



# **Manage SnapMirror relationships**

## **ONTAP 9.8 REST API reference**

NetApp  
May 08, 2024

This PDF was generated from [https://docs.netapp.com/us-en/ontap-restapi-98/ontap/snapmirror\\_relationships\\_endpoint\\_overview.html](https://docs.netapp.com/us-en/ontap-restapi-98/ontap/snapmirror_relationships_endpoint_overview.html) on May 08, 2024. Always check docs.netapp.com for the latest.

# Table of Contents

- Manage SnapMirror relationships ..... 1
  - SnapMirror relationships endpoint overview ..... 1
  - Retrieve information for SnapMirror relationships ..... 1
  - Create a SnapMirror relationship ..... 25
  - Delete a SnapMirror relationship ..... 56
  - Retrieve a SnapMirror relationship ..... 62
  - Update a SnapMirror relationship ..... 81

# Manage SnapMirror relationships

## SnapMirror relationships endpoint overview

### Overview

This API manages asynchronous extended data protection (XDP) relationships for FlexVols, FlexGroups, 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 Consistency Group endpoints which provides zero RTO data protection. To create an asynchronous extended data protection relationship with FlexVol volumes, FlexGroup volumes, 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 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 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 when the "create\_destination" property is not specified. When the "create\_destination" property is specified then 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 the source SVM has the SVM peering permission for the destination cluster. 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`

### 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.state             | string  | query | False    | Filter by transfer.state   |
| transfer.bytes_transferred | integer | query | False    | Filter by transfer.bytes_transferred   |
| transfer.uuid              | string  | query | False    | Filter by transfer.uuid  |
| destination.cluster.name   | string  | query | False    | Filter by destination.cluster.name<br><ul style="list-style-type: none"><li>• Introduced in: 9.7</li></ul> |
| destination.cluster.uuid   | string  | query | False    | Filter by destination.cluster.uuid<br><ul style="list-style-type: none"><li>• Introduced in: 9.7</li></ul> |

| Name  | Type   | In    | Required | Description   |
|---|--------|-------|----------|---|
| destination.consistency_group_volumes.uuid      | string | query | False    | Filter by destination.consistency_group_volumes.uuid<br><br>• Introduced in: 9.8      |
| destination.consistency_group_volumes.name      | string | query | False    | Filter by destination.consistency_group_volumes.name<br><br>• Introduced in: 9.8      |
| destination.path                                | string | query | False    | Filter by destination.path  |
| destination.svm.uuid                            | string | query | False    | Filter by destination.svm.uuid  |
| destination.svm.name                            | string | query | False    | Filter by destination.svm.name  |
| uuid  | string | query | False    | Filter by uuid  |
| consistency_group_failover.error.code           | string | query | False    | Filter by consistency_group_failover.error.code<br><br>• Introduced in: 9.8           |
| consistency_group_failover.error.arguments.code | string | query | False    | Filter by consistency_group_failover.error.arguments.code<br><br>• Introduced in: 9.8 |

| Name   | Type   | In    | Required | Description  |
|--|--------|-------|----------|--|
| consistency_group_failover.error.arguments.message | string | query | False    | Filter by consistency_group_failover.error.arguments.message<br><br>• Introduced in: 9.8 |
| consistency_group_failover.error.message           | string | query | False    | Filter by consistency_group_failover.error.message<br><br>• Introduced in: 9.8           |
| consistency_group_failover.error.target            | string | query | False    | Filter by consistency_group_failover.error.target<br><br>• Introduced in: 9.8            |
| consistency_group_failover.status.message          | string | query | False    | Filter by consistency_group_failover.status.message<br><br>• Introduced in: 9.8          |
| consistency_group_failover.status.code             | string | query | False    | Filter by consistency_group_failover.status.code<br><br>• Introduced in: 9.8             |
| source.cluster.name                                | string | query | False    | Filter by source.cluster.name<br><br>• Introduced in: 9.7                                |
| source.cluster.uuid                                | string | query | False    | Filter by source.cluster.uuid<br><br>• Introduced in: 9.7                                |

| Name                                  | Type    | In    | Required | Description   |
|---------------------------------------|---------|-------|----------|---|
| source.consistency_group_volumes.uuid | string  | query | False    | Filter by source.consistency_group_volumes.uuid<br><br>• Introduced in: 9.8 |
| source.consistency_group_volumes.name | string  | query | False    | Filter by source.consistency_group_volumes.name<br><br>• Introduced in: 9.8 |
| 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   |
| lag_time                              | string  | query | False    | Filter by lag_time  |
| policy.type                           | string  | query | False    | Filter by policy.type   |
| policy.uuid                           | string  | query | False    | Filter by policy.uuid   |
| policy.name                           | string  | query | False    | Filter by policy.name   |
| state                                 | string  | query | False    | Filter by state   |
| unhealthy_reason.parameters           | string  | query | False    | Filter by unhealthy_reason.parameters                                       |
| unhealthy_reason.message              | string  | query | False    | Filter by unhealthy_reason.message  |
| unhealthy_reason.code                 | integer | query | False    | Filter by unhealthy_reason.code   |
| restore                               | boolean | query | False    | Filter by restore   |

| Name              | Type          | In    | Required | Description  |
|-------------------|---------------|-------|----------|--|
| exported_snapshot | string        | query | False    | Filter by exported_snapshot  |
| healthy           | boolean       | query | False    | Filter by healthy  |
| 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_relationship</a> ] |                   |

## Example response

```
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "records": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "consistency_group_failover": {
      "error": {
        "arguments": {
          "code": "string",
          "message": "string"
        },
        "code": "4",
        "message": "entry doesn't exist",
        "target": "uuid"
      }
    },
    "create_destination": {
      "storage_service": {
        "name": "extreme"
      },
      "tiering": {
        "policy": "all"
      }
    },
    "destination": {
      "cluster": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "cluster1",
        "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
      }
    }
  },
}
```

```

"consistency_group_volumes": {
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "name": "volume1",
  "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
},
"ipspace": "Default",
"path": "svm1:volume1",
"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": "Asynchronous",
  "type": "async",
  "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
},
"source": {
  "cluster": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "cluster1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "consistency_group_volumes": {
    "_links": {
      "self": {

```

```

        "href": "/api/resourcelink"
      },
    },
    "name": "volume1",
    "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
  },
  "ipspace": "Default",
  "path": "svm1:volume1",
  "svm": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  },
  "state": "snapmirrored",
  "transfer": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "state": "aborted",
    "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 | <a href="#">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 |
|------|----------------------|-------------|
| next | <a href="#">href</a> |             |
| self | <a href="#">href</a> |             |

\_links

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

error\_arguments

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

error

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

status

| Name | Type   | Description |
|------|--------|-------------|
| code | string | Status code |

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

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> </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 &amp;dash; 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 &amp;dash; This policy allows tiering of both destination endpoint Snapshot copies and the active file system user data to the cloud store none &amp;dash; Destination endpoint volume blocks will not be tiered to the cloud store. snapshot_only &amp;dash; 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, or Consistency Group. 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 <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, or a Consistency Group 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.&lt;/destination-svm-name:&gt;&lt;/destination-svm-name:&gt;</dp-volume-name>

| Name            | Type                            | Description  |
|-----------------|---------------------------------|--|
| 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. |
| 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  |
|--------|------------------------|--|
| _links | <a href="#">_links</a> |  |
| name   | string                 | The name of the volume.  |
| uuid   | string                 | <p>Unique identifier for 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> </ul> |

#### svm

| Name   | Type                   | Description                       |
|--------|------------------------|-----------------------------------|
| _links | <a href="#">_links</a> |                                   |
| name   | string                 | The name of the SVM.              |
| uuid   | string                 | The unique identifier of the SVM. |

#### 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, and Consistency Group 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.   |
| ipspace                   | string   | Optional property to specify the IPspace of the SVM.  |
| path                      | string   | ONTAP FlexVol/FlexGroup - svm1:volume1<br>ONTAP SVM - svm1:<br>ONTAP Consistency Group - svm1:/cg/cg_name <ul style="list-style-type: none"> <li>• example: svm1:volume1</li> <li>• Introduced in: 9.6</li> </ul> |
| svm                       | <a href="#">svm</a>                                |   |

## policy

Basic policy information of the relationship.

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

## transfer

Basic information on the current transfer.

| Name              | Type                   | Description        |
|-------------------|------------------------|--------------------|
| _links            | <a href="#">_links</a> |                    |
| bytes_transferred | integer                | Bytes transferred. |

| Name  | Type   | Description |
|-------|--------|-------------|
| 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. 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>                                |   |
| 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  |
|--------------------|---|--|
| 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, or Consistency Group. 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 &amp;lt;destination-SVM-name:&amp;gt;, and for Consistency Group, the "destination.path" must be specified as &amp;lt;destination-SVM-name:/cg/consistency-group-name&amp;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, or a Consistency Group destination endpoint, the properties in this object can be specified either from the source or the destination</p> |

| Name              | Type                                | Description   |
|-------------------|-------------------------------------|---|
| destination       | <a href="#">snapmirror_endpoint</a> | 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, and Consistency Group 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. |
| 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            | <a href="#">policy</a>              | 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 property is applicable only for relationships with FlexVol volume or FlexGroup volume endpoints and when the PATCH state is being changed to "snapmirrored".   |

| Name                | Type    | Description   |
|---------------------|---------|---|
| 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 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.  |
| 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_endpoint</a> | 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, and Consistency Group 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  |
|-------|--------|--|
| 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", and "synchronizing" 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   |
|------------------|---|---|
| transfer         | <a href="#">transfer</a>                  | Basic information on the current transfer.  |
| 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                                    |   |

## 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 Consistency Group endpoints. For SVM endpoint, provisioning of 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 the source SVM has SVM peering permission 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.
- `destination.consistency_group_volumes` - List of FlexVol volumes of type "RW" that are constituents of a Consistency Group.
- `destination.consistency_group_volumes` - List of FlexVol volumes of type "DP" that are constituents of a 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*
- `source.ipspace` - *Default*
- `destination.ipspace` - *Default*

## Related ONTAP commands

- `snapmirror create`
- `snapmirror protect`

## Examples

The following examples show how to create FlexVol, FlexGroup, SVM and Consistency Group 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": "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": { "enable": "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": { "enable": "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": { "enable": "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": { "enable": "true" }}'
```

Create a SnapMirror relationship with Consistency Group endpoint.

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

Provision the destination Consistency Group endpoint on a Fabricpool with a tiering policy, create a SnapMirror 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": "src_vol_1,
src_vol_2"}, "destination": { "path": "dst_svm:/cg/cg_dst_vol",
"consistency_group_volumes": "dst_vol_1, dst_vol_2"},
"create_destination": { "enable": "true", "tiering": { "supported": "true"
} }, "policy": "AutomatedFailOver", "state": "in_sync" }'
```

Provision the destination Consistency Group endpoint with storage service, create a SnapMirror 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": "src_vol_1,
src_vol_2"}, "destination": { "path": "dst_svm:/cg/cg_dst_vol",
"consistency_group_volumes": "dst_vol_1, dst_vol_2"},
"create_destination": { "enable": "true", "storage_service": { "enabled":
"true", "name": "extreme", "enforce_performance": "true" } }, "policy":
"AutomatedFailOver", "state": "in_sync" }'
```

## 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 |
|--------|------------------------|-------------|
| _links | <a href="#">_links</a> |             |

| 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> |

| 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, or Consistency Group. 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 &amp;lt;destination-SVM-name:&amp;gt;, and for Consistency Group, the "destination.path" must be specified as &amp;lt;destination-SVM-name:/cg/consistency-group-name&amp;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, or a Consistency Group 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</p> |

| Name              | Type                                | Description   |
|-------------------|-------------------------------------|---|
| destination       | <a href="#">snapmirror_endpoint</a> | 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, and Consistency Group 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. |
| 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            | <a href="#">policy</a>              | 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 property is applicable only for relationships with FlexVol volume or FlexGroup volume endpoints and when the PATCH state is being changed to "snapmirrored".   |

| Name                | Type    | Description   |
|---------------------|---------|---|
| 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 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.  |
| 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_endpoint</a> | 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, and Consistency Group 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  |
|-------|--------|--|
| 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", and "synchronizing" 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   |
|------------------|---|---|
| transfer         | <a href="#">transfer</a>                  | Basic information on the current transfer.  |
| 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                                    |   |

## Example request

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "consistency_group_failover": {
    "error": {
      "arguments": {
        "code": "string",
        "message": "string"
      },
      "code": "4",
      "message": "entry doesn't exist",
      "target": "uuid"
    }
  },
  "create_destination": {
    "storage_service": {
      "name": "extreme"
    },
    "tiering": {
      "policy": "all"
    }
  },
  "destination": {
    "cluster": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "cluster1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "consistency_group_volumes": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "volume1",
      "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
    }
  },
}
```

```

"ipospace": "Default",
"path": "svm1:volume1",
"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": "Asynchronous",
  "type": "async",
  "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
},
"source": {
  "cluster": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "cluster1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "consistency_group_volumes": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "volume1",
    "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
  },
  "ipospace": "Default",
  "path": "svm1:volume1",
  "svm": {

```

```

    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  },
  "state": "snapmirrored",
  "transfer": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "state": "aborted",
    "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  | <a href="#">job_link</a> |             |

## Example response

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

## Error

Status: Default

### ONTAP Error Response Codes

| Error Code | Description   |
|------------|---|
| 6620374    | Internal error. Failed to get SVM information.  |
| 6620478    | Internal error. Failed to check SnapMirror capability.                                      |
| 13303819   | Could not retrieve SnapMirror policy information.   |
| 13303821   | Invalid SnapMirror policy UUID.   |
| 13303841   | This operation is not supported for SnapMirror relationships between these endpoints.       |
| 13303852   | destination.path provided does not contain "\"\."   |
| 13303853   | Restore relationships are not supported for SVM-DR endpoints.                               |
| 13303868   | Create of destination endpoint and SnapMirror relationship failed.                          |
| 13303869   | Creating a destination endpoint is not supported for restore relationships.                 |
| 13303870   | A tiering policy cannot be specified if tiering is not being set to supported.              |
| 13303871   | Storage service properties cannot be specified if the storage service is not being enabled. |
| 13303872   | Specified property requires a later effective cluster version.                              |

| Error Code | Description   |
|------------|---|
| 13303873   | Specifying a state when creating a relationship is only supported when creating a destination endpoint.   |
| 13303874   | Specified state is not supported when creating this relationship.   |
| 13303875   | Destination aggregates do not have sufficient space for hosting copies of source volumes.   |
| 13303876   | Destination cluster does not have composite aggregates.   |
| 13303877   | Source or destination cluster must be specified.  |
| 13303878   | The specified fields do not match.  |
| 13303879   | Source cluster name or UUID is needed to provision a destination SVM on the local cluster.  |
| 13303880   | Source cluster must be remote for provisioning a destination SVM on the local cluster.  |
| 13303881   | Network validation failed.  |
| 13303882   | SVM validation failed.  |
| 13303883   | Encryption is not enabled on the destination cluster.   |
| 13303887   | Synchronous SnapMirror relationships between FlexGroup volumes are not supported.   |
| 13303888   | Synchronous SnapMirror relationships require an effective cluster version of 9.5 or later on both the source and destination clusters.  |
| 13303889   | Asynchronous SnapMirror relationships between FlexGroup volumes require an effective cluster version of 9.5 or later on both the source and destination clusters.                                   |
| 13303890   | Asynchronous SnapMirror relationships between FlexVol volumes require an effective cluster version of 9.3, 9.5, or later on both the source and destination clusters.                               |
| 13303891   | Creating a destination endpoint with storage service requires an effective cluster version of 9.7 or later.   |
| 13303892   | Fetching remote information from the destination cluster failed.  |
| 13303893   | Updating job description failed.  |
| 13303894   | Destination volume name is invalid. It must contain the source volume name and have a suffix when creating a destination endpoint on a cluster with an effective cluster version of 9.6 or earlier. |
| 13303895   | Operation on the remote destination cluster is not supported.   |

| Error Code | Description  |
|------------|--|
| 13303916   | FlexGroup volumes are not supported on SnapLock aggregates.  |
| 13303918   | No suitable destination aggregate type is available.   |
| 13303919   | Only FabricPool enabled aggregates are available on the destination.   |
| 13303920   | Only SnapLock aggregates are available on the destination. FlexGroup volumes are not supported on SnapLock aggregates.   |
| 13303921   | Unable to retrieve the SnapMirror capabilities of the destination cluster.   |
| 13303922   | Specified source SVM is not a data SVM.  |
| 13303923   | Specified destination SVM is not a data SVM.   |
| 13303924   | Source SVM has an invalid Snapshot copy policy.  |
| 13303925   | SnapMirror validation has failed.  |
| 13303930   | The specified tiering policy is not supported for destination volumes of Synchronous relationships.  |
| 13303938   | Fetching information from the local cluster failed.  |
| 13303939   | Could not create an SVM peer relationship.   |
| 13303944   | An SVM-DR relationship is not supported because the source SVM has CIFS configured and the associated SnapMirror policy has either the identity_preservation property not set or set to exclude_network_and_protocol_config. |
| 13303966   | Consistency Group relationships require a policy of type "sync" with a sync_type of "automated_failover".  |
| 13303967   | Consistency Group volume is not a FlexVol volume.  |
| 13303968   | Unsupported volume type for the Consistency Group.   |
| 13303969   | SnapMirror relationships between SVM endpoints and object store endpoints are not supported.   |
| 13303970   | Unsupported policy type for the Consistency Group.   |
| 13303971   | SnapMirror relationships between Consistency Group endpoints and object store endpoints are not supported.   |
| 13303976   | Source or destination SVM is already part of an SVM-DR relation.   |
| 13303977   | Destination Consistency Group volume UUIDs are not expected while provisioning the destination volumes.  |
| 13303978   | Number of Consistency Group volume names and UUIDs does not match.   |

| Error Code | Description  |
|------------|--|
| 13303979   | Number of Consistency Group volumes exceeds the allowed limit.   |
| 13303980   | Number of source and destination Consistency Group volumes do not match.   |
| 13303981   | ISCSI or FCP protocol is not configured.   |
| 13303982   | SAN data interface is not configured on the SVM.   |
| 13304021   | No suitable storage can be found meeting the specified requirements. No FabricPool enabled aggregates are available on the destination.                                    |
| 13304022   | No suitable storage can be found meeting the specified requirements. No non-root, non-taken-over, non-SnapLock, non-composite aggregates are available on the destination. |

| 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 | <a href="#">href</a> |             |

error\_arguments

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

error

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

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>  |             |
| status | <a href="#">status</a> |             |

### 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></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 &amp;dash; 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 &amp;dash; This policy allows tiering of both destination endpoint Snapshot copies and the active file system user data to the cloud store none &amp;dash; Destination endpoint volume blocks will not be tiered to the cloud store. snapshot_only &amp;dash; 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, or Consistency Group. 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 <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, or a Consistency Group 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.&lt;/destination-svm-name:&gt;&lt;/destination-svm-name:&gt;</dp-volume-name>

| Name            | Type                            | Description  |
|-----------------|---------------------------------|--|
| 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. |
| 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  |
|--------|------------------------|--|
| _links | <a href="#">_links</a> |  |
| name   | string                 | The name of the volume.  |
| uuid   | string                 | <p>Unique identifier for 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> </ul> |

#### svm

| Name   | Type                   | Description                       |
|--------|------------------------|-----------------------------------|
| _links | <a href="#">_links</a> |                                   |
| name   | string                 | The name of the SVM.              |
| uuid   | string                 | The unique identifier of the SVM. |

#### 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, and Consistency Group 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.   |
| ipspace                   | string   | Optional property to specify the IPspace of the SVM.  |
| path                      | string   | ONTAP FlexVol/FlexGroup - svm1:volume1<br>ONTAP SVM - svm1:<br>ONTAP Consistency Group - svm1:/cg/cg_name <ul style="list-style-type: none"> <li>• example: svm1:volume1</li> <li>• Introduced in: 9.6</li> </ul> |
| svm                       | <a href="#">svm</a>                                |   |

## policy

Basic policy information of the relationship.

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

## transfer

Basic information on the current transfer.

| Name              | Type                   | Description        |
|-------------------|------------------------|--------------------|
| _links            | <a href="#">_links</a> |                    |
| bytes_transferred | integer                | Bytes transferred. |

| Name  | Type   | Description |
|-------|--------|-------------|
| 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. 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>                                |   |
| 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  |
|--------------------|---|--|
| 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, or Consistency Group. 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 &amp;lt;destination-SVM-name:&amp;gt;, and for Consistency Group, the "destination.path" must be specified as &amp;lt;destination-SVM-name:/cg/consistency-group-name&amp;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, or a Consistency Group destination endpoint, the properties in this object can be specified either from the source or the destination</p> |

| Name              | Type                                | Description   |
|-------------------|-------------------------------------|---|
| destination       | <a href="#">snapmirror_endpoint</a> | 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, and Consistency Group 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. |
| 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            | <a href="#">policy</a>              | 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 property is applicable only for relationships with FlexVol volume or FlexGroup volume endpoints and when the PATCH state is being changed to "snapmirrored".   |

| Name                | Type    | Description   |
|---------------------|---------|---|
| 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 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.  |
| 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_endpoint</a> | 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, and Consistency Group 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  |
|-------|--------|--|
| 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", and "synchronizing" 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   |
|------------------|---|---|
| transfer         | <a href="#">transfer</a>                  | Basic information on the current transfer.  |
| 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                                    |   |

job\_link

| Name   | Type                   | Description   |
|--------|------------------------|---|
| _links | <a href="#">_links</a> |   |
| uuid   | string                 | The UUID of the asynchronous job that is triggered by a POST, PATCH, or DELETE operation. |

## 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 "failback" query parameters are only applicable for SVM-DR SnapMirror 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"
```

## Learn more

- [DOC /snapmirror/relationships](#)

## Parameters

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

| Name             | Type    | In    | Required | Description  |
|------------------|---------|-------|----------|--|
| 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.           |
| 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  |
|----------------|---------|-------|----------|--|
| 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: 202, Accepted

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

## 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 |
| 13303865   | Deleting the specified SnapMirror policy is not supported.             |

| 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 | <a href="#">href</a> |             |

job\_link

| Name   | Type                   | Description   |
|--------|------------------------|---|
| _links | <a href="#">_links</a> |   |
| 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[ <a href="#">error_arguments</a> ] | 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`

## 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 |
|--------|------------------------|-------------|
| _links | <a href="#">_links</a> |             |

| 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> |

| 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, or Consistency Group. 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 &amp;lt;destination-SVM-name:&amp;gt;, and for Consistency Group, the "destination.path" must be specified as &amp;lt;destination-SVM-name:/cg/consistency-group-name&amp;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, or a Consistency Group 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</p> |

| Name              | Type                                | Description   |
|-------------------|-------------------------------------|---|
| destination       | <a href="#">snapmirror_endpoint</a> | 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, and Consistency Group 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. |
| 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            | <a href="#">policy</a>              | 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 property is applicable only for relationships with FlexVol volume or FlexGroup volume endpoints and when the PATCH state is being changed to "snapmirrored".   |

| Name                | Type    | Description   |
|---------------------|---------|---|
| 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 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.  |
| 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_endpoint</a> | 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, and Consistency Group 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  |
|-------|--------|--|
| 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", and "synchronizing" 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   |
|------------------|---|---|
| transfer         | <a href="#">transfer</a>                  | Basic information on the current transfer.  |
| 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                                    |   |

## Example response

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "consistency_group_failover": {
    "error": {
      "arguments": {
        "code": "string",
        "message": "string"
      },
      "code": "4",
      "message": "entry doesn't exist",
      "target": "uuid"
    }
  },
  "create_destination": {
    "storage_service": {
      "name": "extreme"
    },
    "tiering": {
      "policy": "all"
    }
  },
  "destination": {
    "cluster": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "cluster1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "consistency_group_volumes": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "volume1",
      "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
    }
  },
}
```

```

"ipospace": "Default",
"path": "svm1:volume1",
"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": "Asynchronous",
  "type": "async",
  "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
},
"source": {
  "cluster": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "cluster1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "consistency_group_volumes": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "volume1",
    "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
  },
  "ipospace": "Default",
  "path": "svm1:volume1",
  "svm": {

```

```

    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  },
  "state": "snapmirrored",
  "transfer": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "state": "aborted",
    "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 | <a href="#">href</a> |             |

error\_arguments

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

error

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

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>  |             |
| status | <a href="#">status</a> |             |

### 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></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 &amp;dash; 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 &amp;dash; This policy allows tiering of both destination endpoint Snapshot copies and the active file system user data to the cloud store none &amp;dash; Destination endpoint volume blocks will not be tiered to the cloud store. snapshot_only &amp;dash; 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, or Consistency Group. 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 <dp-volume-name>, for SVM data protection, the "destination.path" must be specified as <destination-svm-name>, and for Consistency Group, the "destination.path" must be specified as <destination-svm-name> 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, or a Consistency Group 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.</destination-svm-name><destination-svm-name></dp-volume-name>

| Name            | Type                            | Description  |
|-----------------|---------------------------------|--|
| 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. |
| 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  |
|--------|------------------------|--|
| _links | <a href="#">_links</a> |  |
| name   | string                 | The name of the volume.  |
| uuid   | string                 | <p>Unique identifier for 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> </ul> |

#### svm

| Name   | Type                   | Description                       |
|--------|------------------------|-----------------------------------|
| _links | <a href="#">_links</a> |                                   |
| name   | string                 | The name of the SVM.              |
| uuid   | string                 | The unique identifier of the SVM. |

#### 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, and Consistency Group 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.   |
| ipspace                   | string   | Optional property to specify the IPspace of the SVM.  |
| path                      | string   | ONTAP FlexVol/FlexGroup - svm1:volume1<br>ONTAP SVM - svm1:<br>ONTAP Consistency Group - svm1:/cg/cg_name <ul style="list-style-type: none"> <li>example: svm1:volume1</li> <li>Introduced in: 9.6</li> </ul> |
| svm                       | <a href="#">svm</a>                                |   |

## policy

Basic policy information of the relationship.

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

## transfer

Basic information on the current transfer.

| Name              | Type                   | Description        |
|-------------------|------------------------|--------------------|
| _links            | <a href="#">_links</a> |                    |
| bytes_transferred | integer                | Bytes transferred. |

| Name  | Type   | Description |
|-------|--------|-------------|
| 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 |

## 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.

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".

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

## 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. 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 failback 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/?failback=true" '{"state":"snapmirrored"}'
```

To failback 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/?failback=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"}'
```

## Learn more

- [DOC /snapmirror/relationships](#)

## Parameters

| Name     | Type    | In    | Required | Description   |
|----------|---------|-------|----------|---|
| uuid     | string  | path  | True     | Relationship UUID   |
| 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> |

| Name           | Type    | In    | Required | Description   |
|----------------|---------|-------|----------|---|
| 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> |
| 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>  |

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

| 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> |

| 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, or Consistency Group. 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 &amp;lt;destination-SVM-name:&amp;gt;, and for Consistency Group, the "destination.path" must be specified as &amp;lt;destination-SVM-name:/cg/consistency-group-name&amp;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, or a Consistency Group 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</p> |
|                    |   |   |

| Name              | Type                                | Description   |
|-------------------|-------------------------------------|---|
| destination       | <a href="#">snapmirror_endpoint</a> | 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, and Consistency Group 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. |
| 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            | <a href="#">policy</a>              | 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 property is applicable only for relationships with FlexVol volume or FlexGroup volume endpoints and when the PATCH state is being changed to "snapmirrored".   |

| Name                | Type    | Description   |
|---------------------|---------|---|
| 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 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.  |
| 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_endpoint</a> | 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, and Consistency Group 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  |
|-------|--------|--|
| 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", and "synchronizing" 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   |
|------------------|---|---|
| transfer         | <a href="#">transfer</a>                  | Basic information on the current transfer.  |
| 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                                    |   |

## Example request

```
{
  "_links": {
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "consistency_group_failover": {
    "error": {
      "arguments": {
        "code": "string",
        "message": "string"
      },
      "code": "4",
      "message": "entry doesn't exist",
      "target": "uuid"
    }
  },
  "create_destination": {
    "storage_service": {
      "name": "extreme"
    },
    "tiering": {
      "policy": "all"
    }
  },
  "destination": {
    "cluster": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "cluster1",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "consistency_group_volumes": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "volume1",
      "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
    }
  },
}
```

```

"ipospace": "Default",
"path": "svm1:volume1",
"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": "Asynchronous",
  "type": "async",
  "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
},
"source": {
  "cluster": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "cluster1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "consistency_group_volumes": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "volume1",
    "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
  },
  "ipospace": "Default",
  "path": "svm1:volume1",
  "svm": {

```

```

    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "svm1",
    "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
  },
  "state": "snapmirrored",
  "transfer": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "state": "aborted",
    "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  | <a href="#">job_link</a> |             |

## 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                           |

| 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 was 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. |

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

error\_arguments

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

error

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

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>  |             |
| status | <a href="#">status</a> |             |

### 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></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 &amp;dash; 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 &amp;dash; This policy allows tiering of both destination endpoint Snapshot copies and the active file system user data to the cloud store none &amp;dash; Destination endpoint volume blocks will not be tiered to the cloud store. snapshot_only &amp;dash; 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, or Consistency Group. 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 <dp-volume-name>, for SVM data protection, the "destination.path" must be specified as <destination-svm-name>, and for Consistency Group, the "destination.path" must be specified as <destination-svm-name> 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, or a Consistency Group 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.</destination-svm-name></destination-svm-name></dp-volume-name>

| Name            | Type                            | Description  |
|-----------------|---------------------------------|--|
| 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. |
| 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  |
|--------|------------------------|--|
| _links | <a href="#">_links</a> |  |
| name   | string                 | The name of the volume.  |
| uuid   | string                 | <p>Unique identifier for 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> </ul> |

#### svm

| Name   | Type                   | Description                       |
|--------|------------------------|-----------------------------------|
| _links | <a href="#">_links</a> |                                   |
| name   | string                 | The name of the SVM.              |
| uuid   | string                 | The unique identifier of the SVM. |

#### 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, and Consistency Group 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.   |
| ipspace                   | string   | Optional property to specify the IPspace of the SVM.  |
| path                      | string   | ONTAP FlexVol/FlexGroup - svm1:volume1<br>ONTAP SVM - svm1:<br>ONTAP Consistency Group - svm1:/cg/cg_name <ul style="list-style-type: none"> <li>• example: svm1:volume1</li> <li>• Introduced in: 9.6</li> </ul> |
| svm                       | <a href="#">svm</a>                                |   |

## policy

Basic policy information of the relationship.

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

## transfer

Basic information on the current transfer.

| Name              | Type                   | Description        |
|-------------------|------------------------|--------------------|
| _links            | <a href="#">_links</a> |                    |
| bytes_transferred | integer                | Bytes transferred. |

| Name  | Type   | Description |
|-------|--------|-------------|
| 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. 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>                                |   |
| 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  |
|--------------------|---|--|
| 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, or Consistency Group. 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 &amp;lt;destination-SVM-name:&amp;gt;, and for Consistency Group, the "destination.path" must be specified as &amp;lt;destination-SVM-name:/cg/consistency-group-name&amp;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, or a Consistency Group destination endpoint, the properties in this object can be specified either from the source or the destination</p> |

| Name              | Type                                | Description   |
|-------------------|-------------------------------------|---|
| destination       | <a href="#">snapmirror_endpoint</a> | 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, and Consistency Group 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. |
| 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            | <a href="#">policy</a>              | 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 property is applicable only for relationships with FlexVol volume or FlexGroup volume endpoints and when the PATCH state is being changed to "snapmirrored".   |

| Name                | Type    | Description   |
|---------------------|---------|---|
| 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 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.  |
| 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_endpoint</a> | 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, and Consistency Group 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  |
|-------|--------|--|
| 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", and "synchronizing" 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   |
|------------------|---|---|
| transfer         | <a href="#">transfer</a>                  | Basic information on the current transfer.  |
| 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                                    |   |

job\_link

| Name   | Type                   | Description   |
|--------|------------------------|---|
| _links | <a href="#">_links</a> |   |
| uuid   | string                 | The UUID of the asynchronous job that is triggered by a POST, PATCH, or DELETE operation. |

## Copyright information

Copyright © 2024 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.