

□□□□□□ □□□□ DevOps: □□□□□□ □□□□ DevOpsin □
□□□□□□ □□□□□□□□□□ □□□□□□□□□□ □□□□□□
□□□□□□□□

بقلم سانجيف شارما

إنديانابوليس ، إنديانا ، John Wiley & Sons، Inc. حقوق الطبع والنشر © 2017 بواسطة

الفصل الخامس مطورون

يلعبون لـ

قيادة الابتكار

حركة سابر الجديدة لفيدرير

تبين أنه يمكنك تعليم حيل جديدة لكلب عجوز

هذا الصيف ، تلقى عشاق التنس مناورة جديدة من قبل أسطورة التنس روجر فيدرر البالغ من
ويظهر فيدرر وهو يتقدم في إرسال الخصم الثاني ليضرب SABR العمر 34 عامًا. يطلق عليه
نصف كرة في محاولة لإملاء السرعة وإحباط الخصوم. عندما يسمع أي شخص عنها ، فإن السؤال
؟ إنه اختصار SABR الأول هو: ماذا يعني

■ التسلل

■ الهجوم

■ بقلم

■ روجر

استخدمها فيدرر في فوزه المهيمن في سينسيناتي واستخدمها خلال مبارياته الأخرى في بطولة
الولايات المتحدة المفتوحة ، باستثناء فوزه في الدور الرابع على جون إيسنر صاحب الإرسال
، الضخم. تساعده هذه الخطوة في التمرکز بعد أن يضرب عودة عميقة ومتقطعة وتضع خصومه
مثل ريتشارد جاسكيه ، في حالة من عدم التوازن

المجاز ، 2015—

لقد تطورت رياضة الكريكت وابتكرت أكثر من أي رياضة أخرى موجودة. بينما تطورت الرياضات
الأخرى عن طريق تغيير القواعد أو المعدات ، أو تقديم مسرحيات جديدة ومبتكرة مثل □□□□

190 مسرحيات لقيادة الابتكار devops الفصل 5

لم يقم أي منها بإنشاء تنسيقات جديدة تمامًا للعبة. كان يطلق على لعبة الكريكت ، *Federer's SABR* في شكلها الأصلي □□□□□□ □□□□□□ ولا تزال (لا تزال) تلعب مع كل مباراة تدوم خمسة أيام! يتنافس كلا الفريقين على جولتين ، ويبدو أن اللعبة ستستمر إلى الأبد. ومع ذلك ، مع مرور الوقت تم إدخال ابتكارات جذرية إلى اللعبة لجذب الجمهور الذي لم يكن لديه الوقت أو الصبر لمشاهدة لعبة ، لمدة خمسة أيام في المرة الواحدة. نتج عن ذلك إنشاء تنسيق □□□□□□ □□□□□□ ، في عام 1971. انتهت المباراة بأكملها في يوم واحد ، حيث يضرب كل فريق دورًا واحدًا فقط لكل منهما. عدد الكرات □□□□□□ (مجموعة من ست كرات بواسطة نفس الرامي) محدود بما يتراوح بين 50 و 60 لكل جانب ، وبالتالي تحديد المدة التي يمكن أن تدوم فيها اللعبة. مع أحدث جيل من الجماهير المتمحورة حول التلفزيون والتي تتطلب تنسيقًا أكثر إثارة ، جلب عام 2003 تنتهي هذه الألعاب في حوالي . T20 أو Twenty20 المسمى ، over-ابتكارًا آخر للعبة: تنسيق 20 ثلاث ساعات ويمكن بالتالي إكمالها في المساء

إلى نمو هائل للجمهور في عالم لعب الكريكت. واحدة ، T20 أدت هذه الابتكارات ، وخاصة تنسيق السنوية T20 من البطولات الرياضية الأكبر والأكثر مشاهدة والأعلى في العالم هي بطولة تم بيع حقوق البث الخاصة بها وحدها مقابل .(IPL) □□□□□□ □□□□□□ □□□□□□ ، (Kalavalapalli) مليار دولار أمريكي في عام 2009 مقابل صفقة مدتها ثماني سنوات 1.63 دفع السوق الحاجة إلى الابتكار ، ودفع السوق أرباحًا لما نتج ! (2016)

أدى ظهور هذه التنسيقات الجديدة أيضًا إلى تغيير كيفية تشكيل الفرق لتتناسب كل تنسيق. تركز على الرغم من أن الفرق ، T20 نوادي الكريكت المحترفة اليوم بشكل حصري تقريبًا على تنسيق الوطنية لا تزال تلعب جميع التنسيقات الثلاثة. أدركت هيئات لعبة الكريكت في بلدان مثل إنجلترا الحاجة الأقصر وتنسيقات اليوم الواحد لاعبين T20 إلى فرق مختلفة لكل شكل من الأشكال. تتطلب تنسيقات أصغر سنًا وأكثر عدوانية ؛ هنا ، اللياقة البدنية وخفة الحركة تتفوق على كل شيء. يتطلب التنسيق الأطول لمدة خمسة أيام (مع المباريات التجريبية التي يتم لعبها بشكل أقل تكرارًا في الوقت الحاضر) لاعبين أكثر خبرة يتمتعون بمعرفة استراتيجية وقدرة على الاستمرار في الدورة على مدار خمسة أيام ؛ في هذه الحالة ، تتفوق المهارات الإستراتيجية والدفاعية على كل شيء. الشكل الأقصر يكون تكتيكيًا أكثر في مسرحياته ؛ الشكل الأطول هو أكثر استراتيجية. في كل حالة ، هناك حاجة إلى لاعبين ذوي مهارات مختلفة □□□□□□ □□□□□□ للفوز

تحسين للابتكار

بينما هذا الفصل مخصص للعب من أجل الابتكار ، لا يجب تجاهل الحاجة إلى التحسين. كما أناقش في الفصل 4 ، تم تصميم معظم التطبيقات التي تركز على □□□□□□□□ لتقديم قدرات مبتكرة جديدة تعتمد على التكنولوجيا للمؤسسات. تشمل الأمثلة ما يلي

□□□□□□ □□□□□□ □□□□□□ . هذه خدمات وإمكانات جديدة تمامًا لم يتم تقديمها ■ حتى الآن من قبل المؤسسة لمستخدميها - على سبيل المثال ، أحد البنوك التي تقدم خدمة تحويل الأموال من نظير إلى نظير لعملائها

191 مسرحيات لقيادة الابتكار devops الفصل 5

■ هذه نماذج أعمال جديدة للمؤسسة لتقديم خدمات تسهيل الخدمات التي ربما لم تستثمرها حتى الآن - على سبيل المثال ، مكتب تجاري لبنك استثماري يقدم خدمة لخدمة مستخدمة داخليًا لشركائه ، من خلال تعريض واجهات برمجة التطبيقات

■ هذه تقنية جديدة للتفاعل مع المستخدمين. من الأمثلة الجيدة على ذلك الخدمات المصرفية . الرقمية ، حيث يمكن للعملاء استهلاك جميع خدماتهم المصرفية تقريبًا عبر الهاتف المحمول بدءًا من سداد الفواتير وحتى الإيداع بال شيكات وحتى التقدم بطلب للحصول على قرض -

■ هذا ينطوي على الوصول إلى أسواق جديدة تمامًا للمؤسسة. أحد الأمثلة على ذلك هو تقديم البنوك خدمة تحويل الأموال من نظير إلى نظير لجذب عملاء جدد ، مثل جيل الألفية الذين يحتاجون إلى خدمات مصرفية ولكنهم لا يستخدمون البنوك التقليدية

إذا كنت تطبق الحاجة إلى الابتكار في شركة ناشئة ، شركة جديدة ليس بها أنظمة حالية أو قديمة تقدم وظائف أو خدمات تجارية ، فإن المحادثة ستقتصر على التطبيق الذي يركز على الابتكار فقط. ومع ذلك ، فإن معظم المؤسسات ، من الشركات الكبيرة إلى الشركات الصغيرة ، لديها أنظمة برمجيات حالية ، تقدم بالفعل خدمات الأعمال الأساسية ، وبالتالي ، قيمة الأعمال للعملاء. في مثل هذه المؤسسات تعتمد التطبيقات المبتكرة دائمًا تقريبًا على خدمات الأعمال الأساسية التي تقدمها هذه الأنظمة القديمة الموجودة مسبقًا. يحتاج تطبيق تحويل الأموال المبتكر الذي يقدمه أحد البنوك إلى الأنظمة القديمة الحالية لتقديم الخدمات الأساسية مثل إدارة الهوية واكتشاف الاحتيال وخدمات مكافحة غسيل الأموال وما إلى ذلك. إذا كانت هذه الخدمات يتم تقديمها بالفعل لبقية التطبيقات المصرفية من خلال هذه الأنظمة الأساسية ، فلماذا يقوم البنك بإنشاء خدمات جديدة لتقديمها إلى التطبيقات الجديدة التي تركز على الابتكار؟ الاعتماد غير قابل للتفاوض. وبالتالي فهي تبعية معمارية في كل تطبيق مبتكر

يمكن أن تصبح هذه الأنظمة الخلفية عبئًا على حاجة التطبيقات التي تركز على الابتكار إلى السرعة إذا كان التطبيق المبتكر يتطلب خدمة خلفية لإجراء تغيير لتلبية احتياجاته ، فعندئذٍ إذا كان التطبيق القديم لديه دورة زمنية أبدا لتقديم التغيير ، يمكن أن يبطئ وقت تسليم التطبيق المبتكر. يصبح تحسين خط أنابيب التسليم والإمكانات لتقديم تغييرات على أنظمة الواجهة الخلفية بطريقة أكثر رشاقة وفعالية ضرورة من أجل الابتكار وإزالة في سرعة توصيل التطبيق الذي يركز على الابتكار وتلك الخاصة بالتطبيقات القديمة. يصبح شرطًا أساسيًا للابتكار

السبب الآخر الذي يجعل الأنظمة الخلفية بحاجة إلى التحسين هو الموارد. في معظم المؤسسات الكبيرة ، يتم إنفاق غالبية موارد تكنولوجيا المعلومات على الأنظمة وصيانتها. إذا تم استخدام جميع الموارد المتاحة - الأشخاص والأموال - بشكل كامل في تشغيل وصيانة التطبيقات الحالية فلن يتوفر أي منها للاستثمار في الابتكار. يصبح النظام الخلفي لتحرير الموارد للاستثمار ،

في الابتكار أمرًا ضروريًا. متلازمة أوبر

هناك ظاهرة جديدة تدفع بالحاجة إلى الابتكار عبر عالم الأعمال. تُعرف باسم التهديد . هذه هي الحاجة الملحة لمعظم المؤسسات ، كبيرها وصغيرها ، للتصرف ضد التهديد

192 مسرحيات لقيادة الابتكار devops الفصل 5

في صناعة سيارات ، Uber المتصور لمنافس قادم من العدم إلى فضاءهم ، تمامًا كما فعل تطبيق Uber الأجرة. مؤسس أوبر ، ترافيس كالانيك ، ليس من صناعة سيارات الأجرة. قبل أن يؤسس شركة Akamai والتي باعها لشركة ، Red Swoosh بدأ شركة مشاركة ملفات من نظير إلى نظير تسمى ، (Roettgers ، 2008) في عام 2007.

هذا الخوف من أن تكون □□□□□□ على الصناعات التي تبدو عرضة للاضطراب. وجد استطلاع لقيمة الأعمال أن 60 في المائة من الرؤساء التنفيذيين العالميين يتوقعون IBM حديث أجراه معهد أن يأتي منافسهم القادم من خارج صناعتهم (معهد آي بي إم لقيمة الأعمال ، 2016) ، مع نموذج أعمال جديد ومبتكر ، مسلح ربما بما لا يزيد عن تطبيق جوال — مثلما فعلت أوبر. هذا سبب مقنع للعمل مع معظم المؤسسات - من الخدمات المالية إلى التجزئة إلى القطاع العام. هذه الحاجة للابتكار /تستند إلى موضوع: □□□□□□□□ □□ □□□□□□□□

هذه الحاجة إلى الابتكار تدفع في الواقع إلى الحاجة إلى تبني منصات وعمليات تقنية جديدة مصممة والنظام الأساسي المستندة إلى مجموعة النظراء كخدمة مثالين DevOps للابتكار بسرعة. تعد بالتأكيد. هذه الحاجة إلى الابتكار تقود بدورها إلى إدراك أن هذه المنظمات بحاجة إلى تحسين قدراتها القديمة في تقديم التطبيقات. إنهم بحاجة إلى القيام بذلك لتحرير الموارد - الأشخاص والأموال - التي يمكن استثمارها في الابتكار. كما يتعين عليهم القيام بذلك لضمان ألا يصبح التسليم البطيء التقليدي هو الحل مرة أخرى DevOps. عبناً على قدرتهم على الابتكار بسرعة

الابتكار ودور التكنولوجيا

من المفاهيم الخاطئة الشائعة في عالم اليوم ، حيث يتم توجيه معظم الابتكار ، هو دور التكنولوجيا في Apple ، دفع الابتكار. نادراً ما يكون الابتكار في التكنولوجيا نفسها. نعم ، هناك مبتكرون تقنيون مثل حيث جوهر أعمالهم هو ، Tesla وبالطبع ، Salesforce.com ، و Netflix ، و IBM ، و التكنولوجيا الجديدة والمبتكرة. ولكن بالنسبة للغالبية العظمى من الشركات المبتكرة - من شركات إلى بانعي التجارة ، Under Armor إلى شركات الملابس مثل ، Regeneron الأدوية مثل إلى الوكالات الحكومية ، Marriott International إلى فنادق مثل ، Alibaba الإلكترونية مثل فإن التكنولوجيا هي المنصة التي يقدمون فيها خدمات - HM Revenue and Customs مثل الأعمال لمستخدميهم. لا يصبح دور تكنولوجيا المعلومات هو تقديم تقنيات تخريبية ومبتكرة ، بل تقديم منصة مستقرة تتسم بالرشاقة والفعالية ويمكن التنبؤ بها ويمكن من خلالها تقديم خدمات الأعمال المبتكرة والمزعة. إذا كانت خدمات الأعمال تتطلب تطوير تقنيات مبتكرة ومعقدة ، فهذا ما تحتاجه تكنولوجيا المعلومات لتقديمه. لا ينبغي أن يقدموا تقنية مبتكرة من أجل تقديم تكنولوجيا مبتكرة ولكن لتمكين الابتكار للأعمال التجارية

ابتكار نماذج أعمال جديدة

من هو عميلك؟

194 مسرحيات لقيادة الابتكار devops الفصل 5

نموذج العمل لكيفية كتابة الكتاب مبتكر أيضاً. بدلاً من إجراء مقابلات مع العشرات من قادة الأعمال من جميع أنحاء العالم لتطوير أفكارهم ومقترحاتهم وتنقيحها والتحقق من صحتها ، أنشأوا وهو موقع لبناء المجتمع ، وبدأوا في جمع رسوم للأشخاص للانضمام ، Ning.com مجتمعاً على إلى كتاب كمبدعين مشاركين. استمروا في رفع السعر حتى انتقل من 24 دولارًا إلى 243 دولارًا. في هذه المرحلة ، كان لديهم 470 شخصاً من 45 دولة على استعداد للمساهمة في الكتاب ، من خلال مراجعة المحتوى وإعادة كتابته ، والمساهمة بدراسات الحالة ، والمساعدة في تصميم الكتاب المصور بشدة. وكانت النتيجة هي أفضل الكتب مبيعاً التي أصبحت من الأمور التي يجب قراءتها ، للشركات الناشئة والمبتكرين في جميع أنحاء العالم ، ونموذج جديد تماماً لكتابة الكتب (ويلسون 2010).

من منظور تكنولوجيا المعلومات ، عندما يفكر الناس في الابتكار ، فإنهم يفكرون في تطبيق يميزات ، جديدة ورائعة ، واحدة من خلال تقنياتها المتطورة وتصميم تجربة المستخدم التي تم ضبطها بدقة تعطل الصناعة التي تستهدفها. ومع ذلك ، يحدث المزيد من الابتكار من خلال تعطيل نموذج الأعمال لديها تطبيق جوال جيد حقاً وسهل Uber وليس التكنولوجيا. على سبيل المثال ، على الرغم من أن الاستخدام ، إلا أنه لم ينجح بسبب الميزات أو سهولة استخدام التطبيق. إن امتلاك تطبيق سهل الاستخدام ، هو رسم القبول في عالم تطبيقات الأجهزة المحمولة. حيث كان الابتكار في أوبر في نموذج أعمالها وللوصول إلى هناك ، كان على كالتيك وفريقه تجربة نماذج أعمال مختلفة (ولا تزال كذلك). لقد بدأوا من خلال توفير تطبيق ركوب الخيل الذي وجد سيارات ليموزين خاملة يمكن استئجارها بتكلفة أقل على الرغم من UberBLACK. باسم Uber بكثير ، بين رحلاتهم المجدولة. لا يزال هذا موجوداً داخل من ذلك ، فقد تطور العمل الأساسي من خلال التجارب ليصبح أوبر كما تعرفها اليوم: السائقون غير المحترفين يقدمون رحلات مقابل رسوم في سياراتهم الشخصية

هنا ، يصبح دور تكنولوجيا المعلومات دور المنصة التي يمكن أن تحدث بها تجارب نموذج الأعمال يمكن تقديم العديد من نماذج الأعمال والتراجع عنها حسب الحاجة. يمكن جمع التعليقات السريعة وتحليلها لتقييم نتائج كل تجربة لتحديد ما إذا كان يجب التراجع أو التراجع. يمكن استخدام هذه التعليقات لمعرفة النموذج الذي يعمل مع ملف تعريف المستخدم وأيهما لا يعمل. الابتكار ليس التكنولوجيا نفسها ، عندما يقف الزبون الذي أشاد بسيارة أجرة عند زاوية شارع ، في ليلة ممطرة ، في انتظار رحلتهم فإنهم لا يهتمون بالتكنولوجيا المستخدمة. إنهم لا يهتمون بواجهة برمجة تطبيقات الخرائط التي تعرض موقع رحلتهم على الخريطة أو ما إذا كانت بعض الخدمات الموجودة على الواجهة الخلفية للهاتف المحمول معبأة في □□□□□□ . يصبح الهدف من تكنولوجيا المعلومات هو هدف □□□□□□□□□□ لمنصة بسيطة وفعالة يمكن من خلالها تسليم نماذج الأعمال وتغييرها سريعاً لإجراء التجارب.

ابتكار نماذج تفاعل المستخدم الجديد

هناك مسار شائع آخر للابتكار وهو استكشاف نماذج مشاركة جديدة للمستخدم. قد تهدف هذه النماذج الجديدة إلى تقديم قدرات تجارية جديدة للعملاء الحاليين أو الاستحواذ على أسواق جديدة للقدرة الحالية. تلعب التجربة مرة أخرى دوراً رئيسياً هنا لاكتشاف نماذج تفاعل جديدة

فكر في مثال آخر من العالم الحقيقي. كانت مكدونالدز ، شركة الامتياز للوجبات السريعة ، تحاول (The Movie Network) اتباع نهج جديد للتواصل مع عملائها ، من خلال التوسع في بيع البقالة ،

195 مسرحيات لقيادة الابتكار devops الفصل 5

لقد طوروا متجرًا آليًا بالكامل ، مباشرة في طريق المستهلكين الذين كانوا في طريقهم إلى (2014) المنزل من العمل. يحتوي المتجر الآلي ، الموضح في الشكل 5-1 ، على آلات بيع تباع كل شيء يبيعه حتى أن DVD. متجر صغير: من الأطعمة الطازجة إلى الحليب إلى الوجبات الخفيفة إلى أقراص المتاجر صُممت لتكون محمولة ؛ سمح لهم ذلك بتحريك المتاجر بسهولة وتجربة المواقع. لسوء الحظ ، كانت التجربة فاشلة. على ما يبدو ، لا يمكن بيع كل شيء من خلال آلة البيع الآلي - وليس الحليب ، أو الأطعمة الطازجة ، بالتأكيد

بدلاً من التخلي عن المفهوم بأكمله ، نظر المسؤولون التنفيذيون إلى ما نجح. كانت مبيعات وتأجير من آلة البيع. من يحتاج إلى إنسان ليقوم DVD تعمل. لم يتردد أحد في استئجار قرص DVD أقراص ؟ قررت الشركة التخلص من جميع آلات البيع الآلي الأخرى والاحتفاظ بآلة بيع DVD بفحص قرص ! وألعاب الفيديو Blu-ray و DVD وأكشاك تأجير أقراص ، Redbox وهكذا ولدت شركة DVD.



(Imgur ، 2013) في واشنطن العاصمة ، 2002 Redbox الشكل 1-5: كشك البقالة

في هذا المثال ، ظل نموذج العمل الفعلي ، في جوهره ، دون تغيير - توفير كشك الخدمة الذاتية لبيع المنتجات للعملاء. ما تغير هو النظام الأساسي التكنولوجي الذي تم تقديم خدمات الأعمال عليه. تم تحسينها بالتجربة ، وتغيير كيفية تفاعل العملاء معها ، وتم تقليصها من كشك واسع النطاق يسلم فقط. تم تحسين هذه المنصة منذ ذلك الحين لتشمل DVD البقالة للبيع إلى كشك أصغر يقدم أقراص وألعاب الفيديو. يجب أن يكون فريق تسليم المنصة مرناً وأن يكون قادراً على Blu-ray أيضاً أقراص

، مثلما أنظر إلى بعض الموضوعات الأساسية التي تتسج من خلال المسرحيات المختلفة في الفصل 4 سأقدم الآن بعض الموضوعات الأساسية التي تشكل أساس المسرحيات المتعددة التي تركز على الابتكار. هذه كالتالي

تحقيق تقنية المعلومات متعددة السرعات

بناء الشيء

التجريب

تسليم أنظمة

تحقيق ذلك متعدد السرعات

الرياضيون متعدّدو الرياضة

سيطر لاعبو كرة القدم الذين شاركوا في رياضات إضافية أثناء وجودهم في المدرسة الثانوية على مساء الخميس 2016 NFL الجولة الأولى من مشروع

كان 28 من أصل 31 لاعباً تم اختيارهم من الرياضيين متعددي الرياضات في أيامهم التحضيرية بما في ذلك المصنف الأول بشكل عام جاريد جوف ، الذي لعب البيسبول وكرة السلة في مدرسة ، مارين الكاثوليكية الثانوية في جرينبرا ، كاليفورنيا

جوف هو واحد من 12 رياضياً سابقاً في ثلاث رياضات. انضم إليه لاعبون من أمثال زملائه في الوسط كارسون وينترز (البيسبول ، كرة السلة) ، المتلقي كوري كولمان (كرة السلة ، المسار) ورجل الخطاف جاك كونكلين (كرة السلة ، المسار)

—سيپواك ، 2016—

دون الخوض في الجدل حول ما إذا كان يجب على الرياضيين الشباب ممارسة رياضات متعددة لتطوير مهارات متعددة التخصصات ، في عالم تكنولوجيا المعلومات ، والقدرة على دعم خطوط أنابيب توصيل متعددة ، بأوقات دورات مختلفة وسرعات توصيل متفاوتة ، مما يؤدي إلى

عبر خطوط أنابيب توصيل DevOps حقيقة. كما ناقش في الفصل 4 ، يتطلب العالم الحقيقي اعتماد متعددة لدعم السرعات المختلفة والأنظمة الأساسية التقنية التي قد تعتمدها فرق التسليم المختلفة. نظرًا لأن معظم أنظمة الأعمال تتطلب خدمات من تطبيقات متعددة ، فإن التنسيق عبر خطوط أنابيب التسليم هذه أمر ضروري. يعد تحديد وفهم بين الخدمات المختلفة التي تقدمها خطوط أنابيب التسليم المختلفة أمرًا ضروريًا لضمان تنسيق تسليم وإصدار كل تطبيق مع الخدمات والتطبيقات الأخرى التي يعتمد عليها التطبيق أو التي تعتمد عليها. بالإضافة إلى ذلك ، يعد ضمان عبر خطوط أنابيب التسليم هذه أمرًا ضروريًا للتقاط وتحليل حالة كل خط أنابيب تسليم بشكل فعال وتحديد تأثير أي مشكلات أو تأخيرات أو تغييرات في خطط التسليم والإصدار لأي تطبيق أو خدمة بشكل استباقي.

المحددة جيداً والتي يتم استخدامها للاتصال بين التطبيقات أو الخدمات. تسمح واجهات برمجة التطبيقات هذه بتطوير تطبيق دون معرفة تفاصيل التنفيذ التفصيلية للتطبيق الآخر والقضاء على عمليات التكامل من نقطة إلى نقطة التي تتغير مع كل إصدار جديد من التطبيق. عادةً لا تتغير واجهات برمجة التطبيقات كثيرًا بمرور الوقت ، مما يسمح للواجهات المعمارية بين التطبيقات بالبقاء دون تغيير. تصميم وتنفيذ واجهات برمجة التطبيقات الجيدة هو موضوع مسرحية سأناقشها لاحقاً في هذا الفصل

3. □□□□□□ □□□□□□□□□□ . أناقش □□□□□□ □□□□□□ بالتفصيل في الفصل 4. هذه منطقة تحتاج إلى توحيد معياري عبر جميع خطوط أنابيب التسليم. القدرة على نشر أي تطبيق على

البيئة المناسبة ، عند الحاجة ، ضرورية. عندما تحتوي خطوط أنابيب التسليم المختلفة على أدوات نشر مختلفة ، يؤدي ذلك إلى حدوث تعقيدات عند الحاجة إلى تنسيق عمليات النشر عبر خطوط أنابيب التسليم. يسمح توحيد عمليات النشر والأدوات بإنشاء عملية نشر واحدة لنشر جميع المكونات والتطبيقات والخدمات عبر خطوط وبيئات تسليم متعددة ، إذا لزم الأمر وينطبق الشيء نفسه على بيئات التزويد والتنظيم. يسمح وجود منصة واحدة لتوفيرها وتنسيقها بتبسيط عمليات التزويد والتنسيق ويسمح بتوفير بيئات متعددة حسب الحاجة الآن ، ليس من العملي دائماً التوحيد على تقنية أو أداة واحدة فقط لأتمتة النشر أو لتوفير البيئة وتنسيقها. ومع ذلك ، يجب أن يكون الهدف هو تقليل عدد الأدوات. يعد امتلاك أداة معيارية واحدة أو أكثر أدوات معيارية لكل مجموعة تقنية هدفاً جيداً يجب السعي لتحقيقه

4. □□□□□□□□ □□□□□□□□ □□□□□□□□□□□□□□□□ . نظرًا لأن العديد من التطبيقات والخدمات أصبحت جاهزة للاختبار ، فقد لا تكون التطبيقات والخدمات الأخرى التي تعتمد عليها جاهزة ، حيث من المحتمل أن يكون لها أوقات دورات وخطط إصدار مختلفة. لا يمكن تنسيق التوفر دائماً. بالإضافة إلى ذلك ، قد لا تتوفر بيانات الاختبار وبيانات الاختبار عند الحاجة لكل تطبيق أو خدمة. تعالج خدمات المحاكاة الافتراضية والتطبيقات والبيانات ومصادر البيانات هذه المشكلة ، مما يسمح باختبار كل تطبيق وخدمة ، بشكل مستقل عن التطبيقات والخدمات الأخرى. أناقش هذه الحلول بالتفصيل في الفصل 4.

5. □□□□□□ □□□□□□ . يجب أن يكون إصدار نظام أعمال ، يتألف من تطبيقات وخدمات متعددة ويتم تسليمه عبر خطوط أنابيب توصيل متعددة ، جهداً منسقاً لتنفيذ خطط التسليم والإفراج عبر جميع خطوط أنابيب التسليم المعنية. تقليدياً ، اعتمدت المؤسسات على جداول بيانات ضخمة تحتوي على بيانات عن كل مكون يتم تطويره وتسليمه ، والتبعيات بينها ، وحالتها في خطوط التسليم الخاصة بكل منها. يتضمن ذلك تتبع وإدارة التطبيقات والخدمات أثناء تقدمها عبر بوابات الجودة المختلفة التي تحتاج إلى التحقق من صحتها قبل طرحها للإنتاج. إنهم بحاجة إلى تتبع وإدارة جميع نقاط التكامل واختبارها ، عبر جميع التطبيقات والخدمات التابعة. من الضروري الاستفادة من أداة قياسية تتعقب وتدير إصدار كل تطبيق وخدمة عبر جميع خطوط أنابيب التسليم ومداخل التكنولوجيا. وتتمثل الميزة الإضافية في إمكانية استخلاص هذه الأداة معلومات الحالة لكل خط أنابيب تسليم تلقائياً من المشروع وإدارة عنصر العمل وأيضاً استخراج معلومات الحالة لأدوات أتمتة النشر التي يتم استخدامها بواسطة كل خط أنابيب تسليم

200 مسرحيات لقيادة الابتكار devops الفصل 5

يعد تمكين القدرة على دعم تقنية المعلومات متعددة السرعات أمرًا ضروريًا لتعزيز الابتكار. عند تقديم ابتكار جديد ، فأنت بحاجة إلى السرعة. يجب أن تكون قادرًا على تقديم المنتجات عند الحاجة وأن تكون قادرًا على إجراء تجارب متعددة (كما سناقش لاحقًا). يجب أن تكون غير مثقل بتبعية خطوط أنابيب التوصيل الأخرى. لا يمكن تقييدك بسرعة أبطأ خط أنابيب تسليم

بناء الشيء الصحيح

الإفراط في تحديد النظام

ليس سرًا بين الرياضيين أنه من أجل تحسين الأداء عليك أن تعمل بجد. ومع ذلك ، فإن التدريب الشاق يكسرك ويجعلك أضعف. الراحة هي التي تجعلك أقوى. يحدث التحسن الفسيولوجي في الرياضة فقط خلال فترة الراحة بعد التدريب الشاق. هذا التكيف هو استجابة للتحميل الأقصى للقلب والأوعية الدموية والجهاز العضلي ويتم تحقيقه من خلال تحسين كفاءة القلب ، وزيادة الشعيرات الدموية في العضلات ، وزيادة مخازن الجليكوجين وأنظمة إنزيم الميتوكوندريا داخل خلايا العضلات. خلال فترات الاسترداد ، تتطور هذه الأنظمة إلى مستويات أعلى لتعويض الضغط الذي قمت بتطبيقه. والنتيجة هي أنك الآن في مستوى أعلى من الأداء

إذا لم يتم تضمين الراحة الكافية في برنامج التدريب ، فلا يمكن أن يحدث التجديد وهضبة الأداء إذا استمر عدم التوازن بين التدريب الزائد والراحة غير الكافية ، فسوف ينخفض الأداء. يمكن تعريف التدريب المفرط بشكل أفضل على أنه الحالة التي يتعرض فيها الرياضي للتوتر بشكل متكرر "من خلال التدريب إلى درجة أن الراحة لم تعد كافية للسماح بالتعافي." متلازمة الإفراط في التدريب هو الاسم الذي يطلق على مجموعة الأعراض العاطفية والسلوكية والجسدية بسبب الإفراط في التدريب الذي استمر لأسابيع إلى شهور. يعرفه الرياضيون والمدربون أيضًا باسم "الإرهاق" أو "الركود". هذا يختلف عن التباين اليومي في الأداء والتعب بعد التمرين الشائع في الرياضيين "المكفيين". يتميز التدريب المفرط بالإرهاق التراكمي الذي يستمر حتى بعد فترات التعافي

أكثر الأعراض شيوعًا هو التعب. قد يحدث هذا من التدريبات وقد يكون موجودًا في حالة الراحة قد يصبح الرياضي أيضًا متقلب المزاج ، وسهل الانزعاج ، وقد يغير أنماط النوم ، أو يصاب بالاكنتاب ، أو يفقد الرغبة التنافسية والحماس لهذه الرياضة. سيبطئ البعض عن انخفاض الشهية وفقدان الوزن. تشمل الأعراض الجسدية وجع العضلات المستمر ، وزيادة تواتر الأمراض الفيروسية ، وزيادة حدوث الإصابات

جينكينز ، 1998 —

في حين أن الإفراط في التدريب قد يتسبب في أن يصبح الرياضيون "متقلبين المزاج ، وسهل الانزعاج ، وقد غيروا أنماط النوم ، ويصابون بالاكنتاب" ، فإن الشيء الوحيد الذي يجعل أي متخصص في تكنولوجيا المعلومات ، وخاصة المطورين ، يتمتع بنفس التجربة العاطفية هو العمل بشغف في مشروع ، ووضع في ساعات العمل والجهد لإنجاز المشروع ، فقط لاكتشاف أنهم بنوا الشيء الخطأ سواء كان مشروعًا كاملاً أو مجرد وحدة نمطية من التعليمات البرمجية ، فهو إهدار كامل للموارد والمساهمات الفردية للوقت والطاقة وحل المشكلات لتقديم شيء ما ، فقط لعدم استخدامه مطلقًا لأنه كان يحل مشكلة خاطئة. لقد انهارت شركات بأكملها لأنها استثمرت مستقبلها ، أو في بعض الحالات

201 مسرحيات لقيادة الابتكار devops الفصل 5

من - dotcom حتى وجودها ، في منتج لم يكن السوق يريده أو يقبله. فكر في أي فشل في بدء تشغيل Pets.com إلى eToys.com إلى Webvan.com. الشركات للمعلومات الكبرية تمثل هذه المشاريع التي لم تر النور مطلقاً أو تم تعليقها بعد وقت قصير من إطلاقها بسبب قلة اهتمام العملاء وجذبهم. في معظم الحالات ، لم يتم اكتشاف عدم التطابق بين إمكانيات المنتج وتوقعات العملاء حتى يتم تطوير المنتج وتسليمه بالكامل ، فقط لئيم تجنبه من قبل المستخدمين

□□□□□□ □□□

في كتابه الأساسي (Eric Ries (Ries ، 2011) ، التي أطلقها *Lean* تقدم حركة . الذي يحمل نفس الاسم ، نهجاً بديلاً لتقديم المنتجات (تكنولوجيا المعلومات أو غير ذلك). في نهجه لا يتمثل الهدف في بناء منتج كامل ولكن التأكد من أنك تبني المنتج المناسب من خلال . رؤية المنتج مع المستخدمين. يتم إجراء هذا التحقق من خلال تطوير دورة تسمح لك بالاستمرار في التأكد من أنك تبني الشيء . بينما تقدم خطوات صغيرة من القدرة الجديدة (فكر في) والحصول بسرعة على تعليقات حول ما قمت بإنشائه للتو من المستخدمين الفعليين (فكر في) . كما ترى . التي قدمتها في الفصل 4 DevOps تتوافق هذه المبادئ تماماً مع مبادئ ، :الناشئة هي كما يلي Lean الركائز الأساسية الأربعة لشركة

1. لديك منهجية واضحة المعالم حول تقديم منتج الهدف ليس بذل جهود بطولية لإخراج المنتج من الباب ، بغض النظر عن أي شيء ، ولكن القيام DevOps بذلك بطريقة جيدة التنظيم ومنضبطة. مرة أخرى ، يتماشى هذا مع مسرحيات المعروضة في هذا الكتاب
2. السؤال الأكثر أهمية الذي يجب طرحه ليس "هل يمكن بناء هذا المنتج؟" ولكن "هل يجب بناء هذا المنتج؟" إن إشراك المستخدمين والعملاء المحتملين في وقت مبكر للتحقق من صحة فرضية المنتج ، حتى قبل كتابة سطر من التعليمات البرمجية ، والعمل معهم خلال دورة حياة التطوير هو الطريقة الوحيدة لضمان تسليم المنتج الصحيح
3. الحد الأدنى من المنتج القابل للتسليم (MVP). أو في بعض الحالات ، هو الحد الأدنى من القدرة التي يمكنك أن تأخذها إلى هو الطريق إلى MVP. المستخدم لمعرفة ما إذا كان يلبي احتياجاته ويقدم قيمة تجارية بأقل استثمار ممكن
4. في طريقة بدء التشغيل الخالي من الهدر . عندما تقضي الفرق وقتهم باستمرار في اكتشاف ما إذا كانوا يبنون الشيء لمستخدميهم ، فإن ذلك يضمن أنهم قادرون على تغيير الناشئة - بمجرد اكتشافهم أنهم انحرفوا Lean المسارات - ما تسميه حركة . عن احتياجات المستخدم

يتم تقديم المبادئ الثلاثة التي تشكل ركائز DevOps في الشكل 5-1. هنا، توجد هذه المبادئ لضمان أنك تبني DevOps بشكل صحيح.

1. DevOps هو نهج عمل يهدف إلى تحسين التعاون والتواصل بين فرق التطوير والتشغيل. يتم تحقيق ذلك من خلال تبني ممارسات DevOps التي تركز على الأتمتة والتكامل المستمر. DevOps هو نهج عمل يهدف إلى تحسين التعاون والتواصل بين فرق التطوير والتشغيل. يتم تحقيق ذلك من خلال تبني ممارسات DevOps التي تركز على الأتمتة والتكامل المستمر.
2. DevOps هو نهج عمل يهدف إلى تحسين التعاون والتواصل بين فرق التطوير والتشغيل. يتم تحقيق ذلك من خلال تبني ممارسات DevOps التي تركز على الأتمتة والتكامل المستمر. DevOps هو نهج عمل يهدف إلى تحسين التعاون والتواصل بين فرق التطوير والتشغيل. يتم تحقيق ذلك من خلال تبني ممارسات DevOps التي تركز على الأتمتة والتكامل المستمر.
3. DevOps هو نهج عمل يهدف إلى تحسين التعاون والتواصل بين فرق التطوير والتشغيل. يتم تحقيق ذلك من خلال تبني ممارسات DevOps التي تركز على الأتمتة والتكامل المستمر. DevOps هو نهج عمل يهدف إلى تحسين التعاون والتواصل بين فرق التطوير والتشغيل. يتم تحقيق ذلك من خلال تبني ممارسات DevOps التي تركز على الأتمتة والتكامل المستمر.

—IBM Design ، 2016

بدمج مبادئ بدء التشغيل الخالي من الهدر ومبادئ التفكير التصميمي في IBM قامت منظمات مثل عمليات إدارة عروض المنتجات الخاصة بهم. لقد قاموا بتفعيل عملية تطوير الشيء DevOps. النتيجة: يبنون الشيء DevOps. DevOps هو نهج عمل يهدف إلى تحسين التعاون والتواصل بين فرق التطوير والتشغيل. يتم تحقيق ذلك من خلال تبني ممارسات DevOps التي تركز على الأتمتة والتكامل المستمر.

تمكين التجريب

CoACHing تجربة في

يطور العديد من المدربين مهاراتهم وخبراتهم من خلال تجاربهم ومشاهدة المدربين الآخرين. ومع ذلك، فإن اكتساب الخبرات ببساطة لا يضمن كفاءة التدريب. إنه تكامل الخبرة والمعرفة بطريقة هادفة تعزز التعلم وتطور الخبرة بدورها. يحتاج المدربون إلى معرفة أفضل طريقة للتعلم من خلال... تجاربهم. الممارسة الانعكاسية هي أداة تعليمية رئيسية في هذا الصدد. بمجرد أن يتم تزويد المدربين ببعض الحلول، فإنهم يحتاجون بعد ذلك إلى استكشاف النتائج المحتملة لكل منها واختيار الاستجابة الأكثر ملاءمة. في بعض الحالات، قد تكون هذه التجربة افتراضية. قد يقدمون أفكارهم إلى مدربيهم الأقران للحصول على التغذية الراجعة. يمكن أن تكون التجربة الافتراضية طريقة عملية للتفكير بعد انتهاء الموسم ويستعد المدربون لموسم جديد. تحدث التجارب الواقعية في مجال الرياضة حيث يمكن للمدربين تنفيذ الحل المتصور ومراجعة تأثيره. من المهم ملاحظة أن التجريب في الممارسة العاكسة يختلف عن ممارسة التجربة والخطأ. تتضمن التجربة والخطأ ببساطة فعل شيء ما وعندما تفشل، فعل شيء آخر حتى ينجح شيء ما. النهج عشوائي ولا يمكن التنبؤ به مقارنة بالتجربة العاكسة. في التجريب الانعكاسي، تكمن الفكرة

في البناء على المعرفة الموجودة من خلال الاستفادة من الخبرات والتعلم لعمل اختيارات مستنيرة بناءً على المعلومات ذات الصلة. هذا النهج أكثر قابلية للتنبؤ ومدرّوس ويعزز بيئة تعليمية أكثر فعالية للرياضيين

فارس ، 2004-

يحتاج المدربون إلى الاستعداد للمباراة التالية من خلال تجربة مسرحيات جديدة لمواجهة تحديات اللعب ضد فريق معين أو لمعالجة أوجه القصور في المسرحيات التي لوحظت في اللعبة أو الممارسة الأخيرة. وبالمثل ، يجب أن تكون المؤسسة التي تفقد الابتكار قادرة على تجربة الأفكار الجديدة والميزات الجديدة وتجارب المستخدم الجديدة وواجهات المستخدم الجديدة ونماذج الأعمال الجديدة والتقنيات الجديدة. كما ساقدم في المسرحية الأولى التي ناقشتها في هذا الفصل ، هناك مفهومان في الناشئة Lean قيادة الابتكار ، اشتهرت بهما حركة

■ الحد الأدنى من المنتج القابل للتطبيق

■ تفشل بسرعة

الهدف من التجربة هو منع تطوير منتجات تعمل بكامل طاقتها لمعرفة ما إذا كانت هي المنتج والفشل سريعاً معها حتى تتمكن في النهاية من الوصول إلى MVP المناسب وبدلاً من ذلك تجربة المنتج المناسب الذي ينجح

هذا النهج لتقديم الحد الأدنى من المنتجات القابلة للتطبيق □□□□□□ بسرعة لإجراء التجارب السريعة. سرعة الابتكار مدفوعة بالسرعة التي يمكنك بها إجراء تجارب متعددة والفشل السريع مع التجارب التي يفترض أن تفشل من أجل اكتشاف التجارب الناجحة

A / B □□□□□□

هو أسلوب شائع يستخدم لتمكين هذه التجربة السريعة عن طريق إجراء A / B □□□□□□ هو تقديم نسختين من صفحة A / B اختبارات لمتغيرات متعددة لميزة بشكل متواز. الهدف من اختبار الويب أو الميزة أو التطبيق بشكل عشوائي للمستخدمين لمعرفة أيهما يعمل بشكل أفضل أو أكثر إنتاجية تتطلب العملية أدوات مضمنة في التطبيق للتأكد من أنه يمكنك قياس التفاصيل المختلفة لكيفية تفاعل المستخدمين مع الإصدارين ومقارنتهم. يسمح هذا الاختبار للمطورين وفرق العمل باتخاذ قرارات بشأن إصدارات الميزات التي يجب الاحتفاظ بها ومتابعة تطويرها وأي إصدارات يتم حذفها للحصول على أقصى قدر من النتائج A / B فيما يلي بعض الممارسات الأساسية لتنفيذ اختبار: وتنفيذ فشل سريع للميزات التي يتم اختبار إصداراتها

□□□□□□ □□□□□□ □□□□□□ . ما المقاييس والمقاييس التي ستحدد الإصدار

الذي نجح؟ العناصر المباعة ، والتسجيلات التي تم إجراؤها ، وعدد النقرات ، والوقت المنقضي ، والعناصر التي تم تصفحها ، والخروج بعيداً عن الصفحة (معدل الارتداد) ، وعربات التسوق المتروكة - يجب تحديد مقاييس النجاح أو الفشل مسبقاً ، ووضع الأجهزة لقياسها

يجب تقديم كلا الإصدارين ، A / B □□□□□□□□□□□□□□□□ . لكي يعمل اختبار

إلى عدد متساوٍ من المستخدمين الفريدين لنفس الفترة الزمنية

205 مسرحيات لقيادة الابتكار devops الفصل 5

■ . يجب أن يظهر الإصدار نفسه من
الميزة عبر التطبيق ، ولا يقتصر على مناطق قليلة فقط ، حتى يعمل النهج ويؤدي إلى نتائج
متسقة.

■ . يجب جمع مجموعات البيانات ذات الدلالة
الإحصائية لضمان التحليل المناسب لنتائج الاختبار من كلا الإصدارين. لذلك يجب تشغيل
الاختبارات لفترة طويلة بما يكفي للحصول على عدد كافٍ من المستخدمين لاستخدام الميزة
. يتيح لك هذا أيضًا التقاط الاتجاهات والقضاء على تأثير القيم المتطرفة

■ . الفكرة هي إنتاج تغييرات صغيرة باستمرار (دفعات A/B)
بسرعة لتحديد مجموعات التغييرات التي تعمل. هذا يسمح A/B صغيرة) وإجراء اختبار
للفشل السريع مع الحد الأدنى من الهدر

فإنك تحتاج إلى بعض القدرات الأساسية في نظام التسليم ، A/B لتتمكن من إجراء اختبارات
:وبينات الإنتاج

■ القدرة على نشر نسختين من التطبيقات على مجموعتين من الخوادم للسماح بنصف
المستخدمين القادمين ليتم توجيههم إلى كل إصدار من الإصدارين. هذا يتطلب عمليات النشر
لاستهداف مجموعات من الخوادم. لن يعمل نموذج نشر الكل أو لا شيء ، والذي ينشر تطبيقًا
على جميع الخوادم

■ إمكانية التوجيه لتوجيه المستخدمين بشكل عشوائي إلى كل مجموعة من الخوادم مع عدد متساوٍ
من المستخدمين الذين ينتقلون إلى كل مجموعة

■ القدرة على تمرير عمليات النشر للأمام لاستبدال الإصدار الذي فشل بالإصدار الذي نجح

ولذلك فإن التجريب السريع يحتاج إلى منصة تقنية - خط أنابيب لتسليم التطبيق - مصمم لتمكين
التجريب. يجب أن تكون قادرة على تقديم إمكانيات وميزات جديدة بسرعة لمجموعات مستهدفة من
الخوادم ، لتكون قادرة على مراقبة التطبيق وسلوك المستخدم ، ولتتمكن من التقاط بيانات الاختبار
الصحيحة وتحليلها حتى تفشل بسرعة

تسليم أنظمة مضادة للكسر

قم بشراء رياضي مضيف

لقد عدت للتو من عطلة نهاية الأسبوع في علوم الرياضة في سياتل ساوندرز 2014 حول بناء
رياضي مضاد للكسر. . . قدمت [الجلسة] الأولى ، التي أعدها ديف تيني ، مدير الأداء في
نظرة عامة عامة عن مصدر مفهوم بناء رياضي مضاد للكسر (خسر الكثير من ، Sounders
الألعاب بسبب الإصابة) ، وما الدروس التي يمكن أن نتعلمها من أكثر الألعاب المضادة للضعف
الصناعات (شركات الطيران والطاقة النووية) ، وإذا تمكنا في الواقع من أخذ رياضي هش وتغييره
إلى رياضي مقاوم للإصابة (يبدو كذلك)

لكونها في طليعة الحصول على بيانات الرياضة Sounders في هذا المجال ، تم التعرف على
والأداء وتحليلها بشأن التعافي ، واستعداد الجهاز العصبي ، والعلاقات بين الإصابات والامتثال

لمبادرات معينة ، والنوم ، والمسافات التي يقطعها الرياضيون ، وأنواع تحميل تجربة الرياضيين .قفز في وجهي مقياسين

أحدهما أنهم قادرون على قياس ما يسمى "حمل السرعة" مقابل "حمل الجسم". هذا هو المكان الذي تذهلني فيه التكنولوجيا المتاحة مرة أخرى. لا يقتصر الأمر على استخدامهم لنظام تحديد لقياس المسافة التي يقطعها كل رياضي أثناء الألعاب والممارسات ، ولكنهم (GPS) المواقع العالمي ، أيضاً يقيسون □□□□□ تلك المسافات. يمثل "حمل السرعة" حركة خطية (الركض لأسفل على سبيل المثال) ، وهو مسيطر على البساطة ، وهو أكثر شيوعاً خلال مواقف اللعبة العادية. تمثل أحمال الجسم تغيرات في الاتجاه. تميل إلى تحميل السلسلة الخلفية وتحدث بتردد أكبر أثناء الألعاب الصغيرة. بمجرد حصولهم على هذه البيانات ، يمكنهم تحليلها لاكتساب نظرة ثاقبة للضغوط العضلية الهيكلية على الرياضيين الفرديين

مقياس آخر مثير للاهتمام كان بيانات النوم. يستخدمون الشاشات لتتبع الوقت الذي يذهب فيه الرياضيون إلى الفراش كل ليلة ، وعندما يستيقظون ، ومقدار القذف والاستدارة التي يقومون بها وأشار ديف إلى أن اللاعبين الأكثر أداءً في ناديتهم هم الذين ينامون بشكل أفضل ، ليس فقط من حيث عدد الساعات والانتظام ولكن أيضاً من حيث الجودة. متوسط مرات ذهاب اثنين من أفضل الرياضيين إلى الفراش؟ 9:44 و 10:15 مساءً

كافن ، 2014 —

التحدي الأكبر للرياضي هو الإصابة. ومع ذلك ، لا يمكن للرياضيين تجنب الإصابة. التعرض لمواقف يمكن أن تسبب إصابة هو جزء مما يفعله الرياضي. وأنا لا أعني فقط رياضة تتطلب الاحتكاك مثل كرة القدم الأمريكية أو الرياضات عالية الخطورة مثل القفز في الجمباز ، وهي بطبيعتها رياضات ذات تأثير عالي السرعة ، ولكنها أيضاً رياضات تبدو "آمنة" مثل الجولف أو التنس حيث يمكن للإصابة تأتي من الرياضيين الذين يبالبون في تمديد عضلاتهم أو إجهادها. يمكن أن يتأثر الرياضيون بالإصابة في أي مكان ، من فقدان بضع دقائق من المباراة إلى إصابة تنتهي مسيرتهم ، وحتى الإعاقة الدائمة أو الوفاة. تم إجراء قدر هائل من الأبحاث لمساعدة الرياضيين على التدريب والتكيف لجعلهم يزدهرون في هذه البيئات عالية الضغط ، لجعلهم مرنين ، ولجعلهم يتعافون بشكل أسرع ويكونوا أقوى ، ولجعل أجسامهم ضعيفة. ينطبق هذا أيضاً على أنظمة تكنولوجيا المعلومات ، لا سيما أنظمة تكنولوجيا المعلومات المصممة للتغيير المستمر ، للتجربة المستمرة ، والتي تحتاج إلى أن تكون قادرة على الازدهار في ظل الفوضى ، في ظروف مرهقة حيث تتعطل الخوادم ، ويتم توصيل الخوادم الجديدة بإلنترنترنت طوال الوقت - لتكون غير قابلة للكسر

مصطلح □□□□□□ نسيم نيكولاس طالب ، وهو تاجر خيارات كتب سلسلة من الكتب عن العشوائية والاحتمالات وتأثيرها على الأسواق وعلى الحياة. قدم المصطلح لأول مرة في كتابه □□□□□□□□□□ (طالب ، 2007) ، حيث ناقش الأحداث النادرة ، التي أكد أنها ليست (طالب) Antifragile عشوائية ونادرة كما يعتقد الناس - مثل انهيار سوق الأسهم. ثم كتب كتاباً بعنوان حيث توسع في مفهوم □□□□□□□□□□ ليصف الأشياء التي ليست هشة ولا □ (2012) ، قوية ، ولكنها في الواقع تستفيد من الفوضى

207 مسرحيات لقيادة الابتكار devops الفصل 5

إذا كنت تقرأ هذا الكتاب بتنسيق مطبوع ، فهو قطعة أثرية قوية. يمكنك إسقاطه ولن ينكسر. من ناحية أخرى ، إذا كنت تقرأ هذا على قارئ إلكتروني أو جهاز لوحي ، فإن الجهاز هش بطبيعته. من المحتمل جداً أن يؤدي إسقاطه على سطح صلب إلى إتلافه. هذان مثالان على أنظمة مألوفة لديك يتحداً طالب للتفكير في الأنظمة المضادة للكسر - تلك التي ليست هشة ولا قوية ولكنها من النوع الذي يجعلها أقوى عندما توضع في المواقف العصيبة. جميع الأنظمة الحية ، والأكثر عضوية حقاً ، قابلة ، للكسر بطبيعتها. مثال يعطيه طالب هو عظم. عندما يتم كسرها ، إذا سمح لها بالشفاء بشكل صحيح ، فإن البقعة الملتئمة الناتجة على العظم تكون في الواقع أكثر كثافة وبالتالي أقوى من العظم غير التالف. تعمل اللقاحات بطبيعتها على زيادة قابلية الجهاز المناعي للحيوان لهشاشة. ينتج الإنسان أو الحيوان الذي يتم حقنه بكمية صغيرة من الجرثومة أجساماً مضادة يمكنها حمايته من العدوى الفعلية للجرثومة الدماغ البشري هو المثال النهائي لنظام مضاد للكسر. عندما تقرأ هذا الكتاب - في المطبوعات أو على جهازك اللوحي - إذا وجدت أجزاء من الكتاب مثيرة للاهتمام بما يكفي للالتزام بالذاكرة ، فإن حالة الإجهاد المتمثلة في محاولة نسيانها تجعل الذاكرة المطبوعة في خلاياك العصبية أقوى. لا يمكنك أبداً أن تحاول بوعي أن تنسى شيئاً ؛ إنه يجعلك تتذكر أكثر فقط - أسأل أي مراهق محطم القلب. لنجرب تجربة: إذا طلبت منك □□□ تفكر في "قرد راقص يأكل موزة" أثناء قراءتك لبقية هذا الكتاب ، فقد خلقت لك حالة توتر من خلال المطالبات □□□□ التفكير في شيء ، وهو استحال. بينما تستمر في قراءة هذا الكتاب ، ستعرض لك خلايا دماغك المضادة للكسر بشكل عشوائي صورة لقرد يقوم □□□□□□□□□□ أثناء التهام موزة. ليس لديك طريقة لمنع حدوث ذلك

يمكنك أن تشكرني لاحقاً ... للتلخيص

□□□□□ الأشياء تنكسر أو تتلف تحت الضغط

□□□□□□□ الأشياء لا تتأثر بالتوتر ؛ إنهم لا ينمون أقوى ولا أضعف

تصبح الأنظمة □□□□□□□□□□ أقوى في ظل ظروف الإجهاد

نظم تكنولوجيا المعلومات والقابلية للهشاشة

لطالما كانت فرق العمليات تسعى دائماً إلى جعل أنظمتها قوية. يريدون أنظمة لا تنهار. إنهم يريدون أنظمة لا تتأثر بأي مواقف ضغط قد يواجهونها. إنهم يريدون أنظمة يمكن التنبؤ بها في سلوكهم حتى يتمكنوا من تخفيف كل حالة إجهاد إذا حدث ذلك ، مما قد يتسبب في تعطل الأنظمة أو حتى تدهور الأداء من أجل إنشاء نظام قوي حقاً ، يجب أن تكون قادراً على التنبؤ بالبناء في النظام والتخفيف من كل مصدر من مصادر الفشل في النظام. في عالم اليوم المجهول المتعدد ، هذا مستحيل حرفياً. الأنظمة اليوم ليست ثابتة. إنها ديناميكية مع الخوادم التي يتم توفيرها وإلغاء توفيرها في الوقت الفعلي. إنهم يستفيدون من خدمات متعددة من مزودين متعددين ، قد يكون العديد منها مملوكاً لفرق الموردين الآخرين ، بما في ذلك موردي الطرف الثالث خارج مؤسستهم. العدد الهائل لنقاط الفشل المحتملة يجعل من المستحيل التنبؤ بالفشل ووضع خطط التخفيف لها. أضيف إلى ذلك الضغوط الإضافية للتغيير الناتج عن التسليم المستمر لإصدارات جديدة من التطبيقات والخدمات التي تقدمها فرق التطوير التي تعمل DevOps باستمرار على ابتكارات جديدة ، وتزداد نقاط الفشل بشكل كبير. لتصميم الأنظمة حقاً لتمكين

ChAos MoNkeY AND NetFLix sIMIAN ArMY

Imagine getting a flat tire. Even if you have a spare tire in your trunk, do you know if it is inflated? Do you have the tools to change it? And, most importantly, do you remember how to do it right? One way to make sure you can deal with a flat tire on the freeway, in the rain, in the middle of the night is to poke a hole in your tire once a week in your driveway on a Sunday afternoon and go through the drill of replacing it. This is expensive and time-consuming in the real world but can be (almost) free and automated in the cloud.

This was our philosophy when we built Chaos Monkey (Figure 5-3), a tool that randomly disables our production instances to make sure we can survive this common type of failure without any customer impact. The



Figure 5-3: Netflix simian Army (image source: github.com/Netflix)

continued

بدأنا في إنشاء حيوانات شبيهة جديدة تسبب أنواعاً ، Chaos Monkey مستوحاة من نجاح مختلفة من الإخفاقات ، أو تكتشف الظروف غير الطبيعية ، وتختبر قدرتنا على البقاء على قيد الحياة . افتراضى للحفاظ على سحابتنا آمنة وأمنة ومتاحة للغاية Simian ؛ جيش

يجب أن تكون الأنظمة المضادة للكسر قادرة على معالجة الأعطال باستمرار. لسوء الحظ لديها خطط وبروتوكولات للتعامل مع الحوادث IT Ops على الرغم من أن جميع مؤسسات ، إلا أنه نادرًا ما يتم اختبارها. يتعين على الفريق الرياضي أن يتدرب باستمرار ، خلال فترة ، ما قبل الموسم والموسم ، لإتقان المسرحيات التي يريد تشغيلها. قد تكون هذه مجموعة قياسية من المسرحيات التي يتم تشغيلها طوال الوقت ، أو مسرحية مغيرة للعبة يريدون تشغيلها كتكتيك مفاجئ للفوز بمباراة حاسمة. فاز فريق كرة القدم في نيو أورليانز ساينتس بسوبر بول في عام 2010 من خلال تنفيذ ركلة "جانبية" مفاجئة ضد فريق إنديانابوليس كولتس XLIV ومع ذلك ، فقد نجحوا في المسرحية ليس لأنهم ضبطوا الفريق المنافس على حين غرة في كمين (وهو ما فعلوه) ولكن لأنهم مارسوه عدة مرات وأضفوه فقط إلى قواعد اللعب الخاصة . (Triplett ، 2014) بهم بعد أن نجح بشكل مثالي في الممارسة

من الرغبة في جعلها قوية للغاية. يتخذ مسؤولو النظام الذين يحتفظون بخوادم فردية للحفاظ عليها دائماً خطوات لتوفير كل الرعاية والتغذية التي تحتاجها الخوادم للتعامل مع أي مشكلة

رياضياً مثيراً للإعجاب سيمثلون أحصنة طروادة في ريو. يدور ستانفورد في المراكز الثلاثة الأولى مع 39 رياضياً

مارتينيز ، 2016 —

النقاط الرئيسية في هذا المقال هي كما يلي: ■ 1018 رياضياً كانوا

طلاباً حاليين وسابقين

كانوا يمثلون 107 دولة (ما مجموعه 206 دولة تنافست في أولمبياد ريو 2016) ■
كانوا من 223 كلية وجامعة أمريكية ■

وهكذا طورت الكليات والجامعات في الولايات المتحدة منصة للألعاب الرياضية وألعاب القوى على مستوى عالمي حقاً. إنها منصة واسعة بما يكفي لدعم كل رياضة أولمبية تقريباً (وأكثر من ذلك - فكر في كرة القدم الأمريكية ، وهي ليست رياضة أولمبية ، ورياضات الألعاب الأولمبية الشتوية ، والتي لم يتم تضمينها في هذا العدد). وهي أيضاً منصة استقطبت ، من خلال المنح الدراسية ومنشآت التدريب إنها منصة رشيقة. (Farrell ، 2008) ذات المستوى العالمي ، أفضل الرياضيين الناشئين في العالم ومرونة وقابلة للتطوير وموثوقة حقاً

الرشيق والمرن والقابل للتطوير DevOps ما الذي قد يستلزمه تقديم مثل هذا النظام الأساسي والموثوق؟ في الفصل 4 ، أقدم طرقاً متعددة لبناء خط أنابيب لتسليم التطبيق. وتشمل هذه كلاً من وجهة نظر حيادية للتكنولوجيا والنهج الخاصة بالتكنولوجيا. دعنا ندمج هذه الأفكار لبناء خط أنابيب التي قدمتها سابقاً في هذا الفصل. نظراً لأن Antifragile توصيل متكامل مع الحاجة إلى أنظمة للابتكار ، فإن الأولوية هي الحاجة إلى نظام DevOps التركيز في هذا الفصل ينصب على ألعاب يدعم السمات التي قدمتها سابقاً (والتي يتم تكرارها هنا لسهولة الاستهلاك) DevOps أساسي

تحقيق تقنية المعلومات متعددة السرعات ■

بناء الشيء □ □ □ □ □ □ □ ■

تمكين التجريب ■

توصيل أنظمة مضادة للكسر ■

بينما يمكن تقديم مثل هذه المنصة على أي بنية أساسية أساسية ، فإن الاستفادة من التكنولوجيا السحابية لتقديمها أمر ضروري. يمكن للنظام الأساسي السحابي فقط توفير المرونة والمرونة والحجم والسرعة اللازمة لتلبية متطلبات هذه السمات
يحتوي الفصل 4 على قائمة الإمكانات التالية التي تدخل في مجموعة أدوات خط أنابيب التسليم

إدارة شفرة المصدر ■

بناء ■

التكامل المستمر ■

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☒ **للبينات والتطبيقات لتحديد المشكلات في الوقت الفعلي**

والتخفيف من حدتها إن أمكن

يجب أن يكون التركيز الأساسي على تنفيذ عمليات النشر باللونين الأزرق ، DevOps من منظور والأخضر لضمان عدم حدوث أي توقف بسبب عمليات نشر التطبيقات في الإنتاج

ارتفاع معدل التوفر مع التخلص من الزرق

باستمرار - كثيرًا يوميًا. للتأكد من أن الانتقال **Bluemix Garage Method** يتم تسليم موقع ويب إلى الإصدار الذي تمت ترفيقته من موقع الويب لا يتطلب أي وقت تعطل ، قام الفريق بتنفيذ عملية النشر بالونين الأزرق والأخضر. عندما يتم دفع وظيفة جديدة إلى الإنتاج ، يتم نشرها إلى مثل **URL** ليس مثل التشغيل الفعلي. بعد التحقق من صحة طبعة التطبيق الجديد ، يتم تعيين عنوان العام للمثل الجديد للتطبيق. يتضمن النشر الأزرق والأخضر الخطوات التالية

■ إذا كان التطبيق الأزرق موجودًا ، فاحذفه يدويًا قبل إعادة التشغيل

■ دفع إصدار جديد من التطبيق الأزرق

■ تعيين متغيرات البيئة للتطبيق الأزرق

■ إنشاء وربط الخدمات للتطبيق الأزرق

■ **ابدأ التطبيق الأزرق**

■ اختبار التطبيق الأزرق

■ **تعيين حركة المرور إلى الإصدار الجديد من التطبيق الأزرق من خلال ربطها بالمضيف العام**

■ **احذف المسار المؤقت للتطبيق الأزرق الذي تم استخدامه للاختبار**

■ أعد تسمية التطبيق الأخضر إلى "النسخة الاحتياطية للتطبيق الأخضر". يستمر تشغيل تطبيق

النسخ الاحتياطي بحيث لا يتم إنهاء الجلسات النشطة

■ **"إعادة تسمية التطبيق الأزرق إلى التطبيق الأخضر"**

Cloud يكمل الفريق خطوات النشر باللونين الأزرق والأخضر باستخدام واجهة سطر أوامر DevOps Services. المضمنة في وقت تشغيل Foundry.

—جو لووینجروپر ، 2016—

تجريد البيئة

المناسبة ، خاصة تلك التي تكون مضادة للكسر ، في DevOps يمثل الهدف الأول والأهم لبناء منصة تجريد البيئة والبنية التحتية من الممارسين الذين يطورون التطبيقات والخدمات ويختبرونها. بالنسبة لهم ، يجب أن تظهر البيئة كمجموعة مجردة من خدمات البنية التحتية أو النظام الأساسي التي يمكنهم الاستفادة منها والاستفادة منها لتطويرها وتقديمها. هذا يسمح لهم بتحقيق تسليم تطبيقات متعددة من

خلال خطوط توصيل متعددة ، تعمل بسرعات متعددة ؛ لتحقيق التوجيه السريع □ باستخدام تقنيات الأفكار العمل والميزات الجديدة ؛ أن تفشل بسرعة وتفشل في كثير من الأحيان ؛ A / B مثل اختبار وعدم الاهتمام باستقرار البيئة بسبب التجارب التي فشلت أو التطبيقات والخدمات المقدمة التي لا تعمل أو.أ.تعمل بالشكل المطلوب

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

تتمثل طريقة تقديم طبقة من التجريد فوق البنية التحتية في الاستفادة من البيانات المعروفة بالبرمجيات داخلياً أو Ops بتقديم البنية التحتية كخدمة - سواء كان فريق Ops لفريق SDE يسمح (SDEs) بإنعاش حسابياً - لكشف البنية التحتية كمجموعة من الخدمات التي يمكن لأي ممارس في دورة حياة تسليم التطبيق الوصول إليها واستخدامها عبر مجموعة محددة جيداً من واجهات برمجة التطبيقات. يمكن أو ، Hypervisor من خلال وجود طبقة افتراضية فقط ، أو تشغيلها على برنامج SDEs تسليم سحابية تعمل بكامل طاقتها

مستويات مختلفة من التجريد من خلال السماح بتعريف المكونات المختلفة للبيانات SDEs توفر
وتشمل هذه ما يلي (Li ، 2014) وإدارتها كبرنامج

- (SDS) التخزين المعرف بالبرمجيات
- (SDNs) الشبكات المعرفة بالبرمجيات
- (SDC) الحوسبة المعرفة بالبرمجيات
- (SDM) إدارة معرفة بالبرمجيات
- التنسيب. وأتمتة عبء العمل

الآن حلولاً برمجية لإدارة مراكز البيانات بالكامل ، وليس IBM و VMware تقدم شركات مثل (SDDCs) فقط البيانات ، في إشارة إلى حلولها كمراكز بيانات محددة بالبرمجيات البيانات المعرفة بالبرمجيات ، أثناء تقديم طبقة من التجريد والوصول إلى البيانات عبر واجهات برمجية التطبيقات للمستهلكين في البيئة ، توفر أيضاً القدرة على أتمتة إدارة البيانات عبر البرامج :لموفري البيئة. يمكن لموفري البيئة القيام بما يلي

- **بيانات الإصدار عن طريق تعيين إصدارات البرنامج الذي يحدد البيانات**
- **تخزين إصدارات البيئة في مستودع ، مما يسمح بإمكانية الوصول بسهولة إلى الإصدارات السابقة من البيانات في حالة الحاجة إلى إعادة إنشائها لتقييم الخلل**
- **على بيئة ما - سواء كانت Ops تغيير إدارة البيانات. يصبح هذا أسهل لأن كل تغيير تقوم به تقوم بتطبيق تصحيح أو إجراء تغيير في التكوين - يصبح مدفوعاً بالبرمجيات ، عن طريق إنشاء إصدار جديد من البيئة. يتم تطبيق التغييرات عبر البرامج النصية (لا يتم تسجيل الدخول إلى وحدات تحكم المسؤول - لا يتم توسيع نطاقها ولا يمكن إدارتها بسهولة). يتم إصدار هذه البرامج النصية مع التعليمات البرمجية والعناصر الأخرى الخاصة بمكدس التطبيق الذي يتم تسليمه**
- **تنفيذ ممارسات هندسة البرمجيات التقليدية مثل إدارة تكوين البرامج ، والتي يمكن تطبيقها علم الكود الذي يمثل البيانات**

وسأركز بدلاً من ذلك على تأثيرها على الطريقة التي ، ESPN سافر للقراء ولادة وتاريخ نشاهد بها الرياضة ونستهلكها ونلتهمها على مدار الساعة طوال أيام الأسبوع. يوجد اليوم المنصات عبر كل وسيط يمكن تخيله. بالنسبة للمبتدئين ، هناك قنوات ESPN وربما الآلاف - من منصات - ثم هناك مواقع ESPN Classic و ESPNNews و ESPN2 و ESPN تلفزيونية متعددة مثل وبالطبع شبكة espnradio.com ، و espn360.com ، espn.com ، الويب الخاصة بهم الراديو التي تحظى بشعبية كبيرة. يقدم لنا كل منها تغطية لا تنتهي للنتائج والمعاملات والشائعات والآراء والبصيرة والتحليل. إذا لم تكن هذه الخيارات كافية ، ف لديهم تطبيقات متعددة مثل تطبيقات على سبيل المثال لا الحصر. مع ظهور ، ESPN Radio و Sports Center و Score Center المفضلة لديك هناك حتى لا تضطر أبداً إلى ESPN يمكنك حتى متابعة شخصية ، Twitter الاستغناء عن الرياضة إذا كنت لا ترغب في ذلك. الحقيقة هي أن معظمنا لا يريد الاستغناء عن والشبكات الرياضية الأخرى - تزودنا باختيارات ESPN. الرياضة. إنها مسألة عرض وطلب بسيطة لا حصر لها في بوفيه الرياضة لأننا نطلبها. أنا لا أقول أن هذا أمر سيء - بصفتي من عشاق الرياضة ، فانا أشعر بالرهبة من مدى تزواج الرياضة والتكنولوجيا منذ الأيام التي كنت أعدل فيها الهوائي في تلفازي الملون أحياناً ، قبل أن يكون لدينا كابل وعندما كان الاثراء فقط هم من يستطيعون شراء القنوات الفضائية

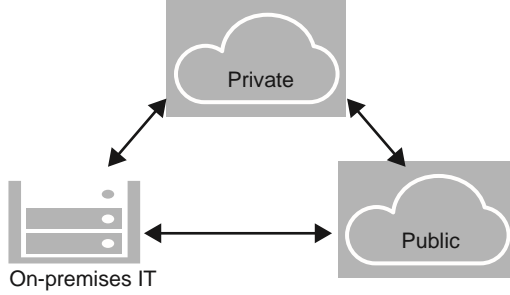
—أنطونيو لوبيز (لوبيز ، 2014) —

نظراً لمدى حداثة السحابة ، لا يمكنك حتى أن تقول ، " هذه ليست سحابة والدك " أو (مثل مؤلف هذا المقال) تحدث عن كيف "لن يختبر الأطفال اليوم أبداً" الشعور بتوفير مثيل خادم على سحابة الأمس. إن معدل التغيير في التكنولوجيا السحابية يحرك بسرعة فائقة والتسارع بحيث أصبح من المستحيل تقريباً على العديد من المؤسسات مواكبة ذلك. أدت الطبيعة المتطورة باستمرار للتكنولوجيا السحابية إلى حدوث تطور كبير في نماذج استهلاك السحابة. لم يعد اختيار الطريقة التي تريدها لاستخدام السحابة أمراً بسيطاً مثل الاختيار بين "خاص" و "عام" ، كما كان الحال في الأيام غير القديمة. هناك عدد لا يحصى من الخيارات للاختيار من بينها اليوم ، والقرار لا يتعلق فقط بالتكلفة أو الموقع ، بل بالعديد من العوامل الأخرى. إن اختيار النموذج الصحيح ، أو في معظم الحالات ، جميع النماذج لإنشاء ما يُشار إليه □□□□□□□□ □□□□□□□□ ، سيحدد مكونات البنية التي ستواجه عليها. وستحتاج إلى دعمها DevOps منصة

[illegible]

219 مسرحيات لقيادة الابتكار devops الفصل 5

لهم بالتحكم في نواذ الانقطاع ، إن وجدت. وبالتالي فهي عبارة عن سحابة خاصة يديرها البائع ، ومع ذلك ، فهي محلية. لذلك يمكنك التفكير في المحلية على أنها خدمة □□□□ □□□□□□ . □□□□□□ □□□□□□ .



الشكل 5-5: نماذج استهلاك السحابة

اختيار نموذج الاستهلاك الصحيح ليس بالأمر الهين. يعد استهلاك السحابة العامة بحكم التعريف □□□ ثابت - فهو (نسبيًا) سهل التشغيل / سهل التشغيل من السحابة العامة. يتم تخصيص نفس الطريقة أيضًا ، لأنك تستفيد بشكل أساسي من السحابة العامة التي تم تخصيصها الآن لمؤسستك. إنه مشابه لاستئجار مبنى قائم بذاته ، على عكس استئجار شقة في مبنى متعدد المستأجرين. المحلي ، من ناحية أخرى ، مختلف. يعد فهم تقسيم المهام بين مؤسستك والمورد أمرًا بالغ الأهمية. فهم عمليات إدارة التغيير أمر ضروري. أنت تقوم بتكوين شراكة مع البائع تكون أكثر تعقيدًا ، وبالتالي تكون أكثر قيمة من استخدام السحابة العامة.

الخاصة بك ، فإن DevOps مدار ذاتيًا مقابل مدار من البائع أثناء اختيار النموذج الأفضل لمنصة أول نقطة قرار يجب اتخاذها هي الأكثر أهمية: هل تبحث عن بيئة سحابية يديرها البائع ، أم ستدير ذاتيًا؟ إذا كنت تفضل الخيار الأخير ، فإن النموذج الصحيح هو سحابة خاصة محلية تقليدية. تشتري المؤسسة تقنية النظام الأساسي السحابي من البائع ، وتدريب موظفيها ، وتجهز بنيتها التحتية في مركز البيانات ، وتبني النظام الأساسي الذي يستفيد من السحابة المقدمة.

إذا كنت تفضل النموذج الذي يديره البائع ، فستحتاج المؤسسة بعد ذلك إلى الاختيار بين منصة سحابية محلية وخارجية. يجب اتخاذ هذا القرار بشأن السحابة المُدارة مقابل السحابة المُدارة ذاتيًا أولاً إنه لا يؤثر فقط على السحابة المختارة ولكن أيضًا على موظفي تكنولوجيا المعلومات. إذا واجهت مؤسسة ما تحديًا بسبب قدرتها على إدارة سحابة مع موظفي تكنولوجيا المعلومات لديها ، بسبب المهارات أو عدد الموظفين ، أو إذا كانت غير متأكدة من قدرتها ورغبتها في أن تكون قادرة على تقديم الخدمات واتفاقيات مستوى الخدمة المرتبطة التي يتوقعها عملاؤها من السحابة ، إذاً يجب أن ننظر في السحابة المُدارة. لم يكن هذا القرار متاحًا حتى الآن لأن جميع خيارات السحابة المُدارة كانت متاحة فقط لم يعد هذا ، PaaS و IaaS على السحابة الخارجية. مع ظهور عروض السحابة المحلية ، لكل من DevOps هو الحال. يمكن للمؤسسة الحصول على سحابة محلية ، كخدمة مُدارة ، وإنشاء منصة عليها.

220 مسرحيات لقيادة الابتكار devops الفصل 5

بمجرد اختيار نموذج استهلاك السحابة ، فإن مجموعة القرار التالية هي تحديد ما إذا كان نموذج اعتماد السحابة الصحيح هو البنية التحتية كخدمة ، أو النظام الأساسي كخدمة ، أو الحاويات. عندما كيف تريد المنظمة تقديم DevOps. تتخذ هذا القرار ، فإليك تبدأ في اختيار الخدمات التي تشكل منصة IaaS التي تعمل أعلى سحابة DevOps هذه الخدمات؟ هل ستكون خدمات تم إنشاؤها باستخدام أدوات أم أنه من الأفضل أن تكون جزءًا من النظام الأساسي الفعلي الذي يتم تقديمه؟ دعنا نفحص هذه الخيارات بعد ذلك.

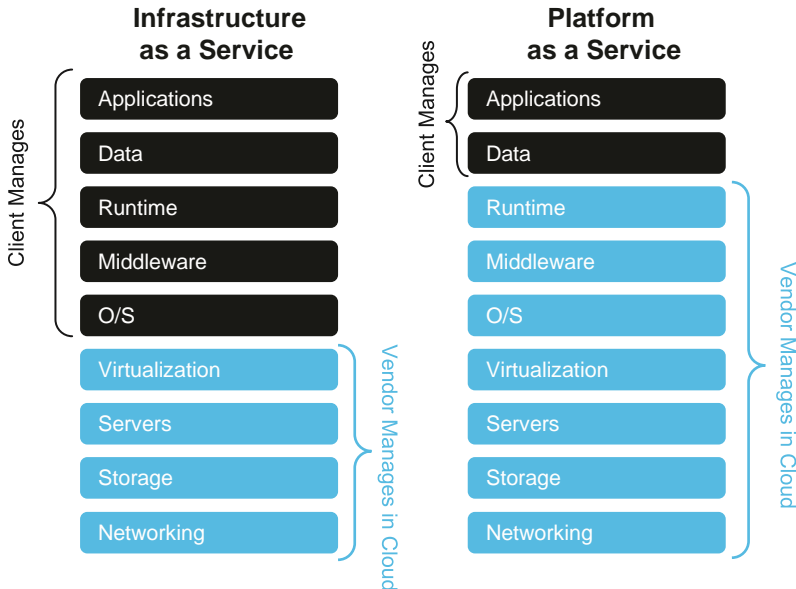
البنية التحتية كخدمة

البنية التحتية كخدمة

تتمثل القدرة المقدمة للمستهلك في توفير المعالجة والتخزين والشبكات وغيرها من موارد الحوسبة الأساسية حيث يكون المستهلك قادرًا على نشر وتشغيل البرامج التصفية ، والتي يمكن أن تشمل أنظمة التشغيل والتطبيقات. لا يدير المستهلك أو يتحكم في البنية التحتية السحابية الأساسية ولكنه يتحكم في أنظمة التشغيل والتخزين والتطبيقات المنشورة وربما تحكمًا محدودًا في مكونات الشبكات المحددة (على سبيل المثال ، جدران الحماية المضيفة).

وزارة التجارة الأمريكية ، 2011 ، NIST —

يوضح الشكل 5-6 بشكل أفضل التعريف الذي وصفه المعهد الوطني الأمريكي للمعايير والتكنولوجيا مقارنةً بتلك التي لا تتم ، IaaS يُظهر المكونات التي يديرها مزود السحابة في عرض (NIST) إدارتها. قد يكون موفر السحابة هنا هو مورد السحابة ، أو فريق العمليات الخاص بالمؤسسة ، اعتمادًا على ما إذا كانت السحابة يديرها البائع أو تدار ذاتيًا



بصريًا ، أي مقدار - PaaS و IaaS - يصف الشكل 6-5 أيضًا ما يميز نموذجي اعتماد السحابة المقدس الذي يديره المستخدم (أو العميل) مقابل مقدار ما تتم إدارته بواسطة موثر النظام الأساسي السحابي ، والذي يمكنه مرة أخرى أن تكون مورّدًا أو فريق العمليات الخاص بمؤسسة ما. سأتأقش PaaS لاحقًا في هذا الفصل ، في القسم الخاص بـ PaaS للاستفادة من DevOps تقديم لهذه البيانات هو إضافة طبقة التجريد التي IaaS الهدف من الاستفادة من النظام الأساسي السحابي ناقشتها من قبل بين الممارسين الذين يقدمون البرامج والبنية التحتية. إنهم يرون رؤية واحدة مبسطة للبنية التحتية ، بينما تحت السطح قد تواجه جميع التحديات الكلاسيكية وتحتاج إلى نفس مستويات الصيانة التي تحتاجها أي بنية تحتية للأجهزة

من المهم ملاحظة أنه ليس من غير المألوف رؤية المؤسسات تتخذ نهجًا مختلطًا لاعتماد منصة سحابية. على الرغم من أنهم قد يستفيدون من سحابة عامة أو مخصصة خارج مقر العمل لبيانات اختبار التطوير ، نظرًا لانخفاض التكاليف وعدم وجود احتياجات لمواطن البيانات (حيث يتم تخزين البيانات) لا توجد حاجة إلى بيانات إنتاج حقيقية لاختبار التطوير ، لذلك المؤسسات قد تختار الاحتفاظ ببيانات ، الإنتاج في أماكن العمل. (راجع قسم "اختبار إدارة البيانات" في الفصل 4 لمزيد من المعلومات.) وهذا يجعل التطبيقات قيد الإنتاج أقرب إلى مصادر البيانات ويسمح أيضًا بإدارة أفضل لمتطلبات الامتثال المتعلقة بـ البيانات. مع تطور النماذج مثل السحب المخصصة ، أصبحت هذه المخاوف أقل عقبة أمام اعتماد السحب الخارجية ، حتى للإنتاج

iaas PaaS PaaS PaaS PaaS

أنت بحاجة إلى IaaS وإدارته أن تقوم أولاً بإنشاء البنية اللازمة لسحابة IaaS يتطلب إنشاء عرض القدرات التالية ، التي تم التقاطها أيضًا في الشكل 7-5

- السمسرة والفهرس
- تنسيق الخدمة والتكامل
- تنسيق السحابة
- انتشار تزامن

223 مسرحيات لقيادة الابتكار devops الفصل 5

وإدارة الخدمة - للمراقبة والأمان والقياس والفوترة وما إلى ذلك ، لواجهات برمجة API أدوات التطبيقات والخدمات - تقع أيضاً في هذه المساحة

تنسيق السحابة

CoACHes أو ChestrAtIng the Blg gAMe

تصور للمدربين الذين يستعدون للمباراة الكبيرة. إنها تنطوي على العاطفة والطاقة Fandom لدى هذا صحيح ، لكن العاطفة والطاقة مجرد عوامل ، مثل نحلان في خلية. أكثر من ذلك بكثير. إن الطريقة التي يختار بها طاقم التدريب إعداد فريق للعبة التنافس الوحشية والمحمية التي اتخذت طابعاً خارجياً لمباراة ضعيفة أمر مهم للغاية. تماماً مثل قادة الفرق الموسيقية الذين يقررون كيفية لعب على المدربين تحديد نغمة ، Berlioz □□□□□ من *Symphonie Fantastique* لعب لاحظ هذا: التحضير بدأ قبل وقت Notre Dame و Michigan كاملة لإعداد فريق لمباراة مثلطويل من يوم الاثنين ، عندما يكون أي شخص على وجه التحديد تخمين لا يوجد تسامح مع أي انحرافات في فترة التحضير للعبة ، باستثناء الخلاف العرضي "الأولاد سيكونون الأولاد" في الملعب. حتى

□□□□□

□□□□□

يمكن التلاعب بهذه الحلقات من قبل الموظفين واستخدامها لمزيد من التحفيز. لا داعي للقلق بشأن نقص التركيز إذا اشترى الجميع المهمة. يتحول المنسقون (المدربون) إلى جنون عندما يبتعد اللاعب عن المسار ويسبب نوعاً من الإلهاء غير الضروري ، وعادة ما يكون الفم مفرط النشاط. لا يريد المدربون أن يكون للخصم المزيد من الحافز. هناك بالفعل إمدادات وفيرة

ولفيرين ، 2014—

يتضمن تنسيق السحابة إدارة أتمتة المكونات والخدمات المختلفة التي تقدمها البيانات السحابية الهدف من التنسيق هو إدارة التزويد وإلغاء التزويد وتخصيص الموارد والتكوين وسير العمل لجميع المكونات في مكدس التكنولوجيا السحابية. وجميع خدمات البنية التحتية التي يتم تقديمها بواسطة السحابة. تتمثل إحدى طرق تعريف التنسيق في النظر إليه على أنه تدوين لأفضل الممارسات وسير IaaS العمل لإدارة سحابة

تشمل قدرات تنسيق السحابة ما يلي (بيرنانادام ، 2012)

تكمّل إمكانات السحابة عبر البيانات والبنية التحتية غير المتجانسة لتبسيط نشر الخدمة وأتمتة وتحسينها

■ أتمتة للسماح بنسبة أقل من المسؤولين للخوادم الفعلية والظاهرية

■ التوفير الآلي واسع النطاق وإلغاء توفير الموارد باستخدام الأدوات المستندة إلى السياسة لإدارة

انتشار الآلة الافتراضية عن طريق استعادة الموارد تلقائياً

224 مسرحيات لقيادة الابتكار devops الفصل 5

- القدرة على دمج سير العمل وسلاسل الموافقة عبر صوامع التكنولوجيا لتحسين التعاون وتقليل التأخيرات
- مراقبة في الوقت الفعلي لموارد السحابة المادية والافتراضية ، فضلاً عن إمكانيات الاستخدام وإعادة الشحن المحاسبي للتتبع والتحسين
- استخدام النظام
- قوالب الأتمتة المعبأة مسبقاً ومهام سير العمل لأنواع الموارد الأكثر شيوعاً لتسهيل اعتماد أفضل الممارسات وتقليل وقت الانتقال

مثالين على رواد السوق VMware vRealize و IBM Cloud Orchestrator يعد كل من في مساحة تنسيق السحابة

نموذج كلود

حل قابل للتكرار يعتمد على مجموعات محددة من الصور الافتراضية والبرامج الوسيطة [A] والتطبيقات وتكوينات وقت التشغيل. نتيجة نشر النمط هي بيئة تطبيق مهيأة ومضبوطة ومحسنة

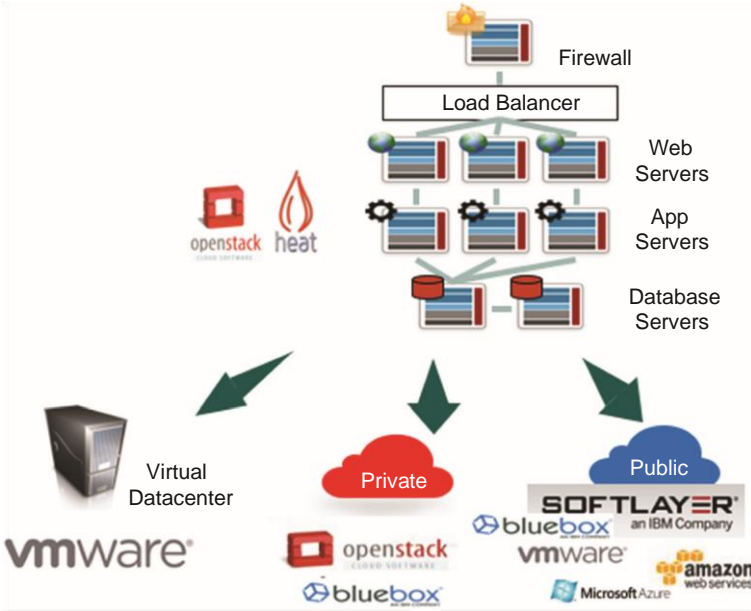
شيارا براندل ، 2014 —

يمكن تسجيل التنسيق ، جنباً إلى جنب مع التعريف المعماري ، في □□□□□ □□□□□ والتي تسمى □□□□□ □□□□□ ، OpenStack Heat المعيار الشائع للأنماط هو أنماط (HOTS). □□□□□□□

في حين أنها قد نضجت بشكل كبير في العامين الماضيين ، إلا أن أنماط تعريف السحابة وتدوينها رائدة في هذا المجال عندما ظهرت أنماط السحابة الخاصة بها لأول IBM ليست جديدة. كانت شركة IBM في عام 2009 ، والذي تطور في عام 2011 إلى CloudBurst مرة مع إنشاء جهاز قد ، (vSys) هذه الأنماط ، التي تسمى أنماط النظام الافتراضية. (IWD) Workload Deployer IBM PureApplication Systems (PureAS) تطورت أيضاً وهي مدعومة اليوم بشكل أساسي من خلال عروض واستبدالها ب IBM في vSys يتم التخلص التدريجي من أنماط OpenStack Heat خارج IBM كـ معيار. خارج OpenStack Heat ، التي تم إصدارها في عام 2011 ، (AWS) Amazon Web Services CloudFormation الواسعة Amazon الأكثر شعبية ، نظراً لبصمة

تطوير معيار المجتمع في قوالب AWS CloudFormation من منظور المعايير ، ألهم يدعم معظم بائعي السحابة OpenStack Heat. مما يدعم محرك توفير ، OpenStack HOT ، مما ، OpenStack مباشرة أو يقدمون دعماً لواجهات برمجة تطبيقات OpenStack Heat اليوم OpenStack. لتوفير البيانات على هذه السحابة غير التابعة لـ Heat يسمح لـ فيما يلي بعض الأمثلة على مكونات السحابة المحددة والمضمنة في نمط

- التثبيت المسبق على نظام التشغيل
- التكامل المسبق عبر المكونات



الشكل 8-5: حرارة مكدسة تدعم غيومًا متعددة

في منشور المدونة الذي أشرت إليه Frederick “Freddie” Sudhakar كما ذكر صديقي بعض OpenStack. هنا ، لم تتم معالجة إمكانية النقل عبر السحب بشكل كامل حتى الآن ، حتى مع الأشياء في بعض السحابة لن يكون لها ببساطة ما يعادلها في السحب الأخرى ، لذلك كلما كانت الخدمات أقل قابلية للنقل. منصة HOT الخاصة والسحابة المستخدمة أكثر خصوصية ، أصبح مستند خدمة

منصة خدمة

تتمثل القدرة المقدمة للمستهلك في النشر على تطبيقات البنية التحتية السحابية التي أنشأها المستهلك أو حصل عليها والتي تم إنشاؤها باستخدام لغات وأدوات البرمجة التي يدعمها الموفر لا يدير المستهلك أو يتحكم في البنية التحتية السحابية الأساسية بما في ذلك الشبكة أو الخوادم أو أنظمة التشغيل أو التخزين ، ولكنه يتحكم في التطبيقات المنشورة وربما تكوينات بيئة استضافة التطبيقات.

—NIST ، 2011 ، وزارة التجارة الأمريكية

مثل معظم المصطلحات في صناعة تكنولوجيا المعلومات (أو للتعميم حقًا ، في وسيط الاتصال محملة بشكل زائد ، ومفرط في الاستخدام ، ويساء PaaS البشري المعروف باسم □□□□) ، فإن هذه النقطة PaaS على Wikipedia فهمها. يثبت البحث السريع على الويب أو حتى زيارة صفحة

PaaS

عن نموذج استهلاك "البنية PaaS والذي يميز ، PaaS - NIST القسم الأكثر أهمية في تعريف :التحتية كخدمة" للسحابة ، هو كما يلي

```

... 
□□□□□□ □□ □□□□□□ □□ □□□□□□□□ □□□□ □□
□□□ □□ □□□ □□□□□□□□ □□□□□□□□ □□□□□□□□
□□ □□□□□□□ □□□□□ □□ □□□□□□□ □□ □□□□□□
□□□□□□□□□ □□ □□□□□ □□□□□ □ □□□□□□□□
□□□□□□□ □□□□ □□□□□□□ □□□□□ □□□□□□□□
□□□□□□□□□□.

```

كما ترى في الشكل ، فإن ما يميز PaaS و IaaS يوضح الشكل 5-6 بشكل أفضل الاختلافات بين هو مقدار المكس الذي يديره المستخدم (أو العميل) - PaaS و IaaS - نمودجي اعتماد السحابة مقابل مقدار ما تتم إدارته بواسطة موثر النظام الأساسي السحابي (والذي قد يكون فريق العمليات الداخلية أو بائع السحابة الخارجية). من المهم إبراز أمرين

- كل قدرة في المكس متاحة كخدمة مُدارة ؛ ستكون هذه خدمة مشتركة متعددة المستأجرين متاحة للمستخدمين ، مع استخلاص التطبيق الأساسي من المستخدم
- يجب أن يهتم العميل / المستخدم فقط بإدارة التطبيقات والبيانات الخاصة به ووصول المستخدم والاستفادة من الخدمات المتاحة على النظام الأساسي ، الذي يدير الباقي ،

Salesforce و Pivotal و IBM Bluemix قد تكون الأنظمة الأساسية كخدمة إما عامة ؛ Cloud Foundry وما إلى ذلك) أو خاصة (مستضافة Google App Engine و Heroku نظام أساسي مبني ذاتياً ومستضاف). قد تقوم المنظمات أيضاً ببناء بيئاتها المستضافة والمدارة الخاصة ، بها بالاستفادة من تقنيات متعددة والاستضافة الذاتية على سحابة خاصة. على سبيل المثال ، بالطبع ستكون المنظمة التي تستضيفها في مركز البيانات الخاص بها مسؤولة عن إدارة جميع الخدمات المدار ذاتياً Cloud Foundry بالاستفادة من PaaS المستضافة على النظام الأساسي. سيكون بناء مثلاً جيداً Pivotal من

PaaS DevOps

عليك إلقاء نظرة على الخدمات التي يجب استضافتها على ، PaaS على DevOps عندما تنظر إلى على النظام الأساسي. إذا نظرت DevOps النظام الأساسي من أجل تنفيذ خط أنابيب تسليم تطبيقات والمكونات الأساسية التي تشكل خط أنابيب التسليم ، فيجب DevOps إلى خط أنابيب تسليم تطبيقات . PaaS □□ □□□□ □□□□□□□□ DevOps أن يكون كل مكون متاحاً كخدمة من أجل توفير عند إعادة النظر في قائمة القدرات التي تشكل خط أنابيب توصيل ، يجب إتاحتها جميعاً كخدمات في

PaaS:

■ بناء

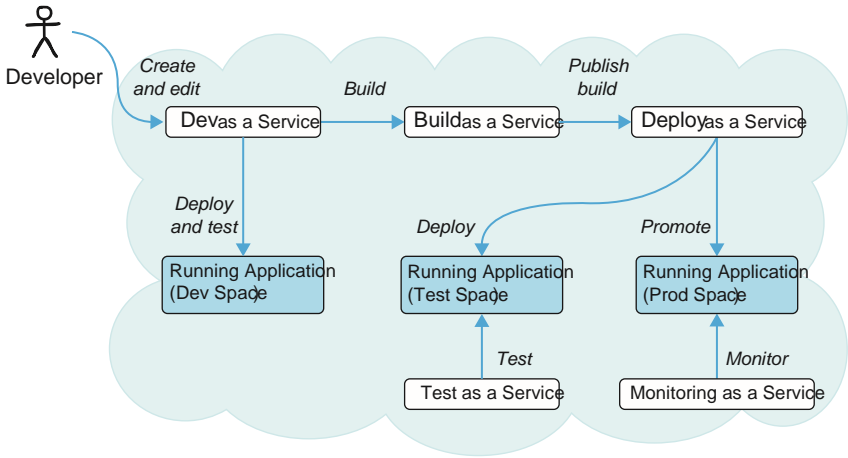
■ التكامل المستمر

■ أتمتة النشر

■ تكوين البرامج الوسيطة

■ الاختبار الآلي

الموضحة في الشكل 5-9. إذا انتبهت ، ستلاحظ أن هذه DevOps هذه هي ما تشير إليه بخدمات تتضمن القدرات المتعلقة PaaS وذلك لأن IaaS مجموعة فرعية من القائمة المعروضة في قسم هو إنشاء بيئات PaaS الهدف الكامل من PaaS بتوفير البيئة وإدارة التكوين كخدمات أساسية لـ مستخرجة بعيداً عن مستهلكي النظام الأساسي



الشكل 5-9: خدمات التطوير

DevOps خدمات IBM Bluemix التي تعمل بكامل طاقتها وتديرها مثل PaaS تشمل عروض Bluemix على DevOps كجزء لا يتجزأ من النظام الأساسي. فيما يلي بعض خدمات

■ خدمة GitHub Enterprise كخدمة ، و Git

■ المستند إلى الويب IDE

■ التخطيط السريع والتتبع والتعاون الجماعي كخدمة

■ خط أنابيب التسليم كخدمة

■ العولمة كخدمة

■ أتمتة النشر كخدمة

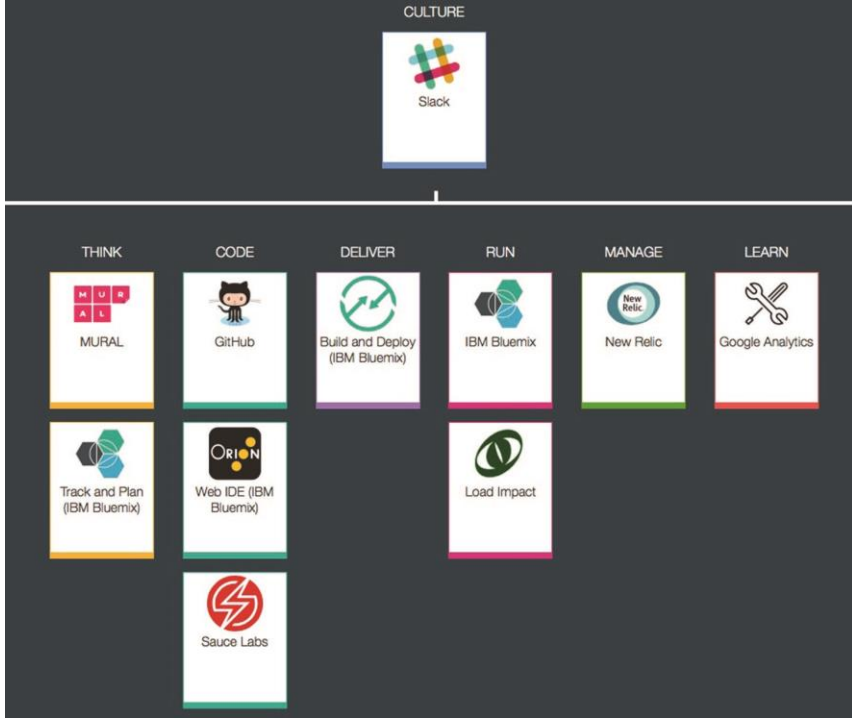
■ التحجيم التلقائي كخدمة

■ مراقبة الأداء كخدمة

■ تنبيه الإخطار كخدمة

229 مسرحيات لقيادة الابتكار devops الفصل 5

IBM Bluemix PaaS توفر هذه الإمكانيات ، التي تعمل بالتنسيق ، خط أنابيب تسليم مستمر على . كما هو موضح في الشكل 10-5 ،



IBM Bluemix (IBM) الشكل 10-5 :خط أنابيب توصيل مطور

DevOps ؟؟؟؟؟

أمر بديهي. بالنسبة لأي مؤسسة تتطلع إلى اعتماد PaaS إن عرض القيمة لاعتماد نظام أساسي بتكلفة دخول DevOps اعتماد DevOps الذي يتضمن خدمات PaaS يتيح لك عرض ، DevOps منخفضة للغاية. لا تحتاج إلى صياغة خط أنابيب تسليم وتنفيذ سلسلة أدوات التسليم المستمر بأكملها. واسمح بالحجم ، Payas-you-go . عمليات الدمج والاستضافة والخدمة - هذه ليست مشكلتك

PaaS ؟؟ Cloud Foundry

الذي أصبح بلا شك المعيار لتقديم ، Cloud Foundry دون ذكر PaaS لا يمكنك ترك موضوع Cloud Foundry مفتوح المصدر تديره الآن مؤسسة PaaS هو عبارة عن نظام PaaS. Cloud Foundry Foundation. يتألف مجلس إدارة المؤسسة من ممثلين من كبار مستخدمي .VMware ، SAP و EMC و HP و Pivotal و IBM والموردين مثل Cloud Foundry IBM Bluemix Public يعد . Cloud Foundry القائمة على PaaS يدعمون أو يوفرون أنظمة Cloud Foundry القائم على PaaS اعتبارًا من كتابة هذا الكتاب ، أكبر مثال لـ ، PaaS

231 مسرحيات لقيادة الابتكار devops الفصل 5

وزناً بكثير من الأجهزة الافتراضية ، مما يسمح لها بالعمل على خادم على نطاق أعلى بكثير من الأجهزة أكثر قابلية للنقل ، مما Linux كمشروع مفتوح المصدر لجعل حاويات Docker الافتراضية. بدأ منذ إنطلاقه في عام 2013 Cloud. أو على أي خادم فعلي أو في Linux يسمح بنقلها عبر أي مثل أحد أكثر المشاريع مفتوحة المصدر نجاحاً على الإطلاق ، مع أكثر من 100 مليون Docker أصبح ، Docker (Martin N. ، 2015) عملية تنزيل لمحرك

باختصار ، تتمثل الميزة الرئيسية للحاويات في أنها توفر طريقة قياسية لحزم تطبيق ما ، وتكويناته وجميع تبعياته ، بحيث يصبح قابلاً للنقل عبر البيئات. يتم عزل رمز التطبيق الفريد داخل الحاوية عن ، البنية خارج الحاوية. لا يحتاج المطورون إلى الاهتمام بماهية DevOps البيئة خارج الحاوية. إنه تقريباً . يركزون على تغليف تطبيقاتهم في صورة الحاوية القياسية الأفضل لتطبيقهم وبالمثل ، لا يحتاج فريق العمليات إلى القلق بشأن تغيير الكود والتكوينات والتبعيات الخاصة بالتطبيقات داخل الحاوية. إنهم يحتاجون فقط إلى التركيز على تشغيل الحاويات القياسية المعتمدة في بيئتهم

DevOps

يمكن نشر الأدوات التي تستخدمها المؤسسات لبناء خط أنابيب تسليم التطبيقات في حاويات ، مما يجعلها متاحة كخدمات للاستفادة من تسليم إمكانات خط تسليم التطبيقات ، كما ناقشت سابقاً. بدلاً من ذلك ، يمكن نشر أدوات خط أنابيب تسليم التطبيقات في أي مكان خارج الحاويات بطريقة تقليدية لتطوير التطبيقات والخدمات ، وتعبئتها فقط في حاويات عندما تحتاج إلى نشرها للاختبار ، وما قبل الإنتاج وبيئات الإنتاج. في أي من السيناريوهين ، تتمثل الميزة الرئيسية للحاويات في توفير مجموعة موحدة والبيئات الأخرى حيث يمكن نشر التطبيق في حاويات محمولة وترقيته من بيئة devtest-prod من إلى أخرى دون أي مخاوف تتعلق بالتوافق أو إدارة التكوين التي يتعين القيام بها عندما يتعلق الأمر بخط أنابيب تسليم التطبيق ، فإن التغيير الوحيد الذي يجب إجراؤه هو عملية الإنشاء ، والتي تتضمن الآن خطوة تعبئة التطبيق في الحاوية. تدعم أدوات البناء الحديثة ، مثل الإمكانات التالية ، Jenkins و IBM UrbanCode Deploy

■ Dockerfile من Docker إنشاء صورة

■ إلى سجل Docker نشر صورة

في البيئة المرغوبة ، والاستفادة من أدوات Docker بمجرد إنشائها ، يمكن نشر حاوية أو إذا كانت هناك حاجة إلى مستوى أعلى من إمكانية النشر ، فيمكن استخدام أداة أتمتة Docker ، IBM UrbanCode Deploy للنشر مثل CoreOS من المهم ملاحظة أن هناك العديد من تقنيات الحاويات في السوق ، مثل حاويات هي الشركة الرائدة في السوق إلى حد Docker ومع ذلك ، فإن VMWare Photon و Rocket بعيد ، مع اعتماد السوق بشكل كبير على البدائل. في أبريل 2016 ، أطلق العشرات من البانعين مبادرة الحاوية المفتوحة لتوحيد تنسيقات الحاويات وأوقات التشغيل

ينتج عن قابلية التوسع الهائلة للحاويات حالات متعددة من الحاويات المتعددة التي يتم نشرها. تتضمن إدارة هذه الحاويات ، عبر دورة حياتها ، عدة أنواع من المهام والإمكانات ، بما في ذلك ما يلي

- الانتشار
- تحديثات الحاويات
- التزويد
- اكتشاف الحاويات
- المراقبة
- الجدولة
- التجميع والتجميع
- تجاوز الفشل
- إدارة السياسات
- إدارة القيد

نتيجة للتحديات المقدمة لتوفير هذه القدرات ، ظهرت عدة تقنيات لإدارة تنسيق الحاويات

- في تقديم الدعم لمجموعات كبيرة من حاويات Docker ، بدأ Docker مستفيداً من واجهات برمجة تطبيقات ، Docker Swarm باستخدام Docker بدلاً من واحدة في كل مرة ، Docker الأساسية لإدارة مجموعة من محركات تقنية التزامن في مساحة الحاوية التي تحصل على أكبر قدر من الجر هي . **Kubernetes** أنها تدير أكثر من ملياري حاوية يوميًا Google تدعي **Kubernetes for Google** هي Kubernetes مما يوفر بعض المصادقية الجادة. هندسة ، Kubernetes باستخدام تلك الخاصة Kubernetes Kubernetes المتعددة . يدير السيد عمليات الإدارة والتنسيق لإدارة جميع التوابع. هذه Kubernetes بدورها لديها مجموعات متعددة من الحاويات تسمى Kubernetes المنتشرة عليها
- إدارة Docker هو مشروع مفتوح المصدر كان موجوداً بشكل مستقل عن Mesos : Mesos المهام المعقدة قيد التشغيل على مجموعة مشتركة من الخوادم. ومنذ ذلك الحين ، أضاف دعمًا نموذج Mesos لدى ، Kubernetes مما يسمح بإدارة الحاويات. على غرار ، Docker لـ Kubernetes . يدير السيد المهام عالية المستوى ويفوض المهام إلى الذين يديرون الحاويات. مستوى أعلى من التحكم والحجم والتوافر العالي والذي يمكنه تنسيق مجموعة من العقد ، ZooKeeper يتم توفيرها بواسطة برنامج يسمى بإعلام جميع الأسياء والعبيد الذي يعتبر السيد الرائد الحالي ZooKeeper الرئيسية. يقوم

????? ?????????

PaaS شائعة. هنا ، مثل (CaaS) أصبحت عروض PaaS كبدل لـ الداخلية أو بواسطة بائع. يحتاج المطورون Ops تتوفر البينات كخدمة ويتم إدارتها بواسطة فرق ، فقط إلى تشغيل الحاويات التي يبنونها على الخدمة ، دون أي مخاوف بشأن إعداد البنية أو إدارتها أو بطبيعتها أي تطبيقات أو خدمات وسيطة. إذا احتاج التطبيق CaaS لا توفر ، PaaS تكوينها. بخلاف فسيحتاجون إلى نشر حاوية تقوم بتشغيل الخدمة على ، CaaS إلى أي من هذه الخدمات ، بخلاف أو الاستفادة من خدمة قيد التشغيل خارجيًا ، مباشرةً من التطبيق الذي يحتاج إلى استهلاكها CaaS

IBM Bluemix وحاويات ، (سابقًا Tutum) Docker Cloud الشائعة CaaS تشمل خدمات و ، Rancher Labs من Rancher و ، CoreOS Tectonic و ، Amazon ECS و ، Google Container Engine.

تقديم بني الخدمات المصغرة: play

فرق المنظمات مقابل الفرق المتخصصة

تسعة مواسم في الدوري وساهم كرجل أساسي Coy Wire السابق NFL لعبت سلامة ولاعب في فرق خاصة. لقد أقر بأن تغطية الركلات ليست على رأس قائمة الجميع عندما يأتون إلى اتحاد كرة القدم الأميركي ، لكنه تحدث عن أهمية التكيف مع دور جديد كمحترف معظم اللاعبين الذين يلعبون دور البطولة في فرق خاصة في اتحاد كرة القدم الأميركي لم يقللوا ، من شأنها في حياتهم الجامعية. لذا ، عليك أن تكون مثل الحرياء وأن تتكيف مع الظروف الجديدة قال واير. "كل لاعب في فرق خاصة يتمنى لو كان لاعبًا أساسيًا بدوام كامل في اتحاد كرة القدم" الأميركي. ولكن لكي تشتري حقًا وحدات الفرق الخاصة وتتفوق عليها ، عليك أن تضع ذاتك جانبًا ". وتتبنى الدور الجديد الذي تلعبه لفريقك

بوين ، 2015—

يعد □□□□□ □□□□□ موضوعًا أساسيًا - كما هو ، DevOps من منظور موضح في الفصل 4. إن عرض القيمة للدفعات الصغيرة أمر بديهي: التغييرات الصغيرة التي يتم تقديمها بشكل متكرر ، مما يقلل من □□□□□ □□□□□ للتعليقات ، ويقلل من تأثير التغيير. يصبح الاختبار والتحقق من الأمان أسهل ، حيث تقوم بالتحقق من التغييرات الصغيرة بشكل متكرر. يصبح النشر أسهل DevOps. عندما تقوم بنشر مجموعات أصغر من التغييرات. وبالمثل ، تصبح إدارة التغيير أقل تعقيدًا نيرفانا

من منظور معماري ، مع ذلك ، فإن تسليم دفعات صغيرة ليس دائمًا قابلاً للتطبيق. معظم تطبيقات المؤسسة □□□□□□□ في طبيعتها. لديهم عدد قليل من المكونات الكبيرة ، يتم تسليم كل منها كإصدار واحد قابل للنشر

والتي تكون عادةً مكونًا واحدًا لكل نوع من أنواع واجهة ، (UI) □□□□□ المستخدم المستخدم - صفحة الويب ، وتطبيق الجوال ، وواجهة برمجة التطبيقات لتطبيقات الجهات الخارجية ، وما إلى ذلك

□□□□□ □□□□□□□□□□ والتي تنتشر عادةً عبر قواعد بيانات متعددة تعمل على الخوادم

مكونات □□□□□□□□□□ ، والتي قد تكون عدة مكونات قابلة للتنفيذ ، على الرغم من أنها إذا كانت أكثر من مكون ، فإن المكونات كبيرة وقليلة العدد

لذلك ، من منظور النشر ، لديك ما لا يقل عن ثلاث وحدات □□□□□□□□□□ لنشرها
يمكن تطوير كل هذه المكونات من قبل فرق متعددة ، ومن المحتمل أن تستخدم □□□□□□
، □□□□□□□□□□ تقدم مجموعات صغيرة من التغييرات على الكود مع كل سباق. ومع ذلك
يجب دمج جميع هذه التغييرات ودمجها في المكون الوحيد الكبير القابل للنشر. هذا يلغي الغرض من
تسليم دفعات صغيرة. بينما تقدم كل دورة تسليم تغييرات صغيرة وتزايدية للمكون القابل للنشر ، يجب
تسليم المكون نفسه على أنه نشر □□□□□□□□□□ للمكون المتجانس

يتمثل التحدي التالي من منظور تزامن النشر في نشر مثيلات متعددة للمكونات. إذا كان أي مكون
سيتم تحجيمه ، فلا يمكن قياسه □□□□□□□□□□ خلال نشر مثيلات متعددة على خوادم
متعددة. هنا مرة أخرى ، حتى لو كان هناك جزء فقط من الوظيفة بحاجة إلى التوسع - على سبيل المثال
عربة التسوق لموقع التجارة الإلكترونية - تتطلب البنية المتجانسة نشر مثيلات متعددة للمكون بأكمله ،
المكون الأحادي هو أدنى الأصول الذرية القابلة للنشر

هندسة الخدمات المصغرة

أسلوب الخدمات الدقيقة

الأسلوب المعماري للخدمة المصغرة هو أسلوب لتطوير تطبيق واحد كمجموعة من الخدمات [T] الصغيرة ، كل منها يعمل في عملياته الخاصة ويتواصل مع آليات خفيفة الوزن ، غالبًا ما تكون تم بناء هذه الخدمات حول إمكانات العمل ويمكن نشرها HTTP واجهة برمجة تطبيقات لموارد بشكل مستقل بواسطة آلية نشر مؤتمتة بالكامل. هناك حد أدنى من الإدارة المركزية لهذه الخدمات و التي يمكن كتابتها بلغات برمجة مختلفة واستخدام تقنيات تخزين بيانات مختلفة ،

— 2014 ، فاولر

تتم معالجة الحلول لكل من تقديم دفعات صغيرة وقابلية التوسع (الشكل 5-11) بشكل فعال من خلال نهج الخدمات المصغرة. يحدد مارتين فاولر الخصائص الرئيسية للخدمات المصغرة على النحو التالي

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ . تسمح الخدمات المصغرة بتطوير وتقديم خدمات مكونة من ☐ ☐ ، مع واجهات برمجة تطبيقات محددة جيداً ، والتي يمكن معًا لتقديم خدمات معقدة.

السحابة الأصلية

قبل الانتقال ، من الضروري تحديد مصطلح □□□□□□ □□□□□□ □□□□□□ الذي قدمته في هذا القسم بشكل أفضل. تطبيقات السحابة الأصلية هي تتويج لتجمع المفاهيم التالية معاً

■ DevOps

■ الخدمات المصغرة

■ حاويات

هذه المفاهيم - التي يتم دمجها مع الهدف الوحيد لتقديم التطبيقات التي تم إنشاؤها للسحابة والتي تحتاج إلى نموذج تطوير وتسليم وعمليات جديد للتشغيل على نطاق سحابي - لديها القدرة على إدارة التنسيق المطلوب لهذا النطاق وتقديم بيئة قابلة للكسر قادرة على التعامل مع هذا الحجم

خصائص النظام غير المتوفرة

- تعبئة الحاوية. تشغيل التطبيقات والعمليات في حاويات البرامج كوحدة معزولة لنشر التطبيق وكالدية لتحقيق مستويات عالية من عزل الموارد. يعمل على تحسين تجربة المطور الإجمالية وتعزيز إعادة استخدام التعليمات البرمجية والمكونات ، وتبسيط العمليات للتطبيقات السحابية ، الأصلية.
- تدار ديناميكياً. مجدولة بنشاط وتتم إدارتها بشكل فعال من خلال عملية تنظيم مركزية. تحسين كفاءة الماكينة واستخدام الموارد بشكل جذري مع تقليل التكلفة المرتبطة بالصيانة والعمليات.
- الموجهة نحو الخدمات الصغيرة. مقترنة بشكل فضفاض مع التبعيات الموصوفة صراحة (على سبيل المثال ، من خلال نقاط نهاية الخدمة). يزيد بشكل كبير من السرعة الإجمالية وقابلية الصيانة للتطبيقات. ستشكل المؤسسة تطور التكنولوجيا لتطوير أحدث ما توصلت إليه إدارة التطبيقات ولجعل التكنولوجيا في كل مكان ومتاحة بسهولة من خلال واجهات موثوقة

— 2015 ، Native Computing Foundation كلود—

من أجل التشغيل - الاستفادة CaaS أو PaaS ، تتطلب تطبيقات السحابة الأصلية ، بحكم تعريفها من الحاويات للنشر والتشغيل في أي من نموذج السحابة. تم تصميمها باستخدام بنية الخدمات المصغرة تم تطويرها وتسليمها باستخدام منهجية تطبيق 12 عاملاً. دفعات صغيرة ، والتسليم السريع ، وقابلية التوسع ، ومقاومة الهشاشة هي بعض من الموضوعات الأساسية

Dev تطوير □□□□□□ □□□□□□ □□□□□□ وتقديمها أن يكون لدى فريقي فهم جيد لكيفية اختلاف هذه التطبيقات عن التطبيقات التقليدية في العديد من الافتراضات Ops و (Brown ، 2016): حول أحد التطبيقات Ops و Dev الأساسية التي وضعها فريق

في تطبيق السحابة الأصلي ، فإن التطبيق والخدمات نفسها هي التي توفر أي متطلبات غير ■ بواسطة البنية NFR على عكس التطبيقات التقليدية حيث يتم توفير (NFR) وظيفية التحتية. تشمل الأمثلة موازنة التحميل والتوافر العالي ومراقبة التطبيق

239 مسرحيات لقيادة الابتكار devops الفصل 5

■ تتغير البنية التحتية باستمرار (فهي □□□□ بطبيعتها) ، على عكس التطبيقات التقليدية ، التي تعمل على بنية أساسية ثابتة ذات هيكل ثابت

■ قد يتم توزيع مكونات التطبيق عالميًا ، عبر بيئات سحابية متعددة ، تعمل كخدمات ، على عكس التطبيق التقليدي حيث توجد المكونات عادةً في نفس البيئة ، وهي مصنوعة من خوادم مشتركة

■ في خوادم الإنتاج ، بدلاً من التطبيق التقليدي حيث يكون فريق DevOps يتحكم أعضاء فريق مسؤولاً عن تشغيل خوادم الإنتاج. بالنسبة إلى التطبيق السحابي الأصلي ، يقوم فريق Ops الذي يمتلك التطبيق DevOps بتشغيل النظام الأساسي أو خدمة الحاويات. يصبح فريق Ops لتشغيل مثيلات الإنتاج الخاصة بهم على النظام الأساسي - سواء كانت مثيلات خادم فردية أو حاويات

■ مسؤولية التأكد من استمرار عمل DevOps في حالة حدوث كارثة ، تقع على عاتق فريق إلى الإنتاج ، ومن Dev التطبيق ، على عكس التطبيقات التقليدية حيث يوجد تسليم رسمي من بتشغيل التطبيق باستخدام الحد الأدنى من مشاركة Ops هذه النقطة فصاعدًا ، يقوم فريق النظام Ops إن وجدت. مرة أخرى ، بالنسبة للتطبيقات السحابية الأصلية ، يدير فريق ، Dev الأساسي أو خدمة الحاويات. طالما أن خدمات النظام الأساسي الأساسية متوفرة ، فهي جيدة الذي يمتلك DevOps في حالة تعطل تطبيق يعمل على النظام الأساسي ، فإن مسؤولية فريق التطبيق هو إعادة تشغيله مرة أخرى

لا يعد تطوير تطبيقات السحابة الأصلية وتقديمها مجرد تحول هيكلي أو منهجي. يجب تغيير التكوين الكامل للفريق ، وأدوار أعضاء الفريق ، والمهارات التي يحتاجون إليها من أجل تطوير وتقديم هذه التطبيقات بفعالية وكفاءة

الخدمات الدقيقة والحاويات

تكتسب الخدمات الصغيرة والحاويات قوة دفع جنبًا إلى جنب ، ولسبب وجيه. بينما لا يعتمد أي منهما على الآخر - لا تحتاج الحاويات إلى ما يتم تعبئته بداخلها لتكون خدمات صغيرة ، ولا تحتاج الخدمات المصغرة إلى نشر حاويات - اقترح القيمة للاستفادة من الحاويات لتمكين إنشاء الخدمات المصغرة ونشرها وتشغيلها هائل. هناك ثلاث فوائد رئيسية

1. □□□□□□ □□□□□□ . بحكم التعريف ، تتسبب الحاويات في استخلاص ما يتم تشغيله بداخلها من البيئات التي تعمل عليها الحاويات. إمكانية نقل الحاويات هي الفكرة كلها. بالنسبة للخدمة المصغرة ، فهم دائمًا يستفيدون من خدمات البيئة بنفس الطريقة ، بغض النظر عن البيئة التي يتم نشر الحاويات فيها

2. □□□□□□ □□□□□□ . نظرًا لأن الحاويات أخف بكثير من الأجهزة الافتراضية الكاملة ، فيمكن تحجيمها بتكلفة أقل بكثير. وبالتالي ، يمكن توسيع نطاق الخدمات الصغيرة التي يديرونها بشكل أفقي بسهولة أكبر دون الحاجة إلى فرض ضرائب غير فعالة على البنية التحتية الأساسية

3. □□□□□□ . يمكن تشغيل الخدمات المصغرة في الحاويات كعمليات منفصلة ومستقلة ، غير مرتبط بمكان وكيفية تشغيل الخدمات الأخرى التي يستهلكونها أو يتفاعلون معها. يمكن تشغيل

240 مسرحيات لقيادة الابتكار devops الفصل 5

خدمات مصغرة متعددة على نفس طبعة الخادم ، أو توزيعها عبر مثيلات خادم متعددة ، دون القلق بشأن التداخل بين الخدمات المصغرة المتعددة. وبالتالي ، يمكن تخطيط وإدارة موازنة التحميل بشكل أفضل بكثير لزيادة استخدام الخادم ، ويمكن تحقيق التكرار لخدمة مصغرة بشكل أفضل عن طريق تشغيل مثيلات متعددة من نفس الخدمة المصغرة عبر مثيلات خادم متعددة

Microservices الترحيل إلى

تبسيط لعبة كرة القدم كمينج

هو أن المسرحيات - مسرحيات التمرير على وجه Erhardt-Perkins العمود الفقري لنظام الخصوص - لا يتم تنظيمها بواسطة شجرة مسار أو عن طريق استدعاء مسار مستقبل واحد ولكن من خلال ما يشير إليه المدربون باسم "المفاهيم". كل مسرحية لها اسم و

□□□□□

□□□□□

هذا الاسم يستحضر صورة لكل من قورترك والملاعبين الآخرين في الهجوم. والأهم من ذلك ، يمكن استدعاء المفهوم من أي تكوين أو مجموعة تقريبًا. من يفعل ماذا يتغير ، لكن النظرية والتكتيكات التي تقود المسرحية لا تتغير. قال بيركنز: "من حيث الجوهر ، أنت تدير نفس المسرحية". "أنت فقط تمنحهم بعض تزيين النوافذ لجعلها تبدو مختلفة"

:أكبر ميزة للنظام القائم على المفاهيم هي أنه يعمل من منظور اللاعب الأكثر أهمية في الهجوم قورترك. في أنظمة أخرى ، حتى لو كانت المبادئ الأساسية هي نفسها تمامًا ، فقد تختلف المسرحية واسمها كثيرًا. بدلاً من التلاعب بكل هذه المعلومات في الوقت الفعلي ، يتعين على لاعب الوسط في قراءة ترتيب معين من أجهزة الاستقبال. "يمكنك تقليص عدد المسرحيات Erhardt-Perkins والحصول على مظاهر مختلفة عن تشكيلاتك ومن بداخلها. من السهل على اللاعبين التعلم. قال تشارلي وايس ، منسق هجوم باتريوتس السابق في عام 2000 ، إنه من الأسهل على لاعب الوسط أن يتعلم. "يمكنك الحصول على مظهر مختلف دون تغيير قراءاته. لا تحتاج إلى عدد مفتوح من المسرحيات".

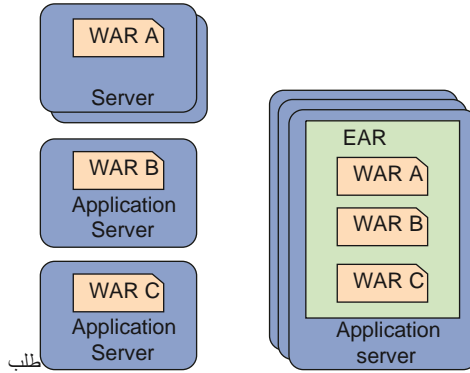
—Brown C.، 2013

لإغلاق هذه المسرحية لتقديم بنى الخدمات المصغرة ، دعنا نعود إلى الموضوع الأساسي للمسرحية: تطوير هياكل الخدمات المصغرة وتقديمها. يوجد سيناريو هنا

تطوير وتقديم تطبيقات السحابة الأصلية الجديدة ■

ترحيل □□□□□□□□ □□□□□□□□ □□□□□□□□ إلى الخدمات ■ المصغرة

ومبادئ التطبيقات Factor-يمكن تحقيق السيناريو الأول باستخدام الخدمات المصغرة وتطبيق 12 السحابة الأصلية والإرشادات المقدمة مسبقًا. هذا تطبيق جديد ، على الرغم من أنه قد يستهلك



في خدمات مصغرة Java الشكل 5-13: إعادة تجميع تطبيقات

■ . أنت الآن تبحث في أخذ التبعية المعمارية التي لا
معنى لها وتقسيمها إلى خدمات صغيرة ذات طبيعة محبة حقًا. يتطلب هذا الجهد النظر في
الكود لتحديد مكان وجود سلوكيات مدمجة في المكونات الحالية
ثم إعادة بناء الكود في مكونه الخاص. سيحتاج هذا المكون بعد ذلك إلى الحصول على واجهة
واضحة المعالم ، إذا لم تكن موجودة ، من أجل إنشاء خدمة مصغرة كاملة. اعتمادًا RESTful
على أنماط التصميم المستخدمة ، يمكن أن تكون عملية إعادة البناء هذه جهدًا معقدًا إلى حد
ما ويجب التعامل معه على هذا النحو

■ . لا يتعلق التحدي الأصعب لترحيل التطبيقات
الحالية إلى الخدمات المصغرة بالرمز على الإطلاق ، ولكن بالبيانات. على مر السنين ، تجعل
التطبيقات الحالية ذات مصادر البيانات الكبيرة والمتعددة في كثير من الأحيان وتدقق البيانات
المعقدة بين قواعد البيانات ومكونات التطبيق حتى فهم ما يتم تخزينه ولماذا مهمة معقدة. ومع
ذلك ، عندما يتعلق الأمر بإعادة هيكلة التطبيقات إلى خدمات مصغرة ، تصبح إعادة بناء نماذج
بنية البيانات أمرًا بالغ الأهمية. لكي تتصرف الخدمات المصغرة وتعمل بشكل مستقل عن
الخدمات المصغرة الأخرى ، يجب أن تكون قادرة على التفاعل مع بنية البيانات الصحيحة
المخزنة في مخزن البيانات الصحيح. لتحقيق ذلك ، يجب إعادة النظر في جميع نماذج البيانات
وقرارات تخزين البيانات

■ هل يتم استخدام بنية البيانات الصحيحة؟

■ هل مخزن البيانات الصحيح قيد الاستخدام؟

■ هل يتم استخدام النهج الصحيح للاستعلام عن البيانات؟

في الماضي ، كانت قاعدة البيانات العلائقية هي الخيار الوحيد. نتج عن ذلك جميع تنسيقات
، المهيكلة إلى نماذج بيانات الرسم البياني Java البيانات ، من البيانات الثنائية إلى كائنات
وكلها مخزنة في قواعد البيانات العلائقية. علاوة على ذلك ، تم تطبيع قواعد البيانات العلائقية
بشكل كبير لتوفير المساحة. كل هذا أدى إلى هياكل بيانات علائقية شديدة التعقيد ، أو تخزين
الثنائيات كنقاط لا يمكن الاستعلام عنها بشكل فعال ، أو استعلامات معقدة للغاية للرسم
البيانية. توجد اليوم مخازن بيانات متعددة اللغات - من تخزين العناصر إلى مخازن مستندات
إلى قواعد بيانات الرسم البياني - لتخزين البيانات في مخزن بيانات يتناسب بشكل JSON
أفضل مع متطلبات التنسيق والاستعلام. مرة أخرى ، إعادة بناء ديون الشركات هذه ليست

تافهة. في الواقع ، عادةً ما تكون مهمة أكثر تعقيدًا من كود إعادة البناء ، نظرًا لتعقيد أنواع هياكل البيانات ونماذج البيانات ومخازن البيانات التي قد تكون متضمنة. ومع ذلك ، للحصول على بنية خدمات مصغرة حقيقية ، لا داعي للمراهنة □□□□□□□□ عليها

Api تطوير اقتصاد: play

StUpld! ، eCoNoMY إنه

بدأت أندية كرة القدم العالمية ، المقيدة بعجزها عن زيادة دخلها من أعمالها التقليدية ، في تبني استراتيجيات لتحويل نفسها إلى شركات رياضية وإعلامية حديثة. في ظل الرناسة الأخيرة لفلورنتينو بيريز ، يقدم نادي كرة القدم الإسباني ريال مدريد مثالاً جيداً على تطبيق هذه الرؤية الموسعة تضمنت إحدى الركائز الأساسية لهذا النموذج تصميم وتنفيذ إستراتيجية تسويق جديدة تهدف إلى تعزيز قيمة العلامة التجارية للنادي. أدى اعتماد هذا النموذج إلى زيادة كبيرة في الدخل من التسويق. لا يمكن إنكاره ، في هذا المجال ، أصبح ريال مدريد الرائد في كرة القدم العالمية

فورداسيل ، 2006—

تماماً مثل نوادي كرة القدم التي تبحث عن طرق مبتكرة لتطوير نماذج جديدة لتحقيق الدخل لزيادة قيمة علامتها التجارية ، يمكن أن تساعد واجهات برمجة التطبيقات المؤسسات التي تتطلع إلى الابتكار باستخدام نماذج أعمال تتجاوز كفاءاتها وقدراتها الأساسية

للتطبيقات والخدمات بعرض وظائفها وإمكانياتها خارج (API) تسمح واجهات برمجة التطبيقات فريقها لتطبيقات وخدمات أخرى ، داخل وخارج مؤسساتها. يُشار أحياناً إلى واجهات برمجة التطبيقات أيضاً على هذا النحو - واجهات برمجة التطبيقات □□□□□□□□ هي تلك التي يتم رفعها داخلياً داخل المؤسسة ، ويتم تعزيز واجهات برمجة التطبيقات □□□□□□□□ خارجياً. تختلف متطلبات الأمان والإدارة والقياس ، بالطبع ، بناءً على الجمهور المستهدف المقصود من واجهة برمجة التطبيقات.

يجب أن تكون واجهات برمجة التطبيقات الآلية الافتراضية التي من خلالها تتواصل التطبيقات والخدمات مع بعضها البعض. إذا قمت بتطوير بنية قائمة على الخدمات المصغرة ، فهذا هو الوضع الافتراضي. في التطبيقات القديمة ، قد تكون هناك سيناريوهات حيث توجد عمليات تكامل مخصصة من نقطة إلى نقطة بين التطبيقات

يجب استبدالها بواجهات برمجة تطبيقات محددة جيداً

واجهات برمجة التطبيقات ليست جديدة. كان هذا المفهوم موجوداً منذ فترة طويلة ، لا سيما أنه في الواقع ، غالباً ما يكون هناك جدل (SOA) اكتسب قوة جذب مع شعبية العمارة الموجهة للخدمة في أنقى صورهما ، يوفر كلاهما آلية لفصل التطبيقات إلى خدمات. APIs و SOA حول الاختلافات بين والتي لها واجهة معمارية محددة جيداً للتواصل والتواصل مع الخدمات الأخرى. بحكم التعريف ، هذه ، الواجهة المعمارية هي واجهة برمجة تطبيقات. يتمثل الاختلاف الرئيسي بين ما يشار إليه اليوم بواجهة مرتبطة (API) التقليدية في أن واجهات برمجة التطبيقات SOA وواجهات (API) برمجة التطبيقات يتيح ذلك لواجهات برمجة SOAP و XML بـ SOA بينما ترتبط ، REST / JSON بواجهات SOA. أن تكون خفيفة الوزن للغاية ومرنة وسهلة الإدارة والاستهلاك ، مقارنةً بـ (API) التطبيقات

244 مسرحيات لقيادة الابتكار devops الفصل 5

أصبحت واجهات برمجة التطبيقات آلية لدفع الابتكار من خلال السماح للمطورين بإنشاء تطبيقات جديدة تستفيد من واجهات برمجة التطبيقات الداخلية والخارجية لإضافة المزيد من الخدمات إلى تطبيقاتهم. يفعلون ذلك بطريقتين:

■ يمكن للمؤسسات القيام بذلك عن طريق تطوير تطبيقات جديدة ومبتكرة تستفيد من واجهات برمجة التطبيقات من خدمات الجهات الخارجية الخارجية. لماذا تُنشئ خدمة الخرائط الخاصة بك إذا كان هناك واجهة برمجة تطبيقات عامة تابعة لجهة خارجية تقدم الخدمة؟ لماذا يتم إنشاء خدمة إدارة هوية المستخدم إذا قام فريق آخر في المؤسسة بالفعل ببناء واحدة وعرضها عبر واجهة برمجة التطبيقات؟ الرياضية ، والتي تسمح ESPN يوضح الشكل 5-14 واجهات برمجة التطبيقات العامة لشبكة لأي مطور بتضمين البيانات الرياضية في تطبيقاته

■ يمكن للمؤسسات الآن استثمار قدراتها التجارية الخاصة بما يتجاوز قدرات تحقيق الدخل لنماذج الأعمال الخاصة بها من خلال تعريضها لنظام بيئي أوسع عبر واجهات برمجة التطبيقات. إذا طور البنك خدمة تحسب اليونانيين" بكفاءة ودقة أو ملف تعريف مخاطر تجارة المشتقات ، فلماذا لا تسهيلها من" خلال السماح للشركاء الخارجيين باستهلاك تلك الخدمة ودفع ثمنها عبر واجهة برمجة التطبيقات؟

ESPN API EXPLORER

All roads lead to apps

Explore the ESPN API by making live calls.

API Key:

Toggle All Endpoints

Sports API

Sports /sports/sportName/leagueName/

Sports and leagues supported in the ESPN API.

PARAMETER	VALUE	TYPE	DESCRIPTION
:sportName	All Sports	fixed	Sport name
:leagueName	All Leagues	string	League / organizing body within the sport.
_accept	application/json	string	Payload output format

Try It!

Headlines API

Athletes API

Teams API

ESPN Now API

Standings API

Research Notes API

Medals API

Audio API

Video API

Score & Schedules API

Apls أنتمة النشر و

فإن واجهات برمجة التطبيقات هي نفسها الخدمة. تحتاج المنظمات إلى التعامل ، DevOps من منظور مع واجهات برمجة التطبيقات كمنتج. هذا ينطبق بشكل خاص على واجهات برمجة التطبيقات الخارجية" ، سواء تم تحقيق الدخل منها أم لا. حتى داخلياً ، على الرغم من ذلك ، فإن واجهة برمجة التطبيقات هي □□□□ بين الموفر والمستهلك ويجب معاملتها على هذا النحو. لأغراض تسليم التطبيق هو توفير الخدمة التي تعرضها واجهة برمجة التطبيقات (API) فإن تقديم واجهة برمجة التطبيقات ، ومع ذلك ، فإن دورة حياة الخدمة تتضمن الآن أيضاً الخطوات التي تتناول واجهات برمجة التطبيقات : على سبيل المثال

لا يكفي □□□□□□ وظائف وأداء وأمان التطبيق أو الخدمات. عليك أن تفعل الشيء نفسه ■ مع واجهات برمجة التطبيقات ، والاستفادة من حالات الاستخدام الخاصة بكيفية استهلاك واجهات برمجة التطبيقات.

عادةً ما تتضمن إدارة □□□□□□□□ للتطبيق والخدمات أيضاً إدارة تكوين البرامج الوسيطة ■ يتم استخدامه API وأي برنامج إدارة API المرتبطة. الآن ، يجب أيضاً معالجة إدارة تكوين بالنسبة □□□□□□□□ □□□□□□ فإن إصدار القدرة التجارية هو إصدار لجميع ■ التطبيقات والخدمات التي تدخل في تقديم القدرة التجارية. مع تضمين واجهات برمجة التطبيقات ، سيحتاج الإصدار أيضاً إلى تضمين جميع واجهات برمجة التطبيقات الخارجية التي تستهلكها التطبيقات والخدمات التي يتم إصدارها. أصبحت إدارة اتفاقيات مستوى الخدمة لواجهات برمجة التطبيقات هذه مصدر قلق لفريق إدارة الإصدار

لقد ناقشت بالفعل اقتراح القيمة الأساسية لواجهات برمجة التطبيقات للتسليم المستمر بشكل مطول بفصل (API) في القسم الخاص بهندسة الخدمات المصغرة. تسمح واجهات برمجة التطبيقات التطبيقات والخدمات. وبالتالي يمكن بناؤها ونشرها وتشغيلها بشكل مستقل عن بعضها البعض ، دون القلق بشأن موقع نشر الخدمات الأخرى التي يستهلكونها. كل ما يهم هو توافر واجهات برمجة التطبيقات. يمكن أيضاً توسيع نطاق التطبيقات والخدمات من خلال نشر المزيد من مثيلات الخدمة التي تحتاج إلى توسيع نطاقها ، دون الحاجة إلى توسيع نطاق الخدمات الأخرى التي لا تتأثر بحجم الخدمة بالتسليم المستمر الحقيقي ، من (API) المطلوبة للتوسع. وبالتالي تسمح واجهات برمجة التطبيقات خلال السماح بتسليم دفعات صغيرة من التغيير ، مع أوقات دورات تغذية مرتدة سريعة

Apls و devops منصة

نفسها يمكن الوصول إليها DevOps يعد دور واجهات برمجة التطبيقات عنصراً مهماً أيضاً لأن منصة دعونا CaaS أو PaaS أو IaaS أيضاً عبر واجهات برمجة التطبيقات ، سواء تم تنفيذها باستخدام لنقي نظرة على دور واجهات برمجة التطبيقات لكل من هذه المتغيرات

IaaS فسيتم عرض جميع خدمات ، IaaS إذا تم تنفيذ النظام الأساسي بالاستفادة من سحابة ■ على أنها واجهات برمجة تطبيقات. يتم استهلاك جميع هذه الخدمات والموارد التي تقدمها بواسطة التطبيقات التي يتم تسليمها وتشغيلها على النظام الأساسي عبر واجهات برمجة

246 مسرحيات لقيادة الابتكار devops الفصل 5

التطبيقات. في السيناريو الذي وصفته سابقاً في قسم "تنسيق السحابة" لتوفير المكدس الكامل عبر العديد من السحب باستخدام أنماط الحرارة المحمولة ، والاستفادة من واجهات برمجة على السحب المتعددة. يوضح الشكل 15-5 HEAT لتوفير أنماط OpenStack تطبيقات OpenStack. مثلاً لواجهة برمجة تطبيقات الشبكات المتاحة لـ

POST

/v2.0/networks

Create network

Creates a network.

A request body is optional. An administrative user can specify another tenant UUID, which is the tenant who owns the network, in the request body.

Error response codes: 201,401,400

Request

Request Example

```
{
  "network": {
    "name": "sample_network",
    "admin_state_up": true
  }
}
```

Response Parameters

Name	In	Type	Description
status	body	string	The network status.
router:external (Optional)	body	boolean	Indicates whether this network is externally accessible.
availability_zone_hints	body	array	The availability zone candidate for the network.
availability_zones	body	array	The availability zone for the network.
name	body	string	Human-readable name of the resource.
admin_state_up	body	boolean	The administrative state of the resource, which is up (<code>true</code>) or down (<code>false</code>).
tenant_id	body	string	The ID of the tenant who owns the resource.
mtu	body	integer	The MTU of a network resource.
qos_policy_id (Optional)	body	string	The UUID of the QoS policy.
subnets	body	array	The associated subnets.
shared (Optional)	body	boolean	Admin-only. Indicates whether this network is shared across all tenants.
id	body	string	The UUID of the network.
network	body	object	A <code>network</code> object.

الشكل 15-5: Openstack Networking Api (openstack.org ، 2016)

■ فسيصبح الوصول إلى الأدوات أيضاً ، CaaS أو PaaS إذا تم تنفيذ النظام الأساسي باستخدام DevOps من خلال الاستفادة من واجهات برمجة التطبيقات. أدوات أو في الحاويات. يتم توصيل هذه الخدمات وتنظيمها لتنفيذ مسار PaaS هي خدمات تعمل على تسليم التطبيق باستخدام واجهات برمجة التطبيقات الخاصة بهم.

التنظيم للابتكار play:

تطوير الإيجابيات

جمهورية الدومينيكان ونيكاراغوا متشابهة تقريباً من حيث عدد السكان وتشارك في شغف عميق بالبيسبول. لكن جمهورية الدومينيكان أرسلت أكثر من 100 لاعب إلى فرق دوري البيسبول الرئيسي في حين أن نيكاراغوا لا تضم سوى ثلاثة لاعبين كبار ، (MLB) تتطلع نيكاراغوا إلى تغيير هذا التفاوت. وربما تكون الخطوة الأولى في تحقيق ذلك قد جاءت في وقت مبكر من هذا العام بافتتاح أكاديمية نيكاراغوا للبيسبول. يساعد دينيس مارتينيز ، الذي أصبح في عام 1976 أول نيكاراغوا يصل إلى الشركات الكبرى ، في قيادة هذا الجهد لتدريب الدوريين الطموحين بالمهارات اللازمة لجذب انتباه الكشافة.

لن يؤدي النجاح إلى تعزيز جيل جديد من اللاعبين في نيكاراغوا فحسب ، بل سيضخ بعض السيولة المطلوبة في اقتصاد البلاد - إذا حكمنا من خلال مثال جمهورية الدومينيكان في جمهورية الدومينيكان ، يعتبر الرماة ، والضاربون ، واللاعبون في الميدان من سلع التصدير الرئيسية ، ويكسبون عملات أجنبية جادة للبلاد وتحويلات ضخمة لعائلات الرياضيين. شهرة نجوم الدومينيكان مثل ديفيد أورتيز وروبنسون كانو وسامي سوسا (قبل فضيحة المنشطات) ملأت ملاعب وتساعد أيضاً في تعزيز السياحة - وهي صناعة نمو رئيسية - إلى وطنهم MLB بنفس القدر من الأهمية ، كانت لعبة البيسبول هي السبيل للخروج من الفقر لآلاف من شباب الدومينيكان - وهي بديل صحي ومريح للمهن الخطرة في العصابات والجريمة.

يراهن مارتينيز من نيكاراغوا ، الذي أصبح في عام 1991 الرامي الوحيد المولود في أمريكا اللاتينية الذي قدم مباراة مثالية ، على أن شباب نيكاراغوا - مثل نظرائهم الدومينيكيين - لديهم كان التحدي يتمثل في تطوير نموذج الأعمال المناسب. MLB الموهبة والطموح للنجاح في فاينبرج ، 2011—

إلا عندما يكون لدى اللاعب *Moneyball* كيف تحدد وتطور اللاعبين المحترفين؟ لا يعمل نموذج بالفعل سجل حافل إحصائياً ليبنى عليه قرار الاختيار. كيف تتعامل مع الآلاف من اللاعبين الشباب الذين يسعون جاهدين للنجومية لتحديد الأفاق الناشئة والاستثمار فيها؟

هناك الآلاف من الشركات الناشئة ، Apple و Microsoft وحتى قبلهما PayPal و Facebook التي فشلت. لم يتجاوز البعض تشكيل الفريق المؤسس ، في حين غادر البعض الآخر عن طريق الاكتتاب العام ، أو عن طريق الاستحواذ ، ليموت على الرغم من التوقعات. أنت ، بالطبع ، لا تسمع عن كل من لم ينجح.

Google ، داخل المنظمات الكبيرة ، الوضع أسوأ. هناك عدد قليل جداً من الشركات الكبيرة مثل التي لم تفقد ثقافة الابتكار لديها مطلقاً ولديها العديد من الآليات المعمول بها لتشجيع الأفكار الجديدة معظم المؤسسات الكبيرة ليست مصممة للابتكار. ليس لديهم ثقافة (He ، 2013) من موظفيها الابتكار. في الواقع ، معظمهم لديهم عكس ذلك تماماً. يتم خلق الموظفين من خلال إجراءات الحوكمة الصارمة وإدارة الأداء التي تشجعهم على القيام بعملهم المحدد ، والبقاء "في الصندوق" ، والقيام بعملهم بأفضل ما لديهم ، ولا يتم تشجيع الأفكار الجديدة. هذا ، بالطبع ، ليس نموذج عمل قابل للتطبيق

ومنصات تقنية فعالة مصممة DevOps لعالم اليوم التنافسي والمتغير بسرعة. إن امتلاك أفضل منصة للابتكار ليست ذات قيمة إذا لم تكن هناك أفكار مبتكرة يتم تطويرها وتجربتها

000000 000000 000 000.00000 00 00000000 00
 00000 0 00000 00000 00000 0000000 00 000000
 0000000000 00000000 00000 00 0000000 000000 00.

Digitized by Google

تعد هذه الثقافة أمراً بالغ الأهمية ، ويتفوق إنسانها على جميع ، DevOps من منظور اعتماد التحسينات التقنية والعملية التي يمكن أن تتبناها فرق تكنولوجيا المعلومات. تم تصميم جميع الموضوعات والمسرحيات التي تمت مناقشتها في هذا الفصل لتمكين فرق إيريك شميدت من التحدث عنها. هذه هي الفرق التي تبتكر الأفكار المبتكرة وتعمل بجد لتجربتها لمعرفة ما إذا كانت هناك فرصة عمل موجودة وما إذا كانت الفكرة المبتكرة قابلة للتطبيق. كل هذا محل نقاش إذا كانت الثقافة في المنظمة لا تسمح حتى للفرق بالعمل على الفكرة بالحرية والموارد التي يحتاجونها ، إذا كان الأشخاص من حولهم والمديرين التنفيذيين الذين ذكرهم إريك لا يملكون ثقافة تسمح للأفكار تنمو وتزدهر وتفشل دون عواقب. ليس من السهل تأسيس هذه الثقافة في منظمة كبيرة ذات □□□□□□ □□□□□□ ، لكنها ضرورية. غالباً ما تكون المؤسسات الكبيرة قادرة على القيام بذلك في فرق صغيرة قادرة على التحليق تحت الرادار ولكنها تعمل فقط للمشاريع الصغيرة بمعزل عن غيرها. يتطلب تحقيق ثقافة الابتكار على مستوى المؤسسة القدرة على الابتكار عبر المنظمة ، على نطاق المؤسسة

تطوير ثقافة الابتكار بشكل عام المنظمات

□□□□ □□□□ □□ □□□□ □ □□□□□□□ □□ □□□□□□ □□□□□ □
□□□□□□□□ □□□□ □□□□ □□□ □□□□□□□□ □□□□□□ □□□□□□
□□□□□□□□ □□□□□□ □□□□□□□□ □□□□□□□□ □□□□□□□□ □□□□□□□□

لورنس جاكسون ، لاعب كرة قدم أمريكي سابق—

قدمت شركة ماكينزي وشركتها أربعة مناهج مثبتة حول كيفية إنشاء ثقافة الابتكار هذه وتوسيع نطاقها. يتحدثون في سياق (Edelman ، 2015): □□□□□□ □□□□□□□□

249 مسرحيات لقيادة الابتكار devops الفصل 5

1. □□□□□□□□ □□□□□□□□ . تضع المنظمة قادة الابتكار مسؤولين عن المنظمة بأكملها ، وليس فقط قيادة الابتكار. يتيح ذلك للمؤسسة تطوير ثقافة □□□□□□ □□□□□□ ، والتي تسرع أيضاً من تحسين الأنظمة القديمة.
2. □□□□□□ □□□□□□□□□□ . يعد هذا خياراً أكثر عدوانية حيث يتولى قادة الابتكار التطبيقات والأنظمة القديمة ويحولون عملياتهم وتقنياتهم وفرقهم لتكون مماثلة لتلك الخاصة بفرق الابتكار ، مما يتيح عمليات الابتكار والتكنولوجيا والثقافة على مستوى المؤسسة.
3. هذا هو نهج الشرائح والنرد حيث يتم تجميع الفرق التي تركز على الابتكار في قسم . *Spinoff* منفصل ومستقل للسماح لهم جميعاً بالنمو والازدهار وتطوير العمليات الصحيحة ومنصات تكنولوجيا المعلومات والثقافة الخاصة بهم ، دون أن يتقلمهم بقية الجمود الثقافي للمنظمة. في بعض الحالات القصوى ، تم أيضاً نسج هذه الأقسام كشركات منفصلة إما مستقلة أو كشركة تابعة.
4. □□□□□□ □□□□ . هذا هو نهج الشراكة حيث تتعاون المنظمة مع شركة أخرى لديها المهارات الضرورية التي تركز على الابتكار والمعرفة والثقافة. يبدأ الشريك بعد ذلك في تقديم أنظمة الابتكار لاستكمال الأنظمة القديمة من المؤسسة الأساسية ، دون حاجة المنظمة إلى تحويل نفسها.

من أجل التحسين والابتكار ، في الفصل التالي ، DevOps سأتناقش نماذج فريق محددة لتبني ثقافة

ملخص

التي تمثل أنماطاً للنجاح التي شوهدت عند اعتماد DevOps ركزت في هذا الفصل على مسرحيات للمشاريع والبرامج التي تهدف إلى الابتكار. من المهم ملاحظة أنه بعيداً عن الشركات DevOps ، الناشئة ، نادراً ما يكون لدى معظم المؤسسات مشاريع مبتكرة تعمل بمعزل عن غيرها. ومن ثم سيعتمد كل منتج مبتكر على التطبيقات والخدمات الحالية التي ليست مبتكرة بطبيعتها ولكنها تقدم في المؤسسات الكبيرة سيشمل دائماً Real DevOps خدمات الأعمال الأساسية. ومن ثم ، فإن اعتماد كل من عمليات التحسين والابتكار المسرحيات الأربع التي قدمتها في هذا الفصل هي:

- DevOps إنشاء منصة
- تقديم بنى الخدمات المصغرة
- تطوير اقتصاد API
- التنظيم للابتكار

: □□□□□□□□ الأربعة التي تم إبرازها في بداية هذا الفصل

- تحقيق تقنية المعلومات متعددة السرعات
- بناء الشيء الصحيح
- تمكين التجريب
- توصيل أنظمة مضادة للكسر

