Université Constantine 2
جامعة قسنطينة 2

حوسبة سحابية

–الدورة – 3

الفصل الثالث: الحاويات

دكتور منور ر.
كلية التقنيات الجديدة
rostom.mennour@univ-constantine2.dz

الطلاب المعنيين

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

اهداف الدورة

- وصف التاريخ والتكنولوجيا والمصطلحات المستخدمة في الحاويات
- التمييز بين الحاويات و
  الخوادم المعدنية والأجهزة الافتراضية (VMs)
- توضيح مكونات دوكر
  وتفاعلهم
- التعرف على خصائص بنية الخدمات المصغرة
- فهم استخدام K8s وتنسيق الحاويات

خريطة الدورة

- عصر الخدمات المصغرة
- عصر DevOps • الحاويات
- الحاويات مقابل المحاكاة الافتراضية
- عامل الميناء
- التنسيق مع K8s
- خاتمة

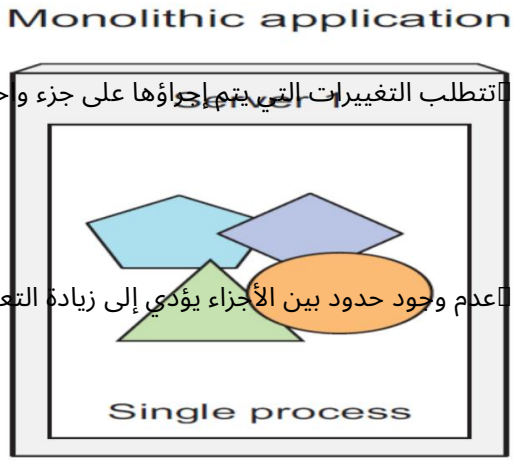
القسم :1
عصر الخدمات المصغرة

الانتقال من التطبيقات المتجانسة إلى الخدمات الصغيرة

- تطبيقات متجانسة

تعريف

تتكون التطبيقات المتجانسة من مكونات ترتبط جميعها ارتباطًا وثيقًا ويجب تطويرها ونشرها وإدارتها ككيان واحد لأنها تعمل جميعها كعملية نظام تشغيل واحدة.



- تتطلب التغييرات التي يتم إجراؤها على جزء واحد من التطبيق إعادة نشر التطبيق بأكمله.
- عدم وجود حدود بين الأجزاء يؤدي إلى زيادة التعقيد وما يترتب على ذلك من تدهور في جودة النظام بأكمله.

الانتقال من التطبيقات المتجانسة إلى الخدمات الصغيرة

- نطاق التطبيقات المتجانسة

- يتطلب تشغيل تطبيق متجانس عادةً عددًا صغيرًا من الخوادم القوية التي يمكنها توفير موارد كافية لتشغيل التطبيق.

- للتعامل مع الأحمال المتزايدة على النظام، يجب عليك إما توسيع نطاق الخوادم عموديًا (Scaling up)أو توسيع نطاق النظام بأكمله أفقيًا • .(Scaling out)على الرغم من أن القياس الرأسي لا يتطلب عمومًا إجراء تغييرات في التطبيق، إلا أنه يصبح مكلفًا بسرعة، وفي الممارسة العملية، دائمًا ما يكون له حد أعلى.

- يعد القياس الأفقي غير مكلف نسبيًا من منظور الأجهزة، ولكنه قد يتطلب تغييرات كبيرة في كود التطبيق وهو ليس ممكنًا دائمًا.

- إذا كان جزء من تطبيق متجانس غير قابل للتوسع، فالتطبيق
يصبح كاملًا غير قابل للتطوير (غير قابل للتطوير).

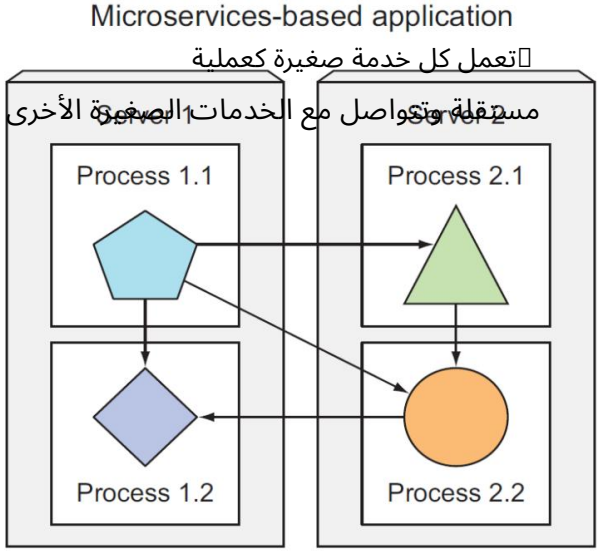
الانتقال من التطبيقات المتجانسة إلى الخدمات الصغيرة

- تقسيم التطبيقات المتجانسة إلى خدمات صغيرة

تعريف

تعريف بنية الخدمات الصغيرة بسيط: تطبيق

تتألف من الخدمات "الصغرى" الفردية. الخدمة، ضمن بنية الخدمات الصغيرة ، هي وحدة مستقلة تقدم وظائف كاملة .



- تعمل كل خدمة صغيرة كعملية مستقلة وتتواصل مع الخدمات الصغيرة الأخرى عبر واجهات برمجة التطبيقات البسيطة.

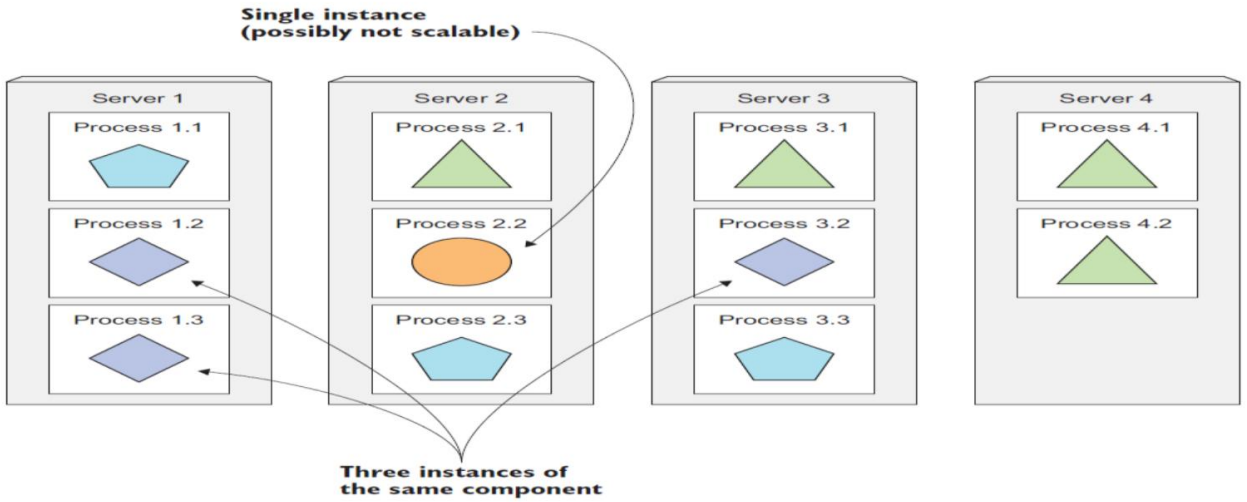
- من الممكن تطوير ونشر كل منهما الخدمات المصغرة بشكل منفصل.
  - يمكن كتابة كل خدمة صغيرة باللغة أكثر ملاءمة لتنفيذه.
  - تغيير أحدهما لا يتطلب أي شيء تعديل أو إعادة نشر أي خدمة أخرى.

الانتقال من التطبيقات المتجانسة إلى الخدمات الصغيرة

- توسيع نطاق الخدمات الصغيرة

- يتم توسيع نطاق الخدمات الصغيرة على أساس كل خدمة على حدة، مما يسمح لك بتوسيع نطاق الخدمات التي تتطلب المزيد من الموارد فقط ، مع ترك الخدمات الأخرى على نطاقها الأصلي.

- يتيح لك تقسيم التطبيق إلى خدمات صغيرة وضع الأجزاء التي تسمح به أفقيًا، والأجزاء التي لا تسمح به، رأسيًا وليس أفقيًا.



الانتقال من التطبيقات المتجانسة إلى الخدمات الصغيرة

- نشر الخدمات الصغيرة

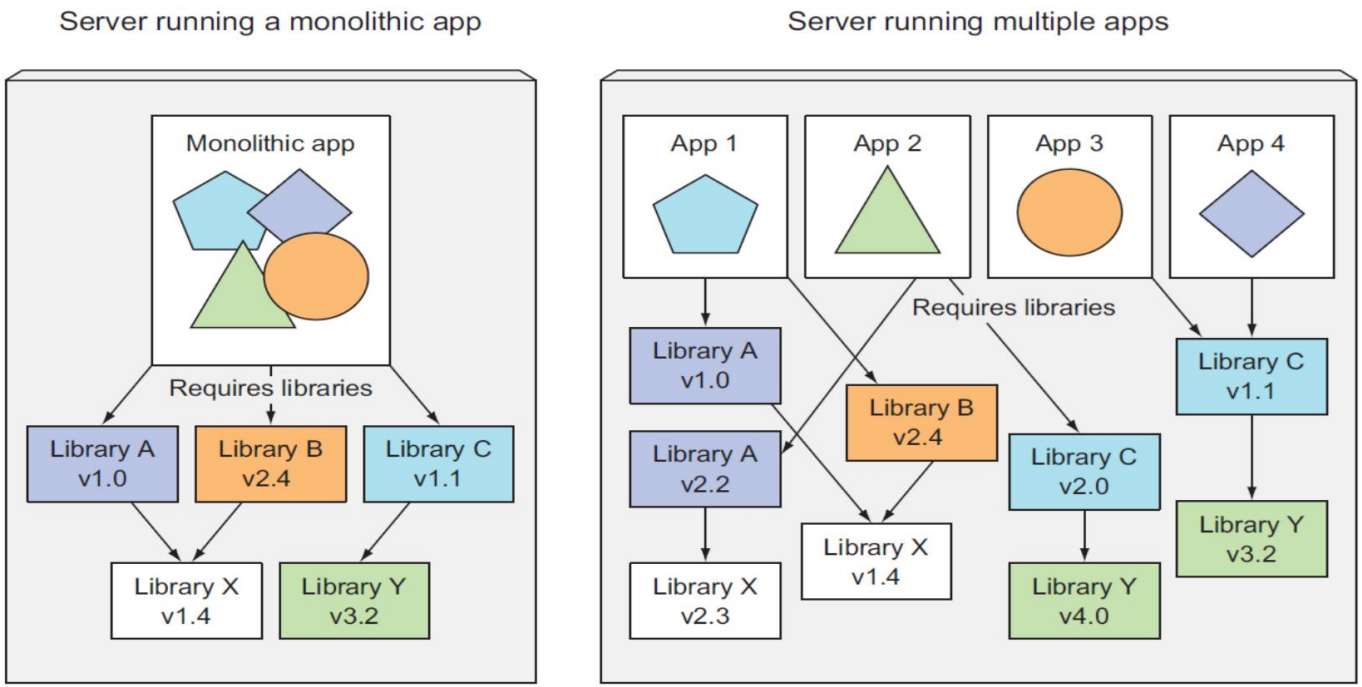
تعليق

مع زيادة عدد المكونات، تصبح قرارات النشر صعبة بشكل متزايد لأنه لا يزيد عدد مجموعات النشر فحسب، بل يزيد أيضًا عدد الترابط بين المكونات.

تعمل الخدمات المصغرة ضمن فرق، لذا يجب أن يجتمعوا معًا ويتحدثوا مع بعضهم البعض.

أثناء النشر، يجب على شخص ما أو شيء ما تكوينها بشكل صحيح للسماح لها بالعمل معًا كنظام واحد. مع تزايد عدد الخدمات الصغيرة، يصبح هذا أمرًا مملاً وعرضة للأخطاء، خاصة عندما تفكر في ما يتعين على فرق العمليات/مسؤول النظام القيام به في حالة تعطل الخادم.

الانتقال من التطبيقات المتجانسة إلى الخدمات الصغيرة



- يمكن أن يصبح نشر التطبيقات المرتبطة ديناميكيًا والتي تتطلب إصدارات مختلفة من المكتبات المشتركة و/أو مواصفات البيئة الأخرى كابوسًا سريعًا لفريق العمليات الذي ينشرها ويديرها على خوادم الإنتاج.
- كلما زاد عدد المكونات التي سيتم نشرها على نفس المضيف، أصبح الأمر أكثر صعوبة

إدارة جميع تبعياتهم لتلبية جميع متطلباتهم.

القسم :2
حاويات

فهم الحاويات

ما هي الحاوية؟

فهم الحاويات

Before shipping containers

- Goods were shipped in a variety of vessels with no standardized weight, shape, or size.
- Transporting goods was slow, inefficient, and costly.

After shipping containers

- Uniformly sized shipping containers simplified loading, unloading, storing, and transferring between transport types.
- Abstraction of shipment details improved efficiency, increased productivity, and reduced costs.

التجريد

فهم الحاويات

•عندما يتكون التطبيق من عدد صغير من المكونات الكبيرة، فمن المقبول تمامًا تعيين جهاز ظاهري مخصص (VM)لكل مكون وعزل بيئاته من خلال تزويد كل منها بمثيل نظام التشغيل الخاص به.

•عندما تكون هذه المكونات صغيرة ويبدأ عددها في التزايد، لا يمكنك تعيين جهاز افتراضي خاص بها إذا كنت لا تريد إهدار موارد الأجهزة والحفاظ على انخفاض تكاليف الأجهزة.

•ولكن الأمر لا يتعلق فقط بإهدار موارد الأجهزة. نظرًا لأن كل جهاز افتراضي يحتاج عادةً إلى تكوينه وإدارته بشكل فردي، فإن العدد المتزايد من أجهزة الكمبيوتر يؤدي أيضًا إلى إهدار الموارد البشرية، حيث يؤدي ذلك إلى زيادة عبء العمل على مسؤولي النظام بشكل كبير.

فهم الحاويات

•تسمح لك الحاويات بتشغيل خدمات متعددة على نفس الكمبيوتر المضيف، مع عدم الكشف عن بيئة مختلفة لكل منها فحسب، بل أيضًا عزلها عن بعضها البعض، على غرار الأجهزة الافتراضية، ولكن بحمل أقل بكثير.

•عندما تكون هذه المكونات صغيرة ويبدأ عددها في التزايد، لا يمكنك تعيين جهاز افتراضي خاص بها إذا كنت لا تريد إهدار موارد الأجهزة والحفاظ على انخفاض تكاليف الأجهزة.

•العملية التي يتم تنفيذها في حاوية يتم تشغيلها في نظام التشغيل الخاص بالمضيف، مثل كافة العمليات الأخرى. لكن العملية في الحاوية لا تزال معزولة عن العمليات الأخرى.

•بالنسبة للعملية نفسها، يبدو أنها العملية الوحيدة التي تعمل على الجهاز وفي نظام التشغيل الخاص به.

ملحوظة (مهم)

السمة الرئيسية للحاوية هي العزل.

حاويات

الحاويات مقابل المحاكاة الافتراضية

النقل بالحاويات

الحاويات خفيفة

عدد أكبر من الحالات

يعمل على نظام التشغيل المضيف

تشغيل تطبيق واحد

تشغيل التطبيقات مباشرة

العزل الجزئي

الافتراضية

المركبات المتوسطة ثقيلة

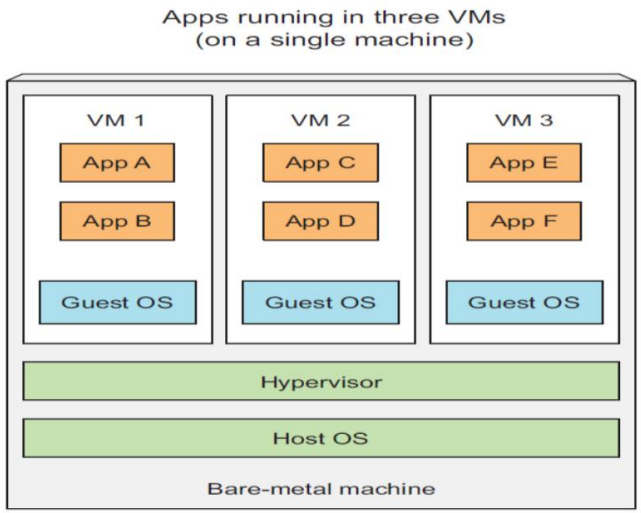
انخفاض عدد الحالات

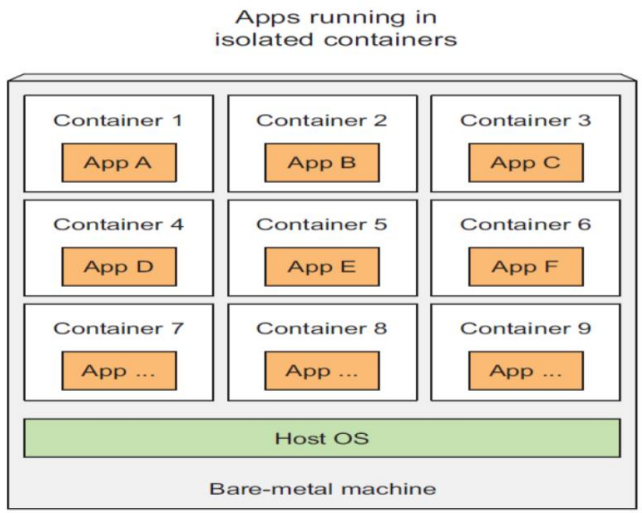
عمليات الإطلاق هي نظام تشغيل نظيف

مجموعة تطبيقات متعددة

تمر التطبيقات عبر برنامج Hypervisor

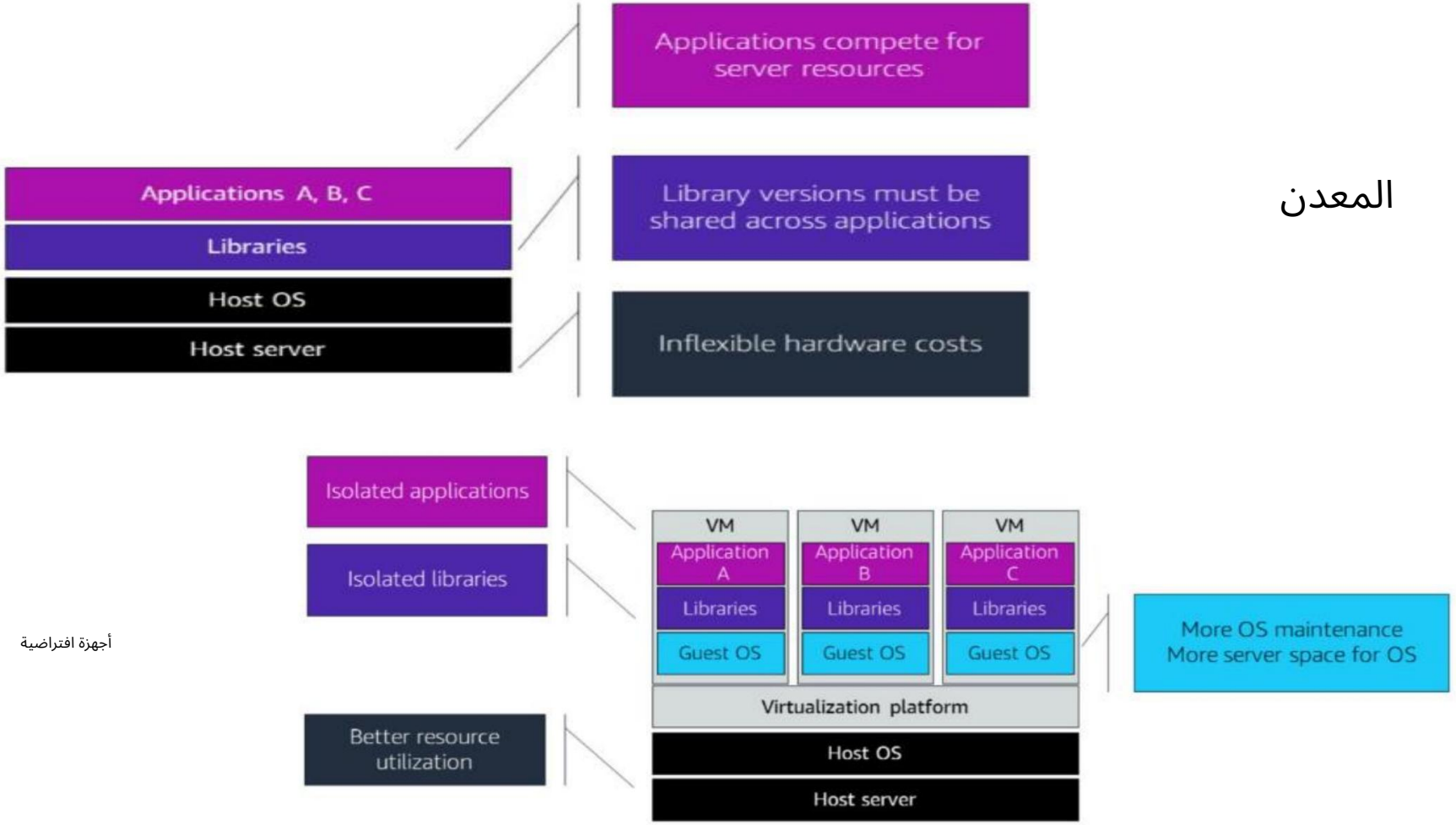
عزل كامل





حاويات

تطور النشر

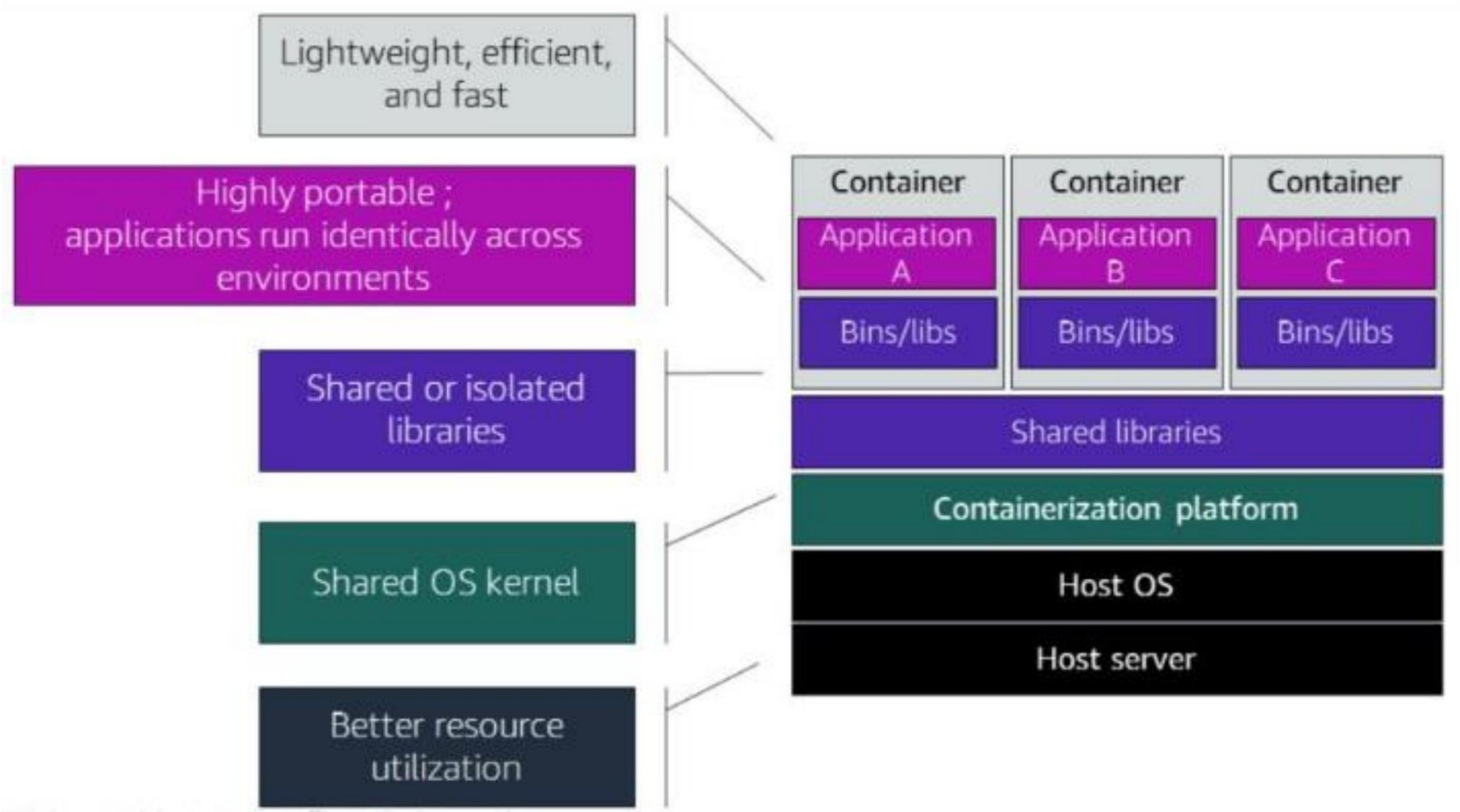


المعدن

أجهزة افتراضية

حاويات

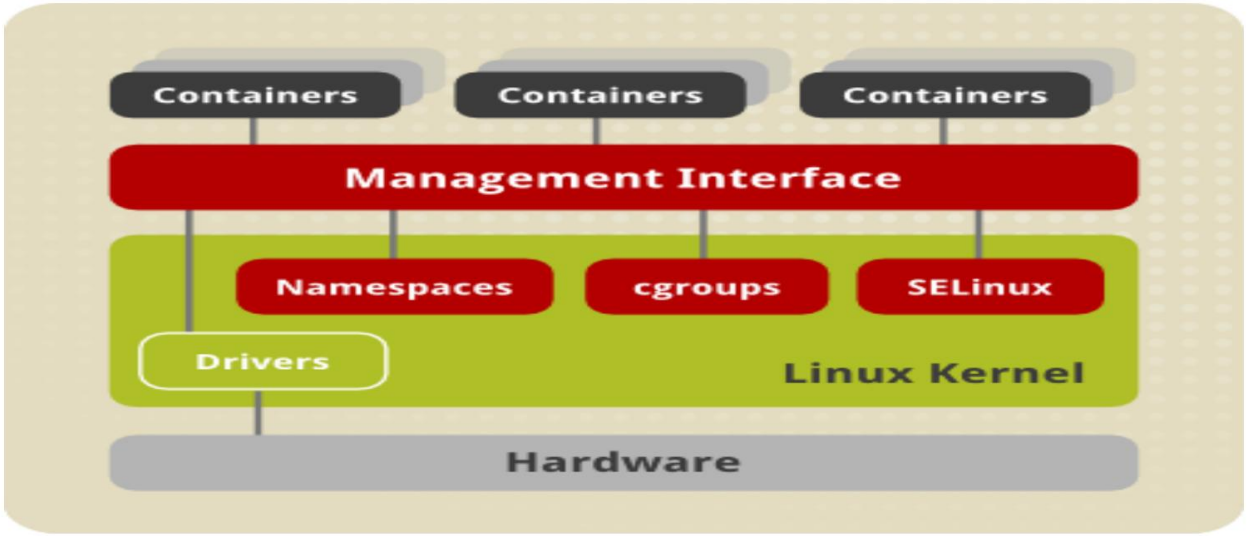
تطور النشر



عزل المكونات في الحاويات

كيف يمكن للحاويات عزل العمليات بالضبط إذا كانت تعمل على نفس نظام التشغيل؟

- مساحات أسماء Linux: تضمن أن كل عملية ترى رؤيتها الشخصية للنظام (الملفات والعمليات وواجهات الشبكة واسم المضيف وما إلى ذلك).
- مجموعات التحكم في Linux (cgroups): تحد من كمية الموارد التي تحتاجها العملية يمكن أن تستهلك (وحدة المعالجة المركزية، الذاكرة، النطاق الترددي للشبكة، وما إلى ذلك).



عزل المكونات في الحاويات

- مساحات أسماء لينكس

- بشكل افتراضي، يحتوي كل نظام Linux في البداية على مساحة اسم واحدة. تنتمي جميع موارد النظام، مثل أنظمة الملفات ومعرفات العمليات ومعرفات المستخدم وواجهات الشبكة وغيرها، إلى مساحة الاسم الواحدة. • يمكنك إنشاء مساحات أسماء إضافية وتنظيم الموارد فيما بينها. عندما تقوم بتشغيل عملية ما، فإنك تقوم بتشغيلها في إحدى مساحات الأسماء هذه.

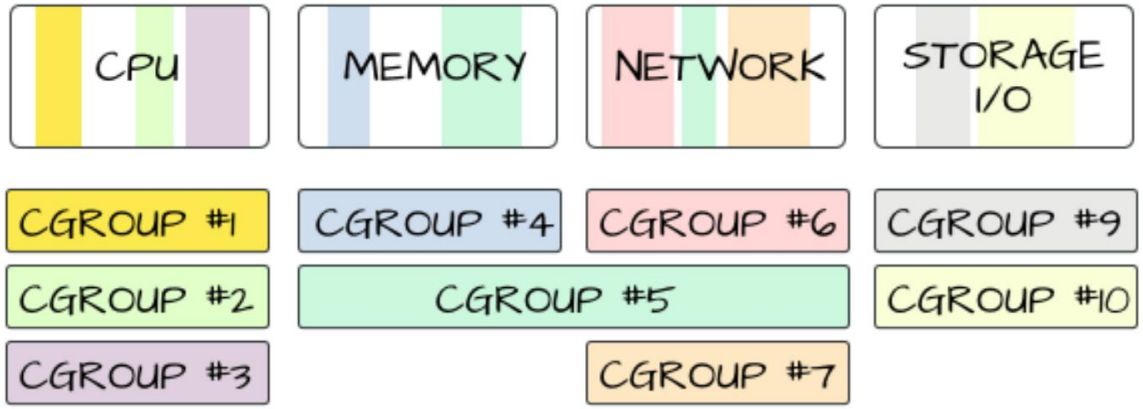
- هناك عدة أنواع من مساحات الأسماء، وبالتالي فإن العملية لا تنتمي إلى مساحة اسم واحدة، ولكن إلى مساحة اسم من كل نوع.

- جبل (مونت)
- معرف العملية (معرف المنتج)
- الشبكة (الصافي)
- الاتصال بين العمليات (IPC)
- UTS

عزل المكونات في الحاويات

- مجموعات التحكم في Linux

- يحد من كمية موارد النظام التي يمكن أن تستهلكها الحاوية. • لا يمكن أن تستخدم العملية أكثر من المقدار الذي تم تكوينه من وحدة المعالجة المركزية والذاكرة وعرض النطاق الترددي للشبكة وما إلى ذلك. • لا يمكن للعمليات تحويل الموارد المحجوزة لعمليات أخرى، وهو ما يشبه تشغيل كل عملية على جهاز منفصل.

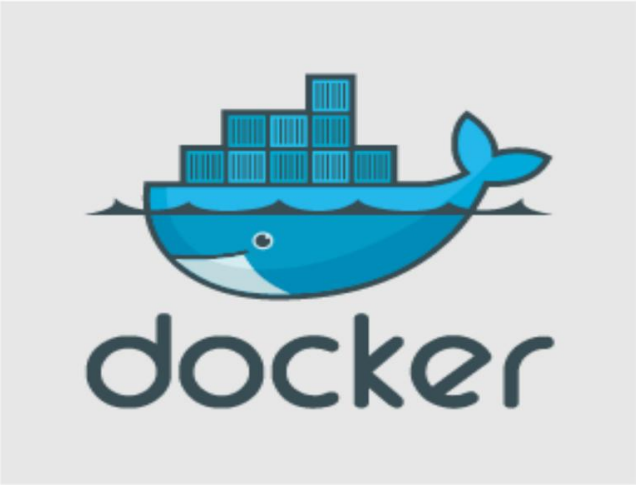


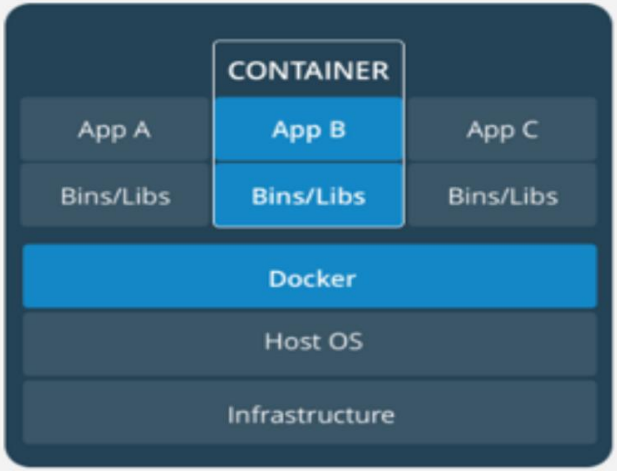
القسم :4
عامل ميناء

فهم مفاهيم دوكر

تعريف

Dockerعبارة عن منصة لتعبئة التطبيقات وتوزيعها وتشغيلها. يسمح لك بحزم تطبيقك مع بيئته بأكملها.





فهم مفاهيم دوكر

يتضمن Dockerثلاثة مفاهيم رئيسية:

•صورة حاوية Dockerهي شيء تقوم بتجميع تطبيقك وبيئته فيه. فهو يحتوي على نظام الملفات الذي سيكون متاحًا للتطبيق وبيانات التعريف الأخرى، مثل المسار إلى الملف القابل للتنفيذ ليتم تشغيله عند تشغيل الصورة.:روصلا

•سجل Dockerهو مستودع يخزن صور Dockerالخاصة بك ويسهل مشاركة تلك الصور بين مختلف الأشخاص وأجهزة الكمبيوتر. عندما تقوم بإنشاء صورتك، يمكنك تشغيلها على الكمبيوتر الذي أنشأتها عليه أو يمكنك دفع الصورة إلى السجل ثم تنزيلها (سحبها) إلى كمبيوتر آخر وتشغيلها في هذا الموقع. بعض السجلات عامة، مما يسمح لأي شخص باستخراج الصور منها، في حين أن البعض الآخر خاص ولا يمكن الوصول إليه إلا لأشخاص أو أجهزة معينة.:تالي‌جس‌تل‌ا

•حاوية Dockerهي حاوية Linuxقياسية تم إنشاؤها من صورة حاوية Docker. الحاوية قيد التشغيل هي عملية يتم تشغيلها على المضيف الذي يقوم بتشغيل Docker، ولكنها معزولة تمامًا عن المضيف وأي عمليات أخرى تعمل عليه. كما أن العملية محدودة الموارد أيضًا، مما يعني أنها لا يمكنها الوصول إلا إلى كمية الموارد المخصصة لها واستخدامها (وحدة المعالجة المركزية، وذاكرة الوصول العشوائي، وما إلى ذلك).:تاي‌واح‌ل‌ا

سمات

تلخص القائمة التالية بعض المزايا المهمة للحاويات
عامل ميناء:

- Docker هي بيئة تطبيقات محمولة لوقت التشغيل.

- كنكمي تجميع التطبيق وتبعياته في قطعة أثرية واحدة غير قابلة للتغيير تسمى صورة.

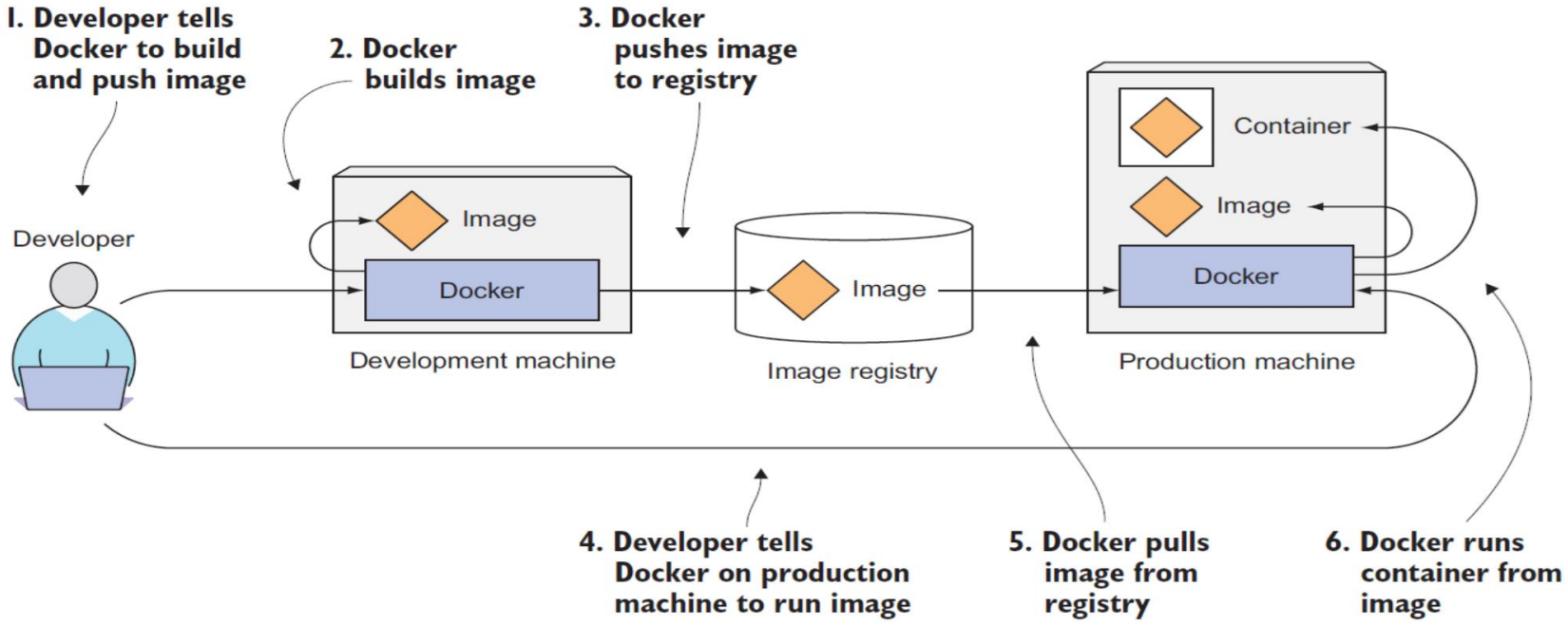
- دعب إنشاء صورة حاوية، يمكنها الانتقال إلى أي مكان يدعمه Docker.

- يمكنك تشغيل إصدارات مختلفة من التطبيقات بتبعيات مختلفة في وقت واحد.

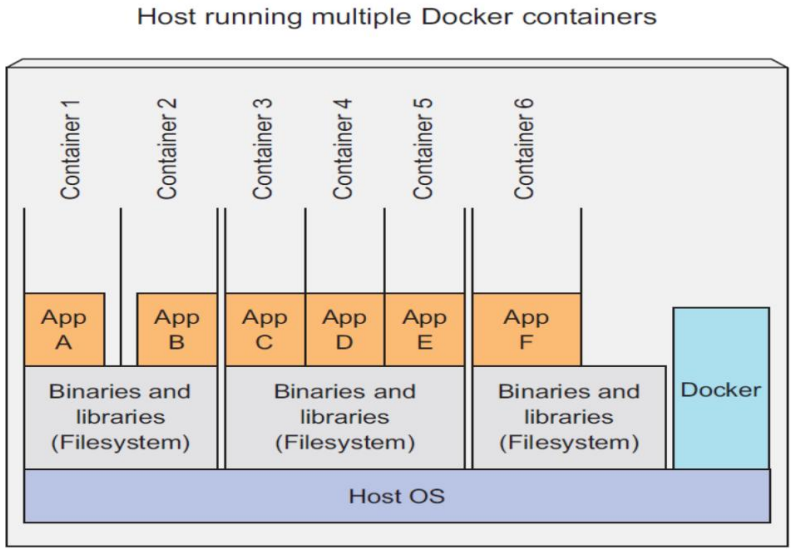
وتؤدي هذه الفوائد إلى دورات تطوير ونشر أسرع بكثير، فضلاً عن تحسين استخدام الموارد وكفاءتها. وترتبط كل هذه القدرات بالرشاقة.

بناء وتوزيع وتشغيل صورة Docker

- يقوم المطور أولاً بإنشاء صورة ثم يدفعها إلى السجل.
- يمكننا بعد ذلك استخراج الصورة على أي جهاز آخر يعمل بنظام Dockerو قم بتشغيل الصورة.
- يقوم Dockerبإنشاء حاوية معزولة بناءً على الصورة ويقوم بتشغيل الملف الثنائي المحدد القابل للتنفيذ كجزء من الصورة.

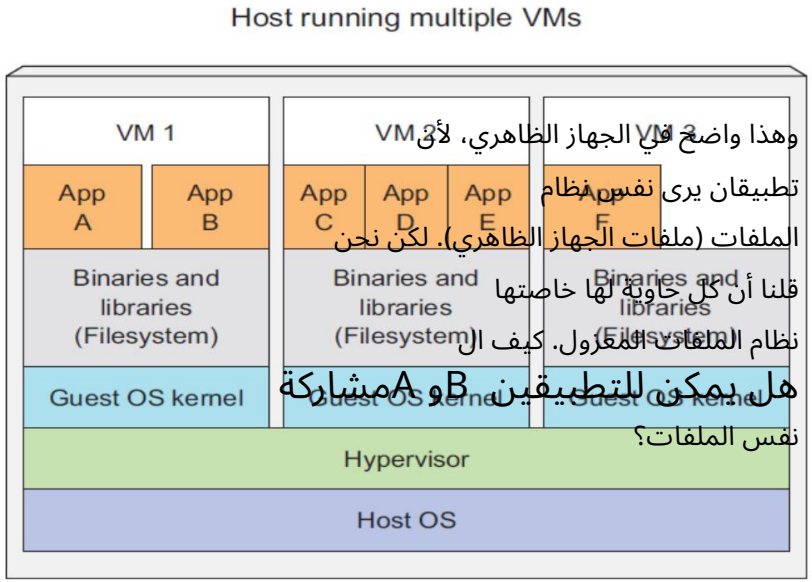


مقارنة بين صور عامل الإرساء وصور VM



نفس التطبيقات الستة التي تعمل في الأجهزة الافتراضية وكحاويات .Docker

ستلاحظ أن التطبيقات Bو Aلها حق الوصول إلى نفس الثنائيات والمكتبات



وهذا واضح في الجهاز الظاهري، لأن VM تطبيقان يرى نفس نظام الملفات (ملفات الجهاز الظاهري). لكن نحن قلنا أن كل حاوية لها خاصتها نظام الملفات المعزول. كيف ال هل يمكن للتطبيقين Bو Aمشاركة نفس الملفات؟

مستويات الصورة في Docker

•تتكون الصور في Dockerمن عدة طبقات.

•يمكن أن تحتوي الصور المختلفة على نفس الطبقات تمامًا، لأن كل صورة Dockerمبنية فوق صورة أخرى ويمكن لصورتين مختلفتين استخدام نفس الصورة الأصلية كقاعدة.

•يتم تخزين كل طبقة مرة واحدة فقط.

•يمكن للحاويتين اللتين تم إنشاؤهما من صورتين بناءً على نفس الطبقات الأساسية قراءة نفس الملفات، ولكن إذا كتبت إحداهما على هذه الملفات، فإن الأخرى لا ترى هذه التعديلات (طبقات الصورة للقراءة فقط).

•عند تنفيذ حاوية، يتم إنشاء طبقة كتابة جديدة أعلى طبقات الصورة.

•عندما تقوم العملية الموجودة في الحاوية بالكتابة إلى ملف في إحدى الطبقات الأساسية، يتم إنشاء نسخة من الملف بأكمله في الطبقة العليا وتكتب العملية إلى النسخة.

النهايات

- إذا كان التطبيق الموجود في حاوية يتطلب إصدارًا محددًا من kernel، فقد لا يعمل على جميع الأجهزة. إذا كان الجهاز يعمل بإصدار مختلف من Linux kernel أو لا يحتوي على نفس وحدات kernel، فلن يتمكن التطبيق من العمل عليه.

•على الرغم من أن الحاويات أخف بكثير من الأجهزة الافتراضية، إلا أنها تفرض قيودًا معينة على التطبيقات التي تعمل فيها. ليس لدى الأجهزة الافتراضية مثل هذه القيود، لأن كل جهاز افتراضي يدير نواة خاصة به.

•لا يمكن تشغيل التطبيق المحتوي على حاوية والمصمم لبنية أجهزة معينة إلا على أجهزة كمبيوتر أخرى لها نفس البنية. لا يمكنك وضع تطبيق مصمم لبنية x86 في حاوية وتتوقع تشغيله على جهاز ARM، حتى لو كان يقوم بتشغيل Docker أيضًا. لا تزال بحاجة إلى جهاز افتراضي لهذا الغرض.

القسم 5:
التنسيق مع Kubernetes (K8s)

التنسيق مع Kubernetes (K8s)

فهم الأصول

•ربما كانت Googleالشركة الأولى التي أدركت أنها بحاجة إلى طريقة أكثر كفاءة لنشر وإدارة مكونات برامجها وبنيتها التحتية لتوسيع نطاقها على نطاق عالمي.

•إنها واحدة من الشركات القليلة في العالم التي تدير مئات الآلاف من الخوادم وتدير عمليات النشر على هذا النطاق.

•على مر السنين، طورت Googleنظامًا داخليًا يسمى Borg(ثم نظامًا جديدًا يسمى Omega)، والذي ساعد كلاً من مطوري التطبيقات ومسؤولي النظام على إدارة هذه الآلاف من التطبيقات والخدمات.

•كما ساعدهم ذلك على استخدام بنيتهم التحتية بشكل أكثر كفاءة.

•بعد إبقاء Omegaو Borgسرًا لمدة عقد من الزمن، قدمت Googleفي عام 2014نظام Kubernetes، وهو نظام مفتوح المصدر يعتمد على الخبرة المكتسبة من خلال Omegaو Borgوأنظمة Googleالداخلية الأخرى.

التنسيق مع Kubernetes (K8s)

فهم K8s

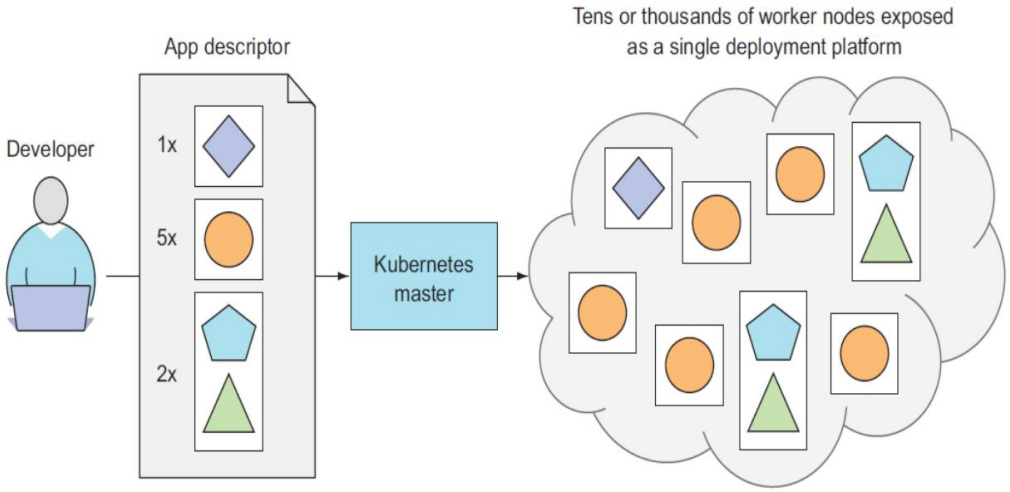
تعريف

Kubernetesعبارة عن منصة محمولة وقابلة للتوسيع ومفتوحة المصدر لأتمتة وإدارة أعباء العمل والخدمات المضمنة في حاويات . نظامها البيئي واسع وينمو بسرعة. تتوفر خدمات Kubernetesوالدعم والأدوات على نطاق واسع.

kubernetes

يأتي اسم Kubernetesمن اللغة اليونانية ويعني قائد الدفة أو الطيار

فهم K8s



عندما يرسل المطور قائمة بالتطبيقات إلى الرئيس، يقوم Kubernetesبنشرها إلى مجموعة العقدة العاملة. لا يهم (ولا ينبغي أن يهم) العقدة التي يصل إليها المكون، سواء للمطور أو لمسؤول النظام.

يمكن للمطور تحديد ضرورة تشغيل تطبيقات معينة معًا وسيقوم Kubernetesبنشرها على نفس العقدة العاملة. وسيتم توزيع الآخرين حول المجموعة، لكن يمكنهم التحدث مع بعضهم البعض بنفس الطريقة بغض النظر عن مكان انتشارهم.

التنسيق مع Kubernetes (K8s)

ما هو استخدام K8s؟

•اكتشاف الخدمة وموازنة التحميل: يمكن لـ Kubernetes الكشف عن حاوية باستخدام اسم DNS أو باستخدام عنوان IP الخاص بها. إذا كانت حركة المرور إلى الحاوية عالية، فيمكن لـ Kubernetes تحميل التوازن وتوزيع حركة مرور الشبكة بحيث يكون النشر مستقرًا.

- تنسيق التخزين: يسمح لك Kubernetes بتثبيت نظام تخزين من اختيارك تلقائيًا، مثل وحدات التخزين المحلية وموفري السحابة العامة وما إلى ذلك.

•النشر والتراجع التلقائي: يمكنك وصف الحالة المطلوبة للحاويات الخاصة بك المنشورة باستخدام Kubernetes وتغيير الحالة الحالية إلى الحالة المطلوبة بسرعة يمكن التحكم فيها. على سبيل المثال، يمكنك أتمتة Kubernetes لإنشاء حاويات جديدة للنشر الخاص بك، وحذف الحاويات الموجودة، واعتماد جميع مواردها في الحاوية الجديدة.

حاوية.

Autopack: •أنت تزود Kubernetes بمجموعة من العقد التي يمكنه استخدامها
تشغيل المهام الحاوية. يمكنك إخبار Kubernetes بمقدار موارد وحدة المعالجة المركزية والذاكرة (RAM) التي تتطلبها كل حاوية. يمكن لـ Kubernetes توسيع نطاق الحاويات الموجودة على العقد الخاصة بك لتحقيق أقصى استفادة من مواردك.

•الإصلاح الذاتي: يقوم Kubernetes بإعادة تشغيل الحاويات التي فشلت، واستبدالها، وحذفها
الحاويات التي لا تستوفي الفحص الصحي المحدد من قبل المستخدم ولا يتم تحريرها حتى تصبح جاهزة.

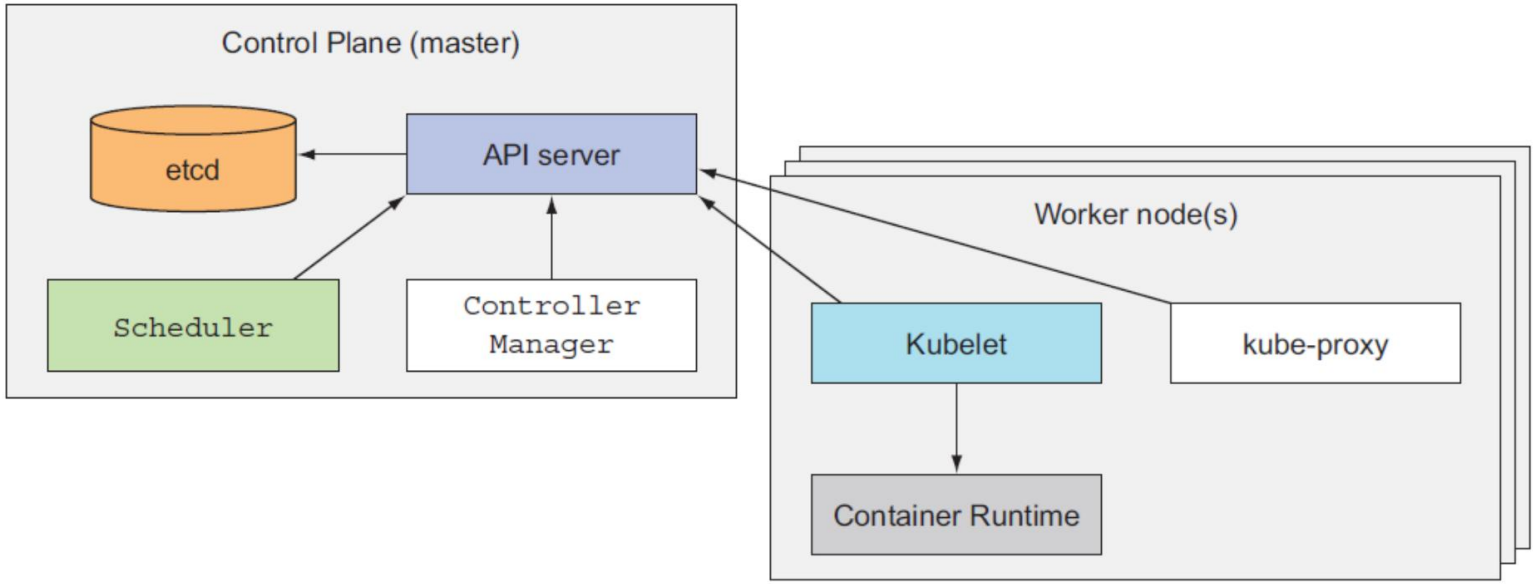
- إدارة السرية والتكوين: يتيح لك Kubernetes تخزين المعلومات الحساسة وإدارتها، مثل كلمات المرور والرموز المميزة ومفاتيح ssh.

بنية K8s

على مستوى الأجهزة، تتكون مجموعة Kubernetesمن عدة عقد، والتي يمكن تقسيمها إلى نوعين:

•العقدة الرئيسية، التي تستضيف مستوى التحكم Kubernetesالذي يتحكم ويدير نظام Kubernetesبأكمله.

دقع•العمل التي تقوم بتشغيل التطبيقات التي تنشرها



بنية K8s

•طائرة التحكم

مستوى التحكم هو ما يتحكم في المجموعة ويجعلها وظيفية. وهو يتألف من مكونات متعددة يمكن تشغيلها على عقدة رئيسية واحدة أو توزيعها عبر عقد متعددة وتكرارها لضمان التوفر العالي.

هذه المكونات هي:

•خادم Kubernetes API، الذي تتواصل من خلاله مع مكونات مستوى التحكم الأخرى.

•برنامج الجدولة، الذي يقوم بجدولة تطبيقاتك (يقوم بتعيين عقدة عاملة لكل مكون قابل للنشر في تطبيقك).

•مدير وحدة التحكم، الذي ينفذ وظائف على مستوى المجموعة مثل النسخ المتماثل للمكونات، ومراقبة عقدة العامل، ومعالجة فشل العقدة، والمزيد.

•مخزن بيانات موزع موثوق به يقوم بتخزين البيانات بشكل دائم etcd، تكوين الكتلة.

بنية K8s

•عقد العمل

العقد العاملة هي الأجهزة التي تقوم بتشغيل تطبيقاتك
في حاويات. يتم تشغيل ومراقبة وتقديم الخدمات لتطبيقاتك باستخدام المكونات التالية:

• Docker أو rkt أو أي بيئة تشغيل أخرى للحاويات، والتي يتم تشغيلها
حاويات.

• Kubelet، الذي يتصل بخادم API ويدير الحاويات الموجودة عليه
العقدة.

•وكيل خدمة Kubernetes (kube-proxy)، الذي يوازن حركة مرور الشبكة بين مكونات التطبيق.

مفهوم الكبسولة

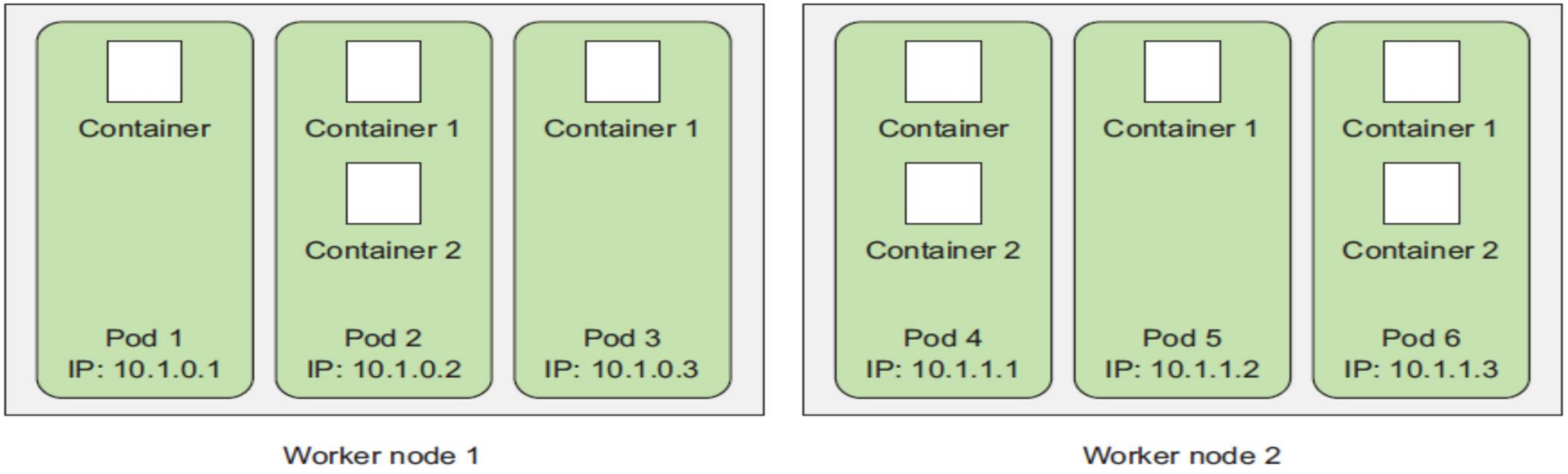
•لا يتعامل Kubernetesبشكل مباشر مع الحاويات الفردية. بدلاً من ذلك، يستخدم مفهوم الحاويات المجمعة المتعددة. تسمى هذه المجموعة من الحاويات بالجراب.

•الكبسولة عبارة عن مجموعة من حاوية واحدة أو أكثر مرتبطة ارتباطًا وثيقًا والتي سيتم تشغيلها دائمًا معًا على نفس العقدة العاملة وفي نفس مساحة اسم Linux.

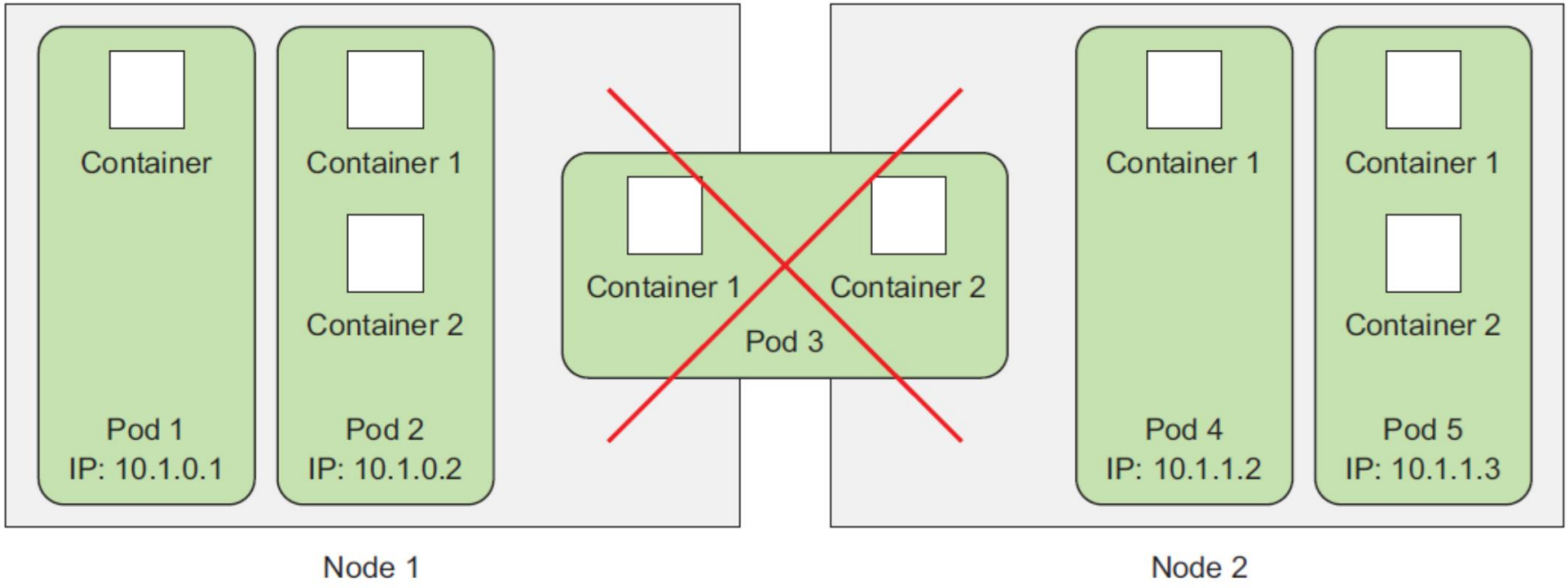
تبدو كل وحدة وكأنها آلة منطقية منفصلة لها عناوين IPالخاصة بها واسم المضيف والعمليات وما إلى ذلك. تشغيل تطبيق واحد.

•

يمكن أن يكون التطبيق عملية واحدة، تعمل في حاوية واحدة أو عملية تطبيق رئيسية وعمليات دعم إضافية، كل منها تعمل في حاويتها الخاصة.



مفهوم الكبسولة



ملحوظة (مهم)

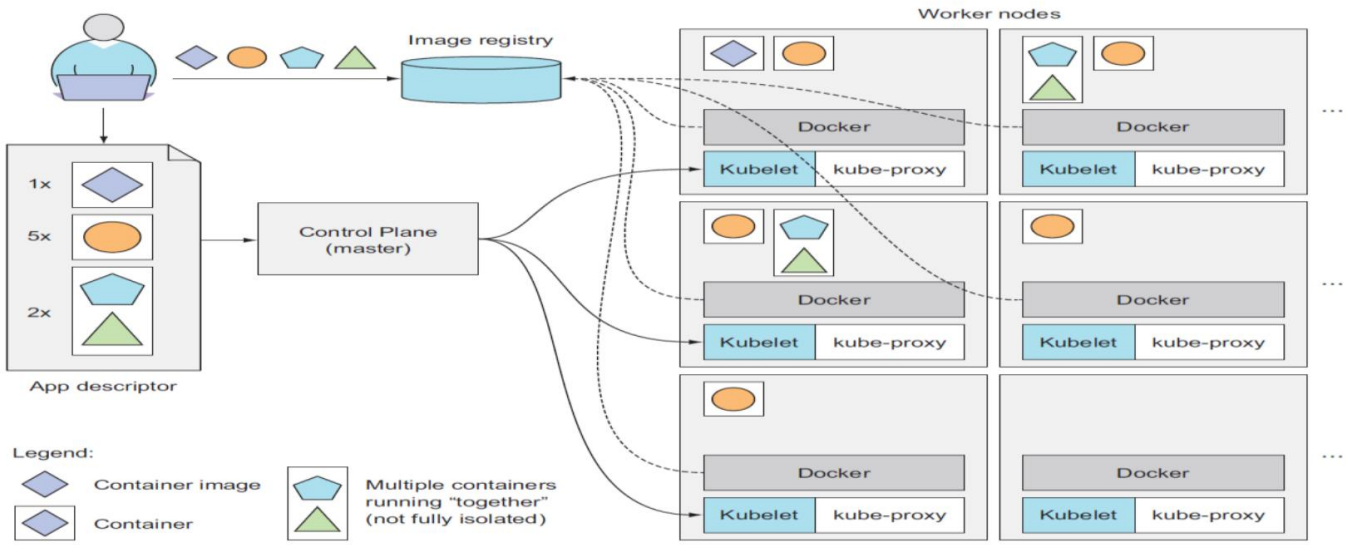
- خلاصة القول حول البودات هي أنه عندما تحتوي الكبسولة على حاويات متعددة، فإنها تعمل دائمًا على عقدة عاملة واحدة ولا تمتد أبدًا إلى عقد عاملة متعددة.

تشغيل التطبيقات باستخدام Kubernetes

•لتشغيل تطبيق في Kubernetes، يجب عليك أولاً تنظيمه في صورة حاوية واحدة أو أكثر، ودفعها إلى سجل الصور، ثم دفع وصف تطبيقك إلى خادم Kubernetes API.

•عندما يقوم خادم API بمعالجة وصف التطبيق الخاص بك، يقوم المجدول بجدولة مجموعات الحاويات المحددة على العقد العاملة المتاحة استنادًا إلى موارد الحساب التي تتطلبها كل مجموعة والموارد غير المخصصة على كل عقدة في ذلك الوقت.

•يقوم Kubelet الموجود على هذه العقد بعد ذلك بتوجيه الحاوية (مثل Docker) لسحب صور الحاوية المطلوبة وتشغيل الحاويات.



التنسيق مع Kubernetes (K8s)

مزايا K8s

- تبسيط نشر التطبيق
- استخدام أفضل للمعدات
- فحص الصحة والشفاء الذاتي
- القياس التلقائي
- تبسيط تطوير التطبيقات

القسم :6
خاتمة