

دورة مهندس السحابة (The Cloud Engineer Course)

من الصفر المعرفي إلى مهندس سحابة جاهز للتوظيف — دورة من سلسلة B4LCILC
13 أغسطس 2026

دورة مهندس السحابة (The Cloud Engineer Course)

من الصفر المعرفي إلى الجاهزية الحقيقية للتوظيف كمهندس سحابة (cloud engineer).

كيف تعمل هذه الدورة

أنت تتدرب لكي تكون مهندس سحابة: الشخص الذي يبني الخوادم بالشفيرة، ويُبقي الأنظمة حية في الثالثة فجراً، ويؤتمت العمل الممل بعيداً، ويستطيع أن ينظر إلى نظام معطوب فيكتشف السبب بهدوء. تلك حرفة عملية، وهذه الدورة تعاملها كذلك. ستقضي في الكتابة على لوحة المفاتيح وقتاً أطول بكثير مما تقضيه في القراءة — كل وحدة فيها مختبرات (labs)، والمختبرات هي الدورة. القراءة عن السحابة تبني القدرة على التعرف؛ أما البناء في السحابة فيبني مسيرة مهنية.

تتكوّن الدورة من خمس مراحل. المرحلة 0 (الأسابيع 1-4): الأساسات — كيف تعمل الحواسيب والشبكات فعلاً، ولينكس (Linux)، نظام التشغيل الذي تعمل عليه السحابة. المرحلة الأولى (الأسابيع 5-12): جوهر السحابة، عملياً — خدمات AWS الخمس التي تظهر في كل وصف وظيفي، تتعلمها كمختبرات تبنيها ثم تفكّكها. المرحلة الثانية (الأسابيع 13-20): الأتمتة — البرمجة النصية (scripting) و Git و Terraform و CI/CD والحاويات: المهارات التي تفصل المهندس عن ناقر الـ console. المرحلة الثالثة (الأسابيع 21-28): العمليات (operations) — المراقبة والحوادث والأمن والتكلفة: المهارات التي تتقاضى أجرك عليها فعلاً. المرحلة الرابعة (الأشهر 8-12): الجاهزية للتوظيف — ثلاثة مشاريع للملف المهني، وثلاث شهادات، وتدريب على المقابلات.

قواعد تسري على الدورة كلها: ادرس من ساعة ونصف إلى ساعتين يوميًا، ستة أيام في الأسبوع — فالاتمرارية تتفوق على الكثافة، وهذه حرفة تُتعلّم بالترار اليومي، كآلة موسيقية. كل وحدة تنتهي بـ جدول

مفردات (الكلمات التي يجب أن تمتلكها)، و**تمارين** “قلها بصوت مسموع” (جمل تكررهما حتى تصبح طبيعية — فمقابلات العمل تُجرى كلامًا لا كتابة)، و**تمارين عملية**، و**إنجاز مرحلي (Milestone)** يحرس التقدم: لا تنتقل قبل أن تتقنه، لأن كل مرحلة تقف فوق التي قبلها. ومنذ الأسبوع الأول، احتفظ بـ **دفتر هندسة (Engineering Journal)**: كل مختبر، وكل رسالة خطأ، وكل إصلاح، في فقرة واحدة صادقة. بحلول الشهر العاشر يصبح ذلك الدفتر المادة الخام لملفك المهني وقصص مقابلاتك.

ووعد واحد في المقابل: لا شيء في هذه الدورة يفترض أنك تعرف شيئًا. كل مصطلح يُعرّف عند أول ظهور له. إذا كنت تستطيع استخدام متصفح ويب وكنت مستعدًا لكتابة أوامر ستبدو غريبة خلال الأسبوعين الأولين، فأنت تملك كل المتطلبات المسبقة الموجودة أصلًا.

الوصف الوظيفي الذي تتدرّب من أجله

في أغسطس 2026 جمعنا أوصافًا وظيفية حقيقية لوظيفة Cloud Engineer من قوالب أصحاب العمل وأدلة التوظيف (Arc.dev وWiz وDevsData وXOPA وBetterteam — القائمة الكاملة في المصادر). انزع أسماء الشركات وستجد المتطلبات نفسها تتكرر مرة بعد مرة. هذا الجدول هو عقدك مع الدورة — كل بند يطلبه صاحب عمل يُقرّن بالمرحلة التي تعلّمه:

ما تطلبه الأوصاف الوظيفية الحقيقية (بنصّها تقريبًا)		أين تعلّمه هذه الدورة
«معرفة بأنظمة تشغيل Linux/Unix»	المرحلة 0، الوحدة 1	
«خبرة في الشبكات السحابية بما فيها الـ VPCs والـ subnets وموزّعات الحمل وDNS»	المرحلة 0 الوحدة 2 + المرحلة 1 المختبر 3	
«خبرة مثبتة في خدمات AWS الأساسية، بما فيها EC2 وS3 وRDS وVPC وIAM»	المرحلة 1 (المختبرات 1-5)	
«تصميم وتطوير ونشر بنية تحتية سحابية باستخدام أدوات البنية التحتية كشيفرة مثل Terraform وCloudFormation»	المرحلة 2، الوحدة 7	
«إتقان لغات برمجة نصية مثل Python أو Bash أو PowerShell أو Go»	المرحلة 2، الوحدة 6	
«بناء وصيانة خطوط أنابيب CI/CD بأدوات مثل Jenkins أو GitLab CI أو GitHub Actions»	المرحلة 2، الوحدة 8	
«خبرة في تقنيات الحاويات بما فيها Docker وKubernetes»	المرحلة 2، الوحدة 8	

ما تطلبه الأوصاف الوظيفية الحقيقية (بنصّها تقريبًا)	أين تعلّم هذه الدورة
«مراقبة صحة البنية التحتية وأدائها باستخدام أدوات مراقبة سحابية أصيلة»	المرحلة 3، الوحدة 9
«المشاركة في الاستجابة للحوادث، بما فيها تحليل السجلات»؛ «مهارات استكشاف الأخطاء والتحليل»	المرحلة 3، الوحدة 10
«تطبيق وفرض ضوابط أمنية تشمل التشفير وإدارة الهوية والوصول»؛ «وصول بالحد الأدنى من الامتيازات»	المرحلة 1 المختبر 5 + المرحلة 3 الوحدة 10
«إدارة تكاليف السحابة عبر ضبط أحجام الموارد وتطبيق الـ auto-scaling ووسم الموارد»	المرحلة 3، الوحدة 10
«صيانة واختبار وتنفيذ إجراءات التعافي من الكوارث»	المرحلة 3 + مشروع الملف المهني 3
«يُفضّل حَمَلَة شهادات AWS»	جدول شهادات المرحلة 4 (CCP ثم SAA ثم Terraform Associate)
«تقديم إرشاد تقني وتوثيق»؛ «مهارات تواصل وتعاون جيدة»	الدفتر + الـ runbooks + تحضير مقابلات المرحلة 4

سياق الرواتب، لتعرف ما الذي تعمل نحوه: تتجمّع رواتب مهندسي السحابة في الولايات المتحدة حول وسيط يقارب 104,000 دولار، بنطاق نموذجي بين 85 و140 ألف دولار، وتُعلن أدوار AWS العليا/المتخصصة بأعلى من ذلك بكثير. الأدوار المبتدئة موجودة بأسماء كثيرة — cloud support associate و junior cloud engineer و cloud operations engineer — وهذه الدورة تستهدف متطلباتها بالضبط.

المرحلة 0 — الأساسات (الأسابيع 1-4)

الوحدة 1 (الأسبوعان 1-2): كيف تعمل الحواسيب، ولينكس — لغة الخوادم

الفكرة الكبرى: الخادم (server) ليس سوى حاسوب وظيفته خدمة حواسيب أخرى. الحاسوب المحمول أمامك والآلات التي تشغّل Netflix يختلفان في الحجم والموثوقية لا في النوع: فكلاهما CPU (الجزء الذي ينجز العمل)، وذاكرة/RAM (مساحة عمل قصيرة الأمد سريعة، تُمسح عند إعادة التشغيل)، وقرص (disk) (تخزين طويل الأمد بطيء يبقى بعد إعادة التشغيل)، وبطاقة شبكة (الوصلة إلى كل ما عداه)، ينسّق بينها جميعًا نظام تشغيل (OS). حاسوبك يعمل غالبًا بويندوز أو macOS. أما الخوادم فتعمل في أغليتها الساحقة بـ

لينكس (Linux) — نظام تشغيل حر مفتوح المصدر، مستقر وقابل للبرمجة النصية ويُتحكَّم به كليًا بأوامر مكتوبة. وهذا الجزء الأخير هو بيت القصيد: لا يمكنك أتمتة نقرات الفأرة، لكن يمكنك أتمتة الأوامر، وهندسة السحابة هي الأتمتة. لذا فإن طلائقتك الأولى هي **الطرفية (terminal)**.

الطرفية، بلا غموض. الطرفية (أو “الصدفة shell” — والبرنامج بداخلها عادة هو **Bash**) محادثة نصية مع الحاسوب. تكتب أمرًا؛ فيجيب. هذا كل شيء. علامة \$ التي سترها في الأمثلة هي **المِحَّت (prompt)** — الصدفة تقول “دورك.” كل شيء في لينكس ملف، والملفات تعيش في شجرة واحدة تبدأ من الجذر /، ومجلدك الشخصي هو home/yourname/ (ولقبه ~). **والمسار (path)** هو عنوان الملف في تلك الشجرة: .home/anna/notes.txt/

دبّر لنفسك لينكس تتمرّن عليه (اختر خيارًا واحدًا، عشر دقائق): على ويندوز، ثبّت **WSL** (نظام ويندوز الفرعي للينكس — لينكس Ubuntu حقيقي داخل ويندوز: <https://learn.microsoft.com/en-us/windows/wsl/install>)؛ وعلى Mac، تطبيق Terminal المدمج قريب بما يكفي للبدء (فـ macOS ابن عم Unix)؛ أو انتظر الأسبوع الخامس حين ستستأجر خادم لينكس حقيقيًا من AWS مجانًا. **WSL** هو الجواب الأفضل لمعظم الناس.

أهم 25 أمرًا — هذا الجدول هو الأسبوعان 1-2. اكتب كل واحد منها، مرات كثيرة:

الأمر	ما يفعله	مثال
pwd	اطبع دليل العمل — “أين أنا؟”	pwd
ls	اعرض الملفات هنا (-l تفصيل مطوّل، -a مع المخفية)	ls -la
cd	غيّر الدليل — تنقّل في الشجرة	cd /var/log
mkdir	أنشئ دليلًا (مجلدًا)	mkdir projects
touch	أنشئ ملفًا فارغًا	touch notes.txt
cp	انسخ ملفًا (-r للمجلدات)	cp a.txt backup.txt
mv	انقل أو أعد التسمية	mv old.txt new.txt
rm	احذف — نهائيًا، بلا سلة مهملات. عامله باحترام.	rm notes.txt
cat	اطبع محتوى الملف كاملاً	cat notes.txt
less	افقرأ ملفًا طويلًا صفحة صفحة (q للخروج)	less /var/log/syslog

الأمـر	ما يفعله	مثال
head / tail	أول / آخر أسطر الملف. tail -f يراقب سجلًا حيًا — كلاسيكية تشغيلية	tail -f app.log
grep	ابحث في النص عن نمط — أكثر أوامر التشغيل استخدامًا على الإطلاق	grep "ERROR" app.log
find	اعثر على الملفات بالاسم/الحجم/ العمر	"find / -name "*.conf
echo	اطبع نصًا (كثيرًا ما يذهب إلى ملفات أو متغيرات)	"echo "hello
nano	محرّر نصوص ودود داخل الطرفية	nano notes.txt
man	دليل أي أمر (q للخروج)	man grep
sudo	نقِّذ أمرًا واحدًا بصفة المدير الكلي القدرة ("root"). باحترام.	sudo apt update
apt	ثَبِّت/حَدِّث البرمجيات (عائلة Ubuntu/Debian)	sudo apt install htop
chmod	غيّر صلاحيات الملف	chmod 644 notes.txt
chown	غيّر مالك الملف	sudo chown anna file
ps	اعرض العمليات الجارية (ps aux لجميعها)	ps aux
top (أو htop)	لوحة حية للمعالج/الذاكرة/العمليات — "لماذا الخادم بطيء؟" تبدأ هنا	top
df -h / du -sh	المساحة الحرة على القرص / المساحة التي يستهلكها مجلد — "القرص ممتلئ" تبدأ هنا	df -h
ssh	سجِّل الدخول إلى طرفية آلة أخرى عبر الشبكة — هو الباب الأمامي لمهندس السحابة	ssh anna@server-ip

الأمر	ما يفعله	مثال
curl	أرسل طلب ويب من الطرفية — “هل الموقع يعمل؟”	curl https://example.com

الصلاحيات، في 60 ثانية. لكل ملف مالك ونمط مثل `rw-r-xr-x` : ثلاث ثلاثيات — المالك، والمجموعة، والجميع — من قراءة (read) وكتابة (write) وتنفيذ (execute). بالأرقام: $r=4$ و $w=2$ و $x=1$ ، فيكون معنى `chmod 755 script.sh`: “المالك يفعل كل شيء ($1+2+4=7$)؛ وكل من عداه يقرأ ويشغّل ($1+4=5$)”. حين “لا يملك برنامج الصلاحية”، فهذا هو النظام يقول لا — وقد صرت الآن قادرًا على قراءة السبب.

SSH، في 60 ثانية. يفتح ssh طرفية بعيدة آمنة على آلة أخرى — من حاسوبك إلى خادم في فرجينيا كأنك تجلس أمامه. بدل كلمات المرور، يستخدم المحترفون زوج مفاتيح (key pair): مفتاح خاص (ملف سري على حاسوبك لا يُشارك أبدًا) ومفتاح عام (يوضع على الخادم). يتطابقان كالمفتاح والقفل. كل خادم AWS تطلقه في المرحلة الأولى سيسلمك هذا بعينه.

العمليات والخدمات: العملية (process) برنامج قيد التشغيل؛ **والخدمة (service)** (أو “daemon”) عملية تعمل إلى الأبد في الخلفية — خوادم الويب وقواعد البيانات. الأمر `systemctl status nginx` يسأل لينكس: “هل خدمة nginx سليمة؟” — جملة ستكتبها مهنيًا لسنوات.

المفردات:

المصطلح	التعريف
الخادم (server)	حاسوب وظيفته خدمة حواسيب أخرى. في السحابة، حاسوب تستأجره.
CPU / RAM / القرص	العامل، ومساحة العمل المؤقتة السريعة (تُمسح عند إعادة التشغيل)، والتخزين الدائم البطيء.
نظام التشغيل (OS)	البرمجية التي تدير الآلة وتستضيف البرامج. الخوادم تعمل بلينكس.
Linux / التوزيعة (distribution)	نظام الخوادم الحر مفتوح المصدر. “التوزيعة” (Ubuntu و Amazon Linux و Debian) نكهة مغلفة واحدة منه.
الطرفية / الصدفة / Bash	الواجهة النصية لنظام التشغيل / البرنامج الذي يفسّر أوامرك / اسم الصدفة القياسية.
المِحَتَّ (prompt)	علامة \$ — الصدفة في انتظار أمرك.

المصطلح	التعريف
الدليل / المسار	مجلد / العنوان الكامل لملف في الشجرة الواحدة التي تبدأ من /.
الجذر root (بمعنيين)	قمة شجرة الملفات (/) والمستخدم المدير الكلي القدرة. السياق يحدد المقصود.
sudo	“Superuser do” — نَقِّذْ أمرًا واحدًا بصلاحيات المدير.
الصلاحيات (rwx)	قواعد لكل ملف حول من يقرأ ومن يكتب ومن يَنْقُذْ — تُعرض ثلاثيات للمالك/المجموعة/الجميع.
العملية / الخدمة (daemon)	برنامج قيد التشغيل / برنامج يعمل إلى الأبد في الخلفية (خوادم الويب، قواعد البيانات).
SSH / زوج المفاتيح	تسجيل دخول بعيد آمن إلى طرفية آلة أخرى / ملفا المفاتيح الخاص+العام اللذان يحلان محل كلمات المرور.
السجل (log)	ملف نصي تكتب فيه البرمجيات ما حدث — أول مكان تنظر فيه حين يتعطل أي شيء.
مدير الحزم (package manager)	مُثَبِّت نظام التشغيل (apt, yum) — برمجيات بالأوامر لا بصفحات التنزيل.

فيديوهات هذه الوحدة (الروابط متحقَّقة منها):

الفيديو	القناة	المدة	الرابط
Linux Operating System — Crash Course for Beginners	freeCodeCamp	~2 hr	https://www.youtube.com/watch?v=ROjZy1WbCIA
WSL install guide (مرجع، ليس فيديو)	Microsoft Learn	—	https://learn.microsoft.com/en-us/windows/wsl/install

قلها بصوت مسموع حتى تصبح طبيعية: “Let me SSH in and check the logs.” (دعني أدخل عبر SSH وأفحص السجلات.) · “Grep the log for the error, then tail -f it while we retry.” (ابحث بـ grep في السجل عن الخطأ، ثم تابعه بـ tail -f بينما نعيد المحاولة.) · “It’s a permissions problem — who owns the file and”

Check top — is it CPU, “what’s the mode?memory, or disk” (افحص top — هل هي مشكلة معالج أم ذاكرة أم قرص؟)

التمارين: (1) في طرفية لينكس لديك، ابنِ شجرة مشروع صغيرة بـ `mkdir` و `touch`، وانسخ أشياء وانقلها، ثم احذفها — وأنت تسرد كل أمر بصوت مسموع. (2) أنشئ `hello.sh` يحتوي `echo "hello from $(whoami)"`، واجعله قابلاً للتنفيذ بـ `chmod +x`، وشغله بـ `./hello.sh`. (3) شغل `tail -f` على ملف سجل (على `Ubuntu: sudo tail -f /var/log/syslog`) وراقب الأسطر وهي تصل. (4) الدفتر: اشرح لصديق متخيّل لماذا تعمل الخوادم بـ لينكس، في ثلاث جمل.

الإنجاز المرحلي: دون ملاحظات، تستطيع التنقل إلى أي مكان في شجرة الملفات، وإنشاء الملفات ونسخها ونقلها وحذفها، وشرح `chmod 755`، واستخدام `grep` للعثور على كلمة في ملف. إن تطلّب أيّ من ذلك بحثًا عن معلومة، فامكث هنا يومين إضافيين. هذه الوحدة حاملة لكل ما يُبنى بعدها.

الوحدة 2 (الأسبوعان 3-4): أساسيات الشبكات + حسابك على AWS

الفكرة الكبرى: الشبكة حواسيب تتبادل مطاريّف معنونة. كل آلة تحصل على عنوان IP (مثل 172.31.8.14 — عنوان شارع رقمي). تُقطّع البيانات إلى حزم (packets) (مطاريّف) وتُوجّه قفزةً قفزةً نحو العنوان الوجهة. وعند الوصول، يحدّد رقم المنفذ (port) أي برنامج هو المقصود بالمطروف — المبنى نفسه، وآلاف الأبواب المرقّمة: المنفذ 22 هو SSH، و80 هو الويب غير المشفّر (HTTP)، و443 هو الويب المشفّر (HTTPS)، و5432 هو PostgreSQL. “افتح المنفذ 443” تعني “اسمح بالمطاريّف المعنونة إلى الباب 443”.

خاص مقابل عام: منزلك وكل شبكة سحابية يعيدان استخدام نطاقات الـ IP الخاصة (10.x.x.x) و (172.16-31.x.x و 192.168.x.x) التي لا تعمل إلا داخل الشبكة المحلية؛ أما الـ IP العام فيمكن بلوغه من الإنترنت كله. هذا الانقسام هو صخرة الأساس في أمن السحابة: ما لا يحتاج إلى مواجهة الإنترنت لا يحصل على عنوان عام أصلاً.

DNS — دليل هاتف الإنترنت. البشر يستخدمون الأسماء (example.com)؛ والحزم تحتاج إلى أرقام. DNS (نظام أسماء النطاقات) يترجم: تسأل آلتك خادم DNS “ما الـ IP لـ example.com؟”، فتحصل على الرقم، ثم تتصل. نصف الانقطاعات الغامضة كلها متورط فيها DNS؛ ومزحة الصناعة “إنها دائماً DNS” موجودة لأنها كثيراً ما تكون صحيحة. الأمر `nslookup example.com` يجري الاستعلام يدوياً.

HTTP — كيف يتكلم الويب. يرسل العميل (client) (المتصفح) طلباً — طريقة (method) (GET = اجلب، POST = أرسل) مع مسار — ويرد الخادم بـ رمز حالة (status code): 200 حسناً، و301 انتقل، و403 ممنوع، و404 غير موجود، و500 خطأ خادم، و502/503 “الخادم الذي خلفي معطوب/مثقل”. احفظ هذه الستة؛ فبوصفك مهندساً ستقرأها يومياً. الأمر `curl -I https://example.com` يعرض لك سطر حالة حيّاً وترويسات.

الجدران النارية: قائمة قواعد تقرر أي الحزم يُسمح لها بالمرور، بحسب المصدر والوجهة والمنفذ — “اسمح بـ 443 من أي مكان؛ اسمح بـ 22 من المكتب فقط؛ ارفض الباقي.” في AWS يسمى الجدار الناري الخاص بكل خادم **security group**، وسوء إعداداته هو الثغرة الأمنية رقم 1 عند المبتدئين. **زمن الاستجابة (latency)** (التأخير، بالميلي ثانية) و**عرض النطاق (bandwidth)** (السعة في الثانية) يكملان المفردات: المسافة تولّد ال latency، ولهذا للسحب مناطق حول العالم.

والآن، حسابك على AWS — النصف الثاني من هذه الوحدة. اذهب إلى <https://aws.amazon.com/free> وأنشئ حساب Free Tier (بريد إلكتروني، وهاتف، وبطاقة ائتمان/خصم لإثبات الهوية — البطاقة لا تُحمّل مبالغ تُذكر إن التزمت انضباط التفكير في هذه الدورة). تمنحك **الفئة المجانية (Free Tier)** حصصًا شهرية من الأساسيات (منها 750 ساعة شهريًا من خادم EC2 صغير في سنتك الأولى)، وكل مختبر في هذه الدورة مصمّم ليبقى داخلها. ثم، قبل أي شيء آخر، ثلاث خطوات أمان غير قابلة للتفاوض — إنجازها هو أول تمرين في مسيرتك الأمنية:

1. **MFA على المستخدم الجذر.** المستخدم الجذر (root) هو المفتاح الرئيسي للحساب. أضف مصادقة متعددة العوامل (تطبيق مصادقة على الهاتف) في IAM ثم Security credentials. ثم توقف عن استخدام root للعمل اليومي.
2. **أنشئ مستخدم IAM إداريًا** (المختبر 5 يشرح IAM بعمق؛ الآن: ال console ثم IAM ثم Users ثم أنشئ مستخدمًا بصلاحيّة إدارية) وسجّل الدخول به من الآن فصاعدًا.
3. **تنبيه فوترة.** ال console ثم Billing ثم Budgets ثم أنشئ **ميزانية إنفاق صفري (zero-spend budget)** (قالب AWS الذي يرأسك لحظة ظهور أي فوترة على الإطلاق). فعّل أيضًا تنبيهات استخدام الفئة المجانية. المهندس العاجز عن ضبط التكلفة عبء؛ وأنت بعد 20 دقيقة من إنشاء حسابك متقدّم بالفعل على كثير من المحترفين. (المرجع: <https://docs.aws.amazon.com/cost-management/latest/userguide/budgets-managing-costs.html>)

المفردات:

المصطلح	التعريف
عنوان IP	العنوان الشبكي الرقمي لآلة ما، مثل 172.31.8.14.
الحزمة (packet)	مظروف بيانات معنون واحد؛ كل الحركة تيارات منها.
المنفذ (port)	باب مرصم على الآلة يحدّد أي برنامج تقصده الحركة: 22 لـ SSH، و80 لـ HTTP، و443 لـ HTTPS.
IP خاص / عام	عنوان صالح داخل شبكة محلية فقط / عنوان يمكن بلوغه من الإنترنت كله.
DNS	النظام الذي يترجم الأسماء (example.com) إلى عناوين IP. “إنها دائمًا DNS.”

المصطلح	التعريف
HTTP / HTTPS	بروتوكول الطلب-والرد الخاص بالويب / البروتوكول نفسه مشفّرًا بـ TLS.
رمز الحالة (status code)	خلاصة رد الخادم: 200 حسنًا، 404 غير موجود، 500 خطأ خادم، 503 مثقل.
الجدار الناري (firewall)	قائمة القواعد التي تقرر أي الحزم تمر، بحسب المصدر والوجهة والمنفذ.
Security group	جدار AWS الناري لكل خادم. سوء إعداداته هو ثغرة المبتدئين الكلاسيكية.
زمن الاستجابة / عرض النطاق	التأخير (بالميلي ثانية) / السعة (في الثانية). المسافة تولّد زمن الاستجابة.
العميل / الخادم (كدورين)	السائل والمجيب في أي محادثة شبكية.
فئة AWS المجانية (Free Tier)	الحصة الشهرية المجانية في حساب AWS الجديد — ميزانية هذه الدورة بأكملها.
المستخدم الجذر (root user)	الهوية الرئيسية لحساب AWS. فعّل عليها MFA ثم توقف عن استخدامها.
تنبيه الفوترة / الميزانية	البريد التلقائي حين يتجاوز الإنفاق عتبة ما. عتبتك مضبوطة على 0 دولار.

فيديوهات هذه الوحدة:

الفيديو	القناة	المدة	الرابط
?What is AWS	Amazon Web Services (official)	min 2~	https://www.youtube.com/watch?v=a9__D53WsUs
Top 50+ AWS Services Explained in 10 Minutes	Fireship	min 10~	https://www.youtube.com/watch?v=JlbiYCM48to
AWS Networking Basics — VPC & Subnets	KodeKloud	min 30~	https://www.youtube.com/watch?v=QM63dyA_4Pc

قلها بصوت مسموع: “What’s the IP, and is it public or private?” (ما ال IP، وهل هو عام أم خاص؟) · “Is port 443 open in the security group?” (هل المنفذ 443 مفتوح في ال security group؟) · “Curl it —” (جربه ب curl — ما رمز الحالة الذي تحصل عليه؟) · “Did DNS resolve?” (هل استجاب DNS؟ تحقق ب nslookup قبل لوم الخادم).

التمارين: (1) نَقِّذ ping google.com (لاحظ زمن الاستجابة)، و nslookup google.com (لاحظ أن DNS يعيد عدة عناوين IP)، و curl -I https://aws.amazon.com (اقرأ سطر الحالة وثلاث ترويسات، ثم ابحث عن معنى كل منها). (2) اعثر على ال IP الخاص لأتتك (ip addr على لينكس) وعلى ال IP العام لك (ابحث عن “what is my IP”) — و اشرح في دفترك سبب اختلافهما. (3) أكمل إعداد حساب AWS: MFA، ومستخدم إداري، وميزانية إنفاق صفري — والتقط لقطة شاشة للميزانية لدفترك. (4) ارسم من الذاكرة: الحاسوب المحمول، فاستعلام DNS، فطلب HTTPS عبر المنفذ 443، فالجدار الناري، ف خادم الويب. ثلاث مرات، حتى يصبح الرسم تلقائيًا.

الإنجاز المرحلي — نهاية المرحلة 0: أن تستطيع سرد ما يحدث حين تكتب عنوان URL وتضغط Enter — ال DNS، وال IP، والمنفذ، والجدار الناري، وطلب HTTP، ورمز الحالة — في أقل من دقيقتين، مستخدمًا كل مصطلح استخدامًا صحيحًا؛ وأن يكون حسابك على AWS قائمًا مع MFA وتنبيه ميزانية 0 دولار. أنت الآن تعرف عن طريقة عمل الإنترنت أكثر من معظم من يكسبون رزقهم باستخدامه. والمرحلة الأولى هي حيث تبدأ البناء فوق ذلك.

المرحلة الأولى — جوهر السحابة، عمليًا (الأسابيع 5-12)

كيف تعمل هذه المرحلة. كل خدمة من الخدمات الخمس أدناه **مختبر (Lab):** هدف، ومخطط خطوات (مفصّل بما يكفي لاتباعه، وموجز بما يجبرك على التفكير — فالتفكير هو التعلم)، وما تعلمته، و— دائمًا — **التفكيك (teardown).** انضباط التفكير مهم من وجهين: فهو يبيّنك داخل الفئة المجانية، و”لا يترك شيئًا يعمل بلا حاجة” غريزة مهنية يسبرها المحاورون فعلاً. أعد بناء كل مختبر مرتين على الأقل: مرة باتباع المخطط، ومرة من الذاكرة. البناء الثاني هو حيث تنتقل المعرفة إلى يديك. خصص نحو أسبوع ونصف لكل مختبر؛ واستعمل الفائض للأعطال، لأن الأشياء ستتعطل، وتصحيحها هو أفضل تدريس لا تستطيع هذه الدورة كتابته سلفًا.

أولًا، ثلاث أفكار تؤطّر كل ما توشك على بنائه. **المناطق ومناطق الإتاحة:** المنطقة (Region) عنقود جغرافي من مراكز بيانات AWS (اختر واحدة قريبة منك والزمها — فالموارد في منطقة ما غير مرئية من منطقة أخرى، وهذا هو الالتباس رقم 1 في “أين ذهب خادمي؟”). **ومنطقة الإتاحة (Availability Zone — AZ)** مركز بيانات معزول داخل المنطقة؛ والأنظمة الجادة تعمل في اثنتين حتى لا يُسقطها تعطل مبنى واحد. ال **console** مقابل **ال CLI:** ال (https://console.aws.amazon.com) console هو لوحة تحكم AWS على الويب

— ممتاز للتعليم والاطلاع؛ أما **AWS CLI** (الأمر aws في طرفيتك) فيفعل كل ما يفعله الـ console، وبقابلية للبرمجة النصية. ستبدأ في الـ console وتخرج إلى الـ CLI، لأن المرحلة الثانية تُوِّمت كل ما تفعله هنا يدويًا.

المختبر 1 (الأسبوعان 5-6): EC2 — خادمك الأول

EC2 (Elastic Compute Cloud) يؤجّر آلات افتراضية تسمى **instances**. هذا هو الفعل البدئي في هندسة السحابة: خادم لينكس حقيقي، على الإنترنت، خلال 60 ثانية، مجانًا.

الهدف: أطلق خادم لينكس، وادخل إليه عبر SSH، واجعله يقدّم صفحة ويب للعالم، ثم دّمّه.

مخطط الخطوات: 1. الـ console ثم EC2 ثم Launch instance. سمّه. اختر **Amazon Linux 2023** بوصفه **AMI** (Amazon Machine Image — القرص القالب الذي يقلع منه خادمك) و**t2.micro** أو **t3.micro** بوصفه نوع الـ **instance** (الحجم؛ وهذان ضمن الفئة المجانية). 2. أنشئ زوج مفاتيح؛ سيُنزّل ملف مفتاح خاص بامتداد **.pem**. ذلك هو مفتاح SSH من الوحدة 1 — احرسه ونقّذ عليه **chmod 400**. في إعدادات الشبكة، اسمح بـ SSH (المنفذ 22) من "My IP" فقط — وأنت الآن تعرف تمامًا معنى قاعدة الـ **security group** هذه — واسمح بـ HTTP (المنفذ 80) من أي مكان. 4. أطلق، وانتظر حالة "running"، وانسخ الـ IP العام، ثم من طرفيتك: **ssh -i mykey.pem ec2-user@<public-ip>**. خذ نفسًا: أنت داخل حاسوب في مركز بيانات AWS. 5. على الخادم: **sudo dnf install -y nginx && sudo systemctl start nginx && sudo systemctl enable nginx**. ثم تصفح **<http://<public-ip>** — ذلك خادمك أنت يجيب العالم. 6. استبدل الصفحة الافتراضية: **echo "<h1>Built by me, on EC2</h1>" | sudo tee /usr/share/nginx/html/index.html**. حدّث الصفحة. لقطة شاشة للدفتر. 7. تجوّل كمهندس: **top** -h و **df** و **sudo tail -f /var/log/nginx/access.log** بينما تحدّث الصفحة — راقب زياراتك أنت وهي تصل إلى السجل.

ما تعلمته: الـ AMIs، وأنواع الـ instances، وأزواج المفاتيح، والـ security groups في معمعة العمل، وSSH، إلى خادم حقيقي، وتثبيت خدمة لينكس وتشغيلها، وقراءة سجلاتها — أي الحركات الجسدية اليومية للوظيفة.

التفكيك: EC2 ثم اختر الـ instance ثم Instance state ثم **Terminate**. تأكد أنها صارت "terminated". تمنح الفئة المجانية 750 ساعة شهريًا من instance صغيرة واحدة، فحتى تركها تعمل ما كان ليفوتر — لكن فكّكها على أي حال. فالعادة هي المقصد.

المختبر 2 (الأسبوع 7): S3 — ملفات لا تختفي أبدًا

S3 (Simple Storage Service) تخزين كائني (object storage): دلو بلا قاع للملفات بمئاته "أحد عشر تسعة". إنه الجواب الافتراضي على "أين نضع الملفات؟"، وهو — إذا أُسيء إعدادُه — مصدر أشهر تسريبات البيانات في التاريخ، ولهذا فإن هذا المختبر نصفه تخزين ونصفه أمن.

الهدف: أنشئ bucket، واشتغل عليه من ال CLI، واستصف موقعًا ساكنًا صغيرًا، وافهم بالضبط معنى "bucket عام".

مخطط الخطوات: 1. ال console ثم S3 ثم Create bucket (الأسماء فريدة عالميًا — yourname-lab-2026 يفي بالغرض). لاحظ أن **Block Public Access** مفعّل افتراضيًا. ارفع أي ملف عبر ال console؛ ثم نزلّه من جديد. 2. ثبّت AWS CLI محليًا وشغّل aws configure بمفتاح **access key** تنشئه لمستخدم IAM الإداري (مفتاح الوصول هو اسم مستخدم + كلمة مرور برمجيان للـ API — عامله معاملة كلمة المرور، ولا تضعه في الشيفرة أبدًا؛ ستستوعب هذه القاعدة في المختبر 5). 3. من طرفيتك: `aws s3 ls · aws s3 cp notes.txt s3://yourname-lab-2026/ · aws s3 sync ./myfolder s3://yourname-lab-2026/backup`. تحسّس الفرق: ال console زيارة؛ وال CLI هندسة. 4. الموقع الساكن: أنشئ bucket ثانيًا، وفعّل استضافة المواقع الساكنة، وارفّع ملف `index.html`، وأضف **سياسة ال bucket** الموثّقة للقراءة العامة (وثيقة صلاحيات بصيغة JSON — اقرأها سطرًا سطرًا: من يجوز له فعل ماذا على أي bucket). صفحتك الآن على الإنترنت دون أي خادم إطلاقًا. 5. استكشف **فئات التخزين** (Standard ثم Infrequent Access ثم Glacier: أرخص لكل جيجابايت، وأبطأ/أعلى في الاسترجاع) واضبط **قاعدة دورة حياة (lifecycle rule)** ("انقل الكائنات إلى IA بعد 30 يومًا") — أول مذاق لإدارة التكلفة المؤتمتة.

ما تعلمته: التخزين الكائني مقابل الأقراص، وال CLI ومفاتيح الوصول، وسياسات ال buckets والوصول العام (عمدًا لا سهوًا)، وفئات التخزين كرافعة تكلفة.

التفكيك: أفرغ الدلوين، واحذفهما، و— هذا مهم — عطّل أي مفتاح وصول لا تستخدمه. حصة S3 المجانية صغيرة (5 جيجابايت) لكن هذه الملفات كيلوبايتات؛ الانضباط، مرة أخرى، هو المقصد.

المختبر 3 (الأسبوعان 8-9): VPC — الشبكة التي تملكها

VPC (Virtual Private Cloud) شريحتك الخاصة المسيّجة من شبكة AWS. حتى الآن كنت تستخدم الافتراضية دون أن تنظر إليها؛ أما المهندسون فيبنون شبكاتهم بأنفسهم، لأن جملة "قاعدة البيانات تقع في subnet خاصة لا مسار لها إلى الإنترنت" هي نصف أمن السحابة، وأنت على وشك أن تجعلها حقيقية بيدك.

الهدف: ابن شبكة من طبقتين — subnet عامة لخادم ويب، و subnet خاصة لقاعدة بيانات مستقبلية — وأثبت أن النصف الخاص لا يمكن بلوغه من الإنترنت.

مخطط الخطوات: 1. ال console ثم VPC ثم Create VPC. أعطها كتلة العناوين `10.0.0.0/16` — ترميز CIDR، حيث تعني `/16` أن "أول 16 بتًا ثابتة، والباقي لي": `65,536` عناوينًا خاصًا. 2. أنشئ **شبكتين فرعيتين (subnets)**: `10.0.1.0/24` (عامة، في AZ-a) و `10.0.2.0/24` (خاصة، في AZ-b). ال subnet كتلة أصغر داخل ال VPC تعيش في منطقة إتاحة واحدة بالضبط. 3. أنشئ **Internet Gateway** (باب ال VPC إلى الإنترنت) وأوصله. أنشئ **جدول توجيه (route table)** بالقاعدة `0.0.0.0/0` → `internet gateway` ("كل ما ليس محليًا يذهب إلى باب الإنترنت") واربطه بالـ subnet العامة فقط. تحتفظ ال subnet الخاصة بالمسار المحلي وحده — وغياب المسار ذاك هو الأمن. 4. أطلق instance صغيرة واحدة من EC2 في كل

subnet (العامة بعنوان IP عام، والخاصة بدونه). 5. البرهان: SSH إلى الـ instance العامة — ينجح. جرّب عنوان الخاصة من حاسوبك — يعلق إلى الأبد، وقد صرت الآن قادرًا على قول السبب بدقة. ثم نفّذ SSH من العامة إلى الخاصة (فهي قابلة للبلوغ من داخل الـ VPC): الصندوق العام يعمل بوصفه **bastion host** (مضيف عبور) — نمط قياسي في الصناعة اكتشفته للتو ببنائه. 6. مفهوم إضافي تبحث عنه وتدوّنه: NAT gateway يتيح للآلات الخاصة الوصول إلى الخارج (للتحديثات) مع بقائها غير قابلة للبلوغ من الخارج — لكنه يفوتر بالساعة، فاقراً عنه ولا تبينه.

ما تعلمته: CIDR، والـ subnets، وجداول التوجيه، وبوابات الإنترنت، والفصل بين العام والخاص، والـ bastion hosts — بند الشبكات بعينه الموجود في كل وصف وظيفي سحابي، وقد صار ذاكرة عضلية.

التفكيك: أنه الـ instances أولاً، ثم احذف الـ VPC (وسيكنس حذفها معها الـ subnets وجداول التوجيه والبوابة). تحقق في EC2 أن لا شيء مكتوب عنده "running".

المختبر 4 (الأسبوع 10): RDS — قاعدة بيانات لا تسهر عليها

RDS (Relational Database Service) قاعدة بيانات مُدارة: تشغّل AWS محرك قاعدة البيانات (PostgreSQL و MySQL...) وتتولى النسخ الاحتياطي والترقيع والـ failover، بينما تملك أنت البيانات والاستعلامات. "الخدمة المُدارة" هي صفقة السحابة الجوهريّة — قايز بعض التحكم بكثير من الجهد غير المميّز — وهذا المختبر هو حيث تتحسس تلك الصفقة.

الهدف: أطلق قاعدة PostgreSQL في subnet خاصة، واتصل بها من instance في EC2، وافهم النسخ الاحتياطي والـ multi-AZ — ثم فكّكها فورًا، لأن RDS أسهل مختبر يُترك يعمل سهوًا.

مخطط الخطوات: 1. أعد بناء VPC المختبر 3 سريعًا (بناء ثانٍ من الذاكرة — هذا تكرار متباعد مقصود)، مضيّقًا subnet خاصة ثانية في منطقة إتاحة أخرى، لأن RDS تتطلب **subnet group** تمتد على منطقتي إتاحة. 2. الـ console ثم RDS ثم Create database ثم PostgreSQL ثم قالب **Free tier** (يختار مسبقًا db.t3.micro/db.t4g.micro، أحادي منطقة الإتاحة). عيّن كلمة المرور الرئيسية. ضعها في الـ VPC خاصتك، **Public access: No**، في security group يسمح بالمنفذ 5432 فقط من security group خادم الويب — قاعدة تشير إلى قاعدة أخرى بدل عنوان IP. أنيقة، وقياسية. 3. أطلق EC2 صغيرة في الـ subnet العامة، وثبّت عميل postgresql، واتصل: postgres -U postgres -h <rds-endpoint>. نقطة **النهاية (endpoint)** اسم DNS لا عنوان IP — فقد تنقل AWS الآلة التحتية، فيتبعها الاسم. (الوحدة 2 تدفع إيجارها بالفعل). 4. عند مَحّت postgres: أنشئ جدولًا، وأدخل ثلاثة صفوف، واسترجعها. لا تحتاج عمقًا في SQL اليوم؛ تحتاج أن تكون قد لمستها. 5. تفقّد دون تفعيل: إعداد النسخ الاحتياطي الآلي (استعادة إلى أي نقطة زمنية من لقطات ليلية + سجلات)، وخيار **Multi-AZ** (احتياطي حي في منطقة إتاحة أخرى مع failover تلقائي — التكلفة ضعف تقريبًا، ولهذا يقول الإنتاج نعم وتقول المختبرات لا). خذ لقطة (snapshot) يدوية، واعثر عليها في الـ console، وافهم أن بوسعك استعادة نسخة مطابقة منها. 6. سؤال دفتر يستحق عشر دقائق: ما الذي تفعله AWS لك هنا بالضبط مما كنت ستفعله بنفسك في الثانية فجّرًا؟ (الترقيع، والنسخ الاحتياطي، والـ failover، والعتاد). ذلك الجواب هو جواب مقابلات الخدمات المُدارة.

ما تعلمته: قواعد البيانات المُدارة، وال subnet groups، وقواعد security-group إلى security-group، ونقاط النهاية، واللقطات، وال multi-AZ/failover — إضافة إلى عرض حي لدفع التحكم ثمنًا لشراء الموثوقية.

التفكيك: احذف ال instance ال RDS (ارفض اللقطة الأخيرة لأنه مختبر؛ ولاحظ أن الإنتاج كان سيأخذها)، واحذف اللقطة اليدوية (اللقطات تفوتر عن التخزين!)، وأنت EC2، واحذف ال VPC. راجع لوحة الفوترة في اليوم التالي — فقراءتها أسبوعيًا عادة من عادات المرحلة الثالثة تبدأ الآن.

المختبر 5 (الأسبوعان 11-12): IAM — من يجوز له فعل ماذا

IAM (Identity and Access Management — إدارة الهوية والوصول) تقرر أي الأشخاص وأي البرامج يجوز لهم فعل ماذا على أي الموارد. لا تفوتر شيئًا ولا تجهّز شيئًا — وهي مع ذلك الخدمة الأكثر خضوعًا للتدقيق، والأكثر سبَرًا في المقابلات، والأكثر تورطًا في الاختراقات في AWS كلها. البنود الأمنية في الأوصاف الوظيفية ("ضوابط وصول بالحد الأدنى من الامتيازات"، "سياسات IAM") تعني هذا المختبر.

الهدف: أنشئ مستخدمين ومجموعات وسياسات و— الأهم — دورًا (role)، واستبطن مبدأ الحد الأدنى من الامتيازات بأن تشعّر بـ AWS وهي ترفض طلبك.

مخطط الخطوات: 1. المفاهيم أولًا، خمس دقائق: **المستخدم (user)** هوية لإنسان أو برنامج؛ **المجموعة (group)** تحزم المستخدمين؛ **السياسة (policy)** وثيقة JSON تمنح صلاحيات ("اسمح بـ s3:GetObject على arn:aws:s3:::my-bucket/*"); **والدور (role)** هوية لها سياسات لكن بلا كلمة مرور يتقمصها مؤقتًا طرف موثوق — وهكذا تحصل الخوادم والخدمات على صلاحيات دون أي سر مخزن. 2. أنشئ مستخدمة readonly-rita في مجموعة عليها سياسة ReadOnlyAccess المُدارة من AWS. سجّل الدخول بها في نافذة تصفح خاصة: ترى كل شيء، لكن كل زر إنشاء/حذف يفشل برفض صريح. اقرأ أحد تلك الأخطاء كاملاً — فتعلّم تفكيك رسالة "not authorized to perform X on Y" مهارة وظيفية يومية. 3. اكتب أول سياسة مخصصة لك في محرر JSON: اسمح بـ s3:ListBucket و s3:GetObject على bucket واحد محدد. أوصلها بمستخدمة جديدة؛ وتحقق أنها تستطيع قراءة ذلك ال bucket ولا شيء غيره. لقد نفّذت الآن الحد الأدنى من الامتيازات، لا عرّفته فحسب. 4. الدور، وهو الجائزة: أنشئ دورًا لـ EC2 عليه AmazonS3ReadOnlyAccess، وأطلق instance صغيرة موصولًا بها الدور، وادخل عبر SSH، وشغّل aws s3 ls — يعمل الأمر دون أي مفاتيح وصول على الآلة إطلاقًا. ال instance تتقمص الدور وتتلقى بيانات اعتماد قصيرة العمر تلقائيًا. هذا أهم نمط أمني منفرد في AWS: أدوار للآلات، لا مفاتيح على الآلات أبدًا. قلها مرتين. 5. دقّق حسابك كما يفعل المحترفون: هل على root مصادقة MFA (الأسبوع 3)؟ هل من مفاتيح وصول أقدم من 90 يومًا؟ هل لدى أي مستخدم صلاحيات أكثر مما يستخدم؟ شغّل تقرير بيانات الاعتماد (credential report) في IAM واقرأه. هذا الطقس ذو الدقائق العشر، شهريًا، هو قائمة نظافة IAM في المرحلة الثالثة وهي تولد.

ما تعلمته: المستخدمين/المجموعات/السياسات/الأدوار، وقراءة JSON السياسات وكتابته، والحد الأدنى من الامتيازات مفروضًا بالتجربة، وقاعدة الأدوار-لا-المفاتيح، وأول تدقيق أمني لك.

التفكيك: احذف مستخدمي الاختبار وبيانات اعتمادهم، وأنه ال instance، واحتفظ بمعرفة الأدوار إلى الأبد.

مفردات المرحلة الأولى (جدول واحد للمختبرات الخمسة):

المصطلح	التعريف
المنطقة / منطقة الإتاحة	عنقود جغرافي من مراكز بيانات AWS / مركز بيانات معزول واحد داخله. الأنظمة الجادة تمتد على منطقتي إتاحة.
EC2 / instance	خدمة تأجير الآلات الافتراضية / خادم مستأجر واحد.
AMI	Amazon Machine Image — القرص القالب الذي تقلع منه ال instance.
نوع ال instance	الحجم/المواصفات التي اخترتها (t3.micro = صغير جدًا، ضمن الفئة المجانية).
Security group	جدار ناري لكل مورد: أي المنافذ، ومن أي المصادر.
زوج المفاتيح	مفتاحا SSH الخاص/العام للوصول إلى ال instance.
S3 / bucket / object	التخزين الكائني / حاوية مسماة واحدة / ملف مخزن واحد.
فئة التخزين / قاعدة دورة الحياة	درجة سعر-سرعة للكائنات / قاعدة تلقائية تنقلها إلى فئات أرخص مع تقادمها.
سياسة ال bucket	وثيقة JSON تنص على من يجوز له فعل ماذا على ال bucket. العامة منها تصدر عناوين الأخبار.
AWS CLI / access key	واجهة الطرفية إلى AWS / بيانات الاعتماد البرمجية لها (عاملها معاملة كلمة المرور).
VPC / subnet	شريحتك الشبكية الخاصة / كتلة أصغر منها تعيش في منطقة إتاحة واحدة، عامة أو خاصة.
CIDR	ترميز كتل العناوين: 10.0.0.0/16 = "أول 16 بتًا ثابتة، و65,536 عنوانًا لي."
بوابة الإنترنت / جدول التوجيه	باب ال VPC إلى الإنترنت / القواعد التي تقرر إلى أين تُرسل الحركة. لا مسار = لا بلوغ = أمان.

المصطلح	التعريف
NAT gateway	يتيح للآلات الخاصة النداء إلى الخارج دون أن تكون قابلة للبلوغ من الخارج. يفوتر بالساعة — اعرفه ولا تتركه خاملاً.
Bastion host	الآلة العامة المحصنة التي تعبر منها بـ SSH لبلوغ الآلات الخاصة.
RDS / endpoint	خدمة قواعد البيانات العلائقية المُدارة / اسم DNS الذي تتصل به.
Multi-AZ / failover / اللقطة	نسخة لنقطة زمنية / احتياطي حي في منطقة إتاحة ثانية / التحويل التلقائي إليه.
مستخدم / مجموعة / سياسة / دور IAM	هوية / حزمة هويات / منحة صلاحيات بصيغة JSON / هوية قابلة للتمكص بلا كلمة مرور — كيف تحصل الآلات على صلاحياتها.
الحد الأدنى من الامتيازات (least privilege)	القاعدة الذهبية: أدنى صلاحيات تنجز العمل، لا أكثر.
الخدمة المُدارة	تدير AWS الجهد غير المميز (الترقيع، النسخ الاحتياطي، الـ failover)؛ وتبقى لك البيانات والقرارات.

فيديوهات هذه المرحلة:

الفيديو	القناة	المدة	الرابط
Getting Started with EC2	AWS Developers (official)	min 27	https://www.youtube.com/watch?v=nJ-djerESW0
Introduction to Amazon S3	Amazon Web Services (official)	min 5~	https://www.youtube.com/watch?v=ecv-19sYL3w
AWS Networking Basics — VPC & Subnets	KodeKloud (أعد مشاهدته بعد المختبر 3 — سيصلك مختللاً هذه المرة)	min 30~	https://www.youtube.com/watch?v=QM63dyA_4Pc
The AWS Shared Responsibility Model	Digital Cloud Training	min 4	https://www.youtube.com/watch?v=ESPBBEK-cvo

قلها بصوت مسموع: "It's in a private subnet; there's no route to the internet gateway" (إنها في subnet خاصة؛ لا مسار إلى بوابة الإنترنت). "The security group only allows 5432 from the web tier's security group" (ال security group لا يسمح بـ 5432 إلا من security group طبقة الويب). "The instance uses a role — there are no keys on the box" (ال instance تستخدم دورًا — لا مفاتيح على الجهاز). "Terminate it when you're done; nothing idles in this account" (أنهها حين تفرغ؛ لا شيء يبقى خاملاً في هذا الحساب).

الإنجاز المرحلي — نهاية المرحلة الأولى: بناء التحدي الشامل. في جلسة واحدة، من الذاكرة: VPC بـ subnets عامة وخاصة، ف خادم ويب EC2 (عام، موصول بدور، يقدم صفحة)، فقاعدة RDS PostgreSQL (خاصة، لا تُبلغ إلا من خادم الويب)، ف bucket في S3 تقرأه ال instance عبر دورها — ثم فكك كل ذلك تفكيكًا نظيفًا. أقل من ثلاث ساعات يعني أنك جاهز للمرحلة الثانية. وأيضًا: ابدأ التحضير لامتحان CLF-C02 الآن (دورة freeCodeCamp الكاملة — <https://www.youtube.com/watch?v=7HKot-brXFE> — بسرعة 1.25× بوصفها مراجعة؛ والطبعة الكلاسيكية ذات ال 14 ساعة هي <https://www.youtube.com/watch?v=NhDYbskXRgc>، واجلس لامتحان AWS Cloud Practitioner نحو الأسبوع 14-16.

المرحلة الثانية — الأتمتة (الأسابيع 13-20)

فكرة هذه المرحلة: كل ما نقرته في المرحلة الأولى ستفعله الآن بالشفيرة. الإهانة اللطيفة في الصناعة للنقر في ال console هي ClickOps؛ وعبارة الوصف الوظيفي لما يحل محله هي "تصميم وتطوير ونشر البنية التحتية السحابية باستخدام البنية التحتية كشفيرة". هذه المرحلة هي الفرق بين شخص استخدم AWS وشخص ستدفع له شركة لكي يديرها.

الوحدة 6 (الأسابيع 13-15): البرمجة النصية — Bash وPython للتشغيل، إضافة إلى Git

البرمجة النصية بـ Bash هي وضع أوامر الوحدة 1 في ملف ليكررها الحاسوب بلا خطأ. تعلّم، بهذا الترتيب: المتغيرات (NAME="web-1")، واستبدال الأوامر (TODAY=\$(date +%F))، والشرط (if [-f "\$FILE"]; then ... fi)، والحلقات (for f in *.log; do gzip "\$f"; done)، ورموز الخروج (exit codes) (كل أمر يعيد 0 عند النجاح وغير الصفرة عند الفشل؛ و&& تسلسل الأوامر عند النجاح — بهذا تتخذ السكريبتات قراراتها)، وقراءة المعطيات (arguments) (\$1, \$2). هذا 90% من Bash التشغيلي الحقيقي. اكتب هذا الأسبوع أول سكريبت نافع لك حقًا: backup.sh — يضغط دليلًا بـ tar، ويسمي الأرشيف بتاريخ اليوم، ويرفعه بـ cp s3 aws إلى bucket، ويحذف الأرشيفات المحلية الأقدم من 7 أيام، ويكتب بـ

echo سطر نجاح/فشل في ملف سجل. ذلك السكربت الواحد يجمع المتغيرات والاستبدال والشروط ورموز الخروج وال CLI في قطعة واحدة — وهو بند في سيرتك الذاتية ("أتمتُ النسخ الاحتياطي إلى S3").

Python الشقيق الأكبر لـ Bash: أفضل لكل ما فيه منطق أو بيانات أو تخاطب مع الـ APIs. تحتاج إلى Python تشغيلي وظيفي، لا إلى عمق هندسة البرمجيات: المتغيرات والأنواع، والقوائم والقواميس، و if/for، والدوال، وقراءة الملفات وكتابتها، ومعالجة الأخطاء بـ try/except، وثبيت المكتبات بـ pip. ثم تعرّف على **boto3**، مكتبة AWS للغة Python: `import boto3; ec2 = boto3.client("ec2"); ec2.describe_instances()` — وفجأة صارت معرفتك من المرحلة الأولى قابلة للبرمجة. اكتب `audit.py`: سكربت يعرض كل instance في الحساب مع اسمها ونوعها وحالتها ووقت إطلاقها، ويطبع سطر تحذير لأي شيء يعمل منذ أكثر من 24 ساعة. تهانينا — لقد كتبت أداة حقيقية لضبط التكلفة يدفع أصحاب العمل مقابلها.

Git هو نظام التحكم بالإصدارات الذي يعيش فيه كل عمل هندسي. المفاهيم: **المستودع (repository)** (مجلد يُسجّل تاريخه بأكمله)، و**الإيداع (commit)** (لقطة محفوظة واحدة مع رسالة)، و**الفرع (branch)** (خط عمل مواز)، و**البعيد (remote)** (النسخة على GitHub)، و**طلب السحب (pull request)** (طلب مراجعة فرع ودمجه). الحلقة اليومية ستة أوامر: `git init / git clone`، و**git status**، و**git add**، و**git**، و**git commit -m "message"**، و**git push**، و**git pull**. أنشئ حساب GitHub اليوم، وأنشئ مستودعًا باسم `cloud-journey`، ومن الآن فصاعدًا كل سكربت وكل ملف إعداد تكتبه في هذه الدورة يُودع فيه. بعد عشرة أشهر، يصبح سجل الإيداعات ذاك برهانًا علنيًا مؤرّجًا على كل ما تدّعيه هذه الدورة أنك قادر عليه.

المفردات:

المصطلح	التعريف
السكربت (script)	ملف أوامر يُنفَّذ من أعلى إلى أسفل — ذرة الأتمتة.
المتغير / المعطى	قيمة مسماة في السكربت / قيمة تُمرّر إليه عند تشغيله (\$1).
رمز الخروج	إشارة نجاح كل أمر (0) أو فشله (غير الصفر) — بها تتخذ السكربتات قراراتها.
Cron	مجدول لينكس: شغل سكربتًا كل ليلة في الثانية فجرًا، إلى الأبد.
Python / pip	لغة البرمجة المفضلة في عالم التشغيل / مثبت حزمها.
boto3	مكتبة Python لقيادة AWS — الـ console، على هيئة شيفرة.

المصطلح	التعريف
try/except	“حاول هذا؛ وإن فشل فافعل ذاك بدلاً منه” في Python — كيف تفشل السكريبتات بلباقة.
Git / المستودع / الإيداع	التحكم بالإصدارات / التاريخ المسجل لمشروع واحد / لقطة محفوظة واحدة مع رسالة.
الفرع / الدمج	خط عمل موازٍ / إعادته إلى الخط الرئيسي.
GitHub / remote / pull request	موقع الاستضافة / نسخة مستودعك السحابية / طلب مُراجع ومُناقش لدمج فرع.
README	وثيقة الواجهة للمستودع تشرح ما هو وكيف يُشغّل. مسؤولو التوظيف يقرؤونها.

فيديوهات هذه الوحدة:

الفيديو	القناة	المدة	الرابط
Learn Python — Full Course for Beginners	freeCodeCamp	~4.5 hr (شاهدها موزعة على الأسابيع الثلاثة)	https://www.youtube.com/watch?v=rfscVS0vtbw

قلها بصوت مسموع: “I’ll script it — it’ll be wrong once, then never again.” (سأكتبه سكريبتًا — سيخطئ مرة واحدة، ثم لن يخطئ أبدًا). · “Check the exit code before the next step runs.” (افحص رمز الخروج قبل أن تعمل الخطوة التالية). · “Commit it with a message that says why, not what.” (أودعه برسالة تقول لماذا، لا ماذا). · “It’s in the repo — pull the latest.” (إنه في المستودع — اسحب الأحدث).

التمارين: (1) backup.sh كما هو موصوف أعلاه، مجدولًا ليليًا بـ cron على instance مختبرية في EC2. (2) audit.py كما وُصف أعلاه. (3) كلاهما مودع في cloud-journey مع README يشرح كلاً منهما. (4) اكسر سكريبتك عمدًا (مسار خاطئ، bucket مفقود) واجعله يفشل بصوت عالٍ ووضوح — فمعالجة الأخطاء لغة حب في عالم التشغيل.

الإنجاز المرحلي: ينجو سكريبت النسخ الاحتياطي من تشغيله مرتين متتاليتين ومن مجلد مفقود (لا انهيار، ورسالة واضحة)؛ ويعمل سكريبت تدقيق Python على حسابك الحقيقي؛ ويعرض مستودعك أسبوعًا من الإيداعات.

الوحدة 7 (الأسبوعان 16-17): Terraform — البنية التحتية كشيفرة

الفكرة الكبرى: بدل نقر الموارد إلى الوجود، تكتب ملفات نصية تعلن ما ينبغي أن يوجد — “VPC واحدة، وsubnets اثنتان، وinstance واحدة، وهذه الوسوم” — ويجعل Terraform واقع AWS مطابقًا للملف. تصريح (declarative) لا أمري (imperative): تذكر الوجهة، لا الانعطافات. لماذا يطلبه كل وصف وظيفي: الملفات تعيش في Git، فتصبح البنية التحتية مُراجعة (طلبات سحب قبل التغييرات)، وقابلة للتكرار (الملف نفسه يبني dev و staging و prod متطابقة)، وقابلة للتراجع (تراجع بعكس الإيداع)، وقابلة للتدقيق (التاريخ يقول من غيّر ماذا، ومتى، ولماذا). ال console ورشة؛ و Terraform مصنع.

الحلقة الجوهرية التي ستديرها مئات المرات: terraform init (نزل الموقر provider — الإضافة التي تترجم ملفاتك إلى نداءات API لدى AWS) ثم terraform plan (تشغيلة جافة تطبع بالضبط ما سيُنشأ/ يُغيّر/ يُدمّر — اقرأ كل سطر؛ فالمهندسون الذين ينفذون apply دون قراءة الخطط يسببون الانقطاعات) ثم terraform apply (اجعله واقعًا) ثم terraform destroy (أزل ما صنعت — التفكيك بأمر واحد، وستجده الآن جميلًا). يتتبع Terraform ما بناه في ملف حالة (state file) — ذاكرته عن الواقع؛ إن فقدته أو عدّلت الواقع يدويًا من خلف ظهره (“الانحراف drift”) فسيتبعك الألم. تُعلن الموارد بلغة HCL، وهي لغة إعداد مقروءة:

```
resource "aws_instance" "web" {
  ami           = "ami-0abcdef1234567890"
  instance_type = "t3.micro"
  tags = { Name = "web-1", Project = "lab" }
```

المختبر (وهو الوحدة كلها): أعد بناء تحدي المرحلة الأولى الشامل في Terraform، تدريبًا. (1) هيئ مستودع terraform-labs؛ واكتب إعدادًا لمجرد bucket موسوم في S3؛ خطط، واقرأ، وطبّق، وافحص ال console — إنه هناك؛ ثم دمّر. (2) وسّع الإعداد: VPC وsubnets اثنتان وبوابة إنترنت وجدول توجيه — بناء المختبر 3، وقد صار نحو 60 سطر (3). HCL أضف ال security group وinstance من EC2 مع variables.tf (المنطقة، حجم ال instance) وoutputs.tf (اطبع ال IP العام بعد ال apply). (4) غيّر نوع ال instance في الملف وأعد ال apply — راقب Terraform يحسب الفرق ويغيّر ذلك فقط. هذا السلوك المدفوع بالفروق هو السحر كله. (5) دمّر كل شيء، وتأكد أن ال console فارغ، وأودع، ووسّم المستودع v1.0. الوثائق المرجعية على <https://developer.hashicorp.com/terraform> — ضعه في مفضلتك؛ فقراءة وثائق الموقر نصف عمل Terraform الحقيقي.

المفردات:

المصطلح	التعريف
IaC (البنية التحتية كشيفرة)	إعلان البنية التحتية في ملفات نصية مضبوطة الإصدارات تحوّلها أداة إلى واقع.
Terraform / HCL	أداة ال IaC المهيمنة متعددة السحب / لغة إعدادها.

المصطلح	التعريف
الموَقَّر (provider)	الإضافة التي تترجم ملفاتك إلى نداءات API لسحابة واحدة (هنا: AWS).
plan / apply / destroy	فرق التشغيل الجافة / جعله واقعًا / إزالة كل شيء. اقرأ كل خطة.
ملف الحالة (state file)	سجل Terraform لما بناه — ذاكرته عن الواقع. احيمه.
الانحراف (drift)	تغيّر الواقع من خلف ظهر Terraform (أحدهم نقر). العدو.
المتغير / المخرج / الوحدة (module)	مُدخل إعداد / نتيجة مطبوعة (عنوان IP، عنوان URL) / قطعة إعداد معلّبة قابلة لإعادة الاستخدام.
تصريحي مقابل أمري	ذكر الوجهة مقابل برمجة الانعطافات. Terraform تصرّحي.
CloudFormation	خدمة AWS الخاصة للـ IaC — الفكرة نفسها، لـ AWS فقط. الأوصاف الوظيفية تقبل أيًا منهما؛ تعلّم Terraform أولًا.

فيديوهات هذه الوحدة:

الفيديو	القناة	المدة	الرابط
Terraform explained in 15 mins	TechWorld with Nana	min 18	https://www.youtube.com/watch?v=l5k1ai_GBDE
Terraform Course — Automate your AWS cloud infrastructure	freeCodeCamp	hr 2.5~	https://www.youtube.com/watch?v=SLB_c_ayRMo

قلها بصوت مسموع: "Nothing changes in prod except through Terraform". (لا شيء يتغير في prod إلا عبر Terraform). "Show me the plan before you apply". (أرني ال plan قبل أن تنفذ ال apply). "That was clicked in manually — that's drift; let's import it or remove it". (هذا أُدخل نقرًا يدويًا — ذلك drift؛ فلنستورده أو نزله). "It's in the module; reuse it, don't rewrite it". (إنه في ال module؛ أعد استخدامه ولا تعد كتابته).

التمارين: المختبر أعلاه، إضافة إلى: (1) أنشئ انحرافًا عمداً — غير رسم الـ instance يدويًا في الـ console، وشغل terraform plan، وراقب Terraform وهو يلاحظ؛ ودوّن في دفترك ما اقترحه. (2) اكتب فقرة بلغة سهلة في README المستودع: “لماذا يتفوق هذا على النقر.” إن لم تستطع كتابتها، فأنت لم تمتلك الفكرة بعد.

الإنجاز المرحلي: من دليل فارغ، تستطيع إقامة حزمة الـ VPC + الـ instance بالأوامر init/plan/apply وإزالتها بـ destroy، شارحًا بصوت مسموع ما يفعله كل أمر — دون ملاحظات.

الوحدة 8 (الأسابيع 18-20): CI/CD و Docker وأول نظرة على Kubernetes

CI/CD (التكامل المستمر / التسليم المستمر) هو السير الناقل الآلي الموصول بمستودع Git خاصتك: مع كل push يفحص تغييرك ويختبره وبينه و— حين يُعدّ لذلك — ينشره تلقائيًا. الغاية إصدارات صغيرة متكررة مملّة بدل إصدارات نادرة مرعبة. **GitHub Actions** هو نظام CI/CD المدمج في GitHub: الـ workflow ملف YAML في .github/workflows/ يقول “عند الـ push، شغل هذه الخطوات على runner (آلة افتراضية مؤقتة) نظيف.” المختبر: أضف إلى terraform-labs ملف workflow يشغل terraform fmt - check و terraform validate مع كل push — ادفع ملفًا مشوّهاً عمداً وراقب علامة X الحمراء، ثم أصلحه وراقب علامة ✓ الخضراء. لديك الآن pipeline؛ ومهما صغر فهو من الفصيلة نفسها الموجودة في كل وصف وظيفي. (الخطوة الثانية، في المشروع 1: ملف workflow يشغل terraform plan على طلبات السحب، فتعرض تغييرات البنية التحتية المقترحة فرقها داخل المراجعة.)

Docker يغلف تطبيقًا مع كل ما يحتاجه في حاوية (container) — علبة غداء محكمة الإغلاق تعمل بشكل متطابق على حاسوبك وعلى EC2 وفي كل مكان آخر، منهيّة الطاعون القديم “يعمل على جهازي.” **الصورة (image)** هي الوصفة المجمّدة (تُبنى من Dockerfile، ملف نصي قصير من خطوات البناء: ابدأ FROM من صورة أساس، وانسخ تطبيقك بـ COPY، وحدّد أمر البدء)؛ **والحاوية** نسخة عاملة واحدة من صورة؛ **والسجل (registry)** (Docker Hub، أو ECR لدى AWS) هو حيث تُدفع الصور وتُسحب. الحاويات ليست آلات افتراضية: فهي تشارك نواة لينكس الخاصة بالمضيف، فتبدأ في نحو ثانية ويمكنك تشغيل عشرات منها على instance صغيرة. المختبر: على instance في EC2، ثبت Docker، ونقّذ docker run hello-world، ثم `docker run -d -p 80:8080 <a sample web image>` وتصفح إليه؛ ثم اكتب Dockerfile من خمسة أسطر يقدّم صفحتك الساكنة من المختبر 2 انطلاقًا من صورة أساس nginx، وابنه وشغله وادفعه إلى Docker Hub. تعلّم الأفعال اليومية: `docker ps` و `docker logs` و `docker exec -it <id>` و `bash` (صدفة داخل الحاوية) و `docker stop`.

Kubernetes (K8s) — في هذه الدورة، مهارة على مستوى القراءة، بتسمية صادقة. حين تشغل شركة مئات الحاويات عبر آلات كثيرة، لا بد من شيء يحدولها، ويعيد تشغيل المنهار منها، ويوسّع المزدحم، ويوجه الحركة بينها: ذلك المنسّق (orchestrator) هو Kubernetes. تعلّم خريطة المفاهيم الآن — **cluster** من **nodes** يشغل **Pods** (أصغر وحدة قابلة للنشر، وعادة حاوية واحدة)؛ و **deployment** يعلن “أبقي 3 نسخ من هذا الـ pod حية” ويجعلها الـ cluster صحيحة باستمرار (تصريح مجدّدًا — Kubernetes هو فلسفة Terraform مطبّقة على البرمجيات العاملة)؛ و **service** يمنح الـ pods عنوانًا ثابتًا. وعرض AWS المُدار هو

EKS. الأوصاف الوظيفية للمبتدئين تريد هذه المعرفة القرائية بالضبط مع طلاقة في Docker؛ أما مهارة تشغيل K8s الحقيقية (وشهادة CKA) فهدف قوي للسنة الثانية، لا للشهر الخامس. وهذه الدورة تخبرك بصدق أيهما أي.

المفردات:

المصطلح	التعريف
CI/CD	خط الأنابيب الآلي الذي يفحص ويبني ويختبر ويشحن كل تغيير.
GitHub Actions / workflow / runner	نظام CI/CD المدمج في GitHub / ملف YAML الذي يعرّف الـ pipeline / الآلة الافتراضية المؤقتة التي تنفّذه.
YAML	صيغة الإعداد القائمة على الإزاحة في الـ pipelines وKubernetes. الإزاحة معنى — فانتبه لها.
Docker / الصورة / الحاوية	عدة الحاويات / الوصفة المجمّدة التطبيقية / نسخة عاملة واحدة منها.
Dockerfile	الملف النصي القصير من الخطوات الذي يبني صورة.
ال سجل / ECR	حيث تُدفع الصور وتُسحب / سجل AWS.
Kubernetes (K8s) / cluster / node	منسّق الحاويات / مجموعة آلاته / آلة واحدة فيها.
Pod / deployment / service	أصغر وحدة قابلة للنشر / "أبقي N نسخة حية" مفروضة باستمرار / عنوان ثابت أمام الـ pods.
EKS	مستوى التحكم المُدار لـ Kubernetes لدى AWS.
التراجع (rollback)	العودة السريعة إلى الإصدار السابق حين يسيء إصدار جديد التصرف — شبكة الأمان التي يجعلها CI/CD روتينًا.

فيديوهات هذه الوحدة:

الرابط	المدة	القناة	الفيديو
https://www.youtube.com/watch?v=scEDHsr3APg	min 2	Fireship	DevOps CI/CD Explained in 100 Seconds
https://www.youtube.com/watch?v=0yWAtQ6wYNM	min 15~	TechWorld with Nana	What is DevOps? REALLY understand it
https://www.youtube.com/watch?v=Gjnup-PuquQ	min 2	Fireship	Docker in 100 Seconds
https://www.youtube.com/watch?v=3c-iBn73dDE	hr 3	TechWorld with Nana	Docker Tutorial for Beginners [FULL COURSE in 3 Hours]
https://www.youtube.com/watch?v=VnvRFRk_51k	min 16~	TechWorld with Nana	Kubernetes explained in 15 mins

قلها بصوت مسموع: "Don't merge until the pipeline is green". (لا تدمج حتى يخضّر الـ pipeline). · It's · containerized — same image in dev and prod. (إنه في حاوية — الصورة نفسها في dev و prod). · "Exec into the container and check its logs". (ادخل الحاوية بـ exec وافحص سجلاتها). · "K8s keeps three replicas up; kill one and watch it come back". (يبقي K8s ثلاث نسخ عاملة؛ اقتل واحدة وراقبها تعود).

التمارين: المختبران أعلاه، إضافة إلى: (1) دوّن في الدفتر جوابًا من فقرة واحدة عن سؤال "VM مقابل container مقابل serverless — متى يصلح كل منها؟" (فيديو Fireship بعنوان Serverless in 100 Seconds https://www.youtube.com/watch?v=W_VV2Fx32_Y — يكمل الخيار الثالث). (2) أنه instance مختبر Docker؛ وتأكد أن صورك باقية في السجل — ولاحظ أن القطعة المنتجة صارت الآن تعمّر أطول من الخادم، وتلك هي رؤية النشر الحديثة كلها في ملاحظة واحدة.

الإنجاز المرحلي — نهاية المرحلة الثانية: اعرض GitHub خاصتك: مستودع سكربتات تشغيلية عاملة بـ Bash وPython، ومستودع Terraform يبني حزمة حقيقية ويدمرها، وworkflow أخضر في Actions، وDockerfile مدفوعًا إلى سجل — وتستطيع شرح كل ملف منها في مقابلة. ملف GitHub ذاك لم يعد ملف طالب. إنه ملف مهندس مبتدئ.

المرحلة الثالثة — العمليات (الأسابيع 21-28)

فكرة هذه المرحلة: بناء الأنظمة يجعلك تُوظف؛ أما تشغيلها فهو الوظيفة الفعلية. بنود الوصف الوظيفي التي تجيب عنها هذه المرحلة هي النصف التشغيلي — “راقب صحة البنية التحتية”، و”شارك في الاستجابة للحوادث، بما فيها تحليل السجلات”، و”تحديد وتحليل وحل ثغرات البنية التحتية”، و”أدر تكاليف السحابة”. أسبوع مهندس السحابة معظمه من هذه المرحلة.

الوحدة 9 (الأسابيع 21-24): المراقبة والتسجيل والحوادث

المراقبة — أن تعرف قبل المستخدمين. CloudWatch هي خدمة الرصد (observability) المدمجة في AWS. ثلاثة عناصر أولية: **المقاييس (metrics)** (أرقام عبر الزمن — نسبة المعالج، نسبة القرص، عدد الطلبات، عدد الأخطاء)، و**الإنذارات (alarms)** (قاعدة تراقب مقياسًا: “إن تجاوز المعالج 80% لخمس دقائق فأبلغ”), و**لوحات المعلومات (dashboards)** (مقاييس مرتبة على شاشة واحدة). وتتدفق الإشعارات عبر **SNS** (Simple Notification Service — موضوع تنشر فيه فيصل المشتركين بريد أو نداء). الجرفة في ماذا/ تُنذر: لا تستدع إنسانًا إلا لما يحتاج إلى إنسان، وإلا تعلّم الناس تجاهل جهاز النداء (**إجهاد التنبيهات alert fatigue** — نمط الفشل الذي سبق كثيرًا من الانقطاعات الشهيرة). الإشارات الذهبية الأربع الجديرة بالحفظ: زمن الاستجابة، والحركة، والأخطاء، والتشيع — كم هو بطيء، وكم هو مشغول، وكم هو معطوب، وكم هو ممتلى.

التسجيل — أن تعرف لماذا. المقاييس تقول إن شيئًا ما خطأ؛ والسجلات تقول ما الذي حدث. CloudWatch Logs يركزها: وكيل (agent) على كل instance يشحن ملفات مثل `var/log/nginx/access.log` إلى **مجموعات سجلات (log groups)**، حيث يتيح **Logs Insights** الاستعلام عبر الآلات (“عُدّ استجابات 5xx بالدقيقة لآخر ساعة”). والمركزة مهمة لأن الخوادم صارت قابلة للاستغناء (أثبت ذلك في الوحدة 8) — فيجب أن تعمّر السجلات أطول من الآلات التي كتبتها.

المختبر (الأسبوعان 21-22): أقم خادم ويب صغيرًا مراقبًا، كله في Terraform (حزمتك من الوحدة 7، موسّعة): `EC2 + nginx` + وكيل CloudWatch؛ وموضوع SNS يرأسك بالبريد؛ وإنذار على ارتفاع المعالج وآخر على فشل فحص حالة الـ instance. ثم هاجم نفسك: ادخل عبر SSH وشغّل حارق معالج (`> yes /dev/null` & عدة مرات)؛ راقب المقياس يتسلق، والإنذار ينطلق، والبريد يصل؛ اقتل العمليات؛ وراقب التعافي. ثم استعلم عن سجلات وصولك في Logs Insights. لقد شهدت الآن حلقة الاكتشاف فالإبلاغ فالتشخيص فالحل كاملة على نظام بنيته أنت.

الاستجابة للحوادث — النصف البشري. الحادث (incident) حالة غير مخططة من “النظام ليس بخير”؛ ودرجات الخطورة (severity) (sev-1 = العملاء متوقفون، استنفار عام) تضبط حجم الاستجابة. الحلقة المهنية: اكتشاف (عبر الإنذار، لا عبر بريد من عميل، مثاليًا) ثم افرز (كم الأمر سيئ، ومن يُحتاج إليه) ثم خُفّف (أوقف النزيف أولًا — تراجع، أعد التشغيل، حوّل failover؛ والسبب الجذري لاحقًا) ثم حل ثم مراجعة ما بعد الحادث (post-mortem): مراجعة مكتوبة بلا لوم لما حدث، ولماذا، وما الذي سيمنع التكرار. انعدام اللوم

ليس تساهلاً؛ إنه هندسة: فالمعاقبون يخفون المعلومات، والمعلومات المخفية تسبب انقطاعات متكررة. **المناوبة (on-call)** هي تناوب من يحمل جهاز النداء؛ الأوصاف الوظيفية تذكرها والمحاورون يسألون عنها، وجوابك الصادق بعد هذه الوحدة: “لقد حاكيتها وأنا أعرف الحلقة.”

ال runbooks — بند التوثيق، مجسّدًا. ال runbook وصفة خطوة بخطوة لموقف تشغيلي واحد، مكتوبة بحيث يستطيع شخص مضغوط في الثالثة فجرًا اتباعها: الأعراض، فالفحوص (أوامر دقيقة)، فالإصلاحات (أوامر دقيقة)، فالتصعيد (من توقظه إن لم ينجح الأمر). **المختبر (الأسبوعان 23-24):** اكتب في مستودعك ملفي runbook — “خادم الويب متوقف” و”القرص يمتلئ” — ثم تدرب عليهما: اكسر الشيء، واتبع وثيقتك حرفيًا، وأصلح كل خطوة ثبت غموضها. ثم أدر يوم لعب (game day) كاملاً: اجعل صديقًا (أو ذكاء اصطناعيًا) يكسر حزمة مختبرك سرًا؛ فُتستدعى أنت، وتشخص من المقاييس والسجلات، وتخفف، وتكتب ال post-mortem في دفترك. تلك المراجعة قصة مقابلة، وجيدة.

المفردات:

المصطلح	التعريف
CloudWatch	خدمة المراقبة لدى AWS: مقاييس وإنذارات ولوحات وسجلات.
المقياس / الإنذار / اللوحة	رقم عبر الزمن / قاعدة تنطلق بناءً عليه / شاشة واحدة تجمعها.
SNS	خدمة الإشعارات التي تنشر فيها الإنذارات — بريد ورسائل نصية ونداء.
الإشارات الذهبية	زمن الاستجابة والحركة والأخطاء والتشيع — الأرقام الأربعة التي تصف صحة أي خدمة.
إجهاد التنبيهات	نداءات كثيرة لا تستدعي تصرفًا، فنداءات متجاهلة، فنداءات حقيقية مفعّوة. عدو تصميم الإنذارات.
مجموعة السجلات / Logs Insights	حيث تهبط السجلات الممركزة / لغة الاستعلام فوقها.
الحادث / الخطورة	تدهور غير مخطط / ترتيب درجة سوءه (sev-1 = الأسوأ).
الفرز / التخفيف / الحل	قيّم سريعًا / أوقف النزيف أولًا / أصلح فعليًا.
Post-mortem	المراجعة المكتوبة بلا لوم: ماذا، ولماذا، وما الذي يمنع التكرار.

المصطلح	التعريف
Runbook	الوصفة الصامدة أمام الثالثة فجراً لموقف واحد: أعراض وفحوص وإصلاحات وتصعيد.
المنابذة / يوم اللعب	تناوب جهاز النداء / حادث تدريبي مقصود.
MTTR	متوسط زمن التعافي (Mean time to recovery) — مقياس التشغيل الذي تحسّنه الفرق الناضجة.

قلها بصوت مسموع: “Did we find out from the alarm or from a customer?” (هل عرفنا من الإنذار أم من عميل؟) · “Mitigate first — root cause after we’re stable.” (خَفِّفْ أولاً — السبب الجذري بعد أن نستقر.) · “Is there a runbook for this? There will be by tomorrow.” (هل لهذا runbook؟ سيكون موجوداً بحلول الغد.) · “What did the post-mortem conclude, and what’s the prevention item?” (ما الذي خلص إليه الـ post-mortem، وما بند المنع؟)

الإنجاز المرحلي: الـ post-mortem الخاص بيوم لعبك موجود، وبلا لوم، ويسمي إجراء منع ملموساً واحداً نَقَّذته فعلاً بعده (إنذار أُصيف، أو runbook أُصلح).

الوحدة 10 (الأسابيع 25–28): عمليات الأمن وإدارة التكلفة

عمليات الأمن — نظافة لا بطولات. نموذج المسؤولية المشتركة (shared responsibility model) أولاً (أعد المشاهدة: <https://www.youtube.com/watch?v=ESPBBEK-cvo>): تؤمّن AWS السحابة ذاتها؛ وأنت تؤمّن ما تضعه فيها — ومعظم الاختراقات الحقيقية أخطاء إعداد من العملاء لا إخفاقات من AWS. قائمتك الأمنية التشغيلية، تتمرّن عليها حتى تصبح مملة:

1. **نظافة IAM** (شهرياً): MFA في كل مكان؛ لا مفاتيح وصول طويلة العمر حيث يفني دور بالغرض؛ شغّل تقرير بيانات الاعتماد؛ احذف المستخدمين والمفاتيح غير المستخدمة؛ وشكّك في كل * في كل سياسة. لقد بنيت هذه الغريزة في المختبر 5 — والآن صارت موعداً في التقويم.
2. **الترقيع (patching):** البرمجيات غير المرفّعة هي بداية معظم الاقتحامات. على الـ instances خاصتك: طبّق التحديثات وفق جدول (واعلم أن **SSM Patch Manager** يؤتمت ذلك على مستوى الأسطول)؛ وفي عالم الحاويات، الترقيع كثيراً ما يعني إعادة بناء الصورة من أساس محدّث — اربط ذلك بالوحدة 8.
3. **النسخ الاحتياطية — المختبرة منها:** النسخة الاحتياطية غير المختبرة أمنية لا خطة. لقطات RDS، وخاصية **versioning** في S3 (كل كتابة فوقية تحفظ النسخة القديمة — الضابط المضاد للهفوات والمضاد لبرمجيات الفدية). المختبر: خذ لقطة لقاعدة مختبرك، واستعدّها إلى instance جديدة، وتحقق من الصفوف، ثم فكّك. الآن يجوز لك قول “مختبرة” في المقابلات.

4. **الدرايزينات (guardrails):** فعّل CloudTrail (سجل التدقيق لكل نداء API في الحساب — من فعل ماذا، ومتى) وتصفّح أحداث الأمس مرة واحدة؛ وفعّل التجربة المجانية لـ GuardDuty (كشف تهديدات آلي) واقرأ ما الذي يراقبه. واعرف عبارة **التشفير أثناء التخزين وأثناء النقل (encryption at rest and in transit)** وأنها في AWS في معظمها خانة اختيار تدعمها خدمة KMS — من البدهيات، مفعّلة دائمًا.

إدارة التكلفة — المهارة التي تجعل المبتدئين يبدون كبارًا. بند الوصف الوظيفي حرفيًا: “أدر تكاليف السحابة عبر ضبط أحجام الموارد وتطبيق الـ auto-scaling ووسم الموارد.” كيف يدور العداد: الحوسبة تفوتر عن كل ثانية تكون فيها مشغّلة (الخمول ≠ مجاني — “تركناه يعمل” هو الهدر الكلاسيكي)، والتخزين بالجيجابايت-شهر، والـ egress (البيانات الخارجة من AWS) بالجيجابايت بينما الداخلة مجانية — مفاجأة الفاتورة الشهيرة. الروافع، بالترتيب الذي يستطيع مبتدئ سحبها: **أطفئه** (أنظمة التطوير ليلاً؛ عادة تفكيكك، وقد صارت صناعة) ثم **اضبط الحجم rightsize** (معظم الخوادم أكبر من حاجتها؛ افحص تاريخ المعالج في CloudWatch وقُلص) ثم **وسِّم كل شيء** (Project وOwner وEnvironment — الإنفاق غير الموسوم إنفاق بلا محاسبة؛ افرض الوسوم في Terraform خاصتك) ثم **دَرِّج التخزين** (قواعد دورة الحياة من المختبر 2) ثم اعلم أن **السعة المحجوزة/Savings Plans** (التزم من سنة إلى ثلاث سنوات ووفّر 30-70%) و**Spot** (خصم حتى 90%، قابل للمقاطعة) موجودان للأحمال المستقرة والدفعية على الترتيب. الأدوات: **Cost Explorer** (إنفاق الحساب، مرسومًا — اقرأه أسبوعيًا، إلى الأبد) و**AWS Budgets** (تنبيهك ذو الـ 0 دولار من الأسبوع 3، وقد فهمته الآن بوصفه أصغر فرد في عائلة جادة). والانضباط الأوسع يسمى **FinOps** — يستحق دقيقتين: https://www.youtube.com/watch?v=Y-c_xw9bHFw.

المفردات:

المصطلح	التعريف
نموذج المسؤولية المشتركة	تؤمّن AWS السحابة؛ وتؤمّن أنت ما تضعه فيها. معظم الاختراقات في النصف الثاني.
CloudTrail	سجل تدقيق الحساب: كل نداء API، ممن، ومتى.
GuardDuty	كشف التهديدات الآلي لدى AWS فوق السجلات والحركة.
الترقيع / SSM	تطبيق التحديثات الأمنية / خدمة Systems Manager لدى AWS التي تؤتمته على مستوى الأسطول.
S3 versioning	احفظ كل نسخة كُتِب فوقها — الضابط المضاد للهفوات والمضاد لبرمجيات الفدية.
KMS / التشفير أثناء التخزين والنقل	خدمة المفاتيح لدى AWS / بيانات مبعثرة على القرص وعلى السلك. مفعّل دائمًا.

المصطلح	التعريف
Egress	البيانات المغادرة لـ AWS — تفوتر بالجيجابايت؛ والواردة مجانية. مفاجأة الفاتورة الكلاسيكية.
ضبط الحجم (rightsizing)	تقليص الموارد المتضخمة إلى الحاجة المقيسة. مال مجاني.
الوسم (tagging)	وسم كل مورد بالمالك/المشروع/البيئة حتى يكون كل إنفاق قابلاً للمحاسبة.
Savings Plans / Spot	التزام من سنة إلى ثلاث سنوات بخصم 30-70% للحمل المستقر / سعة فائضة قابلة للاسترداد بخصم حتى 90% للدفعات.
Cost Explorer / Budgets	رسوم الإنفاق التي تقرأها أسبوعياً / التنبيهات التي تعني ألا تعرف الخبر من الفاتورة أبداً.
FinOps	انضباط جعل الإنفاق السحابي مرئياً ومورّعاً ومحسّناً باستمرار.

قلها بصوت مسموع: “Who has access to prod, and when did we last review the list?” (من يملك وصولاً إلى prod، ومتى راجعنا القائمة آخر مرة؟) · “When did we last restore a backup?” (متى استعدنا نسخة احتياطية آخر مرة؟) · “What’s untagged, and what died but is still billing?” (ما غير الموسوم، وما الذي مات لكنه ما زال يفوتر؟) · “It’s over-provisioned — the CPU history says we can halve it.” (إنه متضخم التجهيز — تاريخ المعالج يقول إن بوسعنا تنصيفه).

التمارين: (1) مختبر اختبار الاستعادة أعلاه. (2) ملف قائمة أمن شهرية في مستودعك، ثم شغلها فعلياً على حسابك. (3) في Cost Explorer، اعثر على أعلى شيء في تاريخ حسابك وشرحه في جملة دفتر واحدة. (4) أضف وسومًا افتراضية لكل مورد في مستودع Terraform خاصتك.

الإنجاز المرحلي — نهاية المرحلة الثالثة: أدر تدقيقاً ذاتياً كاملاً واحداً — قائمة الأمن، ومراجعة التكلفة، واختبار الإنذار، واستعادة نسخة احتياطية — واكتب تقرير الصفحة الواحدة. صرت الآن قادراً على أداء الوظيفة، لا البناء فحسب. وما تبقى هو إثبات ذلك للغرباء: المرحلة الرابعة.

المرحلة الرابعة — الجاهزية للتوظيف (الأشهر 8-12)

الوحدة 11 (الأشهر 8-10): ثلاثة مشاريع للملف المهني

الشهادات تقول إنك درست؛ والمشاريع تثبت أنك تستطيع البناء. ثلاثة مشاريع، كل واحد في مستودع GitHub خاص به، وكل واحد بمخطط معمارية، وREADME مكتوب لمدير توظيف (ماذا، ولماذا، وكيف يُشغَّل، وكَم يكلف)، وسكرت تفكيك. ابن، ثم صوّر/سجّل، ثم دَمّر — فالمستودع هو القطعة المنتجة، لا فاتورة تعمل.

المشروع 1 — تطبيق ويب ثلاثي الطبقات، مؤتمت بالكامل (القطعة المركزية). يبني Terraform كل شيء: VPC بـ subnets عامة وخاصة عبر منطقتي إتاحة؛ و**Application Load Balancer** (جهاز نشر الحركة — جديد عليك، وهو الخطوة التالية الطبيعية بعد المختبر 3) أمام **Auto Scaling Group** من instances الويب (تضيف AWS الـ instances وتزيلها مع الحمل — ابحث عنها ووضّلها)؛ وRDS في الـ subnets الخاصة؛ وS3 للأصول الساكنة؛ وإنذارات CloudWatch مع runbook. ويشغّل GitHub Actions الأمر terraform plan على كل طلب سحب وapply عند الدمج — خط أنابيب بنية تحتية حقيقي. هذا المشروع الواحد يبرهن على عشرة من بنود الوصف الوظيفي الأربعة عشر في جدول الموائمة؛ فتوقّع أن تمشي كل مقابلة عبره، وتمرّن على سرده في خمس دقائق.

المشروع 2 — خط أنابيب بيانات serverless (الاتساع). بلا خوادم إطلاقًا: ملف يهبط في bucket في S3 فيطلق دالة **Lambda** (شيفرتك بـ Python، تشغّلها AWS لكل حدث وتغوتر لكل استدعاء) تعالجه — حلّل ملف CSV، ولخّصه، واكتب النتائج إلى bucket ثانٍ أو جدول DynamoDB — مع النقاط الإخفاقات وتسجيلها في CloudWatch وإنذارها إلى بريدك. انشره بـ Terraform. إنه يعرض Python والتفكير المدفوع بالأحداث واتساعًا يتجاوز EC2. دقيقتا توجيهه أولًا: https://www.youtube.com/watch?v=W_VV2Fx32_Y.

المشروع 3 — عرض تشغيلي بأسلوب الإنتاج (المُؤمّن). خذ المشروع 1 وشغّله كأن الأمر جاد: لوحة CloudWatch للإشارات الذهبية؛ وإنذارات بسياسة استدعاء موثقة؛ وثلاثة runbooks؛ وإجراء نسخ احتياطي واستعادة مختبر مع أدلته؛ وجولة تحصين أمني (تدقيق IAM، وملاحظات الترقيع، وCloudTrail مفعّل) مكتوبة؛ وتحليل تكلفة ("هذه الحزمة تكلف X دولارًا شهريًا؛ وهذه التغييرات الثلاثة التي ستنصف التكلفة")؛ و-mortem واحد ليوم لعب. لا يكاد مرشح مبتدئ يملك هذا. إنه يجيب عن السؤال الوحيد الذي تطرحه المقابلات فعلاً — "هل يمكن ائتمان هذا الشخص على الإنتاج؟" — بوثائق بدل صفات.

الوحدة 12 (الأشهر 8-12، بالتوازي): جدول الشهادات

الشهادات تفتح مرشحات مسؤولي التوظيف وتنظّم دراستك. المسار القياسي لعام 2026 لهذا الدور، بتحضير واقعي على إيقاع هذه الدورة — لاحظ كيف يأتي كل امتحان مباشرة بعد مراحل الدورة التي تعلّم محتواها، ولهذا فإن هذه المدد أقصر من مدد الإنترنت:

الترتيب	الشهادة	ما تثبته	مدة التحضير من موقعك حينها
1	AWS Certified Cloud Practitioner (CLF-C02)	مفردات السحابة، والفوترة، والمسؤولية المشتركة	3-4 أسابيع (تقول الأدلة الشيء نفسه للمبتدئين؛ والمرحلتان 0-1 نحل
2	AWS Solutions Architect Associate (SAA-C03)	تصميم معماريات AWS حقيقية — الشهادة التي ترشّح بها الأوصاف الوظيفية فعلاً	6-8 أسابيع من التحضير المركّز (التقدير القياسي في الصناعة بعد الـ 8-10
3	HashiCorp Terraform Associate (003)	طلاقة IaC، موثّقة	2-3 أسابيع — بعد الوحدة 7 والمشروع 1 أنت تراجع في معظم الوقت (التحضير الرسمي: https://developer.hashicorp.com/terraform/tutorials/certification-003)

ورابعة اختيارية إن كنت تستهدف أدوارًا بمسمى تشغيلي: **AWS SysOps Administrator / CloudOps Associate** — محتواها هو المرحلة الثالثة حرفيًا. لامتحان CLF-C02، دورة freeCodeCamp الكاملة (<https://www.youtube.com/watch?v=7HKot-brXFE>)، والطبعة الكلاسيكية (<https://www.youtube.com/watch?v=NhDYbskXRgc>) مع امتحانات تدريبية هي الطريق المجاني المطروق؛ ولامتحان SAA-C03، اجمع بين دورة وكثير من الاختبارات التدريبية الموقوتة — فالامتحان قائم على السيناريوهات والجّد فيه مهم.

الوحدة 13 (الشهران 11-12): السيرة الذاتية والمقابلات والأسئلة العشرة

السيرة الذاتية: صفحة واحدة. ابدأ بالمهارات (AWS و Terraform و Python/Bash و Docker و CI/CD و CloudWatch — طابق كلمات الوصف الوظيفي ذاتها؛ فالمرشحات الآلية تطابق الكلمات المفتاحية) وبالمشاريع الثلاثة، كل مشروع في بندين على شكل فعلت X بواسطة Y محققًا Z: “Deployed a three-tier web app on AWS with Terraform and GitHub Actions; zero-downtime deploys via ALB + Auto

Scaling. "اذكر الشهادات بتواريخها. ضع رابط GitHub — ثم افترض أنهم سيفتحونه فعلاً، لأن الجيدين منهم يفعلون، وملفك الآن يكافئ الزيارة.

تحضير المقابلات: ثلاث نكهات تتمرّن عليها — المعلومات (جداول مفردات المراحل)، والسيناريوهات (المختبرات التي أجريتها فعلاً)، والسلوكية (قصص دفترك، مروية على شكل STAR: الموقف Situation، والمهمة Task، والفعل Action، والنتيجة Result). تمرّن بصوت مسموع، يوميًا، لأسبوعين — ومثاليًا مع محاور ذكاء اصطناعي، بلا رحمة. الأسئلة العشرة أدناه تغطي الكلاسيكيات؛ والإجابات النموذجية مكثفة عمدًا — وسّع كلاً منها بتفاصيل مختبراتك أنت، لأن "...وحين بنيت هذا، ما حدث فعلاً هو..." هي الجملة التي تفصلك عن المرشحين الذين اكتفوا بالقراءة.

الأسئلة العشرة، مع إجابات قوية:

1. "Explain the difference between a public and a private subnet." (اشرح الفرق بين الـ subnet العامة والخاصة.) جدول توجيه الـ subnet العامة فيه مسار إلى بوابة إنترنت، فيمكن بلوغ مواردها من الإنترنت؛ أما الـ subnet الخاصة فلا مسار كهذا لديها. طبقات الويب توضع عامة، وقواعد البيانات خاصة، والوصول الإداري يبلغ الآلات الخاصة عبر bastion أو SSM. في مختبراتي أثبتُ أن الـ instance الخاصة لا تُبلغ من حاسوبي لكنها تُبلغ من الطبقة العامة — فالمسار الغائب هو الأمن.
2. "What is IAM, and what is least privilege?" (ما IAM، وما الحد الأدنى من الامتيازات؟) يتحكم IAM في أي الهويات تستطيع أداء أي الأفعال على أي الموارد، عبر المستخدمين والمجموعات والأدوار وسياسات JSON. والحد الأدنى من الامتيازات يعني منح الأدنى الذي ينجز العمل — فمستخدم سكرتير التدقيق عندي له قراءة فقط على خدمة واحدة بالضبط، و EC2 instances عندي تستخدم أدواتًا بدل مفاتيح مخزنة، فلا بيانات اعتماد طويلة العمر يمكن أن تتسرب.
3. "An EC2 web server stopped responding. Walk me through your troubleshooting." (توقف خادم ويب EC2 عن الاستجابة. خذني عبر خطوات استكشافك للمشكلة.) أولاً النطاق: خادم واحد أم كل شيء — أفحص لوحة CloudWatch وأي إنذارات. ثم الطبقات بالترتيب: فحوص حالة الـ instance؛ والـ security group — هل 80/443 مفتوحان فعلاً؛ ثم أدخل عبر SSH — هل nginx يعمل (systemctl status)، هل القرص ممتلئ (df -h)، هل المعالج محشور (top)؛ ثم السجلات (tail لسجل الأخطاء). خفف أولاً — أعد تشغيل الخدمة أو استبدل الـ instance — والسبب الجذري بعد أن نستقر. ذلك هو الترتيب الذي أتدرب عليه في الـ runbooks خاصتي.
4. "Why Terraform instead of clicking in the console?" (لماذا Terraform بدل النقر في الـ console؟) الـ console لا يترك أثراً قابلاً للمراجعة ولا يستطيع إعادة بناء أي شيء. أما Terraform فتصريحه ومضبوط الإصدارات: التغييرات تمر بطلبات سحب، والإعداد نفسه يبني بيئات متطابقة، والتغيير السيئ يُراجع عنه بعكس إيداع، و plan يعرض الفرق الدقيق قبل حدوث أي شيء. والـ pipeline عندي يشغل plan على كل PR فيكون الفرق جزءًا من المراجعة.
5. "What's the difference between a container and a virtual machine?" (ما الفرق بين الحاوية والآلة الافتراضية؟) الآلة الافتراضية تحاكي العتاد وتحمل نظام تشغيل كاملاً وتقلع في دقائق؛ والحاوية تشارك نواة المضيف وتغلّف التطبيق وتبعياته فقط وتبدأ في نحو ثانية. الحاويات تمنح سلوكًا

متطابقًا عبر البيئات — فالصورة نفسها تعمل على حاسوبي وعلى EC2. وتبقى الآلات الافتراضية مهمة لل عزل وتشغيل الأحمال غير اللينكسية.

6. “How would you reduce our AWS bill?” (كيف ستخفض فاتورة AWS لدينا؟) بترتيب الجهد: أجد ما يعمل ولا ينبغي له — instances خاملة، وأقرص غير موصولة، ولقطات قديمة؛ Cost Explorer والوسوم يجعلان ذلك مرئيًا. ثم أضبط الأحجام بناءً على تاريخ CloudWatch. ثم أجدول إطفاء بيئات التطوير خارج ساعات العمل. ثم أدّج بيانات S3 القديمة إلى فئات أرخص. ثم ألنزم بالأحمال المستقرة في Savings Plans وأضع الدفعات القابلة للمقاطعة على Spot. وأراقب ال egress — فالبيانات الخارجة هي سطر المفاجأة الكلاسيكي.

7. “What happens when you type a URL and press Enter?” (ماذا يحدث حين تكتب عنوان URL وتضغط Enter؟) يحل DNS الاسم إلى عنوان IP؛ ويفتح المتصفح اتصال TCP على المنفذ 443 ويتفاوض على TLS؛ ثم يرسل HTTP GET؛ وفي الطرف الآخر يصطدم الطلب بموَزّع حمل يمرره إلى instance سليمة في subnet عامة؛ وقد يستعلم التطبيق قاعدة بيانات في subnet خاصة؛ ويعود الرد برمز حالة — 200 إن كان كل شيء سليمًا — فيعرضه المتصفح. أستطيع التعمق في أي قفزة، وقد بنيت كلاً منها.

8. “A teammate’s change broke production. What happens next?” (تغيير أجراه زميل كسر الإنتاج. ماذا يحدث تاليًا؟) خُفّف أولاً — تراجع عبر ال pipeline؛ فالتعافي يتفوق على التشخيص في اللحظة. ثم post-mortem بلا لوم: ماذا حدث، ولماذا سمح النظام به، وأي درابزين كان غائبًا — اختبار، أو مراجعة خطة، أو إنذار. انعدام اللوم مهم عمليًا: فمن يخافون العقاب يخفون المعلومات، والمعلومات المخفية تسبب التكرار.

9. “What is the shared responsibility model?” (ما نموذج المسؤولية المشتركة؟) تؤمّن AWS السحابة — مراكز البيانات، والعتاد، وال hypervisor؛ ويؤمّن العميل ما فيها — البيانات، و IAM، وقواعد الشبكة، والترقيع، وإعدادات التشفير. معظم اختراقات السحابة الحقيقية على جانب العميل، أخطاء إعداد كالبuckets العامة والمفاتيح المسرّبة. لذا فإن “AWS آمنة” و”نحن آمنون على AWS” جملتان مختلفتان، والثانية هي وظيفتي.

10. “Tell me about a problem you debugged.” (حدثني عن مشكلة صحتها.) (قصتك أنت، من الدفتر، على شكل STAR. هيكل مثالي:) خلال يوم لعب المراقبة عندي، انطلق إنذاري بسبب تشبع المعالج (الموقف). كان علي إيجاد السبب وإيقافه قبل أن يلاحظ “المستخدمون” (المهمة). أظهرت المقاييس بداية القفزة؛ وربطها Logs Insights بعمليات جامحة؛ فدخلت عبر SSH، وتأكدت بـ top، وقتلتها، وراقبت التعافي على اللوحة (الفعل). حُلّت في إحدى عشرة دقيقة، وأنتج ال post-mortem إنذار عددٍ عمليات نُقذته لاحقًا في Terraform (النتيجة).

ميكانيكا البحث عن عمل: قدّم طلباتك من الشهر 11 — بعد SAA، ولا تنتظر “الجاهزية”، فالمقابلات تدريب. المسميات المستهدفة: cloud engineer (junior/associate)، و cloud support engineer، و cloud، و operations engineer، و junior DevOps engineer، و AWS support associate. كل رفض جاء بعد مقابلة درس مجاني؛ فدوّن ما سألوه.

المفردات:

المصطلح	التعريف
الملف المهني (portfolio)	برهان علني موثّق على قدرتك على البناء — وفي هذه الحرفة: مستودعات GitHub بمخططات وملفات README.
Application Load Balancer (ALB)	ناشر الحركة لدى AWS: يوزع الطلبات على الـ instances السليمة ويسقط المريضة.
Auto Scaling Group	تبقى N من الـ instances حية وتعُدّل N مع الحمل — معالجة ذاتية ومرونة في أداة واحدة.
Lambda / serverless	شيفرة تشغيلها AWS لكل حدث وتفوتر لكل استدعاء — لا خوادم تُدار إطلاقًا.
DynamoDB	جدول NoSQL الـ serverless لدى AWS — يقترن طبيعياً بـ Lambda.
STAR	الموقف، والمهمة، والفعل، والنتيجة (Situation, Task, Action, Result) — شكل قصة المقابلة الجيدة.
ATS	نظام تتبع المتقدمين — مرشّح الكلمات المفتاحية الذي يجب أن تجتازه سيرتك ذات الصفحة الواحدة.
CLF-C02 / SAA-C03 / Terraform Associate 003	امتحاناتك الثلاثة: المفردات، والمعمارية، والـ IaC.
SysOps / CloudOps Associate	شهادة AWS المساعدة الاختيارية ذات التركيز التشغيلي — المرحلة الثالثة، ممتحنة.
الإنجاز المرحلي — نهاية الدورة: ثلاثة مستودعات يستطيع غريب فهمها، وثلاث شهادات مجدولة أو مجتازة، وعشر إجابات متمرّن عليها بصوت مسموع، وطلبات توظيف مقدّمة. أنت لست "تأمل في دخول عالم السحابة". أنت مهندس سحابة مبتدئ يملك الأدلة، ويجري المقابلات.	

منهج الصفحة الواحدة

متى	التركيز	البرهان العملي	البرهان الخارجي
الأسبوعان 1-2	كيف تعمل الحواسيب؛ لينكس والطرفية والصلاحيات وSSH	عمليات ملفات مبرمجة؛ أول سكربت صدفة	—
الأسبوعان 3-4	الشبكات: IP وDNS والمنافذ وHTTP والجدران النارية؛ حساب AWS	حساب + MFA + ميزانية 0 دولار	—
الأسبوعان 5-6	المختبر 1: EC2	خادم ويب على الإنترنت، ثم مفكّك	—
الأسبوع 7	المختبر 2: S3 + CLI	موقع ساكن؛ طلاقة CLI	—
الأسبوعان 8-9	المختبر 3: VPC	شبكة عامة/خاصة، برهان bastion	—
الأسبوع 10	المختبر 4: RDS	قاعدة بيانات خاصة، لقطة، تفكيك	—
الأسبوعان 11-12	المختبر 5: IAM؛ إعادة بناء تحدي المرحلة الأولى	الحزمة الكاملة من الذاكرة، >3 ساعات	حجز CCP
الأسابيع 13-15	Bash وPython/boto3 وGit	audit.py و backup.sh وتاريخ المستودع	امتحان CCP (نحو الشهر 4)
الأسبوعان 16-17	Terraform	الحزمة كشيفرة، plan/apply/destroy	—
الأسابيع 18-20	CI/CD وDocker ومعرفة K8s القرائية	pipeline أخضر؛ صورة في سجل	—
الأسابيع 21-24	CloudWatch والسجلات والحوادث والـ runbooks	إنذار انطلق + post- mortem يوم لعب	—
الأسابيع 25-28	عمليات الأمن؛ إدارة التكلفة	تقرير تدقيق ذاتي؛ اختبار استعادة	—
الأشهر 8-10	مشاريع الملف المهني 1-3	ثلاثة مستودعات موثقة	SAA-C03 (الشهران 8- 9)

متى	التركيز	البرهان العملي	البرهان الخارجي
الشهران 10-11	تحضير Terraform Associate	—	Terraform Associate
الشهران 11-12	السيرة والمقابلات والطلبات	عشر إجابات، بصوت مسموع	أول العروض

كلمة ختامية من معلمك. اثنا عشر شهرًا مدة قصيرة لتغيير مهنة وطويلة لعادة يومية، فإليك الصفقة الصادقة: من ينهون هذه الدورة ليسوا الأذكي — بل هم من كتبوا الأوامر في أيام التعب أيضًا. كل رسالة خطأ تصطدم بها هي المنهج وهو يعمل، لا وهو يفشل؛ والمهندس الذي كسر وأصلح مئة شيء صغير هو بالضبط ما يقصده مدير التوظيف بكلمة “الخبرة”. احتفظ بالدفتر، وفكّك ما تبنيه، ولا تدع في مقابلة ما لا تسنده مستودعاتك — وبعد عام من الآن، حين ينطلق جهاز النداء في الثالثة فجرًا، ستشعر بشيء غير متوقع تحت الأدريالين: الكفاءة. اذهب وابن.

المصادر

بحث الأوصاف الوظيفية (أغسطس 2026): [Wiz](#) · [Arc.dev — AWS Cloud Engineer Job Description](#) · [XOPA Cloud Engineer Job Description Guide](#) · [DevsData — AWS Cloud Engineer JD Template](#) · [Betterteam — Cloud Engineer Job Description](#) · [Cloud Engineer JD Template 2026](#) — مسار الشهادات ومدد التحضير: [Cloud Evolvers — Cloud](#) · [StudyTech — AWS Certification Roadmap 2026](#) · [HashiCorp — Terraform Associate 003 prep](#) · [Engineer Roadmap 2026](#). سياق الرواتب (الوسيط الأمريكي ≈ 104 آلاف دولار، والنطاق 85–140 ألف دولار): [Wiz](#). أعلاه. جميع روابط YouTube تُحقّق منها عبر بيانات YouTube الوصفية وقت الكتابة.

دورة من سلسلة B4LCILC — مجلد مرافق لدورة “The Cloud Leader Course”.

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 استنجد، 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 موجّهة 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، وتمرين 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، وتمرين تقدير تكلفة كامل — تسعير تصميم قبل أن يبينه أحد.

لدورة مهندس السحابة

ما هو الـ serverless فعلاً

كل ما أتممت أتمتته حتى الآن ما زال يفترض وجود آلة: فمهمة cron في مختبر هذه الوحدة تعمل دقيقتين في اليوم على آلة افتراضية تدفع ثمنها الشهر كله، وترقّعها، وتراقبها، وستصلحها يومًا ما في الثالثة فجرًا. والـ serverless هو جواب السحابة على ذلك التنافر. ما زالت هناك خوادم — فالاسم تسويق — لكنها مشكلة المزوّد. تسلّمه دالة، فيشغّلها هو حين يستثيرها شيء ما، وتدفع أنت عن كل استدعاء، بالميلي ثانية. يسمى هذا النموذج **FaaS — الدالة كخدمة (Function as a Service)**: وهو Lambda على AWS، و Azure Functions على Azure. والمفاهيم تنتقل بينهما اسمًا باسم.

التشريح يسعه كفّ واحد. الـ handler هو الدالة التي تستدعيها المنصة، وهي تتلقى حدثًا (event) — وصفاً بصيغة JSON (ترميز كائنات JavaScript — JavaScript Object Notation) لما حدث أيّا كان: ملف حطّ في bucket، أو طلب HTTP وصل، أو جدول زمني انطلق. انتبه جيدًا للأخير: “جدول زمني انطلق” هو بالضبط ما يفعله cron — إلا أن Lambda على جدول زمني لا تحتاج آلة مشغّلة بين التشغيلات. وتحويل أحد السكريبتات التشغيلية في هذه الوحدة هو مشروع serverless أول كلاسيكي.

الاقتصاد هو سبب وجود النموذج، وهو يقطع في الاتجاهين. فالطبقة المجانية لدى AWS تشمل مليون استدعاء Lambda في الشهر، إلى الأبد — يمكن لدالة مختبرك أن تعمل كل دقيقة من السنة دون أن تكلّف شيئًا، مما يجعل الـ serverless أرخص مأوى لأعمال اللصق التشغيلية. لكن التسعير بالميلي ثانية الذي يكون خطأً تقريب عند الأحجام المنخفضة يصبح مألًا حقيقياً عند الأحجام العالية؛ فالدالة التي تعمل باستمرار تكون

عادةً أرخص كحاوية أو كآلة افتراضية. والمهارة الاحترافية هي معرفة أي جانب من ذلك الخط يقع عليه عبء عمل ما.

تحفظان صادقان. الأول، **الإقلاع البارد (cold start)**: الدالة التي لم تعمل مؤخرًا تستغرق بضع مئات من الملي ثانية حتى تستيقظ — أمر لا يُذكر لتقرير ليلى، واعتبار تصميمي لـ API حساسة للتأخير. الثاني، أن الدالة ما زالت تعمل بهوية: فـ Lambda تتقمص دور IAM (إدارة الهوية والوصول — Identity and Access Management) تمامًا كما تفعل نسخة EC2، والدالة ذات الصلاحيات المفرطة هي الخطأ الأمني نفسه في غلاف أصغر.
