

دورة مهندس معماري السحابة (The Cloud Architect Course)

من الصفر المعرفي إلى تصميم الأنظمة السحابية — دورة من سلسلة B4LCILC
13 أغسطس 2026

دورة مهندس معماري السحابة (The Cloud Architect Course)

من الصفر المعرفي إلى تصميم أنظمة سحابية تراهن عليها شركة بعملها كله.

كيف تعمل هذه الدورة

أنت لا تتدرب لكي تشغل الأنظمة السحابية. أنت تتدرب لكي تصممها: أن تأخذ مشكلة عمل فوضوية ("نريد البيع لمليون عميل دون أن نفقد طلبًا واحدًا أبدًا")، وتحولها إلى خطة تقنية مرسومة ومُكَلَّفة وقابلة للدفاع عنها، ثم تقنع بها المهندسين الذين سينونها والتنفيذيين الذين سيدفعون ثمنها. تلك هي وظيفة **مهندس السحابة المعماري (cloud architect)** (وُعلن أيضًا بمسميات *solutions architect* و *cloud solutions architect* و *cloud domain architect*)، وهي حرفة قابلة للتعليم بمنهج واضح — هو هذا المنهج.

تتكوّن الدورة من خمس مراحل. المرحلة الأولى (الأسابيع 1-6): اللبنة الأساسية، بوصفها قرارات. كل مكوّن سحابي يُعاد تدريسه لا بوصفه معلومة بل بوصفه خيارًا له مفاضلاته — لأن وحدة عمل المعماري هي القرار. المرحلة الثانية (الأسابيع 7-14): مجالات التصميم. الأعمدة الستة لإطار Well-Architected — الموثوقية، والأمن، والأداء، والتكلفة، والعمليات، والاستدامة — يُدرّس كل منها بعمق مع أنماطه القياسية وتصميم عملي محلول. المرحلة الثالثة (الأسابيع 15-20): كتالوج الأنماط. المعماريات الاثنتا عشرة التي تحل 95% من المشكلات الحقيقية، و— الأهم — متى لا تستخدم كلاً منها. المرحلة الرابعة (الأسابيع 21-26): الهجرة والعالم الحقيقي. نقل الشركات القائمة إلى السحابة، والقيود (القانون، والأنظمة الموروثة legacy، والارتهاق للموّرّد lock-in) التي تجعل المعمارية الحقيقية أصعب من معمارية السبورة البيضاء. المرحلة

الخامسة (الأشهر 7-12): الجِرفة. الوثائق والمخططات والمراجعات والعروض والشهادات التي تجعلك قابلاً للتوظيف معمارياً، وتُتَوَجَّ بثلاثة مشاريع تصميم بمستوى الملف المهني.

لمن هذه الدورة. إنها تفترض صفر معرفة سابقة: فكل ما تعتمد عليه يُعاد تدريسه بإيجاز قبل استخدامه. وإن كنت قد أكملت بالفعل **دورة مهندس السحابة (The Cloud Engineer Course)** (أو كنت تعمل في السحابة يدوياً اليوم)، فستكون المرحلة الأولى مألوفة لك — تصفّحها سريعاً، لكن نَقِّذ تمارينها على أي حال، لأنها تعيد تأطير ما تعرفه بوصفه حقائق إلى أشياء يجب أن تزنها بوصفها قرارات، وإعادة التأطير هذه هي التحوُّل الكامل من مهندس إلى معماري.

قواعد تسري على الدورة كلها: ادرس ساعة واحدة يومياً، ستة أيام في الأسبوع — فالاستمرارية تتفوق على الكثافة. كل وحدة تنتهي بـ **جدول مفردات** (الكلمات التي يجب أن تمتلكها)، و**تمارين** (عمل تصميمي لا قراءة — فالمعمارية تُتعلَّم بالرسم واتخاذ القرار)، و**إنجاز مرحلي (Milestone)** (برهان على جاهزيتك للانتقال؛ لا تقفز فوق إنجاز مرحلي لم تحققه). ومنذ الأسبوع الأول، احتفظ بـ **دفتر تصميم (Design Journal)**: كل معمارية ترسمها، وكل مفاضلة تحسمها، وكل توقع تتوقعه عن التكلفة أو الفشل. بعد عشرة أشهر يصبح ملف مقابلاتك؛ ولا شيء يبهر لجنة توظيف مثل دفتر مؤرَّخ يضم أربعين تصميمًا بملاحظات صادقة عما أخطأت فيه.

ما الذي يفعله مهندس السحابة المعماري فعلاً

فكّر في مهندسة معمارية للمباني. إنها لا ترصف الطوب — لكن عليها أن تعرف بدقة ما يستطيعه الطوب وما لا يستطيعه، وكم يكلف، وكيف تنهار المباني. تُنصت إلى عميل ("بيت عائلي، مشمس، ضمن هذه الميزانية، على هذه الأرض الوعرة")، وتحوُّل الرغبات والقيود إلى **رسومات ومواصفات** دقيقة بما يكفي لبنى البناؤون منها ويوقّع العميل عليها. ثم تبقى طوال البناء، تجيب عن الأسئلة وتعُدّل الخطة حين تتبيّن الأرض أطرى مما قال المسح.

استبدل بالطوب خوادم، وتلك هي الوظيفة. يقضي معماري السحابة أسبوعه في مزيج من: **الإنصات** إلى أصحاب المصلحة في العمل واستخراج المتطلبات الحقيقية من الأمانى الغامضة؛ و**التصميم** — اختيار المكونات، ورسم المخططات، وتدوين القرارات وأسبابها؛ و**مراجعة** تصاميم الآخرين على ضوء الأعمدة الستة؛ و**تقدير** ما سيكلفه تصميم ما شهرياً والدفاع عن ذلك الرقم أمام المالية؛ و**تخطيط هجرات الأنظمة** القديمة إلى السحابة؛ و**العرض والتقديم** — التصميم نفسه يُشرح بطريقة للمهندسين وبطريقة مختلفة تماماً للتنفيذيين؛ و**الإرشاد (mentoring)** للمهندسين حتى تصمد المعايير عند اصطدامها بالمواعيد النهائية. وفي فرق كثيرة هناك أيضاً ما قبل البيع (**pre-sales**): الجلوس بجانب مندوب مبيعات أمام عميل محتمل، ورسم الحل الذي يكسب الصفقة.

وما ليس المعماري: ليس أفضل مبرمج في الغرفة (عادة ليس كذلك)، ولا الشخص الذي يكتب أكثر (بل يكتب أقل)، ولا عبقرياً منعزلاً يُملي المخططات من فوق (ذلك أسرع طريق لـ **لُتْجَاهَل**). الناتج الحقيقي للمعماري هو قرارات جيدة، مدوّنة، يبنها الآخرون عن طيب خاطر.

الوصف الوظيفي الذي تتدرَّب من أجله

في أغسطس 2026 جمعنا إعلانات وقوالب توظيف حقيقية لوظيفة Cloud/Solutions Architect — قالب معماري سحابة لدى شركة توظيف (KORE1)، ووصفًا وظيفيًا لدى شركة توظيف أخرى (Corner 4 Resources)، وتعريف Microsoft نفسها لدور المعمار في إطار Azure Well-Architected، وإعلان Solutions Architect لما قبل البيع (Arpio)، للتعافي من الكوارث على (AWS)، وإعلان Solution Architect مؤسسي (Intel، عبر Built In)، وإعلان Cloud Domain Architect مؤسسي (Halliburton، بأولوية Azure). انزع نكهة الشركات وستجد المتطلبات الاثني عشر نفسها تتكرر مرة بعد مرة. هذه الدورة مبنية على تلك القائمة — وإليك بالضبط أين يُدرَّس كل متطلب:

ما تطلبه الأوصاف الوظيفية (بكلماتها، بتصرف)	أين تعلِّمه هذه الدورة
«تصميم معماريات سحابية قابلة للتوسع وأمنة مفصَّلة على متطلبات العمل والمتطلبات التقنية» — تصميم حلول من طرف إلى طرف	المراحل 1-3، والمشاريع الختامية في المرحلة 5
«خبرة منصات عميقة» في AWS و/أو Azure — الحوسبة والتخزين والشبكات و IAM وبنية الحسابات	المرحلة 1 + مسار الشهادات في المرحلة 5
«إدارة مراجعات Well-Architected» على ضوء الأعمدة الستة	المرحلة 2 (الأعمدة)، والمرحلة 5 الوحدة 12 (إدارة المراجعات)
«قيادة مبادرات الهجرة/التحديث — أي الأحمال تنتقل كما هي، وأيها يُعاد تصميمه معماريًا، وأيها يُتقاعَد»	المرحلة 4 الوحدة 10 (الراءات السبع، وتخطيط الموجات)
«تصميم مناطق الهبوط (landing zones)، وبنية الحسابات، والدرايزينات، والأنماط المرجعية التي تنشر الفرق ضمنها»	المرحلة 4 الوحدة 10
«دمج الأمن والامتثال في التصميم بدل تثبيتهما لاحقًا» — انعدام الثقة، والهوية	المرحلة 2 الوحدة 5، والمرحلة 4 الوحدة 11
الإتاحة العالية والتعافي من الكوارث — «فجوات RTO/RPO، وتكاليف التوقف، والتعرض لبرمجيات الفدية»	المرحلة 2 الوحدة 4، والمرحلة 3 الوحدة 9 (تعدد المناطق)
«امتلاك نموذج تكلفة السحابة — الوسم، و showback، والسعة المحجوزة، وضبط الأحجام»؛ وتقدير تكاليف الحلول	المرحلة 2 الوحدة 6، والمرحلة 5 الوحدة 13 (تكليف العروض)

المرحلة 1 الوحدة 2، والمرحلة 3 الوحدة 9، والمرحلة 4
تصميم معمارية الشبكات — ال VPCs، وهيكل hub-and-spoke، والاتصال الهجين

المرحلة 1 الوحدة 3 (المخططات)، والمرحلة 5 الوحدة 12 (ال ADRs وأطقم المخططات)
«إنشاء التوثيق المعماري والمخططات والمعايير»؛
«صيانة سجلات القرارات المعمارية (ADRs)»

المرحلة 5 الوحدة 13
التواصل مع أصحاب المصلحة — «عرض المفاهيم التقنية على جمهور تنفيذي (C-level) وجمهور تقني»؛
«الدفاع عن القرارات المعمارية أمام الأمن والمالية والهندسة»

المرحلة 5 الوحدة 13
«إرشاد مهندسي السحابة والمنصات الذين يبنون وفق معايير»؛ ودعم ما قبل البيع — عروض توضيحية RFPs و POCs

الشهادات، متحقّقًا منها في السوق نفسها: السّلم القياسي هو – AWS Certified Solutions Architect – Associate (SAA-C03) — الشهادة الواحدة الأكثر طلبًا، وتحضيرها عادة 2–3 أشهر — تليها AWS Certified Solutions Architect – Professional (SAP-C02)، وعادة 4–8 أشهر إضافية؛ والشركات الغارقة في Azure تطلب (AZ-305 (Azure Solutions Architect Expert؛ وبعض المؤسسات الكبرى تضيف TOGAF لمنهجية المعمارية المؤسسية. الخطة الكاملة في المرحلة 5، الوحدة 14.

المرحلة الأولى — اللبنة الأساسية، بوصفها قرارات (الأسابيع 1–6)

خريجو دورة مهندس السحابة: تصفّحوا الشروح سريعًا، لكن نَقِّدوا كل تمرين. الحقائق هي نفسها؛ أما الأسئلة فجديدة.

الوحدة 1 (الأسبوعان 1–2): ما السحابة، وكيف يفكّر المعمارون

السحابة في فقرة واحدة (مراجعة من نقطة الصفر). السحابة هي حواسيب يملكها غيرك، تُستأجر بالساعة، وتديرها البرمجيات. تدير AWS و Microsoft Azure و Google Cloud مستودعات من الخوادم (مراكز بيانات — data centers) مجمّعة في مناطق (Regions) جغرافية (مثل “Asia Pacific (Bangkok)”)؛ وكل منطقة تضم عدة مناطق إتاحة (Availability Zones — AZs) معزولة — مبانٍ منفصلة بطاقة مستقلة، قريبة بما يكفي

لاتصالات سريعة، وبعيدة بما يكفي حتى لا يُغرق فيضان أو حريق واحد اثنتين منها. تستأجر شرائح من هذا كله بالثانية: آلات خام (IaaS — تدير كل ما عليها)، أو منصات مُدارة (PaaS — لا تُحضر سوى تطبيقك)، أو برمجيات جاهزة (SaaS — تستخدمها فقط). هذا هو الأساس كله. وكل ما يصمّمه المعماري يُرتَّب فوقه.

والآن حركة المعماري: حوّل كل حقيقة إلى سؤال. المهندس يتعلم أن "منطقة الإتاحة مركز بيانات معزول". أما المعماري فيسأل فورًا: "كم منطقة إتاحة يستحق هذا workload؟" — لأن اثنتين تكلفان أكثر من واحدة، وثلاثًا أكثر من اثنتين، وموقع كتّيب تسويقي لا يستحق ما يستحقه نظام مدفوعات. هذه أول أفكار الدورة وأهمها:

المعمارية هي انضباط جعل المفاضلات صريحة. لا تكاد توجد مكونات خاطئة — بل مكونات خاطئة لهذا workload، وهذه الميزانية، وهذا الفريق، وهذا الموعد النهائي.

مثلث المفاضلة. كل تصميم يفاوض بين ثلاثة أركان: **سريع** (الأداء — سريع للمستخدمين وسريع في البناء)، و**رخيص** (فاتورة شهرية منخفضة وجهد هندسي منخفض)، و**صامد** (ينجو من الأعطال، ويتوسع، ويبقى آمنًا). يمكنك الدفع نحو أي ركنين؛ والثالث يدفع الثمن. نموذج شركة ناشئة الأولي ينبغي أن يكون سريعًا ورخيصًا — فالصمود يمكن أن ينتظر. ودفتر أستاذ بنك يجب أن يكون صامدًا وسريعًا — ولن يكون رخيصًا. حين يقول صاحب مصلحة "نريد الثلاثة كلها"، فوظيفتك أن تتبسم وتساءل أيها يريدون أكثر، لأن التصميم لا يمكن أن يبدأ قبل أن يجيبوا. ارسّم هذا المثلث أعلى كل تصميم تنجزه في هذه الدورة، وعلم موضع ال workload عليه. سيوفر عليك ألف جدال.

المتطلبات: المادة الخام للتصميم. يفصل المعماريون بين المتطلبات الوظيفية (functional requirements) (ما الذي يفعله النظام — "يستطيع العملاء طلب الغداء") والمتطلبات غير الوظيفية (non-functional requirements — NFRs) (كم يجب أن يحسن فعله — "في أقل من اثنتين، لعشرة آلاف طالب في آن واحد، بنسبة 99.9% من الوقت، وضمن قانون PDPA"). المبتدئون يهوسون بالقائمة الأولى؛ والمعماريون يكسبون رواتبهم من الثانية، لأن ال NFRs هي ما يحدد المعمارية فعليًا. سؤالان سحريان يستخرجان ال NFRs من أصحاب مصلحة لا يعرفون أنها لديهم: "ماذا يحدث للعمل لو توقف هذا ساعة؟" و"كيف يبدو النجاح عند عشرة أضعاف حجم اليوم؟"

المفردات:

المصطلح	التعريف
المنطقة / منطقة الإتاحة (AZ)	المنطقة: عنقود جغرافي من مراكز البيانات تختار العمل فيه. منطقة الإتاحة: مركز بيانات معزول (أو مجموعة صغيرة) داخله — وحدة "يمكن لمبنى واحد أن يفشل".

المصطلح	التعريف
IaaS / PaaS / SaaS	استئجار آلات خام / منصة مُدارة / برمجيات جاهزة. المؤشر المنزلق من أقصى التحكم إلى أدنى الصيانة.
Workload (جَمَل التشغيل)	أي تطبيق أو نظام يُعامل بوصفه وحدة تصميم — “workload المدفوعات”.
المتطلب الوظيفي	ما يجب أن يفعله النظام (“يستطيع العملاء الطلب”).
المتطلب غير الوظيفي (NFR)	كم يجب أن يحسن فعله — السرعة، والحجم، والجاهزية، والأمن، والامتثال، والتكلفة. الـ NFRs تقود المعمارية .
المفاضلة (trade-off)	ما تتخلى عنه لتحصل على ما اخترت. لكل قرار تصميمي واحدة؛ ووظيفة المعماري تسميتها بصوت مسموع.
القيود (constraint)	حد غير قابل للتفاوض: الميزانية، والموعد النهائي، والقانون، والأنظمة القائمة، ومهارات الفريق. القيود ليست عقبات أمام التصميم — إنها هي موجز التصميم.
صاحب المصلحة (stakeholder)	كل من له مصلحة في النظام: المستخدمون، والمهندسون، والمالية، والأمن، والتنفيذيون، والمنظمون. أصحاب مصلحة مختلفون، لغات مختلفة — وأنت تتحدثها كلها.
Greenfield / brownfield	نظام جديد كليًا بلا تاريخ (greenfield) مقابل نظام متشابك مع أنظمة قائمة (brownfield) — وهو معظم العمل الحقيقي).
الخدمة المُدارة	مكوّن يشغله المزود نيابة عنك (النسخ الاحتياطي والترقيع والـ failover مشمولة). الخيار الافتراضي للمعماري، ما لم يوجد سبب مكتوب لغير ذلك.

فيديوهات هذه الوحدة (الروابط متحقّق منها):

الفيديو	القناة	المدة	الرابط
Top 50+ AWS Services Explained in 10 Minutes	Fireship	~10 min	https://www.youtube.com/watch?v=JlbiYCM48to

شاهد جولة Fireship مرتين: مرة الآن من أجل الخريطة، ومرة في نهاية المرحلة الأولى — وستفاجأ بعدد الخدمات التي صرت قادرًا على وضعها في تصميم.

التمارين: (1) اختر ثلاثة تطبيقات تستخدمها يوميًا (تطبيق بنك، وتطبيق توصيل طعام، وتطبيق فيديو)، واكتب لكل منها أهم ثلاثة NFRS وعلم موضعها على مثلث المفاضلة. (2) حاور صديقًا عن فكرة عمل لعشر دقائق واستخرج خمسة متطلبات وظيفية وخمسة غير وظيفية — ولاحظ أن الـ NFRS لا تخرج إلا حين تطرح السؤالين السحريين. (3) ابدأ دفتر تصميمك بالمدخل رقم 1: المثلث، وفقرة واحدة عن لماذا "أي ركن نصحي به؟" سؤال عمل لا سؤال تقني.

الإنجاز المرحلي: إذا أعطيت أي وصف نظام من جملة واحدة، تستطيع إنتاج NFRS المرحّجة له وموضعه على المثلث في خمس دقائق، بصوت مسموع، دون ملاحظات.

الوحدة 2 (الأسبوعان 3-4): الحوسبة والتخزين وقاعدة البيانات والشبكة — القرارات الأربعة

كل تصميم سترسمه يومًا هو هذه الخيارات الأربعة، متخذة عن عمد. وإليك كل واحد منها مُدرّسًا بوصفه قرارًا.

القرار 1 — الحوسبة: VM أم حاوية أم serverless؟ الآلة الافتراضية (virtual machine — VM) شريحة مستأجرة من خادم تتصرف كحاسوب كامل — أقصى تحكم، وأقصى صيانة (أنت ترقّعها، وأنت توسّعها، وتدفع بينما هي خاملة). **الحاوية (container)** تغلف تطبيقًا واحدًا مع كل ما يحتاجه فيعمل بشكل متطابق في أي مكان؛ والمنسق (Kubernetes) يشغل أساطيل منها ويعالجها — كثافة ونقل ممتازان، لكنك تبني منصة معقدة تحتاج إلى أشخاص مهرة. **Serverless** (الحوسبة دون خوادم — AWS Lambda و Azure Functions) تشغل شيفرتك فقط عند استدعائها وتفوتر لكل تنفيذ — صيانة شبه معدومة ومثالية للحركة المتقطعة، لكن بحدود (سقوف زمن التنفيذ، والانطلاقات الباردة cold starts، وزواج أعمق بمزوّد واحد). جدول القرار الذي يجب أن تستطيع إعادة إنتاجه من الذاكرة:

احذر من	متى	اختر
أنت تملك الترقية والتوسيع والثالثة فجراً؛ ووقت الخمول يفوتر كاملاً	برمجيات موروثية، وتراخيص خاصة، وحاجة إلى تحكم كامل في نظام التشغيل، وحمل ثابت متوقع	VM
تعقيد المنصة — K8s وظيفة بدوام كامل؛ ومبالغة للفرق الصغيرة	خدمات كثيرة، وفريق يملك المهارات أصلاً، وأهمية لقابلية النقل	الحاويات + Kubernetes
حدود زمن التشغيل، والانطلاقات الباردة، وصعوبة التنبؤ بالتكلفة عند	حركة متقطعة أو غير متوقعة، ولحام مدفوع بالأحداث، وفرق	Serverless

صغيرة، وسرعة وصول إلى السوق الحجم الثابت الهائل، والارتهاق للمورّد

الرؤية العميقة: هذا خيار لكل workload على حدة، لا دين للشركة. المنشآت الحقيقية تشغل الثلاثة جنبًا إلى جنب، وبحق.

القرار 2 — التخزين: كائني أم كتلي أم ملفات — وكم هو ساخن؟ التخزين الكائني (object storage) (Amazon S3) دلو بلا قاع للملفات — رخيص، ومتين بشكل مذهل ("أحد عشر تسعة")، والجواب الافتراضي على "أين تذهب الملفات؟" **التخزين الكتلي (block storage) (EBS)** هو القرص الافتراضي المثبت على آلة افتراضية. **تخزين الملفات (file storage) (EFS)** قرص مشترك تركّبه آلات كثيرة في آن واحد. والبعد الإضافي عند المعماري هو درجة الحرارة: بيانات ساخنة (تُقرأ باستمرار، مسجّرة للسرعة) مقابل فئات باردة/أرشفية (Glacier — بقروش، لكن الاسترجاع بدقائق إلى ساعات). تصميم قواعد دورة حياة تُزحج البيانات القديمة نحو الفئات الباردة هو أرخص مكسب تكلفة في السحابة؛ ونسيانه أكثر الأخطاء شيوعًا.

القرار 3 — قاعدة البيانات: SQL أم NoSQL (وأي نكهة مُدارة)؟ قواعد البيانات العلائقية/SQL (PostgreSQL وMySQL؛ مُدارة بوصفها RDS/Aurora) تحفظ البيانات في جداول صارمة باتساق مضمون — الخيار الافتراضي لكل ما تكون فيه الصحة مقدّسة: المال، والطلبات، والمخزون، والمستخدمون. وقواعد **NoSQL** (DynamoDB وMongoDB) تقايض البنية الصارمة بالمرونة والتوسع الأفقي شبه اللامحدود — الخيار الافتراضي للجلسات والكتالوجات والخلاصات والقياسات. قاعدة القرار: *ابدأ بـ SQL ما لم تستطع تسمية السبب المحدد الذي يمنع نجاحها* (حجم متطرف، أو مخطط مرّن، أو قراءات عالمية بميلي ثوانٍ أحادية الرقم). وفي السحابة، ينبغي أن تعني "قاعدة بيانات" في الغالب الأعم "قاعدة بيانات مُدارة" — فالمزوّد يتولى النسخ الاحتياطي والترقيع وال failover؛ وأي فريق يشغل قاعدته بنفسه على آلات افتراضية يجب أن يملك سببًا مكتوبًا. وأصف المتخصصين إلى مفرداتك: **الكاش (cache) (Redis)** — بيانات ساخنة في الذاكرة، قراءات بالميكروثانية، **والمستودع التحليلي (warehouse)** (تحليلات على نطاق واسع — المرحلة 3)، **والطابور (queue)** (ليس قاعدة بيانات، لكنه كثيرًا ما يكون القطعة الناقصة — المرحلة 3).

القرار 4 — الشبكة: شكل العالم الخاص. الـ **VPC (Virtual Private Cloud)** قسمك المسيّج من شبكة المزوّد. بداخله تحمل الـ **subnets العامة** ما يجوز للإنترنت بلوغه (مورّعات الحمل)، وتحمل الـ **subnets الخاصة** كل ما عداه — خوادم التطبيقات، ودائمًا، قواعد البيانات. "قاعدة البيانات تقع في subnet خاصة" هي الجملة الأكثر تكرارًا في مراجعات المعمارية؛ ومنطقها — لا شيء يستطيع مهاجمة ما لا مسار إليه من الإنترنت — نصف أمن الشبكات. وحول الـ VPC: **مورّع الحمل (load balancer)** ينشر الحركة على الخوادم ويلتف حول المريضة منها؛ و**DNS (Route 53)** يحوّل الأسماء إلى عناوين؛ و**CDN (CloudFront)** يخزّن المحتوى في مئات المدن فيكون سريعًا في كل مكان؛ و**API gateway** هي مكتب الاستقبال المُدار لواجهاتك البرمجية (مصادقة، وحدود معدل، وتسجيل). ولوصل العالم القديم: **VPN** (نفق مشفّر عبر الإنترنت) أو **Direct Connect** (خط مادي خاص) — الحبلان السريان لكل تصميم هجين في المرحلة 4.

المصطلح	التعريف
Instance / نوع ال instance	آلة افتراضية مستأجرة واحدة / حجمها (المعالج + الذاكرة)، وهو ما يحدد سعرها بالساعة.
الحاوية / Docker / Kubernetes (K8s)	حزمة قابلة للنقل لتطبيق واحد / الأداة التي تبنيتها وتشغلها / المنسق الذي يشغل أساطيل منها ويعالجها.
Serverless / Lambda	شيفرة تعمل فقط عند استدعائها، وتفوتر لكل تشغيل، بلا خوادم تُدار. وLambda نسخة AWS منها.
الانطلاقة الباردة (cold start)	التأخير الإضافي حين تعمل دالة serverless بعد خمول — المفاضلة الكلاسيكية لل serverless.
bucket / S3 / المتانة	تخزين AWS الكائني / حاوية ملفات مسماة واحدة / احتمال نجاة البيانات — نسبة S3 البالغة 99.999999999% تعني أن الفقد شبه مستحيل.
فئة التخزين / سياسة دورة الحياة	درجة سعر-سرعة (ساخن ثم بارد ثم أرشيف) / القاعدة التلقائية التي تنقل البيانات المتقدمة إلى فئات أرخص.
RDS / Aurora / DynamoDB	خدمات SQL المُدارة لدى AWS / نسختها السحابية عالية الأداء / قاعدة NoSQL المُدارة الرائدة لديها.
الاتساق (consistency)	ضمان أن كل من يقرأ البيانات يرى الحقيقة نفسها في اللحظة نفسها — قوة SQL الخارقة، وما ترخيه NoSQL لتكسب التوسع.
الكاش / Redis	نسخة بسرعة الذاكرة من البيانات الساخنة توضع أمام قاعدة البيانات / الأداة القياسية لذلك.
VPC / subnet (عامة، خاصة)	قسمك الشبكي الخاص / تقسيماته — العامة تواجه الإنترنت، والخاصة لا. قواعد البيانات تعيش خاصة. دائمًا.
موزّع الحمل / فحص السلامة	موجّه الحركة عبر الخوادم / اختبار النبض الذي يستخدمه ليتوقف عن الإرسال إلى الميته منها.
CDN / الحافة (edge)	نسخ من محتواك في مدن العالم / "الحافة" = القرب من المستخدمين.
API / API gateway	الباب المضبوط الذي تعرضه قطعة برمجية على أخرى / مكتب الاستقبال المُدار لتلك الأبواب.

المصطلح	التعريف
VPN / Direct Connect	نفق مشفّر عبر الإنترنت / خط مادي خاص إلى السحابة — الطريقتان اللتان يلتقي بهما ال on-prem بالسحابة.

فيديوهات هذه الوحدة (الروابط متحقّق منها):

الفيديو	القناة	المدة	الرابط
Kubernetes explained in 15 mins	TechWorld with Nana	min 16~	https://www.youtube.com/watch?v=VnvRFRk_51k
Serverless Computing in 100 Seconds	Fireship	min 2~	https://www.youtube.com/watch?v=W_VV2Fx32_Y
AWS Networking Basics (VPC & Subnets)	KodeKloud	min 30~	https://www.youtube.com/watch?v=QM63dyA_4Pc

التمارين: (1) لكل واحد من هذه الأحمال الخمسة، اختر الحوسبة وقاعدة البيانات والتخزين، واكتب جملة تسويق واحدة لكل خيار: موقع طلب غداء مدرسي؛ ودفتر معاملات بنك؛ وتطبيق مشاركة صور؛ ومولّد تقارير ليلي يعمل 20 دقيقة؛ وتطبيق دردشة لخمسة ملايين مستخدم. (2) خذ أحد خياراتك ودافع عن الخيار *المعاكس* بأقصى إقناع تستطيعه — فالمعماري العاجز عن الدفاع الأمين عن البديل لم يفهم المفاضلة. (3) *الدفتر*: جدول قرار الحوسبة، من الذاكرة.

الإنجاز المرحلي: إذا أعطيت أي workload في جملة واحدة، تستطيع تسمية قراراته الأربعة بتسوياتها في أقل من ثلاث دقائق — ولقرار واحد منها على الأقل، تسمية ما الذي كان سيغيّر رأيك.

الوحدة 3 (الأسبوعان 5-6): قراءة مخططات المعمارية ورسمها

المعماري الذي لا يستطيع الرسم مستشار لا يجيد سوى الكلام. المخططات هي لغة عملك: هذه الوحدة تجعلك طليقًا في قراءتها وكفؤًا في رسمها.

المخطط القانوني — تعلّم هذا أولًا. المعمارية ثلاثية الطبقات (three-tier architecture) هي “بنية الجملة” في مخططات السحابة؛ ومعظم التصميمات تنويعات عليها:

Users → DNS → CDN → Load Balancer → [Web/App servers, ×N, multi-AZ, auto-scaled]

[Cache] →

Database: primary +]

[standby, private subnets

الطبقة 1 (العرض presentation): ما يلمسه المستخدمون — المحتوى الساكن من الـ CDN، والطلبات عبر موزّع الحمل. الطبقة 2 (التطبيق application): أسطول الخوادم المتبادلة القابلة للاستبدال الذي يشغل منطقك، في subnets خاصة، موزّع تلقائيًا. الطبقة 3 (البيانات data): قاعدة البيانات، الأعماق والأشد حماية، مع احتياطي في منطقة إتاحة ثانية. تتدفق الحركة باتجاه واحد: المستخدمون لا يلمسون طبقة التطبيق مباشرة أبدًا، وطبقة التطبيق وحدها تخاطب طبقة البيانات. تدرب على رسم هذا حتى ترسمه يدك دون دماغك — فهو افتتاحية مقابلة السبورة البيضاء في كل مكان على الأرض.

اصطلاحات الترميز التي تجعلك تبدو محترفًا (لأنها تجعلك تفكر كمحترف): الصناديق مكونات — سمّ كلًّا منها بما هو وبأي خدمة ("App servers — EC2, auto-scaling group"). الأسهم تظهر اتجاه تدفق الطلب، مسماة بالبروتوكول حين يهتم. الأطر المتقطعة تظهر الحدود — الـ VPC، وكل subnet، وكل منطقة إتاحة (ارسم حدود مناطق الإتاحة فيصبح الـ multi-AZ مرئيًا بدل أن يكون ادعاءً). والمستخدم/الفاعل يقف خارج النظام. والأرقام على الأسهم (1، 2، 3...) تتيح لك سرد رحلة طلب واحد. وكل مخطط يحمل عنوائًا وتاريخًا ومفتاح رموز. والقاعدة الأعماق: مخطط واحد، جمهور واحد، سؤال واحد. المخطط الذي يعرض كل شيء لا يعرض شيئًا! وستتعلم الطقم القياسي لمستويات التقريب (السياق ثم الحاويات ثم النشر) في المرحلة 5.

قراءة مخططات الآخرين — أشعة المعماري السينية. حين يُسلّم إليك مخطط، أجر هذا المسح بصوت مسموع: أين يلمس الإنترنت هذا النظام (كل نقطة لمس سطح هجوم)؟ أين البيانات، وهل هي في subnet خاصة؟ ما المكرر (صامد) وما نقطة الفشل الوحيدة — صندوق بلا توأم؟ أين سيؤول الأمر عند عشرة أضعاف الحركة؟ كم يكلف كل صندوق شهريًا؟ خمسة أسئلة، ثلاثون ثانية، وقد قرأت المخطط كما يقرأ الطبيب صورة أشعة. تصفّح مركز معماريات AWS (<https://aws.amazon.com/architecture/>) وأجر المسح على ثلاث معماريات مرجعية منشورة.

المفردات:

المصطلح	التعريف
المعمارية ثلاثية الطبقات	الفصل الكلاسيكي: العرض (مدخل المستخدمين) ثم التطبيق (المنطق) ثم البيانات (قاعدة البيانات). الشكل الافتراضي لأنظمة الويب.
الطبقة (tier / layer)	شريحة أفقية من النظام بمسؤولية واحدة، لا تخاطب سوى جيرانها.

المصطلح	التعريف
نقطة الفشل الوحيدة (SPOF)	أي مكوّن يُسقط موته المنفرد النظام كله. أول ما تصطاده في أي مخطط.
Multi-AZ	تشغيل نسخ مكررة عبر منطقتي إتاحة على الأقل حتى يفشل مركز بيانات كامل دون أن يظهر ذلك.
التوسع التلقائي (auto-scaling group)	آلات تُضاف وتُزال تلقائيًا مع الطلب — سعة تتنفس.
Stateless / stateful	خادم لا يحمل بيانات فريدة (أي توأم يحل محله، فيتوسع بحرية) مقابل خادم يحمل بيانات يجب ألا تضع. هدف التصميم: طبقة تطبيق stateless، والحالة مدفوعة نزولاً إلى قاعدة البيانات والكاش.
المعمارية المرجعية	تصميم مثالي منشور ومبارك من المزود لمشكلة شائعة — المعماربيون يركّبون من هذه قبل أن يخترعوا.
مخطط السياق (context diagram)	أعلى مستوى تقريب: نظامك صندوقًا واحدًا، مع المستخدمين والأنظمة الخارجية التي يلمسها.
سطح الهجوم (attack surface)	كل نقطة يستطيع العالم الخارجي لمس النظام عبرها. الأصغر أسلم.
حركة الشمال-الجنوب / الشرق-الغرب	الحركة الداخلة إلى النظام والخارجة منه مقابل الحركة بين المكونات في داخله.

التمارين: (1) ارسم المخطط ثلاثي الطبقات من الذاكرة، خمسة أيام متتالية، حتى يستغرق أقل من أربع دقائق بكل الحدود (VPC، والـ subnets، ومنطقتا إتاحة) مرسومة. (2) خذ أحمال الوحدة 2 الخمسة وارسم كل واحد منها — عشرون دقيقة لكل مخطط، مع فرض قواعد الترميز. (3) اعثر على أي مخطط معمارية حقيقي على الإنترنت (مركز معماريات AWS يضم المئات)، واكتب مسحه بالأسئلة الخمسة في دفترك. (4) تمرين السرد: المخطط أمامك، اسرد نقرة مستخدم واحد من المتصفح إلى قاعدة البيانات وعودة، بصوت مسموع، مرقّمًا الأسهم وأنت تمضي.

الإنجاز المرحلي — نهاية المرحلة الأولى: على سبورة بيضاء (أو ورقة، مصوّرة لدفترك)، تستطيع رسم تصميم ثلاثي الطبقات صحيح وسليم الترميز لـ workload جديد من جملة واحدة في أقل من خمس عشرة دقيقة، وسرد طلب عبره، والإجابة عن "ما الذي يفشل لو مات هذا الصندوق؟" لكل صندوق. هذا الرسم مع الاستجواب هو بالضبط النصف الأول من مقابلة معماري حقيقية — ومن هنا فصاعدًا، كل شيء عمق.

المرحلة الثانية — مجالات التصميم (الأسابيع 7-14)

خريطة هذه المرحلة هي إطار AWS Well-Architected — قائمة الصناعة المشتركة لما يعنيه “مصمم كما ينبغي”، منظّمة في ستة أعمدة: التميز التشغيلي، والأمن، والموثوقية، وكفاءة الأداء، وتحسين التكلفة، والاستدامة. (لدى Azure إطار شبه مطابق؛ تعلّم أحدهما بعمق تكن قد تعلمت كليهما.) الأوصاف الوظيفية تطلب بالاسم معماريين يستطيعون “إدارة مراجعات Well-Architected”، لذا نتناول الأعمدة عمودًا عمودًا، وتتعلم لكل منها ثلاثة أشياء: أسئلته المفتاحية، وأنماطه القياسية (الإجابات المملة المجربة — فالمعماريون يركّبون قبل أن يبتدعوا)، ومثال محلول على سيناريو مستمر.

السيناريو المستمر للمرحلة الثانية كلها: ThaiTicket، منصة تذاكر فعاليات خيالية في بانكوك. الحمل العادي: 2,000 زائر في الساعة. لكن حين تُطرح تذاكر فنان شهير في العاشرة صباحًا، تستقبل 400,000 زائر في عشر دقائق، ويجب ألا تتبع المدفوعات المقعد الواحد مرتين، وتخضع البيانات الشخصية للعملاء التايلانديين لقانون PDPA. سريعة ورخيصة وصامدة — ThaiTicket تحتاج إلى الثلاثة ولا يمكنها الحصول عليها، وهذا ما يجعلها المريض التدريبي المثالي.

شاهد قبل الوحدة 4 (روابط متحقّق منها):

الفيديو	القناة	المدة	الرابط
The Five Pillars of the AWS Well-Architected Framework	Amazon Web Services (official)؛ أضيف لاحقًا عمود سادس هو الاستدامة	~4 min	https://www.youtube.com/watch?v=KvEDbPmha6o
What is the AWS Well-Architected Framework?	Tech With Lucy	~10 min	https://www.youtube.com/watch?v=MpDJ6TCWKjk

وضع الإطار نفسه في مفضلتك — <https://aws.amazon.com/architecture/well-architected/> والوثيقة الكاملة على <https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html> — فستعيش فيه ثمانية أسابيع.

الوحدة 4 (الأسبوعان 7-8): الموثوقية — التصميم ليوم تنكسر فيه الأشياء

عقيدة هذا العمود: كل شيء يفشل، طوال الوقت. الأقراص تموت، ومناطق الإناحة تغرق، والنشرات تسوء، وثمة شهادة في مكان ما توشك دائمًا على الانتهاء. الموثوقية ليست غياب الفشل — بل /نعدم/ أهميته: التصميم بحيث حين (لا إذا) يموت مكوّن، لا يلاحظ المستخدمون شيئًا أبدًا.

الأسئلة المفتاحية (اطرحها على كل تصميم، إلى الأبد): ماذا يحدث حين يفشل كل مكوّن — هل هناك دائمًا “وماذا بعد”؟ كيف يتعامل النظام مع عشرة أضعاف الحمل؟ كيف نعرف أن شيئًا فشل قبل أن يخبرنا العملاء؟ ما قيمتا RTO و RPO لدينا — ومن في الجانب التجاري صادق عليهما؟ متى / اختبرنا التعافي آخر مرة؟

RTO و RPO — الرقمان اللذان هما محادثة الكوارث. هدف زمن التعافي (Recovery Time Objective): كم يجوز لنا البقاء متوقفين؟ وهدف نقطة التعافي (Recovery Point Objective): كم من البيانات يجوز لنا خسارته (أي كم يجوز أن يكون عمر آخر نسخة احتياطية سليمة)؟ RTO أربع ساعات و RPO ساعة واحدة يعنيان: عائدون خلال أربع ساعات، وقد خسرنا على الأكثر ساعة من البيانات. هذان قراران تجاريان بأثمان أُسّية — ف RPO من 24 ساعة نسخة احتياطية ليلية (رخيصة)؛ و RPO يقارب الصفر نسخ متواصل (مكلف)؛ و RTO بالدقائق يعني بنية تحتية دافئة خاملة على أهبة الاستعداد (مكلفة جدًا). وظيفة المعمارى أن يجعل الجانب التجارى يختار الرقمين عن علم — وأن يصمّم على مقاسهما بالضبط، لا رومانسيًا بما يتجاوزهما.

الأنماط القياسية: التكرارية (redundancy) (نسختان من كل ما يهم — $N+1$) · ال multi-AZ (نسخ مكررة عبر مراكز البيانات؛ وموّرّع الحمل و failover قاعدة البيانات المُدارة يجعلانها تلقائية) · التوسع التلقائي (السعة تتبع الطلب) · فحوص السلامة + المعالجة الذاتية (ال instances الميته تُكتشف وتُستبدل بالآلات لا بالبشر) · نسخ احتياطية، مختبّرة (النسخة غير المختبّرة أمنية لا خطة — جدول تدريبات استعادة) · التدهور الرشيق (graceful degradation) (عند الإنقال، تخلّ عن الميزات الأقل أهمية أولًا: تستطيع ThaiTicket إسقاط معايير خريطة المقاعد والإبقاء على الدفع) · الطوابير كمتصات صدمات (المرحلة 3) · تجنب الفشل المتسلسل (مُهل timeouts، وإعادة محاولات بتراجع backoff، وقواطع دارة — حتى لا تُغرق تبعية بطيئة واحدة الأسطول كله).

مثال محلول — تصميم موثوقية ThaiTicket. منطقتا إتاحة في منطقة بانكوك. طبقة تطبيق stateless في مجموعة توسع تلقائي خلف موّرّع حمل، تُسجّن مسبقًا بجدول قبل مواعيد الطرح المعلنة (التوسع التلقائي يستجيب في دقائق؛ وطرح العاشرة يحتاج السعة في التاسعة و45 دقيقة). قاعدة Aurora، الرئيسية في AZ-a واحتياطي متزامن في AZ-b، و failover تلقائي في أقل من دقيقة تقريبًا. وطابور بين نقرات “اشتر” ومعالجة الدفع حتى يصفّ تباطؤ مزوّد الدفع للطلبات بدل أن يُسقط الموقع. النسخ الاحتياطي: متواصل، واستعادة إلى أي نقطة زمنية، وتدريب استعادة شهري. الأرقام المتفق عليها مع الجانب التجارى: RTO خمس عشرة دقيقة، و RPO يقارب الصفر للطلبات (فهى مال)، و RPO أربع وعشرون ساعة لبيانات التحليلات (فهى ليست كذلك). تمرين دفتر: ماذا كلّفنا هذا من ركن “الرخص” في المثلث؟

المفردات:

المصطلح	التعريف
RTO / RPO	هدف زمن التعافي: كم يجوز لك البقاء متوقفًا. هدف نقطة التعافي: كم من البيانات يجوز لك خسارته.
	الرقمان اللذان يعرّفان كل محادثة تعافٍ من الكوارث — وهما قراران تجاريان.

المصطلح	التعريف
الإتاحة العالية (HA)	التصميم بحيث لا تسبب الأعطال الروتينية (instance)، أو منطقة إتاحة) انقطاعًا يراه المستخدم.
التعافي من الكوارث (DR)	خطة الأعطال /الكبرى — منطقة كاملة، أو حدث برمجيات فدية — بأهداف RTO/RPO وrunbook ومختبر.
التكرارية / N+1	سعة احتياطية تسمح بموت أي مكّون واحد: N مطلوبة، وN+1 عاملة.
Failover	التحويل التلقائي إلى احتياطي حين تموت الرئيسية.
فحص السلامة (health check)	النبض الآلي الذي يكشف المكونات الميتة حتى تلتف الآلات حولها.
التدهور الرشيق	التخلي عن الميزات الأقل أهمية تحت الضغط حتى ينجو الجوهر.
المهلة / إعادة المحاولة بتراجع	التخلي عن نداء بطيء بعد حد / إعادة المحاولة بفواصل متزايدة — آداب تمنع الفشل المتسلسل.
قاطع الدارة (circuit breaker)	مكّون يتوقف عن نداء تبعية فاشلة فترةً حتى تتعافى — كقاطع الكهرباء في البيت.
هندسة الفوضى (chaos engineering)	حقن الفشل عمدًا (على طريقة Netflix) لإثبات ادعاءات الصمود قبل أن يختبرها الواقع نيابة عنك.

التمارين: (1) خذ تصميم طلب الغداء من المرحلة الأولى ورقّه لينجو من: موت instance، وفقدان منطقة إتاحة، وفشل قاعدة بيانات، وازدحام غداء بعشرة أضعاف — ارسم قبل وبعد. (2) اكتب محادثة RTO/RPO لثلاثة أعمال (مدونة، وموقع تجارة إلكترونية، ونظام سجلات مستشفى) بوصفها حوارًا قصيرًا بين المعماري والمالك — وتدرب على جعل التكلفة مرئية في الحوار. (3) الدفتر: احصر كل نقاط الفشل الوحيدة في تصميم ThaiTicket أعلاه. هناك واحدة على الأقل متروكة عمدًا. (تلميح: كم عدد المناطق؟)

الإنجاز المرحلي: تستطيع إدارة استجواب “ماذا يحدث حين يفشل هذا؟” فوق أي مخطط لعشر دقائق متواصلة دون أن تنضب، وتستطيع شرح RTO مقابل RPO لمالك غير تقني بتشبيه فيضان يجتاح متجرًا في تسعين ثانية.

الوحدة 5 (الأسبوعان 9-10): الأمن — تصميم أنظمة يصعب إizardؤها

الإطار: نموذج المسؤولية المشتركة (shared responsibility model). يؤمّن المزود السحابة ذاتها — المباني، والعتاد، والـ hypervisor. وتؤمّن أنت كل ما تضعه فيها — البيانات، والهويات، والإعدادات، والشفرة. يكاد كل اختراق سحابي شهير يكون سوء إعداد من جهة العميل (bucket عام، أو مفتاح مسرّب، أو دور مفرط الصلاحيات)، ولهذا فالأمن مشكلة معمارية قبل أن يكون مشكلة أدوات: الأوصاف الوظيفية تقول “ادمج الأمن في التصميم بدل تثبيته لاحقًا”، وهذه الوحدة هي الكيفية.

الأسئلة المفتاحية: من وما الذي يستطيع الوصول إلى كل مكوّن، وهل كل صلاحية هي الحد الأدنى المطلوب؟ أين تُشفّر البيانات — أثناء التخزين وأثناء النقل، ومن يمسك المفاتيح؟ أين حدود الشبكة، وما الذي يعبرها؟ كيف سنكتشف اختراقًا — وهل نستطيع رواية القصة بعده من السجلات؟ أي قانون ينطبق على هذه البيانات، وأين تقيم البيانات فيزيائيًا؟

الأنماط القياسية: الحد الأدنى من الامتيازات (least privilege) (كل إنسان وبرنامج ينال أدنى وصول تتطلبه وظيفته — القاعدة الذهبية التي تُحكم أمامها كل سياسة IAM) · MFA في كل مكان، و root مقفول بعيدًا · التشفير أثناء التخزين والنقل، مفعّل دائمًا (إنه خانة اختيار في السحابة؛ ولا عذر) · تجزئة الشبكة (subnets) عامة/خاصة، و security groups جدرانًا نارية لكل خادم؛ ونطاق انفجار الاختراق تحدده الجدران التي رسمتها سلفًا) · الأسرار في خزنة، لا في الشيفرة أبدًا · انعدام الثقة (zero trust) (تحقق من كل طلب صراحة — الهوية، والجهاز، والسياق — ولا تثق بشيء لمجرد كونه “داخل الشبكة”؛ الأوصاف الوظيفية تسميه، فسّمه أنت أيضًا) · الدفاع في العمق (طبقات، حتى لا يكون فشل ضابط واحد نهاية اللعبة) · سجلات التدقيق (سجلات على طراز CloudTrail لمن فعل ماذا — غير قابلة للتعديل، ومراقبة) · الأمن درابزينات لا بوابات (رمز القواعد سياسة مؤتمنة تجعل المسار الآمن هو المسار السهل، بدل اجتماع مراجعة يجعل الأمن عدو التسليم).

مثال محلول — تصميم أمن ThaiTicket. بيانات العملاء (أسماء وبريد ومراجع دفع) مصنّفة بيانات شخصية بموجب PDPA، فُتخّزن مشفّرة في Aurora في منطقة بانكوك (إقامة البيانات)، والمفاتيح في KMS. الشبكة: موّزع الحمل وحده عام؛ وطبقة التطبيق خاصة؛ و subnet قاعدة البيانات لا تقبل الاتصالات إلا من security group طبقة التطبيق. البشر: SSO + MFA؛ والمهندسون ينالون افتراضيًا وصول قراءة فقط إلى الإنتاج، ووصولًا مرتفعًا محدود الوقت عند الطلب (حد أدنى من الامتيازات مع أثر تدقيق). ومعالجة بطاقات الدفع مفوّضة إلى مزود دفع معتمد حتى لا تلمس أرقام البطاقات الخام أنظمتنا أبدًا — قرار تصميمي يزيل عبء امتثال كاملاً، وذلك هو الأمن المعماري في أبهى صوره. CloudTrail مفعّل، وتنبيهات على الوصول غير المعتاد. وسؤال الاختراق مُجاب سلفًا: لدى PDPA واجب إبلاغ خلال 72 ساعة — ويجب أن تتيح لنا السجلات والـ runbook رواية القصة في أقل من ذلك.

المفردات:

المصطلح	التعريف
نموذج المسؤولية المشتركة	يؤمّن المزود السحابة؛ وتؤمّن أنت ما فيها. معظم الاختراقات على جانب العميل من الخط.

المصطلح	التعريف
IAM / الدور / السياسة	إدارة الهوية والوصول — من يجوز له فعل ماذا. السياسة قائمة صلاحيات؛ والدور حزمة منها قابلة للارتداء.
الحد الأدنى من الامتيازات	القاعدة الذهبية: أدنى وصول مطلوب، لا أكثر، ويُراجع دوريًا.
MFA	برهان ثانٍ على الهوية فوق كلمة المرور. غير قابل للتفاوض للبشر.
التشفير أثناء التخزين / النقل / KMS	بيانات مبعثرة على القرص / على السلك / الخدمة المُدارة الماسكة بالمفاتيح.
Security group	قائمة قواعد جدار ناري لكل خادم — “حركة الويب تدخل من مؤرّع الحمل فقط.”
تجزئة الشبكة / نطاق الانفجار	تقسيم الشبكة إلى مناطق مسوّرة / إلى أي مدى يصل المهاجم بعد اختراق واحد. الجدران المرسومة سلفًا تحدده.
انعدام الثقة (zero trust)	تحقق من كل طلب صراحة؛ ولا تثق بشيء لكونه في “الداخل”. الوضعية الافتراضية الحديثة.
الدفاع في العمق	ضوابط متعددة متراكبة حتى لا يكون فشل واحد قاتلاً.
إدارة الأسرار	كلمات المرور والمفاتيح والرموز تعيش في خدمة خزنة — لا في الشيفرة أبدًا، ولا في جدول بيانات أبدًا.
سجل التدقيق / CloudTrail	السجل غير القابل للتعديل لمن فعل ماذا ومتى — كيف تكتشف المتاعب وتعيد بناءها بعد وقوعها.
تصنيف البيانات	وسم البيانات بحسب حساسيتها (عامة / داخلية / شخصية / خاضعة لتنظيم) حتى تطابق الضوابط الوسم لا التخمين.
إقامة البيانات (data residency)	إبقاء البيانات فيزيائيًا داخل حدود بلد ما — متطلب قانوني في صناعات كثيرة، ومُدخل معماري دائمًا.
PDPA / GDPR	قانونا البيانات الشخصية في تايلاند وأوروبا: الموافقة، والإبلاغ عن الاختراق (72 ساعة)، وقواعد النقل عبر الحدود. المرحلة 4 تتعمق أكثر.

التمارين: (1) ارسم مخطط ThaiTicket وضع فوقه طبقة أمنه: علّم كل حد ثقة، وكل نقطة تشفير، وكل مكان تعيش فيه بيانات اعتماد. عادة الطبقة هذه — المخطط نفسه بعدسة الأمن — هي بالضبط ما تطلبه منك المراجعة. (2) اقرأ تقرير post-mortem علنيًا واحدًا لاخترق ناتج عن سوء إعداد سحابي (حادثة Capital One عام 2019 هي الكلاسيكية التعليمية) واكتب في دفترك أي نمط مما سبق كان سيمنعها. (3) لعب أدوار مع ذكاء اصطناعي: يلعب دور مؤسس شركة ناشئة يقول “سنضيف الأمن لاحقًا” — أفضعه بالعدول بحساب تكلفة الاختراق، بلطف.

الإنجاز المرحلي: تستطيع أخذ أي من مخططات مرحلتك الأولى وإنتاج طبقة أمنه في عشرين دقيقة، وشرح الحد الأدنى من الامتيازات وانعدام الثقة ونموذج المسؤولية المشتركة لصاحب مصلحة غير تقني دون مصطلحات متخصصة.

الوحدة 6 (الأسبوعان 11-12): الأداء والتكلفة — السرعة والمال، العمودان المتلازمان

يُدْرَس هذان العمودان معًا لأنهما الرافعة نفسها مدفوعة في اتجاهين متعاكسين، والمعماري هو الشخص الذي يده على الرافعة.

كفاءة الأداء — الأسئلة المفتاحية: أين سيشعر المستخدمون بالبطء أولًا؟ ما الذي يُحسب مرارًا ويمكن حسابه مرة واحدة وتخزينه في الكاش؟ هل كل مكوّن هو *الأداة الصحيحة* (SQL تؤدي عمل محرك بحث عيب أداء في المعمارية لا في الشيفرة)؟ كيف يتغير الأداء عند عشرة أضعاف — وأين أول عنق زجاجة؟

أنماط الأداء: طبقات الكاش — النمط الأداءي الأعلى رافعةً على الإطلاق: كاش المتصفح ثم **CDN** عند الحافة (المحتوى الساكن يُقدّم من مدينة قريبة من المستخدم؛ ويستطيع امتصاص الغالبية العظمى من حركة القراءة) ثم كاش التطبيق (**Redis** أمام قاعدة البيانات للقراءات الساخنة: خرائط المقاعد، والجلسات، وصفحات المنتجات) ثم كاش استعلامات قاعدة البيانات. كل طبقة تجيب الطلبات قبل أن تبلغ اللبّ المكلف؛ وأسئلة التصميم دائمًا هي *ما الذي يجوز تخزينه في الكاش، ولكم من الوقت، وكيف يُبطل حين تتغير الحقيقة*. ثم: **نسخ القراءة المتماثلة (read replicas)** (نسخ من قاعدة البيانات تخدم القراءات حتى تواصل الرئيسية الكتابة) • **المعالجة اللامتزامنة** (لا تجعل المستخدمين ينتظرون عملاً يمكن أن يحدث لاحقًا — إيصالات البريد، والمصغرات، والتقارير) • **الأداة الصحيحة لكل عمل** (البحث لمحرك بحث؛ والتحليلات لمستودع؛ والاستعلامات الساخنة لمخزن مفتاح-قيمة) • **والقياس أولًا** (عمل الأداء دون قياس شعوذة).

تحسين التكلفة — الأسئلة المفتاحية: كم يكلف هذا التصميم شهريًا عند حمل اليوم — ولكل وحدة (لكل طلب، ولكل عميل)؟ ما الذي يعمل في الثالثة فجراً ولا يحتاج إلى ذلك؟ أي استخدام ثابت يمكن الالتزام به بخصم؟ ماذا ستقول المالية عن الاتجاه؟

أنماط التكلفة: ضبط الأحجام (rightsizing) (معظم الأساطيل متضخمة التجهيز بصمت؛ والتقليص إلى الحاجة المقيسة مال مجاني) • **استراتيجية الحجز** — قائمة التسعير: **on-demand** (السعر الكامل، والحرية

الكاملة) للحمل المتقطع/المجهول، و **reserved instances / savings plans** (التزام سنة إلى ثلاث سنوات، وخصم 30–70%) للخط القاعدي الثابت، و **spot** (خصم حتى 90%)، قابل للاسترداد بإخطار قصير) لأعمال الدفعات القابلة للمقاطعة؛ وحركة المعماري هي التطبيق طبقات: احجز الأرضية، ووسّع الوسط تلقائيًا on-demand، وضع الدفعات على spot · **دورة حياة التخزين** (تدريج الوحدة 2، مؤتميًا) · **أطفئ الأشياء** (بيئات غير الإنتاج النائمة ليلاً وفي عطلات الأسبوع يمكن أن تخفض تكلفتها بالثلثين) · **راقب ال egress** (البيانات /المغادرة للسحابة تفوتر؛ والتصاميم الثرثرة عبر المناطق والتنزيلات العامة الكبيرة تفاجئ الجميع مرة واحدة) · **الوسم و showback** (كل مورد موسوم بمالكة ومشروعه حتى يكون كل باهت قابلاً للنسبة) · والمقياس التاجي، **اقتصاديات الوحدة (unit economics)**: ليس “الفاتورة 800 ألف باهت شهريًا” بل “تكلفة التذكرة المبعة 1.90 باهت وهي في انخفاض.” الفواتير المتنامية لا بأس بها؛ أما تكاليف الوحدة المتنامية فرائحة خلل معماري. ولهذا الانضباط اسم — **FinOps** — والمعماريون يجلسون في قلبه.

مثال محلول — ThaiTicket، بالعدستين. الأداء: CloudFront يقدّم صفحات الفنانين وصور خرائط المقاعد (فمعظم المتفرجين الـ 400,000 لا يلمسون الخوادم أبدًا)؛ و Redis يخزّن توافر المقاعد بمهلة TTL ثانيتين — قديم بما يكفي ليكون رخيصًا، وطازج بما يكفي لأن الدفع (الذي يتحقق مجددًا من Aurora، مصدر الحقيقة) يمنع البيع المزدوج؛ والإيصالات والتذاكر تُولّد لامتزاميًا بعد الدفع. التكلفة: الخط القاعدي الثابت البالغ 2,000 زائر في الساعة يعمل على savings plan (خصم نحو 40%)؛ وموجات الطرح تعمل on-demand للساعة التي توجد فيها؛ ومهام التحليلات تعمل على spot ليلاً؛ وبيئة الاختبار تنام خارج ساعات العمل؛ وكل مورد موسوم project:thaiticket. والعرض المقدّم إلى المدير المالي يقول: “62,000 باهت شهريًا خطأ قاعديًا، ونحو 9,000 باهت لكل حدث طرح كبير، وتكلفة التذكرة نحو 1.90 باهت وتنخفض مع الحجم” — **وتلك الجملة هي ما يبدو عليه معماري طليق في التكلفة.**

المفردات:

المصطلح	التعريف
زمن الاستجابة / الإنتاجية (throughput)	كم يستغرق طلب واحد / كم طلبًا في الثانية يتحمل النظام. متصلان، لا متطابقان.
عنق الزجاجة	أضيق نقطة تحدد إيقاع النظام كله. التحسين في أي مكان آخر زخرفة.
الكاش / TTL / الإبطال (invalidation)	نسخة سريعة من بيانات مكلفة الجلب / كم يجوز الوثوق بها / المشكلة الصعبة: تحديثها حين تتغير الحقيقة.
CDN	الكاش الأبعد — محتواك في مئات المدن. الجواب الأول على “اجعله سريعًا عالميًا.”
نسخة القراءة المتماثلة	نسخة من قاعدة البيانات تخدم القراءات، فتحتفظ الرئيسية بقوتها للكتابات.

المصطلح	التعريف
المعالجة اللامترامنة	إنجاز العمل غير العاجل بعد الرد على المستخدم، عادة عبر طاوور.
On-demand / reserved / savings plan / spot	كامل السعر مرّن / التزام 1-3 سنوات بخصم 30-70% / الفكرة نفسها بمرونة أكبر / خصم حتى 90% لكنه قابل للاسترداد. المعماري يطبّق الأربعة طبقات.
ضبط الأحجام (rightsizing)	تقليص الموارد المتضخمة إلى الحاجة المقيسة. مال مجاني في كل حساب تقريبًا.
Egress	البيانات المغادرة للسحابة — تفوتر بالجيجابايت. مفاجأة الفاتورة الشهيرة؛ صمّم تدفقات البيانات وهي في بالك.
الوسم / showback	وسم كل مورد بالمالك والمشروع / عرض فاتورة كل فريق عليه. المحاسبة تغيّر السلوك.
اقتصاديات الوحدة	التكلفة لكل وحدة عمل (لكل طلب، لكل مستخدم). المقياس الذي يجعل الفاتورة ذات معنى — والجملة المفضلة للمعماري أمام مدير مالي.
TCO	التكلفة الإجمالية للملكية — سعر الملتصق زائد الأشخاص والتراخيص والهجرة وتكاليف الخروج على مدى عمر الخيار كله.
FinOps	انضباط جعل الإنفاق السحابي مرئيًا ومورّغًا ومحسّنًا باستمرار.

فيديو هذه الوحدة (رابط متحقّق منه): — ~2 min — What is FinOps? — FinOps Foundation (official) — https://www.youtube.com/watch?v=Y-c_xw9bHFW

التمارين: (1) سّعّر الخط القاعدي لـ ThaiTicket في حاسبة أسعار AWS (ابحث عن "AWS pricing calculator" — تمرين تفصيل تصميم حقيقي مرة واحدة يزيل الغموض عن كل محادثة تكلفة قادمة)؛ وقرّن مجموعك بمبلغ الـ 62,000 أعلاه وفسّر أي فجوة. (2) أضف طبقة الكاش إلى اثنين من مخططات مرحلتك الأولى: علّم كل كاش، وTTL الخاص به، وقصة إبطاله. (3) ذكاء اصطناعي يلعب دور مهندس يريد كل شيء on-demand "حفاظًا على البساطة" — فاوض على استراتيجية الحجز، بالأرقام.

الإنجاز المرحلي: لأي تصميم رسمته، تستطيع ذكر أكبر ثلاثة بنود تكلفة فيه وأول عنق زجاجة مرّج — واقتراح تغيير واحد يحسّن الاثنين معًا (يوجد واحد في الغالب الأعم؛ والكاش هو عادة الجواب).

الوحدة 7 (الأسبوعان 13-14): التميز التشغيلي والاستدامة — وإدارة الأعمدة الستة معًا

التميز التشغيلي هو العمود الذي يسأل: هل يستطيع البشر فعلًا تشغيل هذا الشيء، بهدوء؟ الأسئلة المفتاحية: كيف يصل تغيير إلى الإنتاج — عبر pipeline مؤتمت مختبر، أم عبر إنسان بطولي؟ كيف نعرف أن النظام سليم الآن؟ حين ينكسر في الثالثة فجراً، ماذا يفعل مهندس المناوبة فعليًا؟ الأنماط: **البنية التحتية كشيفرة** (البيئة معرّفة في نص مُراجع مضبوط الإصدارات — Terraform/CloudFormation — فنكون قابلة لإعادة الإنتاج وللتدقيق؛ والمعمارية التي لا توجد إلا نقراتٍ في الـ console فولكلور لا هندسة) · **خطوط أنابيب CI/CD** باستراتيجيات نشر آمنة (الأزرق-الأخضر blue-green: أقم النسخة الجديدة بجانب القديمة وبدّل؛ والكناري canary: أعطِ النسخة الجديدة 5% من الحركة وراقب) · **قابلية الرصد (observability)** (سجلات ومقاييس وتتبعات وتنبيهات مربوطة بأعراض يراها المستخدم، لا بتوافه الآلات) · الـ **runbooks** (المكتوب مما يجب فعله لكل عطل معروف) · **مراجعات ما بعد الحوادث بلا لوم** (بعد الحوادث: ما الذي فشل، ولماذا، وما الذي يمنع التكرار — بلا أشرار، وإلا توقف الناس عن قول الحقيقة). ودور المعماري: صمّم من أجل قابلية التشغيل — فنظام أذكى مرتين وأقل قابلية للرصد بمرتبتين نظام أسوأ.

الاستدامة، العمود السادس، تطلب منك أن تهدر أقل من العالم المادي: اضبط الأحجام (المعالجات الخاملة تحرق كهرباء حقيقية)، ووسّع مع الطلب بدل التجهيز للذروة، واستخدم الخدمات المُدارة والـ serverless (البنية التحتية المشتركة بنية أكثر امتلاءً)، ودرّج التخزين، واحذف الميت. ومن حسن الحظ أن الخيار المستدام والخيار الرخيص هما عادة الخيار نفسه — قل الاثنين في المراجعات تكسب الغرفة مرتين.

إدارة الأعمدة معًا — أول مراجعة Well-Architected لك. في المراجعات الحقيقية تتعارض الأعمدة: موثوقية تعدد المناطق تصارع التكلفة؛ والمراجعة الأمنية الصارمة تصارع السرعة التشغيلية؛ وكاش الأداء يصارع صحة طزاجة البيانات. والإطار لا يحل التعارضات نيابة عنك — بل يجبرها على الخروج إلى العلن، حيث يستطيع الجانب التجاري الاختيار. تلك عبقريته كلها، وهي لك لتستخدمها. أما منهجية المراجعة نفسها (مجموعات الأسئلة، وترتيب النتائج، والتقارير) فتُدّرّس كاملة في المرحلة 5، الوحدة 12؛ وفي هذين الأسبوعين تتمرن على المهارة الخام: استجواب تصميم واحد من ستة اتجاهات في جلسة واحدة.

المفردات:

المصطلح	التعريف
البنية التحتية كشيفرة (IaC) / Terraform	بنية تحتية معلنة في ملفات نصية مضبوطة الإصدارات تحوّلها الأدوات إلى واقع / أشهر أداة لذلك.
خط أنابيب CI/CD	السير الناقل المؤتمت الذي يبني ويختبر وينشر كل تغيير.
النشر الأزرق-الأخضر / الكناري	نمطا إطلاق آمان: بدّل الحركة بين نسختين قديمة وجديدة / سرّب الحركة إلى النسخة الجديدة وراقب.

المصطلح	التعريف
قابلية الرصد (سجلات، مقاييس، تتبعات)	قدرة النظام على شرح نفسه: سجلات الأحداث، وأرقام عبر الزمن، وخرائط رحلة كل طلب.
التنبه / المناوبة / الـ runbook	النداء الآلي حين تنكسر العتبات / التناوب البشري الذي يجيبه / النص المكتوب الذي يتبعونه.
مراجعة ما بعد الحادث بلا لوم	المراجعة الصادقة بلا أشرار بعد الحادث، التي تنتج منعًا لا عقابًا.
الانحراف (drift)	ابتعاد الواقع عن تعريف الـ IaC لأن أحدهم نقر شيئًا. عدو قابلية إعادة الإنتاج.
الكدح (toil)	العمل التشغيلي اليدوي المتكرر الذي كان ينبغي للأتمتة امتصاصه. المعماربيون يصمّمونه خارجًا.
مراجعة Well-Architected	استجواب منظّم لـ workload على ضوء الأعمدة الستة كلها، ينتج نتائج مرتّبة بالأولوية. المرحلة 5 تعلّمك إدارتها.

التمارين: (1) تمرين العدسات الست: خذ أفضل مخططاتك واقضِ عشر دقائق لكل عمود تكتب فيها النتائج — ستون دقيقة، وتصميم واحد، وست عدسات؛ هذا التمرين هو المرحلة الثانية مصغّرة، فكرره أسبوعيًا من الآن. (2) اقرأ تقريرين من تقارير AWS لما بعد الحوادث (تُنشر على صفحات الحالة لديها بعد الانقطاعات الكبرى) وحدّد أي نمط من أنماط أي عمود هو الذي فشل. (3) الدفتر: اكتب لـ ThaiTicket التعارضات الثلاثة بين الأعمدة التي سترفعها إلى الجانب التجاري، كل منها في جملة خيار واحدة ("يمكننا الحصول على X أو Y بهذه الميزانية — أيهما؟").

الإنجاز المرحلي — نهاية المرحلة الثانية: المراجعة التجريبية: يولّد ذكاء اصطناعي معمارية معيبة لشركة إقراض تايلاندية ناشئة عبر الإنترنت؛ وعليك إنتاج نتائج مكتوبة عبر الأعمدة الستة كلها، تشمل على الأقل: فجوة موثوقية (قاعدتهم أحادية منطقة الإتاحة)، وفجوة أمنية (IAM مفرط الاتساع)، وفجوة تكلفة (كل شيء on-demand)، وفجوة تشغيلية (لا IaC)، ومحادثة RTO/RPO التي لم يجروها قط — كل نتيجة مصوغة سؤالًا يستطيع زميل سماعه دون أن يجفل. حين تُقرأ قائمة نتائجك كأنها من زميل كبير معطاء لا من مدقق حسابات، تكون المرحلة الثانية قد اكتملت.

المرحلة الثالثة — كتالوج الأنماط (الأسابيع 15-20)

المعماريون لا يخترعون؛ إنهم يختارون. هذه المرحلة كتالوجك للمعماريات القياسية — ولكل منها: ما هي، ومتى تُستخدم، ومتى لا تُستخدم، وسيتمتها من حيث التكلفة/التعقيد. أعمدة "متى لا" هي الحمولة الحقيقية لهذه المرحلة: فأي دورة تستطيع إخبارك بما هي الـ microservices؛ أما معرفة متى ستدمر شركة فذلك ما يُدفع لك مقابلته. وأثناء دراستك، واصل تصفح المعماريات المرجعية الحقيقية على <https://aws.amazon.com/architecture> — فرصد الأنماط في البرية هو التمرين الذي يجعل الكتلوج يرسخ.

الوحدة 8 (الأسابيع 15-17): أنماط التطبيقات

الـ monolith مقابل الـ microservices — أعلى جدالات الصناعة ضجيجًا، محسومًا بهدوء. الـ monolith (المتراص) تطبيق واحد قابل للنشر يحتوي كل الميزات؛ والـ microservices (الخدمات المصغرة) تقسم النظام إلى خدمات صغيرة كثيرة تُنشر كل منها مستقلة، وتملك كل منها بياناتها، وتتخاطب عبر APIs وأحداث. الجدول الصادق:

النمط	ما هو	استخدمه حين	لا تستخدمه حين	التكلفة/التعقيد
Monolith	تطبيق واحد، ونشر واحد، وقاعدة بيانات واحدة	فريق صغير، ومنتج فتي، وحدود مجال غير واضحة — أي معظم الأنظمة الجديدة	فرق متعددة يعطل بعضها بعضًا؛ وأجزاء تحتاج إلى توسع متباين جدًا	تعقيد منخفض، وتكلفة منخفضة؛ ويتوسع أبعد مما تعترف به الموضة — "الممل" ميزة
Microservices	خدمات صغيرة كثيرة، تُبنى وتُنشر وتُوسَّع مستقلة	فرق كثيرة تحتاج إلى إيقاع إصدار مستقل؛ وتوسع متباين بشدة بين الأجزاء؛ وحدود مجال مثبتة مستقرة	فريق صغير ("الـ monolith هو monolith أضيفت إليه أعطال الشبكة")؛ ومجال ما زال يتقلب	تعقيد عالٍ: كل نداء دالة يصبح نداء شبكة قد يفشل؛ ويحتاج إلى CI/CD واضح، وقابلية رصد، ومناوبة

حكم المعماري: ابدأ متراصًا، معياريًا من الداخل؛ واستخرج الخدمات حين — وفقط حين — يصل فعلاً ألم توسع الفريق أو توسع الحمل. قول هذا في المقابلات، بأسبابه، يسمك كبيرًا؛ والأيديولوجيا في أي من الاتجاهين تسمك مبتدئًا.

المعمارية المدفوعة بالأحداث — الطواوير والمواضيع، ممتصات الصدمات. بدل أن تتنادى المكونات وتنتظر (متزامنًا)، تنشر المكونات أحداثًا ("OrderPlaced") إلى طابور (queue) SQS — مستهلك واحد يأخذ كل رسالة، بإيقاعه هو) أو إلى موضوع (topic) SNS — كل مشترك ينال نسخة؛ التوزيع المروحي (fan-out). والمنتج لا يعرف من يستمع ولا يهتم. استخدمها حين: ينبغي للمكونات النجاة من انقطاعات بعضها بعضًا

(الطابور يحفظ الرسائل بينما المستهلك متوقف — صمود وفك اقتران في شراء واحد)؛ أو يكون الحمل متقطعًا (الطابور يمتص القفزة؛ والعمال يفرغونه بثبات)؛ أو يطلق حدوث واحد ردود فعل كثيرة (طلب وضع، فتحصيل وبريد ومخزون وتحليلات — أربعة مشتركين، وصفر اقتران). لا تستخدمها حين: يحتاج المستخدم إلى الجواب/الآن (الدفع لا يستطيع التأكيد "في نهاية المطاف")؛ أو يكون الفريق صغيرًا وفي نداء متزامن بسيط بالغرض — فكل طابور يضيف دلالات تسليم (التسليم مرة-على-الأقل يعني أن على المستهلكين أن يكونوا **idempotent** — آمنين عند التشغيل مرتين)، ومعالجة رسائل مितّة، وعبء مراقبة، التكلفة/التعقيد: المكونات رخيصة، والتصحيح أعلى — فقصة الطلب صارت الآن مبعثرة عبر خدمات وطوابير، ولهذا تتوقف قابلية الرصد (الوحدة 7) هنا عن كونها اختيارية.

الثلاثي الطبقات: صار ملكك (الوحدة 3). استخدمه في: الوسط العريض من تطبيقات الويب — فهو النمط الذي تُقاس عليه الأنماط الأخرى. وليس في: الأحمال شديدة التقطع ذات فترات الخمول (ال serverless أرخص) أو المنتجات الضخمة حقًا متعددة الفرق (انظر أعلاه). السّمة: مفهوم في كل مكان، وسهل التوظيف له، ومتوسط التكلفة.

Serverless أولًا: ركب قطعًا مُدارة — API Gateway ثم دوال Lambda ثم DynamoDB وS3 والطوابير — دون امتلاك أي خوادم إطلاقًا. استخدمه حين: الحركة متقطعة أو منخفضة (التوسع إلى الصفر يعني أن تكاليف الخمول تقارب لا شيء)، والفرق صغيرة، وسرعة الوصول إلى السوق مهمة، واللحام المدفوع بالأحداث. لا تستخدمه حين: حوسبة طويلة أو متخصصة (حدود زمن التشغيل)، ومسارات فائقة الحساسية لزمن الاستجابة (الانطلاقات الباردة)، وحمل ثابت هائل (التسعير لكل استدعاء قد يتجاوز الخوادم المحجوزة — أجر الحساب عند الحجم الكبير)، أو حين تكون قابلية النقل بين السحب متطلبًا حقيقيًا (هذا أعرق ارتهان للمورّد في الكتالوج كله — يستحق ثمنه كثيرًا، لكن قله بصوت مسموع). السّمة: أدنى عبء تشغيلي في الكتالوج؛ والتكلفة رائعة عند الحجم المنخفض/المتقطع، وتحتاج إلى تدقيق عند الحجم الثابت العالي.

المفردات:

المصطلح	التعريف
Monolith / ال monolith المعياري	تطبيق واحد قابل للنشر يضم كل شيء / النسخة المنضبطة: قابل نشر واحد بحدود وحدات داخلية نظيفة — أفضل خيار افتراضي للأنظمة الجديدة.
Microservices	خدمات صغيرة كثيرة تُنشر مستقلة، وتملك كل منها بياناتها. أداة لتوسيع الفرق تتقاضى ضريبة أنظمة موزعة.
الاقتران / فك الاقتران (coupling / decoupling)	كم يعتمد كل مكوّن على توافر الآخر وتفصيله. المعمارية إلى حد كبير فن شراء المقدار الصحيح من فك الاقتران.

المصطلح	التعريف
متزامن / لامتزامن	نادٍ وانتظر مقابل أرسل-وواصل. الخيار الجوهري على كل سهم ترسمه.
الحدث / المدفوع بالأحداث	حقيقة تُعلن لمن يستمع ("OrderPlaced") / معمارية مبنية من إعلانات كهذه.
الطابور (SQS)	صف رسائل؛ كل رسالة يأخذها مستهلك واحد بإيقاعه هو. ممتص صدمات وفاك اقتران.
الموضوع / (SNS) pub-sub	قناة بث؛ كل مشترك ينال كل رسالة. أداة التوزيع المروحي.
طابور الرسائل الميتة (dead-letter queue)	حيث تُنحَى الرسائل التي تفشل معالجتها مرارًا لينظر فيها البشر — شبكة أمان النمط.
Idempotent	آمن عند معالجته مرتين بالنتيجة نفسها — مطلوب من المستهلكين، لأن الطوابير قد تسلم الرسالة أكثر من مرة.
Serverless أو لا	تركيب قطع مُدارة تتوسع إلى الصفر (دوال، وقواعد مُدارة، وطوابير) بدل تشغيل خوادم.
الارتهاق للموَّرد (lock-in)	الاعتماد على خدمات مزوِّد واحد الاحتكارية، مسعَّرًا بوصفه تكلفة تبديل. ليس خطيئة — بل مصطلح اقتصادي يوزن في العن (المرحلة 4).

التمارين: (1) صمّم تدفق طلبات ThaiTicket مرتين — ثلاثي الطبقات متزامنًا، ثم مدفوعًا بالأحداث بـ SQS/SNS — واكتب فقرة واحدة عما تفعله كل نسخة أثناء انقطاع مزوِّد الدفع؛ تلك الفقرة هي حجة الأحداث كلها. (2) لأربع شركات (ناشئة من 3 أشخاص، وشركة نامية من 50 مهندسًا، وبنك، وتطبيق تصويت تلفزيوني يُستخدم 4 ليالٍ في السنة)، اختر نمطًا ودافع عنه — ثم سمِّ المحقِّز الذي سيجعل كل شركة تغيّر نمطها. (3) اعثر على قصة حقيقية واحدة من مدونات الهندسة بعنوان "هاجرنا إلى الـ microservices وندمنا" وقصة نجاح واحدة؛ ودوّن الفرق في دفترك (إنه في الغالب الأعم حجم الفريق ونضج المجال).

الإنجاز المرحلي: إذا أعطيت وصف شركة، تستطيع التوصية بنمط مع منحدر خروجه — "ابدأ هنا؛ وحين يحدث X، تطوّر إلى Y" — في خمس دقائق. مسارات التطور، لا الأحكام النهائية، هي الطريقة التي يتكلم بها المعمارون فعلاً.

الوحدة 9 (الأسابيع 18-20): أنماط البيانات والأنماط العالمية والهجينة

معماريات البيانات — البحيرة مقابل المستودع. قواعد البيانات المعاملاتية (المرحلة الأولى) تدير العمل؛

والتحليلات تريد طرح الأسئلة عليه دون إبطائه. **مستودع البيانات (data warehouse)** (Redshift وSnowflake وBigQuery) يخزن بيانات منظمة ومنظمة محسنة لتحليلات SQL السريعة — استخدمه حين

تكون الأسئلة معروفة ويجب أن تكون اللوحات سريعة؛ يكلف أكثر لكل تيرابايت، ويحتاج إلى انضباط نمذجة مسبقًا. وبحيرة البيانات **(data lake)** (S3 + كتالوج + محركات استعلام مثل Athena) تخزن كل شيء، خافًا،

بشمن بخس — سجلات وتدفقات نقر وصورًا — وتطبق البنية عند القراءة؛ استخدمها حين تريد الاحتفاظ بكل شيء الآن وتقرير الأسئلة لاحقًا؛ لكنها إن تُركت بلا حوكمة تنحط إلى مزحة الصناعة، **مستنقع البيانات (data swamp)**.

والإجماع الحديث هو الاثنان معًا، طبقات ("lakehouse"): الحقيقة الخام في البحيرة، والأسواق المنسقة في المستودع، تغذيها **خطوط أنابيب ETL/ELT**. قاعدتا المعماري: التحليلات لا تستعلم قاعدة

الإنتاج أبدًا (النسخ المتماثلة أو خطوط الأنابيب تغذيها)، ولكل مجموعة بيانات مالك ومدخل كتالوج وسياسة احتفاظ — سطر الحوكمة ذاك جملة واحدة في تصميمك وعام من الألم إن أغفلته.

تعدد المناطق: نشط-خامل مقابل نشط-نشط. حين لا تكفي منطقة واحدة — لأن العمل يطالب بتعافي من كوارث بحجم المنطقة، أو لأن المستخدمين يمتدون عبر القارات — تختار:

النمط	ما هو	استخدمه حين	لا تستخدمه حين	التكلفة/التعقيد
نشط-خامل (active-passive)	منطقة واحدة تخدم؛ ومنطقة احتياطية تحمل بيانات منسوخة، من "نسخ احتياطية فقط" (باردة) إلى "نسخة مصغرة عاملة" (دافئة)، تُرقى عند الكارثة	حين تسوّغه قيمتا RTO/RPO المعلنتان من الجانب التجاري؛ أو حين يطالب الامتثال بقصة تعافي	حين لم يوّع أحد على RTO/RPO للذين يسوّغان الإنفاق (حاجة معظم الشركات الصادرة: multi-AZ جيد + نسخ احتياطية عبر المناطق)	1.1-1.7× من التكلفة بحسب دفعء الاحتياطي؛ وتعقيد متوسط — المتطلب القاتل هو <i>failover</i> تتدرب عليه فعليًا، وإلا فالاحتياطي مسرحية
نشط-نشط (active-active)	منطقتان أو أكثر تخدمان معًا؛ والمستخدمون يوجهون إلى الأقرب؛ والبيانات تُنسخ في الاتجاهين	قاعدة مستخدمين عالمية تريد زمن استجابة محليًا؛ أو RTO يقارب الصفر مطلوب حقًا (شبكات مدفوعات، وتداول، SaaS كبير)	كل من عداهم تقريبًا — فـ تعارضات الكتابة عبر المناطق من المشكلات الصعبة حقًا في الحوسبة	2×+ من التكلفة، وأعلى تعقيد في هذه الدورة؛ يحتاج إلى مخازن بيانات حائلة للتعارضات وفرق كبيرة

الجملة بمستوى المقابلات: "multi-AZ" من بدهيات المائدة؛ أما تعدد المناطق فقضية عمل تُدرس. "اجعل الجانب التجاري يعلن RTO/RPO، وسعّر الخيارات، ودع الأرقام تختار.

السحابة الهجينة. جزء on-prem وجزء في السحابة، يصلهما VPN أو Direct Connect — وهي لمعظم المؤسسات الراسخة ليست نمطًا بل واقعًا يمتد عقدًا كاملاً: حواسيب مركزية mainframes لا تستطيع الانتقال، وأنظمة مصانع مقيدة بزمان الاستجابة، وبيانات خاضعة لتنظيم مثبّنة محليًا، وهجرة (المرحلة 4) تمر عبرها. استخدمها بوصفها: جسرًا مقصودًا له اتجاه سير. ولا تقبل: “هجين” تلطيفًا لعبارة “لم نقرر قط”. ملاحظات تصميم: يجب توحيد الهوية أولًا (تسجيل دخول واحد عبر العالمين — وهو ما تعنيه الأوصاف الوظيفية المؤسسية بعبارة “hybrid identity integration”); وسمّ أي الأنظمة هو مصدر الحقيقة؛ وراقب تكاليف ال egress عبر الوصلة؛ وتوقع أن تكون وصلة الشبكة هي نقطة الفشل الوحيدة ما لم تُضاعَف. التعقيد: كل شيء بمقدار منشأتين — وهذا هو السبب الصادق الذي يجعل المماريين يدفعون نحو تقليص الجانب المحلي باطراد.

المفردات:

المصطلح	التعريف
OLTP / OLAP	معالجة المعاملات (قراءات/كتابات كثيرة صغيرة سريعة — تدير العمل) مقابل المعالجة التحليلية (مسوح ضخمة — تدرس العمل). افصل بينهما.
مستودع البيانات	تخزين منظّم منسّق محسّن لتحليلات SQL السريعة (BigQuery و Snowflake و Redshift).
بحيرة البيانات / مستنقع البيانات	تخزين رخيص لكل شيء، خامًا، يُبنى عند القراءة (S3 + Athena) / النسخة بلا حوكمة، حيث تذهب البيانات لتضيع.
ETL / ELT	خطوط الأنابيب التي تنقل البيانات من الأنظمة المصدّرة إلى البحيرة/المستودع (استخراج، تحويل، تحميل — والترتيب يتفاوت).
حوكمة البيانات / الكتالوج / الاحتفاظ	قواعد الملكية والتوثيق والعمر لكل مجموعة بيانات — جملة تصميم واحدة توفر عامًا من الألم.
النسخ المتماثل (متزامن / لامتزامن)	نسخ البيانات باستمرار إلى قاعدة أو منطقة أخرى — متسق فورًا لكنه محدود بالمسافة مقابل متأخر قليلًا لكنه يصل إلى أي مكان. تأخر اللامتزامن هو مصدر RPO.
نشط-خامل / تدريب ال failover	تعافي كوارث بمنطقة احتياطية / البروفة المجدولة التي تثبت أنه يعمل. ال failover غير المتدرّب عليه أمنية.

المصطلح	التعريف
نشط-نشط	مناطق متعددة تخدم معًا بنسخ متماثل ثنائي الاتجاه. قوي؛ وصعب حقًا؛ وغير ضروري في العادة.
تعارض الكتابة	منطقتان تغيّران البيانات نفسها في آن واحد — السبب التقني لكون النشط-نشط أرض خبراء.
السحابة الهجينة / الهوية الهجينة	on-prem + سحابة موصولتين منشأة واحدة / تسجيل دخول واحد عبر الاثنين — أول ما يجب تويحيده.
Direct Connect	الخط المادي الخاص الواصل بين مركز البيانات والسحابة — الحبل السري الهجين، مضاعفًا إن كان مهمًا.

التمارين: (1) ThaiTicket تتوسع إقليميًا: صمّم توسعة سنغافورة مرتين — نشط-خامل (بانكوك رئيسية) ونشط-نشط — بمضاعفات التكلفة وRTO/RPO لكل منهما؛ واكتب توصية الصفحة الواحدة واحسم القرار. (2) لدى شركة تجزئة 15 عامًا من المبيعات في قاعدة SQL إنتاجية وتريد “تحليلات جاهزة للذكاء الاصطناعي”؛ ارسم تصميم البحيرة + المستودع واكتب جمل الحوكمة الثلاث. (3) رصد الأنماط: اختر ثلاث معماريات مرجعية من <https://aws.amazon.com/architecture> وسمّ كل نمط من الكتالوج حاضرًا في كل منها.

الإنجاز المرحلي — نهاية المرحلة الثالثة: تحدي الكتالوج: يعطيك ذكاء اصطناعي ستة سيناريوهات متلاحقة؛ تسمي لكل منها النمط، والسبب، وتحذير متى-لا، وبيمة التكلفة/التعقيد — في أقل من خمس دقائق للواحد. هذا التبادل بعينه، بهذه السرعة بعينها، هو الثلث الأوسط من مقابلة معماري حقيقية.

المرحلة الرابعة — الهجرة والعالم الحقيقي (الأسابيع 21-26)

تصميم الـ greenfield هو أقلية الوظيفة. معظم المعمارية تحدث في شركات موجودة أصلاً — بغرف خادماها، وبرمجيات عتيقة يعتمد عليها العمل، وعقود، وقوانين. هذه المرحلة هي ذلك العالم، وفيها يُدرّس بند “قيادة مبادرات الهجرة والتحديث” — الموجود في كل وصف وظيفي درسنه تقريبًا.

الوحدة 10 (الأسابيع 21-23): الهجرة — نقل شركة إلى السحابة

الراءات السبع (7 Rs) — المفردات المشتركة لكل محادثة هجرة. لكل workload في المنشأة، تختار واحدة:

الرائ	المعنى	متى	الجهد / العائد
Retire (تقاعد)	أطفئه — لا أحد يستخدمه فعليًا	كل منشأة فيها 10-20% من هذه؛ اعثر عليها أولاً	تافه / وفر فوري — أفضل الرءاءات
Retain (إبقاء)	اتركه on-prem، حاليًا	الأنظمة المقيدة بزمان الاستجابة، أو المثبتة امتثالًا، أو القريبة من نهاية عمرها	لا شيء / يؤجل التكلفة — "ليس بعد" صادقة
Rehost ("الرفع والنقل" lift and shift)	انقله إلى آلات سحابية كما هو	السرعة مهمة، والتطبيقات مستقرة، والمهارات قليلة	منخفض / خروج سريع من مركز البيانات، لكن دون فائدة سحابية تُذكر بعد — خطوة أولى لا وجهة
Relocate (إعادة توطين)	انقله على مستوى الـ hypervisor (مثل أسطول VMware إلى VMware-على السحابة)	منشآت افتراضية كبيرة على موعد نهائي	منخفض / أسرع نقل جماعي؛ محطة عبور
Repurchase ("الترك والشراء drop and shop")	استبدل به SaaS	التطبيقات غير المميّزة — البريد، والموارد البشرية، وCRM	منخفض-متوسط / فئات كاملة تغادر منشأتك
Replatform ("رفع وتعديل ونقل lift, tinker, shift")	ترقيات صغيرة أثناء النقل — قاعدة مُدارة ذاتيًا إلى RDS، وتطبيق إلى حاويات	الوسط العملي: فائدة حقيقية بمخاطرة محدودة	متوسط / حضان العمل بين رءاءات معظم الهجرات
Refactor / إعادة التصميم المعماري	أعد الكتابة لتكون سحابية أصيلة (أنماط المرحلة 3)	الأنظمة الجوهرية المميّزة التي تؤلم حدودها العمل	عالٍ / أعلى عائد — أنفقه على الأحمال القليلة التي تستحقه

منهجية الهجرة: قَيِّم ثم عَيِّ ثم هاجر. قَيِّم (assess): احصر كل شيء (أدوات اكتشاف إضافة إلى أركيولوجيا سؤال الناس)، وقَيِّم كل workload على قيمته للعمل وصعوبة هجرته — المخرجات: جرد تطبيقات براء لكل صف، وقضية عمل (business case) (التكلفة الإجمالية للبقاء مقابل الانتقال؛ وكن صادقًا في أن الفاتورة ترتفع خلال فترة التداخل حين تعمل المنشأتان معًا — فالقادة الذين لم يُحذِّروا من فقاعة الهجرة يصبحون قادة بلغون الهجرات في منتصفها). عَيِّ (mobilize): ابنِ منطقة الهبوط (landing zone) — الأساس السحابي المحكوم المبني مسبقًا قبل أن يهبط أول workload: بنية حسابات متعددة (حسابات منفصلة لكل بيئة وفريق، حتى تبقى نطاقات الانفجار صغيرة)، وهوية وتسجيل ممرزان، ومحور الشبكة (VPCS) بهيكل

hub-and-spoke، Direct Connect عائد إلى الـ on-prem)، ودرابزينات (guardrails) — سياسات مؤتمتة تجعل المسار الآمن الموسوم الممثل هو الافتراضي (AWS Control Tower هي عدة البدء المُدارة). الأوصاف الوظيفية تقول “صُمِّم منطقة الهبوط التي يرثها كل workload” — هذا هو ذاك. وتخطيها من أجل “البدء بالهجرة فورًا” يعيد إنتاج مركز البيانات الفوضوي في السحابة بتكلفة أعلى؛ وذلك هو التوقيع الكلاسيكي للهجرة الفاشلة. هاجر في موجات: جَمِّع الأحمال في موجات قليلة العدد، مرتبةً الأسهل أولًا: الموجة 1 منخفضة الرهان عمدًا (تعلَّم الآلية حيث الأخطاء رخيصة)، والموجات اللاحقة تأخذ جواهر التاج بعمليات قطع cutover متمرّن عليها، لكل منها خطة تراجع وفترة hypercare من المراقبة المشددة. ولكل نقل قاعدة بيانات، السؤالان المهمان هما رقما الوحدة 4 متكررين: كم من التوقف يجوز أن يستغرق القطع (RTO) — والنسخ المتواصل مع تحويل قصير موجود حين يكون الجواب “لا شيء تقريبًا”.

المفردات:

المصطلح	التعريف
الراءات السبع (Rs 7)	Retire و retain و rehost و relocate و repurchase و replatform و refactor — قائمة الهجرة لكل workload. (ستسمع أيضًا “Rs 6” — القائمة الأقدم دون relocate).
الاكتشاف / جرد التطبيقات	معرفة ما يعمل فعليًا (أدوات + مقابلات) / القائمة الناتجة، صف لكل workload، بالمالك والتبعيات ورائه.
رسم خرائط التبعيات	تخطيط ما يخاطب ماذا — سبب هجرة الأحمال في مجموعات، والهجرات بدونه تفشل من اليوم الأول.
قضية العمل / فقاعة الهجرة	حجة التكلفة الإجمالية للبقاء-مقابل-الانتقال / سنام التكلفة المؤقت بينما تعمل المنشأتان. حدّر منه أو ثبّأته به.
منطقة الهبوط (landing zone)	الأساس السحابي المحكوم المبني مسبقًا — حسابات، وهوية، وشبكة، وتسجيل، ودرابزينات — الذي يرثه كل workload. يُبنى قبل الهجرة.
استراتيجية الحسابات المتعددة	حسابات سحابية منفصلة لكل بيئة/فريق، فتكون الفوترة قابلة للنسبة ونطاق الانفجار محتوى.
الدرابزين (guardrail)	سياسة مؤتمتة تمنع الأفعال غير الممثلة أو تعلّمها — الحوكمة آلات لا مذكرات.
Control Tower	خدمة AWS المُدارة لإقامة منطقة هبوط متعددة الحسابات بدرابزينات.

جدول الهجرة في مجموعات صغيرة، الأسهل أولاً،
والتبعيات معًا.

خطة الموجات

لحطة التحويل إلى النسخة السحابية / التراجع المتمرن
عليه / نافذة الدعم المشدد بعدها.

القطع / التراجع / hypercare

التمارين: (1) الهجرة الورقية: يولد ذكاء اصطناعي جردًا لشركة خيالية من 40 خادمًا (ستلقاها مجددًا بوصفها المشروع الختامي 2)؛ عيّن رأيًا لكل صف ودافع عن أصعب عشرة قرارات. (2) ارسم منطقة هبوط: مخطط حسابات، وتدفق هوية، ومحور شبكة hub-and-spoke، وخمسة درابزينات ستفرضها من اليوم الأول. (3) اكتب تحذير "فقاعة الهجرة" من فقرتين إلى مدير مالي — فالتدرب على قول الخبر السيئ مبكرًا هو تمارين قلب المعماري.

الإنجاز المرحلي: إذا أُعطيت جردًا من نحو 20 workload بأوصافها، تنتج جدول راء-لكل workload قابلاً للدفاع عنه، وخطة من ثلاث موجات بمنطقها، ورسم منطقة هبوط — في جلسة واحدة.

الوحدة 11 (الأسابيع 24-26): القيود — القانون، والأنظمة الموروثة، والارتهان للمورد

إقامة البيانات وقانون الخصوصية مُدخلين معماريين. قوانين البيانات الشخصية — PDPA التايلاندي، وGDPR الأوروبي، وأبناء عمومتهما حول العالم — تتشارك شكلاً يجب أن يعرفه المعماري عن ظهر قلب: البيانات الشخصية تحتاج إلى أساس قانوني (الموافقة غالبًا)؛ ولأفراد حقوق (الاطلاع، والتصحيح، والمحو — يجب أن يستطيع تصميمك العثور على بيانات شخص واحد وحذفها، وذلك صعب إن كنت بعثرتها عبر نسخ بلا حكمة)؛ والاختراقات تحمل واجب إبلاغ خلال 72 ساعة (يجب أن تتيح لك سجلاتك رواية القصة في أقل من ذلك)؛ وقواعد النقل عبر الحدود تقيد أين يجوز للبيانات أن تقيم فيزيائيًا — وذلك قيد على اختيار المنطقة، وقيد على تصميم النسخ المتماثل (نسخة التحليلات تلك في سنغافورة قد تكون حديثًا قانونيًا)، وسبب وجود المناطق داخل البلدان. منهجية المعماري، بالترتيب: صنّف البيانات (ما الشخصي منها؟)، وارسم رحلاتها (كل مخزن، وكل نسخة، وكل حدود تُعبر — فالتدفقات التي لم يرسمها أحد هي حيث تعيش الانتهاكات)، ثم صمّم الضوابط: مناطق ممثلة للإقامة، وتشفير، وجداول احتفاظ، ومسار حذف يصل فعليًا إلى انقضاء النسخ الاحتياطية. قل "حماية البيانات بالتصميم (data protection by design)" — فهي العبارة التي يستخدمها القانونان كلاهما، وهي حرفيًا مسماك الوظيفي في جملة.

التكامل مع الأنظمة الموروثة. نظام فوترة الحاسوب المركزي لن ينتقل هذا العام، وعلى التطبيق السحابي اللامع أن يخاطبه. الأنماط: طبقة مانعة للفساد (anti-corruption layer) — خدمة ترجمة بين الجديد والقديم، حتى لا تتسرب أشكال بيانات النظام الموروث الغريبة إلى تصميمك الجديد فتفسده؛ وتينة الخنق (strangler fig) — مَرّر الحركة عبر واجهة أمامية، ثم اقشر الوظائف عن النظام الموروث واحدة واحدة حتى يمكن، بعد سنوات، إطفاءه (سميت باسم التينة التي تلتف ببطء حول شجرة مضيضة؛ وهي تتفوق على إعادات الكتابة بالضربة الكبرى، التي تفشل بمعدل أسطوري)؛ وجسور دفعات وصنابير أحداث للبيانات التي

يجب أن تتدفق بين العالمين؛ واحترام لفيزياء زمن الاستجابة في النداءات الثرثرة بين السحابة وال on-prem (يساعد Direct Connect؛ والأفضل: صمّم الثرثرة خارجًا). القاعدة: احتو الموروث، ولا تلتقط عدواه — فكل مكوّن جديد ينبغي أن يُبنى كأن النظام الموروث قد زال بالفعل.

اقتصاديات الارتهان للمورّد. كل خدمة مُدارة مريحة تعمّق زواجك بمزوّد واحد؛ وقابلية النقل (تجريدات متعددة السحب، وإدارة ذاتية لكل شيء) خيار حقيقي يكلف مألّا حقيقيًا تعقيدًا وسرعةً مفقودة. لا هذا خطيئة ولا ذاك — نمط الفشل ليس اختيار الارتهان، بل عدم ملاحظته. أداة المعماري تقدير تكلفة خروج يُكتب في التصميم: “استخدام DynamoDB يوفر نحو سنتي مهندس الآن؛ والتبديل لاحقًا يقارب إعادة كتابة طبقة البيانات في 6 أشهر — نقبل هذا، وإليكم حد الواجهة الذي سيقصص إعادة الكتابة.” ثلاث جمل، مسعّرة بصدق، وقرار مدوّن (في ADRs المرحلة 5) — تلك إدارة ارتهان ناضجة. ال multi-cloud / ستر/تيجيّة (تشغيل ال workload نفسه بقابلية نقل على سحابتين) هو عادة أعلى إجابة ممكنة ولا يشتره إلا المنظمات التي يفرضه حجمها أو منظّمها بالضبط؛ أما ال multi-cloud واقعا (أحمال مختلفة على سحب مختلفة بحكم التاريخ أو الملاءمة) فحياة عادية.

المفردات:

المصطلح	التعريف
PDPA / GDPR	قانونا حماية البيانات الشخصية في تايلاند وأوروبا — الثنائي النموذجي لقيود قوانين الخصوصية حول العالم.
حماية البيانات بالتصميم	بناء ضوابط الخصوصية في المعمارية من أول رسمه — العبارة القانونية وواجب المعماري.
الأساس القانوني / الموافقة	التسويغ القانوني المطلوب لمعالجة البيانات الشخصية.
الحق في المحو	حق الفرد في حذف بياناته — الذي يجب أن يجعله تصميمك ممكنًا.
رسم خرائط تدفق البيانات	تخطيط كل مخزن ونسخة وعبور حدود لفئة من البيانات. حيث التدفقات غير المرسومة، تعيش الانتهاكات.
النقل عبر الحدود	البيانات الشخصية المغادرة للبلد — خاضع لتنظيم؛ يجعل تصميم النسخ المتماثل سؤالًا قانونيًا.
الطبقة المانعة للفساد	مكوّن ترجمة يمنع غرائب النظام الموروث من التسرب إلى تصميم جديد.
تينة الخنق (strangler fig)	التحديث عبر التمرير من واجهة أمامية واستبدال النظام الموروث قطعة قطعة حتى يمكن إطفائه.

المصطلح	التعريف
إعادة الكتابة بالضربة الكبرى	استبدال نظام دفعة واحدة، بقطع في يوم واحد. معدل فشل أسطوري؛ وتينة الخنق وُجدت بسببها.
تكلفة الخروج / تكلفة التبديل	التكلفة المسعّرة واقعيًا لمغادرة مزوّد أو خدمة — الرقم الذي يحوّل الارتهان من خوف إلى اقتصاد.
Multi-cloud (استراتيجية مقابل واقع)	التشغيل المقصود بقابلية نقل على سحب متعددة (مكلف، ونادرًا ما يُسوَّغ) مقابل مجرد امتلاك أحمال على عدة سحب (عادي).

التمارين: (1) ارسم خريطة تدفق بيانات عملاء ThaiTicket: كل مخزن، وكل نسخة (لا تنسَ السجلات، والنسخ الاحتياطية، وخط أنابيب التحليلات، وتصديرات فريق الدعم)، وكل حدود؛ ثم اكتب مسار المحو. تحسّس كيف تجد الخريطة مشكلات أخفاها المخطط. (2) صمّم خطة تينة الخنق لنظام مخزون on-prem عمره 20 عامًا، مسميًا القشرات الثلاث الأولى. (3) اكتب فقرة الارتهان على طراز DynamoDB (الفائدة الآن، وتكلفة الخروج لاحقًا، ومقبولة أو مخفّفة) لثلاث خدمات ستستخدمها فعليًا.

الإنجاز المرحلي — نهاية المرحلة الرابعة: تحدي القيود: موجز تصميم واحد مضفور بالقيود الثلاثة كلها ("شركة تأمين تايلاندية، وبيانات عملاء، ونظام وثائق على حاسوب مركزي، ومجلس إدارة قلق من الاعتماد على AWS") — تنتج نهج الصفحة الواحدة الذي يمس الإقامة والتكامل والارتهان، لكل منها نمط مسمى وتكلفة صادقة. حين تشارك القيود أكثر مما يشارك الـ greenfield — لأن القيود هي حيث يتفوق المعماريون في الكسب على رسامي المخططات — تكون المرحلة الرابعة قد انتهت.

المرحلة الخامسة — حرفة المعماري (الأشهر 7-12)

كل ما سبق جعلك قادرًا على التصميم. وهذه المرحلة تجعلك قادرًا على العمل معماريًا — الوثائق والمراجعات والعروض والبراهين التي يتكون منها الدور فعليًا، إضافة إلى الشهادات التي تعبر بك بوابة الموارد البشرية، وثلاثة مشاريع ختامية تصبح ملفك المهني.

الوحدة 12 (الشهران 7-8): المخرجات — الـ ADRs وأطقم المخططات وإدارة المراجعات

سجلات القرارات المعمارية (Architecture Decision Records — ADRs). الـ ADR وثيقة من صفحة واحدة تلتقط قرارًا مهمًا واحدًا: ما اخترناه، وما لم نختره، ولماذا — تُكتب ساعة اتخاذ القرار، وتُرقّم، وتُحفظ

في مستودع المشروع إلى الأبد. لماذا تسميها الأوصاف الوظيفية بالاسم: بعد عامين سيسأل أحدهم "لماذا بحق السماء هذه DynamoDB؟" فيجب الـ ADR في ثلاثين ثانية — بالسياق، والقيود، والبدايل التي درست بصدق. الفرق التي لديها ADRs لا تعيد التقاضي في شيء؛ والفرق التي بدونها تتجادل في دوائر سنويًا. **ال قالب — احفظه:**

ADR-014: Use DynamoDB for the session store #

Status: Accepted

Date: 2026-08-13

Context: What situation forced a decision? (Load, constraints, deadlines – the facts.)

Decision: What we chose, in one sentence, active voice: "We will

Options considered: 2–3 real alternatives, each with honest pros/cons – including the one you rejected reluctantly

Consequences: What becomes easier; what becomes harder; the risks we accept; the exit cost

اكتب ADR واحدًا لكل قرار مهم في كل مشروع ختامي. وملف مقابلات يحتوي ADRs حقيقية نادر وفعال إلى حدٍ ساحق.

طقم المخططات — نظام واحد، ثلاثة مستويات تقريبا. التوثيق المعماري الحقيقي طقم (نموذج C4 هو الذي نشر هذا الانضباط): **مخطط السياق** — النظام صندوقًا واحدًا، مع مستخدميه والأنظمة الخارجية التي يلمسها؛ **منظور التنفيذيين والمنضمين الجدد**، والذي ستعرضه على القيادة. **ومخطط الحاويات** — قَرَّب: القطع العاملة الكبرى (تطبيق الويب، وAPI، وقاعدة البيانات، والطابور، والكاش) وكيف تتخاطب؛ **المنظور الهندسي اليومي**، وهو تقريبًا رسوماتك في المراحل 1–3. **ومخطط النشر** — أين يعمل كل ذلك فيزيائيًا: المنطقة، ومناطق الإتاحة، والـ VPC، والـ subnets، ومجموعات التوسع؛ **منظور المراجعات والأمن والعمليات**. نظام واحد، وثلاثة جماهير، وثلاثة مخططات — محدّثة، ومؤرّخة، في ضبط الإصدارات بجانب الـ ADRs. المخطط الذي لا يمكن العثور عليه أو الوثوق به مخطط غير موجود.

إدارة مراجعة Well-Architected. أنت تعرف الأعمدة الستة (المرحلة 2)؛ وإليك الاجتماع نفسه. **قبل:** اختر الـ workload والنطاق، وحدّث طقم المخططات، وادعُ من يشغّلون الشيء (لا مصمّميه فقط)، واضبط النبرة — هذا فحص صحي لمصلحة الفريق، لا تدقيق لملف أحد. **أثناء (نصف يوم):** امشِ عبر الأعمدة بمجموعات أسئلة الإطار (تنشرها AWS، مع أداة Well-Architected Tool مجانية في الـ console تهيكل التمرين كله — <https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html>)؛ ولكل إجابة، دوّن النتيجة، ورّب الأولويات وأنت تمضي — حفنة من القضايا عالية الخطورة، لا مئة ملاحظة تافهة. نبرة أسئلتك هي المراجعة: "ساعدني أن أفهم ماذا يحدث حين تنقضي مهلة مزوّد الدفع" تفتح أبوابًا تصفعها "ألا تعالجون المُهل؟". **بعد:** تقرير قصير — أهم النتائج، لكل منها خطورة وجهد وتوصية — وتهبط بنود التحسين في backlog الفريق الحقيقي بمالكين، وإلا كانت المراجعة مسرحية. تدرب: أدر مراجعة كاملة على تصميم مشروعك الختامي الأول؛ فإيجاد عيوبك بطريقة منظّمة هو أسرع مهارات هذه الدورة تراكُمًا.

التمارين: (1) اكتب بأثر رجعي ADRs لخمسة قرارات كبرى في تصاميم دفترك من المرحلتين 2-3 — ستجد واحدًا على الأقل لم تعد قادرًا على تسويغه، وذلك هو الدرس. (2) أنتج طقم المخططات الثلاثي الكامل لـ ThaiTicket. أدر المراجعة التجريبية أعلاه، وأنتج تقرير النتائج من صفحة واحدة.

الإنجاز المرحلي: يستطيع غريب (أو ذكاء اصطناعي يلعب دوره) التقاط حزمة ThaiTicket خاصتك — ثلاثة مخططات + خمسة ADRs — والإجابة الصحيحة عن “ما هذا، وكيف يعمل، ولماذا بُني هكذا؟” دون وجودك في الغرفة. تلك الحزمة هي مخرج وظيفة المعماري.

الوحدة 13 (الشهران 8-9): الحرفة الإنسانية — العرض، والتكليف، والاعتراضات، والإرشاد

العرض على التنفيذيين مقابل المهندسين — تصميم واحد، لغتان. التنفيذيون يشترون النتائج؛ والمهندسون يشترون الآليات. للتنفيذيين: ابدأ بالقرار والرقم (“هذا التصميم يدعم هدف المليون عميل بتكلفة 1.90 باهت للطلب، وإليك القرارين اللذين أحتهما منكم”)؛ وشريحة واحدة، ومخطط السياق، والمخاطر مؤطرة لمخاطر عمل (الإيرادات، والامتثال، والسمعة)، ولا تقل “Kubernetes” أبدًا حين تكفي “المنصة”. واستعد للأسئلة الثلاثة التي يطرحها التنفيذيون دائمًا: كم يكلف، وما الذي قد يسوء، ولماذا ليس الخيار الأرخص؟ وللمهندسين: ابدأ بالمشكلة والقيود قبل الحل (فالمهندسون الذين يتحسسون المشكلة يقبلون الحل؛ والمهندسون الذين يُسلمون حكمًا جاهزًا يصطادون العيوب من حيث المبدأ)، واعرض مخططي الحاويات والنشر، وسمِّ المفاضلات والخيارات المرفوضة بنفسك (المصادقية تأتي مما تعترف به)، واترك مجالًا حقيقيًا لتغيير التصميم — فالأسئلة في الغرفة مراجعة مجانية. أسوأ عادات المعماري عرض واحد للجمهورين؛ وأفضلها كتابة الملخص التنفيذي/وَلَا، لأن التصميم إن لم ينبُج من ضغطه إلى خمس جمل، فهو غير مكتمل.

تقدير التكاليف لعرض مقترح. المنهجية: فكِّك التصميم إلى مكوناته العشرة الحاملة للتكلفة تقريبًا؛ وسعِّر كل واحد عند الحمل المتوقع في حاسبة الأسعار؛ ودوِّن افتراضاتك كتابة (الطلبات في اليوم، ونمو البيانات، وال egress — فالافتراضات هي التقدير؛ وحين تتغير، يتحرك الرقم بضمير مرتاح)؛ وطبِّق استراتيجية الحجز على الأجزاء الثابتة؛ وأضف سيناريوهات نمو (اليوم، و2×، و10× — فالتنفيذيون يتذكرون رقم الـ 10×)؛ واعرض النتيجة *نطاقًا* بمحركاته (“55-70 ألف باهت شهريًا، يقوده الـ egress أساسًا — وهذه هي الرافعة”)، لا رقمًا واحدًا بدقة زائفة أبدًا. وأدرج فقاعة الهجرة إن وُجدت. التقدير المنخفض لكسب الموافقة هو خطيئة المعماري المبتدئ الكلاسيكية؛ والمشروع يتذكر.

التعامل مع الاعتراضات. سُدِّد في التكلفة (“غالبًا جدًا” — عد إلى المثلث: “نستطيع خصم 30 ألف باهت بقبول منطقة إتاحة واحدة — إليك حساب الانقطاع؛ القرار قرارك، وسأدوِّنه” — فجعل المقايضة صريحة ومسجلة يحوّل معظم الاعتراضات إلى اتفاق أو إلى خطر مقبول عن علم، وكلاهما جيد)؛ وفي الذوق (“مهندس يفصل حزمة تقنية أخرى” — دافع عن رأيه بصدق بصوت مسموع، ثم انقل الأمر إلى معايير لا أذواق: الملاءمة للـ NFRs، ومهارات الفريق، والمنظومة، وتكلفة الخروج — وحين يكون الأمر متقاربًا حقًا، دع تفضيل المهندس المنفَّذ يفوز، فذلك شراء رخيص للالتزام)؛ وفي السلطة (“صديق المدير التقني يقول استخدموا X” — لا تقا تل رأياً برأى أبدًا؛ اطلب المعايير، ومُرّر X عبر التقييم العلني نفسه ككل شيء آخر،

ودع المصفوفة تجيب بأدب). وحين تكون أنت المخطئ — وستصم شيئًا يفشل؛ كل معماري يفعل — الحركة هي مراجعة ما بعد الحادث بلا لوم مطبقة على نفسك، علنًا، مع تحديث ال-ADR. لا شيء يبني مصداقية عقد كامل أسرع؛ ولا شيء يدمرها أسرع من الدفاع عن جثة.

إرشاد المهندسين. عبارة الوصف الوظيفي هي “أرشد المهندسين الذين يبنون وفق معايير”، والأشكال العملية: **ساعات مكتبية لمراجعة التصميم** (باب مفتوح دائم، فيحدث التوجيه قبل الشيفرة لا بعدها)؛ **ومراجعة تصاميمهم بلطف** — أسئلة قبل الأحكام، ودائمًا لماذا وراء المعيار (الدرابزين المشروح يجتد حليقًا؛ والدرابزين المفروض يجتد التفافًا)؛ **وتفويض قرارات حقيقية** بشبكة أمان (“أنت تملك تصميم الكاش؛ وهذه القيود؛ وسأراجع، وسأساند قرارك”)؛ **والتعليم في العلن** — كل ADR، وكل تقرير مراجعة، وكل جلسة غداء تعليمية توسّعك إلى ما بعد ساعاتك. والحقيقة المهنية الفجة: المعمارى العبقري المنفرد يبلغ سقفه؛ والمعمارى الذى نمى خمسة مهندسين إلى زملاء قادرين على التصميم هو من يدير الممارسة.

التمارين: (1) اعرض ThaiTicket مرتين — نسخة تنفيذية من 5 دقائق ونسخة هندسية من 20 دقيقة — وسجل الاثنين، واستمع بحثًا عن مصطلحات تقنية تتسرب إلى الأولى. (2) أنتج عرض التكلفة المكتوب الكامل (الافتراضات، والنطاق، والمحركات، وسيناريوهات 2×10×). (3) مسرح اعتراضات مع ذكاء اصطناعي: ثلاث جولات — هجوم تكلفة من مدير مالي، وتفضيل حزمة تقنية من مهندس كبير، ولعبة سلطة “صديق المدير التقني” — ودون ما نجح. (4) راجع تصميم مبتدئ كتابةً (يستطيع ذكاء اصطناعي توليد تصميم معيب)، أسئلة أولًا، ودع الذكاء الاصطناعي يقيّم نبرتك.

الإنجاز المرحلي: المواجهة المزدوجة: قدّم العرض التنفيذي وانج من عشرين دقيقة من أسئلة مختلطة عدائية-لكن-عادلة (لجنة ذكاء اصطناعي: مدير مالي، ومهندس متشكك) دون تسرب مصطلحات، أو دفاعية، أو مفاضلة واحدة غير مدوّنة.

الوحدة 14 (الأشهر 9-12): الشهادات والمشاريع الختامية الثلاثة

مسار الشهادات، بجدول زمنية صادقة. الشهادات لا تجعلك معماريًا — الوحدات الثلاث عشرة السابقة تفعل — لكنها تجلب لك المقابلات، والتحضير لها يوصل خدمات المزود بأصابعك. السلم، بإيقاع ساعة يوميًا: **AWS Certified Solutions Architect – Associate (SAA-C03)** 2-3 — شهر؛ شهادة المعمارى الأكثر طلبًا في إعلانات الوظائف، وبعد المراحل 1-3 سيبدو معظمها مراجعة بأسماء خدمات ملحقة؛ خذها أولًا. **AWS Certified Solutions Architect – Professional (SAP-C02)** 4-8 — شهر بعد ال-Associate؛ قائمة على السيناريوهات، وصعبة حقًا، وأقوى سطر منفرد في السيرة الذاتية في هذا الميدان؛ ومادة المرحلة 4 في الهجرة والحسابات المتعددة نصف منهجها. **Azure AZ-305 (Azure Solutions Architect Expert)** — أضفها إن كان سوقك مثقلًا بمنتجات مايكروسوفت (معظم الأسواق المؤسسية ثنائية اللغة على الأقل)؛ وتوقع 2-3 أشهر مع انتقال معرفة AWS بخصم كبير (ابدأ من <https://learn.microsoft.com/en-us/credentials/certifications/azure-solutions-architect>)

TOGAF Foundation (اختيارية، 2-4 أسابيع؛ تطلبها بعض المؤسسات الكبرى؛ وهي تشهد على منهجية المعمارية المؤسسية ومفرداتها لا على مهارة سحابية. التسلسل لهذه الدورة: دراسة SAA بالتوازي مع الشهرين 7-8، والامتحان نحو الشهر 8؛ ثم

إما SA Pro (المسار التقني العميق) أو AZ-305 (مسار الاتساع) حتى الشهر 12، مع إتمام SA Pro بحلول الشهر 14-16 بالإيقاع الصادق.

المشاريع الختامية الثلاثة. كل منها حزمة كاملة بمستوى الملف المهني: طقم مخططات ثلاثي المستويات، وخمسة ADRs فأكثر، وتقدير تكلفة بافتراضاته، ومراجعة Well-Architected ذاتية بنتائجها. خصص 3-4 أسابيع لكل منها. هذه الأعمال الثلاثة، مع دفترك، هي دليلك في المقابلات على "تصميم الحلول من طرف إلى طرف".

المشروع الختامي 1 — الـ greenfield: منصة تجارة إلكترونية تايلاندية لمليون مستخدم. الموجز: سوق إلكترونية، ومليون مستخدم مسجّل، و50 ألف متزامن في ذروة تخفيضات خاطفة، وأولوية للجوال، وتكامل دفع على طراز PromptPay، وامثال لـ PDPA، وإتاحة 99.9%، ووعي بميزانية مرحلة التمويل الأولي. يجب أن يشمل: اختيار نمط بمسار تطوره، واستراتيجية كاش، ومحادثة RTO/RPO مكتوبة حوارًا، وتقدير اقتصاديات وحدة، وخريطة تدفق بيانات للبيانات الشخصية.

المشروع الختامي 2 — الـ brownfield: هجرة شركة on-prem من 40 خادمًا. الموجز: شركة لوجستيات تايلاندية؛ 40 خادمًا (نظام ERP على Oracle، وتطبيق إدارة مستودعات عمره 12 عامًا، ومشاركات ملفات، وActive Directory، وتطبيقات أقسام متنوعة — ولّد الجرد الكامل بذكاء اصطناعي وجمّده)؛ وعقد إيجار مركز البيانات ينقضي خلال 14 شهرًا؛ والمجلس يريد الخروج. يجب أن يشمل: جدول الرءاءات السبع الكامل، وخطة من ثلاث موجات بمنطق التبعيات، وتصميم منطقة الهبوط، وجدولًا زمنيًا لتكلفة فقاعة الهجرة، ومذكرة المدير المالي.

المشروع الختامي 3 — الصعب: تعافي من الكوارث متعدد المناطق لشركة تقنية مالية. الموجز: شركة مدفوعات ملزّمة تنظيميًا بـ $RTO \leq 15$ دقيقة و $RPO \approx 0$ لدفتر الأستاذ، مع قيود إقامة بيانات على بيانات العملاء واختبار failover سنوي بحضور المنظم. يجب أن يشمل: تحليل نشط-خامل مقابل نشط-نشط بالتكاليف، وتصميم النسخ المتماثل مع خريطة الإقامة، وfailover الـ runbook، وخطة التدريب.

قائمة التقييم الذاتي (طبّقها على كل مشروع ختامي، بصدق، في دفترك):

البعد	1 — ليس بعد	3 — متين	5 — وظّفوا هذا الشخص
المتطلبات	قفز إلى الحل	NFRs معلنة ومتتبّعة إلى خيارات التصميم	مثلت المفاضلة موضّع، والتعارضات مرفوعة إلى "الجانب التجاري"، والقرارات مستخرجة
جودة التصميم	نقاط فشل وحيدة باقية؛ ونمط غير ملائم	أنماط صحيحة، و-multi-AZ، وأعطال معالجة	منطق متى-لا معروض؛ ومسار تطور؛ وأبسط تصميم يلبي المتطلبات

البعد	1 — ليس بعد	3 — متين	5 — وظّفوا هذا الشخص
الاعمدة الستة	الاعمدة متجاهلة	كل عمود معالج بوضوح	المراجعة الذاتية وجدت عيوبًا حقيقية — والتصميم رُوجع استجابةً لها
التكلفة	لا أرقام	تقدير مفصّل بافتراضات معلنة	نطاق بمحركاته، واقتصاديات وحدة، وسيناريو 10x، واستراتيجية حجز
التوثيق	مخطط فقط	طقم ثلاثي المستويات + ADRs، بترميز صحيح	يستطيع غريب الإجابة عن “ماذا/كيف/لماذا” من الحزمة وحدها
التواصل	عمل واحد لكل الجماهير	نسختان تنفيذية وهندسية موجودتان	ملخص من خمس جمل يصمد؛ والاعتراضات مُجاب عنها سلفًا كتابةً

سجّل 4 فأكثر في كل بعد في المشاريع الثلاثة كلها — معيّدًا المراجعة حتى تفعل — وتكون قد أكملت الدورة.

عشرة أسئلة مقابلات للمعماريين (مع إجابات قوية)

تمرّن عليها بصوت مسموع؛ فالقوة في بنية كل إجابة، وهي الآن ملكك.

1. “Design a URL shortener / ticketing site / photo app for a million users” (صمّم مختصر روابط / موقع تذاكر / تطبيق صور لمليون مستخدم). الشكل القوي: المتطلبات أولاً، بصوت مسموع (“القراءات تغلب الكتابات؟ هدف الإتاحة؟ الميزانية؟”), فالموضع على المثلث، فهيكّل ثلاثي الطبقات أو serverless-أولاً بـ multi-AZ وكاش وCDN، فتسمية المفاضلات دون أن تُسأل، فالختم برتبة حجم التكلفة ومسار التطور عند 10x. المحاورون ينجحونك على الأسئلة التي تطرحها، لا على الصناديق التي ترسمها.

2. “When would you choose NoSQL over a relational database?” (متى تختار NoSQL بدل قاعدة علائقية؟) “خيارى الافتراضي SQL — لضمانات الصحة والمهارات الشائعة — حتى يتغلب سبب مسمى: توسع أفقي متطرف، أو مخطط مرّن، أو وصول مفتاح-قيمة بميلي ثوانٍ أحادية الرقم. حينها أختار

نكهة NoSQL الملائمة لنمط الوصول، وأكتب الـ ADR متضمنًا ما نتخلى عنه: الـ joins، وبعض دلالات الاتساق.

3. “Explain RTO and RPO, and how they drive design.” (اشرح RTO و RPO وكيف يقودان التصميم.) عرّف الاثنين بإحكام، ثم “هما قراران تجاريان بأثمان أُسّية”، ثم السّلم: نسخ احتياطية ليلية، فنسخ متواصل، فاحتياطي دافئ، فنشط-نشط، والتكلفة ترتفع مع كل درجة — ثم “وظيفتي جعل الجانب التجاري يختار عن علم، ثم التصميم على مقاس الرقم بالضبط — والتدرب عليه، لأن failover غير مختبر أمنيّة.”

4. “Monolith or microservices?” (متراص أم خدمات مصغّرة؟) حكم الوحدة 8، حرفيًا: ابدأ بمتراص معياري؛ واستخرج حين يصل فعلاً ألم توسع الفريق أو الحمل؛ والـ microservices أداة توسيع فرق تتقاضى ضريبة أنظمة موزعة. ونقاط إضافية لجملته “أفضّل تشغيل monolith جيد على نظام موزع سيئ.”

5. “How do you handle a large cloud bill / cost optimization?” (كيف تتعامل مع فاتورة سحابية كبيرة / تحسين التكلفة؟) “الرؤية أولاً — الوسم و showback؛ ثم الحصاد بترتيب الجهد: اقتل الموارد الميته، واضبط الأحجام من القياسات، ودّج التخزين، وجدول نوم بيئات غير الإنتاج، ثم احجز الخط القاعدي الثابت. وأنا أبلغ باقتصادات الوحدة لا بالمجاميع — فالفاتورة المتنامية مع تكلفة-للطلب متراجعة نجاح لا مشكلة.”

6. “How would you migrate a legacy on-prem application?” (كيف تهاجر بتطبيق on-prem موروث؟) “قيّم قبل النقل — جرد، وتبعيات، وراء لكل workload من الرءات السبع؛ وابن منطقة الهبوط قبل الموجة الأولى؛ وموجات الأسهل أولاً بعمليات قطع متمرّن عليها وخطط تراجع؛ ولقاعدة بيانات جواهر التاج، قطع قائم على النسخ المتواصل مفضّل على التوقف الذي وقّع عليه الجانب التجاري. وأيضًا: تحذير فقاعة الهجرة الصادق مقدّمًا.”

7. “How do you secure a cloud architecture?” (كيف تؤمّن معمارية سحابية؟) امش عبر الطبقات: الهوية (حد أدنى من الامتيازات، و MFA، ولا root يومي)، فالشبكة (subnets خاصة، وتجزئة، وأصغر سطح هجوم)، فالبيانات (تشفير أثناء التخزين/النقل، وتصنيف، وإقامة)، فالكشف (سجلات تدقيق، وتنبيهات) — ثم “وبالتصميم، لا بالتثبيت اللاحق — فأرخص ضابط أمني هو القرار المعماري الذي يزيل الخطر كليًا، كألا تلمس بيانات البطاقات الخام أبدًا.”

8. “Tell me about a design decision you got wrong.” (حدثني عن قرار تصميمي أخطأت فيه.) إنهم يختبرون أنا لا التاريخ. الشكل: مثال حقيقي (من المشاريع الختامية)، فما كنت تعتقده، فما قاله الواقع، فمراجعة ما بعد الحادث، والـ ADR المحدث، والنمط الذي صرت تتفقده. المعماري العاجز عن إنتاج هذه الإجابة معماري لم يُراجع قط.

9. “How do you explain a complex technical decision to a non-technical executive?” (كيف تشرح قرارًا تقنيًا معقدًا لتنفيذي غير تقني؟) “القرار ورقم العمل أولاً، والآلية عند الطلب فقط؛ ومخطط السياق لا الحاويات؛ والمخاطر بلغة الإيرادات-والامتثال-والسمعة؛ وأحضر القرارين اللذين أحتهما منهم، مكتوبين خيارات بأثمانها — فالتنفيذيون يقررون بين خيارات؛ ولا يوافقون على ألغاز.”

10. “An engineer strongly disagrees with your design. What do you do?” (مهندس يخالف تصميمك بشدة. ماذا تفعل؟) “أولاً أدافع عن رأيه بأمانة وبصوت مسموع — فقد يكون محقًا، والمراجعة التي تغيّر تصميمي هي المراجعة وهي تعمل. وإن كان الأمر متقاربًا حقًا، فالمعايير تقرر (ملاءمة الـ NFRs، ومهارات الفريق، وتكلفة الخروج)، لا الأقدمية — وأدع تفضيل المنقذ يفوز في حالات التعادل، لأن الالتزام رخيص بذلك الثمن. وفي الحالتين يذهب القرار والخيار المرفوض إلى الـ ADR، فلا نتجادل فيه مرتين أبدًا.”

منهج الصفحة الواحدة

متى	الوحدة	التركيز	البرهان الخارجي
الأسبوعان 1-2	1	مراجعة السحابة؛ مثلث المفاضلة؛ الـ NFRs	بدء دفتر التصميم
الأسبوعان 3-4	2	الحوسبة/التخزين/قاعدة البيانات/الشبكة بوصفها قرارات	—
الأسبوعان 5-6	3	قراءة المخططات ورسمها؛ طلاقة الثلاثي الطبقات	إنجاز السبورة في 15 دقيقة
الأسبوعان 7-8	4	الموثوقية: multi-AZ، والتوسع التلقائي، RTO/RPO	—
الأسبوعان 9-10	5	الأمن: الحد الأدنى من الامتيازات، وانعدام الثقة، والتجزئة	—
الأسبوعان 11-12	6	الأداء والتكلفة: الكاش، وCDN، والحجوزات، واقتصاديات الوحدة	تصميم مسعّر في الحاسبة
الأسبوعان 13-14	7	التميز التشغيلي والاستدامة؛ تمرين الأعمدة الستة	مراجعة تجريبية بستة أعمدة

متى	الوحدة	التركيز	البرهان الخارجي
الأسابيع 15-17	8	الأنماط: المتراص/ الخدمات المصغّرة، والمدفوع بالأحداث، وال- serverless	—
الأسابيع 18-20	9	الأنماط: بحيرة/مستودع البيانات، وتعدد المناطق، والهجين	تحدي الكتالوج
الأسابيع 21-23	10	الهجرة: الرءاءات السبع، والموجات، ومناطق الهبوط	الهجرة الورقية
الأسابيع 24-26	11	القيود: PDPA/GDPR، والموروث، واقتصاديات الارتها	تحدي القيود
الشهران 7-8	12	ال ADRs، وأطقم المخططات، وإدارة مراجعات Well- Architected	حزمة ThaiTicket؛ امتحان SAA-C03 (تحضير 2-3 أشهر)
الشهران 8-9	13	العرض، وتكليف المقترحات، والاعتراضات، والإرشاد	المواجهة المزدوجة مسجّلة
الأشهر 9-12	14	المشاريع الختامية 1-3	الملف المهني مكتمل؛ SA Pro (4-8 أشهر) أو AZ-305 جارية

كلمة ختامية من معلمك. بعد ستة وعشرين أسبوعًا، صرت قادرًا على التصميم؛ وبعد اثني عشر شهرًا، صرت قادرًا على الممارسة — وثمة فرق، وهو الفرق الذي بُنيت هذه الدورة لسدّه. العادات الآن هي المسيرة: ارسـم قبل أن تجادل، واكتب ال ADR يوم تقرر، وسعّر ما تقترح، وسمّ المفاضلة قبل أن يسأل أحد، ونمّ المهندسين حولك حتى تعمّر معاييرك أطول من حضورك في الغرفة. المعمارية هي الوظيفة التقنية النادرة التي تتحسن كلما تقدمت فيها عمرًا، لأن مادتها الخام الحكم السديد، والحكم السديد يتراكم. واصل الدفتر. وبعد أربعين تصميمًا من الآن، لن تحتاج إلى هذه الدورة — بل ستصحّحها.

الأوصاف الوظيفية وتعريفات الدور المستخدمة في جدول موائمة المتطلبات (استُرجعت في أغسطس 2026): [قالب KORE1 للوصف الوظيفي لمعماري السحابة](#) · [وصف 4 Corner Resources الوظيفي لمعماري السحابة](#) · [مسؤوليات Solution Architect لدى Microsoft في إطار Azure Well-Architected](#) · [إعلان Arpio لوظيفة Pre-Sales Solutions Architect – AWS](#) · [إعلان Intel لوظيفة Solution Architect – Enterprise \(غير Built In\)](#) · [إعلان Halliburton لوظيفة Cloud Domain Architect](#). تقديرات مدد التحضير للشهادات: [استطلاع CBT Nuggets لمدة دراسة SAA-C03 وإرشادات Whizlabs للتحضير لـ SAP-C02](#)؛ وتعريفات الشهادات من [Microsoft Learn \(AZ-305\)](#) و [إطار AWS Well-Architected](#). مجلد مرافق لدورة [The Cloud Leader Course](#) في سلسلة B4LCILC.

دروس جديدة — طبعة أغسطس 2026

في أغسطس 2026 أُضيفت دروس جديدة إلى الدورات التفاعلية الثلاث على [b4lcilc.org](#) — دورة قائد السحابة، ودورة مهندس السحابة، ودورة معماري السحابة. وهذا الملحق يجعل طبعة القراءة مواكبة لما استجدّ، عبر تكتيفات أمينة لتلك المادة الجديدة، وبالصوت نفسه الذي كُتب به سائر هذا الكتاب.

كلمة عمّا بين يديك. هذه الصفحات هي طبعة القراءة: الأفكار، والاستعارات، والجمل التي تستحق أن تُقال بصوت مسموع. أما النسخ التفاعلية الكاملة على الإنترنت فتحمل أكثر من ذلك بكثير — فكل درس جديد يأتي مصحوبًا باختبارات قصيرة تتحقق من فهمك قبل أن تسمح لك بالمتابعة، ومختبرات عملية (من بينها terminal تدريبي يتصرف مثل الحقيقي تمامًا لكنه لا يستطيع أن يكلّفك مألًا)، ومقاطع فيديو، وتدريبات مفردات، و— في المادة الختامية — تمارين تصميم موجهة بقوائم متطلبات وإجابات نموذجية محلولة. اقرأ هذه التكتيفات لتستقر صورة كل فكرة في رأسك؛ ثم، متى استطعت، اجلس إلى النسخ التفاعلية لترسّخ المعرفة.

تنقسم الإضافات إلى ثلاث مجموعات. لدورة **قائد السحابة**: مدخل لطيف إلى الـ terminal، وقانون الخصوصية الخاص بتايلاند نفسها، ودليل القائد إلى مقترحات الذكاء الاصطناعي وفواتيره، وتمارين ختامي يجمع الدورة بأكملها في صفحة واحدة. ولدورة **مهندس السحابة**: الحوسبة الـ serverless — النمط الذي يحوّل السكريبتات التشغيلية الصغيرة إلى دوال لا يكلّف تشغيلها شيئًا. ولدورة **معماري السحابة**: حساب وعود الإتاحة، ونمط التوليد المعزّز بالاسترجاع (Retrieval-Augmented Generation) الذي بات العملاء يسألون عنه أسبوعيًا، وتمارين تقدير تكلفة كامل — تسعير تصميم قبل أن يبنيه أحد.

لدورة معماري السحابة

حساب التسعات — SLA وSLO وميزانيات الخطأ

تُقاس الإتاحة بالتسعات، وأول ما يفعله المعمارى بالتسعة هو تحويلها إلى دقائق — لأن الدقائق هي ما يستطيع العمل التجاري الإحساس به. الشهر المكوّن من 30 يومًا يحوي 43,200 دقيقة:

التعطّل المسموح به شهريًا	الإتاحة
نحو 43 دقيقة	99.9% ("ثلاث تسعات")
نحو 4.3 دقائق	99.99% ("أربع تسعات")
نحو 26 ثانية	99.999% ("خمس تسعات")

خمس تسعات — الرقم الذي يحب المديرون التنفيذيون قوله بصوت عالٍ — تتجاوز قدرة أي استعادة يديرها بشر؛ فإما أن يكون تجاوز الفشل (failover) تلقائيًا وإما أن يكون الوعد خيالًا. وكل تسعة إضافية تكلف هندستها نحو عشرة أضعاف؛ والتحويل إلى دقائق هو طريقتك لجعل صاحب المصلحة يختار عن بيّنة.

ثلاثة اختصارات تحمل الحوار. الـ SLI — مؤشر مستوى الخدمة (Service-Level Indicator) — قياس. والـ SLO — هدف مستوى الخدمة (Service-Level Objective) — الهدف الداخلي المرسوم على ذلك القياس. والـ SLA — اتفاقية مستوى الخدمة (Service-Level Agreement) — العقد الخارجي، ومعه مال يُدفع عند الإخلال. والانضباط: الـ SLO أشد صرامةً دائمًا من الـ SLA — فأنت تصوّب داخل الخط الذي تتقاضى أجرًا على حراسته — والـ SLI يقيس ما يعيشه المستخدمون: فالخادم الذي يعمل بينما يعيد أخطاءً هو خادم معطلّ بكل معنى يهّم.

ميزانية الخطأ (error budget) هي ما يجعل أهداف SLO قابلة للتشغيل: الفجوة بين الكمال والهدف، تُعامل موردًا يُنفق منه. فهدف SLO عند 99.9% يمنح 43 دقيقة فشل في الشهر؛ تُنفق منها عمليات النشر والتجارب؛ وحين تنفذ، يتوقف الفريق عن الشحن ويعمل على الموثوقية — فينهي بذلك أقدم جدال في البرمجيات، الشحن مقابل الاستقرار، بإحلال الحساب محل الرأي.

والآن الفخ: الإتاحة تُضرب بعضها في بعض على طول السلسلة. ثلاث خدمات على التوالي، كل منها بصدق عند 99.9%، تتراكم إلى $99.7\% \approx 0.999 \times 0.999 \times 0.999$ — أي نحو 130 دقيقة في الشهر، ثلاثة أضعاف ما تعترف به أي خدمة منفردة. أما التكرار الاحتياطي فيعمل في الاتجاه المعاكس: نسختان مستقلتان عند 99.9% لا تفشلان معًا إلا $0.1\% \times 0.1\%$ من الوقت — ولهذا يمثل التوزيع متعدد المناطق قيمةً ممتازة. والجملة الجديرة بمقابلة عمل: لا تستطيع أن تعد بإتاحة تفوق حاصل ضرب اعتمادياتك المترامنة، فإما أن تقصّر السلسلة وإما أن تخفّض الوعد.

المعمارية في عصر الذكاء الاصطناعي — RAG وأقاربه

بحلول 2026 دخل كتالوج الأنماط مُدخّل جديد يسأل عنه العملاء أسبوعيًا: ما هو، ومتى يستحق كلفته، ومتى سيجرك. فواجهة API لنموذج لغوي كبير (Large Language Model — LLM) — مثل Amazon Bedrock أو Azure OpenAI Service — هي، معماريًا، اعتمادية مكلفة وبطيئة وعديمة الحالة. تُعابير بال token — شظية كلمة تعادل نحو أربعة أحرف. تجيب في ثوانٍ لا ميلي ثوانٍ، ولا تتذكر شيئًا بين الاستدعاءات، وتخطئ أحيانًا بثقة تامة — مكوّن ذو أنماط فشل غير مألوفة، يستحق نفس المهلات وإعادة المحاولات وقواطع الدارة وإنذارات الميزانية التي تستحقها أي اعتمادية، لأن حلقة استدعاءات منفلة تحرق مالا حقيقيا بالساعة.

النموذج لا يعرف مستنداتك؛ والجواب المعياري هو التوليد المعزّز بالاسترجاع (Retrieval-Augmented Generation — RAG): بدلًا من تعليم النموذج بياناتك، تبحث عن الحقائق ذات الصلة وقت السؤال. تُقطّع المستندات إلى مقاطع، يحوّل كل منها إلى تضمين (embedding) — قائمة أرقام تحدد موضع معناه في الفضاء — تُخزّن في قاعدة بيانات متجهية (vector database) (pgvector على AWS، و AI Search على Azure)، تعيد أقرب المقاطع. تُلصق تلك المقاطع في ال prompt: فالنموذج لا يستشير بياناتك؛ بل يلخّص المقطّعات التي جلبتها أنت له.

على السبورة، RAG خطا أنابيب: مسار استيعاب (ingestion path) يعمل مسبقًا (المستندات تُقطّع وتُضمّن وتُخزّن) ومسار استعلام (query path) يعمل مع كل طلب — نمطان موجودان أصلاً في الكتالوج، ولهذا يستوعبه المعمارون في أمسية واحدة.

والمقايضات هي الثلاث المعيارية نفسها، وقد ازدادت حدّة: التكلفة (كل استعلام يكلف مالا حقيقيا — خزّن المتكرر في cache، ووجّه الأسئلة السهلة إلى نموذج أرخص)، والتأخير (ثوانٍ لا ميلي ثوانٍ — بُتّ الجواب تدفقيا)، والخصوصية (مستنداتك تغادر حدودك؛ وإذا تجاهلت قاعدة البيانات المتجهية ضبط الوصول على مستوى المستند، فسيلخّص روبوت المحادثة بكل سرور جدول الرواتب لأي شخص يسأل). واعرف متى لا تبنيه: كثير من طلبات "نحتاج روبوت محادثة" ليست إلا مربع بحث يرتدي الموضة، والأجوبة الدقيقة القابلة للتدقيق — المجاميع والأرصدة والدفاتر — عمل خاطئ يُوكّل إلى نظام يصيب 97% من الوقت. والجملة الجديرة بمقابلة عمل: RAG مسألة بحث مدبّسة إلى API تلخيص.

تقدير تكلفة تصميم قبل أن يبينه أحد

وحدة ركيّزي الأداء والتكلفة في الدورة تُختتم الآن بتمرين تصميم كامل: التقدير تحت النار. المطلوب: تريد القيادة تقديرًا للتكلفة الشهرية لمنتج إدارة مستندات قبل الموافقة على البناء. عبء العمل — 200,000 مستخدم مسجّل، منهم 40,000 نشط يوميًا، يجري كل منهم نحو 25 طلب API في اليوم ويرفع في المتوسط 2 ميغابايت من المستندات شهريًا، تُحفظ سبع سنوات. التصميم: طبقة API موزونة الحمل عبر منطقتين، وقاعدة بيانات PostgreSQL مُدارة متعددة المناطق، وتخزين كائنات، وتصدير ليلي بحجم 50 غيغابايت إلى شريك، وبيئتا تطوير وتهيئة منفصلتان. المهمة: تقدير شهري في صفحة واحدة بأرقام مقرّبة، مع كتابة الافتراضات قبل ظهور أي أسعار، وإظهار حساب السعة (المستخدمون ← الطلبات في الثانية ← عدد النسخ)، وسيناريوهات اليوم / الضعفين / عشرة الأضعاف.

قائمة التحقق في التمرين تعلّم الانضباط: سعر الحوسبة من ذروة الحمل لا من متوسطه؛ واحسب النسخة الاحتياطية الجاهزة لقاعدة البيانات؛ ونمذج التخزين نموًا على مدى سبع سنوات مع طبقات دورة الحياة؛

واحسب حركة الخروج (egress) بدل تخمينها — البند الذي تنساه معظم التقديرات وهو الذي يهيمن عليها أكثر من غيره؛ وتذكّر بوابات NAT (ترجمة عناوين الشبكة — Network Address Translation)، والبيئات غير الإنتاجية، والنسخ الاحتياطية، والمراقبة، وخطة الدعم؛ وطبّق خصومات الالتزام على خط الأساس المستقر وحده؛ واختم بنطاق تقديري مع تسمية محركاته إلى جانب فحص سلامة باقتصاد الوحدات — التكلفة لكل مستخدم نشط في الشهر. فالرقم الدقيق الواحد يكون خاطئًا يوم يُنطَق به.