



Manage SnapMirror relationships

REST API reference

NetApp
February 11, 2026

This PDF was generated from https://docs.netapp.com/us-en/ontap-restapi-96/manage_snapmirror_relationships.html on February 11, 2026. Always check docs.netapp.com for the latest.

Table of Contents

Manage SnapMirror relationships	1
Manage SnapMirror relationships	1
Overview	1
Retrieves information for snapmirror relationships whose destination endpoints are in the current svm or the current cluster, depending on the cluster context	1
Related ONTAP commands	1
Examples	2
Learn more	2
Parameters	2
Response	4
Error	6
Definitions	7
Creates a snapmirror relationship this api must be executed on the cluster containing the destination endpoint	12
Required properties	13
Recommended optional properties	13
Default property values	13
Related ONTAP commands	13
Examples	13
Learn more	14
Parameters	14
Request Body	14
Response	18
Error	19
Definitions	19
Deletes a snapmirror relationship	23
Important notes	24
Related ONTAP commands	24
Examples	24
Learn more	25
Parameters	25
Response	27
Error	27
Definitions	28
Retrieves a snapmirror relationship	29
Related ONTAP commands	30
Example	30
Learn more	30
Parameters	30
Response	30
Error	34
Definitions	35
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	38
Related ONTAP commands	38
Examples	39
Learn more	39
Parameters	40
Request Body	40
Response	44
Error	45
Definitions	46

Manage SnapMirror relationships

Manage SnapMirror relationships

Overview

This API manages asynchronous extended data protection (XDP) relationships for FlexVols, FlexGroups, or SVMs. It is also used to create a synchronous relationship between FlexVol volumes, which provides zero RPO data protection. It supports the SnapMirror policy types "mirror-vault", "sync-mirror", and "strict-sync-mirror". You can create a relationship between the source and destination which can be used to transfer APIs to perform SnapMirror "restore" operations.

To create FlexVol or FlexGroup 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. 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. 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.

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 *# 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/"`` *2 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"`` *
 # learn more*
- [doc /snapmirror/relationships](#docs-snapmirror-snapmirror_relationships)

GET /snapmirror/relationships

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

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"
```

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.
transfer.uuid	string	query	False	Filter by transfer.uuid
transfer.bytes_transferred	integer	query	False	Filter by transfer.bytes_transferred
transfer.state	string	query	False	Filter by transfer.state
healthy	boolean	query	False	Filter by healthy
source.path	string	query	False	Filter by source.path
source.svm.name	string	query	False	Filter by source.svm.name
source.svm.uuid	string	query	False	Filter by source.svm.uuid

Name	Type	In	Required	Description
uuid	string	query	False	Filter by uuid
lag_time	string	query	False	Filter by lag_time
destination.path	string	query	False	Filter by destination.path
destination.svm.name	string	query	False	Filter by destination.svm.name
destination.svm.uuid	string	query	False	Filter by destination.svm.uuid
state	string	query	False	Filter by state
unhealthy_reason.parameters	string	query	False	Filter by unhealthy_reason.parameters
unhealthy_reason.code	integer	query	False	Filter by unhealthy_reason.code
unhealthy_reason.message	string	query	False	Filter by unhealthy_reason.message
policy.uuid	string	query	False	Filter by policy.uuid
policy.type	string	query	False	Filter by policy.type
policy.name	string	query	False	Filter by policy.name
exported_snapshot	string	query	False	Filter by exported_snapshot
restore	boolean	query	False	Filter by restore
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	The default is true for GET calls. When set to false, only the number of records is returned.
return_timeout	integer	query	False	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.
order_by	array[string]	query	False	Order results by specified fields and optional [asc

Response

Status: 200, Ok

Name	Type	Description
_links	_links	
num_records	integer	Number of records
records	array[snapmirror_relationship]	

Example response

```
{  
  "_links": {  
    "next": {  
      "href": "/api/resourcelink"  
    },  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "records": [  
    {  
      "destination": {  
        "path": "svm1:volumel",  
        "svm": {  
          "_links": {  
            "self": {  
              "href": "/api/resourcelink"  
            }  
          },  
          "name": "svm1",  
          "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
        }  
      },  
      "exported_snapshot": "string",  
      "lag_time": "PT8H35M42S",  
      "policy": {  
        "_links": {  
          "self": {  
            "href": "/api/resourcelink"  
          }  
        },  
        "name": "MirrorAndVault",  
        "type": "string",  
        "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"  
      },  
      "restore_to_snapshot": "string",  
      "source": {  
        "path": "svm1:volumel",  
        "svm": {  
          "_links": {  
            "self": {  
              "href": "/api/resourcelink"  
            }  
          }  
        },  
      }  
    }  
  ]  
}
```

```

        "name": "svm1",
        "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
    }
},
"state": "string",
"transfer": {
    "_links": {
        "self": {
            "href": "/api/resourcelink"
        }
    },
    "state": "string",
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
},
"unhealthy_reason": [
    {
        "code": "6621444",
        "message": "Failed to complete update operation on one or
more item relationships.",
        "parameters": []
    },
    {
        "code": "6621445",
        "message": "Group Update failed",
        "parameters": []
    }
],
"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	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	

svm

Name	Type	Description
_links	_links	
name	string	The name of the SVM.
uuid	string	The unique identifier of the SVM.

snapmirror_endpoint

Endpoint of a SnapMirror relationship.

Name	Type	Description
path	string	ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: <ul style="list-style-type: none">example: svm1:volume1readCreate: 1
svm	svm	

policy

Basic policy information of the relationship.

Name	Type	Description
_links	_links	
name	string	
type	string	
uuid	string	

transfer

Basic information on the current transfer.

Name	Type	Description
_links	_links	
bytes_transferred	integer	Bytes transferred.
state	string	
uuid	string	

snapmirror_error

SnapMirror error

Name	Type	Description
code	integer	Error code
message	string	Error message
parameters	array[string]	Parameters for the error message

snapmirror_relationship

SnapMirror relationship information

Name	Type	Description
destination	snapmirror_endpoint	Endpoint of a SnapMirror relationship.
exported_snapshot	string	Snapshot copy exported to clients on destination.
healthy	boolean	Is the relationship healthy?
lag_time	string	Time since the exported Snapshot copy was created.

Name	Type	Description
policy	policy	Basic policy information of the 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 field is applicable only for relationships with FlexGroup or FlexVol endpoints and when the PATCH state is "snapmirrored".
quick_resync	boolean	Set to true to reduce resync time by not preserving storage efficiency. This field is applicable only for relationships with FlexVol endpoints and when the PATCH state is "snapmirrored".
recover_after_break	boolean	Set to true to recover from a failed SnapMirror break operation on a FlexGroup relationship. This restores all destination FlexGroup constituents to the latest Snapshot copy, and any writes to the read-write constituents are lost. This field is applicable only for SnapMirror relationships with FlexGroup endpoints and when the PATCH state is "broken_off".
restore	boolean	Set to true to create a relationship for restore. To trigger restore-transfer, use transfers POST on the restore relationship.
restore_to_snapshot	string	Specifies the Snapshot copy to restore to on the destination after a break operation. This field is applicable only for SnapMirror relationships with FlexVol endpoints and when the PATCH state is "broken_off".
source	snapmirror_endpoint	Endpoint of a SnapMirror relationship.

Name	Type	Description
state	string	<p>State of the relationship. To initialize the relationship, PATCH the state to "snapmirrored". To break the relationship, PATCH the state to "broken_off". To resync the broken relationship, PATCH the state to "snapmirrored" for relationships with async policy type or "in_sync" for relationships with sync policy type. To pause the relationship, suspending further transfers, PATCH the state to "paused". To resume transfers for a paused relationship, PATCH the state to "snapmirrored" or "in_sync". The entries "in_sync", "out_of_sync", and "synchronizing" are only applicable to relationships of the sync policy type. 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" fields using GET to determine if the transition is successful.</p>
transfer	transfer	Basic information on the current transfer.
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	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

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.

Creates a snapmirror relationship this api must be executed on the cluster containing the destination endpoint

required properties

- sourcepath - path to the source endpoint of the snapmirror relationship
- destinationpath - path to the destination endpoint of the snapmirror relationship **#** recommended optional properties
- policymame or policyuuid - policy governing the snapmirror relationship **#** default property values if not specified in post, the following default property values are assigned:
 - policymame - *mirrorandvault*
 - restore - *false* **#** related ontap commands
- snapmirror create **# examples** the following examples show how to create flexvol, flexgroup and svm snapmirror relationships note that the source svm name should be the local name of the peer svm
 creating a flexvol snapmirror relationship of type xdp
 ` post "/api/snapmirror/relationships/" {"source": {"path": "test_vserv_src:src_vol_rw"}, "destination": {"path": "test_vserv_dst:dst_vol_rw"} }`
 creating a flexgroup snapmirror relationship of type xdp
 ` post "/api/snapmirror/relationships/" {"source": {"path": "test_vserv_src:source_flexgrp"}, "destination": {"path": "test_vserv_dst: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": "test_vserv_src:src_vol_rw"}, "destination": {"path": "test_vserv_dst:dst_vol_rw"}, "restore": "true"}`
 # learn more
- [doc /snapmirror/relationships](#docs-snapmirror-snapmirror_relationships)

POST /snapmirror/relationships

Creates a SnapMirror relationship. This API must be executed on the cluster containing the destination endpoint.

Required properties

- `source.path` - Path to the source endpoint of the SnapMirror relationship.
- `destination.path` - Path to the destination endpoint of the SnapMirror relationship.

Recommended optional properties

- `policy.name` or `policy.uuid` - Policy governing the SnapMirror relationship.

Default property values

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

- `policy.name` - *MirrorAndVault*
- `restore` - *false*

Related ONTAP commands

- `snapmirror create`

Examples

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

Creating a FlexVol SnapMirror relationship of type XDP.

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "test_vserv_src:src_vol_rw"}, "destination": {"path": "test_vserv_dst:dst_vol_rw"} }'
```

Creating a FlexGroup SnapMirror relationship of type XDP.

```
POST "/api/snapmirror/relationships/" '{"source": {"path": "test_vserv_src:source_flexgrp"}, "destination": {"path": "test_vserv_dst: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": "test_vserv_src:src_vol_rw"}, "destination": {"path": "test_vserv_dst:dst_vol_rw"}, "restore": "true"}'
```

Learn more

- [DOC /snapmirror/relationships](#)

Parameters

Name	Type	In	Required	Description
return_timeout	integer	query	False	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.

Request Body

Name	Type	Description
destination	snapmirror_endpoint	Endpoint of a SnapMirror relationship.

Name	Type	Description
exported_snapshot	string	Snapshot copy exported to clients on destination.
healthy	boolean	Is the relationship healthy?
lag_time	string	Time since the exported Snapshot copy was created.
policy	policy	Basic policy information of the 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 field is applicable only for relationships with FlexGroup or FlexVol endpoints and when the PATCH state is "snapmirrored".
quick_resync	boolean	Set to true to reduce resync time by not preserving storage efficiency. This field is applicable only for relationships with FlexVol endpoints and when the PATCH state is "snapmirrored".
recover_after_break	boolean	Set to true to recover from a failed SnapMirror break operation on a FlexGroup relationship. This restores all destination FlexGroup constituents to the latest Snapshot copy, and any writes to the read-write constituents are lost. This field is applicable only for SnapMirror relationships with FlexGroup endpoints and when the PATCH state is "broken_off".
restore	boolean	Set to true to create a relationship for restore. To trigger restore-transfer, use transfers POST on the restore relationship.

Name	Type	Description
restore_to_snapshot	string	Specifies the Snapshot copy to restore to on the destination after a break operation. This field is applicable only for SnapMirror relationships with FlexVol endpoints and when the PATCH state is "broken_off".
source	snapmirror_endpoint	Endpoint of a SnapMirror relationship.
transfer	transfer	Basic information on the current transfer.
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	

Example request

```

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

```

Response

Status: 202, Accepted

Name	Type	Description
job	job_link	

Example response

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

Error

Status: Default

ONTAP Error Response codes

Error code	Description
13303841	Cross platform relationships are not allowed
13303821	Invalid SnapMirror policy UUID
13303819	Could not retrieve SnapMirror policy information
13303852	destination.path provided does not contain \"\":\"
13303853	Restore relationships are not supported for SVM-DR endpoints

Name	Type	Description
error	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	

svm

Name	Type	Description
_links	_links	
name	string	The name of the SVM.
uuid	string	The unique identifier of the SVM.

snapmirror_endpoint

Endpoint of a SnapMirror relationship.

Name	Type	Description
path	string	ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: <ul style="list-style-type: none">example: svm1:volume1readCreate: 1
svm	svm	

policy

Basic policy information of the relationship.

Name	Type	Description
_links	_links	
name	string	
type	string	
uuid	string	

transfer

Basic information on the current transfer.

Name	Type	Description
_links	_links	
bytes_transferred	integer	Bytes transferred.
state	string	
uuid	string	

`snapmirror_error`

SnapMirror error

Name	Type	Description
code	integer	Error code
message	string	Error message
parameters	array[string]	Parameters for the error message

`snapmirror_relationship`

SnapMirror relationship information

Name	Type	Description
destination	snapmirror_endpoint	Endpoint of a SnapMirror relationship.
exported_snapshot	string	Snapshot copy exported to clients on destination.
healthy	boolean	Is the relationship healthy?
lag_time	string	Time since the exported Snapshot copy was created.
policy	policy	Basic policy information of the relationship.

Name	Type	Description
preserve	boolean	Set to true on resync to preserve Snapshot copies on the destination that are newer than the latest common Snapshot copy. This field is applicable only for relationships with FlexGroup or FlexVol endpoints and when the PATCH state is "snapmirrored".
quick_resync	boolean	Set to true to reduce resync time by not preserving storage efficiency. This field is applicable only for relationships with FlexVol endpoints and when the PATCH state is "snapmirrored".
recover_after_break	boolean	Set to true to recover from a failed SnapMirror break operation on a FlexGroup relationship. This restores all destination FlexGroup constituents to the latest Snapshot copy, and any writes to the read-write constituents are lost. This field is applicable only for SnapMirror relationships with FlexGroup endpoints and when the PATCH state is "broken_off".
restore	boolean	Set to true to create a relationship for restore. To trigger restore-transfer, use transfers POST on the restore relationship.
restore_to_snapshot	string	Specifies the Snapshot copy to restore to on the destination after a break operation. This field is applicable only for SnapMirror relationships with FlexVol endpoints and when the PATCH state is "broken_off".
source	snapmirror_endpoint	Endpoint of a SnapMirror relationship.
transfer	transfer	Basic information on the current transfer.

Name	Type	Description
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	

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

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.

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 # 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" `
 # learn more

• [doc /snapmirror/relationships](#docs-snapmirror-snapmirror_relationships)

DELETE /snapmirror/relationships/{uuid}

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.

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"
```

Learn more

- [DOC /snapmirror/relationships](#)

Parameters

Name	Type	In	Required	Description
uuid	string	path	True	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.

Name	Type	In	Required	Description
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.
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.
return_timeout	integer	query	False	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.

Response

Status: 202, Accepted

Name	Type	Description
job	job_link	

Example response

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

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

Name	Type	Description
error	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	

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

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.

Retrieves a snapmirror relationship

related ontap commands

- snapmirror show
- snapmirror list-destinations **# example**


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


 # learn more * [doc /snapmirror/relationships](#docs-snapmirror-snapmirror_relationships)

GET /snapmirror/relationships/{uuid}

Retrieves a SnapMirror relationship.

Related ONTAP commands

- snapmirror show
- snapmirror list-destinations

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	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
destination	snapmirror_endpoint	Endpoint of a SnapMirror relationship.
exported_snapshot	string	Snapshot copy exported to clients on destination.
healthy	boolean	Is the relationship healthy?
lag_time	string	Time since the exported Snapshot copy was created.
policy	policy	Basic policy information of the 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 field is applicable only for relationships with FlexGroup or FlexVol endpoints and when the PATCH state is "snapmirrored".
quick_resync	boolean	Set to true to reduce resync time by not preserving storage efficiency. This field is applicable only for relationships with FlexVol endpoints and when the PATCH state is "snapmirrored".
recover_after_break	boolean	Set to true to recover from a failed SnapMirror break operation on a FlexGroup relationship. This restores all destination FlexGroup constituents to the latest Snapshot copy, and any writes to the read-write constituents are lost. This field is applicable only for SnapMirror relationships with FlexGroup endpoints and when the PATCH state is "broken_off".
restore	boolean	Set to true to create a relationship for restore. To trigger restore-transfer, use transfers POST on the restore relationship.

Name	Type	Description
restore_to_snapshot	string	Specifies the Snapshot copy to restore to on the destination after a break operation. This field is applicable only for SnapMirror relationships with FlexVol endpoints and when the PATCH state is "broken_off".
source	snapmirror_endpoint	Endpoint of a SnapMirror relationship.
state	string	State of the relationship. To initialize the relationship, PATCH the state to "snapmirrored". To break the relationship, PATCH the state to "broken_off". To resync the broken relationship, PATCH the state to "snapmirrored" for relationships with async policy type or "in_sync" for relationships with sync policy type. To pause the relationship, suspending further transfers, PATCH the state to "paused". To resume transfers for a paused relationship, PATCH the state to "snapmirrored" or "in_sync". The entries "in_sync", "out_of_sync", and "synchronizing" are only applicable to relationships of the sync policy type. 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" fields using GET to determine if the transition is successful.
transfer	transfer	Basic information on the current transfer.
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	

Example response

```
{  
  "destination": {  
    "path": "svm1:volumel",  
    "svm": {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "name": "svm1",  
      "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
    }  
  },  
  "exported_snapshot": "string",  
  "lag_time": "PT8H35M42S",  
  "policy": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "name": "MirrorAndVault",  
    "type": "string",  
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"  
  },  
  "restore_to_snapshot": "string",  
  "source": {  
    "path": "svm1:volumel",  
    "svm": {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "name": "svm1",  
      "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
    }  
  },  
  "state": "string",  
  "transfer": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    }  
  }  
}
```

```

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

svm

Name	Type	Description
_links	_links	
name	string	The name of the SVM.
uuid	string	The unique identifier of the SVM.

snapmirror_endpoint

Endpoint of a SnapMirror relationship.

Name	Type	Description
path	string	ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: <ul style="list-style-type: none">example: svm1:volume1readCreate: 1
svm	svm	

policy

Basic policy information of the relationship.

Name	Type	Description
_links	_links	
name	string	
type	string	
uuid	string	

transfer

Basic information on the current transfer.

Name	Type	Description
_links	_links	
bytes_transferred	integer	Bytes transferred.
state	string	
uuid	string	

`snapmirror_error`

SnapMirror error

Name	Type	Description
code	integer	Error code
message	string	Error message
parameters	array[string]	Parameters for the error message

`error_arguments`

Name	Type	Description
code	string	Argument code
message	string	Message argument

`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.

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

related ontap commands

- snapmirror modify
- snapmirror initialize
- snapmirror resync
- snapmirror break
- snapmirror quiesce
- snapmirror resume **# examples the following examples show how to perform the snapmirror "resync", "initialize", "resume", "quiesce", and "break" operations**
 performing a snapmirror "resync"
 patch "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" {"state":"snapmirrored"}
 performing a snapmirror "initialize"
 patch "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" {"state":"snapmirrored"}
 performing a snapmirror "resume"
 patch "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" {"state":"snapmirrored"}
 performing a snapmirror "quiesce"
 patch "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" {"state":"paused"}
 performing a snapmirror "break"
 patch "/api/snapmirror/relationships/98bb2608-fc60-11e8-aa13-005056a707ff/" {"state":"broken_off"}
 updating an associated snapmirror policy
 patch "/api/snapmirror/relationships/9e922e65-1818-11e9-8b22-005056bbe73/" {"policy": { "name" : "mirrorandvaultdiscardnetwork" }}
 # learn more
- [doc /snapmirror/relationships](#docs-snapmirror-snapmirror_relationships)

PATCH /snapmirror/relationships/{uuid}

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.

Related ONTAP commands

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

- snapmirror resume

Examples

The following examples show how to perform the SnapMirror "resync", "initialize", "resume", "quiesce", and "break" operations.

Performing a SnapMirror "resync"

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

Performing a SnapMirror "initialize"

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

Performing a SnapMirror "resume"

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

Performing a SnapMirror "quiesce"

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

Performing a SnapMirror "break"

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

Updating an associated SnapMirror policy

```
PATCH "/api/snapmirror/relationships/9e922e65-1818-11e9-8b22-005056bbe73/" '{"policy": { "name" : "MirrorAndVaultDiscardNetwork" }}'
```

Learn more

- [DOC /snapmirror/relationships](#)

Parameters

Name	Type	In	Required	Description
uuid	string	path	True	Relationship UUID

Request Body

Name	Type	Description
destination	snapmirror_endpoint	Endpoint of a SnapMirror relationship.
exported_snapshot	string	Snapshot copy exported to clients on destination.
healthy	boolean	Is the relationship healthy?
lag_time	string	Time since the exported Snapshot copy was created.
policy	policy	Basic policy information of the 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 field is applicable only for relationships with FlexGroup or FlexVol endpoints and when the PATCH state is "snapmirrored".
quick_resync	boolean	Set to true to reduce resync time by not preserving storage efficiency. This field is applicable only for relationships with FlexVol endpoints and when the PATCH state is "snapmirrored".

Name	Type	Description
recover_after_break	boolean	Set to true to recover from a failed SnapMirror break operation on a FlexGroup relationship. This restores all destination FlexGroup constituents to the latest Snapshot copy, and any writes to the read-write constituents are lost. This field is applicable only for SnapMirror relationships with FlexGroup endpoints and when the PATCH state is "broken_off".
restore_to_snapshot	string	Specifies the Snapshot copy to restore to on the destination after a break operation. This field is applicable only for SnapMirror relationships with FlexVol endpoints and when the PATCH state is "broken_off".
source	snapmirror_endpoint	Endpoint of a SnapMirror relationship.
state	string	State of the relationship. To initialize the relationship, PATCH the state to "snapmirrored". To break the relationship, PATCH the state to "broken_off". To resync the broken relationship, PATCH the state to "snapmirrored" for relationships with async policy type or "in_sync" for relationships with sync policy type. To pause the relationship, suspending further transfers, PATCH the state to "paused". To resume transfers for a paused relationship, PATCH the state to "snapmirrored" or "in_sync". The entries "in_sync", "out_of_sync", and "synchronizing" are only applicable to relationships of the sync policy type. 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" fields using GET to determine if the transition is successful.

Name	Type	Description
transfer	transfer	Basic information on the current transfer.
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	

Example request

```
{  
  "destination": {  
    "svm": {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "name": "svm1",  
      "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
    }  
  },  
  "exported_snapshot": "string",  
  "lag_time": "PT8H35M42S",  
  "policy": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    },  
    "name": "MirrorAndVault",  
    "type": "string",  
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"  
  },  
  "restore": null,  
  "restore_to_snapshot": "string",  
  "source": {  
    "svm": {  
      "_links": {  
        "self": {  
          "href": "/api/resourcelink"  
        }  
      },  
      "name": "svm1",  
      "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"  
    }  
  },  
  "state": "string",  
  "transfer": {  
    "_links": {  
      "self": {  
        "href": "/api/resourcelink"  
      }  
    }  
  },  
},  
}
```

```

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

```

Response

Status: 202, Accepted

Name	Type	Description
job	job_link	

Example response

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

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
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

Error code	Description
13303810	Transition to snapmirrored state failed
13303811	Transition from paused state failed
13303820	SnapMirror policy was successfully updated, state transition failed

Name	Type	Description
error	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	

svm

Name	Type	Description
_links	_links	
name	string	The name of the SVM.
uuid	string	The unique identifier of the SVM.

snapmirror_endpoint

Endpoint of a SnapMirror relationship.

Name	Type	Description
svm	svm	

policy

Basic policy information of the relationship.

Name	Type	Description
_links	_links	
name	string	
type	string	
uuid	string	

transfer

Basic information on the current transfer.

Name	Type	Description
_links	_links	

Name	Type	Description
bytes_transferred	integer	Bytes transferred.
state	string	
uuid	string	

`snapmirror_error`

SnapMirror error

Name	Type	Description
code	integer	Error code
message	string	Error message
parameters	array[string]	Parameters for the error message

`snapmirror_relationship`

SnapMirror relationship information

Name	Type	Description
destination	snapmirror_endpoint	Endpoint of a SnapMirror relationship.
exported_snapshot	string	Snapshot copy exported to clients on destination.
healthy	boolean	Is the relationship healthy?
lag_time	string	Time since the exported Snapshot copy was created.
policy	policy	Basic policy information of the 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 field is applicable only for relationships with FlexGroup or FlexVol endpoints and when the PATCH state is "snapmirrored".

Name	Type	Description
quick_resync	boolean	Set to true to reduce resync time by not preserving storage efficiency. This field is applicable only for relationships with FlexVol endpoints and when the PATCH state is "snapmirrored".
recover_after_break	boolean	Set to true to recover from a failed SnapMirror break operation on a FlexGroup relationship. This restores all destination FlexGroup constituents to the latest Snapshot copy, and any writes to the read-write constituents are lost. This field is applicable only for SnapMirror relationships with FlexGroup endpoints and when the PATCH state is "broken_off".
restore_to_snapshot	string	Specifies the Snapshot copy to restore to on the destination after a break operation. This field is applicable only for SnapMirror relationships with FlexVol endpoints and when the PATCH state is "broken_off".
source	snapmirror_endpoint	Endpoint of a SnapMirror relationship.

Name	Type	Description
state	string	<p>State of the relationship. To initialize the relationship, PATCH the state to "snapmirrored". To break the relationship, PATCH the state to "broken_off". To resync the broken relationship, PATCH the state to "snapmirrored" for relationships with async policy type or "in_sync" for relationships with sync policy type. To pause the relationship, suspending further transfers, PATCH the state to "paused". To resume transfers for a paused relationship, PATCH the state to "snapmirrored" or "in_sync". The entries "in_sync", "out_of_sync", and "synchronizing" are only applicable to relationships of the sync policy type. 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" fields using GET to determine if the transition is successful.</p>
transfer	transfer	Basic information on the current transfer.
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	

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

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 © 2026 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—with 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.