

LOS CONTENEDORES Y KUBERNETES IMPULSAN LA TRANSFORMACIÓN DIGITAL

Sistema de Kubernetes de gestión de las aplicaciones contenedorizadas para ofrecer valor a la empresa

Contenido

Introducción.....	3
La transformación digital y el cambio a las aplicaciones contenedorizadas.....	3
Aplicaciones nativas de cloud y aplicaciones de 12 factores.....	4
Metodología para ofrecer software como servicio.....	5
Descripción concisa de Kubernetes	6
Modelo de objetos de Kubernetes	7
Mantenimiento del estado deseado	7
El valor de Kubernetes para la empresa	8
Kubernetes para aplicaciones nativas de cloud y aplicaciones de 12 factores.....	8
Un ejemplo de caso de uso	10
Soluciones de tecnologías de contenedores de VMware.....	11
vSphere Integrated Containers	11
Wavefront by VMware.....	13
Pivotal Container Service.....	13
Conclusión	14

DEFINICIÓN DE CONTENEDOR DE DOCKER

Con los contenedores, Docker ha definido un formato estándar para empaquetar y portar software, de manera similar a como los contenedores ISO definen una norma para el transporte de carga. Un contenedor es una instancia de tiempo de ejecución de una imagen de Docker y consta de tres partes:

- Una imagen de Docker.
- Un entorno en el que se ejecuta la imagen.
- Un conjunto de instrucciones para ejecutar la imagen.

—Contenido adaptado del [glosario de Docker](#)

Introducción

Kubernetes gestiona contenedores. Los contenedores empaquetan las aplicaciones y sus dependencias en una imagen distribuible que puede ejecutarse en cualquier lugar, optimizando el desarrollo y la implementación de software. La adopción de contenedores permite a las organizaciones dar un paso fundamental para convertirse en empresas digitales ágiles centradas en acelerar la distribución de productos, servicios y experiencias de cliente que se caracterizan por la innovación. Las empresas pueden tomar la iniciativa en vez de verse sorprendidas.

Pero los contenedores también presentan problemas de gestión de la tecnología, especialmente cuando es necesario implementar y gestionar aplicaciones contenedorizadas según las necesidades. Con Kubernetes se pueden solucionar. Kubernetes coordina estas aplicaciones contenedorizadas para gestionar y automatizar la utilización de los recursos, la gestión de fallos, la disponibilidad, la configuración, la escalabilidad y el estado deseado.

En este artículo se describe Kubernetes, se explica el valor que aporta a la empresa, se exploran sus casos de uso y se ilustra cómo puede acelerar la transformación digital de su organización.

La transformación digital y el cambio a las aplicaciones contenedorizadas

Según The New York Times, la velocidad a la que crece y se extiende la innovación tecnológica está aumentando.¹ Como consecuencia, la transformación digital se ha convertido en un claro objetivo para muchas empresas, y la adopción de iniciativas digitales se ha generalizado.²

Los motivos por los que las empresas están experimentando la transformación digital están claros:

- Crear nuevas aplicaciones que capten clientes de formas innovadoras y atractivas.
- Mejorar las operaciones para ofrecer a la empresa mejores productos y servicios de manera más eficaz y a un precio más asequible.
- Generar nuevos flujos de ingresos mediante una rápida adaptación a los cambios de las condiciones del mercado y las preferencias del consumidor.

Según Gartner Research: «El futuro pertenece a las empresas que logren crear las soluciones de software más eficaces, inteligentes y autónomas».³ Pero los ingredientes para crear aplicaciones eficaces y autónomas no están tan claros como los resultados deseados.

Para ser eficaces en esta era, las aplicaciones necesitan una arquitectura que fomente un desarrollo y una implementación fluidos, rápidos y dinámicos, sin renunciar a la seguridad, el rendimiento y la rentabilidad de los patrones establecidos. Los contenedores proporcionan la base de una nueva arquitectura de aplicaciones que contribuye a la transformación digital y prepara el terreno para la innovación.

1 «[Digital Transformation Going Mainstream in 2016, IDC Predicts](#)», Steve Lohr, The New York Times, 4 de noviembre de 2015.

2 «[New Research Finds Investment from Outside IT Is Key to Digital Transformation Success](#)», de Business Wire, The New York Times, 11 de mayo de 2017.

3 «[Digital Transformation Going Mainstream in 2016, IDC Predicts](#)», Steve Lohr, The New York Times, 4 de noviembre de 2015.

Cada vez hay más empresas que adoptan la tecnología de contenedores. Una reciente encuesta de 451 Research reveló un perfil de implementación impresionante para un ecosistema emergente.⁴ Las organizaciones que adoptan los contenedores consideran que son una vía rápida para crear e implementar aplicaciones nativas de cloud y aplicaciones de doce factores.

Aplicaciones nativas de cloud y aplicaciones de 12 factores

La Cloud Native Computing Foundation, que es un proyecto de The Linux Foundation, define las aplicaciones nativas de cloud de la manera siguiente:⁵

1. **Contenedorizadas:** cada parte (aplicaciones, procesos, etc.) se empaqueta en su propio contenedor. Esto facilita la reproducibilidad, la transparencia y el aislamiento de recursos.
2. **Coordinadas dinámicamente:** los contenedores se planifican y gestionan de forma activa para optimizar la utilización de recursos.
3. **Orientadas a los microservicios:** las aplicaciones se segmentan en microservicios. Esta segmentación aumenta considerablemente la agilidad global y la facilidad de mantenimiento de las aplicaciones.

Kubernetes cubre la segunda parte de la definición proporcionando planificación y gestión de contenedores. En cuanto a la tercera parte, tanto Kubernetes como Docker ayudan a implementar microservicios.

Sin embargo, el elemento clave es el contenedor: un proceso que se ejecuta en un ordenador o una máquina virtual, con una aplicación, un sistema de archivos y una red aislados que se autodescriben. Un contenedor empaqueta una aplicación de manera reproducible: se puede distribuir y reutilizar de manera sencilla.

Los contenedores más implementados son los de Docker. Un manifiesto, llamado Dockerfile, describe cómo deben ejecutarse en un host la imagen y sus componentes en un contenedor. Para ilustrar la relación entre Dockerfile y la imagen, veamos un ejemplo de Dockerfile que instala MongoDB en una máquina Ubuntu que se ejecuta en un contenedor. Las líneas que empiezan con el símbolo de almohadilla son comentarios que describen los comandos que van a continuación.

```
# MongoDB Dockerfile from https://github.com/dockerfile/mongodb
# Pull base image.
FROM dockerfile/ubuntu
# Install MongoDB.
RUN \
    apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv
    7F0CEB10 && \
    echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-
    upstart dist 10gen' > /etc/apt/sources.list.d/mongodb.list && \
    apt-get update && \
    apt-get install -y mongodb-org && \
    rm -rf /var/lib/apt/lists/*
# Define mountable directories.
```

⁴ «Application containers will be a \$2.7bn market by 2020, representing a small but high-growth segment of the CloudEnabling Technologies market», 451 Research, 10 de enero de 2017.

⁵ Esta definición es de la sección de preguntas frecuentes de la Cloud Native Computing Foundation, <https://www.cncf.io/about/faq/>.

```

VOLUME ["/data/db"]
# Define working directory.

WORKDIR /data
# Define default command.
CMD ["mongod"]
# Expose port 27017 for the process and port 28017 for http
EXPOSE 27017
EXPOSE 28017

```

Metodología para ofrecer software como servicio

En cambio, la aplicación de 12 factores se define tanto por sus procesos como por sus propiedades sistémicas. Es una metodología para desarrollar una aplicación de software como servicio (SaaS) (es decir, una aplicación web) y, generalmente, implementarla en una plataforma como servicio (PaaS), como Pivotal Cloud Foundry. Estos son los 12 factores, con una breve descripción de cada uno de ellos:⁶

1. **Implementar la aplicación muchas veces desde una base de código.** La base de código se almacena en un repositorio, se gestiona con un sistema de control de versiones como Git mientras se modifica, y después se implementa muchas veces como una instancia en ejecución de la aplicación a partir de la misma base de código. Como consecuencia, una implementación se suele ejecutar en tres entornos: en el entorno local de cada desarrollador, en un entorno de preproducción y en el entorno de producción.
2. **Declarar y aislar dependencias.** La aplicación no utiliza implícitamente paquetes disponibles en todo el sistema, sino que declara las dependencias en un manifiesto de declaración. La declaración explícita de las dependencias facilita a los nuevos desarrolladores configurar su entorno de desarrollo.
3. **Almacenar la configuración, en vez del código, en el entorno.** En cuanto a la información de configuración que varía por implementación, la aplicación almacena la información en las variables de entorno. Las variables de entorno ofrecen un control detallado que se gestiona por separado para cada implementación, a fin de que la aplicación pueda escalarse fácilmente a más implementaciones con el tiempo.
4. **Conectar con los servicios de soporte,** como una base de datos o un sistema de almacenamiento, en lugar de incluirlos en el código. La aplicación trata estos servicios como recursos que se pueden conectar o desconectar en una implementación modificando la configuración.
5. **Tratar la compilación y la ejecución como etapas independientes.** La implementación de la base de código se realiza en tres etapas independientes: compilación, lanzamiento y tiempo de ejecución. La etapa de compilación convierte la base de código en un ejecutable (una compilación) que en la etapa de lanzamiento se combina con la configuración para producir una versión lista para ejecutarse en el entorno de tiempo de ejecución.
6. **Ejecutar la aplicación como procesos sin estado.** Los procesos no comparten nada con otros procesos, y los datos que deben persistir se almacenan en una base de datos que se ejecuta como un servicio de soporte con estado.

⁶ Los doce factores son paráfrasis de las descripciones del [sitio web Twelve-Factor App](#).

GESTIÓN DE APLICACIONES CONTENEDORIZADAS CON KUBERNETES

Kubernetes coordina aplicaciones contenedorizadas distribuidas para:

- Optimizar la utilización de los recursos informáticos.
- Proporcionar políticas de programación.
- Mantener el estado deseado.
- Gestionar fallos y errores mediante la automatización.
- Proporcionar una alta disponibilidad.
- Supervisar trabajos en tiempo real.
- Gestionar la configuración de una aplicación.
- Escalar dinámicamente para adaptarse a los cambios de demanda.

7. **Exponer servicios mediante vinculación de puertos.** Si se toma HTTP como ejemplo, la aplicación exporta HTTP como un servicio vinculándolo a un puerto y escuchando las solicitudes entrantes en el puerto.
8. **Escalar horizontalmente añadiendo procesos simultáneos.** La aplicación controla las cargas de trabajo asignando cada tipo de trabajo a un tipo de proceso. Por ejemplo, un proceso web gestiona solicitudes HTTP, mientras que un proceso subordinado gestiona las tareas en segundo plano.
9. **Desechar para garantizar la durabilidad.** Los procesos son desechables; se pueden iniciar o detener rápidamente para garantizar que la aplicación se modifica o escala fácilmente.
10. **Tratar el desarrollo y la producción como homólogos.** La aplicación está diseñada para una implementación continua, ya que permite a los desarrolladores integrar código nuevo rápidamente e implementar por sí mismos la aplicación en un entorno de producción. Los entornos de producción y desarrollo deben ser tan similares como sea posible.
11. **Procesar registros como flujos de eventos.** La aplicación ni redirige ni almacena el flujo de salida de sus registros, sino que lo escribe como un flujo de datos en la salida estándar, donde el entorno de ejecución debe recogerlo y redirigirlo a una herramienta o un sistema, como Hadoop, para su almacenamiento o análisis.
12. **Ejecutar scripts y tareas de gestión puntuales,** como una migración de base de datos, en un entorno idéntico al de los procesos de larga duración de la aplicación.

Los contenedores y Kubernetes ayudan a satisfacer diversos aspectos de estas exigencias. Por ejemplo, los contenedores desempeñan un papel fundamental en las aplicaciones de 12 factores, ya que permiten declarar y aislar las dependencias. Los contenedores también ayudan a garantizar la durabilidad, y pueden desecharse, además de iniciarse rápidamente y detenerse con facilidad, entre otras características. Kubernetes ofrece muchos de los demás factores.

Descripción concisa de Kubernetes

El desarrollo de Kubernetes tuvo su origen en Google. La empresa usaba su predecesor, llamado Borg, para iniciar, programar, reiniciar y supervisar aplicaciones orientadas al público, como Gmail y Google Docs, y algunos marcos internos, como MapReduce.⁷ Se basaron en el sistema original de Google e incorporaron algunas mejoras derivadas de las lecciones aprendidas con Borg para crear Kubernetes, un sistema de coordinación de código abierto para contenedores que puede funcionar en el centro de datos, en clouds y en un centro de datos híbrido. Kubernetes asigna automáticamente las cargas de trabajo, reinicia las aplicaciones y añade recursos para satisfacer la demanda.

A grandes rasgos, funciona de la siguiente manera. Un clúster de Kubernetes contiene un nodo maestro y varios nodos subordinados. A continuación, cuando se implementa una aplicación en el clúster, los componentes de la aplicación se ejecutan en los nodos subordinados. El nodo maestro gestiona la implementación.

Kubernetes incluye los siguientes componentes:

- La API de Kubernetes
- La interfaz de línea de comandos de Kubernetes, kubectl
- El plano de control de Kubernetes

⁷ «Large-Scale Cluster Management at Google with Borg», estudio de investigación de Google de 2015.

VENTAJAS DE USAR KUBERNETES

- Consolida los servidores y reduce los costes a través del uso eficaz de los recursos.
- Gestiona de forma refinada los fallos de la máquina mediante la recuperación automática y la alta disponibilidad.
- Facilita y agiliza la implementación, el registro y la supervisión de las aplicaciones.
- Automatiza la escalabilidad de los contenedores y las aplicaciones contenedorizadas.
- Desvincula las aplicaciones de las máquinas para una mejor portabilidad y flexibilidad.
- Modifica, actualiza, amplía o vuelve a implementar aplicaciones de forma sencilla sin que afecte a otras cargas de trabajo.

El plano de control consta de los procesos que se ejecutan en el nodo maestro y en cada nodo subordinado de Kubernetes. Por ejemplo, en el nodo maestro Kubernetes ejecuta varios procesos: el servidor de API, el controlador, el planificador y etc. Los nodos subordinados se ejecutan en el proceso «kubect» para comunicarse con el nodo maestro y el proceso de proxy a fin de gestionar la red.

Modelo de objetos de Kubernetes

Una de las claves del sistema Kubernetes es la manera de representar el estado de las aplicaciones contenedorizadas y las cargas de trabajo que se han implementado. Kubernetes representa el estado con «objetos», como servicio, espacio de nombres y volumen. Estos objetos se suelen establecer mediante una especificación de objetos («spec») que se crea para el clúster.

En el modo de objetos de Kubernetes existe el concepto de módulo, que es el componente implementable más básico. Un módulo representa una instancia de una aplicación que se ejecuta como un proceso en un clúster de Kubernetes. Es aquí donde vuelve a entrar en escena el entorno de tiempo de ejecución de Docker, que suele utilizarse en un módulo de Kubernetes.

Kubernetes también incluye controladores que implementan la mayor parte de la lógica de Kubernetes. Los controladores proporcionan características como el conjunto de réplicas y el conjunto con estado.

Mantenimiento del estado deseado

El plano de control de Kubernetes gestiona el estado de todos estos objetos para garantizar que se correspondan con el estado deseado. Puede especificar un estado deseado creando una especificación de objeto para un servicio con un archivo YAML. Veamos un ejemplo:

```

apiVersion: v1
kind: Service
metadata:
  name: nginx-demo-service
  labels:
    app: nginx-demo
spec:
  type: NodePort
  ports:
  - port: 80
    protocol: TCP
    name: http
  selector:
    app: nginx-demo
---
apiVersion: v1
kind: ReplicationController
metadata:
  name: nginx-demo
spec:
  replicas: 3

```

```
template:  
metadata:  
  labels:  
    app: nginx-demo  
  spec:  
    containers:  
      - name: nginx-demo  
        image: myrepo/nginx  
    ports:  
      - containerPort: 80
```

Cuando este archivo se envía al nodo maestro de Kubernetes a través de la interfaz de línea de comandos kubectl, el plano de control de Kubernetes aplica las instrucciones del archivo, iniciando y programando aplicaciones para que el estado del clúster coincida con el estado deseado. Posteriormente, el nodo maestro y el plano de control de Kubernetes se encargan de mantener el estado deseado coordinando los nodos del clúster, que pueden ser servidores reales o máquinas virtuales.

El núcleo de la arquitectura es un servidor de API que gestiona el estado de los objetos del sistema. El servidor de API funciona con subcomponentes de Kubernetes, también llamados clientes, que se crean como microservicios componibles, como el controlador de replicación especificado en el archivo YAML. El controlador de replicación regula el estado deseado de réplicas de módulo cuando se producen fallos.

El valor de Kubernetes para la empresa

Volviendo a la transformación digital, Kubernetes usa esta arquitectura para gestionar aplicaciones contenedorizadas en un clúster distribuido. Los resultados ayudan a cumplir la promesa de la transformación digital:

- Kubernetes simplifica y abarata la ejecución de aplicaciones en clouds públicas, privadas o híbridas.
- Kubernetes acelera el desarrollo y la implementación de aplicaciones.
- Kubernetes aumenta la agilidad, la flexibilidad y la capacidad de reaccionar a los cambios.

Kubernetes para aplicaciones nativas de cloud y aplicaciones de 12 factores

Kubernetes hace que las aplicaciones contenedorizadas funcionen de una manera gestionable según las necesidades. Recordemos la segunda parte de la definición de las aplicaciones nativas de cloud: se coordinan dinámicamente de manera que los contenedores se programan y gestionan de forma activa para optimizar la utilización de recursos. Es justo lo que hace Kubernetes. Coordina los contenedores y sus cargas de trabajo para optimizar la utilización de las máquinas virtuales y los servidores físicos que forman los nodos de un clúster.

Revisemos los 12 factores para ilustrar cómo optimiza Kubernetes la gestión de aplicaciones. En general, Kubernetes puede implementar y ejecutar aplicaciones de 12 factores.

CÓMO OPTIMIZAR LA GESTIÓN DE APLICACIONES CON KUBERNETES Y CONTENEDORES	
Factor	Ventaja
1. Implementar la aplicación muchas veces desde una sola base de código.	Kubernetes puede implementar aplicaciones muchas veces con una sola base de código, proporcionando a un módulo una especificación que incluye una referencia de imagen de contenedor.
2. Declarar y aislar dependencias.	Los contenedores pueden expresar dependencias.
3. Almacenar la configuración, en vez del código, en el entorno.	Puede almacenar aspectos de la configuración de una aplicación en el entorno de Kubernetes. Por ejemplo, la estructura ConfigMaps separa los elementos de configuración de las instrucciones de una imagen.
4. Conectar con los servicios de soporte, como una base de datos, en lugar de incluirlos en el código.	Kubernetes permite implementar servicios de soporte (como una base de datos) en contenedores independientes, y después gestiona todos los componentes contenedorizados de forma conjunta para garantizar la disponibilidad y el rendimiento.
5. Tratar la compilación y la ejecución como etapas independientes.	Por ejemplo, se puede generar la aplicación con Jenkins (un servidor de automatización de flujo independiente de Kubernetes) y después ejecutar las imágenes de Docker con Kubernetes.
6. Ejecutar la aplicación como procesos sin estado.	Kubernetes facilita la ejecución de aplicaciones sin estado. Kubernetes permite mantener los estados por separado en un almacén de datos etcd (por ejemplo) mientras se ejecuta la aplicación. Kubernetes también permite conectar un almacenamiento persistente. Por ejemplo, el archivo de especificación que define un módulo puede requerir un volumen persistente; si el módulo deja de funcionar, se conecta un módulo de reemplazo al mismo volumen persistente.
7. Exponer servicios mediante vinculación de puertos.	Kubernetes incluye opciones de configuración para exponer servicios en puertos. En el archivo YAML del ejemplo de nginx que se mencionó antes, el servidor web nginx estaba vinculado al puerto 80 y se exponía como un servicio.
8. Escalar horizontalmente añadiendo procesos simultáneos.	Kubernetes escala una aplicación añadiendo más módulos. Kubernetes puede usar el controlador de replicación para, por ejemplo, añadir varios módulos simultáneamente.
9. Desechar para garantizar la durabilidad.	Los contenedores que se ejecutan en Kubernetes pueden considerarse mutables: se detienen y sustituyen según las necesidades o de forma programada.
10. Tratar el desarrollo y la producción como homólogos.	El entorno de Kubernetes permite llevar a cabo las mismas pruebas rigurosas con el código de desarrollo y de producción. Por ejemplo, puede usar una implementación de Kubernetes con dos módulos, un módulo que contiene el entorno de producción y otro módulo que contiene el entorno de preproducción, lo que equipara la preproducción a la producción de forma efectiva. Además, el entorno especificado en un contenedor es uniforme para los entornos de desarrollo y de producción.
11. Procesar registros como flujos de eventos.	Kubernetes permite acceder a la salida estándar de un contenedor para que pueda procesar la salida como un flujo de datos utilizando la herramienta que prefiera, como por ejemplo VMware vRealize® Log Insight™.
12. Ejecutar tareas de gestión como procesos puntuales.	Se puede programar un módulo formado por el contenedor de aplicaciones con un punto de entrada distinto para ejecutar un proceso diferente, como un script para migrar una base de datos.

VENTAJAS PARA LOS DESARROLLADORES

El valor empresarial de los contenedores y Kubernetes no se limita a la empresa en su conjunto o a la oficina del director de informática. A los desarrolladores les gustan los contenedores, ya que les facilitan las tareas y hacen que el desarrollo sea más interesante y el trabajo más productivo.

- **Portabilidad:** los contenedores permiten a los desarrolladores elegir cómo y cuándo se implementa una aplicación.
- **Velocidad:** los contenedores agilizan los flujos de trabajo (por ejemplo, la realización de pruebas) y aceleran las iteraciones.
- **Flujo CI/CD:** Kubernetes y los contenedores permiten la integración e implementación continuas.
- **Flexibilidad:** los desarrolladores pueden programar en sus portátiles cuando quieran y donde quieran, con sus herramientas preferidas.
- **El decimotercer factor:** los contenedores y Kubernetes son tecnologías de moda. Los desarrolladores se sienten muy motivados para usarlas.

Un ejemplo de caso de uso

Este breve caso práctico muestra un caso de uso de alto nivel para gestionar contenedores con Kubernetes.

Una empresa de taxis de una importante zona metropolitana está perdiendo clientes que optan por los servicios de coche compartido. Esta situación está poniendo en peligro su cuota de mercado local, que antes era muy sólida. Debe transformarse en una empresa digital para poder competir con las empresas de coches compartidos. Para ello, la empresa quiere desarrollar su propia aplicación móvil, ejecutar de forma rentable la aplicación en su modesto centro de datos, e intentar ofrecer servicios innovadores.

La empresa de taxis tiene a su favor varias ventajas: una marca local conocida y de larga tradición cuyos conductores tienen fama de ser puntuales, amables y responsables.

Mientras los nuevos desarrolladores contratados trabajan en la aplicación móvil, la empresa de taxis está modernizando su centro de datos con hardware básico y tecnología de virtualización. Para maximizar la utilización de recursos de su pequeño centro de datos y minimizar costes, la empresa prevé ejecutar su nueva aplicación en contenedores Docker alojados en máquinas virtuales. Kubernetes coordinará la aplicación contenedorizada.

Tras implementar la aplicación y publicarla dentro y encima de sus coches, se convierte en un éxito inmediato. Para adaptarse a las fluctuaciones de uso de la aplicación, la empresa utiliza Kubernetes para escalar de forma dinámica el número de contenedores que ejecutan la aplicación. Por ejemplo, cuando los parámetros de la aplicación alcanzan un umbral predefinido que indica un uso elevado, algo que suele ocurrir en las horas punta, el equipo de DevOps de la empresa utiliza la característica de escalabilidad horizontal automática de los módulos de Kubernetes para maximizar el número de contenedores de forma que el sistema pueda satisfacer la demanda. En cambio, a las 4 de la madrugada el número de contenedores se reduce para adaptarse de forma flexible a la baja demanda de ese momento, ahorrando recursos.

La aplicación móvil correlaciona las solicitudes de viajes con la ubicación. Combinando la minería de datos con su profundo conocimiento histórico de los patrones de la ciudad, la empresa de taxis puede situar taxis en las ubicaciones perfectas para atraer clientes, quitando a la competencia algunos servicios. Además, como la empresa procesa los registros de la aplicación como flujos de eventos, puede hacer esto de manera dinámica, día y noche, moviendo coches a los puntos estratégicos.

Gracias a la implementación de la aplicación mediante contenedores, los desarrolladores pueden implementar los nuevos cambios a diario. Los datos que la aplicación recopila ayudan a la empresa a identificar nuevas características y a innovar rápidamente para centrarse en sus puntos fuertes, como identificar a los clientes que repiten y ofrecerles un programa de fidelización para conservarlos.

Las ventajas empresariales de la agilidad técnica de la empresa, la aplicación contenedorizada y la coordinación de Kubernetes se suman para ofrecer una ventaja competitiva:

- Las políticas de planificación de Kubernetes ofrecen a la empresa la flexibilidad que necesita para cubrir dinámicamente la demanda de una manera rentable con su centro de datos modesto, pero modernizado.
- Kubernetes gestiona automáticamente los fallos y errores, reduciendo las solicitudes de solución de problemas que recibe el reducido equipo de DevOps.
- La eficaz modificación de la aplicación y de sus características permite a la empresa vencer a sus principales rivales, menos locales, al ofrecer más agilidad y tener más capacidad para aplicar sus conocimientos sobre los patrones locales.

- Los contenedores y Kubernetes simplifican y abaratan la aplicación.
- La facilidad con la que el equipo de DevOps puede portar contenedores del entorno de pruebas al entorno de producción acelera el desarrollo y la implementación de nuevas características.

Soluciones de tecnologías de contenedores de VMware

En un reciente informe titulado «Closing the Digital Transformation Confidence Gap in 2017», The Hackett Group entrevistó a ejecutivos de más de 180 grandes empresas. El reporte determinó que hay una brecha «entre las altas expectativas para el impacto empresarial de la transformación digital y la percepción deficiente de la capacidad de la empresa para llevar a cabo dicha transformación». The Hackett Group indicó que sus hallazgos demuestran la necesidad de que los departamentos de TI inviertan en las herramientas necesarias y adopten técnicas de desarrollo rápido de aplicaciones, como los procesos ágiles.⁸

Las soluciones nativas de cloud de VMware le ayudan a poner los contenedores en producción de forma rápida y rentable, mejorando su capacidad para llevar a cabo la transformación digital.

La ejecución de contenedores en máquinas virtuales también añade un nivel de seguridad beneficioso para las aplicaciones contenedorizadas, especialmente en el contexto de la tercera faceta de las aplicaciones nativas de cloud: los microservicios. Según un documento técnico de Docker sobre seguridad, «La implementación de contenedores de Docker junto con máquinas virtuales permite aislar entre sí un grupo completo de servicios y después agruparlos en un host de máquinas virtuales».⁹

La implementación de contenedores en las máquinas virtuales permite proteger la aplicación con dos capas de aislamiento. Este enfoque es adecuado para entornos de tipo cloud multicliente con múltiples cargas de trabajo. En el documento técnico sobre seguridad de Docker se menciona que «los contenedores de Docker y las tecnologías de virtualización son una buena combinación que protege a la propia máquina virtual y proporciona una defensa mejorada para el host».

vSphere Integrated Containers

VMware vSphere Integrated Containers, una solución de contenedor completa basada en VMware vSphere, ejecuta en paralelo cargas de trabajo modernas y tradicionales en el centro de datos definido por software de VMware con red, almacenamiento, seguridad, rendimiento y visibilidad de nivel empresarial.

vSphere Integrated Containers ofrece compatibilidad con los contenedores de Docker, lo que permite usar inmediatamente la tecnología de contenedores para aumentar la productividad de los desarrolladores y la agilidad de la empresa. La solución ayuda a transformar su organización en una empresa digital y a modernizar el centro de datos mediante la implementación de aplicaciones contenedorizadas.

⁸ «Despite High Expectations for Digital Transformation Led by Cloud, Analytics, Robotic Process Automation, Cognitive & Mobile, IT & Other Business Services Areas See Low Capability to Execute», The Hackett Group, 16 de marzo de 2017. Si desea descargar una versión de este estudio, regístrese en <http://www.thehackettgroup.com/research/2017/social-media/key17it/>.

⁹ «Introduction to Container Security», documento técnico de Docker, Docker.com.

Descripción de la arquitectura

El siguiente diagrama ilustra la arquitectura de alto nivel de vSphere Integrated Containers.

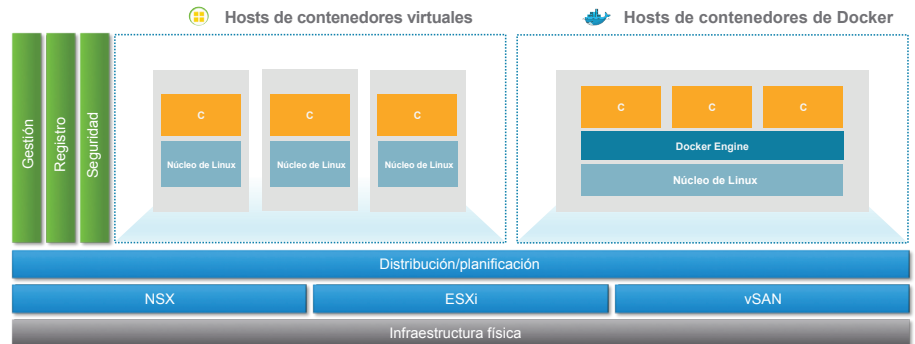


Figura 1: Arquitectura de alto nivel de vSphere Integrated Containers

Esta arquitectura le ofrece dos modelos de implementación de contenedores:

- **Hosts de contenedores virtuales:** vSphere Integrated Containers aprovecha las estructuras nativas de vSphere para distribuir contenedores. Mediante la implementación de cada imagen de contenedor como una máquina virtual, vSphere Integrated Containers extiende a las cargas de trabajo contenedorizadas la disponibilidad y las características de rendimiento de vSphere, incluidos VMware HA, vMotion y Distributed Resource Scheduler. Además, los desarrolladores pueden utilizar una API de Docker.
- **Hosts de contenedores de Docker:** los desarrolladores pueden autoimplementar hosts de contenedores de Docker según las necesidades y ejecutarlos en vSphere. El entorno sin tiques que vSphere Integrated Containers proporciona permite a los desarrolladores utilizar las herramientas de Docker y ofrece a los equipos de TI control sobre la infraestructura.

Componentes

Cada uno de los componentes de vSphere Integrated Containers es un proyecto de código abierto:

- **vSphere Integrated Containers Engine:** este motor, que ofrece un entorno de tiempo de ejecución de contenedores para vSphere, permite a los ingenieros de software desarrollar en contenedores e implementar aplicaciones contenedorizadas junto con cargas de trabajo tradicionales basadas en máquinas virtuales en clústeres de vSphere.
- **Harbor:** registro de contenedores privado de nivel empresarial que almacena y distribuye imágenes de contenedores. Harbor amplía la distribución de código abierto de Docker con funcionalidad empresarial como gestión de identidades, control de acceso basado en funciones y auditoría.
- **Admiral:** portal de gestión de contenedores que suministra una interfaz de usuario para equipos de DevOps y otros usuarios con la que pueden distribuir y gestionar contenedores. Por ejemplo, Admiral puede mostrar métricas de instancias de contenedor. Los administradores de cloud pueden gestionar hosts de contenedores y aplicar controles sobre su uso, incluidas cuotas de capacidad.

LAS VENTAJAS DE LOS MICROSERVICIOS

Los microservicios, combinados con los contenedores, se están convirtiendo en el patrón de arquitectura preferido para desarrollar una nueva aplicación. La arquitectura descompone las funciones de una aplicación en un conjunto de procesos pequeños, discretos, descentralizados y orientados a objetivos, cada uno de los cuales se puede desarrollar, probar, implementar, sustituir y escalar por separado.

- Aumentan la modularidad.
- Facilitan las actividades de desarrollo y pruebas de las aplicaciones.
- Paralelizan el desarrollo: un equipo puede desarrollar e implementar un servicio por separado de los otros equipos que trabajan en otros servicios.
- Permiten la refactorización continua de código para maximizar las ventajas a largo plazo de los microservicios.
- Impulsan un modelo integración e implementación continuas.
- Mejoran la escalabilidad.
- Simplifican las actualizaciones de componentes.

Prestaciones

vSphere Integrated Containers incluye un portal de gestión unificada que se integra con la gestión de identidades para distribuir contenedores de forma segura. Los desarrolladores y los equipos de DevOps pueden cumplir sus requisitos creando hosts de contenedores de Docker según las necesidades.

El resultado permite a los equipos de desarrollo de aplicaciones generar, probar e implementar aplicaciones contenedorizadas. La solución admite prácticas de desarrollo ágiles y metodologías de DevOps como la integración e implementación continuas (CI/DI).

Wavefront by VMware

Wavefront® by VMware supervisa eficazmente los contenedores según las necesidades. La plataforma Wavefront incluye paneles de gestión que ofrecen a los equipos de DevOps visibilidad en tiempo real de las operaciones y el rendimiento de aplicaciones contenedorizadas y clústeres de Kubernetes.

El servicio Wavefront puede medir, correlacionar y analizar datos de contenedores y clústeres de Kubernetes. El panel de gestión muestra datos sobre el rendimiento de los microservicios y la utilización de recursos para ayudarle a identificar problemas y optimizar las aplicaciones. Por ejemplo, los datos pueden ayudar a tomar decisiones sobre cómo y cuándo escalar un entorno de contenedor. Para obtener más información, vea [VMware y Wavefront](#).

Pivotal Container Service

VMware® Pivotal Container Service ofrece una versión de Kubernetes de nivel de producción con la que las empresas pueden implementar, ejecutar y poner en marcha aplicaciones modernas y tradicionales en clouds públicas y privadas de manera fiable. Pivotal Container Service se basa en el proyecto de código abierto Kubo y ofrece alta disponibilidad, seguridad avanzada y eficiencia operativa para que las empresas puedan reducir el tiempo de comercialización, aumentar la productividad de los desarrolladores y reducir los gastos operativos.

A fin de proporcionar una vía rápida para la producción de microservicios y cargas de trabajo contenedorizadas, Pivotal Container Service establece una infraestructura unificada de virtualización y contenedores en VMware vSphere o en un centro de datos definido por software de VMware.

Componentes

Pivotal Container Service cuenta con los componentes siguientes:

- **Kubernetes de nivel de producción.**
- **BOSH.** Es un sistema de código abierto que unifica la ingeniería, la implementación y la gestión del ciclo de vida de versiones para software de cloud a pequeña y gran escala. Es adecuado para grandes sistemas distribuidos y se encarga de la supervisión, la recuperación ante fallos y las actualizaciones de software con un tiempo de inactividad mínimo o nulo. BOSH es compatible con múltiples proveedores de servicios como infraestructura, como VMware vSphere, Google Cloud Platform, Amazon Elastic Compute Cloud (EC2) y OpenStack.

MÁS INFORMACIÓN SOBRE...

Para descubrir cómo le pueden ayudar las soluciones de VMware a crear, ejecutar y gestionar aplicaciones nativas de cloud, visite: cloud.vmware.com/es/cloud-native-apps

- **VMware ESXi™.** ESXi es un hipervisor nativo especialmente diseñado, líder del mercado. Se instala directamente en el servidor físico, lo que permite dividirlo en máquinas virtuales.
- **VMware NSX®.** Esta tecnología de virtualización de red para arquitecturas de aplicaciones modernas proporciona las funcionalidades de red principales a los clústeres de Kubernetes.

Conclusión

La creación e implementación de aplicaciones contenedorizadas en infraestructura de VMware impulsa el valor para la empresa a través de la transformación digital. Las soluciones de VMware aumentan la productividad de los desarrolladores, la agilidad empresarial, la flexibilidad de TI y la escalabilidad de aplicaciones. El resultado ayuda a adaptarse a los cambios del mercado y a reducir el tiempo de comercialización de las aplicaciones.



VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

C/ Rafael Botí, 26 - 2.ª planta, 28023 Madrid, España. Tel. +34 914125000 Fax +34 914125001 www.vmware.com/es

Copyright © 2017-2018 VMware, Inc. Todos los derechos reservados. Este producto está protegido por las leyes de derechos de autor y de propiedad intelectual de Estados Unidos e internacionales. Los productos de VMware están cubiertos por una o varias de las patentes enumeradas en <http://www.vmware.com/go/patents>. VMware es una marca comercial o marca registrada de VMware Inc. en Estados Unidos o en otras jurisdicciones. Las demás marcas y nombres mencionados en este documento pueden ser marcas comerciales de sus respectivas empresas.

N.º artículo: VMW_17Q3_WP_Driving-Digital-Transformation-with Kubernetes_FINAL2_081617

08/17