



# **Detalles de implementación de REST**

## ONTAP automation

NetApp  
January 13, 2026

# Tabla de contenidos

Detalles de implementación de REST .....	1
Características operativas de la API de REST DE ONTAP .....	1
Transacción de API de solicitud y respuesta .....	1
Compatibilidad con operaciones CRUD .....	1
Identificadores de objeto .....	1
Instancias y colecciones de objetos .....	1
Operaciones síncronas y asíncronas .....	1
Seguridad .....	2
Variables de entrada para una solicitud de API DE REST DE ONTAP .....	3
Métodos HTTP .....	3
Variables de ruta .....	3
Solicitar encabezados .....	3
Solicitar el cuerpo .....	4
Filtrando objetos .....	4
Solicitando campos de objeto específicos .....	5
Ordenar objetos del conjunto de resultados .....	5
Paginación al recuperar objetos de una colección .....	6
Propiedades de tamaño .....	6
Interpretar una respuesta de la API de REST DE ONTAP .....	7
Código de estado HTTP .....	7
Encabezados de respuesta .....	7
Cuerpo de respuesta .....	8
Ligando HAL .....	8
Errores .....	8
Procesamiento asíncrono con la API de REST DE ONTAP .....	9
Controlar el modo en que se procesa una solicitud .....	9
Consulta del objeto Job asociado a una solicitud API .....	10
Procedimiento general para emitir una solicitud asincrónica .....	10
Acceso y referencias de objetos de la API de REST DE ONTAP .....	11
Rutas de acceso a objetos .....	11
Acceder a un objeto mediante el UUID .....	11
Obtener acceso a un objeto mediante una propiedad de objeto .....	11
Contexto de clúster frente a SVM .....	12
Uso DE PATCH y DELETE en una colección de objetos .....	12
Acceda a métricas de rendimiento con la API de REST DE ONTAP .....	12

# Detalles de implementación de REST

## Características operativas de la API de REST DE ONTAP

Mientras QUE REST establece un conjunto común de tecnologías y prácticas recomendadas, los detalles de cada API pueden variar en función de las opciones de diseño.

### Transacción de API de solicitud y respuesta

Cada llamada de API REST se realiza como una solicitud HTTP al sistema ONTAP, que genera una respuesta asociada al cliente. Este par de solicitud/respuesta se considera una transacción de API. Antes de utilizar la API, debería estar familiarizado con las variables de entrada disponibles para controlar una solicitud y el contenido de la salida de la respuesta.

### Compatibilidad con operaciones CRUD

Se accede a cada uno de los recursos disponibles a través de la API REST de ONTAP en función del modelo CRUD:

- Cree
- Lea
- Actualizar
- Eliminar

Para algunos de los recursos, solo se admite un subconjunto de las operaciones. Debe revisar la página de documentación de API de ONTAP en el clúster de ONTAP para obtener más información sobre cada recurso.

### Identificadores de objeto

A cada instancia u objeto de recurso se le asigna un identificador único cuando se crea. En la mayoría de los casos, el identificador es un UUID de 128 bits. Estos identificadores son únicos globalmente dentro de un clúster ONTAP específico. Después de emitir una llamada API que crea una nueva instancia de objeto, se devuelve una dirección URL con el valor de id asociado a la persona que llama en el encabezado de ubicación de la respuesta HTTP. Puede extraer el identificador y utilizarlo en llamadas posteriores cuando haga referencia a la instancia del recurso.

 El contenido y la estructura interna de los identificadores de objeto pueden cambiar en cualquier momento. Solo se deben usar los identificadores en las llamadas API aplicables según sea necesario cuando se hacen referencia a los objetos asociados.

### Instancias y colecciones de objetos

Dependiendo de la ruta de recursos y del método HTTP, una llamada API puede aplicarse a una instancia de objeto específica o a una colección de objetos.

### Operaciones síncronas y asíncronas

Hay dos maneras en que ONTAP realiza una solicitud HTTP recibida desde un cliente.

## Procesamiento sincrónico

ONTAP realiza la solicitud inmediatamente y responde con un código de estado HTTP de 200 o 201 si se realiza correctamente.

Cada solicitud que utilice los métodos GET, HEAD y OPTIONS se realiza siempre de forma síncrona. Además, las solicitudes que utilizan POST, PATCH y DELETE están diseñadas para ejecutarse de manera síncrona si se espera que se completen en menos de dos segundos.

## Procesamiento asíncrono

Si una solicitud asíncrona es válida, ONTAP crea una tarea en segundo plano para procesar la solicitud y un objeto de trabajo para anclar la tarea. El estado HTTP 202 se devuelve al llamante junto con el objeto de trabajo. Para determinar el éxito o fracaso final, debe recuperar el estado del trabajo.

Las solicitudes que utilizan los métodos POST, PATCH y DELETE están diseñadas para ejecutarse de manera asíncrona si se espera que tardan más de dos segundos en completarse.



La `return_timeout` El parámetro de consulta está disponible con llamadas API asíncronas y puede convertir una llamada asíncrona a completado de forma síncrona. Consulte "["Procesamiento asíncrono mediante el objeto Job"](#)" si quiere más información.

## Seguridad

La seguridad proporcionada con la API DE REST se basa principalmente en las funciones de seguridad existentes disponibles con ONTAP. La API utiliza la siguiente seguridad:

### Seguridad de la capa de transporte

Todo el tráfico enviado a través de la red entre el cliente y el LIF de ONTAP se cifra normalmente con TLS, según los ajustes de configuración de ONTAP.

### Autentificación de clientes

Las mismas opciones de autenticación disponibles con ONTAP System Manager y el SDK de gestión de red también se pueden usar con la API de REST de ONTAP.

### Autenticación HTTP

En un nivel HTTP, por ejemplo, cuando se accede a la API de REST DE ONTAP directamente, hay dos opciones de autenticación como se describe a continuación. En cada caso, debe crear un encabezado de autorización HTTP e incluirlo con cada solicitud.

Opción	Descripción
Autenticación básica HTTP	El nombre de usuario y la contraseña de ONTAP se concatenan con dos puntos. La cadena se convierte a base64 y se incluye en la cabecera de solicitud.
OAuth 2,0	A partir de ONTAP 9,14, puede solicitar un token de acceso desde un servidor de autorización externo e incluirlo como un token portador en el encabezado de la solicitud.

Para obtener más detalles sobre OAuth 2,0 y cómo se implementa en ONTAP, consulte "["Descripción general de la implementación de ONTAP OAuth 2,0"](#)". Consulte también "["Prepárese para usar los flujos de trabajo"](#)" abajo en este sitio.

## Autorización de ONTAP

ONTAP implementa un modelo de autorización basado en roles. La cuenta que utilice al acceder a la API DE REST de ONTAP o a la página de documentación de API deben tener la autoridad adecuada.

# Variables de entrada para una solicitud de API DE REST DE ONTAP

Puede controlar cómo se procesa una llamada API mediante parámetros y variables definidas en la solicitud HTTP.

## Métodos HTTP

En la siguiente tabla, se muestran los métodos HTTP compatibles con la API DE REST de ONTAP.



No todos los métodos HTTP están disponibles en cada extremo DE REST. Además, TANTO EL PARCHE como EL BORRADO se pueden utilizar en una colección. Consulte *Object reference and access* para obtener más información.

Método HTTP	Descripción
OBTENGA	Recupera propiedades de objeto en una instancia o colección de recursos.
PUBLICAR	Crea una nueva instancia de recurso basada en la entrada proporcionada.
PARCHE	Actualiza una instancia de recurso existente basada en la entrada proporcionada.
ELIMINAR	Elimina una instancia de recurso existente.
CABEZA	Emite una solicitud GET pero sólo devuelve los encabezados HTTP.
OPCIONES	Determine qué métodos HTTP se admiten en un punto final específico.

## Variables de ruta

La ruta de punto final que se usa con cada llamada API DE REST puede incluir varios identificadores. Cada ID corresponde a una instancia de recurso específica. Entre los ejemplos se incluyen el ID del clúster y el ID de SVM.

## Solicitar encabezados

Debe incluir varios encabezados en la solicitud HTTP.

### Tipo de contenido

Si el cuerpo de la solicitud incluye JSON, este encabezado debe estar establecido en application/json.

### Acepte

Este encabezado debe estar configurado en application/hal+json. Si en su lugar está establecido en application/json No se devolverá ninguno de los enlaces HAL excepto un enlace necesario para recuperar el siguiente lote de registros. Si el encabezado está separado de estos dos valores, el valor predeterminado de content-type la cabecera en la respuesta será application/hal+json.

## Autorización

La autenticación básica se debe establecer con el nombre de usuario y la contraseña codificados como una cadena base64. Por ejemplo:

```
Authorization: Basic YWRtaW46cGV0ZXJzb24=.
```

## Solicitar el cuerpo

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

- Objeto JSON con variables de entrada
- Objeto JSON vacío

## Filtrando objetos

Al emitir una llamada a la API con el método GET, puede limitar o filtrar los objetos devueltos en función de cualquier atributo mediante un parámetro de consulta.

### Análisis e interpretación de parámetros de consulta

Se puede agregar un conjunto de uno o más parámetros a la cadena de URL a partir del ? carácter. Si se proporciona más de un parámetro, los parámetros de consulta se dividen en función del & carácter. Cada clave y valor del parámetro se dividen en el = carácter.

Por ejemplo, puede especificar un valor exacto para que coincida utilizando el signo igual:

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

Para una consulta más compleja, el operador adicional se coloca después del signo igual. Por ejemplo, para seleccionar el juego de objetos basado en un campo específico que sea mayor o igual que algún valor, la consulta sería:

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

### Operadores de filtrado

Además de los ejemplos proporcionados anteriormente, hay operadores adicionales disponibles para devolver objetos en un rango de valores. En la siguiente tabla se muestra un resumen de los operadores de filtrado compatibles con la API de REST DE ONTAP.



Los campos que no están configurados generalmente se excluyen de consultas coincidentes.

Operador	Descripción
=	Igual a.
<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que
!	No es igual a.

*	Comodín codicioso
---	-------------------

También puede devolver un conjunto de objetos basándose en si se ha establecido o no un campo específico mediante el `null` palabra clave o su negación `!null` como parte de la consulta.

### Ejemplos de flujos de trabajo

A continuación se incluyen algunos ejemplos desde los flujos de trabajo de la API de REST en este sitio.

- ["Enumere los discos"](#)

Filtrar según el `state` variable para seleccionar los discos de repuesto.

## Solicitando campos de objeto específicos

De forma predeterminada, al emitir una llamada API mediante GET, sólo se devuelven los atributos que identifican de forma única el objeto o los objetos, junto con un vínculo de HAL. Este conjunto mínimo de campos actúa como clave para cada objeto y varía según el tipo de objeto. Puede seleccionar propiedades de objeto adicionales mediante la `fields` parámetro de consulta de las siguientes formas:

- Campos comunes o estándar

Especifique `fields=*` para recuperar los campos de objeto más utilizados. Estos campos normalmente se mantienen en la memoria del servidor local o requieren poco procesamiento para acceder. Estas son las mismas propiedades que se devuelven para un objeto después de utilizar GET con una clave de ruta de URL (UUID).

- Todos los campos

Especifique `fields=**` para recuperar todos los campos de objeto, incluidos los que requieren procesamiento de servidor adicional para tener acceso.

- Selección de campo personalizado

Uso `fields=<field_name>` para especificar el campo exacto que desea. Al solicitar varios campos, los valores deben separarse con comas sin espacios.



Como práctica recomendada, siempre debe identificar los campos específicos que desea. Sólo debe recuperar el conjunto de campos comunes o todos los campos cuando sea necesario. Los campos se clasifican como comunes y se devuelven mediante `fields=*`. Viene determinado por NetApp según un análisis interno del rendimiento. La clasificación de un campo puede cambiar en versiones futuras.

## Ordenar objetos del conjunto de resultados

Los registros de una colección de recursos se devuelven en el orden predeterminado definido por el objeto. Puede cambiar el pedido mediante `order_by` consulte el parámetro con el nombre del campo y la dirección de ordenación de la siguiente manera:

`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
```

Tenga en cuenta lo siguiente:

- Si se especifica un campo de ordenación pero no se proporciona una dirección, los valores se ordenan en orden ascendente.
- Cuando se incluyan varios parámetros, los campos deben separarse con una coma.

## Paginación al recuperar objetos de una colección

Al emitir una llamada API mediante GET para acceder a una colección de objetos del mismo tipo, ONTAP intenta devolver tantos objetos como sea posible basándose en dos restricciones. Puede controlar cada una de estas restricciones utilizando parámetros de consulta adicionales en la solicitud. La primera restricción alcanzada para una solicitud GET específica termina la solicitud y, por lo tanto, limita el número de registros devueltos.



Si una solicitud finaliza antes de iterar todos los objetos, la respuesta contiene el vínculo necesario para recuperar el siguiente lote de registros.

### Limitar el número de objetos

De forma predeterminada, ONTAP devuelve un máximo de 10,000 objetos para UNA solicitud GET. Puede cambiar este límite con `max_records` parámetro de consulta. Por ejemplo:

```
max_records=20
```

El número de objetos realmente devueltos puede ser menor que el máximo en efecto, basándose en la restricción de tiempo relacionada, así como en el número total de objetos del sistema.

### Limitar el tiempo utilizado para recuperar los objetos

De forma predeterminada, ONTAP devuelve tantos objetos como sea posible dentro del tiempo permitido para LA solicitud GET. El tiempo de espera predeterminado es 15 segundos. Puede cambiar este límite con `return_timeout` parámetro de consulta. Por ejemplo:

```
return_timeout=5
```

El número de objetos realmente devueltos puede ser menor que el máximo en efecto, basándose en la restricción relacionada en el número de objetos así como en el número total de objetos del sistema.

### Reducción del conjunto de resultados

Si es necesario, puede combinar estos dos parámetros con parámetros de consulta adicionales para restringir el conjunto de resultados. Por ejemplo, el siguiente devuelve hasta 10 eventos de ems generados después de la hora especificada:

```
time=> 2018-04-04T15:41:29.140265Z&max_records=10
```

Puede emitir varias solicitudes para desplazarse por los objetos. Cada llamada API posterior debe utilizar un nuevo valor de tiempo basado en el último evento del último conjunto de resultados.

## Propiedades de tamaño

Los valores de entrada utilizados con algunas llamadas API, así como ciertos parámetros de consulta son numéricos. En lugar de proporcionar un entero en bytes, puede usar de manera opcional un sufijo como se

muestra en la siguiente tabla.

Sufijo	Descripción
KB	Kilobytes de KB (1024 bytes) o kibibytes
MB	MB megabytes (KB x 1024 bytes) o mebibbytes
GB	GB Gigabytes (MB x 1024 bytes) o gibibbytes
TB	Terabytes de TB (GB x 1024 bytes) o tebibbytes
PB	Petabytes de PB (TB x 1024 bytes) o gibibbytes

#### Información relacionada

- ["Referencias de objeto y acceso"](#)

## Interpretar una respuesta de la API de REST DE ONTAP

Cada solicitud de API genera una respuesta al cliente. Debe examinar la respuesta para determinar si ha tenido éxito y recuperar datos adicionales según sea necesario.

### Código de estado HTTP

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

Codificación	Frase de motivo	Descripción
200	DE ACUERDO	Indica que las llamadas que no crean un objeto nuevo se han realizado correctamente.
201	Creado	Se ha creado correctamente un objeto. El encabezado de ubicación de la respuesta incluye el identificador único del objeto.
202	Aceptado	Se ha iniciado un trabajo en segundo plano para realizar la solicitud, pero aún no se ha completado.
400	Solicitud incorrecta	La entrada de la solicitud no se reconoce o no es apropiada.
401	No autorizado	Error en la autenticación del usuario.
403	Prohibido	Se deniega el acceso 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 método HTTP de la solicitud no es compatible con el recurso.
409	Conflicto	Se ha producido un error al intentar crear un objeto porque primero se debe crear otro objeto o ya existe el objeto solicitado.
500	Error interno	Se ha producido un error interno general en el servidor.

### Encabezados de respuesta

Se incluyen varios encabezados en la respuesta HTTP generada por ONTAP.

## **Ubicación**

Cuando se crea un objeto, el encabezado de ubicación incluye la dirección URL completa del nuevo objeto, incluido el identificador único asignado al objeto.

## **Tipo de contenido**

Esto será normalmente application/hal+json.

## **Cuerpo de respuesta**

El contenido del cuerpo de respuesta que resulta de una solicitud API varía en función del objeto, el tipo de procesamiento y el éxito o el fallo de la solicitud. La respuesta siempre se representa en JSON.

- Un solo objeto

Un solo objeto se puede devolver con un conjunto de campos basados en la solicitud. Por ejemplo, se puede usar GET para recuperar las propiedades seleccionadas de un clúster mediante el identificador único.

- Varios objetos

Se pueden devolver varios objetos de una colección de recursos. En todos los casos, existe un formato coherente utilizado, con num\_records indica el número de registros y registros que contienen una matriz de las instancias de objeto. Por ejemplo, puede recuperar los nodos definidos en un clúster específico.

- Objeto de trabajo

Si una llamada API se procesa de forma asíncrona, se devuelve un objeto Job que ancla la tarea en segundo plano. Por ejemplo, la solicitud DE REVISIÓN utilizada para actualizar la configuración del clúster se procesa de forma asíncrona y devuelve un objeto Job.

- Objeto de error

Si se produce un error, siempre se devuelve un objeto error. Por ejemplo, recibirá un error al intentar cambiar un campo no definido para un clúster.

- Objeto JSON vacío

En ciertos casos, no se devuelven datos y el cuerpo de respuesta incluye un objeto JSON vacío.

## **Ligando HAL**

La API REST de ONTAP usa HAL como mecanismo para apoyar a Hypermedia como Motor de aplicación Estado (HATEOAS). Cuando se devuelve un objeto o atributo que identifica un recurso específico, también se incluye un vínculo codificado con HAL que permite localizar y determinar fácilmente detalles adicionales sobre el recurso.

## **Errores**

Si se produce un error, se devuelve un objeto de error en el cuerpo de respuesta.

## **Formato**

Un objeto de error tiene el siguiente formato:

```

"error": {
  "message": "<string>",
  "code": <integer>[,,
  "target": "<string>"]
}

```

Puede utilizar el valor del código para determinar el tipo o la categoría de error general y el mensaje para determinar el error específico. Si está disponible, el campo de destino incluye la entrada de usuario específica asociada al error.

### códigos de error comunes

Los códigos de error comunes se describen en la siguiente tabla. Las llamadas API específicas pueden incluir códigos de error adicionales.

Codificación		Descripción
1	409	Ya existe un objeto con el mismo identificador.
2	400	El valor de un campo no es válido o falta, o se ha proporcionado un campo adicional.
3	400	La operación no es compatible.
4	405	No se puede encontrar un objeto con el identificador especificado.
6	403	Se deniega el permiso para realizar la solicitud.
8	409	El recurso está en uso.

## Procesamiento asíncrono con la API de REST DE ONTAP

Después de emitir una solicitud de API diseñada para ejecutarse de forma asíncrona, siempre se crea un objeto de trabajo y se devuelve a la persona que llama. El trabajo describe y ancla una tarea en segundo plano que procesa la solicitud. Dependiendo del código de estado HTTP, debe recuperar el estado del trabajo para determinar si la solicitud se realizó correctamente.

Consulte ["Referencia de API"](#) Para determinar qué llamadas API se han diseñado para realizarse de forma asíncrona.

### Controlar el modo en que se procesa una solicitud

Puede utilizar el `return_timeout` Parámetro de consulta para controlar cómo se procesa una llamada API asíncrona. Hay dos resultados posibles cuando se usa este parámetro.

#### El temporizador finaliza antes de que se complete la solicitud

Para solicitudes válidas, ONTAP devuelve un código de estado HTTP 202 junto con el objeto de trabajo.

Debe recuperar el estado del trabajo para determinar si la solicitud se completó correctamente.

### La solicitud se completa antes de que caduque el temporizador

Si la solicitud es válida y se completa correctamente antes de que caduque la hora, ONTAP devuelve un código de estado HTTP 200 junto con el objeto de trabajo. Dado que la solicitud se completa de forma síncrona, como lo indica 200, no es necesario recuperar el estado del trabajo.



El valor predeterminado para `return_timeout` el parámetro es cero segundos. Por lo tanto, si no incluye el parámetro, el código de estado 202 HTTP siempre se devuelve para una solicitud válida.

### Consulta del objeto Job asociado a una solicitud API

El objeto Job devuelto en la respuesta HTTP contiene varias propiedades. Puede consultar la propiedad `state` en una llamada API posterior para determinar si la solicitud se completó correctamente. Un objeto Job siempre está en uno de los estados siguientes:

#### estados no terminales

- En cola
- Ejecutando
- En pausa

#### estados de terminal

- Correcto
- Fallo

### Procedimiento general para emitir una solicitud asincrónica

Puede utilizar el siguiente procedimiento de alto nivel para completar una llamada API asincrónica. En este ejemplo se asume el `return_timeout` no se utiliza el parámetro o que la hora caduca antes de que finalice el trabajo en segundo plano.

1. Emite una llamada API que se haya diseñado para realizarse de forma asincrónica.
2. Recibir una respuesta HTTP 202 que indique la aceptación de una solicitud válida.
3. Extraiga el identificador del objeto Job del cuerpo de respuesta.
4. Dentro de un bucle temporizado, realice lo siguiente en cada ciclo:
  - a. Obtener el estado actual del trabajo.
  - b. Si el trabajo se encuentra en un estado no terminal, vuelva a realizar el bucle.
5. Deténgase cuando el trabajo alcance un estado terminal (correcto, fallo).

#### Información relacionada

- "[Actualice el contacto del clúster](#)"
- "[Obtener instancia de trabajo](#)"

# Acceso y referencias de objetos de la API de REST DE ONTAP

Es posible hacer referencia a las instancias o los objetos de recursos expuestos mediante la API DE REST de ONTAP y acceder a ellos de varias maneras diferentes.

## Rutas de acceso a objetos

En un nivel alto, hay dos tipos de ruta al acceder a un objeto:

- Primario

El objeto es el destino principal o directo de la llamada API.

- Extranjero

El objeto no es la referencia principal de la llamada API, sino que está vinculado desde el objeto principal. Por lo tanto, es un objeto externo o descendente y se hace referencia a través de un campo del objeto primario.

## Acceder a un objeto mediante el UUID

A cada objeto se le asigna un identificador único cuando se crea, que en la mayoría de los casos es un UUID de 128 bits. Los valores de UUID asignados son inmutables y se utilizan internamente en ONTAP para acceder a los recursos y gestionarlos. Por este motivo, el UUID generalmente proporciona la forma más rápida y estable de acceder a los objetos.

Para muchos de los tipos de recursos, se puede proporcionar un valor de UUID como parte de la clave de ruta en la URL para acceder a un objeto específico. Por ejemplo, puede usar los siguientes métodos para acceder a una instancia del nodo: `/cluster/nodes/{uuid}

## Obtener acceso a un objeto mediante una propiedad de objeto

Además de un UUID, también puede tener acceso a un objeto mediante una propiedad de objeto. En la mayoría de los casos, es conveniente utilizar la propiedad name. Por ejemplo, puede utilizar el siguiente parámetro de consulta en la cadena URL para acceder a una instancia de nodo por su nombre:

/cluster/nodes?name=node\_one. Además de un parámetro de consulta, se puede tener acceso a un objeto extraño a través de una propiedad del objeto primario.

Aunque puede utilizar el nombre u otra propiedad para tener acceso a un objeto en lugar del UUID, existen varias desventajas posibles:

- El campo de nombre no es inmutable y se puede cambiar. Si se cambia el nombre de un objeto antes de acceder a un objeto, se devuelve el objeto incorrecto o se producirá un error de acceso al objeto.



Este problema puede ocurrir con un método POST o PATCH en un objeto extraño o con un método GET en un objeto principal.

- ONTAP debe convertir el campo de nombre en el UUID correspondiente. Este es un tipo de acceso indirecto que puede convertirse en un problema de rendimiento.

En concreto, es posible una degradación del rendimiento cuando se da alguna de las siguientes situaciones:

- Se utiliza EL método GET
- Se accede a una gran colección de objetos
- Se utiliza una consulta compleja o elaborada

## Contexto de clúster frente a SVM

Existen varios extremos DE REST que admiten tanto un clúster como una SVM. Cuando se usa uno de estos extremos, se puede indicar el contexto de la llamada API a través de `scope=[svm|cluster]` valor. Entre los ejemplos de extremos que admiten un contexto doble se incluyen interfaces IP y roles de seguridad.



El valor de ámbito tiene un valor predeterminado base en las propiedades proporcionadas para cada llamada API.

## Uso DE PATCH y DELETE en una colección de objetos

Todos los extremos REST que admiten PARCHE o ELIMINACIÓN en una instancia de recurso también admiten el mismo método en una colección de objetos. El único requisito es que se debe proporcionar al menos un campo a través de un parámetro de consulta en la cadena URL. Al emitir UN PARCHE o ELIMINARLO sobre una colección, esto equivale a realizar lo siguiente internamente:

- OBTENER basado en consultas para recuperar la colección
- Secuencia en serie de LLAMADAS DE PARCHE o BORRADO en cada objeto de la colección

Puede establecer el tiempo de espera de la operación `return_timeout` con un valor predeterminado de 15 segundos. Si no se ha completado antes del tiempo de espera, la respuesta incluye un vínculo al siguiente objeto. Debe volver a emitir el mismo método HTTP mediante el siguiente enlace para continuar con la operación.

## Acceda a métricas de rendimiento con la API de REST DE ONTAP

ONTAP recopila métricas de rendimiento sobre los objetos y protocolos de almacenamiento de SVM seleccionados, e informa esta información a través de la API DE REST. Puede utilizar estos datos para supervisar el rendimiento de un sistema ONTAP.

Para un objeto o protocolo de almacenamiento determinado, los datos de rendimiento se clasifican en tres categorías:

- IOPS
- Latencia
- Rendimiento

Dentro de cada categoría, se encuentran disponibles uno o varios de los siguientes tipos de datos:

- Lectura ®
- Escritura (W)
- Otro (o)

- Total (T)

La tabla siguiente resume los datos de rendimiento disponibles a través de la API REST de ONTAP, incluida la versión cuando se agregó. Consulte la página de documentación en línea de la API DE REST en su sistema ONTAP para obtener más información.

<b>Objeto o protocolo de almacenamiento</b>	<b>IOPS</b>	<b>Latencia</b>	<b>Rendimiento</b>	<b>Versión de ONTAP</b>
Puerto Ethernet	No aplicable	No aplicable	TWT	9.8
Puerto FC	RUZOT	RUZOT	TWT	9.8
Interfaz de IP	No aplicable	No aplicable	TWT	9.8
Interfaz de FC	RUZOT	RUZOT	TWT	9.8
Espacio de nombres NVMe	RUZOT	RUZOT	RUZOT	9.8
Estadísticas de Qtree	RUZOT crudos	No aplicable	RUZOT crudos	9.8
FlexCache para volúmenes	RUZOT	RUZOT	TWT	9.8
Nodo: Utilización de procesos	Utilización del proceso como valor numérico	Utilización del proceso como valor numérico	Utilización del proceso como valor numérico	9.8
Volumen de cloud	RUZOT	RUZOT	No se puede aplicar	9.7
LUN	RUZOT	RUZOT	RUZOT	9.7
Agregado	RUZOT	RUZOT	RUZOT	9.7
Protocolo NFS SVM	RUZOT	RUZOT	TWT	9.7
Protocolo CIFS SVM	RUZOT	RUZOT	TWT	9.7
Protocolo FCP de SVM	RUZOT	RUZOT	TWT	9.7
Protocolo iSCSI de SVM	RUZOT	RUZOT	TWT	9.7
Protocolo NVMe de SVM	RUZOT	RUZOT	TWT	9.7
Clúster	RUZOT	RUZOT	RUZOT	9.6
Volúmenes	RUZOT	RUZOT	RUZOT	9.6

## **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.