



SnapMirror replication

Cloud Manager Automation

NetApp
October 07, 2021

Table of Contents

- SnapMirror replication 1
 - Create a SnapMirror replication relationship 1
 - Update a SnapMirror replication relationship 3
 - Delete a SnapMirror replication relationship 4
 - Get the SnapMirror relationships 4
 - Get the status of the replication relationships 5
 - Get status of the replication relationships for a working environment 8
 - Get the intercluster LIFs 10
 - Get the replication schedules 12

SnapMirror replication

Create a SnapMirror replication relationship

You can use this workflow to create a new SnapMirror replication relationship to an ONTAP working environment. You can replicate data between working environments by choosing a one-time data replication for data transfer, or a recurring schedule for disaster recovery or long-term retention.



This workflow varies slightly depending on the cloud provider you are using.

1. Select the working environment

Based on the cloud provider, you need to perform a workflow to get the identifier for the working environment as shown in the following table.

Provider	Workflow
AWS	Perform the workflow Get working environments and choose the <code>publicId</code> and <code>svmName</code> values for the source and destination.

2. Select the LIFs

Perform the workflow [Get intercluster LIFs](#) and choose the `address` value for the source and destination.

3. Select the SnapMirror policy

Perform the workflow [Get SnapMirror policies](#) and choose the `name` value for the required schedule.

4. Select the SnapMirror schedule

Perform the workflow [Get schedules](#) and choose the `name` value for the required policy.

5. Select the volume names and related storage parameters

Based on the cloud provider, you need to perform a workflow to get the volume information.

Provider	Workflow
AWS	Perform the workflow Get volumes and choose the <code>name</code> and <code>svmName</code> and <code>aggregateName</code> values.

6. Create a quote

Based on the cloud provider, you need to perform a workflow to get the volume information.

Provider	Workflow
AWS	Perform the workflow Create quote and choose the <code>providerVolumeType</code> and <code>name</code> values.

7. Create the relationship

HTTP method	Path
POST	/occm/api/replication/vsa

curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/replication/vsa' --header
'Content-Type: application/json' --header 'x-agent-id: <AGENT_ID>'
--header 'Authorization: Bearer <ACCESS_TOKEN>' --d @JSONinput
```

Input

The JSON input example includes the minimum list of input parameters. The `maxTransferRate` is maximum transfer rate limit in KB/s. Specify 0 to indicate no limit or an integer between 1024 and 2,147,482,624.

JSON input example

```
{
  "replicationRequest": {
    "sourceWorkingEnvironmentId": "vsaworkingenvironment-sfrf3wvj",
    "destinationWorkingEnvironmentId": "vsaworkingenvironment-2qkd75xv",
    "sourceInterclusterLifIps": [
      "10.138.0.147"
    ],
    "destinationInterclusterLifIps": [
      "10.138.0.154"
    ],
    "policyName": "MirrorAllSnapshots",
    "scheduleName": "daily",
    "maxTransferRate": 102400
  },
  "replicationVolume": {
    "sourceSvmName": "svm_zivgcp01we02",
    "sourceVolumeName": "zivagg01vol01",
    "destinationVolumeName": "zivagg01vol03_copy",
    "destinationAggregateName": "aggr1",
    "numOfDisksApprovedToAdd": 0,
    "advancedMode": false,
    "destinationProviderVolumeType": "pd-ssd",
    "destinationSvmName": "svm_zivgcp01we03"
  }
}
```

Output

None

Update a SnapMirror replication relationship

You can use this workflow to update an existing SnapMirror replication relationship.

1. Select the working environment and related values

Perform the workflow [Get relationships status](#) and choose the `workingEnvironment` and `svmName` and `volumeName` values for the path parameters. All values are for the destination.

2. Update the relationship

HTTP method	Path
PUT	<code>/occm/api/replication/{workingEnvironmentId}/{destinationSvmName}/{destinationVolumeName}</code>

curl example

```
curl --location --request PUT
'https://cloudmanager.cloud.netapp.com/occm/api/replication/<WORKING_ENV_ID/<SVM_NAME/<VOLUME_NAME>' --header 'Content-Type: application/json'
--header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer
<ACCESS_TOKEN>' --d @JSONinput
```

Input

Path parameters:

- `<WORKING_ENV_ID>` (`workingEnvironmentId`)
- `<SVM_NAME>` (`destinationSvmName`)
- `<VOLUME_NAME>` (`destinationVolumeName`)

The JSON input example includes some of the parameters you can update.

JSON input example

```
{
  "maxTransferRate": 0
}
```

Output

None

Delete a SnapMirror replication relationship

You can use this workflow to delete an existing SnapMirror replication relationship.

1. Select the working environment and related values

Perform the workflow [Get relationships status](#). Choose the `workingEnvironmentId` and `svmName` and `volumeName` values for the path parameters. All values are for the destination.

2. Delete the relationship

HTTP method	Path
DELETE	<code>/occm/api/replication/{destinationWorkingEnvironmentId}/{destinationSvmName}/{destinationVolumeName}</code>

curl example

```
curl --location --request DELETE
'https://cloudmanager.cloud.netapp.com/occm/api/replication/<WORKING_ENV_ID>/<SVM_NAME>/<VOLUME_NAME>' --header 'Content-Type: application/json'
--header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
```

Input

There are three path parameters, all of which apply to the destination:

- `<WORKING_ENV_ID>` (`destinationWorkingEnvironmentId`)
- `<SVM_NAME>` (`destinationSvmName`)
- `<VOLUME_NAME>` (`destinationVolumeName`)

Output

None

Get the SnapMirror relationships

You can retrieve all the SnapMirror relationship pairs.

1. Retrieve the relationships

HTTP method	Path
GET	<code>/occm/api/replication/all-relationships</code>

curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/replication/all-
relationships' --header 'Content-Type: application/json' --header 'x-
agent-id: <AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
```

Input parameters

None

Output

An array of relationship pairs is returned as shown in the JSON output example.

JSON output example

```
{
  "relationships": [
    {
      "source": {
        "id": "vsaworkingenvironment-sfrf3wvj"
      },
      "target": {
        "id": "vsaworkingenvironment-2qkd75xv"
      }
    }
  ]
}
```

Get the status of the replication relationships

You can use this workflow to retrieve the status of all the SnapMirror replication relationships.



This workflow varies slightly depending on the cloud provider you are using.

1. Optionally select the tenant ID

Based on the cloud provider, you need to perform a workflow to get the tenant ID as shown in the following table.

Provider	Workflow
AWS	Perform the workflow Get working environments and choose the <code>tenantId</code> value.

2. Get the status of the relationships

HTTP method	Path
GET	/occm/api/replication/status

curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/replication/status'
--header 'Content-Type: application/json' --header 'x-agent-id:
<AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
```

Input

The query parameter `tenantId` is optional.

Output

The JSON output example includes the list of all the SnapMirror relationships with status.

JSON output example

```
[
  {
    "source": {
      "workingEnvironmentId": "vsaworkingenvironment-sfrf3wvj",
      "workingEnvironmentType": "VSA",
      "workingEnvironmentStatus": "ON",
      "clusterName": "zivgcp01we02",
      "region": "us-west1-b",
      "availabilityZone": null,
      "svmName": "svm_zivgcp01we02",
      "nodeName": null,
      "volumeName": "zivagg01vol01"
    },
    "destination": {
      "workingEnvironmentId": "vsaworkingenvironment-2qkd75xv",
      "workingEnvironmentType": "VSA",
      "workingEnvironmentStatus": "ON",
      "clusterName": "zivgcp01we03",
      "region": "us-west1-b",
      "availabilityZone": null,
      "svmName": "svm_zivgcp01we03",
      "nodeName": "zivgcp01we03-01",
      "volumeName": "zivagg01vol01_copy"
    },
    "mirrorState": "snapmirrored",
    "relationshipType": "extended_data_protection",
    "relationshipStatus": "idle",
    "relationshipProgress": null,
    "policy": "MirrorAllSnapshots",
    "policyType": "async_mirror",
    "schedule": "daily",
    "maxTransferRate": {
```



```
    "size": 102400.0,  
    "unit": "KB"  
  },  
  "networkCompressionRatio": "1:1",  
  "healthy": true,  
  "unhealthyReason": null,  
  "lagTime": {  
    "length": 14012,  
    "unit": "SECONDS"  
  },  
  "newestSnapshotName": "snapmirror.e7179420-5e45-11eb-8f27-  
d7fea0402bd2_2150573386.2021-01-25_123451",  
  "newestSnapshotCreated": 1611578092,  
  "lastTransferInfo": {  
    "transferType": "update",  
    "transferSize": {  
      "size": 6240.0,  
      "unit": "Byte"  
    },  
    "transferDuration": {  
      "length": 4,  
      "unit": "SECONDS"  
    },  
    "transferEnded": 1611578097,  
    "transferError": null  
  },  
  "currentTransferInfo": {  
    "transferType": null,  
    "transferPriority": null,  
    "transferError": null  
  },  
  "totalTransferTime": {  
    "length": 6,  
    "unit": "SECONDS"  
  },  
  "totalTransferSize": {  
    "size": 23792.0,  
    "unit": "Byte"  
  },  
  "volumeUsedSize": {  
    "size": 1032192.0,  
    "unit": "Byte"  
  },  
  "volumeCapacityTier": {  
    "size": 0.0,  
    "unit": "Byte"  
  }
```

```
}  
}  
]
```

Get status of the replication relationships for a working environment

You can use this workflow to retrieve the status of all the SnapMirror replication relationships for a specific working environment.



This workflow varies slightly depending on the cloud provider you are using.

1. Select the working environment

Based on the cloud provider, you need to perform a workflow to get the identifier for the working environment as shown in the following table.

Provider	Workflow
AWS	Perform the workflow Get working environments and choose the <code>publicId</code> value for the <code>workingEnvironmentId</code> query parameter.

2. Get the status of the relationships

HTTP method	Path
GET	<code>/occm/api/replication/status/{workingEnvironmentId}</code>

curl example

```
curl --location --request GET  
'https://cloudmanager.cloud.netapp.com/occm/api/replication/status/<WORKING_ENV_ID>' --header 'Content-Type: application/json' --header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
```

Input

Path parameters:

- `<WORKING_ENV_ID>` (`workingEnvironmentId`)

Output

The JSON output example includes the list of all the SnapMirror relationships with status for a specific working environment.

JSON output example

```
[  
  {
```

```
"source": {
  "workingEnvironmentId": "vsaworkingenvironment-sfrf3wvj",
  "workingEnvironmentType": "VSA",
  "workingEnvironmentStatus": "ON",
  "clusterName": "zivgcp01we02",
  "region": "us-west1-b",
  "availabilityZone": null,
  "svmName": "svm_zivgcp01we02",
  "nodeName": null,
  "volumeName": "zivagg01vol01"
},
"destination": {
  "workingEnvironmentId": "vsaworkingenvironment-2qkd75xv",
  "workingEnvironmentType": "VSA",
  "workingEnvironmentStatus": "ON",
  "clusterName": "zivgcp01we03",
  "region": "us-west1-b",
  "availabilityZone": null,
  "svmName": "svm_zivgcp01we03",
  "nodeName": "zivgcp01we03-01",
  "volumeName": "zivagg01vol01_copy"
},
"mirrorState": "snapmirrored",
"relationshipType": "extended_data_protection",
"relationshipStatus": "idle",
"relationshipProgress": null,
"policy": "MirrorAllSnapshots",
"policyType": "async_mirror",
"schedule": "daily",
"maxTransferRate": {
  "size": 102400.0,
  "unit": "KB"
},
"networkCompressionRatio": "1:1",
"healthy": true,
"unhealthyReason": null,
"lagTime": {
  "length": 14886,
  "unit": "SECONDS"
},
"newestSnapshotName": "snapmirror.e7179420-5e45-11eb-8f27-
d7fea0402bd2_2150573386.2021-01-25_123451",
"newestSnapshotCreated": 1611578092,
"lastTransferInfo": {
  "transferType": "update",
  "transferSize": {
```

```

        "size": 6240.0,
        "unit": "Byte"
    },
    "transferDuration": {
        "length": 4,
        "unit": "SECONDS"
    },
    "transferEnded": 1611578097,
    "transferError": null
},
"currentTransferInfo": {
    "transferType": null,
    "transferPriority": null,
    "transferError": null
},
"totalTransferTime": {
    "length": 6,
    "unit": "SECONDS"
},
"totalTransferSize": {
    "size": 23792.0,
    "unit": "Byte"
},
"volumeUsedSize": {
    "size": 1032192.0,
    "unit": "Byte"
},
"volumeCapacityTier": {
    "size": 0.0,
    "unit": "Byte"
}
}
]

```

Get the intercluster LIFs

You can use this workflow to retrieve the intercluster LIFs used in a cluster peering relationship.



This workflow varies slightly depending on the cloud provider you are using.

1. Select the working environment

Based on the cloud provider, you need to perform a workflow to get the identifier for the working environment as shown in the following table.

Provider	Workflow
AWS	Perform the workflow Get working environments and choose the <code>publicId</code> values for the working environment query parameters.

2. Get the intercluster LIFs

HTTP method	Path
GET	/occm/api/replication/intercluster-lifs

curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/replication/interclusterlifs?workingEnvironmentId=<WORKING_ENV_ID>&peerWorkingEnvironmentId=<WORKING_ENV_ID>' --header 'Content-Type: application/json' --header 'x-agent-id: <AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
```

Input

Query parameters:

- <WORKING_ENV_ID> (workingEnvironmentId)
- <WORKING_ENV_ID> (peerWorkingEnvironmentId)

Output

The JSON output example includes the list of LIFs.

JSON output example

```

{
  "interClusterLifs": [
    {
      "name": "intercluster",
      "address": "10.138.0.154",
      "netmaskLength": 32,
      "port": "e0a",
      "node": "zivgcp01we03-01",
      "status": "up",
      "isPeered": true
    }
  ],
  "peerInterClusterLifs": [
    {
      "name": "intercluster",
      "address": "10.138.0.147",
      "netmaskLength": 32,
      "port": "e0a",
      "node": "zivgcp01we02-01",
      "status": "up",
      "isPeered": true
    }
  ]
}

```

Get the replication schedules

You can use this workflow to retrieve the replication schedules used for a specific working environment.



This workflow varies slightly depending on the cloud provider you are using.

1. Select the working environment

Based on the cloud provider, you need to perform a workflow to get the identifier for the working environment as shown in the following table.

Provider	Workflow
AWS	Perform the workflow Get working environments and choose the <code>publicId</code> value for the working environment path parameter.

2. Get the schedules

HTTP method	Path
GET	/occm/api/replication/schedules/{workingEnvironmentId}

curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/replication/schedules/<WORKING_ENV_ID>' --header 'Content-Type: application/json' --header 'x-agent-id:<AGENT_ID>' --header 'Authorization: Bearer <ACCESS_TOKEN>'
```

Input

Path parameter <WORKING_ENV_ID> (workingEnvironment)

Output

The JSON output example includes the list of replication schedules.

JSON output example

```

[
  {
    "name": "10min",
    "description": "@:00,:10,:20,:30,:40,:50",
    "cronJobSchedule": {
      "months": [],
      "days": [],
      "weekDays": [],
      "hours": [],
      "minutes": [
        0,
        10,
        20,
        30,
        40,
        50
      ]
    }
  },
  {
    "name": "5min",
    "description": "@:00,:05,:10,:15,:20,:25,:30,:35,:40,:45,:50,:55",
    "cronJobSchedule": {
      "months": [],
      "days": [],
      "weekDays": [],
      "hours": [],
      "minutes": [
        0,
        5,
        10,
        15,
        20,
        25,
        30,
        35,
        40,
        45,
        50,
        55
      ]
    }
  }
]

```


Copyright Information

Copyright © 2021 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means-graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system-without prior written permission of the copyright owner.

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

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

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

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

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark Information

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