Conceptos

ONTAP Select

NetApp
November 08, 2026

This PDF was generated from https://docs.netapp.com/es-es/ontap-select-9191/concept_api_rest.html on November 08, 2026. Always check docs.netapp.com for the latest.

Tabla de contenidos

Conceptos . . . 1
- Infórmate sobre los servicios web REST para implementar y gestionar clústeres de ONTAP Select . . . 1
  - Arquitectura y limitaciones clásicas . . . 1
  - Recursos y representación de estado . . . 1
  - Puntos finales URI . . . 1
  - Mensajes de HTTP . . . 1
  - Formato JSON . . . 2
- Infórmate sobre cómo acceder a la API de ONTAP Select Deploy . . . 2
  - Implementa la interfaz de usuario nativa de la utilidad . . . 2
  - Página de documentación en línea de ONTAP Select Deploy . . . 2
  - Programa personalizado . . . 2
- Características operativas básicas de la API de ONTAP Select Deploy . . . 2
  - Host del hipervisor frente a nodo ONTAP Select . . . 3
  - Identificadores de objetos . . . 3
  - Identificadores de solicitud . . . 3
  - Llamadas síncronas y asíncronas . . . 3
  - Confirma la finalización de un trabajo de larga duración . . . 3
  - 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 asíncrono mediante el objeto Job para ONTAP Select . . . 7
  - Solicitudes asíncronas descritas mediante el objeto Job . . . 7
  - Consultar el objeto Job asociado a una solicitud de API . . . 7
  - Procedimiento general para emitir una solicitud asíncrona . . . 8

Conceptos

Infórmate sobre los servicios web REST para implementar y gestionar clústeres de ONTAP Select

Representational State Transfer (REST) es un estilo para crear aplicaciones web distribuidas. Al aplicarse al diseño de una API de servicios web, establece un conjunto de tecnologías y buenas prácticas para exponer recursos basados en servidores y gestionar sus estados. Utiliza protocolos y estándares convencionales para proporcionar una base flexible para la implementación y gestión de clústeres ONTAP Select.

Arquitectura y limitaciones clásicas

REST fue formalmente descrito por Roy Fielding en su tesis doctoral "disertación" en UC Irvine en 2000. Define un estilo arquitectónico mediante un conjunto de restricciones, que en conjunto han mejorado 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 de estado

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

- Identificación de recursos del sistema o del servidor Cada 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 estados de recursos y operaciones de estado asociadas Los recursos siempre se encuentran en uno de un número finito de estados. Tanto los estados como las operaciones asociadas que modifican dichos estados deben estar claramente definidos.

Se intercambian mensajes entre el cliente y el servidor para acceder a los recursos y modificar su estado según el modelo CRUD (Crear, Leer, Actualizar y Eliminar) genérico.

Puntos finales URI

Cada recurso REST debe definirse y estar disponible mediante un esquema de direccionamiento bien definido. Los endpoints 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 Uniform Resource Locator (URL) es un tipo de URI que se utiliza con los servicios web para identificar y acceder a los recursos. Los recursos suelen exponerse en una estructura jerárquica similar a un directorio de archivos.

Mensajes de 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. Como parte del diseño de 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 mantiene estado. Por lo tanto, para asociar un conjunto de solicitudes y respuestas relacionadas

bajo 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 diversas maneras, la opción más popular (y la que se utiliza con la API de REST de Deploy) es JavaScript Object Notation (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.

Infórmate sobre cómo acceder a la API de ONTAP Select Deploy

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

> La API de REST incluida con ONTAP Select Deploy tiene asignado un número de versión. El número de versión de la API es independiente del número de versión de Deploy. La utilidad de administración de ONTAP Select 9.19.1 Deploy incluye la versión 3 de la API de REST.

Implementa la interfaz de usuario nativa de la utilidad

La principal forma de acceder a la API es a través de la interfaz web de usuario de ONTAP Select Deploy. El navegador realiza llamadas a la API y reformatea los datos según el diseño de la interfaz de usuario. También accedes a la API mediante 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 ejecutar llamadas a la API directamente, la página también incluye una descripción detallada de la API, incluidos los parámetros de entrada y otras opciones para cada llamada. Las llamadas a la API están organizadas en varias áreas o categorías funcionales diferentes.

Programa personalizado

Puedes acceder a la API de Deploy utilizando diversos lenguajes y herramientas de programación. Entre las opciones más populares se encuentran Python, Java y cURL. Un programa, script o herramienta que utilice la API actúa como cliente de servicios web REST. El uso de un lenguaje de programación te permite comprender mejor la API y te da la oportunidad de automatizar las implementaciones de ONTAP Select.

Características operativas básicas de la API de ONTAP Select Deploy

Si bien REST establece un conjunto común de tecnologías y mejores prácticas, los detalles de cada API pueden variar según las decisiones de diseño. Debes familiarizarte con los detalles y las características operativas de la API de ONTAP Select Deploy antes de usar la API.

Host del hipervisor frente a nodo ONTAP Select

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

Identificadores de objetos

Cada instancia u objeto de recurso recibe un identificador único al crearse. Estos identificadores son únicos a nivel global dentro de una instancia específica de ONTAP Select Deploy. Tras realizar una llamada a la API que crea una nueva instancia de objeto, el valor del identificador asociado se devuelve al emisor en el `location` encabezado de la respuesta HTTP. Puedes extraer el identificador y usarlo en llamadas posteriores cuando te refieras a la instancia del recurso.



El contenido y la estructura interna de los identificadores de objeto pueden cambiar en cualquier momento. Debes utilizar los identificadores únicamente en las llamadas a la API de REST correspondientes, cuando sea necesario para referirte a los objetos asociados.

Identificadores de solicitud

Cada solicitud API exitosa recibe un identificador único. El identificador se devuelve en el `request-id` header de la respuesta HTTP asociada. Puedes usar un identificador de solicitud para referirte colectivamente a las actividades de una transacción API de solicitud-respuesta específica. Por ejemplo, puedes recuperar todos los mensajes de evento de una transacción basándote en el request id.

Llamadas síncronas y asíncronas

Existen dos formas principales en que un servidor procesa una solicitud HTTP recibida de un cliente:

- Síncrono El servidor realiza la solicitud inmediatamente y responde con un código de estado 200, 201 o 204.
- Asíncrono. El servidor acepta la solicitud y responde con un código de estado 202. Esto indica que el servidor ha aceptado la solicitud del cliente y ha iniciado una tarea en segundo plano para completarla. El resultado final (éxito o fracaso) no está disponible de inmediato y debe determinarse mediante llamadas adicionales a la API de REST.

Confirma la finalización de un trabajo de larga duración

Por lo general, cualquier operación que pueda tardar mucho tiempo en completarse se procesa de forma asíncrona mediante una tarea en segundo plano en el servidor. Con la API de REST de Deploy, cada tarea en segundo plano está vinculada a un objeto Job que realiza un seguimiento de la tarea y proporciona información, como el estado actual. Un objeto Job, incluido su identificador único, se devuelve en la respuesta HTTP después de que se crea una tarea en segundo plano.

Puedes consultar directamente el objeto Job para determinar el éxito o el fallo de la llamada a la API asociada. Consulta *procesamiento asíncrono mediante el objeto Job* para obtener más información.

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

- Mensajes de evento Puedes recuperar todos los mensajes de evento asociados a una llamada específica a la API utilizando el id de solicitud que se devuelve con la respuesta original. Los mensajes de evento suelen contener una indicación de éxito o fallo, y también pueden ser útiles al depurar una condición de error.
- Estado o condición del recurso Varios de los recursos mantienen un valor de estado o condición que puedes consultar para determinar indirectamente el éxito o el fracaso de una solicitud.

Seguridad

La API de Deploy 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 Deploy y el cliente está cifrado mediante TLS. No se admite el uso del protocolo HTTP a través de un canal no cifrado. Se admite la versión 1.2 de TLS.
- La autenticación HTTP básica se utiliza para cada transacción de API. A cada solicitud se le agrega 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 solicitud/respuesta se considera una transacción de API. Antes de usar la API de Deploy, debes familiarizarte 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

Puedes controlar cómo se procesa una llamada a la API mediante parámetros establecidos en la solicitud HTTP.

Encabezados de la solicitud

Debes incluir varios encabezados en la solicitud HTTP, entre ellos:

- content-type Si el cuerpo de la solicitud incluye JSON, este encabezado debe establecerse en application/json.
- accept Si el cuerpo de la respuesta incluye JSON, este encabezado debe establecerse en 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 (como, por ejemplo, el nombre de un nuevo clúster)
- Vacío

Filtrar objetos

Al realizar una llamada a la API que utiliza GET, puedes limitar o filtrar los objetos devueltos en función de cualquier atributo. Por ejemplo, puedes 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 dentro de 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 que
>=	Mayor o igual que
	O
!	No es igual a
*	Comodín codicioso

También puedes devolver un conjunto de objetos en función de si un campo específico está definido o no, usando la palabra clave null o su negación (!null) como parte de la consulta.

Selección de campos de objeto

Por defecto, al realizar 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. Puedes seleccionar propiedades adicionales del objeto usando el parámetro de consulta fields de las siguientes maneras:

- Campos económicos Especifica `fields=*` para recuperar los campos del objeto que se mantienen en la memoria del servidor local o que requieren poco procesamiento para acceder a ellos.
- Campos costosos Especifica `fields=**` para recuperar todos los campos del objeto, incluidos aquellos que requieren procesamiento adicional del servidor para acceder a ellos.
- Selección de campo personalizado Usa `fields=FIELDNAME` para especificar el campo exacto que quieres. Al solicitar varios campos, los valores deben estar separados por comas sin espacios.



Como buena práctica, siempre debes identificar los campos específicos que quieres. Solo debes recuperar el conjunto de campos económicos o costosos cuando lo necesites. La clasificación de campos económicos y costosos la determina NetApp según un análisis interno de rendimiento. La clasificación de un campo determinado puede cambiar en cualquier momento.

Ordena los objetos en el conjunto de salida

Los registros de una colección de recursos se devuelven en el orden predeterminado definido por el objeto. Puedes cambiar el orden utilizando el parámetro de consulta order_by con el nombre del campo y la dirección

de ordenación de la siguiente manera:
`order_by=<field name> asc|desc`

Por ejemplo, puedes ordenar el campo tipo en orden descendente, seguido de id en orden ascendente:
`order_by=type desc, id asc`

Cuando incluyas varios parámetros, debes separar los campos con una coma.

Paginación

Al realizar una llamada a la API mediante GET para acceder a una colección de objetos del mismo tipo, se devuelven por defecto todos los objetos coincidentes. Si lo necesitas, puedes limitar el número de registros devueltos usando el parámetro de consulta max_records en la solicitud. Por ejemplo:
`max_records=20`

Si es necesario, puedes combinar este parámetro con otros parámetros de consulta para acotar el conjunto de resultados. Por ejemplo, la siguiente consulta devuelve hasta 10 eventos del sistema generados después de la hora especificada:
`time⇒ 2019-04-04T15:41:29.140265Z&max_records=10`

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

Interpretar una respuesta de API

Cada solicitud a la API genera una respuesta que se envía al cliente. Puedes examinar la respuesta para determinar si fue exitosa y recuperar datos adicionales según lo necesites.

Código de estado HTTP

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

Código	Significado	Descripción
200	OK	Indica el éxito para las llamadas que no crean un nuevo objeto.
201	Creado	Se ha creado un objeto correctamente; el encabezado de respuesta de ubicación incluye el identificador único del objeto.
202	Aceptado	Se ha iniciado un proceso en segundo plano de larga duración para realizar la solicitud, pero la operación aún no ha finalizado.
400	Solicitud incorrecta	La solicitud introducida no se reconoce o es inapropiada.
403	Prohibido	El acceso está denegado debido a un error de autorización.
404	No encontrado	El recurso al que se hace referencia en la solicitud no existe.
405	Método no permitido	El verbo HTTP en la solicitud no es compatible con el recurso.
409	Conflicto	Se ha producido un intento de crear un objeto, pero el objeto ya existe.
500	Error interno	Se ha producido un error interno general en el servidor.
501	No implementado	Se conoce la URI, pero no es capaz de realizar la solicitud.

Encabezados de respuesta

La respuesta HTTP generada por el servidor Deploy incluye varios encabezados, entre ellos:

- ID de solicitud A cada solicitud de API exitosa se le asigna un identificador de solicitud único.
- location Cuando se crea un objeto, el encabezado de location incluye la URL completa al nuevo objeto, incluido el identificador único del objeto.

Cuerpo de respuesta

El contenido de la respuesta asociada a una solicitud de API varía según el objeto, el tipo de procesamiento y el éxito o fracaso de la solicitud. El cuerpo de la respuesta se muestra en formato JSON.

- Objeto único: se puede devolver un único objeto con un conjunto de campos según la solicitud. Por ejemplo, puedes usar GET para recuperar propiedades seleccionadas de un clúster usando el identificador único.
- Se pueden devolver varios objetos de una colección de recursos. En todos los casos, se utiliza un formato consistente, donde `num_records` indica el número de registros y los registros contienen una matriz de instancias de objetos. Por ejemplo, se pueden recuperar todos los nodos definidos en un clúster específico.
- Objeto Job Si una llamada a la API se procesa de forma asíncrona, se devuelve un objeto Job que sirve de ancla para la tarea en segundo plano. Por ejemplo, la solicitud POST utilizada para implementar un clúster se procesa de forma asíncrona y devuelve un objeto Job.
- Objeto Error Si ocurre un error, siempre se devuelve un objeto Error. Por ejemplo, recibirás un error al intentar crear un clúster con un nombre que ya existe.
- En ciertos casos, no se devuelve ningún dato 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 asíncrono mediante el objeto Job para ONTAP Select

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

Solicitudes asíncronas descritas mediante el objeto Job

Tras realizar una llamada a la API que se ejecuta de forma asíncrona, 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 tras la respuesta HTTP inicial al cliente. La respuesta incluye el objeto Job que ancla la solicitud, incluido su identificador único.

Deberías consultar la página de documentación online de ONTAP Select Deploy para determinar qué llamadas a la API de REST funcionan de forma asíncrona.

Consultar el objeto Job asociado a una solicitud de API

El objeto Job devuelto en la respuesta HTTP contiene varias propiedades. Puedes consultar la propiedad state para determinar si la solicitud se ha completado correctamente. Un objeto Job puede encontrarse en uno de los siguientes estados:

- En cola
- En ejecución
- Éxito
- Fallo

Hay dos técnicas que puedes usar cuando sondeas un objeto Job para detectar un estado terminal de la tarea, ya sea éxito o fracaso:

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

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



Siempre debes usar el sondeo prolongado para reducir la carga de trabajo en la máquina virtual Deploy.

Procedimiento general para emitir una solicitud asíncrona

Puedes usar el siguiente procedimiento a grandes rasgos para completar una llamada asíncrona a la API:

1. Emite la llamada asíncrona a la API.
2. Recibe una respuesta HTTP 202 indicando la aceptación satisfactoria de la solicitud.
3. Extrae el identificador del objeto Job del cuerpo de la respuesta.
4. Dentro de un bucle, realiza lo siguiente en cada ciclo:
   a. Obtén el estado actual del Job con una petición long-poll
   b. Si el trabajo se encuentra en un estado no terminal (en cola, en ejecución), realiza el bucle de nuevo.
5. Detén cuando el Job alcanza 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.