



Cloud Volumes APIs

Cloud Volumes Service

Tom Onacki, Julie Chen
February 08, 2021

Table of Contents

- Cloud Volumes APIs 1
 - Finding the API URL, API key, and Secret key 1
 - Listing the available APIs 1
 - Using the Cloud Volumes APIs 1

Cloud Volumes APIs

The Cloud Volumes capabilities that are available through the web UI are also available through RESTful APIs. The APIs enable you to create and manage cloud volumes and develop provisioning scripts and tools.

Finding the API URL, API key, and Secret key

You need to obtain the Cloud Volumes API URL, API key, and Secret key for running an API call.

Steps

1. Click **API access** on the storage page or in the drop-down menu under your username.
2. Record the Cloud Volumes API URL, API key, and Secret key.

[Sample file showing the API URL, API key, and Secret key for an account](#)

Listing the available APIs

The storage page displays the available APIs that you can use.

Steps

1. Click **API documentation** on the storage page.

The page lists the available APIs.

2. Scroll through the page to see the available APIs.

The APIs are listed by function, for example:

- `volumes`
- `mounttargets`
- `storage`
- `snapshots`

3. To obtain details and examples of how to use an API call, select the function and click one of the following actions:
 - `GET`: reads
 - `POST`: creates
 - `PUT`: updates or modifies
 - `DELETE`: destroys

Using the Cloud Volumes APIs

This section shows you how to use the Cloud Volumes APIs. The examples use curl from a Linux bash shell. You need to replace `<api_url>`, `<api_key>`, and `<secret_key>` with the values you recorded from [Finding the API URL, API key, and Secret key](#).

Syntax

```
curl -s -H accept:application/json -H "Content-type: application/json" -H api-key:<api_key> -H secret-key:<secret_key> -X [GET,POST,PUT,DELETE] <api_url>/v2/<command>
```

Examples

Listing volumes

The following example displays information about all volumes:



Piping the command through `jq` improves the formatting of the `json` output. You might need to install `jq` on your system.

```
curl -s -H accept:application/json -H "Content-type: application/json" -H api-key:<api_key> -H secret-key:<secret_key> -X GET <api_url>/v2/Volumes | jq
```

[Script to list cloud volumes in an account](#)

Listing the details for a specific volume

Each volume has an ID called `volumeId`, for example, `07c9ab6c-b655-a9fe-f904-b9b97ef9baaa`. Including the ID in the API call provides details for the specific volume:

```
curl -s -H accept:application/json -H "Content-type: application/json" -H api-key:<api_key> -H secret-key:<secret_key> -X GET <api_url>/v2/Volumes/<volumeId> | jq
```

Creating a volume

The following example uses a `POST` call to create a volume called `Test`, in region `us-west-1`, with an allocated capacity of 100 GB and exported using `nfsv3`:

```
curl -s -H accept:application/json -H "Content-type: application/json" -H
api-key:<api_key> -H secret-key:<secret_key> -X POST <api_url>/v2/Volumes
-d '{
  "name": "Test",
  "creationToken": "grahams-test-volume3",
  "region": "us-west-1",
  "serviceLevel": "standard",
  "quotaInBytes": 10000000000,
  "exportPolicy": {"rules": [{"ruleIndex": 1, "allowedClients":
"0.0.0.0/0", "unixReadOnly": false, "unixReadWrite": true, "cifs": false
, "nfsv3": true, "nfsv4": false}]},
  "protocolTypes": ["NFSv3"],
  "labels": ["test"]
}'
```

Script to create a cloud volume

Updating a volume

The following example uses a `PUT` call to update a volume called `Test`, change the service level to `extreme`, and change the allocated capacity to 600 GB:

```
curl -s -H accept:application/json -H "Content-type: application/json" -H
api-key:<api_key> -H secret-key:<secret_key> -X PUT <api_url>/v
2/Volumes/<volumeId> -d '{
  "serviceLevel": "extreme",
  "quotaInBytes": 60000000000
}'
```

Script to update a cloud volume

Deleting a volume

The following example uses a `DELETE` call to delete a volume specified by `volumeId`:

```
curl -s -H accept:application/json -H "Content-type: application/json" -H
api-key:<api_key> -H secret-key:<secret_key> -X DELETE <api_url>/v
2/Volumes/<volumeId>
```

Script to delete a cloud volume by mountpoint



Use with caution. This API call deletes the volume and all its data.

Creating a snapshot

The following example uses a `POST` call to create a snapshot called `snappy` for a specific volume:

```
curl -s -H accept:application/json -H "Content-type: application/json" -H
api-key:<api_key> -H secret-key:<secret_key> -X POST <api_url>/v
2/Volumes/<volumeId>/Snapshots -d '{
  "name": "<snapshot-name>"
}'
```

[Script to create snapshots of a cloud volume by mountpoint](#)

Creating a snapshot policy

The following example uses a `PUT` call to create snapshot policies for a specific volume:

```
curl -s -H accept:application/json -H "Content-type: application/json" -H
api-key:<api_key> -H secret-key:<secret_key> -X PUT <api_url>/v
2/Volumes/<volumeId> -d '{
  "snapshotPolicy": {
    "dailySchedule": {},
    "enabled": true,
    "hourlySchedule": {
      "minute": 33,
      "snapshotsToKeep": 24
    },
    "monthlySchedule": {},
    "weeklySchedule": {}
  }
}'
```

[Script to create snapshot policies for a cloud volume by mountpoint](#)

Listing snapshots for a specific volume

The following example uses a `GET` call to list the snapshots for a specific volume:

```
curl -s -H accept:application/json -H "Content-type: application/json" -H
api-key:<api_key> -H secret-key:<secret_key> -X GET <api_url>/v
2/Volumes/<volumeId>/Snapshots
```

[Script to list snapshots of a cloud volume by mountpoint](#)

Reverting a snapshot

The following example uses a `POST` call to revert a volume from a snapshot specified by `snapshotId` and `volumeId`:

```
curl -s -H accept:application/json -H "Content-type: application/json" -H
api-key:<api_key> -H secret-key:<secret_key> -X POST <api_url>/v
2/Volumes/<volumeId>/Revert -d '{
  "snapshotId": "<snapshotId>"
}'
```

Script to revert to a snapshot of a cloud volume by mountpoint and snapshotId



Use with caution. This API call causes any data written after the date of that snapshot to be lost.

Creating a new volume from a snapshot

The following example uses a `POST` call to create a new volume based on a snapshot of an existing volume, specified by `snapshotId`:

```
curl -s -H accept:application/json -H "Content-type: application/json" -H
api-key:<api_key> -H secret-key:<secret_key> -X POST <api_url>/v2/Volumes
-d '{
  "snapshotId": "<snapshotId>",
  "name": "Copy",
  "creationToken": "perfectly-copied-volume",
  "region": "us-west-1",
  "serviceLevel": "extreme",
  "protocolTypes": ["NFSv3"]
}'
```

Script to copy a cloud volume

Deleting a snapshot

The following example uses a `DELETE` call to delete a snapshot specified by `snapshotId`:

```
curl -s -H accept:application/json -H "Content-type: application/json" -H
api-key:<api_key> -H secret-key:<secret_key> -X DELETE <api_url>/v
2/Volumes/<volumeId>/Snapshots/<snapshotId>
```

Script to delete a snapshot of a cloud volume by mountpoint and snapshotId



Use with caution. This API call deletes the snapshot and all its data.

Joining a directory service

The following example uses a **POST** call to join a directory service and provides the DNS IP address, domain, the NetBIOS name for the SMB server, the username and password for a directory service admin, and the organizational unit (optional and defaults to CN=Computers).

```
curl -s -H accept:application/json -H "Content-type: application/json" -H
api-key:<api_key> -H secret-key:<secret_key> -X POST <api_url/v
2/Storage/ActiveDirectory -d '{
  "DNS": "<ip-address>",
  "domain": "<domain>",
  "netBIOS": "<netbios-name>",
  "organizationalUnit": "OU=Cloud Servers,DC=nas-cloud,DC=local",
  "password": "secret",
  "region": "us-west-1",
  "username": "Administrator"
}'
```

[Script to join a directory service](#)

Viewing directory service integration

The following example uses a **GET** call to display the configuration for directory service integration.

```
curl -s -H accept:application/json -H "Content-type: application/json" -H
api-key:<api_key> -H secret-key:<secret_key> -X GET <api_url/v
2/Storage/ActiveDirectory
```

[Script to view directory service integration](#)

Unjoining a directory service

The following example uses a **DELETE** call to unjoin a directory service integration. This requires the UUID for the current join, which can be found using the **GET** call listed above.



You cannot unjoin a directory service that is in use; status "in use".

```
curl -s -H accept:application/json -H "Content-type: application/json" -H
api-key:<api_key> -H secret-key:<secret_key> -X DELETE <api_url/v
2/Storage/ActiveDirectory/<UUID>
```

[Script to unjoin a directory service](#)

Get performance statistics

The following example uses a `GET` call to list the read and write IOPS, throughput, and latency statistics over a specific time period for a volume specified by `volumeId`.

```
curl -s -H accept:application/json -H 'Content-type: application/json' -H
api-key:<api_key> -H secret-key:<secret_key> -X GET '<api_url>/v
2/Volumes/<volumeId>/PerformanceMetrics?startDate=2021-02-05T09:
00&endDate=2021-02-05T09:
05&type=READ_IOPS,WRITE_IOPS,TOTAL_THROUGHPUT,AVERAGE_OTHER_LATENCY'
```

[Script to get performance statistics of a cloud volume by mountpoint](#)

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.