



Conceptos

ONTAP Select

NetApp
January 29, 2026

Tabla de contenidos

- Conceptos 1
 - Fundación de servicios web REST para implementar y administrar clústeres ONTAP Select 1
 - Arquitectura y restricciones clásicas 1
 - Recursos y representación estatal 1
 - Puntos finales URI 1
 - Mensajes HTTP 1
 - Formato JSON 2
 - Cómo acceder a la API de implementación de ONTAP Select 2
 - Implementar la interfaz de usuario nativa de la utilidad 2
 - Página de documentación en línea de ONTAP Select Deploy 2
 - Programa personalizado 2
 - Control de versiones de la API de implementación de ONTAP Select 2
 - Características operativas básicas de la API de implementación de ONTAP Select 3
 - Host de hipervisor versus ONTAP Select 3
 - Identificadores de objetos 3
 - Identificadores de solicitud 3
 - Llamadas sincrónicas y asincrónicas 3
 - Confirmar la finalización de un trabajo de larga duración 4
 - Seguridad 4
 - Transacción de API de solicitud y respuesta para ONTAP Select 4
 - Variables de entrada que controlan una solicitud de API 4
 - Interpretar una respuesta de API 6
 - Procesamiento asincrónico mediante el objeto Job para ONTAP Select 7
 - Solicitudes asincrónicas descritas mediante el objeto Trabajo 8
 - Consultar el objeto de trabajo asociado con una solicitud de API 8
 - Procedimiento general para emitir una solicitud asincrónica 8

Conceptos

Fundación de servicios web REST para implementar y administrar clústeres ONTAP Select

La Transferencia de Estado Representacional (REST) es un estilo para crear aplicaciones web distribuidas. Al aplicarla al diseño de una API de servicios web, establece un conjunto de tecnologías y mejores prácticas para exponer recursos basados en servidor y gestionar sus estados. Utiliza protocolos y estándares convencionales para proporcionar una base flexible para la implementación y gestión de clústeres de ONTAP Select .

Arquitectura y restricciones clásicas

REST fue articulado formalmente por Roy Fielding en su tesis doctoral. "[disertación](#)" En la Universidad de California en Irvine en el año 2000. Define un estilo arquitectónico mediante un conjunto de restricciones que, en conjunto, mejoran las aplicaciones web y los protocolos subyacentes. Las restricciones establecen una aplicación de servicios web RESTful basada en una arquitectura cliente-servidor que utiliza un protocolo de comunicación sin estado.

Recursos y representación estatal

Los recursos son los componentes básicos de un sistema web. Al crear una aplicación de servicios web REST, las primeras tareas de diseño incluyen:

- Identificación de recursos del sistema o del servidor. Todo sistema utiliza y mantiene recursos. Un recurso puede ser un archivo, una transacción comercial, un proceso o una entidad administrativa. Una de las primeras tareas al diseñar una aplicación basada en servicios web REST es identificar los recursos.
- Definición de los estados de los recursos y las operaciones de estado asociadas. Los recursos siempre se encuentran en uno de un número finito de estados. Los estados, así como las operaciones asociadas que se utilizan para modificarlos, deben estar claramente definidos.

Se intercambian mensajes entre el cliente y el servidor para acceder y cambiar el estado de los recursos de acuerdo con el modelo genérico CRUD (Crear, Leer, Actualizar y Eliminar).

Puntos finales URI

Cada recurso REST debe definirse y estar disponible mediante un esquema de direccionamiento bien definido. Los puntos finales donde se ubican e identifican los recursos utilizan un Identificador Uniforme de Recursos (URI). El URI proporciona un marco general para crear un nombre único para cada recurso en la red. El Localizador Uniforme de Recursos (URL) es un tipo de URI que se utiliza con servicios web para identificar y acceder a los recursos. Los recursos suelen exponerse en una estructura jerárquica similar a la de un directorio de archivos.

Mensajes HTTP

El Protocolo de Transferencia de Hipertexto (HTTP) es el protocolo que utilizan el cliente y el servidor de servicios web para intercambiar mensajes de solicitud y respuesta sobre los recursos. Al diseñar una aplicación de servicios web, los verbos HTTP (como GET y POST) se asignan a los recursos y a las acciones de gestión de estado correspondientes.

HTTP no tiene estado. Por lo tanto, para asociar un conjunto de solicitudes y respuestas relacionadas en una misma transacción, se debe incluir información adicional en los encabezados HTTP que acompañan a los flujos de datos de solicitud/respuesta.

Formato JSON

Si bien la información se puede estructurar y transferir entre un cliente y un servidor de varias maneras, la opción más popular (y la que se utiliza con la API REST de Deploy) es la Notación de Objetos JavaScript (JSON). JSON es un estándar de la industria para representar estructuras de datos simples en texto plano y se utiliza para transferir información de estado que describe los recursos.

Cómo acceder a la API de implementación de ONTAP Select

Debido a la flexibilidad inherente de los servicios web REST, se puede acceder a la API de implementación de ONTAP Select de varias maneras diferentes.

Implementar la interfaz de usuario nativa de la utilidad

La principal forma de acceder a la API es a través de la interfaz web de ONTAP Select Deploy. El navegador realiza llamadas a la API y reformatea los datos según el diseño de la interfaz. También se accede a la API a través de la interfaz de línea de comandos de la utilidad Deploy.

Página de documentación en línea de ONTAP Select Deploy

La página de documentación en línea de ONTAP Select Deploy ofrece un punto de acceso alternativo al usar un navegador. Además de permitir la ejecución directa de llamadas a la API individuales, la página también incluye una descripción detallada de la API, incluyendo parámetros de entrada y otras opciones para cada llamada. Las llamadas a la API se organizan en varias áreas o categorías funcionales.

Programa personalizado

Puede acceder a la API de implementación mediante diversos lenguajes de programación y herramientas. Las opciones más populares incluyen Python, Java y cURL. Un programa, script o herramienta que utiliza la API actúa como un cliente de servicios web REST. Usar un lenguaje de programación le permite comprender mejor la API y le brinda la oportunidad de automatizar las implementaciones de ONTAP Select .

Control de versiones de la API de implementación de ONTAP Select

La API REST incluida con ONTAP Select Deploy tiene asignado un número de versión. Este número es independiente del número de versión de Deploy. Debe conocer la versión de la API incluida con su versión de Deploy y cómo esta podría afectar su uso de la API.

La versión actual de la utilidad de administración Deploy incluye la versión 3 de la API REST. Las versiones anteriores de la utilidad Deploy incluyen las siguientes versiones de la API:

Implementar 2.8 y posteriores

ONTAP Select Deploy 2.8 y todas las versiones posteriores incluyen la versión 3 de la API REST.

Implementar 2.7.2 y anteriores

ONTAP Select Deploy 2.7.2 y todas las versiones anteriores incluyen la versión 2 de la API REST.



Las versiones 2 y 3 de la API REST no son compatibles. Si actualiza a Deploy 2.8 o posterior desde una versión anterior que incluye la versión 2 de la API, debe actualizar todo el código existente que acceda directamente a la API, así como cualquier script que utilice la interfaz de línea de comandos.

Características operativas básicas de la API de implementación de ONTAP Select

Si bien REST establece un conjunto común de tecnologías y mejores prácticas, los detalles de cada API pueden variar según las opciones de diseño. Debe conocer los detalles y las características operativas de la API ONTAP Select Deploy antes de usarla.

Host de hipervisor versus ONTAP Select

Un host de hipervisor es la plataforma de hardware principal que aloja una máquina virtual de ONTAP Select. Cuando una máquina virtual de ONTAP Select se implementa y está activa en un host de hipervisor, se considera un nodo de ONTAP Select. Con la versión 3 de la API REST de Deploy, los objetos de host y nodo son independientes. Esto permite una relación de uno a muchos, donde uno o más nodos de ONTAP Select pueden ejecutarse en el mismo host de hipervisor.

Identificadores de objetos

A cada instancia u objeto de recurso se le asigna un identificador único al crearse. Estos identificadores son únicos globalmente dentro de una instancia específica de ONTAP Select Deploy. Tras emitir una llamada a la API que crea una nueva instancia de objeto, el valor de ID asociado se devuelve al emisor en el `location` Encabezado de la respuesta HTTP. Puede extraer el identificador y usarlo en llamadas posteriores al hacer referencia a la instancia del recurso.



El contenido y la estructura interna de los identificadores de objeto pueden cambiar en cualquier momento. Solo debe usar los identificadores en las llamadas a la API correspondientes según sea necesario al referirse a los objetos asociados.

Identificadores de solicitud

A cada solicitud de API exitosa se le asigna un identificador único. El identificador se devuelve en el `request-id` Encabezado de la respuesta HTTP asociada. Puede usar un identificador de solicitud para referirse colectivamente a las actividades de una única transacción de solicitud-respuesta de API específica. Por ejemplo, puede recuperar todos los mensajes de evento de una transacción según el ID de la solicitud.

Llamadas sincrónicas y asincrónicas

Hay dos formas principales en que un servidor ejecuta una solicitud HTTP recibida de un cliente:

- Sincrónico El servidor ejecuta la solicitud inmediatamente y responde con un código de estado de 200, 201 o 204.
- Asíncrono: El servidor acepta la solicitud y responde con un código de estado 202. Esto indica que el servidor aceptó la solicitud del cliente e inició una tarea en segundo plano para completarla. El éxito o el

fracaso final no está disponible de inmediato y debe determinarse mediante llamadas API adicionales.

Confirmar la finalización de un trabajo de larga duración

Generalmente, cualquier operación que tarde mucho tiempo en completarse se procesa asincrónicamente mediante una tarea en segundo plano en el servidor. Con la API REST de Deploy, cada tarea en segundo plano está anclada por un objeto Job que la rastrea y proporciona información, como su estado actual. Un objeto Job, con su identificador único, se devuelve en la respuesta HTTP tras crear una tarea en segundo plano.

Puede consultar directamente el objeto "Job" para determinar si la llamada a la API asociada se realizó correctamente o no. Consulte "Procesamiento asíncrono con el objeto "Job"" para obtener más información.

Además de utilizar el objeto Trabajo, existen otras formas de determinar el éxito o el fracaso de una solicitud, entre ellas:

- Mensajes de evento: Puede recuperar todos los mensajes de evento asociados a una llamada API específica utilizando el ID de solicitud devuelto con la respuesta original. Los mensajes de evento suelen indicar si la operación fue correcta o no, y también pueden ser útiles al depurar una condición de error.
- Estado o estado del recurso Varios de los recursos mantienen un valor de estado o estado que puede consultar para determinar indirectamente el éxito o el fracaso de una solicitud.

Seguridad

La API de implementación utiliza las siguientes tecnologías de seguridad:

- Seguridad de la capa de transporte: Todo el tráfico enviado a través de la red entre el servidor de implementación y el cliente se cifra mediante TLS. No se admite el uso del protocolo HTTP en un canal sin cifrar. Se admite la versión 1.2 de TLS.
- Autenticación HTTP. La autenticación básica se utiliza para cada transacción de la API. Se añade a cada solicitud un encabezado HTTP que incluye el nombre de usuario y la contraseña en una cadena base64.

Transacción de API de solicitud y respuesta para ONTAP Select

Cada llamada a la API de Deploy se realiza como una solicitud HTTP a la máquina virtual de Deploy, la cual genera una respuesta asociada para el cliente. Este par de solicitud/respuesta se considera una transacción de API. Antes de usar la API de Deploy, debe familiarizarse con las variables de entrada disponibles para controlar una solicitud y el contenido de la salida de la respuesta.

Variables de entrada que controlan una solicitud de API

Puede controlar cómo se procesa una llamada API a través de parámetros establecidos en la solicitud HTTP.

Encabezados de solicitud

Debe incluir varios encabezados en la solicitud HTTP, incluidos:

- content-type Si el cuerpo de la solicitud incluye JSON, este encabezado debe establecerse en application/json.

- aceptar Si el cuerpo de la respuesta incluirá JSON, este encabezado debe configurarse como application/json.
- La autenticación básica debe configurarse con el nombre de usuario y la contraseña codificados en una cadena base64.

Cuerpo de la solicitud

El contenido del cuerpo de la solicitud varía según la llamada específica. El cuerpo de la solicitud HTTP consta de uno de los siguientes elementos:

- Objeto JSON con variables de entrada (por ejemplo, el nombre de un nuevo clúster)
- Vacío

Filtrar objetos

Al ejecutar una llamada a la API que usa GET, puede limitar o filtrar los objetos devueltos según cualquier atributo. Por ejemplo, puede especificar un valor exacto para que coincida:

```
<field>=<query value>
```

Además de la coincidencia exacta, existen otros operadores disponibles para devolver un conjunto de objetos en un rango de valores. ONTAP Select admite los operadores de filtrado que se muestran a continuación.

Operador	Descripción
=	Igual a
<	Menos que
>	Más que
≤	Menor o igual a
≥	Mayor o igual que
	O
!	No es igual a
*	Comodín codicioso

También puede devolver un conjunto de objetos en función de si un campo específico está configurado o no utilizando la palabra clave null o su negación (!null) como parte de la consulta.

Selección de campos de objeto

De forma predeterminada, al ejecutar una llamada a la API mediante GET, solo se devuelven los atributos que identifican de forma única el objeto o los objetos. Este conjunto mínimo de campos actúa como clave para cada objeto y varía según el tipo de objeto. Puede seleccionar propiedades adicionales del objeto mediante el parámetro de consulta "campos" de las siguientes maneras:

- Campos económicos Especificar `fields=*` para recuperar los campos de objeto que se mantienen en la memoria del servidor local o que requieren poco procesamiento para acceder.
- Campos costosos Especificar `fields=**` para recuperar todos los campos de objeto, incluidos aquellos que requieren procesamiento adicional del servidor para acceder.

- Selección de campo personalizado Usar `fields=FIELDNAME` Para especificar el campo exacto que desea. Al solicitar varios campos, los valores deben separarse con comas y sin espacios.



Como práctica recomendada, siempre debe identificar los campos específicos que desea. Solo debe recuperar el conjunto de campos económicos o costosos cuando sea necesario. NetApp determina la clasificación de económicos y costosos basándose en un análisis interno de rendimiento. La clasificación de un campo determinado puede cambiar en cualquier momento.

Ordenar objetos en el conjunto de salida

Los registros de una colección de recursos se devuelven en el orden predeterminado definido por el objeto. Puede cambiar el orden mediante el parámetro de consulta `order_by` con el nombre del campo y la dirección de ordenación, como se indica a continuación:

```
order_by=<field name> asc|desc
```

Por ejemplo, puede ordenar el campo de tipo en orden descendente seguido de id en orden ascendente:

```
order_by=type desc, id asc
```

Al incluir varios parámetros, debe separar los campos con una coma.

Paginación

Al ejecutar una llamada a la API mediante GET para acceder a una colección de objetos del mismo tipo, se devuelven todos los objetos coincidentes por defecto. Si es necesario, se puede limitar el número de registros devueltos mediante el parámetro de consulta `max_records` con la solicitud. Por ejemplo:

```
max_records=20
```

Si es necesario, puede combinar este parámetro con otros parámetros de consulta para limitar el conjunto de resultados. Por ejemplo, la siguiente operación devuelve hasta 10 eventos del sistema generados después del tiempo especificado:

```
time⇒ 2019-04-04T15:41:29.140265Z&max_records=10
```

Puedes emitir varias solicitudes para navegar por los eventos (o cualquier tipo de objeto). Cada llamada a la API posterior debe usar un nuevo valor de tiempo basado en el último evento del último conjunto de resultados.

Interpretar una respuesta de API

Cada solicitud de API genera una respuesta para el cliente. Puede examinarla para determinar si se realizó correctamente y recuperar datos adicionales según sea necesario.

Código de estado HTTP

A continuación se describen los códigos de estado HTTP utilizados por la API REST de implementación.

Código	Significado	Descripción
200	DE ACUERDO	Indica éxito para llamadas que no crean un nuevo objeto.
201	Creado	Se ha creado correctamente un objeto; el encabezado de respuesta de ubicación incluye el identificador único del objeto.
202	Aceptado	Se ha iniciado un trabajo en segundo plano de larga ejecución para ejecutar la solicitud, pero la operación aún no se ha completado.

Código	Significado	Descripción
400	Solicitud incorrecta	La entrada solicitada no se reconoce o es inadecuada.
403	Prohibido	Se deniega el acceso debido a un error de autorización.
404	Extraviado	El recurso al que se refiere la solicitud no existe.
405	Método no permitido	El verbo HTTP en la solicitud no es compatible con el recurso.
409	Conflicto	Se produjo un error al intentar crear un objeto porque el objeto ya existe.
500	Error interno	Se produjo un error interno general en el servidor.
501	No implementado	Se conoce el URI pero no es capaz de realizar la solicitud.

Encabezados de respuesta

Se incluyen varios encabezados en la respuesta HTTP generada por el servidor de implementación, incluidos:

- request-id A cada solicitud de API exitosa se le asigna un identificador de solicitud único.
- Ubicación Cuando se crea un objeto, el encabezado de ubicación incluye la URL completa del nuevo objeto, incluido el identificador de objeto único.

Cuerpo de la respuesta

El contenido de la respuesta asociada a una solicitud de API varía según el objeto, el tipo de procesamiento y si la solicitud se ha realizado correctamente o no. El cuerpo de la respuesta se representa en JSON.

- Objeto único. Se puede devolver un objeto único con un conjunto de campos según la solicitud. Por ejemplo, se puede usar GET para recuperar propiedades seleccionadas de un clúster mediante el identificador único.
- Múltiples objetos Se pueden devolver varios objetos de una colección de recursos. En todos los casos, se utiliza un formato consistente, con `num_records` Indica el número de registros y los registros que contienen una matriz de instancias de objeto. Por ejemplo, puede recuperar todos los nodos definidos en un clúster específico.
- Objeto de trabajo. Si una llamada a la API se procesa asincrónicamente, se devuelve un objeto de trabajo que ancla la tarea en segundo plano. Por ejemplo, la solicitud POST utilizada para implementar un clúster se procesa asincrónicamente y devuelve un objeto de trabajo.
- Objeto de error. Si se produce un error, siempre se devuelve un objeto de error. Por ejemplo, recibirá un error al intentar crear un clúster con un nombre ya existente.
- Vacío. En ciertos casos, no se devuelven datos y el cuerpo de la respuesta está vacío. Por ejemplo, el cuerpo de la respuesta está vacío después de usar DELETE para eliminar un host existente.

Procesamiento asincrónico mediante el objeto Job para ONTAP Select

Algunas llamadas a la API de Deploy, en particular las que crean o modifican un recurso, pueden tardar más en completarse que otras. ONTAP Select Deploy procesa estas solicitudes de larga duración de forma asincrónica.

Solicitudes asincrónicas descritas mediante el objeto Trabajo

Tras realizar una llamada a la API asincrónica, el código de respuesta HTTP 202 indica que la solicitud se ha validado y aceptado correctamente, pero aún no se ha completado. La solicitud se procesa como una tarea en segundo plano que continúa ejecutándose después de la respuesta HTTP inicial al cliente. La respuesta incluye el objeto Job que la ancla, incluyendo su identificador único.



Debe consultar la página de documentación en línea de ONTAP Select Deploy para determinar qué llamadas API funcionan de forma asincrónica.

Consultar el objeto de trabajo asociado con una solicitud de API

El objeto Job devuelto en la respuesta HTTP contiene varias propiedades. Puede consultar la propiedad de estado para determinar si la solicitud se completó correctamente. Un objeto Job puede estar en uno de los siguientes estados:

- En cola
- Correr
- Éxito
- Falla

Hay dos técnicas que puede utilizar al sondear un objeto de trabajo para detectar un estado terminal para la tarea, ya sea éxito o fracaso:

- Solicitud de sondeo estándar El estado actual del trabajo se devuelve inmediatamente
- Solicitud de sondeo largo El estado del trabajo se devuelve solo cuando ocurre una de las siguientes situaciones:
 - El estado ha cambiado más recientemente que el valor de fecha y hora proporcionado en la solicitud de encuesta
 - El valor de tiempo de espera ha expirado (de 1 a 120 segundos)

El sondeo estándar y el sondeo largo utilizan la misma llamada API para consultar un objeto de trabajo. Sin embargo, una solicitud de sondeo largo incluye dos parámetros de consulta: `poll_timeout` y `last_modified`.



Siempre debe utilizar un sondeo largo para reducir la carga de trabajo en la máquina virtual de implementación.

Procedimiento general para emitir una solicitud asincrónica

Puede utilizar el siguiente procedimiento de alto nivel para completar una llamada API asincrónica:

1. Emite la llamada API asincrónica.
2. Recibir una respuesta HTTP 202 indicando la aceptación exitosa de la solicitud.
3. Extraiga el identificador del objeto Trabajo del cuerpo de la respuesta.
4. Dentro de un bucle, realice lo siguiente en cada ciclo:
 - a. Obtenga el estado actual del trabajo con una solicitud de sondeo largo
 - b. Si el trabajo está en un estado no terminal (en cola, en ejecución), realice el bucle nuevamente.

5. Detenerse cuando el trabajo alcance un estado terminal (éxito, fracaso).

Información de copyright

Copyright © 2026 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 (b)(3) de los derechos de datos técnicos y productos no comerciales de DFARS 252.227-7013 (FEB de 2014) y FAR 52.227-19 (DIC de 2007).

Los datos aquí contenidos pertenecen a un producto comercial o servicio comercial (como se define en FAR 2.101) y son propiedad de NetApp, Inc. Todos los datos técnicos y el software informático de NetApp que se proporcionan en este Acuerdo tienen una naturaleza comercial y se han desarrollado exclusivamente con fondos privados. El Gobierno de EE. UU. tiene una licencia limitada, irrevocable, no exclusiva, no transferible, no sublicenciable y de alcance mundial para utilizar los Datos en relación con el contrato del Gobierno de los Estados Unidos bajo el cual se proporcionaron los Datos. Excepto que aquí se disponga lo contrario, los Datos no se pueden utilizar, desvelar, reproducir, modificar, interpretar o mostrar sin la previa aprobación por escrito de NetApp, Inc. Los derechos de licencia del Gobierno de los Estados Unidos de América y su Departamento de Defensa se limitan a los derechos identificados en la cláusula 252.227-7015(b) de la sección DFARS (FEB de 2014).

Información de la marca comercial

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