



Université Constantine 2
جامعة قسنطينة 2

حوسبة سحابية

-الدورة - 4

الفصل الرابع: الحوسبة بدون خادم

دكتور منور ر.

كلية التقنيات الجديدة

rostom.mennour@univ-constantine2.dz

الطلاب المعنيين

الكلية/المعهد	قسم	مستوى	تخصص
التكنولوجيات الجديدة	إذا كان	سيد 1	سديا

خريطة الدورة

• كيف وصلنا إلى هنا ؟

• ما هي الحوسبة بدون خادم؟ • كيف تعمل الحوسبة بدون خادم

• لماذا يختلف عنه؟

• المزايا والقيود



القسم الأول: كيف وصلنا إلى هنا؟

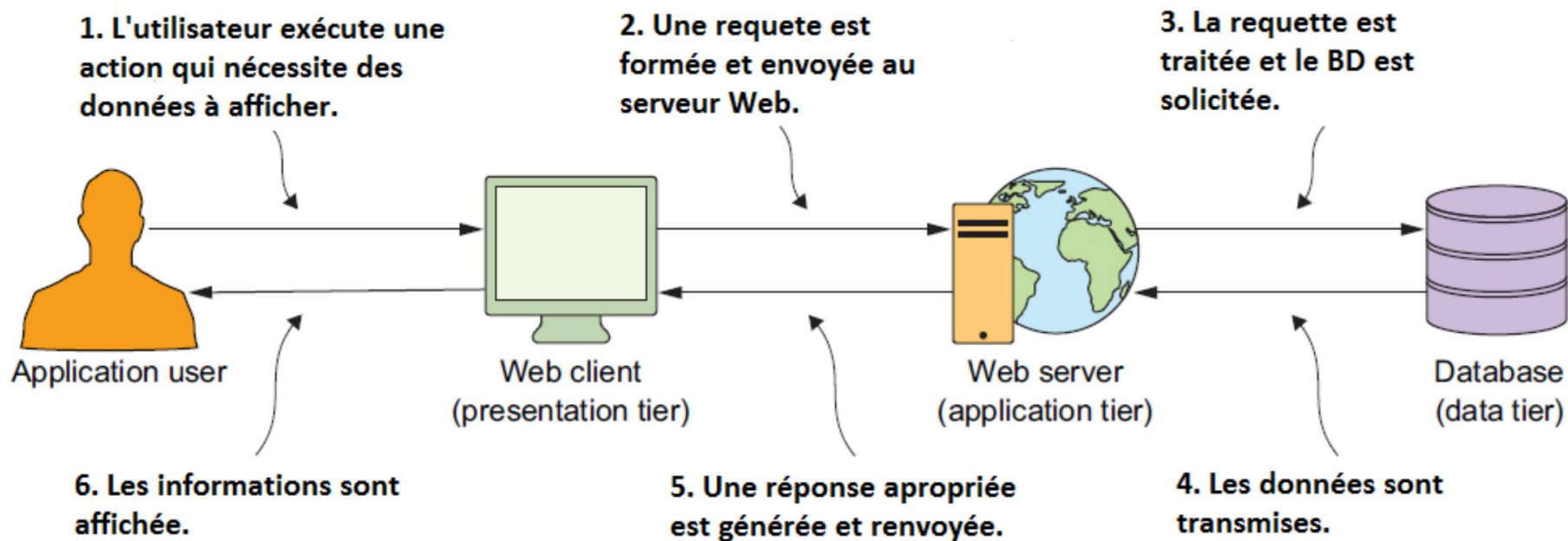
كيف وصلنا إلى هنا ؟

العمارة الكلاسيكية

- إذا نظرت إلى الأنظمة التي تشغل معظم برامج الويب اليوم، فسترى خوادم خلفية تقوم بأشكال مختلفة من العمليات الحسابية والواجهات الأمامية من جانب العميل التي توفر واجهة للمستخدمين للعمل من خلال المتصفح أو الهاتف المحمول أو سطح المكتب الخاص بهم.
- في تطبيق ويب نموذجي، يقبل الخادم طلبات HTTP من الواجهة الأمامية ويقوم بمعالجة الطلبات. يمكن أن تمر البيانات عبر العديد من طبقات التطبيق قبل حفظها في قاعدة البيانات.
- وبطبيعة الحال، تصبح معظم الأنظمة أكثر تعقيدًا عندما يتم أخذ أشياء مثل موازنة التحميل، والمعاملات، والتجميع، والتخزين المؤقت، والمراسلة، وتكرار البيانات في الاعتبار.
- تتطلب معظم هذه البرامج خوادم تعمل في مراكز البيانات أو في السحابة والتي تحتاج إلى الإدارة والصيانة والتصحيح والنسخ الاحتياطي.
- يعد توفير الخوادم وإدارتها وتصحيحها مهمة تستغرق وقتًا طويلًا وتتطلب غالبًا أشخاصًا متخصصين في العمليات. من الصعب إعداد بيئة غير تافهة وتشغيلها بكفاءة.
- تعد البنية التحتية والأجهزة مكونات ضرورية لأي نظام لتكنولوجيا المعلومات، ولكنها غالبًا ما تكون أيضًا مصدر إلهاء عما يجب أن يكون الهدف الرئيسي: حل مشكلة العمل.

كيف وصلنا إلى هنا ؟

العمارة الكلاسيكية



كيف وصلنا إلى هنا ؟

البنى الموجهة نحو الخدمة والخدمات الدقيقة

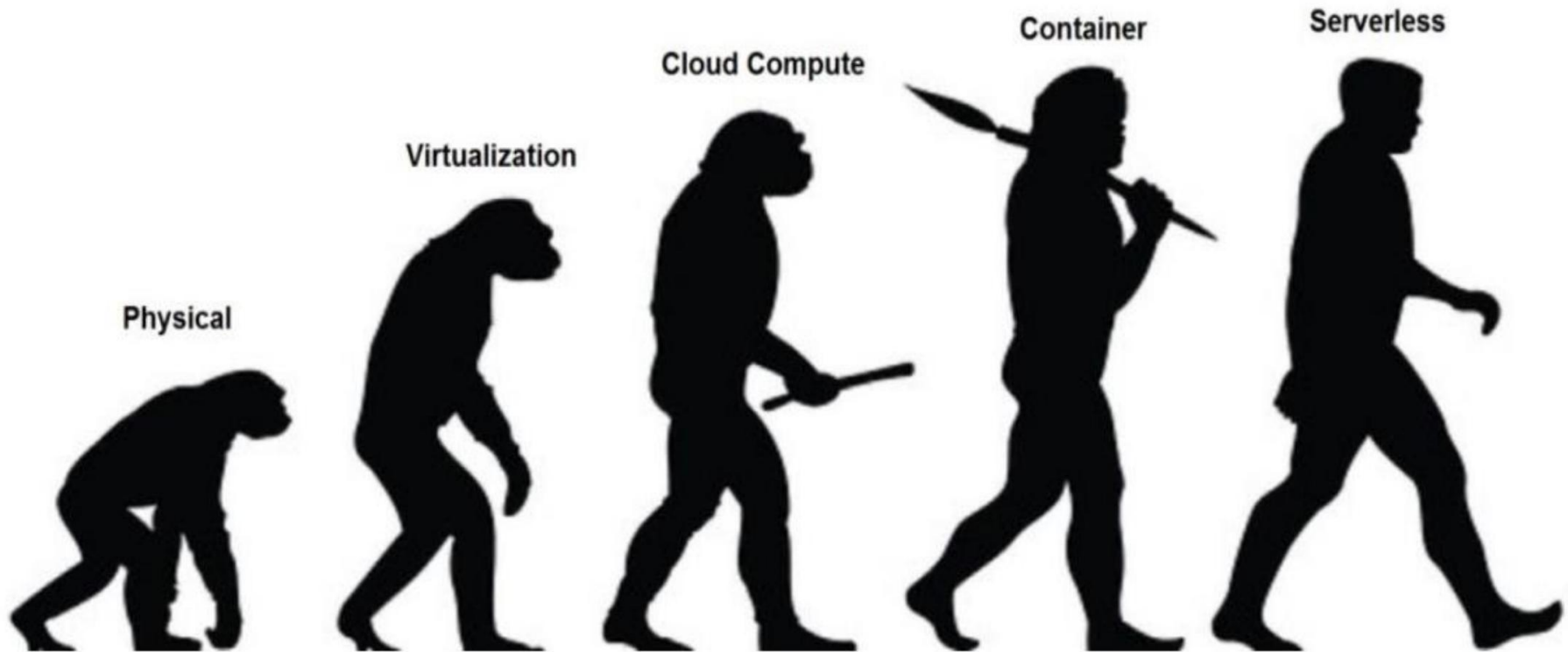
• من بين بنى الأنظمة والتطبيقات، تتمتع البنية الموجهة نحو الخدمة (SOA) بتقدير كبير بين مطوري البرامج. إنها بنية تصور بوضوح فكرة أن النظام يمكن أن يتكون من العديد من الخدمات المستقلة. لقد كتب الكثير عن SOA، ولكنه لا يزال مثيرًا للجدل وغير مفهوم جيدًا لأن المطورين غالبًا ما يخلطون بين فلسفة التصميم وبين التنفيذ والسمات المحددة.

• لا تفرض SOA استخدام أي تقنية معينة. وبدلاً من ذلك، فهو يشجع النهج المعماري الذي يقوم فيه المطورون بإنشاء خدمات قائمة بذاتها تتواصل عبر تمرير الرسائل.

• تعتبر الخدمات المصغرة والبنى التي لا تحتوي على خادم بمثابة أحفاد روجيين للبنية الموجهة نحو الخدمة. إنهم يحتفظون بالعديد من المبادئ والأفكار المذكورة أعلاه أثناء محاولتهم حل تعقيد البنى القديمة الموجهة نحو الخدمة.

كيف وصلنا إلى هنا ؟

البنى الموجهة نحو الخدمة والخدمات الدقيقة



القسم 2: ما هي الحوسبة بدون خادم؟

ما هي الحوسبة بدون خادم؟

تعريف

تعريف

تشير البنى بدون خادم إلى التطبيقات التي تعتمد بشكل كبير على خدمات الجهات الخارجية أو على تعليمات برمجية مخصصة تعمل في حاويات سريعة الزوال.

حذاري !

المصطلح "بدون خادم" مربك لأنه يوجد في مثل هذه التطبيقات أجهزة وعمليات خادم. يتمثل الاختلاف عن الأساليب الأساسية في أن الشركة التي تقوم بإنشاء ودعم ما يسمى بالتطبيق بدون خادم لا تهتم بالأجهزة أو العمليات، بل تقوم بالاستعانة بمصادر خارجية لموفر الخدمة (AWS، Google، وما إلى ذلك) وبالتالي تركز فقط على الجزء الوظيفي من التطبيق.

ما هي الحوسبة بدون خادم؟

تعريف

• الحوسبة بدون خادم هي تقنية، تُعرف أيضًا باسم الوظيفة كخدمة (FaaS)، والتي توفر لموفر السحابة إدارة كاملة للحاوية التي تعمل عليها الوظائف حسب الحاجة لتلبية الطلبات.

• ومن خلال القيام بذلك، تلغي هذه البنية الحاجة إلى أنظمة التشغيل بشكل مستمر وتكون بمثابة حسابات تعتمد على الأحداث. إن جدوى بناء تطبيقات قابلة للتطوير ضمن هذه البنية هائلة.

• تخيل أن لديك القدرة على كتابة التعليمات البرمجية وتنزيلها وتشغيلها، دون الحاجة إلى القلق بشأن البنية التحتية الأساسية أو تكوين البيئة أو صيانتها... فلاحتمالات لا حصر لها وتزداد سرعة التطوير بسرعة.

• باستخدام بنية بدون خادم، يمكنك نشر تطبيقات كاملة الوظائف وقابلة للتطوير في نصف الوقت الذي يستغرقه إنشائها من البداية.

ما هي الحوسبة بدون خادم؟

بدون خادم كحوسبة تعتمد على الأحداث

• الحوسبة المبنية على الأحداث هي نمط معماري يتم التركيز عليه الإجراء استجابةً لاستقبال الأحداث أو بناءً عليها.

• يفضل هذا النموذج الخدمات المترابطة بشكل غير محكم ويضمن عدم تنفيذ الوظيفة إلا عند تشغيلها.



• كما أنه يشجع المطورين على التفكير في أنواع الأحداث والاستجابات التي تحتاجها الوظيفة للتعامل مع تلك الأحداث قبل برمجة الوظيفة.

• في هذه البنية التي تعتمد على الأحداث، تعد الوظائف مستهلكة للأحداث لأنه من المفترض أن تنبض بالحياة عند وقوع حدث وتكون مسؤولة عن معالجتها.

• أمثلة: طلبات واجهة برمجة التطبيقات (API) والكائنات الموضوعية والمسترجعة من مخزن الكائنات، والتغييرات في عناصر قاعدة البيانات، والأحداث المجدولة، والأوامر الصوتية (على سبيل المثال، Amazon Alexa والروبوتات مثل Azure LUIS و AWS Lex وهما محركان لمعالجة اللغة الطبيعية).

ما هي الحوسبة بدون خادم؟

العمل كخدمة (FaaS)

تدار من قبل المورد 
تدار من قبل المنظمة/العميل 

طلب	طلب	طلب	طلب	طلب
بيانات	بيانات	بيانات	بيانات	بيانات
الوسيط	الوسيط	الوسيط	الوسيط	الوسيط
SE	SE	SE	SE	SE
الافتراضية	الافتراضية	الافتراضية	الافتراضية	الافتراضية
الخوادم	الخوادم	الخوادم	الخوادم	الخوادم
تخزين	تخزين	تخزين	تخزين	تخزين
الشبكات	الشبكات	الشبكات	الشبكات	الشبكات

على فرضية / سحابة
خاصة

IaaS

أجزاء من الخدمة



إدارة العلاقات مع

فا

س

ما هي الحوسبة بدون خادم؟

العمل كخدمة (FaaS)

تدار من قبل المورد 
تدار من قبل المنظمة/العميل 

طلب	طلب	طلب	طلب	طلب
بيانات	بيانات	بيانات	بيانات	بيانات
الوسيط	الوسيط	الوسيط	الوسيط	الوسيط
SE	SE	SE	SE	SE
الافتراضية	الافتراضية	الافتراضية	الافتراضية	الافتراضية
الخوادم	الخوادم	الخوادم	الخوادم	الخوادم
تخزين	تخزين	تخزين	تخزين	تخزين
الشبكات	الشبكات	الشبكات	الشبكات	الشبكات

على فرضية / سحابة
خاصة

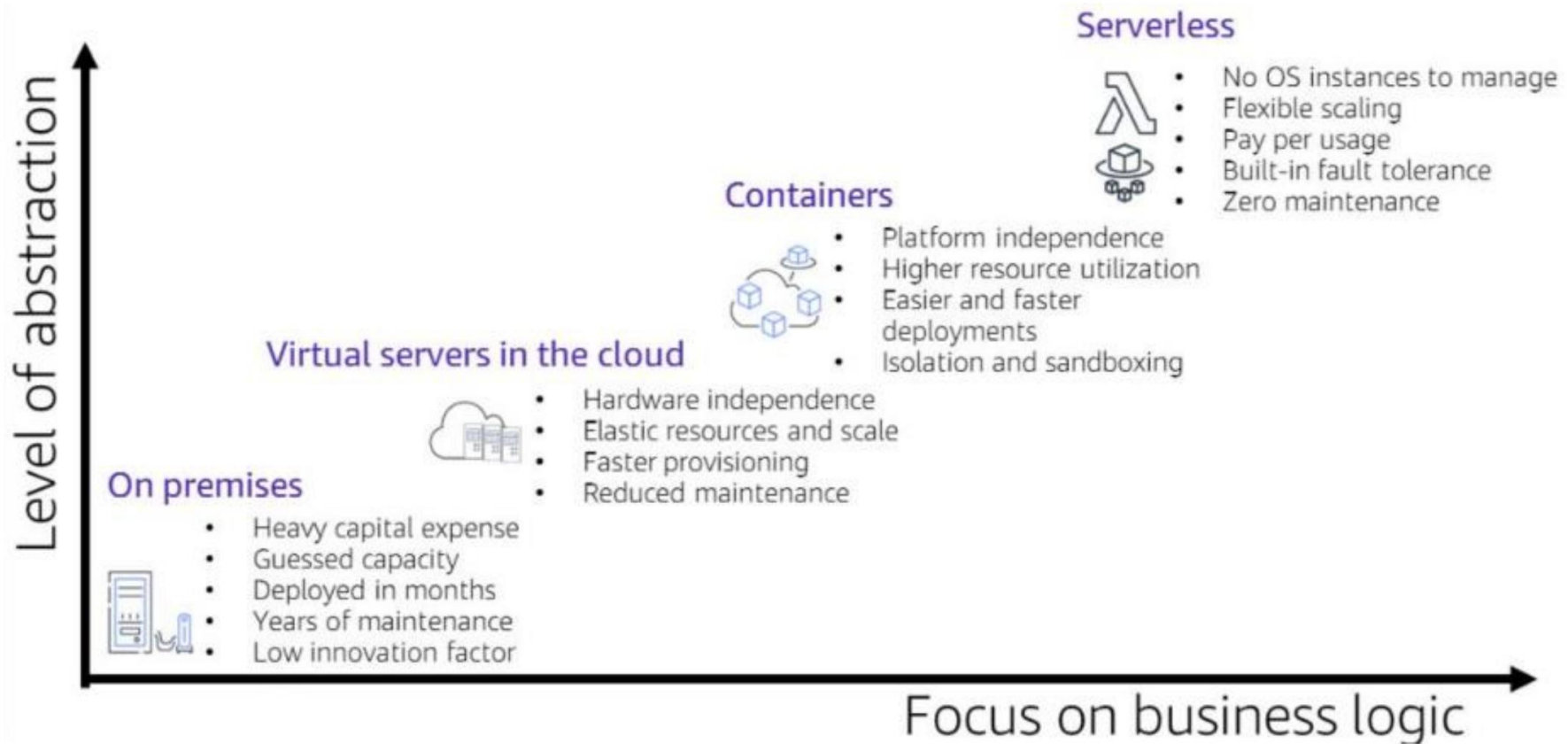
IaaS

أجزاء من الخدمة

إدارة العلاقات مع

فا

س



القسم 3: كيف تعمل الحوسبة بدون خادم؟

كيف تعمل الحوسبة بدون خادم؟

كيف تعمل الحوسبة بدون خادم

• لنفترض أن لدينا موقع ويب يمكننا من خلاله البحث عن الكتب المدرسية وشرائها. في البنية التقليدية، قد يكون لديك عميل، وخادم متوازن التحميل، وقاعدة بيانات للكتب المدرسية.

• في البنية بدون خادم، يمكن تغيير العديد من الأشياء، بما في ذلك الخادم وقاعدة البيانات. أحد الأمثلة على هذا التغيير هو إنشاء واجهة برمجة التطبيقات (API) المتوفرة على السحابة وتعيين طلبات أسلوب محددة لوظائف مختلفة.

• بدلاً من وجود خادم، أصبح تطبيقنا الآن يحتوي على وظائف لكل جزء من الوظائف والخوادم المتوفرة على السحابة والتي يتم إنشاؤها بناءً على الطلب.

• يمكن أن تكون لدينا وظيفة للبحث عن كتاب، وأيضًا وظيفة لشراء كتاب. يمكننا أيضًا اختيار تقسيم قاعدة البيانات الخاصة بنا إلى قاعدتي بيانات منفصلتين تتوافقان مع كلتاوظيفتين.

• سيؤدي حدث إلى تشغيل رمز التطبيق، ثم يقوم موفر السحابة بتخصيص الموارد لهذا الرمز ديناميكيًا، ويتوقف المستخدم عن الدفع عند انتهاء تشغيل الرمز.

كيف تعمل الحوسبة بدون خادم؟

طريقة لأداء الحوسبة بدون خادم

- هناك طريقتان رئيسيتان للحوسبة بدون خادم.
 - الأول هو من خلال الخلفية كخدمة (BaaS) حيث مجموعة متنوعة من الخدمات و تشكل تطبيقات الطرف الثالث تطبيقك.
 - والثاني هو عبر الوظيفة كخدمة (FaaS) حيث لا يزال المطورون يكتبون منطقًا مخصصًا من جانب الخادم، ولكن يتم تنفيذه في حاويات تتم إدارتها بالكامل بواسطة موفر السحابة.
- تجدر الإشارة أيضًا إلى أنه يمكنك إنشاء تطبيق بدون خادم تمامًا باستخدام هذه الطرق، أو التسوية مع تطبيق يتكون من مكونات خدمات صغيرة بدون خادم جزئيًا ومكونات تقليدية جزئيًا.
- الواجهة الخلفية كخدمة (BaaS) هي طريقة حوسبة بدون خادم يعتمد بشكل كبير على تطبيقات وخدمات الطرف الثالث.
- تتضمن الوظيفة كخدمة (FaaS) درجة أكبر من التحكم مقارنة بـ BaaS لأن المنطق من جانب الخادم لا يزال مكتوبًا بواسطة المطورين. ومع ذلك، بمجرد كتابته، يتم نشره في حاويات يديرها موفر السحابة، وهي الميزة الرئيسية للحوسبة بدون خادم.

القسم 3: كيف تختلف؟

بماذا يختلف عنه؟

تطوير

• تتغير عملية تطوير التطبيقات التي لا تحتوي على خادم بشكل طفيف عن كيفية تطوير نظام محلي. يمكن تحديد جوانب بيئة التطوير، بما في ذلك IDEs والتحكم في المصدر والإصدارات وخيارات النشر، بواسطة المطور محليًا أو مع موفر السحابة.

• الطريقة المفضلة للتطوير المستمر هي كتابة وظائف بدون خادم باستخدام IDE، مثل IntelliJ وEclipse وVisual Studio ونشرها في أجزاء صغيرة لموفر السحابة باستخدام واجهة سطر أوامر موفر السحابة.

- إذا كانت الوظائف صغيرة بما يكفي، فيمكن تطويرها في بوابة موفر السحابة. ومع ذلك، فإن معظمها لديها حد لحجم الوظيفة قبل طلب تنزيل المشروع بتنسيق ZIP.

• تعد واجهة سطر الأوامر (CLI) أداة تطوير قوية لأنها تجعل الوظائف بدون خادم وخدماتها الضرورية قابلة للنشر بسهولة وتسمح لك بمواصلة استخدام أدوات التطوير التي تريد استخدامها لكتابة وإنتاج الكود الخاص بك.

بماذا يختلف عنه؟

عمليات مستقلة

• هناك طريقة أخرى للتفكير في الوظائف التي لا تحتوي على خادم وهي الخدمات الصغيرة التي لا تحتوي على خادم

الخادم.

• تخدم كل وظيفة غرضها الخاص وتكمل العملية بشكل مستقل
وظائف أخرى.

• تعتبر الحوسبة بدون خادم عديمة الحالة وموجهة بالأحداث، وهذه هي الطريقة التي ينبغي بها أيضًا تطوير الوظائف.

• على سبيل المثال، في البنية التقليدية التي تتضمن عمليات CRUD API الأساسية (GET، POST، PUT، DELETE) يمكن أن يكون لديك نماذج قائمة على الكائنات مع تحديد هذه الأساليب على كل كائن. لا تزال فكرة الحفاظ على النمطية تنطبق على الطبقة بدون خادم. يمكن أن تمثل كل وظيفة طريقة واجهة برمجة التطبيقات (API) وتنفذ عملية.

القسم 4: مثال للاستخدام (AWS Lambda)

أمثلة الاستخدام (AWS Lambda)

يمكنك استخدام Lambda بطرق مختلفة. تعد تطبيقات الويب ومعالجة التدفق في الوقت الفعلي حالتين من حالات الاستخدام الشائعة. تشمل حالات الاستخدام الشائعة الأخرى ما يلي:

تأتي في شكل: يمكنك إنشاء واجهات خلفية بدون خادم باستخدام Lambda للتعامل مع طلبات الويب والهاتف المحمول وإنترنت الأشياء (IoT) وطلبات واجهة برمجة التطبيقات التابعة لجهات خارجية. يمكنك إنشاء واجهات خلفية باستخدام Amazon API Gateway و Lambda لمصادقة طلبات واجهة برمجة التطبيقات ومعالجتها.

معالجة البيانات: يمكنك استخدام Lambda لتشغيل التعليمات البرمجية استجابة للمشغلات، مثل تغييرات البيانات أو تغييرات حالة النظام أو إجراءات المستخدم.

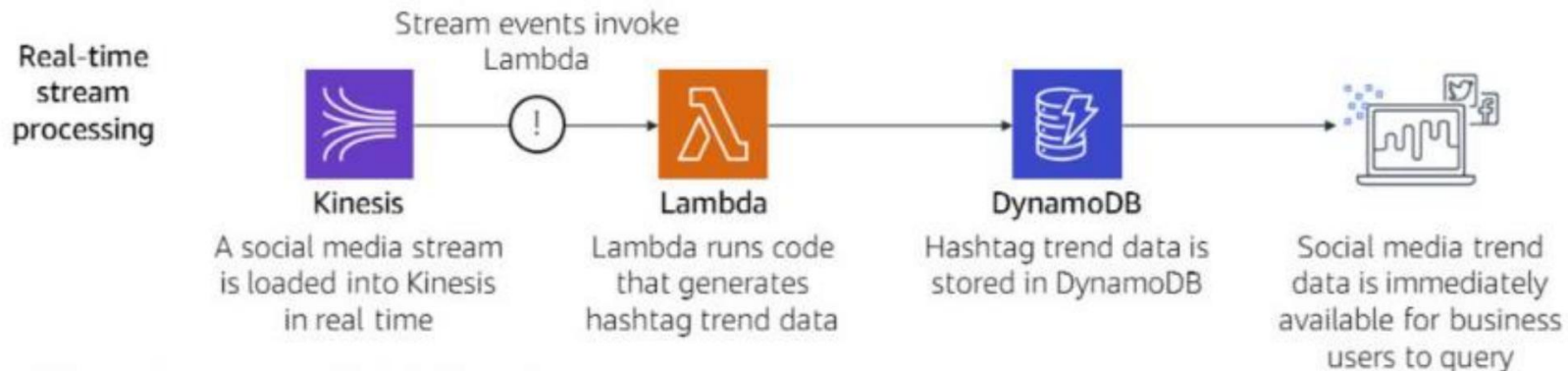
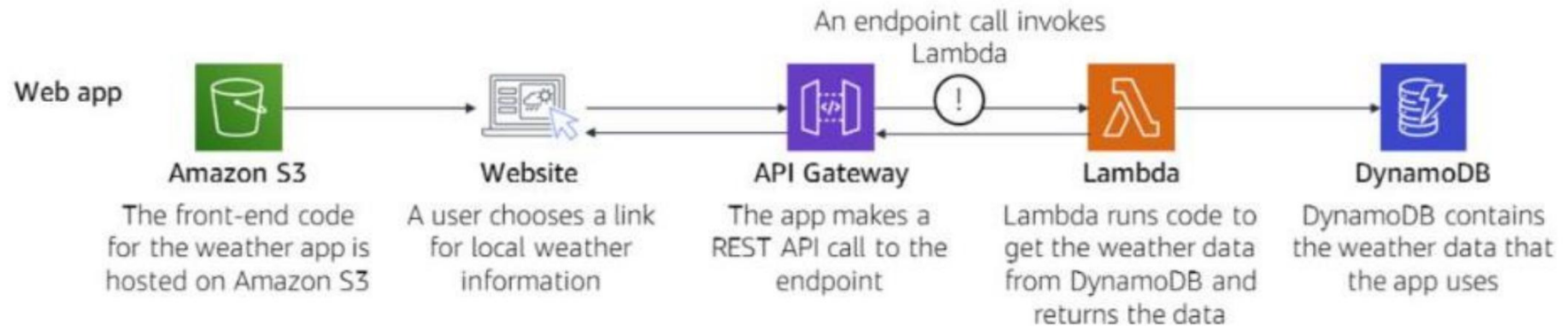
Chatbots: Amazon Lex هي خدمة لإنشاء واجهات محادثة في أي تطبيق باستخدام الصوت والنص. يستخدم Amazon Lex نفس تقنيات التعلم العميق التي تدعم Amazon Alexa ويتكامل مع Lambda.

Amazon Alexa: يمكنك استخدام Lambda لإنشاء الواجهة الخلفية لمهارات Alexa المخصصة

تمت أتمتة تكنولوجيا المعلومات: يمكنك استخدام Lambda لأتمتة العمليات المتكررة عن طريق استدعائها مع الأحداث أو تشغيلها وفقًا لجدول زمني محدد. يمكنك تحديث البرامج الثابتة للجهاز تلقائيًا، أو بدء تشغيل مثيلات Amazon Elastic Compute Cloud (Amazon EC2) وإيقافها، أو جدولة تحديثات مجموعة الأمان. بالإضافة إلى ذلك، يمكنك أتمتة عملية الاختبار والنشر.

أي شيء تريد القيام به مع الخوادم، ربما يمكنك القيام به باستخدام Lambda. استخدم Lambda وبدون خادم لإعادة ابتكار أسلوبك في التعامل مع التطبيقات الجديدة أو المحدث.

أمثلة الاستخدام (AWS Lambda)



القسم 4: المزايا والقيود

المزايا والقيود

فوائد

• التطوير والنشر السريع : نظرًا لأن كل البنية التحتية والصيانة والتوسيع التلقائي يتم التعامل معها بواسطة موفر السحابة، فإن وقت التطوير أصبح أسرع بكثير والنشر أسهل من ذي قبل. المطور مسؤول فقط عن التطبيق نفسه، مما يلغي الحاجة إلى تخطيط مقدار الوقت الذي سيتم إنفاقه في إعداد الخادم.

- سهولة الاستخدام: يرجع جزء كبير من هذه البساطة إلى الخدمات التي يقدمها موفرو الخدمات السحابية، والتي تسهل تنفيذ الحلول الشاملة. يتم إنشاء الأحداث اللازمة لتشغيل وظيفتك وتوفيرها بسهولة في البيئة السحابية وتتطلب القليل من الصيانة.

• تكلفة أقل : بالنسبة للحلول التي لا تعتمد على خادم، تتم محاسبتك على كل عملية تنفيذ بدلاً من تكلفة وجود جميع التطبيقات. هذا يعني أنك تدفع مقابل ما تستخدمه بالضبط. بالإضافة إلى ذلك، نظرًا لأن خوادم التطبيق تتم إدارتها وتوسيع نطاقها تلقائيًا بواسطة موفر السحابة، فإن سعرها أيضًا أرخص مما تدفعه داخل الشركة.

- قابلية التوسع المحسنة : مع الحلول التي لا تحتوي على خادم، يتم دمج قابلية التوسع تلقائيًا نظرًا لأن الخوادم تتم إدارتها بواسطة موفري خدمات خارجيين. وهذا يعني أنه تتم إزالة الوقت والمال والتحليل الذي يتم إنفاقه عادةً في إعداد القياس والموازنة التلقائية. بالإضافة إلى قابلية التوسع، يتم أيضًا زيادة التوفر حيث يحافظ مقدمو الخدمات السحابية على سعة الحوسبة عبر مناطق ومناطق التوفر. وهذا يجعل تطبيقك بدون خادم آمنًا ومتاحًا لأنه يحمي الكود من حالات الفشل الإقليمية.

المزايا والقيود

حدود

• التحكم في البنية التحتية: يتمثل أحد القيود المحتملة للبنية بدون خادم في الحاجة إلى التحكم في البنية التحتية. على الرغم من احتفاظ موفري الخدمات السحابية بالتحكم وتوفير البنية التحتية ونظام التشغيل، فإن هذا لا يعني أن المطورين يفقدون القدرة على تحديد عناصر البنية التحتية. في بوابة الميزات الخاصة بكل موفر خدمة سحابية، يمكن للمستخدمين اختيار وقت التنفيذ والذاكرة والأذونات ووقت انتهاء الصلاحية. بهذه الطريقة يتمتع المطور دائمًا بالسيطرة دون الحاجة إلى الصيانة.

• تطبيق خادم طويل الأمد: تتمثل إحدى مزايا البنى التي لا تحتوي على خادم في أنها مصممة لتكون سريعة وقابلة للتطوير ومعتمدة على الأحداث. ولذلك، فإن العمليات المجمعة طويلة الأمد ليست مناسبة لهذه البنية. لدى معظم موفري الخدمات السحابية مهلة مدتها خمس دقائق، لذا يتم إنهاء أي عملية تستغرق وقتًا أطول من هذا الوقت المخصص. تتمثل الفكرة في الابتعاد عن المعالجة المجمعة إلى وظائف سريعة وسريعة الاستجابة في الوقت الفعلي.

المزايا والقيود

حدود

• تقييد البائع: أحد أكبر المخاوف من الانتقال إلى التكنولوجيا بدون خادم هو تقييد البائع. وهذا خوف شائع مع أي انتقال إلى التكنولوجيا السحابية. تشعر الشركات بالقلق من أنه من خلال الالتزام باستخدام Lambda، فإنها تلتزم بـ AWS ولن تتمكن من الانتقال إلى موفر سحابي آخر أو لن تكون قادرة على تحمل تكاليف الانتقال مرة أخرى إلى موفر سحابي.

• البداية الباردة: تكمن المشكلة في "البداية الباردة" في أن الوظيفة تستغرق وقتًا أطول قليلاً للاستجابة لحدث ما بعد فترة من عدم النشاط. يميل هذا إلى الحدوث، ولكن هناك طرقًا للتغلب على البداية الباردة إذا كنت بحاجة إلى ميزة سريعة الاستجابة. إذا كنت تعلم أن وظيفتك سيتم تشغيلها بشكل دوري فقط، فإن إحدى الطرق للتغلب على البداية الباردة هي إنشاء برنامج جدولة يستدعي وظيفتك لتنشيطها بين الحين والآخر.

• البنية التحتية المشتركة: يمكن أن يكون هذا مصدر قلق من منظور الأعمال لأن الوظائف بدون خادم يمكن أن تعمل جنبًا إلى جنب بغض النظر عن ملكية الشركة.

خاتمة