NetApp

Manage SnapMirror relationships

ONTAP 9.12.1 REST API reference

NetApp
February 13, 2024

Table of Contents

Manage SnapMirror relationships

SnapMirror relationships endpoint overview

Overview

This API manages asynchronous extended data protection (XDP) relationships for FlexVols, FlexGroups, object store, or SVMs. It is also used to create and 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". It supports the SnapMirror policy types "mirror-vault", "async-mirror" with "copy_all_source_snapshots", "vault" with "create_snapshot_on_source" set to "false", "sync-mirror", and "strict-sync-mirror". 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. In the case of a SnapMirror relationship for a Consistency Group of FlexVol volumes, note that the SnapMirror creation will result in the creation of Consistency Group on the source cluster if it did not already exist with the exact same name and set of FlexVol volumes specified in the current operation. Additionally, if the specified Consistency Group was already present and was already a part of an existing SnapMirror relationship, the process will fail. On the other hand, the creation of Consistency Group on destination cluster is a part of SnapMirror creation workflow itself. 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. DP FlexVol SnapMirror relationships cannot be created using this API but existing relationships can be listed or managed. The SnapMirror functionality is subdivided into relationship APIs and transfer APIs:

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

- SnapMirror transfer APIs are used to manage data transfers.

Retrieve information for SnapMirror relationships

GET `/snapmirror/relationships`

Introduced In: 9.6

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

Related ONTAP commands

- `snapmirror show`

- `snapmirror list-destinations`

Expensive properties

* `fabriclink.pull_byte_count`
* `fabriclink.push_byte_count`
* `fabriclink.pending_work_task_count`
* `fabriclink.status`
* `source.consistency_group_volumes.name`
* `destination.consistency_group_volumes.name`

Examples

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

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

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

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

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

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

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

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

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

1. Retrieving the destination UUID information for object store SnapMirror relationships.

```
GET "/api/snapmirror/relationships/?fields=destination.uuid"
```

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.
destination.cluster.name	string	query	False	Filter by destination.cluster.name • Introduced in: 9.7
destination.cluster.uuid	string	query	False	Filter by destination.cluster.uuid • Introduced in: 9.7
destination.uuid	string	query	False	Filter by destination.uuid • Introduced in: 9.11
destination.svm.uuid	string	query	False	Filter by destination.svm.uuid
destination.svm.name	string	query	False	Filter by destination.svm.name
destination.path	string	query	False	Filter by destination.path
destination.consistency_group_volumes.name	string	query	False	Filter by destination.consistency_group_volumes.name • Introduced in: 9.8
lag_time	string	query	False	Filter by lag_time
healthy	boolean	query	False	Filter by healthy

Name	Type	In	Required	Description
uuid	string	query	False	Filter by uuid
policy.type	string	query	False	Filter by policy.type
policy.name	string	query	False	Filter by policy.name
policy.uuid	string	query	False	Filter by policy.uuid
exported_snapshot	string	query	False	Filter by exported_snapshot
identity_preservation	string	query	False	Filter by identity_preservation • Introduced in: 9.11
consistency_group_failover.error.target	string	query	False	Filter by consistency_group_failover.error.target • Introduced in: 9.8
consistency_group_failover.error.arguments.message	string	query	False	Filter by consistency_group_failover.error.arguments.message • Introduced in: 9.8
consistency_group_failover.error.arguments.code	string	query	False	Filter by consistency_group_failover.error.arguments.code • Introduced in: 9.8
consistency_group_failover.error.code	string	query	False	Filter by consistency_group_failover.error.code • Introduced in: 9.8

Name	Type	In	Required	Description
consistency_group_failover.error.message	string	query	False	Filter by consistency_group_failover.error.message • Introduced in: 9.8
consistency_group_failover.status.code	string	query	False	Filter by consistency_group_failover.status.code • Introduced in: 9.8
consistency_group_failover.status.message	string	query	False	Filter by consistency_group_failover.status.message • Introduced in: 9.8
throttle	integer	query	False	Filter by throttle • Introduced in: 9.11
group_type	string	query	False	Filter by group_type • Introduced in: 9.11
transfer_schedule.name	string	query	False	Filter by transfer_schedule.name • Introduced in: 9.11
transfer_schedule.uuid	string	query	False	Filter by transfer_schedule.uuid • Introduced in: 9.11

Name	Type	In	Required	Description
last_transfer_type	string	query	False	Filter by last_transfer_type • Introduced in: 9.11
unhealthy_reason.parameters	string	query	False	Filter by unhealthy_reason.parameters
unhealthy_reason.code	integer	query	False	Filter by unhealthy_reason.code
unhealthy_reason.message	string	query	False	Filter by unhealthy_reason.message
transfer.total_duration	string	query	False	Filter by transfer.total_duration • Introduced in: 9.9
transfer.uuid	string	query	False	Filter by transfer.uuid
transfer.state	string	query	False	Filter by transfer.state
transfer.bytes_transferred	integer	query	False	Filter by transfer.bytes_transferred
transfer.end_time	string	query	False	Filter by transfer.end_time • Introduced in: 9.9
state	string	query	False	Filter by state
restore	boolean	query	False	Filter by restore

Name	Type	In	Required	Description
source.cluster.name	string	query	False	Filter by source.cluster.name • Introduced in: 9.7
source.cluster.uuid	string	query	False	Filter by source.cluster.uuid • Introduced in: 9.7
source.uuid	string	query	False	Filter by source.uuid • Introduced in: 9.11
source.svm.uuid	string	query	False	Filter by source.svm.uuid
source.svm.name	string	query	False	Filter by source.svm.name
source.path	string	query	False	Filter by source.path
source.consistency_group_volumes.name	string	query	False	Filter by source.consistency_group_volumes.name • Introduced in: 9.8
fields	array[string]	query	False	Specify the fields to return.
max_records	integer	query	False	Limit the number of records returned.
return_records	boolean	query	False	The default is true for GET calls. When set to false, only the number of records is returned. • Default value: 1

Name	Type	In	Required	Description
return_timeout	integer	query	False	The number of seconds to allow the call to execute before returning. When iterating over a collection, the default is 15 seconds. ONTAP returns earlier if either max records or the end of the collection is reached. • Default value: 1 • Max value: 120 • Min value: 0
order_by	array[string]	query	False	Order results by specified fields and optional [asc

Response

```
Status: 200, Ok
```

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

Example response

```json
{
  "_links": {
    "next": {
      "href": "/api/resourcelink"
    },
    "self": {
      "href": "/api/resourcelink"
    }
  },
  "num_records": 1,
  "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": {
            "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"
          },
          "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
        },
        "exported_snapshot": "string",
        "fabriclink": {
          "destination_role": "active_mirror",
          "source_role": "active_mirror"
        },
        "group_type": "consistency_group",
        "identity_preservation": "full",
        "lag_time": "PT8H35M42S",
        "last_transfer_type": "initialize",
        "policy": {
          "_links": {
            "self": {
              "href": "/api/resourcelink"
            }
          },
          "name": "Asynchronous",
          "transfer_schedule": {
            "name": "hourly"
          },
          "type": "async",
          "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
        },
        "source": {
          "cluster": {
            "_links": {
              "self": {
                "href": "/api/resourcelink"
              }
```

```json
    },
    "name": "cluster1",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "consistency_group_volumes": {
      "name": "volume1",
      "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
    },
    "ipspace": "Default",
    "path": "svm1:volume1",
    "svm": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "svm1",
      "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
    },
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
  },
  "state": "snapmirrored",
  "throttle": 0,
  "transfer": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "end_time": "2020-12-02T18:36:19-08:00",
    "state": "aborted",
    "total_duration": "PT28M41S",
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
  },
  "transfer_schedule": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "weekly",
    "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
  },
  "unhealthy_reason": [
    {
      "code": "6621444",
```

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

Error

```
Status: Default
```

ONTAP Error Response codes

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

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
next	href	
self	href	

_links

Name	Type	Description
self	href	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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	error	
status	status	

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	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. • enum: ["extreme", "performance", "value"] • Introduced in: 9.6

tiering

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

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

snapmirror_destination_creation

Use this object to provision the destination endpoint when establishing a SnapMirror relationship for a FlexVol volume, FlexGroup volume, SVM, Consistency Group or ONTAP S3 Bucket. Given a source endpoint, the destination endpoint is provisioned in the SVM specified in the "destination.path" property. While protecting an SVM, the SVM destination endpoint can only be provisioned on the local cluster. To provision the SVM destination endpoint use the optional "source.cluster.name" property to specify the remote cluster name or use the optional "source.cluster.uuid" property to specify the remote cluster UUID. When "create_destination.enabled" option is specified while making a POST for a SnapMirror relationship, the relationship can be automatically initialized by setting the "state" either to "snapmirrored" when the policy is of type "async" or to "in_sync" when the policy is of type "sync". The "destination.path" property must specify the destination endpoint path. For example, for FlexVol volume and FlexGroup volume, the "destination.path" can be specified as <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, Consistency Group or a Bucket destination endpoint, the properties in this object can be specified either from the source or the destination cluster. For an SVM destination endpoint, the properties in this object can be specified from the destination cluster only. This object is not supported for non ONTAP endpoints. While protecting a S3 Bucket, the optional "size" property can be used to create ONTAP S3 Bucket destination endpoint of the specified size.</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.
size	integer	Optional property to specify the size of destination endpoint in bytes. This property is applicable only to ONTAP S3 Bucket endpoints. The minimum size for S3 bucket is 80MB and maximum size is 64TB. If not specified, system will create destination with default size of 800GB. • Introduced in: 9.10
storage_service	storage_service	
tiering	tiering	

cluster

Name	Type	Description
_links	_links	
name	string	
uuid	string	

consistency_group_volumes

Name	Type	Description
name	string	The name of the volume.
uuid	string	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. • example: 028baa66-41bd-11e9-81d5-00a0986138f7 • x-ntap-createOnly: true • Introduced in: 9.8

svm

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

snapmirror_endpoint

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

Name	Type	Description
cluster	cluster	
consistency_group_volumes	array[consistency_group_volumes]	Mandatory property for a Consistency Group endpoint. Specifies the list of FlexVol volumes for a Consistency Group.
ipspace	string	Optional property to specify the IPSpace of the SVM.
path	string	ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name ONTAP S3 - svm1:/bucket/bucket1 NON-ONTAP - objstore1:/objstore • example: svm1:volume1 • Introduced in: 9.6
svm	svm	
uuid	string	UUID of the endpoint. Applicable for object store SnapMirror relationships only.

fabriclink

SnapMirror FabricLink relationship

Name	Type	Description
destination_bucket	string	Specifies the name of the destination bucket of the FabricLink SnapMirror relationship. This parameter is supported for FabricLink SnapMirror relationships with ONTAP or external object store vendor endpoints. This parameter is only available on ONTAP 9.10.1 or later. • readOnly: 1
destination_role	string	This field represents the destination bucket's role within FabricLink. Possible values for this field are "active_mirror" and "backup". In a bi-directional ONTAP to ONTAP mirroring relationship, both FabricLink SnapMirror records will contain "active_mirror" in both the source-role and destination-role fields. This field conveys that both mirrors are capable of performing an active role at any time. In an ONTAP to Non-ONTAP FabricLink replication relationship, the source-role is "active_mirror" and the destination-role is "backup". This parameter is supported for FabricLink SnapMirror relationships with ONTAP or external object store vendor endpoints. This parameter is only available in ONTAP 9.10.1 or later. • enum: ["active_mirror", "backup"] • readOnly: 1

Name	Type	Description
pending_work_task_count	integer	Conveys how many discrete tasks the bucket in this unidirectional FabricLink SnapMirror relationship knows it will eventually have to perform to fully update the destination to reflect the source's state. Each work item might represent the need to push a newly created object from the source to the destination mirror, or perhaps to advise the destination mirror that an existing object has been modified or deleted. If the link is configured with an RPO of one hour and the user creates an object on the source bucket, that bucket will immediately show that it now has a pending work item to send that new object to the destination; FabricLink intentionally will not act on that work item until the hour has elapsed. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1
pull_byte_count	integer	Represents the approximate number of bytes of data that the destination bucket anticipates reading from the source bucket for the FabricLink SnapMirror relationship. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1
push_byte_count	integer	Represents the approximate number of bytes of data that must be pushed from the source bucket to the destination bucket for the FabricLink SnapMirror relationship. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1

Name	Type	Description
source_bucket	string	Specifies the name of the source bucket of the FabricLink SnapMirror relationship. This parameter is supported only for FabricLink SnapMirror relationships with ONTAP endpoints. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1
source_role	string	This field represents the source bucket's role within FabricLink. One possible value for this field is "active_mirror". In a bi-directional ONTAP to ONTAP mirroring relationship, both FabricLink SnapMirror records will contain "active_mirror" in both the source-role and destination-role fields. This field conveys that both mirrors are capable of performing an active role at any time. In an ONTAP to Non-ONTAP FabricLink replication relationship, the source-role is "active_mirror" and the destination-role is "backup". This parameter is supported only for FabricLink SnapMirror relationships with an ONTAP endpoint. This parameter is only available in ONTAP 9.10.1 or later. • enum: ["active_mirror"] • readOnly: 1
status	string	Specifies the status of the FabricLink SnapMirror relationship. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1

Name	Type	Description
topology_uuid	string	Topology UUID is used to refer to a unique data set. All the links that connect the same data set share the same topology UUID. Thus, in a fan-out topology, all links fanning from the same source bucket will carry the same topology-uuid value. Similarly, in a cascading mirror arrangement, all links in the cascade will carry the same topology-uuid value. This parameter is only available in ONTAP 9.10.1 or later. • format: uuid • readOnly: 1

transfer_schedule

The schedule used to update the asynchronous relationship. This is only applicable when provisioning a destination endpoint on a remote cluster with an effective cluster version of 9.6 or earlier. Only cron schedules are supported for SnapMirror.

Name	Type	Description
name	string	

policy

Basic policy information of the relationship.

Name	Type	Description
_links	_links	
name	string	
transfer_schedule	transfer_schedule	The schedule used to update the asynchronous relationship. This is only applicable when provisioning a destination endpoint on a remote cluster with an effective cluster version of 9.6 or earlier. Only cron schedules are supported for SnapMirror.
type	string	
uuid	string	

transfer

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

Name	Type	Description
_links	_links	
bytes_transferred	integer	Bytes transferred.
end_time	string	End time of the last transfer.
state	string	
total_duration	string	Transfer elapsed time.
uuid	string	Transfer UUID. This property is applicable only for active transfers.

transfer_schedule

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

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

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 relatiosnhip 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	_links	
consistency_group_failover	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. • Introduced in: 9.8 • readOnly: 1

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

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

Name	Type	Description
identity_preservation	string	Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type. This "identity_preservation" overrides the "identity_preservation" set on the SnapMirror relationship's policy.
lag_time	string	Time since the exported Snapshot copy was created.
last_transfer_type	string	Specifies the operation type of the last transfer that occured on the relationship. The *initialize* transfer occurs when the relationship state changes from uninitialized to snapmirrored or in_sync. The *update* transfer occurs when the snapshots are transferred from the source endpoint to the destination endpoint as part of scheduled or manual update. The *resync* transfer occurs when the relationship state changes from broken_off to snapmirrored or in_sync. The *restore* transfer occurs when the snapshot is restored from a destination endpoint to another endpoint.
policy	policy	Basic policy information of the relationship.
preserve	boolean	Set to true on resync to preserve Snapshot copies on the destination that are newer than the latest common Snapshot copy. This 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 SVMDR relationships when the PATCH state is being changed to "snapmirrored".
recover_after_break	boolean	Set to true to recover from a failed SnapMirror break operation on a FlexGroup volume relationship. This restores all destination FlexGroup constituent volumes to the latest Snapshot copy, and any writes to the read-write constituents are lost. This property is applicable only for SnapMirror relationships with FlexGroup volume endpoints and when the PATCH state is being changed to "broken_off".
restore	boolean	Set to true to create a relationship for restore. To trigger restore-transfer, use transfers POST on the restore relationship. SnapMirror relationships with the policy type "async" can be restored. SnapMirror relationships with the policy type "sync" cannot be restored.
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	snapmirror_endpoint	Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint. • Introduced in: 9.6

Name	Type	Description
state	string	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

Name	Type	Description
throttle	integer	Throttle, in KBs per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited.
transfer	transfer	Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.
transfer_schedule	transfer_schedule	Schedule used to update asynchronous relationships. This "transfer_schedule" overrides the "transfer_schedule" set on the SnapMirror relationship's policy. To remove the "transfer_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror.
unhealthy_reason	array[snapmirror_error]	Reason the relationship is not healthy. It is a concatenation of up to four levels of error messages.
uuid	string	

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.

Important notes

- Only "async" type policies with property "create_snapshot_on_source" set to false can be used in the creation of SnapMirror relationships having object store endpoints.

- Policies with property "network_compression_enabled" to "true" cannot be used in the creation of SnapMirror relationships having object store endpoints.

- Policies with property "retention.creation_schedule" cannot be used in the creation of SnapMirror relationships having object store endpoints.

- Policies with the property "snapshot_lock_mode" set to "compliance" or "enterprise" can only be used in the creation of SnapMirror relationships having a source endpoint as a FlexVol volume and a destination endpoint as an object store.

- Optional property "create_destination.size" is applicable only for ONTAP S3 Bucket destination endpoints.

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.

- `create_destination.size` - Specify the size to provision the ONTAP S3 Bucket 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*

- `throttle` - *0*

Related ONTAP commands

- `snapmirror create`
- `snapmirror protect`

Important notes

- The property "transfer_schedule" if set on a SnapMirror relationship overrides the "transfer_schedule" set on the policy being used with the SnapMirror relationship.
- The property "throttle" if set on a SnapMirror relationship overrides the "throttle" set on the policy being used with the SnapMirror relationship.
- The properties "transfer_schedule" and "throttle" are not supported when "restore" is set to "true".
- The property "transfer_schedule" cannot be set to null (no-quotes) during SnapMirror relationship POST.
- The property "throttle" is not supported when "create_destination.enabled" is set to "true".
- The property "identity_preservation" is applicable to only SnapMirror relationships with SVM endpoints and it indicates which configuration of the source SVM is replicated to the destination SVM.

Examples

The following examples show how to create FlexVol, FlexGroup, object store, 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": { "enabled": "true" }}'
```

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

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

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

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

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

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

Create 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": { "enabled": "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": { "enabled": "true", "storage_service": { "enabled":
"true", "name": "extreme", "enforce_performance": "true" } }, "policy":
"AutomatedFailOver", "state": "in_sync" }'
```

Creating a FlexVol volume SnapMirror relationship of type XDP with transfer_schedule and throttle.

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

Creating an object store SnapMirror relationship of type XDP.

```
  POST "/api/snapmirror/relationships/" '{"source": {"path":
"test_vserv_src:source_vol_rw"}, "destination": { "path":
"S3store1:/objstore/object1"}}'
```

Creating a SnapMirror relationship in order to restore from an object store destination.

```
  POST "/api/snapmirror/relationships/" '{"source": {"path":
"S3store1:/objstore/object1", "uuid":"817500fa-092d-44c5-9c10-
7b54f7b2f20a"}, "destination": { "path": "test_vserv_dst:dst_vol_dp"},
"restore": "true"}'
```

Creating an object store SnapMirror relationship in order to resync.

```
 POST "/api/snapmirror/relationships/" '{"source": {"path":
"test_vserv_src:source_vol_rw2"}, "destination": { "path":
"S3store1:/objstore/object1", "uuid":"817500fa-092d-44c5-9c10-
7b54f7b2f20a"}}'
```

Learn more

- DOC /snapmirror/relationships

Parameters

Name	Type	In	Required	Description
return_records	boolean	query	False	The default is false. If set to true, the records are returned. • Default value:

Name	Type	In	Required	Description
return_timeout	integer	query	False	The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202. • Default value: 1 • Max value: 120 • Min value: 0
validate_only	boolean	query	False	Validate the operation and its parameters, without actually performing the operation. • Introduced in: 9.7

Request Body

Name	Type	Description
_links	_links	

Name	Type	Description
consistency_group_failover	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. • Introduced in: 9.8 • readOnly: 1

Name	Type	Description
create_destination	snapmirror_destination_creation	Use this object to provision the destination endpoint when establishing a SnapMirror relationship for a FlexVol volume, FlexGroup volume, SVM, Consistency Group or ONTAP S3 Bucket. Given a source endpoint, the destination endpoint is provisioned in the SVM specified in the "destination.path" property. While protecting an SVM, the SVM destination endpoint can only be provisioned on the local cluster. To provision the SVM destination endpoint use the optional "source.cluster.name" property to specify the remote cluster name or use the optional "source.cluster.uuid" property to specify the remote cluster UUID. When "create_destination.enabled" option is specified while making a POST for a SnapMirror relationship, the relationship can be automatically initialized by setting the "state" either to "snapmirrored" when the policy is of type "async" or to "in_sync" when the policy is of type "sync". The "destination.path" property must specify the destination endpoint path. For example, for FlexVol volume and FlexGroup volume, the "destination.path" can be specified as <destination-SVM-name:dp-volume-name>, for SVM data protection, the "destination.path" must be specified as <destination-SVM-name:>, and for Consistency Group, the "destination.path" must be specified as <destination-SVM-name:/cg/consistency-group-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, Consistency Group or a Bucket destination endpoint, the properties in this object can be specified either from the source or the destination cluster. For an SVM destination endpoint, the properties in this object can be specified from 41

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

Name	Type	Description
identity_preservation	string	Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type. This "identity_preservation" overrides the "identity_preservation" set on the SnapMirror relationship's policy.
lag_time	string	Time since the exported Snapshot copy was created.
last_transfer_type	string	Specifies the operation type of the last transfer that occured on the relationship. The *initialize* transfer occurs when the relationship state changes from uninitialized to snapmirrored or in_sync. The *update* transfer occurs when the snapshots are transferred from the source endpoint to the destination endpoint as part of scheduled or manual update. The *resync* transfer occurs when the relationship state changes from broken_off to snapmirrored or in_sync. The *restore* transfer occurs when the snapshot is restored from a destination endpoint to another endpoint.
policy	policy	Basic policy information of the relationship.
preserve	boolean	Set to true on resync to preserve Snapshot copies on the destination that are newer than the latest common Snapshot copy. This 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 SVMDR relationships when the PATCH state is being changed to "snapmirrored".
recover_after_break	boolean	Set to true to recover from a failed SnapMirror break operation on a FlexGroup volume relationship. This restores all destination FlexGroup constituent volumes to the latest Snapshot copy, and any writes to the read-write constituents are lost. This property is applicable only for SnapMirror relationships with FlexGroup volume endpoints and when the PATCH state is being changed to "broken_off".
restore	boolean	Set to true to create a relationship for restore. To trigger restore-transfer, use transfers POST on the restore relationship. SnapMirror relationships with the policy type "async" can be restored. SnapMirror relationships with the policy type "sync" cannot be restored.
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	snapmirror_endpoint	Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint. • Introduced in: 9.6

Name	Type	Description
state	string	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

Name	Type	Description
throttle	integer	Throttle, in KBs per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited.
transfer	transfer	Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.
transfer_schedule	transfer_schedule	Schedule used to update asynchronous relationships. This "transfer_schedule" overrides the "transfer_schedule" set on the SnapMirror relationship's policy. To remove the "transfer_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror.
unhealthy_reason	array[snapmirror_error]	Reason the relationship is not healthy. It is a concatenation of up to four levels of error messages.
uuid	string	

Example request

```json
{
  "_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": {
      "name": "volume1",
      "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
    },
    "ipspace": "Default",
    "path": "svm1:volume1",
    "svm": {
      "_links": {
        "self": {
```

```json
          "href": "/api/resourcelink"
        }
      },
      "name": "svm1",
      "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
    },
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
  },
  "exported_snapshot": "string",
  "fabriclink": {
    "destination_role": "active_mirror",
    "source_role": "active_mirror"
  },
  "group_type": "consistency_group",
  "identity_preservation": "full",
  "lag_time": "PT8H35M42S",
  "last_transfer_type": "initialize",
  "policy": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "Asynchronous",
    "transfer_schedule": {
      "name": "hourly"
    },
    "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": {
      "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"
      },
      "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
    },
    "state": "snapmirrored",
    "throttle": 0,
    "transfer": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "end_time": "2020-12-02T18:36:19-08:00",
      "state": "aborted",
      "total_duration": "PT28M41S",
      "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
    },
    "transfer_schedule": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "weekly",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "unhealthy_reason": [
      {
        "code": "6621444",
        "message": "Failed to complete update operation on one or more
item relationships.",
        "parameters": []
      },
      {
        "code": "6621445",
        "message": "Group Update failed",
        "parameters": []
      }
    ],
```

```
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
  }
}
```

Response

```
Status: 202, Accepted
```

Name	Type	Description
job	job_link	

Example response

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

Headers

Name	Description	Type
Location	Useful for tracking the resource location	string

Error

```
Status: Default
```

ONTAP Error Response Codes

Error Code	Description
1115545	Access token has an invalid signature.
1115546	Access token expired at a time.
1115547	Access token not valid until a time.

Error Code	Description
1115548	Access token is malformed.
1115549	Internal error. Failed to validate access token.
6619637	Relationship with specified destination volume already exists.
6619699	Schedule not found.
6620374	Internal error. Failed to get SVM information.
6620478	Internal error. Failed to check SnapMirror capability.
6621834	Object store configuration does not exist for specified vserver.
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.
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.

Error Code	Description
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.
13303897	Specifying transfer_schedule is only supported for an asynchronous SnapMirror relationship with a remote destination cluster that has an effective cluster version of 9.6 or earlier.
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.

Error Code	Description
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".
13303945	Schedule specified cannot be associated with the relationship because the policy used for the relationship has a schedule.
13303953	SnapMirror relationships between FlexGroup volumes and object store endpoints are not supported.
13303957	Restore to the specified destination endpoint is not supported.
13303958	The \"source.uuid\" property must be specified.
13303959	The \"source.uuid\" property can only be specified when using the \"restore\" property to create a SnapMirror relationship from an object store endpoint.
13303960	The \"destination.uuid\" property cannot be specified for these endpoints.
13303961	Creating a destination endpoint is not supported with the \"source.uuid\" property.
13303962	Creating a destination endpoint is not supported with the \"destination.uuid\" property.
13303964	Specified property is not supported for object store relationships.
13303965	Specified endpoint not found.
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.

Error Code	Description
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.
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.
13304026	API license token is required for this operation.
13304027	Invalid API license token specified.
13304032	In an "All SAN Array", an SVM-DR relationship is not supported when the associated SnapMirror policy does not have the "identity_preservation" property set to "exclude_network_and_protocol_config".
13304034	SnapMirror Cloud license must be installed to use this feature.
13304035	SnapMirror Cloud license capacity limit has been exceeded.
13304036	SnapmMirror Cloud license term has expired.
13304038	SnapMirror Cloud License capacity limit would be exceeded with this SnapMirror operation.
13304042	The specified destination or source cluster for relationships with an object store destination must be local.

Error Code	Description
13304043	The property must be specified for a restore from an object store endpoint.
13304044	File restore from object store endpoints is not supported for relationship restore.
13304045	Incremental restore from an object store endpoint is not supported.
13304077	Specified property is only supported for creating a destination endpoint for FabricLink SnapMirror relationships.</private>
13304080	Specified UUID and name do not match.
13304082	Specified properties are mutually exclusive.
13304083	The specified property is not supported because all nodes in the cluster are not capable of supporting the property.<private></private>
13304098	This SnapMirror policy is not supported for SnapMirror relationhips with SnapLock volumes.
13304099	SnapLock Compliance Clock is not running on all nodes in the destination cluster.
13304108	Schedule not found in the Administrative SVM or the SVM for the relationship.

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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	error	
status	status	

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	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. • enum: ["extreme", "performance", "value"] • Introduced in: 9.6

tiering

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

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

snapmirror_destination_creation

Use this object to provision the destination endpoint when establishing a SnapMirror relationship for a FlexVol volume, FlexGroup volume, SVM, Consistency Group or ONTAP S3 Bucket. Given a source endpoint, the destination endpoint is provisioned in the SVM specified in the "destination.path" property. While protecting an SVM, the SVM destination endpoint can only be provisioned on the local cluster. To provision the SVM destination endpoint use the optional "source.cluster.name" property to specify the remote cluster name or use the optional "source.cluster.uuid" property to specify the remote cluster UUID. When "create_destination.enabled" option is specified while making a POST for a SnapMirror relationship, the relationship can be automatically initialized by setting the "state" either to "snapmirrored" when the policy is of type "async" or to "in_sync" when the policy is of type "sync". The "destination.path" property must specify the destination endpoint path. For example, for FlexVol volume and FlexGroup volume, the "destination.path" can be specified as <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, Consistency Group or a Bucket destination endpoint, the properties in this object can be specified either from the source or the destination cluster. For an SVM destination endpoint, the properties in this object can be specified from the destination cluster only. This object is not supported for non ONTAP endpoints. While protecting a S3 Bucket, the optional "size" property can be used to create ONTAP S3 Bucket destination endpoint of the specified size.</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.
size	integer	Optional property to specify the size of destination endpoint in bytes. This property is applicable only to ONTAP S3 Bucket endpoints. The minimum size for S3 bucket is 80MB and maximum size is 64TB. If not specified, system will create destination with default size of 800GB. • Introduced in: 9.10
storage_service	storage_service	
tiering	tiering	

cluster

Name	Type	Description
_links	_links	
name	string	
uuid	string	

consistency_group_volumes

Name	Type	Description
name	string	The name of the volume.
uuid	string	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. • example: 028baa66-41bd-11e9-81d5-00a0986138f7 • x-ntap-createOnly: true • Introduced in: 9.8

svm

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

snapmirror_endpoint

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

Name	Type	Description
cluster	cluster	
consistency_group_volumes	array[consistency_group_volumes]	Mandatory property for a Consistency Group endpoint. Specifies the list of FlexVol volumes for a Consistency Group.
ipspace	string	Optional property to specify the IPSpace of the SVM.
path	string	ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name ONTAP S3 - svm1:/bucket/bucket1 NON-ONTAP - objstore1:/objstore • example: svm1:volume1 • Introduced in: 9.6
svm	svm	
uuid	string	UUID of the endpoint. Applicable for object store SnapMirror relationships only.

fabriclink

SnapMirror FabricLink relationship

Name	Type	Description
destination_bucket	string	Specifies the name of the destination bucket of the FabricLink SnapMirror relationship. This parameter is supported for FabricLink SnapMirror relationships with ONTAP or external object store vendor endpoints. This parameter is only available on ONTAP 9.10.1 or later. • readOnly: 1
destination_role	string	This field represents the destination bucket's role within FabricLink. Possible values for this field are "active_mirror" and "backup". In a bi-directional ONTAP to ONTAP mirroring relationship, both FabricLink SnapMirror records will contain "active_mirror" in both the source-role and destination-role fields. This field conveys that both mirrors are capable of performing an active role at any time. In an ONTAP to Non-ONTAP FabricLink replication relationship, the source-role is "active_mirror" and the destination-role is "backup". This parameter is supported for FabricLink SnapMirror relationships with ONTAP or external object store vendor endpoints. This parameter is only available in ONTAP 9.10.1 or later. • enum: ["active_mirror", "backup"] • readOnly: 1

Name	Type	Description
pending_work_task_count	integer	Conveys how many discrete tasks the bucket in this unidirectional FabricLink SnapMirror relationship knows it will eventually have to perform to fully update the destination to reflect the source's state. Each work item might represent the need to push a newly created object from the source to the destination mirror, or perhaps to advise the destination mirror that an existing object has been modified or deleted. If the link is configured with an RPO of one hour and the user creates an object on the source bucket, that bucket will immediately show that it now has a pending work item to send that new object to the destination; FabricLink intentionally will not act on that work item until the hour has elapsed. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1
pull_byte_count	integer	Represents the approximate number of bytes of data that the destination bucket anticipates reading from the source bucket for the FabricLink SnapMirror relationship. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1
push_byte_count	integer	Represents the approximate number of bytes of data that must be pushed from the source bucket to the destination bucket for the FabricLink SnapMirror relationship. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1

Name	Type	Description
source_bucket	string	Specifies the name of the source bucket of the FabricLink SnapMirror relationship. This parameter is supported only for FabricLink SnapMirror relationships with ONTAP endpoints. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1
source_role	string	This field represents the source bucket's role within FabricLink. One possible value for this field is "active_mirror". In a bi-directional ONTAP to ONTAP mirroring relationship, both FabricLink SnapMirror records will contain "active_mirror" in both the source-role and destination-role fields. This field conveys that both mirrors are capable of performing an active role at any time. In an ONTAP to Non-ONTAP FabricLink replication relationship, the source-role is "active_mirror" and the destination-role is "backup". This parameter is supported only for FabricLink SnapMirror relationships with an ONTAP endpoint. This parameter is only available in ONTAP 9.10.1 or later. • enum: ["active_mirror"] • readOnly: 1
status	string	Specifies the status of the FabricLink SnapMirror relationship. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1

Name	Type	Description
topology_uuid	string	Topology UUID is used to refer to a unique data set. All the links that connect the same data set share the same topology UUID. Thus, in a fan-out topology, all links fanning from the same source bucket will carry the same topology-uuid value. Similarly, in a cascading mirror arrangement, all links in the cascade will carry the same topology-uuid value. This parameter is only available in ONTAP 9.10.1 or later. • format: uuid • readOnly: 1

transfer_schedule

The schedule used to update the asynchronous relationship. This is only applicable when provisioning a destination endpoint on a remote cluster with an effective cluster version of 9.6 or earlier. Only cron schedules are supported for SnapMirror.

Name	Type	Description
name	string	

policy

Basic policy information of the relationship.

Name	Type	Description
_links	_links	
name	string	
transfer_schedule	transfer_schedule	The schedule used to update the asynchronous relationship. This is only applicable when provisioning a destination endpoint on a remote cluster with an effective cluster version of 9.6 or earlier. Only cron schedules are supported for SnapMirror.
type	string	
uuid	string	

transfer

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

Name	Type	Description
_links	_links	
bytes_transferred	integer	Bytes transferred.
end_time	string	End time of the last transfer.
state	string	
total_duration	string	Transfer elapsed time.
uuid	string	Transfer UUID. This property is applicable only for active transfers.

transfer_schedule

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

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

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 relatiosnhip 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	_links	
consistency_group_failover	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. • Introduced in: 9.8 • readOnly: 1

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

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

Name	Type	Description
identity_preservation	string	Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type. This "identity_preservation" overrides the "identity_preservation" set on the SnapMirror relationship's policy.
lag_time	string	Time since the exported Snapshot copy was created.
last_transfer_type	string	Specifies the operation type of the last transfer that occured on the relationship. The *initialize* transfer occurs when the relationship state changes from uninitialized to snapmirrored or in_sync. The *update* transfer occurs when the snapshots are transferred from the source endpoint to the destination endpoint as part of scheduled or manual update. The *resync* transfer occurs when the relationship state changes from broken_off to snapmirrored or in_sync. The *restore* transfer occurs when the snapshot is restored from a destination endpoint to another endpoint.
policy	policy	Basic policy information of the relationship.
preserve	boolean	Set to true on resync to preserve Snapshot copies on the destination that are newer than the latest common Snapshot copy. This 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 SVMDR relationships when the PATCH state is being changed to "snapmirrored".
recover_after_break	boolean	Set to true to recover from a failed SnapMirror break operation on a FlexGroup volume relationship. This restores all destination FlexGroup constituent volumes to the latest Snapshot copy, and any writes to the read-write constituents are lost. This property is applicable only for SnapMirror relationships with FlexGroup volume endpoints and when the PATCH state is being changed to "broken_off".
restore	boolean	Set to true to create a relationship for restore. To trigger restore-transfer, use transfers POST on the restore relationship. SnapMirror relationships with the policy type "async" can be restored. SnapMirror relationships with the policy type "sync" cannot be restored.
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	snapmirror_endpoint	Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint. • Introduced in: 9.6

Name	Type	Description
state	string	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

Name	Type	Description
throttle	integer	Throttle, in KBs per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited.
transfer	transfer	Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.
transfer_schedule	transfer_schedule	Schedule used to update asynchronous relationships. This "transfer_schedule" overrides the "transfer_schedule" set on the SnapMirror relationship's policy. To remove the "transfer_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror.
unhealthy_reason	array[snapmirror_error]	Reason the relationship is not healthy. It is a concatenation of up to four levels of error messages.
uuid	string	

job_link

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

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.

- When a SnapMirror relationship associated with a pair of source and destination Consistency Groups is deleted, the corresponding Consistency Groups on the source and destination clusters are not automatically deleted and remain in place.

Related ONTAP commands

- `snapmirror delete`

- `snapmirror release`

Examples

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

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

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

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

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

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

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

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

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

Learn more

- DOC /snapmirror/relationships

Parameters

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

Name	Type	In	Required	Description
source_info_only	boolean	query	False	Deletes relationship information on the source only. This parameter is applicable only when the call is executed on the cluster that contains the source endpoint.
return_timeout	integer	query	False	The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202. • Default value: 1 • Max value: 120 • Min value: 0

Response

```
Status: 202, Accepted
```

Name	Type	Description
job	job_link	

Example response

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

Error

```
Status: Default
```

ONTAP Error Response codes

Error code	Description
13303825	Could not retrieve information for the SnapMirror policy type
13303814	Could not retrieve the source or destination SVM UUID
13303815	Could not retrieve information for the peer cluster
13303822	SnapMirror release has failed
13303813	SnapMirror release was successful but delete has failed
13303854	Cleanup of restore relationship failed
13303855	DELETE call on a restore relationship does not support the given flags
13303865	Deleting the specified SnapMirror policy is not supported.
6619715	Modification of relationship is in progress. Retry the command after a few minutes.

Name	Type	Description
error	error	

Example error

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

Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

job_link

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

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

Retrieve a SnapMirror relationship

GET `/snapmirror/relationships/{uuid}`

Introduced In: 9.6

Retrieves a SnapMirror relationship.

Related ONTAP commands

- `snapmirror show`

- `snapmirror list-destinations`

Expensive properties

- `fabriclink.pull_byte_count`

- `fabriclink.push_byte_count`

- `fabriclink.pending_work_task_count`

- `fabriclink.status`

- `source.consistency_group_volumes.name`

- `destination.consistency_group_volumes.name`

Example

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

Learn more

- DOC /snapmirror/relationships

Parameters

Name	Type	In	Required	Description
uuid	string	path	True	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	_links	
consistency_group_failover	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. • Introduced in: 9.8 • readOnly: 1

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

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

Name	Type	Description
identity_preservation	string	Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type. This "identity_preservation" overrides the "identity_preservation" set on the SnapMirror relationship's policy.
lag_time	string	Time since the exported Snapshot copy was created.
last_transfer_type	string	Specifies the operation type of the last transfer that occured on the relationship. The *initialize* transfer occurs when the relationship state changes from uninitialized to snapmirrored or in_sync. The *update* transfer occurs when the snapshots are transferred from the source endpoint to the destination endpoint as part of scheduled or manual update. The *resync* transfer occurs when the relationship state changes from broken_off to snapmirrored or in_sync. The *restore* transfer occurs when the snapshot is restored from a destination endpoint to another endpoint.
policy	[policy](#policy)	Basic policy information of the relationship.
preserve	boolean	Set to true on resync to preserve Snapshot copies on the destination that are newer than the latest common Snapshot copy. This 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 SVMDR relationships when the PATCH state is being changed to "snapmirrored".
recover_after_break	boolean	Set to true to recover from a failed SnapMirror break operation on a FlexGroup volume relationship. This restores all destination FlexGroup constituent volumes to the latest Snapshot copy, and any writes to the read-write constituents are lost. This property is applicable only for SnapMirror relationships with FlexGroup volume endpoints and when the PATCH state is being changed to "broken_off".
restore	boolean	Set to true to create a relationship for restore. To trigger restore-transfer, use transfers POST on the restore relationship. SnapMirror relationships with the policy type "async" can be restored. SnapMirror relationships with the policy type "sync" cannot be restored.
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	snapmirror_endpoint	Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint. • Introduced in: 9.6
source	snapmirror_endpoint	

Name	Type	Description
state	string	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

Name	Type	Description
throttle	integer	Throttle, in KBs per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited.
transfer	transfer	Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.
transfer_schedule	transfer_schedule	Schedule used to update asynchronous relationships. This "transfer_schedule" overrides the "transfer_schedule" set on the SnapMirror relationship's policy. To remove the "transfer_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror.
unhealthy_reason	array[snapmirror_error]	Reason the relationship is not healthy. It is a concatenation of up to four levels of error messages.
uuid	string	

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": {
      "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"
      },
      "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
    },
    "exported_snapshot": "string",
    "fabriclink": {
      "destination_role": "active_mirror",
      "source_role": "active_mirror"
    },
    "group_type": "consistency_group",
    "identity_preservation": "full",
    "lag_time": "PT8H35M42S",
    "last_transfer_type": "initialize",
    "policy": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "Asynchronous",
      "transfer_schedule": {
        "name": "hourly"
      },
      "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": {
        "name": "volume1",
        "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
      },
      "ipspace": "Default",
      "path": "svm1:volume1",
```

```json
      "svm": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "svm1",
        "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
      },
      "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
    },
    "state": "snapmirrored",
    "throttle": 0,
    "transfer": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "end_time": "2020-12-02T18:36:19-08:00",
      "state": "aborted",
      "total_duration": "PT28M41S",
      "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
    },
    "transfer_schedule": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "weekly",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "unhealthy_reason": [
      {
        "code": "6621444",
        "message": "Failed to complete update operation on one or more
item relationships.",
        "parameters": []
      },
      {
        "code": "6621445",
        "message": "Group Update failed",
        "parameters": []
      }
    ],
```

```
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
  }
```

Error

```
Status: Default
```

ONTAP Error Response codes

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

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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	error	
status	status	

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	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. • enum: ["extreme", "performance", "value"] • Introduced in: 9.6

tiering

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

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

snapmirror_destination_creation

Use this object to provision the destination endpoint when establishing a SnapMirror relationship for a FlexVol volume, FlexGroup volume, SVM, Consistency Group or ONTAP S3 Bucket. Given a source endpoint, the destination endpoint is provisioned in the SVM specified in the "destination.path" property. While protecting an SVM, the SVM destination endpoint can only be provisioned on the local cluster. To provision the SVM destination endpoint use the optional "source.cluster.name" property to specify the remote cluster name or use the optional "source.cluster.uuid" property to specify the remote cluster UUID. When "create_destination.enabled" option is specified while making a POST for a SnapMirror relationship, the relationship can be automatically initialized by setting the "state" either to "snapmirrored" when the policy is of type "async" or to "in_sync" when the policy is of type "sync". The "destination.path" property must specify the destination endpoint path. For example, for FlexVol volume and FlexGroup volume, the "destination.path" can be specified as <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, Consistency Group or a Bucket destination endpoint, the properties in this object can be specified either from the source or the destination cluster. For an SVM destination endpoint, the properties in this object can be specified from the destination cluster only. This object is not supported for non ONTAP endpoints. While protecting a S3 Bucket, the optional "size" property can be used to create ONTAP S3 Bucket destination endpoint of the specified size.</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.
size	integer	Optional property to specify the size of destination endpoint in bytes. This property is applicable only to ONTAP S3 Bucket endpoints. The minimum size for S3 bucket is 80MB and maximum size is 64TB. If not specified, system will create destination with default size of 800GB. • Introduced in: 9.10
storage_service	storage_service	
tiering	tiering	

cluster

Name	Type	Description
_links	_links	
name	string	
uuid	string	

consistency_group_volumes

Name	Type	Description
name	string	The name of the volume.
uuid	string	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. • example: 028baa66-41bd-11e9-81d5-00a0986138f7 • x-ntap-createOnly: true • Introduced in: 9.8

svm

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

snapmirror_endpoint

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

Name	Type	Description
cluster	cluster	
consistency_group_volumes	array[consistency_group_volumes]	Mandatory property for a Consistency Group endpoint. Specifies the list of FlexVol volumes for a Consistency Group.
ipspace	string	Optional property to specify the IPSpace of the SVM.
path	string	ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name ONTAP S3 - svm1:/bucket/bucket1 NON-ONTAP - objstore1:/objstore • example: svm1:volume1 • Introduced in: 9.6
svm	svm	
uuid	string	UUID of the endpoint. Applicable for object store SnapMirror relationships only.

fabriclink

SnapMirror FabricLink relationship

Name	Type	Description
destination_bucket	string	Specifies the name of the destination bucket of the FabricLink SnapMirror relationship. This parameter is supported for FabricLink SnapMirror relationships with ONTAP or external object store vendor endpoints. This parameter is only available on ONTAP 9.10.1 or later. • readOnly: 1
destination_role	string	This field represents the destination bucket's role within FabricLink. Possible values for this field are "active_mirror" and "backup". In a bi-directional ONTAP to ONTAP mirroring relationship, both FabricLink SnapMirror records will contain "active_mirror" in both the source-role and destination-role fields. This field conveys that both mirrors are capable of performing an active role at any time. In an ONTAP to Non-ONTAP FabricLink replication relationship, the source-role is "active_mirror" and the destination-role is "backup". This parameter is supported for FabricLink SnapMirror relationships with ONTAP or external object store vendor endpoints. This parameter is only available in ONTAP 9.10.1 or later. • enum: ["active_mirror", "backup"] • readOnly: 1

Name	Type	Description
pending_work_task_count	integer	Conveys how many discrete tasks the bucket in this unidirectional FabricLink SnapMirror relationship knows it will eventually have to perform to fully update the destination to reflect the source's state. Each work item might represent the need to push a newly created object from the source to the destination mirror, or perhaps to advise the destination mirror that an existing object has been modified or deleted. If the link is configured with an RPO of one hour and the user creates an object on the source bucket, that bucket will immediately show that it now has a pending work item to send that new object to the destination; FabricLink intentionally will not act on that work item until the hour has elapsed. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1
pull_byte_count	integer	Represents the approximate number of bytes of data that the destination bucket anticipates reading from the source bucket for the FabricLink SnapMirror relationship. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1
push_byte_count	integer	Represents the approximate number of bytes of data that must be pushed from the source bucket to the destination bucket for the FabricLink SnapMirror relationship. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1

Name	Type	Description
source_bucket	string	Specifies the name of the source bucket of the FabricLink SnapMirror relationship. This parameter is supported only for FabricLink SnapMirror relationships with ONTAP endpoints. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1
source_role	string	This field represents the source bucket's role within FabricLink. One possible value for this field is "active_mirror". In a bi-directional ONTAP to ONTAP mirroring relationship, both FabricLink SnapMirror records will contain "active_mirror" in both the source-role and destination-role fields. This field conveys that both mirrors are capable of performing an active role at any time. In an ONTAP to Non-ONTAP FabricLink replication relationship, the source-role is "active_mirror" and the destination-role is "backup". This parameter is supported only for FabricLink SnapMirror relationships with an ONTAP endpoint. This parameter is only available in ONTAP 9.10.1 or later. • enum: ["active_mirror"] • readOnly: 1
status	string	Specifies the status of the FabricLink SnapMirror relationship. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1

Name	Type	Description
topology_uuid	string	Topology UUID is used to refer to a unique data set. All the links that connect the same data set share the same topology UUID. Thus, in a fan-out topology, all links fanning from the same source bucket will carry the same topology-uuid value. Similarly, in a cascading mirror arrangement, all links in the cascade will carry the same topology-uuid value. This parameter is only available in ONTAP 9.10.1 or later. • format: uuid • readOnly: 1

transfer_schedule

The schedule used to update the asynchronous relationship. This is only applicable when provisioning a destination endpoint on a remote cluster with an effective cluster version of 9.6 or earlier. Only cron schedules are supported for SnapMirror.

Name	Type	Description
name	string	

policy

Basic policy information of the relationship.

Name	Type	Description
_links	_links	
name	string	
transfer_schedule	transfer_schedule	The schedule used to update the asynchronous relationship. This is only applicable when provisioning a destination endpoint on a remote cluster with an effective cluster version of 9.6 or earlier. Only cron schedules are supported for SnapMirror.
type	string	
uuid	string	

transfer

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

Name	Type	Description
_links	_links	
bytes_transferred	integer	Bytes transferred.
end_time	string	End time of the last transfer.
state	string	
total_duration	string	Transfer elapsed time.
uuid	string	Transfer UUID. This property is applicable only for active transfers.

transfer_schedule

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

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

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". For relationships with a policy of type "async" and relationship state as "snapmirrored", stop IO on the source endpoint and perform a SnapMirror transfer POST operation before reversing the direction of the relationship to prevent any loss of data.

The values "in_sync", "out_of_sync", and "synchronizing" are only applicable to relationships with a policy of type "sync".

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

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

The policy cannot be changed if the policy has an "archive.after_days" value of "0" and the new policy does not have an "archive.after_days" value of "0".

Examples

Related ONTAP commands

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

Important notes

- The property "transfer_schedule" if set on a SnapMirror relationship overrides the "transfer_schedule" set on the policy being used with the SnapMirror relationship.
- The property "throttle" if set on a SnapMirror relationship overrides the "throttle" set on the policy being used with the SnapMirror relationship.
- The properties "transfer_schedule" and "throttle" are not supported when "failback" is set to "true".
- The properties "transfer_schedule" and "throttle" are not supported when "failover" is set to "true".
- The properties "transfer_schedule" and "throttle" are not supported when "force_failover" is set to "true".
- The properties "transfer_schedule" and "throttle" are not supported when the direction of the relationship is being reversed.
- To remove a transfer_schedule on a SnapMirror relationship set the "transfer_schedule" to null (no-quotes) during SnapMirror relationship PATCH.
- The property "identity_preservation" value can be changed from a higher "identity_preservation" threshold value to a lower "identity_preservation" threshold value but not vice-versa. For example, the threshold value of the "identity_preservation" property can be changed from "full" to "exclude_network_config", but cannot be increased from "exclude_network_and_protocol_config" to "exclude_network_config" to "full". The threshold value of the "identity_preservation" cannot be changed to "exclude_network_and_protocol_config" for IDP SVMDR.
- Policies with property "network_compression_enabled" to "true" cannot be used in the creation of SnapMirror relationships having object store endpoints.
- Policies with property "retention.creation_schedule" cannot be used in the creation of SnapMirror relationships having object store endpoints.
- Policies with the property "snapshot_lock_mode" set to "compliance" or "enterprise" can only be used for SnapMirror relationships having a source endpoint as a FlexVol volume and a destination endpoint as an object store.

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"}'
```

Updating SnapMirror transfer_schedule and throttle for an asynchronous SnapMirror relationship. Transfer_schedule can be specified as UUID or name or both.

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

Removing the SnapMirror transfer_schedule for an asynchronous SnapMirror relationship.

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

Removing the SnapMirror throttle for an asynchronous SnapMirror relationship.

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

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

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

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

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

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

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

Learn more

- DOC /snapmirror/relationships

Parameters

Name	Type	In	Required	Description
uuid	string	path	True	Relationship UUID

Name	Type	In	Required	Description
failover	boolean	query	False	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. • Introduced in: 9.7 • Default value:
force-failover	boolean	query	False	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. • Introduced in: 9.7 • Default value:

Name	Type	In	Required	Description
force	boolean	query	False	If this parameter is set while specifying the state as "broken_off", indicates a forced failover overriding the validation errors. • Introduced in: 9.8 • Default value:
failback	boolean	query	False	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. • Introduced in: 9.8 • Default value:

Name	Type	In	Required	Description
return_timeout	integer	query	False	The number of seconds to allow the call to execute before returning. When doing a POST, PATCH, or DELETE operation on a single record, the default is 0 seconds. This means that if an asynchronous operation is started, the server immediately returns HTTP code 202 (Accepted) along with a link to the job. If a non-zero value is specified for POST, PATCH, or DELETE operations, ONTAP waits that length of time to see if the job completes so it can return something other than 202. • Default value: 1 • Max value: 120 • Min value: 0
validate_only	boolean	query	False	Validate the operation and its parameters, without actually performing the operation. • Introduced in: 9.7

Request Body

Name	Type	Description
_links	_links	

Name	Type	Description
consistency_group_failover	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. • Introduced in: 9.8 • readOnly: 1

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

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

Name	Type	Description
identity_preservation	string	Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type. This "identity_preservation" overrides the "identity_preservation" set on the SnapMirror relationship's policy.
lag_time	string	Time since the exported Snapshot copy was created.
last_transfer_type	string	Specifies the operation type of the last transfer that occured on the relationship. The *initialize* transfer occurs when the relationship state changes from uninitialized to snapmirrored or in_sync. The *update* transfer occurs when the snapshots are transferred from the source endpoint to the destination endpoint as part of scheduled or manual update. The *resync* transfer occurs when the relationship state changes from broken_off to snapmirrored or in_sync. The *restore* transfer occurs when the snapshot is restored from a destination endpoint to another endpoint.
policy	policy	Basic policy information of the relationship.
preserve	boolean	Set to true on resync to preserve Snapshot copies on the destination that are newer than the latest common Snapshot copy. This 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 SVMDR relationships when the PATCH state is being changed to "snapmirrored".
recover_after_break	boolean	Set to true to recover from a failed SnapMirror break operation on a FlexGroup volume relationship. This restores all destination FlexGroup constituent volumes to the latest Snapshot copy, and any writes to the read-write constituents are lost. This property is applicable only for SnapMirror relationships with FlexGroup volume endpoints and when the PATCH state is being changed to "broken_off".
restore	boolean	Set to true to create a relationship for restore. To trigger restore-transfer, use transfers POST on the restore relationship. SnapMirror relationships with the policy type "async" can be restored. SnapMirror relationships with the policy type "sync" cannot be restored.
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	snapmirror_endpoint	Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint. • Introduced in: 9.6

Name	Type	Description
state	string	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

Name	Type	Description
throttle	integer	Throttle, in KBs per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited.
transfer	transfer	Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.
transfer_schedule	transfer_schedule	Schedule used to update asynchronous relationships. This "transfer_schedule" overrides the "transfer_schedule" set on the SnapMirror relationship's policy. To remove the "transfer_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror.
unhealthy_reason	array[snapmirror_error]	Reason the relationship is not healthy. It is a concatenation of up to four levels of error messages.
uuid	string	

Example request

```json
{
  "_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": {
      "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"
    },
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
  },
  "exported_snapshot": "string",
  "fabriclink": {
    "destination_role": "active_mirror",
    "source_role": "active_mirror"
  },
  "group_type": "consistency_group",
  "identity_preservation": "full",
  "lag_time": "PT8H35M42S",
  "last_transfer_type": "initialize",
  "policy": {
    "_links": {
      "self": {
        "href": "/api/resourcelink"
      }
    },
    "name": "Asynchronous",
    "transfer_schedule": {
      "name": "hourly"
    },
    "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": {
      "name": "volume1",
      "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7"
    },
    "ipspace": "Default",
    "path": "svm1:volume1",
```

```json
      "svm": {
        "_links": {
          "self": {
            "href": "/api/resourcelink"
          }
        },
        "name": "svm1",
        "uuid": "02c9e252-41be-11e9-81d5-00a0986138f7"
      },
      "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
    },
    "state": "snapmirrored",
    "throttle": 0,
    "transfer": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "end_time": "2020-12-02T18:36:19-08:00",
      "state": "aborted",
      "total_duration": "PT28M41S",
      "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
    },
    "transfer_schedule": {
      "_links": {
        "self": {
          "href": "/api/resourcelink"
        }
      },
      "name": "weekly",
      "uuid": "1cd8a442-86d1-11e0-ae1c-123478563412"
    },
    "unhealthy_reason": [
      {
        "code": "6621444",
        "message": "Failed to complete update operation on one or more
item relationships.",
        "parameters": []
      },
      {
        "code": "6621445",
        "message": "Group Update failed",
        "parameters": []
      }
    ],
```

```
    "uuid": "4ea7a442-86d1-11e0-ae1c-123478563412"
  }
}
```

Response

```
Status: 202, Accepted
```

Name	Type	Description
job	job_link	

Example response

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

Error

```
Status: Default
```

ONTAP Error Response codes

Error code	Description
13303825	Could not retrieve information for the SnapMirror policy type
13303817	Unknown value for the SnapMirror state
13303829	Invalid state
13303830	Transient state
13303831	Invalid state for async SnapMirror relationship
13303834	Given input valid only for FlexGroup SnapMirror relationship

Error code	Description
13303835	Given flag is valid only when PATCH state is broken_off
13303836	Given flag is valid only when PATCH state is snapmirrored or in_sync
13303818	Invalid state transition requested
13303828	Given state change is not possible for SVM SnapMirror relationship
13303833	Requested state change is not possible
13303832	SnapMirror relationship is already initialized
13303824	Quiescing the SnapMirror relationship has failed
13303826	Required environment variables are not set
13303827	Internal Error
13303823	Quiesce operation timed out
13303821	Invalid SnapMirror policy name/UUID
13303819	Could not retrieve SnapMirror policy information
13303851	Cannot modify attributes of SnapMirror restore relationship
13303816	Could not retrieve state or status values
13303837	Given flags are valid only if SnapMirror state change is requested
6619546	Destination must be a dp volume
13303808	Transition to broken_off state failed
13303809	Transition to paused state failed
13303810	Transition to snapmirrored state failed
13303811	Transition from paused state failed
13303820	SnapMirror policy, transfer_schedule, and throttle, if specified were successfully updated, state transition failed
13303856	SVM is not configured with any data protocol
13303857	SVM is not configured with any network interface
13303858	Internal error. Failed to check LIF and protocols details for SVM
13303859	Internal error. SVM Failover operation failed. SVM operational state is unavailable.
13303865	Modifying the specified SnapMirror policy is not supported.

Error code	Description
13303866	Cannot use the specified policy to modify the policy of the relationship.
13303867	Modifying the policy of an async-mirror or a vault relationship is not supported.
13303884	LIF and protocols details are configured incorrectly for SVM.
13303996	The source and destination clusters both have a policy with the same name, but they have different properties.
13304062	Cannot reverse the direction of a SnapMirror DP relationship when the source cluster version is earlier than the destination cluster version.
13304070	Remote peer cluster requires the dp_rest_support capability to support reversing the direction of a DP relationship.
13304071	Failed to access capabilities on remote cluster.
13304080	Specified uuid and name do not match.
13304081	Modifying a property during the opertaion is not supported.
13304082	The specified properties are mutually exclusive.
13304083	The specified property is not supported because all nodes in the cluster are not capable of supporting the property.
13304086	Reversing the direction of a SnapMirror relationship associated with a policy containing the property create_snapshot_on_source set to false is not supported.
6619715	Modification of relationship is in progress. Retry the command after a few minutes.
6619699	Schedule not found.
13304108	Schedule not found in the Administrative SVM or the SVM for the relationship.
13304026	API license token is required for this operation.
13304027	Invalid API license token specified.
6621103	Modifying the policy is not supported because the original policy has an archive.after_days value of zero, and the new policy has a non-zero archive.after_days value.
13303951	This SnapMirror policy is not supported for relationships with object store destination endpoints.

Name	Type	Description
error	error	

Example error

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

Definitions

See Definitions

href

Name	Type	Description
href	string	

_links

Name	Type	Description
self	href	

error_arguments

Name	Type	Description
code	string	Argument code
message	string	Message argument

error

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

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	error	
status	status	

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	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. • enum: ["extreme", "performance", "value"] • Introduced in: 9.6

tiering

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

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

snapmirror_destination_creation

Use this object to provision the destination endpoint when establishing a SnapMirror relationship for a FlexVol volume, FlexGroup volume, SVM, Consistency Group or ONTAP S3 Bucket. Given a source endpoint, the destination endpoint is provisioned in the SVM specified in the "destination.path" property. While protecting an SVM, the SVM destination endpoint can only be provisioned on the local cluster. To provision the SVM destination endpoint use the optional "source.cluster.name" property to specify the remote cluster name or use the optional "source.cluster.uuid" property to specify the remote cluster UUID. When "create_destination.enabled" option is specified while making a POST for a SnapMirror relationship, the relationship can be automatically initialized by setting the "state" either to "snapmirrored" when the policy is of type "async" or to "in_sync" when the policy is of type "sync". The "destination.path" property must specify the destination endpoint path. For example, for FlexVol volume and FlexGroup volume, the "destination.path" can be specified as <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, Consistency Group or a Bucket destination endpoint, the properties in this object can be specified either from the source or the destination cluster. For an SVM destination endpoint, the properties in this object can be specified from the destination cluster only. This object is not supported for non ONTAP endpoints. While protecting a S3 Bucket, the optional "size" property can be used to create ONTAP S3 Bucket destination endpoint of the specified size.</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.
size	integer	Optional property to specify the size of destination endpoint in bytes. This property is applicable only to ONTAP S3 Bucket endpoints. The minimum size for S3 bucket is 80MB and maximum size is 64TB. If not specified, system will create destination with default size of 800GB. • Introduced in: 9.10
storage_service	storage_service	
tiering	tiering	

cluster

Name	Type	Description
_links	_links	
name	string	
uuid	string	

consistency_group_volumes

Name	Type	Description
name	string	The name of the volume.
uuid	string	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. • example: 028baa66-41bd-11e9-81d5-00a0986138f7 • x-ntap-createOnly: true • Introduced in: 9.8

svm

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

snapmirror_endpoint

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

Name	Type	Description
cluster	cluster	
consistency_group_volumes	array[consistency_group_volumes]	Mandatory property for a Consistency Group endpoint. Specifies the list of FlexVol volumes for a Consistency Group.
ipspace	string	Optional property to specify the IPSpace of the SVM.
path	string	ONTAP FlexVol/FlexGroup - svm1:volume1 ONTAP SVM - svm1: ONTAP Consistency Group - svm1:/cg/cg_name ONTAP S3 - svm1:/bucket/bucket1 NON-ONTAP - objstore1:/objstore • example: svm1:volume1 • Introduced in: 9.6
svm	svm	
uuid	string	UUID of the endpoint. Applicable for object store SnapMirror relationships only.

fabriclink

SnapMirror FabricLink relationship

Name	Type	Description
destination_bucket	string	Specifies the name of the destination bucket of the FabricLink SnapMirror relationship. This parameter is supported for FabricLink SnapMirror relationships with ONTAP or external object store vendor endpoints. This parameter is only available on ONTAP 9.10.1 or later. • readOnly: 1
destination_role	string	This field represents the destination bucket's role within FabricLink. Possible values for this field are "active_mirror" and "backup". In a bi-directional ONTAP to ONTAP mirroring relationship, both FabricLink SnapMirror records will contain "active_mirror" in both the source-role and destination-role fields. This field conveys that both mirrors are capable of performing an active role at any time. In an ONTAP to Non-ONTAP FabricLink replication relationship, the source-role is "active_mirror" and the destination-role is "backup". This parameter is supported for FabricLink SnapMirror relationships with ONTAP or external object store vendor endpoints. This parameter is only available in ONTAP 9.10.1 or later. • enum: ["active_mirror", "backup"] • readOnly: 1

Name	Type	Description
pending_work_task_count	integer	Conveys how many discrete tasks the bucket in this unidirectional FabricLink SnapMirror relationship knows it will eventually have to perform to fully update the destination to reflect the source's state. Each work item might represent the need to push a newly created object from the source to the destination mirror, or perhaps to advise the destination mirror that an existing object has been modified or deleted. If the link is configured with an RPO of one hour and the user creates an object on the source bucket, that bucket will immediately show that it now has a pending work item to send that new object to the destination; FabricLink intentionally will not act on that work item until the hour has elapsed. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1
pull_byte_count	integer	Represents the approximate number of bytes of data that the destination bucket anticipates reading from the source bucket for the FabricLink SnapMirror relationship. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1
push_byte_count	integer	Represents the approximate number of bytes of data that must be pushed from the source bucket to the destination bucket for the FabricLink SnapMirror relationship. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1

Name	Type	Description
source_bucket	string	Specifies the name of the source bucket of the FabricLink SnapMirror relationship. This parameter is supported only for FabricLink SnapMirror relationships with ONTAP endpoints. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1
source_role	string	This field represents the source bucket's role within FabricLink. One possible value for this field is "active_mirror". In a bi-directional ONTAP to ONTAP mirroring relationship, both FabricLink SnapMirror records will contain "active_mirror" in both the source-role and destination-role fields. This field conveys that both mirrors are capable of performing an active role at any time. In an ONTAP to Non-ONTAP FabricLink replication relationship, the source-role is "active_mirror" and the destination-role is "backup". This parameter is supported only for FabricLink SnapMirror relationships with an ONTAP endpoint. This parameter is only available in ONTAP 9.10.1 or later. • enum: ["active_mirror"] • readOnly: 1
status	string	Specifies the status of the FabricLink SnapMirror relationship. This parameter is only available in ONTAP 9.10.1 or later. • readOnly: 1

Name	Type	Description
topology_uuid	string	Topology UUID is used to refer to a unique data set. All the links that connect the same data set share the same topology UUID. Thus, in a fan-out topology, all links fanning from the same source bucket will carry the same topology-uuid value. Similarly, in a cascading mirror arrangement, all links in the cascade will carry the same topology-uuid value. This parameter is only available in ONTAP 9.10.1 or later. • format: uuid • readOnly: 1

transfer_schedule

The schedule used to update the asynchronous relationship. This is only applicable when provisioning a destination endpoint on a remote cluster with an effective cluster version of 9.6 or earlier. Only cron schedules are supported for SnapMirror.

Name	Type	Description
name	string	

policy

Basic policy information of the relationship.

Name	Type	Description
_links	_links	
name	string	
transfer_schedule	transfer_schedule	The schedule used to update the asynchronous relationship. This is only applicable when provisioning a destination endpoint on a remote cluster with an effective cluster version of 9.6 or earlier. Only cron schedules are supported for SnapMirror.
type	string	
uuid	string	

transfer

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

Name	Type	Description
_links	_links	
bytes_transferred	integer	Bytes transferred.
end_time	string	End time of the last transfer.
state	string	
total_duration	string	Transfer elapsed time.
uuid	string	Transfer UUID. This property is applicable only for active transfers.

transfer_schedule

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

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

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 relatiosnhip 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	_links	
consistency_group_failover	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. • Introduced in: 9.8 • readOnly: 1

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

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

Name	Type	Description
identity_preservation	string	Specifies which configuration of the source SVM is replicated to the destination SVM. This property is applicable only for SVM data protection with "async" policy type. This "identity_preservation" overrides the "identity_preservation" set on the SnapMirror relationship's policy.
lag_time	string	Time since the exported Snapshot copy was created.
last_transfer_type	string	Specifies the operation type of the last transfer that occured on the relationship. The *initialize* transfer occurs when the relationship state changes from uninitialized to snapmirrored or in_sync. The *update* transfer occurs when the snapshots are transferred from the source endpoint to the destination endpoint as part of scheduled or manual update. The *resync* transfer occurs when the relationship state changes from broken_off to snapmirrored or in_sync. The *restore* transfer occurs when the snapshot is restored from a destination endpoint to another endpoint.
policy	policy	Basic policy information of the relationship.
preserve	boolean	Set to true on resync to preserve Snapshot copies on the destination that are newer than the latest common Snapshot copy. This 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 SVMDR relationships when the PATCH state is being changed to "snapmirrored".
recover_after_break	boolean	Set to true to recover from a failed SnapMirror break operation on a FlexGroup volume relationship. This restores all destination FlexGroup constituent volumes to the latest Snapshot copy, and any writes to the read-write constituents are lost. This property is applicable only for SnapMirror relationships with FlexGroup volume endpoints and when the PATCH state is being changed to "broken_off".
restore	boolean	Set to true to create a relationship for restore. To trigger restore-transfer, use transfers POST on the restore relationship. SnapMirror relationships with the policy type "async" can be restored. SnapMirror relationships with the policy type "sync" cannot be restored.
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	snapmirror_endpoint	Endpoint of a SnapMirror relationship. For a GET request, the property "cluster" is populated when the endpoint is on a remote cluster. A POST request to create the destination SVM endpoint or to establish an SVM DR relationship must have the property "cluster" populated with the remote cluster details. A POST request to create the destination FlexVol volume, FlexGroup volume, Consistency Group, ONTAP S3 bucket and NON-ONTAP object-store endpoints can optionally specify the "cluster" property when the source SVM and the destination SVM are peered. A POST request to establish a SnapMirror relationship between the source endpoint and destination endpoint and when the source SVM and the destination SVM are not peered, must specify the "cluster" property for the remote endpoint. • Introduced in: 9.6

Name	Type	Description
state	string	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

Name	Type	Description
throttle	integer	Throttle, in KBs per second. This "throttle" overrides the "throttle" set on the SnapMirror relationship's policy. If neither of these are set, defaults to 0, which is interpreted as unlimited.
transfer	transfer	Basic information on the current transfer or the last transfer if there is no active transfer at the time of the request.
transfer_schedule	transfer_schedule	Schedule used to update asynchronous relationships. This "transfer_schedule" overrides the "transfer_schedule" set on the SnapMirror relationship's policy. To remove the "transfer_schedule", set its value to null (no-quotes). Only cron schedules are supported for SnapMirror.
unhealthy_reason	array[snapmirror_error]	Reason the relationship is not healthy. It is a concatenation of up to four levels of error messages.
uuid	string	

job_link

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