



Volumes

Cloud Manager Automation

NetApp
October 07, 2021

Table of Contents

- Volumes 1
 - Create volume using NFS 1
 - Create volume using CIFS 2
 - Create volume using iSCSI 4
 - Get volumes 8
 - Modify volume 11
 - Delete volume 12
 - Create quote 13
 - Get iGroups 14

Volumes

Create volume using NFS

You can use this workflow to create a volume accessed through NFS.



If the properties `aggregateName` and `maxNumOfDisksApprovedToAdd` are not provided on the REST API call, the response will fail with a suggested name for the aggregate and the number of disks needed to fulfill the request.

1. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` value for the `workingEnvironmentId` (working environment) and the `svmName` (SVM name).

2. Select the aggregate

Perform the workflow [Get aggregates](#) and choose the name for the aggregate.



If aggregate name does not exist and the `createAggregateIfNotFound` query parameter is set `true`, the create volume request is allowed if the named aggregate is not found.

3. Select the virtual private cloud

Perform the workflow [Get virtual private clouds](#) and choose the `cidrBlock` value of the required VPC for the `ips` parameter or fill in the desired `exportPolicyInfo` value manually.

4. Choose the size for the disk

Choose the size value for the `size:size` parameter. The `size:unit` must be one of the following: TB, GB, MB, KB, or Byte.

5. Create the quote

Perform the workflow [Create quote](#). This is a recommended step but is not mandatory.

6. Create the volume

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

curl example

```
curl --location --request POST
'https://staging.cloudmanager.cloud.netapp.com/occm/api/vsa/volumes'
--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, including:

- <WORKING_ENV_ID> (workingEnvironmentId)
- <SVM_NAME> (svmName)
- <AGGR_NAME> (aggregateName)

If aggregate name does not exist, you can set the `createAggregateIfNotFound` query parameter to `true` which allows the aggregate not-found condition.

JSON input example

```
{
  "workingEnvironmentId": "VsaWorkingEnvironment-5dZfyKS5",
  "svmName": "svm_ziv01we01",
  "aggregateName": "ziv01agg01",
  "name": "ziv01vol04",
  "size": {
    "size": 100,
    "unit": "GB"
  },
  "exportPolicyInfo": {
    "policyType": "custom",
    "ips": [
      "172.31.0.0/16"
    ],
    "nfsVersion": [
      "nfs3",
      "nfs4"
    ]
  },
  "snapshotPolicyName": "default",
  "enableThinProvisioning": true,
  "enableCompression": true,
  "enableDeduplication": true,
  "maxNumOfDisksApprovedToAdd": 0
}
```

Output

None

Create volume using CIFS

You can use this workflow to create a volume accessed through CIFS.



If the properties `aggregateName` and `maxNumOfDisksApprovedToAdd` are not provided on the REST API call, the response will fail with a suggested name for the aggregate and the number of disks needed to fulfill the request.

1. Choose the CIFS configuration

A CIFS server configuration must be defined for your working environment. You can do one of the following:

- If a CIFS configuration already exists, perform the workflow [Get CIFS server configurations](#) to access the configuration parameters.
- If a CIFS configuration does not exist, perform the workflow [Create CIFS server configuration](#) to create one.

2. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` value for the `workingEnvironmentId` (working environment) and the `svmName` (SVM name).

3. Select the aggregate

Perform the workflow [Get aggregates](#) and choose the name for the `aggregateName` value.



If aggregate name does not exist and the `createAggregateIfNotFound` query parameter is set `true`, the create volume request is allowed if the named aggregate is not found.

4. Choose the size for the disk

Choose the size value for the `size:size` parameter. The `size:unit` must be one of the following: TB, GB, MB, KB, or Byte.

5. Create the quote

Perform the workflow [Create quote](#). This is a recommended step but is not mandatory.

6. Create the volume

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

curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/vsa/volumes' --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, including:

- <WORKING_ENV_ID> (workingEnvironmentId)
- <SVM_NAME> (svmName)
- <AGGR_NAME> (aggregateName)

If an aggregate name does not exist, you can set the `createAggregateIfNotFound` query parameter to `true` which allows the aggregate not-found condition.

JSON input example

```
{
  "workingEnvironmentId": "VsaWorkingEnvironment-SfpVUZSc",
  "svmName": "svm_zivaws02we01",
  "aggregateName": "aggr1",
  "name": "zivaws02we02vol02Cifs",
  "size": {
    "size": 100,
    "unit": "GB"
  },
  "shareInfo": {
    "accessControl": {
      "permission": "full_control",
      "users": [
        "Everyone"
      ],
      "users": "Everyone;"
    },
    "shareName": "zivaws02we01vol02Cifs_share"
  },
  "snapshotPolicyName": "default",
  "enableThinProvisioning": true,
  "enableCompression": true,
  "enableDeduplication": true,
  "maxNumOfDisksApprovedToAdd": 0
}
```

Output

None

Create volume using iSCSI

You can use this workflow to create a volume accessed through iSCSI. There are two workflows available depending on whether a new or existing iGroup is used. You need to select the correct workflow:

- [Create volume using iSCSI with a new iGroup](#)

- [Create volume using iSCSI with an existing iGroup](#)

Create volume using iSCSI with a new iGroup



If the properties `aggregateName` and `maxNumOfDisksApprovedToAdd` are not provided on the REST API call, the response will fail with a suggested name for the aggregate and the number of disks needed to fulfill the request.

1. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` value for the `workingEnvironmentId` parameter and the `svmName` value for the `svmName` parameter.

2. Select the aggregate

Perform the workflow [Get aggregates](#) and choose the name for the `aggregateName` value.

3. Choose the size for the disk

Choose the size value for the `size:size` parameter. The `size:unit` must be one of the following: TB, GB, MB, KB, or Byte.

4. Choose the iscasInfo parameters

You must choose the following values for the REST API call:

- A unique `igroup` name for `igroupCreationRequest` → `igroupName` parameter
- The required `iqn's` to `igroupCreationRequest` → `initiators` parameter.
- The required operating system for the `osName` parameter from one of the following:
 - windows
 - linux
 - vmware
 - windows_2008
 - windows_gpt

5. Create the quote

Perform the workflow [Create quote](#). This is a recommended step but is not mandatory.

6. Create the volume

HTTP method	Path
POST	<code>/occm/api/vsa/volumes</code>

curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/vsa/volumes' --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, including:

- <WORKING_ENV_ID> (workingEnvironmentId)
- <SVM_NAME> (svmName)
- <AGGR_NAME> (aggregateName)

If aggregate name does not exist, you can set the `createAggregateIfNotFound` query parameter to `true` which allows the aggregate not-found condition.

JSON input example

```
{
  "workingEnvironmentId": "VsaWorkingEnvironment-SfpVUZSc",
  "svmName": "svm_zivaws02we01",
  "aggregateName": "aggr1",
  "name": "zivaws02we01vol01Iscsi",
  "size": {
    "size": 100,
    "unit": "GB"
  },
  "iscsiInfo": {
    "igroupCreationRequest": {
      "igroupName": "zivIgroup",
      "initiators": [
        "iqn.1994-05.com.redhat:96de86825216",
        "iqn.1994-05.com.redhat:96de86823426"
      ]
    },
    "osName": "linux"
  },
  "snapshotPolicyName": "default",
  "enableThinProvisioning": true,
  "enableCompression": true,
  "enableDeduplication": true,
  "maxNumOfDisksApprovedToAdd": 0
}
```


Output

None

Create volume using iSCSI with an existing iGroup



If the properties `aggregateName` and `maxNumOfDisksApprovedToAdd` are not provided on the REST API call, the response will fail with a suggested name for the aggregate and the number of disks needed to fulfill the request.

1. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` value for the `workingEnvironmentId` parameter and the `svmName` value for the `svmName` parameter.

2. Select the aggregate

Perform the workflow [Get aggregates](#) and choose the name for the `aggregateName` value.

3. Choose the size for the disk

Choose the size value for the `size:size` parameter. The `size:unit` must be one of the following: TB, GB, MB, KB, or Byte.

4. Choose the iGroup

Perform the workflow [Get iGroups](#) and choose the `igroups` for the `iscasiInfo → igroups` value. Also select the `osType` value for the `iscasiInfo → osName`.

5. Create the quote

Perform the workflow [Create quote](#). This is a recommended step but is not mandatory.

6. Create the volume

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

curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/vsa/volumes' --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, including:

- `<WORKING_ENV_ID>` (`workingEnvironmentId`)
- `<SVM_NAME>` (`svmName`)

- <AGGR_NAME> (aggregateName)

If an aggregate name does not exist, you can set the `createAggregateIfNotFound` query parameter to `true` which allows the aggregate not-found condition.

JSON input exmaple

```
{
  "workingEnvironmentId": "VsaWorkingEnvironment-UvFmWXoD",
  "svmName": "svm_zivaws01we01",
  "aggregateName": "aggr1",
  "name": "zivaws01we01vol05Iscsi",
  "size": {
    "size": 100,
    "unit": "GB"
  },
  "iscsiInfo": {
    "igroups": ["zivIgroup1"],
    "osName": "linux"
  },
  "snapshotPolicyName": "default",
  "enableThinProvisioning": true,
  "enableCompression": true,
  "enableDeduplication": true,
  "maxNumOfDisksApprovedToAdd": 0
}
```

Output

None

Get volumes

You can retrieve the list of volumes.

1. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` value for the working environment used in the `workingEnvironmentId` path parameter.

2. Get the volumes

HTTP method	Path
GET	/occm/api/vsa/volumes

curl example

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

Input

Query parameter <WORKING_ENV_ID> (workingEnvironmentId)

Output

The JSON output example includes the list of volumes for the working environment.

JSON output example

```
[
  {
    "name": "ziv02vol01",
    "uuid": "cb488216-5bd1-11eb-8a9b-615eb82c79d8",
    "svmName": "svm_ziv01we02",
    "size": {
      "size": 100.0,
      "unit": "GB"
    },
    "usedSize": {
      "size": 5.18798828125E-4,
      "unit": "GB"
    },
    "junctionPath": "/ziv02vol01",
    "volumeTotalInodes": 3112959,
    "volumeUsedInodes": 96,
    "mountPoint": "172.31.1.199:/ziv02vol01",
    "compressionSpaceSaved": {
      "size": 0.0,
      "unit": "GB"
    },
    "deduplicationSpaceSaved": {
      "size": 0.0,
      "unit": "GB"
    },
    "thinProvisioning": true,
    "compression": true,
    "deduplication": true,
    "snapshotPolicy": "default",
    "securityStyle": "unix",
    "exportPolicyInfo": {
      "name": "export-svm_ziv01we02-ziv02vol01",
```

```

    "policyType": "custom",
    "ips": [
        "172.31.0.0/16"
    ],
    "nfsVersion": [
        "nfs3"
    ]
},
"shareNames": [],
"shareInfo": [],
"parentVolumeName": "",
"rootVolume": false,
"state": "online",
"volumeType": "rw",
"aggregateName": "aggr1",
"parentSnapshot": null,
"autoSizeMode": "grow",
"maxGrowSize": {
    "size": 1100.0,
    "unit": "GB"
},
"providerVolumeType": "gp2",
"cloneNames": [],
"moving": false,
"primaryNoFailoverMountPoint": null,
"secondaryNoFailoverMountPoint": null,
"capacityTier": null,
"capacityTierUsedSize": null,
"cifsShareAccessPoint": null,
"primaryCifsShareAccessPoint": null,
"secondaryCifsShareAccessPoint": null,
"tieringPolicy": "none",
"tierInactiveUserData": {
    "size": 0.0,
    "unit": "GB"
},
"tierInactiveUserDataPercent": 0,
"comment": null,
"qosPolicyGroupName": null,
"snaplockType": "non_snaplock",
"constituentsAggregates": [],
"snapshotsUsedSize": {
    "size": 1597440.0,
    "unit": "Byte"
},
},
"cbsBackupsInfo": null,

```

```

        "minimumCoolingDays": null,
        "targetName": "iqn.1992-
08.com.netapp:sn.7d147b755bd011ebb076ef46475a0933:vs.2",
        "iscsiEnabled": false,
        "isFlexGroupVolume": false
    }
]

```

Modify volume

You can modify the configuration of an existing volume.

1. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` value for the working environment and `svmName`.

2. Select the volume

Perform the workflow [Get volumes](#) and choose the `name` for the volume.

3. Modify the volume

HTTP method	Path
PUT	/occm/api/vsa/volumes/{workingEnvironmentId}/{svmName}/{volumeName}

curl example

```

curl --location --request PUT
'https://cloudmanager.cloud.netapp.com/occm/api/vsa/volumes/<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>` (`svmName`)
- `<VOLUME_NAME>` (`volumeName`)

The JSON input example includes the minimum list of input parameters.

JSON input example

```
{
  "exportPolicyInfo": {
    "policyType": "custom",
    "ips": [
      "172.31.0.0/16"
    ],
    "nfsVersion": [
      "nfs3"
    ]
  }
}
```

Output

None

Delete volume

You can delete an existing volume.

1. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` value for the working environment and `svmName`.

2. Select the volume

Perform the workflow [Get volumes](#) and choose the `name` for the volume.

3. Delete the volume

HTTP method	Path
DELETE	/occm/api/vsa/volumes/{workingEnvironmentId}/{svmName}/{volumeName}

curl example

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

Input

Path parameters:

- `<WORKING_ENV_ID>` (`workingEnvironmentId`)

- <SVM_NAME> (svmName)
- <VOLUME_NAME> (volumeName)

Output

None

Create quote

You can create a quote for a new volume which returns a resource quote needed to satisfy the request. The resource quote contains aggregate information where the volume will be created and confirms if the space is available. This is a recommended step but is not mandatory.

1. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` value for the working environment and `svmName`.

2. Select the aggregate

Perform the workflow [Get aggregates](#) and choose the `name` value.

3. Select the permutations configuration

Perform the workflow [Get permutations](#) and choose the `size` and `diskType` values for step 4.

4. Generate the volume quote

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

curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/vsa/volumes/quote'
--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 list of input parameters.

JSON input example

```

{
  "workingEnvironmentId": "VsaWorkingEnvironment-5dZfyKS5",
  "svmName": "svm_ziv01we01",
  "aggregateName": "ziv01agg01",
  "name": "ziv1vol02",
  "size": {
    "size": "100",
    "unit": "GB"
  },
  "enableThinProvisioning": "true",
  "providerVolumeType": "gp2",
  "verifyNameUniqueness": "true"
}

```

Output

The JSON output example includes an example of the quote details.

JSON output example

```

{
  "numOfDisks": 0,
  "diskSize": {
    "size": 100.0,
    "unit": "GB"
  },
  "aggregateName": "ziv01agg01",
  "newAggregate": false,
  "autoVsaCapacityManagement": true
}

```

Get iGroups

You can use this workflow to retrieve all the iGroups.

1. Select the working environment

Perform the workflow [Get working environments](#) and choose the `publicId` and `svmName` values for the working environment `workingEnvironmentId` and `svmName` path parameters.

2. Get the CIFS configurations

HTTP method	Path
GET	/occm/api/vsa/volumes/igroups/{workingEnvironmentId}/{svmName}

curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/vsa/volumes/igroups/<WORKI
NG_ENV_ID>/<SVM_NAME>' --header 'x-agent-id: <AGENT_ID>' --header
'Authorization: Bearer <ACCESS_TOKEN>' --header 'Content-Type:
application/json'
```

Input

- Path parameter <WORKING_ENV_ID>
- Path parameter <SVM_NAME>

Output

The JSON output example includes a list of iGroups.

JSON output example

```
[
  {
    "igroupName": "zivIgroup1",
    "osType": "linux",
    "portsetName": "",
    "igroupType": "iscsi",
    "initiators": [
      "iqn.1994-05.com.redhat:1d9ac633937c"
    ]
  },
  {
    "igroupName": "zivIgroup2",
    "osType": "linux",
    "portsetName": "",
    "igroupType": "iscsi",
    "initiators": [
      "iqn.1994-05.com.redhat:96de86825216"
    ]
  }
]
```

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.