

**César de la Torre**

**Microsoft Corp.**

PUBLICADO POR

microsoft Press

Una división de Microsoft Corporation

Una manera Microsoft

Redmond, Washington 98052-6399

Copyright © 2017 por Microsoft Corporation

Todos los derechos reservados. Ninguna parte del contenido de este libro puede ser reproducida o transmitida en cualquier forma o por cualquier medio sin el permiso por escrito del editor.

Microsoft Press libros están disponibles a través de libreros y distribuidores en todo el mundo. Si necesita apoyo relacionados con este libro, el correo electrónico de Microsoft Press soporte enmspinput@microsoft.com. Por favor, díganos lo que piensa de este libro enhttp://aka.ms/tellpress.

Este libro se proporciona “tal -is” y expresa puntos de vista y las opiniones del autor. Los puntos de vista, opiniones e informaciones expresadas en este libro, incluidas las direcciones URL y otras referencias a sitios web de Internet, pueden cambiar sin previo aviso.

Algunos ejemplos descritos en el presente documento se proporcionan solamente con ilustración y son ficticios. No se encontró asociación o conexión real se pretende ni se debe inferir.

Microsoft y las marcas que figuran en el http://www.microsoft.comen la página web “marcas” son marcas comerciales del grupo de compañías de Microsoft. Todas las demás marcas son propiedad de sus respectivos dueños.

**Adquisiciones Editor:** Janine Patrick

**Editor de desarrollo:** Bob Russell, [Octal Publishing, Inc.](http://www.octalpub.com/)

**Producción editorial:** Dianne Russell, Octal Publishing, Inc.

**Editor de copia:** Bob Russell

2 Contenido



Contenido

**Introducción** **iv**

libros electrónicos gratuitos de Microsoft Press iv

Queremos escuchar de ti iv

Mantente en contacto iv

**Capítulo 1: Introducción a los contenedores y acoplable** **1**

Lo que es acoplable? 2

Comparando contenedores acoplables con máquinas virtuales 3

terminología ventana acoplable 4

Contenedores Docker, imágenes y registros 6

**Capítulo 2: Introducción al ciclo de vida de la aplicación del estibador** **8**

Contenedores, ya que las bases de colaboración DevOps 8

Introducción a una de extremo a extremo de flujo de trabajo del estibador ciclo de vida de la aplicación genérica 10

Beneficios de DevOps para aplicaciones en contenedores 11

**Capítulo 3: Introducción a la plataforma Microsoft y herramientas para aplicaciones en contenedores** **12**

**Capítulo 4: Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y**

**microsoft Azure** **dieciséis**

Diseñar aplicaciones Docker 17

principios de diseño de contenedores comunes 17

Recipiente es igual a un proceso 17

aplicaciones monolíticas 17

aplicación monolítica desplegado como un contenedor 19

La publicación de una sola aplicación envase acoplable al servicio Azure App 20

Estado y datos en aplicaciones Docker 21

aplicaciones SOA 22

Orquestar microservicios y aplicaciones multicontainer de alta escalabilidad

y la disponibilidad 23

ii Contenido

Usando orchestrators a base de contenedores en Azure 25

Utilización del servicio de contenedores Azure 25

Utilización del servicio de Tela 28

Sin estado con estado frente microservicios 31

Entorno de desarrollo para aplicaciones Docker 32

Herramientas de desarrollo de opciones: IDE o editor 32

opciones de idioma y marco 32

flujo de trabajo de desarrollo de lazo interno para aplicaciones Docker 33

La construcción de una sola aplicación dentro de un contenedor acoplable utilizando código de Visual Studio y acoplable CLI 33

El uso de Visual Studio Tools para Docker (Visual Studio en Windows) 41

Configuración de su entorno local 41

El uso de herramientas acoplables en Visual Studio 2015 41

El uso de herramientas acoplables en Visual Studio 2017 42

El uso de comandos de Windows PowerShell en un DockerFile para configurar Windows Contenedores

(Estándar acoplable basa) 43

**Capítulo 5: Docker DevOps aplicaciones de flujo de trabajo con las herramientas de Microsoft** **44**

Los pasos en el flujo de trabajo DevOps-bucle externo para una aplicación acoplable 45

Paso 1: flujo de trabajo de desarrollo-bucle interior 46

Paso 2: Integración de control de código fuente y la gestión con Visual Studio

Servicios de equipo y Git 46

Paso 3: Construir, CI, integrar y prueba con Visual Studio Team Services y acoplable 46

Paso 4: CD, desplegar 51

Paso 5: Ejecutar y gestionar 56

Paso 6: Monitorear y diagnosticar 56

**Capítulo 6: Correr, gestión y seguimiento de los entornos de producción Docker** **57**

La ejecución de aplicaciones compuestas y basados ​​microservicios-en entornos de producción 58

Introducción a orquestadores, programadores, y los racimos de contenedores 58

La gestión de entornos de producción Docker 59

Servicio de contenedores y herramientas de gestión 59

Azure Servicio Tela 60

Supervisión de los servicios de aplicaciones en contenedores 61

Microsoft Application Insights 61

Suite de Gestión de Operaciones de Microsoft 62

**Capítulo 7: Conclusiones** **sesenta y cinco**

comida para llevar clave sesenta y cinco

iii Contenido

[Teclee el texto]



Introducción

Este libro electrónico es una de las muchas ofrecido por Microsoft Press, y es parte de una biblioteca que se extiende por la amplia gama de productos de Microsoft, servicios y tecnologías. Esperamos que le sea útil y le invitamos a explorar los muchos otros títulos disponibles. Para obtener más información sobre esta biblioteca y aprender cómo puede comunicarse con nosotros, haga clic en los enlaces en los apartados siguientes.

libros electrónicos gratuitos de Microsoft Press

A partir de descripciones técnicas a la información en profundidad sobre temas especiales, los libros electrónicos gratuitos de Microsoft Press cubren una amplia gama de temas. Estos libros electrónicos están disponibles en formato PDF, EPUB, y Mobi para Kindle formatos, listo para su descarga en:

<http://aka.ms/mspressfree>

Vuelve a menudo para ver qué hay de nuevo!

Queremos escuchar de ti

En Microsoft Press, su satisfacción es nuestra principal prioridad, y su regeneración nuestro activo más valioso.

Por favor, díganos lo que piensa de este libro en:

http://aka.ms/tellpress

La encuesta es corta, y leemos cada uno de sus comentarios e ideas. ¡Gracias de antemano por sus comentarios!

Mantente en contacto

Vamos a mantener la conversación! Estamos en Twitter:http://twitter.com/MicrosoftPress*.*

iv Introducción

CAPÍTULO 1

Introducción a los contenedores y acoplable

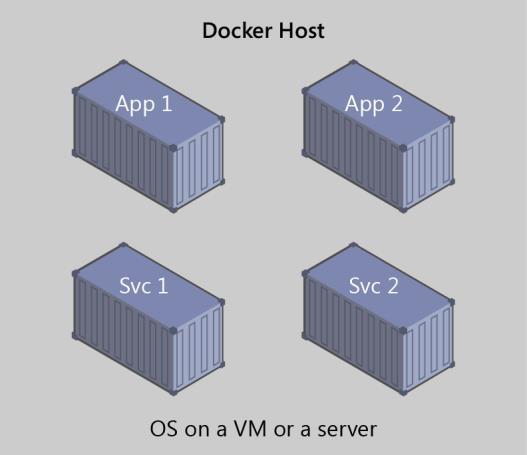
El uso de contenedores es un enfoque para el desarrollo de software en el que una aplicación o servicio, sus dependencias, y su configuración (resumieron como archivos de manifiesto de implementación) se empaquetan junto con la imagen de un contenedor como. A continuación, puede probar la aplicación en contenedores como una unidad y desplegarlo como una instancia imagen depósito al sistema operativo anfitrión.

Al igual que la industria del transporte marítimo utiliza contenedores estandarizados para mover mercancías por barco, tren o camión, con independencia de la carga dentro de ellos, los contenedores de software actúan como una unidad estándar de software que puede contener código y dependencias diferente. La colocación de software en contenedores hace que sea posible que los desarrolladores y profesionales de TI implementar esos contenedores en entornos con poca o ninguna modificación.

Los contenedores también aislar aplicaciones unas de otras en un sistema operativo común (OS). aplicaciones en contenedores se ejecutan en la parte superior de una gran cantidad de contenedores, que a su vez se ejecuta en el sistema operativo (Linux o Windows). Por lo tanto, los contenedores tienen una huella significativamente más pequeño que imágenes de máquina virtual (VM).

Cada contenedor puede ejecutar una aplicación web completo o un servicio, tal como se muestra en la Figura 1-1.

1 CAPÍTULO 1 | Introducción a los contenedores y acoplable



**Figura 1-1:** Múltiples recipientes ejecutan en un host envase

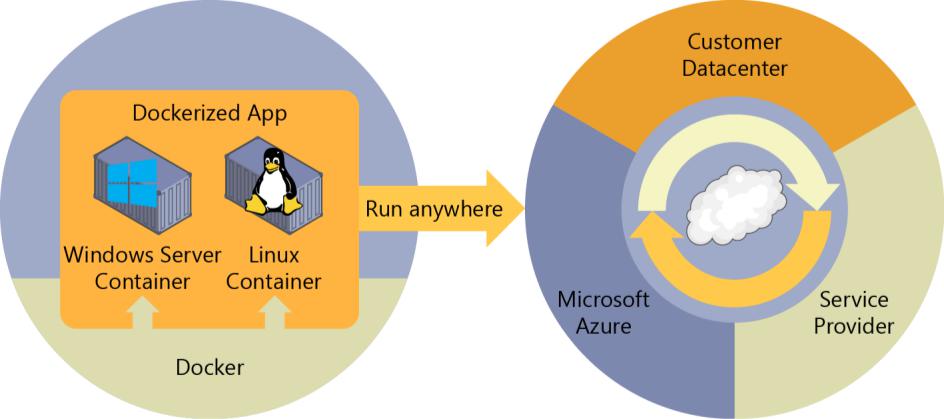
En este ejemplo, acoplable huésped es un huésped contenedor, y App 1, App 2, Svc 1, y Svc 2 son aplicaciones o servicios en contenedores.

Otro beneficio que puede derivarse de la contenerización es la escalabilidad. Puede cambiar la escala de salida de forma rápida mediante la creación de nuevos contenedores para tareas a corto plazo. Desde un punto de vista de aplicación, crear instancias de una imagen (la creación de un contenedor) es similar a instancias de un proceso como una aplicación de servicio o web. Para mayor fiabilidad, sin embargo, cuando se ejecuta varias instancias de la misma imagen a través de múltiples servidores de sistema, normalmente se desea que cada contenedor (instancia imagen) para ejecutar en un servidor host diferente o VM en diferentes dominios de fallo.

En resumen, los contenedores ofrecen las ventajas del aislamiento, la portabilidad, la agilidad, escalabilidad y control a través de todo el flujo de trabajo del ciclo de vida de la aplicación. El beneficio más importante es el aislamiento previsto entre Dev y Operaciones.

Lo que es acoplable?

[Estibador](https://www.docker.com/) es un [proyecto de código abierto](https://github.com/docker/docker) para automatizar el despliegue de aplicaciones como contenedores portátiles, autosuficientes que se pueden ejecutar en la nube o en locales (véase la figura 1-2). Es también una ventana acoplable[empresa](https://www.docker.com/) ese promueve y desarrolla esta tecnología, trabajando en colaboración con la nube, Linux, y los vendedores de Windows, incluyendo Microsoft.



**Figura 1-2:** Acoplable despliega contenedores en todas las capas de la nube híbrida

2 CAPÍTULO 1 | Introducción a los contenedores y acoplable

contenedores imagen acoplables pueden ejecutarse de forma nativa en Linux y Windows. Sin embargo, las imágenes de Windows pueden ejecutarse sólo en los hosts de Windows y las imágenes de Linux pueden ejecutarse sólo en hosts Linux, es decir, un servidor o de una máquina virtual.

Los desarrolladores pueden utilizar entornos de desarrollo en Windows, Linux o MacOS. En el equipo de desarrollo, el desarrollador ejecuta una serie acoplable a la que se despliegan imágenes Docker, incluyendo la aplicación y sus dependencias. Los desarrolladores que trabajan en Linux o en Mac utilizan una serie acoplable que está basado en Linux, y pueden crear imágenes sólo para contenedores de Linux. (Desarrolladores que trabajan en el Mac puede editar código o ejecutar la interfaz de línea de comandos del estibador [CLI] desde MacOS, pero, a partir de este escrito, los contenedores no se ejecutan directamente en MacOS.) Los desarrolladores que funciona en Windows puede crear imágenes, ya sea para Linux o contenedores de Windows.

Para alojar contenedores en entornos de desarrollo y proporcionar herramientas de desarrollo adicionales, barcos Docker [Acoplable Community Edition (CE)](https://www.docker.com/community-edition) para Windows o para MacOS. Estos productos instalar el VM necesario (el anfitrión Docker) para acoger los recipientes. Acoplable también pone a disposición[Acoplable Enterprise Edition](https://www.docker.com/enterprise-edition) [(EE),](https://www.docker.com/enterprise-edition) cual está diseñado para el desarrollo empresarial y es utilizado por los equipos de TI que construyen, enviar y ejecución grandes aplicaciones críticas de negocio en la producción.

Correr [Contenedores de windows,](https://msdn.microsoft.com/en-us/virtualization/windowscontainers/about/about_overview) hay dos tipos de tiempos de ejecución:

* **Servidor de contenedor de Windows** Este tiempo de ejecución proporciona aislamiento de aplicaciones a través de la tecnología de proceso y espacio de nombres aislamiento. Un servidor de contenedor de Windows comparte un núcleo con el anfitrión contenedor y con todos los contenedores que se ejecutan en el host.
* **Hyper-V Container** Esto expande en el aislamiento proporcionado por Contenedores Windows Server mediante la ejecución de cada recipiente en una máquina virtual altamente optimizado. En esta configuración, el núcleo del huésped contenedor no se comparte con el Hyper-V Contenedores, proporcionando un mejor aislamiento.

Las imágenes de estos contenedores se crean de la misma manera y funcionan de la misma. La diferencia está en cómo se crea el contenedor de la imagen a ejecutar un Hyper-V de contenedores requiere un parámetro adicional. Para más detalles, véase[Contenedores de Hyper-V.](https://msdn.microsoft.com/en-us/virtualization/windowscontainers/about/about_overview)

**Comparando contenedores acoplables con máquinas virtuales**

La Figura 1-3 muestra una comparación entre los contenedores VM y acoplables.

Debido a que los contenedores requieren muchos menos recursos (por ejemplo, que no necesitan un sistema operativo completo), que son fáciles de implementar y empiezan rápido. Esto hace que sea posible para que tenga una mayor densidad, lo que significa que puede ejecutar más servicios en la misma unidad de hardware, reduciendo así los costes.

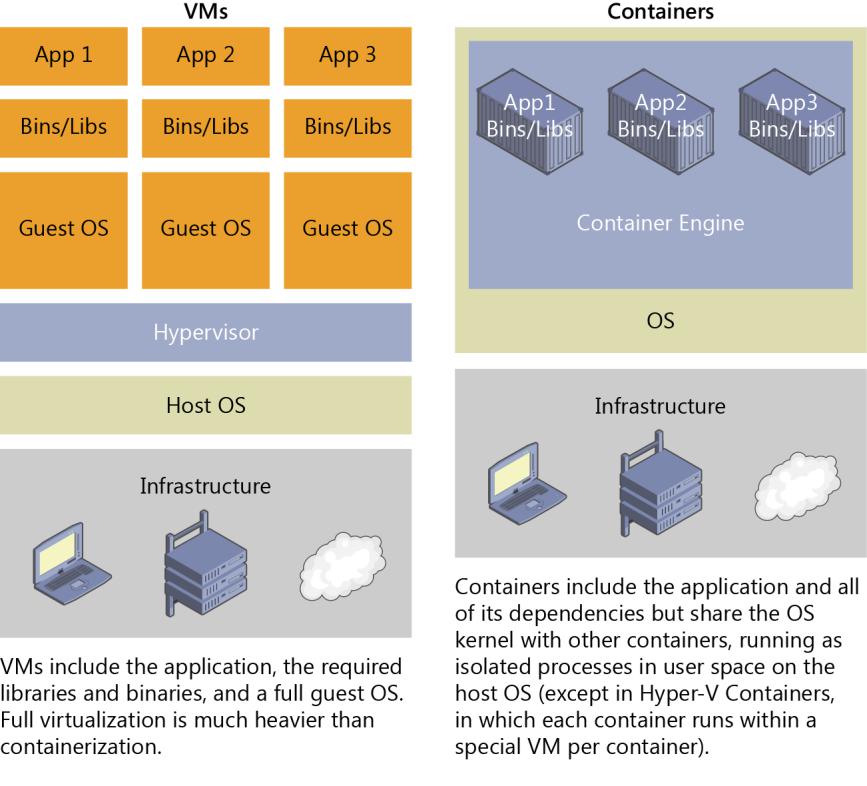
Como un efecto secundario de que se ejecuta en el mismo núcleo, a lograr el aislamiento de menos de máquinas virtuales.

El objetivo principal de una imagen es que hace que el medio ambiente (dependencias) de la misma a través de diferentes implementaciones. Esto significa que puede depurar en su máquina y luego implementarlo en otra máquina con el mismo entorno garantizada.

Una imagen contenedor es una manera de empaquetar una aplicación o servicio y desplegarlo de manera fiable y reproducible. En este sentido, acoplable no es sólo una tecnología, es también una filosofía y un proceso.

Al utilizar estibador, no se oirá dicen los desarrolladores, “Funciona en mi máquina, ¿por qué no en la producción?” Pueden decir simplemente: “Se ejecuta en estibador,” porque la aplicación del estibador empaquetado se puede ejecutar en cualquier entorno acoplable con el apoyo, y se ejecutará la manera que se esperaba que en todos los destinos de despliegue (Dev, QA, puesta en escena, producción, etc.).

3 CAPÍTULO 1 | Introducción a los contenedores y acoplable



**Figura 1-3:** Comparación de la tradicional máquinas virtuales a los contenedores acoplables

terminología ventana acoplable

Esta sección enumera los términos y definiciones con los que debería conocer antes de profundizar en Docker (Para definiciones más detalladas, consulte la extensa [glosario](file:///C:/Users/v-mikepo/AppData/Roaming/Microsoft/Word/Docker) proporcionado por acoplable en

<https://docs.docker.com/v1.11/engine/reference/glossary/>):

* **imagen Envase** Un paquete con todas las dependencias y la información necesaria para crear un contenedor. Una imagen incluye todas las dependencias (tales como marcos), además de despliegue y la configuración para ser utilizada por un tiempo de ejecución recipiente. Por lo general, una imagen se deriva de varias imágenes base que son capas apiladas una encima de la otra para formar el sistema de archivos del contenedor. Una imagen es inmutable después de que ha sido creado.
* **Envase** Un ejemplo de una imagen de Docker. Un contenedor representa un tiempo de ejecución para una sola aplicación, proceso o servicio. Consiste en el contenido de una imagen del estibador, un entorno de ejecución, y un conjunto estándar de instrucciones. Al escalar un servicio, se crea varias instancias de un contenedor de la misma imagen. O bien, un trabajo por lotes puede crear varios contenedores desde la misma imagen, pasando diferentes parámetros para cada instancia.
* **Etiqueta** Una marca o etiqueta que se puede aplicar a las imágenes para que las diferentes imágenes o versiones de la misma imagen (dependiendo del número de versión o el entorno de destino) pueden ser identificados.
* **Dockerfile** Un archivo de texto que contiene instrucciones sobre cómo construir una imagen de estibador.
* **Construir** La acción de construir una imagen Envase con base en la información y el contexto proporcionado por su Dockerfile, así como archivos adicionales en la carpeta donde se construye la imagen. Usted puede construir imágenes mediante el uso de la ventana acoplableacumulación ventana acoplable mando.

4 CAPÍTULO 1 | Introducción a los contenedores y acoplable

* **Repositorios (repos aka)** Una colección de imágenes relacionadas Docker marcadas con una etiqueta que indica la versión de la imagen. Algunos repositorios contienen múltiples variantes de una imagen específica, tal como una que contiene la imagen SDK (más pesado), una imagen que contiene sólo los tiempos de ejecución (más ligero), y así sucesivamente. Esas variantes se pueden marcar con etiquetas. Un único repositorio puede contener variantes de la plataforma, como una imagen de una imagen Windows y Linux.
* **Registro** Un servicio que proporciona acceso a los repositorios. El registro por defecto para la mayoría de las imágenes públicas es[acoplable Hub](https://hub.docker.com/) (Propiedad de estibador como organización). Un registro por lo general contiene los repositorios de múltiples equipos. Las empresas a menudo tienen registros privadas para almacenar y gestionar las imágenes que hemos creado. Azure Container Registro es otro ejemplo.
* **acoplable Hub** Un registro público para subir imágenes y trabajar con ellos. Acoplable Hub ofrece acoplable alojamiento de imágenes, registros públicos o privados, construir disparadores y ganchos web, y la integración con GitHub y Bitbucket.
* **Registro de contenedores Azure** Un recurso público para trabajar con imágenes Docker y sus componentes en Azure. Esto proporciona un registro que está cerca de sus despliegues en Azure y que le da control sobre el acceso, por lo que es posible utilizar sus grupos y permisos de Active Directory Azure.
* **Acoplable Registro de confianza (DTR)** Un servicio de registro Docker (de Docker) que se puede instalar en locales de modo que reside dentro del centro de datos y la red de la organización. Es conveniente para las imágenes particulares que deben ser manejados dentro de la empresa. Acoplable Registro de confianza se incluye como parte del producto del estibador del centro de datos. Para obtener más información, visite[https://docs.docker.com/](https://docs.docker.com/docker-trusted-registry/overview/) [cargador de muelle-confianza-registro / overview /.](https://docs.docker.com/docker-trusted-registry/overview/)
* **Acoplable Community Edition (CE)** herramientas de desarrollo para Windows y MacOS para la construcción, funcionamiento y prueba de contenedores a nivel local. Acoplable CE para Windows proporciona entornos de desarrollo para Linux y Windows Contenedores. El anfitrión de Linux en Windows acoplable se basa en una[Hyper-V](https://www.microsoft.com/en-us/server-cloud/solutions/virtualization.aspx) VM. El anfitrión para Windows contenedores se basa directamente en Windows. Acoplable CE para Mac se basa en el marco de Apple y el hipervisor[hipervisor xhyve,](https://github.com/mist64/xhyve) que proporciona un host Linux acoplable máquina virtual en Mac OS X. acoplable CE para Windows y para Mac reemplaza acoplable Caja de herramientas, que se basa en Oracle VirtualBox.
* **Acoplable Enterprise Edition (EE)** Una versión a escala empresarial de herramientas acoplables para el desarrollo de Linux y Windows.
* **Componer** Una herramienta de línea de comandos y el formato de archivo YAML con metadatos para la definición y ejecución de aplicaciones multicontainer. Se define una única aplicación basada en múltiples imágenes con uno o más archivos .yml que puede anular los valores en función del entorno. Después de haber creado las definiciones, se puede implementar toda la aplicación multicontainer mediante el uso de un solo comando (cargador de muelle-componer hasta) Que crea un contenedor por imagen en el host del estibador.
* **Racimo** Una colección de Docker hosts expuesto como si fueran un único host acoplable virtual, de modo que la aplicación puede escalar para varias instancias de los servicios repartidos a través de múltiples hosts dentro de la agrupación. Se pueden crear grupos acoplables mediante el uso del estibador Swarm, Mesosphere CC / CA, Kubernetes, y Azure Servicio de Tela. (Si utiliza acoplable Swarm para la gestión de un clúster, por lo general se refieren a la agrupación como un enjambre en lugar de un clúster.)
* **Orchestrator** Una herramienta que simplifica la gestión de clusters y anfitriones acoplables. Usando orchestrators, puede administrar sus imágenes, contenedores, y los host a través de un CLI o una interfaz gráfica de usuario. Puede gestionar redes contenedor, configuraciones, balanceo de carga, detección de servicios, alta disponibilidad, configuración de host del estibador, y mucho más. Orquestador es responsable de ejecutar, distribuir, descamación y las cargas de trabajo de sanación a través de una colección de nodos. Típicamente, los productos Orchestrator son los mismos productos que proporcionan infraestructura de cluster, como Mesosphere CC / CA, Kubernetes, acoplable Swarm, y Azure Servicio de Tela.

5 CAPÍTULO 1 | Introducción a los contenedores y acoplable

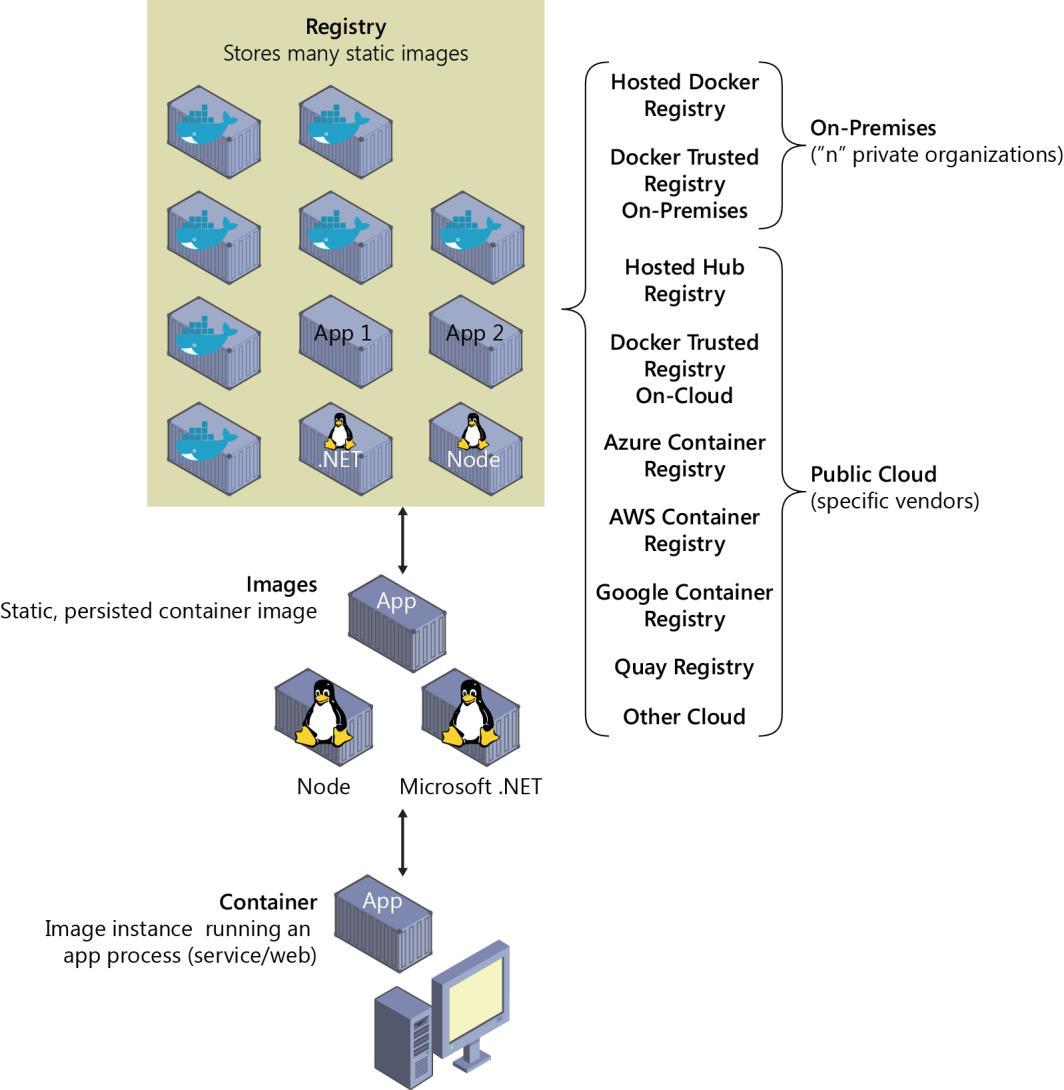
Contenedores Docker, imágenes y registros

Al utilizar estibador, se crea una aplicación o servicio y empaquetarlo y sus dependencias imagen en un recipiente. Una imagen es una representación estática de la aplicación o servicio y su configuración y dependencias.

Para ejecutar la aplicación o servicio, la imagen de la aplicación se crea una instancia para crear un contenedor, que se ejecuta en el host del estibador. Los contenedores se ensayaron inicialmente en un entorno de desarrollo o PC.

Almacenar imágenes en un registro, que actúa como una biblioteca de imágenes. Es necesario un registro cuando se despliega a orchestrators de producción. Cargador de muelle mantiene un registro público a través[acoplable Hub;](https://hub.docker.com/) otros proveedores proporcionan registros de diferentes colecciones de imágenes. Por otra parte, las empresas pueden tener un registro privado en las instalaciones para sus propias imágenes acoplables.

La Figura 1-4 muestra cómo las imágenes y los registros en acoplable relacionan con otros componentes. También muestra las ofertas de registro de múltiples proveedores.



**Figura 1-4:** Taxonomía de los términos y conceptos Docker

6 CAPÍTULO 1 | Introducción a los contenedores y acoplable

Al poner imágenes en un registro, puede almacenar bits de aplicaciones estáticas e inmutables, incluyendo todas sus dependencias, a un nivel de marco. A continuación, puede versión y desplegar imágenes en múltiples entornos y por lo tanto proporcionar una unidad de implementación consistente.

registros de imágenes privadas, ya sean alojados en las instalaciones o en la nube, se recomiendan para las siguientes situaciones:

* Sus imágenes no deben ser compartidos públicamente debido a la confidencialidad.
* ¿Quieres tener la latencia de red mínima entre las imágenes y el entorno de despliegue elegida. Por ejemplo, si su entorno de producción es azul, es probable que desee almacenar las imágenes en Azure Registro recipiente para que la latencia de red será mínimo. De manera similar, si su entorno de producción es en las instalaciones, es posible que desee tener un estibador registro en las instalaciones de confianza disponibles dentro de la misma red local.

7 CAPÍTULO 1 | Introducción a los contenedores y acoplable

CAPÍTULO 2

Introducción al ciclo de vida de la aplicación del estibador

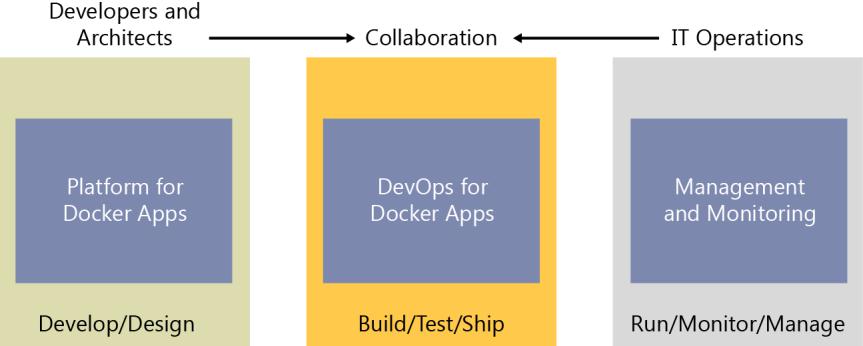
El ciclo de vida de las aplicaciones en contenedores es un viaje que comienza con el desarrollador. El desarrollador elige implementar contenedores y estibador, ya que elimina las fricciones en el despliegue y las operaciones de TI, que en última instancia ayuda a todos a ser más ágil, más productivo de extremo a extremo, y más rápido.

Contenedores, ya que las bases de colaboración DevOps

Por la propia naturaleza de los contenedores y la tecnología del estibador, los desarrolladores pueden compartir su software y dependencias fácilmente con las operaciones de TI y entornos de producción al tiempo que elimina el típico “funciona en mi máquina” excusa. Contenedores resolver conflictos entre aplicaciones entre diferentes entornos. Indirectamente, contenedores y acoplable llevar a los desarrolladores y las operaciones de TI más juntos, por lo que es más fácil para ellos para colaborar de manera efectiva. Adoptando el flujo de trabajo contenedor proporciona muchos clientes

con la continuidad DevOps que han buscado, pero antes tenían que poner en práctica a través de una configuración más compleja para la liberación y construir oleoductos. Contenedores simplifican la acumulación / test / desplegar tuberías en DevOps.

8 CAPÍTULO 2 | Introducción al ciclo de vida de la aplicación del estibador



**Figura 2-1:** Principales cargas de trabajo por “personas” en la vidaciclo para aplicaciones acoplables en contenedores

Con contenedores Docker, los desarrolladores tener lo que es dentro del contenedor (aplicaciones y servicios, y las dependencias de los marcos y componentes) y cómo los contenedores y servicios comportan juntos como una aplicación compuesta por un conjunto de servicios. Las interdependencias de los múltiples contenedores se definen en un archivo de cargador de muelle-compose.yml, o lo que podría llamarse un manifiesto de implementación. Mientras tanto, los equipos de operaciones de TI (profesionales y de gestión de TI) pueden centrarse en la gestión de los entornos de producción; infraestructura; escalabilidad; supervisión; y, en última instancia, lo que garantiza que las aplicaciones están cumpliendo adecuadamente para los usuarios finales, sin necesidad de conocer los contenidos de los diferentes contenedores. Por lo tanto, el “contenedor”, nombre que recuerda la analogía con los contenedores de transporte en el mundo real. Así, los propietarios del contenido de un recipiente no tienen por qué se preocupan de cómo será enviado al recipiente, y la compañía naviera transporta un contenedor desde su punto de origen hasta su destino sin saber o preocuparse acerca de los contenidos. De manera similar, los desarrolladores pueden crear y poseer el contenido dentro de un recipiente acoplable sin la necesidad de preocuparse por los mecanismos de “transporte”.

En el pilar en el lado izquierdo de la figura 2-1, los desarrolladores a escribir y ejecutar código localmente en contenedores de Docker utilizando acoplable para Windows o Mac. Ellos definen el entorno operativo para el código mediante el uso de un Dockerfile que especifica el sistema operativo base para funcionar, así como los pasos de generación para la construcción de su código de una imagen de estibador en. Los desarrolladores definen cómo las una o más imágenes van a funcionar utilizando la ventana acoplable-compose.yml manifiesto de implementación mencionado fichero. A medida que completen su desarrollo local, empujan su código de aplicación además de los archivos de configuración del Docker al repositorio de código de su elección (es decir, repositorio Git).

El pilar DevOps define las tuberías de construcción-continua Integración (CI) utilizando el Dockerfile proporcionada en el repositorio de código. El sistema de CI tira de las imágenes de contenedores de base del registro acoplable seleccionado y construye las imágenes personalizadas de Docker para la aplicación. Las imágenes a continuación, se validan y empujaron al registro del estibador utilizado para los despliegues en múltiples entornos.

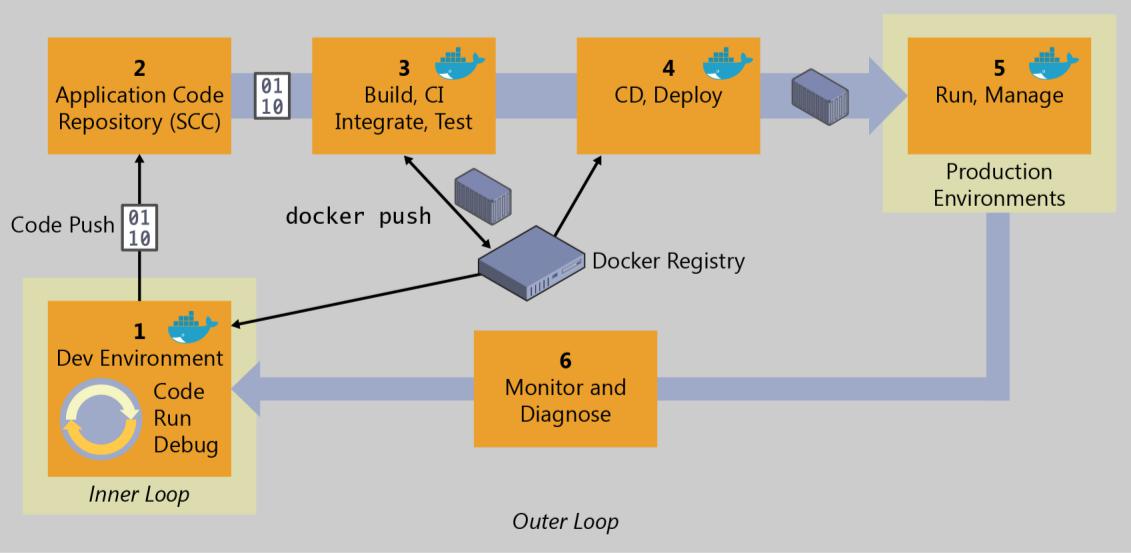
En la columna de la derecha, los equipos de operaciones administrar aplicaciones e infraestructuras desplegadas en la producción mientras se monitoriza el medio ambiente y aplicaciones para que puedan dar su opinión y puntos de vista al equipo de desarrollo sobre la manera de mejorar la aplicación. aplicaciones de contenedores están normalmente se ejecutan en la producción usando orchestrators contenedores.

Los dos equipos están colaborando a través de una plataforma fundamental (contenedores de Docker) que proporciona una separación de intereses como un contrato, al tiempo que mejora en gran medida la colaboración de los dos equipos en el ciclo de vida de la aplicación. Los desarrolladores poseen el contenido del recipiente, su entorno operativo, y las interdependencias de contenedores, mientras que los equipos de operaciones toman las imágenes construidas junto con el manifiesto y los ejecuta en su sistema de orquestación.

9 CAPÍTULO 2 | Introducción al ciclo de vida de la aplicación del estibador

**Introducción a una de extremo a extremo de flujo de trabajo del estibador ciclo de vida de la aplicación genérica**

Figura 2-2 presenta un flujo de trabajo más detallado para un ciclo de vida de la aplicación del estibador, centrándose en este caso sobre las actividades y activos DevOps específicos.



**Figura 2-2:** flujo de trabajo de alto nivel para el ciclo de vida de las aplicaciones en contenedores acoplable

Todo comienza con el promotor, que empieza a escribir código en el flujo de trabajo de bucle interno. La etapa de bucle interior es donde los desarrolladores definen todo lo que sucede antes de empujar código en el repositorio de código (por ejemplo, un sistema de control de fuente tal como Git). Después de que se ha comprometido, desencadena el repositorio de integración continua (CI) y el resto del flujo de trabajo.

El bucle interno, básicamente, consiste en los pasos típicos como el “código”, “correr”, “prueba” y “depuración”, además de medidas adicionales directamente antes de ejecutar la aplicación a nivel local. Esto es cuando el desarrollador se ejecuta y pone a prueba la aplicación como un contenedor estibador. El flujo de trabajo de bucle interior se explicará en las secciones que siguen.

Dando un paso atrás para mirar el extremo a extremo de flujo de trabajo, el flujo de trabajo DevOps es más que una tecnología o un conjunto de herramientas: es una mentalidad que requiere la evolución cultural. Es la gente, los procesos y las herramientas adecuadas para hacer de su ciclo de vida de las aplicaciones más rápido y más predecible. Las empresas que adoptan un flujo de trabajo en contenedores normalmente reestructurar sus organizaciones para representar a las personas y procesos que coinciden con el flujo de trabajo en contenedores.

DevOps practicantes pueden ayudar a los equipos responden más rápido, junto a las presiones competitivas mediante la sustitución de los procesos manuales propensos a errores con la automatización, lo que se traduce en una mejora de la trazabilidad y los flujos de trabajo repetibles. Las organizaciones también pueden gestionar entornos más eficiente y conseguir ahorros de costes con una combinación de en las instalaciones y recursos de la nube, así como herramientas altamente integrado.

Al implementar el flujo de trabajo para aplicaciones DevOps Docker, verá que las tecnologías de Docker están presentes en casi todas las etapas del flujo de trabajo, desde su caja de desarrollo mientras se trabaja en el bucle interno (código, ejecutar, depurar), a la acumulación de los Ensayos fase CI, y, por supuesto, en los entornos de producción y de parada y cuando el despliegue de sus contenedores a esos ambientes.

Mejora de las prácticas de calidad ayuda a identificar defectos tempranos en el ciclo de desarrollo, lo que reduce el costo de la fijación de ellos. Al incluir el medio ambiente y las dependencias de la imagen y la adopción de una filosofía de la implementación de la misma imagen en múltiples entornos, se promueve una disciplina de la extracción de las configuraciones específicas del entorno que hacen despliegues más fiable.

10 CAPÍTULO 2 | Introducción al ciclo de vida de la aplicación del estibador

riqueza de datos obtenidos a través de la instrumentación efectiva (monitorización y diagnóstico) proporciona una idea de los problemas de rendimiento y comportamiento de los usuarios para orientar las prioridades y las inversiones futuras.

DevOps deben considerarse como un viaje, no un destino. Se debe implementar de forma incremental a través de proyectos de ámbito apropiada desde el que puede demostrar el éxito, aprender y evolucionar.

**Beneficios de DevOps para aplicaciones en contenedores**

Éstos son algunos de los beneficios más importantes que ofrece un flujo de trabajo DevOps sólida:

* Entregar software de mejor calidad, más rápido y con mejor cumplimiento
* Impulsar la mejora continua y ajustes antes y de forma más económica
* Aumentar la transparencia y la colaboración entre los actores involucrados en la entrega y el software operativo
* Controlar los costes y utilizar los recursos aprovisionados con mayor eficacia y reducir al mínimo los riesgos de seguridad
* Enchufar y jugar bien con muchos de sus inversiones DevOps existentes, incluyendo las inversiones en código abierto

11 CAPÍTULO 2 | Introducción al ciclo de vida de la aplicación del estibador

CAPÍTULO 3

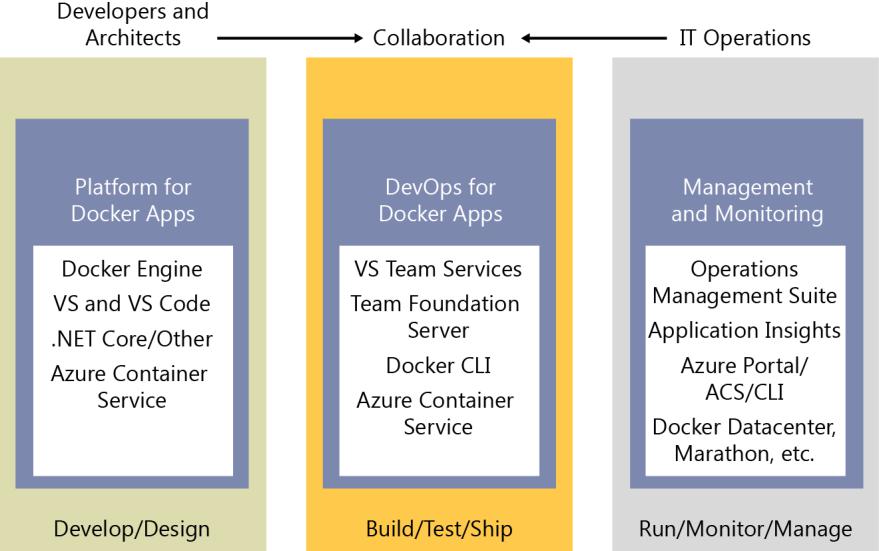
Introducción a la plataforma Microsoft

y herramientas para aplicaciones en contenedores

Visión: Crear una, de nivel empresarial adaptable, ciclo de vida de las aplicaciones en contenedores que se extiende por su desarrollo, las operaciones de TI y gestión de la producción.

La Figura 3-1 muestra los pilares fundamentales en el ciclo de vida de aplicaciones acoplable clasifican por el tipo de trabajos efectuados por los múltiples equipos (App-desarrollo, DevOps procesos de infraestructura, y la gestión de TI y operaciones). Por lo general, en la empresa, los perfiles de “la persona” responsable de cada área son diferentes. También lo son sus habilidades.

12 CAPÍTULO 3 | Introducción a la plataforma Microsoft y herramientas para aplicaciones en contenedores



**Figura 3-1:** Principales pilares en el ciclo de vida de aplicaciones acoplables en contenedores con plataforma y herramientas de Microsoft

Un flujo de trabajo del ciclo de vida del estibador en contenedores puede ser inicialmente prescriptivo basado en “de opciones de productos -default”, por lo que es fácil para los desarrolladores para empezar más rápido, pero es fundamental que bajo el capó tiene que haber un marco abierto de modo que será una flujo de trabajo flexible, capaz de adaptarse a los diferentes contextos de cada organización o empresa. La infraestructura de flujo de trabajo (componentes y productos) debe ser lo suficientemente flexible como para cubrir el medio ambiente que cada empresa tendrá en el futuro, incluso siendo capaz de cambiar o de desarrollo DevOps productos a otros. Esta flexibilidad, apertura y amplia variedad de tecnologías en la plataforma y la infraestructura son precisamente las prioridades de Microsoft para aplicaciones acoplables en contenedores, como se explica en los capítulos que siguen.

Tabla 3-1 demuestra que la intención de los DevOps Microsoft para aplicaciones acoplables en contenedores es proporcionar un flujo de trabajo DevOps abierta para que pueda elegir qué productos usar para cada fase (Microsoft o un tercero) mientras que proporciona un flujo de trabajo simplificado que proporciona “por -default-productos”ya conectado; por lo tanto, que rápidamente puede empezar con su nivel de empresa DevOps flujo de trabajo para aplicaciones acoplables.

**Tabla 3-1:** DevOps abierta a cualquier tecnología de flujo de trabajo



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Anfitrión** |  | **tecnologías de Microsoft** | |  |  | **Tercero-Azure enchufable** | |  |
|  |  |  |  |  |  |  | |  |
| plataforma de |  | • | Microsoft Visual Studio y Visual |  |  | • Cualquier editor de código (por ejemplo, Sublime) | |  |
| aplicaciones Docker |  | • | Código Studio |  |  | • Cualquier idioma (Node.js, Java, Ir, | |  |
|  |  | .RED |  |  |  | etc.) |  |
|  |  | • | Servicio Microsoft Azure de contenedores |  |  | • Cualquier orquestador y planificador | |  |
|  |  | • | Azure Servicio Tela |  |  | • | Cualquier registro acoplable |  |
|  |  | • | Registro de contenedores Azure |  |  |  |  |  |
|  |  |  |  |  |  |  | |  |
| DevOps para |  | • | Servicios de Visual Studio Team |  |  | • GitHub, Git, Subversion, etc. | |  |
| aplicaciones Docker |  | • | Microsoft Team Foundation Servidor |  |  | • | Jenkins, cocinero, de marionetas, velocidad, |  |
|  |  | • | Contenedor de servicios Azure |  |  |  | CircleCI, TravisCI, etc. |  |
|  |  |  |  | • En las instalaciones del centro de datos del estibador, | |  |
|  |  | • | Azure Servicio Tela |  |  |  |
|  |  |  |  |  | Acoplable Swarm, mesos CC / CA, |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | Kubernetes, etc. |  |
|  |  |  |  |  |  |  |  |  |
| Administración y |  | • | Suite de Gestión de Operaciones |  |  | • | Maratón, Cronos, etc. |  |
| supervisión |  | • | Aplicaciones Insights |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

13 CAPÍTULO 3 | Introducción a la plataforma Microsoft y herramientas para aplicaciones en contenedores



La plataforma de Microsoft y herramientas para aplicaciones acoplables en contenedores, tal como se define en la Tabla 3-1, comprenden los siguientes componentes:

* **Plataforma para el desarrollo del estibador Aplicaciones** El desarrollo de un servicio o conjunto de servicios que componen una “aplicación”. La plataforma de desarrollo ofrece a todos los desarrolladores de trabajo requiere antes de empujar su código a un repositorio de código compartido. El desarrollo de los servicios, desplegados como contenedores, son muy similares a la elaboración de las mismas aplicaciones o servicios sin estibador. Continúa utilizando su idioma preferido (.NET, Node.js, Ir, etc.) y editor preferido o IDE

como Visual Studio o código de Visual Studio. Sin embargo, en lugar de considerar acoplable un destino de despliegue, que desarrolla sus servicios en el entorno del estibador. Crear, ejecutar, probar y depurar el código en contenedores a nivel local, que proporciona un entorno de destino en el tiempo de desarrollo. Al proporcionar el entorno de destino a nivel local, contenedores Docker establecieron lo que va drásticamente ayudará a mejorar su ciclo de vida DevOps. Visual Studio y Visual Studio Código tienen extensiones para integrar contenedores acoplables dentro de su proceso de desarrollo.

* **DevOps para Aplicaciones Docker** Los desarrolladores que crean aplicaciones acoplables pueden utilizar los servicios de Visual Studio Team (VSTS) o cualquier otro producto de terceros, como Jenkins, para construir una gestión integral del ciclo de vida de aplicaciones automatizado (ALM).

Con VSTS, los desarrolladores pueden crear DevOps centrado de contenedor para un proceso rápido, iterativo que cubre el control de código fuente desde cualquier lugar (VSTS-Git, GitHub, cualquier repositorio Git remoto, o Subversion), Continuo Integración (CI), las pruebas de unidad interna, entre otras contenedor / pruebas de integración de servicios, entrega continua (DC), y la gestión de la liberación (RM). Los desarrolladores también pueden automatizar sus versiones de aplicaciones acoplables en Azure Contenedor de servicios, desde el desarrollo de entornos de ensayo y producción.

* gestión de la producción y el monitoreo de TI.

**administración** TI puede administrar aplicaciones y servicios de producción de varias maneras:

* + **portal de Azure** Si está utilizando orchestrators de código abierto, Azure Servicio de Contenedores (ACS), además de gestión de clústeres herramientas como estibador de centros de datos y Marathon Mesosphere le ayudan a crear y mantener sus entornos acoplables. Si está utilizando Azure Servicio de Tela, el Servicio

Tela herramienta Explorador hace posible visualizar y configurar el clúster.

* + **herramientas acoplables** Puede gestionar sus aplicaciones de recipientes usando herramientas conocidas. No hay necesidad de cambiar sus prácticas de gestión del estibador existentes para mover cargas de trabajo de contenedores a la nube. Utilice las herramientas de gestión de aplicaciones que ya está familiarizado con y conectarse a través de los puntos finales de API estándar para el orquestador de su elección. También puede utilizar otras herramientas de terceros para administrar sus aplicaciones acoplables, como estibador centro de datos o incluso herramientas acoplables CLI.
  + **herramientas de código abierto** Debido a ACS expone los puntos finales de API estándar para el motor de organización, las herramientas más populares son compatibles con ACS y, en la mayoría de los casos, va a funcionar fuera de la caja, incluyendo visualizadores, el seguimiento, herramientas de línea de comandos, e incluso instrumentos de futuros a medida que estén disponible.

**Supervisión** Mientras se ejecuta entornos de producción, puede supervisar todos los ángulos utilizando el siguiente:

* **Operaciones Management Suite (OMS)** La “Solución de contenedores de OMS” puede gestionar y controlar los ejércitos y contenedores acoplables al mostrar información acerca de dónde están sus recipientes y contenedores son anfitriones, que los contenedores se están ejecutando o fracasaron y demonio y contenedores registros acoplables. También muestra las métricas de rendimiento, tales como CPU, memoria, red y almacenamiento para el envase y acoge para ayudarle a solucionar y encontrar contenedores vecino ruidoso.

14 CAPÍTULO 3 | Introducción a la plataforma Microsoft y herramientas para aplicaciones en contenedores

* **Perspectivas de aplicación** Puede supervisar las aplicaciones de producción Docker simplemente establecer su SDK en sus servicios de modo que usted puede obtener los datos de telemetría de las aplicaciones.

Por lo tanto, Microsoft ofrece una base completa para un ciclo de vida de las aplicaciones extremo a extremo en contenedores estibador. Sin embargo, es una colección de productos y tecnologías que le permiten seleccionar opcionalmente y se integran con herramientas y procesos existentes. La flexibilidad en un amplio enfoque, junto con la fuerza de la profundidad de las capacidades coloque Microsoft en una posición fuerte para el desarrollo de aplicaciones en contenedores estibador.

15 CAPÍTULO 3 | Introducción a la plataforma Microsoft y herramientas para aplicaciones en contenedores

CAPÍTULO 4

Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

Visión: Diseñar y desarrollar soluciones escalables con estibador en mente.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

Hay muchos casos de uso gran-aptos para contenedores, no sólo para las arquitecturas microservicios orientado, sino también en las que simplemente tiene servicios regulares o aplicaciones web para correr y desea reducir las fricciones entre desarrollo y medio ambiente de producción despliegues.

Diseñar aplicaciones Docker

Capítulo 1 introduce los conceptos fundamentales con respecto a los envases y estibador. Esa información es el nivel básico de información que necesita para empezar. Sin embargo, las aplicaciones empresariales pueden ser complejos y estar compuestos de múltiples servicios en lugar de un único servicio o contenedor. Para esos casos de uso opcional, lo que necesita saber enfoques adicionales para el diseño, como la arquitectura orientada a servicios (SOA) y los más avanzados conceptos microservicios y conceptos de orquestación contenedor. El alcance de este documento no se limita a microservicios sino a cualquier ciclo de vida de la aplicación del estibador, por lo tanto, no explora microservicios la arquitectura en profundidad porque también se puede utilizar contenedores y acoplable con SAO, tareas regulares de fondo o puestos de trabajo, o incluso con aplicación monolítica enfoques de implementación.

Sin embargo, antes de entrar en el ciclo de vida de las aplicaciones y DevOps, es importante saber cómo se va a diseñar y construir su aplicación y cuáles son sus opciones de diseño.

principios de diseño de contenedores comunes

Por delante de entrar en el proceso de desarrollo hay algunos conceptos básicos dignos de mención con respecto a cómo se utiliza contenedores.

**Recipiente es igual a un proceso**

En el modelo de recipiente, un contenedor representa un único proceso. Mediante la definición de un recipiente como un límite de proceso, se empieza a crear las primitivas utilizadas a escala, o -off por lotes, procesos. Cuando se ejecuta un recipiente estibador, verá una[PUNTO DE ENTRADA](https://docs.docker.com/engine/reference/builder/#/entrypoint) definición. Esto define el proceso y el tiempo de vida del recipiente. Cuando finalice el proceso, el ciclo de vida del recipiente termina. Hay procesos de larga duración, tales como servidores web, y los procesos de corta duración, tales como trabajos por lotes, que podrían haber sido implementado como Microsoft Azure[WebJobs.](https://azure.microsoft.com/en-us/documentation/articles/websites-webjobs-resources/) Si el proceso falla, termina el contenedor y el orquestador se hace cargo. Si el orquestador fue instruido para mantener cinco casos corriendo y uno falla, el orquestador creará otro recipiente para reemplazar el fallido proceso. En un trabajo por lotes, el proceso se inicia con los parámetros. Cuando finalice el proceso, el trabajo se haya completado.

Es posible encontrar un escenario en el que desea que varios procesos que se ejecutan en un solo contenedor. En cualquier documento de arquitectura, nunca hay un “nunca”, ni hay siempre un “siempre”. Para los escenarios que requieren varios procesos, un patrón común es utilizar[Supervisor.](http://supervisord.org/)

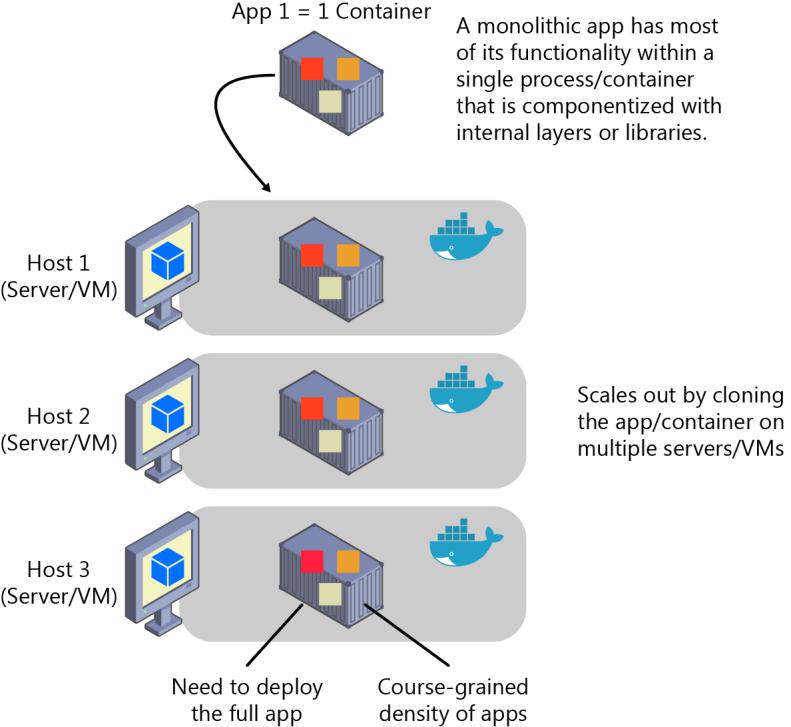
aplicaciones monolíticas

En este escenario, se está construyendo una aplicación web única y monolítica o servicio y su despliegue como un contenedor. Dentro de la aplicación, la estructura podría no ser monolítica; que podría comprender varias bibliotecas, componentes, o incluso capas (capa de aplicación, la capa de dominio, la capa de acceso de datos, etc.). Externamente, se trata de un solo contenedor, como un solo proceso, aplicación web única, sola o servicio.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

Para gestionar este modelo, se implementa un único contenedor para representar la aplicación. Modificar la escala, simplemente añada unas cuantas más copias con un equilibrador de carga en el frente. La simplicidad viene de la gestión de un solo despliegue en un único contenedor o de la máquina virtual (VM).

Siguiendo el principio de que un contenedor hace una cosa única, y lo hace en un solo proceso, el patrón monolítico está en conflicto. Puede incluir múltiples componentes / bibliotecas o capas internas dentro de cada contenedor, como se ilustra en la Figura 4-1.



**Figura 4-1:** Un ejemplo de arquitectura de la aplicación monolítica

La desventaja de este enfoque viene si o cuando la aplicación crece, lo que requiere que a escala. Si toda la aplicación a escala, no es realmente un problema. Sin embargo, en la mayoría de los casos, algunas partes de la aplicación son los cuellos de botella que requieren de escala, mientras que otros componentes se utilizan menos.

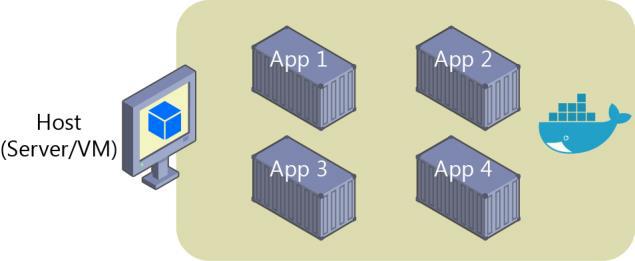
Usando el ejemplo típico -commerce correo, lo que probablemente es la necesidad de ampliar el componente de información del producto. Muchos más clientes navegar por productos que comprarlos. Más clientes utilizan su cesta de usar la tubería de pago. Menos clientes añadir comentarios o ver su historial de compras. Y es probable que tenga sólo un puñado de empleados, en una sola región, que necesidad de gestionar los contenidos y campañas de marketing. Escalando el diseño monolítico, todo el código se despliega en múltiples ocasiones.

Además del problema de “escala-todo”, los cambios en un único componente requieren repetición de pruebas completa de toda la aplicación, así como una redistribución completa de todos los casos.

El enfoque monolítico es común, y muchas organizaciones están desarrollando con este método arquitectónico. Muchos disfrutan suficientemente buenos resultados, mientras que otros se encuentran con límites. Muchos diseñado sus aplicaciones en este modelo porque las herramientas y la infraestructura eran demasiado difíciles de construir SOA, y ellos no vieron la necesidad, hasta la aplicación creció.

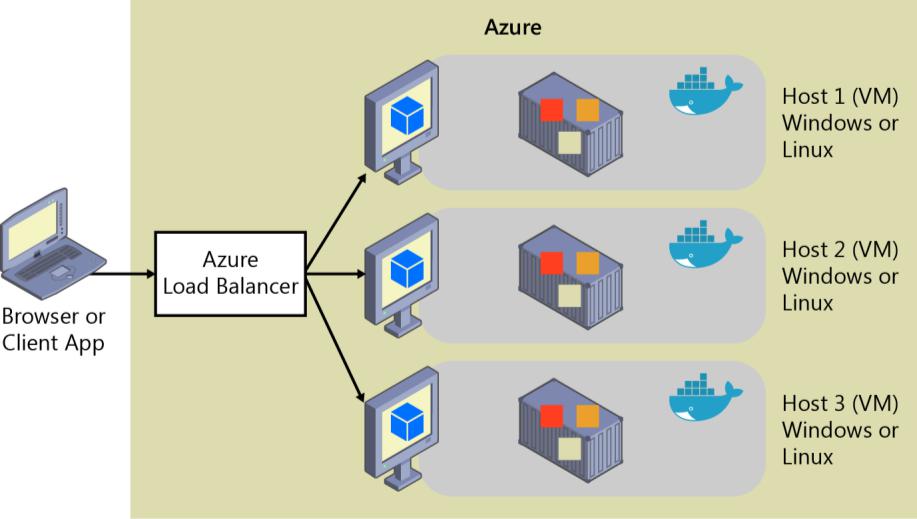
Desde una perspectiva de la infraestructura, cada servidor puede ejecutar muchas aplicaciones dentro de la misma máquina y tienen una relación aceptable de eficiencia en su uso de los recursos, como se muestra en la Figura 4-2.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure



**Figura 4-2:** Un host que ejecuta múltiples aplicaciones / contenedores

Puede implementar aplicaciones monolíticas en Azure mediante el uso de máquinas virtuales dedicados para cada instancia. Utilizando[Scale Sets Azure VM,](https://azure.microsoft.com/en-us/documentation/services/virtual-machine-scale-sets/) tú puede escalar fácilmente las máquinas virtuales. [Azure Servicios App](https://azure.microsoft.com/en-us/services/app-service/) poder ejecutar aplicaciones monolíticas y escalar fácilmente casos sin tener que gestionar las máquinas virtuales. Desde 2016, Azure App Servicios puede ejecutar instancias únicas de recipientes estibador, así, lo que simplifica la implementación. Y, utilizando estibador, se puede implementar una única máquina virtual como anfitrión del estibador y ejecutar múltiples instancias. Utilizando el equilibrador Azure, como se ilustra en la Figura 4-3, se puede administrar de escala.



**Figura 4-3:** Múltiples hosts de escala de salida de una sola Aplicaciones de aplicación acoplable / contenedores

Puede gestionar el despliegue de los diversos anfitriones a través de técnicas de despliegue tradicionales. Puede administrar hosts acoplables mediante el uso de comandos comocarrera ventana acoplable de forma manual, a través de la automatización tales como tuberías (CD) entrega continua, que explicaremos más adelante en este libro electrónico.

**aplicación monolítica desplegado como un contenedor**

Hay beneficios en el uso de recipientes para gestionar implementaciones monolíticas. Escalado las instancias de contenedores es mucho más rápido y más fácil que el despliegue de máquinas virtuales adicional. Aunque Scale Sets VM son una gran característica para escalar las máquinas virtuales, que son necesarios para albergar los contenedores de Docker, se toman el tiempo de configurar. Cuando se implementa como instancias de aplicaciones, la configuración de la aplicación es administrado como parte de la máquina virtual.

La implementación de actualizaciones como imágenes Docker es mucho más rápido y eficiente de la red. Las instancias Vn se pueden configurar en los mismos hosts como sus instancias Vn-1, eliminando costes añadidos resultantes de máquinas virtuales adicional. Docker imágenes suelen comenzar en cuestión de segundos, lo que acelera los despliegues. Derribando una instancia del estibador es tan fácil como invocando elparada ventana acoplable de comandos, por lo general completar en menos de un segundo.

Debido a que los contenedores son inherentemente inmutable, por su diseño, que no tendrá que preocuparse por las máquinas virtuales dañado debido a un script de actualización se olvidó de dar cuenta de una cierta configuración o archivo específico a la izquierda en el disco.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

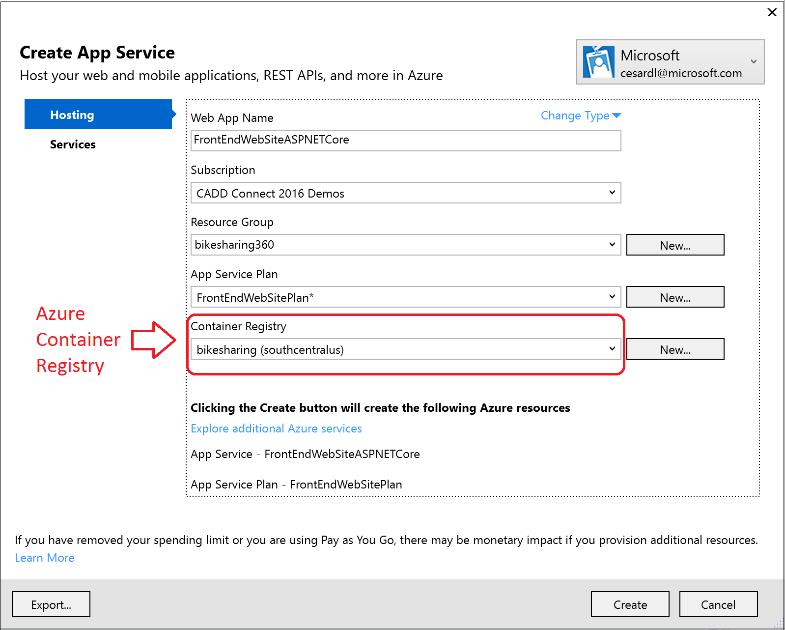
Aunque las aplicaciones monolíticas pueden beneficiarse de estibador, estamos tocando sólo las puntas de los beneficios. Los beneficios más grandes de contenedores gestión proviene de despliegue con orchestrators contenedores que gestionan las diversas instancias y ciclo de vida de cada instancia del contenedor. Romper la aplicación monolítica en subsistemas que se pueden ampliar, desarrollados y desplegados de forma individual es su punto de entrada en el reino de la microservicios.

**La publicación de una sola aplicación envase acoplable al servicio Azure App**

Ya sea porque desea obtener una validación rápida de un contenedor desplegado en Azure o porque la aplicación es simplemente una aplicación de un solo contenedor, Azure App Services proporciona una gran manera de proporcionar los servicios de un solo contenedor escalables.

Azure usando la aplicación de servicio es intuitivo y se puede obtener en funcionamiento rápidamente, ya que proporciona una gran integración Git para tomar su código, construirla en Microsoft Visual Studio, y directamente implementarlo en Azure. Pero, tradicionalmente (sin acoplable), si se necesitan otras capacidades, marcos, o dependencias que no son compatibles con la aplicación de Servicios, que tenía que esperar a que hasta que el equipo azul actualiza esas dependencias en la aplicación de servicio o cambiado a otros servicios como Tela servicio, servicios en la nube, o incluso llanura máquinas virtuales, para los que tiene más control y puede instalar un componente necesario o marco para su aplicación.

Ahora, sin embargo, (anunciado en Microsoft Connect 2016 en noviembre de 2016) y como se muestra en la Figura 4-4, cuando se utiliza Visual Studio 2017, el apoyo de contenedores en Azure App de servicio le da la posibilidad de incluir lo que quiera en su entorno de aplicación. Si ha añadido una dependencia a su aplicación, ya que se está ejecutando en un recipiente, se obtiene la capacidad de incluir esas dependencias en su imagen Dockerfile o acoplable.



**Figura 4-4:** La publicación de un contenedor para Azure App Studio Servicio de Aplicaciones Visuales / contenedores

Figura 4-4 muestra también que el publicar flujo empuja una imagen a través de un registro de contenedor, que puede ser el Registro Azure Container (un registro cerca de sus despliegues en Azure y asegurado por grupos y cuentas de Active Directory Azure) o cualquier otro Registro acoplable como cargador de muelle de concentradores o de registros en las instalaciones.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

Estado y datos en aplicaciones Docker

Una primitiva de contenedores es inmutabilidad. Cuando se compara con una máquina virtual, los contenedores no desaparecen como una ocurrencia común. Una máquina virtual puede fallar en diversas formas de los procesos de muertos, la CPU sobrecargada, o un disco completo o no. Sin embargo, esperamos que la máquina virtual esté disponible y unidades RAID son comunes para asegurar fallos de la unidad mantienen los datos.

Sin embargo, se cree que los contenedores para ser instancias de procesos. Un proceso no mantiene duramadre estado ble. A pesar de que un contenedor puede escribir en su almacenamiento local, suponiendo que esa instancia será de alrededor indefinidamente sería equivalente a suponer una memoria de solo ejemplar que será duradera. Usted debe asumir que los contenedores, al igual que los procesos, se duplican, murieron, o bien, cuando se maneja con un orquestador de contenedores, que se pueden mover.

Acoplable utiliza una característica conocida como un sistema de archivo de superposición para implementar un proceso de copia en escritura que almacena toda la información actualizada al sistema de archivos raíz de un contenedor, en comparación con la imagen original en el que se basa. Estos cambios se perderán si el contenedor se elimina posteriormente del sistema. Un contenedor, por lo tanto, no tiene almacenamiento persistente por defecto. Aunque es posible guardar el estado de un contenedor, el diseño de un sistema en torno a este entraría en conflicto con el principio de la arquitectura de contenedores.

Para gestionar los datos persistentes en aplicaciones acoplables, hay soluciones comunes:

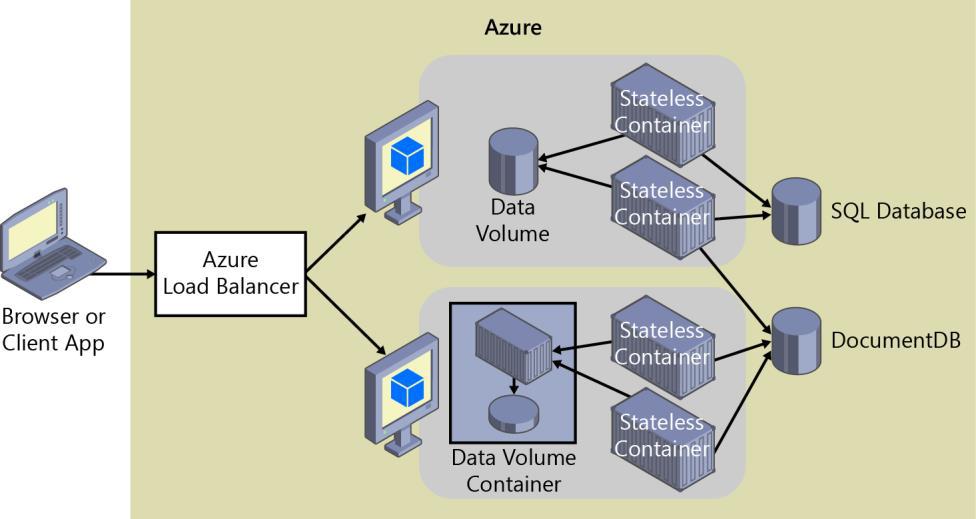
* [**Los volúmenes de datos**](https://docs.docker.com/engine/tutorials/dockervolumes/) Estasmontar al host, como se acaba de señalar.
* [**contenedores de volumen de datos**](https://docs.docker.com/engine/tutorials/dockervolumes/#/creating-and-mounting-a-data-volume-container) Estasproporcionar almacenamiento compartido a través de contenedores, usando un externacontenedor que puede ciclo.
* [**Plugins Volumen**](https://docs.docker.com/engine/tutorials/dockervolumes/#/mount-a-shared-storage-volume-as-a-data-volume) Estasvolúmenes se montan en lugares remotos, proporcionando persistencia a largo plazo.
* **fuentes de datos remotas** Los ejemplos incluyen bases de datos SQL y NO-SQL o servicios de caché como Redis.
* [**Azure Storage**](https://azure.microsoft.com/en-us/documentation/services/storage/) Estaproporciona geo plataforma distribuible como un almacenamiento de servicio (PaaS), proporcionandoel mejor de los envases como persistencia a largo plazo.

Los volúmenes de datos están especialmente designados directorios dentro de uno o más contenedores que no utilizan el [Sistema de archivos Unión.](https://docs.docker.com/v1.8/reference/glossary#union-file-system) Datos volúmenes están diseñados para mantener los datos, independiente de la vida del envase ciclo. Acoplable por lo tanto nunca elimina automáticamente los volúmenes cuando se quita un contenedor, ni va a “basura recoger” los volúmenes que ya no se hace referencia mediante un contenedor. El sistema operativo host puede navegar y editar los datos en cualquier volumen libremente, que es sólo otra razón para utilizar los volúmenes de datos con moderación.

UN [contenedor de volumen de datos](https://docs.docker.com/v1.8/userguide/dockervolumes/) es una mejora con respecto a los volúmenes de datos regulares. Se trata esencialmente de un recipiente en estado latente que tiene uno o más volúmenes de datos creados dentro de ella (como se describe anteriormente). El recipiente de volumen de datos proporciona acceso a los contenedores a partir de un punto de montaje central. La ventaja de este método de acceso es que abstrae la ubicación de los datos originales, por lo que el contenedor de datos de una lógica punto de montaje. También permite que los contenedores de "aplicación" Acceso a los volúmenes de contenedores que los datos se crean y se destruyen mientras que mantiene los datos persistente en un recipiente dedicado.

La figura 4-5 muestra que los volúmenes regulares acoplables se pueden colocar en el almacenamiento de los contenedores mismos pero dentro del servidor host / límites físicos VM. Docker volúmenes no tienen la capacidad de utilizar un volumen de un servidor host / VM a otro.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure



**Figura 4-5:** Los volúmenes de datos y fuentes de datos externas para contenedores de aplicaciones / contenedores

Debido a la incapacidad para gestionar los datos compartidos entre los contenedores que se ejecutan en los servidores físicos separados, se recomienda que no use volúmenes de datos de negocio a menos que el anfitrión del estibador es un host fijo / VM, porque cuando el uso de contenedores acoplables en un orquestador, los contenedores son esperado para ser movido de uno a otro host, dependiendo de las optimizaciones a realizar por el clúster.

Por lo tanto, los volúmenes de datos regulares son un buen mecanismo para trabajar con archivos de seguimiento, archivos temporales, o cualquier concepto similar que no afecte a la consistencia de los datos de negocio si o cuando los contenedores se mueven a través de múltiples hosts.

Volumen de plug-ins como [flocker](https://clusterhq.com/flocker/) proporcionar datos a través de todos los hosts en un clúster. Aunque no todo el volumen de plug-ins se crean igualmente, volumen plug-ins proporcionan típicamente exteriorizan almacenamiento fiable persistente de los contenedores inmutables.

fuentes y cachés como base de datos SQL, DocumentDB, o una memoria caché remoto como Redis datos remotas serían los mismos que en desarrollo sin contenedores. Este es uno de los preferidos, y demostrado, formas de almacenar datos de aplicaciones comerciales.

aplicaciones SOA

SOA era un término usado en exceso y significa muchas cosas diferentes para diferentes personas. Pero, como mínimo, y como denominador común, SOA, o la orientación al servicio, significa que se estructura la arquitectura de su aplicación mediante la descomposición en múltiples servicios (más comúnmente como servicios HTTP) que se pueden clasificar en diferentes tipos como subsistemas o, en otros casos, como niveles.

Hoy en día, se puede implementar esos servicios como contenedores de Docker, que resuelve cuestiones relacionadas con la implementación, porque todas las dependencias están incluidos dentro de la imagen del envase. Sin embargo, cuando se necesita para escalar de salida SOA, puede encontrarse con problemas si está la introducción de servicios en casos individuales. Aquí es donde un software de clustering acoplable o Orchestrator le ayudará. Vamos a ver esto en mayor detalle en la siguiente sección cuando examinamos microservicios enfoques.

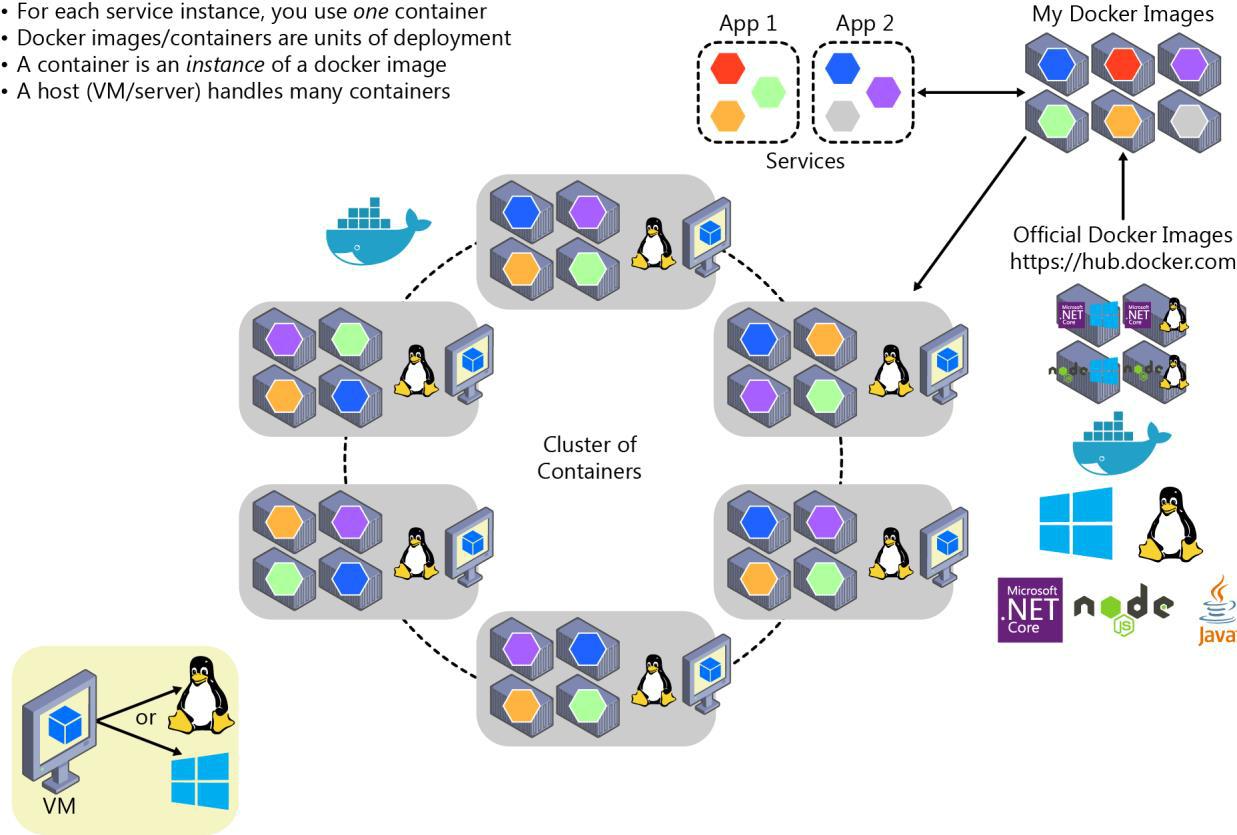
Al final del día, las soluciones de clustering contenedor son útiles tanto para una arquitectura SOA tradicional o para una arquitectura microservicios más avanzada en la que cada microService posee su modelo de datos. Y, gracias a múltiples bases de datos, también se pueden escalar de salida el nivel de datos en lugar de trabajar con bases de datos monolíticas compartidos por los servicios SOA. Sin embargo, la discusión sobre la división de los datos es puramente sobre la arquitectura y el diseño.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

Orquestar microservicios y aplicaciones multicontainer de alta escalabilidad y disponibilidad

Usando orchestrators para aplicaciones listas para la producción es esencial si su aplicación se basa en microservicios o simplemente dividen en varios contenedores. Como previamente introducido, en un enfoque basado en microService, cada microService posee su modelo y los datos de modo que será autónoma desde el punto de vista del desarrollo y despliegue. Pero incluso si usted tiene una aplicación más tradicional que se compone de múltiples servicios (como SOA), que también tendrá varios contenedores o servicios que comprenden una sola aplicación empresarial que necesitan ser desplegado como un sistema distribuido. Este tipo de sistemas son complejos de escalar y gestionar; Por lo tanto, es absolutamente necesario un orquestador, si usted quiere tener una aplicación multicontainer listo para la producción y escalable.

Figura 4-6 ilustra el despliegue en un clúster de una aplicación compuesta de múltiples microservicios (recipientes).



**Figura 4-6:** Un grupo de contenedores

Se ve como un enfoque lógico. Pero, ¿cómo está manejando el equilibrio de carga, enrutamiento y la orquestación de estas aplicaciones compuestas?

La interfaz de comandos -line Docker (CLI) satisface las necesidades de la gestión de un contenedor en un host, pero se queda corto cuando se trata de la gestión de múltiples contenedores desplegados en varios hosts para aplicaciones distribuidas más complejas. En la mayoría de los casos, se necesita una plataforma de gestión que se iniciará automáticamente contenedores, suspenderlos o cerrarlas cuando sea necesario, e idealmente también controlar su acceso a recursos como la red y almacenamiento de datos.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

Para ir más allá de la gestión de los envases individuales o aplicaciones compuestas muy simples y avanzar hacia las aplicaciones empresariales más grandes con microservicios, debe activar a las plataformas de orquestación y clustering.

Desde el punto de vista de la arquitectura y el desarrollo, si usted está construyendo grandes, empresa, microservicios-basada, aplicaciones, es importante comprender las siguientes plataformas y productos que soportan escenarios avanzados:

* **Clusters y orchestrators** Cuando tenga que escalar de salida aplicaciones a través de muchos anfitriones acoplables, como con una aplicación basada en microservicios grandes, es fundamental para poder gestionar todos los hosts como un solo grupo mediante la abstracción de la complejidad de la plataforma subyacente. Eso es lo que los grupos de contenedores y orchestrators proporcionan. Ejemplos de orchestrators son acoplable Swarm, Mesosphere CC / CA, Kubernetes (los tres primeros disponible a través de Azure Container Service), y Azure Servicio de Tela.
* **programadores** *programación*significa tener la capacidad de un administrador para lanzar contenedores en un clúster de modo que también proporcionan una interfaz de usuario. Un planificador de clúster tiene varias responsabilidades: a utilizar los recursos del cluster de manera eficiente, para establecer las limitaciones impuestas por el usuario, para equilibrar la carga de manera eficiente los contenedores en los nodos o los ejércitos, ya sea robusto frente a errores al tiempo que proporciona una alta disponibilidad.

Los conceptos de un grupo y un programador están estrechamente relacionados, por lo que los productos suministrados por diferentes proveedores a menudo proporcionan ambos conjuntos de capacidades. La Tabla 4-1 enumera las plataformas y software más importantes opciones que tiene para los clústeres y programadores. Estas agrupaciones se ofrecen generalmente en las nubes públicas como Azure.

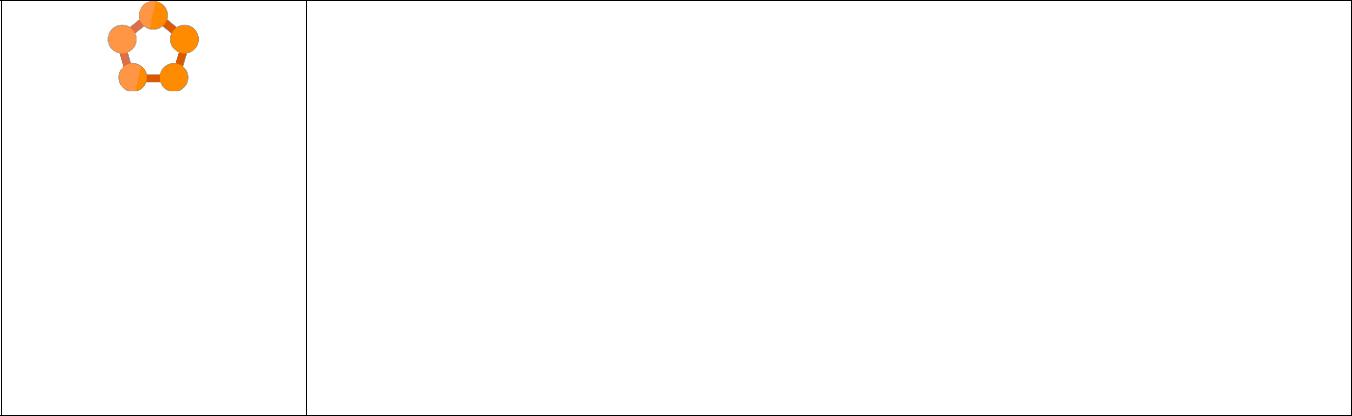
**Tabla 4-1:** Las plataformas de software para la agrupación de contenedores, orquestación, y la programación

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Plataforma** |  |  | **Descripción** | | | |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| acoplable Swarm | | |  | Acoplable Swarm le da la posibilidad de agrupar y programar ventana acoplable | | | | |  |
|  |  |  |  | contenedores. Mediante el uso de Swarm, puede activar un conjunto de Docker aloja en una | | | | |  |
|  |  |  |  | solo, anfitrión acoplable virtual. Los clientes pueden realizar solicitudes de API a pulular en | | | | |  |
|  |  |  |  | de la misma manera que lo hacen a los hosts, lo que significa que Swarm hace que sea fácil para | | | | |  |
|  |  |  |  | las aplicaciones escalar a varios hosts. | | | | |  |
|  |  |  |  | Acoplable Swarm es un producto de la ventana acoplable, la compañía. | | | | |  |
|  |  |  |  | Acoplable v1.12 o posterior pueden ejecutar nativa y construido en el Modo Swarm. | | | | |  |
|  | | |  |  |  |  |  | |  |
| Mesosfera CC / CA | | |  | Mesosfera Empresa CC / CA (basado en Apache mesos) es un producción- | | | | |  |
|  |  |  |  | plataforma lista para el funcionamiento de los contenedores y las aplicaciones distribuidas. | | | | |  |
|  |  |  |  | CC / CA funciona mediante la abstracción de una colección de los recursos disponibles en el | | | | |  |
|  |  |  |  | racimo y para hacer que estos recursos disponibles para los componentes construidos en la parte superior de | | | | |  |
|  |  |  |  | eso. Maratón se utiliza por lo general como un planificador integrado con CC / CA. | | | | |  |
|  | | |  |  |  |  |  | |  |
| google Kubernetes | | |  | Kubernetes es un producto de código abierto que proporciona funcionalidad que | | | | |  |
|  |  |  |  | rangos de infraestructura de cluster y la programación recipiente para orquestar | | | | |  |
|  |  |  |  | capacidades. Con él, se puede automatizar el despliegue, la escala y las operaciones | | | | |  |
|  |  |  |  | de contenedores de aplicaciones a través de grupos de hosts. | | | | |  |
|  |  |  |  | Kubernetes proporciona una infraestructura recipiente centrado en que los grupos | | | | |  |
|  |  |  |  | contenedores de aplicaciones en unidades lógicas para una fácil gestión y descubrimiento. | | | | |  |
|  | | |  |  | | | | |  |
| Azure Servicio Tela | | |  | [Tela servicio](https://azure.microsoft.com/en-us/documentation/articles/service-fabric-overview/) es una plataforma microservicios Microsoft para crear aplicaciones. | | | | |  |
|  |  |  |  |  |  |  |  | |  |
|  |  |  |  | Es un [orquestador](https://azure.microsoft.com/en-us/documentation/articles/service-fabric-cluster-resource-manager-introduction/) de los servicios y crea grupos de máquinas. Por defecto, | | | | |  |



1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure





Servicio Tela despliega y activa los servicios como procesos, pero el servicio de la tela puede desplegar servicios en imágenes de contenedores acoplables. Más importante aún, se pueden mezclar los servicios en los procesos con los servicios de contenedores en la misma aplicación.

A partir de mayo de 2017, la característica del servicio de tela que soporta el despliegue de servicios como contenedores de Docker se encuentra en estado de vista previa.

Puede desarrollar servicios Servicio de telas en muchos aspectos, el uso de la

[Tela modelos de programación de servicios](https://azure.microsoft.com/en-us/documentation/articles/service-fabric-choose-framework/) a el despliegue [ejecutables invitados](https://azure.microsoft.com/documentation/articles/service-fabric-deploy-existing-app/) también

como contenedores. Tela servicio es compatible con los modelos de aplicación como prescriptivos

[servicios con estado](https://azure.microsoft.com/en-us/documentation/articles/service-fabric-reliable-services-introduction/) y [Los actores fiables.](https://azure.microsoft.com/en-us/documentation/articles/service-fabric-reliable-actors-introduction/)

**Usando orchestrators a base de contenedores en Azure**

Varios proveedores de nube ofrecen apoyo acoplables contenedores más Docker clusters y de apoyo, incluyendo la orquestación Azure, Amazon EC2 Servicio de contenedor, y Google Container Engine. Azure proporciona clúster acoplable y apoyo Orchestrator través de Azure Servicio de contenedor, como se explica en la siguiente sección.

Otra opción es utilizar Azure Servicio de Tela, que también es compatible con acoplable basado en Linux y contenedores de Windows. Tela servicio se ejecuta en Azure o cualquier otra nube, así como[En las instalaciones.](https://azure.microsoft.com/en-us/documentation/articles/service-fabric-deploy-anywhere/)

**Utilización del servicio de contenedores Azure**

Estibador piscinas racimo múltiples hosts acoplables y los expone como un único host acoplable virtual, por lo que puede implementar varios contenedores en el clúster. El grupo se encargará de toda la gestión compleja de plomería como la escalabilidad y la salud. Figura 4-7 representa cómo un cúmulo acoplable para aplicaciones compuestas se asigna al servicio de contenedores.

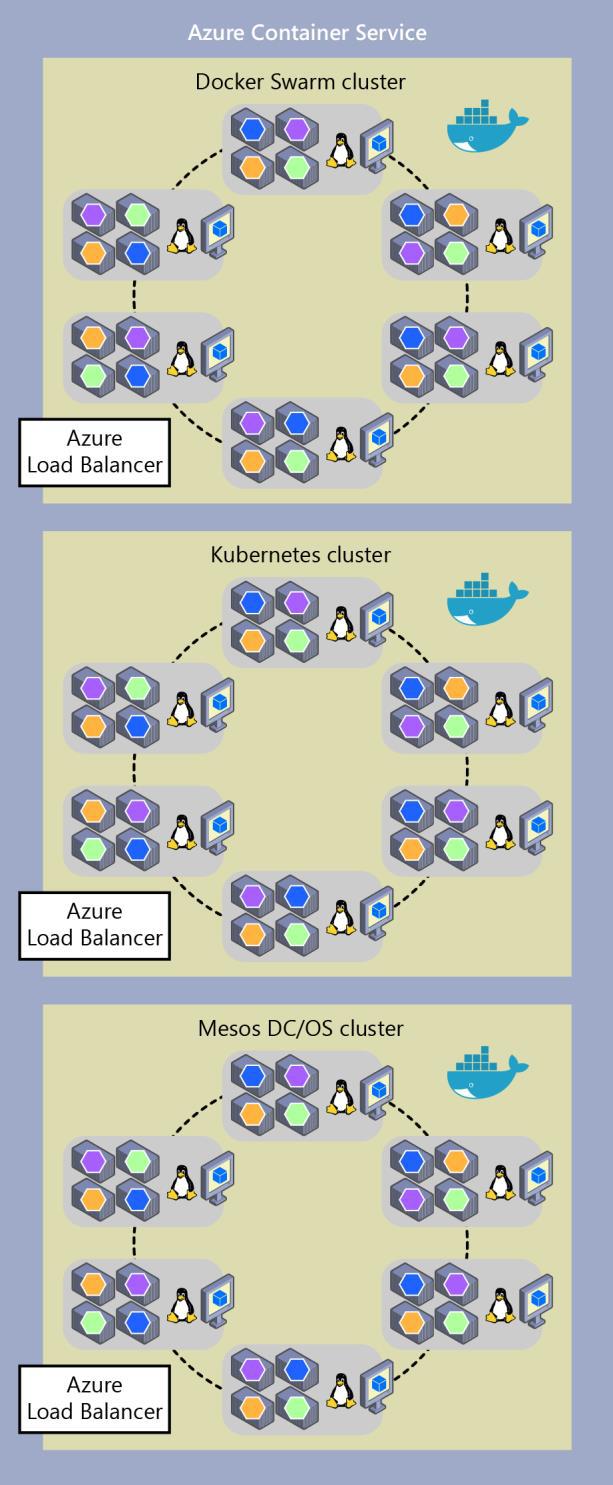
Servicio de contenedores proporciona una manera de simplificar la creación, configuración yla gestión de un grupo de máquinas virtuales que están preconfigurados para ejecutar aplicaciones en contenedores. El uso de una configuración optimizada de herramientas de programación de código abierto y orquestación populares, Container Service le da la capacidad de utilizar sus habilidades existentes o dibujar en un cuerpo grande y creciente de conocimientos de la comunidad para desplegar y gestionar aplicaciones basadas en contenedor en Azure.

Servicio de contenedores optimiza la configuración de las herramientas de código abierto muy popular agrupación acoplable y tecnologías específicamente para Azure. Se obtiene una solución abierta que ofrece portabilidad tanto para sus envases y configuración de la aplicación. Seleccionar el tamaño, el número de los ejércitos, y las herramientas Orchestrator y servicio de contenedores se encarga de todo lo demás.

Servicio de contenedor utiliza imágenes acoplables para asegurar que sus contenedores de aplicaciones son totalmente portátil. Es compatible con su elección de plataformas de orquestación de código abierto como CC / CA, Kubernetes y acoplable Swarm para asegurar que estas aplicaciones pueden escalar a miles o incluso decenas de miles de contenedores.

Con Azure Container Service, usted puede tomar ventaja de las características de nivel empresarial de Azure, manteniendo la portabilidad de aplicaciones, incluso en las capas de orquestación.

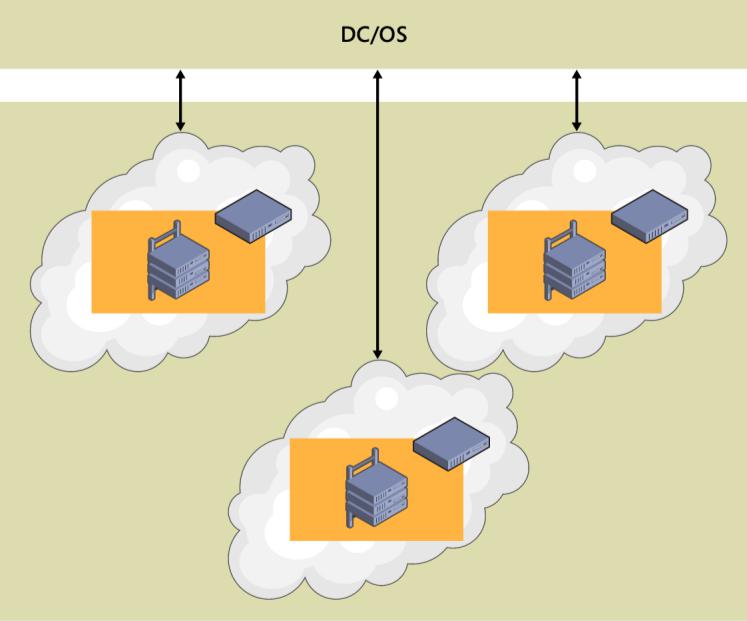
1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure



**Figura 4-7:** opciones de agrupamiento en Azure Container Service

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

Como se muestra en la Figura 4-8, Servicio de contenedor es simplemente la infraestructura proporcionada por Azure con el fin de desplegar CC / CA, Kubernetes o acoplable Swarm, pero no implementa ninguna orquestador adicional. Por lo tanto, Contenedor de servicios no es un orquestador, como tal; es sólo una infraestructura que se aprovecha de orchestrators de código abierto existentes para contenedores.

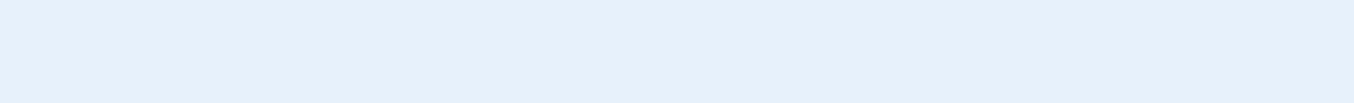


**Figura 4-8:** Orchestrators en Container Service

Desde una perspectiva de uso, el objetivo del Servicio de contenedores es proporcionar un entorno de alojamiento de contenedores mediante el uso de herramientas y tecnologías de código abierto. Con este fin, expone los puntos finales de API estándar de su orquestador elegido. Mediante el uso de estos criterios de valoración, se puede utilizar cualquier software que pueda comunicarse a aquellos parámetros. Por ejemplo, en el caso del punto final del estibador Swarm, es posible optar por utilizar la ventana acoplable CLI. Para DC / OS, se puede optar por utilizar la CC / CA CLI.

**Introducción al Servicio de contenedores**

Para comenzar a utilizar Contenedor de servicios, se implementa un clúster de servicio de contenedores desde el portal Azure mediante el uso de una plantilla de Resource Manager o el Azure [CLI.](https://azure.microsoft.com/documentation/articles/xplat-cli-install/) plantillas disponibles incluyen [acoplable Swarm,](https://github.com/Azure/azure-quickstart-templates/tree/master/101-acs-swarm) [Kubernetes,](https://github.com/Azure/azure-quickstart-templates/tree/master/101-acs-kubernetes) y [CC / CA.](https://github.com/Azure/azure-quickstart-templates/tree/master/101-acs-dcos) Tú puede modificar las plantillas de inicio rápido para incluir adicional o avanzado configuración Azure.



**Más información** Para obtener más información acerca de cómo implementar un cluster Container Service, en la página web Azure, leer [Implementar un clúster Azure Container Service.](https://azure.microsoft.com/en-us/documentation/articles/container-service-deployment/)

No hay cuotas para cualquier software instalado por defecto como parte de ACS. Todas las opciones por defecto se implementan con el software de código abierto.

Servicio de contenedores se encuentra actualmente disponible para el estándar A, D, DS, G, y la serie GS máquinas virtuales Linux en Azure. Se le cobrará sólo para las instancias de proceso que decide, así como los demás recursos de infraestructura subyacentes que se consumen, como el almacenamiento y redes. No hay cargos adicionales para el propio Servicio de contenedores.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

**Recursos adicionales**

Los siguientes son lugares donde se puede encontrar información adicional:

* Introducción al contenedor acoplable soluciones de alojamiento con servicio de contenedores: <https://azure.microsoft.com/documentation/articles/container-service-intro/>
* visión general del estibador enjambre: <https://docs.docker.com/swarm/overview/>
* Vista general del modo enjambre: <https://docs.docker.com/engine/swarm/>
* Mesosfera CC / CA Descripción general: <https://docs.mesosphere.com/1.7/overview/>
* Kubernetes (sitio oficial): <http://kubernetes.io/>

**Utilización del servicio de Tela**

Servicio Tela surgió de la transición de Microsoft a partir de la entrega de productos de “caja”, que eran normalmente monolítica en el estilo, a la prestación de servicios. La experiencia de la construcción y operación de grandes servicios a escala, como la base de datos SQL Azure, Azure documento DB, Azure Service Bus o Cortana del backend, en forma de tela de servicio. La plataforma ha evolucionado con el tiempo a medida que más y más servicios adoptaron. Es importante destacar que, Servicio de Tela tuvo que correr no sólo en Azure, sino también en las implementaciones independientes de Windows Server.

El objetivo del Servicio de Tela es resolver los difíciles problemas de la construcción y funcionamiento de un servicio y la utilización de los recursos de infraestructura eficiente para que los equipos puedan resolver problemas de negocio utilizando un enfoque microservicios.

Tela servicio ofrece dos amplias áreas para ayudar a crear aplicaciones que utilizan un enfoque microservicios:

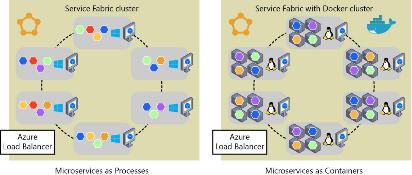
* Una plataforma que proporciona servicios de sistema a implementar, escala, actualizar, detectar y reiniciar los servicios fallidos, descubrir la ubicación del servicio, administrar el estado, y monitorear la salud. Estos servicios de sistema en efecto proporcionan muchas de las características de microservicios descritos previamente.
* API de programación, o marcos, para ayudar a crear aplicaciones como microservicios: [actores fiables](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-choose-framework) [y servicios confiables.](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-choose-framework) De Por supuesto, usted puede elegir cualquier código para construir su microService, pero estos API hacen el trabajo más sencillo, y se integran con la plataforma a un nivel más profundo. De esta manera se puede obtener información sobre la salud y el diagnóstico, o puede tomar ventaja de la administración del estado fiable.

Servicio tela es agnóstico con respecto a la forma de construir su servicio, y se puede utilizar cualquier tecnología.

Sin embargo, proporciona una función de API de programación que hacen que sea más fácil de construir microservicios.

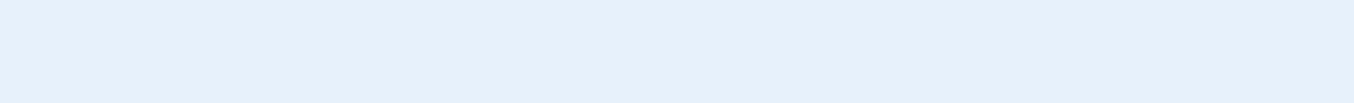
La figura 4-9 muestra cómo se puede crear y ejecutar microservicios en servicio, ya sea Tela procesos tan simple o tan contenedores de Docker. También es posible mezclar microservicios basadas en contenedor con microservicios basados ​​en procesos dentro de la misma agrupación Servicio Tela.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure



**Figura 4-9:** Implementación de microservicios como procesos o como contenedores en Azure Servicio Tela

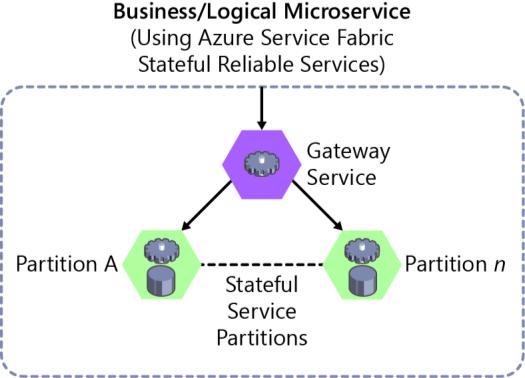
racimos de tela de servicios basados ​​en anfitriones Linux y Windows pueden ejecutar contenedores acoplable Linux y contenedores de Windows.



**Más información** Para arriba-a -Fecha información sobre el soporte contenedores en servicio Tela, en el sitio web Azure, leer [Tela servicio y contenedores.](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-containers-overview)

Servicio de Tela es un buen ejemplo de una plataforma con la que se puede definir una arquitectura diferente lógica (microservicios negocios o Limitado Contextos) que la implementación física. Por ejemplo, si se implementa[Servicios fiables con estado](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-services-introduction) en [Azure Servicio Tela,](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-overview) que se explican en la siguiente sección, “[Sin estado con estado frente microservicios](#page36),”Usted tiene un concepto microService negocios con múltiples servicios físicos.

Como se muestra en la figura 4-10, y pensando desde una perspectiva lógica microService / negocio, al implementar un servicio confiable servicio Tela con estado, por lo general tendrá que aplicar dos capas de servicios. El primero es el servicio confiable con estado de back-end, que maneja varias particiones. El segundo es el servicio de extremo frontal, o servicio de puerta de enlace, a cargo de enrutamiento y agregación de datos a través de múltiples particiones o instancias de servicio con estado. Que el servicio de puerta de enlace también se encarga de la comunicación del lado del cliente con repetición de bucles de acceso al servicio de back-end se utiliza conjuntamente con el Servicio de Tela[proxy inverso.](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reverseproxy)



**Figura 4-10:** microService negocios con varios servicios de estado y sin estado en servicio Tela

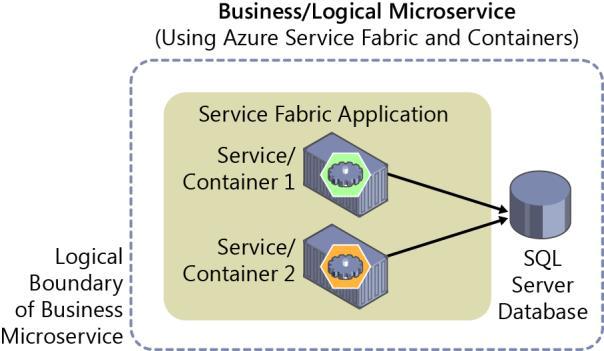
En cualquier caso, cuando se utiliza con estado de servicio Tela servicios confiables, también tiene una lógica microService o negocio (Contexto Delimitado) que por lo general se compone de múltiples servicios físicos. Cada uno de ellos, el servicio de puerta de enlace, y el servicio de reparto podrían ser implementadas como servicios Web ASP.NET API, como se muestra en la figura 4-10.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

En Servicio de Tela, puede agrupar e implementar grupos de servicios como una [Tela servicio de aplicaciones,](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-application-model) que es la unidad de envasado y de despliegue para la Orchestrator o clúster. Por lo tanto, la aplicación de servicio de tela podría ser asignada a este negocio autónomo y microService límite lógico o Contexto Delimitado, también.

**Tela servicio y contenedores**

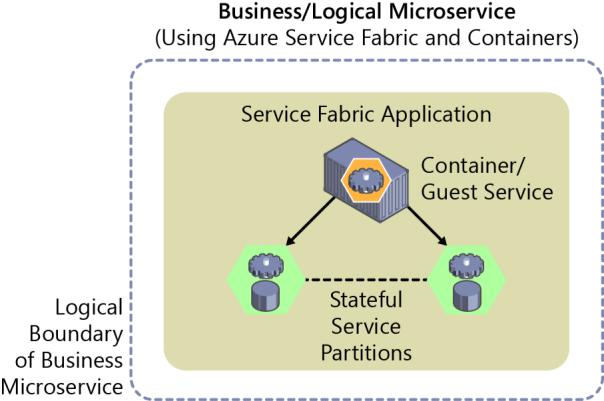
Con respecto a los contenedores en servicio de la tela, también se puede implementar servicios de imágenes de contenedores dentro de un grupo de servicio de tela. Figura 4-11 ilustra que la mayoría de las veces sólo habrá un contenedor por servicio.



**Figura 4-11:** microService negocios con varios servicios (contenedores) en el Servicio de Tela

Sin embargo, los llamados contenedores “sidecar” (dos contenedores que deben ser desplegados juntos como parte de un servicio lógico) también son posibles en servicio de la tela. Lo importante es que una microService negocio es el límite lógico en torno a varios elementos de cohesión. En muchos casos, puede ser que sea un único servicio con un único modelo de datos, pero en otros casos es posible que tenga varios servicios físicos, también.

Al escribir estas líneas (abril de 2017), en el Servicio de Tela no se puede implementar con estado SF Servicios fiables sobre contenedores-puede implementar únicamente los envases sin Estado huésped, servicios, o servicios actor en contenedores. Pero tenga en cuenta que puede mezclar los servicios en los procesos y servicios de contenedores en la misma aplicación de servicio de tela, como se muestra en la figura 4-12.



**Figura 4-12:** microService negocio asigna a una aplicación de servicio de tela con contenedores y servicios con estado

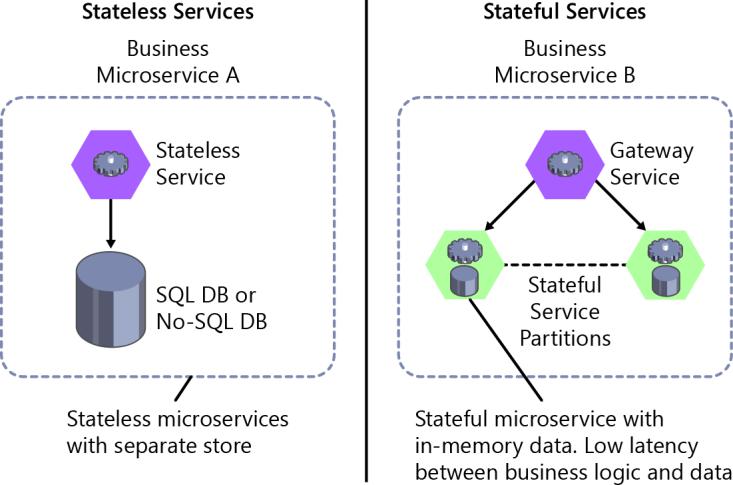
El apoyo es también diferente dependiendo de si está utilizando recipientes Docker en Linux o contenedores de Windows. Soporte para contenedores en servicio de la tela va a ampliar en las próximas versiones. Para arriba-hasta la fecha actualizada sobre la compatibilidad de contenedores en servicio Tela, en el sitio web Azure, leer[Tela servicio y](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-containers-overview) [contenedores.](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-containers-overview)

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

**Sin estado con estado frente microservicios**

Como se mencionó anteriormente, cada microService (Contexto Limita lógica) debe poseer su modelo de dominio (de datos y lógica). En el caso de microservicios sin estado, las bases de datos serán externos, empleando opciones relacionales como opciones NoSQL como MongoDB o Azure DocumentDB SQL Server o.

Pero los propios servicios también pueden ser de estado, lo que significa que los datos residen en el microService. Estos datos pueden existir no sólo en el mismo servidor, pero dentro del proceso de microService, en la memoria, y persistió en las unidades y se replica a otros nodos. Figura 4-13 muestra los diferentes enfoques.



**Figura 4-13:** Sin estado con estado frente microservicios

Un enfoque sin estado es perfectamente válido y es más fácil de implementar que microservicios con estado, porque el enfoque es similar a los patrones tradicionales y conocidas. Pero sin estado microservicios imponen latencia entre el proceso y fuentes de datos. También implican más piezas en movimiento cuando se está tratando de mejorar el rendimiento de la memoria caché y las colas adicional. El resultado es que se puede terminar con arquitecturas complejas que tienen demasiados niveles.

A diferencia de, [microservicios con estado](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-services-introduction#when-to-use-reliable-services-apis) puede sobresalir en escenarios avanzados porque no hay latencia entre la lógica de dominio y datos. procesamiento de datos pesados, juegos back-ends, bases de datos como un servicio, y otros escenarios de baja latencia todos se benefician de los servicios con estado, que proporcionan el estado local para un acceso más rápido.

servicios sin estado y con estado son complementarios. Por ejemplo, un servicio de estado podría dividirse en varias particiones. Para acceder a esas particiones, es posible que necesite un servicio sin estado actuando como un servicio de puerta de enlace que sabe cómo hacer frente a cada partición basado en claves de partición.

servicios con estado tienen inconvenientes. Imponen un nivel de complejidad que les permite escalar. Funcionalidad que normalmente se pondría en práctica por los sistemas de base de datos externa debe abordarse para tareas tales como la replicación de datos a través microservicios con estado y la partición de datos. Sin embargo, esta es una de las áreas en las que la orquestación como[Tela servicio](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-services-platform-architecture) con su [servicios fiables con estado](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-services-introduction#when-to-use-reliable-services-apis) puede ayudar a los más-simplificando el desarrollo y el ciclo de vida de microservicios con estado utilizando el [De confianza](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-work-with-reliable-collections) [servicios API](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-work-with-reliable-collections) y [Los actores fiables.](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-actors-introduction)

Otros marcos MICROSERVICE que permiten a los servicios con estado, que apoyan el patrón Actor, y que mejoran la tolerancia a fallos y la latencia entre la lógica de negocio y los datos son Microsoft [Orleans,](https://github.com/dotnet/orleans) de Microsoft Research, y [Akka.NET.](http://getakka.net/) Ambos marcos están mejorando actualmente su apoyo a acoplable.

Tenga en cuenta que los contenedores de Docker son ellos mismos sin estado. Si se desea implementar un servicio de estado, se necesita uno de los marcos normativos y de nivel superior adicionales señaladas anteriormente. Sin embargo, a partir de este

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

la escritura, los servicios con estado en servicio Tela no se admiten como contenedores, sólo como microservicios lisos.

el soporte de servicios fiables en contenedores estará disponible en las próximas versiones de servicio Tela.

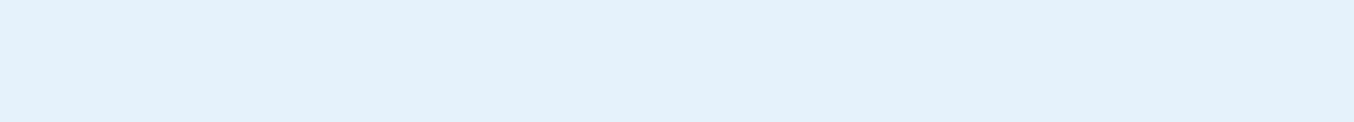
Entorno de desarrollo para aplicaciones Docker

**Herramientas de desarrollo de opciones: IDE o editor**

No importa si usted prefiere un IDE completo y potente o un editor ligero y ágil, Microsoft tiene cubierto cuando se trata de desarrollar aplicaciones acoplables.

**(Herramientas multiplataforma para Mac, Linux y Windows) de código de Visual Studio y estibador de la CLI**

Si prefiere un editor ligero, multi-plataforma de apoyo cualquier lenguaje de desarrollo, puede utilizar código de Visual Studio y acoplable CLI. Estos productos proporcionan una experiencia sencilla y robusta, que es fundamental para agilizar el flujo de trabajo del desarrollador. Mediante la instalación de “acoplable para Mac” o “acoplable para Windows” (entorno de desarrollo), los desarrolladores acoplables pueden usar un solo acoplable CLI para construir aplicaciones para Windows o Linux (entorno de tiempo de ejecución). Además, código de Visual Studio es compatible con las extensiones para acoplable con IntelliSense para Dockerfiles y de acceso directo-tareas a ejecutar los comandos del estibador del editor.

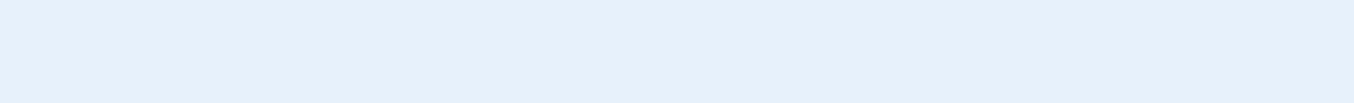


**Nota** Para descargar código de Visual Studio, visite [https://code.visualstudio.com/download.](https://code.visualstudio.com/download)

Para descargar acoplable para Mac y Windows, vaya a [http://www.docker.com/products/docker.](http://www.docker.com/products/docker)

**Visual Studio con herramientas acoplables**

Cuando se está utilizando Visual Studio 2015 puede instalar el add-on “Herramientas herramientas acoplables para Visual Studio.” Para Visual Studio 2017, Docker herramientas vienen construidos en ya. En ambos casos se puede desarrollar, ejecutar y validar sus aplicaciones directamente en el entorno del estibador elegido. F5 su aplicación (solo contenedor o varios contenedores) directamente en un huésped acoplable con la depuración, o pulse Ctrl + F5 para editar y actualizar su aplicación sin tener que reconstruir el recipiente. Esta es la opción más simple y más potente para los desarrolladores de Windows que crean contenedores acoplables para Linux o Windows.



**Nota** Para descargar Docker Herramientas para Visual Studio, visite [https://visualstudiogallery.msdn.microsoft.com/0f5b2caa-ea00-41c8-b8a2-058c7da0b3e4.](https://visualstudiogallery.msdn.microsoft.com/0f5b2caa-ea00-41c8-b8a2-058c7da0b3e4)

**opciones de idioma y marco**

Puede desarrollar aplicaciones acoplables y herramientas de Microsoft con la mayoría de las lenguas modernas. La siguiente es una lista inicial, pero no se limitan a la misma:

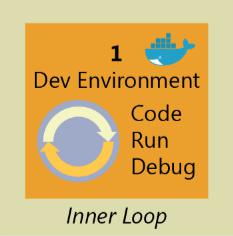
* **.**NET y ASP.NET Core Core
* Node.js
* Golang
* Java
* Rubí
* Pitón

Básicamente, se puede utilizar cualquier lenguaje moderno apoyado por Estibador en Linux o Windows.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

flujo de trabajo de desarrollo de lazo interno para aplicaciones Docker

Antes de la activación del flujo de trabajo de bucle exterior que abarca el ciclo entero DevOps, todo comienza en la máquina de cada desarrollador, la codificación de la aplicación en sí, usando sus lenguajes o plataformas preferidas, y las pruebas de forma local (Figura 4 -14). Pero en todos los casos, tendrá un punto muy importante en común, sin importar el idioma, el marco, o plataformas que elija. En este flujo de trabajo específico, siempre se están desarrollando y probando contenedores estibador, pero localmente.



**Figura 4-14:** contexto de desarrollo-bucle interno

El recipiente o ejemplo de una imagen de acoplable de contendrán estos componentes:

* Una selección del sistema operativo (por ejemplo, una distribución de Linux o Windows)
* Archivos agregados por el promotor (por ejemplo, los binarios de aplicaciones)
* Configuración (por ejemplo, la configuración del entorno y dependencias)
* Las instrucciones para qué procesos a cargo de Docker

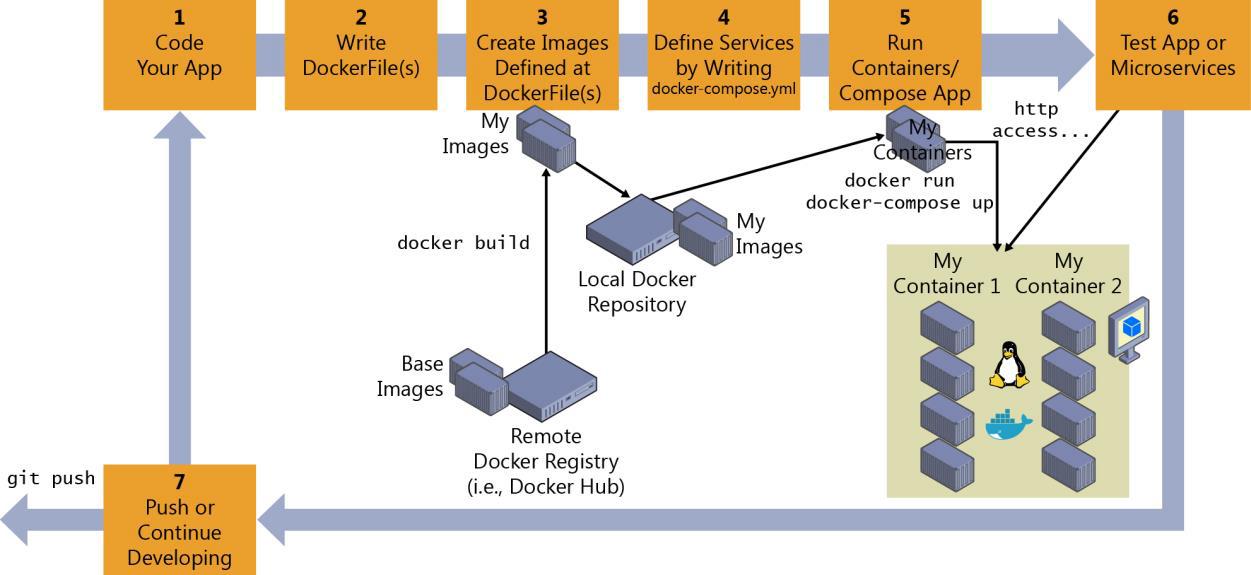
Puede configurar el flujo de trabajo de desarrollo de bucle interior que utiliza acoplable como el proceso (descrito en la siguiente sección). Tener en cuenta que los pasos iniciales para configurar el entorno no está incluido, porque hay que hacerlo sólo una vez.

**La construcción de una sola aplicación dentro de un contenedor acoplable utilizando código de Visual Studio y acoplable CLI**

Las aplicaciones se realizan a partir de sus propios servicios, además de bibliotecas adicionales (dependencias).

La figura 4-15 muestra los pasos básicos que normalmente necesita para llevar a cabo cuando se construye una aplicación acoplable, seguida de una descripción detallada de cada paso.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure



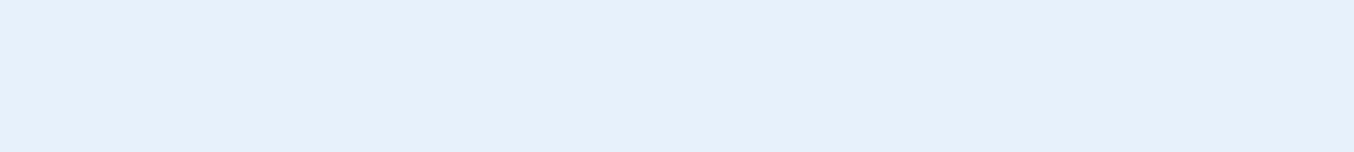
**Figura 4-15:** flujo de trabajo de alto nivel para el ciclo de vida de aplicaciones en contenedores Docker usando acoplable CLI

**Paso 1: Iniciar la codificación en código de Visual Studio y crear su aplicación inicial de referencia / servicio**

La forma en que desarrolla su aplicación es bastante similar a la forma en que lo hace sin estibador. La diferencia es que mientras que el desarrollo, va a implementar y probar su aplicación o servicios que se ejecutan dentro de contenedores de Docker colocados en su entorno local (como una máquina virtual Linux o Windows).

**Configuración de su entorno local**

Con las últimas versiones de estibador para Mac y Windows, es más fácil que nunca para desarrollar aplicaciones acoplables, y la configuración es sencilla.

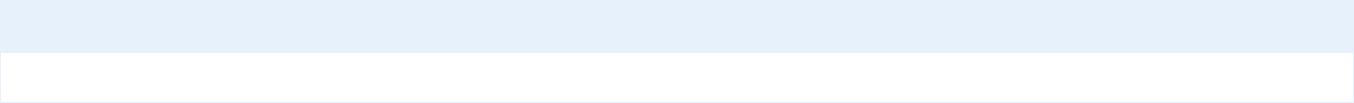


**Más información** Para obtener instrucciones sobre la configuración de estibador para Windows, vaya a [https://docs.docker.com/](https://docs.docker.com/docker-for-windows/)[cargador de muelle-para-ventanas /.](https://docs.docker.com/docker-for-windows/)

Para obtener instrucciones sobre la configuración de estibador para Mac, vaya a [https://docs.docker.com/docker-for-mac/.](https://docs.docker.com/docker-for-mac/)

Además, se necesita un editor de código para que en realidad se puede desarrollar su aplicación durante el uso del estibador CLI.

Microsoft proporciona código de Visual Studio, que es un editor de código ligero que se utiliza en Mac, Windows y Linux, y proporciona IntelliSense con [soporte para varios idiomas](https://code.visualstudio.com/docs/languages/overview) (JavaScript, .NET, Go, Java, Ruby, Python, y la mayoría de las lenguas modernas), [depuración,](https://code.visualstudio.com/Docs/editor/debugging) [integración con Git](https://code.visualstudio.com/Docs/editor/versioncontrol) y [extensiones](https://code.visualstudio.com/docs/extensions/overview) [apoyo.](https://code.visualstudio.com/docs/extensions/overview) Esta editor es un gran ajuste para los desarrolladores de Mac y Linux. En Windows, también se puede utilizar todo el aplicación de Visual Studio.



**Más información** Para obtener instrucciones sobre la instalación de Visual Studio para Windows, Mac o Linux, vaya a [http://code.visualstudio.com/docs/setup/setup-overview/https://docs.docker.com/docker-for-mac/.](http://code.visualstudio.com/docs/setup/setup-overview/https:/docs.docker.com/docker-for-mac/)

Puede trabajar con acoplable CLI y escribir el código utilizando cualquier editor de código, pero si se utiliza código de Visual Studio, que hace que sea fácil al autor Dockerfile y archivos ventana acoplable-compose.yml en su espacio de trabajo. Además, puede ejecutar tareas desde el IDE que le pedirá secuencias de comandos que se pueden ejecutar operaciones elaborados utilizando CLI acoplable por debajo de código de Visual Studio.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

Código Visual Studio es proporcionada por una extensión, que tendrá que instalar. Para ello, pulse Ctrl + Shift + P, el tipo ext instale y ejecute las Extensiones: Instalar extensión de comandos para llevar

la lista de extensiones del mercado. A continuación, tipo ventana acoplable para filtrar los resultados y, a continuación, seleccionar el Dockerfile Y acoplable Componer extensión de archivo de soporte (yml), tal como se representa en la figura 4-16.



**Figura 4-16:** Instalación de la extensión del estibador en el código de Visual Studio

**Paso 2: Crear un DockerFile relacionada con una imagen existente (OS liso o dev ambientes como .NET Core, Node.js, y Ruby)**

Usted necesitará un DockerFile imagen personalizada para ser construido por y por contenedor para ser desplegado, por lo tanto, si su aplicación se compone de un único servicio personalizado, se necesitará un solo DockerFile. Pero, si su aplicación se compone de múltiples servicios (como en una arquitectura microservicios), tendrá una Dockerfile por servicio.

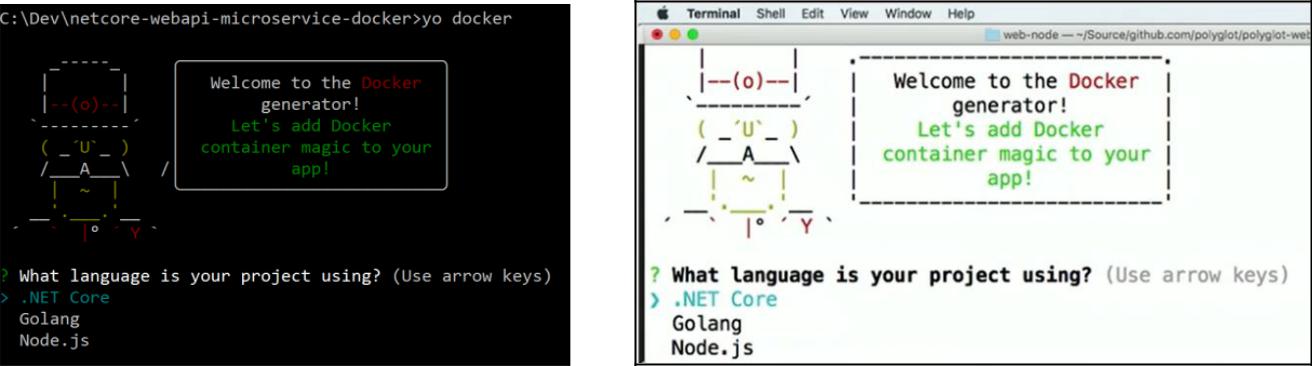
El DockerFile generalmente se coloca dentro de la carpeta raíz de la aplicación o servicio y contiene los mandatos necesarios para que acoplable sabe cómo configurar y ejecutar esa aplicación o servicio. Puede crear su DockerFile y añadirlo a su proyecto junto con su código (Node.js, .NET Core, etc.), o, si usted es nuevo en el medio ambiente, echar un vistazo a la siguiente Tip.



**Propina** Se puede utilizar una herramienta de línea de comandos llamada [yo ventana acoplable,](https://github.com/Microsoft/generator-docker) **cual** andamios archivos de su proyecto en el idioma que elija y añade los archivos de configuración del Docker requeridos. Básicamente, para ayudar a los desarrolladores empezar, se crea la DockerFile apropiada, -compose.yml estibador, y otras secuencias de comandos asociados a la construcción y explotación de los contenedores de Docker. Este generador Yeoman le pedirá que con un par de preguntas, preguntando a su lenguaje de desarrollo y contenedor de destino huésped seleccionado.

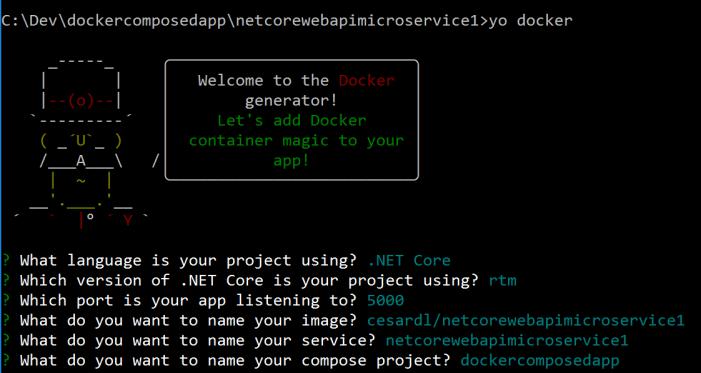
Por ejemplo, la Figura 4-17 muestra dos capturas de pantalla de los terminales en Windows y en Mac, en ambos casos, corriendo yo estibador, que generarán los proyectos de código muestra (actualmente .NET Core, Golang y Node.js como idiomas soportados ) ya configurada para trabajar en la parte superior de la ventana acoplable.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure



**Figura 4-17:** yo herramienta de comando ventana acoplable en Windows (izquierda) y en un Mac

Figura 4-18 presenta un ejemplo usando yo cargador de muelle después de haber un proyecto Core .NET existentes en el lugar para servir de andamiaje.



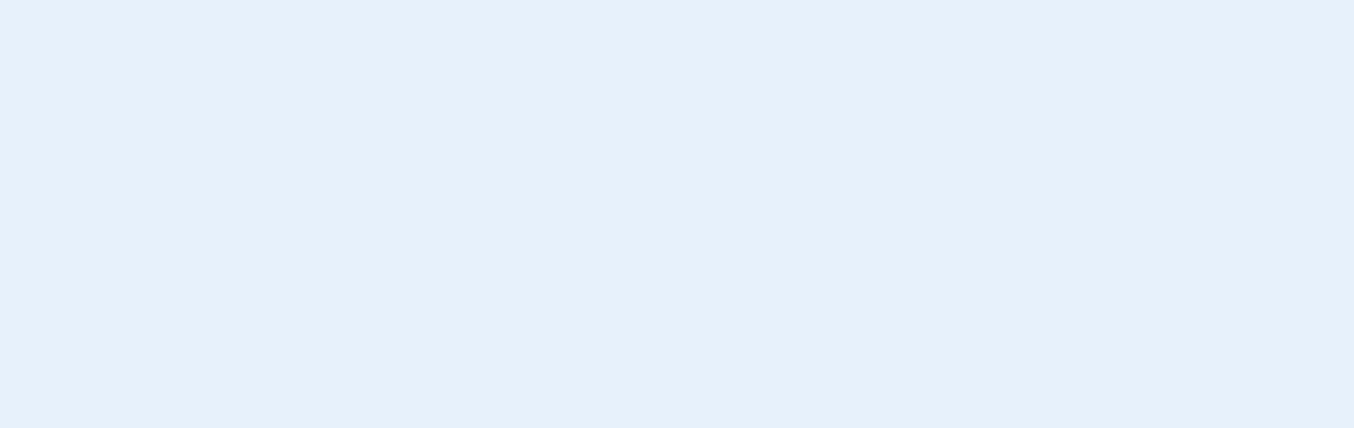
**Figura 4-18:** yo ventana acoplable con un proyecto .NET Core existentes en su sitio

Desde el DockerFile, se especifica la imagen de base acoplable lo que va a utilizar (como usar “de Microsoft / dotnet: 1.0.0-core”). Por lo general, va a construir su imagen personalizada en la parte superior de una imagen de la base de que se puede obtener de cualquier repositorio oficial en el[Registro acoplable Hub](https://hub.docker.com/) (como un [imagen para .NET Core](https://hub.docker.com/r/microsoft/dotnet/) o uno [para Node.js)](https://hub.docker.com/_/node/).

***Opción A: Utilizar una imagen acoplable oficial existente***

El uso de un repositorio oficial de una pila de idiomas con un número de versión asegura que las mismas características del lenguaje están disponibles en todas las máquinas (incluyendo el desarrollo, prueba y producción).

La siguiente es una DockerFile muestra para un contenedor de .NET Core:



* imagen de base acoplable a utilizar de Microsoft / aspnetcore: 1.0.1
* Establecer el directorio y los archivos de trabajo que va a copiar a la fuente de la imagen ARG

WORKDIR / app

COPIA $ {source: -bin / Liberación / PublishOutput}.

* Configurar el puerto de escucha al 80 (Puerto interno / asegurado dentro de host del estibador) EXPONER 80
* punto de entrada de la aplicación

EntryPoint [ "dotnet", "MyCustomMicroservice.dll"]

En este caso, se está utilizando la versión 1.0.1 de la imagen ASP.NET Core acoplable el funcionario nombrado para Linux Microsoft / aspnetcore: 1.0.1. Para más detalles, consultar el [página ASP.NET Core acoplable Imagen](https://hub.docker.com/r/microsoft/aspnetcore/)

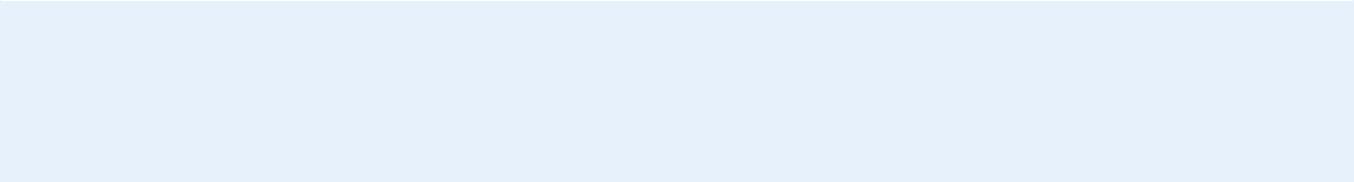
1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

y el [página .NET Core acoplable Imagen.](https://hub.docker.com/r/microsoft/dotnet/) También podría usar otra imagen como comparables nodo: 4 onbuild- para Node.js, o muchas otras imágenes preconfiguradas para lenguajes de desarrollo, que están disponibles en [acoplable Hub.](https://hub.docker.com/explore/)

En el DockerFile, también es necesario para instruir acoplable a escuchar el puerto TCP que va a utilizar en tiempo de ejecución (por ejemplo, el puerto 80).

Hay otras líneas de configuración que se pueden agregar en el DockerFile en función del idioma / marco que está utilizando, por lo acoplable sabe cómo ejecutar la aplicación. Por ejemplo, se necesita el

PUNTO DE ENTRADA linea con [ "Dotnet", "MyCustomMicroservice.dll"] para ejecutar una aplicación .NET Core, aunque puede tener múltiples variantes, dependiendo del enfoque para crear y ejecutar el servicio. Si está utilizando el SDK y dotnet CLI para generar y ejecutar la aplicación .NET, sería ligeramente diferente. La conclusión es que laPUNTO DE ENTRADA la línea más líneas adicionales serán diferentes dependiendo del lenguaje / plataforma que elija para su aplicación.



**Más información** Para obtener información sobre la construcción de imágenes acoplables para aplicaciones .NET Core, vaya a [https://docs.microsoft.com/dotnet/articles/core/docker/building-net-docker-images.](https://docs.microsoft.com/dotnet/articles/core/docker/building-net-docker-images)

Para obtener más información sobre la construcción de sus propias imágenes, vaya a [https://docs.docker.com/engine/](https://docs.docker.com/engine/tutorials/dockerimages/) [tutoriales / dockerimages /.](https://docs.docker.com/engine/tutorials/dockerimages/)

**repositorios de imágenes multiplataforma**

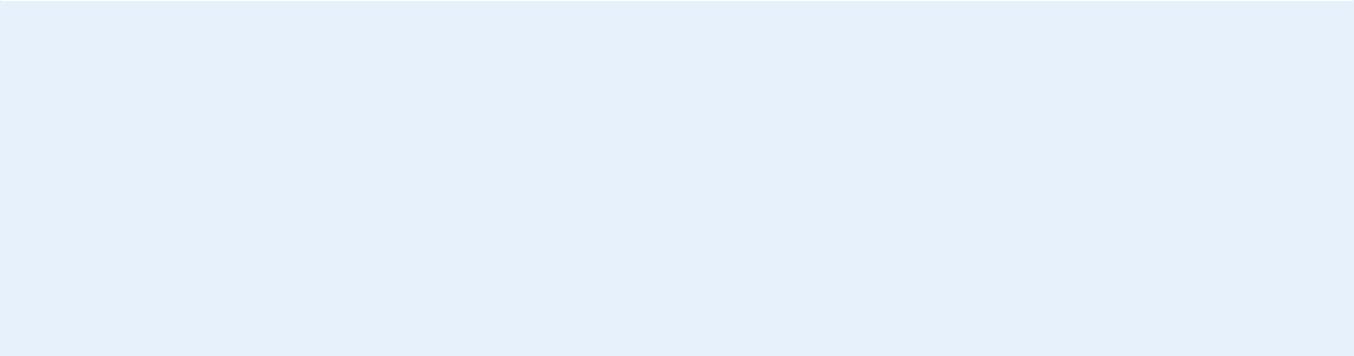
Como contenedores de Windows cada vez más frecuentes, un único repositorio puede contener variantes de la plataforma, como una imagen de Linux y Windows. Esta es una característica nueva que entra acoplable que hace posible que los proveedores utilizan un único repositorio para cubrir múltiples plataformas, como[Microsoft / aspdotnetcore](https://hub.docker.com/r/microsoft/aspnetcore/)repositorio, que está disponible en el Registro DockerHub. A medida que la función se llena de vida, tirando de esta imagen desde un host de Windows tirará de la variante de Windows, mientras que tirando del mismo nombre de la imagen de un host Linux se tire de la variante de Linux.

***Opción B: crear la imagen base desde cero***

Usted puede crear su propia imagen de base acoplable a partir de cero, como se explica en este [artículo](https://docs.docker.com/engine/userguide/eng-image/baseimages/) a partir del estibador. Este es un escenario que probablemente no es el mejor para usted si usted está recién empezando con estibador, pero si desea establecer los bits específicos de su propia imagen base de, usted puede hacerlo.

**Paso 3: Crear sus imágenes personalizadas de Docker incrustación de su servicio en ella**

Para cada servicio personalizado que comprende la aplicación, tendrá que crear una imagen relacionada. Si su aplicación se compone de un único servicio o aplicación web, necesitará sólo una única imagen.



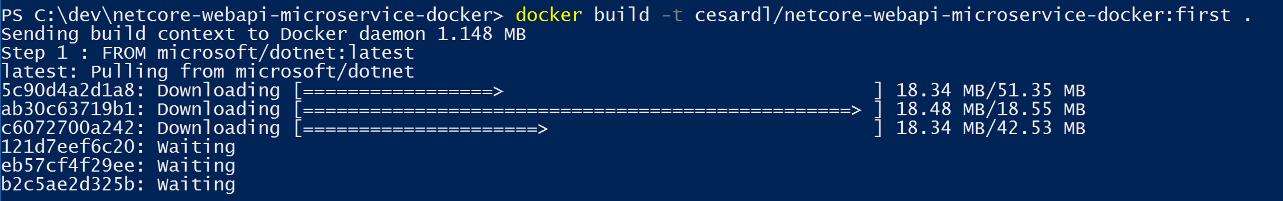
**Nota** Al tomar en cuenta el “flujo de trabajo DevOps -loop exterior”, las imágenes serán creadas por un proceso automático de generación cada vez que se presiona su código fuente a un repositorio Git (integración continua) para que las imágenes serán creadas en ese entorno global de su fuente código.

Pero, antes de considerar ir a esa ruta lazo externo, tenemos que garantizar que la aplicación del estibador es en realidad funciona correctamente para que no empujan código que podría no funcionar correctamente con el sistema de control de código fuente (Git, etc.).

Por lo tanto, cada desarrollador primero que tiene que hacer todo el proceso de bucle interno para poner a prueba a nivel local y continuar desarrollando hasta que quieren empujar una característica completa o cambiar al sistema de control de código fuente.

Para crear una imagen en su entorno local y el uso de la DockerFile, se puede utilizar el estibador construir de comandos, como se demuestra en la figura 4-19 (también se puede ejecutar cargador de muelle-componer hasta --build para aplicaciones compuestas por varios contenedores / servicios).

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure



**Figura 4-19:** Corriendoacumulación ventana acoplable

Opcionalmente, en lugar de ejecutar directamente acumulación ventana acoplable Del una carpeta desplegable con las librerías .NET necesarias mediante el uso de la y luego ejecutar acumulación ventana acoplable.

carpeta de proyecto, primero se puede generar ejecutar dotnet publicar mando,

En este ejemplo, se crea una imagen acoplable con el nombre cesardl / Netcore-WebAPI-microService-ventana acoplable: primero (:primero es una etiqueta, como una versión específica). Usted puede tomar este paso para

cada imagen personalizada que necesita crear para su aplicación acoplable compuesta con varios contenedores.

Puede encontrar las imágenes existentes en su repositorio local (el equipo de desarrollo) mediante el uso de la imágenes ventana acoplable comando, como se ilustra en la figura 4-20.

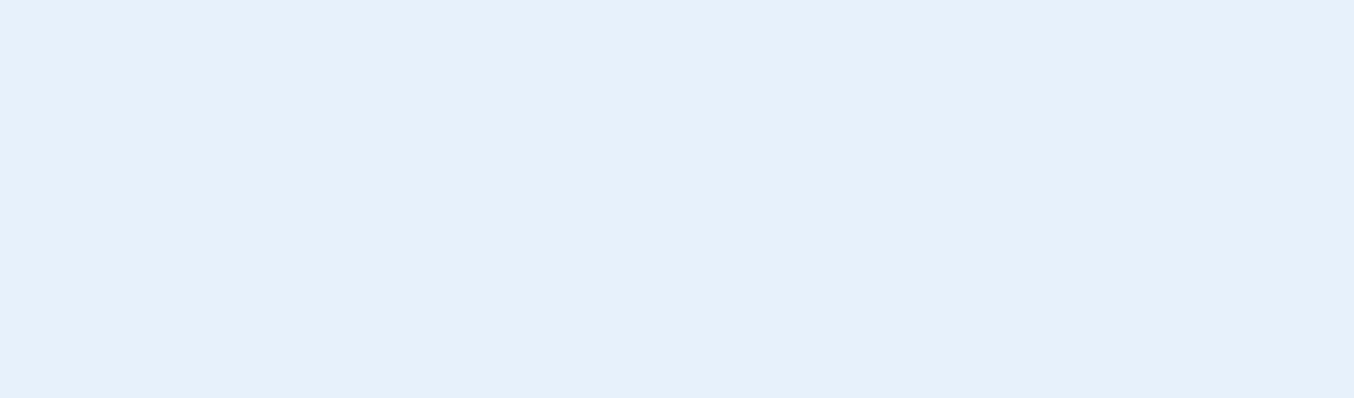


**Figura 4-20:** Visualización de imágenes existentes que utilizanimágenes ventana acoplable

**Paso 4: (Opcional) Definir sus servicios en la ventana acoplable-compose.yml en la construcción de una aplicación compuesta acoplable con múltiples servicios**

Con el archivo de cargador de muelle-compose.yml puede definir un conjunto de servicios relacionados a ser desplegado como una aplicación compuesta con los comandos de despliegue se explica en la sección siguiente paso.

Es necesario crear ese archivo en la carpeta principal o una solución de raíz; que debe tener un contenido similar al siguiente archivo de cargador de muelle-compose.yml:



versión 2'

servicios:

web:

construir: .

puertos:

* "81:80"

volúmenes:

* .: / Código depends\_on:
* Redis

Redis: Imagen: Redis

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

En este caso particular, este archivo se definen dos servicios: la web servicio (su servicio personalizado) y la Redis servicio (un servicio de caché popular). Cada servicio se desplegará como un contenedor, por lo que necesitamos utilizar una imagen de Docker concreto para cada uno. Para este servicio web en particular, la imagen tendrá que hacer lo siguiente:

* Construir desde el DockerFile en el directorio actual
* Reenviar el puerto expuesta 80 en el recipiente al puerto 81 en el equipo host
* Montar el directorio del proyecto en el host a / código dentro del contenedor, por lo que es posible que usted pueda modificar el código sin tener que reconstruir la imagen
* Vincular el servicio web para el servicio Redis

los Redis servicio utiliza el [Redis última imagen pública](https://hub.docker.com/_/redis/) retirado del registro del estibador de concentradores. [Redis](http://redis.io/) es un sistema de caché muy popular para aplicaciones de servidor.

**Paso 5: Generar y ejecutar su aplicación acoplable**

Si su aplicación tiene un solo contenedor, sólo tiene que ejecutarlo mediante el despliegue a su acoplable Anfitrión (VM o servidor físico). Sin embargo, si su aplicación se compone de múltiples servicios, que necesita para componer, también. Vamos a ver las diferentes opciones.

***Opción A: Ejecutar un solo contenedor o servicio***

Puede ejecutar la imagen mediante el uso de la ventana acoplable carrera ventana acoplable de comandos, como se muestra aquí:

**carrera ventana acoplable** -t -d -p 80: 5000 cesardl / Netcore-WebAPI-microService-ventana acoplable: primero

Tenga en cuenta que para esta implementación en particular, vamos a redirigir las peticiones enviadas al puerto 80 al puerto interno de 5000. Ahora, la aplicación está escuchando en el puerto externo 80 en el nivel de acogida.

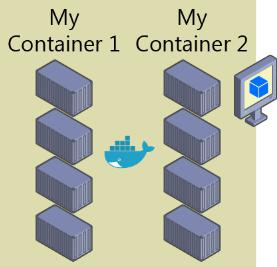
***Opción B: componer y ejecutar una aplicación de múltiples contenedores***

En la mayoría de escenarios empresariales, una aplicación acoplable se compone de múltiples servicios. Para estos casos, se puede ejecutar el comandocargador de muelle-componer hasta(Figura 4-21), que utilizará el archivo de cargador de muelle-compose.yml que pudiera haber creado previamente. Al ejecutar este comando despliega una aplicación compuesta con todos sus contenedores correspondientes.



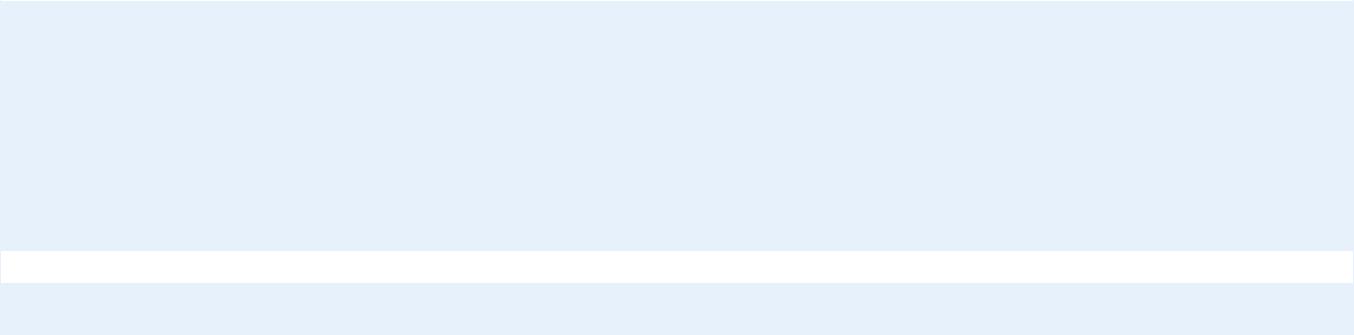
**Figura 4-21:** Resultados de la ejecución de la "ventana acoplable a componer hasta" comando

Después de ejecutar cargador de muelle-componer hasta , Implementar su aplicación y su recipiente (s) relacionada a su anfitrión acoplable, como se ilustra en la figura 4-22, en la representación VM.



**Figura 4-22:** VM con contenedores de Docker desplegado

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure



**Nota** cargador de muelle-componer hastaycarrera ventana acoplablepodría ser suficiente para probar sus contenedores en suentorno de desarrollo, pero es posible que no usarlos en absoluto si usted está esperando para trabajar con clusters y orchestrators como estibador Swarm, Mesosphere CC / CA, o Kubernetes acoplables con el fin de ser capaz de escalar. Si está utilizando un cluster como[modo de ventana acoplable Swarm](https://docs.docker.com/engine/swarm/) (Disponible en acoplable para Windows y Mac desde la versión 1.12), que necesita para implementar y probar con comandos adicionales, tales como Servicio ventana acoplable a crear para los servicios individuales, o cuando se va a implementar una aplicación compuesta de varios contenedores, usando docker haz componga y estibador desplegar myBundleFile, Mediante el despliegue de la aplicación compuesta como una apilar, Como se explica en el artículo [Bundles aplicación distribuida](https://blog.docker.com/2016/06/docker-app-bundle/) a partir del estibador.

por [CC / CA](https://mesosphere.com/blog/2015/09/02/dcos-cli-command-line-tool-datacenter/) y [Kubernetes](http://kubernetes.io/docs/user-guide/deployments/#creating-a-deployment) que usaría diferentes comandos y scripts de implementación, así.

**Paso 6: Probar la aplicación del estibador (a nivel local, en su CD local VM)**

Este paso puede variar dependiendo de lo que su aplicación se está haciendo.

En una muy simple API Web .NET Core “Hello World” desplegado como un único contenedor o servicio, usted sólo tiene que acceder a los servicios facilitando el puerto TCP especificado en la DockerFile.

Si localhost no está activada, para navegar a su servicio, encontrar la dirección IP de la máquina mediante el uso de este comando:

cargador de muelle-máquina IP de su-nombre-contenedor

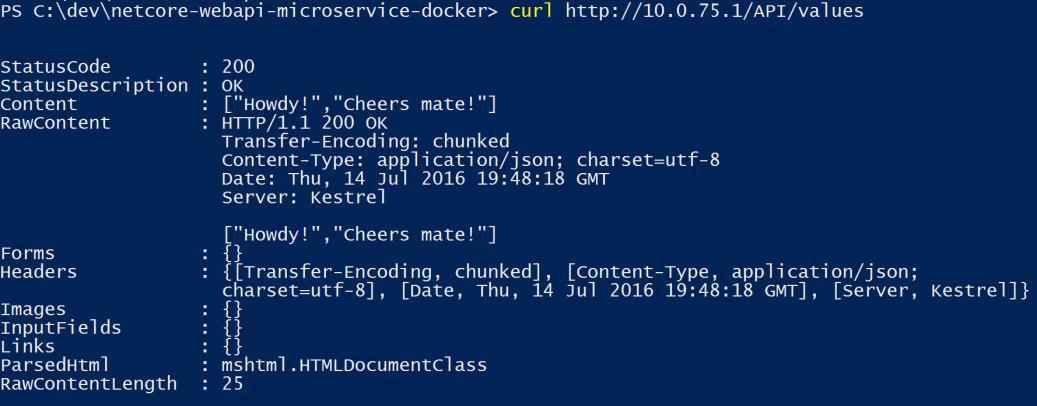
En el host del estibador, abra un navegador y vaya a ese sitio; debería ver su funcionamiento aplicación / servicio, como se demuestra en la figura 4-23.



**Figura 4-23:** Pruebas de una aplicación acoplable localmente usando localhost

Tenga en cuenta que está utilizando el puerto 80, pero internamente estaba siendo redirigido a puerto 5000, porque esa es la forma en que se desplegó con carrera ventana acoplable, Como se explicó anteriormente.

Esto se comprueba mediante el uso de rizo de la terminal. En una instalación acoplable en Windows, el valor predeterminado IP es 10.0.75.1, como se representa en la figura 4-24.



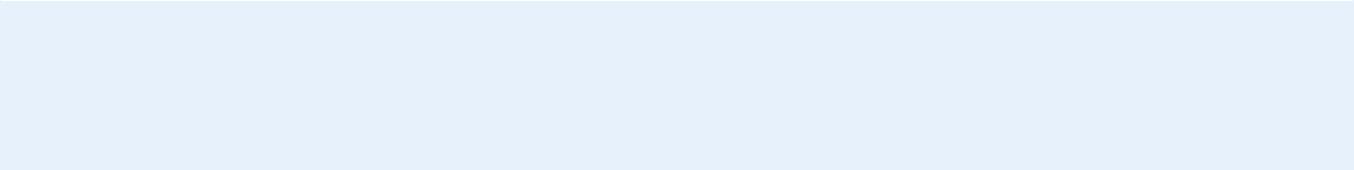
**Figura 4-24:** Prueba de una aplicación acoplable a nivel local mediante el uso de CURL

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

**Depuración de un contenedor en ejecución en acoplable**

Código Visual Studio es compatible con la depuración del estibador si está usando Node.js y otras plataformas como .NET Core contenedores.

También puede depurar contenedores .NET Core en acoplable al utilizar Visual Studio, como se describe en la siguiente sección.



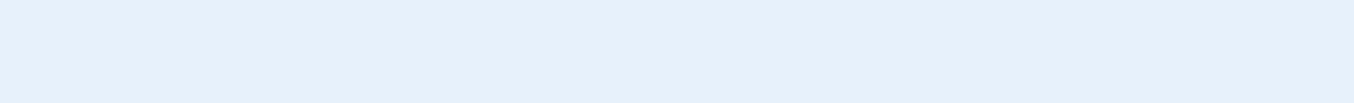
**Más información:** Para obtener más información acerca de la depuración contenedores Node.js Docker, vaya a <https://blog.docker.com/2016/07/live-debugging-docker/>y [https://blogs.msdn.microsoft.com/](https://blogs.msdn.microsoft.com/user_ed/2016/02/27/visual-studio-code-new-features-13-big-debugging-updates-rich-object-hover-conditional-breakpoints-node-js-mono-more/) [user\_ed / 2016/02/27 / visual-studio-código-new-características -13-grandes-depuración-updates-rica-objeto-asomar-condicionales-puntos de interrupción de nodo js-mono-más /.](https://blogs.msdn.microsoft.com/user_ed/2016/02/27/visual-studio-code-new-features-13-big-debugging-updates-rich-object-hover-conditional-breakpoints-node-js-mono-more/)

El uso de Visual Studio Tools para Docker (Visual Studio en Windows)

El flujo de trabajo de desarrolladores al utilizar Visual Studio Tools para Docker es similar al flujo de trabajo cuando se utiliza código de Visual Studio y acoplable CLI (de hecho, se basa en la misma ventana acoplable CLI), pero es más fácil para empezar, simplifica el proceso, y proporciona una mayor productividad para la construcción, funcionamiento, y componer tareas. También es capaz de ejecutar y depurar los contenedores a través de acciones simples como F5 y Ctrl + F5 de Visual Studio. Aún más, con Visual Studio 2017, además de ser capaz de ejecutar y depurar un solo contenedor, también puede ejecutar y depurar un grupo de contenedores (una solución entera) al mismo tiempo si están definidos en la misma ventana acoplable de composición del .yml archivo en el nivel de la solución.

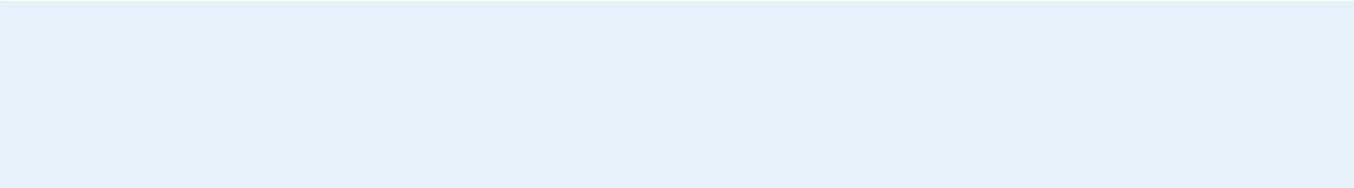
**Configuración de su entorno local**

Con las últimas versiones de estibador para Windows, es más fácil que nunca para desarrollar aplicaciones acoplables debido a que la configuración es sencilla, como se explica en las siguientes referencias.



**Más información:** Para obtener más información acerca de la instalación del estibador para Windows, vaya a [https://docs.docker.com/docker-for-windows/.](https://docs.docker.com/docker-for-windows/)

Si está utilizando Visual Studio 2015, debe tener Actualización 3 o una versión posterior además de las herramientas de Visual Studio para estibador.



**Más información:** Para obtener instrucciones sobre la instalación de Visual Studio, visite [https://www.visualstudio.com/](https://www.visualstudio.com/products/vs-2015-product-editions)[productos / vs -2015 a productos ediciones.](https://www.visualstudio.com/products/vs-2015-product-editions)

Para ver más información sobre la instalación de Visual Studio Tools para Docker, vaya a <http://aka.ms/vstoolsfordocker> y [https://docs.microsoft.com/dotnet/articles/core/docker/visual-studio-tools-for-docker.](https://docs.microsoft.com/dotnet/articles/core/docker/visual-studio-tools-for-docker)

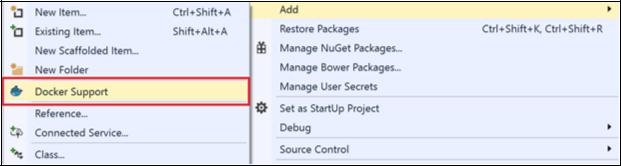
Si está utilizando Visual Studio 2017, el apoyo del estibador ya está incluido.

**El uso de herramientas acoplables en Visual Studio 2015**

Visual Studio Tools para ventana ofrece una manera coherente para desarrollar y validar localmente los contenedores de Docker para Linux en un servidor Linux o acoplable VM, o su Windows contenedores directamente en Windows.

Si está utilizando un único contenedor, E l primero que se necesita para comenzar es activar el soporte acoplable en su proyecto Core .NET. Para ello, haga clic en el archivo de proyecto, como se muestra en la figura 4-25.

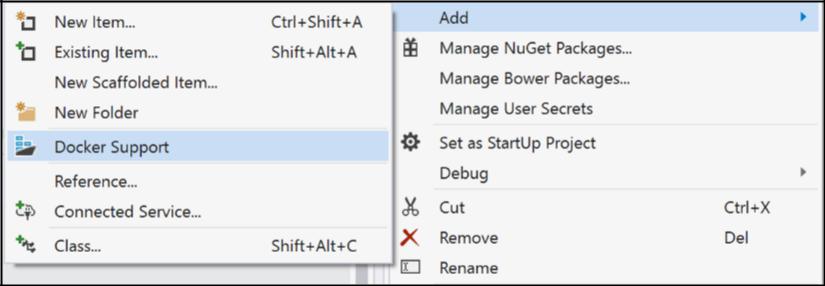
1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure



**Figura 4-25:** Encendido de soporte acoplable para su proyecto de Visual Studio

**El uso de herramientas acoplables en Visual Studio 2017**

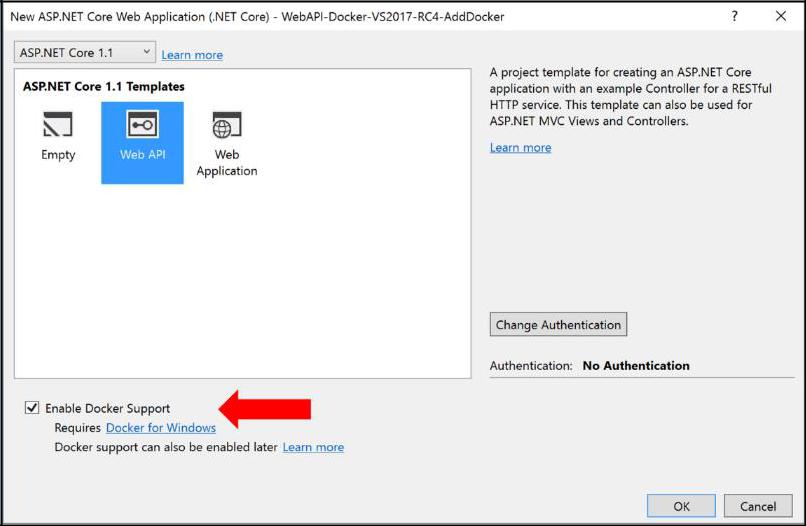
Cuando se agrega soporte acoplable a un proyecto de servicio en la solución (ver Figura 4-26), Visual Studio no es sólo la adición de un archivo DockerFile a su proyecto, sino que también es la adición de una sección de servicios en los archivos ventana acoplable-compose.yml de su solución ( o la creación de los archivos si no existieran). Es una manera fácil de empezar a escribir su solución multicontainer; a continuación, puede abrir los archivos ventana acoplable-compose.yml y actualizarlos con características adicionales.



**Figura 4-26:** Encendido de apoyo Solución Estibador en un proyecto de Visual Studio 2017

Esta acción no sólo se suma el DockerFile a su proyecto, sino que también añade las líneas de configuración requeridos de código a un cargador de muelle-compose.yml global establecida en el nivel de la solución.

También puede activar el soporte acoplable al crear un proyecto ASP.NET Core en Visual Studio 2017, como se muestra en la figura 4-27.



**Figura 4-27:** Encendido de soporte acoplable al crear un proyecto

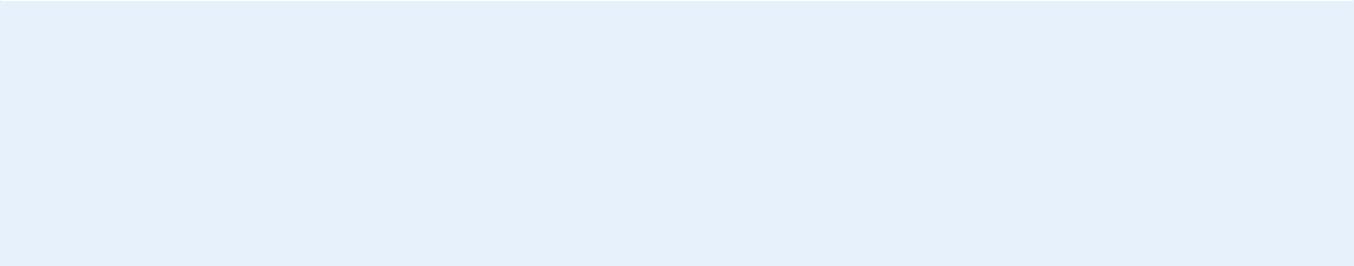
Después de agregar soporte acoplable a su solución en Visual Studio, también verá un nuevo árbol de nodos en el Explorador de soluciones con los archivos ventana acoplable-compose.yml agregados, como se muestra en la figura 4-28.

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure



**Figura 4-28:** archivos ventana acoplable-compose.yml ahora se muestran en el Explorador de soluciones

Se podría implementar una aplicación multicontainer mediante el uso de un único archivo de cargador de muelle-compose.yml cuando se ejecuta cargador de muelle-componer hasta; Sin embargo, Visual Studio agrega un grupo de ellos, por lo que puede anular los valores en función del entorno (desarrollo frente a la producción) y el tipo de ejecución (liberación frente de depuración). Esta capacidad se explicará mejor en los siguientes capítulos.



**Más información:** Para más detalles sobre la implementación de servicios y el uso de Visual Studio Tools para

Estibador, leer los siguientes artículos:

Construir, depurar, actualizar y refrescar aplicaciones en un contenedor local acoplable: [https://azure.microsoft.com/](https://azure.microsoft.com/documentation/articles/vs-azure-tools-docker-edit-and-refresh/) [documentación / articulos / vs -azure-tools-ventana acoplable-edit-y-refresh /](https://azure.microsoft.com/documentation/articles/vs-azure-tools-docker-edit-and-refresh/)

Implementar un recipiente ASP.NET a un host remoto acoplable: [https://azure.microsoft.com/](https://azure.microsoft.com/documentation/articles/vs-azure-tools-docker-hosting-web-apps-in-docker/) [documentación / artículos / VS-azules-tools-Docker-hosting-web-apps-en-ventana acoplable /](https://azure.microsoft.com/documentation/articles/vs-azure-tools-docker-hosting-web-apps-in-docker/)

Comandos de uso de Windows PowerShell en un DockerFile para configurar Windows Contenedores (estándar basado acoplable)

Con [Contenedores de windows,](https://msdn.microsoft.com/en-us/virtualization/windowscontainers/about/about_overview) usted puede convertir sus aplicaciones de Windows existentes para Döcker imágenes y desplegarlos con las mismas herramientas que el resto del ecosistema del estibador.

Para usar Windows Contenedores, sólo tiene que escribir los comandos de Windows PowerShell en el DockerFile, como se demuestra en el siguiente ejemplo:

De Microsoft / windowsservercore

Descripción LABEL = "IIS" Vendor = "Microsoft" versión = "10"

PowerShell se ejecutan -Command Add-WindowsFeature Web-Server

CMD [ "ping", "localhost", "-t"]

En este caso, estamos usando Windows PowerShell para instalar una imagen base de Windows Server Core, así como IIS.

De manera similar, también se puede utilizar comandos de Windows PowerShell para configurar componentes adicionales como el 4.x ASP.NET tradicional y .NET 4.6 o cualquier otro software de Windows, como se muestra aquí:

RUN PowerShell complemento WindowsFeature Web-asp-net45

1. CAPÍTULO 4 | Diseño y desarrollo de aplicaciones en contenedores utilizando acoplable y Microsoft Azure

CAPÍTULO 5

Estibador

solicitud

DevOps

flujo de trabajo con

herramientas de Microsoft

Microsoft Visual Studio, Visual Studio Team Services, Team Foundation Server, y puntos de vista de aplicaciones proporcionan un ecosistema integral para las operaciones de TI que dan a su equipo las herramientas para gestionar proyectos y rápidamente construir, probar y desplegar aplicaciones en contenedores de desarrollo.

Con Visual Studio Team Services Visual Studio y en la nube, junto con Team Foundation Server en las instalaciones, los equipos de desarrollo pueden construir de forma productiva, probar y lanzar aplicaciones en contenedores dirigidos hacia cualquier plataforma (Windows o Linux).

herramientas de Microsoft pueden automatizar la tubería para implementaciones específicas de aplicaciones en contenedores -Docker, .NET Core, o cualquier combinación con otras plataformas de lo global construye y de integración continua (CI) y pruebas con servicios de estudio o equipo de Visual Team Foundation Server, hasta el despliegue continuo (CD) para entornos Docker (desarrollo, puesta en escena, producción), y a

44 CAPÍTULO 5 | Acoplable DevOps aplicaciones de flujo de trabajo con las herramientas de Microsoft

transmitir análisis de información acerca de los servicios al equipo de desarrollo de aplicaciones a través de Insights. Cada código de cometer puede iniciar una acumulación (IC) y automáticamente desplegar los servicios a los entornos en contenedores específicos (CD).

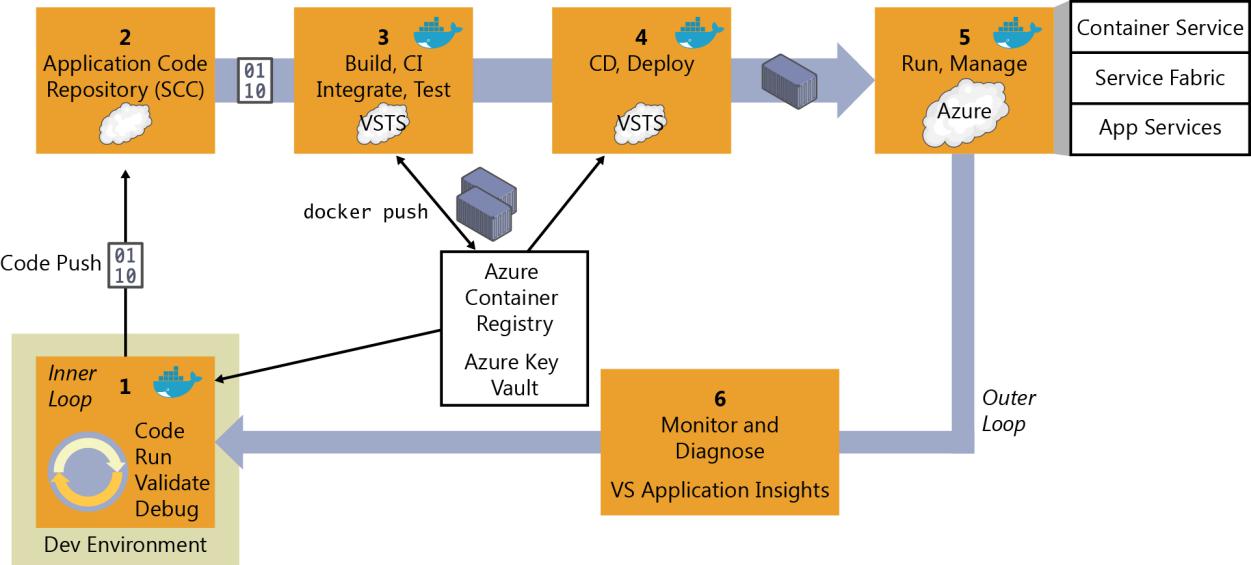
Los desarrolladores y probadores puede fácil y rápidamente prestación desarrollo de la producción similar y la prueba entornos basados ​​en acoplable mediante el uso de plantillas en Microsoft Azure.

La complejidad de los aumentos de desarrollo de aplicaciones en contenedores dependiendo de la complejidad de manera constante negocio y necesidades de escalabilidad. Un buen ejemplo de esto son las aplicaciones basadas en arquitecturas microservicios. Para tener éxito en este entorno, el proyecto debe automatizar todo el ciclo de vida no solamente la construcción y despliegue, sino que también debe gestionar versiones, junto con la colección de telemetría. Visual Studio Team Servicios Azure y ofrecen las siguientes capacidades:

* Studio Team Foundation Servicios / Equipo de Gestión visual de código fuente del servidor (basado en Git o Fundación de control de versiones Team), la planificación ágil (Agile, Scrum, CMMI y son compatibles), CI, gestión de versiones, y otras herramientas para equipos ágiles.
* Estudio Equipo de Servicios visual / Team Foundation Server incluye un potente y creciente ecosistema de extensiones de primera y de terceros con los que fácilmente puede construir un IC, construir, probar, de entrega, y la liberación de tuberías para la gestión microservicios.
* Ejecutar pruebas automatizadas como parte de su canal de acumulación en el estudio de los servicios del equipo visual.
* Visual Studio Team Services puede apretar el ciclo de vida DevOps con entrega en múltiples entornos, no sólo para los entornos de producción, sino también para las pruebas, incluyendo la experimentación A / B, [comunicados de Canarias,](http://martinfowler.com/bliki/CanaryRelease.html) y así.
* Las organizaciones pueden fácilmente provisión de contenedores de Docker de imágenes privadas almacenadas en Azure Registro de contenedores junto con cualquier dependencia de los componentes de Azure (datos, PaaS, etc.) usando las plantillas del Administrador de recursos Azure con las herramientas con las que ya se sienta cómodo trabajando.

Los pasos en el flujo de trabajo DevOps-bucle externo para una aplicación acoplable

Figura 5-1 presenta una representación de extremo a extremo de las etapas que comprenden el flujo de trabajo de bucle exterior DevOps.



**Figura 5-1:** DevOps flujo de trabajo-bucle externo para aplicaciones acoplables con las herramientas de Microsoft

45 CAPÍTULO 5 | Acoplable DevOps aplicaciones de flujo de trabajo con las herramientas de Microsoft

Ahora, vamos a examinar cada uno de estos pasos con mayor detalle.

**Paso 1: flujo de trabajo de desarrollo-bucle interior**

Este paso se explica en detalle en el capítulo 4, pero, para recapitular, aquí es donde empieza el lazo externo, el momento en que un desarrollador empuja código para el sistema de gestión de control de código fuente (como Git) iniciar acciones de tuberías CI.

**Paso 2: Integración de control de código fuente y la gestión con Visual Studio Team Services y Git**

En este paso, es necesario tener un sistema de control de versiones para reunir una versión consolidada de todo el código que viene de los diferentes desarrolladores en el equipo.

A pesar de que el control de código fuente (SCC) y la gestión de código fuente puede parecer de segunda naturaleza para la mayoría de los desarrolladores, al crear aplicaciones Estibador en un ciclo de vida DevOps, es muy importante hacer hincapié en que no debe presentar las imágenes acoplables con la aplicación directa acoplable al Registro mundial (como Azure Registro de contenedores o acoplable Hub) de la máquina del desarrollador. Por el contrario, las imágenes acoplables a ser liberados y desplegados en entornos de producción se debe crear únicamente en el código fuente que se está integrando en su construcción o CI tubería global basada en su repositorio de código fuente (como Git).

Las imágenes locales generadas por los desarrolladores puedan ser realmente utilizadas por los desarrolladores solo cuando se prueba dentro de sus propias máquinas. Por esta razón, es fundamental contar con la tubería DevOps activado a partir del código SCC.

Visual Studio Team Services y la fundación del equipo de soporte de servidor Git y de control de versiones de Team Foundation. Se puede elegir entre ellos y utilizarlo para una experiencia de extremo a extremo Microsoft. Sin embargo, también se puede administrar su código en repositorios externos (como GitHub, en las instalaciones repositorios Git o Subversion) y aún así ser capaz de conectarse a él y obtener el código como el punto de partida para su tubería DevOps CI.

**Paso 3: Construir, CI, integrar y prueba con Visual Studio Team Services y acoplable**

CI ha convertido en un estándar para las pruebas de software moderno y entrega. La solución acoplable mantiene una clara separación de intereses entre los equipos de desarrollo y operaciones. La inmutabilidad de imágenes Docker asegura un despliegue repetible entre lo que está desarrollado, probado a través de CI, y se ejecutan en la producción. Acoplable motor desplegado a través de los ordenadores portátiles para desarrolladores y la infraestructura de pruebas hace que los contenedores de entornos a través de portátiles.

En este punto, después de tener un sistema de control de versiones con el código correcto presentado, es necesario un servicio de generación para recoger el código y ejecutar la construcción y pruebas global.

El flujo de trabajo interno para este paso (CI, construir, probar) es acerca de la construcción de una tubería de CI consiste en su repositorio de código (Git, etc.), el servidor de compilación (Studio Team Visual Servicios), acoplable del motor, y un Registro acoplable .

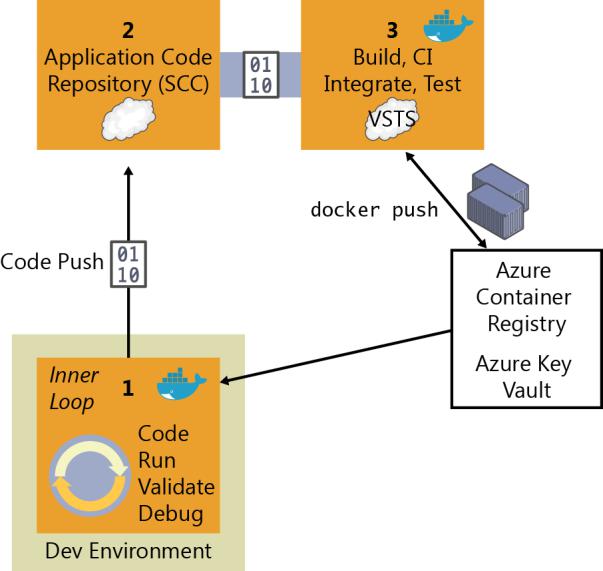
Puede utilizar Studio Team Services visuales como la base para la construcción de sus aplicaciones y el establecimiento de su canal de CI, y para la publicación de los “artefactos construidos” a un “repositorio artefactos”, que se explica en el siguiente paso.

Al utilizar acoplable para el despliegue, los “artefactos finales” que se desplegarán imágenes son acoplables con su aplicación o servicios incorporados dentro de ellos. Esas imágenes son empujados o publicados a una

46 CAPÍTULO 5 | Acoplable DevOps aplicaciones de flujo de trabajo con las herramientas de Microsoft

*Registro acoplable* (Un depósito privado como los que se puede tener en Azure Registro de contenedores, o una pública como estibador Registro de concentradores, que se utiliza comúnmente para las imágenes oficiales de bases).

Aquí es el concepto básico: La tubería de CI se dio inicio por una comprometerse a un repositorio Git como SCC. El cometen causará Studio Team Services visuales para ejecutar un trabajo de construcción dentro de un contenedor acoplable y, al término de ese trabajo, empujar una imagen acoplable al Registro del estibador, como se ilustra en la Figura 5-2.



**Figura 5-2:** Los pasos involucrados en la IC

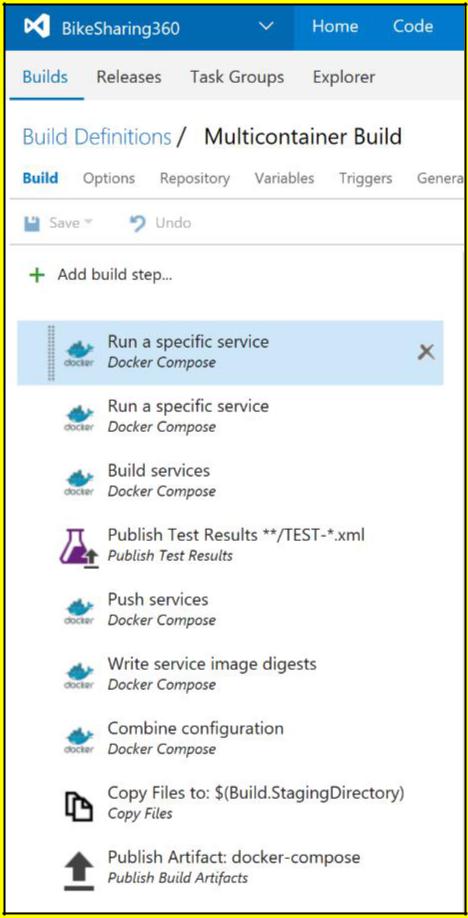
Estos son los pasos básicos del flujo de trabajo con IC del estibador y Servicios Visual Studio Team:

1. El desarrollador de una confirmación empuja a un repositorio de SCC (Git / Studio Team Servicios Visuales, GitHub, etc.).
2. Si está utilizando Servicios de Estudio equipo de Visual o Git, CI está integrada, lo que significa que es tan simple como seleccionar una casilla de verificación en el estudio de los servicios del equipo visual. Si está utilizando un SCC externa (como GitHub), un web hook notificará Studio Team Servicios Visuales de la actualización o empujar a Git / GitHub.
3. Visual Studio Team Servicios tira del repositorio de SCC, incluyendo el DockerFile la descripción de la imagen, así como el código de la aplicación y prueba.
4. Visual Studio Team Services genera una imagen de estibador y lo etiqueta con un número de compilación.
5. Visual Studio Team Servicios instancia del contenedor acoplable dentro del huésped acoplable aprovisionado, y se ejecutan las pruebas adecuadas.
6. Si las pruebas tienen éxito, la imagen se vuelve a etiquetar primero un nombre significativo para que sepa que es una “acumulación bendito” (como “/1.0.0” o cualquier otra etiqueta), y luego empujó hasta su registro estibador (estibador Hub, Azure Container Registro, DTR, etc.)

**La implementación de la tubería con IC Servicios de Visual Studio Team y la extensión del estibador de Studio Team Services Visuales**

los [extensión de Visual Studio Team Servicios acoplable](https://aka.ms/vstsdockerextension) agrega una tarea a la tubería de CI con el que se puede construir imágenes Docker, empujar imágenes acoplables a un registro autenticado estibador, imágenes Docker correr, o ejecutar otras operaciones ofrecidas por la ventana acoplable CLI. También agrega una tarea del estibador de composición que se puede utilizar para construir, empujar, y ejecutar aplicaciones multicontainer estibador, o ejecutar otras operaciones ofrecidas por la ventana acoplable Componer CLI, como se muestra en la Figura 5-3.

47 CAPÍTULO 5 | Acoplable DevOps aplicaciones de flujo de trabajo con las herramientas de Microsoft



**Figura 5-3:** El oleoducto acoplable CI en Studio Team Services Visuales

La extensión del estibador puede utilizar extremos de servicio rural por acoplables y de contenedores o de imagen registros. El valor por defecto para las tareas usando una serie acoplable local, si está disponible (esto requiere actualmente un agente personalizado Visual Studio Team Services); de lo contrario, requieren que proporcione una conexión de host del estibador. Acciones que dependen de ser autenticado con un registro del estibador, tales como empujar una imagen, requieren que se proporcione una conexión acoplable registro.

La extensión de Visual Studio Team Servicios del estibador instala los siguientes componentes en su cuenta de Visual Studio Team Servicios:

* Un extremo de servicio para la conexión a un registro acoplable
* Un extremo de servicio para la conexión a un host del estibador de contenedores
* Una tarea acoplable para hacer lo siguiente:
  + Construir una imagen
  + Empujar una imagen o un depósito a un registro
  + Ejecutar una imagen en un contenedor
  + Ejecutar un comando acoplable
* Una tarea del estibador Componer para ejecutar un comando del estibador Componer

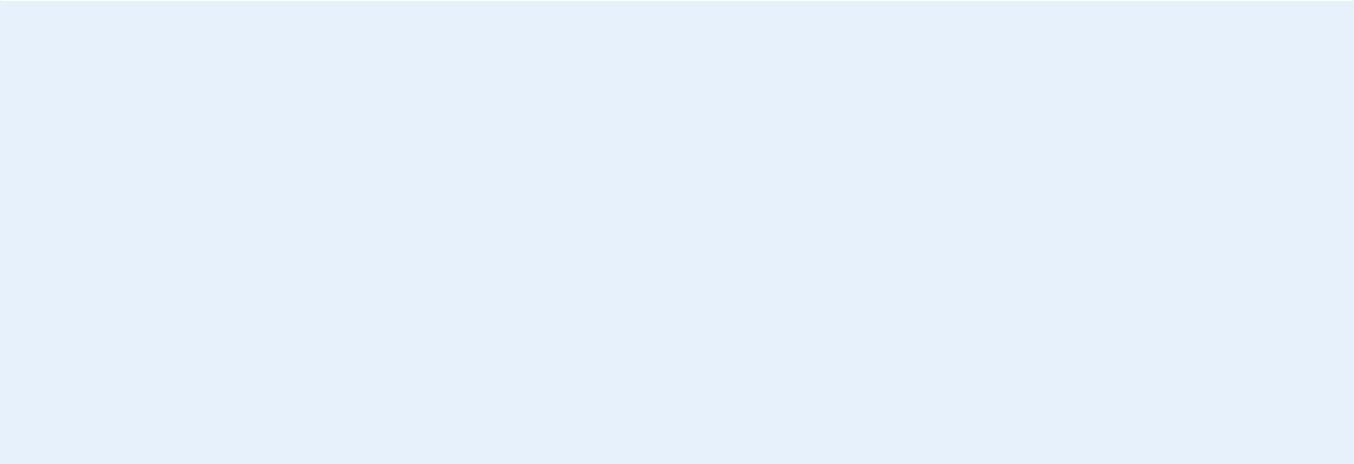
Con estas tareas de Visual Studio Team Services, una acumulación de Linux-Docker Host / VM provisionados en Azure y su registro preferido Docker (Azure Registro de contenedores, acoplable Hub, privado acoplable DTR, o cualquier otro registro del estibador) se puede montar su canal de Docker CI en una forma muy consistente.

48 CAPÍTULO 5 | Acoplable DevOps aplicaciones de flujo de trabajo con las herramientas de Microsoft

***requisitos:***

* Estudio Equipo de Servicios visual, o para instalaciones de correo locales, Team Foundation Server 2015 Actualización 3 o posterior.
* Un agente de Visual Studio Team Services que tiene los binarios acoplables.

Una manera fácil de crear una de ellas es utilizar acoplable a ejecutar un contenedor, según la imagen del estibador del agente de Visual Studio Team Services.



**Más información** Para leer más sobre el montaje de una tubería de Visual Studio Team Servicios del estibador CI y para ver tutoriales, visite los siguientes sitios:

Ejecución de un agente de Visual Studio Team Services como un recipiente acoplable: [https://hub.docker.com/r/](https://hub.docker.com/r/microsoft/vsts-agent/) [Microsoft / VSTS-agente /](https://hub.docker.com/r/microsoft/vsts-agent/)

extensión VSTS acoplable: <https://aka.ms/vstsdockerextension>

Edificio imágenes .NET Core Linux acoplables con Studio Team Services visuales: [https://blogs.msdn.microsoft.com/stevelasker/2016/06/13/building-net-core-linux-docker-images-con-visual-studio-equipo-services /](https://blogs.msdn.microsoft.com/stevelasker/2016/06/13/building-net-core-linux-docker-images-with-visual-studio-team-services/)

La construcción de una máquina de construcción Estudio del servicio de atención visual basado en Linux con soporte acoplable: [http://donovanbrown.com/post/2016/06/03/Building-a-Linux-Based-Visual-Studio-Team-Service-Construir-Máquina-con-Docker-Support](http://donovanbrown.com/post/2016/06/03/Building-a-Linux-Based-Visual-Studio-Team-Service-Build-Machine-with-Docker-Support)

**Integrar, probar y validar las aplicaciones multicontainer Docker**

Típicamente, la mayoría de aplicaciones acoplables se componen de múltiples recipientes en lugar de un único contenedor. Un buen ejemplo es una aplicación orientada a microservicios para el que tiene un contenedor por microService. Pero, incluso sin seguir estrictamente las pautas de aproximación microservicios, es muy probable que su aplicación acoplable se compone de varios contenedores o servicios.

Por lo tanto, después de la construcción de los contenedores de aplicaciones en la tubería CI, también es necesario para desplegar, integrar y probar la aplicación en su conjunto con todos sus contenedores dentro de una integración de host del estibador o incluso en un grupo de ensayo al que se distribuyen sus contenedores.

Si está utilizando un único host, puede utilizar los comandos tales como estibador cargador de muelle de composición delpara construir y desplegar contenedores relacionados para probar y validar el entorno del estibador en una sola máquina virtual. Sin embargo, si se está trabajando con un grupo de orquestador como CC / CA, Kubernetes o acoplable Swarm, que necesita para implementar sus contenedores a través de un mecanismo o orquestador diferente, dependiendo de su cluster / planificador seleccionado.

A continuación se presentan varios tipos de pruebas que se pueden ejecutar en contra de contenedores Docker:

* Las pruebas unitarias para contenedores acoplables
* Los grupos de ensayo de solicitudes relacionadas entre sí o microservicios
* Prueba en la producción y libera “canario”

49 CAPÍTULO 5 | Acoplable DevOps aplicaciones de flujo de trabajo con las herramientas de Microsoft

El punto importante es que cuando se ejecuta la integración y las pruebas de funcionamiento, debe ejecutar las pruebas desde fuera de los contenedores. Las pruebas no deben ser definidos y se ejecutan dentro de los contenedores que va a implementar, ya que los contenedores se basan en imágenes estáticas que deben ser exactamente iguales a los que se va a implementar en la producción.

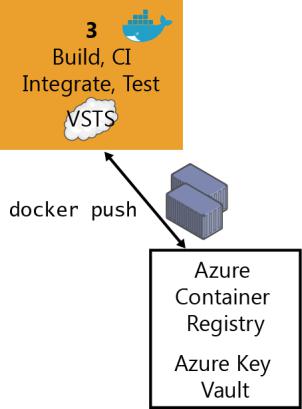
Una opción muy viable al probar escenarios más avanzados como el de probar varios grupos (grupo de prueba, transferencia clúster y clúster de producción) es la publicación de las imágenes para un registro a prueba en varios grupos.

**Empuje la imagen del estibador la aplicación personalizada en su registro global acoplable**

Después de las imágenes Docker se han probado y validado, se le quiere etiquetar y publicarlos en su registro del estibador. El registro del estibador es una pieza fundamental en el ciclo de vida de la aplicación del estibador, ya que es el lugar central donde se almacena la prueba personalizada (también conocido como “bendito imágenes”) para ser desplegados en entornos de control de calidad y producción.

De manera similar a cómo el código de la aplicación almacenada en el repositorio de SCC (Git, etc.) es su “fuente de verdad,” el registro del estibador es su “fuente de verdad” para su aplicación binaria o bits para ser desplegados en los entornos de control de calidad o de producción .

Por lo general, es posible que desee tener sus repositorios privados para sus imágenes personalizadas, ya sea en un depósito privado en Azure Registro de contenedores o en un registro como el -premises Registro acoplable de confianza, o en un registro público en la nube con acceso restringido (como estibador Hub) , aunque en este último caso si el código no es de código abierto, debe confiar en la seguridad del proveedor. De cualquier manera, el método por el cual se hace esto es bastante similar y en última instancia, sobre la base de lacargador de muelle de empuje comando, como se representa en la Figura 5-4.

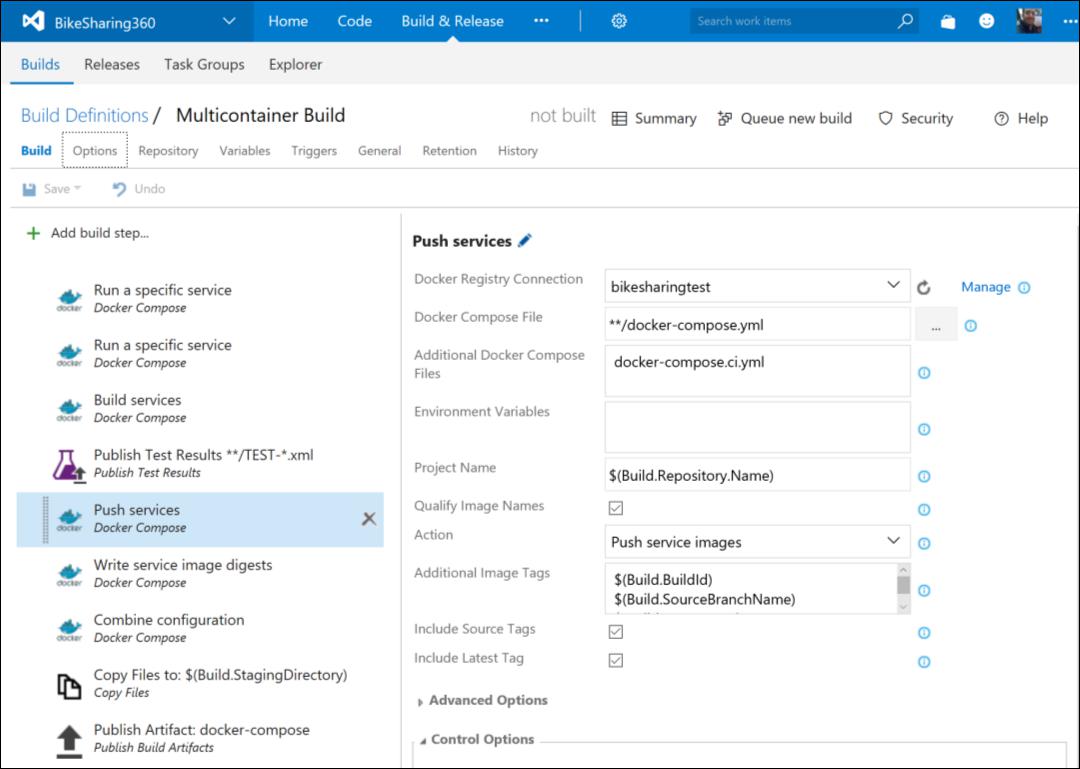


**Figura 5-4:** La publicación de imágenes personalizadas a acoplable Registro

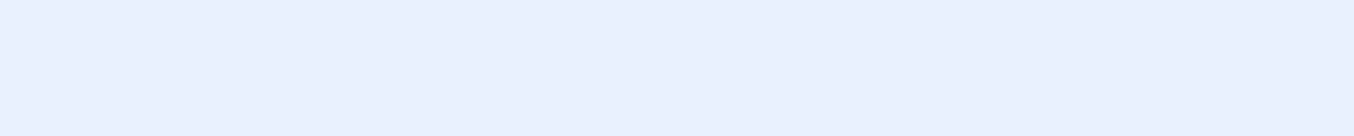
Hay múltiples ofertas de registros acoplables de proveedores de la nube como Azure de contenedores Registro, Amazon Web Services Registro de contenedores, Registro Google Container, Registro Quay, y así sucesivamente.

El uso de la extensión Visual Studio Team Servicios del estibador, puede empujar un conjunto de imágenes de servicio definidas por un archivo de cargador de muelle-compose.yml, con varias etiquetas, a un registro autenticado Docker (como Azure Registro de contenedores), como se muestra en la Figura 5-5 .

50 CAPÍTULO 5 | Acoplable DevOps aplicaciones de flujo de trabajo con las herramientas de Microsoft



**Figura 5-5:** El uso de Studio Team Services visuales para la publicación de imágenes personalizadas a un Registro acoplable



**Más información** Para leer más sobre la extensión del estibador para Visual Studio Team Services, vaya a [https://aka.ms/vstsdockerextension.](https://aka.ms/vstsdockerextension) A aprender más sobre Azure Registro de contenedores, vaya a [https://aka.ms/azurecontainerregistry.](https://aka.ms/azurecontainerregistry)

**Paso 4: CD, desplegar**

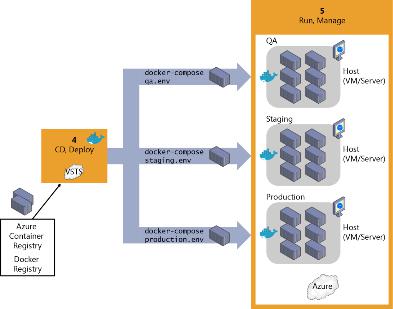
La inmutabilidad de imágenes Docker asegura un despliegue repetible con lo que está desarrollado, t Ested través de CI, y se ejecutan en la producción. Una vez que tenga la aplicación Imágenes Docker publicados en su registro Docker (ya sean privadas o públicas), puede distribuir a los diversos entornos que pueda tener (producción, control de calidad, puesta en escena, etc.) de su tubería de CD mediante el uso de Visual Studio Team tareas de servicios de tuberías o Studio Team Services Gestión de la Entrega visual.

Sin embargo, en este punto depende de qué tipo de aplicación acoplable va a implementar. La implementación de una aplicación simple (de una composición y despliegue punto de vista) como una aplicación monolítica que comprende un par de recipientes o servicios y desplegado a unos pocos servidores o máquinas virtuales es muy diferente de la implementación de una aplicación más compleja como una aplicación microservicios orientado con capacidades hiperescala . Estos dos escenarios se explican en las siguientes secciones.

51 CAPÍTULO 5 | Acoplable DevOps aplicaciones de flujo de trabajo con las herramientas de Microsoft

**Despliegue de aplicaciones acoplables compuestas a múltiples entornos acoplables**

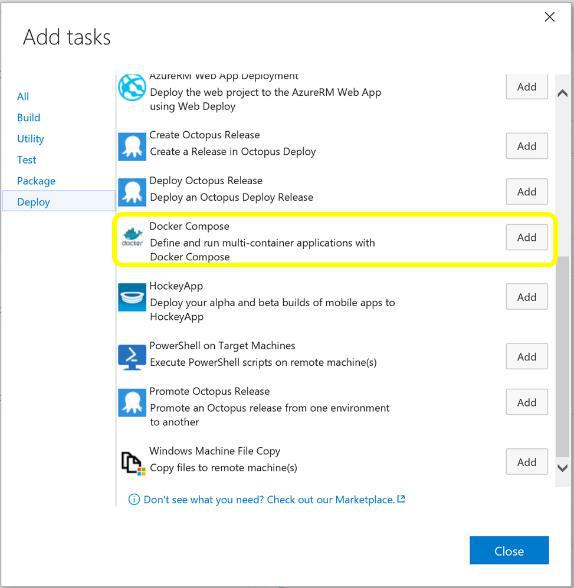
Veamos primero en el escenario menos complejo: el despliegue de los ejércitos Docker simples (VM o servidores) en un único entorno o múltiples entornos (QA, puesta en escena, y producción). En este escenario, internamente su canal de CD se puede utilizardocker- componer (De sus tareas de implementación Visual Studio Team Services) para implementar las aplicaciones acoplables con su conjunto relacionado de contenedores o servicios, como se ilustra en la Figura 5-6.



**Figura 5-6:** Implementación de contenedores de aplicaciones a sencillo registro entornos host acoplable

Figura 5-7 pone de relieve cómo se puede conectar su construcción CI a entornos / prueba de control de calidad a través de Studio Team Services Visuales haciendo clic en Redactar estibador en el cuadro de diálogo Agregar tarea. Sin embargo, cuando el despliegue de entornos de ensayo o de producción, que se suele utilizar Administración de versión se caracteriza por el manejo de múltiples entornos (como control de calidad, puesta en escena, y la producción). Si va a implementar para individuales anfitriones Docker, se está utilizando el equipo de servicios de Visual Studio “acoplable Componer” tarea (que se llama alcargador de muelle-componer hasta mando bajo la campana). Si va a implementar en Azure Servicio de contenedor, se utiliza el Acoplable tarea de despliegue, como se explica en la siguiente sección.

52 CAPÍTULO 5 | Acoplable DevOps aplicaciones de flujo de trabajo con las herramientas de Microsoft



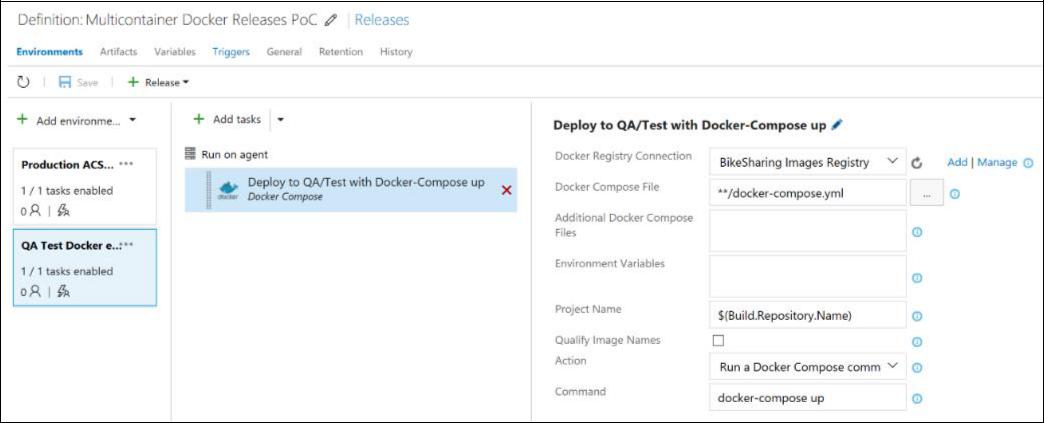
**Figura 5-7:** Adición de una tarea de redacción del estibador en una tubería Visual Studio Team Servicios

Cuando se crea un lanzamiento en Studio Team Visual Servicios, se necesita un conjunto de artefactos de entrada. Estos están destinados a ser inmutable durante toda la vida de la liberación a través de múltiples entornos. Al introducir los contenedores, los artefactos de entrada a identificar imágenes en un registro de implementar. Dependiendo de cómo estos se identifican, que no se garantiza que permanecerá igual durante toda la duración de la liberación, siendo el caso más evidente cuando hace referencia a “myimage: últimas” de un archivo de composición del cargador de muelle.

La extensión del estibador de Studio Team Services visuales le da la capacidad de generar artefactos de construcción que contienen resúmenes de imagen de registro específicos que están garantizados para identificar de forma exclusiva la misma imagen binaria. Estos son lo que realmente desea utilizar como entrada para un lanzamiento.

**La gestión de las emisiones a entornos acoplables mediante el uso de administración de la versión Visual Studio Team Servicios**

A través de las extensiones de Visual Studio Team Services, se puede construir una nueva imagen, publicarla en un registro del estibador, ejecutarlo en anfitriones Linux o Windows, y utilizar comandos como cargador de muelle de composición del implementar varios contenedores como toda una aplicación, todo a través de las capacidades de Visual Studio Team Services Gestión de la Entrega destinados a múltiples entornos, como se muestra en la Figura 5-8.



**Figura 5-8:** Configuración de las tareas de Visual Studio Team Services acoplable de redacción de Studio Team Services VisualesGestión de la liberación

Sin embargo, tenga en cuenta que el escenario que se muestra en la Figura 5 -6 e implementado en la Figura 5-8 es bastante básico (que está desplegando a los hosts Docker simples y máquinas virtuales, y habrá un solo contenedor o instancia por imagen) y probablemente debería ser utilizado sólo para el desarrollo o pruebas escenarios. En la mayor parte de la empresa

53 CAPÍTULO 5 | Acoplable DevOps aplicaciones de flujo de trabajo con las herramientas de Microsoft

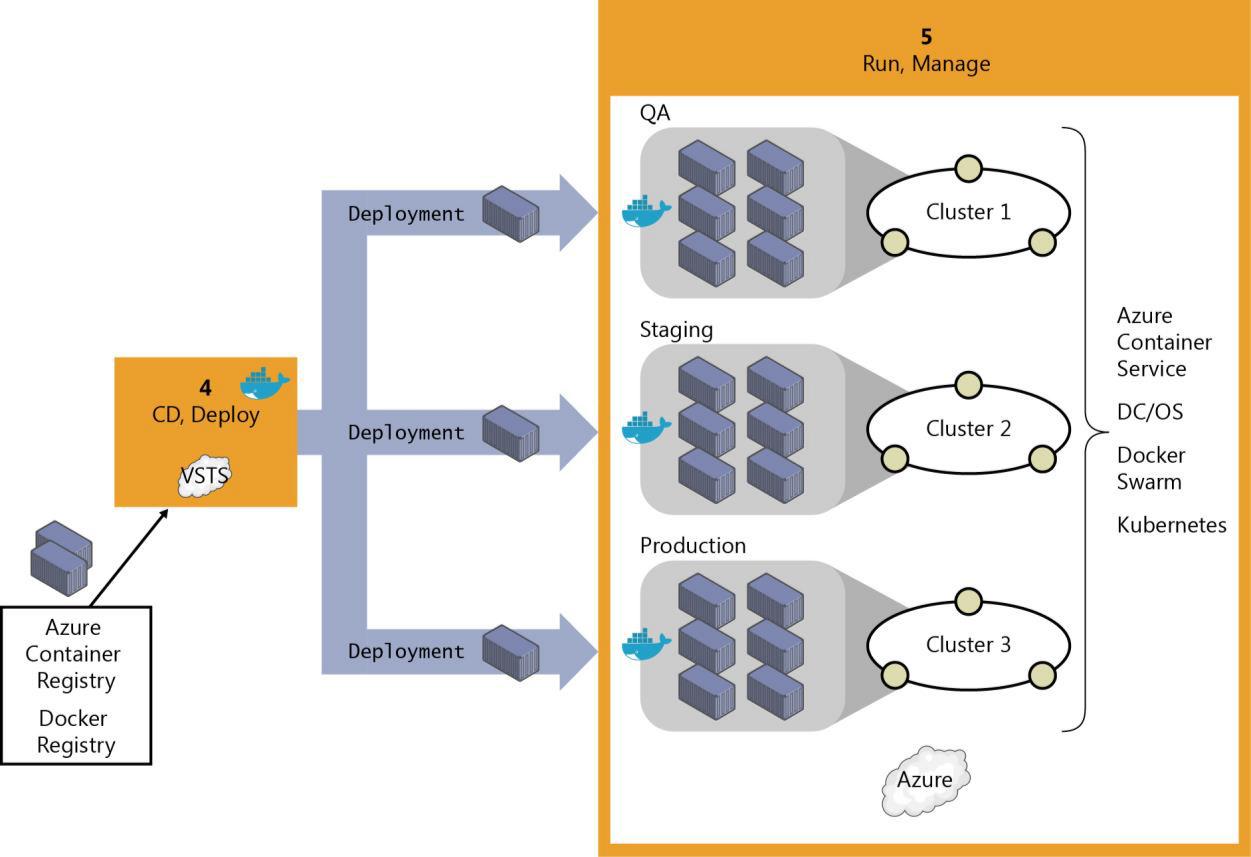
escenarios de producción, que querrían tener alta disponibilidad (HA) y fácil de gestionar la escalabilidad al equilibrio de carga entre múltiples nodos, servidores y máquinas virtuales, además de “conmutaciones por error inteligentes” de modo que si un servidor o nodo falla, sus servicios y contenedores será trasladado a otro servidor o de la máquina virtual. En ese caso, es necesario tecnologías más avanzadas como racimos de contenedores, orchestrators, y programadores. Por lo tanto, la manera de desplegar a esas agrupaciones es precisamente a través de los escenarios avanzados explican en la siguiente sección.

**La implementación de aplicaciones complejas acoplables a Döcker racimos (CC / CA, Kubernetes y acoplable Swarm)**

La naturaleza de las aplicaciones distribuidas requiere recursos informáticos que se distribuyen también. Para tener capacidades de producción a gran escala, es necesario tener capacidades de clustering que proporcionan una alta escalabilidad y HA en base a los recursos comunes.

Se podría implementar manualmente los contenedores a los racimos de una herramienta CLI como estibador Swarm (como usar [Servicio ventana acoplable a crear)](https://docs.docker.com/engine/swarm/swarm-tutorial/deploy-service/) o una interfaz web como [Maratón mesosfera](https://mesosphere.github.io/marathon/docs/marathon-ui.html) para los clústeres / OS DC, pero que debe reservar sólo para probar la distribución puntual o con fines de gestión como los propósitos de monitoreo de escala de salida o.

Desde el punto de vista de CD, y Studio Servicios Visuales del equipo en concreto, puede ejecutar especialmente hecho tareas de despliegue de sus entornos de Visual Studio Team Services Gestión de la Entrega que desplegar sus aplicaciones en contenedores a grupos distribuidos en Contenedor de servicios, como se ilustra en la Figura 5-9 .

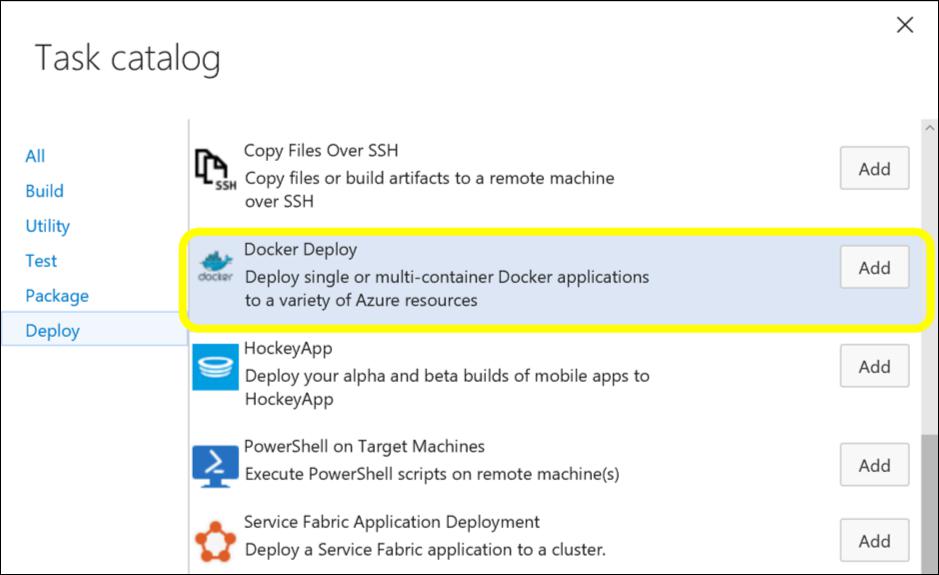


**Figura 5-9:** La implementación de aplicaciones distribuidas a Container Service

Inicialmente, cuando se despliega a ciertos grupos o orchestrators, se utiliza tradicionalmente guiones y mecanismos de despliegue específicos para cada Orchestrator (es decir, Mesosphere DC / OS o Kubernetes tienen diferentes mecanismos de despliegue que acoplable y acoplable Swarm) en lugar de la más simple y

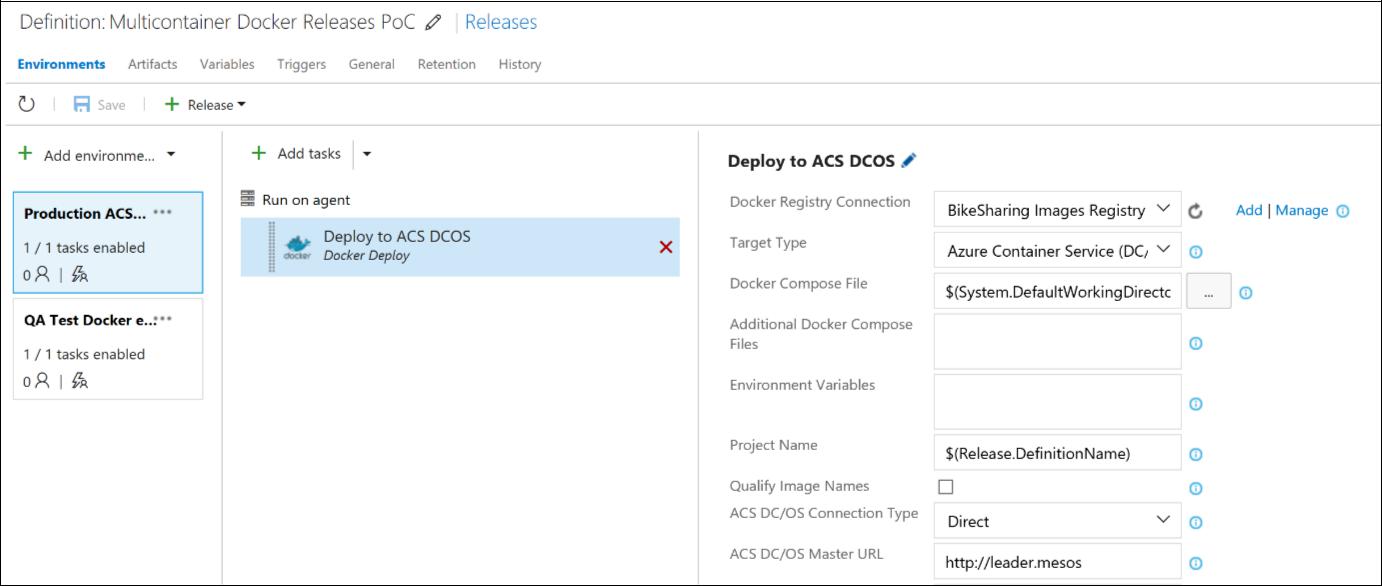
54 CAPÍTULO 5 | Acoplable DevOps aplicaciones de flujo de trabajo con las herramientas de Microsoft

fácil de usar herramienta de composición del cargador de muelle basándose en el archivo de definición de ventana acoplable-compose.yml. Sin embargo, gracias a la tarea de Microsoft Visual Studio Team Servicios del estibador de Despliegue, que se muestra en la figura 5-10, que ahora también se pueden implementar en CC / CA con sólo usar su archivo familiarizado ventana acoplable-compose.yml porque Microsoft realiza esa “traducción” para usted (de su archivo de cargador de muelle-compose.yml a otros formatos requeridos por CC / CA).



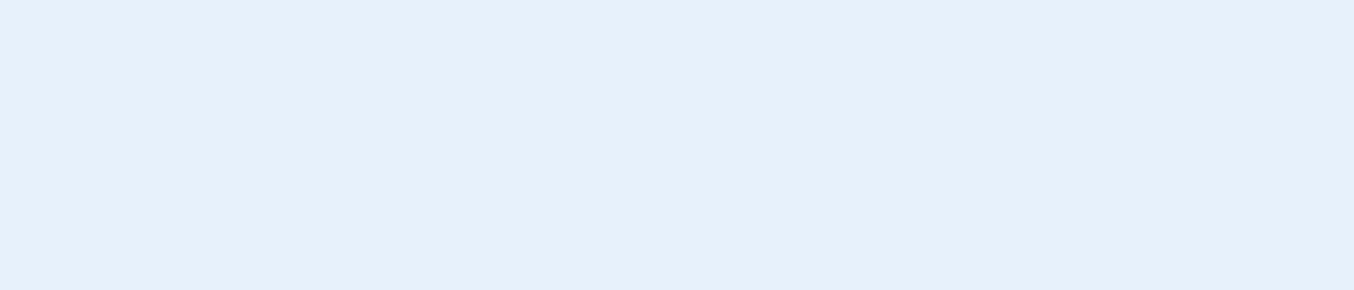
**Figura 5-10:** La adición de la tarea del estibador despliegan en una RM Medio Ambiente

La figura 5-11 muestra cómo se puede editar la tarea del estibador Implementar y especificar el tipo de destino (azul Contenedor de Servicio CC / CA, en este caso), el estibador Componer Archivo, y la conexión acoplable Registro (como Azure Registro de contenedores o acoplable Hub) . Aquí es donde la tarea va a recuperar sus imágenes personalizadas listas para su uso acoplables a desplegarse como recipientes de la agrupación de CC / CA.



**Figura 5-11:** Acoplable Implementar la definición de tareas despliegue de Azure Contenedor de Servicio CC / CA

55 CAPÍTULO 5 | Acoplable DevOps aplicaciones de flujo de trabajo con las herramientas de Microsoft



**Más información** Para leer más sobre la tubería CD con Visual Studio Team Servicios y estibador, visite los siguientes sitios:

extensión de Visual Studio Team Servicios de estibador y Azure Container Service: [https://aka.ms/](https://aka.ms/vstsdockerextension) [vstsdockerextension](https://aka.ms/vstsdockerextension)

Azure Container Service: <https://aka.ms/azurecontainerservice>

Mesosfera DC / OS: <https://mesosphere.com/product/>

**Paso 5: Ejecutar y gestionar**

Debido a ejecutar y administrar aplicaciones a nivel de empresa-producción es un tema importante en sí mismo, y debido al tipo de operaciones y las personas que trabajan en ese nivel (operaciones de TI), así como el amplio alcance de esta área, hemos dedicado el toda siguiente capítulo para explicarla.

**Paso 6: Monitorear y diagnosticar**

Este tema también se trata en el capítulo siguiente, como parte de las tareas que lleva a cabo las operaciones de TI en sistemas de producción; Sin embargo, es importante destacar que los conocimientos obtenidos en este paso deben alimentar de nuevo al equipo de desarrollo para que la aplicación se mejora constantemente. Desde ese punto de vista, sino que también es parte de DevOps, aunque las tareas y operaciones se realizan generalmente por ella.

Sólo cuando la vigilancia y el diagnóstico son 100 por ciento en el ámbito de DevOps son los procesos de seguimiento y análisis realizados por el equipo de desarrollo frente a entornos de prueba o beta. Esto se realiza mediante la realización de pruebas de carga o simplemente mediante el control de entornos beta o de control de calidad, en los probadores beta están tratando las nuevas versiones.

56 CAPÍTULO 5 | Acoplable DevOps aplicaciones de flujo de trabajo con las herramientas de Microsoft

CAPÍTULO 6

Corriendo,

la gestión y

supervisión

Estibador

producción

entornos

Visión: Las aplicaciones empresariales necesitan para funcionar con alta disponibilidad y alta escalabilidad; operaciones de TI deben ser capaces de gestionar y controlar los entornos y las propias aplicaciones.

57 CAPÍTULO 6 | Correr, gestión y seguimiento de los entornos de producción Docker

Este último pilar en el ciclo de vida de aplicaciones en contenedores acoplable se centra en cómo puede ejecutar, gestionar y controlar sus aplicaciones en entornos de producción escalables y de alta disponibilidad (HA).

La forma de ejecutar sus aplicaciones en contenedores en la producción (arquitectura de infraestructura y tecnologías de plataforma) está también muy relacionado y completamente fundada en las plataformas de arquitectura y desarrollo elegidas que hemos examinado en el capítulo 1 de este libro electrónico. Este capítulo examina los productos y tecnologías de Microsoft y otros proveedores específicos que se pueden utilizar para ejecutar con eficacia, aplicaciones altamente escalables HA distribuidos, además de cómo se pueden gestionar y controlar desde el punto de vista informático.

La ejecución de aplicaciones compuestas y basados ​​microservicios-en entornos de producción

Aplicaciones compuestas por múltiples microservicios necesitan ser desplegados en grupos Orchestrator con el fin de simplificar la complejidad de la implementación y que sea viable desde el punto de vista informático. Sin un clúster orquestador, sería muy difícil de implementar y de crecimiento modular una aplicación microservicios compleja.

**Introducción a orquestadores, programadores, y los racimos de contenedores**

Al principio de este libro electrónico, introdujimos racimos y programadores como parte de la discusión sobre la arquitectura de software y desarrollo. Ejemplos de grupos acoplables son acoplable Swarm y Mesosphere centro de datos del Sistema Operativo (DC / OS). Ambos pueden funcionar como parte de la infraestructura proporcionada por Microsoft Azure servicio de contenedores.

Cuando las aplicaciones se escalan de salida a través de múltiples sistemas host, la capacidad de gestionar cada sistema anfitrión y abstraer la complejidad de la plataforma subyacente se convierte en atractivo. Eso es precisamente lo orchestrators y programadores proporcionan. Vamos a echar un breve vistazo a ellos aquí:

* **programadores** “Programación” se refiere a la capacidad de un administrador para cargar un archivo de servicio en un sistema host que establece cómo ejecutar un contenedor específico. Lanzamiento de contenedores en un clúster acoplable tiende a ser conocido como la programación. Aunque la programación se refiere al acto específico de la carga de la definición del servicio, en un sentido más general, los programadores son responsables de enganchar en sistema de inicio de una serie de gestionar los servicios en cualquier capacidad necesaria.

Un planificador de clúster tiene varios objetivos: el uso de los recursos del cluster de manera eficiente, trabajar con restricciones de posición suministrados por el usuario, la programación de aplicaciones rápidamente para no dejarlos en un estado pendiente, con un grado de “equidad”, siendo robusto a errores, y estar siempre disponibles .

* **Orquestación** Plataformas se extienden las capacidades de gestión del ciclo de vida a las cargas de trabajo, multicontainer complejas desplegadas en un clúster de hosts. Mediante la abstracción de la infraestructura de acogida, herramientas de orquestación ofrecen a los usuarios una manera de tratar todo el clúster como un destino de despliegue único.

El proceso de la orquestación implica utillaje y una plataforma que puede automatizar todos los aspectos de gestión de la aplicación de la colocación inicial o el despliegue por contenedor; recipientes a diferentes hosts en función de la salud o el rendimiento de su anfitrión en movimiento; de versiones y actualizaciones de rodadura y vigilancia de la salud que apoyan las funciones de escala y de conmutación por error; y muchos más.

Orquestación es un término amplio que se refiere a la programación de contenedores, gestión del cluster, y posiblemente el aprovisionamiento de los ejércitos adicionales.

58 CAPÍTULO 6 | Correr, gestión y seguimiento de los entornos de producción Docker

Las capacidades proporcionadas por orchestrators y programadores son muy complejos para desarrollar y crear desde cero, y por lo tanto por lo general quieren hacer uso de soluciones de orquestación ofrecidos por los proveedores.

La gestión de entornos de producción Docker

gestión de clústeres y orquestación es el proceso de control de un grupo de hosts. Esto puede implicar la adición y eliminación hosts de un clúster, la obtención de información sobre el estado actual de los ejércitos y contenedores, y de partida y procesos de parar. la gestión de clusters y orquestación están estrechamente ligado a la programación debido a que el programador debe tener acceso a cada host del clúster, para marcar servicios. Por esta razón, la misma herramienta se utiliza a menudo para ambos propósitos.

**Servicio de contenedores y herramientas de gestión**

Servicio de contenedor proporciona un rápido despliegue de soluciones de código abierto de agrupamiento de contenedores y orquestación populares. Utiliza imágenes acoplables para asegurar que sus contenedores de aplicaciones son totalmente portátil. Mediante el uso de servicio de contenedores, puede implementar CC / CA (alimentado por la mesosfera y Apache mesos) y racimos acoplable Swarm con plantillas Resource Manager Azure o el portal de Azure para asegurar que se puede escalar estas aplicaciones a miles-incluso decenas de miles de contenedores .

Implementar estos grupos mediante el uso de conjuntos de escala Azure Virtual Machine, y los grupos de aprovechar las ofertas de red y almacenamiento de Azure. Para tener acceso a servicio de contenedores, necesita una suscripción Azure. Con servicio de contenedores, se puede tomar ventaja de las características de nivel empresarial de Azure, manteniendo la portabilidad de aplicaciones, incluso en las capas de orquestación.

La Tabla 6-1 enumera las herramientas de gestión comunes relacionados con su orchestrators, programadores, y la plataforma de la agrupación.

**Tabla 6-1:** herramientas de gestión de ventana acoplable

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Herramientas administrativas** |  | **Descripción** | | | | | |  | **orchestrators relacionados** |
|  |  |  |  |  |  |  |  |  |  |
| Servicio de contenedores |  | [Servicio de contenedores](https://azure.microsoft.com/en-us/services/container-service/) proporciona una forma fácil de conseguir | | | | | |  | Mesosfera CC / CA |
|  |  |  |  |  |  |  |  |  |  |
| (Gestión de la interfaz de usuario en Azure |  | comenzado manera de [desplegar un contenedor-cluster en](https://azure.microsoft.com/en-us/documentation/articles/container-service-deployment/) | | | | | |  | Kubernetes |
|  |  |  |  |  | |  |  |  |  |
| portal) |  | [Azur](https://azure.microsoft.com/en-us/documentation/articles/container-service-deployment/) basado en orchestrators populares como | | | | | |  | acoplable Swarm |
|  |  |  |  | | |  | |  |  |
|  |  | Mesosfera CC / CA, y Kubernetes | | | | | |  |  |
|  |  | Acoplable Swarm. | | | | | |  |  |
|  |  | Servicio de contenedores optimiza el | | | | | |  |  |
|  |  | configuración de esas plataformas. usted acaba | | | | | |  |  |
|  |  | necesitará seleccionar el tamaño, el número de | | | | | |  |  |
|  |  | anfitriones, y la elección de las herramientas, y orquestador | | | | | |  |  |
|  |  | Servicio de contenedores se encarga de todo lo demás. | | | | | |  |  |
|  |  |  | | | | | |  |  |
| Control Universal ventana acoplable |  | [Cargador de muelle plano de control universal](https://docs.docker.com/v1.11/ucp/overview/) es el | | | | | |  | acoplable Swarm |
|  |  |  | | | |  | |  |  |
| Avión |  | la gestión de clusters a nivel empresarial | | | | | |  | (Apoyado por Container |
| (En las instalaciones o nube) |  | solución de acoplable. Le ayuda a gestionar | | | | | |  | Servicio) |
|  |  | todo el clúster desde un solo lugar. | | | | | |  |  |
|  |  | Cargador de muelle plano de control universal se incluye | | | | | |  |  |
|  |  | como parte del producto comercial llamado | | | | | |  |  |
|  |  | Acoplable centro de datos que proporciona acoplable | | | | | |  |  |

59 CAPÍTULO 6 | Correr, gestión y seguimiento de los entornos de producción Docker



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Enjambre, Plano de Control Universal y acoplable | | | | | | |  |  |
|  |  | Cargador de muelle de confianza del Registro. | | | | | | |  |  |
|  |  | Acoplable centro de datos se puede instalar en sitio | | | | | | |  |  |
|  |  | locales o aprovisionado de un público | | | | | | |  |  |
|  |  | nube como Azure. | | | | | | |  |  |
|  |  |  |  |  | |  |  |  |  |  |
| Nube ventana acoplable |  | [Nube ventana acoplable](https://docs.docker.com/docker-cloud/overview/) es una gestión organizada | | | | | | | acoplable Swarm |  |
|  |  |  |  |  |  |  |  |  |  |  |
| (Aka Tutum; nube SaaS) |  | de servicio (SaaS) que proporciona orquestación | | | | | | | (Apoyado por Container |  |
|  |  | capacidades y con un registro acoplable | | | | | | | Servicio) |  |
|  |  | construir y probar las instalaciones para Dockerized | | | | | | |  |  |
|  |  | imágenes de aplicaciones, herramientas para ayudarle a configurar | | | | | | |  |  |
|  |  | y gestionar su infraestructura de acogida, | | | | | | |  |  |
|  |  | y características de implementación para ayudarle a | | | | | | |  |  |
|  |  | automatizar el despliegue de sus imágenes a su | | | | | | |  |  |
|  |  | infraestructura de hormigón. Se puede conectar | | | | | | |  |  |
|  |  | su cuenta SaaS Nube acoplable a su | | | | | | |  |  |
|  |  | infraestructura de servicio de contenedores en ejecución | | | | | | |  |  |
|  |  | un clúster acoplable Swarm. | | | | | | |  |  |
|  |  |  | |  | | | | |  |  |
| Maratón mesosfera |  | [Maratón](https://mesosphere.github.io/marathon/docs/marathon-ui.html) es un recipiente de producción de grado | | | | | | | Mesosfera CC / CA |  |
|  |  |  |  |  | |  |  |  |  |  |
| (En las instalaciones o nube) |  | orquestación y planificador plataforma para | | | | | | | (Basado en Apache |  |
|  |  | CC / CA de mesosfera y mesos Apache. | | | | | | | mesos; Apoyado por |  |
|  |  | Funciona con mesos (CC / CA se basa en | | | | | | | Servicio de contenedores) |  |
|  |  |  |  |
|  |  | Apache mesos) para controlar larga ejecución | | | | | | |  |  |
|  |  | servicios y proporciona una [Interfaz web para el proceso de](https://mesosphere.github.io/marathon/docs/marathon-ui.html) | | | | | | |  |  |
|  |  |  | |  | |  |  |  |  |  |
|  |  | [y gestión de contenedores.](https://mesosphere.github.io/marathon/docs/marathon-ui.html) Eso provee un | | | | | | |  |  |
|  |  | herramienta de gestión de interfaz de usuario web | | | | |  | |  |  |
|  |  |  |  |  | |  |  |  |  |  |
| google Kubernetes |  | [Kubernetes](http://kubernetes.io/docs/user-guide/ui/#dashboard-access) palmos orquestación, | | | | | | | google Kubernetes |  |
|  |  |  | |  | | | | |  |  |
|  |  | programación, y la infraestructura de clúster. Es | | | | | | | (Apoyado por Container |  |
|  |  | una plataforma de código abierto para la automatización | | | | | | | Servicio) |  |
|  |  | despliegue, la escala y las operaciones de | | | | | | |  |  |
|  |  | contenedores de aplicaciones a través de grupos de | | | | | | |  |  |
|  |  | hosts, proporcionando recipiente centrada | | | | | | |  |  |
|  |  | infraestructura. | | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

**Azure Servicio Tela**

Otra opción para el clúster de despliegue y gestión es Azure Servicio de Tela. [Tela servicio](https://azure.microsoft.com/en-us/services/service-fabric/) es una plataforma microservicios de Microsoft que incluye la orquestación de contenedores, así como los modelos de programación de desarrolladores para crear aplicaciones microservicios altamente escalables. Servicio Tela apoya Estibador en los actuales versiones previas de Linux, como en el[Servicio Tela vista previa en Linux,](https://azure.microsoft.com/en-us/documentation/articles/service-fabric-linux-overview/) y para Windows Contenedores [en la próxima versión.](https://azure.microsoft.com/en-us/documentation/articles/service-fabric-containers-overview/)

60 CAPÍTULO 6 | Correr, gestión y seguimiento de los entornos de producción Docker

A continuación se presentan las herramientas de gestión de servicio de la tela:

* [portal de Azure para el Servicio Tela](https://azure.microsoft.com/en-us/documentation/articles/service-fabric-cluster-creation-via-portal/) las operaciones relacionadas con el clúster (crear / actualizar / borrar) un clúster o configurar su infraestructura (VM, equilibrador de carga, redes, etc.)
* [Azure Servicio Tela Explorador](https://azure.microsoft.com/en-us/documentation/articles/service-fabric-visualizing-your-cluster/) es una herramienta de interfaz de usuario web especializado que proporciona información detallada y cierta operaciones en el clúster de servicio de la tela desde los nodos / VM punto de vista y desde el punto de vista de aplicación y servicios.

Supervisión de los servicios de aplicaciones en contenedores

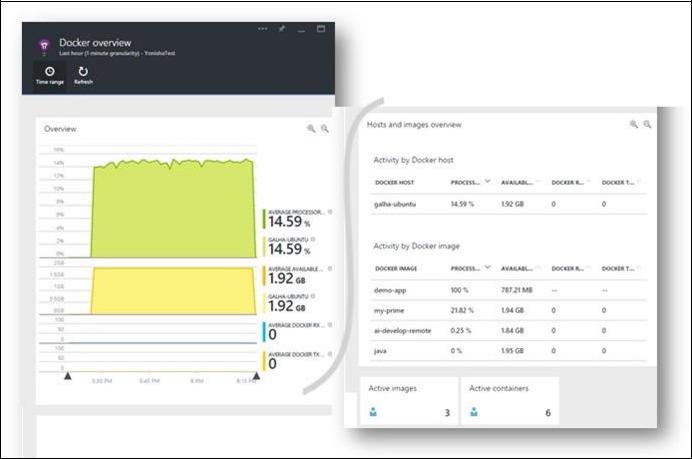
Es crítico para las aplicaciones divididas en varios contenedores y microservicios tener una forma de supervisar y analizar el comportamiento de la aplicación.

**Microsoft Application Insights**

[Perspectivas de aplicación](https://azure.microsoft.com/en-us/documentation/articles/app-insights-overview/) es un servicio de análisis extensible que supervisa su aplicación en vivo. Esto te ayuda para detectar y diagnosticar problemas de rendimiento y de entender lo que los usuarios realmente hacen con su aplicación. Está diseñado para los desarrolladores, con la intención de ayudar a mejorar continuamente el rendimiento y la facilidad de uso de sus servicios o aplicaciones. Insights aplicación funciona con ambos Web / servicios y aplicaciones independientes en una amplia variedad de plataformas como .NET, Java, Node.js y muchas otras plataformas, alojados en las instalaciones o en la nube.

**Analizando acoplable aplicaciones en entornos de control de calidad usando Insights de aplicación**

Lo que se refiere a estibador, se puede trazar eventos del ciclo de vida y contadores de rendimiento de contenedores acoplables a consultar Insights. Sólo tiene que ejecutar el[Insights imagen acoplable aplicación](https://hub.docker.com/r/microsoft/applicationinsights/) como un contenedor en su huésped, y se mostrará contadores de rendimiento para el anfitrión, así como para las otras imágenes acoplables. Esta imagen Aplicación Insights Docker (Figura 6-1) le ayuda a controlar sus aplicaciones en contenedores mediante la recopilación de telemetría sobre el rendimiento y la actividad de su proveedor de alojamiento acoplable (es decir, sus máquinas virtuales Linux), contenedores de Docker y las aplicaciones que se ejecutan dentro de ellos.



**Figura 6-1:** Insights aplicación de monitoreo de hosts y contenedores acoplables

61 CAPÍTULO 6 | Correr, gestión y seguimiento de los entornos de producción Docker

Cuando se ejecuta el [Insights imagen acoplable aplicación](https://hub.docker.com/r/microsoft/applicationinsights/) en el host del estibador, se beneficiará de lo siguiente:

* -Ciclo de vida de telemetría sobre todos los contenedores que se ejecutan en el host-arranque, parada, y así sucesivamente.
* Contadores de rendimiento para todos los contenedores: CPU, memoria, uso de la red, y mucho más.
* Si también ha instalado [Insights aplicación SDK](https://azure.microsoft.com/en-us/documentation/articles/app-insights-asp-net/) en las aplicaciones que se ejecutan en los contenedores, toda la telemetría de esas aplicaciones tendrá propiedades adicionales que identifican la máquina recipiente y host. Así, por ejemplo, si tiene instancias de una aplicación que se ejecuta en más de un huésped, usted fácilmente podrá filtrar su aplicación telemetría por el anfitrión.

**La creación de Insights de aplicación para monitorear aplicaciones acoplables y anfitriones Docker**

Para crear un recurso Insights aplicación, siga las instrucciones de los artículos presentados en la lista que sigue. Azure Portal creará el script necesario para usted.

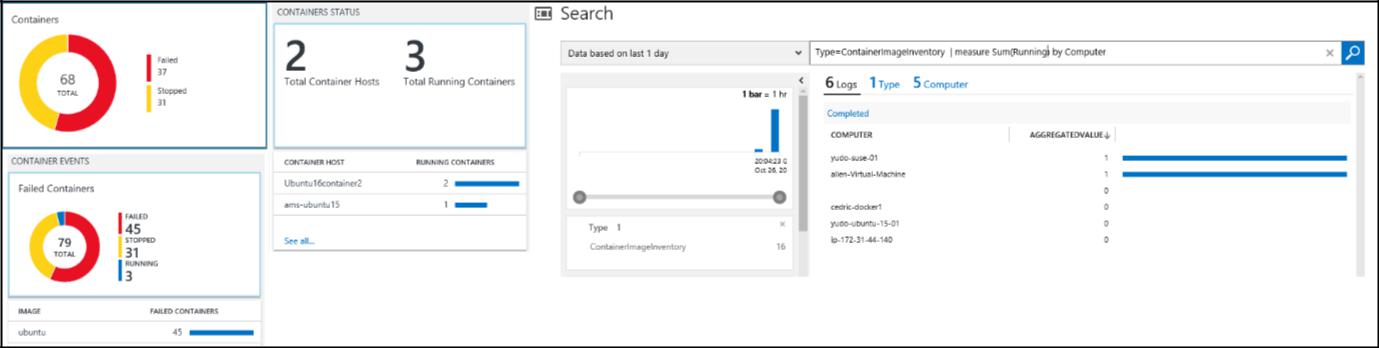
* **Monitorear aplicaciones acoplables en la Solicitud Insights:** [https://azure.microsoft.com/](https://azure.microsoft.com/documentation/articles/app-insights-docker/)[documentación / articulos / app-ideas-ventana acoplable /](https://azure.microsoft.com/documentation/articles/app-insights-docker/)
* **Aplicación de imágenes Insights acoplable al estibador Hub y Github:** [https://hub.docker.com/](https://hub.docker.com/r/microsoft/applicationinsights/)[r / Microsoft / applicationinsights /](https://hub.docker.com/r/microsoft/applicationinsights/) y <https://github.com/Microsoft/ApplicationInsights-Docker>
* **Configurar Insights de aplicación en ASP.NET:** [https://azure.microsoft.com/documentation/](https://azure.microsoft.com/documentation/articles/app-insights-asp-net/)[artículos / APP-ideas-asp-net /](https://azure.microsoft.com/documentation/articles/app-insights-asp-net/)
* **Insights aplicaciones para páginas web:** [https://azure.microsoft.com/documentation/articles/app-ideas-javascript /](https://azure.microsoft.com/documentation/articles/app-insights-javascript/)

**Suite de Gestión de Operaciones de Microsoft**

[Suite de Gestión de Operaciones](http://microsoft.com/oms) es una solución de administración de TI simplificada que proporciona análisis de registro, automatización, copia de seguridad y recuperación del sitio. Residencia en[consultas](https://blogs.technet.microsoft.com/msoms/2016/01/21/easy-microsoft-operations-management-suite-search-queries/) en las operaciones de Management Suite, puede aumentar [alertas](https://azure.microsoft.com/en-us/documentation/articles/operations-management-suite-monitoring-alerts/) y establecer remediación vía [Azure Automatización.](https://azure.microsoft.com/en-us/documentation/services/automation/) También se integra sin problemas con sus soluciones de gestión existentes para proporcionar una única vista del panel de vidrio. Operaciones Management Suite le ayuda a gestionar y proteger sus en las instalaciones e infraestructuras de nube.

[**Suite de Gestión de Operaciones**](http://microsoft.com/oms) **Envase Solución para Docker**

Además de proporcionar servicios valiosos por sí solo, la solución de contenedores de Gestión de Operaciones Suite puede gestionar y controlar los ejércitos y contenedores acoplables al mostrar información acerca de dónde están sus contenedores y anfitriones de contenedores son, que los contenedores se están ejecutando o fracasaron y Docker de demonio y contenedores registros enviados a stdout y stderr. También muestra las métricas de rendimiento, tales como CPU, memoria, red y almacenamiento para el envase y acoge para ayudarle a solucionar y encontrar contenedores vecino ruidoso.



**Figura 6-2:** Información sobre contenedores Docker mostrado por Operaciones Management Suite

62 CAPÍTULO 6 | Correr, gestión y seguimiento de los entornos de producción Docker

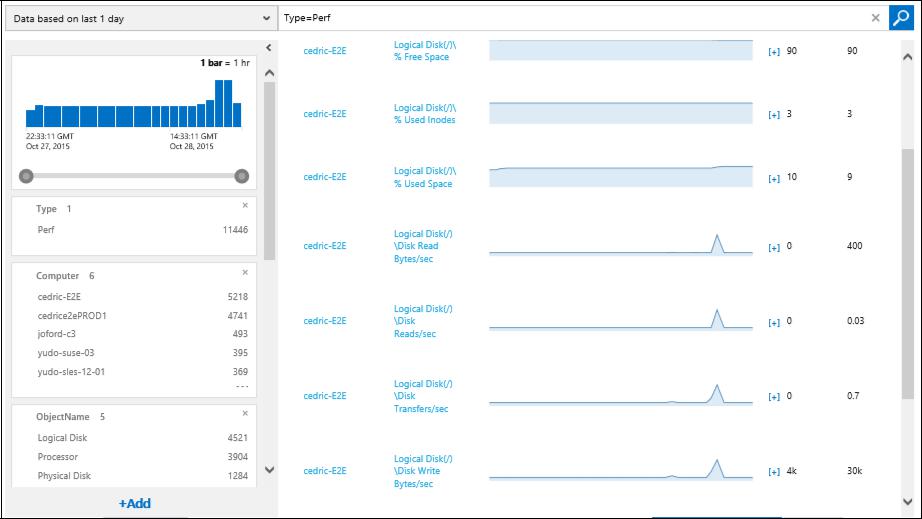
Insights de aplicaciones y operaciones de Management Suite ambas se centran en las actividades de vigilancia; Sin embargo, la aplicación Insights se centra más en el seguimiento de las mismas aplicaciones gracias a su SDK corriendo dentro de la aplicación. Sin embargo, las operaciones de Management Suite se centra mucho más en la infraestructura alrededor de los anfitriones, además de que ofrece un análisis profundo de los registros a escala al tiempo que proporciona un sistema de búsqueda / consulta basada en datos muy flexible.

Dado que las operaciones de Management Suite se implementa como un servicio basado en la nube, se puede tener en funcionamiento rápidamente con una mínima inversión en servicios de infraestructura. Las nuevas características se entregan automáticamente, ahorrándole de mantenimiento y costes de actualización.

El uso de soluciones de Gestión de Operaciones Suite de contenedor, puede hacer lo siguiente:

* Centralizar y correlacionar millones de registros de contenedores acoplables a escala
* Ver información sobre todos los hosts de contenedores en un solo lugar
* Saben qué contenedores se están ejecutando, lo que la imagen que se está ejecutando, y donde se están ejecutando
* diagnosticar rápidamente contenedores “ruidoso vecino” que puede causar problemas en los ejércitos de contenedores
* Ver un registro de auditoría de las acciones en los contenedores
* Solucionar problemas de visión y de buscar registros centralizados y sin interacción remota a los anfitriones Docker
* Encuentra los contenedores que podrían ser “vecinos ruidosos” y que consumen un exceso de recursos en un host
* Ver CPU centralizada, memoria, almacenamiento y uso de la red y la información de rendimiento para contenedores
* Generar contenedores acoplables prueba con Azure Automation

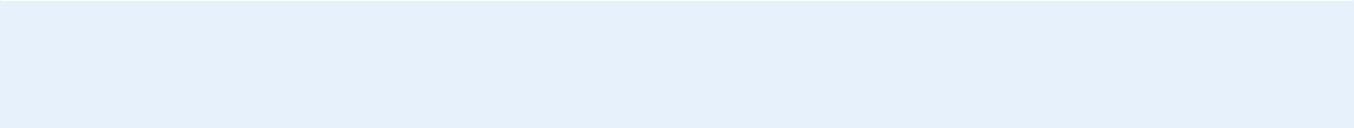
Se puede ver la información de rendimiento mediante la ejecución de consultas como Type = Perf, Como se muestra en la Figura 6-3.



**Figura 6-3:** métricas de rendimiento de Docker anfitriones muestran mediante operaciones de Management Suite

63 CAPÍTULO 6 | Correr, gestión y seguimiento de los entornos de producción Docker

Ahorro de consultas es también una característica estándar en las operaciones de Management Suite y puede ayudar a mantener las consultas que hayas encontrado útil y descubrir las tendencias en su sistema.



**Más información** Para encontrar información sobre cómo instalar y configurar la solución de contenedores en acoplable [Suite de Gestión de Operaciones,](http://microsoft.com/oms) ir a [https://azure.microsoft.com/documentation/articles/log -](https://azure.microsoft.com/documentation/articles/log-analytics-containers/) [Analytics-contenedores /.](https://azure.microsoft.com/documentation/articles/log-analytics-containers/)

64 CAPÍTULO 6 | Correr, gestión y seguimiento de los entornos de producción Docker

CAPÍTULO 7

conclusiones

comida para llevar clave

* soluciones basadas en contenedores proporcionan importantes beneficios de ahorro de costes debido a los contenedores son una solución a los problemas de implementación causados ​​por la falta de dependencias en los entornos de producción, por lo tanto, la mejora de DevOps y las operaciones de producción de manera significativa.
* Docker se está convirtiendo en el estándar de facto en la industria del envase, con el apoyo de los proveedores más importantes en los ecosistemas Linux y Windows, incluyendo Microsoft. En el futuro, acoplable será omnipresente en cualquier centro de datos en la nube o en las instalaciones.
* Un contenedor acoplable está convirtiendo en la unidad estándar de despliegue para cualquier aplicación o servicio basado en servidor.
* orchestrators Docker como los proporcionados en Azure Container Service (Datacenter Operating System Mesosfera (CC / CA), acoplable Swarm, Kubernetes) y Azure Servicio de Tela son fundamentales e indispensables para cualquier aplicación basada microservicios-o multicontainer con las necesidades de escalabilidad y complejidad significativos.
* Un entorno DevOps de extremo a extremo que soporta la integración de despliegue continuo / continuo que conecta a la producción ambientes ventana ofrece agilidad y en última instancia, mejora el tiempo de comercialización de sus aplicaciones.

Studio Team Services visuales simplifica en gran medida su entorno DevOps designado para entornos acoplables de sus tuberías continuas de implementación, incluyendo entornos Docker simples o orchestrators MICROSERVICE y contenedores más avanzadas basadas en Azure.

sesenta y cinco CAPÍTULO 7 | conclusiones