Volumes

Cloud Manager Automation

NetApp
February 16, 2022

Table of Contents

Volumes

Create a volume using NFS

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

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 environement and choose the `publicId` and the `svmName` values for the `workingEnvironmentId` and the `svmName` parameters.

2. Select the aggregate

Perform the workflow Get aggregates and choose the `name` value of the aggregate for the `name` parameter.

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. 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. Select the region

Perform the workflow Get regions workflow and pick `ipCidrRange` value of the required `region→ subnets` for `exportPolicy→ips` value.

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/gcp/vsa/volumes

curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/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-sfrf3wvj",
  "svmName": "svm_zivgcp01we02",
  "aggregateName": "ziv01agg01",
  "name": "zivagg01vol01",
  "size": {
    "size": 100,
    "unit": "GB"
  },
  "snapshotPolicyName": "default",
  "enableThinProvisioning": true,
  "enableCompression": true,
  "enableDeduplication": true,
  "maxNumOfDisksApprovedToAdd": 0,
  "exportPolicyInfo": {
      "ips": [
          "10.138.0.0/20"
      ],
      "nfsVersion": [
          "nfs3", "nfs4"
      ],
      "policyType": "custom"
  }
}
```

Output

None

Create a 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/gcp/vsa/volumes

curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/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)` string
- <SVM_NAME> `(svmName)` string
- <AGGR_NAME> `(aggregateName)` string

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 a 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 iscasiInfo 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	/occm/api/gcp/vsa/volumes

curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/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)` string
- <SVM_NAME> `(svmName)` string
- <AGGR_NAME> `(aggregateName)` string

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/gcp/vsa/volumes

curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/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)` string
- <SVM_NAME> `(svmName)` string

- <AGGR_NAME> `(aggregateName)` string

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 of a single node Azure working environment.

1. Select the working environment

Perform the workflow Get working environment and choose the `publicId` value of the working environment for the `workingEnvironmentId` query parameter.

2. Get the volumes

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

curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/volumes?workingEnv
ironmentId=<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)` string

Output

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

JSON output example


```
[
    {
        "name": "zivagg01vol01",
        "uuid": "0x00000-0xx0-00xx-xx00-00xxxx000",
        "svmName": "svm_zivgcp01we02",
        "size": {
            "size": 100.0,
            "unit": "GB"
        },
        "usedSize": {
            "size": 2.93731689453125E-4,
            "unit": "GB"
        },
        "junctionPath": "/zivagg01vol01",
        "volumeTotalInodes": 3112959,
        "volumeUsedInodes": 96,
        "mountPoint": "10.138.0.150:/zivagg01vol01",
        "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_zivgcp01we02-zivagg01vol01",
```

```
            "policyType": "custom",
            "ips": [
                "10.138.0.0/20"
            ],
            "nfsVersion": [
                "nfs3",
                "nfs4"
            ]
        },
        "shareNames": [],
        "shareInfo": [],
        "parentVolumeName": "",
        "rootVolume": false,
        "state": "online",
        "volumeType": "rw",
        "aggregateName": "ziv01agg01",
        "parentSnapshot": null,
        "autoSizeMode": "grow",
        "maxGrowSize": {
            "size": 1100.0,
            "unit": "GB"
        },
        "providerVolumeType": "pd-ssd",
        "cloneNames": [],
        "moving": false,
        "primaryNoFailoverMountPoint": null,
        "secondaryNoFailoverMountPoint": null,
        "capacityTier": null,
        "capacityTierUsedSize": null,
        "cifsShareAccessPoint": null,
        "primaryCifsShareAccessPoint": null,
        "secondaryCifsShareAccessPoint": null,
        "tieringPolicy": "none",
        "tierInactiveUserData": null,
        "tierInactiveUserDataPercent": null,
        "comment": null,
        "qosPolicyGroupName": null,
        "snaplockType": "non_snaplock",
        "constituentsAggregates": [],
        "snapshotsUsedSize": {
            "size": 0.0,
            "unit": "Byte"
        },
        "cbsBackupsInfo": null,
        "minimumCoolingDays": null,
        "targetName": "iqn.1992-
```

```
08.com.netapp:sn.986656ab5e3e11eb9cb735a0758d479a: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 environment and choose the `publicId` and `svmName` values of the working environment for used as the `workingEnvironmentId` and `svmName` path parameters.

2. Select the volume

Perform the workflow Get volumes and choose the `name` for the `volumeName` path parameter.

3. Modify the volume

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

curl example

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

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

JSON input example

```
{
  "exportPolicyInfo": {
    "policyType": "custom",
    "ips": [
      "10.000.0.0/20"
    ],
    "nfsVersion": [
      "nfs3"
    ]
  }
}
```

Output

None

Delete volume

You can delete an existing volume.

1. Select the working environment

Perform the Get working environment workflow and choose the `publicId` and `svmName` values of the required working environment for `workingEnvironmentId` and `svmName` path parameters.

2. Select the volume

Perform the workflow Get volumes and choose the `name` for the `volumeName` path parameter.

3. Delete the volume

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

curl example

```
curl --location --request DELETE
'https://cloudmanager.cloud.netapp.com/occm/api/azure/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)` string

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

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 environment and choose the `publicId` and 'svmName' values of the required working environment for `workingEnvironmentId` and `svmName` parameters in the JSON input.

2. Select the aggregate

Perform the Get aggregates workflow and choose the `name` value of the required aggregate for the `aggregateName` parameter in the JSON input.

3. Select the gcp disk type

Perform the Get gcp disk types workflow and choose the `size` and `supportedDiskType` values of the required disk type for the `diskSize` and `providerVolumeType` parameters in the JSON input.

4. Generate the volume quote

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

curl example

```
curl --location --request POST
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/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-sfrf3wvj",
  "svmName": "svm_zivgcp01we02",
  "aggregateName": "ziv01agg01",
  "name": "zivagg01vol01",
  "size": {
    "size": "100",
    "unit": "GB"
  },
  "enableThinProvisioning": "true",
  "providerVolumeType": "pd-ssd",
  "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 initiator groups (iGroups).

1. Create the working environment

Perform the workflow Create GCP single node working environment 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/gcp/vsa/volumes/igroups/{workingEnvironmentId}/{svmName}

curl example

```
curl --location --request GET
'https://cloudmanager.cloud.netapp.com/occm/api/gcp/vsa/volumes/igroups/<W
ORKING_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> `workingEnvironmentId` string
- Path parameter <SVM_NAME> `svmName` string

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"
        ]
    }
]
```