



# **SnapMirror**

## REST API reference

NetApp  
September 12, 2025

# Table of Contents

|  |     |
|--|-----|
| SnapMirror .....   | 1   |
| SnapMirror overview .....  | 1   |
| Overview .....   | 1   |
| Manage SnapMirror policies .....   | 1   |
| SnapMirror policies endpoint overview .....                                  | 1   |
| Retrieve SnapMirror async and sync policy types .....                        | 2   |
| Create a SnapMirror policy .....   | 16  |
| Delete a SnapMirror policy .....   | 31  |
| Retrieve a specific SnapMirror policy .....                                  | 36  |
| Update the SnapMirror policy .....   | 45  |
| Manage SnapMirror relationships .....  | 58  |
| SnapMirror relationships endpoint overview .....                             | 58  |
| Retrieve information for SnapMirror relationships .....                      | 59  |
| Create a SnapMirror relationship .....                                       | 91  |
| Delete a SnapMirror relationship .....                                       | 128 |
| Retrieve a SnapMirror relationship .....                                     | 134 |
| Update a SnapMirror relationship .....                                       | 158 |
| Manage SnapMirror relationship transfers .....                               | 194 |
| SnapMirror relationships relationship.uuid transfers endpoint overview ..... | 194 |
| Retrieve ongoing SnapMirror transfers for a relationship .....               | 194 |
| Start a SnapMirror transfer operation .....                                  | 207 |
| Retrieve attributes of an ongoing SnapMirror transfer .....                  | 218 |
| Cancel an ongoing SnapMirror transfer .....                                  | 226 |

# SnapMirror

## SnapMirror overview

### Overview

SnapMirror technology enables customers to copy and backup their production data. The secondary or destination volume in such a backup relationship can reside anywhere, locally or remotely, and can be used to restore access to the protected data. You can restore access to protected data by drawing upon a vault of backups or, in the event that a primary is unusable, by using a disaster recovery copy that can be activated for clients and applications. SnapMirror periodically updates a replica to create new backups and/or to keep a replica up-to-date with changes that have been written to the primary. The SnapMirror subsystems are designed to keep many pairs of source (primary) and destination (secondary) copies up-to-date in an efficient and scalable manner.

The SnapMirror APIs can be used to create and manage SnapMirror relationships of type "async", and "sync". These APIs can also be used to manage restore-relationships. These APIs allow you to manage the following endpoints:

- SnapMirror policies - When applied to a SnapMirror relationship, the SnapMirror policy controls the behavior of the relationship and specifies the configuration attributes for that relationship.
- SnapMirror relationships - You can create and manage SnapMirror relationships, and you can change the state of the SnapMirror relationship using a PATCH request.
- SnapMirror transfers - You can manage data transfers on the specified SnapMirror relationship.

## Manage SnapMirror policies

### SnapMirror policies endpoint overview

#### Managing SnapMirror policies

This API is used to manage SnapMirror policies of type "async" and "sync". When applied to a SnapMirror relationship, the SnapMirror policy controls the behavior of the relationship and specifies the configuration attributes for that relationship. The policy type "async" can be associated with a SnapMirror relationship that has either a FlexVol volume, FlexGroup volume, or SVM as the endpoint. The policy type "sync" can be associated with a SnapMirror relationship that has a FlexVol volume or a consistency group as the endpoint. The policy type "sync" can have a "sync\_type" of either "sync", "strict\_sync", "automated\_failover" or "automated\_failover\_duplex". If the "sync\_type" is "sync", a write success is returned to the client after writing the data to the primary endpoint and before writing the data to the secondary endpoint. If the "sync\_type" is "strict\_sync", a write success is returned to the client after writing the data to both primary and secondary endpoints. A "sync\_type" of "automated\_failover" can be associated with a SnapMirror relationship that has a consistency group as the endpoint and provides asymmetric active active access to the two storage copies.

A "sync\_type" of "automated\_failover\_duplex" can be associated with a SnapMirror relationship that has a consistency group as the endpoint and provides symmetric active active access to the two storage copies.

Mapping of SnapMirror policies from CLI to REST

| CLI          | REST  |
|--------------|-------|
| mirror-vault | async |

| CLI                  | REST                        |
|----------------------|-----------------------------|
| async-mirror w/      | async w/                    |
| all_source_snapshots | copy_all_source_snapshots   |
| async-mirror w/o     | async w/                    |
| all_source_snapshots | copy_latest_source_snapshot |
| vault                | async w/                    |
|                      | create_snapshot_on_source   |

| CLI                       | REST | sync_type                 |
|---------------------------|------|---------------------------|
| sync-mirror               | sync | sync                      |
| strict-sync-mirror        | sync | strict_sync               |
| automated-failover        | sync | automated_failover        |
| automated-failover-duplex | sync | automated_failover_duplex |

## Retrieve SnapMirror async and sync policy types

GET /snapmirror/policies

**Introduced In:** 9.6

Retrieves SnapMirror policies of type "async" and "sync".

### Related ONTAP commands

- snapmirror policy show

### Example

The following example shows how to retrieve a collection of SnapMirror policies.

```
GET "/api/snapmirror/policies"
```

### Learn more

- [DOC /snapmirror/policies](#)

### Parameters

| Name                        | Type    | In    | Required | Description   |
|-----------------------------|---------|-------|----------|---|
| copy_all_source_snapshots   | boolean | query | False    | Filter by copy_all_source_snapshots <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> </ul>   |
| uuid                        | string  | query | False    | Filter by uuid  |
| transfer_schedule.name      | string  | query | False    | Filter by transfer_schedule.name  |
| transfer_schedule.uid       | string  | query | False    | Filter by transfer_schedule.uid   |
| copy_latest_source_snapshot | boolean | query | False    | Filter by copy_latest_source_snapshot <ul style="list-style-type: none"> <li>• Introduced in: 9.11</li> </ul> |
| network_compression_enabled | boolean | query | False    | Filter by network_compression_enabled   |
| svm.uuid                    | string  | query | False    | Filter by svm.uuid  |
| svm.name                    | string  | query | False    | Filter by svm.name  |
| comment                     | string  | query | False    | Filter by comment   |
| scope                       | string  | query | False    | Filter by scope   |
| name                        | string  | query | False    | Filter by name  |
| retention.count             | integer | query | False    | Filter by retention.count   |
| retention.preserve          | boolean | query | False    | Filter by retention.preserve <ul style="list-style-type: none"> <li>• Introduced in: 9.13</li> </ul>          |

| Name                               | Type    | In    | Required | Description  |
|------------------------------------|---------|-------|----------|--|
| retention.label                    | string  | query | False    | Filter by retention.label  |
| retention.creation_schedule.name   | string  | query | False    | Filter by retention.creation_schedule.name   |
| retention.creation_schedule.uuid   | string  | query | False    | Filter by retention.creation_schedule.uuid   |
| retention.warn                     | integer | query | False    | Filter by retention.warn <ul style="list-style-type: none"> <li>• Introduced in: 9.13</li> </ul>   |
| retention.period                   | string  | query | False    | Filter by retention.period <ul style="list-style-type: none"> <li>• Introduced in: 9.11</li> </ul> |
| retention.prefix                   | string  | query | False    | Filter by retention.prefix   |
| sync_common_snapshot_schedule.name | string  | query | False    | Filter by sync_common_snapshot_schedule.name   |
| sync_common_snapshot_schedule.uuid | string  | query | False    | Filter by sync_common_snapshot_schedule.uuid   |
| identity_preservation              | string  | query | False    | Filter by identity_preservation  |
| throttle                           | integer | query | False    | Filter by throttle   |
| sync_type                          | string  | query | False    | Filter by sync_type  |
| type                               | string  | query | False    | Filter by type   |
| rpo                                | integer | query | False    | Filter by rpo <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> </ul>              |

| Name                      | Type          | In    | Required | Description  |
|---------------------------|---------------|-------|----------|--|
| create_snapshot_on_source | boolean       | query | False    | <p>Filter by create_snapshot_on_source</p> <ul style="list-style-type: none"> <li>Introduced in: 9.11</li> </ul>   |
| fields                    | array[string] | query | False    | Specify the fields to return.  |
| max_records               | integer       | query | False    | Limit the number of records returned.  |
| return_records            | boolean       | query | False    | <p>The default is true for GET calls. When set to false, only the number of records is returned.</p> <ul style="list-style-type: none"> <li>Default value: 1</li> </ul>  |
| return_timeout            | integer       | query | False    | <p>The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached.</p> <ul style="list-style-type: none"> <li>Default value: 1</li> <li>Max value: 120</li> <li>Min value: 0</li> </ul> |
| order_by                  | array[string] | query | False    | Order results by specified fields and optional [asc]   |

## Response

Status: 200, Ok

| Name        | Type                                       | Description       |
|-------------|--|-------------------|
| _links      | <a href="#">_links</a>                     |                   |
| num_records | integer                                    | Number of records |
| records     | array[ <a href="#">snapmirror_policy</a> ] |                   |

## Example response

```
{  
  "_links": {  
    "next": {  
      "href": "/api/resourcelink"  
    },  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "num_records": 1,  
  "records": [  
    {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "comment": "string",  
      "copy_all_source_snapshots": 1,  
      "copy_latest_source_snapshot": 1,  
      "create_snapshot_on_source": "",  
      "identity_preservation": "string",  
      "name": "Asynchronous",  
      "retention": [  
        {  
          "count": 7,  
          "creation_schedule": {  
            "_links": {  
              "self": {  
                "href": "/api/resourcelink"  
              }  
            },  
            "name": "weekly",  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
          },  
          "label": "hourly",  
          "period": "P30D",  
          "prefix": "string",  
          "preserve": 1,  
          "warn": 4  
        }  
      ],  
      "rpo": 0,  
      "scope": "string",  
    }  
  ]  
}
```

```

"svm": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "svm1",
  "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
},
"sync_common_snapshot_schedule": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "weekly",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"sync_type": "string",
"throttle": 0,
"transfer_schedule": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "weekly",
  "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"type": "string",
"uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
}
]
}

```

## Error

Status: Default

## ONTAP Error Response codes

| Error code | Description                         |
|------------|-------------------------------------|
| 13303842   | SnapMirror policy is not supported. |

| Name  | Type           | Description |
|-------|----------------|-------------|
| error | returned_error |             |

## Example error

```
{  
  "error": {  
    "arguments": [  
      {  
        "code": "string",  
        "message": "string"  
      }  
    ],  
    "code": "4",  
    "message": "entry doesn't exist",  
    "target": "uuid"  
  }  
}
```

## Definitions

## See Definitions

href

| Name | Type   | Description |
|------|--------|-------------|
| href | string |             |

\_links

| Name | Type | Description |
|------|------|-------------|
| next | href |             |
| self | href |             |

\_links

| Name | Type | Description |
|------|------|-------------|
| self | href |             |

creation\_schedule

Schedule used to create Snapshot copies on the destination for long term retention. Only cron schedules are supported for SnapMirror.

| Name   | Type   | Description       |
|--------|--------|-------------------|
| _links | _links |                   |
| name   | string | Job schedule name |
| uuid   | string | Job schedule UUID |

snapmirror\_policy\_rule

SnapMirror policy rule for retention.

| Name              | Type              | Description   |
|-------------------|-------------------|---|
| count             | integer           | Number of Snapshot copies to be kept for retention. Maximum value will differ based on type of relationship and scaling factor.       |
| creation_schedule | creation_schedule | Schedule used to create Snapshot copies on the destination for long term retention. Only cron schedules are supported for SnapMirror. |

| Name     | Type    | Description   |
|----------|---------|---|
| label    | string  | Snapshot copy label   |
| period   | string  | Specifies the duration for which the Snapshot copies are locked. The retention period value represents a duration and must be in the ISO-8601 duration format. Years, months, days, hours, minutes, and seconds are represented as "P<num>Y", "P<num>M", "P<num>D", "PT<num>H", "PT<num>M" and "PT<num>S". Value "infinite" is also a valid input for Flexvol volumes and FlexGroup volumes. A duration which combines different periods is not supported, for example "P1Y10M" is not supported. The range of supported retention period values is between 1 second to infinite.</num></num></num></num></num></num> |
| prefix   | string  | Specifies the prefix for the Snapshot copy name to be created as per the schedule. If no value is specified, then the label is used as the prefix.  |
| preserve | boolean | Specifies the behavior when the Snapshot copy retention count is reached on the SnapMirror destination for the rule. The default value is false, which means that the oldest Snapshot copy is deleted to make room for new ones but only if the number of Snapshot copies has exceeded the retention count specified in the 'count' property. When set to true and where the Snapshot copies have reached the retention count, an incremental SnapMirror transfer will fail or if the rule has a schedule, Snapshot copies will no longer be created on the SnapMirror destination.                                   |

| Name | Type    | Description   |
|------|---------|---|
| warn | integer | Specifies the warning threshold count for the rule. The default value is zero. When set to a value greater than zero, an event is generated after the number of Snapshot copies (for the particular rule) retained on a SnapMirror destination reaches the specified warning limit. The preserve property for the rule must be true in order to set the warn property to a value greater than zero. |

## svm

SVM, applies only to SVM-scoped objects.

| Name   | Type                   | Description   |
|--------|------------------------|---|
| _links | <a href="#">_links</a> |   |
| name   | string                 | The name of the SVM. This field cannot be specified in a PATCH method.              |
| uuid   | string                 | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

## sync\_common\_snapshot\_schedule

Schedule used to create common Snapshot copies for synchronous relationships.

| Name   | Type                   | Description       |
|--------|------------------------|-------------------|
| _links | <a href="#">_links</a> |                   |
| name   | string                 | Job schedule name |
| uuid   | string                 | Job schedule UUID |

## transfer\_schedule

The schedule used to update asynchronous relationships. Only cron schedules are supported for SnapMirror.

| Name   | Type                   | Description |
|--------|------------------------|-------------|
| _links | <a href="#">_links</a> |             |

| Name | Type   | Description       |
|------|--------|-------------------|
| name | string | Job schedule name |
| uuid | string | Job schedule UUID |

### snapmirror\_policy

SnapMirror policy information. SnapMirror policy can either be of type "async", "sync" or "continuous". The policy type "async" can be associated with a SnapMirror relationship that has either a FlexVol volume, FlexGroup volume or SVM as the endpoint. The policy type "sync" along with "sync\_type" as "sync" or "strict\_sync" can be associated with a SnapMirror relationship that has a FlexVol volume as the endpoint. The policy type "sync" can have a "sync\_type" of either "sync", "strict\_sync", "automated\_failover" or "automated\_failover\_duplex". If the "sync\_type" is "sync", a write success is returned to the client after writing the data to the source endpoint and before writing the data to the destination endpoint. If the "sync\_type" is "strict\_sync", a write success is returned to the client after writing the data to both source and destination endpoints. If the "sync\_type" is "automated\_failover", the policy can be associated with a SnapMirror relationship that has a consistency group as the endpoint and provides asymmetric active active access to the two storage copies. If the "sync\_type" is "automated\_failover\_duplex", the policy can be associated with a SnapMirror relationship that has a consistency group as the endpoint and provides symmetric active active access to the two storage copies. Use the "sync" policy with "sync\_type" as "automated\_failover" or "automated\_failover\_duplex" to establish SnapMirror relationships for business continuity use cases. SnapMirror relationships with policy types as "sync" and "sync\_type" as "automated\_failover" or "automated\_failover\_duplex" can be monitored by the Mediator, if configured. If the source Consistency Group endpoint is not reachable, the Mediator might trigger a failover to the destination consistency group endpoint. A policy type of "continuous" can be associated with SnapMirror relationships that have either ONTAP S3 buckets or non-ONTAP object stores as endpoints. This type of policy is used for FabricLink owned targets.

| Name                      | Type                   | Description   |
|---------------------------|------------------------|---|
| _links                    | <a href="#">_links</a> |   |
| comment                   | string                 | Comment associated with the policy.   |
| copy_all_source_snapshots | boolean                | Specifies that all the source Snapshot copies (including the one created by SnapMirror before the transfer begins) should be copied to the destination on a transfer. "Retention" properties cannot be specified along with this property. This is applicable only to async policies. Property can only be set to 'true'. |

| Name                        | Type                          | Description  |
|-----------------------------|-------------------------------|--|
| copy_latest_source_snapshot | boolean                       | Specifies that the latest source Snapshot copy (created by SnapMirror before the transfer begins) should be copied to the destination on a transfer. "Retention" properties cannot be specified along with this property. This is applicable only to async policies. Property can only be set to 'true'. |
| create_snapshot_on_source   | boolean                       | Specifies whether a new Snapshot copy should be created on the source at the beginning of an update or resync operation. This is applicable only to async policies. Property can only be set to 'false'.   |
| identity_preservation       | string                        | Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type.   |
| name                        | string                        | Name of the policy.  |
| network_compression_enabled | boolean                       | Specifies whether network compression is enabled for transfers. This is applicable only to the policies of type "async".   |
| retention                   | array[snapmirror_policy_rule] | Rules for Snapshot copy retention.   |
| rpo                         | integer                       | Specifies the duration of time for which a change to be propagated to a mirror should be delayed, in seconds. This is an intentional propagation delay between mirrors and is configurable down to zero, which means an immediate propagation. This is supported for policies of type 'continuous'.      |

| Name                          | Type  | Description   |
|-------------------------------|---|---|
| scope                         | string  | Set to "svm" for policies owned by an SVM, otherwise set to "cluster".                                    |
| svm                           | svm   | SVM, applies only to SVM-scoped objects.  |
| sync_common_snapshot_schedule | <a href="#">sync_common_snapshot_schedule</a> | Schedule used to create common Snapshot copies for synchronous relationships.                             |
| sync_type                     | string  |   |
| throttle                      | integer                                       | Throttle in KB/s. Default to unlimited.   |
| transfer_schedule             | <a href="#">transfer_schedule</a>             | The schedule used to update asynchronous relationships. Only cron schedules are supported for SnapMirror. |
| type                          | string  |   |
| uuid                          | string  | Unique identifier of the SnapMirror policy.   |

#### error\_arguments

| Name    | Type   | Description      |
|---------|--------|------------------|
| code    | string | Argument code    |
| message | string | Message argument |

#### returned\_error

| Name      | Type                                   | Description                                 |
|-----------|--|---|
| arguments | <a href="#">array[error_arguments]</a> | Message arguments                           |
| code      | string                                 | Error code                                  |
| message   | string                                 | Error message                               |
| target    | string                                 | The target parameter that caused the error. |

## Create a SnapMirror policy

POST /snapmirror/policies

**Introduced In:** 9.6

Creates a SnapMirror policy. The property "identity\_preservation" is applicable to only SnapMirror relationships with SVM endpoints and it indicates which configuration of the source SVM is replicated to the destination SVM.

It takes the following values:

- `full` - indicates that the source SVM configuration is replicated to the destination SVM endpoint.
- `exclude_network_config` - indicates that the source SVM configuration other than network configuration is replicated to the destination SVM endpoint.
- `exclude_network_and_protocol_config` - indicates that the source SVM configuration is not replicated to the destination SVM endpoint.

### Important notes

- The property "identity\_preservation" is applicable to only SnapMirror relationships with SVM endpoints and it indicates which configuration of the source SVM is replicated to the destination SVM.
- The properties "identity\_preservation" and "transfer\_schedule" are not applicable for "sync" type policies.
- The properties "copy\_all\_source\_snapshots", "copy\_latest\_source\_snapshot", and "create\_snapshot\_on\_source" are mutually exclusive.
- The properties "copy\_all\_source\_snapshots", "copy\_latest\_source\_snapshot", and "create\_snapshot\_on\_source" are not applicable for "sync" type policies.
- No "retention" properties can be specified if "copy\_all\_source\_snapshots" or 'copy\_latest\_source\_snapshot' is specified.
- The properties "retention.creation\_schedule" and "retention.prefix" are not applicable for "sync" type policies.
- The property "retention.creation\_schedule" is not applicable for "async" policies with "create\_snapshot\_on\_source" set to "false".
- The property "sync\_common\_snapshot\_schedule" is not applicable for an "async" type policy.
- The property "retention.count" specifies the maximum number of Snapshot copies that are retained on the SnapMirror destination volume.
- When the property "retention.label" is specified, the Snapshot copies that have a SnapMirror label matching this property is transferred to the SnapMirror destination.
- When the property "retention.creation\_schedule" is specified, Snapshot copies are directly created on the SnapMirror destination. The Snapshot copies created have the same content as the latest Snapshot copy already present on the SnapMirror destination.
- The property "transfer\_schedule" cannot be set to null (no-quotes) during SnapMirror policy POST.
- The properties "retention.label" and "retention.count" must be specified for "async" policies with "create\_snapshot\_on\_source" set to "false".
- The property "retention.warn" is not supported for a policy when the "retention.preserve" property is false.
- The property "retention.warn" value must be less than the property "retention.count" value for a rule in a policy.

## Required properties

- `name` - Name of the new SnapMirror policy.

## Recommended optional properties

- `svm.name` or `svm.uuid` - Name or UUID of the SVM that owns the SnapMirror policy.

## Default property values

If not specified in POST, the following default property values are assigned:

- `type` - `async`
- `sync_type` - `sync` (when `type` is `sync`)
- `network_compression_enabled` - `false`
- `throttle` - `0`
- `identity_preservation` - `exclude_network_and_protocol_config`

## Related ONTAP commands

- `snapmirror policy create`

## Examples

Creating a SnapMirror policy of type "sync"

```
POST "/api/snapmirror/policies/" '{"name": "policy1", "svm.name": "VSO",  
"type": "sync", "sync_type": "sync"}'
```

Creating a SnapMirror policy of type "async" with two sets of retention values, one with a `creation_schedule`

```
POST "/api/snapmirror/policies" '{"name": "policy_ret", "svm": {"name":  
"vs1"}, "retention": [{"label": "weekly", "count": "2",  
"creation_schedule": {"name": "weekly"}}, {"label": "daily",  
"count": "7"}]}'
```

Creating a SnapMirror policy of type "async"

```
POST "/api/snapmirror/policies" '{"name": "newPolicy", "svm": {"name":  
"vs1"}, "type": "async"}'
```

Creating a SnapMirror policy of type "async" which replicates all Snapshot copies

```
POST "/api/snapmirror/policies" '{"name": "newPolicy", "svm": {"name" : "vs1"}, "type": "async", "copy_all_source_snapshots": "true"}'
```

Creating a SnapMirror policy of type "async" which replicates latest Snapshot copy

```
POST "/api/snapmirror/policies" '{"name": "newPolicy2", "svm": {"name" : "vs1"}, "type": "async", "copy_latest_source_snapshot": "true"}'
```

Creating a SnapMirror policy of type "async" which does not create Snapshot copies on source

```
POST "/api/snapmirror/policies" '{"name": "newPolicy", "svm": {"name" : "vs1"}, "type": "async", "create_snapshot_on_source": "false", "retention": [{"label": "daily", "count": 7}]}'
```

Creating a SnapMirror policy of type "sync" with sync\_type as "automated\_failover"

```
POST "/api/snapmirror/policies/" '{"name": "policy1", "svm.name": "VS0", "type": "sync", "sync_type": "automated_failover"}'
```

Creating a SnapMirror policy of type "sync" with sync\_type as "automated\_failover\_duplex"

```
POST "/api/snapmirror/policies/" '{"name": "policy_afd", "svm.name": "VS0", "type": "sync", "sync_type": "automated_failover_duplex"}'
```

Creating a SnapMirror policy of type "async" with two sets of retention values and retention periods

```
POST "/api/snapmirror/policies" '{"name": "policy_ret", "svm": {"name": "vs1"}, "retention": [{"label": "weekly", "count": "2", "period": "P7D"}, {"label": "daily", "count": "7", "period": "PT3H"}]}'
```

Creating a SnapMirror policy of type "async" with retention value as "infinite"

```
POST "/api/snapmirror/policies" '{"name": "policy_ret", "svm": {"name": "vs1"}, "retention": [{"label": "weekly", "count": "5", "period": "infinite"}]}'
```

Creating a SnapMirror policy of type "async" with properties retention preserve as true and retention warn as 3.

```
POST "/api/snapmirror/policies" '{"name": "policy_ret", "svm": {"name": "vs1"}, "retention": [{"label": "weekly", "count": "5", "preserve": true, "warn": 3}]}'
```

## Learn more

- [DOC /snapmirror/policies](#)

## Parameters

| Name           | Type    | In    | Required | Description   |
|----------------|---------|-------|----------|---|
| return_records | boolean | query | False    | <p>The default is false. If set to true, the records are returned.</p> <ul style="list-style-type: none"><li>• Default value:</li></ul> |

| Name           | Type    | In    | Required | Description  |
|----------------|---------|-------|----------|--|
| return_timeout | integer | query | False    | <p>The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul> |

## Request Body

| Name    | Type   | Description                         |
|---------|--------|-------------------------------------|
| comment | string | Comment associated with the policy. |

| Name                        | Type  | Description   |
|-----------------------------|---|---|
| copy_all_source_snapshots   | boolean   | Specifies that all the source Snapshot copies (including the one created by SnapMirror before the transfer begins) should be copied to the destination on a transfer. "Retention" properties cannot be specified along with this property. This is applicable only to async policies. Property can only be set to 'true'. |
| copy_latest_source_snapshot | boolean   | Specifies that the latest source Snapshot copy (created by SnapMirror before the transfer begins) should be copied to the destination on a transfer. "Retention" properties cannot be specified along with this property. This is applicable only to async policies. Property can only be set to 'true'.                  |
| create_snapshot_on_source   | boolean   | Specifies whether a new Snapshot copy should be created on the source at the beginning of an update or resync operation. This is applicable only to async policies. Property can only be set to 'false'.  |
| identity_preservation       | string  | Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type.  |
| name                        | string  | Name of the policy.   |
| network_compression_enabled | boolean   | Specifies whether network compression is enabled for transfers. This is applicable only to the policies of type "async".  |
| retention                   | array[ <a href="#">snapmirror_policy_rule</a> ] | Rules for Snapshot copy retention.  |

| Name                          | Type  | Description   |
|-------------------------------|---|---|
| rpo                           | integer                                       | Specifies the duration of time for which a change to be propagated to a mirror should be delayed, in seconds. This is an intentional propagation delay between mirrors and is configurable down to zero, which means an immediate propagation. This is supported for policies of type 'continuous'. |
| scope                         | string  | Set to "svm" for policies owned by an SVM, otherwise set to "cluster".  |
| svm                           | svm   | SVM, applies only to SVM-scoped objects.  |
| sync_common_snapshot_schedule | <a href="#">sync_common_snapshot_schedule</a> | Schedule used to create common Snapshot copies for synchronous relationships.   |
| sync_type                     | string  |   |
| throttle                      | integer                                       | Throttle in KB/s. Default to unlimited.   |
| transfer_schedule             | <a href="#">transfer_schedule</a>             | The schedule used to update asynchronous relationships. Only cron schedules are supported for SnapMirror.   |
| type                          | string  |   |
| uuid                          | string  | Unique identifier of the SnapMirror policy.   |

## Example request

```
{  
    "comment": "string",  
    "copy_all_source_snapshots": 1,  
    "copy_latest_source_snapshot": 1,  
    "create_snapshot_on_source": "",  
    "identity_preservation": "string",  
    "name": "Asynchronous",  
    "retention": [  
        {  
            "count": 7,  
            "creation_schedule": {  
                "name": "weekly",  
                "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
            },  
            "label": "hourly",  
            "period": "P30D",  
            "prefix": "string",  
            "preserve": 1,  
            "warn": 4  
        }  
    ],  
    "rpo": 0,  
    "scope": "string",  
    "svm": {  
        "name": "svml",  
        "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
    },  
    "sync_common_snapshot_schedule": {  
        "name": "weekly",  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "sync_type": "string",  
    "throttle": 0,  
    "transfer_schedule": {  
        "name": "weekly",  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "type": "string",  
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"  
}
```

## Response

Status: 202, Accepted

| Name | Type     | Description |
|------|----------|-------------|
| job  | job_link |             |

## Example response

```
{  
  "job": {  
    "uuid": "string"  
  }  
}
```

## Headers

| Name     | Description                               | Type   |
|----------|---|--------|
| Location | Useful for tracking the resource location | string |

## Response

Status: 201, Created

## Error

Status: Default

## ONTAP Error Response codes

| Error code   | Description   |
|--|---|
| -----  | -----   |
| -----  | -----   |
| 6619714  | Schedule specified is an interval schedule. SnapMirror does not support interval schedules. |
| 13304126   |   |
| Enter a value for "count" in the range indicated in the error message. |   |

| Error code   | Description   |
|--|---|
| 13303850   | Invalid input parameter   |
| 13303887   |   |
| Failed to create SnapMirror policy. Reason: Maximum number of allowed retention rules reached  |   |
| 13304083   | The specified property is not supported because all nodes in the cluster are not capable of supporting this property. |
| 13304084   |   |
| Properties specified are mutually exclusive. Provide only one property.  |   |
| 13304085   | The specified property does not support the specified value.  |
| 13304092   |   |
| Input value of the retention period property is invalid. For relationships with FlexVol volume or FlexGroup volume destinations, the duration must be in ISO 6801 format or can be infinite. For relationships with object store destinations, only duration values with Y, M or D and supported and must be in the specified range. |   |
| 6621045  | The property retention.warn is not supported for a policy when the retention.preserve property is false.              |
| 13304129   | The property retention.warn value must be less than the property retention.count value for a rule in a policy.        |
| 13304130   | The total retention.count value for all rules in a policy cannot exceed the value indicated in the error message."    |

## Definitions

## See Definitions

href

| Name | Type   | Description |
|------|--------|-------------|
| href | string |             |

\_links

creation\_schedule

Schedule used to create Snapshot copies on the destination for long term retention. Only cron schedules are supported for SnapMirror.

| Name | Type   | Description       |
|------|--------|-------------------|
| name | string | Job schedule name |
| uuid | string | Job schedule UUID |

snapmirror\_policy\_rule

SnapMirror policy rule for retention.

| Name              | Type              | Description   |
|-------------------|-------------------|---|
| count             | integer           | Number of Snapshot copies to be kept for retention. Maximum value will differ based on type of relationship and scaling factor.       |
| creation_schedule | creation_schedule | Schedule used to create Snapshot copies on the destination for long term retention. Only cron schedules are supported for SnapMirror. |
| label             | string            | Snapshot copy label   |

| Name     | Type    | Description  |
|----------|---------|--|
| period   | string  | <p>Specifies the duration for which the Snapshot copies are locked. The retention period value represents a duration and must be in the ISO-8601 duration format. Years, months, days, hours, minutes, and seconds are represented as "P&lt;num&gt;Y", "P&lt;num&gt;M", "P&lt;num&gt;D", "PT&lt;num&gt;H", "PT&lt;num&gt;M" and "PT&lt;num&gt;S". Value "infinite" is also a valid input for Flexvol volumes and FlexGroup volumes. A duration which combines different periods is not supported, for example "P1Y10M" is not supported. The range of supported retention period values is between 1 second to infinite.</p> |
| prefix   | string  | <p>Specifies the prefix for the Snapshot copy name to be created as per the schedule. If no value is specified, then the label is used as the prefix.</p>  |
| preserve | boolean | <p>Specifies the behavior when the Snapshot copy retention count is reached on the SnapMirror destination for the rule. The default value is false, which means that the oldest Snapshot copy is deleted to make room for new ones but only if the number of Snapshot copies has exceeded the retention count specified in the 'count' property. When set to true and where the Snapshot copies have reached the retention count, an incremental SnapMirror transfer will fail or if the rule has a schedule, Snapshot copies will no longer be created on the SnapMirror destination.</p>                                   |

| Name | Type    | Description   |
|------|---------|---|
| warn | integer | Specifies the warning threshold count for the rule. The default value is zero. When set to a value greater than zero, an event is generated after the number of Snapshot copies (for the particular rule) retained on a SnapMirror destination reaches the specified warning limit. The preserve property for the rule must be true in order to set the warn property to a value greater than zero. |

## svm

SVM, applies only to SVM-scoped objects.

| Name | Type   | Description   |
|------|--------|---|
| name | string | The name of the SVM. This field cannot be specified in a PATCH method.              |
| uuid | string | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

## sync\_common\_snapshot\_schedule

Schedule used to create common Snapshot copies for synchronous relationships.

| Name | Type   | Description       |
|------|--------|-------------------|
| name | string | Job schedule name |
| uuid | string | Job schedule UUID |

## transfer\_schedule

The schedule used to update asynchronous relationships. Only cron schedules are supported for SnapMirror.

| Name | Type   | Description       |
|------|--------|-------------------|
| name | string | Job schedule name |
| uuid | string | Job schedule UUID |

## .snapmirror\_policy

SnapMirror policy information. SnapMirror policy can either be of type "async", "sync" or "continuous". The policy type "async" can be associated with a SnapMirror relationship that has either a FlexVol volume, FlexGroup volume or SVM as the endpoint. The policy type "sync" along with "sync\_type" as "sync" or "strict\_sync" can be associated with a SnapMirror relationship that has a FlexVol volume as the endpoint. The policy type "sync" can have a "sync\_type" of either "sync", "strict\_sync", "automated\_failover" or "automated\_failover\_duplex". If the "sync\_type" is "sync", a write success is returned to the client after writing the data to the source endpoint and before writing the data to the destination endpoint. If the "sync\_type" is "strict\_sync", a write success is returned to the client after writing the data to both source and destination endpoints. If the "sync\_type" is "automated\_failover", the policy can be associated with a SnapMirror relationship that has a consistency group as the endpoint and provides asymmetric active active access to the two storage copies. If the "sync\_type" is "automated\_failover\_duplex", the policy can be associated with a SnapMirror relationship that has a consistency group as the endpoint and provides symmetric active active access to the two storage copies. Use the "sync" policy with "sync\_type" as "automated\_failover" or "automated\_failover\_duplex" to establish SnapMirror relationships for business continuity use cases. SnapMirror relationships with policy types as "sync" and "sync\_type" as "automated\_failover" or "automated\_failover\_duplex" can be monitored by the Mediator, if configured. If the source Consistency Group endpoint is not reachable, the Mediator might trigger a failover to the destination consistency group endpoint. A policy type of "continuous" can be associated with SnapMirror relationships that have either ONTAP S3 buckets or non-ONTAP object stores as endpoints. This type of policy is used for FabricLink owned targets.

| Name                        | Type    | Description   |
|-----------------------------|---------|---|
| comment                     | string  | Comment associated with the policy.   |
| copy_all_source_snapshots   | boolean | Specifies that all the source Snapshot copies (including the one created by SnapMirror before the transfer begins) should be copied to the destination on a transfer. "Retention" properties cannot be specified along with this property. This is applicable only to async policies. Property can only be set to 'true'. |
| copy_latest_source_snapshot | boolean | Specifies that the latest source Snapshot copy (created by SnapMirror before the transfer begins) should be copied to the destination on a transfer. "Retention" properties cannot be specified along with this property. This is applicable only to async policies. Property can only be set to 'true'.                  |

| Name                          | Type                          | Description   |
|-------------------------------|-------------------------------|---|
| create_snapshot_on_source     | boolean                       | Specifies whether a new Snapshot copy should be created on the source at the beginning of an update or resync operation. This is applicable only to async policies. Property can only be set to 'false'.  |
| identity_preservation         | string                        | Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type.  |
| name                          | string                        | Name of the policy.   |
| network_compression_enabled   | boolean                       | Specifies whether network compression is enabled for transfers. This is applicable only to the policies of type "async".  |
| retention                     | array[snapmirror_policy_rule] | Rules for Snapshot copy retention.  |
| rpo                           | integer                       | Specifies the duration of time for which a change to be propagated to a mirror should be delayed, in seconds. This is an intentional propagation delay between mirrors and is configurable down to zero, which means an immediate propagation. This is supported for policies of type 'continuous'. |
| scope                         | string                        | Set to "svm" for policies owned by an SVM, otherwise set to "cluster".  |
| svm                           | svm                           | SVM, applies only to SVM-scoped objects.  |
| sync_common_snapshot_schedule | sync_common_snapshot_schedule | Schedule used to create common Snapshot copies for synchronous relationships.   |
| sync_type                     | string                        |   |

| Name              | Type              | Description   |
|-------------------|-------------------|---|
| throttle          | integer           | Throttle in KB/s. Default to unlimited.   |
| transfer_schedule | transfer_schedule | The schedule used to update asynchronous relationships. Only cron schedules are supported for SnapMirror. |
| type              | string            |   |
| uuid              | string            | Unique identifier of the SnapMirror policy.   |

#### job\_link

| Name | Type   | Description   |
|------|--------|---|
| uuid | string | The UUID of the asynchronous job that is triggered by a POST, PATCH, or DELETE operation. |

#### error\_arguments

| Name    | Type   | Description      |
|---------|--------|------------------|
| code    | string | Argument code    |
| message | string | Message argument |

#### returned\_error

| Name      | Type                   | Description                                 |
|-----------|------------------------|---|
| arguments | array[error_arguments] | Message arguments                           |
| code      | string                 | Error code                                  |
| message   | string                 | Error message                               |
| target    | string                 | The target parameter that caused the error. |

## Delete a SnapMirror policy

DELETE /snapmirror/policies/{uuid}

**Introduced In:** 9.6

Deletes a SnapMirror policy.

## Related ONTAP commands

- `snapmirror policy delete`

## Example

```
DELETE "/api/snapmirror/policies/510c15d4-f9e6-11e8-bdb5-0050568e12c2"
```

## Learn more

- [DOC /snapmirror/policies](#)

## Parameters

| Name | Type   | In   | Required | Description            |
|------|--------|------|----------|------------------------|
| uuid | string | path | True     | SnapMirror policy UUID |

| Name           | Type    | In    | Required | Description  |
|----------------|---------|-------|----------|--|
| return_timeout | integer | query | False    | <p>The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul> |

## Response

Status: 200, Ok

| Name | Type                     | Description |
|------|--------------------------|-------------|
| job  | <a href="#">job_link</a> |             |

## Example response

```
{  
  "job": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "uuid": "string"  
  }  
}
```

## Response

Status: 202, Accepted

## Error

Status: Default, Error

| Name  | Type                           | Description |
|-------|--------------------------------|-------------|
| error | <a href="#">returned_error</a> |             |

## Example error

```
{  
  "error": {  
    "arguments": [  
      {  
        "code": "string",  
        "message": "string"  
      }  
    ],  
    "code": "4",  
    "message": "entry doesn't exist",  
    "target": "uuid"  
  }  
}
```

## Definitions

### See Definitions

href

| Name | Type   | Description |
|------|--------|-------------|
| href | string |             |

\_links

| Name | Type | Description |
|------|------|-------------|
| self | href |             |

job\_link

| Name   | Type   | Description   |
|--------|--------|---|
| _links | _links |   |
| uuid   | string | The UUID of the asynchronous job that is triggered by a POST, PATCH, or DELETE operation. |

error\_arguments

| Name    | Type   | Description      |
|---------|--------|------------------|
| code    | string | Argument code    |
| message | string | Message argument |

returned\_error

| Name      | Type                   | Description                                 |
|-----------|------------------------|---|
| arguments | array[error_arguments] | Message arguments                           |
| code      | string                 | Error code                                  |
| message   | string                 | Error message                               |
| target    | string                 | The target parameter that caused the error. |

## Retrieve a specific SnapMirror policy

GET /snapmirror/policies/{uuid}

**Introduced In:** 9.6

Retrieves a specific SnapMirror policy.

### Example

```
GET "/api/snapmirror/policies/567aaac0-f863-11e8-a666-0050568e12c2"
```

### Learn more

- [DOC /snapmirror/policies](#)

### Parameters

| Name   | Type          | In    | Required | Description                   |
|--------|---------------|-------|----------|-------------------------------|
| uuid   | string        | path  | True     | SnapMirror policy UUID.       |
| fields | array[string] | query | False    | Specify the fields to return. |

### Response

```
Status: 200, Ok
```

| Name                      | Type                   | Description   |
|---------------------------|------------------------|---|
| _links                    | <a href="#">_links</a> |   |
| comment                   | string                 | Comment associated with the policy.   |
| copy_all_source_snapshots | boolean                | Specifies that all the source Snapshot copies (including the one created by SnapMirror before the transfer begins) should be copied to the destination on a transfer. "Retention" properties cannot be specified along with this property. This is applicable only to async policies. Property can only be set to 'true'. |

| Name                        | Type  | Description  |
|-----------------------------|---|--|
| copy_latest_source_snapshot | boolean   | Specifies that the latest source Snapshot copy (created by SnapMirror before the transfer begins) should be copied to the destination on a transfer. "Retention" properties cannot be specified along with this property. This is applicable only to async policies. Property can only be set to 'true'. |
| create_snapshot_on_source   | boolean   | Specifies whether a new Snapshot copy should be created on the source at the beginning of an update or resync operation. This is applicable only to async policies. Property can only be set to 'false'.   |
| identity_preservation       | string  | Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type.   |
| name                        | string  | Name of the policy.  |
| network_compression_enabled | boolean   | Specifies whether network compression is enabled for transfers. This is applicable only to the policies of type "async".   |
| retention                   | array[ <a href="#">snapmirror_policy_rule</a> ] | Rules for Snapshot copy retention.   |
| rpo                         | integer   | Specifies the duration of time for which a change to be propagated to a mirror should be delayed, in seconds. This is an intentional propagation delay between mirrors and is configurable down to zero, which means an immediate propagation. This is supported for policies of type 'continuous'.      |
| scope                       | string  | Set to "svm" for policies owned by an SVM, otherwise set to "cluster".   |
| svm                         | <a href="#">svm</a>                             | SVM, applies only to SVM-scoped objects.   |

| Name                          | Type  | Description   |
|-------------------------------|---|---|
| sync_common_snapshot_schedule | <a href="#">sync_common_snapshot_schedule</a> | Schedule used to create common Snapshot copies for synchronous relationships.                             |
| sync_type                     | string  |   |
| throttle                      | integer                                       | Throttle in KB/s. Default to unlimited.   |
| transfer_schedule             | <a href="#">transfer_schedule</a>             | The schedule used to update asynchronous relationships. Only cron schedules are supported for SnapMirror. |
| type                          | string  |   |
| uuid                          | string  | Unique identifier of the SnapMirror policy.   |

## Example response

```
{  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "comment": "string",  
  "copy_all_source_snapshots": 1,  
  "copy_latest_source_snapshot": 1,  
  "create_snapshot_on_source": "",  
  "identity_preservation": "string",  
  "name": "Asynchronous",  
  "retention": [  
    {  
      "count": 7,  
      "creation_schedule": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "name": "weekly",  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
      },  
      "label": "hourly",  
      "period": "P30D",  
      "prefix": "string",  
      "preserve": 1,  
      "warn": 4  
    }  
  ],  
  "rpo": 0,  
  "scope": "string",  
  "svm": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "name": "svml1",  
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
  },  
  "sync_common_snapshot_schedule": {  
    "_links": {  
    }  
  }  
}
```

```

    "self": {
      "href": "/api/resourcelink"
    },
    "name": "weekly",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "sync_type": "string",
  "throttle": 0,
  "transfer_schedule": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "weekly",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "type": "string",
  "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
}

```

## Error

Status: Default

### ONTAP Error Response codes

| Error code | Description                         |
|------------|-------------------------------------|
| 13303842   | SnapMirror policy is not supported. |

| Name  | Type           | Description |
|-------|----------------|-------------|
| error | returned_error |             |

## Example error

```
{  
  "error": {  
    "arguments": [  
      {  
        "code": "string",  
        "message": "string"  
      }  
    ],  
    "code": "4",  
    "message": "entry doesn't exist",  
    "target": "uuid"  
  }  
}
```

## Definitions

## See Definitions

href

| Name | Type   | Description |
|------|--------|-------------|
| href | string |             |

\_links

| Name | Type | Description |
|------|------|-------------|
| self | href |             |

creation\_schedule

Schedule used to create Snapshot copies on the destination for long term retention. Only cron schedules are supported for SnapMirror.

| Name   | Type   | Description       |
|--------|--------|-------------------|
| _links | _links |                   |
| name   | string | Job schedule name |
| uuid   | string | Job schedule UUID |

snapmirror\_policy\_rule

SnapMirror policy rule for retention.

| Name              | Type              | Description   |
|-------------------|-------------------|---|
| count             | integer           | Number of Snapshot copies to be kept for retention. Maximum value will differ based on type of relationship and scaling factor.       |
| creation_schedule | creation_schedule | Schedule used to create Snapshot copies on the destination for long term retention. Only cron schedules are supported for SnapMirror. |
| label             | string            | Snapshot copy label   |

| Name     | Type    | Description  |
|----------|---------|--|
| period   | string  | <p>Specifies the duration for which the Snapshot copies are locked. The retention period value represents a duration and must be in the ISO-8601 duration format. Years, months, days, hours, minutes, and seconds are represented as "P&lt;num&gt;Y", "P&lt;num&gt;M", "P&lt;num&gt;D", "PT&lt;num&gt;H", "PT&lt;num&gt;M" and "PT&lt;num&gt;S". Value "infinite" is also a valid input for Flexvol volumes and FlexGroup volumes. A duration which combines different periods is not supported, for example "P1Y10M" is not supported. The range of supported retention period values is between 1 second to infinite.&lt;/num&gt;&lt;/num&gt;&lt;/num&gt;&lt;/num&gt;&lt;/num&gt;&lt;/num&gt;</p> |
| prefix   | string  | <p>Specifies the prefix for the Snapshot copy name to be created as per the schedule. If no value is specified, then the label is used as the prefix.</p>  |
| preserve | boolean | <p>Specifies the behavior when the Snapshot copy retention count is reached on the SnapMirror destination for the rule. The default value is false, which means that the oldest Snapshot copy is deleted to make room for new ones but only if the number of Snapshot copies has exceeded the retention count specified in the 'count' property. When set to true and where the Snapshot copies have reached the retention count, an incremental SnapMirror transfer will fail or if the rule has a schedule, Snapshot copies will no longer be created on the SnapMirror destination.</p>   |

| Name | Type    | Description   |
|------|---------|---|
| warn | integer | Specifies the warning threshold count for the rule. The default value is zero. When set to a value greater than zero, an event is generated after the number of Snapshot copies (for the particular rule) retained on a SnapMirror destination reaches the specified warning limit. The preserve property for the rule must be true in order to set the warn property to a value greater than zero. |

## svm

SVM, applies only to SVM-scoped objects.

| Name   | Type                   | Description   |
|--------|------------------------|---|
| _links | <a href="#">_links</a> |   |
| name   | string                 | The name of the SVM. This field cannot be specified in a PATCH method.              |
| uuid   | string                 | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

## sync\_common\_snapshot\_schedule

Schedule used to create common Snapshot copies for synchronous relationships.

| Name   | Type                   | Description       |
|--------|------------------------|-------------------|
| _links | <a href="#">_links</a> |                   |
| name   | string                 | Job schedule name |
| uuid   | string                 | Job schedule UUID |

## transfer\_schedule

The schedule used to update asynchronous relationships. Only cron schedules are supported for SnapMirror.

| Name   | Type                   | Description |
|--------|------------------------|-------------|
| _links | <a href="#">_links</a> |             |

| Name | Type   | Description       |
|------|--------|-------------------|
| name | string | Job schedule name |
| uuid | string | Job schedule UUID |

error\_arguments

| Name    | Type   | Description      |
|---------|--------|------------------|
| code    | string | Argument code    |
| message | string | Message argument |

returned\_error

| Name      | Type                   | Description                                 |
|-----------|------------------------|---|
| arguments | array[error_arguments] | Message arguments                           |
| code      | string                 | Error code                                  |
| message   | string                 | Error message                               |
| target    | string                 | The target parameter that caused the error. |

## Update the SnapMirror policy

PATCH /snapmirror/policies/{uuid}

**Introduced In:** 9.6

Updates the SnapMirror policy.

### Important notes

- The properties "retention.label" and "retention.count" are mandatory if "retention" is provided in the input. The provided "retention.label" is the final list and it replaces the existing values.
- The value of the "identity\_preservation" property cannot be changed if the SnapMirror relationships associated with the policy have different identity\_preservation configurations.
- If the SnapMirror policy "identity\_preservation" value matches the "identity\_preservation" value of the associated SnapMirror relationships, then the "identity\_preservation" value can be changed from a higher "identity\_preservation" threshold value to a lower "identity\_preservation" threshold value but not vice-versa. For example, the threshold value of the "identity\_preservation" property can be changed from "full" to "exclude\_network\_config", but cannot be increased from "exclude\_network\_and\_protocol\_config" to "exclude\_network\_config" to "full". The threshold value of the "identity\_preservation" cannot be changed to "exclude\_network\_and\_protocol\_config" for IDP SVMDR.

- The policy properties "copy\_all\_source\_snapshots", "copy\_latest\_source\_snapshot", and "create\_snapshot\_on\_source" cannot be modified.
- No "retention" properties can be modified if the "copy\_all\_source\_snapshots" or "copy\_latest\_source\_snapshot" property is present in the policy.
- Replacing or deleting all retention rules of a policy that has the "create\_snapshot\_on\_source" property set to false in a single API call is not supported.
- Modifying the property "retention.label" for all retention rules of a policy that has the "create\_snapshot\_on\_source" property set to false in a single API call is not supported.
- To remove a transfer\_schedule on a SnapMirror policy set the "transfer\_schedule" to null (no-quotes) during SnapMirror policy PATCH.

## Related ONTAP commands

- `snapmirror policy modify`

### Example

Updating the "retention" property to add rules to a policy without any rules.

```
PATCH "/api/snapmirror/policies/fe65686d-00dc-11e9-b5fb-0050568e3f83"
'{"retention": [{"label": "newlabel", "count": 2}, {"label": "weekly", "count": 2, "creation_schedule": {"name": "weekly"}}, {"label": "daily", "count": 14}]}'
```

Updating the "retention" property to add rules to a policy with existing rules {"retention": [{"label": "oldLabel1", "count": 2}, {"label": "oldLabel2", "count": 5}]}

```
PATCH "/api/snapmirror/policies/fe65686d-00dc-11e9-b5fb-0050568e3f83"
'{"retention": [{"label": "oldLabel1", "count": 2}, {"label": "oldLabel2", "count": 5}, {"label": "newlabel", "count": 3}, {"label": "weekly", "count": 1}]}'
```

Updating the "retention" property to remove a rule (oldLabel1) and add new rule to a policy with existing rules {"retention": [{"label": "oldLabel1", "count": 2}, {"label": "oldLabel2", "count": 3}]}

```
PATCH "/api/snapmirror/policies/fe65686d-00dc-11e9-b5fb-0050568e3f83"
'{"retention": [{"label": "oldLabel2", "count": 3}, {"label": "newlabel", "count": 2}]}'
```

Updating "transfer\_schedule", "throttle", and "identity\_preservation" properties

```
PATCH "/api/snapmirror/policies/8aef950b-3bef-11e9-80ac-0050568ea591"
'{"transfer_schedule.name" : "weekly", "throttle" : "100",
"identity_preservation":"exclude_network_and_protocol_config"}'
```

Removing the SnapMirror transfer\_schedule for a SnapMirror policy. Transfer\_schedule can be specified as UUID or name or both with the value set to null (no-quotes).

```
PATCH "/api/snapmirror/policies/98bb2608-fc60-11e8-aa13-005056a707ff"
'{"transfer_schedule": {"uuid": null, "name": null}}'
```

Updating the "retention" property to have retention.preserve and retention.warn for existing rule.

```
PATCH "/api/snapmirror/policies/fe65686d-00dc-11e9-b5fb-0050568e3f83"
'{"retention": [{"label": "oldLabel1", "count": 3, "preserve": true,
"warn": 2}]}'
```

## Learn more

- [DOC /snapmirror/policies](#)

## Parameters

| Name | Type   | In   | Required | Description            |
|------|--------|------|----------|------------------------|
| uuid | string | path | True     | SnapMirror policy UUID |

| Name           | Type    | In    | Required | Description  |
|----------------|---------|-------|----------|--|
| return_timeout | integer | query | False    | <p>The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul> |

## Request Body

| Name                  | Type   | Description  |
|-----------------------|--------|--|
| comment               | string | Comment associated with the policy.  |
| identity_preservation | string | Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type. |

| Name                          | Type  | Description   |
|-------------------------------|---|---|
| network_compression_enabled   | boolean   | Specifies whether network compression is enabled for transfers. This is applicable only to the policies of type "async".  |
| retention                     | array[ <a href="#">snapmirror_policy_rule</a> ] | Rules for Snapshot copy retention.  |
| rpo                           | integer   | Specifies the duration of time for which a change to be propagated to a mirror should be delayed, in seconds. This is an intentional propagation delay between mirrors and is configurable down to zero, which means an immediate propagation. This is supported for policies of type 'continuous'. |
| scope                         | string  | Set to "svm" for policies owned by an SVM, otherwise set to "cluster".  |
| sync_common_snapshot_schedule | <a href="#">sync_common_snapshot_schedule</a>   | Schedule used to create common Snapshot copies for synchronous relationships.   |
| throttle                      | integer   | Throttle in KB/s. Default to unlimited.   |
| transfer_schedule             | <a href="#">transfer_schedule</a>               | The schedule used to update asynchronous relationships. Only cron schedules are supported for SnapMirror.   |
| uuid                          | string  | Unique identifier of the SnapMirror policy.   |

## Example request

```
{  
    "comment": "string",  
    "copy_all_source_snapshots": 1,  
    "copy_latest_source_snapshot": 1,  
    "create_snapshot_on_source": "",  
    "identity_preservation": "string",  
    "retention": [  
        {  
            "count": 7,  
            "creation_schedule": {  
                "name": "weekly",  
                "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
,  
                "label": "hourly",  
                "period": "P30D",  
                "prefix": "string",  
                "preserve": 1,  
                "warn": 4  
            }  
        },  
        "rpo": 0,  
        "scope": "string",  
        "sync_common_snapshot_schedule": {  
            "name": "weekly",  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
,  
            "throttle": 0,  
            "transfer_schedule": {  
                "name": "weekly",  
                "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
,  
                "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"  
        }  
    }  
}
```

## Response

Status: 200, Ok

| Name | Type     | Description |
|------|----------|-------------|
| job  | job_link |             |

## Example response

```
{  
  "job": {  
    "uuid": "string"  
  }  
}
```

## Response

Status: 202, Accepted

## Error

Status: Default

## ONTAP Error Response codes

| Error code   | Description  |
|--|--|
| -----  | -----  |
| 6619714  | Schedule specified is an interval schedule. SnapMirror does not support interval schedules.  |
| 13303842   |  |
| SnapMirror policy is not supported.  |  |
| 13303843   | Conflicting values between SnapMirror policy and SnapMirror relationships for either 'transfer_schedule', 'throttle' or 'identity_preservation' properties |
| 13303850   |  |
| Invalid input parameter  |  |
| 13303887   | Failed to create SnapMirror policy. Reason: Maximum number of allowed retention rules reached  |
| 13304050   |  |
| Retention cannot be empty for a SnapMirror policy with 'create_snapshot_on_source' set to false. |  |

| Error code   | Description  |
|--|--|
| 13304092   | Input value of the retention period property is invalid. For relationships with FlexVol volume or FlexGroup volume destinations, the duration must be in ISO 6801 format or can be infinite. For relationships with object store destinations, only duration values with Y, M or D and supported and must be in the specified range. |
|  | 6621045  |
| The property retention.warn is not supported for a policy when the retention.preserve property is false. |  |
| 13304129   | The property retention.warn value must be less than the property retention.count value for a rule in a policy.   |
|  | 13304126   |
| Enter a value for "count" in the range indicated in the error message.                                   |  |
| 13304130   | The total retention.count value for all rules in a policy cannot exceed the value indicated in the error message."   |

## Definitions

## See Definitions

href

| Name | Type   | Description |
|------|--------|-------------|
| href | string |             |

\_links

creation\_schedule

Schedule used to create Snapshot copies on the destination for long term retention. Only cron schedules are supported for SnapMirror.

| Name | Type   | Description       |
|------|--------|-------------------|
| name | string | Job schedule name |
| uuid | string | Job schedule UUID |

snapmirror\_policy\_rule

SnapMirror policy rule for retention.

| Name              | Type              | Description   |
|-------------------|-------------------|---|
| count             | integer           | Number of Snapshot copies to be kept for retention. Maximum value will differ based on type of relationship and scaling factor.       |
| creation_schedule | creation_schedule | Schedule used to create Snapshot copies on the destination for long term retention. Only cron schedules are supported for SnapMirror. |
| label             | string            | Snapshot copy label   |

| Name     | Type    | Description  |
|----------|---------|--|
| period   | string  | <p>Specifies the duration for which the Snapshot copies are locked. The retention period value represents a duration and must be in the ISO-8601 duration format. Years, months, days, hours, minutes, and seconds are represented as "P&lt;num&gt;Y", "P&lt;num&gt;M", "P&lt;num&gt;D", "PT&lt;num&gt;H", "PT&lt;num&gt;M" and "PT&lt;num&gt;S". Value "infinite" is also a valid input for Flexvol volumes and FlexGroup volumes. A duration which combines different periods is not supported, for example "P1Y10M" is not supported. The range of supported retention period values is between 1 second to infinite.&lt;/num&gt;&lt;/num&gt;&lt;/num&gt;&lt;/num&gt;&lt;/num&gt;&lt;/num&gt;</p> |
| prefix   | string  | <p>Specifies the prefix for the Snapshot copy name to be created as per the schedule. If no value is specified, then the label is used as the prefix.</p>  |
| preserve | boolean | <p>Specifies the behavior when the Snapshot copy retention count is reached on the SnapMirror destination for the rule. The default value is false, which means that the oldest Snapshot copy is deleted to make room for new ones but only if the number of Snapshot copies has exceeded the retention count specified in the 'count' property. When set to true and where the Snapshot copies have reached the retention count, an incremental SnapMirror transfer will fail or if the rule has a schedule, Snapshot copies will no longer be created on the SnapMirror destination.</p>   |

| Name | Type    | Description   |
|------|---------|---|
| warn | integer | Specifies the warning threshold count for the rule. The default value is zero. When set to a value greater than zero, an event is generated after the number of Snapshot copies (for the particular rule) retained on a SnapMirror destination reaches the specified warning limit. The preserve property for the rule must be true in order to set the warn property to a value greater than zero. |

## svm

SVM, applies only to SVM-scoped objects.

| Name | Type   | Description   |
|------|--------|---|
| name | string | The name of the SVM. This field cannot be specified in a PATCH method.              |
| uuid | string | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

## sync\_common\_snapshot\_schedule

Schedule used to create common Snapshot copies for synchronous relationships.

| Name | Type   | Description       |
|------|--------|-------------------|
| name | string | Job schedule name |
| uuid | string | Job schedule UUID |

## transfer\_schedule

The schedule used to update asynchronous relationships. Only cron schedules are supported for SnapMirror.

| Name | Type   | Description       |
|------|--------|-------------------|
| name | string | Job schedule name |
| uuid | string | Job schedule UUID |

## snapshot\_policy

SnapMirror policy information. SnapMirror policy can either be of type "async", "sync" or "continuous". The policy type "async" can be associated with a SnapMirror relationship that has either a FlexVol volume, FlexGroup volume or SVM as the endpoint. The policy type "sync" along with "sync\_type" as "sync" or "strict\_sync" can be associated with a SnapMirror relationship that has a FlexVol volume as the endpoint. The policy type "sync" can have a "sync\_type" of either "sync", "strict\_sync", "automated\_failover" or "automated\_failover\_duplex". If the "sync\_type" is "sync", a write success is returned to the client after writing the data to the source endpoint and before writing the data to the destination endpoint. If the "sync\_type" is "strict\_sync", a write success is returned to the client after writing the data to both source and destination endpoints. If the "sync\_type" is "automated\_failover", the policy can be associated with a SnapMirror relationship that has a consistency group as the endpoint and provides asymmetric active active access to the two storage copies. If the "sync\_type" is "automated\_failover\_duplex", the policy can be associated with a SnapMirror relationship that has a consistency group as the endpoint and provides symmetric active active access to the two storage copies. Use the "sync" policy with "sync\_type" as "automated\_failover" or "automated\_failover\_duplex" to establish SnapMirror relationships for business continuity use cases. SnapMirror relationships with policy types as "sync" and "sync\_type" as "automated\_failover" or "automated\_failover\_duplex" can be monitored by the Mediator, if configured. If the source Consistency Group endpoint is not reachable, the Mediator might trigger a failover to the destination consistency group endpoint. A policy type of "continuous" can be associated with SnapMirror relationships that have either ONTAP S3 buckets or non-ONTAP object stores as endpoints. This type of policy is used for FabricLink owned targets.

| Name                        | Type                          | Description  |
|-----------------------------|-------------------------------|--|
| comment                     | string                        | Comment associated with the policy.  |
| identity_preservation       | string                        | Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type. |
| network_compression_enabled | boolean                       | Specifies whether network compression is enabled for transfers. This is applicable only to the policies of type "async".   |
| retention                   | array[snapmirror_policy_rule] | Rules for Snapshot copy retention.   |

| Name                          | Type  | Description   |
|-------------------------------|---|---|
| rpo                           | integer                                       | Specifies the duration of time for which a change to be propagated to a mirror should be delayed, in seconds. This is an intentional propagation delay between mirrors and is configurable down to zero, which means an immediate propagation. This is supported for policies of type 'continuous'. |
| scope                         | string  | Set to "svm" for policies owned by an SVM, otherwise set to "cluster".  |
| sync_common_snapshot_schedule | <a href="#">sync_common_snapshot_schedule</a> | Schedule used to create common Snapshot copies for synchronous relationships.   |
| throttle                      | integer                                       | Throttle in KB/s. Default to unlimited.   |
| transfer_schedule             | <a href="#">transfer_schedule</a>             | The schedule used to update asynchronous relationships. Only cron schedules are supported for SnapMirror.   |
| uuid                          | string  | Unique identifier of the SnapMirror policy.   |

## job\_link

| Name | Type   | Description   |
|------|--------|---|
| uuid | string | The UUID of the asynchronous job that is triggered by a POST, PATCH, or DELETE operation. |

## error\_arguments

| Name    | Type   | Description      |
|---------|--------|------------------|
| code    | string | Argument code    |
| message | string | Message argument |

## returned\_error

| Name      | Type                   | Description                                 |
|-----------|------------------------|---|
| arguments | array[error_arguments] | Message arguments                           |
| code      | string                 | Error code                                  |
| message   | string                 | Error message                               |
| target    | string                 | The target parameter that caused the error. |

## Manage SnapMirror relationships

### SnapMirror relationships endpoint overview

#### Overview

This API manages asynchronous extended data protection (XDP) relationships for FlexVol volumes, FlexGroup volumes, or SVMs. It is also used to manage a synchronous relationship between FlexVol volumes, which provides zero RPO data protection and active synchronous relationship with automated failover between Application Consistency Group endpoints which provides zero RTO data protection.

To create an asynchronous extended data protection relationship with FlexVol volumes, FlexGroup volumes, Application Consistency Groups or SVMs, use the policy of type "async".

To create a synchronous relationship between FlexVol volumes, use the policy of type "sync" with sync\_type of either "sync" or "strict\_sync". To create an active synchronous relationship with automated failover between Application Consistency Group endpoints, use the policy of type "sync" with sync\_type "automated\_failover". You can create an asynchronous extended data protection relationship between the source and destination which can be used by the transfer APIs to perform SnapMirror "restore" operations.

To create FlexVol volume or FlexGroup volume SnapMirror relationships, the source volume must be in the "online" state and be a read\_write type; the destination volume must be in the "online" state and be a data protection type.

In the case of an asynchronous or synchronous SnapMirror relationship for an Application Consistency Group of FlexVol volumes, SnapMirror creation results in the creation of an Application Consistency Group on the source cluster if it did not already exist with the exact same name and set of FlexVol volumes specified in the current operation. Additionally, if the specified Application Consistency Group is already present and is already a part of an existing SnapMirror relationship, the process fails. Creating an Application Consistency Group on the destination cluster is part of the SnapMirror creation workflow.

To create SnapMirror relationships between SVMs, the source SVM must be of subtype "default" and the destination SVM of subtype "dp\_destination". Additionally, SVMs must be peered before a relationship can be established between them when the "create\_destination" property is not specified. When the "create\_destination" property is specified, the destination SVM is provisioned on the destination cluster and the SVM peer relationship is established between the source SVM and the new destination SVM, provided that the source SVM has SVM peering permissions for the destination cluster.

Data protection FlexVol volume SnapMirror relationships cannot be created using this API but existing relationships can be listed or managed.

The SnapMirror functionality is subdivided into relationship APIs and transfer APIs:

- SnapMirror relationship APIs are used to create and manage the SnapMirror relationships.
- SnapMirror transfer APIs are used to manage data transfers.

## Retrieve information for SnapMirror relationships

GET /snapmirror/relationships

**Introduced In:** 9.6

Retrieves information for SnapMirror relationships whose destination endpoints are in the current SVM or the current cluster, depending on the cluster context.

### Related ONTAP commands

- snapmirror show
- snapmirror list-destinations

### Expensive properties

- source.consistency\_group\_volumes.name
- destination.consistency\_group\_volumes.name
- svmdr\_volumes.name

### Examples

The following examples show how to retrieve the list of SnapMirror relationships and the list of SnapMirror destinations.

1. Retrieving the list of SnapMirror relationships. This API must be run on the cluster containing the destination endpoint.

```
GET "/api/snapmirror/relationships/"
```

1. Retrieving the list of SnapMirror destinations on source. This must be run on the cluster containing the source endpoint.

```
GET "/api/snapmirror/relationships/?list_destinations_only=true"
```

1. Retrieving the relationship UUID of SnapMirror relationships with lag time greater than 2 days. This API must be run on the cluster containing the destination endpoint.

```
GET "/api/snapmirror/relationships/?fields=uuid&lag_time=>P2DT"
```

1. Retrieving the list of SnapMirror relationships with lag time less than 10 hours. This API must be run on the

cluster containing the destination endpoint.

```
GET "/api/snapmirror/relationships/?lag_time=<PT10H"
```

1. Retrieving the list of constituent volumes for SVM DR Snapmirror relationships.

```
GET "/api/snapmirror/relationships/?fields=svmdr_volumes.name"
```

</private>

## Learn more

- [DOC /snapmirror/relationships](#)

## Parameters

| Name                                   | Type    | In    | Required | Description   |
|--|---------|-------|----------|---|
| list_destinations_only                 | boolean | query | False    | Set to true to show relationships from the source only.   |
| throttle                               | integer | query | False    | Filter by throttle <ul style="list-style-type: none"><li>• Introduced in: 9.11</li></ul>                              |
| identity_preservation                  | string  | query | False    | Filter by identity_preservation <ul style="list-style-type: none"><li>• Introduced in: 9.11</li></ul>                 |
| restore                                | boolean | query | False    | Filter by restore   |
| consistency_group failover.status.code | string  | query | False    | Filter by consistency_group_failover.status.code <ul style="list-style-type: none"><li>• Introduced in: 9.8</li></ul> |

| Name  | Type   | In    | Required | Description  |
|---|--------|-------|----------|--|
| consistency_group_f ailover.status.message          | string | query | False    | Filter by consistency_group_f ailover.status.message <ul style="list-style-type: none"> <li>• Introduced in: 9.8</li> </ul>          |
| consistency_group_f ailover.type                    | string | query | False    | Filter by consistency_group_f ailover.type <ul style="list-style-type: none"> <li>• Introduced in: 9.14</li> </ul>                   |
| consistency_group_f ailover.state                   | string | query | False    | Filter by consistency_group_f ailover.state <ul style="list-style-type: none"> <li>• Introduced in: 9.14</li> </ul>                  |
| consistency_group_f ailover.error.code              | string | query | False    | Filter by consistency_group_f ailover.error.code <ul style="list-style-type: none"> <li>• Introduced in: 9.8</li> </ul>              |
| consistency_group_f ailover.error.arguments.code    | string | query | False    | Filter by consistency_group_f ailover.error.arguments.code <ul style="list-style-type: none"> <li>• Introduced in: 9.8</li> </ul>    |
| consistency_group_f ailover.error.arguments.message | string | query | False    | Filter by consistency_group_f ailover.error.arguments.message <ul style="list-style-type: none"> <li>• Introduced in: 9.8</li> </ul> |

| Name                                       | Type   | In    | Required | Description   |
|--|--------|-------|----------|---|
| consistency_group_failover.error.message   | string | query | False    | Filter by consistency_group_failover.error.message <ul style="list-style-type: none"> <li>• Introduced in: 9.8</li> </ul>   |
| io_serving_copy                            | string | query | False    | Filter by io_serving_copy <ul style="list-style-type: none"> <li>• Introduced in: 9.14</li> </ul>                           |
| destination.consistency_group_volumes.name | string | query | False    | Filter by destination.consistency_group_volumes.name <ul style="list-style-type: none"> <li>• Introduced in: 9.8</li> </ul> |
| destination.cluster.uid                    | string | query | False    | Filter by destination.cluster.uid <ul style="list-style-type: none"> <li>• Introduced in: 9.7</li> </ul>                    |
| destination.cluster.name                   | string | query | False    | Filter by destination.cluster.name <ul style="list-style-type: none"> <li>• Introduced in: 9.7</li> </ul>                   |
| destination.uuid                           | string | query | False    | Filter by destination.uuid <ul style="list-style-type: none"> <li>• Introduced in: 9.11</li> </ul>                          |
| destination.svm.uuid                       | string | query | False    | Filter by destination.svm.uuid  |
| destination.svm.name                       | string | query | False    | Filter by destination.svm.name  |

| Name                                    | Type   | In    | Required | Description   |
|---|--------|-------|----------|---|
| destination.path                        | string | query | False    | Filter by destination.path  |
| exported_snapshot                       | string | query | False    | Filter by exported_snapshot   |
| lag_time                                | string | query | False    | Filter by lag_time  |
| state                                   | string | query | False    | Filter by state   |
| last_transfer_network_compression_ratio | string | query | False    | Filter by last_transfer_network_compression_ratio <ul style="list-style-type: none"> <li>• Introduced in: 9.13</li> </ul> |
| uuid                                    | string | query | False    | Filter by uuid  |
| transfer.last_updated_time              | string | query | False    | Filter by transfer.last_updated_time <ul style="list-style-type: none"> <li>• Introduced in: 9.14</li> </ul>              |
| transfer.uuid                           | string | query | False    | Filter by transfer.uuid   |
| transfer.total_duration                 | string | query | False    | Filter by transfer.total_duration <ul style="list-style-type: none"> <li>• Introduced in: 9.9</li> </ul>                  |
| transfer.state                          | string | query | False    | Filter by transfer.state  |
| transfer.end_time                       | string | query | False    | Filter by transfer.end_time <ul style="list-style-type: none"> <li>• Introduced in: 9.9</li> </ul>                        |

| Name                       | Type    | In    | Required | Description   |
|----------------------------|---------|-------|----------|---|
| transfer.type              | string  | query | False    | Filter by transfer.type <ul style="list-style-type: none"> <li>• Introduced in: 9.14</li> </ul>           |
| transfer.bytes_transferred | integer | query | False    | Filter by transfer.bytes_transferred  |
| total_transfer_duration    | string  | query | False    | Filter by total_transfer_duration <ul style="list-style-type: none"> <li>• Introduced in: 9.13</li> </ul> |
| master_bias_active_site    | string  | query | False    | Filter by master_bias_active_site <ul style="list-style-type: none"> <li>• Introduced in: 9.14</li> </ul> |
| healthy                    | boolean | query | False    | Filter by healthy   |
| group_type                 | string  | query | False    | Filter by group_type <ul style="list-style-type: none"> <li>• Introduced in: 9.11</li> </ul>              |
| last_transfer_type         | string  | query | False    | Filter by last_transfer_type <ul style="list-style-type: none"> <li>• Introduced in: 9.11</li> </ul>      |
| backoff_level              | string  | query | False    | Filter by backoff_level <ul style="list-style-type: none"> <li>• Introduced in: 9.14</li> </ul>           |
| unhealthy_reason.message   | string  | query | False    | Filter by unhealthy_reason.message  |

| Name                       | Type    | In    | Required | Description  |
|----------------------------|---------|-------|----------|--|
| unhealthy_reason.code      | string  | query | False    | Filter by unhealthy_reason.code  |
| unhealthy_reason.arguments | string  | query | False    | Filter by unhealthy_reason.arguments <ul style="list-style-type: none"> <li>• Introduced in: 9.14</li> </ul> |
| total_transfer_bytes       | integer | query | False    | Filter by total_transfer_bytes <ul style="list-style-type: none"> <li>• Introduced in: 9.13</li> </ul>       |
| source.path                | string  | query | False    | Filter by source.path  |
| source.svm.uuid            | string  | query | False    | Filter by source.svm.uuid  |
| source.svm.name            | string  | query | False    | Filter by source.svm.name  |
| source.cluster.uuid        | string  | query | False    | Filter by source.cluster.uuid <ul style="list-style-type: none"> <li>• Introduced in: 9.7</li> </ul>         |
| source.cluster.name        | string  | query | False    | Filter by source.cluster.name <ul style="list-style-type: none"> <li>• Introduced in: 9.7</li> </ul>         |
| source.uuid                | string  | query | False    | Filter by source.uuid <ul style="list-style-type: none"> <li>• Introduced in: 9.11</li> </ul>                |

| Name                                  | Type          | In    | Required | Description  |
|---------------------------------------|---------------|-------|----------|--|
| source.consistency_group_volumes.name | string        | query | False    | Filter by source.consistency_group_volumes.name <ul style="list-style-type: none"> <li>• Introduced in: 9.8</li> </ul> |
| preferred_site                        | string        | query | False    | Filter by preferred_site <ul style="list-style-type: none"> <li>• Introduced in: 9.14</li> </ul>                       |
| policy.name                           | string        | query | False    | Filter by policy.name  |
| policy.uuid                           | string        | query | False    | Filter by policy.uuid  |
| policy.type                           | string        | query | False    | Filter by policy.type  |
| transfer_schedule.name                | string        | query | False    | Filter by transfer_schedule.name <ul style="list-style-type: none"> <li>• Introduced in: 9.11</li> </ul>               |
| transfer_schedule.uid                 | string        | query | False    | Filter by transfer_schedule.uid <ul style="list-style-type: none"> <li>• Introduced in: 9.11</li> </ul>                |
| svmdr_volumes.name                    | string        | query | False    | Filter by svmdr_volumes.name <ul style="list-style-type: none"> <li>• Introduced in: 9.13</li> </ul>                   |
| fields                                | array[string] | query | False    | Specify the fields to return.  |
| max_records                           | integer       | query | False    | Limit the number of records returned.  |

| Name           | Type          | In    | Required | Description  |
|----------------|---------------|-------|----------|--|
| return_records | boolean       | query | False    | <p>The default is true for GET calls. When set to false, only the number of records is returned.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> </ul>  |
| return_timeout | integer       | query | False    | <p>The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul> |
| order_by       | array[string] | query | False    | Order results by specified fields and optional [asc]   |

## Response

Status: 200, Ok

| Name        | Type   | Description       |
|-------------|--|-------------------|
| _links      | <a href="#">_links</a>                           |                   |
| num_records | integer  | Number of records |
| records     | array[ <a href="#">snapmirror_relationship</a> ] |                   |

## Example response

```
{  
  "_links": {  
    "next": {  
      "href": "/api/resourcelink"  
    },  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "num_records": 1,  
  "records": [  
    {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "backoff_level": "medium",  
      "consistency_group_failover": {  
        "error": {  
          "arguments": [  
            {  
              "code": "string",  
              "message": "string"  
            }  
          ],  
          "code": "4",  
          "message": "entry doesn't exist"  
        },  
        "state": "string",  
        "status": {  
          "code": "string",  
          "message": "string"  
        },  
        "type": "string"  
      },  
      "destination": {  
        "cluster": {  
          "_links": {  
            "self": {  
              "href": "/api/resourcelink"  
            }  
          },  
          "name": "cluster1",  
          "type": "string"  
        }  
      }  
    }  
  ]  
}
```

```
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "consistency_group_volumes": [
        {
            "name": "volume1"
        }
    ],
    "path": "svm1:volumel",
    "svm": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "name": "svm1",
        "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
    }
},
"exported_snapshot": "string",
"group_type": "consistency_group",
"identity_preservation": "string",
"io_serving_copy": "C1_sti85-vsimm-ucs209a_cluster, C1_sti85-vsimm-ucs209c_cluster",
"lag_time": "PT8H35M42S",
"last_transfer_network_compression_ratio": 61,
"last_transfer_type": "initialize",
"master_bias_activated_site": "C1_sti85-vsimm-ucs209a_cluster",
"policy": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "name": "Asynchronous",
    "type": "string",
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
},
"preferred_site": "C1_sti85-vsimm-ucs209a_cluster",
"restore_to_snapshot": "string",
"source": {
    "cluster": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        }
    },
    "name": "Asynchronous"
}
```

```
        "name": "cluster1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "consistency_group_volumes": [
        {
            "name": "volume1"
        }
    ],
    "path": "svm1:volume1",
    "svm": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "name": "svm1",
        "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
    }
},
"state": "snapmirrored",
"svmdr_volumes": [
    {
        "name": "volume1"
    }
],
"throttle": 0,
"total_transfer_bytes": 1098210312,
"total_transfer_duration": "PT3M21S",
"transfer": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    }
},
"end_time": "2020-12-02 21:36:19 -0500",
"last_updated_time": "2023-09-14 18:39:19 -0400",
"state": "string",
"total_duration": "PT28M41S",
"type": "initialize",
"uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
},
"transfer_schedule": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    }
}
```

```

        },
        "name": "weekly",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "unhealthy_reason": [
        {
            "arguments": [],
            "code": "6621444",
            "message": "Failed to complete update operation on one or
more item relationships."
        },
        {
            "arguments": [],
            "code": "6621445",
            "message": "Group Update failed"
        }
    ],
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
}
]
}
}

```

## Error

Status: Default

### ONTAP Error Response codes

| Error code | Description  |
|------------|--|
| 13303825   | Could not retrieve information for the SnapMirror<br>policy type |
| 13303817   | Unknown value for the Snapmirror State                           |

| Name  | Type           | Description |
|-------|----------------|-------------|
| error | returned_error |             |

## Example error

```
{  
  "error": {  
    "arguments": [  
      {  
        "code": "string",  
        "message": "string"  
      }  
    ],  
    "code": "4",  
    "message": "entry doesn't exist",  
    "target": "uuid"  
  }  
}
```

## Definitions

## See Definitions

href

| Name | Type   | Description |
|------|--------|-------------|
| href | string |             |

\_links

| Name | Type | Description |
|------|------|-------------|
| next | href |             |
| self | href |             |

\_links

| Name | Type | Description |
|------|------|-------------|
| self | href |             |

error\_arguments

| Name    | Type   | Description      |
|---------|--------|------------------|
| code    | string | Argument code    |
| message | string | Message argument |

error

SnapMirror Consistency Group failover error message.

| Name      | Type                   | Description       |
|-----------|------------------------|-------------------|
| arguments | array[error_arguments] | Message arguments |
| code      | string                 | Error code        |
| message   | string                 | Error message     |

status

| Name    | Type   | Description                                   |
|---------|--------|---|
| code    | string | Status code                                   |
| message | string | SnapMirror Consistency Group failover status. |

## snapmirror\_consistency\_group\_failover

SnapMirror Consistency Group failover information. The SnapMirror Consistency Group failover can be a planned or an unplanned operation. Only active SnapMirror Consistency Group failover operation progress can be monitored using this object. In case of an error during the failover operation, the property "consistency\_group\_failover.error" holds the reason for the error. ONTAP automatically retries any failed SnapMirror Consistency Group failover operation.

| Name   | Type                   | Description  |
|--------|------------------------|--|
| error  | <a href="#">error</a>  | SnapMirror Consistency Group failover error message. |
| state  | string                 | SnapMirror Consistency Group failover state.         |
| status | <a href="#">status</a> |  |
| type   | string                 | SnapMirror Consistency Group failover type.          |

## bucket\_retention

Specifies the retention-mode and default retention period configured on the destination bucket.

| Name           | Type   | Description  |
|----------------|--------|--|
| default_period | string | Specifies the default retention period that is applied to objects while committing them to the WORM state without an associated retention period. The retention period can be in years, or days. The retention period value represents a duration and must be specified in the ISO-8601 duration format. A period specified for years and days is represented in the ISO-8601 format as "P<num>Y" and "P<num>D" respectively, for example "P10Y" represents a duration of 10 years. The period string must contain only a single time element that is, either years, or days. A duration which combines different periods is not supported, for example "P1Y10D" is not supported.</num></num> |

| Name | Type   | Description   |
|------|--------|---|
| mode | string | The lock mode of the bucket. compliance – A SnapLock Compliance (SLC) bucket provides the highest level of WORM protection and an administrator cannot destroy a compliance bucket if it contains unexpired WORM objects. governance – An administrator can delete a Governance bucket. no_lock – Indicates the bucket does not support object locking. |

### storage\_service

| Name                | Type    | Description   |
|---------------------|---------|---|
| enabled             | boolean | This property indicates whether to create the destination endpoint using storage service.   |
| enforce_performance | boolean | Optional property to enforce storage service performance on the destination endpoint. This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints. |

| Name | Type   | Description   |
|------|--------|---|
| name | string | <p>Optional property to specify the storage service name for the destination endpoint. This property is considered when the property "create_destination.storage_service.enabled" is set to "true". When the property "create_destination.storage_service.enabled" is set to "true" and the "create_destination.storage_service.name" for the endpoint is not specified, then ONTAP selects the highest storage service available on the cluster to provision the destination endpoint. This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints.</p> <ul style="list-style-type: none"> <li>• enum: ["extreme", "performance", "value"]</li> <li>• Introduced in: 9.6</li> <li>• x-nullable: true</li> </ul> |

tiering

| Name   | Type   | Description   |
|--------|--------|---|
| policy | string | <p>Optional property to specify the destination endpoint's tiering policy when "create_destination.tiering.supported" is set to "true". This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints. This property determines whether the user data blocks of the destination endpoint in a FabricPool will be tiered to the cloud store when they become cold. FabricPool combines flash (performance tier) with a cloud store into a single aggregate. Temperature of the destination endpoint volume blocks increases if they are accessed frequently and decreases when they are not. all – This policy allows tiering of both destination endpoint Snapshot copies and the user transferred data blocks to the cloud store as soon as possible by ignoring the temperature on the volume blocks. This tiering policy is not applicable for Consistency Group destination endpoints or for synchronous relationships. auto – This policy allows tiering of both destination endpoint Snapshot copies and the active file system user data to the cloud store none – Destination endpoint volume blocks will not be tiered to the cloud store. snapshot_only – This policy allows tiering of only the destination endpoint volume Snapshot copies not associated with the active file system. The default tiering policy is "snapshot_only" for a FlexVol volume and "none" for a FlexGroup volume.</p> |

| Name      | Type    | Description   |
|-----------|---------|---|
| supported | boolean | Optional property to enable provisioning of the destination endpoint volumes on FabricPool aggregates. This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints. Only FabricPool aggregates are used if this property is set to "true" and only non FabricPool aggregates are used if this property is set to "false". Tiering support for a FlexGroup volume can be changed by moving all of the constituents to the required aggregates. Note that in order to tier data, not only do the destination endpoint volumes need to support tiering by using FabricPools, the "create_destination.tiering.policy" must not be "none". A destination endpoint that uses FabricPools but has a tiering "policy" of "none" supports tiering but will not tier any data. |

#### snapmirror\_destination\_creation

Use this object to provision the destination endpoint when establishing a SnapMirror relationship for a FlexVol volume, FlexGroup volume, SVM, Consistency Group or ONTAP S3 Bucket. Given a source endpoint, the destination endpoint is provisioned in the SVM specified in the "destination.path" property. While protecting an SVM, the SVM destination endpoint can only be provisioned on the local cluster. To provision the SVM destination endpoint use the optional "source.cluster.name" property to specify the remote cluster name or use the optional "source.cluster.uuid" property to specify the remote cluster UUID. When "create\_destination.enabled" option is specified while making a POST for a SnapMirror relationship, the relationship can be automatically initialized by setting the "state" either to "snapmirrored" when the policy is of type "async" or to "in\_sync" when the policy is of type "sync". The "destination.path" property must specify the destination endpoint path. For example, for FlexVol volume and FlexGroup volume, the "destination.path" can be specified as <destination-svm-name:dp-volume-name>, for SVM data protection, the "destination.path" must be specified as &lt;destination-svm-name:&gt;, and for Consistency Group, the "destination.path" must be specified as &lt;destination-svm-name:&gt; along with the "destination.consistency\_group\_volumes" property to indicate the list of destination volumes of type "DP" in the Consistency Group. For a FlexVol volume, a FlexGroup volume, Consistency Group or a Bucket destination endpoint, the properties in this object can be specified either from the source or the destination cluster. For an SVM destination endpoint, the properties in this object can be specified from the destination cluster only. This object is not supported for non ONTAP endpoints. While protecting a S3 Bucket, the optional "size" property can be used to create ONTAP S3 Bucket destination endpoint of the specified size.&lt;/destination-svm-name:&gt;&lt;/destination-svm-name:&gt;</destination-svm-name:dp-volume-name>

| Name                     | Type                             | Description  |
|--------------------------|----------------------------------|--|
| bucket_retention         | <a href="#">bucket_retention</a> | Specifies the retention-mode and default retention period configured on the destination bucket.  |
| enabled                  | boolean                          | Optional property to create the destination endpoint when establishing a SnapMirror relationship. It is assumed to be "false" if no other property is set and assumed to be "true" if any other property is set.   |
| size                     | integer                          | Optional property to specify the size of destination endpoint in bytes. This property is applicable only to ONTAP S3 Bucket endpoints. The minimum size for S3 bucket is 80MB and maximum size is 64TB. If not specified, system will create destination with default size of 800GB. <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> <li>• x-nullable: true</li> </ul> |
| snapshot_locking_enabled | boolean                          | Optional property to create the destination endpoint with Snapshot copy locking enabled when establishing a SnapMirror relationship. This property is applicable to FlexVol volumes and FlexGroup volumes.   |
| storage_service          | <a href="#">storage_service</a>  |  |
| tiering                  | <a href="#">tiering</a>          |  |

cluster

| Name   | Type                   | Description |
|--------|------------------------|-------------|
| _links | <a href="#">_links</a> |             |
| name   | string                 |             |
| uuid   | string                 |             |

consistency\_group\_volumes

| Name | Type   | Description             |
|------|--------|-------------------------|
| name | string | The name of the volume. |

## svm

SVM, applies only to SVM-scoped objects.

| Name   | Type                   | Description   |
|--------|------------------------|---|
| _links | <a href="#">_links</a> |   |
| name   | string                 | The name of the SVM. This field cannot be specified in a PATCH method.              |
| uuid   | string                 | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

## snapmirror\_endpoint

Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.

| Name                      | Type   | Description   |
|---------------------------|--|---|
| cluster                   | <a href="#">cluster</a>                            |   |
| consistency_group_volumes | array[ <a href="#">consistency_group_volume</a> s] | Mandatory property for a Consistency Group endpoint. Specifies the list of FlexVol volumes for a Consistency Group. |

| Name | Type   | Description  |
|------|--------|--|
| path | string | ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name<br>ONTAP S3 - svm1:/bucket/bucket1 NON-ONTAP - objstore1:/objstore <ul style="list-style-type: none"> <li>• example: svm1:volume1</li> <li>• Introduced in: 9.6</li> <li>• x-nullable: true</li> </ul> |
| svm  | svm    | SVM, applies only to SVM-scoped objects.   |

## policy

Basic policy information of the relationship.

| Name   | Type   | Description                                 |
|--------|--------|---|
| _links | _links |   |
| name   | string | Name of the SnapMirror policy.              |
| type   | string |   |
| uuid   | string | Unique identifier of the SnapMirror policy. |

## snapmirror\_source\_endpoint

Source endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.

| Name                      | Type                             | Description   |
|---------------------------|----------------------------------|---|
| cluster                   | cluster                          |   |
| consistency_group_volumes | array[consistency_group_volumes] | Mandatory property for a Consistency Group endpoint. Specifies the list of FlexVol volumes for a Consistency Group. |

| Name | Type   | Description   |
|------|--------|---|
| path | string | ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name<br>ONTAP S3 - svm1:/bucket/bucket1 NON-ONTAP - objstore1:/objstore <ul style="list-style-type: none"> <li>• example: svm1:volume1</li> <li>• Introduced in: 9.14</li> <li>• x-nullable: true</li> </ul> |
| svm  | svm    | SVM, applies only to SVM-scoped objects.  |

#### svmdr\_volumes

| Name | Type   | Description             |
|------|--------|-------------------------|
| name | string | The name of the volume. |

#### transfer

Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.

| Name              | Type                   | Description  |
|-------------------|------------------------|--|
| _links            | <a href="#">_links</a> |  |
| bytes_transferred | integer                | Total bytes transferred in the current or last successful transfer.            |
| end_time          | string                 | End time of the last transfer.   |
| last_updated_time | string                 | Last updated time of the bytes transferred in the current transfer.            |
| state             | string                 |  |
| total_duration    | string                 | Elapsed time to transfer all Snapshot copies for the last successful transfer. |

| Name | Type   | Description  |
|------|--------|--|
| type | string | Specifies the operation type of the current transfer on the relationship. The <i>initialize</i> transfer occurs when the relationship state changes from "uninitialized" to "snapmirrored" or "in_sync". The <i>update</i> transfer occurs when Snapshot copies are being transferred from the source endpoint to the destination endpoint as part of a scheduled or manual update. The <i>resync</i> transfer occurs when the relationship state changes from "broken_off" to "snapmirrored" or "in_sync". The <i>restore</i> transfer occurs when a Snapshot copy is being restored from a destination endpoint to another endpoint. |
| uuid | string | Transfer UUID. This property is applicable only for active transfers.  |

### transfer\_schedule

Schedule used to update asynchronous relationships. This "transfer\_schedule" overrides the "transfer\_schedule" set on the SnapMirror relationship's policy. To remove the "transfer\_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror.

| Name   | Type                   | Description       |
|--------|------------------------|-------------------|
| _links | <a href="#">_links</a> |                   |
| name   | string                 | Job schedule name |
| uuid   | string                 | Job schedule UUID |

### snapmirror\_error

#### SnapMirror error

| Name      | Type          | Description   |
|-----------|---------------|---|
| arguments | array[string] | Arguments present in the error message encountered. |
| code      | string        | Error code  |

| Name    | Type   | Description   |
|---------|--------|---------------|
| message | string | Error message |

### snapmirror\_relationship

SnapMirror relationship information. The SnapMirror relationship can be either "async" or "sync" based on the type of SnapMirror policy associated with the relationship. The source and destination endpoints of a SnapMirror relationship must be of the same type, for example, if the source endpoint is a FlexVol volume then the destination endpoint must be a FlexVol volume. The SnapMirror policy type "async" can be used when the SnapMirror relationship has FlexVol volume or FlexGroup volume or SVM as the endpoint. The SnapMirror policy type "sync" can be used when the SnapMirror relationship has FlexVol volume as the endpoint. The SnapMirror policy type "sync" with "sync\_type" as "automated\_failover" can be used when the SnapMirror relationship has Consistency Group as the endpoint.

| Name                       | Type  | Description   |
|----------------------------|---|---|
| _links                     | <a href="#">_links</a>                                |   |
| backoff_level              | string  | Specifies the SnapMirror backoff level due to Client Ops for FlexVol SnapMirror relationships.  |
| consistency_group_failover | <a href="#">snapmirror_consistency_group_failover</a> | <p>SnapMirror Consistency Group failover information. The SnapMirror Consistency Group failover can be a planned or an unplanned operation. Only active SnapMirror Consistency Group failover operation progress can be monitored using this object. In case of an error during the failover operation, the property "consistency_group_failover.error" holds the reason for the error. ONTAP automatically retries any failed SnapMirror Consistency Group failover operation.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.8</li> <li>• readOnly: 1</li> </ul> |

| Name              | Type                | Description   |
|-------------------|---------------------|---|
| destination       | snapmirror_endpoint | <p>Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> </ul> |
| exported_snapshot | string              | Snapshot copy exported to clients on destination.   |
| group_type        | string              | Specifies the group type of the top level SnapMirror relationship. The volume relationships are shown as <i>none</i> , the SVMDR relationships are shown as <i>svm_dr</i> , the Consistency Group relationships are shown as <i>consistency_group</i> , and the FlexGroup volume relationships are shown as <i>flexgroup</i> .  |
| healthy           | boolean             | Is the relationship healthy?  |

| Name                                    | Type   | Description  |
|---|--------|--|
| identity_preservation                   | string | Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type. This "identity_preservation" overrides the "identity_preservation" set on the SnapMirror relationship's policy.   |
| io_serving_copy                         | string | Specifies the sites serving I/O for the SnapMirror Business Continuity(SM-BC) relationship.  |
| lag_time                                | string | Time since the exported Snapshot copy was created.   |
| last_transfer_network_compression_ratio | string | Specifies the compression ratio achieved for the data sent over the wire with network compression enabled for the last successful transfer.  |
| last_transfer_type                      | string | Specifies the operation type of the last transfer that occurred on the relationship. The <i>initialize</i> transfer occurs when the relationship state changes from uninitialized to snapmirrored or in_sync. The <i>update</i> transfer occurs when the snapshots are transferred from the source endpoint to the destination endpoint as part of scheduled or manual update. The <i>resync</i> transfer occurs when the relationship state changes from broken_off to snapmirrored or in_sync. The <i>restore</i> transfer occurs when the snapshot is restored from a destination endpoint to another endpoint. |
| master_bias_activated_site              | string | Specifies the Master Bias Activated Site for the SnapMirror Business Continuity(SM-BC) relationship.   |

| Name                | Type    | Description   |
|---------------------|---------|---|
| policy              | policy  | Basic policy information of the relationship.   |
| preferred_site      | string  | Specifies the Primary Site of the SnapMirror Business Continuity(SM-BC) relationship.   |
| preserve            | boolean | Set to true on resync to preserve Snapshot copies on the destination that are newer than the latest common Snapshot copy. This property is applicable only for relationships with FlexVol volume or FlexGroup volume endpoints and when the PATCH state is being changed to "snapmirrored".   |
| quick_resync        | boolean | Set to true to reduce resync time by not preserving storage efficiency. This property is applicable only for relationships with FlexVol volume endpoints and SVMDR relationships when the PATCH state is being changed to "snapmirrored".   |
| recover_after_break | boolean | Set to true to recover from a failed SnapMirror break operation on a FlexGroup volume relationship. This restores all destination FlexGroup constituent volumes to the latest Snapshot copy, and any writes to the read-write constituents are lost. This property is applicable only for SnapMirror relationships with FlexGroup volume endpoints and when the PATCH state is being changed to "broken_off". |

| Name                | Type                                       | Description  |
|---------------------|--|--|
| restore             | boolean                                    | Set to true to create a relationship for restore. To trigger restore-transfer, use transfers POST on the restore relationship. SnapMirror relationships with the policy type "async" can be restored. SnapMirror relationships with the policy type "sync" cannot be restored.   |
| restore_to_snapshot | string                                     | Specifies the Snapshot copy to restore to on the destination during the break operation. This property is applicable only for SnapMirror relationships with FlexVol volume endpoints and when the PATCH state is being changed to "broken_off".  |
| source              | <a href="#">snapmirror_source_endpoint</a> | Source endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint. |

| Name  | Type   | Description   |
|-------|--------|---|
| state | string | <p>State of the relationship. To initialize the relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync". To break the relationship, PATCH the state to "broken_off" for relationships with a policy of type "async" or "sync".</p> <p>SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" cannot be "broken_off". To resync the relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync".</p> <p>SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" can be in "broken_off" state due to a failed attempt of SnapMirror failover. To pause the relationship, suspending further transfers, PATCH the state to "paused" for relationships with a policy of type "async" or "sync". SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" cannot be "paused". To resume transfers for a paused relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync". The entries "in_sync", "out_of_sync", "synchronizing", and "expanding" are only applicable to relationships with a policy of type "sync". A PATCH call on the state change only triggers the transition to the specified state. You must poll on the "state", "healthy" and "unhealthy_reason" properties using a GET request to determine if the transition is successful. To automatically initialize the relationship when specifying "create_destination" property, set the state to "snapmirrored" for</p> |

| Name                    | Type                    | Description   |
|-------------------------|-------------------------|---|
| svmdr_volumes           | array[svmdr_volumes]    | Specifies the list of constituent FlexVol volumes and FlexGroup volumes for an SVM DR SnapMirror relationship. FlexGroup constituents are not considered.   |
| throttle                | integer                 | Throttle, in KBs per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited.   |
| total_transfer_bytes    | integer                 | Cumulative bytes transferred for the relationship.  |
| total_transfer_duration | string                  | Indicates the cumulative duration of all transfers since the last aggregate relocation, takeover/giveback, or metrocluster switchover/switchback involving the node that hosts the relationship.  |
| transfer                | transfer                | Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.   |
| transfer_schedule       | transfer_schedule       | Schedule used to update asynchronous relationships. This "transfer_schedule" overrides the "transfer_schedule" set on the SnapMirror relationship's policy. To remove the "transfer_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror. |
| unhealthy_reason        | array[snapmirror_error] | Reason the relationship is not healthy. It is a concatenation of up to four levels of error messages.   |
| uuid                    | string                  | Unique identifier of the SnapMirror relationship.   |

## returned\_error

| Name      | Type                   | Description                                 |
|-----------|------------------------|---|
| arguments | array[error_arguments] | Message arguments                           |
| code      | string                 | Error code                                  |
| message   | string                 | Error message                               |
| target    | string                 | The target parameter that caused the error. |

## Create a SnapMirror relationship

POST /snapmirror/relationships

**Introduced In:** 9.6

Creates a SnapMirror relationship. This API can optionally provision the destination endpoint when it does not exist. This API must be executed on the cluster containing the destination endpoint unless the destination endpoint is being provisioned. When the destination endpoint is being provisioned, this API can also be executed from the cluster containing the source endpoint. Provisioning of the destination endpoint from the source cluster is supported for the FlexVol volume, FlexGroup volume and Application Consistency Group endpoints.

For SVM endpoints, provisioning the destination SVM endpoint is not supported from the source cluster. When the destination endpoint exists, the source SVM and the destination SVM must be in an SVM peer relationship. When provisioning the destination endpoint, the SVM peer relationship between the source SVM and the destination SVM is established as part of the destination provision, provided that the source SVM has SVM peering permissions for the destination cluster.

### Required properties

- `source.path` - Path to the source endpoint of the SnapMirror relationship.
- `destination.path` - Path to the destination endpoint of the SnapMirror relationship.
- `source.consistency_group_volumes` - List of FlexVol volumes of type "RW" that are constituents of an Application Consistency Group.
- `destination.consistency_group_volumes` - List of FlexVol volumes of type "DP" that are constituents of an Application Consistency Group.

### Recommended optional properties

- `policy.name` or `policy.uuid` - Policy governing the SnapMirror relationship.
- `state` - Set the state to "snapmirrored" to automatically initialize the relationship.
- `create_destination.enabled` - Enable this property to provision the destination endpoint.

## Default property values

If not specified in POST, the following default property values are assigned:

- `policy.name` - *Asynchronous*
- `restore` - *false*
- `create_destination.tiering.policy` - *snapshot\_only* (when `create_destination.tiering.supported` is *true* for FlexVol volume)
- `create_destination.tiering.policy` - *none* (when `create_destination.tiering.supported` is *true* for FlexGroup volume)
- `create_destination.storage_service.enforce_performance` - *false*
- `destination.ipspace` - *Default*
- `throttle` - *0*
- `backoff_level` - *high*

## Related ONTAP commands

- `snapmirror create`
- `snapmirror protect`

## Important notes

- The property "transfer\_schedule" if set on a SnapMirror relationship overrides the "transfer\_schedule" set on the policy being used with the SnapMirror relationship.
- The property "throttle" if set on a SnapMirror relationship overrides the "throttle" set on the policy being used with the SnapMirror relationship.
- The properties "transfer\_schedule" and "throttle" are not supported when "restore" is set to "true".
- The property "transfer\_schedule" cannot be set to null (no-quotes) during SnapMirror relationship POST.
- The property "throttle" is not supported when "create\_destination.enabled" is set to "true".
- The property "identity\_preservation" is applicable to only SnapMirror relationships with SVM endpoints and it indicates which configuration of the source SVM is replicated to the destination SVM.
- The property "backoff\_level" is not supported when "create\_destination.enabled" is set to "true".
- The property "backoff\_level" is only applicable for FlexVol SnapMirror relationships.

## Examples

The following examples show how to create FlexVol volumes, FlexGroup volumes, SVM and Application Consistency Group SnapMirror relationships. Note that the source SVM name must be the local name of the peer SVM.

Creating a FlexVol SnapMirror relationship of type XDP.

```
POST "/api/snapmirror/relationships/" '{ "source": { "path": "src_svm:src_vol" }, "destination": { "path": "dst_svm:dst_vol" } }'
```

Creating a FlexGroup SnapMirror relationship of type XDP.

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "src_svm:source_flexgrp"}, "destination": {"path": "dst_svm:dest_flexgrp"}}'
```

Creating a SVM SnapMirror relationship of type XDP.

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "src_svm:"}, "destination": {"path": "dst_svm:"}}'
```

Creating a SnapMirror relationship in order to restore from a destination.

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "src_svm:src_vol"}, "destination": {"path": "dst_svm:dst_vol"}, "restore": "true"}'
```

Provision the destination FlexVol volume endpoint and create a SnapMirror relationship of type XDP.

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "src_svm:src_vol"}, "destination": {"path": "dst_svm:dst_vol"}, "create_destination": {"enabled": "true"}}'
```

Provision the destination FlexVol volume endpoint on a Fabricpool with a tiering policy and create a SnapMirror relationship of type XDP.

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "src_svm:src_vol"}, "destination": {"path": "dst_svm:dst_vol"}, "create_destination": {"enabled": "true", "tiering": {"supported": "true", "policy": "auto"} }}'
```

Provision the destination FlexVol volume endpoint using storage service and create a SnapMirror relationship of type XDP.

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "src_svm:src_vol"}, "destination": {"path": "dst_svm:dst_vol"}, "create_destination": {"enabled": "true", "storage_service": {"enabled": "true", "name": "extreme", "enforce_performance": "true"} }}'
```

Provision the destination SVM endpoint and create a SnapMirror relationship of type XDP.

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "src_svm:", "cluster": {"name": "cluster_src"}}, "destination": {"path": "dst_svm:"}, "create_destination": {"enabled: "true" }}'
```

Create an asynchronous SnapMirror relationship with Application Consistency Group endpoint.

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "src_svm:/cg/cg_src_vol"}, "consistency_group_volumes": [{"name": "src_vol_1"}, {"name": "src_vol_2"}], "destination": {"path": "dst_svm:/cg/cg_dst_vol"}, "consistency_group_volumes": [{"name": "dst_vol_1"}, {"name": "dst_vol_2"}], "policy": "Asynchronous"}'
```

Provision the destination Application Consistency Group endpoint on a Fabricpool with a tiering policy, create an asynchronous SnapMirror relationship with a SnapMirror policy of type "async", and initialize the SnapMirror relationship with state as "snapmirrored".

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "src_svm:/cg/cg_src_vol"}, "consistency_group_volumes": [{"name": "src_vol_1"}, {"name": "src_vol_2"}], "destination": {"path": "dst_svm:/cg/cg_dst_vol"}, "consistency_group_volumes": [{"name": "dst_vol_1"}, {"name": "dst_vol_2"}], "create_destination": {"enabled": "true", "tiering": {"supported": "true"}}, "policy": "Asynchronous", "state": "snapmirrored"}'
```

Create a SnapMirror Business Continuity (SMBC) relationship with the Application Consistency Group endpoint.

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "src_svm:/cg/cg_src_vol"}, "consistency_group_volumes": [{"name": "src_vol_1"}, {"name": "src_vol_2"}], "destination": {"path": "dst_svm:/cg/cg_dst_vol"}, "consistency_group_volumes": [{"name": "dst_vol_1"}, {"name": "dst_vol_2"}], "policy": "AutomatedFailOver"}'
```

Provision the destination Application Consistency Group endpoint on a Fabricpool with a tiering policy, create a SnapMirror Business Continuity (SMBC) relationship with a SnapMirror policy of type "sync" and sync\_type of "automated\_failover", and initialize the SnapMirror relationship with state as "in\_sync".

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "src_svm:/cg/cg_src_vol", "consistency_group_volumes": [{"name": "src_vol_1"}, {"name": "src_vol_2"}]}, "destination": {"path": "dst_svm:/cg/cg_dst_vol", "consistency_group_volumes": [{"name": "dst_vol_1"}, {"name": "dst_vol_2"}]}, "create_destination": {"enabled": "true", "tiering": {"supported": "true"}}, "policy": "AutomatedFailOver", "state": "in_sync"}
```

Provision the destination Application Consistency Group endpoint with storage service, create a SnapMirror Business Continuity (SMBC) relationship with a SnapMirror policy of type "sync" and sync\_type of "automated\_failover", and initialize the SnapMirror relationship with state as "in\_sync".

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "src_svm:/cg/cg_src_vol", "consistency_group_volumes": [{"name": "src_vol_1"}, {"name": "src_vol_2"}]}, "destination": {"path": "dst_svm:/cg/cg_dst_vol", "consistency_group_volumes": [{"name": "dst_vol_1"}, {"name": "dst_vol_2"}]}, "create_destination": {"enabled": "true", "storage_service": {"enabled": "true", "name": "extreme", "enforce_performance": "true"}}, "policy": "AutomatedFailOver", "state": "in_sync"}
```

Provision the destination Application Consistency Group endpoint with storage service, create an asynchronous application consistency group relationship with a SnapMirror policy of type "async" and an async\_type of "XDPDefault", and initialize the SnapMirror relationship with state as "SnapMirrored".

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "src_svm:/cg/cg_src_vol", "consistency_group_volumes": [{"name": "src_vol_1"}, {"name": "src_vol_2"}]}, "destination": {"path": "dst_svm:/cg/cg_dst_vol", "consistency_group_volumes": [{"name": "dst_vol_1"}, {"name": "dst_vol_2"}]}, "create_destination": {"enabled": "true", "storage_service": {"enabled": "true", "name": "extreme", "enforce_performance": "true"}}, "policy": "XDPDefault", "state": "snapmirrored"}
```

Creating a FlexVol volume SnapMirror relationship of type XDP with transfer\_schedule and throttle.

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "src_svm:src_vol"}, "destination": {"path": "dst_svm:dst_vol"}, "transfer_schedule": {"uuid": "817500fa-092d-44c5-9c10-7b54f7b2f20a", "name": "5min"}, "throttle": 100}'
```

Creating an asynchronous SnapMirror relationship with backoff\_level set to medium.

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "src_svm:src_vol"}, "destination": {"path": "dst_svm:dst_vol"}, "backoff_level": "medium"}
```

## Learn more

- [DOC /snapmirror/relationships](#)

## Parameters

| Name           | Type    | In    | Required | Description   |
|----------------|---------|-------|----------|---|
| return_records | boolean | query | False    | <p>The default is false. If set to true, the records are returned.</p> <ul style="list-style-type: none"><li>• Default value:</li></ul> |

| Name           | Type    | In    | Required | Description  |
|----------------|---------|-------|----------|--|
| return_timeout | integer | query | False    | <p>The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul> |
| validate_only  | boolean | query | False    | <p>Validate the operation and its parameters, without actually performing the operation.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.7</li> </ul>  |

## Request Body

| Name          | Type   | Description  |
|---------------|--------|--|
| backoff_level | string | Specifies the SnapMirror backoff level due to Client Ops for FlexVol SnapMirror relationships. |

| Name               | Type  | Description   |
|--------------------|---|---|
| create_destination | <a href="#">snapmirror_destination_creation</a> | <p>Use this object to provision the destination endpoint when establishing a SnapMirror relationship for a FlexVol volume, FlexGroup volume, SVM, Consistency Group or ONTAP S3 Bucket. Given a source endpoint, the destination endpoint is provisioned in the SVM specified in the "destination.path" property. While protecting an SVM, the SVM destination endpoint can only be provisioned on the local cluster. To provision the SVM destination endpoint use the optional "source.cluster.name" property to specify the remote cluster name or use the optional "source.cluster.uuid" property to specify the remote cluster UUID. When "create_destination.enabled" option is specified while making a POST for a SnapMirror relationship, the relationship can be automatically initialized by setting the "state" either to "snapmirrored" when the policy is of type "async" or to "in_sync" when the policy is of type "sync". The "destination.path" property must specify the destination endpoint path. For example, for FlexVol volume and FlexGroup volume, the "destination.path" can be specified as &lt;destination-SVM-name:dp-volume-name&gt;, for SVM data protection, the "destination.path" must be specified as &lt;destination-SVM-name:&gt;; and for Consistency Group, the "destination.path" must be specified as &lt;destination-SVM-name:/cg/consistency-group-name&gt; along with the "destination.consistency_group_volumes" property to indicate the list of destination volumes of type "DP" in the Consistency Group. For a FlexVol volume, a FlexGroup volume, Consistency Group or a Bucket destination endpoint, the properties in this object can be specified either from the source or the destination cluster. For an SVM destination endpoint, the properties in this object can be specified from</p> |

| Name              | Type                                | Description   |
|-------------------|-------------------------------------|---|
| destination       | <a href="#">snapmirror_endpoint</a> | <p>Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> </ul> |
| exported_snapshot | string                              | Snapshot copy exported to clients on destination.   |
| group_type        | string                              | Specifies the group type of the top level SnapMirror relationship. The volume relationships are shown as <i>none</i> , the SVMDR relationships are shown as <i>svm_dr</i> , the Consistency Group relationships are shown as <i>consistency_group</i> , and the FlexGroup volume relationships are shown as <i>flexgroup</i> .  |
| healthy           | boolean                             | Is the relationship healthy?  |

| Name                                    | Type   | Description  |
|---|--------|--|
| identity_preservation                   | string | Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type. This "identity_preservation" overrides the "identity_preservation" set on the SnapMirror relationship's policy.   |
| io_serving_copy                         | string | Specifies the sites serving I/O for the SnapMirror Business Continuity(SM-BC) relationship.  |
| lag_time                                | string | Time since the exported Snapshot copy was created.   |
| last_transfer_network_compression_ratio | string | Specifies the compression ratio achieved for the data sent over the wire with network compression enabled for the last successful transfer.  |
| last_transfer_type                      | string | Specifies the operation type of the last transfer that occurred on the relationship. The <i>initialize</i> transfer occurs when the relationship state changes from uninitialized to snapmirrored or in_sync. The <i>update</i> transfer occurs when the snapshots are transferred from the source endpoint to the destination endpoint as part of scheduled or manual update. The <i>resync</i> transfer occurs when the relationship state changes from broken_off to snapmirrored or in_sync. The <i>restore</i> transfer occurs when the snapshot is restored from a destination endpoint to another endpoint. |
| master_bias_activated_site              | string | Specifies the Master Bias Activated Site for the SnapMirror Business Continuity(SM-BC) relationship.   |
| policy                                  | policy | Basic policy information of the relationship.  |

| Name                | Type    | Description   |
|---------------------|---------|---|
| preferred_site      | string  | Specifies the Primary Site of the SnapMirror Business Continuity(SM-BC) relationship.   |
| preserve            | boolean | Set to true on resync to preserve Snapshot copies on the destination that are newer than the latest common Snapshot copy. This property is applicable only for relationships with FlexVol volume or FlexGroup volume endpoints and when the PATCH state is being changed to "snapmirrored".   |
| quick_resync        | boolean | Set to true to reduce resync time by not preserving storage efficiency. This property is applicable only for relationships with FlexVol volume endpoints and SVMDR relationships when the PATCH state is being changed to "snapmirrored".   |
| recover_after_break | boolean | Set to true to recover from a failed SnapMirror break operation on a FlexGroup volume relationship. This restores all destination FlexGroup constituent volumes to the latest Snapshot copy, and any writes to the read-write constituents are lost. This property is applicable only for SnapMirror relationships with FlexGroup volume endpoints and when the PATCH state is being changed to "broken_off". |
| restore             | boolean | Set to true to create a relationship for restore. To trigger restore-transfer, use transfers POST on the restore relationship. SnapMirror relationships with the policy type "async" can be restored. SnapMirror relationships with the policy type "sync" cannot be restored.  |

| Name                | Type                                       | Description  |
|---------------------|--|--|
| restore_to_snapshot | string                                     | Specifies the Snapshot copy to restore to on the destination during the break operation. This property is applicable only for SnapMirror relationships with FlexVol volume endpoints and when the PATCH state is being changed to "broken_off".  |
| source              | <a href="#">snapmirror_source_endpoint</a> | Source endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint. |

| Name  | Type   | Description   |
|-------|--------|---|
| state | string | <p>State of the relationship. To initialize the relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync". To break the relationship, PATCH the state to "broken_off" for relationships with a policy of type "async" or "sync". SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" cannot be "broken_off". To resync the relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync". SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" can be in "broken_off" state due to a failed attempt of SnapMirror failover. To pause the relationship, suspending further transfers, PATCH the state to "paused" for relationships with a policy of type "async" or "sync". SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" cannot be "paused". To resume transfers for a paused relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync". The entries "in_sync", "out_of_sync", "synchronizing", and "expanding" are only applicable to relationships with a policy of type "sync". A PATCH call on the state change only triggers the transition to the specified state. You must poll on the "state", "healthy" and "unhealthy_reason" properties using a GET request to determine if the transition is successful. To automatically initialize the relationship when specifying "create_destination" property, set the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for</p> |

| Name                    | Type                                      | Description   |
|-------------------------|---|---|
| svmdr_volumes           | array[ <a href="#">svmdr_volumes</a> ]    | Specifies the list of constituent FlexVol volumes and FlexGroup volumes for an SVM DR SnapMirror relationship. FlexGroup constituents are not considered.   |
| throttle                | integer                                   | Throttle, in KBs per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited.   |
| total_transfer_bytes    | integer                                   | Cumulative bytes transferred for the relationship.  |
| total_transfer_duration | string                                    | Indicates the cumulative duration of all transfers since the last aggregate relocation, takeover/giveback, or metrocluster switchover/switchback involving the node that hosts the relationship.  |
| transfer                | <a href="#">transfer</a>                  | Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.   |
| transfer_schedule       | <a href="#">transfer_schedule</a>         | Schedule used to update asynchronous relationships. This "transfer_schedule" overrides the "transfer_schedule" set on the SnapMirror relationship's policy. To remove the "transfer_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror. |
| unhealthy_reason        | array[ <a href="#">snapmirror_error</a> ] | Reason the relationship is not healthy. It is a concatenation of up to four levels of error messages.   |
| uuid                    | string                                    | Unique identifier of the SnapMirror relationship.   |

## Example request

```
{  
    "backoff_level": "medium",  
    "create_destination": {  
        "bucket_retention": {  
            "default_period": "P10Y",  
            "mode": "governance"  
        },  
        "storage_service": {  
            "name": "string"  
        },  
        "tiering": {  
            "policy": "string"  
        }  
    },  
    "destination": {  
        "cluster": {  
            "name": "cluster1",  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "consistency_group_volumes": [  
            {  
                "name": "volume1",  
                "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"  
            }  
        ],  
        "ipspace": "Default",  
        "path": "svm1:volumel",  
        "svm": {  
            "name": "svm1",  
            "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
        }  
    },  
    "exported_snapshot": "string",  
    "group_type": "consistency_group",  
    "identity_preservation": "string",  
    "io_serving_copy": "C1_sti85-vsimg-ucs209a_cluster, C1_sti85-vsimg-  
ucs209c_cluster",  
    "lag_time": "PT8H35M42S",  
    "last_transfer_network_compression_ratio": 61,  
    "last_transfer_type": "initialize",  
    "master_bias_activated_site": "C1_sti85-vsimg-ucs209a_cluster",  
    "policy": {  
        "name": "Asynchronous",  
        "type": "string",  
    }  
}
```

```
        "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
    },
    "preferred_site": "C1_sti85-vsim-ucs209a_cluster",
    "restore_to_snapshot": "string",
    "source": {
        "cluster": {
            "name": "cluster1",
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
        },
        "consistency_group_volumes": [
            {
                "name": "volume1",
                "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
            }
        ],
        "path": "svm1:volume1",
        "svm": {
            "name": "svm1",
            "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
        }
    },
    "state": "snapmirrored",
    "svmdr_volumes": [
        {
            "name": "volume1"
        }
    ],
    "throttle": 0,
    "total_transfer_bytes": 1098210312,
    "total_transfer_duration": "PT3M21S",
    "transfer": {
        "end_time": "2020-12-02 21:36:19 -0500",
        "last_updated_time": "2023-09-14 18:39:19 -0400",
        "state": "string",
        "total_duration": "PT28M41S",
        "type": "initialize",
        "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
    },
    "transfer_schedule": {
        "name": "weekly",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "unhealthy_reason": [
        {
            "arguments": [],
            "code": "6621444",
            "message": "The volume is currently being replicated to another host. This may cause performance issues or data loss if the replication fails. Please check the replication status and configuration on the target host to ensure it is healthy and properly configured for the replication task. If the issue persists, please contact support for further assistance."}
    ]
}
```

```

    "message": "Failed to complete update operation on one or more
item relationships."
},
{
    "arguments": [],
    "code": "6621445",
    "message": "Group Update failed"
}
],
"uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
}

```

## Response

Status: 202, Accepted

| Name | Type     | Description |
|------|----------|-------------|
| job  | job_link |             |

## Example response

```
{
    "job": {
        "uuid": "string"
    }
}
```

## Headers

| Name     | Description                               | Type   |
|----------|---|--------|
| Location | Useful for tracking the resource location | string |

## Response

Status: 201, Created

## Error

Status: Default

## ONTAP Error Response Codes

| Error Code | Description  |
|------------|--|
| 1115545    | Access token has an invalid signature.   |
| 1115546    | Access token expired at a time.  |
| 1115547    | Access token not valid until a time.   |
| 1115548    | Access token is malformed.   |
| 1115549    | Internal error. Failed to validate access token.   |
| 6619441    | The source volume cannot be the same as the destination volume.  |
| 6619599    | Only the "none" tiering policy is supported when creating a destination volume with "snapshot_locking_enabled" set to true or if "snapshot_locking_enabled" is set to true on the source volume. |
| 6619637    | Relationship with specified destination volume already exists.   |
| 6619699    | Schedule not found.  |
| 6620374    | Internal error. Failed to get SVM information.   |
| 6620478    | Internal error. Failed to check SnapMirror capability.   |
| 6620654    | Invalid SnapMirror Consistency Group name.   |
| 6621458    | The destination Consistency Group is the source of a SnapMirror Synchronous (SM-S) relationship. Sources of SM-S relationships cannot be the destination of any other SnapMirror relationship.   |
| 6621782    | A property of the policy is not valid for relationships between these endpoints.   |
| 13304080   | Specified UUID and name do not match.  |
| 13304082   | Specified properties are mutually exclusive.   |
| 13304083   | The specified property is not supported because all nodes in the cluster are not capable of supporting the property.<private></private>  |
| 13304093   | The property specified is not supported for the specified relationships.   |
| 13304098   | This SnapMirror policy is not supported for SnapMirror relationships with SnapLock volumes.  |
| 13304099   | SnapLock Compliance Clock is not running on all nodes in the destination cluster.  |

| Error Code | Description  |
|------------|--|
| 13304108   | Schedule not found in the Administrative SVM or the SVM for the relationship.  |
| 13304110   | Policy has "snapshot_lock_mode" set to "compliance" or "enterprise" and the cloud provider associated with this object store relationship requires an effective cluster version of ONTAP 9.12.1 or later to support Snapshot copy locking. |
| 13304112   | File restore from a Consistency Group asynchronous SnapMirror relationship endpoint is not supported.  |
| 13304117   | Policy has "snapshot_lock_mode" set to "compliance" or "enterprise" and FlexGroup volume object store relationships require an effective cluster version of ONTAP 9.13.1 or later to support Snapshot copy locking.<\private>              |
| 13304123   | SnapMirror object store endpoint name for a FlexVol volume relationship exceeds the character limit.   |
| 13304124   | SnapMirror object store endpoint name for a FlexGroup volume relationship exceeds the character limit.<\private>   |
| 13304132   | Creating a destination endpoint is not supported with the "backoff_level" property.  |
| 53411897   | The specified source volumes do not match the volumes contained in the source consistency group.   |

Also see the table of common errors in the [Response body](#) overview section of this documentation.

## Definitions

## See Definitions

href

| Name | Type   | Description |
|------|--------|-------------|
| href | string |             |

\_links

error\_arguments

| Name    | Type   | Description      |
|---------|--------|------------------|
| code    | string | Argument code    |
| message | string | Message argument |

error

SnapMirror Consistency Group failover error message.

| Name      | Type                   | Description       |
|-----------|------------------------|-------------------|
| arguments | array[error_arguments] | Message arguments |
| code      | string                 | Error code        |
| message   | string                 | Error message     |

status

| Name    | Type   | Description                                   |
|---------|--------|---|
| code    | string | Status code                                   |
| message | string | SnapMirror Consistency Group failover status. |

snapmirror\_consistency\_group\_failover

SnapMirror Consistency Group failover information. The SnapMirror Consistency Group failover can be a planned or an unplanned operation. Only active SnapMirror Consistency Group failover operation progress can be monitored using this object. In case of an error during the failover operation, the property "consistency\_group\_failover.error" holds the reason for the error. ONTAP automatically retries any failed SnapMirror Consistency Group failover operation.

| Name   | Type                   | Description                                  |
|--------|------------------------|--|
| state  | string                 | SnapMirror Consistency Group failover state. |
| status | <a href="#">status</a> |  |
| type   | string                 | SnapMirror Consistency Group failover type.  |

## bucket\_retention

Specifies the retention-mode and default retention period configured on the destination bucket.

| Name           | Type   | Description  |
|----------------|--------|--|
| default_period | string | Specifies the default retention period that is applied to objects while committing them to the WORM state without an associated retention period. The retention period can be in years, or days. The retention period value represents a duration and must be specified in the ISO-8601 duration format. A period specified for years and days is represented in the ISO-8601 format as "P<num>Y" and "P<num>D" respectively, for example "P10Y" represents a duration of 10 years. The period string must contain only a single time element that is, either years, or days. A duration which combines different periods is not supported, for example "P1Y10D" is not supported.</num></num> |
| mode           | string | The lock mode of the bucket. compliance – A SnapLock Compliance (SLC) bucket provides the highest level of WORM protection and an administrator cannot destroy a compliance bucket if it contains unexpired WORM objects. governance – An administrator can delete a Governance bucket. no_lock – Indicates the bucket does not support object locking.  |

## storage\_service

| Name                | Type    | Description   |
|---------------------|---------|---|
| enabled             | boolean | This property indicates whether to create the destination endpoint using storage service.   |
| enforce_performance | boolean | Optional property to enforce storage service performance on the destination endpoint. This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints.   |
| name                | string  | <p>Optional property to specify the storage service name for the destination endpoint. This property is considered when the property "create_destination.storage_service.enabled" is set to "true". When the property "create_destination.storage_service.enabled" is set to "true" and the "create_destination.storage_service.name" for the endpoint is not specified, then ONTAP selects the highest storage service available on the cluster to provision the destination endpoint. This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints.</p> <ul style="list-style-type: none"> <li>• enum: ["extreme", "performance", "value"]</li> <li>• Introduced in: 9.6</li> <li>• x-nullable: true</li> </ul> |

## tiering

| Name   | Type   | Description   |
|--------|--------|---|
| policy | string | <p>Optional property to specify the destination endpoint's tiering policy when "create_destination.tiering.supported" is set to "true". This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints. This property determines whether the user data blocks of the destination endpoint in a FabricPool will be tiered to the cloud store when they become cold. FabricPool combines flash (performance tier) with a cloud store into a single aggregate. Temperature of the destination endpoint volume blocks increases if they are accessed frequently and decreases when they are not. all – This policy allows tiering of both destination endpoint Snapshot copies and the user transferred data blocks to the cloud store as soon as possible by ignoring the temperature on the volume blocks. This tiering policy is not applicable for Consistency Group destination endpoints or for synchronous relationships. auto – This policy allows tiering of both destination endpoint Snapshot copies and the active file system user data to the cloud store none – Destination endpoint volume blocks will not be tiered to the cloud store. snapshot_only – This policy allows tiering of only the destination endpoint volume Snapshot copies not associated with the active file system. The default tiering policy is "snapshot_only" for a FlexVol volume and "none" for a FlexGroup volume.</p> |

| Name      | Type    | Description   |
|-----------|---------|---|
| supported | boolean | Optional property to enable provisioning of the destination endpoint volumes on FabricPool aggregates. This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints. Only FabricPool aggregates are used if this property is set to "true" and only non FabricPool aggregates are used if this property is set to "false". Tiering support for a FlexGroup volume can be changed by moving all of the constituents to the required aggregates. Note that in order to tier data, not only do the destination endpoint volumes need to support tiering by using FabricPools, the "create_destination.tiering.policy" must not be "none". A destination endpoint that uses FabricPools but has a tiering "policy" of "none" supports tiering but will not tier any data. |

#### snapmirror\_destination\_creation

Use this object to provision the destination endpoint when establishing a SnapMirror relationship for a FlexVol volume, FlexGroup volume, SVM, Consistency Group or ONTAP S3 Bucket. Given a source endpoint, the destination endpoint is provisioned in the SVM specified in the "destination.path" property. While protecting an SVM, the SVM destination endpoint can only be provisioned on the local cluster. To provision the SVM destination endpoint use the optional "source.cluster.name" property to specify the remote cluster name or use the optional "source.cluster.uuid" property to specify the remote cluster UUID. When "create\_destination.enabled" option is specified while making a POST for a SnapMirror relationship, the relationship can be automatically initialized by setting the "state" either to "snapmirrored" when the policy is of type "async" or to "in\_sync" when the policy is of type "sync". The "destination.path" property must specify the destination endpoint path. For example, for FlexVol volume and FlexGroup volume, the "destination.path" can be specified as <destination-svm-name:dp-volume-name>, for SVM data protection, the "destination.path" must be specified as &lt;destination-svm-name:&gt;, and for Consistency Group, the "destination.path" must be specified as &lt;destination-svm-name:&gt; along with the "destination.consistency\_group\_volumes" property to indicate the list of destination volumes of type "DP" in the Consistency Group. For a FlexVol volume, a FlexGroup volume, Consistency Group or a Bucket destination endpoint, the properties in this object can be specified either from the source or the destination cluster. For an SVM destination endpoint, the properties in this object can be specified from the destination cluster only. This object is not supported for non ONTAP endpoints. While protecting a S3 Bucket, the optional "size" property can be used to create ONTAP S3 Bucket destination endpoint of the specified size.&lt;/destination-svm-name:&gt;&lt;/destination-svm-name:&gt;</destination-svm-name:dp-volume-name>

| Name                     | Type                             | Description  |
|--------------------------|----------------------------------|--|
| bucket_retention         | <a href="#">bucket_retention</a> | Specifies the retention-mode and default retention period configured on the destination bucket.  |
| enabled                  | boolean                          | Optional property to create the destination endpoint when establishing a SnapMirror relationship. It is assumed to be "false" if no other property is set and assumed to be "true" if any other property is set.   |
| size                     | integer                          | Optional property to specify the size of destination endpoint in bytes. This property is applicable only to ONTAP S3 Bucket endpoints. The minimum size for S3 bucket is 80MB and maximum size is 64TB. If not specified, system will create destination with default size of 800GB. <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> <li>• x-nullable: true</li> </ul> |
| snapshot_locking_enabled | boolean                          | Optional property to create the destination endpoint with Snapshot copy locking enabled when establishing a SnapMirror relationship. This property is applicable to FlexVol volumes and FlexGroup volumes.   |
| storage_service          | <a href="#">storage_service</a>  |  |
| tiering                  | <a href="#">tiering</a>          |  |

cluster

| Name | Type   | Description |
|------|--------|-------------|
| name | string |             |
| uuid | string |             |

consistency\_group\_volumes

| Name | Type   | Description  |
|------|--------|--|
| name | string | The name of the volume.  |
| uuid | string | <p>Unique identifier of the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move.</p> <ul style="list-style-type: none"> <li>example: 028baa66-41bd-11e9-81d5-00a0986138f7</li> <li>x-ntap-createOnly: true</li> <li>Introduced in: 9.8</li> <li>x-nullable: true</li> </ul> |

## svm

SVM, applies only to SVM-scoped objects.

| Name | Type   | Description   |
|------|--------|---|
| name | string | The name of the SVM. This field cannot be specified in a PATCH method.              |
| uuid | string | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

## snapmirror\_endpoint

Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.

| Name                      | Type                             | Description   |
|---------------------------|----------------------------------|---|
| cluster                   | cluster                          |   |
| consistency_group_volumes | array[consistency_group_volumes] | Mandatory property for a Consistency Group endpoint. Specifies the list of FlexVol volumes for a Consistency Group. |

| Name    | Type   | Description  |
|---------|--------|--|
| ipspace | string | Optional property to specify the IPspace of the SVM.   |
| path    | string | ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name<br>ONTAP S3 - svm1:/bucket/bucket1 NON-ONTAP - objstore1:/objstore <ul style="list-style-type: none"> <li>example: svm1:volume1</li> <li>Introduced in: 9.6</li> <li>x-nullable: true</li> </ul> |
| svm     | svm    | SVM, applies only to SVM-scoped objects.   |

## policy

Basic policy information of the relationship.

| Name | Type   | Description                                 |
|------|--------|---|
| name | string | Name of the SnapMirror policy.              |
| type | string |   |
| uuid | string | Unique identifier of the SnapMirror policy. |

## snapmirror\_source\_endpoint

Source endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.

| Name                      | Type                             | Description   |
|---------------------------|----------------------------------|---|
| cluster                   | cluster                          |   |
| consistency_group_volumes | array[consistency_group_volumes] | Mandatory property for a Consistency Group endpoint. Specifies the list of FlexVol volumes for a Consistency Group. |

| Name | Type   | Description   |
|------|--------|---|
| path | string | ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name<br>ONTAP S3 - svm1:/bucket/bucket1 NON-ONTAP - objstore1:/objstore <ul style="list-style-type: none"> <li>• example: svm1:volume1</li> <li>• Introduced in: 9.14</li> <li>• x-nullable: true</li> </ul> |
| svm  | svm    | SVM, applies only to SVM-scoped objects.  |

#### svmdr\_volumes

| Name | Type   | Description             |
|------|--------|-------------------------|
| name | string | The name of the volume. |

#### transfer

Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.

| Name              | Type    | Description  |
|-------------------|---------|--|
| bytes_transferred | integer | Total bytes transferred in the current or last successful transfer.            |
| end_time          | string  | End time of the last transfer.   |
| last_updated_time | string  | Last updated time of the bytes transferred in the current transfer.            |
| state             | string  |  |
| total_duration    | string  | Elapsed time to transfer all Snapshot copies for the last successful transfer. |

| Name | Type   | Description  |
|------|--------|--|
| type | string | Specifies the operation type of the current transfer on the relationship. The <i>initialize</i> transfer occurs when the relationship state changes from "uninitialized" to "snapmirrored" or "in_sync". The <i>update</i> transfer occurs when Snapshot copies are being transferred from the source endpoint to the destination endpoint as part of a scheduled or manual update. The <i>resync</i> transfer occurs when the relationship state changes from "broken_off" to "snapmirrored" or "in_sync". The <i>restore</i> transfer occurs when a Snapshot copy is being restored from a destination endpoint to another endpoint. |
| uuid | string | Transfer UUID. This property is applicable only for active transfers.  |

### transfer\_schedule

Schedule used to update asynchronous relationships. This "transfer\_schedule" overrides the "transfer\_schedule" set on the SnapMirror relationship's policy. To remove the "transfer\_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror.

| Name | Type   | Description       |
|------|--------|-------------------|
| name | string | Job schedule name |
| uuid | string | Job schedule UUID |

### snapmirror\_error

SnapMirror error

| Name      | Type          | Description   |
|-----------|---------------|---|
| arguments | array[string] | Arguments present in the error message encountered. |
| code      | string        | Error code  |
| message   | string        | Error message                                       |

## `snapmirror_relationship`

SnapMirror relationship information. The SnapMirror relationship can be either "async" or "sync" based on the type of SnapMirror policy associated with the relationship. The source and destination endpoints of a SnapMirror relationship must be of the same type, for example, if the source endpoint is a FlexVol volume then the destination endpoint must be a FlexVol volume. The SnapMirror policy type "async" can be used when the SnapMirror relationship has FlexVol volume or FlexGroup volume or SVM as the endpoint. The SnapMirror policy type "sync" can be used when the SnapMirror relationship has FlexVol volume as the endpoint. The SnapMirror policy type "sync" with "sync\_type" as "automated\_failover" can be used when the SnapMirror relationship has Consistency Group as the endpoint.

| Name          | Type   | Description  |
|---------------|--------|--|
| backoff_level | string | Specifies the SnapMirror backoff level due to Client Ops for FlexVol SnapMirror relationships. |

| Name               | Type  | Description  |
|--------------------|---|--|
| create_destination | <a href="#">snapmirror_destination_creation</a> | <p>Use this object to provision the destination endpoint when establishing a SnapMirror relationship for a FlexVol volume, FlexGroup volume, SVM, Consistency Group or ONTAP S3 Bucket. Given a source endpoint, the destination endpoint is provisioned in the SVM specified in the "destination.path" property. While protecting an SVM, the SVM destination endpoint can only be provisioned on the local cluster. To provision the SVM destination endpoint use the optional "source.cluster.name" property to specify the remote cluster name or use the optional "source.cluster.uuid" property to specify the remote cluster UUID.</p> <p>When "create_destination.enabled" option is specified while making a POST for a SnapMirror relationship, the relationship can be automatically initialized by setting the "state" either to "snapmirrored" when the policy is of type "async" or to "in_sync" when the policy is of type "sync". The "destination.path" property must specify the destination endpoint path. For example, for FlexVol volume and FlexGroup volume, the "destination.path" can be specified as &lt;destination-SVM-name:dp-volume-name&gt;, for SVM data protection, the "destination.path" must be specified as &lt;destination-SVM-name:&gt;, and for Consistency Group, the "destination.path" must be specified as &lt;destination-SVM-name:/cg/consistency-group-name&gt; along with the "destination.consistency_group_volumes" property to indicate the list of destination volumes of type "DP" in the Consistency Group. For a FlexVol volume, a FlexGroup volume, Consistency Group or a Bucket destination endpoint, the properties in this object can be specified either from the source or the destination</p> |

| Name              | Type                | Description   |
|-------------------|---------------------|---|
| destination       | snapmirror_endpoint | <p>Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> </ul> |
| exported_snapshot | string              | Snapshot copy exported to clients on destination.   |
| group_type        | string              | Specifies the group type of the top level SnapMirror relationship. The volume relationships are shown as <i>none</i> , the SVMDR relationships are shown as <i>svm_dr</i> , the Consistency Group relationships are shown as <i>consistency_group</i> , and the FlexGroup volume relationships are shown as <i>flexgroup</i> .  |
| healthy           | boolean             | Is the relationship healthy?  |

| Name                                    | Type   | Description  |
|---|--------|--|
| identity_preservation                   | string | Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type. This "identity_preservation" overrides the "identity_preservation" set on the SnapMirror relationship's policy.   |
| io_serving_copy                         | string | Specifies the sites serving I/O for the SnapMirror Business Continuity(SM-BC) relationship.  |
| lag_time                                | string | Time since the exported Snapshot copy was created.   |
| last_transfer_network_compression_ratio | string | Specifies the compression ratio achieved for the data sent over the wire with network compression enabled for the last successful transfer.  |
| last_transfer_type                      | string | Specifies the operation type of the last transfer that occurred on the relationship. The <i>initialize</i> transfer occurs when the relationship state changes from uninitialized to snapmirrored or in_sync. The <i>update</i> transfer occurs when the snapshots are transferred from the source endpoint to the destination endpoint as part of scheduled or manual update. The <i>resync</i> transfer occurs when the relationship state changes from broken_off to snapmirrored or in_sync. The <i>restore</i> transfer occurs when the snapshot is restored from a destination endpoint to another endpoint. |
| master_bias_activated_site              | string | Specifies the Master Bias Activated Site for the SnapMirror Business Continuity(SM-BC) relationship.   |

| Name                | Type    | Description   |
|---------------------|---------|---|
| policy              | policy  | Basic policy information of the relationship.   |
| preferred_site      | string  | Specifies the Primary Site of the SnapMirror Business Continuity(SM-BC) relationship.   |
| preserve            | boolean | Set to true on resync to preserve Snapshot copies on the destination that are newer than the latest common Snapshot copy. This property is applicable only for relationships with FlexVol volume or FlexGroup volume endpoints and when the PATCH state is being changed to "snapmirrored".   |
| quick_resync        | boolean | Set to true to reduce resync time by not preserving storage efficiency. This property is applicable only for relationships with FlexVol volume endpoints and SVMDR relationships when the PATCH state is being changed to "snapmirrored".   |
| recover_after_break | boolean | Set to true to recover from a failed SnapMirror break operation on a FlexGroup volume relationship. This restores all destination FlexGroup constituent volumes to the latest Snapshot copy, and any writes to the read-write constituents are lost. This property is applicable only for SnapMirror relationships with FlexGroup volume endpoints and when the PATCH state is being changed to "broken_off". |

| Name                | Type                                       | Description  |
|---------------------|--|--|
| restore             | boolean                                    | Set to true to create a relationship for restore. To trigger restore-transfer, use transfers POST on the restore relationship. SnapMirror relationships with the policy type "async" can be restored. SnapMirror relationships with the policy type "sync" cannot be restored.   |
| restore_to_snapshot | string                                     | Specifies the Snapshot copy to restore to on the destination during the break operation. This property is applicable only for SnapMirror relationships with FlexVol volume endpoints and when the PATCH state is being changed to "broken_off".  |
| source              | <a href="#">snapmirror_source_endpoint</a> | Source endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint. |

| Name  | Type   | Description   |
|-------|--------|---|
| state | string | <p>State of the relationship. To initialize the relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync". To break the relationship, PATCH the state to "broken_off" for relationships with a policy of type "async" or "sync".</p> <p>SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" cannot be "broken_off". To resync the relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync".</p> <p>SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" can be in "broken_off" state due to a failed attempt of SnapMirror failover. To pause the relationship, suspending further transfers, PATCH the state to "paused" for relationships with a policy of type "async" or "sync". SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" cannot be "paused". To resume transfers for a paused relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync". The entries "in_sync", "out_of_sync", "synchronizing", and "expanding" are only applicable to relationships with a policy of type "sync". A PATCH call on the state change only triggers the transition to the specified state. You must poll on the "state", "healthy" and "unhealthy_reason" properties using a GET request to determine if the transition is successful. To automatically initialize the relationship when specifying "create_destination" property, set the state to "snapmirrored" for</p> |

| Name                    | Type                    | Description   |
|-------------------------|-------------------------|---|
| svmdr_volumes           | array[svmdr_volumes]    | Specifies the list of constituent FlexVol volumes and FlexGroup volumes for an SVM DR SnapMirror relationship. FlexGroup constituents are not considered.   |
| throttle                | integer                 | Throttle, in KBs per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited.   |
| total_transfer_bytes    | integer                 | Cumulative bytes transferred for the relationship.  |
| total_transfer_duration | string                  | Indicates the cumulative duration of all transfers since the last aggregate relocation, takeover/giveback, or metrocluster switchover/switchback involving the node that hosts the relationship.  |
| transfer                | transfer                | Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.   |
| transfer_schedule       | transfer_schedule       | Schedule used to update asynchronous relationships. This "transfer_schedule" overrides the "transfer_schedule" set on the SnapMirror relationship's policy. To remove the "transfer_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror. |
| unhealthy_reason        | array[snapmirror_error] | Reason the relationship is not healthy. It is a concatenation of up to four levels of error messages.   |
| uuid                    | string                  | Unique identifier of the SnapMirror relationship.   |

## job\_link

| Name | Type   | Description   |
|------|--------|---|
| uuid | string | The UUID of the asynchronous job that is triggered by a POST, PATCH, or DELETE operation. |

## returned\_error

| Name      | Type                   | Description                                 |
|-----------|------------------------|---|
| arguments | array[error_arguments] | Message arguments                           |
| code      | string                 | Error code                                  |
| message   | string                 | Error message                               |
| target    | string                 | The target parameter that caused the error. |

## Delete a SnapMirror relationship

`DELETE /snapmirror/relationships/{uuid}`

**Introduced In:** 9.6

Deletes a SnapMirror relationship.

### Important notes

- The "destination\_only", "source\_only", and "source\_info\_only" flags are mutually exclusive. If no flag is specified, the relationship is deleted from both the source and destination and all common Snapshot copies between the source and destination are also deleted.
- For a restore relationship, the call must be executed on the cluster containing the destination endpoint without specifying the destination\_only, source\_only, or source\_info\_only parameters.
- Additionally, ensure that there are no ongoing transfers on a restore relationship before calling this API.
- The "failover", "force-failover" and "fallback" query parameters are only applicable for SVM-DR SnapMirror relationships.
- When a SnapMirror relationship associated with a pair of source and destination Consistency Groups is deleted, the corresponding Consistency Groups on the source and destination clusters are not automatically deleted and remain in place.
- The "delete\_lun\_maps\_in\_destination" query parameter is applicable only for SM-BC relationships.

### Related ONTAP commands

- `snapmirror delete`

- snapmirror release

## Examples

The following examples show how to delete the relationship from both the source and destination, the destination only, and the source only.

Deleting the relationship from both the source and destination. This API must be run on the cluster containing the destination endpoint.

```
DELETE "/api/snapmirror/relationships/4512b2d2-fd60-11e8-8929-005056bbfe52"
```

Deleting the relationship on the destination only. This API must be run on the cluster containing the destination endpoint.

```
DELETE "/api/snapmirror/relationships/fd1e0697-02ba-11e9-acc7-005056a7697f/?destination_only=true"
```

Deleting the relationship on the source only. This API must be run on the cluster containing the source endpoint.

```
DELETE "/api/snapmirror/relationships/93e828ba-02bc-11e9-acc7-005056a7697f/?source_only=true"
```

Deleting the source information only. This API must be run on the cluster containing the source endpoint. This does not delete the common Snapshot copies between the source and destination.

```
DELETE "/api/snapmirror/relationships/caf545a2-fc60-11e8-aa13-005056a707ff/?source_info_only=true"
```

Deleting the relationship from source and destination cluster along with deleting the LUN maps for the volumes of the CG in destination cluster. This API must be run on the cluster containing the destination endpoint.

```
DELETE "/api/snapmirror/relationships/feda8f5e-e29e-11ed-94aa-005056a78ce2/?delete_lun_maps_in_destination=true"
```

## Learn more

- [DOC /snapmirror/relationships](#)

## Parameters

| Name             | Type    | In    | Required | Description  |
|------------------|---------|-------|----------|--|
| uuid             | string  | path  | True     | SnapMirror relationship UUID   |
| destination_only | boolean | query | False    | Deletes a relationship on the destination only. This parameter is applicable only when the call is executed on the cluster that contains the destination endpoint.   |
| source_only      | boolean | query | False    | Deletes a relationship on the source only. This parameter is applicable only when the call is executed on the cluster that contains the source endpoint. In the case of a relationship that uses SnapMirror policy of type 'continuous', this parameter is used to delete the source end of the relationship while ignoring errors when the destination endpoint is not reachable. |
| source_info_only | boolean | query | False    | Deletes relationship information on the source only. This parameter is applicable only when the call is executed on the cluster that contains the source endpoint.   |

| Name                           | Type    | In    | Required | Description  |
|--------------------------------|---------|-------|----------|--|
| delete_lun_maps_in_destination | boolean | query | False    | <p>Deletes LUN mapping for the volumes of the CG in destination.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.14</li> <li>• Default value:</li> </ul>   |
| return_timeout                 | integer | query | False    | <p>The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul> |

## Response

Status: 200, Ok

| Name | Type     | Description |
|------|----------|-------------|
| job  | job_link |             |

## Example response

```
{
  "job": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "uuid": "string"
  }
}
```

## Response

Status: 202, Accepted

## Error

Status: Default

## ONTAP Error Response codes

| Error code | Description  |
|------------|--|
| 13303825   | Could not retrieve information for the SnapMirror policy type          |
| 13303814   | Could not retrieve the source or destination SVM UUID                  |
| 13303815   | Could not retrieve information for the peer cluster                    |
| 13303822   | SnapMirror release has failed  |
| 13303813   | SnapMirror release was successful but delete has failed                |
| 13303854   | Cleanup of restore relationship failed                                 |
| 13303855   | DELETE call on a restore relationship does not support the given flags |
| 13303865   | Deleting the specified SnapMirror policy is not supported.             |

| Error code | Description   |
|------------|---|
| 6619715    | Modification of relationship is in progress. Retry the command after a few minutes. |

| Name  | Type                           | Description |
|-------|--------------------------------|-------------|
| error | <a href="#">returned_error</a> |             |

### Example error

```
{
  "error": {
    "arguments": [
      {
        "code": "string",
        "message": "string"
      }
    ],
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```

### Definitions

## See Definitions

href

| Name | Type   | Description |
|------|--------|-------------|
| href | string |             |

\_links

| Name | Type | Description |
|------|------|-------------|
| self | href |             |

job\_link

| Name   | Type   | Description   |
|--------|--------|---|
| _links | _links |   |
| uuid   | string | The UUID of the asynchronous job that is triggered by a POST, PATCH, or DELETE operation. |

error\_arguments

| Name    | Type   | Description      |
|---------|--------|------------------|
| code    | string | Argument code    |
| message | string | Message argument |

returned\_error

| Name      | Type                   | Description                                 |
|-----------|------------------------|---|
| arguments | array[error_arguments] | Message arguments                           |
| code      | string                 | Error code                                  |
| message   | string                 | Error message                               |
| target    | string                 | The target parameter that caused the error. |

## Retrieve a SnapMirror relationship

GET /snapmirror/relationships/{uuid}

## Introduced In: 9.6

Retrieves a SnapMirror relationship.

### Related ONTAP commands

- snapmirror show
- snapmirror list-destinations

### Expensive properties

- source.consistency\_group\_volumes.name
- destination.consistency\_group\_volumes.name

### Example

```
GET "/api/snapmirror/relationships/caf545a2-fc60-11e8-aa13-005056a707ff/"
```

### Learn more

- [DOC /snapmirror/relationships](#)

### Parameters

| Name                   | Type          | In    | Required | Description   |
|------------------------|---------------|-------|----------|---|
| uuid                   | string        | path  | True     | SnapMirror relationship UUID                            |
| list_destinations_only | boolean       | query | False    | Set to true to show relationships from the source only. |
| fields                 | array[string] | query | False    | Specify the fields to return.                           |

### Response

```
Status: 200, Ok
```

| Name          | Type                   | Description  |
|---------------|------------------------|--|
| _links        | <a href="#">_links</a> |  |
| backoff_level | string                 | Specifies the SnapMirror backoff level due to Client Ops for FlexVol SnapMirror relationships. |

| Name                       | Type  | Description   |
|----------------------------|---|---|
| consistency_group_failover | <a href="#">snapmirror_consistency_group_failover</a> | <p>SnapMirror Consistency Group failover information. The SnapMirror Consistency Group failover can be a planned or an unplanned operation. Only active SnapMirror Consistency Group failover operation progress can be monitored using this object. In case of an error during the failover operation, the property "consistency_group_failover.error" holds the reason for the error. ONTAP automatically retries any failed SnapMirror Consistency Group failover operation.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.8</li> <li>• readOnly: 1</li> </ul>   |
| destination                | <a href="#">snapmirror_endpoint</a>                   | <p>Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> </ul> |
| exported_snapshot          | string  | Snapshot copy exported to clients on destination.   |

| Name                                    | Type    | Description  |
|---|---------|--|
| group_type                              | string  | Specifies the group type of the top level SnapMirror relationship. The volume relationships are shown as <i>none</i> , the SVMDR relationships are shown as <i>svm_dr</i> , the Consistency Group relationships are shown as <i>consistency_group</i> , and the FlexGroup volume relationships are shown as <i>flexgroup</i> . |
| healthy                                 | boolean | Is the relationship healthy?   |
| identity_preservation                   | string  | Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type. This "identity_preservation" overrides the "identity_preservation" set on the SnapMirror relationship's policy.                                       |
| io_serving_copy                         | string  | Specifies the sites serving I/O for the SnapMirror Business Continuity(SM-BC) relationship.  |
| lag_time                                | string  | Time since the exported Snapshot copy was created.   |
| last_transfer_network_compression_ratio | string  | Specifies the compression ratio achieved for the data sent over the wire with network compression enabled for the last successful transfer.  |

| Name                       | Type    | Description  |
|----------------------------|---------|--|
| last_transfer_type         | string  | Specifies the operation type of the last transfer that occurred on the relationship. The <i>initialize</i> transfer occurs when the relationship state changes from uninitialized to snapmirrored or in_sync. The <i>update</i> transfer occurs when the snapshots are transferred from the source endpoint to the destination endpoint as part of scheduled or manual update. The <i>resync</i> transfer occurs when the relationship state changes from broken_off to snapmirrored or in_sync. The <i>restore</i> transfer occurs when the snapshot is restored from a destination endpoint to another endpoint. |
| master_bias_activated_site | string  | Specifies the Master Bias Activated Site for the SnapMirror Business Continuity(SM-BC) relationship.   |
| policy                     | policy  | Basic policy information of the relationship.  |
| preferred_site             | string  | Specifies the Primary Site of the SnapMirror Business Continuity(SM-BC) relationship.  |
| preserve                   | boolean | Set to true on resync to preserve Snapshot copies on the destination that are newer than the latest common Snapshot copy. This property is applicable only for relationships with FlexVol volume or FlexGroup volume endpoints and when the PATCH state is being changed to "snapmirrored".  |
| quick_resync               | boolean | Set to true to reduce resync time by not preserving storage efficiency. This property is applicable only for relationships with FlexVol volume endpoints and SVMDR relationships when the PATCH state is being changed to "snapmirrored".  |

| Name                | Type                                       | Description   |
|---------------------|--|---|
| recover_after_break | boolean                                    | Set to true to recover from a failed SnapMirror break operation on a FlexGroup volume relationship. This restores all destination FlexGroup constituent volumes to the latest Snapshot copy, and any writes to the read-write constituents are lost. This property is applicable only for SnapMirror relationships with FlexGroup volume endpoints and when the PATCH state is being changed to "broken_off". |
| restore             | boolean                                    | Set to true to create a relationship for restore. To trigger restore-transfer, use transfers POST on the restore relationship. SnapMirror relationships with the policy type "async" can be restored. SnapMirror relationships with the policy type "sync" cannot be restored.  |
| restore_to_snapshot | string                                     | Specifies the Snapshot copy to restore to on the destination during the break operation. This property is applicable only for SnapMirror relationships with FlexVol volume endpoints and when the PATCH state is being changed to "broken_off".   |
| source              | <a href="#">snapmirror_source_endpoint</a> | Source endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.                        |

| Name  | Type   | Description   |
|-------|--------|---|
| state | string | <p>State of the relationship. To initialize the relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync". To break the relationship, PATCH the state to "broken_off" for relationships with a policy of type "async" or "sync". SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" cannot be "broken_off". To resync the relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync". SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" can be in "broken_off" state due to a failed attempt of SnapMirror failover. To pause the relationship, suspending further transfers, PATCH the state to "paused" for relationships with a policy of type "async" or "sync". SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" cannot be "paused". To resume transfers for a paused relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync". The entries "in_sync", "out_of_sync", "synchronizing", and "expanding" are only applicable to relationships with a policy of type "sync". A PATCH call on the state change only triggers the transition to the specified state. You must poll on the "state", "healthy" and "unhealthy_reason" properties using a GET request to determine if the transition is successful. To automatically initialize the relationship when specifying "create_destination" property, set the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for</p> |

| Name                    | Type                                      | Description   |
|-------------------------|---|---|
| svmdr_volumes           | array[ <a href="#">svmdr_volumes</a> ]    | Specifies the list of constituent FlexVol volumes and FlexGroup volumes for an SVM DR SnapMirror relationship. FlexGroup constituents are not considered.   |
| throttle                | integer                                   | Throttle, in KBytes per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited.  |
| total_transfer_bytes    | integer                                   | Cumulative bytes transferred for the relationship.  |
| total_transfer_duration | string                                    | Indicates the cumulative duration of all transfers since the last aggregate relocation, takeover/giveback, or metrocluster switchover/switchback involving the node that hosts the relationship.  |
| transfer                | <a href="#">transfer</a>                  | Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.   |
| transfer_schedule       | <a href="#">transfer_schedule</a>         | Schedule used to update asynchronous relationships. This "transfer_schedule" overrides the "transfer_schedule" set on the SnapMirror relationship's policy. To remove the "transfer_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror. |
| unhealthy_reason        | array[ <a href="#">snapmirror_error</a> ] | Reason the relationship is not healthy. It is a concatenation of up to four levels of error messages.   |
| uuid                    | string                                    | Unique identifier of the SnapMirror relationship.   |

## Example response

```
{  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "backoff_level": "medium",  
  "consistency_group_failover": {  
    "error": {  
      "arguments": [  
        {  
          "code": "string",  
          "message": "string"  
        }  
      ],  
      "code": "4",  
      "message": "entry doesn't exist"  
    },  
    "state": "string",  
    "status": {  
      "code": "string",  
      "message": "string"  
    },  
    "type": "string"  
  },  
  "destination": {  
    "cluster": {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "name": "cluster1",  
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
    },  
    "consistency_group_volumes": [  
      {  
        "name": "volume1"  
      }  
    ],  
    "path": "svm1:volume1",  
    "svm": {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      }  
    }  
  }  
}
```

```

        "href": "/api/resourcelink"
    }
},
"name": "svm1",
"uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
}
},
"exported_snapshot": "string",
"group_type": "consistency_group",
"identity_preservation": "string",
"io_serving_copy": "C1_sti85-vsimg-ucs209a_cluster, C1_sti85-vsimg-
ucs209c_cluster",
"lag_time": "PT8H35M42S",
"last_transfer_network_compression_ratio": 61,
"last_transfer_type": "initialize",
"master_bias_activated_site": "C1_sti85-vsimg-ucs209a_cluster",
"policy": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "name": "Asynchronous",
    "type": "string",
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
},
"preferred_site": "C1_sti85-vsimg-ucs209a_cluster",
"restore_to_snapshot": "string",
"source": {
    "cluster": {
        "_links": {
            "self": {
                "href": "/api/resourcelink"
            }
        },
        "name": "cluster1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "consistency_group_volumes": [
        {
            "name": "volume1"
        }
    ],
    "path": "svm1:volume1",
    "svm": {
        "_links": {

```

```
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "name": "svml1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
}
},
"state": "snapmirrored",
"svmdr_volumes": [
{
    "name": "volume1"
}
],
"throttle": 0,
"total_transfer_bytes": 1098210312,
"total_transfer_duration": "PT3M21S",
"transfer": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "end_time": "2020-12-02 21:36:19 -0500",
    "last_updated_time": "2023-09-14 18:39:19 -0400",
    "state": "string",
    "total_duration": "PT28M41S",
    "type": "initialize",
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
},
"transfer_schedule": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "name": "weekly",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"unhealthy_reason": [
{
    "arguments": [],
    "code": "6621444",
    "message": "Failed to complete update operation on one or more item relationships."
}
],
```

```
{
  "arguments": [],
  "code": "6621445",
  "message": "Group Update failed"
}
],
"uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
}
```

## Error

Status: Default

### ONTAP Error Response codes

| Error code | Description   |
|------------|---|
| 13303825   | Could not retrieve information for the SnapMirror policy type |
| 13303817   | Unknown value for the Snapmirror State                        |

| Name  | Type           | Description |
|-------|----------------|-------------|
| error | returned_error |             |

### Example error

```
{
  "error": {
    "arguments": [
      {
        "code": "string",
        "message": "string"
      }
    ],
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}
```

## Definitions

## See Definitions

href

| Name | Type   | Description |
|------|--------|-------------|
| href | string |             |

\_links

| Name | Type | Description |
|------|------|-------------|
| self | href |             |

error\_arguments

| Name    | Type   | Description      |
|---------|--------|------------------|
| code    | string | Argument code    |
| message | string | Message argument |

error

SnapMirror Consistency Group failover error message.

| Name      | Type                   | Description       |
|-----------|------------------------|-------------------|
| arguments | array[error_arguments] | Message arguments |
| code      | string                 | Error code        |
| message   | string                 | Error message     |

status

| Name    | Type   | Description                                   |
|---------|--------|---|
| code    | string | Status code                                   |
| message | string | SnapMirror Consistency Group failover status. |

snapmirror\_consistency\_group\_failover

SnapMirror Consistency Group failover information. The SnapMirror Consistency Group failover can be a planned or an unplanned operation. Only active SnapMirror Consistency Group failover operation progress can be monitored using this object. In case of an error during the failover operation, the property "consistency\_group\_failover.error" holds the reason for the error. ONTAP automatically retries any failed SnapMirror Consistency Group failover operation.

| Name   | Type                   | Description  |
|--------|------------------------|--|
| error  | <a href="#">error</a>  | SnapMirror Consistency Group failover error message. |
| state  | string                 | SnapMirror Consistency Group failover state.         |
| status | <a href="#">status</a> |  |
| type   | string                 | SnapMirror Consistency Group failover type.          |

#### bucket\_retention

Specifies the retention-mode and default retention period configured on the destination bucket.

| Name           | Type   | Description  |
|----------------|--------|--|
| default_period | string | Specifies the default retention period that is applied to objects while committing them to the WORM state without an associated retention period. The retention period can be in years, or days. The retention period value represents a duration and must be specified in the ISO-8601 duration format. A period specified for years and days is represented in the ISO-8601 format as "P<num>Y" and "P<num>D" respectively, for example "P10Y" represents a duration of 10 years. The period string must contain only a single time element that is, either years, or days. A duration which combines different periods is not supported, for example "P1Y10D" is not supported.</num></num> |

| Name | Type   | Description   |
|------|--------|---|
| mode | string | The lock mode of the bucket. compliance – A SnapLock Compliance (SLC) bucket provides the highest level of WORM protection and an administrator cannot destroy a compliance bucket if it contains unexpired WORM objects. governance – An administrator can delete a Governance bucket. no_lock – Indicates the bucket does not support object locking. |

### storage\_service

| Name                | Type    | Description   |
|---------------------|---------|---|
| enabled             | boolean | This property indicates whether to create the destination endpoint using storage service.   |
| enforce_performance | boolean | Optional property to enforce storage service performance on the destination endpoint. This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints. |

| Name | Type   | Description   |
|------|--------|---|
| name | string | <p>Optional property to specify the storage service name for the destination endpoint. This property is considered when the property "create_destination.storage_service.enabled" is set to "true". When the property "create_destination.storage_service.enabled" is set to "true" and the "create_destination.storage_service.name" for the endpoint is not specified, then ONTAP selects the highest storage service available on the cluster to provision the destination endpoint. This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints.</p> <ul style="list-style-type: none"> <li>• enum: ["extreme", "performance", "value"]</li> <li>• Introduced in: 9.6</li> <li>• x-nullable: true</li> </ul> |

tiering

| Name   | Type   | Description   |
|--------|--------|---|
| policy | string | <p>Optional property to specify the destination endpoint's tiering policy when "create_destination.tiering.supported" is set to "true". This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints. This property determines whether the user data blocks of the destination endpoint in a FabricPool will be tiered to the cloud store when they become cold. FabricPool combines flash (performance tier) with a cloud store into a single aggregate. Temperature of the destination endpoint volume blocks increases if they are accessed frequently and decreases when they are not. all – This policy allows tiering of both destination endpoint Snapshot copies and the user transferred data blocks to the cloud store as soon as possible by ignoring the temperature on the volume blocks. This tiering policy is not applicable for Consistency Group destination endpoints or for synchronous relationships. auto – This policy allows tiering of both destination endpoint Snapshot copies and the active file system user data to the cloud store none – Destination endpoint volume blocks will not be tiered to the cloud store. snapshot_only – This policy allows tiering of only the destination endpoint volume Snapshot copies not associated with the active file system. The default tiering policy is "snapshot_only" for a FlexVol volume and "none" for a FlexGroup volume.</p> |

| Name      | Type    | Description   |
|-----------|---------|---|
| supported | boolean | Optional property to enable provisioning of the destination endpoint volumes on FabricPool aggregates. This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints. Only FabricPool aggregates are used if this property is set to "true" and only non FabricPool aggregates are used if this property is set to "false". Tiering support for a FlexGroup volume can be changed by moving all of the constituents to the required aggregates. Note that in order to tier data, not only do the destination endpoint volumes need to support tiering by using FabricPools, the "create_destination.tiering.policy" must not be "none". A destination endpoint that uses FabricPools but has a tiering "policy" of "none" supports tiering but will not tier any data. |

#### snapmirror\_destination\_creation

Use this object to provision the destination endpoint when establishing a SnapMirror relationship for a FlexVol volume, FlexGroup volume, SVM, Consistency Group or ONTAP S3 Bucket. Given a source endpoint, the destination endpoint is provisioned in the SVM specified in the "destination.path" property. While protecting an SVM, the SVM destination endpoint can only be provisioned on the local cluster. To provision the SVM destination endpoint use the optional "source.cluster.name" property to specify the remote cluster name or use the optional "source.cluster.uuid" property to specify the remote cluster UUID. When "create\_destination.enabled" option is specified while making a POST for a SnapMirror relationship, the relationship can be automatically initialized by setting the "state" either to "snapmirrored" when the policy is of type "async" or to "in\_sync" when the policy is of type "sync". The "destination.path" property must specify the destination endpoint path. For example, for FlexVol volume and FlexGroup volume, the "destination.path" can be specified as <destination-svm-name:dp-volume-name>, for SVM data protection, the "destination.path" must be specified as &lt;destination-svm-name:&gt;, and for Consistency Group, the "destination.path" must be specified as &lt;destination-svm-name:&gt; along with the "destination.consistency\_group\_volumes" property to indicate the list of destination volumes of type "DP" in the Consistency Group. For a FlexVol volume, a FlexGroup volume, Consistency Group or a Bucket destination endpoint, the properties in this object can be specified either from the source or the destination cluster. For an SVM destination endpoint, the properties in this object can be specified from the destination cluster only. This object is not supported for non ONTAP endpoints. While protecting a S3 Bucket, the optional "size" property can be used to create ONTAP S3 Bucket destination endpoint of the specified size.&lt;/destination-svm-name:&gt;&lt;/destination-svm-name:&gt;</destination-svm-name:dp-volume-name>

| Name                     | Type                             | Description  |
|--------------------------|----------------------------------|--|
| bucket_retention         | <a href="#">bucket_retention</a> | Specifies the retention-mode and default retention period configured on the destination bucket.  |
| enabled                  | boolean                          | Optional property to create the destination endpoint when establishing a SnapMirror relationship. It is assumed to be "false" if no other property is set and assumed to be "true" if any other property is set.   |
| size                     | integer                          | Optional property to specify the size of destination endpoint in bytes. This property is applicable only to ONTAP S3 Bucket endpoints. The minimum size for S3 bucket is 80MB and maximum size is 64TB. If not specified, system will create destination with default size of 800GB. <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> <li>• x-nullable: true</li> </ul> |
| snapshot_locking_enabled | boolean                          | Optional property to create the destination endpoint with Snapshot copy locking enabled when establishing a SnapMirror relationship. This property is applicable to FlexVol volumes and FlexGroup volumes.   |
| storage_service          | <a href="#">storage_service</a>  |  |
| tiering                  | <a href="#">tiering</a>          |  |

## cluster

| Name   | Type                   | Description |
|--------|------------------------|-------------|
| _links | <a href="#">_links</a> |             |
| name   | string                 |             |
| uuid   | string                 |             |

## consistency\_group\_volumes

| Name | Type   | Description             |
|------|--------|-------------------------|
| name | string | The name of the volume. |

## svm

SVM, applies only to SVM-scoped objects.

| Name   | Type                   | Description   |
|--------|------------------------|---|
| _links | <a href="#">_links</a> |   |
| name   | string                 | The name of the SVM. This field cannot be specified in a PATCH method.              |
| uuid   | string                 | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

## snapmirror\_endpoint

Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.

| Name                      | Type   | Description   |
|---------------------------|--|---|
| cluster                   | <a href="#">cluster</a>                            |   |
| consistency_group_volumes | array[ <a href="#">consistency_group_volume</a> s] | Mandatory property for a Consistency Group endpoint. Specifies the list of FlexVol volumes for a Consistency Group. |

| Name | Type   | Description  |
|------|--------|--|
| path | string | ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name<br>ONTAP S3 - svm1:/bucket/bucket1 NON-ONTAP - objstore1:/objstore <ul style="list-style-type: none"> <li>• example: svm1:volume1</li> <li>• Introduced in: 9.6</li> <li>• x-nullable: true</li> </ul> |
| svm  | svm    | SVM, applies only to SVM-scoped objects.   |

## policy

Basic policy information of the relationship.

| Name   | Type   | Description                                 |
|--------|--------|---|
| _links | _links |   |
| name   | string | Name of the SnapMirror policy.              |
| type   | string |   |
| uuid   | string | Unique identifier of the SnapMirror policy. |

## snapmirror\_source\_endpoint

Source endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.

| Name                      | Type                             | Description   |
|---------------------------|----------------------------------|---|
| cluster                   | cluster                          |   |
| consistency_group_volumes | array[consistency_group_volumes] | Mandatory property for a Consistency Group endpoint. Specifies the list of FlexVol volumes for a Consistency Group. |

| Name | Type   | Description   |
|------|--------|---|
| path | string | ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name<br>ONTAP S3 - svm1:/bucket/bucket1 NON-ONTAP - objstore1:/objstore <ul style="list-style-type: none"> <li>• example: svm1:volume1</li> <li>• Introduced in: 9.14</li> <li>• x-nullable: true</li> </ul> |
| svm  | svm    | SVM, applies only to SVM-scoped objects.  |

#### svmdr\_volumes

| Name | Type   | Description             |
|------|--------|-------------------------|
| name | string | The name of the volume. |

#### transfer

Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.

| Name              | Type                   | Description  |
|-------------------|------------------------|--|
| _links            | <a href="#">_links</a> |  |
| bytes_transferred | integer                | Total bytes transferred in the current or last successful transfer.            |
| end_time          | string                 | End time of the last transfer.   |
| last_updated_time | string                 | Last updated time of the bytes transferred in the current transfer.            |
| state             | string                 |  |
| total_duration    | string                 | Elapsed time to transfer all Snapshot copies for the last successful transfer. |

| Name | Type   | Description  |
|------|--------|--|
| type | string | Specifies the operation type of the current transfer on the relationship. The <i>initialize</i> transfer occurs when the relationship state changes from "uninitialized" to "snapmirrored" or "in_sync". The <i>update</i> transfer occurs when Snapshot copies are being transferred from the source endpoint to the destination endpoint as part of a scheduled or manual update. The <i>resync</i> transfer occurs when the relationship state changes from "broken_off" to "snapmirrored" or "in_sync". The <i>restore</i> transfer occurs when a Snapshot copy is being restored from a destination endpoint to another endpoint. |
| uuid | string | Transfer UUID. This property is applicable only for active transfers.  |

### transfer\_schedule

Schedule used to update asynchronous relationships. This "transfer\_schedule" overrides the "transfer\_schedule" set on the SnapMirror relationship's policy. To remove the "transfer\_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror.

| Name   | Type                   | Description       |
|--------|------------------------|-------------------|
| _links | <a href="#">_links</a> |                   |
| name   | string                 | Job schedule name |
| uuid   | string                 | Job schedule UUID |

### snapmirror\_error

SnapMirror error

| Name      | Type          | Description   |
|-----------|---------------|---|
| arguments | array[string] | Arguments present in the error message encountered. |
| code      | string        | Error code  |

| Name    | Type   | Description   |
|---------|--------|---------------|
| message | string | Error message |

returned\_error

| Name      | Type                   | Description                                 |
|-----------|------------------------|---|
| arguments | array[error_arguments] | Message arguments                           |
| code      | string                 | Error code                                  |
| message   | string                 | Error message                               |
| target    | string                 | The target parameter that caused the error. |

## Update a SnapMirror relationship

PATCH /snapmirror/relationships/{uuid}

**Introduced In:** 9.6

Updates a SnapMirror relationship. This API is used to initiate SnapMirror operations such as "initialize", "resync", "break", "quiesce", and "resume" by specifying the appropriate value for the "state" field. It is also used to modify the SnapMirror policy associated with the specified relationship. Additionally, a SnapMirror relationship can be failed over to the destination endpoint or a failed over SnapMirror relationship can be failed back to the original state or a SnapMirror relationship direction can be reversed using this API. This API can also be used to expand the SnapMirror Business Continuity relationship with the specified source and destination volumes.

To initialize the relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or "in\_sync" for relationships with a policy of type "sync".

To break the relationship or to failover to the destination endpoint and start serving data from the destination endpoint, PATCH the state to "broken\_off" for relationships with a policy of type "async" or "sync". SnapMirror relationships with the policy type as "sync" and sync\_type as "automated\_failover" cannot be "broken\_off".

To resync the broken relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or "in\_sync" for relationships with a policy of type "sync".

To failback the failed over relationship and start serving data from the source endpoint, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or "in\_sync" for relationships with a policy of type "sync" and set the query flag "failback" as "true". SnapMirror relationships with the policy type as "sync" and sync\_type as "automated\_failover" can be in "broken\_off" state due to a failed attempt of automated SnapMirror failover operation.

To pause the relationship, suspending further transfers, PATCH the state to "paused" for relationships with a policy of type "async" or "sync". SnapMirror relationships with the policy type as "sync" and sync\_type as "automated\_failover" cannot be "paused".

To resume transfers for a paused relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or "in\_sync" for relationships with a policy of type "sync".

To reverse the direction of the relationship, PATCH the "source.path" with the destination endpoint and the "destination.path" with the source endpoint and the relationship state to "snapmirrored" for relationships with a policy of type "async" or "in\_sync" for relationships with a policy of type "sync". For relationships with a policy of type "async" and relationship state as "snapmirrored", stop IO on the source endpoint and perform a SnapMirror transfer POST operation before reversing the direction of the relationship to prevent any loss of data.

The values "in\_sync", "out\_of\_sync", and "synchronizing" are only applicable to relationships with a policy of type "sync".

When "transfer\_schedule" is specified along with "state" during PATCH, first the schedule is modified on the relationship and then the respective SnapMirror operation is initiated. The "transfer\_schedule" specified is used to update asynchronous relationships.

When "throttle" is specified along with "state" during PATCH, first the throttle is modified on the relationship, which will be used by any upcoming transfers and then the respective SnapMirror operation is initiated. If the SnapMirror operation initiated a transfer then it will also use the new throttle. If "throttle" needs to be applied for a specific transfer use SnapMirror Transfer REST API.

For SnapMirror Business Continuity (SMBC) relationships, when "consistency\_group\_volumes" is specified during PATCH, first the existing FlexVol volume relationship is deleted and released and then the volumes are added to the SMBC relationship.

## Examples

### Related ONTAP commands

- snapmirror modify
- snapmirror initialize
- snapmirror resync
- snapmirror break
- snapmirror quiesce
- snapmirror resume

### Important notes

- The property "transfer\_schedule" if set on a SnapMirror relationship overrides the "transfer\_schedule" set on the policy being used with the SnapMirror relationship.
- The property "throttle" if set on a SnapMirror relationship overrides the "throttle" set on the policy being used with the SnapMirror relationship.
- The properties "transfer\_schedule" and "throttle" are not supported when "failback" is set to "true".
- The properties "transfer\_schedule" and "throttle" are not supported when "failover" is set to "true".
- The properties "transfer\_schedule" and "throttle" are not supported when "force\_failover" is set to "true".
- The properties "transfer\_schedule" and "throttle" are not supported when the direction of the relationship is being reversed.

- To remove a transfer\_schedule on a SnapMirror relationship set the "transfer\_schedule" to null (no-quotes) during SnapMirror relationship PATCH.
- The property "identity\_preservation" value can be changed from a higher "identity\_preservation" threshold value to a lower "identity\_preservation" threshold value but not vice-versa. For example, the threshold value of the "identity\_preservation" property can be changed from "full" to "exclude\_network\_config", but cannot be increased from "exclude\_network\_and\_protocol\_config" to "exclude\_network\_config" to "full". The threshold value of the "identity\_preservation" cannot be changed to "exclude\_network\_and\_protocol\_config" for IDP SVMDR.
- The property "backoff\_level" is only applicable for FlexVol SnapMirror relationships.

## Examples

The following examples show how to perform the SnapMirror "resync", "initialize", "resume", "quiesce", and "break" operations. In addition, a relationship can be failed over to the destination endpoint and start serving data from the destination endpoint. A failed over relationship can be failed back to the source endpoint and serve data from the source endpoint. Also a relationship can be reversed by making the source endpoint as the new destination endpoint and the destination endpoint as the new source endpoint.

To update an associated SnapMirror policy.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{"policy": { "name" : "MirrorAndVaultDiscardNetwork" } }'
```

To perform SnapMirror "resync" for an asynchronous SnapMirror relationship.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{"state":"snapmirrored"}'
```

To perform SnapMirror "initialize" for an asynchronous SnapMirror relationship.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{"state":"snapmirrored"}'
```

To perform SnapMirror "resume" for an asynchronous SnapMirror relationship.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{"state":"snapmirrored"}'
```

To perform SnapMirror "quiesce" for an asynchronous SnapMirror relationship.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff" '{"state":"paused"}'
```

To perform SnapMirror "break" for an asynchronous SnapMirror relationship. This operation does a failover to

the destination endpoint. After a the failover, data can then be served from the destination endpoint.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff" '{"state":"broken_off"}'
```

To forcefully failover to the destination endpoint and start serving data from the destination endpoint.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/?force=true" '{"state":"broken_off"}'
```

To fallback to the source endpoint and start serving data from the source endpoint for an asynchronous relationship.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/?fallback=true" '{"state":"snapmirrored"}'
```

To fallback to the source endpoint and start serving data from the source endpoint for a synchronous relationship.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/?fallback=true" '{"state":"in_sync"}'
```

To reverse the direction of an asynchronous relationship, that is, make the source endpoint as the new destination endpoint and make the destination endpoint as the new source endpoint.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{"source": {"path": "dst_svm:dst_vol"}, "destination": {"path": "src_svm:src_vol"}, "state": "snapmirrored"}'
```

To reverse the direction of a synchronous relationship, that is, make the source endpoint as the new destination endpoint and make the destination endpoint as the new source endpoint.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{"source": {"path": "dst_svm:dst_vol"}, "destination": {"path": "src_svm:src_vol"}, "state": "in_sync"}'
```

Updating SnapMirror transfer\_schedule and throttle for an asynchronous SnapMirror relationship. Transfer\_schedule can be specified as UUID or name or both.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{"transfer_schedule":{"uuid":"817500fa-092d-44c5-9c10-7b54f7b2f20a", "name":"5min"}, "throttle":100}'
```

Removing the SnapMirror transfer\_schedule for an asynchronous SnapMirror relationship.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{"transfer_schedule":{"uuid":null, "name":null}}'
```

Removing the SnapMirror throttle for an asynchronous SnapMirror relationship.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{"throttle":0}'
```

To perform SnapMirror "resync" and update the SnapMirror transfer\_schedule for an asynchronous SnapMirror relationship. First the transfer\_schedule is modified and then the resync transfer is initiated.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{"state":"snapmirrored", "transfer_schedule":{"uuid":"817500fa-092d-44c5-9c10-7b54f7b2f20a", "name":"5min"}}'
```

To perform SnapMirror "initialize" and update the SnapMirror throttle for an asynchronous SnapMirror relationship. First the throttle is modified and then the initialize transfer is initiated. The initialize transfer will use this new throttle.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{"state":"snapmirrored", "throttle":100}'
```

To perform SnapMirror "resync" and update the SnapMirror throttle for an asynchronous SnapMirror relationship. First the throttle is modified and then the resync transfer is initiated. The resync transfer will use this new throttle.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{"state":"snapmirrored", "throttle":100}'
```

To perform a SnapMirror Business Continuity or Asynchronous Consistency Group expansion.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{ "source" : {"consistency_group_volumes": [{"name": "vol"}]}, "destination" : {"consistency_group_volumes": [{"name": "voldp"}]} }'
```

Updating SnapMirror backoff\_level for an asynchronous SnapMirror relationship.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{"backoff_level": "none"}'
```

To perform SnapMirror "initialize" and update the SnapMirror backoff\_level for an asynchronous SnapMirror relationship. First the backoff\_level is modified and then the initialize transfer is initiated. The initialize transfer will use this new backoff\_level.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{"state": "snapmirrored", "backoff_level": "medium"}'
```

To perform SnapMirror "resync" and update the SnapMirror backoff\_level for an asynchronous SnapMirror relationship. First the backoff\_level is modified and then the resync transfer is initiated. The resync transfer will use this new backoff\_level.

```
PATCH "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" '{"state": "snapmirrored", "backoff_level": "medium"}'
```

## Learn more

- [DOC /snapmirror/relationships](#)

## Parameters

| Name | Type   | In   | Required | Description                  |
|------|--------|------|----------|------------------------------|
| uuid | string | path | True     | SnapMirror relationship UUID |

| Name           | Type    | In    | Required | Description   |
|----------------|---------|-------|----------|---|
| failover       | boolean | query | False    | <p>If this parameter is set, validation and failover will occur to the SVM-DR SnapMirror relationship destination endpoint. Any other fields specified with this parameter will be ignored. This parameter is supported only for SVM-DR SnapMirror relationships.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.7</li> <li>• Default value:</li> </ul>                    |
| force-failover | boolean | query | False    | <p>If this parameter is set, failover will occur to the SVM-DR SnapMirror relationship destination endpoint, overriding the validation errors. Any other fields specified with this parameter will be ignored. This parameter is supported only for SVM-DR SnapMirror relationships.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.7</li> <li>• Default value:</li> </ul> |

| Name     | Type    | In    | Required | Description   |
|----------|---------|-------|----------|---|
| force    | boolean | query | False    | <p>If this parameter is set while specifying the state as "broken_off", indicates a forced failover overriding the validation errors.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.8</li> <li>• Default value:</li> </ul>  |
| fallback | boolean | query | False    | <p>If this parameter is set while specifying the state as "snapmirrored", indicates recovery of the failed over SnapMirror relationship by preserving the data written on the destination endpoint when the SnapMirror relationship was in failed over state. This flag is only applicable to SVM-DR SnapMirror relationships.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.8</li> <li>• Default value:</li> </ul> |

| Name           | Type    | In    | Required | Description  |
|----------------|---------|-------|----------|--|
| return_timeout | integer | query | False    | <p>The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul> |
| validate_only  | boolean | query | False    | <p>Validate the operation and its parameters, without actually performing the operation.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.7</li> </ul>  |

## Request Body

| Name          | Type   | Description  |
|---------------|--------|--|
| backoff_level | string | Specifies the SnapMirror backoff level due to Client Ops for FlexVol SnapMirror relationships. |

| Name              | Type                                | Description   |
|-------------------|-------------------------------------|---|
| destination       | <a href="#">snapmirror_endpoint</a> | <p>Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> </ul> |
| exported_snapshot | string                              | Snapshot copy exported to clients on destination.   |
| group_type        | string                              | Specifies the group type of the top level SnapMirror relationship. The volume relationships are shown as <i>none</i> , the SVMDR relationships are shown as <i>svm_dr</i> , the Consistency Group relationships are shown as <i>consistency_group</i> , and the FlexGroup volume relationships are shown as <i>flexgroup</i> .  |
| healthy           | boolean                             | Is the relationship healthy?  |

| Name                                    | Type   | Description  |
|---|--------|--|
| identity_preservation                   | string | Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type. This "identity_preservation" overrides the "identity_preservation" set on the SnapMirror relationship's policy.   |
| io_serving_copy                         | string | Specifies the sites serving I/O for the SnapMirror Business Continuity(SM-BC) relationship.  |
| lag_time                                | string | Time since the exported Snapshot copy was created.   |
| last_transfer_network_compression_ratio | string | Specifies the compression ratio achieved for the data sent over the wire with network compression enabled for the last successful transfer.  |
| last_transfer_type                      | string | Specifies the operation type of the last transfer that occurred on the relationship. The <i>initialize</i> transfer occurs when the relationship state changes from uninitialized to snapmirrored or in_sync. The <i>update</i> transfer occurs when the snapshots are transferred from the source endpoint to the destination endpoint as part of scheduled or manual update. The <i>resync</i> transfer occurs when the relationship state changes from broken_off to snapmirrored or in_sync. The <i>restore</i> transfer occurs when the snapshot is restored from a destination endpoint to another endpoint. |
| master_bias_activated_site              | string | Specifies the Master Bias Activated Site for the SnapMirror Business Continuity(SM-BC) relationship.   |
| policy                                  | policy | Basic policy information of the relationship.  |

| Name                | Type    | Description   |
|---------------------|---------|---|
| preferred_site      | string  | Specifies the Primary Site of the SnapMirror Business Continuity(SM-BC) relationship.   |
| preserve            | boolean | Set to true on resync to preserve Snapshot copies on the destination that are newer than the latest common Snapshot copy. This property is applicable only for relationships with FlexVol volume or FlexGroup volume endpoints and when the PATCH state is being changed to "snapmirrored".   |
| quick_resync        | boolean | Set to true to reduce resync time by not preserving storage efficiency. This property is applicable only for relationships with FlexVol volume endpoints and SVMDR relationships when the PATCH state is being changed to "snapmirrored".   |
| recover_after_break | boolean | Set to true to recover from a failed SnapMirror break operation on a FlexGroup volume relationship. This restores all destination FlexGroup constituent volumes to the latest Snapshot copy, and any writes to the read-write constituents are lost. This property is applicable only for SnapMirror relationships with FlexGroup volume endpoints and when the PATCH state is being changed to "broken_off". |
| restore_to_snapshot | string  | Specifies the Snapshot copy to restore to on the destination during the break operation. This property is applicable only for SnapMirror relationships with FlexVol volume endpoints and when the PATCH state is being changed to "broken_off".   |

| Name   | Type                                       | Description  |
|--------|--|--|
| source | <a href="#">snapmirror_source_endpoint</a> | Source endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint. |

| Name  | Type   | Description   |
|-------|--------|---|
| state | string | <p>State of the relationship. To initialize the relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync". To break the relationship, PATCH the state to "broken_off" for relationships with a policy of type "async" or "sync". SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" cannot be "broken_off". To resync the relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync". SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" can be in "broken_off" state due to a failed attempt of SnapMirror failover. To pause the relationship, suspending further transfers, PATCH the state to "paused" for relationships with a policy of type "async" or "sync". SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" cannot be "paused". To resume transfers for a paused relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync". The entries "in_sync", "out_of_sync", "synchronizing", and "expanding" are only applicable to relationships with a policy of type "sync". A PATCH call on the state change only triggers the transition to the specified state. You must poll on the "state", "healthy" and "unhealthy_reason" properties using a GET request to determine if the transition is successful. To automatically initialize the relationship when specifying "create_destination" property, set the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for</p> |

| Name                    | Type                    | Description   |
|-------------------------|-------------------------|---|
| throttle                | integer                 | Throttle, in KBs per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited.   |
| total_transfer_bytes    | integer                 | Cumulative bytes transferred for the relationship.  |
| total_transfer_duration | string                  | Indicates the cumulative duration of all transfers since the last aggregate relocation, takeover/giveback, or metrocluster switchover/switchback involving the node that hosts the relationship.  |
| transfer                | transfer                | Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.   |
| transfer_schedule       | transfer_schedule       | Schedule used to update asynchronous relationships. This "transfer_schedule" overrides the "transfer_schedule" set on the SnapMirror relationship's policy. To remove the "transfer_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror. |
| unhealthy_reason        | array[snapmirror_error] | Reason the relationship is not healthy. It is a concatenation of up to four levels of error messages.   |
| uuid                    | string                  | Unique identifier of the SnapMirror relationship.   |

## Example request

```
{  
    "backoff_level": "medium",  
    "destination": {  
        "cluster": {  
            "name": "cluster1",  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "path": "svm1:volumel",  
        "svm": {  
            "name": "svm1",  
            "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
        }  
    },  
    "exported_snapshot": "string",  
    "group_type": "consistency_group",  
    "identity_preservation": "string",  
    "io_serving_copy": "C1_sti85-vsimg-ucs209a_cluster, C1_sti85-vsimg-  
ucs209c_cluster",  
    "lag_time": "PT8H35M42S",  
    "last_transfer_network_compression_ratio": 61,  
    "last_transfer_type": "initialize",  
    "master_bias_activated_site": "C1_sti85-vsimg-ucs209a_cluster",  
    "policy": {  
        "name": "Asynchronous",  
        "type": "string",  
        "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"  
    },  
    "preferred_site": "C1_sti85-vsimg-ucs209a_cluster",  
    "restore": true,  
    "restore_to_snapshot": "string",  
    "source": {  
        "cluster": {  
            "name": "cluster1",  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
        },  
        "path": "svm1:volumel",  
        "svm": {  
            "name": "svm1",  
            "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
        }  
    },  
    "state": "snapmirrored",  
    "throttle": 0,  
    "total_transfer_bytes": 1098210312,  
    "total_transfer_time": "PT1H35M42S",  
    "transfer_id": 1  
}
```

```

"total_transfer_duration": "PT3M21S",
"transfer": {
    "end_time": "2020-12-02 21:36:19 -0500",
    "last_updated_time": "2023-09-14 18:39:19 -0400",
    "state": "string",
    "total_duration": "PT28M41S",
    "type": "initialize",
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
},
"transfer_schedule": {
    "name": "weekly",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
},
"unhealthy_reason": [
    {
        "arguments": [],
        "code": "6621444",
        "message": "Failed to complete update operation on one or more item relationships."
    },
    {
        "arguments": [],
        "code": "6621445",
        "message": "Group Update failed"
    }
],
"uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
}

```

## Response

Status: 200, Ok

| Name | Type     | Description |
|------|----------|-------------|
| job  | job_link |             |

## Example response

```
{  
  "job": {  
    "uuid": "string"  
  }  
}
```

## Response

Status: 202, Accepted

## Error

Status: Default

## ONTAP Error Response codes

| Error code | Description  |
|------------|--|
| 13303825   | Could not retrieve information for the SnapMirror policy type        |
| 13303817   | Unknown value for the SnapMirror state                               |
| 13303829   | Invalid state  |
| 13303830   | Transient state  |
| 13303831   | Invalid state for async SnapMirror relationship                      |
| 13303834   | Given input valid only for FlexGroup SnapMirror relationship         |
| 13303835   | Given flag is valid only when PATCH state is broken_off              |
| 13303836   | Given flag is valid only when PATCH state is snapmirrored or in_sync |
| 13303818   | Invalid state transition requested                                   |
| 13303828   | Given state change is not possible for SVM SnapMirror relationship   |
| 13303833   | Requested state change is not possible                               |
| 13303832   | SnapMirror relationship is already initialized                       |
| 13303824   | Quiescing the SnapMirror relationship has failed                     |
| 13303826   | Required environment variables are not set                           |

| Error code | Description   |
|------------|---|
| 13303827   | Internal Error  |
| 13303823   | Quiesce operation timed out   |
| 13303821   | Invalid SnapMirror policy name/UUID   |
| 13303819   | Could not retrieve SnapMirror policy information  |
| 13303851   | Cannot modify attributes of SnapMirror restore relationship   |
| 13303816   | Could not retrieve state or status values   |
| 13303837   | Given flags are valid only if SnapMirror state change is requested  |
| 6619546    | Destination must be a dp volume   |
| 13303808   | Transition to broken_off state failed   |
| 13303809   | Transition to paused state failed   |
| 13303810   | Transition to snapmirrored state failed   |
| 13303811   | Transition from paused state failed   |
| 13303820   | SnapMirror policy, transfer_schedule, and throttle, if specified were successfully updated, state transition failed                           |
| 13303856   | SVM is not configured with any data protocol  |
| 13303857   | SVM is not configured with any network interface  |
| 13303858   | Internal error. Failed to check LIF and protocols details for SVM   |
| 13303859   | Internal error. SVM Failover operation failed. SVM operational state is unavailable.  |
| 13303865   | Modifying the specified SnapMirror policy is not supported.   |
| 13303866   | Cannot use the specified policy to modify the policy of the relationship.   |
| 13303867   | Modifying the policy of an async-mirror or a vault relationship is not supported.   |
| 13303884   | LIF and protocols details are configured incorrectly for SVM.   |
| 13303996   | The source and destination clusters both have a policy with the same name, but they have different properties.                                |
| 13304062   | Cannot reverse the direction of a SnapMirror DP relationship when the source cluster version is earlier than the destination cluster version. |

| Error code | Description   |
|------------|---|
| 13304070   | Remote peer cluster requires the dp_rest_support capability to support reversing the direction of a DP relationship.  |
| 13304071   | Failed to access capabilities on remote cluster.  |
| 13304080   | Specified uuid and name do not match.   |
| 13304081   | Modifying a property during the operation is not supported.   |
| 13304082   | The specified properties are mutually exclusive.  |
| 13304083   | The specified property is not supported because all nodes in the cluster are not capable of supporting the property.  |
| 13304086   | Reversing the direction of a SnapMirror relationship associated with a policy containing the property create_snapshot_on_source set to false is not supported.                                      |
| 6619715    | Modification of relationship is in progress. Retry the command after a few minutes.   |
| 6619699    | Schedule not found.   |
| 13304108   | Schedule not found in the Administrative SVM or the SVM for the relationship.   |
| 13304111   | The SnapMirror Business Continuity (SM-BC) relationship consistency groups are nested. Expanding an SM-BC relationship with a pre-existing DP volume is only supported for flat consistency groups. |
| 6621458    | The destination Consistency Group is the source of a SnapMirror Synchronous (SM-S) relationship. Sources of SM-S relationships cannot be the destination of any other SnapMirror relationship.      |
| 6621782    | A property of the policy is not valid for relationships between these endpoints.  |

| 13304120 | Values specified for the source.path and destination.path properties do not match the relationship's source.path or destination.path properties. | 13304093 | The property specified is not supported for the specified relationships. |

## Definitions

## See Definitions

href

| Name | Type   | Description |
|------|--------|-------------|
| href | string |             |

\_links

error\_arguments

| Name    | Type   | Description      |
|---------|--------|------------------|
| code    | string | Argument code    |
| message | string | Message argument |

error

SnapMirror Consistency Group failover error message.

| Name      | Type                   | Description       |
|-----------|------------------------|-------------------|
| arguments | array[error_arguments] | Message arguments |
| code      | string                 | Error code        |
| message   | string                 | Error message     |

status

| Name    | Type   | Description                                   |
|---------|--------|---|
| code    | string | Status code                                   |
| message | string | SnapMirror Consistency Group failover status. |

snapmirror\_consistency\_group\_failover

SnapMirror Consistency Group failover information. The SnapMirror Consistency Group failover can be a planned or an unplanned operation. Only active SnapMirror Consistency Group failover operation progress can be monitored using this object. In case of an error during the failover operation, the property "consistency\_group\_failover.error" holds the reason for the error. ONTAP automatically retries any failed SnapMirror Consistency Group failover operation.

| Name   | Type                   | Description                                  |
|--------|------------------------|--|
| state  | string                 | SnapMirror Consistency Group failover state. |
| status | <a href="#">status</a> |  |
| type   | string                 | SnapMirror Consistency Group failover type.  |

#### bucket\_retention

Specifies the retention-mode and default retention period configured on the destination bucket.

| Name           | Type   | Description  |
|----------------|--------|--|
| default_period | string | Specifies the default retention period that is applied to objects while committing them to the WORM state without an associated retention period. The retention period can be in years, or days. The retention period value represents a duration and must be specified in the ISO-8601 duration format. A period specified for years and days is represented in the ISO-8601 format as "P<num>Y" and "P<num>D" respectively, for example "P10Y" represents a duration of 10 years. The period string must contain only a single time element that is, either years, or days. A duration which combines different periods is not supported, for example "P1Y10D" is not supported.</num></num> |

#### storage\_service

| Name    | Type    | Description   |
|---------|---------|---|
| enabled | boolean | This property indicates whether to create the destination endpoint using storage service. |

| Name                | Type    | Description   |
|---------------------|---------|---|
| enforce_performance | boolean | Optional property to enforce storage service performance on the destination endpoint. This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints.   |
| name                | string  | <p>Optional property to specify the storage service name for the destination endpoint. This property is considered when the property "create_destination.storage_service.enabled" is set to "true". When the property "create_destination.storage_service.enabled" is set to "true" and the "create_destination.storage_service.name" for the endpoint is not specified, then ONTAP selects the highest storage service available on the cluster to provision the destination endpoint. This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints.</p> <ul style="list-style-type: none"> <li>• enum: ["extreme", "performance", "value"]</li> <li>• Introduced in: 9.6</li> <li>• x-nullable: true</li> </ul> |

tiering

| Name   | Type   | Description   |
|--------|--------|---|
| policy | string | <p>Optional property to specify the destination endpoint's tiering policy when "create_destination.tiering.supported" is set to "true". This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints. This property determines whether the user data blocks of the destination endpoint in a FabricPool will be tiered to the cloud store when they become cold. FabricPool combines flash (performance tier) with a cloud store into a single aggregate. Temperature of the destination endpoint volume blocks increases if they are accessed frequently and decreases when they are not. all – This policy allows tiering of both destination endpoint Snapshot copies and the user transferred data blocks to the cloud store as soon as possible by ignoring the temperature on the volume blocks. This tiering policy is not applicable for Consistency Group destination endpoints or for synchronous relationships. auto – This policy allows tiering of both destination endpoint Snapshot copies and the active file system user data to the cloud store none – Destination endpoint volume blocks will not be tiered to the cloud store. snapshot_only – This policy allows tiering of only the destination endpoint volume Snapshot copies not associated with the active file system. The default tiering policy is "snapshot_only" for a FlexVol volume and "none" for a FlexGroup volume.</p> |

| Name      | Type    | Description   |
|-----------|---------|---|
| supported | boolean | Optional property to enable provisioning of the destination endpoint volumes on FabricPool aggregates. This property is applicable to FlexVol volume, FlexGroup volume, and Consistency Group endpoints. Only FabricPool aggregates are used if this property is set to "true" and only non FabricPool aggregates are used if this property is set to "false". Tiering support for a FlexGroup volume can be changed by moving all of the constituents to the required aggregates. Note that in order to tier data, not only do the destination endpoint volumes need to support tiering by using FabricPools, the "create_destination.tiering.policy" must not be "none". A destination endpoint that uses FabricPools but has a tiering "policy" of "none" supports tiering but will not tier any data. |

#### snapmirror\_destination\_creation

Use this object to provision the destination endpoint when establishing a SnapMirror relationship for a FlexVol volume, FlexGroup volume, SVM, Consistency Group or ONTAP S3 Bucket. Given a source endpoint, the destination endpoint is provisioned in the SVM specified in the "destination.path" property. While protecting an SVM, the SVM destination endpoint can only be provisioned on the local cluster. To provision the SVM destination endpoint use the optional "source.cluster.name" property to specify the remote cluster name or use the optional "source.cluster.uuid" property to specify the remote cluster UUID. When "create\_destination.enabled" option is specified while making a POST for a SnapMirror relationship, the relationship can be automatically initialized by setting the "state" either to "snapmirrored" when the policy is of type "async" or to "in\_sync" when the policy is of type "sync". The "destination.path" property must specify the destination endpoint path. For example, for FlexVol volume and FlexGroup volume, the "destination.path" can be specified as <destination-svm-name:dp-volume-name>, for SVM data protection, the "destination.path" must be specified as &lt;destination-svm-name:&gt;, and for Consistency Group, the "destination.path" must be specified as &lt;destination-svm-name:&gt; along with the "destination.consistency\_group\_volumes" property to indicate the list of destination volumes of type "DP" in the Consistency Group. For a FlexVol volume, a FlexGroup volume, Consistency Group or a Bucket destination endpoint, the properties in this object can be specified either from the source or the destination cluster. For an SVM destination endpoint, the properties in this object can be specified from the destination cluster only. This object is not supported for non ONTAP endpoints. While protecting a S3 Bucket, the optional "size" property can be used to create ONTAP S3 Bucket destination endpoint of the specified size.&lt;/destination-svm-name:&gt;&lt;/destination-svm-name:&gt;</destination-svm-name:dp-volume-name>

| Name                     | Type                             | Description  |
|--------------------------|----------------------------------|--|
| bucket_retention         | <a href="#">bucket_retention</a> | Specifies the retention-mode and default retention period configured on the destination bucket.  |
| enabled                  | boolean                          | Optional property to create the destination endpoint when establishing a SnapMirror relationship. It is assumed to be "false" if no other property is set and assumed to be "true" if any other property is set.   |
| size                     | integer                          | Optional property to specify the size of destination endpoint in bytes. This property is applicable only to ONTAP S3 Bucket endpoints. The minimum size for S3 bucket is 80MB and maximum size is 64TB. If not specified, system will create destination with default size of 800GB. <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> <li>• x-nullable: true</li> </ul> |
| snapshot_locking_enabled | boolean                          | Optional property to create the destination endpoint with Snapshot copy locking enabled when establishing a SnapMirror relationship. This property is applicable to FlexVol volumes and FlexGroup volumes.   |
| storage_service          | <a href="#">storage_service</a>  |  |
| tiering                  | <a href="#">tiering</a>          |  |

cluster

| Name | Type   | Description |
|------|--------|-------------|
| name | string |             |
| uuid | string |             |

consistency\_group\_volumes

| Name | Type   | Description             |
|------|--------|-------------------------|
| name | string | The name of the volume. |

## svm

SVM, applies only to SVM-scoped objects.

| Name | Type   | Description   |
|------|--------|---|
| name | string | The name of the SVM. This field cannot be specified in a PATCH method.              |
| uuid | string | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

## snapmirror\_endpoint

Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.

| Name    | Type    | Description   |
|---------|---------|---|
| cluster | cluster |   |
| path    | string  | ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name ONTAP S3 - svm1:/bucket/bucket1 NON-ONTAP - objstore1:/objstore <ul style="list-style-type: none"> <li>example: svm1:volume1</li> <li>Introduced in: 9.6</li> <li>x-nullable: true</li> </ul> |
| svm     | svm     | SVM, applies only to SVM-scoped objects.  |

## policy

Basic policy information of the relationship.

| Name | Type   | Description                                 |
|------|--------|---|
| name | string | Name of the SnapMirror policy.              |
| type | string |   |
| uuid | string | Unique identifier of the SnapMirror policy. |

#### snapmirror\_source\_endpoint

Source endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.

| Name    | Type    | Description  |
|---------|---------|--|
| cluster | cluster |  |
| path    | string  | ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name ONTAP S3 - svm1:/bucket/bucket1 NON-ONTAP - objstore1:/objstore <ul style="list-style-type: none"> <li>example: svm1:volume1</li> <li>Introduced in: 9.14</li> <li>x-nullable: true</li> </ul> |
| svm     | svm     | SVM, applies only to SVM-scoped objects.   |

#### svmdr\_volumes

| Name | Type   | Description             |
|------|--------|-------------------------|
| name | string | The name of the volume. |

#### transfer

Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.

| Name              | Type    | Description   |
|-------------------|---------|---|
| bytes_transferred | integer | Total bytes transferred in the current or last successful transfer. |

| Name              | Type   | Description  |
|-------------------|--------|--|
| end_time          | string | End time of the last transfer.   |
| last_updated_time | string | Last updated time of the bytes transferred in the current transfer.  |
| state             | string |  |
| total_duration    | string | Elapsed time to transfer all Snapshot copies for the last successful transfer.   |
| type              | string | Specifies the operation type of the current transfer on the relationship. The <i>initialize</i> transfer occurs when the relationship state changes from "uninitialized" to "snapmirrored" or "in_sync". The <i>update</i> transfer occurs when Snapshot copies are being transferred from the source endpoint to the destination endpoint as part of a scheduled or manual update. The <i>resync</i> transfer occurs when the relationship state changes from "broken_off" to "snapmirrored" or "in_sync". The <i>restore</i> transfer occurs when a Snapshot copy is being restored from a destination endpoint to another endpoint. |
| uuid              | string | Transfer UUID. This property is applicable only for active transfers.  |

#### transfer\_schedule

Schedule used to update asynchronous relationships. This "transfer\_schedule" overrides the "transfer\_schedule" set on the SnapMirror relationship's policy. To remove the "transfer\_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror.

| Name | Type   | Description       |
|------|--------|-------------------|
| name | string | Job schedule name |
| uuid | string | Job schedule UUID |

#### snapmirror\_error

## SnapMirror error

| Name      | Type          | Description   |
|-----------|---------------|---|
| arguments | array[string] | Arguments present in the error message encountered. |
| code      | string        | Error code  |
| message   | string        | Error message                                       |

## snapmirror\_relationship

SnapMirror relationship information. The SnapMirror relationship can be either "async" or "sync" based on the type of SnapMirror policy associated with the relationship. The source and destination endpoints of a SnapMirror relationship must be of the same type, for example, if the source endpoint is a FlexVol volume then the destination endpoint must be a FlexVol volume. The SnapMirror policy type "async" can be used when the SnapMirror relationship has FlexVol volume or FlexGroup volume or SVM as the endpoint. The SnapMirror policy type "sync" can be used when the SnapMirror relationship has FlexVol volume as the endpoint. The SnapMirror policy type "sync" with "sync\_type" as "automated\_failover" can be used when the SnapMirror relationship has Consistency Group as the endpoint.

| Name          | Type   | Description  |
|---------------|--------|--|
| backoff_level | string | Specifies the SnapMirror backoff level due to Client Ops for FlexVol SnapMirror relationships. |

| Name              | Type                | Description   |
|-------------------|---------------------|---|
| destination       | snapmirror_endpoint | <p>Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> </ul> |
| exported_snapshot | string              | Snapshot copy exported to clients on destination.   |
| group_type        | string              | Specifies the group type of the top level SnapMirror relationship. The volume relationships are shown as <i>none</i> , the SVMDR relationships are shown as <i>svm_dr</i> , the Consistency Group relationships are shown as <i>consistency_group</i> , and the FlexGroup volume relationships are shown as <i>flexgroup</i> .  |
| healthy           | boolean             | Is the relationship healthy?  |

| Name                                    | Type   | Description  |
|---|--------|--|
| identity_preservation                   | string | Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type. This "identity_preservation" overrides the "identity_preservation" set on the SnapMirror relationship's policy.   |
| io_serving_copy                         | string | Specifies the sites serving I/O for the SnapMirror Business Continuity(SM-BC) relationship.  |
| lag_time                                | string | Time since the exported Snapshot copy was created.   |
| last_transfer_network_compression_ratio | string | Specifies the compression ratio achieved for the data sent over the wire with network compression enabled for the last successful transfer.  |
| last_transfer_type                      | string | Specifies the operation type of the last transfer that occurred on the relationship. The <i>initialize</i> transfer occurs when the relationship state changes from uninitialized to snapmirrored or in_sync. The <i>update</i> transfer occurs when the snapshots are transferred from the source endpoint to the destination endpoint as part of scheduled or manual update. The <i>resync</i> transfer occurs when the relationship state changes from broken_off to snapmirrored or in_sync. The <i>restore</i> transfer occurs when the snapshot is restored from a destination endpoint to another endpoint. |
| master_bias_activated_site              | string | Specifies the Master Bias Activated Site for the SnapMirror Business Continuity(SM-BC) relationship.   |

| Name                | Type    | Description   |
|---------------------|---------|---|
| policy              | policy  | Basic policy information of the relationship.   |
| preferred_site      | string  | Specifies the Primary Site of the SnapMirror Business Continuity(SM-BC) relationship.   |
| preserve            | boolean | Set to true on resync to preserve Snapshot copies on the destination that are newer than the latest common Snapshot copy. This property is applicable only for relationships with FlexVol volume or FlexGroup volume endpoints and when the PATCH state is being changed to "snapmirrored".   |
| quick_resync        | boolean | Set to true to reduce resync time by not preserving storage efficiency. This property is applicable only for relationships with FlexVol volume endpoints and SVMDR relationships when the PATCH state is being changed to "snapmirrored".   |
| recover_after_break | boolean | Set to true to recover from a failed SnapMirror break operation on a FlexGroup volume relationship. This restores all destination FlexGroup constituent volumes to the latest Snapshot copy, and any writes to the read-write constituents are lost. This property is applicable only for SnapMirror relationships with FlexGroup volume endpoints and when the PATCH state is being changed to "broken_off". |
| restore_to_snapshot | string  | Specifies the Snapshot copy to restore to on the destination during the break operation. This property is applicable only for SnapMirror relationships with FlexVol volume endpoints and when the PATCH state is being changed to "broken_off".   |

| Name   | Type                                       | Description  |
|--------|--|--|
| source | <a href="#">snapmirror_source_endpoint</a> | Source endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint. |

| Name  | Type   | Description   |
|-------|--------|---|
| state | string | <p>State of the relationship. To initialize the relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync". To break the relationship, PATCH the state to "broken_off" for relationships with a policy of type "async" or "sync".</p> <p>SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" cannot be "broken_off". To resync the relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync".</p> <p>SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" can be in "broken_off" state due to a failed attempt of SnapMirror failover. To pause the relationship, suspending further transfers, PATCH the state to "paused" for relationships with a policy of type "async" or "sync". SnapMirror relationships with the policy type as "sync" and "sync_type" as "automated_failover" cannot be "paused". To resume transfers for a paused relationship, PATCH the state to "snapmirrored" for relationships with a policy of type "async" or to state "in_sync" for relationships with a policy of type "sync". The entries "in_sync", "out_of_sync", "synchronizing", and "expanding" are only applicable to relationships with a policy of type "sync". A PATCH call on the state change only triggers the transition to the specified state. You must poll on the "state", "healthy" and "unhealthy_reason" properties using a GET request to determine if the transition is successful. To automatically initialize the relationship when specifying "create_destination" property, set the state to "snapmirrored" for</p> |

| Name                    | Type                    | Description   |
|-------------------------|-------------------------|---|
| throttle                | integer                 | Throttle, in KBs per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited.   |
| total_transfer_bytes    | integer                 | Cumulative bytes transferred for the relationship.  |
| total_transfer_duration | string                  | Indicates the cumulative duration of all transfers since the last aggregate relocation, takeover/giveback, or metrocluster switchover/switchback involving the node that hosts the relationship.  |
| transfer                | transfer                | Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.   |
| transfer_schedule       | transfer_schedule       | Schedule used to update asynchronous relationships. This "transfer_schedule" overrides the "transfer_schedule" set on the SnapMirror relationship's policy. To remove the "transfer_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror. |
| unhealthy_reason        | array[snapmirror_error] | Reason the relationship is not healthy. It is a concatenation of up to four levels of error messages.   |
| uuid                    | string                  | Unique identifier of the SnapMirror relationship.   |

job\_link

| Name | Type   | Description   |
|------|--------|---|
| uuid | string | The UUID of the asynchronous job that is triggered by a POST, PATCH, or DELETE operation. |

returned\_error

| Name      | Type                   | Description                                 |
|-----------|------------------------|---|
| arguments | array[error_arguments] | Message arguments                           |
| code      | string                 | Error code                                  |
| message   | string                 | Error message                               |
| target    | string                 | The target parameter that caused the error. |

## Manage SnapMirror relationship transfers

### SnapMirror relationships relationship.uuid transfers endpoint overview

#### Overview

This API is used to manage transfers on an existing SnapMirror relationship.

You can initiate SnapMirror operations such as "initialize", "update", "restore-transfer", and "abort" using this API on asynchronous SnapMirror relationship. On a synchronous SnapMirror relationship, you can initiate SnapMirror "initialize" operation. The GET for this API reports the status of both active transfers and transfers that have terminated within the past 24 hours. For the restore relationships, the POST on transfers API triggers "restore-transfer". Successful completion of "restore" also deletes the restore relationship. If the "restore" fails, DELETE on relationships must be called to delete the restore relationship.

A transfer on an asynchronous SnapMirror relationship with Application Consistency Group endpoints expands the destination Application Consistency Group endpoint if the source Application Consistency Group endpoint is already expanded.

### Retrieve ongoing SnapMirror transfers for a relationship

GET /snapmirror/relationships/{relationship.uuid}/transfers

**Introduced In:** 9.6

Retrieves the list of ongoing SnapMirror transfers for the specified relationship.

#### Related ONTAP commands

- snapmirror show

## Example

```
GET "/api/snapmirror/relationships/293baa53-e63d-11e8-bff1-005056a793dd/transfers"
```

## Learn more

- [DOC /snapmirror/relationships/{relationship.uuid}/transfers](#)

## Parameters

| Name  | Type    | In    | Required | Description  |
|---|---------|-------|----------|--|
| relationship.uuid                                       | string  | path  | True     | SnapMirror relationship UUID   |
| bytes_transferred                                       | integer | query | False    | Filter by bytes_transferred  |
| relationship.destination.consistency_group_volumes.name | string  | query | False    | Filter by relationship.destination.consistency_group_volumes.name <ul style="list-style-type: none"><li>• Introduced in: 9.8</li></ul> |
| relationship.destination.cluster.uuid                   | string  | query | False    | Filter by relationship.destination.cluster.uuid <ul style="list-style-type: none"><li>• Introduced in: 9.7</li></ul>                   |
| relationship.destination.cluster.name                   | string  | query | False    | Filter by relationship.destination.cluster.name <ul style="list-style-type: none"><li>• Introduced in: 9.7</li></ul>                   |
| relationship.destination.uuid                           | string  | query | False    | Filter by relationship.destination.uuid <ul style="list-style-type: none"><li>• Introduced in: 9.11</li></ul>                          |

| Name                              | Type    | In    | Required | Description  |
|-----------------------------------|---------|-------|----------|--|
| relationship.destination.svm.uuid | string  | query | False    | Filter by relationship.destination.svm.uuid  |
| relationship.destination.svm.name | string  | query | False    | Filter by relationship.destination.svm.name  |
| relationship.destination.path     | string  | query | False    | Filter by relationship.destination.path  |
| relationship.restore              | boolean | query | False    | Filter by relationship.restore   |
| error_info.message                | string  | query | False    | Filter by error_info.message <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> </ul> |
| error_info.code                   | integer | query | False    | Filter by error_info.code <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> </ul>    |
| end_time                          | string  | query | False    | Filter by end_time <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> </ul>           |
| uuid                              | string  | query | False    | Filter by uuid   |
| state                             | string  | query | False    | Filter by state  |
| on_demand_attrs                   | string  | query | False    | Filter by on_demand_attrs <ul style="list-style-type: none"> <li>• Introduced in: 9.13</li> </ul>    |
| throttle                          | integer | query | False    | Filter by throttle <ul style="list-style-type: none"> <li>• Introduced in: 9.9</li> </ul>            |

| Name                      | Type          | In    | Required | Description  |
|---------------------------|---------------|-------|----------|--|
| network_compression_ratio | string        | query | False    | Filter by network_compression_ratio <ul style="list-style-type: none"> <li>• Introduced in: 9.13</li> </ul>  |
| checkpoint_size           | integer       | query | False    | Filter by checkpoint_size  |
| snapshot                  | string        | query | False    | Filter by snapshot   |
| total_duration            | string        | query | False    | Filter by total_duration <ul style="list-style-type: none"> <li>• Introduced in: 9.10</li> </ul>   |
| last_updated_time         | string        | query | False    | Filter by last_updated_time <ul style="list-style-type: none"> <li>• Introduced in: 9.14</li> </ul>  |
| fields                    | array[string] | query | False    | Specify the fields to return.  |
| max_records               | integer       | query | False    | Limit the number of records returned.  |
| return_records            | boolean       | query | False    | The default is true for GET calls. When set to false, only the number of records is returned. <ul style="list-style-type: none"> <li>• Default value: 1</li> </ul> |

| Name           | Type          | In    | Required | Description  |
|----------------|---------------|-------|----------|--|
| return_timeout | integer       | query | False    | <p>The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached.</p> <ul style="list-style-type: none"> <li>• Default value: 1</li> <li>• Max value: 120</li> <li>• Min value: 0</li> </ul> |
| order_by       | array[string] | query | False    | Order results by specified fields and optional [asc]   |

## Response

Status: 200, Ok

| Name        | Type   | Description       |
|-------------|--|-------------------|
| _links      | <a href="#">_links</a>                       |                   |
| num_records | integer                                      | Number of records |
| records     | array[ <a href="#">snapmirror_transfer</a> ] |                   |

## Example response

```
{  
  "_links": {  
    "next": {  
      "href": "/api/resourcelink"  
    },  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "num_records": 1,  
  "records": [  
    {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "bytes_transferred": 0,  
      "checkpoint_size": 0,  
      "end_time": "2020-12-02 21:36:19 -0500",  
      "error_info": {  
        "code": 6620046,  
        "message": "Transfer aborted"  
      },  
      "last_updated_time": "2023-09-15 19:58:39 -0400",  
      "network_compression_ratio": 61,  
      "on_demandAttrs": "read_write_with_user_data_pull",  
      "relationship": {  
        "destination": {  
          "cluster": {  
            "_links": {  
              "self": {  
                "href": "/api/resourcelink"  
              }  
            },  
            "name": "cluster1",  
            "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
          },  
          "consistency_group_volumes": [  
            {  
              "name": "volume1"  
            }  
          ],  
          "path": "svm1:volume1",  
          "type": "cluster"  
        }  
      }  
    }  
  ]  
}
```

```

"svm": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "svml",
  "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
},
{
  "uuid": "d2d7ceea-ab52-11e8-855e-00505682a4c7"
},
"snapshot": "string",
"state": "string",
"storage_efficiency_enabled": true,
"throttle": 0,
"total_duration": "PT28M41S",
"uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
}
]
}

```

## Error

Status: Default, Error

| Name  | Type           | Description |
|-------|----------------|-------------|
| error | returned_error |             |

## Example error

```
{  
  "error": {  
    "arguments": [  
      {  
        "code": "string",  
        "message": "string"  
      }  
    ],  
    "code": "4",  
    "message": "entry doesn't exist",  
    "target": "uuid"  
  }  
}
```

## Definitions

## See Definitions

href

| Name | Type   | Description |
|------|--------|-------------|
| href | string |             |

\_links

| Name | Type                 | Description |
|------|----------------------|-------------|
| next | <a href="#">href</a> |             |
| self | <a href="#">href</a> |             |

\_links

| Name | Type                 | Description |
|------|----------------------|-------------|
| self | <a href="#">href</a> |             |

error\_info

Error information for the transfer.

| Name    | Type    | Description   |
|---------|---------|---------------|
| code    | integer | Error code    |
| message | string  | Error message |

files

Specifies a file or LUN consisting of a source\_path and an optional destination\_path. If not specified, the destination\_path is the same as the source\_path. File restore is not supported if the source\_path or destination\_path contains commas in its directory or file name.

| Name             | Type   | Description |
|------------------|--------|-------------|
| destination_path | string |             |
| source_path      | string |             |

options

cluster

| Name   | Type                   | Description |
|--------|------------------------|-------------|
| _links | <a href="#">_links</a> |             |
| name   | string                 |             |
| uuid   | string                 |             |

## consistency\_group\_volumes

| Name | Type   | Description             |
|------|--------|-------------------------|
| name | string | The name of the volume. |

## svm

SVM, applies only to SVM-scoped objects.

| Name   | Type                   | Description   |
|--------|------------------------|---|
| _links | <a href="#">_links</a> |   |
| name   | string                 | The name of the SVM. This field cannot be specified in a PATCH method.              |
| uuid   | string                 | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

## snapmirror\_endpoint

Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.

| Name                      | Type   | Description   |
|---------------------------|--|---|
| cluster                   | <a href="#">cluster</a>                            |   |
| consistency_group_volumes | array[ <a href="#">consistency_group_volumes</a> ] | Mandatory property for a Consistency Group endpoint. Specifies the list of FlexVol volumes for a Consistency Group. |

| Name | Type   | Description   |
|------|--------|---|
| path | string | <p>ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name</p> <p>ONTAP S3 - svm1:/bucket/bucket1 NON-ONTAP - objstore1:/objstore</p> <ul style="list-style-type: none"> <li>• example: svm1:volume1</li> <li>• Introduced in: 9.6</li> <li>• x-nullable: true</li> </ul> |
| svm  | svm    | SVM, applies only to SVM-scoped objects.  |

#### relationship

| Name        | Type                                | Description   |
|-------------|-------------------------------------|---|
| destination | <a href="#">snapmirror_endpoint</a> | <p>Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> </ul> |
| restore     | boolean                             | Is the relationship for restore?  |

| Name | Type   | Description                                       |
|------|--------|---|
| uuid | string | Unique identifier of the SnapMirror relationship. |

## snapmirror\_transfer

SnapMirror transfer information

| Name                      | Type                         | Description   |
|---------------------------|------------------------------|---|
| _links                    | <a href="#">_links</a>       |   |
| bytes_transferred         | integer                      | Bytes transferred   |
| checkpoint_size           | integer                      | Amount of data transferred in bytes as recorded in the restart checkpoint.  |
| end_time                  | string                       | End time of the transfer.   |
| error_info                | <a href="#">error_info</a>   | Error information for the transfer.   |
| last_updated_time         | string                       | Last updated time of the bytes transferred in an active transfer.   |
| network_compression_ratio | string                       | Specifies the compression ratio achieved for the data sent over the wire with network compression enabled. This property is only valid for active transfers.  |
| on_demandAttrs            | string                       | Specifies whether or not an on-demand restore is being carried out. This is only supported for the transfer of restore relationships for entire volumes from the object store. A value for <code>read_write_with_user_data_pull</code> should be provided to start an on-demand restore. A file restore from the object store does not support this option. |
| relationship              | <a href="#">relationship</a> |   |
| snapshot                  | string                       | Name of Snapshot copy being transferred.  |

| Name           | Type    | Description  |
|----------------|---------|--|
| state          | string  | Status of the transfer. Set PATCH state to "aborted" to abort the transfer. Set PATCH state to "hard_aborted" to abort the transfer and discard the restart checkpoint. To find "queued" transfers refer to relationships GET API. |
| throttle       | integer | Throttle, in KBs per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship or SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited.       |
| total_duration | string  | Elapsed transfer time.   |
| uuid           | string  | Unique identifier of the SnapMirror transfer.  |

#### error\_arguments

| Name    | Type   | Description      |
|---------|--------|------------------|
| code    | string | Argument code    |
| message | string | Message argument |

#### returned\_error

| Name      | Type                   | Description                                 |
|-----------|------------------------|---|
| arguments | array[error_arguments] | Message arguments                           |
| code      | string                 | Error code                                  |
| message   | string                 | Error message                               |
| target    | string                 | The target parameter that caused the error. |

## Start a SnapMirror transfer operation

POST /snapmirror/relationships/{relationship.uuid}/transfers

**Introduced In:** 9.6

Starts a SnapMirror transfer operation. This API initiates a restore operation if the SnapMirror relationship is of type "restore". Otherwise, it initiates a SnapMirror "initialize" operation or "update" operation based on the current SnapMirror state.

### Default property values

- storage\_efficiency\_enabled - *true*

### Related ONTAP commands

- snapmirror update
- snapmirror initialize
- snapmirror restore

### Examples

The following examples show how to perform SnapMirror "initialize", "update", and "restore" operations.

Perform SnapMirror initialize or update

```
POST "/api/snapmirror/relationships/e4e7e130-0279-11e9-b566-0050568e9909/transfers" '{}'
```

Perform SnapMirror initialize, update or restore with throttle value set

```
POST "/api/snapmirror/relationships/e4e7e130-0279-11e9-b566-0050568e9909/transfers" '{"throttle":"100"}'
```

Perform SnapMirror restore transfer of a file

```
POST "/api/snapmirror/relationships/c8c62a90-0fef-11e9-b09e-0050568e7067/transfers" '{"source_snapshot": "src", "files": [{"source_path": "/a1.txt.0", "destination_path": "/a1-renamed.txt.0"}]}'
```

Performing a SnapMirror initialize or update using a particular Snapshot copy.

```
POST "/api/snapmirror/relationships/e4e7e130-0279-11e9-b566-0050568e9909/transfers" {"source_snapshot": "snap1"}
```

## Learn more

- [DOC /snapmirror/relationships/{relationship.uuid}/transfers](#)

## Parameters

| Name              | Type    | In    | Required | Description  |
|-------------------|---------|-------|----------|--|
| relationship.uuid | string  | path  | True     | SnapMirror relationship UUID   |
| return_records    | boolean | query | False    | <p>The default is false. If set to true, the records are returned.</p> <ul style="list-style-type: none"><li>• Default value:</li></ul>  |
| return_timeout    | integer | query | False    | <p>The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202.</p> <ul style="list-style-type: none"><li>• Default value: 1</li><li>• Max value: 120</li><li>• Min value: 0</li></ul> |

## Request Body

| Name                      | Type                             | Description   |
|---------------------------|----------------------------------|---|
| bytes_transferred         | integer                          | Bytes transferred   |
| checkpoint_size           | integer                          | Amount of data transferred in bytes as recorded in the restart checkpoint.  |
| end_time                  | string                           | End time of the transfer.   |
| error_info                | <a href="#">error_info</a>       | Error information for the transfer.   |
| files                     | array[ <a href="#">files</a> ]   | This is supported for transfer of restore relationship only. This specifies the list of files or LUNs to be restored. Can contain up to eight files or LUNs.  |
| last_updated_time         | string                           | Last updated time of the bytes transferred in an active transfer.   |
| network_compression_ratio | string                           | Specifies the compression ratio achieved for the data sent over the wire with network compression enabled. This property is only valid for active transfers.  |
| on_demand_attrs           | string                           | Specifies whether or not an on-demand restore is being carried out. This is only supported for the transfer of restore relationships for entire volumes from the object store. A value for <code>read_write_with_user_data_pull</code> should be provided to start an on-demand restore. A file restore from the object store does not support this option. |
| options                   | array[ <a href="#">options</a> ] | Options for snapmirror transfer.  |
| relationship              | <a href="#">relationship</a>     |   |
| snapshot                  | string                           | Name of Snapshot copy being transferred.  |

| Name                       | Type    | Description  |
|----------------------------|---------|--|
| source_snapshot            | string  | Specifies the Snapshot copy on the source to be transferred to the destination.  |
| storage_efficiency_enabled | boolean | This is supported for transfer of restore relationship only. Set this property to "false" to turn off storage efficiency for data transferred over the wire and written to the destination.                                  |
| throttle                   | integer | Throttle, in KBs per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship or SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited. |
| total_duration             | string  | Elapsed transfer time.   |
| uuid                       | string  | Unique identifier of the SnapMirror transfer.  |

## Example request

```
{  
    "bytes_transferred": 0,  
    "checkpoint_size": 0,  
    "end_time": "2020-12-02 21:36:19 -0500",  
    "error_info": {  
        "code": 6620046,  
        "message": "Transfer aborted"  
    },  
    "files": [  
        {  
            "destination_path": "/dirb/file2",  
            "source_path": "/dira/file1"  
        }  
    ],  
    "last_updated_time": "2023-09-15 19:58:39 -0400",  
    "network_compression_ratio": 61,  
    "on_demand_attrs": "read_write_with_user_data_pull",  
    "options": [  
        {}  
    ],  
    "relationship": {  
        "destination": {  
            "cluster": {  
                "name": "cluster1",  
                "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
            },  
            "consistency_group_volumes": [  
                {  
                    "name": "volume1",  
                    "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"  
                }  
            ],  
            "ipspace": "Default",  
            "path": "svm1:volume1",  
            "svm": {  
                "name": "svm1",  
                "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
            }  
        },  
        "uuid": "d2d7ceea-ab52-11e8-855e-00505682a4c7"  
    },  
    "snapshot": "string",  
    "source_snapshot": "string",  
    "throttle": 0,  
    "type": "file_copy"  
}
```

```

  "total_duration": "PT28M41S",
  "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
}

```

## Response

Status: 201, Created

## Headers

| Name     | Description                               | Type   |
|----------|---|--------|
| Location | Useful for tracking the resource location | string |

## Error

Status: Default

## ONTAP Error Response codes

| Error code | Description   |
|------------|---|
| 13303845   | Restore operation failed  |
| 13303812   | Initialize operation failed   |
| 13303844   | Update operation failed   |
| 13303846   | Empty source path file list   |
| 13303847   | Invalid arguments   |
| 13304040   | Throttle not supported for update of Synchronous SnapMirror relationships |
| 6620237    | SnapMirror relationship database write failed                             |
| 6620238    | SnapMirror relationship database read failed                              |

## Definitions

## See Definitions

href

| Name | Type   | Description |
|------|--------|-------------|
| href | string |             |

\_links

error\_info

Error information for the transfer.

| Name    | Type    | Description   |
|---------|---------|---------------|
| code    | integer | Error code    |
| message | string  | Error message |

files

Specifies a file or LUN consisting of a source\_path and an optional destination\_path. If not specified, the destination\_path is the same as the source\_path. File restore is not supported if the source\_path or destination\_path contains commas in its directory or file name.

| Name             | Type   | Description |
|------------------|--------|-------------|
| destination_path | string |             |
| source_path      | string |             |

options

cluster

| Name | Type   | Description |
|------|--------|-------------|
| name | string |             |
| uuid | string |             |

consistency\_group\_volumes

| Name | Type   | Description             |
|------|--------|-------------------------|
| name | string | The name of the volume. |

| Name | Type   | Description   |
|------|--------|---|
| uuid | string | <p>Unique identifier of the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move.</p> <ul style="list-style-type: none"> <li>example: 028baa66-41bd-11e9-81d5-00a0986138f7</li> <li>x-nitap-createOnly: true</li> <li>Introduced in: 9.8</li> <li>x-nullable: true</li> </ul> |

## svm

SVM, applies only to SVM-scoped objects.

| Name | Type   | Description   |
|------|--------|---|
| name | string | The name of the SVM. This field cannot be specified in a PATCH method.              |
| uuid | string | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

## snapmirror\_endpoint

Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.

| Name                      | Type                              | Description   |
|---------------------------|-----------------------------------|---|
| cluster                   | cluster                           |   |
| consistency_group_volumes | array[consistency_group_volume_s] | Mandatory property for a Consistency Group endpoint. Specifies the list of FlexVol volumes for a Consistency Group. |

| Name    | Type                | Description  |
|---------|---------------------|--|
| ipspace | string              | Optional property to specify the IPSpace of the SVM.   |
| path    | string              | <p>ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name</p> <p>ONTAP S3 - svm1:/bucket/bucket1</p> <p>NON-ONTAP - objstore1:/objstore</p> <ul style="list-style-type: none"> <li>example: svm1:volume1</li> <li>Introduced in: 9.6</li> <li>x-nullable: true</li> </ul> |
| svm     | <a href="#">svm</a> | SVM, applies only to SVM-scoped objects.   |

relationship

| Name        | Type                                | Description   |
|-------------|-------------------------------------|---|
| destination | <a href="#">snapmirror_endpoint</a> | <p>Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> </ul> |
| restore     | boolean                             | Is the relationship for restore?  |
| uuid        | string                              | Unique identifier of the SnapMirror relationship.   |

### `snapmirror_transfer`

SnapMirror transfer information

| Name              | Type                       | Description  |
|-------------------|----------------------------|--|
| bytes_transferred | integer                    | Bytes transferred  |
| checkpoint_size   | integer                    | Amount of data transferred in bytes as recorded in the restart checkpoint. |
| end_time          | string                     | End time of the transfer.  |
| error_info        | <a href="#">error_info</a> | Error information for the transfer.  |

| Name                       | Type                             | Description   |
|----------------------------|----------------------------------|---|
| files                      | array[ <a href="#">files</a> ]   | This is supported for transfer of restore relationship only. This specifies the list of files or LUNs to be restored. Can contain up to eight files or LUNs.  |
| last_updated_time          | string                           | Last updated time of the bytes transferred in an active transfer.   |
| network_compression_ratio  | string                           | Specifies the compression ratio achieved for the data sent over the wire with network compression enabled. This property is only valid for active transfers.  |
| on_demandAttrs             | string                           | Specifies whether or not an on-demand restore is being carried out. This is only supported for the transfer of restore relationships for entire volumes from the object store. A value for <code>read_write_with_user_data_pull</code> should be provided to start an on-demand restore. A file restore from the object store does not support this option. |
| options                    | array[ <a href="#">options</a> ] | Options for snapmirror transfer.  |
| relationship               | <a href="#">relationship</a>     |   |
| snapshot                   | string                           | Name of Snapshot copy being transferred.  |
| source_snapshot            | string                           | Specifies the Snapshot copy on the source to be transferred to the destination.   |
| storage_efficiency_enabled | boolean                          | This is supported for transfer of restore relationship only. Set this property to "false" to turn off storage efficiency for data transferred over the wire and written to the destination.   |

| Name           | Type    | Description  |
|----------------|---------|--|
| throttle       | integer | Throttle, in KBs per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship or SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited. |
| total_duration | string  | Elapsed transfer time.   |
| uuid           | string  | Unique identifier of the SnapMirror transfer.  |

#### error\_arguments

| Name    | Type   | Description      |
|---------|--------|------------------|
| code    | string | Argument code    |
| message | string | Message argument |

#### returned\_error

| Name      | Type                   | Description                                 |
|-----------|------------------------|---|
| arguments | array[error_arguments] | Message arguments                           |
| code      | string                 | Error code                                  |
| message   | string                 | Error message                               |
| target    | string                 | The target parameter that caused the error. |

## Retrieve attributes of an ongoing SnapMirror transfer

GET /snapmirror/relationships/{relationship.uuid}/transfers/{uuid}

**Introduced In:** 9.6

Retrieves the attributes of a specific ongoing SnapMirror transfer.

### Related ONTAP commands

- snapmirror show

## Example

```
GET "/api/snapmirror/relationships/293baa53-e63d-11e8-bff1-005056a793dd/transfers/293baa53-e63d-11e8-bff1-005056a793dd"
```

## Learn more

- [DOC /snapmirror/relationships/{relationship.uuid}/transfers](#)

## Parameters

| Name              | Type          | In    | Required | Description                   |
|-------------------|---------------|-------|----------|-------------------------------|
| relationship.uuid | string        | path  | True     | SnapMirror relationship UUID  |
| uuid              | string        | path  | True     | SnapMirror transfer UUID      |
| fields            | array[string] | query | False    | Specify the fields to return. |

## Response

```
Status: 200, Ok
```

| Name              | Type                       | Description  |
|-------------------|----------------------------|--|
| _links            | <a href="#">_links</a>     |  |
| bytes_transferred | integer                    | Bytes transferred  |
| checkpoint_size   | integer                    | Amount of data transferred in bytes as recorded in the restart checkpoint. |
| end_time          | string                     | End time of the transfer.  |
| error_info        | <a href="#">error_info</a> | Error information for the transfer.  |
| last_updated_time | string                     | Last updated time of the bytes transferred in an active transfer.          |

| Name                      | Type         | Description  |
|---------------------------|--------------|--|
| network_compression_ratio | string       | Specifies the compression ratio achieved for the data sent over the wire with network compression enabled. This property is only valid for active transfers.   |
| on_demand_attrs           | string       | Specifies whether or not an on-demand restore is being carried out. This is only supported for the transfer of restore relationships for entire volumes from the object store. A value for read_write_with_user_data_pull should be provided to start an on-demand restore. A file restore from the object store does not support this option. |
| relationship              | relationship |  |
| snapshot                  | string       | Name of Snapshot copy being transferred.   |
| state                     | string       | Status of the transfer. Set PATCH state to "aborted" to abort the transfer. Set PATCH state to "hard_aborted" to abort the transfer and discard the restart checkpoint. To find "queued" transfers refer to relationships GET API.   |
| throttle                  | integer      | Throttle, in KBs per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship or SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited.   |
| total_duration            | string       | Elapsed transfer time.   |
| uuid                      | string       | Unique identifier of the SnapMirror transfer.  |

## Example response

```
{  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "bytes_transferred": 0,  
  "checkpoint_size": 0,  
  "end_time": "2020-12-02 21:36:19 -0500",  
  "error_info": {  
    "code": 6620046,  
    "message": "Transfer aborted"  
  },  
  "last_updated_time": "2023-09-15 19:58:39 -0400",  
  "network_compression_ratio": 61,  
  "on_demand_attrs": "read_write_with_user_data_pull",  
  "relationship": {  
    "destination": {  
      "cluster": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "name": "cluster1",  
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
      },  
      "consistency_group_volumes": [  
        {  
          "name": "volumel"  
        }  
      ],  
      "path": "svm1:volumel",  
      "svm": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "name": "svm1",  
        "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
      }  
    },  
    "uuid": "d2d7ceea-ab52-11e8-855e-00505682a4c7"  
  }  
}
```

```

} ,
"snapshot": "string",
"state": "string",
"storage_efficiency_enabled": true,
"throttle": 0,
"total_duration": "PT28M41S",
"uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
}

```

## Error

Status: Default, Error

| Name  | Type           | Description |
|-------|----------------|-------------|
| error | returned_error |             |

### Example error

```

{
  "error": {
    "arguments": [
      {
        "code": "string",
        "message": "string"
      }
    ],
    "code": "4",
    "message": "entry doesn't exist",
    "target": "uuid"
  }
}

```

## Definitions

## See Definitions

href

| Name | Type   | Description |
|------|--------|-------------|
| href | string |             |

\_links

| Name | Type | Description |
|------|------|-------------|
| self | href |             |

error\_info

Error information for the transfer.

| Name    | Type    | Description   |
|---------|---------|---------------|
| code    | integer | Error code    |
| message | string  | Error message |

files

Specifies a file or LUN consisting of a source\_path and an optional destination\_path. If not specified, the destination\_path is the same as the source\_path. File restore is not supported if the source\_path or destination\_path contains commas in its directory or file name.

| Name             | Type   | Description |
|------------------|--------|-------------|
| destination_path | string |             |
| source_path      | string |             |

options

cluster

| Name   | Type   | Description |
|--------|--------|-------------|
| _links | _links |             |
| name   | string |             |
| uuid   | string |             |

consistency\_group\_volumes

| Name | Type   | Description             |
|------|--------|-------------------------|
| name | string | The name of the volume. |

## svm

SVM, applies only to SVM-scoped objects.

| Name   | Type                   | Description   |
|--------|------------------------|---|
| _links | <a href="#">_links</a> |   |
| name   | string                 | The name of the SVM. This field cannot be specified in a PATCH method.              |
| uuid   | string                 | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

## snapmirror\_endpoint

Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.

| Name                      | Type   | Description   |
|---------------------------|--|---|
| cluster                   | <a href="#">cluster</a>                          |   |
| consistency_group_volumes | <a href="#">array[consistency_group_volumes]</a> | Mandatory property for a Consistency Group endpoint. Specifies the list of FlexVol volumes for a Consistency Group.   |
| path                      | string   | ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name<br>ONTAP S3 - svm1:/bucket/bucket1 NON-ONTAP - objstore1:/objstore<br><ul style="list-style-type: none"><li>example: svm1:volume1</li><li>Introduced in: 9.6</li><li>x-nullable: true</li></ul> |
| svm                       | <a href="#">svm</a>                              | SVM, applies only to SVM-scoped objects.  |

## relationship

| Name        | Type                                | Description   |
|-------------|-------------------------------------|---|
| destination | <a href="#">snapmirror_endpoint</a> | <p>Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> </ul> |
| restore     | boolean                             | Is the relationship for restore?  |
| uuid        | string                              | Unique identifier of the SnapMirror relationship.   |

## error\_arguments

| Name    | Type   | Description      |
|---------|--------|------------------|
| code    | string | Argument code    |
| message | string | Message argument |

## returned\_error

| Name      | Type                                   | Description       |
|-----------|--|-------------------|
| arguments | <a href="#">array[error_arguments]</a> | Message arguments |

| Name    | Type   | Description                                 |
|---------|--------|---|
| code    | string | Error code                                  |
| message | string | Error message                               |
| target  | string | The target parameter that caused the error. |

## Cancel an ongoing SnapMirror transfer

PATCH /snapmirror/relationships/{relationship.uuid}/transfers/{uuid}

**Introduced In:** 9.6

Aborts an ongoing SnapMirror transfer. This operation is applicable on asynchronous SnapMirror relationships.

### Related ONTAP commands

- snapmirror abort

### Example

```
PATCH "/api/snapmirror/relationships/293baa53-e63d-11e8-bff1-005056a793dd/transfers/293baa53-e63d-11e8-bff1-005056a793dd"
'{"state":"aborted"}'
```

### Learn more

- [DOC /snapmirror/relationships/{relationship.uuid}/transfers](#)

### Parameters

| Name              | Type   | In   | Required | Description                  |
|-------------------|--------|------|----------|------------------------------|
| relationship.uuid | string | path | True     | SnapMirror relationship UUID |
| uuid              | string | path | True     | SnapMirror transfer UUID     |

### Request Body

| Name              | Type    | Description       |
|-------------------|---------|-------------------|
| bytes_transferred | integer | Bytes transferred |

| Name                      | Type                         | Description   |
|---------------------------|------------------------------|---|
| checkpoint_size           | integer                      | Amount of data transferred in bytes as recorded in the restart checkpoint.  |
| end_time                  | string                       | End time of the transfer.   |
| error_info                | <a href="#">error_info</a>   | Error information for the transfer.   |
| last_updated_time         | string                       | Last updated time of the bytes transferred in an active transfer.   |
| network_compression_ratio | string                       | Specifies the compression ratio achieved for the data sent over the wire with network compression enabled. This property is only valid for active transfers.  |
| on_demand_attrs           | string                       | Specifies whether or not an on-demand restore is being carried out. This is only supported for the transfer of restore relationships for entire volumes from the object store. A value for <code>read_write_with_user_data_pull</code> should be provided to start an on-demand restore. A file restore from the object store does not support this option. |
| relationship              | <a href="#">relationship</a> |   |
| snapshot                  | string                       | Name of Snapshot copy being transferred.  |
| state                     | string                       | Status of the transfer. Set PATCH state to "aborted" to abort the transfer. Set PATCH state to "hard_aborted" to abort the transfer and discard the restart checkpoint. To find "queued" transfers refer to relationships GET API.  |
| total_duration            | string                       | Elapsed transfer time.  |
| uuid                      | string                       | Unique identifier of the SnapMirror transfer.   |

## Example request

```
{  
    "bytes_transferred": 0,  
    "checkpoint_size": 0,  
    "end_time": "2020-12-02 21:36:19 -0500",  
    "error_info": {  
        "code": 6620046,  
        "message": "Transfer aborted"  
    },  
    "last_updated_time": "2023-09-15 19:58:39 -0400",  
    "network_compression_ratio": 61,  
    "on_demand_attrs": "read_write_with_user_data_pull",  
    "relationship": {  
        "destination": {  
            "cluster": {  
                "name": "cluster1",  
                "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"  
            },  
            "path": "svm1:volume1",  
            "svm": {  
                "name": "svm1",  
                "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
            }  
        },  
        "uuid": "d2d7ceea-ab52-11e8-855e-00505682a4c7"  
    },  
    "snapshot": "string",  
    "state": "string",  
    "storage_efficiency_enabled": true,  
    "total_duration": "PT28M41S",  
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"  
}
```

## Response

```
Status: 200, Ok
```

## Error

```
Status: Default
```

## ONTAP Error Response codes

| Error code | Description                             |
|------------|---|
| 13303848   | Abort of sync SnapMirror is not allowed |
| 13303849   | SnapMirror transfer state is invalid    |

## Definitions

## See Definitions

href

| Name | Type   | Description |
|------|--------|-------------|
| href | string |             |

\_links

error\_info

Error information for the transfer.

| Name    | Type    | Description   |
|---------|---------|---------------|
| code    | integer | Error code    |
| message | string  | Error message |

files

Specifies a file or LUN consisting of a source\_path and an optional destination\_path. If not specified, the destination\_path is the same as the source\_path. File restore is not supported if the source\_path or destination\_path contains commas in its directory or file name.

| Name             | Type   | Description |
|------------------|--------|-------------|
| destination_path | string |             |
| source_path      | string |             |

options

cluster

| Name | Type   | Description |
|------|--------|-------------|
| name | string |             |
| uuid | string |             |

consistency\_group\_volumes

| Name | Type   | Description             |
|------|--------|-------------------------|
| name | string | The name of the volume. |

svm

SVM, applies only to SVM-scoped objects.

| Name | Type   | Description   |
|------|--------|---|
| name | string | The name of the SVM. This field cannot be specified in a PATCH method.              |
| uuid | string | The unique identifier of the SVM. This field cannot be specified in a PATCH method. |

### snapmirror\_endpoint

Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.

| Name    | Type    | Description   |
|---------|---------|---|
| cluster | cluster |   |
| path    | string  | ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name ONTAP S3 - svm1:/bucket/bucket1 NON-ONTAP - objstore1:/objstore <ul style="list-style-type: none"> <li>example: svm1:volume1</li> <li>Introduced in: 9.6</li> <li>x-nullable: true</li> </ul> |
| svm     | svm     | SVM, applies only to SVM-scoped objects.  |

### relationship

| Name        | Type                                | Description   |
|-------------|-------------------------------------|---|
| destination | <a href="#">snapmirror_endpoint</a> | <p>Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint.</p> <ul style="list-style-type: none"> <li>• Introduced in: 9.6</li> </ul> |
| restore     | boolean                             | Is the relationship for restore?  |
| uuid        | string                              | Unique identifier of the SnapMirror relationship.   |

### `snapmirror_transfer`

SnapMirror transfer information

| Name              | Type                       | Description  |
|-------------------|----------------------------|--|
| bytes_transferred | integer                    | Bytes transferred  |
| checkpoint_size   | integer                    | Amount of data transferred in bytes as recorded in the restart checkpoint. |
| end_time          | string                     | End time of the transfer.  |
| error_info        | <a href="#">error_info</a> | Error information for the transfer.  |

| Name                      | Type         | Description   |
|---------------------------|--------------|---|
| last_updated_time         | string       | Last updated time of the bytes transferred in an active transfer.   |
| network_compression_ratio | string       | Specifies the compression ratio achieved for the data sent over the wire with network compression enabled. This property is only valid for active transfers.  |
| on_demand_attrs           | string       | Specifies whether or not an on-demand restore is being carried out. This is only supported for the transfer of restore relationships for entire volumes from the object store. A value for <code>read_write_with_user_data_pull</code> should be provided to start an on-demand restore. A file restore from the object store does not support this option. |
| relationship              | relationship |   |
| snapshot                  | string       | Name of Snapshot copy being transferred.  |
| state                     | string       | Status of the transfer. Set PATCH state to "aborted" to abort the transfer. Set PATCH state to "hard_aborted" to abort the transfer and discard the restart checkpoint. To find "queued" transfers refer to relationships GET API.  |
| total_duration            | string       | Elapsed transfer time.  |
| uuid                      | string       | Unique identifier of the SnapMirror transfer.   |

#### error\_arguments

| Name    | Type   | Description      |
|---------|--------|------------------|
| code    | string | Argument code    |
| message | string | Message argument |

## returned\_error

| Name      | Type                   | Description                                 |
|-----------|------------------------|---|
| arguments | array[error_arguments] | Message arguments                           |
| code      | string                 | Error code                                  |
| message   | string                 | Error message                               |
| target    | string                 | The target parameter that caused the error. |

## Copyright information

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

**LIMITED RIGHTS LEGEND:** Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

## Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.