

WHITEPAPER
Nociones sencillas
sobre almacenamiento
persistente para
contenedores

Aceleración del desarrollo de
aplicaciones y aumento de la
eficiencia de DevOps



El reto: aumentar la madurez de DevOps	3
El gran obstáculo de los contenedores: gestión del almacenamiento persistente para aplicaciones con estado	3
Breve información general sobre contenedores, persistencia de datos y aplicaciones con y sin estado	4
Un buen comienzo: aprovisionamiento básico de almacenamiento persistente para contenedores	4
El veredicto: hay avances, pero la espera sigue siendo excesiva	4
Tipos de almacenamiento, pools de almacenamiento y Trident: Almacenamiento persistente bajo demanda	5
¿Puede ser tan sencillo un catálogo de tipos de almacenamiento?	5
El futuro de DevOps	6
Transformación del modo de trabajar	6
Le ayudamos a usted y a su negocio	7
Recursos adicionales	7

El reto: aumentar la madurez de DevOps

Innovación continua. Un despliegue de productos mejor y más rápido. Operaciones más sencillas. Cada vez más clientes satisfechos.

Estos son los objetivos universales de muchas organizaciones. También son momentos clave en el camino para conseguir mejores resultados comerciales: un mayor crecimiento de los ingresos brutos y mayores beneficios base.

El lado negativo de estos avances suelen ser las dificultades y frustraciones de una organización en plena transformación. Los desarrolladores de aplicaciones, siempre ansiosos por acelerar los desarrollos que resolverán necesidades de los clientes, pueden sentirse frustrados con los sistemas de tickets y las largas esperas para obtener recursos de almacenamiento y computación. Los equipos de infraestructura y operaciones (I&O), por su parte, hacen lo que pueden por mantenerse al día con los tickets de servicio entrantes y controlar el modo en que se utiliza la infraestructura.

Esta mezcla de objetivos y dificultades crecientes lleva a las organizaciones a buscar metodologías ágiles y ligeras que permitan alcanzar mayor velocidad y eficiencia en el desarrollo de productos. Y, aunque estas metodologías han proporcionado importantes beneficios, se podría lograr mucho más si la infraestructura y los procesos tecnológicos estuvieran optimizados para lograr la eficiencia y la velocidad que las organizaciones desean.

Aquí es donde interviene DevOps. DevOps es un enfoque diseñado para lograr más rápidamente la transformación tanto en el desarrollo de aplicaciones como en las operaciones de infraestructura tecnológica.

Las organizaciones que logran alcanzar la madurez de DevOps suelen destacar por dominar seis capacidades clave:

- **Gestión de código, artefactos y binarios** Repositorios para la retención y gestión de componentes de software
- **Gestión de la configuración** Configuración y gestión de la infraestructura y los sistemas de software de formas conocidas
- **Cloud/PaaS** Uso de la infraestructura de cloud público, privado e híbrido para dar soporte al desarrollo de software
- **Contenedores** Entornos de ejecución de aplicaciones ligeros pero muy escalables
- **Análisis** Supervisión y gestión automatizadas de la infraestructura
- **Integración continua/implementación continua (CI/CD)** Procesos automatizados integrales que permiten a los desarrolladores escribir e implementar código de manera automática

Una de estas seis áreas en concreto (los contenedores) ha ganado una importancia cada vez mayor para el desarrollo de aplicaciones y para los equipos de operaciones de infraestructura que buscan más velocidad y eficiencia. Este whitepaper trata de esta transición hacia la madurez de DevOps por medio de contenedores, al tiempo que señala uno de los problemas clave que se deben tener en cuenta: la gestión del almacenamiento persistente para aplicaciones con estado.

El gran obstáculo de los contenedores: gestión del almacenamiento persistente para aplicaciones con estado

A medida que los equipos de DevOps han comenzado a tener más en cuenta las implementaciones de producción reales de aplicaciones con contenedores, han surgido algunas complicaciones. Una de ellas, y no la menor, es la gestión del almacenamiento de datos persistente para contenedores.

¿Necesita comprender mejor conceptos tales como la persistencia de datos con contenedores o la diferencia entre las aplicaciones con y sin estado? Consulte el resumen de la página 4.

Al principio de la transición hacia los contenedores de aplicaciones se hizo evidente que la gestión del almacenamiento de datos suponía un desafío. En una encuesta realizada por la Cloud Native Computing Foundation (CNCF), casi la mitad de los participantes (42 %) dijo que la gestión del almacenamiento y los recursos era una de las principales dificultades para la adopción de los contenedores (figura 1).¹ Muchos de ellos citaron problemas continuados con la persistencia del almacenamiento. Otros deseaban que el acceso al almacenamiento en red fuera más sencillo.

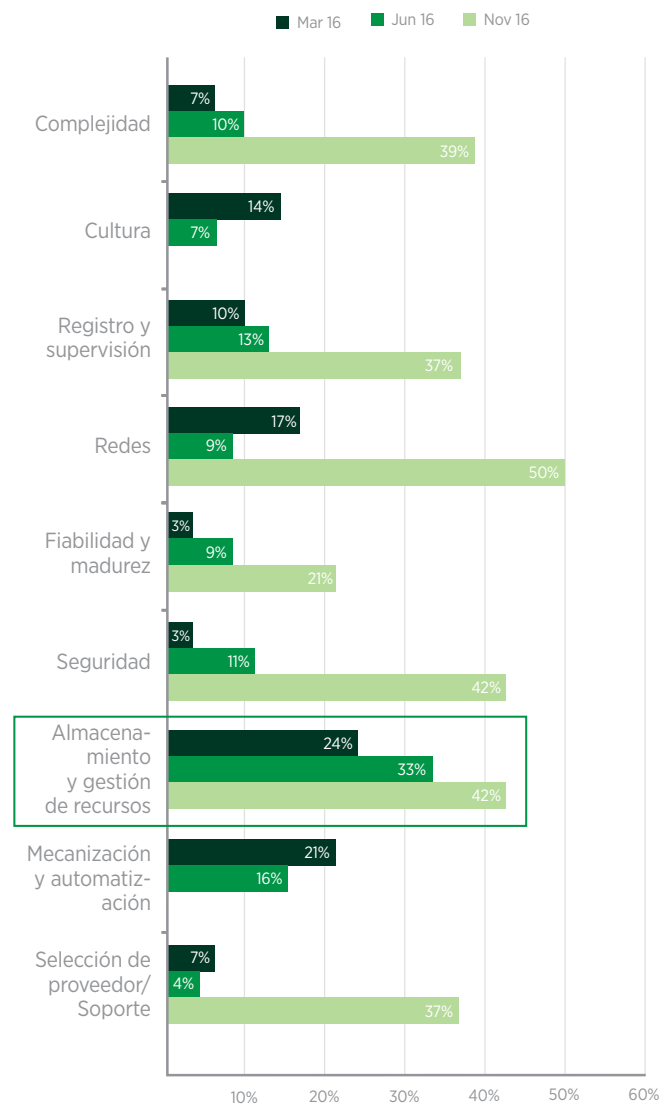


Figura 1) Desafíos para la adopción de contenedores (se permitían varias respuestas).

Fuente: Cloud Native Computing Foundation.

Como resultado, las plataformas de contenedores comenzaron a ocuparse del desafío del almacenamiento persistente. Los primeros esfuerzos para el aprovisionamiento de almacenamiento persistente dieron frutos, aunque seguían siendo poco flexibles y debían gestionarse de forma manual. Además, acelerar la canalización de DevOps no parecía posible, especialmente cuando las aplicaciones debían codificarse, probarse e implementarse en cientos o miles de contenedores.

Desde la perspectiva de la escalabilidad, la necesidad de aprovisionar manualmente el almacenamiento para miles de contenedores empezó a parecer inmanejable: resultaba demasiado pesado, propenso a errores y difícil de mantener. Tenía que haber un modo más fácil.

Antes de los contenedores, muchas aplicaciones empresariales habían cubierto con éxito sus necesidades de almacenamiento de datos persistentes mediante una conexión centralizada al almacenamiento empresarial compartido. En un mundo de contenedores, ¿no podrían operar estas aplicaciones con estado igual de bien si tuvieran acceso sencillo a funciones similares de almacenamiento compartido? Pero quedaba esta pregunta: ¿sería sencillo llegar ahí?

Un buen comienzo: aprovisionamiento básico de almacenamiento persistente para contenedores

Los entornos de contenedores como Docker y Kubernetes respondían originalmente al desafío de la persistencia del almacenamiento con un mecanismo útil y semiautomatizado. Este mecanismo permitía a los usuarios crear y “reclamar” un volumen de almacenamiento persistente particular para su uso en uno o más procesos de contenedores. Para ello, era necesario que el administrador de almacenamiento creara primero volúmenes persistentes a partir de varios recursos de almacenamiento en red subyacentes.

Breve información general sobre contenedores, persistencia de datos y aplicaciones sin estado

Por su misma naturaleza, los contenedores no tienen estado. Su contenido es efímero. Esto significa que las aplicaciones o procesos relacionados con un contenedor se pueden iniciar, detener y reiniciar rápidamente durante una sesión. Por defecto, esto significa que los datos creados mientras un contenedor estaba activo se destruían en cuanto el contenedor se terminaba o destruía.

Por desgracia, a medida que los equipos de DevOps trataban de desarrollar e implementar nuevas aplicaciones en plataformas de contenedores, surgió la necesidad de que los datos se conservaran o persistieran una vez acabada la vida de un contenedor.

Muchos descubrieron pronto que prácticamente en todas las aplicaciones había al menos un proceso o microservicio que requería un volumen persistente (VP) para aquellos datos con estado que debían persistir una vez terminada la vida de un contenedor dado.

Aquí tiene algunos ejemplos de aplicaciones con estado que requieren la persistencia de datos:

Kubernetes: aprovisionamiento básico en acción

Para Kubernetes, el proceso adoptaba mecanismos basados en código asociados con un volumen persistente (VP) y una reclamación de volumen persistente (RVP) correspondiente.

Primero, un administrador debía crear un volumen persistente (por ejemplo, un volumen con 8 GB de capacidad de almacenamiento all-flash). A continuación, el usuario de la aplicación (o el desarrollador) podía solicitar o reclamar ese volumen persistente específico mediante unas pocas líneas de código.

Esta reclamación y enlace de VP y RVP resultaba intrigante para los entornos de contenedores, ya que permitía a las aplicaciones con contenedores comenzar a consumir almacenamiento en red persistente por medio de código.

Aunque era un buen primer paso hacia la posibilidad de consumir almacenamiento en red, con este enfoque básico del aprovisionamiento aún quedaban varias dificultades. La principal era la necesidad de automatización.

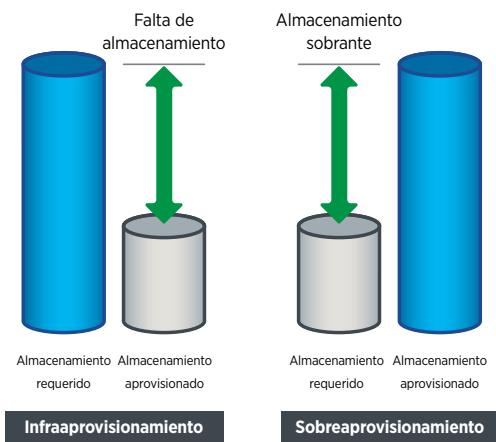
El veredicto: hay avances, pero la espera sigue siendo excesiva

El uso de pares de VP y RVP era un excelente primer paso para resolver los problemas del almacenamiento persistente mediante contenedores. Sin embargo, buena parte del proceso de aprovisionamiento de almacenamiento seguía siendo manual, estático y repetitivo tanto para el lado Dev como para el lado Ops de la ecuación. Se producían diversas ineficiencias en el desarrollo de aplicaciones y los equipos de DevOps trataban de reducir los rechazos y automatizar los esfuerzos hacia el CI/CD:

- **Entornos de base de datos.** Un contenedor de base de datos necesita almacenamiento persistente para su almacén de datos. Sin embargo, dada la naturaleza efímera de los contenedores, eso no es posible sin ayuda. El almacenamiento local tampoco era una buena opción. Si el contenedor se trasladaba o era destruido, se perdía el acceso a esos datos.
- **Datos de entorno o de sesión.** Las aplicaciones con estado a menudo recogen o guardan también atributos de entorno de la aplicación o datos de sesión del cliente (estado). Actúan como datos históricos que mejoran la experiencia del cliente. En este caso, la próxima vez que el cliente interactúa con la aplicación, esta puede presentar datos relevantes o manipular mejor los datos creados en una sesión previa.

Igual de importante que la necesidad de preservar los datos de estado era la de compartir datos. Las organizaciones que creaban aplicaciones con contenedores o que migraban a ellas descubrieron enseguida que todos (desde el equipo de desarrollo hasta los de pruebas y operaciones) necesitaban acceder a los mismos conjuntos de datos desde los mismos recursos centralizados de almacenamiento en red. Además, los volúmenes persistentes guardados en almacenamiento en red también aseguraban que las aplicaciones con contenedores estuvieran más protegidas, pues las funciones de almacenamiento de nivel empresarial mejoraban la disponibilidad de las aplicaciones, la fiabilidad, la seguridad y la protección de datos.

- **Solicitudes manuales de almacenamiento.** A veces, los desarrolladores de aplicaciones o el personal de QA que trabajaba en las sesiones de programación o prueba debía parar lo que estuviera haciendo para solicitar al administrador de almacenamiento que creara un VP. Al escalar aplicaciones con contenedores más grandes, podía haber cientos o miles de solicitudes de almacenamiento en curso.
- **Tiempos de espera frustrantes para el almacenamiento.** Por tanto, los equipos de desarrollo o QA podían quedar a la espera de que se aprobaran sus solicitudes o de que el administrador programara una hora para crear manualmente uno o más VP.
- **Uso ineficiente del almacenamiento.** Al utilizar almacenamiento preaprovisionado para VP en beneficio de los equipos de desarrollo y QA, era muy fácil que el administrador de almacenamiento reservara demasiada capacidad de almacenamiento (sobreaprovisionamiento) o demasiado poca (infraaprovisionamiento) (figura 2). También era posible que reservaran demasiadas IOPS por volumen o demasiado pocas:
 - **El riesgo del infraaprovisionamiento.** Cuando se preaprovisiona demasiado poco almacenamiento para VP, se puede producir un cuello de botella que limita a los desarrolladores que precisan más capacidad de la que hay disponible.
 - **El riesgo del sobreaprovisionamiento.** Los recursos de almacenamiento sobreaprovisionados, que quedan bloqueados en uno o más VP, podrían malgastarse al vincularlos más adelante a una RVP con un requisito de capacidad o rendimiento mucho menor. (¿Recuerda esos viejos silos de almacenamiento sobreaprovisionados e infrautilizados?)
- **Uso ineficiente de los recursos del administrador.** En muchos trabajos de desarrollo de aplicaciones, los participantes en la administración del almacenamiento o los clústeres podían encontrarse el nuevo cuello de botella en una o más canalizaciones de DevOps. Si a última hora llegaban varias solicitudes nuevas de volúmenes, era posible que el administrador debiera seguir realizando varios pasos manuales para la configuración y creación de cada VP estático. Esta práctica tampoco ayudaba a los administradores encargados de automatizar la infraestructura y supervisar el consumo general.



(Figura 2) El riesgo del infraaprovisionamiento y el sobreaprovisionamiento.

Si cree que esto se parece mucho al aprovisionamiento de almacenamiento de antes, sepa que no es el único que piensa así. Al concluir su charla [DevOps Through Desired State](#) en el Tech Field Day de 2017, Andrew Sullivan, ingeniero de marketing técnico de NetApp, se compadeció de los equipos de DevOps que trataban de liberarse de este tipo de aprovisionamiento heredado. **“El almacenamiento no es un recurso que tenga que tolerar porque sí, no tendría que ir con la cabeza gacha a la mesa del equipo de almacenamiento para suplicarles más capacidad”,** dijo. **“Yo no quiero aprovisionar almacenamiento. No quiero consumir almacenamiento en 2017 como lo hacía en 1989”.**²

Por suerte, NetApp (y otros que apoyaron los entornos con contenedores) sabía que debía haber un sistema mejor para que los equipos de DevOps pudieran consumir almacenamiento bajo demanda y aprovisionarlo de forma dinámica, dónde y cuando fuera necesario.

Tipos de almacenamiento, pools de almacenamiento y Trident: almacenamiento persistente bajo demanda

Lo que Andrew Sullivan quería decir en el último apartado era que hacía falta un modo mejor de aprovisionar los recursos de almacenamiento. Un método que hiciera realidad la promesa inicial de los contenedores: creación e implementación dinámica y bajo demanda, con poca o ninguna intervención manual.

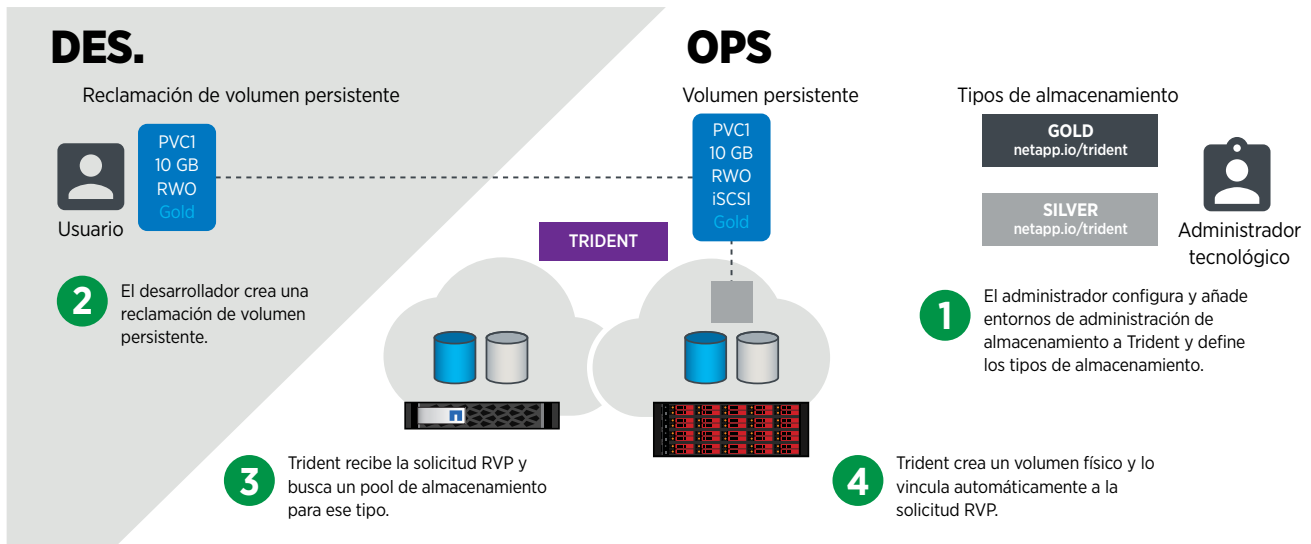
En el caso de los Kubernetes, se introdujo el aprovisionamiento de almacenamiento dinámico mediante un nuevo concepto: el tipo de almacenamiento. También llegaron grandes funciones de automatización de aprovisionadores de almacenamiento, como Trident for Kubernetes:

- **Almacenamiento persistente bajo demanda.** El proyecto Trident de código abierto, desarrollado por NetApp, es un aprovisionador de almacenamiento dinámico que ofrece a las aplicaciones con contenedores acceso bajo demanda a los volúmenes persistentes de almacenamiento NetApp® que necesitan, cuando lo necesiten. Se acabaron las esperas.
- **Acceso al potente almacenamiento subyacente.** Los entornos de contenedores que utilizan Trident son capaces de desbloquear potentes funciones de almacenamiento desde plataformas de gestión de datos subyacentes de NetApp (como NetApp HCI, ONTAP®, NetApp SolidFire® Element® OS o SANtricity®).
- **No es solo para Kubernetes.** También hay disponibles funcionalidades similares para volúmenes persistentes a través de Trident for Docker y OpenShift.

¿Puede ser tan sencillo un catálogo de tipos de almacenamiento?

Con Trident, los desarrolladores de entornos Kubernetes pueden ahora aprovisionar de forma dinámica volúmenes persistentes con solo solicitar un tipo de almacenamiento desde un pool virtual de almacenamiento subyacente.

Vamos a ver cómo funciona. Con los tipos de almacenamiento, el aprovisionamiento de volúmenes persistentes se produce automáticamente bajo demanda, mediante código. El usuario solo tiene que hacer una reclamación de volumen persistente y nombrar un tipo de almacenamiento específico de Trident, como Gold, Silver o Bronze.



(Figura 3) Aprovisionamiento dinámico de volúmenes persistentes mediante Trident.

Nota: La configuración de los atributos y convenciones de nomenclatura de los tipos de almacenamiento subyacentes son una función back-end que suele realizar un administrador cuando crea un catálogo inicial de tipos de almacenamiento persistente. Los tipos de almacenamiento de una organización podrían tener nombres como Desarrollo, Preparación y Producción. Los de otra podrían llamarse Rápido o Lento. Para obtener más información detallada sobre estas funciones de configuración, consulte la [documentación](#) de Trident.³

A continuación, se crea un volumen persistente a partir de ese tipo de almacenamiento en el pool de almacenamiento subyacente de NetApp. Dicho volumen queda vinculado a la reclamación de volumen persistente (RVP) del usuario. Los usuarios ya no tienen por qué saber acerca del almacenamiento subyacente. Trident se ocupa de todos esos detalles. Así de fácil.

Vea Trident en acción

El mejor modo de comprender el valor de Trident es ver su aprovisionamiento dinámico en funcionamiento. Dedique unos momentos a ver una de las siguientes demostraciones en línea:

- **Demostración corta** (3:20 minutos): [Using Trident for Dynamic Storage Provisioning with OpenShift \(Uso de Trident para el aprovisionamiento de almacenamiento dinámico con OpenShift\)](#)⁴
- **Demostración larga** (23:53 minutos): [Managing Persistent Data in Kubernetes \(Gestión de datos persistentes en Kubernetes\)](#)⁵

¿Qué significa este tipo de funcionalidad de aprovisionamiento para los distintos miembros del equipo de tecnología y DevOps?

- **Para los equipos de desarrollo o QA.** Se acabó el esperar la aprobación de tickets de servicio y solicitudes de almacenamiento. Se acabaron los rechazos. Consumo de almacenamiento con un acuerdo de nivel de servicio prometido que funciona cuando y donde lo necesite y que se aprovisiona de forma dinámica mediante interfaces de programación conocidas. Ahora, los desarrolladores cuentan con un sistema de aprovisionamiento automático, flexible y dinámico que los libera sin quitarles el control de las operaciones.

- **Para administradores tecnológicos o de almacenamiento.** Se acabaron las carreras con las solicitudes de aprovisionamiento de almacenamiento de última hora y los interminables tickets de servicio. Menos administración de almacenamiento. Más escalado automático de la infraestructura. Control más previsible sobre el consumo de almacenamiento y supervisión más sencilla de los recursos.
- **Para los ejecutivos de tecnología.** Entrega de productos más rápida, mejores procesos y un significativo ahorro de recursos.

El futuro de DevOps

DARZ, un proveedor de servicios de tecnología, ofrece la agilidad de DevOps con su oferta Docker y contenedor como servicio. Se basa en el almacenamiento all-flash de NetApp y en Trident for Docker. Los clientes pueden aumentar o reducir rápidamente la velocidad de giro de los contenedores de aplicaciones sin necesidad de un sistema operativo completo, lo que reduce hasta en cuatro veces los requisitos de computación.

Gracias a su entorno ligero y flexible de contenedores, los clientes pueden acortar los ciclos de prueba, acelerar el desarrollo e implementar nuevos productos más rápido. Al simplificar la interacción con el almacenamiento de contenedores mediante el conjunto de comandos del volumen Docker, Trident hace que la gestión de datos persistentes en un entorno Docker resulte sencilla.⁶

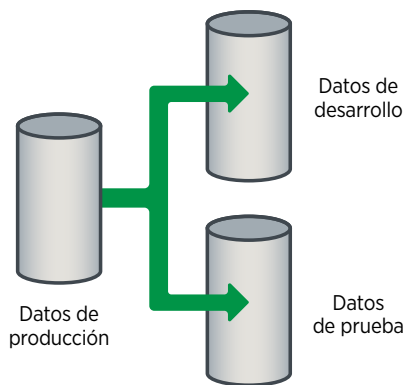
Transformación del modo de trabajar

Aparte de la posibilidad de aprovisionar de forma dinámica volúmenes persistentes, herramientas como Trident permiten a los equipos de DevOps hacer muchas más cosas.

Por ejemplo, hacen posible el acceso bajo demanda y mediante código a funciones de eficiencia del almacenamiento de NetApp como las copias y el clonado de Snapshot™. Estas funciones pueden cambiar por completo el trabajo de los equipos de desarrollo o QA que necesiten hacer más en menos tiempo y con el mínimo de recursos.

El uso de copias y el clonado de Snapshot mediante NetApp y Trident permite lo siguiente:

- **Creación rápida y en tiempo real de clones de conjuntos de datos de producción completos.** Significa que es posible crear nuevos espacios de trabajo de desarrollo o prueba de forma dinámica y en cuestión de segundos mediante solo unas líneas de código. Consulte la figura 4.
- **Recuperación de datos rápida y sencilla.** Con las copias de Snapshot, los desarrolladores pueden revertir rápidamente conjuntos de datos a una versión anterior. Es útil para probar código, ya que los desarrolladores pueden iterar rápidamente y sin miedo a tener que recrear los conjuntos de datos de prueba.
- **Ahorro de capacidad de almacenamiento con las copias o clones de Snapshot.** Las organizaciones que realizan múltiples clones de sus datos de producción para desarrollo o prueba suelen experimentar ahorros de capacidad de almacenamiento de entre el 40 y el 90 %⁷.



Clonado para permitir rápidamente flujos de trabajo de desarrollo y prueba

(Figura 4) La capacidad del clonado de NetApp para acelerar los flujos de trabajo de DevOps.

Obtenga más información sobre cómo funcionan con Trident el clonado de NetApp y Snapshot:

- [Clonado de volúmenes en autoservicio con Kubernetes⁸](#)
- [Copias de Snapshot y recuperación de volúmenes en autoservicio con Trident⁹](#)
- Este [podcast](#) de Tech ONTAP describe cómo Jeff Steiner, experto en NetApp Oracle, clonó una gran base de datos de Oracle en solo 22 segundos utilizando Docker with Trident¹⁰

Le ayudamos a usted y a su negocio

Resolver problemas de desarrolladores e ingenieros no es una novedad para NetApp. De hecho, es parte de nuestra herencia. Aprendimos pronto cómo podía utilizarse la infraestructura de almacenamiento para hacer posible y acelerar el importante trabajo de los desarrolladores e ingenieros, de modo que pudieran alcanzar los objetivos de sus empresas. En NetApp ayudamos a las organizaciones a mejorar la forma de gestionar sus datos y consumir almacenamiento. Vemos los ecosistemas abiertos, por ejemplo, los contenedores, como nuevas y florecientes áreas de innovación. Esta innovación recibe el impulso de los miembros de la comunidad. NetApp es parte de esta comunidad y se

esfuerza activamente para desarrollar nuevas capacidades. Con este fin, trabajamos en muchas direcciones para que a los miembros de la comunidad les resulte más fácil consumir almacenamiento donde y cuando lo necesiten. Hoy en día, NetApp se enorgullece de hacer posible un consumo de almacenamiento fluido en una variedad de ecosistemas abiertos. Seguimos desarrollando uno de los conjuntos más completos de API e integraciones para entornos como Docker, Kubernetes, OpenShift, OpenStack, Ansible, Chef, Puppet y más.

Trident no es más que un ejemplo de estos esfuerzos. Le animamos a que pruebe Trident con los entornos de contenedores de su organización y a que compruebe los increíbles datos de eficiencia y ahorro que logra en su canalización de DevOps.

Recursos adicionales

Para obtener más información sobre Trident y cualquier otra integración DevOps de NetApp, no deje de consultar los siguientes recursos:

Más información sobre NetApp y Trident

[Soluciones NetApp para contenedores](#)



Información general sobre funciones de Trident: [presentación de Trident](#)



[Introducción al almacenamiento persistente Kubernetes con Trident](#)



Clonado: [presentación del clonado de volúmenes con Trident para Kubernetes](#)



[Documentación de Trident](#)



[Descarga de Trident GitHub](#)



Más información sobre NetApp con DevOps

[Soluciones de NetApp para DevOps](#)



[thePub \(netapp.io\)](#)



Canal de NetApp en Slack ([netapp.io/slack](#))



[@NetAppPub](#)



Notas finales

¹ "Meeting Challenges in Using and Deploying Containers", por Sarah Conway, 27 de abril de 2017, Cloud Native Computing Foundation, <https://www.cncf.io/blog/2017/04/27/meeting-challenges-using-deploying-containers/>. Reproducido con un círculo rojo añadido en virtud de la [licencia Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/).

² "DevOps Through Desired State", presentado por Andrew Sullivan, NetApp, día 14, 11 de mayo de 2017, Tech Field Day, <https://www.youtube.com/watch?v=btLZ17M6gnY&list=PLInuRwPnsHacYmunO7zyES6SyrsrFfu5O&index=4>.

³ La documentación más reciente de Trident se puede encontrar en <https://netapp-trident.readthedocs.io/>.

⁴ "Using Trident for Dynamic Storage Provisioning with OpenShift", demostración en línea, 3:20 minutos, por The Pub @ NetApp, 17 de febrero de 2017, <https://www.youtube.com/watch?v=97VZWYssl2E>.

⁵ "Managing Persistent Data in Kubernetes", demostración en línea, 23:53 minutos, por The Pub @ NetApp, 15 de mayo de 2017, <https://www.youtube.com/watch?v=XluN9lvG2wM>.

⁶ "DARZ Docker & Container-as-a-Service Drives Digital Transformation Through DevOps", historia de éxito de un cliente, NetApp, 2017, <https://www.netapp.com/us/media/cs-darz-devops.pdf>.

⁷ "How NetApp IT Shortened Development Cycles Using FlexClone", blog de la comunidad NetApp, por Gopal Parthasarathy, 8 de octubre de 2015, <https://community.netapp.com/t5/Technology/How-NetApp-IT-Shortened-Development-Cycles-Using-FlexClone/ba-p/110581>.

⁸ "Trident 18.01 beta 1: Introducing volume cloning to Kubernetes!", por Garrett Mueller, NetApp, 14 de diciembre de 2017, <https://netapp.io/2017/12/14/trident-18-01-beta-1-introducing-volume-cloning-kubernetes/>. (Véase también: "Trident 18.01 is Here", por Andrew Sullivan, NetApp, 25 de enero de 2018, <https://netapp.io/2018/01/25/trident-18-01/>).

⁹ "Self-Service Data Recovery using Trident and NFS", por Andrew Sullivan, 3 de abril de 2018, NetApp, <https://netapp.io/2018/04/03/self-service-data-recovery-using-trident-nfs/>.

¹⁰ "Episode 99 - Databases as a Service: Containers", podcast de Tech ONTAP, 2017, https://soundcloud.com/techontap_podcast/episode-99-databases-as-a-service-containers.

Consulte el apartado de la [Herramienta de Matriz de Interoperabilidad \(IMT\)](#) en el sitio web de soporte de NetApp para confirmar que las versiones exactas del producto y las funciones descritas en este documento son compatibles con su entorno concreto. La herramienta IMT de NetApp define los componentes y las versiones del producto que pueden utilizarse para crear configuraciones que sean compatibles con NetApp. Los resultados específicos dependen de la instalación que realice cada cliente de acuerdo con las especificaciones publicadas.

Información de copyright

Copyright © 2018 NetApp, Inc. Todos los derechos reservados. Imprimido en EE. UU. No se puede reproducir este documento protegido por copyright ni parte del mismo de ninguna forma ni por ningún medio (gráfico, electrónico o mecánico, incluidas fotocopias, grabaciones o almacenamiento en un sistema de recuperación electrónico) sin la autorización previa y por escrito del propietario del copyright.

El software derivado del material de NetApp con copyright está sujeto a la siguiente licencia y exención de responsabilidad:

ESTE SOFTWARE LO PROPORCIONA NETAPP «TAL CUAL» Y SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITAR, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO, CUYA RESPONSABILIDAD QUEDA EXIMIDA POR EL PRESENTE DOCUMENTO. EN NINGÚN CASO NETAPP SERÁ RESPONSABLE DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL, EJEMPLAR O RESULTANTE (INCLUYENDO, ENTRE OTROS, LA OBTENCIÓN DE BIENES O SERVICIOS SUSTITUTIVOS, PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS, O INTERRUPCIÓN DE LA ACTIVIDAD EMPRESARIAL) CUALQUIERA SEA EL MODO EN EL QUE SE PRODUJERON Y LA TEORÍA DE RESPONSABILIDAD QUE SE APLIQUE, YA SEA EN CONTRATO, RESPONSABILIDAD OBJETIVA O AGRAVIO (INCLUIDA LA NEGLIGENCIA U OTRO TIPO), QUE SURJAN DE ALGÚN MODO DEL USO DE ESTE SOFTWARE, INCLUSO SI HUBIEREN SIDO ADVERTIDOS DE LA POSIBILIDAD DE TALES DAÑOS.

NetApp se reserva el derecho de modificar cualquiera de los productos aquí descritos en cualquier momento y sin aviso previo. NetApp no asume ningún tipo de responsabilidad que surja del uso de los productos aquí descritos, excepto aquello expresamente acordado por escrito por parte de NetApp. El uso o adquisición de este producto no lleva implícita ninguna licencia con derechos de patente, de marcas comerciales o cualquier otro derecho de propiedad intelectual de NetApp.

Es posible que el producto que se describe en este manual esté protegido por una o más patentes de EE. UU., patentes extranjeras o solicitudes pendientes.

LEYENDA DE DERECHOS LIMITADOS: El uso, la copia o la divulgación por parte del gobierno están sujetos a las restricciones establecidas en el subpárrafo (c)(1)(ii) de la cláusula sobre derechos de datos técnicos y software informático de DFARS 252.277-7103 (octubre de 1988) y FAR 52-227-19 (junio de 1987).

Información sobre marcas

NETAPP, el logotipo de NETAPP y las marcas que constan en <http://www.netapp.com/TM> son marcas registradas de NetApp, Inc. El resto de nombres de empresa y de producto pueden ser marcas registradas de sus respectivos propietarios.

WP-7270-071-esES