
DNS	L'annuaire téléphonique d'internet : transforme les noms (google.com) en adresses IP.
HTTP / HTTPS	Les règles de requête-et-réponse que parlent navigateurs et serveurs ; le S ajoute le chiffrement (le cadenas).
TCP/IP	La plomberie d'internet : déplace des paquets numérotés, de façon fiable, entre les adresses.
Client / serveur	Celui qui demande et celui qui répond. Une relation, pas un type de machine.
Côté client (front end)	Le travail fait dans le navigateur de l'utilisateur : dessiner la page, réagir instantanément.
Côté serveur (back end)	Le travail fait sur la machine lointaine : la confiance, les règles, les enregistrements permanents.
Proxy	Un serveur intermédiaire qui gère le trafic pour le compte de l'une des deux parties.
HTML / CSS / JavaScript	Les trois langages des pages : structure / apparence / comportement.

Exercice de fluidité : racontez le voyage en six étapes à voix haute, seul, jusqu'à ce qu'il vous prenne moins de trente secondes et que vous ne cherchiez plus jamais un mot. Puis racontez-le à quelqu'un d'autre.

Jalon : vous savez répondre, à table, sans préparation : « Que se passe-t-il réellement quand je tape le nom d'un site web sur mon téléphone ? » — comme une histoire, pas comme une liste de jargon.

Module B : Comment les sites web sont construits et hébergés — d'un seul ordinateur jusqu'au cloud

La grande idée : tout vrai site web est construit en **trois couches** — un front end que les gens touchent, une couche intermédiaire où tournent les règles de l'organisation, et une couche de données qui n'oublie jamais — et depuis vingt-cinq ans, presque tout cela est assemblé à partir

de l'une des deux grandes familles d'outils. Apprenez les trois couches et les deux familles, et les réunions avec les prestataires cesseront d'être une langue étrangère pour devenir un menu.

Le restaurant, que vous utiliserez pendant tout le reste de ce cours : le front end (UI) est la salle du restaurant — tout ce que le client voit, construit en HTML, CSS et JavaScript. La **couche intermédiaire (la logique métier)** est la cuisine — le code sur le serveur qui applique les règles de la maison : « un étudiant ne peut pas s'inscrire deux fois », « les commandes au-dessus de la limite exigent une approbation ». La **couche de données** est le garde-manger et le livre de recettes — la **base de données**, où l'on conserve tout ce qui ne doit jamais être oublié. L'industrie appelle cette forme l'**architecture N-tiers**. Pourquoi des couches ? Pour que chacune puisse être rénovée sans reconstruire les autres — et quand un ingénieur dit « le goulot d'étranglement est dans la couche de données », il vous tend exactement cette carte avec un cercle dessiné dessus.

La machine et le programme : un **serveur** est un ordinateur bien réel — une boîte métallique sans écran, alignée avec des centaines d'autres dans les racks d'un **data centre**, construite pour fonctionner chaque seconde de l'année. Sur cette machine tourne un *programme* appelé lui aussi serveur web, qui mène la conversation HTTP. Connaissez les deux plus célèbres par leur nom : **IIS** (celui de Microsoft, tourne sous Windows) et **Apache** (gratuit, tourne surtout sous Linux). Gardez les mots bien en place : *un navigateur demande ; un serveur web répond*. Apache est un **serveur** web — le côté qui répond — pas un navigateur. L'**hébergement web** consiste à louer de l'espace sur le serveur de quelqu'un d'autre, pour que sa machine réponde en votre nom.

La base de données et SQL, en douceur : une **base de données** est un programme qui stocke des enregistrements et répond aux questions qu'on lui pose — un archiviste fanatiquement organisé qui ne dort jamais. **SQL** (prononcé « sequel ») est simplement **le langage qui sert à poser des questions à une base de données**. Une *requête* (query) est une question. Lisez cette ligne — vous en êtes déjà capable : `SELECT name FROM students WHERE village = 'Mae La'` — *dans la table des étudiants, donne-moi les noms de tous ceux qui viennent de Mae La*. Deux marques dominent : **SQL Server** (celle de Microsoft, sous licence) et **MySQL** (gratuite, open source). Les deux parlent SQL. Le point pour un leader : **la base de données est l'endroit où l'organisation vit réellement** — les pages peuvent être reconstruites, mais perdez les dossiers d'inscription et il n'y a plus d'école. Voilà pourquoi les bases de données reçoivent l'ingénierie la plus protectrice de tout l'édifice.

Les deux familles :

Rôle dans la machine	Famille Microsoft	Famille open source LAMP
Système d'exploitation	Windows Server	Linux
Serveur web (répond au HTTP)	IIS	Apache
Langage de la logique métier	C# sur .NET	PHP
Base de données	SQL Server	MySQL
Modèle de coût	Frais de licence, un seul fournisseur	Gratuit et open source ; vous payez pour les personnes

Rôle dans la machine	Famille Microsoft	Famille open source LAMP
Terrain de prédilection classique	Banques, gouvernements, grandes entreprises	Le web ouvert — WordPress à lui seul fait tourner un tiers de tous les sites web

L'**open source** signifie que la recette du logiciel est publique — chacun peut l'inspecter, l'améliorer et l'utiliser sans frais de licence. C'est ce qui a toujours permis à une organisation débrouillarde de construire un site web de classe mondiale pour le prix d'une seule machine louée.

La route vers le cloud — la phrase sur laquelle repose tout ce livre : pendant des décennies, tout cela tournait **on-premises** (sur site) : dans une salle de serveurs que l'organisation possédait, achetée pour la journée la plus chargée imaginable, inutilisée tous les autres jours, maintenue au frais aux frais de l'organisation. Puis Amazon et Microsoft ont proposé de louer, à l'heure, de la capacité dans leurs propres entrepôts. Les stacks n'ont pas changé — les mêmes Linux, Apache, IIS et SQL Server tournent dans le cloud aujourd'hui. **Ce qui a changé, c'est l'endroit où vivent les machines et qui les possède.** Retenez cette phrase : le Module 1 de l'Étape 1 s'ouvre exactement là.

Vocabulaire :

Terme	Définition
Serveur / data centre	La machine allumée en permanence, et le bâtiment rempli de racks de ces machines.
Serveur web (programme)	Le logiciel qui répond aux requêtes HTTP : IIS (Microsoft), Apache (open source).
Hébergement web	Louer de l'espace sur le serveur de quelqu'un d'autre pour votre site web.
Architecture N-tiers	Construire en couches : front end (UI), couche intermédiaire (logique), couche de données.
Base de données	Le programme qui stocke les enregistrements et répond aux questions — là où vit l'organisation.
SQL / requête (query)	Le langage quasi anglais pour poser des questions à une base de données ; une requête est une question.
SQL Server / MySQL	La base de données sous licence de Microsoft / la favorite du monde open source. Les deux parlent SQL.
Stack Microsoft	Windows Server + IIS + C#/.NET + SQL Server.
Stack LAMP	Linux + Apache + MySQL + PHP — tout est open source.
Open source	Recette publique : libre à inspecter, à

Terme	Définition
On-premises (sur site)	améliorer et à utiliser. Des machines que vous possédez, dans une salle que vous possédez — le monde d'avant le cloud.

Exercice de fluidité : déployez LAMP sur quatre doigts, à voix haute, trois jours de suite. Puis expliquez le restaurant — salle, cuisine, garde-manger — à quelqu'un qui n'a jamais entendu parler d'un serveur.

Jalon : un prestataire vous dit : « Vous êtes sur une vieille installation Microsoft on-prem — IIS et SQL Server. Nous, nous sommes une maison LAMP, et nous vous ferions migrer vers le cloud. » Vous savez traduire chaque mot pour votre conseil d'administration — et vous savez que la partie risquée à sonder est la migration de la base de données.

ÉTAPE 1 — PARLER LA LANGUE (Semaines 1-8)

Module 1 (Semaines 1-2) : Ce qu'est réellement le cloud

La grande idée : le cloud, ce sont *les ordinateurs de quelqu'un d'autre, loués à l'heure, gérés par du logiciel*. Avant le cloud, une entreprise achetait des serveurs physiques, les installait dans une salle et payait des gens pour les maintenir — lent, coûteux, rigide. AWS, Microsoft Azure et Google Cloud ont construit des entrepôts contenant des millions d'ordinateurs (des data centres, ou centres de données) et permettent à quiconque d'en louer des tranches à la seconde, depuis n'importe où, via une page web ou une ligne de code. C'est tout. Tout le reste de ce cours n'est que du détail au-dessus de cette idée unique.

Les trois grands fournisseurs, définis :

Fournisseur	Ce que c'est
AWS (Amazon Web Services)	La division cloud d'Amazon et le plus grand fournisseur cloud au monde (~30 % du marché mondial). Lancée en 2006 quand Amazon a commencé à louer les systèmes informatiques qu'il avait construits pour sa propre boutique. Elle propose aujourd'hui plus de 200 services — serveurs, stockage, bases de données, IA, et plus — loués à la seconde. Quand ce cours dit « le cloud », AWS est l'exemple par défaut, et il a ouvert sa propre Région Thaïlande (Bangkok) en janvier 2025.
Microsoft Azure	Le cloud de Microsoft, n°2 mondial et le plus fort au sein des grandes entreprises, car il se

Fournisseur	Ce que c'est
	connecte naturellement aux outils Microsoft que les entreprises utilisent déjà (Windows, Office, Active Directory). Il construit actuellement sa première région de datacenters en Thaïlande.
Google Cloud (GCP)	Le cloud de Google, n°3 — le plus fort en analyse de données et en outillage IA. Il a engagé 1 milliard de dollars dans un centre de données thaïlandais et une région cloud à Bangkok.

Votre compte AWS gratuit — créez-le dès la Semaine 1. Rendez-vous sur <https://aws.amazon.com/free> et cliquez sur « Create a Free Account » (la page d'inscription directe est https://signin.aws.amazon.com/signup?request_type=register). Il vous faudra une adresse e-mail, un numéro de téléphone et une carte de crédit/débit pour la vérification d'identité — le **Free Tier** (offre gratuite) vous donne une allocation mensuelle gratuite des services de base (dont 750 heures/mois d'un petit serveur EC2 la première année), et les exercices de ce cours restent dans ces limites. Deux habitudes de sécurité dès le premier jour : activez la MFA (authentification multifacteur — le Module 6 l'explique) et configurez une **alerte de facturation** à 5 \$ (le Module 7 explique comment), afin de ne jamais être pris par surprise. Après l'inscription, vous vous connectez sur <https://console.aws.amazon.com> — la « console » est simplement la page web de contrôle d'AWS.

Pourquoi les entreprises utilisent le cloud : la vitesse (un nouveau serveur en 60 secondes au lieu de 6 semaines), l'élasticité (louer 100 serveurs pour le grand jour de soldes, en rendre 90 le lendemain), l'absence de coût initial (dépense d'exploitation au lieu de dépense d'investissement) et la portée mondiale (placer votre application près de vos clients à Bangkok, Tokyo et Francfort sans rien construire).

Les trois couches de services — le modèle mental le plus utile du cloud :

Couche	Ce que vous louez	Analogie de la cuisine	Exemple
IaaS (Infrastructure as a Service — l'infrastructure en tant que service)	Des ordinateurs bruts, du stockage, des réseaux — vous gérez tout ce qui s'y trouve	Louer une cuisine vide : vous apportez les chefs, les recettes, les ingrédients	Amazon EC2, Azure Virtual Machines
PaaS (Platform as a Service — la plateforme en tant que service)	Une plateforme gérée — vous n'apportez que votre application	Louer une cuisine avec personnel : vous apportez la recette	AWS Elastic Beanstalk, Azure App Service
SaaS (Software as a Service — le logiciel en tant que service)	Un logiciel fini	Commander au restaurant	Gmail, Salesforce, Canva

Régions et zones de disponibilité : une **Région** est un groupe géographique de centres de données (la Thaïlande possède sa propre Région AWS depuis janvier 2025, à Bangkok et dans ses environs). Une **zone de disponibilité (Availability Zone, AZ)** est un ou plusieurs centres de données isolés au sein d'une région. Les applications qui comptent tournent dans au moins deux AZ, pour que la défaillance d'un bâtiment ne les fasse pas tomber. Quand les ingénieurs disent « nous sommes multi-AZ », ils veulent dire « un centre de données peut brûler et nous restons en ligne ».

Vocabulaire :

Terme	Définition
Fournisseur cloud / hyperscaler	Une entreprise qui possède d'immenses centres de données et loue de la puissance informatique via internet (AWS, Azure, Google Cloud). « Hyperscaler » = les quelques plus grands, construits pour une échelle quasi illimitée.
Data centre (centre de données)	Un entrepôt sécurisé rempli de milliers de serveurs, avec alimentation et refroidissement industriels — le lieu physique où « le cloud » vit réellement.
Région	Un groupe géographique de centres de données proposé comme un choix de localisation, p. ex. « Asia Pacific (Bangkok) ». Vous choisissez la région où tournent vos systèmes.
Zone de disponibilité (AZ)	Un ou plusieurs centres de données isolés au sein d'une région, avec alimentation et réseau indépendants. Tourner dans deux AZ signifie qu'un bâtiment peut tomber en panne et que vous restez en ligne.
On-premises (« on-prem », sur site)	L'ancienne méthode : des serveurs que votre entreprise possède, dans votre propre bâtiment. Le contraire du cloud.
Migration	Le projet de déplacer des systèmes depuis le on-prem vers le cloud.
Workload (charge de travail)	Toute application ou tout système qui tourne — « le workload de la paie », « le workload du site web ». Un mot pratique : il couvre tout.
Provisionner	Créer/mettre en place une ressource cloud (un serveur, une base de données). « Provisionner un serveur » = le faire exister.
Scale up / scale out (montée en charge verticale / horizontale)	Absorber plus de demande en rendant une machine plus grosse (up) ou en ajoutant plus

Terme	Définition
Latence	de machines (out). Le cloud favorise le <i>out</i> . Le délai avant que les données n'arrivent, mesuré en millisecondes. La distance crée de la latence — la raison pour laquelle une région à Bangkok compte pour les utilisateurs thaïlandais.
Console	Le panneau de contrôle web du fournisseur, où vous voyez et gérez tout ce que vous louez.

Vidéos pour ce module (toutes gratuites, liens vérifiés) :

Vidéo	Chaîne	Durée	Lien
What is AWS?	Amazon Web Services (official)	~2 min	https://www.youtube.com/watch?v=a9_D53WsUs
Top 50+ AWS Services Explained in 10 Minutes	Fireship	~10 min	https://www.youtube.com/watch?v=JlBYCM48to
What is Microsoft Azure? An Introduction	Eye on Tech	~3 min	https://www.youtube.com/watch?v=l9JkLhvaKA8
รู้จัก AWS Cloud คืออะไร (Thai-language intro)	Aware Corporation	courte	https://www.youtube.com/watch?v=nrSpZKGxXd0

Exercice de fluidité — répétez ces phrases jusqu'à ce qu'elles soient naturelles : “Is that workload on-prem or in the cloud?” (« Ce workload est-il sur site ou dans le cloud ? ») · “Which region are we in — and are we multi-AZ?” (« Dans quelle région sommes-nous — et sommes-nous multi-AZ ? ») · “Is this an IaaS approach or is there a managed service that removes the maintenance?” (« Est-ce une approche IaaS ou existe-t-il un service managé qui supprime la maintenance ? »)

Exercices : (1) Créez votre compte AWS gratuit sur <https://aws.amazon.com/free> — le parcours d'inscription lui-même vous apprend plus de vocabulaire qu'un chapitre de lecture. (2) Regardez les quatre vidéos du tableau ci-dessus (moins de 20 minutes au total). (3) Dans votre journal : rédigez l'explication « ascenseur » du cloud que vous donneriez à un directeur d'école thaïlandais en une seule respiration.

Jalon : vous savez expliquer IaaS vs PaaS vs SaaS avec votre propre analogie, et expliquer pourquoi l'ouverture d'une région AWS à Bangkok a compté pour les banques thaïlandaises (réponse : latence + résidence des données — voir Module 6).

Module 2 (Semaines 3-4) : Les briques de base — calcul, stockage et bases de données

Le calcul (compute) — les moteurs qui exécutent le code. Une **machine virtuelle (VM/instance)** est une tranche d'un serveur physique qui se comporte comme un ordinateur complet ; vous choisissez une taille (CPU/RAM) et payez à la seconde d'exécution. Un **container** (conteneur) est une façon plus légère et plus rapide d'empaqueter une application pour qu'elle tourne à l'identique partout ; **Docker** les empaquette et **Kubernetes** (K8s) en orchestre des flottes entières — quand vous entendez « K8s », pensez « le système qui fait tourner et répare automatiquement nos centaines de containers ». Le **serverless** (informatique sans serveur à gérer — AWS Lambda) signifie que vous ne téléversez qu'une fonction de code ; le cloud l'exécute quand elle est déclenchée et vous payez à l'invocation — aucun serveur à gérer. Le schéma à remarquer : VM → container → serverless est un curseur allant de « plus de contrôle, plus de maintenance » vers « moins de contrôle, maintenance quasi nulle ». Les bonnes équipes choisissent par workload, pas par effet de mode.

Le stockage — trois formes. **Le stockage objet** (Amazon S3) : un seau sans fond pour les fichiers — images, vidéos, sauvegardes, jeux de données ; bon marché, durable à onze neufs, la réponse par défaut à « où met-on les fichiers ? » **Le stockage bloc** (EBS) : le disque dur virtuel attaché à une VM. **Le stockage fichier** (EFS) : un lecteur partagé que de nombreuses machines montent en même temps. Puis les **tiers** (classes de stockage) : chaud (accès fréquent, coûteux) vs froid/archive (Glacier — bon marché, lent) — déplacer les vieilles données vers les tiers froids est l'une des économies les plus faciles à livrer pour n'importe quelle équipe.

Les bases de données — deux familles. **Relationnelles/SQL** (MySQL, PostgreSQL ; gérées sous forme d'Amazon RDS/Aurora) : des données en tables à structure stricte ; le choix par défaut pour l'argent, les commandes, les utilisateurs — tout ce où l'exactitude est sacrée. **NoSQL** (DynamoDB, MongoDB) : flexibles, massivement scalables ; le choix par défaut pour des données énormes, rapides et de forme plus simple (sessions, catalogues, fils d'actualité). « Base de données managée » signifie que le fournisseur gère les sauvegardes, les correctifs et le basculement — les équipes devraient avoir une raison solide pour faire tourner la leur elles-mêmes.

Vocabulaire :

Terme	Définition
Instance	Une machine virtuelle (VM) louée. « Spin up an instance » = démarrer un serveur.
Type / taille d'instance	La spécification que vous avez choisie — combien de CPU, combien de mémoire. Type plus gros = prix horaire plus élevé.
Container (conteneur)	Un paquet léger contenant une application plus tout ce dont elle a besoin, pour qu'elle tourne à l'identique sur n'importe quelle machine. Plus rapide et moins cher qu'une VM complète.

Terme	Définition
Docker	L'outil standard pour construire et exécuter des containers.
Kubernetes (K8s)	L'orchestrateur qui fait tourner et répare automatiquement des flottes de containers — redémarre ceux qui plantent, en ajoute sous la charge. « K8s » est le surnom du secteur.
Cluster / nœud	Un cluster (grappe) est un groupe de machines fonctionnant comme un seul système ; chaque machine en son sein est un nœud.
Serverless	Exécuter du code sans gérer aucun serveur — le cloud exécute votre fonction quand elle est déclenchée et facture à l'exécution.
Lambda / fonction	Le produit serverless d'AWS ; une « fonction » est le petit morceau de code qu'il exécute.
S3 / bucket	Le stockage objet d'AWS pour les fichiers ; un bucket (seau) est un conteneur nommé de fichiers. La réponse par défaut à « où met-on les fichiers ? »
Volume EBS	Le disque dur virtuel attaché à une instance.
Durabilité	La probabilité que les données stockées survivent. Les « onze neufs » de S3 (99,999999999 %) signifient que la perte n'arrive pratiquement jamais.
Tier de stockage	Classe prix/vitesse pour les données : chaud (instantané, coûteux) — froid/archive (Glacier — bon marché, de quelques minutes à quelques heures pour récupérer).
RDS	Le service de bases de données relationnelles managé d'AWS — AWS gère pour vous les sauvegardes, les correctifs et le basculement.
SQL vs NoSQL	SQL : tables strictes, parfait pour l'argent et les commandes. NoSQL : flexible et massivement scalable, pour les sessions, catalogues, fils.
Sauvegarde / snapshot	Une copie sauvegardée des données (un snapshot est une copie à un instant T d'un disque ou d'une base de données) à partir de laquelle vous pouvez restaurer.
Failover (basculement)	Le passage automatique à une copie de secours quand le système principal tombe —

Terme	Définition
	la raison pour laquelle les bases de données managées survivent aux mauvaises nuits.

Vidéos pour ce module :

Vidéo	Chaîne	Durée	Lien
Getting Started with EC2	AWS Developers (official)	27 min	https://www.youtube.com/watch?v=nJ-djerESW0
Introduction to Amazon S3	Amazon Web Services (official)	~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/watch?v=VnvRFRk_51k
Serverless Computing in 100 Seconds	Fireship	~2 min	https://www.youtube.com/watch?v=W_VV2Fx32_Y

Exercice de fluidité : “Should this run on VMs, containers, or serverless — and what’s the operational cost of each choice?” (« Cela doit-il tourner sur des VM, des containers ou en serverless — et quel est le coût opérationnel de chaque choix ? ») · “Is that data hot or can it go to a cheaper tier?” (« Ces données sont-elles chaudes ou peuvent-elles passer à un tier moins cher ? ») · “Why are we self-managing that database instead of using RDS?” (« Pourquoi gérons-nous nous-mêmes cette base de données au lieu d’utiliser RDS ? »)

Exercices : (1) Dans votre compte gratuit : lancez la plus petite instance EC2, puis résiliez-la. Téléversez un fichier dans S3. Vous venez d’utiliser personnellement de l’IaaS. (2) Demandez à un assistant IA de vous interroger : « Donne-moi 10 scénarios ; je répondrai VM, container ou serverless, et tu me noteras. »

Jalon : pour toute application simple décrite en une phrase (« un site web où les élèves commandent leur déjeuner »), vous savez nommer à voix haute ses briques de calcul, de stockage et de base de données en 60 secondes.

Module 3 (Semaines 5-6) : Les réseaux, et comment les briques se parlent

Le VPC — votre quartier privé. Un **Virtual Private Cloud** (cloud privé virtuel) est votre propre section clôturée du réseau du fournisseur. À l'intérieur se trouvent des **subnets** (sous-réseaux) — publics (accessibles depuis internet, p. ex. les serveurs web) et privés (inaccessibles, p. ex. les bases de données). La phrase de sécurité la plus courante que vous entendrez : « *la base de données se trouve dans un subnet privé.* » Si vous comprenez pourquoi — les attaquants ne peuvent pas toucher ce qui n'a aucun chemin depuis internet — vous comprenez la moitié de la sécurité réseau.

La couche trafic : un équilibreur de charge (load balancer) répartit le trafic entrant entre plusieurs serveurs (et retire discrètement ceux qui sont malades) ; le **DNS** (Route 53) traduit des noms comme *voentreprise.com* en adresses ; un **CDN** (réseau de diffusion de contenu — CloudFront) met votre contenu en cache dans des centaines de villes pour qu'il se charge vite partout ; une **API** (interface de programmation) est la porte qu'un logiciel offre à un autre — quand les ingénieurs disent « *we'll expose an API* », ils veulent dire « nous donnerons à d'autres logiciels un moyen contrôlé d'utiliser le nôtre » ; une **API gateway** (passerelle API) est l'accueil qui gère ces portes.

Relier les mondes : un **VPN** (tunnel chiffré via internet) ou **Direct Connect** (une ligne physique privée) relie les bureaux/systèmes on-prem d'une entreprise à son cloud. **Cloud hybride** = faire tourner à la fois du on-prem et du cloud (la plupart des grandes entreprises thaïlandaises) ; **multi-cloud** = utiliser plus d'un fournisseur.

Vocabulaire :

Terme	Définition
VPC (Virtual Private Cloud)	Votre propre section privée et clôturée du réseau du fournisseur, où vivent vos systèmes.
Subnet (public / privé)	Une subdivision du VPC. Les subnets publics sont accessibles depuis internet (serveurs web) ; les privés ne le sont pas (bases de données).
Adresse IP	L'adresse numérique d'une machine sur un réseau — la façon dont les ordinateurs se trouvent.
Pare-feu / security group	La liste de règles indiquant quel trafic peut atteindre une machine (« le trafic web entre, rien d'autre »). Un security group (groupe de sécurité) est le pare-feu par serveur d'AWS.
Équilibreur de charge (load balancer)	L'aiguilleur de trafic qui répartit les requêtes entrantes entre plusieurs serveurs et cesse d'envoyer vers ceux en mauvaise santé.
DNS	L'annuaire téléphonique d'internet — il traduit <i>voentreprise.com</i> en adresse IP. Le

Terme	Définition
CDN (Content Delivery Network)	service DNS d’AWS est Route 53. Des copies de votre contenu mises en cache dans des centaines de villes pour qu’il se charge vite partout. Celui d’AWS est CloudFront.
Edge location (point de présence)	L’un de ces points CDN au niveau des villes — « the edge » (la périphérie) signifie proche des utilisateurs.
API	La porte qu’un logiciel offre à un autre. « We’ll expose an API » = nous donnerons à d’autres logiciels un moyen contrôlé d’utiliser le nôtre.
API gateway	L’accueil qui gère ces portes — authentification, limites de débit, journalisation.
VPN	Un tunnel chiffré via l’internet public reliant deux réseaux (p. ex. votre bureau à votre VPC).
Direct Connect	Une ligne physique privée vers le cloud — plus rapide et plus stable que le VPN, à un certain prix.
Hybride / multi-cloud	Hybride : faire tourner on-prem et cloud ensemble (la plupart des grandes entreprises thaïlandaises). Multi-cloud : utiliser plus d’un fournisseur.
Ingress / egress (trafic entrant / sortant)	Le trafic qui entre dans / sort du cloud. L’egress coûte de l’argent — retenez ce mot pour le Module 7.

Vidéo pour ce module : AWS Networking Basics — VPC & Subnets for Beginners — KodeKloud — https://www.youtube.com/watch?v=QM63dyA_4Pc (regardez la première moitié maintenant ; revenez pour le reste après le Module 5).

Exercice de fluidité : “Is the database in a private subnet?” (« La base de données est-elle dans un subnet privé ? ») · “What happens when one web server dies — is the load balancer health-checking?” (« Que se passe-t-il quand un serveur web meurt — l’équilibreur de charge fait-il des contrôles de santé ? ») · “Are we exposing that as an API or is it internal only?” (« Exposons-nous cela comme une API ou est-ce interne uniquement ? »)

Exercices : (1) Demandez à une IA de vous guider pour dessiner, sur papier, le réseau d’une application de livraison de repas : utilisateurs → CDN → équilibreur de charge → serveurs web (subnet public) → base de données (subnet privé). Dessinez-le trois fois, jusqu’à pouvoir le faire de mémoire. (2) Trouvez le diagramme d’architecture de votre entreprise (ou n’importe quel exemple) et entourez chaque boîte que vous savez désormais nommer.

Jalon : vous savez esquisser ce diagramme trois-tiers standard sur un tableau blanc et raconter le chemin du clic d'un client à travers lui.

Module 4 (Semaines 7-8) : Comment travaillent les équipes cloud — DevOps, IaC, et lire la salle

DevOps est la culture qui fusionne « les gens qui écrivent le logiciel » (Dev) et « les gens qui le font tourner » (Ops) en une seule équipe qui livre de petits changements fréquemment et en sécurité, l'automatisation faisant le gros du travail. Son battement de cœur est le **pipeline CI/CD** (chaîne d'intégration et de livraison continues) : chaque changement de code est automatiquement construit, testé et déployé (Continuous Integration / Continuous Delivery). Quand une équipe dit « c'est dans le pipeline », elle veut dire que le robot est en train de le tester et de le livrer.

L'Infrastructure as Code (IaC, l'infrastructure en tant que code) — l'idée qui a tout changé : au lieu de cliquer dans une console pour créer des serveurs, les ingénieurs écrivent des *fichiers texte* qui déclarent l'infrastructure (« deux serveurs, un équilibreur de charge, une base de données, ces règles de pare-feu »), et un outil (**Terraform**, CloudFormation) fait correspondre la réalité au fichier. Pourquoi les leaders s'en soucient : les fichiers vivent dans **Git** (le système de gestion de versions), donc chaque changement d'infrastructure est relu, réversible et auditable — la différence entre un atelier et une usine.

Les rituels auxquels vous assisterez : le **stand-up** (réunion debout quotidienne de 15 minutes : ce qui est fait, ce qui vient, ce qui bloque — écoutez les blocages ; les lever est votre travail), le **sprint** (une unité de travail planifié de 1-2 semaines), la **rétro** (ce qu'il faut améliorer), le **post-mortem/revue d'incident** (après une panne : analyse *sans blâme* de ce qui a échoué et de ce qui empêchera la répétition — la santé d'une équipe se voit à l'honnêteté de ces revues), l'**astreinte (on-call)** (la rotation de qui est réveillé à 3 h du matin ; si l'astreinte est pénible, vos meilleurs éléments partiront — posez la question chaque mois).

Les métriques de santé qui comptent : l'**uptime/la disponibilité** (« trois neufs » = 99,9 % ≈ 8,8 heures d'arrêt/an ; chaque neuf supplémentaire multiplie le coût), le **SLA** (l'engagement contractuel, avec pénalités), le **SLO** (l'objectif interne), le **MTTR** (la vitesse de rétablissement — les équipes matures optimisent le rétablissement, pas le fantasme du zéro panne), la **fréquence de déploiement** (les équipes saines livrent souvent de petits changements ; la peur de déployer est un mauvais signe).

Vocabulaire :

Terme	Définition
DevOps	La culture d'une équipe unique qui construit et fait tourner son logiciel, livrant fréquemment de petits changements grâce à l'automatisation.
Pipeline CI/CD	Le tapis roulant automatisé qui construit, teste et déploie chaque changement de code (Continuous Integration / Continuous Delivery).

Terme	Définition
Déployer / rollback	Déployer : mettre un changement en production. Rollback (retour arrière) : l'annuler vite quand il se comporte mal.
Git / repository / pull request	Git : le système de gestion de versions qui enregistre chaque changement. Repository (« repo », dépôt) : la maison du code d'un projet. Pull request (PR) : un changement proposé qu'un autre ingénieur relit avant sa fusion.
IaC (Infrastructure as Code)	Déclarer l'infrastructure dans des fichiers texte qu'un outil transforme en réalité — relu, réversible, auditable.
Terraform	L'outil d'IaC le plus populaire (fonctionne sur tous les clouds). Celui d'AWS est CloudFormation.
Staging vs production (« prod »)	Staging (pré-production) : la copie de répétition du système. Prod : le vrai système que touchent les clients. « Broke prod » (casser la prod) = la mauvaise journée.
Stand-up	Synchronisation quotidienne de 15 minutes : fait / à venir / bloqué. Écoutez les blocages — les lever est votre travail.
Sprint / backlog	Sprint : une unité de travail planifié de 1–2 semaines. Backlog : la liste de tâches ordonnée qui alimente les sprints.
Rétro	La réunion de fin de sprint sur comment mieux travailler la prochaine fois.
Post-mortem	La revue sans blâme après une panne : ce qui a échoué, pourquoi, ce qui empêche une répétition.
Astreinte (on-call)	La rotation de qui répond à l'alarme de 3 h du matin. Si l'astreinte est pénible, vos meilleurs éléments partent.
Incident / sev-1	Une interruption non planifiée ; « sev-1 » (sévérité un) = la pire sorte, tout le monde sur le pont.
SLA / SLO	SLA : la promesse au client avec pénalités (p. ex. 99,9 % de disponibilité). SLO : l'objectif interne plus strict qui protège le SLA.
MTTR	Mean Time To Recovery (temps moyen de rétablissement) — la vitesse à laquelle vous êtes de retour après une panne. Les équipes matures optimisent cela, pas le fantasme de

Terme	Définition
Monitoring / observabilité	ne jamais tomber. Surveiller la santé en direct via les logs (journaux d'événements), les métriques (chiffres dans le temps) et les alertes (notifications automatiques quand des seuils sont franchis).

Vidéos pour ce module :

Vidéo	Chaîne	Durée	Lien
What is DevOps? REALLY understand it	TechWorld with Nana	~ 15 min	https://www.youtube.com/watch?v=0yWAtQ6wYNM
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=l5klai_GBDE

Exercice de fluidité : “Is that change through the pipeline or was it manual?” (« Ce changement est-il passé par le pipeline ou a-t-il été manuel ? ») · “What did the post-mortem conclude, and what’s the prevention item?” (« Qu’a conclu le post-mortem, et quelle est l’action de prévention ? ») · “What’s our MTTR trending like?” (« Quelle est la tendance de notre MTTR ? ») · “Is this in Terraform, or did someone click it into existence?” (« Est-ce dans Terraform, ou quelqu’un l’a-t-il créé à coups de clics ? ») — ce dernier cas s’appelle le « ClickOps », prononcé avec une moue désapprobatrice.

Exercices : (1) Regardez la discussion d’un vrai compte rendu de post-mortem d’incident (beaucoup sont publics — les rapports de pannes de Cloudflare et d’AWS sont célèbres) et résumez-le dans votre journal en cinq phrases. (2) Assistez à (ou regardez l’enregistrement de) n’importe quel stand-up et notez les trois blocages que vous avez entendus.

Jalon — fin de l’Étape 1 : vous savez assister à une réunion de planification technique de 30 minutes et en suivre ≥ 80 %, et votre journal le prouve : des notes d’une réunion réelle ou simulée avec chaque acronyme correctement développé. **C’est aussi le moment de planifier l’examen AWS Cloud Practitioner (CLF-C02)** — 2 à 4 semaines de préparation concentrée en plus de ces modules est la norme publiée, et le réussir est votre première preuve externe.

ÉTAPE 2 — PENSER COMME LA SALLE (Semaines 9-16)

Module 5 (Semaines 9-10) : L'architecture — juger des conceptions sans les dessiner

Votre rôle dans une revue de conception n'est pas de concevoir — c'est d'interroger. Le cadre qu'utilise toute l'industrie est le **Well-Architected Framework** d'AWS (cadre de bonne architecture), six piliers auxquels toute conception doit répondre : excellence opérationnelle, **sécurité**, **fiabilité**, efficacité des performances, **optimisation des coûts**, durabilité. Apprenez en profondeur les trois en gras ; c'est là que vivent l'argent et le risque.

La fiabilité, en termes simples : tout finit par tomber en panne, donc les bons systèmes le présupposent. La **redondance** (aucun point de défaillance unique — deux exemplaires de tout ce qui est important), le **multi-AZ** (survivre à la perte d'un centre de données), l'**auto-scaling** (mise à l'échelle automatique — des machines ajoutées/retirées automatiquement selon la demande), des **sauvegardes testées** (une sauvegarde non testée est un espoir, pas un plan), le **RTO/RPO** (Recovery Time Objective : combien de temps pouvons-nous être à l'arrêt ; Recovery Point Objective : combien de données pouvons-nous perdre — ces deux chiffres *sont* la conversation sur la reprise après sinistre, et ce sont des décisions business, c'est-à-dire **les vôtres**).

Le triangle des compromis : rapide, pas cher, résilient — choisissez-en deux. Chaque débat d'architecture que vous arbitrez se réduit à savoir où, sur ce triangle, le business doit se situer pour ce workload. Un système de paiement et un site marketing ne méritent pas la même réponse.

Les sept questions du leader — mémorisez-les ; elles vous rendent redoutable dans toute revue de conception :

1. “What happens when this component fails?” (« Que se passe-t-il quand ce composant tombe en panne ? ») — il y a toujours un « quand »
2. “What are the RTO and RPO, and who signed off on them?” (« Quels sont le RTO et le RPO, et qui les a validés ? »)
3. “Where does this run — single AZ, multi-AZ, multi-region — and why?” (« Où cela tourne-t-il — AZ unique, multi-AZ, multi-région — et pourquoi ? »)
4. “What will this cost per month at 10× today's load?” (« Combien cela coûtera-t-il par mois à 10× la charge actuelle ? »)
5. “What's the simplest version that meets the requirement?” (« Quelle est la version la plus simple qui satisfait le besoin ? ») — la sur-ingénierie est la maladie du junior ; la simplicité est la vertu du senior
6. “What are we locked into, and what would leaving cost?” (« À quoi sommes-nous liés, et combien coûterait un départ ? ») — le vendor lock-in (l'enfermement chez un fournisseur) est un prix, parfois digne d'être payé — en connaissance de cause
7. “Who else has built this before — are we inventing or assembling?” (« Qui d'autre a déjà construit cela — inventons-nous ou assemblons-nous ? ») — préférez les patterns ennuyeux et éprouvés

Exercice de fluidité : entraînez-vous à poser les questions 1, 4 et 5 sur un ton chaleureux — elles passent pour de la sagesse ou pour une attaque selon la seule façon de les livrer. “Help me understand what happens if the cache goes down” (« Aidez-moi à comprendre ce qui se passe si le cache tombe ») vaut mieux que “did you think about failure?” (« avez-vous pensé aux pannes ? »).

Exercices : (1) Prenez trois architectures d'exemple (demandez à une IA d'en générer : un site e-commerce, un backend d'application mobile, une plateforme d'analyse de données) et passez les sept questions sur chacune, en écrivant les réponses que vous attendriez. (2) Lisez le résumé d'une page des piliers Well-Architected sur <https://aws.amazon.com/architecture/well-architected/>.

Vidéos pour ce module :

Vidéo	Chaîne	Durée	Lien
The Five Pillars of the AWS Well-Architected Framework	Amazon Web Services (official; a sixth pillar, Sustainability, was added later)	~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

Jalon : lors d'une revue de conception simulée (faites-la avec une IA jouant l'ingénieur), vous posez cinq questions de fond et résumez correctement le point le plus faible de la conception à la fin.

Module 6 (Semaines 11-12) : Sécurité et conformité — la conversation qui peut tuer des entreprises

Le modèle de responsabilité partagée — la première chose à comprendre : le fournisseur sécurise le cloud lui-même (bâtiments, matériel, hyperviseurs) ; **vous** sécurisez ce que vous mettez *dedans* (vos données, vos règles d'accès, vos configurations). La plupart des fuites sont des erreurs de configuration côté client — un bucket S3 laissé public, une clé d'accès divulguée — pas des défaillances du fournisseur. Donc « AWS est sécurisé » et « nous sommes sécurisés sur AWS » sont deux phrases différentes.

Le vocabulaire de la défense : IAM (Identity and Access Management — la gestion des identités et des accès : qui a le droit de faire quoi ; la chose la plus auditée du cloud), le **moindre privilège** (chacun reçoit le minimum d'accès nécessaire — la règle d'or), la **MFA** (deuxième facteur sur chaque connexion humaine — non négociable), le **chiffrement au repos et en transit** (données brouillées sur disque et sur le réseau — le minimum vital, toujours activé), le **compte root** (la clé maîtresse — mise sous clé, jamais utilisée au quotidien), la **gestion des secrets** (mots de passe/clés dans un coffre-fort, jamais dans le code), le **zero trust** (confiance zéro — tout vérifier, ne faire confiance à aucun emplacement réseau par défaut), le **test d'intrusion (penetration test)** (des attaquants engagés qui

éprouvent vos défenses), le **ransomware** (rançongiciel — la raison pour laquelle des sauvegardes testées et hors ligne sont un contrôle de sécurité, pas seulement un contrôle d'exploitation).

Le PDPA — la loi thaïlandaise sur les données, et votre avantage sur le marché. Le Personal Data Protection Act est le RGPD de la Thaïlande : consentement requis pour la collecte de données personnelles, une obligation de **notification de violation sous 72 heures**, des délégués à la protection des données pour les grands traitants, et des règles sur l'envoi de données à l'étranger. L'application est devenue réelle en 2025 — plus de 21,5 millions de THB de premières amendes, dont un hôpital sanctionné pour avoir mal supervisé un *prestataire*. Deux conséquences pour vous : (1) chaque grande entreprise thaïlandaise a désormais un problème de conformité que votre société peut être payée pour gérer ; (2) la **résidence des données** — garder les données thaïlandaises sur le sol thaïlandais — est un vrai moteur qui pousse les workloads vers les régions de Bangkok. Ce paragraphe est la moitié de votre argumentaire commercial en Thaïlande ; connaissez-le par cœur.

Les questions de sécurité du leader : “Who has access to production, and when did we last review the list?” (« Qui a accès à la production, et quand avons-nous revu la liste pour la dernière fois ? ») · “Are we alerted on unusual access, or would we find out from the news?” (« Sommes-nous alertés en cas d'accès inhabituel, ou l'apprendrions-nous par la presse ? ») · “When did we last *restore* a backup?” (« Quand avons-nous *restauré* une sauvegarde pour la dernière fois ? ») — pas « avons-nous des sauvegardes » · “If we lost this dataset, is it a PDPA-notifiable breach — and could we notify within 72 hours?” (« Si nous perdions ce jeu de données, serait-ce une violation à notifier au titre du PDPA — et pourrions-nous notifier sous 72 heures ? ») · “What did the last pen test find, and what's still open?” (« Qu'a trouvé le dernier test d'intrusion, et qu'est-ce qui reste ouvert ? »)

Vocabulaire :

Terme	Définition
IAM (Identity and Access Management)	Le système qui contrôle qui (personnes et logiciels) a le droit de faire quoi dans votre cloud — la chose la plus audité qui s'y trouve.
Rôle / policy	Une policy (politique) est un ensemble de permissions écrit ; un rôle est un lot de policies « endossable » qu'une personne ou un programme assume.
Moindre privilège	La règle d'or : chacun reçoit le minimum d'accès dont son travail a besoin, rien de plus.
MFA (authentification multifacteur)	Une deuxième preuve (code sur téléphone, clé matérielle) en plus d'un mot de passe. Non négociable sur chaque connexion humaine.
Chiffrement au repos / en transit	Données brouillées pendant leur stockage / pendant leur trajet sur le réseau. Les deux

Terme	Définition
	toujours activés ; le minimum vital.
KMS (gestion des clés)	Le service qui détient et fait tourner les clés de chiffrement.
Secrets	Mots de passe, clés d'API, jetons — stockés dans un coffre-fort, jamais écrits dans le code.
Compte root	La clé maîtresse de tout le compte cloud. Mise sous clé avec MFA, jamais utilisée pour le travail quotidien.
Zero trust	Une posture de sécurité qui vérifie chaque requête et ne fait confiance à aucun emplacement réseau par défaut.
Vulnérabilité / patching	Une faiblesse connue d'un logiciel / l'application du correctif. Les systèmes non corrigés sont le point de départ de la plupart des intrusions.
Pen test (test d'intrusion)	Des attaquants éthiques engagés pour prouver où vos défenses échouent avant que les vrais ne le fassent.
Ransomware	Une attaque qui chiffre vos données contre rançon — la raison pour laquelle des sauvegardes testées et hors ligne sont un contrôle de <i>sécurité</i> .
SOC 2 / ISO 27001	Les labels d'audit de sécurité indépendants que les clients grands comptes exigent avant de signer.
PDPA	Le Personal Data Protection Act de la Thaïlande — consentement, notification de violation sous 72 heures, règles de transfert transfrontalier. Appliqué avec de vraies amendes depuis 2025.
Résidence des données	Garder les données physiquement à l'intérieur des frontières d'un pays — une raison clé pour laquelle les workloads thaïlandais migrent vers les régions de Bangkok.
DPO / notification de violation	Data Protection Officer (délégué à la protection des données, requis pour les grands traitants) / l'obligation de signaler une violation de données personnelles sous 72 heures.

Vidéo pour ce module : The AWS Shared Responsibility Model — Digital Cloud Training — 4 min — <https://www.youtube.com/watch?v=ESPBBEK-cvo>

Exercice de fluidité : “Is that bucket public? Why?” (« Ce bucket est-il public ? Pourquoi ? ») · “Least privilege — does the intern really need prod access?” (« Moindre privilège — le stagiaire a-t-il vraiment besoin d’un accès à la prod ? ») · “Where does the PDPA line sit in this design — what’s personal data here and where does it physically live?” (« Où passe la ligne du PDPA dans cette conception — qu’est-ce qui relève des données personnelles ici et où vivent-elles physiquement ? »)

Exercices : (1) Lisez l’histoire d’une célèbre fuite due à une mauvaise configuration cloud (Capital One 2019 est le classique) et écrivez-en la version en trois phrases pour votre journal. (2) Faites simuler par une IA un client demandant « pourquoi devrais-je vous confier mes données ? » et entraînez-vous à répondre avec le langage de la responsabilité partagée et du PDPA.

Jalon : vous savez expliquer le modèle de responsabilité partagée et la signification pratique du PDPA à un chef d’entreprise thaïlandais non technique en moins de trois minutes — car cette explication est le rendez-vous commercial d’un MSP (fournisseur de services managés).

Module 7 (Semaines 13-14) : L’argent — l’économie du cloud et le FinOps

C’est ici qu’un leader non-ingénieur apporte de la valeur le plus vite, car la plupart des ingénieurs ne l’ont jamais appris et le gaspillage est énorme — les enquêtes montrent invariablement qu’environ un tiers des dépenses cloud est gaspillé.

Comment tourne le compteur : le calcul est facturé à la seconde où il est *allumé* (pas à l’usage — un serveur inactif facture plein pot ; « on l’a laissé tourner » est le gaspillage classique), le stockage au Go-mois, et — le piège célèbre — l’**egress** : les données qui *sortent* du cloud coûtent de l’argent, tandis que les données entrantes sont gratuites. Les grosses factures d’egress surprennent tout le monde une fois ; après ce module, pas vous.

Le menu tarifaire : **on-demand** (à la demande : plein tarif, flexibilité totale) · **reserved instances / savings plans** (instances réservées / plans d’épargne : s’engager 1 à 3 ans pour 30-70 % de réduction — le plus grand levier sur un workload stable) · **spot instances** (instances spot : jusqu’à 90 % de réduction pour le travail interruptible comme les traitements par lots) · **rightsizing** (redimensionnement : la plupart des serveurs sont surdimensionnés ; les réduire est de l’argent gratuit) · le tiering du stockage (Module 2) · **l’auto-scaling comme outil de coût** (pourquoi payer la capacité de minuit au prix de midi ?).

Le FinOps (la gestion financière du cloud) est la pratique consistant à rendre la dépense cloud visible, allouée et optimisée en continu : le **tagging** (étiquetage) de chaque ressource avec son propriétaire et son projet (une dépense non taguée est une dépense sans responsable), le **showback/chargeback** (montrer sa facture à chaque équipe — le comportement change instantanément), des budgets avec **alertes** (ne jamais découvrir un dépassement sur la facture), et les **unit economics** (économie unitaire) — la métrique du leader : non pas « notre facture est de 800 k€ / mois » mais « notre coût *par transaction client* baisse ». La certification FinOps Certified Practitioner demande environ deux semaines et est, pour un leader côté business, le certificat au meilleur ratio crédibilité/heure de tout l’univers cloud. Obtenez-la.

Les questions d'argent du leader : “What’s our cost per [customer/transaction/tenant], and which direction is it moving?” (« Quel est notre coût par [client/transaction/locataire], et dans quel sens évolue-t-il ? ») · “What percentage of our steady workload is on reservations?” (« Quel pourcentage de notre workload stable est sous réservation ? ») · “What’s untagged?” (« Qu’est-ce qui n’est pas tagué ? ») · “What died but is still billing?” (« Qu’est-ce qui est mort mais facture encore ? ») — disques orphelins et IP inutilisées : chaque compte en a · “What would this bill look like at 10× growth — does our architecture get cheaper or more expensive per unit?” (« À quoi ressemblerait cette facture à une croissance de 10× — notre architecture devient-elle moins chère ou plus chère à l’unité ? »)

Vocabulaire :

Terme	Définition
On-demand	La tarification à l’usage : plein tarif, résiliable à tout moment. Le choix par défaut, et la façon la plus chère de faire tourner des workloads stables.
Reserved / savings plan	Un engagement de 1 à 3 ans sur un usage stable en échange de 30–70 % de réduction — le plus grand levier de coût.
Spot	De la capacité excédentaire jusqu’à 90 % moins chère, que le fournisseur peut reprendre avec quelques minutes de préavis — parfait pour le travail par lots interruptible.
Rightsizing	Réduire les serveurs surdimensionnés à ce qu’ils utilisent réellement. De l’argent gratuit dans presque tous les comptes.
Egress	Les données qui quittent le cloud — facturées au Go alors que les données entrantes sont gratuites. La célèbre surprise sur les factures.
Tagging	Étiqueter chaque ressource avec propriétaire/projet/environnement pour que chaque baht dépensé soit attribuable. Dépense non taguée = dépense sans responsable.
Showback / chargeback	Montrer à chaque équipe sa propre facture cloud (showback) ou la lui facturer réellement en interne (chargeback). Le comportement change instantanément.
Alerte budgétaire	Un avertissement automatique à un seuil de dépense — pour ne jamais apprendre un dépassement par la facture.
Unit economics	Le coût par unité business — par client, par transaction — la métrique du leader, plus

Terme	Définition
TCO (coût total de possession)	parlante que la facture totale. Le coût complet d'un choix sur sa durée de vie : licences, personnes, énergie, migration — pas seulement le prix affiché.
FinOps	La discipline (et la culture d'équipe) consistant à rendre la dépense cloud visible, allouée et optimisée en continu.

Vidéo pour ce module : What is FinOps? — FinOps Foundation (official) — 2 min — https://www.youtube.com/watch?v=Y-c_xw9bHFW · puis le cours gratuit « Introduction to FinOps » sur <https://learn.finops.org>.

Exercices : (1) Ouvrez le calculateur de prix AWS et chiffrez un petit système réel (deux serveurs, une base de données, 500 Go de stockage, 1 To d'egress) — le faire une fois démystifie toutes les futures conversations sur les coûts. (2) Demandez à une IA de jouer un ingénieur défendant un serveur surdimensionné ; entraînez-vous à négocier le rightsizing avec bienveillance.

Jalon : passez (ou planifiez) l'examen FinOps Practitioner et, dans une revue simulée, trouvez quatre problèmes de coûts dans un exemple de facture qu'une IA génère pour vous.

Module 8 (Semaines 15–16) : Bootcamp de fluidité — assembler la langue

Cette quinzaine est de l'intégration pure — la différence entre connaître les mots et *être fluide en réunion*.

Exercice quotidien (30 min) : faites générer par une IA une transcription de réunion réaliste (revue de conception, rétro d'incident ou revue de coûts), avec deux erreurs techniques délibérément enfouies dedans. Votre travail : résumer la réunion en cinq phrases, repérer les erreurs et écrire les trois questions que vous auriez posées. Alternez les types de réunion chaque jour.

Le muscle de la traduction (15 min) : prenez une affirmation technique par jour et traduisez-la pour trois publics — le directeur financier (l'argent), un client (risque/bénéfice) et un jeune ingénieur junior (l'enseignement). Exemple : « Nous déplaçons le stockage des sessions de la base de données vers Redis » → directeur financier : « cela réduit la charge de la base de données, donc nous reportons une mise à niveau de 2 M€ » → client : « les pages se chargent plus vite en heure de pointe » → junior : « Redis garde les données chaudes en mémoire, donc nous arrêtons de marteler la base de données à chaque clic. »

Pratique de lecture (15 min) : un vrai article de blog d'ingénierie par jour (AWS Architecture Blog, ou les blogs d'ingénierie de Netflix/Grab — Grab est particulièrement pertinent : échelle sud-est-asiatique, marché proche de la Thaïlande). Vous en comprendrez désormais 70–80 %. Cherchez le reste.

Votre compagnon de préparation à l'examen pour cette étape : le célèbre cours complet gratuit — *AWS Certified Cloud Practitioner (CLF-C02) 2026* d'Andrew Brown sur freeCodeCamp

— <https://www.youtube.com/watch?v=7HKot-brXFE> (l'édition antérieure de 14 heures, valable pour le même code d'examen, est sur <https://www.youtube.com/watch?v=NhDYbskXRgc>). Regardez-le à 1,25× au fil des Semaines 13–16 ; après les Modules 1–7, l'essentiel vous fera l'effet d'une révision, ce qui est exactement le signe que vous êtes prêt.

Jalon — fin de l'Étape 2, votre examen de fin d'études : (1) Réussissez l'**AWS Cloud Practitioner** si ce n'est pas déjà fait. (2) L'épreuve simulée : une IA joue un ingénieur senior qui vous présente une architecture défectueuse pour un client e-commerce thaïlandais ; vous devez trouver la base de données en AZ unique, le bucket S3 public, l'estimation d'egress manquante et la considération PDPA absente — et livrer le retour sur un ton qui donne à « l'ingénieur » le sentiment d'être aidé, pas pris en faute. Quand vous savez faire cela, vous êtes redoutable en conversation, seize semaines après le début.

ÉTAPE 3 — DIRIGER (Mois 5–24)

Module 9 (Mois 5–8) : La certification approfondie et le plancher technique

Travaillez le matériel de l'**AWS Solutions Architect Associate (SAA-C03)** — 2 à 3 mois à votre rythme d'une heure par jour. Vous passerez ou non l'examen (en tant que leader, c'est le *matériel* qui a de la valeur ; le badge est du théâtre optionnel), mais c'est ici que les régions, les VPC, l'IAM et la tarification cessent d'être du vocabulaire pour devenir un système connecté dans votre tête. En parallèle, maintenez les exercices quotidiens du Module 8 à demi-dose. C'est aussi la fenêtre pour ajouter les **fondamentaux Azure (niveau AZ-900)** — la Thaïlande est un marché d'entreprises très orienté Microsoft, et une littératie bilingue (AWS+Azure) élargit vos conversations clients.

Module 10 (Mois 5–12, en continu) : Recruter et manager des ingénieurs

Recruter quand on ne peut pas pleinement juger la compétence : la structure bat l'instinct. Utilisez une boucle cohérente : un entretien de présélection que vous menez (motivation, communication, la façon dont la personne explique un projet passé à un non-ingénieur — si elle n'y arrive pas, elle échouera aussi avec vos clients) + un entretien technique conduit par votre **bar-raiser** (le garant du niveau : votre n°2 technique ou un contractuel senior rémunéré) + un appel de référence où vous posez exactement la seule question qui compte : “would you hire this person again for this role?” (« réembaucheriez-vous cette personne pour ce poste ? »). Guettez les deux archétypes d'échec : le beau parleur sans profondeur (votre bar-raiser le repère) et l'expert profond mais silencieux (souvent de l'or dans les équipes thaïlandaises, où la modestie est culturelle — ne laissez pas le vernis d'entretien peser plus lourd que les preuves de travail réel).

Le n°2 technique — la décision la plus importante de l'aventure : recrutez-le en premier, payez-le avec une part de capital significative (10–20 % s'il s'agit d'un vrai cofondateur), et définissez l'accord explicitement : il tient le niveau technique et possède les décisions d'architecture ; vous possédez les clients, l'argent, les priorités et les personnes ; vos désaccords se règlent en privé et se résolvent avant que l'équipe ne les voie.

Les rituels qui font rester les ingénieurs : des **1:1** (entretiens individuels) hebdomadaires ou bimensuels qui parlent d'*eux* (carrière, frictions, énergie — pas de points d'avancement) ; un plan de progression écrit par personne (dans le marché thaïlandais en pénurie de 70 000 talents/an, la progression et les budgets de certification retiennent mieux que le salaire seul — payez chaque certification, avec une prime acquise sur 12 mois à l'obtention) ; le crédit en public, la correction en privé ; et une protection impitoyable de leur temps de concentration — un leader qui annule une réunion en plein sprint est un héros.

Les forums de décision — comment prendre des décisions techniques que vous ne pouvez pas pleinement évaluer : pour toute grande décision, exigez de l'ingénieur qui propose un **document de décision** d'une page : le problème, 2–3 options, coûts, risques, recommandation. Puis animez la réunion avec vos sept questions du Module 5. Vous n'êtes pas le meilleur architecte de la salle et n'avez jamais besoin de l'être — vous êtes la personne qui fait gagner l'option *la mieux argumentée*, dans les délais, avec les compromis d'argent et de risque rendus explicites. Les ingénieurs respectent profondément cela quand c'est fait honnêtement ; notez qui a prédit quoi (votre Journal de décisions, encore) et revoyez les prédictions chaque trimestre — cela vous calibre, eux comme vous.

Module 11 (Mois 6–24) : La courbe de crédibilité, honnêtement

L'expérience publiée dans la littérature du leadership d'ingénierie converge vers cette chronologie, et prétendre le contraire est la façon dont les fondateurs non techniques échouent : **~90 jours** pour mener avec compétence le rythme business · **~6 mois** pour une efficacité de base (réunions fluides, décisions structurées, équipe stable) · **12–18 mois** avant que vos intuitions techniques ne valent quelque chose par elles-mêmes · **~2 ans** avant de tenir véritablement tête aux ingénieurs seniors sur les compromis d'architecture. Les parades pendant que la courbe grimpe : empruntez de la crédibilité (votre n°2 présente la moitié technique des rendez-vous commerciaux — les clients ont de toute façon besoin de voir le banc), ne bluffez jamais (le tueur de crédibilité le plus rapide ; “I don't know — walk me through it” (« Je ne sais pas — expliquez-moi ») est une phrase de leader), et laissez vos questions parler pour vous : un leader qui demande “what's our RPO and who signed off on it?” (« quel est notre RPO et qui l'a validé ? ») sonne comme trente ans de cicatrices, et après ce cours, vous le penserez vraiment.

Module 12 (En continu) : La couche marché thaïlandais

Intégrez les spécificités de la Partie 5 du TSI comme savoir opérationnel : le **PDPA** à la fois comme obligation de conformité et comme produit (§Module 6) ; les **échelons de partenariat** (AWS Select exige une poignée de collaborateurs certifiés + 3 projets lancés ; Microsoft Solutions Partner exige un score de capacité de 70/100 — les certifications de votre équipe sont littéralement des actifs commerciaux, une raison de plus de les financer) ; la **promotion BOI** pour une détention étrangère à 100 % de la société ; les fourchettes salariales de Bangkok (junior ฿50–75k — architecte ฿180–280k/mois) pour tarifier correctement offres et propositions ; et les chiffres de vente du moment — plus de 27 milliards de dollars d'investissements approuvés en centres de données, un mandat gouvernemental « cloud-first » et un déficit de compétences de 70 000 personnes/an que votre banc certifié existe pour combler.

Le programme en une page

Quand	Focus	Preuve externe
Semaines 1–2	Ce qu'est le cloud ; IaaS/PaaS/SaaS ; régions/AZ	—
Semaines 3–4	Calcul, stockage, bases de données	—
Semaines 5–6	Réseaux ; lecture de diagrammes d'architecture	—
Semaines 7–8	DevOps, IaC, rituels d'équipe, métriques d'exploitation	Planifier le Cloud Practitioner
Semaines 9–10	Jugement d'architecture ; les sept questions	—
Semaines 11–12	Sécurité ; PDPA ; responsabilité partagée	Examen AWS Cloud Practitioner
Semaines 13–14	Économie du cloud ; FinOps	FinOps Practitioner (≈2 semaines de préparation)
Semaines 15–16	Bootcamp de fluidité ; épreuve simulée	Diplôme : l'épreuve finale
Mois 5–8	Matériel SAA-C03 ; Azure AZ-900	Examen SAA (optionnel)
Mois 5–12	Recrutement ; n°2 technique ; forums de décision	Premiers recrutements bien faits
Mois 6–24	Courbe de crédibilité ; couche marché thaïlandais	Échelon partenaire ; premiers contrats récurrents

Un mot de conclusion de votre professeur. Dans seize semaines, vous suivrez chaque conversation dans la salle. Ce n'est pas la ligne d'arrivée — c'est le permis de commencer. La courbe de deux ans vers un vrai jugement technique n'est pas un mur ; c'est un fossé défensif : chaque semaine que vous en accomplissez est une semaine que les fondateurs non techniques de vos concurrents n'ont pas faite. Étudiez chaque jour, consignez chaque décision, ne bluffez jamais, et embauchez des gens meilleurs que vous en faisant en sorte qu'ils soient heureux d'être venus. C'est tout le métier.

Compagnon de « The Thailand Strategic Investment (TSI) », Partie 5. Les estimations de temps de préparation aux certifications et les chronologies de leadership sont tirées des sources qui y sont citées (CBT Nuggets, StudyTech, FinOps Foundation, First Round Review, The Pragmatic Engineer).

NOUVELLES LEÇONS — ÉDITION D'AOÛT 2026

En août 2026, de nouvelles leçons ont été ajoutées aux trois cours interactifs de b4lcilc.org — les cours *Leader Cloud*, *Ingénieur Cloud* et *Architecte Cloud*. Cet addendum met l'édition de lecture à jour avec des condensés fidèles de ce nouveau matériel, dans la même voix que le reste de ce livre.

Un mot sur ce que vous tenez entre les mains. Ces pages sont l'édition de lecture : les idées, les métaphores, et les phrases qui valent la peine d'être dites à voix haute. Les versions interactives complètes en ligne contiennent considérablement plus — chaque nouvelle leçon est livrée avec des quiz qui vérifient votre compréhension avant de vous laisser avancer, des laboratoires pratiques (dont un terminal d'entraînement qui se comporte comme le vrai mais ne peut pas coûter d'argent), des vidéos, des exercices de vocabulaire, et — pour le matériel de synthèse — des exercices de conception guidés avec des listes de contrôle des exigences et des corrigés types. Lisez ces condensés pour installer la forme de chaque idée dans votre tête ; puis, quand vous le pourrez, passez les versions interactives pour ancrer la connaissance.

Les ajouts se répartissent en trois groupes. Pour le cours **Leader Cloud** : une introduction en douceur au terminal, la loi thaïlandaise sur la vie privée, un guide du leader face aux propositions d'IA et aux factures d'IA, et un exercice de synthèse qui assemble le cours entier sur une page. Pour le cours **Ingénieur Cloud** : l'informatique serverless — le modèle qui transforme les petits scripts d'exploitation en fonctions qui ne coûtent rien à exécuter. Pour le cours **Architecte Cloud** : l'arithmétique des promesses de disponibilité, le modèle de génération augmentée par récupération (Retrieval-Augmented Generation) sur lequel les clients posent désormais des questions chaque semaine, et un exercice complet de chiffrage — estimer le prix d'une conception avant que quiconque ne la construise.

Pour le cours *Leader Cloud*

La fenêtre noire — comment parler directement à un ordinateur

Avant votre premier laboratoire, faites connaissance avec l'outil dans lequel il s'exécute. Un **terminal** (aussi appelé *ligne de commande* ou *console*) est un programme qui ressemble à une simple fenêtre noire avec du texte clignotant. C'est la plus ancienne façon d'utiliser un ordinateur, et c'est toujours ainsi que les ingénieurs font le travail sérieux — parce qu'une instruction tapée est précise, répétable, et peut être notée et partagée, d'une manière dont cliquer dans des menus ne le sera jamais.

Le terminal affiche un **prompt** (l'invite de commande) — habituellement une courte ligne se terminant par \$ — qui signifie exactement une chose : *j'écoute*. Une **commande** est une phrase avec une grammaire : le premier mot est l'*outil* que vous appelez (comme `git` ou `aws`), les mots suivants disent *quoi faire*, et les morceaux commençant par un tiret, appelés **flags** (comme `--output table`), ajustent *comment* le faire.

Deux coutumes surprennent chaque nouveau venu. Premièrement : **le silence signifie généralement le succès**. Beaucoup de commandes, quand elles fonctionnent parfaitement, n'affichent rien du tout — la machine considère que faire le travail est une réponse suffisante. Deuxièmement : le terminal ne devine pas. Là où un moteur de recherche pardonne

l'orthographe, le terminal ne pardonne rien — une seule lettre erronée et il vous dit poliment que la commande n'existe pas. Cette rigueur semble dure pendant une journée, puis devient un réconfort : la machine ne fait jamais à *peu près* ce que vous avez demandé.

Encore un morceau de grammaire : les ordinateurs organisent tout en **fichiers** vivant dans des **dossiers** (aussi appelés *répertoires*) — ainsi `mkdir decisions` signifie « *make directory* : crée un répertoire appelé decisions ». Les laboratoires du cours s'exécutent dans un **simulateur** : un terminal d'entraînement qui ne peut rien créer, rien supprimer et rien coûter — mais les commandes sont les vraies, lettre pour lettre. Maîtrisez-les ici, et le jour où vous vous assierez devant un vrai terminal, vos mains connaîtront déjà le chemin.

Le PDPA de la Thaïlande — la loi sur la vie privée la plus proche de chez vous

Le reste du module de conformité est mondial ; ceci est la loi dont le régulateur peut réellement frapper à la porte. Le PDPA de la Thaïlande — le **Personal Data Protection Act (loi sur la protection des données personnelles), B.E. 2562 (2019)** — est pleinement entré en vigueur en 2022 et s'applique à toute organisation en Thaïlande manipulant des données personnelles. Une association caritative de Mae Sot détenant des dossiers d'élèves est en plein dedans.

La bonne nouvelle est véritablement bonne : le PDPA est une quasi-copie du RGPD — le Règlement général sur la protection des données européen. Les rédacteurs thaïlandais ont pris le règlement européen comme modèle, si bien que le RGPD vous a déjà enseigné environ quatre-vingt-dix pour cent de cette loi. Les données personnelles appartiennent à la personne qu'elles décrivent ; vous avez besoin d'une base légale — le plus souvent le **consentement**, librement donné et révoquant — pour les collecter ; et la **limitation des finalités** interdit d'utiliser des données recueillies pour un objectif à un autre : des dossiers collectés pour inscrire un élève ne peuvent pas devenir discrètement une liste de diffusion pour la collecte de fonds.

Le régulateur est le **PDPC — le Personal Data Protection Committee** (comité de protection des données personnelles) — dont les sanctions vont jusqu'à la responsabilité pénale, pas seulement des amendes. L'horloge des violations vous est familière elle aussi : une violation susceptible de présenter un risque pour les personnes doit être signalée au PDPC **sous 72 heures**.

Le point de leadership mérite d'être énoncé clairement. Une organisation détenant des dossiers sur des élèves — des noms, des familles, et dans cette région parfois le statut de documentation, à peu près ce qu'il y a de plus sensible comme données — **répond d'abord à cette loi** : la réponse doit satisfaire Bangkok avant de se soucier de Bruxelles. Les questions à poser à votre équipe : quelles données personnelles détenons-nous, et avons-nous énoncé la finalité au moment de les collecter ? Pourrions-nous supprimer complètement les dossiers d'un élève si on nous le demandait ? En cas de violation cette nuit, pourrions-nous dire au PDPC sous 72 heures ce qui a été pris et à qui cela appartenait ? Aucune de ces questions ne requiert un juriste pour être posée. Toutes requièrent un leader pour être exigées.

Diriger à travers la vague de l'IA

Tôt ou tard — et ces derniers temps c'est plutôt tôt — une conception arrivera sur votre bureau avec de l'IA dedans. Votre travail ne change pas : vous interrogez, vous ne concevez pas.

Le moteur derrière la plupart des fonctionnalités d'IA est un **LLM — un Large Language Model** (grand modèle de langage) — et le fait commercial crucial : vous n'en possédez presque jamais un ; vous louez l'accès via une **API d'IA** (Application Programming Interface — interface de programmation d'applications), facturée **au token** — un fragment de mot représentant environ les trois quarts d'un mot anglais — de sorte qu'une fonctionnalité coûte plus cher dès que les questions ou les réponses s'allongent. Demandez à voir le calcul des tokens, par utilisateur, par mois ; si personne ne l'a fait, la fonctionnalité n'a pas encore de prix, seulement un espoir.

L'alternative — faire tourner votre propre modèle — signifie louer des **GPU (Graphics Processing Units)**, les puces spécialisées que l'IA exige : rares, facturées plusieurs fois le prix des serveurs ordinaires, et payées pour chaque heure où elles sont allumées, qu'elles réfléchissent ou qu'elles ne fassent rien. La première règle de la ligne IA : **un GPU inactif est l'inactivité la plus chère du bâtiment**. Un serveur de test oublié gaspille le prix du café ; une instance GPU oubliée gaspille un salaire. Et quand quelqu'un dit *affinons notre propre modèle (fine-tuning)*, souvenez-vous de la séquence honnête : d'abord un prompting soigné sur une API louée, puis donner vos documents au modèle au moment de la question, et seulement quand les deux échouent, le fine-tuning.

Trois questions vous portent à travers n'importe quelle proposition d'IA. Un : combien coûte une mauvaise réponse ? Les LLM se trompent avec assurance une petite fraction irréductible du temps ; le taux d'erreur acceptable est une décision business — ce qui signifie qu'elle est la vôtre. **Deux : quelles données nous quittent ?** Chaque question envoyée à une API d'IA voyage vers les serveurs du fournisseur ; si elle contient le nom d'un élève ou un statut de réfugié, le module de conformité s'applique en entier. **Trois : quel est le coût mensuel par utilisateur à dix fois plus d'utilisateurs ?** La tarification au token évolue linéairement — un confortable pilote à 30 \$ devient 300 \$ quand il réussit, sans qu'aucune remise de volume n'arrive pour vous sauver. Les trois mêmes piliers que toujours — fiabilité, sécurité, coût — portant un vocabulaire nouveau. La technologie est inédite ; l'interrogatoire ne l'est pas.

La synthèse — un seul exercice qui assemble tout le cours

Le cours se termine désormais par un unique exercice de conception qui utilise tout ce qu'il contient. Le brief : vous venez d'être nommé leader cloud d'une association caritative d'éducation en région frontalière, à Mae Sot. Son parc — un site web public, une base de données d'environ 2 000 dossiers d'élèves (incluant, pour certains, le statut de documentation), trois centres d'apprentissage, et un ingénieur talentueux et épuisé détenant le seul accès à la production et le seul téléphone d'astreinte. La facture cloud du mois dernier est arrivée à presque le double du montant habituel et personne ne peut dire pourquoi. Le seul point technologique du conseil d'administration : « Peut-on utiliser l'IA pour le tutorat ? » La tâche : vos 90 premiers jours sur une page, avec la séquence justifiée.

La version interactive note le plan par rapport à une liste de contrôle : l'inventaire avant les opinions ; la carte en trois listes des machines, des étiquettes et de l'argent ; une victoire rapide ; le balayage de sécurité et le test de restauration de sauvegarde avec des dates attachées ; un fil d'alerte budgétaire ; une position honnête sur le risque de l'ingénieur unique ; le PDPA nommé comme cadre pour les dossiers d'élèves ; la question de l'IA traitée avec les trois questions de l'IA ; et une revue de conseil au jour 90, rapportée en termes unitaires.
