

معماريّة التكامل الرشيق

أوقات تشغيل التكامل الخفيف القائم على الحاوية
container-based والموائم للخدمات المُصغرة
Microservices-aligned

IBM

تبنّي التحوّل الرقمي بالتكامل الرشيق المُتمركز حول نهج بذات الرشاقة حيث يُمكنك من التحرك سريعاً لتلبية مُتطلبات السُحب المُتعددة multicloud واللامركزية decentralization والخدمات المُصغرة microservices.

مُلخص موجز

يتعين على المُنظمات الساعية إلى تحقيق التحوّل الرقمي أن تتبني أساليب جديدة لاستخدام تقنيات التكامل ونشرها، مما يتيح لها سرعة التحرك بما يتناسب مع تحقيق أهداف السُحب المُتعددة multicloud واللامركزية decentralization والخدمات المُصغرة microservices. يجب أن تتحول طبقة تكامل التطبيق من أجل السماح للمُنظمات بأن تتصرف بجرأة عند بناء تجارب عملاء جديدة بدلاً من فرض نماذج للمعمارية وللتطوير تُبعد المُنظمات عن الوصول بإنتاجيتها لأقصى حد.

بدأت العديد من المُنظمات في تبني تقنيات التطبيق الرشيق مثل معمارية الخدمات المُصغرة، وتشعر اليوم في أن تشهد فوائد ذلك التحوّل. يُتم هذا النهج استراتيجياً واجهة برمجة التطبيقات API في المؤسسة، ويجعلها أسرع. يجب أيضاً على المؤسسات التجارية أن تسعى إلى استخدام هذا النهج لتحديث البنية التحتية القائمة لنقل الخدمات المؤسسية ESB لديهم بهدف الوصول إلى أساليب أكثر فعالية لإدارة خدمات التكامل خاصتهم وتشغيلها في سُحبهم الخاصة أو العامة على حد سواء.

هذا المُستند مأخوذ من [كتاب](#) يستكشف مزايا ما نُشير إليه بـ **معمارية التكامل الرشيق** - نهج قائم على الحاوية container-based وغير مُتمركز وموائم للخدمات المُصغرة microservices-aligned لحلول التكامل التي تُلبي مُتطلبات التحوّل الرقمي من رشاقة وقابلية للتوسّع وقدرة على التكيّف.

المحتويات

- 1 مُلخص موجز
- 2 تغيّرات طرأت على التكامل
- 2 الرحلة حتى الآن - المعمارية خدمية التوجّه SOA، ونواقل الخدمات المؤسسية ESBS، وواجهات برمجة التطبيقات APIs
- 3 الحالة بالنسبة لمعمارية التكامل الرشيق
- 3 الجانب 1: نشر التكامل على مستوى التفاصيل الدقيقة fine-grained
- 4 الجانب 2: ملكية التكامل اللامركزية
- 5 الجانب 3: البنية التحتية للتكامل القائم على السحابة cloud-native
- 5 كيف تغيّر وقت تشغيل التكامل الحديث ليلانم معمارية التكامل الرشيق؟
- 6 معمارية التكامل الرشيق لمنصة التكامل
- 11 منصة تكامل سحابة أي بي إم IBM Cloud

IBM

تغيّرات طرأت على التكامل

تُشير تقديرات شركة البيانات الدولية IDC إلى أن الإنفاق على مبادرات التحوّل الرقمي سيُمثّل فرصة سوقية قيمتها ٢٠ تريليون دولار على مدى الخمس سنوات القادمة^١، ما السبب وراء هذه الزيادة الهائلة في الإنفاق؟ إنها الحاجة الدائمة والمُتنامية إلى بناء تجارب عملاء جديدة بواسطة تجارب مُتصلة عبر شبكة من التطبيقات تستفيد من جميع أنواع البيانات.

ليست بالمهمة السهلة - جمع العمليات ومصادر المعلومات في الوقت والسياس الصحيحين أمر صعب على أقل تقدير، خاصة إذا أخذت في الاعتبار الاعتماد الجريء على تطبيقات أعمال البرمجيات كخدمة SaaS. يجب إدخال مصادر بيانات جديدة في العمليات التجارية لخلق تمييز تنافسي.

”قيادة تجارب عملاء جديدة، يجب على المنظمات استغلال مجموعة مُتنامية من التطبيقات والعمليات ومصادر المعلومات، التي تؤدي جميعها إلى توسيع كبير في نطاق حاجة المؤسسة للاستثمار في قدرات التكامل.”

أهمية تكامل التطبيقات بالنسبة للتحوّل الرقمي

إذا أخذت جدول أعمالك الخاص ببناء تجارب عملاء جديدة في الاعتبار، ورَكَزت على كيفية الوصول إلى البيانات وجعلها مُتاحة للخدمات وواجهات برمجة التطبيقات APIs التي تحفّز هذه المبادرات، سيكون بإمكانك أن تشهد العديد من الفوائد الهامة التي يقدمها تكامل التطبيقات:

- التناول الفعّال للتفاوت - الوصول إلى البيانات من أي نظام وبأي صيغة، واستخدامها لإنشاء تجانس بغض النظر عن مدى اختلاف نمو المشهد العام للسُحب المُتعددة multicloud خاصتك.
- خبرة النقاط النهائية - يتضمن التكامل الحديث مسائل ذكية بشأن البروتوكولات وصيغ البيانات المُعقدة، ويُدمج كذلك الذكاء فيما يتعلق بالكائنات والأعمال والوظائف الحقيقية داخل النُظم النهائية.

- الابتكار بواسطة البيانات - يرجع الكثير من الفضل في الابتكار الذي تشهده التطبيقات إلى الفرصة المُتاحة لها لجمع بيانات تتجاوز حدودها وخلق معاني منها، وهي سمة بارزة خصيصًا في معمارية الخدمات المُصغّرة.
- مكونات على مستوى المؤسسة enterprise-grade artifacts - تستمد تدفقات التكامل قدرًا هائلًا من القيمة من وقت التشغيل، مما يتضمن خصائص على مستوى المؤسسة من أجل استعادة الخطأ، وتجاوز الخطأ، وحصر السجلات، وتحليل الأداء، وأكثر من ذلك بكثير.

يتغير المشهد العام للتكامل لجاري متطلبات الحوسبة في المؤسسة والسوق، ولكن كيف انتقلنا من المعمارية خدمية التوجّه SOA ونواقل الخدمات المؤسسية ESBs إلى معمارية تكامل حديثة ومُحوّرة ورشيقة؟

الرحلة حتى الآن - المعمارية خدمية التوجّه SOA، ونواقل الخدمات المؤسسية ESBs، وواجهات برمجة التطبيقات APIs

قبل أن نتمكّن من التطلّع إلى مُستقبل التكامل الرشيق، نحتاج إلى فهم ما سبقه. ظهرت أنماط الـ SOA (المعمارية خدمية التوجّه) في مطلع الألفية، وفي البداية بشرّ القبول الكبير للمعايير التي بُنيت عليها الـ SOA بمُستقبل مشرق حيث يُمكن لكل نظام أن يكتشف أي نظام آخر ويخاطبه من خلال أنماط تعرّض تزامنية.

فلنتقدم بسرعة قليلًا، وستجد نفسك تمامًا في منتصف حركة الـ ESB (نواقل الخدمات المؤسسية)، وهي تقنية قُصد بها توفير إمكانية الوصل بنظم النهاية الخلفية بدءًا من نمط المحور والأفرع hub-and-spoke السابق. بينما نجحت العديد من المؤسسات في تنفيذ نمط الـ ESB، إلا إنه لم يلقَ قبولًا كبيرًا لدى المجال القائم على السحابة cloud-native. يبدو ثقيلًا ومُفتقرًا إلى الرشاقة. كيف انتقلنا من نقيض إلى آخر؟

تتلخص حقيقة الأمر في القليل من العوامل التي غالبًا ما تكون مُتعلقة ببعضها البعض:

- كانت المعمارية خدمة التوجّه SOA أكثر تعقيدًا من مجرد تنفيذ لنمط من أنماط الـ ESB خاصة لدى من سيمول برنامج على نطاق المؤسسة.
- شكّلت أنماط الـ ESB بنية تحتية واحدة للمؤسسة بأكملها مع تنصيب العشرات أو المئات من عمليات التكامل على مجموعة خادم الإنتاج. على الرغم من أن نمط الـ ESB لا يتطلب مركزية ثقيلة، إلا أن الأشكال الناتجة topologies تسقط فريسة لها بصفة شبه دائمة.

- ٢. **قابلية التوسع المرنة** – يُمكن ربط استخدامها للموارد تمامًا بنموذج الأعمال.
- ٣. **قدرة على التكيّف بشكل مُنفصل** – بالفصل المُلائم، لا تؤثر التغييرات المُدخلة على خدمة مُصغّرة واحدة على غيرها في وقت التشغيل.

مع أخذ تلك الفوائد في الاعتبار، كيف سيبدو الأمر إذا أعدنا تحيّل التكامل الذي يُنشر عادة في مستودعات متمرّكة centralized silos من منظور جديد قائم على معمارية الخدمات المُصغّرة؟ ذلك ما نُطلق عليه **"معمارية التكامل الرشيق"**.

- غالبًا ما فشلت أنماط ال-ESB المركزية في تحقيق التوفير الملحوظ الذي كانت تتمناه الشركات بما أنه لم يُكن مُمكنًا إعادة استخدام الواجهات انطلاقًا من مشروع إلى آخر.
- عانت المبادرات عبر المؤسسات cross-enterprise مثل ال-ESB للعثور على تمويل، وغالبًا ما طُبّق ذلك التمويل فقط على الخدمات التي ستكون قابلة لإعادة الاستخدام بحدٍ كافٍ لتغطية تكلفة إنشائها.

تُعرّف معمارية التكامل الرشيق بأنها "معمارية قائمة على الحاوية container-based، ولا مركزية، وموائمة للخدمات المُصغّرة microservices-aligned لحلول التكامل".

واجهت أنماط ناقل الخدمات المؤسسية ESB مشاكل في ضمان تمويل مُتواصل للمبادرات عبر المؤسسات بما أن هذه المبادرات لا تنطبق خصيصًا داخل سياق مبادرة أعمال مُعينة.

هناك ثلاثة جوانب متعلقة بمعمارية التكامل الرشيق، ولكنها مُنفصلة:

الجانب ١: نشر التكامل على مستوى التفاصيل الدقيقة fine-grained.

ما الذي قد نجنيه من تقسيم عمليات التكامل في ناقل الخدمة المؤسسية المنزل siloed ESB إلى أوقات تشغيل مُنفصلة؟

الجانب ٢: ملكية التكامل اللامركزية.

كيف علينا أن نُكيّف الهيكل التنظيمي لتحقيق استفادة أفضل من نهج التفاصيل الأدق؟

الجانب ٣: البنية التحتية للتكامل القائم على السحابة cloud native.

ما هي المزايا الأخرى التي يُمكننا الحصول عليها من خلال نهج قائم على السحابة بالكامل لتحقيق التكامل.

الجانب ١:

نشر التكامل على مستوى التفاصيل الدقيقة fine-grained

اتضح أن النشر المركزي لمحور التكامل أو أنماط ناقل الخدمات المؤسسية ESB حيث تُنشر جميع عمليات التكامل في زوج واحد ثقيل التغذية (HA) من خوادم التكامل، يمثّل عبءًا أمام المشاريع. يتضمن أي نشر على الخوادم المُشتركة خطر زعزعة استقرار الواجهات المهمة القائمة. لا يُمكن لأي مشروع مُنفرد أن يختار تحديث نسخة برنامج وسطيّ للتكامل من أجل الحصول على إمكانية دخول إلى خصائص جديدة.

نتج عنها أن خلق خدمات بيد فريق المعمارية خدمية التوجّه SOA المُتخصص هذا، أصبح عبءًا أمام المشاريع بدلًا من قيامه بدور المساعد الذي فُصد له أن يؤديه. مما أعطى بالتأكيد لنمط ناقل الخدمات المؤسسية ESB المتمركز سمعة سيئة بالمُصاحبة.

المعمارية خدمية التوجّه على النحو المُطبّق على أنماط ناقل الخدمات المؤسسية ESB هي مبادرة على نطاق المؤسسة من أجل إنشاء خدمات وواجهات برمجة تطبيقات APIs قابلة لإعادة الاستخدام ومتوفرة بشكل مُتزامن لإمكانية خلق تطبيقات جديدة بشكل أسرع عن طريق دمج البيانات القادمة من نُظم أخرى.

على الجانب الآخر، هناك معمارية الخدمات المُصغّرة microservices، وهي خيار مُناسب لكيفية اختيارك لأن تُنشئ تطبيقًا فرديًا بطريقة تجعله أكثر رشاقة وقابلية للتوسع وقدرة على التكيّف.

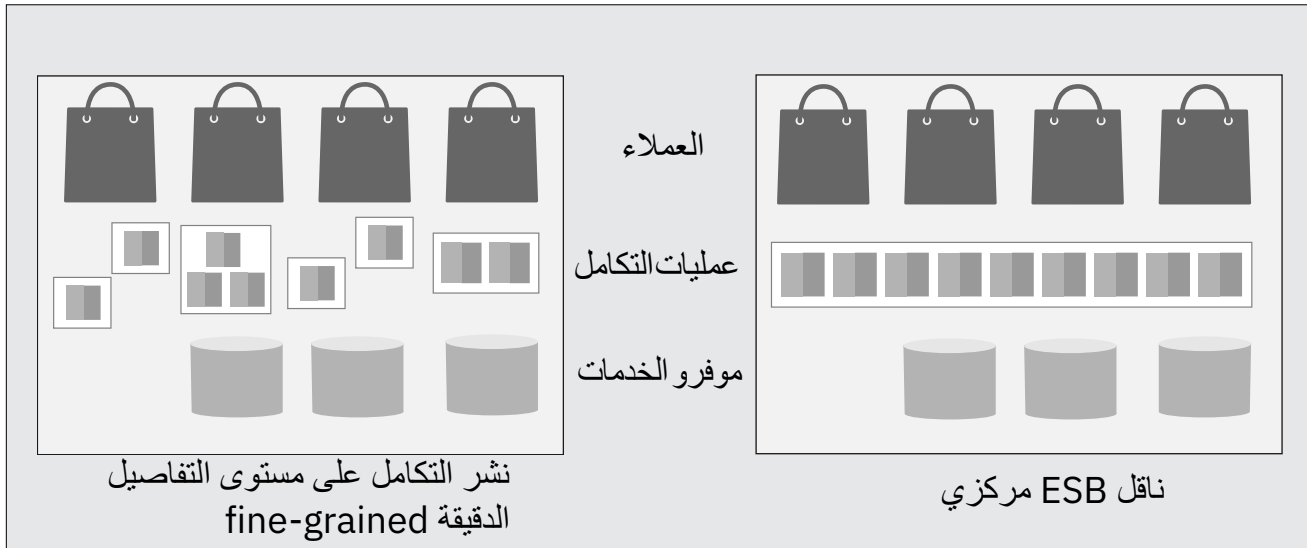
الحالة بالنسبة لمعمارية التكامل الرشيق

لماذا أصبحت مفاهيم الخدمات المُصغّرة مشهورة للغاية في مجال التطبيقات؟ هي تمثل نهجًا بديلًا لتطبيقات الهيكلية. بدلًا من أن يصبح التطبيق مستودعًا ضخمًا للأكواد العاملة على الخادم ذاته، يُصمّم التطبيق كمجموعة من المكونات الأصغر والعاملة باستقلالية كاملة.

تضمن معمارية الخدمات المُصغّرة microservices ثلاث فوائد هامة:

١. **رشاقة أكبر** - الخدمات المُصغّرة صغيرة بما يكفي لتُفهم منعزلة تمامًا ولتتغير باستقلالية.

يُمكننا تقسيم مكون ناقل الخدمات المؤسسية ESB على نطاق المؤسسة إلى أجزاء أصغر وأكثر قابلية للإدارة وأكثر تخصيصاً. ربما يُمكننا في بعض الحالات أن نصل أيضاً إلى وقت تشغيل واحد لكل واجهة نعرضها. توفر أنماط "نشر التكامل على مستوى التفاصيل الدقيقة Fine-grained" المذكورة حاويات مُخصصة بحجم مُناسب مع تقديم مستوى أفضل من الرشاقة وقابلية التوسع والقدرة على التكيّف والظهور مختلفة تماماً عن أنماط الـ ESB المركزية الماضية. يوضح الشكل ١ بمصطلحات بسيطة كيف يختلف نموذج الـ ESB المركزي عن نشر التكامل على مستوى التفاصيل الدقيقة fine-grained.



الشكل ١: مقارنة مبسطة بين نمط ناقل الخدمات المؤسسية ESB المركزي ونشر التكامل على مستوى التفاصيل الدقيقة.

عندما تُفكر في الرشاقة وقابلية التوسع والقدرة على التكيّف، من الهام أن تتذكر أنك لن تستطيع الاستفادة من هذه المزايا الخاصة بالتكامل على مستوى التفاصيل الدقيقة fine-grained دون وجود تكامل لامركزي.

تعرف على المزيد عن التكامل على مستوى التفاصيل الدقيقة fine-grained في كتابنا [Agile Infrastructure Architecture المتوفر الآن للتنزيل!](#)

الجانب ٢: ملكية التكامل اللامركزية

من التحديات المهمة التي واجهت المعماريّة خدمية التوجّه هي الطريقة التي قصدت بها أن تفرّض خلق فرق تكامل مركزي وبنية تحتية لإنشاء طبقة الخدمات.

خلق هذا الأمر اختلافاً مُستمرّاً في الوتيرة التي يُمكن للمشاريع أن تعمل وفقها حيث أنها لطالما تمتعت بفرق تكامل مركزي كاعتمادية. كان الفريق المركزي على دراية جيدة بتقنية التكامل الخاصة به، ولكنه غالباً لم يفهم التطبيقات التي يُجري التكامل عليها، لذا قد تكون ترجمة المتطلبات بطيئة ومُعرّضة للأخطاء.

يعتمد نشر التكامل على مستوى التفاصيل الدقيقة fine-grained على مزايا معماريّة الخدمات المُصغّرة. فلنعود إلى ما سردنا من مزايا الخدمات المُصغّرة على ضوء نشر التكامل على مستوى التفاصيل الدقيقة fine-grained:

- **الرشاقة** – يُمكن لفرق مُختلفة أن تعمل على عمليات التكامل باستقلالية دون الرجوع إلى مجموعة أو بنية تحتية مركزية، والذي سرعان ما قد يُصبح عقبة. يُمكن تغيير تدفقات التكامل الفردية وإعادة بنائها ونشرها بشكل مُستقل عن التدفقات الأخرى مما يُفعل زيادة الأمان في تطبيق التغييرات ويُحسّن سرعة الإنتاج لأقصى حد.
- **قابلية التوسع** – يُمكن للتدفقات الفردية أن تتوسع تلقائياً مما يسمح لك بالاستفادة من التوسع المرن الفعّال للبنى التحتية السحابية.
- **القدرة على التكيّف** – لا يُمكن للتدفقات التكامل المُنعزلة التي تُنشر في حاويات مُنفصلة أن تؤثر على بعضها البعض من خلال اقتباس موارد مُشتركة مثل الذاكرة أو الوصلات أو وحدة المعالجة المركزية CPU.

كانت بعض المنظمات تُفضّل أن تكون فرق التطبيقات هي المالكة لإنشاء خدماتها، ولكن التقنية والبنية التحتية للزمن لم تسمح بذلك.

يفتح الانتقال إلى نشر التكامل على مستوى التفاصيل الدقيقة fine-grained باباً يسمح بإمكانية توزيع ملكية إنشاء وصيانة عمليات التكامل. من المنطقي لفرق تطبيقات الأعمال أن تتولى مهمة التكامل مع تبسيط تنفيذ قدرات جديدة.

هل أثير فضولك بشأن نشر التكامل على مستوى التفاصيل الدقيقة fine-grained؟ ستجد الإجابة عن سؤالك في كتابنا [Agile Infrastructure Architecture book](#) [متوفر الآن!](#)

الاجاب ٣: البنية التحتية للتكامل القائم على السحابة cloud-native

تغيرت أوقات التشغيل الخاصة بالتكامل تغييراً كبيراً في السنوات الأخيرة. كُثر حدوث ذلك لدرجة أنه من الممكن استخدام أوقات التشغيل الخفيفة هذه بطرق قائمة على السحابة cloud-native حقاً. بهذا نُشير إلى قدرتهم على تسليم عبأ الكثير من الأليات التي سبق امتلاكهم لها، الخاصة بإدارة المجموعات والقياس والتوفّر، وإلى المنصة السحابية cloud platform التي يعملون بها.

يستلزم هذا أكثر بكثير من مجرد تشغيلهم في بيئة محوأة. هذا يعني أن عليهم أن يكونوا قادرين على العمل مُجتمعين وليس مُنفردين كـ"ماشية وليس حيوانات أليفة cattle not pets"، مُستفيدين أقصى استفادة من قدرات التزامن مثل الـ Kubernetes والعديد من إطارات العمل القياسية السحابية المعروفة الأخرى.

يؤثر اعتماد نهج الجمع "نهج الماشية cattle approach" على الأساليب التي ستتفاعل بها فرق الـ DevOps خاصتك مع البيئة والحل ككل، وسيخلق مُميزات مُتزايدة كلما انتقلت حلول أكثر إلى البنيات الخفيفة.

كيف تغيير وقت تشغيل التكامل الحديث ليلانم معمارية التكامل الرشيق؟

من الواضح أن معمارية التكامل لرشيق تتطلب نشر شكل التكامل نشرًا مُختلفًا للغاية. جانب أساسي لذلك هو وقت تشغيل التكامل الحديث الذي يُمكن تنفيذه في بيئة قائمة على الحاوية container-based، ومناسب بشكل جيد لتقنيات النشر القائم على السحابة cloud-native. تكاد تكون أوقات تشغيل التكامل الحديثة غير قابلة للتمييز عن مثيلاتها السابقة. فلنلق نظرة على بعض من تلك الاختلافات:

- **وقت تشغيل سريع وخفيف:** تعمل في حاويات containers مثل Docker، وهي خفيفة بقدر كافٍ بحيث يُمكنها أن تبدأ وتتوقف في ثوانٍ، وتُدار بسهولة بواسطة إطارات العمل الخاصة بالترامن مثل Kubernetes.
- **دون اعتمادية:** لم تعد بحاجة إلى قواعد بيانات أو قوائم انتظار الرسائل، بالرغم من أنه من الواضح كونها بارعة في الاتصال بهم إذا احتاجت ذلك.
- **تنصيب قائم على نظام الملفات:** من المُمكن تنصيبها بسهولة بوضع الثنائيات الخاصة بها في نظام ملفات، وتشغيلها، وهو أمر مثالي لنظم الملفات الطبقيّة لصور Docker images.
- **دعم أدوات DevOps:** يجب أن يكون وقت التشغيل بمثابة تكامل متواصل وجاهز للنشر. تنصيب البرامج النصية والخصائص القائم على الملفات، وإنشاؤها، ونشرها، وإعدادها لتمكين ممارسات "البنية التحتية ككود". ينبغي توفير البرامج النصية للقوالب template scripts من أجل أدوات الإنشاء والنشر القياسية لتسريع الضمّ إلى أنابيب الـ ديف وبس DevOps pipelines.
- **واجهة برمجة التطبيقات API أولاً:** يجب أن يكون بروتوكول الاتصال الأولي واجهات برمجة التطبيقات RESTful APIs. يتعين أن يكون كشف عمليات التكامل مثل واجهات برمجة التطبيقات RESTful APIs بسيطاً وقائماً على اتفاقيات مُشتركة مثل مواصفة واجهة برمجة التطبيقات المفتوحة Open API specification. يجب أن يكون استدعاء واجهات برمجة التطبيقات RESTful APIs في اتجاه المسار بذات البساطة بما في ذلك الاكتشاف عبر ملفات التعريف.
- **الربط الرقمي:** بالإضافة إلى الربط المؤسسي الغني الذي لطالما وفرته أوقات تشغيل التكامل، فيجب ربطها أيضاً بموارد حديثة. على سبيل المثال: قواعد البيانات NoSQL (مثل MonhoDb، وCloudant، وغيرهم)، وخدمات الرسائل مثل Kafka. فضلاً عن ذلك، تحتاج إلى الوصول إلى كتالوج غني بموصلات ذكية للتطبيقات كتطبيقات SaaS (البرمجيات كخدمة) مثل Salesforce.

من المسائل الرئيسية التي تلاحظها "جارتنر" هي أن منصة التكامل تُتيح للعديد من الأشخاص عبر المنظمة أن يعملوا داخل تجارب مُستخدمين تناسب احتياجاتهم على نحو أفضل. يعني هذا أنه بإمكان مُستخدمي الأعمال أن يكونوا مُنتجين في تجربة أبسط من شأنها أن تقودهم في مسار حل المشاكل المُباشرة، أما مُتخصصو تكنولوجيا المعلومات فيتمتعون بمستويات الخبراء الخاصة بالتحكم للتعامل مع السيناريوهات المؤسسية الأكثر تعقيداً. يُمكن لهؤلاء المُستخدمين بعد ذلك أن يعملوا سوياً أثناء إعادة استخدام هذه الأصول التي تشاركوها مع الحفاظ على الإدارة عبر جميع أجزائها.

تحمل تلبية احتياجات حالات الاستخدام الناشئة الخاصة بالتحول الرقمي نفس أهمية دعم مجموعات المُستخدمين المتنوعة. سيتناول الجزء الأكبر من هذا المُستند حالات الاستخدام الناشئة المذكورة، ولكن علينا أولاً تقديم المزيد من المعالجة فيما يخص القدرات الرئيسية التي يجب أن تساهم في منصة التكامل.

منصة تكامل سحابة أي بي إم IBM cloud

يجمع تكامل سحابة أي بي إم IBM Cloud™ Integration مجموعة قدرات التكامل الرئيسية في منصة مُجانسة تتميز بالبساطة والسرعة والجدارة بالثقة. تسمح لك بسهولة إنشاء عمليات تكامل وواجهات برمجة تطبيقات API قوية في دقائق، وتوفر أداءً رائداً وقابلية للتوسع، وتُقدّم قدرات شاملة لا تُضاهيها مع ضمان الأمن على مستوى المؤسسة.

دمجنا داخل منصة تكامل سحابة أي بي إم IBM Cloud Integration platform تخصصات التكامل الرئيسية الست، وكل منها الصفة الأفضل في مجالها. هي:

إدارة واجهة برمجة التطبيقات API:

تعرض وتدير خدمات الأعمال كواجهات برمجة تطبيقات APIs قابلة لإعادة الاستخدام بالنسبة لمجموعات مطوّرين مُعينين داخل وخارج مُنظمتك على حدٍ سواء. تتبنّى المنظمات استراتيجية واجهة برمجة التطبيقات API لتسريع تحقيق الفعالية التي يُمكنهم وفها أن يشاركوها ما يمتلكونه من بيانات وخدمات فريدة من أجل دعم تطبيقات وفرص أعمال جديدة بعد ذلك.

بوابة الأمن security gateway:

مدّ نطاق الربط والتكامل ليتجاوز المؤسسة بواسطة قدرات مُتقدمة بجهوية المنطقة المكشوفة DMZ-ready التي تحمي واجهات برمجة التطبيقات APIs والبيانات التي تنقلها والنظم وراءها.

تكامل التطبيقات:

تربط التطبيقات ومصادر البيانات محلياً on-premises أو في السحابة من أجل تنسيق تبادل معلومات الأعمال حتى تكون البيانات مُتاحة حينما وحيثما تقتضي الحاجة إليها.

• **التسليم المتواصل:** التسليم المتواصل مُمكن بواسطة واجهات سطر الأوامر command-line والبرامج النصية للقولاب template scripts، التي تتداخل في أدوات أنابيب الديويس وبس DevOps pipeline القياسية. يقلل هذا أكثر من حجم المعرفة اللازمة لتنفيذ الواجهات، ويزيد من وتيرة التسليم.

• **الأدوات المُعززة:** الأدوات المُعززة للتكامل تعني أنه يُمكن بناء مُعظم الواجهات بواسطة الإعداد فقط، وغالباً ما يقوم به أفراد لا يتمتعون بمعرفة سابقة عن التكامل. بإضافة قولاب لأنماط التكامل المعروفة، تتلخص أفضل ممارسات التكامل في الأدوات مما يُبسّط المهام أكثر. أصبحت الحاجة إلى مُتخصصين مُتعمقين في التكامل أقل، وربما يُمكن لفرق التطبيقات أن تؤدي بعض عمليات التكامل كما سنرى في القسم التالي عن التكامل اللامركزي.

أوقات تشغيل التكامل الحديثة مُناسبة تماماً للجوانب الثلاثة الخاصة بمعمارية التكامل الرشيق: النشر على مستوى التفاصيل الدقيقة fine-grained، والملكية اللامركزية، والبنية التحتية القائمة على السحابة cloud-native حقاً.

هل تُريد التعمق أكثر في البنية التحتية القائمة على السحابة cloud-native؟

[نزل كتابنا Agile Integration Architecture الآن!](#)

معمارية التكامل الرشيق لمنصة التكامل

ركزنا في هذا المُستند على خصائص تكامل التطبيقات على نحو نشرها في معمارية التكامل الرشيق. بالرغم من ذلك، لا يُمكن أن تُنفذ الكثير من حلول المؤسسات إلا بتطبيق العديد من قدرات التكامل المهمة. تجمع منصة التكامل (أو ما يُشير إليها بعض من المُحللين بأنها "منصة تكامل هجينة") هذه القدرات معاً حتى تتمكن المنظمات من بناء حلول أعمال بطريقة أكثر كفاءة واتساقاً.

ينفق الكثير من مُتخصصي المجال على أهمية منصة التكامل هذه. ملاحظات "جارتنر":

منصة التكامل الهجينة (HIP) هي إطار عمل خاص بالتكامل المحلي on-premises والقائم على السحابة cloud-based، وقدرات الإدارات التي تُمكن الأشخاص مُختلفي المهارات (المُتخصصين وغير المُتخصصين في التكامل) من أن يدعموا نطاق كبير من حالات استخدام التكامل.... ينبغي على قادة فرق التطبيقات المسؤولين عن التكامل الاستفادة من إطار عمل قدرات منصة التكامل الهجينة HIP لتحديث الاستراتيجيات والبنية التحتية الخاصين بالتكامل لديهم، وذلك ليستطيعوا معالجة حالات الاستخدام الطارئة للأعمال الرقمية.

إرسال الرسائل:

تضمن توفير معلومات في الوقت الحقيقي real-time من أي مكان وفي أي وقت عن طريق توفير تسليم رسائل يمكن الاعتماد عليه دون فقدان للرسائل أو تكرارها أو صعوبة استردادها في حالة وجود مشكلة في النظام أو في الشبكة.

تكامل البيانات:

تصل إلى البيانات، وتنظفها، وتجهزها من أجل خلق نظرة مُتسقة عن عمك داخل ورشة بيانات أو بحيرة بيانات للأغراض التحليلية.

النقل عالي السرعة:

نقل كميات ضخمة من البيانات بين النطاق المحلي والسحابة أو من سحابة إلى أخرى بسرعة وبصورة متوقعة بمستويات أمن مُعززة. يُسهل تحقيق سرعة أكبر لتبني المنظمات للمنصات السحابية عندما تكون البيانات ضخمة للغاية.

نأمل أن تكون قد حصلت من هذا المستند التشويقي على منظور أوسع للقدرات المهمة المُختلفة اللازمة كجزء من منصة التكامل، ونبذة عن المتطلبات التي تحتاجها تلك القدرات للعمل معًا، وتقدير لكيفية تبني معمارية التكامل الرشيق لتمكين رشاقة وقابلية للتوسع وقدرة على التكيف أكبر للمنصة.

منصة تكامل سحابة أي بي إم IBM cloud

خبرة تكامل فائقة



نقل عالي السرعة



تكامل البيانات



إرسال الرسائل والأحداث



تكامل التطبيقات



بوابة الأمن



دورة حياة واجهة برمجة التطبيقات

تحليلات | أمن | إدارة

على السحابة | هجين | محلي

الشكل ٢: منصة تكامل سحابة أي بي إم IBM cloud

تأكد من تنزيل الكتاب الإلكتروني الشامل e-book لتعرف أكثر من ذلك عن معمارية التكامل الرشيق.



© حقوق النشر والتأليف لشركة IBM Corporation ٢٠١٨

IBM Corporation
Global Technology Services
100 Route
10589 Somers, NY

تحريراً في الولايات المتحدة الأمريكية
أغسطس / آب ٢٠١٨

IBM وشعار IBM و **ibm.com** و iSeries و Power و System Storage و zEnterprise و TDMF و AIX و BladeCenter و pSeries هي علامات تجارية لشركة International Business Machines Corp. المسجلة في كثير من الولايات المتحدة في جميع أنحاء العالم. أسماء المنتجات والخدمات الأخرى قد تكون علامات تجارية لشركة IBM أو غيرها من الشركات. تتوفر قائمة بحالية بعلامات IBM التجارية على الإنترنت في "معلومات حقوق النسخ والعلامة التجارية" على موقع الويب www.ibm.com/legal/copytrade.shtml.

Linux هي علامة تجارية مسجلة لشركة Linus Torvalds في الولايات المتحدة الأمريكية أو البلدان الأخرى أو كلاهما.

Microsoft و Windows و Windows NT هي علامات تجارية لشركة Microsoft Corporation في الولايات المتحدة الأمريكية أو البلدان الأخرى أو كلاهما.

محتوى هذا المستند (بما في ذلك العملة أو مراجع السعر غير متضمنة للضرائب المطبقة) معمول به اعتباراً من تاريخ النشر الأولي، ويجوز لـ IBM تغييره في أي وقت. ليست جميع العروض متاحة في كل بلد تعمل فيه IBM.

بيانات الأداء ونماذج العملاء المذكورة مقدمة لأغراض توضيحية فقط. قد تختلف نتائج الأداء الحقيقية وفقاً لإعدادات وظروف تشغيل خاصة.

يتحمل المستخدم مسؤولية تقييم أي منتجات أو برامج أخرى والتحقق من تشغيلها مع منتجات أو برامج IBM.

المعلومات الواردة في هذا المستند مُقدّمة "كما هي" دون أي ضمان، صريح أو ضمني، ويشمل هذا الضمانات الخاصة بالتسويق أو الملائمة لغرض معين أو أي ضمان أو شرط لعدم المخالفة. تتوفر منتجات IBM بضمان وفق الشروط والأحكام للاتفاقيات المُقدّمة بموجبها.

قد ترد سعة التخزين المُتاحة الحقيقية سواء للبيانات المضغوطة أو غير المضغوطة، وستختلف، وقد تقل عن تلك المذكورة.

يرجى إعادة التدوير 